


IDENTIFICAÇÃO NEURAL DE PARTÍCULAS EM CALORIMETRIA DE
ALTAS ENERGIAS

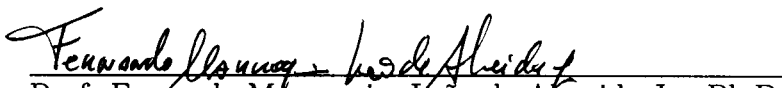
Denis Oliveira Damazio

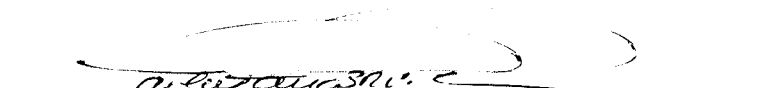
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

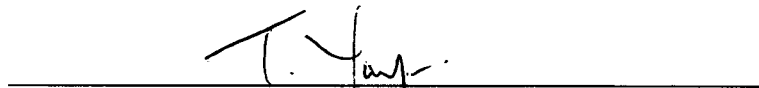
Aprovada por:


Prof. José Manoel de Seixas, D.Sc.


Prof. Luiz Pereira Caloba, Dr.Ing.


Prof. Fernando Marroquim Leão de Almeida Jr., Ph.D.


Prof. Marley Maria B.R. Vellasco, Ph.D.


Prof. Takashi Yoneyama, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

DEZEMBRO DE 2002

OLIVEIRA DAMAZIO, DENIS

Identificação Neural de Partículas em Calorimetria de Altas Energias [Rio de Janeiro] 2002

XXV, 251 pp 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Elétrica, 2002)

Tese - Universidade Federal do Rio de Janeiro, COPPE

- 1.Redes Neurais
- 2.Física de Altas Energias
- 3.Análise de Dados
- 4.Processamento em tempo real

I.COPPE/UFRJ II.Título (série)

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

IDENTIFICAÇÃO NEURAL DE PARTÍCULAS EM CALORIMETRIA DE ALTAS ENERGIAS

Denis Oliveira Damazio

Dezembro/2002

Orientador: José Manoel Seixas

Programa: Engenharia Elétrica

Durante testes de calibração de um detector da física de altas energia, um calorímetro, módulos do mesmo são submetidos a feixes de partículas. No processo de produção dos feixes, quando selecionada uma dada classe de partículas, podem ser identificados eventos de outros tipos. Sendo assim, há a necessidade de um método de identificação de eventos.

A abordagem discutida foi a utilização de um classificador neural treinado pelo método de *backpropagation*. Este é um método supervisionado, logo, depende de uma prévia etiquetagem dos eventos que não é possível de se obter dada a contaminação. Entretanto, a caracterização na informação proveniente do calorímetro consegue minorar este problema do treinamento e levar o classificador a um excelente nível de identificação. Diferentes estudos são realizados visando uma caracterização dos limites de aplicação da técnica.

Dada a dispersão geográfica dos pesquisadores da colaboração responsável pelo calorímetro, foi implementado um sistema neural classificador através de páginas na *Web*, disponibilizando a metodologia estudada para a colaboração.

Foi também implementado o treinamento neural online, durante a aquisição de dados, quando foi possível manter o alto nível de classificação sem impacto para a taxa de aquisição do sistema.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

NEURAL PARTICLE IDENTIFICATION IN HIGH-ENERGY CALORIMETRY

Denis Oliveira Damazio

December/2002

Advisor: José Manoel Seixas

Department: Electrical Engineering

During calibration tests of a high-energy physics detector, a calorimeter, modules are submitted to particle beams. In the beam production, when selected the particle class, events from other particle types can be observed, jeopardizing the calibrations to be performed. So, it is important to establish an event identification procedure.

The methodology presented was based in using a neural classifier trained by the backpropagation algorithm. This is a supervised method, so, depends on a previous labeling of the events. This is not possible due to the data contamination. However, the feature extraction from the calorimeter information can reduce this training problem and leads the calorimeter to a high identification level. Different studies are performed in order to verify the limits of the technique.

Due to the geographical dispersion of the researchers in the international collaboration responsible for the calorimeter, a neural classifier via Web pages was implemented, making the discriminator a usable tool.

It was also implemented the online neural network training, during data acquisition, when it was possible to verify the classification level without impacting the data acquisition rate.

”Não, nosso senhor,
Não há de ter lançado em movimento terra e céu,
Estrelas percorrendo o firmamento em carrossel
Pra circular em torno ao Criador.”

Chico Buarque, disco ”Paratodos”.

”Et quand’on devra t’ensevelir,
Sur la pierre on pourra lire,
Mais qui pourra lire?”

Gerald de Palmas. disco ”Marcher dans le Sable”

”We read to know we’re
not alone!”

Do filme **Shadowlands**

”Mas meu último desejo,
você não pode negar.”

Noel Rosa

Agradecimentos

Inicialmente, agradeço à minha mãe. Primeiro porque do eixo da mãe raia o filho e segundo porque se perde na noite dos tempos o momento em que ela me apresentou "Os Doze Trabalhos de Hércules", estimulando em mim o gosto pela leitura, a curiosidade pelos conhecimentos e muito mais. O Sérgio também esteve muito presente, colaborando do jeito dele, em muitas coisas.

Agradeço também a minha família, tia Shirley, Sheily e Xaninha e toda a turma que vem junto delas. Embora nem sempre possamos estar próximos, a torcida deste pessoal nunca deixou de ser fundamental para o meu trabalho. Mais recentemente, a família de minha esposa também entrou pra valer neste time, reforçando os votos. Minha irmã, mesmo de longe, nunca deixou de estar presente.

Na seqüência, agradeço ao pessoal que me deu apoio mais direto neste trabalho. Algumas das pessoas que merecem destaque são Robert (Bob) Stanek, Alexandre Solodkov, Mihai Caprini, Richard Teuscher, Robert Jones entre outros amigos do CERN. No Brasil, contei com o apoio dos professores Sérgio Villas-Boas, Carmen Lodi Maidantchik, Paulo Sérgio Ramirez Diniz, Antônio Petraglia, Luiz Wagner Pereira Biscainho entre muitos outros. Com especial atenção, dedico este trabalho também ao professor Roosevelt José Dias, um exemplo da pessoa dedicada a aprender e ensinar.

Agradeço também aos amigos, alguns ficaram no meio do caminho mas não deixam de ter sido importantes. Neste time estão Gustavo Golçalves Gomes, Gustavo Frank Sassi, Ricardo Araujo Silva, Gerson Barbosa, João Batista de Oliveira Souza Filho, Paulo Vitor Magacho, William Soares Filho e muitos outros. Em destaque, coloco meu grande amigo D.Sc. Augusto Santiago Cerqueira, companheiro de viagem durante toda esta tese e com quem dividi os vários momentos complicados que estão associadas ao tipo de vida que levamos, particularmente na estadia no

CERN.

Agradeço também aos professores da banca. Aos professores Ph.D. Fernando Marroquim Leão de Almeida Jr e Dr. Ing. Luiz Pereira Caloba, que nunca deixaram de acompanhar de perto este trabalho, dando, muitas vezes, sugestões que valeram pontos importantes ao prosseguimento do trabalho. Sem contar a convivência pessoal com ambos, é claro, quando tinha a oportunidade de me relacionar com estes grandes amigos. Agradeço também a professora Ph.D. Marley Maria B.R. Vellasco, cujas sugestões dadas na defesa de tema de tese foram fundamentais para melhorar a qualidade da tese em si. Ao professor Ph.D Takashi Yoneyama, agradeço pelas perguntas fundamentais para o aprimoramento final da tese.

Agradeço, finalmente, ao trabalhador incansável e preocupado que foi o meu orientador, o professor D.Sc José Manoel de Seixas. Ele mostrou a importância de se valorizar ao máximo aquilo que se faz e que não se deve deixar de, pelo menos, perseguir a perfeição até onde seja possível. Além disso, ele nunca deixou de me apoiar, dentro e fora da área de trabalho. Também foi graças a ele que conheci o melhor lugar do mundo para desenvolvimento do meu potencial de pesquisa, o CERN. Seixas é, muito mais do que orientador, um dos maiores amigos que fiz neste tempo.

Agradeço, principalmente, à minha esposa, Paula Carnevale Damazio, que tem compartilhado comigo a vida. Seu papel vai muito além daquele esperado de uma esposa. Paula me ajuda a manter os pés no chão e ver que existe vida além do trabalho. Paula me ensina todo dia coisas novas sobre convivência a dois, cuidado e carinho. Finalmente, quando a física e a matemática fazem falta, em Paula sobra amor, atenção, paciência e admiração para continuar a meu lado, buscando não deixar nenhuma brecha para o descuido. Por isso, a cada dia, amo mais Paulinha.

Agradeço, por último, àquele que virá, o que muito me motiva e estimula.

18 de dezembro de 2002

Denis Oliveira Damazio

Sumário

1	Introdução	1
1.1	O ambiente de Física Experimental	2
1.2	O que foi Realizado	4
1.3	Organização do Documento	5
2	Ambiente da Física de Altas Energias	7
2.1	A Física de Partículas	7
2.2	O CERN	11
2.3	O LHC	12
3	O Detector TileCal e a Contaminação do Feixe de Partículas	18
3.1	O Detector	18
3.1.1	Topologia do Calorímetro e Arranjo Experimental	19
3.1.2	Leitura do Sinal Luminoso das Fibras	22
3.2	Deposição de Energia	27
3.2.1	Contaminação no Feixe	31
4	O Sistema de Aquisição de Dados	35
4.1	O DAQ para Testes do TileCal	35
4.2	Exercícios no DAQ para o TileCal	41
4.2.1	O Sistema de Monitoração <i>Online</i>	42
4.2.2	Transmissão de Dados no Barramento VME	46
4.2.3	Compactação de Dados	48
5	Análise <i>Offline</i>	51
5.1	Análise <i>Offline</i> Inicial	51

5.1.1	Noções Iniciais de Redes Neurais	52
5.1.2	Rede Neural : Estrutura e Treinamento	53
5.1.3	Separação Píon-múon	56
5.1.4	Separação Elétron-píon-múon	61
5.1.5	Resultados para outras energias	71
5.2	Normalizações Alternativas	74
5.2.1	Analisando a Normalização pela Energia Total	75
5.2.2	Normalização pela raiz quadrada	78
5.2.3	Normalização pelo Valor Máximo	83
5.2.4	Normalização por Coordenada Extra	85
5.2.5	Sumário das Normalizações Estudadas	87
5.3	Análise de Parâmetros	88
5.3.1	Estudo da taxa de aprendizado	89
5.3.2	Controlando a taxa de aprendizado	94
5.4	Dados Simulados	100
5.4.1	Simuladores para o LHC	101
5.4.2	Produzindo os Dados Simulados	102
5.4.3	Separação Píon/múon	104
5.4.4	Separação Elétron/píon/múon	108
5.5	Extensões do Barril	110
5.5.1	Arranjo Experimental e Geometria	110
5.5.2	Resultados da Aplicação Neural	112
5.5.3	Estudos das Relevâncias	114
5.6	Análise de Sensibilidade	115
5.7	Estudo das Saídas da Rede Neural	119
5.7.1	Desvio por Múons de Contaminação	121
6	Discriminador Neural Remoto de Partículas	124
6.1	Tecnologia e Propriedades	125
6.1.1	Controle de Usuários	127
6.2	Exemplo de Análise	128
6.2.1	Realizando Análises Pré-definidas com o DNRP	129
6.2.2	Criando sua Própria Análise	131

7	Pré-processamento e Compactação de Sinais	135
7.1	Mapeamentos Topológicos	135
7.1.1	Média da Energia na Célula	136
7.1.2	Torres de <i>Trigger</i>	137
7.1.3	Camadas de Segmentação	142
7.1.4	Anéis Concêntricos	144
7.1.5	Sumário	148
7.2	Redes Especialistas	148
7.2.1	Resultados	152
7.3	Componentes Principais de Discriminação	156
7.3.1	Resultados	157
8	Análise <i>Online</i>	164
8.1	Preparando para a Análise <i>Online</i>	166
8.1.1	Separação Píon/múon	167
8.1.2	Separação Elétron/píon/múon	169
8.2	Implementação da Classificação Neural <i>Online</i>	170
8.2.1	Aplicação Neural no DAQ	173
8.2.2	Estrutura Final da Aplicação Neural	176
8.3	Resultados do Experimento <i>Online</i>	178
9	Conclusões	183
	Referências Bibliográficas	189
A	Artigos e Notas Técnicas	196
B	Código Neural usado para os estudos <i>offline</i>	202
C	Código Neural usado <i>online</i>	224

Lista de Figuras

2.1	Experimento de Rutherford sobre o modelo atômico.	8
2.2	Partículas do Modelo Padrão.	9
2.3	Vista aérea do anel LEP sob várias cidades na fronteira franco-suíça. Extraído de [1].	12
2.4	Desenho esquemático do detector ATLAS. Extraído de [2].	14
3.1	Protótipo do calorímetro hadrônico do ATLAS. Extraído de [3].	19
3.2	Esquema experimental para teste com o protótipo de cinco módulos. Extraído de [3].	20
3.3	Módulos do TileCal a volta do ponto de colisão.	21
3.4	Distribuição de células para o barril (esquerda) e a extensão do barril.	22
3.5	Célula fotomultiplicadora usada no TileCal (esquerda). Sinal produ- zido por esta mesma célula (direita) e já conformado pelo conformador de sinais.	23
3.6	Foto da gaveta com equipamentos eletrônicos do TileCal.	24
3.7	Linha do feixe na área H8 aonde são testados os protótipos do TileCal (esquerda) e arranjo experimental utilizado de 1996 a 1999.	25
3.8	Módulos do TileCal sendo testados na área H8. Vê-se dois módulos da extensão do barril sobre dois módulos do barril (o inferior é o Módulo 0).	26
3.9	Conjuntos de dados experimentais adquiridos com contaminação.	28
3.10	Um evento típico de elétron depositando energia no calorímetro TileCal.	29
3.11	Evento de pión (à esquerda) e de múon (à direita).	30
3.12	Eventos um feixe de píons de 20 GeV com contaminação de múons.	30

3.13	Fração da energia depositada na primeira camada de células do TileCal para eventos de um feixe de elétrons com contaminação por píons (topo) e eventos de um feixe purificado de píons (embaixo). Ambos feixes de 100 GeV.	32
3.14	Contador Cherenkov correlacionado com a energia total depositada no módulo do TileCal para eventos de um feixe de elétrons a 20 GeV contaminado por píons e múons.	33
4.1	Sistema de aquisição de dados do TileCal para testes com feixes.	36
4.2	Janela do sistema de controle do DAQ do TileCal.	39
4.3	Formato de dados para uma ROD do ATLAS.	41
4.4	Organização do sistema de monitoração.	42
4.5	Organização em classes da Atividade de Monitoração.	43
4.6	Janela com histogramas de variáveis do TileCal em HBook e HPlot (esquerda) e ROOT (direita).	44
4.7	Interfaces Gráficas ao Usuário em QT (esquerda) e integrado ao sistema de aquisição (direita).	45
4.8	Medidas da velocidade da transmissão de dados em modo DMA.	47
4.9	Esquema de compactação das amostras do TileCal.	49
4.10	Distribuição do máximo menos mínimo para eventos de múons (topo) e de pedestal (embaixo).	50
5.1	Esquema de um neurônio artificial.	52
5.2	Esquema da rede neural usada no estudo.	54
5.3	Histograma da saída da rede neural para os conjuntos de treino (esquerda) e de teste (direita), para píons (acima) e múons (abaixo).	57
5.4	Correlação entre a saída da rede neural e a energia depositada no módulo.	58
5.5	Relevâncias da rede neural para todas as células do calorímetro.	61
5.6	Saídas da rede neural (colunas da esquerda pra direita : elétrons, píons e múons) para os conjuntos de treino (linhas de cima pra baixo: elétrons, píons e múons).	62

5.7	Correlação entre as saídas de elétrons e píons no conjunto de elétrons (esquerda) e entre as saídas de píons e múons para o conjunto de píons (direita).	63
5.8	Correlação entre a subtração das saídas de elétrons menos a saída de píons e a subtração das saídas de píons menor a de múons para eventos do feixe de elétrons.	65
5.9	Correlação com a energia depositada no detector com a subtração calculada da saída de elétrons menos a de píons para eventos do feixe de elétrons (esquerda), e com a subtração calculada da saída de píons menos a de múons para o feixe de píons (direita).	66
5.10	Correlação entre a saída de elétrons menos a de píons no conjunto de elétrons com a fração de energia depositada na primeira camada do detector (esquerda) e com o valor do contador Cherenkov(direita). . .	67
5.11	Relevâncias para a separação elétron-píon-múon a 20 GeV.	70
5.12	Fração da energia depositada na terceira camada (esquerda) e nos cinco módulos (direita) por um feixe de 100 GeV.	72
5.13	Energia depositada no calorímetro para eventos de 100 GeV de um feixe de elétrons.	73
5.14	Correlação entre a saída da rede neural e a energia depositada no calorímetro por eventos de um feixe de píons (com contaminação) de 20 GeV. Notar o grupo com alta energia, mas considerado múon pela rede neural.	75
5.15	Perfil médio de deposição de energia para múons (esquerda) e para eventos com alta energia depositada e considerados múons pela rede neural (direita).	76
5.16	Fração de energia na terceira camada do calorímetro para eventos provenientes de um feixe de píons de 20 GeV e que depositaram mais de 10 GeV no detector. No topo, temos os eventos que foram considerados múons pela rede neural. No meio, todos os eventos com mais de 10 GeV. Abaixo temos a divisão do primeiro histograma pelo segundo.	77

5.17	Correlação da saída da rede neural com a energia total, quando a raiz quadrada é usada no processo de normalização para eventos de píons a 20 GeV.	80
5.18	Saídas da rede para feixes de 20 GeV, usando a normalização pela raiz quadrada da energia (esquerda). Relevâncias para este classificador (direita).	83
5.19	Correlação entre a saída da rede e a energia depositada no detector (esquerda), para a normalização pelo máximo de energia, considerando-se um feixe contaminado de píons. Análise da deposição de energia na terceira camada (direita), no histograma do topo temos os eventos que depositaram mais de 10 GeV e foram considerados múons pela rede neural. No histograma do meio, temos todos os eventos com mais 10 GeV e, abaixo, a divisão do histograma superior pelo do meio.	84
5.20	Coordenada extra já normalizada (esquerda) e saída da rede neural (direita) para um feixe de píons de 20 GeV, a qual é correlacionada com a energia total depositada no detector.	86
5.21	Relevâncias para as células do detector e coordenada extra na posição 47 (esquerda). Detalhe das relevâncias das células (direita).	87
5.22	Relevâncias, considerando-se a coordenada extra na posição 47, na separação elétron/píon/múon.	88
5.23	Acertos (lado esquerdo) e erros (direito) da rede neural (RN) em relação ao corte em energia (clas) em função da taxa de aprendizado.	90
5.24	Concordância e discordância total em função da taxa de aprendizado para a normalização pela raiz quadrada, separação píon/múon a 20 GeV.	91
5.25	Correlação entre a saída da rede e a energia depositada no detector para $lr = 10^{-3.5}$ (esquerda) e $lr = 10^{-2}$ (direita). Notar dispersão da saída de múons neste segundo caso.	92
5.26	Acertos (lado esquerdo) e erros (direito) da rede neural (RN) em relação ao corte em energia (clas) em função da taxa de aprendizado. Caso com normalização pela energia total.	93

5.27	Curvas com concordância (superior) e discordância (inferior) entre o método clássico e o método neural em função da taxa de aprendizado.	94
5.28	Histograma dos valores mínimos de concordância entre o método neural e o método clássico para 10 diferentes treinamentos com mesmos parâmetros (queda da taxa de aprendizado e época).	95
5.29	Variação da concordância da rede neural para separação pión/múon a 20 GeV em função da queda da taxa de aprendizado e da época (esquerda). Na direita temos as curvas de nível geradas com escala de tons de cinza.	96
5.30	Concordância mínima nos três conjuntos de dados estudados combinadas para a rede com normalização pela raiz quadrada da energia total (esquerda) e pela energia no detector (direita).	97
5.31	Concordância mínima nos três conjuntos de dados estudados combinadas para a rede com normalização pela raiz quadrada da energia total (esquerda) e pela energia no detector (direita). Separação elétron/pión/múon.	98
5.32	Concordância máxima nos três conjuntos de dados estudados combinadas para a rede com normalização pela raiz quadrada da energia total (esquerda) e pela energia no detector (direita). Separação elétron/pión/múon.	100
5.33	Distribuição para elétrons simulados de 20 GeV (esquerda) e a linearidade da simulação (direita).	103
5.34	Módulos da Extensão do Barril configurados para teste. Os cinco módulos também foram utilizados.	110
5.35	Disposição das células para módulos da Extensão do Barril.	111
5.36	Correlação entre a saída da rede, com normalização pela energia total (esquerda) e pela raiz quadrada da energia (direita), com a energia depositada no módulo da extensão do barril para um feixe de píons de 80 GeV.	113
5.37	Relevâncias da energia depositada em cada célula do detector para feixes de 80 GeV (esquerda) e 100 GeV (direita).	115

5.38	Saídas da rede neural para feixes de píons contaminados por múons (histograma superior) e de múons (inferior), considerando-se os conjuntos de treinamento. Feixes de 20 GeV.	119
5.39	Distribuição da saída da rede tomada antes da aplicação da função de ativação do neurônio de saída para os feixes de píons contaminado (histograma superior) e puro de múons (inferior). Feixes de 20 GeV. .	120
5.40	Médias das saídas da rede neural para o feixe puro de múons de 20 GeV (esquerda) e o resultado da aplicação da tangente hiperbólica sobre estas médias (direita).	121
5.41	Curva da tangente hiperbólica das medidas quando a rede é treinada com normalização pela raiz quadrada da energia. A função linear é estimada por processo de <i>fitting</i>	122
6.1	Tela inicial do Discriminador Neural Remoto de Partículas.	125
6.2	Somente algumas opções utilizáveis (esquerda); todas as opções utilizáveis (direita).	126
6.3	Janela para entrada de dados para treino da rede (esquerda) e janela após treino com opções de análise (direita).	128
6.4	Saídas da rede neural para cada conjunto de dados produzida pelo DNRP.	129
6.5	Correlação entre as saídas de elétrons e píons no conjunto de elétrons.	130
6.6	Resultados de uma análise de comparação com método clássico realizada pelo usuário.	131
6.7	Figuras resultantes de uma análise do usuário. Correlação entre saída de elétrons menos a de píons com a fração de energia na primeira camada para o conjunto de elétrons (esquerda), e correlação entre saída de píons menos a de múons com a energia total para o conjunto de píons (direita).	132
6.8	Análise de comparação entre o conjunto de dados descontaminado (em cima) e conjunto contaminado (embaixo).	133

7.1	Saída da rede neural contra a energia absorvida no detector para um feixe de píons de 20 GeV usando 23 entradas com normalização pela raiz quadrada da energia total.	138
7.2	Definição das torres do TileCal. As células da última camada tem cada uma de suas leituras associadas a uma das duas torres que nela tocam.	139
7.3	Relevâncias para a análise com torres de <i>trigger</i> a 20 e 100 GeV (esquerda). Quantidade de energia absorvida pelas torres laterais para 20 e 100 GeV (direita). Em ambos os casos, os três feixes de partículas foram utilizados.	141
7.4	Relevância para as entradas da rede neural na separação elétron-píon-múon a 20 GeV.	143
7.5	Definição das camadas para análise neural.	143
7.6	Definição das estruturas em anel para elétrons (esquerda) e píons (direita).	145
7.7	Histogramas da energia total depositada nos cinco módulos por eventos de elétrons (contaminado por píons) e píons. Em ambos os casos um corte na energia total no módulo em teste permitiu a eliminação da contaminação de múons.	150
7.8	Esquema para extração das componentes de redes especializadas em cada uma das informações vindas separadamente do módulo em teste e dos cinco módulos.	151
7.9	Terceira rede neural baseada nas informações extraídas pelas duas primeiras redes.	151
7.10	Correlação entre a resposta da rede neural (eixo horizontal) treinada apenas com as informações dos cinco módulos e a energia total depositada nestes módulos para o conjunto de elétrons contaminados. . . .	153
7.11	Energia total nos cinco módulos para eventos de elétrons contaminados por píons (acima) e múons (abaixo). Vemos que as médias são próximas.	154

7.12	Energia depositada nos cinco módulos por eventos de elétrons com contaminação de píons (histograma superior) e eventos de píons (histograma inferior).	156
7.13	Esquema para extração da primeira Componente Principal (a) e da m-ésima componente (b).	157
7.14	Concordâncias médias entre rede extraindo Componentes Principais de Discriminação e método clássico para rede com normalização pela soma (esquerda) e pela raiz quadrada (direita).	158
7.15	Evolução das concordâncias para 100 GeV com normalização pela raiz quadrada mostrando rápida ascensão e queda também brusca. . .	159
7.16	Saída de elétrons menos a de múons correlacionada com a energia no detector no final da extração da terceira (esquerda) e da quarta (direita) componentes.	161
7.17	Evolução do erro médio quadrático para cada componente com a normalização pela energia total no módulo (esquerda) e pela raiz da energia total (direita).	162
7.18	Correlação entre a saída de elétrons de uma rede treinada normalmente (8 neurônios na camada escondida) com a rede treinada com Componentes Principais de Discriminação (com apenas 2 neurônios).	163
8.1	Análise da capacidade de generalização da rede com um número restrito de eventos durante o treino. Normalização pela energia total (esquerda) e pela raiz quadrada da energia total (direita).	168
8.2	Análise da capacidade de generalização com número restrito de eventos para treinamento na separação elétron/píon/múon. Normalização pela energia total (esquerda) e pela raiz quadrada da energia (direita).	169
8.3	Inserção do processamento neural na ROD Crate.	174
8.4	Aplicação <i>online</i> da rede neural em duas <i>threads</i> para testes com feixe do TileCal.	177
8.5	Deposição de energia no detector por <i>Runs</i> de píons a 180 GeV com contaminação. <i>Run</i> 120324 (esquerda) e 120326. Notar que a primeira tem uma índice maior de contaminação.	179

8.6	Correlação entre a resposta da rede neural (eixos horizontais) com a energia depositada nos módulos (eixos verticais) para a <i>run</i> 120324 (esquerda) e a <i>run</i> 120326 (direita).	180
8.7	Valores de saída da rede neural para os primeiros eventos da <i>run</i> 120324. Vê-se que parte dos eventos vai para valores entre -0.2 e 0 e que logo começam a ir para os picos em -0.8 e +1 já caracterizando o efeito do treinamento.	181

Lista de Tabelas

5.1	Eficiências de identificação para a separação pión-múon com energia de 20 GeV após 54000 passos de treinamento em 90 épocas de 600 passos de treinamento.	57
5.2	Tabela de comparação do método neural com o método clássico (corte em energia) para o conjunto de píons (esquerda com 91,20% de concordância) e de múons (direita com 97,01% de concordância).	59
5.3	Eficiências de acerto para a separação elétron-pión-múon a 20 GeV.	61
5.4	Correlação entre o método clássico e a rede neural para o feixe de elétrons (alto à esquerda), com 91,26% de concordância, píons (alto à direita) com 86,75% de concordância, e múons (embaixo) com 96,43% de concordância.	68
5.5	Correlação entre o método clássico e o processamento neural para elétrons (alto à esquerda), com 89,08% de concordância, píons (alto à direita), com 88,49% de concordância, e múons (embaixo) com 96,47%.	71
5.6	Concordância para diferentes energias de feixe (em GeV) para o caso pión-múon (esquerda) e o caso elétron-pión-múon (direita).	74
5.7	Concordância com a metodologia clássica quando utilizamos a normalização pela raiz quadrada da energia. Concordâncias para feixes de 20 GeV de píons (esquerda, total 99,05%) e múons (direita, total 98,60%).	80

5.8	Concordância entre os resultados da rede neural, com normalização pela raiz quadrada da energia total, e o método clássico, baseado no contador Cherenkov e no corte em energia. Acima, à esquerda, eventos adquiridos do feixe de elétrons, com concordância de 95,22%. No alto, à direita, eventos de um feixe de píons, com concordância de 93,34%, e, embaixo, eventos do feixe de múons com concordância de 97,41%. a energia dos feixes é de 20 GeV.	81
5.9	Concordâncias entre o método clássico e o processamento neural para diferentes energias, quando se utiliza a normalização baseada em raiz quadrada da energia. Resultados para um feixe contaminado de píons (à esquerda) e para a contaminação no feixe de elétrons (à direita). .	82
5.10	Concordância com a análise clássica para o processamento neural com normalização pelo valor máximo da energia. Na esquerda, temos o caso pión/múon e, na direita, o caso elétron/pión/múon, considerando-se feixes de 20, 100 e 180 GeV.	85
5.11	Resultados para a normalização pela coordenada extra para feixes contaminados de 20, 100 e 180 GeV. Na esquerda, temos o caso pión/múon e na direita o caso elétron/pión/múon.	86
5.12	Desempenho dos métodos de normalização para a separação pión-múon nas várias energias (tabela superior) e para a separação elétron-pión-múon (tabela inferior), com respeito ao método clássico de análise.	89
5.13	Número de pontos no espaço de parâmetros e porcentagem com relação ao total (460 pontos) dos pontos que atingiram mais do que 95% de eficiência.	98
5.14	Número de pontos no espaço de parâmetros e porcentagem com relação ao total (460 pontos) dos pontos que atingiram mais do que 90% e 91% de eficiência. Separação elétron/pión/múon.	99
5.15	Número de pontos no espaço de parâmetros e porcentagem com relação ao total (460 pontos) dos pontos que atingiram mais do que 95% e 96% de eficiência. Separação elétron/pión/múon.	100

5.16	Concordância entre os métodos neural com normalização pela energia total e clássico, para feixes puros de píons (esquerda, com concordância de 97,60%) e múons (direita, com concordância de 98,70%), usando dados simulados pelo GEANT. Feixes de 20 GeV.	105
5.17	Concordância entre os métodos neural, com normalização baseada na raiz quadrada da energia, e clássico, para feixes de 20 GeV puros de píons (esquerda, com concordância total de 99,95%) e múons (direita, com concordância de 99,85%) usando dados simulados pelo GEANT.	105
5.18	Concordância entre o método neural, com normalização baseada em raiz quadrada da energia e o método clássico para feixes de píons (esquerda, com concordância de 100%) e múons (direita, com concordância de 99,90%) usando dados simulados pelo GEANT.	106
5.19	Percentual de identificação para píons e múons pela rede neural nos três métodos (normalização pela energia total, pela raiz da energia e com conjunto de píons contaminado por múons) e concordância com o método clássico usando um feixe de 100 GeV.	107
5.20	Percentual de identificação para píons e múons pela rede neural nos três métodos (normalização pela energia total, pela raiz da energia e com conjuntos de píons contaminados por múons) e concordância com o método clássico usando feixes de 180 GeV.	107
5.21	Identificação das partículas em cada conjunto de dados adquirido de um feixe simulado segundo o método clássico.	108
5.22	Identificação neural para dados simulados em três diferentes energias e as respectivas concordâncias com o método clássico, considerando-se dois métodos de normalização (pela energia total, pela raiz quadrada da energia e a indução de contaminação do feixe).	109
5.23	Concordância da resposta da rede neural com o corte em energia (método clássico) para três energias e dois enfoques de normalizações. Resultados para as Extensões do Barril e dados provenientes de feixes de píons e múons.	113

5.24	Concordância da resposta da rede neural com o corte em energia (método clássico) para diferentes energias e dois enfoques de normalizações. Resultados para a separação elétron/píon/múon nas Extensões do Barril.	114
5.25	Índices de concordância entre os métodos clássico e neural para dados provenientes de feixes de 100 e 180 GeV para um treinamento da rede neural com feixes de 20 GeV. Normalização pela energia total (esquerda) e pela raiz da a energia (direita).	116
5.26	Índices de concordância entre os métodos clássico e neural para dados provenientes de feixes de 20 e 180 GeV para um treinamento da rede neural com feixes de 100 GeV. Normalização pela energia total (esquerda) e pela raiz da a energia (direita).	117
5.27	Índices de concordância entre os métodos clássico e neural para dados provenientes de feixes de 20 e 100 GeV para um treinamento da rede neural com feixes de 180 GeV. Normalização pela energia total (esquerda) e pela raiz da a energia (direita).	118
7.1	Concordância com a metodologia clássica da energia média por célula (23 componentes) para a normalização pela energia total (esquerda) e pela raiz quadrada da energia total(direita) considerando-se fixos de píons contaminados por múons.	137
7.2	Concordância entre o método clássico e o método neural para o caso elétron/píon/múon para diferentes energias quando usada a média da energia por célula. Normalização pela energia total (acima) e pela raiz quadrada da energia (abaixo).	138
7.3	Concordância entre o método clássico e o método neural usando as torres de <i>trigger</i> . Acima, temos os resultados por energia para a separação píon/múon. Em baixo, temos a separação elétron/píon/múon.	140
7.4	Eficiências para sistema neural com pré-processamento híbrido com torres e células. Acima, temos o caso píon/múon e abaixo o caso elétron/píon/múon para as energias consideradas.	142

7.5	Concordância entre método clássico e processamento neural para análise por camadas. Acima, a análise píon/múon. Abaixo, análise elétron/píon/múon.	144
7.6	Comparação entre os métodos baseados em anéis para redes neurais e a metodologia clássica. Temos anéis otimizados para eventos de elétrons (alto esquerda), píons (alto direita) e ambos (embaixo). . . .	146
7.7	Comparação entre os métodos clássicos e os métodos neurais baseados em pré-processamento pelo sistema de anéis de células no caso elétron/píon/múon. No topo, anéis otimizados para elétrons. No meio, anéis otimizados para píons. Em baixo, ambos os anéis. . . .	147
7.8	Concordância entre o método clássico e os métodos neurais com pré-processamento topológico. Análise para o caso píon/múon (topo) e para o caso elétron/píon/múon.	149
7.9	Resultados da separação elétron/píon para a rede treinada apenas com o módulo 0. Na esquerda temos o conjunto de elétrons (concordância de 95,81% e na direita o conjunto de píons (95,94% de concordância).	152
7.10	Resultados da separação elétron/píon para a rede treinada apenas com os cinco módulos. Na esquerda temos o conjunto de elétrons (concordância de 91,14% e na direita o conjunto de píons (93,59% de concordância).	153
7.11	Resultados da separação elétron/píon para a rede treinada totalizadora. Na esquerda temos o conjunto de elétrons (concordância de 97,98% e na direita o conjunto de píons (98,50% de concordância). . .	154
7.12	Concordâncias para as três redes baseando para cada conjunto e na média. As três primeiras colunas trazem os resultados para 20 GeV e as três últimas para 100 GeV.	155
7.13	Concordâncias para extração das Componentes Principais de Discriminação com normalização pela soma e pela raiz da energia no módulo. A componente de máxima concordância para soma em cada energia é cs e para normalização pela raiz é cr.	159

7.14	Índice de identificação de eventos para cada conjunto de dados de acordo com a componente a partir da componente de máxima concordância para 100 GeV com normalização pela raiz quadrada.	160
7.15	Erros médios quadráticos após a extração da sétima componente para normalização pela soma da energia no detector e pela raiz quadrada. .	162
8.1	Tabela de índice de concordância médio nos conjuntos de teste para as três energias e duas metodologias de normalização quando 200 eventos são usados no treinamento para cada tipo de partícula.	168
8.2	Tabela de índice de concordância médio para as três energias e dois métodos de normalização quando 300 eventos são usados no treinamento para cada tipo de partícula.	170
8.3	Comparação entre velocidade de processamento em duas linguagem de programação diferentes.	171
8.4	Concordâncias entre o corte em energia e a rede neural treinada <i>online</i> para a <i>run</i> 120324 (esquerda, com 94,21% de concordância) e a <i>run</i> 120326 (direita, com concordância de 96,49%), ambas relativas a feixes de píons a 180 GeV.	181

Capítulo 1

Introdução

Durante o trabalho em engenharia, são encontrados diferentes problemas para os quais os dados fundamentais para o estudo a ser realizado possuem inconsistências. Nesta classe de problemas, incluem-se aplicações em que uma aquisição de dados de má qualidade foi realizada ou quando os dados são naturalmente inconsistentes. Também podemos pensar em situações para as quais novas classes de dados devam ser automaticamente adquiridas e identificadas.

Tipicamente, neste tipo de situação, duas abordagens podem ser realizadas. A primeira se refere a adquirir mais eventos do que os necessários para o estudo em questão, visando manter uma quantidade suficiente de dados para que sejam estatisticamente significantes. Outra abordagem seria a de adicionarmos uma metodologia de identificação *online* ao sistema de aquisição de dados para eliminar dados com inconsistência. Claramente, esta segunda abordagem deve garantir que não exista nenhuma tendência estatística na amostra deixada para os estudos *offline*.

O problema principal abordado nesta tese se refere à classificação de padrões. Conforme veremos mais adiante, queremos realizar uma separação de classes de partículas em física de altas energias. A dificuldade especial que aparece em nosso caso se resume no fato de que os conjuntos de dados aparecem naturalmente contaminados. Assim, temos eventos de contaminação adquiridos de um feixe experimental que são provenientes de diferentes classes de partículas, quando apenas uma classe de partículas foi selecionada.

Esta dificuldade leva, no problema alvo que abordamos, ao fato de não termos uma definição estatística coerente das classes que queremos distinguir, dificultando a

aplicação de qualquer método para realizar a separação. No caso, abordamos nosso problema com o uso de redes neurais artificiais, buscando utilizar sua reconhecida capacidade de extração de padrões.

Embora o discriminador neural fosse treinado utilizando-se um método que dependia do conhecimento *a priori* sobre os conjuntos de dados (o que não era, na verdade, possível de se obter dadas as contaminações existentes), este se mostrou capaz de, dando maior importância às propriedades intrínsecas dos dados, selecionar corretamente os eventos segundo as suas classes.

1.1 O ambiente de Física Experimental

O presente trabalho foi desenvolvido a partir de um experimento da física de altas energias realizado no Centro Europeu para a Pesquisa Nuclear, o CERN, localizado em Genebra, Suíça. Este centro de pesquisas, um consórcio de vários países europeus, realiza vários experimentos em física de partículas, buscando a comprovação ou descarte de teorias que expliquem a estrutura de nosso universo.

As pesquisas em física de altas energias exigem a montagem de enormes detectores de partículas, que geram sinais elétricos relacionados às medidas físicas de interesse. Os requisitos de interesse podem levar aos experimentos com colisionadores, nos quais partículas são aceleradas e feitas colidir em pontos específicos em torno dos quais os detectores são posicionados. Tipicamente, tais detectores são segmentados em módulos, produzindo dezenas de milhares (ou até centenas de milhares) de canais de leitura.

Os detectores de partículas para os colisionadores modernos são cada vez mais sofisticados, com requisitos desafiadores em termos de acuidade, faixa dinâmica e velocidade de resposta. Para atingir os objetivos de projeto, diversas gerações de protótipos são construídos e testados experimentalmente. Por fim, os módulos componentes, na fase de produção, são calibrados para então serem montados na posição definitiva.

Para testes em protótipos ou mesmo na calibração de módulos, feixes de partículas de altas energias são atirados contra os detectores. É neste ponto que o problema de contaminação de dados surge, dado o fato de que, embora os feixes

sejam gerados com todo cuidado, é impossível evitar que, no processo de geração de feixes de um tipo de partícula, por vezes não seja gerada uma partícula de classe diferente daquela que se deseja. Assim, contaminações de até 70% nos conjuntos de dados podem vir a ser encontradas.

Para se compensar a existência desta contaminação, uma grande quantidade extra de dados é adquirida, visando manter alta a estatística relativa à calibração do detector, após a retirada dos dados contaminados. Assim, uma grande quantidade de dados inúteis acaba sendo armazenada, o que leva a um alto custo de manutenção do sistema de armazenagem e a uma perda na eficiência de utilização do feixe.

Considerando que, durante os testes com feixes, a taxa de aquisição é relativamente elevada (da ordem de centenas ou mesmo milhares de eventos por segundo) e que cada evento requer a aquisição de vários canais de informação (tanto do detector em foco, como também dos detectores auxiliares que são utilizados), requerendo, portanto, uma elevada quantidade de memória, um sistema de descontaminação capaz de operar de forma *online* é, ao mesmo tempo, atraente e de difícil projeto. Além do mais, é essencial que o sistema *online* de descontaminação não introduza qualquer tendência estatística, pois toda a análise *offline* do detector seria afetada. Assim, usualmente, se aplica a descontaminação da amostra experimental por meio da análise *offline*.

Esta análise *offline* é realizada com apoio de detectores extras, além do calorímetro em calibração, e deve ser ajustada para cada faixa de energia abordada. Sendo assim, esta metodologia não pode ser automatizada, dependendo que análises anteriores à aplicação da metodologia de descontaminação sejam realizadas.

Assim são os testes com os calorímetros, detectores que medem a energia das partículas incidentes e que são provenientes das colisões. Os calorímetros são, normalmente, segmentados em dezenas de milhares de canais e, com tal granularidade, oferecem também a possibilidade de se identificar o tipo da partícula que com eles interagem. Nos modernos experimentos com colisão de partículas, os calorímetros têm um papel fundamental, fazendo medições acuradas e permitindo validar os eventos físicos de interesse.

Nossa proposta se baseia na utilização de rede neurais para a realização desta separação onde apenas a informação proveniente do calorímetro é explorada. A

análise por redes neurais apresenta a vantagem de ser facilmente automatizada, podendo ser utilizada tanto *online* quanto *offline* para realizar a identificação de contaminações.

O trabalho aqui descrito foi desenvolvido para o calorímetro Tilecal, que faz parte do experimento com o colisionador de nova geração LHC (Large Hadron Collider), que se encontra em construção no CERN e que entrará em operação em 2007.

1.2 O que foi Realizado

Inicialmente, foi implementada e testada uma metodologia neural, com treinamento supervisionado, para tratar os sinais adquiridos de um feixe contaminado (análise *offline*). Um profundo estudo de possíveis normalizações para o processamento neural foi realizado com feixes de diferentes energias, visando otimizar a eficiência de classificação de partículas e considerando a aplicação de reconhecimento de contaminação em feixes de elétrons (que sofrem contaminação por píons e múons), píons (contaminados apenas por múons) e múons (suficientemente puro para serem considerado livres de contaminação).

Métodos de compactação de sinais, baseados na física do problema, também foram pesquisados, bem como a estimação de um conjunto ótimo de parâmetros de treinamento da rede neural. Para validar a eficiência do sistema neural de análise *offline*, uma metodologia baseada nas propriedades físicas do detector e em detectores auxiliares foi estabelecida para cada caso estudado. Esta metodologia, ao contrário da aplicação neural, é extremamente dependente da energia das partículas e da posição de seu feixe, sendo normalmente utilizada pelos especialistas em calorimetria.

Como tais métodos possuem suas limitações e incertezas, utilizamos dados obtidos através de simulação na avaliação da metodologia neural. Os simuladores, embora não consigam atingir toda a faixa de características dos processos físicos, ainda assim fornecem um importante meio de validação da metodologia estudada.

Uma vez que os experimentos em física de partículas são usualmente desenvolvidos de forma colaborativa e geograficamente distribuída, a análise *offline* neural

foi tornada acessível de forma remota, via *Web*. Para tal, um sistema capaz de lidar com os dados do calorímetro e realizar, de forma integrada, as análises neurais e clássicas, foi desenvolvido, instalado e aprimorado.

Como alternativa para o experimento, foi desenvolvido um sistema capaz de operar de forma *online*, classificando eventos e gravando junto com o conjunto de dados as decisões por evento da rede neural. Uma metodologia especial para esta situação teve de ser desenvolvida, visando aproveitar as possíveis configurações do feixe relativas às diferentes contaminações para cada tipo de feixe. O sistema foi completamente integrado ao sistema de aquisição, fornecendo sua informação de classificação *online* para o mesmo e treinando o classificador com os dados vindos diretamente da eletrônica do detector.

Assim, conseguimos obter resultados, no pior caso, da ordem de 91% de identificação dos eventos do feixe e suas contaminações. No melhor caso analisado, foram atingidas eficiências de identificação superiores a 99%.

1.3 Organização do Documento

No próximo capítulo, faremos uma breve introdução à física de altas energias, incluindo detalhes que indicam a necessidade de se construir os gigantescos aceleradores de partículas e seus detectores. No segundo capítulo, abordaremos o CERN e o seu novo projeto de um acelerador de partículas, o LHC. No terceiro capítulo, de forma bem mais específica, discutiremos detalhes sobre o detector de partículas diretamente envolvido nas pesquisas realizadas, o calorímetro Tilecal. No quarto capítulo, o sistema de aquisição de dados será apresentado, o que nos levará a um melhor esclarecimento sobre a aplicação *online* do sistema neural de identificação de partículas. No quinto capítulo é que abordaremos em detalhe todas as análises *offline* realizadas, incluindo os diferentes procedimentos de normalização para o projeto da rede neural. Neste capítulo, abordaremos também uma análise de parâmetros para treinamento da rede neural e outras análises para validar a metodologia apresentada. No sexto capítulo apresentaremos o sistema remoto de discriminação de partículas que disponibiliza a aplicação para toda a colaboração de pesquisa. O sétimo capítulo discute a aplicação de métodos de pré-processamento e

compactação das informações de entrada para a rede neural e possíveis compactações do próprio classificador neural. O oitavo capítulo trará o estudo relativo a análise *online*. Determinação do conjunto mínimo de dados para um treino em tempo real, bem como os códigos desenvolvidos e os resultados obtidos na pesquisa realizada no CERN farão também parte deste capítulo. Finalmente, o nono capítulo trará as conclusões da presente tese. Em apêndice, temos os artigos publicados durante o desenvolvimento da presente tese e os códigos em Fortran (*offline*) e em C++ (*online*) usados na implementação da metodologia neural de análise.

Capítulo 2

Ambiente da Física de Altas Energias

Neste capítulo, descreveremos algumas idéias básicas relativas aos experimentos da física de partícula que levam à necessidade de se construir enormes aceleradores e seus detectores. A partir destas questões, vamos descrever o CERN aonde alguns dos mais importantes experimentos na área foram realizados, comprovando algumas teorias da física moderna. Finalmente, vamos descrever o próximo experimento do CERN, o LHC.

2.1 A Física de Partículas

Durante a primeira e início da segunda década do século XX, havia no meio científico uma grande discussão sobre o modelo de átomo que mais se aproximaria da realidade. Experimentos realizados nos anos precedentes tinham esclarecido que o átomo era uma unidade neutra (sem carga elétrica aparente) de massa que, em algumas situações específicas, poderia mostrar uma composição baseada num equilíbrio de corpos com carga positiva e negativa, vindo daí sua neutralidade. Vários modelos foram propostos na época, sendo um dos mais conhecidos deste período o modelo de Thomson.

Segundo este modelo, o átomo seria uma esfera de carga positiva distribuída, de um diâmetro conhecido e com cargas pontuais negativas coladas a suas paredes por atração mútua. Esta descrição estava de acordo com vários experimentos, até

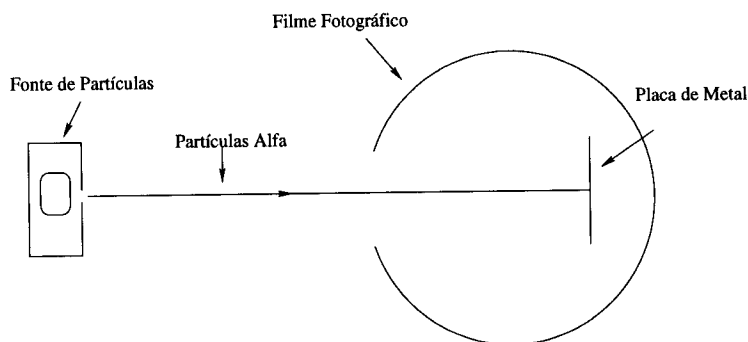


Figura 2.1: Experimento de Rutherford sobre o modelo atômico.

que Ernest Rutherford realizou um cuidadoso experimento, que devemos descrever por possuir vários elementos que podem ser encontrados na pesquisa de nossos dias e que contestou o modelo de Thomson.

Rutherford estava analisando o ângulo de espalhamento de partículas alfa (de carga positiva) ao ultrapassarem uma finíssima folha de ouro. Um esquema de tal experimento pode ser visto na Figura 2.1. As partículas eram geradas pelo decaimento de átomos de um material radioativo e as trajetórias das partículas, após passarem pela folha de ouro, eram registradas no filme fotográfico. Pelos cálculos possíveis de se efetuar na época, sabia-se que a largura de tal folha não deveria conter mais do que algumas dezenas de átomos do metal. Segundo o modelo atômico de Thomson, pela distribuição da carga positiva por toda a esfera atômica, seriam esperados pequenos desvios na trajetória da partícula ao atravessarem a estrutura do átomo. Entretanto, este não foi o resultado obtido, tendo sido encontrados, em alguns casos, espalhamentos de praticamente 180° , ou seja, as partículas praticamente voltavam para sua fonte.

Este comportamento indicou a Rutherford que os cálculos deveriam ser refeitos de alguma forma [4]. Sua proposta, que resultou num novo modelo atômico, foi que a distribuição de cargas positivas deveria ser muito mais concentrada do que o suposto por Thomson. Essa conclusão veio do fato que apenas uma carga muito concentrada faria as partículas se espalharem com ângulos tão grandes. Segundo Rutherford, seguindo os resultados da experiência acima descrita, o diâmetro da região aonde estariam presentes as cargas positivas seria, pelo menos, quatro ordens de grandeza menor do que o tamanho completo do átomo.

Eletrofraca Unificada			Forte Cor			Leptons			Quarks		
Nome	Mass GeV/c ²	Carga Eletrica	Nome	Mass GeV/c ²	Carga Eletrica	Tipo	Mass GeV/c ²	Carga Eletrica	Tipo	Mass GeV/c ²	Carga Eletrica
γ photon	0	0	g gluon	0	0	ν_e eletron neutrino	$<1 \times 10^{-6}$	0	u up	0.003	2/3
W ⁻	80.4	-1				e eletron	0.000511	-1	d down	0.006	-1/3
W ⁺	80.4	+1				ν_μ muon neutrino	<0.0002	0	c charme	1.3	2/3
Z ⁰	91.187	0				μ muon	0.106	-1	s estranho	0.1	-1/3
						ν_τ tau neutrino	<0.02	0	t top	175	2/3
						τ tau	1.7771	-1	b bottom	4.3	-1/3

Figura 2.2: Partículas do Modelo Padrão.

Seguindo este raciocínio, chegou-se a um modelo mais completo, em que existe um núcleo com cargas positivas e uma esfera de elétrons em eterno movimento a sua volta. O diâmetro desta esfera eletrônica é o diâmetro total do átomo. O núcleo concentraria não só as cargas positivas como também toda a massa do átomo. Como o número de cargas negativas (chamadas elétrons) é igual ao de cargas positivas (chamadas de prótons) no núcleo do átomo, este, no total, é eletricamente neutro.

O modelo de Rutherford para o átomo apresentava, entretanto, algumas inconsistências, como, por exemplo, o fato de que um elétron rodando em torno do núcleo atômico deveria perder energia (partículas carregadas perdem energia em trajetórias curvas emitindo fótons de luz). Niels Bohr resolveu parcialmente esse problema utilizando a hipótese quântica que, aprimorada por Edwin Schrödinger, levou a física para o domínio da mecânica quântica. Mais recentemente, pesquisas descobriram que existem partículas mais fundamentais dentro dos prótons, os quarks. Toda essa pesquisa levou ao estabelecimento de um pequeno número de partículas que constituem a matéria e mais algumas que são responsáveis pelas forças da natureza. Estas descobertas foram consolidadas no chamado Modelo Padrão [5].

Na Figura 2.2 vemos as partículas que, segundo o Modelo Padrão, constituem toda a matéria do Universo. Muitas destas partículas já tiveram sua existência comprovada, através de variadas experiências. A partícula responsável pela transmissão da força fraca, por exemplo, foi descoberta no CERN em 1983, dando aos cientistas responsáveis pelos experimentos, o professor italiano Carlo Rubbia e o engenheiro físico holandês Simon van der Meer, o prêmio Nobel de Física de 1984 [6]. A partir desta comprovação, foi possível unificar as teorias que explicam a existência da força

fraca e da força eletromagnética.

Atualmente, um dos pontos importantes ainda não confirmados do Modelo Padrão é a existência de uma partícula que explique a existência de massa. Tal partícula foi prevista pelo cientista inglês Peter Higgs em 1964 e por isso se chama o Bóson de Higgs [7]. Este ponto, atualmente, suscita grande interesse por uma resposta experimental.

Voltando ao experimento de Rutherford, podemos identificar neste vários elementos que fazem parte dos atuais experimentos. Claramente, os equipamentos atualmente usados são muito mais complexos, utilizando a mais alta tecnologia disponível, mas as idéias básicas continuam sendo as mesmas. Por exemplo, a fonte de partículas alfa usada no experimento de Rutherford consistia de uma pequena quantidade de um material radioativo, como o Rádio ou o Urânio. Estas fontes de partículas são fontes naturais. Para os experimentos atuais, elas não produzem partículas com os altos níveis de energia exigidos, nem na quantidade necessária. Para tais experimentos são, então, montados grandes aceleradores de partículas que aceleram as mesmas através de intensos campos elétricos. Desta forma são produzidas enormes quantidades de partículas com um alto nível de energia. Mais ainda, em muitos experimentos, em vez de se utilizar um alvo fixo (a folha de metal), são utilizados dois feixes de partículas contrários, aumentando a energia de colisão entre as mesmas.

Da mesma forma que Rutherford precisou de um filme fotográfico para registrar as partículas que espalhavam da colisão com o núcleo, são necessários detectores para as partículas que são aceleradas nos atuais aceleradores de partículas. Dada a mais larga faixa de energia atualmente utilizada, os detectores são muito mais complexos, especializados em detectar alguns tipos de partículas, medindo algumas características das mesmas, como energia (momento) e trajetória. Também, como muitos dos fenômenos pesquisados só ocorrem com baixíssimas probabilidades, devem ser gerados eventos de colisão de partículas a uma altíssima taxa por unidade de tempo. Isso exige detectores rápidos, capazes de se recuperar de uma detecção em pouco tempo, ficando prontos para a próxima detecção. Também é necessário que sejam definidos esquemas automáticos (através de eletrônica altamente especializada) de rejeição de eventos que não trazem informação nova, otimizando a chance

de se encontrar uma nova física.

2.2 O CERN

Cerca de quatro décadas depois das descobertas de Rutherford, no início dos anos 50, foi criado o inicialmente chamado *Conseil Européen pour la Recherche Nucléaire* (Conselho Europeu para a Pesquisa Nuclear), o CERN, com a intenção de manter muitos dos cientistas que haviam abandonado a Europa por causa da Segunda Guerra em solo europeu. Hoje em dia, o CERN [8] é um instituto de pesquisa na fronteira franco-suíça, cujos países membros europeus contribuem com parte de seu produto interno. Além disso, cerca de uma centena de países colaboradores de todo o mundo participam dos vários experimentos de pesquisa em física de partículas realizados no CERN.

Algumas exigências destes experimentos levam ao desenvolvimento de técnicas que encontram aplicação em outras áreas, algumas vezes bastante distintas da física de partículas. O exemplo mais conhecido é a *World-Wide Web*, o protocolo de comunicação da Internet. Este sistema foi criado no começo dos anos 90, no CERN, para facilitar a comunicação entre seus diversos pesquisadores no mundo inteiro [9] e também a distribuição das informações.

O último grande experimento montado pelo CERN foi o *Large Electron-Positron Collider* (Grande Colisionador Elétron Pósitron), o LEP. Na Figura 2.3 temos uma vista aérea do LEP.

O LEP [1], que começou a operar em 1989, é a maior máquina científica já construída. O LEP era feito de um acelerador de partículas circular de 27Km de extensão sob solo francês e suíço nas proximidades da cidade de Genebra. Tal acelerador, através de intensos campos elétricos, impulsionava grupos de partículas (elétrons e sua anti-partícula, os pósitrons) em direções contrárias até quatro pontos de colisão. Nestes pontos, quatro conjuntos de detectores especializados se encontravam dispostos. Eram eles : Aleph, Opal, L3 e Delphi [10]. Foram utilizados diferentes detectores, de forma a se otimizar a análise dos dados produzidos a partir dos diferentes enfoques de detecção.

Durante o último ano de operação do LEP, em 2000, suas cavidades de ace-



Figura 2.3: Vista aérea do anel LEP sob várias cidades na fronteira franco-suíça. Extraído de [1].

leração foram modificadas, visto que se esperava que o Bóson de Higgs tivesse uma massa entre 108 e 190 GeV (Giga-elétron-Volts), sendo esta ligeiramente maior do que a atingida até então pelo LEP [11]. No final do ano de 2000, o LEP varreu uma vasta região de energia candidata a produzir o Bóson de Higgs. Embora os quatro detectores tivessem identificados alguns eventos candidatos, verificou-se que não se levantaria uma quantidade de eventos necessária para provar a existência da partícula com suficiente significância estatística [12, 13]. Assim, o acelerador foi desligado e inicio-se o seu processo de desmontagem. Isso teve de ser feito posto que, no próprio túnel do LEP, passava a ser construído um novo acelerador de partículas, o LHC.

2.3 O LHC

O LHC, *Large Hadron Collider* (Grande Colisionador de Hadrons) [14], será montado no mesmo túnel do LEP. Entretanto, graças a utilização de poderosos ímãs supercondutores para melhor manter a trajetória das partículas [15], no caso prótons, espera-se que o LHC atinja até 14 TeV (Tera-elétron-Volts) de energia no centro de massa de suas colisões. Isso seria mais de 70 vezes a energia atingida pelo LEP. Claro que isso implica num gasto adicional de energia e tecnologia, posto

que os ímãs devem ser resfriados a temperaturas próximas do zero absoluto (cerca de 18° acima do zero absoluto). Entretanto, o mesmo campo magnético manterá a trajetória de ambos os feixes de prótons que, vindo em sentidos contrários, colidirão em dois pontos específicos.

O LHC deve produzir colisões de pacotes de partículas com grande luminosidade (cerca de $10^{34} \text{cm}^{-2} \text{s}^{-1}$), iniciando suas operações em 2006. Para tentar estabelecer o nível estatístico necessário para provar a existência de partículas com tão pequena probabilidade de vir a ser observada, o LHC produzirá colisões com intervalos de apenas 25 nano-segundos, realizando 40 milhões de colisões por segundo. Assim, os pacotes de partículas produzidos pelo pré-acelerador SPS do CERN serão injetados no LHC guardando uma distância mútua de 7,5 m entre si. Para preencher o LHC serão necessários cerca de 3600 destes pacotes de partículas.

Como são muitos pacotes de dados sendo trabalhados por segundo, estes serão identificados por um número de até 12 bits que é o *Bunch Cross ID* (Identificação do cruzamento de pacote). Todos os detectores (e seus sub-detectores) têm de produzir, de forma síncrona, dados para cada colisão. Isso resulta na necessidade de um sistema de sincronia para todos os sistemas. Além disso, a eletrônica de alguns detectores precisa de um sinal de controle que abra uma janela de aquisição, exatamente quando, efetivamente, houver um possível evento a ser adquirido. Considerando estes requisitos, o LHC terá um sistema de coordenação chamado *Timing, Trigger and Control* (TTC - Temporização, Gatilhamento e Controle) [16]. Este sistema distribui os *Bunch Cross IDs* para todos os detectores e permite que os sub-detectores utilizem um relógio centralizado, em sincronia com a produção de pacotes de partículas do LHC, para enviar comandos para sua eletrônica rápida. Assim, como exemplificaremos mais adiante, é possível criar uma partição TTC, na qual todos os sistemas podem trocar entre si comandos e informações em perfeita sincronia, além da correção de efeitos, como tempo de voo das partículas no detector, bem como a compensação dos tempos de aquisição da própria eletrônica. Estas partições, sempre mantendo a sincronia com o LHC, podem ser completamente independentes entre si, possibilitando que apenas comandos relacionados com um único subdetector sejam trocados dentro da partição.

O LHC irá possuir dois grandes detectores para a colisão de feixes, o ATLAS,

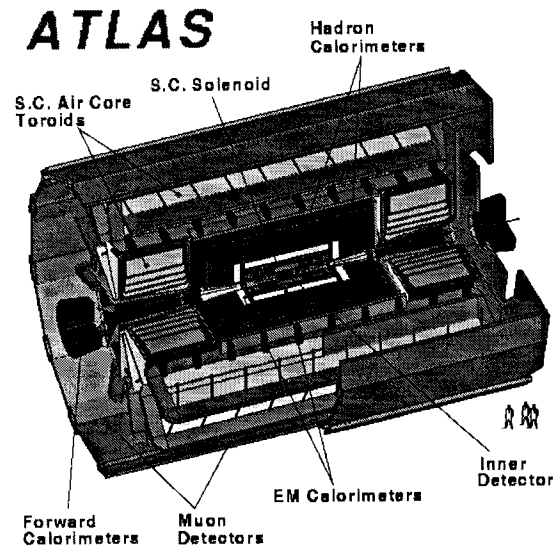


Figura 2.4: Desenho esquemático do detector ATLAS. Extraído de [2].

A *Toroidal Lhc AparatuS* (Um Aparato Toroidal do LHC) [2, 17] e o CMS, *Compact Muon Solenoid* (Solenóide Compacto para Múons) [18] e também dois detectores de alvo simples, o LHCb e o ALICE. Os detectores de colisão procurarão, inicialmente, o Bóson de Higgs, cobrindo a faixa deixada pelo LEP, com uma maior probabilidade de possuir eventos provindos de uma nova física (dada a sua, imensamente maior, luminosidade). Nosso trabalho se concentra no detector ATLAS, por isso iremos detalhar tal detector.

O ATLAS será um detector projetado de acordo com uma linha tradicional na física de partículas, empregando, entretanto, as mais sofisticadas tecnologias para implementação deste projeto. Na Figura 2.4 temos um esquema do detector. Nesta figura, vemos em detalhe os vários sub-detectores que compõem o ATLAS. Os dois feixes de partículas do LHC virão na direção eixo do toróide formado pelo detector, vindo a colidir na parte mais central do mesmo. Espera-se que os sub-produtos daí advindos se espalhem de dentro pra fora no detector, sensibilizando seus sub-detectores.

Assim, de dentro pra fora, temos, inicialmente, o *Inner Detector* (Detector Interno), responsável por, através de sistemas de pequenos sensores, detectar os traços que formam as trajetórias das partículas. Na seqüência, as partículas passarão pelo *S.C. Solenoid*, que é um grande solenóide, capaz de produzir um intenso campo magnético através do uso de supercondutividade. Este solenóide leva as partículas

com carga a percorrerem trajetórias curvas, permitindo uma melhor medida de seu momento. Na seqüência, as partículas atingem os calorímetros do detector, iniciando pelo *E.M. Calorimeter*, o calorímetro eletromagnético, que se constitui num medidor da energia das partículas eletromagnéticas (elétrons, pósitrons, entre outros). Estas partículas são detectadas através de absorção total da sua energia, o que significa que as partículas que tiverem sua energia medida não passarão deste sub-detector, sendo completamente absorvidas pela massa do mesmo.

O segundo calorímetro, do qual trataremos mais extensivamente neste trabalho, é o *Hadron Calorimeter* (Calorímetro Hadrônico), que é responsável por absorver e medir a energia de partículas hadrônicas, tais como os píons, prótons e nêutrons. Este detector será completamente descrito no próximo capítulo.

Em seguida, as partículas que tiverem passado pelos dois calorímetros terão sua trajetória desviada por outro solenóide, o *S.C. Air Core Toroids*. Estas partículas poderão ser, principalmente, múons (positivos e negativos) e neutrinos (sem carga). Os primeiros terão suas rotas alteradas por este solenóide. Para detectar os múons (os neutrinos não são facilmente detectáveis), os *Muon Detectors* funcionam como paredes com pequenas câmeras de fios para detectar, através de interação eletromagnética, a passagem de tais partículas. Finalmente, os *Forward Calorimeters* (Calorímetros dianteiros) servem para estudar partículas que saiam a ângulos próximos de zero da colisão dos feixes de partículas.

O LHC vai colidir partículas com uma taxa muito elevada (40 milhões de eventos por segundo). Entretanto, o fenômeno que se deseja detectar é extremamente raro, de tal maneira que a maior parte dos eventos gerados provém de física já conhecida e sem interesse algum para as pesquisas. Deve existir, então, um sistema de rejeição para estes eventos que formam o ruído do LHC. Além disso, cada evento no ATLAS deve significar em torno de 2 MB de dados. Supondo que fosse possível adquirir todos os eventos produzidos, seriam necessários, por segundo de operação, mais de 80 TB de espaço de armazenagem, o que, rapidamente, saturaria a rede de computadores do CERN.

Assim, o sistema de *Trigger* (validação) do ATLAS opera de forma *online* e tem três níveis de rejeição [19]. O primeiro nível é baseado em fatores bastantes diretos como, por exemplo, a existência ou não de depósitos de energia de partículas

em certas partes do detector. Por exemplo, no calorímetro hadrônico, parte do sinal de Trigger será produzido pela existência de sinal de energia acima de uma certa intensidade (para separarmos sinal do ruído) em certas direções deste calorímetro. Estas direções, que cortam o ponto de impacto são as *Trigger Towers* (Torres de Trigger) [20]. Estes cálculos são realizados com uma eletrônica bastante especializada e rápida. Os eventos já adquiridos pela eletrônica diretamente associada aos sub-detectores são mantidos em uma memória especial. Dessa forma, apenas os eventos que receberem uma aceitação do primeiro nível de Trigger são adquiridos pelo resto do sistema, baixando a taxa de aquisição de 40 MHz para, no máximo, 100 KHz [21].

O segundo nível de Trigger é realizado utilizando apenas a fração da informação vinda dos eventos aonde a presença de partículas foi identificada pelo nível 1. Já em meio digital, estas regiões são utilizadas para processamentos rápidos em meios baseados em PCs [22]. Como apenas uma pequena parte dos dados dos eventos (cerca de 4% dos canais de informação do ATLAS) deve ser utilizada nestes cálculos e os mesmos podem ser processados em paralelo, espera-se que se possa realizar estes procedimentos e tomar a decisão de se aceitar ou não o evento, de tal modo que o terceiro nível terá uma frequência de entrada em torno 1 KHz. Este terceiro nível realizará, em grupos de computadores, o processamento necessário para reduzir um pouco mais a taxa de aquisição (agora considerando a informação completa do evento). Espera-se atingir a taxa final de 100 eventos por segundo para armazenagem, o que reduzirá a quantidade de dados a se armazenar para 200 MB por segundo, uma quantidade mais realística de se manejar *offline*.

Todo este complexo sistema, sincronizado pelo sistema de TTC com o relógio de colisões do LHC, tem de ser controlado por um sistema geral, de forma a que se possa ativar todos os subsistemas de aquisição em conjunto. Para tal, vem sendo desenvolvido um sistema de controle e monitoração de todo o *Data Acquisition* (Aquisição de Dados), o DAQ [23]. Este sistema, em sua versão protótipo "-1", está buscando uma avaliação de tecnologias através da implementação de uma completa fatia vertical do sistema de DAQ, tal qual se espera que venha a ser no ATLAS. Esta implementação foi realizada, inicialmente, com o grupo do calorímetro hadrônico e agora está sendo expandida para demais sub-detectores do ATLAS [24].

No próximo capítulo iremos detalhar o sub-detector do ATLAS que concentra as principais informações deste trabalho, assim como o seu sistema de aquisição de dados.

Capítulo 3

O Detector TileCal e a Contaminação do Feixe de Partículas

Neste capítulo detalharemos o calorímetro hadrônico do ATLAS. Explicaremos inicialmente sua estrutura, o processo de deposição de energia e o equipamento utilizado para aquisição das informações acoplado diretamente nos módulos do calorímetro. Na segunda parte deste capítulo, iremos discutir os diferentes tipos de partículas utilizados nos testes e suas interações com o calorímetro. Esta parte será a base necessária para melhor entendermos os métodos utilizados classicamente para identificação de partículas, os quais usaremos no próximo capítulo para validar o nosso procedimento neural.

3.1 O Detector

O calorímetro hadrônico do ATLAS é um detector especializado na medida de energia depositada por partículas hadrônicas. Este detector é constituído por dois tipos de material, que são dispostos em camadas, formando a estrutura do detector. Na Figura 3.1, vemos essa disposição de materiais. A massa do detector é constituída de placas de ferro, material passivo, capaz de absorver a energia das partículas incidentes. O ferro, com seus átomos bastante estáveis, ao receber as partículas, tem uma baixa probabilidade de sofrer um decaimento em algum outro

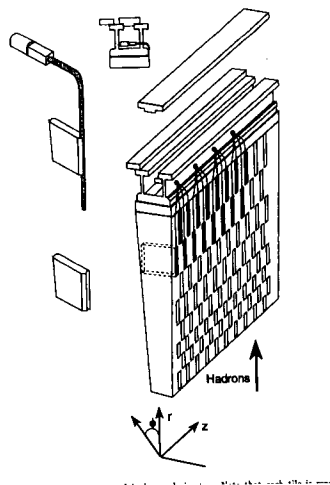


Figura 3.1: Protótipo do calorímetro hadrônico do ATLAS. Extraído de [3].

tipo de núcleo. O outro material constituinte do calorímetro é um material cintilante. Quando da interação de alguma partícula com o calorímetro, a energia da interação excita os átomos deste material cintilante, que produz uma emissão de luz, em quantidade proporcional à energia captada. No desenho esquemático, as plaquetas em destaque são as unidades (telhas) cintilantes. Tais unidades têm um acoplamento direto com fibras ópticas dispostas sobre a superfície do calorímetro. A luz emitida é recolhida pelas fibras ópticas e levada para uma reentrância, na parte superior do módulo, aonde pode ser disposto um sistema de fotomultiplicadores para leitura eletrônica do sinal.

Dada a disposição do material como telhas sobre um telhado, o calorímetro hadrônico do ATLAS também é chamado de *Tile Calorimeter* (Calorímetro de Telhas), ou TileCal, que é como passaremos agora a chamá-lo.

3.1.1 Topologia do Calorímetro e Arranjo Experimental

As fibras ópticas são agrupadas segundo a região do calorímetro a que atendem, fazendo com que sejam formadas células na superfície dos módulos que compõem o calorímetro. No módulo protótipo da Figura 3.1, haviam 20 células, as quais eram lidas por meio de fibras ópticas conectadas a cada um dos dois lados livres de cada telha cintilante, que transportavam a luz gerada até os tubos fotomultiplicadores. No total, 40 canais de leitura eram produzidos por módulo. Em um dos arranjos experimentais montados para o teste do detector, cinco módulos foram agrupados,

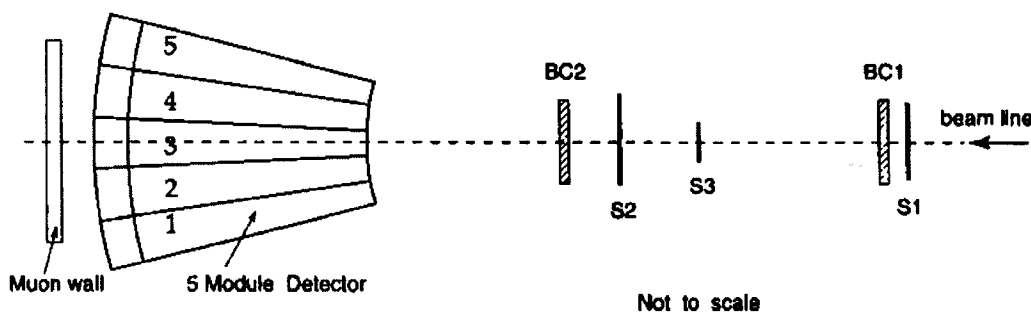


Figura 3.2: Esquema experimental para teste com o protótipo de cinco módulos. Extraído de [3].

sendo então gerados 200 canais de leitura.

Na Figura 3.2, temos o arranjo experimental destes cinco módulos, dispostos uns sobre os outros, sobre uma mesa giratória. A linha de feixe do acelerador SPS (usado durante este tipo de teste) é fixa, devendo o calorímetro girar para que o feixe possa atingi-lo em variados ângulos de incidência.

É importante notar que o feixe do SPS não permanece constantemente ativo. O feixe fica ativo durante 2,4 segundos, emitindo uma rajada de eventos e, logo após, fica inativo por 11,6 segundos. Em cada rajada podem vir cerca de 200 eventos (num feixe de múons) ou cerca de 2000 a 3000 eventos (num feixe de elétrons ou píons). Claramente, isso impõe restrições de tempo ao sistema de aquisição e também às aplicações *online* que serão estudadas mais adiante.

Conforme se pode notar na figura, duas câmaras de fios (*Beam Chambers* marcadas como BC1 e BC2) definem a posição do feixe, sendo possível, em uma análise posterior, se descartar eventos em que o feixe possa ter incidido com um ângulo muito grande em relação a sua própria linha de referência. Os contadores cintilantes (*Scintillating Counters* marcados como S1, S2 e S3 na figura) servem para identificar a presença de uma partícula vinda do feixe, atuando como um detector de coincidência. Assim, quando os três cintiladores dão o sinal da passagem de uma partícula, o sistema de aquisição do detector pode ser disparado. Finalmente, vemos, na figura, a Parede de Múons (*Muon Wall*), que fica atrás do calorímetro para ajudar na identificação de múons. Estas partículas não são absorvidas pelo calorímetro, e passam pela Parede de Múons, deixando um sinal bem pontual. Píons, quando não completamente absorvidos pelo detector, também sensibilizam a Parede de Múons,

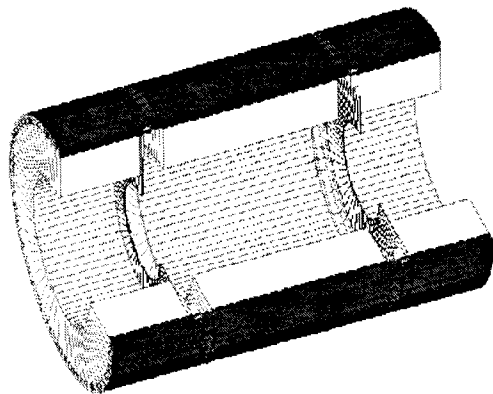


Figura 3.3: Módulos do TileCal a volta do ponto de colisão.

mas deixam um sinal que se espalha por uma região bem maior do que o ponto deixado pelos múons.

Os sinais provindos deste calorímetros foram tratados como mostra [25], quando foi aplicado pela primeira vez o processamento neural para discriminação de partículas no detector. Foi neste primeiro trabalho que surgiram interessantes resultados com a aplicação neural. Os demais trabalhos se referem ao protótipo cabeça de série que foi produzido em 1996, tendo sido testado com feixes de partículas a partir de então. Como a maior parte do trabalho se realiza, na verdade, sobre este último protótipo, vamos detalhá-lo a seguir.

O calorímetro hadrônico do ATLAS, assim como todas as partes do detector, circundará o ponto de colisão, de modo a detectar todos os produtos hadrônicos que uma dada colisão venha, por ventura, produzir. Na Figura 3.3, temos o calorímetro hadrônico completo do ATLAS em detalhe, aonde aparece sua divisão em 64 módulos côncavos cobrindo, cada um, um sessenta e quatro-avo da curva completa.

Conforme pode se ver também na figura, notamos que além da parte central chamada de *Barrel*, o barril, temos duas continuações laterais que visam estudar partículas que tenham trajetórias com ângulos mais distantes da direção ortogonal com relação à linha de colisão no LHC. Estas partes receberam o nome de *Extended Barrel*, extensão do barril.

A disposição de células no barril e da extensão do barril é diferente. Podemos ver estas distribuições na Figura 3.4. Nesta figura, vemos meio módulo do barril (à esquerda na figura). Este módulo possui 10 células marcadas de A1 até A10 na camada de célula inferior (a qual chamaremos de primeira), 9 células na segunda

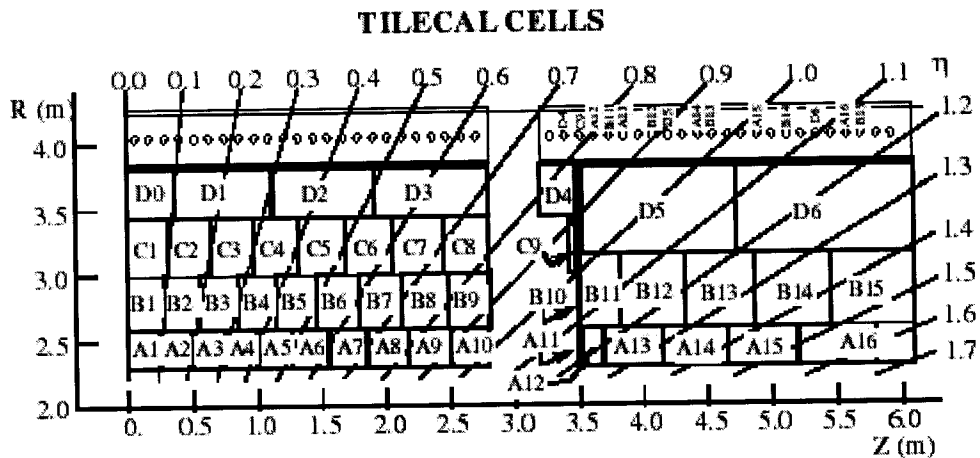


Figura 3.4: Distribuição de células para o barril (esquerda) e a extensão do barril.

camada marcadas de BC1 até BC8 e também B9 (as células marcadas com B e C são lidas juntas), e finalmente, a terceira camada com apenas 4 células marcadas de D0 até D4. No total temos 23 células para esta metade do barril. Como estas células são lidas pelos dois lados da telha, temos no total 46 canais de leituras.

Temos também direções que são marcadas na figura como ângulos. Estas direções marcadas são chamadas de pseudo-rapidez (η que, no caso, varia de 0.0 a 1.7). Nas direções intermediárias (0.05, 0.15, 0.25, etc) temos grupos de células que chamaremos de torres. Estes grupamentos formam direções de detecção ótima para as partículas que chegam.

A extensão do barril tem um número menor de células, com uma menor granularidade. Assim, temos 6 células na primeira camada, marcadas de A11 até A16, 7 na segunda camada, marcadas como C9 mais B10 até B15, e, por fim, na terceira camada, temos mais 3 células, marcadas de D4 até D6. No total, cada módulo da extensão do barril possui 16 células a serem lidas, que resultam em 32 leituras. No total, considerando todas as células do barril e da extensão do barril, temos um total de cerca de 10000 canais de leitura, somente para o calorímetro hadrônico no ATLAS.

3.1.2 Leitura do Sinal Luminoso das Fibras

O sinal luminoso, produzido pelas telhas do calorímetro e conduzido para fora pelas fibras óticas, é injetado em células fotomultiplicadoras que o convertem em corrente elétrica. Na Figura 3.5 temos, na figura da esquerda, uma célula fo-

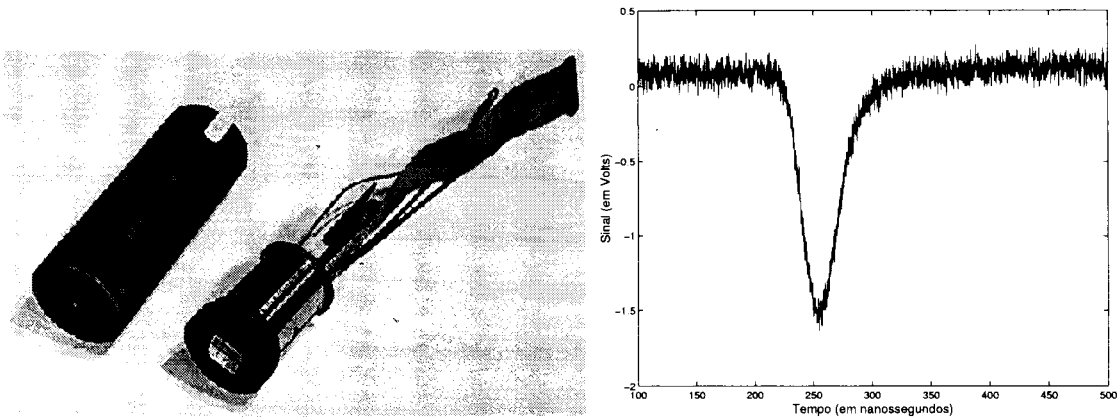


Figura 3.5: Célula fotomultiplicadora usada no TileCal (esquerda). Sinal produzido por esta mesma célula (direita) e já conformado pelo conformador de sinais.

tomultiplicadora utilizada no experimento. Na sua parte frontal temos o acoplador ótico da célula, e, por traz, a parte eletrônica. Tais células têm um ganho na relação fóton para elétron que depende de vários fatores, como por exemplo, a alta tensão utilizada pela célula para converter os fótons em elétrons e a temperatura ambiente, entre outros.

Na figura a direita da Figura 3.5 temos um sinal produzido por esta célula fotomultiplicadora após passar por um conformador de sinais como veremos mais adiante. O sinal é negativo e sua integral é proporcional a energia depositada na célula fotomultiplicadora. A linha de base, que representa o ruído quando não temos sinal, é chamada de sinal de pedestal. Este sinal deve ser sempre descontado para o cálculo da energia. No caso, este sinal foi adquirido amostrando-se a uma taxa bem mais alta do que a que normalmente se adquire (cerca de 5 GHz) a saída do sistema de soma analógica, quando apenas uma das células do calorímetro estava habilitada a produzir sinais.

O sinal de pedestal pode também variar, dependendo da temperatura da gaveta que carrega todo o equipamento eletrônico, e que se encontra encaixada num orifício localizado na parte posterior do detector [26]. Na Figura 3.6, temos a gaveta do TileCal. Na sua lateral, temos os acopladores óticos das células fotomultiplicadoras, que vão se posicionar, quando a gaveta é inserida no detector, exatamente na frente dos terminais das fibras óticas. Na parte inferior (não vista na figura), temos a parte que fornece a alta tensão para as fotomultiplicadoras.

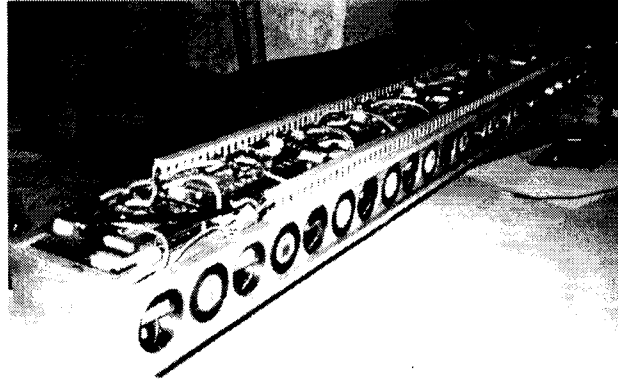


Figura 3.6: Foto da gaveta com equipamentos eletrônicos do TileCal.

Na parte superior da gaveta, temos varias placas de circuitos sobrepostas uma sobre as outras. A placa mais abaixo, é a placa mãe [27]. Esta placa distribui as baixas tensões necessárias a todos os sistemas, distribui os sinais de TTC, que são usados para enviar comandos aos digitalizadores e placas 3-em-1. As placas 3-em-1 (não visíveis na figura) são responsáveis por receber o sinal das fotomultiplicadoras e conformar seu formato para uma melhor leitura do restante do sistema [28]. Estas placas também podem ser programadas (através do sistema de TTC) para simular sinais. As placas digitalizadoras (segundo e terceiro nível de placas na figura) recebem o sinal de *trigger* via o sistema de TTC, que inicia uma janela de aquisição, e digitalizam o sinal conformado pelas placas 3-em-1 [29]. Uma placa de interface transmite o sinal para o restante do sistema de aquisição (que se encontra fora da gaveta, numa sala reservada) por meio transmissores óticos. São adquiridas, de cada célula, nove amostras do sinal de energia. A primeira (ou a última) ou as duas primeiras (ou as duas últimas) podem ser usadas para se estimar o valor do pedestal e descontar das amostras que estejam na janela de sinal (são usadas 5 amostras de sinal) para o cálculo da energia.

Além destas placas, temos, na gaveta, os somadores analógicos [30]. Estes componentes fornecem sinais analógicos rápidos para a soma dos sinais provenientes de todas as células que formam uma dada *trigger tower*. Este sinal, indo para o primeiro nível de *trigger*, faz parte, juntamente com um sinal vindo do Detector de Múons, da decisão pela aceitação ou não de um evento, antes que o mesmo tenha sido já adquirido pelos digitalizadores.

Toda esta eletrônica produz uma grande quantidade de calor que, no ambi-

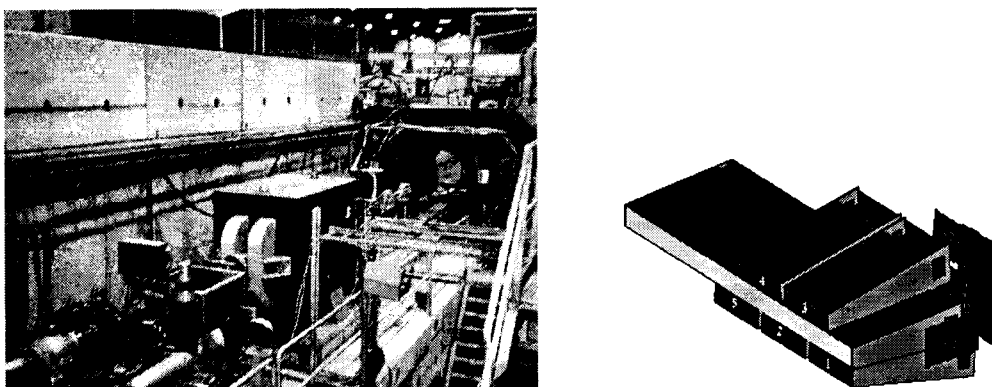


Figura 3.7: Linha do feixe na área H8 aonde são testados os protótipos do TileCal (esquerda) e arranjo experimental utilizado de 1996 a 1999.

ente fechado da gaveta, pode chegar a níveis críticos. Para se reduzir este problema, um pequeno duto, com água constantemente resfriada, é levado por caminhos curvilíneos a todo interior da gaveta. O pequeno duto preto que aparece na parte frontal superior da gaveta faz parte deste sistema de resfriamento. Um sistema com nitrogênio líquido [31] mantém a temperatura da água constante. Em caso de necessidade, basta variar o fluxo de água para se variar a temperatura no interior da gaveta.

Todo este conhecimento foi acumulado no período diretamente anterior ao início dos testes de 2001, quando foi possível se montar uma pequena bancada com uma placa mãe, junto com diversas placas 3-em-1, somadores e uma placa digitalizadora. Este trabalho serviu para se acumular algum conhecimento sobre a programação, via TTC, que estas placas aceitam.

De 1996 até 1999 (excetuando-se em 1997), foi utilizado um módulo cabeça de série em todos os testes. Este módulo ficou conhecido como Módulo 0. Estes testes serviram para se desenvolver o sistema eletrônico descrito acima. O arranjo experimental utilizado é similar àquele usado para o protótipo de 5 módulos, que incluía cintiladores e câmaras de fios e pode ser visto na parte esquerda da Figura 3.7. Esta figura traz a linha de feixe H8, produzida pelo acelerador SPS, quando da sua chegada na área de testes. Desde 2001, os testes com feixes passaram a ser realizados no intuito de se calibrar os módulos. 25% dos módulos do TileCal serão calibrados antes do início do ATLAS, em 2006. O processo de calibração visa encontrar constantes que convertam os sinais produzidos pela eletrônica do detector em constantes

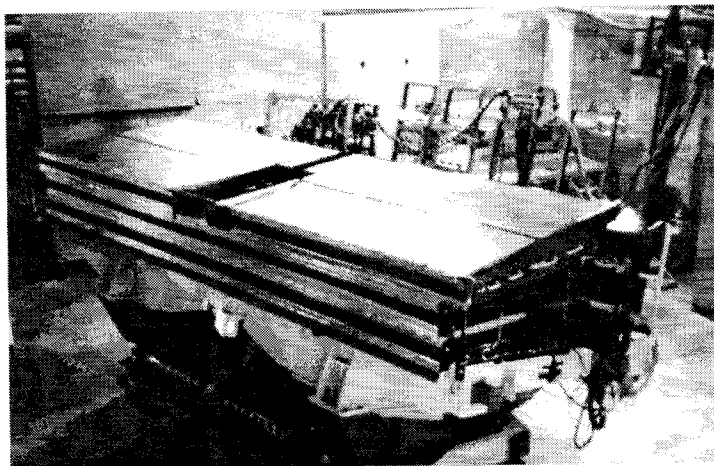


Figura 3.8: Módulos do TileCal sendo testados na área H8. Vê-se dois módulos da extensão do barril sobre dois módulos do barril (o inferior é o Módulo 0).

físicas (no caso, valores de energia).

Como o Módulo 0 era insuficiente para conter completamente a cascata de partículas gerada quando da interação de um píon, por exemplo, foi utilizado, até 1999, o esquema que aparece na parte direita da Figura 3.7. Neste esquema, vemos que os cinco módulos do protótipo anterior foram dispostos a volta do Módulo 0 (na metade do módulo que estava instrumentalizada). Os detectores de múons (Muon Walls), também estavam presentes, para facilitar a identificação de múons de contaminação presentes nos feixes de elétrons e píons (ver próxima seção).

Atualmente, durante a fase de calibração, são testados três módulos ao mesmo tempo, um módulo do barril e dois da extensão do barril (estes últimos têm aproximadamente metade da dimensão longitudinal do barril). A Figura 3.8 mostra os módulos sobre a mesa de testes (giratória e comandada através de *software*). Neste caso, os cinco módulos da geração anterior de protótipos não estão sendo utilizados. No total temos 3 módulos em calibração, dois da Extensão do Barril na parte superior, um do Barril, e, finalmente, na parte inferior o Módulo 0. Este último não está sendo calibrado, servindo apenas como referência ao processo de calibração. No canto direito, vemos em cada módulo um ressalto da onde partem vários cabos elétricos e óticos. É neste ponto que são instaladas as gavetas. Atrás dos detectores, vemos as fontes de alimentação dos mesmos e uma peça destinada a acomodar os cabos que devem ir para a sala de controle.

3.2 Deposição de Energia

Durante as diferentes fases de teste com protótipos e calibração dos módulos de produção, o calorímetro TileCal foi submetido a feixes de três tipos de partículas : elétrons, píons e múons. Num calorímetro hadrônico, os elétrons são usados para estudos de uniformidade e de resolução do calorímetro, posto que depositam praticamente toda sua energia na primeira célula em que tocam. Os múons são usados porque, como tendem a depositar uma pequena e fixa quantidade de energia em toda sua trajetória, permitem que, compensando os efeitos da diferença de distância percorrida na trajetória para cada célula, seja efetuada a calibração nas células do interior do calorímetro. Finalmente, os píons (hadrons), partículas isoladas da classe de partículas para a qual o TileCal é especializado, são utilizados para se verificar a performance do detector nos diversos parâmetros. Efeitos como percentual de vazamento, sinal nas *trigger towers*, espalhamento entre as torres, entre outros, só podem ser bem avaliados com a ajuda de feixes de píons [3, 32].

Assim, conforme apontado acima, a qualidade da calibração depende em muito da quantidade de eventos (de forma a se ter uma boa estatística nos resultados obtidos) e na certeza de que estamos medindo a resposta para os exatos tipos de partículas que pensamos ter. Com relação a este segundo ponto, existe, entretanto, um problema sério. Os feixes gerados pelo SPS, embora de excelente qualidade, produzem, em geral, partículas de contaminação pelo próprio processo de produção do feixe. Como exemplo, podemos citar o caso dos píons, que são sempre produzidos com uma "dose" de múons (até 30%, na prática).

Os elétrons sofrem de um problema ainda maior, pois têm contaminação por píons (tipicamente 30%, embora casos de 40% já tenham sido encontrados) e múons (de 2 a 25%, na prática). Só o feixe de múons é completamente livre de contaminações, posto que é produzido com a passagem de um feixe de píons (com múons) por um bloco de massa (que funciona como um calorímetro mais rudimentar na verdade). Os píons são absorvidos por este calorímetro rudimentar, mas os múons seguem seu caminho até atingir o módulo em teste. Por fim, o processo de alteração do tipo de feixe é um processo complexo e demorado (alguns minutos para cada troca) o que representa a perda de milhares de eventos durante o período de teste experimental.

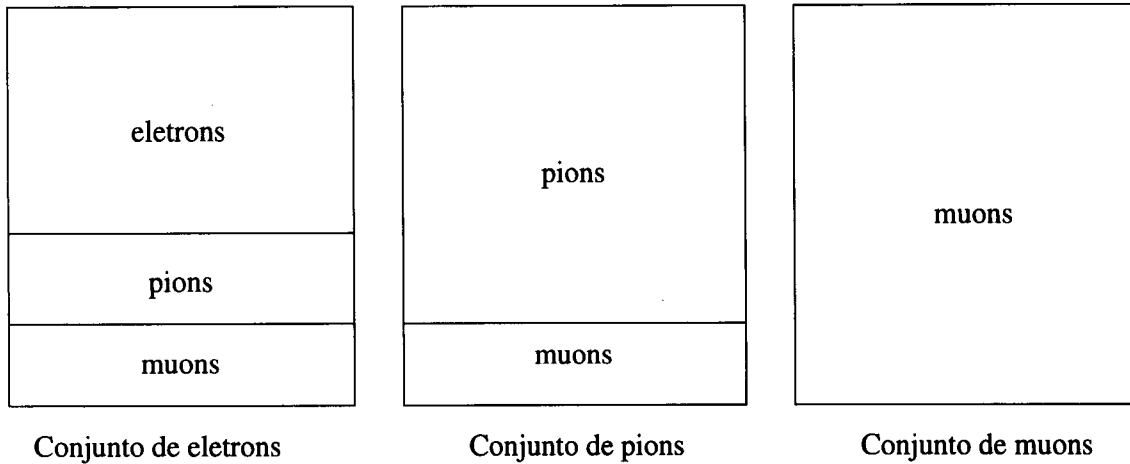


Figura 3.9: Conjuntos de dados experimentais adquiridos com contaminação.

Assim, na prática, os conjuntos de dados adquiridos ficam como na Figura 3.9. O conjunto de múons é o único considerado puro, estando livre de contaminação. O conjunto de píons possui também múons de contaminação. Claramente, a distribuição destes eventos é aleatória dentro do conjunto de dados, sendo necessária alguma técnica que realize a identificação destas classes de partículas. Finalmente, o conjunto de elétrons é o que mais sofre com contaminação, posto que além de elétrons, píons e múons podem ser observados com intensidade.

Como já apontado em [33] a resposta do calorímetro varia com a classes da partícula incidente, tanto no seu sinal temporal, como no espectro de distribuição de energia entre as células e na energia total.

Na Figura 3.10, temos a deposição de energia no TileCal para um evento típico de elétrons com uma energia nominal do feixe de 20 GeV. Cada pequena área definida na superfície que aparece nesta figura representa uma célula do calorímetro. A altura que se eleva cada torre, a partir da superfície, representa a quantidade de energia depositada em cada célula. Para referência, marcamos no desenho o valor de energia da célula que mais recebeu energia, que neste caso foi a primeira encontrada pelo elétron (a terceira célula da primeira camada).

As interações eletromagnéticas, como a do elétron, do pósitron ou fótons, se dão a nível da eletrosfera dos átomos, logo, com uma seção de choque relativamente grande. Além do mais a força eletromagnética opera a qualquer distância. Dessa forma, o elétron incidente pode interagir rapidamente perdendo energia quando sua

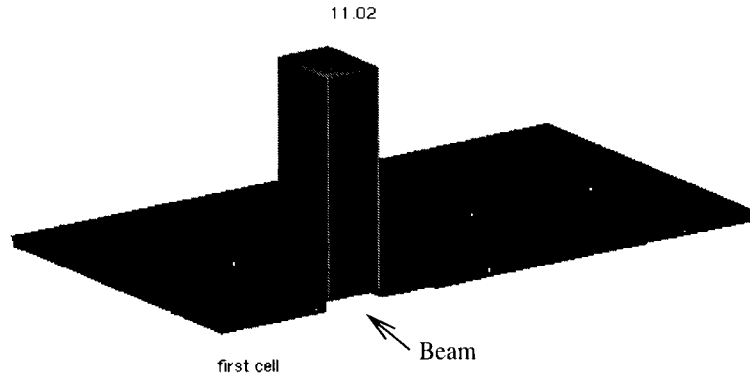


Figura 3.10: Um evento típico de elétron depositando energia no calorímetro TileCal.

trajetória se curva pela atração exercida por algum átomo do material. Por isso o elétron deixa um rastro de energia concentrado, a partir do seu ponto de impacto.

Na Figura 3.11, temos eventos típicos de píons (à esquerda) e de múons (à direita). O evento de pión sensibiliza completamente a torre de *trigger* sobre a qual este incidiu, sensibilizando não só a primeira célula em que toca, mas, principalmente, a segunda, e, mesmo a terceira e última célula da torre. Isso ocorre, posto que a interação do pión se dá mais ao nível da força nuclear forte, que, embora tenha um menor raio de ação, é de magnitude muito maior. Isso implica que, das interações produzidas pelo pión, algumas partículas neutras tais como π^0 e nêutrons podem ser geradas, levando a uma flutuação muito grande da estrutura de deposição de energia apresentada por eventos de píons. A partícula π^0 , por exemplo, não sofre de nenhuma interferência relativa as eletrosferas dos átomos, logo, podendo percorrer longas distâncias na massa do calorímetro, vindo a colidir mais no meio da torre. Isso explica o formato visto na figura e também o valor de pico bem menor (11 GeV aproximadamente, no caso dos elétrons, e 5,6, no caso dos píons). A energia, no caso dos píons, se dispersa por uma região bem maior no detector, impossibilitando um pico tão alto como vemos no caso dos elétrons.

Os múons, finalmente, não deixam praticamente nenhuma energia no detector. Eles deixam apenas um pequeno traçado na direção em que penetraram no detector. No caso, podemos ver na figura da direita da Figura 3.11, que apenas a torre de *trigger* sobre a qual incidiu a partícula foi sensibilizada. Vemos que o valor máximo não atingiu nem mesmo 1 GeV, mesmo quando o feixe era de 20 GeV.

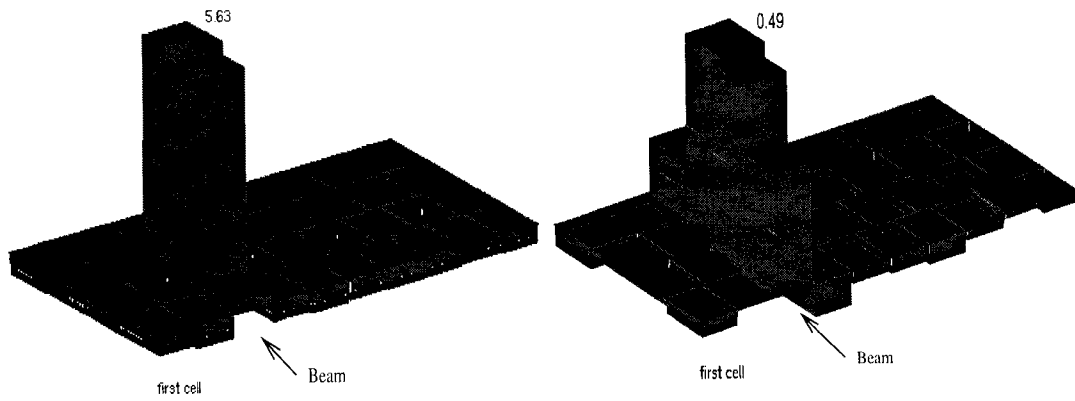


Figura 3.11: Evento de pión (à esquerda) e de múon (à direita).

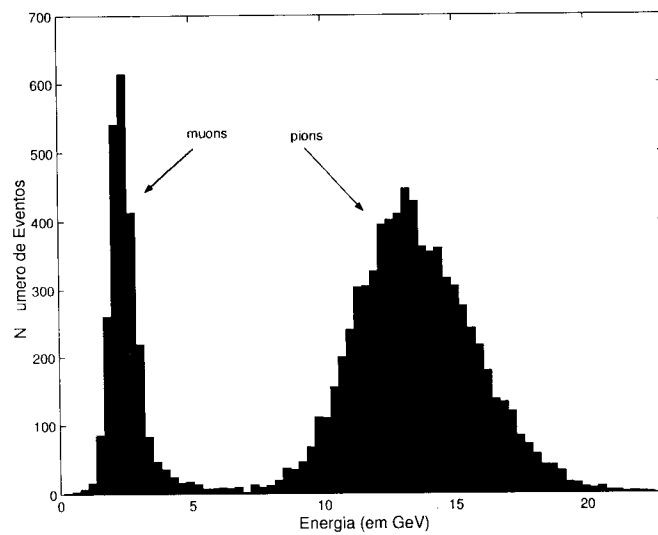


Figura 3.12: Eventos um feixe de pions de 20 GeV com contaminação de múons.

3.2.1 Contaminação no Feixe

Toda esta discussão, que até agora apresentamos de forma qualitativa, merece um tratamento mais estatístico, na tentativa de estabelecermos cortes que possam servir para discriminar entre as diferentes partículas. Na Figura 3.12, temos um histograma da energia depositada por eventos provenientes de um feixe de píons DE 20 GeV. A princípio, eventos de píons deveriam depositar um valor de energia próximo ao valor nominal de energia do feixe (que nesse caso era 20 GeV). Na figura, podemos observar dois modos na distribuição, com uma boa parte dos eventos cobrindo a faixa de 7 GeV até próximo à 20 GeV. Entretanto, uma outra distribuição de eventos cobre a faixa de menor energia, estendendo-se até cerca de 6 GeV. Esses eventos com um menor valor de energia são, na verdade, os eventos de múons que estão contaminando o conjunto de dados, que deveria conter somente eventos de píons. Usando um corte na energia total é, então, possível se obter um método para identificar este tipo de contaminação. Usaremos este tipo de informação para estabelecer cortes que possam validar o sistema neural desenvolvido no presente trabalho. Para baixas energias, como o presente caso de 20 GeV, temos uma faixa na qual há uma dúvida sobre os eventos, dados o fato de que as duas distribuições se encontram. Quanto mais alta a energia do feixe (por exemplo, 100 GeV), menor é esta região de dúvidas.

No caso de elétrons, o problema é bem mais complexo. Múons continuam depositando pouca energia e, da mesma forma que no caso da contaminação do feixe de píons, podem ser eliminados com as mesmas restrições anteriores. Já a eliminação de píons de contaminação apresenta uma dificuldade adicional, já que tanto elétrons quanto píons depositam uma quantidade de energia elevada no detector e um simples corte de energia não é suficiente para a discriminação de elétrons e jatos.. Ambos os tipos de partícula depositam energia ao mesmo nível. Isso significa que devem ser exploradas outras características da deposição para se realizar a discriminação.

Tipicamente, a informação longitudinal da cascata de partículas traz informação relevante ao problema. Uma possível variável é a fração de energia depositada na primeira camada de células do detector. Como havíamos já comentado, os elétrons tendem a depositar a maior parte de sua energia na primeira camada de células, enquanto os píons distribuem sua energia por todas as camadas, tendendo

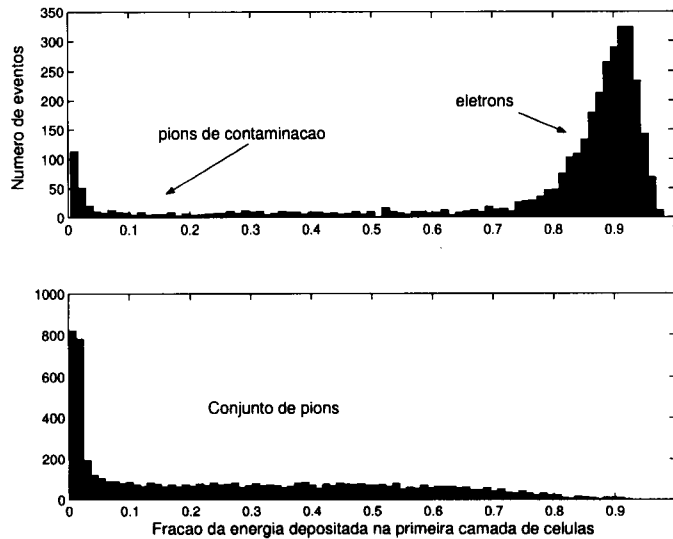


Figura 3.13: Fração da energia depositada na primeira camada de células do TileCal para eventos de um feixe de elétrons com contaminação por píons (topo) e eventos de um feixe purificado de píons (embaixo). Ambos feixes de 100 GeV.

a depositar, em média, mais energia na segunda camada. Na Figura 3.13, temos, no topo, a fração de energia depositada na primeira camada por eventos de um feixe de elétrons com contaminação de píons. Na parte inferior, temos um conjunto de píons puro, aonde os eventos de múons foram removidos por um corte em energia. Como podemos ver, os píons, no caso, a 100 GeV, se estendem pela faixa de 0 (0%) a, aproximadamente, 0.7 (70%) nesta variável. Este valor depende, é claro, da energia considerada. Os eventos do histograma superior deveriam depositar muita energia nesta primeira camada de células (apenas eventos com mais de 70% de sua energia seriam, então, encontrados) mas a contaminação de píons leva a existência de alguns eventos que depositam menos do que este valor nesta segmentação do calorímetro. Assim, podemos estabelecer também um corte para distinguir os elétrons dos píons.

Finalmente, em alguns testes com feixes foi utilizado o contador Cherenkov [34]. Este detector, na verdade, é apenas um recipiente contendo uma mistura de gases. As partículas no feixe estão a velocidades próximas a da luz. Ao entrarem neste meio, no qual a velocidade da luz é bem inferior a do vácuo, emitem luz numa tentativa de perder energia e chegar a velocidade da luz no meio. Esta luz é coletada através de cintiladores e adquirida. Ora, para uma mesma energia, elétrons, sendo partículas de massa muito menor do que os píons, adquirem uma velocidade bem

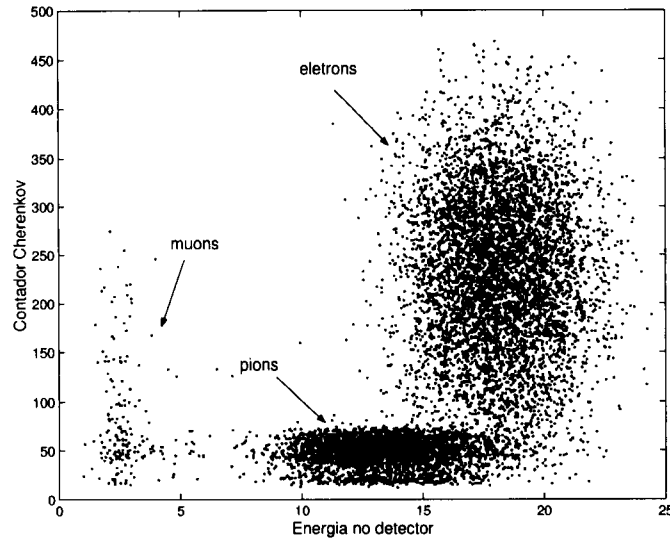


Figura 3.14: Contador Cherenkov correlacionado com a energia total depositada no módulo do TileCal para eventos de um feixe de elétrons a 20 GeV contaminado por píons e múons.

maior do que estes últimos no contador Cherenkov. Dessa forma, os elétrons emitirão mais luz de Cherenkov, como é chamada essa luz, seguindo o nome do pesquisador que a previu, do que os píons, sendo esta mais uma forma de se separar estes dois tipos de partículas.

Para os testes em que foi instalado o Contador de Cherenkov, temos este método auxiliar para verificação de resultados. Estes resultados estavam disponíveis nos testes realizados, a 20 GeV. Neste caso, os resultados provenientes de cortes como o da primeira camada são bem mais pobres, dado o fato de que, a 20 GeV, elétrons e píons desenvolvem menos sua cascata no interior do calorímetro do que a altas energias (100 GeV, por exemplo). Logo, as cascatas geradas por estas partículas ficam semelhantes (depositando um pouco de energia na primeira camada) para a resolução fornecida pelo TileCal. Se a primeira camada de células fosse fracionada em células menores, provavelmente seria mantido o mesmo nível de separabilidade mesmo para menores energias. Entretanto, a resolução é escolhida em função dos requerimentos do experimento ATLAS, que exige células mais largas para as partículas altamente energéticas que devem ser produzidas no LHC.

Na Figura 3.14 temos o resultado do contador Cherenkov correlacionado com a energia total. Podemos ver que as partículas que depositam altas energias passam

a ser distinguíveis em dois grupos, quando aplicado o corte no sinal do contador Cherenkov, mostrando os elétrons (com maior valor no contador) e os píons, com menor contagem. Dessa forma, temos mais uma metodologia que pode ser usada para realizar a discriminação, neste caso complexo.

Capítulo 4

O Sistema de Aquisição de Dados

Neste capítulo, descreveremos melhor o sistema responsável pela aquisição dos dados já digitalizados pela eletrônica do detector, detalhando as diversas etapas realizadas durante a aquisição e as máquinas utilizadas. Além do conhecimento relevante sobre o sistema de aquisição de dados (DAQ) e das contribuições desta tese na área, o trabalho junto ao DAQ leva a uma melhor compreensão do sistema desenvolvido para a aplicação *online* que será discutida no Capítulo 8.

4.1 O DAQ para Testes do TileCal

Conforme havíamos descrito, o último estado de tratamento dos dados produzidos pelo detector durante o teste com feixes constitui na coleta dos sinais digitais produzidos pelas placas digitalizadoras realizada pela placa de interface. Estes dados são enviados através de fibras óticas para fora da gaveta que contém a eletrônica do detector, aonde serão recebidos por um S-Link [35]. Esta conexão (*link*), que é utilizada para transmissão rápida de dados entre os vários estágios do sistema de aquisição de dados, foi implementada por pesquisadores do próprio CERN para as exigências de velocidade dos experimentos do LHC. Os dados recebidos por via ótica são inseridos, por meio de uma placa eletrônica de controle, no espaço de memória de uma placa com o processador Power-PC. Desta forma, os dados passam a um meio digital programável, o que é bastante cômodo para se realizar os processos necessários ao tratamento dos dados.

Na Figura 4.1, temos o esquema geral do sistema de aquisição de dados

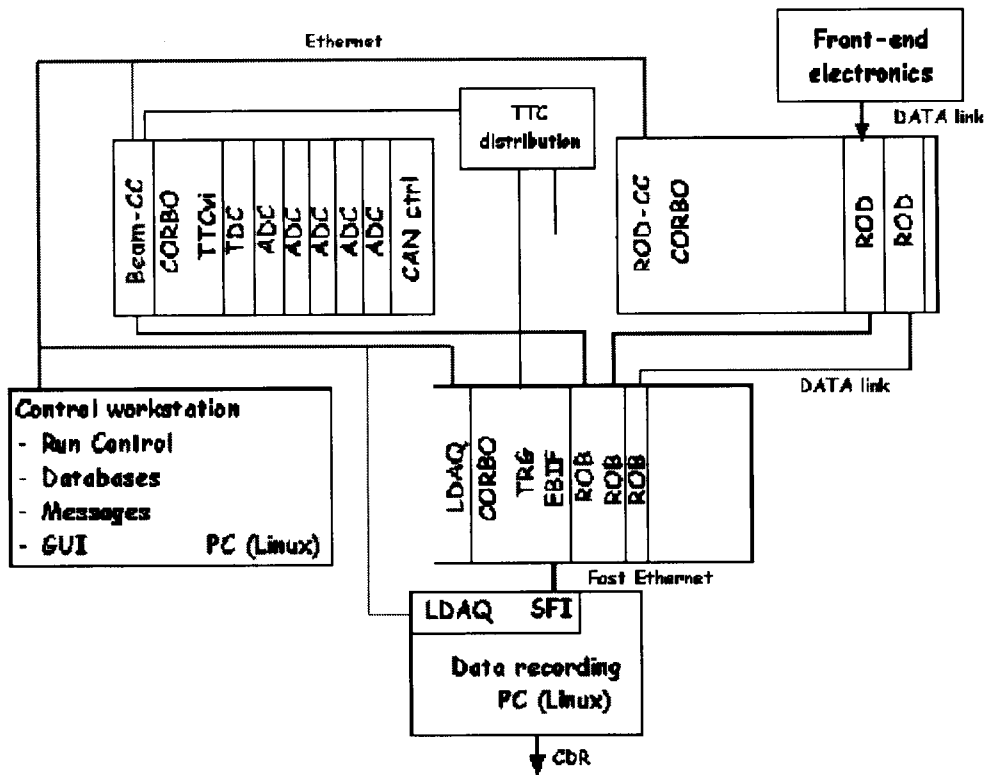


Figura 4.1: Sistema de aquisição de dados do TileCal para testes com feixes.

para testes com feixes do TileCal. A parte indicada como *Front-end electronics* é a gaveta do módulo em teste, que envia os sinais digitalizados por via ótica. O sistema é constituído de três bastidores (*Crates*) VME (os dois quadros da parte superior, marcados a esquerda com Beam-CC e ROD-CC e o quadro logo abaixo destes, marcado com LDAQ) e dois PCs, rodando o sistema operacional Linux [36]. As Crates são constituídas de módulos funcionais (como ADCs, TDCs, e sistemas de apoio ao sistema de *Trigger* e TTC) e módulos Power-PC, rodando o sistema operacional em tempo real Lynx-OS [37].

As duas Crates VME na parte superior da figura são a *Beam Crate* (esquerda) e a *Read-Out Driver* (ROD) Crate (direita). A Crate abaixo destas é a *Read-Out Crate*. A *Beam Crate* é responsável, inicialmente, por criar uma partição TTC somente para o TileCal. Desta forma, o sistema de *Trigger* completo do experimento fica em sincronia com o sistema do acelerador SPS. Através do módulo de controle (Beam-CC), é possível se programar o módulo TTCvi [38], que, através de registros mapeados no barramento VME, ajusta esta sincronia e manda comandos para a eletrônica da gaveta. Além disso, o Beam-CC, recolhe através do barramento VME,

os dados adquiridos pelos vários ADCs (*Analog-to-Digital Converters*), TDCs (*Time-to-Digital Converters*), que são as informações vindas de detectores auxiliares, tais como as Câmaras de Fios, os Contadores de Cintilação, o Contador *Cherenkov*, e, finalmente, as respostas dos somadores analógicos das *Trigger Towers*. Essas informações formarão um bloco no conjunto de dados final, relativo a um único evento. O módulo marcado como CORBO é responsável por gerar um sinal de *busy*. Este sinal marca o período no qual o sistema está trabalhando (no caso da *Beam Crate*, os sinais estão sendo adquiridos e enviados via barramento VME ao módulo Beam-CC). Enquanto o sinal de *busy* estiver ativado, o sistema não irá processar um outro evento, se alguma outra partícula vier a ser enviada pela linha do feixe.

A ROD Crate é responsável por receber diretamente os dados vindos do calorímetro. As placas marcas como ROD (*Read-Out Driver*) têm, cada uma, quatro S-Links, para receber os dados vindos de até três gavetas (em três dos S-Links) e poder emitir, pelo quarto S-Link, os dados já processados para o próximo estágio de aquisição. Os dados recebidos são agrupados pelas RODs em pequenos blocos, com uma formatação e marcadores específicos, que permitem, mais tarde, a reconstrução do evento completo através de *software* específico. A ROD Crate possui também um módulo CORBO para geração de sinal de *busy*, que é, neste caso, o intervalo de tempo em que um evento é recebido, processado e enviado adiante, estando o sistema pronto para receber novos eventos. O módulo ROD-CC somente controla e monitora as RODs através da transmissão de pequenas mensagens pelo barramento VME. O processamento realizado por esta placa, que é bem leve em termos computacionais, será de vital importância quando, mais tarde, discutirmos a implementação do sistema neural *online* no Capítulo 8.

A terceira Crate é a Read-Out Crate, ROC. A ROC possui três módulos *Read-Out Buffers* (ROBs), que recebem, através de seus S-Links, todos os blocos de dados vindos das RODs, e dos ADC/TDCs da *Beam Crate* já coletados pelo módulo Beam-CC. Estes blocos de dados de eventos, vindos de diferentes canais de aquisição, são chamados de fragmentos. Os fragmentos possuem identificadores do evento a que pertencem, assim fragmentos de mais de um evento podem ser armazenados nas ROBs e posteriormente reconhecidos. O Módulo *Event Builder InterFace* (EBiF), devidamente acionado pelo módulo de *Trigger* (TRG), coleta, através do barramento

VME da Crate, os dados de um único evento que estão armazenados nas três ROBs e monta uma única estrutura de dados no formato específico do ATLAS. Os eventos assim agrupados são, então, enviados por uma conexão de alta velocidade (marcada como *Fast Ethernet*) para o estágio de armazenamento. Esta Crate também possui um módulo CORBO para sinal de *busy*, que, no caso, só existe quando alguma das ROBs está completamente cheia de fragmentos de eventos e o sistema tem que esperar que o EBiF colete parte destes fragmentos para que novos eventos possam ser aceitos. O módulo LDAQ controla e monitora todos estes módulos, assegurando o envio de mensagens entre os mesmos e seu perfeito funcionamento.

O PC responsável pelo armazenamento de dados (*Data Recording*), recebe o nome de *Sub Farm* (que representa um dos conjuntos de potentes processadores que realiza esta tarefa). É que, no ATLAS, neste estágio, antes de se armazenar os eventos, o último estágio de *trigger* realiza um processamento computacionalmente pesado, que será realizados por grupos ("fazendas") de computadores, usando todas as informações vindas de todos os detectores na seleção de eventos. O ponto marcado como SFi (*Sub-Farm Input*) recebe através da conexão rápida e dedicada os eventos já completamente montados e os organiza em arquivos. Estes arquivos correspondem a um número de eventos fixado pelo usuário, quando as condições do feixe são mantidas constantes. Os dados que compõem estes arquivos compõem uma rodada *run* do experimento; ou seja, uma dada seleção de feixe, ponto de impacto e seleção de parâmetros para o teste. Os arquivos assim produzidos são depois transmitidos por uma conexão *Ethernet* comum para o CDR (*Central Data Recording*), aonde gravadores de fitas armazenam os dados para análises futuras (*offline*).

O outro PC que faz parte do sistema, executa o *software* de controle do sistema de aquisição. Este PC envia, através de conexões de Internet, as informações necessárias para configurar (através de parâmetros fornecidos pelo usuário) e acionar, de forma seqüencial e síncrona, todos os subsistemas pertencentes ao sistema de aquisição de dados. As unidades ROD-CC, Beam-CC e LDAQs estão sempre em comunicação com este controle. Além disso, uma interface ao usuário (*GUI-Graphical User Interface*) permite ao mesmo um controle gráfico através de seleções por mouse ou imagens coloridas. Na Figura 4.2, temos a tela inicial do sistema de aquisição de dados para o TileCal.

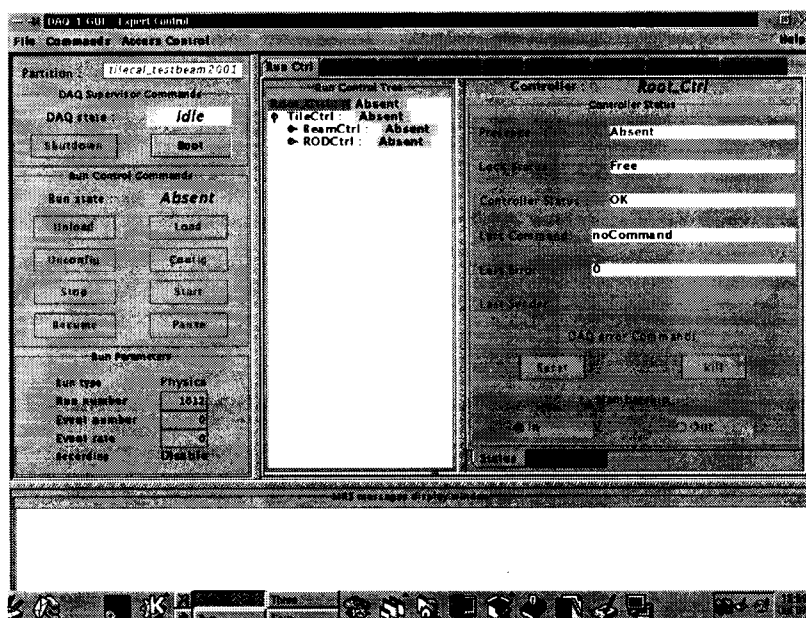


Figura 4.2: Janela do sistema de controle do DAQ do TileCal.

Este *software*, que é chamado de *Online Software* [39], escrito em C++ com ajuda de algumas bibliotecas de padronização para diferentes sistemas operacionais [40], constrói objetos de *software* que se comunicam para transmissão de informações. Os principais objetos são chamadas de controladores. Dois principais conjuntos de informações, uma base de dados unificada, escrita em linguagem XML [41], e o IS - *Information System* (Sistema de Informações) [42] são utilizados como sistema de distribuição de informações entre os controladores. Um sistema de mensagens, o *Message Report System* [43], é utilizado para mensagens que devem aparecer na tela principal da interface gráfica.

Os módulos Beam-CC, ROD-CC e LDAQs possuem, cada um, seu controlador. Estes controladores são inicializados pelo *Online Software*. Uma vez inicializados, estes programas devem monitorar os módulos por eles controlados em suas Crates. Programas nos módulos Power-PC das Crates podem ser também inicializados. Na Figura 4.2, temos, na parte central, um exemplo de uma árvore de controladores. Nesse caso, apenas a *Beam Crate* e a *ROD Crate* estavam presentes. Um terceiro Controlador *TileCtrl* tem que ser definido para sincronizar os demais (em algumas situações, os controladores devem receber comandos numa ordem específica e isso leva à necessidade de um controlador extra para gerenciar essa ordem). O *Root_Ctrl* é o controlador principal e tem sempre que existir. Com os botões na

parte esquerda da tela, pode-se inicializar o sistema inteiro (Comando *Boot*). O botão *Load* carrega a base de dados em todos os controladores, fazendo com que eles recebam os parâmetros que lhes são necessários. O botão *Config* faz com que os controladores passem os seus parâmetros para os módulos programáveis, preparando o sistema para entrar em funcionamento, o que se consegue com o botão *Start*. No momento em que a figura foi capturada, os controladores não tinham sido inicializados e por isso estão todos marcados como *Absent* (ausentes). Outra informação importante nesta figura é o número da rodada (*Run Number*). Este número designa o conjunto de dados aonde foram mantidas as mesmas características de posição e tipo de feixe.

O pacote do *Online software* vem sendo editado em versões periódicas [44] nas quais erros vão sendo corrigidos e novas ferramentas vão sendo adicionadas. Desta forma, o produto se torna cada vez mais robusto e flexível, atendendo às necessidades de todos os subdetectores do ATLAS. O subdetector de múons, foi o segundo a aplicar o *Online software* em seus testes com feixes. No caso deste subdetector, a ROC Crate foi substituída por um PC aonde os módulos (ROBs) eram placas conectadas ao barramento PCI dos mesmos [24]. Dada a possibilidade de se comprar PCs com um relógio interno de maior frequência e com processadores cada vez mais robustos a preços bem mais em conta do que os módulos Power-PC, consegue-se um sistema com velocidade crescente com um preço cada vez menor. Isso viabiliza muitas das experiências que vêm sendo realizadas atualmente.

Para que se possa reconstruir os dados de forma offline, os mesmos devem ter um formato específico. Este formato, especificado para todo o detector ATLAS [45], define marcas de início de blocos de dados. O número hexadecimal 0xAAAAAAAAA marca o início de um novo evento. Já o número 0xEEEEEEEEE marca, dentro de um evento, o início de um grupo de dados referente a uma das RODs de um subdetector do ATLAS. Na Figura 4.3 temos um exemplo de um bloco de dados para a ROD do TileCal.

Como podemos verificar, o bloco começa com o marcador do cabeçalho que, como havíamos dito, é o número 0xEEEEEEEEE e é produzido pelas RODs. Outros números importantes são o Identificador da Fonte (*Source Identifier*) que diz qual subdetector (existe uma tabela com os números para cada subdetector do ATLAS)

Start of header marker
Header size
format version number
Source identifier
Level 1 ID
Bunch Crossing ID
Level I trigger type
Detector event type
Data Elements
Status Elements
Number of status elements
Number of data elements

Figura 4.3: Formato de dados para uma ROD do ATLAS.

criou aquele bloco de dados. O *Bunch Cross ID* também faz parte deste bloco, assim como o *Trigger Type*, que traz informação sobre o tipo de evento (evento físico, de calibração, entre outros). O trecho mais importante corresponde aos Elementos de Dados (*Data Elements*), que são completamente específicos do detector. No caso do TileCal, neste trecho dos dados vêm as amostras do sinal produzido por cada fotomultiplicadora de cada célula, junto com bits verificadores de erros numa seqüência já conhecida [46].

Os trabalhos desenvolvidos no âmbito do *Online Software* estarão registrados na próxima seção. Através destes trabalhos, foram adquiridos vários conhecimentos necessários para a aplicação neural *online*, conforme veremos mais adiante.

4.2 Exercícios no DAQ para o TileCal

Nesta seção, descreveremos três importantes exercícios que foram fundamentais para o posterior desenvolvimento da aplicação neural. Vamos descrever, inicialmente, o sistema de monitoramento *online*, utilizado para realizar histogramas durante a aquisição de dados. Na seqüência, vamos descrever o sistema de transmissão de dados via barramento VME. Este sistema, mais flexível e mais rápido que o implementado anteriormente, possibilitou que o desenvolvimento do sistema neural *online* fosse realizada sem reduzir significativamente a taxa de aquisição do

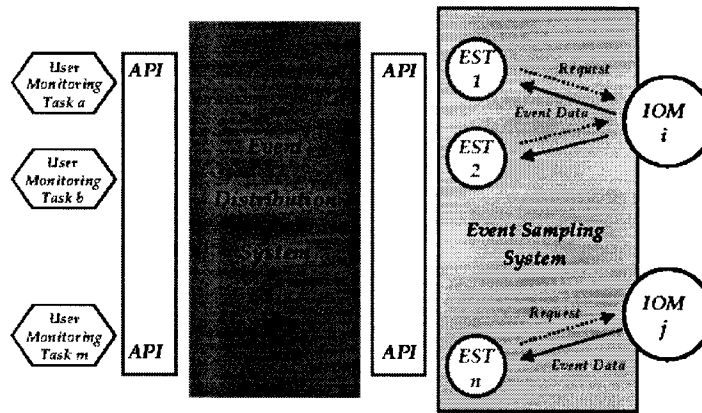


Figura 4.4: Organização do sistema de monitoração.

sistema de aquisição de dados. Finalmente, discutiremos uma metodologia de compressão de dados que está sendo estudada como uma alternativa para se reduzir o fluxo de dados no sistema de aquisição, melhorando sua performance.

4.2.1 O Sistema de Monitoração *Online*

No projeto do sistema de aquisição do ATLAS, está prevista a existência de um sistema de monitoração dos subdetectores. Este sistema, a princípio, deve ter a opção de amostrar uma parte dos eventos sem perdas na taxa de aquisição, ou adquirir todos os eventos com uma redução devidamente quantificada desta taxa [47]. O sistema de monitoração foi implementado através do Sistema de Informações para possuir duas entidades principais : O Sistema de Distribuição de Eventos (*Event Distribution system*) e o Sistema de Amostragem de Eventos (*Event Sampling System*). Na Figura 4.4 vemos a organização destas entidades.

As Atividades de Monitoração (*Monitoring Tasks*) pedem, através de um sistema de endereçamento, os fragmentos vindos dos diferentes Módulos de Entrada e Saída (*Input-Output Modules - IOM*) ao Sistema de Distribuição de Eventos. Os IOMs podem ser RODs, ROBs, o EBiF ou o SF, ou seja qualquer unidade funcional que lide diretamente com o fluxo de dados. Se o Sistema de Distribuição possuir fragmentos recentemente adquiridos, fornece-os a Atividade de Monitoração que esteja pedindo. Caso contrário, o mesmo faz uma requisição ao Sistema de Amostragem de Eventos que se comunica diretamente com a IOM requisitada parte (ou todos)

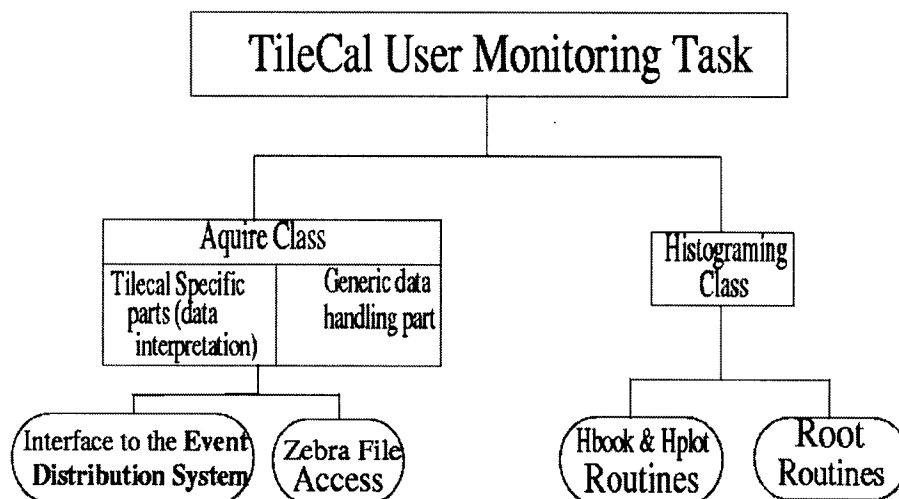


Figura 4.5: Organização em classes da Atividade de Monitoração.

e acessa os fragmentos. Esta organização exige que as IOMs sejam projetadas com habilidade para responder ao sistema de monitoração. Isso se faz através de uma interface específica, chamada API (*Application Public Interface*). Também as Atividades de Monitoração devem obedecer a uma API específica para corretamente endereçarem as IOMs e receber destas os fragmentos de eventos.

Assim, projetamos uma Atividade de Monitoração com a estrutura de classes (posto que o programa foi feito em linguagem C++) que pode ser vista na Figura 4.5. Duas classes principais, a de aquisição (*Acquire Class* para receber os fragmentos do sistema de monitoração) e a de produção de Histogramas (*Histograming Class* - para acumular e plotar histogramas), foram implementadas para processar os fragmentos de eventos vindos do detector do TileCal e dos detectores auxiliares (cujos dados são lidos pela *Beam Crate*) [48].

A *Acquire Class* tem uma parte de sua aquisição feita por rotinas que usam as interfaces com o sistema de distribuição de dados. Estas rotinas são feitas de forma geral, que poderia ser utilizada em qualquer outro subdetector do ATLAS. Esta parte foi desenvolvida para trabalhar numa *thread* diferente da do resto do programa. Este caminho foi utilizado para evitar que um bloqueio qualquer no sistema de monitoração, ou na própria IOM que estava fornecendo os eventos, bloqueasse também o restante do programa, incluindo aí a apresentação de histogramas ou até mesmo a interface com o usuário que depois foi implementada. A parte específica para interpretação dos fragmentos de dados como variáveis com significado físico é

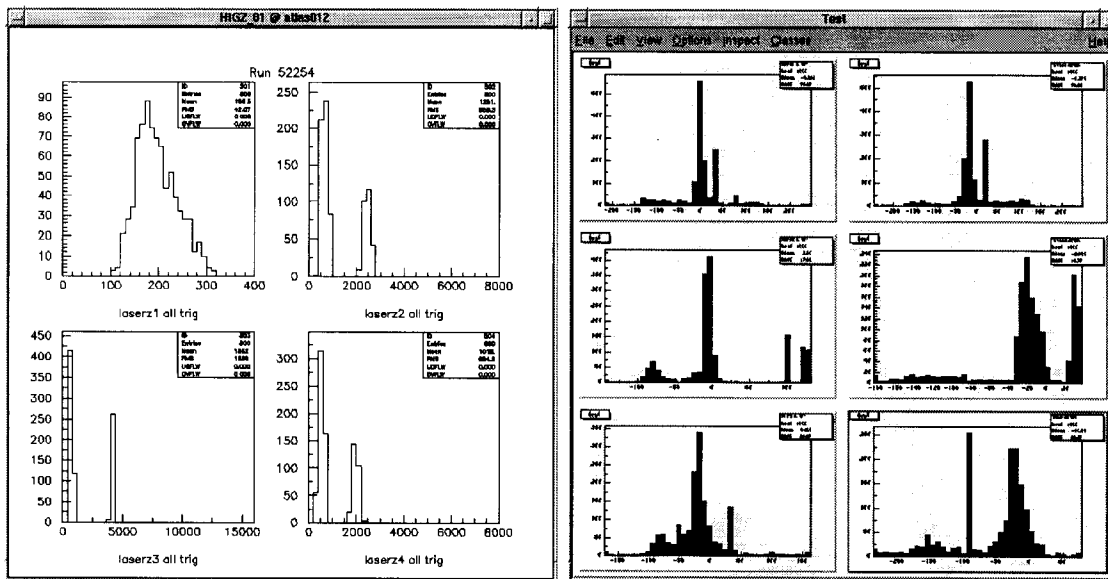


Figura 4.6: Janela com histogramas de variáveis do TileCal em HBook e HPlot (esquerda) e ROOT (direita).

uma cópia simplificada das rotinas utilizadas em *offline* para decodificar os eventos. Era também possível, para ajudar no desenvolvimento, receber eventos de arquivos Zebra, que é um formato padrão do CERN para armazenagem de dados [49].

A segunda classe importante, a de produção de histogramas, recebia os eventos devidamente decodificados em variáveis de uma estrutura em estilo C e verificava quais destas variáveis tinham sido requeridas pelos usuários do programa. Estas variáveis selecionadas enchiam histogramas que iam sendo apresentados assim que um certo número de eventos fosse adquirido (por exemplo, como foi utilizado, a cada 200 eventos a tela era atualizada). Para acumulação e apresentação de histogramas, ou seja, para a implementação da classe de produção de histogramas, dois pacotes de rotinas do CERN foram utilizados : O HBOOK (para acumulo) e HPLOT (para apresentação) fizeram parte de uma dessas implementações [50] e o pacote ROOT fez parte da outra [51]. Na Figura 4.6, temos estas duas opções apresentadas. Os pacotes HBOOK e HPLOT produziram para as variáveis do sistema de laser (injetor de luz nas telhas para teste do sistema de aquisição) o histograma da esquerda. Já a tela da direita foi produzida com ajuda das rotinas do ROOT, fazendo histogramas relativos às câmaras de fios. Os pacotes HBOOK e HPLOT (que foram implementados originalmente em Fortran) são mais tradicionais e mais conhecidos dos físicos,

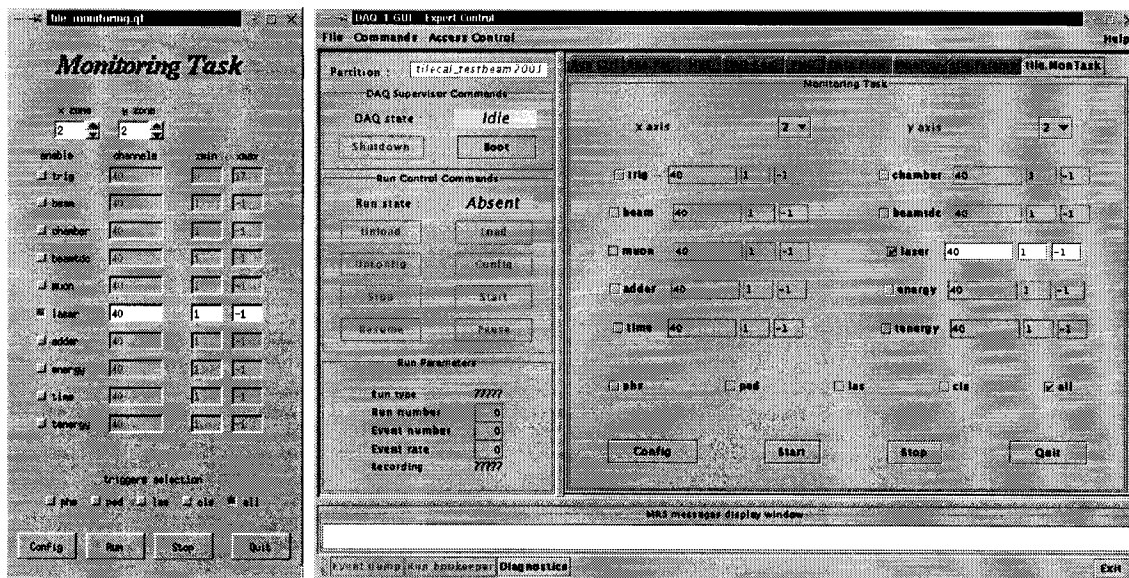


Figura 4.7: Interfaces Gráficas ao Usuário em QT (esquerda) e integrado ao sistema de aquisição (direita).

por isso, foi esta opção que teve uma melhor aceitação de uso. O pacote ROOT de visual mais arrojado e desenvolvido completamente em C++, também foi utilizado por representar uma opção mais moderna de uso.

As primeiras versões do programa exigiam um grande número de variáveis na linha de comando, dificultando em muito sua utilização. Para contornar este problema, foram implementadas duas versões de uma interface gráfica ao usuário. A primeira, para uso independente da interface ao usuário do sistema de aquisição de dados, utilizou o pacote QT [52] para desenvolvimento dos painéis, botões, entre outros, para compor a interface. Também foi desenvolvido um outro painel, em linguagem JAVA, utilizando os recursos do Sistema de Informações, para transmitir as informações do painel até a aplicação real em C++. Estas duas opções aparecem na Figura 4.7. Na esquerda temos o painel QT e na direita o painel em JAVA integrado a tela do sistema de aquisição de dados.

Os botões em ambas implementações servem para selecionar quais histogramas devem ser apresentados. No caso, ambos estão com os histogramas de laser, prontos a serem apresentados. Abaixo, pode-se selecionar o tipo do evento (de física, de pedestal, de laser, de calibração ou todos) que se deseja ver. Pode-se selecionar a aparência da tela, o que altera o número de histogramas que aparecem na tela a

cada atualização. Isto ajuda muito, posto que algumas variáveis são, na verdade, vetores que devem ser acumulados em histogramas separados. Pode-se atingir até 48 histogramas expostos por vez (o que inclui dois canais não utilizados), o que é um número impossível de se ver numa tela tão pequena de um monitor comum. Com a seleção da geometria, pode se dividir este número em grupos menores, facilitando a visualização.

Foi feito um teste para se verificar a taxa de eventos que o sistema é capaz de processar na unidade de tempo. O sistema processou 149 eventos por segundo, o que é uma amostra bastante razoável dos dados adquiridos pelo TileCal. Durante a fase de preparação para testes com feixes, muitas calibrações devem ser realizadas, e para tal, são gravados arquivos com enormes quantidades de dados inúteis. Com o uso desta Atividade de Monitoração, foi possível se realizar toda esta tarefa sem se gravar nenhum dado, apenas com os resultados que apareciam nos histogramas *online* [53].

O ponto mais importante deste projeto foi o estudo das rotinas de reconstrução de dados *offline* e o trabalho com arquivos Zebra e as rotinas HBOOK e HPLOT. Este estudo permitiu uma maior experiência de programação no ambiente do DAQ, além de fornecer uma visão bem mais profunda dos procedimentos de reconstrução de valores de energia a partir das amostras do sinal nas células do calorímetro. Esta informação foi fartamente utilizada durante a implementação do sistema *online* de redes neurais, dado o fato de que é necessário se converter as amostras contidas no fragmento do evento para valores de energia, antes de se utilizar estes valores como entrada para o processamento neural. Analisaremos melhor estes detalhes quando formos discutir este processamento (Capítulo8). Outro ponto vital, que será de máxima importância mais tarde, é a transmissão de dados no barramento VME, que investigaremos na próxima subseção.

4.2.2 Transmissão de Dados no Barramento VME

Conforme já apontado, quando descrevemos a ROD Crate, algumas comunicações são realizadas entre o ROD-CC e os módulos ROD. Esta comunicação é realizada através do barramento VME. Porções da memória de um dos processadores envolvidos na transmissão de informações são mapeadas no outro processador,

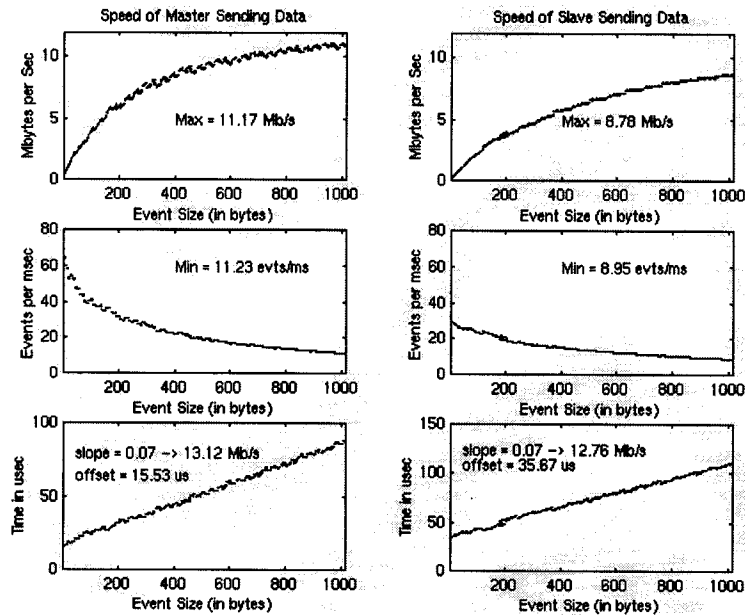


Figura 4.8: Medidas da velocidade da transmissão de dados em modo DMA.

podendo-se fazer leituras e escritas nesta memória remota através da interface VME existente em cada um dos módulos Power-PC. Como as transmissões realizadas aconteciam a uma pequena taxa e envolviam uma pequena quantidade de dados, nunca se havia considerado uma possível restrição de velocidade.

Dada a forma simples com que foram implementadas, inicialmente, as rotinas de transmissão de dados, o nível de velocidade atingido era realmente bem pequeno (da ordem de 1,5 a 2 MB por segundo). De forma a se realizar transmissões de dados mais intensas, necessárias à aplicação neural, foram implementadas rotinas opcionais que se baseavam no uso de transmissões DMA (Direct Move Accelerator). Este procedimento é realizado por um processador dedicado, capaz de transmitir diretamente blocos de dados do espaço de memória de um processador para o outro. Como estas rotinas de DMA exigem um tempo para inicialização do processador DMA, elas são contra-indicadas quando o número de bytes a ser transmitido é pequeno (menor que aproximadamente 40 bytes).

Na Figura 4.8 vemos algumas medidas realizadas da velocidade de processamento do sistema. Na primeira linha, temos a quantidade de MBytes transmitidos por segundo. Na segunda linha, o número de eventos transmitido por milissegundo, e, finalmente, na última linha, temos o tempo por transmissão. Na coluna esquerda, temos o controlador (a ROD-CC é a placa mestra) enviando dados. Na

da direita, o módulo ROD (que chamamos de escravo) é quem gerencia o envio de dados. Foi realizada a curva para diferentes tamanhos do fragmento de dados (No caso do TileCal o tamanho do fragmento é em torno de 750 Bytes). Utilizando-se interrupções VME para parar o processamento enquanto a transmissão não havia ainda sido completada, foi possível atingir uma taxa máxima de 11 MB/s nas transmissões realizadas pela placa mestra e 8,8 MB/s no sentido inverso. Ao avaliarmos o tempo unicamente utilizado para transmissão e o tempo morto antes do início de cada transmissão (*offset*), verificamos que as transmissões DMA levam a um tempo morto alto (15 μ s para transmissões do mestre e 35 μ s do escravo). Este tempo é bem superior ao da transmissão comum de dados sem DMA (cerca de 3 a 4 μ s por palavra transmitida) justificando o uso deste último método quando não for transmitida uma grande quantidade de dados [54].

Dada a flexibilidade da biblioteca em C++, que disponibiliza as transmissões no barramento VME com e sem o auxílio de interrupções, esta foi disponibilizada para outros detectores do ATLAS. Para auxiliar os novos usuários, que ela possa vir a ter, foi escrito um guia do usuário [55].

Quando formos estudar a implementação *online*, veremos a importância deste ganho de velocidade na viabilização do procedimento neural durante os testes com feixes.

4.2.3 Compactação de Dados

Como já havíamos mencionado anteriormente, o volume de dados no detector ATLAS é um problema, não só em termos de armazenagem, como também em fluxo de dados pelo sistema de aquisição. Tentando contornar este problema, algumas pesquisas foram realizadas na intenção de se encontrar um método de compactação de dados. Como, no TileCal, a maior parte das células não é sensibilizada em cada evento, poderíamos suprimir parte da informação de energia destas células, obtendo uma redução no volume de dados por evento.

Na Figura 4.9, temos o método estudado para se realizar a compactação de um grupo de amostras (9 amostras de 10 bits para cada célula). O método é realizado em duas fases, uma primeira de análise e a segunda de organização das amostras. Do conjunto de amostras, é descoberta a amostra com máxima e com a mínima

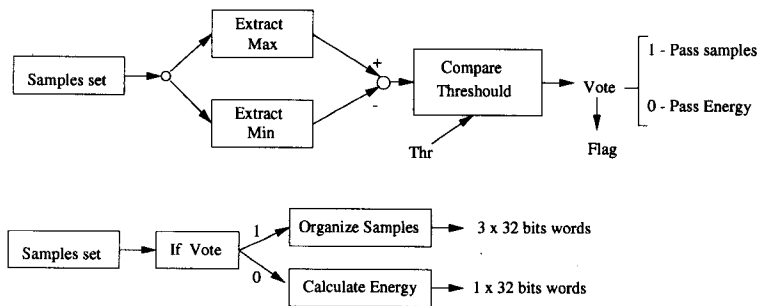


Figura 4.9: Esquema de compactação das amostras do TileCal.

amplitude. Subtrai-se este máximo do mínimo. O resultado é comparado com um patamar determinado experimentalmente, como descreveremos em seqüência. Se o resultado da subtração for maior que o patamar pré-definido, as amostras do sinal são todas passadas adiante para uma melhor análise *offline*, por se considerar que pode existir aí um sinal que seja de interesse e, portanto, possa merecer um tratamento mais preciso. Caso contrário, um cálculo simples do valor de energia e do tempo exato de colisão da partícula são passados adiante. O resultado deste processo de seleção é colocado numa palavra extra, no conjunto de dados, para possibilitar a reconstrução do sinal quando for realizada a análise *offline*.

Na segunda fase, dependendo do resultado da primeira fase, são organizadas no bloco de dados ou as amostras (o que nos leva a 3 palavras de 32 bits) ou o cálculo de energia e tempo de impacto (o que acarreta numa transmissão de 1 palavra de 32 bits). Isto resulta numa compressão que, no caso extremo, pode ser de até três vezes.

É necessário, entretanto, se buscar um método para estabelecimento do patamar de separação entre o sinal de pedestal (quando nenhuma partícula tocou a célula) ou um sinal físico. O caso mais grave é quando temos de separar eventos de pedestal de eventos gerados por múons. Esta partícula, como deixa pouquíssima energia no detector, pode ser mais facilmente confundida com o sinal de pedestal. Na Figura 4.10, temos a distribuição da subtração da amostra máxima pela mínima para sinal de múons (topo) e sinal de pedestal (embaixo). Como pode-se claramente ver, é possível estabelecer um limite em torno de 6 a 8 contagens de ADC para a separação, comprimindo eventos de pedestal e mantendo rica a informação nos eventos de múons.

Este último trabalho desenvolvido foi importante, novamente, para aplicação

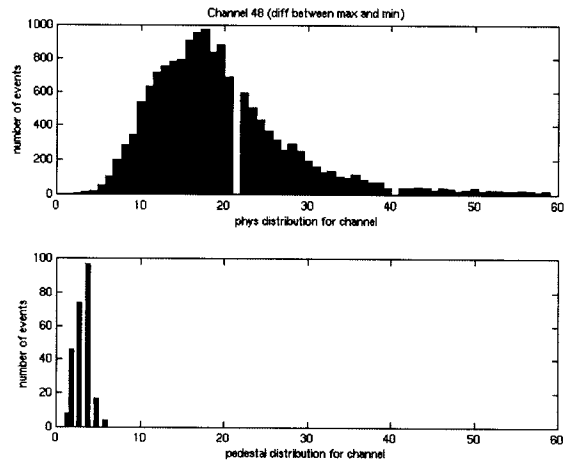


Figura 4.10: Distribuição do máximo menos mínimo para eventos de múons (topo) e de pedestal (embaixo).

neural. De novo, uma análise do código usado para cálculo de energia a partir das amostras nas células foi realizada, chegando-se a um código mais simples, gastando um tempo menor de processamento. Este conhecimento fez parte da reconstrução de energia necessária à aplicação neural.

Capítulo 5

Análise *Offline*

Na primeira parte deste capítulo descreveremos o processamento neural desenvolvido para análise *offline*, assim como a análise comparativa com outros métodos de análise. Na segunda parte, iremos estudar a influência de diferentes métodos de normalização da informação do calorímetro que alimenta o discriminador neural. Na seqüência, estudaremos a análise de parâmetros de treinamento da rede neural, buscando um conjunto de parâmetros que agilize o treinamento e garanta uma boa identificação dos eventos. A partir da quarta parte deste capítulo traremos algumas análises especiais que visam validar a metodologia estudada. Estudos com dados produzidos por um simulador do detector serão aí apresentados. Também discutiremos a aplicação do procedimento neural aos módulos da extensão do barril na quinta parte do capítulo e faremos uma análise da sensibilidade da rede neural treinada em certas condições (por exemplo, com um feixe de partículas a uma certa energia) quando aplicada em outra situação (por exemplo, outra energia diferente da usada no treino) na sexta parte. Abordaremos, finalmente, na sétima parte, um pequeno estudo sobre o valor obtido na saída da rede neural em função do nível de contaminação presente nas amostras.

5.1 Análise *Offline* Inicial

Nesta seção, estudaremos os resultados da análise inicial relativa ao sistema *offline*. Aplicaremos a informação de energia em cada célula do calorímetro, gravada em um arquivo após o teste com feixe, como entradas para o processamento neural.

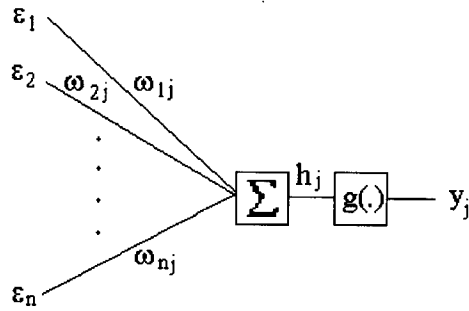


Figura 5.1: Esquema de um neurônio artificial.

Conforme veremos, será feito o estudo de duas separações diferentes, pión/múon e elétron/pión/múon. Posteriormente iremos ampliar a faixa de energia em que aplicamos nosso processamento.

5.1.1 Noções Iniciais de Redes Neurais

Redes neurais são estruturas que visam imitar o comportamento do cérebro humano em algumas tarefas específicas. A idéia básica vem da imitação do neurônio humano. Em termos matemáticos, a simulação do neurônio é como aparece na Figura 5.1.

Nesta figura temos três partes de processamento básicas. Cada informação, obtida de cada entrada do neurônio, é inicialmente multiplicada por um peso, w_{ij} , para um dado neurônio j . Os valores das entradas multiplicados pelos pesos são todos somados formando a saída h_j do neurônio. Finalmente, a saída do neurônio é aplicada a uma função não-linear, geralmente com um comportamento de saturação como a função sigmóide ou a tangente hiperbólica, produzindo a saída final do neurônio.

Camadas com vários neurônios artificiais são chamadas de redes neurais e variados estudos mostram que esta estrutura possui certas características que podem ser úteis para processamento de sinais, como, por exemplo, capacidade de reconhecimento de padrões e processamento não linear [56].

Os pesos da rede, conforme definimos acima, devem ser ajustados através de algum dos vários métodos de treinamento de rede neural conhecidos. Existem

métodos de treinamento supervisionados, ou seja, nos quais a rede deve ser instruída com relação à qual deve ser a saída a ser produzida para uma dada entrada. Existem também métodos não supervisionados, em que características baseadas em propriedades estatísticas do próprio conjunto de dados são utilizadas para definir a saída da rede. No nosso caso, como nosso problema é de classificação de padrões, escolhemos um método supervisionado para nosso estudo.

A rede é treinada por passos de treinamento. Cada passo, representa a apresentação, no caso de um método supervisionado, de um par entrada-saída para um treinamento dos pesos da rede. Para cada passo, há um fator que é utilizado para ponderar a correção máxima que cada peso pode ser submetido. Este fator é chamado de taxa de aprendizado (*lr - learning rate*).

Para que a rede possa se aperfeiçoar em torno do ponto ótimo para realizar sua tarefa, esta taxa de aprendizado vai sendo reduzida conforme se avança no treinamento, evitando assim oscilações muito grandes dos pesos da rede a partir de um ponto em que a mesma é considerada treinada. Esta redução se dá após um número fixo de passos de treinamento, que chamamos de época, e se faz através da multiplicação da taxa de aprendizado por um número menor do que um, chamado razão de queda da taxa de aprendizado (*ldr - learning decreasing rate*). Assim, a cada época (que pode consistir, por exemplo, de 100 passos de treinamento), a nova taxa de aprendizado (lr_n) é calculada da seguinte forma:

$$lr_n = lr * ldr \quad (5.1)$$

Após um ciclo de treinamento, que pode consistir de um certo número de épocas ou um certo número de passos de treinamento, um ciclo de teste é realizado. Este ciclo de teste consiste em apresentar cada uma das entradas à rede neural e avaliar sua saída, verificando alguma medida de acerto para o treinamento da rede.

5.1.2 Rede Neural : Estrutura e Treinamento

Os valores de energia depositados nas células do calorímetros (considerando sua dupla leitura) formam um vetor de 46 posições e serve de entrada à rede neural, que, então, possui 46 nós de entrada. Esta rede usa o método *backpropagation* para o treino de seus neurônios. Estes neurônios utilizam como função de ativação a

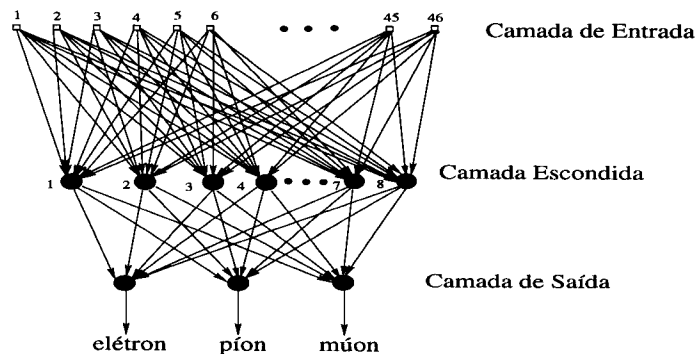


Figura 5.2: Esquema da rede neural usada no estudo.

tangente hiperbólica. O método de *backpropagation* exige, enquanto método de treinamento supervisionado, que sejam fornecidos pares entrada-saída para cada evento, com a etiquetagem das classes às quais os eventos pertencem. Entretanto, como já havíamos apontado, uma parte deste pares estarão necessariamente errados, quando existe contaminação do feixe, uma vez que as classes das partículas de contaminação serão substituídas pela classe do feixe de partículas originalmente selecionado. Conforme veremos logo adiante, esta situação de erro para o treinamento supervisionado pode ser superada pela extração adequada das características reais de deposição de energia no detector, o que evidenciará o erro na etiquetagem destas partículas de contaminação.

Na Figura 5.2, temos a arquitetura da rede neural usada no problema. Esta é uma rede completamente conectada (cada neurônio de uma camada está ligado a todos os neurônios das camadas anterior e posterior), de três camadas e com três saídas no caso de estarmos separando elétrons, píons e múons. Neste caso, cada neurônio de saída se encontra associado a uma dada classe de partículas. Quando o objetivo passa a ser a separação entre píons e múons, a rede é praticamente a mesma, exceto pela camada de saída, que só terá um neurônio.

A etiquetagem para a fase de treinamento é feita da seguinte forma, quando a rede de três neurônios de saída é utilizada. Um evento, proveniente de uma dada seleção de feixe de partículas, tem como saída desejada a classe do feixe selecionado. Assim, o vetor-alvo de saída terá o valor +1 na componente do neurônio associado àquela classe e -1 nas demais componentes. Como exemplo, para um evento proveniente de um feixe de elétrons (que pode ser, na verdade, um elétron ou mesmo um píon ou um múon de contaminação), o vetor de saída desejada será +1 para a

saída de elétrons e -1 para as outras duas. No caso de estarmos separando píons de múons, o único neurônio de saída será treinado para ser +1, no caso de píons, e -1 no de múons.

Com respeito à decisão quanto a classe da partícula incidente, na fase de operação da rede, a máxima probabilidade é utilizada quando elétrons, píons e múons devam ser separados (ou seja, a saída com maior valor identifica a classe da partícula). Quando a separação é entre píons e múons, o patamar em zero é utilizado para a determinação da classe da partícula, visando a média dos alvos para este caso.

A rede foi implementada através do pacote JetNet-2.0 [57]. Este pacote, escrito em Linguagem Fortran, foi devidamente interfaceado com as rotinas HBOOK, que codifica os dados experimentalmente adquiridos, de forma que os eventos pudessem ser acumulados em dois conjuntos distintos, de treino e de teste. Os eventos de teste não participaram da fase de treinamento, de forma a verificar-se a capacidade de generalização do classificador neural.

Os eventos são sorteados aleatoriamente na fase de treinamento, sorteando-se um par (ou triplo) de exemplares de cada classe a cada ciclo de treinamento. A normalização do vetor de entrada é feita pela soma da energia depositada em todas as células do detector evento a evento, ou seja, pela energia total absorvida. Assim, cada entrada da rede neural (En_i) é calculada da seguinte forma :

$$En_i = \frac{E_i}{E_t} = \frac{E_i}{\sum_{j=1}^{46} E_j} \quad (5.2)$$

Mais tarde, na Seção 5.2, estudaremos os efeitos desta normalização.

Após um dado número de ciclos de treinamento avaliam-se ambas as eficiências de classificação para os conjuntos de treinamento e teste. Quando há uma certa estabilização na evolução destas eficiências, consideramos a rede treinada. As eficiências de discriminação são medidas em relação à etiquetagem efetuada para cada conjunto de dados. Deste modo, como todos os eventos provenientes do feixe, quer sejam píons efetivos ou múons de contaminação, são etiquetados da mesma forma, como píons, uma baixa eficiência pode significar elevada taxa de contaminação.

Os parâmetros de treinamento da rede, tais como taxa de aprendizado, razão de decréscimo da taxa de aprendizado, época, valores iniciais dos pesos, número

de neurônios na camada escondida da rede, número de passos de treinamento [58], devem ser todos fornecidos às rotinas do JetNet antes do início do treino. Alguns destes valores foram determinados de forma empírica, outros terão um estudo mais extensivo na Seção 5.3.

O número de neurônios na camada escondida da rede neural, por exemplo, foi um destes parâmetros determinados empiricamente. Inicialmente, foram utilizados 20 neurônios na camada escondida, mas testes subseqüentes foram realizados mostrando que até com oito neurônios a rede neural era capaz de atingir os mesmos níveis de classificação, ganhando, contudo, em velocidade de processamento.

O calorímetro de cinco módulos chegou a ser utilizado no mesmo tipo de pesquisas que iremos aqui expor [59]. Dado o fato de que este protótipo não é utilizado na fase de calibração do detector por ser de uma geração anterior àquela do Módulo 0, nos restringiremos apenas aos resultados obtidos com o Módulo 0.

5.1.3 Separação Píon-múon

O primeiro caso que analisaremos e que servirá de base aos demais é a separação entre píons e múons com 20 GeV de energia nominal do feixe. Escolhemos esta faixa de energia por ser aquela em que a separação se torna mais difícil, dado o fato de que tanto píons quanto múons depositam, para esta energia nominal, quantidades semelhantes de energia no detector.

Treinamos a rede neural com um total de 54000 passos de treinamento divididos em 90 épocas de 600 passos. O conjunto de treino possuía 9914 eventos de píons (com uma certa porcentagem de múons de contaminação) e 4645 eventos de múons.

As eficiências de identificação (quantidade de eventos sendo reconhecidos como corretamente pertencendo a seu conjunto) para os conjuntos de treino e teste de ambas as partículas podem ser vistas na Tabela 5.1.

Os resultados de treino e teste são muito semelhantes, o que mostra a boa generalização da rede para os eventos que não foram apresentados na fase de treinamento. Este resultado, já era de certa forma esperado, já que a grande quantidade de eventos (mais de 4000 eventos para cada conjunto de dados de treinamento utilizado neste trabalho), garantia uma boa descrição estatística dos padrões envolvidos no

conjunto	tipo de feixe	
	píons	múons
treino	67,40%	96,84%
teste	67,06%	96,40%

Tabela 5.1: Eficiências de identificação para a separação pión-múon com energia de 20 GeV após 54000 passos de treinamento em 90 épocas de 600 passos de treinamento.

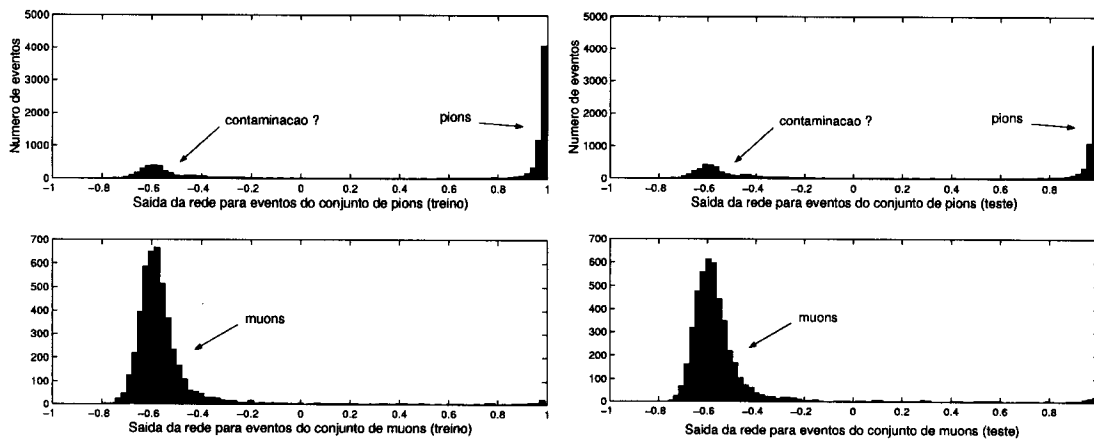


Figura 5.3: Histograma da saída da rede neural para os conjuntos de treino (esquerda) e de teste (direita), para píons (acima) e múons (abaixo).

problema de classificação. Vemos, entretanto, que apenas cerca de 67% dos eventos do feixe de píons foram identificados como pertencendo à classe de píons, ou seja, apenas 67% dos eventos vindos do conjunto de dados de píons seriam realmente píons, e não múons, segundo a rede neural. Como esta alta generalização se repete em todos os casos estudados, exploraremos apenas os resultados de treino. Assim, a não ser que expressamente mencionado, estamos nos referindo a resultados para o conjunto de treino da rede.

A Figura 5.3 apresenta as saídas da rede neural para os conjuntos de treino (esquerda) e teste (direita). Na parte superior, temos a saída para o conjunto de píons e, na parte inferior, a saída para múons. Vemos que, no conjunto de píons, existe um pico de eventos que se concentra na parte positiva da escala (logo, considerado, pela rede neural, como sendo píons). A este grupo de eventos bem numeroso, acrescenta-se outro de menor monta, no lado negativo da escala, que marcamos

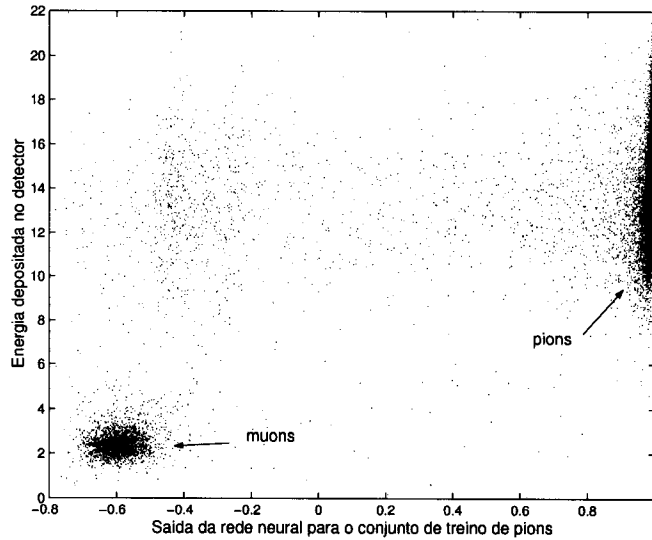


Figura 5.4: Correlação entre a saída da rede neural e a energia depositada no módulo.

como possível contaminação. No conjunto de múons, que supomos ser puro, vemos que praticamente todos os eventos se concentram na parte negativa da escala. É interessante notar que estes eventos estão exatamente na mesma posição dos eventos marcados como contaminação na parte superior, sugerindo o reconhecimento da contaminação. Se considerarmos, somente para comparação, no conjunto de píons os eventos que apresentam uma saída da rede menor do que $-0,2$, temos para os eventos (cerca de 3122 eventos) que passariam a ser considerados como contaminação uma média de $-0,5471$ para a distribuição da saída da rede para estes eventos, com um desvio de $0,012$ da média. No conjunto de múons, também para os eventos com saída da rede inferior a $-0,4$, temos a média de $-0,5738$ com um desvio de $0,0063$ (neste caso, temos 4445 eventos). A similaridade destes resultados reforça a indicação de que tais eventos sejam de fato múons que contaminam o conjunto de píons. Precisamos, entretanto, de alguma confirmação mais precisa deste fato, de modo a validar o processamento neural.

5.1.3.1 Validação do Processamento Neural

Na Figura 5.4, temos a correlação entre a saída da rede neural, para os eventos etiquetados como píons, e a energia total depositada no detector por estes mesmos eventos. Conforme havíamos indicado na Seção 3.2, um fator importante para a separação de píons (ou elétrons) de múons é a quantidade de energia depositada no

Método	Rede Neural		Método	Rede Neural	
	píons	múons		Clássico	píons
píons	67,00%	8,39%	píons	1,92%	1,74%
múons	0,40%	24,21%	múons	1,25%	95,09%

Tabela 5.2: Tabela de comparação do método neural com o método clássico (corte em energia) para o conjunto de píons (esquerda com 91,20% de concordância) e de múons (direita com 97,01% de concordância).

detector. Múons tipicamente não depositam mais do que alguns GeVs, enquanto píons (e elétrons) depositam valores próximos ao valor nominal do feixe. Assim, podemos ver na figura que a grande maioria dos eventos que obtiveram uma saída positiva na rede neural (logo, considerados como sendo píons pela rede) depositaram um valor de energia mais alto (na faixa dos 8 a 22 GeV) no detector. Já a maioria dos eventos que obtiveram um valor negativo da rede neural (logo considerados múons pela rede neural), foram os mesmos que depositaram muito pouca energia no detector. Isso parece demonstrar uma clara correlação entre o fator físico (nível de energia depositada) e o resultado da aplicação da técnica de redes neurais. Devemos salientar que a rede neural não recebe os valores absolutos de energia (por causa da normalização), sendo que seu acerto é unicamente baseado no perfil da cascata.

Precisamos ainda quantificar esta correlação e para tal vamos utilizar como fator de corte exatamente a deposição da energia no detector. Estabelecemos um valor de 7 GeV para separar entre píons e múons e isso nos leva à Tabela 5.2, que descreve a concordância entre a metodologia baseada em redes neurais e a metodologia baseada em cortes de energia, que chamaremos a partir de agora de clássica.

Como podemos ver os níveis de concordância são altos, sendo maiores do que 91% para o conjunto de píons e cerca de 97% no conjunto de múons. Vemos, entretanto, que, no conjunto de píons, uma parte dos eventos que depositaram mais de 7 GeV, e, portanto, foram considerados, pelo método clássico, como sendo píons, foram considerados múons pela rede neural. Esse problema pode ser grave, posto que a amostra purificada de píons, obtida através da saída da rede neural, pode sofrer de alguma tendência estatística (eventos de píons com alguma característica espe-

cial podem estar sendo identificados como múons). Analisaremos esta possibilidade melhor na Seção 5.2 mais adiante.

5.1.3.2 Relevância

Para podermos verificar que tipo de informação a rede neural está levando realmente em consideração para efetuar a separação entre as classes de partículas, foi realizada uma análise de relevância [60]. Esta análise consiste em se substituir, considerando-se todo o conjunto de treinamento ou teste, uma das entradas da rede neural de cada vez por sua média (no conjunto de dados) e, em seguida, avaliar o desvio quadrático da saída da rede com relação à saída original, evento a evento. A expressão da relevância é a seguinte para cada componente j do vetor de entrada :

$$relevancia(j) = \frac{1}{N} \sum_{i=1}^N |Out_{orig}(i) - Out_j(i)|^2 \quad (5.3)$$

onde N é o número de eventos, $Out_{orig}()$ representa as saídas originais e $Out_j()$ representa as saídas com o valor da entrada alterado para sua média no conjunto de dados (um valor fixo para todos os eventos). Assim, o que medimos é efetivamente o erro (efeito) acumulado quando desta substituição. Se, para uma dada componente do vetor de entrada, não houver um efeito muito significativo em termos de erro acumulado, a componente sob análise não é muito importante (relevante), segundo o classificador neural. Caso haja uma grande alteração na saída da rede, isto significa que a componente sob análise é relevante para a extração de características do conjunto de dados. Na Figura 5.5 temos esta análise para as 46 células do calorímetro.

Como podemos verificar na figura, apenas algumas informações são essenciais para a decisão da rede neural. Conforme havíamos comentado, quando foi discutido o método de compressão de dados, a maior parte das células do detector recebe apenas sinal de pedestal (ruído), logo o sinal fica retido apenas em algumas células. No caso, o feixe incidiu na terceira torre, a torre correspondente a um valor de $\eta = -0.25$, que compreende os canais de leitura 5, 6, 25, 26, 41 e 42, que são exatamente aquelas entradas que tiveram uma alta relevância para as saídas da rede. Assim, confirma-se que a rede é sensível ao processo físico da deposição de energia no detector pelas partículas que com ele interagem.

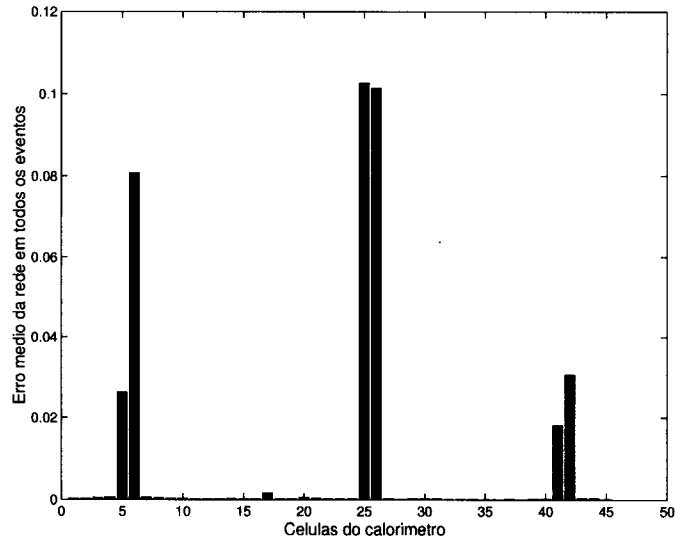


Figura 5.5: Relevâncias da rede neural para todas as células do calorímetro.

conjunto	feixe utilizado		
	elétrons	píons	múons
treino	62,80%	65,25%	96,75%
teste	64,04%	65,25%	96,10%

Tabela 5.3: Eficiências de acerto para a separação elétron-píon-múon a 20 GeV.

5.1.4 Separação Elétron-píon-múon

Nesta subseção, vamos nos dedicar ao estudo da separação de elétrons, píons e múons. Vamos, da mesma forma que na subseção anterior, considerar o caso mais complexo de 20 GeV, quando píons e elétrons são mais facilmente confundidos pelo fato de terem suas cascatas bem reduzidas a este baixo nível de energia.

Os conjuntos são, então, elétrons (com píons e múons de contaminação), píons (ainda com os múons de contaminação) e múons. Foram utilizados 9621 eventos de treino de elétrons, 9914 eventos de píons e 4645 eventos de múons. A mesma quantidade de eventos foi separada para teste. A rede foi treinada por 120.000 passos. A Tabela 5.3 resume os resultados de eficiência encontrados.

Como podemos novamente verificar, a rede neural demonstrou capacidade de generalização, dando resultados bastante semelhantes (no caso de píons temos mesmo resultados iguais) para os conjuntos de treino e teste. Mais uma vez, também,

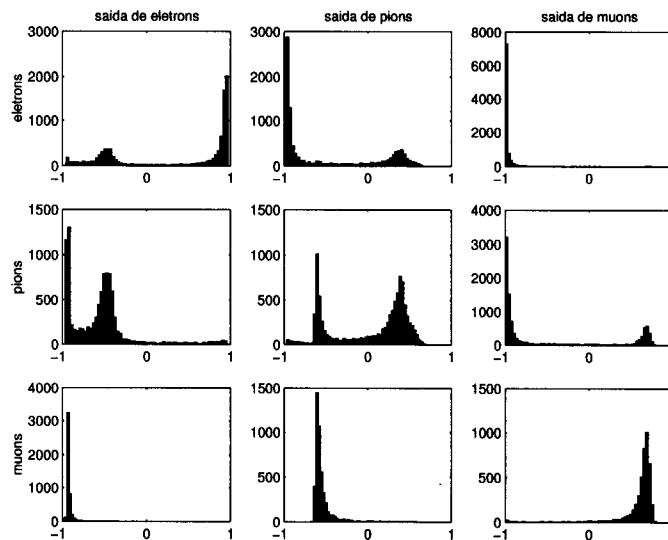


Figura 5.6: Saídas da rede neural (colunas da esquerda pra direita : elétrons, píons e múons) para os conjuntos de treino (linhas de cima pra baixo: elétrons, píons e múons).

verificamos que as eficiências para elétrons e píons não atingiram valores elevados, o que pode confirmar a existência de contaminação.

Para verificarmos esta possibilidade, vamos avaliar a saída da rede neural ao fim da fase de treinamento. Na Figura 5.6, temos uma matriz de histogramas, na qual as colunas são os três nós de saída da rede neural (saída de elétrons, píons e múons) e as linhas representam os três conjuntos de dados, ou seja, os eventos incidentes, num feixe de elétrons, píons e múon, de cima para baixo. Se todos os conjuntos de dados fossem livres de contaminações, deveríamos ter nos histogramas da diagonal principal desta matriz distribuições próximas a +1 (alvo de cada saída ativa da rede) e, nos histogramas fora da diagonal principal distribuições próximas a -1. Entretanto, apesar de podermos ver no primeiro histograma acima e à esquerda, que corresponde a saída de elétrons para eventos de um feixe de elétrons que, a maior parte dos eventos apresenta uma saída positiva, para uma parte dos eventos a saída se tornou negativa. Isto pode nos indicar a presença de contaminação no feixe de elétrons. No histograma a seguir, à direita deste primeiro, relativo à saída de píons para o mesmo conjunto de elétrons etiquetados, podemos verificar que, para a grande maioria dos eventos a rede apresentou o valor negativo esperado, enquanto que uma segunda distribuição pode ser observada na parte positiva do eixo, o que não seria

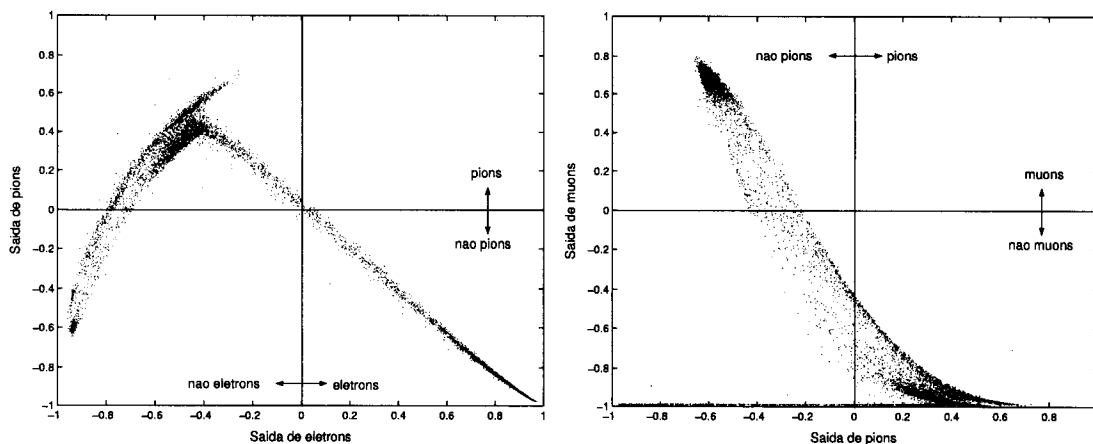


Figura 5.7: Correlação entre as saídas de elétrons e píons no conjunto de elétrons (esquerda) e entre as saídas de píons e múons para o conjunto de píons (direita).

esperado. Assim, este histograma confirma o indicativo de píons no conjunto de elétrons.

Também na segunda linha da matriz de histogramas, vemos um caso semelhante, posto que no histograma central (saída de píons para o feixe de píons), parte dos eventos produz uma distribuição no lado positivo do eixo (logo eventos considerados píons pela rede neural), enquanto que outra parte se distribui no lado negativo. Na saída de múons, à direita deste histograma, vemos que parte da saída da rede neural para eventos do conjunto de píons se deslocou para o lado positivo, logo, sinalizando a presença de múons neste conjunto, que só deveria conter eventos de píons. A saída de elétrons para o feixe de píons apresenta duas distribuições. Estas estão, entretanto, no semi-eixo negativo, mostrando que a saída de elétrons, ao contrário da de múons, não identifica eventos de sua própria classe, no caso, elétrons, no conjunto de píons.

Para verificarmos a consistência entre as saídas da rede neural, vejamos, a Figura 5.7. Podemos ver, na figura da esquerda, a correlação entre as saída de elétrons e a de píons para os eventos coletados do feixe de elétrons. A saída de elétrons apenas indica se os eventos são considerados elétrons (semi-eixo horizontal positivo) ou "não elétrons" (semi-eixo negativo). A saída de píons, analogamente, indica se os eventos foram considerados pela rede neural como sendo píons (parte superior no eixo vertical), ou "não píons" (parte inferior). Assim, podemos concluir que praticamente todos eventos que foram considerados elétrons pela saída de elétrons foram

também considerados "não píons" pela saída de píons. Da mesma forma, quase todos os eventos que foram considerados "não elétrons" pela saída da rede deste tipo, foram também considerados píons pela saída de píons. Alguns eventos vão para os setores negativos de ambas as saídas, mas não devemos esquecer que, nesta amostra, existe uma pequena quantidade de múons no feixe de elétrons, que provavelmente mereceram uma saída negativa das duas saídas em consideração. Outras correlações mostradas adiante provarão este fato.

Na figura da direita da Figura 5.7, temos a mesma situação para as saídas de píons e de múons, considerando-se o feixe de píons. Vemos que a maior parte dos eventos também se situam nos dois quadrantes feitos por píons e "não múons" e, também, "não píons" e múons, demonstrando, assim, consistência entre estas duas saídas da rede neural.

Podemos observar a consistência entre as saídas quando uma combinação das três saídas é realizada. Na Figura 5.8, temos esta combinação. Esta figura traz a subtração das saídas de elétrons menos a saída de píons (eixo horizontal) correlacionada com a saída de píons menos a de múons (eixo vertical) no feixe de elétrons, que, ao que parece, contém contaminação por píons e múons. Estas subtrações atuam como dois quesitos para se julgar cada evento do feixe em questão, sendo que cada quesito separa apenas duas classes de partículas, não servindo para julgar a terceira classe.

Assim, no semi-eixo horizontal positivo estão os eventos que são elétrons, ou não são píons ou múons (sobre esta última possibilidade a subtração da saída de elétrons menos a saída de píons não nos permite tirar nenhuma conclusão). O semi-eixo vertical positivo traz os eventos que podem ser elétrons, píons ou não múons. Sendo assim, a única possibilidade para o quadrante com os dois semi-eixos positivos é que os eventos aí sejam elétrons, já que um evento não pode ser, ao mesmo tempo, píon e não píon ou múon e não múon.

Usando esta mesma lógica, conseguimos definir as regiões marcadas na figura, onde devemos ter somente eventos de elétrons, píons e múons. Há uma quarta região em que não poderia, a princípio ter nenhum tipo de evento. Podemos ver que não existem situações ambíguas, ou seja, a rede claramente dispõe cada evento como pertencendo a uma das três classes de partículas em questão.

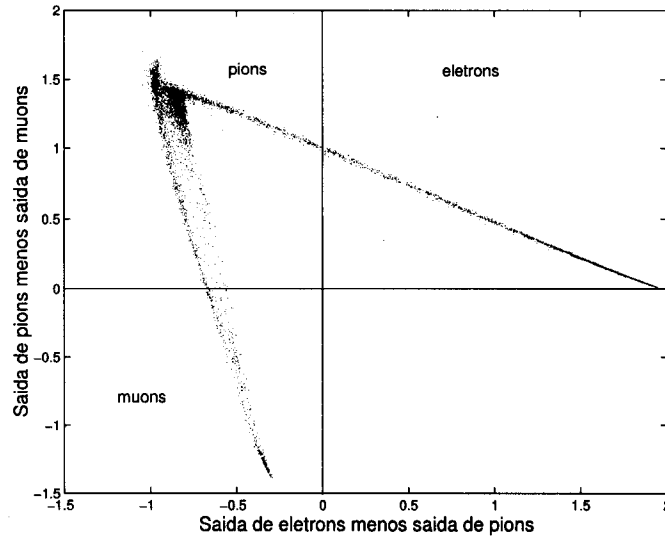


Figura 5.8: Correlação entre a subtração das saídas de elétrons menos a saída de píons e a subtração das saídas de píons menor a de múons para eventos do feixe de elétrons.

5.1.4.1 Validação do Processamento Neural

Para verificarmos a correlação dos resultados da rede com fatores clássicos, analisaremos a Figura 5.9. Na figura da esquerda, temos a correlação entre a subtração da saída de píons da saída de elétrons e a energia total depositada no detector, para eventos adquiridos do feixe de elétrons. Sempre lembrando que a rede não recebe esta informação de energia (por causa da normalização feita pela energia total absorvida), realce-se que, no lado positivo da subtração, temos todos os eventos considerados como sendo elétrons pela rede, enquanto que, no negativo, aqueles considerados píons ou múons. Podemos observar três concentrações de eventos. Duas concentrações se dão em alta energia : são elétrons e píons. A terceira menor seria a de múons com baixa energia.

No gráfico da direita na Figura 5.9, temos a mesma correlação considerando-se agora a subtração calculada entre a saída de píons e a de múons para o feixe de píons. Como podemos ver, a rede conseguiu dividir os eventos segundo sua energia. Os eventos do lado positivo da subtração (logo considerados pela rede como píons) têm alta energia, enquanto que os do lado negativo (logo considerados múons) têm baixa energia.

A partir destas figuras, podemos inferir que a rede parece estar fazendo o

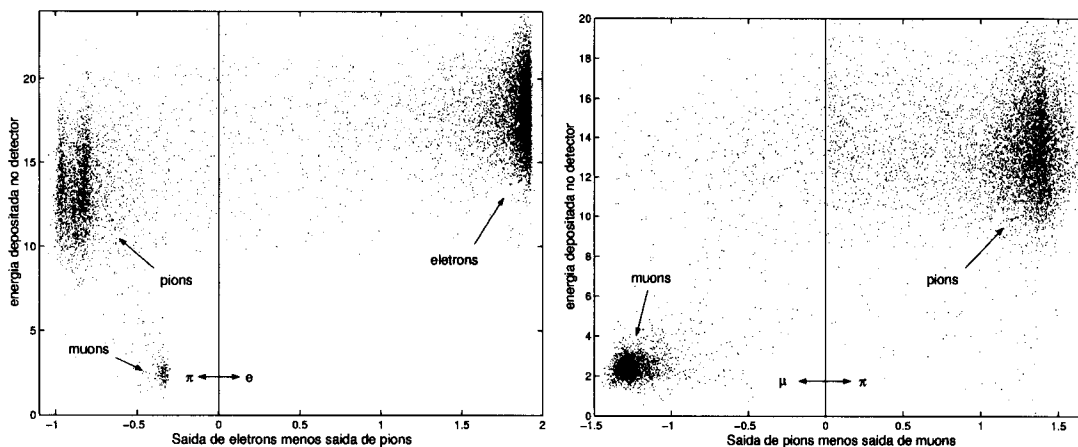


Figura 5.9: Correlação com a energia depositada no detector com a subtração calculada da saída de elétrons menos a de píons para eventos do feixe de elétrons (esquerda), e com a subtração calculada da saída de píons menos a de múons para o feixe de píons (direita).

seu trabalho de identificação de forma correta, particularmente no que tange à discriminação de eventos de múons contaminantes no conjunto de elétrons e píons. Precisamos, entretanto, verificar melhor se a discriminação entre elétrons e píons está sendo satisfatória. Conforme havíamos explicado no capítulo anterior, dois fatores podem ajudar nesta discriminação : a fração de energia depositada na primeira camada de células do detector e o valor do contador Cherenkov.

Na Figura 5.10, o gráfico da esquerda traz a correlação da subtração calculada da saída de elétrons menos a de píons, para os eventos do feixe de elétrons, com a fração (em %) da energia depositada na primeira camada de células do detector. Conforme podemos facilmente verificar, os eventos considerados pela rede como sendo elétrons (no semi-eixo horizontal positivo) depositaram mais de 75% de sua energia na primeira camada de células, o que é uma característica típica de cascatas de elétrons. Ao contrário, os eventos rejeitados como elétrons pela rede (no semi-eixo horizontal negativo) depositaram, em geral menos de 75% de sua energia na primeira camada de células do calorímetro. Vemos aqui uma forte correlação entre o que se esperava pela física do problema e o resultado da rede neural. Alguns eventos considerados como píons, entretanto, têm um alto valor depositado na primeira camada. Esta é uma característica dos eventos pertencente a esta classe (de píons) que dificulta a separação entre elétrons e píons.

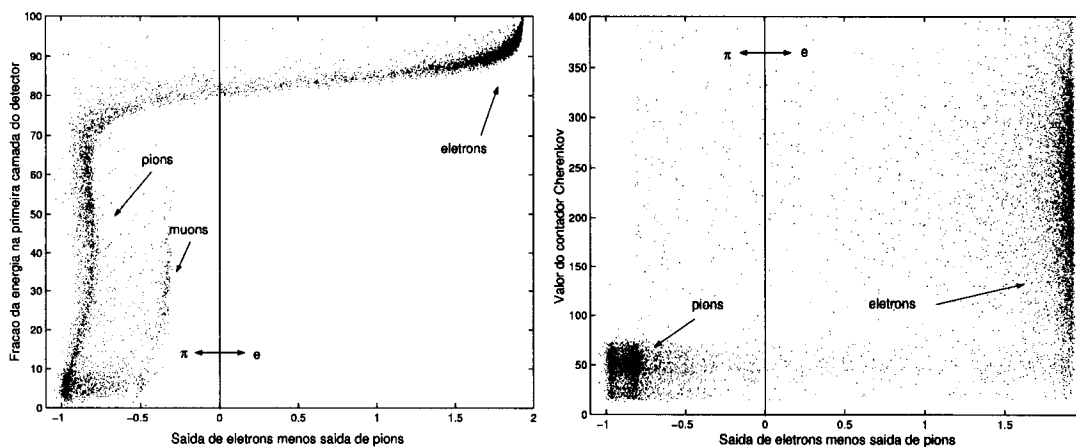


Figura 5.10: Correlação entre a saída de elétrons menos a de píons no conjunto de elétrons com a fração de energia depositada na primeira camada do detector (esquerda) e com o valor do contador Cherenkov(direita).

No gráfico da direita da Figura 5.10, temos a correlação desta mesma subtração com a contagem de Cherenkov. Como já havíamos explicado, os elétrons (que, por terem uma menor massa, têm mais velocidade a uma mesma energia), produzem uma contagem maior no contador Cherenkov. Vemos que os eventos com uma contagem menor do que 80 no eixo vertical (logo píons pelo contador Cherenkov) estão no semi-eixo horizontal negativo, enquanto que os eventos restantes, que têm uma contagem maior, estão no semi-eixo horizontal positivo, sendo considerados, portanto, elétrons pela rede neural. Vemos aqui também uma boa concordância entre a física do problema e a rede neural.

Precisamos, para completar, quantificar essas conclusões sobre a validação do processamento neural na tarefa de classificação de partículas. Para tal, visto que o sinal do contador Cherenkov estava disponível para os dados em análise, podemos estabelecer os seguintes cortes como forma de definir um método clássico de separação elétron-píon-múon :

- Se o evento depositar menos de 7 GeV no calorímetro, será considerado múon.
- Se o evento depositar mais de 7 GeV no calorímetro e tiver um valor no contador Cherenkov menor que 80, tal evento será considerado píon.
- Finalmente, se o evento depositar mais de 7 GeV no calorímetro e tiver mais do que 80 contagens no contador Cherenkov, este evento será considerado um

Método Clássico	Rede Neural			Método Clássico	Rede Neural		
	elétron	píon	múon		elétron	píon	múon
elétron	58,24%	1,85%	0,03%	elétron	0,24%	1,92%	0,19%
píon	4,56%	31,09%	2,17%	píon	5,87%	62,65%	4,52%
múon	0,0%	0,12%	1,93%	múon	0,07%	0,69%	23,86%

Método Clássico	Rede Neural		
	elétron	píon	múon
elétron	0,0%	1,01%	0,88%
píon	0,04%	1,14%	0,58%
múon	0,0%	1,05%	95,29%

Tabela 5.4: Correlação entre o método clássico e a rede neural para o feixe de elétrons (alto à esquerda), com 91,26% de concordância, píons (alto à direita) com 86,75% de concordância, e múons (embaixo) com 96,43% de concordância.

elétron.

Assim, utilizando esta metodologia para comparação com o processamento neural, as Tabelas 5.4 indicam um elevado nível de concordância. Nestas tabelas, cada coluna traz a fração dos eventos classificados pela rede neural para cada classe de partículas que é confrontada com as linhas, que trazem a mesma análise para a discriminação clássica, baseada em cortes de energia e Cherenkov. Assim, da tabela superior à esquerda, nota-se que 58,24% coletados do feixe de elétrons, por exemplo, foram considerados elétrons pelos dois métodos. Por sua vez, 1,85% destes eventos foram considerados píons pela rede neural e elétrons pelo método clássico, indicando discordância.

Conforme podemos ver, tanto as contaminações por píons e múons no conjunto de elétrons, como a contaminação de múons no conjunto de píons foram identificadas por ambos os métodos. Os níveis de concordância atingem valores maiores do que 86% no pior caso, que foi para o feixe de píons, o que pode ser considerado bastante satisfatório, considerando-se que a rede neural utiliza apenas a informação normalizada em energia do calorímetro. Note-se que os conjuntos de dados expe-

rimentais continuam níveis de contaminação bastante altos. No caso do feixe de elétrons a contaminação por múons não foi muito alta (cerca de 1,9%), mais a de píons foi elevada (cerca de 31%). No feixe de píons, a contaminação por múons também foi significativa (cerca de 24%).

Claramente, estes altos níveis de contaminação acarretam em uma caracterização errônea para muitos eventos de píons e múons (neste caso os que estão contaminando o feixe) durante a fase de treinamento. Graças às características marcantes dos perfis de deposição de energia, o método de *backpropagation* consegue realizar a separação de partículas, mesmo com toda essa informação de etiquetagem (alvo) errada. Podemos, no entanto, notar que as saídas da rede, para eventos de píons e múons, se desviam de seu alvo, o que é causado pelo fato de que alguns eventos de um mesmo tipo serem treinados ora com alvo em +1 e ora em -1. Na Figura 5.6, podemos ver que, na saída de píons para eventos do feixe de píons, a distribuição do lado positivo está longe do alvo desta saída (que seria +1). O mesmo se dá para a saída de múons, no conjunto de múons.

Além deste desvio, vemos que uma parcela de eventos do feixe de elétrons foi considerada, pela rede, como sendo elétrons (cerca de 4,5%) ou múons (mais do que 2%), enquanto que o método clássico os identificou como píons. No feixe de píons, a mesma situação se reproduz, eventos considerados píons pelo método clássico são considerados, pela rede, como sendo elétrons (quase 6%) e múons (cerca de 4,5%). Estes problemas serão revisitados na Seção 5.2, quando identificaremos as razões destes desvios e possíveis soluções. Além disso, podemos explorar a possível existência de um melhor conjunto de parâmetros de treino da rede, que possa conduzir a uma redução destes erros. Revisitaremos esta questão na Seção 5.3.

5.1.4.2 Análise de Relevância

Da mesma forma que anteriormente, podemos também pensar em termos das relevâncias. Neste caso, a relevância de cada componente foi extraída verificando-se o erro (agora vetorial) do vetor de saída da rede (com três componentes), quando se substitui, de cada vez, uma das entradas da rede por sua média. Sendo assim, a Figura 5.11 resume o resultado desta análise. Se compararmos estas relevâncias com as da separação pión-múon (veja a Figura 5.5), veremos que a rede "concentrou" muito

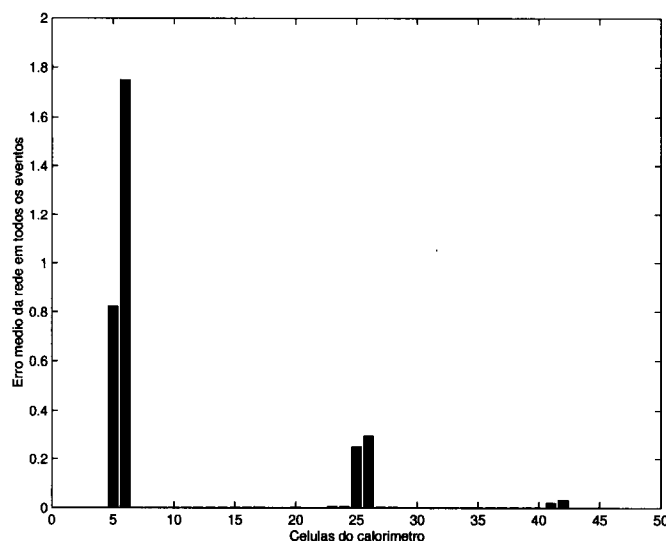


Figura 5.11: Relevâncias para a separação elétron-píon-múon a 20 GeV.

mais sua atenção nas células da primeira camada (5 e 6), claramente mais solicitadas nesta situação, dada a maior dificuldade de se distinguir entre elétrons e píons. Como podemos ver, ao se retirar a informação da célula 6, de maior relevância, a rede praticamente colapsa, com erro muito maior do que das outras componentes. Esta idéia está também em clara concordância com a física do problema.

5.1.4.3 Treino com Conjunto Descontaminados de Píons

Foi treinada também uma segunda rede, na qual foram utilizados os mesmos conjuntos de dados dos feixes de elétrons e múons, mas com o conjunto de dados adquiridos do feixe de píons purificado segundo o resultado da separação neural píon/múon. Desta forma, este seria, a princípio, um conjunto livre de contaminações. Com esta remoção da potencial contaminação de múons, sobraram, no conjunto de píons, 6665 eventos dos 9914 que existiam anteriormente. Os resultados da identificação de contaminação no feixe de elétrons estão sumarizados na Tabela 5.5. Vemos que, excetuando-se o conjunto de eventos do feixe de píons, para o qual não foi mais verificada a presença significativa de múons, nenhuma grande alteração pode ser identificada.

Vemos, entretanto, que os erros para o feixe de píons parecem ter crescido significativamente (de 5,87 para 7,34%, por exemplo, no caso de eventos identificados como elétrons pela rede neural e píons pelo método clássico), mas devemos nos

Método Clássico	Rede Neural			Método Clássico	Rede Neural		
	elétron	píon	múon		elétron	píon	múon
elétron	57,75%	2,34%	0,03%	elétron	0,32%	2,64%	0,05%
píon	3,98%	29,34%	4,50%	píon	7,34%	88,06%	1,01%
múon	0,01%	0,06%	1,99%	múon	0,09%	0,41%	0,11%

Método Clássico	Rede Neural		
	elétron	píon	múon
elétron	0,0%	0,90%	0,99%
píon	0,02%	0,90%	0,84%
múon	0,0%	0,78%	95,57%

Tabela 5.5: Correlação entre o método clássico e o processamento neural para elétrons (alto à esquerda), com 89,08% de concordância, píons (alto à direita), com 88,49% de concordância, e múons (embaixo) com 96,47%.

lembrar que estes percentuais agora estão sendo calculados sobre um número de eventos menor (6665 eventos). Assim, por exemplo, a discordância sobre os eventos considerados como píons pela rede neural mas que os métodos clássicos avaliaram como elétrons, poderia ter passado de 5,87 para $5,87 \times 9914 / 6665 = 8,73\%$, no caso de escalarmos o resultado com o número de eventos após a eliminação da potencial contaminação. Entretanto, este nível de discordância atingiu 7,34%, indicando uma melhora efetiva na performance da rede e uma redução na quantidade absoluta de eventos classificados de modo diverso ao realizado pela metodologia clássica.

5.1.5 Resultados para outras energias

Conforme havíamos mencionado anteriormente, não temos a nossa disposição o contador Cherenkov para todas as condições de aquisição. Isso ocorre porque a mistura de gases que compõe o sistema do contador Cherenkov tem de ser ajustada para cada diferente nível de energia, o que nem sempre pode ser feito durante o teste com feixes. Assim, para situações nas quais não havia este recurso de detecção, uma outra metodologia de separação entre elétrons e píons teve de ser estabelecida.

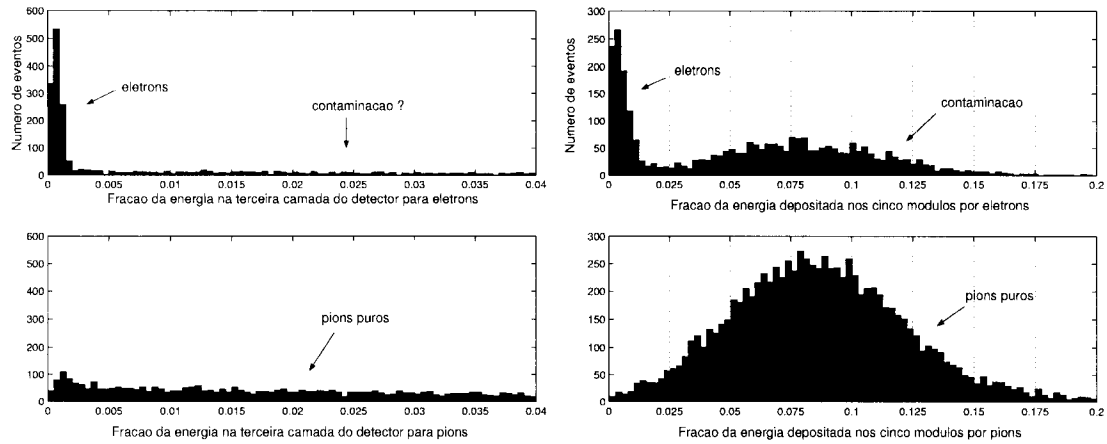


Figura 5.12: Fração da energia depositada na terceira camada (esquerda) e nos cinco módulos (direita) por um feixe de 100 GeV.

Para tal, contamos com a ajuda da fração de energia depositada na primeira camada (conforme discutimos no capítulo anterior), mas precisamos da ajuda de outras duas variáveis que discutiremos agora. Assim, podemos usar a informação na terceira camada e a fração de energia que vaza do Módulo 0 para os cinco módulos, quando estes estão instalados ao redor do detector.

A Figura 5.12 traz à esquerda a fração de energia depositada na terceira camada de células. No histograma superior, temos esta medida para o feixe de elétrons com contaminação por píons. Elétrons nunca chegam a depositar alguma energia nesta camada de células, por isso marcamos o bloco do conjunto de elétrons como sendo a distribuição próxima de zero. Ao contrário, píons, embora possam não vir a depositar nenhuma energia, podem deixar algum sinal nesta camada. Desta forma, podemos separar os dois conjuntos de partículas através de um corte nesta informação.

Na parte direita da Figura 5.12, temos a fração de energia depositada nos cinco módulos usados ao redor do detector. Estes módulos foram dispostos ao redor do detector exatamente para tentar conter os píons que podem atingir os cinco módulos. Assim, vemos que nos eventos adquiridos do feixe de elétrons (com contaminação por píons) temos um pico próximo a zero que, por saber que elétrons não tendem a vazar do detector, associamos a elétrons, e um outro grupo de eventos com distribuição semelhante aquela dos píons puros do histograma inferior. Por isso, associamos este segundo grupo com os píons contaminantes do conjunto de elétrons.

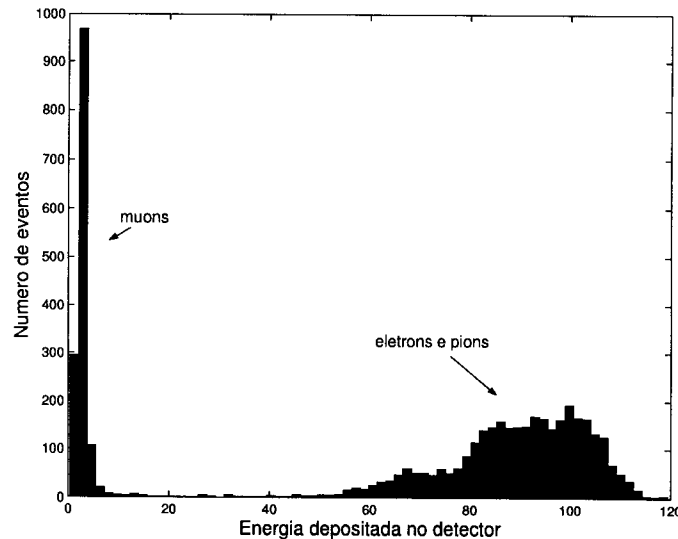


Figura 5.13: Energia depositada no calorímetro para eventos de 100 GeV de um feixe de elétrons.

Assim, podemos estabelecer uma metodologia alternativa, que usaremos enquanto método clássico para feixes de partículas com valores de energia acima de 20 GeV. É importante lembrar que esta metodologia falha em 20 GeV, posto que, em baixa energia, a cascata de píons é menor e mais curta, não atingindo a terceira camada e nem os cinco módulos. Isto impossibilita o uso destas variáveis para estabelecer cortes.

A metodologia para separar elétrons e píons de múons continua a mesma, qual seja, um corte no valor de energia depositada no detector. Na Figura 5.13, temos a energia depositada por eventos de um feixe de elétrons a 100 GeV no calorímetro. Como podemos ver, temos uma distribuição na faixa de mais alta energia, que é estabelecida pela deposição da energia de elétrons e píons. Na faixa de escala mais baixa, temos outra distribuição, que corresponde aos múons. Neste caso, podemos estabelecer o corte em 20 GeV como fator de separação entre estas duas distribuições. Para separar elétrons, píons e múons o método final é um pouco mais complexo :

- Se o evento depositou menos de 20 GeV no detector, ele será considerado múon.
- Se não for múon, e, depositar menos de 2,5% de sua energia nos cinco módulos ao mesmo tempo que ou deposita mais de 72% de sua energia na primeira camada ou menos de 1% na terceira camada de células, será considerado elétron.

E	π	μ	tot	E	e	π	μ	tot
20	91,20%	97,01%	94,06%	20	91,26%	86,75%	96,43%	91,39%
100	94,35%	92,79%	93,76%	100	93,31%	89,25%	93,41%	91,97%
180	93,98%	98,62%	96,27%	180	95,23%	94,92%	96,86%	95,67%

Tabela 5.6: Concordância para diferentes energias de feixe (em GeV) para o caso pión-múon (esquerda) e o caso elétron-pión-múon (direita).

Estes valores são estabelecidos a partir das Figuras 3.13 e Figura 5.12.

- Os eventos restantes serão considerados píons.

Este critério se relaciona profundamente com o que vínhamos discutindo com relação às variáveis de auxílio, buscando o máximo que nos pode fornecer a física do problema.

Assim sendo, temos nas Tabelas 5.6 os resultados de concordância entre esta metodologia clássica alternativa e o classificador neural para 100 e 180 GeV nas duas separações realizadas. No lado esquerdo, temos a discriminação entre píons e múons e, no lado direito, a discriminação entre elétrons, píons e múons.

Nas tabelas, os resultados de 20 GeV, embora tenham sido obtidos pelo método descrito na seção anterior (com uso do contador Cherenkov), foram incluídos a título de comparação. Vemos que o pior resultado para a separação pión/múon foi para 20 GeV (91,39%), enquanto que o menor resultado da separação elétron/pión/múon por conjunto foi o de píons a 20 GeV (86,75%).

Na próxima seção iremos discutir alternativas que visem melhorar bastante este desempenho.

5.2 Normalizações Alternativas

Algumas discussões com físicos do CERN, durante os testes com feixes, acabaram apontando possíveis melhorias para uma das eficiências do processamento neural no problema de separação de partículas. Ao se investigar tais questões, acabamos descobrindo que parte dos erros de classificação era advindo do processo de normalização. Assim, nesta seção, vamos discutir este problema e propor alterações

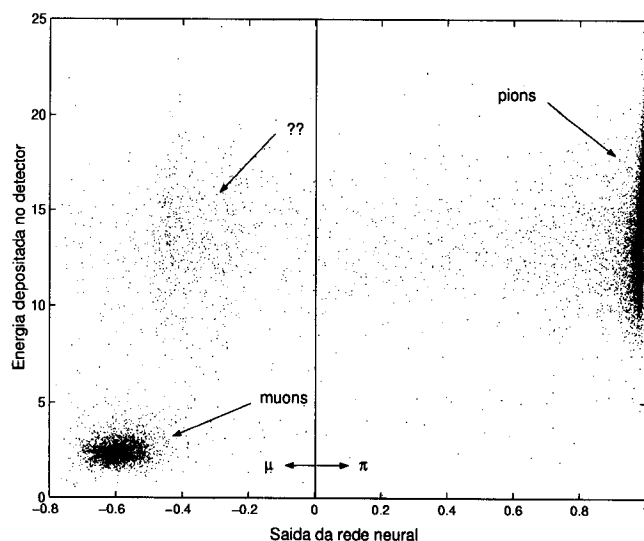


Figura 5.14: Correlação entre a saída da rede neural e a energia depositada no calorímetro por eventos de um feixe de píons (com contaminação) de 20 GeV. Notar o grupo com alta energia, mas considerado múon pela rede neural.

na metodologia de normalização dos dados.

5.2.1 Analisando a Normalização pela Energia Total

Quando fizemos a análise de comparação entre o método clássico e o processamento neural na separação pión-múon a 20 GeV, obtivemos uma eficiência elevada para o classificador neural, com um nível de concordância de 91,20%. Entretanto, observou-se uma discordância de 8,39% (como pode ser visto na Tabela 5.2) para os eventos que eram estranhamente associados pela rede neural a eventos de múons, embora depositassem mais de 7 GeV no detector. Precisamos investigar, então, dois pontos importantes. Primeiro, se o conjunto de eventos de contaminação identificados pela rede neural possui alguma tendência estatística, que possa vir a polarizar a amostra de píons que sobra quando usamos o critério da saída da rede neural positiva para definir o conjunto puro de píons. O segundo ponto consiste em investigar se, no caso da existência de alguma tendência na classificação neural, pode-se reduzir este erro de classificação, aumentando a eficiência da rede.

Para iniciarmos nossa abordagem, vejamos a Figura 5.14. Nesta figura, conforme havíamos mostrado anteriormente, temos a saída da rede neural correlacionada com a energia total para o feixe de píons de 20 GeV. Vemos, de forma clara, o

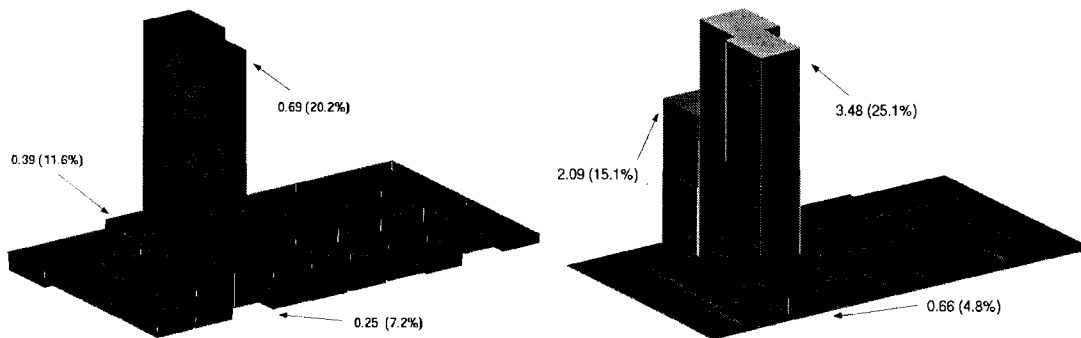


Figura 5.15: Perfil médio de deposição de energia para múons (esquerda) e para eventos com alta energia depositada e considerados múons pela rede neural (direita).

grupo de píons com elevada energia de deposição no calorímetro e uma resposta positiva da rede neural e o grupo de múons com resposta negativa da rede e baixo nível de deposição de energia. Vemos também um grupo de pontos, que marcamos com pontos de interrogação, que apresenta um comportamento inesperado. Este grupo é formado por eventos com um nível de energia maior que 7 GeV, os quais, portanto, não deveriam ser considerados múons, embora a rede neural os tenha considerado como tal. Estes são exatamente os eventos que formam o conjunto de discordância de 8,39% na Tabela 5.2, para o feixe de píons.

Para tentarmos elucidar esta questão, podemos verificar o perfil de deposição de energia destes eventos. Assim, na Figura 5.15 temos o perfil médio para múons (à esquerda) e o perfil médio referente a estes eventos que parecem ter sido mal identificados pela rede neural (à direita).

Podemos verificar que tais perfis são bastante similares, particularmente com relação a primeira e segunda camada de células do detector. Os níveis de energia em cada célula estão anotados, junto com o percentual que representam no total da energia depositada no detector. Assim, vemos que, na célula sensibilizada da primeira camada, o nível de energia é de apenas 0,25 GeV para o perfil médio de múon e 0,66 GeV, no perfil médio do conjunto potencialmente mal classificado. Vemos, entretanto, que os níveis percentuais são próximos, 4,8% neste e 7,2% naquele. Vemos também que algo semelhante acontece na célula sensibilizada da segunda camada,

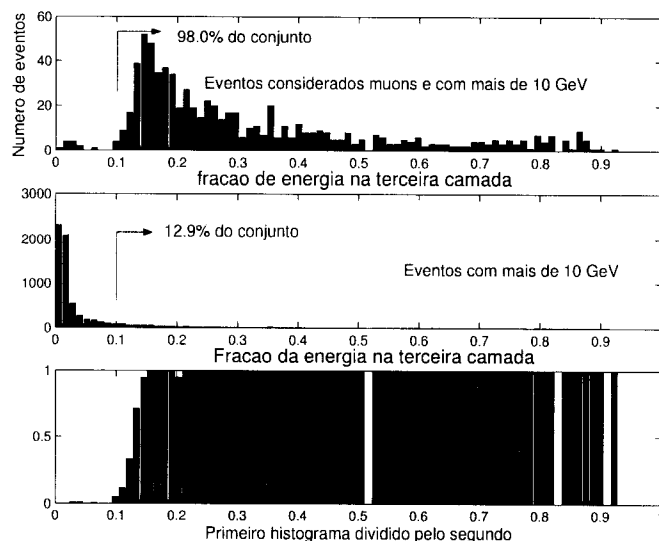


Figura 5.16: Fração de energia na terceira camada do calorímetro para eventos provenientes de um feixe de píons de 20 GeV e que depositaram mais de 10 GeV no detector. No topo, temos os eventos que foram considerados múons pela rede neural. No meio, todos os eventos com mais de 10 GeV. Abaixo temos a divisão do primeiro histograma pelo segundo.

onde apesar de termos 0,69 GeV no perfil de múons e 3,48 GeV no perfil de eventos com erro potencial, os percentuais são semelhantes (20,2% e 25,1%). Até mesmo na terceira camada, os percentuais também são aproximados (11,6% e 15,1%). Ora, a rede dada a normalização, não recebe valores absolutos de energia, e sim os valores percentuais, relacionados com os percentuais médios que estivemos comentando. Logo, isso significa que a rede pode estar confundindo eventos de píons que tenham perfis semelhantes, em termos percentuais, a múons. Assim, a solução para este problema pode estar numa mudança do método de normalização da excitação da rede neural.

Podemos acrescentar a informação longitudinal do perfil de deposição de energia, evento a evento, para um aprofundamento da análise. Na Figura 5.16, temos histogramas relativos à deposição da fração de energia na terceira camada de células do calorímetro para eventos que depositaram mais de 10 GeV no detector completo (assim, temos uma maior garantia de estarmos tratando com píons). Na primeira figura, temos dentre estes eventos aqueles que foram considerados como múons pela rede neural (conjunto de erro). Vemos que 98,0% dos eventos deste

conjunto depositou mais de 10% de sua energia na terceira camada. Vemos que dos eventos que depositaram mais de 10 GeV no calorímetro, que aparece no histograma do meio, apenas 12,9% dos eventos ocupam esta mesma faixa. Finalmente, vemos no histograma inferior a divisão do histograma superior pelo inferior, canal a canal. As barras de altura próxima a um (ou igual a um) representam canais do histograma aonde todos os eventos do histograma superior também estão no inferior. Da mesma forma, barras próximas a zero indicam que eventos do histograma inferior que não estão presentes naquele superior. Algumas barras estão com valor 0 na parte superior do histograma (acima de 0.5). Isto se deve a não existência de nenhum sinal tanto no histograma do meio quanto no superior (canal vazio do histograma). Assim, vemos que na faixa de 0 a 10% de deposição de energia na terceira camada, na qual a maior parte dos eventos se situa, não houve erro de classificação da rede neural. Entretanto, passando-se o patamar dos 10% praticamente todos os eventos são erroneamente classificados. Isto significa que os eventos do feixe de píons que tiverem depositado uma fração significativa da sua energia na última camada serão erroneamente considerados como múons. Ora, a partir deste erro, posto que existem eventos de píons com muita energia na terceira camada, há a introdução de uma tendência estatística na purificação do feixe de píons pela análise neural, com o conjunto purificado contendo menos eventos com a característica de deposição de energia elevada na última camada de células do calorímetro.

5.2.2 Normalização pela raiz quadrada

Assim sendo, pelos motivos expostos acima, uma possibilidade de modificar o processo de normalização seria introduzir a informação de energia (valores absolutos de energia) na entrada da rede neural. Isto iria facilitar a separação entre píons e múons. O único problema é que esta atitude irá introduzir a dependência em energia como consequência no processo de separação de partículas. A normalização baseada na energia total nos dá um classificador independente de energia, o que não é uma vantagem. Assim, se a dependência com a energia pode ser admitida, a eficiência na classificação neural pode ser ainda mais aprimorada.

Uma maneira de se normalizar a informação de entrada da rede neural classificadora e introduzir o valor de energia, evento a evento, no perfil de energia é usar

a raiz quadrada da energia total absorvida pelo calorímetro como fator de normalização. A idéia de se utilizar esta normalização se reside no fato de a raiz quadrada da energia se relacionar com a resolução na medida da própria energia através de [33]:

$$\frac{\delta}{\overline{E}} = \frac{a}{\sqrt{\overline{E}}} + b \quad (5.4)$$

Aonde a e b são constantes determinadas experimentalmente. Desta forma, para baixos valores de energia, a resolução (definido como o desvio da média dividido pela média) é dominada pelo fator relacionado com a raiz quadrada da energia e ponderado por a , indicando a variabilidade do próprio processo de produção da cascata na estrutura do detector (que em baixas energias pode variar muito de evento a evento). Quando a energia aumenta, a dispersão relativa ao processo passa a ser bem menor em relação ao nível de energia e o fator b , que se liga a dispersão relativa a ganho da célula fotomultiplicadora, quantidade de fóton-elétrons produzidos pela telha cintilante, entre outros fatores, passa a ser mais importante.

Desta feita, como a raiz quadrada da energia é de interesse para a física do problema, resolvemos aplicar a normalização que se segue. A energia En_i , em cada célula do detector será, com esta normalização:

$$En_i = \frac{E_i}{\sqrt{|E_t|}} = \frac{E_i}{\sqrt{|\sum_{j=1}^{46} E_j|}} \quad (5.5)$$

Com este novo enfoque de normalização, a noção de energia (na verdade, a raiz quadrada da energia) fica preservada em cada célula. Conseqüentemente, a separação entre píons e múons fica mais simples para a rede neural. Realizamos, assim, o treino da rede tanto para o caso pión/múon (feixe contaminado de píons) quanto para o elétron-pión-múon (feixe contaminado de elétrons).

Para 20 GeV a Tabela 5.7 traz o resultado de concordância com a metodologia clássica de análise (corte em energia).

Como podemos nitidamente verificar, para um feixe contaminado de píons, o procedimento proposto para a normalização do vetor de entrada da rede neural melhorou bastante a eficiência do sistema neural. A discordância anterior de 8,39 caiu para apenas 0,92%, o que, para o caso de 20 GeV, que, conforme apontamos, é um caso mais difícil, é uma melhora excelente. No total, temos uma melhora

Método	Rede Neural		Método	Rede Neural	
	Clássico	π		Clássico	π
π	74,47%	0,92%	π	2,30%	1,36%
μ	0,03%	24,58%	μ	0,04%	96,30%

Tabela 5.7: Concordância com a metodologia clássica quando utilizamos a normalização pela raiz quadrada da energia. Concordâncias para feixes de 20 GeV de píons (esquerda, total 99,05%) e múons (direita, total 98,60%).

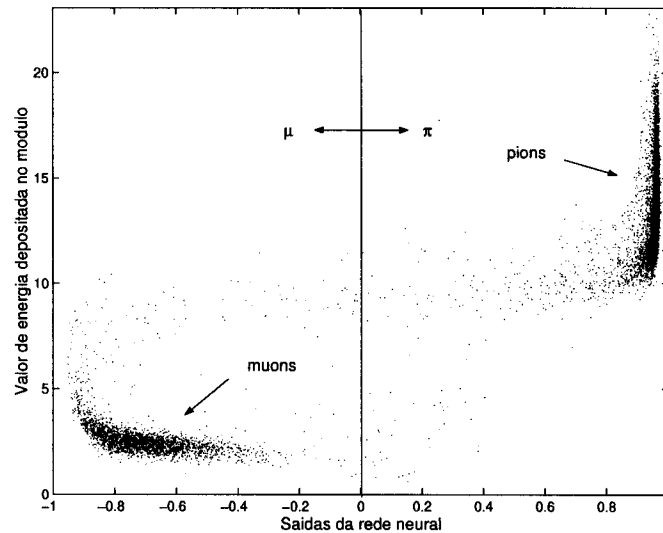


Figura 5.17: Correlação da saída da rede neural com a energia total, quando a raiz quadrada é usada no processo de normalização para eventos de píons a 20 GeV.

bastante expressiva nos níveis de concordância, de 91,20% para 99,05% para píons, e de 97,01% para 98,60% para múons. Esta significativa melhora elimina, praticamente, o problema da introdução de tendências nas amostras purificadas resultantes da análise neural.

Na Figura 5.17, temos a correlação da saída da rede com a energia total depositada no calorímetro, para esta nova situação. Como podemos ver, somente duas classes são definidas, com redução expressiva na incerteza de classificação. Assim, dois aglomerados são observados, um com baixa energia, no lado negativo da saída da rede, que reúne os múons de contaminação identificados no feixe de píons; outro, na faixa de alta energia, do lado positivo da saída da rede, reunindo os píons identificados.

Método Clássico	Rede Neural			Método Clássico	Rede Neural		
	e	π	μ		e	π	μ
e	59,24%	0,87%	0,01%	e	0,16%	2,06%	0,11%
π	3,75%	34,09%	0,38%	π	3,06%	69,15%	0,83%
μ	0,0%	0,17%	1,89%	μ	0,0%	0,59%	24,03%

Método Clássico	Rede Neural		
	e	π	μ
e	0,30%	0,73%	0,86%
π	0,22%	0,86%	0,69%
μ	0,02%	0,06%	96,25%

Tabela 5.8: Concordância entre os resultados da rede neural, com normalização pela raiz quadrada da energia total, e o método clássico, baseado no contador Cherenkov e no corte em energia. Acima, à esquerda, eventos adquiridos do feixe de elétrons, com concordância de 95,22%. No alto, à direita, eventos de um feixe de píons, com concordância de 93,34%, e, embaixo, eventos do feixe de múons com concordância de 97,41%. a energia dos feixes é de 20 GeV.

É claro que a normalização teve uma consequência positiva no funcionamento da rede na separação pión-múon, mas não é necessariamente óbvio que o caso elétron-pión-múon também melhore muito. Algum ganho de performance deve se verificar, posto que os múons de contaminação nos feixes de elétrons e de píons podem ser melhor identificados.

Na Tabela 5.8, apresentamos os resultados de concordância para os feixes de 20 GeV. As tabelas de concordância foram construídas usando-se para comparação o método clássico baseado no contador Cherenkov e corte em energia, conforme anteriormente exposto na normalização baseada na energia total (Veja a seção 5.1.3). Desta forma, podemos ver que os níveis de concordância subiram muito, principalmente para elétrons (de 91,26% para 95,22%) e píons (de 86,75% para 93,34%). Para múons, a concordância com o método clássico, sofreu também uma melhora de 96,43% para 97,41%.

E	π	μ	tot	E	e	π	μ	tot
20	99,05%	98,60%	98,82%	20	95,22%	93,34%	97,48%	95,33%
100	99,43%	98,88%	99,15%	100	94,19%	96,90%	99,68%	96,89%
180	99,85%	99,53%	99,69%	180	94,62%	95,10%	97,61%	95,77%

Tabela 5.9: Concordâncias entre o método clássico e o processamento neural para diferentes energias, quando se utiliza a normalização baseada em raiz quadrada da energia. Resultados para um feixe contaminado de píons (à esquerda) e para a contaminação no feixe de elétrons (à direita).

Como podemos ver, a maior parcela de discordâncias residuais nestas tabelas está na identificação de elétrons e píons. Este resultado era, de certa forma, esperado. Como havíamos já salientado, neste baixo nível de energia, com a granularidade do Módulo 0, é muito mais difícil separarem-se estes dois tipos de partículas, posto que suas cascatas de decaimento em energia são bastante similares.

Finalmente, temos que verificar o nível de acerto para diferentes energias. Na Tabela 5.9, apresentamos estas comparações para os feixes de píons e elétrons para 20, 100 e 180 GeV. Como podemos ver, utilizando o processo de normalização baseada na raiz quadrada da energia, todos os resultados totais de concordância foram melhores e praticamente todos os resultados de concordância para cada classe de partícula foram melhores que os expostos na Tabela 5.6, referentes a normalização pela energia total. Notamos também que o pior caso nas duas situações foi para a energia de 20 GeV, acumulando os piores resultados globais. Notamos, finalmente, que as eficiências na separação pión/múon são todas maiores que 98,6% e, no caso elétron/pión/múon, maiores que 93,3%, o que indica um alto nível de eficiência do sistema neural.

Para o caso 20 GeV as saídas da rede neural e as relevâncias por célula são mostradas na Figura 5.18, quando se projeta o classificador com a nova normalização. Como podemos ver, as saídas de elétrons e píons ficaram bem mais concentradas, demonstrando uma melhor definição dos conjuntos de dados. O conjunto de múons, nas saídas de píons e múons, ficou um pouco mais disperso.

Quanto às relevâncias, a importância das células da torre que contém a maior energia levam a uma relevância acentuada.

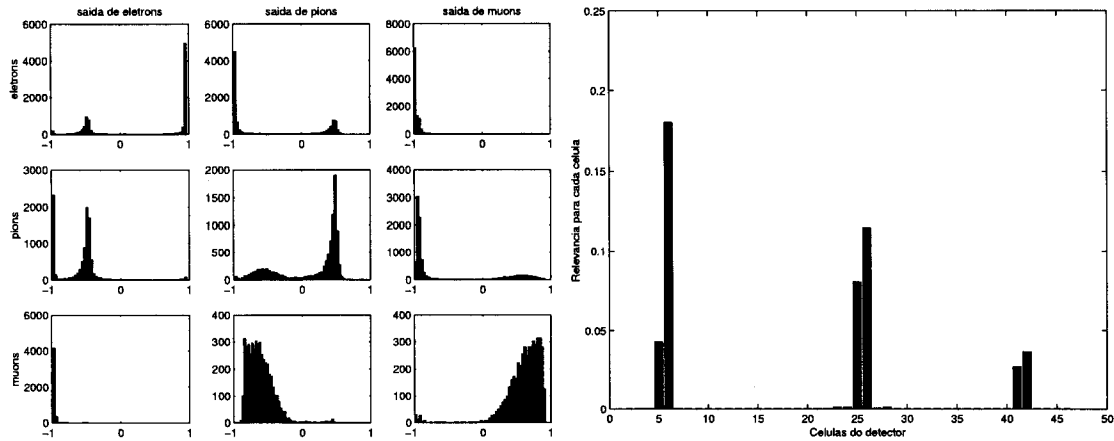


Figura 5.18: Saídas da rede para feixes de 20 GeV, usando a normalização pela raiz quadrada da energia (esquerda). Relevâncias para este classificador (direita).

Assim, temos dois métodos diferentes de normalizar a informação de entrada da rede neural classificadora:

1. De forma independente da energia da partícula incidente, que pode levar à introdução de uma tendência estatística na amostra.
2. De uma forma que depende da energia da partícula incidente, mas que se mostra ser mais precisa e não introduz tendência significativa.

5.2.3 Normalização pelo Valor Máximo

Podemos explorar ainda mais duas normalizações diferentes. A primeira que iremos estudar é a divisão das componentes do vetor de entrada pela máxima energia de deposição numa única célula. Assim, dividimos a energia de cada célula pela energia da célula com o máximo de deposição de energia. Tentamos com isso recuperar a independência com relação à energia.

Na Figura 5.19, temos, na parte esquerda, a correlação entre a saída da rede neural e a quantidade de energia depositada no detector, para o feixe de píons de 20 GeV. Como podemos ver, píons e múons se encontram bem definidos, apesar de haver um pequeno grupo de eventos com alta energia e identificados como múons pela rede neural. Isso significa que podemos estar sofrendo de novos problemas de introdução de tendência na amostra purificada de píons.

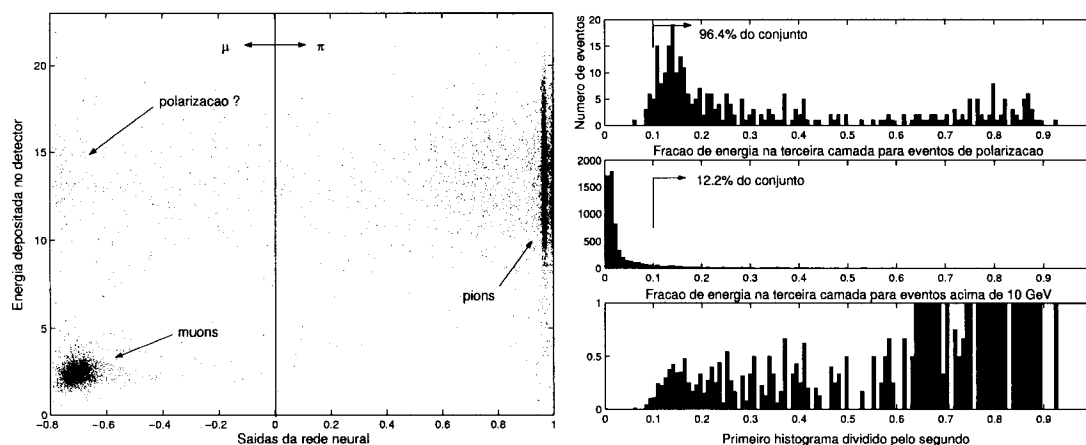


Figura 5.19: Correlação entre a saída da rede e a energia depositada no detector (esquerda), para a normalização pelo máximo de energia, considerando-se um feixe contaminado de píons. Análise da deposição de energia na terceira camada (direita), no histograma do topo temos os eventos que depositaram mais de 10 GeV e foram considerados múons pela rede neural. No histograma do meio, temos todos os eventos com mais 10 GeV e, abaixo, a divisão do histograma superior pelo do meio.

Nos histogramas à direita da Figura 5.19, analisamos a tendência estatística introduzida. O primeiro histograma traz exatamente os eventos do feixe píons que foram considerados múons, mesmo depositando mais de 10 GeV no calorímetro. O segundo histograma traz todos os eventos adquiridos do feixe de píons que deixaram mais de 10 GeV no calorímetro. O terceiro, é a divisão, canal a canal, do histograma superior pelo intermediário. No conjunto de eventos que apresenta um potencial erro de classificação, 96,4% dos eventos deixaram mais de 10% de sua energia na terceira camada, enquanto, num conjunto purificado pela análise clássica apenas 12,2% dos eventos depositaram mais de 10% de sua energia nesta camada de células do calorímetro. Vemos que, praticamente, nenhum evento na faixa de 0 a 10% de energia é observado neste conjunto com potencial erro de classificação. Assim, identificamos uma tendência estatística nesta amostra obtida pelo processamento neural, pois eventos que depositaram mais de 10% de sua energia na terceira camada são confundidos com múons.

Isso nos permite confirmar que métodos de normalização que venham a suprimir a informação de energia total produzem tal tendência, quando purificamos o feixe de píons. A título conclusivo desta terceira forma de normalização, apre-

E	π	μ	tot	E	e	π	μ	tot
20	95,45%	96,21%	95,83%	20	91,59%	86,07%	97,20%	91,51%
100	94,95%	97,55%	96,24%	100	93,64%	96,39%	99,37%	96,44%
180	95,84%	98,62%	97,22%	180	96,75%	93,58%	99,42%	96,55%

Tabela 5.10: Concordância com a análise clássica para o processamento neural com normalização pelo valor máximo da energia. Na esquerda, temos o caso pión/múon e, na direita, o caso elétron/pión/múon, considerando-se feixes de 20, 100 e 180 GeV.

sentamos, na Tabela 5.10, a concordância obtida para este enfoque de normalização com respeito à análise clássica, para diferentes energias de feixe de píons e elétrons. Podemos verificar que, praticamente, todos os resultados são um pouco melhores do que os da divisão pela soma da energia. Isso pode encontrar alguma explicação pelo fato de que a divisão pelo máximo da energia geralmente levar a números numa escala um pouco maior do que a divisão pela soma, logo, o evento é apresentado à rede com um maior detalhamento. Por outro lado, vemos que praticamente todos os resultados são inferiores à normalização pela raiz quadrada.

5.2.4 Normalização por Coordenada Extra

Veremos então o último tipo de normalização que exploraremos neste trabalho, que corresponde a uma normalização utilizando uma coordenada extra no vetor de entrada da rede neural [61]. A coordenada extra é adicionada com valor fixo +1. Assim, o vetor de entrada da rede passa a ter 47 componentes. Neste novo vetor de entrada é realizada uma normalização exatamente igual a que fizemos anteriormente, dividindo-se cada componente pela soma de todas as componentes do vetor, que agora atinge o valor $E_t + 1$. Desta forma, a última componente fica sendo $\frac{1}{E_t + 1}$, guardando em si, de alguma forma, a informação de energia.

Na parte esquerda da Figura 5.20 podemos ver a distribuição desta nova coordenada já normalizada, para um feixe de píons a 20 GeV. Como podemos ver, temos uma distribuição que pode ser utilizada pela rede para extrair a informação de energia discriminante para as classes de partículas e que produz uma faixa dinâmica

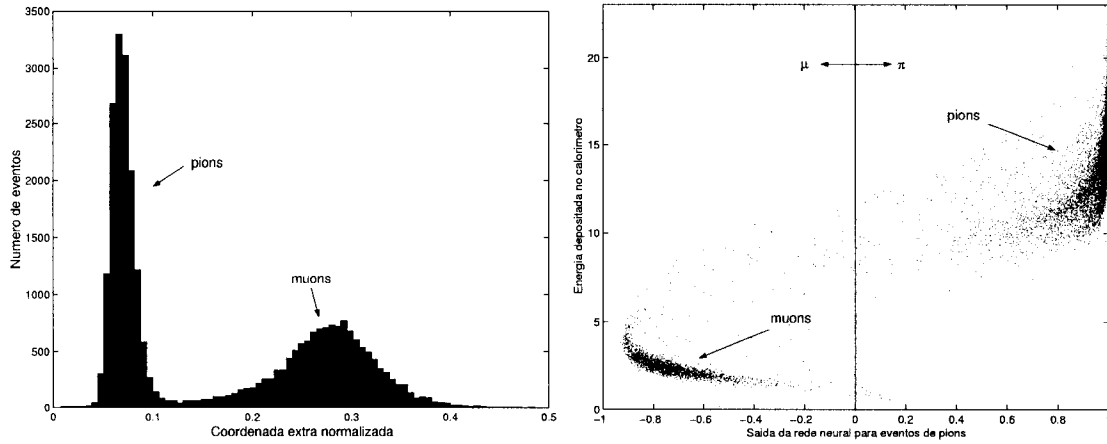


Figura 5.20: Coordenada extra já normalizada (esquerda) e saída da rede neural (direita) para um feixe de píons de 20 GeV, a qual é correlacionada com a energia total depositada no detector.

E	π	μ	tot	E	e	π	μ	tot
20	99,03%	98,56%	98,79%	20	93,63%	90,13%	97,46%	93,69%
100	99,81%	99,35%	99,58%	100	94,18%	96,58%	97,22%	95,98%
180	99,89%	99,27%	99,58%	180	97,02%	95,03%	99,67%	97,22%

Tabela 5.11: Resultados para a normalização pela coordenada extra para feixes contaminados de 20, 100 e 180 GeV. Na esquerda, temos o caso pión/múon e na direita o caso elétron/pión/múon.

compatível com a função de ativação da rede neural. Na parte direita da Figura 5.20, temos as saídas da rede neural correlacionadas com a energia total depositada no módulo, para o mesmo feixe de píons a 20 GeV. Desta forma, vemos que os conjuntos de píons e de múons estão bem separados, sem a existência de uma tendência significativa aparente.

Utilizando esta técnica de normalização, levantamos as tabelas de concordância com a metodologia clássica conforme vemos na Tabela 5.11. Como podemos ver, os resultados são em geral não muito diferentes dos resultados da normalização pela raiz quadrada, reforçando a idéia de que é importante fornecer à rede neural informação sobre a energia absoluta de deposição no calorímetro, evento a evento, como forma de se separar elétrons e píons de múons.

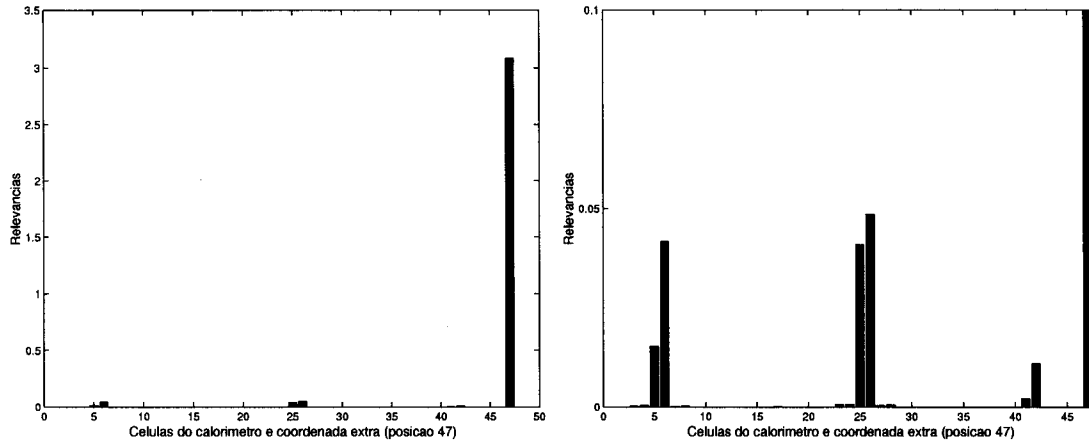


Figura 5.21: Relevâncias para as células do detector e coordenada extra na posição 47 (esquerda). Detalhe das relevâncias das células (direita).

Um aspecto interessante desta análise é o nível de relevância da coordenada extra, com relação à relevância das células do detector. Na Figura 5.21, temos a relevância calculada para o nosso projeto do classificador neural, considerando-se o feixe contaminado de píons de 20 GeV. Podemos ver que a coordenada extra é muitas vezes mais relevante (cerca de 19 vezes) do que as outras componentes de entrada para realizar a discriminação pión/múon. Na parte da direita, temos um detalhe das relevâncias das demais componentes do vetor de entrada, onde podemos ver que as células da torre de impacto das partículas incidentes ainda trazem alguma relevância para a discriminação.

Na Figura 5.22, mostramos as relevâncias para o caso elétron/píon/múon. Neste caso, a célula da primeira camada (posições 5 e 6) e a coordenada extra (posição 47) são claramente mais relevantes. Isto se dá porque é necessária a coordenada extra para se separar elétrons e píons de múons (baseando-se na informação de energia aí existente) e da célula da primeira camada, para que se possa realizar a discriminação entre elétrons e píons.

5.2.5 Sumário das Normalizações Estudadas

A título de conclusão deste estudo do impacto dos diferentes enfoques de normalização do vetor de entrada, um sumário dos resultados alcançados pode ser visto na Tabela 5.12. Na parte superior, são mostradas as eficiências para a se-

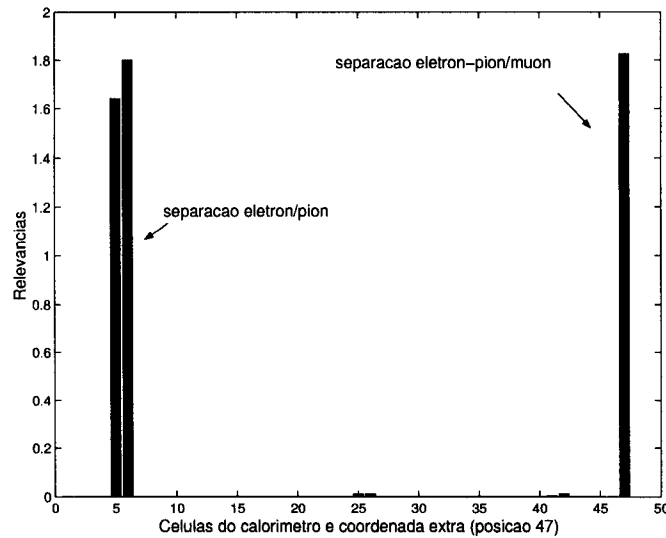


Figura 5.22: Relevâncias, considerando-se a coordenada extra na posição 47, na separação elétron/píon/múon.

paração píon/múon, enquanto que, na parte inferior, são mostradas as eficiências na separação elétron/píon/múon.

Como podemos inferir desta tabela, há em praticamente todos os casos um aumento da eficiência do método neural de identificação com o aumento da energia da partícula incidente. Também podemos notar que, em praticamente todos os casos, temos que os métodos dependentes da energia (raiz e coordenada extra) superaram em desempenho os métodos independentes da energia (soma e máximo).

5.3 Análise de Parâmetros

Durante nossa análise do comportamento da rede para variadas situações de normalização ou compactação dos dados vindos do calorímetro, sempre consideramos os mesmos parâmetros de treino da rede neural. No caso, três parâmetros são de particular interesse : a taxa de aprendizado (*learning rate* - lr), a queda da taxa de aprendizado (*learning decreasing rate* - ldr) e a época de treino.

A lr é a constante que controla a variação dos pesos para um passo de treino. Assim, se a correção de um peso da rede é calculada como sendo +1, apenas 0.2 será corrigido. Este mecanismo facilita a procura de um mínimo da função de erro da rede, evitando saltos em torno deste mínimo que poderiam levar a um erro maior.

método	20 GeV	100 GeV	180 GeV
E_t	94,06%	93,76%	96,27%
$\sqrt{ E_t }$	98,82%	99,15%	99,69%
$max E_i$	95,83%	96,24%	97,22%
coord. extra.	98,79%	99,58%	99,58%

método	20 GeV	100 GeV	180 GeV
E_t	91,39%	91,97%	95,67%
$\sqrt{ E_t }$	95,33%	96,89%	95,77%
$max E_i$	91,51%	96,44%	96,55%
coord. extra.	93,69%	95,98%	97,22%

Tabela 5.12: Desempenho dos métodos de normalização para a separação pión-múon nas várias energias (tabela superior) e para a separação elétron-pión-múon (tabela inferior), com respeito ao método clássico de análise.

A busca no espaço de pesos é mais suave e mais garantida temos com relação aos nossos resultados.

A ldr é usada como fator de redução da taxa de aprendizado. Assim, a cada época de treino (um certo número determinado de passos de treino), a taxa de aprendizado é multiplicada pela ldr, que é sempre menor ou igual a 1. A lr reduzida permite que a busca no espaço de pesos seja realizada com passos menores. Assim, quando nos aproximamos do mínimo, a procura com passos menores permite que uma maior aproximação ocorra.

Assim sendo, uma avaliação interessante de ser realizada seria descobrir se existem intervalos nos espaços de definição destes parâmetros que podem levar a uma maior ou menor acuidade da rede neural.

5.3.1 Estudo da taxa de aprendizado

Iniciaremos nossa análise pela taxa de aprendizado (lr). Assim, para a separação pión/múon a 20 GeV, realizamos 20 treinamentos com diferentes inicializações dos pesos da rede para cada valor dentro de uma faixa de valores da taxa

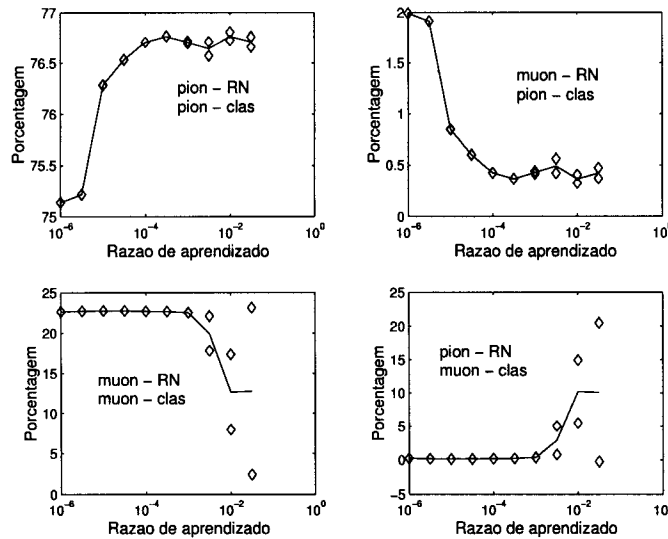


Figura 5.23: Acertos (lado esquerdo) e erros (direito) da rede neural (RN) em relação ao corte em energia (clas) em função da taxa de aprendizado.

de aprendizado. Para podermos não nos preocupar com a queda da taxa de aprendizado e a época, fixamos a queda da taxa de aprendizado em 1. Foram realizadas 90000 épocas de 200 passos (num total de 18000k passos de treino) visando se estabelecer um resultado no qual a rede não varie praticamente em torno do ponto de convergência.

Assim, na Figura 5.23, temos para o conjunto de píons, as concordâncias e discordâncias da rede neural em relação ao corte em energia (método clássico) em função do valor da taxa de aprendizado. No caso, a normalização pela raiz quadrada foi utilizada. Assim, o primeiro quadro no alto à esquerda traz a porcentagem de eventos que foram ditos píons pelos dois métodos e o quadro imediatamente abaixo deste os eventos ditos múons por ambos os métodos. No lado direito alto temos a porcentagem de eventos ditos múons pela rede neural e píons pelo método clássico. No quadro abaixo deste temos a porcentagem de eventos considerados píons pela rede neural e múons pelo método clássico.

Podemos ver claramente que o índice de acerto para múons (quadro inferior esquerdo) é bastante alto para pequenas razões de aprendizado. Com o aumento da taxa de aprendizado vemos que a partir do valor 10^{-3} a rede começa a ter uma grande queda no valor deste acerto. Vemos que o resultado de erro em que os eventos são ditos píons pela rede neural e múons pelo método clássico (quadro a

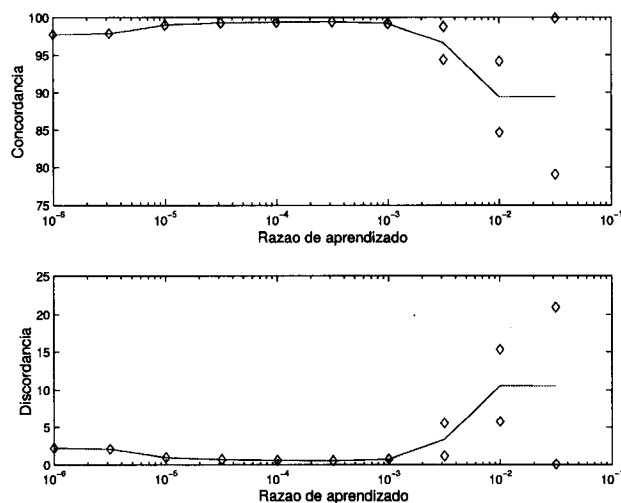


Figura 5.24: Concordância e discordância total em função da taxa de aprendizado para a normalização pela raiz quadrada, separação pión/múon a 20 GeV.

direita) começa a subir em correspondência com a queda de acerto.

Se pensarmos em termos de totalização de erros e acertos, temos a Figura 5.24. Na parte superior desta figura temos o índice de concordância que no valor da taxa de aprendizado em $10^{-3.5}$ atinge seu máximo igual a 99.4%. Na parte inferior temos o índice de discordância que, a partir da taxa de aprendizado em 10^{-3} começa a subir vertiginosamente, de acordo com o problema exposto acima.

A primeira conclusão importante que podemos tirar deste resultado é que um efetivo controle da taxa de aprendizado é fundamental para o funcionamento do sistema neural. Para podermos assumir este controle e, ao mesmo tempo, garantir uma alta velocidade no treino, iremos incluir adiante o efeito da queda da taxa de aprendizado e da época.

Precisamos, entretanto, encontrar uma causa para o problema de má classificação quando a taxa de aprendizado é alta. Para tanto, treinamos individualmente duas redes neurais, uma com a taxa de aprendizado em $10^{-3.5}$, ou seja, no ponto de maior concordância entre o método neural e o método clássico, e outra em 10^{-2} , ou seja, no ponto de maior discordância. O erro médio quadrático calculado sobre a saída da rede para os últimos 200 eventos de treino, ficou, com a taxa de aprendizado menor em 0.335. Quando treinamos a rede com a taxa de aprendizado maior, ficamos em 0.118. Sendo assim, a situação de maior discordância entre o método clássico e a rede neural leva a um erro médio quadrático menor do que a situação

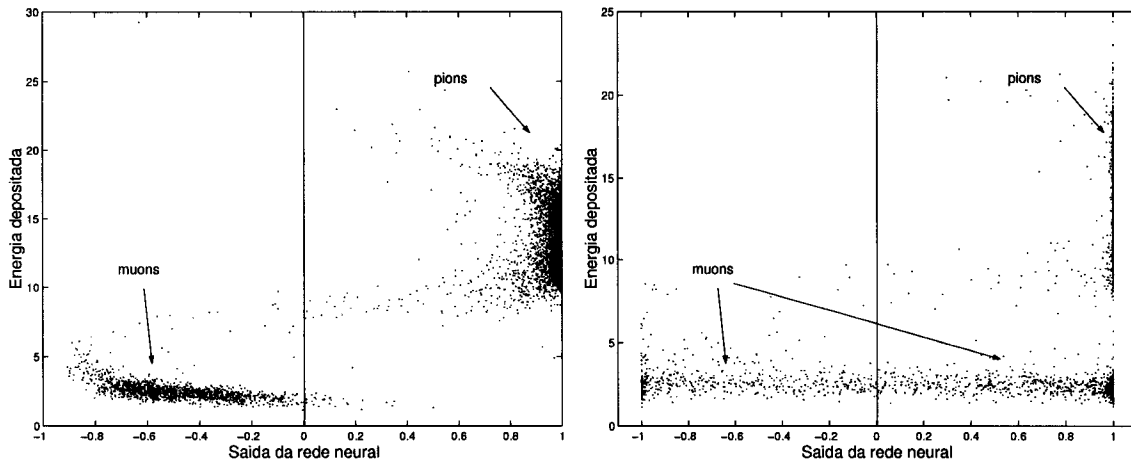


Figura 5.25: Correlação entre a saída da rede e a energia depositada no detector para $lr = 10^{-3.5}$ (esquerda) e $lr = 10^{-2}$ (direita). Notar dispersão da saída de múons neste segundo caso.

de maior concordância.

Na Figura 5.25, temos uma possível explicação para este fato. Esta figura traz para as rede treinadas com diferentes razões de aprendizado as correlações entre a energia depositada no detector e a saída da rede.

Na figura da esquerda, temos as correlações para a taxa de aprendizado em $10^{-3.5}$. Neste caso, podemos verificar claramente a boa correlação entre os eventos. Os eventos considerados píons pela rede neural (com saída positiva), depositam bastante energia no módulo. Por outro lado, os eventos com baixa energia são considerados múons pela rede neural (com saída negativa). Na figura da direita que leva em consideração os resultados quando usamos a taxa de aprendizado 10^{-2} , vemos uma situação completamente diferente. Neste caso, os eventos de múons foram completamente dispersados por todo os valores de saída da rede. Temos múons com saídas da rede positivas e saídas negativas.

A explicação para este erro da rede reside no fato de que o método de treino *backpropagation* tenta reduzir o erro médio global, incluindo aí o erro para múons. O problema é que, dada a presença de eventos de múon no conjunto de píons, muitos eventos de múons recebem o valor alvo de píons para treino da rede. A rede, ao ser deixada com uma maior liberdade (com uma taxa de aprendizado maior), consegue atingir um valor de erro menor dispersando os eventos de múons do que colocando-

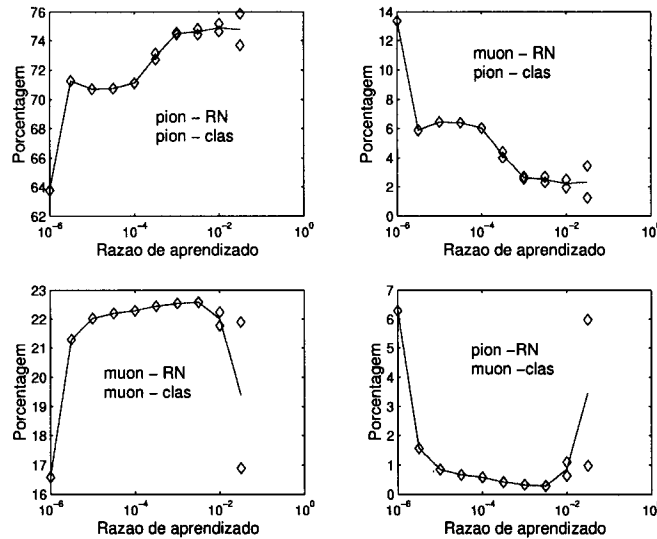


Figura 5.26: Acertos (lado esquerdo) e erros (direito) da rede neural (RN) em relação ao corte em energia (clas) em função da taxa de aprendizado. Caso com normalização pela energia total.

os apenas no semi-eixo negativo. Desta forma, a rede consegue em parte satisfazer a ambos os valores alvo recebidos pelos eventos de múons.

Podemos também verificar se o mesmo problema aparece para a rede treinada sem o uso da raiz quadrada como fator de normalização. Na Figura 5.26, temos as figuras de evolução das concordâncias e discordâncias em função da taxa de aprendizado. De novo, 20 diferentes treinamentos da rede foram realizados e cada ponto é a média destes treinamentos. Como podemos ver a mesma queda no acerto aparece, entretanto, apenas após um valor bem maior de taxa de aprendizado (cerca de 10^{-2}).

Podemos notar também que o erro em que eventos com alta energia, logo considerados píons pelo método clássico e múons pela rede neural (curva no alto à direita), nunca fica menor do que 2%, bastante acima do mínimo em torno de 0.5% obtido para o mesmo erro com a normalização pela raiz quadrada. Este é o erro que analisamos nas seções anteriores, relativo a eventos de múons que, por possuírem alguma característica especial (por exemplo, depositarem muita energia na terceira camada de células do detector) são erradamente confundidos com múons.

As curvas totalizadoras para esta situação (com normalização pela energia total) podem ser vistas na Figura 5.27.

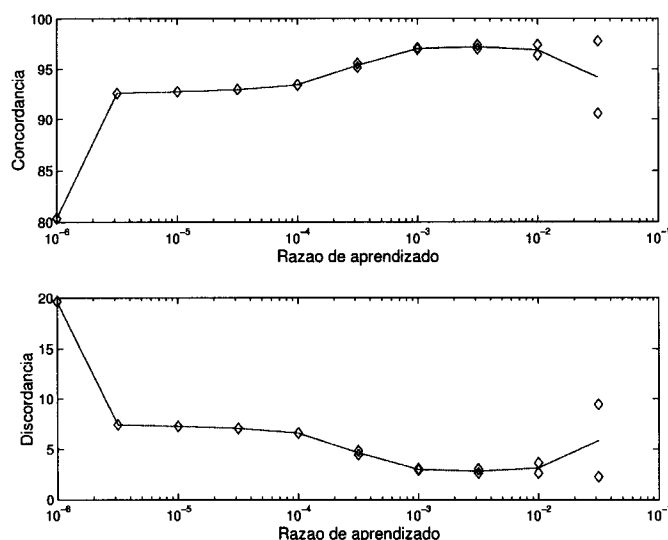


Figura 5.27: Curvas com concordância (superior) e discordância (inferior) entre o método clássico e o método neural em função da taxa de aprendizado.

Neste caso, vemos que o máximo da concordância se registra um pouco mais tarde, em torno de $10^{-2.5}$, e seu valor é de 97.2%, ou seja, bem abaixo do valor obtido para a situação com a normalização pela raiz quadrada (99.4%). Assim sendo, concluímos que, em ambos os casos, um controle da taxa de aprendizado através da definição de uma taxa de queda para esta taxa e da época de treino são importantes.

5.3.2 Controlando a taxa de aprendizado

Assim, dadas as razões expostas acima, parece-nos bastante importante controlar a evolução da taxa de aprendizado, visando atingirmos rapidamente um ponto de boa classificação e limitando o treino antes que o mesmo nos leve para problemas como o exposto acima com o conjunto de múons.

Devemos nos lembrar que os fatores queda na taxa de aprendizado e época estão bastante inter-relacionados, e, portanto, o ideal seria estudá-los ao mesmo tempo. Um fator de queda muito brusco (por exemplo 0.9) pode reduzir drasticamente a capacidade de treino da rede se a época for menor que, por exemplo, 200 ou 300 passos de treino. Se, por outro lado, a época for muito grande (maior que 3k passos de treino), uma rede que pode ser treinada com cerca de 60k passos não vai ter muito reduzida sua taxa de aprendizado.

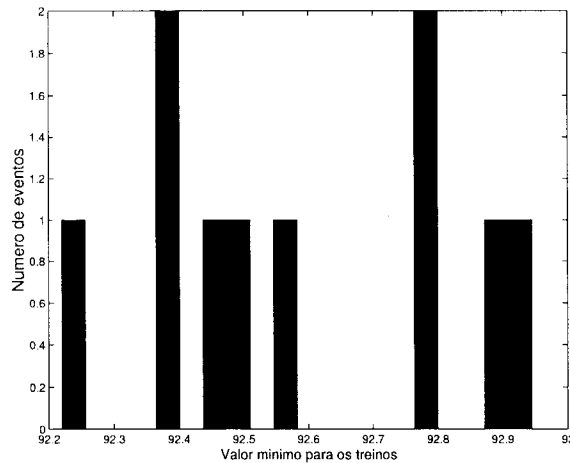


Figura 5.28: Histograma dos valores mínimos de concordância entre o método neural e o método clássico para 10 diferentes treinamentos com mesmos parâmetros (queda da taxa de aprendizado e época).

Assim, fizemos o estudo conjugado de ambos os fatores, pesquisando vários valores de época (23 diferentes épocas de 2 até mais de 6500) e para cada uma destas variamos também a queda na taxa de aprendizado de 0.985 até 0.999. Estes valores foram estabelecidos com base na experiência prática, e, como veremos adiante, são suficientes para nossas avaliações. Para cada ponto deste realizamos 10 treinamentos da rede com 60k passos de treino, parando de 1500 em 1500 passos para realizar um teste de performance. Neste teste todos os eventos são apresentados à rede e os resultados são comparados com o método clássico. Assim, após três quartos do treino (quando já podemos supor que exista alguma convergência da rede), descobrimos para cada treino o valor mínimo fornecido de acerto fornecido pela rede em todos os conjuntos de dados (média dos valores obtidos em cada conjunto).

Na Figura 5.28 temos, para 10 treinamentos da rede com os parâmetros fixos, um histograma dos valores mínimos obtidos nos últimos três quartos destes treinamentos. Utilizamos como medida da qualidade para cada ponto no espaço de parâmetros estudados o mínimo destes valores mínimos de cada treino. Fazemos isso com a intenção de garantir que, dentro dos limites da estatística fornecida por apenas 10 treinamentos, possamos garantir que após os três quartos do treino de uma rede neural com os parâmetros dados não encontramos nenhuma situação em que a rede tenha ficado abaixo do valor que assumimos como medida. Um pouco mais adiante,

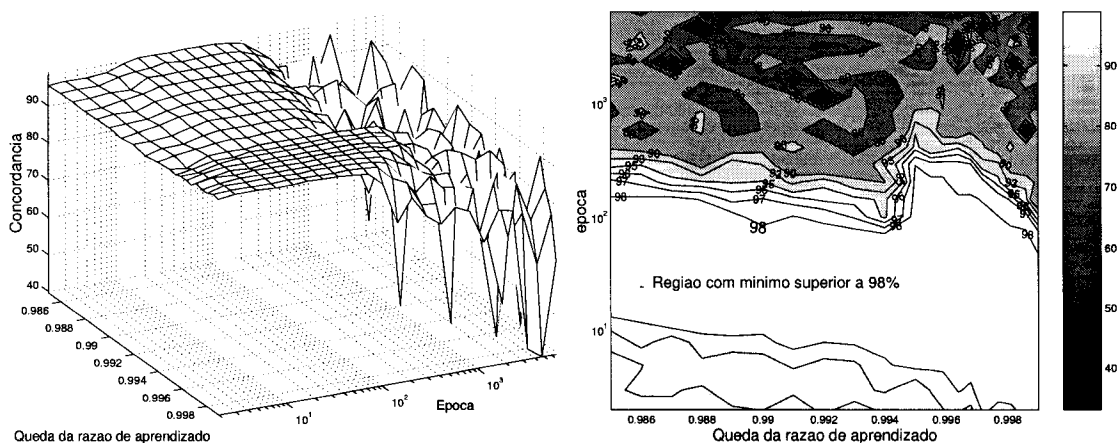


Figura 5.29: Variação da concordância da rede neural para separação píon/múon a 20 GeV em função da queda da taxa de aprendizado e da época (esquerda). Na direita temos as curvas de nível geradas com escala de tons de cinza.

investigaremos também o limite superior do acerto da rede, investigando o máximo dos valores máximos de concordância para cada ponto no espaço de parâmetros.

Assim, para a situação de 20 GeV, na separação píon/múon com o uso da raiz quadrada da energia total como fator de normalização para as entradas da rede neural, temos os gráficos de concordância que podem ser vistos na Figura 5.29. Nesta figura, podemos ver à esquerda a variação de nossa medida (mínimo dos mínimos dos 10 treinamentos para cada ponto) em função de nossos parâmetros de treino. Como podemos ver, existe uma faixa de valores em que o comportamento da rede é bem estável e acima de um valor bastante alto. Na figura da direita temos as curvas de nível geradas a partir da curva da esquerda. Podemos ver que para uma longa faixa de valores de nossos parâmetros em estudo a eficiência mínima encontrada esteve acima dos 98%, o que é um resultado bastante promissor.

Também podemos ver que um aumento da época parece levar a medidas bem menores. Isto se deve ao fato de que a rede com uma época bastante grande tende a reduzir pouco a sua taxa de aprendizado. Como o número de passos de treino é fixo, com, por exemplo, uma época definida em 100 passos de treino, são realizadas 600 épocas, durante o treinamento, o que reduz bastante a taxa de aprendizado. Já com uma época definida com 1000 passos de treinamento, apenas 60 épocas serão realizadas, assim, a redução na taxa de aprendizado é bem inferior, o que leva a uma oscilação bastante elevada no final do treino entre os mínimos locais da função

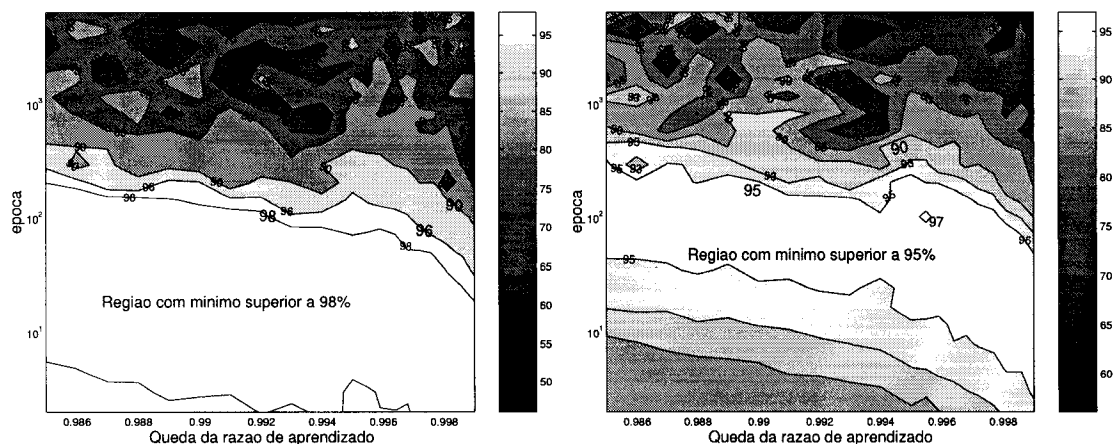


Figura 5.30: Concordância mínima nos três conjuntos de dados estudados combinadas para a rede com normalização pela raiz quadrada da energia total (esquerda) e pela energia no detector (direita).

de erro, e não conseguindo convergir para um bom valor e lá se estabilizar.

Para podermos avaliar melhor o procedimento neural realizado, resolvemos combinar para cada caso estudado as informações para 20 GeV, 100 GeV e 180 GeV. A forma para realizarmos esta combinação foi, para cada ponto considerado, utilizarmos a média geométrica dos pontos nas três energias. A intenção era, neste caso, rejeitar pontos que tivessem uma medida de qualidade baixa mesmo em apenas um dos casos. Assim, os resultados nos dariam a garantia de obter o valor pesquisado em qualquer situação.

A Figura 5.30 traz para os dois tipos de normalização estudados os resultados mínimos de concordância obtidos quando comparamos o método neural e o método clássico. Na figura da esquerda temos os resultados considerados quando a normalização pela raiz quadrada da energia é aplicada e na da direita quando aplicamos apenas a energia.

Como podemos ver, nos dois casos temos uma faixa mais ou menos central aonde os maiores valores são obtidos. Para épocas muito grandes, problemas como o que estudamos na subseção anterior e problemas de instabilidade são verificados, fazendo que os resultados de concordância rapidamente se degenerem. Na Tabela 5.13 temos a avaliação de quantos pontos no espaço de parâmetros receberam uma medida superior a 95% de concordância.

Podemos ver que os resultados para normalização pela raiz quadrada nos le-

medida	Norm. raiz	Norm. soma
# pontos	228	130
%	49,57	28,26

Tabela 5.13: Número de pontos no espaço de parâmetros e porcentagem com relação ao total (460 pontos) dos pontos que atingiram mais do que 95% de eficiência.

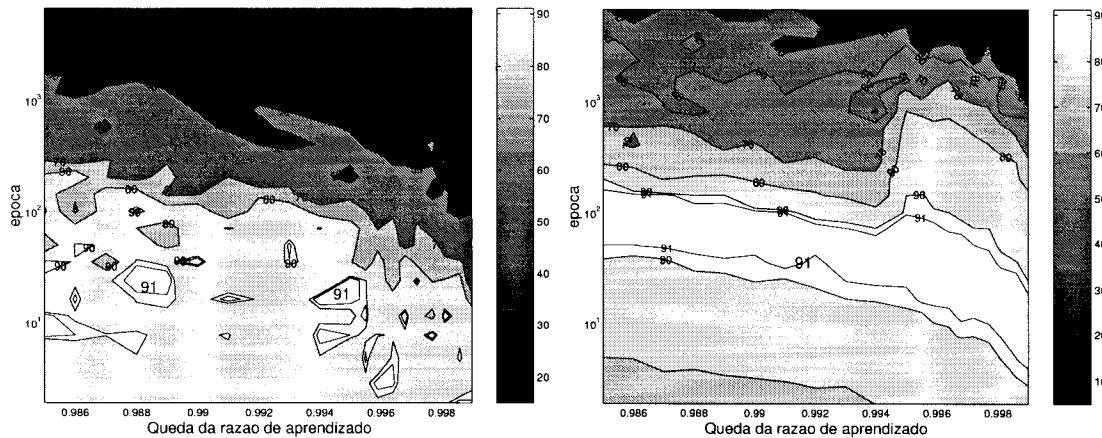


Figura 5.31: Concordância mínima nos três conjuntos de dados estudados combinadas para a rede com normalização pela raiz quadrada da energia total (esquerda) e pela energia no detector (direita). Separação elétron/píon/múon.

vam a quase o dobro de pontos acima de 95% de concordância do que a normalização pela energia simplesmente.

Da mesma forma que fizemos a análise para a separação píon/múon, fizemos também para a separação elétrons/píon/múon combinando os resultados para as três energias (20, 100 e 180 GeV) que dispomos. A Figura 5.31 traz os resultados destas combinações para a normalização pela raiz quadrada (esquerda) e pela energia (direita).

Como podemos claramente identificar, neste caso a normalização pela raiz quadrada produziu apenas algumas pequenas regiões aonde temos resultados acima dos 91%, enquanto a normalização pela soma mantém uma faixa acima deste mesmo valor. Isto nos leva a concluir que o método baseado em raiz quadrada sofre mais dos problemas apontados na sub-seção anterior. Isto era de certa forma esperado, posto que como o método separa bem os múons de elétrons e píons, evidencia também o

medida	acima de 90%		acima de 91%	
	Norm. raiz	Norm. soma	Norm. raiz	Norm. soma
# pontos	43	110	26	75
%	9,35	23,91	5,65	16,30

Tabela 5.14: Número de pontos no espaço de parâmetros e porcentagem com relação ao total (460 pontos) dos pontos que atingiram mais do que 90% e 91% de eficiência. Separação elétron/píon/múon.

erro de se incluir os múons nos conjuntos destas duas partículas, fazendo com que a rede mais rapidamente convirja para a situação de má classificação de múons. Na normalização pela soma, há de certa forma o benefício da dúvida sobre estes eventos, o que diminui a tendência a se atingir o estado de má classificação de múons.

A Tabela 5.14 traz a quantidade de pontos que estão acima de dois valores diferentes (90 e 91% de concordância entre método clássico e método neural).

Como podemos ver, em acordo com a informação provinda das figuras, fica claro que temos mais pontos acima de 90 e 91% no caso da normalização pela soma das energias em todas as células do detector. Assim, ficamos na dúvida sobre se devemos ou não utilizar o método neural com normalização pela raiz quadrada, já que o mesmo garante um valor relativamente mais baixo de acerto. Uma possível justificativa para mantermos o estudo do método baseado na normalização pela raiz quadrada da energia total pode estar no estudo dos máximos.

Neste estudo, buscamos avaliar o máximo de acerto que a rede forneceu em algum ponto de seu treino, e usamos este valor como medida do treino. Como medida de qualidade para o ponto no espaço de parâmetros usamos o máximo dos máximos obtidos nos 10 treinamentos para aquele ponto. Sendo assim, temos na Figura 5.32 as medidas do máximo para ambos os métodos de normalização.

Como podemos ver, para a figura da normalização pela raiz quadrada (esquerda), conseguimos manter uma eficiência superior a 96% por uma longa faixa, o que não se dá com a normalização pela energia total (direita) aonde valores acima deste patamar só aparecem numa estreita faixa no canto superior direito.

Na Tabela 5.15 temos a avaliação da porcentagem ocupada por estas faixas. Como podemos ver a normalização pela raiz quadrada leva a valores máximos por

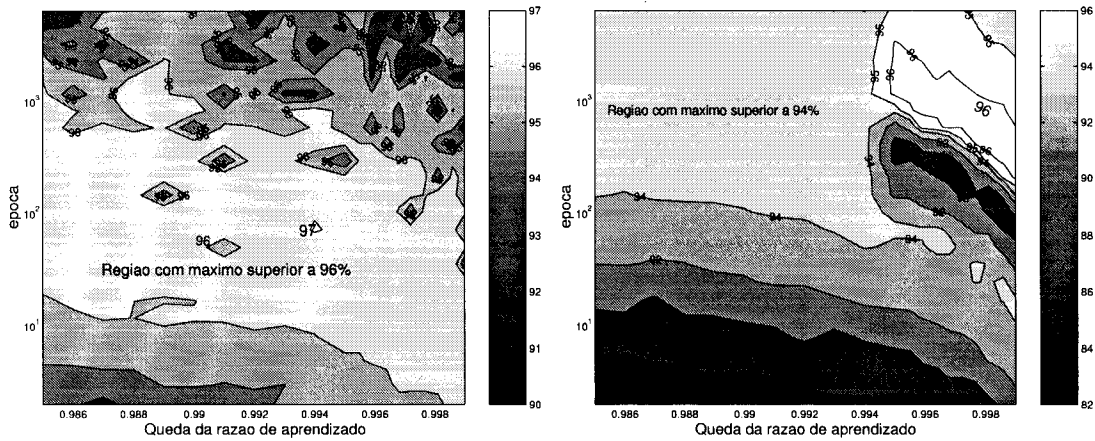


Figura 5.32: Concordância máxima nos três conjuntos de dados estudados combinadas para a rede com normalização pela raiz quadrada da energia total (esquerda) e pela energia no detector (direita). Separação elétron/píon/múon.

medida	acima de 95%		acima de 96%	
	Norm. raiz	Norm. soma	Norm. raiz	Norm. soma
# pontos	377	77	239	33
%	81,96	16,74	51,91	7,17

Tabela 5.15: Número de pontos no espaço de parâmetros e porcentagem com relação ao total (460 pontos) dos pontos que atingiram mais do que 95% e 96% de eficiência. Separação elétron/píon/múon.

faixas bem mais largas no espaço de parâmetros.

Sendo assim, concluímos que, com um aumento no risco de cairmos numa região de menor eficiência, temos, usando a normalização pela raiz quadrada, a chance de atingirmos valores mais altos de concordância entre o método clássico e o método neural.

5.4 Dados Simulados

Nesta seção, exploraremos o uso de dados simulados como entrada para ambos classificador neural e método clássico. O uso desta ferramenta nos permite avaliar melhor a performance do método neural, confrontado com o método clássico utili-

zado como validador, já que, usando dados simulados, temos certeza sobre a classe a que pertence o evento em análise, não havendo dúvidas sobre a sua classificação clássica. A única dificuldade, com relação ao uso de simuladores do calorímetro, reside no fato de que a física que descreve os processos de deposição de energia não é muito simples, podendo não ser mesmo muito realística.

Além disso, uma perfeita descrição do detector deve ser realizada, o que, muitas vezes, envolve a aplicação de complexos descritores de geometria e materiais. Atualmente, as interações eletromagnéticas (causadas por partículas como elétrons e fótons) são bem realizadas pelos programas. As interações hadrônicas, causadas por píons ou prótons, são mais complexas e discrepâncias significativas são encontradas entre os dados experimentais e os simulados.

5.4.1 Simuladores para o LHC

Durante o desenvolvimento do projeto do detector ATLAS, um dos pontos-chaves para uma perfeita compreensão das necessidades de detecção é a utilização de simuladores de colisões, da geometria do detector e da interação das partículas incidentes com o detector. Estes simuladores estudam os processos que podem vir a ocorrer no ATLAS, a resposta do detector e mesmo a frequência com que alguns dos processos físicos podem vir a aparecer. Assim, várias bibliotecas de *software* vêm sendo produzidas, com estimativas do comportamento esperado para o ATLAS nas condições de operação do LHC. Claramente, dado que muitos destes resultados dependem de fatores para os quais só existem previsões teóricas, existem muitas partes onde existem incertezas, principalmente na parte geradora das colisões de partículas, quando muitas hipóteses, conjecturas e simplificações sobre qual teoria física deve ser aplicada a cada caso podem interferir nos resultados.

O programa PYTHIA [62] é um destes programas que gera colisões de partículas no ambiente do LHC, baseando-se no Modelo Padrão. A saída deste programa produz dados sobre as partículas vindas da linha do feixe, após uma colisão, num formato de descrição específico, já projetado para funcionar corretamente com os simuladores do detector, que processarão estes eventos.

Existem vários simuladores de detectores. Um dos mais utilizados é o simulador GEANT [63]. Este simulador consegue caracterizar os diferentes efeitos físicos

resultantes da incidência das partículas sobre os sub-detectors do ATLAS, incluindo aí, possíveis efeitos de distorção de trajetória decorrentes dos campos magnéticos do detector. Num caso mais simples, pode-se simular apenas um sub-detector, no caso o calorímetro de telhas, para avaliação de suas características, tais como nível de radiação, extensão da cascata na estrutura do detector, resposta temporal do mesmo, entre outras. O simulador calcula a seção de choque das partículas na estrutura do calorímetro e, baseando-se num gerador de Monte Carlo [64], simula as interações das partículas com os átomos dos materiais que compõem esta estrutura.

O único problema do simulador GEANT é que a acuidade da descrição das interações hadrônicas (as interações de píons, prótons e nêutrons) deste pacote, apesar de continuamente melhorada, ainda não permite uma plena simulação das condições experimentais para calorímetros hadrônicos, como é o caso do TileCal. Desta forma, a cada nova versão do programa, novas tentativas são feitas no sentido de melhor simular os efeitos destas interações no detector [65]. A maior dificuldade é a alta flutuação estatística, que pode acontecer nas interações hadrônicas. Assim, embora não seja um método de validação definitivo, podemos aplicar o método neural proposto nos conjuntos de dados simulados pelo GEANT, numa situação em que, ao contrário da realidade, temos certeza do tipo de interação com a qual estamos lidando, porém, sem plena reprodução das interações reais que ocorrem no detector. Além disso, aplicamos também a análise clássica aos mesmos dados simulados, de modo a verificar o nível de acuidade de ambos os métodos para este conjunto de dados de referência.

5.4.2 Produzindo os Dados Simulados

Assim, foram produzidos, através do GEANT, conjuntos de dados para as mesmas energias utilizadas nos testes desenvolvidos até aqui. Deve-se destacar que o GEANT produz a energia depositada em cada conjunto de telhas que representa uma célula do detector o que corresponde a uma pequena fração da energia total absorvida no detector, sem nenhuma correção da eletrônica, como a que temos no caso dos dados experimentais. Assim, precisamos re-escalar os valores obtidos de energia para que a média da energia total no detector atinja o valor nominal do feixe que está sendo simulado. Este procedimento é similar àquele adotado durante as

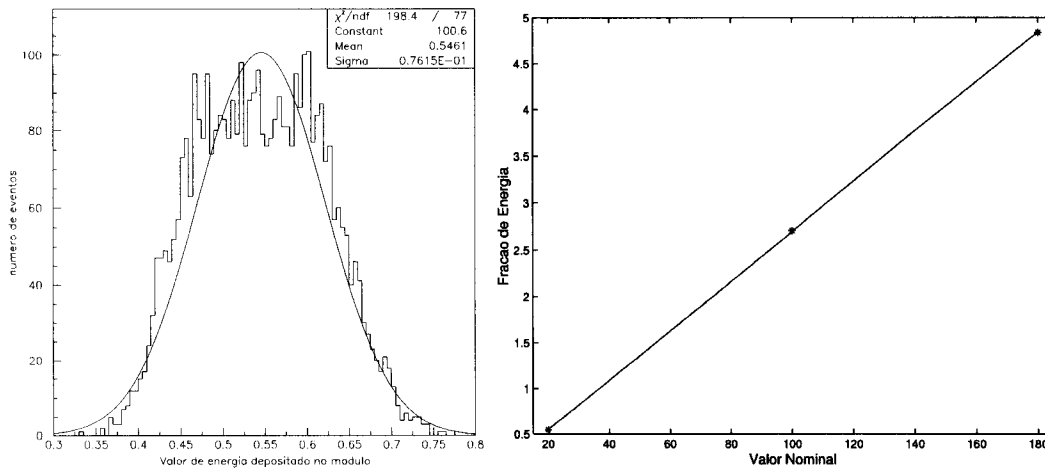


Figura 5.33: Distribuição para elétrons simulados de 20 GeV (esquerda) e a linearidade da simulação (direita).

calibrações dos módulos com feixes experimentais de partículas, quando a resposta da eletrônica do detector é re-escalada para que o valor médio medido seja o valor nominal da energia do feixe. No histograma da Figura 5.33, temos um exemplo da energia depositada por uma simulação de 20 GeV, considerando-se um feixe de elétrons. No caso, a média obtida foi 0,5461, através do *fitting* Gaussiano sobre o histograma. À direita, na Figura 5.33, temos os valores obtidos para a faixa de energia coberta (incluindo, ainda, 100 e 180 GeV), indicando a variação linear (não linearidade menor do que 1%) da deposição de energia no calorímetro.

A geometria usada, neste caso, considerou cinco módulos do Barril, com o feixe incidindo no módulo do meio, visando a máxima contenção possível da cascata. Como o feixe de elétrons fica contido no módulo sobre o qual incidiu, o seu valor é utilizado para re-escalar também eventos dos feixes de píons e de múons simulados. Entretanto, dada a não compensação do TileCal [33] (o fato de que a energia da parte hadrônica da cascata não ser medida da mesma forma que a energia na parte eletromagnética), o calorímetro responde com sinais ligeiramente inferiores para o feixe de píons. Este fato, entretanto, é uma propriedade intrínseca do detector, e, portanto, aparece também em condições experimentais.

Assim, realizamos o experimento que consistia nos seguintes passos: primeiramente, re-escalamos o valor de cada evento segundo a média dos eventos estimada a partir do *fitting* Gaussiano para o conjunto de elétrons encontrada para eventos

da mesma energia nominal que a do evento em questão. Assim, eventos de elétrons, píons e múons a 20 GeV, foram corrigidos pelo valor $20/0,5461$ (valor nominal dividido pelo valor médio encontrado no *fitting*, resultando em 36.62. Em seguida, aplicamos os dois principais tipos de normalização que discutimos na seção 5.2, quais sejam, a normalização pela energia total e a normalização pela raiz quadrada do valor absoluto da energia total. Finalmente, aplicamos o processamento neural, para verificarmos o índice de acerto e o comparamos com o método clássico aplicado aos mesmos eventos. Além destes resultados, exploramos também a possibilidade de induzirmos contaminações nos dados simulados, simplesmente utilizando parte dos eventos de um dado conjunto (por exemplo, eventos de múons), para treino em outro conjunto (de píons, por exemplo), simulando o caso real de contaminação por múons no feixe experimental de píons. Assim, conseguimos simular a situação real e avaliar a performance de acerto nos conjuntos puros, já que temos como controlar quais eventos pertencem a que conjuntos.

5.4.3 Separação Píon/múon

Inicialmente, vamos analisar o caso de 20 GeV, com normalização pela energia total. Na Tabela 5.16, temos, da mesma forma que nas seções anteriores, o índice de concordância entre ambos os métodos neural e clássico para cada tipo de partícula. Neste caso, entretanto, incluímos uma coluna e linha extra em cada tabela, totalizando o percentual de eventos identificados por cada método. Assim, vemos que, para o conjunto de píons, na tabela da esquerda, 96,85% dos eventos foram reconhecidos como píons por ambos os métodos. Por outro lado, 2,25% dos eventos foram reconhecidos como píons pelo método clássico (corte em 7 GeV) e como múons pela rede neural. No total, 99,10% dos eventos foram reconhecidos como píons pelo método clássico. Por seu turno, a rede neural reconheceu 97,00% dos eventos como píons. Neste caso, vemos que o método neural teve uma performance inferior em relação ao método clássico. O mesmo pode se dizer para o conjunto de múons, no qual o método clássico errou em 0,60% dos eventos e o método neural errou em 1,80% dos eventos. Ainda assim, entretanto, a eficiência de acerto do método neural é bastante elevada, em ambos os casos.

Em seguida, realizamos o treino da rede neural, utilizando não mais a energia

Método Clássico	Rede Neural			Método Clássico	Rede Neural		
	π	μ	tot		π	μ	tot
π	96,85%	2,25%	99,10%	π	0,55%	0,05%	0,60%
μ	0,15%	0,75%	0,90%	μ	1,25%	98,15%	99,40%
tot	97,00%	3,00%	100%	tot	1,80%	98,20%	100%

Tabela 5.16: Concordância entre os métodos neural com normalização pela energia total e clássico, para feixes puros de píons (esquerda, com concordância de 97,60%) e múons (direita, com concordância de 98,70%), usando dados simulados pelo GEANT. Feixes de 20 GeV.

Método Clássico	Rede Neural			Método Clássico	Rede Neural		
	π	μ	tot		π	μ	tot
π	99,05%	0,05%	99,10%	π	0,50%	0,10%	0,60%
μ	0,00%	0,90%	0,90%	μ	0,05%	99,35%	99,40%
tot	99,05%	0,95%	100%	tot	0,55%	99,45%	100%

Tabela 5.17: Concordância entre os métodos neural, com normalização baseada na raiz quadrada da energia, e clássico, para feixes de 20 GeV puros de píons (esquerda, com concordância total de 99,95%) e múons (direita, com concordância de 99,85%) usando dados simulados pelo GEANT.

total, mas sim a raiz quadrada do absoluto da energia depositada em todo o detector como fator de normalização. Os resultados estão sumarizados na Tabela 5.17. Desta vez, podemos ver que, para ambos os conjuntos, o índice de concordância entre os métodos subiu bastante, e que, embora a rede erre 0,05% na identificação no conjunto de píons (tabela à esquerda), o processamento neural acerta melhor do que o método clássico exatamente este mesmo índice (0,05%) no conjunto de múons (o erro na identificação de múons é 0,60% pelo método clássico e 0,55% pelo método neural).

No terceiro caso, simulamos uma contaminação de 23% de múons no conjunto de píons, mantendo, é claro, o conjunto de múons puro, visando simular uma situação próxima a encontrada experimentalmente. Neste caso, a Tabela 5.18 apresenta os

Método Clássico	Rede Neural			Método Clássico	Rede Neural		
	π	μ	tot		π	μ	tot
π	99,10%	0,00%	99,10%	π	0,60%	0,00%	0,60%
μ	0,00%	0,90%	0,90%	μ	0,10%	99,30%	99,40%
tot	99,10%	0,90%	100%	tot	0,70%	99,30%	100%

Tabela 5.18: Concordância entre o método neural, com normalização baseada em raiz quadrada da energia e o método clássico para feixes de píons (esquerda, com concordância de 100%) e múons (direita, com concordância de 99,90%) usando dados simulados pelo GEANT.

resultados. A normalização escolhida foi a raiz quadrada, por ter se mostrado a melhor metodologia.

Neste último caso, podemos ver que as concordâncias entre os métodos atingiram valores bastante altos e também que o método neural teve uma performance inferior àquela do método clássico em apenas 0,1%, no conjunto de múons, igualando a performance do método clássico no conjunto de píons.

Para as outras energias, encontramos resultados bastante semelhantes. Para 100 GeV, temos, na Tabela 5.19, as identificações neurais (média da identificação nos conjuntos de píons e múons) e concordâncias com o método clássico (baseado no corte em 20 GeV para feixes de 100 e 180 GeV) e os três métodos de análise que utilizamos até aqui (normalização pela energia total, pela raiz da energia e simulação de 23% de contaminação). A identificação por métodos clássicos para píons ficou em 99,50% e, para múons, em 99,65%. Na média dos dois conjuntos de dados (píons e múons) a concordância foi de 99,57%.

Para 180 GeV, temos os resultados que estão expressos na Tabela 5.20. A identificação usando o método clássico, neste caso, ficou em 100% para píons e 98,95% para múons. Na média, tivemos 99,47% de identificação por métodos clássicos.

Normalização	Id. π	Id. μ	Média	Conc.
E_t	98,65%	99,20%	98,92%	99,10%
$\sqrt{ E_t }$	99,40%	99,65%	99,52%	99,95%
23% contaminação	99,50%	99,65%	99,57%	100%

Tabela 5.19: Percentual de identificação para píons e múons pela rede neural nos três métodos (normalização pela energia total, pela raiz da energia e com conjunto de píons contaminado por múons) e concordância com o método clássico usando um feixe de 100 GeV.

Normalização	Id. π	Id. μ	Média	Conc.
E_t	99,40%	99,20%	99,30%	98,97%
$\sqrt{ E_t }$	99,90%	99,70%	99,80%	99,57%
23% contaminação	99,90%	99,60%	99,75%	99,62%

Tabela 5.20: Percentual de identificação para píons e múons pela rede neural nos três métodos (normalização pela energia total, pela raiz da energia e com conjuntos de píons contaminados por múons) e concordância com o método clássico usando feixes de 180 GeV.

energia	Id. Clássica			
	e	π	μ	total
20	99,95%	79,20%	99,40%	92,85%
100	100%	95,90%	99,65%	98,51%
180	100%	98,51%	98,95%	99,15%

Tabela 5.21: Identificação das partículas em cada conjunto de dados adquirido de um feixe simulado segundo o método clássico.

5.4.4 Separação Elétron/píon/múon

Da mesma forma, realizamos a separação entre elétrons, píons e múons. A Tabela 5.21 traz os resultados de identificação pelo método clássico baseado nos mesmos cortes nas frações de energia e energia total que havíamos utilizado com dados experimentais, para os três níveis de energia discutidos. No caso, a média da identificação em cada conjunto de dados proveniente de um feixe (elétrons, píons e múons) foi também calculada por faixa de energia.

Como podemos ver nesta tabela, o índice de identificação para 20 GeV é relativamente baixo. Isto se deve ao fato de não termos o Contador de Cherenkov para esta situação, já que o mesmo é um detector auxiliar, posicionado na linha do feixe, durante testes com partículas no ambiente experimental. A solução adotada consistiu em usarmos o mesmo método clássico que vimos utilizando para as energias mais altas de feixe (100 e 180 GeV), baseado em cortes de energia nas frações depositadas nas camadas. Esta é a maior causa deste erro ter sido relativamente elevado, já que tal método clássico não funciona bem para níveis mais baixos de energia, dada a redução no tamanho das cascatas de elétrons e píons, o que leva a uma confusão entre as duas, tanto nas frações de energia depositadas na primeira e terceira camada de células do detector, quanto na energia que atinge os módulos periféricos. Estes módulos se encontram situados acima e abaixo do módulo que recebe o feixe, posto que foram simulados cinco módulos do barril com o feixe atingindo o módulo do meio.

Os resultados da análise neural para este conjunto estão na Tabela 5.22. Nesta tabela, os resultados são apresentados considerando-se a média dos resultados

Normalização	20 GeV		100 GeV		180GeV	
	Id. Neu	Conc.	Id. Neu.	Conc.	Id. Neu.	Conc.
energia	95,65%	90,65%	98,92%	98,01%	99,14%	98,59%
raiz	96,15%	91,66%	98,65%	97,63%	98,99%	98,68%
raiz cont.	96,57%	91,88%	98,68%	97,70%	98,87%	98,53%

Tabela 5.22: Identificação neural para dados simulados em três diferentes energias e as respectivas concordâncias com o método clássico, considerando-se dois métodos de normalização (pela energia total, pela raiz quadrada da energia e a indução de contaminação do feixe).

relativos a cada um dos três tipos de partículas (elétrons, píons e múons). Assim, por exemplo, para o feixe simulado de elétrons a 20 GeV, 98% dos eventos foram identificados pelo método neural usando a normalização pela energia total como sendo elétrons. No conjunto de píons, esta identificação cai para 90,20% (pior de todos os casos) e, para múons, a rede identifica 98,75%. Assim sendo, temos na média 95,65% que é o resultado apresentado na tabela.

Vemos, claramente, que nos três métodos estudados, os casos de 20 GeV e de 100 GeV obtiveram melhores resultados do que o método clássico apresentado na Tabela 5.21, 92,85% para 20 GeV e 98,51% para 100 GeV. Conforme já dissemos, nenhuma eficiência individual ficou menor do que 90,2%. O caso de 180 GeV teve resultados de identificação neural um pouco abaixo daqueles atingidos pelo método clássico, sem, entretanto, comprometer muito o resultado final. Vemos também que, para os casos de 100 e 180 GeV, quando o método tem melhor desempenho do que a 20 GeV, o índice de concordância entre ambos os métodos (neural e clássico) se mantém alto (acima de 97,6%).

Concluimos assim que o método neural mantém um alto nível de correspondência com o método clássico, o superando em alguns casos (particularmente em níveis mais baixos de energia), perdendo em outros.

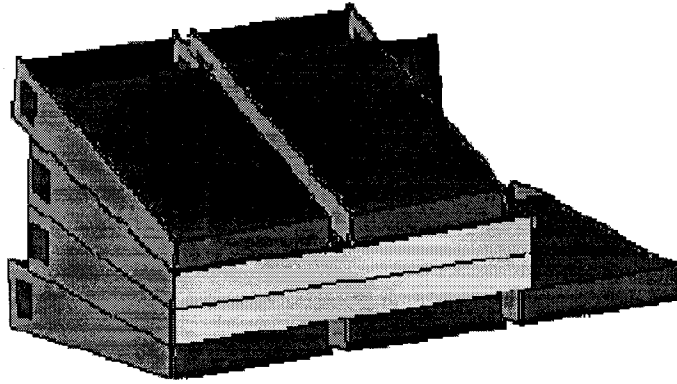


Figura 5.34: Módulos da Extensão do Barril configurados para teste. Os cinco módulos também foram utilizados.

5.5 Extensões do Barril

Nesta seção estudaremos o desempenho de nosso sistema neural para os módulos da Extensão do Barril. Alguns testes experimentais foram realizados com o feixe de partículas atingindo uma combinação destes módulos do TileCal e discutiremos aqui os resultados obtidos.

5.5.1 Arranjo Experimental e Geometria

A configuração utilizada para teste experimentais com a seção do barril estendido pode ser vista na Figura 5.34. Como podemos ver, temos dois módulos da Extensão do Barril, que se encontram no meio, com três módulos dos cinco protótipos da geração anterior posicionados embaixo e os outros dois por cima, para reduzir o vazamento de energia das cascatas de partículas mais largas.

Os dois módulos foram testados conjuntamente, mas com feixe direcionado para apenas um deles de cada vez. Desta forma, uma utilização mais eficaz do feixe de partículas pode ser realizada, posto que não há pausa no direcionamento do feixe de partículas para a incidência nos módulos. Além do mais, durante o teste dois institutos de pesquisas diferentes, que realizaram a montagem dos módulos individualmente, puderam verificar possíveis diferenças na resposta dos mesmos, que estivessem associadas, eventualmente, a alguma diferença no processo de produção de módulos.

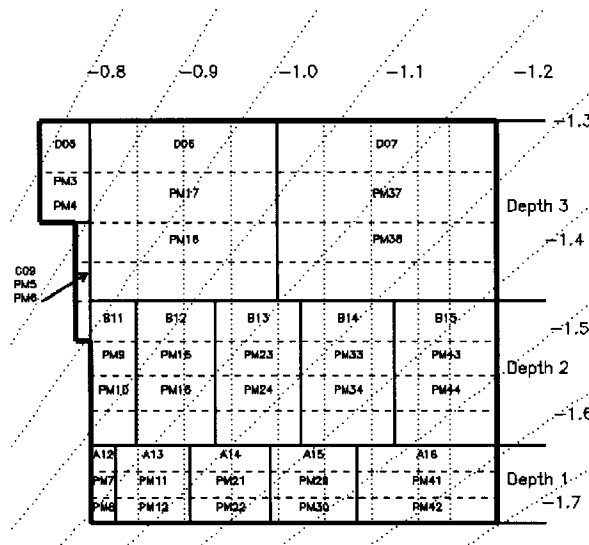


Figura 5.35: Disposição das células para módulos da Extensão do Barril.

Como já apontamos anteriormente, a avaliação do processamento neural para esta configuração experimental é de suma importância, por representar o seu comportamento para uma geometria diferente daquela do Módulo 0.

A geometria das seções da Extensão do Barril pode ser vista na Figura 5.35. Neste caso, contamos também com três camadas de células, mas a geometria delas é bastante diferente da geometria dos módulos do Barril. As células da terceira camada (D05, D06 e D07), por exemplo, são bem maiores do que as da terceira camada do Barril. Dependendo do teste a ser realizado, algumas vezes o feixe chegava a penetrar no módulo pela sua lateral esquerda. O estudo dos métodos clássicos fica, inclusive, bem mais complicado. Por exemplo, a definição da primeira camada (bastante usada nos nossos estudos anteriores) fica dependendo bem mais da posição do feixe do que nos estudos do Barril. Como a primeira camada inclui, na verdade, a célula de impacto do feixe de partículas no calorímetro, temos, para a posição $\eta = -1, 1$, usada em nossos testes adiante, a "primeira camada" sendo a primeira célula da segunda camada (B11 na figura).

Outro detalhe é o menor número de células deste caso. Temos 5 células na primeira camada, 6 na segunda (a pequena célula marcada como C09 na figura entre a segunda e a terceira camadas é considerada como sendo da segunda camada) e 3 células na terceira. Assim, temos no total 14 células com dupla leitura, o que leva a 28 canais de leitura para a rede neural. Assim, nossa rede neural para este caso

terá 28 entradas ao contrário das 46 usadas para módulos do Barril.

Uma dificuldade que enfrentamos, entretanto, foi encontrar dados experimentais suficientes com incidência projetiva do feixe (segundo a direção η). Assim, não pudemos reproduzir as análises para as mesmas energias de feixe usadas no estudo do Barril, tendo trabalhado com as energias 20, 80 (ambas em $\eta = -1, 1$) e 100 (em $\eta = -1, 3$).

5.5.2 Resultados da Aplicação Neural

A rede foi projetada usando-se os mesmos parâmetros de treinamento definidos para o caso do módulo do Barril, modificando-se apenas o número de nós de entrada. Mantiveram-se os 8 neurônios da camada intermediária e 1 (ou 3) neurônio na camada de saída. Quando se utiliza um único neurônio de saída, realiza-se a separação pión/múon, que é desenvolvida para identificar eventos de um feixe de píons. Para um feixe de elétrons, a separação elétron/pión/múon necessita dos três neurônios de saída, cada um associado a uma classe de partículas. A comparação com os métodos clássicos foi feita baseando-se no corte em energia (7 GeV) mais corte no contador Cherenkov (80 contagens) para 20 GeV e cortes nas frações de energia para as outras energias, variando com a posição e energia do feixe. Assim, por exemplo, para o feixe em $\eta = 1, 1$ a célula B11 foi utilizada como primeira camada (com corte para separar elétrons de píons em 64% da energia total absorvida pelo conjunto de calorímetros) a terceira camada foi utilizada completa (com corte para separar píons de elétrons em 0,5%) e corte na energia que vaza para o outro módulo da extensão do Barril e os cinco módulos protótipos (em 4,2% para definir elétrons). O corte de 20 GeV, em altas energias, para separar elétrons e píons de múons também foi utilizado. Mais uma vez, estudamos a resposta da rede para as duas normalizações (pela energia total depositada no módulo e pela raiz quadrada desta energia).

Os resultados de concordância na separação de partículas para feixes de píons e múons, através da rede neural e pelo método clássico, estão apontados na Tabela 5.23. Os resultados são a média das concordâncias nos dois conjuntos de dados experimentais (píons e múons) para cada caso.

Como podemos ver, temos resultados de concordância bastante altos (maio-

Energia	E_t	$\sqrt{ E_t }$
20	96,58%	97,38%
80	96,72%	99,60%
100	97,25%	97,70%

Tabela 5.23: Concordância da resposta da rede neural com o corte em energia (método clássico) para três energias e dois enfoques de normalizações. Resultados para as Extensões do Barril e dados provenientes de feixes de píons e múons.

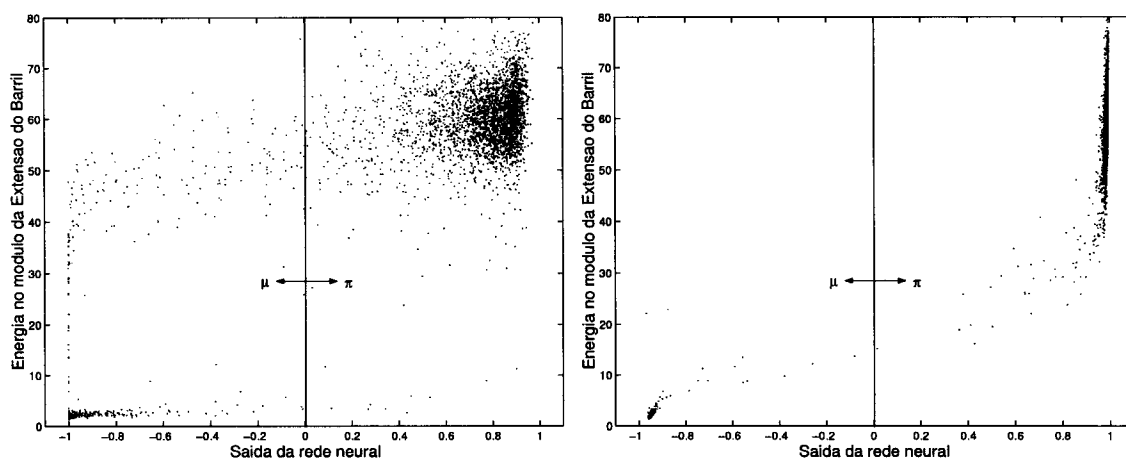


Figura 5.36: Correlação entre a saída da rede, com normalização pela energia total (esquerda) e pela raiz quadrada da energia (direita), com a energia depositada no módulo da extensão do barril para um feixe de píons de 80 GeV.

res do que 96,58%), o que denota um bom funcionamento do sistema baseado em redes neurais. Vemos também que, de novo, temos resultados melhores quando a normalização é baseada na raiz quadrada da energia total. Assim, podemos dizer que os resultados ora obtidos não são muito diferentes do que os discutidos para o módulo do Barril.

Na Figura 5.36, temos as saída da rede correlacionada com a energia total para a situação de feixe de píons a 80 GeV e com normalização pela energia total depositada no módulo (esquerda) ou pela raiz quadrada da energia (direita). Vemos que os eventos com mais baixa energia foram considerados múons enquanto os eventos de alta energia foram considerados píons.

Podemos também observar o efeito da normalização pela energia que volta a

Energia	E_t	$\sqrt{ E_t }$
20	92,52%	94,11%
80	93,60%	95,03%
100	91,02%	93,40%

Tabela 5.24: Concordância da resposta da rede neural com o corte em energia (método clássico) para diferentes energias e dois enfoques de normalizações. Resultados para a separação elétron/píon/múon nas Extensões do Barril.

exibir tendência na classificação de partículas, como já vimos no caso dos módulos do Barril. Alguns eventos, apesar de terem deixado bastante energia no detector, são identificados como múons pela rede neural classificadora. Vemos que este efeito desaparece com o uso da raiz quadrada da energia como fator de normalização.

Realizamos também os testes para a separação elétron/píon/múon. Neste caso, temos a Tabela 5.24 que sumariza os resultados de concordância entre o método neural e o método clássico. Novamente, consideramos as duas normalizações e os três níveis de energia utilizados até agora.

Podemos ver que a concordância entre os métodos, que já era superior a 91% para a normalização pela energia total, aumentou quando a normalização pela raiz quadrada é aplicada, o que concorda com todos os resultados que exploramos até o momento. Desta maneira, conclui-se que o processamento neural é bem sucedido na tarefa de identificação de partículas também para a seção da extensão do barril.

5.5.3 Estudos das Relevâncias

Para os casos de 80 e 100 GeV, como há uma mudança de posição do feixe (de $\eta = -1,1$ a 80 GeV para $\eta = -1,3$ a 100 GeV), é interessante observarmos a relevância relativa à estes dois pontos de impacto.

Na parte da esquerda da Figura 5.37, temos as relevâncias para 80 GeV com o feixe em $\eta = -1,1$. Vemos, neste caso, que a maior relevância se deu para a segunda célula da primeira camada (célula B11 na Figura 5.35), que corresponde às componentes 13 e 14 em termos do vetor de entrada da rede neural. Na figura da direita, temos que a célula de maior relevância acabou por ser encontrar na primeira

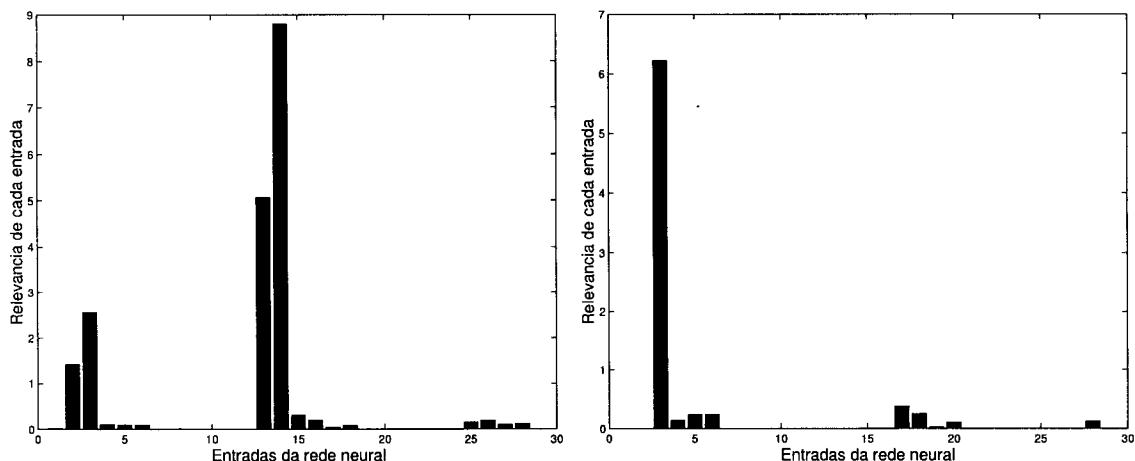


Figura 5.37: Relevâncias da energia depositada em cada célula do detector para feixes de 80 GeV (esquerda) e 100 GeV (direita).

camada (célula A13 na Figura 5.35, correspondente às entradas 3 e 4 da rede neural), o que está de acordo com a nova posição do feixe em $\eta = -1, 3$. Vemos também que alguma relevância foi registrada nas posições 17 e 18, que correspondem à célula da segunda camada por onde passa o feixe.

Sendo assim, concluímos que a metodologia neural apresentada não depende necessariamente do calorímetro em questão, bastando apenas que este disponibilize uma informação detalhada do perfil de deposição das partículas.

5.6 Análise de Sensibilidade

Nesta seção, iremos detalhar a análise relativa à sensibilidade do resultado de identificação de partículas obtido pela rede neural, quando aplicada a uma situação diferente daquela em que foi treinada, em relação à energia do feixe incidente. Esta análise permite um alargamento da aplicabilidade de uma dada rede a uma situação diferente daquela do projeto inicial, mostrando a sua possível independência da energia.

Assim sendo, utilizamos as redes projetadas para identificar elétrons, píons e múons para três diferentes energias e com dois métodos de normalização (pela energia total e pela raiz quadrada da energia), considerando-se um módulo do barril do TileCal. Para cada uma das redes projetadas, avaliamos o índice de concordância

Ener.	e	π	μ	Ener.	e	π	μ
100 GeV	91,20%	93,09%	97,04%	100 GeV	86,00%	37,08%	98,14%
180 GeV	89,56%	89,83%	97,04%	180 GeV	57,65%	6,48%	98,14%

Tabela 5.25: Índices de concordância entre os métodos clássico e neural para dados provenientes de feixes de 100 e 180 GeV para um treinamento da rede neural com feixes de 20 GeV. Normalização pela energia total (esquerda) e pela raiz da a energia (direita).

das mesmas com os métodos clássicos, quando aplicados a dados de feixes de outras energias. No primeiro caso, utilizando as redes projetadas para 20 GeV obtivemos os resultados de concordância com métodos clássicos para a aplicação nos dados provenientes dos feixes (elétrons, pión e múons) com 100 e 180 GeV. Os resultados se encontram na Tabela 5.25.

Como podemos ver, os resultados com normalização pela energia total no detector (na tabela a esquerda) conseguem ser bem mais estáveis. Isto se deve ao fato de que o treino com esta normalização torna a rede bastante independente da energia, haja visto que esta normalização não introduz uma informação definida sobre o nível de energia do feixe incidente, concentrando-se no perfil de deposição de energia relativo de cada classe de partículas. Entretanto, uma pequena queda de performance é observada nos dados de feixes de elétrons e píons (veja a Tabela 5.6), o que era de se esperar, posto que as cascatas destes tipos de partícula têm um perfil de deposição que varia com a energia. A cascata de píons, por exemplo, invade muito mais a segunda e mesmo a terceira camadas de células do detector quando a energia do feixe se eleva significativamente.

Entretanto, quando se utiliza a normalização pela raiz quadrada da energia total, quando, então, a informação de energia absoluta é introduzida diretamente, o classificador neural colapsa na identificação de elétrons e píons. Este resultado era esperado, já que a dependência com a energia, introduzida para diminuir a possibilidade de tendência na amostra de dados filtrada, pondera os perfis de deposição de energia, fazendo uma maior dependência da rede da sua fase de treinamento. Assim, temos aqui uma forte limitação do uso de raiz quadrada como processo normaliza-

Ener.	e	π	μ	Ener.	e	π	μ
20 GeV	87,59%	76,06%	92,73%	20 GeV	56,66%	25,09%	96,66%
180 GeV	91,23%	88,57%	93,41%	180 GeV	97,04%	94,70%	99,61%

Tabela 5.26: Índices de concordância entre os métodos clássico e neural para dados provenientes de feixes de 20 e 180 GeV para um treinamento da rede neural com feixes de 100 GeV. Normalização pela energia total (esquerda) e pela raiz da a energia (direita).

dor. Toda vez que altera-se a energia do feixe de partículas, é necessário um retreino da rede, neste caso, para podermos garantir uma alta eficiência. Ou seja, quando uma única rede treinada deve ser utilizada na aplicação, a opção pela normalização baseada na energia total é, claramente, a melhor opção.

Vamos agora verificar a situação das redes treinadas com feixes de 100 GeV aplicados a conjuntos de dados provenientes de feixe de 20 e 180 GeV. Os resultados estão na Tabela 5.26.

Mais uma vez, na tabela da esquerda, temos os resultados para a normalização pela energia total e, na da direita, pela raiz quadrada. Como podemos ver, na tabela da esquerda, a rede treinada para 100 GeV, quando aplicada a dados com 20 GeV, perdeu bastante em termos de capacidade de classificação para píons. Isto é explicável pelo fato de a cascata de píons para 20 GeV não penetrar muito no detector, podendo mesmo penetrar menos do que uma cascata de elétrons a 100 GeV, depositando, praticamente, a sua energia na primeira camada de células do detector, em relação ao ponto de impacto. Ora, a rede treinada com dados a 100 GeV, ou seja, com cascatas de elétrons um pouco mais profundas, confunde mais estes dois tipos, reduzindo a eficiência na identificação de píons. Para 180 GeV, o índice de concordância fica maior, posto que na faixa de energias de 100 a 180 GeV, a diferença de estrutura da cascata tende a ser menor e a normalização, neste caso, não prejudica muito a análise.

Para a tabela da direita, que traz a normalização pela raiz quadrada da energia total, notamos que a rede treinada com feixes de 100 GeV colapsa para os feixes de 20 GeV, Entretanto, quando aplicada a feixes de 180 GeV apresenta

Ener.	e	π	μ	Ener.	e	π	μ
20 GeV	86,13%	73,77%	96,26%	20 GeV	81,85%	50,64%	18,78%
100 GeV	94,06%	94,69%	96,86%	100 GeV	22,53%	27,82%	80,28%

Tabela 5.27: Índices de concordância entre os métodos clássico e neural para dados provenientes de feixes de 20 e 100 GeV para um treinamento da rede neural com feixes de 180 GeV. Normalização pela energia total (esquerda) e pela raiz da a energia (direita).

uma performance bastante boa. Uma possível explicação para este fato é a pequena diferença de ponderação; a raiz quadrada de 100 é 10 e a de 180 é 13,4, o que se traduz num pequeno fator de escala.

Finalmente, na Tabela 5.27, uma rede treinada com feixes de 180 GeV é aplicada em dados de 20 e 100 GeV. Na tabela relativa à normalização pela energia no módulo, vemos novamente a performance para píons diminuída, o que pode ser explicado pela mesma taxa do caso anterior da rede treinada a 100 GeV. As cascatas de píons a 20 GeV penetram pouco no calorímetro e são confundidas com elétrons. Na aplicação a 100 GeV, vemos, de novo, uma boa performance, mostrando a provável semelhança entre os perfis de deposição de energia na afixa de energia de 100 a 180 GeV.

No caso da normalização pela raiz quadrada da energia total, que temos na tabela da direita, vemos que as performances de classificação estão bem degradadas. O maior problema, no caso, se registra pelo fato de que a rede considerou os eventos de elétrons e píons, por estarem numa faixa de energia menor (20 e 100 GeV) do que os elétrons e píons usados no treino da rede (180 GeV) como sendo eventos de múons. Ao que parece, o limite usado pela rede para diferenciar elétrons e píons de múons, deve ter sido um limite de energia bem elevado, posto que mesmo para 100 GeV a rede não conseguiu manter um bom nível de classificação.

Diante dos resultados, a normalização pela energia total confirma a sua robustez e pode ser indicada para um projeto de classificador que tenha que operar numa faixa de energia do feixe experimental.

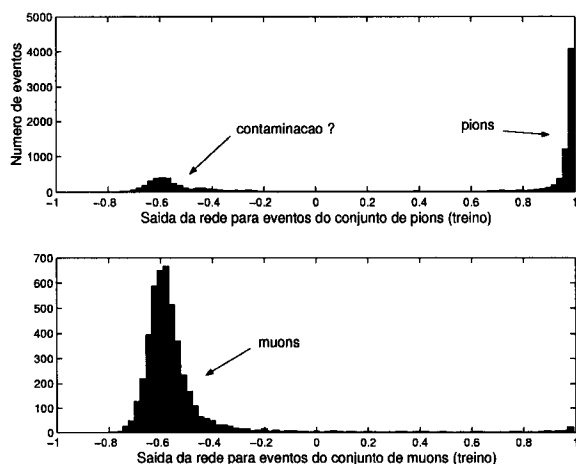


Figura 5.38: Saídas da rede neural para feixes de píons contaminados por múons (histograma superior) e de múons (inferior), considerando-se os conjuntos de treinamento. Feixes de 20 GeV.

5.7 Estudo das Saídas da Rede Neural

Quando começamos a análise *offline* no Capítulo 5, um dos primeiros resultados abordados foi a saída da rede neural para o problema da separação píon/múon. Reproduzimos, na Figura 5.38, os resultados obtidos naquela primeira seção.

Nesta figura, temos as saídas da rede para um feixe incidente de píons a 20 GeV, no histograma superior, e múons, no histograma inferior. Havíamos identificado, naquele momento, uma contaminação presente na amostra de dados do feixe de píons, cuja disposição na escala do histograma sugeria (como depois pudemos confirmar), que estes eventos eram, na verdade, múons presentes no feixe experimental de píons. Entretanto, como pudemos também observar, enquanto píons identificados apresentavam resultados próximos a +1, seu valor-alvo para a fase de treinamento, os múons identificados (contaminação ou não) eram ligeiramente desviados do seu valor-alvo, atingindo uma média de -0,57, como já enunciado no Capítulo 5.

Este desvio é explicável considerando-se que durante o treinamento supervisionado da rede, os múons de contaminação identificados pela rede recebem um valor-alvo errôneo (no caso +1), enquanto que os eventos equivalentes e que são provenientes do feixe de múons recebem um valor-alvo igual a -1. Assim sendo, o treinamento, visando reduzir o erro produzido por esta dupla marcação de eventos de um mesmo padrão tenta adequar a média da saída a um valor que satisfaça

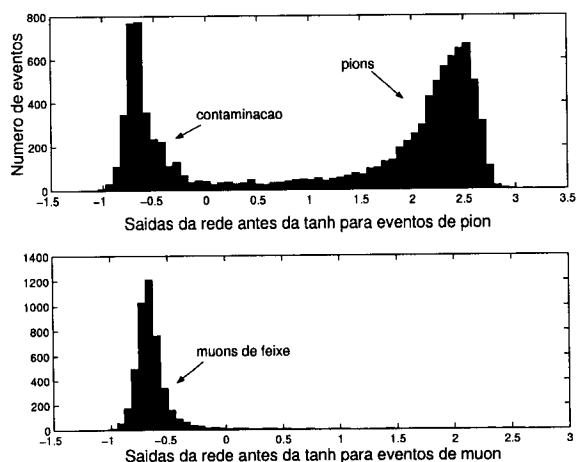


Figura 5.39: Distribuição da saída da rede tomada antes da aplicação da função de ativação do neurônio de saída para os feixes de píons contaminado (histograma superior) e puro de múons (inferior). Feixes de 20 GeV.

melhor a ambas as demandas.

Algumas perguntas surgem então desta observação. A primeira se relaciona com a existência de alguma relação entre o nível de contaminação de um dado feixe (ou conjunto de dados) e a quantidade de desvio que esta provoca na saída da rede (no caso desta discussão, a saída para múons). Uma outra dúvida seria a existência de algum impacto para a classificação de eventos quando este desvio é considerado, ou seja, se o desvio poderia levar, no caso ao qual nos referimos, a saída da rede para múons além do patamar de corte (zero para a separação pión/múon) para as duas partículas.

Na Figura 5.39, temos a resposta do neurônio da terceira camada da rede para feixes de píons e múons, sendo esta coletada antes da aplicação da função de ativação do neurônio (a tangente hiperbólica). Fizemos isto buscando descobrir alguma possível estrutura probabilística que possa refletir o efeito da contaminação na resposta da rede neural. Vemos que a saída para o feixe de múons (histograma inferior) aparece bastante centrada em torno de um valor bastante negativo, com uma pequena cauda para o lado positivo do eixo. Utilizaremos o valor médio desta distribuição para um conjunto puro de múons como medida de nosso desvio, quando o feixe de píons incidente sofre por contaminação por múons.

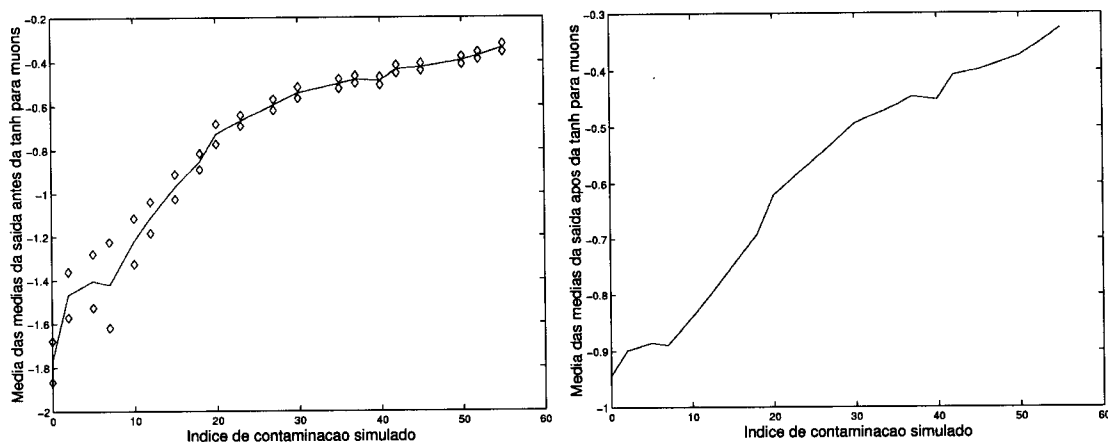


Figura 5.40: Médias das saídas da rede neural para o feixe puro de múons de 20 GeV (esquerda) e o resultado da aplicação da tangente hiperbólica sobre estas médias (direita).

5.7.1 Desvio por Múons de Contaminação

Assim, realizamos o seguinte experimento : através de corte em valores de energia, foram simulados diferentes níveis de contaminação para os conjuntos de dados. Assim, para uma dada situação, por exemplo, para um feixe de píons de 20 GeV, foram mantidos 20 eventos de baixa energia (menor do que 7 GeV) para cada 80 eventos com energia elevada (maior do que 7 GeV). Desta forma, simulamos uma contaminação de 20%. Outros níveis de contaminação foram simulados, numa faixa de 0 a 55%. Os dados com esta contaminação simulada foram aplicados à rede neural (foram realizados 100 treinamentos com diferentes inicializações da rede) e as médias da saída do neurônio da terceira camada, antes da tangente hiperbólica ser aplicada, foi avaliada para o feixe puro de múons. Assim, foi possível se traçar uma curva do desvio da saída em função dos diferentes níveis de contaminação, conforme pode ser visto na Figura 5.40.

Na parte esquerda desta figura, temos as medidas realizadas para os diferentes níveis de contaminação que simulamos. Como podemos ver, inicialmente, quando pouca contaminação é registrada, a média fica bastante negativa, em torno de -1.8. Conforme a contaminação aumenta, vemos que esta média vai sendo naturalmente "puxada" para a direção do semi-eixo positivo. No final da curva, quando nos aproximamos do semi-eixo positivo, temos uma tendência a atingir a saturação. Na

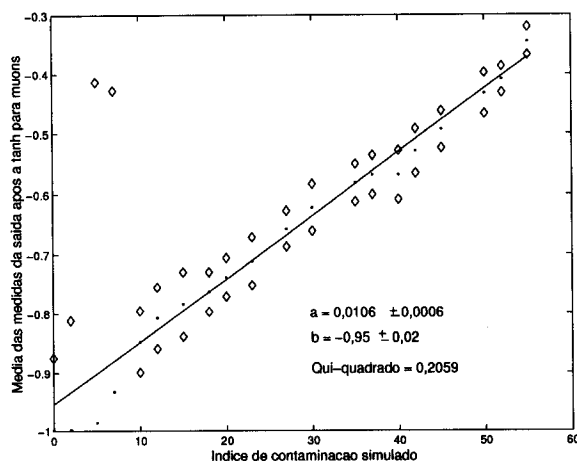


Figura 5.41: Curva da tangente hiperbólica das medidas quando a rede é treinada com normalização pela raiz quadrada da energia. A função linear é estimada por processo de *fitting*.

figura da direita, aplicamos sobre as médias da curva esquerda a função de ativação do neurônio (tangente hiperbólica). Obtemos uma curva não muito distante de uma relação linear entre o índice de contaminação e o desvio produzido no alvo de treinamento.

Os resultados apresentados até o momento se referem à separação pión/múon a 20 GeV, utilizando a normalização pela energia total absorvida no módulo. Podemos novamente estudar o problema quando utilizamos a normalização pela raiz quadrada da energia. Neste caso, a curva com a aplicação da tangente hiperbólica às médias parece bem mais linear como podemos ver na Figura 5.41.

Nesta curva, além dos pontos traçamos também a estimativa de uma curva linear (feita através de processo de *fitting* que passe por todos os pontos). Vemos que esta curva continua subindo conforme a contaminação aumenta, e, baseando-se nos parâmetros do *fitting*, que podem ser vistos na própria figura (sendo a o coeficiente angular e b o coeficiente linear), podemos estimar que com 100% de contaminação a média dos múons ficaria em 0,106 com um erro de 0,083. Idealmente, com 100% de contaminação, a rede estaria sendo treinada com dois alvos para apenas um tipo de evento. Seria então normal se esperar que a rede tendesse para a média dos alvos (zero), o que não está muito longe dos resultados obtidos com este experimento.

Sendo assim, podemos responder às duas perguntas que levantamos no início

desta seção. Primeiro, fica clara a relação entre o índice de contaminação e o desvio da média para a distribuição da saída da rede para o feixe de múons, ao final da fase de treinamento. Segundo, porque em casos extremos de contaminação, este desvio pode ter conseqüências nos resultados de classificação, quando comparamos com o patamar de discriminação até aqui adotado em zero. Poderíamos, por exemplo, ter, para cada situação, diferentes pontos de corte da resposta da rede para a classificação de eventos do feixe, nos baseando em estimativas *a priori* da contaminação (quando estas fossem, é claro, disponíveis).

Capítulo 6

Discriminador Neural Remoto de Partículas

Neste capítulo iremos estudar o sistema desenvolvido[66] na análise *offline* que vimos descrevendo nas seções anteriores deste capítulo, que consiste em uma interface via páginas http da internet capaz de realizar a análise neural e, através de um programa de análise estatística feito por pesquisadores do CERN, podem ser realizadas análises complementares. Este sistema foi implementado para que se pudesse disponibilizar a pesquisadores do mundo inteiro a técnica de análise neural, sendo que, como veremos, muitas das análises aqui realizadas podem ser feitas no ambiente disponibilizado pelo trabalho, sendo este bastante amigável.

Na Figura 6.1 temos a tela do Discriminador Neural Remoto de Partículas, que, por simplicidade, iremos chamar a partir de agora de DNRP. Esta tela pode ser simplesmente invocada em qualquer *Web Browser* com acesso à internet pelo endereço <http://www.lps.ufrj.br/~neural.w>. Para se utilizar o sistema deve-se inicialmente cadastrar uma senha através da própria interface gráfica. A apresentação do sistema é como se mostra nesta tela, uma janela na lateral esquerda traz opções que podem ser ativadas com o uso do *mouse* e uma janela maior na direita com campos para entrada de dados ou saídas das análises.

Vamos na primeira parte desta seção descrever a tecnologia empregada e as propriedades do sistema. Na segunda parte, iremos realizar uma análise passo a passo para demonstrar a potencialidade do sistema.

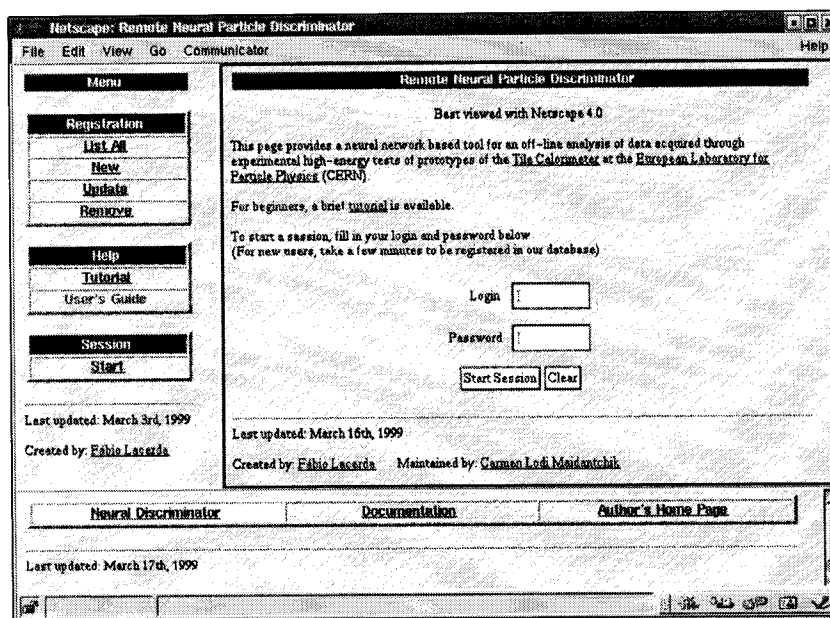


Figura 6.1: Tela inicial do Discriminador Neural Remoto de Partículas.

6.1 Tecnologia e Propriedades

O sistema DNRP usa páginas desenvolvidas com a linguagem HTTP que foi criada por pesquisadores do CERN para poder espalhar informações pela internet através de simples interfaces gráficas. A página inicial que vemos na Figura 6.1 é toda construída desta forma. Entretanto, alguns detalhes adicionais são importantes. Por exemplo, toda vez que são introduzidos dados através dos campos (neste caso, o nome e a senha), algumas verificações iniciais podem e devem ser realizadas. No caso, se um número muito pequeno ou muito grande de caracteres for introduzidos nestes campos, fora de uma faixa permitida, seria interessante avisar ao usuário do sistema sobre este problema antes mesmo de se enviar os dados através da internet. Isso poupa tráfego pela internet, e evita que o servidor trabalhe sobre dados seguramente incorretos, diminuindo a carga computacional sobre o mesmo. Estas análises iniciais são realizadas com a integração aos códigos HTTP, códigos baseados em linguagem JavaScript [67, 68], que são suportados por praticamente todos os *Web Browsers* atualmente. Em outros trechos, aonde devem ser introduzidos parâmetros para o treino da rede neural, os JavaScripts são usados para se verificar se os números introduzidos estão dentro da faixa esperada para os mesmos.

Outra tecnologia importante que teve de ser utilizada se relaciona com o fato

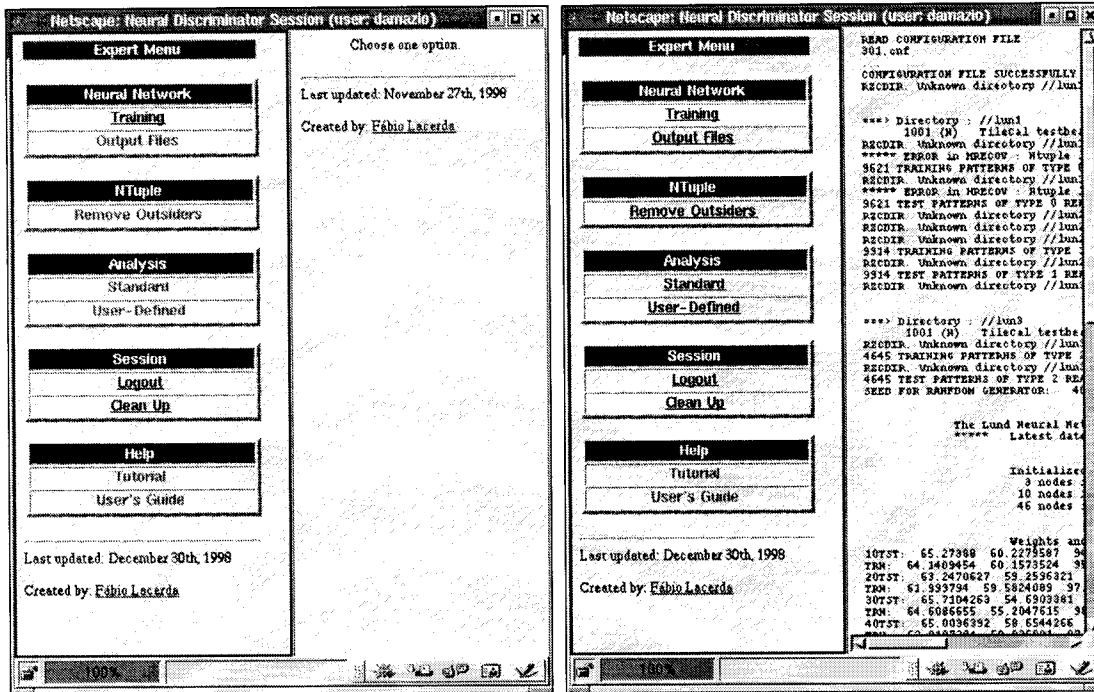


Figura 6.2: Somente algumas opções utilizáveis (esquerda); todas as opções utilizáveis (direita).

de que, dependendo do estágio da análise atingido por cada usuário, podem ser apresentadas a estes uma certa gama de opções. Assim, por exemplo, um usuário que já tenha realizado a fase de treino da rede neural tem mais dados que podem ser trabalhados pelas rotinas de análise, e, portanto, tem mais opções a serem exploradas do que um usuário que ainda não realizou este treino.

Para se realizar estas alterações das páginas *Web* em tempo de abertura das mesmas, foi utilizada a tecnologia de *Common Gateway Interface* [69, 70, 71], que chamaremos de CGI. As características fundamentais de cada página eram armazenadas em arquivos que foram nomeados por extensões ".src". Estes arquivos são lidos por programas CGI escritos em linguagem C++ quando as páginas são solicitadas pelos usuários. Assim, as páginas vão modificando as opções dependendo de condições que são verificadas pelos programas em C++.

Como exemplo, temos a Figura 6.2. Nesta figura vemos a esquerda uma página gerada para um usuário que não realizou ainda a operação de treino da rede. Vemos que as únicas opções possíveis (marcadas em negrito na parte esquerda da página) são as opções de treino (*Training*), de saída do sistema (*Logout*) e limpeza

dos resultados conseguidos até o momento (*Clean Up*). Na figura da direita temos a mesma página após o treino da rede, quando todas as outras opções aparecem marcadas. Estas novas opções são: *Output Files*, para se ter acesso a arquivos de saída; *Remove Outsiders*, para a remoção de eventos de contaminação; *Standard Analysis* para se realizar um conjunto de análises padrão conforme veremos mais tarde; *User-Defined Analysis* para se realizar uma análise definida pelo usuário.

Outro ponto importante é o programa de análise neural. Este programa, escrito em linguagem Fortran, é o mesmo que utilizamos para todas as nossas análises *offline* descritas até agora. Este programa usa as rotinas HBOOK para ler os arquivos gerados pelo sistema de aquisição do calorímetro e rotinas do pacote JetNet-2.0 para simular a rede neural.

Para se realizar as análises sobre os resultados do programa neural é utilizado um pacote de análises físicas, o *Physics Analysis Workstation*, PAW [72]. Este pacote também tem acesso aos arquivos HBOOK e pode fazer variados tipos de análises com os dados do calorímetro ou dados em arquivos texto, como são os dados produzidos pelo programa neural. Para se realizar estas análises devem ser utilizados scripts para o PAW. Alguns scripts, produzindo análises padrão já estão prontos, incluindo um script para descontaminação dos conjuntos de dados baseado na resposta da rede neural, mas o DNRP aceita que novos scripts sejam produzidos, dando total flexibilidade ao usuário. Cuidaremos melhor deste tópico quando fizermos um exemplo de análise.

6.1.1 Controle de Usuários

De forma a controlar o acesso do usuário aos dados que ele produziu, o sistema mantém uma base de dados para os usuários, constando do nome, senha, e um diretório aonde serão armazenados os dados. Assim, dados de usuários diferentes são armazenados em diretórios diferentes no servidor, para que não ocorra nenhuma mistura. Este sistema também possibilita que o usuário possa parar a análise que vem realizando num certo ponto e a retome mais tarde, sem perder os dados produzidos até o momento da parada.

A base de dados também define três diferentes níveis de usuários, o iniciante, avançado ou expert. Estes níveis foram estabelecidos para que o usuário possa

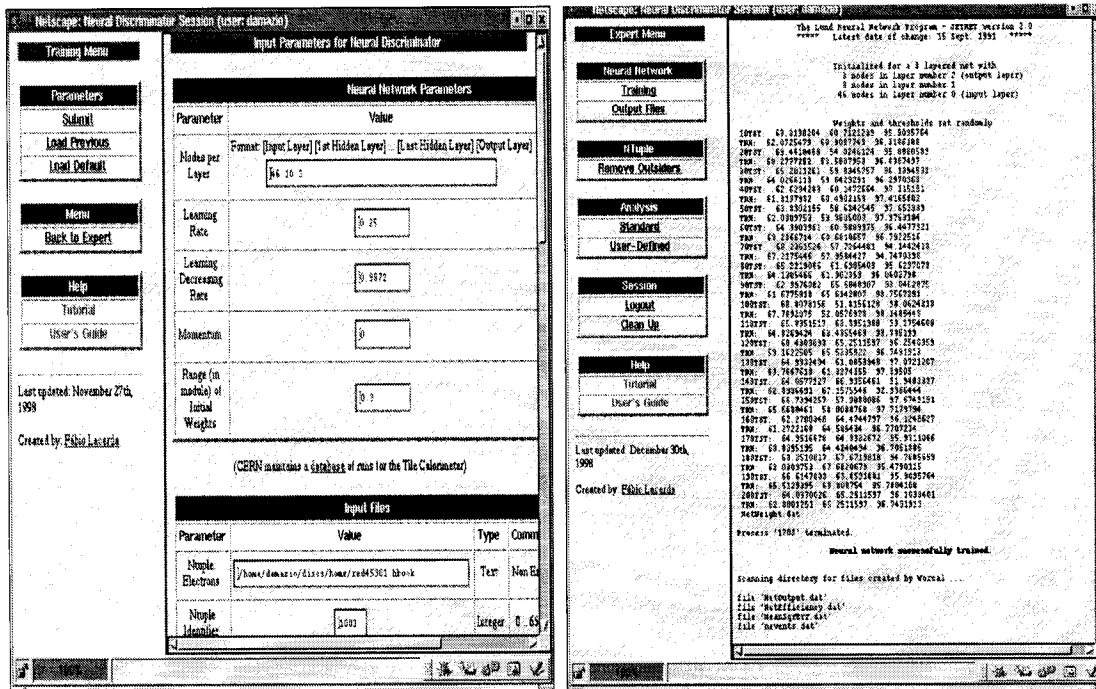


Figura 6.3: Janela para entrada de dados para treino da rede (esquerda) e janela após treino com opções de análise (direita).

ir realizando operações cada vez mais complexas conforme sua intimidade com o sistema for aumentando. O usuário iniciante pode realizar a análise neural com um conjunto predefinido de parâmetros, ver resultados das análises padrão e mesmo ter acesso aos gráficos gerados pelo PAW. O usuário avançado já tem um controle maior, podendo definir por si mesmo os parâmetros de treino da rede. O usuário expert já pode definir não só os parâmetros de treino da rede como também novas análises através da escrita de novos scripts para o PAW. Se forem gerados arquivos ou gráficos desta análise criada pelo usuário, estes podem ser visualizados ou descarregados para o computador do usuário.

Para podermos ter uma melhor compreensão do que foi explicado até agora, realizaremos uma análise de exemplo.

6.2 Exemplo de Análise

Para podermos realizar nossa análise exemplo, vamos utilizar o nível expert, de forma a ter acesso a toda a potencialidade do sistema.

A primeira coisa importante é o treino da rede neural. Na Figura 6.3 temos

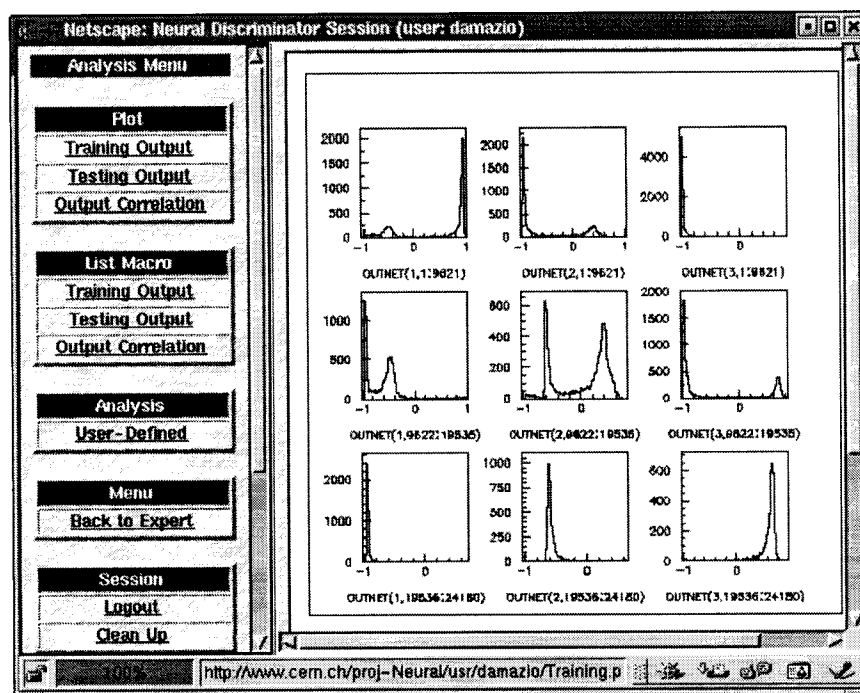


Figura 6.4: Saídas da rede neural para cada conjunto de dados produzida pelo DNRP.

os detalhes necessários para este tópico. Na figura da esquerda, temos a entrada de parâmetros para a rede neural. Os campos relativos a alguns destes parâmetros aparecem na figura, tais como número de neurônios da rede para cada camada, a taxa de aprendizado (*Learning Rate*), etc. Além destes parâmetros, podemos (e devemos) definir os arquivos com os dados do calorímetro. Inseridos estes parâmetros, pode-se selecionar o *link Submit* no canto esquerdo superior que o treino da rede neural começa a ser realizado. Na figura da direita da Figura 6.3, a rede neural já foi treinada. Vemos para cada passo de teste realizado as eficiências obtidas em cada conjunto de dados (elétron, pión e múons) para treino e teste. Ao fim do treino a mensagem *Neural Network successfully trained* indica que o processamento foi realizado perfeitamente. Vemos também que o Menu de opções mudou automaticamente para as opções de análise, para trabalharmos com os dados novos gerados pela rede.

6.2.1 Realizando Análises Pré-definidas com o DNRP

Usando a opção *Standard Analysis*, temos várias análises padrão que iremos discutir a partir de agora. Na Figura 6.4, temos a saída de uma das análises predefi-

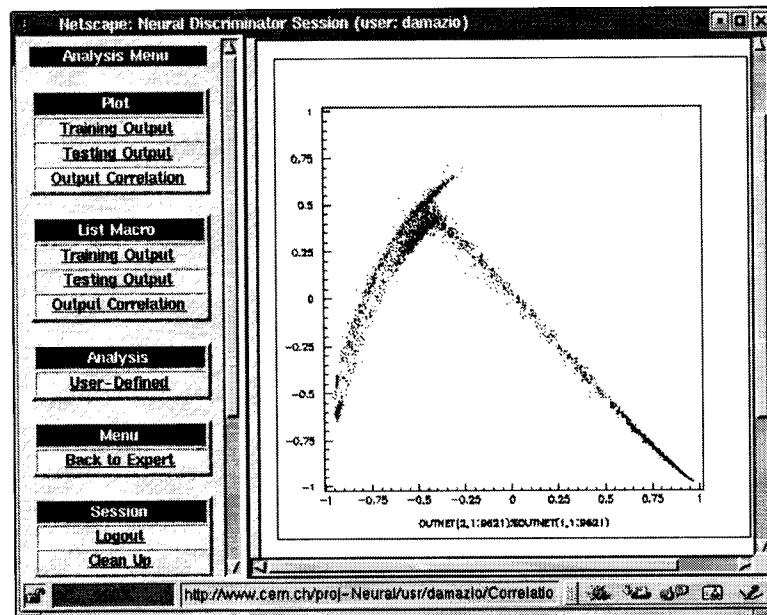


Figura 6.5: Correlação entre as saídas de elétrons e píons no conjunto de elétrons.

nidas. No canto esquerdo, podemos ver que a opção *Training Output* foi usada. Esta opção gerou a figura que vemos na parte direita da figura. Estes histogramas são do mesmo tipo que os que vimos durante a análise *offline*. As três colunas de histogramas se relacionam com as três saídas da rede neural e as três linhas se relacionam com os conjuntos de dados. Esta figura, pode ser carregada para o computador em que estiver o usuário em formato *PostScript*, podendo ser utilizada em textos e artigos.

Conforme fizemos anteriormente, uma outra análise importante é a correlação entre as saídas da rede para demonstrar a coerência interna da rede neural. Se fizermos a correlação entre as saídas na opção *Output Correlation*, podemos encontrar a Figura 6.5.

Nesta figura temos a correlação entre as saídas de elétrons e píons para o conjunto de elétrons. Da mesma forma que antes, podemos ver que uma parte dos eventos recebeu valores positivos pela saída de elétrons (eixo horizontal) e valores negativos na saída de píons (eixo vertical), logo permitindo que se conclua que os eventos rejeitados pela saída de píons são os mesmos aceitos pela saída de elétrons. Por outro lado, uma parte dos eventos foi rejeitada pela saída de elétrons e aceita pela saída de píons, logo representando uma possível contaminação de píons no conjunto de elétron. Uma parte dos eventos foi rejeitada por ambas as saídas, tendo

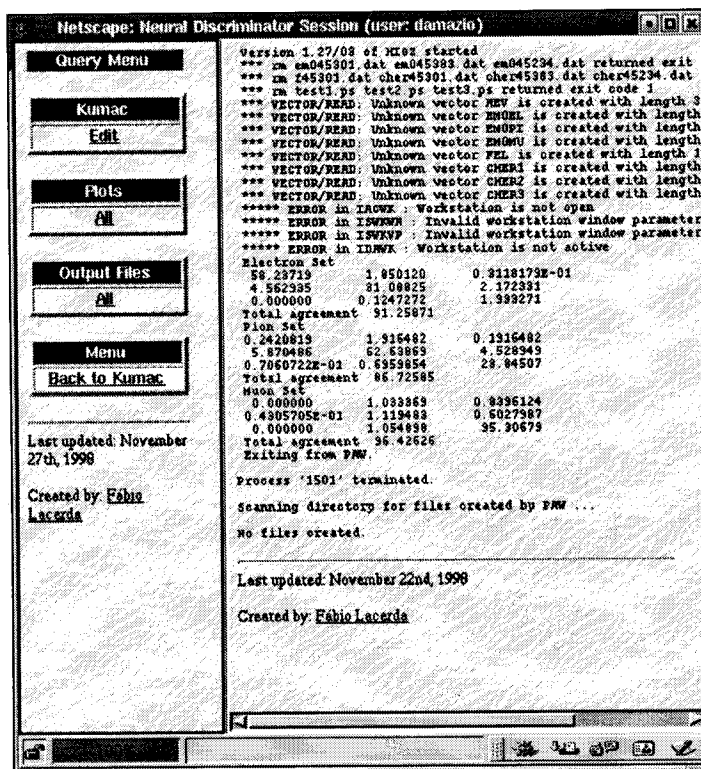


Figura 6.6: Resultados de uma análise de comparação com método clássico realizada pelo usuário.

valores negativos e constituindo a contaminação de múons.

6.2.2 Criando sua Própria Análise

Estes foram dois exemplos de análises predefinidas que podemos utilizar diretamente do sistema. Podemos também realizar nossas próprias análises, bastando para isso algum conhecimento para preparar os scripts para o PAW. Seguindo os exemplos dos scripts já prontos e incluídos na ferramenta, fica bastante fácil progredir nesta tarefa. Fizemos um pequeno exemplo, em que, usando trechos de programa em linguagem Fortran (o PAW interpreta pequenos trechos nesta linguagem), realizamos toda a análise de comparação entre métodos clássicos (baseado em corte de energia e no contador Cherenkov) para cada conjunto de dados (podemos ver *Electron set*, *Pion set* e *Muon set*). Na Figura 6.6, temos a tela de saída desta análise.

Nesta figura, podemos ver as tabelas de comparação entre o método clássico e o método neural com o resultado final de concordância para cada conjunto de dados

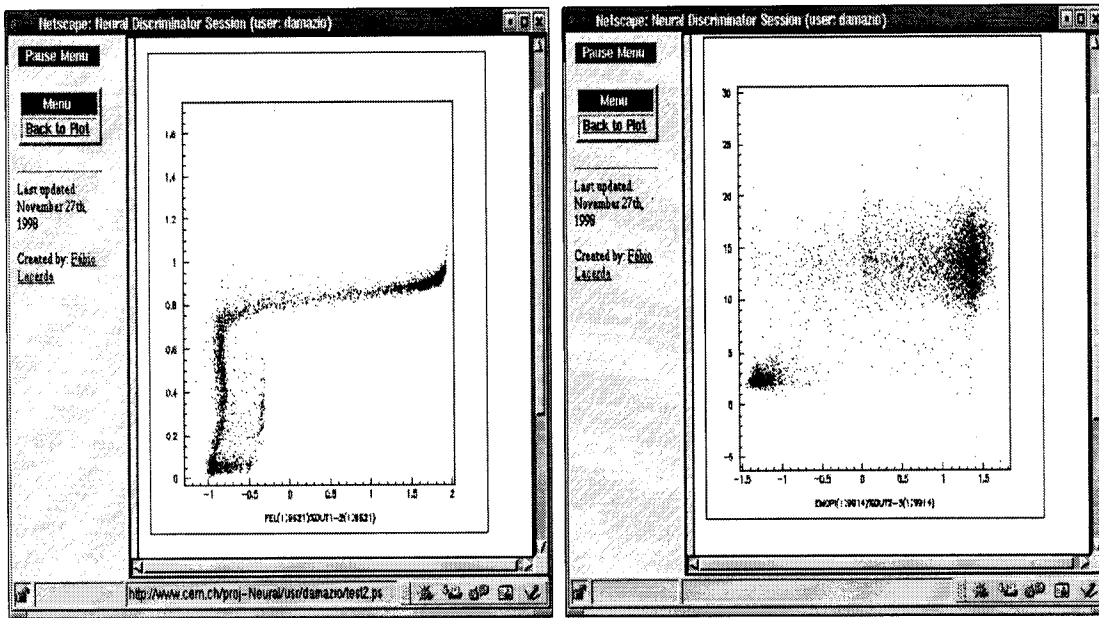


Figura 6.7: Figuras resultantes de uma análise do usuário. Correlação entre saída de elétrons menos a de píons com a fração de energia na primeira camada para o conjunto de elétrons (esquerda), e correlação entre saída de píons menos a de múons com a energia total para o conjunto de píons (direita).

estudado. Estes resultados são os mesmos que estivemos explorando nas análises *offline*, e também mostram que o sistema neural está perfeitamente identificando as contaminações. Esta análise gerou também algumas figuras usando a parte gráfica do PAW. O sistema reconhece automaticamente estes arquivos e, usando a opção *Plots All* que podemos ver no canto esquerdo da tela, podemos ter acesso a estas figuras.

As figuras resultantes desta análise podem ser vistas na Figura 6.7. Na figura da esquerda temos a correlação da saída de elétrons menos a de píons da rede neural contra a fração da energia depositada na primeira camada de células do detector para o conjunto de elétrons contaminado. Esta figura comprova de forma qualitativa que os eventos identificados pela rede como sendo eventos de elétrons tem um alto nível de deposição de energia na primeira camada (acima de 75%), enquanto os outros eventos deste mesmo conjunto, identificados como sendo píons pela rede neural depositam menos energia na primeira camada. Isso comprova que a rede neural está corretamente identificando esta contaminação.

Na figura da direita, temos a correlação entre a saída de píons menos a de

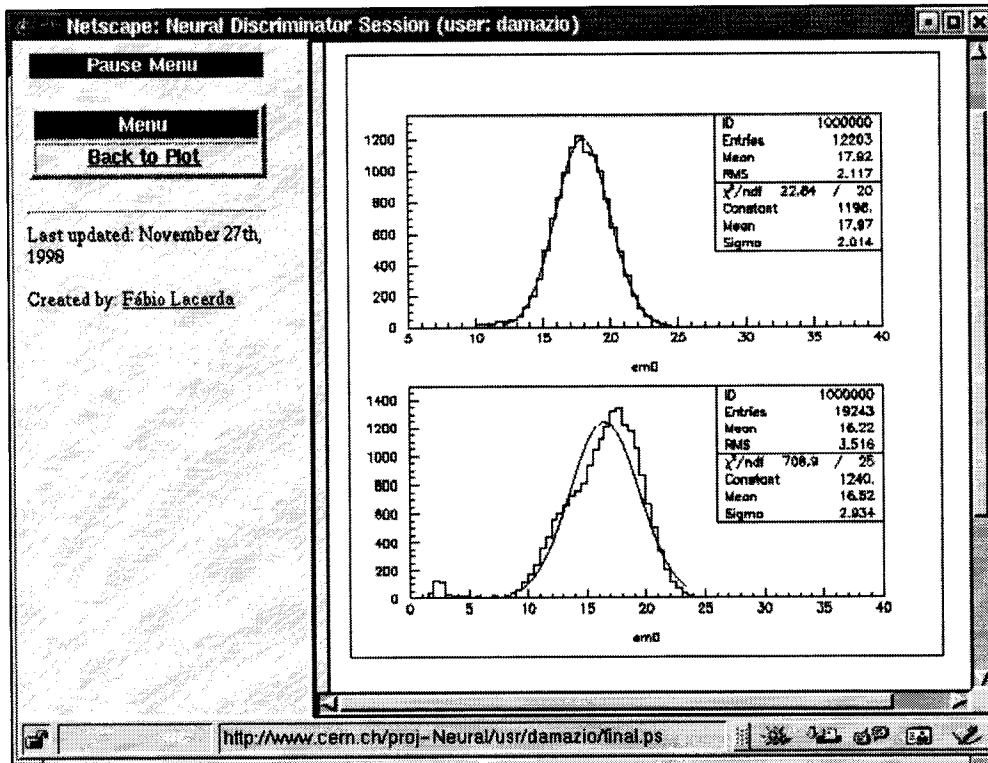


Figura 6.8: Análise de comparação entre o conjunto de dados descontaminado (em cima) e conjunto contaminado (embaixo).

múons da rede contra o valor de energia depositado no calorímetro inteiro para o conjunto píons. Esta correlação comprova que os eventos de múons que estão contaminando o conjunto de píons (e que depositam pouca energia no detector) estão sendo corretamente isolados pela rede neural que produz valor negativos da subtração de suas saídas para os eventos de pouca energia (múons). Estas figuras estão disponíveis para qualquer usuário que queira fazer estas mesmas análises.

Uma das análises padrão que ainda não discutimos e que é também importante é a eliminação de contaminações através do resultado da rede neural. É possível se gerar um novo conjunto de dados sem nenhuma contaminação de dados através de um outro script do PAW já definido. Após a retirada desta contaminação com um clique do *mouse*, temos um novo conjunto de dados que pode ser acessado pelo usuário. É interessante se verificar as propriedades do conjunto de dados que ficou. Assim, sobre este novo conjunto de dados (no caso o conjunto de elétrons, do qual retiramos as contaminações de píons e múons), realizamos outra análise definida pela usuário.

Na Figura 6.8, temos a energia total depositada no calorímetro pelos eventos de elétrons. No histograma inferior temos todos os eventos (incluindo as contaminações identificadas pelo método neural), e, na figura superior temos apenas os eventos selecionados como sendo elétrons pela rede. Elétrons e píons isoladamente tendem a depositar sua energia no detector no formato de uma Gaussiana. Assim, podemos realizar um processo de *fitting* para estimar os parâmetros desta Gaussiana. A qualidade do *fitting* realizado é medida pelo fator chamado qui-quadrado. Quanto mais próximo o qui-quadrado for de 1, melhor foi o *fitting* realizado, e, isso significa que os dados podem ser realmente representados pela curva teórica (no caso a Gaussiana). No caso dos eventos selecionados pela rede neural como sendo elétrons puros, o qui-quadrado foi 1.14. Enquanto isso para todos os eventos (com contaminações), mesmo selecionando apenas os canais do histograma do pico de mais alta energia, temos um qui-quadrado de 28.35, mostrando que esse segundo conjunto de dados não é descrito por uma Gaussiana (por conter, na verdade, uma mistura de elétrons e píons representadas por curvas Gaussiana com parâmetros diferentes).

Capítulo 7

Pré-processamento e Compactação de Sinais

Neste capítulo iremos investigar alguns usos especiais de rede neurais aplicadas ao problema de calorímetros da física experimental. Na primeira parte deste capítulo, estudaremos os possíveis mapeamentos topológicos que podem ser realizados com as informações vindas do calorímetro. Esta análise visa aproveitar características básicas do perfil de deposição de energia da partícula incidente no calorímetro como apoio ao processamento neural.

Na segunda parte desta seção, iremos investigar o uso de redes especialistas, tentando incluir alguma informação extra em nossa análise vinda dos calorímetro auxiliares. Adiante, estudaremos o uso de Componentes Principais de Discriminação no nosso problema, o que ajuda a definir uma rede de dimensões menores (e conseqüentemente de processamento mais rápido) para realizar a mesma discriminação.

7.1 Mapeamentos Topológicos

Nesta seção, iremos estudar formas de compactar a informação do calorímetro, agrupando as informações fornecidas pela leitura das células do detector. Assim, a forma de se agrupar a informação explora o conhecimento físico especialista do modo pelo qual uma partícula de uma dada classe deposita energia no calorímetro. Ou seja, aproveita-se o conhecimento que se tem do perfil de deposição de energia para se definir um mapeamento de células que melhor se adequa à topologia do processo

físico de deposição de energia no calorímetro.

As vantagens de se encontrar um tal mapeamento eficiente são basicamente de dois aspectos. Primeiro, agrupando-se topologicamente as células do calorímetro, reduzimos a dimensão do espaço de entrada de dados, o que pode tanto reduzir a complexidade do sistema neural de identificação de partículas, como até mesmo melhorar a eficiência de discriminação, já que a distribuição dos sinais de entrada tende a ser menos esparsa no espaço compactado, além do processo de otimização requerido pela fase de treinamento ser facilitado.

Uma segunda vantagem é em termos da velocidade de processamento. Com a dimensão do espaço de entrada reduzida pelo pré-processamento topológico, o processamento tende a ser mais veloz, o que pode ser essencial numa aplicação *online*.

7.1.1 Média da Energia na Célula

A primeira forma mais simples de se agrupar as leituras das células é reduzir a uma única leitura a dupla leitura das células, através de algum processo. A mais simples forma de se fazer isso é pela média aritmética das leituras. Assim, as novas informações que a rede passa a receber são :

$$E_i = (E_{2i} + E_{2i+1})/2$$

Este pré-processamento já reduz para aproximadamente a metade o número de entradas da rede de 46 para 23. Por exemplo, para uma rede de discriminação entre píons e múons, esta redução faz com que o número de pesos seja reduzido de 376 (46 vezes 8 mais 8 vezes 1) para 192 (23 vezes 8 mais 8 vezes 1), representando uma significativa diferença no tempo de treino.

Sobre este novo vetor de entrada, podemos aplicar as normalizações que discutimos no capítulo 5, como, por exemplo, a normalização pela soma de todas as componentes da entrada ou pela raiz quadrada desta soma. Na Tabela 7.1 temos os níveis de concordância com a metodologia clássica (corte em energia) da média da dupla leitura com ambos enfoques de normalização na separação pión-múon.

Podemos ver, se compararmos a tabela da esquerda com a Tabela 5.6, que em todos os resultados globais houve uma melhora nas concordâncias entre os métodos clássico e neural. Podemos assumir que a compactação das informações reduz o

E (GeV)	π	μ	tot	E (GeV)	π	μ	tot
20	96,86%	96,77%	96,81%	20	98,93%	98,73%	98,83%
100	97,37%	94,12%	95,73%	100	99,97%	99,85%	99,91%
180	99,17%	97,63%	98,56%	180	99,94%	99,63%	99,78%

Tabela 7.1: Concordância com a metodologia clássica da energia média por célula (23 componentes) para a normalização pela energia total (esquerda) e pela raiz quadrada da energia total(direita) considerando-se fixos de píons contaminados por múons.

nível de ruído existente em cada uma das entradas da rede neural, ao produzir a média de duas entradas diferentes (e ruidosas). Podemos também comparar a tabela da direita que traz os resultados da normalização pela raiz quadrada com a Tabela 5.9. Nota-se que os resultados globais são ligeiramente melhores para o sistema compactado (23 entradas).

Na Figura 7.1, temos a saída da rede contra a energia no detector para o caso 20 GeV, quando a normalização pela raiz quadrada da energia total é utilizada. Se compararmos com a Figura 5.17, veremos que praticamente não existe diferença entre as duas figuras, mostrando que é razoável se utilizar esta compactação de informação.

Na Tabela 7.2, temos as concordâncias para o caso elétron-píon-múon. Se compararmos estes resultados com aqueles da Tabela 5.6 e da Tabela 5.9, veremos que ora há um ganho para o método compactado, ora este apresenta uma queda de eficiência em relação ao uso da informação não compactada. A maior diferença em termos de percentuais é de apenas 2,12%, logo ficando difícil eleger um dos dois sistemas como sendo o melhor. Podemos então pensar em termos do aumento de velocidade do método compactado como critério de diferenciação e tendermos a optar por este numa aplicação *online*.

7.1.2 Torres de *Trigger*

O segundo método de compactação que estudaremos é o das torres de *trigger*. Conforme havíamos explicado na Seção 3.1.1, as torres de *trigger* são direções

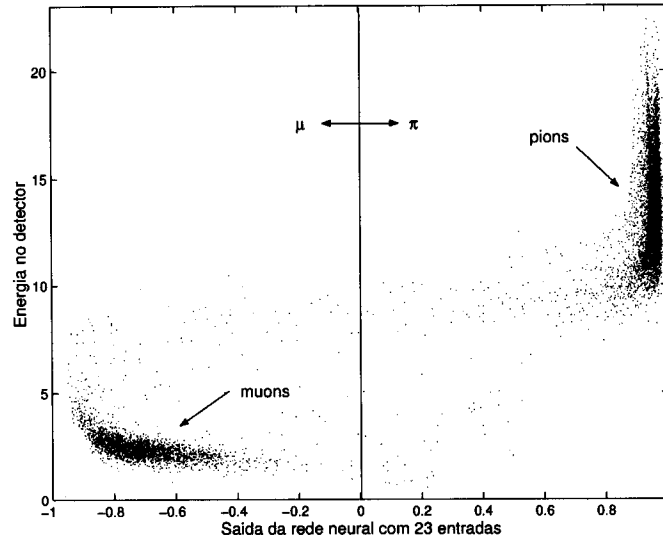


Figura 7.1: Saída da rede neural contra a energia absorvida no detector para um feixe de píons de 20 GeV usando 23 entradas com normalização pela raiz quadrada da energia total.

E (GeV)	e	π	μ	tot
20	90,22%	84,70%	96,65%	90,39%
100	93,76%	92,73%	95,80%	94,09%
180	95,78%	93,93%	98,04%	95,90%

E (GeV)	e	π	μ	tot
20	94,65%	91,98%	97,25%	94,60%
100	95,31%	96,41%	97,33%	96,35%
180	96,50%	95,19%	99,46%	97,03%

Tabela 7.2: Concordância entre o método clássico e o método neural para o caso elétron/píon/múon para diferentes energias quando usada a média da energia por célula. Normalização pela energia total (acima) e pela raiz quadrada da energia (abaixo).

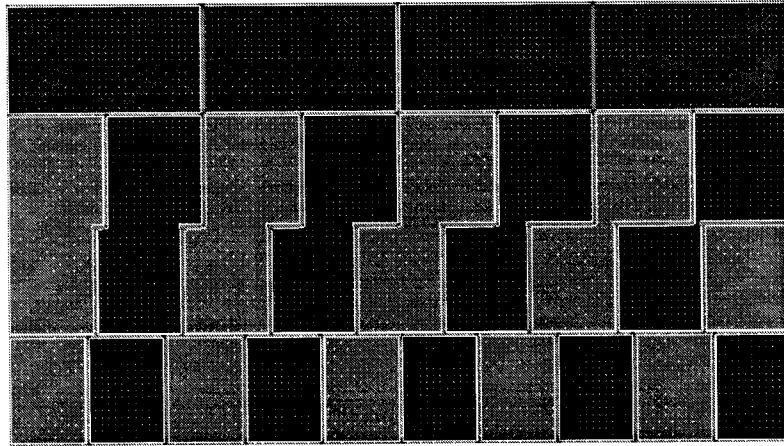


Figura 7.2: Definição das torres do TileCal. As células da última camada tem cada uma de suas leituras associadas a uma das duas torres que nela tocam.

ótimas de detecção de partículas que provenham do ponto de colisão relativo ao ATLAS. Estas direções seguem em ângulos, os quais se encontram bem definidos na estrutura de detector. Nos módulos do calorímetro, as torres estão definidas conforme a Figura 7.2. As direções estão marcadas com tons diferentes para facilitar a visualização. As células da última camada têm sua dupla leitura dividida entre torres diferentes, por isso seu tom no desenho é diferente das demais.

Os sinais das 10 torres existentes no Módulo 0 foram utilizados como entrada para o sistema neural. As torres foram normalizadas pela raiz quadrada da soma das mesmas (ou seja, pela raiz quadrada do valor de energia depositado em todas as células). Assim, temos uma redução drástica do conjunto de entrada de 46 para apenas 10 variáveis.

Os resultados da concordância entre o método neural aplicado às torres e o método clássico estão sumarizados na Tabela 7.3. Como podemos ver, praticamente todos os resultados estão no nível das melhores análises feitas até agora. A única exceção seria a análise elétron/píon/múon para o caso de 20 GeV. Neste caso, os resultados de elétrons e píons ficaram bem piores do que aqueles obtidos pela normalização através da raiz quadrada da energia (23 ou 46 entradas). A explicação para este fato pode estar no problema de ser a discriminação entre estas duas partículas extremamente dependente da distribuição de energia entre as células da mesma torre, no caso, a torre em que as partículas estão incidindo, o que perdemos ao utilizar

E (GeV)	π	μ	tot
20	99,46%	99,35%	99,40%
100	99,18%	97,63%	98,40%
180	99,78%	99,57%	99,67%

E (GeV)	e	π	μ	tot
20	92,68%	89,69%	96,98%	93,07%
100	97,55%	97,62%	97,80%	97,66%
180	97,70%	94,42%	99,48%	97,18%

Tabela 7.3: Concordância entre o método clássico e o método neural usando as torres de *trigger*. Acima, temos os resultados por energia para a separação pión/múon. Em baixo, temos a separação elétron/pión/múon.

este sistema de compactação.

Para verificarmos quais informações estão sendo mais utilizadas pela rede, vemos na figura a esquerda da Figura 7.3 as relevâncias para 20 (em cima) e 100 GeV (embaixo). Como podemos ver, nestas figuras a relevância da torre número três, que é a torre para qual o feixe foi direcionado, é bem maior que do que a relevância das outras torres, tanto pra 20 quanto pra 100 GeV. Vemos também que as torres laterais à torre principal também têm uma certa importância na discriminação. Isso se dá porque os eventos de píons vazam da torre principal (posto que sua cascata é mais larga), tocando um pouco as torres laterais. Sendo assim, o toque nas torres laterais é um fator identificador de píons. Vemos também na figura, que, a 20 GeV, a importância relativa da torre principal em relação às torres laterais é bem menor do que a 100 GeV. A razão da relevância da torre principal contra a soma das relevâncias das duas torres laterais no caso 100 GeV é cinco vezes maior do que a mesma razão a 20 GeV.

Uma possível explicação para este fato pode ser obtida no gráfico à direita da Figura 7.3, onde temos a soma do sinal na segunda e quarta torres (torres laterais à principal) para 20 GeV (em cima) e 100 GeV (embaixo) para os três feixes de partículas estudados. Como podemos ver a 20 GeV, não há praticamente nenhuma separação visível entre os conjuntos, parecendo o histograma de um único de

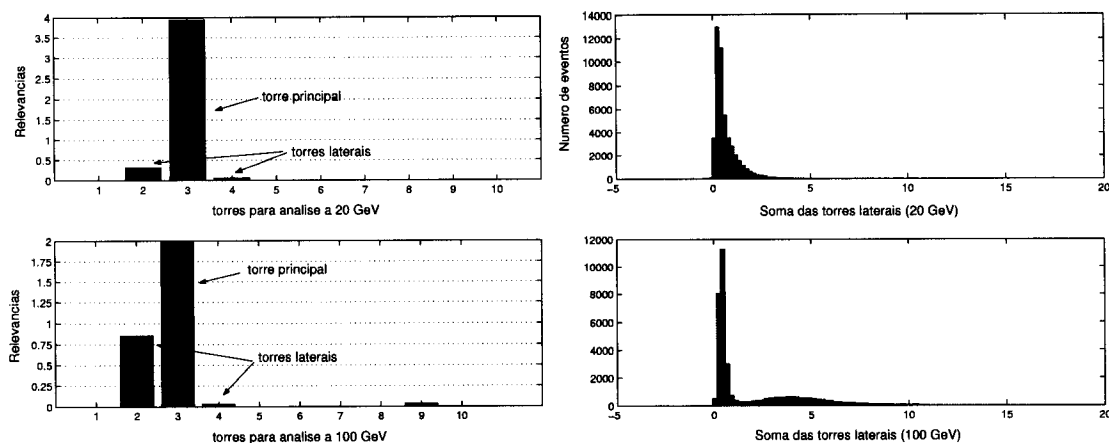


Figura 7.3: Relevâncias para a análise com torres de *trigger* a 20 e 100 GeV (esquerda). Quantidade de energia absorvida pelas torres laterais para 20 e 100 GeV (direita). Em ambos os casos, os três feixes de partículas foram utilizados.

dados. A 100 GeV, vemos um pico próximo ao zero (relativo a elétrons e múons) e um outro pico com mais energia ligeiramente acima. Isto ocorre porque, como dissemos, a cascata de píons a 100 GeV se espalha pelas outras torres, ajudando a separar píons das outras partículas, mas, a 20 GeV, nem os eventos de píons conseguem desenvolver uma cascata suficiente para vazar da torre principal. Por isso não vemos separabilidade no sinal das torres a 20 GeV. A rede reconhecendo esse problema dá menos importância às torres laterais no caso 20 GeV. Como a informação principal que sobra para se realizar a separação a 20 GeV é a informação nas células, demonstrando a profundidade da interação da partícula, e esta é mascarada pela soma das células na torre, também a eficiência para este caso cai.

7.1.2.1 Método Híbrido

De forma a se tentar descobrir uma metodologia alternativa para resolver esse problema mantendo o nível de compactação alto, uma outra metodologia de compactação pode ser proposta. Mantemos a estrutura de torres, exceto na torre principal. Nesta torre usamos as células individuais. Assim, mantemos a redução de dimensionalidade na entrada da rede em um bom nível de 46 para 14 (9 torres e 5 células da torre principal). Este método alternativo fornece melhores resultados do que as torres como podemos ver na Tabela 7.4.

Os resultados apresentados nesta tabela são comparáveis aos obtidos com

E (GeV)	π	μ	tot
20	98,22%	98,69%	98,45%
100	99,50%	99,19%	99,34%
180	99,88%	99,66%	99,77%

E (GeV)	e	π	μ	tot
20	95,60%	93,83%	97,57%	95,65%
100	95,94%	96,12%	99,73%	97,25%
180	96,98%	93,66%	99,48%	96,68%

Tabela 7.4: Eficiências para sistema neural com pré-processamento híbrido com torres e células. Acima, temos o caso pión/múon e abaixo o caso elétron/pión/múon para as energias consideradas.

a raiz quadrada (com 23 ou 46 entradas). Para verificarmos se foi feita uma boa escolha de variáveis para esta rede neural, temos na Figura 7.4 as relevâncias para as entradas da rede neural. Vemos que as células da torre principal tem um alto nível de relevância, enquanto as torres laterais, que agora acrescentam pouco a esta discriminação, aparecem com uma pequena relevância.

Com isso encerramos as análises de torres e do sistema híbrido. Podemos, entretanto, continuar nossa análise com diferentes métodos de compactação.

7.1.3 Camadas de Segmentação

O método que veremos agora é o que causa maior compactação. Este método consiste em somarmos todas as células de cada camada. Os métodos clássicos são baseados, em parte, na distribuição de energia entre as camadas, logo, é de se esperar que este método possa ter um certo sucesso na análise neural. Na Figura 7.5 temos a definição das camadas de célula.

Os valores de energia das camadas são normalizados pela raiz quadrada da soma destes mesmos valores. Este método reduz de 46 entradas da rede neural para apenas três entradas, representando uma absurda redução no espaço de dados.

Nas Tabelas 7.5 temos os resultados da comparação entre o método clássico

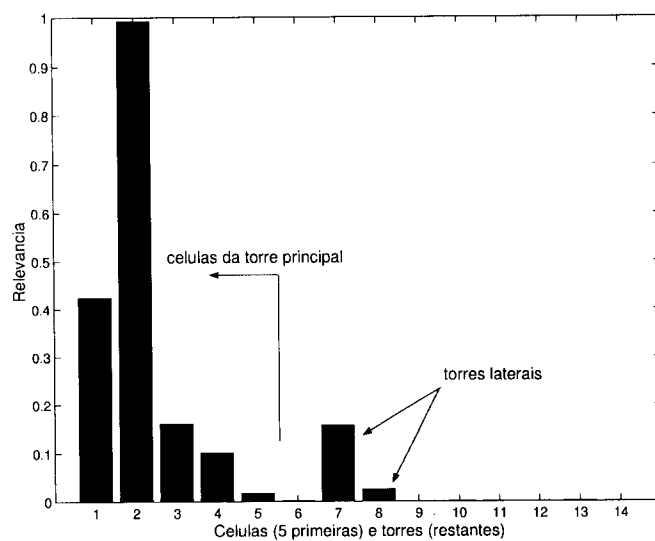


Figura 7.4: Relevância para as entradas da rede neural na separação elétron-píon-múon a 20 GeV.

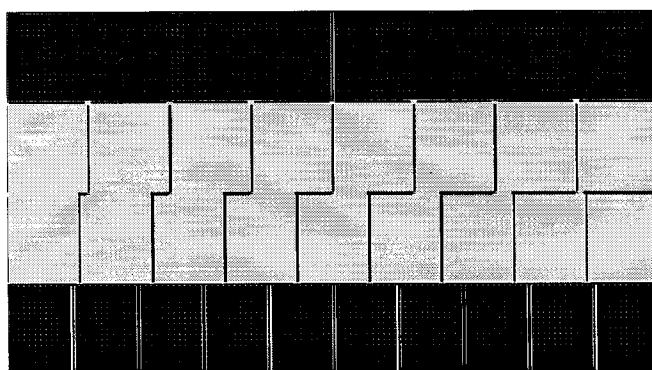


Figura 7.5: Definição das camadas para análise neural.

E (GeV)	π	μ	tot
20	99,13%	99,44%	99,28%
100	99,97%	99,85%	99,91%
180	99,78%	99,27%	99,52%

E (GeV)	e	π	μ	tot
20	92,77%	87,90%	97,59%	92,67%
100	94,46%	96,81%	99,12%	96,78%
180	96,64%	94,49%	99,54%	96,87%

Tabela 7.5: Concordância entre método clássico e processamento neural para análise por camadas. Acima, a análise pión/múon. Abaixo, análise elétron/pión/múon.

e as rede neurais para diferentes energias. Como podemos verificar facilmente, este método é extremamente satisfatório, sendo comparável à análise com as 46 células com a normalização pela raiz quadrada. O único ponto fraco foi a 20 GeV na análise elétron-pión-múon. Para os conjuntos de elétrons e píons, a definição exata do ponto de divisão entre um tipo e outro é mais confuso (lembremo-nos que a 20 GeV, temos as cascatas de píons e de elétrons muito concentradas na primeira camada), precisando, de informações mais detalhadas (uma maior granularidade das informações) para realizar bem a separação.

7.1.4 Anéis Concêntricos

O último método, ou melhor, grupo de métodos conforme veremos, é a análise por anéis de energia. Esta análise se baseia na tentativa de montar um sistema que tenha um certo nível de casamento com o sinal esperado. Assim, como sabemos que um sinal de elétrons tem, em geral, a maior parte de sua energia no ponto de choque da partícula e que esta energia decresce rapidamente conforme nos afastamos do ponto de colisão, podemos criar anéis de células, sendo o primeiro anel exatamente este ponto com mais energia, o segundo, as células em torno do ponto mais forte, depois as células em volta deste segundo anel e assim por diante.

Para píons também podemos fazer algo semelhante, só que utilizando anéis

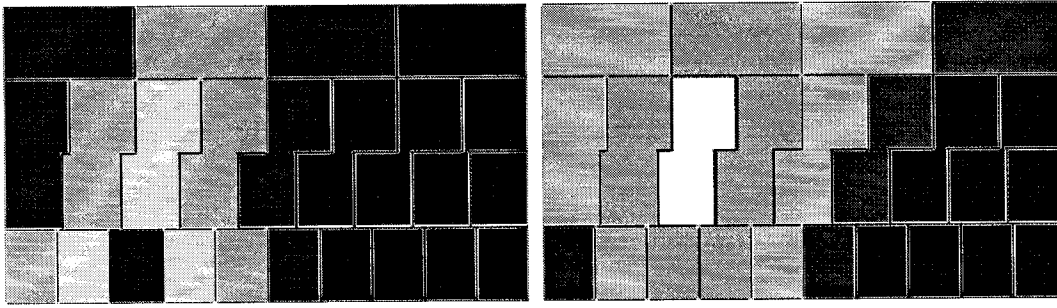


Figura 7.6: Definição das estruturas em anel para elétrons (esquerda) e píons (direita).

que comecem na segunda camada do calorímetro, posto que é aí que eventos desta espécie de partícula costumam deixar o máximo de sua energia. Na Figura 7.6 temos as estruturas de anéis para elétrons e píons. Na estrutura de elétrons o primeiro anel se situa na terceira célula (da esquerda para a direita) da primeira camada. O segundo anel, em tom mais claro, circunda este ponto e o terceiro circunda o segundo. O último anel considerado é aberto, consistindo de duas células da extrema esquerda superior e algumas células do lado direito do ponto de colisão. As células na direita de tom mais escuro não são utilizadas por praticamente não possuem sinal. Assim, temos a redução de 46 para apenas 4 no vetor de entrada.

Na figura direita da Figura 7.6, temos a estrutura em anéis otimizada para eventos de píons. A célula central em cor mais clara é o ponto de máximo e primeiro anel. Vemos os outros anéis circundando este até termos também o primeiro anel aberto. Também foi utilizada a normalização pela raiz quadrada da soma dos valores dos anéis para esta análise.

Implementamos assim, três análises separadas, uma utilizando os anéis de elétrons (4 entradas da rede neural), outra utilizando os anéis de píons (também 4 entradas), e, finalmente, uma última análise em que foram utilizados ambos os anéis como entrada (8 entradas). Esta última visava deixar a rede com entradas redundantes, de forma que esta última pudesse, por si só, selecionar quais eram as informações que deveriam ser utilizadas e fossem importantes para realizar a discriminação. Não temos anéis definidos na estrutura de múons, posto que estas partículas tendem a depositar sua energia segundo a torre em que incide, não havendo como interpretar esta deposição como um anel.

E (GeV)	π	μ	tot	E (GeV)	π	μ	tot
20	99,18%	99,03%	99,10%	20	98,74%	98,97%	98,85%
100	99,87%	99,63%	99,75%	100	99,70%	99,68%	99,69%
180	99,80%	99,44%	99,62%	180	100,0%	99,78%	99,89%

E (GeV)	π	μ	tot
20	99,49%	99,61%	99,55%
100	99,72%	99,48%	99,60%
180	99,92%	99,54%	99,73%

Tabela 7.6: Comparação entre os métodos baseados em anéis para redes neurais e a metodologia clássica. Temos anéis otimizados para eventos de elétrons (alto esquerda), píons (alto direita) e ambos (embaixo).

Assim sendo, para estas três diferentes análises temos, nas Tabelas 7.6 os resultados destas três análises para a separação pión-múon.

Como podemos observar da tabela, todos os resultados estão muito próximos do que poderíamos chamar de acerto absoluto, ou seja, todos os resultados estão acima de 98,74% o que é uma marca bastante forte para os métodos neurais.

Na Tabela 7.7 temos os resultados das mesmas análises para o caso elétron-pión-múon. Também temos resultados bastante confiáveis, com destaque para o caso 20 GeV, aonde, apesar dos anéis individuais terem deixado bastante a desejar principalmente para eventos de píons, parece que o uso de ambas estruturas de anéis favoreceu bastante o Discriminador que conseguiu melhorar tanto individualmente em cada conjunto como no total.

A detecção de píons a 180 GeV parece ter sido um pouco prejudicada em todas as análises (sendo melhor exatamente nos anéis para píons). Uma possível explicação para este fato é que eventos de píons a 180 GeV interagem mais perto ainda do fim do calorímetro, o que pode fazer com que os anéis centrados na segunda camada não sejam sempre os ótimos para detectar este tipo de partícula.

E (GeV)	e	π	μ	tot
20	93,23%	90,33%	97,82%	93,74%
100	96,13%	97,50%	99,53%	97,71%
180	96,68%	90,77%	99,53%	95,59%

E (GeV)	e	π	μ	tot
20	92,22%	89,20%	98,04%	93,08%
100	93,42%	95,43%	98,66%	95,81%
180	95,06%	93,60%	99,05%	95,88%

E (GeV)	e	π	μ	tot
20	93,62%	93,52%	98,10%	95,06%
100	95,73%	95,91%	99,39%	97,00%
180	96,46%	91,60%	99,33%	95,74%

Tabela 7.7: Comparação entre os métodos clássicos e os métodos neurais baseados em pré-processamento pelo sistema de anéis de células no caso elétron/píon/múon. No topo, anéis otimizados para elétrons. No meio, anéis otimizados para píons. Em baixo, ambos os anéis.

7.1.5 Sumário

Agora, que acabamos de explorar todos os métodos topológicos que foram propostos para este trabalho. Apenas a título de recordação e conclusão, temos a Tabela 7.8, em que resumimos todos os resultados totais das análises baseadas em mapeamentos topológicos realizadas para cada energia.

Para a análise píon-múon (tabela superior), vemos que, excetuando-se o caso de normalização pela soma, todos os métodos atingiram um alto nível de eficiência em todas as energia. Para a análise elétron-píon-múon, (tabela inferior), a variação é maior, mas podemos concluir, que, de novo excetuando-se a normalização pela soma, todos os métodos são bastante eficientes, havendo alguma variação dependendo da faixa de energia considerada.

7.2 Redes Especialistas

Conforme havíamos dito anteriormente, uma parte da energia das cascatas não consegue ficar contida no interior de um único módulo do calorímetro hadrônico, posto que o mesmo é fino para suportar toda a dimensão da cascata de partículas. Assim, durante os testes experimentais utilizamos os cinco módulos do protótipo anterior na esperança de conter completamente a cascata. Tipicamente, o problema maior aparece em cascatas de píons, que tendem a se expandir além dos limites do módulo. Assim, vemos na Figura 7.7 a energia total depositada nos cinco módulos por eventos de elétrons (aonde os múons de contaminação foram retirados através de corte em energia e a contaminação por píons não foi retirada) no histograma superior. No histograma inferior temos os eventos de píons (também sem contaminação de múons).

Como podemos ver, existe uma estrutura que pode ser descrita por uma Gaussiana na parte de mais baixa energia no histograma superior. Relacionamos estes eventos com os elétrons presentes no feixe, dado que as cascatas de elétrons não atingem estes módulos. No histograma inferior, temos uma Gaussiana descrevendo os eventos na parte de mais alta energia do histograma. Estes são os eventos de píons, posto que as cascatas de píons invadem os módulos. Assim, temos no histograma superior uma parte dos eventos que não se parecem muito com elétrons e

Método	20	100	180
23 energia	96,81%	95,73%	98,56%
23 raiz	98,83%	99,91%	99,78%
torre	99,40%	98,40%	99,67%
híbrido	98,45%	99,34%	99,77%
camadas	99,28%	99,91%	99,52%
anel el.	99,10%	99,75%	99,62%
anel pi.	98,85%	99,69%	99,89%
anel ambos	99,55%	99,60%	99,73%

Método	20	100	180
23 energia	90,39%	94,09%	95,90%
23 raiz	94,60%	96,35%	97,03%
torre	93,07%	97,66%	97,18%
híbrido	95,65%	97,25%	96,68%
camadas	92,67%	96,78%	96,87%
anel el.	93,74%	97,71%	95,59%
anel pi.	93,08%	95,81%	95,88%
anel ambos	95,06%	97,00%	95,74%

Tabela 7.8: Concordância entre o método clássico e os métodos neurais com pré-processamento topológico. Análise para o caso pión/múon (topo) e para o caso elétron/pión/múon.

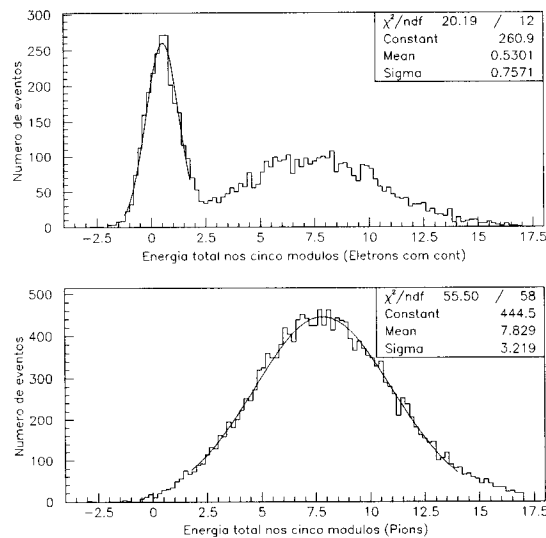


Figura 7.7: Histogramas da energia total depositada nos cinco módulos por eventos de elétrons (contaminado por píons) e píons. Em ambos os casos um corte na energia total no módulo em teste permitiu a eliminação da contaminação de múons.

sim com píons. Claramente, esta é a nossa contaminação. Concluimos que pode ser interessante utilizar esta informação extra para a separação de partículas. Assim, montamos a experiência detalhada na Figura 7.8.

Nesta experiência demos um tratamento diferenciados para os dois tipos de entradas que temos no sistema neural. As 46 entradas do Módulo 0, devidamente normalizadas pela raiz quadrada, foram utilizadas numa rede neural com 10 nós na camada intermediária. Esta rede era treinada para separar elétrons de píons (mais tarde veremos porque os múons não foram incluídos nesta pesquisa). Uma outra rede, baseada na resposta dos cinco módulos (no total eles têm duzentas células) e sem nenhuma normalização foi treinada para realizar a mesma separação. Desta vez apenas 5 nós foram usados na camada intermediária. Dessa forma, as saídas dos neurônios das primeiras camadas das duas redes acumulam informações discriminantes sobre os conjuntos de dados. Utilizando uma terceira rede, como a que podemos ver na Figura 7.9, faremos a discriminação final.

Esta rede recebe como 15 entradas os sinais extraídos pelas duas redes anteriores, sendo 10 vindos da rede relativa ao Módulo 0 e 5 vindos da rede dos cinco módulos. Assim sendo, esta rede (completa, com duas camadas) é treinada de novo pelo método de *backpropagation* para diferenciar elétrons de píons. Esta nova rede

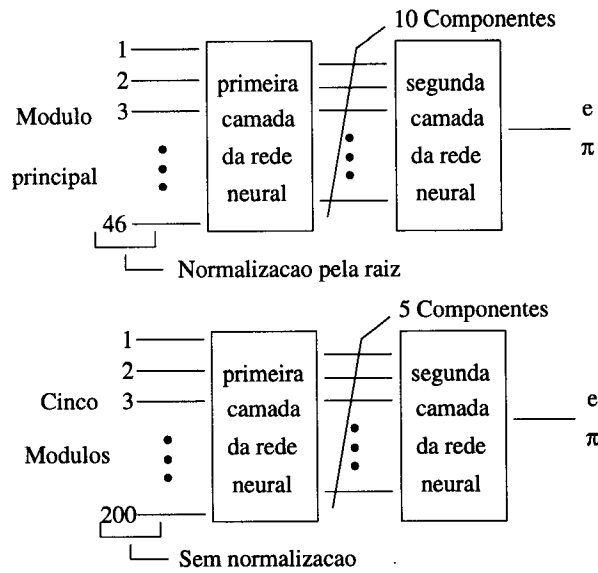


Figura 7.8: Esquema para extração das componentes de redes especializadas em cada uma das informações vindas separadamente do módulo em teste e dos cinco módulos.

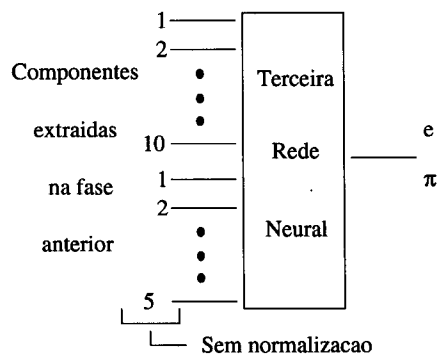


Figura 7.9: Terceira rede neural baseada nas informações extraídas pelas duas primeiras redes.

Método	Rede Neural		Método	Rede Neural		
	Clássico	e		π	Clássico	e
e		35,74%	2,17%	e	0,47%	0,90%
π	2,02%	60,17%	π	3,16%	95,47%	

Tabela 7.9: Resultados da separação elétron/píon para a rede treinada apenas com o módulo 0. Na esquerda temos o conjunto de elétrons (concordância de 95,81% e na direita o conjunto de píons (95,94% de concordância).

recebe informações privilegiadas das outras redes, com as características principais para realizar a discriminação devidamente extraídas, facilitando o trabalho de classificação dos eventos.

7.2.1 Resultados

Assim, para o conjunto de 180 GeV, usando a rede apenas para o Módulo 0 na separação entre elétrons e píons, temos os resultados que estão na Tabela 7.9.

Como podemos ver a rede consegue operar mesmo com uma contaminação de píons de 60% no conjunto de elétrons, o que é um resultado bastante interessante. O nível médio de concordância entre o método neural e o clássico (baseado em frações de energia) para os dois conjuntos fica sendo então de 95,87%. Este alto índice de contaminação por píons no conjunto de elétrons era esperado pelas análises do método clássico.

Nas Tabelas 7.10, temos a mesma análise de concordância aplicada ao caso em que os 200 sinais dos cinco módulos foram utilizados na entrada da rede sem nenhuma normalização.

Como podemos reparar, neste segundo caso, temos um maior erro com relação ao caso anterior. Vemos inclusive que os níveis de concordância são mais baixos dos que aqueles que tínhamos na separação realizada com informações do módulo 0. No caso, podemos reparar que uma parcela razoável dos eventos ditos píons pelo método clássico foi considerada como sendo elétrons pelo método neural. Na Figura 7.10 vemos a correlação entre a saída da rede neural e a energia depositada nos cinco módulos. Como podemos ver, uma parte dos eventos com alta energia nos cinco

Método Clássico	Rede Neural		Método Clássico	Rede Neural	
	e	π		e	π
e	37,66%	0,25%	e	1,01%	0,36%
π	8,60%	53,48%	π	6,05%	92,57%

Tabela 7.10: Resultados da separação elétron/píon para a rede treinada apenas com os cinco módulos. Na esquerda temos o conjunto de elétrons (concordância de 91,14% e na direita o conjunto de píons (93,59% de concordância).

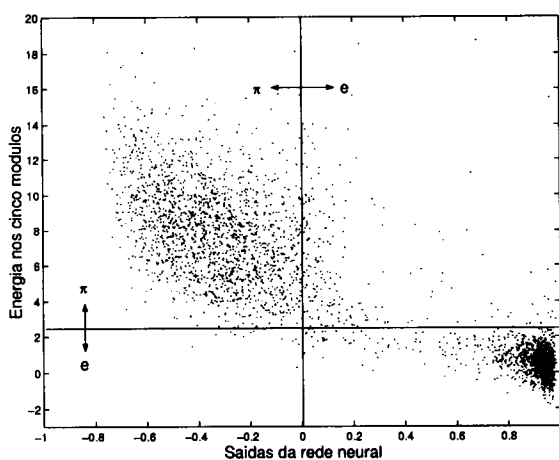


Figura 7.10: Correlação entre a resposta da rede neural (eixo horizontal) treinada apenas com as informações dos cinco módulos e a energia total depositada nestes módulos para o conjunto de elétrons contaminados.

módulos (logo considerados píons pelos métodos clássicos), fica dentro do intervalo positivo da saída da rede. Assim, podemos concluir que o nosso erro provem desta cauda que ultrapassa os limites do corte em 0 da rede neural.

Quando utilizamos as saídas das primeiras camadas de neurônios das duas redes que discutimos até agora para implementarmos a terceira rede neural, temos os resultados que estão expressos nas Tabelas 7.11.

O índice de concordância médio nestas tabelas ficou em 98,24%. Assim, vemos que houve um aumento do nível de acerto com relação aos dois casos anteriores. Podemos ver que o aumento se deu não só na média como também nos conjuntos individuais, reforçando a idéia de que a rede especializada nos dados dos cinco módulos, mesmo não obtendo tanto acerto quanto a rede baseada no Módulo

Método	Rede Neural		Método	Rede Neural		
	Clássico	e		π	Clássico	e
e		36,68%	1,23%	e	0,38%	0,99%
π		0,79%	61,30%	π	0,51%	98,12%

Tabela 7.11: Resultados da separação elétron/píon para a rede treinada totalizadora. Na esquerda temos o conjunto de elétrons (concordância de 97,98% e na direita o conjunto de píons (98,50% de concordância).

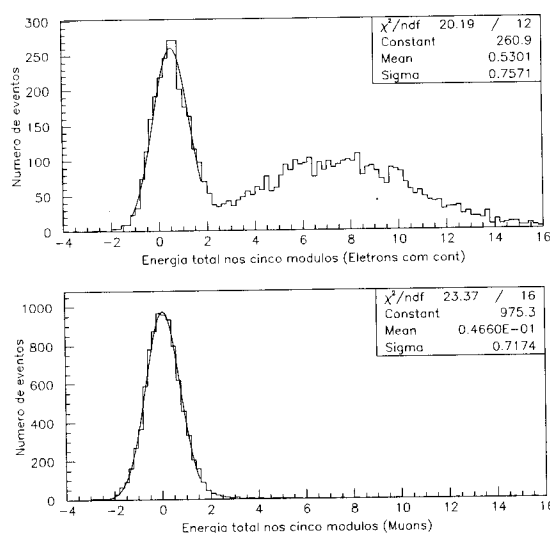


Figura 7.11: Energia total nos cinco módulos para eventos de elétrons contaminados por píons (acima) e múons (abaixo). Vemos que as médias são próximas.

0, trouxe alguma informação importante para a rede final que conjuga ambos os resultados.

Precisamos ainda responder a pergunta de porque não incluímos eventos de múons nesta análise. A resposta está na Figura 7.11.

Podemos ver que a energia depositada por eventos de elétrons, no histograma superior, é bastante próximos do nível de pedestal (sinal físico nulo) da resposta do detector. Isto significa que não há nenhuma possível estrutura de deposição de energia a ser analisada neste caso, diferentemente do que acontece com os píons que contaminam este feixe de elétrons. Com os eventos de múons, que aparecem no histograma inferior na figure, acontece a mesma coisa do que com elétrons, ou seja, não há nenhum sinal a ser observado. A conclusão é que a rede especializada nos

Rede	20 GeV			100 GeV		
	e	π	tot	e	π	tot
Mod. 0	95,54%	95,22%	95,38%	96,69%	96,53%	96,61%
5 Mod.	69,32%	96,76%	82,54%	97,15%	96,97%	97,06%
Tot.	70,48%	95,73%	83,11%	97,42%	97,46%	97,44%

Tabela 7.12: Concordâncias para as três redes baseando para cada conjunto e na média. As três primeiras colunas trazem os resultados para 20 GeV e as três últimas para 100 GeV.

cinco módulos não consegue diferenciar elétrons de múons, não convergindo para nenhum resultado interessante. Dessa forma, uma rede total não se beneficiaria com os resultados desta rede especialista, ao contrário, seria prejudicada por este resultado.

Podemos então estender os resultados até agora obtidos para outros níveis de energia. No caso, fizemos para o nível de 100 GeV (cortando eventos de múons com corte na energia total de 20 GeV) e 20 GeV (cortando múons com corte em 7 GeV). A Tabela 7.12 sumariza os resultados nestes dois casos.

Podemos ver que o caso de 100 GeV obteve alguma melhora com o procedimento de usar a rede neural que totaliza os resultados. Já para 20 GeV temos um problema, já que os resultados da rede especializada nos cinco módulos perde performance de identificação de forma bastante significativa. A rede totalizadora não consegue se recuperar, posto que a mesma tem que basear parte de sua decisão nos resultados da rede especializada em cinco módulos. A Figura 7.12 traz a explicação para este problema.

Como podemos ver, os eventos de elétrons (com contaminação de píons), ficam na mesma faixa em que encontramos os eventos de píons. Isto acontece porque os eventos de píons a 20 GeV tendem a produzir uma cascata bem menor do que aquela produzida a altas energias (100 e 180 GeV, por exemplo), ficando possível se conter a cascata quase completamente no Módulo 0. Conseqüentemente, a informação disponível para separar os eventos de elétrons dos eventos de píons através apenas dos cinco módulos é bem pouca. Assim, tanto a rede especialista quanto a totalizadora são prejudicadas.

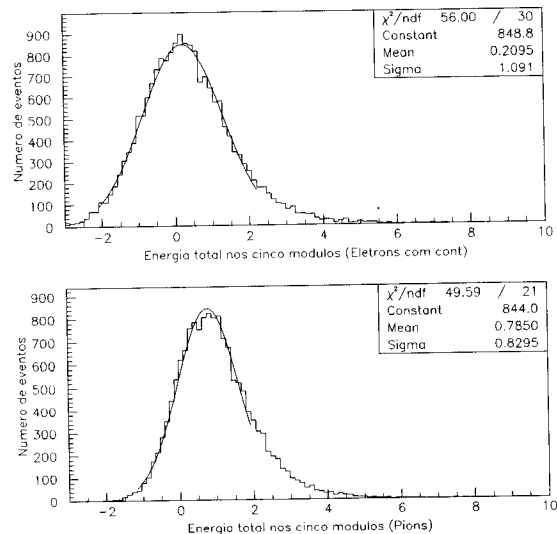


Figura 7.12: Energia depositada nos cinco módulos por eventos de elétrons com contaminação de píons (histograma superior) e eventos de píons (histograma inferior).

7.3 Componentes Principais de Discriminação

Nesta seção, exploraremos uma outra forma de treinamento da rede visando otimizar sua capacidade de discriminação para o nosso problema de classificação. Ao mesmo tempo, tentaremos reduzir o número de neurônios na camada escondida da rede, o que pode ser importante por reduzir o tempo total de treino de nossa rede.

A técnica é baseada em Componentes Principais de Discriminação [73]. Estas Componentes, ao contrário das conhecidas Componentes Principais [74], não são extraídas com a intenção de se manter uma boa descrição do conjunto de dados, mas sim uma alta capacidade de discriminação entre os conjuntos que compõe nosso universo de dados. A técnica de extração destas componentes baseando-se em redes neurais pode ser vista na Figura 7.13.

Na rede mostrada em (a) na figura, estamos extraindo a primeira componente. Neste caso, treinamos uma rede neural com as 46 entradas (para as 46 células do módulo 0) e três saídas (para as três partículas envolvidas - elétrons, píons e múons). O treinamento é realizado com o método de *backpropagation* visando atingir a máxima eficiência de acerto na classificação. Esta eficiência de acerto é a porcentagem de eventos identificados como sendo da classe a que pertencem, por

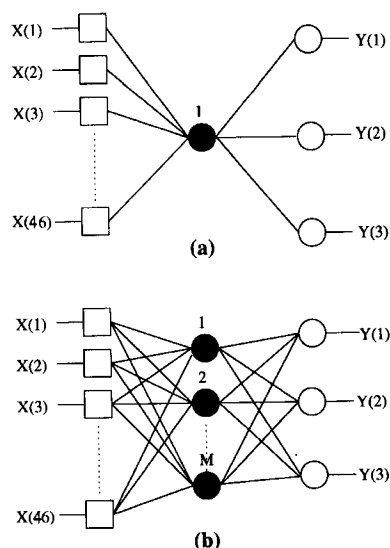


Figura 7.13: Esquema para extração da primeira Componente Principal (a) e da m -ésima componente (b).

exemplo, os elétrons no conjunto de elétrons. Claramente, esta medida é influenciada pelos níveis de contaminação que possam vir a existir. Veremos, entretanto, que este fator não prejudica o funcionamento do sistema. Ao fim deste treinamento, consideramos que a melhor representação possível do conjunto de dados para realizar a separação foi atingida e se encontra nos pesos da rede neural entre os nós de entrada e o neurônio da camada escondida. Sendo assim, fixamos estes pesos e adicionamos um segundo neurônio a nossa rede, sendo este completamente conectado com a camada de entrada e a de saída. Assim, com os pesos da primeira camada ao primeiro neurônio fixos, treinamos esta segunda rede para, de novo, atingir a máxima classificação possível. É importante notar, que os pesos da camada intermediária (agora com dois neurônios) para a camada de saída são deixados livres, para que o treinamento em si dê a devida importância para cada uma das componentes que estão sendo estudadas. Assim, seguimos adiante extraindo cada componente até que a eficiência da rede atinja um patamar bem alto e não melhore mais com a extração de novas componentes.

7.3.1 Resultados

Embora em nosso caso, durante o treino das redes para extração de cada componente, tenhamos usado o índice de reconhecimento como medida de acerto

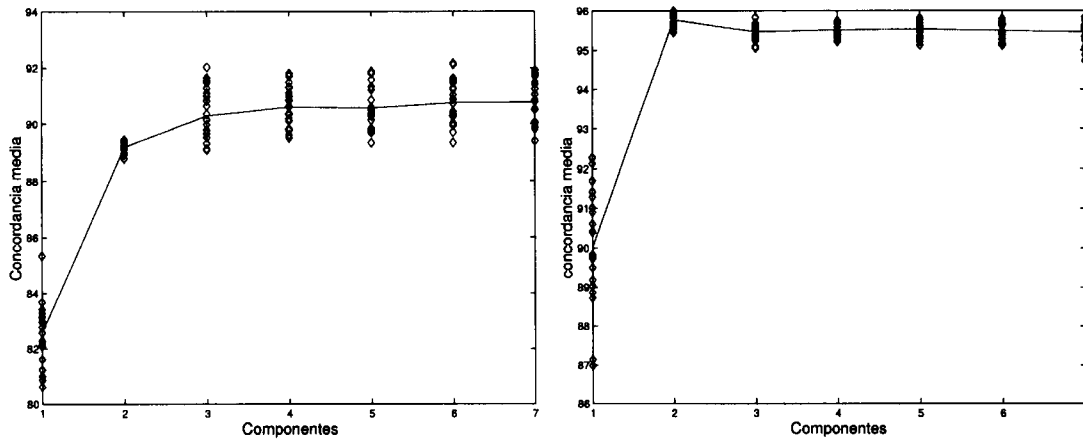


Figura 7.14: Concordâncias médias entre rede extraído Componentes Principais de Discriminação e método clássico para rede com normalização pela soma (esquerda) e pela raiz quadrada (direita).

para definirmos o ponto de parada do treinamento, estamos mais interessados é no nível de concordância dos resultados da rede com os resultados obtidos por métodos clássicos. Assim, estudaremos os nossos resultados baseando-nos exatamente nesta medida. Na Figura 7.14 temos exatamente estas concordâncias (média da concordância para os três conjuntos de dados - elétrons, píons e múons) para cada componente em cada treinamento realizado (a concordância é representada pelo pequeno losango). A energia do feixe neste caso era 20 GeV. Mais uma vez realizamos vários treinamento para garantir os resultados que estivemos explorando com alguma estatística. A linha que aparece é a média destas concordâncias para cada componente. Mais uma vez, temos o uso da normalização pela soma da energia no módulo (esquerda) e pela raiz quadrada (gráfico à direita).

Como podemos ver claramente, o gráfico relativo a evolução da concordância para cada componente com a normalização pela soma da energia, que temos uma evolução deste nível de concordância até a sétima componente, atingido aí um máximo. Quanto ao nível de concordância relativo à análise com normalização pela raiz quadrada, vemos que o mesmo atinge um máximo logo na segunda componente (máximo este maior do que qualquer treino com a normalização pela soma) e, ao contrário do caso anterior, o nível de concordância começa a diminuir a partir daí.

Realizamos este estudo para cada uma das energias que viemos utilizando até agora e o nível de concordância médio com métodos clássicos mais alto obtido

Energia	soma	cs	raiz	cr
20	90,77%	7	95,77%	2
100	92,54%	5	96,45%	3
180	94,23%	3	94,70%	2

Tabela 7.13: Concordâncias para extração das Componentes Principais de Discriminação com normalização pela soma e pela raiz da energia no módulo. A componente de máxima concordância para soma em cada energia é cs e para normalização pela raiz é cr.

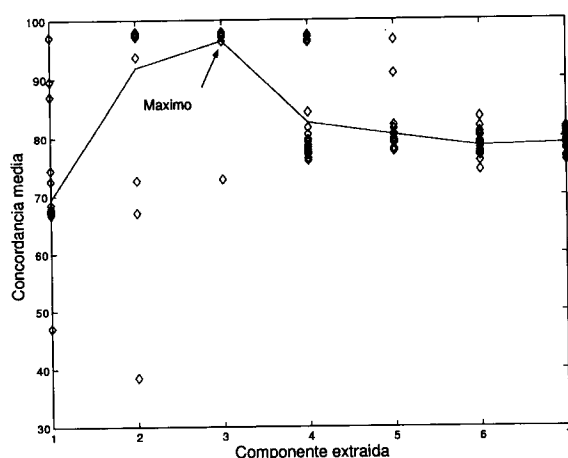


Figura 7.15: Evolução das concordâncias para 100 GeV com normalização pela raiz quadrada mostrando rápida ascensão e queda também brusca.

para uma dada componente, está na Tabela 7.13.

Podemos ver que o nível de concordância para o método baseado em raiz é bem maior do que a concordância para as redes treinadas com normalização pela soma das energias. Também podemos notar que a componente aonde é registrado o máximo é anterior no caso da normalização pela raiz quadrada do que aquelas encontradas com a normalização pela soma. Entretanto, em alguns casos, com a normalização pela raiz quadrada da energia, com a continuação da extração das componentes, o índice de concordância cai muito. Por exemplo, no caso de 100 GeV, temos o gráfico que pode ser visto na Figura 7.15.

Como podemos ver, embora o máximo registrado tenha sido bem alto (96.45% na terceira componente), na quarta componente já temos uma queda bastante forte,

Comp.	e	π	μ
3	56,15%	62,80%	99,50%
4	78,29%	62,84%	86,59%
5	75,62%	63,00%	88,48%
6	78,75%	63,09%	85,75%
7	78,44%	62,97%	86,31%

Tabela 7.14: Índice de identificação de eventos para cada conjunto de dados de acordo com a componente a partir da componente de máxima concordância para 100 GeV com normalização pela raiz quadrada.

ficando a média da concordância já abaixo de 83%. Ao que parece a rede está tendo algum problema na classificação de eventos com o aumento de componentes. Ao observarmos o índice de identificação pela rede em cada conjunto de dados (elétrons, píons, múons) que se encontra na Tabela 7.14 vemos que, no final de uma das extrações da terceira componente mais de 99,5% dos múons eram identificados com tal. Na extração da quarta componente utilizando esta terceira componente já extraída, esta identificação já cai para menos de 86,6%. Vemos também que o índice de identificação para elétrons sobe bastante, o que na prática significa que eventos de múons estão sendo incorretamente identificado como elétrons.

Isso demonstra que a rede começa a encontrar alguma dificuldade. Na Figura 7.16 temos para a extração da terceira e da quarta componente a saída da rede neural de elétrons menos a de múons (fator para separação entre estas duas partículas) correlacionada com a energia depositada no módulo para eventos de elétrons contaminados por píons e múons.

Como podemos ver, na região de alta energia, temos dois grupos de eventos. Um deles pode ser relacionado com elétrons (o mais próximo do alvo de elétrons) e o outro com píons (no meio da escala). Como podemos também reparar, há pouca mudança quando passamos de uma situação para a outra nesta região de mais alta energia, ou seja, os eventos de elétrons parecem continuar aonde estavam e os de píons também. Na região de baixa energia, entretanto, pode se notar um movimento dos eventos que na terceira componente eram considerados pela rede

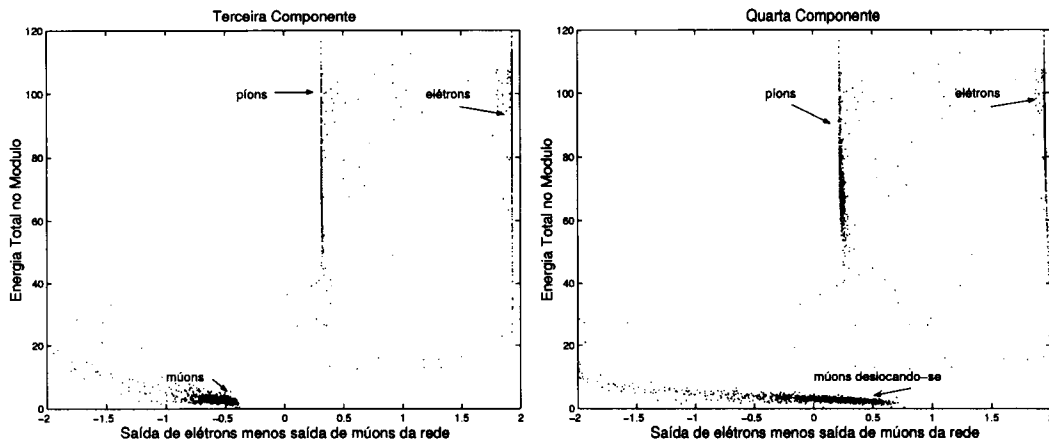


Figura 7.16: Saída de elétrons menos a de múons correlacionada com a energia no detector no final da extração da terceira (esquerda) e da quarta (direita) componentes.

como sendo múons (subtração das saídas negativa) passam na quarta componente a ser considerados como sendo elétrons (subtração das saídas negativa).

Sabemos de nossas análises anteriores que este conjunto de elétrons possui uma grande quantidade de eventos de múons (cerca de 25%), o que nos leva a concluir que a rede neural está tentando diminuir o erro encontrado ao trabalhar com estes eventos de múons que contaminam o conjunto de elétrons. Devemos lembrar que estes eventos são treinados com o alvo de elétrons (+1 na saída de elétrons e -1 na saída de múons), embora apresentem as mesmas características de eventos de múons de verdade (treinados com a saída de elétrons em -1 e a de múons em +1). Assim, a rede neural tende a levar o conjunto de múons para valores próximos a 0, satisfazendo ambas as condições conflitantes encontradas no treino.

A análise baseada na normalização pela raiz quadrada parece evidenciar as diferenças entre os conjuntos (particularmente a diferença entre elétrons/píons e múons), isso faz com que o erro da rede (que o método de treino está tentando minimizar) fique bastante alto, fazendo com que esta tendência de trazer os eventos para a média fique mais forte.

Podemos também observar a queda do erro em ambos os casos. Na Figura 7.17 temos o erro médio quadrático da rede neural (erro entre a saída obtida e a saída de alvo) na última época de treino (na prática nos últimos 600 passos do treino). No caso, cada ponto representa o erro médio em um treinamento e a linha

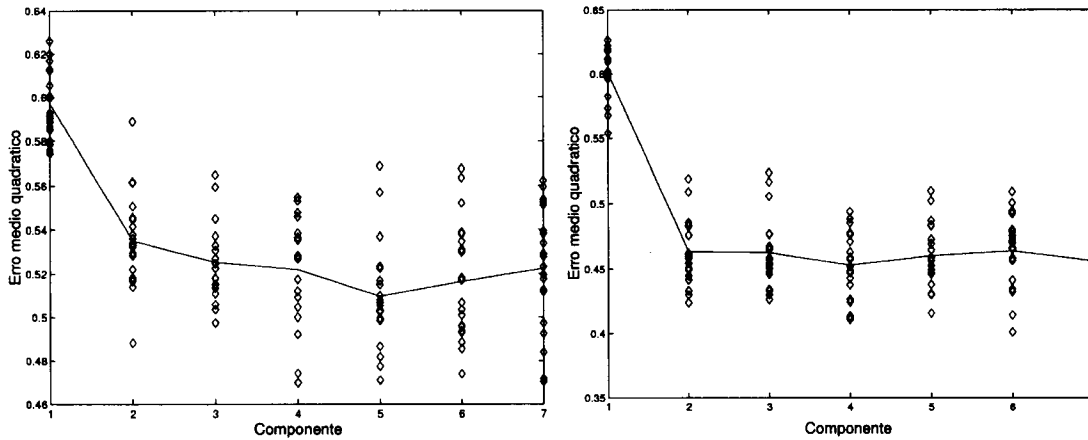


Figura 7.17: Evolução do erro médio quadrático para cada componente com a normalização pela energia total no módulo (esquerda) e pela raiz da energia total (direita).

Energia	Soma	Raiz
20	0,5224	0,4564
100	0,5773	0,4730
180	0,5772	0,5452

Tabela 7.15: Erros médios quadráticos após a extração da sétima componente para normalização pela soma da energia no detector e pela raiz quadrada.

a média dos erros médios quadráticos. A energia é de 20 GeV.

Como podemos notar, a rede com normalização pela energia total tem seu erro médio quadrático caindo até atingir um patamar mínimo em torno do qual ela fica oscilando. No último ponto do treino este valor é de 0,5224. Quanto a rede baseada na normalização pela raiz quadrada da energia total, vemos que o erro consegue cair mais ainda, chegando ao limite de 0,4564 na última componente. Neste caso, vemos uma queda do erro entre os dois métodos, o que não é necessariamente uma vantagem para a classificação como argumentamos acima. Para as outras energias envolvidas temos então os resultados que estão sumarizados na Tabela 7.15.

Como podemos ver o método com normalização pela raiz quadrada da energia tem sempre um erro menor do que pela energia total.

Para o caso 20 GeV com normalização pela raiz quadrada temos na Figura 7.18 a correlação entre a saída de elétrons de uma rede neural treinada conven-

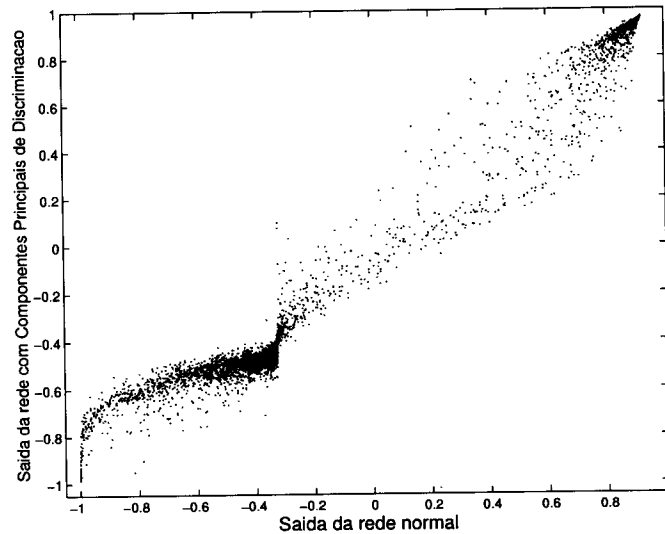


Figura 7.18: Correlação entre a saída de elétrons de uma rede treinada normalmente (8 neurônios na camada escondida) com a rede treinada com Componentes Principais de Discriminação (com apenas 2 neurônios).

cionalmente e da rede treinada usando Componentes Principais de Discriminação para o conjunto de elétrons contaminado por píons e múons.

O dado interessante é que, enquanto a rede convencional utilizou 8 neurônios na camada escondida a rede com Componentes Principais de Discriminação utilizou apenas 2 nós para obter uma performance semelhante de classificação. Vemos aí a vantagem deste método de compactação da rede neural.

Capítulo 8

Análise *Online*

Neste capítulo estudaremos a aplicação *online* do sistema neural que vimos estudando até o momento de forma *offline*. Uma metodologia foi desenvolvida e testada durante um dos períodos de testes com feixes para o TileCal.

Esta metodologia se baseou em dois fatos. Inicialmente, foi considerado o fato de que a rede trabalha de forma comparativa. Assim, para se retirar uma contaminação de múons de um conjunto de píons, é necessário dispor de amostras de eventos puros de múons, o que forma o padrão médio desta partícula, facilitando sua identificação no conjunto de píons.

O segundo ponto importante considerado foi a definição dos tipos de contaminações presentes nos feixes. Assim, no feixe de múons lembramos que não há nenhuma contaminação, enquanto no feixe de píons temos a contaminação típica por múons, e, finalmente, no feixe de elétrons temos a contaminação de píons e múons.

Assim, a metodologia desenvolvida para o processamento neural *online* foi estabelecida em três passos a seguir :

- Inicialmente, são adquiridos eventos de múons. Um certo número de eventos deste feixe é armazenado, ajudando a formar o padrão médio de eventos de múons para o processamento neural. Neste estágio todos os eventos são considerados múons, já que este é um conjunto tido como puro. Não há, assim, realmente execução de processamento neural.
- Na segunda fase começam a ser adquiridos eventos de píons com contaminação de múons. Após a aquisição de um certo número de eventos (para que se possa

ter um padrão deste segundo tipo - píons - definido), uma rede neural começa a ser treinada usando os eventos de múons adquiridos no passo anterior e os eventos novos do feixe de píons com contaminação de múons. Esta rede possui apenas uma saída realizando a identificação dos múons no feixe de píons. Desta forma, os eventos que vão chegando, recebem a resposta da rede neural. Para que a rede se mantenha atualizada com relação ao padrão de píons, os mais novos eventos devem ser constantemente utilizados no treino da rede, mesmo que a mesma já esteja corretamente classificando os eventos. Assim, um banco de dados com os últimos eventos é sempre mantido e utilizado no treinamento.

- Na terceira fase, elétrons são adquiridos. Uma outra rede neural começa a ser treinada, utilizando o conjunto de dados relativo ao feixe de múons adquirido na primeira fase, o conjunto de dados de píons (que pode manter ou não a contaminação de múons) adquirido na segunda fase e os elétrons de feixe que vão sendo adquiridos (com contaminação de píons e múons). Esta segunda rede possui três saídas, posto que será responsável por classificar entre elétrons, píons e múons.

Esta metodologia está de acordo com as condições que discutimos anteriormente, mas algumas questões ficam um pouco abertas :

- Qual o número mínimo de eventos que deve ser utilizados para que as duas redes neurais iniciem seu treino. Ou seja, no caso da rede da segunda fase, quantos eventos de píons devem já ter sido adquiridos para que a rede neural possa, mantendo um alto nível de acerto, classificar os novos eventos que estão chegando. O mesmo pode se dizer do número de eventos de elétrons que já devem ter sido adquiridos na terceira fase.
- Se o sistema deve funcionar *online*, a velocidade de processamento deve ser considerada em duas vertentes, a saber : na velocidade em que a rede neural deve estar pronta para dar respostas corretas (velocidade de treino), e no tempo de resposta para cada evento que é adquirido (avaliação de um único evento), de forma a não ocorrer atrasos no sistema de aquisição de dados. Além disso, como realizar as duas operações (treino e avaliação de novos eventos) de forma balanceada.

- Os dados vindos do calorímetro são dados brutos, sem os valores de energia por célula que alimentam a rede neural. Precisamos realizar esta conversão de forma a obter valores iguais aos que seriam obtidos pela análise *offline*, o que coloca de novo o problema de velocidade (as rotinas para decodificação de eventos são relativamente pesadas computacionalmente).
- Finalmente, temos o problema de definir em que ponto do fluxo de dados iremos inserir o nosso processamento neural. Esta questão é importante, pois pode implicar em uma redução na taxa de processamento do sistema de aquisição de dados do calorímetro, o que seria completamente contra-produtivo e inviabilizar a utilização *online* da técnica. Podemos também, caso os eventos de contaminação sejam rejeitados do fluxo de dados, diminuir o peso computacional sobre as unidades do sistema de aquisição após o ponto onde os eventos são rejeitados.

As respostas para a primeira questão colocada serão estudadas na próxima seção, onde análises *offline* verificarão melhor os limites de aplicação da técnica neural, quando são aplicadas restrições relativas a execução *online*.

As outras questões serão estudadas na segunda seção deste capítulo, quando descreveremos as soluções utilizadas para se ter o processamento *online* corretamente implementado.

Finalmente, a terceira seção deste capítulo traz os resultados que foram obtidos com a aplicação neural *online*, bem como uma avaliação dos resultados e da performance relativa a questões como a taxa de aquisição.

8.1 Preparando para a Análise *Online*

Embora o título desta seção se refira a análise *online*, aqui faremos uma pequena análise *offline* para demonstrar a viabilidade da aplicação. No caso, o fator mais importante que podemos aqui discutir se refere a quantidade mínima de eventos que podemos ter durante o treino da rede para podermos manter um alto nível de classificação de novos eventos. Como no nosso caso, o treinamento da rede é realizado *online*, é fundamental saber após quantos eventos adquiridos a rede pode fornecer respostas corretas, já que teremos que desconsiderar seu resultado para os

primeiros eventos de píons (na segunda fase de nossa metodologia) e de elétrons (na terceira fase de nossa metodologia).

8.1.1 Separação Píon/múon

Começamos então pela análise de 20 GeV na separação píon/múon com normalização pela soma e pela raiz quadrada da energia total no detector. Realizamos um treino no qual restringíamos significativamente o tamanho do conjunto de treino da rede, mantendo o teste para todo o resto do conjunto. Este treinamento foi realizado diversas vezes, visando acumular alguma estatística no procedimento, através de uma inicialização aleatória da rede. Assim, fomos computando o nível de concordância ao final do treinamento da rede com relação ao método clássico correspondente a situação (corte em energia para a separação píon/múon e contador Cherenkov ou cortes baseados em fração de energia para a separação elétron/píon/múon) exatamente como havíamos realizado nos capítulos de análise *offline*. Esta concordância era, obviamente, computada no conjunto de teste de todas as partículas, para estimar a qualidade da generalização produzida pela rede neural.

Na Figura 8.1 podemos ver o resultado desta análise. Os pequenos losangos representam a concordância com métodos clássicos no conjunto de teste para cada treinamento realizado, em função do número de eventos no conjunto de treinamento. A linha passa pela média dos valores obtidos em cada ponto. Como podemos ver, em ambos casos, com normalização pela energia total (esquerda) ou pela raiz quadrada da energia total (direita), a capacidade da rede fica bastante comprometida quando usamos apenas 20 eventos de cada tipo no treinamento. A concordância vai gradualmente aumentando conforme avançamos no número de eventos utilizados no treinamento até atingir um patamar aonde ocorre praticamente o estacionamento desta melhora, com variações normais relativas às diferentes inicializações.

Podemos ver na figura relativa a normalização pela soma que um resultado bastante positivo pode ser encontrado por volta de 100 eventos no treinamento, ficando nitidamente acima dos 90% de concordância com o método clássico. Já no caso de normalização pela raiz quadrada, a estabilização parece chegar mais tarde. Entretanto, esta estabilização ocorre num patamar mais alto. Podemos utilizar um patamar perto de 200 para termos alguma garantia em nossos resultados. Assim,

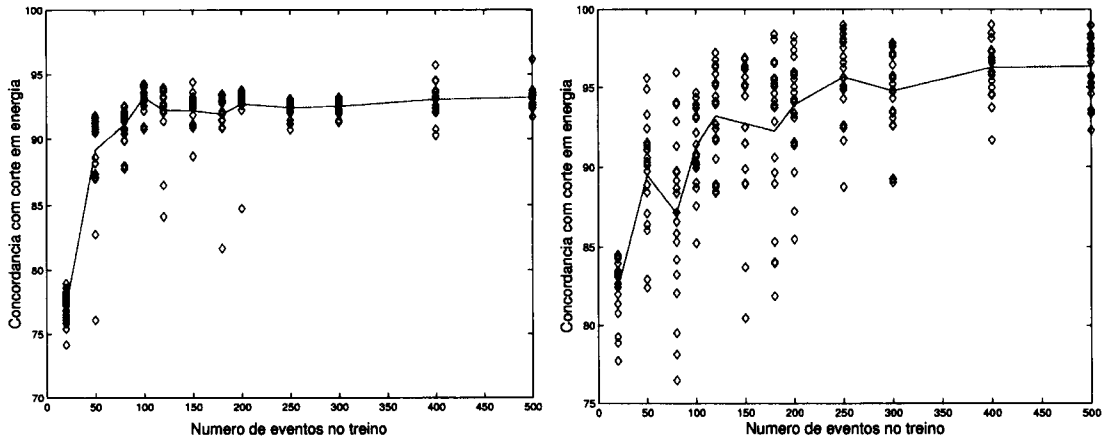


Figura 8.1: Análise da capacidade de generalização da rede com um número restrito de eventos durante o treino. Normalização pela energia total (esquerda) e pela raiz quadrada da energia total (direita).

energia	norm. energia	norm. raiz
20	92,71%	93,88%
100	94,92%	95,15%
180	98,13%	99,70%

Tabela 8.1: Tabela de índice de concordância médio nos conjuntos de teste para as três energias e duas metodologias de normalização quando 200 eventos são usados no treinamento para cada tipo de partícula.

temos na Tabela 8.1 o nível de concordância médio com corte em energia para as três energias que temos até agora considerado (20, 100, 180) para os dois métodos de normalização. O corte em energia utilizado foi 7 GeV, para feixes de 20 GeV, e 20 GeV, para feixes de 100 e 180 GeV.

Como podemos ver na tabela, os valores obtidos com a normalização pela soma são um pouco menores do que os obtidos pela normalização pela raiz quadrada. Também notamos que há um aumento do nível de concordância com o aumento da energia, num resultado semelhante ao que exploramos na análise *offline*.

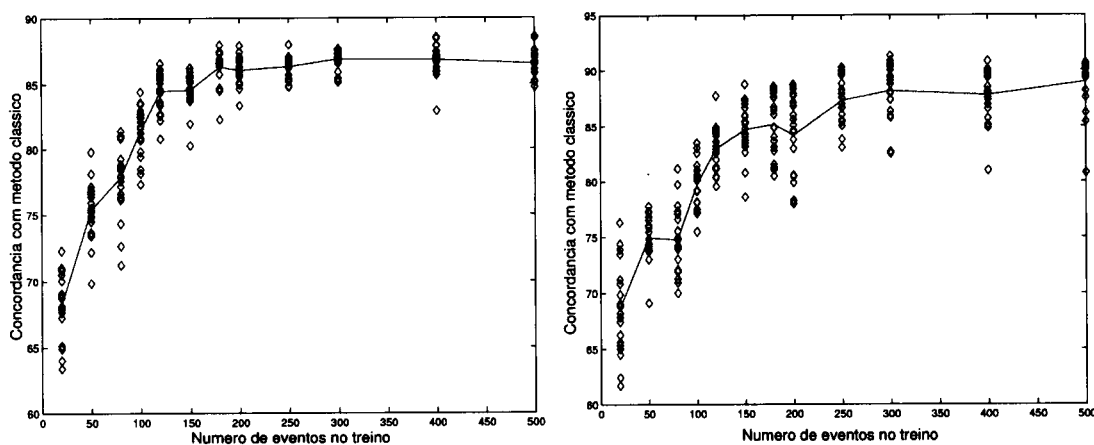


Figura 8.2: Análise da capacidade de generalização com número restrito de eventos para treinamento na separação elétron/píon/múon. Normalização pela energia total (esquerda) e pela raiz quadrada da energia (direita).

8.1.2 Separação Elétron/píon/múon

Da mesma forma, realizamos o mesmo procedimento para a separação elétron/píon/múon. Visamos também neste caso determinar o número de eventos mínimo a ser considerado durante o treino. A Figura 8.2 traz este resultado de concordância para o método clássico, utilizando corte em 7 GeV e 80 contagens do contador Cherenkov para 20 GeV. Da mesma forma que anteriormente, temos o resultado para normalização pela soma das energias em todas as células (esquerda) e pela raiz quadrada desta soma (direita). Também da mesma forma que anteriormente, temos níveis de concordância bastante ruins para o começo, quando usamos apenas 20 eventos por feixe considerado, sendo que adiante temos um comportamento assintótico para um dado valor mais alto. Novamente, o treino realizado com normalização pela energia total parece atingir o seu patamar mais rapidamente do que o pela raiz quadrada da energia, estabilizando-se por volta de 200 eventos por tipo de partícula. O caso em que usamos normalização por raiz quadrada parece demorar um pouco mais para atingir o seu patamar, fazendo-o apenas após os 300 eventos.

Assim, usando 300 eventos como ponto de determinação do número mínimo para este caso, podemos levantar a tabela de nível médio de concordância para este novo caso. A Tabela 8.2 traz este resultado. Nesta tabela os valores são inferiores

energia	norm. soma.	norm. raiz
20	86,85%	88,20%
100	88,67%	86,14%
180	92,18%	92,65%

Tabela 8.2: Tabela de índice de concordância médio para as três energias e dois métodos de normalização quando 300 eventos são usados no treinamento para cada tipo de partícula.

ao da tabela anterior, mas também o problema de classificar entre três partículas é bem mais complexo, como já havíamos notado anteriormente. Nos casos de 20 e 180 GeV manteve-se a tendência de concordâncias ligeiramente superiores para a normalização pela raiz quadrada da soma.

8.2 Implementação da Classificação Neural *Online*

Nesta seção, discutiremos o sistema que foi implementado durante a estadia no CERN para análise neural *online* durante testes com feixes. Conforme apontamos no começo deste capítulo, um dos problemas cruciais deste sistema é a velocidade, tanto para treinamento quanto para teste da rede neural.

O pacote JetNet-2.0 que viemos utilizando até então para simulação da rede neural, é escrito em linguagem Fortran. Dado o fato de que os compiladores Fortran atuais não passam de tradutores para C, se esperava que esta versão não seria a mais otimizada para lidarmos com uma situação aonde a restrição de tempo é muito forte. Assim, durante o desenvolvimento de um experimento para a marinha brasileira [75], desenvolvemos um código para treino e teste de uma rede neural pelo método de *backpropagation* utilizando linguagem C ansii para a aplicação num Processador de Sinais Digitais (*Digital Signal Processor - DSP*) da Analog Digital Inc., o ADSP-21060 [76].

Considerando a estrutura repetitiva das operações para treino e teste de uma rede neural, a operação desta num DSP, seria extremamente rápida. Entretanto, como o código foi desenvolvido em linguagem C ansii, o mesmo pode ser aplicado em qualquer processador para o qual exista este compilador. Para alcançar o máximo

Linguagem	600K passos	1 passo
Fortran	7,2 sec.	12 μ sec
C ansii	3,7 sec.	6,17 μ sec

Tabela 8.3: Comparação entre velocidade de processamento em duas linguagem de programação diferentes.

de velocidade, utilizamos estruturas que pudessem reduzir ao máximo as operações de localização na memória do processador dos valores dos pesos, parte que pode desacelerar bastante os procedimentos de treino e teste da rede neural. Assim, por exemplo, no trecho abaixo, variáveis locais (*indice1* e *indice2*) são criadas com a intenção de economizar o tempo de chamada das funções *find_neuron* e *find_weight*. Normalmente, as chamadas a esta função viriam dentro do *loop* relativo a *neuron1*. Na verdade, só é necessário se encontrar o primeiro peso de um dado neurônio, posto que, dada a estrutura utilizada, os outros pesos daquele mesmo neurônio estão em seqüência após o primeiro. Daí o uso das variáveis *indice1*, *indice2* e do operador de incremento ++.

```

for(neuron=0;neuron<neu_per_samp[1];neuron++){
    indice1=find_neuron(1,neuron);
    indice2=find_weight(1,neuron,0);
    for(neuron1=0;neuron1<neu_per_samp[0];neuron1++){
        weight[indice2]+=parf[0]*delta[indice1]*in_vec[neuron1];
        indice2++;
    }
}

```

Laços como este fazem que sejam economizadas *neu_per_samp[0]-1* (que no caso é o número de neurônios da primeira camada menos um, ou seja, no caso do barril, este número é igual a 45) operações de *find_neuron* e *find_weight* por *loop*, otimizando bastante o processamento.

A título de comparação temos a Tabela 8.3 com os tempos de processamento para rotinas de treino da rede neural em Fortran (JetNet-2.0) e em C ansii (realizado por nós para esta tese). Nesta tabela, os valores medidos de tempo para as operações

foram realizados no mesmo processador (um AMD K7 rodando a 1,3 GHz). Temos na tabela a medida que constitui em se rodar 600.000 vezes o treino da rede em ambas as implementações. O valor considerado como sendo o tempo de um passo é simplesmente o tempo de 600.000 passos de treinamento dividido por este mesmo número. Vemos que houve um ganho de quase 2 vezes só nas versões do *software*. Foi também medido o erro entre as duas implementações para os valores dos pesos e das saídas da rede após 50.000 passos. O erro medido foi inferior a 10^{-5} , o que nos faz considerar as duas implementações como sendo totalmente equivalentes. Mais tarde veremos que este código foi portado para um computador Power-PC com o sistema operacional LynxOS. Como a frequência deste processador era menor, 200 MHz, além de sua diferente estrutura, o tempo de uma chamada da rotina de treinamento passava a ser de $65 \mu s$.

De forma a se aumentar a flexibilidade das aplicações nas quais poderíamos aplicar o código neural, resolvemos encapsular o mesmo em classes C++, dando ao sistema a possibilidade de trabalho com redes neurais como objetos simples desta linguagem, que podem ser criados e destruídos em tempo de execução. Através de uma interface simples os objetos "Neurais" podem ser manipulados, podendo ser chamadas rotinas de treino e teste. Os pesos são parte de cada objeto, sendo possível que duas rede coexistam e sejam treinadas ao mesmo tempo.

A última alteração importante feita no código neural *online* foi a conversão de seu código a um código *multithread*. Este tipo de implementação permite que um mesmo processo se divida em dois (ou mais), de forma que o conteúdo de suas variáveis continue a ser compartilhado. No caso de termos uma máquina com mais de um processador pode-se inclusive executar *threads* ao mesmo tempo, uma em cada processador. Assim, partes do processamento podem ser realizadas por uma *thread*, e outras partes por outra *thread* [77]. Desta forma, temos uma resposta simples para a segunda questão que apresentamos na introdução deste capítulo. Nosso sistema tem duas prioridades que devem ser atendidas de forma imediata : responder o mais rápido possível à chegada de novos eventos e treinar mais rapidamente possível os pesos da mesma. A parte de resposta a novos eventos é implementada numa *thread*. Esta *thread*, fica parada esperando uma interrupção que significa a chegada de um novo evento. Quando o evento chega, a interrupção "acorda" a *thread* que realiza

o processamento daquele único evento. Enquanto nenhum evento novo chega, o processador fica liberado automaticamente pelo sistema operacional para executar a outra *thread* que é responsável por realizar o treino com os eventos que já foram acumulados até o momento.

8.2.1 Aplicação Neural no DAQ

O último ponto, que devemos observar para completar as exigências relativas à nossa aplicação neural *online*, é como inserir corretamente a aplicação no sistema de aquisição de dados. Dois aspectos importantes devem ser considerados. Inicialmente, devemos lembrar que deve se interferir o mínimo possível no fluxo de eventos. Também é importante o acesso aos dados. Assim, conforme estudamos no capítulo 4 e está descrito na Figura 4.1 a *Beam Crate* não tinha acesso aos dados do calorímetro, logo não nos interessa como ponto de aplicação do sistema neural. A *ROC Crate* tem acesso aos dados do calorímetro, mas, infelizmente, esta parte tem um *software* muito pesado computacionalmente, sobrando pouca disponibilidade para um processamento a mais. Nos PCs para gravação dos dados também o processamento é pesado, e além disso, se houver a intenção de se rejeitar eventos, este seria um ponto muito alto na escala do sistema de aquisição para tal, tendo o evento a ser descartado já sobrecarregado todas as unidades anteriores.

Assim, o melhor ponto para a inserção do sistema neural seria a *ROD Crate*. Esta *Crate* está logo no começo do processamento e tem dados do calorímetro, logo é boa candidata a aplicação neural. O único problema é que os processadores que recebem os dados do calorímetro tem uma carga computacional muito forte, ficando mesmo inacessíveis durante o período de aquisição (é impossível se manter um mesmo terminal de acesso funcionando para esta máquina durante estes períodos). Assim, a única opção seria aplicar o sistema neural na placa controladora (marcada como *ROD-CC* na Figura 4.1). Esta placa tem um nível de processamento que se restringe a algumas operações por segundo para comunicação com o sistema geral do DAQ.

Assim, na Figura 8.3 temos, de novo, a *ROD Crate* com os seus processamentos normais do controlador (*ROD Controller*) e o processamento dos dados do calorímetro (*ROD Board*). Adicionamos na placa controladora o processo neural

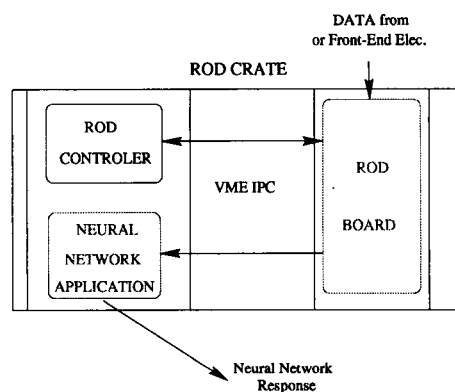


Figura 8.3: Inserção do processamento neural na ROD Crate.

(*Neural Network Application*). O único problema desta configuração é o fato de que os eventos chegam para o processo *ROD Board*, e logo não estão disponíveis de forma simples na placa controladora. A forma de se vencer este obstáculo é utilizar a comunicação VME para se trazer os eventos da placa *ROD Board* para a controladora. Exatamente por isso, trabalhamos com as bibliotecas desenvolvidas com métodos de transmissão usando DMA conforme descrito na seção 4.2.2. Conforme descrevemos nesta seção, para o tamanho aproximado dos fragmentos de eventos do TileCal, a velocidade de transmissão desta biblioteca atinge cerca de 7500 eventos por segundo, da placa escrava (*ROD Board*) para a controladora. Assim, o programa de redes neurais também utiliza objetos C++ especializados para transmissão de dados no barramento VME.

As respostas da rede neural podem ser armazenadas em um arquivo texto ou serem devolvidas para a *ROD Board* para serem inseridas no fluxo de dados. No caso, o ponto para inserção das respostas na estrutura de dados do fragmento do TileCal é a *Status Word* conforme apresentamos na Figura 4.3. A resposta recebida pelo barramento VME, é re-escalada de -1 a +1 para 0 a 1.000.000 (posto que os números tem que ser inteiros no fragmento de dados), e adicionada ao fragmento. Um pequeno programa para extrair os valores de resposta da rede neural foi desenvolvido posteriormente para facilitar a análise *offline* dos resultados. Notamos que nesta primeira experiência, não foram realizados cortes de eventos baseados na resposta neural, apenas a associação das respostas à estrutura dos mesmos.

Cabe ainda salientar que a *ROD Board* e a placa controladora são máquinas Power-PC rodando o sistema LynxOS, como já havíamos descrito na seção relativa

ao sistema de aquisição. Assim, nosso código teve de ser também compilado neste sistema. Podemos também adicionar que como o sistema operacional LynxOS é um sistema em tempo real, os processamentos relativos a parte *multithread* e a gerência de interrupções se dão em tempo real, o que só veio facilitar nossa aplicação. No sistema Linux, obedecendo ao *scheduler* do sistema operacional, o processador seria disponibilizados para outras tarefas (tais como *swap* de dados entre o disco e a memória) pelo tempo "infinito" para a aplicação de 10 ms (tempo padrão do Linux). Isso arrasaria a capacidade do sistema de responder na taxa requerida pelo DAQ do TileCal. Assim, nosso código neural acabou por ser completamente testado e utilizado em três plataformas completamente distintas, a saber : para o DSP ADSP-21060 da Analog Digital Inc. (apenas as rotinas de treino e teste, já que não existiam na época compiladores C++ nem um sistema operacional com capacidades como ser *multithread* para este DSP); PCs rodando o sistema operacional Linux (como a linguagem utilizada é C ansii, o mesmo vale para o sistema operacional Windows); e, finalmente, Power-PCs em barramento VME com o sistema operacional LynxOS em tempo real.

Conforme já havíamos dito anteriormente, os dados brutos vindos do detector são recebidos na *ROD Board*, e devem ser processados para se extrair as informações de energia por célula do calorímetro. No presente caso, apenas nove amostras do sinal das células (ver Figura 3.5 a direita), estava disponível para se extrair a informação de energia. As rotinas utilizadas para este cálculo são bastante complexas, pois dependem de um cálculo inicial para se localizar o início do sinal (fase) para se estimar o pedestal (ruído de fundo, como já havíamos mencionado) e se calcular não só a energia, mas também a fase correta do sinal. Durante o trabalho com o sistema de monitoração do detector 4.2.1, tivemos a oportunidade de lidar bastante com estas rotinas de decodificação dos dados brutos.

Foi implementada uma versão simplificada deste processamento dos dados, aonde todo cálculo que pudesse ser considerado supérfluo foi eliminado. Por exemplo, o cálculo do tempo de chegada do sinal (fase), foi eliminado, assim como a determinação de várias outras variáveis que são utilizadas para dar maior garantia ao processamento *offline*. Assim, apenas o cálculo da energia foi mantido com total consistência, mantendo o mesmo resultado da análise *offline*. A única alteração com

relação a estes dados, foi o fato de não dispormos de constantes de calibração que só são disponibilizadas muito depois dos testes com feixe, quando algumas *runs* especiais de calibração (com o sinal sendo gerado por uma pequena amostra de céσιο que percorre a estrutura do calorímetro) são completamente analisadas. Entretanto, todos os testes que fizemos na seção *offline* foram realizados com estas constantes brutas, garantindo que não há problemas em se utilizar uma calibração crua (todas as constantes em um mesmo valor para todos os canais).

Mais uma questão importante é que, durante o teste com feixes, alguns eventos de calibração (como por exemplo, com a simulação do sinal eletrônico em substituição ao sinal vindo da foto-multiplicadora, ou simples medidas do sinal de pedestal) são realizados entre os eventos ocasionados por partículas. Isto é feito para se verificar a estabilidade do sistema com relação a calibração que é produzida por *runs* de calibração anteriores as *runs* de eventos físicos. Estes eventos representam um problema, posto que seus valores fictícios podem atrapalhar muito o treino da rede com os eventos da física. Assim, tivemos de alterar a complicada lógica de *trigger* associada ao sistema de aquisição para cancelar estes eventos de calibração pelo menos durante nossos testes, somente permitindo que eventos de física (eventos relativos à partículas vindas do feixe), pudessem ocorrer. Na seqüência do trabalho, com auxílio do sistema de TTC, podemos, de forma relativamente fácil, realizar esta identificação por *software*, e somente enviar para o processamento neural os eventos relativos à física.

8.2.2 Estrutura Final da Aplicação Neural

Assim, a aplicação desenvolvida pode ser descrita pela Figura 8.4. Nesta figura separamos o processo em três partes principais, suas duas *threads* separadas para resposta aos eventos e treino da rede (parte esquerda e parte inferior direita respectivamente). Também marcamos a parte de memória que os processos dividem (na parte superior a direita).

Dados todos estes detalhes, a parte responsável pela resposta do sistema e introdução de novos dados no *Buffer* de dados trabalha da seguinte forma : Esta *thread* permanece completamente desativada enquanto não chega uma interrupção VME vinda da *ROD Board* indicando a presença de um novo evento. Quando esta

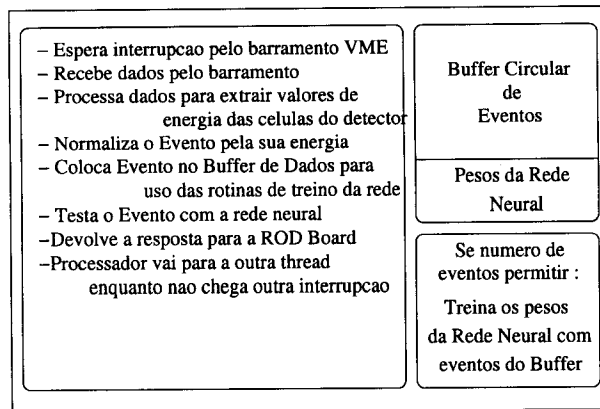


Figura 8.4: Aplicação *online* da rede neural em duas *threads* para testes com feixe do TileCal.

interrupção é recebida pelo processador, o sistema operacional "acorda" a *thread* que, utilizando a biblioteca de transmissão rápida de eventos, recebe o bloco de dados contendo as amostras do sinal. É, então, aplicado o processamento para se converter as amostras em sinal de energia. Quando a energia de todas as células é calculada, o evento é normalizado pela energia total (na época em que o teste foi feito não havia sido explorada ainda a solução da normalização pela raiz quadrada). O evento normalizado é colocado no *Buffer* de Dados, que é circular (um número máximo de eventos é aceito, quando o limite é atingido, cada novo evento substitui o mais antigo ainda no *Buffer*). O evento é testado pela rede neural para que se possa ter a resposta da mesma ao evento. Os pesos na memória compartilhada são usados para este teste (devemos lembrar que estes pesos estão sendo treinados pela outra *thread*). A resposta da rede, devidamente re-escalada para um número inteiro é devolvida pela comunicação VME para a *ROD Board* para ser incluída na estrutura do evento. Neste momento esta *thread* volta a esperar a interrupção VME, sendo desativada pelo sistema operacional.

A outra *thread* espera até que exista um número mínimo de eventos para que se tenha uma certa representatividade dos conjuntos de dados. Quando o número mínimo é atingido a rede começa a ser treinada com os eventos que estejam presentes no *Buffer* de Dados. Os pesos treinados são os da memória compartilhada. Desta forma, foi possível se realizar um processamento balanceado e que incluía todas as fases necessárias para treino e teste da rede.

Assim, podemos passar aos resultados do teste realizado *online* durante o

teste com feixes.

8.3 Resultados do Experimento *Online*

Durante o teste com feixes realizado em setembro de 2001, foram testados para fins de calibração, dois módulos da Extensão do Barril. Estes módulos, com toda sua eletrônica completamente funcional, foram conectados via *link* ótico, a cada um das duas placas Power-PC na ROD Crate. Desta forma, cada placa teria acesso a uma das metades do evento. Como o feixe, só pode atingir uma destas metades de cada vez, a aplicação neural só precisa ser informada (através de um parâmetro de configuração) sobre qual dos módulos estava recebendo o feixe.

Estas placas Power-PC (tanto a controladora aonde o processo neural será realizado, como a *ROD Board* que deve preparar os eventos para a transmissão de dados), são do modelo Rio II [78], com o processador de 200 MHz e 32 M-Bytes de memória. O sistema operacional utilizado é o LynxOS versão 3.0.1 da Creative Labs. Este sistema operacional trabalha com o sistema de *threads* Posix, que é um padrão utilizado em vários sistemas operacionais. As placas possuem um processador próprio para as transmissões DMA no barramento VME. O tempo de treino desta rede (consideramos a rede treinada a partir de 50.000 passos de treino) foi medido em cerca de 3.25 segundos, posto que as rotinas de treino precisam de 65 μs por passo de treino nesta arquitetura.

Como o módulo do barril estendido possui 28 leituras de suas 14 células, e os valores das leituras foram combinados (por simples média aritmética), o vetor de entrada da rede neural tinha dimensão 14. Assim, a rede de três camadas tinha 14 neurônios de entrada, 8 na camada escondida e 1 neurônio na última camada. Devido a dificuldade de se trocar muito rapidamente de feixe de partículas, e o fato de termos pouco tempo para trabalhos especificamente com a rede neural, só conseguimos realizar o experimento de separação entre píons e múons. Por isso, só testamos a rede neural com apenas um neurônio na última camada, indo apenas até o segundo passo de nossa metodologia apresentada no começo deste capítulo.

Os eventos foram normalizados pela soma das energias depositadas nas células. A *learning rate* inicial utilizada para treino da rede foi 0,2 e sua queda a cada época

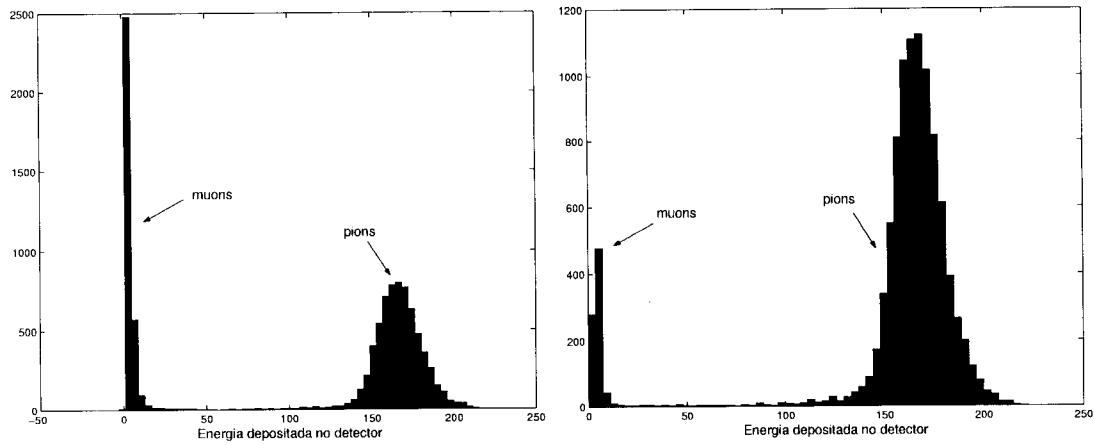


Figura 8.5: Deposição de energia no detector por *Runs* de píons a 180 GeV com contaminação. *Run* 120324 (esquerda) e 120326. Notar que a primeira tem uma índice maior de contaminação.

foi definida como sendo 0,997. A época inicial foi estabelecida em 600 passos de treino. O *Buffer* de dados podia conter no máximo os últimos 200 eventos que chegaram conforme estudamos na primeira seção deste capítulo e a rede começava a ser treinada após os primeiros 50 eventos de píons terem sido adquiridos (o banco de dados de múons ficava com os 200 eventos adquiridos durante a *run* de múons realizada antes).

Os resultados que iremos agora explorar se referem a 4 *runs*, duas de feixe de múons e duas de feixe de píons com contaminação de múons a 180 GeV. Na Figura 8.5, temos, para as *runs* de píons de número 120324 e 120326 com 10.000 eventos cada uma, a deposição de energia no detector. A *run* 120324 tinha um índice de contaminação (estimando pelo corte em energia) de 32%, enquanto a segunda *run* tinha uma contaminação de cerca de 8%. Conforme veremos adiante, estas contaminações foram identificadas pela rede neural. Os diferentes níveis de contaminação foram obtidos através de ajustes realizados no feixe. As *runs* 120323 e 120325 de apenas 200 eventos foram realizadas com feixe de múons apenas para se encher a parte relativa a esta partícula no *Buffer* de Dados.

Para os eventos dos feixes de píons, temos na Figura 8.6 as correlações entre a saída da rede neural treinada *online* para os eventos após o evento de número 200 (quando a rede já pode ser considerada treinada). Como podemos ver, a rede parece ter corretamente identificado os padrões de cada partícula, pois, mesmo sendo ali-

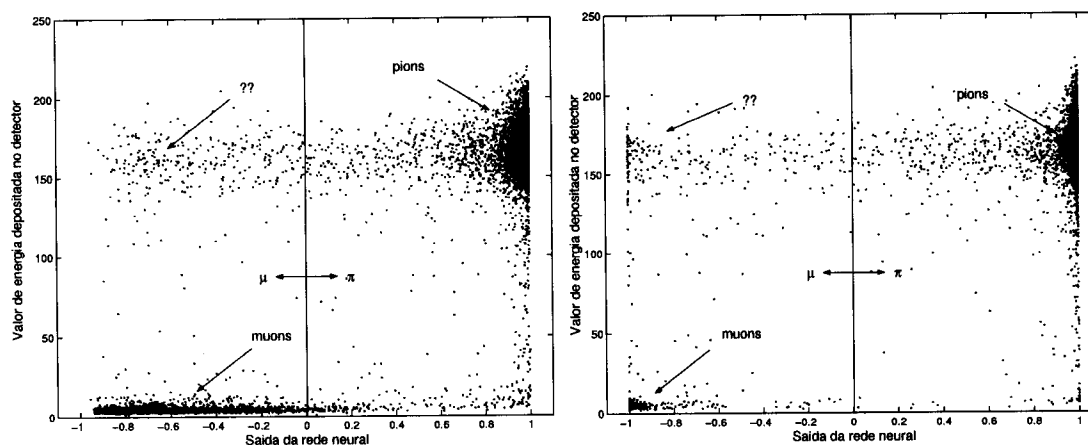


Figura 8.6: Correlação entre a resposta da rede neural (eixos horizontais) com a energia depositada nos módulos (eixos verticais) para a *run* 120324 (esquerda) e a *run* 120326 (direita).

mentada com eventos normalizados pela energia no detector (logo, sem informação de energia absoluta), ela foi capaz de distinguir os conjuntos baseando-se em seus perfis de distribuição de energia. Vemos nos dois casos que os eventos ditos píons pela rede neural (com saída positiva), são os mesmos que possuem um maior nível de energia (logo, píons pela metodologia clássica). Por outro lado, os eventos no lado negativo (logo considerados múons pela rede neural) são os que depositaram menos energia no detector. Finalmente, vemos que em ambos os casos, existe uma certa quantidade de eventos que depositou um grande nível de energia mas foi considerado múon pela rede neural. Marcamos tais eventos com pontos de interrogação. Devemos lembrar, que quando na seção 5.2.1 estudamos as normalizações, vimos que este efeito pode acontecer quando utilizamos a normalização pela energia total. Este efeito pode ser neutralizado pelo uso da normalização pela raiz quadrada ou da coordenada extra para guardar a informação de energia conforme estudamos naquela seção.

Nas Tabelas 8.4 vemos o resultado de concordância para estes conjuntos. Como podemos ver houve um alto nível de concordância com o corte em energia. No caso da *run* 120324 o nível de concordância chegou a 94,21% e da *run* 120326 este mesmo nível chegou a 96,49%. Vemos que existem pequenos erros nos dois casos, que devem ser atribuídos ao fato de estarmos usando a normalização pela energia total.

Método	Rede Neural		Método	Rede Neural	
	π	μ		π	μ
Clássico			Clássico		
π	64,79%	3,43%	π	88,81%	3,09%
μ	2,36%	29,43%	μ	0,42%	7,68%

Tabela 8.4: Concordâncias entre o corte em energia e a rede neural treinada *online* para a *run* 120324 (esquerda, com 94,21% de concordância) e a *run* 120326 (direita, com concordância de 96,49%), ambas relativas a feixes de píons a 180 GeV.

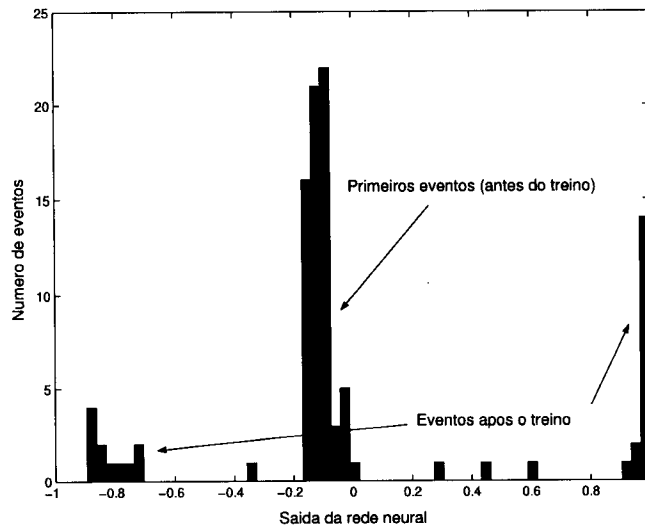


Figura 8.7: Valores de saída da rede neural para os primeiros eventos da *run* 120324. Vê-se que parte dos eventos vai para valores entre -0.2 e 0 e que logo começam a ir para os picos em -0.8 e +1 já caracterizando o efeito do treinamento.

Um outro aspecto importante é analisar como ficam os primeiros 100 eventos, aqueles para os quais a rede ainda não pode ser considerada como treinada. Para tanto, temos na Figura 8.7 as saídas da rede neural para os primeiros 100 eventos da *run* 120324. Vemos que existe um pico com cerca de 60 a 70 eventos com valores entre -0.2 e 0, que foram resultantes de testes com a rede não treinada. Após estes primeiros eventos, começam a aparecer os picos em -0.8 e +1 que são característicos dessa discriminação correta. Isto só ocorre graças ao fato de realizarmos o treinamento de forma *online* em relação a aquisição dos dados. Este primeiros eventos, podem ser forçadamente aceitos numa situação em que cortes de eventos sejam implementados, visando sua análise e retirada posterior em análise *offline*.

Com relação a velocidade de resposta, aumentamos ao máximo a taxa que o feixe poderia produzir. O DAQ com a aplicação neural *online* foi capaz de responder eficientemente com taxas de aproximadamente 2200 eventos por rajada. Cada rajada é um período de 2.3 segundos nos quais o feixe está ativo (depois o feixe fica parado por 11 segundos para possíveis atividades de calibração). Assim, a taxa média obtida foi de 957 eventos por segundo. Um pico com 5502 eventos em um rajada foi registrado (resultando em 2392 eventos em um segundo). Desta forma, o sistema neural pode manter a taxa aproximada do DAQ para a situação em questão. Ao que parece para lidar com um feixe de elétrons que possui taxas mais altas de eventos (podendo atingir mais de 10.000 eventos por rajada), seria necessário se aumentar a velocidade do sistema. Isso poderia se dar através do uso de Power-PCs modelo Rio III que estavam saindo no mercado à época do teste com uma maior frequência de operação e uma revisão do sistema operacional. Entretanto, as placas de *S-Link* para conexão com a eletrônica do calorímetro não eram ainda compatíveis com este modelo de Power-PC.

Assim, algumas experiências como o teste com a rede de três saídas (para incluir a discriminação de elétrons) e o teste com a normalização pela raiz quadrada para se obter uma melhor performance do sistema não foram realizados. Entretanto, o fato de um teste tão significativo quanto o que conseguimos realizar (um treinamento e classificação *online* de uma rede neural) mostra que apenas questões técnicas devem ser resolvidas para que se possa ter um sistema neural treinado *online* pelo método de *backpropagation* mesmo quando contaminações devem ser consideradas. Podemos assim inferir que todas as análises feitas no capítulo anterior de forma *offline* podem ser realizadas de forma *online*.

Capítulo 9

Conclusões

Durante este trabalho, foi apresentado um sistema neural capaz de identificar partículas de um feixe experimental que incidiam sobre um calorímetro em processo de calibração. Durante a geração do feixe, para certas classes de partículas selecionadas, era inerente a produção de eventos de partículas de outras classes. Tal contaminação, naturalmente, dificulta o projeto de um classificador neural, dado que o estabelecimento de um padrão estatístico para cada classe de partículas fica prejudicado pela existência de contaminação.

O sistema neural, mesmo sendo treinado de modo supervisionado e com a etiquetagem eventualmente incorreta, nos eventos de contaminação, consegue contornar esta dificuldade do processo de treinamento, reconhecendo as características básicas do perfil de deposição de energia no calorímetro específico para cada classe de partículas.

A metodologia neural foi criteriosamente comparada com a metodologia classicamente utilizada pelos físicos experimentais. Este método clássico é bastante dependente da energia do feixe de partículas e do ponto de impacto no calorímetro. Além disso, é necessário um conhecimento especializado em calorimetria para o desenvolvimento destes cortes.

Nestes testes de comparação, duas principais vertentes do processamento neural foram estabelecidas. A primeira vertente, utilizando a normalização das entradas da rede pela energia total depositada no detector pela partícula incidente, consegue atingir níveis superiores a 91% de concordância com o método clássico na separação pión/múon e também 91% na separação elétron/pión/múon. Esta nor-

malização, entretanto, poderia produzir tendência estatística na amostra de dados. Por esta causa, uma segunda vertente, baseada na normalização pela raiz quadrada da energia total foi utilizada. Esta versão, dependente da energia, conseguiu reduzir efetivamente as chances de introdução de tendências nos dados e obteve um índice de concordância com os métodos clássicos maior do que a rede com a normalização pela energia. Atingimos mais de 98,5% na separação pión/múon e mais de 95% de concordância na separação elétron/pión/múon.

A introdução da dependência com relação a energia, pela utilização da normalização pela raiz quadrada da energia total, tem, entretanto, um revés demonstrado em nossa análise de sensibilidade da rede: o método dependente da energia praticamente deixa de funcionar se a energia do feixe de partículas variar muito, por exemplo, de 20 para 100 GeV. A normalização pela soma sofre menos com estes efeitos.

Também ampliamos os limites de nossa aplicação ao realizarmos o estudo de parâmetros para treinamento da rede neural. Um número considerável de treinamentos foi realizado para cada conjunto de parâmetros com diferentes inicializações da rede, visando uma maior garantia sobre os resultados. Tal estudo permitiu verificar que mesmo aplicando diferentes normalizações às entradas da rede, existe uma clara dependência da eficiência de identificação do classificador com relação aos parâmetros de treinamento.

Na separação pión/múon conseguimos estabelecer um nível mínimo de concordância entre o método neural e o método clássico superior a 97% (normalização pela energia total) e 99% (pela normalização pela raiz quadrada da energia), sendo definidas faixas no espaço de parâmetros para as quais se garante tais níveis de concordância. A vantagem de uma alta concordância com o método clássico reside no conhecimento físico que foi utilizado para definição deste método.

Na separação elétron/pión/múon também foi estabelecido um limite para o valor mínimo de concordância que podemos garantir. Neste caso, o máximo valor mínimo para normalização pela energia total ficou em 92%. Para a normalização pela raiz quadrada da energia total, este valor também ficou em 92%, sendo que a faixa em que é mantido este valor é bem mais estreita (na verdade se resume a alguns poucos pontos) do que no caso da rede com normalização pela energia

total. Entretanto, em termos do máximo valor de concordância, o valor obtido pela rede neural com normalização pela energia ficou com 96%, um valor inferior aquele obtido com a normalização pela raiz quadrada da energia total, 97%. Isto nos permite concluir que a normalização pela raiz quadrada da energia consegue atingir melhores valores, mas tem uma maior variação em seus valores, podendo, inclusive, ficar com um valor de concordância entre o método neural e clássico menor do que a rede neural usando normalização pela energia total.

Também pudemos verificar, tanto na análise de parâmetros quanto na posterior análise de Componentes Principais de Discriminação, que se o treinamento da rede continua por muitos ciclos com uma alta taxa de aprendizado, pode ocorrer que os tipos de partículas treinados com dois valores alvos diferentes (o seu alvo original e o alvo da classe a qual este tipo contamina), podem vir a tender à média entre estes dois alvos. Isto acontece numa tentativa do método de *backpropagation* de acomodar sua minimização do erro médio quadrático da rede com relação aos dois alvos. De fato, o erro médio quadrático para treinamento diminui quando as saídas sofrem este efeito, mas os resultados de classificação pioram muito. Este problema aparece mais claramente no treinamento da rede com normalização pela raiz quadrada. Neste caso, fica mais óbvia a existência de um mesmo tipo com diferentes alvos. Por esta razão, o controle da taxa de aprendizado deve ser realizado pela época e a queda na taxa de aprendizado.

Estudamos também esta tendência da rede de arrastar o valor das saídas para a média dos dois valores alvos relativos a uma dada classe contaminante, por exemplo, múons que são treinados com o alvo para múons de feixe e para múons no feixe de píons. Foi possível encontrar uma clara relação entre este arrasto e o nível de contaminação.

O método neural também foi aplicado a dados simulados, o que nos permitiu verificar um pouco melhor o nível de acerto obtido pelas rede neurais. Os métodos clássicos, utilizados como medida de acuidade, possuem regiões de dúvida bastante significativas, o que nos levou ao teste com dados simulados quando sabemos exatamente a classe de cada evento estudado. Assim, é possível se avaliar tanto o processamento neural quanto a classificação por métodos clássicos. Vemos que o método neural atinge elevados valores de acerto. Podemos verificar também que,

para a faixa de alta energia (100 e 180 GeV), o método clássico tem uma eficiência de acerto ligeiramente superior ao método neural. A baixas energias (20 GeV), entretanto, sem o apoio do contador Cherenkov, a eficiência do método clássico caiu muito, enquanto a eficiência do método neural ficou bem superior. Isto demonstra uma grande vantagem do método neural.

Foi realizada também a análise neural sobre os módulos da extensão do barril do calorímetro. Esta análise permite verificarmos que a metodologia neural é independente da topologia do detector em si, bastando, apenas, que haja suficiente detalhamento da informação nas entradas do detector para que as classes sejam minimamente separáveis.

Também desenvolvemos diferentes arranjos da rede neural visando compactar a informação de entrada, o que poderia levar a uma rede mais precisa (dado o fato de que, ao combinarmos as entradas, podemos reduzir o ruído de fundo inerente a cada uma delas) e de menores dimensões (o que, aumenta a velocidade de treino e operação da rede). Os resultados comprovam o funcionamento da técnica de pré-processamentos topológicos baseados em torres, camadas, anéis de células do calorímetro, entre outros, que exploram o perfil de deposição de energia para diferentes partículas no detector. Em todos os tipos de pré-processamento as eficiências de classificação ficaram bem elevadas, acima de 98% para a separação pión/múon e acima de 93% para a separação elétron/pión/múon. Também realizamos a decomposição em Componentes Principais de Discriminação, quando uma redução drástica no número de neurônios da camada escondida foi obtida. Também aí discutimos a questão em que um treinamento muito longo pode minimizar o erro médio quadrático e levar a um maior erro de classificação.

Uma outra técnica estudada para pré-processamento foi o uso das redes especialistas. Uma rede foi projetada para trabalhar com os dados do módulo principal do calorímetro (Módulo 0) e a outra para trabalhar com os 5 módulos do protótipo anterior, que são, por vezes, utilizados para conter vazamentos. A combinação das duas redes pode melhorar seus resultados individuais. Entretanto, caso um dos resultados individuais não seja muito bom, o classificador final fica prejudicado.

Ainda no âmbito desta tese, dada a dispersão geográfica dos pesquisadores da colaboração internacional relacionada ao TileCal, foi desenvolvido o Discriminador

Neural Remoto de Partículas. Este sistema, baseado em páginas da *Web*, permite que o sistema neural seja utilizado por pesquisadores no mundo todo. Isto facilita a divulgação do método e disponibiliza a metodologia como ferramenta para estudo e descontaminação dos dados para qualquer pesquisador interessado em ter conjuntos de dados livres da contaminação de feixe.

Finalmente, tivemos a oportunidade de implementar e usar um sistema neural capaz de ser treinado e operar *online*. O sistema etiquetou efetivamente os eventos, realizando a separação pión/múon com eficiência conforme comparações entre a saída da rede e o valor de energia depositada no detector, concordando em mais de 94% com este critério. Pudemos, inclusive, observar, como foi usada a normalização pela energia total, que temos a criação da tendência estatística característica desta normalização. Análises de correlação provaram que eventos de píons (com alta energia) depositando grande parte de sua energia na terceira camada de células do calorímetro estavam sendo erroneamente consideradas como múons.

A integração com o sistema de aquisição para testes de calibração do calorímetro foi realizada no Read-Out Driver. O sistema neural instalado pode ser treinado e operar *online*, usando uma combinação de programação *multithread* em C++ com interrupções e transmissão de dados no barramento VME. O sistema neural foi capaz de produzir sua classificação sem comprometer a taxa normal de operação do sistema de aquisição. Como o sistema *online* reproduziu resultados semelhantes aos encontrados na análise *offline*, concluímos que o experimento foi um sucesso, e que podemos estender os resultados que obtivemos na análise *offline* para a continuação de um trabalho de análise *online*.

Dada a quantidade de tópicos analisados, é impossível conseguirmos abarcar todas as possibilidades de análises que esta tese nos apresenta. Assim, alguns detalhes poderiam constituir uma continuação interessante deste trabalho. Podemos citar como um destes tópicos a análise neural com método de treinamento não supervisionado. Possivelmente, este método levaria a um classificador neural bastante preciso, já que não sofreria dos mesmos problemas de falsa etiquetagem (métodos não-supervisionados não dependem de etiquetagem) que temos com o método de *backpropagation*. A combinação entre um método não supervisionado e o método de *backpropagation* também poderia ser uma alternativa.

Outro tópico interessante seria a aplicação do sistema neural *online* para a separação elétron/píon/múon, a qual não conseguimos realizar durante os curtos espaços de tempo de teste com feixes. Também em termos de análise neural *online*, poderíamos aplicar algumas das técnicas de pré-processamento para aumentar a velocidade do sistema neural ao reduzir seu número de entradas.

Uma outra proposta seria realizar o estudo de parâmetros utilizando como pré-processamento à rede neural os métodos topológicos aqui estudados. Talvez esta exploração fornecesse um classificador mais compacto e com níveis mais altos de eficiência de identificação dos que os obtidos nesta tese.

Uma análise interessante para processamento é utilização de filtragem casada sobre os sinais do calorímetro. No caso, o casamento seria com os sinais de energia para as células no detector. Testes com relação a esta filtragem foram inicializados e os primeiros resultados se mostraram promissores, quando são estudados os patamares de decisão do filtro em relação à concordância entre os resultados do mesmo e métodos clássicos. Sua exploração, entretanto, por não se relacionar com sistemas neurais foge ao escopo desta tese.

Um outro ponto a ser discutido seria a aplicação da metodologia aqui estudada a outros calorímetros. Por exemplo, poderíamos tentar aplicar nossa metodologia ao calorímetro de Argônio Líquido do ATLAS. Outros experimentos, independentes do CERN poderiam ser, também, considerados.

Métodos de descrição dos conjuntos de dados como Componentes Principais (PCA) ou Componentes Principais Não-Lineares (NLPCA), poderiam ser explorados. Uma primeira rede extrairia as características dos conjuntos de dados enquanto uma outra, posteriormente, realizaria a classificação.

Referências Bibliográficas

- [1] Large Electron-Positron Collider, “LEP”, 2002.
<http://publicold.web.cern.ch/publicold/ACCELERATORS/LepAcc.html>.
- [2] The ATLAS Collaboration, “Technical Proposal for a General-Purpose pp Experiment at the Large Hadron Collider at CERN”, 1994.
- [3] AMARAL, P., AMORIM, A., ANDERSON, K., *et al.*, “Hadronic Shower Development in Iron-scintillator Tile Calorimetry”, *Nuclear Instruments and Methods*, v. A443, pp. 51–70, 2000.
- [4] EISBERG, R., RESNICK, R., *Física Quântica - Átomos, Moléculas, Sólidos, Núcleos e Partículas*. Rio de Janeiro, Editora Campus, 1979. ISBN 85-7001-309-4.
- [5] CERN Particle Data Group, “Standard Model”, 2002. <http://pdg.lbl.gov>.
- [6] CARRERAS, R., *Cuando la Energia se Transforma en Materia... - Un vistazo al mundo de las particulas*. CERN, 1987.
- [7] Revista da Faperj, *Cientistas Brasileiros investigam a origem do Universo*, dezembro 2001. No. 21.
- [8] The European Laboratory for Particle Physics, “CERN”, 2002.
<http://www.cern.ch>.
- [9] BERNERS-LEE, T., *Hypertext Transfer Protocol - HTTP 1.0*, Report, Network Working Group - CERN, 1996.
- [10] FRANÇA, M. S. J., HEYMANN, G., “A Luz dentro do Túnel”, *Revista Super Interessante*, v. 3, n. 11, pp. 60–66, Novembro 1989.

- [11] “LEP Achieves Record Energy Levels”, *CERN Courier*, v. 40, n. 5, pp. 5, Junho 2000. <http://www.cerncourier.com/main/article/40/5/1>.
- [12] “The Final Curtain falls on LEP”, *CERN Courier*, v. 41, n. 1, Janeiro 2001. <http://www.cerncourier.com/main/article/41/1/2>.
- [13] “LEP gets a stay of Higgs Execution”, *CERN Courier*, v. 40, n. 9, Novembro 2000. <http://www.cerncourier.com/main/article/40/9/1>.
- [14] Large Hadron Collider, “LHC”, 2002. <http://www.cern.ch/lhc>.
- [15] “Going into the Cold : LHC system reach an important milestone.”, *CERN Courier*, v. 41, n. 10, dezembro 2001. <http://www.cerncourier.com/main/article/41/10/1>.
- [16] TAYLOR, B., “Timing, Trigger and Control (TTC) System for LHC”, December 2002. <http://ttc.web.cern.ch/TTC/intro.html>.
- [17] A Toroidal Lhc ApparatuS, “ATLAS”, 2002. <http://www.cern.ch/Atlas>.
- [18] Compact Muon Solenoid, “CMS”, 2002. http://cms-ultmb.web.cern.ch/CMS_TMB/PERSONAL/dbesson/welcome%20to%20the%20compact%20muon%20solenoid%20outreach%20activities.htm.
- [19] ATLAS Collaboration, “ATLAS High Level Triggers, DAQ and DCS”. In: *CERN/LHCC/2000-17*, Geneva, March 2000.
- [20] CERQUEIRA, A. S., SEIXAS, J. M., CALOBA, L. P., “An Active Adder for Building the Trigger Signal of the Hadronic Calorimeter of ATLAS”. In: *XXII Encontro Nacional de Física de Partículas e Campos*, São Lourenço, 2001.
- [21] ATLAS Collaboration, “First-level trigger technical design report”. In: *CERN/LHCC/98-14*, Genebra, 1998.
- [22] BOGAERTS, J., BOTTERILL, D., LI, W., *et al.*, “Performance Analysis of the ATLAS Second Level Trigger Software”. In: Sanchis, E., Ferrer, A., González, V. (eds.), *12th IEEE Real Time Congress on Nuclear and Plasma Science*, Valencia, Espanha, Junho 2001.

- [23] G. Ambrosini *et al.*, “The ATLAS DAQ and Event Filter Prototype ~-1~ Project”, 1996. <http://atddoc.cern.ch/Atlas/Conferences/CHEP/ID388/ID388-1.html>.
- [24] LEHMANN, G., FRANCIS, D., JOOS, M., *et al.*, “The ATLAS DAQ/EF Prototype-1 as Testbeam Data Acquisition System”. In: Sanchis, E., Ferrer, A., González, V. (eds.), *Proceeding of the 12th IEEE Real Time Congress on Nuclear and Plasma Sciences*, Valencia - Espanha, Junho 2001.
- [25] SEIXAS, J. M. D., DAMAZIO, D. O., “A Neural Particle Discriminator for Calorimetry in High Energy Physics”. In: *Congresso Brasileiro de Redes Neurais*, Florianópolis, 1997.
- [26] GRENIER, P., “Response versus Temperature - <http://documents.cern.ch/cgi-bin/setlink?base=agenda&categ=a01700&id=a01700s1t4/transparencies>”, Outubro 2001.
- [27] Chicago Group to the ATLAS Collaboration, “Fabrication and Assembly Details for Mother Board Section 1”. http://hep.uchicago.edu/atlas/electr/MB_prod/Sec.1/details.1.html.
- [28] Chicago Group to the ATLAS Collaboration, “TileCal Electronics - <http://hep.uchicago.edu/atlas/electr/electronics.html>”.
- [29] BOHM, C., “Design Review of Digitizer Version 2”. <http://www.physto.se/~ker/designreview/dr.html>.
- [30] CERQUEIRA, A. S., “Analogue Summation for the Trigger Towers”. <http://www.lps.ufrj.br/~cern>.
- [31] HENRIQUES, A., “Status of the TileCal Cooling Activity - <http://documents.cern.ch/cgi-bin/setlink?base=agenda&categ=a01341&id=a01341s1t5/transparencies>”, Maio 2001.
- [32] AJALTOUNI, Z., ALBIOL, F., ALIFANOV, A., *et al.*, “Response of the Atlas Tile Calorimeter Prototype to Muons”, *Nuclear Instruments and Methods*, v. A388, pp. 64–78, 1997.

- [33] WIGMANS, R., “Advances in hadron calorimetry”, *Review in Nuclear Particle Science*, v. 41, pp. 131, 1991.
- [34] GREEN, D., *The Physics of Particle Detectors*. New York, Estados Unidos, Cambridge U. Press, 2000. ISBN-0521-6622-5.
- [35] BIJ, E. V. D., “<http://hsi.web.cern.ch/HSI/s-link/products.html>”.
- [36] MCLAGAN, M., “Linux”, 1994. <http://www.linux.org>.
- [37] Creative Electronic Systems S.A., “Lynx-OS Instalation & Configuration for CES PowerPC based systems. User’s Manual, version 3.0”, 1997.
- [38] FATHOUAT, P., GÄLLNÖ, P., “TTC-VMEbus INTERFACE - TTCvi-MkII”. In: *CERN-EP-ATE*, Genebra, 2000. <http://www.cern.ch/TTC/TTCviSpec.pdf>.
- [39] The ATLAS TDAQ Collaboration, “ATLAS Online Software”. <http://atlas-onlsw.web.cern.ch/Atlas-onlsw>.
- [40] VILLAS-BOAS, S. B., *C++ e Orientação a Objetos em Ambiente Multiplataforma*, 2002. <http://www.del.ufrj.br/~villas/livro.c++.html>.
- [41] W3C® (MIT, INRIA, Keio), “Extensible Markup Language (XML)”. <http://www.w3.org/XML/>.
- [42] KOLOS, S., “Implementation of the Information System: Users Guide”, 2000. <http://atddoc.cern.ch/Atlas/Notes/037/Note037-1.html>.
- [43] BURCKHART, D., CAPRINI, M., SIMONE, A. D., *et al.*, “MRS - Message Reporting System ATLAS DAQ/EF-1”, 2000. http://lnxatd01.cern.ch/Atlas/Notes/059/mrs_ug.html.
- [44] The ATLAS TDAQ Collaboration, “ATLAS DAQ Backend Software Repository”. <http://atddoc.cern.ch/srt-bin/srt-build.pl>.
- [45] The ATLAS TDAQ Collaboration, “The Event Format in the ATLAS DAQ/EF Prototype ~-1~”. <http://atddoc.cern.ch/Atlas/Notes/Note050/Note050-1.html>.

- [46] RAMSTEDT, M., “A Description of the Tile DMU Readout Format”.
<http://www.physto.se/~mank/rf/readoutformat.html>.
- [47] CAPRINI, M., FRANCIS, D., JONES, R., *et al.*, “User Requirements for the Online Monitoring of the ATLAS DAQ/EF Prototype “-1””, 2000.
<http://atddoc.cern.ch/ATLAS/Notes/152/Note152-1.html>.
- [48] DAMAZIO, D. O., PETERSEN, J., TREMBLET, L., “An Online Monitoring Task to Produce Histograms at the TileCal Test Beam”, 2001. [http://edmsoraweb.cern.ch:8001/cedar/doc.info?cookie=782781 &document_id=320880&version=1](http://edmsoraweb.cern.ch:8001/cedar/doc.info?cookie=782781&document_id=320880&version=1).
- [49] CERN-CN Division, “The ZEBRA System”.
<http://wwwinfo.cern.ch/asdoc/zebra.html3/zebramain.html>.
- [50] CERN IT GROUP, “HBook manual”, 2000.
<http://wwwinfo.cern.ch/asdoc/higz/HIGZMAIN.html>.
- [51] BRUN, R., “ROOT Analysis Tool”, 2002. <http://root.cern.ch/>.
- [52] Trolltech Company, “QT Library”, 2002. <http://www.trolltech.com/products/qt/>.
- [53] DAMAZIO, D. O., “TileCal DAQ”, 2001. [http://documents.cern.ch/cgi-bin/setlink?base=agenda &categ=a01704&id=a01704s1t1/transparencies](http://documents.cern.ch/cgi-bin/setlink?base=agenda&categ=a01704&id=a01704s1t1/transparencies).
- [54] DAMAZIO, D. O., JOOS, M., PETERSEN, J., *et al.*, “Event Sampling in the ROD Crate.”, agosto 2001.
http://edmsoraweb.cern.ch:8001/cedar/doc.info?cookie=782781&document_id=320878&version=1.
- [55] DAMAZIO, D. O., JOOS, M., PETERSEN, J., *et al.*, “User’s Guide to the RODIPC library”, agosto 2001.
http://edmsoraweb.cern.ch:8001/cedar/doc.info?cookie=782781&document_id=320874&version=1.
- [56] HAYKIN, S., *Neural Networks, A Comprehensive Foundation*. 2 ed. Prentice-Hall, 2001.

- [57] L. LÖNNBLAD, C. PETERSON, T. R., “Pattern recognition in high energy physics with artificial neural networks - JetNet 2.0”, *Computing Physics Communications*, v. 70, pp. 167, 1992.
- [58] HERTZ, J., KROGH, A., PALMER., R., *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1990.
- [59] SEIXAS, J. M. D., DAMAZIO, D. O., “A Neural Discriminator Capable to Identify Impurities in the Data Sample”, *5th IEEE International Conference on Electronics, Circuits and Systems*, pp. 261–264, 1998. Lisboa, Portugal.
- [60] Gruber, A. *et al*, “Performance of Backpropagation Networks in the Second-level Trigger of the H1-experiment”, *New Computing Techniques in Physics Research III*, pp. 429–434, 1994.
- [61] DAYHOFF, J., *Neural Network Architectures - An Introduction*. New York - Estados Unidos, Van Nostrand Reinhold, 1990. Capítulo 10 - CounterPropagation.
- [62] Lund University, “PYTHIA (and JETSET) webpage”, <http://www.thep.lu.se/tf2/staff/torbjorn/Pythia.html>.
- [63] IT/ASD/Simulation Section, “GEANT - Detector Description and Simulation Tool”, 1995. http://wwwinfo.cern.ch/asdoc/geant_html3/geantall.html.
- [64] MANGANO, M., “Monte Carlo for LHC”, 2002. <http://lhc-monte-carlo.web.cern.ch/lhc-monte-carlo/>.
- [65] SOLODKOV, A., “G3, G4, Fluka & Status”, 2001. <http://documents.cern.ch/cgi-bin?setlink?base=agenda&categ=a01700&id=a01700s2t10/transparenties>.
- [66] LACERDA, F., *Discriminador Neural*. Rio de Janeiro, 1998. http://www.lps.ufrj.br/~neural_w/.
- [67] Netscape Communications Corp., “JavaScript Guide - Netscape Navigator 4.0”, 1996.
- [68] Netscape Communications Corp., “JavaScript Reference”, 1997.

- [69] Coak K. A. L., *et al.*, “The Common Gateway Interface Version 1.1”, *The Apache Group*, , 1998.
- [70] “The Common Gateway Interface”, <http://hoo.hoo.ncsa.uiuc.edu:80/cgi/overview.html>.
- [71] “The WDVL: CGI: The Common Gateway Interface for Server-side Processing”, <http://www.wdvl.com/Authoring/CGI>.
- [72] Physics Analysis Workstation, “PAW”, 2002. <http://www.cern.ch/paw>.
- [73] CALOBA, L., SEIXAS, J., PEREIRA, F., “Neural Discriminating Analysis for a Second-Level Trigger System”. In: *Computing in High Energy Physics*, Rio de Janeiro, 1995.
- [74] OJA, E., *Neural Network for Chemical Engineers*. Elsevier Science, 1995.
- [75] DAMAZIO, D. O., SOARES FILHO, W., SEIXAS, J. M. D., “Efeito da Estimativa do Ruído Ambiente na Classificação com Redes Neurais do Ruído Irradiado por Navios”. In: *Congresso Brasileiro de Automática*, 2000.
- [76] Analog Digital Inc., “DSP Products”, <http://www.analog.com>.
- [77] LEROY, X., “The Linux Threads Library”, <http://pauillac.inria.fr/~xleroy/linuxthreads/>.
- [78] Creative Electronic Systems S.A., “Rio II Boards products”, <http://www.ces.ch>.
- [79] J.A.C Bogaerts *et al.*, “Operating System Services Interfaces”, 1998. <http://atlas.web.cern.ch/Atlas/project/LVL2testbed/www/docxx/OperatingSystemServices.html>.

Apêndice A

Artigos e Notas Técnicas

No decorrer deste trabalho vários artigos e notas técnicas foram escritos. Na seqüência listaremos os artigos com um pequeno texto explicativo.

Artigos Publicados em Revista :

1. "An Online Neural Network Triggering System for the Tile Calorimeter". Denis Oliveira Damazio, José Manoel de Seixas e Paulo Vitor Magacho. IEEE Transactions on Nuclear Science. V. 49, No. 2, Abril 2002.

Este artigo traz a proposta do sistema neural a ser utilizado *online* no sistema de aquisição do calorímetro de telhas. A metodologia a ser utilizada e a inserção do sistema neural são abordados.

2. "An Online Calorimeter Trigger for Removing Outsiders from Particle Beam Calibration Tests" Denis Oliveira Damazio e José Manoel de Seixas. Aprovado para publicação na Revista *Nuclear Instruments and Methods*, na edição especial do congresso Advanced Computing and Analysis Techniques. 2002.

Este artigo traz os resultados obtidos através da análise *online*. A inserção do classificador neural no sistema de aquisição de dados e comparação com métodos clássicos para garantir o acerto são apresentados.

Notas Técnicas publicadas no CERN :

1. "Data Compression on Tilecal ROD". D. O. Damazio. <http://edms.cern.ch/document/339395/1>. janeiro, 2002.

Esta nota expõe a proposta de um mecanismo para suprimir dados que não

tragam nenhuma informação relevante (zero supress) do sistema de aquisição do calorímetro hadrônico.

2. "User's Guide to the RODIPC Library". D. O. Damazio, M. Joos, J. Petersen, L. Tremblet. http://edmsoraweb.cern.ch:8001/cedar/doc.info?cookie=782781&document_id=320874&version=1. agosto,2001.

Esta nota traz a descrição da interface C++ para um sistema de transmissão de dados através do barramento VME. Transmissões do tipo single cycle e DMA são abordadas.

3. "Event Sampling in the ROD Crate". D. O. Damazio, M. Joos, J. Petersen, L. Tremblet. http://edmsoraweb.cern.ch:8001/cedar/doc.info?cookie=782781&document_id=320878&version=1. agosto, 2001.

Esta nota descreve uma análise de desempenho da biblioteca C++ para transmissão de dados através do barramento VME.

4. "An Online Monitoring Task to Produce Histograms at the TileCal Test Beam". D. O. Damazio, J. Petersen, L. Tremblet. http://edmsoraweb.cern.ch:8001/cedar/doc.info?cookie=782781&document_id=320880&version=1. agosto,2001.

Esta nota descreve o sistema de monitoração utilizado no CERN durante os testes com feixes de partículas. Este sistema era parte do sistema de aquisição de dados e facilitava a calibração e ajustes do feixe.

Artigos Publicados em Congressos :

1. "Neural Networks to Identify Particles Using Topological Properties of Calorimeters". D.O. Damazio e J.M. Seixas. XIV Congresso Brasileiro de Automática. Natal, Rio grande do Norte. 2002.
Este artigo traz a decomposição em torres para o calorímetro hadrônico como entrada para a rede neural. Análises de comparação com o método clássico e de relevância são apresentadas para assegurar os resultados da rede neural.
2. "Neural Network Discrimination Using Calorimeter Segmentation Preprocessing". D.O. Damazio e J.M. Seixas. XXIII Encontro Nacional de Física de Partículas e Campos. Águas de Lindóia, São Paulo. 2002.

Este artigo descreve a aplicação do sinal vindo das camadas de células do detector como entradas para o processamento neural. É enfatizada a redução de dimensionalidade como fator de redução de ruído de fundo do experimento e ganho de velocidade de processamento.

3. "Um Classificador On-Line para a Física de Partículas". D.O. Damazio, P.V. Magacho e J.M. Seixas. V Congresso Brasileiro de Redes Neurais. 2001.

Este artigo trata das questões relativas a tempo de treino para uma rede neural. Este quesito é básico para a operação *online* do sistema.

4. "Wavelet Transform as a Preprocessing Method for Neural Classification of Passive Sonar Signals". J.M. Seixas, D.O. Damazio, P.S.R. Diniz e W. Soares Filho. IEEE International Conference on Electronics, Circuits and Systems. Saint-Julians. Estados Unidos. 2001.

Este artigo traz a decomposição do sinal de sonar passivo em componentes *wavelet*. Esta decomposição é usada como pré-processamento para um sistema neural classificador de sons de contactos marinhos (navios, submarinos, etc).

5. "Efeito da Estimção do Ruído Ambiente na Classificação com Redes Neurais do Ruído Irradiado por Navios". D.O. Damazio, J.M. Seixas e W. Soares Filho. XIII Congresso Brasileiro de Automática. Florianópolis, Santa Catarina. 2000.

Este artigo trata da descrição de um método para estimção do ruído de fundo quando se lida com sinais de sonar passivo. Os efeitos desta estimção na aplicação neural para classificação de navios é observado.

6. "Um Classificador Neuronal Compacto e Eficiente com Capacidade de Identificar Contaminação em Dados Experimentais. D.O. Damazio, A.C. Soares e J.M. Seixas. XIII Congresso Brasileiro de Automática. Florianópolis, Santa Catarina. 2000.

Este artigo trata da aplicação das Componentes Principais de Discriminação às informações produzidas pelas extensões do barril.

7. "A Compact Online Neural System for Classifying Passive Sonar Signals". J.M. Seixas, W. Soares Filho, J.B.O. Souza Filho, D.O. Damazio, N.N. Moura.

International Conference on Signal Processing Application and Technology, Orlando, Florida, Estados Unidos. 1999. <http://www.lps.ufrj.br/~damazio/artigos/icspat992.ps.gz>.

Este artigo traz a descrição do trabalho desenvolvido com apoio da marinha brasileira para a implementação de um sistema neural para classificação de contactos marinhos (navios, submarinos, entre outros) a partir de sinais de sonar passivo.

8. "Um Identificador Neural Compacto de Contactos". D.O. Damazio, W.S. Filho, J.M. Seixas. Apresentado no IV Encontro de Tecnologia em Acústica Submarina, Rio de Janeiro. 1999. <http://www.lps.ufrj.br/~damazio/artigos/ETAS99.zip>

Este trabalho traz os resultados obtidos com a classificação por redes neurais aplicadas ao sinal de sonar passivo.

9. "Particle Discriminator Using Matched Filters and Expert Neural Networks". W.S. Filho, D.O. Damazio, J.M. Seixas. IEEE International Conference on Electronics, Circuits and Systems. Paphos, Chipre. 1999. <http://www.lps.ufrj.br/~damazio/artigos/icecs924.ps.gz>

Este artigo traz uma análise dos sinais do calorímetro usando filtros casados e redes neurais especialistas.

10. "Um Discriminador Neural em Física de Altas Energias Usando Pré-processamento". D.O.Damazio, J.M. Seixas, B.O.B. Corrêa. II Encontro Nacional de Inteligência Artificial, Rio de Janeiro. 1999. http://www.lps.ufrj.br/~damazio/artigos/enia_ex.ps.gz

Este trabalho traz uma análise inicial de diferentes pré-processamentos abordados para reduzir a quantidade de dados apresentados a rede neural.

11. "Um Discriminador de Partículas de Altas-Energias Baseado em um Calorímetro Projetivo". D.O.Damazio, J.M. Seixas. IV Congresso Brasileiro de Redes Neurais. São José dos Campos, SP. 1999. <http://www.lps.ufrj.br/~damazio/artigos/cbrn991.ps.gz>.

Este artigo apresenta a análise neural para as extensões do barril.

12. "A Remote Neural Discriminator for the Scintillating Tile Calorimeter of ATLAS". J.M. Seixas, C. Maidantchik, D.O. Damazio, F. Lacerda. VIII International Conference on Calorimetry in High Energy Physics. Lisbon, Portugal. 1999.
<http://www.lps.ufrj.br/~damazio/artigos/icchep99.ps.gz>
Este artigo apresenta um exemplo de análise *offline* que pode ser realizado com o Discriminador Neural Remoto de Partículas.
13. "Outsider Identification in a Neural Particle Discriminator Based on Calorimetry". D.O. Damazio, J.M. Seixas. Artificial Intelligence on High Energy Physics. Creta, Grécia. 1999.
<http://www.lps.ufrj.br/~damazio/artigos/aihenp299.ps.gz>
Este artigo apresenta a identificação *offline* de partículas de contaminação no feixe de outras partículas. Possíveis implementações *online* são discutidas.
14. "A Neural Discriminator Capable to Identify Impurities in the Data Sample". J.M. Seixas, D.O. Damazio. 5th IEEE International Conference on Electronics, Circuits and Systems, pp 261-264, Lisboa, Portugal. 1998.
<http://www.lps.ufrj.br/~damazio/artigos/icecs982.ps.gz>
Este trabalho discute também a identificação *offline* de partículas. Neste caso o calorímetro protótipo de 5 módulos é usado. Análise de correlação com métodos clássicos são realizadas.
15. "High Energy Particle Identification Based on Calorimetry and Neural Networks". J.M. Seixas, D.O. Damazio. Congresso Brasileiro de Automática, pp 331-335, Uberlândia, Brasil. 1998.
<http://www.lps.ufrj.br/~damazio/artigos/cba981.ps.gz>
Este trabalho descreve o problema de classificação elétron/píon/múon *offline* quando as contaminações são retiradas através de métodos clássicos. As correlações com métodos clássicos são realizadas para demonstrar a capacidade do método.
16. "Mapeamentos Topológicos para um Classificador Neuronal de Partículas Baseado no Calorímetro Hadrônico de Telhas Cintilantes". D.O. Damazio, J.M. Seixas, B.O. Brasil. XIX Encontro Nacional de Física de Partículas e Campos,

pp 33, Caxambú, Brasil. 1998.

Este trabalho descreve as várias possíveis análises topológicas que podem ser realizadas com os dados do calorímetro hadrônico de telhas cintilantes.

17. "A Neural Particle Discriminator for Calorimetry in High Energy Physics". J.M. Seixas, D.O. Damazio. III Congresso Brasileiro de Redes Neurais, Florianópolis. 1997.

<http://www.lps.ufrj.br/~damazio/artigos/cbrn972.ps.gz>.

Este é o primeiro artigo em que a metodologia neural é aplicada a protótipos de calorímetro. No caso a separação elétron/píon era realizada (as contaminações de múons eram retiradas por corte em energia).

Apêndice B

Código Neural usado para os estudos *offline*

Neste apêndice apresentaremos o código neural usando nos estudos *offline*. Este código foi usado como base para vários estudos desta tese como, por exemplo, o código para diferentes normalizações, alguns estudos de parâmetros, a análise por decomposição em Componentes Principais de Discriminação, Redes Especialistas entre outros. As rotinas de aquisição de dados são do pacote Hbook [50] e as rotinas de treino da rede são do pacote JetNet-2.0 [57].

```
PROGRAM pos3

PARAMETER(maxpat=30000)
PARAMETER(MAXI=1000, MAXO=1000)
PARAMETER(INPUT=46)
PARAMETER(INPUT5m=200)
PARAMETER(nvar=46)

COMMON/PAWC/H(500000)
DIMENSION TR_PAT(MAXPAT,INPUT), TR_DESO(MAXPAT,3)
DIMENSION TST_PAT(MAXPAT,INPUT), TST_DESO(MAXPAT,3)
DIMENSION X(246)

COMMON/JNINT1/O(2000),A(2000),D(2000),T(2000),DT(2000),
&W(150000),DW(150000),NSELF(150000),NTSELF(2000)
```

```

COMMON/JNDAT1/MSTJN(40),PARJN(40),MSTJM(20),PARJM(20),
&      OIN(MAXI),OUT(MAXO),MXNDJM

COMMON/JNDAT2/TINV(10),IGFN(10),ETAL(10),WIDL(10),SATM(10)

COMMON/SLATE/ISL(40)
DOUBLE PRECISION ISEED

CHARACTER*80 TRAIN_FILE_0, TRAIN_FILE_1, TRAIN_FILE_2
CHARACTER*80 TEST_FILE_0, TEST_FILE_1, TEST_FILE_2

CHARACTER*80 WEIGHT_FIL_NAME
CHARACTER*80 CONFIG_FIL_NAME

CHARACTER*80 DUMP_FIL_NAME
CHARACTER*80 ERRTRN_FIL_NAME
CHARACTER*80 ERRTST_FIL_NAME
CHARACTER*80 OUTNET_FIL_NAME
CHARACTER*80 TRN_FIL_NAME
CHARACTER*80 TST_FIL_NAME

INTEGER*4 NCYCLES, NTSTFREQ, STEP
INTEGER*4 NTRPATS_0, NTRPATS_1, NTRPATS_2
INTEGER*4 NTSTPATS_0, NTSTPATS_1, NTSTPATS_2
INTEGER*4 IPAT, ICYCLE, INODE
INTEGER*4 ntuple1, ntuple2, ntuple3

REAL INP_MULT, INP_SHIFT
REAL OUT_CUT_TST
DOUBLE PRECISION INP_MULT5m

INTEGER*4 ITA, ITB, ITRA, ITRB, ICF, ION, IWEI, IWEI2, ISCR, IER

```

```

INTEGER*4 ITST, ITR
include 'hcwn.inc'
  Equivalence(x(1), s1m0(1))
  Equivalence(x(21), s2m0(1))
  Equivalence(x(39), s3m0(1))

CALL HLIMIT(200000)

C  /*  Units Number  */
ICF  = 22
ITRA = 24
ITRB = 26
ITA  = 28
ITB  = 30
ION  = 40
IWEI = 20
IWEI2 = 10
ISCR = 50
IER  = 60
ITRN = 70
ITST = 71

C//*****
C//.....Configure the network.....
  CONFIG_FIL_NAME='303.cnf'
  PRINT*, 'READ CONFIGURATION FILE ', CONFIG_FIL_NAME, '...'
  OPEN(ICF, FILE=CONFIG_FIL_NAME, STATUS='OLD', ERR=999)
C//...read number of layers
  READ(ICF, *) MSTJN(1)

C//...read number of nodes in input layer:
  READ(ICF, *) MSTJN(10)

```

```

C//...read number of nodes in hidden layer(s):
    IF(MSTJN(1).GE.3) THEN
        READ(ICF,*) (MSTJN(10+I),I=1,MSTJN(1)-2)
    ENDIF

C//...read number of nodes in output layer:
    READ(ICF,*) MSTJN(10+MSTJN(1)-1)

C//...read learning rate:
    READ(ICF,*) PARJN(1)

C//...read learning decreasing rate:
    READ(ICF,*) PARJN(11)

C//...read momentum:
    READ(ICF,*) PARJN(2)

C//...read width of initial weights:
    READ(ICF,*) PARJN(4)

C//...read name of trainfile
    READ(ICF,*) TRAIN_FILE_0
    READ(ICF,*) TRAIN_FILE_1
    READ(ICF,*) TRAIN_FILE_2

    READ(ICF,*) nntuple1
    READ(ICF,*) nntuple2
    READ(ICF,*) nntuple3

C//...read name of weight file
    READ(ICF,*) WEIGHT_FIL_NAME

C//...read name of weight file
    READ(ICF,*) DUMP_FIL_NAME

```

```
C//...read name of weight file
      READ(ICF,*) ERRTRN_FIL_NAME

C//...read name of weight file
      READ(ICF,*) ERRSTST_FIL_NAME

C//...read name of weight file
      READ(ICF,*) OUTNET_FIL_NAME

C////...read no of train cycles
      READ(ICF,*) NCYCLES

C////...after how many cycles to test
      READ(ICF,*) NTSTFREQ

C//...read no of patterns per update in JNTRAL
      READ(ICF,*) MSTJN(2)

C//...read no of updates per epoch
      READ(ICF,*) MSTJN(9)

C//...read which error measure to use
      READ(ICF,*) MSTJN(4)

C//...READ WHICH GAIN FUCTION TO USE
      READ(ICF,*) MSTJN(3)

C//...read factor for multiplication of input
      READ(ICF,*) INP_MULT

C//...read factor for multiplication of modules
      READ(ICF,*) INP_MULT5m
```

```

C//...read shift of input
    READ(ICF,*) INP_SHIFT

C//...definition of cuts on output
    READ(ICF,*) OUT_CUT_TST      ! OUTPUT CUT USED FOR TESTING

    CLOSE(ICF)
    PRINT*, 'CONFIGURATION FILE SUCCESSFULLY READ.'
    OPEN(58, file='energies.dat', status='UNKNOWN')

C//.....ENDOF Configure the network.....
C//*****

C//***** START READ TRAINING AND TESTING PATTERNS *****
C//
C//....READ TRAINING AND TESTING PATTERNS OF TYPE 0 (Eletrons)
C//*****

C.. open the ntuple
    call hropen(1, 'lun1', train_file_0, ' ', 8190, istat)

        if (istat.ne.0) stop 1
        call hcdire('//lun1', ' ')
        call hldire(' ', 'T')

C.. obtain the address and parameters of ntuple id
c    call hgnpar(1000, 'hproj1')

C.. obtain the total number of events in the ntuple
    call hrin(0, 9999, 0)
    call hrecov(nntuple1, ' ')
    call hnoent(nntuple1, nevnt)
    call hrend('lun1')
    close(1)

c.. split into training and testing files
    Ntrpats_0 = nevnt/2
    Ntstpats_0 = Ntrpats_0

```



```

c.. copy into an array the event contents of the ntuple
      call hropen(1,'lun1',train_file_0,' ',8190,istat)

      if (istat.ne.0) stop 1
      call hcdir('///lun1',' ')
      call hrin(0,9999,0)
      call hrecov(nntuple1,' ')
      DO IPAT=1,NTRPATS_0
      call hbname(nntuple1,' ',.0,'$clear')
      call hbname(nntuple1,'module0',sim0(1),'$set')

      call hgnt(nntuple1,ipat,ierr)
      if (ierr.ne.0) stop 2
x(39)=x(40)
x(25)=x(26)

      do i=1,(INPUT)
      tr_pat(ipat,i)=x(i)
      enddo

C// *****
C//  NORMALIZING/MULTIPLYING AND SHIFTING THE INPUT
C// *****
TOTEN=0
DO I=1,INPUT
  TOTEN=TOTEN+TR_PAT(IPAT,I)
ENDDO

      write(58,*) toten

if(toten.eq.0) toten = 1
IF(INP_MULT.EQ.0) THEN
  INP_MULT = 1
  TOTEN = 1
ENDIF

```

```

        toten=sqrt(abs(toten))
DO I=1,INPUT
    TR_PAT(IPAT,I)=INP_MULT*(TR_PAT(IPAT,I)/TOTEN)+INP_SHIFT
ENDDO

C// *****
TR_DESO(IPAT,1)=1
TR_DESO(IPAT,2)=-1
TR_DESO(IPAT,3)=-1
    ENDDO

    call hrend('lun1')
        close(1)

100 PRINT*,IPAT-1,' TRAINING PATTERNS OF TYPE 0 READ'
    NTRPATS_0=IPAT-1
C//***** START READING TEST PATTERNS *****
C//.....READ TEST PATTERNS OF TYPE 0

    call hropen(1,'lun1',train_file_0,' ',8190,istat)

    if (istat.ne.0) stop 1
    call hcdire('//lun1',' ')
    call hrin(0,9999,0)
    call hrecov(1001,' ')
    DO IPAT=ntrpats_0+1,ntrpats_0+NTSTPATS_0
    call hbname(nntuple1,' ',.0,'$clear')
    call hbname(nntuple1,'module0',sim0(1),'$set')

        call hgnt(nntuple1,ipat,ierr)
        if (ierr.ne.0) stop 2

x(39)=x(40)
x(25)=x(26)

```

```

do i=1,(INPUT)
  tst_pat(ipat-ntrpats_0,i)=x(i)
enddo

C// *****
C//  NORMALIZING/MULTIPLYING AND SHIFTING THE INPUT
C//  *****

TOTEN=0
DO I=1,INPUT
  TOTEN=TOTEN+TST_PAT(IPAT-ntrpats_0,I)
ENDDO

write(58,*) toten
if(toten.eq.0) toten = 1
IF(INP_MULT.EQ.0) THEN
  INP_MULT = 1
  TOTEN = 1
ENDIF

  toten=sqrt(abs(toten))

DO I=1,INPUT
  TST_PAT(IPAT-ntrpats_0,I)
  1=INP_MULT*(TST_PAT(IPAT-ntrpats_0,I)/TOTEN)+INP_SHIFT
ENDDO

C// *****
Tst_DESO(IPAT-ntrpats_0,1)=1
Tst_DESO(IPAT-ntrpats_0,2)=-1
Tst_DESO(IPAT-ntrpats_0,3)=-1
ENDDO

call hrend('lun1')
close(1)

300 PRINT*,IPAT-ntrpats_0-1,' TEST PATTERNS OF TYPE 0 READ'
  NTSTPATS_0=IPAT-ntrpats_0-1

C//*****
C//.....read training patterns of type 1 (Pions)

```

```

C//*****
      nevnt=0
C.. open the ntuple
      call hropen(2,'lun2',train_file_1,' ',8190,istat)
      if (istat.ne.0) stop 1
      call hcdire('//lun2',' ')
      call hldire('//lun2','T')
C.. obtain the address and parameters of ntuple id
c      call hgnpar(1001,'hproj1')
C.. obtain the total number of events in the ntuple
      call hrin(0,9999,0)
      call hnoent(nntuple2,nevnt)
      call hrend('lun2')
      close (2)
c.. split into training and testing files
      Ntrpats_1 = nevnt/2
      Ntstpats_1 = Ntrpats_1
c.. copy into an array the event contents of the ntuple

      call hropen(2,'lun2',train_file_1,' ',8190,istat)
      if (istat.ne.0) stop 1
      call hcdire('//lun2',' ')
      call hrin(0,9999,0)
      DO IPAT=1,NTRPATS_1
      call hbname(nntuple2,' ',.0,'$clear')
      call hbname(nntuple2,'module0',sim0(1) ,'$set')

      call hgnt(nntuple2,ipat,ierr)
      if (ierr.ne.0) stop 2
x(39)=x(40)
x(25)=x(26)

      do i=1,(INPUT)
      tr_pat(ipat+ntrpats_0,i)=x(i)

```

```

        enddo

C// *****
C//  NORMALIZING/MULTIPLYING AND SHIFTING THE INPUT
C// *****
TOTEN=0
DO I=1,INPUT
    TOTEN=TOTEN+TR_PAT(IPAT+ntrpats_0,I)
ENDDO

    write(58,*) toten
if(toten.eq.0) toten = 1
IF(INP_MULT.EQ.0) THEN
    INP_MULT = 1
    TOTEN = 1
ENDIF

    toten=sqrt(abs(toten))
DO I=1,INPUT
    TR_PAT(IPAT+ntrpats_0,I)
    1=INP_MULT*(TR_PAT(IPAT+ntrpats_0,I)/TOTEN)+INP_SHIFT
ENDDO
C// *****
TR_DESO(IPAT+ntrpats_0,1)=-1
TR_DESO(IPAT+ntrpats_0,2)=1
TR_DESO(IPAT+ntrpats_0,3)=-1
    ENDDO
    call hrend('lun2')
    close(2)
200 PRINT*,IPAT-1,' TRAINING PATTERNS OF TYPE 1 READ'
    NTRPATS_1=IPAT-1

C//*****
C//.....read test patterns of type 1

```

```

C//*****
      call hropen(2,'lun2',train_file_1,' ',8190,istat)
      call hcdir('//lun2',' ')
      call hrin(0,9999,0)
      DO IPAT=1+ntrpats_1,ntrpats_1+NTSTPATS_1
      call hbname(nntuple2,' ',.0,'$clear')
      call hbname(nntuple2,'module0',sim0(1) ,'$set')

      call hgnt(nntuple2,ipat,ierr)
      if (ierr.ne.0) stop 2
x(39)=x(40)
x(25)=x(26)

      do i=1,(INPUT)
      tst_pat(ipat+ntstpats_0-ntrpats_1,i)=x(i)
      enddo

C// *****
C//  NORMALIZING/MULTIPLYING AND SHIFTING THE INPUT
C// *****
TOTEN=0
DO I=1,INPUT
  TOTEN=TOTEN+TST_PAT(IPAT+ntstpats_0-ntrpats_1,I)
ENDDO
  write(58,*) toten
if(toten.eq.0) toten = 1
IF(INP_MULT.EQ.0) THEN
  INP_MULT = 1
  TOTEN = 1
ENDIF
  toten=sqrt(abs(toten))
DO I=1,INPUT
  TST_PAT(IPAT+ntstpats_0-ntrpats_1,I)
  1=INP_MULT*(TST_PAT(IPAT+ntstpats_0-ntrpats_1,I)/TOTEN)

```

```

&+INP_SHIFT
ENDDO
C// *****
Tst_DESO(ipat+ntstpats_0-ntrpats_1,1)=-1
Tst_DESO(ipat+ntstpats_0-ntrpats_1,2)=1
Tst_DESO(ipat+ntstpats_0-ntrpats_1,3)=-1

ENDDO
call hrend('lun2')
close(2)

400 PRINT*,IPAT-NTrPATS_1-1,' TEST PATTERNS OF TYPE 1 READ'
NTSTPATS_1=IPAT-NTrPATS_1-1

C//*****
C//.....read training patterns of type 2 (Muons)
C//*****
    nevnt=0
C.. open the ntuple
    call hropen(3,'lun3',train_file_2,' ',8190,istat)
    if (istat.ne.0) stop 1
    call hcdir('//lun3',' ')
    call hldir('//lun3','T')
C.. obtain the address and parameters of ntuple id
c    call hgnpar(1001,'hproj1')
C.. obtain the total number of events in the ntuple
    call hrin(0,9999,0)
    call hnoent(ncmp3,nevnt)
    call hrend('lun3')
    close (3)
c.. split into training and testing files
    Ntrpats_2 = nevnt/2
    Ntstpats_2 = Ntrpats_2
c.. copy into an array the event contents of the ntuple

```

```

call hropen(3,'lun3',train_file_2,' ',8190,istat)
if (istat.ne.0) stop 1
call hcdir('//lun3',' ')
call hrin(0,9999,0)
DO IPAT=1,NTRPATS_2
call hbname(nntuple3,' ',.0,'$clear')
call hbname(nntuple3,'module0',s1m0(1) ,'$set')

        call hgnt(nntuple3,ipat,ierr)
        if (ierr.ne.0) stop 2
x(39)=x(40)
x(25)=x(26)

do i=1,(INPUT)
    tr_pat(ipat+ntRpats_0+ntRpats_1,i)=x(i)
enddo

C// *****
C//  NORMALIZING/MULTIPLYING AND SHIFTING THE INPUT
C// *****
TOTEN=0
DO I=1,INPUT
    TOTEN=TOTEN+TR_PAT(IPAT+ntRpats_0+ntRpats_1,I)
ENDDO
    write(58,*) toten
if(toten.eq.0) toten = 1
IF(INP_MULT.EQ.0) THEN
    INP_MULT = 1
    TOTEN = 1
ENDIF
    toten=sqrt(abs(toten))
DO I=1,INPUT
    TR_PAT(IPAT+ntRpats_0+ntRpats_1,I)

```



```

1=INP_MULT*(TR_PAT(IPAT+ntrpats_0+ntrpats_1,I)/TOTEN)+INP_SHIFT
ENDDO
C// *****
TR_DESO(IPAT+ntrpats_0+ntrpats_1,1)=-1
TR_DESO(IPAT+ntrpats_0+ntrpats_1,2)=-1
TR_DESO(IPAT+ntrpats_0+ntrpats_1,3)=1
    ENDDO
    call hrend('lun3')
    close(3)
250 PRINT*,IPAT-1,' TRAINING PATTERNS OF TYPE 1 READ'
    NTRPATS_2=IPAT-1

C//*****
C//.....read test patterns of type 2
C//*****
    call hropen(3,'lun3',train_file_2,' ',8190,istat)
    call hcdir('//lun3',' ')
    call hrin(0,9999,0)
    DO IPAT=1+ntrpats_2,ntrpats_2+NTSTPATS_2
    call hbname(nntuple3,' ',.0,'$clear')
    call hbname(nntuple3,'module0',s1m0(1) ,'$set')

        call hgnt(nntuple3,ipat,ierr)
        if (ierr.ne.0) stop 2
x(39)=x(40)
x(25)=x(26)

do i=1,(INPUT)
    tst_pat(ipat+ntstpats_0+ntstpats_1-ntrpats_2,i)=x(i)
enddo

C// *****
C// NORMALIZING/MULTIPLYING AND SHIFTING THE INPUT
C// *****

```

```

TOTEN=0
DO I=1,INPUT
  TOTEN=TOTEN+TST_PAT(IPAT+ntstpats_0
    1 +ntstpats_1-ntrpats_2,I)
ENDDO
  write(58,*) toten
if(toten.eq.0) toten = 1
IF(INP_MULT.EQ.0) THEN
  INP_MULT = 1
  TOTEN = 1
ENDIF
  toten=sqrt(abs(toten))
DO I=1,INPUT
  TST_PAT(IPAT+ntstpats_0+ntstpats_1-ntrpats_2,I)
  1=INP_MULT*(TST_PAT(IPAT+ntstpats_0+ntstpats_1-ntrpats_2,I)/TOTEN)
  &+INP_SHIFT
  ENDDO
C// *****
  Tst_DESO(ipat+ntstpats_0+ntstpats_1-ntrpats_2,1)=-1
  Tst_DESO(ipat+ntstpats_0+ntstpats_1-ntrpats_2,2)=-1
  Tst_DESO(ipat+ntstpats_0+ntstpats_1-ntrpats_2,3)=1

  ENDDO
  call hrend('lun3')
  close(3)

450 PRINT*,IPAT-NTRPATS_2-1,' TEST PATTERNS OF TYPE 1 READ'
  NTSTPATS_2=IPAT-NTRPATS_2-1
  CLOSE(58)

C//***** INITIALIZE TRAINING RUN *****

```

```

C// Opening file for net outputs
  IF(OUTNET_FIL_NAME.NE.'NONE') THEN
    OPEN(ION,FILE=OUTNET_FIL_NAME,STATUS='UNKNOWN',ERR=999)
  ENDIF

C// Opening file for screen net outputs
  IF(ERRTST_FIL_NAME.NE.'NONE') THEN
    OPEN(ISCR,FILE=ERRTST_FIL_NAME,STATUS='UNKNOWN',ERR=999)
  ENDIF

C// Opening file for net error
  IF(ERRTRN_FIL_NAME.NE.'NONE') THEN
    OPEN(IER,FILE=ERRTRN_FIL_NAME,STATUS='UNKNOWN',ERR=999)
  ENDIF

C//...initialize the net:
  CALL JNINIT

C//***** OPTION TO READ NET FROM DISK *****
  IF(WEIGHT_FIL_NAME.NE.'NONE') THEN
PRINT*, 'READING WEIGHTS...'
OPEN(IWEI,FILE=WEIGHT_FIL_NAME,STATUS='OLD',ERR=999)
CALL JNREAD(IWEI)
CLOSE(IWEI)
  ENDIF

C//.....*****
C// .....***** START TRAINING NET *****
C//.....*****
C//.....loop over training cycles
C//.....SWITCH TO WORKSTATIONS.....
  CALL DATIME(ID,IT)

```

```

ISEED=ISL(2)
x0=ntrpats_0-1
x1=ntrpats_1-1
x2=ntrpats_2-1

C//***** INITIALIZING TRAINING CYCLE *****/
DO ICYCLE=1,NCYCLES
C//.....loop over training patterns
DO IPAT=1,MSTJN(9)/3 !Begin training cycle..

C//.....SELECT RANDOMLY A PATTERN OF TYPE 0
ISELPAT=RANF(ISEED)*X0+1

C//.....loop over input nodes
DO INODE=1,nvar
OIN(INODE)=TR_PAT(iselpat,INODE)
ENDDO
do INODE=1,3
OUT(inode)=TR_DESO(iselpat,inode)
enddo
CALL JNTRAL

C//.....select randomly a pattern of type 1
ISELPAT=RANF(ISEED)*X1+NTRPATS_0+1

C//.....loop over input nodes
DO INODE=1,nvar
OIN(INODE)=TR_PAT(iselpat,INODE)
ENDDO
do INODE=1,3
OUT(inode)=TR_DESO(iselpat,inode)
enddo

```

```

CALL JNTRAL

C//.....select randomly a pattern of type 1
ISELPAT=RANF(ISEED)*X2+NTRPATS_0+ntrpats_1+1

C//.....loop over input nodes
DO INODE=1,nvar
    OIN(INODE)=TR_PAT(iselpat,INODE)
ENDDO
do INODE=1,3
    OUT(inode)=TR_DESO(iselpat,inode)
enddo
CALL JNTRAL

    ENDDO
C//.....INIT TEST PERFORMANCE
    IF(MOD(ICYCLE,NTSTFREQ).EQ.0) THEN
C//  Output training error
    WRITE(IER,*)ICYCLE,PARJN(1),PARJN(9)

C//  LOOP OVER TRAINING PATTERNS
    N1=0
        N1ok=0
    N2=0
        N2ok=0
    N3=0
        N3ok=0

C//.....LOOP OVER TEST PATTERNS
    DO IPAT=1,NTRPATS_0+ntrpats_1+ntrpats_2

C//.....LOOP OVER INPUT NODES
    DO INODE=1,nvar

```

```

        OIN(INODE)=TR_PAT(IPAT,INODE)
ENDDO
CALL JNTEST

C//....WRITE TO A FILE THE NET OUTPUT

IF(ICYCLE.EQ.NCYCLES) WRITE(ION,*)OUT(1),OUT(2),OUT(3),ipat
        IF(TR_DESO(IPAT,1).GE.(0.98)) THEN
                N1=N1+1
                IF((OUT(1).GT.out(2)).and.(out(1).gt.out(3)))
& THEN
                        N1ok=N1ok+1
                ENDIF
                endif
                IF(TR_DESO(IPAT,2).GE.0.98) THEN
                N2=N2+1
                IF((OUT(2).GT.out(1)).and.(out(2).gt.out(3)))
& THEN
                        N2ok=N2ok+1
                ENDIF
                endif
                IF(TR_DESO(IPAT,3).GE.0.98) THEN
                N3=N3+1
                IF((OUT(3).GT.out(1)).and.(out(3).gt.out(2)))
& THEN
                        N3ok=N3ok+1
                ENDIF
                endif
ENDDO

        PT1=100*FLOAT(N1ok)/FLOAT(N1)
        PT2=100*FLOAT(N2ok)/FLOAT(N2)
        PT3=100*FLOAT(N3ok)/FLOAT(N3)

C//  END LOOP OVER TRAINING PATTERNS

```

```

N1=0
        N1ok=0
    N2=0
        N2ok=0
    N3=0
        N3ok=0
C//.....LOOP OVER TEST PATTERNS
    DO IPAT=1,NTSTPATS_0+ntstpats_1+ntstpats_2

C//.....LOOP OVER INPUT NODES
    DO INODE=1,nvar
        OIN(INODE)=TST_PAT(IPAT,INODE)
    ENDDO
    CALL JNTEST

C//....WRITE TO A FILE THE NET OUTPUT
    IF(ICYCLE.EQ.NCYCLES) WRITE(ION,*)OUT(1),out(2),out(3),ipat
        IF(Tst_DESO(IPAT,1).GE.0.98) THEN
            N1=N1+1
            IF((OUT(1).GT.out(2)).and.(out(1).gt.out(3))) then
                N1ok=N1ok+1
            ENDIF
        ENDIF
    IF(Tst_DESO(IPAT,2).GE.0.98) THEN
        N2=N2+1
        IF((OUT(2).GT.out(1)).and.(out(2).gt.out(3))) then
            N2ok=N2ok+1
        ENDIF
    ENDIF
    IF(Tst_DESO(IPAT,3).GE.0.98) THEN
        N3=N3+1
        IF((OUT(3).GT.out(1)).and.(out(3).gt.out(2))) then
            N3ok=N3ok+1
        ENDIF
    ENDIF

```

```

        ENDIF
    ENDDO

        Ptt1=100*FLOAT(N1ok)/FLOAT(N1)

        Ptt2=100*FLOAT(N2ok)/FLOAT(N2)

        Ptt3=100*FLOAT(N3ok)/FLOAT(N3)

    PRINT*,ICYCLE,'TST:',ptt1,ptt2,ptt3

        PRINT*,'TRN:',pt1,pt2,pt3

        WRITE(ISCR,*)ICYCLE,ptt1,ptt2,ptt3,pt1,pt2,pt3

C//***** Saves weights of net each teste cycle *****

        ENDIF

C//***** END TEST NET PERFORMANCE *****

C// End of training cycle

        ENDDO

C//***** END OF TRAINING CYCLE *****

C// Saving trained weights on file

        IF(DUMP_FIL_NAME.NE.'NONE') THEN

            print *,dump_fil_name

            OPEN(IWEI2,FILE=DUMP_FIL_NAME,STATUS='UNKNOWN',ERR=999)

            CALL JNDUMP(IWEI2)

            ENDIF

            open(81,FILE='nevents.dat',STATUS='UNKNOWN')

            write(81,*) NTRPATS_0

            write(81,*) NTRPATS_1

            write(81,*) NTRPATS_2

            CLOSE(81)

        CLOSE(IER)

        CLOSE(ION)

        CLOSE(IWEI2)

        CLOSE(ISCR)

C//-----
999    STOP

        END

```


Apêndice C

Código Neural usado *online*

Neste apêndice traremos o código neural utilizado durante os experimentos realizados *online* durante o período de aquisição de dados do feixe de partículas. Este código está dividido em três partes principais. A primeira parte que consta deste apêndice é a parte que traz a interface utilizada para acessar a classe Neural, responsável pela criação da rede completamente conectada, seu treino e teste. A segunda parte traz as rotinas em si. Finalmente, a terceira parte é o programa que acessa via VME os dados e os utiliza no treino e teste da rede. Um detalhe importante é que a classe Neural herda da classe Thread as características de ser uma *thread*. Esta classe Thread [79] foi desenvolvida para o sistema de seleção (*Trigger*) do detector ATLAS.

Abaixo temos a interface da classe Neural :

```
//First Definitions
#define MAX_NEURONS 30
#define MAX_WEIGHTS 1000
#define SIZE_LUT 1024
#define X_LUT 4.0
#define USE_LUT 0

// Includes
#include <stdio.h>
#include <iostream.h>
#include <unistd.h>
#include "os/Thread.h"
```

```

#include "neural/NeuralTypes.h"
#include "acquire/AcquireTypes.h"
#include "neural/table.h"
#include <sched.h>
#include "devs/SemInt.h"

class Neural : public Thread
{

public:
Neural(int numtypes, int num_samp,int * neu_per_samp,
        int * pari,float * parf, int maxim); //Constructor
~Neural(); //Destructor
void train(float * in_vec, float * out_vec);
void test(float * in_vec, float * out_vec_got, float * err);
void event_input(int type, T_Event * event);
void event_inputno(int type, T_Event * event);
void event_input_tr(int type, T_Event * event, float* out);
void dump_out(void)
        {flag_dump=true;while(flag_dump) ::sleep(1);}
void dump_out1(void);
void dump_in(void);
void eff(void)
{flag_eff=true;while(flag_eff) ::sleep(1);}
void eff1(void);
void cycle();
virtual void execute(); // code to run in loop (Acquisition)
        float * weight;
void interact(int interaction)
{m_interact = interaction;}
SemIntReq* sem_req1;

private:
        int find_weight(int layer, int neuron1, int neuron2);

```

```

int find_neuron(int layer, int neuron1);
float func(float input);
float func_lin(float input);

private: // Variables
int num_samp;
int * neu_per_samp;
float * trsh;
int * pari;
float * parf;
nn_input *** type;
nn_output *** out;
volatile int * num;
FILE * fileh;
bool first;
int sum_wei;
int sum_neu;
int numtypes;
int m_maxim;
    volatile State m_state;
volatile bool flag_eff;
volatile bool flag_dump;
volatile bool flag_input;
int temptype;
T_Event* tempevent;
float* tempout;
SemIntReq* sem_req;
SemIntHan* sem_han;
SemIntHan* sem_han1;
int Counter;
bool stoptrain;
nn_output Out[3];
int m_interact;

```

```
}; //End of Class Neural
```

Vamos agora ao código da rede neural em si. A maior parte do código se relaciona com o teste e treino da rede :

```
// Includes
#include <stdlib.h>
#include <iostream.h>
#include <fstream.h>
#include <math.h>
#include "neural/Neural.h"
#include "neural/tstamp.h"

Neural::Neural(int numtypes, int num_samp_in,int * neu_per_samp_in,
               int * pari_in,float * parf_in,int maxim) :
               numtypes(numtypes),num_samp(num_samp_in),
               neu_per_samp(neu_per_samp_in),pari(pari_in),
               parf(parf_in),m_maxim(maxim) // Constructor
{

fileh = fopen("randomics.dat","w");
num = new int[numtypes];
for(int i=0;i<numtypes;i++) num[i]=0;
int sum_wei=0;
int sum_neu=0;
for(int i=1;i<num_samp;i++) {
sum_wei+=neu_per_samp[i-1]*neu_per_samp[i];
sum_neu+=neu_per_samp[i];
}
weight= new float[sum_wei];
cout << (unsigned int) weight << endl;
trsh = new float[sum_neu];
for(int i=0;i<sum_wei;i++){
int temp=rand();
```

```

    int k;
    if (temp>(RAND_MAX/2))
        k=1;
    else
        k=-1;
    weight[i]=parf[3]*k*( (float) temp/RAND_MAX);
    fprintf(fileh,"%f \n",weight[i]);
}
for(int i=0;i<sum_neu;i++) {
    int temp=rand();
    int k;
    if (temp>(RAND_MAX/2))
        k=1;
    else
        k=-1;
    trsh[i]=parf[3]*k*( (float) temp/RAND_MAX);
    fprintf(fileh,"%f \n",trsh[i]);
}
type = new (nn_input**) [numtypes];
for(int i=0;i<numtypes;i++)
    type[i] = new (nn_input*) [m_maxim];
out = new (nn_output**) [numtypes];
for(int i=0;i<numtypes;i++)
    out[i] = new (nn_output*) [m_maxim];
for(int j=0;j<numtypes;j++)
    for(int i=0;i<m_maxim;i++){
        type[j][i]=new nn_input;
        out[j][i]=new nn_output;
    }
cout << "passei 1" << endl;
for(int j=0;j<numtypes;j++)
    for(int i=0;i<m_maxim;i++){
        delete type[j][i]; type[j][i] = NULL;
        delete out[j][i]; out[j][i] = NULL;
    }

```

```

    }
    cout << "passei 2" << endl;

    printf("number of weights %d \n",sum_wei);
    printf("number of thresholds %d \n",sum_neu);
    first=true;

    Counter = 0;
    for(int i=0;i<numtypes;i++){
        if(i==0) {
            Out[i].outputchan[0]=1.0;
            Out[i].outputchan[1]=-1.0;
            Out[i].outputchan[2]=-1.0;
        }
        if(i==1) {
            Out[i].outputchan[0]=-1.0;
            Out[i].outputchan[1]=1.0;
            Out[i].outputchan[2]=-1.0;
        }
        if(i==2) {
            Out[i].outputchan[0]=-1.0;
            Out[i].outputchan[1]=-1.0;
            Out[i].outputchan[2]=1.0;
        }
    }
    for(int i=0;i<28;i++)
        place[i]--;

}

Neural::~Neural() // Destructor
{
    cout << "Counter inside train " << Counter << endl;
    cout << "Weight inside train " << weight[0] << endl;
}

```

```

dump_in();
cout << "I dumped the input" << endl;

for(int j=0;j<numtypes;j++) {
    delete type[j];
    delete out[j];
}
sleep(2);

if(weight!=NULL) delete weight;
if(trsh!=NULL) delete trsh;
delete type;
delete out;
fclose(fileh);

}

void
Neural::train(float * in_vec, float * out_vec) // Train method
{
    float out_vec_got[MAX_NEURONS];

    int sample;
    int neuron;
    int neuron1;
    int indice1,indice2,indice3;
    float input[MAX_WEIGHTS];
    float out_neu[MAX_NEURONS];
    float delta[MAX_NEURONS];
    float sum;
    for(neuron=0;neuron<neu_per_samp[0];neuron++){
        input[neuron]=in_vec[neuron];
    }
    for(sample=1;sample<num_samp;sample++){ /* For each sample */

```

```

indice2=find_neuron(sample,0);
for(neuron=0;neuron<neu_per_samp[sample];neuron++)
{ /* For each neuron in that sample */
sum=0.0;
indice1=find_weight(sample,neuron,0);
for(neuron1=0;neuron1<neu_per_samp[sample-1];neuron1++){
sum += (weight[indice1]) * (input[neuron1]);
/* Calculates the inner product between Weight and input */
indice1++;
}
sum += trsh[indice2];
out_neu[indice2]=func(sum);
delta[indice2]=(1-out_neu[indice2]*out_neu[indice2]);
indice2++;
}
indice3=find_neuron(sample,0);
for(neuron=0;
(! (sample==num_samp-1))&&neuron<neu_per_samp[sample];neuron++){
input[neuron]=out_neu[indice3];
indice3++;
} /* Save Output of a prior sample as input to the next one */
}

for(neuron=0;neuron<neu_per_samp[num_samp-1];neuron++){
out_vec_got[neuron] = out_neu[find_neuron(num_samp-1,neuron)];
};

/*Calculates delta at the last layer*/
for(neuron=0;neuron<neu_per_samp[num_samp-1];neuron++){
delta[find_neuron(num_samp-1,neuron)]*=
(out_vec[neuron]-out_neu[find_neuron(num_samp-1,neuron)]);
}

/*Calculates delta for the other layers*/
for(sample=num_samp-2;sample>0;sample--){

```



```

for(neuron=0;neuron<neu_per_samp[sample];neuron++){
    sum=0.0;
    for(neuron1=0;neuron1<neu_per_samp[sample+1];neuron1++){
        sum+=weight[find_weight(sample+1,neuron1,neuron)]
            *delta[find_neuron(sample+1,neuron1)];
    }
    delta[find_neuron(sample,neuron)]*=sum;
}
}

```

/*Updates Thresholds*/

```

for(sample=1;sample<num_samp;sample++){
    indice1=find_neuron(sample,0);
    for(neuron=0;neuron<neu_per_samp[sample];neuron++){
        trsh[indice1]+=delta[indice1]*parf[0];
        indice1++;
    }
}

```

/*Updates weights in the first layer*/

```

for(neuron=0;neuron<neu_per_samp[1];neuron++){
    indice1=find_neuron(1,neuron);
    indice2=find_weight(1,neuron,0);
    for(neuron1=0;neuron1<neu_per_samp[0];neuron1++){
        weight[indice2]+=parf[0]*delta[indice1]*in_vec[neuron1];
        indice2++;
    }
}

```

/*Updates weights in the other layers*/

```

for(sample=2;sample<num_samp;sample++){
    for(neuron=0;neuron<neu_per_samp[sample];neuron++){
        indice1=find_neuron(sample,neuron);
        indice2=find_weight(sample,neuron,0);
    }
}

```

```

    indice3=find_neuron(sample-1,0);
    for(neuron1=0;neuron1<neu_per_samp[sample-1];neuron1++){
        weight[indice2]+=parf[0]*delta[indice1]*out_neu[indice3];
        indice2++;
        indice3++;
    }
}
}
for(neuron=0;neuron<neu_per_samp[num_samp-1];neuron++){
    out_vec[neuron]=out_vec_got[neuron];
}

/*Updates learning at each epoch*/
pari[0]++;
if((pari[0]%pari[1])==0){
    parf[0]*=parf[1];
}

return;
}

int
Neural::find_weight(int layer, int neuron1, int neuron2){
    // Finds weights in the net
    int i;
    int posic=0;

    for(i=1;i<layer;i++){
        posic+=neu_per_samp[i]*neu_per_samp[i-1];
    }
    posic+=neuron1*neu_per_samp[layer-1];
    posic+=neuron2;
    return(posic);
}

```

```

int
Neural::find_neuron(int layer, int neuron1){
    // find a neuron in the net
    int i;
    int posic=0;

    for(i=1;i<layer;i++){
        posic+=neu_per_samp[i];
    }
    posic+=neuron1;
    return(posic);
}

void
Neural::test(float * in_vec, float * out_vec_got, float * err)
{
    // Tests the network
    int sample;
    int neuron;
    int neuron1;
    int indice1,indice2,indice3;
    float input[MAX_WEIGHTS];
    float out_neu[MAX_NEURONS];
    float sum;

    for(neuron=0;neuron<neu_per_samp[0];neuron++){
        input [neuron]=in_vec [neuron];
    }
    for(sample=1;sample<num_samp;sample++){ /* For each sample */
        indice2=find_neuron(sample,0);
        for(neuron=0;neuron<neu_per_samp[sample];neuron++){
            /* For each neuron in that sample */
            sum=0.0;

```

```

    indice1=find_weight(sample,neuron,0);
    for(neuron1=0;neuron1<neu_per_samp[sample-1];neuron1++){
        sum += (weight[indice1]) * (input[neuron1]);
/* Calculates the inner product between Weight and input */
        indice1++;
    }
    sum += trsh[indice2];
    out_neu[indice2] = func(sum);
    indice2++;
}
indice3=find_neuron(sample,0);
for(neuron=0;neuron<neu_per_samp[sample];neuron++){
    input[neuron]=out_neu[indice3];
    indice3++;
} /* Save Output of a prior sample as input to the next one */
}
for(neuron=0;neuron<neu_per_samp[num_samp-1];neuron++){
    *err=*err+(out_vec_got[neuron]
        -out_neu[find_neuron(num_samp-1,neuron)])
    *(out_vec_got[neuron]-out_neu[find_neuron(num_samp-1,neuron)]);
    out_vec_got[neuron] = out_neu[find_neuron(num_samp-1,neuron)];
}
return;
}
float
Neural::func(float input){
    /* Non linear functions for the backprop calc */
    float output;
    int pos;

    if(!USE_LUT){
        output=(float)tanh((double)input);
        return(output);
    }else{

```

```

    pos=(int)(SIZE_LUT*input/X_LUT)+SIZE_LUT;
    if(pos<0) pos=0;
    if(pos>=SIZE_LUTx2) pos=SIZE_LUTx2-1;
    //return(table[pos]);
    return(0);
}
}

float
Neural::func_lin(float input){
    /* Non linear derivative functions for the backprop calc */
    float output;

    output=1-input*input;
    return(output);
}

void
Neural::event_input(int type_of_input, T_Event * event){
    // Inputing events to circular buffer
    float sum=0.0;
    nn_input* tempor;
    int number = num[type_of_input]%m_maxim;

    if(num[type_of_input]<m_maxim){ // Brackets added for test
        type[type_of_input][number] = new nn_input;
        tempor = type[type_of_input][number];
        if((num[0] == 2) || (num[0] == 3)) {
            cout << num[0] << endl;
            for(int i=0; i< 28; i++)
                cout << event->signal[0].Energy[i] << " ";
            cout << endl;
        }
        for(int i=0;i<NUMBER_CHAN_IN;i++){

```

```

        int j1=place[2*i];int j2=place[2*i+1];
        tempor->inputchan[i]=(event->signal[0].Energy[j1]+
            event->signal[0].Energy[j2]);
        sum+=tempor->inputchan[i];
    }
    if(sum == 0.0) sum=1.0;
    for(int i=0;i<NUMBER_CHAN_IN;i++){
        // Normalization step
        tempor->inputchan[i]/=sum;
    }
    tempor->number_of_event=(int)event->trig.Evt;
    tempor->run_number=(int)event->trig.Run;

    num[type_of_input]++;
return ;
}

void
Neural::event_inputno(int type_of_input, T_Event * event){
// No normalization applied
float sum=0.0;
nn_input* tempor;
int number = num[type_of_input]%m_maxim;

if(num[type_of_input]<m_maxim)
    type[type_of_input][number] = new nn_input;
    tempor = type[type_of_input][number];
    for(int i=0;i<NUMBER_CHAN_IN;i++){
        int j1=2*i;int j2=2*i+1;
        tempor->inputchan[i]=(event->signal[0].Energy[j1]
            +event->signal[0].Energy[j2]);
    }
    tempor->number_of_event=(int)event->trig.Evt;
    tempor->run_number=(int)event->trig.Run;
}

```

```

        num[type_of_input]++;
return ;
}

void
Neural::event_input_tr(int type_of_input, T_Event * event, float* out){
float sum=0.0;
float err;
nn_input* temporptr;
int number = num[type_of_input]%m_maxim;

if(num[type_of_input]<m_maxim)
    type[type_of_input][number] = new nn_input;
    temporptr = type[type_of_input][number];

    for(int i=0;i<NUMBER_CHAN_IN;i++){
        int j1=place[2*i];int j2=place[2*i+1];
        temporptr->inputchan[i]=(event->signal[0].Energy[j1]
+event->signal[0].Energy[j2]);
        sum+=temporptr->inputchan[i];
    }
    if(sum == 0.0) sum=1.0;
    sum=sqrt(abs(sum));
    for(int i=0;i<NUMBER_CHAN_IN;i++){
        temporptr->inputchan[i]/=sum;
    }
    temporptr->number_of_event=(int)event->trig.Evt;
    temporptr->run_number=(int)event->trig.Run;

    num[type_of_input]++;
    this->test(temporptr->inputchan,out,&err);
return ;
}

```

```

void
Neural::dump_out1(void){
// Dump output and weights
fstream outfile;
outfile.open("outfile.dat",ios::out);

for(int k=0;k<numtypes;k++)
  for(int i=0;i<num[k];i++){
    if(out[k][i]!=NULL){
      outfile << out[k][i]->number_of_event << " ";
      outfile << out[k][i]->run_number << " ";
      for(int j=0;j<NUMBER_CHAN_OUT;j++)
        outfile << out[k][i]->outputchan[j] <<" ";
      outfile << endl;
    }
  }

outfile.close();

FILE * fp;
fp=fopen("pesosout.dat","w");
for(int i=0;i<260;i++)
  fprintf(fp,"%17.10e\n",weight[i]);
fclose(fp);
fp=fopen("biassout.dat","w");
for(int i=0;i<13;i++)
  fprintf(fp,"%17.10e\n",trsh[i]);
fclose(fp);
}

void
Neural::dump_in(void){
// Dump input data
fstream outfile;

```



```

float outnet[NUMBER_CHAN_OUT];

float err;

outfile.open("infile.dat",ios::out);

int maximum;

for(int k=0;k<numtypes;k++){
    if(num[k] < m_maxim)
        maximum = num[k];
    else
        maximum = m_maxim;
    for(int i=0;i<maximum;i++){
        if(type[k][i]!=NULL){
            outfile << type[k][i]->number_of_event << " ";
            outfile << type[k][i]->run_number << " ";
            for(int j=0;j<NUMBER_CHAN_IN;j++) {
                outfile << type[k][i]->inputchan[j] << " ";
            }
            this->test(&(type[k][i]->inputchan[0]),outnet,&err);
            for(int j=0;j<NUMBER_CHAN_OUT;j++) {
                outfile << outnet[j] << " ";
            }
            outfile << endl;
        }
    }
}

outfile.close();

}

void
Neural::cycle(void)
{ // Cycle for neural network training (all data sets)
float outnet[NUMBER_CHAN_OUT];

```

```

int* select = new int[numtypes];
Counter++;

for(int i=0;i<numtypes;i++){
    int maximum;
    if(num[i]<m_maxim)
        maximum = num[i];
    else
        maximum = m_maxim;
    select[i] = rand()%(maximum-1);
    for(int j=0;j<NUMBER_CHAN_OUT;j++)
        outnet[j]=Out[i].outputchan[j];
    this->train(&(type[i][select[i]]->inputchan[0]),outnet);
}
}

void
Neural::eff1(void){ // Calculates efficiency
int *n,*nok;
float err[1];
*err=0.0;
n = new int[numtypes];
nok = new int[numtypes];

for(int k=0;k<numtypes;k++){
    nok[k]=0;
    n[k]=0;
    for(int i=0;i<num[k];i++){
        if(out[k][i] !=NULL) {delete out[k][i]; out[k][i] = NULL;
out[k][i] = new nn_output; }
        else {out[k][i] = new nn_output; }
        out[k][i]->number_of_event = type[k][i]->number_of_event;

```

```

        out[k][i]->run_number = type[k][i]->run_number;
        n[k]++;
        test(type[k][i]->inputchan,out[k][i]->outputchan,err);
        if(out[k][i]->outputchan[0]>0) nok[k]++;
    }
}

printf(" %f - %f \n", 100*((float)nok[0]/(float)n[0]),
        100*((float)nok[1]/(float)n[1]));
}

void
Neural::execute(void){
    // Code to run in other thread to train network
    flag_eff=false;
    flag_dump=false;
    flag_input=false;
    m_state=Running;
    stoptrain = false;
    int counter=0;
    // Definitions to use time_stamp
    tstamp ts1;
    tstamp ts2;
    float delta;
    int train_steps = 25000;

    cout << "Starting Neural Network training" << endl;
// Time measurement functions
    ts_open(1,TS_DUMMY);
    ts_clock(&ts1);

    while(Thread::state() == Thread::Running) {
        if(num[numtypes-1] < 50) {
            cout << "Gonna Sleep " ;

```

```

        ::sleep(1);
    }
    else {
        if(counter==train_steps) {
            ts_clock(&ts2);
            delta = ts_duration(ts1,ts2);
            ts_close(TS_DUMMY);
            cout << delta << endl;
            cout << " End of train" << endl ;
            cout << " latest train event : ";
            cout << num[1] << endl;
        }

        if(counter< train_steps+1){
            counter+=m_interact;
            for(int i=0;i<m_interact;i++){
                cycle();
            }
        }

        if(counter > train_steps+1){
            ::sleep(1);
            counter+=m_interact;
            for(int i=0;i<m_interact;i++){
                cycle();
            }
        }
    }
}

cout << "Counter = " << dec << counter << endl;
cout << "Weight inside train " << weight[0] << endl;
return;
}

```

Finalmente, temos o código que usando as funções desta biblioteca e da biblioteca de acesso VME consegue acessar os dados e treinar a rede neural. Algumas funções de interpretação de dados são definidas antes do bloco principal (*main*) do código em si.

```
/*
Neural Network dealing program
developped by Denis Oliveira Damazio
*/
#define NUMBER_OF_EVENTS 200
#define MAX_BURST 4
#define NUMBER_PIONS 200
#define NUMBER_MIN 50
//#define ONLINE
#include "neural/Neural.h"
#include "devs/RODIPC.h"
#include "devs/VMEih.h"
#include <unistd.h>
#include <iostream.h>
#include <fstream.h>
#include "app/tile_event.h"
#ifdef ONLINE
#include <rc/rc_is_infotype.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <confdb/dfdal.h>
#endif
extern int first1;
extern T_TileCalibChannel calib[MAX_CALIB_FRAG*MAX_RAW_CHAN];

// To produce time measuements
#include "tstamp.h"
```

```

extern void fill_digi( void * rod, unsigned int rodsizes,
                     int rodnum, int verbosity, T_Event * event);
extern void InitParam(int verbosity, T_Event * event);

int main( int argc, char **argv )
{
    static T_Event ** events = NULL;
    int * nevents;
    int i ;
    char* filename;
    unsigned long vme_address;
    unsigned long vme_address_resp;
    char vme_address_char[10];
    int maximum;
    int first2=1;
    first1=1;
    nevents = new int;
    int waitvar;
    int interaction=100;

    // Arguments understanding
    cout << "interpreting cmd line arg ..." << endl << endl;
    if(argc>=2) {
        maximum=catoi(argv[1]);
    } else {
        maximum=NUMBER_PIONS;
    }
    if(argc>=3) {
        interaction = catoi(argv[2]);
    }

    // Neural Network Inicialization
    int num_samp = 3;

```

```

int *num_per_samp = new int[num_samp];
num_per_samp[0] = 14;
num_per_samp[1] = 8;
num_per_samp[2] = 1;
int pari[2];
float parf[4];
pari[0] = 0; pari[1] = 600;
parf[0] = 0.2 ; parf[1] = 0.997;
parf[2] = 1.0 ; parf[3] = 0.3;
cout << "neural network initialization ..." << endl << endl;
Neural* neural =
new Neural(2,num_samp,num_per_samp,pari,parf,maximum);
neural->interact(interaction);

// Some general variables
int len;
unsigned long buffer[400];

// Interrupts mechanism will be used untill the end
const int VME_vector = 140;
const int VME_level = 5;
VMEih* ih = new VMEih(VME_level, VME_vector);

// Access to system status
#ifdef ONLINE
IPCPartition p("tilecal_testbeam2001");
ISInfoDictionary id(p);
string server_name("RunCtrl.RODCtrl");
ISRCStateInfo c;
#endif

// Muons Acquisition Phase

// Waiting the system to be in running state

```

```

#ifdef ONLINE
    id.findValue(server_name.c_str(),c);
    while(!c.inSameState(ISRCStateInfo::ACT_Running)){
        id.findValue(server_name.c_str(),c);
        sleep(1);
    }
#endif

// Some time to let the rodst create the new RODIPC
sleep(2);

// Waiting to start up
ih->wait();

// Reading the RODIPC address from the file
ifstream* input = new ifstream("MonAddrs.dat");
(*input) >> vme_address_char;
vme_address = atoi(vme_address_char);
(*input) >> vme_address_char;
vme_address_resp = atoi(vme_address_char);
// Crating the RODIPC object to fetch events
RODIPCmaster* ipc = new RODIPCmasterDMA(vme_address);
RODIPCmaster* ipc_resp = new RODIPCmasterSC(vme_address_resp);
ipc_resp->reset();
cout << "Value used as address ";
cout << hex << vme_address << " and " << endl;
cout << vme_address_resp << dec << endl;
T_Event * event1 = new T_Event;
InitParam(0,event1);
int count = 0;
float out[3];
unsigned int outint[3];
tstamp ts_mu_1,ts_mu_2;
float delta;

```



```

ts_open(1,TS_DUMMY);
ts_clock(&ts_mu_1);

for(;;){
    if(first2 != 1 ) {
        ih->wait();
    } else
        first2=0;
    while(!ipc->read(len,(char*)buffer))
        cout << "pooling 1" << endl;
    //cout << len << " : " << hex << ((unsigned int*)buffer)[0];
    //cout << hex << " " << ((unsigned int*)buffer)[4];
    //cout << dec << endl;
    if((len!=4)&&((*(unsigned int*)buffer!=0x111))){
        fill_digi(buffer,len,0,0,event1);
        event1->trig.Run=0;
        event1->trig.Evt=count++;
        neural->event_input(0,event1);
        out[0]=0.5;
        outint[0] = 1500000;
        while(!ipc_resp->write(4,(char*)&outint[0]))
            cout << "pooling 2" << endl;
    } else
        break;
}
ts_clock(&ts_mu_2);
delta = ts_duration(ts_mu_1,ts_mu_2);
ts_close(TS_DUMMY);

// finish the communications
delete ipc;
delete ipc_resp;
delete event1;

```

```

cout << "Finished muons acquisition" << endl;
cout << (count-1) << " Events in during ";
cout << delta << " seconds" << endl;
cout << "That means " << (count-1)/(delta);
cout << " events per second" << endl;

// Pions Acquisition Phase

// Waiting the system to be in running state
#ifdef ONLINE
    id.findValue(server_name.c_str(),c);
    while(!c.inSameState(ISRCStateInfo::ACT_Running)){
        id.findValue(server_name.c_str(),c);
        sleep(1);
    }
#endif

// Some time to let the rodst create the new RODIPC
sleep(2);

// Waiting to start up
ih->wait();

// Reading the RODIPC address from the file
input = new ifstream("MonAdrs.dat");
(*input) >> vme_address_char;
vme_address = atoi(vme_address_char);
(*input) >> vme_address_char;
vme_address_resp = atoi(vme_address_char);
// Crating the RODIPC object to fetch events
ipc = new RODIPCmasterDMA(vme_address);
ipc_resp = new RODIPCmasterSC(vme_address_resp);
ipc_resp->reset();
cout << "Values used as address";
cout << hex << vme_address << dec << endl;

```

```

cout << hex << vme_address_resp << dec << endl;
event1 = new T_Event;
InitParam(0,event1);
first2=1;
count = 0;
tstamp ts_pi_1,ts_pi_2;
#ifdef TB_OPTIMIZE
    ofstream outout("outnet.dat");
#endif

ts_open(1,2);

neural->start();
sleep(1);
ts_clock(&ts_pi_1);
for(;;){
    if(first2 != 1 ) {
        ih->wait();
    } else{
        first2=0;
    }
    while(!ipc->read(len,(char*)buffer))
        cout << "pooling 1" << endl;
    //cout << len << " : " << hex << ((unsigned int*)buffer)[0];
    //cout << hex << " " << ((unsigned int*)buffer)[4];
    //cout << dec << endl;
    if((len!=4)&&((*((unsigned int*)buffer)!=0x111))){
        fill_digi(buffer,len,0,0,event1);
        event1->trig.Run=1;
        event1->trig.Evt=count++;
        neural->event_input_tr(1,event1,out);
#ifdef TB_OPTIMIZE
        outout << out[0] << endl;
#endif
    }
}
#endif

```

```

        outint[0] = (unsigned int)((out[0]+1.0)*1000000.0);
        while(!ipc_resp->write(4,(char*)&outint[0]))
            cout << "pooling 2" << endl;
    } else{
        break;
    }
    //usleep(1);
}
ts_clock(&ts_pi_2);
delta = ts_duration(ts_pi_1,ts_pi_2);
ts_close(2);
neural->cancel();
cout << "Finished acquiring pions data" << endl;
cout << (count-1) << " Events in during ";
cout << delta << " seconds" << endl;
cout << "That means " << (count-1)/(delta);
cout << " per second" << endl;

// finish the communications
delete event1;
delete ipc;
delete ipc_resp;

#ifdef TB_OPTIMIZE
    outout.close();
#endif

delete neural; // Turn off the neural network system
delete ih;
return 0;
}

```