TÉCNICAS DE REGULARIZAÇÃO DE MODELOS NEURAIS APLICADAS À PREVISÃO DE CARGA A CURTO PRAZO

Vitor Hugo Ferreira

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 MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

_____
Prof. Alexandre Pinto Alves da Silva, Ph.D.

_____
Prof. Djalma Mosqueira Falcão, Ph.D.

_____
Prof. Luiz Pereira Calôba, Dr.Ing.

_____
Prof. Marcelo Cunha Medeiros, D.Sc.

_____
Luciano de Souza Moulin, D.Sc.

RIO DE JANEIRO – BRASIL

FEVEREIRO DE 2005

FERREIRA, VITOR HUGO

Técnicas de Regularização de Modelos Neurais Aplicadas à Previsão de Carga a Curto Prazo

[Rio de Janeiro], 2005.

X, 204 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2005)

Tese – Universidade Federal do Rio de Janeiro, COPPE.

1. Previsão de Carga a Curto Prazo
2. Máquina de Vetor Suporte
3. Redes Neurais Artificiais
4. Técnicas de Regularização
5. Treinamento Bayesiano
6. Escalonamento do Ganho da Função de Ativação

I. COPPE/UFRJ II. Título (série)

DEDICATÓRIA

Este trabalho é dedicado às pessoas mais importantes da minha vida, que acreditaram e depositaram extrema confiança no meu trabalho. Pessoas como o melhor pai do mundo, também conhecido como Seu Hugo; a melhor mãe do planeta, que também atende pelo nome de Dona Tina; meu avô materno, Seu Alcides, que lá em cima deve estar tomando umas e outras para comemorar mais uma fase ultrapassada na vida do seu neto; minha avó paterna, Dona Filhinha, que conseguiu ver o neto Engenheiro, porém não está presente para celebrar mais esta vitória; e minha avó materna Mariana, que acredita muito nesse neto aqui e é o principal ponto de convergência e união da melhor família que um ser humano pode ter. Apesar de não ter conhecido em vida, dedico este trabalho ao meu avô paterno, Hugo, sem o qual eu não teria a dádiva divina de ter o exemplo de homem e de pai que tenho ao meu lado.

Dedico também a todas as pessoas que me apoiaram ao longo deste caminho. Por ordem cronológica, não posso esquecer do grande Wilson Leão, uma das grandes referências da minha vida, que ajudou a forjar o homem que sou hoje. Joseana Rocha do Monte, uma pessoa que por muito tempo dividiu este caminho, deixando ensinamentos e experiências que certamente levarei para o resto da vida. Aos meus amigos, que formam a família que Deus permitiu que eu escolhesse, que sempre me apoiaram nos momentos de necessidade. Ao meu orientador, Alexandre, que sempre orientou, estimulou e apoiou minha vida acadêmica, desde a época da graduação, em Itajubá.

Não dedico a Deus esta tese por que sei que este trabalho é ínfimo diante da infinidade da sua bondade. Porém, dedico a Ele todo meu esforço, trabalho, suor e dedicação na busca por um mundo mais unido, solidário e justo, onde o amor, o respeito ao próximo e ao meio ambiente formem os pilares de uma nova civilização.

AGRADECIMENTOS

Primeiramente a Deus, por manter sempre meu caminho iluminado, concedendo sabedoria, confiança, saúde e paz a mim e a todos que estão a minha volta.

Aos meus pais, por terem me dado a vida e me ensinado a vivê-la. Por me aturarem por 24 anos e mesmo assim ainda me amarem. Por serem o porto seguro ao qual recorro nos raros momentos turbulentos. Por rirem comigo nos abundantes momentos de alegria da minha maravilhosa vida. Enfim, por constituírem a base do que sou hoje.

A toda a minha família, pela confiança depositada e pelo carinho enorme que a mantêm unida.

A todos os meus amigos, que sempre apoiaram nos momentos difíceis, configurando realmente a família que Deus permitiu que escolhêssemos. Colegas de porta de boteco existem vários, mas são raros aqueles que surgem em hospitais na hora do aperto. Ou que ligam no exato momento em que descobrem uma notícia triste. Agradeço todas as noites pela família e pelos amigos que tenho!

À família LASPOT, nova porém unida, pela calorosa acolhida e pelo apoio incondicional durante esses dois anos. E vocês vão ter que me aturar por mais quatro anos!

Por último, mas com importância semelhante aos anteriores, agradeço ao meu orientador, Alexandre, pelo suporte dado desde os tempos de graduação, estimulando e apoiando minha evolução dentro da área acadêmica.

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

TÉCNICAS DE REGULARIZAÇÃO DE MODELOS NEURAIS APLICADAS À PREVISÃO DE CARGA A CURTO PRAZO

Vitor Hugo Ferreira

Fevereiro / 2005

Orientador: Alexandre P. Alves da Silva

Programa: Engenharia Elétrica

O conhecimento do comportamento futuro da carga é de suma importância na tomada de decisões referentes à operação e planejamento de sistemas de potência. Ao longo dos últimos anos, vários modelos vêm sendo propostos para a modelagem da carga a curto prazo, dentre os quais vêm apresentando destaque as redes neurais *feedforward*. Uma das desvantagens dos modelos neurais reside na possibilidade de aproximação excessiva dos dados de treinamento, o chamado *overfitting*, comprometendo, assim, a capacidade de generalização dos modelos estimados. Este problema pode ser abordado através do uso de técnicas de regularização. Esta tese investiga a aplicação de técnicas promissoras de controle de complexidade de modelos neurais para previsão de carga a curto prazo.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requiremenets for the degree of Master of Science (M.Sc.)

REGULARIZATION TECHNIQUES OF NEURAL MODELS APPLIED TO SHORT-TERM LOAD FORECASTING

Vitor Hugo Ferreira

February / 2005

Advisors: Alexandre P. Alves da Silva

Department: Electrical Engineering

The knowledge of loads' future behavior is very important for decision making in power system operation and planning. During the last years, many short term load forecasting models have been proposed, and feedforward neural networks have presented the best results. One of the disadvantages of the neural models is the possibility of excessive adjustment of the training data, named *overfitting*, degrading the generalization performance of the estimated models. This problem can be tackled by using regularization techniques. The present Thesis investigates the application of promising procedures for complexity control of short term load forecasting neural models.

Índice

Índice de Figuras

Índice de Tabelas

1 Introdução

A previsão de carga elétrica apresenta importância vital para a operação e o planejamento confiável, seguro e econômico dos sistemas de potência. Em função disso, esta área de estudo vem adquirindo maior interesse por parte da comunidade científica ao longo dos anos, principalmente após o advento da competição nos mercados de energia. Neste novo cenário, os agentes integrantes destes mercados devem operar em regime de máxima eficiência, com a minimização dos custos operacionais e a correta avaliação do aporte de recursos financeiros para expansão dos sistemas contribuindo sobremaneira para o alcance desta condição.

Diante da importância da previsão de carga a curto prazo, várias metodologias vêm sendo propostas ao longo dos últimos anos para abordagem deste problema, com destaque para os modelos neurais *feedforward*, que vêm apresentando sucesso em diversas aplicações de sistemas de previsão de carga em todo mundo. Entretanto, a grande maioria dos modelos neurais encontrados na literatura não aborda de maneira sistemática o problema de seleção e regularização das estruturas desenvolvidas, até mesmo negligenciando o problema do chamado *overfitting*, dando origem a modelos com reduzida capacidade de generalização. Buscando a contraposição ao excessivo empirismo na seleção e regularização das estruturas, esta tese focará na implementação de algumas técnicas de regularização de modelos neurais para previsão de carga a curto prazo, objetivando o estudo de metodologias que conduzam a modelos com capacidade de generalização satisfatória. As metodologias selecionadas são: treinamento *bayesiano* (seção 2.2.2.3), escalonamento do ganho da função de ativação (seção 2.2.3.3) e máquina de vetor suporte (seção 2.1).

Neste capítulo, serão apresentadas as formas nas quais a previsão de carga pode ser classificada, sendo destacados os diversos horizontes de previsão desejados e as

áreas de aplicação de cada tipo de previsão, com ênfase especial à previsão de carga a curto prazo. Visto que são encontradas na literatura diversas metodologias aplicadas ao horizonte de curto prazo, será apresentada uma extensa revisão bibliográfica de propostas de modelos de previsão de carga a curto prazo, sendo destacados os modelos neurais. Após a descrição do problema de previsão de carga a curto prazo, os principais objetivos desta tese, além da motivação para o desenvolvimento e as contribuições resultantes deste trabalho, serão apresentados.

As técnicas de regularização de modelos neurais *feedforward* são tratadas no capítulo 2. Neste capítulo, inicialmente são apresentadas as máquinas de vetor suporte, que podem ser consideradas como modelos *feedforward* que utilizam uma forma implícita de regularização na sua formulação. Posteriormente, são descritas algumas das metodologias mais utilizadas para controle de complexidade de MLP's, dando destaque àquelas utilizadas nesta tese.

No capítulo 3, são descritos os casos estudados, sendo mencionadas as bases de dados utilizadas, os processamentos efetuados em cada uma delas e as definições das estruturas utilizadas para cada problema. Os resultados obtidos para cada estudo, assim como as conclusões associadas, são motivo de discussão dos capítulos 4 e 5, respectivamente.

1.1 Classificação da previsão de carga

De uma maneira geral, segundo os diferentes horizontes de previsão, a previsão de carga pode ser classificada em três tipos, a saber: previsão de carga a longo prazo, previsão de carga a médio prazo e previsão de carga a curto prazo. Entretanto, conforme será discutido neste item, a definição de tais horizontes não é única. Previsões consideradas de médio prazo para algumas empresas de energia podem ser consideradas de longo prazo para outras, por exemplo.

São classificadas como previsões a longo prazo aquelas realizadas para horizontes variando de alguns meses a até trinta anos à frente, em base mensal ou anual [1]-[19]. Conforme assinalado anteriormente, a definição do intervalo de previsão a longo prazo não é única, podendo ser confundida em muitos casos com aquela relacionada com o horizonte de médio prazo. Por exemplo, DJUKANOVIC *et. al.* [3] definem previsão de carga a longo prazo como sendo aquelas realizadas para horizontes de alguns meses a até um ano à frente, enquanto que para HUANG [6], previsões para este horizonte são consideradas de médio prazo, com as de longo prazo sendo as realizadas para horizontes de cinco a dez anos.

A previsão de carga a longo prazo é utilizada em diversas funções relacionadas com o planejamento técnico e financeiro das empresas de energia, tais como planejamento da expansão dos sistemas de transmissão e distribuição, e do parque gerador, em função do possível aumento da demanda [4], [6], [7], [9]-[13], [15], [19]-[22], programação anual da manutenção de unidades geradoras [1], [4], [9], [12], [19], [22], gerenciamento energético de longo prazo [23], desenvolvimento de estratégias operacionais [23], estudos de viabilidade econômica [2], [10], planejamento dos investimentos e do orçamento das empresas de energia [10], e pesquisa de mercado [23]. Em mercados regulamentados, tais previsões podem ser utilizadas também para o desenvolvimento de políticas tarifárias [2].

Em função dos distantes horizontes de previsão, as previsões para o horizonte de longo prazo estão sempre associadas a elevados níveis de incerteza, visto que as variáveis explicativas relacionadas com o comportamento de longo prazo da carga, em sua grande maioria de origem econômica, são de difícil previsão. Dentre as variáveis econômicas comumente utilizadas, podem ser citadas: produto nacional bruto (PNB), produto interno bruto (PIB), população, número de residências, número de aparelhos de

ar condicionado, índice de CO_2 na atmosfera, índice de produção industrial e preço da energia [23], [24]. São encontrados na literatura diversos modelos para previsão de carga para este horizonte, baseados em três metodologias gerais: análise de séries temporais, regressão múltipla e sistemas inteligentes, abrangendo redes neurais artificiais, sistemas especialistas, lógica *fuzzy* e modelos híbridos, dentre outros. Dentre os trabalhos encontrados na literatura, os modelos baseados em sistemas inteligentes vem apresentando os melhores resultados, conforme verificado em [10], [17], [18], [23], [24], [25] e [26].

Previsões de carga a médio prazo são aquelas realizadas para horizontes variando de uma semana a até cinco anos, em base diária [5] e [28], semanal [3], [8] e [27] ou mensal [1], [4], [6], [11], [12], [13], [15] e [19]. De modo semelhante à previsão de carga a longo prazo, a definição do horizonte de médio prazo varia dentro da literatura. Por exemplo, DOVEH *et. al.* [28] consideram a previsão do pico de carga diário e do consumo de energia total diária, para horizontes de até um ano à frente, como previsões de médio prazo, enquanto que PARK *et. al.* [1], KIM *et. al.* [4], RAHMAN *et. al.* [29] e outros consideram previsões para estes horizontes como sendo de curto prazo.

A previsão de carga a médio prazo fornece subsídios para diversas atividades relacionadas ao planejamento da expansão e ao planejamento da operação de sistemas de potência, podendo ser citados programação da compra de combustíveis [11], [20], [27], planejamento da manutenção de equipamentos [6], [13], [15], [20], [27], planejamento do intercâmbio entre áreas [20], otimização da programação das unidades geradoras [4], [13], [15], [30], planejamento das transações (compra e venda) de energia entre empresas [6], desenvolvimento de estratégias de gerenciamento energético [28] e

planejamento do orçamento [27]. Tais previsões também podem ser utilizadas para desenvolvimento de políticas tarifárias em mercados privatizados [11], [15].

Abundam na literatura modelos para modelagem da carga a médio prazo, baseados em diversas abordagens, tais como regressão múltipla [31], [35], modelos estocásticos [27], [30], [32], [33], e sistemas inteligentes [11], [28], [34]. De forma análoga ao horizonte de longo prazo, as metodologias baseadas em sistemas inteligentes vêm apresentando os melhores resultados na solução do problema de previsão de carga a médio prazo.

Passando para o horizonte de curto prazo, são consideradas aquelas realizadas para intervalos variando de alguns minutos a até um mês à frente, em base de minutos [13], [14], [39], [41], [49], [52], [53], [59] e [61], horas [1], [2], [4] - [9], [11] - [14], [19], [29], [36], [39], [41], [43] - [49], [51] - [55], [56] - [62], ou dias [1], [3], [9], [12] - [15], [37], [39], [40], [49], [50], [53], [54], [59] e [61]. De forma análoga aos horizontes descritos nos itens anteriores, a definição do horizonte de curto prazo não é única, conforme evidenciado na literatura. Em [39] e [41], dentre outras referências, previsões para intervalos variando de alguns minutos a até uma hora à frente são consideradas de curto prazo, enquanto que em [3], [7], [15] e [63] previsões para este horizonte são consideradas de curtíssimo prazo.

A previsão de carga para o horizonte de curto prazo apresenta importância vital para a operação diária e o controle em tempo real de sistemas de potência, sendo utilizada em diversas atividades relacionadas com cada uma destas áreas. Dentre as funções inerentes ao planejamento da operação diária de sistemas de potência, a previsão de carga a curto prazo fornece subsídios para: análise de segurança [1], [3]-[5], [8], [13], [15], [22], [37], [39], [41], [44], [47]-[50], [52], [56]-[58], [60]-[62], [64], [66], [70], [74], [75], [78], [86], [90], [92]-[94], incluindo análise de contingências [3],

[8], [9], [19], [58], [62]; elaboração de estratégias de gerenciamento da carga [3], [6], [8], [20], [29], [37], [46], [47], [49], [53]-[54], [58], [62], [68], [85], [92]-[94]; programação da geração [4], [44], [48], [52], [57], [58], [72], [73], [75], [78], [79], [90], [94], abrangendo coordenação hidrotérmica [3], [8], [29], [37], [38], [43], [50], [51], [53], [54], [55], [56], [58], [62], [67], [68], [70], [92], [93], programação da compra e alocação de combustível [3], [5]-[7], [15], [21], [40], [44], [45], [48], [50], [53], [57], [58], [60], [62], [69], [72], [75], [77], [86], [90], comissionamento de unidades térmicas [3], [5], [7]-[9], [11]-[13], [15], [20], [21], [29], [36], [37], [40], [42], [46], [51], [53], [54], [55], [56], [57], [60], [62], [67]-[70], [74], [76], [77], [80], [85], [86], [89], [93], [94], e despacho econômico [3], [4], [6], [9], [11], [13], [15], [29], [36], [37], [40], [42], [45]-[46], [53], [54], [57], [58], [60], [64], [66], [68], [69], [74], [76], [77], [79], [86], [89], [92], [94]; estudos de fluxo de potência [1], [3], [9], [19], [38], [58], [62], [84], [94], incluindo fluxo de potência ótimo [3] e programação do intercâmbio entre áreas [3], [9], [11], [20], [36], [38], [39], [42], [47], [50]-[52], [55], [56], [62], [64], [68], [70], [73], [76], [85], [93]; programação da alocação de reserva girante [3], [39], [53], [61], [64], [73], [94]; programação e avaliação das transações de compra e venda de energia [5], [7], [15], [29], [43], [45], [48], [53], [62], [74], [79], [86], [90], [92]; e programação da manutenção [2], [7], [15], [21], [40], [44], [50], [53], [56], [58], [60], [62], [64], [69]. [72], [74], [75], [77], [80], [89], [92], [93]. Dentre as atividades relacionadas ao controle em tempo real de sistemas de potência, previsões realizadas para o horizonte de curto prazo fornecem informações importantes para controle automático da geração [13], [19], [22], [53], [58], [94] e controle do fluxo de potência reativa [94]. Para o caso específico de empresas de distribuição de energia, a previsão de carga a curto prazo também apresenta importância significativa, visto que o conhecimento da carga, particularmente do pico de carga, nas diversas barras do sistema é um dos requisitos

mais importantes para a operação eficiente. Estas informações constituem a base para a estimação do estado do sistema e para cálculos técnicos e econômicos, possibilitando a melhoria na operação e na manutenção dos equipamentos elétricos e no planejamento da operação dos sistemas de distribuição [3], [47], [62], [73], [81], [82], [84], fornecendo subsídios para possíveis instalações de equipamentos de emergência, desligamento de circuitos, transferências de carga, aumento da refrigeração de equipamentos críticos e ajuste dos *tap's* dos transformadores nas subestações [83].

Além da importância sob o ponto de vista técnico, tais previsões também apresentam importância significativa sob o prisma econômico. Com o advento da competição oriunda da privatização, e conseqüente desregulamentação dos mercados de energia de diversos países, os agentes de tais mercados foram obrigados a trabalhar em níveis elevados de eficiência. Neste cenário, a minimização dos custos operacionais contribui sobremaneira para a maximização da eficiência econômica dos agentes [5], [9], [47], [48], [61], [62], [79], [86], [90], [95], [96]. Em virtude disto, já que a previsão de carga a curto prazo está diretamente associada a diversas atividades relacionadas com a operação de sistemas de potência, a precisão de tais previsões está intimamente ligada à redução dos custos operacionais das empresas de energia. Em [53] é apresentado um estudo sobre o efeito da precisão da previsão de carga a curto prazo nos custos operacionais das indústrias de energia, sendo mostrado que melhorias da ordem de 1 % na precisão de tais previsões podem resultar em reduções nos custos operacionais da ordem de centenas de milhares de dólares por ano, para empresas que apresentem gastos com combustível da ordem de centenas de milhões de dólares anuais. RANAWEERA *et. al.* [75] apresentam um estudo do impacto mensal e anual nos custos operacionais de uma empresa de energia em função dos erros na previsão de carga a curto prazo, mostrando que erros acima de 5 % acarretam em aumentos consideráveis nos custos

operacionais. Em [96] é apresentada uma metodologia para o cálculo do risco associado ao planejamento da operação de curto prazo de sistemas de potência, em função das incertezas inerentes ao processo de previsão de carga a curto prazo. Os autores apresentam os resultados em [$/MWh], ou seja, o aumento dos custos operacionais em função da precisão da previsão de carga a curto prazo. VALENZUELA *et. al.* [97] apresentam um estudo sobre a influência da precisão das previsões de temperatura e de carga na estimação dos custos de geração, mostrando que quanto maior a precisão associada a tais previsões, maior será a fidedignidade das estimativas de tais custos.

Ainda dentro da ótica econômica, a previsão de carga a curto prazo fornece informações essenciais tanto para a formação do preço da energia em mercados desregulamentados, cujo conhecimento do seu comportamento futuro constitui informação imprescindível para avaliação do seu preço atual, quanto em mercados regulamentados, embasando o desenvolvimento de políticas tarifárias [14], [36], [53], [71], [87], [88], [90], [91]. Portanto, a previsão de carga para horizontes de curto prazo apresenta relevância significativa tanto para as empresas fornecedoras de energia quanto para os grandes consumidores industriais, já que estes últimos podem programar seu consumo em função do preço da energia em mercados competitivos, ou em função da tarifa estabelecida em mercados regulamentados [9], [46], [48], [65], [71].

Sob o ponto de vista exclusivo dos grandes consumidores industriais e comerciais, a previsão de carga a curto prazo também pode fornecer informações valiosas para a redução de seus custos operacionais. De uma maneira geral, o contrato de fornecimento de energia é elaborado através da especificação de um limite máximo de energia a ser consumida, dentro de um intervalo de tempo especificado, sendo previstas severas multas para situações em que este limite é ultrapassado. Neste contexto, a previsão de carga a curto prazo pode ser utilizada em programas de

gerenciamento da carga, com o intuito de evitar a ultrapassagem dos limites de energia contratados, resultando em redução dos custos operacionais dos consumidores [80].

A previsão de carga a curto prazo pode ser aplicada também em sistemas inteligentes de automação residencial [98]. Nestes sistemas, a rede de alimentação AC pode ser utilizada tanto para fornecimento de energia quanto para troca de informações entre os diversos equipamentos eletro-eletrônicos. Dados digitais podem ser trocados entre os diversos aparelhos através da rede de alimentação, com o intuito de garantir maior segurança, conforto e entretenimento aos moradores. Neste contexto, a previsão de carga a curto prazo constitui uma informação importante, contribuindo para o aumento da confiabilidade do sistema de comunicação e para a programação ótima do consumo de energia. Visto que a atenuação do sinal de comunicação em uma linha AC é proporcional à carga conectada à rede, estas previsões podem ser utilizadas por um sistema de gerenciamento da carga dedicado à manutenção do nível de carga em um nível pré-fixado pela rede de comunicação. Para a programação ótima do consumo de energia, a previsão de carga a curto prazo pode ser utilizada no controle da ocorrência de picos de carga, com as cargas extras sendo atendidas por baterias existentes na residência em questão [98]. Esta aplicação da previsão de carga a curto prazo é bastante diferente das demais, visto que a grande maioria das aplicações encontradas na literatura trata da previsão da carga de um determinado sistema ou barra, ao contrário deste caso específico, que trata da previsão de uma carga residencial.

De uma maneira geral, o comportamento de curto prazo da carga é influenciado por uma série de fatores, que podem ser separados em climáticos e não-climáticos. Dentre os fatores climáticos, podem ser citados temperatura [3]-[5], [7], [12], [19]-[21], [27], [38]-[41], [43]-[49], [51], [52], [54], [56], [60], [64], [66], [67], [69], [71]-[74], [76], [78], [79], [81], [83], [87], [89], [90], [93], [94], [97], [99]-[127], ponto de orvalho

[20], umidade relativa do ar [5], [20], [45], [73], [76], [81], [104], [105], [110], [113], [117], [126], velocidade do vento [12], [20], [73], [88], [89], [94], [99], [110], [117], direção do vento [20], nebulosidade [20], [73], [81], [88], índice pluviométrico [21], [73], [81], [89], [115], índice de evaporação [89], índices de calor [12], [73], índice de radiação solar [126] e combinações destas variáveis em índices meteorológicos [20], [68]. DOUGLAS *et. al.* [79] apresentam um estudo sobre os impactos da incerteza das previsões de temperatura na previsão de carga a curto prazo, mostrando que este impacto está diretamente relacionado com as estações do ano. Em [109] e [125] são apresentados modelos para previsão da curva diária de temperatura, em base horária, para utilização desta informação como entrada de um modelo de previsão de carga a curto prazo. Um dos softwares de previsão de carga mais utilizados na América do Norte, chamado *Artificial Neural Network Short Term Load Forecaster* (ANNSTLF), apresentado em [5], [48] e [90], possui um módulo de previsão das curvas diárias de temperatura e umidade relativa do ar, para utilização como entradas do modelo de previsão a curto prazo.

Com relação aos fatores não-climáticos diretamente ligados à dinâmica de curto prazo da carga, apresentam destaque: mês em estudo [7], [12], [76], [89], [111], [134]; dia da semana [1], [3], [4], [5], [7], [8], [12], [19], [20], [21], [38], [40], [41], [43], [44], [45], [47], [48], [49], [51], [60], [64], [66], [69], [70], [74], [76], [88], [89], [93], [95], [102], [105], [107], [111], [112], [113], [117], [120], [126], [129], [130], [131], [133], [134]; ocorrência de eventos especiais [20], [38], [48], [64], [87], [88], como feriados [4], [7], [21], [44], [49], [69], [70], [105], [107], [131], [132], greves [110] e eventos televisivos [110]; hora do dia [1], [5], [8], [9], [12], [20], [38], [41], [45], [47], [48], [49], [52], [66], [69], [70], [78], [88], [100], [101], [107], [111], [112], [114], [117],

[120], [133]; preço da energia [90], [128], [133], em mercados privatizados; e políticas tarifárias [3], [133], em mercados regulamentados.

A diversidade de modelos propostos na literatura para solução do problema de previsão de carga para o horizonte de curto prazo é diretamente proporcional à importância do mesmo. Dentre as abordagens encontradas podem ser citados os modelos baseados em regressão múltipla, análise de séries temporais, modelos de *Box-Jenkins*, alisamento exponencial, modelos de espaço de estados, sistemas caóticos, sistemas inteligentes, onde estão inseridos modelos neurais, sistemas especialistas, lógica *fuzzy* e programação evolucionária, e modelos híbridos. Visto que esta tese aborda este horizonte de previsão, e a literatura é abundante com relação a cada estrutura citada, a revisão bibliográfica de cada uma delas será apresentada em um item independente ao longo deste capítulo.

1.2 Previsão de carga a curto prazo

Conforme detalhado no item 1.1, abundam na literatura metodologias para solução do problema de previsão de carga a curto prazo. Em virtude disso, a revisão da bibliografia encontrada sobre cada uma das principais abordagens existentes (regressão múltipla, análise de séries temporais, modelos de espaço de estados, sistemas inteligentes e modelos híbridos) será apresentada em um tópico independente, começando pelos modelos baseados em regressão múltipla. Vale lembrar também que, apesar de serem apresentados alguns valores percentuais de erro, os mesmos não devem ser comparados diretamente, visto que cada série de carga apresenta dinâmica própria.

1.2.1 Modelos baseados em regressão múltipla

PAPALEXOPOULOS e HESTERBERG [38] apresentam um modelo baseado em regressão múltipla, abordando feriados, eventos especiais, dia da semana e informações de temperatura de maneira direta, ou seja, como entradas dos modelos,

para previsão de carga a curto prazo. Os parâmetros dos modelos desenvolvidos são estimados através de mínimos quadrados ponderados, com tais modelos apresentando resultados superiores aos obtidos pelo sistema implantado na época (1990) pela *Pacific Gas and Electric Company* (PGE), uma empresa de gás e eletricidade da Califórnia, EUA.

Em [47], os autores utilizam uma modelagem não-paramétrica da carga a curto prazo, através da implementação de regressão múltipla não-paramétrica. Como variáveis explicativas são utilizadas informações de temperatura e hora da previsão. Os autores apresentam uma metodologia para estimação dos parâmetros que definem os *kernels* utilizados na regressão, neste caso *gaussianos*. São estimados modelos distintos para dias úteis e finais de semana. Os resíduos gerados pelos modelos não-paramétricos são representados através de um modelo AR, contribuindo para o aumento da precisão das estruturas propostas. Para comparação com estes modelos, os autores utilizaram dois MLP's, um para dias da semana e outro para fins de semana. Apesar do MLP ter apresentado melhores resultados, o método proposto tem como principal vantagem a sua simplicidade, tanto no que diz respeito à sua implementação quanto à estrutura do modelo, requerendo a estimação de poucos parâmetros em comparação à quantidade de parâmetros livres a serem estimados pelos modelos neurais.

Em [99] é apresentado um estudo comparativo entre cinco modelos utilizados para previsão da curva de carga diária, a saber: regressão linear múltipla, séries temporais, alisamento exponencial, modelo de espaço de estado e sistema especialista. Os autores apresentam as diversas vantagens e desvantagens de cada um dos modelos, concluindo que a escolha do "melhor" modelo é extremamente dependente do problema em estudo, neste caso, a série de carga em questão.

HAIDA e MUTO [104] desenvolvem um modelo para previsão do pico de carga diário utilizando regressão linear múltipla. Como variáveis explicativas são utilizadas informações de temperatura e umidade relativa do ar. As variáveis de temperatura são transformadas através de polinômios de quarta ordem, com os parâmetros do modelo sendo estimados através de mínimos quadrados.

Em [110] é apresentado um modelo baseado em regressão linear, que divide a série de carga em quatro componentes: uma componente base, insensível a variações climáticas; uma componente relacionada com as variações climáticas; uma componente relacionada à ocorrência de eventos especiais, como greves, eventos televisivos, etc.; e uma componente aleatória, responsável pela parcela inexplicável da carga. Com o intuito de refinamento das previsões, os erros de previsão para os últimos cinco dias são incorporados ao modelo. Também é descrita a estratégia implementada para atualização automática do modelo proposto, à medida que novos dados são disponibilizados.

RAMANATHAN *et. al.* [111] propõem uma estrutura baseada em regressão linear múltipla para previsão de carga a curto prazo. São desenvolvidos 48 modelos, ou seja, um modelo para cada hora e modelos diferentes para dias de semana e finais de semana. Cada um dos modelos utiliza diversos tipos de variáveis, a saber: determinísticas, como dia da semana, mês e ano; variáveis de temperatura, incluindo transformações não-lineares da temperatura para o instante de previsão; variáveis relacionadas com a carga, como o valor da carga às 8 da manhã; e variáveis relacionadas com o resíduo do modelo, contendo os erros de previsão para os últimos cinco dias para a respectiva hora. São realizados testes de significância dos parâmetros, com o intuito de redução do número de entradas, através da exclusão das menos significativas. Este modelo venceu a competição promovida pelo *Puget Sound Power and Light Company*, empresa de energia de Seattle, EUA, apresentando melhores

resultados que alguns modelos neurais propostos. Entretanto, é importante ressaltar que a eficiência deste modelo está intimamente relacionada com a natureza das variáveis explicativas selecionadas, visto que algumas associações não-lineares entre as variáveis de entrada citadas acima não são intuitivamente verificadas. Portanto, a grande vantagem do modelo proposto reside na forma com que o problema foi representado, isto é, no conjunto selecionado de variáveis de entrada, e não na modelagem baseada em regressão múltipla por si só.

A principal vantagem dos modelos baseados em regressão múltipla reside na simplicidade dos mesmos, no que concerne à estimação dos parâmetros que definem estas estruturas. Entretanto, tais metodologias são extremamente dependentes da escolha das variáveis regressoras a serem utilizadas. Conforme evidenciado em [111], a aplicação de modelos de regressão múltipla a problemas representados de maneira adequada conduz a desempenho de previsão satisfatório. Porém, a escolha de variáveis explicativas constitui um dos principais desafios de problemas de regressão linear em geral, pois estes apresentam elevada dependência com o problema abordado.

1.2.2 Modelos baseados em séries temporais

MBAMALU e EL-HAWARY [2] apresentam um modelo SARIMA para previsão da carga horária. Antes da aplicação do modelo SARIMA, a série de carga é processada com o intuito de retirada da sazonalidade diária, através da aplicação da vigésima quarta diferença, ou seja, $S(k) = C(k) - C(k-24)$. Os parâmetros deste modelo são estimados através da aplicação de mínimos quadrados ponderados iterativamente, *iteratively reweighted least squares* (IRLS).

Em [56] os autores utilizam um modelo ARIMA para previsão de carga a curto prazo. Nesse modelo, além de séries históricas de carga e temperatura, é utilizada uma outra variável representando a estimativa da carga uma hora à frente por parte dos

operadores do sistema. Além de exigir intervenção constante por parte do usuário, a metodologia proposta requer uma série histórica de cargas previstas pelos operadores para fins de estimativa dos parâmetros do modelo, aumentando, com isso, as dificuldades no desenvolvimento do modelo proposto. São apresentados modelos para previsão da carga horária e do pico de carga diário. Vale ressaltar que o modelo de previsão da carga horária, segundo os autores, realiza previsões 24 passos à frente de maneira estável, característica inesperada de modelos ARIMA, que costumam apresentar instabilidade na presença de constantes realimentações nas suas entradas.

FAN e McDONALD [68] apresentam um modelo baseado na decomposição da série de carga em quatro parcelas. A primeira, chamada de determinística, está relacionada com a operação do sistema, representando alterações anormais da carga devido a fatores inerentes à operação, como desligamento de linhas de transmissão, saída de operação de plantas industriais, dentre outras. Os valores desta parcela são fornecidos ao modelo diretamente pelos operadores do sistema. A segunda parcela responde pelas variações sazonais e sociais, independentes das condições climáticas, sendo modelada através de séries de Fourier. A terceira componente representa a dinâmica da carga dependente das condições climáticas, sendo modelada através de uma relação não-linear entre a carga e o índice de temperatura equivalente, obtido através da combinação das informações de temperatura, umidade relativa do ar e velocidade do vento. A quarta e última parcela, denominada carga residual, apresenta relação com o comportamento estocástico da série de carga, sendo modelada através de uma estrutura ARMA. Os modelos relacionados com cada uma das componentes são estimados através de mínimos quadrados recursivos ponderados, *weighted recursive least squares* (WRLS), realizando previsão da carga horária.

Em [91], os autores desenvolvem um modelo ARMA com ruído aditivo apresentando função densidade de probabilidade hiperbólica para previsão de carga a curto prazo. Inicialmente, a série de carga é processada, com o intuito de retirada da média amostral e das sazonalidades verificadas através da análise do periodograma da referida série. O modelo ARMA é aplicado à série de carga tratada, com a ordem do modelo e os respectivos parâmetros do mesmo sendo estimados através da minimização da versão não-tendenciosa do critério de *Akaike, Akaike's information corrected criterion* (AICC). Entretanto, na análise dos resíduos do modelo obtido, é verificado que a distribuição dos mesmos pode ser ajustada através de uma distribuição hiperbólica, ressaltando a necessidade do desenvolvimento de modelos ARMA com ruído aditivo hiperbólico. O modelo proposto apresentou erros de previsão da ordem de 1,7 %, para dois meses de previsões horárias, quando aplicado à série de carga do *Californian System Operator* (CAISO), operador do sistema elétrico da Califórnia, EUA.

HUANG e SHIH [92] propõem um modelo que utiliza estruturas ARMA tanto para casos de ruído *gaussiano* quanto não-*gaussiano*. Para tal, é realizada uma série de testes estatísticos para verificação da hipótese de ruído *gaussiano* na série de carga processada, ou seja, onde já foram realizados os tratamentos referentes à retirada de média amostral, sazonalidades e tendência. Se esta série apresenta ruído gaussiano, técnicas tradicionais baseadas nas funções de autocorrelação e autocorrelação parcial são utilizadas para determinação da ordem do modelo ARMA a ser utilizado. Do contrário, se a hipótese de ruído *gaussiano* for rejeitada, técnicas baseadas em estatísticas de ordem superior são utilizadas para determinação da ordem do modelo ARMA associado.

Em [135] os autores apresentam um estudo comparativo entre modelos ARIMA e MLP's para previsão de carga a curto prazo, utilizando funções de autocorrelação

linear e não-linear para seleção de variáveis de entrada, com esta última sendo diretamente relacionada com a teoria de informação mútua. Através do estudo destas funções, os autores concluem que as autocorrelações existentes na série de carga em questão são predominantemente lineares, favorecendo obviamente a utilização de modelos lineares. De fato, os resultados obtidos pelo modelo ARIMA e pelo MLP não diferem consideravelmente. A grande contribuição do artigo reside na sugestão, por parte dos autores, do estudo estatístico detalhado do problema antes da abordagem do mesmo, ou seja, antes do projetista escolher um modelo extremamente não-linear para solução do problema, é de vital importância confirmar a sua natureza não-linear. A análise das funções de autocorrelação linear e não-linear é uma das diversas formas de verificação da existência de possíveis não-linearidades no problema.

Assim como as estruturas baseadas em regressão múltipla, o desempenho de modelos do tipo ARMA para análise de séries temporais está diretamente relacionado com a escolha adequada das variáveis de entrada, ou seja, com a seleção dos atrasos das componentes auto-regressivas e média-móveis a serem utilizadas. Conforme apresentado em [135], o estudo das funções de autocorrelação linear pode ser utilizado como ferramenta para identificação da estrutura (ordem) do modelo. Porém, a utilização de funções de autocorrelação linear para escolha de variáveis de entrada de problemas não-lineares, e vice-versa, pode conduzir à escolha equivocada de atrasos das séries disponíveis, comprometendo o desempenho dos modelos estimados.

1.2.3 Modelos de espaço de estados

ZHENG *et. al.* [13] desenvolvem um modelo baseado na combinação de análise de multi-resolução e filtro de *Kalman* para previsão da carga horária. São desenvolvidos dois modelos, um utilizando apenas informações da série de carga, e outro incorporando

informações de temperatura, com este último apresentando melhor desempenho em termos do erro de previsão.

TRUDNOWSKI *et. al.* [63] propõem um modelo baseado em filtro de *Kalman* tanto para previsão de carga a curto prazo, ou seja, em intervalos de uma hora, quanto para previsão a curtíssimo prazo, em intervalos de cinco minutos. Os modelos são testados com dados da *Bonneville Power Administration* (BPA), uma empresa de transmissão de energia do Noroeste dos EUA, apresentando resultados satisfatórios.

Em [136] os autores apresentam um modelo baseado em filtro de *Kalman* para previsão de carga a curto prazo, utilizando um ponderador responsável por enfatizar os erros para os dados mais recentes. São desenvolvidos modelos para previsão uma semana à frente e uma hora à frente, com ambos apresentando resultados satisfatórios para a série em questão.

Apesar da aplicabilidade dos modelos de espaço de estados em problemas de previsão de carga a curto prazo permitir o mapeamento de múltiplas entradas em múltiplas saídas (MIMO), correlacionando previsões para mais de um passo à frente, esta abordagem não têm sido muito explorada na literatura.

1.2.4 Sistemas especialistas

Neste item, será apresentada uma breve revisão bibliográfica das metodologias baseadas em sistemas especialistas encontradas na literatura para solução do problema de previsão de carga a curto prazo. Revisões bibliográficas a respeito da aplicação de sistemas inteligentes em previsão de carga, incluindo sistemas especialistas, modelos neurais e lógica *fuzzy*, podem ser encontradas em [14], [61] e [137].

Em [36], os autores desenvolvem um sistema especialista para previsão da carga horária, da carga para as próximas seis horas e da curva de carga diária. São apresentados os procedimentos de identificação de variáveis explicativas, de formação

da base de dados e elaboração do conjunto de regras que define o sistema especialista. O modelo proposto é testado com dados da *Virginia Power Company*, uma empresa de energia da Virgínia, EUA, apresentando desempenho melhor que os modelos baseados em regressão múltipla utilizados na época (1988).

HO *et. al.* [37] apresentam um sistema especialista para previsão de carga a curto prazo da *Taiwan Power Company*. Ao longo do artigo, é apresentado o desenvolvimento do conjunto de regras, que incorporam conhecimento dos operadores do sistema e análises estatísticas, e a implementação, em PROLOG, do sistema proposto. O desempenho deste modelo foi superior ao apresentado pela metodologia de *Box-Jenkins*.

RAHMAN e BABA [65] descrevem o software de previsão de carga a curto prazo desenvolvido pelos mesmos, citando hardware utilizado, linguagem de programação aplicada no trabalho, dentre outras especificações. O algoritmo de previsão é baseado em um sistema especialista, sendo apresentados diversos detalhes do desenvolvimento deste, como filtragem dos dados, tratamento de incertezas, e obtenção e modificação automática do conjunto de regras do sistema especialista. O modelo proposto realiza previsões um passo à frente, visto que o mesmo é atualizado a cada hora, à medida que novos dados de carga e das condições climáticas são disponibilizados.

A utilização de sistemas especialistas para previsão de carga a curto prazo apresenta como principal vantagem a inserção, no cerne do modelo, do conhecimento de operadores sobre o problema. Entretanto, a obtenção destas informações não é trivial, visto que para tal é necessária intervenção, ao longo do desenvolvimento do modelo, de operadores experientes e familiarizados com o sistema elétrico em estudo, objetivando a construção de um conjunto consistente de regras.

1.2.5 Redes Neurais Artificiais

Este item apresentará uma descrição dos artigos encontrados na literatura que utilizam modelos neurais, incluindo *perceptron* de múltiplas camadas (MLP), redes de função de base radial (RBFN's), mapas de *Kohonen*, dentre outros, para previsão de carga a curto prazo. Em virtude do elevado número de propostas de modelos neurais verificadas na literatura, a discussão sobre estes trabalhos será dividida em dois itens. O primeiro tratará das contribuições relacionadas com a forma de representação do problema, com o segundo abordando aquelas associadas ao tipo de modelo utilizado.

1.2.5.1 Representação

Neste item serão apresentadas as propostas de modelos neurais para previsão de carga a curto prazo cuja principal contribuição reside na forma de representação do problema. Em outras palavras, serão discutidos os trabalhos que inovaram na seleção das variáveis de entrada e na forma como elas foram utilizadas. As possíveis representações foram divididas em dois grupos, a saber: modelos univariados e modelos multivariados. Os modelos univariados utilizam apenas informações da série de carga, representando a relação entre a carga e os fatores climáticos e não-climáticos que a influenciam de forma implícita. Os modelos multivariados utilizam explicitamente informações tanto da série de carga quanto de séries contendo informações climáticas, sendo a principal contribuição destes modelos a forma na qual estas variáveis são utilizadas.

1.2.5.1.1 Modelos univariados

ASAR e McDONALD [42] apresentam um estudo sobre os efeitos no desempenho de modelos neurais devido à utilização de diferentes procedimentos de normalização e de escolha das entradas a serem utilizadas. Os autores afirmam que a

normalização dos dados através dos valores máximo e mínimo do conjunto de treinamento apresentou os piores resultados. Esta conclusão não era esperada, visto que esta forma de normalização é uma das mais encontradas na literatura. Também é citado no artigo o problema do *overfitting*, sem a abordagem desta questão através da aplicação de alguma técnica de controle da complexidade de modelos neurais, sendo apenas salientado que tal problema pode ser evitado através da utilização de uma base de dados maior. Entretanto, esta proposta não soluciona o problema do *overfitting*, visto que, para o caso específico de previsão de carga, dados muito antigos podem não mais representar fidedignamente o período onde serão realizadas as previsões, comprometendo a capacidade de generalização dos modelos estimados.

MARÍN et. al. [57] propõem um modelo neural constituído de três estágios. O primeiro consiste em um mapa de *Kohonen*, desenvolvido para agrupamento das curvas de carga diárias em função das condições climáticas e de fatores econômicos e sociais. Realizada a classificação dos dias, para cada grupo é desenvolvido um modelo neural baseado em redes de *Elman* recorrentes. No terceiro e último estágio, são realizadas previsões de carga horárias e da curva de carga diária diretamente, em base horária, com os dois modelos produzindo previsões de maneira independente. O sistema proposto é testado com dados de uma empresa da região central da Espanha, apresentando erros de previsão da ordem de 1,9 %, para previsões da curva de carga diária ao longo de dois anos.

Em [58], os autores utilizam mapas de *Kohonen* em conjunto com *wavelets* para previsão da curva de carga diária, em base de 3 horas. Inicialmente, os dias são agrupados em quatro grupos, correspondendo aos dias úteis, sábados, domingos e segundas-feiras, sendo desconsiderados os feriados. Posteriormente, para cada grupo, é aplicado um modelo baseado em *wavelets* para realização das previsões.

YAO *et. al.* [85] propõem uma metodologia que combina análise de multi-resolução, baseada em *wavelets*, e redes de função de base radial (RBFN's), para previsão da curva de carga diária, em base de 15 minutos. A série de carga é decomposta, através da análise de multi-resolução, em uma aproximação e três detalhes. A curva de carga prevista é obtida através da soma da aproximação, prevista pela RBFN, e dos detalhes, previstos através da análise de multi-resolução.

LAMEDICA *et. al.* [131] desenvolvem um modelo neural para previsão de carga a curto prazo, dando especial atenção para os dias especiais, como feriados e períodos de férias. Inicialmente, é desenvolvido um MLP treinado através do algoritmo de retropropagação do erro tradicional para previsão da curva de carga diária para dias normais, sendo estimados doze modelos, um para cada mês. Posteriormente, os autores propõem um modelo para previsão da curva de carga apenas para dias anormais, ou especiais, como finais de semana prolongados, períodos de férias e feriados. Este modelo utiliza dois módulos, um não-supervisionado e outro supervisionado. O módulo não-supervisionado é implementado através de um mapa auto-organizável de *Kohonen* para clusterização da base de dados. Como entrada deste sistema, é utilizada a curva de carga diária, gerando como saída um código de identificação do cluster associado àquele padrão. Este código identificador do grupo é utilizado como entrada do módulo supervisionado, que utiliza o mesmo modelo utilizado para previsão de dias normais, com as entradas binárias responsáveis pela codificação do dia da semana sendo substituídas pelas entradas contendo os códigos de identificação dos clusters associados às curvas de carga de um dia antes, dois dias antes e do dia a ser previsto, respectivamente. O código de identificação do dia a ser previsto é fornecido diretamente pelos operadores do previsor, tomando por base a data do dia a ser previsto e comparando esta com os resultados obtidos pelo algoritmo de clusterização. O modelo

proposto apresentou desempenho satisfatório tanto para os dias ditos “normais” quanto para os especiais. Entretanto, uma das principais desvantagens da metodologia proposta reside na necessidade de identificação, por parte dos operadores do sistema, da ocorrência de dias especiais, aumentando, assim, a necessidade de intervenção do usuário.

Em [139], é apresentado um modelo neural, baseado no MLP, cuja estrutura é basicamente a mesma do modelo desenvolvido em [7], porém com uma modificação interessante no que concerne ao algoritmo de treinamento utilizado. Neste novo algoritmo, os erros para o conjunto de treinamento são ponderados pelos custos marginais de operação para cada hora do dia, com o intuito de minimizar os erros do MLP para as horas do dia de maior custo para a empresa. Apesar de ser uma heurística de treinamento interessante, a extrema dificuldade na estimação dos custos horários de produção constitui uma das principais desvantagens do algoritmo proposto, visto que tais custos devem incorporar custos de aquisição de combustível, custos de pessoal e custos de manutenção, grandezas de difícil estimativa em base horária.

Em [141], os autores desenvolvem um modelo combinando análise de multi-resolução através de representação de células de autocorrelação (*autocorrelation shell representation*) e MLP’s. O modelo proposto apresenta três estágios. No primeiro, a série temporal é decomposta em diferentes escalas através da decomposição em células de autocorrelação. Posteriormente, cada escala obtida é prevista através de um MLP. No terceiro e último estágio, é utilizado um MLP para previsão final da carga, utilizando como entradas as previsões realizadas pelos diferentes MLP’s. Para o desenvolvimento dos MLP’s do segundo estágio, é utilizado o método de determinação automática de relevância (ARD) para determinação do número de entradas a serem utilizadas, no caso o número de atrasos das respectivas séries decompostas, sendo utilizado para

treinamento dos modelos o regularizador conhecido como decaimento de pesos sinápticos, *weight decay*. Esta técnica será apresentada em detalhes no capítulo 2.

Os modelos univariados buscam informação direta única e exclusivamente da série a ser prevista, não correlacionando este conjunto de dados com outra série temporal. Uma das principais vantagens desta abordagem reside no menor requisito de dados, visto que esta necessita apenas da série de carga para previsão da mesma, relacionando os diversos fatores que influenciam a dinâmica de curto prazo da carga de maneira implícita, através do agrupamento de padrões semelhantes de carga, por exemplo. Entretanto, em virtude da supressão de informações meteorológicas diretas, os modelos univariados não são suficiente robustos para antever a mudança no comportamento de curto prazo da carga devido à ocorrência de mudanças climáticas bruscas, como entradas de frentes frias e quentes, por exemplo.

1.2.5.1.2 Modelos multivariados

KHOTANZAD *et. al.* [5] apresentam as duas gerações do software de previsão de carga a curto prazo ANNSTLF, *artificial neural network short-term load forecaster*. A primeira geração, inicialmente proposta em [48], consiste de três grupos de MLP's distintos, cada um contendo uma única camada escondida e 24 saídas, representando os 24 valores de carga previstos para o próximo dia. O primeiro grupo é responsável pela previsão do comportamento semanal da carga, apresentando 72 entradas, relacionadas com valores atrasados das séries de carga e temperatura, sendo estimado um modelo para cada dia da semana. O segundo grupo está relacionado com o comportamento diário da carga, possuindo 72 entradas também relacionadas com valores atrasados das séries de carga e temperatura, porém com tais atrasos diferindo daqueles utilizados pelo modelo responsável pela dinâmica semanal da carga, sendo também estimado um modelo para cada dia da semana. O terceiro e último grupo é responsável pela

modelagem da dinâmica horária da carga, apresentando nove entradas relacionadas com valores atrasados das séries de carga, temperatura e umidade relativa do ar, além de uma variável indicadora do dia da semana, e uma única saída, sendo estimados 24 modelos, um para cada hora do dia. Todos os modelos são treinados diariamente utilizando o algoritmo clássico de retropropagação do erro. As saídas de cada um dos modelos são ponderadas, para obtenção da curva de carga diária prevista para o dia em questão, com tais ponderações sendo estimadas também diariamente através de mínimos quadrados ponderados. O modelo proposto foi testado e implementado em vinte companhias de energia norte-americanas, apresentando desempenho satisfatório em todas as empresas citadas no artigo.

Na segunda geração, são desenvolvidos 24 MLP's, um para cada hora do dia, com as estruturas destes modelos neurais, no que tange a número e tipos de variáveis de entrada e número de neurônios na camada escondida, sendo especificadas para cada um dos quatro períodos nos quais o dia é dividido pelos autores. Com o intuito de incrementar o desempenho do modelo para os feriados, os mesmos são tratados como sábados ou domingos, dependendo da empresa onde o sistema esteja implementado.

Já que o modelo proposto necessita de previsões horárias tanto de temperatura quanto de umidade relativa do ar, e os serviços de meteorologia disponibilizam apenas previsões de valores máximo e mínimo diários destas grandezas, os autores desenvolvem modelos neurais, baseados em MLP's, para previsão da curva diária de temperatura e umidade relativa do ar.

A segunda geração do ANNSTLF, na época da publicação de [5], estava implantada em diversas empresas de energia dos EUA e Canadá, apresentando desempenho satisfatório, em termos de precisão, para todas as empresas.

SATISH *et. al.* [19] desenvolvem um modelo combinando diversos MLP's para previsão da curva de carga diária. O primeiro MLP, chamado de MLP básico, consiste em um modelo neural contendo 54 entradas, contendo informações de temperatura, do dia da semana e as 24 cargas do dia anterior, uma única camada escondida e 24 saídas, contendo a curva de carga prevista. O segundo módulo apresenta dois MLP's, um para o pico de carga e outro para o valor mínimo de carga para o dia em estudo, ambos apresentando três entradas contendo informações de carga e temperatura, uma camada escondida e uma saída, contendo o pico de carga ou mínimo de carga diário, dependendo do modelo. Além destes módulos neurais, existem outros dois módulos utilizados pelo modelo. O primeiro, chamado de *averager*, calcula a curva de carga média, utilizando as curvas de carga das últimas dez semanas para o mesmo dia da semana onde será realizada a previsão. Por exemplo, se serão realizadas previsões para quarta-feira, este módulo calcula a curva média tomando por base as últimas dez quartas-feiras existentes na base de dados. O segundo, chamado de previsor, combina as previsões realizadas pelos módulos de previsão do pico de carga, do valor mínimo de carga e da curva média, obtendo uma previsão da curva de carga diária. O último módulo combina as previsões realizadas pelo MLP básico e pelo módulo previsor de maneira ponderada, com as ponderações sendo definidas para cada período do dia de maneira heurística, tomando por base o conhecimento dos autores acerca do problema no que diz respeito ao desempenho de cada módulo para cada período do dia. Os autores comparam os resultados obtidos pelos modelos que utilizam entradas de temperatura com aqueles que negligenciam esta informação, com os primeiros apresentando os melhores resultados. Vale ressaltar que estes modelos utilizam informações de temperatura efetivamente medidas, visto que necessitam apenas de valores atrasados da série de temperatura, e não de valores previstos desta grandeza. A previsão de

temperatura constitui um dos principais empecilhos para utilização desta grandeza em implementações reais de modelos de previsão de carga, em virtude da complexidade requerida no desenvolvimento de sistemas precisos de previsão de temperatura. Entretanto, na presença de previsões precisas desta informação climática, disponibilizadas por serviços de meteorologia confiáveis, a temperatura prevista é uma variável de suma importância para o processo de previsão de carga, sendo vital na antecipação de mudanças no comportamento de curto prazo da carga devido à ocorrência de mudanças climáticas bruscas, como entrada de frentes frias e quentes, por exemplo.

PENG *et. al.* [40] desenvolvem um MLP para previsão da carga total consumida em um determinado dia. Além do MLP totalmente conectado, é testada uma outra arquitetura neural, onde uma das entradas do neurônio de saída é uma combinação linear das entradas originais. O treinamento, que utiliza o algoritmo de retropropagação de erro tradicional, é realizado através da seleção de padrões similares dentro da base de dados. Inicialmente, a base de dados é subdividida em 5 grupos: segunda-feira, terça-feira a quinta-feira, sexta-feira, sábado e domingo. Dentro dos respectivos grupos são selecionados para treinamento os sete dias mais similares ao que será previsto, com o grau de similaridade entre os dias sendo obtido através de uma medida de distância entre as entradas associadas a cada dia. O modelo que utiliza uma combinação linear das entradas como entrada adicional do neurônio de saída apresentou melhores resultados.

Em [43], os autores apresentam o desenvolvimento de um MLP para previsão da curva de carga diária da *Pacific Gas and Electric Company* (PGE), descrevendo os procedimentos utilizados para seleção das variáveis de entrada e para escolha da estrutura do modelo e do conjunto de treinamento a ser utilizado. O modelo apresentado

foi implementado na PGE, apresentando erros da ordem de 1,8 % e 2,0 %, para previsão do pico de carga diário e da curva de carga diária, respectivamente, para as previsões realizadas para todo o ano de 1991.

Em [48] os autores apresentam a terceira geração do software de previsão de carga a curto prazo ANNSTLF. A primeira geração foi apresentada em [45] e a segunda em [5]. O modelo proposto no artigo apresenta três módulos, dois baseados em MLP's e um que realiza a combinação adaptativa das saídas dos dois outros modelos. Ambos MLP's apresentam a mesma estrutura, com a diferença entre os MLP's residindo nas suas saídas. Enquanto um módulo é treinado para previsão do comportamento regular da carga (módulo previsor da carga base), ou seja, previsão da curva de carga diária, o outro módulo é treinado para previsão da diferença entre a carga do dia a ser previsto e do dia anterior a este (módulo previsor da mudança no comportamento da carga). A previsão obtida pelo último módulo é somada à curva de carga do dia anterior para finalização da previsão da curva de carga diária. Finalmente, a saída do modelo é obtida através da combinação das saídas dos módulos anteriores, realizada pelo módulo de combinação adaptativa, que pondera as saídas de cada MLP para obtenção da previsão da curva de carga diária. Estas ponderações são obtidas para cada hora, utilizando o algoritmo de mínimos quadrados recursivos.

Outra melhoria implementada na terceira geração do ANNSTLF diz respeito ao tratamento dado a feriados e dias especiais. Nas duas gerações anteriores, estes dias eram tratados como sábados ou domingos, dependendo da empresa em estudo. Nesta nova versão, estes dias são tratados de maneira específica, através da aplicação do chamado algoritmo de *Reza*. A idéia básica deste algoritmo consiste na busca, na base de dados, de feriados com padrão de carga semelhante ao do feriado a ser previsto, tomando como indicativo de semelhança o valor da temperatura no momento da

ocorrência do pico de carga. Posteriormente, o pico de carga do feriado a ser previsto é obtido através da interpolação ponderada dos picos de carga dos feriados semelhantes. Obtida esta nova previsão para o pico de carga, a saída gerada pelo modelo neural é modificada, tomando por base este novo valor para o pico de carga.

Na época da publicação de [48], o software ANNSTLF era o sistema de previsão de carga a curto prazo baseado em redes neurais mais utilizado na América do Norte, operando em 35 empresas de energia elétrica do Canadá e EUA. O Operador Nacional do Sistema Elétrico Brasileiro (ONS) também já se utiliza deste software para realização de estudos de previsão de carga a curto prazo.

Em [67], é desenvolvido um modelo que combina filtros digitais e modelo neural ADALINE. Inicialmente, é feita análise espectral da série de carga, dividindo a série original em três componentes: componente base, componente de baixa freqüência e componente de alta freqüência. A componente base representa a sazonalidade semanal, a componente de baixa freqüência representa a sazonalidade diária e a componente de alta freqüência representa a sazonalidade horária. Cada componente é obtida através da filtragem da série original por três diferentes filtros, sendo realizadas previsões sobre cada uma das séries. Também são realizadas previsões sobre a série residual, ou seja, a série de erros de previsão. Nesta série é identificada a componente da carga dependente da temperatura, através de análise espectral, sendo esta extraída através de um filtro digital e posteriormente também sendo prevista. Cada uma das séries obtidas é prevista utilizando o modelo ADALINE, que nada mais é que um neurônio convencional cuja função de ativação é do tipo linear. Os pesos sinápticos deste modelo neural são estimados através do algoritmo de mínimos quadrados.

MOHAMMED *et. al.* [69] relatam o desenvolvimento de um modelo neural implementado na *Florida Power and Light Company* (FPL), empresa de energia do

estado da Flórida, EUA. Primeiramente, a base de dados foi dividida em 4 conjuntos: verão, inverno, transição I (março) e transição II (outubro). Além destes grupos, foram obtidos outros três, chamados dias especiais, que são: frente fria, frente quente e feriados. Com exceção dos dias especiais, cada conjunto obtido foi subdividido em outros três: segunda-feira, dias da semana menos segunda-feira e fim de semana. Por fim, cada grupo diário foi subdividido em 5 períodos, cujos horários começam à uma hora, 6 horas, 11 horas, 16 horas e 21 horas, respectivamente. Para cada subconjunto é estimado um modelo neural. O artigo apresenta uma técnica de adaptação das redes estimadas, que é feita de maneira diária, semanal e mensal. Inicialmente, a rede é treinada *offline*, sendo estimado um conjunto de pesos. O primeiro dia do período é previsto utilizando este conjunto de pesos. Para o segundo dia, os dados referentes ao dia anterior são incorporados a um conjunto contendo dados referentes aos últimos cinco dias e dados com características de temperatura similares ao dia a ser previsto, com o índice de similaridade sendo obtido através da distância *euclidiana*. Utilizando o conjunto de pesos iniciais como ponto de partida do treinamento, a rede é re-treinada utilizando o conjunto de dados citado, realizando previsões para o segundo dia e sendo adaptada diariamente. A adaptação mensal é feita de maneira similar, porém utilizando como conjunto inicial de pesos aquele obtido um mês atrás, sendo incorporados ao conjunto de treinamento todos os dados referentes ao último mês. A adaptação semanal é feita apenas para as segundas-feiras e finais de semana, incorporando informações do mesmo dia da semana anterior no conjunto de dados utilizado para re-treinamento. Para o conjunto das frentes frias, frentes quentes e feriados, a adaptação é anual.

CHOW e LEUNG [73] desenvolvem um MLP para previsão da curva de carga diária. São utilizadas 81 variáveis de entradas, contendo valores atrasados da série de carga e valores atuais e atrasos de diversas variáveis climáticas, como temperatura,

umidade relativa do ar, índice pluviométrico, nebulosidade, dentre outras. Este modelo apresenta como saída o incremento que deve ser dado à curva de carga do dia anterior para obtenção da curva de carga prevista, ou seja, é aplicada a vigésima quarta diferença à série de carga original, sendo realizadas previsões apenas para dias úteis. Os resultados obtidos para este modelo foram comparados àqueles obtidos por um MLP com a mesma estrutura, porém apresentando como saída desejada a série de carga original não diferenciada, com o modelo proposto apresentando melhor desempenho em termos de erro de previsão médio.

TAYLOR e BUIZZA [88] desenvolvem modelos neurais para previsão da carga horária, utilizando como informações de entrada cenários climáticos. São disponibilizados 51 cenários diferentes de clima, com cada cenário caracterizado pelas entradas climáticas utilizadas pelos modelos propostos. Cada cenário é apresentado à rede, totalizando 51 valores de carga para um dado dia. De posse desses valores, é estimada a função densidade de probabilidade, *pdf*, da carga para o dia em questão, através de histogramas. A previsão da carga é realizada tomando o valor médio da pdf obtida. Para treinamento dos modelos, é aplicada a teoria da regularização, sendo utilizados dois funcionais para controle da complexidade do modelo. O primeiro funcional está relacionado com a soma do quadrado dos pesos que ligam às entradas aos neurônios da camada intermediária, com o segundo associado com a soma do quadrado dos pesos que ligam os neurônios da camada intermediária à saída. Cada funcional está associado a um parâmetro de regularização. Para determinação do número de neurônios na camada intermediária e dos parâmetros de regularização, é utilizando um conjunto independente de dados.

MORI e YUIHARA [140] apresentam um modelo que utiliza um algoritmo de clusterização, chamado *determinstc annealing clustering*, em conjunto com diversos

MLP’s, para previsão do pico de carga diário. O algoritmo de clusterização é utilizado para determinação de clusters dentro do conjunto de treinamento, associando posteriormente a cada cluster um MLP. Os resultados obtidos pelo modelo proposto foram comparados com os obtidos por um MLP convencional, com o primeiro apresentando melhores resultados.

SENJYU *et. al.* [142] propõem um modelo baseado na seleção de dias similares em conjunto com um MLP para previsão da curva de carga diária. Uma curva de carga “base” é estimada através da média aritmética entre as curvas de carga de cinco dias similares, com tal índice de similaridade sendo obtido através da aplicação de uma técnica analítica intitulada norma *euclidiana* com ponderações. A determinação dos dias similares é efetuada em um conjunto restrito de dados, através de uma heurística para seleção de padrões de treinamento, baseada na utilização apenas de dados referentes a meses próximos ao dia onde serão realizadas as previsões.

A modelagem multivariada procura a representação direta dos fatores climáticos e não-climáticos que afetam, de forma não linear, a dinâmica de curto prazo da carga. Em virtude disso, o desempenho destes modelos está diretamente relacionado com a precisão das informações climáticas disponibilizadas e com a forma na qual os fatores não-climáticos, como hora do dia e dia da semana, são codificados. Entretanto, diferentemente dos modelos univariados, estes modelos apresentam como principal virtude capacidade de antecipação de mudanças de curto prazo na carga em função de variações climáticas rigorosas, desde que sejam fornecidas ao modelo informações meteorológicas confiáveis.

1.2.5.2 Modo de treinamento

Este item conterá uma breve descrição dos modelos neurais utilizados para previsão de carga a curto prazo cuja principal inovação reside na estrutura proposta. De

outra forma, serão apresentados os trabalhos que apresentaram uma nova abordagem, no que concerne a tipo de modelo neural e método de treinamento, para solução do problema de previsão de carga a curto prazo. As contribuições serão divididas em dois itens, a saber: modelos supervisionados e modelos não-supervisionados.

1.2.5.2.1 Modelo supervisionado

Em [7] é desenvolvido um estudo sobre a sensibilidade do MLP para o problema de previsão de carga a curto prazo, com tal sensibilidade sendo medida em relação a dez fatores, relacionados com o número de camadas ocultas, tipo de função de ativação dos neurônios das camadas oculta e de saída, algoritmo de treinamento utilizado, inserção de ruído nas entradas, critério de parada do treinamento, considerando o erro para o conjunto de treinamento ou para um conjunto independente de dados, tipo de rede utilizada, *feedforward* ou recorrente, número de padrões utilizados para treinamento, período do ano em que ocorre o pico de carga, inverno ou verão, e porcentagem de consumidores industriais. Como conclusões do artigo, os autores sugerem um conjunto de regras a serem adotadas no desenvolvimento de MLP's para previsão de carga a curto prazo. Entretanto, tal conjunto deve ser utilizado apenas como indicativo, visto que o desenvolvimento de MLP's é uma tarefa extremamente dependente do problema em estudo.

KODOGIANNIS e ANAGNOSTAKIS [9] apresentam um estudo comparativo entre diversos modelos neurais para previsão da curva de carga diária, a saber: MLP treinado através do algoritmo de retropropagação do erro tradicional; MLP treinado com o mesmo algoritmo, porém utilizando uma técnica de representação das entradas da rede chamada *spread encoding*; modelo neural chamado *window random activation weight neural network* (WRAWNN), estrutura onde os pesos sinápticos que ligam as entradas à camada oculta são escolhidos de maneira aleatória, sendo estimados, através de

mínimos quadrados, apenas os pesos que ligam a camada oculta à saída linear; redes de função de base radial (RBF's); redes neurais recorrentes, mais especificamente, a rede neural recorrente de *Elman* modificada; e um sistema de inferência neuro-*fuzzy*. Além destes modelos neurais, também é desenvolvido um modelo auto-regressivo (AR). Utilizando cada uma das estruturas anteriormente descritas, são desenvolvidos 24 modelos, um para cada hora do dia, sendo apresentadas as vantagens e desvantagens de cada metodologia.

Em [22] os autores comparam oito modelos neurais para previsão da carga horária, a saber: MLP treinado através do algoritmo de retropropagação de erro tradicional, com taxa de aprendizagem adaptativa e utilizando uma representação das entradas conhecida como codificação gaussiana (*gaussian encoding*), modelos neurais do tipo WRAWNN [9] e uma variante deste modelo, chamada *moving window regression trained random activation weight neural network* (MWRAWNN's), redes de função de base radial (RBFN's), redes neurais recorrentes treinadas através do algoritmo de aprendizagem em tempo real, *real-time recurrent learning algorithm* (RTRL) e redes neurais recorrentes auto-regressivas. Para cada uma das estruturas propostas são desenvolvidos 24 modelos, um para cada hora do dia, com a saída de modelos horários consecutivos sendo utilizadas como entradas de outros modelos. A saída do modelo que realiza previsões para a primeira hora do dia, por exemplo, é utilizada como uma das entradas do modelo que prevê a carga para a segunda hora do dia. Em termos de precisão das previsões, apresentaram melhor desempenho para os dados do sistema elétrico da ilha de Creta, na Grécia, as redes de função de base radial, a rede neural recorrente auto-regressiva e o MLP que utiliza codificação gaussiana das entradas.

Em [41], os autores propõe a utilização de um MLP não totalmente conectado para previsão da carga horária. O modelo proposto utiliza como entradas valores atrasados de carga e temperatura e informações sobre o dia da semana e sobre a hora do dia, apresentando desempenho razoável para a série de carga da *Wisconsin Electric Power Company*, empresa de energia do estado de Wisconsin, EUA.

RANAWEERA *et. al.* [44] apresentam um modelo baseado em redes de função de base radial, *radial basis function network* (RBFN). O algoritmo dos *k*-vizinhos mais próximos é utilizado para determinação dos centros das funções de base radial. O número de funções de base radial e os parâmetros que definem o algoritmo não-supervisionado são determinados através de validação cruzada múltipla, onde o conjunto de treinamento é dividido em *k* partições, sendo utilizada $(k-1)$ partições para treinamento e uma para validação. São realizadas previsões tanto da demanda máxima quanto da energia total consumida no dia em questão. São desenvolvidos sete modelos para cada caso, um para cada dia da semana, sendo excluído os feriados da base de dados, não sendo realizadas previsões para estes dias. Também é desenvolvido um MLP contendo as mesmas entradas da RBFN, treinado através do algoritmo de retropropagação do erro com parada antecipada (*early stopping*). A RBFN apresentou desempenho melhor que o MLP, em termos de precisão das previsões, além de apresentar as vantajosas características de determinação direta de intervalos de confiança para as previsões e menor requisito de esforço computacional para treinamento em relação ao MLP.

DASH *et. al.* [46] desenvolvem um modelo neural para previsão de carga a curto prazo intitulado *auto-enhanced functional link network* (AEFLN). Esta rede utiliza 25 entradas relacionadas com a expansão da série de carga em série de Fourier e três entradas relacionadas com a temperatura da hora a ser prevista. Um diagrama

esquemático do modelo utilizado é apresentado na Figura 1.1. Nesta figura, $\omega = 2\pi/\tau$, com $\tau = 24$ para previsões 24 passos à frente e $\tau = 168$ para previsões 168 passos à frente. Este modelo pode ser visto como um modelo neural contendo um único neurônio linear, apresentando como entradas um *bias*, 24 componentes senoidais e cossenoidais, uma entrada de temperatura, e uma série de transformações não-lineares desta variável.

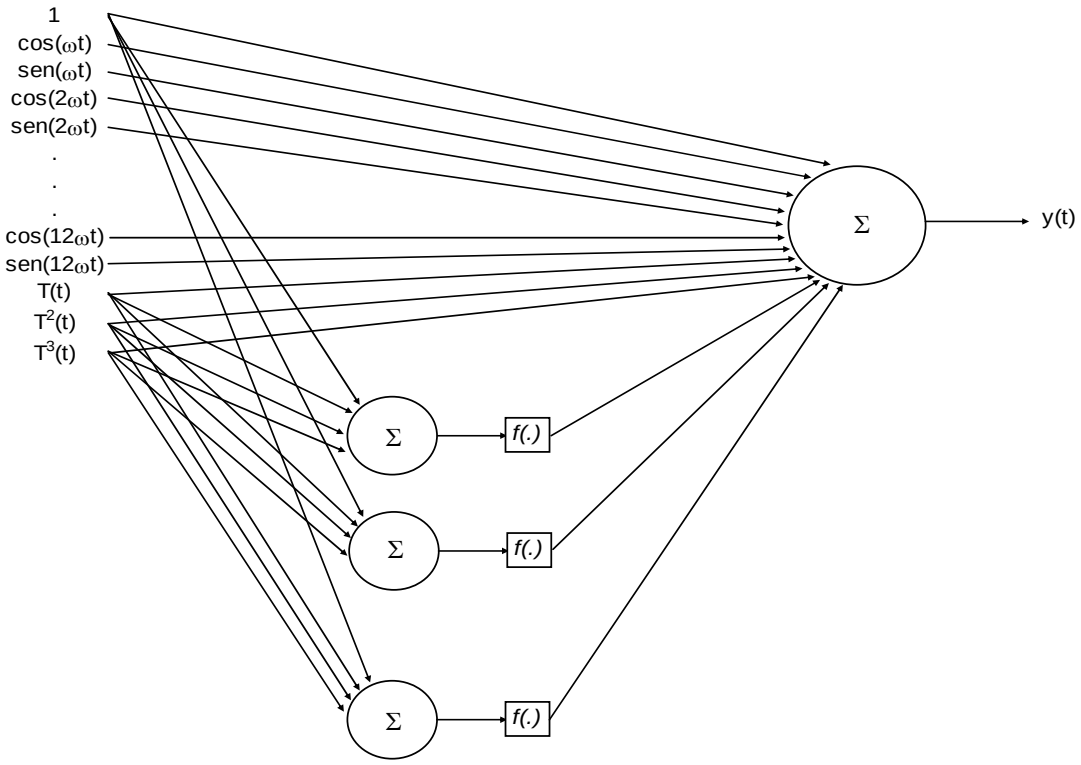

Figura 1.1 – Diagrama esquemático da rede AEFLN utilizada.

Matematicamente, a saída do modelo apresentado na figura é dada por:

$$y(t) = a_o + \sum_{k=1}^{12} a_k sen(2\pi kt/\tau) + b_k \cos(2\pi kt/\tau) + \sum_{i=1}^{3} c_i \left[T(t)\right]^i + \sum_{j=1}^{s} w_j f\left\{\sum_{i=1}^{3} \alpha_i \left[T(t)\right]^i\right\} \quad (1.1)$$

As duas primeiras transformações são "diretas", $\left(T^2(t)\right.$ e $\left.T^3(t)\right)$. As restantes são definidas a partir da transformação não-linear $f(x) = \tanh(x)$ da combinação linear da variável de temperatura, $T(t)$, e das duas transformações iniciais desta variável, $T^2(t)$

e $T^3(t)$, com os coeficientes α_i desta combinação linear sendo definidos de maneira aleatória. A quantidade de transformações não-lineares é definida pelo usuário, ou seja, o número *s* de funções $f(\cdot)$ a serem utilizadas. Os pesos que ligam estas entradas ao neurônio de saída são estimados através da regra de *Widrow-Hoff*. Com o intuito de incrementar a precisão das previsões obtidas pelo modelo proposto, o erro de previsão para as últimas 24 horas também é utilizado como entrada do modelo. Esta estrutura foi testada com dados de uma empresa de energia do estado de Virgínia, EUA, apresentando erro médio em torno de 2,0 % para previsão da curva de carga diária.

ABDEL-ALL [62] apresenta um modelo auto-organizável baseado no método chamado *group method of data handling* (GMDH). Dado um conjunto de padrões contendo *m* entradas e uma única saída, a idéia geral desta metodologia consiste na determinação em cascata de polinômios para realização de previsões. Exemplificando, as *m* entradas originais são combinadas duas a duas, formando $m(m-1)/2$ polinômios de segunda ordem completos, cada um deles responsável pela modelagem da saída desejada. Através de um critério de avaliação da saída de cada polinômio, são escolhidos aqueles que "melhor" aproximem a saída. Desta forma, um novo conjunto de entradas é formado pelas saídas de cada um dos polinômios escolhidos no passo anterior. Este processo é repetido até que o melhor critério de parada seja piorado, ou então até que um determinado limite de complexidade do modelo seja atingido. O método acima descrito é aplicado através da utilização de redes de aprendizado adaptativo, *adaptive learning network* (ALN), mas especificamente aquelas que utilizam o chamado *abductory inductive mechanism* (AIM). Como critério de avaliação, estes modelos utilizam um índice chamado erro quadrático previsto, *predicted squared error* (PSE), que combina o erro para o conjunto de treinamento com uma parcela relacionada com o controle da complexidade do modelo. O modelo proposto é testado

com dados da *Puget Sound Power and Light Company*, apresentando melhores resultados, em termos de precisão das previsões, que o MLP.

LU *et. al.* [73] desenvolvem MLP's para previsão da carga horária, da curva de carga diária, em base horária, e para previsão do pico de carga diário. Os autores testam a sensibilidade dos modelos neurais obtidos com relação a dados corrompidos, verificando a degradação do desempenho dos modelos devido à presença de tais dados, o que já era esperado.

Em [72] os autores apresentam um modelo neural, baseado no MLP, que incorpora às previsões as incertezas relacionadas com as variáveis de entrada utilizadas. Para demonstração da eficiência do algoritmo proposto, é desenvolvido um MLP, treinado através do algoritmo de retropropagação do erro tradicional com parada antecipada do treinamento, para previsão do pico de carga diário. São derivadas expressões aproximadas para incorporação, na saída do modelo neural, das incertezas associadas às variáveis de entrada e para estimação de intervalos de confiança para as previsões. Dentre algumas premissas seguidas para derivação de tais expressões, a principal reside na consideração de ausência de correlação entre as variáveis de entrada, o que não é verificado principalmente quando se utilizam diversos atrasos de uma determinada série de temperatura, por exemplo. Além desta questão, o algoritmo proposto requer uma estimativa para incerteza relacionada a cada variável de entrada, aumentando a necessidade de intervenção por parte do projetista.

Em [76], os autores apresentam dois MLP's para previsão da curva de carga diária, em base de 30 minutos. O primeiro modelo utiliza um MLP convencional, com 104 entradas, contendo informações de carga, temperatura, umidade relativa do ar e de calendário (mês e dia da semana), duas camadas ocultas e 48 saídas, contendo a curva de carga diária prevista. O outro modelo apresenta dois MLP's em cascata. O primeiro

MLP apresenta 16 entradas, contendo informações de energia total, carga, fator de carga, temperatura, umidade relativa do ar, instante da ocorrência do pico de carga e do dia da semana onde serão realizadas as previsões, uma única camada escondida e três saídas, representando as cargas máxima e mínima e a energia total previstas para o dia em questão. Estas três grandezas são utilizadas como entradas de um outro MLP, que apresenta além destas, as mesmas 104 entradas do primeiro modelo desenvolvido, que possui duas camadas escondidas e 48 saídas, representando a curva de carga diária prevista. Apesar do modelo em cascata apresentar desempenho satisfatório em termos de precisão para os dados de uma empresa de energia do Kuwait, não é esperado que tal modelo apresente desempenho consistente, por dois motivos principais. A primeira questão diz respeito às três entradas oriundas de outro MLP, as quais carregam consigo os erros de previsão inerentes ao modelo que as gerou, aumentando com isso o nível de incerteza do modelo final. A segunda crítica está relacionada com a complexidade excessiva do modelo proposto (107 entradas, duas camadas escondidas com 35 neurônios cada e 48 saídas, totalizando 6768 parâmetros livres), aumentando a possibilidade de ocorrência de *overfitting*, diminuindo a capacidade de generalização do modelo desenvolvido. O problema de complexidade excessiva de modelos neurais será apresentado no capítulo 2.

Em [81], é apresentada uma breve descrição dos modelos ARIMA e a conseqüente generalização destes para modelos NARIMA, *nonlinear auto regressive integrated moving average* , com a função não-linear existente nesta estrutura sendo modelada por um MLP. Na realidade, visto que são utilizadas diversas variáveis climáticas como entradas, a estrutura proposta pode ser mais bem definida como NARIMAX, *nonlinear auto regressive integrated moving average with exogenous input*. São desenvolvidos dois modelos neurais, cujas estruturas diferem apenas em

termos das saídas desejadas. O primeiro MLP não leva em consideração a sazonalidade diária existente na série de carga, realizando previsões diretamente sobre a série de carga original, sendo, portanto, um modelo NARX. O segundo modelo neural considera a presença da sazonalidade diária, realizando previsões sobre a série obtida após a aplicação da vigésima quarta diferença à série de carga original, sendo, portanto, um modelo NARIX. Na realidade, o segundo modelo realiza a previsão do valor que deve ser adicionado à curva de carga do dia anterior para obtenção da curva de carga prevista para o dia desejado. Esta última estrutura apresentou os melhores resultados, conforme já era esperado.

Em [87], é descrito o desenvolvimento de um modelo de previsão de carga a curto prazo desenvolvido pelos autores em parceria com FURNAS Centrais Elétricas S.A, intitulado Oráculo. O modelo realiza previsões de carga, em intervalos de 15 minutos, para até uma semana à frente, com tais previsões sendo atualizadas a cada 15 minutos, sem necessidade de intervenção por parte do usuário. Este modelo combina duas estruturas, uma baseada em regressão linear e outra baseada em um modelo neural polinomial do tipo GMDH. Este sistema utiliza séries históricas de carga e temperatura, em conjunto com um histórico de eventos, especificando a ocorrência de eventos especiais, tais como dias atípicos, interrupções de carga e entradas e saídas programadas de consumidores, apresentando resultados satisfatórios tanto em termos de precisão das previsões quanto em praticidade no seu uso, visto que foi considerado um dos sistemas de previsão de carga a curto prazo mais autônomos dentre os existentes na época do desenvolvimento do mesmo.

Em [89], os autores apresentam um estudo comparativo entre diversos algoritmos de treinamento de MLP’s para previsão do pico de carga diário, a saber: algoritmo de retropropagação de erro tradicional e diversos algoritmo de

retropropagação de erro utilizando gradiente conjugado, mais especificamente os algoritmos de *Fletcher-Reeves*, de *Polak-Ribiere* e de *Powell-Beale*, e o algoritmo de retropropagação de erro utilizando gradiente conjugado escalonado. Como critério de parada do treinamento, é utilizado o erro para o conjunto de validação, *early stopping.* Além desta comparação entre estes algoritmos, os autores propõem a aplicação de análise de componentes principais para redução do número de entradas dos modelos apresentados. No caso em estudo, o algoritmo de *Powell-Beale* apresentou o melhor desempenho em termos de precisão das previsões.

PARK *et. al.* [101] apresentam uma metodologia de atualização *on-line* dos pesos sinápticos de um MLP à medida que novos dados são disponibilizados. Para ilustração da técnica proposta, os autores desenvolvem um modelo neural para previsão de carga a curto prazo, apresentando resultados satisfatórios.

Em [103] é apresentado um MLP para previsão dos valores máximo e mínimo da carga diária. Estas previsões são utilizadas para incrementar as previsões realizadas pelo modelo descrito em [37], que obtêm, através de um sistema especialista, a previsão da curva de carga normalizada, com os valores máximo e mínimo sendo necessários para transformação da curva normalizada para a escala de [MW].

VERMAAK e BOTHA [112] desenvolvem duas redes neurais, uma baseada no MLP *feedforward* tradicional e outra baseado em redes recorrentes, para previsão da carga horária. Ambos modelos utilizam informações de carga, temperatura, dia da semana e hora do dia como entradas, com o modelo recorrente apresentando melhores resultados para os dados de uma empresa de energia da África do Sul.

DREZGA e RAHMAN [122] argumentam sobre as principais diferenças entre previsores neurais globais (treinados uma vez por ano, por exemplo) e previsores neurais locais (treinados uma vez por semana, por exemplo), citando a complexidade

dos modelos e os requisitos de dados de cada um. Afirmam que, devido à desregulamentação e conseqüente mudança do comportamento da indústria de energia, os modelos de previsão passaram a utilizar bases de dados menores, favorecendo o uso de previsores locais. É apresentada uma metodologia de seleção do conjunto de treinamento utilizando a técnica dos *k*-vizinhos mais próximos, como também uma técnica de escolha do número de neurônios na camada intermediária, através da chamada simulação piloto (*pilot simulation*). Como entradas do modelo que seleciona o conjunto de treinamento, são utilizadas informações de temperatura. Para seleção das variáveis de entrada dos MLP's, é utilizado o procedimento descrito em [116].

Em [124] os autores apresentam um estudo identificando as influências da estrutura dos modelos neurais na qualidade do modelo de previsão de carga a curto prazo. Para tal, os autores utilizam um MLP contendo uma camada de entrada, duas camadas ocultas e uma saída, para previsão das curvas de carga para os próximos dois dias. Os modelos desenvolvidos são treinados através do algoritmo de retropropagação de erro com taxa de aprendizagem adaptativa. Tais modelos não apresentam realimentação, isto é, para previsão da carga no instante *k*, o valor mais recente de carga utilizado como entrada é aquele verificado no instante $(k-48)$. São realizados diversos experimentos, onde são testadas várias configurações, sendo modificados o número de entradas, número de camadas escondidas e o número de neurônios por camada oculta, comparando os desempenhos obtidos para os conjuntos de treinamento e teste, com o problema do *overfitting* sendo verificado em duas situações. Na primeira, fixada uma arquitetura neural, em uma determinada iteração do algoritmo de treinamento, a minimização do erro para o conjunto de treinamento resulta na deterioração do erro para o conjunto de teste. Na segunda situação, à medida que a complexidade do modelo aumenta, ou seja, o número de parâmetros livres cresce, o erro para o conjunto de

treinamento diminui, ao passo que o erro para o conjunto de teste é degradado. Estas situações são exemplos típicos do problema de *overfitting*, ou *overtraining*, para modelos neurais, evidenciando a necessidade do controle de complexidade dos mesmos com o intuito de maximizar a capacidade de generalização dos modelos estimados.

Em [134] os autores aplicam uma técnica de regularização, baseada em uma metodologia de poda de rede chamada dano cerebral ótimo, *optimal brain damage* (OBD), a um MLP desenvolvido para previsão da curva de carga diária e da carga média diária. A técnica proposta é baseada na estabilização da estrutura, ou seja, determinação do número ótimo de parâmetros a serem utilizados pelo modelo neural. Este tópico será abordado com maior riqueza de detalhes no capítulo 2. Para treinamento do MLP, é utilizado o algoritmo de retropropagação do erro de *Broyden-Fletcher-Goldfarb-Shanno* (BFGS), incluído na categoria dos algoritmos de retropropagação de erro conhecidos como *quasi-Newton*, que fornece automaticamente uma aproximação da matriz *Hessiana* requerida pelo algoritmo de poda de rede utilizado (OBD). Os resultados apresentados confirmam o ganho de desempenho do modelo neural regularizado em relação ao modelo estimado sem regularização, visto que o primeiro apresentou uma melhoria de 17 % e 4,7 %, em termos do erro percentual médio obtido para o conjunto de teste para a previsão da curva de carga diária e para o consumo médio diário, respectivamente, em relação ao segundo. Em termos do erro máximo de previsão para este mesmo conjunto de dados, o ganho de desempenho da rede regularizada foi significativo apenas para o modelo de previsão da curva de carga diária, sendo da ordem de 38 %. Para a previsão do consumo médio diário, este ganho foi da ordem de 3,0 %. A melhoria de performance, em função da aplicação de uma técnica de regularização, verificada apenas para o modelo de previsão da curva de carga diária é devida principalmente ao fato de que este modelo apresenta complexidade

muito maior que o modelo de consumo médio diário, visto que, enquanto o primeiro apresenta 1179 parâmetros livres (23-20-15-24), o último possui 313 parâmetros livres (23-13-1), evidenciando a necessidade maior de controle de complexidade do primeiro.

DASH *et. al.* [138] desenvolvem um MLP cujo algoritmo de treinamento é baseado filtro de *Kalman* adaptativo. Nesta abordagem, os ganhos das funções de ativação dos neurônios da camada escondida, o ganho do filtro de *Kalman* e a taxa de aprendizagem são ajustados ao longo do treinamento visando à minimização do erro para o conjunto de treinamento. O modelo proposto é comparado com dois outros MLP's, um treinado através do algoritmo de retropropagação do erro tradicional e outro através da utilização do filtro de *Kalman* original, ou seja, sem adaptação, apresentando os melhores resultados tanto em termos de performance quanto de convergência.

Conforme verificado acima, abundam na literatura contribuições de estruturas neurais supervisionadas para previsão de carga a curto prazo. Visto que esta abordagem utiliza modelos neurais *feedforward*, a principal vantagem destes modelos reside na sua capacidade intrínseca de aproximar, com precisão arbitrária, qualquer função contínua não-linear. Em outras palavras, é esperado que um modelo deste tipo, estimado de maneira adequada, possa representar fidedignamente a dinâmica de curto prazo da carga. Entretanto, para problemas de regressão como o de previsão de carga, onde os dados estão sujeitos a ruído de medição, esta qualificada virtude dos modelos *feedforward* pode ser prejudicial, visto que estes, além de representar o comportamento regular da série em estudo, podem modelar também o ruído presente nesta.

1.2.5.2.2 Modelo não-supervisionado

Em [3] os autores propõem a utilização de um modelo neural não-supervisionado em conjunto com um modelo supervisionado para previsão da curva de carga diária, em base horária. O modelo não-supervisionado é baseado nos chamados

mapas auto-organizáveis, enquanto que o modelo supervisionado é baseado em redes *feedforward* conhecidas como *functional link network* (FLN). O mesmo conjunto de entradas é apresentado aos dois modelos. O modelo não-supervisionado determina quais padrões integrantes da base de dados serão utilizados para treinamento do modelo supervisionado. São testadas duas estruturas, diferindo entre si apenas no número de entradas, com ambas apresentando desempenho satisfatório para dados de uma empresa de energia da Sérvia.

Em [100], os autores apresentam um estudo comparativo entre o MLP e árvores de decisão, através da aplicação destes modelos a três problemas, um dos quais a previsão de carga horária. Neste trabalho, para previsão de carga horária, o MLP apresentou ligeira vantagem em relação à árvore de decisão.

Em [102] e [129], os autores propõem a utilização de um mapa de *Kohonen* em conjunto com um MLP para previsão de carga a curto prazo. Inicialmente, em [129], é proposto um mapa auto-organizável de *Kohonen* para classificação dos tipos de dias, utilizando para tal um conjunto de curvas de carga normalizadas pelos valores máximo e mínimo da carga verificados no respectivo dia. Para realização de previsões, primeiramente o dia em que a previsão será realizada é classificado tomando por base os resultados obtidos pelo modelo não-supervisionado para o mesmo mês do ano anterior. Por exemplo, para realização de previsões para o dia 05/05/1987, quinta-feira, este dia é inicialmente classificado pelo mapa de *Kohonen* como um dia normal de trabalho do mês de maio, através da classificação dos 30 dias do mês de maio de 1986. Portanto, a curva de carga normalizada prevista é obtida através da curva de carga normalizada média obtida para os últimos dez dias normais de trabalho antes do dia 05/05/1987. Obtida a curva normalizada, é necessária a previsão dos valores mínimo e máximo da carga para o referido dia, com o intuito de obter a curva de carga diária, em [MW]. Esta

é a função dos modelos descritos em [102], que utilizam dois MLP's para previsão do pico de carga e do valor mínimo de carga, respectivamente. A abordagem proposta é interessante, com a ressalva da não-utilização de todo o potencial do mapa de *Kohonen*, visto que a classificação realizada por este modelo poderia ser feita de maneira empírica, através da separação da base de dados por dia da semana, por exemplo. Outro comentário pertinente diz respeito à complexidade excessiva dos MLP's utilizados com relação ao pequeno número de dados relacionados para treinamento. Os modelos supervisionados apresentam quarenta e seis entradas, duas camadas ocultas contendo quarenta e seis neurônios em cada camada, e uma única saída, totalizando 4371 parâmetros a serem estimados. No entanto, para estimação dessa infinidade de parâmetros, são utilizados apenas 35 padrões de treinamento. Portanto, a possibilidade de ocorrência de *overfitting* é elevada, visto que o número de parâmetros é muitas ordens de grandeza maior que o número de dados disponíveis para treinamento [114].

Em [121], é proposto um algoritmo adaptativo baseado em modelos neurais topológicos, mais especificamente o modelo de *Lutrell*, para previsão da carga horária. Dentre as vantagens destacadas pelos autores, além do fato do modelo proposto apresentar resultados similares aos apresentados por outros modelos, como o ANNSTLF, a estrutura proposta é parcimoniosa e facilmente adaptável, ou seja, requer esforço computacional mínimo para treinamento, ao contrário dos modelos baseados no MLP.

BECCALI *et. al.* [126] combinam mapas auto-organizáveis de *Kohonen* com MLP's para previsão da curva de carga diária. Primeiramente, o mapa de *Kohonen* divide a base de dados em diversos clusters, tomando por base a curva de carga diária, associando a cada curva de carga um rótulo referente ao cluster à qual pertence. Realizada a identificação de cada padrão da base de dados, é utilizado um MLP para

realização das previsões efetivamente. Este MLP apresenta 136 entradas, contendo informações de carga, temperatura, radiação solar, umidade relativa do ar, do dia onde serão realizadas as previsões e do cluster ao qual o dia anterior está associado, uma camada escondida e 24 saídas lineares, contendo a curva de carga prevista. Para treinamento do modelo é utilizado o algoritmo de retropropagação do erro com taxa de aprendizagem adaptativa e constante de momento, com o critério de parada definido pelo erro para o conjunto de teste, *early stopping*. São utilizados dados de carga da cidade de Palermo, Itália, para teste da metodologia proposta, com a mesma apresentado resultados razoáveis em termos de precisão das previsões.

Em [143], os autores apresentam um modelo hierárquico para previsão da carga horária. Este modelo apresenta dois mapas de *Kohonen* em cascata, ou seja, a saída do primeiro é apresentada como entradas para o segundo. Utilizando dados de uma empresa de energia brasileira, o modelo proposto foi comparado com o MLP, apresentando melhores resultados.

Os modelos neurais não-supervisionados, também conhecidos como estruturas auto-organizáveis, quando aplicados para previsão de carga a curto prazo, são utilizados na grande maioria dos trabalhos para agrupamento da base de dados. Este agrupamento apresenta como principal objetivo a detecção de padrões similares de carga, buscando desenvolver modelos específicos para cada grupo. Desta forma, são aglutinados os padrões que apresentarem o mesmo comportamento diante de certos fatores exógenos, evitando a necessidade de representação dos mesmos.

1.2.5.3 Sistemas baseados em lógica *fuzzy*

MASTOROCOSTAS *et. al.* [51] utilizam um modelo *fuzzy* para previsão da curva de carga diária, mais especificamente o modelo TSK, desenvolvido por *Takagi, Sugeno e Kang*. No modelo *fuzzy* proposto, é necessária a identificação das premissas e

dos conseqüentes das regras. Na identificação das premissas, a tarefa consiste na partição do espaço em regiões *fuzzy* extraindo, posteriormente, o conjunto de regras. Já para os conseqüentes, é necessária a identificação das variáveis de entrada a serem utilizadas pelos modelos lineares e a estimação dos parâmetros destes. No artigo, é utilizado o método dos mínimos quadrados ortogonais para determinação das premissas e dos conseqüentes. Primeiramente, o método é aplicado para determinação das premissas e seus parâmetros, sendo posteriormente aplicado para determinação das variáveis de entrada e estimação dos modelos lineares associados a cada uma das regras obtidas na primeira parte. São desenvolvidos 28 modelos, ou seja, um para cada dia da semana de cada estação do ano, para previsão da curva de carga horária. Dentre as principais vantagens do modelo utilizado, podem ser citadas extração automática de regras, determinação automática das variáveis de entrada e simplicidade do modelo.

Em [55] e [145] os autores desenvolvem um sistema de inferência *fuzzy* para a previsão da curva de carga diária. Os autores apresentam um algoritmo para determinação automática do conjunto de regras que constituem o sistema de inferência e das diversas funções de pertinência associadas a cada variável *crisp*.

SENJYU *et. al.* [115] apresentam um modelo baseado na busca por dias similares na base de dados, em conjunto com lógica *fuzzy*, para previsão de carga a curto prazo. Para determinação dos dias similares, é utilizado o mesmo procedimento descrito em [142], que faz uso de um conjunto restrito de dados e um índice de similaridade chamado de norma *euclidiana* ponderada. São selecionados cinco dias similares, sendo aplicada uma taxa de crescimento anual para correção das respectivas curvas de carga. A previsão é realizada através da média ponderada das cinco curvas similares obtidas, com estas ponderações estimadas por um sistema de inferência *fuzzy*, que possui um conjunto de regras formulado tomando por base o conhecimento dos

operadores da empresa em estudo, no caso a *Okinawa Electric Power Company*, empresa de energia elétrica japonesa.

MORI e KOBAYASHI [144] desenvolvem um sistema de inferência *fuzzy* para previsão da carga horária, utilizando recozimento simulado, *simulated annealing* (SA), para determinação e estimação automáticas do número de funções de pertinência associadas a cada variável *crisp* e dos parâmetros de cada uma delas, respectivamente.

Os sistemas baseados em lógica *fuzzy* utilizam conjuntos de regras para previsão de carga a curto prazo, representando o conhecimento dos operadores do sistema elétrico de forma linguística. Estas regras podem ser obtidas de maneira automática, através da extração automática de regras, ou não-automática, por meio de interação com operadores. A viabilidade da extração automática de regras, apesar de prescindir da intervenção de especialistas, depende da qualidade dos dados da série de carga em questão.

1.2.6 Sistemas híbridos

Nesta seção, são apresentados alguns sistemas híbridos encontrados na literatura utilizados para previsão de carga a curto prazo. Tais sistemas nada mais são que combinações de alguns dos modelos citados anteriormente, como modelos neuro-*fuzzy*, sistemas especialistas em conjunto com MLP's, programação evolucionária para treinamento de MLP's, dentre outros.

Em [4] é utilizado um modelo híbrido, combinando um MLP com um sistema de inferência *fuzzy*, para previsão da carga horária. Na realidade, o sistema de inferência *fuzzy* é responsável pela modificação das previsões efetuadas pelo MLP, utilizando para tal informações de temperatura e ocorrência de feriados. O modelo proposto foi implementado na *Korea Electric Power Corporation*, empresa de energia da Coréia do Sul, apresentando performance satisfatória.

TAMIMI e EGBERT [12] apresentam um modelo combinando lógica *fuzzy* e MLP's para previsão da carga horária. O sistema de inferência *fuzzy* utiliza informações de temperatura para determinação da variação da curva de carga em relação ao dia anterior, em função das condições climáticas. Esta informação é utilizada como entrada do MLP, que utiliza também informações meteorológicas e de carga para previsão da carga horária. O modelo proposto é comparado com um MLP e com um modelo ARMA, apresentando os melhores resultados em termos de precisão das previsões. Apesar dos resultados razoáveis apresentados, uma estratégia mais consistente seria a utilização do sistema de inferência *fuzzy* para correção das previsões realizadas pelo MLP, e não o inverso, conforme proposto.

SRINIVASAN *et. al.* [21] desenvolvem um modelo combinando mapas de *Kohonen* e lógica *fuzzy* para previsão da curva de carga diária. Os autores desenvolvem um mapa de *Kohonen* para cada dia da semana, dedicando também um específico mapa para feriados, objetivando a obtenção de curvas típicas, tomando por base a curva de carga do dia anterior. O ajuste da curva típica estimada pelo modelo auto-organizável, em função das condições climáticas e do conhecimento dos operadores do sistema, é determinado por um sistema de inferência *fuzzy*.

SRINIVASAN [49] propõe a utilização de algoritmos genéticos para treinamento de MLP's para previsão de carga a curto prazo. No contexto apresentado, o algoritmo genético é utilizado apenas para estimação dos pesos da rede neural, utilizando como função de adequabilidade o erro para o conjunto de treinamento, não sendo utilizado para determinação da estrutura do MLP, isto é, número de entradas e neurônios na camada intermediária.

Em [50] é desenvolvido um modelo auto-organizável *neuro-fuzzy* para previsão do pico de carga e da carga média diária. Basicamente, este modelo pode ser

considerado como um modelo *feedforward* contendo quatro camadas. A primeira camada apresenta as variáveis *crisp* a serem utilizadas. A segunda é responsável pela *fuzzyficação* de tais entradas. A terceira realiza a inferência tomando por base as variáveis lingüísticas fornecidas como saída da segunda camada. A última camada efetua a *defuzzyficação* das saídas do sistema de inferência contido na camada anterior. Os parâmetros que definem as funções de pertinência, a camada de inferência e o sistema de *defuzzyficação* são estimados através de um algoritmo baseado na idéia de retropropagação do erro. Também são apresentadas algumas heurísticas para determinação do número satisfatório de entradas e de regras, isto é, número de neurônios na camada de inferência, com o intuito de otimizar a estrutura do modelo. Apesar do modelo utilizar treinamento supervisionado, os autores intitulam a estrutura de auto-organizável em função da maneira automática em que os parâmetros que definem o sistema *fuzzy* são obtidos. Formalmente, modelos auto-organizáveis são aqueles que utilizam métodos não-supervisionados de treinamento.

WU e LU [52] propõe um modelo baseado em lógica *fuzzy*, em conjunto com modelos ARX, para previsão da curva de carga diária. Inicialmente, os autores utilizam análise de variância (ANOVA) para determinação das entradas de carga e temperatura a serem utilizadas. Após a escolha das entradas a serem utilizadas, é aplicado um algoritmo de clusterização da base de dados e determinação do conjunto de regras integrantes do sistema de inferência *fuzzy*. Na realidade, o número de variáveis lingüísticas associadas a cada variável *crisp*, como também o número de regras do sistema de inferência, é igual ao número de clusters obtidos. Para cada regra, é estimado um modelo ARX, com a previsão sendo gerada através da *defuzzyficação* das saídas do sistema de inferência. O modelo proposto é comparado com o MLP e com o modelo de *Box-Jenkins*, apresentando melhores resultados.

Em [54] é desenvolvido um modelo baseado em regressão linear múltipla em conjunto com um sistema de inferência *fuzzy* para previsão do pico de carga e do valor mínimo de carga para um dado dia. De uma maneira geral, o sistema de inferência *fuzzy* fornece o ajuste que deve ser feito à previsão realizada pelo modelo de regressão múltipla, em função das condições climáticas e do conhecimento prévio dos operadores.

LIANG e CHENG [60] propõem um modelo de previsão da curva de carga diária baseado em sistemas neuro-*fuzzy*. Inicialmente, são estimadas curvas de carga normalizada para cada dia da semana, com a normalização de cada curva sendo realizada em função do pico de carga e do valor mínimo de carga verificado para o respectivo dia. Em seguida, é aplicado o método de *Pearson* às séries de pico de carga e de valor mínimo de carga diário para determinação dos atrasos a serem utilizados como entradas dos modelos que realizarão previsões sobre estas séries. Estas entradas são então *fuzzyficadas*, com as variáveis lingüísticas associadas a cada uma delas sendo utilizadas como entradas de dois MLP's, um para o pico e outro para o valor mínimo. As saídas dos respectivos MLP's são posteriormente *defuzzyficadas*, fornecendo as previsões do pico de carga e do valor mínimo de carga diário. Por último, estes valores são corrigidos por um sistema de inferência *fuzzy*, fornecendo as respectivas previsões finais, que são utilizadas para previsão da curva de carga diária, através da conversão da curva normalizada típica estimada para o dia da previsão.

Em [70] os autores apresentam uma rede neural *fuzzy*, *fuzzy neural networks* (FNN), para previsão da curva de carga semanal, em base horária. São desenvolvidos modelos independentes para cada hora de cada dia da semana, totalizando 168 modelos, que utilizam como entradas apenas valores atrasados de carga. O modelo proposto é comparado com um MLP tradicional, utilizando dados de carga do sistema interligado grego, apresentado resultados similares ao último com relação à precisão das previsões.

Entretanto, os autores afirmam que a metodologia proposta apresenta como principal vantagem em relação ao MLP o menor esforço computacional requerido para treinamento da FNN.

PIRAS *et. al.* [71] utilizam um modelo híbrido contendo 3 módulos. O primeiro consiste em um sistema não-supervisionado de clusterização, chamado *Neural Gas*, que divide a base de dados em dois clusters, com cada cluster sendo definido pelo seu vetor médio e pela sua matriz de covariância. O segundo consiste em dois MLP's, um para cada cluster. As saídas de cada MLP são ponderadas através de uma média *fuzzy*, com as ponderações sendo definidas através de funções de pertinência gaussianas definidas pelo vetor médio e pela matriz de covariância de cada um dos clusters.

HUANG e HUANG [74] propõem a combinação de algoritmos genéticos com o algoritmo de retropropagação de erro para treinamento de MLP's para previsão da carga horária, combinando as características globais e locais de cada algoritmo. Visto que o algoritmo de retropropagação de erro é extremamente dependente do ponto inicial, o objetivo da metodologia proposta reside na busca por este ponto inicial através de algoritmos genéticos, que apresentam como principal característica capacidade de determinação de regiões promissoras no espaço de busca, visto que esta técnica de otimização realiza a busca em diversas regiões deste espaço. O erro para o conjunto de treinamento é utilizado como função de adequabilidade do algoritmo genético, sendo utilizado *early stopping* como critério de parada do algoritmo de retropropagação do erro.

Em [77] os autores apresentam um modelo híbrido, combinando sistemas especialistas e redes neurais, para previsão da carga horária. Neste modelo, a saída prevista pelo sistema especialista é utilizada como entrada do MLP, sendo também estimados intervalos de confiança para as previsões realizadas. Apesar do sistema

proposto apresentar bons resultados para dados de uma empresa de Taiwan, seria mais interessante desenvolver uma metodologia inversa à proposta no artigo, ou seja, utilizar um sistema especialista para correção da saída de um MLP, com o intuito de agregar a este último modelo o conhecimento dos especialistas acerca do problema em estudo.

KIATZIS *et. al.* [78] propõem um modelo que combina as previsões realizadas por três modelos, dois baseados em regressão linear e um MLP, para a previsão da curva de carga diária da *Greek Public Power Corporation*, empresa de energia grega. As previsões realizadas por cada um dos modelos são combinadas através da aplicação recursiva da regra de *Bayes*, levando em consideração os erros de previsão de cada um dos modelos para os dados disponíveis. Comparada com as previsões realizadas por cada um dos três modelos de maneira isolada, a estrutura proposta apresentou os melhores resultados.

Em [86] os autores apresentam uma rede baseada em *wavelets* para previsão de carga horária, utilizando algoritmos genéticos para treinamento do modelo proposto. Esta estrutura é testada com dados da *Taipower*, empresa de energia elétrica de Taiwan, apresentando resultados satisfatórios.

KHOTANZAD *et. al.* [90] apresentam uma inovação do software ANNSTLF, incorporando os efeitos do preço da energia no comportamento da carga em mercados desregulamentados. Na metodologia proposta, as previsões realizadas pelo ANNSTLF apresentado em [48] são corrigidas por um sistema de inferência *fuzzy*, que modela as variações na dinâmica da carga em função da flutuação do preço da energia e das condições climáticas. Os parâmetros que definem o sistema *fuzzy*, como número de funções de pertinência, tipos e parâmetros de tais funções, número e estrutura das regras, são determinados através da aplicação de algoritmos genéticos para extração automática destes parâmetros.

Em [93] os autores propõe um modelo neuro-*fuzzy*, treinado através da aplicação de algoritmos genéticos e recozimento simulado, para a previsão da curva de carga diária. Como função de adequabilidade do algoritmo genético, é utilizado o erro relativo para o conjunto de treinamento. É esperado que a estratégia proposta apresente melhores resultados se na função de adequabilidade for inserido um termo regularizador.

SRINIVASAN *et. al.* [105] desenvolvem um modelo baseado em lógica *fuzzy* em conjunto com um MLP para previsão da curva de carga diária. Inicialmente, um conjunto de variáveis *crisp*, abrangendo informações de carga, temperatura e proximidade de feriados, é apresentado a um sistema de inferência *fuzzy*, que determina a mudança no comportamento da carga ao longo do dia, em função das variáveis observadas e do conhecimento prévio dos operadores, embutido no conjunto de regras integrante deste sistema de inferência. As saídas deste módulo são utilizadas como entradas do MLP, que além destas, apresenta mais um conjunto de entradas *fuzzy* relacionadas com as temperaturas máxima e mínima. Este modelo apresenta uma camada escondida e 24 saídas, contendo a curva de carga *fuzzyficada* prevista. Para obtenção das previsões em escala real, tais previsões são apresentadas a um módulo *defuzzyficador*, responsável pela previsão final. O modelo proposto é comparado com um MLP tradicional, com um modelo baseado em regressão múltipla e com um modelo ARMA, apresentando os melhores resultados.

Em [107] é apresentado um modelo neuro-*fuzzy* para previsão da curva de carga diária, bastante similar ao desenvolvido em [105], com a principal diferença entre os sistemas propostos residindo nas entradas do MLP utilizado em série com um sistema de inferência *fuzzy*. Neste trabalho, o MLP utiliza apenas as entradas referentes à mudança no comportamento da carga ao longo do dia, fornecida como saída do sistema

de inferência *fuzzy*, desprezando as variáveis climáticas utilizadas pelo MLP desenvolvido em [105]. Os resultados obtidos em [107] são ligeiramente superiores aos obtidos em [105], com vantagem do primeiro ser mais parcimonioso que o último, em termos de variáveis de entrada e parâmetros livres.

YANG e HUANG [113] desenvolvem um modelo que combina lógica *fuzzy*, programação evolucionária, árvore de decisão, heurísticas e modelos ARMAX na previsão de carga horária. Após a determinação do conjunto de variáveis *crisp* a serem utilizadas, através da análise das funções de autocorrelação e correlação cruzada, é utilizado um algoritmo que combina lógica *fuzzy*, árvores de decisão, programação evolucionária e algumas heurísticas para determinação automática do número de variáveis lingüísticas associadas a cada variável *crisp*, estimação dos parâmetros que definem as funções de pertinência associadas e elaboração do conjunto de regras constituinte do sistema de inferência *fuzzy*. Como conseqüente de cada uma das regras, é estimado um modelo ARMAX, através da metodologia de *Box-Jenkins*, com a previsão sendo obtido através da *defuzzyficação* do resultado obtido pelo sistema de inferência *fuzzy*.

Em [117] é apresentado um modelo para previsão da curva de carga diária para até cinco dias à frente, baseado em um classificador *fuzzy* em conjunto com um MLP. O sistema de classificação *fuzzy* divide a base de dados de carga em 48 clusters, utilizando como entradas informações de temperatura e umidade relativa do ar. Posteriormente, é desenvolvido um MLP para cada uma das classes, utilizando para treinamento todos os padrões que pertencem à respectiva classe, ou seja, aqueles que apresentam pertinência não-nula para esta a classe em questão. Entretanto, na etapa de previsão, apenas um MLP é selecionado, sendo escolhido aquele que estiver associado ao cluster que apresentar maior pertinência para o dia da previsão. Seria mais interessante utilizar

todos os MLP's associados a clusters com pertinência não-nula para o dia da previsão, obtendo a previsão final através da *defuzzyficação* da saída de cada um.

SRINIVASAN *et. al.* [118] desenvolvem um sistema híbrido que combina mapas de *Kohonen*, MLP e um sistema *fuzzy* para previsão da curva de carga diária. Inicialmente, são utilizados mapas de *Kohonen* para classificação dos dias da semana e obtenção de curvas de carga típicas associadas a cada dia. Posteriormente, para correção da curva típica em função das mudanças climáticas, é desenvolvido um MLP responsável pele modelagem da relação entre a carga e as variáveis climáticas, no caso temperatura. Por último, com o intuito de agregar ao modelo o conhecimento prévio dos operadores do sistema, é desenvolvido um sistema de inferência *fuzzy* para correção da previsão realizada pelos dois modelos anteriores. A estrutura proposta é testada utilizando dados de carga e temperatura da *Singapore Power Pte Ltd.*, empresa de energia elétrica de Cingapura, apresentando performance satisfatória.

Em [120] é apresentado um modelo híbrido, combinando MLP's e lógica *fuzzy* para previsão da carga horária. Para tal, os autores dividem a carga em duas parcelas, uma componente chamada de carga base, independente das condições climáticas, e outra componente sensível em relação às condições climáticas. A primeira componente é modelada por um MLP, com a parcela relacionada com a temperatura sendo prevista por um sistema de inferência *fuzzy*.

KIM *et. al.* [132] desenvolvem um modelo neural, em conjunto com um sistema de inferência *fuzzy*, para previsão de carga em dias atípicos, como feriados e dias adjacentes a estes. Segundo a metodologia proposta, a série de carga é normalizada, tomando por base os valores mínimo e máximo de carga verificados em cada dia, com os MLP's desenvolvidos realizando previsões sobre a série normalizada. Para previsão dos valores máximo e mínimo, é utilizado um sistema de inferência *fuzzy*. O modelo

proposto é testado com dados de uma empresa de energia sul-coreana, apresentados resultados razoáveis para os chamados dias atípicos.

Em [146] os autores propõem a utilização de um sistema especialista em conjunto com lógica *fuzzy* para previsão de carga a curto prazo. O sistema especialista modela o conjunto de regras e heurísticas utilizadas pelos operadores da empresa de energia em estudo, enquanto que o sistema *fuzzy* tenta modelar as incertezas existentes nas variáveis climáticas e nos modelos estatísticos utilizados. O modelo proposto foi implementado em uma empresa de energia elétrica de Taiwan, apresentando resultados satisfatórios para previsão de carga horária.

DASH *et. al.* [147] desenvolvem três FNN's para previsão do pico de carga diária e da carga média diária. O primeiro modelo utiliza uma FNN cujas entradas e saídas são variáveis lingüísticas. As entradas relacionadas com as condições climáticas e valores atrasados da carga, e a saída relacionada com a carga a ser prevista, seja ela horária ou o pico de carga diário. Os outros dois modelos combinam redes neurais com sistemas especialistas *fuzzy*, com a rede neural sendo treinada para extração automática de um conjunto de regras *fuzzy*.

Em [148] os autores apresentam modelos híbridos, combinando MLP, lógica *fuzzy* e sistemas especialistas para previsão da carga horária, da curva de carga horária, da carga média diária e do pico de carga diária. De uma maneira geral, um conjunto de variáveis *crisp* é *fuzzyficado*, com cada variável possuindo três valores lingüísticos: baixa, média e alta. A pertinência associada a cada variável lingüística é apresentada a um MLP, que fornece três saídas contendo as pertinências associadas às três variáveis lingüísticas da carga prevista: baixa, média e alta. Esta saída é *defuzzyficada*, fornecendo uma estimativa inicial da carga prevista. O conjunto de variáveis lingüísticas utilizadas como entradas do MLP é apresentado a um sistema especialista *fuzzy*,

constituído por um conjunto de regras e um sistema de inferência, onde está inserido o conhecimento dos operadores da empresa em estudo. Este sistema especialista fornece a correção que deve ser aplicada à previsão realizada pelo módulo *neuro-fuzzy*. Segundo os autores, a metodologia proposta apresentou resultados promissores para todos os horizontes estudados no trabalho.

Em [149] os autores propõe a utilização de programação evolucionária para determinação da ordem e estimação dos parâmetros que definem os modelos ARMAX, para previsão de carga horária. Como função de adequabilidade do algoritmo de programação evolucionária, os autores utilizam dois índices, um para avaliação da ordem do modelo e outro para verificação da estimativa dos parâmetros. O primeiro índice utilizado é FPE, *final prediction error*, de *Akaike*, enquanto que, para avaliação do modelo estimado, é utilizada a variância do erro para o conjunto de treinamento. A abordagem proposta é comparada com as técnicas tradicionais de determinação da ordem e estimação dos parâmetros de modelos ARMAX, apresentando melhores resultados para os dados de carga e temperatura da *Taipower*, empresa de energia de Taiwan.

LING *et. al.* [150] desenvolvem um modelo neural alternativo, com os neurônios da camada oculta possuindo duas funções de ativação, uma chamada estática e outra dinâmica. A saída da função de ativação estática depende apenas das entradas do neurônio, enquanto que a saída da dinâmica depende da saída de outros neurônios e da saída da sua função de ativação estática. Na realidade, cada função de ativação, estática e dinâmica, apresenta mais duas funções associadas. A determinação de qual função utilizar depende do somatório das entradas de cada função de ativação. Portanto, neste tipo de rede, os neurônios da camada oculta tem comunicação entre si. O conjunto de pesos que define esta rede é determinado através de algoritmos genéticos. Para

avaliação da aplicabilidade da estrutura proposta, este modelo é aplicado para previsão da curva de carga diária, apresentando melhores resultados que o MLP.

Os modelos híbridos buscam agregar as características vantajosas dos diversos métodos e amenizar as desvantagens de cada um deles. Exemplificando, estes modelos podem agrupar a base de dados disponível, aglutinando os padrões similares de carga via treinamento supervisionado, desenvolver modelos de previsão para cada grupo através de redes *feedforward*, combinar, utilizando sistemas *fuzzy*, as previsões realizadas para cada grupo, corrigir, através de um sistema de inferência *fuzzy*, tais previsões em virtude da ocorrência de eventos especiais ou mudanças climáticas bruscas, etc. Portanto, ao combinarem diversas metodologias no desenvolvimento de sistemas de previsão de carga a curto prazo, os modelos híbridos vêm apresentando desempenho promissor.

1.3 Objetivos

Conforme exposto no item 1.1, o problema de previsão de carga pode ser classificado em três tipos, com cada um deles apresentando características próprias e diversas abordagens. Além disto, os modelos neurais, conforme verificado no item 1.2, apresentam maior número de aplicações para o horizonte de curto prazo, no que concerne aos trabalhos encontrados na literatura, comprovando o sucesso desta abordagem para este tipo de problema. Entretanto, uma das principais características encontradas nestas aplicações diz respeito ao empirismo na seleção da estrutura e até mesmo na regularização dos modelos neurais desenvolvidos. Diante disso, a motivação deste trabalho reside na implementação de algumas técnicas de regularização no desenvolvimento de modelos neurais para previsão de carga a curto prazo, buscando selecionar e regularizar de maneira sistemática a estrutura dos modelos neurais propostos.

Uma das principais vantagens dos modelos neurais reside no teorema da aproximação universal [151], que demonstra que estes modelos, com uma única camada oculta contendo um número suficiente de neurônios, podem aproximar com precisão arbitrária qualquer função contínua não-linear. Entretanto, na presença de dados ruidosos, esta vantajosa característica dos modelos neurais pode ser prejudicial, visto que estes, ao invés de representar o comportamento regular da série em estudo, podem modelar o ruído existente na série, resultando na degradação do desempenho para novos dados que não aqueles utilizados no treinamento, comprometendo, assim, a capacidade de generalização dos modelos desenvolvidos. Este problema é conhecido como *overfitting* ou *overtraining*, ilustrando a necessidade de controle da complexidade dos modelos neurais, com o intuito de maximizar a capacidade de generalização dos mesmos.

Apesar desta indesejável característica inerente aos modelos neurais, a grande maioria das aplicações destas estruturas para previsão de carga a curto prazo encontrada na literatura não aborda este problema, conforme pode ser verificado em [3], [4], [8], [9], [12], [22], [39]-[43], [45], [46], [48]-[50], [60], [66], [67], [69]-[71], [76]-[78], [81], [85], [86], [93], [95], [102], [103], [106], [107], [112], [117], [120], [130], [131], [135], [138]-[140], [142], [148], [150]. Dentre as propostas que utilizam alguma técnica de controle de complexidade, a metodologia mais encontrada na literatura é a parada antecipada do treinamento, *early stopping*, implementada em [5], [7], [44], [57], [72], [74], [80], [88], [89], [118], [122], [126], [132], [133]. Além do *early stopping*, existem na literatura outras propostas de controle de complexidade de modelos neurais para a previsão de carga a curto prazo.

ABDEL-AAL [62] apresenta um modelo auto-organizável baseado no método chamado *group method of data handling* (GMDH) para previsão de carga a curto prazo,

conforme descrito no item 1.2.5. O critério de avaliação utilizado no algoritmo proposto faz uso do índice chamado erro quadrático previsto, *predicted squared error* (PSE), que combina o erro para o conjunto de treinamento com uma parcela relacionada com o controle da complexidade do modelo.

Em [114], os autores abordam o problema da estimação de modelos neurais para previsão de carga a curto prazo na presença de poucos padrões para treinamento, em relação ao número de parâmetros livres a serem estimados. Para solução deste problema, é proposta uma técnica de controle de complexidade de modelos neurais, baseada no controle da estrutura do modelo, através da redução do número de variáveis de entrada e da escolha apropriada do número de neurônios na camada intermediária. Para seleção das variáveis de entrada, os autores apresentam um algoritmo que utiliza uma estatística chamada *difference-based estimator of residual variance* (DBERV). Esta estatística procura medir a importância individual de cada variável de entrada na previsão da saída, considerando que "boas" variáveis de entrada são aquelas em que pequenas variações na mesma provocam pequenas variações na variável de saída. Esta premissa é similar à de suavidade na função a ser aproximada, considerada na teoria de regularização, a ser apresentada no capítulo 2. Após a definição do conjunto de variáveis de entrada, é proposto um algoritmo para determinação do número de neurônios na camada intermediária utilizando a combinação de duas técnicas conhecidas como *projection pursuit regression* (PPR) e *slicing inverse regression* (SIR).

BACZYNSKI e PAROL [124] apresentam um estudo identificando as influências da estrutura dos modelos neurais na qualidade do modelo de previsão de carga a curto prazo. Neste estudo, a ocorrência de *overfitting* é verificada em duas situações típicas: minimização única e exclusiva do erro para o conjunto de treinamento

e complexidade excessiva dos modelos estimados, no que diz respeito ao número de parâmetros livres.

Em [134], os autores aplicam uma técnica de regularização, mais especificamente o algoritmo de poda de rede chamado dano cerebral ótimo, *optimal brain damage* (OBD), a um MLP desenvolvido para previsão da curva de carga diária.

ZHANG e DONG [141] utilizam o termo regularizador conhecido como decaimento dos pesos, *weight decay*, conforme exposto no item 1.2.5.1.1, para previsão de carga a curto prazo.

Visto que o controle de complexidade dos modelos neurais constitui uma condição essencial para o desenvolvimento de modelos com boa capacidade de generalização, e que este tema ainda não mereceu a devida atenção por parte da comunidade científica, quando aplicado ao problema de previsão de carga a curto prazo, vide o escasso número de trabalhos encontrados na literatura, este trabalho apresentará a aplicação de algumas técnicas de regularização de modelos neurais para a previsão de carga a curto prazo. Mais especificamente, este trabalho investigará o treinamento *Bayesiano* e o escalonamento do ganho da função de ativação de MLP's e as máquinas de vetores suporte, aplicados à previsão de carga para o horizonte de curto prazo.

2

Regularização de redes neurais artificiais

De uma forma geral, as RNA's podem ser vistas como um processador de sinais paralelamente distribuído, constituído de unidades de processamento simples, conhecidas como neurônios, que adquirem conhecimento acerca de uma determinada tarefa através da integração com o ambiente via algum algoritmo de aprendizagem, com esse conhecimento sendo armazenado nos pesos sinápticos que interligam os diversos neurônios. A Figura 2.1 apresenta um diagrama esquemático de um neurônio, cuja saída c é dada pela equação (2.1):

$$c = \varphi\left(\sum_{i=1}^{n} \omega_i x_i + b\right) \tag{2.1}$$

Nesta equação, c representa a saída do neurônio, $\omega_i, i = 1, 2, ..., n$, o conjunto de pesos sinápticos que ligam as entradas $x_i, i = 1, 2, ..., n$, deste neurônio, que podem ser oriundas da saída de um outro neurônio ou da própria camada de entrada, b o *bias* associado a este neurônio e $\varphi(\cdot): \mathbb{R}^n \to \mathbb{R}$, a função de ativação do mesmo.

Mantendo a analogia biológica, os neurônios constituintes das RNA's são dispostos em camadas, e a forma com que estas camadas são interligadas define a arquitetura do modelo neural. Basicamente, existem duas estruturas, as redes alimentadas adiante, *feedforward*, e as redes recorrentes. Nas redes *feedforward*, como o próprio nome já diz, as camadas são conectadas consecutivamente, com cada camada ligada única e exclusivamente com as suas camadas adjacentes, e o sinal fluindo da camada de entrada para a camada de saída em sentido único, conforme ilustrado na Figura 2.2. As redes recorrentes apresentam um o mais laços de realimentação na estrutura apresentada na Figura 2.2.

Visto que a maioria das propostas de modelos neurais para previsão de carga a curto prazo utiliza modelos *feedforward*, conforme verificado na seção 1.2.5, este trabalho focará apenas neste tipo de estrutura.

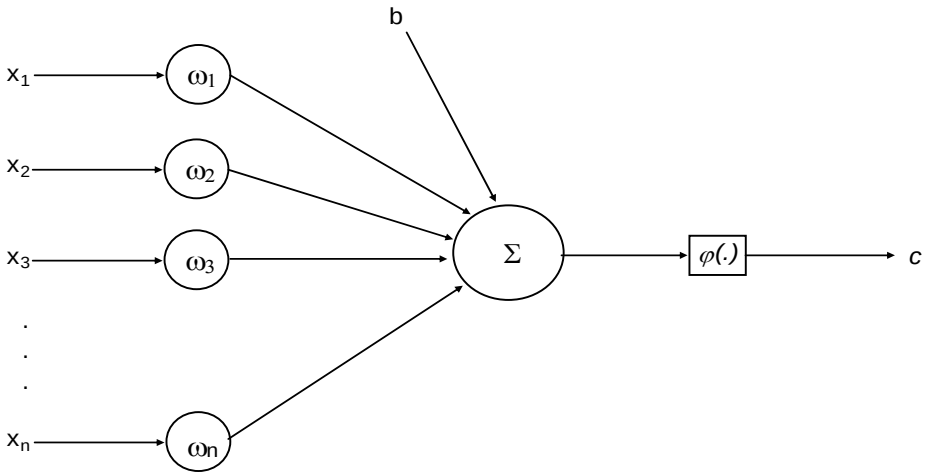

Figura 2.1 – Diagrama esquemático de um neurônio

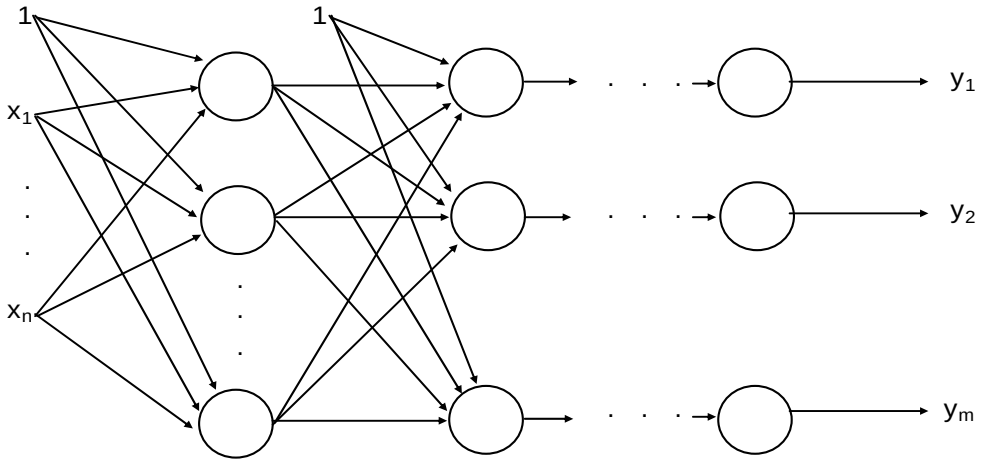

Figura 2.2 – Rede neural *feedforward* com múltiplas camadas

Dentre as principais vantagens destes modelos neurais, podem ser citadas as seguintes:

- Não-linearidade: para o caso mais comumente utilizado, em que a função de ativação dos neurônios da camada oculta é não-linear, o modelo neural resultante da interconexão destas unidades mais simples apresenta elevado grau

de não-linearidade, podendo abordar problemas de complexidade elevada. Entretanto, esta vantajosa característica dos modelos neurais pode ser prejudicial na presença de dados ruidosos (*overfitting*), motivo de discussão deste capítulo;

- Mapeamento entrada-saída: a partir de um conjunto de pares entrada-saída, os modelos neurais realizam um mapeamento entrada-saída deste conjunto, sem a necessidade de desenvolvimento de modelos matemáticos abordando a dinâmica do processo em estudo;
- Adaptabilidade: os modelos neurais apresentam elevada capacidade de adaptação em virtude de mudanças nas condições do ambiente para o qual a rede foi treinada para operar. Para tal, basta treinar a rede novamente, incluindo no conjunto de treinamento os padrões referentes às novas condições ambientais;
- Implementação simples: visto que os modelos *feedforward* podem ser vistos como grafos orientados, a implementação destas estruturas é extremamente simples, quando comparada com o grau de complexidade dos modelos que podem ser gerados.

Além das vantagens acima citadas, o teorema da aproximação universal [151] afirma que alguns modelos neurais *feedforward* podem aproximar com precisão arbitrária qualquer função contínua não-linear $f(\underline{x}):\mathbb{R}^n \to \mathbb{R}$. Para tal, é necessário que a estrutura apresente ao menos uma camada escondida contendo neurônios com função de ativação, $\varphi(z):\mathbb{R} \to \mathbb{R}$, contínua, não-constante, limitada, e uma saída linear, representando a aproximação da função $f(\underline{x})$ obtida pelo modelo. Portanto, modelos neurais *feedforward,* com uma única camada escondida contendo um número suficiente de neurônios com função de ativação com as características anteriormente citadas, podem aproximar qualquer função contínua não-linear.

Conforme evidenciado no item 1.2.5, várias propostas de modelos de previsão de carga a curto prazo baseados em redes neurais *feedforward* com uma única camada escondida podem ser encontradas na literatura, podendo ser citados o *perceptron* de múltiplas camadas (MLP), as redes de função de base radial (RBFN), *functional link network* (FLN), *window random activation weight neural network* (WRAWNN), *moving window regression trained random activation weight neural network* (MWRAWNN), dentre outros. Neste trabalho serão utilizados apenas modelos *feedforward* com uma única camada oculta, mais especificamente, o *perceptron* de múltiplas camadas (MLP) e as máquinas de vetor suporte (SVM's). Uma breve discussão sobre MLP's está apresentada no Apêndice A, com o desenvolvimento teórico das SVM's sendo descrito neste capítulo.

Apesar destes modelos possuírem a vantajosa característica de aproximação universal, o objetivo do desenvolvimento de uma máquina de aprendizagem não reside na representação exata do conjunto de dados disponíveis, mas sim na obtenção de um modelo estatístico do processo gerador de tais dados [158]. Logo, é desejado que o modelo desenvolvido apresente resultados satisfatórios tanto para os dados disponíveis quanto para novos dados a serem apresentados. Em outras palavras, o modelo desenvolvido deve apresentar boa capacidade de generalização.

Entretanto, a concepção original do algoritmo de retropropagação de erro para treinamento de MLP's, baseada no princípio da minimização do risco empírico, demonstra preocupação única e exclusiva com o ajuste dos dados de treinamento. Esta abordagem pode conduzir, na presença de dados ruidosos, a modelos com reduzida capacidade de generalização, em virtude da ocorrência de *overfitting*, ou, para o caso de modelos neurais, *overtraining*. Em virtude disto, a aplicação direta do princípio da minimização do risco empírico pode resultar em modelos com instabilidade numérica e

fraco desempenho de generalização, ou seja, modelos apresentando resultados insatisfatórios para novos padrões, diferentes daqueles utilizados para treinamento, porém provenientes da mesma população [161]. Por outro lado, todo o desenvolvimento teórico das SVM's encontra sustentação no princípio da minimização do risco estrutural, que busca a minimização do limite superior do erro de generalização. Portanto, as SVM's apresentam na sua essência a preocupação com a capacidade de generalização do modelo estimado.

Esta questão evidencia a necessidade do controle de complexidade, ou da regularização, de modelos neurais. Visto que a SVM, por utilizar o princípio de minimização do risco estrutural, já apresenta no seu desenvolvimento um termo responsável pela regularização do modelo, relacionado com a maximização da margem de separação ρ, as técnicas de regularização apresentadas nesta tese estarão limitadas ao MLP. Técnicas avançadas de regularização de SVM's podem ser encontradas em [161].

2.1 Máquina de vetor suporte (SVM)

As chamadas máquinas de vetor suporte (SVM's) foram desenvolvidas com base em um novo paradigma da área de aprendizado de máquina, conhecido como aprendizado estatístico. Diferentemente dos métodos estatísticos clássicos aplicados a problemas de classificação, que necessitam de uma quantidade elevada de dados em conjunto com a inserção de conhecimento prévio acerca do problema abordado, a teoria de aprendizado estatístico foi desenvolvida para solução de problemas de classificação cuja quantidade de dados disponíveis é reduzida e pouco, ou até mesmo nenhum, conhecimento prévio sobre o problema pode ser utilizado, características estas comumente encontradas em muitas aplicações reais [160].

A teoria de SVM foi originalmente elaborada para solução de problemas de classificação, através da aplicação do conceito de hiperplano ótimo de separação, baseado na maximização da margem de separação ρ. A Figura 2.3 ilustra a margem de separação ρ para o caso de padrões linearmente separáveis em duas classes. Nesta figura, a reta vermelha representa o hiperplano ótimo de separação, com os chamados vetores suporte sendo aqueles situados exatamente em cima das retas negras tracejadas. Estes vetores recebem esta denominação em virtude da sua proximidade da superfície de decisão, contribuindo de maneira importante para a definição de tal superfície [160].

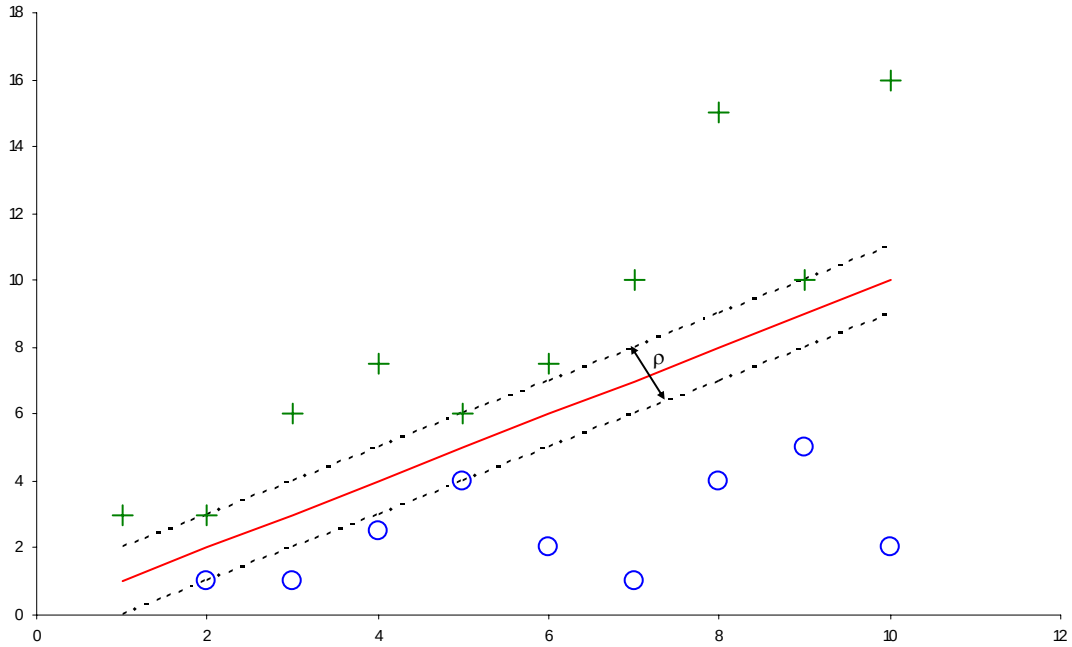

Figura 2.3 – Ilustração da margem de separação ρ para o caso de duas classes linearmente separáveis.

O conceito de hiperplano ótimo de separação, também conhecido como hiperplano de margem máxima de separação, pode ser expandido para problemas de classificação de padrões não-linearmente separáveis, através do mapeamento do espaço original de representação em um espaço de dimensão elevada, onde o problema passa a ser linearmente separável. Desta forma, as SVM's podem ser vistas como máquinas lineares aplicadas a um espaço de representação expandido, de dimensão maior que o

espaço de representação original do problema, com o mapeamento que governa esta expansão sendo obtido de maneira intrínseca, ou seja, de forma não-explícita. Seguindo esta idéia, matematicamente, a saída de uma SVM pode ser dada por:

$$f\left(\underline{x},\underline{W}\right)=\sum_{j=0}^{n_1} W_j\phi_j\left(\underline{x}\right)=\left[\underline{W}\right]^t\underline{\phi}\left(\underline{x}\right) \tag{2.2}$$

$$\underline{W}=\left[b,W_1,W_2,...,W_{n_1}\right]^t$$

$$\underline{\phi}\left(\underline{x}\right)=\left[1,\phi_1\left(\underline{x}\right),\phi_2\left(\underline{x}\right),...,\phi_{n_1}\left(\underline{x}\right)\right]^t$$

Na equação (2.2), $\underline{\phi}\left(\underline{x}\right):\mathbb{R}^n\rightarrow\mathbb{R}^{n_1}$ representa o mapeamento não-linear das entradas $\underline{x}$ do espaço original de representação $\mathbb{R}^n$ para um espaço de dimensionalidade elevada $\mathbb{R}^{n_1}$, $n_1>n$, com $\underline{W}\in\mathbb{R}^{n_1}$ representando o conjunto de parâmetros que define a máquina linear aplicada no espaço expandido. Em outras palavras, o objetivo do mapeamento não-linear $\underline{\phi}\left(\underline{x}\right)$ consiste na mudança do espaço de representação do problema, originalmente não-linearmente separável no espaço $\mathbb{R}^n$, para um espaço $\mathbb{R}^{n_1}$ onde o problema passa a ser linearmente separável, podendo ser aplicado neste novo espaço de representação o conceito de hiperplano ótimo de separação. Este novo espaço de representação também é conhecido como espaço de características [151], [160] e [161]. A Figura 2.4 apresenta um diagrama esquemático de uma SVM.

A idéia de mapear o espaço de representação original em um espaço de maior dimensão com o intuito de tornar o problema linearmente separável encontra justificativa no teorema de *Cover*. De uma maneira informal, este teorema afirma que se o mapeamento $\underline{\phi}\left(\underline{x}\right)$ for não-linear e a dimensionalidade n_1 do espaço de características for suficientemente elevada, a probabilidade do problema ser linearmente separável neste novo espaço de representação é elevada [151].

A teoria de SVM desenvolvida para problemas de classificação foi expandida para problemas mais gerais de reconhecimento de padrões, como problemas de

aproximação funcional, regressão e processamento de sinais, aumentando assim a aplicabilidade das SVM's. Visto que o problema abordado nesta tese pode ser enquadrado na classe de problemas de aproximação funcional, a apresentação da teoria das SVM's estará restrita a essa área de aplicação, podendo ser estendida para problemas de regressão, identificação de sistemas e processamento de sinais de maneira direta.

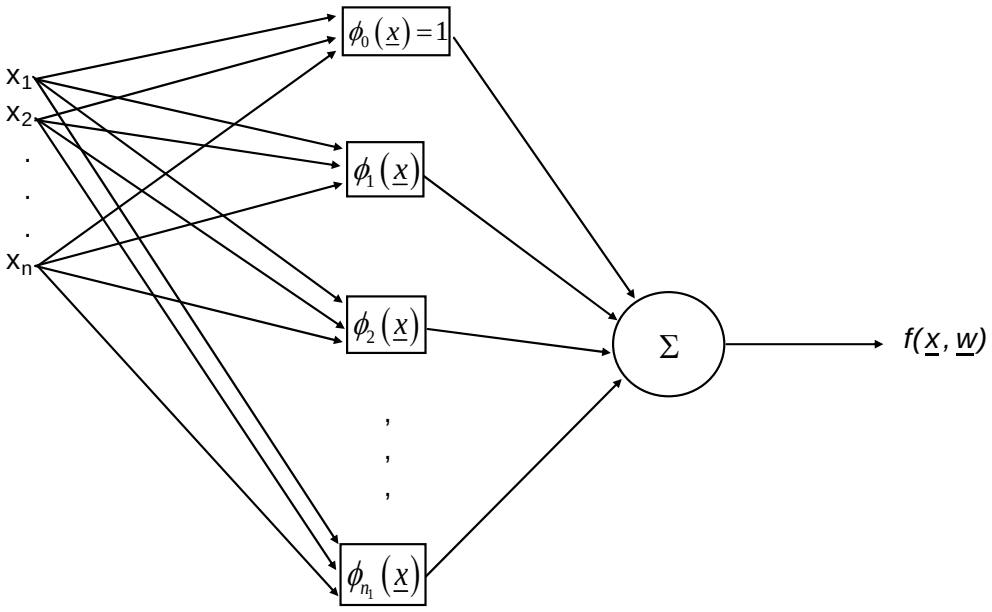

Figura 2.4 – Diagrama esquemático de uma SVM

Basicamente, problemas de aproximação funcional buscam a aproximação, ou interpolação, de uma função contínua $F(\underline{x}): \mathbb{R}^n \to \mathbb{R}$, por uma função $f(\underline{x}, \underline{w}): \mathbb{R}^n \to \mathbb{R}$, definida por um vetor de parâmetros $\underline{w} = [w_1, w_2, ..., w_M]^t$. Escolhido o tipo da função $f(\underline{x}, \underline{w})$, o problema reside na estimação do vetor $\underline{w}$ através da utilização de um conjunto de exemplos $D = \{\underline{x}_k, d_k\}$, $k = 1, 2, ..., N$, $\underline{x}_k = [x_{k1}, x_{k2}, ..., x_{kn}]^t$, e $d_k = F(\underline{x}_k)$, para problemas de interpolação, ou $d_k = F(\underline{x}_k) + \zeta_k$, para problemas de aproximação. Para problemas de aproximação,

categoria na qual os problemas de regressão podem ser inseridos, apresentando ruído aditivo ζ com distribuição normal de probabilidade, a minimização do erro médio quadrático resulta na melhor estimativa não-tendenciosa do vetor $\underline{w}$ [160]. Entretanto, a maioria dos problemas reais de regressão não apresenta ruído aditivo *gaussiano*, trazendo à tona a necessidade da utilização de outros tipos de função de erro. Em virtude disso, a SVM aplicada a problemas de regressão utiliza uma função de erro conhecida como função de perda com tolerância ε, dada pela seguinte equação:

$$L_\varepsilon\left[d, f\left(\underline{x},\underline{W}\right)\right] = \begin{cases} \left|d - f\left(\underline{x},\underline{W}\right)\right| - \varepsilon, & \left|d - f\left(\underline{x},\underline{W}\right)\right| \geq \varepsilon \\ 0, & \left|d - f\left(\underline{x},\underline{W}\right)\right| < \varepsilon \end{cases} \tag{2.3}$$

Na equação (2.3), $f\left(\underline{x},\underline{w}\right)$ representa a saída calculada pela SVM, d representa a saída desejada associada ao vetor $\underline{x}$, e ε é um parâmetro especificado pelo usuário. Para problemas de regressão com ruído aditivo *gaussiano*, ε pode representar a variância de tal ruído [162]. Outras funções de erro com tolerância ε podem ser utilizadas, como a função quadrática de perda com tolerância ε e a função de perda de *Huber*, dadas pelas equações (2.4) e (2.5), respectivamente :

$$L_\varepsilon\left[d, f\left(\underline{x},\underline{W}\right)\right] = \begin{cases} \left[\left|d - f\left(\underline{x},\underline{W}\right)\right| - \varepsilon\right]^2, & \left|d - f\left(\underline{x},\underline{W}\right)\right| \geq \varepsilon \\ 0, & \left|d - f\left(\underline{x},\underline{W}\right)\right| < \varepsilon \end{cases} \tag{2.4}$$

$$L_\varepsilon\left[d, f\left(\underline{x},\underline{W}\right)\right] = \begin{cases} \varepsilon\left|d - f\left(\underline{x},\underline{W}\right)\right| - \dfrac{\varepsilon^2}{2}, & \left|d - f\left(\underline{x},\underline{W}\right)\right| \geq \varepsilon \\ \dfrac{1}{2}\left[\left|d - f\left(\underline{x},\underline{W}\right)\right|\right]^2, & \left|d - f\left(\underline{x},\underline{W}\right)\right| < \varepsilon \end{cases} \tag{2.5}$$

As figuras Figura 2.5 a Figura 2.7 apresentam os gráficos das funções de perda com tolerância ε dadas pelas equações (2.3), (2.4) e (2.5), respectivamente.

O uso deste tipo de função de erro também encontra motivação nos problemas de classificação, para os quais as SVM's foram originalmente desenvolvidas. Nestes

problemas, existe uma vasta área do espaço de representação dos padrões cujo valor da função de erro é nulo, ou seja, nesta área os padrões são corretamente classificados. Em

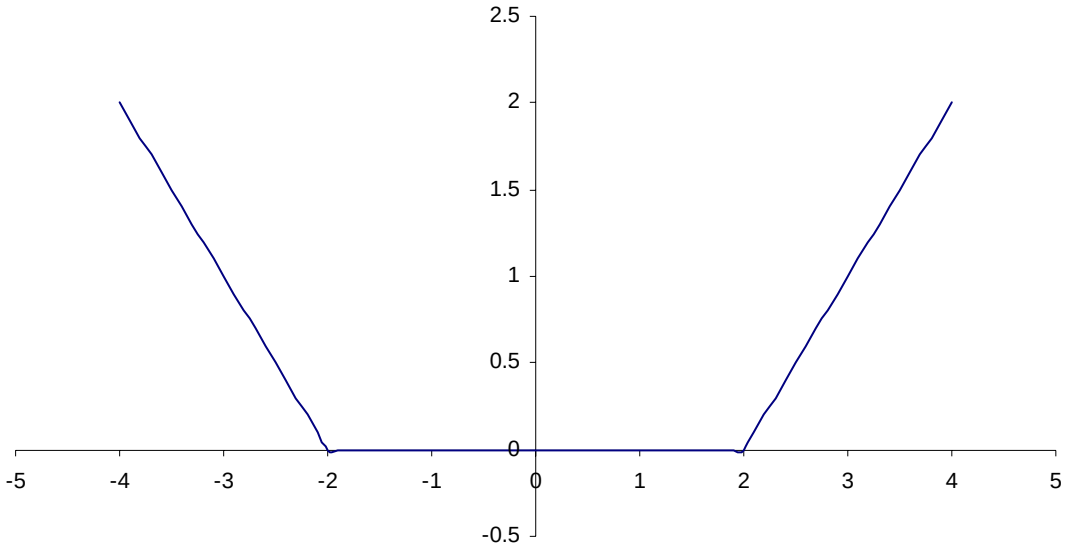

Figura 2.5 – Gráfico da função linear de perda dada pela equação (2.3), para $\varepsilon = 2$

outras palavras, só contribuem para o processo de otimização responsável pela determinação do hiperplano ótimo de separação, os padrões situados no interior da margem ilustrada na Figura 2.3, para o caso específico de padrões linearmente separáveis. Portanto, com o intuito de manter a analogia com o desenvolvimento das SVM's para problemas de classificação, é importante que a função de erro utilizada para problemas de aproximação funcional também apresente uma região cujo valor seja nulo, característica esta marcante das chamadas funções de perda com tolerância ε [161]. A Figura 2.8 ilustra esta característica deste tipo de função de perda, para o caso específico da função dada (2.3), também conhecida como função linear de perda com tolerância ε. Nesta figura, a linha verde representa a função de aproximação $f\left(\underline{x}, \underline{w}\right)$, com as linhas vermelhas determinando a margem, ou tolerância, da aproximação realizada por $f\left(\underline{x}, \underline{w}\right)$, igual a $f\left(\underline{x}, \underline{w}\right) \pm \varepsilon$. Desta forma, serão penalizados, ou seja, apresentarão valores não-nulos da função $L_\varepsilon\left[d, f\left(\underline{x}, \underline{w}\right)\right]$, apenas os pontos situados fora da banda

determinada por $f\left(\underline{x},\underline{w}\right)\pm\varepsilon$. Apesar de terem sido apresentados três tipos de funções de erro com tolerância ε, este trabalho focará apenas na função dada pela equação (2.3). Portanto, a teoria sobre SVM apresentada neste trabalho utilizará apenas a função linear de perda com tolerância ε. O desenvolvimento da teoria de SVM utilizando as funções de perda dadas pelas equações (2.4) e (2.5) pode ser encontrado em [160] e [161].

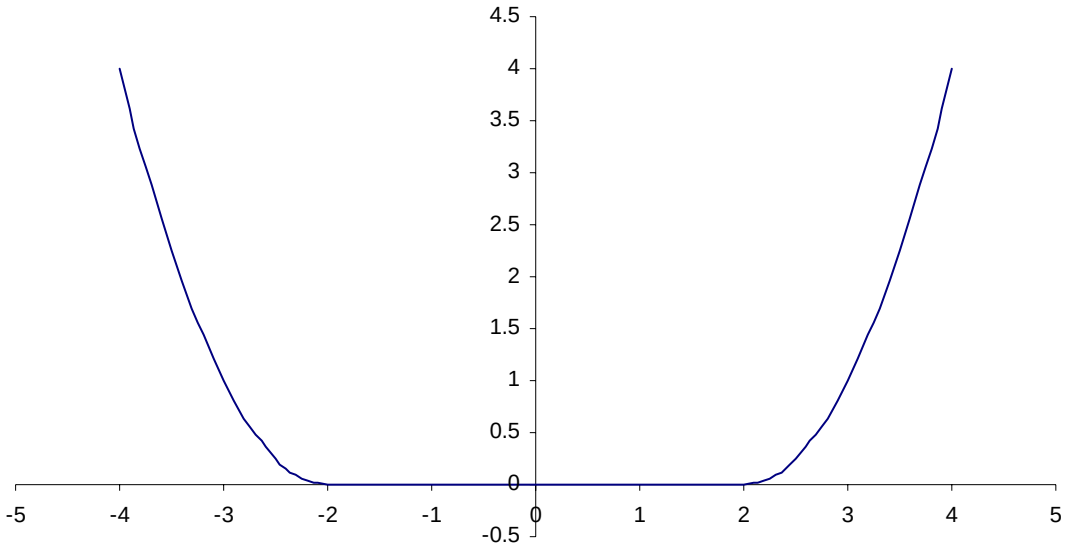

Figura 2.6 – Gráfico da função quadrática de perda dada pela equação (2.4), para $\varepsilon = 2$

Dada a função linear de perda com tolerância ε, o objetivo do treinamento de uma SVM para problemas de aproximação funcional reside na solução do problema de minimização restrito do risco empírico $E_s\left(\underline{w}\right)$ dado pela equação:

$$\min_{\underline{W}} E_s\left(\underline{W}\right) = \frac{1}{N}\sum_{i=1}^{N} L_\varepsilon\left[d_k, f\left(\underline{x}_k,\underline{W}\right)\right] \tag{2.6}$$

s.a.

$$\left\|\underline{W}\right\|^2 \le c_0$$

A restrição do problema de minimização descrito na equação (2.6) tem origem na maximização da margem de separação ρ para problemas de classificação, com c_0 sendo uma constante. Para problemas de aproximação funcional, esta restrição está

diretamente relacionada com o controle de complexidade do modelo estimado, objetivando a maximização da capacidade de generalização de tal modelo. Desta forma,

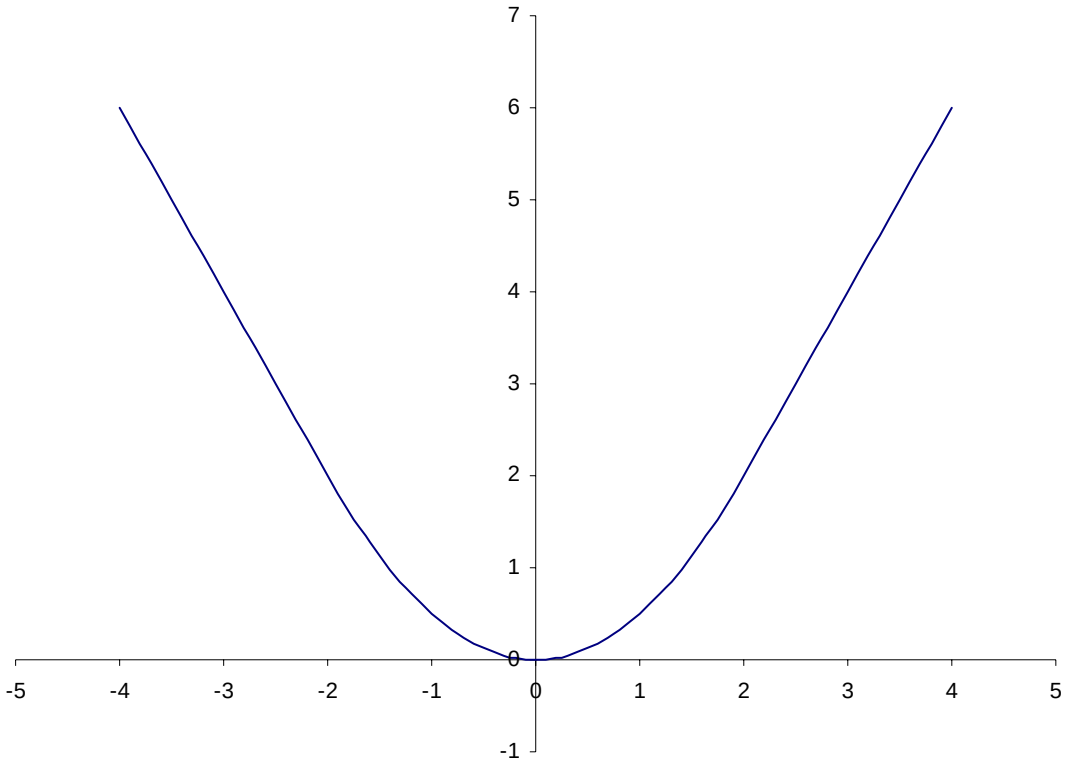

Figura 2.7 – Gráfico da função de perda de *Huber* dada pela equação (2.5), para $\varepsilon = 2$

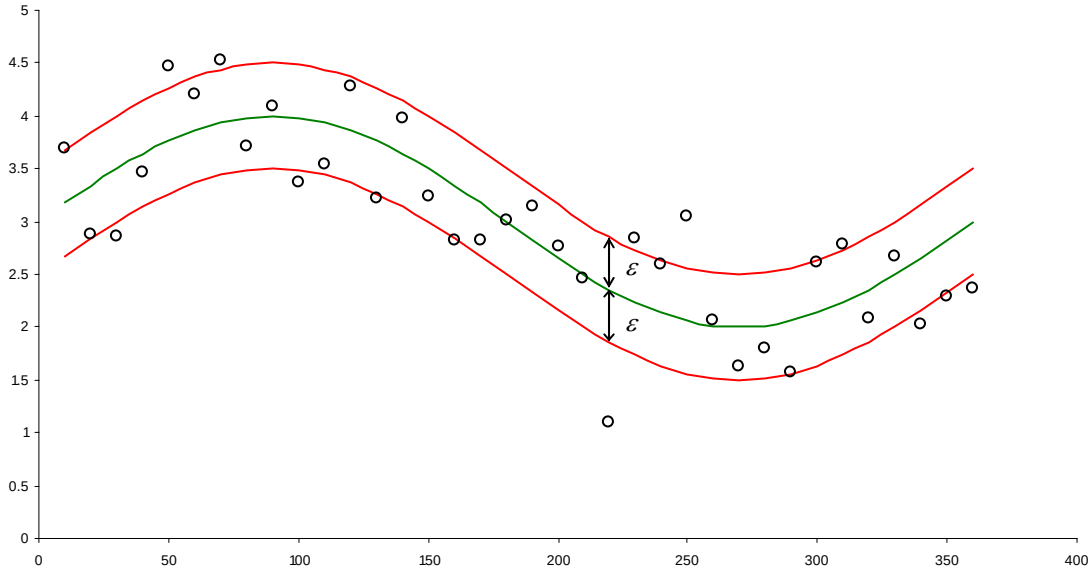

Figura 2.8 – Ilustração do papel do parâmetro ε

ao contrário do treinamento de MLP's, o qual é baseado na minimização do risco empírico, o treinamento de SVM's busca a minimização do risco estrutural, cujo objetivo não reside na minimização única e exclusiva do erro para o conjunto de treinamento, e sim na minimização do limite superior do erro para um conjunto independente de dados não utilizados para treinamento do modelo [160] e [161].

Conforme apresentado na Figura 2.5, a função linear de perda com tolerância ε não é continuamente diferenciável. Esta característica indesejada desta função de perda pode ser abordada através da introdução de dois conjuntos de variáveis de folga, $\underline{\xi}_1 = \left[\xi_{11}, \xi_{12}, ..., \xi_{1N}\right]^t$ e $\underline{\xi}_2 = \left[\xi_{21}, \xi_{22}, ..., \xi_{2N}\right]^t$, definidas através da seguinte equação:

$$
\begin{aligned}
& d_k - f\left(\underline{x}_k, \underline{W}\right) = d_k - \left[\underline{W}\right]^t \underline{\phi}\left(\underline{x}_k\right) \leq \varepsilon + \xi_{1k} \\
& -\left[d_k - f\left(\underline{x}_k, \underline{W}\right)\right] = \left[\underline{W}\right]^t \underline{\phi}\left(\underline{x}_k\right) - d_k \leq \varepsilon + \xi_{2k} \\
& \xi_{1k} \geq 0 \\
& \xi_{2k} \geq 0 \\
& k = 1, 2, ..., N
\end{aligned}
\tag{2.7}
$$

Desta forma, o problema de minimização descrito na equação (2.6) é equivalente ao seguinte problema de minimização restrito [160]:

$$
\begin{aligned}
& \min_{\underline{W}} \mathrm{A}\left(\underline{\xi}_1, \underline{\xi}_2\right) = \sum_{k=1}^{N} \xi_{1k} + \sum_{k=1}^{N} \xi_{2k} \\
& s.a \\
& d_k - \left[\underline{W}\right]^t \underline{\phi}\left(\underline{x}_k\right) \leq \varepsilon + \xi_{1k} \\
& \left[\underline{W}\right]^t \underline{\phi}\left(\underline{x}_k\right) - d_k \leq \varepsilon + \xi_{2k} \\
& \xi_{1k} \geq 0 \\
& \xi_{2k} \geq 0 \\
& \left\|\underline{W}\right\|^2 \leq c_0 \\
& k = 1, 2, ..., N
\end{aligned}
\tag{2.8}
$$

Este problema de otimização apresenta uma restrição não-linear, $\left\|\underline{W}\right\|^2 \leq c_0$, impossibilitando a aplicação de técnicas analíticas de otimização, visto que foram desenvolvidas para problemas de otimização com restrições lineares. Entretanto, esta

restrição não-linear pode ser abordada diretamente na função objetivo de um problema de otimização quadrática, dado por:

$$\min \Phi\left(\underline{W}, \underline{\xi}_1, \underline{\xi}_2\right) = C\left[\sum_{k=1}^{N} \xi_{1k} + \sum_{k=1}^{N} \xi_{2k}\right] + \frac{1}{2}\left[\underline{W}\right]^t \underline{W} \tag{2.9}$$

s.a

$$d_k - \left[\underline{W}\right]^t \underline{\phi}\left(\underline{x}_k\right) \leq \varepsilon + \xi_{1k}$$

$$\left[\underline{W}\right]^t \underline{\phi}\left(\underline{x}_k\right) - d_k \leq \varepsilon + \xi_{2k}$$

$$\xi_{1k} \geq 0$$

$$\xi_{2k} \geq 0$$

$$k = 1, 2, ..., N$$

Neste novo problema de minimização restrito, C é uma constante, responsável pelo equilíbrio entre o ajuste dos dados de treinamento e a complexidade do modelo, seguindo o princípio de minimização do risco estrutural.

Portanto, visto que a descontinuidade da função linear de perda com tolerância ε foi tratada através da inserção das variáveis de folga $\underline{\xi}_1$ e $\underline{\xi}_2$, e a restrição não-linear $\left\|\underline{W}\right\|^2 \leq c_0$ foi abordada diretamente na função objetivo do problema descrito na equação (2.9), técnicas analíticas de otimização podem ser utilizadas para solução deste problema, como por exemplo a regra dos multiplicadores de *Lagrange*. Para tal, seja o funcional *Lagrangeano* $L\left(\underline{w}, \underline{\xi}_1, \underline{\xi}_2, \underline{\alpha}_1, \underline{\alpha}_2, \underline{\gamma}_1, \underline{\gamma}_2\right)$ relacionado ao problema descrito na equação (2.9), dado por:

$$L\left(\underline{w}, \underline{\xi}_1, \underline{\xi}_2, \underline{\alpha}_1, \underline{\alpha}_2, \underline{\gamma}_1, \underline{\gamma}_2\right) = C\left[\sum_{k=1}^{N} \xi_{1k} + \sum_{k=1}^{N} \xi_{2k}\right] + \frac{1}{2}\left[\underline{W}\right]^t \underline{W} - \sum_{k=1}^{N}\left(\gamma_{1k}\xi_{1k} + \gamma_{2k}\xi_{2k}\right) \tag{2.10}$$

$$-\sum_{k=1}^{N} \alpha_{1k}\left\{\left[\underline{W}\right]^t \underline{\phi}\left(\underline{x}_k\right) - d_k + \varepsilon + \xi_{2k}\right\} - \sum_{k=1}^{N} \alpha_{2k}\left\{d_k - \left[\underline{W}\right]^t \underline{\phi}\left(\underline{x}_k\right) + \varepsilon + \xi_{1k}\right\}$$

$$\underline{\alpha}_1 = \left[\alpha_{11}, \alpha_{12}, ..., \alpha_{1N}\right]^t$$
$$\underline{\alpha}_2 = \left[\alpha_{21}, \alpha_{22}, ..., \alpha_{2N}\right]^t$$
$$\underline{\gamma}_1 = \left[\gamma_{11}, \gamma_{12}, ..., \gamma_{1N}\right]^t$$
$$\underline{\gamma}_2 = \left[\gamma_{21}, \gamma_{22}, ..., \gamma_{2N}\right]^t$$

Na equação (2.10), $\underline{\alpha}_1$, $\underline{\alpha}_2$, $\underline{\gamma}_1$ e $\underline{\gamma}_2$ representam os vetores contendo os multiplicadores de *Lagrange*.

A solução do problema de otimização descrito na equação (2.9) pode ser obtida através da obtenção do ponto de sela do funcional *Lagrangeano* dado pela equação (2.10) [160]. Este ponto é determinado através da minimização, em relação ao vetor de parâmetros $\underline{W}$ e às variáveis de folga $\underline{\xi}_1$ e $\underline{\xi}_2$, de $L\left(\underline{w}, \underline{\xi}_1, \underline{\xi}_2, \underline{\alpha}_1, \underline{\alpha}_2, \underline{\gamma}_1, \underline{\gamma}_2\right)$, e a posterior maximização deste mesmo funcional com relação aos multiplicadores de *Lagrange* $\underline{\alpha}_1$, $\underline{\alpha}_2$, $\underline{\gamma}_1$ e $\underline{\gamma}_2$. O problema de minimização a ser resolvido também é conhecido como problema primal, com o posterior problema de maximização sendo chamado de problema dual.

Utilizando as condições de otimalidade do cálculo, ou seja, $\nabla L\left(\underline{w}, \underline{\xi}_1, \underline{\xi}_2, \underline{\alpha}_1, \underline{\alpha}_2, \underline{\gamma}_1, \underline{\gamma}_2\right) = \underline{0}$, a minimização do funcional *Lagrangeano* em relação a $\underline{W}$, $\underline{\xi}_1$ e $\underline{\xi}_2$ resulta nas seguintes expressões:

$$\underline{W} = \sum_{k=1}^{N} \left(\alpha_{1k} - \alpha_{2k}\right) \underline{\phi}\left(\underline{x}_k\right) \tag{2.11}$$
$$\gamma_{1k} = C - \alpha_{1k}$$
$$\gamma_{2k} = C - \alpha_{2k}$$
$$k = 1, 2, ..., N$$

Portanto, as equações (2.11) constituem a solução do problema primal. Substituindo este resultado na equação (2.10), é obtido o problema dual de maximização, dado por:

$$\max \Psi\left(\underline{\alpha}_1, \underline{\alpha}_2\right) = \sum_{k=1}^{N} d_k \left(\alpha_{1k} - \alpha_{2k}\right) - \varepsilon \sum_{k=1}^{N} \left(\alpha_{1k} + \alpha_{2k}\right) \tag{2.12}$$

$$+ \frac{1}{2} \sum_{k=1}^{N} \sum_{j=1}^{N} \left(\alpha_{1k} - \alpha_{2k}\right)\left(\alpha_{1j} - \alpha_{2j}\right) K\left(\underline{x}_k, \underline{x}_j\right)$$

s.a

$$\sum_{k=1}^{N} \left(\alpha_{1k} - \alpha_{2k}\right) = 0$$

$$0 < \alpha_{1k} < C$$

$$0 < \alpha_{2k} < C$$

$$k = 1, 2, ..., N$$

No problema descrito na equação (2.12), $K\left(\underline{x}_k, \underline{x}_j\right)$ é o chamado núcleo do produto interno, dado por:

$$K\left(\underline{x}_k, \underline{x}_j\right) = \left[\underline{\phi}\left(\underline{x}_k\right)\right]^t \underline{\phi}\left(\underline{x}_j\right) \tag{2.13}$$

O núcleo do produto interno $K\left(\underline{x}_k, \underline{x}_j\right)$, também conhecido como *kernel*, o qual deve ser definido segundo o teorema de *Mercer* [160], [161] e [162]. Dentre alguns exemplos de *kernels* $K\left(\underline{x}_k, \underline{x}_j\right)$, podem ser citadas as funções polinomiais, *gaussianas* e sigmoidais (com restrições), dadas pelas seguintes equações:

$$K\left(\underline{x}_k, \underline{x}_j\right) = \left\{\left[\underline{x}_k\right]^t \underline{x}_j + 1\right\}^p \tag{2.14}$$

$$K\left(\underline{x}_k, \underline{x}_j\right) = e^{-\frac{1}{2\sigma^2}\left\|\underline{x}_k - \underline{x}_j\right\|^2} \tag{2.15}$$

$$K\left(\underline{x}_k, \underline{x}_j\right) = \tanh\left\{\beta_0 \left[\underline{x}_k\right]^t \underline{x}_j + \beta_1\right\} \tag{2.16}$$

Para os *kernels* descritos acima, p e σ são parâmetros definidos *a priori*. Vale ressaltar que, para as funções sigmoidais, as condições de *Mercer* são satisfeitas apenas para valores de $\beta_0 > 0$ e $\beta_1 < 0$ [161]. Portanto, os MLP's e as redes de função de base radial podem ser vistas também como uma SVM com um tipo específico de *kernel.*

Portanto, utilizando a definição de *kernel* $K\left(\underline{x}_k, \underline{x}_j\right)$, e substituindo a equação (2.11) na equação (2.2), a saída de uma SVM passa a ser dada por:

$$f\left(\underline{x}, \underline{W}\right) = \sum_{k=1}^{N} \left(\alpha_{1k} - \alpha_{2k}\right) K\left(\underline{x}, \underline{x}_k\right) \tag{2.17}$$

Para problemas de classificação, a definição do hiperplano ótimo de separação, que em última análise está intimamente ligada com a saída de uma SVM para problemas de classificação, depende exclusivamente da posição dos vetores suporte. Mantendo analogia com estes problemas, a solução do problema de maximização descrito na equação (2.12) apresenta $\alpha_{1k} \neq \alpha_{2k}$ apenas para alguns vetores $\underline{x}_k$ integrantes do conjunto $D = \left\{\underline{x}_k, d_k\right\}$. Estes vetores são os chamados vetores suporte. Conceitualmente, estes vetores estão situados fora da banda definida por ε na Figura 2.8.

A equação (2.17) evidencia a forma intrínseca em que é realizado o mapeamento do espaço de representação original no espaço de características, visto que a função $\underline{\phi}\left(\underline{x}\right)$ que define este mapeamento não precisa ser especificada, apenas o *kernel* $K\left(\underline{x}, \underline{x}_k\right)$. Esta equação também mostra que as SVM podem ser entendidas como modelos *feedforward* com uma única camada escondida, contendo neurônios cuja função de ativação é definida pelo *kernel* $K\left(\underline{x}, \underline{x}_k\right)$. A Figura 2.4 ilustra de maneira esquemática a estrutura de uma SVM, com NS representando o número de vetores suporte.

Apesar da necessidade de especificação das constantes C e ε e dos parâmetros do *kernel* $K\left(\underline{x}, \underline{x}_k\right)$, o desenvolvimento das SVM's conduz a uma metodologia que, de uma certa maneira, une a escolha da estrutura e o treinamento de modelos neurais *feedforward*, visto que o número de neurônios na camada oculta surge como subproduto

do algoritmo de treinamento, através dos vetores suporte obtidos. Além deste fato, ao utilizar o princípio da minimização do risco estrutural, o treinamento de SVM's inclui na sua formulação uma parcela responsável pelo controle de complexidade do modelo, objetivando a estimação de modelos com considerável capacidade de generalização.

A questão da regularização não é abordada no desenvolvimento do algoritmo clássico de retropropagação de erro para MLP's, o qual é baseado no princípio da minimização do risco empírico. Em virtude disso, para estimação, através do algoritmo de retropropagação de erro, de MLP's com boa capacidade de generalização, é necessária a aplicação de técnicas de controle de complexidade destes modelos, com algumas delas sendo apresentadas a seguir.

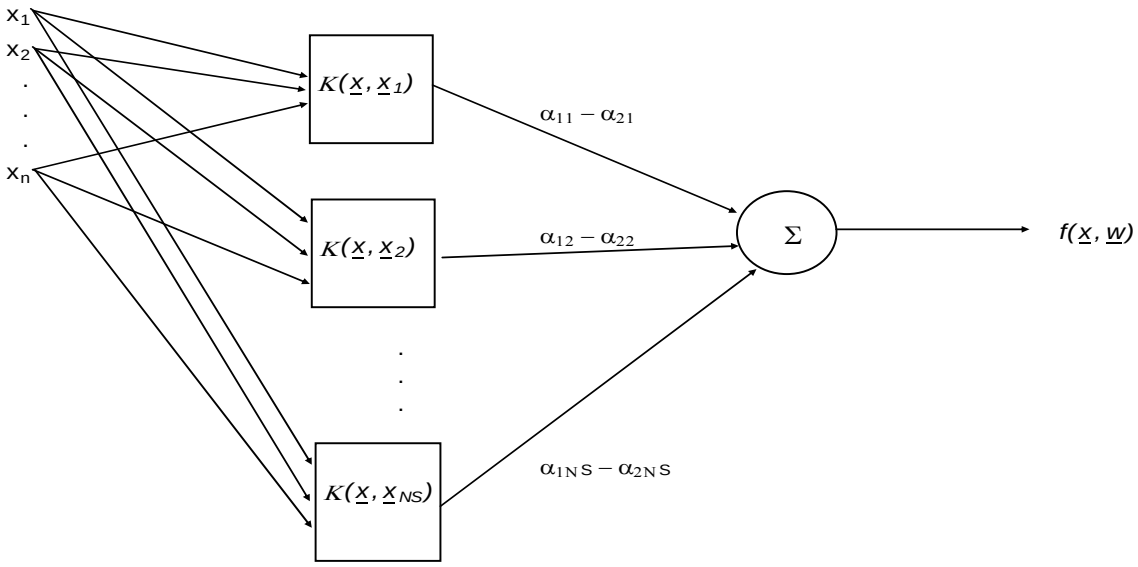

Figura 2.9 – Diagrama esquemático de uma SVM

2.2 Regularização em MLP's

A regularização de MLP's pode ser realizada através de dois procedimentos gerais. O primeiro, conhecido como estabilização de estrutura, está baseado na determinação da estrutura suficiente para o problema em questão. O outro procedimento tem por base a teoria da regularização, desenvolvida por *Tikhonov* para solução de

problemas mal-formulados. Além destes dois procedimentos, existem algumas heurísticas que podem ser aplicadas com o intuito de melhorar a capacidade de generalização de MLP's treinados através da aplicação do princípio de minimização do risco empírico. Estas três metodologias gerais são apresentadas ao longo desta seção.

2.2.1 Estabilização de estrutura

Problemas de aproximação funcional buscam a aproximação, ou interpolação, de uma função contínua $F(\underline{x}):\mathbb{R}^n \to \mathbb{R}$, por uma função $f(\underline{x},\underline{w}):\mathbb{R}^n \to \mathbb{R}$, definida por um vetor de parâmetros $\underline{w} = [w_1, w_2, ..., w_M]^t$. Uma das principais dificuldades para a obtenção da solução deste tipo de problema reside na escolha adequada da função de aproximação $f(\underline{x},\underline{w})$ a ser utilizada. Porém, o teorema da aproximação universal demonstra que MLP's com uma única camada escondida contendo número suficiente de neurônios podem aproximar com precisão arbitrária qualquer função contínua não-linear [151]. Portanto, um dos principais desafios encontrados na aplicação de MLP's a problemas de aproximação funcional reside na determinação do número suficiente de neurônios na camada oculta. Este processo também é conhecido como estabilização de estrutura, podendo ser aplicado de três formas.

A primeira forma de estabilização de estrutura consiste na comparação entre diversos modelos, com quantidades diferentes de neurônios na camada intermediária, escolhendo a estrutura através da análise do desempenho para um conjunto independente de dados, utilizando algumas técnicas de re-amostragem, ou utilizando índices analíticos de qualificação de modelos.

A segunda metodologia parte de um modelo demasiadamente complexo, ou seja, contendo um número elevado de neurônios na camada oculta, sendo aplicado a este

modelo alguns algoritmos de poda de rede, com o intuito de extirpar os neurônios em excesso, chegando ao número suficiente de neurônios para aplicação em estudo.

O terceiro e último procedimento de estabilização de estrutura pode ser considerado como o antípoda da segunda metodologia. Em outras palavras, a idéia consiste em começar com um modelo extremamente simples, por exemplo, com a saída sendo obtida através da combinação linear das entradas, sendo adicionados neurônios à camada oculta ao longo do processo, objetivando a obtenção do número suficiente de neurônios na camada intermediária. Procedimentos baseados nesta metodologia são conhecidos como métodos construtivos.

Estes três procedimentos serão detalhados nos próximos itens.

2.2.1.1 Técnicas de re-amostragem

Informalmente, as técnicas de re-amostragem podem ser entendidas como a partição do conjunto de dados disponível em três subconjuntos distintos, um para treinamento, outro para validação e o último para teste do modelo. O primeiro subconjunto é utilizado para estimação dos parâmetros que definem o modelo, com o subconjunto de validação sendo utilizado para validar o desempenho do modelo para um conjunto de dados ainda não apresentados ao mesmo. Visto que a escolha do modelo, no caso de MLP's o número de neurônios na camada oculta, tomando por base o desempenho para o conjunto de validação pode acarretar no ajuste excessivo dos padrões presentes no conjunto de validação, o conjunto de teste é utilizado para o teste efetivo do desempenho do modelo. Este procedimento é conhecido como validação única [165], ou método *hold-out* [158]. Vale ressaltar que a estimativa dos parâmetros deve ser realizada com a utilização apenas do conjunto de treinamento, a escolha de uma determinada estrutura deve ser efetuada através da análise do desempenho para o conjunto de validação, e o teste final do modelo realizado para o conjunto de teste. A

utilização de um padrão do conjunto de validação para treinamento do modelo, por exemplo, invalida o método.

Entretanto, na prática a quantidade de dados disponíveis para estimação do modelo pode ser limitada, impossibilitando a separação de alguns dados para avaliação do desempenho do modelo. Nestes casos, pode ser aplicado o procedimento popularmente conhecido como validação cruzada [158] e [165]. Esta técnica de re-amostragem utiliza a partição aleatória do conjunto de treinamento em δ partes. Destes subconjuntos, $(\delta - 1)$ são utilizados para treinamento, com o desempenho do modelo sendo avaliado para o subconjunto restante. Este processo é repetido δ vezes, até que todos os δ subconjuntos tenham sido utilizados para validação do modelo. O desempenho do modelo é então obtido através da média aritmética dos desempenhos obtidos para cada um dos δ subconjuntos utilizados para validação. O caso específico em que $\delta = N$, N igual ao número de padrões para treinamento, é conhecido como método *leave-one-out* [158] e [165].

Os procedimentos de re-amostragem descritos anteriormente apresentam uma série de desvantagens. A primeira diz respeito ao tempo de processamento, visto que, para o método *leave-one-out* por exemplo, são necessários N treinamentos de MLP's para obtenção do desempenho de um determinado modelo. Outra desvantagem está relacionada à seleção do conjunto de validação utilizado para validação única. Visto que o modelo escolhido é aquele que apresentar melhor desempenho para o conjunto de validação, esta escolha é extremamente dependente da definição de tal conjunto. Portanto, se o conjunto de validação utilizado não for representativo do problema em estudo, o modelo escolhido não apresentará desempenho satisfatório para o conjunto de teste, comprometendo assim a capacidade de generalização do modelo estimado. Entretanto, a principal desvantagem dos métodos de re-amostragem descritos nesta tese

tem lugar na ausência de um desenvolvimento formal, teórico, comprovando matematicamente a eficiência de tais métodos. A aplicação bem sucedida destas técnicas a problemas reais sugere que, mesmos sem o conhecimento formal dos pressupostos admitidos por estes métodos, estes são verificados em muitas aplicações reais. Porém, enquanto estes pressupostos não forem definidos, não existirá nenhuma garantia destas técnicas serem aplicadas com eficiências a novos problemas [165]. Propostas para abordagem de algumas das restrições citadas acima podem ser encontradas em [166] e [167].

2.2.1.2 Métodos analíticos de qualificação de modelos

As técnicas de re-amostragem descritas no item anterior buscam a estimativa da capacidade de generalização do modelo através da utilização de um conjunto independente de dados. Entretanto, a capacidade de generalização de máquinas de aprendizagem pode também ser estimada através de índices analíticos, sem a necessidade do uso de um conjunto de dados dedicado para avaliação do desempenho. Alguns destes índices serão apresentados a seguir.

2.2.1.2.1 AIC

O AIC, ou critério de informação de *Akaike* (*Akaike's Information Criterion)*, foi desenvolvido com o intuito de fornecer uma estimativa do erro de generalização de modelos lineares, sem a necessidade do uso de um conjunto de validação. A estimativa do erro de generalização $E_G\left[f\left(\underline{x},\underline{w}\right)\right]$ do modelo $f\left(\underline{x},\underline{w}\right)$, utilizando o AIC, é dada por [165]:

$$E_{G_{AIC}}\left[f\left(\underline{x},\underline{w}\right)\right] = E_s\left[f\left(\underline{x},\underline{w}\right)\right] + \frac{2\hat{P}}{N}\hat{\sigma}^2 \tag{2.18}$$

Na equação (2.18), $E_s\left[\, f\left(\underline{x},\underline{w}\right)\right]$ representa o risco empírico, ou erro para o conjunto de treinamento, do modelo $f\left(\underline{x},\underline{w}\right)$, N o número de padrões utilizados para estimação do modelo, $\hat{\sigma}^2$ a estimativa da variância do ruído existentes na saída e $\hat{P}$ o número efetivo de parâmetros do modelo. O conceito de número efetivo de parâmetros $\hat{P}$ será apresentado junto com a teoria da regularização, motivo de discussão do item 2.2.2, à qual este conceito esta intimamente ligado.

A estimativa da variância $\hat{\sigma}^2$ pode ser obtida através da seguinte expressão:

$$\hat{\sigma}^2 = \frac{N}{N-\hat{P}} E_s\left[\, f\left(\underline{x},\underline{w}\right)\right] \tag{2.19}$$

Substituindo esta estimativa na equação (2.18), é obtido o erro final de previsão de *Akaike*, *Final Prediction Error* (FPE), dado por [165]:

$$E_{G_{FPE}}\left[\, f\left(\underline{x},\underline{w}\right)\right] = \frac{N+\hat{P}}{N-\hat{P}} E_s\left[\, f\left(\underline{x},\underline{w}\right)\right] \tag{2.20}$$

Para modelos estimados através da utilização da teoria da regularização, pode ser obtido o erro final de previsão de modelos regularizados, *Final Prediction Error for a Regularized Cost* (FPER), dado pela equação [165]:

$$E_{G_{FPE}}\left[\, f\left(\underline{x},\underline{w}\right)\right] = \frac{N+\widehat{P_2}}{N-2\widehat{P_1}+\widehat{P_2}} E_s\left[\, f\left(\underline{x},\underline{w}\right)\right] \tag{2.21}$$

Nesta equação, $\widehat{P_1}$ e $\widehat{P_2}$ são quantidades relacionadas com o número efetivo de parâmetros, assim como $\hat{P}$.

2.2.1.2.2 BIC

O critério de informação *bayesiano*, *Bayesian Information Criterion* (BIC), assim como o AIC, busca uma estimativa analítica do erro de generalização, dispensando o uso de um conjunto de validação. Entretanto, ao contrário do AIC, que é

obtido tomando por base a média do erro de generalização, o BIC é determinado através da utilização de técnicas de estimação *bayesianas*. De uma forma geral, o erro de generalização do modelo $f\left(\underline{x},\underline{w}\right)$, utilizando o BIC, pode ser estimado por:

$$E_{G_{BIC}}\left[\, f\left(\underline{x},\underline{w}\right)\right] = E_s\left[\, f\left(\underline{x},\underline{w}\right)\right] + \frac{\widehat{P}\ln N}{N}\widehat{\sigma}^2 \tag{2.22}$$

2.2.1.2.3 NIC

A principal limitação da aplicação, em modelos neurais, dos métodos analíticos descritos anteriormente reside nas hipóteses consideradas ao longo do desenvolvimento de tais estimativas. De uma maneira geral, tanto o AIC quanto o BIC utilizam os seguintes pressupostos [165]:

- $d_k = F\left(\underline{x}_k\right) + \zeta_k$, com ζ_k representando um ruído branco estacionário, com média zero e variância σ^2, independente da entrada $\underline{x}_k$;
- O modelo $f\left(\underline{x},\underline{w}\right)$ é completo, ou seja, existe um vetor de parâmetros $\underline{w}^*$ tal que $f\left(\underline{x},\underline{w}^*\right) = F\left(\underline{w}\right)$;

A solução obtida através da minimização do risco empírico está localizada na vizinhança do ponto $\underline{w}^*$, fazendo com que a expansão em séries de *Taylor* do risco empírico e do erro de generalização sejam válidas.

Conforme descrito no item 2.1, a hipótese de ruído branco aditivo nas saídas não é verificada para a grande maioria das aplicações reais. Além disso, apesar dos modelos neurais serem considerados aproximadores universais, não existe garantia que estes modelos sejam completos, mesmo com um número suficiente de neurônios na camada intermediária, visto que as verdadeiras entradas $\underline{x}$ utilizadas pela função a ser aproximada $F\left(\underline{x}\right)$ são desconhecidas. Este argumento também invalida o terceiro

pressuposto, que constitui uma restrição grave para aplicação destes índices a MLP's, visto que o funcional de risco empírico utilizado para treinamento destes modelos apresenta diversos mínimos locais, comprometendo a afirmativa que a solução obtida pelo algoritmo de treinamento está localizada nas vizinhanças do mínimo global do funcional utilizado.

Com o intuito de superar algumas destas deficiências, MURATA *et. al.* [168] desenvolveram o chamado critério de informação de rede, *Network Information Criterion* (NIC), que em linhas gerais pode ser considerado como uma generalização do AIC, para casos em que o modelo não é completo, ou seja, para os casos em que não existe o vetor $\underline{w}^*$ tal que $f\left(\underline{x}, \underline{w}^*\right) = F\left(\underline{w}\right)$. Entretanto, este índice só pode ser aplicado a um conjunto hierárquico de modelos, ou seja, um conjunto de modelos onde os modelos de complexidade inferior, por exemplo, com menor número de parâmetros, possam ser considerados submodelos daqueles com complexidade maior. Exemplificando, só podem ser aplicados a MLP's com o mesmo número de camadas escondidas, porém com número de neurônios diferente em cada camada, não podendo ser utilizando para comparação entre MLP's e RBF's, por exemplo.

Apesar de buscar uma generalização do AIC para aplicação aos MLP's, o uso do índice NIC possui diversas restrições, visto que, assim como os outros índices, este índice apresenta uma série de características assintóticas que não são verificadas na prática. Além disso, o NIC requer uma estimativa da distribuição de probabilidade condicional $q\left(\underline{x}, d\right)$ geradora dos dados disponíveis para treinamento. Tal estimativa não é obtida de forma trivial, podendo ser calculada através de técnicas não-paramétricas, como janelas de *Parzen*. Porém, esta estimativa representará mais uma fonte de erro do processo de estimação, ou neste caso, de qualificação, das diversas máquinas de aprendizagem obtidas, comprometendo a aplicabilidade do índice.

A derivação do índice NIC é extensa, e, visto que o mesmo não será utilizado nesta tese, tal desenvolvimento não será apresentado. Maiores detalhes sobre este índice de qualificação de modelos podem ser encontrados em [168].

2.2.1.2.4 Outros índices

Além dos índices citados anteriormente, existem outras técnicas analíticas para qualificação de modelos, podendo ser citados *minimum description lenght* (MDL), a comparação *bayesiana* de modelos e a estimativa da dimensão V-C.

O chamado MDL é um princípio que encontra base na teoria da informação. Informalmente, este princípio indutivo afirma que uma "mensagem" deve ser enviada ao seu "receptor" da forma mais compacta possível, ou seja, contendo o valor mínimo de bits necessário para a sua descrição. Este princípio é similar à chamada "navalha de *Occam*", que diz que, se dois modelos descrevem de maneira adequada um determinado fenômeno físico, deve ser adotado aquele que apresentar menor complexidade [164]. Sob este ponto de vista, o MDL fornece índices para comparação entre modelos, visto que busca modelos que "enviem a mensagem" com o número mínimo de bits necessário para codificação desta "mensagem". Uma breve descrição sobre o MDL pode ser encontrada em [158] e [160].

Outra forma interessante de comparação de modelos pode ser obtida através da aplicação de técnicas de inferência *bayesianas.* Estas técnicas buscam o cálculo da chamada evidência de cada modelo [164], com o intuito da estimação da probabilidade *a posteriori* de cada estrutura, dado um conjunto D de dados disponíveis para treinamento. Neste contexto, o modelo que apresentar a maior evidência é o mais provável de ser o mais correto. Maiores detalhes desta técnica podem ser encontrados em [158], [164] e [165].

Tomando por base o princípio da "navalha de *Occam*", que favorece modelos mais simples, o cálculo da complexidade efetiva de cada modelo pode ser utilizado para comparação entre estes, sendo escolhido o que apresentar menor complexidade. Neste sentido, a estimativa da dimensão V-C de cada modelo pode ser utilizada para comparação dos mesmos, visto que esta dimensão fornece informação efetiva sobre a complexidade de cada estrutura, diferentemente do número de parâmetros, por exemplo. Entretanto, esta estimativa não é feita de forma trivial, não existindo uma fórmula fechada para o cálculo da dimensão V-C. O desenvolvimento da teoria que embasa o cálculo da dimensão V-C, e a sua conseqüente aplicação à comparação de modelos, pode ser encontrada em [160] e [161]. Aplicações de índices analíticos para qualificação, e consequente comparação, de modelos neurais podem ser verificadas em [169] - [173].

2.2.1.3 Algoritmos de poda de rede

Conforme citado anteriormente, os algoritmos de poda de rede são utilizados para redução do número de neurônios nas camadas ocultas de um MLP, objetivando a obtenção de modelos com a melhor capacidade de generalização possível. De uma maneira geral, estas técnicas utilizam uma rede demasiadamente complexa, sendo determinado ao longo do processo quais pesos sinápticos, ou neurônios, devem ser eliminados, podendo gerar, portanto, modelos não conectados totalmente. Existem dois algoritmos gerais de poda de rede, ambos baseados na matriz *hessiana* $\underline{\underline{H}}(\underline{w})$. O primeiro é o chamado método do dano cerebral ótimo, *optimal brain damage* (OBD), o qual utiliza uma aproximação de $\underline{\underline{H}}(\underline{w})$ por uma matriz diagonal. O segundo, que pode ser considerado uma generalização do primeiro, é o chamado método do cirurgião cerebral ótimo, *optimal brain surgeon* (OBS), que não utiliza esta aproximação da

matriz $\underline{\underline{H}}(\underline{w})$, requerendo o cálculo completo desta matriz. Um algoritmo para o cálculo de $\underline{\underline{H}}(\underline{w})$ para MLP's pode ser encontrado em [174]. Ambos algoritmos selecionam os pesos que devem ser retirados da rede através do cálculo da chamada saliência de cada peso, definida como a variação no funcional de erro $E_s(\underline{w})$ em função da retirada deste peso específico do modelo [158].

Outra técnica de poda de rede existente busca a remoção não apenas de pesos sinápticos individualmente, mas sim de neurônios inteiros. Estes algoritmos utilizam o cálculo da saliência de cada neurônio, definida como a variação no funcional de erro $E_s(\underline{w})$ em função da retirada do respectivo neurônio do modelo.

Descrições mais detalhadas destes algoritmos podem ser encontradas em [151] e [158].

2.2.1.4 Métodos construtivos.

Estes métodos de estabilização de estrutura utilizam inicialmente uma rede excessivamente simples, sendo adicionados neurônios ao modelo ao longo do desenvolvimento do algoritmo. Um dos algoritmos construtivos mais conhecidos é o chamado correlação em cascata, *cascade correlation*, que determina o procedimento a ser seguido para inserção de um neurônio sigmoidal à rede. Este método é aplicado apenas a MLP's com neurônios sigmoidais na camada oculta, com a camada de saída apresentando função de ativação linear ou sigmoidal. Uma descrição deste algoritmo pode ser encontrada em [158]. Um estudo sobre a aplicação de algoritmos construtivos de MLP's pode ser encontrado em [175], com uma revisão bibliográfica sobre estes algoritmos podendo ser encontrada em [176].

2.2.2 Teoria da regularização de *Tikhonov*

De uma forma geral, problemas de aproximação funcional buscam a aproximação, ou interpolação, de uma função contínua $F(\underline{x}):\mathbb{R}^n \to \mathbb{R}$, por uma função $f(\underline{x},\underline{w}):\mathbb{R}^n \to \mathbb{R}$, definida por um vetor de parâmetros $\underline{w}=[w_1, w_2, ..., w_M]^t$, utilizando para tal um conjunto de exemplos $D=\{\underline{x}_k, d_k\}$, $k=1,2,...,N$, $\underline{x}_k=[x_{k1}, x_{k2}, ..., x_{kn}]^t$, e $d_k = F(\underline{x}_k)$, para problemas de interpolação, ou $d_k = F(\underline{x}_k)+\zeta_k$, para problemas de aproximação. Definido desta forma, este problema é classificado como mal-formulado, visto que a informação contida no conjunto de exemplos não é suficiente para reconstrução única do mapeamento entrada-saída em regiões onde não existem dados disponíveis [177]. Entretanto, é sabido que modelos com boa capacidade de generalização devem apresentar resultados satisfatórios exatamente nas regiões onde não existem dados disponíveis para treinamento.

Para solução de problemas de reconstrução de superfície mal-formulados, categoria na qual o problema de aproximação funcional acima definido está inserido, foi desenvolvida a teoria de regularização. Nesta teoria, a inserção de conhecimento prévio acerca do problema é necessária para tornar o problema bem formulado, e, na ausência de conhecimento sobre o problema, a única informação que pode ser inserida *a priori* diz respeito ao elevado grau de suavidade da função a ser aproximada [177]. Em linhas gerais, a suavidade de uma função está relacionada com as características globais desta, ou seja, o valor da função em um específico ponto depende do valor da mesma nas vizinhanças deste ponto. Neste contexto, a teoria da regularização afirma que a função de aproximação $f(\underline{x},\underline{w})$ deve ser obtida através da minimização do funcional de *Tikhonov*, dado por:

$$R\left[f\left(\underline{x},\underline{w}\right)\right] = E_s\left[f\left(\underline{x},\underline{w}\right)\right] + \lambda E_c\left[f\left(\underline{x},\underline{w}\right)\right]$$
$$E_s\left[f\left(\underline{x},\underline{w}\right)\right] = \frac{1}{2N}\sum_{i=1}^{N}\left[d_k - f\left(\underline{x}_k,\underline{w}\right)\right]^2 \qquad (2.23)$$
$$E_c\left[f\left(\underline{x},\underline{w}\right)\right] = \left\|Pf\left(\underline{x},\underline{w}\right)\right\|^2$$

Na equação (2.23), P é um operador diferencial responsável pela inserção da restrição de suavidade, $\|\cdot\|$ é uma norma definida no espaço ao qual $Pf\left(\underline{x},\underline{w}\right)$ pertence, e λ é o chamado parâmetro de regularização, responsável pelo equilíbrio entre o ajuste dos dados de treinamento e o controle de complexidade, ou regularização, do modelo [177]. O desenvolvimento desta teoria deu origem às chamadas redes de regularização, fornecendo embasamento teórico para o desenvolvimento das redes de função de base radial, *radial basis function networks* (RBFN's) [177].

No contexto de MLP's, onde a função $f\left(\underline{x},\underline{w}\right)$ será implementada por este modelo neural, esta teoria pode ser aplicada para treinamento de tais modelos, com a equação (2.23) resultando em:

$$R\left[\underline{w}\right] = E_s\left(\underline{w}\right) + \lambda E_c\left(\underline{w}\right)$$
$$E_s\left(\underline{w}\right) = \frac{1}{2N}\sum_{i=1}^{N}\left[d_k - f\left(\underline{x}_k,\underline{w}\right)\right]^2 \qquad (2.24)$$
$$E_c\left(\underline{w}\right) = \left\|Pf\left(\underline{x},\underline{w}\right)\right\|^2$$

A aplicação desta teoria para treinamento de MLP's apresenta duas dificuldades principais. A primeira está relacionada com a determinação do valor ótimo do parâmetro de regularização λ. Dentre as metodologias mais utilizadas para determinação deste parâmetro, merecem destaque as técnicas de re-amostragem descritas no item 2.2.1.1, com os métodos analíticos descritos do item 2.2.1.2 também podendo ser utilizados. Um método analítico para estimativa do parâmetro de regularização λ pode ser encontrado em [163]. Além destes procedimentos, a aplicação de técnicas de estimação *bayesiana* ao treinamento de MLP's fornece uma forma de

estimação automática deste parâmetro ao longo do algoritmo de treinamento, conforme será apresentado nos próximos itens.

A outra dificuldade da aplicação da teoria da regularização ao treinamento de MLP's reside na determinação do funcional $E_c\left(\underline{w}\right)$ responsável pela regularização do modelo, também conhecido como funcional regularizador. Diversas formas para este funcional foram propostas na literatura, com algumas destas técnicas sendo apresentadas nos próximos itens.

2.2.2.1 Decaimento dos pesos (*weight decay*)

Uma das formas mais simples e mais utilizadas para o funcional regularizador $E_c\left(\underline{w}\right)$ é dada pelo quadrado da norma euclidiana do vetor de parâmetros $\underline{w}$, dado por:

$$E_c\left(\underline{w}\right) = \left\|\underline{w}\right\|^2 = \sum_{j=1}^{M} w_j^2 \tag{2.25}$$

Apesar deste funcional regularizador ser bastante utilizado na literatura, uma das suas limitações reside na sua inconsistência com relação às propriedades de escalonamento dos mapeamentos realizados pelos MLP's [158]. Entretanto, esta indesejável característica pode ser superada através da utilização de um outro regularizador, dado pela seguinte equação, para MLP's com uma única camada oculta:

$$E_c\left(\underline{w}\right) = \sum_{j=1}^{M_1} w_{1j}^2 + \sum_{j=1}^{M_2} w_{2j}^2 \tag{2.26}$$

Na equação (2.26), $\underline{w}_1 = \left[w_{11}, w_{12}, ..., w_{1M_1} \right]^t$ representa o conjunto de pesos sinápticos que ligam as entradas aos neurônios da camada oculta, excluídos os *bias*, $\underline{w}_2 = \left[w_{21}, w_{22}, ..., w_{2M_2} \right]^t$ representa o conjunto de pesos que ligam os neurônios da camada oculta à camada de saída, com cada somatório possuindo o respectivo parâmetro de regularização λ. Qualitativamente, o regularizador de decaimento dos

pesos será consistente a transformações lineares das entradas e das saídas se, para cada camada oculta, for utilizado um funcional regularizador independente, garantindo, assim, consistência com relação às propriedades de escalonamento dos mapeamentos realizados pelos MLP's.

2.2.2.2 Eliminação dos pesos (*weight elimination*)

Outro funcional regularizador encontrado na literatura é o chamado de eliminação dos pesos, *weight elimination*, dado por:

$$E_c\left(\underline{w}\right) = \sum_{j=1}^{M} \frac{w_j^2}{\hat{w}^2 + w_j^2} \tag{2.27}$$

Na equação (2.27), $\hat{w}$ é uma constante definida *a priori*. Esse regularizador apresenta esta denominação devido ao fato que, ao contrário do decaimento dos pesos, que tende a fazer com que todos os pesos assumam valores pequenos, este regularizador permite a presença de alguns pesos com valores elevados, relativamente muito maiores que $\hat{w}$, com outros assumindo valores pequenos. Desta forma, os pesos com valores pequenos podem ser interpretados como sendo de pouca relevância para a saída do modelo, podendo, assim, ser eliminados. Uma das principais desvantagens deste regularizador reside na presença de outro parâmetro a ser especificado, $\hat{w}$.

2.2.2.3 Treinamento *bayesiano*

O treinamento *bayesiano* de MLP's, como o próprio nome já diz, resulta da aplicação de técnicas de inferência *bayesianas* ao problema de treinamento de MLP's. Esta abordagem foi proposta originalmente por *David J.C. Mackay*, em 1992 [164], cujas principais vantagens são as seguintes [174]:

- O algoritmo de retropropagação do erro tradicional pode ser visto como um caso particular dos resultados obtidos através da aplicação de técnicas de inferência *bayesianas*;
- A teoria da regularização apresenta uma interpretação natural dentro desta abordagem. Um dos motivos para este fato reside na obrigatoriedade de inserção de algum conhecimento prévio acerca do problema para obtenção da solução, característica marcante tanto das técnicas de inferência *bayesianas* quanto da teoria de regularização de *Tikhonov*;
- Para problemas de regressão, intervalos de confiança podem ser gerados automaticamente;
- O treinamento *bayesiano* de MLP's fornece uma estimativa automática do parâmetro de regularização λ, o qual é atualizado ao longo do algoritmo de treinamento, sem a necessidade da utilização de técnicas de re-amostragem ou de qualificação analítica de modelos para estimativa deste parâmetro;
- Esta metodologia permite a comparação entre diferentes modelos através da utilização apenas dos dados disponíveis para treinamento;
- As técnicas de inferência *bayesiana* permitem afirmar em qual região do espaço de entrada devem ser obtidos novos dados com o intuito de aumentar a informação contida no modelo, característica conhecida como aprendizado ativo;
- Este procedimento permite o desenvolvimento de algoritmo de determinação automática de relevância das entradas, *automatic relevance determination* (ARD), uma técnica interessante para seleção de variáveis de entrada de modelos neurais.

Visto que esta tese utilizará o algoritmo de treinamento *bayesiano* de MLP's para problemas de aproximação funcional, utilizando MLP's com uma única camada escondida e uma única saída linear, a discussão sobre este assunto estará restrita à apresentação deste algoritmo para este tipo de problema. Maiores detalhes sobre a aplicação de técnicas *bayesianas* ao treinamento de MLP's, comparação de modelos e determinação automática da relevância das entradas podem ser encontrados em [158], [164] e [165].

Definida a estrutura do MLP a ser utilizado, ou seja, número de camadas ocultas, número de neurônios por camada e tipo de função de ativação de cada neurônio, dado o conjunto $D=\{X,Y\}$, $X=\{\underline{x}_1,...,\underline{x}_k\}$, $Y=\{d_1,...,d_k\}$, $k=1,2,...,N$, $\underline{x}_k=[x_{k1},...,x_{kn}]^t$, $d_k=F(\underline{x}_k)+\zeta_k$, contendo N pares entrada-saída, o objetivo do treinamento do modelo, sob o ponto de vista da inferência *bayesiana*, reside na determinação do vetor de parâmetros $\underline{w}=[w_1,w_2,...,w_M]^t$ que maximize a probabilidade *a posteriori* $p(\underline{w}|Y,X)$, dada por:

$$p(\underline{w}|Y,X)=\frac{p(Y|\underline{w},X)\,p(\underline{w}|X)}{p(Y|X)} \tag{2.28}$$

Na equação (2.28), $p(Y|X)=\int p(Y|\underline{w},X)\,p(\underline{w}|X)\,d\underline{w}$ é um fator de normalização, que garante que $\int p(\underline{w}|Y,X)\,d\underline{w}=1$. Além disso, visto que os MLP's em geral não modelam a distribuição de probabilidade $p(\underline{x})$ geradora dos padrões de entrada do conjunto de treinamento, e que o conjunto X aparece como variável condicionante em todas as probabilidades envolvidas na equação (2.28), este conjunto será omitido da notação a partir deste ponto.

Portanto, para o cálculo da probabilidade a *posteriori* $p(\underline{w}|Y)$ do vetor $\underline{w}$, é necessário o conhecimento da distribuição de probabilidade *a priori* $p(\underline{w})$ deste vetor, como também a sua função de verossimilhança $p(Y|\underline{w})$, a qual está relacionada com a distribuição de probabilidade do ruído existente na saída desejada. Na ausência de conhecimento prévio sobre a solução, conforme é o caso do treinamento de MLP's, a escolha da distribuição $p(\underline{w})$ deve refletir tal falta de conhecimento. Entretanto, conforme evidenciado nos funcionais regularizadores descritos nos itens 2.2.2.1 e 2.2.2.2, redes com pesos sinápticos assumindo valores pequenos tendem a apresentar maior capacidade de generalização. Diante disso, uma escolha razoável para a distribuição $p(\underline{w})$ reside na distribuição *gaussiana* com vetor média nulo e matriz de covariância $\alpha^{-1}\underline{\underline{I}}$, $\underline{\underline{I}}$ igual à matriz identidade de dimensão $M \times M$, dada por [158]:

$$p(\underline{w}) = \frac{1}{Z_{\underline{W}}(\alpha)} e^{-\left(\frac{\alpha}{2}\|\underline{w}\|^2\right)} \tag{2.29}$$

$$Z_{\underline{W}}(\alpha) = \left(\frac{2\pi}{\alpha}\right)^{\frac{M}{2}}$$

Na equação (2.29), α é o chamado hiperparâmetro, cuja estimativa será apresentada ao longo deste item, e que, para este estágio da apresentação do algoritmo, é admitido que seja um valor conhecido.

A principal vantagem da utilização da distribuição de probabilidade *a priori* $p(\underline{w})$ dada pela equação (2.29) reside única e exclusivamente na simplificação das análises subseqüentes. Outras distribuições podem ser utilizadas, conforme apresentado em [158].

Definida a distribuição $p(\underline{w})$, resta agora definir a distribuição de probabilidade do ruído ζ existente na saída desejada. Supondo que a função a ser aproximada $F(\underline{x})$

apresenta um certo grau de suavidade, e que o ruído ζ apresenta distribuição gaussiana com média nula e variância β^{-1}, probabilidade da ocorrência de uma saída alvo específica d_k, dado o vetor de entrada $\underline{x}_k$ e o vetor de parâmetros $\underline{w}$ é dada ela seguinte expressão [158]:

$$p\left(d_k \middle| \underline{x}_k, \underline{w}\right) = \frac{e^{\left\{-\frac{\beta}{2}\left[d_k - f\left(\underline{x}_k, \underline{w}\right)\right]^2\right\}}}{\int e^{\left\{-\frac{\beta}{2}\left[d_k - f\left(\underline{x}_k, \underline{w}\right)\right]^2\right\}} dd_k} \tag{2.30}$$

Portanto, partindo do pressuposto que os padrões do conjunto de treinamento foram obtidos de maneira independente a partir desta distribuição, podemos obter uma expressão para a verossimilhança $p\left(Y \middle| \underline{w}\right)$, dada pela equação:

$$p\left(Y \middle| \underline{w}\right) = \frac{1}{Z_Y\left(\beta\right)} e^{\left\{-\frac{\beta}{2}\sum_{k=1}^{N}\left[d_k - f\left(\underline{x}_k, \underline{w}\right)\right]^2\right\}} \tag{2.31}$$

$$Z_Y\left(\beta\right) = \left(\frac{2\pi}{\beta}\right)^{\frac{N}{2}}$$

Na equação (2.31), β é também um hiperparâmetro, cuja estimativa será apresentada ao longo deste item. Da mesma forma que α, para este estágio da discussão é assumido que este parâmetro apresente um valor conhecido.

De posse das expressões (2.29) e (2.31), podemos agora calcular a probabilidade *a posteriori* de $\underline{w}$ dado o conjunto de saídas desejadas Y, através da aplicação da regra de *Bayes* descrita na equação (2.28), resultando na seguinte expressão [158]:

$$p\left(\underline{w} \middle| Y\right) = \frac{1}{Z_S} e^{\left[-S\left(\underline{w}\right)\right]} \tag{2.32}$$

$$Z_s = \int e^{-S\left(\underline{w}\right)} d\underline{w}$$

$$S\left(\underline{w}\right) = \frac{\beta}{2}\sum_{k=1}^{N}\left[d_k - f\left(\underline{x}_k, \underline{w}\right)\right]^2 + \frac{\alpha}{2}\sum_{j=1}^{M} w_j^2$$

Visto que a primeira parcela do funcional $S(\underline{w})$ é igual ao funcional de risco empírico $E_s(\underline{w})$ dado pela equação (2.24), a menos da constante $1/\beta$, podemos concluir que maximizar a probabilidade *a posteriori* de $\underline{w}$ é equivalente à aplicação do regularizador de decaimento de pesos para treinamento de MLP's. Além disso, visto que a busca é pelo ponto ótimo $\underline{w}^*$ que minimiza $S(\underline{w})$, da equação (2.32), $\lambda = \alpha/\beta$. Portanto, o cálculo dos hiperparâmetros α e β fornece uma estimativa do parâmetro de regularização λ.

Antes de apresentar a estimativa dos hiperparâmetros α e β, vale a pena ressaltar que o mesmo problema de consistência destacado no item 2.2.2.1 para o procedimento de decaimento dos pesos vale para a escolha de distribuição *gaussiana* para a probabilidade *a priori* $p(\underline{w})$. Neste novo contexto, esta inconsistência pode estar relacionada com o uso de uma única distribuição *a priori* para todos os pesos sinápticos, parecendo ser mais razoável a utilização de uma distribuição *a priori* para cada conjunto específico de pesos, com estes conjuntos sendo definidos pelo usuário. Exemplificando, um conjunto pode estar relacionado com os pesos que ligam um determinado conjunto de entradas à camada oculta, outro contendo os pesos que ligam os neurônios da camada oculta à saída, e um outro contendo os pesos restantes. Entretanto, a utilização de distribuições que não levem em consideração os *bias*, resultando em funcionais regularizadores como o da equação (2.26), podem resultar em distribuições impróprias, ou seja, que não podem ser normalizadas, comprometendo a estimativa dos hiperparâmetros [158], o que deve ser evitado.

Conforme citado anteriormente, a aplicação das técnicas de inferência *bayesianas* foi limitada à estimativa do vetor de parâmetros $\underline{w}$, supondo que os

hiperparâmetros α e β eram conhecidos. Entretanto, sabemos que estes valores são desconhecidos *a priori*, sendo necessária uma estimativa para estas variáveis.

Portanto, visto que $\underline{w}$, α e β são desconhecidos, a probabilidade *a posteriori* de $\underline{w}$, $p\left(\underline{w}|Y\right)$, pode ser dada por:

$$p\left(\underline{w}|Y\right)=\iint p\left(\underline{w},\alpha,\beta|Y\right)d\alpha d\beta=\iint p\left(\underline{w}|\alpha,\beta,Y\right)p\left(\alpha,\beta|Y\right)d\alpha d\beta \tag{2.33}$$

Qualitativamente, visto que $\underline{w}$ não é a única variável desconhecida, a sua probabilidade *a posteriori* $p\left(\underline{w}|Y\right)$ deve ser obtida através da integração da probabilidade *a posteriori* da ocorrência das três variáveis desconhecidas $p\left(\underline{w},\alpha,\beta|Y\right)$ sobre todo o espaço de hiperparâmetros.A partir da equação (2.33), existe duas abordagens para a estimativa dos hiperparâmetros α e β. Uma utiliza a integração analítica sobre os hiperparâmetros, abordagem que foge do escopo desta tese. A segunda abordagem, conhecida como aproximação da evidência [158], proposta por *Mackay* [164], será utilizada nesta tese. As duas abordagens levam a resultados semelhantes, conforme pode ser verificado em [158]. Uma breve discussão sobre estes dois procedimentos pode ser encontrada na mesma referência.

A abordagem proposta por *Mackay* [164] parte do pressuposto que a probabilidade *a posteriori* $p\left(\alpha,\beta|Y\right)$ apresenta pouca dispersão em torno dos valores mais prováveis α^* e β^*, permitindo a seguinte simplificação da equação (2.33):

$$p\left(\underline{w}|Y\right)\simeq p\left(\underline{w}|\alpha^*,\beta^*,Y\right)\iint p\left(\alpha,\beta|Y\right)d\alpha d\beta=p\left(\underline{w}|\alpha^*,\beta^*,Y\right) \tag{2.34}$$

Portanto, para determinação dos valores mais prováveis para os hiperparâmetros, a probabilidade *a posteriori* destes, $p\left(\alpha,\beta|Y\right)$, deve ser maximizada. Aplicando a regra de *Bayes*, é obtida a seguinte expressão para esta probabilidade:

$$p(\alpha,\beta|Y)=\frac{p(Y|\alpha,\beta)\,p(\alpha,\beta)}{p(Y)} \tag{2.35}$$

Da equação (2.35), é visto que algum conhecimento prévio acerca dos hiperparâmetros α e β deve ser inserido, na forma da distribuição de probabilidade $p(\alpha,\beta)$. Visto que pouco, ou mesmo nenhum, conhecimento sobre os hiperparâmetros é disponível, a única informação prévia que pode ser inserida diz respeito a esta ausência de conhecimento acerca da solução. Portanto, a probabilidade $p(\alpha,\beta)$ deve ser escolhida de tal forma que esta distribuição seja insensível a valores específicos de α e β. Logo, visto que a probabilidade $p(Y)$ é independente dos hiperparâmetros, a maximização da probabilidade *a posteriori* $p(\alpha,\beta|Y)$ é obtida através da maximização da probabilidade $p(Y|\alpha,\beta)$, também conhecida como evidência para os hiperparâmetros [158]. Esta probabilidade pode ser obtida através da seguinte expressão:

$$p(Y|\alpha,\beta)=\int p(Y|\underline{w},\alpha,\beta)\,p(\underline{w}|\alpha,\beta)\,d\underline{w} \tag{2.36}$$

Visto que o hiperparâmetro α está relacionado apenas com a probabilidade a *priori* do vetor $\underline{w}$, e que β está relacionado apenas com a distribuição do ruído ζ existente na saída desejada, a equação (2.36) passa a ser dada por:

$$p(Y|\alpha,\beta)=\int p(Y|\underline{w},\beta)\,p(\underline{w}|\alpha)\,d\underline{w} \tag{2.37}$$

Portanto, utilizando as equações (2.29) e (2.31) na equação (2.37), a seguinte expressão é obtida:

$$p(Y|\alpha,\beta)=\frac{Z_s(\alpha,\beta)}{Z_Y(\beta)\,Z_{\underline{W}}(\alpha)} \tag{2.38}$$

$$Z_s(\alpha,\beta)=\int e^{\left[-S(\underline{w})\right]}d\underline{w}$$

Na equação (2.38), $S(\underline{w})$ é dado pela equação (2.32). Considerando uma aproximação *gaussiana* da distribuição de probabilidade *a posteriori* do vetor $\underline{w}$, $p(\underline{w}|Y)$, que é equivalente à aproximação quadrática em séries de *Taylor* do funcional $S(\underline{w})$ em torno do ponto $\underline{w}^*$, o funcional $Z_s(\alpha,\beta)$ passa a ser dado por:

$$Z_s(\alpha,\beta) = e^{-S(\underline{w}^*)}(2\pi)^{\frac{M}{2}}\left\{\det\left[\underline{\underline{A}}(\underline{w})\Big|_{\underline{w}=\underline{w}^*}\right]\right\}^{-\frac{1}{2}} \tag{2.39}$$
$$\underline{\underline{A}}(\underline{w})\Big|_{\underline{w}=\underline{w}^*} = \beta\,\underline{\underline{H}}(\underline{w})\Big|_{\underline{w}=\underline{w}^*} + \alpha\,\underline{\underline{I}}$$

Na equação (2.39), $\underline{w}^*$ e'o vetor de parâmetros que minimiza o funcional $S(\underline{w})$, $\underline{\underline{H}}(\underline{w})\Big|_{\underline{w}=\underline{w}^*}$ a matriz *hessiana* do funcional $S(\underline{w})$, calculada no ponto $\underline{w}^*$ e $\underline{\underline{I}}$ a matriz identidade de dimensão $M \times M$. Utilizando esta equação em conjunto com as expressões obtidas para $Z_{\underline{W}}(\alpha)$ e $Z_Y(\beta)$, dadas pelas expressões (2.29) e (2.31), respectivamente, o logaritmo *neperiano* da expressão (2.38) é dado por:

$$\ln p(Y|\alpha,\beta) = -\frac{\alpha}{2}\sum_{j=1}^{M}\left(w_j^*\right)^2 - \frac{\beta}{2}\sum_{k=1}^{N}\left[d_k - f\left(\underline{x}_k,\underline{w}^*\right)\right]^2 - \frac{1}{2}\ln\left\{\det\left[\underline{\underline{A}}(\underline{w})\Big|_{\underline{w}=\underline{w}^*}\right]\right\} \tag{2.40}$$
$$+\frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi)$$

A maximização do logaritmo natural da evidência para os hiperparâmetros α e β, dado pela equação (2.40), em relação a α resulta na seguinte expressão:

$$\gamma = \alpha\sum_{j=1}^{M}\left(w_j^*\right)^2 = M - trace\left\{\left[\underline{\underline{A}}(\underline{w})\Big|_{\underline{w}=\underline{w}^*}\right]^{-1}\right\} \tag{2.41}$$

Na equação (2.41), a grandeza γ é o chamado número efetivo de parâmetros, utilizado pelos métodos analíticos de qualificação de modelos apresentados no item 2.2.1.2. Com o intuito de evitar a inversão da matriz $\underline{\underline{A}}(\underline{w})\Big|_{\underline{w}=\underline{w}^*}$, seja o conjunto de M autovalores da

matriz *hessiana* $\underline{\underline{H}}(\underline{w})\big|_{\underline{w}=\underline{w}^*}$, dado por $\lambda = \{\nu_1, \nu_2, ..., \nu_M\}$. Desta forma, a equação (2.41) passa a ser dada por:

$$\gamma = \sum_{i=1}^{M} \frac{\nu_i}{\nu_i + \alpha} \tag{2.42}$$

A maximização do logaritmo natural da evidência para os hiperparâmetros α e β, dado pela equação (2.40), em relação a β resulta na seguinte equação:

$$\beta \sum_{k=1}^{N} \left[d_k - f\left(\underline{x}_k, \underline{w}^*\right) \right]^2 = N - \gamma \tag{2.43}$$

As expressões (2.41) e (2.43) foram obtidas a partir da aproximação quadrática do funcional $S(\underline{w})$ em torno do seu ponto de mínimo $\underline{w}^*$. Portanto, assim como os métodos de segunda ordem de treinamento de MLP's descritos no Apêndice A, que utilizam a cada iteração uma aproximação quadrática do funcional de risco empírico em torno do ponto de operação $\underline{w}(l)$, a aproximação quadrática do funcional $S(\underline{w})$ pode ser utilizada em algoritmo iterativo, dando origem às seguintes equações recursivas para estimativa dos hiperparâmetros α e β:

$$\alpha(l+1) = \frac{\gamma(l+1)}{\sum_{j=1}^{M} \left[w_j(l) \right]^2} \tag{2.44}$$

$$\beta(l+1) = \frac{N - \gamma(l+1)}{\sum_{k=1}^{N} \left\{ d_k - f\left[\underline{x}_k, \underline{w}(n) \right] \right\}^2} \tag{2.45}$$

$$\gamma(l+1) = \sum_{i=1}^{M} \frac{\nu_i(l)}{\nu_i(l) + \alpha(l)} \tag{2.46}$$

Na equação (2.46), $\nu_i(l)$ representa o *i*-ésimo autovalor da matriz *hessiana* $\underline{\underline{H}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}$.

Conforme descrito anteriormente, a abordagem *bayesiana* do treinamento de MLP's conduz automaticamente à minimização de um funcional $S(\underline{w})$, dado pela equação (2.32), que inclui um termo responsável pela regularização do modelo estimado. Visto que o objetivo da utilização deste funcional reside na sua minimização, a equação (2.32) pode ser reescrita na seguinte forma:

$$S(\underline{w}) = \frac{1}{2N}\sum_{k=1}^{N}\left[d_k - f(\underline{x}_k, \underline{w})\right]^2 + \frac{\alpha}{\beta}\frac{1}{2}\sum_{j=1}^{M} w_j^2 \tag{2.47}$$

$$E_s(\underline{w}) = \frac{1}{2N}\sum_{k=1}^{N}\left[d_k - f(\underline{x}_k, \underline{w})\right]^2$$

$$E_c(\underline{w}) = \frac{1}{2}\sum_{j=1}^{M} w_j^2$$

Comparando a equação (2.47) com a equação (2.23), o treinamento *bayesiano* pode ser interpretado como a aplicação da teoria de regularização de *Tikhonov*, com a vantagem de fornecer um procedimento automático para estimativa do parâmetro de regularização $\lambda = \alpha/\beta$, através da aplicação das equações (2.44) a (2.46).

Portanto, o algoritmo de treinamento *bayesiano* de MLP's pode ser resumido da forma que segue:

1. Faça $l = 0$;
2. Inicialize o vetor de parâmetros $\underline{w}(l)$ e os hiperparâmetros $\alpha(l)$ e $\beta(l)$;
3. Utilizando alguma técnica de otimização, atualize o vetor de parâmetros $\underline{w}(l+1)$ buscando a minimização do funcional $S(\underline{w})$ dado pela equação (2.47);
4. Se o critério de parada for atendido, encerre o algoritmo. Do contrário, vá para o passo 5;
5. Calcule os autovalores da matriz *hessiana* $\underline{\underline{H}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}$. Um algoritmo completo para cálculo desta matriz para MLP's pode ser encontrado em [174];

6. Atualize os hiperparâmetros $\alpha(l+1)$ e $\beta(l+1)$ utilizando as equações (2.44) a (2.46);
7. Faça $l = l+1$ e retorne ao passo 3.

Apesar da metodologia proposta utilizar uma série de pressupostos não verificados na prática, como a aproximação *gaussiana* da probabilidade *a posteriori* do vetor $\underline{w}$, $p(\underline{w}|Y)$, por exemplo, este procedimento apresenta como principal vantagem a estimativa automática do parâmetro de regularização λ, evitando a necessidade da utilização de um conjunto de validação, permitindo que todos os dados sejam utilizados para treinamento do modelo. Além disso, esta estimativa é feita iterativamente ao longo do algoritmo, fazendo com que o valor de λ seja ajustado à medida em que o processo iterativo evolui. Conforme citado anteriormente neste item, o desempenho deste algoritmo pode ser melhorado a partir da escolha de uma distribuição de probabilidade *a priori* diferente para cada conjunto de pesos sinápticos, resultando em um funcional regularizador para cada conjunto de pesos, de forma semelhante ao descrito para o procedimento de decaimento de pesos.

2.2.2.4 Outros funcionais regularizadores

Além dos funcionais descritos nos itens anteriores, existe diversos outros funcionais $E_c(\underline{w})$ propostos na literatura para regularização de MLP's. Um exemplo de funcional regularizador é o chamado suavizador aproximativo [151], proposto para MLP's contendo uma única camada escondida e uma única saída linear, dado por:

$$E_c(\underline{w}) = \sum_{j=1}^{n_1} \left(w_{21j}\right)^2 \left\|\underline{w}_j\right\|^p \tag{2.48}$$

Na equação (2.48) w_{21j} representa o pesos sináptico que liga o *j*-ésimo neurônio da camada oculta à saída, $\underline{w}_j$ representa um vetor coluna contendo os pesos sinápticos que

ligam as entradas ao *j*-ésimo neurônio da camada oculta, n_1 representa o número de neurônios na camada escondida, e *p* representa um parâmetro definido pelo usuário, relacionado com o nível de suavidade admitido *a priori* para a função a ser aproximada $F(\underline{x})$.

Outra forma de regularização é conhecida como suavização da curvatura, *cruvature-driven smoothing*, que pode ser considerada uma versão discreta do funcional contínuo de *Tikhonov*, dado pela equação (2.23) [158]. Matematicamente, o funcional regularizador $E_c(\underline{w})$ utilizado por este procedimento de regularização é dado por:

$$E_c(\underline{w}) = \frac{1}{2N}\sum_{k=1}^{N}\sum_{i=1}^{n}\sum_{j=1}^{m}\left[\left.\frac{\partial^p f_j(\underline{x},\underline{w})}{\partial x_i^{\,p}}\right|_{\underline{x}=\underline{x}_k}\right]^2 \tag{2.49}$$

Na equação (2.49) $\left.\partial^p f_j(\underline{x},\underline{w})/\partial x_i^{\,p}\right|_{\underline{x}=\underline{x}_k}$ representa a *p*-ésima derivada parcial da *j*-ésima saída do MLP em relação a *i*-ésima entrada, calculada no ponto $\underline{x}_k$, N representa o número de padrões para treinamento, n o número de entradas e m o número de saídas. A descrição de um algoritmo para treinamento de MLP's utilizando este funcional regularizador, para $p = 2$, pode ser encontrada em [178].

2.2.3 Heurísticas para controle de complexidade

Além dos dois procedimentos gerais de regularização de MLP's apresentados anteriormente, baseados na determinação da estrutura suficiente, no sentido do teorema da aproximação universal, para abordagem do problema em estudo, para o caso dos procedimentos de estabilização de estrutura, ou na inserção de restrições ao problema de minimização a ser resolvido ao longo do treinamento, para o caso da teoria de regularização, existe um outro conjunto de metodologias para controle de complexidade de MLP's, as quais utilizam algumas heurísticas para obtenção de modelos com boa

capacidade de generalização. Nesta categoria podem ser inseridos a parada antecipada do treinamento, *early stopping*, o treinamento com inserção de ruído e o escalonamento do ganho da função de ativação. Estes três procedimentos serão brevemente apresentados nos próximos itens.

2.2.3.1 Parada antecipada do treinamento (*early stopping*)

Conforme apresentado no início deste capítulo, na presença de dados ruidosos, a aplicação do princípio da minimização do risco empírico pode resultar na estimação de modelos com reduzida capacidade de generalização, em função da ocorrência do chamado *overfitting*, que nada mais é que o ajuste excessivo dos dados de treinamento. Em virtude disto, a parada antecipada de treinamento é uma heurística utilizada com o intuito de evitar a ocorrência deste problema, através da monitoração do erro para um conjunto independente de dados.

De uma maneira geral, esta heurística utiliza três conjunto de dados, um para treinamento, outro para validação e um último para teste, escolhidos e utilizados de maneira semelhante ao procedimento de validação única descrito no item 2.2.1.1. Entretanto, apesar dos conjuntos de treinamento e de teste serem utilizados com o mesmo objetivo da validação única, neste contexto o erro para o conjunto de validação é utilizado como critério de parada do algoritmo de treinamento do MLP, e não para comparação entre modelos, com o intuito de evitar a ocorrência de *overfitting*.

Para o caso de funções de erro quadrático, como é o caso do treinamento de MLP's através da aplicação do princípio da minimização do risco empírico, a parada antecipada do treinamento apresenta um comportamento similar à aplicação do funcional regularizador de decaimento dos pesos, descrito no item 2.2.2.1, conforme mostrado em [158].

Entretanto, apesar do elevado número de aplicações encontradas na literatura que utilizam esta heurística, conforme mostrado no item 1.3, o trabalho apresentado em [179] sobre o ajuste excessivo dos dados de treinamento, baseado na teoria estatística, recomenda que esta heurística seja aplicada com cautela. Neste trabalho, os autores mostram que esta metodologia só apresenta resultados satisfatórios em termos de melhoria da capacidade de generalização para problemas em que o número de dados disponíveis para treinamento é menor que o número de parâmetros a serem estimados pelo modelo. Para os casos que o número de padrões é muito maior que o número de parâmetros, a parada antecipada fornece pouco, ou até mesmo nenhum, ganho em termos de melhoria da capacidade de generalização, quando comparado com a aplicação direta do princípio da minimização do risco empírico. Restrições a aplicação desta técnica podem ser encontradas também em [180].

2.2.3.2 Treinamento com inserção de ruído

Esta heurística de regularização de MLP's consiste na inserção de ruído aditivo aos padrões de entrada integrantes do conjunto de treinamento. Basicamente, para um dado padrão $\underline{x}_k$, esta técnica está baseada na criação de versões corrompidas deste padrão, ou seja, $\underline{x}_k' = \underline{x}_k + \underline{\upsilon}$, com $\underline{\upsilon}$ sendo um vetor de números aleatórios gerados no mesmo intervalo utilizado pelas entradas, com a saída desejada associada a este padrão corrompido igual à saída associada ao padrão $\underline{x}_k$, ou seja, para um dado par entrada-saída $(\underline{x}_k, d_k)$, são gerados diversos pares $(\underline{x}_k', d_k)$. Desta forma, é esperado que a tarefa de ajustar um específico par $(\underline{x}_k, d_k)$ seja dificultada, diminuindo a possibilidae da ocorrência de *overfitting*.

A inserção de ruído nas entradas pode ser interpretada de outra maneira. Qualitativamente, a geração de versões corrompidas das entradas originais, sem

alteração das saídas desejadas associadas, pode significar que, para padrões de entrada similares, a saída sofrerá pouca ou nenhuma alteração, o que é equivalente a supor que a função a ser aproximada $F(\underline{x})$ apresenta um determinado grau de suavidade, conforme assumido na teoria de regularização de *Tikhonov*. A relação existente entre o treinamento com inserção de ruído e a teoria de regularização pode ser encontrada em [158] e [181].

2.2.3.3 Escalonamento do ganho da função de ativação

Conforme apresentado no Apêndice A, as funções de ativação sigmoidais utilizadas nos neurônios da camada oculta de MLP's apresentam um parâmetro a, conhecido como ganho destas funções. De uma forma qualitativa, o efeito da variação deste parâmetro reside no aumento, ou diminuição, da região linear de operação destas funções, conforme evidenciado na Figura A.1 e na Figura A.2. Neste sentido, informalmente pode ser afirmado que, para um dado MLP contendo uma única camada oculta e uma única saída linear, quanto maior for a região linear de operação das funções de ativação dos neurônios da camada escondida, menor será a não-linearidade modelada pela saída do MLP. No caso extremo, em que as funções sigmoidais estejam definidas apenas em suas respectivas regiões de operação linear, podendo assim ser consideradas como funções de ativação lineares, a saída do MLP será dada pela soma ponderada de uma série de transformações lineares do espaço de entrada, fazendo com o MLP seja considerado puramente uma máquina de aprendizagem linear. Portanto, o ajuste dos ganhos a das funções de ativação dos neurônios da camada oculta de um MLP parece ser uma heurística razoável para controle de complexidade destes modelos, visto que tais ganhos estão diretamente relacionados com o nível de não-linearidade modelado pela saída deste tipo de modelo.

Neste contexto, REED *et. al.* [181] propuseram uma metodologia de ajuste pós-treinamento dos ganhos a das funções de ativação de MLP's com uma única camada oculta e uma única saída linear, mostrando as semelhanças entre esta metodologia e a heurística de inserção de ruído nas entradas, apresentada no item 2.2.3.2, e com a teoria de regularização, apresentada no item 2.2.2.

Conforme exposto em [181], para MLP's com uma única camada oculta contendo neurônios não-lineares e uma única saída linear, se as amostras do conjunto de treinamento forem obtidas seguindo uma distribuição uniforme e o ruído adicionado às entradas integrantes do conjunto original de dados disponíveis para treinamento for *gaussiano*, com vetor média nulo e matriz de covariância $\sigma_{ruido}^2 \underline{\underline{I}}$, $\underline{\underline{I}}$ igual à matriz identidade de dimensão $n \times n$, o MLP estimado através da aplicação do princípio da minimização do risco empírico irá apresentar capacidade de generalização similar a um modelo treinado com o conjunto original de dados, ou seja, contendo apenas os padrões não-corrompidos, porém utilizando alguma técnica de controle de complexidade baseada na teoria de regularização descrita no item 2.2.2. Sob este mesmo conjunto de hipóteses, um MLP treinado através da minimização única e exclusiva do erro para o conjunto original de dados irá apresentar capacidade de generalização similar aos modelos estimados através da aplicação da teoria de regularização ou da heurística de inserção de ruído, se os ganhos das funções de ativação dos neurônios da camada oculta deste MLP forem multiplicados pelo fator a_j, dado por:

$$a_j = \frac{1}{\sqrt{\left\|\underline{w}_j\right\|^2 \sigma_{ruido}^2 + 1}} \qquad (2.50)$$

$$j = 1, 2, ..., n_1$$

Na equação (2.50) a_j representa o ganho da função de ativação do *j*-ésimo neurônio da camada oculta, $\underline{w}_j$ o vetor coluna contendo os pesos sinápticos que ligam as entradas ao

j-ésimo neurônio, excluído o *bias* associado a este neurônio, n_1 representa o número de neurônios na camada escondida, e σ^2_{ruido} representa a variância do ruído aditivo utilizado para aplicação da heurística de inserção de ruído. O procedimento de ajuste dos ganhos da função de ativação é conhecido como escalonamento do ganho da função de ativação.

Apesar do procedimento de escalonamento do ganho da função de ativação ter sido desenvolvido para modelos neurais tipo MLP, os resultado obtidos em [181] podem ser expandidos para modelos neurais *feedforward* com uma única camada oculta contendo neurônios não-lineares e uma única saída linear, sugerindo um procedimento pós-treinamento de ajuste dos ganhos das funções de ativação de modelos deste tipo, através da utilização da equação (2.50), com o intuito de melhorar a capacidade de generalização do modelo estimado. Além disso, estes resultados sugerem que, ao invés de aplicar a heurística de inserção de ruído nas entradas, que demanda um maior esforço computacional para treinamento, em função da necessidade de gerar, para cada padrão $\underline{x}_k$, pelo menos duas versões corrompidas deste, triplicando com isso o conjunto de treinamento, basta escalonar os ganhos das funções de ativação, utilizando a equação (2.50), do modelo estimado através da minimização do erro para o conjunto de treinamento original para que seja obtido um modelo com capacidade de generalização similar ao estimado através da inserção de ruído nas entradas.

Entretanto, da mesma forma que na teoria de regularização, onde o parâmetro de regularização λ deve ser estimado, o procedimento de escalonamento do ganho da função de ativação apresenta como principal desvantagem a necessidade de estimar a variância σ^2_{ruido} utilizada na equação (2.50). Usualmente, este parâmetro é estimado através da aplicação de técnicas de re-amostragem, como as apresentadas no item

2.2.1.1, com os métodos analíticos de qualificação de modelos apresentados no item 2.2.1.2 também podendo ser utilizados para a estimativa de σ^2_{ruido}.

2.2.4 Comparação entre as técnicas de controle de complexidade de MLP's

Na seção 2.2, foram apresentadas algumas técnicas de controle de complexidade de modelos neurais, aplicadas especificamente aos MLP's. Basicamente, estas técnicas podem ser agrupadas em três procedimentos gerais.

As metodologias baseadas na estabilização de estrutura buscam a determinação da estrutura suficiente para solução do problema em estudo, suficiente no sentido de evitar a utilização de modelos com complexidade excessiva, apresentando reduzida capacidade de generalização. Os procedimentos de estabilização de estrutura apresentam uma série de desvantagens, podendo ser citadas: a necessidade da utilização de conjuntos independentes de dados para aplicação de técnicas de re-amostragem; a utilização de índices desenvolvidos para modelos lineares, como AIC, BIC, cujas características assintóticas não são verificadas para modelos não-lineares; a utilização de índices de difícil cálculo analítico, como o NIC, MDL, dimensão V-C e comparação *bayesiana* de modelos; cálculo da inversa da matriz *hessiana* para aplicação em algoritmos de poda de rede, que pode ser computacionalmente custoso para problemas de grande porte, conforme citado no Apêndice A. Entretanto, estes procedimentos, mais especificamente aqueles baseados na comparação entre modelos com estruturas diferentes, utilizando para tal comparação técnicas de re-amostragem, apresentam como principal vantagem a sua facilidade de implementação. Para problemas em que a quantidade de dados disponíveis para treinamento não constitui uma restrição, sendo possível a utilização de alguns destes dados para validação e teste dos diversos modelos estimados, a utilização este tipo específico de procedimento de estabilização de estrutura é interessante.

As técnicas baseadas na teoria de regularização apresentam como principais desvantagens a escolha adequada do funcional regularizador $E_c(\underline{w})$ para o problema em estudo, e a estimativa do parâmetro de regularização λ, responsável pelo equilíbrio entre o ajuste dos dados de treinamento e a regularização do modelo estimado. Entretanto, conforme visto no item 2.2.2.3, a aplicação de técnicas de inferência *bayesiana* fornece uma base teórica para a escolha do funcional $E_c(\underline{w})$, através da escolha adequada da distribuição de probabilidade *a priori* $p(\underline{w})$ do vetor de pesos $\underline{w}$, refletindo a necessidade de inserção de conhecimento prévio acerca do problema, característica marcante das técnicas de inferência *bayesiana* e da teoria da regularização. Além disto, a abordagem *bayesiana* fornece um procedimento automático para estimação do parâmetro de regularização λ ao longo do algoritmo de treinamento, sem a necessidade da utilização de técnicas de re-amostragem, por exemplo, para estimativa deste parâmetro. Entretanto, o treinamento *bayesiano* apresenta como principal desvantagem o elevado número de aproximações e hipóteses consideradas ao longo do desenvolvimento do mesmo, com muitas delas podendo não ser verificadas em aplicações práticas. Todavia, o procedimento de estimação automática do parâmetro de regularização λ torna esta técnica atraente, visto que necessita de menor intervenção por parte do usuário do que técnicas de controle de complexidade baseadas na teoria de regularização de *Tikhonov*.

Por último, as heurísticas utilizadas para controle de complexidade de MLP's, além de não apresentarem um desenvolvimento teórico que forneça uma base sólida para as mesmas, apresentam também uma série de desvantagens. Para o caso da parada antecipada do treinamento, além das restrições apresentadas em [179], esta metodologia, assim como a validação única, é extremamente dependente da escolha do conjunto de validação. Se este conjunto não for representativo do problema em questão,

a minimização do erro para tal conjunto não resultará em um modelo com boa capacidade de generalização, visto que o desempenho do modelo foi avaliado para um conjunto de dados obtidos de maneira inadequada. Uma das formas de amenizar esta indesejável possibilidade reside na utilização de validação cruzada múltipla. O treinamento com inserção de ruído, por outro lado, demanda um esforço computacional excessivo, visto que o conjunto de treinamento utilizado nesta heurística é pelo menos três vezes maior que o conjunto original de dados, conforme discutido no item 2.2.3.3. Além disso, é necessária a estimação da variância σ^2_{ruido} do ruído a ser inserido nas entradas, constituindo mais um parâmetro a ser estimado no modelo através de técnicas de re-amostragem, por exemplo. A questão da determinação da variância σ^2_{ruido} constitui também uma das principais restrições da aplicação do escalonamento do ganho da função de ativação. Entretanto, visto que esta técnica pode ser vista como um procedimento pós-treinamento a ser aplicado a MLP's com o intuito de aumentar a capacidade de generalização dos modelos estimados, não requerendo para tal um elevado esforço computacional, para problemas onde o número de dados para treinamento não constitui uma restrição, a aplicação desta técnica é interessante, com a variância σ^2_{ruido} podendo ser estimada através de técnicas de re-amostragem.

Em função da ponderação entre as principais vantagens e desvantagens de cada uma das metodologias expostas neste capítulo, esta tese utiliza como procedimento de regularização dos MLP's desenvolvidos o treinamento *bayesiano* e o escalonamento do ganho da função de ativação, com a comparação de modelos através de validação única sendo utilizada para determinação do número adequado de neurônios na camada intermediária. Esta metodologia de comparação de modelos também é utilizada para determinação dos parâmetros que definem a SVM, conforme será exposto no capítulo 3.

2.3 Comparação entre o MLP e a SVM

De uma maneira geral, o MLP pode ser visto como um tipo específico de SVM, visto que as funções sigmoidais atendem as condições do teorema de *Mercer* para valores específicos dos parâmetros β_0 e β_1, fazendo com que os neurônios da camada escondida de um MLP contendo uma única camada oculta desempenhem o papel do *kernel* $K(\underline{x}, \underline{x}_k)$. Entretanto, esta pode ser vista como a única semelhança entre o MLP e a SVM.

A primeira diferença entre MLP's e SVM's reside na definição da estrutura dos modelos. Enquanto que, para o MLP, o número de camadas escondidas e o número de neurônios em cada camada deve ser definido antes da etapa de treinamento, a estrutura da SVM é obtida como um subproduto do treinamento deste modelo. Definido o tipo de *kernel* $K(\underline{x}, \underline{x}_k)$ e os parâmetros que o definem, assim como as constantes ε e C, a estrutura é determinada ao final do algoritmo de treinamento, através do número de vetores suporte obtidos.

Outra questão que merece destaque diz respeito à superfície a ser otimizada ao longo do treinamento destes modelos. Enquanto que para o MLP esta superfície é extremamente não-convexa, repleta de mínimos locais, a superfície a ser otimizada ao longo do treinamento de uma SVM é quadrática, apresentando um único ponto de máximo. Portanto, enquanto que, para um mesmo conjunto de dados, o treinamento de MLP's apresenta elevada dependência do ponto inicial do algoritmo, o treinamento de SVM resulta em uma única solução, mantidos constantes o tipo de *kernel* e seus parâmetros, e as constantes ε e C.

Entretanto, a principal diferença entre estes dois modelos *feedforward* reside no objetivo do treinamento de cada modelo. Enquanto que o treinamento de MLP's está baseado no princípio da minimização do risco empírico, que busca a minimização única

e exclusiva do erro para o conjunto de dados disponível para treinamento, o treinamento de SVM's encontra fundamento no princípio da minimização do risco estrutural, cujo principal objetivo reside na minimização do limite superior do erro de generalização. O princípio da minimização do risco estrutural parte do pressuposto que o erro para um conjunto independente de dados, ou seja, um conjunto de dados não utilizados para treinamento do modelo, é limitado superiormente pela soma entre o erro para o conjunto de treinamento e uma parcela relacionada com a dimensão de *Vapnik-Chervonenkis*, também conhecida como dimensão *V-C* [151]. Visto que a dimensão *V-C* é proporcional à complexidade do modelo, ao utilizar o princípio da minimização do risco estrutural, a SVM, além de buscar o ajuste dos dados de treinamento, apresenta na sua essência uma forma de controle de complexidade do modelo estimado, gerando um modelo intrinsecamente regularizado. Vale ressaltar que, apesar da dimensão *V-C* estar relacionada com a complexidade do modelo, esta grandeza não está diretamente associada com a dimensionalidade do espaço de entrada. Em outras palavras, a dimensão *V-C*, ao contrário da grande maioria dos índices de complexidade de modelos, não aumenta necessariamente em função do número de entradas do modelo, permitindo a abordagem de problemas de elevada dimensionalidade do espaço de representação. Além disso, o objetivo do desenvolvimento de uma máquina de aprendizagem reside na capacidade desta apresentar respostas satisfatórias para padrões ainda não apresentados ao modelo. Mantendo a analogia biológica, é esperado que a máquina de aprendizagem "aprenda" o mapeamento entrada-saída, e não "decore" este mapeamento. Intuitivamente, a minimização do risco estrutural objetiva o "aprendizado", visto que minimiza o limite superior do erro para novos padrões. O mesmo não pode ser dito da minimização do risco empírico.

3

Casos Estudados

Este capítulo apresentará uma descrição dos casos estudados nesta tese, sendo apresentadas as séries de carga e temperatura utilizadas, algumas de suas características e os processamentos efetuados a cada uma delas. Além disso, serão definidos os tipos de modelos utilizados. Visto que os MLP's utilizados nesta tese possuem uma única camada escondida e uma única saída linear, serão definidos neste capítulo o tipo de função de ativação dos neurônios da camada oculta, os algoritmos de treinamento utilizados e os procedimentos para determinação do número de neurônios na camada escondida. Para as SVM's, serão definidos o tipo de função de perda com tolerância ε utilizada, o tipo de *kernel* aplicado e a metodologia utilizada para estimação dos parâmetros que definem o *kernel* e das constantes C e ε que definem a SVM. Além das estruturas, serem apresentadas as metodologias utilizadas para determinação dos conjuntos de treinamento, validação e teste, assim como as medidas de desempenho utilizadas para avaliação dos modelos.

Antes da descrição das séries, vale ressaltar que nesta tese serão consideradas previsões de carga para o horizonte de curto prazo aquelas realizadas para o horizonte definido no item 1.1, ou seja, previsões realizadas para horizontes variando de uma hora a até um mês à frente, discretizadas em base horária, diária ou semanal.

Outra questão que merece ser salientada diz respeito à definição dos modelos utilizados. Visto que esta tese busca a comparação entre metodologias e não o desenvolvimento de um sistema completo de previsão de carga a curto prazo, as estruturas a serem utilizadas foram definidas buscando este objetivo. Exemplificando, para o caso da previsão da curva de carga diária, em base horária, faria mais sentido prático a utilização de 24 modelos, um para cada hora do dia, ou de um modelo para previsão da carga horária 24 passos à frente, do que a utilização de um modelo para

previsão da carga horária um passo à frente, com as previsões para as 23 horas restantes sendo obtidas através da realimentação das entradas deste modelo com as previsões realizadas para as horas anteriores. Entretanto, para fins de comparação entre modelos, a utilização de realimentação é interessante, visto que modelos instáveis divergem para poucos passos à frente, transformando o desempenho para previsões até 24 passos à frente em um bom indicativo da aplicabilidade da metodologia proposta.

3.1 Dados utilizados

Neste item, serão apresentadas as séries de carga e temperatura utilizadas nesta tese, assim como os processamentos efetuados a cada uma delas antes da apresentação destas aos modelos neurais.

Nesta tese, são utilizadas duas bases de dados para verificação da aplicabilidade das metodologias propostas, com ambas objetivando horizontes de previsão diferentes. A primeira base de dados, amplamemte utilizada na literatura, diz respeito aos dados de carga e temperatura da *Puget Sound Power and Light Company*, empresa de energia de Seattle, EUA. Os modelos desenvolvidos para esta série objetivam a previsão da carga horária um passo à frente, com estes valores sendo utilizados para previsão da curva de carga diária, em base horária. A segunda base de dados está relacionada com os dados de carga e temperatura de uma concessionária européia, a qual foi utilizada em uma competição promovida em 2001 pelo *European Network on Intelligent Technologies for Smart Adaptive Systems*, popularmente conhecido pela sigla EUNITE. Para esta base de dados, os modelos desenvolvidos objetivam a previsão do pico de carga diário um passo à frente, com estas previsões sendo utilizadas para previsão dos 31 picos de carga verificados em janeiro de 1999. Estas bases de dados serão descritas com maior riqueza de detalhes nos próximos itens.

3.1.1 Previsão da carga horária

Para desenvolvimento do modelo de previsão de carga horária, foram utilizadas as séries de carga e temperatura da *Puget Sound Power and Light Company*, base de dados de domínio público, a qual pode ser obtida no endereço eletrônico http://www.ee.washington.edu/class/559/2002spr . Estas séries apresentam informações de carga, em [MW], e temperatura, em [°F], em base horária, cobrindo o período de 1º. de janeiro de 1985 a 12 de outubro de 1992, totalizando 68208 dados horários de carga e temperatura. Entretanto, para o desenvolvimento e avaliação dos modelos desenvolvidos, serão utilizados apenas os dados referentes ao período de 1º. de outubro de 1990 a 31 de dezembro de 1991.

O primeiro processamento efetuado a estas séries está relacionado com os chamados dados faltantes, *missing data*, visto que foi verificada a presença de valores nulos isolados tanto de carga quanto de temperatura. Para correção deste problema, estes valores nulos foram substituídos pelas respectivas médias aritméticas entre o valor anterior ao dado faltante e o valor posterior a este. Matematicamente, se $Q(k)=0$, este valor foi substituído por:

$$Q(k)=\frac{Q(k-1)+Q(k+1)}{2} \tag{3.1}$$

Conforme descrito no item 1.1, a dinâmica de curto prazo da carga é fortemente influenciada pelo período do ano, em função das condições climáticas relacionadas com as estações do ano, pelo dia da semana, em virtude da diferença do padrão de consumo entre dias úteis e finais de semana, e pela hora do dia, refletindo as atividades diárias da população da área de atendimento. A influência da época do ano no comportamento da carga a curto prazo pode ser verificada na Figura 3.1, onde a curva verde representa a curva de carga semanal, em base horária, para a primeira semana de janeiro de 1990, e a

curva vermelha representa a mesma curva de carga para a primeira semana de julho de 1990. Nesta figura, a diferença do comportamento da carga no inverno e no verão é nítida, com a estação do ano influenciando tanto o pico de carga diário quanto o consumo médio diário, conforme evidenciado na Figura 3.1. Esta influência será tratada através da utilização de informações climáticas, obtidas da série de temperatura, em conjunto com a utilização apenas de padrões referentes à mesma época do ano para treinamento dos modelos, conforme será apresentado ao longo deste capítulo.

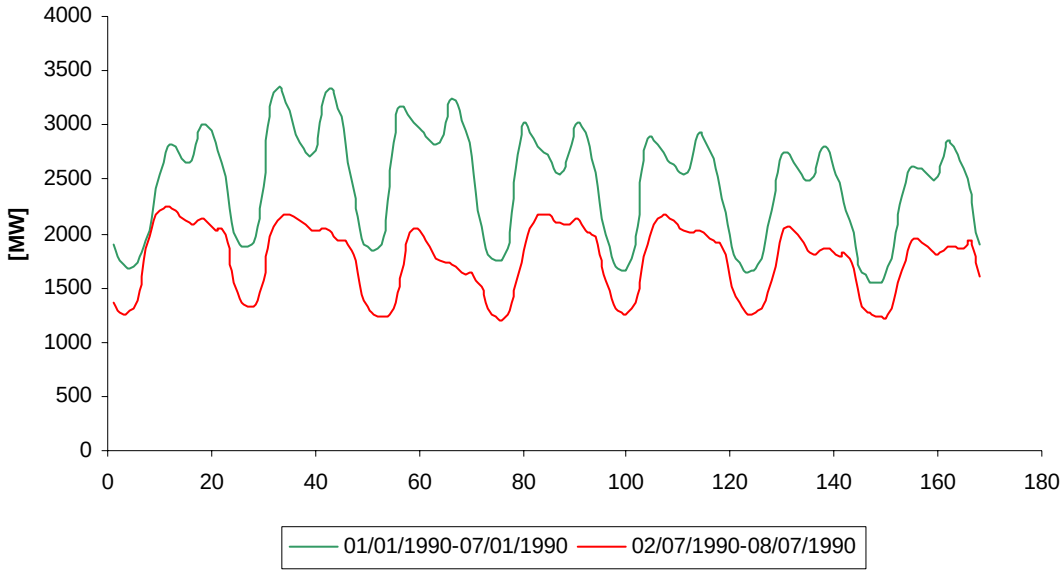

Figura 3.1 – Ilustração da influência da estação do ano no comportamento da carga

A influência do dia da semana e da hora do dia na dinâmica de curto prazo da carga pode ser verificada na Figura 3.2, onde a curva verde representa a curva de carga semanal, em base horária, para a primeira semana de janeiro de 1990, e a curva vermelha representa a mesma curva de carga para a segunda semana de janeiro de 1990. Na Figura 3.2, pode ser verificado que, para uma mesma época do ano, a curva de carga apresenta dois padrões, um semanal, representando a influência do dia da semana, e outro diário, representando a dependência da hora do dia, refletindo, assim, a presença de sazonalidade semanal e diária, respectivamente.

Com o intuito de tratar as sazonalidades existentes na série de carga, objetivando também a retirada de uma possível tendência de crescimento na série, a qual pode surgir em virtude do crescimento no consumo de energia elétrica da região de interesse, foram aplicadas as seguintes transformações à série de carga:

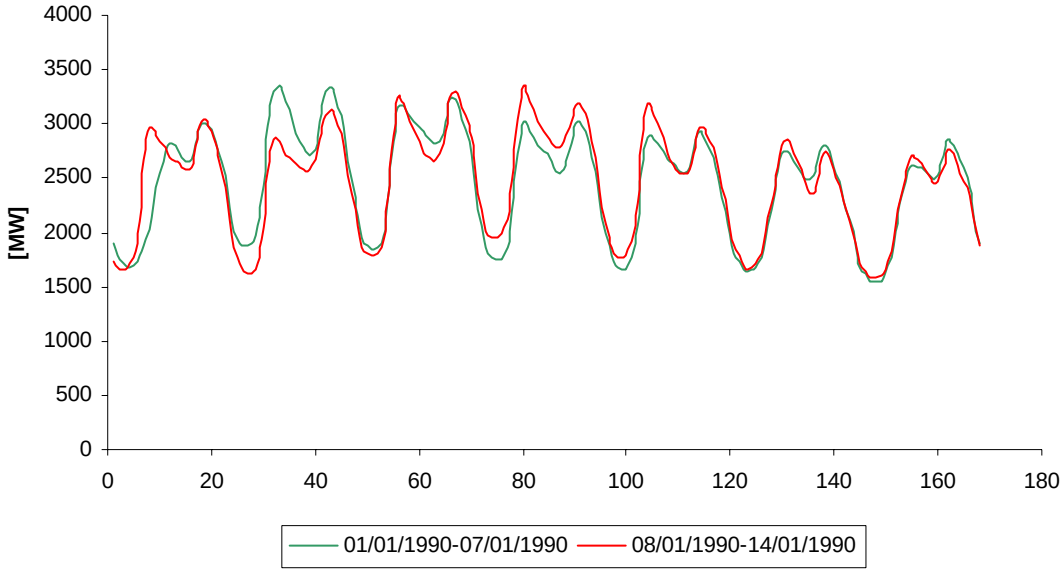

Figura 3.2 – Ilustração dos padrões semanal e diário existentes na série de carga

$$S_1(k) = S(k) - S(k-1) \tag{3.2}$$

$$S_2(k) = S_1(k) - S_1(k-24) \tag{3.3}$$

$$S_3(k) = S_2(k) - S_2(k-168) \tag{3.4}$$

A primeira transformação aplicada à série de carga original $S(k)$, dada pela equação (3.2), foi efetuada com o intuito da retirada da possível tendência local de crescimento desta série, dando origem à série $S_1(k)$. A segunda transformação, aplicada agora à série $S_1(k)$ e dada pela equação (3.3), foi realizada objetivando a retirada da sazonalidade diária, dando origem à série $S_2(k)$. Por último, a transformação dada pela equação (3.4), aplicada à série $S_2(k)$, foi efetuada com o

objetivo de retirada da sazonalidade semanal, dando origem à série $S_3(k)$. As transformações dadas pelas equações (3.3) e (3.4) são conhecidas como diferenças sazonais. Vale ressaltar que a aplicação destas transformações, sugerida em diversos trabalhos encontrados na literatura, não garante necessariamente a retirada da tendência e da sazonalidade da série original, podendo inclusive inserir novas componentes indesejáveis no sinal processado. Procedimentos mais sofisticadas para tratamento de sazonalidades e retirada de tendências podem ser encontrados em [182].

Após o tratamento das sazonalidades e da tendência da série de carga $S(k)$, resultando na série $S_3(k)$, esta última foi reduzida, ou seja, foi aplicada uma transformação a esta série com o intuito de tornar sua média nula e sua variância unitária. Esta transformação é dada por:

$$S_4(k) = \frac{S_3(k) - \hat{\mu}\left[S_3(k)\right]}{\hat{\sigma}\left[S_3(k)\right]} \tag{3.5}$$

$$\hat{\mu}\left[S_3(k)\right] = \frac{1}{P}\sum_{k=1}^{P} S_3(k)$$

$$\hat{\sigma}\left[S_3(k)\right] = \sqrt{\frac{1}{P-1}\sum_{k=1}^{P}\left\{S_3(k) - \mu\left[S_3(k)\right]\right\}^2}$$

Na equação (3.5), P representa o número de pontos da série $S_3(k)$, $\hat{\mu}\left[S_3(k)\right]$ representa a média amostral desta série, e $\hat{\sigma}\left[S_3(k)\right]$ o desvio padrão amostral.

Por último, com o intuito de melhorar o desempenho do treinamento dos modelos, a série $S_4(k)$ foi normalizada no intervalo $[-1;1]$, dando origem à série final $S_5(k)$, através da aplicação da seguinte transformação:

$$S_5(k) = 2\frac{\left\{S_4(k) - \min\left[S_4(k)\right]\right\}}{\max\left[S_4(k)\right] - \min\left[S_4(k)\right]} - 1 \tag{3.6}$$

Na equação (3.6), $\min\left[S_4(k)\right]$ e $\max\left[S_4(k)\right]$ representam os valores máximos e mínimos da série $S_4(k)$.

Vale ressaltar que as transformações de redução e normalização da série $S_4(k)$ foram realizadas utilizando apenas os dados do conjunto de treinamento, conforme será detalhado nos próximos itens.

Efetuado o processamento da série de carga, é necessária neste momento a descrição do tratamento efetuado à série de temperatura $T(k)$, que será utilizada para inserção de informações climáticas. Conforme é de conhecimento amplo, um dos principais fatores que influem no comportamento das séries de temperatura é a hora do dia. A título de ilustração, a Figura 3.3 apresenta a curva de temperatura semanal, em base horária, para as duas primeiras semanas de janeiro de 1990. Conforme pode ser verificado neste gráfico, diferentemente da série de carga, a série de temperatura não apresenta padrão semanal, ou seja, o comportamento de curto prazo da temperatura não está relacionado com o dia da semana. Entretanto, da mesma forma que a série de carga, a dinâmica de curto prazo da temperatura está diretamente relacionada com a hora do dia, conforme verificado na Figura 3.4, que mostra as curvas de temperatura para as quatro primeiras horas do dia, para os primeiros 14 dias de janeiro. Exemplificando, a curva verde mostra a temperatura medida na primeira hora dos primeiros 14 dias de janeiro. Este fato evidencia a presença de sazonalidade diária na série de temperatura.

Portanto, com o intuito de retirada da sazonalidade diária presente na série de temperatura, foi aplicada a esta série a mesma transformação aplicada à série de carga para tratamento desta sazonalidade, dada pela equação (3.3), dando origem à série $T_1(k)$. Posteriormente, assim como foi feito para a série de carga, a série de temperatura foi reduzida e normalizada, utilizando as equações (3.5) e (3.6). Vale

lembrar que este procedimento de redução e normalização foi realizado utilizando apenas o conjunto de treinamento, da mesma forma que o efetuado para a série de carga.

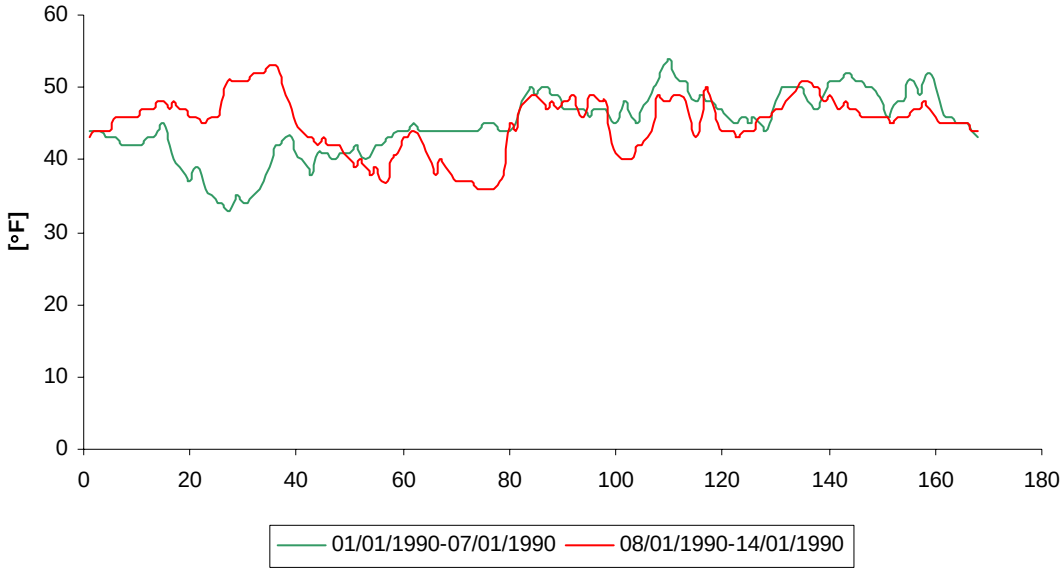

Figura 3.3 – Curvas de temperatura semanal para as duas primeiras semanas de janeiro de 1990

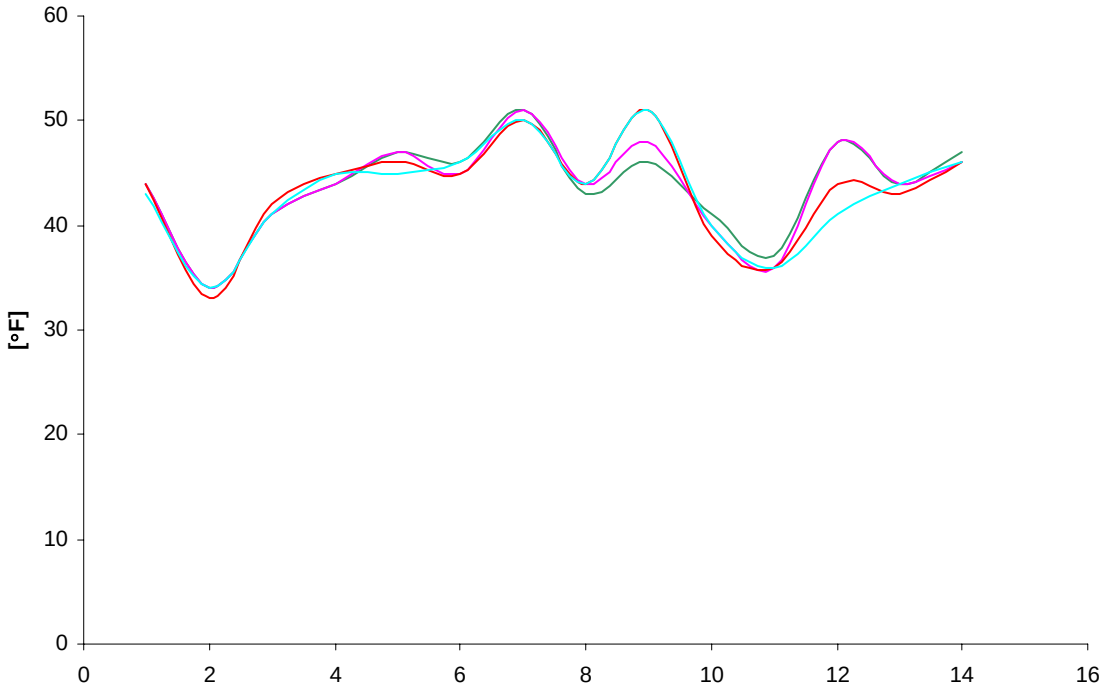

Figura 3.4 – Ilustração da influência da hora do dia na temperatura

Processadas as séries de carga e temperatura, o problema reside na determinação do conjunto de entradas a serem utilizadas pelos modelos. Em outras palavras, devem ser determinados os atrasos das séries processadas de carga e temperatura que devem ser utilizados para modelagem da carga horária. Para modelos lineares, a teoria de análise de séries temporais sugere que o estudo das funções de autocorrelação, autocorrelação parcial e correlação cruzada podem ser utilizadas como indicativos dos atrasos a serem utilizados, conforme apresentado em [182]. Entretanto, visto que nesta tese serão utilizados apenas modelos não-lineares, a seleção de variáveis de entrada destes modelos a partir da análise das funções de autocorrelação não é o procedimento mais indicado. Na realidade, o procedimento de escolha de variáveis de entrada de modelos neurais é um problema ainda em aberto, podendo ser encontradas na literatura diversas metodologias para abordagem deste problema. Exemplos de metodologias de seleção de entradas de modelos neurais podem ser encontrados em [116], [135], [165], [169] - [171].

Portanto, visto que a solução do problema de seleção de variáveis de entrada de modelos não-lineares foge ao escopo desta tese, neste trabalho serão utilizadas as mesmas entradas propostas em [183], onde a mesma série de carga e temperatura é utilizada. Portanto, para modelagem da série de carga processada $S_3(k)$, serão utilizados como entradas os atrasos $S_3(k-1)$, $S_3(k-2)$, $S_3(k-24)$ e $S_3(k-168)$, referentes à série $S_3(k)$, juntamente com estes mesmos atrasos para a série de temperatura processada $T_1(k)$, ou seja, $T_1(k-1)$, $T_1(k-2)$, $T_1(k-24)$ e $T_1(k-168)$, juntamente com o valor atual $T_1(k)$ desta série. Portanto, serão utilizadas quatro entradas relacionadas com valores atrasados da série de carga processada e cinco entradas relacionadas com valores atrasados, e com o valor atual, da série de

temperatura. Vale ressaltar que, para as previsões até 24 passos à frente, serão utilizados os valores medidos, e não previstos, da série de temperatura, com a ocorrência de realimentação sendo restrita às entradas de carga. Na prática, este procedimento não é o correto, visto que estas informações de temperatura não estão disponibilizadas no instante de previsão. Entretanto, conforme citado no início deste capítulo, o objetivo desta tese reside na comparação entre metodologias, e não no desenvolvimento de um sistema completo de previsão de carga. Portanto, neste contexto, a utilização dos valores verificados de temperatura faz sentido, não constituindo uma restrição para comparação das metodologias, visto que todas utilizarão o mesmo conjunto de entradas.

Além destas nove entradas, serão utilizadas duas entradas adicionais, com o intuito de codificar a hora do dia na qual será realizada a previsão, dadas pelas equações [133] e [183]:

$$HS(k) = sen\left(\frac{2\pi k}{24}\right),\ k = 1, 2, ..., 24 \tag{3.7}$$

$$HC(k) = \cos\left(\frac{2\pi k}{24}\right),\ k = 1, 2, ..., 24 \tag{3.8}$$

Desta forma, os padrões entrada-saída serão definidos da forma apresentada na Figura 3.5.

Entradas de carga				Entradas de temperatura					Hora do dia		Saída
$S_3(k-1)$	$S_3(k-2)$	$S_3(k-24)$	$S_3(k-168)$	$T_1(k)$	$T_1(k-1)$	$T_1(k-2)$	$T_1(k-24)$	$T_1(k-168)$	$HS(k)$	$HC(k)$	$S_3(k)$

Figura 3.5 – Formato dos padrões entrada-saída utilizados pelos modelos de previsão da carga horária

Definido o conjunto de entradas utilizadas, resta agora a definição do conjunto de treinamento a ser utilizado pelos modelos. Apesar dos modelos desenvolvidos tratarem da modelagem da carga horária um passo à frente, estes modelos deverão fazer

previsões da curva de carga diária, em base horária, através da realimentação das entradas de carga.

Visto que o comportamento de curto prazo da carga está diretamente relacionado com o dia da semana, são desenvolvidos sete modelos, um para cada dia da semana. Neste ponto, deve ser destacado que não serão realizadas previsões para feriados, visto que um tratamento especial deve ser dedicado a estes dias, o que foge ao escopo desta tese. Além do desenvolvimento de um modelo para cada dia da semana, em virtude da dependência da carga em relação à estação do ano, para treinamento destes modelos serão utilizados apenas padrões referentes ao mesmo período do ano. Mais especificamente, para treinamento de cada um dos sete modelos, serão utilizados apenas os padrões referentes às últimas seis semanas, para o respectivo dia da semana, totalizando 144 padrões para treinamento. Além disso, visto que os modelos desenvolvidos deverão realizar previsões da curva de carga diária, em base horária, através da realimentação das entradas de carga, estes serão treinados uma vez por dia. Exemplificando, para previsão da curva de carga do dia 1º. de fevereiro de 1990, quinta-feira, os modelos são treinados utilizando as últimas seis quintas-feiras existentes na base de dados, desconsiderando feriados, sendo realizadas posteriormente as previsões 24 passos à frente para este dia, sendo gerada a curva de carga prevista. Vale ressaltar que são utilizados como entradas os respectivos valores atrasados da série de carga medidos no dia 31 de janeiro de 1990, conforme ilustrado na Figura 3.5, sendo posteriormente efetuada a realimentação dos valores previstos. Posteriormente, para previsão da curva de carga diária referente a 8 de fevereiro de 1990, os 24 valores de carga medidos no dia 1º. de janeiro são incorporados à base de dados, com o modelo sendo treinado novamente para previsão da curva de carga do dia em estudo. A Figura 3.6 ilustra a construção do conjunto de treinamento para o caso utilizado como exemplo.

Desta forma, o procedimento geral utilizado para desenvolvimento dos modelos para previsão da curva de carga diária, em base horária, pode ser resumido no diagrama apresentado na Figura 3.7. Visto que os modelos desenvolvidos realizam previsões sobre a série processada $S_3(k)$, o bloco intitulado pós-processamento na Figura 3.7 é responsável pela aplicação das transformações inversas das efetuadas à série de carga, com o intuito de obter a previsão da curva de carga diária, em base horária, na escala original, ou seja, em [MW].

CONJUNTO DE TREINAMENTO						PREVISÃO
21/12/1989	28/12/1989	4/1/1990	11/1/1990	18/1/1990	25/1/1990	1/2/1990
24 CARGAS	24 CARGAS	24 CARGAS	24 CARGAS	24 CARGAS	24 CARGAS	24 CARGAS

	CONJUNTO DE TREINAMENTO						PREVISÃO
	28/12/1989	4/1/1990	11/1/1990	18/1/1990	25/1/1990	1/2/1990	8/2/1990
	24 CARGAS	24 CARGAS	24 CARGAS	24 CARGAS	24 CARGAS	24 CARGAS	24 CARGAS

Figura 3.6 – Ilustração da formação do conjunto de treinamento

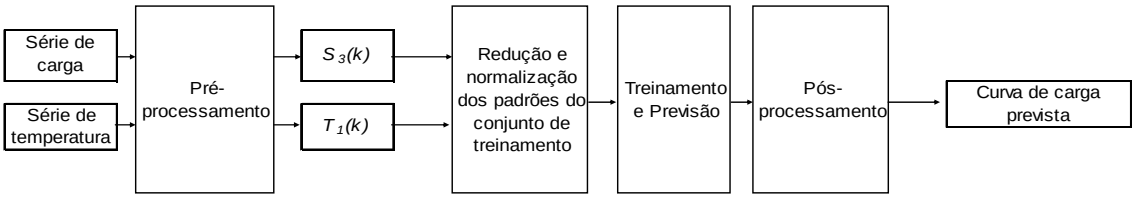

Figura 3.7 – Diagrama esquemático da metodologia utilizada para previsão da curva de carga diária.

As diferentes estruturas utilizadas para cada dia da semana, assim como os critérios de avaliação de desempenho, serão apresentadas ao longo deste capítulo.

3.1.2 Previsão do pico de carga diário

Os modelos para previsão do pico de carga diário foram desenvolvidos utilizando os dados de carga e temperatura de uma empresa de energia européia, chamada *East-Slovakia Power Distribution Company*, disponibilizados pelo *European Network on Intelligent Technologies for Smart Adaptive Systems*, popularmente conhecido pela sigla EUNITE, para realização de uma competição entre modelos de

previsão em 2001. Este conjunto de dados pode ser encontrado no site da competição, cujo endereço é http://neuron.tuke.sk/competition. Basicamente, esta base de dados apresenta valores de carga, em [MW], coletados a cada trinta minutos, para o período de 1º. de janeiro de 1997 a 31 de dezembro de 1998, em conjunto com valores médios diários de temperatura, em [°C], cobrindo o mesmo período da série de carga. Na competição realizada em 2001, a tarefa dos competidores residia no desenvolvimento de modelos para previsão do pico de carga diário para todo o mês de janeiro de 1999. Portanto, uma forma interessante de avaliar as metodologias propostas nesta tese reside na comparação entre os resultados obtidos pelos modelos desenvolvidos neste trabalho e aqueles obtidos pelos competidores em 2001.

Neste contexto, visto que não são disponibilizadas *a priori* informações de temperatura para o período desejado de previsão, ou seja, para janeiro de 1999, a única forma de inserir esta informação no modelo de previsão do pico de carga diário reside no desenvolvimento de uma estrutura para previsão da temperatura. Entretanto, além da complexidade exigida no desenvolvimento de modelos de previsão de temperatura, estes carregam consigo um elevado nível de incerteza. Modelos de previsão de temperatura para utilização desta informação para previsão de carga podem ser encontrados em [5], [109] e [125]. Portanto, visto que a utilização de uma entrada cujo nível de incerteza é elevado pode comprometer o desempenho do modelo de previsão de carga, as informações de temperatura não serão utilizadas como entradas dos modelos de previsão do pico de carga diário a serem desenvolvidos nesta tese.

Outra questão diz respeito à base de discretização da série de carga a ser utilizada. Apesar dos dados disponíveis estarão discretizados em base de trinta minutos, este nível de discretização não apresenta vantagem para a previsão do pico de carga diário, visto que as informações em base de trinta minutos não estarão disponíveis para

o período de previsão desejado. Portanto, a partir da base de dados de carga original, será obtida uma série de pico de carga diário, através da coleta do valor máximo medido para cada dia. As séries de pico de carga diário obtidas estão apresentadas na Figura 3.8. Nesta figura, a curva vermelha representa a série para o ano de 1997, com a curva verde fazendo referência à série obtida para o ano de 1998.

Conforme pode ser verificado na Figura 3.8, a série de pico de carga diário apresenta um padrão anual, sugerindo a presença de sazonalidade anual. Entretanto, visto que são disponibilizados apenas dois anos de dados, a afirmativa de que existe uma padrão sazonal anual nesta série é arriscada, visto que existem poucos padrões para confirmação desta afirmação. Além disso, se esta afirmativa for verdadeira, a aplicação da diferença sazonal de maneria análoga à utilizada no item 3.1.1 acarretará na perda dos dados para o ano de 1997, fazendo com que apenas os dados referentes a 1998 sejam utilizados para desenvolvimento dos modelos.

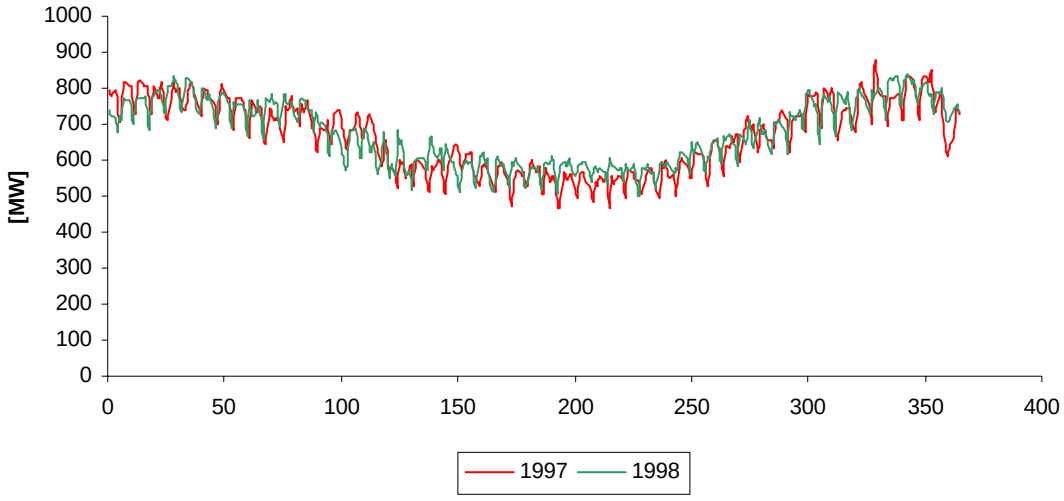

Figura 3.8 – Séries de pico de carga diário para 1997 e 1998

Apesar da existência de outras técnicas para tratamento de sazonalidades, as quais não requerem a perda de dados resultante da aplicação da diferença sazonal, a exigência de tratamento específico deste tipo de comportamento sistemático verificado

em algumas séries temporais, com o intuito de tornar a série o mais estacionária possível, constitui uma característica dos modelos lineares de análise de séries temporais, como o SARIMA e suas variantes. Entretanto, a aplicação de modelos não-lineares na previsão de séries temporais não apresenta esta restrição, com o comportamento sistemático representado pelas sazonalidades sendo tratado automaticamente pelo modelo não-linear. Em virtude disto e da ausência de dados faltantes para esta série, nenhum processamento será efetuado à série de pico de carga diário, com exceção da normalização e padronização desta série, de maneira análoga à realizada para a série de carga horária.

Definida a série a ser utilizada, resta agora a seleção das variáveis de entrada e do conjunto de treinamento a ser utilizado. Assim como a carga horária, o comportamento do pico de carga diário está diretamente relacionado com as estações do ano e com o dia da semana, pelos mesmos motivos citados no item 3.1.1. A dependência o pico de carga em relação à época do ano será tratada de maneira análoga à utilizada para previsão da carga horária, ou seja, através da seleção dos padrões a serem utilizados para treinamento. Por outro lado, a relação entre o pico de carga e o dia da semana será abordada de maneira diferente.

Em virtude da escassez de dados disponíveis para treinamento, a utilização de um modelo para cada dia da semana é inviável, visto que este procedimento reduzirá ainda mais o número de dados para treinamento de cada modelo, originalmente igual a N e passando para $N/7$ seguindo esta abordagem. Para superar esta restrição relacionada com o número de dados, para a previsão do pico de carga diário, o dia da semana será utilizado como entrada do modelo. Esta informação será codificada no formato conhecido como 1 de *n*, que é a forma mais recomendada de representação de informações que não possuem relação de ordem. Portanto, serão utilizadas sete entradas

binárias, cada uma associada a um dia da semana, sendo atribuído o valor unitário para a entrada referente ao dia da semana associado à saída, com as entradas binárias restantes apresentando valor nulo. Além destas variáveis binárias, será utilizada mais uma variável deste tipo para sinalização da ocorrência de feriados.

Juntamente com as entradas responsáveis pela codificação do dia da semana, devem ser selecionados os atrasos da série de pico de carga que devem ser utilizados pelo modelo. Visto que, para esta base de dados, não foi encontrada nenhuma referência na literatura sugerindo quais atrasos utilizar como entrada, tendo em mente que este procedimento não é o mais indicado para seleção de variáveis de entrada de modelos não-lineares, um indicativo sobre quais atrasos devem ser utilizados pode ser obtido através da análise da função de autocorrelação parcial amostral da série de pico de carga. Utilizando a função *parcorr* do software *Matlab 6.5*, pode ser obtida uma estimativa da função de autocorrelação parcial, cujo gráfico está apresentado na Figura 3.9. Nesta figura, as linhas azuis determinam o nível de relevância dos índices de autocorrelação parcial amostral associados a cada atraso, com significância de 95 %. Matematicamente, a estimativa da função de autocorrelação amostral da série $S(t)$ para o *k*-ésimo atraso é dada por [182]:

$$\rho(k) = \frac{\sum_{t=1}^{N-k} \left\{ S(t) - \mu\left[S(t)\right] \right\} \left\{ S(t+k) - \mu\left[S(t)\right] \right\}}{\sum_{t=1}^{N} \left\{ S(t) - \mu\left[S(t)\right] \right\}^2} \tag{3.9}$$

Os coeficientes $\rho(k)$ são necessários para estimativa dos coeficientes $\alpha(k)$ da função de autocorrelação parcial amostral, os quais podem ser obtidos através da solução das equações de *Yule-Walker*, dadas por [182]:

$$\begin{bmatrix} 1 & \rho(1) & \cdots & \rho(k-1) \\ \rho(1) & 1 & \cdots & \rho(k-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(k-1) & \rho(k-2) & \cdots & 1 \end{bmatrix} \begin{bmatrix} \alpha(1) \\ \alpha(2) \\ \vdots \\ \alpha(k) \end{bmatrix} = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(k) \end{bmatrix} \tag{3.10}$$

A análise do gráfico apresentado na Figura 3.9 permite algumas interpretações. O elevado índice de autocorrelação parcial amostral registrado para o oitavo atrasado evidencia a relação existente entre o pico de carga diário e o dia da semana. Além disso, já que os limiares de relevância apresentados nesta figura foram obtidos para um nível de 95 % de significância, e visto que, para atrasos maiores que o oitavo, foram obtidos índices de autocorrelação parcial amostral próximos, ou na grande maioria dentro, da banda definida pelos níveis de relevância, podemos concluir que os atrasos maiores que o oitavo não apresentam correlação com o pico de carga.

Portanto, a análise da função de autocorrelação parcial amostral sugere a utilização dos últimos oito valores de pico de carga registrados como entrada do modelo. Entretanto, conforme assinalado acima, o oitavo atraso está relacionado com a dependência existente entre o pico de carga diário e o dia da semana, a qual já está sendo abordada pelas entradas binárias responsáveis pela codificação do dia da semana. Portanto, sob um certo aspecto, a informação contida no oitavo atraso é redundante, podendo, assim, ser desconsiderada. Desta forma, serão utilizados como entradas dos modelos de previsão do pico de carga os valores registrados dos picos para os últimos sete dias. O formato utilizado para o padrão entrada-saída dos modelos desenvolvidos para modelagem do pico de carga é apresentado na Figura 3.10.

Definido o conjunto de entradas a serem utilizadas pelos modelos desenvolvidos para previsão do pico de carga diário, resta neste ponto a definição do conjunto de treinamento utilizado. Em virtude da influência das estações do ano na dinâmica do pico de carga diário, serão utilizados para treinamento apenas dados referentes à estação do

ano onde serão realizadas as previsões. Portanto, visto que os modelos desenvolvidos para modelagem do pico de carga deverão realizar previsões para janeiro, ou seja, no inverno europeu, serão utilizados para treinamento apenas os padrões referentes a esta estação do ano. Em outras palavras, serão utilizados apenas os padrões referentes aos meses de janeiro a março e de outubro a dezembro.

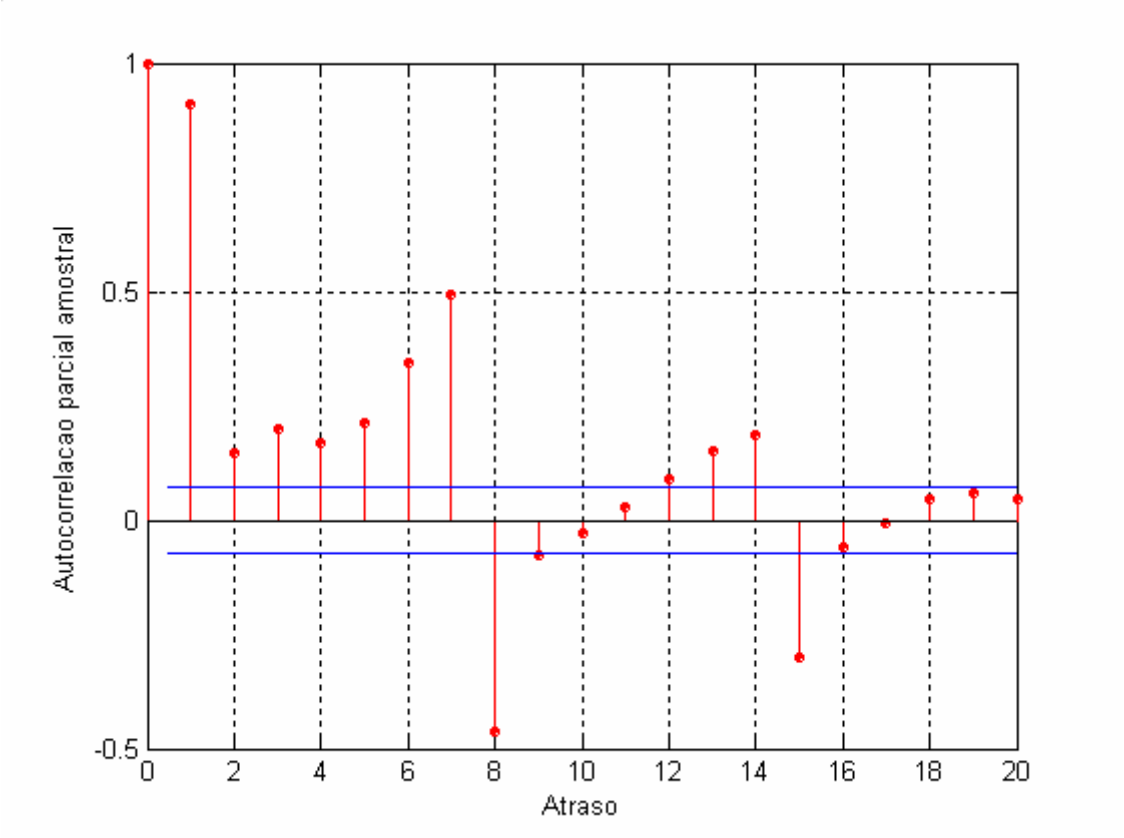

Figura 3.9 – Função de autocorrelação parcial amostral da série de pico de carga

Terminada a descrição dos processamentos efetuados às séries utilizadas, com a consequente apresentação dos conjuntos de entrada utilizados e dos procedimentos considerados para definição dos respectivos conjuntos de treinamento, o único ponto de discussão pendente neste capítulo diz respeito à definição das estruturas utilizadas, as quais serão definidas a seguir.

Entradas de carga				Dia da semana				Feriado	Saída
				Seg.	Ter.	...	Dom.		
S(k-1)	*S(k-2)*	*...*	*S(k-7)*	0/1	0/1	...	0/1	0/1	*S(k)*

Figura 3.10 – Formato dos padrões entrada-saída utilizados pelos modelos de previsão do pico de carga diário

3.2 Definição dos modelos utilizados

Definido o conjunto de entradas a serem utilizadas pelos modelos neurais *feedforward*, estes apresentam ainda uma série de parâmetros a serem definidos pelo usuário antes da abordagem efetiva do problema em estudo. Mais especificamente, para o caso do MLP, devem ser especificados: número de camadas ocultas existentes no modelo, número de neurônios por camada, tipo de função de ativação destes neurônios, e o algoritmo de treinamento a ser utilizado, assim como os parâmetros que definem este algoritmo. Para o caso da SVM, devem ser definidos: tipo de função de perda com tolerância ε utilizada, tipo de *kernel* aplicado e a forma na qual os parâmetros que o definem serão estimados, e o procedimento utilizado para estimativa das constantes C e ε que definem a SVM. As metodologias utilizadas para determinação dos parâmetros que definem os dois modelos neurais *feedforward* estudados nesta tese serão apresentadas nos próximos itens.

3.2.1 MLP

O primeiro parâmetro a ser especificado em um MLP diz respeito ao número de camadas escondidas utilizadas pelo modelo. Entretanto, conforme apresentado no item 2, o teorema da aproximação universal afirma que, para problemas de aproximação funcional, onde os casos abordados nesta tese podem ser enquadrados, uma única camada oculta é suficiente para este tipo de aplicação. Portanto, os MLP's utilizados nesta tese, conforme citado anteriormente, apresentarão uma única camada escondida.

Apesar do teorema da aproximação universal garantir que MLP's com uma única camada oculta podem aproximar com precisão arbitrária qualquer função contínua não-linear, este mesmo teorema afirma que, para que os MLP's apresentem esta característica, estes modelos devem apresentar um número suficiente de neurônios na camada escondida. A determinação do número suficiente de neurônios nesta camada constitui o segundo parâmetro a ser especificado pelo usuário. Nesta tese, o procedimento de estabilização de estrutura baseado na comparação de modelos, utilizando a técnica de validação única apresentada no item 2.2.1.1, será utilizado para determinação do número de neurônios na camada oculta dos MLP's desenvolvidos. Para os modelos desenvolvidos para previsão da carga horária, cuja base de dados foi descrita no item 3.1.1, o desempenho dos modelos para o mês de janeiro de 1991 será utilizado para determinação do número de neurônios na camada oculta. Visto que os modelos desenvolvidos para este caso são treinados uma vez por dia, como medida de desempenho será considerada o erro absoluto percentual médio de previsão para este específico mês. Matematicamente, este erro é dado por:

$$E_{janeiro} = \frac{1}{31}\sum_{j=1}^{31} E_{dia_j} \tag{3.11}$$

$$E_{dia_j} = \frac{100}{24}\sum_{k=1}^{24} \frac{\left|S_j(k) - \widehat{S_j}(k)\right|}{S_j(k)}$$

Na equação (3.11), $S_j(k)$ representa a carga verificada na *k*-ésima hora do *j*-ésimo dia de janeiro de 1991, $\widehat{S_j}(k)$ representa a saída prevista pela rede para este mesmo instante, E_{dia_j} representa o erro absoluto percentual médio para o *j*-ésimo dia de janeiro, e $E_{janeiro}$ a medida de desempenho, ou seja, o erro percentual médio obtida para janeiro.

Para os modelos desenvolvidos para previsão do pico de carga diário, cuja base de dados utilizada foi descrita no item 3.1.2, o desempenho dos diferentes modelos para

dezembro de 1998 foi utilizado como medida de desempenho. Neste contexto, a medida de desempenho é dada por:

$$E_{dezembro} = \frac{100}{31} \sum_{j=1}^{31} \frac{\left| S_j - \widehat{S_j} \right|}{S_j} \tag{3.12}$$

Na equação (3.12), S_j representa o pico de carga verificado no *j*-ésimo dia de dezembro de 1998, $\widehat{S_j}$ representa o pico de carga previsto pelo modelo para este mesmo dia, e $E_{dezembro}$ representa o erro absoluto percentual médio obtido para dezembro de 1998, o qual é utilizado como medida de desempenho.

Apresentadas as metodologias utilizadas para determinação do número suficiente de neurônios na camada oculta dos MLP's desenvolvidos nesta tese, é necessária a definição do tipo de função de ativação utilizada por estes neurônios. Nesta tese, os neurônios da camada escondidas dos MLP's apresentarão funções de ativação sigmoidais, mais especificamente a função tangente hiperbólica, dada pela equação (A.3), com o ganho $a = 1$.

Para treinamento dos MLP's, serão utilizados dois algoritmos. Serão comparados o algoritmo de retropropagação de erro por batelada, apresentado no Apêndice A, porém com a inserção de uma parcela de momento na modificação aplicada aos pesos sinápticos apresentada nas equações (A.15) e (A.16), e o treinamento *bayesiano* de MLP's, apresentado no item 2.2.2.3. Estes algoritmos foram implementados através das funções *traingdx* e *trainbr* do software *Matlab 6.5*, com os parâmetros que definem estes algoritmos sendo assumidos como os valores padrão adotados pelo *Matlab*. Como critério de parada do treinamento, foi utilizado o número máximo de épocas (2000 para os modelos de previsão de carga horária e 30000 para os de previsão do pico de carga diário), para ambos algoritmos de treinamento, e o valor máximo do parâmetro λ existente no algoritmo de *Levenberg-Marquardt*, apresentado

no Apêndice A e utilizado pela função *trainbr*. Além destes dois algoritmos, também será utilizado o escalonamento do ganho da função de ativação, heurística de regularização de MLP's descrita no item 2.2.3.3. Visto que esta técnica é aplicada a um MLP previamente treinado, o algoritmo de retropropagação do erro por batelada, com a inclusão da parcela de momento, será utilizado para treinamento, com o escalonamento do ganho da função de ativação sendo aplicado após o término da sessão de treinamento. O valor da variância σ^2_{ruido} será estimado através da aplicação das mesmas técnicas utilizadas para determinação do número suficiente de neurônios, descrita anteriormente neste item.

Especificados os procedimentos utilizados para determinação das estruturas dos MLP's utilizados nesta tese, resta a definição das metodologias utilizadas para especificação da SVM, apresentadas no item seguinte.

3.2.2 SVM

Conforme descrito no item 2.1, uma SVM é definida a partir da escolha da função de perda com tolerância ε, da especificação do tipo de *kernel* utilizado pelo modelo, pela determinação dos parâmetros que definem o *kernel*, e pela especificação das constantes C e ε relacionadas com a função objetivo a ser minimizada ao longo do treinamento e com a função de perda utilizada pelo modelo, respectivamente.

Visto que a teoria apresentada sobre SVM foi fundamentada na utilização da função linear de perda com tolerância ε, dada pela equação (2.3), esta função de perda será utilizada para desenvolvimento das SVM's utilizadas nesta tese.

Definido o tipo de função de perda com tolerância ε utilizada, é necessária a escolha do tipo de *kernel* aplicado. Nesta tese, o tipo de *kernel* $K\left(\underline{x},\underline{x}_k\right)$ utilizado é o chamado *gaussiano*, dado pela equação (2.15). O *kernel* polinomial também foi testado,

apresentando, porém, resultados inferiores aos obtidos utilizando o *kernel gaussiano*. Portanto, para SVM's que utilizam este tipo de *kernel*, devem ser especificados três parâmetros: a largura σ_{kernel} que define o *kernel gaussiano*, e as constantes C e ε que definem a SVM. Para tal, algumas das técnicas de estabilização de estrutura utilizadas para comparação de modelos, descritas no item 2.2, podem ser utilizadas. Nesta tese, para estimativa destes três parâmetros será utilizado o mesmo procedimento adotado para determinação do número suficiente de neurônios na camada oculta dos MLP's e da variância σ^2_{ruido} utilizada para o escalonamento do ganho da função de ativação, descrito no item 3.2.1.

Especificados os valores dos parâmetros σ_{kernel}, C e ε, o algoritmo de treinamento das SVM's descrito no item 2.1 pode ser aplicado para treinamento dos modelos. Conforme evidenciado no item 2.1, o algoritmo de treinamento de SVM's envolve a aplicação de técnicas de programação quadrática para estimação dos pesos $\underline{W}$ e determinação dos vetores suporte. A implementação deste algoritmo foi efetuada através da utilização da *toolbox* de SVM do software *Matlab 6.5*, desenvolvida por *Steve Gunn* e que pode ser encontrada no endereço eletrônico http://www.isis.ecs.soton.ac.uk/resources/svminfo.

3.3 Avaliação final dos modelos

No item anterior, foram descritas as metodologias utilizadas para especificação dos parâmetros que definem as estruturas dos modelos neurais *feedforward* utilizados nesta tese. Entretanto, ainda não foi apresentada a medida de desempenho final adotada nesta tese para comparação entre as metodologias.

Para os modelos desenvolvidos para previsão da carga horária, os quais são utilizados para previsão da curva de carga diária, em base horária, a medida de

desempenho utilizada para comparação final entre os modelos corresponderá ao erro absoluto percentual médio de previsão para o período de 1º. de fevereiro de 1991 a 31 de dezembro de 1991. Matematicamente, esta medida de desempenho é dada por:

$$E_{teste} = \frac{1}{11}\sum_{mes=1}^{11} E_{mes} \tag{3.13}$$

$$E_{mes} = \frac{1}{N_{mes}}\sum_{j=1}^{N_{mes}} E_{dia_j}$$

$$E_{dia_j} = \frac{100}{24}\sum_{k=1}^{24}\frac{\left|S_j(k) - \widehat{S_j}(k)\right|}{S_j(k)}$$

Na equação (3.13), $S_j(k)$ representa a carga verificada na *k*-ésima hora do *j*-ésimo dia de um dado mês, $\widehat{S_j}(k)$ a carga prevista pelo modelo para este mesmo instante, E_{dia_j} o erro percentual médio de previsão verificado para o *j*-ésimo dia deste mesmo mês, N_{mes} o número de dias deste mês, E_{mes} o erro absoluto percentual médio de previsão para este mês, e E_{teste} o erro absoluto percentual médio de previsão obtido para o período de 1º. de fevereiro de 1991 a 31 de dezembro de 1991. Nesta equação, $mes = 1$ equivale a fevereiro e $mes = 11$ a dezembro. Vale lembrar que os modelos desenvolvidos para este caso não realizam previsões para feriados, com estes dias, portanto, sendo desconsiderados no cálculo da medida de desempenho E_{teste}.

Para os modelos desenvolvidos para previsão do pico de carga diário, será utilizada como medida de desempenho o erro absoluto percentual médio de previsão para o mês de janeiro de 1999, com o intuito de comparar os resultados obtidos pelas metodologias utilizadas nesta tese com os obtidos pelos modelos desenvolvidos na competição organizada pela EUNITE em 2001. Neste ponto, vale ressaltar que, em virtude da escassez de dados disponíveis para treinamento dos modelos, após a utilização dos padrões referentes a dezembro de 1998 para validação das diversas

estruturas, estes dados serão incorporados ao conjunto de treinamento utilizado para treinamento dos modelos finais, ou seja, aqueles que realizarão previsões efetivas para janeiro de 1999. Destacado este ponto, a medida de desempenho para avaliação final dos modelos desenvolvidos para previsão do pico de carga diário será dada por:

$$E_{teste} = \frac{100}{31} \sum_{j=1}^{31} \frac{\left| S_j(k) - \widehat{S}_j(k) \right|}{S_j(k)} \tag{3.14}$$

Definidas as medidas de desempenho utilizadas, resta apenas apresentação dos resultados obtidos, assunto principal do próximo capítulo.

4

Resultados

Após a breve descrição realizada no capítulo 3 acerca dos casos estudados nesta tese, onde foram apresentadas algumas características das séries estudadas, as metodologias utilizadas para determinação das estruturas dos modelos desenvolvidos, e as medidas de desempenho utilizadas para avaliação de tais modelos, este capítulo tártara exclusivamente da apresentação dos resultados obtidos para as duas séries estudadas. Estes resultados serão apresentados em termos das estruturas obtidas através da aplicação das metodologias descritas no capítulo 3, sendo apresentadas posteriormente as medidas de desempenho finais associadas a cada um dos modelos desenvolvidos, descritas no item 3.3.

4.1 Previsão da carga horária

Utilizando os procedimentos para determinação dos parâmetros que definem as estruturas dos modelos utilizados nesta tese, descritas no item 3.2, foram obtidas as estruturas listadas na Tabela 4.1, as quais apresentaram o melhor desempenho para janeiro de 1991. Visto que são desenvolvidos sete modelos, um para cada dia da semana, para cada metodologia são apresentadas sete estruturas, uma em cada linha da Tabela 4.1. Nesta tabela, a primeira coluna apresenta o número de neurônios na camada oculta do MLP treinado através da aplicação do algoritmo de retropropagação do erro por batelada, com inserção da parcela de momento. Em outras palavras, nesta coluna são apresentadas as estrutura estimada sem a aplicação de nenhuma técnica de regularização. A segunda coluna apresenta o número de neurônios na camada escondida obtido para os modelos treinados através da aplicação do algoritmo de treinamento *bayesiano*. A terceira coluna apresenta o número de neurônios dos MLP's estimados através da aplicação do escalonamento do ganho da função de ativação, com a quarta coluna apresentando a variância σ^2_{ruido} estimada para cada um dos sete modelos. As

últimas quatro colunas apresentam os parâmetros estimados para a SVM, com a última coluna apresentando o número médio de vetores suporte, apresentados como porcentagem dos dados de treinamento, obtido para os modelos desenvolvidos para o mês de janeiro de 1991. Por exemplo, a SVM estimada para modelagem do comportamento da carga para as quartas-feiras, utilizou, em média, cerca de 41 % dos padrões disponíveis para treinamento como vetores suporte, ou seja, este modelo apresentou, em média, cerca de 59 vetores suporte para janeiro de 1991. Visto que, para problemas de regressão, os vetores suporte são aqueles situados fora da banda definida por ε, o número de vetores suporte obtidos por uma SVM pode ser utilizado como medida da complexidade do modelo estimado. Neste sentido, modelos com complexidade elevada tendem a apresentar um elevado número de vetores suporte.

Tabela 4.1 – Estruturas obtidas a partir dos resultados para janeiro de 1991

	Sem Regularizador	*Bayesiano*	Escalonamento		SVM			
	Neurônios	Neurônios	Neurônios	σ^2_{ruido}	C	ε	σ_{kernel}	NMVS
Segunda	2	2	2	0.12	0.1	0.100	4.24	39.79
Terça	2	2	2	0.16	0.1	0.001	4.24	97.61
Quarta	3	2	3	0.07	1.0	0.100	2.72	40.76
Quinta	3	2	2	0.17	1.0	0.400	1.96	8.889
Sexta	2	2	2	0.12	1.0	0.400	3.48	8.125
Sábado	4	2	4	0.05	0.1	0.001	5	99.6
Domingo	2	2	2	0.11	1.0	0.100	1.96	41.04

Utilizando as estruturas especificadas na Tabela 4.1, foram obtidos os erros absolutos percentuais médios de previsão para o período de 1º. de fevereiro de 1991 a 31 de dezembro de 1991, dado pela equação (3.13), e apresentados na Tabela 4.2. Nesta tabela, foi inserido também o erro obtido por um modelo ARX que utiliza as mesmas entradas dos modelos neurais, estimado através da aplicação do algoritmo de mínimos quadrados. Nesta tabela, pode ser verificado que a SVM apresentou o melhor desempenho, em termos do erro absoluto percentual médio de previsão, para cinco dos

sete modelos estimados, com o MLP estimado através da aplicação do algoritmo de treinamento *bayesiano* superando a SVM para a modelagem do comportamento da carga nas quartas e quintas-feiras. Na última coluna desta tabela, é apresentada a diferença percentual entre o menor e o maior erro obtido para cada um dos sete modelos, utilizando cada uma das metodologias descritas na tabela. Exemplificando, para a modelagem do comportamento da carga para os sábados, a SVM apresentou desempenho superior, em termos do erro absoluto percentual médio de previsão, de cerca de 45 % em relação ao MLP treinado sem a utilização de nenhuma técnica de regularização, ou seja, o erro médio percentual de previsão da SVM foi cerca de 45 [%] inferior ao obtido pelo MLP citado. Para melhor ilustração dos resultados obtidos, a Tabela 4.3 apresenta o erro absoluto percentual máximo obtido para o mesmo período, sendo apresentado de maneira análoga ao erro médio percentual apresentado na Tabela 4.2.

Tabela 4.2 – Erro absoluto percentual médio para o período de 01/02/1991 a 31/12/1991

	ARX	Sem Regularizador	*Bayesiano*	Escalonamento	SVM	Ganho de Desempenho
Segunda	8.23	8.76	6.43	7.00	5.23	40.3
Terça	8.16	7.04	7.47	6.52	4.97	39.1
Quarta	8.15	6.94	7.08	6.18	5.00	38.7
Quinta	8.15	10.21	6.93	8.41	7.83	32.1
Sexta	9.54	7.33	6.29	7.18	6.39	34.1
Sábado	7.38	9.57	7.38	8.17	5.24	45.3
Domingo	7.02	8.19	6.91	7.42	5.00	38.9
Média	8.09	8.29	6.93	7.27	5.66	31.7

4.2 Previsão do pico de carga

Para os modelos desenvolvidos para previsão do pico de carga, utilizando os procedimentos descritos no item 3.2 para seleção dos diversos parâmetros que definem os modelos neurais utilizados nesta tese, foram obtidas as estruturas descritas na Tabela 4.4. Esta tabela contêm basicamente as mesmas informações apresentadas na Tabela 4.1, porém para os modelos desenvolvidos para modelagem do pico de carga diário.

Tabela 4.3 – Erro absoluto percentual máximo para o período de 01/02/1991 a 31/12/1991

	ARX	Sem Regularizador	Bayesiano	Escalonamento	SVM	Ganho de Desempenho
Segunda	59.08	57.36	42.54	53.95	43.11	28.00
Terça	66.58	37.44	70.24	34.38	36.62	51.05
Quarta	87.13	164.04	171.62	53.74	30.97	81.95
Quinta	60.15	95.05	57.09	80.28	16.01	83.16
Sexta	131.26	89.71	101.08	64.13	15.31	88.34
Sábado	47.79	92.99	55.83	57.66	32.34	65.22
Domingo	68.65	287.78	101.94	94.44	19.39	93.26
Média	74.38	117.77	85.76	62.65	27.68	76.50

Utilizando as estruturas descritas na Tabela 4.4, foram obtidos, para cada um dos modelos, os erros médios percentuais para janeiro de 1999, dado pela equação (3.14) e apresentados na Tabela 4.5. Visto que a competição promovida pelo EUNITE em 2001 utilizou para avaliação dos modelos, além do erro absoluto percentual médio para janeiro de 1999, o erro máximo absoluto, em [MW], obtido para as previsões realizadas para este mesmo período, a segunda linha da Tabela 4.5 apresenta os erros máximos percentual e absoluto obtidos para cada um dos modelos. Para esta aplicação, dentre os modelos utilizados nesta tese, o MLP treinado através da aplicação do algoritmo de treinamento *bayesiano* apresentou os melhores resultados tanto em termos do erro médio percentual quanto em relação ao erro máximo percentual.

Tabela 4.4 – Estruturas obtidas a partir dos resultados para dezembro de 1998

Sem Regularizador	*Bayesiano*	Escalonamento		SVM		
Neurônios	Neurônios	Neurônios	σ^2_{ruido}	C	ε	σ_{kernel}
2	2	2	0.148	1.0	0.100	4.80

Tabela 4.5 – Erro absoluto percentual médio e erro máximo, percentual e absoluto, em [MW], obtido para cada um dos modelos para as previsões realizadas para janeiro de 1999

	Sem Regularizador	*Bayesiano*	Escalonamento	SVM	Ganho de Desempenho
Erro [%]	2.47	2.02	7.53	2.36	73.2
Erro Máximo [%]	7.64	6.41	13.64	7.79	53.0
Erro Máximo [MW]	61.20	51.34	109.26	62.40	53.0

Os resultados obtidos pelos dez primeiros colocados na competição promovida em 2001 pelo EUNITE, disponibilizados no site da competição (http://neuron.tuke.sk/competition), estão apresentados na Tabela 4.1. Observando esta tabela, o MLP que utiliza treinamento *bayesiano* desenvolvido nesta tese obteria o segundo melhor desempenho na competição, tanto em termos do erro absoluto percentual médio quanto em termos do erro absoluto máximo. A discussão dos resultados apresentados neste capítulo constitui a essência do capítulo apresentado a seguir.

Tabela 4.6 – Erro absoluto percentual médio e erro máximo absoluto, em [MW], obtidos pelos dez primeiros colocados da competição de 2001

	Erro Médio [%]	Erro Máximo [MW]
1	1.9824	51.4
2	2.1489	40.0
3	2.4979	60.5
4	2.8733	65.8
5	2.9846	68.6
6	3.2234	55.2
7	3.2644	77.0
8	3.3797	74.0
9	3.388	109.0
10	3.3887	74.5

5

Conclusões

Esta tese teve por objetivo a aplicação de algumas técnicas de controle de complexidade, ou regularização, no desenvolvimento de modelos neurais *feedforward* para previsão de carga elétrica para o horizonte de curto prazo, mais especificamente os MLP's e as SVM's. Conforme evidenciado pelos resultados apresentados no capítulo 4, as técnicas utilizadas neste trabalho apresentam potencial significativo, visto que as mesmas deram origem a modelos com capacidade de generalização superior aos modelos obtidos através da aplicação do algoritmo clássico de retropropagação do erro, que utiliza na sua formulação o princípio da minimização do risco empírico. Diante do exposto no capítulo 2, este resultado já era esperado, visto que, na presença de dados ruidosos, a aplicação do princípio da minimização do risco empírico, que busca única e exclusivamente o ajuste dos dados disponíveis, conduz à estimativa de modelos com reduzida capacidade de generalização.

Diante dos resultados apresentados no capítulo 4, deve ser salientado o desempenho destacado da máquina de vetor suporte, que apresentou os melhores resultados para cinco dos sete modelos desenvolvidos para previsão da carga horária. Apesar deste desempenho preliminar superior, é esperado que os resultados obtidos possam ser incrementados através da busca, no espaço tridimensional, de valores ótimos para os parâmetros que definem a SVM, ou seja, a largura σ_{kernel} utilizada pelo *kernel gaussiano* $K\left(\underline{x},\underline{x}_k\right)$, e as constantes C e ε, que possuem influência decisiva no processo de otimização utilizado para treinamento das SVM's. Uma das principais dificuldades encontradas na estimativa destes parâmetros reside na ausência de valores de referência para estas constantes para o problema aqui abordado, visto que esta tese representa uma das primeiras experiências na utilização de SVM para previsão de carga elétrica. Para corroborar o significativo potencial das SVM's como ferramenta para

previsão de carga a curto prazo, vale a pena mencionar que o modelo vencedor da competição promovida pelo EUNITE em 2001, cujos principais resultados estão apresentados na Tabela 4.6, utiliza esta metodologia, conforme exposto em [184].

Outra questão interessante que pode ser levantada em função dos resultados obtidos diz respeito à comparação entre o desempenho do modelo ARX e da SVM para previsão da carga horária. Conforme exposto no item 2.1, a SVM pode ser vista como um modelo linear aplicado em um espaço de representação expandido, chamado de espaço de características. Sob este prisma, o desempenho razoavelmente superior da SVM quando comparada ao modelo ARX dá margem à interpretação intuitiva de que a série de carga em estudo, não-estacionária no espaço de representação original mesmo com a aplicação das diferenças sazonais, pode ser feita estacionária no espaço de características, visto que, neste novo espaço de representação, a aplicação de um modelo linear foi, de uma certa maneira, bem sucedida. A análise do comportamento da SVM para problemas de aproximação funcional, ou de regressão, ainda é um campo de estudo amplo, visto que, além de ser uma tecnologia recente, este tipo de modelo neural *feedforward* foi desenvolvido originalmente para abordagem de problemas de classificação, com a grande maioria das aplicações desta metodologia encontradas na literatura abordando este tipo de problema.

Além do desempenho satisfatório das SVM's para previsão da curva de carga horária, também merece destaque o desempenho do algoritmo de treinamento *bayesiano* de MLP's como técnica de regularização. Além de apresentar os melhores resultados, em termos do erro médio percentual de previsão, para dois dos sete modelos desenvolvidos para previsão da curva de carga diária, esta técnica deu origem ao modelo com melhor desempenho para previsão do pico de carga diário, tanto em termos do erro absoluto percentual médio de previsão quanto em relação ao erro máximo de

previsão. Da mesma forma que para as SVM's, o desempenho dos MLP's estimados através da aplicação desta técnica, que apresenta como principal vantagem o procedimento automático e iterativo para o cálculo do parâmetro de regularização λ, pode ser melhorado através da utilização de outras distribuições de probabilidades *a priori* $p(\underline{w})$, dando origem a outros funcionais regularizadores $E_c(\underline{w})$. Uma forma de implementar esta melhoria reside na utilização de distribuições de probabilidade *a priori* diferentes para diversos conjuntos de pesos, procurando agrupar em cada conjunto os pesos relacionados com uma mesma grandeza. Para o caso do modelo de previsão do pico de carga diário, por exemplo, poderiam ser identificados quatro conjuntos de pesos. O primeiro agrupando os pesos que ligam as entradas de carga aos neurônios da camada oculta. O segundo relacionado os pesos que ligam as entradas binárias aos neurônios da camada oculta. O terceiro abrangendo os *bias* dos neurônios da camada intermediária. O quarto e último incorporando os pesos que ligam os neurônios da camada oculta ao neurônio de saída, incluindo neste grupo o *bias* do neurônio de saída. Este procedimento de agrupamento dos pesos em diferentes conjuntos, tomando por base as possíveis similaridades existentes entre cada um, é recomendada em [158] e [164] para melhoria do desempenho do algoritmo de treinamento *bayesiano*, em termos do aumento da capacidade de generalização do modelo estimado.

Apesar do escalonamento do ganho da função de ativação apresentar ganho de desempenho, em relação ao MLP treinado através do algoritmo de retropropagação de erro tradicional, para todos os modelos desenvolvidos para previsão da curva de carga diária, esta heurística de regularização de MLP's mostrou severas limitações quando aplicada ao modelo de previsão do pico de carga diário, visto que esta técnica degradou a capacidade de generalização do MLP treinado sem regularizador, o que não era

esperado. Das técnicas utilizadas, esta era, sem dúvida, a mais interessante sob o ponto de vista de esforço computacional, visto que utiliza um procedimento extremamente simples de ajuste dos ganhos das funções de ativação dos neurônios da camada oculta do MLP. Apesar do desenvolvimento apresentado por *Reed* et. al. em [181] não abordar esta situação, o uso de valores diferentes para a variância σ^2_{ruido} para diferentes grupos de entradas, fazendo analogia com o treinamento *bayesiano*, pode resultar em melhorias nesta heurística de regularização de MLP's.

Em virtude das considerações feitas acima, a SVM e o treinamento *bayesiano* de MLP's podem ser qualificados como métodos eficientes e robustos para desenvolvimento de modelos neurais *feedforward* para previsão de carga a curto prazo, restando para confirmação do potencial destes dois procedimentos a aplicação destes a problemas reais, ou seja, a aplicação destas técnicas no desenvolvimento de um sistema completo de previsão de carga para o horizonte de curto prazo.

6

Referências Bibliográficas

[1] PARK, J.H.; PARK, Y.M.; LEE, K.Y.; "Composite Modeling for Adaptive Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.6, n.2, pp. 450-457, May 1991.

[2] MBAMALU, G.A.N.; EL-HAWARY, M.E.; "Load Forecasting via Suboptimal Seasonal Autoregressive Models and Iteratively Reweighted Least Squares Estimation", *IEEE Transactions on Power Systems*, v.8, n.1, pp. 343-348, Feb. 1993.

[3] DJUKANOVIC, M.; BABIC, B.; SOBAJIC, D.J.; PAO, Y,H.; "Unsupervised/Supervised Learning Concept for 24-hour Load Forecasting", *IEE Proceedings C*, v.140, n.4, July 1993.

[4] KIM, K.H.; PARK, J.K.; HWANG, K.J.; KIM, S.H.; "Implementation of Hybrid Short-term Load Forecasting System Using Artificial Neural Networks and Fuzzy Expert Systems", *IEEE Transactions on Power Systems*, v.10, n.3, pp. 1534-1539, Aug. 1995.

[5] KHOTANZAD, A.; AFKHAMI-ROHANI, R.; LU, T.L.; ABAYE, A.; DAVIS, M.; MARATUKULAM, D.J.; "ANNSTLF – A Neural-Network-Based Electric Load Forecasting System", *IEEE Transactions on Neural Networks*, v.8, n.4, pp. 835-846, July 1997.

[6] HUANG, S.R.; "Short-term Load Forecasting Using Threshold Autoregressive Models", *IEE Proceedings on Generation, Transmission and Distribution*, v.144, n.5, pp. 477-481, Sept. 1997.

[7] CHOUEIKI, M.H.; MOUNT-CAMPBELL, C.A.; AHALT, S.C.; "Building a "Quasi Optimal" Neural Network to Solve the Short-term Load Forecasting

Problem", *IEEE Transactions on Power Systems*, v.12, n.4, pp. 1432-1439, Nov. 1997.

[8] MOHAMED, E.A.; MANSOUR, M.M.; EL-DEBEIKY, S.E.; MOHAMED, K.G.; "Egyptian Unified Grid Hourly Load Forecasting Using Artificial Neural Networks", *International Journal of Electrical Power & Energy Systems*, v.20, n.7, pp. 495-500, Oct. 1998.

[9] KODOGIANNIS, V.S.; ANAGNOSTAKIS, E.M.; "A Study of Advanced Learning Algorithms for Short-term Load Forecasting", *Engineering Applications of Artificial Intelligence*, v.12, n.2, pp. 159-173, April 1999.

[10] AL-SABA, T.; EL-AMIN, I.; "Artificial Neural Networks as Applied to Long-term Demand Forecasting", *Artificial Intelligence in Engineering*, v.13, n.2, pp. 189-197, April 1999.

[11] EL DESOUKY, A.A.; ELKATEB, M.M.; "Hybrid Adaptive Techniques for Electric-Load Forecast Using ANN and ARIMA", *IEE Proceedings on Generation, Transmission and Distribution*, v.147, n.4, pp. 213-217, July 2000.

[12] TAMIMI, M.; EGBERT, R.; "Short Term Electric Load Forecasting via Fuzzy Neural Collaboration", *Electric Power Systems Research*, v.56, n.3, pp. 243-248, Dec. 2000.

[13] ZHENG, T.; GIRGIS, A.A.; MAKRAM, E.B.; "A Hybrid Wavelet-Kalman Filter Method for Load Forecasting", *Electric Power Systems Research*, v.54, n.1, pp. 11-17, April 2000.

[14] HIPPERT, H.S; PEDREIRA, C.E.; SOUZA, R.C.; "Neural Networks for Short-Term Load Forecasting: A Review and Evaluation", *IEEE Transactions on Power Systems*, v.16, n.1, pp. 44-55, Feb. 2001.

[15] JIA, N.X.; YOKOYAMA, R.; ZHOU, Y.C.; GAO, Z.Y.; “A Flexible Long-Term Load Forecasting Approach Based on New Dynamic Simulation Theory – GSIM”, *International Journal of Electrical Power & Energy Systems*, v.23, n.7, pp. 549-556, Oct. 2001.

[16] KANDIL, M.S.; EL-DEBEIKY, S.M.; HASANIEN, N.E.; “Overview and Comparison of Long-Term Forecasting Techniques for a Fast Developing Utility: Part I”, *Electric Power Systems Research*, v.58, n.1, pp. 11-17, May 2001.

[17] KANDIL, M.S.; EL-DEBEIKY, S.M.; HASANIEN, N.E.; “The Implementation of Long-Term Forecasting Strategies Using a Knowledge-Based Expert System: Part-II”, *Electric Power Systems Research*, v.58, n.1, pp. 19-25, May 2001.

[18] KANDIL, M.S.; EL-DEBEIKY, S.M.; HASANIEN, N.E.; “Long Term Load Forecasting for Fast Developing Utility Using a Knowledge-Based Expert System”, *IEEE Transactions on Power Systems*, v.17, n.2, pp. 491-496, May 2002.

[19] SATISH, B.; SWARUP, K.S.; SRINIVAS, S.; RAO, A.H.; “Effect of Temperature on Short Term Load Forecasting Using a Integrated ANN”, *Electric Power Systems Research*, v.72, n.1, pp. 95-101, Nov. 2004.

[20] RAHMAN, S.; “Formulation and Analysis of a Rule-Based Short-term Load Forecasting”, *Proceedings of IEEE*, v.78, n.5, pp. 805-815, May 1990.

[21] SRINIVASAN, D.; TAN, S.S.; CHANG, C.S.; CHAN, E.K.; “Parallel Neural Network-Fuzzy Expert System Strategy for Short-Term Load Forecasting: System Implementation and Performance Evaluation”, *IEEE Transactions on Power Systems*, v.14, n.3, pp. 1100-1106, Aug. 1999.

[22] KALAITZAKIS, K.; STAVRAKAKIS, G.S.; ANAGNOSTAKIS, E.M.; "Short-Term Load Forecasting Based on Artificial Neural Networks Parallel Implementation", *Electric Power Systems Research*, v.63, n.3, pp. 185-196, Oct. 2002.

[23] HSU, C.C.; CHEN, C.Y.; "Regional Load Forecasting in Taiwan – Applications of Artificial Neural Networks", *Energy Conversion and Management*, v.44, n.12, pp. 1941-1949, July 2003.

[24] KERMANSHAHI, B.; IWAMIYA, H.; "Up to Year 2020 Load Forecasting Using Neural Nets", *International Journal of Electrical Power & Energy Systems*, v.24, n.9, pp. 789-797, Nov. 2002.

[25] KERMANSHAHI, B.; "Recurrent Neural Network for Forecasting Next 10 Years Loads of Nine Japanese Utilities" *Neurocomputing*, v.23, n.1-3, pp. 125-133, Dec. 1998

[26] CHEN, G.J.; LI, K.K; CHUNG, T.S.; SUN, H.B; TANG, G.Q.; "Application of an Innovative Combined Forecasting Method in Power System Load Forecasting", *Electric Power Systems Research*, v.59, n.2, pp. 131-137, Sept. 2001.

[27] ELRAZAZ, Z.S.; MAZI, A.A.; "Unified Weekly Peak Load Forecasting for Fast Growing Power System", *IEE Proceedings C*, v.136, n.1, pp. 29-34, Jan. 1989.

[28] DOVEH, E.; FEIGIN, P.; GREIG, D.; HYAMS, L.; "Experience with FNN Models for Medium Term Power Demand Predictions", *IEEE Transactions on Power Systems*, v.14, n.2, pp. 538-546, May 1999.

[29] RAHMAN, S.; SHRESTHA, G.; “A Priority Vector Based Technique for Load Forecasting”, *IEEE Transactions on Power Systems*, v.6, n.4, pp. 1459-1465, Nov. 1991.

[30] BARAKAT, E.H.; AL-QASSIM, J.M.; AL RASHED, S.A.; “New Model for Peak Demand Forecasting Applied to Highly Complex Load Characteristics of a Fast Developing Area”, *IEE Proceedings C*, v.139, n.2, pp. 136-140, Mar. 1992.

[31] BARAKAT, E.H.; EISSA, M.A.M.; “Forecasting Monthly Peak Demand in Fast Growing Electric Utility Using a Composite Multiregression-Decomposition Model”, *IEE Proceedings C*, v.136, n.1, pp. 35-41, Jan. 1989.

[32] BARAKAT, E.H.; QAYYUM, M.A.; HAMED, M.N.; AL RASHED, S.A.; “Short-term Peak Demand Forecasting in Fast Developing Utility with Inherit Dynamic Load Characteristics: Part-I – Application of Classical Time-Series Methods”, *IEEE Transactions on Power Systems*, v.5, n.3, pp. 813-824, Aug. 1990.

[33] TEMRAZ, H.K.; SALAMA, M.M.A.; QUINTANA, V.H.; “Application of the Decomposition Technique for Forecasting the Load of a Large Electric Power Network”, *IEE Proceedings on Generation, Transmission and Distribution*, v.143, n.1, pp. 1318, Jan. 1996.

[34] ELKATEB, M.M.; SOLAIMAN, K.; AL-TURKI, Y.; “A Comparative Study of Medium-weather-dependent Load Forecasting Using Enhanced Artificial/Fuzzy Neural Network and Statistical Techniques”; *Neurocomputing*, v.23, n.1, pp. 3-13, Dec.1998.

[35] BARAKAT, E.H.; AL-QASEM, J.M.; “Methodology for Weekly Load Forecasting”, *IEEE Transactions on Power Systems*, v.13, n.4, pp. 1548-1555, Nov. 1998.

[36] RAHMAN, S.; BHATNAGAR, R.; “An Expert System Based Algorithm for Short-Term Load Forecasting”, *IEEE Transactions on Power Systems*, v.3, n.2, pp. 392-399, May 1988.

[37] HO, K.L.; HSU, Y.Y.; CHEN, C.F.; LEE, T.E.; “Short-term Load Forecasting of Taiwan Power System Using a Knowledge-based Expert System”, *IEEE Transactions on Power Systems*, v. 5, n.4, pp. 1214-1221, Nov. 1990.

[38] PAPALEXOPOULOS, A.D.; HESTERBERG, T.C.; “A Regression-based Approach to Short-term Load Forecasting”, *IEEE Transactions on Power Systems*, v.5, n.4, pp. 1535-1550, Nov. 1990.

[39] PARK, D.C.; EL-SHARKAWI, M.A.; MARKS II, R.J.; ATLAS, L.E.; DAMBORG, M.J.; “Electric Load Forecasting Using An Artificial Neural Network”, *IEEE Transactions on Power Systems*, v.6, n.2, pp. 442-449, May 1991.

[40] PENG, T.M.; HUBELE, N.F.; KARADY, G.G.; “Advancement in the Application of Neural Networks for Short-term Load Forecasting”, *IEEE Transactions on Power Systems*, v.7, n.1, pp. 250-257, Feb. 1992.

[41] CHEN, S.T; YU, D.C.; MOGHADDAMJO, A.R.; “Weather Sensitive Short-term Load Forecasting Using Nonfully Connected Artificial Neural Network”, *IEEE Transactions on Power Systems*, v.7, n.3, pp. 1098-1105, Aug. 1992.

[42] ASAR, A.U.; MCDONALD, J.R.; “A Specification of Neural Network Applications in the Load Forecasting Problem”, *IEEE Transactions on Control Systems Technology*, v.2, n.2, pp. 135-141, June 1994.

[43] PAPALEXOPOULOS, A.D.; HAO, S.; PENG, T.M.; “An Implementation of a Neural Network Based Load Forecasting Model for the EMS”, *IEEE Transactions on Power Systems*, v.9, n.4, pp. 1956-1962, Nov. 1994.

[44] RANAWEERA, D.K.; HUBELE, N.F.; PAPALEXOPOULOS, A.D.; “Application of Radial Basis Function Neural Network Model for Short-Term Load Forecasting”, *IEE Proceedings on Generation, Transmission and Distribution*, v.142, n.1, pp. 45-50, Jan. 1995.

[45] KHOTANZAD, A.; HWANG, R.C.; ABAYE, A.; MARATUKULAM, D.; “An Adaptive Modular Artificial Neural Network Hourly Load Forecaster and Its Implementation at Electric Utilities”, *IEEE Transactions on Power Systems*, v.10, n.3, pp. 1716-1722, Aug. 1995.

[46] DASH, P.K.; SATPATHY, H.P.; LIEW, A.C.; RAHMAN, S.; “A Real-Time Short-term Load Forecasting System Using Functional Link Network”, *IEEE Transactions on Power Systems*, v.12, n.2, pp. 675-680, May 1997.

[47] CHARYTONIUK, W.; CHEN, M.S.; VAN OLINDA, P.; “Non Parametric Regression Based Short-term Load Forecasting”, *IEEE Transactions on Power Systems*, v.13, n.3, pp. 725-730, Aug. 1998.

[48] KHOTANZAD, A.; AFKHAMI-ROHANI, R.; MARATUKULAM, D.; “ANNSTLF – Artificial Neural Network Short-Term Load Forecaster – Generation Three”, *IEEE Transactions on Power Systems*, v.13, n.4, pp. 1413-1422, Nov. 1998.

[49] SRINIVASAN, D.; “Evolving Artificial Neural Networks for Short Term Load Forecasting”, *Neurocomputing*, v.23, n.1, pp. 265-276, Dec. 1998.

[50] DASH, P.K.; SATPATHY, H.P.; LIEW, A.C.; “A Real-time Short-term Peak and Average Load Forecasting System Using a Self-organizing Fuzzy Neural

Network", *Engineering Applications of Artificial Intelligence*, v.11, n.2, pp. 307-316, April 1998.

[51] MASTOROCOSTAS, P.A.; THEOCHARIS, J.B.; BAKIRTZIS, A.G.; "Fuzzy Modeling for Short Term Load Forecasting Using the Orthogonal Least Squares Method", *IEEE Transactions on Power Systems*, v.14, n.1, pp. 29-36, Feb. 1999.

[52] WU, H.C.; LU, C.N.; "Automatic Fuzzy Model Identification for Short-term Load Forecast", *IEE Proceedings on Generation, Transmission and Distribution*, v.146, n.5, pp. 477-482, Sept. 1999.

[53] HOBBS, B.F.; JITPRAPAIKULSARN, S.; MARATUKULAM, D.J.; KONDA, S.; CHANKONG, V.; LOPARO, K.A.; "Analysis of the Value for Unit Commitment of Improved Load Forecasts", *IEEE Transactions on Power Systems*, v.14, n.4, pp. 1342-1348, Nov. 1999.

[54] LIANG, R.H.; CHENG, C.C.; "Combined Regression-fuzzy Approach for Short-term Load Forecasting", *IEE Proceedings on Generation, Transmission and Distribution*, v.147, n.4, pp. 261-266, July 2000.

[55] MASTOROCOSTAS, P.A.; THEOCHARIS, J.B.; KIARTZIS, S.J.; BAKIRTZIS, A.G.; "A Hybrid Fuzzy Modeling Method for Short-Term Load Forecasting", *Mathematics and Computers in Simulation*, v.51, n.3, pp. 221-232, Jan. 2000.

[56] AMJADY, N.; "Short-Term Hourly Load Forecasting Using Time-Series Modeling With Peak Load Estimation Capability", *IEEE Transactions on Power Systems*, v. 16, n.4, pp. 798-805, Nov. 2001.

[57] MARIN, F.J.; GARCIA-LAGOS, F.; JOYA, G.; SANDOVAL, F.; "Global Model for Short-Term Load Forecasting Using Artificial Neural Networks",

IEE Procedings on Generation, Transmission and Distribution, v.149, n.2, pp. 121-125, Mar. 2002.

[58] KIM, C.I.; YU, I.K.; SONG, Y.H.; "Kohonen Neural Network and Wavelet Transform Based Approach to Short-Term Load Forecasting", *Electric Power Systems Research*, v.63, n.3, pp. 169-176, Oct. 2002.

[59] KALAITZAKIS, K.; STAVRAKAKIS, G.S.; ANAGNOSTAKIS, E.M.; "Short-Term Load Forecasting Based on Artificial Neural Networks Parallel Implementation", *Electric Power Systems Research*, v.63, n.3, pp. 185-196, Oct. 2002.

[60] LIANG, R.H.; CHENG, C.C.; "Short-Term Load Forecasting by a Neuro-Fuzzy Based Approach", *International Journal of Electrical Power & Energy Systems*, v.24, n.2, pp. 103-111, Feb. 2002.

[61] METAXIOTIS, K.; KAGIANNAS, A.; ASKOUNIS, D.; PSARRAS, J.; "Artificial Intelligence in Short Term Electric Load Forecasting: A State-of-the-art Survey for the Researcher", *Energy Conversion and Management*, v.44, n.9, pp. 1525-1534, June 2003.

[62] ABDEL-AAL, R.E.; "Short-Term Hourly Load Forecasting Using Abductive Networks", *IEEE Transactions on Power Systems*, v.19, n.1, pp. 164-173, Feb. 2004.

[63] TRUDNOWSKI, D.J.; MCREYNOLDS, W.L.; JOHNSON, J.M.; "Real-time Very Short-Term Load Prediction for Power System Automatic Generation Control", *IEEE Transactions on Control Systems Technology*, v.9, n.2, pp. 254-260, Mar. 2001.

[64] JABBOUR, K.; RIVEROS, J.F.V.; LANDSBERGEN, D.; MEYER, W.; "ALFA: Automated Load Forecasting Assistant", *IEEE Transactions on Power Systems*, v.3, n.3, pp. 908-914, Aug. 1988.

[65] RAHMAN, S.; BABA, M.; "Software Design and Evaluation of a Microcomputer-Based Automated Load Forecasting System", *IEEE Transactions on Power Systems*, v.4, n.2, pp. 782-788, May 1989.

[66] LU, C.N.; WU, H.T.; VEMURI, S.; "Neural Network Based Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.8, n.1, pp.336-342, Feb. 1993.

[67] PENG, T.M.; HUBELE, N.F.; KARADY, G.G.; "An Adaptive Neural Network Approach to One-week Ahead Load Forecasting", *IEEE Transactions on Power Systems*, v.8, n.3, pp. 1195-1203, Aug. 1993.

[68] FAN, J.Y.; MCDONALD, J.D.; "A Real Time Implementation of Short-term Load Forecasting for Distribution Power Systems", *IEEE Transactions on Power Systems*, v.9, n.2, pp. 988-994, May 1994.

[69] MOHAMMED, O. PARK, D.; MERCHANT, R.; DINH, T.; TONG, C.; AZEEM, A.; FARAH, J.; DRAKE, C.; "Practical Experiences with an Adaptive Neural Network Short-term Load Forecasting System", *IEEE Transactions on Power Systems*, v.10, n.1, pp. 254-265, Feb. 1995.

[70] BAKIRTZIS, A.G.; THEOCHARIS, J.B.; KIARTZIS, S.J.; SATSIOS, K.J.; "Short-term Load Forecasting Using Fuzzy Neural Networks", *IEEE Transactions on Power Systems*, v.10, n.3, pp. 1518-1524, Aug. 1995.

[71] PIRAS, A.; BUCHENEL, B.; JACCARD, Y.; GERMOND, A.; IMHOF, K.; "Heterogeneous Artificial Neural Network for Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.11, n.1, pp. 397-402, Feb. 1996.

[72] RANAWEERA, D.K.; KARADY, G.G.; FARMER, R.G.; "Effect of Probabilistic Inputs on Neural Network-Based Electric Load Forecasting", *IEEE Transactions on Neural Networks*, v.7, n.6, pp. 1528-1532, Nov. 1996.

[73] CHOW, T.W.S.; LEUNG, C.T.; "Neural Network Based Short-Term Load Forecasting Using Weather Compensation", *IEEE Transactions on Power Systems*, v.11, n.4, pp. 1736-1742, Nov. 1996.

[74] HUANG, S.J.; HUANG, C.L.; "Genetic-Based Multilayered Perceptron for Taiwan Power System Short-Term Load Forecasting", *Electric Power Systems Research*, v.38, n.1, pp. 69-74, July 1996.

[75] RANAWEERA, D.K.; KARADY, G.G.; FARMER, R.G.; "Economic Impact Analysis of Load Forecasting", *IEEE Transactions on Power Systems*, v.13, n.3, pp. 1388-1392, Aug. 1997.

[76] ALFUHAID, A.S.; EL-SAYED, M.A.; MAHMOUD, M.S.; "Cascaded Artificial Neural Networks for Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.12, n.4, pp. 1524-1529, Nov. 1997.

[77] CHIU, C.C.; KAO, L.J.; COOK, D.F.; "Combining a Neural Network with a Rule-Based Expert System Approach for Short-term Power Load Forecasting in Taiwan", *Expert Systems with Applications*, v.13, n.4, pp. 299-305, Nov. 1997.

[78] KIARTZIS, S.; KEHAGIAS, A.; BAKIRTZIS, A.; PETRIDIS, V.; "Short Term Load Forecasting Using a Bayesian Combination Method", *International Journal of Electrical Power & Energy Systems*, v.19, n.3, pp. 171-177, Mar. 1997.

[79] DOUGLAS, A.P.; BREIPHOL, A.M.; LEE, F.N., ADAPA, R.; "The Impacts of Temperature Forecast Uncertainty on Bayesian Load Forecasting", *IEEE Transactions on Power Systems*, v.13, n.4, pp. 1507-1513, Nov. 1998.

[80] VERONA, F.B.; CERAOLO, M.; "Use of Neural Networks for Customer Tariff Exploitation by Means of Short-term Load Forecasting", *Neurocomputing*, v.23, n.1, pp. 135-149, Dec. 1998.

[81] CHOW, T.W.S.; LEUNG, C.T.; "Nonlinear Autoregressive Integrated Neural Network Model for Short-Term Load Forecasting", *IEE Proceedings on Generation, Transmission and Distribution*, v.143, n.5, pp. 500-506, Sept. 1996.

[82] NAZARKO, J.; ZALEWSKI, W.; "The Fuzzy Regression Approach to Peak Load Estimation in Power Distribution Systems", *IEEE Transactions on Power Systems*, v.14, n.3, pp. 809-814, Aug. 1999.

[83] BASU, S.N.; "Short Term Localized Load Prediction", *IEEE Transactions on Power Systems*, v.7, n.1, pp. 389-397, Feb. 1992.

[84] CHARYTONIUK, W.; CHEN, M.S.; KOTAS, P.; VAN OLINDA, P.; "Demand Forecasting in Power Distribution Systems Using Nonparametric Probability Density Estimation", *IEEE Transactions on Power Systems*, v.14, n.4, pp. 1200-1206, Nov. 1999.

[85] YAO, S.J.; SONG, Y.H.; ZHANG, L.Z.; CHENG, X.Y.; "Wavelet Transform and Neural Networks for Short-Term Electrical Load Forecasting", *Energy Conversion and Management*, v.41, n.18, pp. 1975-1988, Dec. 2000.

[86] HUANG, C.M.; YANG, H.T.; "Evolving Wavelet-Based Networks for Short-Term Load Forecasting", *IEE Proceedings on Generation, Transmission and Distribution*, v.148, n.3, pp. 222-228, May 2001.

[87] ALVES DA SILVA, A.P.; RODRIGUES, U.P.; REIS, A.J.R.; MOULIN, L.S.; "Oráculo – Uma ferramenta para Previsão de Carga", *XVI SNPTEE – Seminário*

Nacional de Produção e Transmissão de Energia Elétrica, GOP/012, Campinas, São Paulo, Brasil, 21-26 de Outubro de 2001.

[88] TAYLOR, J.W.; BUIZZA, R.; "Neural Network Load Forecasting with Weather Ensemble Predictions", *IEEE Transactions on Power Systems*, v.17, n.3, pp. 626-632, Aug. 2002.

[89] SAINI, L.M.; SONI, M.K.; "Artificial Neural Network-Based Peak Load Forecasting Using Conjugate Gradient Methods", *IEEE Transactions on Power Systems*, v.17, n.3, pp. 907-912, Aug. 2002.

[90] KHOTANZAD, A.; ZHOU, E.; ELRAGAL, H.; "A Neuro-Fuzzy Approach to Short-Term Load Forecasting in a Price-Sensitive Environment", *IEEE Transactions on Power Systems*, v. 17, n.4, pp. 1273-1282, Nov. 2002.

[91] ZAGRAJEK, J.N.; WERON, R.; "Modeling Electricity Loads in California: ARMA Models with Hyperbolic Noise", *Signal Processing*, v.82, n.12, pp. 1903-1915, Dec. 2002.

[92] HUANG, S.J.; SHIH, K.R.; "Short-Term Load Forecasting Via ARMA Model Identification Including Non-Gaussian Process Considerations", *IEEE Transactions on Power Systems*, v.18, n.2, pp. 673-679, May 2003.

[93] LIAO, G.C.; TSAO, T.P.; "Application of Fuzzy Neural Networks and Artificial Intelligence for Load Forecasting", *Electric Power Systems Research*, v.70, n.3, pp. 237-244, Aug. 2004.

[94] AL-HAMADI, H.M.; SOLIMAN, S.A.; "Short-Term Electric Load Forecasting Based on Kalman Filtering Algorithm with Moving Window Weather and Load Model", *Electric Power Systems Research*, v.68, n.1, pp. 47-59, Jan. 2004.

[95] KIARTZIS, S.J.; ZOUMAS, C.E.; THEOCHARIS, J.B.; BAKIRTZIS, A.G.; PETRIDIS, V.; "Short-term Load Forecasting in an Autonomous Power System

Using Artificial Neural Networks", *IEEE Transactions on Power Systems*, v.12, n.4, pp. 1591-1596, Nov. 1997.

[96] DOUGLAS, A.P.; BREIPHOL, A.M.; LEE, F.N.; ADAPA, R.; "Risk Due to Load Forecast Uncertainty in Short Term Power Systems Planning", *IEEE Transactions on Power Systems*, v.13, n.4, pp. 1493-1499, Nov. 1998.

[97] VALENZUELA, J.; MAZUMDAR, M.; KAPOOR, A.; "Influence of Temperature and Load Forecast Uncertainty on Estimates of Power Generation Costs", *IEEE Transactions on Power Systems*, v.15, n.2, pp. 668-674, May 2000.

[98] LING, S.H.; LEUNG, F.H.F.; LAM, H.K.; TAM, P.K.S.; "Short-Term Electric Load Forecasting Based on a Neural Fuzzy Network", *IEEE Transactions on Industrial Electronics*, v.50, n.6, pp. 1305-1316, Dec. 2003.

[99] MOGHRAM, I.; RAHMAN, S.; "Analysis and Evaluation of Five Short-term Load Forecasting Techniques", *IEEE Transactions on Power Systems*, v.4, n.4, pp. 1484-1491, Oct. 1989.

[100] ATLAS, L.; COLE, R.; MUTHUSAMY, Y.; LIPPMAN, A.; CONNOR, J.; PARK, D.; EL-SHARKAWI, M.; MARKS, R.J.; "A Performance Comparison of Trained Multilayer Perceptrons and Trained Classification Trees", *Proceedings of IEEE*, v.78, n.10, pp. 1614-1619, Oct. 1990.

[101] PARK, D.C.; EL-SHARKAWI, M.A.; MARKS, R.J.; "An Adaptively Trained Neural Network", *IEEE Transactions on Neural Networks*, v.2, n.3, pp. 334-345, May 1991.

[102] HSU, Y.Y.; YANG, C.C.; "Design of Artificial Neural Networks for Short-Term Load Forecasting. Part II: Multilayer Feedforward Networks for Peak

Load and Valley Load Forecasting", *IEE Proceedings C*, v.138, n.5, pp. 414-418, Sept. 1991.

[103] HO, K.L.; HSU, Y.Y.; YANG, C.C.; "Short-term Load Forecasting Using a Multilayer Neural Network with an Adaptive Learning Algorithm", *IEEE Transactions on Power Systems*, v.7, n.1, pp. 141-149, Feb. 1992.

[104] HAIDA, T.; MUTO, S.; "Regression Based Peak Load Forecasting Using a Transformation Technique", *IEEE Transactions on Power Systems*, v.9, n.4, pp. 1788-1794, Nov. 1994.

[105] SRINIVASAN, D.; LIEW, A.C.; CHANG, C.S.; "Forecasting Daily Load Curves Using a Hybrid Fuzzy-Neural Approach", *IEE Proceedings on Generation, Transmission and Distribution*, v.141, n.6, pp. 561-567, Nov. 1994.

[106] GIRGIS, A.A.; VARADAN, S.; "Unit Commitment Using Load Forecasting Based on Artificial Neural Networks", *Electric Power Systems Research*, v.32, n.3, pp. 213-217, Mar. 1995.

[107] SRINIVASAN, D.; CHANG, C.S.; LIEW, A.C.; "Demand Forecasting Using Fuzzy Neural Computation, with Special Emphasis on Weekend And Public Holiday Forecasting", *IEEE Transactions on Power Systems*, v.10, n.4, pp. 1897-1903, Nov. 1995.

[108] YU, Z.; "A Temperature Match Based Optimization Method for Daily Load Prediction Considering DLC Effect", *IEEE Transactions on Power Systems*, v.11, n.2, pp. 728-733, May 1996.

[109] KHOTANZAD, A.; DAVIS, M.H.; ABAYE, A.; MARATUKULAM, D.J.; "An Artificial Neural Network Hourly Temperature Forecaster with

Applications in Load Forecasting", *IEEE Transactions on Power Systems*, v.11, n.2, pp. 870-876, May 1996.

[110] HYDE, O.; HODNETT, P.F.; "An Adaptable Automated Procedure for Short-Term Electricity Load Forecasting", *IEEE Transactions on Power Systems*, v.12, n.1, pp. 84-94, Feb. 1997.

[111] RAMANATHAN, R.; ENGLE, R.; GRANGER, C.W.J.; ARAGHI, F.V., BRACE, C.; "Short-Run Forecasts of Electricity Loads and Peaks", *International Journal of Forecasting*, v.13, n.2, pp. 161-174, June 1997.

[112] VERMAAK, J.; BOTHA, E.C.; "Recurrent Neural Networks for Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.13, n.1, pp. 126-132, Feb. 1998.

[113] YANG, H.T.; HUANG, C.M.; "A New Short-term Load Forecasting Approach Using Self-Organizing Fuzzy ARMAX Models", *IEEE Transactions on Power Systems*, v.13, n.1, pp. 217-225, Feb. 1998.

[114] YUAN, J.L.; FINE, T.L.; "Neural-Network Design for Small Training Sets of High Dimension", *IEEE Transactions on Neural Networks*, v.9, n.2, pp. 266-280, Mar. 1998.

[115] SENJYU, T.; HIGA, S.; UEZATO, K.; "Future Load Curve Shaping Based on Similarity Using Fuzzy Logic Approach", *IEE Proceedings on Generation, Transmission and Distribution*, v.145, n.4, pp. 375-380, July 1998.

[116] DREZGA, I.; RAHMAN, S.; "Input Variable Selection for Ann-Based Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.13, n.4, pp. 1238-1244, Nov. 1998.

[117] DANESHDOOST, M.; LOTFALIAN, M.; BUMROONGGIT, G.; NGOY, J.P.; "Neural Network with Fuzzy Set-Based Classification for Short-term Load

Forecasting", *IEEE Transactions on Power Systems*, v.13, n.4, pp. 1386-1391, Nov. 1998.

[118] SRINIVASAN, D.; TAN, S.S.; CHANG, C.S.; CHAN, E.K.; "Practical Implementation of a Hybrid Fuzzy Neural Network for One-day Ahead Load Forecasting", *IEE Proceedings on Generation, Transmission and Distribution*, v.145, n.6, pp. 687-692, Nov. 1998.

[119] CHARYTONIUK, W.; NIEBRZYDWSKI, J.; "Confidence Interval Construction for Load Forecast", *Electric Power Systems Research*, v.48, n.2, pp. 97-103, Dec. 1998.

[120] KASSAEI, H.R.; KEYHANI, A.; WOUNG, T.; RAHMAN, M.; "A Hybrid Fuzzy, Neural Network Bus Load Modeling and Predication", *IEEE Transactions on Power Systems*, v.14, n.2, pp. 718-724, May 1999.

[121] YOO, H.; PIMMEL, R.L.; "Short-Term Load Forecasting Using a Self-supervised Adaptive Neural Network", *IEEE Transactions on Power Systems*, v.14, n.2, pp. 779-784, May 1999.

[122] DREZGA, I.; RAHMAN, S.; "Short-term Load Forecasting with Local ANN Predictors", *IEEE Transactions on Power Systems*, v.14, n.3, pp. 844-850, Aug. 1999.

[123] DING, A.A.; "Neural-Network Prediction with Noisy Predictors", *IEEE Transactions on Neural Networks*, v.10, n.5, pp. 1196-1203, Sept. 1999.

[124] BACZYNSKI, D.; PAROL, M.; "Influence of Artificial Neural Network Structure on Quality of Short-Term Electric Energy Consumption Forecast", *IEE Proceedings on Generation, Transmission and Distribution*, v.151, n.2, pp. 241-245, Mar. 2004.

[125] HIPPERT, H.S.; PEDREIRA, C.E.; "Estimating Temperature Profiles for Short-Term Load Forecasting: Neural Networks Compared to Linear Models", *IEE Proceedings on Generation, Transmission and Distribution*, v.151, n.4, pp. 543-547, July 2004.

[126] BECCALI, M.; CELLURA, M., LO BRANO, V.; MARVUGLIA, A.; "Forecasting Daily Urban Electric Load Profiles Using Artificial Neural Networks", *Energy Conversion and Management*, v.45, n.18, pp. 2879-2900, Nov. 2004.

[127] AL-KANDARI, A.M.; SOLIMAN, S.A.; EL-HAWARY, M.E.; "Fuzzy Short-Term Electric Load Forecasting", *International Journal of Electrical Power & Energy Systems*, v.26, n.2, pp. 111-122, Feb. 2004.

[128] DAVID, A.K.; "Load Forecasting under Spot Pricing", *IEE Proceedings C*, v.135, Pt.C, n.5, pp. 369-377, Sept. 1988.

[129] HSU, Y.Y.; YANG, C.C.; "Design of Artificial Neural Networks for Short-Term Load Forecasting. Part I: Self-organizing Feature Maps for Day Type Identification", *IEEE Proceedings C*, v.138, n.5, pp. 407-413, Sept. 1991.

[130] LEE, K.Y.; CHA, Y.T.; PARK, J.H.; "Short-term Load Forecasting Using an Artificial Neural Network", *IEEE Transactions on Power Systems*, v.7, n.1, pp. 124-132, Feb. 1992.

[131] LAMEDICA, R.; PRUDENZI, A.; SFORNA, M.; CACIOTTA, M; CENCELLI, V.O.; "A Neural Network Based Technique For Short-Term Load Forecasting of Anomalous Load Periods", *IEEE Transactions on Power Systems*, v.11, n.4, pp. 1749-1756, Nov. 1996.

[132] KIM, K.H.; YOUN, H.S.; KANG, Y.C.; "Short-term Load Forecasting for Special Days in Anomalous Load Conditions Using Neural Networks and

Fuzzy Inference Method", *IEEE Transactions on Power Systems*, v.15, n.2, pp. 559-565, May 2000.

[133] ALVES DA SILVA, A.P; MOULIN, L.; "Confidence Intervals for Neural Network Based Short-term Load Forecasting", *IEEE Transactions on Power Systems*, v.15, n.4, pp. 1191-1196, Nov. 2000.

[134] OSOWSKI, S.; SIWEK, K.; "Regularization of Neural Networks for Improved Load Forecasting in the Power System", *IEE Proceedings on Generation, Transmission and Distribution*, v.149, n.3, pp. 340-344, May 2002.

[135] DARBELLAY, G.A.; SLAMA, M.; "Forecasting the Short-Term Demand for Electricity: Do Neural Networks Stand a Better Chance?", *International Journal of Forecasting*, v.16, n.1, pp. 71-83, Jan. 2000.

[136] SARGUNARAJ, S.; GUPTA, D.P.S.; DEVI, S.; "Short-term Load Forecasting for Demand Side Management", *IEE Proceedings on Generation, Transmission and Distrtibution*, v.144, n.1, pp. 68-74, Jan. 1997.

[137] BUNN, D.W.; "Forecasting Loads and Prices in Competitive Power Markets", *Proceedings of the IEEE,* v.88, n.2, pp. 163-169, Feb.2000.

[138] DASH, P.K.; LIEW, A.C.; RAMAKRISHNA, G.; "Power-Demand Forecasting Using a Neural Network with an Adaptive Learning Algorithm", *IEE Proceedings on Generation, Transmission and Distribution*, v.142, n.6, pp. 560-568, Nov. 1995.

[139] CHOUEIKI, M.H.; MOUNT-CAMPBELL, C.A.; AHALT, S.C.; "Implementing a Weighted Least Squares Procedure in Training a Neural Network to Solve the Short-term Load Forecasting Problem", *IEEE Transactions on Power Systems*, v.12, n.4, pp. 1689-1694, Nov. 1997.

[140] MORI, H.; YUIHARA, A.; “Deterministic Annealing Clustering for ANN-Based Short-Term Load Forecasting”, *IEEE Transactions on Power Systems*, v.16, n.3, pp. 545-551, Aug. 2001.

[141] ZHANG, B.L.; DONG, Z.Y.; “An Adaptive Neural-Wavelet Model for Short Term Load Forecasting”, *Electric Power Systems Research*, v.59, n.2, pp. 121-129, Sept. 2001.

[142] SENJYU, T.; TAKARA, H.; UEZATO, K.; FUNABASHI, T.; “One-Hour-Ahead Load Forecasting Using Neural Network”, *IEEE Transactions on Power Systems*, v.17, n.1, pp. 113-118, Feb. 2002.

[143] CARPINTEIRO, O.A.S.; REIS, A.J.R.; ALVES DA SILVA, A.P.; “A Hierarchical Neural Model in Short-Term Load Forecasting”, *Applied Soft Computing*, v.4, n.4, pp. 405-412, Sept. 2004.

[144] MORI, H.; KOBAYASHI, H.; “Optimal Fuzzy Inference for Short-term Load Forecasting”, *IEEE Transactions on Power Systems*, v.11, n.1, pp. 390-396, Feb. 1996.

[145] MASTOROCOSTAS, P.A.; THEOCHARIS, J.B.; PETRIDIS, V.S.; “A Constrained Orthogonal Least-Squares Method for Generating TSK Fuzzy Models: Application to Short-Term Load Forecasting”, *Fuzzy Sets and Systems*, v.118, n.2, pp. 215-233, Mar. 2001.

[146] HSU, Y.Y.; HO, K.L.; “Fuzzy Expert Systems: an Application to Short-Term Load Forecasting”, *IEE Proceedings C*, v.139, n.6, pp. 471-477, Nov. 1992.

[147] DASH, P.K.; RAMAKRISHNA, G.; LIEW, A.C.; RAHMAN, S.; “Fuzzy Neural Networks for Time-series Forecasting of Electric Load”, *IEE Proceedings on Generation, Transmission and Distribution*, v.142, n.5, pp. 535-544, Sept. 1995.

[148] DASH, P.K.; LIEW, A.C.; RAHMAN, S.; “Fuzzy Neural Network and Fuzzy Expert System for Load Forecasting”, *IEE Proceedings on Generation, Transmission and Distribution*, v.143, n.1, pp. 106-114, Jan. 1996.

[149] YANG, H.T.; HUANG, C.M.; HUANG, C.L.; “Identification of ARMAX Model for Short Term Load Forecasting: An Evolutionary Programming Approach”, *IEEE Transactions on Power Systems*, v.11, n.1, pp. 403-408, Feb. 1996.

[150] LING, S.H.; LEUNG, F.H.F.; LAM, H.K.; LEE, Y.S.; TAM, P.K.S.; “A Novel Genetic-Algorithm-Based Neural Network for Short-Term Load Forecasting”, *IEEE Transactions on Industrial Electronics*, v.50, n.4, pp. 793-799, Aug. 2003.

[151] HAYKIN, S.; *Redes Neurais: Princípios e Prática*, 2ª. Edição, Porto Alegre, RS, Brasil, Editora Bookman, 2001.

[152] ROSENBLATT, F.; “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain”, *Psychological Review*, v.65, pp. 386-408, 1958.

[153] MINSKY, M.L.; PAPERT, S.A.; *Perceptrons*, Cambridge, Massachusetts, MIT Press, 1969.

[154] RUMELHART, D.E.; HINTON, G.E.; WILLIAMS, R.J.; McCLELLAND, J.L.; “Learning Internal Representations by Error Propagation”, In: RUMELHART, D.E.; McCLELLAND, J.L.; (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of the Cognition*, v.1, chapter 8, Cambridge, Massachusetts, MIT Press, 1986.

[155] WERBOS, P.J.; *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences,* Ph.D. Thesis, Harvard University, Cambridge, Massachusetts, USA, 1974.

[156] WIDROW, B.; HOFF, M.E.; "Adaptive Switching Circuits", In: *IRE WESCON Convention Record,* pp. 96-104, 1960.

[157] LUENBERGER, D.G.; *Introduction to Linear and Nonlinear Programming*, Addison-Wesley Publishing Company, 1973.

[158] BISHOP, C.M.; *Neural Networks for Pattern Recognition*, Oxford, New York, Oxford University Press, 1995.

[159] GIL, P.E.; MURRAY, W.; WRIGHT, M.H.; *Practical Optimization*, New York, Academic Press, 1981.

[160] VAPNIK, V.N.; *Statistical Learning Theory*, New York, John Wiley & Sons, 1998.

[161] SCHÖLKOPF, B.; SMOLA, A.J.; *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, Cambridge, Massachusetts, 2002.

[162] CHERKASSKY, V.; MULIER, F.; *Learning from Data: Concepts, Theory and Methods*, John Wiley & Sons, New York, USA, 1998.

[163] GUO, P.; LYU, M.R.; CHEN, C.L.P.; "Regularization Parameter Estimation for Feedforward Neural Networks", *IEEE Tranasctions on Systems, Man, and Cybernetics – Part B: Cybernetics*, v.33, n.1, pp. 35-44, Feb. 2003.

[164] MACKAY, D.J.C.; *Bayesian Methods for Adaptive Models*, Ph.D. dissertation, California Institute of Technology, Pasadena, California, USA, 1992.

[165] GOUTTE, C.; *Statistical Learning and Regularization for Regression: Application to System Identification and Time Series Modelling*, Ph.D. dissertation, Université Paris 6, Paris, France, 1997.

[166] RACINE, J.; "Feasible Cross-Validatory Model Selection for General Stationary Processes", *Journal of Applied Econometrics*, v.12, pp. 169-179, Mar/Apr 1997.

[167] RACINE, J.; "A Consistent Cross-Validatory Method for Dependent Data: hv-Block Cross-Validation", *Journal of Econometrics*, v.99, n.1, pp. 39-61, Nov 2000.

[168] MURATA, N.; YOSHIZAWA, S.; AMARI, S.I.; "Network Information Criterion – Determining the Number of Hidden Units for an Artificial Neural Network", *IEEE Transactions on Neural Networks*, v.5, n.6, pp. 865-872, Nov. 1994.

[169] SWANSON, N.R.; WHITE, H.; "A Model-Selection Approach to Assessing the Information in the Term Structure Using Linear Models and Artificial Neural Networks", *Journal of Businees and Economic Statistics*, v.13, n.3, pp. 265-275, Jul. 1995.

[170] SWANSON, N.R.; WHITE, H.; "Forecasting Economic Time Series Using Flexible versus Fixed Specification and Linear versus Nonlinear Econometric Models", *International Journal of Forecasting*, v.13, n.4, pp. 439-461, Dec. 1997.

[171] SWANSON, N.R.; WHITE, H.; "A Model Selection Approach to Real-time Macroeconomic Forecasting Using Linear Models and Artificial Neural Networks", *Review of Economic and Statistics*, v.79, pp. 540-550, 1997.

[172] MEDEIROS, M.C.; VEIGA, A.; "A Flexible Coefficient Smooth Transition Time Series Model", *IEEE Transactions on Neural Networks*, v.16, n.1, pp. 97-113, Jan. 2005.

[173] ANDERS, U.; KORN, O.; "Model Selection on Neural Networks", *Neural Networks*, v.12, n.2, pp. 309-323, Mar. 1999.

[174] BISHOP, C.M.; "Exact Calculation of the Hessian Matrix for the Multi-layer Perceptron", *Neural Computation*, v.4, n.4, pp. 494-501, 1992.

[175] TREADGOLD, N.K.; GEDEON, T.D.; "Exploring Constructive Cascade Networks", *IEEE Transactions on Neural Networks*, v.10, n.6, pp. 1335-1350, Nov. 1999.

[176] KWOK, T.Y.; YENUG, D.Y.; "Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems", *IEEE Transactions on Neural Networks*, v.8, pp. 630-645, May 1997.

[177] POGGIO, T.; GIROSI, F.; "Networks for Approximation and Learning", *Proceedings of the IEEE*, v.78, n.9, pp. 1481-1497, Sept. 1990.

[178] BISHOP, C.M.; "Curvature-Driven Smoothing: A Learning Algorithm for Feedforward Networks", *IEEE Transactions on Neural Networks*, v.4, n.5, pp. 882-884, Sept. 1993.

[179] AMARI, S.; MURATA, N.; MÜLLER, K.R.; FINKE, M.;YANG, H.; "Statistical Theory of Overtraining – Is Cross-validation Asymptotically Effective?", *Advances in Neural Information Processing Systems*, v.8, pp. 176-182, 1996.

[180] CATALTEPE, Z.; ABU-MOSTAFA, Y.S.; MAGDON-ISMAIL, M.; "No Free Lunch for Early Stopping", *Neural Computation*, v.11, n.4, pp. 995-1009, May 1999.

[181] REED, R.; MARKS II, R.J.; OH, S.; “Similarities of Error Regularization, Sigmoid Gain Scaling, Target Smoothing, and Training with Jitter”, *IEEE Transactions on Neural Networks*, v.6, n.3, pp. 529-538, May 1995.

[182] CHATFIELD, C.; *The Analysis of Time Series: An Introduction*, 6a. edição, Chapman and Hall/CRC, 2004.

[183] REIS, A.J.R.; ALVES DA SILVA, A.P.. “Feature Extraction Via Multi-Resolution Analysis for Short-Term Load Forecasting”, *IEEE Transactions on Power Systems*, Accepted fo Future Publication, pp. 1-10, Issue: 99, 2004.

[184] CHEN, B.J.; CHANG, M.W.; LIN, C.J.; “Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001”, *IEEE Transactions on Power Systems*, v.19, n.4, pp. 1821-1830, Nov. 2004.

APÊNDICE A – *Perceptron* de múltiplas camadas

1 – *Perceptron* de múltiplas camadas (MLP)

Originalmente proposto por ROSENBLATT [152] em 1958, objetivando uma modelagem probabilística da forma na qual as informações são armazenadas e processadas pelo cérebro, o *perceptron* é a estrutura mais simples de um neurônio, similar a apresentada na Figura 2.1, cuja função de ativação $\varphi(x)$ é conhecida como degrau unitário, dada pela equação (A.1):

$$\varphi(x) = \begin{cases} 1, \ se \ x \geq 0 \\ 0, \ se \ x < 0 \end{cases} \tag{A.1}$$

Esta estrutura é utilizada de maneira eficiente para problemas de classificação de padrões linearmente separáveis em duas classes, conforme comprovado pelo teorema da convergência do *perceptron*, que fornece também um algoritmo para treinamento deste modelo. Para classificação de padrões linearmente separáveis em *m* classes, basta combinar *m perceptrons* na camada de saída do modelo neural, obtendo, assim os chamados *perceptrons* de camada única. A limitação do *perceptron* de *Rosenblatt* foi comprovada matematicamente por MINSKY e PAPERT [153], em 1969, cujas conclusões para os modelos de camada única foram expandidas para as estruturas com múltiplas camadas, diminuindo assim o interesse pelos modelos neurais.

Entretanto, o surgimento do algoritmo de retropropagação do erro, proposto em 1986 por RUMELHART *et. al.* [154], reacendeu o interesse pelas redes neurais *feedforward* com múltiplas camadas. Porém, a aplicação deste algoritmo ao *perceptron* de *Rosenblatt* encontrou empecilho na função de ativação utilizada por este modelo, visto que a mesma não é continuamente diferenciável e o algoritmo de retropropagação do erro requer que as funções de ativação apresentem esta característica. Para contornar este problema, o degrau unitário foi substituído por funções sigmoidais, denominadas

desta maneira devido ao seu formato de S. Dois exemplos de funções sigmoidais são as funções logística e tangente hiperbólica, cujas expressões são dadas pelas equações (A.2) e (A.3):

$$\varphi(x) = \frac{1}{1+e^{-ax}} \tag{A.2}$$

$$\varphi(x) = \tanh(ax) \tag{A.3}$$

A Figura A.1 e a Figura A.2 apresentam o formato das funções logística e tangente hiperbólica, respectivamente, sendo destacado o efeito da variação da constante *a*, conhecida como ganho das respectivas funções. Os algoritmos de retropropagação de erro, a serem apresentados no item 1.1.1 deste apêndice, usualmente utilizam um valor constante para a, com um valor sugerido sendo $a = 2/3$ [151]. Entretanto, o valor deste parâmetro pode contribuir para o aumento da capacidade de generalização de MLP's, conforme apresentado no item 2.2.3.3.

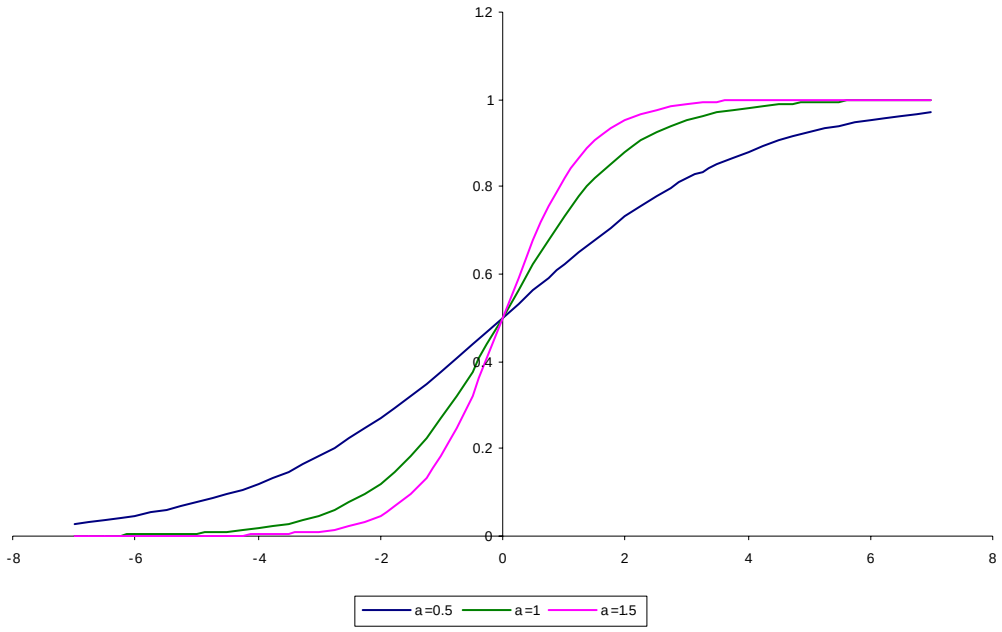

Figura A.1 – Função logística

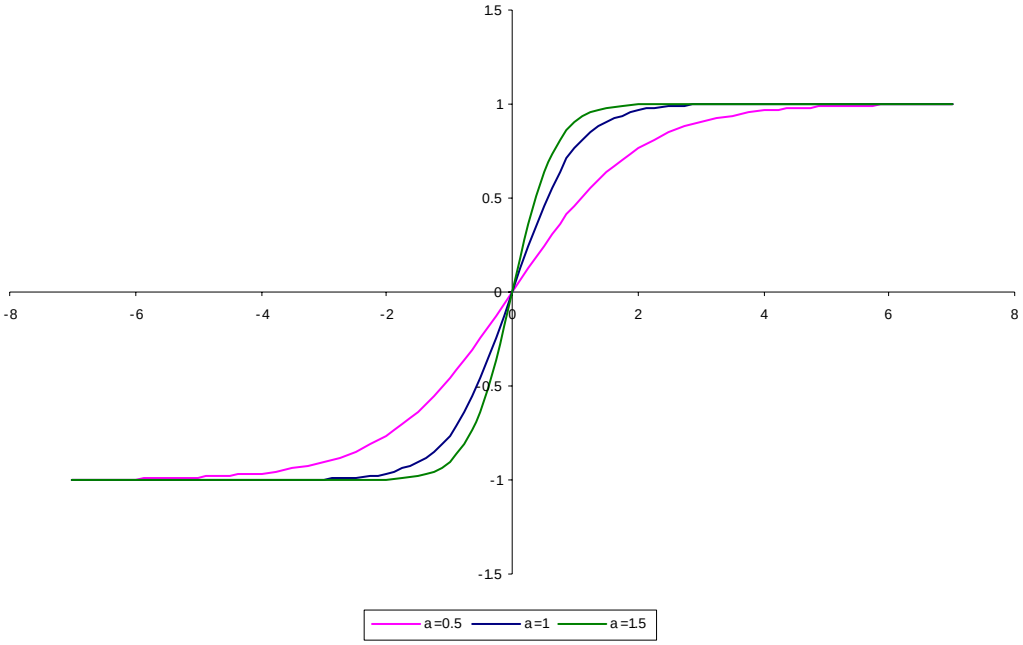

Figura A.2 – Função tangente hiperbólica

O trabalho de RUMELHART *et. al.* [154] deu origem aos modelos popularmente conhecidos como *perceptrons* de múltiplas camadas (MLP), cuja estrutura é semelhante a apresentada na Figura 2.2, porém com os neurônios das camadas ocultas apresentando necessariamente funções de ativação sigmoidais. Os neurônios da camada de saída podem apresentar função de ativação sigmoidal ou linear, com a primeira sendo comumente utilizada para problemas de classificação e a segunda para problemas de aproximação funcional ou de regressão.

Matematicamente, a saída de um MLP é dada pela equação (A.4):

$$
\begin{aligned}
& c_{0i} = x_i, i = 1, 2, ..., n_0, \ n_0 = n \\
& c_{ki} = \varphi_k \left(\sum_{j=1}^{n_{s-1}} w_{kij} c_{(k-1)j} + b_{ki} \right), i = 1, 2, ..., n_k, \ k = 1, 2, ..., s+1 \\
& c_{(s+1)i} = f_i \left(\underline{x}, \underline{w} \right), i = 1, 2, ..., n_{s+1}, \ n_{s+1} = m
\end{aligned}
\tag{A.4}
$$

onde,

- $n \rightarrow$ número de entradas do modelo;
- $s \rightarrow$ número de camadas ocultas;

- $m \rightarrow$ número de saídas do modelo;
- $\underline{x} = \left[x_1, x_2, ..., x_n \right]^t \rightarrow$ vetor contendo as *n* entradas do modelo;
- $c_{0i} \rightarrow$ saída do *i*-ésimo neurônio da camada de entrada, $n_0 = n$;
- $n_k \rightarrow$ número de neurônios da *k*-ésima camada oculta;
- $c_{ki} \rightarrow$ saída do *i*-ésimo neurônio da *k*-ésima camada oculta, $k \neq 0$;
- $\varphi_k \rightarrow$ função de ativação dos neurônios da *k*-ésima camada oculta;
- $w_{kij} \rightarrow$ pesos sinápticos que ligam a saída do *j*-ésimo neurônio da camada $(k-1)$ à entrada do *i*-ésimo neurônio da camada *k*;
- $b_{ki} \rightarrow$ *bias* do *i*-ésimo neurônio da *k*-ésima camada;
- $\underline{w} \in \mathbb{R}^M \rightarrow$ vetor coluna contendo todos os pesos sinápticos w_{kij} e *bias* b_{ki} do modelo;
- $M = \sum_{k=0}^{s} n_{k+1}\left(n_k + 1\right) \rightarrow$ número de parâmetros do modelo;
- $f_i\left(\underline{x}, \underline{w}\right) \rightarrow$ *i*-ésima saída calculada pelo modelo.

A equação (A.4) pode ser modificada, através da inclusão dos bias b_{ki} como pesos sinápticos que ligam uma entrada constante, unitária, ao *i*-ésimo neurônio da *k*-ésima camada. Com isso, a equação (A.4) passa a ser dada por:

$$
\begin{aligned}
&c_{0i} = x_i, i = 1, 2, ..., n_0, \ n_0 = n \\
&c_{ki} = \varphi_k\left(\sum_{j=0}^{n_{s-1}} w_{kij} c_{(k-1)j} \right), i = 1, 2, ..., n_k \\
&c_{(k-1)0} = 1, \ k = 1, 2, ..., s+1 \\
&c_{(s+1)i} = f_i\left(\underline{x}, \underline{w}\right), i = 1, 2, ..., n_{s+1}, \ n_{s+1} = m
\end{aligned}
\qquad \text{(A.5)}
$$

Para problemas de aproximação funcional, categoria onde o problema de previsão de carga a curto prazo pode ser incluído, o teorema da aproximação universal

demonstra que MLP's com uma única camada escondida, contendo número suficiente de neurônios, e uma única saída linear, são suficientes para aproximação de qualquer função contínua não-linear. A saída desta estrutura é dada por:

$$
\begin{aligned}
&c_{0i} = x_i, i = 1, 2, ..., n_0,\ n_0 = n \\
&c_{1i} = \varphi_1\left(\sum_{j=0}^{n_0} w_{1ij} c_{0j}\right), i = 1, 2, ..., n_1 \\
&c_{00} = c_{10} = 1 \\
&f\left(\underline{x}, \underline{w}\right) = c_{21} = \sum_{j=0}^{n_1} w_{21j} c_{1j}
\end{aligned}
\tag{A.6}
$$

A equação (A.6) é um caso particular da equação (A.5), para $s = 1$, $m = 1$ e $\varphi_2\left(x\right) = x$, com $\varphi_1\left(x\right)$ sendo uma função sigmoidal e $f\left(\underline{x}, \underline{w}\right)$ representando a aproximação da função de interesse calculada pelo MLP. Em virtude das características de aproximação universal desta estrutura, somente estes MLP's serão utilizados ao longo desta tese.

Após o surgimento do algoritmo de retropropagação de erro para treinamento de MLP's, diversos algoritmos foram propostos para treinamento destes modelos. A próxima seção abordará alguns deles, começando pelo tradicional algoritmo de retropropagação de erro, chegando aos chamados métodos de segunda ordem.

1.1 – Algoritmos de treinamento

Conforme citado anteriormente, RUMELHART *et. al.* [154] desenvolveram o algoritmo de retropropagação do erro para treinamento de modelos neurais *feedforward* com múltiplas camadas, dando origem ao MLP. Na realidade, este algoritmo foi originalmente desenvolvido por WERBOS [155], em 1974, podendo também ser considerado como uma generalização do algoritmo do mínimo quadrado médio, *least mean square* (LMS), também conhecido como regra delta, desenvolvido por WIDROW e HOFF [156] para filtragem linear adaptativa de sinais. O algoritmo LMS é um caso

particular do algoritmo de retropropagação do erro, para o caso em que a rede apresenta um único neurônio linear.

Após o surgimento deste algoritmo, vários algoritmos para treinamento de MLP's foram propostos, porém com uma abordagem diferente. Neste novo contexto, o treinamento de MLP's passou a ser visto como um problema de otimização, com algumas técnicas desta área do conhecimento sendo aplicadas à estimação de parâmetros de modelos neurais, dando origem aos chamados métodos de segunda ordem, como os métodos *quasi-newton* e os métodos de gradiente conjugado.

Os próximos itens apresentarão uma breve descrição das duas abordagens para treinamento de MLP's, começando pelo algoritmo de retropropagação do erro.

1.1.1 - Algoritmo de retropropagação do erro

O algoritmo de retropropagação de erro é um algoritmo supervisionado, visto que necessita de um conjunto de saídas desejadas para estimação dos parâmetros do modelo através da correção do erro gerado para cada saída. Dado um conjunto D contendo N pares entrada-saída, $D = \{\underline{x}_k, \underline{d}_k\}$, $k = 1, 2, ..., N$, $\underline{x}_k = [x_{k1}, x_{k2}, ..., x_{kn}]^t$, $\underline{d}_k = [d_{k1}, d_{k2}, ..., d_{km}]^t$, o objetivo deste algoritmo reside na estimação do vetor de parâmetros $\underline{w}$ que minimize o erro médio quadrático para este conjunto de dados, também conhecido como risco empírico, dado por:

$$E_s(\underline{w}) = \frac{1}{N}\sum_{k=1}^{N} E_k(\underline{w}) \qquad \text{(A.7)}$$

$$E_k(\underline{w}) = \frac{1}{2}\sum_{j=1}^{m}\left[d_{kj} - f_j(\underline{x}_k, \underline{w})\right]^2$$

Para os MLP's utilizados nesta tese, que apresentam uma única saída ($m = 1$), a equação (A.7) resulta em:

$$E_s\left(\underline{w}\right)=\frac{1}{N}\sum_{i=1}^{N}E_i\left(\underline{w}\right) \tag{A.8}$$

$$E_k\left(\underline{w}\right)=\frac{1}{2}\left[d_k-f\left(\underline{x}_k,\underline{w}\right)\right]^2$$

Visto que o erro só pode ser obtido diretamente para os neurônios da camada de saída, a idéia do algoritmo reside na propagação deste erro através da rede, fazendo com que o MLP, além de propagar os sinais de entrada "para frente", propague os sinais de erro em sentido contrário, objetivando a modificação dos pesos sinápticos e dos *bias* com de forma a minimizar o funcional descrito na equação (A.7). Daí o nome de retropropagação do erro. A derivação deste algoritmo pode ser encontrada em [151], [154], [155] e [158].

Para os MLP's utilizados nesta tese, que apresentam uma única camada escondida e uma única saída linear, este algoritmo pode ser resumido como segue:

1. Faça $l=0$;
2. Inicialize o vetor de parâmetros $\underline{w}(l)$;
3. Apresente o conjunto de treinamento $D=\left\{\underline{x}_k,d_k\right\}$ ao modelo;
4. Para cada par entrada-saída $\left\{\underline{x}_k,d_k\right\}$, efetue os passos 5 a 10;
5. Propague o vetor de entrada $\underline{x}_k$ ao longo da rede, utilizando a equação (A.6);
6. Calcule o erro obtido para este padrão, dado pela equação:

$$e(l)=d_k-f\left[\underline{x}_k,\underline{w}(l)\right] \tag{A.9}$$

7. Atualize os pesos sinápticos w_{21j} que ligam os neurônios da camada intermediária ao neurônio de saída, através da equação:

$$w_{21j}(l+1) = w_{21j}(l) - \eta \left. \frac{\partial E_k(\underline{w})}{\partial w_{21j}} \right|_{\underline{w}=\underline{w}(l)} \tag{A.10}$$

$$\left. \frac{\partial E_k(\underline{w})}{\partial w_{21j}} \right|_{\underline{w}=\underline{w}(l)} = -e(l)\, f\left[\underline{x}_k, \underline{w}(l)\right]$$

8. Atualize os pesos sinápticos w_{1ij} que ligam as entradas aos neurônios sigmoidais da camada intermediária, dada pela equação:

$$w_{1ij}(l+1) = w_{1ij}(l) - \eta \left. \frac{\partial E_k(\underline{w})}{\partial w_{1ij}} \right|_{\underline{w}=\underline{w}(l)} \tag{A.11}$$

$$\left. \frac{\partial E_k(\underline{w})}{\partial w_{1ij}} \right|_{\underline{w}=\underline{w}(l)} = -\left[\left. \frac{d\varphi(a)}{da} \right|_{a=a(l)} \right] w_{21j}(l)\, e(l)\, x_{kj}$$

$$a(l) = \sum_{j=0}^{n_0} w_{1ij}(l)\, x_{kj}$$

9. Faça $l = l+1$;

10. Se todos os padrões $\{\underline{x}_k, d_k\}$ foram apresentados ao modelo, vá para o passo 11. Do contrário, escolha um novo padrão $\{\underline{x}_k, d_k\}$ e retorne ao passo 4;

11. Se o critério de parada for atendido, encerre o algoritmo. Do contrário, retorne ao passo 3.

No algoritmo resumido acima, η representa um parâmetro chamado de taxa de aprendizagem e $d\varphi(a)/da$ a derivada da função de ativação sigmoidal dos neurônios da camada oculta em relação ao somatório ponderado das suas entradas. Como critérios de parada, são utilizados: erro médio para todo o conjunto de treinamento, número máximo de apresentações (épocas) do conjunto de treinamento e erro para um conjunto independente de dados. O algoritmo apresentado acima é conhecido como algoritmo de retropropagação do erro seqüencial, visto que os pesos são atualizados após a apresentação de cada par entrada-saída $\{\underline{x}_i, d_i\}$. A atualização dos pesos pode também

ser feita após a apresentação de uma época inteira de treinamento, dando origem ao chamado treinamento por batelada, ou lote. O algoritmo deste modo de treinamento de MLP's pode ser resumido como segue:

1. Faça $l = 0$;
2. Inicialize o vetor de parâmetros $\underline{w}(l)$;
3. Apresente o conjunto de treinamento $D = \{\underline{x}_k, d_k\}$ ao modelo;
4. Para cada par entrada-saída $\{\underline{x}_k, d_k\}$, efetue os passos 5 a 9;
5. Propague o vetor de entrada $\underline{x}_k$ ao longo da rede, utilizando a equação (A.6);
6. Calcule o erro obtido para este padrão, dado pela equação:

$$e(l) = d_k - f\left[\underline{x}_k, \underline{w}(l)\right] \tag{A.12}$$

7. Calcule as derivadas parciais $\partial E_k(\underline{w})/\partial w_{21j}$, relacionadas com os pesos sinápticos w_{21j} que ligam os neurônios da camada intermediária ao neurônio de saída, através da equação:

$$\left.\frac{\partial E_k(\underline{w})}{\partial w_{21j}}\right|_{\underline{w}=\underline{w}(l)} = -e(l)\, f\left[\underline{x}_k, \underline{w}(l)\right] \tag{A.13}$$

8. Calcule as derivadas parciais $\partial E_k(\underline{w})/\partial w_{1ij}$, relacionadas com os pesos sinápticos w_{1ij} que ligam as entradas aos neurônios sigmoidais da camada intermediária, através da equação:

$$\left.\frac{\partial E_k(\underline{w})}{\partial w_{1ij}}\right|_{\underline{w}=\underline{w}(l)} = -\left[\left.\frac{d\varphi(a)}{da}\right|_{a=a(l)}\right] w_{21j}(l)\, e(l)\, x_{kj} \tag{A.14}$$

$$a(l) = \sum_{j=0}^{n_0} w_{1ij}(l)\, x_{kj}$$

9. Se todos os padrões $\{\underline{x}_k, d_k\}$ foram apresentados ao modelo, vá para o passo 12. Do contrário, escolha o próximo padrão $\{\underline{x}_k, d_k\}$ do conjunto de treinamento *D* e retorne ao passo 4;
10. Atualize os pesos sinápticos w_{21j} que ligam os neurônios da camada intermediária ao neurônio de saída, através da equação:

$$w_{21j}(l+1) = w_{21j}(l) - \eta \left.\frac{\partial E_s(\underline{w})}{\partial w_{21j}}\right|_{\underline{w}=\underline{w}(l)} \tag{A.15}$$

$$\left.\frac{\partial E_s(\underline{w})}{\partial w_{21j}}\right|_{\underline{w}=\underline{w}(l)} = -\frac{1}{N}\sum_{k=1}^{N} \left.\frac{\partial E_k(\underline{w})}{\partial w_{21j}}\right|_{\underline{w}=\underline{w}(l)}$$

11. Atualize os pesos sinápticos w_{1ij} que ligam as entradas aos neurônios sigmoidais da camada intermediária, dada pela equação:

$$w_{1ij}(l+1) = w_{1ij}(l) - \eta \left.\frac{\partial E_s(\underline{w})}{\partial w_{1ij}}\right|_{\underline{w}=\underline{w}(l)} \tag{A.16}$$

$$\left.\frac{\partial E_s(\underline{w})}{\partial w_{1ij}}\right|_{\underline{w}=\underline{w}(l)} = -\frac{1}{N}\sum_{k=1}^{N} \left.\frac{\partial E_k(\underline{w})}{\partial w_{1ij}}\right|_{\underline{w}=\underline{w}(l)}$$

12. Faça $l = l+1$;
13. Se o critério de parada for atendido, encerre o algoritmo. Do contrário, retorne ao passo 3.

Assim como toda técnica baseada em descida em gradiente, categoria na qual o algoritmo de retropropagação de erro está enquadrado, conforme evidenciado nas equações (A.10) e (A.11), este algoritmo apresenta uma série de desvantagens, visto que a função a ser minimizada, apesar de quadrática em relação às saídas, é extremamente não-linear em relação aos pesos, que são os parâmetros que devem ser efetivamente manipulados para minimização do funcional descrito na equação (A.8), acarretando na existência de diversos mínimos locais, comprometendo o desempenho do algoritmo.

Para contornar esse problema, existem uma série de heurísticas propostas para modificação do algoritmo resumido acima, como inserção de uma parcela de momento nas equações (A.10) e (A.11), para o treinamento seqüencial, e (A.14) e (A.15), para o treinamento por batelada, normalização do conjunto de entrada-saída no intervalo [-1;1] e estratégias de escolha do conjunto inicial de pesos $\underline{w}_0$. Com o intuito de sobrepujar as limitações do algoritmo de retropropagação de erro, foram desenvolvidos os chamados métodos de segunda ordem.

1.1.2 - Métodos de segunda ordem

O treinamento supervisionado de MLP's pode ser visto também como um problema de otimização. Para tanto, seja a expansão, em séries de *Taylor*, do funcional descrito na equação (A.7), desprezando os termos de ordem de superior, em torno de um ponto específico $\underline{w}(l)$ no espaço de pesos, dada por:

$$E_s\left[\underline{w}(l)+\Delta\underline{w}(l)\right]=E_s\left[\underline{w}(l)\right]+\left[\left.\frac{\partial E_s(\underline{w})}{\partial\underline{w}}\right|_{\underline{w}=\underline{w}(l)}\right]^t\Delta\underline{w}(l)+\frac{1}{2}\Delta\underline{w}^t(l)\left[\left.\frac{\partial^2 E_s(\underline{w})}{\partial\underline{w}^2}\right|_{\underline{w}=\underline{w}(l)}\right]\Delta\underline{w}(l) \quad \text{(A.17)}$$

$$\nabla E_s(\underline{w})=\frac{\partial E_s(\underline{w})}{\partial\underline{w}}=\left[\frac{\partial E_s(\underline{w})}{\partial w_1},\frac{\partial E_s(\underline{w})}{\partial w_2},\cdots,\frac{\partial E_s(\underline{w})}{\partial w_M}\right]^t$$

$$\underline{\underline{H}}(\underline{w})=\frac{\partial^2 E_s(\underline{w})}{\partial\underline{w}^2}=\begin{bmatrix}\frac{\partial^2 E_s(\underline{w})}{\partial w_1^2} & \frac{\partial^2 E_s(\underline{w})}{\partial w_1\partial w_2} & \cdots & \frac{\partial^2 E_s(\underline{w})}{\partial w_1\partial w_M}\\ \frac{\partial^2 E_s(\underline{w})}{\partial w_2\partial w_1} & \frac{\partial^2 E_s(\underline{w})}{\partial w_2^2} & \cdots & \frac{\partial^2 E_s(\underline{w})}{\partial w_2\partial w_M}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2 E_s(\underline{w})}{\partial w_M\partial w_1} & \frac{\partial^2 E_s(\underline{w})}{\partial w_M\partial w_2} & \cdots & \frac{\partial^2 E_s(\underline{w})}{\partial w_M^2}\end{bmatrix}$$

Na equação (A.17), $\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}$ representa o vetor gradiente, calculado no ponto $\underline{w}(l)$ e $\underline{\underline{H}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}$ a matriz *hessiana* calculada no mesmo ponto. A expressão (A.17) realiza uma aproximação quadrática, em torno do ponto $\underline{w}(l)$, da superfície de erro $E_s(\underline{w})$ no espaço de pesos. Portanto, a modificação $\Delta\underline{w}(l)$ que deve ser aplicada aos

pesos sinápticos $\underline{w}(l)$ de forma a obter o ponto de mínimo dessa superfície quadrática aproximada é dada por:

$$\nabla E_s\left[\underline{w}(l)+\Delta\underline{w}(l)\right]=\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}+\left[\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]\Delta\underline{w}(l)=0 \tag{A.18}$$

$$\Delta\underline{w}(l)=-\left[\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]^{-1}\left[\nabla E_s\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]$$

Na equação (A.18), $\left[\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}\right]^{-1}$ representa a inversa da matriz *hessiana*. A equação (A.18) é o princípio do método de *Newton*, apresentando as seguintes desvantagens quando aplicada diretamente ao treinamento de MLP's:

- Ausência de garantia da existência da inversa da matriz *hessiana* $\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}$, devido à possibilidade de existência de colunas desta matriz linearmente dependentes;
- Se $\underline{\underline{H}}\left(\underline{w}\right)$ for inversível, o cálculo da sua inversa pode ser computacionalmente custoso para problemas de grande porte;
- A modificação $\Delta\underline{w}(l)$ dada pela equação (A.18) conduz à minimização do funcional $E_s\left(\underline{w}\right)$, ou seja, $E_s\left[\underline{w}(l)+\Delta\underline{w}(l)\right]<E_s\left[\underline{w}(l)\right]$, somente para os casos em que a matriz $\underline{\underline{H}}\left(\underline{w}\right)\Big|_{\underline{w}=\underline{w}(l)}$ é positiva definida, ou seja, com todos auto-valores maiores que zero, o que não é sempre válido para o MLP;
- A convergência do método de *Newton* é garantida apenas para casos em que o funcional $E_s\left(\underline{w}\right)$ é quadrático em relação aos parâmetros $\underline{w}$, convergindo em uma única iteração. Entretanto, essa condição não é satisfeita para o MLP.

Apesar das deficiências acima citadas da aplicação direta do método de *Newton* para treinamento de MLP's, algumas das suas características vantajosas podem ser obtidas através da aplicação dos chamados métodos *quasi-newton*, que não requerem o

cálculo direto de $\left[\underline{\underline{H}}\left(\underline{w}\right)\right]^{-1}$, e sim uma estimativa desta utilizando apenas informação do gradiente $\nabla E_s\left(\underline{w}\right)$. Uma outra forma de utilizar informação de segunda ordem para treinamento de MLP's reside na aplicação dos métodos baseados em gradiente conjugado, que buscam a combinação entre a descida em gradiente, base do algoritmo de retropropagação de erro apresentado no item 1.1.1 deste apêndice, e a informação de segunda ordem contida na matriz *hessiana* $\underline{\underline{H}}\left(\underline{w}\right)$, sem a necessidade de cálculo explícito da mesma.

Os métodos baseados em gradiente conjugado, também conhecidos como métodos de direção conjugada [157], podem ser considerados como métodos intermediários entre aqueles baseados em descida em gradiente e o método de *Newton*. Estes métodos foram desenvolvidos com o intuito de acelerar a convergência da descida em gradiente, sem o esforço computacional extensivo necessário para a aplicação do método de *Newton.* A derivação do método de otimização baseado em gradiente conjugado pode ser encontrada em [157] e [159], com a sua aplicação ao treinamento de MLP's podendo ser encontrada em [151] e [158].

De uma maneira geral, o algoritmo para treinamento de MLP's baseado em gradiente conjugado pode ser resumido como segue [158]:

1. Faça $l = 0$;
2. Escolha o vetor inicial de pesos $\underline{w}\left(l\right)$;
3. Utilizando o algoritmo de retropropagação do erro por batelada, calcule $\nabla E_s\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)}$;
4. Determine a direção inicial de busca através da equação:

$$\underline{d}\left(l\right) = -\nabla E_s\left(\underline{w}\right)\big|_{\underline{w}=\underline{w}(l)} \tag{A.19}$$

5. Resolva o problema de otimização irrestrito dado por:

$$\min_{\alpha(l)} E_s\left[\underline{w}(l)+\alpha(l)\underline{d}(l)\right] \tag{A.20}$$

6. Obtida a solução $\alpha_{\min}(l)$ do problema de otimização descrito na equação (A.20), atualize o vetor de pesos $\underline{w}$ através da equação:

$$\underline{w}(l+1)=\underline{w}(l)+\alpha_{\min}(l)d(l) \tag{A.21}$$

7. Se o critério de parada for atendido para $\underline{w}=\underline{w}(l+1)$, encerre o algoritmo. Do contrário, vá para o passo 8;
8. Utilizando o algoritmo de retropropagação do erro por batelada, calcule $\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}$;
9. Calcule a nova direção de busca através da equação:

$$\underline{d}(l+1)=-\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}+\beta(l)\underline{d}(l) \tag{A.22}$$

10. Faça $l=l+1$ e retorne ao passo 5.

No algoritmo descrito acima, os parâmetros $\alpha_{\min}(l)$ e $\beta(l)$ são responsáveis pelo passo e pela direção da busca, respectivamente. Enquanto $\alpha_{\min}(l)$ é obtido através da solução do problema de minimização dado por (A.20), duas expressões podem ser utilizadas para obtenção do parâmetro $\beta(l)$. Uma delas, conhecida como fórmula de *Polak-Ribiere*, é dada por [158]:

$$\beta(l)=\frac{\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}\right]^t\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}-\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]}{\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]^t\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]} \tag{A.23}$$

A segunda, chamada de fórmula de *Fletcher-Reeves*, é dada por [158]:

$$\beta(l) = \frac{\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}\right]^t \left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}\right]}{\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]^t \left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]} \tag{A.24}$$

As expressões (A.23) e (A.24) são equivalentes para problemas de otimização quadrática, onde o treinamento de MLP's não está inserido. No contexto de MLP's, a fórmula de *Polak-Ribiere* apresenta melhores resultados, visto que, a medida em que são obtidos sucessivos vetores $\nabla E_s(\underline{w})$ similares ao longo do algoritmo, $\beta(l)$ tende a zero, reiniciando a busca na forma de descida em gradiente [158]. Entretanto, a convergência do algoritmo baseado em gradiente conjugado utilizando a equação (A.23) é garantida apenas se esta equação sofrer a seguinte modificação:

$$\beta(l) = \max\left\{\frac{\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}\right]^t \left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)} - \nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]}{\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]^t \left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]}, 0\right\} \tag{A.25}$$

Pela expressão (A.25), o algoritmo baseado em gradiente conjugado deve ser reiniciado, começando a busca utilizando a direção da descida em gradiente, para $\beta(l) \leq 0$ [151].

Os chamados métodos *quasi-newton*, como o próprio nome já diz, utilizam a idéia básica do método de *Newton*, buscando superar as deficiências do mesmo quando aplicado ao treinamento de MLP's. Nestes métodos, é calculada uma estimativa da matriz *hessiana* $\underline{\underline{H}}(\underline{w})$ utilizando apenas informações do gradiente $\nabla E_s(\underline{w})$. Para tal, seja $E_s(\underline{w}): \mathbb{R}^M \rightarrow \mathbb{R}$, um funcional com derivadas de segunda ordem contínuas, dois pontos consecutivos, $\underline{w}(l+1)$ e $\underline{w}(l)$, e uma constante θ, $0 < \theta < 1$. Pelo teorema do valor médio, a seguinte expressão é obtida [157]:

$$\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)} - \nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)} = \left\{\underline{\underline{H}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)+\theta\left[\underline{w}(l+1)-\underline{w}(l)\right]}\right\}\left[\underline{w}(l+1) - \underline{w}(l)\right] \tag{A.26}$$

Para o caso em que a matriz *hessiana* $\underline{\underline{H}}(\underline{w})$ é constante, pressuposto inicial dos métodos *quasi-newton* [158], a equação (A.26) passa a ser dada por:

$$\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)} - \nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)} = \underline{\underline{H}}(\underline{w})\left[\underline{w}(l+1) - \underline{w}(l)\right] \tag{A.27}$$

A equação (A.27) mostra que o cálculo do gradiente $\nabla E_s(\underline{w})$ em dois pontos consecutivos fornece informação sobre a matriz *hessiana* $\underline{\underline{H}}(\underline{w})$. Sejam $\underline{\underline{P}}(\underline{w})$ e $\underline{\underline{Q}}(\underline{w})$ matrizes de dimensão $M \times M$, dadas por:

$$\underline{\underline{P}}(\underline{w}) = \begin{bmatrix} w_1(1) - w_1(0) & w_1(2) - w_1(1) & \cdots & w_1(M) - w_1(M-1) \\ w_2(1) - w_2(0) & w_1(2) - w_1(1) & \cdots & w_2(M) - w_2(M-1) \\ \vdots & \vdots & \ddots & \vdots \\ w_M(1) - w_M(0) & w_M(2) - w_M(1) & \cdots & w_M(M) - w_M(M-1) \end{bmatrix} \tag{A.28}$$

$$\underline{\underline{Q}}(\underline{w}) = \begin{bmatrix} \dfrac{\partial E_s(\underline{w})}{\partial w_1}\Big|_{\underline{w}(1)} - \dfrac{\partial E_s(\underline{w})}{\partial w_1}\Big|_{\underline{w}(0)} & \cdots & \dfrac{\partial E_s(\underline{w})}{\partial w_1}\Big|_{\underline{w}(M)} - \dfrac{\partial E_s(\underline{w})}{\partial w_1}\Big|_{\underline{w}(M-1)} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial E_s(\underline{w})}{\partial w_M}\Big|_{\underline{w}(1)} - \dfrac{\partial E_s(\underline{w})}{\partial w_M}\Big|_{\underline{w}(0)} & \cdots & \dfrac{\partial E_s(\underline{w})}{\partial w_M}\Big|_{\underline{w}(M)} - \dfrac{\partial E_s(\underline{w})}{\partial w_M}\Big|_{\underline{w}(M-1)} \end{bmatrix} \tag{A.29}$$

Se as *M* direções $\underline{w}(n+1) - \underline{w}(n)$ forem linearmente independentes, utilizando a expressão (A.27), a matriz *hessiana* $\underline{\underline{H}}(\underline{w})$ pode ser estimada pela equação:

$$\underline{\underline{H}}(\underline{w}) = \underline{\underline{Q}}(\underline{w})\left[\underline{\underline{P}}(\underline{w})\right]^{-1} \tag{A.30}$$

Desta forma, a estimativa $\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}$ da matriz *hessiana* $\underline{\underline{H}}(\underline{w})$ para a $(n+1)$-ésima iteração é dada por:

$$\left[\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(n+1)}\right]\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)} - \nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right] = \underline{w}(l+1) - \underline{w}(l),\ l = 0,1,...,M-1 \tag{A.31}$$

A cada iteração da equação (A.31), é obtido um sistema linear contendo *M* equações e M^2 incógnitas, referentes aos $M \times M$ elementos da matriz $\underline{\underline{S}}(\underline{w})$. Portanto,

após a aplicação de *M* direções $\underline{w}(l+1)-\underline{w}(l)$ linearmente independentes, é obtido um sistema linear contendo M^2 equações e M^2 incógnitas, cuja solução fornece a estimativa final $\underline{\underline{S}}(\underline{w})=\left[\underline{\underline{H}}(\underline{w})\right]^{-1}$. Porém, se o número de direções linearmente independentes for menor que *M*, o sistema linear obtido apresenta mais equações que incógnitas, resultando em infinitas soluções para a matriz $\underline{\underline{S}}(\underline{w})$ [157].

Para abordar o caso em que são possíveis infinitas soluções para a estimativa $\underline{\underline{S}}(\underline{w})$, foi proposto o método de *Davidon-Fletcher-Powell* (DFP), que, para treinamento de MLP's, pode ser resumido da forma que segue:

1. Faça $l=0$;
2. Escolha uma matriz de dimensão $M\times M$ definida positiva como estimativa inicial da matriz $\underline{\underline{S}}(\underline{w})$;
3. Escolha o vetor inicial de parâmetros $\underline{w}(l)$;
4. Utilizando o algoritmo de retropropagação do erro por batelada, calcule $\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}$;
5. Faça $\underline{d}(l)=-\underline{\underline{S}}(\underline{w})\left[\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]$;
6. Resolva o problema de otimização dado por:

$$\begin{aligned}&\min_{\alpha} E_s\left[\underline{w}(l)+\alpha(l)\underline{d}(l)\right]\\ &s.a\\ &\alpha\geq 0\end{aligned} \tag{A.32}$$

7. Obtida a solução $\alpha_{\min}(l)$ do problema de otimização descrito na equação (A.32), atualize o vetor de pesos $\underline{w}$ através da equação:

$$\underline{w}(l+1)=\underline{w}(l)+\alpha_{\min}(l)\underline{d}(l) \tag{A.33}$$

8. Se o critério de parada for atendido para $\underline{w} = \underline{w}(l+1)$, encerre o algoritmo. Do contrário, vá para o passo 9;
9. Utilizando o algoritmo de retropropagação do erro por batelada, calcule $\nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}$;
10. Calcule a nova estimativa $\underline{\underline{S}}(\underline{w})$ através da equação:

$$\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)} = \underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)} + \frac{\left[\alpha_{\min}(l)\underline{d}(l)\right]\left[\alpha_{\min}(l)\underline{d}(l)\right]^t}{\left[\alpha_{\min}(l)\underline{d}(l)\right]^t\left[\underline{q}(l)\right]} - \frac{\left[\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}(l)\right]\left[\underline{q}(l)\right]^t\left[\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]}{\left[\underline{q}(l)\right]^t\left[\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}(l)\right]} \tag{A.34}$$

$$\underline{q}(l) = \nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)} - \nabla E_s(\underline{w})\big|_{\underline{w}=\underline{w}(l)}$$

11. Faça $l = l+1$ e retorne ao passo 5.

O algoritmo descrito acima foi o primeiro dos chamados métodos *quasi-newton* de otimização. Atualmente, o melhor método *quasi-newton* é o chamado método de *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) [151], cuja única modificação em relação ao DFP reside na estimativa recursiva da matriz $\underline{\underline{S}}(\underline{w})$, originalmente dada pela equação (A.34), que passa a ser dada por:

$$\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)} = \underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)} + \frac{\left[\alpha_{\min}(l)\underline{d}(l)\right]\left[\alpha_{\min}(l)\underline{d}(l)\right]^t}{\left[\alpha_{\min}(l)\underline{d}(l)\right]^t\left[\underline{q}(l)\right]} - \frac{\left[\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}(l)\right]\left[\underline{q}(l)\right]^t\left[\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]}{\left[\underline{q}(l)\right]^t\left[\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}(l)\right]} + \left[\underline{q}(l)\right]^t\left[\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}(l)\right]\left[\underline{u}(l)\right]\left[\underline{u}(l)\right]^t \tag{A.35}$$

Na equação (A.35), $\underline{u}(l)$ é dado por:

$$\underline{u}(l) = \frac{\alpha_{\min}(l)\underline{d}(l)}{\left[\alpha_{\min}(l)\underline{d}(l)\right]^t\left[\underline{q}(l)\right]} - \frac{\left[\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}(l)\right]}{\left[\underline{q}(l)\right]^t\left[\underline{\underline{S}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]\left[\underline{q}(l)\right]} \tag{A.36}$$

Tanto o método baseado em gradiente conjugado quanto os métodos *quasi-newton* utilizam a cada iteração uma aproximação quadrática, em torno do ponto $\underline{w}(l)$, de um funcional arbitrário $E_s(\underline{w})$. Para o caso específico em que $E_s(\underline{w})$ é dado pela equação (A.7), ou seja, para problemas de minimização do erro médio quadrático, onde o treinamento de MLP's está inserido, existe o método de *Levenberg-Marquardt*, que, assim como os métodos *quasi-newton*, utiliza uma aproximação da matriz *hessiana* $\underline{\underline{H}}(\underline{w})$ tomando por base informações do gradiente e do erro para cada padrão. Seja o erro quadrático $E_k(\underline{w})$ dado pela equação (A.7) e repetido a seguir:

$$E_k(\underline{w}) = \frac{1}{2}\sum_{j=1}^{m} e_j(\underline{w})^2 \tag{A.37}$$
$$e_j(\underline{w}) = \left[d_j - f_j(\underline{x}_k, \underline{w})\right]$$
$$\underline{e}(\underline{w}) = \left[e_1(\underline{w}), e_2(\underline{w}), ..., e_m(\underline{w})\right]^t$$

A matriz *jacobiana* $\underline{\underline{J}}(\underline{w})$ relacionada com o funcional $E_k(\underline{w})$ dado pela equação (A.37) é definida como segue:

$$\underline{\underline{J}}(\underline{w}) = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \cdots & \frac{\partial e_1}{\partial w_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_m}{\partial w_1} & \cdots & \frac{\partial e_m}{\partial w_M} \end{bmatrix} \tag{A.38}$$

Desta forma, o gradiente $\nabla E_k(\underline{w})$ e a matriz *hessiana* $\underline{\underline{H}}(\underline{w})$ relacionada com o funcional $E_k(\underline{w})$ são dados pelas equações:

$$\nabla E_k(\underline{w}) = \left[\underline{\underline{J}}(\underline{w})\right]^t \underline{e}(\underline{w}) \tag{A.39}$$

$$\underline{\underline{H}}\left(\underline{w}\right)=\left[\underline{\underline{J}}\left(\underline{w}\right)\right]^{t}\underline{\underline{J}}\left(\underline{w}\right)+\sum_{j=1}^{m}e_{j}\left(\underline{w}\right)\underline{\underline{H}}_{e_j}\left(\underline{w}\right) \tag{A.40}$$

$$\underline{\underline{H}}_{e_j}\left(\underline{w}\right)=\begin{bmatrix}\dfrac{\partial^{2}e_{j}\left(\underline{w}\right)}{\partial w_{1}^{\,2}} & \cdots & \dfrac{\partial^{2}e_{j}\left(\underline{w}\right)}{\partial w_{1}\partial w_{M}}\\ \vdots & \ddots & \vdots\\ \dfrac{\partial^{2}e_{j}\left(\underline{w}\right)}{\partial w_{M}\partial w_{1}} & \cdots & \dfrac{\partial^{2}e_{j}\left(\underline{w}\right)}{\partial w_{M}^{\,2}}\end{bmatrix}$$

Desprezando os termos de segunda ordem, ou seja, fazendo $\underline{\underline{H}}_{e_j}\left(\underline{w}\right)\approx\underline{\underline{0}}$, a regra de atualização dos pesos dada pela equação (A.18), princípio do método de *Newton*, passa a ser dada por:

$$\Delta\left(\underline{w}\right)=-\left\{\left[\underline{\underline{J}}\left(\underline{w}\right)\right]^{t}\underline{\underline{J}}\left(\underline{w}\right)\right\}^{-1}\left[\underline{\underline{J}}\left(\underline{w}\right)\right]^{t}\underline{e}\left(\underline{w}\right) \tag{A.41}$$

A utilização direta da equação (A.41) pode resultar em passos de atualização de magnitude elevada, conduzindo a soluções onde aproximação $\underline{\underline{S}}\left(\underline{w}\right)\approx\underline{\underline{0}}$ não é válida, comprometendo a eficiência do algoritmo [158]. Para garantir que o algoritmo realize a busca apenas na região onde esta aproximação é válida, o algoritmo de *Levenberg-Marquardt* utiliza a seguinte modificação da equação (A.41):

$$\Delta\left(\underline{w}\right)=-\left\{\left[\underline{\underline{J}}\left(\underline{w}\right)\right]^{t}\underline{\underline{J}}\left(\underline{w}\right)+\lambda\underline{\underline{I}}\right\}^{-1}\left[\underline{\underline{J}}\left(\underline{w}\right)\right]^{t}\underline{e}\left(\underline{w}\right) \tag{A.42}$$

Na equação (A.42), $\underline{\underline{I}}$ é a matriz identidade de dimensão $M\times M$ e λ é uma constante, relacionada com o tamanho da região onde a aproximação $\underline{\underline{S}}\left(\underline{w}\right)\approx\underline{\underline{0}}$ é válida. Desta forma, o algoritmo de *Levenberg-Marquardt* pode ser considerado como um algoritmo de otimização em regiões viáveis, visto que limita a busca apenas em regiões no entorno do ponto de operação, onde as aproximações consideradas pelo método são válidas [158] e [159]. Na prática, o valor da constante λ deve ser modificado ao longo do processo de otimização. Uma forma de atualização bastante utilizada consiste em fazer $\lambda=0.1$ no início do processo iterativo, e, se o erro diminuir para a iteração n,

diminuir λ em uma ordem de grandeza, ou seja, $\lambda(n+1)=0.1\lambda(n)$. Em caso contrário, aumentar em uma ordem de grandeza, ou seja, $\lambda(n+1)=10\lambda(n)$.

Para os MLP's utilizados nesta tese, contendo uma única camada escondida e uma única saída linear, o algoritmo de *Levenberg-Marquardt* para treinamento de MLP's pode ser resumido como segue:

1. Faça $l=0$;
2. Inicialize o vetor de parâmetros $\underline{w}(l)$;
3. Faça $\lambda(l)=0.1$;
4. Utilizando o algoritmo de retropropagação do erro seqüencial, calcule o vetor gradiente $\nabla E_k(\underline{w})\big|_{\underline{w}=\underline{w}(l)}$;
5. Calcule a matriz *jacobiana* $\underline{\underline{J}}(\underline{w})$ através da equação:

$$\underline{\underline{J}}(\underline{w})\big|_{\underline{w}=\underline{w}(l)}=\frac{1}{e(l)}\left[\nabla E_k(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\right]^t \qquad \text{(A.43)}$$
$$e(l)=d_k-f\left[\underline{x}_k,\underline{w}(l)\right]$$

6. Atualize o vetor de parâmetros $\underline{w}$ através da equação:

$$\underline{w}(l+1)=\underline{w}(l)-\left\{\left[\underline{\underline{J}}(\underline{w})\right]^t\underline{\underline{J}}(\underline{w})+\lambda(l)\underline{\underline{I}}\right\}^{-1}\left[\underline{\underline{J}}(\underline{w})\right]^t\underline{e}(\underline{w}) \qquad \text{(A.44)}$$

7. Atualize a constante λ através da equação:

$$\lambda(l+1)=\begin{cases}0.1\lambda(l), \text{ se } E_k(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}<E_k(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\\ 10\lambda(l), \text{ se } E_k(\underline{w})\big|_{\underline{w}=\underline{w}(l+1)}>E_k(\underline{w})\big|_{\underline{w}=\underline{w}(l)}\end{cases} \qquad \text{(A.45)}$$

8. Se o critério de parada for atendido, encerre o algoritmo. Do contrário, faça $l=l+1$ e retorne ao passo 4.

CONTROLE DE COMPLEXIDADE DE PREVISORES NEURAIS NA PREVISÃO DE CARGA ELÉTRICA PARA O HORIZONTE DE CURTO PRAZO

VITOR HUGO FERREIRA, ALEXANDRE P. ALVES DA SILVA

Laboratório de Sistemas de Potência, Programa de Engenharia Elétrica, COPPE/UFRJ, 21945-970, Rio de Janeiro, RJ, Brasil.
E-mails: vitor@vishnu.coep.ufrj.br, alex@coep.ufrj.br

Abstract— The knowledge of loads' future behavior is very important for decision making in power system operation. During the last years, many load models have been proposed, and the neural ones have presented the best results. One of the disadvantages of the neural models is the possibility of excessive adjustment of the training data, named *overfitting*, degrading the generalization performance of the estimated models. This problem can be tackled by using regularization techniques. The present shows the application of some of these techniques to short term load forecasting.

Keywords— Short-term load forecasting, artificial neural networks, regularization techniques, *Bayesian* training, gain scaling, support vector machines.

Resumo— O conhecimento do comportamento futuro da carga é de suma importância na tomada de decisões referentes à operação de sistemas de potência. Ao longo dos últimos anos, vários modelos vêm sendo propostos para a modelagem de carga, dentre os quais têm se destacado as redes neurais. Uma das desvantagens dos modelos neurais reside na possibilidade de aproximação excessiva dos dados de treinamento, o chamado *overfitting*, comprometendo a capacidade de generalização dos modelos estimados. Este problema pode ser abordado através do uso de técnicas de regularização. O presente trabalho apresenta a aplicação de algumas destas técnicas aos modelos neurais estimados para previsão de carga a curto prazo.

Palavras-chave— Previsão de carga a curto prazo, redes neurais artificiais, técnicas de regularização, treinamento *Bayesiano*, escalonamento do ganho da função de ativação, máquinas de vetor suporte.

1 Introdução

Muitas das decisões a serem tomadas na operação de sistemas de potência, tais como comissionamento de unidades geradoras, despacho econômico, controle automático da geração e elaboração de planos de manutenção, dependem do conhecimento prévio do comportamento da carga (Debs, 1988). Diante disso, vários modelos vêm sendo propostos ao longo dos anos para prever a carga, podendo ser citados os modelos de Box-Jenkins, o perceptron de múltiplas camadas (MLP), as redes de função de base radial (RBFN), modelos híbridos, dentre outros.

Entre os modelos baseados em redes neurais artificiais (RNA's), os MLP's e as RBFN's, têm se mostrado bastante eficientes na tarefa de modelar a carga para o horizonte de curto prazo, que varia de poucos minutos até 168 horas à frente. Uma das vantagens dos modelos neurais reside no teorema da aproximação universal (Haykin, 2001), que demonstra que estes modelos, com uma única camada oculta contendo um número suficiente de neurônios, podem aproximar com precisão arbitrária qualquer função contínua não-linear. Entretanto, na presença de dados ruidosos, esta vantajosa característica das RNA's pode se tornar prejudicial, visto que estas, ao invés de representar o comportamento regular da carga, podem modelar o ruído, o chamado *overfitting*, resultando na degradação do desempenho para novos dados que não aqueles utilizados no treinamento. Modelos com esta característica apresentam pequena capacidade de generalização, devido principalmente à complexidade excessiva, isto é, número elevado de parâmetros livres.

O exposto acima ilustra a necessidade de controle da complexidade das RNA's, com o intuito de obter a melhor capacidade de generalização possível. Neste trabalho são aplicadas algumas técnicas de regularização na estimação de RNA's para previsão de carga a curto prazo, mais especificamente o treinamento *Bayesiano*, o escalonamento do ganho da função de ativação e as máquinas de vetor suporte (SVM).

A base de dados utilizada neste trabalho corresponde às séries de carga e temperatura, discretizadas em base horária, de uma concessionária norte-americana (http://www.ee.washington.edu/class/559/2002spr) que vêm sendo utilizadas em várias propostas de previsores neurais. Para verificação da qualidade dos modelos são

realizadas previsões 24 horas à frente para o ano de 1991.

2 Redes Neurais Artificiais

As RNA's comumente utilizadas em previsão de carga a curto prazo são do tipo *feedforward* com uma única camada oculta de neurônios. As subseções a seguir descrevem os modelos neurais do tipo MLP e SVM.

2.1 Perceptron de Múltiplas Camadas (MLP)

Seja $\underline{x} \in \mathbb{R}^n$ o vetor contendo os sinais de entrada, $d \in \mathbb{R}$ a saída desejada, $\underline{w} \in \mathbb{R}^M$, onde $M = mn + 2m + 1$ (M igual ao número total de parâmetros do modelo e m o número de neurônios na camada oculta) o vetor contendo os pesos das conexões e os bias b_k e b, $k = 1, 2, ..., m$, $\varphi(x): \mathbb{R} \to \mathbb{R}$, uma função não-linear, e $y \in \mathbb{R}$, $y = f(\underline{x}, \underline{w})$, a saída gerada pela rede. Para o caso do MLP, a saída da rede é dada por:

$$c_k = \varphi\left[\sum_{j=1}^{n}\left(w_{kj}x_j\right) + b_k\right], k = 1, 2, ..., m \quad (1)$$

$$y = f(\underline{x}, \underline{w}) = \sum_{i=1}^{m} w_i c_i + b$$

Dado um conjunto D de N pares entrada-saída, $D = \{\underline{x}_i, d_i\}$, $i = 1, 2, ..., N$, o objetivo do treinamento de um MLP reside na estimativa do vetor de pesos $\underline{w}$ que minimize o risco empírico dado por:

$$E_s(\underline{w}, D) = \frac{1}{2}\sum_{j=1}^{N}\left[d_i - f(\underline{x}_i, \underline{w})\right]^2 = \frac{1}{2}\sum_{j=1}^{N}\left[d_i - y_i\right]^2 \quad (2)$$

Existem vários algoritmos para minimização do funcional descrito na equação (2), dentre os quais podem ser citados retro-propagação de erro e *Levenberg-Marquardt.* A principal desvantagem dos métodos de treinamento que minimizam o funcional descrito em (2) é a possibilidade de ocorrência de *overfitting.*

2.2 Máquinas de Vetor Suporte (SVM)

Para as máquinas de vetor suporte, a saída da rede é dada por:

$$y = \sum_{j=0}^{m} W_j \phi_j(\underline{x}) = \underline{W}^T \underline{\phi}(\underline{x}) \quad (3)$$

$$\underline{\phi}(\underline{x}) = \left[1, \phi_1(\underline{x}), \phi_2(\underline{x}), ..., \phi_m(\underline{x})\right]^T$$

$$\underline{W} = \left[b, W_1, W_2, ..., W_m\right]^T$$

onde $\underline{\phi}(\underline{x})$ representa um conjunto de funções de base não-lineares definidas pelo usuário. Seja a função de perda, com tolerância $\varepsilon, L_\varepsilon(d, y)$, dada por (Cherkassky, 1998):

$$L_\varepsilon(d, y) = \begin{cases} |d - y| - \varepsilon, & |d - y| \geq \varepsilon \\ 0, & |d - y| < \varepsilon \end{cases} \quad (4)$$

Na equação (4), ε é um parâmetro especificado pelo usuário. Para problemas de regressão em que a saída está corrompida por um ruído aditivo, este parâmetro pode representar a variância de tal ruído.

O objetivo do treinamento de uma SVM é a minimização do risco empírico dado por:

$$E_s(\underline{W}, D) = \frac{1}{N}\sum_{i=1}^{N} L_\varepsilon(d_i, y_i) \quad (5)$$

sujeito à restrição:

$$\left\|\underline{W}\right\|^2 \leq c_0 \quad (6)$$

onde c_0 responde pelo controle da complexidade do modelo (Cherkassky, 1998).

A restrição não-linear representada na equação (6) pode ser abordada na função objetivo de um problema de programação quadrática dado por:

$$\min \Phi(\underline{W}, \underline{\xi}, \underline{\xi}') = C\left[\sum_{i=1}^{N}(\xi_i + \xi_i')\right] + \frac{1}{2}\underline{W}^T\underline{W} \quad (7)$$

onde

$$\underline{\xi} = \left[\xi_1, \xi_2, ..., \xi_N\right]^T, \quad \underline{\xi}' = \left[\xi_1', \xi_2', ..., \xi_N'\right]^T$$

sujeito às restrições:

$$d_i - \underline{W}^T\underline{\phi}(\underline{x}_i) \leq \varepsilon + \xi_i \quad (8)$$

$$\underline{W}^T\underline{\phi}(\underline{x}_i) - d_i \leq \varepsilon + \xi_i'$$

$$\xi_i \geq 0, \quad \xi_i' \geq 0, \quad i = 1, 2, ..., N$$

Na equação (7), C é um parâmetro especificado pelo usuário, responsável pelo equilíbrio entre o ajuste dos dados de treinamento e a complexidade do modelo. Na prática, esta constante é determinada empiricamente, através de técnicas de reamostragem, como, por exemplo, validação cruzada.

Para solução deste problema de otimização, pode ser definida a função *Lagrangeana*,

$$J(\underline{W}, \underline{\xi}, \underline{\xi}', \underline{\alpha}, \underline{\alpha}', \underline{\gamma}, \underline{\gamma}') = C\sum_{i=1}^{N}(\xi_i + \xi_i') + \frac{1}{2}\underline{W}^T\underline{W} \quad (9)$$

$$-\sum_{i=1}^{N}\alpha_i\left[\underline{W}^T\underline{\phi}(\underline{x}_i) - d_i + \varepsilon + \xi_i\right]$$

$$-\sum_{i=1}^{N}\alpha_i'\left[d_i - \underline{W}^T\underline{\phi}(\underline{x}_i) + \varepsilon + \xi_i'\right]$$

$$-\sum_{i=1}^{N}(\gamma_i\xi_i + \gamma_i'\xi_i')$$

$$\underline{\alpha} = \left[\alpha_1, \alpha_2, ..., \alpha_N\right]^T, \quad \underline{\alpha}' = \left[\alpha_1', \alpha_2', ..., \alpha_N'\right]^T$$

$$\underline{\gamma} = \left[\gamma_1, \gamma_2, ..., \gamma_N\right]^T, \quad \underline{\gamma}' = \left[\gamma_1', \gamma_2', ..., \gamma_N'\right]^T$$

onde $\underline{\alpha}$ e $\underline{\alpha}'$ são os multiplicadores de Lagrange. Das condições de otimalidade do cálculo, são encontradas as seguintes equações:

$$\underline{W} = \sum_{i=1}^{N} (\alpha_i - \alpha_i')\underline{\phi}(\underline{x}_i) \tag{10}$$

$$\gamma_i = C - \alpha_i, \quad \gamma_i' = C - \alpha_i', \quad i = 1, 2, ..., N$$

De posse das equações (10), o problema dual de maximização, correspondente ao problema primal de minimização da função definida em (9), pode ser formulado como:

$$\max Q(\underline{\alpha}, \underline{\alpha}') = \sum_{i=1}^{N} d_i (\alpha_i - \alpha_i') - \varepsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i') \tag{11}$$

$$-\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i - \alpha_i')(\alpha_j - \alpha_j') K(\underline{x}_i, \underline{x}_j)$$

sujeito às restrições:

$$\sum_{i=1}^{N} (\alpha_i - \alpha_i') = 0 \tag{12}$$

$$0 \le \alpha_i \le C, \quad 0 \le \alpha_i' \le C, \quad i = 1, 2, ..., N$$

Na equação (11), $K(\underline{x}_i, \underline{x}_j) = \underline{\phi}^T(\underline{x}_i)\underline{\phi}(\underline{x}_j)$ é o núcleo do produto interno (*kernel*) definido conforme o teorema de Mercer (Cherkassky, 1998). Portanto, a saída da máquina de vetor suporte é dada por:

$$y = f(\underline{x}, \underline{W}) = \sum_{i=1}^{N} (\alpha_i - \alpha_i') K(\underline{x}, \underline{x}_i) \tag{13}$$

Como pode ser visto através da equação (13), os vetores de suporte são definidos como os padrões do conjunto de treinamento que apresentem $\alpha_i \neq \alpha_i'$, ou seja, aqueles situados fora da banda definida por ε. Isto é equivalente a dizer que uma SVM é um modelo *feedforward* com função de ativação definida pelo *kernel* $K(\underline{x}, \underline{x}_i)$.

2.3 MLP versus Máquina de Vetor Suporte

Do exposto acima, algumas diferenças importantes podem ser observadas entre o MLP e a SVM. No treinamento do MLP via retro-propagação de erro, uma estrutura para o mesmo (número de neurônios na camada intermediária) deve ser definida a priori. Nas SVM, a estrutura do modelo é resultado do algoritmo de treinamento, dependente dos parâmetros ε, C e do tipo de *kernel* selecionado. O treinamento do MLP via retro-propagação de erro é baseado na minimização do risco empírico, onde se deseja única e exclusivamente a minimização do erro para o conjunto de treinamento. Já o das SVM é baseado na minimização do risco estrutural, princípio que busca a minimização do limite superior do erro de generalização. Em outras palavras, o problema abordado pela teoria da regularização, que será assunto da próxima seção, é abordado diretamente no desenvolvimento das SVM. Outra consideração importante diz respeito à superfície a ser otimizada no treinamento de um MLP e de uma SVM. Enquanto que para o MLP esta superfície vem a ser extremamente não-convexa, repleta de mínimos locais, para a SVM consiste em uma superfície quadrática com um único ponto de máximo. Consequentemente, para um mesmo conjunto de dados e valores de ε e C constantes, a solução para as SVM é única.

3 Técnicas de Regularização

Conforme mencionado na seção 2, o treinamento de um MLP na presença de dados ruidosos, através da minimização do risco empírico dado pela equação (2), pode levar à ocorrência de *overfitting*. Portanto, é necessário o controle da complexidade do MLP.

Existem dois procedimentos mais indicados para o controle da complexidade do MLP. O primeiro é a chamada *estabilização de estrutura* (Bishop, 1995), onde o objetivo reside na determinação do número suficiente de neurônios na camada oculta, que pode ser implementado de três formas.

Uma delas consiste em comparar modelos com números de neurônios diferentes na camada intermediária, e se decidir pela estrutura que apresentar melhor desempenho em um conjunto independente de dados. Isso pode ser feito através de alguma técnica de reamostragem ou através de métodos analíticos de qualificação de modelos, tais como o *NIC (Network Information Criterion)* (Murata, 1994), a comparação *Bayesiana* de modelos (Mackay, 1992), etc.

De outra forma, o processo de modelagem pode ser iniciado através de um modelo demasiadamente complexo, o qual é submetido a alguns algoritmos de poda. Por último, tal processo pode ser iniciado com uma rede excessivamente inflexível, por exemplo, com nenhum neurônio na camada intermediária, e ao longo do treinamento serem adicionados neurônios nesta camada com o intuito de se obter a estrutura com melhor capacidade de generalização. Aplicações destas variantes podem ser encontradas em (Doveh, 1999) e (Treadgold, 1999).

Outro procedimento utilizado para controlar a complexidade das RNA's tem por base a teoria da regularização. Neste procedimento, o compromisso entre o ajuste dos

dados de treinamento e a capacidade de generalização pode ser obtido através da minimização do risco total, expresso por:

$$R(\underline{w}) = E_s(\underline{w}, D) + \lambda E_c(\underline{w}) \quad (14)$$

Na equação acima, o primeiro termo representa o ajuste do modelo aos dados de treinamento (risco empírico), que, para o MLP, é dado pela equação (2). O segundo termo a complexidade do modelo, impondo à solução conhecimento prévio acerca da mesma, podendo adquirir diversas formas, conforme apresentado em (Haykin, 2001), (Mackay, 1992), (Doveh, 1999), (Treadgold, 1999). O parâmetro λ é conhecido como parâmetro de regularização, o qual define a importância relativa entre o ajuste dos dados de treinamento e a complexidade do modelo. O valor a ser utilizado para este parâmetro está diretamente relacionado com o dilema *bias-variância*, podendo ser estimado através de técnicas de reamostragem ou por meio de procedimentos analíticos, conforme apresentado em (Mackay, 1992).

Uma das formas para o funcional $E_c(\underline{w})$ é obtida através da aplicação de técnicas de inferência *Bayesiana* na determinação do vetor de parâmetros $\underline{w}$ que define o MLP (Mackay, 1992). Pela regra de *Bayes*, a função densidade de probabilidade do vetor $\underline{w}$, $p(\underline{w}|D)$, dado o conjunto de pares entrada-saída *D*, é dada por:

$$p(\underline{w}|D) = \frac{p(D|\underline{w})\,p(\underline{w})}{p(D)} \quad (15)$$

Na equação (15), $p(D|\underline{w})$ representa a função densidade de probabilidade de ocorrência do conjunto *D*, dado o vetor de parâmetros $\underline{w}$, $p(\underline{w})$ representa a função densidade de probabilidade a priori de $\underline{w}$, e $p(D) = \int p(D|\underline{w})\,p(\underline{w})\,d\underline{w}$ representa um fator de normalização, que garante $\int p(\underline{w}|D)\,d\underline{w} = 1$.

Assume-se que o vetor $\underline{w}$ apresenta *a priori* uma distribuição Gaussiana com vetor média nulo e matriz de covariância $\alpha^{-1}\underline{\underline{I}}$, onde $\underline{\underline{I}}$ é a matriz identidade de dimensão MxM, e que as saídas desejadas d_i são dadas por $d_i = f(\underline{x}_i, \underline{w}) + \zeta$, onde ζ é um ruído gaussiano de média nula e variância β^{-1}. Com as hipóteses anteriores, maximizar a distribuição de probabilidade *a posteriori* de $\underline{w}$ é equivalente a minimizar a seguinte expressão (Mackay, 1992):

$$S(\underline{w}) = \frac{\beta}{2}\sum_{i=1}^{N}\left[f(x_i, \underline{w}) - d_i\right]^2 + \frac{\alpha}{2}\sum_{i=1}^{M} w_i^2 \quad (16)$$

Como o único ponto de interesse em $S(\underline{w})$ é o ponto de mínimo, fazendo $\lambda = \alpha/\beta$, $S(\underline{w}) = R(\underline{w})$, e, portanto,

$$E_c(\underline{w}) = \frac{1}{2}\|\underline{w}\|^2 \quad (17)$$

Este termo regularizador é conhecido como *weight decay*, que favorece modelos apresentando valores absolutos menores para as componentes do vetor de parâmetros $\underline{w}$.

A vantagem da abordagem *Bayesiana* está baseada no fato que esta fornece um mecanismo iterativo para estimativa ao longo do treinamento tanto de α quanto de β e, portanto, de λ (Mackay, 1992). Este tipo de treinamento do MLP é conhecido como treinamento *Bayesiano*.

Redes com boa capacidade de generalização podem ser obtidas através de outros métodos, onde a complexidade do modelo não é controlada diretamente, ou seja, $\lambda = 0$. Dentre estes métodos, podem ser citados a parada antecipada do treinamento (*early stopping*), o treinamento com inserção de ruído e o escalonamento do ganho da função de ativação (Reed, 1999).

Na parada antecipada do treinamento, validação cruzada é utilizada para particionar o conjunto de treinamento em dois subconjuntos, um de estimação do vetor de pesos $\underline{w}$ e outro de validação do modelo obtido. A atualização do vetor de pesos é interrompida quando o erro para o conjunto de validação sofrer deterioração em relação às iterações anteriores. Este método é questionável, visto que é depositada uma confiança excessiva no conjunto de validação, além do fato de que este método não leva o vetor de parâmetros para um ponto de mínimo na superfície de erro, invalidando a utilização de muitos resultados obtidos para modelos não-lineares baseados nessa premissa (Doveh, 1999). Apesar destas questões, abundam na literatura estudos apresentando resultados satisfatórios para o método de *early stopping*, como pode ser visto em (Doveh, 1999), (Treadgold, 1999). Entretanto, não pode ser esquecida a perda de informação relacionada à correlação serial, visto que parte dos dados é reservada para validação.

O treinamento com inserção de ruído consiste em adicionar ao conjunto de treinamento versões corrompidas dos padrões originalmente pertencentes a este. Esses padrões são gerados através da adição de ruído nas variáveis de entrada, garantindo assim que, para padrões de entrada similares, a saída sofrerá

pouca ou nenhuma alteração, o que é equivalente a supor que a função a ser aproximada apresenta um certo grau de suavidade, conforme assumido na teoria de regularização de *Tikhonov* (Reed, 1999).

Segundo (Reed, 1999), para RNA's com uma única camada oculta contendo neurônios não-lineares e uma única saída linear, se as amostras do conjunto de treinamento forem obtidas seguindo uma distribuição uniforme e o ruído adicionado às entradas for Gaussiano com vetor média nulo e matriz de covariância $\sigma^2_{ruido}\underline{\underline{I}}$ (neste caso $\underline{\underline{I}}$ tem dimensão *nxn*), a rede obtida através da minimização do risco empírico irá apresentar capacidade de generalização similar à rede treinada com o conjunto original de dados (não corrompido) utilizando uma outra técnica de regularização. Da mesma forma, uma rede treinada com o conjunto original de dados, porém sem utilizar um regularizador, ou seja, $\lambda = 0$, irá apresentar capacidade de generalização similar às regularizadas se os ganhos das funções de ativação dos neurônios da camada oculta forem multiplicados pelo fator a_k dado por:

$$a_k = \frac{1}{\sqrt{\left\|\underline{w}_k\right\|^2 \sigma^2_{ruido} + 1}} \qquad (18)$$

$$\underline{w}_k = \left[w_{k1}, w_{k2}, ..., w_{kn}\right]^T,\ k = 1, 2, ..., m$$

Este procedimento é similar ao da rede treinada utilizando o procedimento de regularização *weight-decay* (Reed, 1999). Este resultado sugere um procedimento pós-treinamento de ajuste dos ganhos das funções de ativação de modelos neurais *feedforward* com uma única camada oculta, com o intuito de melhorar a capacidade de generalização do modelo obtido.

4 Pré-Processamento dos Dados

Como é de conhecimento amplo, o pré-processamento dos dados constitui uma das principais ferramentas de melhoria de desempenho de qualquer modelo estimado por técnicas de identificação de sistemas. Logo, algumas transformações foram efetuadas à base original de dados, devido principalmente à ocorrência de dados faltantes (*missing data)*, às sazonalidades existentes nas séries de carga e temperatura e na possível tendência existente na série de carga.

Além disso, vários estudos (Debs, 1988), (Doveh, 1999), (Alves da Silva, 2000), (Reis, 2002) mostram que a carga elétrica guarda correlação significativa com as condições climáticas da região de atendimento. Estas condições podem ser representadas pela umidade relativa do ar, temperatura, luminosidade, velocidade do vento, dentre outras. Neste trabalho, a série de temperatura da região de atendimento da concessionária em questão será utilizada como informação climática.

O primeiro tratamento das séries foi feito com relação aos dados faltantes, visto que foi verificada a presença de valores nulos isolados tanto de carga quanto de temperatura. Para corrigir este problema, os valores nulos foram substituídos pelas respectivas médias aritméticas entre o valor anterior ao dado faltante e o valor posterior a este.

Em seguida foram retiradas da série de carga as sazonalidades diária e semanal, além da aplicação da primeira diferença, com o intuito de remover a tendência da série em questão. A seqüência de transformações foi a seguinte:

$$\begin{aligned} Serie1(k) &= Serie(k) - Serie(k-1) \\ Serie2(k) &= Serie1(k) - Serie1(k-24) \\ Serie3(k) &= Serie2(k) - Serie2(k-168) \end{aligned} \qquad (19)$$

A série Serie3 corresponde ao período que vai de 30 de outubro de 1990 a 31 de dezembro de 1991. Essa série foi reduzida (média nula e variância unitária) e posteriormente normalizada no intervalo [-1,1], gerando a série *S(k).* Após a normalização, foi calculada a autocorrelação parcial desta série, com o intuito de determinar quais entradas serão utilizadas pelo modelo *NARX (Nonlinear Autoregressive Exogenous)* selecionado. A partir da função de autocorrelação parcial para a série *S(k)*, pode ser verificado que os atrasos que apresentam correlação mais significativa são *S(k-1), S(k-2), S(k-24)* e *S(k-168).*

Após a retirada da sazonalidade diária da série de temperatura e posterior normalização da mesma, foi verificado, através da correlação cruzada entre esta série e *S(k)*, que os atrasos mais significativos são *T(k), T(k-1), T(k-2), T(k-24)* e *T(k-168).*

Vale mencionar que, por comodidade, para realização das previsões de carga para as 24 horas de um determinado dia, as entradas *T(k), T(k-1)* e *T(k-2)* são alimentadas sempre com os respectivos valores medidos da série de temperatura (desazonalizada e normalizada), não sendo realizadas previsões de temperatura.

Estes atrasos, tanto de carga quanto de temperatura, são usados como entradas das redes, e além destas variáveis, são utilizadas também duas entradas contendo informações sobre a hora do dia, codificadas da seguinte forma (Alves da Silva, 2000), (Reis, 2002):

$$HS(k) = sen\left(\frac{2\pi k}{24}\right);\ HC(k) = \cos\left(\frac{2\pi k}{24}\right) \qquad (20)$$

$$k = 1, 2, ..., 24$$

5 Definição dos Modelos Utilizados

Visto que o comportamento da carga varia conforme o dia da semana (Debs, 1988), (Doveh, 1999), (Alves da Silva, 2000), (Reis, 2002), foram desenvolvidos modelos para cada dia da semana, incluindo finais de semana, totalizando sete modelos para cada método de treinamento. O conjunto de treinamento de cada modelo consiste nos padrões referentes às últimas seis semanas do respectivo dia da semana. Por exemplo, para o modelo referente às terças-feiras, o conjunto de treinamento consistirá dos 144 padrões referentes às seis terças-feiras anteriores àquela a ser prevista. Definido o conjunto de treinamento, os modelos são estimados e são realizadas previsões para as 24 horas do dia seguinte, a partir das 0:00 horas. Vale mencionar que não são feitas previsões para os feriados e para os dias posteriores a estes, pois um tratamento especial seria necessário nestes casos, o que vai além do escopo deste trabalho. A figura 1 apresenta um diagrama esquemático exemplificando o procedimento de construção do conjunto de treinamento e posterior previsão.

Para o caso do MLP, foi utilizada como função de ativação dos neurônios da camada oculta a função $\varphi(x) = \tanh(x)$. Para a máquina de vetor suporte, o *kernel* $K(\underline{x}, \underline{x}_i)$ utilizado foi do tipo gaussiano, cujo parâmetro a ser especificado pelo usuário é o desvio padrão σ_{kernel}.

Os métodos de treinamento utilizados foram: treinamento por retro-propagação convencional, implementado no software *Matlab* através da função *traingdx*; treinamento por retro-propagação convencional e posterior escalonamento do ganho da função de ativação; treinamento *Bayesiano*, implementado no software *Matlab*, através da função *trainbr*; e máquina de vetor suporte, utilizando o *toolbox SVM* do software *Matlab*, obtido em http:// www.isis.ecs.soton.ac.uk/resources/svminfo.

A título de comparação, as mesmas entradas de carga e temperatura dos modelos neurais foram utilizadas como entradas de um modelo ARX estimado via mínimos quadrados.

CONJUNTO DE TREINAMENTO						PREVISÃO
17/5/1990	24/5/1990	31/5/1990	7/6/1990	14/6/1990	21/6/1990	28/6/1990
24 HORAS	24 HORAS	24 HORAS	24 HORAS	24 HORAS	24 HORAS	24 HORAS

CONJUNTO DE TREINAMENTO						PREVISÃO
24/5/1990	31/5/1990	7/6/1990	14/6/1990	21/6/1990	28/6/1990	5/7/1990
24 HORAS	24 HORAS	24 HORAS	24 HORAS	24 HORAS	24 HORAS	24 HORAS

Figura 1: Arranjo do conjunto de treinamento e da previsão para um dia específico da semana

6 Resultados

Os resultados obtidos para os modelos neurais para cada dia da semana utilizando os diversos métodos de treinamento são apresentados nas tabelas 1 e 2. Na Tabela 1, estão apresentados as estruturas e os parâmetros dos modelos que apresentaram os melhores resultados. As três primeiras colunas desta tabela mostram o número de neurônios na camada oculta do MLP que apresentou o melhor resultado para o respectivo dia da semana, utilizando o correspondente método de treinamento. A última coluna desta tabela apresenta o número médio de vetores suporte (NMVS) obtido para cada dia da semana.

A Tabela 2 apresenta um quadro comparativo entre os erros percentuais médios obtidos para cada dia da semana, utilizando cada um dos métodos de treinamento citados e também o modelo ARX, sendo destacados os menores erros percentuais obtidos. A última coluna apresenta a diferença percentual entre o melhor e o pior resultado obtido para o respectivo dia.

Tabela 1: Estrutura e parâmetros dos modelos testados

	Sem Regularizador	*Bayesiano*	Escalonamento	SVM				
	Neurônios	Neurônios	Neurônios	$\sigma^2_{ruído}$	C	ε	σ_{kernel}	NMVS
Segunda	2	2	2	0,12	0,1	0,100	4,24	57,3
Terça	2	2	2	0,16	0,1	0,001	4,24	140,6
Quarta	3	2	3	0,07	1,0	0,100	2,72	58,7
Quinta	3	2	2	0,17	1,0	0,400	1,96	12,8
Sexta	2	2	2	0,12	1,0	0,400	3,48	11,7
Sábado	4	2	4	0,05	0,1	0,001	5	143,4
Domingo	2	2	2	0,11	1,0	0,100	1,96	59,1

Tabela 2: Comparação entre os resultados obtidos

	ARX	Sem Regularizador	*Bayesiano*	Escalonamento	SVM	Ganho de Desempenho
Segunda	8,23	8,76	6,43	7,00	5,23	40,3
Terça	8,16	7,04	7,47	6,52	4,97	39,1
Quarta	8,15	6,94	7,08	6,18	5,00	38,7
Quinta	8,15	10,21	6,93	8,41	7,83	32,1
Sexta	9,54	7,33	6,29	7,18	6,39	34,1
Sábado	7,38	9,57	7,38	8,17	5,24	45,3
Domingo	7,02	8,19	6,91	7,42	5,00	38,9

7 Conclusão

Este trabalho teve por objetivo a aplicação de algumas técnicas de controle de complexidade no desenvolvimento de modelos neurais para previsão de carga elétrica para o horizonte de curto prazo. Conforme evidenciado na Tabela 2, as técnicas aqui utilizadas apresentam potencial significativo, visto que as mesmas geraram ganho expressivo de desempenho para todos os dias da semana.

Vale ressaltar o desempenho destacado da máquina de vetor suporte, que apresentou os melhores resultados para todos os dias da semana. Apesar deste resultado preliminar superior, é esperado que estes possam ser melhorados através da busca, no espaço tridimensional, de valores ótimos para os parâmetros *C* e ε, e para o desvio padrão do *kernel*, σ_{kernel}. A dificuldade relacionada ao

ajuste destes parâmetros de treinamento é devida ao fato de que o presente trabalho representa a primeira experiência na utilização de SVM para previsão de carga elétrica.

Apesar do escalonamento do ganho da função de ativação e do treinamento *Bayesiano* não terem apresentado resultados comparáveis aos da maquina de vetor suporte, estas técnicas de regularização apresentaram melhores resultados que o MLP sem regularizador, comprovando a eficiência das mesmas. O escalonamento do ganho da função de ativação é interessante sob o ponto de vista de esforço computacional, já que se trata de um procedimento pós-treinamento muito simples que pode ser aplicado a qualquer MLP com uma única camada oculta e uma única saída linear, independente do algoritmo de treinamento utilizado. Já o treinamento *Bayesiano* apresenta como principal vantagem o cálculo iterativo do parâmetro de regularização λ, requerendo menor intervenção por parte do usuário do que os outros dois métodos testados.

Outro resultado interessante foi obtido para o modelo linear, em comparação ao MLP sem regularizador. O melhor desempenho do modelo ARX evidencia a necessidade do controle de complexidade de modelos não-lineares, sinalizando também para a seleção de variáveis explicativas através de outros procedimentos mais apropriados para modelos neurais.

8 Agradecimentos

Os autores agradecem o apoio da CAPES e do CNPq pelo suporte financeiro.

Referências Bibliográficas

Debs, A.S. (1988), Modern Power Systems Control and Operation, Kluwer Academic Publishers.

Haykin, S. (2001), Redes Neurais, Princípios e Prática, 2ª. Edição, Editora Bookman, Porto Alegre, R.S., Brasil.

Cherkassky, V., Mulier, F. (1998), Learning from Data - Concepts, Theory and Methods, John Wiley & Sons, New York, USA.

Bishop, C.M. (1995), Neural Networks for Pattern Recognition, Oxford University Press, New York.

Murata, N., Yoshizawa, S., Amari, S.I. (1994), Network Information Criterion – Determining the Number of Hidden Units for an Artificial Neural Network Model, *IEEE Transactions on Neural Networks*, v.5, n.6, 865-872.

Mackay, D.J.C. (1992), Bayesian Methods for Adaptative Models, Ph.D. dissertation, California Institute of Technology, Pasadena, California, USA.

Doveh, E., Feigin, P., Greig, D., Hyams, L. (1999), Experience with FNN Models for Medium Term Power Demand Predictions, *IEEE Transactions on Power Systems*, v.14, n.2, 538-546.

Treadgold, N.K., Gedeon, T.D. (1999), Exploring Constructive Cascade Networks, *IEEE Trans. on Neural Networks*, v.10, n.6, 1335-1350.

Reed,R., Marks II, R.J., Oh, S. (1995), Similarities of Error Regularization, Sigmoid Gain Scaling, Target Smoothing and Training with Jitter, *IEEE Trans. on Neural Networks*, v.6, n.3, 529-538.

Alves da Silva, A.P, Moulin, L.S. (2000), Confidence Intervals for Neural Network Based Short-Term Load Forecasting, *IEEE Transactions on Power Systems*, v.15, n.4, 1191-1196.

Reis, A.J., Alves da Silva, A.P., (2002), Aplicação da Transformada Wavelet Discreta na Previsão de Carga a Curto Prazo Via Redes Neurais, *XIV Congresso Brasileiro de Automática*, Natal, Rio Grande do Norte, Brasil, 568-573.