UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MARCUS DE BARROS BRAGA

METODOLOGIAS DE INTELIGÊNCIA COMPUTACIONAL APLICADAS AO PROBLEMA DE PREVISÃO DE CARGA A CURTO PRAZO

DM 26/2010

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2010

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MARCUS DE BARROS BRAGA

METODOLOGIAS DE INTELIGÊNCIA COMPUTACIONAL APLICADAS AO PROBLEMA DE PREVISÃO DE CARGA A CURTO PRAZO

Dissertação submetida à Banca Examinadora do Programa de Pós Graduação em Engenharia Elétrica da UFPA, para a obtenção do Grau de Mestre e Engenharia Elétrica.

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2010

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

METODOLOGIAS DE INTELIGÊNCIA COMPUTACIONAL APLICADAS AO PROBLEMA DE PREVISÃO DE CARGA A CURTO PRAZO

AUTOR: MARCUS DE BARROS BRAGA

DISSERTAÇÃO DE MESTRADO SUBMETIDA À AVALIAÇÃO DA BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DO PARÁ E JULGADA ADEQUADA PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM: 27/12/2010

BANCA EXAMINADORA:

_____
Prof.PhD. Ádamo Lima de Santana
ORIENTADOR – PPGEE – UFPA

_____
Prof.PhD. Rogério Gaspar de Almeida
MEMBRO – UNIFAP

_____
Prof.PhD. Roberto Célio Limão de Oliveira
MEMBRO – ENGCOMP - UFPA

Dedico esta Dissertação aos meus pais e aos meus filhos.

AGRADECIMENTOS

Em primeiro lugar, minha gratidão é dedicada a Deus, criador de todas as coisas, tanto as tangíveis, quanto as intangíveis, as mensuráveis e as sem medida, as visíveis e as que não se percebem. A Ele, o Criador de toda inteligência humana, simulada posteriormente pelo homem, minha reverência, gratidão e louvor. Estendo este item para agradecer a Jesus Cristo, meu Salvador e Senhor pessoal.

Aos meus pais Edilson e Lídia e meus filhos, Rebeca e Caio, que me deram as motivações particulares para chegar até aqui. Foi por causa deles que não desisti no meio do caminho em meio a tantas dificuldades. Por eles, cheguei aqui.

Um agradecimento especial aos meus dois orientadores, Professor Ádamo Lima de Santana e, em especial, à Professora Adriana Garcez Castro, o apoio, compreensão e dedicação em todo o longo percurso até aqui.

Meus agradecimentos se estendem também a todos os professores do PPGEE e PPGCC, com quem tive a oportunidade de aprender e mergulhar neste fascinante mundo da computação inteligente.

Ao Professor Jurandir Garcez, agradeço o apoio na obtenção dos dados junto à Rede Celpa e a grande ajuda no projeto da ida ao Congresso Brasileiro de Automática em 2008, para apresentação do nosso artigo.

Aos amigos de mestrado, Lorena, Wilson, Gabrielle, Felipe, Agenor, Alex e Salomão, dentre outros, agradeço o apoio na construção desta vitória.

Um agradecimento especial eu dedico a meu irmão caçula, Judson, hoje Professor PhD da Universidade de Brasília, que me inspirou e orientou com clareza (e frieza) de raciocínio e, acima de tudo, agradeço sua amizade.

Aos meus novos amigos da Faculdade Ipiranga, agradeço o incentivo.

Enfim, agradeço a todos que, direta ou indiretamente contribuíram para a realização deste estudo.

“We all need some light”.

Neal Morse

RESUMO

Diversas atividades de planejamento e operação em sistemas de energia elétrica dependem do conhecimento antecipado e preciso da demanda de carga elétrica. Por este motivo, concessionárias de geração e distribuição de energia elétrica cada vez mais fazem uso de tecnologias de previsão de carga. Essas previsões podem ter um horizonte de curtíssimo, curto, médio ou longo prazo. Inúmeros métodos estatísticos vêm sendo utilizados para o problema de previsão. Todos estes métodos trabalham bem em condições normais, entretanto deixam a desejar em situações onde ocorrem mudanças inesperadas nos parâmetros do ambiente. Atualmente, técnicas baseadas em Inteligência Computacional vêm sendo apresentadas na literatura com resultados satisfatórios para o problema de previsão de carga. Considerando então a importância da previsão da carga elétrica para os sistemas de energia elétrica, neste trabalho, uma nova abordagem para o problema de previsão de carga via redes neurais Auto-Associativas e algoritmos genéticos é avaliada. Três modelos de previsão baseados em Inteligência Computacional são também apresentados tendo seus desempenhos avaliados e comparados com o sistema proposto. Com os resultados alcançados, pôde-se verificar que o modelo proposto se mostrou satisfatório para o problema de previsão, reforçando assim a aplicabilidade de metodologias de inteligência computacional para o problema de previsão de cargas.

ABSTRACT

Several activities of planning and operation in power systems rely on knowledge of early and accurate demand of electric load. For this reason, power generation and distribution companies are increasingly using technologies for load forecasting. These estimative may have a very short, short, medium or long-term horizon. Numerous statistical methods have been used for the problem of prediction. All these methods work well under normal conditions, but fail in situations where unexpected changes in the parameters of the environment occur. Currently, techniques based on Computational Intelligence have been presented in the literature with satisfactory results for the problem of load forecasting.Considering then the importance of load forecasting for the electric power systems, in this thesis a new approach to the load forecasting problem is evaluated by Auto-Associative Neural Networks and Genetic Algorithms. Three models based on Computational Intelligence are also presented with their performance evaluated and compared with the proposed system. With the obtained results, it was found that the proposed model is satisfactory for the problem of forecasting, thereby strengthening the applicability of computational intelligence methodologies to the problem of load prediction.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

CAPÍTULO 1

INTRODUÇÃO

1.1 CONSIDERAÇÕES GERAIS

A previsão de carga é uma das principais funções na operação de um sistema de energia elétrica. A busca para se prever de forma precisa a carga que será utilizada no futuro deve-se a um motivo simples: a eletricidade não pode ser armazenada por muito tempo, precisa ser distribuída.

A qualidade do controle do sistema de energia e a economia na sua operação são altamente sensíveis a erros de previsão. Estimar a demanda de energia futura de forma correta é prerrogativa obrigatória no gerenciamento da produção e distribuição dessa energia [1] e [2]. Os métodos de previsão de carga podem ser divididos em modelos de previsão a curtíssimo, curto, médio e longo prazo. Na previsão a curtíssimo prazo, o sistema estimará a carga de alguns minutos à frente, o que é muito útil em sistemas de tempo real. A previsão a curto prazo trabalha com um tempo de predição variando de poucas horas até uma semana a frente e é usada geralmente para alimentar ferramentas analíticas de sinalização de tarefas, planejamento do controle de tensão, segurança e economia. Na previsão a médio e longo prazo, o período pode variar de meses, anos até décadas, e a previsão ajuda a determinar a capacidade de geração, transmissão ou distribuição da carga de um sistema [3] e [4].

As metodologias para previsão de carga evoluíram muito nos últimos anos. Métodos estatísticos como Auto-Regressão e Séries Temporais vêm sendo extensivamente usados para previsão de carga. Outros modelos muito usados são Box-Jenkins, ARMA, ARIMA, Filtros de Kalman, modelos baseados na Expansão Espectral, dentre outros. Todas estas técnicas trabalham bem em condições normais, entretanto deixam a desejar em situações onde ocorrem mudanças inesperadas nos parâmetros do ambiente. Relacionamentos extremamente complicados entre variáveis que levam a operações matemáticas complexas na hora da previsão é um outro fator negativo destes modelos [5].

A partir da década de oitenta, o uso de técnicas de Inteligência Computacional incrementou muito a solução de problemas clássicos de engenharia

como a previsão de carga. Atualmente as redes neurais artificiais e os sistemas difusos vêm sendo bastante utilizados para o problema de previsão, principalmente devido a capacidade de ambos em modelar problemas não-lineares. Trabalhos também vêm sendo propostos usando algoritmos genéticos e sistema híbridos, onde várias destas técnicas inteligentes vêm sendo combinadas para o desenvolvimento de sistemas que apresentem o menor erro possível de previsão de carga .

Apesar de todos os métodos inteligentes utilizados até o presente momento apresentarem resultados satisfatórios para o problema de previsão de carga, novos sistemas continuam a ser desenvolvidos com o objetivo de se alcançar melhorias na exatidão da previsão.

1.2 OBJETIVOS E CONTRIBUIÇÕES

Este trabalho pretende:

1. Apresentar uma visão geral do problema de previsão de cargas elétricas.
2. Avaliar o uso de técnicas de inteligência computacional aplicadas ao problema de previsão de cargas.
3. Apresentar e avaliar o uso de uma metodologia híbrida baseada em Redes Neurais Auto-Associativas e Algoritmos Genéticos como ferramenta para o problema de previsão de carga. Esta é a principal contribuição deste trabalho.
4. Comparar o desempenho de algumas técnicas de Inteligência Computacional com o desempenho da metodologia proposta para previsão de carga.

1.3 ESTRUTURA DA TESE

Em adição a este capítulo introdutório, esta tese é composta de 5 capítulos:

- O Capítulo 2 apresenta uma visão geral de algumas das principais técnicas de Inteligência Computacional, como Redes Neurais, Sistemas Difusos, Sistemas Neuro-Difusos e Algoritmos Genéticos.
- O Capítulo 3 apresenta um visão geral sobre Previsão de Cargas, dando destaque para a Previsão a curto prazo, destacando os principais modelos encontrados na literatura. Trabalhos já desenvolvidos utilizando

Inteligência Computacional aplicada ao problema de previsão de carga também serão descritos.

- O Capítulo 4 apresenta a aplicação de técnicas de Inteligência Computacional ao problema da previsão de carga a curto prazo. Neste capítulo é também destacado o modelo proposto por este trabalho, a combinação de rede neural Auto-Associativa com algoritmo genético para o problema de previsão.
- O Capítulo 5 apresenta as conclusões.

CAPÍTULO 2

INTELIGÊNCIA COMPUTACIONAL

A Inteligência Computacional busca, através de técnicas inspiradas na natureza, o desenvolvimento de sistemas inteligentes que imitem aspectos do comportamento humano, tais como: aprendizado, percepção, raciocínio, evolução e adaptação. Inicialmente a abordagem simbólica dominava o cenário científico, entretanto, atualmente, o que se pode perceber é que técnicas conexionistas mais interessantes, como Redes Neurais Artificiais e Lógica Difusa (Nebulosa) vêm ganhando um maior espaço. Outras técnicas tais como Algoritmos Genéticos e Sistemas híbridos também vêm se destacando com diversos trabalhos apresentados na literatura.

As Redes Neurais Artificiais são modelos computacionais não lineares, inspirados na estrutura e operação do cérebro humano, que procuram reproduzir características humanas, tais como aprendizado, associação, generalização e abstração. Elas podem ser utilizadas para resolver problemas de aproximação de funções, classificação e previsão de séries temporais.

Os Sistemas Difusos, baseados na lógica difusa, podem ser enquadrados em uma área de pesquisa sobre o tratamento da imprecisão, ou uma família de modelos matemáticos dedicados ao tratamento da imprecisão. A Lógica difusa converte informações imprecisas, incompletas ou incertas da linguagem em formato numérico facilmente processado por computadores. Assim como as redes neurais podem ser utilizados para problemas de aproximação de funções, classificação e previsão de séries temporais.

Os Algoritmos Genéticos são uma ferramenta poderosa na solução de problemas de otimização combinatória de larga escala. São baseados no conceito de que a evolução natural é capaz de fornecer soluções eficientes e complexas utilizando-se de mecanismos relativamente simples como a sobrevivência do indivíduo mais adequado na transmissão de características vantajosas ou desejáveis aos seus descendentes e em mutações.

Os Sistemas híbridos são métodos que procuram combinar diversas técnicas inteligentes a fim de compensar os problemas e dificuldades de cada uma com os pontos fortes de outra.

Neste capítulo veremos os principais conceitos de alguns sistemas inteligentes que serão utilizados neste trabalho que são as Redes Neurais Artificiais, destacando o modelo Perceptron Multicamadas e as Redes Auto-Associativas. Os sistemas difusos com ênfase para o modelo difuso Takagi-Sugeno de ordem Zero. Os sistemas Neuro-Difusos ANFIS e TFRENN e por último os Algoritmos Genéticos.

2.1 REDES NEURAIS ARTIFICIAIS

Desde que se renovaram os interesses pelas Redes Neurais Artificiais (RNAs) no início dos anos 80, elas tem sido aplicadas de forma bem sucedida em um largo espectro de domínio, tais como reconhecimento de padrões e aproximação de funções. As RNAs apresentam como sua principal vantagem, a capacidade de aprender por exemplos. Estes sistemas, que se baseiam em uma representação distribuída do conhecimento, são capazes de desenvolver uma representação concisa de conceitos complexos, oferecer boa resistência a ruídos e ainda se adaptar a ambientes instáveis e desconhecidos.

Uma Rede Neural Artificial faz parte de um paradigma de processamento de informação inspirado na maneira densamente interconectada que o cérebro processa a informação, bem como em sua estrutura paralela. A rede é uma coleção de modelos matemáticos que emulam propriedades observadas dos sistemas nervosos biológicos e aponta para analogias do aprendizado adaptativo biológico.

O cérebro humano tem cerca de dez bilhões de unidades de processamento interconectadas, os neurônios. Cada neurônio tem a capacidade de transmitir e receber energia. Todavia, quando um neurônio recebe energia, sua reação não é imediata. O neurônio só envia suas próprias quantidades de energia para outros neurônios após a soma das energias recebidas alcançarem um limiar crítico. Os ajustes da força da conexão entre esses neurônios são responsáveis pelo aprendizado do cérebro. Mesmo que isso seja apenas uma simplificação dos fatos biológicos, é suficientemente poderoso para servir de modelo para a rede neural artificial.

A Rede Neural Artificial, também chamada de modelo de processamento distribuído paralelo, assim como um cérebro humano, tem um grande número de unidades de processamento altamente interconectadas, os neurônios artificiais. A computação em uma RNA é feita por estes neurônios que operam todos de forma paralela. A representação do conhecimento adquirido durante o aprendizado de uma RNA é distribuído pelas conexões entre esses neurônios. Durante o aprendizado, um conjunto de padrões de entrada e saída representativos do problema é apresentado continuamente à rede. O aprendizado é processado de forma a diminuir a diferença entre a saída da rede e a saída que se deseja alcançar de forma a modificar os pesos das conexões.

O padrão de conectividade entre os neurônios, também chamado de topologia da rede neural, pode afetar diretamente a sua capacidade de processamento. Existem diferentes topologias de redes neurais, cada qual com suas vantagens e desvantagens. Entre todas as RNAs, o tipo mais popular é a rede neural multicamadas com fluxo para frente (Feedforward Multilayer Neural Network), também chamada de Perceptron Multicamada (Multilayer Perceptron - MLP).

Os MLPs são conhecidos por possuírem uma poderosa capacidade de expressar um relacionamento entre as variáveis de um problema e, desta forma, se tornam poderosas ferramentas de interpolação. As Redes Neurais são consideradas aproximadores universais, o que significa dizer que é sempre possível projetar uma RNA para aproximar uma determinada função com a precisão que se deseje. As redes MLP geralmente são treinadas com o algoritmo da Retropropagação, ou *Backpropagation*, que é considerado um marco na história do desenvolvimento das redes neurais [6]. De fato, apenas após a introdução deste algoritmo as poderosas propriedades das redes neurais foram bem reconhecidas e aceitas.

As redes neurais foram originalmente desenvolvidas como ferramentas para a exploração e reprodução de tarefas de processamento de informação humana tais como fala, visão, olfato, tato, processamento de conhecimento e controle motor. Posteriormente, a maior parte das pesquisas se voltou para atividades como compressão de dados, otimização, reconhecimento de padrões, modelagem de sistemas, aproximação de funções e controle de sistemas.

2.1.1 Neurônio Artificial – Unidade de Processamento

O elemento básico de processamento de uma RNA é chamado de neurônio artificial, ou simplesmente neurônio. Ele é um modelo baseado nas propriedades fundamentais de um neurônio biológico. O neurônio artificial (Figura 2.1) é composto por três elementos funcionais: um conjunto de sinapses, caracterizadas pelos pesos, uma junção somatória e uma função de ativação.

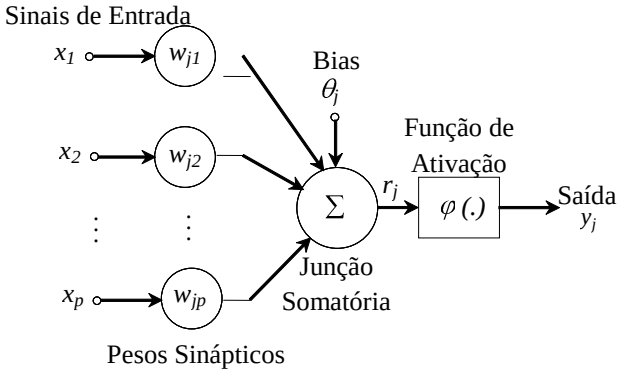


Figura2.1- Neurônio Artificial

Os pesos estabelecem a conectividade particular entre a fonte do sinal e o neurônio. Eles são usados para atenuar ou amplificar, através de um fator de peso w_{ij}, o sinal de entrada x_i vindo do ambiente ou de outros neurônios. O peso w_{ij} é positivo se a sinapse associada for excitatória e negativo se ela for inibitória.

A junção somatória implementa a soma ponderada das entradas de acordo com a expressão:

$$r_j = \sum_{i=1}^{p} w_{ij} x_i + \theta_j \qquad (2.1)$$

onde θ_j é um *bias* aplicado externamente que tem o efeito de incrementar ou reduzir o sinal da junção somatória.

A função de ativação recebe o sinal da junção somatória e então calcula o estímulo interno ou nível de ativação do neurônio. Baseado nesse nível, o neurônio pode ou não produzir uma saída. O relacionamento entre o nível de ativação interna e a saída pode ser linear ou não-linear. A função de ativação limita a amplitude da

saída na faixa de [0 1], ou alternadamente [-1 1]. Considerando a função de ativação $\varphi(.)$, a saída do neurônio pode ser calculada por:

$$y_j = \varphi(\sum_{i=1}^{p} w_{ij} x_i + \theta_j) \tag{2.2}$$

A função de ativação é também conhecida como função de transferência e os três principais tipos reportados na literatura são:

1. *Função Limiar*. Para esse tipo de função, mostrado na Figura 2.2a, temos:

$$\varphi(r_j) = \begin{cases} 1, \text{se } r_j \geq 0 \\ 0, \text{se } r_j < 0 \end{cases} \tag{2.3}$$

2. *Função Linear por Partes*. Essa função, mostrado na Figura 2.2b, é descrita por:

$$\varphi(r_j) = \begin{cases} 1 & \text{se } r_j \geq 1/2 \\ r_j & \text{se } 1/2 > r_j > -1/2 \\ 0 & \text{se } r_j \leq -1/2 \end{cases} \tag{2.4}$$

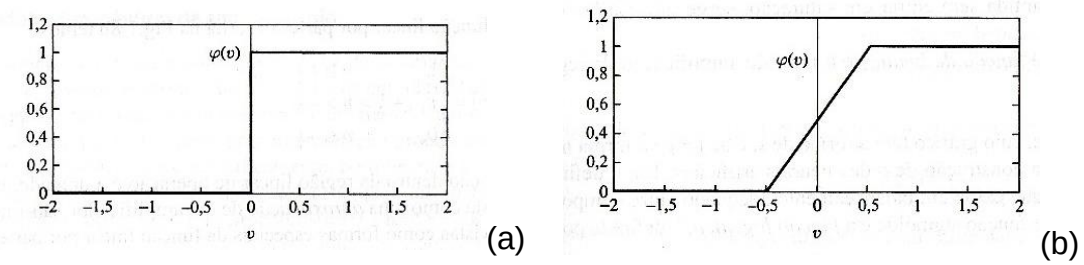


Figura 2.2 - Funções de ativação. (a) Limiar (b) Linear por partes

3. *Função Sigmoide*. É uma função estritamente crescente que exibe suavidade e propriedades assintóticas. Alguns exemplos de função sigmoide são:

- *Função Logística*. Essa função sigmoide, mostrada na Figura 2.3a, é descrita por:

$$\varphi(r_j) = \frac{1}{1 + e^{-ar_j}} \tag{2.5}$$

onde *a* é o parâmetro de declive (ou inclinação) da função. A função limita a saída *a* no intervalo [0 1].

- *Função Tangente Hiperbólica*. Essa função sigmoide, mostrada na Figura 2.3b, é descrita por:

$$\varphi(r_j) = \frac{1 - e^{-ar_j}}{1 + e^{-ar_j}} \tag{2.6}$$

onde *a* é o parâmetro de declive da função. A saída está limitada ao intervalo [-1 1].

A função sigmoide é a função de ativação mais usual em redes neurais e isso se deve principalmente pela sua propriedade de diferenciabilidade, que é uma característica importante da teoria de aprendizado das redes neurais.

De acordo com o uso comum, a função $g:\Re\to\Re$ é chamada sigmoide se o seguinte critério for satisfeito:

$$\begin{cases} \lim_{r\to-\infty} g(r) = 0 \ \text{(não - simétrico) ou} \ lim_{r\to-\infty} g(r) = -1 \ \text{(anti - simétrico)} \\ \lim_{r\to\infty} g(r) = 1 \end{cases} \tag{2.7}$$

Para o escopo do método TFRENN, que será posteriormente apresentado e utilizado neste trabalho, uma função sigmoide antissimétrica, não usual em redes neurais, será também introduzida. Essa função, chamada de *função sigmoide positiva*, obedece ao critério mostrado em (2.7).

A função sigmoide positiva, mostrada na Figura 2.4, é definida por:

$$g(x) = \begin{cases} 1 - e^{-ax} & x \geq 0 \\ e^{ax} - 1 & x < 0 \end{cases} \tag{2.8}$$

onde *a* é o parâmetro de declive da função. A saída e limitada no intervalo [-1 1].

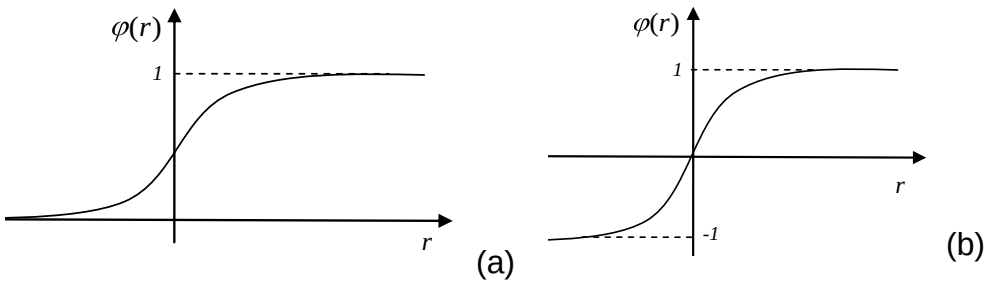


Figura 2.3 - Funções de ativação sigmoides. (a) Logística. (b) Tangente Hiperbólica

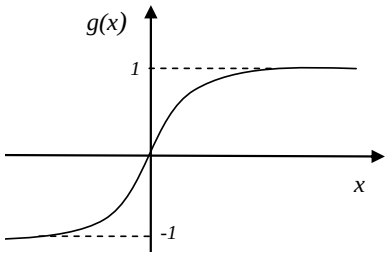


Figura 2.4 - Função Sigmoide Positiva

2.1.2. Perceptron Multicamada - MLP

Motivadas por diferentes redes no sistema biológico, algumas arquiteturas de RNAs têm sido propostas na literatura. Dentre essas, a mais comumente usada é a Rede Neural Multicamadas com Fluxo para Frente (Feedforward), também chamada de Perceptron Multicamadas (Multilayer Perceptron-MLP), mostrado na Figura 2.5.

Em sua forma mais básica, o MLP é um modelo que consiste em um número finito de sucessivas camadas de neurônios. Os neurônios de cada camada são conectados aos neurônios da camada subsequente através de pesos sinápticos. A informação flui de uma camada para a próxima. As entradas para o MLP estão na primeira camada, chamada de camada de entrada. Essa camada é seguida das camadas intermediárias, chamadas de camadas escondidas. A saída da rede neural é obtida na última camada, a camada de saída.

Considerando o MLP na Figura 2.5 com apenas uma camada escondida, cada neurônio nessa camada calcula:

$$s_j = f\left(\sum_{i=1}^{p} x_i w_{ij} + \theta_j\right) \tag{2.9}$$

onde x_i é a *i*-ésima entrada para a rede, w_{ij} é o peso da conexão do neurônio de entrada *i* para o neurônio escondido *j*, θ_j é o bias do *j*-ésimo neurônio escondido e *f*(.) é a função de ativação do neurônio.

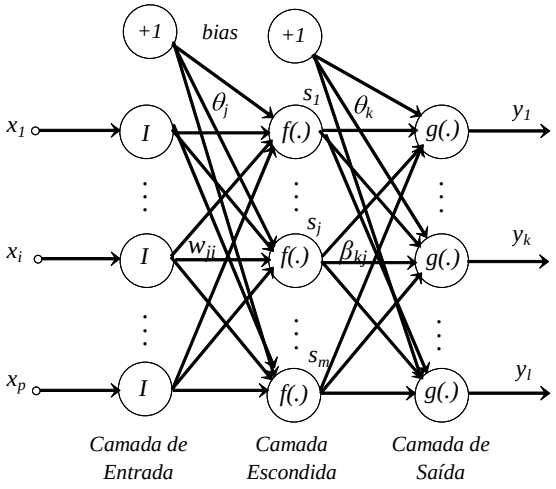


Figura 2.5 – Rede MLP – Perceptron Multicamadas.

Considerando a camada de saída, cada neurônio calcula:

$$y_k = g(\sum_{j=1}^{m} \beta_{kj} s_j) + \theta_k \tag{2.10}$$

onde β_{kj} é o peso da conexão do neurônio escondido *j* com o neurônio de saída *k*, y_k é a *k*-ésima saída da rede, θ_k é o bias da *k*-ésima saída e *g*(.) é a função de ativação do neurônio.

Entre todas as propriedades do MLP, a mais interessante é a sua capacidade de aproximação de funções. Já foi extensamente demonstrado que os MLPs são aproximadores universais. *Hornik et al*, publicaram em [7] o primeiro resultado afirmando que um MLP com um número suficiente de unidades escondidas, usando funções de ativação sigmoidais, é capaz de aproximar virtualmente qualquer função $\delta \in C(I^n)$ de interesse com qualquer grau de exatidão. Formalmente:

$$\left| \delta(x) - \sum_{j=1}^{m} \beta_j f(\sum_{i=1}^{p} w_{ij} x_i + \theta_j) \right| < \varepsilon \text{, para qualquer } \varepsilon > 0 \tag{2.11}$$

Este resultado foi estendido por *Stinchcombe and White* [8] que demonstraram que, mesmo se a função de ativação usada na camada escondida for uma função não-linear bastante geral, o mesmo tipo de MLP ainda assim continua

sendo um aproximador universal. Também demonstraram resultados semelhantes *Funahashi* [9], *Cybenko* [10], *Kreinovichi* [11] e Ito [12].

Em [13], o conjunto de funções de ativação empregado por Hornik foi generalizado. Neste trabalho, foi demonstrado que se a função de ativação (2.11) for contínua, limitada e não constante, então um MLP com suficiente número de neurônios na camada escondida tem a capacidade de aproximar arbitrariamente bem todo $f \in C(I^n)$. Em [14] os resultados foram estendidos para o caso de funções não lineares na camada de saída.

Baseado nas provas de aproximador universal foi concluído que a capacidade de aproximação do MLP não depende da escolha de algum tipo específico de função de ativação, pelo contrário, é a própria arquitetura do MLP que dá à rede neural o potencial de ser um aproximador universal.

Embora os MLPs sejam considerados aproximadores universais, e tenha sido provado que eles podem encontrar um mapeamento entrada-saída se usadas unidades suficientes na camada escondida, ainda não se sabe exatamente como se conseguir um modelo ótimo. Não se sabe ao certo sobre o número exato de neurônios na camada escondida que permitirá a rede neural encontrar uma boa ou a melhor aproximação para o problema.

2.1.3 Aprendizagem na Rede Neural

De acordo com *Mendel* e *McClaren* [15]: *"Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo contínuo de estímulo pelo ambiente no qual a rede está inserida. O tipo da aprendizagem é determinado pela forma como a troca de parâmetros se dá".*

A habilidade de aprender a partir de exemplos é a propriedade mais interessante de uma rede neural e é realizada através de um processo iterativo de ajustes aplicados aos seus pesos sinápticos e limiares.

Para que se possa realizar um processo de aprendizagem, é preciso primeiramente se ter um modelo do ambiente no qual a rede neural vai operar, ou seja, precisamos saber qual informação está disponível para a rede neural. Este modelo é conhecido como paradigma da aprendizagem. De forma geral, existem

dois principais paradigmas de aprendizagem: Aprendizagem Supervisionada e Não-Supervisionada.

Na aprendizagem supervisionada, as saídas da rede neural são comparadas com as saídas desejadas, chamadas também de saídas-alvo e o erro é calculado. Os pesos são então ajustados de forma a minimizar esse erro.

Na aprendizagem não-supervisionada, os pesos são determinados como resultado de um processo de auto-organização, isto é, as conexões com os pesos da rede não são executados por um agente externo. A rede em si decide qual saída é a melhor para uma determinada entrada e a partir daí se reorganiza.

Dentre os algoritmos usados para executar aprendizado supervisionado, o Backprogation emergiu como o mais largamente usado e bem-sucedido algoritmo para treinamento de redes MLP.

2.1.3.1 Algoritmo Backpropagation

O algoritmo de retropropagação do erro, ou *error-backpropagation*, ou simplesmente *backpropagation* é do tipo iterativo e se baseia na minimização de uma soma quadrática do erro utilizando o método do gradiente descendente. Esse algoritmo de aprendizagem é composto de duas fases principais:

- *Propagação* – onde a informação flui para frente (na direção da saída). Nessa fase, a ativação dos neurônios escondidos é propagada para os neurônios da saída. Então se calcula o erro entre os valores de saída e os valores desejados (alvo).
- *Retropropagação* - onde a informação flui para trás (na direção da entrada). Nesta etapa, o erro é propagado na direção inversa e os pesos são atualizados conectando diferentes níveis de unidades.

Durante o processo de aprendizagem, um conjunto de padrões de entradas e saídas é apresentado continuamente à RNA visando minimizar a diferença entre a saída da rede e a saída objetivo através da modificação dos pesos das conexões.

Considerando a RNA da Figura (2.5), o sinal de erro do neurônio de saída *k* na iteração *n* é calculado por:

$$e_k(n) = d_k(n) - y_k(n) \tag{2.12}$$

Para todos os neurônios de saída incluídos no conjunto *C* da RNA, a soma instantânea dos erros quadrados da rede é dada por:

$$\mathrm{E}(n) = \sum_{k \in C} e_k^2(n) \tag{2.13}$$

e para todos os *N* padrões apresentados á RNA, o erro médio quadrado é definido por:

$$\mathrm{E}_{av} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{E}(n) \tag{2.14}$$

O erro médio quadrado E_{av} é uma função de todos os parâmetros livres da rede neural (pesos e bias) e ele representa a função custo que deve ser minimizada durante o processo de aprendizagem.

O ajuste dos pesos, na camada *l*, é feito de acordo com a regra delta generalizada:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \tag{2.15}$$

onde η é o parâmetro taxa de aprendizagem e $\delta_j^{(l)}(n)$ é o gradiente local, calculado por:

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n) g_j'(r_j^{(L)}(n)) & \text{para o neurônio } j \text{ na camada de saída } L \\ f_j'(r_j^{(l)}(n)) \sum \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) & \text{para o neurônio } j \text{ na camada escondida } l \end{cases} \tag{2.16}$$

A partir de (2.16), é possível perceber que a principal restrição para a função de ativação é que ela precisa ser uma função diferenciável. Por este motivo (ser diferenciável), a função sigmoide é a função de ativação mais usada nos neurônios da camada escondida.

2.1.3.2 Modificações e Extensões do Backpropagation

A teoria da aprendizagem deve lidar com três questões importantes: capacidade, complexidade das amostras e complexidade do tempo. A capacidade de aprendizado de uma rede neural é o quanto ela consegue aprender a partir dos exemplos. Complexidade das amostras se refere ao número de exemplos aleatórios necessários para o algoritmo de aprendizagem produzir uma hipótese correta. Já a complexidade do tempo lida com o quão rápido o sistema pode aprender.

Quando se fala de complexidade, trata-se de complexidade computacional do algoritmo de aprendizagem usado para estimar uma solução a partir de padrões de treinamento. Muitos dos algoritmos de aprendizagem existentes hoje em dia têm uma alta complexidade computacional e o *Backpropagation* usado em MLPs é um bom exemplo disso. Sua convergência lenta faz com que, computacionalmente, demande muito esforço. Para lidar com esse tipo de problema, algumas modificações e extensões foram propostas para melhorar o Backpropagation. A seguir se apresenta alguns dos métodos já propostos e apresentados na literatura para melhoria do Backpropagation.

Backpropagation com Momento [16]: Embora taxas de aprendizagem maiores resultem em um aprendizado mais rápido, isso também leva à oscilação sem direção. Uma forma de se driblar esta tendência é modificar (2.15), adicionando um termo de momento:

$$w_{ji}^{(l)}(t+1) = w_{ji}^{(l)}(t) + \alpha \left[w_{ji}^{(l)}(t-1) \right] + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \qquad (2.17)$$

onde α é uma constante positiva com valor pequeno . Um α maior incrementa a influência da última mudança de peso na atual mudança de peso. Essa alteração, no fundo, filtra as oscilações de alta frequência nas mudanças dos pesos já que tendem a cancelar as mudanças nos pesos em direções opostas e reforça a direção predominante da mudança.

Algoritmo Quickprop [17]:Também chamado de algoritmo de aprendizado de segunda ordem, que usa um método de interpolação para estimar mais precisamente o erro mínimo para cada peso. A ideia básica é estimar o ajuste nos pesos assumindo a curvatura da superfície do erro como uma parábola. As

mudanças nos pesos são então feitas com o uso de regras heurísticas que asseguram que haverá uma tendência de “descida do morro” em busca dos mínimos globais em todos os espaços.

Algoritmo RPROP (Resilient Backpropagation)[18]: Esse método se chama *resilient* (resistente) porque ele usa a topologia local da superfície do erro para fazer um ajuste nos pesos mais apropriado. Um valor de atualização pessoal é introduzido para cada peso, que evolui durante o processo de aprendizado de acordo com sua visão local da função erro. Esse método é bastante poderoso e eficiente porque o tamanho do passo do peso não é influenciado pelo tamanho da derivada parcial. Ele é apenas influenciado pela sequência de sinais das derivadas, que dão uma boa ideia sobre a topologia da função erro local.

Algoritmo Levenberg-Marquardt [19]: Essa técnica de otimização é mais poderosa do que a do gradiente descendente, mas requer bem mais memória. Para este algoritmo, o índice de desempenho a ser otimizado é definido por:

$$F(\mathbf{w}) = \sum_{p=1}^{P}\left[\sum_{k=1}^{K}(d_{kp} - o_{kp})^2\right] \qquad (2.18)$$

onde $\mathbf{w} = [w_1\ w_2\ ...\ w_N]^T$ é o vetor de todos os pesos da rede neural, d_{kp} é a saída desejada do *k*-ésimo neurônio de saída e o *p*-ésimo padrão, o_{kp} é o valor atual do *k*-ésimo neurônio de saída e o *p*-ésimo padrão, *N* é o número de pesos, *P* é o número de padrões e *K* é o número de neurônios de saída.

A equação (2.18) pode ser escrita como:

$$F(\mathbf{w}) = \mathbf{E}^T\mathbf{E} \qquad (2.19)$$

Com

$$\mathbf{E} = [e_{11}...e_{k1}\ \ e_{12}...e_{K2}...e_{1P}...e_{KP}]^T \qquad (2.20)$$
$$e_{kp} = d_{kp} - o_{kp},\ k = 1,...,K,\ p = 1,...,P$$

onde **E** é o vetor de erros acumulativo (para todos os padrões). A partir de (2.20), a matriz Jacobiana é definida por:

$$\mathbf{J} = \begin{bmatrix} \frac{\delta e_{11}}{\delta w_1} & \frac{\delta e_{11}}{\delta w_2} & \cdots & \frac{\delta e_{11}}{\delta w_N} \\ \frac{\delta e_{21}}{\delta w_1} & \frac{\delta e_{21}}{\delta w_2} & \cdots & \frac{\delta e_{21}}{\delta w_N} \\ \vdots & \vdots & & \vdots \\ \frac{\delta e_{K1}}{\delta w_1} & \frac{\delta e_{K1}}{\delta w_2} & \cdots & \frac{\delta e_{K1}}{\delta w_N} \\ \vdots & \vdots & & \vdots \\ \frac{\delta e_{1P}}{\delta w_1} & \frac{\delta e_{1P}}{\delta w_2} & \cdots & \frac{\delta e_{1P}}{\delta w_N} \\ \frac{\delta e_{2P}}{\delta w_1} & \frac{\delta e_{2P}}{\delta w_2} & \cdots & \frac{\delta e_{2P}}{\delta w_N} \\ \vdots & \vdots & & \vdots \\ \frac{\delta e_{KP}}{\delta w_1} & \frac{\delta e_{KP}}{\delta w_2} & \cdots & \frac{\delta e_{KP}}{\delta w_N} \end{bmatrix} \tag{2.21}$$

A matriz Jacobiana contém as primeiras derivadas dos erros da rede em relação aos pesos.Os pesos são calculados por:

$$w_{t+1} = w_t - \left(\mathbf{J}_t^T \mathbf{J}_t + \alpha_t \mathbf{I}\right)^{-1} \mathbf{J}_t^T \mathbf{E}_t \tag{2.22}$$

onde **I** é a matriz identidade, α é o parâmetro de aprendizagem e **J** é uma Jacobiana do erro de saída *m* em relação aos *n* pesos da rede neural. O parâmetro α é automaticamente ajustado, a cada iteração, de modo a garantir convergência. Se α for muito grande, a expressão acima se aproxima do gradiente descendente. Para pequenos valores de α a expressão acima tende a funcionar como o método de Gauss-Newton.

2.1.4 Redes Neurais Auto-Associativas

Uma rede MLP Auto-Associativa é um caso especial de rede MLP na qual o vetor de entrada e a saída desejada são idênticos, desta forma, o mapeamento resulta naquilo que é chamado de armazenamento de vetor.

Para o caso de classificação de padrões, as redes Auto-Associativas podem ser treinadas para realizar uma associação entre os padrões de entrada e saída sendo que, após o treinamento, quando são apresentados para a rede padrões cada

vez mais distantes dos padrões utilizados durante o aprendizado, o erro na saída da rede será cada vez maior.

A Figura 2.6 ilustra uma rede Auto-Associativa com a camada escondida de menor dimensão que as camadas de entrada e saída, que são iguais. O processo projeta a informação de um espaço *S* para um espaço *S'*, emulando uma função *f*, e efetua a projeção inversa de *S'* para *S* emulando a função inversa f^{-1}.

Se o vetor de saída for igual ao de entrada, toda a informação flui através do "gargalo" *S'* e se recompõe em *S*. Portanto, a rede "aprende" o sistema real e o memoriza nos pesos. Um vetor de entrada que não seja coerente com o sistema aprendido produzirá na saída um vetor distinto (com erro eventualmente grande) porque esse vetor não se projeta corretamente no espaço não linear *S'* e a sua reconstrução por f^{-1} não é possível.

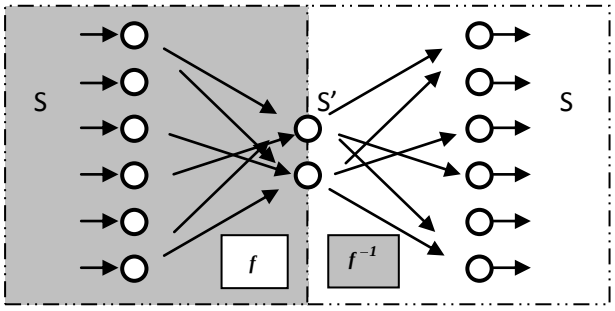


Figura 2.6 – Rede Neural Auto-Associativa

2.2 SISTEMAS DIFUSOS

A Lógica Difusa, concebida por *Lofti Zadeh* nos anos 60, consiste em uma variedade de conceitos e técnicas para representação e inferência do conhecimento impreciso, incerto e não confiável. A introdução desta nova teoria melhorou a abordagem idealística da matemática tradicional. Os novos conceitos e ideias por trás da lógica difusa levaram a representações matemáticas mais realísticas e precisas da percepção de verdade, onde percepção remete ao cérebro humano e a sua maneira de observar e expressar a realidade.

A lógica difusa trabalha com a imprecisão das variáveis de entrada e saída do sistema pela definição de números difusos e conjuntos difusos que podem ser expressos em termos de variáveis linguísticas, como por exemplo, pequeno, médio e grande. Os conjuntos difusos são combinados com regras difusas obtidas de

especialistas humanos ou baseadas no conhecimento do domínio. A coleção adquirida de regras *se-então* é então combinada em um sistema único. Diferentes sistemas difusos usam diferentes princípios para fazer esta combinação. Existem três tipos de sistemas difusos geralmente usados na literatura: sistemas puramente difusos, sistemas *Takagi-Sugeno* e sistemas difusos com fuzzificador e defuzzificador (sistemas *Mandani*).

2.2.1 Teoria dos Conjuntos Difusos

Definição 2.1: Conjuntos Difusos

Os conjuntos difusos são uma generalização dos clássicos conjuntos convencionais (*crisp*) ou puros. Eles são funções que mapeiam um valor ou um membro do conjunto em um número na faixa entre zero e um, indicando seu atual grau de pertinência.

Seja *U* um conjunto não vazio, chamado de Universo, cujos elementos são denotados por *x*. O conjunto difuso *A* em *U* é caracterizado pela função de pertinência:

$$\mu_A(x): x \rightarrow [0,1] \tag{2.23}$$

O conjunto difuso *A* também pode ser representado como um conjunto de pares ordenados de um elemento genérico *x* e seu valor de pertinência:

$$A = \{x, \mu_A(x)\}, x \in U \tag{2.24}$$

O formato geométrico de uma função de pertinência é a caracterização de incerteza na variável difusa correspondente. A função de pertinência triangular é mais frequentemente usada e a mais prática. Outros formatos também encontrados são trapezóide, função-s, função-pi e função-z.

Definição 2.2: Variável Linguística

Uma variável linguística é uma variável que pode assumir como seus valores palavras da linguagem natural e essas palavras são caracterizadas pelos conjuntos difusos definidos no universo de discurso que a variável foi definida [20]. É caracterizada como *X*, *T*, *U*, *M*, onde *X* é o nome da variável linguística, *T* é o conjunto de valores linguísticos que *x* pode assumir, *U* é o domínio físico atual no qual a variável linguística assume seus valores quantitativos e *M* é a regra semântica que relaciona cada valor linguístico em *T* com um conjunto difuso em *U*.

O conceito de variável linguística permitiu a introdução do conhecimento humano em sistemas de uma forma sistemática e eficiente. As variáveis linguísticas permitem formular descrições vagas em linguagem natural de forma matematicamente precisa.

A Figura 2.7 mostra um exemplo de cinco conjuntos difusos no universo de discurso *U* representando o intervalo de possíveis valores para a variável temperatura. Neste caso, “Temperatura Ambiente” é a variável linguística com cinco termos “muito baixa”, “baixa”, “média”, “alta” e “muito alta”, representadas pelos conjuntos difusos com funções de pertinência.

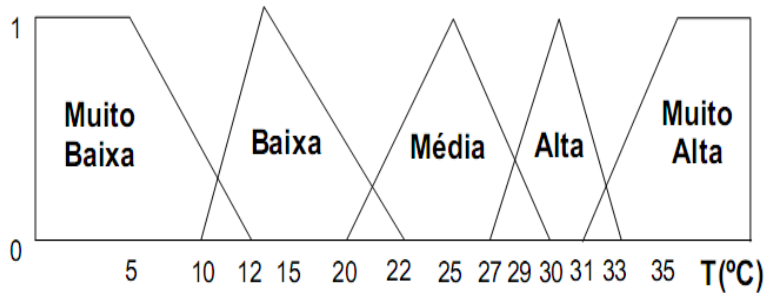


Figura 2.7 - Conjuntos difusos associados à variável “Temperatura Ambiente”

Definição 2.3: Modificadores e Delimitadores Linguísticos

São operadores que alteram as funções de pertinência para os conjuntos difusos associados aos rótulos linguísticos. O significado do conjunto transformado pode ser facilmente interpretado a partir do significado do conjunto original. Um pequeno exemplo de delimitadores linguísticos e seu papel na lógica difusa é listado na Tabela 2.1.

Tabela 2.1 – Delimitadores Linguísticos

Delimitador	Função
Muito, Extremamente	Concentração
De algum modo	Diluição
Definitivamente, Aproximadamente	Intensificação
Mais ou Menos	Relaxamento
Não	Negação
Abaixo, Acima	Restrição

Definição 2.4: União Difusa – normas-S

Considerando $s: [0,1] \times [0,1] \to [0,1]$ como o mapeamento que transforma as funções de pertinência dos conjuntos difusos A e B nas funções de pertinência da união de A com B, isto é, $s[\mu_A(x), \mu_B(x)] = \mu_{A \cup B}(x)$.

A função *s* é qualificada como sendo a união difusa ou norma-S se satisfizer pelo menos os seguintes requisitos:

1) $s(0,1) = 1, s(0,a) = s(a,0) = a$ (condição limite)
2) $s(a,b) = s(b,a)$ (condição comutativa)
3) se $a \leq a'$ e $b \leq b'$, então $s(a,b) \leq s(a',b')$ (condição não decrescente)
4) $s(s(a,b),c) = s(a,s(b,c))$ (condição associativa)

A tabela 2.2 lista algumas normas-S propostas na literatura.

Tabela 2.2 – normas-S

norma-S	
Soma de Einstein	$s(a,b) = \dfrac{a+b}{1+a+b}$
Soma Drástica	$s(a,b) = \begin{cases} a \text{ se } b = 0 \\ b \text{ se } a = 0 \\ 1 \text{ em outros casos} \end{cases}$
Soma Algébrica	$s(a,b) = a + b - ab$
Máximo	$s(a,b) = \max(a,b)$

Definição 2.5: Intersecção Difusa – As Normas-T

Considerando $t: [0,1] \times [0,1] \rightarrow [0,1]$ como sendo o mapeamento que transforma as funções de pertinência dos conjuntos difusos A e B nas funções de pertinência da intersecção de A com B, isto é, $t[\mu_A(x), \mu_B(x)] = \mu_{A \cap B}(x)$.

A função *t* é qualificada como sendo uma intersecção difusa ou norma-t se satisfizer pelo menos os seguintes requisitos:

1) $t(0,0) = 0, t(a,1) = t(1,a) = a$ (condição limite)

2) $t(a,b) = t(b,a)$ (condição comutativa)

3) $se\ a \leq a'\ \text{e}\ b \leq b', \text{então}\ t(a,b) \leq t(a',b')$ (condição não decrescente)

4) $t(t(a,b),c) = t(a,t(b,c))$ (condição associativa).

A tabela 2.3 lista algumas normas-T propostas na literatura.

Tabela 2.3 – normas-T

norma-T	
Produto Einstein	$t(a,b) = \dfrac{ab}{2-(a+b-ab)}$
Produto Drástico	$t(a,b) = \begin{cases} a \text{ se } b = 1 \\ b \text{ se } a = 1 \\ 0 \text{ em outros casos} \end{cases}$
Produto Algébrico	$t(a,b) = ab$
Mínimo	$t(a,b) = \min(a,b)$

Definição 2.6: Complemento Difuso

Considerando $c: [0,1] \rightarrow [0,1]$ como sendo o mapeamento que transforma a função de pertinência do conjunto difuso A na função de pertinência do complemento de A, isto é, $c[\mu_A(x)] = \mu_{\bar{A}}(x)$.

A função *c* é qualificada como sendo o complemento difuso se satisfizer pelo menos os seguintes requisitos:

1) $c(0)=1 \text{ e } c(1)=0$ (condição limite)

2) Para todo $a,b \in [0,1]$, se $a<b$ então $c(a) \geq c(b)$ (condição não decrescente), onde *a* e *b* denotam pertinência de algum conjunto difuso, dizemos que $a=\mu_A(x) \text{ e } b=\mu_B(x)$.

Definição 2.7: Classe Associada – Lei de DeMorgan

Para cada norma-S existe uma norma-T associada, onde o termo associado significa que existe um complemento difuso de modo que os três juntos satisfazem a lei de DeMorgan. Especificamente, a norma-S $s(a,b)$, a norma-T $t(a,b)$ e o complemento difuso $c(a)$ formam uma classe associada se:

$$c[s(a,b)]=t[c(a),c(b)] \qquad (2.25)$$

Definição 2.8: Base de Regras Difusa

Uma base de regras difusa consiste em um conjunto de regras difusas SE-ENTÃO que especificam uma relação linguística entre o rótulo linguístico das variáveis de entrada e saída do sistema. É o coração do sistema difuso, pois todos os outros componentes são usados para implementar estas regras de uma maneira eficiente e inteligível. Especificamente, a base de regras difusas compreende as seguintes regras difusas SE-ENTÃO:

$$\text{Regra } R^1\text{: SE } x_1 \text{ é } A_1^1 \text{ e } \ldots x_n \text{ é } A_n^1 \text{ ENTÃO } y \text{ é } B^1$$
$$\vdots$$
$$\text{Regra } R^m\text{: SE } x_1 \text{ é } A_1^m \text{ e } \ldots x_n \text{ é } A_n^m \text{ ENTÃO } y \text{ é } B^m \qquad (2.26)$$

onde A_i e B são conjuntos difusos em $U \subset \Re$ e $V \subset \Re$, respectivamente, e $X=(x_1,x_2,\ldots x_n) \in U$ e $y \in V$ são a entrada e saída do sistema difuso, respectivamente.

A parte **SE** da regra é chamada de premissa ou antecedente, enquanto a parte **ENTÃO** é chamada conclusão ou consequente da regra.

2.2.2 Sistema Difuso Baseado em Regras

Um Sistema Difuso Baseado em Regras, também chamado de Sistema de Inferência Difusa (SID), como mostrado na Figura 2.8, é composto de quatro blocos funcionais:

- *Fuzzificação*. Normalmente, as entradas em um sistema difuso são valores convencionais (*crisp*) e estes são então convertidos em conjuntos difusos (*fuzzy*). O bloco de fuzzificação transforma as entradas convencionais em graus de combinação com valores lingüísticos.
- *Banco de Dados e Base de Regras.* O banco de dados define as funções de pertinência dos conjuntos difusos usados nas regras difusas SE-ENTÃO que compõem a base de regras. Normalmente, a base de regras e o banco de dados são conjuntamente referidos como *Base de Conhecimento*.
- *Máquina de Inferência.* É quem realiza a inferência sobre as regras difusas e produz um valor difuso para a saída do sistema.
- *Defuzzificação.* Converte um conjunto de variáveis difusas (fuzzy) em valores convencionais (crisp) de forma a permitir que a saída do sistema difuso seja aplicada a outro sistema não difuso.

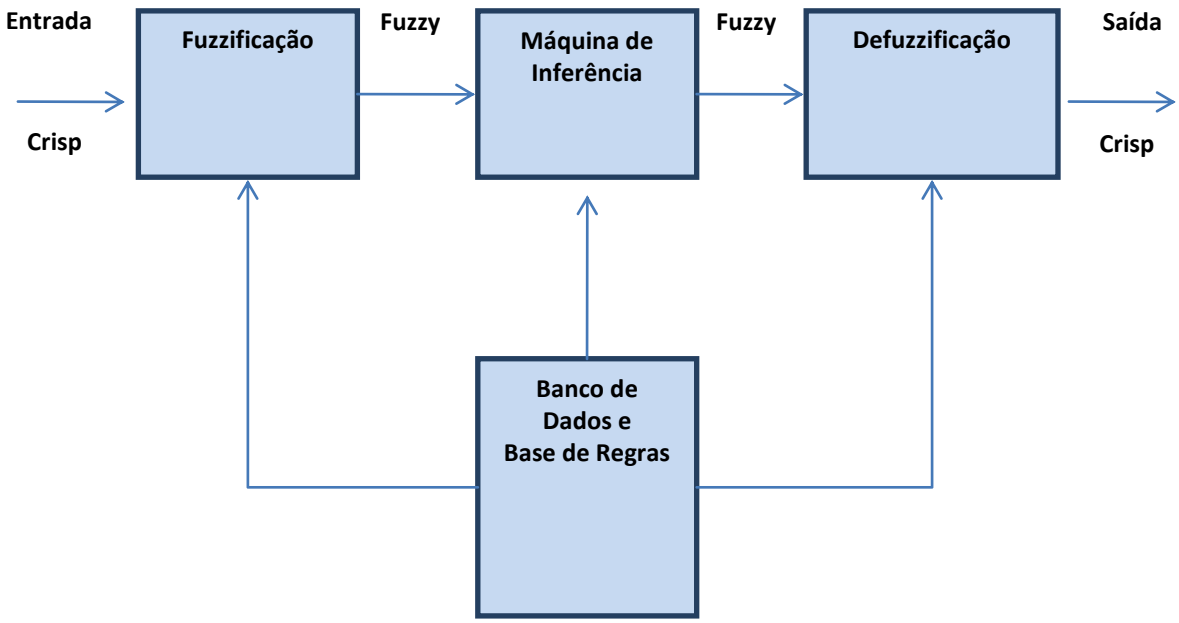


Figura 2.8 – Diagrama Esquemático do Sistema de Inferência Difusa

2.2.3. Modelo Difuso Takagi-Sugeno

O Sistema de Inferência Difuso pode ser categorizado em duas famílias:

- A família de modelos linguísticos baseados em coleções de regras SE-ENTÃO, cujos antecedentes e conseqüentes utilizam valores difusos como inferência difusa *Mamdani*.
- A família dos modelos que usam a estrutura da regra com um antecedente difuso e um consequente funcional (*crisp*).

A segunda família, baseada nos sistemas de inferência difusa *Takagi-Sugeno* (TS), é construída com regras no seguinte formato:

Regra R^j: SE x_1 é A_{1j} E … E x_n é A_{nj} ENTÃO $y_j = g(x_1,..., x_n)$ (2.27)

onde A_{ij} é um conjunto difuso e x_i é a entrada do sistema. O consequente da regra é uma função afim linear ou não-linear das variáveis de entrada.

O modelo difuso *Sugeno* foi proposto por *Takagi, Sugeno* e *Kang* [21] num esforço para formalizar uma abordagem de sistema para gerar regras difusas a partir de um conjunto de entrada e saída de dados. O modelo difuso Sugeno é também conhecido como modelo *Takagi-Sugeno-Kang* (TSK). Quando y_1 é um polinômio de primeira ordem, temos o modelo difuso *Takagi-Sugeno* de primeira ordem. Quando y_1é uma constante, temos o modelo difuso *Takagi-Sugeno* de ordem zero, que pode ser visto como um caso especial do sistema de inferência difusa *Mamdani* com o consequente sendo um *singleton*. O modelo difuso *Takagi-Sugeno* de ordem zero é construído com regras no seguinte formato:

Regra R^j: SE x_1 é A_{1j} E… E x_n é A_{nj} ENTÃO $y_j = c_j$ (2.28)

O grau de disparo de cada regra é calculado por:

$$v_j = \bigcap_{i=1}^{n} \mu_{ij}(x_i) \qquad (2.29)$$

onde $\mu_{ij}(x_i)$ é a função de pertinência associada ao conjunto difuso A_{ij} e $\cap$ representa o operador produto (operador E).

A saída do sistema é calculada pela média dos pesos de y_1, isto é:

$$f(x) = \frac{\sum_{j=1}^{N} y_j v_j}{\sum_{j=1}^{N} v_j} \tag{2.30}$$

onde$N$ é o número de regras do sistema.

A saída pode ser também calculada por:

$$f(x) = \sum_{l=1}^{N} y_j \overline{v_j} \tag{2.31}$$

A Figura 2.9 ilustra o mecanismo de raciocínio de um TSK de ordem zero.

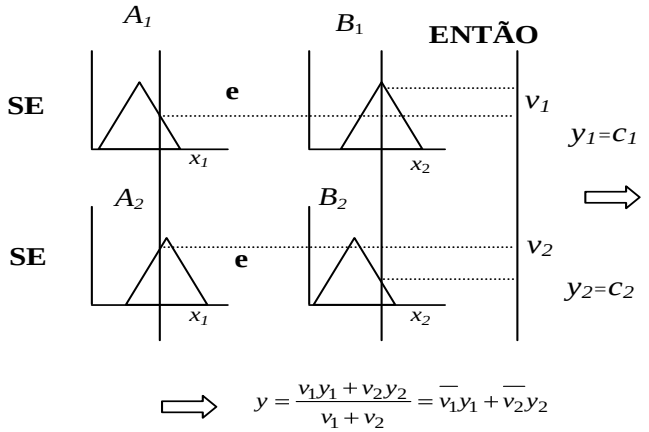


Figura 2.9 – Modelo Difuso Takagi-Sugeno de Ordem Zero

2.2.4 Transparência e Interpretabilidade de Sistemas Difusos

Uma das principais características que distingue um sistema difuso de outras técnicas, com as redes neurais, é a sua capacidade de explicação. Os sistemas difusos têm o poder de expressar o comportamento de sistemas reais de uma maneira compreensível, ou seja, os sistemas difusos são interpretáveis. Porém, essa interpretabilidade só é válida se eles satisfizerem as condições de transparência.

Embora a transparência de um sistema difuso esteja associada ao conceito de interpretabilidade linguística, transparência e interpretabilidade são termos

distintos. A interpretabilidade é uma propriedade inerente ao sistema difuso e normalmente está associada às regras linguísticas e conjuntos difusos, enquanto transparência é a medida de quão válida é a interpretação linguística do sistema [22].

Se o sistema difuso for construído baseado em conhecimento de especialistas, essa condição de transparência pode ser facilmente satisfeita, mas se ele for extraído diretamente a partir dos dados, geralmente a transparência se perde levando a resultados considerados como caixas-pretas, sem qualquer significado linguístico.

De acordo com [22], um sistema difuso é transparente somente se todas as regras da base de regras são transparentes. A regra do sistema difuso é considerada transparente se o seu grau de disparo

$$v_j = \bigcap_{i=1}^{n} \mu_{ij}(x_i) = 1 \tag{2.32}$$

tem como consequente a seguinte saída para o sistema:

$$y = y_j \tag{2.33}$$

ondey_1 é o centro da função de pertinência associada à regra *j*. Para o caso do modelo *Takagi-Sugeno* de ordem zero, y_1 é o consequente constante.

As condições de transparência são definidas com base no grau de sobreposição das funções de pertinência de entrada e da simetria da função de pertinência de saída. Para que o sistema difuso seja transparente, a sobreposição das funções de transparência tem que ser menores do que 50%. Isso garante a existência de pontos de checagem de transparência, que são pontos no espaço de entrada/saída onde a contribuição explícita de uma determinada regra toma lugar e a regra em questão é totalmente ativada. A Figura 2.10 mostra um sistema com regras transparentes [22], onde os asteriscos são pontos de checagem de transparência.

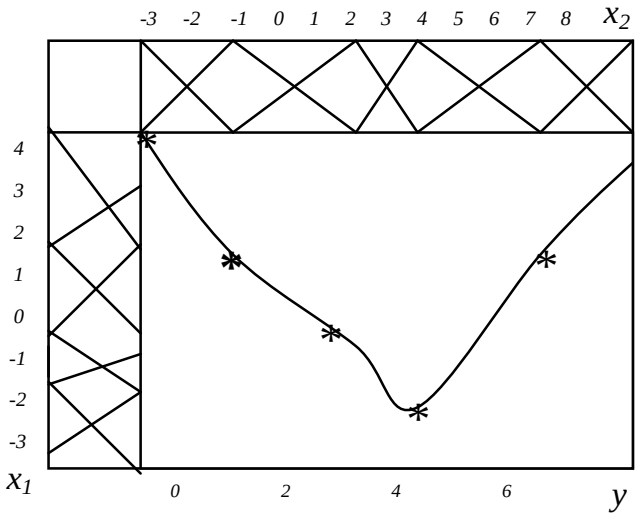


Figura 2.10 – Sistema Difuso Transparente

Todavia, se a sobreposição for maior do que 50%, pelo menos duas regras estarão contribuindo simultaneamente para uma determinada entrada, o que torna a saída sempre o resultado de uma interpolação. Isso faz com que a contribuição da regra na saída se torne imperceptível e o sistema possa ser considerado não transparente. A Figura 2.11 mostra um sistema com regras não transparentes [22].

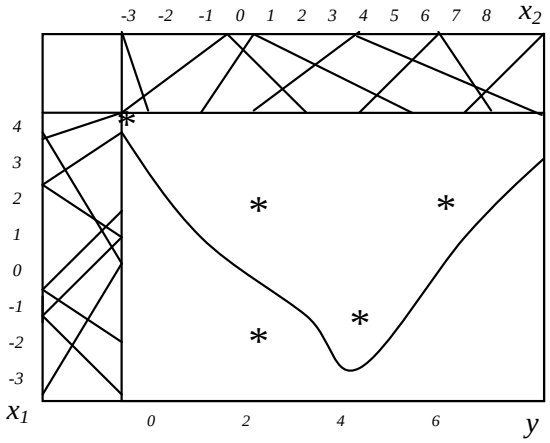


Figura 2.11 – Sistema Difuso Não-Transparente

Vale lembrar que, como os sistemas difusos trabalham com funções de pertinência de suporte infinito, como as funções Gaussianas, a medida de sobreposição não pode ser aplicada diretamente. Nesse caso, a condição de transparência é redefinida com o uso do corte-α.

Alguns pesquisadores se dedicaram a este tópico e hoje existem algoritmos que garantem a transparência dos sistemas difusos. Geralmente, essa transparência

é garantida com a imposição de restrições às funções de pertinência ou usando funções de pertinência que utilizem a transparência como condição predeterminada. Trabalhos nessa área podem ser achados em [22] e [23].

2.3 SISTEMAS NEURO-DIFUSOS – EXTRAÇÃO DE CONHECIMENTO

Diversos trabalhos de pesquisa já foram desenvolvidos na área de extração de regras de redes neurais, em especial fazendo uso de Sistemas Neuro-Difusos (*neuro-fuzzy*). Em um sistema neuro-difuso, os parâmetros de um sistema difuso são ajustados usando-se métodos de aprendizado obtidos das redes neurais. Nestes sistemas, o processo de extração de regras difusas envolve três passos:

1. Inserção do conhecimento especialista já existente na forma de regras difusas na estrutura de uma rede neural (fase de inicialização do conhecimento). Nesta etapa, as representações das funções de pertinência são geradas.
2. Treinamento da rede neural com o algoritmo de aprendizado. Nessa fase, as funções de pertinência são ajustadas de acordo com os padrões no conjunto de dados de treino. É importante observar que, como os algoritmos de aprendizado geralmente são métodos gradiente descendentes, como é o caso do Backpropagation, eles não podem ser aplicados diretamente a um sistema difuso, pois as funções usadas para realizar o processo de inferência normalmente não são diferenciáveis. Para lidar com esse problema as funções usadas nesse problema em particular são diferenciáveis. Se operadores não-diferenciáveis, como *min* e *max* forem usados, então um outro algoritmo de aprendizado que não use gradiente descendente deve ser adotado ou um outro procedimento que se encaixe melhor.
3. Extração de conhecimento refinado com regras modificadas e funções de pertinência difusas adaptadas.

2.3.1 ANFIS

Um importante método descrito na literatura e extensivamente usado para esse propósito é o ANFIS (Adaptative-Network-Based Fuzzy Inference System), introduzido em [24]. Ele implementa um sistema de inferência difusa do tipo *Takagi-*

Sugeno e é estruturado em uma rede neural de cinco camadas, mostrado na Figura 2.12 (considerando apenas duas entradas e duas funções de pertinência).

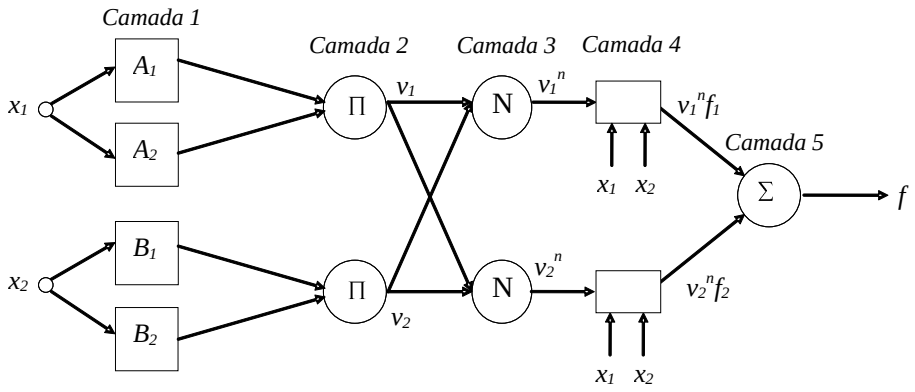


Figura 2.12 – Arquitetura do ANFIS

A fuzzificação das variáveis de entrada é realizada na primeira camada. A segunda camada calcula a parte antecedente da regra usando operadores de norma-T. A terceira camada normaliza as ativações das regras e em seguida a quarta camada determina o parâmetro consequente da regra. A camada de saída calcula a entrada geral como o somatório de todos os sinais de entrada. Para determinar os parâmetros de premissa (para aprender os parâmetros relacionados às funções de pertinência), o ANFIS usa o aprendizado do Backpropagation e para determinar os consequentes da regra, usa a estimação pelos mínimos quadrados.

O passo interno ao procedimento de aprendizado do ANFIS é dividido em duas partes. Na primeira, os padrões de entrada são propagados e os parâmetros consequentes ótimos são estimados por um procedimento iterativo de média de mínimos quadrados, enquanto os parâmetros de premissa são assumidos como sendo fixados pelo atual ciclo pelo conjunto de treino. Na segunda parte, os padrões são propagados novamente e nessa época, o Backpropagation é usado para modificar os parâmetros de premissa, enquanto os parâmetros consequentes permanecem fixos. Ocorre então a iteração desse procedimento.

O aprendizado no ANFIS não fornece meios de aplicar restrições que limitem o tipo de modificações aplicadas às funções de pertinência. Desta forma, algumas vezes se torna difícil interpretar o sistema difuso extraído.

2.3.2 TFRENN

Considerando as vantagens e desvantagens de uma rede neural e de um sistema difuso, em [25] é apresentada uma nova metodologia chamada TFRENN (Transparent Fuzzy Rule Extraction from Neural Network) que realiza o mapeamento de uma rede neural em um sistema difuso, de maneira que o conhecimento aprendido durante a fase de treinamento pode ser extraído em forma de regras difusas transparentes do tipo *Takagi-Sugeno* de Ordem Zero.

A Rede Neural utilizada para o mapeamento no sistema difuso é do tipo Perceptron Multicamadas e tem apenas uma camada escondida e um único neurônio na camada de saída com função de ativação linear. Os neurônios da camada escondida têm uma função de ativação sigmoide particular, denominada *função sigmoide positiva*, cujo gráfico é mostrado na Figura 2.13, e é descrito por:

$$f(x) = \begin{cases} 1 - e^{-x} & x \geq 0 \\ 0 & x < 0 \end{cases} \qquad (2.34)$$

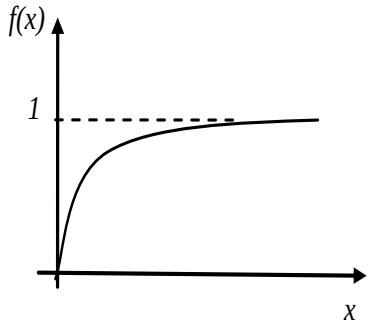


Figura 2.13 - Função Sigmoide Positiva

O conceito de *dualidade-f*, introduzido em [26], permitiu definir uma transformação que estabelece uma equivalência matemática na operação realizada pelos neurônios da camada escondida de uma rede neural, mostrada em (2.9). Esse conceito de dualidade-*f* foi devidamente demonstrado em [27].

Aplicando-se este conceito a uma rede neural sem *bias* θ_j, onde a função de ativação *f* é uma sigmoide positiva (2.34), e considerando que $\sum_{i=1}^{n} x_i w_{ij} \geq 0$, temos que a saída dos neurônios da camada escondida pode ser calculada por:

$$s_j = f(\sum_{i=1}^{n} x_i w_{ij}) = f(x_1 w_{1j}) * \ldots * f(x_n w_{nj}) \quad (2.35)$$
$$= 1 - (1 - f(x_1 w_{1j})) \ldots (1 - f(x_n w_{nj}))$$
$$\text{se} \sum_{i=1}^{n} x_i w_{ij} \geq 0 \text{ e } x_i w_{ij} \geq 0$$

É possível reconhecer em (2.35) um operador lógico comum na lógica difusa, a soma algébrica, que nada mais é do que a norma-*S* (operador OU). Podemos então interpretar $f(x_i w_{ij})$ como sendo uma função de pertinência cuja expressão linguística seria "*x é maior do que* 2.3/w_{ij}", fixando-se $f(2.3/w_{ij}) = 0.9$. A razão para isso é que $f(x)$ só pode alcançar o valor 1 de forma assintótica, então definimos o corte-α para $\alpha = 0.9$, como um limiar linguístico significativo.

Na rede neural mostrada na Figura 2.5, com um único neurônio de saída, sem bias na camada escondida, $\sum_{i=1}^{n} x_i w_{ij} \geq 0$ e $x_i w_{ij} \geq 0$, para cada neurônio na camada escondida, uma regra pode ser extraída na seguinte forma:

$$\textbf{Regra } R_j\textbf{: Se} \sum_{i=1}^{n} x_i w_{ij} \textbf{ é } A \textbf{ Então } y_j = \beta_j \quad (2.36)$$

onde *A* é um conjunto difuso cuja função de pertinência é a sigmoide positiva. Com base em (2.35) e (2.36), podemos escrever as regras como:

$$\textbf{Regra } R_j\textbf{: Se } (x_1 w_{1j} \textbf{ é } A) * \ldots * (x_i w_{ij} \textbf{ é} A) * \ldots \quad (2.37)$$
$$* (x_n w_{nj} \textbf{ é } A) \textbf{ Então } y_j = \beta_j$$

A expressão *"$x_1 w_{1j}$é A"* pode ser interpretada como *"x_1 é A_{1j}",* se o peso w_{ij} for definido como fator de escala do declive de $f(.)$. Desta forma, o conjunto difuso A_{ij} fica definido pela função de pertinência $\mu(A_{ij}) = f(x_i w_{ij})$.

Reconhecendo a operação * como sendo o operador soma algébrica (OU), podemos reescrever (2.37) da seguinte forma:

$$\textbf{Regra } R_j\textbf{: Se } (x_1 \textbf{ é } A_{1j}) \text{ OU} \ldots \text{OU } (x_i \textbf{ é } A_{1j}) \text{ OU} \ldots \text{OU } (x_n \textbf{ é } A_{1j}) \textbf{ Então } y_j \quad (2.38)$$
$$= \beta_j$$

onde o grau de disparo da regra é calculado pelo operador soma algébrica:

$$\begin{aligned} v_j &= \mu(A_{1j}) * .... * \mu(A_{nj}) \\ &= 1 - ((1 - \mu(A_{1j})) ... (1 - \mu(A_{nj})) \end{aligned} \tag{2.39}$$

Finalmente, a saída do sistema difuso é extraída como:

$$y = \sum_{j=1}^{m} \beta_j s_j \tag{2.40}$$

E, desde que $s_j = v_j$ e $\beta_j = y_j$, (2.40) pode ser reescrito como:

$$y = \sum_{j=1}^{m} y_j v_j \tag{2.41}$$

Enfim, obtém-se uma descrição matemática da operação realizada pela rede neural que tem a forma de um sistema de inferência e se assemelha ao modelo *Takagi-Sugeno* de Ordem Zero. Todavia, o operador lógico usado para calcular o grau de ativação de cada regra é do tipo norma-*S* (OU) e não norma-*T* (E) como comumente utilizado em um FIS-TS. Podemos associar norma-*S* e norma-*T* de forma que para cada norma-*S*, exista uma norma-*T* que satisfaça a Lei *DeMorgan* [28].

A norma-*T* associada ao operador soma algébrica *S(a,b)=1–(1-a)(1-b)* é o operador produto algébrico *T(a,b)=ab*. Desta forma, cada regra extraída em (2.38) pode ter a seguinte forma:

Regra *Rj*: **Se** (x_1 **não é** A_{1j}) E …E (x_i **não é** A_{ij}) E…E(x_n **não é** A_{nj}) (2.42)

Então $y_j = \beta_j$

onde o grau de disparo da regra é dado pelo operador produto algébrico (E). Com esta transformação, a saída do sistema difuso em (2.41) passa a ser:

$$y = \sum_{j=1}^{m} \beta_j (1 - v_j) \tag{2.43}$$

E, rearranjando (2.43), temos:

$$y = \sum_{j=1}^{m} \beta_j (1 - v_j) = \sum_{j=1}^{m} \beta_j - \sum_{j=1}^{m} y_j v_j \tag{2.44}$$

onde $\sum_{j=1}^{m} \beta_j$ é o valor default da saída do sistema difuso.

Existe outro objetivo a ser alcançado: fazer com que os antecedentes da regra extraída da rede neural possam ser significativos e interpretáveis por humanos. Para que isso ocorra, algumas restrições foram estabelecidas. Se a negação (NÃO) for aplicada para extrair a função de pertinência $\mu(A_i) = f(x_i w_{ij})$, chega-se a uma nova função de pertinência definida por:

$$f(x_i w_{ij}) = \begin{cases} e^{-x_i w_{ij}} & , x_i w_{ij} \geq 0 \\ 1 & , x_i w_{ij} < 0 \end{cases} \tag{2.45}$$

O peso w_{ij} atua como fator escalar da função $f(.)$ Usando o corte-α com α = 0.999, podemos aproximar (2.45) para:

$$f(x_i) = \begin{cases} e^{-x_i w_{ij}} & , x_i \geq 0.001/w_{ij} \\ 1 & , x_i < 0.001/w_{ij} \end{cases} \tag{2.46}$$

onde $f(0.001/w_{ij}) = 0.999 \approx 1$ (Figura 2.14).

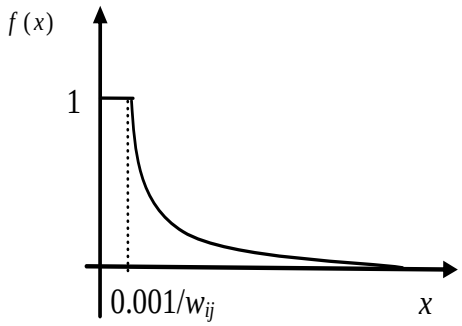


Figura 2.14 - Nova Função de Pertinência Extraída

Esse novo conjunto difuso pode ser interpretado como "*menor do que* $0.001/w_{ij}$" e ele só faz sentido se $0 \leq 0.001/w_{ij} \leq 1$, o que leva a necessidade de que os pesos $w_{ij} \geq 0.001$. Esta restrição está diretamente ligada ao treinamento da rede neural com entradas normalizadas, de forma que todas as funções de pertinência extraídas devem ser definidas para o respectivo intervalo de entrada.

Com $w_{ij} \geq 0.001$, $0 \leq x_i \leq 1$ e $\theta_j \geq 0$, o uso correto de (2.36) estará garantido desde que sempre tenhamos $\sum_{i=1}^{n} x_i w_{ij} \geq 0$ e $x_i w_{ij} \geq 0$. No entanto, durante o treinamento da rede neural, o *bias* pode assumir qualquer valor em [-∞,+∞]. Para resolver esse problema e considerando as restrições $w_{ij} \geq 0.001$ e $\theta \geq 0$, os pesos e *bias* de (2.9) devem ser transformados usando a função exponencial:

$$s_j = f(\sum_{i=1}^{n} x_i (0.001 + e^{w_{ij}}) + e^{\theta_j}) \tag{2.47}$$

Usando essa transformação, o novo peso $w'_{ij} = 0.001 + e^{w_{ij}}$ sempre será maior que 0.001 e o novo *bias* $\theta'_j = e^{\theta_j}$ sempre será maior que zero.

Essas modificações não vão alterar a essência do algoritmo Backpropagation usado para treinar a rede neural. O algoritmo vai ajustar naturalmente w_{ij} e θ_j entre [-∞,+∞] e as restrições estarão garantidas quando a função de restrição (2.47) for aplicada.

Transparência é uma propriedade dos sistemas difusos que permite a interpretação de suas regras por humanos. Um sistema difuso é considerado transparente somente se todas as regras de sua base de regras são transparentes [29]. Uma regra é dita transparente, quando o grau de disparo da regra for:

$$v_j = \bigcap_{i=1}^{n} \mu_{ij}(x_i) = 1 \tag{2.48}$$

Logo, a saída do sistema será:

$$y = y_j \tag{2.49}$$

onde y_j é o centro da função de pertinência de saída associada à regra. No caso do modelo *Takagi-Sugeno*, y_j é igual ao consequente constante. Isso significa dizer que o efeito de uma única regra pode ser isolado. Em sistemas não transparentes, a saída só tem significado com a interpolação de regras.

Para dar essa transparência ao sistema de regras extraído da rede neural, é realizado um processo de aproximação para todas as funções de pertinência extraídas. |Isso é realizado com a combinação de cinco funções de pertinência (Figura 2.15):

$$\mu(x) = a_1\mu_{pequeno}(x) + a_2\mu_{médio}(x) + a_3\mu_{alto}(x) + a_4\mu_{muito\ pequeno}(x) + a_5\mu_{extremamente\ pequeno}(x) \tag{2.50}$$

onde $x \in [0,1]$ e $[a_1, ..., a_5]$ são parâmetros identificados através de um algoritmo de mínimos quadrados recursivo.

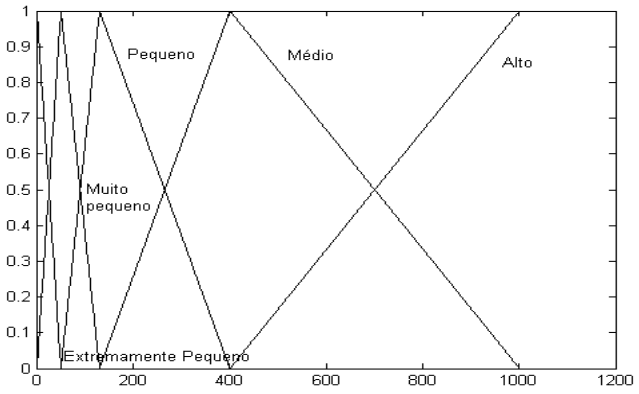


Figura 2.15 - Funções de Pertinência para Sistema Difuso Transparente

Para cada regra, as funções de pertinência aproximadas, calculadas para cada entrada, em (2.50), são então combinadas. Como resultado, um novo sistema de regras com um total de regras transparentes igual a 5^n é formado, sendo n o número de entradas do sistema.

2.4 Algoritmos Genéticos

Algoritmos Genéticos (AGs) são algoritmos matemáticos inspirados nos mecanismos de evolução natural e recombinação genética. A técnica de Algoritmos Genéticos fornece um mecanismo de busca adaptativa que se baseia no princípio Darwiniano de reprodução e sobrevivência dos mais aptos [30].

Os princípios da natureza nos quais os AGs se inspiram são simples. De acordo com a teoria de Charles Darwin, o princípio de seleção privilegia os indivíduos mais aptos com maior longevidade e, portanto, com maior probabilidade de reprodução. Indivíduos com mais descendentes têm mais chance de perpetuarem seus códigos genéticos nas próximas gerações. Tais códigos genéticos constituem a identidade de cada indivíduo e estão representados nos cromossomos.

Estes princípios são imitados na construção de algoritmos computacionais que buscam uma melhor solução para um determinado problema, através da evolução de populações de soluções codificadas através de cromossomos artificiais.

Nos AGs, um cromossomo é uma estrutura de dados que representa uma das possíveis soluções do espaço de busca do problema. Cromossomos são então submetidos a um processo evolucionário que envolve avaliação, seleção, recombinação sexual (crossover) e mutação. Após vários ciclos de evolução a população deverá conter indivíduos mais aptos.

A analogia entre Algoritmos Genéticos e o sistema natural é representada através da tabela 2.4:

Tabela 2.4 – Analogia dos AGs X Natureza

Natureza	AGs
Cromossomo	*Palavra binária, vetor*
Gene	*Característica do problema*
Alelo	*Valor da característica*
Loco	*Posição da palavra, vetor*
Genótipo	*Estrutura*
Fenótipo	*Estrutura submetida ao problema*
Indivíduo	*Solução*
Geração	*Ciclo*

Podemos caracterizar os Algoritmos Genéticos através dos seguintes componentes: o Problema a ser otimizado, a Representação das Soluções de Problema, a Decodificação do Cromossomo, a Avaliação, Seleção, os Operadores Genéticos e a Inicialização da População.

2.4.1 O Problema

Os AGs são particularmente aplicados em problemas complexos de otimização, ou seja, problemas com diversos parâmetros ou características que precisam ser combinadas em busca da melhor solução; problemas com muitas restrições ou condições que não podem ser representadas matematicamente; e problemas com grandes espaços de busca.

Historicamente, os AGs vêm sendo aplicados a problemas de otimização [31], tais como: Otimização de Funções Matemáticas, Otimização Combinatorial, Otimização de Planejamento, Problema do Caixeiro Viajante, Problema de Otimização de Rota de Veículos, Otimização de Layout de Circuitos, Otimização de Distribuição, Otimização em Negócios e Síntese de Circuitos Eletrônicos.

2.4.2 Representação

A representação das possíveis soluções do espaço de busca de um problema define a estrutura do cromossomo a ser manipulado pelo algoritmo. A representação do cromossomo depende do tipo de problema e do que, essencialmente, se deseja manipular geneticamente. Os principais tipos de representação são:

Tabela 2.5 – Representação de AGs

Representação	Problemas
Binária	*Numéricos, Inteiros*
Números Reais	*Numéricos*
Permutação de Símbolos	*Baseados em Ordem*
Símbolos Repetidos	*Grupamento*

A representação binária é simples, fácil de manipular cromossomos através dos operadores genéticos, fácil de ser transformada em inteiro ou real e, ainda, facilita a prova de alguns teoremas. Todavia, a representação por números reais (ponto flutuante) oferece melhor desempenho.

Por exemplo, seja o problema de encontrar o valor máximo da função $f(x) = x^2$, x inteiro [0,63]. Podemos representar as soluções do problema através de um cromossomo de 6 bits:

- C_1: 00 1 0 0 1 representa x=9
- C_2: 00 0 1 0 0 representa x=4

Um binário também pode representar um número real $X_R \in [X_{min}, X_{max}]$, com precisão de p casas decimais. Para isso são necessários k bits, sendo k calculado por:

$$2^k \geq (X_{max} - X_{min}) \times 10p \qquad (2.51)$$

A representação binária, entretanto, nem sempre pode ser empregada; o problema muitas vezes exige um alfabeto de representação com mais símbolos. Qualquer que seja a representação empregada, ela deve ser capaz de representar todo o espaço de busca que se se deseja investigar.

2.4.3 Decodificação

A decodificação do cromossomo consiste basicamente na construção da solução real do problema a partir do cromossomo. O processo de decodificação constrói a solução para que esta seja avaliada pelo problema. A vantagem da representação binária é a fácil transformação para inteiro ou real.

Na transformação para número real, considera-se o intervalo de valores ou comprimento contínuo do domínio (C) dos reais de tal forma que:

$$X_R = X_b \times \frac{C}{2^n - 1} + X_{\min} \qquad (2.52)$$

onde $X_R \in [X_{min}, X_{max}]$, Xb é o inteiro correspondente ao binário, n é o número de bits do cromossomo e C é o comprimento do domínio da variável X, dado por $C = |X_{max} - X_{min}|$.

2.4.4 Avaliação

A avaliação é o elo entre o AG e o mundo externo. A avaliação é feita através de uma função que melhor representa o problema e tem por objetivo fornecer uma medida de aptidão de cada indivíduo na população corrente, que irá dirigir o processo de busca. A função de avaliação é para um AG o que o meio ambiente é para seres humanos. Funções de avaliação são específicas de cada problema.

2.4.5 Seleção

O processo de seleção em algoritmos genéticos seleciona indivíduos para a reprodução. A seleção é baseada na aptidão dos indivíduos: indivíduos mais aptos têm maior probabilidade de serem escolhidos para reprodução. Assim, se f_i é a avaliação do indivíduo *i* na população corrente, a probabilidade p_i do indivíduo *i* ser selecionado é proporcional a:

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_i} \tag{2.53}$$

onde *N* é o número de indivíduos na população.

A seleção em AGs é tipicamente implementada por uma roleta onde cada indivíduo é representado por uma fatia proporcional a sua aptidão relativa. O operador de seleção é um componente essencial de algoritmos genéticos. A literatura identifica cinco principais mecanismos de seleção: proporcional, por torneios, com truncamento, por normalização linear e por normalização exponencial [32].

Um mecanismo de seleção é caracterizado pela pressão seletiva ou intensidade de seleção que o mesmo introduz no algoritmo genético. O termo pressão seletiva é utilizado em diferentes contextos e com significados diferentes na literatura de computação evolucionária. A definição de intensidade de seleção empregada em genética é a variação na aptidão média da população induzida pelo método de seleção [32]. A expressão da intensidade de seleção *I* é dada por:

$$I = \frac{M^* - M}{\sigma} \tag{2.54}$$

onde M é a aptidão média da população, M^* é o valor esperado da aptidão média após a seleção, e σ é o desvio padrão dos valores de aptidão da população antes da seleção.

No caso de seleção proporcional, a probabilidade de um indivíduo ser selecionado é simplesmente proporcional ao seu valor de aptidão, isto é:

$$p_i = \frac{f_i}{NM} \tag{2.55}$$

onde p_i é a probabilidade de seleção de um indivíduo i, f_i é a aptidão do mesmo e N é o tamanho da população. Demonstra-se que a intensidade de seleção é dada por:

$$I = \frac{\sigma}{M} \tag{2.56}$$

Isto é, a pressão seletiva é dada pela razão entre o desvio padrão das aptidões e a média das mesmas.

O método de seleção proporcional apresenta dois problemas: existência de super indivíduos e competição próxima [33]. O primeiro ocorre quando um indivíduo apresenta uma aptidão bem maior que a dos restantes, o que determinará uma convergência prematura do algoritmo. O segundo problema ocorre quando indivíduos apresentam aptidões semelhantes, mas não idênticas; neste caso, a intensidade de seleção pode ser bem menor do que a desejável.

No método de seleção por torneios, um grupo de t indivíduos é aleatoriamente escolhido, e o indivíduo de melhor aptidão é selecionado. A intensidade de seleção é dada neste caso pela solução da seguinte equação integral [32]:

$$I = \int_{-\infty}^{\infty} t.x.\frac{1}{\sqrt{2\pi}}.e^{-\frac{x^2}{2}}\left(\int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}}.e^{\frac{y^2}{2}}dy\right)^{t-1}dx \tag{2.57}$$

As variáveis de integração *x* e *y* representam os valores de aptidão da população.

Assume-se uma distribuição Gaussiana de valores de aptidão por indivíduos. A partir da solução numérica da equação acima, observa-se que a pressão seletiva aumenta na medida em que o número de indivíduos envolvidos no torneio, *t*, aumenta.

No mecanismo de seleção por truncamento, dado um limiar *T*, apenas os *T* melhores indivíduos podem ser selecionados. Cada um desses indivíduos apresenta a mesma probabilidade de seleção. Demonstra-se em [32], que a intensidade de seleção é dada por:

$$I = \frac{1}{T} . \frac{1}{\sqrt{2\pi}} . e^{\frac{f_c^2}{2}} \tag{2.58}$$

onde f_c é o valor da menor aptidão entre os *T* melhores indivíduos. Traçando-se o gráfico da intensidade de seleção em função do limiar *T*, observa-se que a pressão seletiva diminui à medida que *T* aumenta.

No método de seleção por normalização linear, os indivíduos são inicialmente ordenados de acordo com sua aptidão. A seguir, estes valores de aptidão são alterados de acordo com a posição relativa de cada indivíduo. Ao melhor indivíduo é assinalada uma aptidão de valor máximo e ao pior indivíduo uma aptidão mínima. Estes dois valores são determinados pelo usuário, mas a forma original deste método prevê que as condições *max*= 2 – *min* e *mín* ≥ 0 devam ser atendidas. Os demais indivíduos têm valores de aptidão linearmente distribuídos entre *min* e *max*, de acordo com sua posição relativa na ordenação (*i*=1 corresponde ao pior elemento).

$$A_i \equiv \min + \frac{(\max - \min)}{n-1}(i-1) \tag{2.59}$$

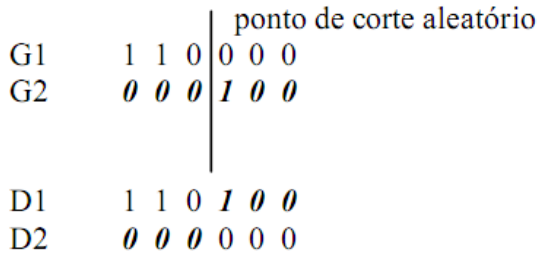
O método de seleção por normalização exponencial diferencia-se da normalização linear pelo fato das probabilidades de seleção de cada indivíduo seguirem uma função exponencial. Esta probabilidade é dada por:

$$p_i = \frac{c-1}{c^N - 1} c^{N-i}; \qquad i \in \{1, ..., N\} \qquad (2.60)$$

onde *c* determina a grau de 'exponencialidade' da função, podendo variar de 0 até 1. Quanto mais próximo de 1, menor a 'exponencialidade'.

2.4.6 Operadores Genéticos

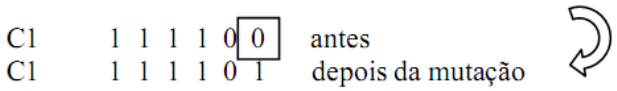
Indivíduos selecionados (e reproduzidos na população seguinte) são recombinados através do operador de crossover (com uma probabilidade p_c). O operador de crossover é considerado como a solução dos AGs. Pares de genitores são escolhidos aleatoriamente da população, baseado na aptidão, e novos indivíduos são criados a partir da troca do material genético. Os descendentes serão diferentes de seus pais, mas com características genéticas de ambos os genitores. Por exemplo:



Na sua forma mais simples, o crossover de um ponto de corte (one-point crossover) corta os dois genitores em uma posição aleatoriamente escolhida, criando dois possíveis descendentes: *D1* é um cromossomo mais apto que seus genitores, todavia *D2* é um indivíduo medíocre (baixa avaliação em $f(x) = x^2$).

Os cromossomos criados a partir do operador de crossover são então submetidos à operação de mutação (com uma probabilidade p_m). Mutação é um operador exploratório que tem por objetivo aumentar a diversidade na população.

O operador de mutação troca o conteúdo de uma posição do cromossomo (alelo, símbolo, valor do gene), com uma determinada probabilidade, em geral baixa (<1%).



Existem várias outras formas de se efetuar mutação. Por exemplo, o operador genético denominado inversão troca de posição dois genes aleatoriamente escolhidos. A importância deste operador é, no entanto, restrita a problemas baseados em ordem [30] [31].

2.4.7 Inicialização da População

A inicialização da população determina o processo de criação dos indivíduos para o primeiro ciclo do algoritmo. Tipicamente, a população inicial é formada a partir de indivíduos aleatoriamente criados. Populações iniciais aleatórias podem ser semeadas com bons cromossomos para uma evolução mais rápida, quando se conhece, a priori, o valor de boas “sementes”. Uma técnica eficiente em AGs para se encontrar boas soluções em um problema consiste em executar evoluções (rodadas) sucessivas, semeando-se a população inicial da evolução seguinte com as melhores soluções encontradas na anterior.

Um algoritmo genético pode ser descrito como um processo contínuo que repete ciclos de evolução controlados por um critério de parada, conforme apresentado pela Figura 2.16.

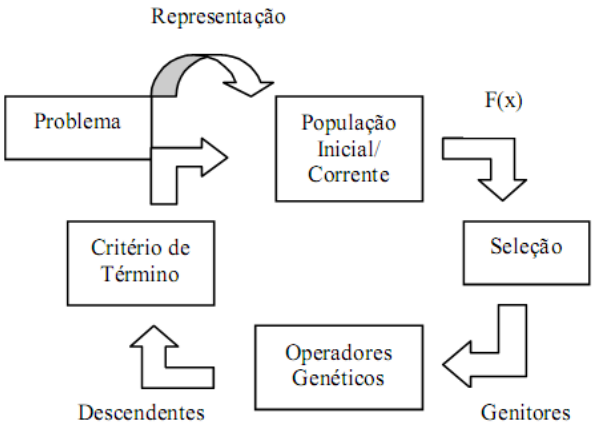


Figura 2.16 – Esquema Geral de um Algoritmo Genético

CAPÍTULO 3

PREVISÃO DE CARGA ELÉTRICA

Deter o conhecimento da demanda de carga elétrica futura de forma precisa é importante para diversas atividades de planejamento e operação em sistemas de potência. Previsões de carga são feitas para períodos de tempo de duração variável. Por exemplo, previsão a curto prazo (varia de poucas horas até uma semana a frente) é realizada para a alocação otimizada da geração de energia, funções de escalonamento, avaliações de trocas na rede e análise da segurança do sistema.

A previsão a médio prazo, que varia entre um dia e um ano, é usada para planejamento de manutenção, escalonamento de combustível e gerenciamento dos reservatórios hídricos. O horizonte de tempo para previsão a longo prazo vai de um até dez anos, tempo necessário para planejamento de sistemas de transmissão e distribuição e unidades (usinas) de geração de energia [4].

As características a seguir tornam a previsão de carga uma tarefa desafiadora [34]:

- A carga atual do sistema elétrico em qualquer ponto no tempo depende de uma série de fatores e eles nem sempre podem ser previstos com confiabilidade.
- A demanda da carga muda ciclicamente em resposta a variações sazonais.
- Em geral, a demanda de carga total do sistema é incrementada continuamente devido ao crescimento industrial, aumento no consumo elétrico e outros fatores relacionados.
- A carga segue um padrão definido durante os dias da semana, mas mudam consideravelmente em eventos como finais de semana e feriados.

A maioria das técnicas convencionais de previsão de carga, especialmente para previsão a curto prazo, podem ser categorizadas a partir de duas vertentes [35]: métodos de regressão e métodos de previsão de séries temporais. Estes modelos de previsão tradicionais tratam o problema como uma instância especial de estimativa de parâmetro, onde o modelo estimado é considerado como o previsor.

Métodos de regressão necessitam identificar variáveis relevantes com correlação forte com a carga elétrica, como a temperatura, umidade, vento, dentre outros. Métodos de previsão de carga baseados em séries temporais trabalham examinando dados históricos, extraindo características de dados essenciais e, efetivamente projetando estas características no futuro.

Os requisitos essenciais para um bom modelo de previsão são precisão e confiabilidade, independente da estação do ano e das condições meteorológicas [36]. Os modelos estatísticos de previsão de carga apresentam algumas limitações que dificultam o preenchimento dos dois requisitos citados acima. O primeiro grupo é ineficiente devido a sua dependência do relacionamento funcional entre a carga e as variáveis meteorológicas e o segundo grupo é numericamente instável [37].

Este capítulo apresenta uma visão geral sobre o problema de previsão de cargas elétricas, sendo que será apresentado com maior destaque os métodos tradicionais para previsão de carga assim com o estado da arte em inteligência computacional aplicada ao problema de previsão.

3.1 CARGA ELÉTRICA

As cargas elétricas são compostas por milhares de aparelhos e dispositivos que, quando consideradas em conjunto, apresentam tendências que podem ser estatisticamente previstas. Tais tendências são influenciadas por fatores que interferem na forma de modelar a carga. Esses fatores podem ser divididos em: condições meteorológicas, fatores econômicos ou demográficos e fatores sazonais. Antes de se pensar em previsão de carga, é necessária uma análise desses fatores para verificar como influenciam o comportamento dessa carga.

Um grande número de métodos tem sido proposto para estabelecer previsões com certo grau de precisão. Os métodos de previsão de carga necessitam, basicamente, de duas condições fundamentais, informações sobre o passado quantificadas em forma de dados e, assumir que o comportamento ocorrido no passado, de certa forma, se repetirá no futuro [38].

A determinação do padrão de comportamento da carga é utilizada na escolha do modelo adequado, visando descrever a evolução da carga no tempo e através dele pode-se dividir a série temporal (evolução da carga no tempo) em duas classes [39][40]:

1. Estacionária: Uma série temporal onde os dados da carga têm flutuações em torno de uma média constante no tempo. Na Figura 3.1, tanto o item (a) como o item (b), possuem dados que mantém uma média constante no tempo, o que indica que a série temporal é estacionária.

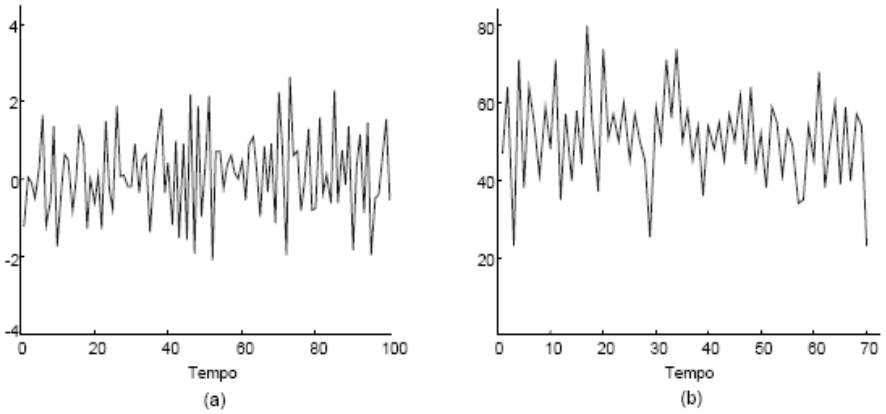


Figura 3.1 - Série Temporal Estacionária

2. Não Estacionária: Uma série temporal onde os dados da carga variam com o tempo, ou seja, se o padrão básico da série histórica dos dados apresenta padrões tendenciosos, cíclicos, sazonais ou aleatórios. A Figura 3.2 apresenta duas séries temporais não estacionárias, em (a) tem-se um padrão sazonal sobreposto semelhante a um padrão horizontal e em (b) tem-se um padrão sazonal com aparente tendência para cima. Trata-se, portanto, de um comportamento que não se observa uma tendência estatística.

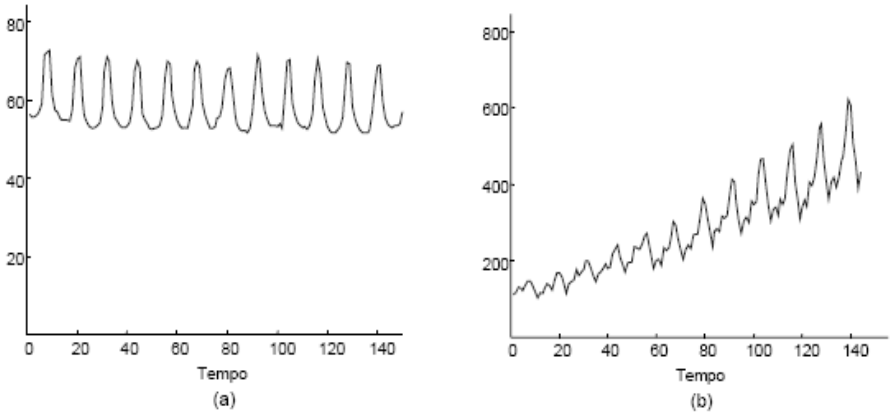


Figura 3.2 - Série Temporal não-Estacionária

Após selecionar a classe de modelos que melhor descreve o comportamento, é preciso escolher um modelo adequado desta classe que melhor descreva a série temporal com o propósito de realizar previsões para valores futuros desta.

A carga de um sistema elétrico é composta por diferentes unidades de consumo. Boa parte da eletricidade é consumida em atividades industriais. Outra parte é usada pelo usuário doméstico, por meio de aquecimento de água, iluminação e o uso de uma gama de aparelhos elétricos ou eletrônicos. Além disso, vários serviços públicos também são responsáveis por um significativo consumo de energia, como iluminação pública, trilhos de trens ou metrôs, dentre outros [41].

O consumo elétrico no setor industrial depende muito do nível da produção e, normalmente é estimável e previsível. Todavia, fatores inesperados podem afetar sensivelmente o nível da carga, como máquinas quebradas ou até mesmo uma greve.

No caso do usuário doméstico é difícil se definir um padrão de consumo de energia, pois cada pessoa se comporta da sua própria maneira. Fatores sociais e comportamentais podem afetar o nível da carga doméstica, como grandes eventos, feriados, até mesmo programas de TV [4]. O clima também é um fator determinante, pois pode aumentar ou diminuir o uso de chuveiros elétricos e condicionadores de ar, por exemplo.

Como a maior parte do consumo de energia elétrica é destinada ao usuário particular e outros pequenos consumidores, uma abordagem bem comum para se prever a carga total a ser usada é se concentrar na carga agregada do aparelho como um todo. Sendo assim, alguns fatores importantes precisam ser considerados [4][42]. Condições meteorológicas podem causar uma variação grande na carga agregada. Além da temperatura, a velocidade do vento, cobertura de nuvens e umidade podem influenciar a medida da carga total [43]. Fatores demográficos e econômicos, como a taxa de crescimento do país para os próximos anos, também têm um papel importante na evolução da demanda elétrica.

Mas, do ponto de vista da previsão da carga, o que influencia mesmo são fatores sazonais, que acabam causando comportamentos cíclicos (padrões diários e semanais de comportamento), bem como a ocorrência de feriados. Outros fatores que possam vir a causar dificuldades na previsão da carga são classificados como

aleatórios [41]. A decisão de se agregar todas as unidades consumidoras em um único valor agregado da carga significa que a previsão vai recair em grande parte nos comportamento passado da carga.

3.2 PREVISÃO DE CARGA

A previsão de carga temporal consiste na determinação de procedimentos efetuados para o fornecimento e consumo da carga, realizados em horas, dias, meses ou anos. Pode-se classificar a previsão de carga de acordo com o período em que a carga é prevista. Segundo [41][44][45][46] e [47], existem quatro tipos de previsão de carga:

1. **Previsão a longo prazo:** realizada principalmente na operação de planejamento de sistemas de potência. Este modelo faz previsão envolvendo um período de 10 a 20 anos. É realizada para planejamento a longo prazo. Não se consideram os dados de carga elétrica diária ou semanal, somente a tendência do consumo de demanda de carga a longo prazo. Podem ser aplicados também na provisão de equipamentos elétricos, preços de eletricidade, preços de fontes alternativas de energia, entre outros.
2. **Previsão a médio prazo:** feita para o planejamento de suprimento de combustível, programas de manutenção e planejamento de produção a médio prazo. Ela abrange o período de poucas semanas ou até mesmo alguns anos.
3. **Previsão a curto prazo:** ocorre no planejamento e operações econômicas e de segurança diária dos sistemas de potência. A previsão é feita no intervalo de tempo de meia hora a poucas horas, ou até mesmo de um dia a uma semana. A previsão de carga a curto prazo estima a carga para cada hora do dia, o pico diário de carga ou a geração de energia diária ou semanal.
4. **Previsão a curtíssimo prazo:** são usadas para o planejamento de produção e controle. A previsão ocorre entre poucos segundos e algo menos do que 15 minutos.

3.3 MÉTODOS DE PREVISÃO DE CARGA

A carga do sistema a ser prevista é um processo aleatório não-estacionário composto de milhares de pontos, o que faz com que a possibilidade de diferentes abordagens para se fazer essa previsão seja imensa. Normalmente o que se faz é focalizar o problema de um ponto de vista macro e tentar modelar a carga futura como reflexo de seu comportamento anterior [41]. Ainda assim, as possibilidades de diferentes soluções são muitas. Devido à natureza da carga, a única forma objetiva para se examinar essas abordagens é por meio de evidências empíricas.

Podemos classificar os diversos métodos de previsão de carga de curto prazo existentes na literatura pelo tipo de modelo de carga usado. São considerados dois modelos básicos: *Pico de Carga* e *Formato da Carga* [4].

1. Modelos de Pico de Carga (Peak Load)

Nesses métodos, o tempo cronológico não tem importância. A carga diária ou semanal normalmente é modelada em função do clima, na forma:

P (pico de carga) = *B* (carga de base) + *F(W)* (componente dependente do clima)

Onde a carga de base *B* é uma média não influenciada pelo clima à qual é adicionado um componente dependente do clima *F(W)*. As variáveis do clima *W* podem incluir a temperatura na hora do pico de carga ou uma combinação de temperaturas preditas e passadas. Ainda é possível se levar em consideração fatores como umidade, intensidade da luz, velocidade do vento e índice pluviométrico. A função *F*(.) é calculada empiricamente e pode ser linear ou não.

Os pontos fortes dos métodos que consideram pico de carga são a sua simplicidade, sendo que se estruturam bem com a pequena quantidade de dados necessária para inicializar e atualizar o sistema. Os parâmetros do modelo são estimados por regressão linear ou não linear. Todavia, estes métodos não definem a hora exata em que o pico de carga ocorre e muito menos guardam informações como o formato da curva da carga. São modelos estáticos, ou seja, não é possível prever a correlação entre períodos distintos [48][49][50][51] e [52].

2. Modelos de Formato de Carga (Load Shape)

Nesse caso, a carga é identificada não apenas como uma função de tempo (horas) do dia, mas também de seus comportamentos mais recentes. Por essa razão, a previsão para determinada hora precisa de previsões anteriores.

Estes métodos podem ser divididos, basicamente em duas categorias: estatísticos e inteligentes.

3.3.1 Métodos Estatísticos

Os métodos estatísticos são baseados em técnicas estatísticas onde a formulação da modelagem matemática da carga é de extrema importância. Dentre os métodos existentes pode-se citar *Regressão Linear* ou *Múltipla*, *Alisamento Exponencial*, *Filtro de Kalman*, *Espaço de estado* e *Série Temporal de Box-Jenkins* [40].

3.3.1.1 Regressão Linear ou Múltipla

Este modelo descreve o comportamento estocástico de padrões de carga horária em um sistema de potência. Ele assume que a carga, em uma determinada hora, pode ser estimada a partir da relação da carga padrão e algumas variáveis explanatórias, que podem ser, por exemplo, variáveis climáticas, como tipo do dia, classes de clientes, entre outros. Segundo [53][46][54][41], o método pode ser descrito por:

$$y(t) = a_o + \sum_{i=1}^{n} a_i q_i(t) + a(t) \tag{3.1}$$

onde, *y(t)* é a carga elétrica; a_0, a_1, ..., a_n são os coeficientes de regressão; q_1, ..., q_n são variáveis explanatórias e *a(t)* é a variável aleatória com média zero e variância constante.

Os coeficientes de regressão são encontrados através do método dos mínimos quadrados. São realizados testes estatísticos para determinar a significância desses coeficientes de regressão. Este é um método de implementação bem simples.

3.3.1.2 Alisamento Exponencial

No alisamento exponencial, considera-se que cada elemento de uma série consiste de uma constante mais uma componente de erro:

$$y(t) = \varphi(t)^T f(t) + \varepsilon(t) \quad (3.2)$$

onde *y(t)* é a carga elétrica, *φ(t)* é o vetor de coeficiente, *f(t)* é o vetor da função de ajuste para o processo e *ε(t)* é o ruído branco.

A modelagem da carga *y*(*t*), para o tempo *t*, utiliza a função de ajuste [53]:

$$f(t) = Lf(t-1) \quad (3.3)$$

onde *L* é a matriz de transição.

3.3.1.3 Espaço de Estado e Filtro de Kalman

Neste método a carga é modelada como uma variável de estado usando a formulação de espaço de estado. O modelo de espaço de estado para carga no tempo *t* é descrito da seguinte forma [53][55][41]:

Equação do espaço de estado:

$$y(t+1) = A(t)\, y(t) + w(t) \quad (3.4)$$

Equação de medida:

$$z(t) = B(t)\, y(t) + v(t) \quad (3.5)$$

onde *y(t)* é a carga elétrica, *A(t)* é a matriz de transição, *w(t)* é o ruído branco com matriz de covariância *Q(t)*, *z(t)* é o vetor de medição no tempo *t*, *B(t)* é a matriz que correlaciona *y(t)* e *z(t)*, e *v(t)* é o erro medido (ruído com matriz de covariância *S(t))*.

A implementação do filtro de Kalman segue os seguintes passos:

1. Encontrar a estimativa *y(t/t-1)* e o erro da matriz de covariância *p(t/t-1)*;
2. Calcular o ganho de Kalman:

$$k(t) = p(t/t-1)B(t)^T[B(t)p(t/t-1)B(t)^T - S(t)^{-1}] \quad (3.6)$$

3. Calcular a estimativa atualizada do erro da matriz de covariância:

$$p(t/t) = [1 - k(t)B(t)]p(t/t-1) \quad (3.7)$$

onde, *K(t)* é o fator de combinação.

4. Projetar, à frente, a estimativa *y(t+1/t)* e o erro da matriz de covariância *p(t+1/t)* associada como segue:

$$y(t+1/t) = A(t)y(t/t) \quad (3.8)$$

$$p(t+1/t) = A(t)p(t/t)A(t)^T + Q(t) \quad (3.9)$$

5. Retornar ao passo 2 mudando para o próximo passo de tempo.

A previsão de espaço de estado é baseada no modelo de Kalman e a sua maior dificuldade está no processo de identificação.

3.3.1.4 Série Temporal de Box & Jenkins

É a classe mais popular de modelos de previsão dinâmica. O princípio básico é que a série temporal da carga pode ser transformada em uma série temporal estacionária (invariante no tempo). A carga *y(t)* é modelada como sendo a saída de um filtro linear que tem como entrada, séries aleatórias *a(t)*, denominadas ruído branco, com média zero e variância $\sigma_a^2(t)$.

A classe de modelos que melhor apresenta as características da série temporal estacionária é a classe de modelos ARMA. Dentre os mesmos, podem-se citar o modelo Ruído Branco, o modelo *Auto-Regressivo* (Auto Regressive-AR), o modelo *Médias Móveis* (Moving Average-MA) e finalmente o modelo *Auto-Regressivo de Médias Móveis* (Auto Regressive Moving Average-ARMA) [39][53][41].

a) Modelo Ruído Branco

O modelo foi definido por [39], como sendo:

$$y(t) = \theta_0 + a(t) \tag{3.10}$$

onde, *y(t)* é a carga elétrica, *a(t)* são séries aleatórias e θ_0 é o termo constante.

b) Modelo Auto-Regressivo *(AR(p))*

O modelo é definido por [39][53]:

$$y(t) = \theta_0 + \phi_1 y(t-1) + \phi_2 y(t-2) + \ldots + \phi_p y(t-p) + a(t) \tag{3.11}$$

onde, φ_1, φ_2 ..., φ_p são os coeficientes auto-regressivos e *p* é a ordem do modelo.

c) Modelo de Médias Móveis *(MA(q))*

O modelo é definido por [39][53]:

$$y(t) = \theta_0 + a(t) - \theta_1 a(t-1) + \theta_2 a(t-2) - \ldots - \theta_q a(t-r) \tag{3.12}$$

onde, θ_1, θ_1, ..., θ_q são os coeficientes de média móvel e *r* é a ordem do modelo.

d) Modelo Auto-Regressivo de Médias Móveis *(ARMA(p,q))*

O modelo é definido por [39][53][41]:

$$y(t) = \theta_0 + \phi_1 y(t-1) + \phi_2 y(t-2) + \ldots + \phi_p y(t-p) + a(t) - \theta_1 a(t-1) - \theta_2 a(t-2) - \ldots - \theta_q a(t-r) \quad (3.13)$$

Este modelo combina as duas técnicas anteriores.

Os modelos ARIMA (*p,d,r*) são as classes mais comuns de modelos que melhor representam uma série temporal não-estacionária.

As séries temporais não estacionárias passam por uma transformação denominada *differencing*, que possui a finalidade de estacionarizá-las para então depois serem utilizadas [39][53][41]. Dentre os modelos de série temporal não estacionárias, tem-se o modelo *Walk Aleatório* e os modelos *Auto Regressive Integrated Moving Average (ARIMA)*.

- **Modelo Walk Aleatório**

O modelo Walk Aleatório é definido por [39]:

$$y(t) = y(t-1) + \theta_0 + a(t) \quad (3.14)$$

Este modelo é caracterizado pelo fato de que as primeiras diferenças regulares formam um modelo de Ruído Branco. Nesse caso, a previsão para todos os períodos de tempo futuro é exatamente o último valor observado da série temporal.

- **Modelo Auto-Regressivo Integrado de Médias Móveis *(ARIMA)***

O modelo é caracterizado pelas primeiras diferenças regulares que formam os modelos ARMA, tais como AR(1), AR(2), MA(1) e MA(2) [39].

O primeiro destes é conhecido como modelo *Auto-Regressivo Integrado* – ARI (1,1) sendo definido como [39]:

$$y(t) = \theta_0 + (1+\phi_1) y(t-1) - \phi_1 y(t-2) + a(t) \quad (3.15)$$

As primeiras diferenças regulares formam um modelo AR(1). O modelo ARI(2,1) é encontrado de forma similar e as primeiras diferenças regulares forma um modelo AR(2).

O modelo IMA(1,1) é definido como [39]:

$$y(t) = \theta_0 + y(t-1) - \theta_1 a(t-1) + a(t) \tag{3.16}$$

onde as primeiras diferenças regulares formam um modelo MA(1). O modelo IMA(1,2) é encontrado, de forma similar, sendo que as primeiras diferenças regulares formam um modelo MA(2).

O último modelo a ser considerado é o modelo ARIMA(1,1,1) sendo que as primeiras diferenças regulares deste modelo formam um modelo ARMA(1,1). Este modelo é descrito como [39]:

$$y(t) = \theta_0 + (1+\phi_1)\, y(t-1) - \phi_1 y(t-2) + a(t) - \theta_1 a(t-1) \tag{3.16}$$

De uma forma geral, os modelos considerados são todos casos particulares dos modelos ARIMA (*p*,1,*r*), cujas primeiras diferenças regulares formam modelos ARMA(*p*,*r*).

Os modelos ARIMA(*p*,*d*,*r*) são aqueles cujas *d*-ésimas diferenças regulares formam modelos ARMA(*p*,*r*). É importante lembrar que na prática, modelos ARIMA(*p*,*d*,*r*) com $d \geq 2$ não ocorrem.

3.3.2 Métodos Baseados em Inteligência Computacional

Atualmente, vêm-se utilizando metodologias alternativas para realizar a previsão de cargas elétricas, sendo estas baseadas no conceito de Inteligência Computacional.

No contexto científico-tecnológico que se apresenta atualmente, onde ferramentas que auxiliem na tomada de decisão são essenciais para que os resultados apresentado sejam ótimos e as perdas mínimas, a previsão de cargas através de técnicas de inteligência computacional tem se mostrado uma linha de pesquisa muito interessante e é imenso o número de trabalhos acadêmicos sendo apresentados na literatura.

Entre as técnicas de inteligência computacional que vêm sendo aplicadas para o problema de previsão de cargas destacam-se as Redes Neurais artificiais, os Sistemas Baseados em Lógica Difusa, os Sistemas Híbridos, que combinam mais de uma técnica inteligente e os Algoritmos Genéticos, usados para aperfeiçoar um destes modelos citados.

Desde 1990 as redes neurais artificiais vêm sendo utilizadas como ferramenta na resolução do problema de previsão de cargas elétricas [46][45][56][54][57][58][59].

Em [64] uma rede MLP é utilizada para previsão de carga, tendo dados climáticos (temperatura) como entrada para RNA. No modelo apresentado, a rede neural aprende a relação entre temperatura e carga. Para calcular a carga prevista, a RNA interpola os dados carga/temperatura no conjunto de treinamento e subdivide o problema em três partes, implementando três sub-redes neurais especializadas, uma para prever o pico da carga, uma para prever a carga total do dia e outra para o perfil horário da carga durante o dia. O modelo é para o horizonte de 24 horas adiante.

Em [65] foi realizado um estudo de previsão de carga elétrica a curto prazo com a utilização de uma rede neural baseada na arquitetura ART (*Adaprative Resonanse Theory*), denominada Rede Neural ART&ARTMAP. Este tipo de rede ART possui estabilidade e plasticidade e permite a implementação do treinamento de modo contínuo. Por tratar separadamente os dados analógicos e binários, o modelo reduz as imprecisões dos resultados de previsão o que melhora a qualidade na obtenção dos resultados, levando a um menor tempo de processamento e melhor precisão.

Para encontrar padrões de carga, em [66] foi concebida uma rede neural baseada nos Mapas Auto-Organizáveis de *Kohonen* para identificar os dias com padrões de cargas similares, agrupando-os em classes. Uma característica especial dos mapas auto-organizáveis é que eles são capazes de identificar um novo tipo de padrão de carga antes mesmo que os operadores possam reconhecer o padrão do novo dia.

Pode-se também construir um sistema de previsão utilizando como base o conhecimento na forma de regras difusas. Em [61], um sistema difuso é proposto para previsão de carga a curto prazo, calculando a carga máxima e mínima diária. Estas cargas são divididas em duas partes, a parte básica e a parte retificada. São

levadas em consideração informações de clima, índice sazonal e informações sobre feriados.

Um sistema especialista difuso para previsão de carga a curto prazo foi desenvolvido em [67]. Ele usa a teoria dos conjuntos difusos para modelar imprecisões no modelo de carga/temperatura e para prever a carga elétrica futura utilizando o que chama de regras heurísticas de operador.

Em [68], foi avaliada a aplicação de modelos difusos do tipo *Takagi-Sugeno* ao problema da previsão de carga a curto prazo. Para a identificação dos modelos com antecedentes lineares foi aplicada a combinação de estimativa de grupo (cluster) com o método dos mínimos quadrados. Para modelar os antecedentes não-lineares foi usado o conjunto transformado difuso de uma MLP.

Modelos híbridos também vêm sendo utilizados para o problema de previsão. Em [70] um sistema neuro-difuso chamado *Fuzzy Neural Network* – FNN é apresentado e aplicado ao problema da previsão de carga a curto prazo. O FNN inicialmente cria uma base de regras a partir dos dados históricos de carga. Os parâmetros da base de regras são então refinados através do processo de treinamento de forma que a saída do FNN adequadamente é combinada com os dados históricos disponíveis. Uma vez treinado, o FNN pode ser usado para prever cargas.

Foi apresentado em [71] um previsor neuro-difuso de carga a curto prazo. No modelo, primeiramente é usada uma rede neural MLP capaz de prever precisamente a demanda de carga na hora seguinte em dias comuns debaixo de condições climáticas. As saídas da rede neural são então refinadas usando-se um sistema difuso para dias especiais ou onde ocorram de mudanças abruptas no clima. Desta forma chega-se a um sistema previsor mais robusto e confiável.

Em [72], os autores apresentam a aplicação de um sistema neuro-difuso TFRENN (*Transparent Fuzzy Rule Extraction from Neural Network*) para o problema da previsão de carga a curto prazo. O modelo é utilizado para realizar a extração do conhecimento de uma rede neural treinada para previsão de cargas e mapear este conhecimento um sistema de inferência difuso matematicamente equivalente. O mapeamento realizado torna explícito o conhecimento capturado durante o treinamento da rede neural e o transforma em regras difusas transparentes que possuem uma representação linguística mais clara e interpretável por parte dos operadores do sistema.

Algoritmos genéticos também vêm sendo utilizados para o problema de previsão sendo utilizados usados como ferramenta auxiliar para otimizar processos ou etapas em outros métodos. Em [73], é apresentado um sistema híbrido que mescla técnicas de redes neurais e algoritmos genéticos. A rede neural artificial utilizada no modelo é uma Rede de Base Radial (RBF), que demonstra facilidade de ajuste e boa velocidade de treinamento, propriedades relevantes em um sistema de previsão voltado ao uso associado a sistemas SCADA (*Supervisory, Control and Data Acquisition*). O algoritmo genético complementa a capacidade de modelagem inerente ao funcionamento da RNA, definindo a camada de entrada e os parâmetros internos da rede.

O trabalho de *Srinivasan* [74] usa um algoritmo genético na otimização dos pesos de uma rede neural MLP para o problema da previsão de cargas a curto prazo.

Outros exemplos são os trabalhos de *El-Sharkawi* [75] e *Dash* [76] na gerência do aprendizado em rede do tipo *Kohonen* e para otimizar a quantidade de regras da camada de regras de uma rede neuro-difusa, respectivamente.

CAPÍTULO 4

PREVISÃO DE CARGA ATRAVÉS DE INTELIGÊNCIA COMPUTACIONAL

A previsão de carga de curto prazo, que pode variar de minutos a dias, vem se tornando a cada dia mais importante desde a criação dos mercados de energias. Alguns países chegaram a desregulamentar seus sistemas elétricos e a energia elétrica se tornou uma *commodity* vendida a preço de mercado. Como a previsão de carga tem uma participação importante na composição do preço, ela se tornou de vital importância para a indústria.Se o erro da previsão for pequeno, pode-se melhorar a segurança do sistema e diminuir os custos de geração, postergando-se investimentos na área.

Os modelos matemáticos e estatísticos de previsão já foram exaustivamente aplicados ao problema da previsão de carga, entretanto, há algum tempo os modelos baseados em inteligência computacional vêm sendo usados com bastante sucesso e resultados satisfatórios.

Estes métodos apresentam uma resposta melhor a não-linearidades grandes e rápidas que podem aparecer nas cargas além de não dependerem de modelos matemáticos complexos. Nesta categoria, podemos citar como mecanismos que conseguem prever carga com eficácia as redes neurais e os sistemas difusos, bem como os sistemas inteligentes híbridos, que associam duas ou mais destas metodologias, como é o caso dos sistemas neuro-difusos e neuro-evolutivos.

Neste capítulo apresentaremos os sistemas de previsão de carga de curto prazo (uma hora adiante) desenvolvidos com base na Inteligência Computacional, mas especificamente baseados em uma Rede Neural Perceptron Multicamadas, no ANFIS e no TFRENN.

Os sistemas foram desenvolvidos com o objetivo de mostrar a eficiência das Técnicas de Inteligência Computacional para o Problema de Previsão de Carga assim como para comparação de desempenho com o sistema previsor proposto, desenvolvido a partir de um sistema híbrido e baseado em uma rede neural autoassociativas e algoritmos genéticos.

4.1 OBTENÇÃO E TRATAMENTO DOS DADOS

A qualidade da previsão de carga através das Técnicas de Inteligência Computacional depende da escolha das variáveis mais significativas para a previsão

e da disponibilidade de um histórico dessas variáveis, a fim de que o comportamento do sistema possa ser aprendido através de um processo de treinamento.

Assim, é importante que sejam determinadas quais grandezas, além da carga, podem afetar o comportamento da mesma. Além da determinação das variáveis, outro passo importante diz respeito à análise qualitativa destes dados para que, caso seja necessário, um processo de tratamento nos mesmos seja realizado.

Sabe-se que o consumo de energia depende diretamente dos hábitos dos consumidores, logo qualquer fator que afete as indústrias, residências, dentre outros, afetará também a curva de carga. Esses fatores são denominados de variáveis exógenas e entre eles citam-se fatores sociais, econômicos e climáticos.

No caso da previsão a curto prazo, a variável temperatura é um dos fatores com maior efeito sobre a curva de carga. Sazonalidades como o dia da semana e as estações do ano também afetam a curva de carga. Fatores aleatórios também podem afetar a curva de carga e são considerados problemáticos para o problema de previsão. Como exemplos de fatores aleatórios têm-se os jogos de copa, finais de novelas e reality shows.

Para o desenvolvimento dos sistemas previsores baseados em Inteligência Computacional deste trabalho, foram utilizados dados de carga de uma concessionária de energia relativos aos anos de 2005, 2006 e 2007. Devido à dificuldade para obtenção de outros dados que de alguma forma poderiam influenciar o comportamento da carga, optou-se pelo desenvolvimento dos sistemas apenas baseados nos valores de carga.

Um processo de tratamento foi realizado nos dados para identificar dados faltosos.Para treinamento dos sistemas previsores foram utilizados os dados de 2007, pois os mesmos não apresentavam nenhum problema. Dados de 2006 e 2005 foram utilizados para validação. O conjunto de treinamento usado consiste de 5096 padrões e o de validação de 2000 padrões.

Os sistemas previsores foram desenvolvidos para previsão de uma hora adiante. Após a realização de alguns testes (considerando todas as técnicas de inteligência empregadas para o desenvolvimento dos sistemas previsores), duas entradas foram estabelecidas como sendo suficientes para os sistemas, sendo estas os valores de carga das duas horas antecedentes (*t e t – 1*) ao valor desejado de previsão (*t + 1*). A saída do sistema previsor é um número real contendo o valor previsto da carga. Os dados de entrada passaram por um processo de normalização entre os valores [-1 1] através de:

$$x_{ij\,norm} = 2\frac{(x_{ij} - x_{\min})}{(x_{\max} - x_{\min})} - 1 \qquad (4.1)$$

onde x_{min} e x_{max}, correspondem aos valores máximos e mínimos da variável a normalizar. Após o processo de normalização individual das variáveis, estas foram concatenadas de forma a constituírem a matriz global de entrada dos sistemas.

4.2 AMBIENTE COMPUTACIONAL

Dentre as ferramentas computacionais disponíveis no mercado para implementação de sistemas de inteligência computacional, optou-se por fazer uso do MATLAB (release 14) e os testes foram realizados em computadores com processador Core2Duo com 4 GB de memória RAM.

4.3 PREVISÃO DE CARGA ATRAVÉS DE UMA REDE NEURAL MLP

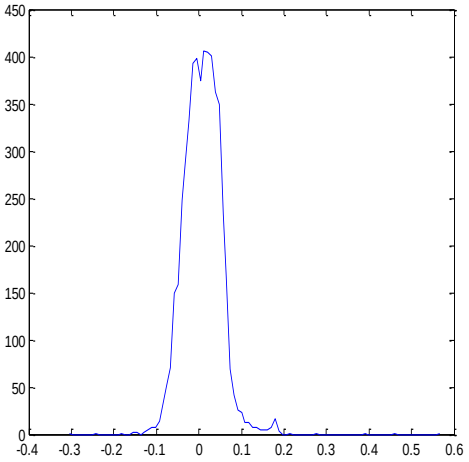
O primeiro modelo testado neste trabalho para realizar previsão de carga a curto prazo, foi uma Rede Neural Perceptron Multicamadas (MLP). O desenvolvimento dessa rede neural foi feito no toolbox de redes neurais do Matlab, utilizando-se para os treinamentos a função "*trainlm*" que é baseada no algoritmo de *Levenberg-Marquardt*. Diversas configurações foram testadas, variando-se o número de neurônios na camada escondida. O melhor resultado, considerando dados de treino e validação, foi obtido para a rede com 10 neurônios na camada escondida, sendo que a função sigmoidal foi utilizada para todos os neurônios desta camada. A camada de saída apresenta 1 neurônio com função de ativação linear.

A Tabela 4.1 apresenta os resultados de previsão de carga a curto prazo obtidos para a RNA tipo MLP.

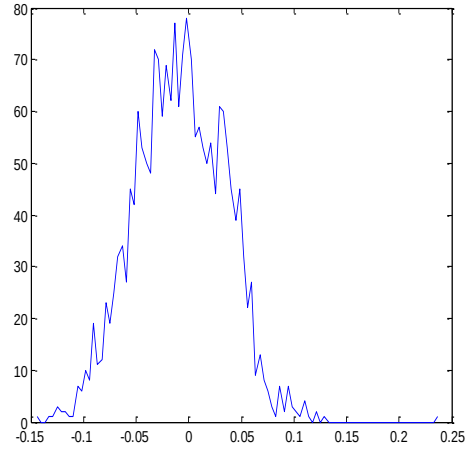
Tabela 4.1 – Resultados da Previsão com MLP

Dados	Erro Médio Quadrático
Treinamento	0,0020
Validação	0,0019

A Figura 4.1 apresenta os gráficos de distribuição do erro para dados de treino e validação.



(a)



(b)

Figura 4.1 – Função densidade probabilidade do erro. (a) dados de treinamento (b) dados de validação no MLP

O índice MAPE (erro absoluto médio), que é geralmente considerado para análise de desempenho de sistemas de previsão de carga para este sistema foi de 7,9%. Este valor está dentro da faixa de valor considerado satisfatório para sistemas de previsão de carga.

4.4 PREVISÃO DE CARGA ATRAVÉS DA METODOLOGIA TFRENN

O segundo método de Inteligência Computacional utilizado neste trabalho para realizar previsão de carga a curto prazo foi o TFRENN (Transparent Fuzzy Rule Extraction from Neural Network), apresentado no Capítulo 2.

O TFRENN realiza o mapeamento de uma rede neural em um sistema difuso matematicamente equivalente, de forma que o conhecimento aprendido durante a fase de treinamento pode ser extraído em forma de regras difusas transparentes do tipo *Takagi-Sugeno* de Ordem Zero.

Desta forma, após o treinamento da rede neural para a previsão de carga, seus pesos sinápticos são convertidos em regras difusas transparentes. De posse dessas regras, especialistas humanos podem ter acesso aos motivos (indicadores) pelos quais a rede chegou a um determinado resultado. O método fornece elementos importantes na análise dos resultados da previsão.

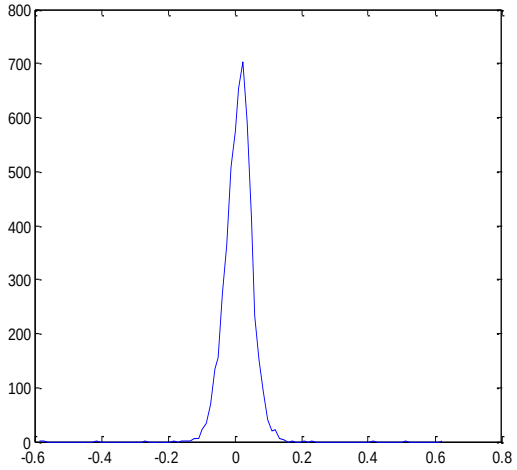
Para o desenvolvimento do sistema previsor baseado no TFRENN, treinamos uma rede neural do tipo MLP descrito no Capítulo 2 (MLP com as devidas restrições de treinamento). Diversas topologias foram testadas sendo que a topologia que apresentou melhor resultado foi a MLP com 40 neurônios na camada escondida.

A Tabela 4.2 mostra os resultados para dados de treino e validação da rede neural treinada com as devidas restrições impostas pela metodologia TFRENN.

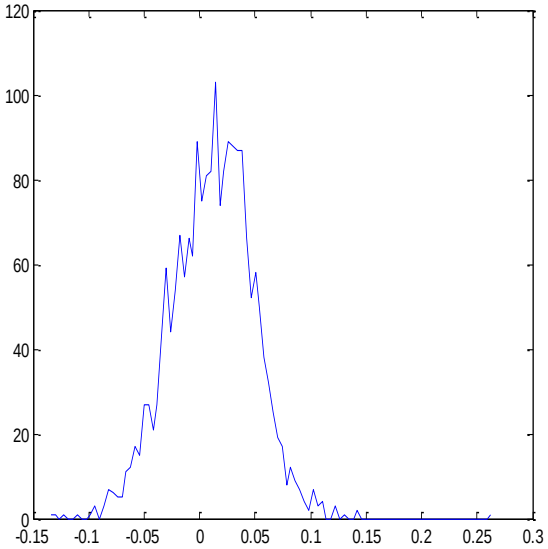
Tabela 4.2 – Resultados com TFRENN

Dados	Erro Médio Quadrático
Treinamento	0,0020
Validação	0,0015

A Figura 4.2 apresenta os gráficos de distribuição do erro para dados de treino e validação.



(a)



(b)

Figura 4.2 – Gráficos de função densidade de probabilidade (a) dados de treinamento (b) dados de validação no TFRENN

Após o treinamento da rede neural, passou-se à fase de extração de conhecimento. O TFRENN foi usado para transformar a rede neural em um sistema difuso. Como a RNA possui duas entradas, foi gerado um sistema difuso com 25 regras transparentes.

Após o mapeamento da rede neural no sistema difuso, o erro médio quadrático foi novamente calculado, ficando em 0,0035 para o treinamento e 0,0021 para validação. Essa pequena diferença se deve à aproximação que é feita das funções de pertinência obtidas em funções transparentes. Ainda assim, se comparados os erros obtidos na rede neural e no sistema difuso extraído, pode-se concluir que a fidelidade no mapeamento foi mantida.

A seguir são apresentadas algumas das regras que fazem parte do sistema difuso extraído da rede neural:

R1: Se (X_1**é** Extremamente Pequeno) **E** (X_2**é** Extremamente Pequeno) **Então** (Y_1**é** 0.7221)

R2: Se (X_1**é** Extremamente Pequeno) **E** (X_2 é Pequeno) **Então** (Y_2**é** 0.2752)

...

R8:Se (X_1**é** Extremamente Pequeno) **E** (X_2**é** Muito Pequeno) **Então** (Y_8**é** 0.6734)

...

R25: Se (X_1**é** Alto) **E** (X_2**é** Alto) **Então** (Y_{25}**é** 0.8)

onde, X_1 e X_2 são os valores de carga das duas horas antecedentes ao valor desejado de previsão e Y é o valor previsto.

A saída do sistema difuso (valor desejado de previsão de carga) é dada por:

$$y = -0.6551 - \sum_{j=1}^{25} v_j y_j \qquad (4.2)$$

As novas regras extraídas podem então ser examinadas por especialistas. O fato de ter sido assegurada a transparência das regras permite ao especialista detectar os méritos individuais de cada regra no processo todo. A partir daí, duas coisas podem acontecer: a regra pode confirmar um conhecimento prévio que o especialista já possuía, corroborando a eficácia do mapeamento rede neural/sistema difuso, ou a regra apresenta um conhecimento novo ou uma nova forma de representá-lo. Nesse caso passamos a ter um momento de descoberta de conhecimento. Em qualquer um dos casos, o processo permite tornar explícito o conhecimento escondido capturado pela rede neural.

O índice MAPE (erro absoluto médio), que é geralmente considerado para análise de desempenho de sistemas de previsão de carga para este sistema, foi de 6.4%. Este valor está dentro da faixa de valor considerado satisfatório para sistemas de previsão de carga.

4.5 PREVISÃO DE CARGA ATRAVÉS DO ANFIS

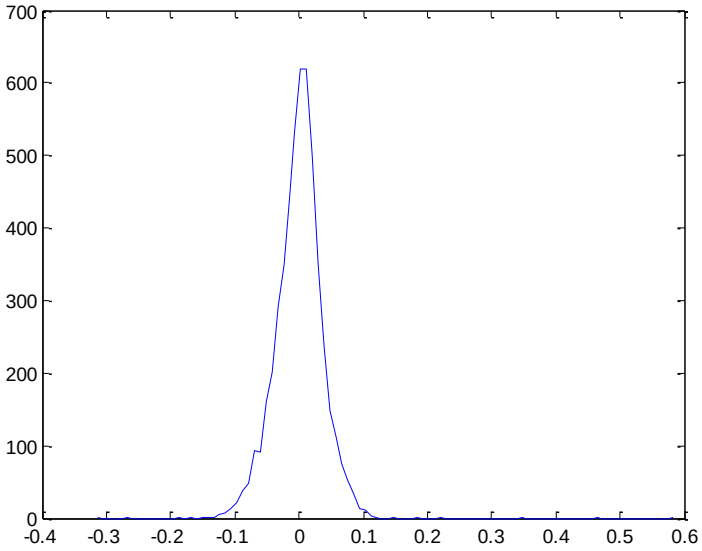
O terceiro método de Inteligência Computacional utilizado neste trabalho para realizar previsão de carga a curto prazo, foi o ANFIS. Alguns testes foram realizados para estabelecer o número de funções de pertinências necessárias para as duas entradas do sistema, sendo que o melhor resultado de previsão foi obtido com duas funções de pertinência do tipo gaussiana.

A Tabela 4.3 mostra os resultados para dados de treino e validação obtidos com o ANFIS.

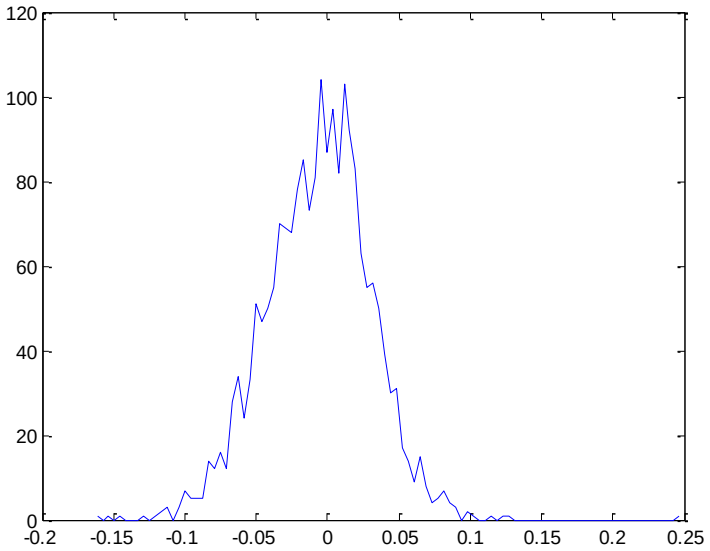
Tabela 4.3 – Resultados com ANFIS

Dados	Erro Médio Quadrático
Treinamento	0.0017
Validação	0.0013

A Figura 4.3 apresenta os gráficos de distribuição do erro para dados de treino e validação.



(a)



(b)

Figura 4.3 - Gráfico de função densiade de probabilidade do erro (a) dados de treinamento (b) dados de validação no ANFIS

O índice MAPE (erro absoluto médio), que é geralmente considerado para análise de desempenho de sistemas de previsão de carga para este sistema, foi de 5,4%. Este valor está dentro da faixa de valor considerado satisfatório para sistemas de previsão de carga.

4.6 PREVISÃO DE CARGA ATRAVÉS DE REDES NEURAIS AUTO-ASSOCIATIVAS E ALGORITMOS GENÉTICOS

Como já apresentado no Capítulo 2, uma rede MLP auto-associativa é um caso especial de rede MLP onde o treinamento da mesma é realizado para que vetor de saída seja idêntico (ou próximo) ao vetor de entrada, desta forma, resultando no que chamamos de armazenamento de vetor.

Após o treinamento da rede auto-associativa, quando são apresentados para a rede padrões cada vez mais distantes dos padrões utilizados durante o aprendizado, o erro na saída da rede será cada vez maior.

Considerando então as características da rede MLP auto-associativa, uma rede foi treinada para associar (aprender) a relação entrada e saída para os dados da série temporal de carga, sendo que a entrada para a MLP foi:

Entrada 3: valor da carga uma hora adiante.
Entrada 2: valor da carga atual.
Entrada 1: valor da carga uma hora antes.

Sendo a saída para a MLP os mesmos valores de entrada já que estamos trabalhando com uma rede auto-associativa, ou seja:

Saída 3: valor da carga uma hora adiante.
Saída 2: valor da carga atual.
Saída 1: valor da carga uma hora antes.

Após a realização de diversos treinamentos variando-se o número de neurônios na camada escondida foi escolhida a MLP com 5 neurônios na camada escondida pois esta apresentou os melhores resultados para dados de treino e validação.

Desde que a entrada 3 e saída 3 da rede são na verdade o valor de previsão desejado para daqui a 1 hora e as entradas e saídas 1 e 2 são os valores passados da carga, então, se entrarmos na rede com os valores de entrada 1 e 2 conhecidos e com um valor da entrada 3 com um valor totalmente diferente do valor de previsão desejado a saída da rede auto-associativa apresentará um valor de erro médio das

saídas alto. E se apresentarmos um valor para entrada 3 próximo ou igual ao valor desejado de previsão, a rede apresentará um valor baixo de erro médio das saídas.

Então para que possamos desenvolver um sistema previsor a partir do uso da rede auto-associativa treinada precisamos a partir dos dados *conhecidos* de entrada 1 e 2, buscar pela entrada 3 *desconhecida* que nos forneça um erro médio das saídas com valor baixo (ou seja, dados os valores da entrada 1 e 2, buscar o valor da entrada 3, que faça com que as entradas e saídas da RNA sejam próximas). Para isto, utilizamos um Algoritmo Genético, que pôde realizar uma busca pelo melhor valor de previsão (entrada 3) para o problema através da minimização da função erro médio na saída da rede auto-associativa.

Note que teremos o valor de previsão próximo ao desejado quando o algoritmo genético encontrar um valor para entrada 3 que faça com que as 3 saídas da rede estejam com valores próximos das três entradas.

A Figura 4.4 apresenta a estrutura do sistema proposto para previsão, dada uma rede auto-associativa já treinada. Uma estrutura deste tipo já vem sendo aplicada com sucesso para problemas de dados faltosos.

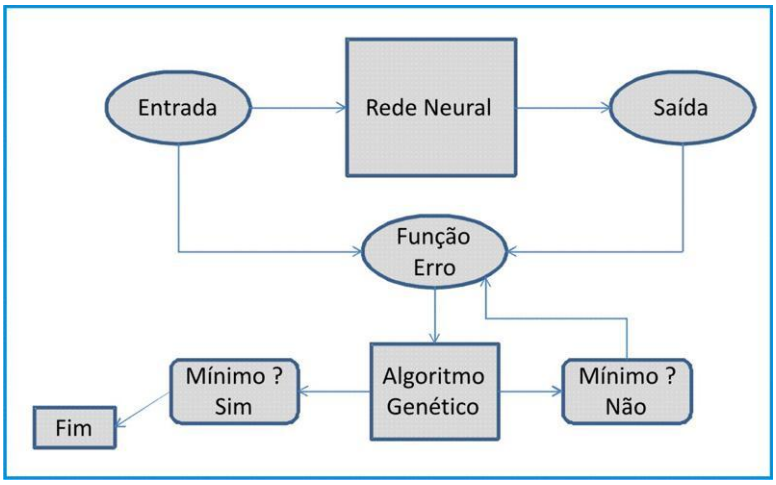


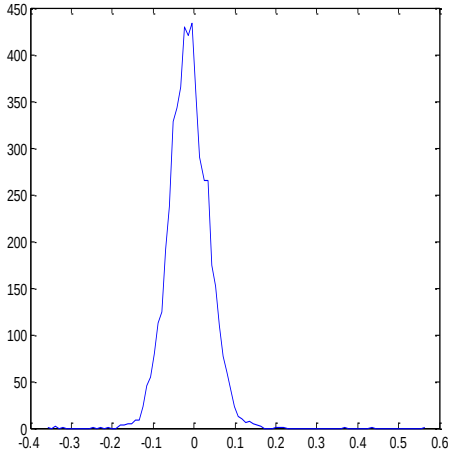
Figura 4.4 – Sistema Híbrido Neuro-Evolutivo proposto para previsão de carga

A Tabela 4.4 mostra os resultados para dados de treino e validação obtidos com o sistema proposto.

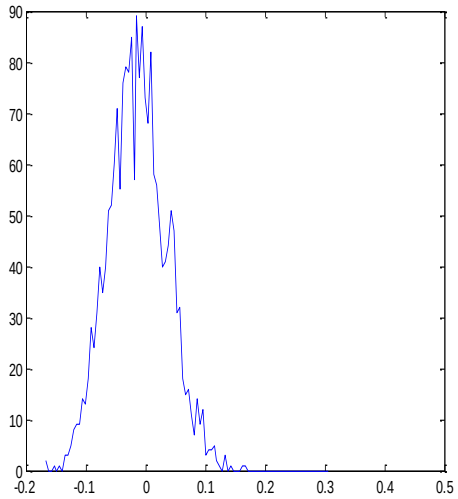
Tabela 4.4 – Resultados com o Sistema Neuro-Evolutivo

Dados	Erro Médio Quadrático
Treinamento	0.0027
Validação	0.0026

A Figura 4.5 apresenta os gráficos de distribuição do erro para dados de treino e validação.



(a)



(b)

Figura 4.5 – Gráfico de função densiade de probabilidade do erro (a) dados de treinamento (b) dados de validação no sistema Neuro-Evolutivo

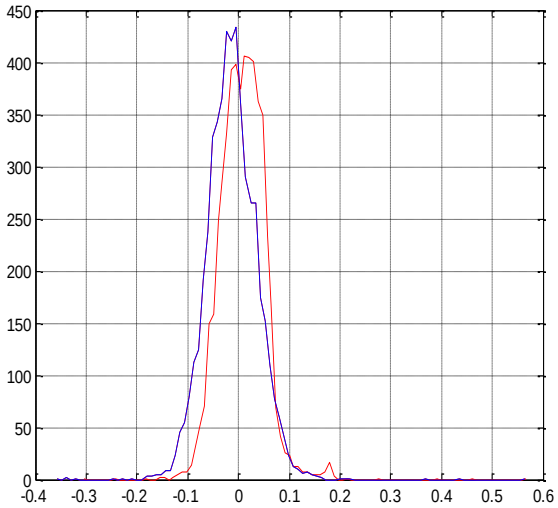
O índice MAPE (erro absoluto médio), que é geralmente considerado para análise de desempenho de sistemas de previsão de carga para este sistema, foi de 7,5%. Este valor está dentro da faixa de valor considerado satisfatório para sistemas de previsão de carga.

4.7 ANÁLISE COMPARATIVA DOS RESULTADOS

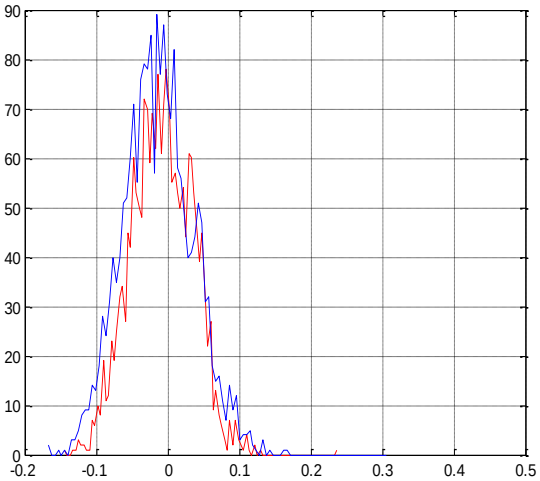
Para comparação de desempenho entre os sistemas previsores desenvolvidos será utilizado o erro médio quadrático e a função densidade probabilidade (fdp) do erro para os dados de validação e treinamento.

4.7.1 Comparando Sistema Previsor baseado na MLP e Sistema Proposto

A Figura 4.6 apresenta os gráficos da função densidade de probabilidade para o sistema previsor baseado na MLP e para o sistema proposto, para dados de treinamento e validação.



(a)



(b)

Figura 4.6 – Gráficos de função densidade de probabilidade para MLP (vermelho) e sistema proposto (azul) - (a) dados de treinamento (b) dados de validação

Apesar de o sistema proposto apresentar um erro médio quadrático maior, pelo gráfico da fdp (e análise dos dados), pode-se perceber que este apresenta um maior número de padrões (de treino e validação) com valores de erro próximos de zero, o que é desejado. A Tabela 4.5 apresenta alguns números comparativos em relação aos valores de erros obtidos para os dois sistemas.

Tabela 4.5 – Comparação de faixas de erro

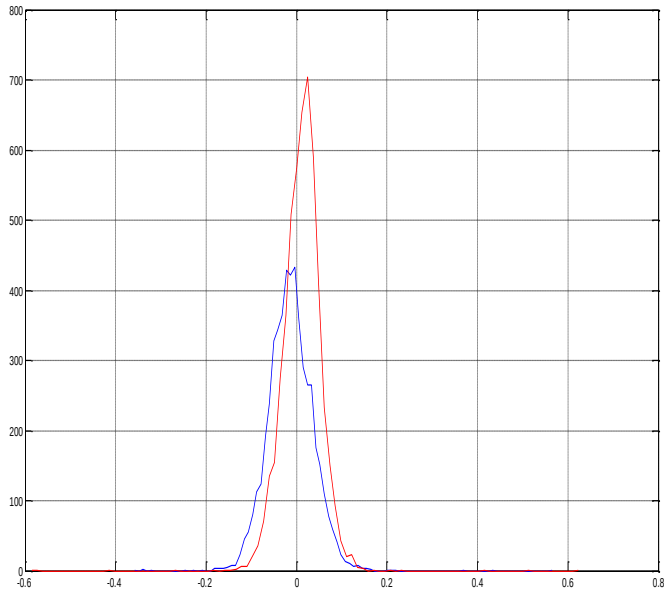
Sistema	**Treino Faixa de erro [0.006 0.003]**	**Validação Faixa de erro [0.0002 0.0094]**
MLP	773	280
Proposto	793	310

Para os dados de treinamento, o sistema MLP apresentou como menor valor de erro o valor de 0.0036, sendo que obteve 375 padrões com valores próximos deste valor. O sistema proposto apresentou como menor valor de erro o valor 0.0039, sendo que obteve 433 padrões com erros próximos deste valor.

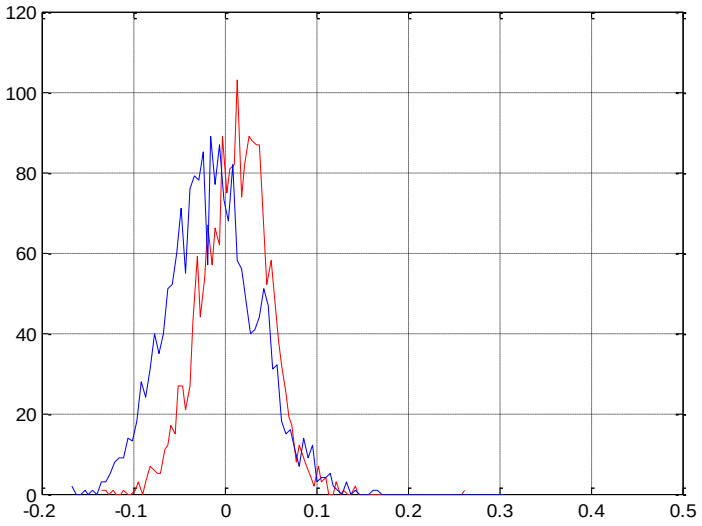
Para os dados de validação, o sistema MLP apresentou como menor valor de erro o valor de 0.0017, sendo que obteve 78 padrões com valores próximos deste valor. O sistema proposto apresentou como menor valor de erro o valor 0.0002, sendo que obteve 73 padrões com erros próximos deste valor.

4.7.2 Comparando Sistema Previsor baseado no TFRENN e Sistema Proposto

A Figura 4.7 apresenta os gráficos da função densidade de probabilidade para o sistema previsor baseado no TFRENN e para o sistema proposto, para dados de treinamento e validação.



(a)



(b)

Figura 4.7 – Gráficos de função densidade de probabilidade para TFRENN (vermelho) e sistema proposto (azul) - (a) dados de treinamento (b) dados de validação

Apesar de o sistema proposto apresentar um erro médio quadrático maior, pelo gráfico da fdp (e análise dos dados) pode-se perceber que este apresenta um maior número de padrões (de treino e validação) com valores de erro próximos de zero, o que é desejado. A Tabela 4.6 apresenta números comparativos em relação aos valores de erros obtidos para os dois sistemas.

Tabela 4.6 – Comparação de faixas de erro

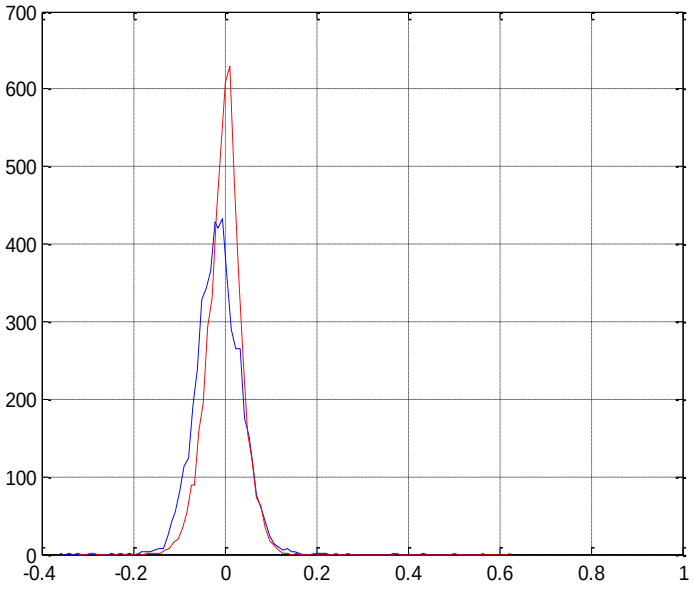
Sistema	**Treino Faixa de erro [0.006 0.002]**	**Validação Faixa de erro [0.0002 0.006]**
TFRENN	575	227
Proposto	793	310

Para os dados de treinamento, o sistema TFRENN apresentou como menor valor de erro o valor de 0.002, sendo que obteve 575 padrões com valores próximos deste valor. O sistema proposto apresentou como menor valor de erro o valor 0.0039, sendo que obteve 433 padrões com erros próximos deste valor.

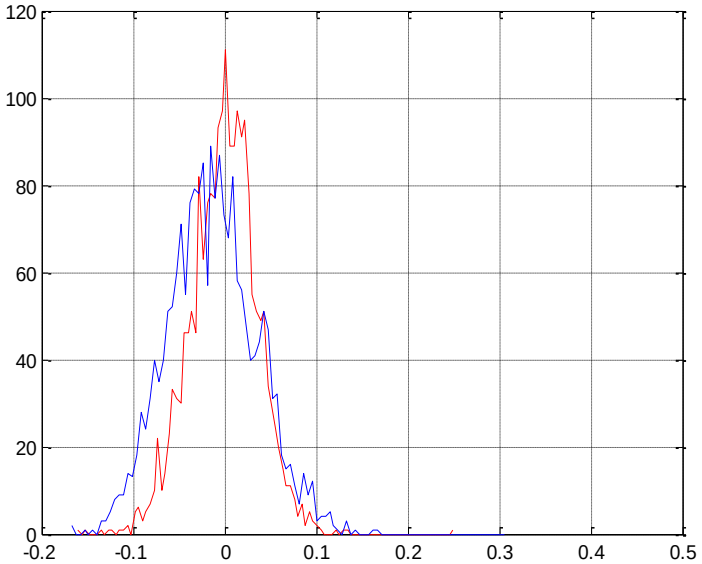
Para os dados de validação, o sistema TFRENN apresentou como menor valor de erro o valor de 0.002, sendo que obteve 144 padrões com valores próximos deste valor. O sistema proposto apresentou como menor valor de erro o valor 0.0002, sendo que obteve 73 padrões com erros próximos deste valor.

4.7.3 Comparando Sistema Previsor baseado no ANFIS e Sistema Proposto

A Figura 4.8 apresenta os gráficos da função densidade de probabilidade para o sistema previsor baseado no ANFIS e para o sistema proposto, para dados de treinamento e validação.



(a)



(b)

Figura 4.8 – Gráficos de função densidade de probabilidade para ANFIS (vermelho) e sistema proposto (azul) - (a) dados de treinamento (b) dados de validação

Apesar de o sistema proposto apresentar um erro médio quadrático maior, pelo gráfico da fdp (e análise dos dados), pode-se perceber que este apresenta um maior número de padrões (de treino e validação) com valores de erro próximos de zero, o que é desejado. A Tabela 4.7 apresenta números comparativos em relação aos valores de erros obtidos para os dois sistemas.

Tabela 4.7 – Comparação de faixas de erro

Sistema	**Treino** **Faixa de erro** **[0.006 0.002]**	**Validação** **Faixa de erro** **[0.0002 0.006]**
ANFIS	609	297
Proposto	793	310

Para os dados de treinamento, o sistema ANFIS apresentou como menor valor de erro o valor de 0.0025, sendo que obteve 609 padrões com valores próximos deste valor. O sistema proposto apresentou como menor valor de erro o valor 0.0039, sendo que obteve 433 padrões com erros próximos deste valor.

Para os dados de validação, o sistema TFRENN apresentou como menor valor de erro o valor de 0.0014, sendo que obteve 111 padrões com valores próximos deste valor. O sistema proposto apresentou como menor valor de erro o valor 0.0002, sendo que obteve 73 padrões com erros próximos deste valor.

CAPÍTULO 5

CONCLUSÕES

Nos dias atuais, principalmente pelo fato de alguns sistemas elétricos serem desregulamentados, o estudo dos problemas de análise, planejamento e operação de sistemas de energia elétrica é de extrema importância para o funcionamento do sistema. Para isso é necessário que se obtenha, com antecedência, o comportamento da carga elétrica com o propósito de garantir o fornecimento de energia aos consumidores de forma econômica, segura e contínua.

A previsão de carga é uma das principais funções na operação de um sistema de energia elétrica. Estimar a demanda de energia futura de forma correta é uma característica desejável no gerenciamento da produção e distribuição dessa energia.

Considerando então a importância da previsão da carga elétrica para o os sistemas de energia elétrica, neste trabalho uma nova abordagem para o problema de previsão de carga via redes autoassociativas e algoritmos genéticos foi avaliada. Três modelos de previsão baseados em Inteligência Computacional foram também apresentados e tiveram seus desempenhos avaliados e comparados com o sistema proposto. Com os resultados alcançados podemos concluir que os 4 modelos se mostraram satisfatórios para o problema de previsão, reforçando assim a aplicabilidade de metodologias de inteligência computacional para o problema de previsão de cargas.

Através das comparações realizadas mostrou-se que apesar do sistema proposto apresentar um erro médio quadrático maior em relação ao outros apresentados, quando da análise do erro absoluto mostrou-se que com esta metodologia conseguiu-se obter um maior número de padrões com erros dentro da faixa [0.006 0.002] para dados de treino e de [0.0002 0.006] para dados de validação. Isto se deve ao fato de que para a metodologia proposta temos a possibilidade de rodar o programa de otimização via algoritmo genético até que o mesmo encontre um valor de previsão que gere um erro mínimo na saída da rede auto-associativa, oportunidade esta que não temos com as outras metodologias testadas. Podemos perceber esta vantagem, quando da geração pela metodologia proposta, de erros absolutos próximos de 0.0002, o que não aconteceu com as outras metodologias cujo menor erro obtido foi de 0.0014.

Em relação ao índice MAPE ter sido maior para a metodologia proposta, percebemos que isto se deu devido ao pequeno número de padrões que apresentou erros maiores que 0.4, que influenciou no índice, o que não aconteceu para os outros sistemas previsores em que o maior valor de erro esteve próximo de 0.3.

É importante salientar que os resultados para o sistema proposto poderiam ainda ser melhorados desde que, para cada padrão, deixássemos o AG (com 100 gerações) rodar apenas 7 vezes para buscar o erro mínimo desejado na saída da rede auto-associativa. Se aumentarmos este número ou aumentarmos o número de gerações do AG, poderemos ter maiores chances do AG encontrar valores de erro de previsão menores.

Em relação aos resultados obtidos para todos os sistemas, os mesmos poderiam ser melhorados com o uso de outras variáveis que de alguma forma poderiam afetar o comportamento da carga, como a temperatura.

REFERÊNCIAS

[1] Lee, K e Park, J. Short-Term Load Forecasting using an Artificial Neural Network. *Transactions on Power Systems, Vol 7, N. 1.* 1992.

[2] Rewagad, A. e Soanawane, V. Artificial Neural Network Based Short-Term Load Forecasting, *0-7803-4886-9/98/IEEE*. 1998.

[3] Murto, P. Neural Network Models for Short-Term Load Forecasting. *Ph.D. thesis*, Dept. of Technology, Helsinki Univ. of Technology, Helsinki, Finlandia. 1998.

[4] Gross, G. e Galiana, F. Short-Term Load Forecasting, *Proceedings of IEEE*, Vol. 75, N. 12. 1987.

[5] Rui, Y. e El-Keib, A. A Review of ANN-based Short-Term Load Forecasting Models, *0094-2898/95 IEEE*, University of Alabama, Tuscaloosa.1995.

[6] S. Haykin, *Neural Networks. A Comprehensive Foundation,* New Jersey: Prentice Hall.

[7] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are Universal Approximators*",* Neural Network, Vol. 2, pp 359-366, 1989.

[8] M. Stinchcombe and H. White, "Universal Approximation using feedforward networks with non-sigmoid hidden layer activation functions". In Porccedings of the International Joint Conference on Neural networks, pp 613-618, San Diego, 1989.

[9] K–I Funahashi, "On the approximate realization of continuous mapping by neural networks"*,* Neural networks,Vol.2, pp183-192, 1989.

[10] G. Cybenko, "Approximation by superposition of sigmoidal functions", Mathematics of Control, Signal and Systems, Vol. 2, pp 303-314, 1989.

[11] V. Y. Kreinovich, "Arbitrary nonlinearity is sufficient to represent all functions by neural networks: a theorem", Neural Networks, Vol. 4, pp. 381-383, 1991.

[12] Y. Ito, "Represntation of functions by superpositions of a step or sigmoidal function and their applications to neural networks theory", Neural Networks, Vol. 4, pp 385-394, 1991.

[13] K. Hornik, "*Approximation* Capabilities of Multilayer Feedforward Networks",Neural Networks, Vol. 4, pp. 251–257, 1991.

[14] J. L. Castro, C. J. Mantas and J. M. Benitez, "Neural networks with a continuous squashing function in the output are universal approximators". Neural Networks, Vol. 13, pp. 561-563, 2000.

[15] J. M. Mendel and R. W. McLaren, "Reinforcement learning control and pattern recognition systems", In Adaptative, learning and Patter Recognition Sytems: Theory and Applications, New York: Academic Press, 1970.

[16] D. E. Rumelhart, G. E. Hinton and R. J. Willians , Learning Internal Representations by Error Propagation. In [Rummelhart and McClelland, pp. 318-362, 1986.

[17] S. E. Fahlman, “An Empirical Study of learning Speed in Back- Propagation Networks”, Carnegie-Mellon Computer Science Rpt. CMU-CS-88-162, 1988.

[18] M. Riedmiller and H M Braun, “A Direct Adaptive Method for Faster Back-propagation Algorithm Learning: The RPROP Algorithm”, In: Proc. of IEEE Int Conf. on Neural Networks (ICNN), pp 586-591, San Francisco, 1993.

[19] M. T. Hagan and M. Menhaj, “Training feedforward networks with the Marquardt algorithm”, IEEE transaction on Neural Networks, Vol. 5, No. 6, pp 989-993, 1994.

[20] L_Xin Wang, *A Course in Fuzzy Systems and Control*, Prentice-Hall International, 1997.

[21] T. Takagi and M. Sugeno, “Fuzzy Identification of Systems and its application to modeling and control”, IEEE Transactions on Systems, man, and Cybernetic, Vol. 15, pp. 116-132, January, 1985.

[22] A. Riid and E. Rustern, “Transparent Fuzzy Systems and Modeling with Transparency Protection”, In Proc. IFAC Symposium on Artifical Intelligence in real Time Control Three Control”, pp 229-235, October, 2000.

[23] A. Lofti, H. C. Andersen and A. C. Tsoi, “Interpretation preservation of adaptative fuzzy inference systems”, Int. J. Approximation Reasoning, Vol. 15, No. 4, pp 379-394, 1996.

[24] J. S. R Jang, “ANFIS: Adaptive-Network-Based Fuzzy Inference System”, IEEE Transaction Systems, Man & Cybernetics, Vol. 23, pp 665-685, 1993.

[25] Castro, A. Knowledge extraction from artificial neural networks - An application to transformer fault diagnosis, *Ph.D. thesis*, Faculty of Eng., Univ. Porto, Porto, Portugal. 2004.

[26] Benitez, J., Castro, J. e Requena, I. Are Artificial Neural Networks Black Boxes? *IEEE Trans. Neural Netw., vol. 8, no. 5, pp. 1156–1164*. 1997.

[27] Castro, A. e Miranda, V. Mapping neural networks into rule sets and making their hidden knowledge explicit—Application to spatial load forecasting, *PSCC02—14th Power Systems Computation Conf*., Sevilla, Spain. 2002.

[28] Wang, L. A Course in Fuzzy Systems and Control.*Upper Saddle River, NJ: Prentice-Hall*. 1997.

[29] Riid, A. e Rustern, E. Transparent fuzzy systems and modeling with transparency protection, *Proc. IFAC Symp. Artifical Intelligence in Real Time Control Three Control*, pp. 229–235. 2000.

[30] Goldberg, D. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley. 1989.

[31] Michalewicz, D. Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag. 1994.

[32] Blickle, T. “Theory of Evolutionary Algorithms and Application to System Synthesis”, Dissertação de Doutorado, Swiss Federal Institute of Technology, Zurique, 1996.

[33] Zebulum, R., Pacheco, M. A. C., Vellasco, M. M. B. R. “Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms”, CRC Press, Boca Raton, Florida. 2001.

[34] Srinivasan. D,. Lee. M. A. “Survey of Hybrid Fuzzy Neural Approaches to Electric Load Forecasting”. IEEE-Berkeley Initiative in Soft Computing.Computer Science Division.University of California. 1995.

[35] Ho, K.L., Hsu, Y.Y. and Yang, C.C.: “Short term load forecasting using a multilayer neural network with an adaptive learning algorithm”. IEEE Transactions on Power Systems. PWRS-7, (1), pp. 141-149. 1992.

[36] Peng, T.M., Hubele, N.F., and Karady, G.G.: “Advancement in the application of neural networks for short-term load forecasting”. IEEE Transactions on Power Systems. PWRS-7, pp. 141-149. 1992.

[37] Rahman, S.: “Generalized knowledge-based short-term load forecasting technique ”. IEEE Transactions on Power Systems. PWRS-8. 1993.

[38] Almeida, C., Fishwich, P. A. e Tang, Z. “Time series forecasting using neural network vs. Box-Jenkins methodology,” Simulation Councies, Inc., pp. 303-310, 1991.

[39] O’Donavan, T. M. “Short Term Forecasting: An introduction to the Box-Jenkins approach,” John Wiley & Sons, New York, 1983.

[40] Lopes, M. L. M.; Minussi, C. R. e Lotufo, A. P. “Electric load forecasting using a fuzzy ART&ARTMAP neural network,” Journal of Applied Soft Computing, Vol. 5, No. 2, pp. 235-244, 2005.

[41] Murto, P. “Neural network models for short-term load forecasting,” Masters Thesis, Department of Engineering Physics and Mathematics, Helsinki University of Technology, 1998.

[42] Karanta, I., "Short Term load forecasting in communal electric utilities", Licentiate thesis, Systems Analysis Laboratory, Helsinki University of Technology. 1990.

[43] Chow, T. W. S., Leung, C. T. "Neural network based short-term load forecasting using weather compensation", IEEE Transactions on Power Systems, Vol.11, No. 4, November 1996, pp. 1736-1742. 1996.

[44] Al-Kandari, A. M.; Soliman, S. A. e El-Hawary, M. E. “Fuzzy short-term electric load forecasting,” Electric Power & Energy Systems, Vol. 26, No. 2, pp. 111-122, 2004.

[45] Metaxiotis, K.; Kagiannas, A.; Askounis, D. e Psarras, J. “Artificial intelligence in short term electric load forecasting: A state-of-the-art survey for the researcher,” Energy Conversion and Management, Vol. 44, No. 9, pp. 1525-1534, 2003.

[46] Swarup, K. S. e Satish, B. “Integrated ANN approach to forecast load,” IEEE Computer Applications in Power, Vol. 15, No. 1, pp. 46-51, 2002.

[47] Yalcinoz, T. e Eminoglu, U. “Short term and medium term power distribution load forecasting by neural networks,” Energy Conversion and Management, Article in Press (Available Online), pp. 1-13, 2004.

[48] Goh, T. N.; Ong, H. L. e Lee, Y. O. “A new approach to statistical forecasting of daily peak power demand,” Elec. Power Systems. Res., vol. 10, no. 2, pp. 145-148, [PL, LS, DY, ARMA, AD, STO, BL, EX]. 1986.

[49] Coperning, S. L.; Reppen, N. D. e Ringlee R. J. ”Experience with weather sensitive load models for short and long-term forecasting,” IEEE Trans. Power App. Syst., vol. PAS-92, no. 6, pp. 1966-1972, [PL,W]. 1973.

[50] Gupta, P. C. “A stochastic approach to peak power demand Syst.”, vol. PAS-90, [PL, AD, STO, EX] forecasting in electric utility systems,” IEEE Trans. Power App Syst. 1971.

[51] Mathewman, P. D. e Nicholson H.“Techniques for load prediction in the electricity-supply,” Roc.Inst. Elec. Eng., vol. 115, no. 10, pp. 1451-1457, [PL, W, EX]. 1968.

[52] Heinemann, G. T. ; Nordman, D. A. e Plant, E. C. “The relationship between summer weather and summer loads - A regression analysis,” IEEE Trans. Power App. Syst., vol. PAS-85, pp. 1144-1154, [PL, W, EX]. 1966.

[53] Moghram, I. and Rahman, S. “Analysis and evaluation of five short-term load forecasting techniques,” IEEE Transactions on Power Systems, Vol. 4, No. 4, pp. 1484-1491, 1989.

[54] Sfetsos, A. “Short-term load forecasting with a hybrid clustering algorithm,” IEEE Proceedings Generation, Transmission, and Distribution, Vol. 150, No. 3, pp. 257-262, 2003.

[55] Villalba, S. A. and Bel, C. A. “Hybrid demand model for load estimation and short term load forecasting in distribution electric systems,” IEEE Transactions on Power Delivery, Vol. 15, No. 2, pp. 764-769, 2000.

[56] Baczynski, D.; Parol, M. “Influence of artificial neural network structure on quality of short-term electric energy consumption forecast,” IEEE Proceedings Generation Transmission Distribution, Vol. 151, No. 2, pp. 241-245, 2004.

[57] Hippert, H. S.; Pedreira, C. E. e Souza, R. C. “Neural networks for short-term load forecasting: Are view and evaluation,” IEEE Transactions on Power Systems, Vol. 16, No. 1, pp. 44-55, 2001.

[58] Liu, K.; Subbarayan, S.; Shouts, R. R.; Manry, M. T.; Kwan, C.; Lewis, F. L. e Naccarino, J. “Comparison of very short-term load forecasting techniques,” IEEE Transactions on Power Systems, Vol. 11, No. 2, pp. 877-882, 1996.

[59] Chen, C. S.; Tzeng, Y. M. e Hwang, J. C. “The application of artificial neural networks to substation load forecasting,” Electric Power Systems Research

---

Vol. 38, No. 2, pp. 153-160, 1996.

[60] Zadeh, L. “Fuzzy Sets,” Information and Control, Vol. 8, pp. 338-353, 1965.

[61] WenXiao, M.; XiaoMin, B. e LianShun, M. “Short-Term Load Forecasting With Artificial Neural Network and Fuzzy Logic”.0-7803-7459-2/02. IEEE Transactions on Power Systems. 2002.

[62] Pacheco, M. A. C. e Vellasco, M. B. R., “Sistemas Inteligentes de Apoio à Decisão: Análise Econômica de Projetos de Desenvolvimento de Campos de Petróleo sob Incerteza”, Série Business Intelligence, ISBN: 978-85-7193-172-5 (brochura), ISBN: 978-85-7193-173-2 (cartonada), 300 págs, Série Business Intelligence, Ed. Interciência e Ed. PUC-Rio, Junho 2007.

[63] Zebulum, R. S., Pacheco, M. A. C., Vellasco, M. B. R., “Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms”, CRC Press, Boca Raton, Florida, ISBN: 0849308658, 2001.

[64] Park, D. C., El-Sharkavi, A. M. e Marks II, R. J. “Electric Load Forecasting using an Artificial Neural Network”. IEEE Transactions on Power Systems 6(2). 442-449. 1991.

[65] Lopes, M. L. M. “Desenvolvimento de Redes Neurais para Previsão de Cargas Elétricas de Sistemas de Energia Elétrica ,” Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Estadual Paulista, 2005.

[66] Ho, K., Hsu, Y. e Yang, C. “Short-Term Load Forecasting using a Multi-Layer Neural Network with an Adaptative Learning Algorithm”. IEEE Transactions on Power Systems. 7: 141-149. 1992.

[67] Hsu, Y., Ho, K. “Fuzz Expert Systems: An Application to Short Term Load Forecasting”. IEEE Proceedings-C, Vol 139, No 6, pp. 471-447, 1992.

[68] Bartkiewicz, W. “Fuzzy Approaches To Short-Term Electrical Load Forecasting”. IEEE Transactions on Power systems. 0-7695-0619-4/00. 2000.

[70] Bakirtzis, A.G., Theocharis, J. B., Kiartzis, S. J. e Satsios, K. J. “Short Term Load Forecasting Using Fuzzy Neural Networks”. IEEE Transactions on Power Systems, Vol. 10, No. 3, 1995.

[71] Barzamini, R., Menhaj, M.B., Khosravi, A. e Kamalvand, SH. “Short Term Load Forecasting for Iran National Power System and Its Regions Using Multi Layer Perceptron and Fuzzy Inference Systems”. Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, 2005.

[72] Braga, M. B. e Castro, A. R. G. “Extração de Conhecimento de Uma Rede Neural Aplicada à Previsão de Carga a Curto Prazo”. CBA – Congresso Brasileiro de Automática. Juiz de Fora, Minas Gerais, Brasil. 2008.

[73] Oliveira, C. M. “Modelo Adaptativo para Previsão de Carga Ativa de Curto Prazo”. Tese de Doutorado. Programa de Pós-Graduação em Engenharia de Produção e Sistemas. Universidade Federal de Santa Catarina. 2004.

[74] Srinivasan, D. “Evolving artificial Neural Networks for short Term Load Forecasting”. Neurocomputing, 23, 265-276. 1998.

[75] El-Sharkawi, M. A., Huang, S. J. “Development of Genetic Algorithm Embedded Kohonen Neural Network for Dynamic”. International Conference on Intelligent Systems Applications to Power Systems.44-49. 1996.

[76] Dash, P. K., Mishra, S., Liew, A. C. “Genetic Optimization of Self Organizing Fuzzy-Neural Network for Load Forecasting”. IEEE Power Engeneering Society Winter Meeting, 2. 2000.