

**Tarcísio Roberto Steinmetz**

**Extração de Regras de Conhecimento a Partir de  
Redes Neurais Artificiais Aplicadas para Previsão  
de Demanda por Energia Elétrica**

Orientador:

Dr. José Vicente Canto dos Santos

São Leopoldo

2009

# RESUMO

Este trabalho apresenta uma metodologia para extração de regras a partir de Redes Neurais Artificiais (RNA) treinadas para previsão de carga. As regras extraídas visam descrever o conhecimento adquirido pela rede neural a respeito do comportamento da demanda por energia elétrica. As regras possuem uma estrutura do tipo *SE premissa ENTÃO consequência*, onde *premissa* refere-se aos dados de entrada submetidos para a rede neural, e *consequência* caracteriza uma equação linear responsável por descrever a saída apresentada pela rede em função dos dados de entrada, caso a premissa da regra seja satisfeita. Dessa forma, além da precisão advinda das características das redes neurais, as mesmas serão capazes de tornar seu conhecimento transparente para o usuário, contribuindo ainda mais para o auxílio das tomadas de decisões táticas e estratégicas das companhias de energia elétrica.

# ABSTRACT

This work presents a methodology for the extraction of rules from Artificial Neural Networks (ANN) trained to forecast the electric load demand. The rules have the ability to express the knowledge regarding the behavior of load demand acquired by the network during training process. The rules are presented to the user in an easy to read format, such as *IF premise THEN consequence*. Where *premise* relates to the input data submitted to the network, and *consequence* appears as a linear equation describing the output to be presented by the network, should the premise part holds true. The rule extraction technique presented here, combined with the predictive abilities of the neural networks, constitutes an important tool for tactical and strategic decision making within the electricity supply companies.

# AGRADECIMENTOS

Agradeço, primeiramente, ao Programa Interdisciplinar de Pós Graduação em Computação Aplicada (PIP-CA) da UNISINOS pela oportunidade de realizar meu curso de mestrado nessa Instituição. Aos professores Adelmo Cechin e José Vicente, deixo um agradecimento especial, pela dedicação e ajuda fornecida durante os dois anos necessários para a conclusão desse trabalho. Obrigado aos amigos (felizmente muitos para serem listados aqui) por terem me ajudado nos momentos difíceis, e por terem me lembrado de que a vida não é feita somente de trabalho e estudo. À minha família, eu agradeço pelo carinho e apoio, sem os quais jamais teria chegado até aqui.

Tarcísio Steinmetz

# LISTA DE FIGURAS

2.1	Gráficos dos dados climáticos antes e após aplicação de AOP . . . . .	18
2.2	Exemplo de uso dos critérios de Kaiser e Jolliffe para seleção dos componentes principais . . . . .	20
2.3	Exemplo de uma arquitetura de rede neural artificial . . . . .	22
2.4	Modelo de um neurônio artificial . . . . .	23
2.5	Funções de Ativação Limiar (a), Sigmóide (b) e Tang. Hiperbólica (c) . . . . .	25
2.6	Rede alimentada adiante com camada única . . . . .	26
2.7	Rede alimentada adiante com múltiplas camadas . . . . .	27
2.8	Rede recorrente sem neurônios ocultos . . . . .	28
2.9	A mesma rede neural treinada com diferentes valores para $\eta$ . . . . .	29
2.10	Conjunto de pessoas jovens no universo de discurso Idade . . . . .	31
2.11	Temperaturas típicas para o verão de Porto Alegre . . . . .	31
2.12	Conjunto difusos normal e subnormal . . . . .	32
2.13	Exemplo de união e intersecção em conjuntos difusos . . . . .	32
2.14	Conjunto difuso A e seu complemento ( $\text{neg}(A)$ ) . . . . .	33
2.15	Comparação de diversos métodos de defuzificação . . . . .	35
2.16	Métodos para particionar o espaço de entrada: grade (a), árvore (b), espalhamento (c) e linear (d) . . . . .	39
2.17	Estrutura do ANFIS . . . . .	41
2.18	Rede neural usada pelo sistema FNES . . . . .	41
2.19	Regra extraída para uma rede neural hipotética . . . . .	45
2.20	Função sigmóide e parte conseqüente das regras difusas (a); função de pertinência ideal para cada região da função sigmóide (b) . . . . .	47

4.1	Desempenho da rede neural usada no estudo de caso . . . . .	64
4.2	Regras extraídas pelo FAGNIS . . . . .	65
4.3	Relevância de cada variável original na composição da saída da rede . . . . .	66
4.4	Regras extraídas pelo método proposto por Benitez . . . . .	67
4.5	Ativação do primeiro neurônio oculto . . . . .	68
5.1	Desempenho da rede neural de extração de regras utilizada no Experimento 1 .	71
5.2	Uso das variáveis originais para compor $PC_1$ . . . . .	74
5.3	Quantidade de informação das variáveis originais para $PC_4$ . . . . .	78
5.4	Demanda por energia elétrica e dados agrupados pela regra 1 . . . . .	79

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>10</b>
<b>2</b>	<b>Fundamentação Teórica</b>	<b>12</b>
2.1	Sistemas de Potência e Demanda de Energia . . . . .	12
2.1.1	Horizontes de previsão . . . . .	14
2.2	Métodos para Seleção de Variáveis . . . . .	15
2.2.1	Critérios para Avaliação de Desempenho de Modelos . . . . .	16
2.2.2	<i>Foward Selection</i> . . . . .	17
2.2.3	<i>Backward Elimination</i> . . . . .	17
2.2.4	<i>Stepwise Regression</i> . . . . .	17
2.3	Análise dos Componentes Principais . . . . .	18
2.4	Redes Neurais Artificiais . . . . .	20
2.4.1	Benefícios das Redes Neurais Artificiais . . . . .	22
2.4.2	Modelos Neurais . . . . .	23
2.4.3	Tipos de Função de Ativação . . . . .	24
2.4.4	Arquiteturas de Redes Neurais . . . . .	25
2.4.4.1	Redes Alimentadas Adiante com Camada Única . . . . .	25
2.4.4.2	Redes Alimentadas Adiante com Múltiplas Camadas . . . . .	26
2.4.4.3	Redes Recorrentes . . . . .	27
2.4.5	Aprendizado em Redes Neurais . . . . .	28
2.5	Conjuntos Difusos . . . . .	30
2.5.1	Operações com conjuntos difusos . . . . .	32

2.5.2	Lógica Difusa . . . . .	33
2.5.2.1	Implicação Difusa e Inferência Difusa . . . . .	34
2.5.2.2	Defuzificação . . . . .	34
2.5.3	Sistemas de inferência difusa . . . . .	35
2.6	Extração de Regras a Partir de Redes Neurais Artificiais . . . . .	37
2.6.1	Separação do Espaço de Dados de Entrada . . . . .	38
2.6.2	ANFIS . . . . .	40
2.6.3	FNES . . . . .	41
2.6.4	NEFCON . . . . .	42
2.6.5	FUN . . . . .	43
2.6.6	Benítez . . . . .	44
2.6.7	FAGNIS . . . . .	46
2.7	Conclusões da Fundamentação Teórica . . . . .	47
<b>3</b>	<b>Revisão Bibliográfica</b>	<b>49</b>
3.1	Previsão de Carga com Redes Neurais . . . . .	49
3.2	Integração de RNAs com Outros Métodos . . . . .	53
3.3	Previsão de Carga com Outros Métodos . . . . .	54
3.4	Previsão de Carga em Feriados e Fins-de-Semana . . . . .	56
3.5	Conclusões Sobre a Revisão Bibliográfica . . . . .	57
<b>4</b>	<b>Metodologia</b>	<b>59</b>
4.1	Descrição dos Métodos de Extração . . . . .	59
4.1.1	Benítez . . . . .	59
4.1.2	FAGNIS . . . . .	60
4.2	Estudo de caso . . . . .	62
4.2.1	Descrição dos Dados Utilizados . . . . .	62

4.2.2	Elaboração do Modelo Neural . . . . .	63
4.2.3	Extração das Regras . . . . .	64
4.2.3.1	Regras Extraídas Pelo Sistema FAGNIS . . . . .	65
4.2.3.2	Regras Extraídas Pelo Método de Benitez . . . . .	67
4.2.4	Conclusões do Estudo de Caso . . . . .	67
<b>5</b>	<b>Experimentos e Resultados</b>	<b>69</b>
5.1	Experimento 1 - Extração de regras a partir de RNA aplicada para previsão de carga para a próxima hora . . . . .	70
5.2	Experimento 2 - Extração de regras a partir de RNA aplicada para previsão de carga para o próximo mês . . . . .	75
<b>6</b>	<b>Conclusões</b>	<b>80</b>
	<b>Referências</b>	<b>82</b>

# 1 INTRODUÇÃO

Atualmente, o Brasil passa por um período de intenso crescimento industrial. Para que esse processo resulte em um benefício para a sociedade brasileira, um importante requisito figura acima de muitos outros: a garantia de um fornecimento de energia elétrica de qualidade e com pequena taxa de interrupção. Essa grande necessidade de um serviço confiável de distribuição de energia elétrica pode ser comprovada ao se verificar a quantidade de órgãos no setor público e privado empenhados em tornar esse requisito uma realidade.

Poucos anos atrás, o Governo Federal iniciou o Programa Nacional de Descentralização (PND) com o objetivo de terceirizar diversos serviços anteriormente desempenhados apenas pelo Estado. Um destes serviços era o fornecimento de energia elétrica para as unidades consumidoras. O PND resultou na criação de diversas empresas privadas que passaram a oferecer à sociedade o serviço de distribuição de energia. Essas empresas deveriam operar sob os critérios e normas técnicas estabelecidos pela Agência Nacional de Energia Elétrica (ANEEL), cujo objetivo é a regulamentação desse serviço no País.

Para medir a qualidade do serviço prestado pelas empresas fornecedoras de energia elétrica, a ANEEL se vale de uma série de métricas e metas como o DEC (Duração Equivalente de Interrupção por Unidade Consumidora) e o FEC (Frequência Equivalente de Interrupção por Unidade Consumidora). Além disso, para garantir a constante atualização das práticas e tecnologias adotadas por essas empresas, a ANEEL também especifica o valor mínimo a ser investido pelas mesmas em atividades de pesquisa e desenvolvimento.

Dentre todas as atividades operacionais executadas nessas empresas, a previsão da demanda de energia ou carga do sistema é de extrema importância. Com base na estimativa da demanda futura, os executivos terão mais segurança nas tomadas de decisões estratégicas como a definição do montante de energia a ser comprado, bem como saber qual será o melhor momento para se realizar essa compra. Além disso, a previsão de carga fornece suporte ao nível gerencial da empresa para as tomadas de decisões táticas como a realização de análises de contingências, gerência de carga e elaboração de políticas de manutenção. Essas atividades, quando apoiadas

por uma previsão de demanda de energia precisa, garantem a redução dos valores resultantes para as métricas avaliadas pela ANEEL descritas acima.

Muitos métodos de previsão de carga foram propostos pela academia, ao longo dos últimos anos. Estes métodos podem ser agrupados em três grandes categorias (CHARYTONIUK; CHEN, Feb 2000) : os métodos paramétricos examinam matematicamente e estatisticamente a demanda por energia elétrica em função dos fatores que afetam a própria demanda. Os métodos não paramétricos visam calcular a demanda diretamente sobre os dados históricos (total de demanda de energia registrado no passado). Finalmente, os métodos de Inteligência Artificial (IA) objetivam a criação de modelos, geralmente utilizando Redes Neurais Artificiais (RNAs), sobre a demanda por energia elétrica. Devido ao sucesso das redes neurais em análises de dados com alta complexidade, ao seu elevado grau de precisão mesmo quando as relações entre as variáveis dependentes e independentes mostram-se altamente não-lineares e à sua grande capacidade de generalização, este tipo de modelo foi selecionado como objeto de estudo dessa dissertação. As redes neurais apresentam, no entanto, uma séria desvantagem: elas são incapazes de demonstrar para o usuário o raciocínio realizado pelas mesmas para chegar até a resposta apresentada com base nos dados de entrada analisados.

Assim, esse trabalho apresenta o seguinte objetivo: propor uma metodologia para extração de regras de conhecimento a partir de RNAs treinadas para previsão de carga. Uma regra pode ser descrita como uma estrutura do tipo  $SE_{premissa} ENTÃO_{consequência}$ , onde *premissa* caracteriza os valores de entrada para a rede neural e *consequência* refere-se ao valor estimado pela rede caso a premissa seja verdadeira. Desse modo, as regras tornam o conhecimento codificado na estrutura interna da rede neural na forma de vetores de pesos, em algo mais próximo do entendimento do usuário. Isso permite a ele aprender o comportamento da demanda por energia ao longo do tempo, quais são as variáveis responsáveis por uma maior perturbação na curva da demanda, entre outras informações importantes. Dois métodos de extração serão avaliados quanto a clareza e a relevância das regras extraídas.

Este trabalho segue o seguinte roteiro: no Capítulo 2 são discutidos os conceitos necessários para o acompanhamento do restante desse documento. O Capítulo 3 apresenta uma revisão bibliográfica sobre o tema previsão de demanda de energia, retratando os esforços da comunidade científica para resolver este problema ainda em aberto. No capítulo 4 encontra-se a metodologia proposta para este trabalho. O Capítulo 5 descreve, com base na metodologia definida anteriormente, uma série de experimentos a serem realizados, bem como os resultados observados. O Capítulo 6 encerra o documento, apresentando as conclusões encontradas e atividades futuras.

## 2 FUNDAMENTAÇÃO TEÓRICA

Esse capítulo apresenta noções gerais de conceitos cujo entendimento é necessário para o acompanhamento deste trabalho. Os tópicos aqui discutidos são complementados com referências a leituras avançadas sobre os mesmos.

### 2.1 Sistemas de Potência e Demanda de Energia

Sistema de Potência é o nome dado ao conjunto de equipamentos utilizados para o fornecimento de energia elétrica. Sistemas de Potência têm como objetivo a captação da energia presente na natureza, sob diversas formas, e a transformação da mesma em energia elétrica, a fim de ser convenientemente transportada para os consumidores através da rede elétrica.

A demanda de energia ou carga do sistema pode ser compreendida como o somatório do uso de energia elétrica de cada ponto de consumo (usuários), atendido por uma estação de fornecimento de energia. O comportamento da curva da demanda por energia é um fenômeno extremamente dinâmico e de difícil compreensão. A quantidade de variáveis responsáveis pela caracterização da curva da demanda por energia é muito grande, e seus efeitos são diferentes nas diversas regiões do mundo. No entanto, trabalhos como (GROSS; GALIANA, Dec. 1987), (SRINIVASAN; CHANG; LIEW, Nov 1995a) e (SRINIVASAN D.; SWEE SIEN TAN; CHENG, Aug 1999) demonstram a existência de um conjunto de fatores presentes em todos os modelos de previsão da demanda por energia. Independente da região mundial em questão, estas variáveis possuem uma forte relação com a carga do sistema elétrico. Estes fatores são discutidos abaixo.

#### **Fatores econômicos**

Os efeitos dos fatores econômicos sobre a carga do sistema são percebidos mais claramente a longo prazo. Esses fatores devem ser levados em conta ao criar ou atualizar modelos de previsão de demanda de um ano para outro (SRINIVASAN D.; SWEE SIEN TAN; CHENG, Aug 1999). Questões econômicas como o crescimento ou decréscimo da atividade industrial, o estado atual do setor agrário e medidas econômicas realizadas pelo governo geram impacto positivo

(aumento) ou negativo (diminuição) da demanda por energia de uma determinada região.

### **Fatores temporais**

Fatores sazonais, ciclo semanal e feriados contribuem largamente para a forma da curva da demanda por energia. Os efeitos dos fatores sazonais na demanda por energia aparecem gradualmente, principalmente com relação à variação da intensidade de iluminação natural, à medida que as estações passam do verão para o inverno. A entrada ou saída do horário de verão, início do ano escolar e a diminuição das atividades durante as férias ou festividades tais como natal e ano novo são outros fatores sazonais responsáveis por grande impacto na demanda por energia.

O ciclo semanal é uma consequência do padrão de trabalho e descanso da população economicamente ativa da região. Este ciclo normalmente apresenta uma baixa demanda por energia durante os fins-de-semana e um aumento da mesma durante os dias úteis. A queda na demanda inicia durante a noite de sexta-feira. Durante a manhã de sábado, a carga do sistema ainda é considerável, pois nem todas as empresas encerram suas atividades durante esse período. Na tarde de sábado e durante todo o domingo a demanda por energia é baixa, e passa a aumentar lentamente na parte da manhã de segunda-feira, assumindo um crescimento intenso durante o meio da manhã. O ciclo então se repete, mantendo alta a demanda por energia durante os dias úteis, vindo a reduzir novamente com a chegada do fim-de-semana.

Feriados têm a tendência de reduzir a demanda por energia abaixo dos níveis “normais”. Esse efeito é ainda mais notável quando o feriado em questão está situado antes ou depois do fim-de-semana. Isso é causado pela criação de um número grande de dias de descanso, onde as atividades comerciais sofrem uma grande redução.

### **Fatores climáticos**

As condições climáticas são responsáveis por grandes variações no padrão da carga do sistema. O principal fator para essa variação é o uso de equipamentos como ar-condicionado nos períodos de inverno e verão e aparelhos de irrigação dos campos de plantações.

A temperatura é o componente mais importante para modelos de previsão de demanda por energia a curto prazo. A variação da temperatura para valores muito diferentes do normal pode resultar na necessidade do replanejamento do uso de energia do sistema de fornecimento. Umidade, temporais, velocidade dos ventos e a cobertura do céu são outros fatores climáticos que possuem forte influência na demanda por energia.

### **Outros fatores**

Sob este grupo encontram-se diversos fatores que não podem ser classificados em função dos grupos apresentados acima. Tratam-se de fatores randômicos como a utilização de equipamentos de grande consumo de energia elétrica (equipamentos da indústria de base, túneis de vento, etc.). Como geralmente não é possível saber com antecedência os horários de uso de tais equipamentos, o impacto causado pela utilização dos mesmos não pode ser previsto com exatidão. Também existem eventos como greves, fechamento de indústrias, programas de televisão importantes (telejornais, por exemplo) cuja ocorrência geralmente é conhecida previamente. No entanto, o impacto causado na carga do sistema devido a estes eventos é incerto.

Convém destacar que não foram encontradas referências a respeito do efeito dessas variáveis sobre o consumo de energia na região metropolitana do Estado do Rio Grande do Sul. Na verdade, a descoberta desta informação caracteriza um dos principais objetivos deste trabalho (ver Capítulo 4, Metodologia).

### 2.1.1 Horizontes de previsão

Uma questão importante a ser definida durante o projeto de um previsor de demanda por energia é o horizonte de previsibilidade. Esta medida mostra para quanto tempo no futuro a previsão informada se refere. Além disso, os autores dedicados a esta linha de pesquisa usam o horizonte de previsibilidade para classificar os sistemas de previsão. Sob este aspecto, um previsor pode ser agrupado em quatro classes distintas:

- Previsores para muito curto prazo ou VSTLF (*Very Short-Term Load Forecaster*): estes previsores informam o estado da demanda para o próximo minuto, ou para várias dezenas de minutos à frente (LIU et al., May 1996). Baseados nessa previsão, os operadores do sistema da unidade de distribuição de energia possuem um maior controle sobre a frequência de carga da unidade.
- Previsores para curto prazo ou STLF (*Short-Term Load Forecaster*): são os previsores discutidos com maior frequência na literatura de previsão de carga. Geram previsões de demanda para as próximas 24 horas, ou até para a próxima semana (VERMAAK; BOTHA, Feb 1998). Também são classificados nessa categoria previsores para a demanda da próxima hora. Estes previsores são usados pelos sistemas de gerenciamento das unidades de distribuição, e para a elaboração do planejamento estratégico das companhias de energia elétrica.
- Previsores para médio prazo ou MTLF (*Midterm Load Forecaster*): realizam previsões para o próximo mês ou para até um ano adiante (DOVEH et al., May 1999). Previsões nessa

base de tempo são usadas principalmente em regiões com mercado de energia desregularizado, ou seja, onde o preço da energia não é fixo. Nesse tipo de mercado, a previsão para médio prazo é usada na firmação de contratos entre companhias de fornecimento de energia e seus clientes (TSEKOURAS; HATZIARGYRIOU; DIALYNAS, Feb. 2006).

- Previsores para longo prazo ou LTLF (*Long-Term Load Forecaster*): trabalham com previsões variando entre dois até cinco ou mais anos no futuro (KERMANSHAHI, 1998). Prever a demanda por energia para horizontes tão longos é uma tarefa extremamente complicada, principalmente pela escassez de literatura sobre o assunto. Segundo os novos regulamentos propostos pela ANEEL, as unidades de fornecimento de energia brasileiras devem apresentar previsões para demanda por energia para os próximos cinco anos (CARPINTEIRO et al., 31 July-4 Aug. 2005).

## 2.2 Métodos para Seleção de Variáveis

Esta seção objetiva apresentar ao leitor as principais metodologias utilizadas para a seleção das variáveis a serem incluídas em modelos, sejam eles redes neurais, regressão linear, entre outros. Ao todo, três métodos serão detalhados (BEAL, 2005): *forward selection*, *backward elimination* e *stepwise regression*.

Os métodos acima permitem selecionar, dentre as variáveis disponíveis para o estudo do problema, aquelas que representam maior impacto no desempenho do modelo. Por desempenho entende-se a exatidão do modelo, ou seja, a precisão das respostas apresentadas pelo mesmo. Dessa forma, as variáveis importantes para o modelo são selecionadas, e as variáveis com pouca ou nenhuma contribuição para o desempenho do modelo devem ser desconsideradas.

Ao remover variáveis irrelevantes ou redundantes, os métodos de seleção de variáveis agregam ao modelo as seguintes contribuições (LIU; HIROSHI, 2006):

1. Melhora da capacidade de generalização
2. Redução do tempo de treinamento
3. Facilita a compreensão do modelo

Antes do detalhamento dos métodos de seleção, é apresentada, abaixo, uma explanação sobre as métricas comumente utilizadas para a avaliação da qualidade do modelo.

### 2.2.1 Critérios para Avaliação de Desempenho de Modelos

Duas métricas básicas são empregadas para quantificar a qualidade de um modelo de qualquer tipo (LEHMANN; CASELLA, 1999): Erro Quadrático Médio (RMSE) e o Coeficiente de Determinação ( $R^2$ ).

O RMSE é uma função da soma do quadrado dos erros (SSE - *Sum of Squared Errors*), o número de observações  $n$  e o número de parâmetros  $p$ . Matematicamente, o RMSE é representado como na seguinte equação:

$$RMSE = \sqrt{\frac{SSE}{n - p}} \quad (2.1)$$

Uma vez analisado os RMSEs de todos os modelos gerados a partir dos dados de treinamento, é destacado como modelo de maior precisão aquele que apresentar o menor valor de RMSE.

O coeficiente de determinação ( $R^2$ ) mede a variabilidade da variável dependente do modelo em função das variáveis independentes. Seu valor é exibido no intervalo (0, 1). O  $R^2$  é uma função da soma total dos quadrados (SST) e do SSE, como mostrado abaixo pela seguinte equação:

$$R^2 = 1 - \frac{SSE}{SST} \quad (2.2)$$

Após o  $R^2$  ter sido calculado para todos os modelos gerados, o modelo com maior valor de  $R^2$  é dito o mais preciso do conjunto.

Embora o  $R^2$  constitua uma forma simples para a avaliação da qualidade do modelo (modelos com  $R^2$  próximos de 1 apresentam boa qualidade, enquanto modelos com  $R^2$  próximos de 0 apresentem baixa qualidade), esta técnica apresenta uma séria desvantagem: o valor de  $R^2$  aumenta com cada variável adicionada ao modelo. Assim, esta abordagem encoraja a adição de todas as variáveis disponíveis para o modelo, inclusive aquelas cujo impacto na melhoria da precisão do mesmo não seja expressivo (BEAL, 2005).

### 2.2.2 *Forward Selection*

O *forward selection* é um processo iterativo onde um modelo pequeno, contendo apenas uma variável independente, é acrescido iterativamente de novas variáveis. O resultado do processo é um modelo composto apenas por variáveis independentes capazes de melhorar a precisão do mesmo (reduzir o RMSE ou aumentar o  $R^2$ ).

Primeiramente, todas as variáveis independentes são analisadas estatisticamente atribuindo-se a cada uma o seu valor  $p$  correspondente, de acordo com o procedimento descrito em (BOZDOGAN, 1987). A variável com menor valor  $p$  abaixo de um  $\alpha$  pré-estabelecido (geralmente 0,15) é selecionada para compor o modelo final. Destaca-se que a variável com menor valor  $p$  apresenta maior relevância estatística para o problema. Na próxima iteração, novos modelos são gerados, adicionando-se à variável selecionada no passo anterior, cada uma das variáveis restantes. Novamente, o valor  $p$  das novas variáveis é calculado, sendo acrescido ao modelo final a variável com menor valor  $p$  abaixo do  $\alpha$ . A execução do algoritmo termina quando não houver variáveis abaixo do alpha estabelecido, ou quando todas as variáveis disponíveis forem adicionadas ao modelo final.

### 2.2.3 *Backward Elimination*

O *backward elimination* é o processo contrário ao *forward selection*. Inicialmente, o modelo é grande, constituindo-se de todas as variáveis disponíveis. Estas são eliminadas do modelo de forma iterativa, até o momento onde todas as variáveis não significativas para o modelo tenham sido descartadas.

O algoritmo começa gerando um modelo contendo todas as variáveis possíveis. Após a geração do modelo, os valores  $p$  para cada uma das variáveis independentes é calculado. A variável com maior valor  $p$  acima do  $\alpha$  estipulado é removida do modelo. Na próxima iteração, um novo modelo é gerado contendo todas as variáveis restantes. Novamente, a variável de maior valor  $p$  acima do  $\alpha$  determinado é removida do modelo. O processo se repete até não restar variáveis cujo valor  $p$  supere o valor de  $\alpha$ .

### 2.2.4 *Stepwise Regression*

O *stepwise regression* é uma composição dos dois métodos de seleção de variáveis citados acima. As variáveis são adicionadas iterativamente ao modelo como no processo de *forward selection*. Após cada inserção, no entanto, é verificada a possibilidade de se remover uma

variável que não mais contribua significativamente com a precisão do modelo. O processo de descarte de variáveis é realizado segundo o *backward elimination*.

O algoritmo inicia com a seleção de uma variável para compor o modelo, como no processo de *forward selection*. Na próxima iteração, a variável com menor valor  $p$  abaixo do  $\alpha$  é adicionada ao modelo. Nesse momento, as variáveis presentes no modelo são verificadas, e aquela com o maior valor  $p$  acima do  $\alpha$  estabelecido é removida do modelo. Este processo é repetido, até o momento onde as variáveis presentes no modelo não apresentem valores  $p$  acima de  $\alpha$ , e as variáveis fora dele não possuam valores abaixo de  $\alpha$ .

## 2.3 Análise dos Componentes Principais

Análise dos Componentes Principais (AOP) ou (*Principal Components Analysis - PCA*) é uma transformação vetorial utilizada para a redução do número de dimensões presentes em uma matriz de dados. Esta técnica é largamente adotada em análise exploratória de dados e na confecção de modelos preditivos tais como redes neurais artificiais. O resultado da aplicação de AOP em uma matriz é uma nova matriz (ou tabela), com um menor número de dimensões (ou colunas) retendo apenas os dados com maior variância e, conseqüentemente, com maior quantidade de informação. Além disso, a estrutura dessa nova tabela armazena em sua primeira coluna os dados com maior variância (o primeiro componente principal), na segunda os dados com a segunda maior variância (o segundo componente principal), e assim por diante.

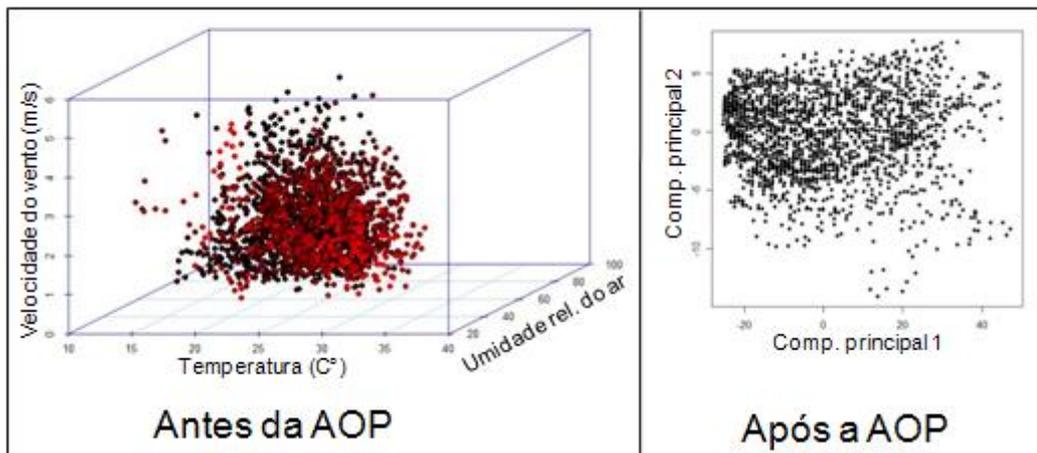


Figura 2.1: Gráficos dos dados climáticos antes e após aplicação de AOP

A extração dos componentes principais de uma tabela de dados exige o cálculo dos auto vetores da matriz de covariância dos dados originais. Este cálculo é geralmente executado após centralizar os dados em relação a sua média, ou seja, cada ponto de dado é subtraído de sua

média. Os auto vetores com maior auto valores representam os dados originais com maior variância, sendo estes os componentes principais. A Figura 2.1 mostra um gráfico com dados climáticos originais (esquerda) e os mesmos dados após a extração dos componentes principais (direita). É possível notar que a AOP provê informações sobre a real estrutura dos dados, apresentando ao usuário os dados sob o melhor ponto de vista apenas, ou seja, os eixos com maior quantidade de informação.

O primeiro componente principal do conjunto de dados analisado, denotado por  $\mathbf{w}_1$ , é descrito na equação abaixo (DUNTEMAN, 1989):

$$\mathbf{w}_1 = \underset{\|W\|=1}{\operatorname{argmax}} \left\{ (W^T X)^2 \right\} \quad (2.3)$$

onde  $X$  é o conjunto de dados,  $W^T$  é a matriz transposta dos auto vetores da matriz de covariância de  $X$ , e  $\operatorname{argmax}$  é o argumento de máximo, ou seja:

$$\operatorname{argmax}_x f(x) \in \{x \mid \forall y : f(y) \leq f(x)\} \quad (2.4)$$

De posse dos  $k-1$  componentes, o  $k$ -ésimo componente pode ser encontrado após a subtração dos  $k-1$  componentes principais de  $X$ :

$$\hat{X}_{k-1} = X - \sum_{i=1}^{k-1} W_i W_i^T X \quad (2.5)$$

e realizando a substituição do mesmo na equação abaixo:

$$\mathbf{w}_k = \underset{\|W\|=1}{\operatorname{argmax}} \left\{ (W^T \hat{X}_{k-1})^2 \right\} \quad (2.6)$$

Com base nas equações acima, pode-se traçar o seguinte conjunto de passos para a extração dos componentes principais de um determinado conjunto de dados  $X$ , para uma matriz  $Y$  contendo um menor número de variáveis (DUNTEMAN, 1989):

1. centrar todos os dados da tabela na média da coluna correspondente (subtrair cada dado da média de sua coluna). Nesse trabalho, os dados também são divididos pelo seu desvio padrão, conforme descrito no Capítulo 4;
2. encontrar os auto vetores e auto valores da matriz de covariância de  $X$ . Armazenar os mesmos nas matrizes  $V$  e  $U$ , respectivamente;
3. ordenar as matrizes  $V$  e  $U$  em ordem decrescente de auto valor;
4. transpor a matriz de auto vetores, de forma que cada auto valor ocupe uma linha da matriz;
5. multiplicar  $V^T$  por  $X$  armazenando o resultado em  $Y$ ;

6. manter as primeiras colunas de  $Y$  seguindo um critério de seleção (demonstrado abaixo). Estes serão os componentes principais. As demais colunas podem ser descartadas.

Dois critérios são geralmente utilizados para selecionar os componentes principais de um determinado conjunto de dados. O critério proposto por Kaiser (KAISER, 1960) sugere selecionar os componentes com valor de auto valor maior que 1. Já o critério definido por Jolliffe (JOLLIFFE, 2002) reforça a seleção dos componentes com auto valor maior que 0,7. A Figura 2.2 apresenta

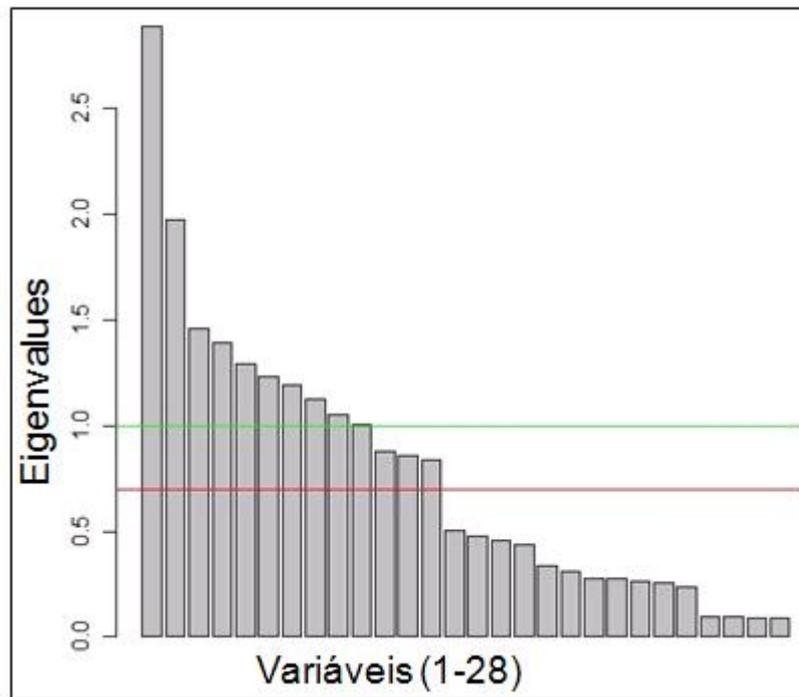


Figura 2.2: Exemplo de uso dos critérios de Kaiser e Jolliffe para seleção dos componentes principais

um gráfico com os auto valores encontrados durante a extração dos componentes principais de uma matriz fictícia. As linhas horizontais indicam as variáveis que seriam selecionadas de acordo com o critério adotado para a escolha dos componentes principais: caso o critério de Kaiser fosse adotado, as variáveis tocadas pela linha verde seriam selecionadas para o estudo. De forma análoga, caso o critério de Jolliffe fosse empregado, as variáveis tocadas pela linha vermelha seriam consideradas. Em qualquer caso, as variáveis não tocadas pelas linhas seriam descartadas.

## 2.4 Redes Neurais Artificiais

As redes neurais artificiais, comumente chamadas apenas de redes neurais, têm sido largamente utilizadas na academia, comércio e indústria. O principal motivo para isso reside na

sua capacidade de mimetizar uma função chave do cérebro humano: a capacidade de aprender através da experiência.

Em seu livro, Haykin (HAYKIN, 2001) descreve uma rede neural como:

Uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso. Ela se assemelha ao cérebro em dois aspectos:

1. O conhecimento é adquirido pela rede a partir do seu ambiente através de um processo de aprendizagem.
2. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido

Uma rede neural aprende através de um algoritmo de aprendizagem, processo no qual os valores de seus pesos sinápticos são ajustados de forma a alcançar o objetivo para o qual a rede foi projetada. A Figura 2.3 ilustra uma rede neural contendo três camadas de neurônios, conectados através de pesos sinápticos. A essa arquitetura de rede neural é atribuída a nomenclatura de *multilayer perceptron*, que será detalhada adiante.

As redes neurais podem ser aplicadas em virtualmente qualquer projeto onde exista uma relação entre variáveis independentes (entradas para o sistema) e variáveis dependentes (saídas para o sistema), mesmo quando este relacionamento for complexo, altamente não-linear, ou de qualquer forma difícil de ser compreendido. Abaixo encontram-se alguns exemplos onde as redes neurais foram aplicadas com sucesso (WITTEN; EIBE, 2005):

- **Conceção de crédito:** diversas características podem ser obtidas a respeito dos candidatos a empréstimos bancários (educação, ocupação atual, idade, etc.). Após treinamento em uma base de dados histórica, uma rede neural pode ser capaz de prever o risco de crédito associado a cada candidato, com base nas informações a ela fornecidas.
- **Monitoramento de condições de maquinário:** uma rede neural pode auxiliar no agendamento de manutenções preventivas em equipamentos, ao aprender os sons emitidos pelas máquinas quando em um estado de funcionamento normal e em estado onde o equipamento está próximo de apresentar problemas. Dessa forma, a rede pode alertar o técnico da possibilidade de uma determinada máquina precisar de manutenção, antes da mesma apresentar defeitos graves, causando prejuízo à empresa.

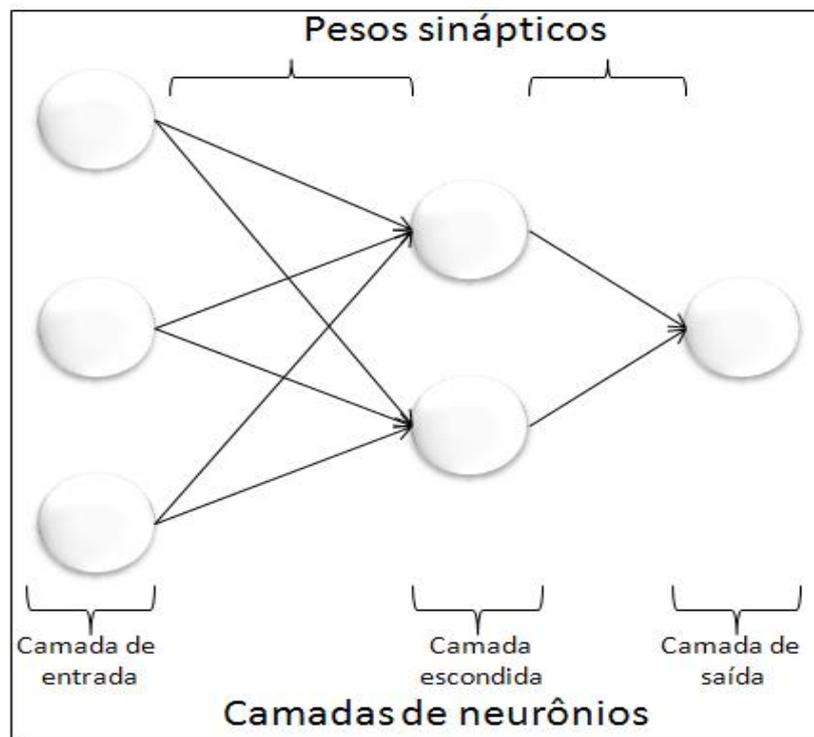


Figura 2.3: Exemplo de uma arquitetura de rede neural artificial

- **Controle de motor:** uma rede neural pode ser treinada para analisar o sinal dos sensores de um motor, ou outro equipamento. Após o treinamento, a rede pode controlar o motor com base nos sinais transmitidos a ela pelo conjunto de sensores, para atingir um determinado objetivo, como por exemplo, reduzir o consumo de combustível.

### 2.4.1 Benefícios das Redes Neurais Artificiais

O principal benefício apresentado pelas redes neurais é sua capacidade de generalização. Por generalização, entende-se a habilidade da rede de apresentar resultados corretos para dados de entrada que não estavam presentes durante o seu treinamento. Além dessa importante característica, outras também podem ser citadas (HAYKIN, 2001):

1. *Não-linearidade:* um neurônio artificial pode ser linear ou não-linear. Uma rede neural com conexões não-lineares também é dita não-linear. A não linearidade é uma característica importante, principalmente se o fenômeno a ser estudado pela rede for intensamente não linear. Entre vários exemplos de fenômenos não lineares, citam-se: Demanda por energia em função da variação da temperatura de uma determinada região, a estimativa da cotação de determinadas ações na bolsa de valores em função de dados históricos, entre outros.

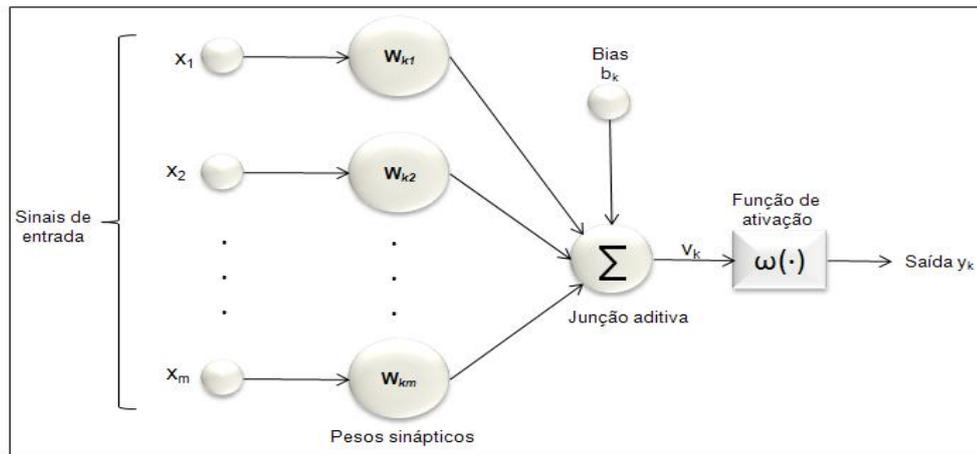


Figura 2.4: Modelo de um neurônio artificial

2. *Adaptabilidade*: uma rede neural tem a capacidade inata de adaptar seus pesos sinápticos para compensar modificações do meio ambiente. Da mesma forma, caso a rede esteja sendo utilizada em um ambiente específico, a mesma pode ser retreinada para lidar com as modificações nas condições do ambiente.
3. *Resposta a evidências*: uma rede neural usada para classificação de padrões pode informar, além do padrão definido para o conjunto de entrada, um valor de confiança para a decisão da rede. Dessa forma, é possível descartar padrões ambíguos (se necessário) melhorando seu desempenho.

## 2.4.2 Modelos Neurais

A Figura 2.4 esquematiza um neurônio não-linear. Ao observar a Figura 2.4, é possível notar a presença de quatro importantes componentes no modelo neuronal:

1. O conjunto de sinapses, ou elos de conexão é responsável por ligar dois neurônios, geralmente situados em camadas adjacentes. É dito que a sinapse  $w_{ik}$  conecta os neurônios  $i$  e  $k$  com determinado valor de peso ou força. Quando um sinal é enviado do neurônio  $i$  para o neurônio  $k$ , o mesmo é multiplicado pelo peso da sinapse. Além disso, o valor do peso da sinapse pode ser positivo ou negativo, resultando em uma excitação ou inibição do neurônio localizado na terminação da sinapse.
2. Um somador, para somar os sinais recebidos pelo neurônio, ponderados pelos pesos das conexões por onde estes sinais foram enviados. Estas operações caracterizam uma combinação linear.

3. Uma função de ativação, com o objetivo de restringir a amplitude do sinal de saída do neurônio. Comumente, é usado como saída do neurônio o intervalo normalizado fechado  $[0, 1]$ . No entanto, existem aplicações de redes neurais onde o intervalo  $[-1, 1]$  é utilizado.
4. Um *bias* aplicado ao resultado da soma ponderada realizada no neurônio. O *bias* tem a função de aumentar ou reduzir o valor de entrada para a função de ativação, caso o mesmo seja positivo ou negativo, respectivamente.

Matematicamente, um neurônio pode ser descrito através do seguinte par de equações:

$$u_k = \sum_{j=1}^m w_{kj}x_j \quad (2.7)$$

e

$$y_k = \varphi(u_k + b_k) \quad (2.8)$$

Onde  $x_1, x_2, \dots, x_n$  são os sinais de entrada;  $w_{k1}, w_{k2}$  e  $w_{kn}$  são os pesos sinápticos do neurônio  $k$ ;  $u_k$  é a saída do combinador linear;  $b_k$  é o *bias*;  $\varphi$  é a função de ativação e  $y$  é o sinal de saída do neurônio.

### 2.4.3 Tipos de Função de Ativação

A função de ativação é responsável por limitar o valor de saída do neurônio em termos do campo local induzido  $v$  (HAYKIN, 2001). Abaixo são apresentadas três funções de ativação comumente usadas em projetos envolvendo redes neurais.

#### Função de Limiar

A função de ativação limiar, retratada na Figura 2.5(a), foi proposta por McCulloch e Pitts (MCCULLOCH; PITTS, 1943). Neurônios com essa função exibem um comportamento do tipo *tudo-ou-nada*, onde sua saída é zero para valores de ativação negativos e 1 para valores de ativação positivos. A função limiar é definida pela seguinte equação.

$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \quad (2.9)$$

#### Função Sigmóide

Essa é de longe a função mais utilizada na construção de redes neurais artificiais. Ela é dita estritamente crescente e possui um balanceamento adequado entre o comportamento linear e não-linear (WITTEN; EIBE, 2005). Um exemplo de função do tipo sigmóide é a Função

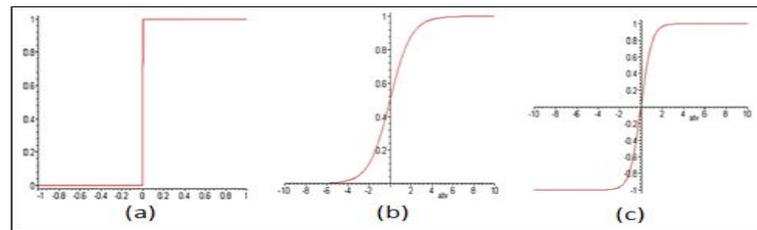


Figura 2.5: Funções de Ativação Limiar (a), Sigmóide (b) e Tang. Hiperbólica (c)

Logística, ilustrada na Figura 2.5(b) e definida por

$$\varphi(v) = \frac{1}{1 + e^{-av}} \quad (2.10)$$

onde  $a$  é o parâmetro de inclinação da curva da função logística. Uma importante característica da função sigmóide é o fato dela ser derivável em qualquer ponto. Essa característica permite a adição de mais camadas escondidas à rede, para aumentar seu desempenho em situações onde a relação entre as entradas e a saída da rede apresentam de grande complexidade. A utilização de uma função de ativação não derivável em qualquer ponto acarreta na impossibilidade da rede neural fazer uso de múltiplas camadas ocultas (HAYKIN, 2001).

### Função Tangente Hiperbólica

Esta função é correspondente à Sigmóide, no entanto, ela assume valores no intervalo  $[-1, 1]$ , conforme a Figura 2.5(c). A função Tangente Hiperbólica é definida por

$$\varphi(v) = \tanh(v) \quad (2.11)$$

## 2.4.4 Arquiteturas de Redes Neurais

A arquitetura define a forma como os neurônios da rede neural estão conectados. Além disso, a arquitetura é o principal fator a ser considerado para a seleção do algoritmo de treinamento utilizado para o aprendizado da rede (HAYKIN, 2001). Nesta seção, serão apresentadas três arquiteturas de redes neurais artificiais.

### 2.4.4.1 Redes Alimentadas Adiante com Camada Única

Este é o tipo mais simples de arquitetura de rede neural disposta em camadas. Trata-se de uma rede com uma camada de neurônios de entrada ligados aos neurônios das camadas de saída através de pesos sinápticos. O fluxo de dados parte da camada de entrada da rede até os neurônios da camada de saída. O caminho contrário nunca é realizado. Dessa forma, redes

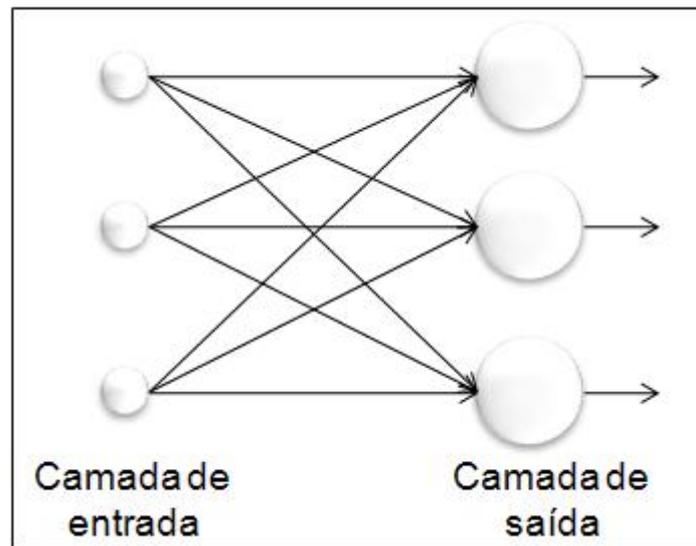


Figura 2.6: Rede alimentada adiante com camada única

neurais dessa categoria são ditas estritamente conectadas adiante ou acíclicas. Na Figura 2.6, encontra-se um exemplo desse tipo de arquitetura para uma rede com três neurônios na camada de entrada e três neurônios na camada de saída. Refere-se a este tipo de rede neural como rede de camada única, pois todo o processamento computacional é realizado somente na camada de saída da rede. A camada de entrada não realiza qualquer computação.

#### 2.4.4.2 Redes Alimentadas Adiante com Múltiplas Camadas

Esta arquitetura distingue-se das redes alimentadas adiante com camada única devido à presença de uma ou mais camadas ocultas, dispostas entre a camada de entrada e de saída da rede. Os neurônios pertencentes a estas camadas são chamados de neurônios ocultos. A adição de camadas ocultas na rede permite dotá-la da capacidade de extrair estatísticas de ordem elevada a partir dos dados de entrada da rede. Em outras palavras, a rede adquire uma perspectiva global, apesar da sua conectividade local (CHURCHLAND; SEJNOWSKI, 1992). O uso de camadas ocultas torna-se interessante quando a camada de entrada da rede possui uma elevada quantidade de neurônios (HAYKIN, 2001).

A rede de múltiplas camadas mais conhecida é a *multilayer perceptron*. Este tipo de rede já foi utilizado em diversos problemas difíceis, inclusive previsão de carga, apresentando altos graus de sucesso. As redes *multilayer perceptron* são treinadas por um algoritmo igualmente popular, conhecido como *backpropagation*, descrito adiante.

O fluxo de dados nas redes de arquiteturas com múltiplas camadas ocorre da seguinte forma: os neurônios da camada de entrada fornecem os estímulos para os neurônios da pri-

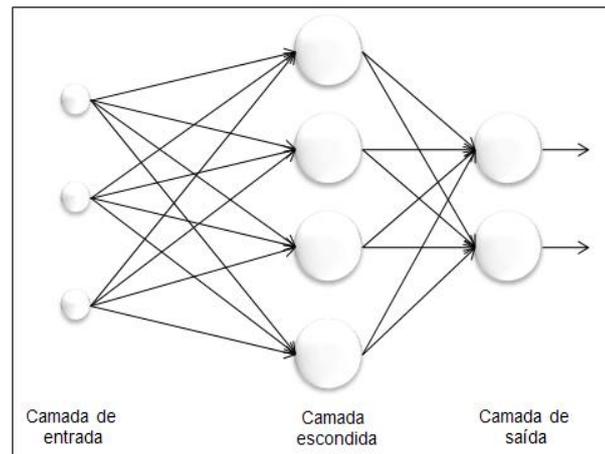


Figura 2.7: Rede alimentada adiante com múltiplas camadas

meira camada oculta. Após o processamento, a saídas destes neurônios são transmitidas para os neurônios da próxima camada oculta (se houver), e assim por diante, até os dados alcançarem a camada de saída. Os sinais dos neurônios da camada de saída da rede representam a resposta global da mesma para os estímulos (dados) coletados pelos neurônios da camada de entrada. A Figura 2.7 ilustra uma rede neural de múltiplas camadas com apenas uma camada escondida. Esta rede é convencionalmente chamada de 3-4-2, por possuir 3 neurônios na camada de entrada, quatro na camada escondida e dois na camada de saída.

Considera-se a rede neural da Figura 2.7 totalmente conectada, pois cada neurônio de uma camada da rede está conectado a todos os neurônios da camada imediatamente posterior. De forma análoga, a rede é considerada parcialmente conectada caso algumas conexões neurais estiverem faltando.

#### 2.4.4.3 Redes Recorrentes

Uma rede neural recorrente diferencia-se de uma rede neural alimentada adiante por possuir pelo menos um laço de realimentação (HAYKIN, 2001). Um exemplo de rede neural recorrente pode ser encontrado na Figura 2.8. Nela, é possível visualizar uma rede composta de apenas uma camada onde a saída de cada neurônio alimenta as entradas de todos os outros neurônios da rede.

As ligações de realimentação de uma rede neural recorrente causam um impacto profundo no seu aprendizado. Os elementos de atraso unitário são responsáveis por acrescentar a rede um comportamento dinâmico não-linear (HAYKIN, 2001). Convém destacar que esta não-linearidade será observada somente se os neurônios presentes na rede forem do tipo não-linear.

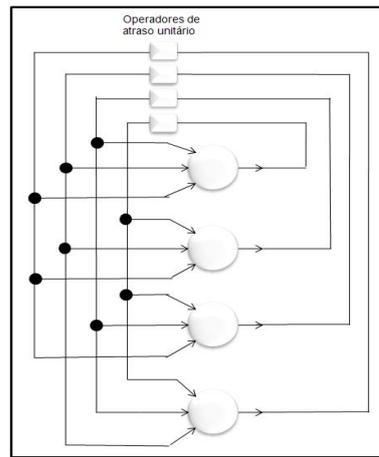


Figura 2.8: Rede recorrente sem neurônios ocultos

### 2.4.5 Aprendizado em Redes Neurais

A aprendizagem é uma das propriedades mais importantes das redes neurais artificiais. A rede aprende a partir de seu ambiente, melhorando seu desempenho através da aprendizagem. Uma rede neural aprende sobre seu ambiente através de um processo iterativo de ajustes sobre os valores de seus pesos sinápticos e *biases* (HAYKIN, 2001). É possível agrupar os algoritmos de aprendizagem em duas classes distintas: algoritmos supervisionados e não-supervisionados.

No aprendizado supervisionado, existe a figura de um professor dotado de conhecimento sobre o ambiente onde a rede neural deve operar. Este conhecimento consiste de um conjunto de entradas e saídas, onde são descritos os dados de entrada para a rede e a saída esperada para estes dados. Assim, os parâmetros livres da rede são adaptados em função do sinal de erro verificado para cada par de entrada e saída. O sinal de erro é obtido através da diferença entre o resultado esperado pelo professor e o resultado apresentado pela rede neural. O processo de treinamento é encerrado quando a rede apresentar sinais de erros baixos para o conjunto de treinamento. Nesse momento, o professor pode ser dispensado, pois a rede está apta para lidar com o ambiente por si mesma.

Já no aprendizado não-supervisionado a figura de um professor não existe. Dessa forma, os parâmetros livres da rede são adaptados segundo uma medida independente da tarefa da qualidade de representação que a rede neural deve aprender (HAYKIN, 2001). Uma vez treinada, a rede está apta para formar representações internas capazes de codificar as características de entrada e assim, gerar novas classes de saída.

O algoritmo *backpropagation* é o algoritmo de treinamento supervisionado mais famoso e utilizado atualmente (WITTEN; EIBE, 2005). Ele pode ser resumidamente descrito como segue. Primeiramente, os valores de *bias* e pesos sinápticos são sorteados para a rede neural a ser

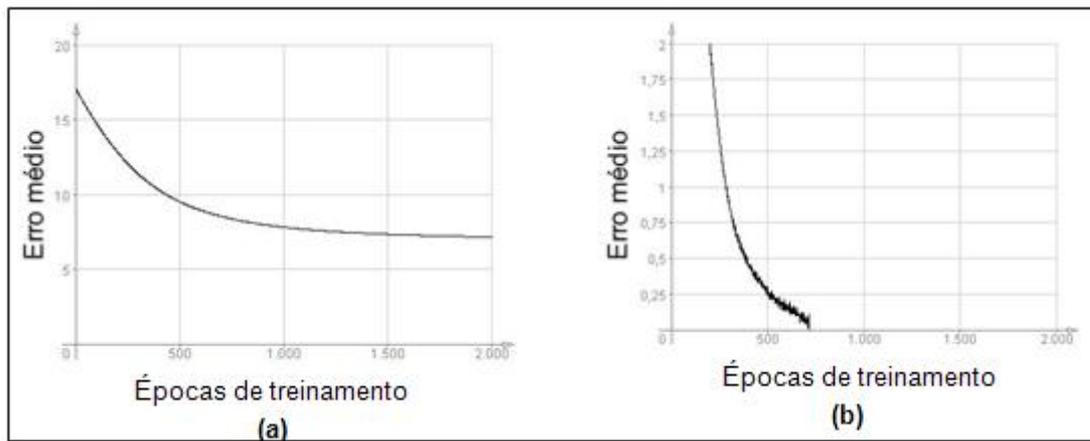


Figura 2.9: A mesma rede neural treinada com diferentes valores para  $\eta$

treinada. Após, os dados do conjunto de treinamento são apresentados à rede. Para cada par de valores de entrada e saída, o sinal de erro é calculado e usado para atualizar os parâmetros da rede. Os pesos sinápticos são ajustados no sentido da camada de saída da rede até a primeira camada oculta. O treinamento é encerrado quando o critério de parada é alcançado. Dentre os critérios de parada mais utilizados, encontram-se: quantidade de épocas de treinamento, valor de erro médio da rede abaixo do especificado, entre outros.

A curva do erro médio apresentada pela rede no decorrer de seu treinamento deve ser tão suave quanto possível (WITTEN; EIBE, 2005). Para isso, deve ser definido um valor de taxa de aprendizagem ( $\eta$ ) para o algoritmo de treinamento. Quanto menor o valor de  $\eta$ , menor será a variação dos pesos sinápticos da rede, resultando em uma convergência do erro médio apresentado pela rede de forma suave, em direção ao zero. No entanto, um valor baixo para  $\eta$  também resulta em uma aprendizagem lenta. Por outro lado, se o valor de  $\eta$  for alto, com o objetivo de acelerar a taxa de aprendizagem, a rede se torna instável (oscilatória). Um meio de aumentar a taxa de aprendizagem e ainda assim evitar a instabilidade da rede é usar o parâmetro *momentum* (RUMELHART; HILTON; WILLIAMS, 1986). O *momentum* ( $\alpha$ ) é usualmente um valor positivo responsável por controlar a correção dos pesos sinápticos, acelerando a aprendizagem quando o erro obtido pela rede diminui de uma iteração para outra, e desacelerando quando o erro aumenta de uma iteração para outra.

A Figura 2.9 mostra o efeito oriundo da utilização de valores baixos e altos para a taxa de aprendizagem. A Figura 2.9(a) apresenta o erro médio obtido em treinamento com taxa de aprendizagem muito pequena, sendo seu valor igual a 0,002. É possível notar a suavidade da curva mas, no entanto, após 2000 épocas de treinamento a rede ainda apresenta um erro elevado. Já a Figura 2.9(b) apresenta o resultado do treinamento com uma taxa de aprendizado igual a 1 (o valor máximo). É possível verificar que embora a rede tenha convergido muito rapidamente,

existe uma grande perturbação na curva de erro médio da rede.

## 2.5 Conjuntos Difusos

Nesta seção, serão apresentados conceitos da teoria dos conjuntos difusos ou *fuzzy*. O conteúdo aqui descrito é necessário para o acompanhamento de algumas das técnicas de extrações de regras a partir de redes neurais artificiais demonstradas na próxima seção. Segundo (KANDEL, 1986), Zadeh formulou a Teoria dos Conjuntos Difusos para usá-la como um meio de representar incerteza. A partir dos conjuntos difusos, foi possível formular os conceitos das operações em conjuntos difusos e inferência difusa discutidos a seguir.

Na teoria dos conjuntos tradicionais, a pertinência de um determinado elemento a um conjunto é avaliada de forma bivalente, onde o elemento pode ou não pertencer ao conjunto em questão. Os conjuntos difusos podem ser vistos como uma extensão dos conjuntos tradicionais, onde um elemento possui certo grau de pertinência àquele conjunto. O grau de pertinência de um elemento a um determinado conjunto é avaliado através da função de pertinência definida para o conjunto. É possível notar, então, que os conjuntos difusos generalizam os conjuntos tradicionais, indicando a pertinência de elementos em um conjunto com valores dentro do intervalo  $[0, 1]$ , enquanto os conjuntos tradicionais permitem pertinência apenas com valores 1 (pertence ao conjunto) ou 0 (não pertence ao conjunto) (DUBOIS; PRADE, 1988).

Um conjunto difuso é um par  $(A, m)$ , onde  $A$  é um conjunto e  $m : A \rightarrow [0, 1]$ .

Para cada  $x \in A$ ,  $m(x)$  É o grau de pertinência de  $x$ .

$x \in (A, m) \Leftrightarrow x \in A \wedge m(x) \neq 0$ . Se  $A = \{x_1, \dots, x_n\}$  o conjunto difuso  $(A, m)$  é descrito como  $\{m(z_1)/z_1, \dots, m(z_n)/z_n\}$ .

Um elemento mapeado pela função de pertinência como 0 não está incluído no conjunto difuso. Em contrapartida, um elemento mapeado como 1 está totalmente incluído no conjunto. Valores encontrados entre os extremos do intervalo caracterizam a intensidade de pertinência dos elementos a este conjunto.

O conjunto  $X$  é chamado de universo de discurso para o conjunto difuso  $F$  e  $\mu_F(x)$  é a função de pertinência de  $F$ . A Figura 2.10 apresenta o conjunto de pessoas jovens no universo de discurso Idade. Este conjunto possui função de pertinência definida por  $\mu_{jovem}$ .

Abaixo encontram-se outros conceitos igualmente importantes:

### Valor lingüístico

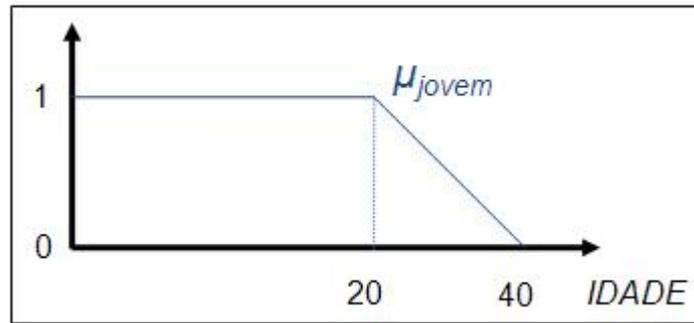


Figura 2.10: Conjunto de pessoas jovens no universo de discurso Idade

É um par  $(adjetivo, F)$ , onde  $adjetivo$  é um valor da linguagem natural e  $F$  é um conjunto difuso. Uma **variável lingüística** é uma variável capaz de armazenar um valor lingüístico.

Por exemplo, *velocidade* é uma variável lingüística capaz de armazenar valores (baixo,  $F_{baixo}$ ), (médio,  $F_{medio}$ ) e (alto,  $F_{alto}$ ).  $F_{baixo}$ ,  $F_{medio}$  e  $F_{alto}$  são conjuntos difusos.

### Suporte

Suporte ou região de influência de um conjunto  $F$  de  $X$  é o subconjunto de  $X$  para qual  $\mu_F(X) > 0$ . Já a tolerância de  $F$  corresponde ao intervalo em  $X$  onde  $\mu_F(X) = 1$ .

A Figura 2.11 mostra o conjunto de temperaturas típicas para o verão de Porto Alegre, presente no universo de discurso Temperatura. Além disso, nela também é possível observar os conceitos de suporte e tolerância.

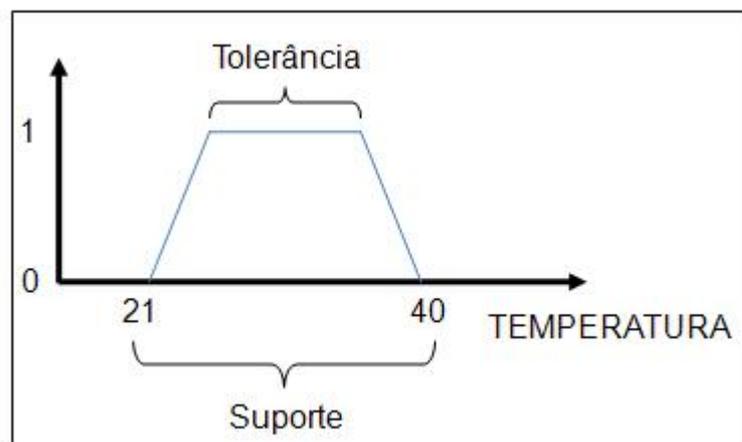


Figura 2.11: Temperaturas típicas para o verão de Porto Alegre

### Conjuntos normais ou subnormais

Um conjunto difuso  $F$  é dito normal se  $\max_{x \in X}(\mu_F(X)) = 1$  do contrário,  $F$  é considerado subnormal.

A Figura 2.12 apresenta exemplos de conjunto normal (a) e subnormal (b).

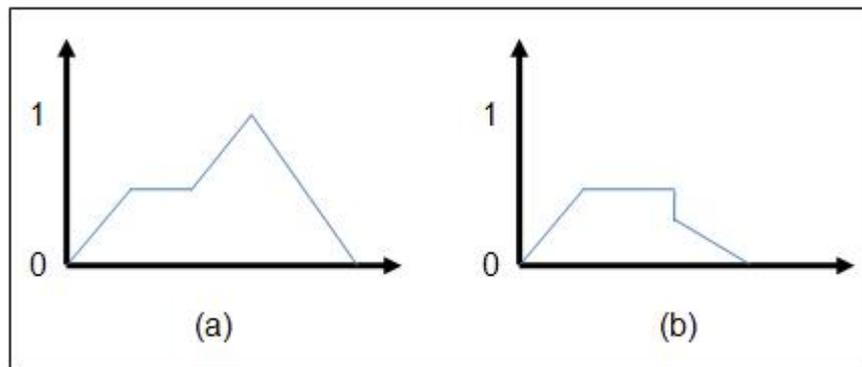


Figura 2.12: Conjunto difusos normal e subnormal

### 2.5.1 Operações com conjuntos difusos

Assim como nos conjuntos convencionais, também é possível realizar operações em conjuntos difusos. As operações mais comuns são a união ( $\cup$ ), intersecção ( $\cap$ ) e o complemento ( $\bar{\cdot}$ ). Estas operações são detalhadas abaixo.

#### União

Sejam  $A$  e  $B$  dois subconjuntos difusos de  $X$ . Sua união é um subconjunto difuso  $A \cup B$  definido por

$$(A \cup B)(x) = \max(A(x), B(x)) = A(x) \vee B(x), \forall x \in X$$

Onde  $\vee$  representa uma disjunção lógica.

A Figura 2.13 (a) mostra um exemplo de união de conjuntos difusos.

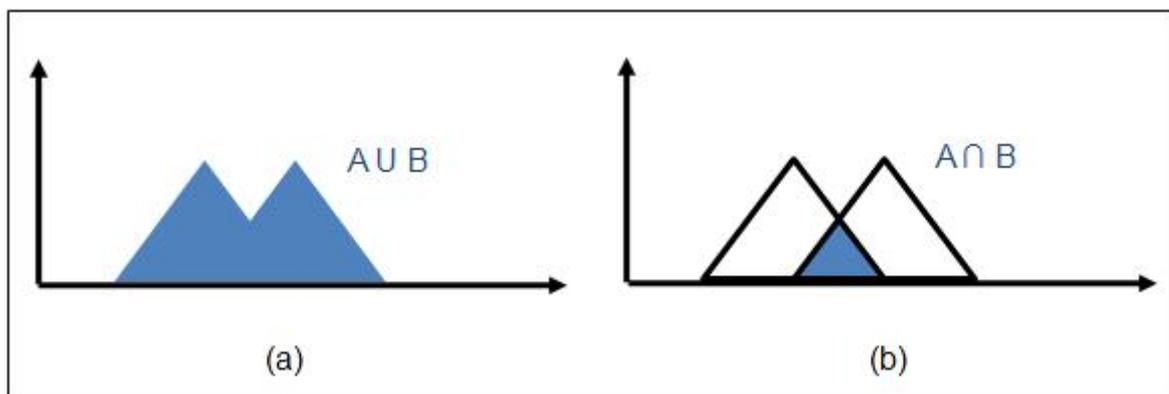


Figura 2.13: Exemplo de união e intersecção em conjuntos difusos

#### Intersecção

Sejam  $A$  e  $B$  dois subconjuntos difusos de  $X$ . Sua intersecção é um subconjunto difuso  $A \cap B$  definido por

$$(A \cap B)(x) = \min(A(x), B(x)) = A(x) \wedge B(x), \forall x \in X$$

Onde  $\wedge$  representa uma conjunção lógica.

A Figura 2.13 (b) ilustra uma intersecção de conjuntos difusos.

### Complemento

Seja  $A$  um conjunto difuso de  $X$ . O complemento ou negação de  $A$ , denotado por  $neg(A)$  é o conjunto difuso definido por

$$neg(A) = X - A \text{ ou } (neg(A)) = 1 - A(x), \forall x \in X$$

A Figura 2.14 apresenta o conjunto difuso  $A$  (direita) e seu complemento (esquerda)

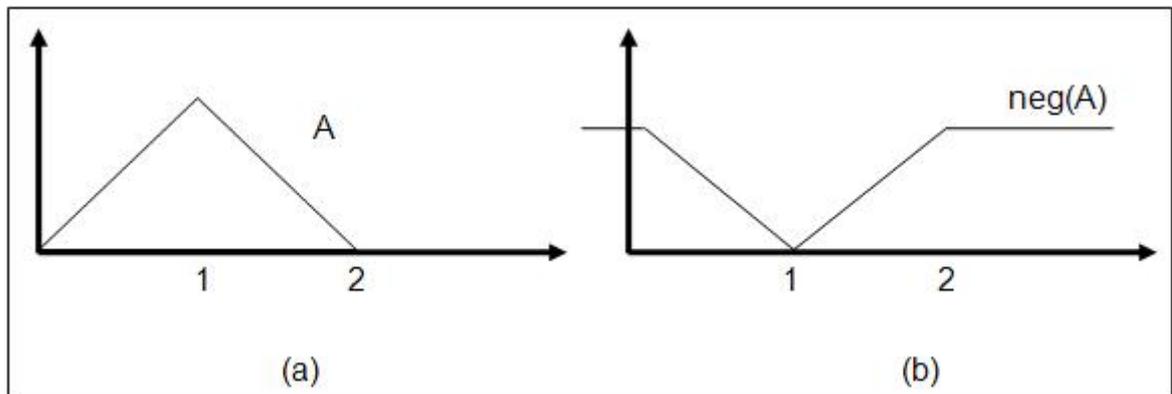


Figura 2.14: Conjunto difuso  $A$  e seu complemento ( $neg(A)$ )

## 2.5.2 Lógica Difusa

Esta seção aborda temas da lógica difusa, tratando de inferência difusa e defuzzificação. Segundo Gottwald, a lógica difusa passou por muitas mudanças e, atualmente, encontra-se relacionada com a lógica multivalorada, raciocínio aproximado e a teoria dos conjuntos difusos (GOTTWALD, 1993).

### 2.5.2.1 Implicação Difusa e Inferência Difusa

A inferência difusa é a computação do valor verdade da premissa e a implicação da sua conclusão para a regra de inferência difusa (CECHIN, 1998).

Assim como a implicação na lógica proposicional, a implicação difusa (denotada por  $A \xrightarrow{FUZ} B$ ) define as possíveis combinações de valores verdade para três proposições: A, B e  $A \xrightarrow{FUZ} B$ . A diferença entre a implicação difusa e a implicação da lógica proposicional reside no domínio dos valores usados nas proposições: os valores das implicações difusas são contínuos, assim, sua avaliação torna-se arbitrária, pois não existe um consenso sobre que valores verdade devem realmente ser usados (CECHIN, 1998).

A Tabela 2.1, retirada de (CECHIN, 1998) apresenta cinco definições de implicações difusas e suas respectivas fórmulas.

Nome	Representação	Fórmula
Implicação Zadeh	$\xrightarrow{ZAD}$	$\max(\min(x,y), 1-x)$
Implicação Mamdani	$\xrightarrow{MAM}$	$\min(x,y)$
Implicação Lukasiewics	$\xrightarrow{LUK}$	$\min(1, 1 - x + y)$
Implicação Larsen	$\xrightarrow{LAR}$	$x \cdot y$
Implicação Gödel	$\xrightarrow{GÖD}$	$\begin{cases} 1, & \text{se } x \leq y \\ 0, & \text{caso contrário} \end{cases}$

Tabela 2.1: Implicações difusa

### 2.5.2.2 Defuzificação

O processo de defuzificação é a transformação do espaço de um conjunto difuso para um valor abrupto (*crisp value*), ou seja, um valor real (PATYRA; MLYNEK, 1996).

O processo de defuzificação é considerado computacionalmente caro, isto é, a sua complexidade  $O(n)$  é geralmente muito alta. Esta afirmativa é válida para todos os métodos de defuzificação, devido a necessidade de se calcular uma integral sobre a área do gráfico da função de pertinência (CECHIN, 1998).

Sejam  $F$  um conjunto difuso ordenado  $X = \{x_1, \dots, x_i, \dots, x_n\}$ , com função de pertinência  $\mu_F$  e  $Y$  o conjunto difuso ordenado  $\{y_1, \dots, y_i, \dots, y_n\} = \{x_i : \mu_F(x_i) = \max_{x \in X}(\mu_F(x))\}$ , os métodos de defuzificação mais utilizados são (HALGAMUGE; GLESNER, 1994):

- **Maximum (MAX):**  $x^{(MAX)} = x_i : \mu_F(x_i) = \max_{x \in X}(\mu_F(x))$ . Este método é utilizado quando existe apenas um ponto onde  $\mu_F$  é máximo.

- **Mean of Maxima (MOM):**  $x^{(MOM)} = \frac{1}{m} \sum_{j=1}^m y_j$ . Este método geralmente é utilizado quando a condição para o uso do método *Maximum* não for satisfeita.
- **Local Mean of Maxima (LMOM):**  $x^{(LMOM)} = \mu_F^{-1}(\mu)$ , onde  $\mu \in [0, 1]$  é o valor de pertinência da premissa da regra.
- **Center of Gravity (COG):**  $x^{(COG)} = \frac{\sum_{i=1}^n \mu_F(x_i) x_i}{\sum_{i=1}^n \mu_F(x_i)}$ . Este é o método de defuzzificação mais comumente utilizado em controle difuso (WATANABE; WAYNE; YOUNT, 1990).
- **Midpoint of Area (MOA):**  $x_{i'} : \sum_{i=1}^{i'} \mu_F(x_i) = \sum_{i=i'}^n \mu_F(x_i)$ . Este método é recomendável somente se uma boa discretização dos valores for possível. O método garante a defuzzificação produzindo como resultado um valor dentro do suporte do conjunto difuso.
- **Center of Mean (COM):**  $x^{(COM)} = y_{j'} : \sum_{j=1}^{j'} \mu_F(y_j)$ . Assim como COG, este método também necessita de uma boa discretização.

A Figura 2.15 apresenta uma comparação dos métodos de defuzzificação acima descritos (CECHIN, 1998).

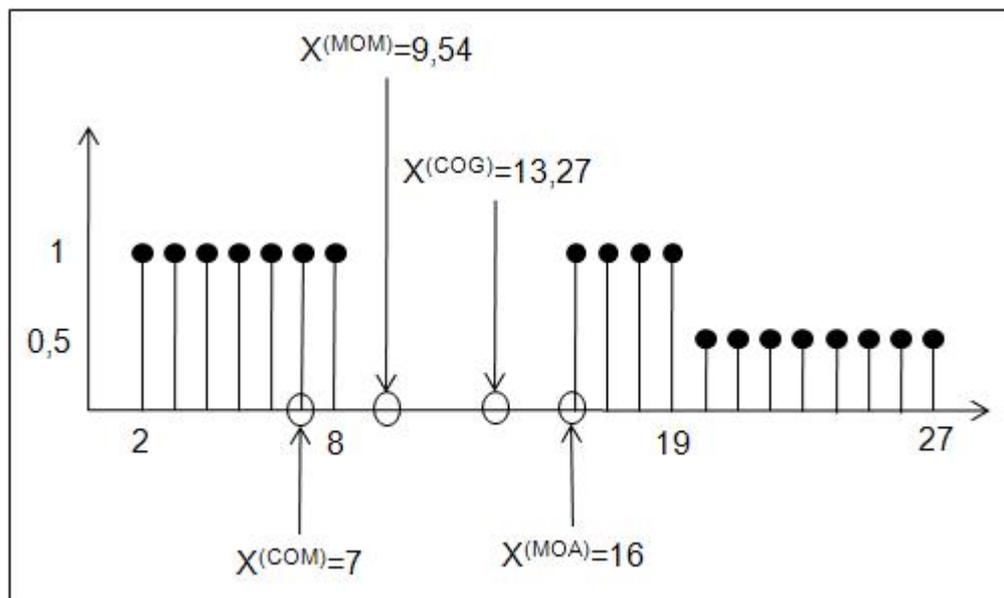


Figura 2.15: Comparação de diversos métodos de defuzzificação

### 2.5.3 Sistemas de inferência difusa

Esta seção apresenta uma descrição de dois sistemas de inferência difusa (*Fuzzy Inference Systems - FIS*) largamente utilizados na área de controle difuso: os sistemas de inferência Mamdani e Sugeno. Os dois sistemas de inferência se diferem na forma como os mesmos

apresentam o resultados de suas operações (CECHIN, 1998): o sistema Mamdani mapeia os conjuntos difusos em outros conjuntos difusos, ou seja, a parte conseqüente da regra é apresentada como um conjunto difuso. Por outro lado, o sistema Sugeno mapeia os conjuntos difusos em uma equação linear, sendo esta a parte conseqüente da regra.

Um sistema de inferência difusa é composto, basicamente, por três módulos (JANG; SUN; MIZUTANI, 1997): uma base de regras, responsável por armazenar as regras difusas, uma base de dados onde encontram-se as funções de pertinência para cada um dos conjuntos definidos na base de regras e uma máquina de inferência, cujo objetivo é raciocinar sobre as regras e fatos conhecidos do sistema e apresentar as conclusões derivadas do processo de raciocínio difuso.

### Sistema de Inferência Mamdani

Um Sistema de Inferência do tipo Mamdani, contendo duas entradas  $x_1$  e  $x_2$ , e uma saída  $y$  apresenta regras difusas com o seguinte formato (CECHIN, 1998):

$$IF x_1 \in F_1 AND x_2 \in F_2 THEN y = F \quad (2.12)$$

onde  $F_1$ ,  $F_2$  e  $F$  são conjuntos difusos. Suas funções de pertinência possuem geralmente formato triangular ( $\mu_{F_1}(x_1) = 1 - \min(k_1|x_1 - k_2|, 1)$ ) onde  $k_1$  define o tamanho do lado e  $k_2$  a altura do triângulo. As funções de pertinência de um sistema de inferência Mamdani devem cobrir todo o espaço de entradas possíveis, ou pelo menos aquelas passíveis de ocorrer na prática (CECHIN, 1998).

A operação da lógica difusa *AND* é computada através do operador *min*. Esta operação pode ser vista como a intersecção de dois conjuntos difusos, ou pelo produto cartesiano entre os mesmos conjuntos. A composição é tratada usando-se o operador *max*. Por esta razão, o algoritmo de inferência difusa do tipo Mamdani é também conhecido como *max-min Rule*.

Como o resultado final da inferência difusa do tipo Mamdani é um conjunto difuso, ainda é necessário obter um valor real para o resultado a partir da função de pertinência do conjunto, com o auxílio de um método de defuzificação.

### Sistema de Inferência Sugeno

Um sistema de inferência do tipo Sugeno, com variáveis de entrada  $x_1, \dots, x_n$  e com  $y$  sendo a variável de saída apresenta regras com o seguinte formato (TAKAGI; SUGENO, 1985):

$$IF f(x_1 \text{ is } F_1, \dots, x_i \text{ is } F_i, x_n \text{ is } F_n) THEN y = g(x_1, \dots, x_n) \quad (2.13)$$

onde  $F_1 \dots, F_n$  são conjuntos difusos representando a região do espaço de entrada onde a regra é válida,  $f$  é a função lógica (AND, OR) responsável por conectar as proposições lógicas ( $x_1 \text{ is } F_1$ )

na premissa da regra e  $g$  é a função responsável por calcular o valor de  $y$  quando a premissa é satisfeita.

Embora a função  $g$  tenha sido apresentada acima de modo genérico, Takagi e Sugeno sugerem o uso de uma função linear do tipo  $g(x_1, \dots, x_n) = k_0 + k_1x_1 + \dots + k_ix_i + k_nx_n$  onde  $k_i$  são constantes (CECHIN, 1998).

A função  $f$  pode ser implementada com o operador *min*. Uma vez que o resultado da inferência é um valor real (e não um conjunto difuso como no sistema Mamdani), este valor pode ser operado com a pertinência da premissa da regra através da implicação do tipo Larsen. A operação de composição pode ser implementada com o uso de uma soma normalizada:

$$y = \frac{\sum_{j=1}^m \mu_j \cdot y_j}{\sum_{j=1}^m \mu_j}$$

onde  $\mu_j$  é o valor de pertinência da premissa da  $j$ -ésima regra,  $y$  é o resultado da equação para a parte conseqüente da  $j$ -ésima regra e  $m$  é o número de regras presentes no sistema de inferência. Como a inferência é implementada com o uso da multiplicação (inferência do tipo Larsen) e a composição com o uso de uma soma, este algoritmo de raciocínio também é conhecido como *sum-prod rule* (CECHIN, 1998).

## 2.6 Extração de Regras a Partir de Redes Neurais Artificiais

O principal motivo para a adoção da tecnologia de Redes Neurais Artificiais em segmentos tão diversos como academia, indústria e comércio é a sua habilidade de aproximar virtualmente qualquer função existente, com base nos dados submetidos a ela durante a fase de treinamento. A essa capacidade é dado o nome de generalização, ou seja, a habilidade de prever o resultado de um determinado fenômeno, com parâmetros nunca antes vistos, a partir de parâmetros analisados anteriormente. Este alto poder de generalização, no entanto, possui um preço: a inabilidade inerente das redes neurais de explicar ao usuário o conhecimento por ela adquirido.

Regras podem ser utilizadas para descrever o conhecimento da rede neural para o usuário. Como mencionado anteriormente, as regras possuem uma estrutura do tipo *SE premissa => ENTÃO conseqüência*, onde a premissa descreve os dados de entrada da rede neural, e a conseqüência descreve a saída a ser apresentada pela rede, caso a premissa seja verdadeira. A maioria dos sistemas de extração de regras atuais usa a Teoria dos Conjuntos Difusos (PATYRA;

MLYNEK, 1996) para compor a parte de premissa da regra (CECHIN, 1998), (BENITEZ; CASTRO; REQUENA, Sep 1997) e (JANG, Sep 1992).

Andrews *et alli* citam, entre outras, as seguintes qualidades relacionadas às redes transparentes, ou seja, redes neurais capazes de demonstrar seu conhecimento para o usuário (ANDREWS; DIEDERICH; TICKLE, 1995):

- capacidade explanatória: dentro da IA simbólica, explicação tem como significado uma estrutura onde, internamente, possui a capacidade de raciocínio e aprendizado e, externamente, a capacidade de explicar os resultados para o usuário. Dessa forma, uma rede neural transparente possui a capacidade de ser utilizada em conjunto com sistemas inteligentes simbólicos, bem como oferecer a oportunidade de usuários inexperientes com o sistema aprenderem com o próprio produto;
- possibilidade de utilização em aplicações críticas: a capacidade da rede neural explicar ao usuário o motivo pelo qual ela apresentou determinada resposta, possibilita a aplicação de RNAs em aplicações críticas como controle de tráfego aéreo e gerenciamento de estações de fornecimento de energia. Isso se torna possível pois permite ao usuário cancelar a ação da rede neural, caso a mesma apresente uma saída errônea;
- auxílio na formulação de teorias científicas: as RNAs possuem alta capacidade de exploração de dados, conseguindo capturar os padrões inerentes aos mesmos com notável precisão. No entanto, uma rede neural usual mantém esse importante conhecimento armazenado em sua estrutura interna, codificado como vetores de pesos. Métodos de extrações de regras podem ajudar os cientistas a capturar esses padrões interessantes e, de posse desse conhecimento, criar ou validar teorias científicas.

### 2.6.1 Separação do Espaço de Dados de Entrada

Nessa seção, serão apresentadas técnicas de extração de regras a partir de redes neurais treinadas. Primeiramente, é necessária a apresentação dos métodos de separação do espaço de dados de entrada e o seu agrupamento em conjuntos difusos. Esta tarefa é importante pois resulta diretamente na construção da premissa das regras extraídas das redes neurais transparentes.

Abaixo são descritos os métodos de separação do espaço de dados de entrada: partição em grade (*grid partition*), partição em árvore (*tree partition*), partição em espalhamento (*scatter partition*) e partição linear (*linear partition*). Uma representação gráfica dos mesmos é

apresentada na Figura 2.16.

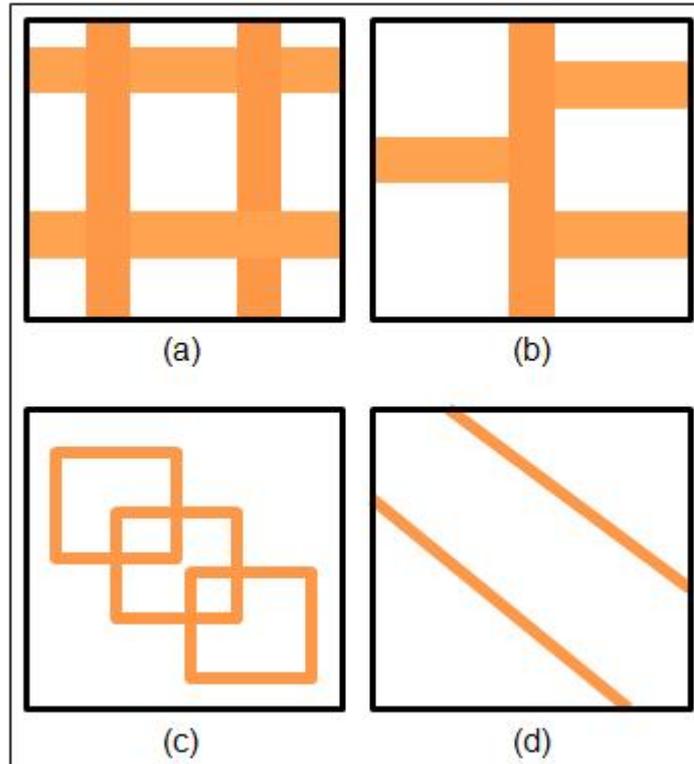


Figura 2.16: Métodos para particionar o espaço de entrada: grade (a), árvore (b), espalhamento (c) e linear (d)

O particionamento em grade é o método mais comumente utilizado. Nele os dados de cada variável de entrada são divididos em conjuntos difusos com valores lingüísticos *baixo*, *médio* e *alto*. Após os conjuntos são combinados através da operação de intersecção difusa. A desvantagem desse método é a consequência do aumento exponencial do número de conjuntos. A quantidade de conjuntos difusos corresponde à quantidade de regras geradas pelos métodos de extração. Esta é a principal métrica para avaliar a capacidade de compreensão de um sistema de inferência difusa.

O particionamento em árvore e espalhamento resultam em regras altamente compreensíveis, embora também apresentam desvantagens: dependendo da aplicação, ou da distribuição dos dados no espaço, pode ser necessário um grande número de partições para cobrir todos os dados de treinamento. Isso pode ocorrer, por exemplo, quando os dados estão distribuídos diagonalmente ao longo do espaço, ou quando existem dependências (correlação) entre as variáveis de entrada. Como pode ser observado na Figura 2.16, a partição em espalhamento gera partições com a desvantagem de não serem mutuamente exclusivas.

A partição linear gera o menor número de regras. No entanto, as funções de pertinência geradas por este método são geralmente de difícil compreensão. Abaixo são descritos

os métodos de extração de regras mais comumente utilizados em projetos científicos.

## 2.6.2 ANFIS

Jang (JANG, Sep 1992) apresenta o sistema ANFIS (*Adaptive-Network-Based Fuzzy Inference System*). Trata-se de um sistema baseado em controlador difuso adaptativo, onde o controlador pode ser treinado através de um algoritmo *backpropagation* modificado.

As modificações no algoritmo de treinamento podem ser analisadas nas equações abaixo:

$$\Delta w = -\eta \frac{\partial E}{\partial w} \quad (2.14)$$

$$\eta = \frac{k_0}{\sqrt{\sum_w \left(\frac{\partial E}{\partial w}\right)^2}} \quad (2.15)$$

Onde  $k_0$  é a transição de gradiente presente no espaço dos parâmetros. A última fórmula mostra uma adaptação na taxa de aprendizagem do algoritmo, permitindo um passo constante dentro do espaço de parâmetros do mesmo. Esta modificação visa corrigir o problema inerente às funções de ativação que trabalham no ponto onde a derivada da ativação é próxima de zero.

O controlador difuso extrai regras do tipo mamdani e é composto por cinco camadas, como pode ser observado na Figura 2.17. A primeira camada realiza o cálculo das funções de pertinência da premissa da regra. Estas geralmente têm o formato de sino com tolerância, suporte e centro adaptáveis.

A segunda camada calcula a intersecção dos conjuntos difusos presentes na regra. Esse cálculo é implementado através da operação de multiplicação de conjuntos difusos. A terceira camada normaliza os valores de pertinência dos conjuntos da premissa da regra. A quarta camada define a consequência da regra através da multiplicação (inferência difusa) dos valores de pertinência obtidos na terceira camada pelo resultado das equações lineares encontradas na quarta camada. As equações lineares possuem coeficientes adaptáveis multiplicados pelas variáveis de entrada do sistema. Os resultados da inferência são combinados na camada 5.

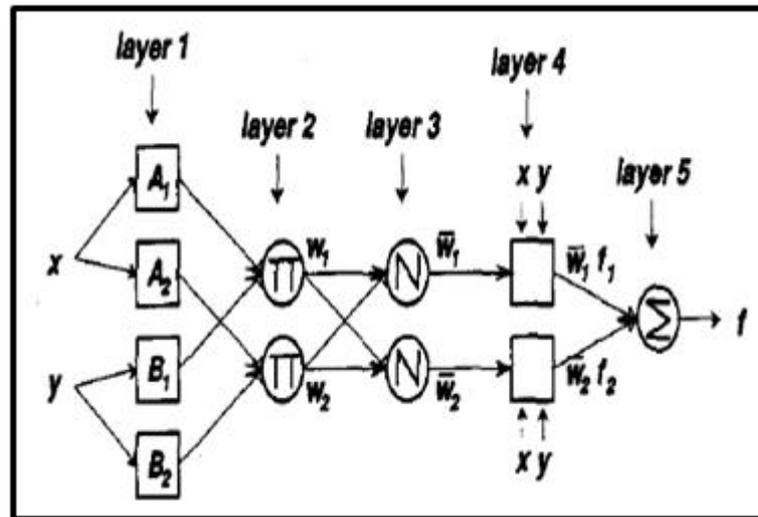


Figura 2.17: Estrutura do ANFIS

### 2.6.3 FNES

Hayashi desenvolveu o FNES (*Fuzzy Neural Expert System*) (HAYASHI; IMURA, 3-5 Dec 1990). Diferentemente da maioria dos sistemas de extração de regras encontrados na literatura, o FNES é capaz de extrair regras difusas automaticamente a partir de uma rede neural treinada com o algoritmo *backpropagation*.

A rede neural usada no FNES é composta de três camadas, conectadas por pesos variando dentro do intervalo  $[-10, +10]$ . É possível a utilização de conexões entre os neurônios da camada de entrada e de saída (conexões de atalho - *shortcut connections*). O esquema da rede neural do FNES é apresentado na Figura 2.18.

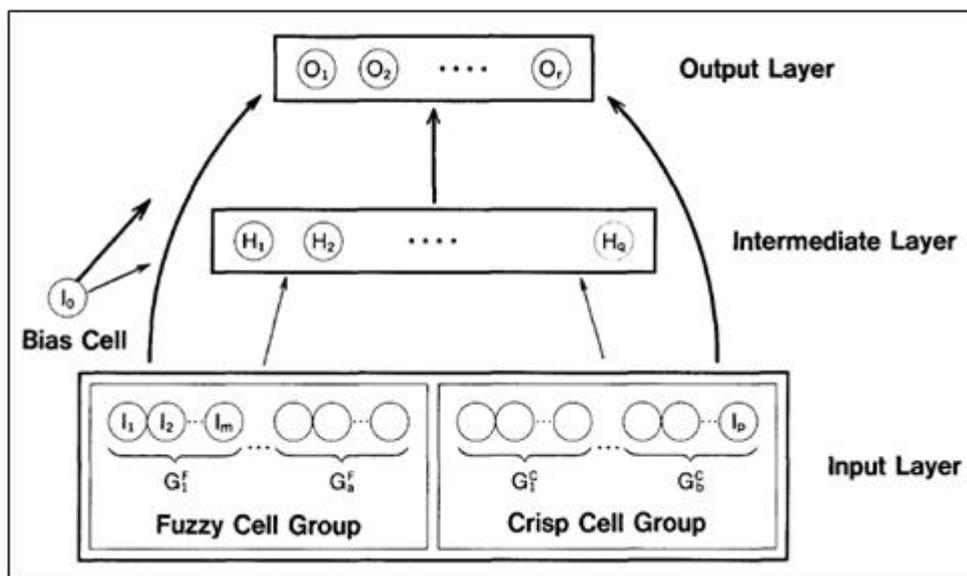


Figura 2.18: Rede neural usada pelo sistema FNES

Ao contrário de uma rede neural comum, os neurônios da camada escondida da rede FNES executam a seguinte equação:

$$a_u = \begin{cases} +1, & net_u > 0 \\ 0, & net_u = 0 \\ -1, & net_u < 0 \end{cases} \quad (2.16)$$

onde  $net_u = \sum_{i:arc(i,u)} a_i$ . É importante destacar a existência de um significado para cada um dos valores possíveis: +1 verdadeiro, 0 desconhecido e -1 falso. Essa característica permite interpretar a rede neural como regras existentes em um sistema especialista.

Como todas as entradas para a rede são do tipo booleano, uma codificação especial é usada para representar valores quantitativos (valores difusos). A Figura 2.18 mostra as entradas para a rede neural separadas em dois grupos: valores abruptos e difusos. Por exemplo, a entrada de valor 3, se codificada em cinco entradas difusas, teria o formato [+1, +1, +1, -1, -1]. Sendo estas cinco unidades de entrada os elementos do grupo de valores difusos da rede.

Para extrair regras da rede neural, é realizada a variação de um grupo de entrada por vez, mantendo os demais grupos em 0 (desconhecido). Com essa abordagem, é possível descobrir a influência das diversas entradas na saída da rede e, dessa forma, obter regras mais simples. O FNES demonstra claramente sua predileção por problemas de classificação. Por outro lado, também é notável a existência de desvantagens como a necessidade de um grande número de neurônios na camada de entrada da rede para permitir a codificação dos valores difusos.

#### 2.6.4 NEFCON

Nauck *et alli* (NAUCK; KLAWONN; KRUSE, 1994) criaram o NEFCON-Model (Neuro-Fuzzy-Controller-Model) para calcular os parâmetros do sistema de inferência difusa e, além disso, obter regras difusas. As regras obtidas são do tipo Mandani e possuem o formato descrito abaixo:

*IF*  $x_1$  *IS*  $F_1$  *AND*  $x_2$  *IS*  $F_3$  *THEN*  $y$  *IS*  $G_1$

*IF*  $x_1$  *IS*  $F_2$  *AND*  $x_2$  *IS*  $F_3$  *THEN*  $y$  *IS*  $G_2$

Onde  $F_i$  e  $G_i$  são conjuntos difusos.

Neste trabalho, os autores apresentaram duas formas de implementação da camada de

saída da rede: (i) o processamento de conjuntos difusos através de *t-norm* e *t-conorm* seguido da defusificação do resultado; e (ii) o processamento das funções de pertinência através do método proposto por Berenji em (BERENJI; KHEDKAR, 1992).

A aprendizagem é dividida em duas partes: aprendizagem das funções de pertinência e aprendizagem das regras. Para isso, os estados "bom" ( $H^{(opt)}$ ) e "próximo de bom" ( $H^{(komp)}$ ) devem ser definidos. O uso destes estados pode ser exemplificado através de um sistema de controle de temperatura: a temperatura de 20°C pertence ao estado "bom", enquanto uma temperatura menor, apresentando um aumento ao longo do tempo, pertenceria ao estado "próximo de bom". Esta definição é implementada através da avaliação da função de pertinência do estado do processo  $H^{(opt)}(s)$ . O estado atual é bom caso ( $H^{(opt)}$ ) esteja próximo de 1, e "próximo de bom" caso ( $H^{(opt)}$ ) resulte em um valor próximo de 0.

Uma combinação dos dois conjuntos difusos apresentados acima é realizada para a geração de um terceiro conjunto difuso  $H$  e uma função de erro  $E(s) = 1 - H(s)$ . A combinação dos conjuntos difusos é definida pelo usuário e depende do conhecimento já existente a respeito do problema em questão. Além disso, também é necessário certo conhecimento sobre o sinal de controle a ser usado no sistema.

O aprendizado das regras começa com a combinação de todas as funções de pertinência da premissa. É necessária uma definição do número de funções de pertinência a serem usadas para cada uma das entradas presentes na rede. Durante a fase de aprendizado, as combinações são verificadas e excluídas caso algum critério previamente definido não tenha sido alcançado.

## 2.6.5 FUN

O FUN (*Fuzzy Net*) possui uma arquitetura especial de rede neural. Esta estrutura é gerada a partir de regras iniciais e das funções de pertinência dos conjuntos difusos informados a priori pelo usuário. Além disso, o usuário também precisa informar ao sistema a quantidade de regras a serem extraídas da estrutura. Este último parâmetro definirá a quantidade de neurônios na camada escondida da rede.

O aprendizado desse sistema é baseado em uma função objetivo, a qual deve ser minimizada durante o treinamento. Uma possível função objetivo é apresentada na equação abaixo:

$$E = \sum_{i=1}^m \left| s_i - s_i^{(d)} \right|^2 \quad (2.17)$$

onde  $m$  é o número de passos considerados,  $s_i$  é o estado atual do sistema e  $s_i^{(d)}$  é o estado desejado para o sistema. Destaca-se que a função objetivo é usada apenas para testar o desempenho da rede, e não como seria utilizada em um treinamento com o algoritmo *backpropagation* para determinar os pesos da rede neural. Dessa forma, a rede neural fica livre da necessidade de produzir valores abruptos e usar operadores deriváveis, além de permitir o uso de regras difusas do tipo Mandami.

O aprendizado das regras pode ser considerado uma busca estocástica. As conexões são modificadas e o desempenho do sistema é medido com o uso da função objetivo. As modificações são mantidas caso apresentem uma melhora em relação à configuração anterior (a função objetivo foi reduzida), ou descartadas caso contrário.

As funções de pertinência possuem um formato triangular, e seu aprendizado segue o seguinte algoritmo: uma função de pertinência é selecionada aleatoriamente e alterada. Caso o desempenho do sistema melhore, a próxima alteração da função de pertinência será na mesma direção. Caso contrário, a modificação da função de pertinência será realizada em outra direção. A alteração realizada deve ser suficientemente pequena para permitir a convergência do algoritmo de aprendizagem.

### 2.6.6 Benítez

Benítez *et alli* propuseram um sistema para extração de regras baseado no conhecimento armazenado nos neurônios presentes na camada escondida da rede neural. O sistema extrai regras difusas para um par de neurônios da camada escondida e da camada de saída,  $(z_i, y_k)$  modeladas através de

$$R_{jk} : \text{If } \sum_{i=1}^n x_i w_{ij} + \tau_j \text{ is } A \text{ then } y_k = \beta_{jk} \quad (2.18)$$

onde  $A$  é um conjunto difuso em  $\mathfrak{R}$  cuja função de pertinência é igual à função de ativação dos neurônios presentes na camada escondida da rede neural.

A saída para o sistema caracteriza um vetor com os componentes:

$$y_k = \sum_{j=1}^n A \left( \sum_{i=1}^n x_i w_{ij} + \tau_j \right) \beta_{jk} \quad (2.19)$$

Com base na análise das equações acima, é possível notar que a saída do sistema (o conseqüente da regra) corresponde exatamente à saída produzida pelo neurônio oculto atualmente analisado.

O conjunto difuso  $A$  deve ser entendido como "maior ou menor que aproximadamente

$r$ ”, sendo  $r$  definido como 0,9. A escolha desse valor para  $r$  pode ser explicada pela própria natureza da curva sigmóide, pois esta permite os valores 0 e 1 apenas assintoticamente. Dessa forma, na literatura sobre redes neurais, atribui-se valores 0,1 e 0,9 para ausência ou total ativação do neurônio analisado, respectivamente.

Para tornar as regras extraídas mais compreensíveis, uma decomposição da premissa das regras é aplicada. Isso é particularmente difícil de se obter em redes neurais, pois o conhecimento sobre as variáveis de entrada da rede é armazenado de forma homogênea pelos neurônios da camada escondida. Esta característica aglutinadora dos neurônios ocultos prejudica a análise individual das variáveis de entrada em termos de como cada uma delas afeta a saída da rede. Uma abordagem coerente de decomposição seria apresentar cada uma das premissas ligadas através do conectivo lógico *AND*. Infelizmente, esta abordagem foi descartada pelos autores com base nas conclusões de Yager (YAGER, 1991).

De forma a possibilitar a decomposição das regras, respeitando as capacidades dos neurônios da camada escondida, os autores criaram o conector lógico *i-or* (*interactive-or* - ou interativo). O *i-or*, representado por \*, opera da seguinte forma:

$$a_1 * a_2 * \dots * a_n = \frac{a_1 \cdot a_2 \cdot \dots \cdot a_n}{(1 - a_1) \cdot (1 - a_2) \cdot \dots \cdot (1 - a_n) + a_1 \cdot a_2 \cdot \dots \cdot a_n} \quad (2.20)$$

onde  $a_i$  equivalem aos valores de entrada para a rede neural, ou seja, a premissa da regra.

O resultado da aplicação do *i-or* para a decomposição das premissas pode ser visto na equação abaixo

$$R_{jk} : \text{If } x_1 \text{ is } A_{jk}^1 \text{ i-or } x_2 \text{ is } A_{jk}^2 \text{ i-or...i-or } x_n \text{ is } A_{jk}^n \text{ THEN } y_k = \beta_{jk} \quad (2.21)$$

onde  $A_{jk}^i$  são conjuntos difusos obtidos a partir de  $A$  (a função de ativação do neurônio), dos pesos da rede  $w_{ij}$  e do *bias*  $\tau_j$ .

A Figura 2.19 apresenta regras obtidas com o uso desse sistema a partir de uma rede neural hipotética.

```

IF x1 < approximately 5,447 i-or
  x2 < approximately 4,118 i-or
  x3 > approximately 3,567 i-or
  x4 > approximately 2,61 i-or
THEN y = 1,227

```

Figura 2.19: Regra extraída para uma rede neural hipotética

Uma consideração é necessária com relação aos pesos da rede neural. O sinal do peso

é responsável por identificar o sinal de maior ou menor presente nas regras. Um peso com sinal negativo faz a premissa da regra expressar o sentido de "menor que aproximadamente". Já um sinal positivo apresenta o sentido de "maior que aproximadamente". Na Figura 2.19, as componentes da premissa com valores de  $x_1$  e  $x_2$  foram derivadas a partir de pesos negativos, de forma análoga, os valores de  $x_3$  e  $x_4$  mostram que os pesos conectados ao neurônio da camada escondida são positivos.

### 2.6.7 FAGNIS

Cechin (CECHIN, 1998) desenvolveu o FAGNIS para extrair regras do tipo Sugeno a partir de redes neurais treinadas. Em seu trabalho, Cechin apresenta três metodologias diferentes para o processo de extração de regras.

Embora o significado armazenado nos neurônios presentes na camada escondida de uma rede neural seja ainda tratado como um problema em aberto pela comunidade científica, é possível notar que este comportamento depende principalmente da finalidade da rede neural em questão. Os neurônios ocultos de uma rede neural de classificação tendem a usar os pontos extremos da função sigmóide (-1) e (+1). Já em tarefas de aproximação os neurônios ocultos tendem a ocupar a parte central da função sigmóide, onde a mesma é praticamente linear.

O FAGNIS usa as características apresentadas no parágrafo acima para decompor a função de uma rede neural em partes lineares, representando o estado atual da rede através de uma declaração do tipo "está" ou "não está" em uma determinada parte da função de ativação da rede. A rede neural passa então a ser caracterizada pelo grau de veracidade relacionado à declaração "está" ou "não está" presente em uma parte linear. O grau de veracidade é calculado através da multiplicação (intersecção de conjuntos difusos) do grau de veracidade das declarações. Por exemplo, "a unidade não-linear  $u$  está presente na parte  $i$ ".

A principal idéia do FAGNIS é decompor a curva sigmóide em três partes, como ilustrado pela Figura 2.20 (a). Cada uma dessas partes torna-se um conjunto difuso, responsável por caracterizar a parte antecedente das regras a serem extraídas. Cada um dos dados usados no treinamento da rede é então associado a um dos conjuntos difusos, com base na região da sigmóide ativada por ele quando propagado adiante na rede neural. Uma vez que todos os dados de treinamento estejam vinculados à um conjunto difuso, é possível definir a saída da rede neural para cada conjunto. A saída é caracterizada por uma equação linear, sendo ela dependente dos valores  $x_i$ , as entradas da rede neural.

Abaixo é apresentado um conjunto de regras fictícias extraídas pelo FAGNIS a partir de

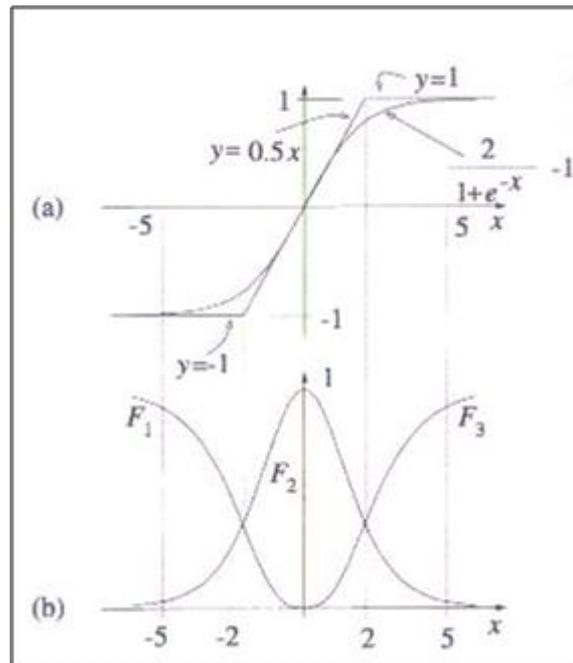


Figura 2.20: Função sigmóide e parte conseqüente das regras difusas (a); função de pertinência ideal para cada região da função sigmóide (b)

uma rede neural com duas entradas:

$$\text{IF } (x_1, x_2) \text{ IS } G_1 \text{ THEN } y = -1.04x_1 - 1.04x_2 - 0.17$$

$$\text{IF } (x_1, x_2) \text{ IS } G_2 \text{ THEN } y = -1.04x_1 - 1.04x_2 + 2.04$$

onde  $G_1$  e  $G_2$  são conjuntos difusos representando diferentes regiões da curva sigmóide.

## 2.7 Conclusões da Fundamentação Teórica

Esta seção objetiva realizar uma análise sobre os tópicos abordados durante a revisão teórica presente nesse documento. Mais especificamente, esta seção concentra-se nas lições aprendidas durante os estudos dos métodos de extração de regras a partir de redes neurais. Este conhecimento será necessário para a execução do estudo de caso presente no Capítulo 4 - Metodologia, onde dois métodos de extração de regras serão comparados quanto a sua eficiência.

Os dois métodos selecionados para o experimento citado acima são o sistema de Benítez e o FAGNIS. Os motivos para a escolha desses dois algoritmos estão relacionados abaixo.

1. Independência do algoritmo de treinamento: os métodos selecionados trabalham sobre as redes neurais após o treinamento das mesmas, dessa forma, o algoritmo de treinamento usado para a geração da rede neural não influi nas regras extraídas.
2. Não necessidade de arquiteturas especiais: os métodos de extração operam sobre redes neurais do tipo *multilayer perceptron*, sem exigir nenhum tipo de modificação na estrutura da rede para facilitar o processo de extração, diferentemente dos demais métodos apresentados.
3. Comparação mais justa: como ambos os métodos possuem todas as características citadas acima, a mesma rede neural pode ser fornecida para os dois algoritmos de extração. Isso permite uma avaliação mais justa do desempenho dos dois algoritmos.

É possível notar que os métodos selecionados para a condução dos trabalhos dessa dissertação são relativamente antigos. Isso de forma alguma significa que o problema de extração de regras seja considerado solucionado pela comunidade científica. No entanto, existem poucos métodos de extração de regras para redes neurais de aproximação, sendo que a maioria deles exige uma rede com arquitetura especialmente projetada para o algoritmo de extração. Baesens *et alli* (BAESENS et al., 2003) apresentam métodos interessantes, entretanto, estes lidam apenas com a extração de regras a partir de redes neurais envolvidas em problemas de classificação. Na verdade, os dois métodos selecionados para este trabalho foram os únicos encontrados que possuem todas as qualidades relacionadas acima.

## 3 REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentado um panorama temático-histórico retratando trabalhos de diversos autores na área de previsão de carga. Estes trabalhos estão agrupados em seções, segundo as diversas metodologias usadas pelos autores para resolver o problema de prever a demanda por energia com a menor margem de erro possível.

### 3.1 Previsão de Carga com Redes Neurais

Ho, Hsu e Yang (HO; HSU; YANG, Feb 1992) implementaram um previsor de carga usando uma rede neural do tipo *multilayer perceptron* para a Companhia de Energia Elétrica de Taiwan. O previsor tem como objetivo prever a demanda de energia para a próxima hora. Na verdade, a principal contribuição deste trabalho foi a demonstração de um algoritmo de treinamento adaptativo para a rede neural, tornando-a capaz de convergir mais rapidamente quando comparado com o *backpropagation* tradicional.

Os dados de entrada para a rede neural de previsão consistem de dados meteorológicos e também dados estimados sobre o valor máximo e mínimo da demanda por energia do dia anterior. Estes dois últimos valores foram obtidos através do uso de uma rede neural projetada para este fim. A estrutura da rede utilizada para a previsão de demanda consiste de 46 neurônios na camada de entrada, 60 na camada escondida e um neurônio na camada de saída. Os autores concentraram grandes esforços na aplicação e teste de um algoritmo de treinamento adaptativo para a rede neural, cujo objetivo principal seria a redução do tempo de treinamento. Primeiramente, os autores realizaram diversos testes para encontrar o valor ideal para a taxa de aprendizagem. Após, o algoritmo de treinamento *backpropagation* foi modificado para variar seu valor de *momentum*, com o objetivo de acelerar o aprendizado quando a mudança dos valores dos pesos sinápticos da rede converge para o valor correto. O algoritmo aumenta o valor de *momentum* quando o erro da rede diminui de uma iteração de treinamento para outra, e reduz o *momentum* quando o contrário acontece. Foi realizada uma comparação entre o algoritmo adaptativo e o *backpropagation* tradicional. O algoritmo adaptativo apresenta melhores resultados tanto na

questão velocidade de treinamento quanto na qualidade do mesmo, mostrando um valor de erro médio menor que o da rede treinada pelo *backpropagation* tradicional.

Botha e Vermaak apresentam uma comparação entre um modelo neural tradicional (*feedforward*) e um modelo neural recorrente totalmente conectado para previsão de carga em (VERMAAK; BOTHA, Feb 1998). Nesse trabalho, o modelo neural recorrente demonstrou um desempenho sensivelmente superior ao modelo *feedforward* tradicional.

Embora as condições climáticas sejam as variáveis mais influentes na demanda por energia, o tempo também representa uma informação importante. Os autores mostram que, mesmo sendo a demanda por energia diferente em dias diferentes, em instantes de tempos iguais, a demanda por energia apresenta um comportamento semelhante para estes dois dias. Colocando esta idéia em prática, os autores treinaram uma rede neural recorrente, valendo-se de sua capacidade de usar os pontos de dados do passado para influenciar a saída atual da rede, para aproximar a demanda por energia em toda a extensão nacional da África do Sul. Para os dois tipos de modelos foram usados como variáveis independentes os valores mínimo, máximo e médio de temperatura. Além disso, também usou-se três neurônios de entrada para informar a qual dia da semana (dias úteis apenas) o dado de treinamento se refere. Como uma rede neural *feedforward* tradicional é incapaz de usar informações do passado para estimar o valor para o ponto de dado atual no treinamento, foram também informados à rede *feedforward* os valores de carga para três, duas e uma hora anteriores ao instante de previsão. Foi produzido um par de redes neurais, recorrente e *feedforward*, para cada semana do ano de 1994. Durante a condução dos experimentos, os autores verificaram que o tamanho das redes *feedforward* obtidas como modelo de previsão eram consideravelmente maiores que os das redes recorrentes. Eles apontam o tamanho compacto das redes recorrentes como o principal fator para o seu maior poder de generalização quando comparadas às redes *feedforward*.

Em (DREZGA; RAHMAN, Aug 1999) Dresga e Rahman discutem a utilização de um previsor composto de um par de redes neurais para previsão de carga em um futuro próximo (*Short-Term Load Forecasting* - STLF). Neste trabalho, duas redes neurais com *shortcut connections* são treinadas para predizer a demanda por energia para a próxima hora, próximo dia, próximos dois dias e próximos cinco dias.

Os autores optaram por separar os dados de treinamento para as redes neurais em dados referentes aos dias úteis e não úteis da semana, alocando previsores distintos para tratar cada um desses grupos de dados. Na verdade o previsor é dotado da capacidade de prever apenas a demanda por energia para a próxima hora. Previsões para um espaço de tempo maior resultam de um processo iterativo, onde a previsão para a demanda da próxima hora é incluída no conjunto

de dados de treinamento e o processo se repete até ser obtida a previsão para o número de horas desejado. A quantidade de neurônios na camada escondida de cada rede neural foi definida através de uma simulação piloto (*pilot simulation*) de onde foram selecionadas as topologias apresentando o menor valor de MAPE (*Mean Absolute Percent Error*). Como cada previsor possui duas redes neurais, a demanda por energia final é o resultado da média prevista para a demanda de cada rede neural. Essa abordagem visa reduzir o erro do previsor diante de uma variação brusca da temperatura. Segundo os autores, uma rede neural isolada não apresentaria um desempenho satisfatório ao prever demanda por energia em situações onde a temperatura sofre mudanças significativas em curtos períodos de tempo.

Charytoniuk e Chen (CHARYTONIUK; CHEN, Feb 2000) apresentaram um modelo neural para a previsão de carga para curtos períodos de tempo. Trata-se de um problema comumente referido como VSTLF (*Very Short-Time Load Forecasting*) onde, diferentemente dos modelos tradicionais cujo objetivo é prever a demanda para as próximas horas ou dias, o modelo proposto tenta prever a demanda para os próximos minutos. O objetivo neste trabalho não era propriamente a descoberta de demanda por energia. Ao invés disso, houve a preocupação em descobrir qual seria o incremento na demanda para o futuro próximo. Os autores esperavam com esta mudança de objetivo tornar seus modelos menos sensíveis aos ruídos causados pelas mudanças climáticas, reduzindo o erro das previsões realizadas pelas redes neurais.

Neste trabalho, os autores utilizaram um conjunto de oito pequenas redes neurais do tipo *feedforward* com poucos neurônios na camada escondida. Cada rede neural foi treinada para prever o incremento na demanda atual para um pequeno espaço de tempo no futuro: de 20 a 90 minutos, com incremento de 10 minutos. Essa abordagem foi adotada para possibilitar o treinamento das redes neurais apenas uma vez por dia, com dados obtidos pelo sistema de gerenciamento de energia da estação informando a quantidade de energia liberada para as unidades consumidoras a cada quatro segundos, durante sete dias. Os autores comentam que o uso de um número reduzido de neurônios na camada escondida da rede foi crucial para manter baixo o tempo de treinamento das mesmas. O conjunto de dados de entrada para as redes neurais também sofreu modificações se comparados com outros trabalhos. Embora seja comum modelar a demanda por energia em função das condições climáticas, as mesmas não foram consideradas pelos autores. Segundo eles, as variáveis climáticas não possuem uma forte influência na demanda por energia necessária para os próximos minutos.

Senjyu *et alli* (SENJYU et al., Feb 2002) utilizaram uma rede neural treinada com dados com características semelhantes ao dia alvo para a previsão de carga para a próxima hora. Com a aplicação dessa técnica, os autores buscavam utilizar uma única rede neural para realizar a

previsão de carga sobre qualquer tipo de dia - dias de semana, fins de semana, feriados, etc..

Para a seleção dos dias semelhantes a partir da base de dados, foi usada a técnica de Norma Euclidiana com Pesos Ponderados. A rede neural proposta pelos autores recebe como entrada a média da temperatura e demanda por energia de cinco dias considerados como semelhantes ao dia envolvido na previsão e as medições da temperatura do dia atual para as últimas três horas. O conjunto de dados apresentado para a rede durante o treinamento é composto dos dados de um mês atrás, a contar do dia anterior à data de previsão, e de um mês atrás e um mês a frente do mesmo dia no ano passado. Quando o dia alvo para a previsão muda, a rede neural precisa ser treinada novamente.

Tsekouras *et alli* apresenta um previsor de demanda de energia de médio prazo, para prever a demanda por energia da Grécia em (TSEKOURAS; HATZIARGYRIOU; DIALYNAS, Feb. 2006). Trata-se de uma rede neural treinada com diversos dados econômicos e climáticos do país, obtidos de diversas fontes, sendo o Sistema Público de Energia um exemplo. O modelo proposto neste trabalho foi comparado com outros 7 modelos clássicos utilizados para previsão de carga, tais como: modelo de regressão linear simples, modelos de regressão linear de 2º ordem, entre outros.

Nesse trabalho foi utilizado o conceito de tipo de região de consumo. Onde regiões de consumo podem ser do tipo residencial, comercial, rural ou industrial. Desenvolveu-se um modelo neural para cada região avaliada. Os modelos de comparação também foram elaborados especificamente para cada tipo de região de consumo. O conjunto de dados de treinamento é constituído de medições horárias de temperatura e consumo de energia. A base de dados conta com 10 anos dessas medições. Outras variáveis foram também analisadas. Na verdade, os autores não detalham no trabalho as variáveis utilizadas no conjunto de dados. Apenas citam que dados de temperatura, crescimento industrial e produção de certos materiais duráveis e não duráveis foram considerados para a elaboração dos modelos. Foi utilizada uma rede neural do tipo *multilayer perceptron* com 5 neurônios na camada escondida. Os neurônios da camada escondida usam uma função de ativação do tipo hiperbólica, diferente das redes neurais citadas nos demais trabalhos desta revisão onde são utilizadas funções do tipo logística. A rede neural foi treinada com o algoritmo *backpropagation* adaptivo, onde parâmetros como taxa de aprendizagem, *momentum* entre outros variam de uma época para outra, de acordo com o erro verificado durante o treinamento. A comparação do desempenho dos modelos foi realizada separadamente para cada agrupamento de tipo de região. O modelo neural mostrou um desempenho melhor que todos os outros modelos, em todos os grupos de regiões.

## 3.2 Integração de RNAs com Outros Métodos

Srinivasan *et alli* relatam o uso de um sistema de previsão de carga constituído de redes neurais e um sistema especialista baseado em regras difusas em (SRINIVASAN D.; SWEE SIEN TAN; CHENG, Aug 1999). Redes neurais do tipo Kohonen definem a demanda de energia para o próximo dia, enquanto o sistema especialista gera as regras difusas com a relação entre a temperatura e a demanda de energia descoberta pela rede neural.

Neste trabalho ficou evidente a preocupação dos autores com o fator de crescimento da economia do país e os impactos causados por esta variável na demanda por energia. Dessa forma, antes de submeter os dados históricos para o treinamento das redes neurais, a demanda por energia registrada nos dados de treinamento sofreu uma correção referente ao crescimento da economia registrado na época da coleta dos dados. No total, oito redes neurais foram geradas. Cada rede ficou responsável pelo tratamento de um dia da semana. A rede neural restante recebeu a responsabilidade de avaliar os dados referentes a feriados. Além da temperatura observada no momento de coleta dos dados, outras variáveis também foram analisadas: a característica do dia (normal, pré ou pós-feriado), tipo de chuva previsto para o dia (sem chuva, pouca ou muita chuva) entre outras. Estas variáveis também são levadas em consideração pelo sistema especialista no momento de geração das regras.

Em Kotanzad *et alli* (KHOTANZAD A.; ENWANG ZHOU; ELRAGAL, Nov 2002) os autores propõem um sistema de previsão de carga para um mercado onde o preço da energia é variável, em outras palavras, o preço da energia elétrica é diferente de um instante para o outro. Técnicas de redes neurais, lógica difusa e algoritmos genéticos são utilizados para abordar o problema.

O sistema de previsão proposto nesse trabalho pode ser decomposto em três subsistemas. Primeiramente, duas redes neurais trabalham em paralelo e sobre o mesmo conjunto de dados para gerar a previsão de carga, com as seguintes diferenças: a primeira rede possui 24 neurônios na camada de saída e apresenta a previsão de demanda para cada hora do dia seguinte. A segunda rede neural tem apenas um neurônio na camada de saída e apresenta a mudança nas características de demanda do dia atual para o dia futuro. Os resultados das duas redes neurais são integrados para formar a previsão de carga final. Essa previsão de demanda obtida pelas redes neurais foi gerada para um mercado onde o preço da energia elétrica é fixo. Para transformar essa previsão para um mercado de preço variável, é usado um sistema baseado em lógica difusa. Um algoritmo de extração de regras é usado para obter as regras das redes neurais treinadas, transformando o conhecimento armazenado nas redes em lógica difusa. As regras obtidas com o sistema de extração são submetidas para um terceiro sistema, cujo objetivo é otimizar

as regras sobre a demanda de energia. Algoritmos genéticos são utilizados para esse propósito. No final do processo, as regras estão refinadas e prontas para serem analisadas por especialistas humanos.

Carpinteiro *et alli* (CARPINTEIRO et al., 31 July-4 Aug. 2005) desenvolveram um previsor híbrido com o uso de redes neurais do tipo SOM e uma camada de *perceptrons* para previsão do pico de demanda de energia para o próximo ano.

O modelo proposto pelos autores é constituído de duas redes do tipo SOM dispostas em camadas e uma única camada de neurônios do tipo *perceptron* para a apresentação do resultado da previsão. A primeira camada de redes SOM recebe como entrada um vetor de dados históricos sobre o pico de demanda por energia dos anos passados. A primeira camada processa a entrada de dados e submete o resultado para a segunda camada de rede SOM. Os resultados da segunda camada são disponibilizados para a camada de *perceptrons*, onde é apresentado o pico de demanda de energia para a próxima semana e para o próximo ano. O modelo apresenta um bom resultado para as previsões, mesmo para um horizonte de previsão de um ano. Os autores acreditam que o uso das duas camadas de redes SOM é o principal fator para o sucesso do modelo.

### 3.3 Previsão de Carga com Outros Métodos

Gross e Galiana apresentam um estudo de diversos métodos de previsão de carga em (GROSS; GALIANA, Dec. 1987). As abordagens discutidas pelos autores são basicamente paramétricas e não-paramétricas. Além disso, os autores sugerem também um método para classificar as diversas técnicas apresentadas no trabalho.

O principal quesito de classificação analisado pelos autores é o tipo de modelo usado para a realização da previsão de carga. Outros aspectos importantes também foram observados: tamanho do conjunto de dados necessário, requisitos computacionais e resultados de experimentos científicos. Segundo os autores, os modelos de previsão de carga podem ser agrupados em duas grandes classes - modelos para previsão de pico da demanda e previsão da forma da curva da demanda. Modelos de pico da demanda visam obter a demanda máxima de energia em função de um conjunto de variáveis, geralmente climáticas. Dentre as principais vantagens inerentes ao modelo de pico de previsão está a sua simplicidade, e a necessidade de um conjunto reduzido de dados para a sua construção. Suas desvantagens residem no fato do modelo ser incapaz de informar a hora em que o pico ocorrerá, bem como o formato da curva do pico da demanda por energia. Os modelos de formato da curva de demanda a descrevem na forma

de séries temporais. Muitos desses modelos apresentam a curva de demanda juntamente com o pico da demanda por energia. Além disso, para tornar o método mais preciso e aumentar a qualidade das informações extraídas do modelo, é citada a possibilidade de combinar a técnica de formato da curva de demanda com modelos capazes de descrever o pico da demanda.

Moghram e Rahman comparam o desempenho de cinco modelos para previsão do perfil de demanda por energia (ou seja, a demanda para cada uma das vinte e quatro horas do dia), gerados a partir de métodos tradicionais nesse campo de estudo: Regressão Linear Múltipla, Séries Temporais Estocásticas, *General Exponential Smoothing*, *State-Space Method* e Sistema Especialista (MOGHRAM; RAHMAN, Nov 1989).

A metodologia de análise seguida pelos autores foi simples: cada modelo seria gerado e testado sobre dois conjuntos de dados, com dados sobre a demanda por energia nas estações de inverno e verão, respectivamente. Os conjuntos de dados foram obtidos de uma estação de energia do sudeste dos Estados Unidos, e são constituídos de leituras de diversos dados meteorológicos e a carga do sistema para cada hora, durante quatro semanas. Após a construção de todos os modelos, os autores realizaram uma análise dos resultados obtidos: para previsão da demanda por energia em períodos de verão, o modelo estocástico Função de Transferência apresentou a menor taxa de erro. Já no período de inverno, o modelo de regressão linear múltipla mostrou um melhor desempenho. Os autores constatam que o bom desempenho deste último método seja devido ao agrupamento dos dados antes dos mesmos serem avaliados pelo modelo: os dados foram agrupados em seis grupos diferentes, de acordo com o comportamento da demanda de energia da região. Esse agrupamento dos dados foi realizado segundo a familiaridade dos autores com o comportamento da demanda existente na estação usada no trabalho.

Em (ABDEL-AAL, Feb. 2004), Abdel-Aal apresenta um estudo sobre previsão de demanda de energia usando *Abductive Networks*. Neste trabalho o autor gera modelos para previsão de carga para o perfil de demanda do dia seguinte, e também para a previsão de demanda para a próxima hora. Os modelos produzidos para estas duas tarefas são comparados com modelos de redes neurais gerados para o mesmo fim, e a partir do mesmo conjunto de dados.

Os modelos de previsão foram gerados a partir de um conjunto de dados onde estão armazenadas medições horárias de temperatura e carga do sistema de uma companhia de energia elétrica dos Estados Unidos. Além disso, a base de dados também contém o tipo de dia para cada uma das medições - dia útil, sábado, domingo e feriado. Os dados correspondem a medições realizadas durante cinco anos. Todos os modelos foram avaliados com o sexto ano do conjunto de dados, sendo este não utilizado durante o treinamento dos mesmos. Para a tarefa de previsão do perfil da demanda do dia seguinte, foram gerados 24 modelos do tipo *abductive* e redes

neurais tradicionais. Para os modelos *abductive*, todos os dados do conjunto de treinamento (cinco anos) foram usados. Cada modelo ficou responsável pela previsão de demanda de energia para uma hora específica. Para os modelos de redes neurais, foi utilizado um treinamento do tipo validação cruzada, usando 20% dos dados para a validação. Para esta tarefa, a abordagem *abductive* apresentou um melhor desempenho. Para a tarefa de previsão de demanda para a próxima hora foram gerados um modelo *abductive* e um modelo neural. Ambos foram gerados com base na carga do sistema nas horas anteriores, a temperatura média do dia e o tipo do dia. Convém destacar que o número de entradas varia de acordo com a hora alvo da previsão. Diferentemente da primeira tarefa, aqui não houve uma diferença significativa entre os modelos. Para as duas tarefas, as redes neurais possuíam uma camada escondida com função de ativação hiperbólica. Ao substituir a função de ativação para sigmóide, mais comumente usada nas aplicações de previsão de carga, é registrada uma melhora no desempenho do previsor para o dia seguinte, mas o erro do previsor para a próxima hora aumenta consideravelmente.

### 3.4 Previsão de Carga em Feriados e Fins-de-Semana

Um sistema híbrido combinando técnicas de redes neurais e lógica difusa para previsão de carga com ênfase em fins-de-semana e feriados foi proposto por Srinivasan *et alli* em (SRINIVASAN; CHANG; LIEW, Nov 1995b). O sistema proposto pelos autores tem como objetivo prever a demanda por energia das 24 horas do próximo dia. O sistema foi submetido a um teste de desempenho baseado em um estudo de caso com dados reais obtidos de uma estação de fornecimento de energia de Singapura.

Segundo os autores, o principal motivo para a composição desse modelo híbrido é a baixa taxa de acerto das redes neurais ao prever a demanda por energia em fins-de-semana e feriados. Em virtude da reduzida quantidade de exemplos para treinamento, o aprendizado da rede neural sobre o comportamento da demanda por energia nesses dias especiais torna-se comprometido. Para contornar este problema, um *front-end* baseado em lógica difusa foi desenvolvido para gerar os dados de treinamento para a rede neural. Este sistema é responsável por realizar a fuzificação e defuzificação das variáveis climáticas e sazonais utilizadas para previsão da demanda de energia. Dentre as variáveis usadas no sistema encontram-se: previsão da temperatura mínima, máxima e média para o dia, índice de chuva previsto para o dia e quantidade de dias para o próximo feriado. O *front-end* também conta com uma base de conhecimento composta por regras do tipo *SE-ENTÃO* desenvolvidas por especialistas em demanda por energia. Essas regras recebem como entrada as variáveis dos conjuntos difusos e apresentam como saída um valor de mudança (para mais ou para menos) na demanda por energia atual. A composição dos

conjuntos difusos e a alteração da demanda resultante são utilizados como dados de treinamento para a rede neural. Na verdade, o módulo neural do sistema conta com três redes neurais do tipo *multilayer perceptron* com três camadas. Cada rede é responsável pela previsão de carga de um determinado período: dias úteis, sábados e domingos e feriados. Tanto os valores de entrada como os de saída da rede neural estão fuzificados. Um processo de defuzificação é aplicado ao resultado da rede para a obtenção da demanda por energia da próxima hora. A execução de todo o processo de modo iterativo resulta na obtenção da demanda por energia das 24 horas do dia seguinte.

Song *et alli* (SONG et al., Feb. 2005) exploram a utilização do método de regressão linear difuso para descobrir o perfil de demanda por energia do próximo dia. O método foi empregado com o principal objetivo de reduzir o erro na previsão de carga em feriados. O modelo de regressão difuso foi construído com base em dados fornecidos por uma companhia de energia elétrica da Coreia.

O modelo construído pelos autores visa descobrir o relacionamento entre o dia atual e o efeito causado por este na demanda por energia. Convém destacar que sábado é considerado dia útil na Coreia. Os valores de entrada (dia e carga da base de dados histórica) e saída (vetor com a demanda por energia para as 24 horas do dia seguinte) do modelo são fuzificados. Foi utilizada uma função de pertinência triangular para avaliar os conjuntos difusos. Ao longo do experimento, verificou-se um aumento no erro da previsão quando o sábado, ou a segunda-feira eram feriados. Isso acontece devido à existência de dois dias consecutivos com demanda por energia diferente do usual (dias úteis). Esse problema é contornado com a utilização de análise de coeficiente relativo. Outra questão considerada pelos autores tem relação com o conjunto de dados usado para a construção do modelo: dados de demanda muito antigos resultavam em um aumento do erro durante os testes do modelo. O comportamento da demanda por energia de tempos atrás era diferente do comportamento da demanda atual. Em vista disso, os autores passaram a usar apenas dados dos últimos 20 anos para a construção do modelo de regressão.

### **3.5 Conclusões Sobre a Revisão Bibliográfica**

Os textos analisados durante a revisão bibliográfica demonstram claramente a concentração dos cientistas nos estudos de previsão de carga de curto prazo. Nesse horizonte de previsibilidade, o uso de redes neurais apresenta ótimos resultados sem a necessidade de combinação de outras tecnologias, ou de conhecimentos avançados na área de previsão de carga e de Redes Neurais Artificiais.

No entanto, quando o horizonte de previsibilidade aumenta para meses ou anos, também aumenta a necessidade de conhecimento não apenas sobre técnicas avançadas de RNA, mas também sobre o próprio comportamento da demanda por energia elétrica da região de estudo. A metodologia de extração de regras proposta nesse trabalho pode fornecer os meios para a obtenção do conhecimento sobre o comportamento da carga. Dessa forma, as regras extraídas a partir das redes neurais contribuirão para a criação de sistemas de previsão de demanda por energia elétrica para longo prazo mais confiáveis.

## 4 METODOLOGIA

Este capítulo apresenta uma descrição da metodologia a ser seguida durante a condução dos experimentos definidos para validar as redes neurais desenvolvidas, bem como as regras extraídas a partir das mesmas. Primeiramente os dois métodos de extração selecionados na seção 2.7, FAGNIS e Benitez, serão descritos em detalhes. Após, será realizado um estudo de caso seguindo o mesmo roteiro definido para a condução dos experimentos. O estudo de caso cobre todas as atividades necessárias para a criação de uma rede neural utilizada para prever o consumo de energia de curto prazo, e também a extração das regras que descrevem o conhecimento adquirido pela rede através do uso dos dois métodos selecionados.

### 4.1 Descrição dos Métodos de Extração

#### 4.1.1 Benítez

O sistema de extração proposto por Benítez *et alli* encontra-se descrito na seção 2.6.6. O processo de aquisição de regras pode ser mapeado para o seguinte algoritmo:

1. Treinar a rede neural
2. Armazenar o melhor conjunto de pesos encontrados para a rede
3. Para cada neurônio presente na camada escondida:
  - Descobrir o valor de  $A$  para cada neurônio da camada de entrada, isto é, o valor necessário para um neurônio da camada de entrada, sem a interferência dos demais neurônios, ativar totalmente o neurônio da camada escondida corrente. O valor de  $A$  para cada variável de entrada é calculado por:

$$x_i = \frac{(2,2 - \tau)}{w_{ij}} \quad (4.1)$$

onde  $\tau$  é o *bias* e  $w_{ij}$  é o peso ligando a entrada  $x_i$  da rede ao neurônio da camada escondida corrente.

- Compor a regra derivada para o neurônio da camada escondida corrente: a premissa da regra é obtida com a junção de cada um dos valores de  $A$  encontrados pelo operador lógico *i-or*. O conseqüente da regra é o próprio valor do peso entre o neurônio da camada escondida analisado e o neurônio da camada de saída.

A próxima seção apresenta o FAGNIS, técnica de extração de regras proposta por Cechin (CECHIN, 1998).

### 4.1.2 FAGNIS

Cechin apresenta três métodos de extração para o FAGNIS. O algoritmo para extração de regras usando conjuntos difusos constantes é apresentado abaixo. Após a definição de cada conjunto difuso relacionado aos neurônios ocultos, é realizada uma varredura no conjunto de treinamento da rede, vinculando cada ponto de dado ao seu conjunto difuso correspondente. O algoritmo também deve contar com uma operação de comparação, para garantir que regras extraídas na iteração atual não tenham sido extraídas anteriormente.

#### Algoritmo para extração de regras usando conjuntos difuso constantes

1.  $R \leftarrow \emptyset$
2. Para cada neurônio oculto  $n$ :
  - Verificar se o neurônio em questão está em  $F_1$ ,  $F_2$  e  $F_3$  com base nas equações (4.2), (4.3) e (4.4)
3. Para cada dado no conjunto de treinamento  $x_i$ 
  - $regra_i \leftarrow$  **extração de regra**
  - Se  $regra_i \notin R$  então  $R \leftarrow regra_i$

$$F_1(x) = \begin{cases} -\text{sigm}(x) + \text{sigm}'(x)x, & x < 0 \\ 0, & \text{senão} \end{cases} \quad (4.2)$$

$$\text{sigm}'(x) = \frac{1}{2}F_2(x) \quad (4.3)$$

$$F_1(x) = \begin{cases} -\text{sigm}(x) + \text{sigm}'(x)x, & x > 0 \\ 0, & \text{senão} \end{cases} \quad (4.4)$$

O algoritmo extrai uma regra contendo como premissa o conjunto difuso no qual o padrão de entrada atual está contido. O algoritmo analisa a entrada  $z_j$  associando aos mesmos um dos três conjuntos difusos representados na Figura 2.17 (b). O conjunto difuso selecionado para o neurônio atual é aquele que apresentar maior valor de pertinência. Dessa forma, apenas uma regra é extraída por padrão de entrada. O algoritmo encontra-se detalhado abaixo.

#### Algoritmo Extração de Regra

1. Propagar  $x$  (dados de entrada) pela rede neural
2.  $j \leftarrow 1$
3. Para cada neurônio oculto  $n$ 
  - $z_j \leftarrow d'_u : \text{arc}(u', u)$
  - Calcular  $i' : (\forall i : F_{i'}(z_j) \geq F_i(z_j))$
  - $\text{pertinencia}_j \leftarrow F_{i'}$
  - $\text{index}_n \leftarrow i'$
  - $j \leftarrow j + 1$
4.  $\text{premissa} \leftarrow z \text{ is } \text{AND}_j \text{ pertinencia}_j$
5.  $P, q \leftarrow$  **conseqüência da regra**
6.  $\text{conseqüência } y = Px + q$
7.  $\text{regra} \leftarrow$  *SE premissa ENTÃO conseqüencia*

O algoritmo para calcular  $P$  e  $q$ , a conseqüência da regra, encontra-se descrito abaixo.

#### Conseqüência da regra

1. Para cada neurônio oculto  $n$ 
  - $i \leftarrow index_n$
  - Substituir a função de transferência do neurônio  $n$  por  $F_i^{sigm} x + q_i^{sigm}$
2.  $q \leftarrow rede([0...0])$
3.  $\forall j : [p_{1j} p_{2j} \dots p_{nj}] \leftarrow rede([0...010...0]) - q$

## 4.2 Estudo de caso

O estudo de caso aqui apresentado possui dois objetivos: (i) descrever as atividades a serem realizadas para a geração das redes neurais de previsão, bem como a extração das regras a partir das mesmas e (ii) comparar os dois métodos de extração selecionados em termos da qualidade das regras encontradas.

O problema definido para o estudo de caso consiste na previsão do consumo médio de energia para a próxima semana, com base nos dados referentes ao clima e à carga do sistema registrados para os dias anteriores ao dia da previsão. Um único modelo neural de previsão será gerado, e o mesmo será submetido aos dois métodos de extração de regras.

### 4.2.1 Descrição dos Dados Utilizados

Para a geração do predictor, foram utilizados dados de consumo de energia elétrica e climáticos referentes a todas as segundas-feiras do período compreendido entre os anos de 2003 e 2007. Após testes preliminares com a rede neural, realizados através de metodologias de seleção de variáveis (seção 2.2), verificou-se a capacidade da mesma de apresentar uma estimativa correta para a segunda-feira seguinte, com base nos dados referentes às quatro segundas-feiras anteriores. Abaixo encontram-se relacionadas as variáveis utilizadas no experimento.

- **Carga:** dados para a carga média do sistema, em GW, registrado para os quatro instantes anteriores ao dia da previsão.
- **Insolação:** dados da insolação média registrada para os quatro instantes anteriores ao dia da previsão.
- **Temperatura máxima:** valores da temperatura máxima, em C°, registrada nos quatro instantes anteriores ao dia da previsão.

- **Temperatura média:** valores da temperatura média, em C°, registrada nos quatro instantes anteriores ao dia da previsão.
- **Temperatura mínima:** valores da temperatura mínima, em C°, registrada nos quatro instantes anteriores ao dia da previsão.
- **Umidade:** o percentual médio de umidade relativa do ar registrado para os quatro instantes anteriores ao dia da previsão.
- **Velocidade do Vento:** a velocidade média do vento, em metros/s, registrada para os quatro instantes anteriores ao dia da previsão.
- **Variável objetivo:** *O consumo médio de energia estimado, em GW, para a próxima segunda-feira.*

Os valores para as variáveis relacionadas acima foram normalizados antes de serem utilizados no experimento, como mostra a equação a seguir:

$$DN_i = \frac{d_i - \mu_D}{\sigma} \quad (4.5)$$

onde  $DN_i$  corresponde ao dado normalizado,  $d_i$  é o dado atual presente no conjunto de dados,  $\mu_D$  é a média para todos os dados coletados para aquela variável e  $\sigma$  é o desvio padrão para a variável.

Os componentes principais (seção 2.3) das vinte e oito variáveis acima servirão de entrada para a rede neural. Seguindo o critério definido por Jolliffe, foram selecionadas as treze primeiras variáveis do conjunto. As demais variáveis foram descartadas, pois estas não fornecem informações relevantes para o modelo de previsão.

## 4.2.2 Elaboração do Modelo Neural

Uma vez definidos os dados de entrada para a rede neural, o foco do trabalho passou a ser a seleção de uma arquitetura para a rede. Para isso, diversas redes do tipo *feedforward* com diferentes quantidades de neurônios na camada escondida foram submetidas ao processo de validação cruzada (KOHAVI, 1995). A arquitetura de rede neural que apresentou melhor desempenho nesse teste (menor erro percentual médio) foi selecionada como modelo neural de previsão.

Dentre todas as arquiteturas avaliadas, selecionou-se a rede com treze neurônios na camada de entrada, um neurônio na camada escondida e um neurônio na camada de saída. Essa topologia apresentou um erro percentual médio de 6%. O desempenho da rede selecionada é ilustrado graficamente na Figura 4.1.

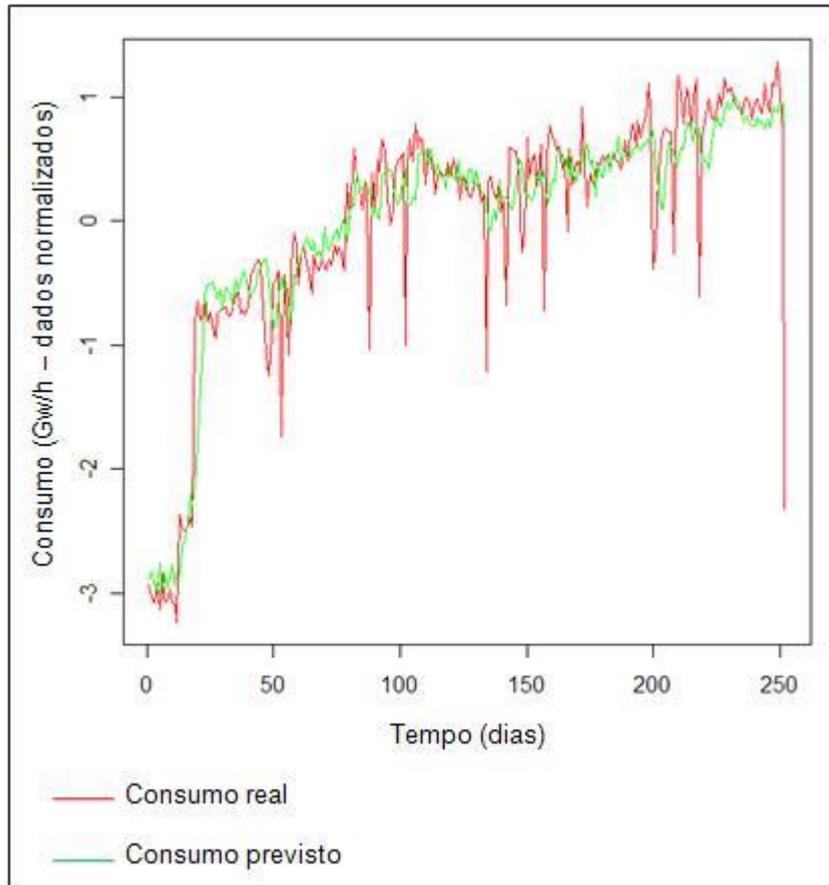


Figura 4.1: Desempenho da rede neural usada no estudo de caso

### 4.2.3 Extração das Regras

Após a conclusão das etapas de treinamento e validação da rede neural, a mesma foi submetida aos métodos de extração de regras descritos na seção anterior. Para se obter um número maior de regras durante a extração, foi adicionado mais um neurônio na camada escondida da rede neural. Essa abordagem resultou em um ligeiro aumento no erro da rede neural. Isso ocorreu devido ao reduzido número de padrões de treinamento disponíveis. Os mesmos não forneceram informações suficientes para possibilitar uma convergência dos pesos vinculados ao novo neurônio. Dessa forma, houve um compromisso entre a precisão da rede e a maior quantidade de regras a serem extraídas devido ao aumento de neurônios na camada escondida. É possível notar, através da descrição de ambos os métodos selecionados, que a quantidade

de regras extraídas depende do número de neurônios presentes na camada escondida da rede neural.

#### 4.2.3.1 Regras Extraídas Pelo Sistema FAGNIS

A Figura 4.2 mostra as regras extraídas a partir da rede neural pelo sistema FAGNIS. Por motivos de espaço e clareza, apenas as duas entradas mais importantes (os coeficientes com maior valor absoluto encontrados nas equações da parte conseqüente das regras) foram apresentadas. Na verdade, as regras fazem referência a todos os treze neurônios (os componentes principais obtidos a partir dos dados descritos na seção 4.2.1) da camada de entrada da rede neural.

<p>Regra 1 (regiões 2,1)  <math>IF (PC_2 \text{ is } -0,18) \text{ AND } (PC_8 \text{ is } -0,04)</math>  <math>THEN y = 0,07 + 0,5PC_2 + 0,09PC_8</math></p> <p>R2 (regiões 1,1)  <math>IF (PC_2 \text{ is } 1,37) \text{ AND } (PC_8 \text{ is } 0,51)</math>  <math>THEN y = 1,08 + 0,1PC_2 - 0,35PC_8</math></p> <p>R3 (regiões 3,1)  <math>IF (PC_2 \text{ is } -2,81) \text{ AND } (PC_8 \text{ is } -1,32)</math>  <math>THEN y = -2,08 + 0,1PC_2 - 0,45PC_8</math></p>
--

Figura 4.2: Regras extraídas pelo FAGNIS

As regras extraídas pelo FAGNIS informam mais que apenas o conhecimento adquirido pela rede neural durante seu treinamento: para cada uma das regras, são apresentadas as regiões da curva sigmóide ativadas em cada um dos neurônios da camada escondida da rede. Como em todas as três regras o segundo neurônio se mantém somente na primeira região da curva sigmóide, pode-se concluir que o mesmo não está sendo totalmente utilizado. Na verdade essa informação reforça as conclusões observadas durante a etapa de seleção da arquitetura da rede, cuja melhor topologia encontrada pelo processo consistia de uma rede neural com apenas um neurônio na camada oculta.

Analisando-se a parte conseqüente da regra 1, conclui-se que  $PC_2$  é a variável mais importante, pois seu coeficiente possui maior valor absoluto dentre as treze variáveis de entrada da rede neural. Isso significa que os dados agrupados na primeira regra, relacionados à variável

$PC_2$ , terão maior relevância na definição da saída apresentada pela rede neural que as demais variáveis de entrada. Assim, seria interessante descobrir quais variáveis do conjunto de dados original (seção 4.2.1) contribuem com mais informações para  $PC_2$ . Dessa forma, seria possível saber quais variáveis climáticas e de carga possuem maior importância para o previsor, para realizar a estimativa de demanda para a próxima semana. Para isso é necessário avaliar a matriz de rotação resultante da extração dos componentes principais. Deve-se verificar os valores absolutos armazenados na coluna do componente principal referenciado pela regra. Nesse caso, tratando-se da regra 1, é preciso analisar a coluna correspondente ao  $PC_2$ . A Figura 4.3 mostra a quantidade de informação de cada variável utilizada para a extração dos componentes principais para todos os dados armazenados em  $PC_2$ .

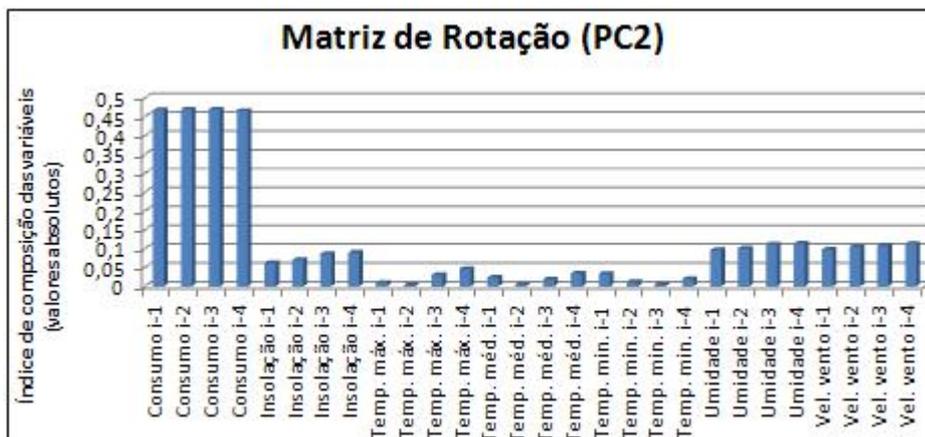


Figura 4.3: Relevância de cada variável original na composição da saída da rede

Ao analisar o gráfico acima, é possível notar a grande influência das variáveis de carga sobre  $PC_2$ . Dessa forma, pode-se concluir que o Componente Principal 2 é constituído principalmente de informações sobre a carga do sistema. As demais variáveis originais praticamente não foram utilizadas para a construção desse componente. Como  $PC_2$  é a variável mais importante para a regra 1, pode-se dizer que os dados de carga influenciam fortemente a saída da rede neural, para todos os dados associados à essa regra. Além disso, verifica-se que as variáveis climáticas não causam mudança significativa na saída da rede, para os dados agrupados nessa regra.

Naturalmente, para se obter um entendimento completo do conhecimento adquirido pela rede neural, é necessário repetir as análises aqui descritas para as demais regras encontradas.

#### 4.2.3.2 Regras Extraídas Pelo Método de Benitez

Ao executar o método de extração proposto por Benitez *et alli* sobre a mesma rede neural utilizada na seção anterior, foram encontradas as regras descritas na Figura 4.4.

<p><b>Regra 1</b>  IF <math>PC_{11} &gt; \text{approx. } 80,4056</math> <i>i-or</i>  <math>PC_{12} &lt; \text{approx. } 65,8941</math>  THEN <math>y = -68,7738</math></p> <p><b>Regra 2</b>  IF <math>PC_{11} &lt; \text{approx. } 205,4922</math> <i>i-or</i>  <math>PC_{12} &gt; \text{approx. } 155,3584</math>  THEN <math>y = -144,9685</math></p>
--

Figura 4.4: Regras extraídas pelo método proposto por Benitez

Após a análise, nota-se claramente que as regras extraídas não refletem a realidade. Os valores indicados pelas regras, para ativação total dos neurônios da camada oculta, jamais serão atingidos. Convém destacar que os dados foram normalizados logo no início do experimento. A Figura 4.5 mostra o valor de ativação do primeiro neurônio oculto da rede neural, para cada um dos dados de entrada utilizados no experimento. No gráfico, é possível observar que nenhum dos dados do conjunto é capaz de ativar completamente o neurônio. Isso mostra que, infelizmente, este método não pode ser aplicado nesse trabalho.

#### 4.2.4 Conclusões do Estudo de Caso

A principal conclusão obtida através do estudo de caso foi a não adequação do método proposto por Benitez *et alli* aos experimentos tratados nessa dissertação. Isso de forma alguma invalida o trabalho proposto pelos autores: apenas mostra que o método tem certa dificuldade em extrair regras quando a ativação dos neurônios ocultos concentra-se na região central da curva sigmóide (ver Figura 4.5). Infelizmente, esse é o comportamento geralmente observado em casos de previsão de carga. Dessa forma, optou-se por não mais utilizar o método ao longo do trabalho.

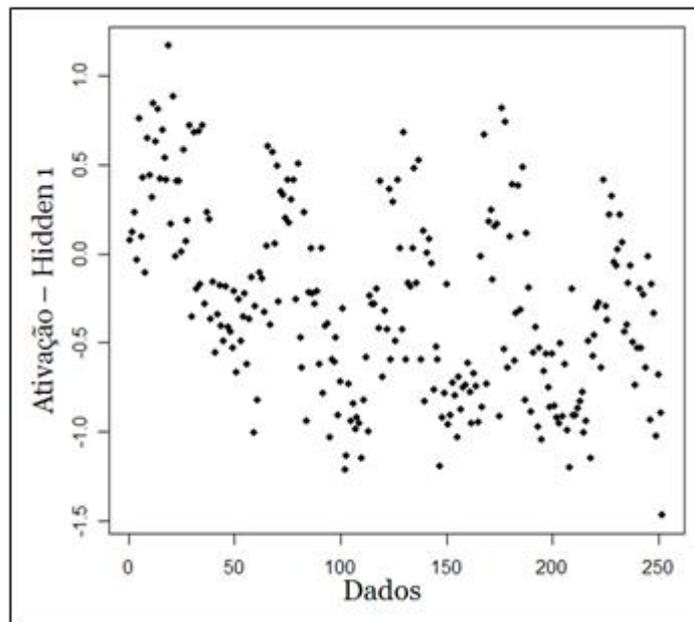


Figura 4.5: Ativação do primeiro neurônio oculto

Além disso, o estudo de caso serviu para demonstrar o desempenho da metodologia definida para extrair regras a partir de redes neurais. Destaca-se nesse sentido a utilização de métodos de seleção de variáveis para compor a camada de entrada da rede neural, bem como a avaliação da precisão da mesma com a metodologia de validação cruzada. Dificilmente essas técnicas são utilizadas em trabalhos envolvendo previsão de carga com redes neurais artificiais. Além disso, a adoção de ACP elimina a correlação entre os dados de entrada da rede neural (PEPER; NODA, Dec 1995), o que contribui para o aumento da sua precisão. A metodologia aqui demonstrada será seguida durante a condução dos experimentos elaborados para este trabalho (seção 5). Ela pode ser resumidamente organizada nos seguintes passos:

1. Selecionar os dados originais para construção do previsor neural (através de um método de seleção de variáveis).
2. Executar Análise dos Componentes Principais sobre o conjunto de dados.
3. Reter como componentes principais somente as variáveis selecionadas pelo critério de Jolliffe, sendo que essas variáveis comporão a camada de entrada da rede neural.
4. Obter, via validação cruzada, a melhor topologia de rede neural.
5. Extrair as regras a partir da rede neural obtida no passo anterior.
6. Analisar as regras obtidas.

## 5 EXPERIMENTOS E RESULTADOS

Este capítulo apresenta os experimentos executados para validar a metodologia proposta nesse trabalho. Embora diversos experimentos tenham sido conduzidos ao longo do desenvolvimento dessa dissertação, acredita-se que os relatados nesse capítulo sejam suficientes para demonstrar as capacidades da solução proposta. Serão executados dois experimentos, envolvendo a previsão de carga para horizontes distintos: (i) uma hora à frente e (ii) um mês à frente. As regras extraídas serão analisadas com o objetivo de formar conhecimento sobre o comportamento da demanda por energia elétrica para os horizontes definidos.

As redes neurais descritas nos experimentos passaram pela mesma metodologia de seleção: algoritmo de validação cruzada, onde diferentes arquiteturas de redes neurais são analisadas. As seguintes variações nas arquiteturas são testadas durante o processo de seleção: quantidade de neurônios presentes na camada escondida (de 0 a 32 neurônios), utilização ou não de atalhos (*shortcut connections*) e semente para os valores aleatórios iniciais dos pesos da rede (sementes com valores de 0 a 10). Todas as redes neurais são do tipo *feedforward*. A arquitetura de rede que apresentar o menor erro percentual médio é selecionada para o experimento. Todas as simulações foram realizadas com o pacote *nnet* (VENABLES; RIPLEY, 2002) do ambiente estatístico R (R Development Core Team, 2008).

O ambiente R foi selecionado devido a sua aceitação pela comunidade científica, por possuir grande disponibilidade de documentação e por ser um software livre distribuído de acordo com a Versão 2 da GNU *General Public License*. Além disso, o pacote *nnet* provê todas as funcionalidades importantes para a realização de estudos com Redes Neurais Artificiais. Na verdade, a única desvantagem encontrada no pacote refere-se ao algoritmo de treinamento disponibilizado: trata-se de um algoritmo desenvolvido pelos autores do pacote, e não do tradicional *backpropagation*. Esta desvantagem, no entanto, não é considerada crítica. Como mencionado anteriormente, o FAGNIS é independente do algoritmo de treinamento usado na rede neural, sendo capaz de extrair regras a partir de redes neurais treinadas por qualquer algoritmo de treinamento.

## 5.1 Experimento 1 - Extração de regras a partir de RNA aplicada para previsão de carga para a próxima hora

Este experimento lida com um problema clássico na área de previsão de carga, onde a utilização de redes neurais artificiais geralmente atinge excelentes resultados. Embora a previsão propriamente dita para um horizonte tão curto quanto o usado nesse experimento seja uma tarefa considerada trivial, seu valor não deve ser desprezado pelas companhias de fornecimento de energia elétrica. A carga para a próxima hora representa uma informação importante para traçar planos táticos da organização como a instalação de novas linhas de transmissão, o agendamento de manutenção nos equipamentos, entre outras atividades de rotina. De fato, é dessa necessidade de prever com precisão a demanda por energia elétrica para a próxima hora que surge a motivação para a realização desse experimento. Espera-se que as regras aqui obtidas possam ajudar os profissionais responsáveis pelo planejamento das operações de rotina das companhias de fornecimento de energia elétrica.

Os dados selecionados para o experimento compreendem a carga horária do período de 2003 a 2007 da cidade de Porto Alegre. Os dados foram organizados de forma a armazenar tendências diárias, semanais e mensais. A listagem abaixo representa a estrutura organizacional do arquivo de dados antes da aplicação da ACP.

- demanda de energia registrada para as últimas vinte e quatro horas (24 colunas),
- demanda para a mesma hora da previsão registrada nos últimos seis dias (6 colunas),
- demanda registrada para o mesmo dia e hora da previsão nas últimas três semanas (3 colunas),
- demanda registrada para a mesma hora e dia da previsão no mês passado (1 coluna),
- *variável objetivo: consumo médio estimado para a próxima hora.*

Após o processamento dos dados, os mesmos foram submetidos à técnica de Análise dos Componentes Principais. Verificou-se a necessidade de utilizar apenas os oito primeiros componentes, sendo os demais descartados do experimento.

A rede neural selecionada para através do processo de validação cruzada possui oito neurônios na camada de entrada, trinta e dois na camada escondida e um na camada de saída. O erro percentual médio verificado para a rede durante a validação foi de 0,022%. Embora essa rede apresente um bom desempenho, a mesma gerou mais de trezentas regras. Isso significa

que foram necessários muitos conjuntos difusos para cobrir todos os dados de treinamento, resultando em regras muito específicas que descrevem um pequeno número de dados. Por esse motivo, optou-se por reduzir a quantidade de neurônios na camada escondida da rede, de forma a se obter regras mais gerais, capazes de descrever um maior número de dados. A rede utilizada para extração de regras conta com quatro neurônios na camada escondida, apresentando um erro percentual médio de 0,027%. Seu desempenho é ilustrado graficamente pela Figura 5.1.

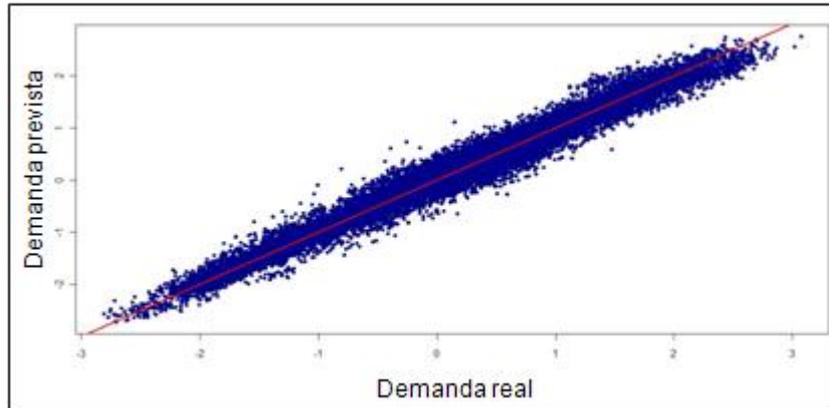


Figura 5.1: Desempenho da rede neural de extração de regras utilizada no Experimento 1

As tabelas 5.1 e 5.2 mostram as regras extraídas a partir da rede neural treinada. À primeira vista, elas informam que as três primeiras variáveis de entrada da rede são responsáveis por aumentar a demanda (seus coeficientes são positivos em todas as regras). De forma análoga, a quarta variável de entrada da rede reduz a carga do sistema (o coeficiente é negativo em todas as regras).

As regras estão dispostas no seguinte formato:

$$\begin{aligned} &\mathbf{IF} (PC_1 \text{ IS } C_1 \mathbf{AND} PC_2 \text{ IS } C_2 \dots \mathbf{AND} PC_8 \text{ IS } C_8) \\ &\mathbf{THEN} y = int + KPC_1 + KPC_2 + \dots + KPC_8 \end{aligned} \quad (5.1)$$

Onde  $C_n$  representam os conjuntos difusos da parte antecedente da regra,  $K$  refere-se aos valores dos coeficientes da equação linear da parte consequente da regra e  $int$  é o ponto onde a reta descrita pela equação linear intercepta o eixo  $y$ .

Dessa forma, a primeira regra presente na Tabela 5.1 pode ser lida como:

$$\begin{aligned} &\mathbf{IF} (PC_1 \text{ IS } 0.319 \mathbf{AND} PC_2 \text{ IS } -0.431 \dots \mathbf{AND} PC_8 \text{ IS } -0.312) \\ &\mathbf{THEN} y = 0.036 + 0.189PC_1 + 0.174PC_2 + \dots - 0.093PC_8 \end{aligned} \quad (5.2)$$

A Figura 5.2 apresenta a quantidade de informação extraída de cada variável do banco de dados original para compor o primeiro componente principal ( $PC_1$ ). Pode-se concluir que

<i>Nº</i>	<i>Descrição</i>	<i>Qtd</i>
1	IF x = (0.319 -1.272 -0.431 -0.091 0.275 -0.349 -0.159 -0.312) THEN y = (0.036 0.189 0.174 0.128 -0.175 0.159 0.047 0.038 -0.093)	8946
2	IF x = (1.913 1.350 0.327 -1.651 -0.609 0.923 0.301 0.317) THEN y = (-0.004 0.208 0.191 0.179 -0.096 0.204 -0.001 0.237 0.131)	3948
3	IF x = (1.179 -0.652 1.299 0.420 0.080 0.006 0.867 0.390) THEN y = (-0.085 0.215 0.207 0.156 -0.183 0.136 0.111 0.192 0.162)	3556
4	IF x = (-2.442 1.471 -1.012 0.537 0.206 -0.459 -0.295 -0.169) THEN y = (0.178 0.220 0.111 0.188 -0.303 0.220 0.018 0.167 -0.165)	2760
5	IF x = (0.634 -1.014 0.272 -1.031 -0.367 0.665 -0.704 -0.302) THEN y = (0.118 0.182 0.159 0.146 -0.092 0.225 -0.063 0.076 -0.125)	2130
6	IF x = (0.132 -0.742 2.721 1.271 1.224 -0.848 -0.152 -0.410) THEN y = (-0.200 0.182 0.153 0.142 -0.097 0.215 -0.062 0.070 -0.125)	805
7	IF x = (-6.143 -0.861 -1.744 0.319 -0.307 0.209 -0.089 0.078) THEN y = (0.064 0.202 0.094 0.179 -0.281 0.202 0.012 0.164 -0.163)	698
8	IF x = (-2.729 0.908 2.079 2.514 0.629 -0.839 -0.357 -0.316) THEN y = (-0.092 0.213 0.094 0.202 -0.217 0.284 -0.100 0.194 -0.209)	654
9	IF x = (2.749 -0.659 2.573 1.401 0.604 -0.916 1.203 0.212) THEN y = (-0.295 0.207 0.194 0.170 -0.112 0.190 0.021 0.225 0.135)	652
10	IF x = (-2.237 1.803 0.315 -0.185 -0.460 0.918 -0.862 -0.201) THEN y = (0.249 0.214 0.099 0.206 -0.227 0.279 -0.082 0.204 -0.198)	641
11	IF x = (-0.489 2.512 -1.068 -0.573 -0.338 1.126 -0.304 0.818) THEN y = (0.157 0.240 0.126 0.244 -0.222 0.266 -0.030 0.364 0.045)	543
12	IF x = (-5.990 -0.582 -2.232 -0.264 -0.566 0.574 -0.262 0.114) THEN y = (0.092 0.196 0.082 0.191 -0.221 0.239 -0.062 0.193 -0.185)	375
13	IF x = (-3.490 1.741 -1.235 1.420 -0.790 -1.079 -0.200 0.427) THEN y = (-0.216 0.192 0.162 0.134 -0.191 0.167 0.042 0.054 -0.095)	375
14	IF x = (-2.346 2.231 -0.744 0.719 0.668 0.086 0.403 0.761) THEN y = (0.084 0.242 0.141 0.216 -0.312 0.200 0.069 0.310 0.049)	364
15	IF x = (6.893 3.447 -0.941 -1.423 1.479 1.022 0.019 -0.118) THEN y = (0.147 0.191 0.176 0.166 -0.085 0.177 0.013 0.227 0.147)	259
16	IF x = (6.732 2.185 -0.453 -0.276 1.297 -0.683 0.186 -0.822) THEN y = (0.236 0.169 0.159 0.110 -0.162 0.124 0.064 0.044 -0.032)	253
17	IF x = (-3.341 -3.204 -1.029 0.113 -0.577 0.518 -0.211 -0.101) THEN y = (-0.022 0.179 0.161 0.128 -0.152 0.157 0.028 0.038 -0.099)	241
18	IF x = (-7.037 0.674 -1.298 1.135 -0.764 -0.314 -0.088 0.589) THEN y = (-0.382 0.173 0.149 0.116 -0.173 0.136 0.052 0.046 -0.077)	230
19	IF x = (-3.648 1.518 -0.590 2.602 -0.101 -1.518 0.068 0.233) THEN y = (-0.457 0.187 0.144 0.151 -0.111 0.229 -0.064 0.091 -0.137)	227
20	IF x = (0.332 2.722 -1.859 0.637 -2.052 0.796 -0.864 1.180) THEN y = (-0.326 0.206 0.185 0.176 -0.086 0.210 -0.005 0.230 0.120)	220

Tabela 5.1: Primeira parte das regras encontradas para o Experimento 1

<i>N°</i>	<i>Descrição</i>	<i>Qtd</i>
21	IF x = (7.414 1.328 1.016 0.005 1.005 -0.851 0.346 0.160) THEN y = (0.082 0.194 0.192 0.140 -0.169 0.109 0.120 0.190 0.183)	195
22	IF x = (-5.023 0.753 -2.007 0.092 -1.832 0.067 -0.623 0.540) THEN y = (-0.299 0.167 0.130 0.139 -0.099 0.190 -0.051 0.087 -0.118)	162
23	IF x = (-2.287 2.900 -1.116 2.558 -0.184 0.259 0.243 0.977) THEN y = (-0.416 0.213 0.206 0.154 -0.181 0.133 0.112 0.179 0.145)	139
24	IF x = (-1.261 1.569 -1.564 0.142 -2.227 -0.111 -1.235 0.460) THEN y = (-0.131 0.184 0.144 0.152 -0.104 0.235 -0.073 0.090 -0.134)	127
25	IF x = (6.567 0.195 3.182 1.280 0.643 -1.488 0.996 -0.083) THEN y = (-0.111 0.184 0.174 0.153 -0.098 0.160 0.025 0.223 0.152)	120
26	IF x = (-3.022 -3.326 -1.095 -0.183 -0.714 0.837 -0.147 0.224) THEN y = (0.021 0.174 0.149 0.141 -0.107 0.203 -0.039 0.068 -0.115)	116
27	IF x = (-4.197 1.173 -1.872 0.002 -2.325 0.280 -0.197 1.337) THEN y = (-0.435 0.189 0.168 0.162 -0.097 0.180 0.002 0.219 0.113)	116
28	IF x = (6.176 0.120 3.069 1.159 0.828 -1.446 0.464 -0.889) THEN y = (0.014 0.163 0.139 0.123 -0.080 0.181 -0.037 0.075 -0.069)	106
29	IF x = (6.340 3.300 -0.209 -1.390 0.874 0.618 -0.027 -1.023) THEN y = (0.329 0.162 0.141 0.131 -0.074 0.192 -0.040 0.078 -0.073)	85
30	IF x = (-7.189 0.576 -1.488 1.668 -0.409 -0.097 0.823 1.346) THEN y = (-0.465 0.196 0.182 0.147 -0.201 0.124 0.111 0.183 0.151)	76
31	IF x = (-1.696 0.961 -0.310 2.439 0.322 -1.334 0.634 1.111) THEN y = (-0.135 0.237 0.127 0.231 -0.234 0.241 -0.015 0.326 0.017)	64
32	IF x = (-1.946 1.715 -0.834 3.053 0.295 -1.088 0.620 1.143) THEN y = (-0.557 0.208 0.195 0.170 -0.124 0.194 0.020 0.199 0.082)	63
33	IF x = (-3.991 0.543 -1.766 -0.451 -1.541 0.926 0.175 1.225) THEN y = (0.010 0.221 0.117 0.225 -0.210 0.239 -0.021 0.355 0.073)	47
34	IF x = (-5.783 -0.886 -0.916 1.280 -0.035 0.445 0.696 1.171) THEN y = (-0.062 0.217 0.122 0.200 -0.289 0.165 0.083 0.270 0.047)	33
35	IF x = (-8.243 0.923 -0.447 2.601 -0.030 -0.786 0.463 0.734) THEN y = (-0.511 0.175 0.145 0.129 -0.113 0.191 -0.020 0.069 -0.094)	28
36	IF x = (10.618 14.251 -2.282 14.535 -4.409 4.681 0.326 -0.051) THEN y = (0.275 0.184 0.172 0.121 -0.165 0.150 0.049 0.036 -0.071)	26
37	IF x = (5.392 2.199 -0.518 -1.565 -0.556 2.259 0.661 1.038) THEN y = (0.505 0.189 0.168 0.162 -0.089 0.226 -0.063 0.136 -0.056)	20
38	IF x = (12.376 16.661 -2.798 13.870 -5.784 5.758 -1.500 0.500) THEN y = (0.508 0.169 0.156 0.128 -0.080 0.212 -0.087 0.059 -0.138)	17
39	IF x = (-3.143 -2.609 -0.615 0.964 -0.743 0.915 0.974 0.726) THEN y = (-0.084 0.208 0.157 0.170 -0.268 0.177 0.055 0.196 0.190)	10

Tabela 5.2: Segunda parte das regras encontradas para o Experimento 1

a variável mais importante para a montagem de  $PC_1$  é o valor da carga para a mesma hora da previsão registrada no dia anterior. Seguindo a mesma abordagem, foi avaliado o índice de utilização das variáveis originais para a construção dos demais componentes principais. O resultado dessa análise é apresentado na Tabela 5.3.

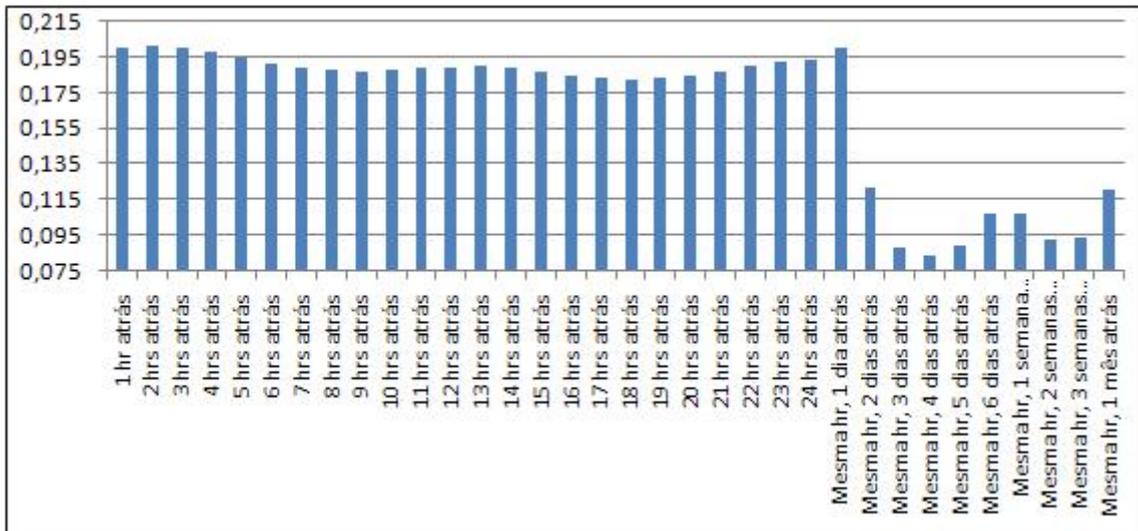


Figura 5.2: Uso das variáveis originais para compor  $PC_1$

$PC\ n^\circ$	Descrição	Val. Lingüístico
1	Consumo na mesma hora da previsão, um dia atrás	C1D
2	Consumo na mesma hora da previsão, cinco dias atrás	C5D
3	Consumo vinte horas atrás	C24H
4	Consumo na mesma hora da previsão, uma semana atrás	C1S
5	Consumo catorze horas atrás	C14H
6	Consumo onze horas atrás	C11H
7	Consumo na mesma hora da previsão, quatro dias atrás	C4D
8	Consumo no mesmo dia e hora da previsão, no mês anterior	C1M

Tabela 5.3: Variáveis mais importantes para a caracterização dos Componentes Principais

Através da análise das regras relacionadas nas tabelas 5.1 e 5.2, e juntamente com a Tabela 5.3, uma série de informações a respeito do comportamento da demanda por energia de curto prazo pôde ser obtida. Essas informações encontram-se abaixo.

Com base em seus três componentes mais importantes (coeficientes com maior valor absoluto), a Regra 1 pode ser reescrita da seguinte forma:

$$\begin{aligned} \mathbf{IF} (PC_1 \text{ IS } 0.319 \mathbf{AND} PC_4 \text{ IS } -0.091 \mathbf{AND} PC_2 \text{ IS } -1.272) \\ \mathbf{THEN} y = 0.036 + 0.189PC_1 - 0.175PC_4 + 0.173PC_2 \end{aligned} \quad (5.3)$$

para facilitar o entendimento da regra, os conjuntos difusos e variáveis lingüísticas podem ser

modificados: (ver Tabela 5.3)

$$\begin{aligned} &\mathbf{IF} (PC_1 \text{ IS positivo AND } PC_4 \text{ IS zero AND } PC_2 \text{ IS negativo}) \\ &\mathbf{THEN} y = 0.036 + 0.189C1D - 0.175C1S + 0.173C5D \end{aligned} \quad (5.4)$$

Obviamente, esse procedimento pode ser repetido para as demais regras encontradas, com o objetivo de extrair todo o conhecimento adquirido pela rede neural. Além disso, o conjunto de regras informa o seguinte:

1. o consumo registrado nas últimas vinte e quatro horas antes da previsão ( $PC_3$ ), bem como há dois dias atrás ( $PC_1$ ), e há cinco dias atrás ( $PC_2$ ) são responsáveis por aumentar a demanda por energia elétrica no dia da previsão,
2. o consumo na mesma hora da previsão, registrado uma semana atrás ( $PC_4$ ) reduz a demanda por energia elétrica.

## 5.2 Experimento 2 - Extração de regras a partir de RNA aplicada para previsão de carga para o próximo mês

A previsão de demanda por energia elétrica para o mês seguinte é uma tarefa mais complexa que a estudada na seção anterior. Como o horizonte de previsão é relativamente grande, a demanda por energia passa a sofrer o efeito das variáveis econômicas, reduzindo ainda mais a segurança na realização de uma previsão de carga confiável.

A carga prevista para o próximo mês é uma informação importante para as distribuidoras de energia elétrica no que tange as atividades de planejamento estratégico das mesmas: de posse de uma previsão correta, os gastos com compra excessiva de energia para distribuição seriam reduzidos, aumentando o lucro dessas empresas.

Para a realização desse experimento, serão usados dados de carga e climáticos referentes à cidade de Gravataí, armazenados em base diária, relativos ao período de 2005 a 2007. A listagem abaixo descreve a estrutura organizacional do arquivo de dados utilizado nessa análise:

- consumo de energia elétrica residencial registrada a 120, 90, 60 e 30 dias atrás (4 colunas),
- consumo de energia elétrica industrial registrada a 120, 90, 60 e 30 dias atrás (4 colunas),
- consumo de energia elétrica comercial registrada a 120, 90, 60 e 30 dias atrás (4 colunas),
- temperatura média registrada a 120, 90, 60 e 30 dias atrás (4 colunas),

- percentual de umidade relativa do ar registrada a 120, 90, 60 e 30 dias atrás (4 colunas),
- ***variável objetivo: consumo médio de energia elétrica para os próximos 30 dias (1 coluna).***

Após a aplicação da Análise dos Componentes Principais sobre os dados, verificou-se a necessidade de utilizar somente as sete primeiras rotações. A razão para a separação da carga nos grupos residencial, industrial e comercial reside na própria estruturação econômica da cidade de Gravataí, cuja transição de uma economia rural para industrial inciou-se na década de 60, sendo a implantação do complexo da *General Motors*, no fim dos anos 90, o ápice do crescimento industrial da cidade (IBGE, 19 de dezembro de 2007) Além disso, a cidade também apresenta um forte crescimento do setor de comércio e prestação de serviços. Dessa forma, espera-se que as regras extraídas a partir da rede neural utilizada no experimento confirmem a importância da carga industrial para determinar o comportamento da demanda por energia da cidade de Gravataí.

A rede neural selecionada pelo algoritmo de validação cruzada apresenta a seguinte estrutura: sete neurônios na camada de entrada, trinta e dois na camada escondida e um neurônio na camada de saída. Atalhos (*shortcut connections*) não foram utilizados. Essa topologia apresentou um erro percentual médio de 3,26%. Novamente, devido ao grande número de neurônios presentes na camada escondida da rede, foi obtida uma elevada quantidade de regras. Por isso foi selecionada uma topologia de rede neural diferente para a extração das regras, onde a mesma conta com oito neurônios na camada escondida, apresentando um erro percentual médio de 6,53%.

As Tabelas 5.4 e 5.5 mostram as regras mais importantes extraídas a partir da rede neural. Algumas regras foram omitidas por representarem poucos dados. As regras podem ser lidas da mesma forma que as apresentadas no experimento anterior.

Analisando-se a regra 1 juntamente com a Figura 5.3 é possível notar que a variável mais importante para a composição da carga representada por essa regra é a carga industrial de 60 dias atrás. Além disso, a carga industrial registrada 30 dias atrás também demonstra grande importância. Essa observação comprova o comportamento esperado pela demanda por energia da cidade de Gravataí, sendo a indústria a principal consumidora de energia elétrica da cidade. A Tabela 5.6 apresenta as variáveis importantes para os demais componentes principais.

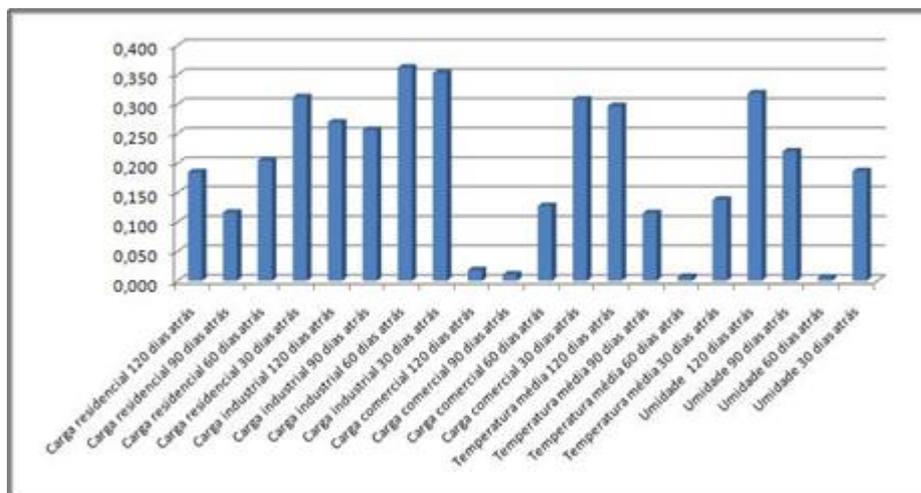
Tendo-se como base apenas os três componentes mais importantes para a regra 1 (mai-

<i>Nº</i>	<i>Descrição</i>	<i>Qtd</i>
1	IF x = (-2.863 -0.715 1.839 -0.294 0.264 -0.014 0.035) THEN y = (1.442 -0.024 0.001 0.032 -0.033 0.005 -0.004 -0.003)	110
2	IF x = (2.554 -1.616 -1.161 0.165 0.154 -0.095 -0.022) THEN y = (-0.290 -0.002 -0.378 -0.117 -0.392 0.006 -0.005 -0.235)	68
3	IF x = (4.356 0.069 2.064 -0.852 -0.152 -0.221 0.596) THEN y = (-2.804 0.874 0.339 -0.386 0.737 -0.027 0.001 0.127)	42
4	IF x = (-0.881 1.601 -1.545 0.859 -0.307 0.318 0.320) THEN y = (-0.799 0.957 0.161 -0.341 0.195 0.007 -0.032 -0.024)	41
5	IF x = (-2.148 -2.607 -0.979 -1.120 -0.930 0.005 -0.041) THEN y = (-0.927 -0.041 0.005 0.038 0.000 -0.010 -0.003 0.022)	41
6	IF x = (-0.170 1.338 0.221 2.064 -0.058 0.157 -0.011) THEN y = (-0.413 -0.228 -0.142 -0.008 -0.155 -0.024 0.017 -0.048)	40
7	IF x = (0.036 -1.674 -0.756 0.368 0.319 -0.068 0.204) THEN y = (-0.795 -0.161 0.059 -0.105 -0.182 -0.031 0.013 -0.084)	40
8	IF x = (-3.576 0.387 1.519 -0.063 -0.214 -0.364 0.211) THEN y = (1.378 -0.040 -0.045 0.030 -0.114 0.004 -0.001 -0.024)	38
9	IF x = (1.426 2.368 -0.905 -0.882 0.227 -0.130 -0.344) THEN y = (-0.803 0.278 -0.038 -0.316 -0.042 -0.039 0.046 -0.101)	37
10	IF x = (3.977 -1.090 -0.473 -1.454 0.158 0.119 -0.531) THEN y = (-0.370 0.204 -0.044 -0.157 0.335 -0.031 0.015 0.015)	37
11	IF x = (-3.735 0.368 0.761 -0.841 -0.318 0.309 -0.368) THEN y = (0.572 -0.162 -0.025 0.145 -0.018 0.001 -0.010 0.035)	35
12	IF x = (4.255 0.559 2.299 -0.503 -0.167 -0.341 0.455) THEN y = (-1.893 0.410 0.043 -0.041 -0.021 -0.009 -0.017 -0.013)	31
13	IF x = (0.134 -0.610 -1.962 0.733 0.387 -0.253 -0.412) THEN y = (-1.538 0.588 0.280 -0.404 0.254 -0.033 -0.004 -0.003)	29
14	IF x = (-0.644 1.314 -1.767 0.631 -0.005 0.000 -0.566) THEN y = (-1.330 0.948 0.222 -0.351 0.237 0.014 -0.020 -0.005)	25
15	IF x = (1.573 1.656 -1.346 -0.959 -0.346 0.375 -1.325) THEN y = (0.031 -0.193 -0.098 -0.134 0.023 -0.035 0.015 -0.055)	25
16	IF x = (0.737 -2.147 0.028 0.983 1.121 0.254 0.061) THEN y = (-1.129 0.048 0.015 -0.076 -0.013 -0.004 -0.013 -0.032)	22
17	IF x = (-0.964 1.868 -1.510 1.053 -0.152 -0.011 0.564) THEN y = (-0.525 0.750 0.015 -0.292 0.092 0.007 -0.017 -0.062)	21
18	IF x = (2.413 2.134 1.525 0.404 -0.244 -0.309 0.117) THEN y = (-0.084 0.120 -0.155 -0.033 -0.448 0.043 -0.027 -0.153)	21
19	IF x = (-0.920 1.874 -0.850 1.626 0.014 0.055 0.483) THEN y = (-0.475 0.155 -0.152 -0.127 0.006 -0.002 -0.006 -0.066)	17
20	IF x = (-3.993 -0.344 -0.774 -1.381 0.377 0.044 -0.119) THEN y = (-2.544 -1.487 0.082 1.849 0.668 -0.109 -0.100 0.801)	17

Tabela 5.4: Primeira parte das regras extraídas

N°	Descrição	Qtd
21	IF x = (0.899 -1.007 0.096 2.670 -0.395 0.107 0.559) THEN y = (-1.136 0.419 0.081 -0.058 0.020 0.003 -0.042 0.038)	16
22	IF x = (-4.254 0.709 0.035 -0.761 -0.572 0.060 -0.484) THEN y = (0.930 -0.239 -0.039 0.044 0.027 -0.014 0.042 0.063)	15
23	IF x = (1.573 3.424 0.408 -0.465 0.302 -0.189 -0.668) THEN y = (-2.758 0.966 0.375 -0.481 0.392 -0.035 0.006 0.025)	14
24	IF x = (-3.137 -1.419 -0.941 -2.014 0.451 0.158 -0.213) THEN y = (-0.667 -0.058 0.184 -0.290 0.458 -0.022 0.020 0.020)	13
25	IF x = (2.442 1.435 0.923 -0.043 -0.053 -0.079 0.335) THEN y = (-1.976 0.713 0.204 -0.322 0.321 0.011 -0.031 0.010)	13
26	IF x = (4.428 -0.484 2.181 -0.797 0.096 0.868 -0.097) THEN y = (-0.653 0.179 -0.037 -0.158 0.355 -0.027 -0.005 0.040)	13
27	IF x = (0.548 -0.207 0.432 2.539 -0.111 0.207 0.185) THEN y = (-0.260 -0.044 -0.184 0.036 -0.256 0.004 0.003 -0.089)	12
28	IF x = (-0.569 0.699 -2.151 0.048 0.221 0.542 0.100) THEN y = (-0.703 0.413 -0.045 -0.098 -0.239 0.010 -0.049 -0.094)	10
29	IF x = (-0.590 0.242 -1.076 1.776 0.257 -0.424 -0.155) THEN y = (-1.493 2.270 0.661 -0.824 0.649 -0.008 0.007 0.065)	9
30	IF x = (0.282 0.486 -0.558 1.404 -0.424 -0.014 1.013) THEN y = (0.855 -0.194 -0.338 0.045 -0.740 0.004 0.016 -0.214 )	9

Tabela 5.5: Segunda parte das regras extraídas

Figura 5.3: Quantidade de informação das variáveis originas para  $PC_4$

<i>PC n°</i>	<i>Descrição</i>	<i>Val. Lingüístico</i>
1	Carga comercial, 90 dias atrás	C90D
2	Temperatura média, 60 dias atrás	T60D
3	Temperatura média, 120 dias atrás	T120D
4	Carga industrial, 60 dias atrás	I60D
5	Umidade rel. do ar, 60 dias atrás	U60D
6	Umidade rel. do ar, 90 dias atrás	U90D
7	Umidade rel. do ar, 120 dias atrás	U120D

Tabela 5.6: Variáveis mais importantes para a caracterização dos Componentes Principais

ores coeficientes na parte conseqüente da regra), a mesma pode ser descrita como:

$$\begin{aligned} \mathbf{IF} (PC_1 \text{ IS } -2.863 \text{ AND } PC_3 \text{ IS } 1.839 \text{ AND } PC_4 \text{ IS } -0.294) \\ \mathbf{THEN} y = 1.442 - 0.024PC_1 + 0.032PC_3 - 0.033PC_4 \end{aligned} \quad (5.5)$$

atribuindo-se nomes mais significativos para os conjuntos difusos e valores lingüísticos, obtém-se a seguinte regra (ver Tabela 5.6):

$$\begin{aligned} \mathbf{IF} (PC_1 \text{ IS } \textit{negativo} \text{ AND } PC_3 \text{ IS } \textit{positivo} \text{ AND } PC_4 \text{ IS } \textit{zero}) \\ \mathbf{THEN} y = 1.442 - 0.024C90D + 0.032T120D - 0.033I60D \end{aligned} \quad (5.6)$$

A Figura 5.4 mostra o consumo de Gravataí (com valores normalizados) ao longo do tempo. Os pontos representam os dados sobre a demanda de energia elétrica explicados pela regra 1. Analisando a figura, é possível concluir que a regra 1 refere-se a conjuntos difusos contendo consumo elevado de energia elétrica. Essas mesmas análises podem ser realizadas sobre as demais regras encontradas, com o objetivo de se obter total conhecimento sobre o comportamento da demanda mensal por energia elétrica para a cidade de Gravataí.

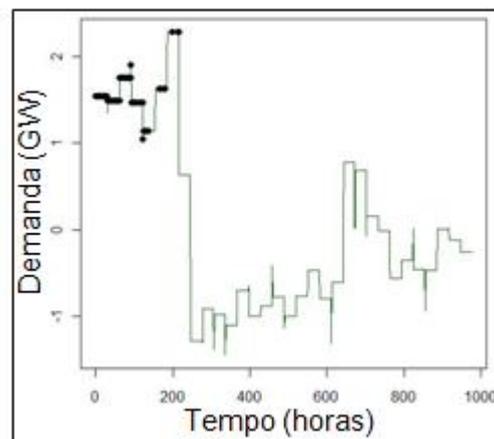


Figura 5.4: Demanda por energia elétrica e dados agrupados pela regra 1

## 6 CONCLUSÕES

Foi apresentada nesse trabalho uma metodologia de extração de regras a partir de redes neurais artificiais treinadas para previsão de demanda por energia elétrica. Os resultados encontrados nos experimentos realizados (Capítulo 5) comprovam a eficiência dessa metodologia. Além disso, é importante destacar que o método demonstrou bom desempenho em extrair regras a partir de redes neurais treinadas para diversos horizontes de previsão, sendo os dados de treinamento da rede neural o ponto mais sensível da arquitetura.

Uma característica importante sobre o método proposto é o baixo acoplamento existente entre o módulo de previsão de carga (a rede neural) e o módulo de extração de regras. Nesse sentido, a rede neural pode ser retreinada, ou ter sua arquitetura totalmente modificada sem a ocorrência de impactos no módulo de extração de regras, que executará seus algoritmos sobre a nova rede neural de forma independente.

Ao longo de diversos experimentos realizados, descobriu-se a necessidade de diferenciar as redes neurais para previsão de carga das redes neurais para extração de regras: para o primeiro caso, a rede deve ser submetida a maiores quantidades de épocas de treinamento, a fim de se obter um melhor aprendizado sobre a curva da demanda por energia elétrica. Já no segundo caso, recomenda-se o treinamento com poucas épocas, para a extração de regras mais gerais, que descrevem grandes quantidades dos dados fornecidos para a rede. Se for necessário o entendimento mais detalhado sobre o comportamento da demanda, basta aumentar o número de épocas de treinamento da rede neural de extração de regras. Dessa forma, será extraída uma quantidade maior de regras, sendo cada uma delas referente a pequenas quantidades de dados de treinamento.

Além disso, destaca-se ainda o uso de técnicas de seleção de variáveis, e de Análise de Componentes Principais. Como mostra o Capítulo 3, essas técnicas raramente são utilizadas em estudos de previsão de carga com Redes Neurais Artificiais. A utilização desses métodos contribuiu fortemente para o bom desempenho da rede neural, e também para a extração de regras bastante significativas, que certamente auxiliarão os profissionais envolvidos em tomadas de

decisões nas empresas de energia elétrica. Por exemplo, poderão ser realizadas simulações, visando o entendimento do comportamento da demanda por energia elétrica em diferentes cenários, como variações climáticas futuras. Essas simulações podem ser realizadas através da replicação dos experimentos apresentados no Capítulo 5, com dados referentes a outras regiões ou cidades atendidas pela empresa de energia elétrica.

Como trabalhos futuros, estão previstos os testes da metodologia sobre outros sistemas elétricos, bem como a utilização das regras encontradas para o desenvolvimento de Cadeias de Markov, referente à segunda parte do projeto que originou essa dissertação de mestrado.

## REFERÊNCIAS

- ABDEL-AAL, R. Short-term hourly load forecasting using abductive networks. *Power Systems, IEEE Transactions on*, v. 19, n. 1, p. 164–173, Feb. 2004. ISSN 0885-8950.
- ANDREWS, R.; DIEDERICH, J.; TICKLE, A. Survey and critique of techniques for extracting rules from trained neural networks. *Elsevier Knowledge-Based Systems*, 1995.
- BAESENS, B. et al. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science INFORMS, Vol. 49, No. 3*, 2003.
- BEAL, D. Sas code to select the best multiple linear regression model for multivariate data using information criteria. *Science Applications International Corporation*, 2005.
- BENITEZ, J.; CASTRO, J.; REQUENA, I. Are artificial neural networks black boxes? *Neural Networks, IEEE Transactions on*, v. 8, n. 5, p. 1156–1164, Sep 1997. ISSN 0000-0000.
- BERENJI, H.; KHEDKAR, P. Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Transactions on Neural Networks*, 1992.
- BOZDOGAN, H. Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, v. 52, n. 3, p. 141–149, 1987.
- CARPINTEIRO, O. et al. A hierarchical hybrid neural model with time integrators in long-term peak-load forecasting. *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, v. 5, p. 2960–2965 vol. 5, 31 July–4 Aug. 2005.
- CECHIN, A. *The Extraction of Fuzzy Rules from Neural Networks*. [S.l.]: Shaker Verlag, 1998.
- CHARYTONIUK, W.; CHEN, M.-S. Very short-term load forecasting using artificial neural networks. *Power Systems, IEEE Transactions on*, v. 15, n. 1, p. 263–268, Feb 2000. ISSN 0885-8950.
- CHURCHLAND, P.; SEJNOWSKI, T. *The Computational Brain*. [S.l.]: MIT Press, 1992.
- DOVEH, E. et al. Experience with fnn models for medium term power demand predictions. *Power Systems, IEEE Transactions on*, v. 14, n. 2, p. 538–546, May 1999. ISSN 0885-8950.
- DREZGA, I.; RAHMAN, S. Short-term load forecasting with local ann predictors. *Power Systems, IEEE Transactions on*, v. 14, n. 3, p. 844–850, Aug 1999. ISSN 0885-8950.
- DUBOIS, D.; PRADE, H. *Fuzzy Sets and Systems*. [S.l.]: Academic Press, New York, 1988.
- DUNTEMAN, G. *Principal Components Analysis: Quantitative Applications in the Social Sciences Series 69*. First. [S.l.]: SAGE Publications, 1989.
- GOTTWALD, S. *Fuzzy Sets and Fuzzy Logic*. [S.l.]: Friedrich Vieweg & Sohn Verlagsgesellschaft, 1993.

- GROSS, G.; GALIANA, F. Short-term load forecasting. *Proceedings of the IEEE*, v. 75, n. 12, p. 1558–1573, Dec. 1987. ISSN 0018-9219.
- HALGAMUGE, S.; GLESNER, M. *A Trainable Transparent Universal Aproximator for Defuzzyfication in Fuzzy Control*. [S.l.], 1994.
- HAYASHI, Y.; IMURA, A. Fuzzy neural expert system with automated extraction of fuzzy if-then rules from a trained neural network. *Uncertainty Modeling and Analysis, 1990. Proceedings., First International Symposium on*, p. 489–494, 3–5 Dec 1990.
- HAYKIN, S. *Redes Neurais: Princípios e Prática*. [S.l.]: Bookman, 2001.
- HO, K.; HSU, Y.-Y.; YANG, C.-C. Short term load forecasting using a multilayer neural network with an adaptive learning algorithm. *Power Systems, IEEE Transactions on*, v. 7, n. 1, p. 141–149, Feb 1992. ISSN 0885-8950.
- IBGE. *Produto Interno Bruto dos Municípios 2002-2005*. 19 de dezembro de 2007.
- JANG, J.; SUN, T.; MIZUTANI, E. *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence*. [S.l.]: Prentice-Hall, 1997.
- JANG, J.-S. Self-learning fuzzy controllers based on temporal backpropagation. *Neural Networks, IEEE Transactions on*, v. 3, n. 5, p. 714–723, Sep 1992. ISSN 1045-9227.
- JOLLIFFE, I. *Principal Component Analysis*. [S.l.]: Springer Series in Statistics, 2002.
- KAISER, F. The application of electronic computers to factor analysis. *Multivariate Behavioral Research*, 1960.
- KANDEL, A. *Fuzzy Mathematical Techniques with Applications*. [S.l.]: Addison-Wesley, 1986.
- KERMANSHAHI, B. Recurrent neural network for forecasting next 10 years load of nine japanese utilities. *Neurocomputing*, v. 23, 1998.
- KHOTANZAD A.; ENWANG ZHOU; ELRAGAL, H. A neuro-fuzzy approach to short-term load forecasting in a price-sensitive environment. *Power Systems, IEEE Transactions on*, v. 17, n. 4, p. 1273–1282, Nov 2002. ISSN 0885-8950.
- KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference on Artificial Intelligence*, 1995.
- LEHMANN, E.; CASELLA, G. Theory of point estimation. *Technometrics*, 1999.
- LIU, H.; HIROSHI, M. *Feature Selection for Knowledge Discovery and Data Mining*. [S.l.]: The Springer International Series in Engineering and Computer Science, 2006.
- LIU, K. et al. Comparison of very short-term load forecasting techniques. *Power Systems, IEEE Transactions on*, v. 11, n. 2, p. 877–882, May 1996. ISSN 0885-8950.
- MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 1943.

- MOGHRAM, I.; RAHMAN, S. Analysis and evaluation of five short-term load forecasting techniques. *Power Systems, IEEE Transactions on*, v. 4, n. 4, p. 1484–1491, Nov 1989. ISSN 0885-8950.
- NAUCK, D.; KLAWONN, F.; KRUSE, R. *Neuronale Netze und Fuzzy-Systeme*. [S.l.]: Vieweg and Sohn, 1994.
- PATYRA, M.; MLYNEK, D. *Fuzzy Logic Implementations and Applications*. [S.l.]: John Wiley and Sons, 1996.
- PEPER, F.; NODA, H. A class of simple nonlinear 1-unit pca neural networks. *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Dec 1995.
- R Development Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2008. ISBN 3-900051-07-0. Disponível em: <<http://www.R-project.org>>.
- RUMELHART, D.; HILTON, G.; WILLIAMS, R. Learning representations of back-propagation erros. *Nature (London)*, 1986.
- SENJYU, T. et al. One-hour-ahead load forecasting using neural network. *Power Systems, IEEE Transactions on*, v. 17, n. 1, p. 113–118, Feb 2002. ISSN 0885-8950.
- SONG, K.-B. et al. Short-term load forecasting for the holidays using fuzzy linear regression method. *Power Systems, IEEE Transactions on*, v. 20, n. 1, p. 96–101, Feb. 2005. ISSN 0885-8950.
- SRINIVASAN, D.; CHANG, C.; LIEW, A. Demand forecasting using fuzzy neural computation, with special emphasis on weekend and public holiday forecasting. *Power Systems, IEEE Transactions on*, v. 10, n. 4, p. 1897–1903, Nov 1995. ISSN 0885-8950.
- SRINIVASAN, D.; CHANG, C.; LIEW, A. Demand forecasting using fuzzy neural computation, with special emphasis on weekend and public holiday forecasting. *Power Systems, IEEE Transactions on*, v. 10, n. 4, p. 1897–1903, Nov 1995. ISSN 0885-8950.
- SRINIVASAN D.; SWEE SIEN TAN; CHENG, C. E. K. C. Parallel neural network-fuzzy expert system strategy for short-term load forecasting: system implementation and performance evaluation. *Power Systems, IEEE Transactions on*, v. 14, n. 3, p. 1100–1106, Aug 1999. ISSN 0885-8950.
- TAKAGI, T.; SUGENO, M. Fuzzy identification systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 1985.
- TSEKOURAS, G.; HATZIARGYRIOU, N.; DIALYNAS, E. An optimized adaptive neural network for annual midterm energy forecasting. *Power Systems, IEEE Transactions on*, v. 21, n. 1, p. 385–391, Feb. 2006. ISSN 0885-8950.
- VENABLES, W. N.; RIPLEY, B. D. *Modern Applied Statistics with S*. Fourth. New York: Springer, 2002. ISBN 0-387-95457-0. Disponível em: <<http://www.stats.ox.ac.uk/pub/MASS4>>.
- VERMAAK, J.; BOTHA, E. Recurrent neural networks for short-term load forecasting. *Power Systems, IEEE Transactions on*, v. 13, n. 1, p. 126–132, Feb 1998. ISSN 0885-8950.

WATANABE, D.; WAYNE, D.; YOUNT, K. A vlsi fuzzy logic controller with reconfigurable, cascable architecture. *IEEE Journal of Solid State Circuits*, 1990.

WITTEN, I.; EIBE, F. *Data Mining, Practical Machine Learning Tools and Techniques*. [S.l.]: Morgan Kaufmann, 2005. ISBN 978-0-12-088407-0.

YAGER, R. The representation of fuzzy relational production rules. *Applied Intelligence*, 1991.