

UNIVERSIDADE FEDERAL DE GOIÁS  
ESCOLA DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA  
ELÉTRICA E DE COMPUTAÇÃO

*Previsores Híbridos de Carga de  
Curto Prazo Baseados em Redes  
Neurais e Sistemas Nebulosos*

Gustavo Dias do Nascimento

Orientador: Prof. Dr. Cássio Dener Noronha Vinhal

Goiânia

2005

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

©Copyright 2005  
Gustavo Dias do Nascimento

GUSTAVO DIAS DO NASCIMENTO

*Previsores Híbridos de Carga de  
Curto Prazo Baseados em Redes  
Neurais e Sistemas Nebulosos*

Dissertação apresentado ao Programa de Pós-graduação em Engenharia Elétrica e de Computação da Universidade Federal de Goiás para obtenção do título de Mestre em Ciências em Engenharia Elétrica e de Computação.

**Área de Concentração:** Engenharia de Computação

**Orientador:** Prof. Dr. Cássio Dener Noronha Vinhal

Goiânia

2005

Universidade Federal de Goiás  
Escola de Engenharia Elétrica e de Computação  
Coordenação do Programa de Pós-Graduação em  
Engenharia Elétrica e de Computação - CPPEEC

## **FOLHA DE APROVAÇÃO**

"Previsores Híbridos de Carga de Curto  
Prazo Baseados em Redes Neurais e Sistemas  
Nebulosos"

**GUSTAVO DIAS DO NASCIMENTO**

Dissertação apresentado e analisado pela banca examinadora constituída  
pelos Senhores:

---

Prof. Dr. Cássio Dener Noronha Vinhal, Orientador - EEEEC-UFG

---

Prof. Dr. Gustavo Augusto Lima de Campos- LARCES-UECE

---

Prof. Dr. Reinaldo Gonçalves Nogueira - EEEEC-UFG

Goiânia, 31 de Maio de 2005.

# *Dedicatória*

Aos meus pais, Miguel Nascimento (in memorian) pelo exemplo de homem que foi, personalidade forte, sempre justo e de grande generosidade;

A minha mãe Maria de Lima, mulher perseverante, de grande sensibilidade e inteligência;

Ao meu irmão Rafael Nascimento, por ter se encarregado de cuidar de nossa família, na minha ausência ;

À minha namorada Márcia Patrícia, por me ajudar a concluir o trabalho e sempre estar ao meu lado em todos os momentos.

# *Agradecimentos*

Ao Prof. Cássio Vinhal, por ter acreditado no meu potencial, pela orientação, paciência nas minhas limitações e todo o apoio durante a execução deste trabalho;

Ao Prof. Gustavo Campos pelo ajuda durante o trabalho;

Aos colegas Yroá (Manga), Leonardo (Perceptron), Gustavo Rocha (Xará), Weysler (Matu) e outros, muito obrigado pela ajuda;

A CAPES por ter sido contemplado com a bolsa de estudos do mestrado;

À minha família que sempre me incentivou nessa jornada e nunca deixou faltar nada para mim nos momentos de dificuldade;

À minha namorada Márcia Patricia, que por várias vezes abriu mão da minha companhia, cuidou de mim, e teve paciência em compartilhar meus diversos problemas;

E por último, mas, não menos importante a Deus por ter iluminado meu caminho e me dado forças para superar os obstáculos.

*"Se planejarmos para um ano, devemos plantar cereais.  
Se planejarmos para uma década, devemos plantar árvores.  
Se planejarmos para toda a vida, devemos treinar e educar homens."*

***Kuentsu, Sec. III A.C.***



# *Resumo*

O presente trabalho direciona seus estudos para o processo de previsão de carga em horizonte de curto prazo no mercado de energia elétrica da Nova Inglaterra (Estados Unidos), utilizando técnicas de Sistemas Inteligentes (SI), Redes Neurais Artificiais e Sistemas Nebulosos, como tecnologias opcionais para previsão de carga em horizonte de curto prazo. As técnicas de SI, foram implementadas em *Oject Pascal* dando origem ao Previsor Inteligente TUPAN.

**Palavras-Chave:** Processos Estocásticos, Séries-Temporais, Redes Neurais Artificiais, Redes Neuro-Fuzzy, Previsão de Carga, Box & Jenkins, Lógica Nebulosa, Sistemas Inteligentes.

# *Abstract*

This work directs its research through the loading forecast process within short term electrical market of New England (United States of America), through Intelligent Systems (IS) techniques, Artificial Neural Networks, and Fuzzy Systems, as an optional mean to loading forecast within short term. The SI's techniques were built with Object Pascal developing the TUPAN Intelligent forecaster.

**Key-Word:** Random Processes, Time Series, Artificial Neural Nets, Neuro-Fuzzy Nets, Load Forecast, Box & Jenkins, Fuzzy.

# *Sumário*

**Lista de abreviaturas e siglas**

**Lista de símbolos**

**Lista de Figuras**

**Lista de Tabelas**

<b>Introdução</b>	p. 23
Motivação e Objetivos . . . . .	p. 23
Descrição do Trabalho . . . . .	p. 24
<b>1 Revisão Sobre as Técnicas de Previsão de Carga</b>	p. 26
1.1 Introdução . . . . .	p. 26
1.2 Métodos Clássicos ou Estatísticos . . . . .	p. 27
1.3 Redes Neurais Artificiais . . . . .	p. 28
1.4 Modelos Híbridos . . . . .	p. 28
1.4.1 Rede Neuro-Nebulosa . . . . .	p. 29
1.4.2 Sistema Híbrido Neuro-Nebuloso . . . . .	p. 30
<b>2 Método Clássico Box &amp; Jenkins</b>	p. 31

2.1	Box & Jenkins . . . . .	p. 31
<b>3</b>	<b>Redes Neurais Artificiais</b>	<b>p. 35</b>
3.1	O Neurônio Biológico ou Natural . . . . .	p. 35
3.2	O Neurônio Artificial . . . . .	p. 38
3.2.1	Função de Ativação . . . . .	p. 40
3.3	Topologia de uma RNA . . . . .	p. 41
3.4	Aprendizagem de uma RNA . . . . .	p. 44
3.5	Algoritmo <i>BackPropagation</i> . . . . .	p. 45
<b>4</b>	<b>Sistemas Nebulosos</b>	<b>p. 48</b>
4.1	Histórico da Lógica Nebulosa . . . . .	p. 48
4.2	Definição de Conjunto Nebuloso . . . . .	p. 49
4.3	Tipos de Função de Pertinência . . . . .	p. 50
4.4	Operações Nebulosas . . . . .	p. 52
4.5	Variáveis Lingüísticas . . . . .	p. 52
4.6	Fuzzificação, Regras Fuzzy e Defuzzificação . . . . .	p. 54
4.7	Sistema de Inferência Nebuloso . . . . .	p. 56
4.7.1	Modelos de Inferências Nebulosas . . . . .	p. 58
<b>5</b>	<b>Sistema Híbrido Neuro-Nebuloso</b>	<b>p. 60</b>
5.1	Introdução . . . . .	p. 60
5.2	Classificação dos SIH . . . . .	p. 61
5.3	O Previsor TUPAN . . . . .	p. 62

5.3.0.1	Padrão de Leitura do TUPAM . . . . .	p.64
5.3.1	O Processo de Previsão . . . . .	p.66
5.4	Os Assistentes . . . . .	p.70
5.4.1	O MARAN . . . . .	p.70
	Módulo de Construção de MLPs . . . . .	p.71
	Módulo de Treinamento . . . . .	p.74
5.4.2	O XAMAN . . . . .	p.76
	Banco de Dados do XAMAN . . . . .	p.77
	Módulo de Construção de um SHNN . . . . .	p.80
	Módulo Treinamento . . . . .	p.84
<b>6</b>	<b>Revisão sobre Previsão de Carga a Curto Prazo</b>	<b>p.87</b>
6.1	Previsão de Carga . . . . .	p.87
6.2	Parâmetros para Previsão . . . . .	p.88
6.2.1	Horizontes de Previsão . . . . .	p.88
6.2.2	Perfil de Carga . . . . .	p.89
6.2.3	Fonte de Dados . . . . .	p.89
6.2.4	Medidas de Performance . . . . .	p.90
<b>7</b>	<b>Estudos de Caso</b>	<b>p.92</b>
7.1	Introdução . . . . .	p.92
7.2	Estudos de Casos Iniciais: Mercado de Energia Regulado . . . . .	p.93
7.2.1	Estudo de Caso A . . . . .	p.93

7.2.2	Estudo de Caso B . . . . .	p. 96
7.2.3	Comentários . . . . .	p. 99
7.3	Estudos de Casos: Mercado Após a Desregulamentação . . . . .	p. 99
7.3.1	Estudo de caso C . . . . .	p. 100
7.3.2	Estudo de caso D . . . . .	p. 103
7.3.3	Comentários . . . . .	p. 106
7.3.4	Estudo de caso E . . . . .	p. 106
7.3.5	Comentário Final . . . . .	p. 111
<b>Conclusão</b>		<b>p. 113</b>
	Considerações Finais . . . . .	p. 113
	Trabalhos Futuros . . . . .	p. 114
<b>Apêndice A - Algoritmos e Métodos de Aprendizado</b>		<b>p. 115</b>
A.1	Aprendizado Supervisionado . . . . .	p. 115
A.1.1	Regra Delta Generalizada . . . . .	p. 116
A.1.2	Momentum . . . . .	p. 118
A.1.3	Weight Decay . . . . .	p. 118
A.2	Aprendizado por Reforço . . . . .	p. 118
A.2.1	Q-Learning . . . . .	p. 119
A.3	Aprendizado Não-Supervisionado . . . . .	p. 120
A.3.1	Otimização Hebbiana . . . . .	p. 121
A.3.2	Regra de Oja . . . . .	p. 121

<b>Anexo A - Simulações com o Previsor</b>	p. 122
A.1 Treino usando todas as séries . . . . .	p. 122
A.2 Treino usando carga e preço . . . . .	p. 123
A.3 Regras Nebulosas . . . . .	p. 123
<b>Referências</b>	p. 125

## *Lista de abreviaturas e siglas*

ACh	Acetil-Colina
ANFIS	Artificial Fuzzy Inference System
AR	Auto Regressive
ARIMA	Auto Regressive Integrated Move Average
ARMA	Auto Regressive Move Average
CANFIS	Coactive Artificial Fuzzy Inference System
COG	Centro da Gravidade
GABA	Gama Amino Butiric-Acid
HTML	HyperText Markup Language
IA	Inteligência Artificial
LABSIM	Laboratório de Simulações
m	Metro
MA	Move Average
MAE	Erro Absoluto Médio
MAPE	Erro Percentual Absoluto Médio
MCP	Perceptron de McCulloch Pitts
MLP	Multi Layer Perceptron
mm	Milímetro
MOM	Média Máxima
mv	milivolt
NEPE	Núcleo de Estudo e Pesquisa em Energia
NEPOOL	North England Pool
POO	Programação Orientada a Objetos
RAD	Rapid Application Development



RBF	Radial Base Function
RNA	Rede Neural Artificial
SI	Sistema Inteligente
Sigm	Função Sigmoidal
Tanh	Função Tangente Hiperbólica
UFG	Universidade Federal de Goiás

## *Lista de símbolos*

$\in$	Pertence
$\notin$	Não Pertence
$\subset$	Contido
$\Sigma$	Somatório
$f'(net)$	Derivada de primeira ordem da função
$g(net)$	Função de Ativação de net
$net$	Resultado da Soma Ponderada
$NP$	Número de Previsões
$w_{i,j}$	Peso entre os nenurônios
$X_i$	Vetor de Entradas da Rede
$y$	Saída Defuzzificada
$Y_i$	Vetor de Saída da Rede
$z_t$	Variável que descreve uma série-temporal

# *Lista de Figuras*

1	Neurônio Biológico ou Natural . . . . .	p. 36
2	Comunicação entre Neurônios . . . . .	p. 37
3	Neurônio Artificial . . . . .	p. 39
5	Topologia de uma Rede Neural Artificial . . . . .	p. 42
6	Topologia do Tipo FeedForward . . . . .	p. 42
7	Modelos de Redes Recorrentes . . . . .	p. 43
8	Modelo de Topologia Alternada . . . . .	p. 44
9	Evolução das idéias sobre Lógica . . . . .	p. 49
10	Funções de Pertinência . . . . .	p. 51
11	Operações com Conjuntos Nebulosos . . . . .	p. 53
12	Gráfico das Variáveis Nebulosas . . . . .	p. 54
13	Esquema de um Sistema de Inferência Fuzzy . . . . .	p. 56
14	Esquema de um Modelo SHNN . . . . .	p. 61
15	Tela de Início do Previsor TUPAN . . . . .	p. 63
16	Telas das Rotinas Implementadas no TUPAN . . . . .	p. 64
17	Arquivo Texto contendo Regras Nebulosas . . . . .	p. 66
18	Janela de Inicial do TUPAM . . . . .	p. 67
19	Janela de Dados para Previsão . . . . .	p. 67

20	Janela de Dados Nebulosos . . . . .	p.68
21	Janela de Escolha da Rede e Pesos . . . . .	p.69
22	Janela Mostrando as Curvas de Carga e Erro . . . . .	p.69
23	Esquema de Funcionamento do MARAN . . . . .	p.71
24	Janela de Início do MARAN . . . . .	p.72
25	Janela de Dados do MARAN . . . . .	p.72
26	Janela de Adicionar Neurônios e Camadas do MARAN . . . . .	p.73
27	Resumo da MLP Criada . . . . .	p.74
28	Janela de Treinamento do MARAN . . . . .	p.74
29	Janela de Parâmetros de Treinamento do MARAN . . . . .	p.75
30	Janela que Informa o Sucesso do Treinamento . . . . .	p.75
31	Janela para Salvar os Pesos Otimizados . . . . .	p.76
32	Esquema de Funcionamento do XAMAN . . . . .	p.77
33	Janela do Microsoft Access . . . . .	p.78
34	Relacionamento das Tabelas dos Antecedentes . . . . .	p.79
35	Relacionamento das Tabelas dos Conseqüentes . . . . .	p.79
36	Janela de Início do XAMAN . . . . .	p.81
37	Janela de Dados do XAMAN . . . . .	p.82
38	Janela de Regras e Dados das Variáveis Linguísticas . . . . .	p.82
39	Janela de Dados para Defuzzificação . . . . .	p.83
40	Janela de Adicionar Neurônios e Camadas do XAMAN . . . . .	p.84
41	Resumo do SHNN Criado . . . . .	p.85

42	Janela de Treinamento do XAMAN . . . . .	p.85
43	Janela de Parâmetros de Treinamento do XAMAN . . . . .	p.86
44	Gráfico da Rede 96x18x24 com Mercado Regulado . . . . .	p.94
45	Gráfico da Rede 96x20x24 com Mercado Regulado . . . . .	p.95
46	Gráfico da Rede 96x22x24 com Mercado Regulado . . . . .	p.95
47	Gráfico da Rede 96x40x24 com Mercado Regulado . . . . .	p.96
48	Gráfico da Rede 48x18x24 com Mercado Regulado . . . . .	p.97
49	Gráfico da Rede 48x20x24 com Mercado Regulado . . . . .	p.97
50	Gráfico da Rede 48x22x24 com Mercado Regulado . . . . .	p.98
51	Gráfico da Rede 48x40x24 com Mercado Regulado . . . . .	p.98
52	Gráfico da Rede 96x18x24 Após Mercado Desregulado . . . . .	p.101
53	Gráfico da Rede 96x20x24 Após Mercado Desregulado . . . . .	p.101
54	Gráfico da Rede 96x22x24 Após Mercado Desregulado . . . . .	p.102
55	Gráfico da Rede 96x40x24 Após Mercado Desregulado . . . . .	p.102
56	Gráfico da Rede 48x18x24 Após Mercado Desregulado . . . . .	p.104
57	Gráfico da Rede 48x20x24 Após Mercado Desregulado . . . . .	p.104
58	Gráfico da Rede 48x22x24 Após Mercado Desregulado . . . . .	p.105
59	Gráfico da Rede 48x40x24 Após Mercado Desregulado . . . . .	p.105
60	Gráfico da Rede 48x18x24 usando COG . . . . .	p.107
61	Gráfico da Rede 48x20x24 usando COG . . . . .	p.108
62	Gráfico da Rede 48x22x24 usando COG . . . . .	p.108
63	Gráfico da Rede 48x40x24 usando COG . . . . .	p.109

64	Gráfico da Rede 48x18x24 usando MOM . . . . .	p. 110
65	Gráfico da Rede 48x20x24 usando MOM . . . . .	p. 110
66	Gráfico da rede 48x22x24 usando MOM . . . . .	p. 111
67	Gráfico da Rede 48x40x24 usando MOM . . . . .	p. 111
68	Esquema da Aprendizado Supervisionado . . . . .	p. 116
69	Esquema do Aprendizado por Reforço . . . . .	p. 119
70	Esquema de Aprendizado Não-Supervisionado . . . . .	p. 120

## *Lista de Tabelas*

1	Exemplo de Entradas Padrão de Leitura do Previsor . . . . .	p. 65
2	Padrão de Entradas das Regras Nebulosas . . . . .	p. 65
3	Esquema de Montagem das Regras Nebulosas . . . . .	p. 66
4	MAPE usando todas as Séries com Preço Pouco Volátil . . . . .	p. 94
5	MAPE usando Séries de Carga e Preço Pouco Volátil . . . . .	p. 96
6	MAPE usando todas as Séries com Preço Volátil . . . . .	p. 100
7	MAPE usando Séries de Carga e Preço Volátil . . . . .	p. 103
8	MAPE do SHNN usando COG e Séries de Carga e Preço Volátil	p. 107
9	MAPE do SHNN usando MOM e Séries de Carga e Preço Volátil	p. 109
10	Treino das Redes usando todas as Séries . . . . .	p. 122
11	Treino usando Carga e Preço Volátil . . . . .	p. 123
12	Regras usadas no Processo de Previsão de Carga . . . . .	p. 124

# *Introdução*

## **Motivação e Objetivos**

Antes dos meios de comunicação divulgarem notícias sobre a possibilidade da falta de energia elétrica no ano de 2002, acontecimento esse que foi batizado de “Apagão”, muito se falava e pouco se fazia sobre: economia de energia, previsão de consumo, expansão no setor de geração de energia elétrica, etc.

Mediante os problemas apresentados, surgiu a necessidade de se conhecer com mais **exatidão** o consumo de energia elétrica, ou seja, há necessidade de se saber quanto é que vai ser gasto de energia para que a empresa distribuidora possa comprar de maneira racional. Erros por menores que sejam provocarão grandes prejuízos para a distribuidora, podendo inclusive refletir na tarifa do usuário final. Com isso, foram desenvolvidas inúmeras técnicas de previsão de carga utilizando várias áreas do conhecimento que vão desde a **Estatística** até a área de **Sistemas Inteligentes**<sup>1</sup> (SI).

A quantidade de trabalhos envolvendo o uso da Computação Flexível, em particular Redes Neurais Artificiais (RNAs), no processo de previsão de carga como mostra Hippert, Pedreira e Souza (2001), Swarup e Satish (2002), é considerável e está em constante atualização. Entretanto, trabalhos que usam uma abordagem híbrida como a Neuro-Nebulosa (NETO, 1999) são poucos os trabalhos, por isso existe a necessidade de uma investigação sobre novas técnicas a serem utilizadas, aumentando assim o conhecimento da comunidade científica

---

<sup>1</sup>São sistemas computacionais que usam as técnicas de Inteligência Artificial para solucionar problemas do mundo real.



a respeito das mesmas.

Os objetivos a serem alcançados com o desenvolvimento desse trabalho consistem basicamente na revisão do "Estado da Arte" e na avaliação da efetividade de **técnicas mais recentes** em contraste com aquelas que estão sendo propostas, e.g., as que incorporam aspectos conhecidos como Neuro-Nebulosos, de modo a contemplar os problemas que surgiram com a abertura dos mercados e a conseqüente introdução de séries discretas que apresentam certa volatilidade no que diz respeito aos valores ou preços da energia gerada, distribuída e comercializada.

## Descrição do Trabalho

Este trabalho é composto resumidamente de duas partes, uma contendo todo o embasamento teórico para o seu bom entendimento e outra contendo aspectos sobre as implementações, estudos de caso e análises destes. Com isso este trabalho se estendeu como descrito a seguir.

O Capítulo 1 procura desenvolver uma revisão das técnicas usadas no processo de previsão de carga. Já os Capítulos 3, 4 e 5 detalham melhor algumas das técnicas esboçadas brevemente no Capítulo 1. No entanto, é no Capítulo 5 que se encontra a proposta para construção de um Sistema Híbrido Neuro-Nebuloso.

No Capítulo 6, discute-se sobre os pré-requisitos necessários para o processo de previsão de carga. Finalmente, o Capítulo 7 trata da análise comparada das técnicas utilizadas, posteriormente são tecidas as conclusões gerais a respeito do que foi tratado.

Com relação à implementação, optou-se pelo Object Pascal utilizado no Borland Delphi versão 7. A motivação se baseia no fato de ser esta uma ferramenta comercialmente disponível e apoiada por um ambiente de programação visual RAD (*Rapid Application Development*) que possui inúmeros componentes prontos

e que facilitam a implementação do sistema. Implementação essa que deu origem ao Previsor TUPAN, que se vale tanto das Redes Neurais Artificiais quanto Sistemas Híbridos Neuro-Nebulosos no processo de previsão de carga.

# *1 Revisão Sobre as Técnicas de Previsão de Carga*

Neste capítulo serão abordados resumidamente as técnicas de previsão de carga de energia elétrica, expondo suas vantagens e desvantagens.

## **1.1 Introdução**

Previsões de carga de curto prazo em sistemas de potência são de grande importância por vários motivos, como por exemplo :

1. Pode-se gerar energia de maneira mais eficiente e se controlar a vazão da represa, no caso das hidrelétricas, para geração de energia satisfazendo a demanda;
2. Segurança, pois detecta-se em tempo hábil qualquer imprevisto evitando assim, que o sistema fique vulnerável;
3. Economia, pois setores industriais ou residenciais podem economizar grandes somas em dinheiro fazendo uma previsão do consumo de carga;
4. Análise de sistemas de potência. Através da análise, pode-se prever se o sistema de potência necessitará de expansão para atender a demanda dos consumidores de modo que o sistema gerador não entre em sobrecarga, originando os apagões.

Existem vários métodos para a previsão de carga de curto prazo. Estes métodos podem ser classificados, basicamente, em dois grupos: os baseados em **Estatística** e métodos baseados em **Sistemas Inteligentes** (Sistemas Especialistas, Redes Neurais Artificiais, Sistemas Nebulosos). Pensando isto, as próximas seções consistem em um pequeno resumo de cada técnica de previsão de carga que vai das técnicas mais antigas até as atuais.

## 1.2 Métodos Clássicos ou Estatísticos

Os métodos estatísticos foram os primeiros métodos usados no processo de previsão de carga, eles têm como característica forte formulação matemática, simplicidade, pois sabe-se todos os passos (formulação matemática) e variáveis do processo de previsão de carga. Dentre os vários processos, os mais famosos são os baseados em séries-temporais <sup>1</sup> e os baseados em regressão <sup>2</sup>.

Os métodos baseados em séries-temporais tratam o padrão de carga como um sinal de série-temporal com periodicidade conhecida, semanalmente, diariamente ou até mesmo em horas. Essa periodicidade dá uma predição acidentada da carga para um dado intervalo da semana e horário do dia. A diferença entre a predição e a carga atual pode ser considerada como um processo estocástico que trata de sinais aleatórios. As técnicas mais usadas para análise de tais sinais randômicos são os filtros de Kalman<sup>3</sup> e o Box & Jenkins, sendo o último, o mais popular.

Já os métodos baseados em regressão são usados para descrever o comportamento estocástico do padrão de carga horário de um sistema de energia. Este modelo assume que uma determinada carga para uma hora em particular possa ser estimada por uma combinação linear das poucas horas anteriores. Para tanto há necessidade de um grande conjunto de dados para que o resultado

---

<sup>1</sup>É uma sequência ordenada de valores de uma variável em intervalos de tempo iguais.

<sup>2</sup>Função usada para realacionar duas variáveis e prever o valor de uma a partir da outra

<sup>3</sup>É uma solução recursiva para o problema de estimação de parâmetro de um sistema.

tenha melhor exatidão. O Capítulo 3 apresenta uma discussão mais detalhada sobre o “Estado da Arte” nestas áreas.

### 1.3 Redes Neurais Artificiais

Redes Neurais Artificiais (RNAs) são sistemas de processamento inspirados no sistema nervoso natural ou biológico. Mais tarde, houve uma modelagem matemática para que o computador pudesse simular o processamento do neurônio natural (RICH; KNIGHT, 1993; BARRETO, 1999). Redes Neurais Artificiais (RNAs) também são usadas na previsão de carga, pois os métodos clássicos baseados em séries-temporais e regressão linear apresentam sérias limitações como: baixa flexibilidade, grande consumo computacional para processamento das informações e tempo de resposta alto, modelagem complexa (PAN; LIU; MEJABI, 1997).

As RNAs de modo a suprirem as eventuais deficiências (EL-SHARKAWI; NIEBUR, 1996; LIMA, 95) se valem de características tais como: capacidade de generalização (pois possuem um processo de aprendizagem que otimiza a rede), alta velocidade de processamento das informações (após a definição da topologia da rede e levantamento do parâmetros da rede), flexibilidade, tempo de resposta curto, baixo consumo computacional e a capacidade de convergir para um erro mínimo aceitável. O Capítulo 4 apresenta uma discussão mais detalhada sobre o “Estado da Arte” nestas áreas.

### 1.4 Modelos Híbridos

Embora as RNAs apresentem vantagens em relação aos métodos clássicos de previsão, elas ainda possuem limitações conforme Neto (1999), tais como: as RNAs podem apresentar *overfitting* (algo como memorização) o que faz com que a qualidade das previsões e o rendimento da rede caia, as RNAs não conseguem

processar dados qualitativos (alto, baixo, frio, quente); o conhecimento que as redes neurais adquirem não é facilmente compreendido, fazendo com que as RNAs tenham o apelido de “Caixas Pretas”; os programas que usam essa tecnologia padecem de uma falta de confiança, pois não se sabe se os conjuntos e os parâmetros de treinamento são suficientes ou adequados para o treinamento da rede até que a mesma fique otimizada.

Dadas as limitações apresentadas, os **Sistemas Híbridos** surgiram da união de dois campos da Computação Flexível, a **Lógica Nebulosa** e as **RNAs**. Esta união faz com que os Sistemas Híbridos apresentem vantagens em relação às tecnologias abordadas anteriormente.

### 1.4.1 Rede Neuro-Nebulosa

As Redes Neuro-Nebulosas (RNNs) processam dados qualitativos (alto, baixo, frio, quente), o conhecimento que a rede adquire é de fácil compreensão, pois é baseado em regras **se então e senão**, possuem uma alta velocidade de processamento (semelhante à de uma RNA); apresentam maior flexibilidade em relação às RNAs. Considerando estas vantagens as Redes Neuro-Nebulosas podem e têm sido apontadas como mais adequadas ao processo de previsão de carga no mercado de energia elétrica, que é muito dinâmico e estocástico em relação a diversas variáveis e/ou parâmetros quantitativos e qualitativos. Surge assim, a necessidade de um sistema flexível, com regras fáceis de se entender e ao mesmo tempo robusto para que acompanhe toda a dinâmica do mercado de energia elétrica deste tipo de mercado.

As Redes Neuro-Nebulosas, entretanto, podem ser reduzidas matematicamente sem prejuízo da qualidade das previsões e melhorando a velocidade de processamento. Isso ocorre especificamente quando não se possui o conhecimento necessário durante o processo de treinamento como mostra Azeen, M e Ahmad (2000).

### 1.4.2 Sistema Híbrido Neuro-Nebuloso

Conforme Thammano (2000), Rezende (2005) um Sistema Híbrido Neuro-Nebuloso (SHNN), consiste em um sistema onde há interoperabilidade de dois ou mais sistemas distintos, para resolução de um determinado problema (nesse caso previsão de carga ) com objetivo de **suprir** as limitações de cada sistema, caso fossem executados separadamente(KASABOV, 1998).

Nesse caso, o Sistema Híbrido Neuro-Nebuloso é composto por um Sistema Nebuloso e RNAs em específico MLPs. De acordo com Liew e Srinivasan (1998) unindo essas duas técnicas,uma processando dados quantitativos e a outra dados qualitativos respectivamente, tem-se uma notável vantagem, especialmente no processo de ajuste dinâmico conforme mudança do ambiente, tolerância a ruído e a descoberta de novas relações entres as variáveis envolvidas.

## 2 *Método Clássico Box & Jenkins*

Dentre os métodos clássicos serão abordados especificamente os métodos Box & Jenkins que são muito utilizados nos processos de previsão de carga que utilizam séries-temporais.

### 2.1 **Box & Jenkins**

Em meados de 1960, os professores George E. P. Box e Gwilym M. Jenkins escreveram diversos trabalhos sobre a teoria de controle e análise de séries temporais. Em 1970 publicaram o livro *Time Series Analysis, forecasting and control* apresentando uma metodologia para a análise de séries temporais e em 1976 foi lançada a versão revisada desse livro e que normalmente é a mais mencionada. O grande mérito desse trabalho foi reunir as técnicas existentes numa **metodologia** para construir modelos que descrevessem com precisão e de forma parcimoniosa o processo gerador da série temporal.

Os métodos Box & Jenkins realizam uma análise e modelagem do comportamento (GROSS; GALIANA, 1987; DINIZ et al., 1998) de uma série-temporal através de execuções sucessivas dos passos descritos abaixo:

1. **Identificação do Modelo:** a função de autocorrelação estimada e a função de autocorrelação parcial podem ser usada apropriadamente para identificar a maior parte do modelo Box & Jenkins bem como determinar sua ordem, isto é, o número de parâmetros para uma dada série-temporal;



2. **Estimação de Parâmetros:** a estimação de parâmetros abrange a maior probabilidade de um modelo, dado o primeiro conjunto de dados, ou seja, os parâmetros são estimados baseando-se na máxima verossimilhança;
3. **Validação do Modelo:** o modelo de validação verifica a satisfação através da realização de uma análise residual da estimação do erro em diferentes conjuntos de dados, isto é, métodos de diagnósticos são utilizados para testar o modelo estimado;
4. **Previsão:** o melhor modelo escolhido é usado para previsão. A qualidade da previsão é monitorada. Os parâmetros podem ser re-estimados ou todo o processo de modelagem pode ser repetido se a qualidade da previsão deteriorar.

Os métodos Box & Jenkins de acordo com E.P.Box, Gwilymand e Gregory (1994) têm em sua essência um modelo linear, o qual pode descrever uma série-temporal estacionária ou não-estacionária. Os modelos estacionários são a base dos métodos Box & Jenkins ,enquanto, os modelos não-estacionários constituem uma extensão do anterior, e são usados para tratar quaisquer comportamento não-estacionários de uma série-temporal. Os métodos Box & Jenkins utilizam um procedimento bem estruturado e sistemático para selecionar o modelo apropriado, de uma variedade de modelos da família *ARMA*, onde *AR* tem origem no termo inglês *Auto Regressive* (Auto-Regressivo) e *MA* também tem sua origem no termo inglês *Moving Averages* (Médias Móveis). Geralmente estes modelos se apresentam como  $AR(p)$  (modelo auto-regressivo de ordem  $p$ ) e  $MA(q)$  (modelo de médias móveis de ordem  $q$ ). A união dos dois modelos dá origem ao modelo  $ARMA(p, q)$ .

O modelo  $AR(p)$  basicamente descreve a observação de uma série-temporal  $z_t$  como um somatório dos  $p$  pesos de valores anteriormente conhecidos, ao qual  $z_{t-1}, z_{t-2}, \dots, z_{t-p}$ , são adicionados para um "ruído branco"<sup>1</sup>  $a_t$  e uma constante  $\mu$ .

---

<sup>1</sup>É um sinal indesejado presente na série-temporal, que pode ter várias origens como por

A equação 2.1 ilustra o modelo  $AR(p)$ .

$$z_t = \mu + a_t + \phi_1 z_{t-1} + \phi_2 z_{t-2} + \dots + \phi_p z_{t-p} \quad (2.1)$$

Onde  $\phi_1, \phi_2, \dots, \phi_p$  são parâmetros e  $p$  é a ordem do modelo. E o "ruído branco" tem normalmente sua distribuição com Energia  $(a_t) = 0$  na Variância  $(a_t) = \sigma_a^2$ .

O modelo  $MA(q)$  representa a observação de uma série-temporal  $z_t$  com uma média móvel do processo  $\mu$  e "ruído branco" atual  $a_t$  adicionado ao somatório dos  $q$  pesos anteriormente independentes com ruído  $a_{t-1}, a_{t-2}, \dots, a_{t-q}$ . A equação 2.2 ilustra o modelo  $MA(q)$ .

$$z_t = \mu + a_t + \theta_1 + a_{t-1} + \theta_2 + a_{t-2} + \dots + \theta_q + a_{t-q} \quad (2.2)$$

Onde  $\theta_1, \theta_2, \dots, \theta_q$  são parâmetros e  $q$  é a ordem do modelo.

Finalizando, o modelo  $ARMA(p, q)$  é uma combinação dos modelos anteriores, originando o modelo autoregressivo de médias móveis de ordem  $p$  e  $q$ , segue a equação 2.3:

$$z_t = \mu + \phi_1 z_{t-1} + \dots + \phi_p z_{t-p} + \dots + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (2.3)$$

Em relação ao modelo não-estacionário, os métodos Box & Jenkins tentam mudar a ordem da série-temporal para eliminar algum comportamento não-estacionário. Por exemplo, o modelo  $ARIMA(p, q, d)$  que significa modelo integrado autoregressivo de médias móveis de ordem  $p$  e  $q$  e número diferente de operações  $d$  é derivado do  $ARMA(p, q)$  (pode ser eliminada a tendência linear e mudar o nível). O modelo  $ARIMA(p, q, d)$  modifica uma série-temporal não-estacionária  $z_t$  para uma série  $w_t$  como mostra a equação 2.4, através da aplicação de operações de diferença  $d$ :

exemplo descargas elétricas e variação da temperatura.

$$w_t = \nabla^d z_t = (1 - B)^d z_t \quad (2.4)$$

onde  $\nabla = (1 - B)$  é o operador diferença, e  $B$  é o operador *back-shift* definido por  $B_k z_t = z_{t-k}$ . Como um resultado, a série-temporal transformada  $w_t$  pode ser representada por algum modelo estacionário previamente descrito.

## 3 *Redes Neurais Artificiais*

Neste capítulo serão mostrados em particular, os métodos baseados em Redes Neurais Artificiais (RNAs) que são atualmente muito utilizados nos processos de previsão de carga.

### 3.1 O Neurônio Biológico ou Natural

O sistema nervoso biológico ou natural é formado por um complexo conjunto de células, os neurônios. Os neurônios possuem um **corpo** ou **soma**, o qual mede cerca de 5 a 10 $mm$  de diâmetro, contém o **núcleo da célula**, e onde ocorre a maioria das atividades bioquímicas necessárias para manter a vida do neurônio; os **dendritos**, que são um conjunto de terminais de entrada que podem atingir cerca de 400 $mm$  de raio, formando assim o que se chama de **árvore dendrítica** a qual se espalha por todo corpo do neurônio e finalizando pelos **axônios** como mostra a figura 1. Estes últimos são fibras nervosas que são longos terminais de saída, que se estendem do corpo do neurônio sendo que ele pode ser curto, medindo em media 10 $mm$ , ou tão longos quanto 1 $m$ . O axônio também se espalha formando uma árvore no final do neurônio, ao contrário dos dendritos que se espalham próximo ao corpo do neurônio. A figura 1 ilustra os principais componentes do neurônio biológico.

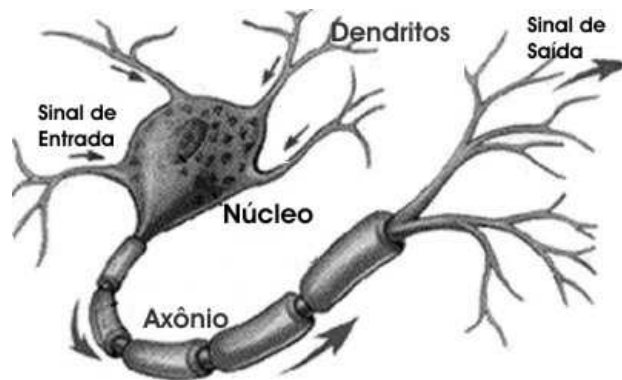


Figura 1: Neurônio Biológico ou Natural

Os neurônios se comunicam através de sinapses. Sinapse é a região onde dois neurônios entram em contato **eletroquimicamente** como mostra a figura 2 e através da qual os impulsos nervosos (que variam de 60 a 90 mv) são transmitidos entre eles. Para melhor entendimento considera-se dois neurônios A e B conforme a figura 2.

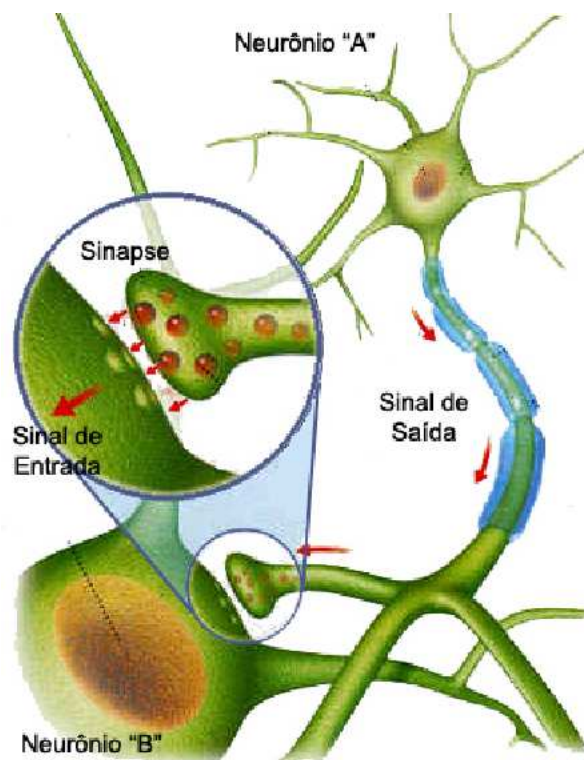


Figura 2: Comunicação entre Neurônios

Os impulsos recebidos por um neurônio A, em um determinado momento, são processados e, atingindo um dado **limiar de ação** ou **potencial de ação**; o neurônio A dispara, fazendo com que o impulso elétrico se propague do corpo da célula, passando pelas fibras axônicas, até chegar ao final do axônio que pode estar conectado a um dendrito de um outro neurônio B como ilustra a figura 2. Esse processo de transmissão se dá através da alteração da concentração dos íons  $K^-$  (Potássio) e  $Na^+$  (Sódio) presente na membrana celular de cada neurônio. Por meio de uma molécula neurotransmissora presente no cérebro como a Acetil-Colina (ACh) ou Gama-Amino-Butíric-Acid (GABA), o neurotransmissor pode diminuir ou aumentar a polaridade da membrana pós-sináptica (membrana do dendrito), inibindo (no caso do GABA) ou excitando (no caso do ACh) gerando assim pulsos elétricos no neurônio B. Este processo depende de vários fatores, como a geometria dos neurônios, a quantidade de neurotransmissores e o estado

do indivíduo.

## 3.2 O Neurônio Artificial

Os elementos básicos que formam uma rede neural artificial são os neurônios artificiais conhecidos também por **nó** ou **elemento processador** (BRAGA; LUDERMIR, 2000). Ele foi inspirado baseado no funcionamento de um neurônio natural. A figura 3 ilustra um modelo do neurônio artificial de McCulloch e Pitts.

O modelo do neurônio artificial proposto por Warren McCulloch<sup>1</sup> e Walter Pitts<sup>2</sup> (MCP) em 1943 é uma simplificação do que se sabia até então do neurônio biológico, só que estes neurônios, posteriormente, foram submetidos a uma modelagem matemática (BARRETO, 1999). Sua modelagem consiste no seguinte: o neurônio artificial possui várias entradas ( $X_i$ ) com um peso associado a cada uma ( $W_{ij}$ ), alguns pesos com sinais excitatórios ( $W_{ij} > 0$ ) e outros com sinais inibitórios ( $W_{ij} < 0$ ). Os valores de entrada e ativação das células podem ser discretos, nos conjuntos  $\{0, 1\}$  ou  $\{-1, 0, 1\}$  ou contínuos nos intervalos  $[0, 1]$  ou  $[-1, 1]$ . Cada célula ou nodo é capaz de processar vários sinais de entradas e transformá-los em um sinal de saída.

---

<sup>1</sup>Psiquiatra e neuroanatomista por treinamento, que buscava uma representação para o sistema nervoso.

<sup>2</sup>Excelente matemático que se uniu a Warren McCulloch em 1942.

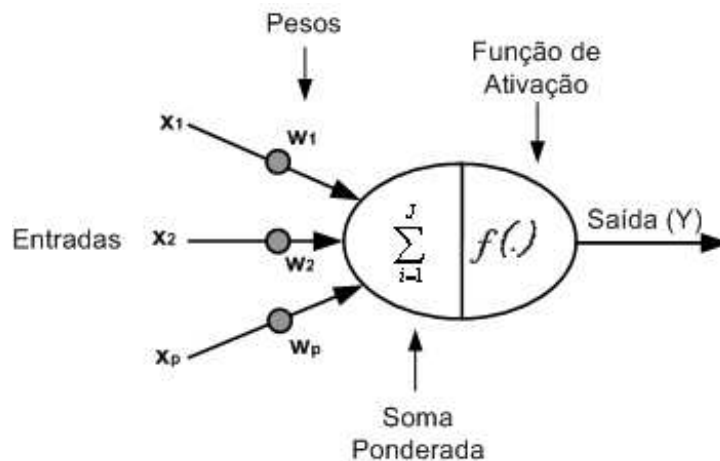


Figura 3: Neurônio Artificial

Supondo que se tenha um número de  $p$  sinais de entrada  $x_1, x_2, \dots, x_p$  do neurônio artificial que ao comparar com o neurônio biológico seria os dendritos e os pesos  $w_1, w_2, \dots, w_p$  que podem ser entendidos como “força” da conexão sináptica quando comparados com o neurônio biológico; com sinais assumindo valores booleanos ou reais e pesos assumem valores reais entre 0 e 1. Sendo assim, tem-se que a soma ponderada dada por:

$$net_j = \sum_{i=1}^J W_{ij} X_i \quad (3.1)$$

onde  $net_j$  é a variável que irá armazenar a soma ponderada; o peso  $W_{ij}$  representa o peso da sinapse conectando o neurônio  $j$  ao neurônio  $i$ ;  $J$  é o número de neurônios que estão conectados ao neurônio  $i$ ;  $X_i$  é o valor de entrada do neurônio  $i$ , conectado ao neurônio  $j$ . A partir desse valor  $net_j$  é submetido por uma função de ativação ( $Y_j = g(net_j)$ ) o que resulta na saída do respectivo neurônio, propagando para os neurônios posteriores.



### 3.2.1 Função de Ativação

A função de ativação é utilizada para o cálculo de ativação do neurônio, e que pode ser uma função não-linear, linear ou limiar (função degrau) como mostra a figura ???. Geralmente emprega-se algum tipo de função não-linear, que garante a plena funcionalidade das redes neurais e simula mais precisamente as características do neurônio biológico. As principais funções de ativação estão esboçadas na figura ???:

1. **Função Linear:**

$$g(net_j) = \alpha net_j \quad (3.2)$$

2. **Função Degrau:**

$$g(net_j) = \begin{cases} 1, & \text{se } net_j \geq 0 \\ 0, & \text{se } x < 0 \end{cases} \quad (3.3)$$

3. **Função Sigmóide:**

$$g(net_j) = \frac{1}{1 + e^{-\alpha net_j}} \quad (3.4)$$

4. **Função Tangente Hiperbólica:**

$$g(net_j) = \frac{e^{net_j} - e^{-net_j}}{e^{net_j} + e^{-net_j}} \quad (3.5)$$

Encontrado o valor de  $net_j$ , esse valor será submetido a qualquer uma das funções de ativação descritas anteriormente ou uma função específica que resultará na saída do neurônio  $Y_j$  essa saída nos dirá se o neurônio está excitado (caso seja maior que zero) ou inibido (seja menor que zero), posteriormente propaga esse sinal  $Y_j$  de saída para a entrada  $X_i$  de um outro neurônio da camada conseqüente de acordo com a topologia ou estrutura em que RNA se encontra.

No início das pesquisas envolvendo RNAs, estas eram emuladas através de circuitos analógicos. Atualmente o mais comum é a utilização de software (linguagem de programação) para a geração de redes neurais, ou seja, o funcionamento dos neurônios é simulado por programas de computador.

### 3.3 Topologia de uma RNA

As topologias das RNAs são normalmente organizadas em camadas, com uma quantidade de neurônios que podem estar conectados a neurônios de: camadas anteriores, de mesma camada e/ou de camadas posteriores. RNAs podem ser vistas como grafos direcionados ponderados nos quais os neurônios são os nós e os arcos direcionados (com pesos) são as conexões entre as saídas dos neurônios e as entradas dos mesmos. Usualmente, as camadas são classificadas em três tipos: camada de entrada, camada oculta ou intermediária e camada de saída. A figura 5 ilustra um esquema particular de organização em camadas e abaixo encontra-se a descrição de cada camada.

1. **Camada de Entrada:** é a camada onde os padrões ou exemplos são apresentados a RNA, ou seja, é a camada que “captura” os dados fornecidos pelo ambiente;
2. **Camada Intermediária ou Oculta:** é a camada onde é feita a maior parte do processamento da RNA, através de somas ponderadas entre as conexões dos neurônios e o resultado submetido a uma função de ativação, essa camada ainda pode ter várias camadas sendo que cada camada pode ter uma quantidade igual ou diferente de neurônios entre si. Pode ser considerada com a extratora de **características** ou **padrões** (BARRETO, 1999) (RICH; KNIGHT, 1993) e é onde se concentra a “Caixa Preta” do processamento das RNAs;
3. **Camada de Saída:** é a última camada onde o resultado final é concluído e apresentado, com isso o ambiente pode sofrer alguma alteração ou não.

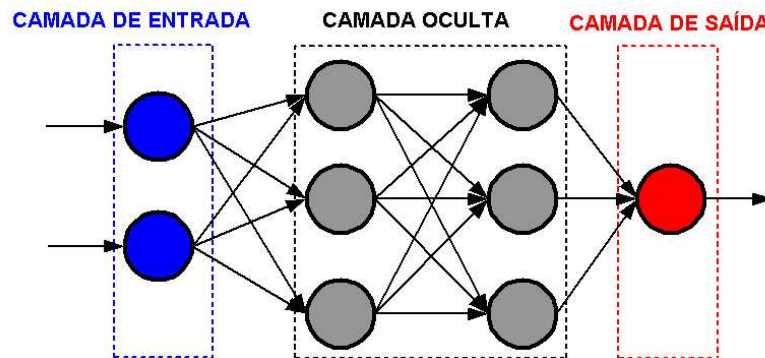


Figura 5: Topologia de uma Rede Neural Artificial

As conexões entre as camadas de neurônios, podem gerar inúmeras topologias de rede diferente. Abaixo seguem as topologias mais comuns:

1. **Avanço ou FeedForward:** a saída do neurônio na  $i$ -ésima camada da rede não pode ser usada como entrada dos neurônios de mesma camadas ou de camada anteriores, ou seja, são redes diretamente alimentadas como mostra a figura 6;

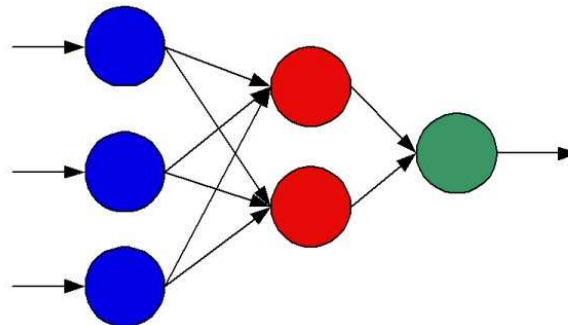


Figura 6: Topologia do Tipo FeedForward

2. **Recorrente ou FeedBack:** a saída de algum neurônio na  $i$ -ésima camada da rede é usada como entrada de neurônios de mesma camadas ou de camada anteriores, como mostra a figura 7. Esse tipo de topologia ainda possui duas características que são:

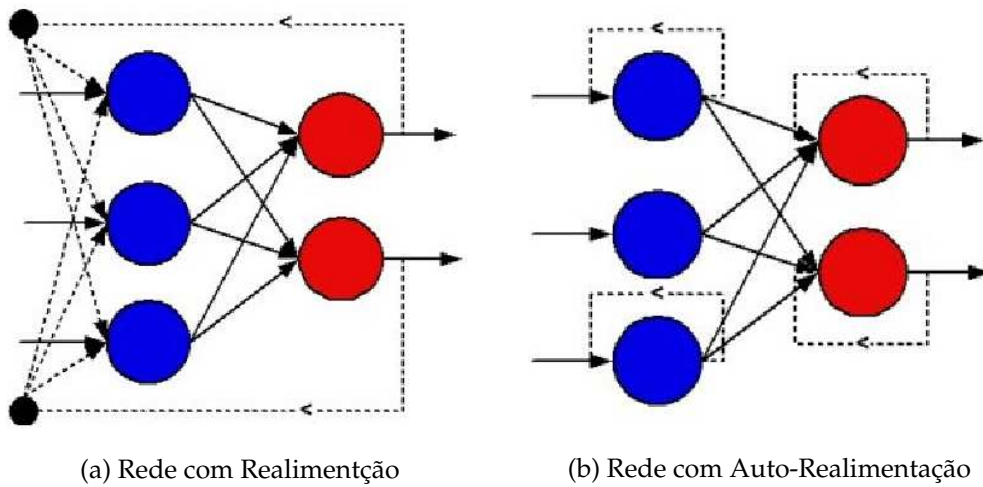


Figura 7: Modelos de Redes Recorrentes

- (a) Redes cuja saída final é ligada às entradas de um ou mais neurônios de mesma camada ou de camadas anteriores, ou seja, são redes onde há presença de realimentação, mas nunca de auto-realimentação, como mostra a figura 7(a).
- (b) Redes cujas ligações são cíclicas, a saída final é ligada na própria entrada e nas entradas dos demais neurônios independente de camada, isto é, esse tipo de rede possui realimentação e auto-realimentação, como mostra a figura 7(b).
3. **Alternadas:** nesse tipo de topologia, as conexões entre os neurônios podem ser realizadas tanto para frente quanto para trás, de uma camada qualquer de neurônios para outra camada qualquer de neurônios como mostra figura 8.

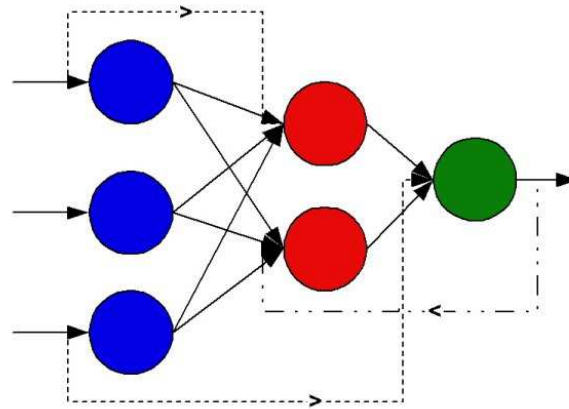


Figura 8: Modelo de Topologia Alternada

### 3.4 Aprendizagem de uma RNA

A aprendizagem para uma rede neural artificial é muito importante, pois a RNA dotada dessa capacidade pode, com o decorrer das iterações, aperfeiçoar e melhorar sua performance através dos exemplos submetidos à RNA fazendo com que a mesma fique mais “inteligente”. Isso ocorre toda vez que a saída da RNA (no caso a saída real), não satisfizer a uma determinada expectativa, ou seja, a RNA não produz a saída que se deseja. Sendo assim, implementa-se algoritmos de aprendizagem para minimizar ou até mesmo excluir a distância entre a saída da rede neural e a saída que se deseja. Pode-se dizer então que o processo de aprendizagem de uma RNA consiste basicamente de um algoritmo de aprendizagem ou algoritmo de otimização e de um método ou paradigma; ambos estão relacionados e são inseparáveis.

Os algoritmos de aprendizagem consistem em procedimentos e rotinas bem estruturados, usados para adaptar ou atualizar os pesos existentes nas RNAs. O método de aprendizagem de uma RNA, consiste em que maneira a RNA mais o ambiente irão interagir para que o algoritmo de aprendizagem corrija os eventuais erros da rede.

Como os algoritmos de aprendizagem, existem também muitos métodos

de aprendizagem, porém, são considerados como principais os métodos de **aprendizado supervisionado**, onde existe a necessidade de um “professor” no processo de treinamento, e **aprendizado não-supervisionado**, onde não há necessidade de um “professor”. Mais informações no Apêndice A.

### 3.5 Algoritmo *BackPropagation*

O *BackPropagation* é um algoritmo de aprendizagem, geralmente usado em redes multicamadas diretamente conectadas (MLPs). Esse algoritmo tem uma grande popularidade, isso se deve a sua **simplicidade de implementação** e pelo **poder de generalização**.

Durante o treinamento com o algoritmo *BackPropagation*, a rede opera em uma **sequência de dois passos**. O Primeiro passo, um padrão ou exemplo é apresentado à camada de entrada da rede. A atividade resultante flui através da rede, camada por camada, até que a resposta seja produzida pela camada de saída. No segundo passo, a saída obtida é comparada à saída desejada essa comparação leva em consideração o cálculo do erro para esse padrão particular. Se este padrão não estiver correto, o erro é propagado a partir da camada de saída até a camada de entrada. Nesse processo, os pesos das conexões dos neurônios vão sendo modificados conforme o erro é retropropagado.

Na medida que um determinado conjunto de exemplos, sejam grandes o suficiente para ser representativo no ambiente em que a rede está inserida, a MLP treinada com o algoritmo *BackPropagation* desenvolverá a capacidade de generalizar. Esta capacidade de generalizar faz com que a rede MLP apresente um resultado satisfatório quando novos dados retirados no mesmo conjunto de exemplos são apresentados à mesma.

O algoritmo de *BackPropagation* consiste de dois passos bem distintos, que são: *ForWard*, que significa propagação direta, e *BackWard*, que significa propagação regressa. Abaixo segue os passos do algoritmo *BackPropagation* que

são:

1. Normalizar as entradas e saídas dos exemplos a serem apresentados à rede no mesmo intervalo de variação da função de ativação;
2. Definir uma topologia da rede, isto é, definir o número de camadas e de neurônios por camada;
3. Inicializar os pesos com valores pequenos e randômicos, geralmente no intervalo de zero a um;
4. Escolha a(s) funções(ão) de ativação que a rede irá ter;
5. Escolher o algoritmo de otimização da rede, ou seja, algoritmo de aprendizagem que pode ser Regra Delta, *Weight Decay* entre outros;
6. Definir o critério de parada da rede;
7. Escolher o vetor de entrada (padrão) e a saída desejada correspondente aleatoriamente;
8. Calcule a soma ponderada (variável **net**);
9. Submeta a o valor da soma ponderada a uma função de ativação para originar a saída da rede, no caso,  $Y$ ;
10. Calcule o **erro** da rede primeiramente da Camada de Saída;
11. Calcule o valor da função de custo;
12. Calcule o erro da(s) Camada(s) Oculta(s);
13. Atualize os pesos (usando o algoritmo de aprendizagem escolhido no passo 5);
14. Verificar se o critério de parada foi satisfeito;

15. Caso o critério de parada não tenha sido satisfeito, ir para passo 7;

onde os itens 8,9 e 10 correspondem ao passo *ForWard* e os itens 11,12 e 13 correspondem ao passo *BackWard*. Os itens de 1 a 7 correspondem ao tratamento dos dados antes de passo *ForWard*, isso é feito para que a rede possa ter plena funcionalidade, e os passos 14 e 15 são executados depois do passo de *BackWard*, para que o processo de treinamento da rede possa ter seqüência.



## 4 *Sistemas Nebulosos*

### 4.1 Histórico da Lógica Nebulosa

A lógica matemática tradicional, proposta por Parmênides 400 anos antes de Cristo, é baseada na idéia de que as proposições possuem apenas dois valores: verdadeiro ( $V$ ) ou falso ( $F$ ). E a mesma não tolera outro valor, assemelhando-se à lógica de Boole onde somente se permite 0 ou 1.

No entanto, essa idéia já vinha tendo algumas objeções desde o tempo de Platão, que por sinal foi o primeiro a contestar a idéia original. No século XIX, o matemático Lukasiewicz descreveu um terceiro termo, como sendo uma proposta para ser um terceiro valor chamado de **possível** e este ficaria entre verdadeiro e falso. Na década de 60, o pesquisador Lotfi Zadeh (1965) da Universidade de Berkeley, introduziu os conceitos e princípios da Lógica Nebulosa (*Fuzzy Logic*) e dos conjuntos Nebulosos (*Fuzzy Sets*), que permitiram expressar conceitos vagos e tomar decisões sobre conhecimento inexato se tornando uma extensão da lógica booleana. A figura 9 mostra a evolução da lógica.

A Lógica Nebulosa é baseada no fato de que os seres humanos são capazes de manusear problemas complexos todos os dias sem ter um exato entendimento matemático do meio onde vive com informações incertas e imprecisas (NAFIS, 2001). Os seres humanos possuem a capacidade de resolver problemas baseados em um modelo heurístico, a experiência de vida acumulada também é usada como meio para promover soluções aproximadas. A Lógica Nebulosa pode ser

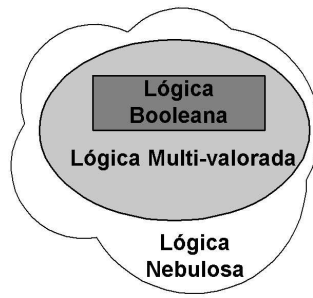


Figura 9: Evolução das idéias sobre Lógica

vista como um sistema baseado na heurística do conhecimento.

## 4.2 Definição de Conjunto Nebuloso

Considera-se que  $X$  é o espaço de objetos e  $x$  é um elemento genérico de  $X$ . Um conjunto  $A$ , onde  $A \subset X$ , é definido como uma coleção de elementos  $x \in X$ . Pode-se representar um conjunto crips ou clássico  $A$  por um conjunto de pares ordenados  $(x, 0)$  ou  $(x, 1)$ , se  $x \in A$  ou  $x \notin A$  respectivamente. Na Lógica Nebulosa, se  $X$  é uma coleção de objetos ou elementos  $x$ , então um conjunto nebuloso  $A$  em  $X$  é definido como um conjunto de pares ordenados de acordo com a equação 4.1:

$$A = (x, \mu_A(x) | x \in X) \quad (4.1)$$

onde  $\mu_A(x)$  é **Função de Pertinência** ou **Função de Inclusão** que define o conjunto nebuloso  $A$ . Qualquer função de pertinência faz o mapeamento de cada elemento de  $X$ , no caso  $x$ , para um grau de pertinência conveniente no intervalo entre 0 e 1. A definição de conjuntos nebulosos é uma simples extensão da definição dos conjuntos clássicos nos quais a função característica possui valores entre 0 e 1.

### 4.3 Tipos de Função de Pertinência

Uma função na forma  $A = (x, \mu_A(x) | x \in X)$  descreve uma função de pertinência associada com um conjunto nebuloso  $A$ , este não depende somente de um conceito para ser representado, mas também de um contexto no qual ela é usada. Os gráficos da função podem ter diferentes formas podendo ter alguma propriedade específica se uma forma particular é apropriada e pode ser determinada somente na aplicação do contexto. Em muitas instâncias práticas, conjuntos nebulosos podem ser representados explicitamente por conjuntos ou famílias de funções parametrizadas. Abaixo seguem as funções mais usadas com seus respectivos gráficos como mostra figura 10:

1. **Função Triangular:** esse tipo de função exige três parâmetros  $(a, b, c)$  como mostra a equação 4.2:

$$\text{Triângulo}(x; a, b, c) = \begin{cases} 0 & \text{se } x < a \\ \frac{x-a}{b-a} & \text{se } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{se } b < x < c \\ 0 & \text{se } c \leq x \end{cases} \quad (4.2)$$

2. **Função Trapezoidal:** esse tipo de função exige quatro parâmetros  $(a, b, c, d)$  como mostra a equação 4.3

$$\text{Trapézio}(x; a, b, c, d) = \begin{cases} 0 & \text{se } x < a \\ \frac{x-a}{b-a} & \text{se } a \leq x \leq b \\ 1 & \text{se } b < x < c \\ \frac{d-x}{d-b} & \text{se } c \leq x < d \\ 0 & \text{se } x \leq d \end{cases} \quad (4.3)$$

3. **Função Gaussiana:** esse tipo de função exige dois parâmetros  $(c, \sigma)$  como mostra a equação 4.4.

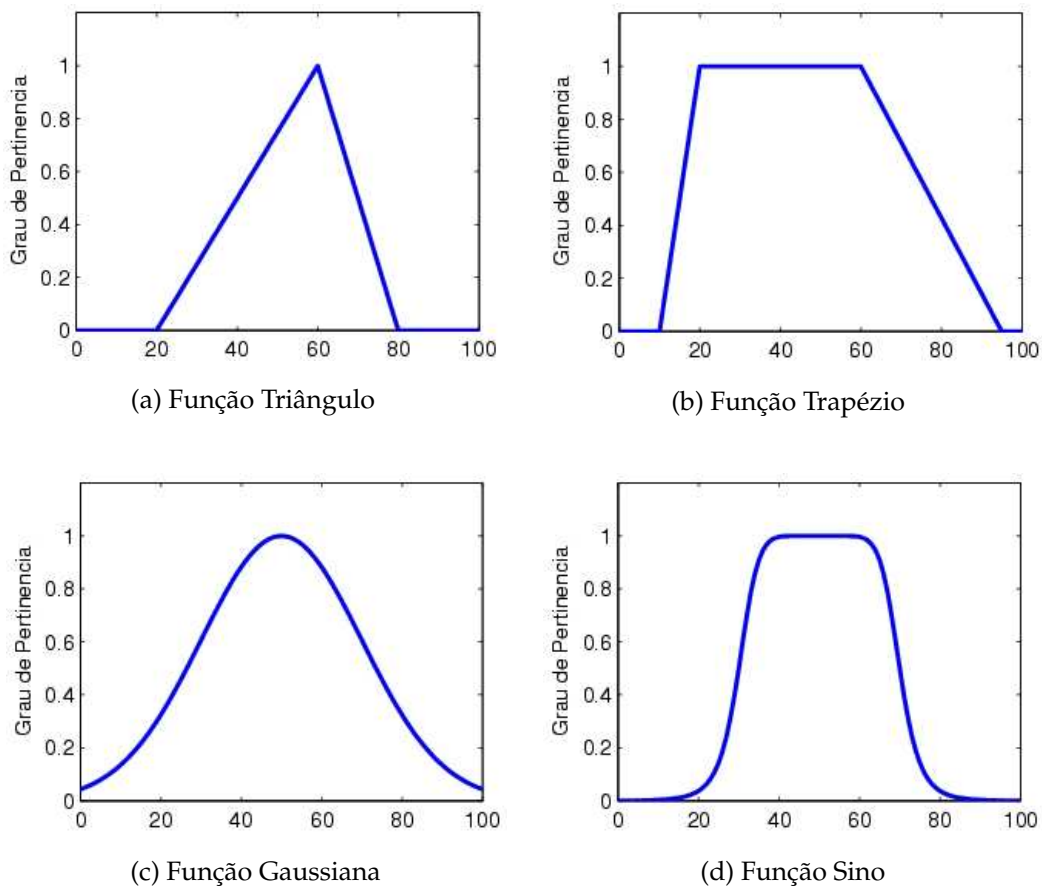


Figura 10: Funções de Pertinência

$$Gaussiana(c, \sigma) = e^{\frac{-0,5(x-c)}{\sigma^2}} \quad (4.4)$$

4. **Função Sino (Bell-Shape)**: esse tipo de função exige três parâmetros (a,b,c) como mostra a equação 4.5

$$Sino(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \quad (4.5)$$

## 4.4 Operações Nebulosas

As operações nos conjuntos nebulosos são similar aos conjuntos clássicos possuindo as operações de União , Interseção e Complemento. Por exemplo, considerando dois conjuntos nebulosos  $A, B$ , como mostra a figura 11(a), no universo  $X$ . Para um dado elemento  $x$  do universo, seguem as operações nebulosas mais comuns e a figura 11 mostra o resultado das operações sobre estes conjuntos.

1. **União:** a operação de união do conjunto  $A \cup B$ , como mostra figura 11(a), é definida como:

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x)) \quad (4.6)$$

2. **Interseção:** a operação de interseção do conjunto  $A \cap B$ , como mostra a figura 11(c), é definida como:

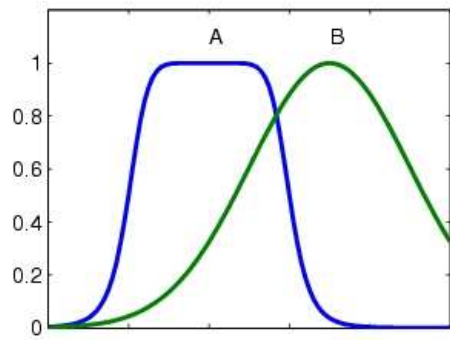
$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \quad (4.7)$$

3. **Complemento:** a operação de complemento do conjunto  $A$ , como mostra a figura 11(d), é definida como:

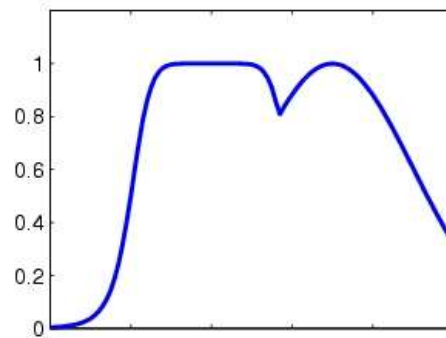
$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (4.8)$$

## 4.5 Variáveis Lingüísticas

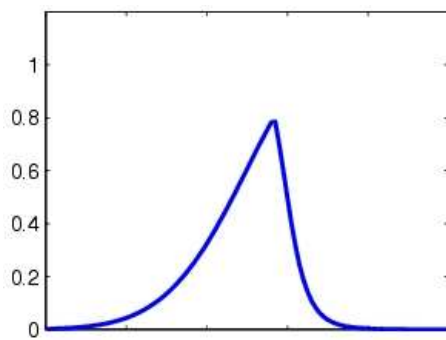
As variáveis lingüísticas têm como função a modelagem do pensamento humano. Variáveis Lingüísticas são caracterizadas por uma quintupla  $(x, T(x), X, G, M)$ , na qual  $x$  é o nome da variável;  $T(x)$  é conjunto de termos de  $x$ , isto é, o conjunto de seus valores lingüísticos ou termos lingüísticos;  $X$  é o universo do discurso;  $G$  é uma função sintática para geração dos termos em  $T(x)$ , e  $M$  é uma função



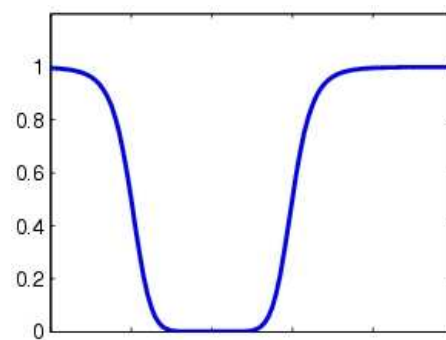
(a) Conjuntos A e B



(b) União de A com B



(c) Interseção de A com B



(d) Complemento de A

Figura 11: Operações com Conjuntos Nebulosos

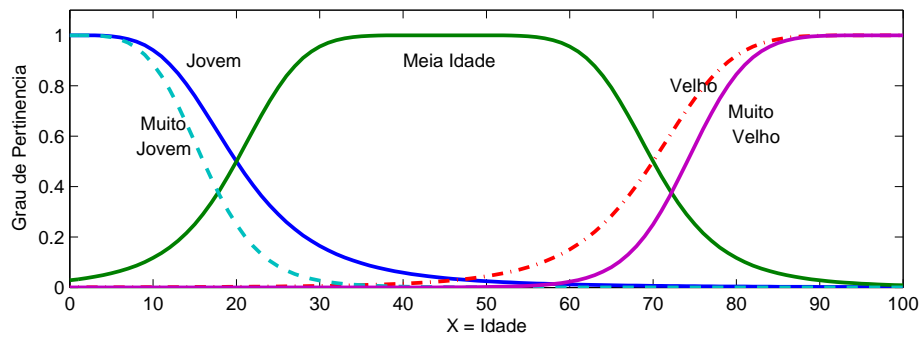


Figura 12: Gráfico das Variáveis Nebulosas

semântica na qual associa a cada valor lingüístico  $A$ , um conjunto nebuloso em  $X$ . Por exemplo, se idade é interpretada como uma variável lingüística, então, seu conjunto de termos  $T(idade)$  pode ser:

$$T(idade) = \{ jovem, muito jovem; \dots; meia\ idade; \dots; velho, muito\ velho \}$$

onde cada termo de  $T(idade)$  é caracterizado por um conjunto nebuloso em um universo de discurso  $X = [0, 100]$  como mostra figura 12.

Quando *idade* é interpretada como uma variável numérica, usa-se a expressão "idade=20" para designar o valor numérico 20 à variável numérica *idade*. A função sintática se refere à maneira como são gerados os termos de  $T(idade)$ . A função semântica define as funções de pertinência de cada valor lingüístico no conjunto de termos.

## 4.6 Fuzzificação, Regras Fuzzy e Defuzzificação

**Fuzzificação** é o processo de conversão de valores, ou seja, transforma uma quantidade de valores clássicos ou crisp em valores nebulosos, com objetivo de descobrir o grau de pertinência de  $\mu_{A_1}(x_1)$  e  $\mu_{A_2}(x_2)$  dadas as entradas  $x_1$  e  $x_2$  pertencentes aos Conjuntos Nebulosos  $A_1$  e  $A_2$  respectivamente. Isso é feito submetendo-se as entradas  $x_1$  e  $x_2$  a uma função de pertinência adequada

discutida na Seção 4.3.

As **Regras Nebulosas** consistem numa série de regras geradas com base no conhecimento adquirido. O relacionamento entre as entradas e as saídas são representados pelas regras que são expressas na forma de proposições<sup>1</sup>, tais como:

$$\text{Se } x \text{ A e/ou } x \text{ B Então } y \text{ C.} \quad (4.9)$$

Os operadores lógicos **e (and)** e **ou (or)** representam as operações nebulosas de união e interseção,  $x$  é um escalar do domínio e  $A, B$  e  $C$  são as variáveis lingüísticas. O efeito da proposição nebulosa  $x \text{ A}$  corresponde ao grau de pertinência de  $x$  no conjunto nebuloso  $A$ .

Diferente dos modelos convencionais de regras aos quais são verificados e executados sequencialmente as regras nebulosas tem como característica o paralelismo, ou seja, duas ou mais regras são executadas ao mesmo tempo pois um determinado escalar qualquer pode pertencer a mais de um conjunto nebuloso.

O processo de **Defuzzificação** é o inverso do processo de Fuzzificação. O processo de Defuzzificação consiste em converter os valores nebulosos em valores clássicos, e para que isso ocorra usamos alguns algoritmos para extrair esse valores. Os algoritmos mais comuns são: Área do Centróide, Área Bisetorizada, Mínimo do Máximo, Máximo do Mínimo e a Média Máxima. Para efeito de melhor entendimento foi escolhido apenas os algoritmos mais usados conforme mostra Kasabov (1998) que são:

1. **Centróide ou Centro da Gravidade (COG):** este método encontra o centro geométrico  $y$  de um conjunto nebuloso  $A$  dada uma variável nebulosa  $x$ . A equação 4.10 demonstra o COG.

---

<sup>1</sup>É uma sentença que pode ser verdadeira ou falsa.



$$y = \frac{\sum \mu_A(x) x}{\sum \mu_A(x)} \quad (4.10)$$

2. **Média Máxima (MOM):** este método encontra um valor  $y$  para uma variável nebulosa  $x$  de um conjunto nebuloso  $A$ . Nesse processo pega-se o maior valor de pertinência  $\mu_A(x)$ , se existir mais de dois valores que atinja o maior grau de pertinência, então, realiza-se a média entre os mesmo.

$$y = \frac{\sum \mu_A(x)}{n} \quad (4.11)$$

Onde  $n$  representa a quantidade de pertinencias existentes na saída de um sistema nebuloso.

## 4.7 Sistema de Inferência Nebuloso

Os Sistemas Nebulosos são sistemas baseados em regras que utilizam variáveis lingüísticas para executar um processo de tomada de decisão. Um Sistema de Inferência Nebuloso pode ser composto de cinco blocos principais que são:

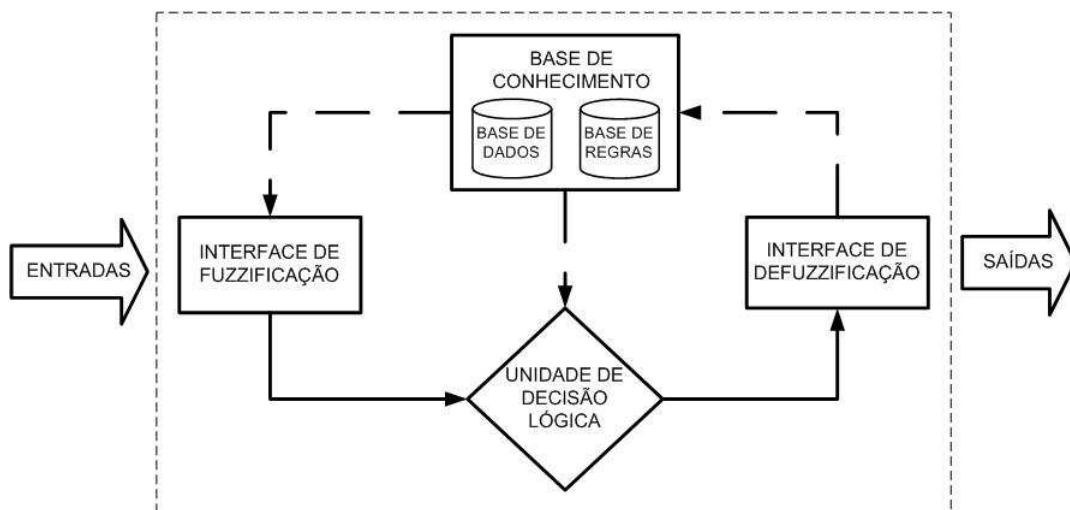


Figura 13: Esquema de um Sistema de Inferência Fuzzy

1. **Base de Regras:** contém um conjunto de regras/proposições nebulosas onde as variáveis antecedentes/consequentes podem ser variáveis linguísticas e os possíveis valores de uma variável linguística são representados por conjuntos nebulosos;
2. **Base de Dados:** define as funções de pertinência dos conjuntos nebulosos adequadas as regras nebulosas;
3. **Unidade de Decisão Lógica:** realiza operações de inferência, para obter, a partir da avaliação dos níveis de compatibilidade das entradas com as condições impostas pela base de regras, uma ação a ser realizada pelo sistema.
4. **Interface de Fuzzificação:** utilizando as funções de pertinência pré-estabelecidas, mapeia cada variável de entrada do sistema em graus de pertinência de algum conjunto nebuloso que representa a variável em questão.
5. **Interface de Defuzzificação:** realiza a conversão de dados nebulos em escalares de saída, ou seja, transforma dados nebulosos em crisp. Calcula a saída com base na inferência obtida no módulo Unidade de Decisão Lógica, com as funções de pertinência das variáveis linguísticas da parte consequente das regras para obter uma saída não nebulosa transforma os resultados nebulosos da inferência em valores de saída.

No Sistema de Inferência Nebuloso, consideram-se entradas **não-nebulosas**, ou precisas - resultantes de medições ou observações (conjuntos de dados de carga, por exemplo), que é o caso da grande maioria das aplicações práticas. Em virtude disto, é necessário efetuar-se um mapeamento destes dados precisos para conjuntos nebulosos (de entrada) relevantes, o que é realizado no estágio de fuzzificação. Neste estágio, ocorre também a ativação ou execução das regras relevantes para uma dada situação.

Uma vez obtido o conjunto nebuloso de saída através do processo de inferência *modus ponens* generalizado <sup>2</sup>, no estágio de defuzzificação é efetuada uma interpretação dessa informação. Isto se faz necessário pois, em aplicações práticas, geralmente são requisitadas saídas precisas.

As regras podem ser **fornecidas por especialistas**, em forma de sentenças lingüísticas, e se constituem em um aspecto fundamental no desempenho de um sistema de inferência nebulosa. E este somente terá um bom desempenho se as regras que definem a estratégia de controle forem consistentes (COX, 1994). Extrair regras de especialistas na forma de sentenças do tipo *se...então* pode não ser uma tarefa fácil, por mais conhecedores que eles sejam do problema em questão. Alternativamente ao uso de especialistas para a definição da base de regras, existem métodos de extração de regras de dados numéricos. Estes métodos são particularmente úteis em problemas de classificação e previsão de séries-temporais.

No estágio de inferência, ocorrem as operações com conjuntos nebulosos propriamente ditas: combinação dos antecedentes das regras, implicação e *modus ponens* generalizado. Os conjuntos nebulosos de entrada, relativos aos antecedentes das regras, e de saída, referente ao conseqüente, podem ser definidos previamente ou, alternativamente, gerados automaticamente a partir dos dados.

Um aspecto importante é a definição dos conjuntos nebulosos correspondentes às variáveis de entrada (antecedentes) e a(s) de saída(s) (consequente(s)), pois o desempenho do sistema de inferência **dependerá do número de conjuntos e de suas formas** (COX, 1994).

#### 4.7.1 Modelos de Inferências Nebulosas

O sistema de inferência nebuloso necessita de um mecanismo ou modelo de inferência para pode funcionar corretamente. Os modelos de inferência resume-

---

<sup>2</sup>Tipo de argumento em que as premissas "Se A, então B" e "A", implicam na conclusão: "B"

se em dois tipos: os **modelos clássicos**, que compreende o modelo de Mamdani e Larsen os quais **necessitam** de um defuzzificador, e os **modelos de interpolação**, que compreende os modelos de Takagi-Sugeno-Kang e Tsukamoto (SANDRI; CORREA, ) os quais **não necessitam** de um defuzzificador. Abaixo segue os modelos mais usados nas aplicações (REZENDE, 2005):

1. **Modelo Mamdani:** é o modelo que o previsor TUPAN utiliza, foi proposto em meados de 1970 e foi por muitos anos um “padrão” para processamento do conhecimento. Esse modelo possui relações nebulosas tanto em seus antecedentes quanto em seu consequentes. Uma regra genérica desse modelo é mostrada em 4.12 e outra regra com modelo mais específica, em nível de linguagem de programação, é mostrada em 4.13.

*Se Temperatura = Aumentando Muito e Carga = Alta Então Carga = Muito Alta* (4.12)

$$if x_i = A_1 \text{ and } x_j = A_2 \text{ then } y_k = B_1 \quad (4.13)$$

2. **Modelo de Takagi-Sugeno-Kang (TSK):** esse modelo proposto em meados de 1980 e similarmente ao modelo anterior possui seus antecedentes nebulosos porém seus consequentes são formados por **equações paramétricas** relacionando as entradas e saídas do sistema como mostra 4.14.

$$if x_i = A_1 \text{ and } x_j = A_2 \text{ then } y = f(x_i, x_j, \dots, x_p) \quad (4.14)$$

Por meio de um índice de desempenho é que os parâmetros dos consequentes são ajustados, esse índice de desempenho é necessário para verificar a performance sistema e assim ajusta-los de acordo com a necessidade, por isso, não há necessidade do processo de defuzzificação. Tem-se como exemplo de índice de desempenho a minimização dos erros quadráticos (usado também em MLPs) ou mínimos quadráticos recursivos.

## 5 *Sistema Híbrido Neuro-Nebuloso*

Neste capítulo será apresentada uma solução híbrida para o processo de previsão de carga , ou seja, nessa solução serão usadas RNAs e Lógica Nebulosa.

### 5.1 **Introdução**

Conforme Rezende (2005) a principal idéia do desenvolvimento de um Sistema Híbrido Inteligente (SIH), nesse caso específico um Sistema Híbrido Neuro-Nebuloso (SHNN), é que uma única técnica, em razão de suas limitações e/ou deficiências pode não ser capaz de, por si só, resolver um determinado problema. Assim, a combinação de várias técnicas pode levar a uma solução mais robusta e eficaz.

Um SHNN é uma área de pesquisa relativamente nova, a qual envolve tecnologias de RNAs e Lógica Nebulosa como informa Medeiros (1996) , podendo também envolver outras técnicas como Algoritmos Genéticos (AG) para manipulação do conhecimento como mostra Ling et al. (2003), Azevedo, Brasil e Oliveira (2000).

Estudos realizados sobre a eficiência e eficácia das RNAs, no processo de previsão de carga no mercado de energia elétrica, é conhecido na literatura de sistemas de potência como mostram Hsu e Yang (1991), Senjyu et al. (2002),

Khotanzad, Zhou e Elragal (1997, 1998). Porém, a velocidade com que o mercado de energia elétrica muda é muito grande, fazendo com que a performance das RNAs se degrade como mostram Khotanzad, Zhou e Elragal (2002), Shao e Sun (1997). Recentes avanços na área de Lógica Nebulosa e um número de aplicações do mundo real realizados com sucesso mostram que essa tecnologia pode ser aplicada em problemas onde há imprecisão de ambigüidade dos dados (KOSKO, 1997, 1992).

Combinando-se essas duas tecnologias como mostra a figura 14, pode-se realçar a capacidade do SHNN de aprender com a experiência e se adaptar melhor às mudanças do ambiente, manipular dados tanto quantitativos quanto qualitativos, incertos e imprecisos, na **tentativa** de melhorar a performance da previsão de carga.

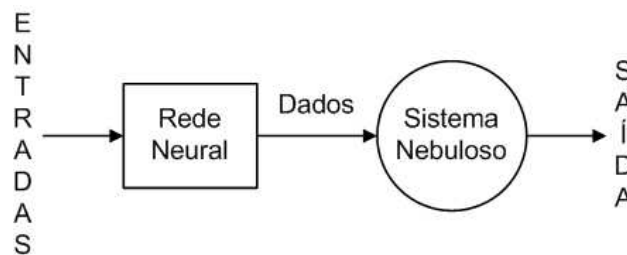


Figura 14: Esquema de um Modelo SHNN

## 5.2 Classificação dos SIH

De acordo com Rezende (2005), Liew e Srinivasan (1998) os SIHs obedecem um esquema de classificação que levam em consideração os seguintes aspectos: **funcionalidade, arquitetura de processamento e pré-requisitos de comunicação**. Deve-se lembrar que esse esquema de classificação não é restrito apenas para sistemas que usam MLPs e Sistemas Nebulosos, porém, para qualquer outro sistema que se deseje hibridar. As classificações que devem ser levadas em consideração são:

1. **Substituição de Função:** utiliza-se essa técnica para implementar uma função de outra técnica. Nesse tipo de hibridismo tenta-se superar a limitação de uma técnica acrescentando-se a funcionalidade de outra, como por exemplo, substituir a função de ativação das MLPs por uma função de pertinência proveniente da Lógica Nebulosa
2. **Híbridos Intercomunicativos:** essa é a categoria ao qual o previsor TUPAN está inserido. É formado por módulos independentes , nos quais cada módulo usa uma técnica inteligente para resolver uma parte do problema. Esse tipo de hibridação é usado quando o problema é complexo e sua solução pode ser alcançadas dividindo-se em subtarefas que serão resolvidas por técnicas de inteligente distintas mas interdependentes pois as mesmo se comunicam.
3. **Híbridos Polimórficos:** nesse tipo de hibridação, adapta-se uma técnica para solucionar problemas de outra técnica. Pode-se citar como exemplo o uso de RNAs raciocínio simbólico.

## 5.3 O Previsor TUPAN

O previsor TUPAN <sup>1</sup> foi desenvolvido usando a ferramenta *Borland Delphi 7* para implementar MLPs e Lógica Nebulosa os quais foram implementados usando a Programação Orientada a Objeto (POO). O previsor TUPAN consiste no módulo principal do programa de previsão de carga como mostra a figura15 e foi inspirado no modelo de Khotanzad, Zhou e Elragal (2002).

O TUPAN possui alguns recursos implementados como mostra a barra de *menu* ilustrada na figura 16. Essa barra de *menu* foi criada para facilitar o processo de previsão de carga e esses recursos (*menus*) consistem nos seguintes:

- **Menu Previsor:** possui as rotinas de administração do sistema como por

---

<sup>1</sup>Nome Tupí que significa Deus do Trovão



Figura 15: Tela de Início do Previsor TUPAN

exemplo: fechar rede, deletar rede, etc. E também rotinas relativos as redes criadas, ou seja, carregar determinada rede e dados para fazer a previsão propriamente dita;

- **Menu Assistentes:** são dois assistentes MARAN e XAMAN que tem por objetivo auxiliar o usuário na construção das redes;
- **Menu Ajuda:** é um tutorial feito em *html*<sup>2</sup> que auxília o usuário como utilizar o programa. No entanto, para usar o tutorial o usuário deve estar conectado a internet pois esse tutorial está em constantes atualizações;
- **Menu Sobre:** possui informações o sobre a versão do programa e também sobre os envolvidos na construção e atualização do programa;

---

<sup>2</sup>É uma linguagem utilizada para transmitir documentos formatados através da rede.



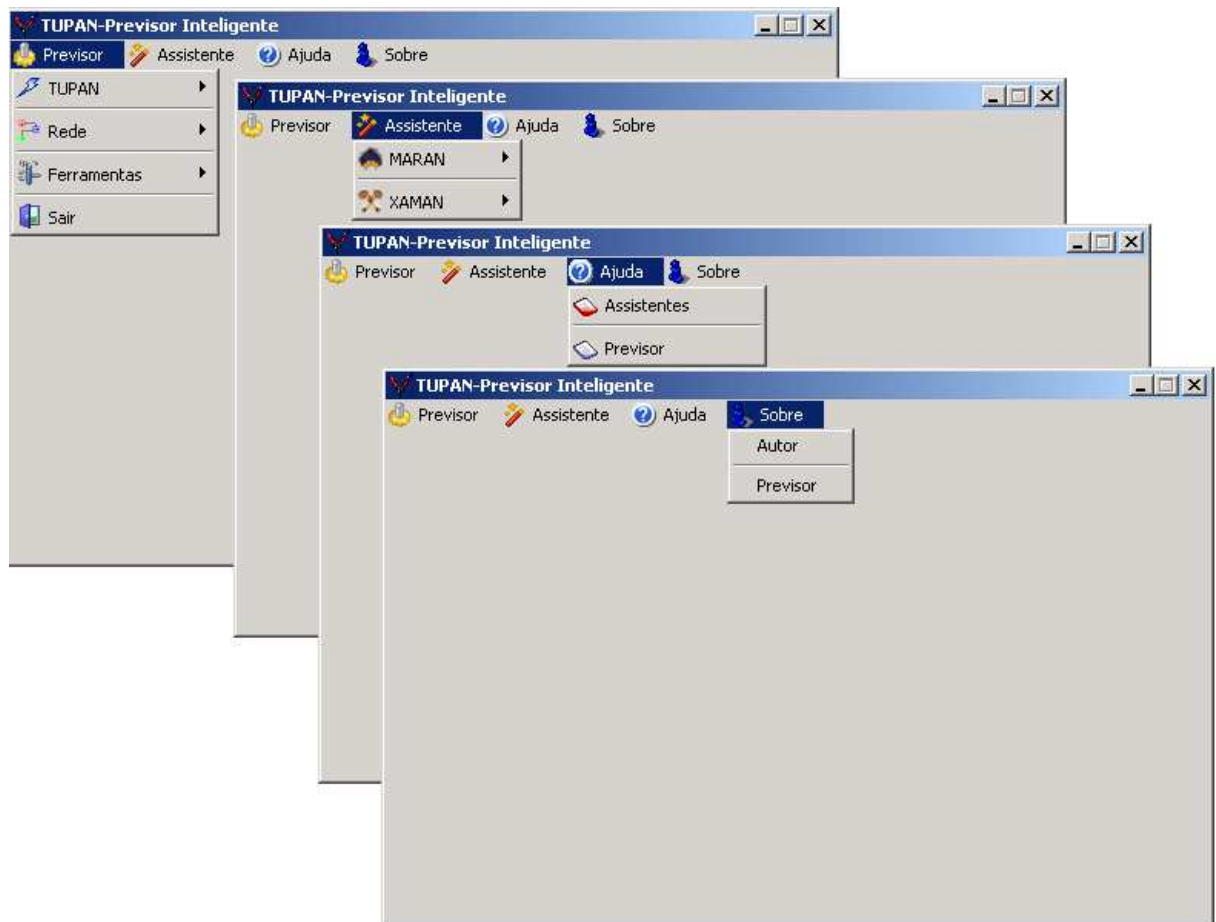


Figura 16: Telas das Rotinas Implementadas no TUPAN

### 5.3.0.1 Padrão de Leitura do TUPAM

O TUPAN possui um padrão de leitura de dados específico para que possa funcionar corretamente. O tipo de arquivo usado pelo previsor é o **arquivo texto**, já que esse tipo de arquivo pode ser aberto em qualquer editor e com relação ao padrão adotado para leitura de dados é mostrado na tabela 1

<b>Entradas</b>	Dado <sub>1</sub>	Dado <sub>2</sub>	...	Dado <sub>j</sub>
Exemplo 1	121	547	...	789
Exemplo 2	145	753	...	147
⋮	⋮	⋮	⋮	⋮
Exemplo <sub>p</sub>	369	256	...	254

Tabela 1: Exemplo de Entradas Padrão de Leitura do Previsor

Esse padrão de leitura deve ser seguido pois cada linha da tabela corresponde a um exemplo, no caso uma série-temporal, e cada coluna corresponderá a um neurônio na camada de entrada e saída da rede. Por exemplo: tem-se uma tabela de dados de entrada com vinte linhas e quatro colunas e uma tabela de dados de saída com vinte linhas e duas colunas, logo, essa rede terá vinte exemplos no total com quatro neurônios na camada de entrada e dois neurônios na camada de saída.

Com relação as entradas dos dados nebulosos, no caso as regras nebulosas, o procedimento muda. A tabela 2 mostra um exemplo de padrão de entrada das regras nebulosas, que serão lidas pelo TUPAM assim como pelo Assitente XAMAN, já que, esse último usa Lógica Nebulosa para criação das redes.

; Carga Alta	; Temperatura Alta	; Carga Normal
; Carga Baixa	; Temperatura Baixa	; Carga Alta
; Carga Normal	; Temperatura Normal	; Carga Baixa

Tabela 2: Padrão de Entradas das Regras Nebulosas

As Regras Nebulosas também precisam de um padrão. O padrão de arquivo contendo as Regras Nebulosas é o arquivo texto (\*.tx) onde as regras deverão seguir o padrão de leitura como mostra a tabela 3.

Antecedente(s)		Conseqüente
Proposição 1	Proposição 2	Proposição 3
; Carga Alta	; Temperatura Alta	; Carga Normal
; Carga Baixa	; Temperatura Baixa	; Carga Alta
;Carga Normal	; Temperatura Normal	;Carga Baixa

Tabela 3: Esquema de Montagem das Regras Nebulosas

O padrão de escrita das regras começam sempre com ponto e vírgulas (;) indicando sempre **início** das proposições e as váriáveis linguísticas (Carga e Temperatura) devem sempre ser acompanhada com seu qualificadores (Alto,Baixo e Normal) separados por um **único** espaço como mostra a figura 17.

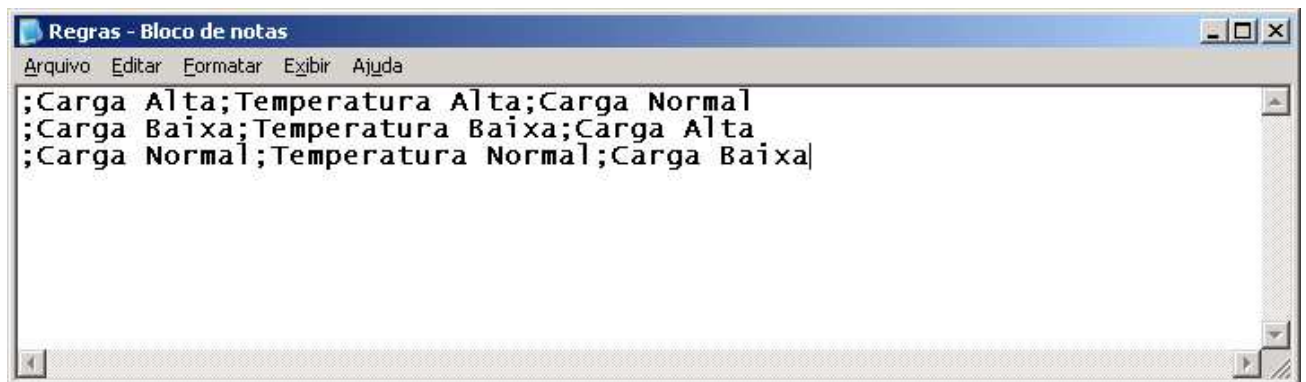


Figura 17: Arquivo Texto contendo Regras Nebulosas

### 5.3.1 O Processo de Previsão

O processo de previsão usando o TUPAN, consiste em carregar uma rede MLP ou um SHNN com seus respectivos pesos. Abaixo segue as janelas do TUPAN contendo os passos para o processo de previsão:

- 1ª Janela: essa é a janela inicial do TUPAM, onde escolhe-se qual tipo de rede o que se deseja usar/carregar. Nessa janela pode-se escolher uma MLP ou um SHNN como mostra figura 18;

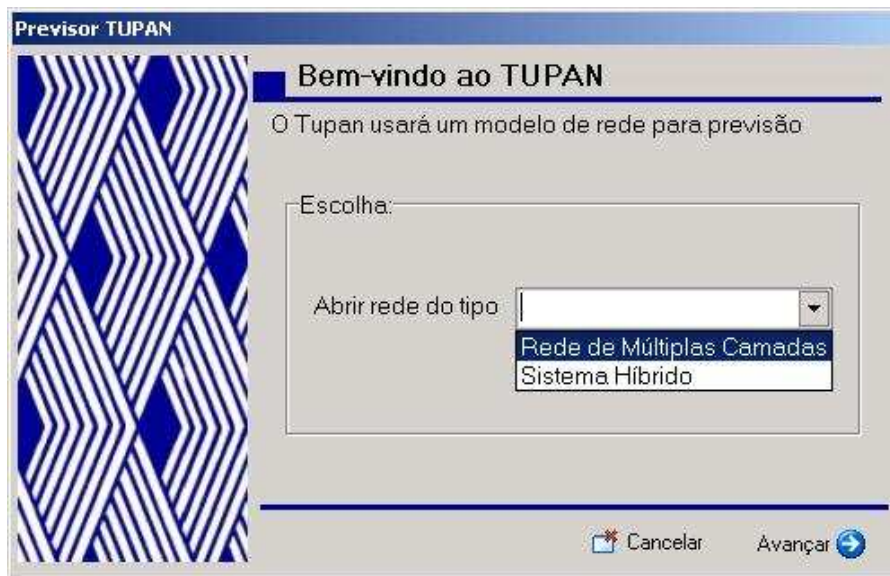


Figura 18: Janela de Inicial do TUPAM

- 2ª Janela: nessa janela deve-se informar os dados para previsão, isto é, os novos dados que serão propagados pela rede. A figura 19 mostra a aquisição dos dados;



Figura 19: Janela de Dados para Previsão

- 3ª Janela: essa janela é responsável pelas informações nebulosas, ou seja, nessa janela deve-se informar as regras nebulosas, dados correspondentes as variáveis linguísticas e parâmetros nebulosos, armazenados no banco de dados nebuloso, como mostra a figura 20;



Figura 20: Janela de Dados Nebulosos

- 4ª Janela: essa janela é responsável pela escolha da rede e dos pesos otimizados para poder ser carregado pelo TUPAN como mostra a figura 21. Para que essa operação tenha sucesso o TUPAN possui um filtro para distinguir os tipos de rede criada assim como os pesos otimizados de cada rede, ou seja, uma determinada rede não pode usar os pesos otimizados da outra;

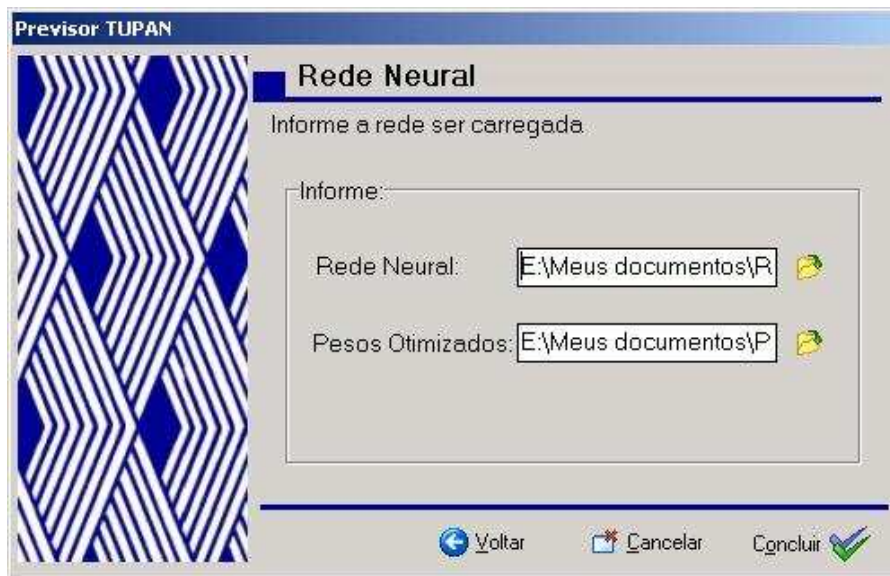


Figura 21: Janela de Escolha da Rede e Pesos

- 5ª Janela: essa é a última janela do TUPAN onde é mostrada de forma visual a curva de carga real, a curva de carga gerada pela rede, a prevista, e a curva de erro como mostra figura 22.



Figura 22: Janela Mostrando as Curvas de Carga e Erro

Como o previsor TUPAN usa MLPs e SHNN no processo de previsão de carga, então, tem-se duas operações distintas, e as janelas responsáveis em auxiliar a previsão de carga usando MLPs são: 1<sup>a</sup>, 2<sup>a</sup>, 4<sup>a</sup> e 5<sup>a</sup> janelas. E com relação a previsão de carga usando SHNN as janelas responsáveis são: 1<sup>a</sup> a 5<sup>a</sup> janelas.

## 5.4 Os Assistentes

Os Assistentes implementado no TUPAN são o Assistente MARAN<sup>3</sup> e o Assistente XAMAN<sup>4</sup> usam técnica de *Wizard*<sup>5</sup> e são responsáveis em auxiliar o usuário em criar e treinar uma rede.

### 5.4.1 O MARAN

O Assistente MARAN auxilia o usuário a criar e treinar MLPs, ou seja, gerência MLPs que serão usadas posteriormente no processo de previsão de carga e sua construção foi inspirada em Simon (1981) por possuir detalhes de grande valia. O MARAN usa o algoritmo de treinamento *BackPropagation*, possui implementado quatro algoritmos<sup>6</sup> de otimização dos pesos. O funcionamento do MARAN é ilustrado na figura 23

---

<sup>3</sup>Nome Tupí que significa guerra, aquele que faz guerra.

<sup>4</sup>Nome Tupí que significa mágico, aquele que domina os elementos.

<sup>5</sup>Técnica que divide os dados do problema em várias janelas ou formulários.

<sup>6</sup>Regrada Delta, *Momentum*, *Weight Decay* *Trainbpx*

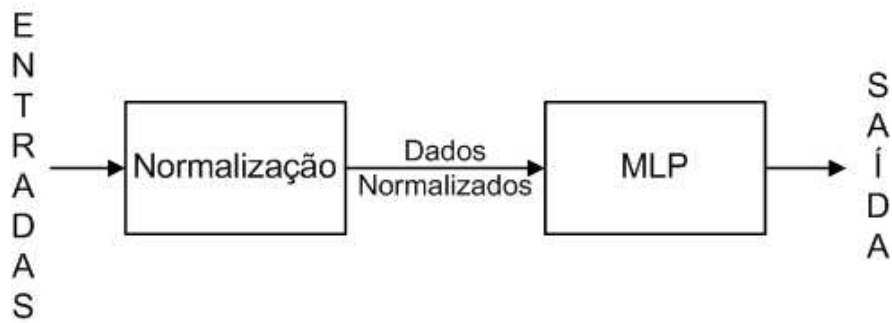


Figura 23: Esquema de Funcionamento do MARAN

O MARAN possui dois módulos, o primeiro é utilizado para **construir** uma MLP e o segundo é usado para **treinar** MLPs que serão descritos a seguir.

### Módulo de Construção de MLPs

Esse módulo de construção consiste de três janelas ou formulários que serão descritas a seguir:

- 1ª Janela: é a janela de boas-vindas do MARAN, nela poderá ser feita a escolha de **abrir** uma rede ou **criar** uma nova rede como ilustra a figura 24



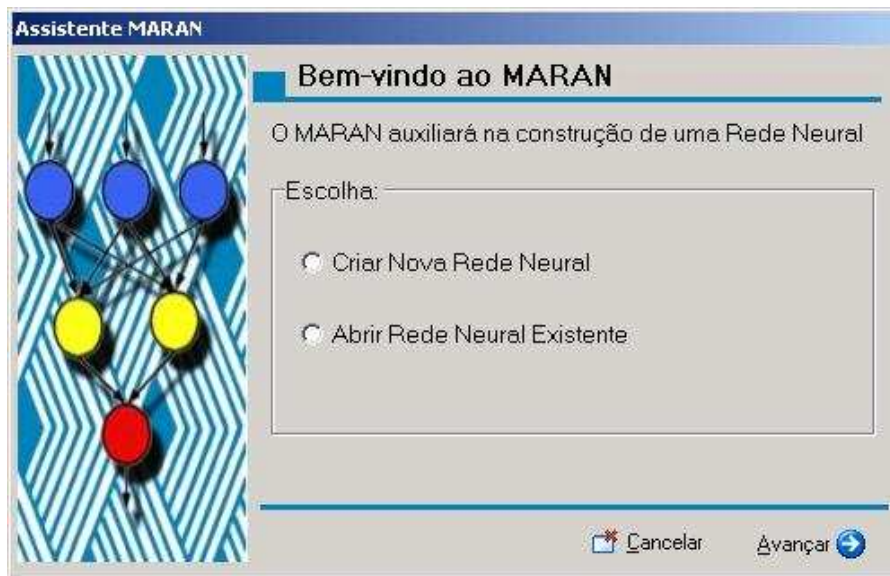


Figura 24: Janela de Início do MARAN

- 2ª Janela: nessa janela deverá ser informado os dados de entrada e saída da rede que deve estar no padrão de leitura TUPAM e no formato texto (\*.txt) e também o tipo de normalização que esses dados devem ser submetidos como ilustra a figura 25;



Figura 25: Janela de Dados do MARAN

- 3ª Janela: essa é a última janela a qual finalizará o processo de construção de uma MLP, nessa janela deve-se adicionar camadas de neurônios e depois definir que tipo de função de ativação específica que cada camada com seus respectivos neurônios deverá ter como ilustra figura 26.

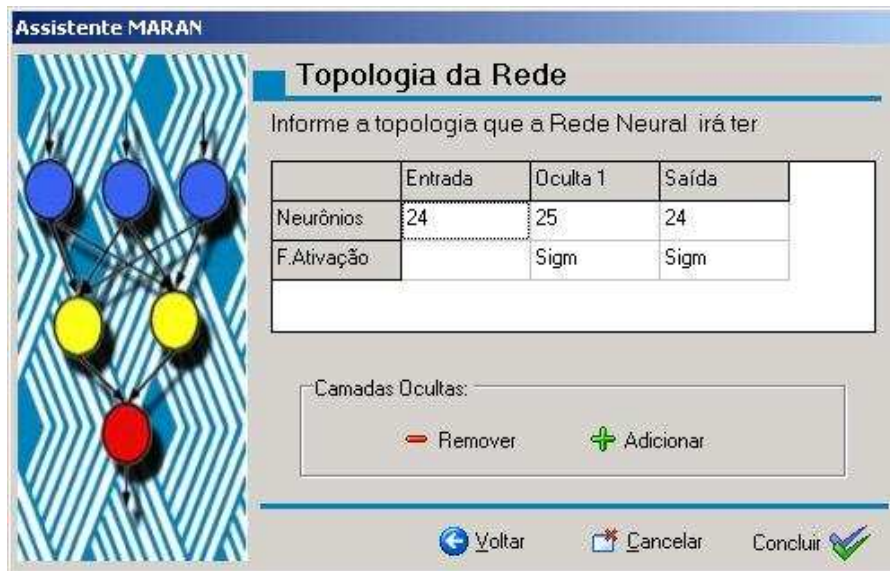


Figura 26: Janela de Adicionar Neurônios e Camadas do MARAN

Finalizado o processo de criação de uma MLP será informado um resumo da MLP criada como mostra a figura 27.



Figura 27: Resumo da MLP Criada

### Módulo de Treinamento

O módulo de treinamento consiste em duas janelas descritas a seguir:

- 1ª Janela: nessa janela tem-se as opções de **criar** um novo treinamento ou **abrir** um treinamento anterior como mostra a figura 28;



Figura 28: Janela de Treinamento do MARAN

- 2ª Janela: essa é a última janela para pode iniciar o treinamento da rede. Nessa janela deverá ser informado os **parâmetros de treinamento** da rede que são: número de ciclos, taxa de aprendizagem, erro máximo permitido, algoritmo de otimização. Sendo que o número de ciclos e erro máximo permitido são as **condições de parada** do treinamento como mostra a figura 29



Figura 29: Janela de Parâmetros de Treinamento do MARAN

Após ser informado os parâmetros de treinamento, abrirá uma nova janela com mostra a figura 31 onde deverá ser informado o nome do arquivos que conterá os **pesos otimizados** da rede a ser treinada para mais tarde serem carregados pelo TUPAN. Caso o treino da rede proceda tenha êxito, o sistema informará o sucesso do treinamento como mostra a figura 30.



Figura 30: Janela que Informa o Sucesso do Treinamento



Figura 31: Janela para Salvar os Pesos Otimizados

### 5.4.2 O XAMAN

O Assistente XAMAN possui as características dos Assistente MARAN<sup>7</sup> acrescentado de um Sistema Nebuloso caracterizando-o como um SHNN. A saída da RNA servirá de entrada no Sistema Nebuloso (passando antes por um conversor de dados) e a saída do Sistema Nebuloso será a saída do sistema, ou seja, são os dados previstos. A figura 32 mostra o funcionamento do XAMAN.

<sup>7</sup>A nível de POO o XAMAN herda as características do MARAN

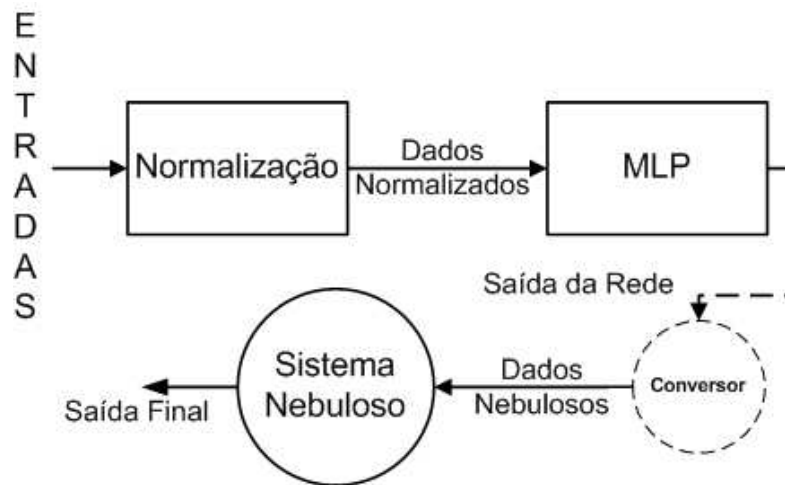


Figura 32: Esquema de Funcionamento do XAMAN

### Banco de Dados do XAMAN

O XAMAN possui um banco de dados para armazenar as dados correspondentes as informações Nebulosas. Esse banco de dados foi implementado em *Microsoft Acess 2003*, como mostra figura 33, devido ao fácil manuseio e implementação. O XAMAN usará esse banco de dados apenas para consultas.

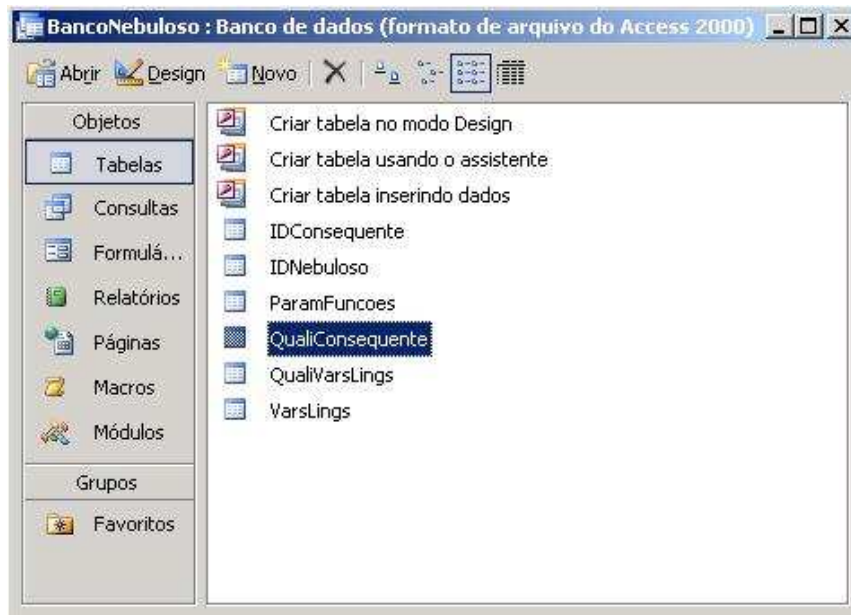


Figura 33: Janela do Microsoft Access

Nas Regras Nebulosas há os **antecedentes** e os **consequentes** na implementação do banco de dados os antecedentes geraram quatro tabelas como mostra o relacionamento da figura 34 e os consequentes geraram duas tabelas como o relacionamento da figura 35

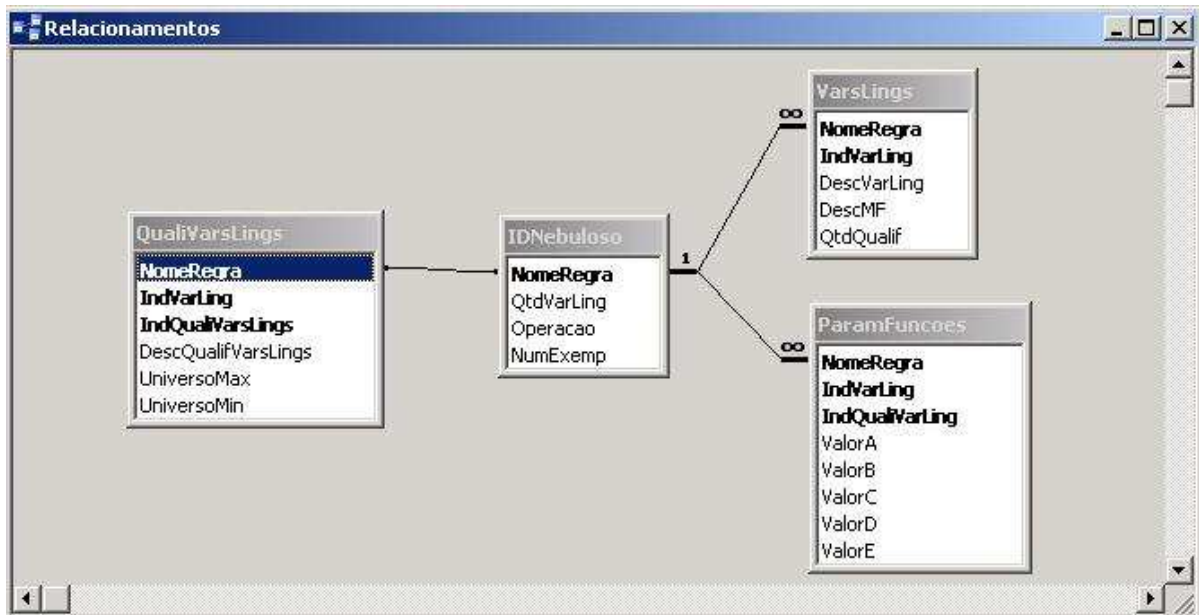


Figura 34: Relacionamento das Tabelas dos Antecedentes

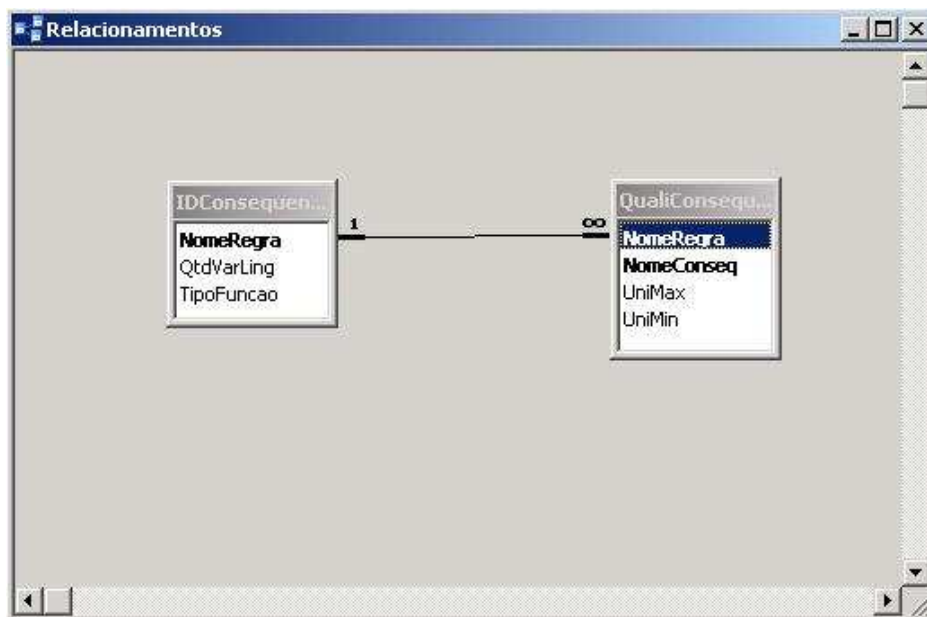


Figura 35: Relacionamento das Tabelas dos Consequentes

Com relação as tabelas correspondentes aos antecedentes mostrados na figura 34, tem-se:



1. **Tabela IDNebuloso:** essa tabela é responsável pelas primeiras informações, ou seja, nela será armazenada dados como: Nome da Regra (que é chave primária), Quantidade de Variáveis Lingüísticas, Operação Nebulosa e a Quantida de Exemplos;
2. **Tabela VarsLings:** nessa tabela será armazenado dados correspondentes às informações de **cada** variável lingüística envolvida tais como: Descrição da Variável, Tipo de Função de Pertinência, Quantidade Qualificadores;
3. **Tabela QualiVarsLings:** nessa tabela será armazenado dados correspendentes a cada qualificador proveniente de cada variável lingüística , essa tabela possuirá dados tais como: Descrição do Qualificador, Universo do Discurso (valor máximo e mínimo);
4. **Tabela ParamFuncoes:** nessa tabela conterà os parâmetros das funções de pertinência de cada variável lingüística para que possa ser montado o gráfico de cada uma.

Em relação as tabelas do conseqüente mostrados na figura 35, tem-se:

1. **Tabela IDConequente:** nessa tabela será armazenado dados relativos ao conseqüente, tais como: Nome do Conseqüente,(que é a chave primária),Quantidade de variáveis linguísticas, Tipo de Função de Pertinência;
2. **Tabela QualiConsequeunte:** nessa tabela será armazenado os parametros relativos a cada conseqente, tais como: Nome do Conseqüente, Universo do Discurso (Valor máximo e mínimo).

### Módulo de Construção de um SHNN

O processo de construção de uma rede híbrida no Assistente XAMAN é muito parecido com o Assistente MARAN. No entanto, foram adicionados duas

janelas contendo informações sobre o Sistema Nebuloso resultando no total de cinco janelas que serão descritas a seguir:

- 1ª Janela: é a janela de boas-vindas do XAMAN nela pode-se escolher em **abrir** uma rede híbrida ou **criar** uma nova rede híbrida como ilustra a figura 36;

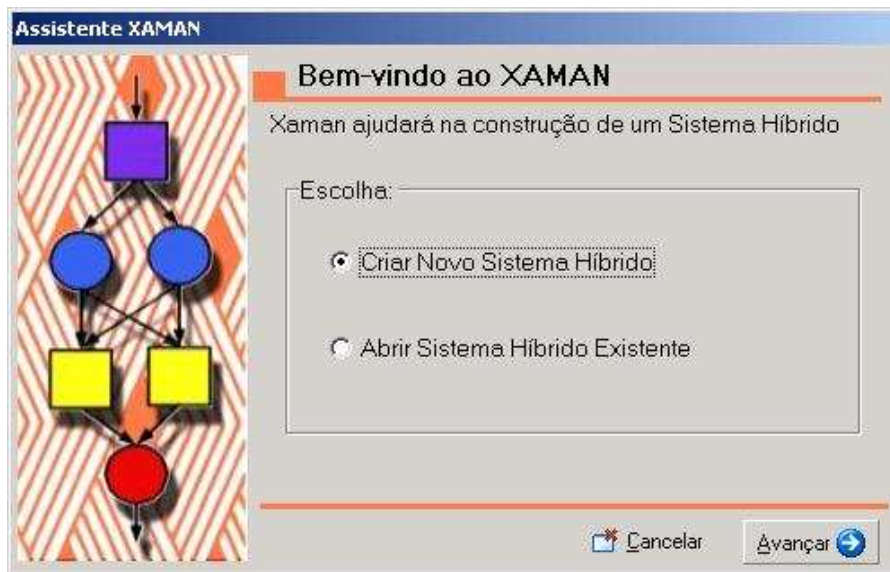


Figura 36: Janela de Início do XAMAN

- 2ª Janela: nessa janela deverá ser informado os dados de entrada e saída da rede e o tipo de normalização que os dados serão submetidos. Esse processo é igual ao Assistente MARAN como mostra a figura 37;



Figura 37: Janela de Dados do XAMAN

- 3ª Janela: essa janela contém as primeiras informações do Sistema Nebuloso, isto é, nessa janela deverá ser informado as Regras Nebulosas e também os dados correspondentes as variáveis lingüísticas que estão envolvidas como mostra figura 38;



Figura 38: Janela de Regras e Dados das Variáveis Lingüísticas

- 4ª Janela: essa janela também possui informações nebulosas. Nessa janela deverá ser informado os **parâmetros nebulosos**, definidos anteriormente no banco de dados, o tipo de inferência nebulosa e o tipo de defuzzificação como mostra a figura 39;



Figura 39: Janela de Dados para Defuzzificação

- 5ª Janela: essa é a última janela a qual finalizará o processo de construção de um SHNN, nessa janela deve-se adicionar camadas de neurônios e depois definir que tipo de função de ativação específica que cada camada, e seus respectivos neurônios, deverá ter como ilustra 40.



Figura 40: Janela de Adicionar Neurônios e Camadas do XAMAN

Finalizado o processo de criação de um SHNN será informado o resumo do SHNN criado como mostra a figura 41. Como o XAMAN utiliza Lógica Nebulosa, observa-se que no resumo do SHNN foi acrescentado mais informações: **Dados Nebulosos e Fonte Nebulosa**.

### Módulo Treinamento

O módulo de treinamento do Assitente XAMAN tem a funcionalidade do módulo de treinamento do Assistente MARAN, descrito anteriormente. O módulo de treinamento do Assitente XAMAN possui duas janelas descritas a seguir:

- 1ª Janela: nessa janela o usuário deve informar se deseja um novo treinamento ou abrir um treinamento existente como motra a figura 42



Figura 41: Resumo do SHNN Criado



Figura 42: Janela de Treinamento do XAMAN

- 2ª Janela: nessa janela deverá ser informado os parâmetros de treinamento como mostra a figura 43.



Figura 43: Janela de Parâmetros de Treinamento do XAMAN

Deve-se lembrar que no processo de treinamento da rede usando um SHNN, o Assistente XAMAN faz inúmeros acessos ao banco de dados nebuloso **escolhido no processo de criação** da rede tornando assim o treinamento um pouco mais lento.

O processo de salvar os pesos otimizados do Assistente XAMAN, resultante do treinamento da rede, é o mesmo processo do Assistente MARAN descritos anteriormente.

## 6 *Revisão sobre Previsão de Carga a Curto Prazo*

Nesse capítulo apresenta as características existentes em um processo de previsão de carga , a metodologia e modelagem usada para o processo de previsão.

### 6.1 **Previsão de Carga**

No Capítulo 2, foram citadas algumas características da importância da previsão de carga. O processo de previsão pode vir com os nomes de **predição** ou **projeção**, mas que também se refere à ação de se cogitar um ato futuro antes mesmo de acontecer. Paulo (2004) apresenta uma idéia bem explicada do que é previsão de carga:

(...) A previsão de carga não é simplesmente uma adivinhação da carga futura, ela se baseia num estudo dos dados históricos, onde são analisados e posteriormente alguma metodologia apropriada para previsão é aplicada sobre os mesmos. É importante salientar que a previsão não é uma ferramenta que lhe diga exatamente qual dado irá acontecer , mas tem o papel de dar suporte a uma tomada de decisão mais segura [para o operador do sistema evitando possíveis problemas]. Além disso, a previsão não se destina apenas à previsão de carga, ela pode ser uma ferramenta útil em diversas áreas afins que utilizem como alicerce as séries-temporais[estacionárias ou não-estacionárias](...). Os resultados obtidos com a previsão servem como um indicativo



da filosofia a ser seguida para o alcance do objetivo esperado.

## 6.2 Parâmetros para Previsão

No processo de previsão de carga, não basta que se tenha uma boa ferramenta de previsão, há necessidade que se tenha definido alguns dados relevantes para que o processo de previsão ocorra com êxito. Esses dados serão comentados a seguir.

### 6.2.1 Horizontes de Previsão

Os horizontes de previsão são intervalos de tempo utilizados para planejamento mais eficiente dos recursos dispostos. Os horizontes de previsão mais comuns são:

- **Previsão de Longo Prazo:** esse tipo de previsão é necessária para o planejamento do sistema de energia, como por exemplo, ordenar a expansão da capacidade do sistema para encontrar o valor esperado do crescimento da demanda. O intervalo de dados geralmente usado para previsão são **dados anuais** cerca de três a cinco anos que são usados para se prever o próximo ano;
- **Previsão de Médio Prazo:** esse tipo de previsão é feita para programar o suprimento de combustível, manutenção da operação e planejamento da interoperabilidade da transferência de energia. O intervalo de dados geralmente usado para previsão são **dados mensais** para poder se prever o próximo mês ;
- **Previsão de Curto Prazo:** esse tipo de previsão é usada na operação diária semelhante como uma unidade de alocação, ou seja, programa a

transferência de energia e administrando o tempo de demanda. O intervalo de dados geralmente usado para previsão são **dados diários** para poder se prever o próximo dia.

### 6.2.2 Perfil de Carga

O perfil de carga é de grande importância a ser definido, porque irá influenciar na performance da rede pois a partir dessa definição é que se pode escolher os dados mais apropriados que irão representar adequadamente o perfil escolhido, ou seja, o ambiente que a rede deverá aprender.

No processo de previsão de carga a curto prazo, existem vários perfis pois são definidos de acordo com a necessidade e a problemática que se está resolvendo. No entanto, conforme Liew e Srinivasan (1998), El-Sharkawi e Niebur (1996), Hippert, Pedreira e Souza (2001) os perfis mais simples usados em previsão de carga são: Segunda, Terça, Quinta, Sexta, Sábado, Domingo, Feriados, Dias Especiais (por exemplo: a final de um campeonato de futebol). Esses perfis podem conter seus históricos de dados diários, hora-a-hora e etc, isso dependerá da previsão a ser realizada.

### 6.2.3 Fonte de Dados

A fonte de dados usada no processo de treinamento e validação das redes, corresponde aos dados da Nova Inglaterra disponibilizados no site <http://www.iso-ne.com/>. Os anos usados foram: 1994 até 1996, onde o mercado de energia elétrica é pouco volátil; e os dados de 2001 até 2003, onde o mercado de energia elétrica possui grande volatilidade. Nesses anos continuam as informações de carga, preço, umidade, etc. Todos esses dados foram descritos **hora-a-hora**, originando assim uma grande quantidade de informações.

Esses dados foram submetidos a um processo de normalização, para que

os mesmos pudessem variar de acordo com o intervalo da função de ativação desejada da RNA, onde os intervalos variam de  $[0, 1]$  como mostra a equação 6.1 e  $[-1, 1]$  como mostra a equação 6.2. Abaixo segue a modelagem matemática para normalização dos dados.

$$x_i = \frac{(x_i - x_{min})}{(x_{max} - x_{min})} \quad (6.1)$$

$$x_i = \frac{2}{(x_{max} - x_{min})} x_i + \frac{(-x_{max} - x_{min})}{(x_{max} - x_{min})} \quad (6.2)$$

Onde as variáveis  $x_{max}$  e  $x_{min}$  referem-se a maior e menor valor encontrado dentro do **conjunto de treinamento** da rede respectivamente.

## 6.2.4 Medidas de Performance

Tendo realizado o treinamento da RNA, há necessidade de se verificar sua performance (acurácia) através de uma avaliação estatística podendo ser os seguintes critérios:

### 1. MAE (Mean Absolute Error)

Significa erro absoluto médio, definido pela média aritmética dos valores absolutos para que o erro positivo não seja anulado por um erro negativo como mostra a equação 6.3.

$$MAE = \frac{\sum_{i=1}^{NP} |E_i|}{NP} \quad (6.3)$$

Onde  $E_i$  representa os erros das previsões e  $NP$  representa o número das previsões.

### 2. MAPE (Mean Absolute Percentage Error)

Significa erro percentual médio absoluto, é o método mais utilizado para verificação de performance, define a média dos desvios entre os valores

observados e os valores previstos com mostra a equação 6.4.

$$MAPE = \frac{1}{NP} \sum_{i=1}^{NP} \left| \frac{o_i - p_i}{p_i} \right| \quad (6.4)$$

Onde,  $NP$  representa o número de previsões,  $o_i$  representa os dados observados,  $p_i$  representa os dados previstos ou desejados.

A escolha do critério estatístico para se medir a acurácia da rede não pode ser escolhido de forma aleatória, deve-se escolher de acordo com o problema em questão. Nesse caso o modelo mais usado no processo de previsão de carga usando SI é o MAPE (SENJYU et al., 2002; KHOTANZAD; ZHOU; ELRAGAL, 1997; EL-SHARKAWI; NIEBUR, 1996; MANN; F.GARCIA-LAGOS; SANDOVAL, 2002).

## 7 *Estudos de Caso*

Este capítulo aborda um problema do mundo real. O estudo e modelagem desse problema foi realizado através de dados disponibilizados pelo NEPOOL (*North England Pool*), responsável pelo fornecimento de energia da Nova Inglaterra, Estados Unidos.

### 7.1 Introdução

No processo de previsão de carga, foi utilizado o previsor TUPAN para realização dos testes, juntamente com seus assistentes MARAN e XAMAN. Os testes de previsão realizados consistem em prever um perfil de carga para um dia da semana que vai de **terça-feira** à **quarta-feira**. As variáveis de entrada da rede que foram levadas em consideração são: Carga, Bulbo Úmido, Bulbo Seco e Preço, descritos hora-a-hora de **todas** as terças-feiras e **cada série** contém **vinte e quatro** dados, pois o histórico foi descrito hora-a-hora.

Em relação aos parâmetros de treinamento, os seguintes dados foram considerados:

1. Número de ciclos: no processo de treinamento das redes, os dados escolhidos foram submetidos à rede, em média 2000 vezes, sendo cada exemplo fora escolhido de forma aleatória;
2. Critério de parada: consiste em duas condições a serem satisfeitas que são:

minimizar a função de custo até que o valor seja menor que 0,0013 como informa Kasabov (1998) ou que o número de ciclos chegue até o final;

3. Escolha da melhor rede: foi levada em consideração a rede cujo treinamento conseguiu minimizar função custo ao máximo, ou seja, a função deverá ter um valor menor que 0,0013.

Escolhidos os perfis de carga, variáveis e horizontes de previsão, começam-se os estudos. Nesse trabalho, foram criados quatcomo ro estudos de casos que são descritos a seguir.

## 7.2 Estudos de Casos Iniciais: Mercado de Energia Regulado

No mercado de energia regulado (1994 até 1996), o preço **não sofre** grandes variações de valores. O treinamento das redes e conseqüentemente a **obtenção dos pesos otimizados** das redes durante o treinamento será feito usando as séries antes da volatilidade do preço.

Serão apresentados dois estudos de caso, onde no primeiro estudo de caso serão usadas todas as séries e no segundo estudo de caso serão usadas apenas as séries de carga e preço. O objetivo desses estudos de caso é saber quais séries influenciam na curva de carga.

### 7.2.1 Estudo de Caso A

Nesse estudo de caso, foram usadas todas as séries, bulbo úmido, bulbo seco, carga e preço, com preço pouco volátil resultando assim em 96 entradas. A análise dos resultados foi feita através do MAPE como mostra a tabela 4 e os gráficos 44, 45, 46 e 47 são das previsões feitas de cada topologia de rede descritas na tabela 4.

- **MAPE das Previsões**

Topologia de Rede	MAPE(%)
96x18x24	6,618173538
96x20x24	3,971359904
96x22x24	6,54697102
96x40x24	2,122750353

Tabela 4: MAPE usando todas as Séries com Preço Pouco Volátil

- **Gráficos das Previsões**

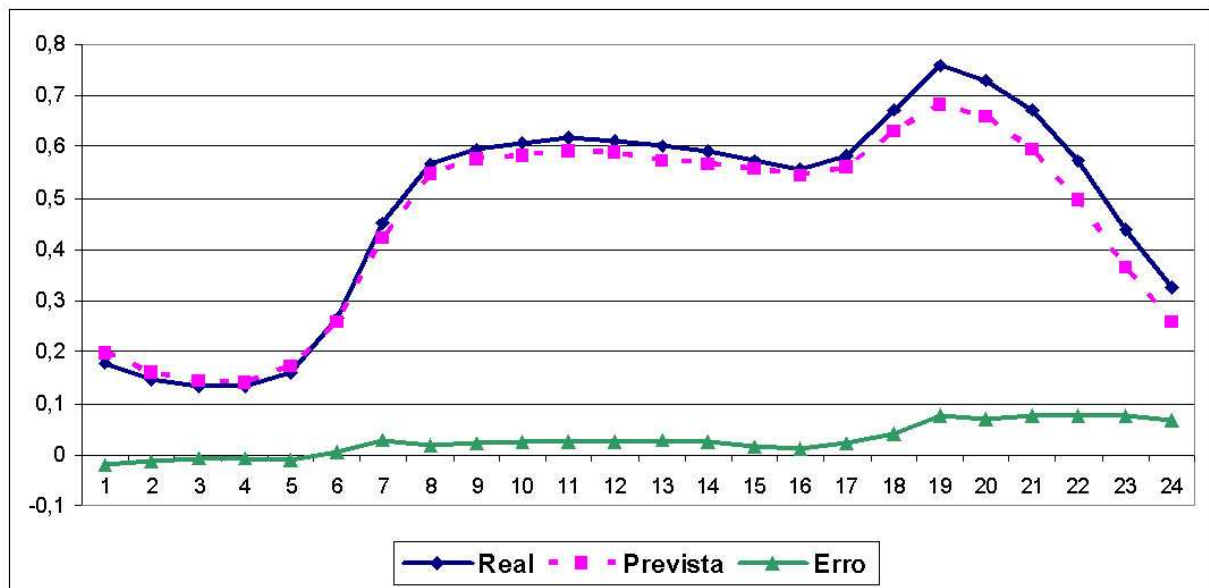


Figura 44: Gráfico da Rede 96x18x24 com Mercado Regulado

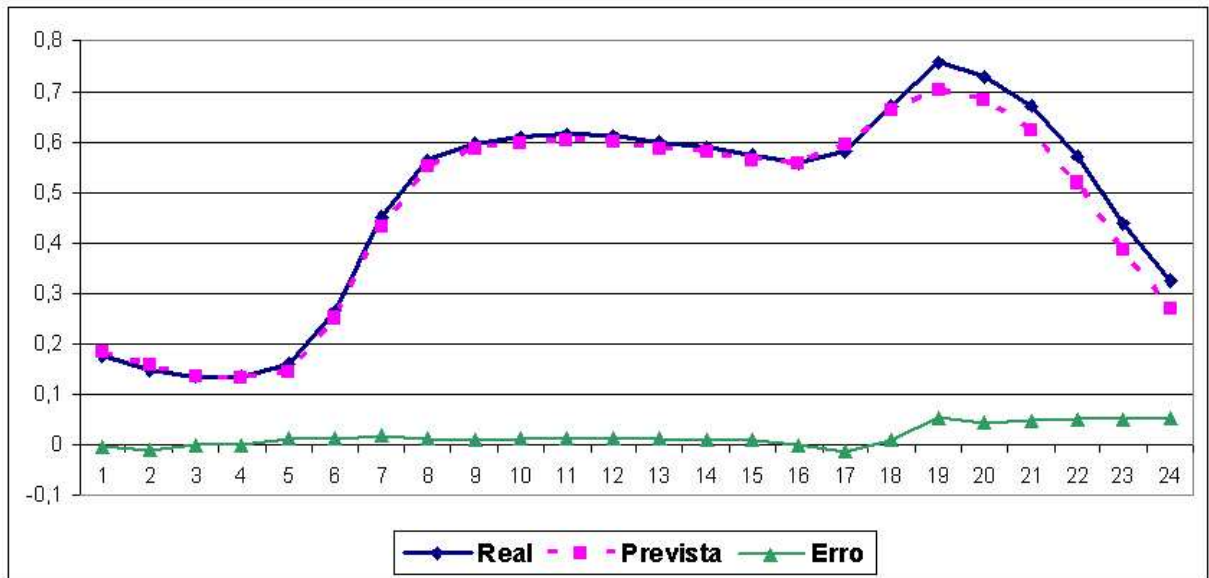


Figura 45: Gráfico da Rede 96x20x24 com Mercado Regulado

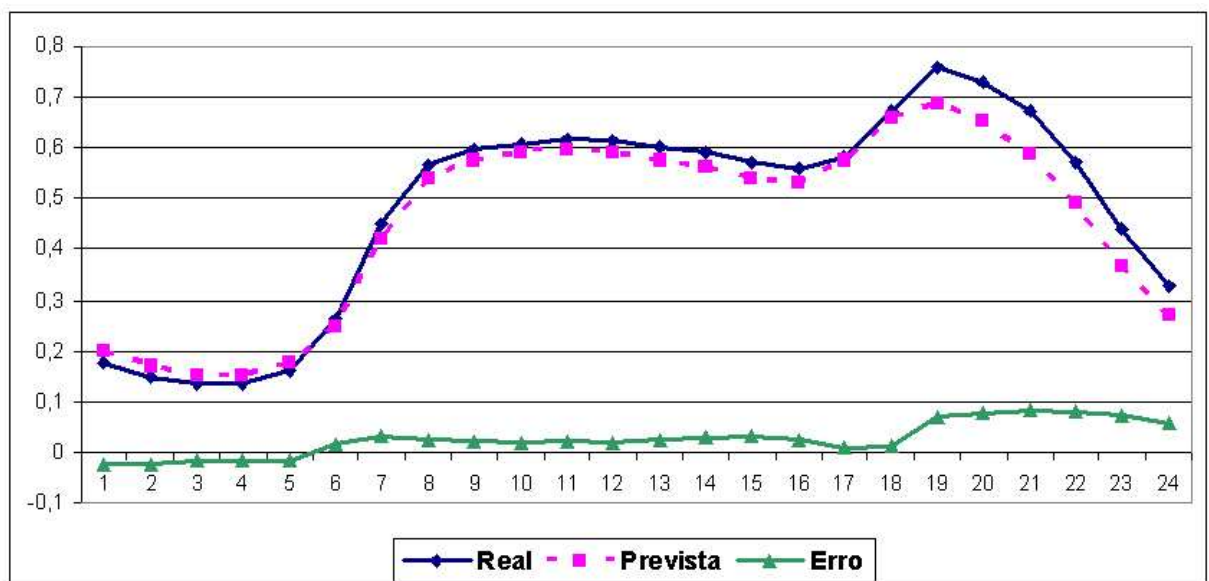


Figura 46: Gráfico da Rede 96x22x24 com Mercado Regulado



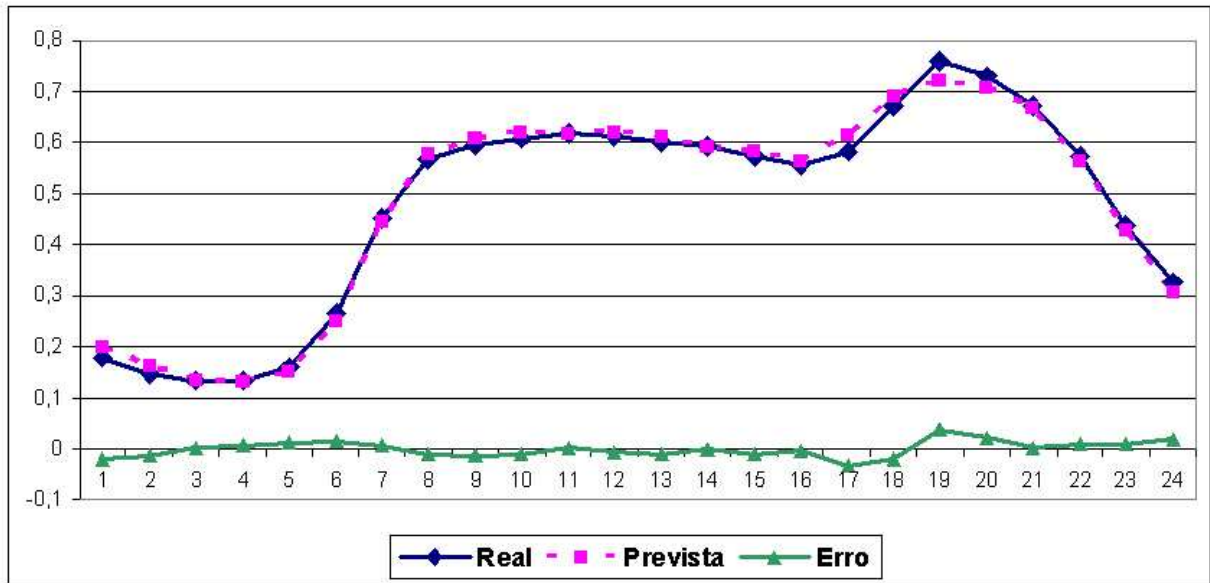


Figura 47: Gráfico da Rede 96x40x24 com Mercado Regulado

### 7.2.2 Estudo de Caso B

Nesse estudo de caso, foram utilizados apenas as séries de carga e preço pouco volátil, resultando assim em 48 entradas, na tentativa de diminuir o erro do estudo de caso A. A análise dos resultados foi feita através do MAPE como mostra a tabela 5 e as figuras 48, 49, 50 e 51 são as previsões de cada topologia de rede descritas na tabela 5.

- MAPE das Previsões

Topologia	MAPE(%)
48x18x24	2,989607
48x20x24	7,3744597
48x22x24	3,3514639
48x40x24	2,690512154

Tabela 5: MAPE usando Séries de Carga e Preço Pouco Volátil

• Gráfico das Previsões

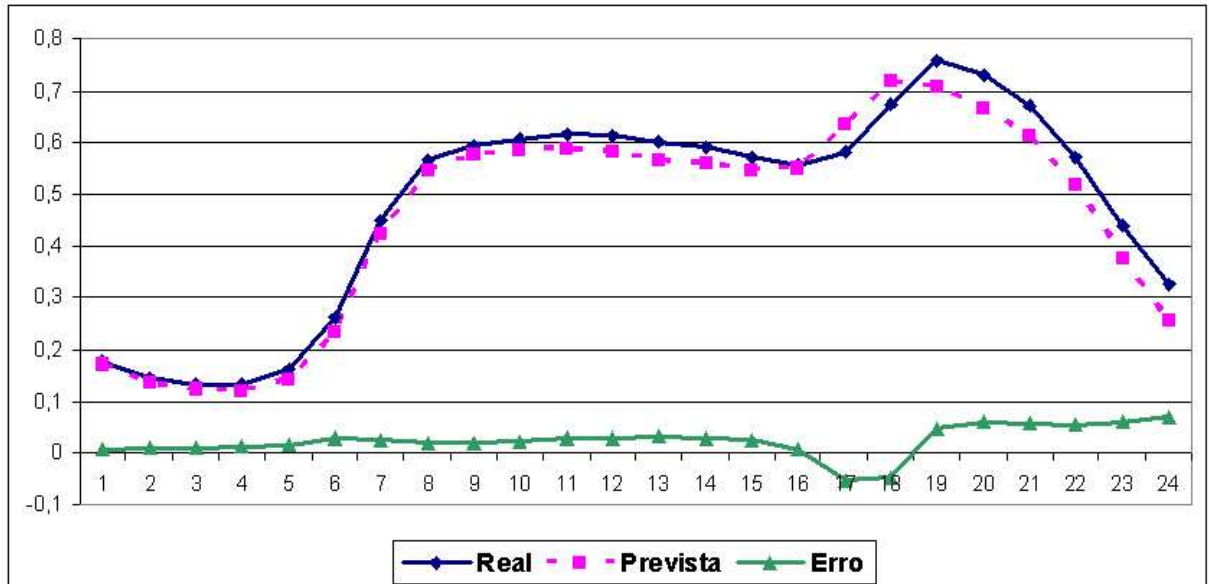


Figura 48: Gráfico da Rede 48x18x24 com Mercado Regulado

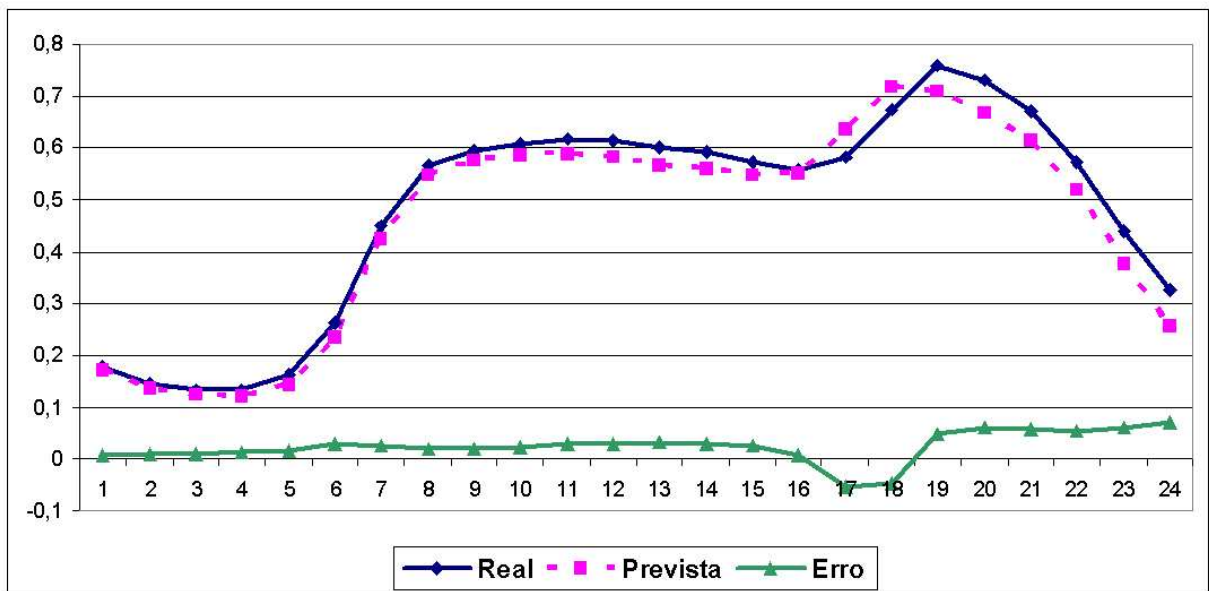


Figura 49: Gráfico da Rede 48x20x24 com Mercado Regulado

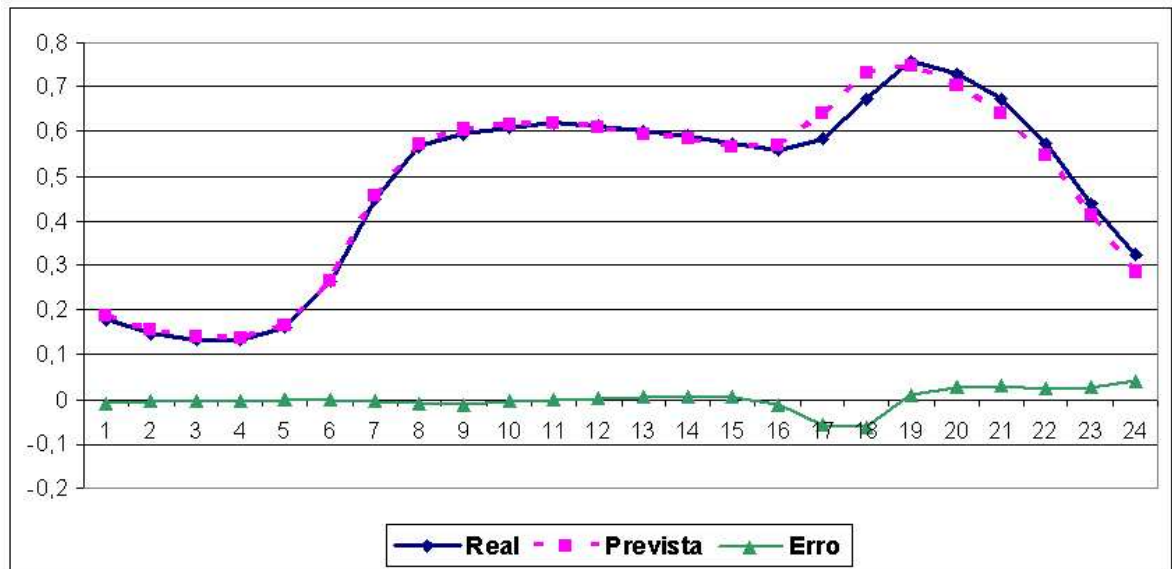


Figura 50: Gráfico da Rede 48x22x24 com Mercado Regulado

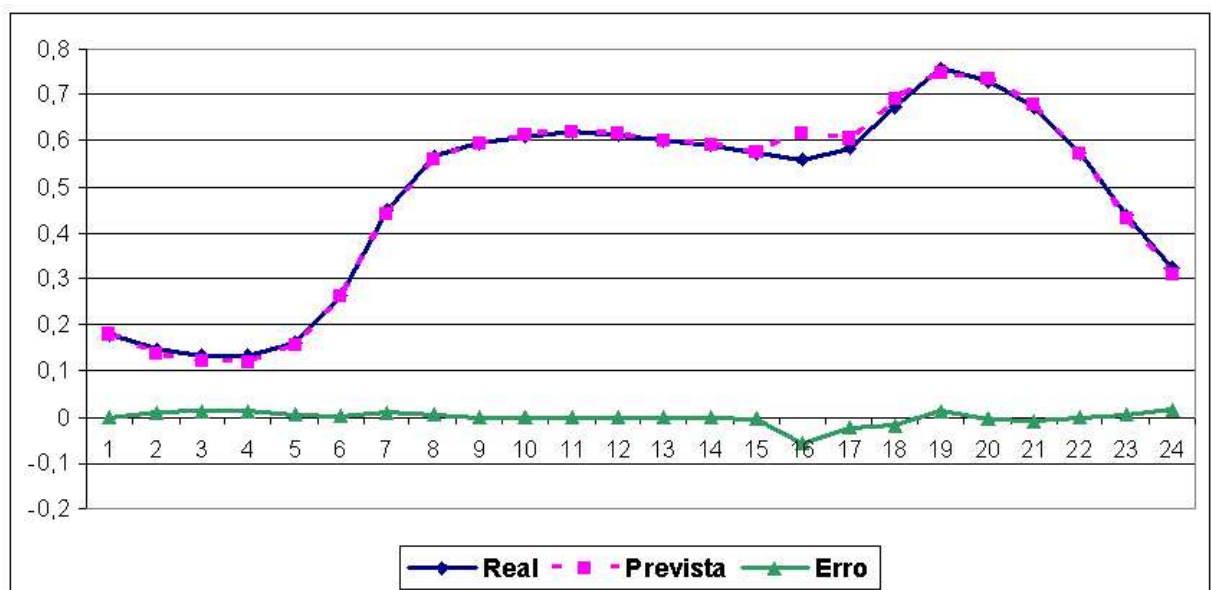


Figura 51: Gráfico da Rede 48x40x24 com Mercado Regulado

### 7.2.3 Comentários

Definir uma topologia ideal para solução de problemas usando MLPs é sempre um estado da arte e quem sabe até de espírito.

No entanto, comparando-se os resultados das tabelas 4 e 5, tem-se que: a rede usando 18 e 22 neurônios na camada oculta com 96 entradas no estudo de caso A não teve uma boa performance, provavelmente por causa do problema clássico de *underfitting*<sup>1</sup> (VEMURI, 1993) existente nas soluções usando MLPs.

No estudo de caso B, a rede com 20 neurônios na camada oculta e 48 entradas teve um aumento significativo no erro. Já a rede com 40 neurônios na camada oculta também teve um acréscimo no seu erro porém não tão significativo como na rede com 20 neurônios na camada oculta.

Percebe-se também, na tabela 5, que o estudo de caso B teve uma performance melhor que o estudo de caso A, pois no estudo de caso B as redes com 18, 22 e 40 neurônios na camada oculta tiveram um erro muito parecido enquanto que na tabela 4 do estado de caso A as rede com 20 e 40 neurônios na camada oculta tiveram um erro parecido. Logo, pode-se concluir que, no geral independente de escolha da rede, o estudo de caso B teve melhor performance em relação ao estudo de caso A.

## 7.3 Estudos de Casos: Mercado Após a Desregulamentação

No mercado de energia após a desregulamentação (2001 até 2003) o preço sofre grandes variações de valores influenciando diretamente na curva de carga. Os testes realizados usaram os pesos otimizados dos estudos de casos anteriores.

---

<sup>1</sup>Ocorre quando a quantidade de neurônios é inadequada para mapeamento dos dados.

Serão apresentado três estudos de caso onde no primeiro e no segundo estudos de caso foram submetidos a um exemplo do mercado após a desregulamentação para se confrontar com os estudos de caso anteriores e no último estudo caso também será usado um exemplo do mercado após a desregulamentação, porém, no processo de previsão foi usado um SHNN para confrontar com todos os estudos de caso anteriores.

### 7.3.1 Estudo de caso C

Esse estudo de caso segue o padrão do estudo de caso A que usa todas as séries origando assim 96 entradas. No entanto, foi submetido a cada topologia de rede um exemplo do mercado após desregulamentação. A tabela 6 mostra a análise dos resultados usando MAPE e as figuras 52, 53, 54, e 55 mostram as previsões de cada topologia de rede descritas na tabela 6.

- **MAPE das Previsões**

Topologia	MAPE(%)
96x18x24	7,1211095
96x20x24	4,4458275
96x22x24	8,261418
96x40x24	3,2656311

Tabela 6: MAPE usando todas as Séries com Preço Volátil

- **Gráficos das Previsões**

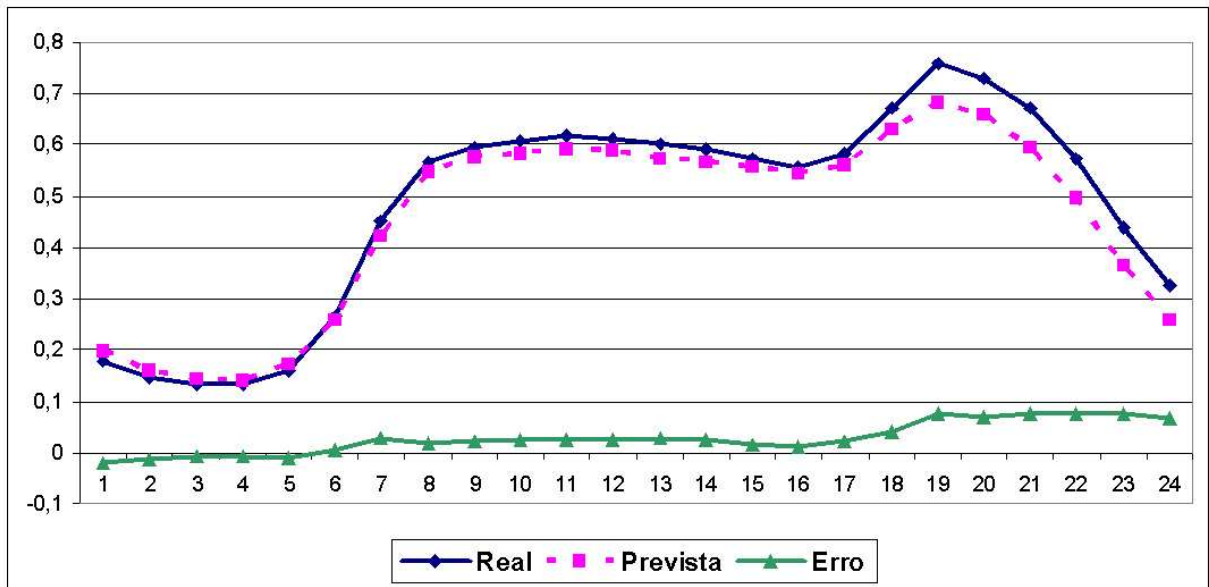


Figura 52: Gráfico da Rede 96x18x24 Após Mercado Desregulado

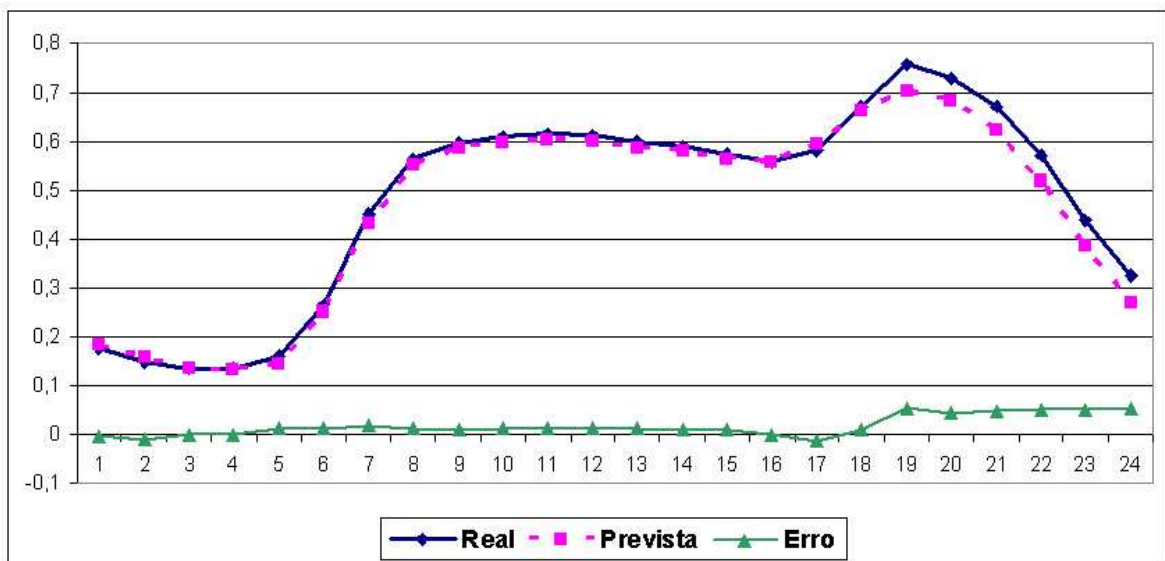


Figura 53: Gráfico da Rede 96x20x24 Após Mercado Desregulado

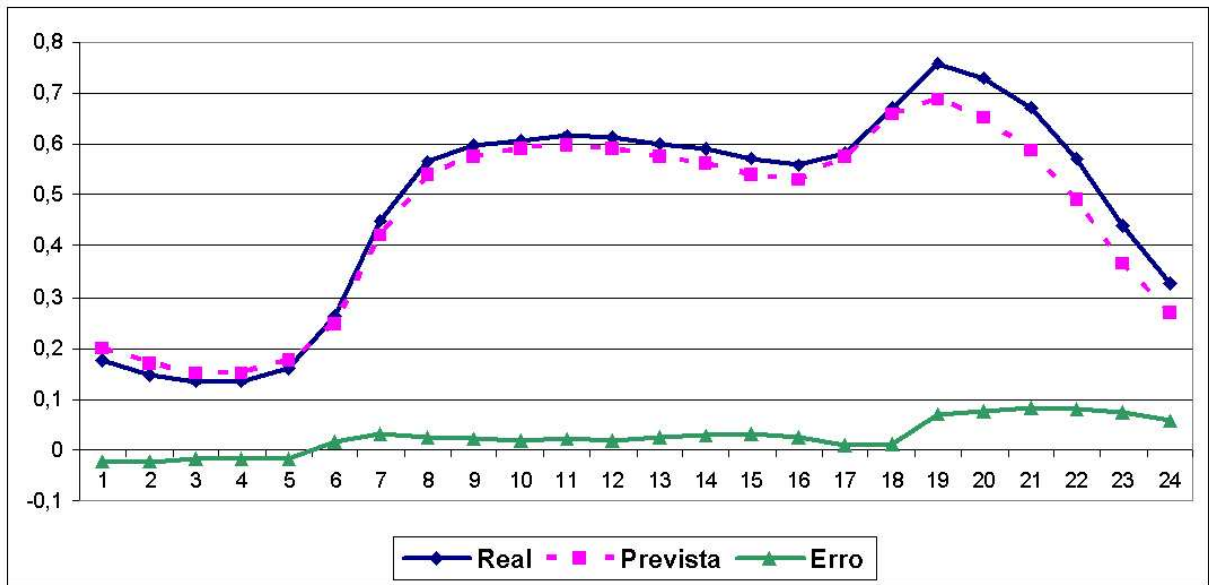


Figura 54: Gráfico da Rede 96x22x24 Após Mercado Desregulado

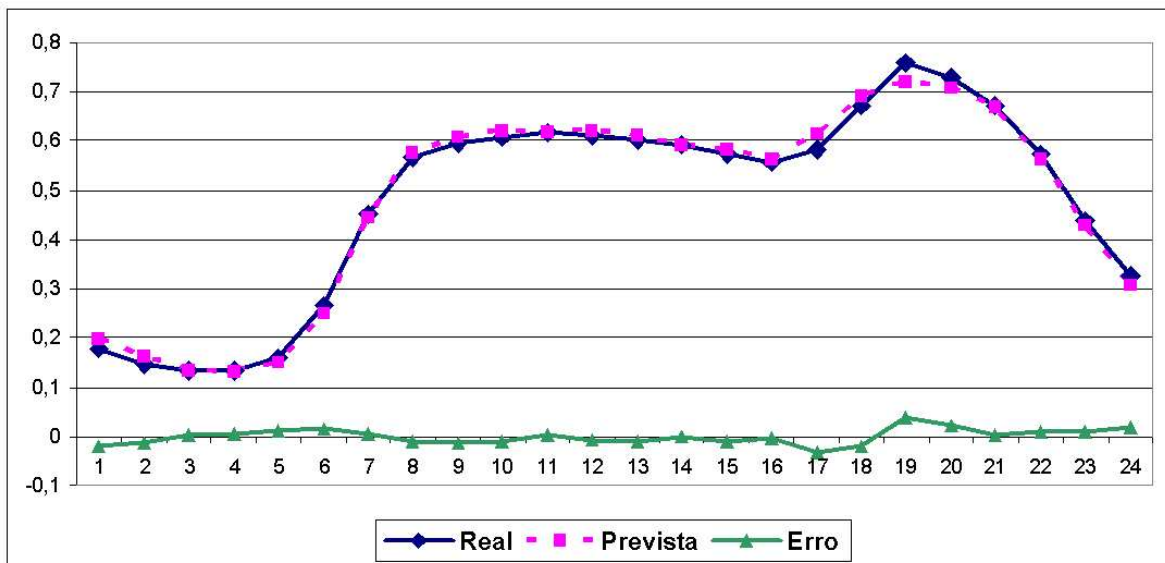


Figura 55: Gráfico da Rede 96x40x24 Após Mercado Desregulado

### 7.3.2 Estudo de caso D

Esse estudo de caso é parecido com o estudo de caso B que usa as séries de carga e preço que resultando em 48 entradas, porém, deve-se lembrar que esses dados são do mercado de energia após a desregulamentação e serão usados para esse teste os pesos otimizados do estudo de caso B. A análise dos resultados fo feita usando MAPE como mostra a tabela 7 e as figuras 56, 57, 58 e 59 mostram a previsão de cada topologia de rede descrita na tabela 7.

- **MAPE das Previsões**

Topologia	MAPE (%)
48x18x24	3,184443
48x20x24	6,6862864
48x22x24	3,8375824
48x40x24	2,967761718

Tabela 7: MAPE usando Séries de Carga e Preço Volátil

- **Gráficos das Previsões**



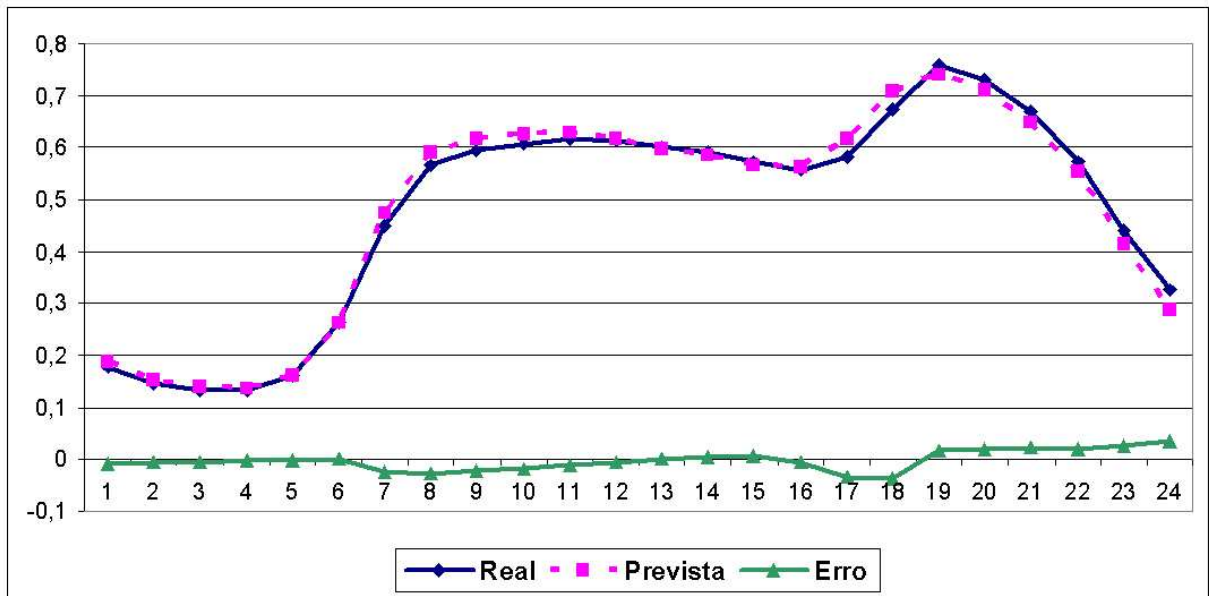


Figura 56: Gráfico da Rede 48x18x24 Após Mercado Desregulado

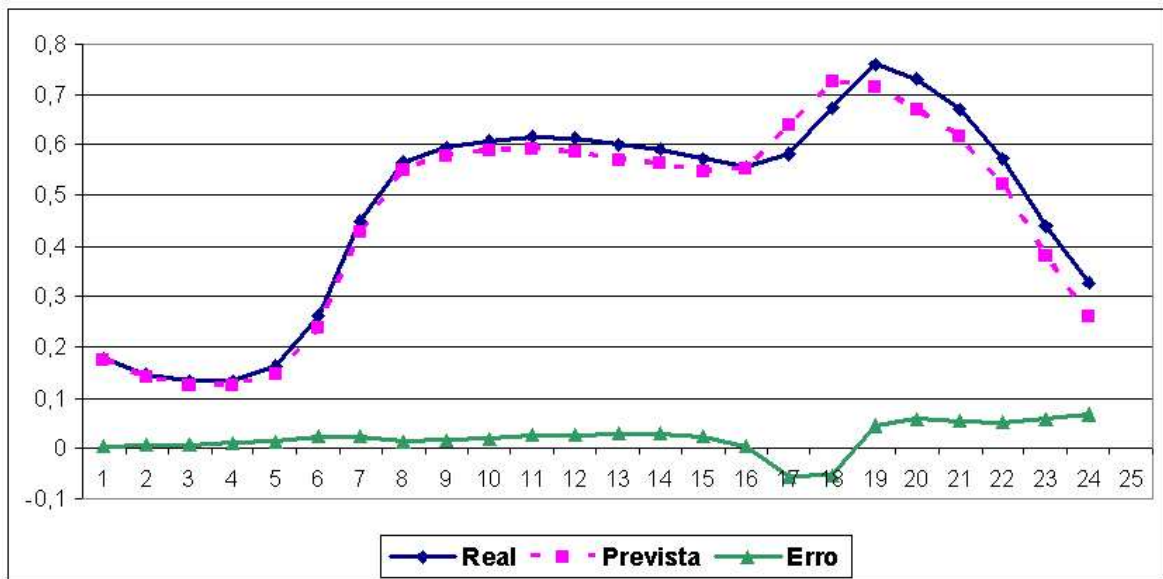


Figura 57: Gráfico da Rede 48x20x24 Após Mercado Desregulado

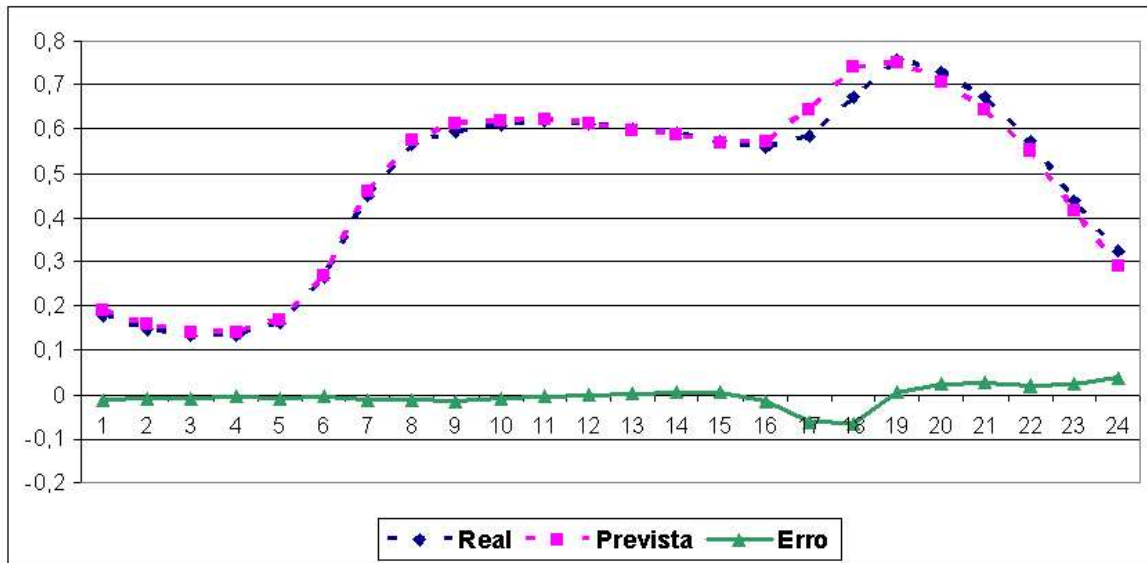


Figura 58: Gráfico da Rede 48x22x24 Após Mercado Desregulado

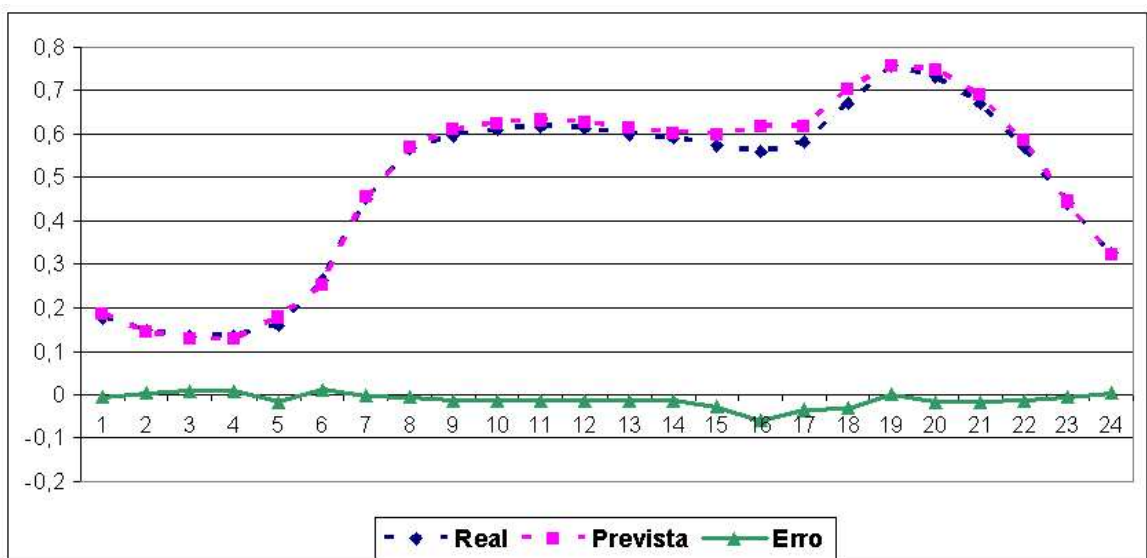


Figura 59: Gráfico da Rede 48x40x24 Após Mercado Desregulado

### 7.3.3 Comentários

Foi observado, nos estudos de casos C e D, que os erros da rede aumentaram significativamente. No primeiro caso, comparando-se a tabela 4 com a tabela 6, observa-se que o erro aumentou em todas as topologias de rede, ocasionando assim uma performance muito baixa. No segundo caso, comparando-se a tabela 5 com a tabela 7, observa-se também que os erros aumentaram em todas as topologias com exceção da rede com 20 neurônios na camada oculta onde o erro diminuiu. Provavelmente nessa topologia aconteceu outro problema clássico das MLPs que se chama *overfitting*<sup>2</sup> (VEMURI, 1993), ou seja, fica pouco sensível a mudança de dados.

Com essa análise, chega-se a conclusão, que em ambientes onde há grandes variações, as soluções usando RNAs no processo de previsão de carga possuem dificuldade em acompanhar essa dinamicidade do ambiente, corroborando assim com (KHOTANZAD; ZHOU; ELRAGAL, 2002) e (SENJYU et al., 2002).

### 7.3.4 Estudo de caso E

Como pôde ser observado nos estudos de casos anteriores, a performance das RNAs fica prejudicada quando o ambiente no qual as redes foram treinadas anteriormente, começa a mudar rapidamente. Na tentativa de solucionar esse problema, foi agregado a MLPs um Sistema de Inferência Nebulosa implementado no previsor TUPAN e gerenciado pelo Assistente XAMAN, para que a rede possa acompanhar a dinamicidade do ambiente mostra Khotanzad, Zhou e Elragal (2002), Padakis (1998), Shao e Sun (1997) .

O Assistente XAMAN possui implementado dois processos de defuzzificação, MOM e COG, que foram utilizados com a intenção de se melhorar a acurácia das previsões. Com relação o estudo de caso escolhido para ser

---

<sup>2</sup>É a situação onde a rede "decora" o conjunto dados.

confrontado foi escolhido foi o estudo de caso D, pois curva de carga é influenciada diretamente pelo preço.

- **MAPE das Previsões usando COG**

Topologia	MAPE(%)
48x18x24	3,201179703
48x20x24	6,686286425
48x22x24	3,75835738
48x40x24	2,164493295

Tabela 8: MAPE do SHNN usando COG e Séries de Carga e Preço Volátil

- **Gráfico das Previsões usando COG**

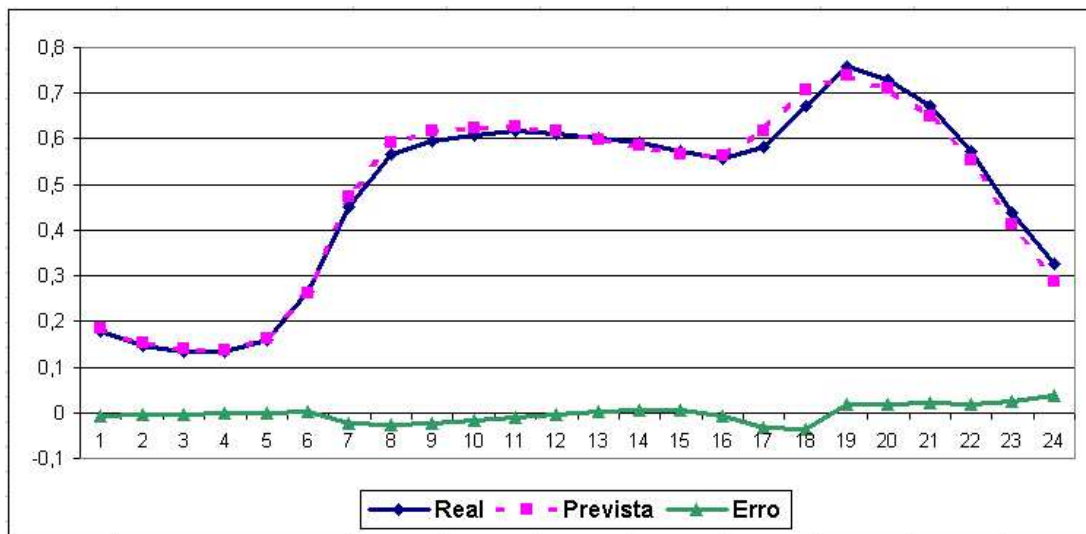


Figura 60: Gráfico da Rede 48x18x24 usando COG

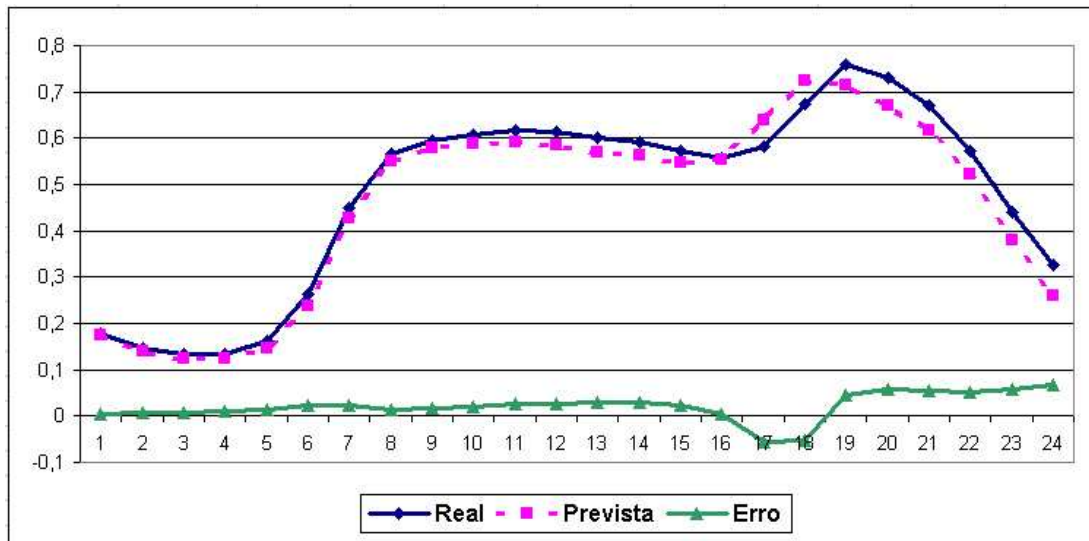


Figura 61: Gráfico da Rede 48x20x24 usando COG

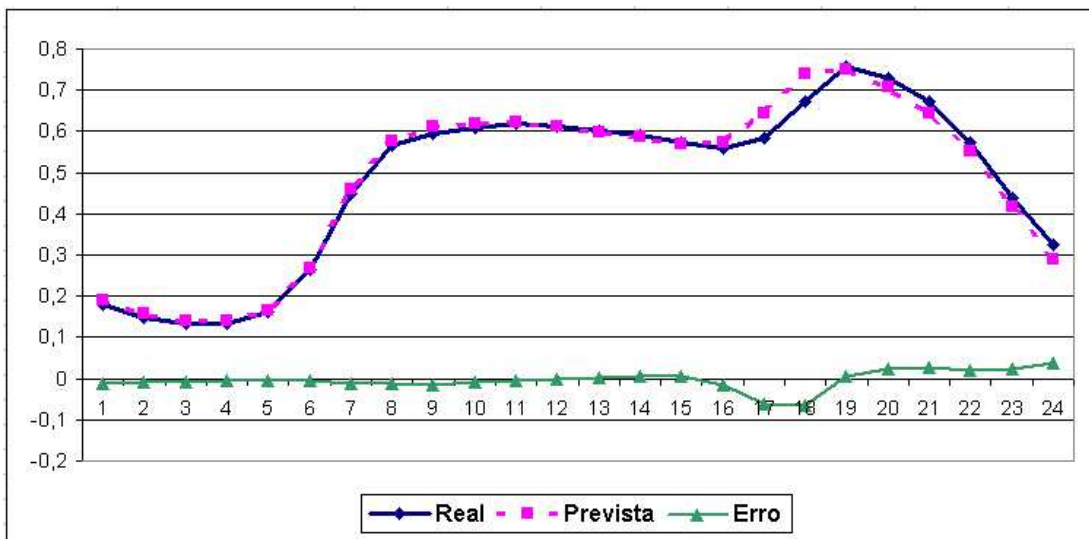


Figura 62: Gráfico da Rede 48x22x24 usando COG

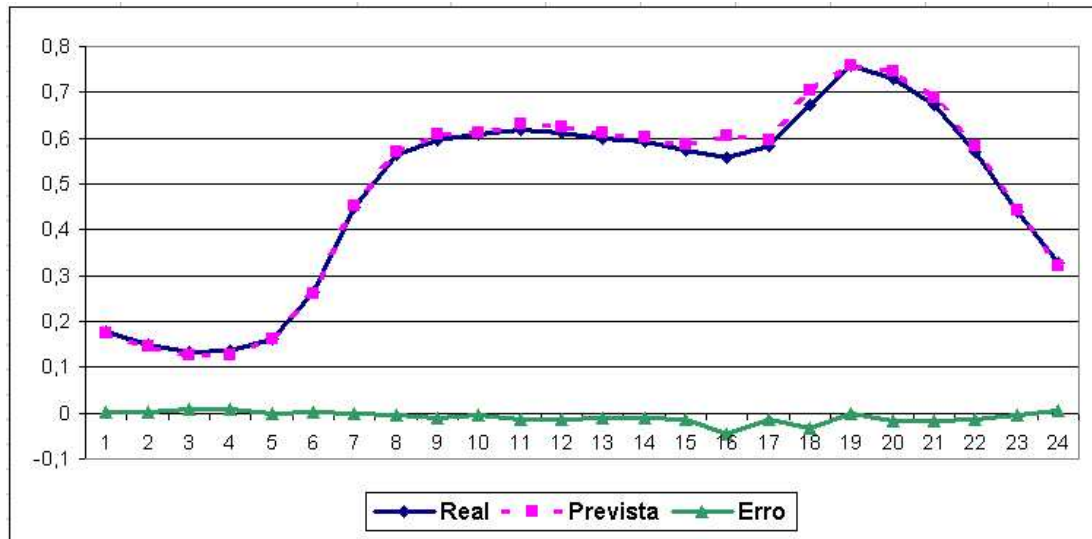


Figura 63: Gráfico da Rede 48x40x24 usando COG

- **MAPE das Previsões usando MOM**

Topologia	MAPE(%)
48x18x24	5,216119943
48x20x24	6,686286425
48x22x24	3,75835738
48x40x24	2,325585363

Tabela 9: MAPE do SHNN usando MOM e Séries de Carga e Preço Volátil

- **Gráfico das Previsões usando MOM**

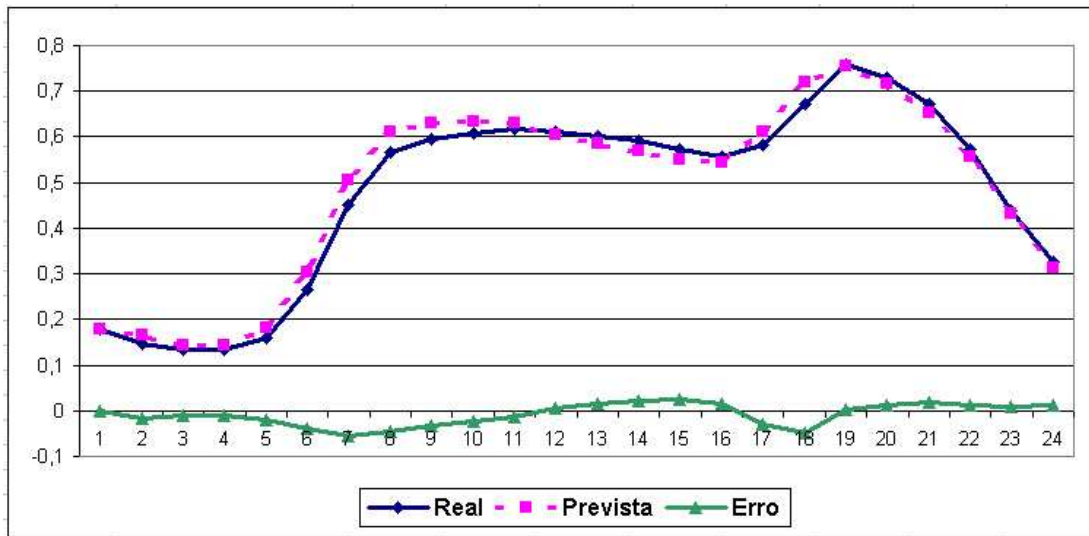


Figura 64: Gráfico da Rede 48x18x24 usando MOM

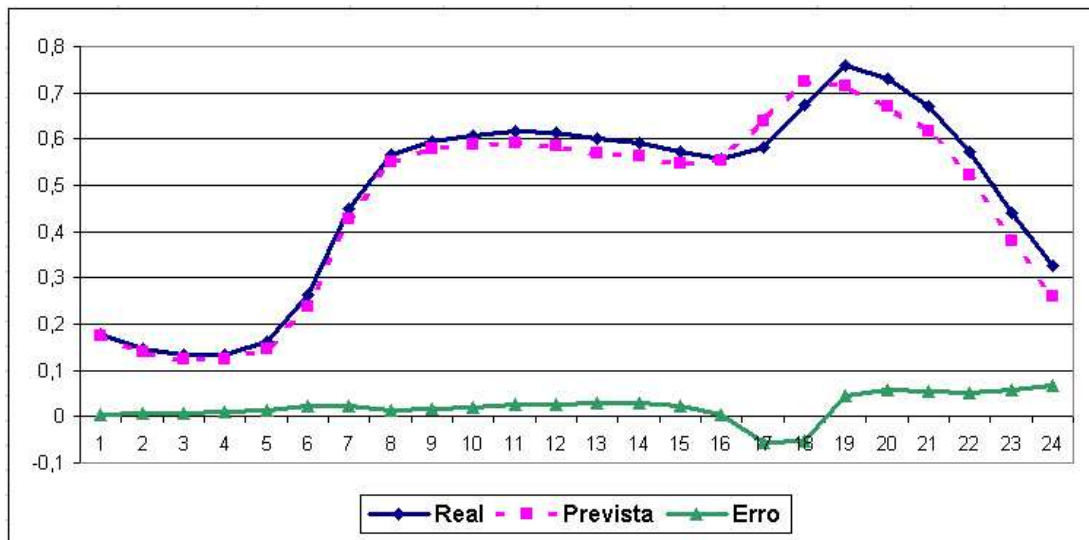


Figura 65: Gráfico da Rede 48x20x24 usando MOM

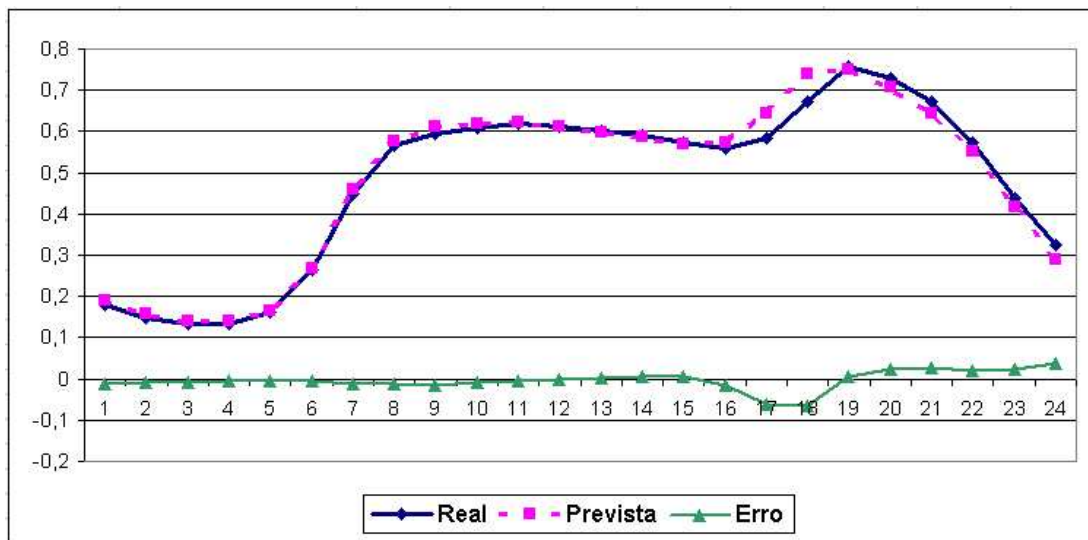


Figura 66: Gráfico da rede 48x22x24 usando MOM

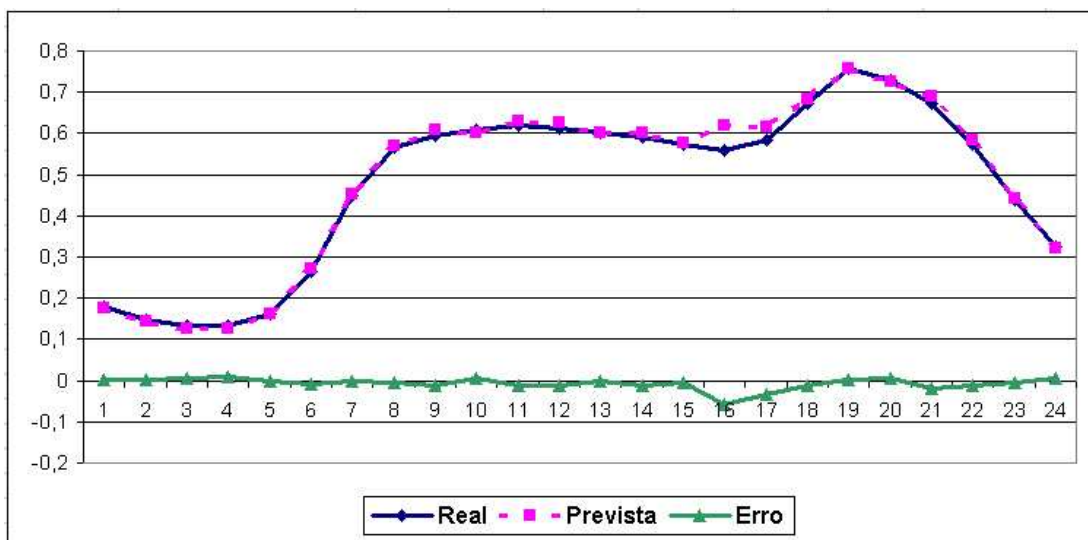


Figura 67: Gráfico da Rede 48x40x24 usando MOM

### 7.3.5 Comentário Final

No processo de desenvolvimento deste trabalho, **não houve a presença do especialista** em mercado de energia elétrica. Com base nas regras de (KHOTAN-



ZAD; ZHOU; ELRAGAL, 2002) é que foram criadas as regras do previsor TUPAN como mostra a tabela 12.

Em se comparando o MAPE das tabelas 8 e 9, a rede que melhor teve performance foi a rede com 40 neurônios na camada oculta, sendo que, em especial na tabela 8 contém melhor performance, ou seja, o sistema nebuloso influenciou no processo de previsão. A rede com 22 neurônios também obteve um bom resultado, pois também conseguiu minimizar o erro. Com relação às demais topologias com 18 e 20 neurônios na camada oculta, o sistema nebuloso não obteve um bom resultado, ou seja, o sistema nebuloso não influenciou no processo de previsão, pois os resultados ficaram muito parecidos.

## *Conclusão*

No desenvolvimento do trabalho, as MLPs usadas no processo de previsão continuam sendo uma solução muito atraente para previsão de carga, porém, como foi observado em ambientes muito dinâmicos, como mercado de energia elétrica da Nova Inglaterra, a rede perde um pouco de sua performance. Desde já, há necessidade de suprir essa queda de performance, por isso, agregou-se um Sistema Nebuloso para conseguir acompanhar a dinamicidade do ambiente, em específico, a variação do preço no mercado de energia que influencia diretamente na série de carga (KHOTANZAD; ZHOU; ELRAGAL, 2002).

Como pôde ser observado através dos testes realizados com o previsor TUPAN, a solução de se hibridar MLPs juntamente com um Sistema Nebuloso é de grande viabilidade pois os teste realizados com o SHNN mostrou-se superior aos modelos que usam MLPs com erros próximos ao limite tolerado em alguns mercados regulados (Califórnia: MAPE máximo de 2 % como informa Khotanzad, Zhou e Elragal (2002) ). A união desses dois sistemas resulta em um sistema mais robusto e eficaz para acompanhar as variações do bruscas do mercado de energia elétrica. Em consequência disso, a previsão de carga é realizada com um desempenho e exatidão melhor como mostrado no estudo de caso E, em se comparando com os demais estudos de caso , resultando em um sistema mais confiável e transparente.

## **Considerações Finais**

No processo de previsão de carga usando o previsor TUPAN, foram encontradas algumas dificuldades, como a ausência do histórico da **temperatura**

variável, essa de grande importância para previsão de carga. Outra dificuldade foi com relação a definição dos **dados qualitativos**, em específico, as variáveis lingüísticas que teriam influência na carga da Nova Inglaterra e a definição do universo do discurso de cada uma das mesmas, pois a realidade do mercado de energia elétrica da Nova Inglaterra é diferente do mercado de energia Brasileiro, exigindo assim um grau de abstração maior para a construção das regras nebulosas. A **ausência do especialista** em mercado de energia elétrica contribuiu para que o previsor TUPAN não conseguisse alcançar uma performance melhor, pois conforme (RAHMAN, 1990) o especialista é de extrema importância para escolha de variáveis adequadas para o sistema nebuloso.

## Trabalhos Futuros

No processo de previsão de carga usando o previsor, a utilização de Algoritmos Genéticos (AG) para encontrar os melhores pesos para uma MLP, seria uma alternativa viável para melhoramento da performance de uma MLP.

Existem outros tipos de redes que são pouco exploradas, como solução no processo de previsão de carga, como exemplo as Redes Neuro-Fuzzy (RNF) e as Redes Função de Base Radial (RBF) . No caso das RNF, tem-se as ANFIS e CANFIS que podem ter uma precisão maior que as redes convencionais (MLPs) e os SHNN; e com relação às redes RBF, seu processo de generalização é maior que as MLPs(B.KARAYIANNIS; BALASUBRAMANIAN; MALKI, 2003), logo seria mais uma solução alternativa para o processo de previsão de carga.

## *APÊNDICE A – Algoritmos e Métodos de Aprendizado*

Neste Capítulo são descritos os paradigmas e algoritmos de aprendizado mais usados no processo de treinamento e otimização de uma RNA.

### **A.1 Aprendizado Supervisionado**

Neste tipo de aprendizado, faz-se necessária a presença de um supervisor ou professor. Este deverá conhecer as respostas a serem fornecidas ao sistema, já que o mesmo conhece o ambiente em que a rede neural irá atuar.

O aprendizado supervisionado vai existir quando for utilizado um agente externo que indique a resposta desejada para o padrão de entrada da rede. A figura 68 ilustra o aprendizado supervisionado.

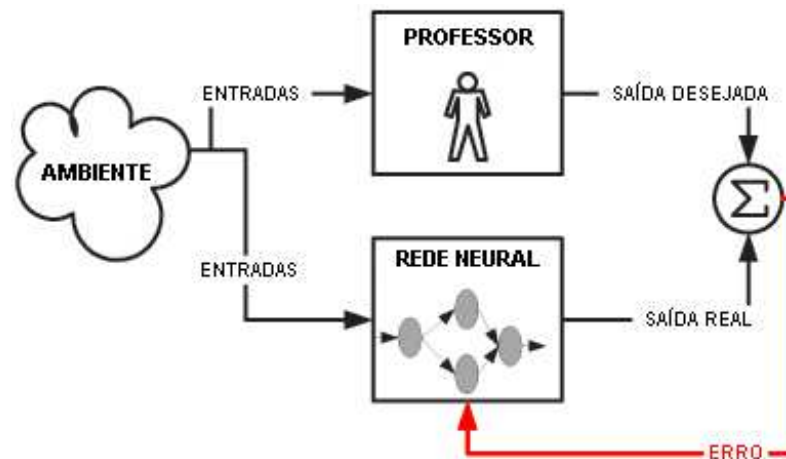


Figura 68: Esquema da Aprendizado Supervisionado

Através da diferença entre o valor desejado (que é o valor que o professor informa) e o valor real (que é o valor que a RNA produz) é que o sistema será capaz de saber qual erro A.1 está sendo produzido e assim acionar o processo de aprendizado da RNA. Assim, a RNA aprende encontrando um conjunto de pesos para evitar que a saída da RNA seja muito distante da saída desejada e minimizar o erro gradativamente no futuro. Dentro do aprendizado supervisionado temos inúmeros algoritmo de aprendizado , no entanto para elucidar melhor citaremos o mais comum:

### A.1.1 Regra Delta Generalizada

É um dos algoritmos de aprendizagem mais famoso, devido à sua simplicidade de implementação ,pois é uma modificação do ADALINE de Widrow-Hoffe em que se considera uma função de ativação não-linear e as respostas assumem valores contínuos. Este algoritmo foi desenvolvido por Rummerlhart, Hinton e Willians, e é extremamente eficiente para a minimização do erro quadrático.

Desde sua redescoberta, o algoritmo de retropropagação tem sido ampla-

mente usado como um algoritmo de aprendizado para RNA com topologia direta com múltiplas camadas. Sua modelagem matemática consiste em:

$$erro_i(t) = d_i(t) - y_i(t) \quad (A.1)$$

$$\Delta w_{i,j} = -\mu \delta_j x_{i,j} \quad (A.2)$$

$$\Delta = \begin{cases} erro_i f'_i(net) & \text{neurônio(s) da camada de saída} \\ f'_i(net) \sum w_{i,j} \delta_j & \text{neurônio(s) da camada oculta} \end{cases} \quad (A.3)$$

$$f'_{i,j}(net) = \begin{cases} y_j(1 - y_j) & \text{função sigmoide} \\ 1 - y^2 & \text{função tangente hiperbólica} \\ 1 & \text{função linear} \end{cases} \quad (A.4)$$

$$Custo = 0.5 \sum_{i=1}^N (erro_i(t))^2 \quad (A.5)$$

$$w_{i,j}(t + 1) = w_{i,j}(t) + \Delta w_{i,j} \quad (A.6)$$

onde  $N$  representa a quantidade de neurônios na camada de saída,  $t$  representa o instante de tempo ou ciclo;  $i$  corresponde ao índice do respectivo neurônio e  $j$  representa o índice do neurônio conseqüente;  $d_{(t)}$  saída desejada da RNA;  $Custo$  representa a função de custo ou energia que deseja-se minimizar;  $y_{(t)}$  saída real da RNA;  $e_{(t)}$  consiste no erro da rede que é obtido pela diferença da saída desejada pela real;  $\eta$  é uma constante que representa a **taxa de aprendizado** que significa com qual velocidade a rede neural irá aprender,  $W_{(t)}$  são os pesos;  $x_{(t)}$  são as entradas da rede e finalizando  $w_{(t+1)}$  consiste nos novos pesos que a rede possui para uma próxima propagação.

### A.1.2 Momentum

Esse algoritmo é baseado no algoritmo anterior e consiste no acréscimo de um **termo momentum** que é a diferença dos pesos multiplicados por uma constante (geralmente menor que a taxa de aprendizagem). Sua modelagem matemática consiste no seguinte:

$$\Gamma = \alpha(w_{x,i}(t-1) - w_{x,i}(t)) \quad (\text{A.7})$$

onde  $\Gamma$  é o valor do termo momentum,  $w_{x,i}(t-1)$  são os pesos de um exemplo anterior, ou seja, de um exemplo no intervalo anterior ( $t-1$ ) e  $w_{x,i}(t)$  são os pesos atuais. A modelagem final fica:

$$w_{i,j}(t+1) = w_{i,j}(t) + \Delta w_{i,j} + \Gamma \quad (\text{A.8})$$

### A.1.3 Weight Decay

Foi introduzido por Paul J. Werbos em 1988 (WERBOS, 1988) e seu funcionamento consiste em decrementar uma conexão sináptica (pesos) durante o processo de treinamento. Em cada atualização dos pesos eles são decrementados por uma constante  $K$ , resultando na seguinte modelagem matemática:

$$\Delta w_{i,j}(t+1) = \mu \delta_j y_j - K w_{i,j} \quad (\text{A.9})$$

## A.2 Aprendizado por Reforço

O modelo de aprendizado supervisionado possui uma extensão que recebeu o nome de **aprendizado por reforço**. O aprendizado por reforço consiste em um mapeamento da entrada-saída realizado através da interação contínua com o ambiente. Tal mapeamento se faz através dos acertos ou erros da rede, ou seja,

para esse método de aprendizado há necessidade de se informar se a rede está **certa** ou **errada**. Existe, assim, a necessidade de um **crítico** qual é diferenciado do professor, mencionado anteriormente, pois o crítico irá informar a rede neural somente se ela está certa, ou seja, se a saída real da rede é satisfatória ou até mesmo igual à saída desejada. Caso contrário, a rede neural irá automaticamente para o processo de atualização dos pesos como mostra a figura 69.

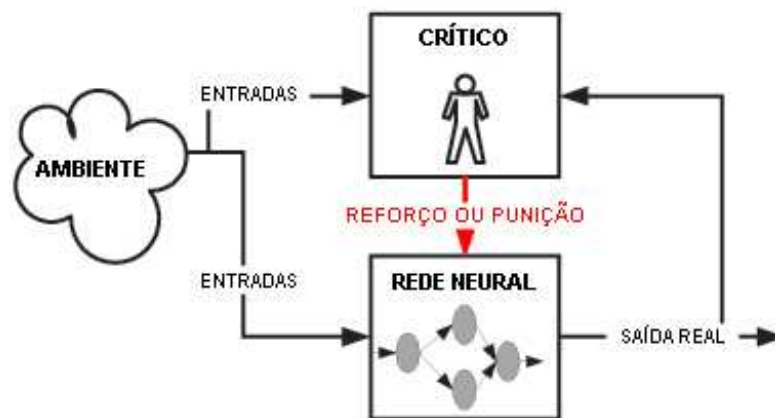


Figura 69: Esquema do Aprendizado por Reforço

### A.2.1 Q-Learning

É um algoritmo também algoritmo muito popular, geralmente é usado no aprendizado de Agentes Inteligentes<sup>1</sup>, e foi proposto por Watkins, em 1989, para auxiliar na escolha da **melhor ação** possível a ser tomada no aprendizado por reforço. Sua modelagem matemática é:

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha[r + \gamma \max_{a'}(Q(s', a'))] \quad (\text{A.10})$$

onde  $s$  representa o estado atual e  $a$  representa a ação do agente dentro de um universo de ações  $A$ ,  $\alpha$  é taxa de aprendizado,  $r$  é uma valor de premiação

<sup>1</sup>São programas desenvolvidos com certa autonomia para realizar alguma tarefa. Mais informações a respeito veja (RUSSEL; NORVIG, 1995)



estando no estado  $s$  e escolhendo a ação  $a$ ,  $s'$  representa o próximo estado e  $a', \max(Q(s', a'))$  significa que: após executar a ação  $a$ , o agente sai do estado  $S$  e vai para um estado  $s'$ , recebendo por esta ação uma recompensa imediata  $r$ . No estado  $s'$  é feita uma busca, entre as ações disponíveis, para encontrar a ação  $a'$  que tenha o maior valor de retorno esperado, ou seja, é uma busca tentando encontrar uma melhor ação (FARIA; ROMERO, 1999).

### A.3 Aprendizado Não-Supervisionado

Esse método de aprendizado limita-se a fazer uma representação da distribuição de probabilidade dos padrões de entrada na rede. Este tipo de treinamento está intimamente ligado à conexão competitiva. Além disso, este tipo de aprendizado ocorre quando, para fazer modificações no valores das conexões sinápticas, não se diz se a resposta foi correta.

Esse aprendizado não requer saídas desejadas como mostra a figura 70, e por isso é conhecido pelo fato de não precisar usar professores para o seu treinamento. A rede trabalha os valores de entrada e vai organizando-se para classificá-los, utilizando critérios próprios. Este tipo de rede utiliza os dados de entrada para classificação e os neurônios como classificadores. O processo de classificação fica a cargo da rede neural e do algoritmo de aprendizado. Assim como no aprendizado supervisionado, o aprendizado não-supervisionado também possui muitos algoritmos de otimização, sendo os mais comuns:

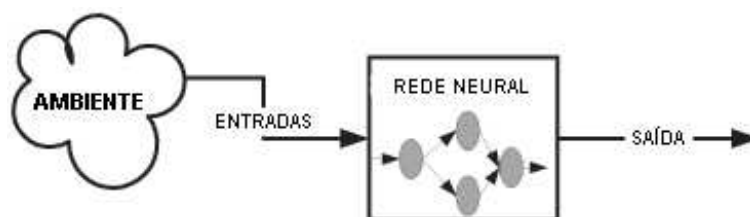


Figura 70: Esquema de Aprendizado Não-Supervisionado

### A.3.1 Otimização Hebbiana

É o mais famoso dos aprendizados não-supervisionados, proposta pelo neuropsicólogo Hebb (1949) em seu livro *The Organization of Behavior*. A regra de Hebb consiste no seguinte: se dois neurônios, em lados distintos da sinapse, são ativados sincronamente, então resultará no fortalecimento da sinapse; no entanto, se forem ativados assincronamente resultará no enfraquecimento da sinapse ou até mesmo sua eliminação.

A modelagem matemática de Hebb consiste em:

$$w_i(t + 1) = \eta y_i(t) x_i(t) \quad (\text{A.11})$$

### A.3.2 Regra de Oja

A regra de Oja é uma modificação da regra de Hebb, e tem por objetivo limitar o aumento excessivo dos vetores de pesos ( $w_i$ ) evitando assim a sua saturação. A regra de Oja tem a seguinte modelagem matemática:

$$w_i(t + 1) = \eta y(x_i(t) - y w_i(t)) \quad (\text{A.12})$$

## *ANEXO A – Simulações com o Previsor*

### **A.1 Treino usando todas as séries**

Topologia	Algoritmo	LMS	Tempo
96x18x24	Regra Delta	0,0011917	00:00:26
	Mommentum	0,000639893	00:00:05
96x20x24	Regra Delta	0,001071785	00:03:01
	Mommentum	0,00083816	00:00:23
96x22x24	Regra Delta	0,0278993	00:05:04
	Mommentum	0,00089457	00:00:20
96x40x24	Regra Delta	0,0375784	00:07:52
	Mommentum	0,00039336	00:06:52

Tabela 10: Treino das Redes usando todas as Séries

## A.2 Treino usando carga e preço

Topologia	Algoritmo	Tempo	LMS
48x18x24	Regra Delta	00:00:02	0,001281
	Mommentun	00:00:23	0,000752
48x20x24	Regra Delta	00:00:24	0,001297
	Mommentun	00:00:26	0,000506
48x22x24	Regra Delta	00:01:11	0,001245
	Mommentun	00:00:25	0,000706
48x40x24	Regra Delta	00:02:43	4,097797
	Mommentun	00:04:22	0,000793

Tabela 11: Treino usando Carga e Preço Volátil

## A.3 Regras Nebulosas

<b>Regras</b>	<b>Antecedente(s)</b>	<b>Consequente(s)</b>
<b>Regra 1</b>	Variacao Preco Muito Baixo;Saida Rede Baixo	Variacao Carga Muito Alta
<b>Regra 2</b>	Variacao Preco Baixo;Saida Rede Baixo	Variacao Carga Alta
<b>Regra 3</b>	Variacao Preco Normal;Saida Rede Baixo	Variacao Carga Normal
<b>Regra 4</b>	Variacao Preco Alto;Saida Rede Baixo	Variacao Carga Normal
<b>Regra 5</b>	Variacao Preco Normal;Saida Rede Normal	Variacao Carga Normal
<b>Regra 6</b>	Variacao Preco Alto;Saida Rede Normal	Variacao Carga Normal
<b>Regra 7</b>	Variacao Preco Muito Baixo;Saida Rede Alto	Variacao Carga Muito Alta
<b>Regra 8</b>	Variacao Preco Muito Alto;Saida Rede Alto	Variacao Carga Normal

Tabela 12: Regras usadas no Processo de Previsão de Carga

## Referências

- AZEEN, M. F.; M, H.; AHMAD, N. Generalization of adaptative neuro-fuzzy inference systems. *IEEE Transactions on Neural Networks*, v. 11, n. 6, p. 1332–1346, November 2000.
- AZEVEDO, F. M. de; BRASIL, L.; OLIVEIRA, R. C. L. de. *Redes Neurais com Aplicações em Controle e em Sistemas Especialistas*. 1. ed. [S.l.]: VisualBooks, 2000.
- BARRETO, J. M. *Inteligência Artificial no Limiar do Século XXI*. 1. ed. [S.l.: s.n.], 1999.
- B.KARAYIANNIS, N.; BALASUBRAMANIAN, M.; MALKI, H. A. Evaluation of cosine radial basis function neural network on electric power load forecasting. *IEEE*, v. 3, p. 2100–2105, Julho 2003.
- BRAGA, A. P.; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicações*. 1. ed. [S.l.]: Rio de Janeiro: Livros Técnicos e científicos(LTC), 2000.
- COX, E. *The Fuzzy Systems Handbook*. 1. ed. [S.l.: s.n.], 1994.
- DINIZ, H. et al. Architecture design of artificial neural networks based on box&jenkins models for time series prediction. *USP*, 1998.
- EL-SHARKAWI, M.; NIEBUR, D. (Ed.). *A Tutorial Course on Artificial Neural Networks with Application to Power Systems*. [S.l.]: IEEE Power Engineering Society, 1996.
- E.P.BOX, G.; GWILYMAND, M.; GREGORY, C. R. *Time Series Analysis Forecasting and Control*. [S.l.]: Prentice Hall, 1994.
- FARIA, G.; ROMERO, R. F. Explorando o potencial de algoritmos de aprendizado com reforço em robôs móveis. *Proceedings of the IV Brazilian Conference on Neural Networks*, p. 237–242, Julho 1999.
- GROSS, G.; GALIANA, F. D. Short-term load forecasting. *Proceedings of the IEEE*, v. 75, n. 12, Dezembro 1987.

HIPPERT, H. S.; PEDREIRA, C. E.; SOUZA, R. C. Neural networks for short-term load forecasting: A review and evolution. *IEEE TRANSACTION ON POWER SYSTEM*, v. 16, n. 1, Fevereiro 2001.

HSU, Y.-Y.; YANG, C.-C. Design of artificial neural networks for short-term load forecasting. part ii: Multilayer feedforward networks for peak load and valley load forecasting. *IEEE Proceedings*, v. 138, n. 5, p. 414–418, Setembro 1991.

KASABOV, N. K. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. 1. ed. [S.l.: s.n.], 1998.

KHOTANZAD, A.; ZHOU, E.; ELRAGAL, H. Annstlf- a neural-network-based eletric load forecasting system. *IEEE Transactions on Power Systems*, v. 8, n. 4, p. 835–846, Julho 1997.

KHOTANZAD, A.; ZHOU, E.; ELRAGAL, H. Annstlf- a neural-network-based eletric load forecaster-generation three. *IEEE Transactions on Power Systems*, v. 13, n. 4, p. 1413–1422, Novembro 1998.

KHOTANZAD, A.; ZHOU, E.; ELRAGAL, H. A neuro-fuzzy approach to short-term load forecasting in a price-sensitive environment. *IEEE Transactions on Power Systems*, v. 17, n. 4, p. 1273–1282, Novembro 2002.

KOSKO, B. *Neural Networks and Fuzzy Systems: a dynamical approach to machine intelligence*. 1. ed. [S.l.]: N.J., EUA, 1992.

KOSKO, B. *Fuzzy Engineering*. 1. ed. [S.l.]: Prentice-Hall, 1997.

LIEW, A.; SRINIVASAN, D. Fuzzy logic approcah to load modeling an forecasting. In: *Power System*. [S.l.: s.n.], 1998.

LIMA, W. da S. *Previsão de Carga no Período de Demanda de Ponta utilizando Redes Neurais Artificiais*. Dissertação (Mestrado) — Unicamp, 95.

LING, S. et al. Short-term eletric load forecasting based on a neuro fuzzy network. *IEEE Transaction on Industrial electronics*, v. 50, n. 6, p. 1305–1316, Dezembro 2003.

MANN, F.; FGARCIA-LAGOS; SANDOVAL, F. Global model for short-term load forecasting unsing artificial neural networks. *IEE Proc-Gener.Transm.*, v. 149, n. 2, Março 2002.

MEDEIROS, A. G. *Modelos Neuro-Difusos: Um Enfoque para Integração de Redes Neurais e Sistemas Difusos*. Dissertação (Mestrado) — Universidade Federal de Pernambuco(UFPE), 1996. Dissertação de Mestrado em Ciência da Computação.

- NAFIS, A. Fuzzy logic and its application. *IEEE*, 2001.
- NETO, L. S. R. *Redes Neurais Artificiais, Lógica Nebulosa e Sistemas Neuro-Fuzzy na Previsão de Carga em Curto Prazo*. Dissertação (Mestrado) — PUC-RJ, Agosto 1999.
- PADAKIS, S. E. A novel approach to short-term load forecasting using fuzzy neural networks. *IEEE Transaction on Power System*, v. 13, n. 2, Maio 1998.
- PAN, Z.; LIU, X.; MEJABI, O. A neural-fuzzy system for forecasting. *IEEE*, 1997.
- PAULO, A. R. G. da S.  
*Previsão de Carga de Curtíssimo Prazo Utilizando Redes Neurais Artificiais* — Escola de Engenharia Elétrica- UFG, Fevereiro 2004. Trabalho de Graduação em Engenharia Elétrica.
- RAHMAN, S. Formulation and analysis of rule-based short-term load forecasting algorithm. *Proceedings of the IEEE*, v. 78, n. 5, p. 805–816, Maio 1990.
- REZENDE, S. O. *Sistemas Inteligentes Fundamentos e Aplicações*. Barueri-São Paulo: Editora Manole Ltda, 2005.
- RICH, E.; KNIGHT, K. *Inteligência Artificial*. 2. ed. [S.l.]: McGraw-Hill, 1993.
- RUSSEL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 1. ed. [S.l.: s.n.], 1995.
- SANDRI, S.; CORREA, C. Lógica nebulosa.
- SENJYU, T. et al. One-hour-ahead load forecasting using neural network. *IEEE Transactions on Power Systems*, v. 17, n. 1, p. 113–118, Fevereiro 2002.
- SHAO, S.; SUN, Y. Short-term load forecasting using fuzzy neural network. In: *APSCOM-97*. Hong Kong: [s.n.], 1997. (Operation and Management), p. 131–134.
- SIMON, H. A. *As ciências do artificial*. 1. ed. [S.l.: s.n.], 1981.
- SWARUP, S.; SATISH, B. Integrated ann approach to forecast load. *IEEE Computer Application in Power*, 2002.
- THAMMANO, A. Forecasting time series using logical combinations neural-based networks. *IEEE*, p. 3573, 2000.
- VEMURI, R. Main problems and issues in neural networks application. *IEEE*, p. 226, 1993.



WERBOS, P. J. Backpropagation: past and future. In: *In Proceedings of the IEEE International Conference on Neural Networks*. [S.l.]: IEEE, 1988. p. 343–353.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)