

# DESENVOLVIMENTO DE UM SISTEMA COMPUTACIONAL DE PREVISÃO DE CARGA BASEADO EM REDES NEURAI E INTELIGÊNCIA ARTIFICIAL ATRAVÉS DE SENSORES IMPLANTADOS NO SISTEMA DE TRANSMISSÃO DE ENERGIA

Prof. Dr. J. A. Jardini, L. C. Magrini, C. R. Guirelli, USP  
M. Bastos, A. C. Campos, CTEEP

**Resumo** - Este trabalho tem por objetivo o desenvolvimento de uma ferramenta computacional para a previsão da carga de curto período em tempo real para o sistema de transmissão da CTEEP utilizando técnicas de inteligência artificial. Os resultados apresentados aqui correspondem ao primeiro ciclo do projeto.

**Palavras-chave**— lógica fuzzy, previsão de carga, redes neurais, sistemas de transmissão, wavelets.

## I. INTRODUÇÃO

A previsão de carga de curto prazo tem grande importância para a operação e planejamento de um sistema de transmissão de energia elétrica. Melhorando a exatidão da previsão podemos aumentar a segurança do sistema e paralelamente diminuir seus custos de geração.

No entanto, o problema da previsão é complexo não somente devido as não linearidades do sistema, comportamentos aleatórios da carga e fatores externos tais como a temperatura ambiente, estação do ano, luminosidade, feriados, etc.

Atualmente a CTEEP realiza a previsão da carga do sistema de transmissão de uma maneira manual baseada na experiência adquirida pelos operadores ao longo dos anos.

O objetivo deste projeto é desenvolver um sistema automático de previsão de carga utilizando técnicas de inteligência artificial (IA). O sistema deverá ter capacidade de prever a carga nas próximas 24 horas (previsão de base) e também refinar a solução ao longo do dia.

Para o desenvolvimento do sistema de previsão, a pesquisa bibliográfica [3], [6], [8], [19] indicou como as melhores opções as redes neurais artificiais (RNAs) e a Lógica Fuzzy. Ambas as técnicas se valem da capacidade de aprender o comportamento de um sistema através de um histórico, não sendo necessário um modelamento complexo do sistema.

Também foi estudado o uso da decomposição Wavelet para a filtragem dos dados e seu uso em combinação com as

técnicas de IA.

A CTEEP forneceu dados de carregamento do sistema, medidos minuto a minuto, para o período de julho de 2001 a abril de 2003 e o IAG forneceu as medições de temperatura ambiente de sua estação meteorológica na cidade de São Paulo para o mesmo período. O objetivo é a verificação do efeito da temperatura ambiente no carregamento.

## II. ANÁLISE DOS DADOS

A massa de dados disponíveis precisa ser adequadamente tratada e analisada para que possa ser utilizada.

Os dados de carga foram filtrados para eliminação de ruídos e medições falhas e calculado um valor médio para cada quarto de hora. Os perfis de carregamento foram analisados para se identificar padrões e sazonalidades que possam auxiliar na previsão e treinamento dos sistemas de IA.

### A. Filtragem

Para a filtragem dos dados foi adaptada uma técnica de identificação de dados distorcidos (*bad data*) usados por estimadores de estados [25].

O método consiste em se calcular o quanto os pontos de um período em estudo estão distantes da curva estimada a carga nesse período. Se os desvios dos pontos estiverem além de um certo limite, provavelmente temos medidas distorcidas

As curvas de carga foram estimadas por retas, encontradas por regressão linear, para períodos de 15 minutos. Para cada ponto foi calculado o resíduo da medição por :

$$J_i = \frac{(x_{i\text{medido}} - x_{i\text{estimado}})^2}{\sigma^2} \quad (1)$$

onde  $\sigma$  é o desvio padrão do conjunto de medição.

Os erros de medição são números aleatórios de modo que  $J$  também o é, e pode ser demonstrado que ele tem uma fun-

ção densidade de probabilidade que corresponde a uma distribuição chi-quadrada  $\chi^2$ .

Dada uma precisão desejada, ou seja, a probabilidade de termos dados distorcidos, podemos encontrar o limite de  $J$  em uma tabela da distribuição  $\chi^2$ .

Se o  $J$  total do sistema, estiver acima do limite encontrado na distribuição, devemos ter uma medição errada. A melhor candidata é a medição com o maior  $J_i$ .

A medição errada é substituída pelo valor estimado pela reta e o processo é repetido até termos  $J$  dentro dos limites.

Na curva abaixo temos um exemplo de curva filtrada pelo processo descrito :

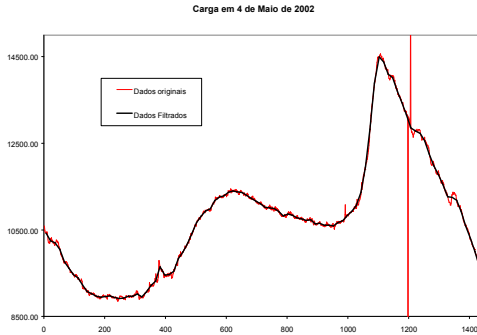


Figura 1 – Exemplo de filtragem de dados para curva de carga

### B. Padrões

A análise das curvas de carga do histórico disponível mostrou algumas informações :

- Segundas, sábados e domingos possuem perfis distintos e terças, quartas, quintas e sextas possuem perfis extremamente semelhantes;
- Feriados e pontes possuem um perfil distinto;
- Com o fim do racionamento em fevereiro de 2002, o perfil de carga vem se alterando e retornando ao que ra antes do racionamento. Desse modos temos um período com carga atípica;
- O horário de verão é um período onde a curva de carga, principalmente no horário de pico, tem um perfil diferenciado do restante do ano;

A partir dessas informações concluí-se que o melhor seria testar as RNAs e Lógica Fuzzy usando sistemas específicos para os dias de semana e desprezar o histórico do período de racionamento por ser atípico. Há ainda a necessidade de se identificar e introduzir no sistema variáveis causais que possam afetar a carga tais como a temperatura e outras que possam representar a sazonalidade do consumo de energia.

Na figura 1 podemos ver a mudança do perfil de carga ao longo do ano de 2002 e na figura 2 vemos o perfil do mês de Maio de 2002 onde se pode ver a semelhança entre os dias de semana.

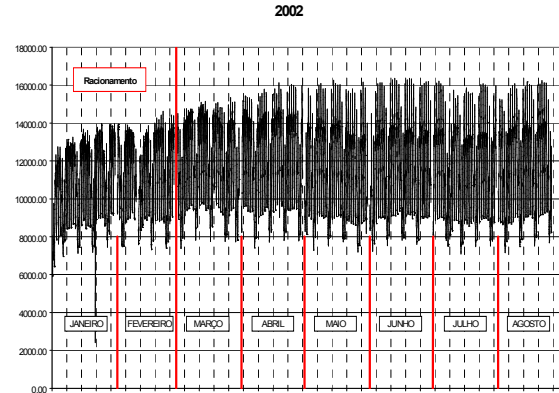


Figura 2 - Curva de carga para o ano de 2002 (MW x tempo)

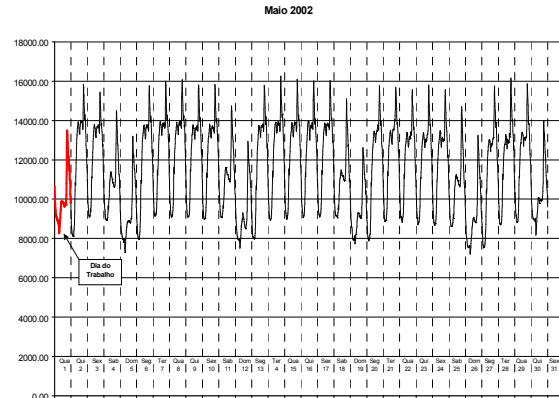


Figura 3 - Curva de carga para Maio de 2002 (MW x tempo)

## III. LÓGICA FUZZY

Um sistema de inferência Fuzzy pode aprender o comportamento de um sistema a partir de um histórico do comportamento do sistema, isso é, dado um conjunto de variáveis de entrada, temos um conjunto de saídas correspondentes a esta situação.

A previsão através de lógica Fuzzy tem a vantagem de possibilitar a inclusão de regras Heurísticas.

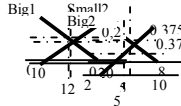
A dificuldade do sistema de inferência é a determinação do conjunto de regras. Ele tem de ser feito experimentalmente já que não há como determinar previamente o sistema ótimo.

### A. Clusterização

A clusterização tem papel fundamental no uso da Lógica Fuzzy. É ela que vai agrupar os conjunto de dados semelhantes e determinar as funções proprietárias do sistema.

A técnica de clusterização adotada consiste em se medir o potencial de cada ponto  $x_i$  de ser o centro de um cluster através da formula :

$$P_i = \sum_{j=1}^n e^{-\alpha \|x_i - x_j\|^2} \quad (2)$$



onde :  $\alpha = \frac{4}{\gamma_a^2}$   
 $\gamma_a$  é uma constante positiva  
 $\|x_i - x_j\|$  distância euclidiana entre  $x_i$  e  $x_j$   
 $n$  é o número de pontos ou conjuntos de entrada

Cada ponto  $x$  representa uma coordenada em  $m$  dimensões onde  $m$  é o número de variáveis de entrada. O ponto com maior potencial ( $x^*$ ) é adotado como o primeiro centro de cluster. Os demais pontos têm o potencial do centro subtraído de seu potencial através de :

$$P_{i\text{ novo}} = P_i - P_i^* e^{-\beta \|x_i - x_i^*\|^2} \quad (3)$$

onde  $\beta = \frac{4}{\gamma_b^2}$   
 $\gamma_b$  é uma constate positiva maior que  $\gamma_a$

Novamente o ponto de maior potencial é adotado como outro centro de cluster e o processo é repetido até que o potencial restante seja :

$$P_k \leq \epsilon P_1^* \quad (4)$$

Os parâmetros  $\gamma_a$ ,  $\gamma_b$  e  $\epsilon$  tem de ser determinados experimentalmente. A escolha de  $\epsilon$  é importante para evitar um número muito pequeno ou muito grande de clusters. Poucos clusters podem tornar o modelo pouco preciso e quando muitos, podem ser redundantes, tornando o modelo desnecessariamente complexo.

**B. Sistema de Inferência Fuzzy**

O sistema de inferência adotado é do tipo Sugeno. O sistema de regras possui a seguinte forma :

$$\begin{cases} R^1 : \text{IF } z_1 \text{ is } A_1^1 \text{ and ...and } z_n \text{ is } A_n^1 \text{ THEN } y^1 = p_0^1 + p_1^1 z_1 + \dots + p_n^1 z_n \\ R^2 : \text{IF } z_1 \text{ is } A_1^2 \text{ and ...and } z_n \text{ is } A_n^2 \text{ THEN } y^2 = p_0^2 + p_1^2 z_1 + \dots + p_n^2 z_n \\ \vdots \\ R^c : \text{IF } z_1 \text{ is } A_1^c \text{ and ...and } z_n \text{ is } A_n^c \text{ THEN } y^c = p_0^c + p_1^c z_1 + \dots + p_n^c z_n \end{cases} \quad (5)$$

$y^c$  são as funções de inferência e  $A_n^c$  as funções de pertinência. Temos  $n$  entradas,  $c$  regras e uma saída  $y$ .

A defuzzificação é feita através de :

$$y = \frac{\sum_{i=1}^c [A_1^i(z_1) \wedge \dots \wedge A_n^i(z_n)] \times (p_0^i + p_1^i z_1 + \dots + p_n^i z_n)}{\sum_{i=1}^c [A_1^i(z_1) \wedge \dots \wedge A_n^i(z_n)]} \quad (6)$$

Para entender melhor o sistema de inferência descrito, vamos tomar como exemplo o seguinte sistema de 3 regras :

- R<sup>1</sup>: If z<sub>1</sub> is small<sub>1</sub> and z<sub>2</sub> is small<sub>2</sub> then y=z<sub>1</sub>+z<sub>2</sub>
- R<sup>2</sup>: If z<sub>1</sub> is big<sub>1</sub> then y=2z<sub>1</sub>
- R<sup>3</sup>: If z<sub>2</sub> is big<sub>2</sub> then y=3z<sub>2</sub>

O sistema de inferência funcionará da seguinte maneira tendo com entradas  $z_1=12$  e  $z_2=5$  :

Regra	Premissa	consequência	True Value
R1		$y=12+5=17$	$5 \wedge 0.375=0.2$
R2		$y=2*12=24$	0.2
R3		$y=3*5=15$	0.375
entrada	$z_1=12$	$z_2=5$	saída $y=17.8$

A saída do sistema será :

$$y = \frac{0.25 \times 17 + 0.2 \times 24 + 0.375 \times 15}{0.25 + 0.2 + 0.375} = 17.8$$

**C. Identificação Automática de Modelos**

A identificação automática de modelos permite que um histórico do comportamento do sistema seja transformado em um conjunto de regras e funções de pertinência e inferência. As funções de pertinência foram adotadas como sendo gaussianas. O número de regras e os parâmetros das funções de pertinência são determinados pela clusterização. Cada cluster corresponde a uma regra e o valor de cada variável de entrada no cluster corresponde ao centro da gaussiana.

$$A_j^i(z_j) = e^{-\frac{(z_j - z_j^*)^2}{2\sigma_j^2}} \quad (7)$$

- onde  $z_j$  j-ésima variável de entrada
- $A_j^i(z_j)$  i-ésima função de pertinência da variável  $z_j$
- $z_{ij}^*$  valor médio da função de pertinência (centro do cluster)
- $\sigma_j$  desvio padrão da função de pertinência da j-ésima variável de entrada. Igual a  $\gamma_j / \sqrt{8}$  vezes o alcance da j-ésima variável de entrada.

A função de consequência é obtida por :

$$y = \frac{\sum_{i=1}^c (A_1^i(z_1) \wedge \dots \wedge A_n^i(z_n)) (p_0^i + p_1^i z_1 + \dots + p_n^i z_n)}{\sum_{i=1}^c [A_1^i(z_1) \wedge \dots \wedge A_n^i(z_n)]} \quad (8)$$

Fazendo :

$$\beta_i = \frac{(A_1^i(z_1) \wedge \dots \wedge A_n^i(z_n))}{\sum_{i=1}^c [A_1^i(z_1) \wedge \dots \wedge A_n^i(z_n)]} \quad (9)$$

Temos :

$$y = \sum_{i=1}^c (p_0^i \beta_i + p_1^i z_1 \beta_i + \dots + p_n^i z_n \beta_i) \quad (10)$$

Com um conjunto redundante de dados, podemos obter os parâmetros  $p$  usando o método dos mínimos quadrados para resolver :

$$\|Z\|P = \|Y\| \quad (11)$$

onde  $Y$  é o vetor da saídas,  $P$  é o vetor dos parâmetros  $p_n^i$  a serem estimados e  $Z$  é uma matriz constante obtida a partir de (10).

#### D. Resultados Obtidos

O modelo Fuzzy adotado trabalha com qualquer número de entradas mas só uma saída o que o torna adequado à previsão de seguimento (previsão do próximo período).

Foram feitas diversas simulações de previsão para a carga da CTEEP, com as seguintes condições :

- para cada dia da semana foi treinado um sistema de inferência Fuzzy específico.
- o conjunto de treinamento vai de 1/2/02 a 30/9/02.
- o conjunto de teste vai de 1/10/02 a 12/2/03.
- foram utilizados dados amostrados a cada 15 minutos já filtrados.

O curto período de tempo com dados disponíveis, bem como outros fatores tais como o racionamento, dificultam a identificação de sazonalidades nas curvas de carga ao longo do ano.

O primeiro conjunto de teste envolveu a previsão da carga 15 minutos à frente (c+15) baseado nos dados das últimas duas horas totalizando 9 entradas : c, c-15, c-30, c-45, c-60, c-75, c-90, c-105, c-120. Isso gerou um sistema com 9 entradas e 1 saída.

Foram mantidos os mesmos parâmetros de clusterização para todos os casos. Os resultados obtidos encontram-se na tabela a seguir :

TABELA I  
PREVISÃO DOS PRÓXIMOS 15 MINUTOS COM OS 9 PERIODOS ANTERIORES

Dia	clusters	Erro % médio	Desvio Padrão
Segunda	5	1.14	1.76
Terça	5	0.92	1.10
Quarta	5	0.98	1.26
Quinta	4	0.89	1.50
Sexta	5	0.91	1.34
Sábado	4	0.79	1.32
Domingo	8	0.86	1.09

O mesmo processo com as mesmas entradas foi repetido, mas tendo como saída a previsão 1 hora à frente (c+60), obtendo-se os erros apresentados na tabela a seguir :

TABELA II  
PREVISÃO DOS PRÓXIMOS 60 MINUTOS COM OS 9 PERIODOS ANTERIORES

Dia	clusters	Erro % médio	Desvio Padrão
Segunda	7	4.45	4.96
Terça	5	3.37	3.10

Quarta	6	3.94	3.57
Quinta	4	3.23	3.82
Sexta	4	3.17	3.24
Sábado	4	2.88	3.32
Domingo	8	3.10	3.52

Verificamos que com o aumento da distância da previsão no tempo, o erro da mesma também aumenta.

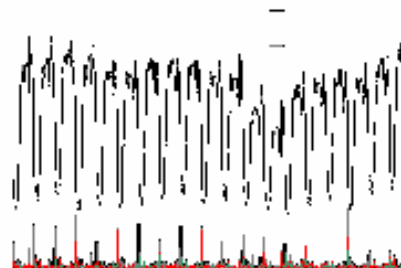


Figura 4 - Previsão de c+15 para segundas

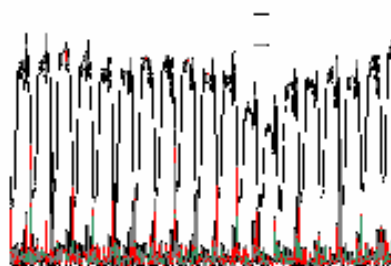


Figura 5 - Previsão de c+60 para segundas

Para verificarmos o efeito da temperatura ambiente na previsão de seguimento, as simulações para a previsão de c+15 minutos forma repetidas mas para cada medição usada como entrada do sistema Fuzzy, foram acrescentadas novas entradas correspondentes à temperatura em cada instante.

Os resultados estão na tabela a seguir :

TABELA III  
PREVISÃO DOS PRÓXIMOS 15 MINUTOS COM OS 9 PERIODOS ANTERIORES UTILIZANDO A TEMPERATURA AMBIENTE

Dia semana	Erro % médio	
	Previsão sem temperatura	Previsão com temperatura
Segunda	0.94	1.14
Terça	1.35	0.92
Quarta	1.16	0.98
Quinta	0.88	0.89
Sexta	0.87	0.91
Sábado	0.66	0.79
Domingo	0.96	0.86

Verificamos que para o caso de previsão de seguimento de curtíssimo período, a temperatura ambiente não tem efeito significativo podendo ser desprezada.

#### IV. REDES NEURAIS

### A. RNAs - O modelo Multi Layer Perceptron

A figura 6 apresenta uma representação do modelo MLP. Unidades básicas (neurônios) são dispostos em camadas. Qualquer neurônio em uma dada camada se conecta com todos os neurônios existentes nas camadas anterior e posterior da camada em questão. A cada conexão está associado um peso, cujo valor é variável durante o treinamento e permite codificar o conhecimento que o MLP tem sobre um determinado problema.

A função da camada 0 (camada de entrada) é simplesmente passar as entradas do MLP a cada neurônio na camada 1 através da matriz de pesos  $W1$ . Esta matriz contém tantas linhas quantas sejam as entradas do MLP, e tantas colunas quantos sejam os neurônios da camada 1. Desta forma, nenhuma operação é realizada na camada 0. Cada neurônio na camada 1 calcula a soma ponderada das entradas (utilizando os pesos da matriz  $W1$ ) e aplica uma função não linear a essa soma. O resultado é então passado a todos os neurônios da camada 2, desta vez através da matriz de pesos  $W2$ . A camada 2 (e todas as demais subseqüentes) efetuam as mesmas operações que a camada 1.

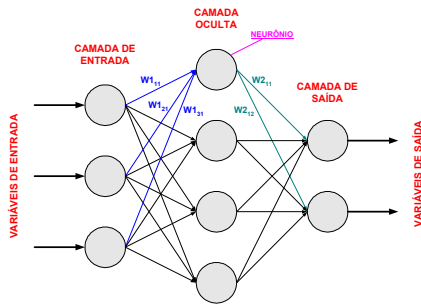


Figura 6 - O modelo MLP

A idéia geral é imitar a estrutura em camadas e a natureza paralela do cérebro humano. As equações (12) e (13) ilustram estas operações, considerando-se o cálculo da saída produzida pelo neurônio  $k$  na camada  $i$ :

$$l_k = \sum_{j=1}^n e_j w_{jk} \quad (12)$$

$$s_k = \frac{1}{1 + e^{-\lambda l_k}} \quad (13)$$

onde:

- $k$  indica cada um dos neurônios da camada  $i$ ;
- $n$  número de entradas que alimentam o neurônio  $k$  (na camada  $i = 1$ , é o número de entradas da rede MLP; para as demais camadas é o número de neurônios na camada anterior);
- $e_j$  valor da entrada  $j$  do neurônio  $k$ ;
- $w_{jk}$  peso que interliga a entrada  $j$  com o neurônio  $k$ ;
- $l_k$  valor líquido das entradas no neurônio  $k$  (ponderadas pelos pesos  $w_{jk}$ );
- $s_k$  saída do neurônio  $k$  após aplicação da função não linear.

A função não linear da Eq. (13), função logística, é responsável pela capacidade de generalização do MLP, e também permite superar o problema da separabilidade linear própria de *perceptrons* que usam a função limiar simples.

O MLP pode operar em dois modos distintos: treinamento e processamento. Durante o treinamento, vetores de entrada são apresentados à rede e os correspondentes vetores de saída são calculados. A diferença entre a saída calculada e uma saída pré-especificada (erro de avaliação) é usado pelo algoritmo de *Backpropagation* para ajustar todos os pesos da rede de forma a minimizar esse erro. A etapa de treinamento é executada repetidas vezes sobre todo o conjunto de treinamento (vetores de entrada mais vetores de saída pré-especificados) até que o erro de avaliação resulte inferior a uma tolerância pré-fixada. Este é um tipo de treinamento supervisionado porque para cada vetor de entrada existe um vetor de saída pré-especificado.

Uma vez que o MLP foi treinado, o conhecimento sobre o problema representado pelo conjunto de treinamento está contido em seus pesos e a rede pode ser usada no modo de processamento. Neste modo, somente são apresentados vetores de entrada e o MLP calcula os correspondentes vetores de saída. Nenhum ajuste no valor dos pesos é realizado no modo de processamento. Quando um vetor de entrada não faz parte do conjunto de treinamento, a rede produz a sua própria resposta para a entrada (generalização), evidenciando uma das principais características do MLP: a capacidade de resolver um problema sem utilizar as ferramentas analíticas que foram utilizadas para construir o conjunto de treinamento. Esta característica é particularmente útil em situações onde a relação funcional entre as entradas e as saídas não é perfeitamente conhecida, como é o caso no problema de previsão de carga, e também em situações nas quais a execução das ferramentas analíticas é muito demorada: o tempo de processamento gasto pelo MLP para calcular um vetor de entrada é geralmente desprezível.

### B. Treinamento Supervisionado e Não-Supervisionado

A finalidade e a aplicabilidade dos treinamentos supervisionados e não-supervisionados são diferentes. O treinamento supervisionado se aplica adequadamente em problemas de interpolação, onde se exige que a rede produza um vetor de saída para uma entrada que ela nunca viu antes. Um bom exemplo de treinamento supervisionado é o algoritmo de *Backpropagation*, o qual é usualmente utilizado em redes MLP (*Multi-Layer Perceptron*). Devido à sua natureza não-linear, o MLP é capaz de produzir com precisão complexos mapeamentos entre as variáveis de entrada e de saída. Em outras palavras, ele fornece um bom modelo para problemas representados por um determinado conjunto de treinamento. Ao fazer isso, a rede permite obviar a formulação analítica do problema. Isto pode ser muito importante em situações onde a aplicação direta de ferramentas matemáticas exige tempos de computação elevados, ou mesmo quando a formulação analítica é difícil ou impossível de ser obtida.

Redes neurais treinadas através de procedimentos não-supervisionados não são capazes de executar tarefas de in-

terpolação pois elas nunca vêm um exemplo de saídas desejadas, e portanto elas não conhecem o relacionamento entre variáveis de entrada e de saída. Exemplos deste tipo de redes são o SOM - *Self Organizing Map* e a arquitetura ART - *Adaptive Resonance Theory*. O aspecto mais importante destas redes é a habilidade em extrair características relevantes do ambiente de entrada. Estas redes podem ser efetivamente utilizadas em problemas de classificação.

### C. Algoritmo de treinamento Backpropagation

No presente projeto de P&D, o MLP é treinado através do algoritmo de *Backpropagation*. Este algoritmo é uma variação do método do gradiente, o qual procura minimizar a função erro calculando deslocamentos na direção do vetor gradiente, sendo as correções determinadas pelo método aplicadas aos pesos  $w_{jk}$  da rede (que são as variáveis independentes no treinamento). Com o algoritmo de *Backpropagation* o tempo total gasto na etapa de treinamento pode alcançar valores elevados, e também não há garantia de que um treinamento satisfatório possa ser sempre alcançado. Frequentemente o algoritmo resulta preso em mínimos locais da função erro, sem poder avançar em direção ao mínimo global desejado. Entre os fatores que afetam o treinamento estão: número de entradas e saídas (fixados pelo problema físico a ser resolvido), número e tamanho das camadas internas do MLP, tamanho e representatividade do conjunto de treinamento, e a relação física entre as variáveis de entrada e de saída (fortemente não linear ou não, por exemplo).

Algumas variantes do algoritmo básico de *Backpropagation* foram desenvolvidas com a finalidade de melhorar o desempenho quanto aos tempos de computação despendidos. Neste P&D foram consideradas as variantes conhecidas por *Método do Momento* e *Método do Suavizamento Exponencial*.

No método do momento, um termo adicional é utilizado na fórmula para correção dos pesos entre uma iteração e a iteração subsequente:

$$\Delta w_{jk}(n+1) = \eta \cdot \delta_k \cdot o_j + \alpha \cdot \Delta w_{jk}(n) \quad (14)$$

onde:

- $\Delta w_{jk}(n)$  é a correção aplicada ao peso  $w_{jk}$  da matriz  $W_i$  (camada  $i$ ) na iteração  $n$ ,
- $\eta$  é a taxa de aprendizado que permite controlar o tamanho médio das correções aplicadas aos pesos;
- $\delta_k$  é, para a última camada da rede, o erro de avaliação (diferença entre a saída desejada e a saída calculada); para as demais camadas é a propagação do erro de avaliação;
- $o_j$  é o valor da saída calculada no neurônio  $j$  na camada  $(i-1)$ ;
- $\alpha$  é o coeficiente do momento.

A parcela  $\eta \cdot \delta_k \cdot o_j$  na equação (14) é a correção do peso  $w_{jk}$  estabelecida pelo método de *Backpropagation* puro. A parcela  $\alpha \cdot \Delta w_{jk}(n)$  representa o método do momento, e corresponde a uma fração da correção aplicada ao peso na ite-

ração anterior; ou seja, cada correção é "lembrada" e aplicada nas iterações seguintes. Assim, a idéia básica do método do momento é propagar as correções iniciais, já que no início do treinamento a convergência do método de *Backpropagation* é usualmente mais rápida que nas etapas finais.

No método de suavizamento exponencial utiliza-se a mesma idéia de considerar correções aplicadas aos pesos em iterações passadas. Neste caso a equação de correção dos pesos é a seguinte:

$$\Delta w_{jk}(n+1) = (1-\alpha) \cdot \eta \cdot \delta_k \cdot o_j + \alpha \cdot \eta \cdot \Delta w_{jk}(n) \quad (15)$$

onde  $\alpha$  é o coeficiente de suavizamento exponencial. Observa-se neste caso que, dependendo do valor de  $\alpha$ , pode-se ter o algoritmo de *Backpropagation* puro ( $\alpha = 0$ , suavizamento mínimo) ou suavizamento máximo ( $\alpha = 1$ , quando a correção atual é ignorada e repete-se a correção aplicada na iteração anterior).

### D. Resultados Obtidos

Como o objetivo do P&D é o de implementar uma RNA capaz de inferir um valor em função de um histórico, ou seja, interpolar uma série de dados, a técnica a ser usada é a da rede neural *multi-layer perceptron*, (RNA MLP).

Para o teste do uso de RNA neurais para a previsão da carga da CTEEP foi usado um programa desenvolvido pelo grupo de pesquisa GAGTD para essa finalidade chamado PCC (Previsão de Curva de Carga).

O PCC permite usar uma base de dados que armazena as curvas de carga e :

- criar um conjunto de dados de treinamento da RNA
- criar um conjunto de dados de teste da RNA
- definir a configuração da RNA ( número de camadas e neurônios por camada).
- escolher os parâmetros de treinamento da RNA.

Foram feitos estudos de previsão de carga através de redes neurais utilizando-se os dados de carga da CTEEP e os dados de temperatura ambiente da cidade de São Paulo medidos na estação meteorológica do IAG, localizada no bairro da Barra Funda na cidade de São Paulo, com o objetivo de se verificar a influência da temperatura ambiente no carregamento.

Apesar da carga da CTEEP atender todo o estado de São Paulo, foi usada a temperatura da cidade de São Paulo nos testes devido à facilidade de obtenção dos dados e pelo fato que a região da grande São Paulo corresponder à cerca de 60% da carga total do estado, de modo que o peso da região no estado é considerável.

Para cada dia da semana e tipo de previsão foi definida uma rede específica e realizado um estudo variando-se os parâmetros de configuração da rede neural, para determinação da melhor arquitetura para cada caso.

O programa PCC treina uma RNA MLP (Rede Neural Artificial *Multi Layer Perceptron*) através de *backpropagation* e permite a variação dos seguintes parâmetros de treinamento:

- Iterações (n)
- Tolerância ( $\delta$ )
- Taxa de aprendizado ( $\eta$ )
- Coeficiente de método ( $\alpha$ )
- Lambda ( $\lambda$ )
- Método de treinamento

Ainda é possível a seleção do número de camadas ocultas na rede e sua quantidade de neurônios.



Figura 7 – Tela do software de redes neurais da USP

Baseados nos estudos prévios e na bibliografia, foi decidido que o melhor seria o uso de redes individuais para cada dia da semana, para cada SE e para a carga total da CTEEP.

O processo de treinamento das redes foi feito da seguinte maneira :

- Entradas : as curvas de carga dos três dias anteriores (amostradas a cada 30 minutos para facilitar os teste reduzindo o número de pontos das curvas), as temperaturas máximas e mínimas desses dias e a temperatura máxima e mínima prevista para o dia que se quer prever a carga, totalizando 152 entradas.
- Saídas : a curva de carga do dia a ser previsto, totalizando 48 saídas.
- Treinamento : foram usados os valores disponíveis no “PI”, no período de 1/02/02 a 31/12/02 para treinamento da rede.
- Teste : foram usadas as medições de 1/1/03 a 30/4/03 para teste da rede
- Os feriados e pontes foram eliminados por possuírem perfis próprios e vão precisar de rede própria para previsão.

Todas medições de carregamento foram tratadas para eliminação de erros de medição e ruídos, e utilizados no treinamento de redes com os seguintes parâmetros :

TABELA IV  
PARÂMETROS USADOS NO TREINAMENTO DAS RNAs

pas-so	Iterações internas	Iterações globais	Tolerância %	Taxa de aprendizado	Coeficiente de método	Lambda
1	10	1000	1	1	0.9	1
2	10	1000	1	0.8	0.9	1
3	10	1000	1	0.6	0.9	1
4	10	1000	1	0.4	0.9	1
5	10	1000	1	0.2	0.9	1

Foram experimentadas redes com as seguintes quantidades de camadas/neurônios :

TABELA V  
ARQUITETURA DAS RNAs TESTADAS

caso	Camadas ocultas	Neurônios			
		Camada 1	Camada 2	Camada 3	Camada 4
1	1	10			
2	1	20			
3	1	30			
4	2	10	10		
5	2	20	20		
6	2	30	30		
7	3	10	10	10	
8	3	20	20	20	
9	3	30	30	30	
10	2	20	10		
11	2	30	10		
12	3	30	20	10	
13	1	60			
14	2	60	60		
15	2	120	48		
16	2	120	120		
17	2	151	47		
18	4	60	60	60	60

Os valores de máximos e mínimos de temperatura utilizados encontram-se mostrados graficamente na figura 8 :

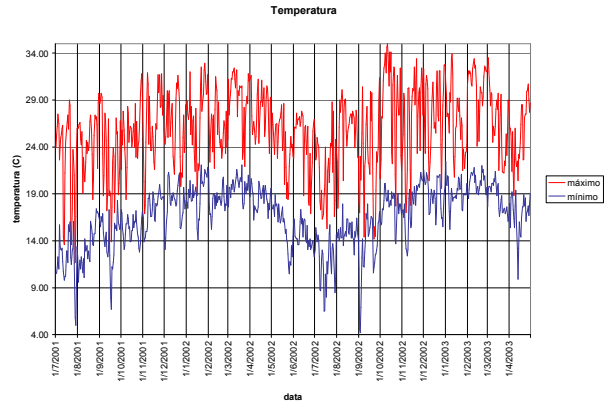


Figura 8 – Curva de temperatura máxima e mínima diária para cidade de São Paulo

Para a carga total da CTEEP, foram obtidos os seguintes resultados estão nas tabelas a seguir. A notação adotada na coluna rede, por exemplo 30\_30, indica que o caso usou uma rede com 2 camadas internas de 30 neurônios cada. Os casos de teste estão ordenados pela média percentual dos erros.

Os resultados obtidos foram:

TABELA VI  
MELHORES RNAs ENCONTRADAS

Dia semana	melhor rede	erro médio previsão	desvio padrão
segunda-feira	60_60	3.60	1.48
terça-feira	30_10	3.10	1.26
quarta-feira	120_48	2.77	0.95
quinta-feira	120_120	2.80	1.76
sexta-feira	30_30	3.40	1.60
sábado	120_120	3.37	1.40
domingo	30_30	3.72	1.15

É possível que os resultados acima sejam ainda melhorados, buscando-se um melhor ajuste nos parâmetros de treinamento da rede e com um horizonte maior de dados de medição, permitindo um maior período de treinamento da rede.

V. WAVELETS

A. Introdução

Wavelet é uma forma de onda de duração limitada e com um valor médio zero.

Enquanto a análise de Fourier se utiliza de ondas senoidais que não possuem limite e são previsíveis, as wavelets são irregulares e assimétricas. A análise de Fourier divide o sinal em senóide de diversas frequências. A análise de wavelets divide o sinal em versões deslocadas e com diferentes escalas da wavelet original (wavelet mãe).



Figura 9- Onda Senoidal



Figura 10- Wavelet

Matematicamente a análise de Fourier é dada por :

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \tag{16}$$

A análise de Wavelet é dada por :

$$C(escala, posição) = \int_{-\infty}^{\infty} f(t)\Psi(escala, posição)dt \tag{17}$$

onde  $\psi$  é a função wavelet adotada.

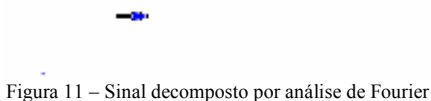


Figura 11 – Sinal decomposto por análise de Fourier



Figura 12 - Sinal decomposto por análise de Wavelet

A análise de wavelets permite que um sinal seja decomposto em duas partes. Uma com as “altas frequências” e outra com o restante do sinal. Ao aplicar o processo sucessivamente podemos decompor o sinal em componentes de frequência cada vez menores.



Figura 13 – Esquema para a decomposição wavelet em múltiplos níveis

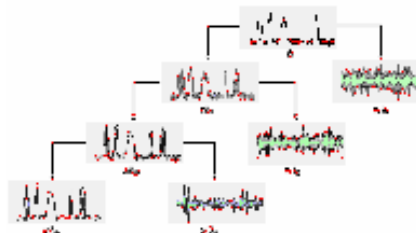


Figura 14 – Exemplo de decomposição wavelet em múltiplos níveis

B. Decomposição da Energia

A análise de wavelets pode ser usada nas curvas de energia diária com o objetivo de executar uma filtragem nas frequências mais altas e obter um sinal mais limpo, de modo a se obter um sinal onde a identificação de padrões seja mais simples.

O sinal filtrado e suas componentes de frequência mais baixa apresentam um comportamento menos aleatório de modo que uma curva de energia, que ao invés de ser prevista diretamente via RNA ou Fuzzy, poderia ser decomposta por Wavelets e ter os seus componentes, previstos separadamente e recompostos novamente no sinal, sendo nesse processo desprezados os componentes que representam os ruídos.

Inicialmente foi feito um estudo da evolução da energia diária através de Wavelets com o objetivo de se identificar padrões ao longo do ano.

Foram utilizadas as curvas de energia de todas as segundas-feiras de 26/11/01 a 10/2/03 compondo um total de 64 pontos para a decomposição através de Wavelet daubechies 4 (db4).

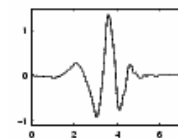


Figura 15 - wavelet db4



Figura 16 – Energia diária da CTEEP ( MWh x data )



Figura 17- Curva de energia com 5 decomposições

Observamos que além de 4 decomposições, não obtemos mais informações das curvas pois as decomposições além de D5 só repetem a forma com escalas diferentes.

A partir dessa informação, foi feito o mesmo estudo de decomposição para todos os dias de semana.

Os períodos abrangidos são :

TABELA VII  
PERÍODOS ABRANGIDOS NA ANÁLISE DE WAVELETS

Dia da semana	inicio	Fim
segunda-feira	26/11/0	10/02/0
terça-feira	27/11/0	11/02/0
quarta-feira	28/11/0	12/02/0
quinta-feira	15/11/0	06/02/0
sexta-feira	23/11/0	07/02/0
Sábado	24/11/0	08/02/0
Domingo	25/11/0	09/02/0

Foram obtidas as seguintes curvas :

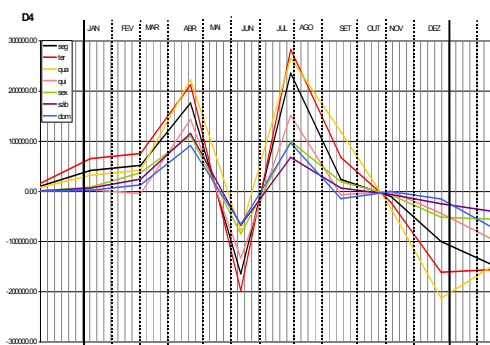


Figura 18 - Curva da decomposição D4 da energia por dia de semana

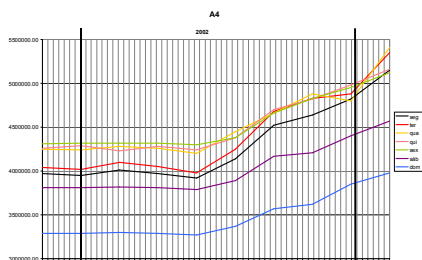


Figura 19 - Curva de energia após 4 filtragens

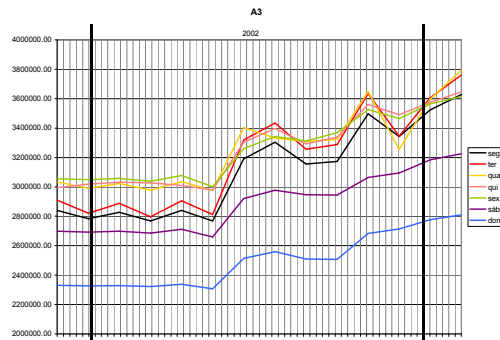


Figura 20 - Curva de energia após 3 filtragens

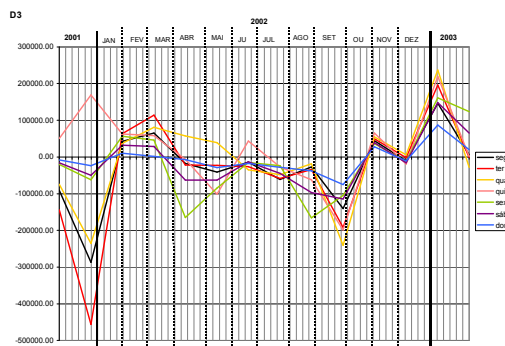


Figura 21 - Curva da decomposição D3 da energia por dia de semana

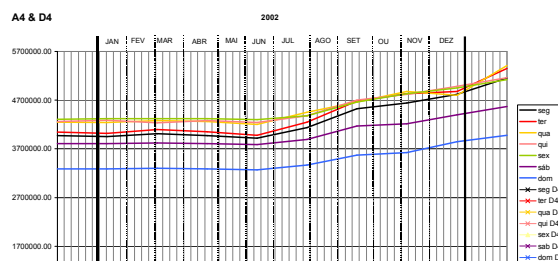


Figura 22 - Curva filtrada por Wavelets e Decomposição D4

Pela análise dos gráficos vemos que a energia ao longo do ano apresenta um crescimento quase linear a partir de julho. Nessa época a carga estava retomando os valores anteriores ao racionamento, que se encerrou oficialmente em Fevereiro. Antes disso, a energia se manteve praticamente constante.

A análise da decomposição D4 mostra que a carga possui períodos de 3 meses onde o crescimento da energia varia. O aumento da energia é mais acentuado de Abril até o final de Junho e diminui em Julho e final de Dezembro.

A análise de D3 mostra variações de cerca de 1 mês de duração onde o crescimento da energia aumenta e diminui alternadamente de setembro a janeiro.

A mesma análise será futuramente aplicada às curvas de carga com o objetivo de se obter componentes que possam vir a ser mais facilmente previstas por RNAs ou Lógica Fuzzy.

## VI. CONCLUSÃO

O objetivo do projeto é o desenvolvimento de um sistema que permita a previsão da curva de carga para as 24 horas do dia, às zero hora e que seja capaz de refinar essa previsão no decorrer do dia utilizando para isso as medições obtidas.

Para atingir tal objetivo, verificamos que as RNAs são uma técnica que nos permite fazer previsões de base (curva de carga do dia), com erros da ordem de 3%, mas que podem ser melhorados com o aprimoramento das redes e com o aumento do histórico de cargas.

Simulações efetuadas indicam que o melhor resultado é obtido com uma RNA específica para cada dia da semana, tendo como entradas os 3 dias anteriores de mesmo tipo e suas respectivas temperaturas máximas e mínimas diárias.

Os algoritmos de inferência Fuzzy fornecem bons resultados para as previsões de seguimento de curto período, de 15 minutos até 1 hora, e podem ser usados para refinar a previsão no decorrer do dia.

As transformadas Wavelets são uma ferramenta promissora para o uso na identificação de padrões e sazonalidades nas curvas de carga e na filtragem de ruídos das medições.

## VII. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] S.H. Ling, H. K. Lam, F.H.F. Leung, P.K.S. Tam, "A Novel GA-Based Neural Network for Short-Term Load Forecasting," *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 3, pp. 2761-2766, 2002.
- [2] T. M. Peng, N. F. Hubele, G. G. Karady, "Advancement in the application of neural networks for short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 7, pp. 250-257, Feb. 1992.
- [3] D. C. Park, M. A. El-Sharkawi, R. J. Marks II, L. E. Atlas, M. J. Damborg, "Electric Load Forecasting Using an Artificial Neural Network," *IEEE Transactions on Power Systems*, vol. 6, pp. 442-449, May 1991.
- [4] M. A Teixeira, G. Zaverucha, "Fuzzy Markov Predictor in Electric Load Forecasting," *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 3, pp. 2416-2421, 2002.
- [5] Q. Chen, J. Milligan, E. H. Germain, P. Shamsollahi, K. W. Cheung, R. Raub, "Implementation and performance analysis of very short term load forecaster based on the electronic dispatch project in ISO New England," *2001 Large Engineering Systems Conference on Power Engineering*, pp. 98-104, 2001.
- [6] K. S. Swarup, B. Satish, "Integrated ANN approach to forecast load," *IEEE Computer Applications in Power*, vol. 15, pp. 46-51, Apr. 2002.
- [7] G. Chicco, R. Napoli, F. Piglione, "Load pattern clustering for short-term load forecasting of anomalous days," *2001 IEEE Porto Power Tech Proceedings*, vol. 2, 2001.
- [8] C. N. Lu, H. T. Wu, S. Vemuri, "Neural Network Based Short Term Load Forecasting," *IEEE Transactions on Power Systems*, vol. 8, pp. 336-342, Feb 1993.
- [9] H. S. Hippert, C. E. Pedreira, R. C. Souza, "Neural networks for short-term load forecasting: a review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, pp. 44-55, Feb. 2001.
- [10] A. P. A. da Silva, U. P. Rodrigues, A. J. R. Reis, L. S. Moulin, "NeuroDem - a neural network based short term demand forecaster," *2001 IEEE Porto Power Tech Proceedings*, vol. 2, 2001.
- [11] T. Senjyu, H. Takara, K. Uezato, T. Funabashi, "One-hour-ahead load forecasting using neural network," *IEEE Transactions on Power Systems*, vol. 17, pp. 113-118, Feb. 2002.
- [12] D. Srinivasan, S. S. Tan, C. S. Chang, K. Chan, "Parallel neural network-fuzzy expert system strategy for short-term load forecasting: system implementation and performance evaluation," *IEEE Transactions on Power Systems*, vol. 14, pp. 1100-1106, Aug. 1999.
- [13] S. Osowski, K. Siwek, "Regularisation of neural networks for improved load forecasting in the power system," *IEE Proceedings Generation, Transmission and Distribution*, vol. 149, pp. 340-344, May 2002.
- [14] S. H. Ling, F. H. F. Leung, H. K. Lam, P. K. S. Tam, "Short-term Daily Load Forecasting in an Intelligent Home with GA-Based Neural Network," *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 1, pp. 997-1001, 2002.
- [15] K-H. Kim, H-S. Youn, Y-C. Kang, "Short-term load forecasting for special days in anomalous load conditions using neural networks and fuzzy inference method," *IEEE Transactions on Power Systems*, vol. 15, pp. 559-565, May 2002.
- [16] K-L. Ho, Y-Y. Hsu, C-C. Yang, "Short-Term load forecasting using a multilayer Neural Network with an adaptive learning algorithm," *IEEE Transactions on Power Systems*, vol. 7, pp. 141-149, Feb. 1992.
- [17] H. Yoo, R. L. Pimmel, "Short-term load forecasting using a self-supervised adaptive neural network," *IEEE Transactions on Power Systems*, vol. 14, pp. 779-784, May. 1999.
- [18] S-T. Chen, D. C. Yu, A. R. Moghaddamjo, "Weather sensitive short-term load forecasting using nonfully connected artificial neural network," *IEEE Transactions on Power Systems*, vol. 7, pp. 1098-1105, Aug. 1992.
- [19] H. C., C. N. Lu, "Automatic Fuzzy Model Identification for Short-term load forecast," *IEE Proc. Gener. Transm. Distrib.*, Vol 146, No 5, Sep 1999.
- [20] M. Sugeno, T. Tomohiro, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEE Transactions on Systems, Man and Cybernetics*, Vol SMC-15, No 1, Jan/Fev 1985.
- [21] J. Yasuoka, J. L. P. Brittes, J. A. Jardini, "Previsão de Carga em Tempo Real em Subestações de Distribuição - uma aplicação prática na CPFL (Brasil)," *IX Encontro Regional Latino-Americano da Cigré - IX ERLAC*, Foz do Iguaçu, Brasil, Maio 2001.
- [22] J. Yasuoka, J. L. P. Brittes, J. A. Jardini, "Artificial Neural Network-Based Distribution Substation and Feeder Load Forecast," *16th International Conference on Electricity Distribution - CIRED 2001*, Amsterdam, The Netherlands, June 2001.
- [23] J. Yasuoka, J. L. P. Brittes, J. A. Jardini, "ANN-based Real-Time Short-Term Load Forecasting in Distribution Substations," *IEEE/PES Transmission & Distribution Latin America 2002*, São Paulo, Brasil, March 2002.
- [24] J. Yahuoka, "Previsão de Carga em Sistemas de Distribuição de Energia Elétrica Utilizando Redes Neurais Artificiais", *Dissertação de Mestrado*, Dept. Eng. Elétrica, Escola Politécnica, Univ. de São Paulo, 2002.
- [25] A. J. Wood, B. F. Wollenberg, *Power Generation, Operation and Control*, New York, John Wiley & Sons, 1984.