

UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

REDE NEURAL ARTIFICIAL PARA
PREVISÃO DE CONSUMO ENERGÉTICO

Trabalho de formatura apresentado à Escola
Politécnica da Universidade de São Paulo para
obtenção do título de Graduação em Engenharia

Eduardo Tomita Campoleoni

Orientador: Prof. Flávio A. S. Fiorelli



São Paulo

2006

RESUMO

Este trabalho apresenta o desenvolvimento e implementação de uma rede neural artificial, utilizando o modelo de *feed-forward*, para prever o consumo energético diário de uma edificação. Como estudo de caso utilizou-se o prédio da Reitoria da USP, localizado na Cidade Universitária “Armando de Salles Oliveira”, em São Paulo.

Partindo da hipótese de que as variações no consumo decorrem predominantemente dos equipamentos associados ao condicionamento de ar, foram considerados como parâmetros para análise as condições climáticas locais do prédio (temperatura, umidade e radiação), obtidas junto ao Instituto Astronômico e Geofísico da USP (IAG-USP).

Treinada a rede com os dados climáticos e de consumo da edificação coletados em 2003 e 2004, conseguiu-se prever o consumo dos três primeiros meses do ano de 2005 com erro médio de 10% em relação aos dados coletados para o mesmo período.

ABSTRACT

This work shows the development and implementation of an artificial neural network, using the feed-forward model, to predict the daily energy load of a building. The administration building at Cidade Universitária "Armando de Salles Oliveira", at São Paulo, was the case study used.

Based on the assumption that load variation are mainly due to air conditioning equipments, climatic conditions (temperature, humidity and radiation) near by the building has been considered as parameters for the analysis. The climatic data has been provided by the Instituto Astronômico e Geofísico da USP (IAG-USP).

Using climatic and energy load data acquired at 2003 and 2004 to train the network, it was possible to predict the load from January to March 2005 with 10% difference in average to the real values for the same months.

SUMÁRIO

1 – INTRODUÇÃO E OBJETIVOS	1
2 – REDES NEURAIS	3
2.1 – Introdução	3
2.2 – As Redes Neurais Artificiais	4
2.3 – Operação	7
3 – DESENVOLVIMENTO E IMPLEMENTAÇÃO DO MODELO	11
3.1 – Premissas	11
3.2 – Implementação do Modelo	12
3.3 – Testes Iniciais	14
4 – TRATAMENTO DOS DADOS	19
4.1 – Descrição da Instalação Predial	19
4.2 – Perfil Climático e de Consumo	19
4.3 – Tratamento dos Dados	21
5 – RESULTADOS	23
5.1 – Previsão de Consumo a partir da Temperatura	23
5.2 – Previsão de consumo a partir da Temperatura, Umidade e Radiação	32
6 – CONCLUSÕES	38
BIBLIOGRAFIA	39

1 – INTRODUÇÃO E OBJETIVOS

Nos últimos anos tem-se intensificado a preocupação com o consumo energético no país, em especial após a possibilidade de ocorrência de “apagões” em 2001. Dessa forma, a correta estimativa do consumo energético torna-se valiosa ferramenta para gestão predial. Dos diversos equipamentos e sistemas que integram uma instalação, os equipamentos associados ao condicionamento de ar são os maiores consumidores, justificando assim a importância de se conseguir prever seu consumo energético.

A utilização de modelos analíticos nesse processo é tarefa difícil, pois além da grande quantidade de variáveis a considerar, há a necessidade de se conhecer propriedades e características da instalação e itens que a compõem e que muitas vezes não estão disponíveis.

Uma solução, quando existe a disponibilidade de dados passados do consumo energético da instalação é a utilização de modelos que, a partir desses dados antigos, extrapolem o consumo para novas situações. Um desses modelos é o de Redes neurais.

Rede neural (a rigor, rede neural artificial) é o nome genérico para diversos modelos matemáticos que simulam, em parte, o funcionamento de uma rede neural biológica (o cérebro humano, por exemplo). A característica mais importante desses modelos, e que os tornam interessantes para o presente estudo, é justamente o fato de a rede ter a capacidade de, uma vez alimentada com dados para situações conhecidas (etapa chamada de “treino” ou “aprendizado”), “aprender” a regra que rege o fenômeno físico em estudo e a partir dessa regra fornecer resultados para situações ainda não vistas.

Assim, o objetivo do presente trabalho foi implementar um rede neural artificial do tipo *feed-forward* que permitisse estimar o consumo energético de sistemas de ar condicionado de uma instalação predial, com base em dados climáticos de fácil acesso (temperatura, umidade e radiação) e em dados anteriores de consumo da edificação.

Como estudo de caso para o presente trabalho foi utilizado o prédio da Reitoria da USP, localizado na Cidade Universitária “Armando de Salles Oliveira”, em São Paulo. As características da edificação e as hipóteses adotadas serão apresentadas ao longo do trabalho.

É importante ressaltar que, por existir uma etapa de aprendizado a partir dos dados específicos de uma edificação, a rede desenvolvida irá fornecer resultados válidos somente para as mesmas condições nas quais ela foi treinada, ou seja, ela não poderá ser usada para prever o consumo de outra instalação predial sem passar novamente pela etapa de treino.

2 – REDES NEURAIS

2.1 – Introdução

Redes Neurais Artificiais (*Artificial Neural Network*, ANN) são modelos matemáticos que se propõem simular o funcionamento de redes neurais biológicas, como o cérebro humano, por exemplo.

A Fig. 2.1 ilustra um neurônio biológico. Ele opera recebendo impulsos de neurônios vizinhos, através de conexões chamadas sinapses. A combinação desses impulsos entre si, associada a um nível mínimo de sensibilização, resulta no impulso disparado deste neurônio para os demais neurônios que nele se conectam. Os impulsos recebidos podem agir como excitadores ou inibidores no comportamento do neurônio, e por consequência no impulso gerado por ele.

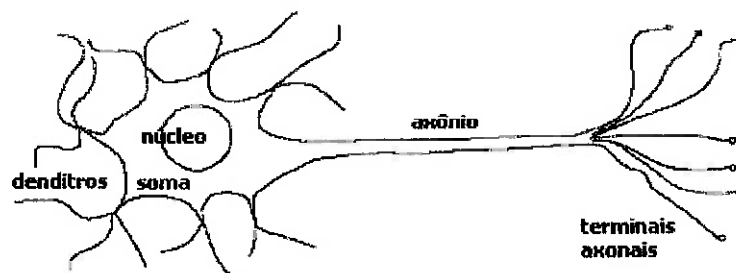


Figura 2.1. Neurônio biológico

A atividade cerebral, e em última análise o pensamento humano, é o resultado da atividade de bilhões de neurônios, cada um conectado a milhares de outros neurônios, cada um deles gerando centenas de impulsos por segundo. Estima-se que o cérebro humano tenha na ordem de 10^{11} neurônios, e que cada um esteja conectado

a 10^4 outros neurônios, gerando impulsos a uma frequência de 100 Hz.

Cada neurônio é formado por um corpo, chamado soma. Ele contém primordialmente o núcleo da célula. Ao seu redor ficam os dendritos, que são os receptores dos impulsos gerados por neurônios próximos. Todos os impulsos recebidos, que podem ter caráter excitatório ou inibidor, são combinados e o resultado irá determinar se o neurônio será ativado, ou não.

Caso ele seja ativado, irá gerar um impulso elétrico na base do axônio. Esse impulso será transmitido por toda sua extensão até os terminais axonais, que por sua vez conectam-se aos dendritos de outros neurônios através das sinapses. Os neurônios não tocam fisicamente uns nos outros, existe entre eles uma separação, e os terminais axonais são responsáveis por transformar o impulso elétrico que recebem em reações eletro-químicas, que são sentidas pelos dendritos do outro neurônio.

Outras características fundamentais do comportamento dos neurônios são a capacidade de adaptar a sua resposta a mudanças nos estímulos recebidos e também a de responder de uma certa forma baseado nos padrões de estímulos previamente recebidos. Ou seja, ele pode aprender.

Ainda que seja bem que é bem conhecido o funcionamento de um neurônio individualmente, ainda há muita conjectura quanto a forma como eles se organizam e quanto aos mecanismos usados para se adaptar conforme os estímulos recebidos.

2.2 – As Redes Neurais Artificiais

Como exposto, o desenvolvimento de redes neurais artificiais baseia-se na analogia com sua contra-parte biológica, tanto que sua unidade básica também é chamada neurônio.

A Fig. 2.2 apresenta o modelo de um neurônio artificial. Ele consiste em

múltiplas entradas (x_1, x_2, \dots) e uma saída (Y). Cada entrada é modificada de acordo com um peso (w_1, w_2, \dots). O neurônio irá combinar a soma do produto dos pesos com as entradas e comparar essa soma com um valor de referência (t). Por fim, o resultado até então obtido é passado a uma função de ativação (F) para determinar o valor de sua saída.

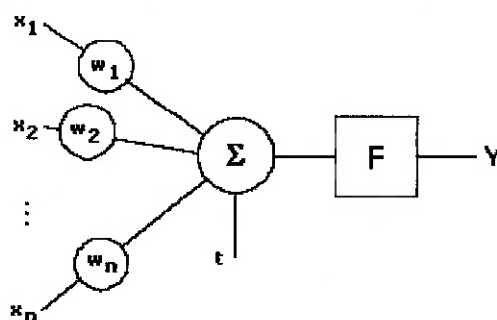


Figura 2.2. Neurônio artificial

Como a forma de organização dos neurônios biológicos não é bem conhecida, diversos modelos artificiais foram propostos, cada um deles originando redes neurais distintas. O mais conhecido e usado é o modelo de *feed-forward*.

Nesse modelo, os neurônios são distribuídos em camadas. A primeira de suas camadas, chamada camada de entrada (*input layer*) recebe os sinais do meio físico. A última de suas camadas, chamada de camada de saída (*output layer*) fornece o resultado gerado pela rede. Entre elas pode-se ter nenhuma, uma ou mais camadas intermediárias, chamadas de camadas ocultas (*hidden layers*).

A Fig. 2.3 ilustra uma rede neural no modelo *feed-forward* com quatro entradas, uma saída e onze neurônios dispostos em quatro camadas (duas ocultas).

De forma geral, a camada de entrada é considerada como um distribuidor do sinal vindo do meio físico. As camadas ocultas são os categorizadores dos sinais recebidos e a camada de saída é considerada o agregador das características descobertas.

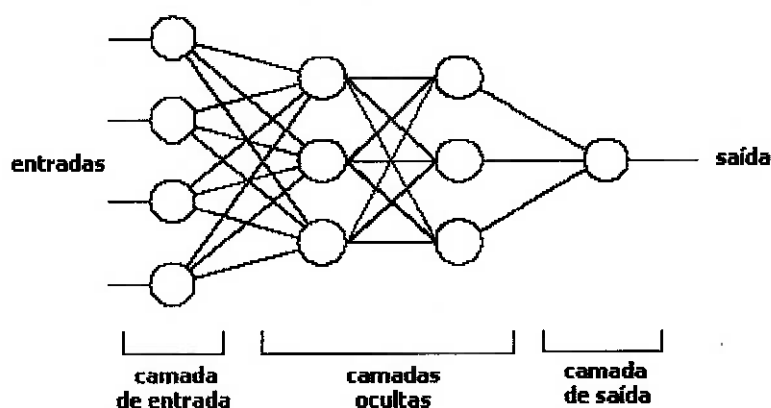


Figura 2.3. Exemplo de rede neural *feed-forward*

Um aspecto deve ser destacado quanto a organização da rede. Enquanto no modelo de *feed-forward*, os neurônios de uma camada conectam-se somente àqueles da camada imediatamente seguinte e anterior, existem outros modelos que permitem formas mais complexas de organização. A Fig. 2.4 ilustra alguns exemplos.

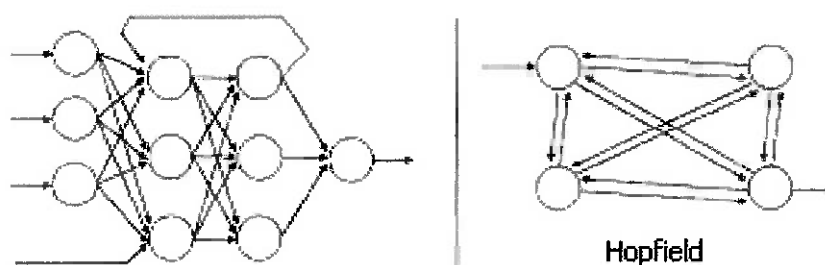


Figura 2.4. Exemplos de redes neurais mais complexas

Além das formas fixas de organização, existem modelos nos quais a própria rede muda sua organização conforme o aprendizado. Esses modelos são chamados *Self Organizing Maps*, e o exemplo mais conhecido é o de Kohonen.

Devido a grande quantidade de modelos para redes neurais propostos, elas podem ser classificadas de diversas formas. O detalhamento de tal classificação foge

ao escopo do presente trabalho, mas um breve resumo é mostrado na Tab. 2.1.

Tabela 2.1. Tipos de redes neurais

Quanto ao	Denominação	Descrição
Aprendizado	<i>Supervised</i>	a própria rede compara o resultado obtido com valores previamente conhecidos
	<i>Unsupervised</i>	a rede não tem conhecimento dos valores corretos dos resultados – é usada em geral para detectar entradas supérfluas ao problema
Topologia	<i>Feedforward</i>	não há referências circulares entre neurônios
	<i>Feedback</i>	a saída de um neurônio pode ser entrada de um neurônio de uma camada anterior
Dados de Entrada	<i>Categorizados</i>	os dados são classificados em categorias (homem e mulher; azul, vermelho ou amarelo, etc)
	<i>Quantitativos</i>	os dados por si só são valores numéricos com significado (velocidade, peso, temperatura, etc)

2.3 – Operação

O valor da saída de cada neurônio, que é uma função de suas entradas, é dado por:

$$U_j = \sum (x_i w_{ij}) \quad (1)$$

$$Y_j = F_j (U_j + t_j) \quad (2)$$

Para cada neurônio j , cada uma de suas entradas i é multiplicada pelo peso w_{ij} pré-estabelecido. A soma U_j desses produtos é o valor interno do neurônio. A soma do valor interno com o valor de referência t_j pré-estabelecido é enviado à função de ativação F_j , resultando no valor de saída Y_j do neurônio.

Comparando com o exemplo biológico, pode-se dizer que a equação (1) retrata a composição das entradas, enquanto a equação (2) corresponde à ativação do neurônio.

Alguns exemplos de funções de ativação geralmente usadas estão listados na Tab. 2.2, e ilustrados na Fig. 2.5.

Tabela 2.2. Funções de ativação

Função de Ativação	Expressão
Linear	$y = \alpha x$
<i>Threshold</i> *	$y = 0$ se $x < 0$ e $y = 1$ se $x \geq 0$
<i>Sigmoid</i> *	$y = 1 / (1 + \exp(-\alpha x))$
Gauss	$y = \exp(-\alpha x^2)$

* Obs: as funções *threshold* e *sigmoid*, podem ser usadas com valor inferior -1 ao invés de 0, e nesse caso são chamadas simétricas.

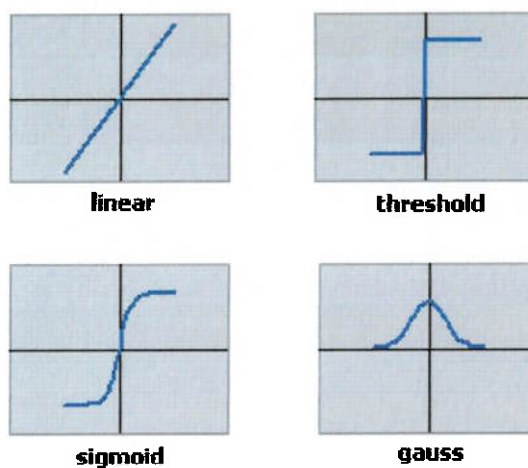


Figura 2.5. Funções de ativação

O aprendizado da rede neural, chamado treino (*training*), é feito comparando-se, para cada conjunto de entrada, o valor da variável de saída da rede com o esperado (conhecido) para o conjunto de entrada. Da diferença entre eles pode-se determinar o erro, que será propagado pela rede de forma a ajustar os pesos e valores de referência de cada neurônio, a partir de sua última camada até chegar à primeira.

Esse método de correção é chamado *back-propagation*, e inicia-se na última camada:

$$e_j = Y_j (1 - Y_j) (d_j - Y_j) \quad (3)$$

$$t_j = t'_j + \lambda e_j \quad (4)$$

$$w_{ij} = w'_{ij} + \lambda e_j x_i \quad (5)$$

O erro e_j do j -ésimo neurônio da camada de saída é dado pelo produto da saída fornecida Y_j , com o seu complemento e com diferença entre a saída esperada d_j da saída fornecida. O novo valor de referência t_j do neurônio é dado pela soma do valor anterior t'_j com o produto da taxa de aprendizado λ com o erro e_j calculado. Já o novo peso w_{ij} para a i -ésima entrada do neurônio é dado pela soma do peso anterior w'_{ij} com o produto da taxa de aprendizado λ com o erro e_j calculado e o valor de sua própria entrada x_i .

A taxa de aprendizado é responsável por definir a contribuição maior ou menor do erro calculado ao ajuste do antigo peso. Quanto maior essa taxa, mais rápido será o aprendizado, mas, para entradas com grande variação, corre-se o risco da rede não conseguir aprender.

Após os erros da última camada terem sido calculados, e os novos valores ajustados, parte-se para a camada anterior. Entretanto, como não se conhece o valor esperado para as saídas dessa camada, faz-se necessário usar uma expressão modificada para o cálculo do erro:

$$e_j = Y_j (1 - Y_j) \sum (e_k w'_{jk}) \quad (6)$$

Neste caso, a diferença entre o valor esperado e o fornecido é substituída pela soma do produto dos pesos w'_{jk} antes do ajuste com os erros e_k dos k neurônios da camada seguinte.

Uma forma alternativa de trabalhar o processo de ajuste dos pesos é dada por:

$$w_{ij} = w'_{ij} + (1 - M) \lambda e_j x_j + M (w'_{ij} - w''_{ij}) \quad (7)$$

A equação (7) é similar à equação (5), com acréscimo do momento M . Esse fator permite que o cálculo do novo peso w_{ij} leve em conta, não somente o peso anterior w'_{ij} , mas também o peso anterior a esse w''_{ij} . Note que, fazendo $M = 0$ temos de volta a equação anterior.

Conforme vai-se treinando a rede, espera-se que a soma dos erros de todos os neurônios diminua e os valores de referência e pesos estabilizem em torno dos valores definitivos. Quando essa situação ocorrer, o processo de treino pode ser interrompido e a rede estará pronta para uso.

Entretanto, os valores dos erros podem atingir um patamar ao invés de sempre diminuir, e os valores de referência e pesos oscilarem sem indício de estabilizar. Neste caso, a rede não conseguiu aprender a regra que rege o conjunto de dados que a alimentou. O treino deve ser interrompido e, como correção, pode-se alterar a quantidade das camadas e de neurônios por camada para uma nova tentativa de aprendizado. Caso essas alterações nos parâmetros da rede ainda não se mostrem eficazes, pode modificar-se o conjunto de dados entrada ou, até mesmo, escolher outro modelo de rede neural.

3 – DESENVOLVIMENTO E IMPLEMENTAÇÃO DO MODELO

3.1 – Premissas

Redes neurais têm sido usadas, em geral, para duas finalidades: categorizar um conjunto de dados com base em conjuntos já categorizados e prever o comportamento de um sistema com base em dados históricos. Elas se mostram úteis para esses fins justamente por seu modelamento não depender do conhecimento da regra física que rege o fenômeno em estudo. Essa é a principal razão por termos escolhido esse modelo para o presente trabalho: sem que haja a necessidade de construir o modelo físico-matemático de uma instalação predial, podemos estimar o seu consumo energético com base em dados passados conhecidos.

Dentre os diversos modelos de Redes neurais já propostos, o *feed-forward* com *back-propagation* adotado é o mais simples, e talvez por isso um dos mais estudados e utilizados.

Apesar do modelo adotado não exigir o conhecimento do fenômeno físico em estudo, esse conhecimento se faz necessário ao escolher o conjunto de dados de entrada para a rede, de modo que ele seja coerente com a saída que se deseja. Para tanto, assumimos as seguintes hipóteses:

- a) Variação no consumo energético da instalação predial em análise decorre predominantemente devido ao uso dos equipamentos de condicionamento de ar;
- b) O uso desses equipamentos é influenciado direta e principalmente pelas condições climáticas locais do prédio;

- c) As condições climáticas são uniformes para toda a instalação e seus arredores.

Dessa forma, foram consideradas como entrada as condições climáticas locais (temperatura, umidade e radiação) e como saída o consumo energético da instalação.

3.2 – Implementação do Modelo

Para implementação do modelo desenvolveu-se um programa de computador em linguagem C. O programa desenvolvido possui dois modos de operação:

- a) “treino da rede”: no qual o programa é alimentado com os dados climáticos ($x_{j1}, x_{j2}, \dots x_{jn}$) e com os respectivos dados de consumo (d_j), e produz, como saída, um arquivo com os dados da rede (pesos w_{ij} e valores de referência t_j);
- b) “uso da rede”: no qual o programa, fazendo uso de um arquivo de parâmetros gerado na etapa de treino, é alimentado com um conjunto de dados climáticos ($x_{j1}, x_{j2}, \dots x_{jn}$) e fornece, como saída, as estimativas para os respectivos consumos energéticos (Y_j).

De modo a facilitar o uso do programa, deixamos os parâmetros que definem a topologia da rede como variáveis a serem informadas no momento de sua execução. Assim, é possível testar para um mesmo conjunto de dados diversas configurações da rede, sem que haja a necessidade de se compilar o programa a cada mudança feita. A Tab. 3.1 traz os parâmetros que o programa pode receber.

Todos esses parâmetros deverão ser passados por meio de um arquivo de configuração. O programa ao ser executado (via linha de comando) pode receber ainda mais dois parâmetros: o arquivo de configuração a ser usado e o modo de operação escolhido. A Tab. 3.2 traz o análogo à Tab. 3.1 para os parâmetros via linha de comando.

Por fim, todos os arquivos que o programa irá receber deverão estar no formato texto (apesar de não ser necessário a extensão .txt) e os dados numéricos

neles deverão usar ponto como separador decimal e nenhum separador de milhar (por exemplo, usar 3456.45 ao invés de 3456,45 ou 3.456,45). Analogamente, todos os arquivos que o programa gerar seguirão esse mesmo formato.

Tabela 3.1. Parâmetros de configuração

Parâmetro *	Descrição	Valor padrão
taxa_aprend (t)	Taxa de aprendizado da rede (ver item 2.3)	0.7
num_entradas (t, u)	Quantidade de entradas que a rede irá ter (e por consequência, a quantidade de neurônios na primeira camada)	1
num_saidas (t, u)	Quantidade de saídas que a rede irá ter (e por consequência, a quantidade de neurônios na última camada)	1
num_camadas (t)	Quantidade de camadas que a rede irá ter, considerando a camada de entrada e de saída (no mínimo 3 e no máximo 7)	2
num_neuronios_i (t) (i = 1 a 5, conforme num_camadas)	Quantidade de neurônios em cada uma das camadas ocultas.	0
funcao_ativacao (t, u)	Função de ativação (ver item 2.3). Opções são: <i>linear</i> , <i>threshold</i> , <i>threshold-sim</i> , <i>sigmoid</i> , <i>sigmoid-sim</i> , <i>gauss</i>	<i>linear</i>
erro (t)	A etapa de treino será terminada quando $\max Y_j - d_j \leq \text{erro}$	0.001
max_iteracoes (t)	Limite para quantidade de iterações (ao atingir esse limite, considera-se que a rede não foi capaz de aprender, e o programa é finalizado)	500000
arq_treino (t)	Arquivo com os dados para a etapa de treino	rneural.treino
arq_rede (t, u)	Arquivo com os dados da rede	rneural.rede
arq_uso (u)	Arquivo com os dados de entrada para a etapa de uso	rneural.uso
arq_saida (u)	Arquivo a ser gerado pelo programa com os resultados para a etapa de uso	rneural.saida

* Obs: as letras t e u ao lado do nome do parâmetro indicam se ele é usado, respectivamente, no modo “treino”, ou no modo “uso”.

Tabela 3.2. Parâmetros de linha de comando

Parâmetro	Descrição	Valor padrão
-f <i>arquivo</i>	Nome do arquivo com parâmetros configuração.	rneural.conf
-m <i>modo</i>	Modo de execução do programa. Opções são: <i>treino</i> , <i>uso</i> e <i>teste</i> *.	<i>uso</i>

* Obs: no modo *teste* o programa apenas mostra na tela os parâmetros que seriam usados com base no arquivo de configurações informado e nos valores padrão para os parâmetros omitidos.

3.3 – Testes Iniciais

Inicialmente, verificamos se a rede criada tem capacidade para aprender alguns exemplos simples. Isso para nos certificarmos que o programa foi, de fato, desenvolvido corretamente.

Para tanto, escolhemos uma função linear, uma função quadrática e a função booleana xor.

3.3.1 – Função Linear

Escolhemos os parâmetros apresentados na Tab. 3.3 para testes. Dentre os parâmetros para a rede, o conjunto apresentado é aquele que forneceu melhores resultados. O resultado está ilustrado na Fig. 3.1.

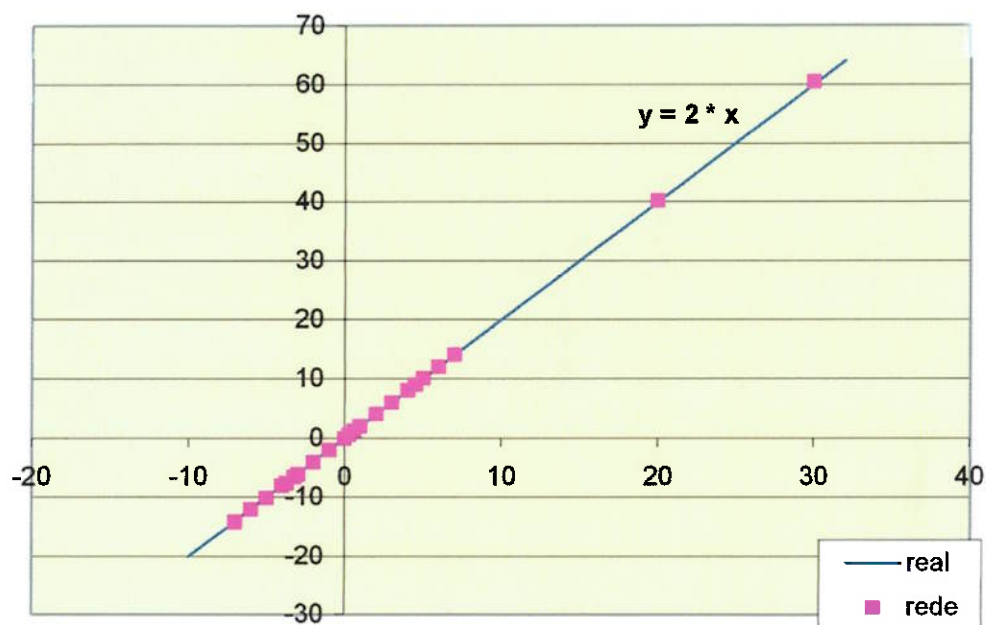
Tabela 3.3. Parâmetros para teste da função linear

Função real	$y = 2x$
Variáveis de Entrada	1 (x)
Variáveis de Saída	1 (y)
Dados para treino	10
Dados para validação	20
Camadas	3
Neurônios na camada de entrada	1
Neurônios na segunda camada	18
Neurônios na camada de saída	1

(continua...)

Tabela 3.3 (cont). Parâmetros para teste da função linear

Taxa de aprendizado λ	0,7
Função de ativação	Linear

**Figura 3.1.** Resultados para função linear

Notamos que os valores apresentados pela rede coincidem de forma bastante satisfatória com os valores esperados.

3.3.2 – Função Quadrática

Analogamente ao feito para a função linear, os parâmetros usados para a função quadrática estão apresentados na Tab. 3.4, e os resultados na Fig. 3.2.

Para este caso, a rede não foi capaz de aprender a regra que rege os dados de entrada. Alterações nos parâmetros da rede não apresentaram resultados significativos. Conclui-se portanto que o modelo adotado usando não é o mais adequado para este tipo de entrada particular.

Note que usamos somente 8 neurônios na camada oculta, pois testes feitos com mais neurônios nesta camada, e ainda com mais camadas ocultas apresentaram o mesmo perfil para os resultados. Optamos portanto em apresentar a situação mais simples que ilustra o resultado da rede.

Tabela 3.4. Parâmetros para teste da função quadrática

Função real	$y = x^2 + 3x - 5$
Variáveis de Entrada	1 (x)
Variáveis de Saída	1 (y)
Dados para treino	80
Dados para validação	80
Camadas	3
Neurônios na camada de entrada	1
Neurônios na segunda camada	8
Neurônios na camada de saída	1
Taxa de aprendizado λ	0,7
Função de ativação	Linear



Figura 3.2. Resultados para função quadrática

3.3.3 – Função XOR

Usamos como último teste a função booleana xor (ou exclusivo) para duas entradas. Essa função retorna falso se as entradas forem iguais, e verdadeiro caso contrário. A tabela verdade para essa função esta explicitada na Tab. 3.5, os parâmetros usados, na Tab. 3.6 e os resultados, na Fig. 3.3.

Escolhemos a função xor pois ela é considerada caso clássico para testes de redes neurais, até mesmo por já ter sido formalmente provado que certas topologias de rede são incapazes de aprender essa função lógica.

Tabela 3.5. Tabela verdade para função xor

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 3.6. Parâmetros para teste da função xor

Função real	xor
Variáveis de Entrada	2 (a, b)
Variáveis de Saída	1 (y)
Dados para treino	4
Dados para validação	4
Camadas	3
Neurônios na camada de entrada	2
Neurônios na segunda camada	4
Neurônios na camada de saída	1
Taxa de aprendizado λ	0,7
Função de ativação	Sigmoid

Para este caso a rede apresentou resultados bem próximos dos esperados. Podemos dizer que a rede foi capaz de aprender a função xor. Note que foram usados somente 4 neurônios na camada oculta (ao passo que para a função linear, foram necessários 18).

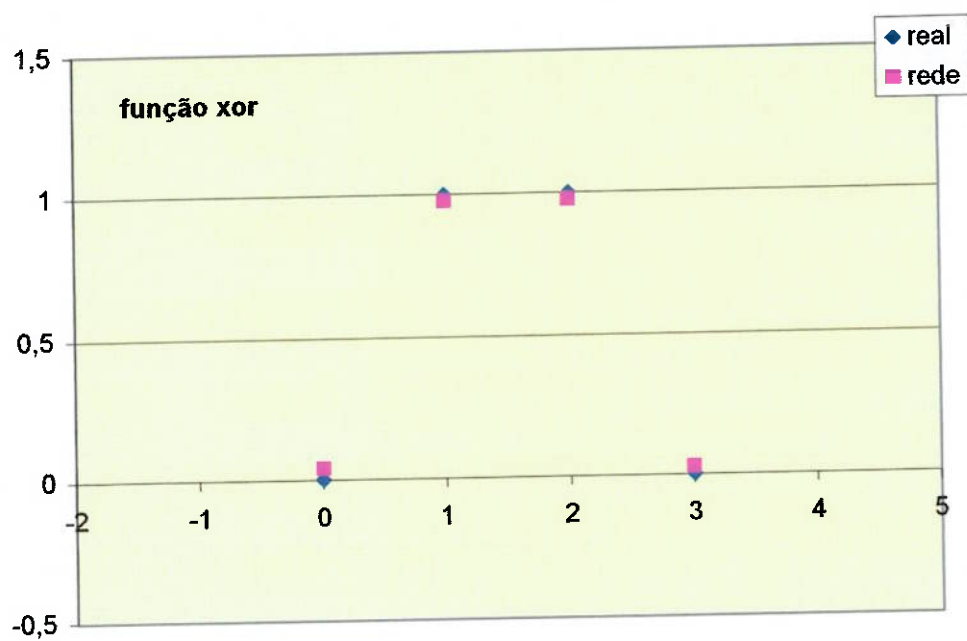


Figura 3.3. Resultados para função xor

4 – TRATAMENTO DOS DADOS

4.1 – Descrição da Instalação Predial

A Reitoria da USP, objeto de nosso estudo, localiza-se no *Campus* Cidade Universitária, na Rua da Reitoria, número 109. Foi construído da década de 70 e abriga os gabinetes do reitor, vice-reitor e pró-reitorias, bem como diversos departamentos administrativos de apoio às atividades universitárias.

O conjunto é constituído por dois prédios, com seis andares cada um. Sua área construída é de 3 mil metros quadrados. Cerca de novecentas pessoas trabalham diariamente no conjunto. Os equipamentos de condicionamento de ar são do tipo *split* e janela.

4.2 – Perfil Climático e de Consumo

Foram levantados os dados climáticos e de consumo de agosto de 2003 até março de 2005, pelo Instituto Astronômico e Geofísico da USP (IAG-USP). Dos dados climáticos, temos disponíveis temperatura (T , °C), umidade relativa (U , %), radiação global (R_{glo} , W/m^2) e radiação difusa (R_{dif} , W/m^2), todos eles para a região próxima ao prédio da reitoria, aquisitados a cada cinco minutos.

Quanto ao consumo energético, temos a potência elétrica consumida (c , kW), a cada quinze minutos. O consumo energético diário (C , kWh), é portanto a soma das medições intermediárias, dividida por quatro.

A partir dos valores médios para cada intervalo do dia, levantou-se o perfil climático e de consumo para a instalação. A Fig. 4.1 traz o perfil de consumo, a Fig.

4.2 traz o perfil climático para temperatura e umidade e a Fig. 4.3 traz o perfil climático para radiação.

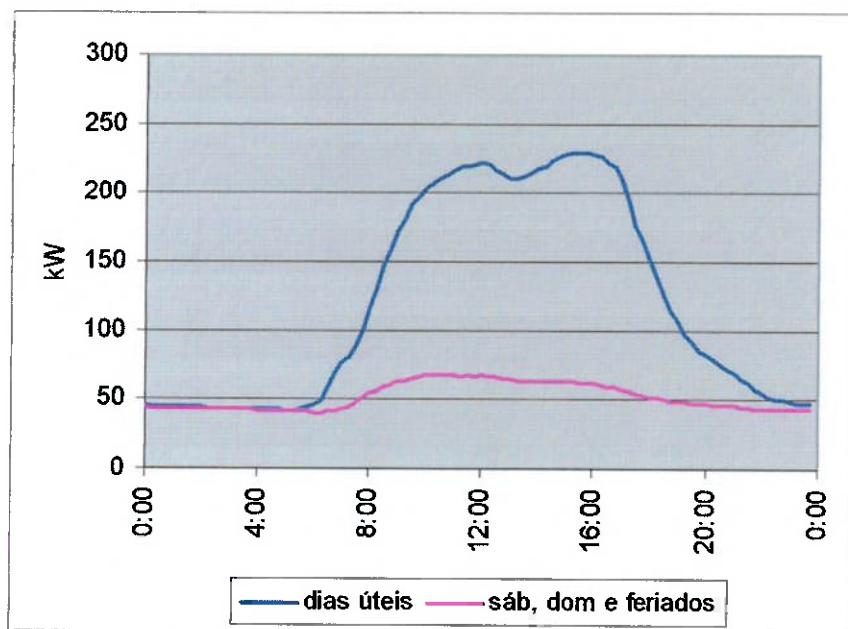


Figura 4.1. Perfil de Consumo

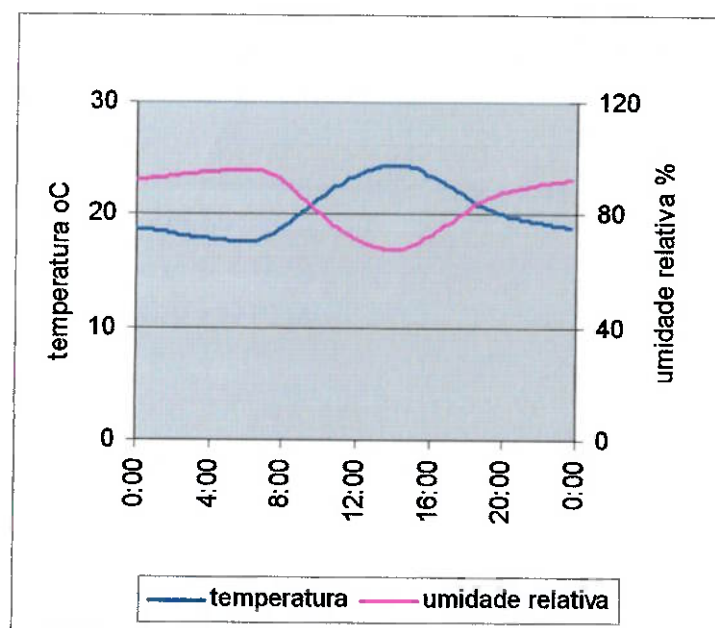


Figura 4.2. Perfil Climático – Temperatura e Umidade

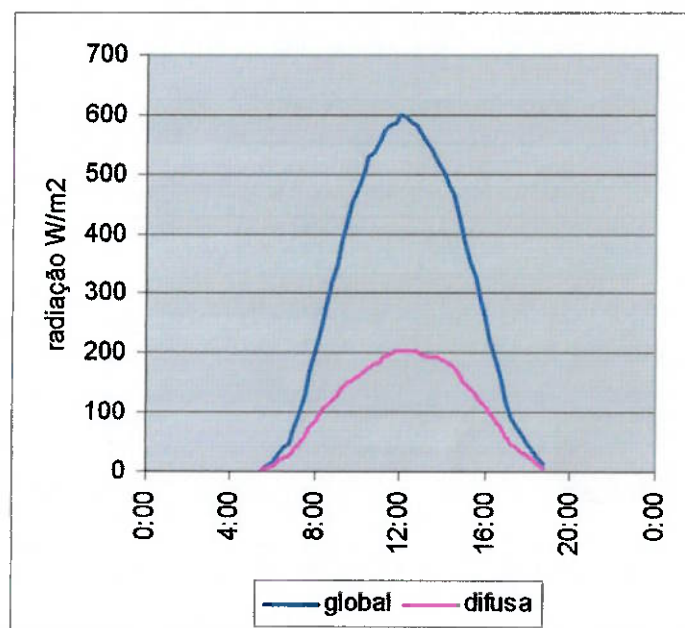


Figura 4.3. Perfil climático – Radiação

Notamos uma grande variação entre o consumo energético de um dia útil para aquele de um final de semana. Isso corresponde ao esperado, pois nos finais de semana o uso do prédio, e por consequência, de seus equipamentos, é consideravelmente reduzido. Isso nos motivou a criar dois testes distintos para a rede neural: no primeiro deles iremos alimentar a rede também com a informação do dia útil ou não e no segundo criar redes distintas para os dias úteis e outra para os finais de semana.

Quanto ao perfil climático, notamos que o da umidade acompanha o perfil da temperatura, sendo uma curva o inverso da outra. Da mesma forma as curvas para radiação apresentam algo grau de similaridade. Dessa forma, consideramos como primeiro teste para a rede neural alimentá-la somente com dados de temperatura e consumo (além da informação do dia útil), e somente num teste posterior incluir os demais dados climáticos.

4.3 – Tratamento dos Dados

A massa total dos dados foi dividida em dois grupos para uso na rede neural:

- a) grupo para treino, composto pelo dados de agosto de 2003 até dezembro de 2004;
- b) grupo para validação, composto pelos dados de janeiro a março de 2005.

O primeiro grupo será usado para alimentar a rede, informando a ela tanto os valores climáticos, quanto os dados de consumo. Uma vez treinada a rede, o segundo grupo será usado para validar a rede montada, alimentando-a somente com os dados climáticos e comparando os dados de consumo que ela fornecer, com os aferidos.

Notamos ainda a existência de “buracos” no conjunto de dados coletados. Essas medições que deixaram de ser feitas aparecem tanto para o consumo, quanto para os dados climáticos. Esses casos manifestam-se desde uma única medição faltante no dia, até situações nas quais não há medição alguma para um dia inteiro.

Para contornar essa situação, pois usando os dados dessa forma, a rede iria assumir os buracos como valor zero, optamos por duas ações: eliminar os registros, e completar os registros.

No primeiro caso, omitimos os dias nos quais mais de 10% das medições estivessem faltando. Já no segundo caso, as medições faltantes foram preenchidas com o valor médio do mês, para o respectivo horário.

Por fim, para os dois casos acima, foram levantados os valores de consumo diários, e os valores mínimos, média e máximos para os dados climáticos. Esses dados consolidados foram usados para treinar e validar a rede.

5 – RESULTADOS

5.1 – Previsão de Consumo a partir da Temperatura

Separamos os dados em dois grupos. O primeiro deles, usado para treino, é composto pelas medições de agosto de 2003 até dezembro de 2004. O segundo grupo, usado para validação, é composto pelas medições de janeiro a março de 2005.

Os dados climáticos coletados foram agrupados por dia, e como entrada, considerados somente a temperatura mínima e máxima (T_{\min} , T_{\max} , °C) de cada dia, e como saída, o consumo total (C, kWh) daquele dia.

Foram consideradas três redes para a análise: a primeira formada por todos os dias do período, a segunda somente com os dias úteis e a terceira contendo somente os finais de semana e feriados.

Por fim, para os dados de entrada, usamos um conjunto do qual foram descartados os dias com menos de 90% das medições feitas. A Tab. 5.1 traz os parâmetros usados para esses casos. Os resultados estão resumidos na Tab. 5.2 e apresentados nas Fig. 5.1, 5.3 e 5.5 para a etapa de treino e nas Fig. 5.2, 5.4 e 5.6 para a etapa de validação.

Tabela 5.1. Parâmetros para temperatura – consumo

Parâmetros	todos os dias	dias úteis	finais de semana e feriados
Variáveis de Entrada	3 (T_{\min} , T_{\max} , dutil)	2 (T_{\min} , T_{\max})	2 (T_{\min} , T_{\max})
Variáveis de Saída	1 (C)	1 (C)	1 (C)
Dados para treino	335	200	135

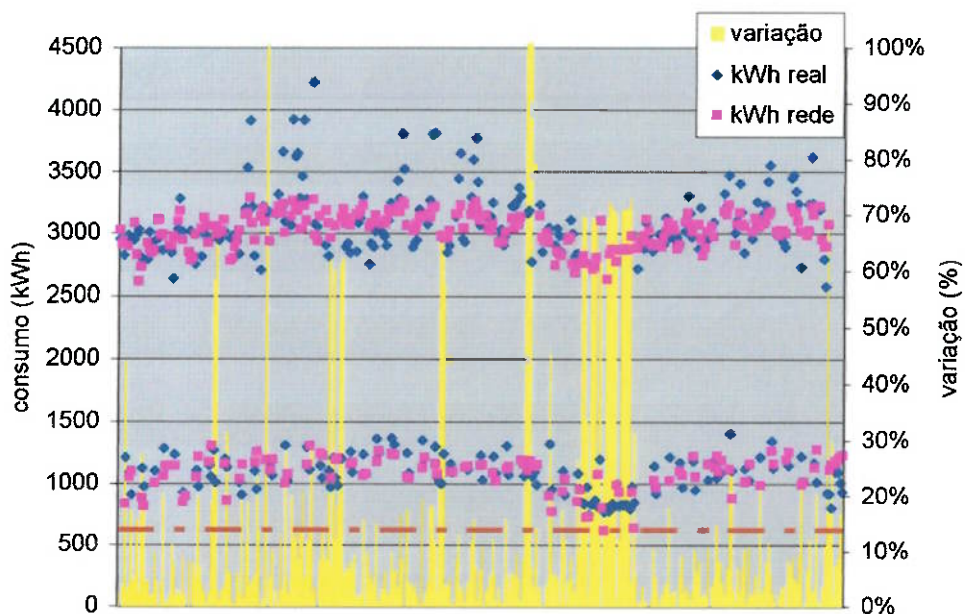
(continua...)

Tabela 5.1 (cont). Parâmetros para temperatura – consumo

Parâmetros	todos os dias	dias úteis	finais de semana e feriados
Dados para validação	79	53	26
Camadas	3	3	3
Neurônios na camada de entrada	3	2	2
Neurônios na segunda camada	18	18	18
Neurônios na camada de saída	1	1	1
Taxa de aprendizado λ	0,7	0,7	0,7
Função de ativação	Linear	Linear	Linear

Tabela 5.2. Resultados para temperatura – consumo

Resultados	todos os dias	dias úteis	finais de semana e feriados
Variação média para treino	13,9%	4,5%	9,8%
Variação média para validação	21%	10,8%	10,5%

**Figura 5.1.** Rede temperatura – consumo; etapa de treino; entradas T_{\min} , T_{\max} , d_{util} ; saída consumo; todos os dias

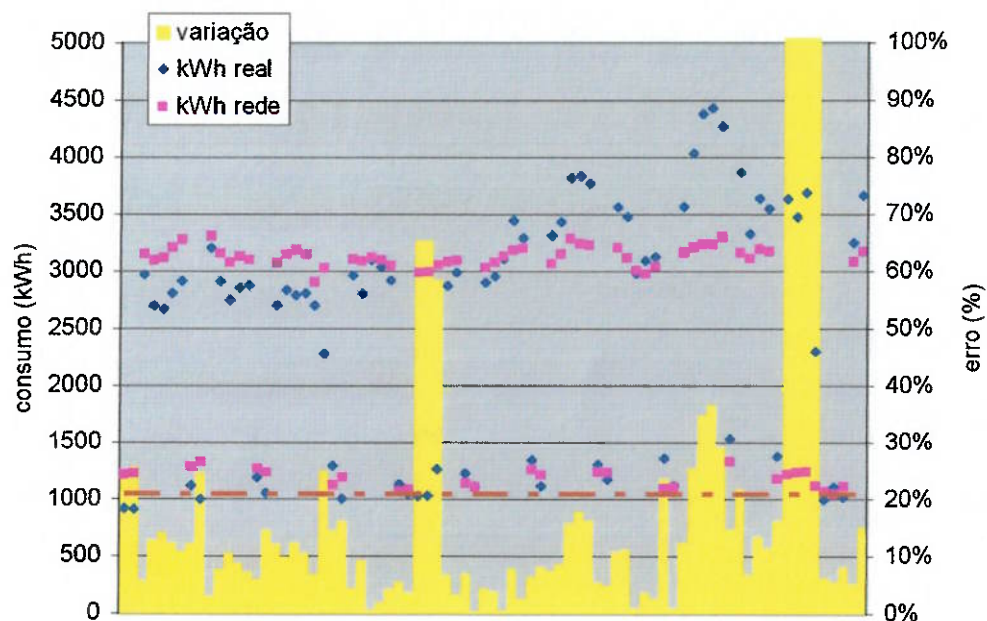


Figura 5.2. Rede temperatura – consumo; etapa de validação;
entradas T_{\min} , T_{\max} , dutil; saída consumo; todos os dias

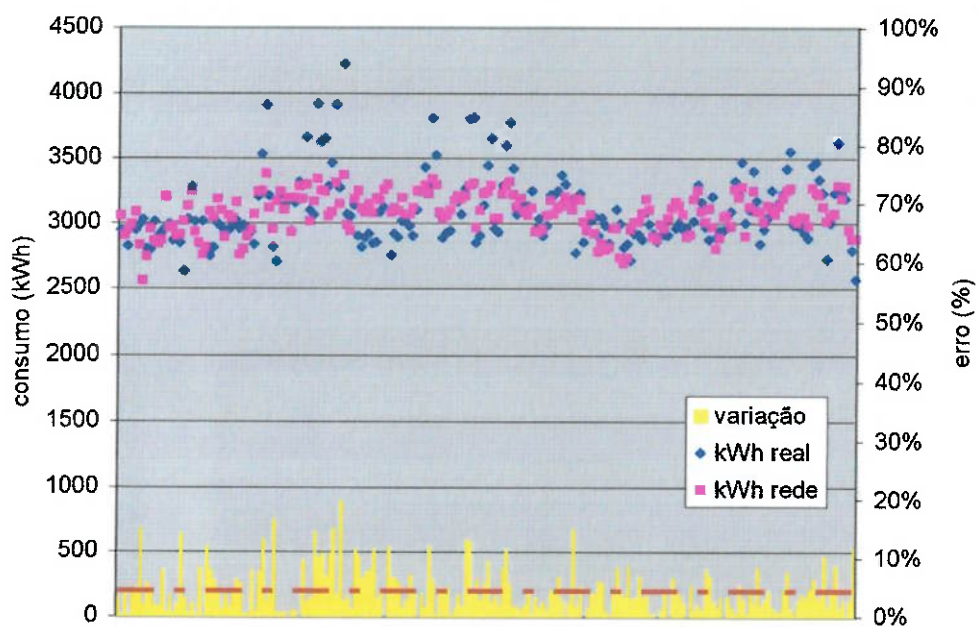


Figura 5.3. Rede temperatura – consumo; etapa de treino;
entradas T_{\min} , T_{\max} ; saída consumo; dias úteis

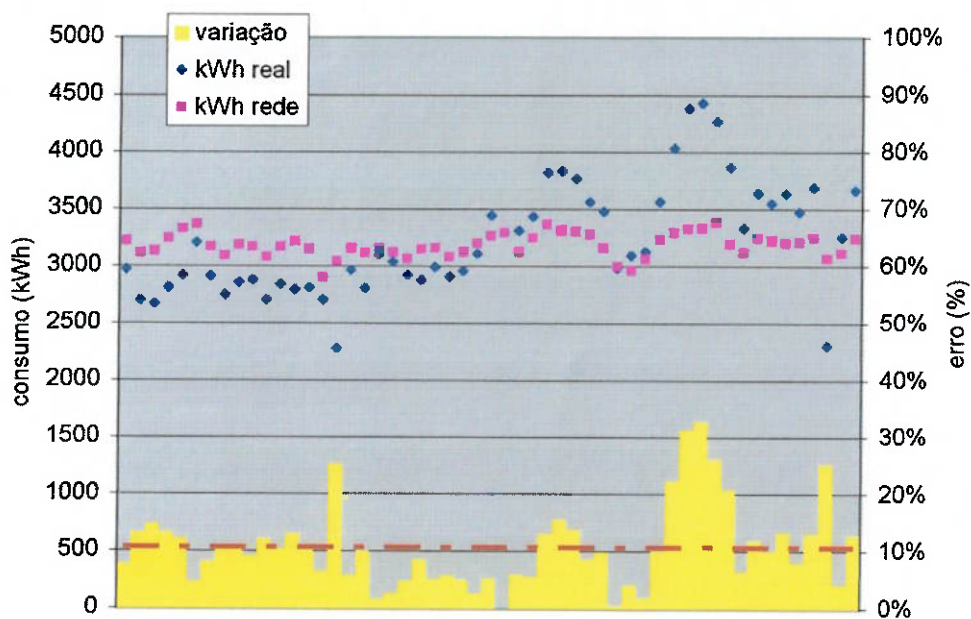


Figura 5.4. Rede temperatura – consumo; etapa de validação;
entradas T_{\min} , T_{\max} ; saída consumo; dias úteis

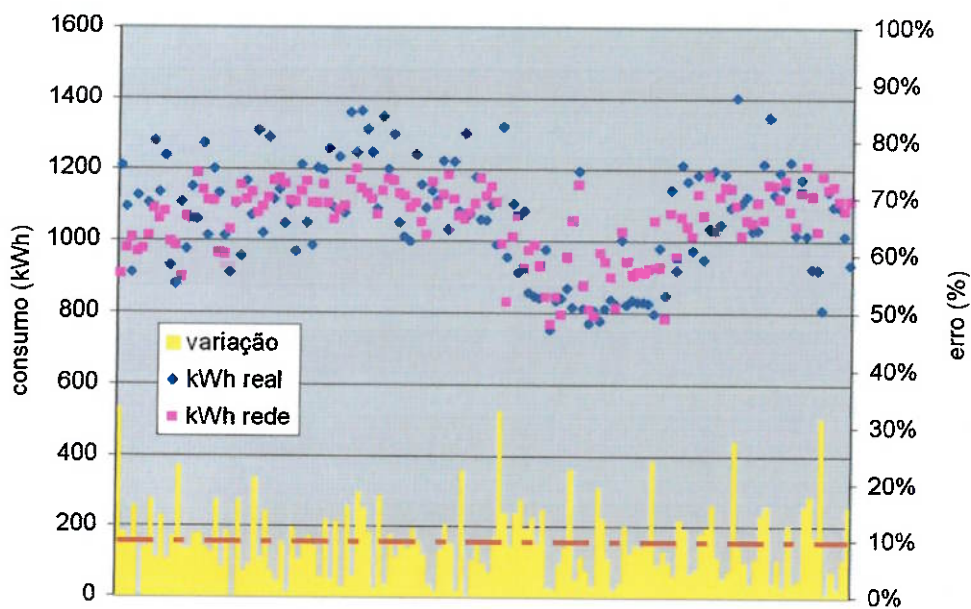


Figura 5.5. Rede temperatura – consumo; etapa de treino;
entradas T_{\min} , T_{\max} ; saída consumo; finais de semana e feriados

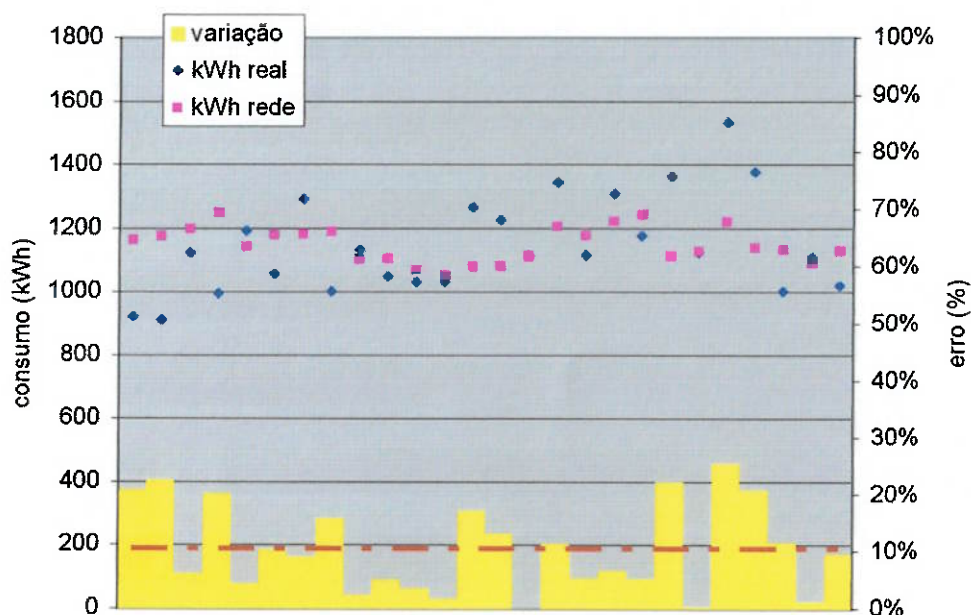


Figura 5.6. Rede temperatura – consumo; etapa de validação; entradas T_{\min} , T_{\max} ; saída consumo; finais de semana e feriados

A rede para todos os dias, apesar de receber uma entrada indicando o tipo do dia, apresentou a maior variação entre valor fornecido para o valor real. Separando os dias úteis dos demais em duas redes distintas, conseguimos melhorar a previsão feita pela rede. Fizemos os mesmos testes em redes com cinco camadas, usando oito neurônios em cada camada oculta. Os resultados obtidos foram similares aos apresentados acima.

A rede que considera somente os dias úteis apresentou os melhores resultados, tanto na etapa de treino, quanto na etapa de validação. Isso leva a crer que o consumo nesses dias sofre menos influência de outros fatores que não os climáticos do que nos finais de semana e feriados.

Nas análises anteriores, descartamos os dias com menos de 90% das medições feitas. Fizemos um outro conjunto de análises, desta vez adicionando a média mensal no lugar das medições faltantes. A Tab. 5.3 traz os parâmetros usados para esses casos. Os resultados estão resumidos na Tab. 5.4 e apresentados nas Fig.

5.7, 5.9 e 5.11 para a etapa de treino e nas Fig. 5.8, 5.10 e 5.12 para a etapa de validação.

Tabela 5.3. Parâmetros para temperatura, umidade, radiação – consumo

Parâmetros	todos os dias	dias úteis	finais de semana e feriados
Variáveis de Entrada	3 (T_{\min} , T_{\max} , dutil)	2 (T_{\min} , T_{\max})	2 (T_{\min} , T_{\max})
Variáveis de Saída	1 (C)	1 (C)	1 (C)
Dados para treino	484	317	167
Dados para validação	88	56	32
Camadas	3	3	3
Neurônios na camada de entrada	3	2	2
Neurônios na segunda camada	18	18	18
Neurônios na camada de saída	1	1	1
Taxa de aprendizado λ	0,7	0,7	0,7
Função de ativação	Linear	Linear	Linear

Tabela 5.4. Resultados para temperatura – consumo

Resultados	todos os dias	dias úteis	finais de semana e feriados
Variação média para treino	24,6%	26,4%	43,2%
Variação média para validação	22,1%	13,9%	36,3%

As variações entre os resultados fornecidos pela rede, com os esperados aumentaram consideravelmente, para todos os casos. Note, entretanto, que as variações para as etapas de validação são menores que as variações para a etapa de treino, enquanto na análise anterior ocorreu o inverso. Como há menos falhas nas medições para os conjuntos de validação, do que para os de treino (por se tratar de três meses em um, contra dezessete no outro), leva-nos a crer que preencher os dados faltantes com as médias mensais acabou por descaracterizar o perfil de consumo e temperatura formados, gerando as diferenças notadas.

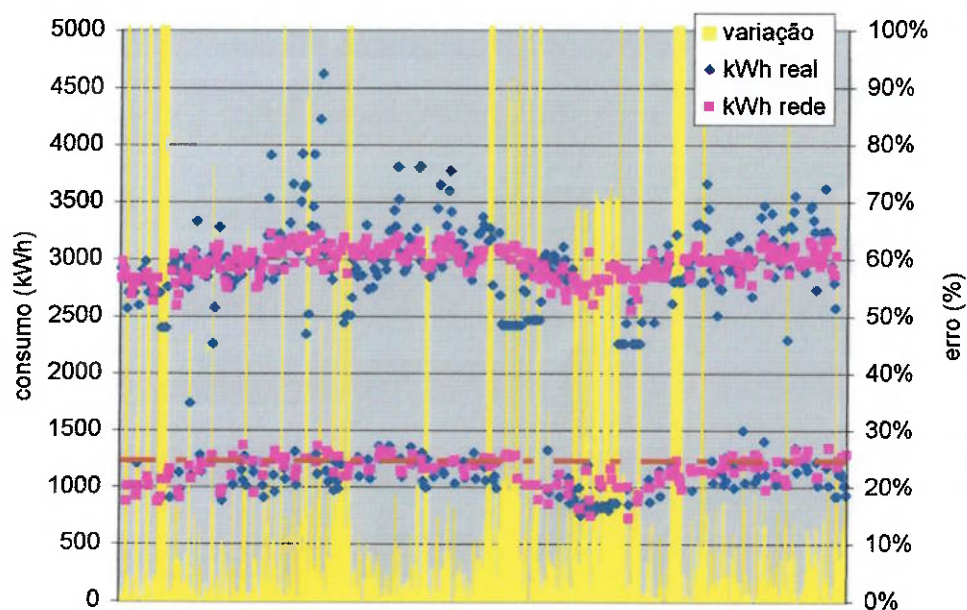


Figura 5.7. Rede temperatura – consumo; etapa de treino;
entradas T_{\min} , T_{\max} , dutil; saída consumo; todos os dias

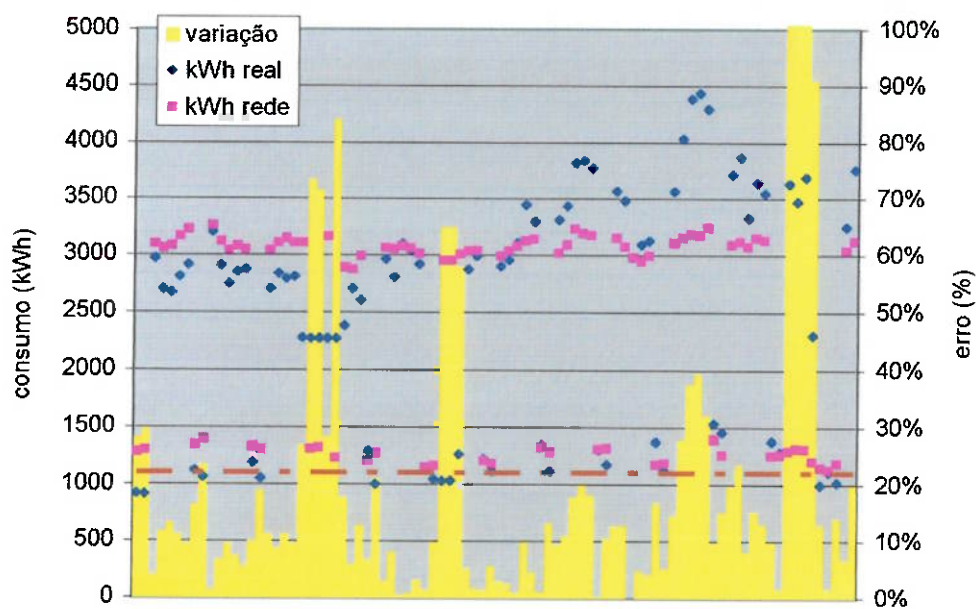


Figura 5.8. Rede temperatura – consumo; etapa de validação;
entradas T_{\min} , T_{\max} , dutil; saída consumo; todos os dias

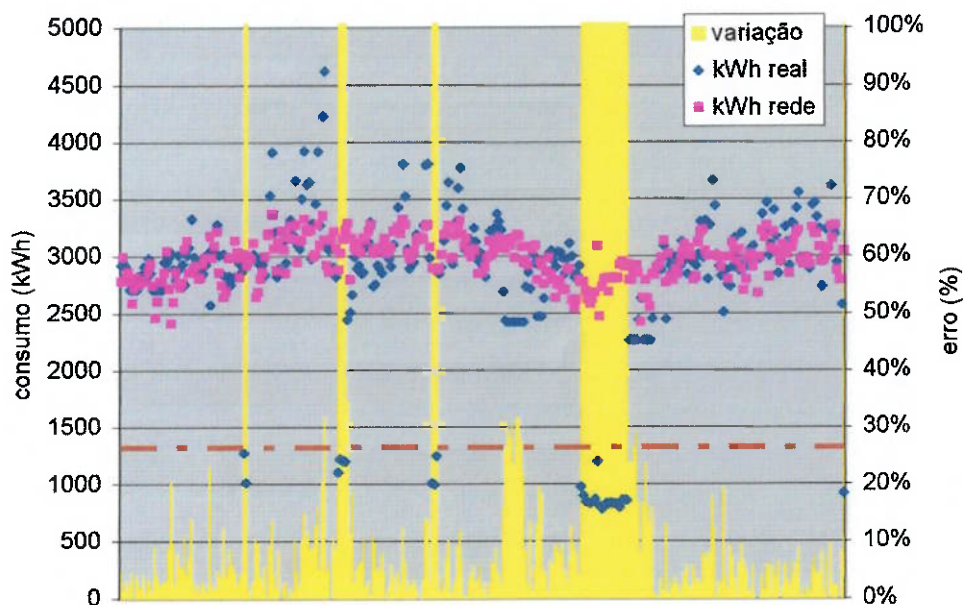


Figura 5.9. Rede temperatura – consumo; etapa de treino;
entradas T_{\min} , T_{\max} ; saída consumo; dias úteis

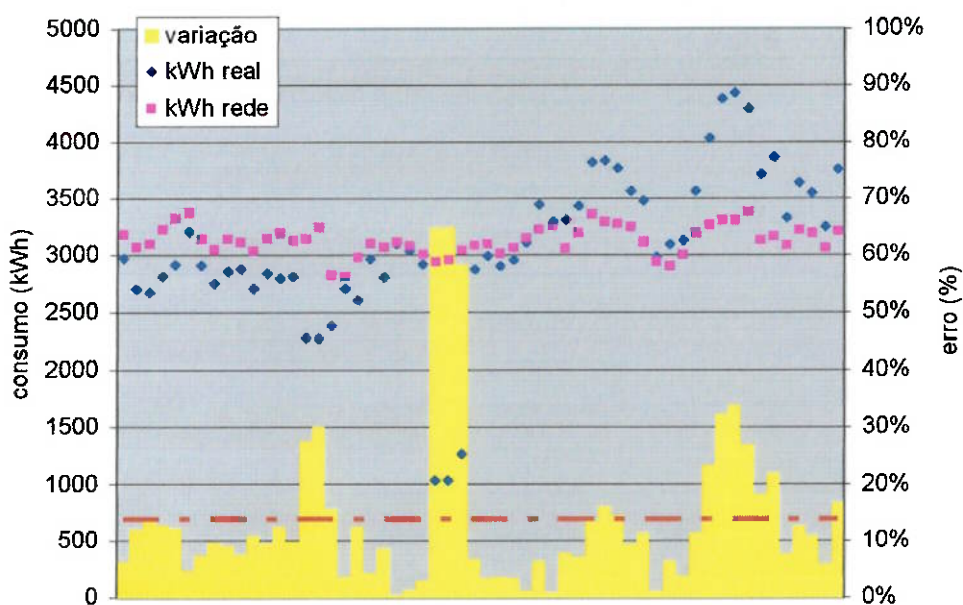


Figura 5.10. Rede temperatura – consumo; etapa de validação;
entradas T_{\min} , T_{\max} ; saída consumo; dias úteis

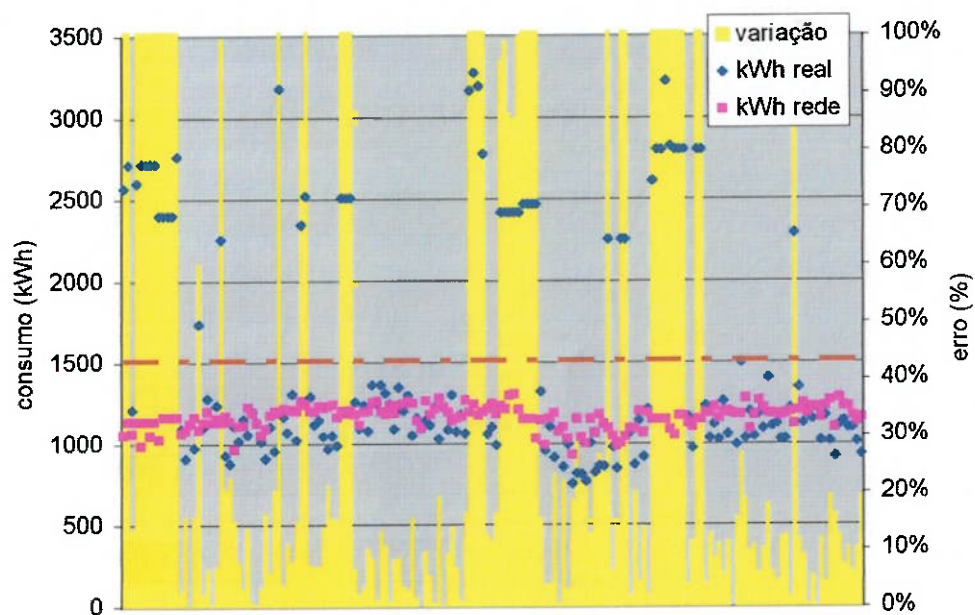


Figura 5.11. Rede temperatura – consumo; etapa de treino;
entradas T_{\min} , T_{\max} ; saída consumo; finais de semana e feriados

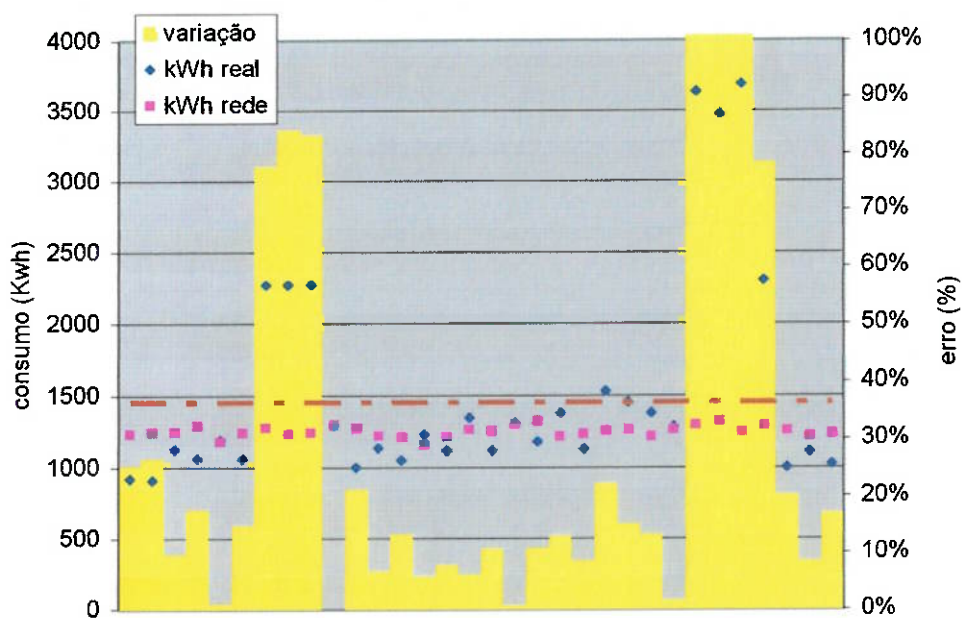


Figura 5.12. Rede temperatura – consumo; etapa de validação;
entradas T_{\min} , T_{\max} ; saída consumo; finais de semana e feriados

5.2 – Previsão de consumo a partir da Temperatura, Umidade e Radiação

Analogamente ao que foi feito para temperatura somente, usamos o grupo de dados para treino, de agosto de 2003 até dezembro de 2004, e o grupo para validação de janeiro a março de 2005.

Os dados foram agrupados por dia, mas desta vez, ao invés de usar os valores mínimos e máximos das variáveis como entrada, usamos seus valores médios. Usamos, portanto: temperatura média (T , °C), umidade relativa média (U , %), radiação global média (R_{glo} , W/m^2) e radiação difusa média (R_{dif} , W/m^2). Como saída, usamos consumo total (C , kWh).

Novamente, consideramos três redes para esta análise: a primeira formada por todos os dias do período, a segunda somente com os dias úteis e a terceira contendo somente os finais de semana e feriados. Descartamos da análise todos os dias com menos de 90% das medições feitas.

A Tab. 5.4 traz os parâmetros usados para esses casos. Os resultados estão resumidos na Tab. 5.5 e apresentados nas Fig. 5.13, 5.15 e 5.17 para a etapa de treino e nas Fig. 5.14, 5.16 e 5.18 para a etapa de validação.

Tabela 5.4. Parâmetros para temperatura, umidade, radiação – consumo

Parâmetros	todos os dias	dias úteis	finais de semana e feriados
Variáveis de Entrada	5 (T , U , R_{glo} , R_{dif} , dutil)	4 (T , U , R_{glo} , R_{dif})	4 (T , U , R_{glo} , R_{dif})
Variáveis de Saída	1 (C)	1 (C)	1 (C)
Dados para treino	227	142	85
Dados para validação	59	39	20
Camadas	3	3	3
Neurônios na camada de entrada	5	4	4
Neurônios na segunda camada	21	21	21
Neurônios na camada de saída	1	1	1

(continua...)

Tabela 5.4 (cont). Parâmetros para temperatura, umidade, radiação – consumo

Parâmetros	todos os dias	dias úteis	finais de semana e feriados
Taxa de aprendizado λ	0,7	0,7	0,7
Função de ativação	Linear	Linear	Linear

Tabela 5.5. Resultados para temperatura, umidade, radiação – consumo

Resultados	todos os dias	dias úteis	finais de semana e feriados
Variação média para treino	10,5%	4,9%	9,4%
Variação média para validação	16,5%	9,5%	9,7%

Da mesma forma como notado para a rede temperatura – consumo, separar os dias úteis dos demais traz uma razoável melhora nas estimativas apresentadas pela rede. A variação encontrada entre os valores fornecidos e os esperados teve pouca melhora com a introdução da umidade e radiação. Isso nos leva a concluir que esses fatores não contribuem, no consumo total, tão significativamente quanto a temperatura.

Fizemos ainda uma última análise para a rede temperatura, umidade, radiação – consumo, incluindo as médias mensais no lugar das medições faltantes. Os parâmetros usados estão apresentados na Tab. 5.6 e os resultados obtidos estão apresentados na Tab. 5.7. Por brevidade, omitimos do presente trabalho os gráficos levantados.

Tabela 5.6. Parâmetros para temperatura, umidade, radiação – consumo

Parâmetros	todos os dias	dias úteis	finais de semana e feriados
Variáveis de Entrada	5 (T, U, R_{glo} , R_{dif} , dutil)	4 (T, U, R_{glo} , R_{dif})	4 (T, U, R_{glo} , R_{dif})
Variáveis de Saída	1 (C)	1 (C)	1 (C)
Dados para treino	484	317	167
Dados para validação	88	56	32

(continua...)

Tabela 5.6 (cont). Parâmetros para temperatura, umidade, radiação – consumo

Parâmetros	todos os dias	dias úteis	finais de semana e feriados
Camadas	3	3	3
Neurônios na camada de entrada	5	4	4
Neurônios na segunda camada	21	21	21
Neurônios na camada de saída	1	1	1
Taxa de aprendizado λ	0,7	0,7	0,7
Função de ativação	Linear	Linear	Linear

Tabela 5.7. Resultados para temperatura, umidade, radiação – consumo

Resultados	todos os dias	dias úteis	finais de semana e feriados
Varição média para treino	23,9%	26,1%	44,3%
Varição média para validação	23,3%	14,4%	38,1%

Similar ao observado para a rede temperatura – consumo, a inclusão das médias no lugar das medições faltantes acabou prejudicando o aprendizado da rede.

Por fim, note que, tanto para a rede temperatura – consumo, quanto para a rede temperatura, umidade, radiação – consumo, para o caso no qual excluímos as medições faltantes, a rede alimentada somente com dias úteis apresentou menor variação que as duas outras. Isso tanto na etapa de treino, quanto na etapa de validação.

Esse fato nos leva supor que não há um perfil de consumo padrão para finais de semana e feriados. Como, nesses dias, o consumo é consideravelmente inferior ao dos demais, qualquer evento que ocorra no prédio da reitoria nesses dias causa variações consideráveis no consumo total da instalação. Essas variações não são levadas ao conhecimento da rede, que só recebe parâmetros climáticos.

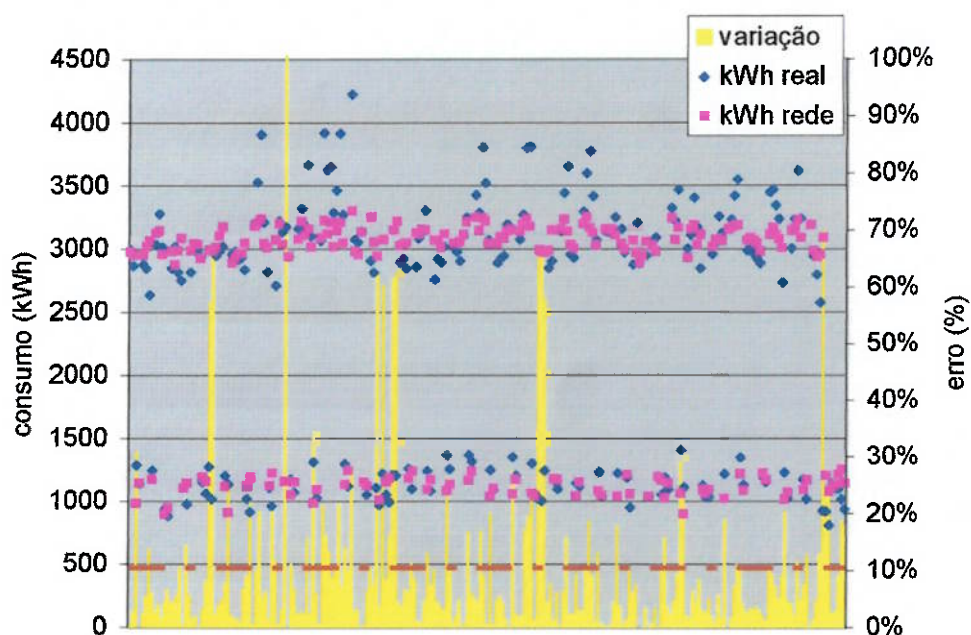


Figura 5.13. Rede temperatura, umidade, radiação – consumo; etapa de treino; entradas T, U, R_{dif} , R_{glo} , dutil; saída consumo; todos os dias

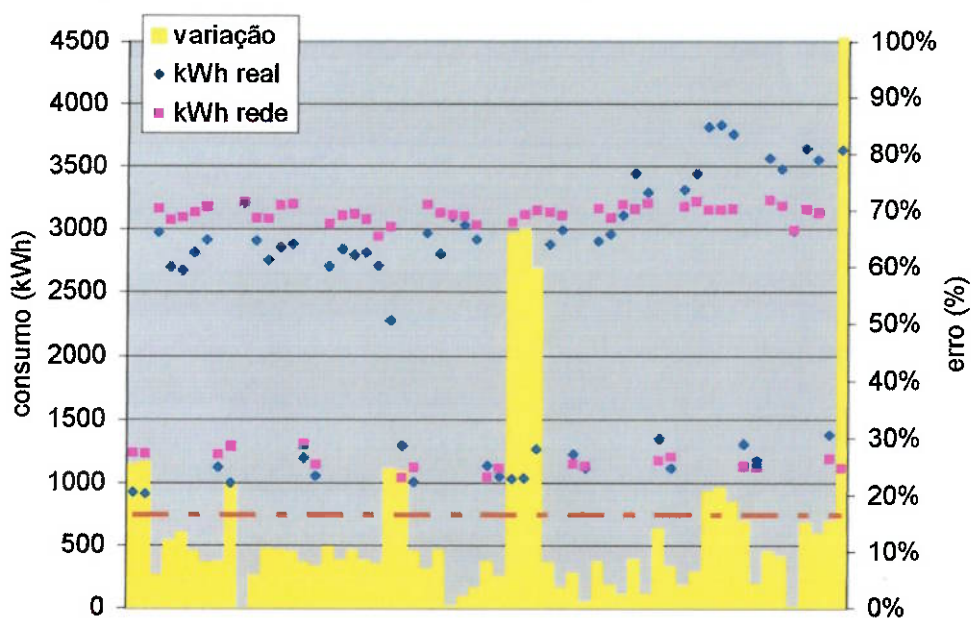


Figura 5.14. Rede temperatura, umidade, radiação – consumo; etapa de validação; entradas T, U, R_{dif} , R_{glo} , dutil; saída consumo; todos os dias

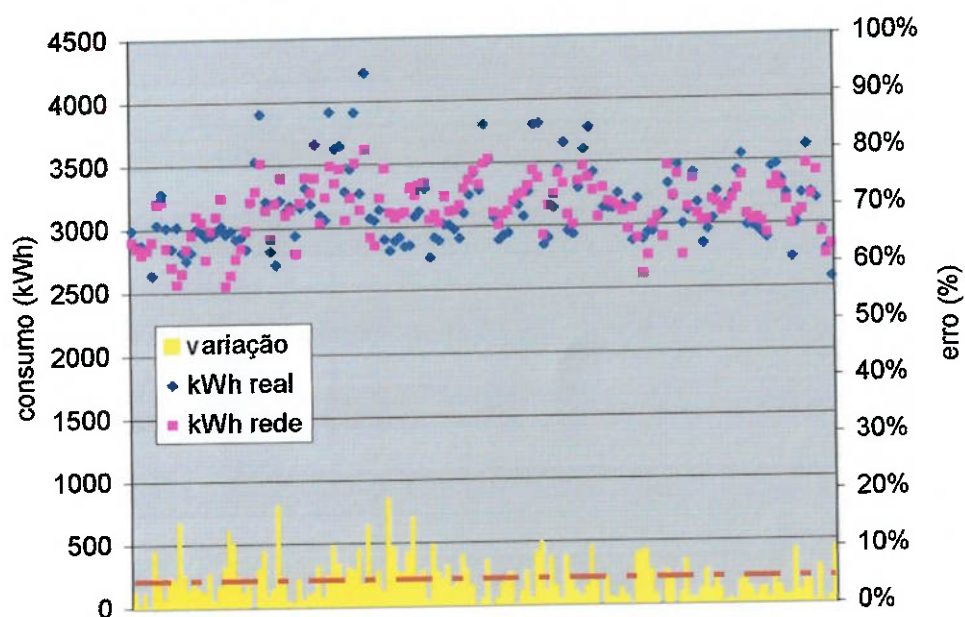


Figura 5.15. Rede temperatura, umidade, radiação – consumo; etapa de treino;
entradas T, U, R_{dif} , R_{glo} ; saída consumo; dias úteis

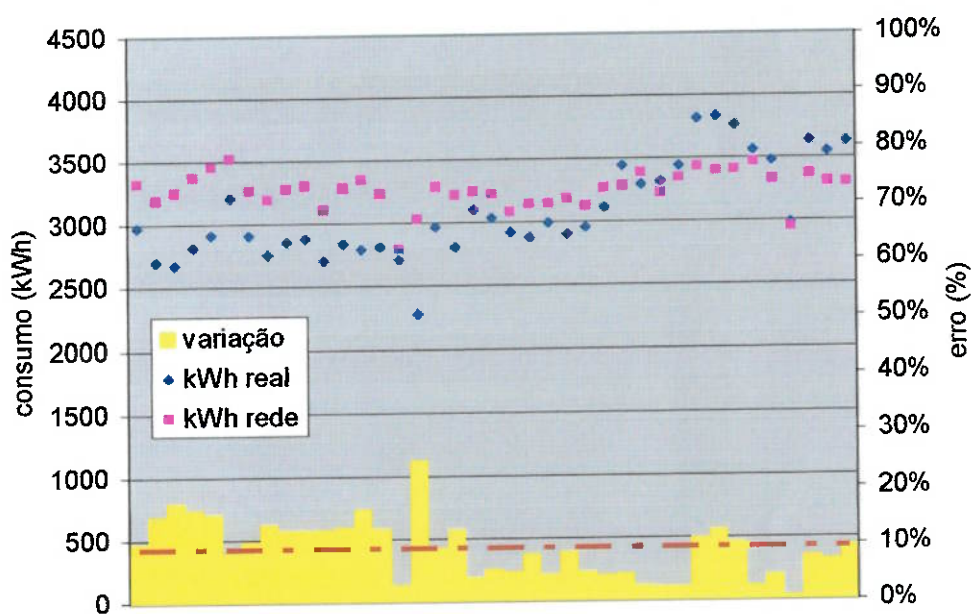


Figura 5.16. Rede temperatura, umidade, radiação – consumo; etapa de validação;
entradas T, U, R_{dif} , R_{glo} ; saída consumo; dias úteis

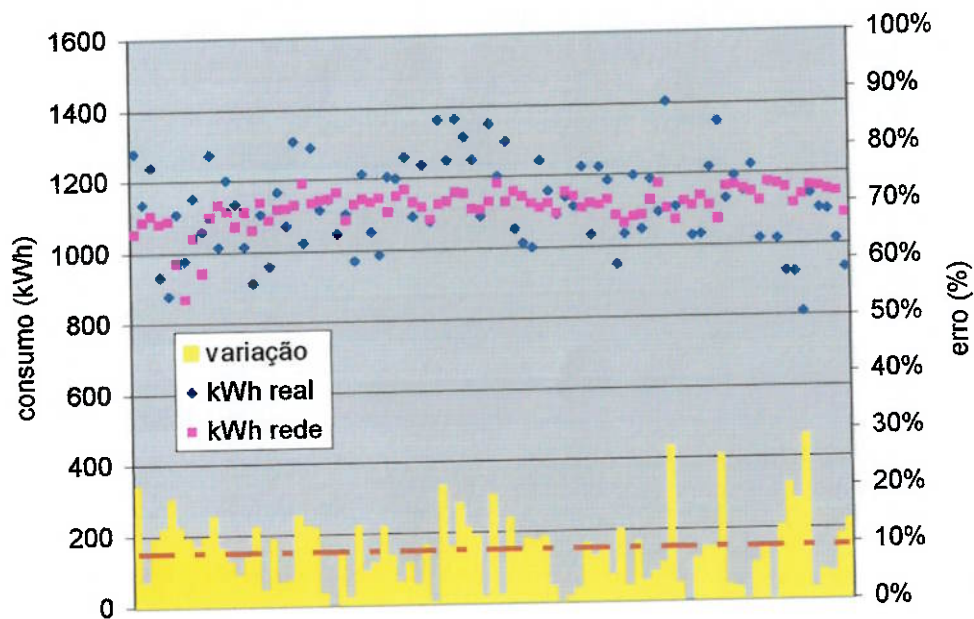


Figura 5.17. Rede temperatura, umidade, radiação – consumo; etapa de treino;
entradas T , U , R_{dif} , R_{glo} ; saída consumo; finais de semana e feriados

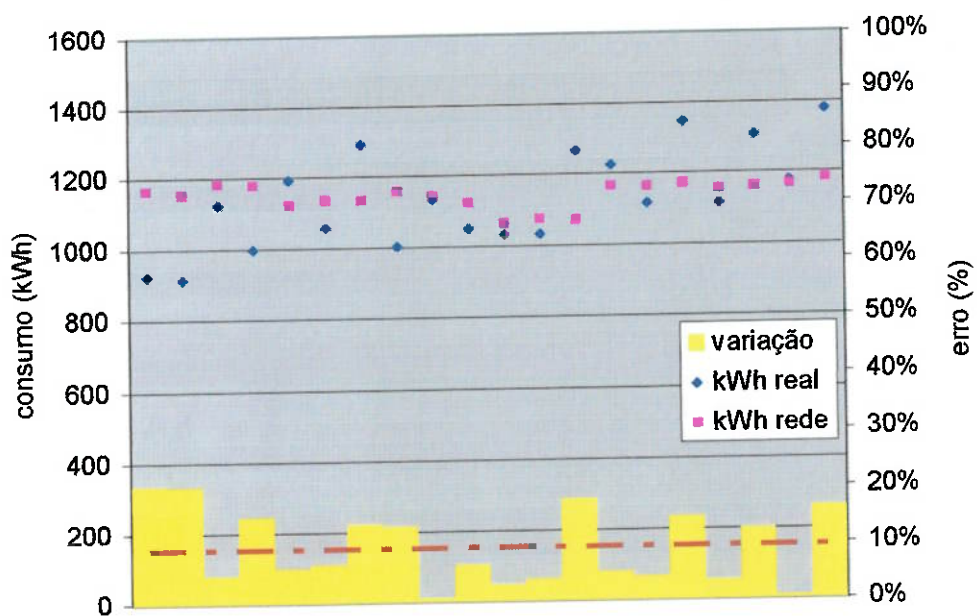


Figura 5.18. Rede temperatura, umidade, radiação – consumo; etapa de validação;
entradas T , U , R_{dif} , R_{glo} ; saída consumo; finais de semana e feriados

6 – CONCLUSÕES

As redes neurais montadas apresentaram previsões com variação, no melhor caso, da ordem de 10% para os valores esperados. Essa variação não chega a impossibilitar o uso do programa desenvolvido. É necessário, entretanto, sempre ter em mente que poderá existir essa diferença entre os resultados fornecidos pela rede e o consumo real da edificação.

Por notarmos pouca diferença entre os resultados das redes que tiveram como entrada somente a temperatura daquelas que trabalharam com todas as variáveis climatológicas disponíveis, podemos concluir que as demais variáveis não têm influência tão significativa no consumo energético da edificação. Talvez pois, muito provavelmente, variações nelas acabem se manifestando em variações na temperatura também.

Como as redes apresentaram variações nos resultados finais mesmo na etapa de treino, pode-se concluir que o aprendizado delas apresentou alguma deficiência. Essa deficiência pode ser explicada pela não consideração de algum item relevante para o processo (como, por exemplo, quantidade de pessoas na edificação), ou então pela hipótese de que o consumo dos equipamentos não relacionados ao condicionamento de ar teria pouca influência no consumo final.

Por fim, como notamos o fato da rede proposta não ter sido capaz de aprender uma função quadrática, é possível que o modelo escolhido não seja o mais adequado para nosso objetivo, restando, portanto, procurar modelos mais avançados que possam fornecer melhores resultados.

BIBLIOGRAFIA

- Haykin, S., 1994, **Neural networks – a comprehensive foundation**, MacMillan Publishing Company, New York, EUA.
- Freeman, J., Skapura, D., 1991, **Neural networks – algorithms, applications, and programming techniques**, Addison-Wesley, Boston, EUA.
- Nissen, S., 2005, **Neural Networks Made Simple**, Software Developer's Journal, Fev. 2005, Varsóvia, Polônia.
- Nissen, S., 2003, **Implementation of a Fast Artificial Neural Network Library**, Department of Computer Science University of Copenhagen, Copenhagen, Dinamarca.
- Farley, S., 2001, **Neuralyst – User's Guide**, Cheshire Engineering Corporation, Pasadena, EUA.
- Sabbatini, R., 2003, **Neurônios e Sinapses – A História de Sua Descoberta**, Revista Cérebro & Mente, maio-agosto 2003, São Paulo, Brasil.
- AQUINO, R., 2005. **Gestão de manutenção de condicionadores de ar do tipo janela e split**, trabalho de graduação apresentado ao Departamento de Engenharia Mecânica, Escola Politécnica, Universidade de São Paulo, São Paulo, Brasil.
- Wikipedia. **Neural network**. Disponível em:
http://en.wikipedia.org/wiki/Neural_network
- HABRA, A. **Neural Networks - An Introduction**. Disponível em
<http://www.tek271.com/articles/neuralNet/IntoToNeuralNets.html>
- SARLE, W. **comp.ai.neural-nets FAQ**. Disponível em
<ftp://ftp.sas.com/pub/neural/FAQ.html>

