

**UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**AVALIAÇÃO DE CONFIGURAÇÕES DE REDES NEURAS
ARTIFICIAIS PARA PREVISÃO DO CONSUMO DE ENERGIA DE
SISTEMAS DE CLIMATIZAÇÃO**

Esteban Patricio Manuel Fernández Arancibia

**SÃO PAULO
2008**

**UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**AVALIAÇÃO DE CONFIGURAÇÕES DE REDE NEURAL PARA
PREVISÃO DO CONSUMO DE ENERGIA DE SISTEMAS DE
CLIMATIZAÇÃO**

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

Esteban Patricio Manuel Fernández Arancibia

Orientador: Flávio Augusto Sanzovo Fiorelli

Área de Concentração: Engenharia Mecânica

São Paulo

2008

FICHA CATALOGRÁFICA

Arancibia, Esteban Patricio Manuel Fernández
Avaliação de Configurações de Redes Neurais Artificiais para
Previsão do Consumo de Energia de Sistemas de Climatização/
E.P.M.F. Arancibia. – São Paulo, 2008.
57 p.

Trabalho de Formatura - Escola Politécnica da Universidade
de São Paulo. Departamento de Engenharia Mecânica.

1. Redes neurais 2. Consumo de energia elétrica 3. Ar condicio-
nado I. Universidade de São Paulo. Escola Politécnica.
Departamento de Engenharia Mecânica II.t.

RESUMO

Este trabalho apresenta um estudo comparativo da utilização de diferentes configurações de rede neural artificial em um programa de simulação para previsão do consumo diário de energia elétrica de uma edificação comercial. Tomando-se como referência uma rede mais simples, do tipo “*feed-forward*”, foram implementadas configurações de rede mais complexas dos tipos “*feed-back*” e mapas auto-organizáveis. Como estudo de caso foram utilizados os dados de consumo diário do prédio da Reitoria da USP, localizado na Cidade Universitária “Armando de Salles Oliveira” em São Paulo. A hipótese principal assumida é que as variações no consumo energético decorrem predominantemente dos equipamentos associados ao condicionamento de ar, e por isso foram utilizados como parâmetros de entrada das redes as condições climáticas locais do prédio (temperatura, umidade e radiação), obtidas junto obtidas junto ao Instituto Astronômico e Geofísico da USP (IAG-USP). Depois de implementar cada uma das configurações, treinando-as com os dados do ano 2004, conseguiu-se prever o consumo energético dos três primeiros meses do ano 2005, obtendo-se melhorias de até 10%, no melhor dos casos, nas previsões feitas com as configurações de rede mais complexas em relação à rede “*feed-forward*”.

ABSTRACT

This work presents a comparative study on the use of different artificial neural network configurations in a simulation program to predict the daily electric power consumption for a commercial building. Taking as reference a simpler network, the “feed-forward” model, it was implemented more complex network arrangements, like the “feedback” and self-organizing maps models. Daily consumption data of the USP Administration Building at Cidade Universitaria “Armando de Salles Oliveira” in São Paulo was used as study case. The main hypotheses assumed is that the variations in energy consumption is primarily due to air-conditioning equipment, and therefore it was used as input parameters for the networks the climatic conditions at the building location (temperature, humidity and radiation), obtained from the USP Astronomical and Geophysical Institute (IAG-USP). After implementing and training each configuration with the available data for 2004, it was performed a validation process using the power consumption of the first three months of 2005, resulting in improvements of up to 10%, in the best scenario, in the forecasts made with more complex network configurations in comparison with the “feed-forward” network.

Capítulo 1

INTRODUÇÃO E OBJETIVOS

Nos últimos anos tem-se intensificado o interesse sobre o tema do consumo energético, em particular após a possibilidade de racionamento energético no ano 2001 no Brasil. Porém é que pesquisas para obter métodos para uma correta estimativa do consumo energético tornam-se uma valiosa ferramenta para a gestão predial. Dos diversos equipamentos e sistemas que integram uma instalação, os equipamentos associados ao acondicionamento de ar são os maiores consumidores, justificando assim a importância de se conseguir prever seu consumo energético.

O uso 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 factível, quando estão disponíveis os dados passados de 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 principal desses modelos, e que os tornam interessantes para este tipo de estudos, é justamente o fato de a rede ter 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, baseando-se em pesquisas anteriores, nesse trabalho foi avaliado o desempenho de redes neurais artificiais com configurações mais complexas do que a do tipo “*feed-forward*”, de forma a analisar o efeito dessas configurações da rede na estimativa do

consumo de energia em edificações com sistemas de ar condicionado, com base em dados climáticos de fácil acesso (como temperatura, umidade e radiação) e em dados anteriores de consumo da edificação.

Como estudo de caso utilizou-se 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.

Partindo da mesma hipótese das pesquisas anteriores, a qual era que as variações no consumo decorrem predominantemente dos equipamentos associados ao condicionamento de ar, vão ser 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).

A rede vai ser treinada com os dados climáticos e de consumo da edificação coletados no ano 2004, e vão ser comparados com os dados de consumo obtidos dos três primeiros meses do ano de 2005.

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.

Capítulo 2

Redes Neurais

O sistema nervoso humano é formado por uma rede de neurônios (unidade básica do cérebro) responsável pelos fenômenos conhecidos como pensamento, emoção e cognição, além da execução das funções sensório-motoras e autônomas. Cada neurônio compartilha várias características com outras células, mas possui capacidades singulares para receber, processar e transmitir sinais eletroquímicos ao longo das fibras nervosas, que compreendem o sistema de comunicação cerebral.

Um neurônio (Figura 2.1) conecta-se a vários outros neurônios por meio dos dendritos e do axônio. Os dendritos, uma complexa rede de prolongamentos, recebem impulsos nervosos de outros neurônios e os conduzem ao corpo celular ou núcleo. Esses impulsos são somados, gerando um novo impulso. Caso o resultado dessa soma exceda um determinado limiar, o axônio transmitirá esse estímulo a outros neurônios através de fenômenos químicos denominados sinapses. Por meio dessas sinapses os neurônios se unem funcionalmente, formando as redes neurais. A força sináptica da conexão neural, ao refletir o nível de excitação ou inibição entre neurônios adjacentes, capacita o cérebro humano ao armazenamento do conhecimento e o conseqüente aprendizado.

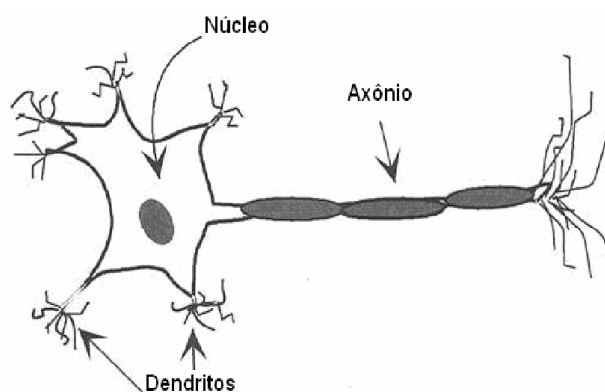


Figura 2.1. Neurônio biológico

2.1. REDES NEURAIS ARTIFICIAIS

Uma *Rede Neural Artificial* (RNA) é um conceito da computação que no qual procura-se processar os dados de maneira semelhante ao cérebro humano. O cérebro é tido como um processador altamente complexo e que realiza processamentos paralelos. Para isso ele organiza sua estrutura (ou seja, os neurônios) de forma que eles realizem as operações necessárias. Isso é feito numa velocidade extremamente alta e não existe qualquer computador no mundo capaz de realizar o que o cérebro humano faz.

Nas redes neurais artificiais, a idéia é realizar o processamento de informações tendo como princípio a organização de neurônios do cérebro. Como o cérebro humano é capaz de aprender e tomar decisões baseadas na aprendizagem, as RNA devem ser capazes de fazer o mesmo. Assim, uma RNA pode ser interpretada como um esquema de processamento capaz de armazenar conhecimento baseado em aprendizagem (experiência) e disponibilizar este conhecimento para a aplicação em questão.

2.1.1. Funcionamento de uma RNA

As redes neurais artificiais são criadas a partir de algoritmos projetados para uma determinada finalidade. Basicamente, uma RNA se assemelha ao cérebro em dois pontos: o conhecimento é obtido através de etapas de aprendizagem e pesos sinápticos são usados para armazenar o conhecimento. Uma sinapse é o nome dado à conexão existente entre neurônios. Nas conexões são atribuídos valores, que são chamados de pesos sinápticos. Isso deixa claro que as RNA têm em sua constituição uma série de neurônios artificiais (ou virtuais), mostrados na Fig. 2.2, que serão conectados entre si, formando uma rede de elementos de processamento.

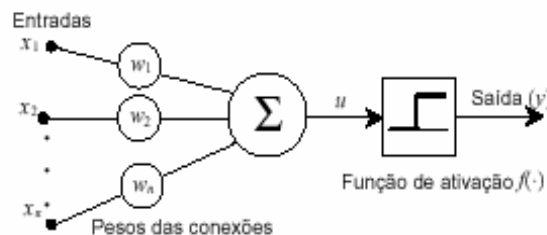


Figura 2.2. Representação esquemática de um neurônio artificial.

Uma vez montada a rede neural, uma série de valores podem ser aplicados sobre um neurônio. Estes valores (ou entradas) são ponderados no neurônio pelo peso de sua sinapse e somados. Se esta soma ultrapassar um valor limite estabelecido, um sinal é enviado pela saída (axônio) deste neurônio para os demais neurônios a ele conectados. Em seguida, essa mesma etapa se realiza com os demais neurônios da rede.

Existem várias formas de se desenvolver uma rede neural, que deve ser montada de acordo com o(s) problema(s) a ser(em) resolvido(s). Em sua arquitetura são determinados uma série de parâmetros, dentre os quais podemos destacar o número de camadas, a quantidade de neurônios em cada camada e o tipo de sinapse utilizado.

2.1.2. Aprendizado da RNA

Dentre todas as características das redes neurais artificiais, nenhuma desperta tanto interesse quanto a sua habilidade em realizar o aprendizado, decorrente do treinamento da rede por meio da apresentação de padrões às suas unidades visíveis.

O objetivo do treinamento consiste em definir os valores apropriados para os pesos sinápticos w (ver Fig. 2.2), de modo a produzir um conjunto de saídas dentro de um intervalo de erro estabelecido.

O processo de aprendizagem das redes neurais é realizado quando ocorrem várias modificações significativas nas sinapses dos neurônios. Essas mudanças ocorrem de acordo com a ativação dos neurônios. Se determinadas conexões são mais usadas, estas são reforçadas enquanto que as demais são enfraquecidas. É por isso que quando uma rede neural artificial é implantada para uma determinada aplicação, é necessário um tempo para que esta seja treinada.

Existem, basicamente, três tipos de aprendizado nas redes neurais artificiais:

- **supervisionado:** neste tipo, a rede neural recebe um conjunto de entradas padronizadas e seus correspondentes padrões de saída, onde ocorrem ajustes nos pesos sinápticos até que o erro entre os padrões de saída gerados pela rede tenham um valor desejado;
- **não-supervisionado:** neste tipo, a rede neural trabalha os dados de forma a determinar algumas propriedades do conjunto de dados. A partir destas propriedades é que o aprendizado é constituído;

- **híbrido:** neste tipo ocorre uma "mistura" dos tipos supervisionado e não-supervisionado. Assim, uma camada pode trabalhar com um tipo enquanto outra camada trabalha com o outro tipo.

2.1.3. Tamanho da Rede

Segundo a teoria, o número de unidades de processamento das camadas de entrada e saída é usualmente determinado pelo tipo de aplicação. No caso das camadas ocultas, a relação não é tão transparente. O ideal é utilizar o menor número possível de unidades ocultas para que a generalização da rede não fique prejudicada. Se o número de neurônios ocultos for muito grande, a rede acaba memorizando os padrões apresentados durante o treinamento. Contudo, se a arquitetura das camadas ocultas possuir unidades de processamento em número inferior ao necessário, o algoritmo “*back-propagation*” pode não conseguir ajustar os pesos sinápticos adequadamente, impedindo a convergência para uma solução.

2.1.4. O Algoritmo “*Back-Propagation*”

O algoritmo “*back-propagation*”, do tipo supervisionado, é o método de treinamento mais utilizado. Ele recebe esse nome porque inicia-se na última camada e retorna ajustando os pesos sinápticos até a camada de entrada. As equações básicas do algoritmo são:

$$e_j = Y_j(1 - Y_j)(d_j - Y_j) \quad (2.1)$$

$$t_j = t'_j + \lambda e_j \quad (2.2)$$

$$w_{ij} = w'_{ij} + \lambda e_j x_i \quad (2.3)$$

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:

$$e_j = Y_j(1 - Y_j) \sum (e_k w'_{jk}) \quad (2.4)$$

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.

2.1.5. Pesos e Parâmetros de Aprendizado

Normalmente os pesos das conexões entre as camadas de uma rede neural são inicializados com valores aleatórios e pequenos, para que se evite a saturação da função de ativação e a conseqüente incapacidade de se realizar a aprendizagem.

À medida que o treinamento evolui, os pesos sinápticos podem passar a assumir valores maiores, forçando a operação dos neurônios na região onde a derivada da função de ativação é muito pequena. Como o erro retro-propagado é proporcional a esta derivada, o processo de treinamento tende a se estabilizar, levando a uma paralisação da rede sem que a solução tenha sido encontrada. Isto pode ser evitado pela aplicação de uma taxa de aprendizagem menor. Teoricamente, o algoritmo de aprendizado exige que a mudança nos pesos seja infinitesimal. Entretanto, a alteração dos pesos nessa proporção é impraticável, pois implicaria em tempo de treinamento infinito. Em vista disso, é recomendável que a taxa de aprendizado assuma valor maior no início do treinamento e, à medida que se observe uma diminuição no erro da rede, essa taxa também seja diminuída.

Outra maneira de aumentar a velocidade de convergência da rede neural artificial treinada pelo algoritmo “*back-propagation*” é a adoção de um método chamado *momentum*. O propósito desse método consiste em adicionar, quando do cálculo do valor da mudança do

peso sináptico (Eq. 2.3), o valor da alteração anterior multiplicado por um momento M (daí o nome do método). Assim, a introdução desse termo na equação de adaptação dos pesos tende a aumentar a estabilidade do processo de aprendizado, favorecendo mudanças na mesma direção. Assim, a Eq. (2.3) é modificada para:

$$w_{ij} = w'_{ij} + (1 - M) e_j x_j + M (w'_{ij} - w''_{ij}) \quad (2.5)$$

2.1.5. Funções de ativação

A função de ativação basicamente fornece o valor da saída de um neurônio, e corresponde a um limiar que condiciona a propagação do impulso nervoso à transposição de um certo nível de atividade, mapeando o potencial da unidade de processamento, para um intervalo pré-especificado de saída.

As funções de ativação usualmente utilizadas em redes neurais são:

– limiar (ou degrau):
$$f(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases} \quad (2.6)$$

– signum:
$$f(x) = b \frac{x}{|x|} \text{ para } x \neq 0 \quad (2.7)$$

– tangente hiperbólica:
$$f(x) = a \frac{e^{(bx)} - e^{(-bx)}}{e^{(bx)} + e^{(-bx)}} \quad (2.8)$$

– sigmóide:
$$f(x) = \frac{1}{1 + e^{(-ax)}} \quad (2.9)$$

2.2. ARQUITETURA DAS REDES NEURAIS ARTIFICIAIS

Em geral é possível distinguir três classes fundamentais de arquiteturas: redes feedforward de uma única camada, redes feedforward de múltiplas camadas e redes recorrentes.

2.2.1. Redes “*Feed-forward*” de Uma Única Camada

Essa é a configuração de rede mais simples, na qual tem-se uma camada de entrada com neurônios cujas saídas alimentam a última camada da rede. Geralmente, os neurônios de entrada são propagadores puros, ou seja, eles simplesmente repetem o sinal de entrada em sua saída distribuída. Por outro lado, as unidades de saída costumam ser unidades processadoras, como apresentado na Fig. 2.3. A propagação de sinais nesta rede é puramente unidirecional (“*feed-forward*”): os sinais são propagados apenas da entrada para a saída.

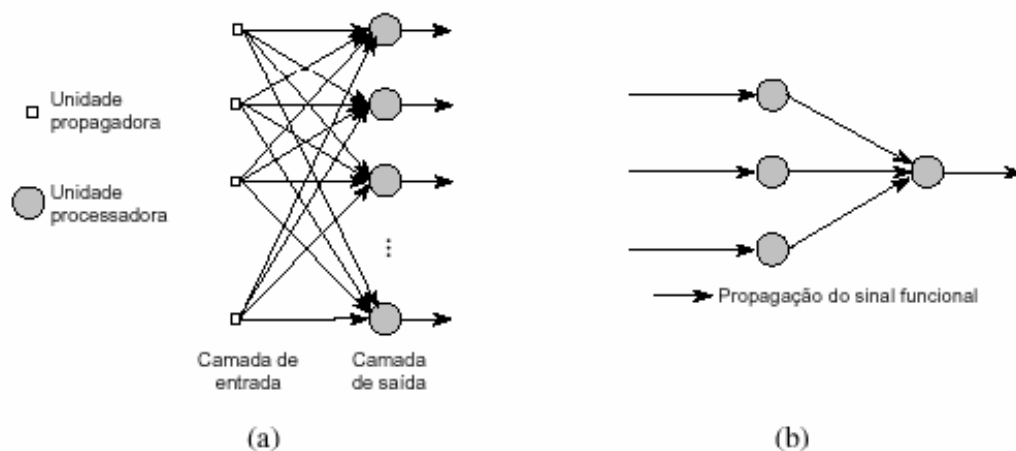


Figura 2.3. Redes neurais do tipo “*feed-forward*” com uma única camada de unidades processadoras.

2.2.2. Redes “*Feed-forward*” de Múltiplas Camadas

A segunda classe de rede “*feed-forward*” se distingue pela presença de uma ou mais camadas intermediárias ou escondidas (camadas em que os neurônios efetivamente unidades processadoras, mas não correspondem à camada de saída).

Adicionando-se uma ou mais camadas intermediárias, aumenta-se o poder computacional de processamento não-linear e armazenagem da rede. O conjunto de saídas dos neurônios de cada camada da rede é utilizada como entrada para a camada seguinte. A Fig. 2.4(a) ilustra uma rede “*feed-forward*” com duas camadas intermediárias.

As redes “*feed-forward*” de múltiplas camadas, são geralmente treinadas usando o algoritmo de retro-propagação do erro (“*error backpropagation*”), embora existam outros algoritmos de treinamento. Este algoritmo requer a propagação direta (“*feed-forward*”) do sinal de entrada através da rede, e a retro-propagação do sinal de erro, como ilustrado na Fig. 2.4(b).

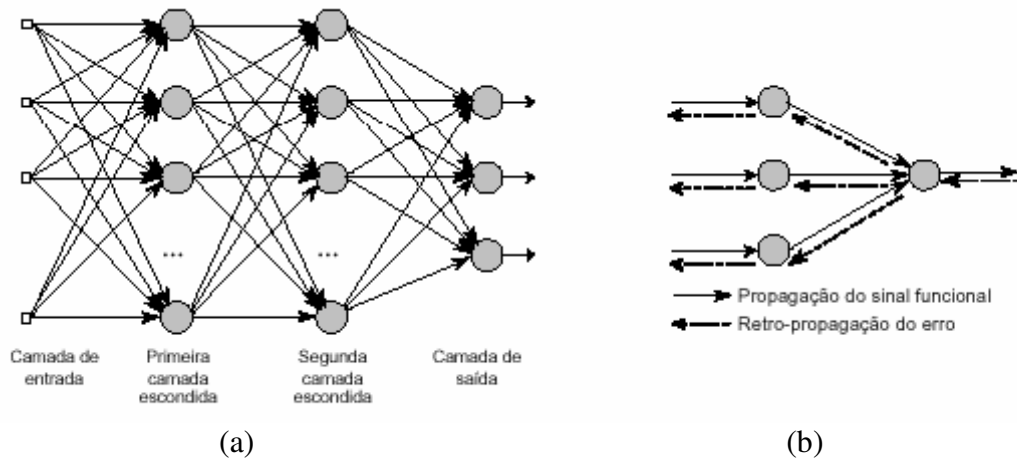


Figura 2.4. Redes neurais tipo “*feed-forward*” com múltiplas camadas: (a) arquitetura. (b) sentido de propagação do sinal funcional e do sinal de erro.

2.2.3. Redes Recorrentes

As redes recorrentes distinguem-se das redes feedforward pela existência de pelo menos um laço (loop) de recorrência (“*feed-back*”). Por exemplo, uma rede recorrente pode consistir de uma única camada de neurônios com cada neurônio alimentando seu sinal de saída de volta para a entrada de todos os outros neurônios, como ilustrado na Fig. 2.5. O laço de recorrência possui um grande impacto na capacidade de aprendizagem e no desempenho da rede. Além disso, a presença desse laço resulta em um comportamento dinâmico não-linear.

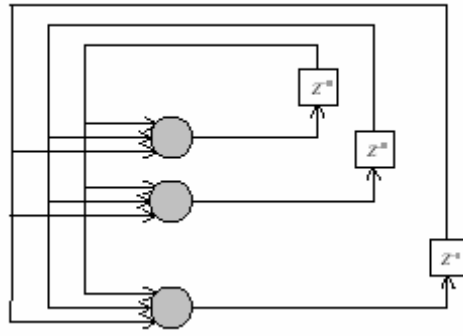


Figura 2.5. Arquitetura recorrente de rede sem nenhuma camada intermediária.

Capítulo 3

Configurações de Rede Implementadas

Para desenvolvimento do presente trabalho utilizou-se as rotinas de redes neurais Neurosolutions (2008) em conjunto com uma planilha EXCEL™. Dentre as opções de configuração de rede disponíveis, decidiu-se utilizar as configurações com a maior diferença nos seus algoritmos, de forma a se abranger o maior espaço de possibilidades possível, permitindo-se que a comparação dos modelos possa ser feita numa ampla gama de aspectos. Assim, os tipos de rede selecionados para utilização na pesquisa foram os seguintes:

- Perceptron Multicamadas (“*Multilayer Perceptron*” ou MLP; rede “*feed-forward*”);
- Rede de Hopfield (rede totalmente recorrente);
- Redes de Elman e Jordan (redes parcialmente recorrentes);
- Mapas Auto-Organizáveis (“*Self-Organizing Maps*” ou SOM);
- Redes Modulares.

A rede MLP já havia sido utilizada no trabalho anterior de Campoleone (2006) pelo fato de ser fácil de implementar e ser muito útil na resolução de problemas simples. Contudo, sob certas condições, o desempenho desse tipo de rede deixa de ser bom e precisa-se usar configurações mais complexas. No caso particular do escopo desse trabalho, o fato de a rede MLP implementada anteriormente ter fornecido resultados com erros médios da ordem de 10% foi o motivador para verificar se existe alguma configuração que permitiria aprimorar o desempenho do modelo de previsão. O restante desse capítulo será dedicado a uma descrição geral dos tipos de rede utilizados.

3.1. PERCEPTRON MULTICAMADAS

O primeiro modelo a revisar, será o modelo Perceptron Multicamadas (“*Multilayer Perceptron*”). Esse foi o modelo usado nas pesquisas anteriores, com o qual foram obtidos erros médios de ao redor de 10% em relação aos valores reais de consumo. Como se trata de uma nova implementação, com um software diferente do anterior, utilizou-se novamente tal algoritmo para confirmar os resultados anteriores e definir um caso de referência para verificação do eventual ganho de desempenho com os demais algoritmos.

3.1.1. O Modelo Perceptron

O modelo Perceptron, proposto por Rosenblatt, é composto por neurônios de com função de ativação do tipo limiar e aprendizado supervisionado. Sua arquitetura consiste em uma camada de entrada e uma camada de saída, como mostrado na Fig. 3.1. A limitação desta rede neural relaciona-se à reduzida gama de problemas que consegue tratar: classificação de conjuntos linearmente separáveis, como mostrado na Fig. 3.2.

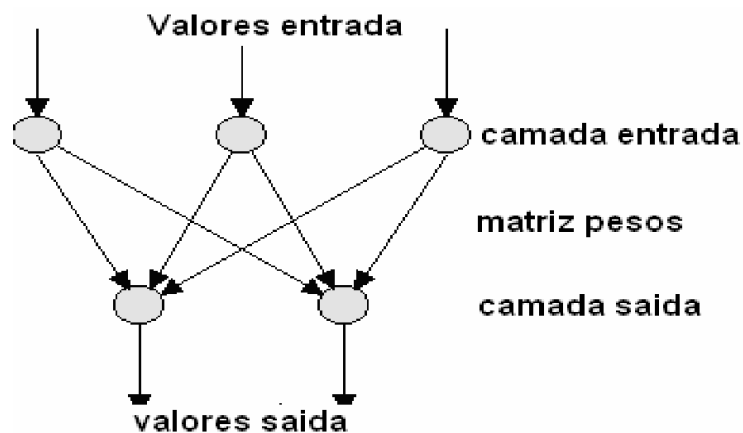


Figura 3.1. Esquema de uma rede neural perceptron.

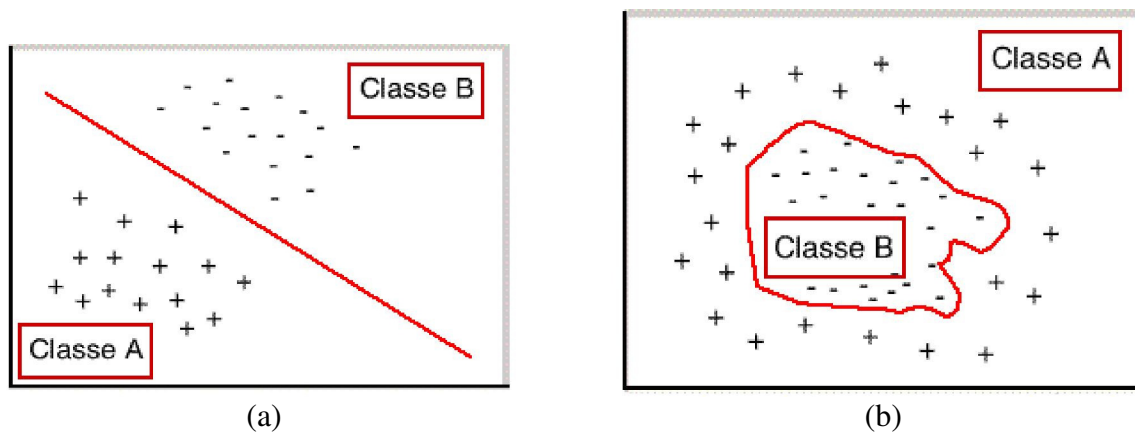


Figura 3.2: Classes (a) linearmente separáveis e (b) não linearmente separáveis.

3.1.2. O Perceptron Multicamadas

O Perceptron Multicamadas é uma extensão do modelo anterior de camada única. Esta arquitetura apresenta uma camada com unidades de entrada, conectada a uma ou mais unidades intermediárias localizadas em *camadas ocultas (hidden layers)*, e uma camada de unidades de saída.

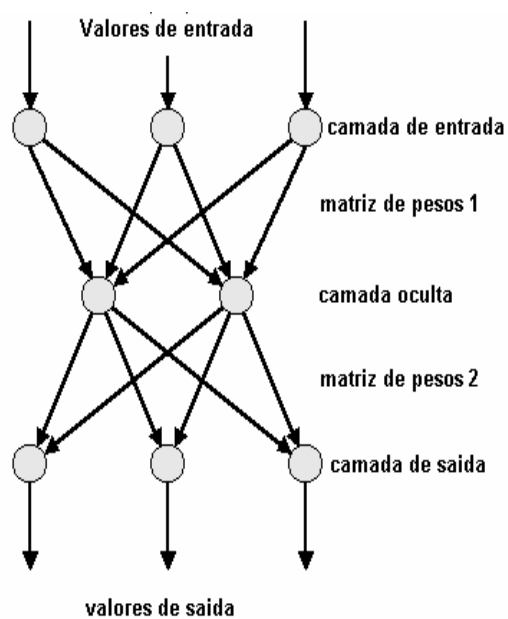


Figura 3.3. Rede Neural Perceptron Multicamadas

Esse tipo de rede utiliza o processo de aprendizado supervisionado, sendo mais comum a utilização do algoritmo “*back-propagation*”. Algumas características importantes devem ser ressaltadas:

- as unidades da rede utilizam uma função de ativação não linear (em geral a função sigmóide);
- a rede possui uma ou mais camadas ocultas, o que lhe permite solucionar problemas complexos, extraíndo as características mais significativas dos padrões de entrada;
- a rede possui alto grau de conectividade, o que permite interação entre as unidades.

A(s) camada(s) oculta(s) diferencia(m) este modelo do Perceptron de camada única, fornecendo-lhe maior poder computacional, embora dificulte(m) o algoritmo de treinamento.

A modelagem da arquitetura de uma rede Perceptron Multicamadas envolve a escolha da quantidade de camadas e o número de unidades em cada camada. Os aspectos importantes nesse processo são:

- escolha do número de unidades de entrada;
- definição da função de ativação que irá ditar o comportamento da rede;
- codificação da camada de saída e a formatação da resposta da rede.

Como já mencionado, o processamento de cada unidade é influenciado pelo processamento efetuado pelas unidades das camadas anteriores. Cada camada desempenha um papel específico. Por exemplo, numa rede com 4 camadas tem-se:

- **camada de entrada:** camada receptora dos estímulos;
- **primeira camada oculta:** cada unidade dessa camada define uma reta no espaço de decisão, refletindo características dos padrões apresentados;
- **segunda camada oculta:** combina as retas definidas pela camada anterior, formando regiões convexas onde o número de lados é definido pelo número de unidades da camada anterior conectados à unidade desta camada;
- **camada de saída:** combina as regiões formadas pela camada anterior, definindo o espaço de saída da rede.

As camadas intermediárias da rede são como detectores de características, as quais serão representadas, internamente, através dos pesos sinápticos. As pesquisas sobre o

número de camadas intermediárias necessárias para implementar uma Rede Neural indicam que:

- uma camada é suficiente para aproximar qualquer função contínua;
- duas camadas são suficientes para aproximar qualquer função matemática.

Para a escolha do número de unidades em cada camada, deve-se considerar:

- O número de exemplos de treinamento;
- A quantidade de ruído presente nos exemplos;
- A complexidade da função a ser aprendida pela rede;
- A distribuição estatística dos dados de treinamento.

Na determinação do número de unidades em cada camada, os seguintes cuidados devem ser tomados:

- não utilizar um número de unidades maior que o necessário: um grande número de unidades pode fazer com que a rede memorize os dados do treinamento; com isto ela torna-se incapaz de generalizar e, portanto, reconhecer padrões não vistos durante o treinamento; isso é chamado de *overfitting*;
- não utilizar um número de unidades inferior ao número necessário: isso pode fazer com que a rede gaste muito tempo para aprender, podendo não alcançar os pesos adequados, ou seja, a rede pode não convergir ou generalizar demais os padrões de entrada.

3.1.3. Treinamento do MLP

Além dos parâmetros de configuração da rede, outros parâmetros referentes ao treinamento devem ser escolhidos: taxa de aprendizado e o conjunto de treinamento. A respeito do conjunto de treinamento devem ser estudados os dados relevantes, os quais destaquem as características que devem realmente ser aprendidas pela rede.

A essência do aprendizado “*back-propagation*” é codificar uma relação funcional entre as entradas e saídas, representada por um conjunto de valores $\{x, d\}$ associados aos pesos sinápticos e limiares (“*thresholds*”) de um perceptron de múltiplas camadas (MLP). A idéia básica é que a rede aprenda bastante sobre o passado para poder generalizar sobre o futuro. Nesta perspectiva, o processo de aprendizado equivale a uma escolha de parametrização da rede para este conjunto de exemplos. Mais especificamente, podemos ver o

problema de seleção da rede como sendo de escolha, dentro de um conjunto de estruturas de modelos candidatas (parametrizações), a “melhor” delas de acordo com um certo critério.

Uma ferramenta estatística muito útil na etapa de treino de uma rede MLP, é a validação cruzada (“*cross validation*”). Primeiramente o o conjunto de dados disponível é particionado aleatoriamente em um *conjunto de treinamento* e um *conjunto de teste*. A seguir, o conjunto de treinamento é novamente particionado em dois subconjuntos:

- *subconjunto de estimação*, com cerca de 80 a 90% dos dados, usado para selecionar (treinar) o modelo;
- *subconjunto de validação*, com os 10 a 20% restantes dos dados, usado para validar (avaliar o desempenho) o modelo.

O algoritmo para treinamento de Redes Multicamadas mais difundido é o “back-propagation”. Esse algoritmo divide-se em duas partes, como mostrado na Fig. 3.4:

1. *propagação*: depois de apresentado o padrão de entrada, a resposta de uma unidade é propagada como entrada para as unidades na camada seguinte, até a camada de saída, onde é obtida a resposta da rede e o erro é calculado;
2. *retro-propagação*: a partir da camada de saída, são feitas alterações nos pesos sinápticos até atingir-se a camada de entrada.

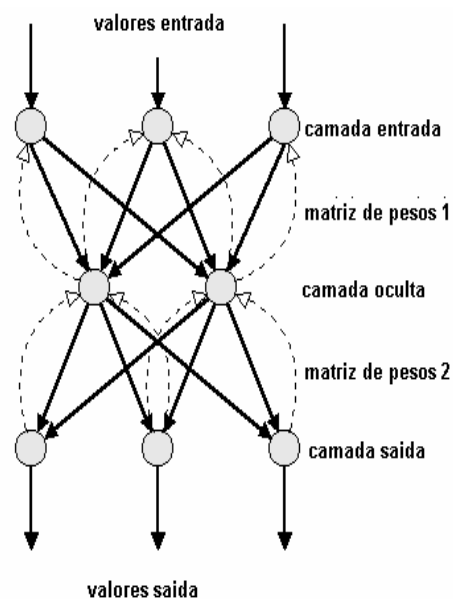


Figura 3.4. Esquema do algoritmo “back-propagation”.

Durante a fase treinamento devemos apresentar um conjunto formado pelo par (*entrada para a rede; valor desejado para resposta a entrada*) . A saída fornecida pela rede será comparada ao valor desejado e será computado o erro global da mesma, que influenciará na correção dos pesos no passo de retro-propagação. Apesar de não haver garantias de que a rede forneça uma solução ótima para o problema, este processo é muito utilizado por apresentar uma boa solução para o treinamento de Perceptrons Multicamadas. Os detalhes e equações utilizadas nesse algoritmo podem ser encontrados nas referências citadas (Freeman & Skapura, 1991; Haykin, 1994).

3.2. REDES NEURAI RECORRENTES (“FEED-BACK”)

Muitos algoritmos de treinamento das RNAs não são capazes de implementar mapeamentos dinâmicos, como por exemplo o algoritmo de retro-propagação simples, o qual pode apenas aprender mapeamentos estáticos. Um artifício utilizado para processamento temporal utilizando estas redes envolve o uso de janelas de tempo, onde a entrada da rede utiliza trechos dos dados temporais como se eles formassem um padrão estático.

Entretanto, esta solução não é a mais indicada para tratar problemas em que haja uma dependência temporal. A principal questão, portanto, é como estender a estrutura das redes MLP para que assumam um comportamento dinâmico, sendo assim capazes de tratar sinais temporais.

Para uma RNA ser considerada dinâmica, é preciso que ela “possua memória”. Existem basicamente duas maneiras de prover memória a uma RNA:

- introduzindo atraso no tempo, como nas técnicas TDNN (“*Time Delay Neural Network*” e FIR “*Multilayer Perceptron*”);
- utilizar configurações de redes neurais recorrentes, tais como “*Backpropagation Through Time*”, “*Real-Time Recurrent Learning*”, “*Cascade Correlation*” ou a rede de Elman/Jordan.

Define-se redes recorrentes como aquelas que possuem conexões de realimentação. Há dois tipos de redes recorrentes: aquelas em que o padrão de entrada é fixo e a saída caminha, dinamicamente, para um estado estável, e aquelas em que ambas, entrada e saída, variam com o tempo, sendo estas últimas mais gerais e utilizadas com maior frequência.

Há muitas variações de arquiteturas de redes recorrentes, sendo que algumas delas permitem o uso de algoritmos de treinamento mais simples ou adaptados a uma tarefa particular. Duas maneiras que podem ser usadas para treinar uma rede recorrente, e que não envolvem o uso de aproximações na computação dos gradientes, são a “*Backpropagation Through Time*” e as redes recorrentes de tempo real. As RNA recorrentes também podem ser classificadas em 2 grupos: Redes Totalmente Recorrentes, ou Redes Parcialmente Recorrentes. No grupo das Redes Totalmente recorrentes, a mais conhecida é a Rede de Hopfield.

Considerando a disponibilidade do software e as características desse trabalho, foram implementadas as RNA recorrentes de Hopfield e Elman/Jordan, descritas a seguir.

3.2.1. Redes de Hopfield

O modelo descrito por Hopfield, também conhecida como Memória Associativa, consiste em um modelo matricial não linear recorrente, ou seja, as saídas estão ligadas às entradas através de um atraso de tempo. Não linearidades são aplicadas às saídas de cada um dos nós. A recorrência dá ao modelo características temporais que implicam na resposta da rede depender sempre do seu estado no intervalo de tempo anterior. O armazenamento e recuperação da informação consiste então na criação de pontos fixos e em uma regra de atualização que defina a dinâmica da rede. A Fig. 3.5 mostra um diagrama esquemático da rede de Hopfield.

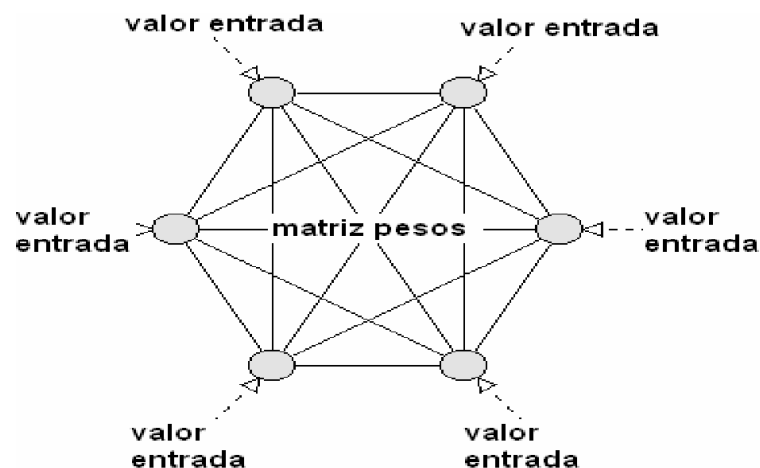


Figura 3.5. Esquema da rede de Hopfield

O modelo de Hopfield é inerentemente auto-associativo, ou seja, para que sejam criados pontos fixos através da recorrência, um vetor é associado aos mesmos. Uma outra característica do modelo de Hopfield é que, na forma como foi descrito originalmente, as saídas dos nodos são discretizados e somente podem assumir valores -1 ou 1, já que suas funções de ativação são do tipo degrau com saturação nestes valores. Posteriormente à descrição inicial, Hopfield mostrou que o modelo com saídas contínuas preserva as características do modelo discreto original.

3.2.2. Redes de Elman e Jordan

As redes de Elman e Jordan são redes parcialmente recorrentes. Nas redes de Elman, além das unidades de entrada, intermediárias e de saída, há também unidades de contexto, como nas redes parcialmente recorrentes em geral. As unidades de entrada e saída interagem com o ambiente externo, enquanto as unidades intermediárias e de contexto não o fazem. As unidades de entrada são apenas unidades de armazenamento (“*buffers*”) que passam os sinais sem modificá-los. As unidades de saída são unidades lineares que somam os sinais que recebem. As unidades intermediárias podem ter funções de ativação lineares ou não lineares, e as unidades de contexto são usadas apenas para memorizar as ativações anteriores das unidades intermediárias, podendo ser consideradas como atraso no tempo em um passo. As conexões “*feed-forward*” são modificáveis e as conexões recorrentes são fixas, motivo pelo qual essa rede (assim como a de Jordan) é apenas parcialmente recorrente.

Em um intervalo de tempo específico k , as ativações das unidades intermediárias (em $k-1$) e as entradas correntes (em k) são utilizadas como entradas da rede. Em um primeiro estágio “*feed-forward*”, estas entradas são propagadas para frente a fim de produzir as saídas. Posteriormente, a rede é treinada com o algoritmo de aprendizagem de retropropagação padrão. Após este passo de treinamento, as ativações das unidades intermediárias no tempo k são reintroduzidas através das ligações recorrentes nas unidades de contexto, sendo salvas nestas unidades para o próximo passo do treinamento ($k+1$). No início do treinamento, as ativações das unidades intermediárias são desconhecidas e, geralmente, são inicializadas para a metade do valor máximo que as unidades intermediárias podem ter.

Na rede de Jordan, a saída da rede é copiada para a unidade de contexto. Adicionalmente, as unidades de contexto são localmente recorrentes. A grande diferença em termos de topologia entre as duas redes é que a recorrência na rede de Elman é feita da camada oculta para as entradas, enquanto que na rede de Jordan a recorrência é feita das saídas para as entradas. A Fig. 3.6 apresenta os esquemas das redes de Elman e Jordan.

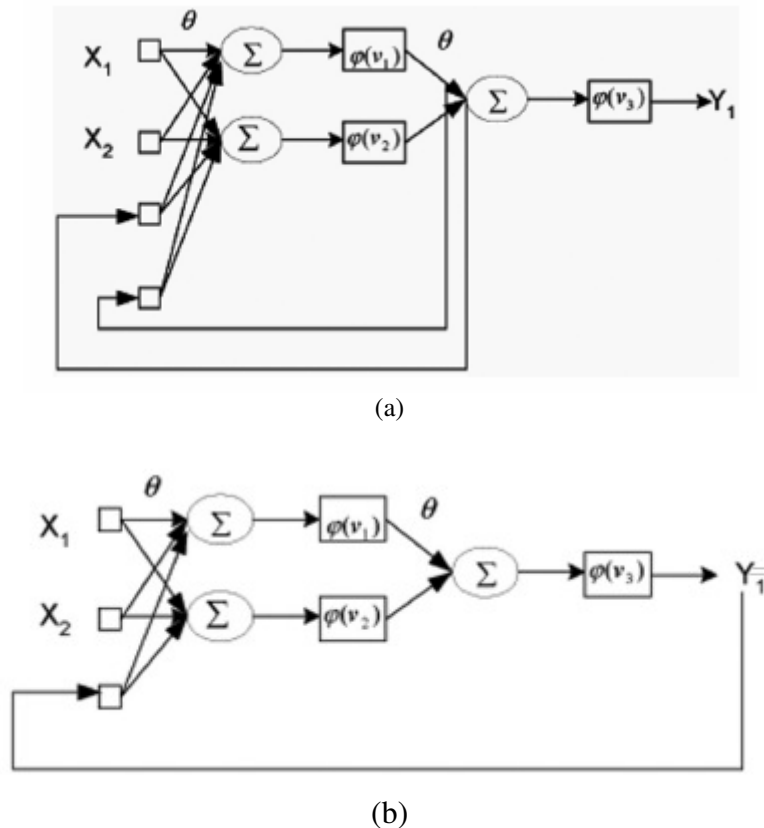


Figura 3.6. Esquema das redes de (a) Elman e (b) Jordan

3.3. MAPAS AUTO-ORGANIZÁVEIS DE KOHONEN

Inspirada nos mapas corticais, a rede de Kohonen (Fig. 3.7) utiliza o aprendizado competitivo, onde os neurônios competem entre si para responder a um estímulo apresentado. Durante o aprendizado, formam-se agrupamentos de neurônios topologicamente organizados, onde cada grupo é responsável por responder a uma classe de estímulos. A característica de auto-organização, que dá nome a esta classe de redes neurais, é devida ao fato de o aprendizado competitivo utilizar regras de aprendizado. A única informação apresentada à rede são os padrões de entrada, e as ligações sinápticas são definidas de forma a recompensarem o neurônio vencedor, sem comparação com padrões desejados. Como já

mencionado anteriormente, maiores detalhes sobre os algoritmos e o processo de treinamento podem ser obtidos nas referências citadas.

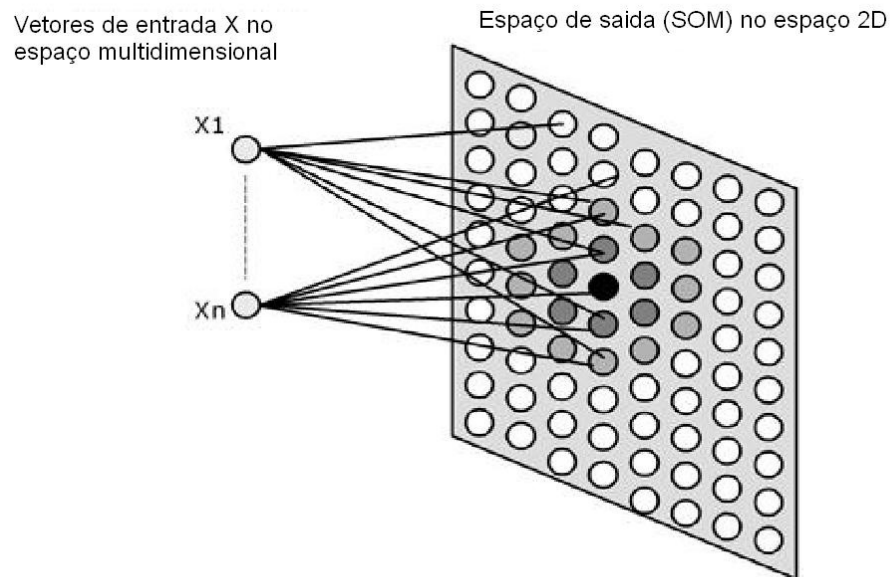


Figura 3.7. Esquema da Rede de Kohonen

3.4. REDES NEURAIAS MODULARES

As redes neurais artificiais mais utilizadas têm uma estrutura rígida, e apresentam um bom desempenho quando o conjunto de dados de entrada é pequeno. Contudo, a complexidade aumenta e o desempenho diminui rapidamente com um conjunto de entrada de grande dimensão. Diversas pesquisas têm sido feitas para superar esse problema, utilizando a modularidade como conceito básico. Contudo, os problemas principais nesse caso são a escolha dos módulos e a forma de estruturar os problemas.

O princípio básico de uma rede modular, mostrado na Fig. 3.8, é constituí-la por vários módulos MLP, cada um deles treinado pelo algoritmo “*back-propagation*”. Dessa forma, o número de conexões de peso na arquitetura proposta é consideravelmente menor do que em uma rede única.

Os módulos normalmente utilizam uma rede MLP de 2 camadas. Cada variável de entrada está conectada a apenas um dos módulos de entrada, escolhido aleatoriamente, e as saídas de todos os módulos de entrada estão conectados à rede de decisão.

O treino supervisionado ocorre em 2 estágios. Na primeira fase, todas as sub-redes na camada de entrada são treinadas. Os dados de treinamento de cada sub-rede são escolhidos do grupo de dados original. O par de treino de cada módulo consiste nos componentes do vetor original, os quais estão conectados a sua rede particular (como vetor de entrada) junto com a saída de decisão representada em código binário. Cada módulo de entrada pode ser treinado em paralelo facilmente porque todos são mutuamente independentes.

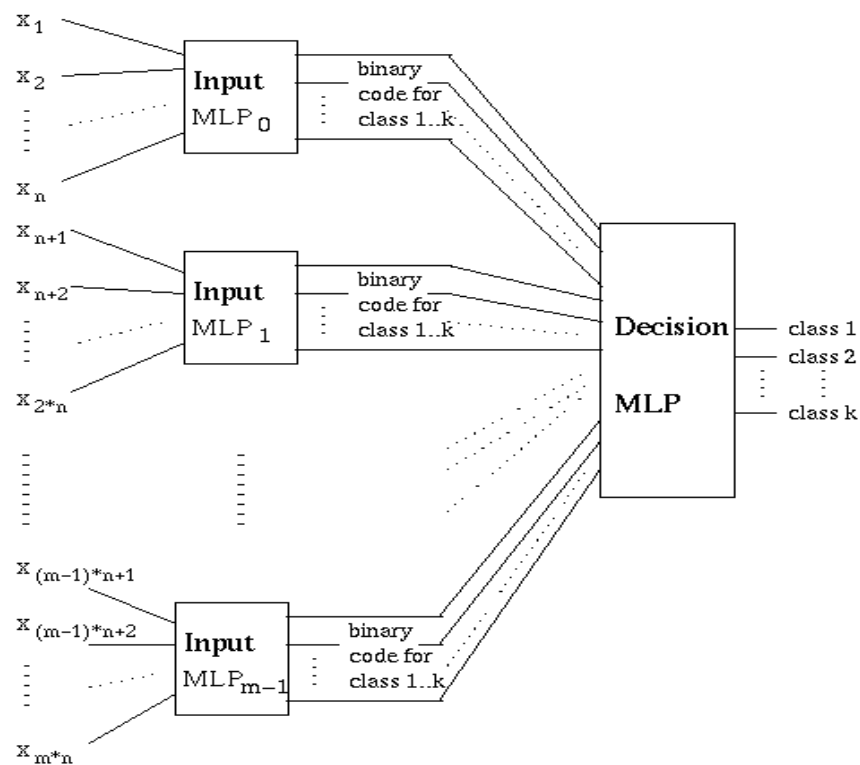


Figura 3.8. Esquema de uma Rede Neural Modular constituída por sub-redes MLP.

No segundo estágio a rede de decisão é treinada. O grupo de treinamento para o módulo de decisão é constituído pelas saídas da camada de entrada junto com o número de classe original. Para calcular o grupo, cada padrão de entrada original é aplicado à camada de entrada, o vetor resultante junto com a classe da saída de decisão (representada num código “1-out-of-k”) forma o par de treino para o módulo de decisão.

O cálculo da saída para novos vetores de entrada também é realizado em 2 estágios. Primeiro o novo vetor de entrada é apresentado nos módulos de entrada. Então as saídas dos módulos de entrada são usadas como entrada no módulo de decisão. Na base desta entrada o

resultado final é calculado. A saída de dimensão k do módulo de decisão é usado para determinar o número de classe da entrada dada.

3.5. IMPLEMENTAÇÃO DOS MODELOS

Apesar dos modelos de redes neurais não exigirem o conhecimento do fenômeno físico em estudo, esse conhecimento se faz necessário na escolha do conjunto de dados de entrada para a rede, de modo que ele seja coerente com a saída que se deseja. Considerando o estudo ora apresentado em que se pretende prever o consumo de energia em uma edificação com sistemas de climatização, foram assumidas as seguintes hipóteses:

- a) que a variação no consumo energético da instalação predial em análise decorre predominantemente devido ao uso dos equipamentos de condicionamento de ar;
- b) que o uso desses equipamentos é influenciado direta e principalmente pelas condições climáticas no local da edificação;
- c) que 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. Como já mencionado, para implementação dos modelos foi utilizado o software NeuroSolutions (2008), utilizando-se como a função tangente hiperbólica como função de ativação e a regra do *momentum* com uma taxa de aprendizagem de 0,7 para todos os modelos implementados.

Capítulo 4

Estudo de Caso

A Reitoria da USP, objeto do estudo, localiza-se no *campus* da Cidade Universitária, na Rua da Reitoria, número 109. Foi construído na 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 três mil metros quadrados. Cerca de novecentas pessoas trabalham diariamente no conjunto. Os equipamentos de condicionamento de ar são do tipo *split* e janela (Aquino, 2005).

4.1. PERFIS CLIMÁTICO E DE CONSUMO DA EDIFICAÇÃO

O edifício da Reitoria conta com um sistema de monitoramento do consumo de energia (SISGEN), implementado pelo Grupo de Energia do Departamento de Engenharia de Energia e Automação Elétricas da Escola Politécnica da USP (GEPEA-USP). Foram obtidos junto aos gestores do SISGEN os dados de potência elétrica consumida (w , kW) a cada quinze minutos para o período de agosto de 2003 a março de 2005, a partir dos quais determinou-se o consumo energético diário (C , kWh).

Para o mesmo período foram levantados os seguintes dados climáticos junto à estação climática do Instituto Astronômico e Geofísico da USP (IAG-USP), localizada no *campus* da Cidade Universitária: temperatura (T , °C), umidade relativa (U , %), radiação global (R_{glo} , W/m²) e radiação difusa (R_{dif} , W/m²), todos eles para a região próxima ao prédio da reitoria, aquisitados a cada cinco minutos. As Figuras 4.1 a 4.3 apresentam os perfis diários típicos de potência elétrica consumida e de dados climáticos obtidos.

Na Fig. 4.1 é possível verificar que existe uma grande variação entre o perfil de potência consumida em um dia útil e aquele de finais de semana e feriados. Isso era esperado, pois nesses dias o uso do prédio, e por conseqüência, de seus equipamentos, é

consideravelmente reduzido. Isso motivou Campoleoni (2006) a realizar três testes distintos para a rede neural: no primeiro deles alimentou-se a rede com a informação de todos os dias, no segundo, testou-se a rede apenas com os dados dos dias úteis e no terceiro, com a informação dos finais de semana e feriados. Por consistência com esse trabalho prévio, manteve-se essa abordagem no presente trabalho.

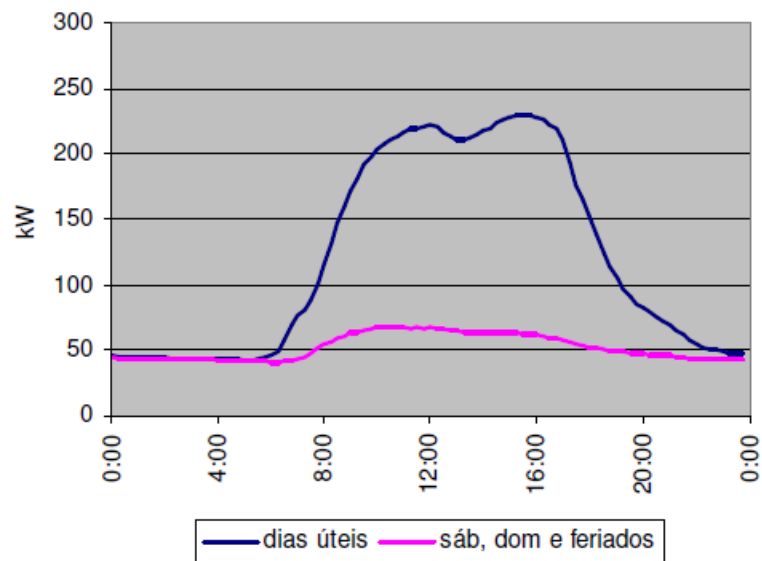


Figura 4.1. Perfil diário de potência elétrica consumida.

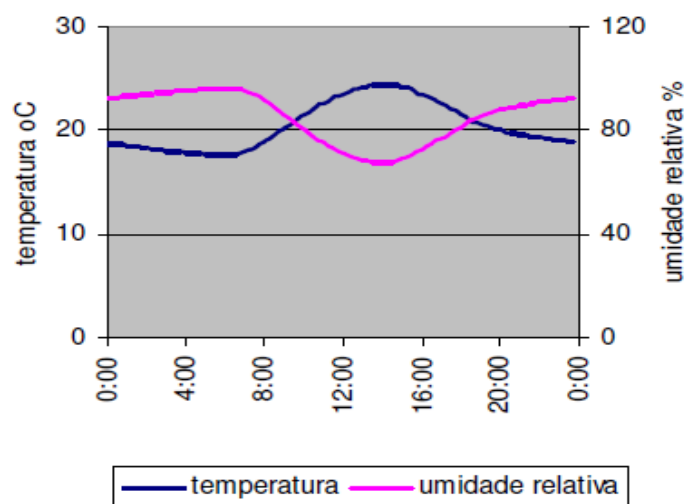


Figura 4.2. Perfil diário de temperatura e umidade

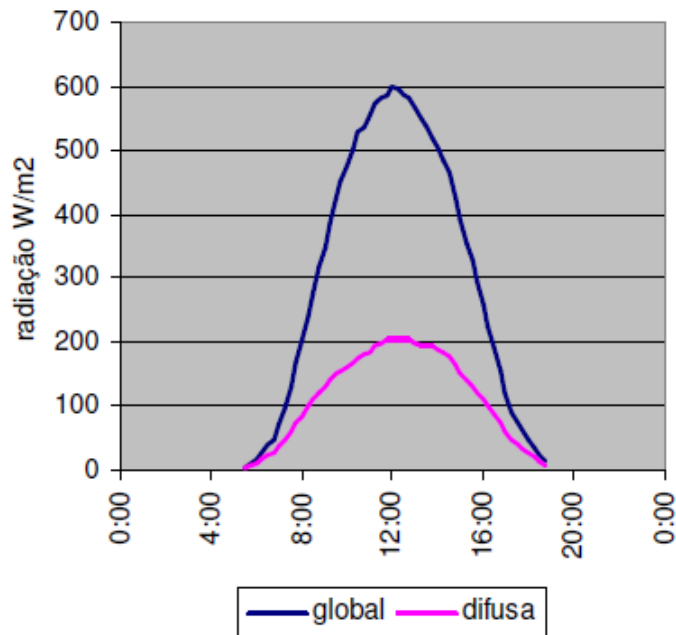


Figura 4.3. Perfil diário de radiação

Outro resultado obtido por Campoleoni, foi que a introdução da umidade relativa e da radiação como dados de entrada na rede não trouxe ganhos significativos em termos de redução do erro quando comparado com uma rede que utiliza apenas a temperatura como entrada. Assim, como o foco desse trabalho foi a comparação de diferentes arquiteturas de rede, optou-se por uma implementação apenas de redes temperatura–consumo.

4.2 TRATAMENTO DOS DADOS DISPONÍVEIS

A massa total dos dados foi dividida em dois grupos para uso no presente estudo:

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

O primeiro grupo foi utilizado para alimentar a rede, informando a ela tanto os valores climáticos, quanto os dados de consumo. Uma vez treinada a rede, o segundo grupo foi utilizado para validar a rede montada, alimentando-a somente com os dados climáticos e comparando os dados de consumo fornecidos pela rede com os dados medidos pelo SISGEN.

É importante destacar 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. No trabalho anterior, para contornar essa situação foram testadas duas situações: eliminar os registros e completar os registros. No primeiro caso, foram omitidos 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.

A análise dos resultados mostrou que, nos testes realizados utilizando os valores médios mensais nos buracos das medições, as variações entre os resultados fornecidos pelas redes aumentaram consideravelmente para todas as configurações, chegando a valores não aceitáveis. A inclusão destes valores acabou descaracterizando o perfil de consumo e temperatura, prejudicando a aprendizagem das redes. Essa conclusão foi assumida como hipótese nesse trabalho, optando-se por eliminar os registros faltantes.

4.3. RESULTADOS OBTIDOS

Os dados climáticos e de consumo foram separados em dois grupos. O primeiro deles, utilizado para treino, foi composto pelas medições de janeiro até dezembro de 2004, e dentro desse grupo criou-se um sub-grupo com os dados dos meses de Abril e Setembro de 2004 para ser usado na validação cruzada. O segundo grupo, usado para validação final da rede, foi 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. É importante destacar que para os dados de entrada foram descartados os dias com menos de 90% das medições feitas.

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. A Tab. 4.1 resume os parâmetros utilizados nas redes temperatura-consumo para as diversas arquiteturas de rede. A comparação do desempenho das diversas redes foi feito por meio do Erro Quadrático Médio Normalizado (EQMN), definido por:

$$EQMN = \frac{1}{N} \sum_{i=1}^N (Y_{norm,i} - f_{norm,i}(x, w))^2 \quad (4.1)$$

onde $Y_{norm,i}$ representa o dado de consumo real normalizado entre -1 e 1, $f_{norm,i}(x,w)$ representa a saída da rede neural, também normalizada entre -1 e 1.

Tabela 4.1: Parâmetros usados para as redes temperatura–consumo

Parâmetros	Todos os dias	Dias úteis	Finais de semana e Feriados
Variáveis de entrada		2 (T_{min}, T_{max})	
Variáveis de saída		1 (C)	
Dados para treino			
Total	241	163	78
Treino	217	147	70
Validação Cruzada	24	16	8
Dados para Validação	81	56	25

4.3.1. Perceptron Multicamadas (MLP)

Os parâmetros da rede MLP / temperatura–consumo são apresentados na Tabs. 6.2 e seus resultados são resumidos na Tab. 6.3 e nas Figs. 6.1 e 6.2 para a rede “Todos os dias”, nas Figs. 6.3 e 6.4 para a rede “Dias úteis” e nas Figs. 6.5 e 6.6 para a rede “Final de Semana e Feriados”.

Tabela 4.2. Parâmetros para a rede MLP / temperatura–consumo

Parâmetros	Todos os dias	Dias úteis	Finais de semana e Feriados
Numero de camadas	3	3	3
Neurônios na camada de entrada	2	2	2
Neurônios na camada de saída	1	1	1
Neurônios na camada oculta	5	14	8
Número de ciclos de treino	1000	1000	1000

Tabela 4.3. Resultados para a rede MLP / temperatura–consumo

Resultados	Todos os dias	Dias úteis	Finais de semana e Feriados
EMQ mínimo no treino	0,1227	0,0725	0,0294
EMQN	0,8897	0,6501	0,9372

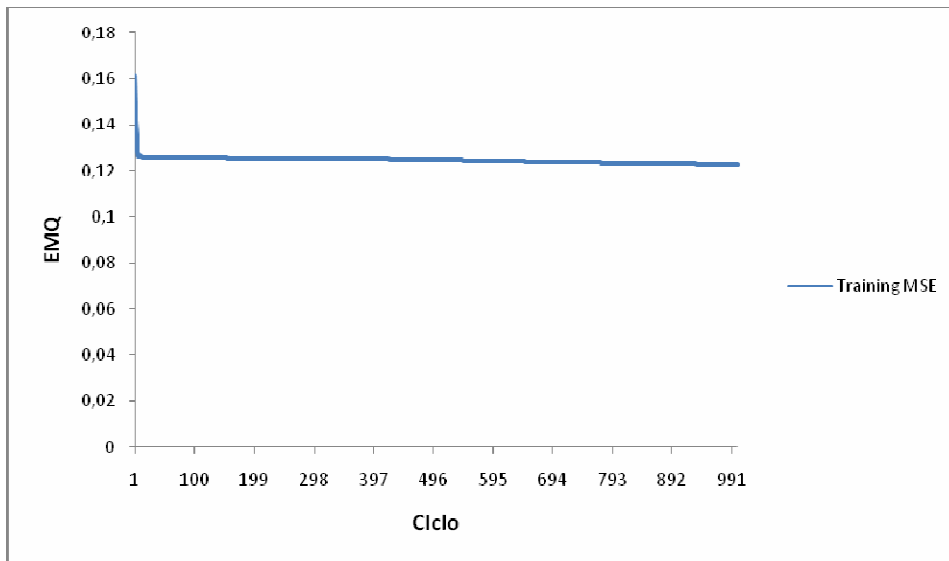


Figura 4.1. Rede MLP / temperatura–consumo
(Etapa de treino, Todos os dias)

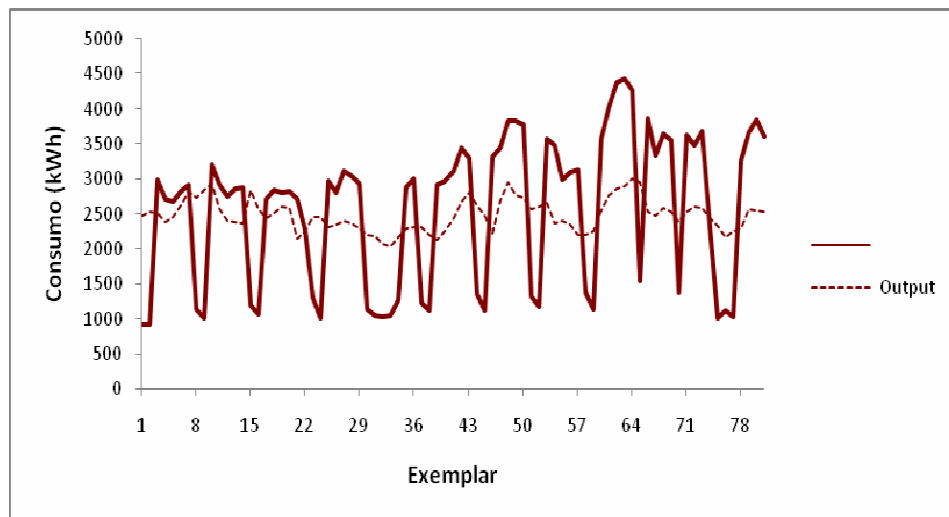


Figura 4.2. Rede MLP / temperatura–consumo
(Etapa de validação, Todos os dias)

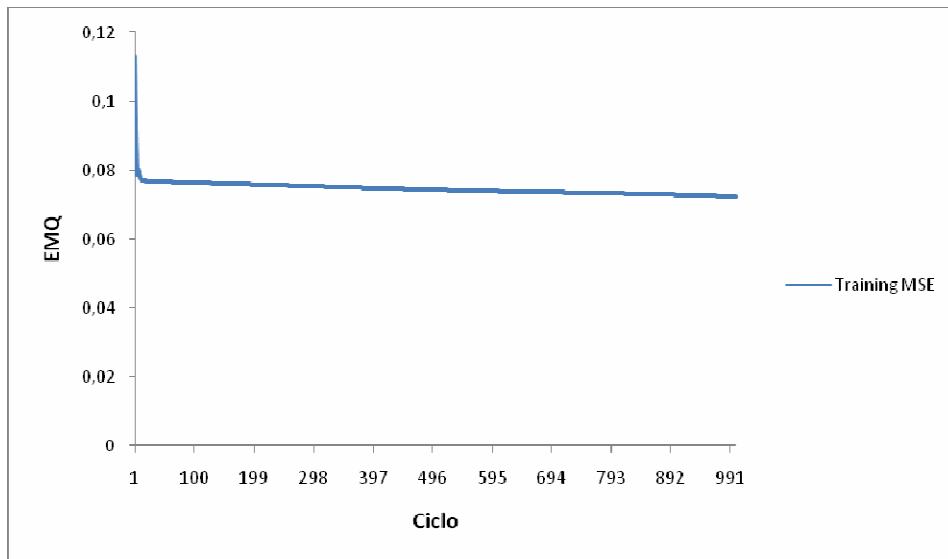


Figura 4.3. Rede MLP / temperatura–consumo
(Etapa de treino, Dias Úteis)

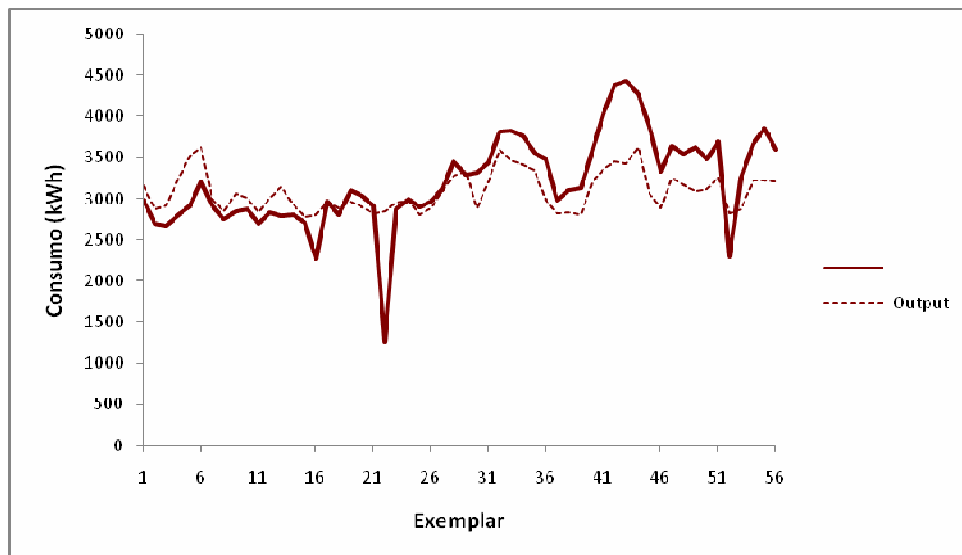


Figura 4.4. Rede MLP / temperatura–consumo
(Etapa de validação, Dias Úteis)

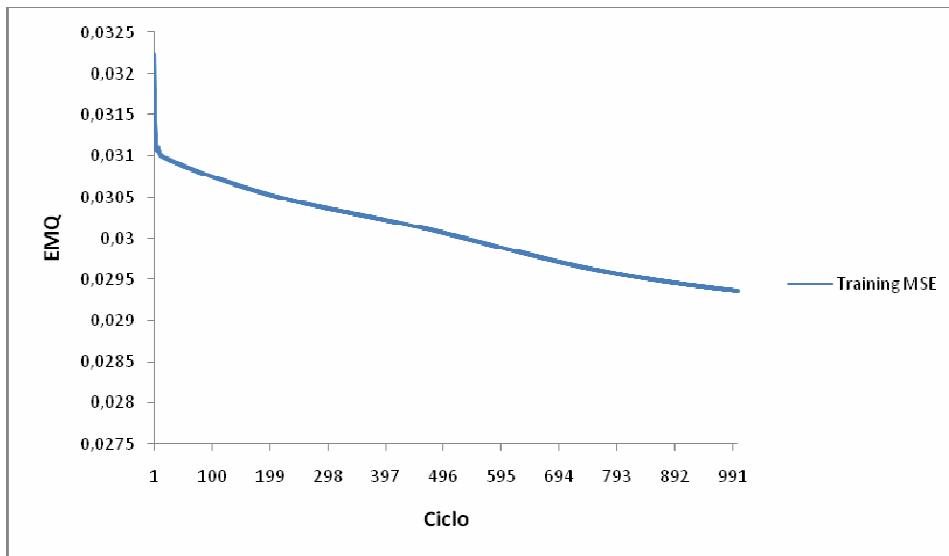


Figura 4.5. Rede MLP / temperatura–consumo
(Etapa de treino, Final de semana e Feriados)

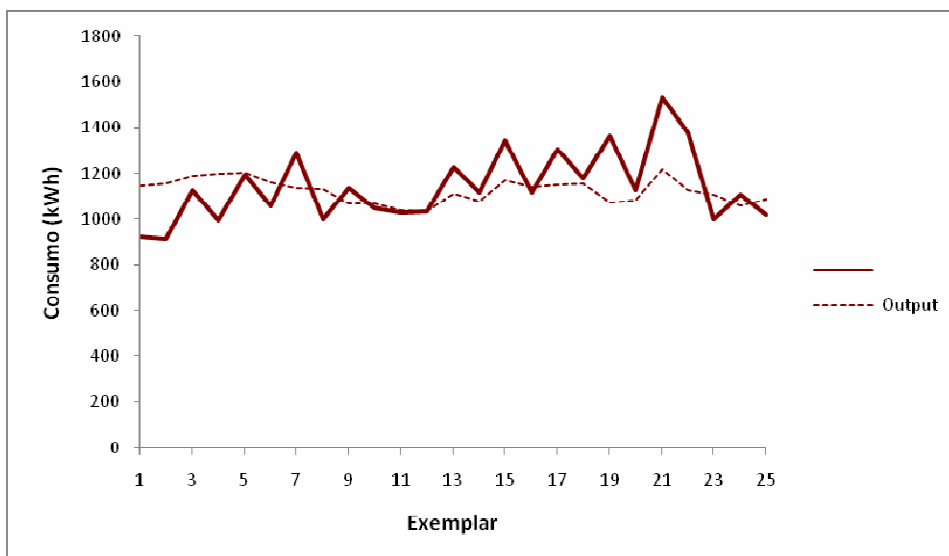


Figura 4.6. Rede MLP / temperatura–consumo
(Etapa de validação, Final de semana e Feriados)

4.3.2. Rede de Hopfield

Os parâmetros da rede Hopfield / temperatura–consumo são apresentados na Tab. 4.4 e seus resultados são resumidos na Tab. 4.5 e apresentados nas Figs. 4.7 e 4.8 para a rede “Todos os dias”, nas Figs. 4.9 e 4.10 para a rede “Dias úteis”, e nas Figs. 4.11 e 4.12 para a rede “Final de semana e Feriados”.

Tabela 4.4. Parâmetros da rede Hopfield / temperatura–consumo

Parâmetros	Todos os dias	Dias úteis	Finais de semana e feriados
Numero de camadas	3	3	3
Neurônios na camada de entrada	2	2	2
Neurônios na camada de saída	1	1	1
Neurônios na camada oculta	6	4	2
Número de ciclos de treino	1000	1000	1000

Tabela 4.4. Resultados da rede Hopfield / temperatura–consumo

Resultados	Todos os dias	Dias úteis	Finais de semana e Feriados
EMQ mínimo no treino	0,1114	0,0567	0,0264
EMQN	0,9693	0,9053	0,9656

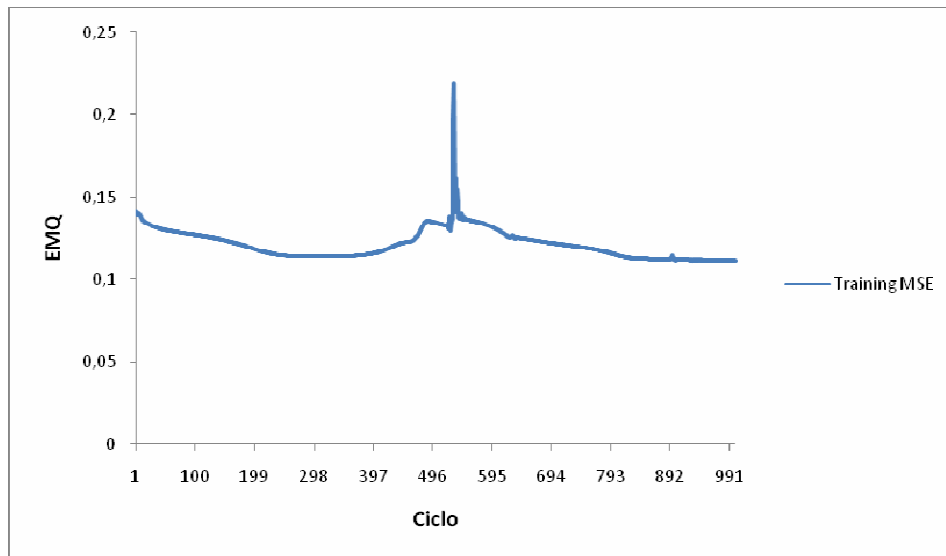


Figura 4.7. Rede Hopfield / temperatura–consumo
(Etapa de treino, Todos os dias)

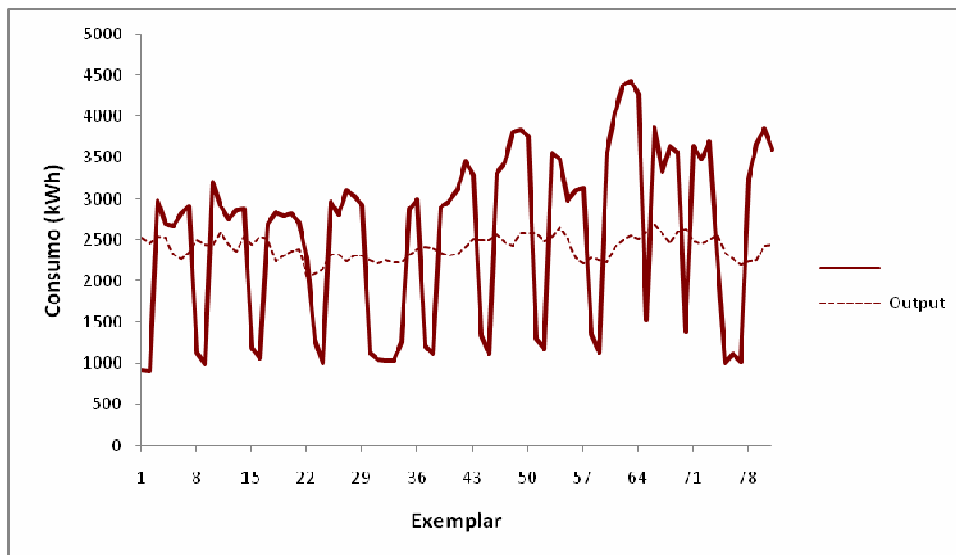


Figura 4.8. Rede Hopfield / temperatura–consumo
(Etapa de validação, Todos os dias)

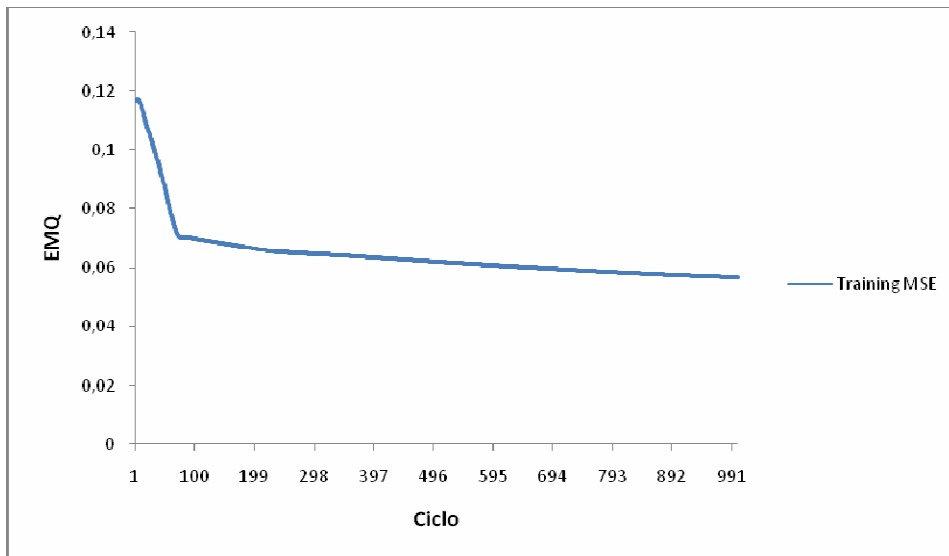


Figura 4.9. Rede Hopfield / temperatura–consumo
(Etapa de treino, Dias úteis)

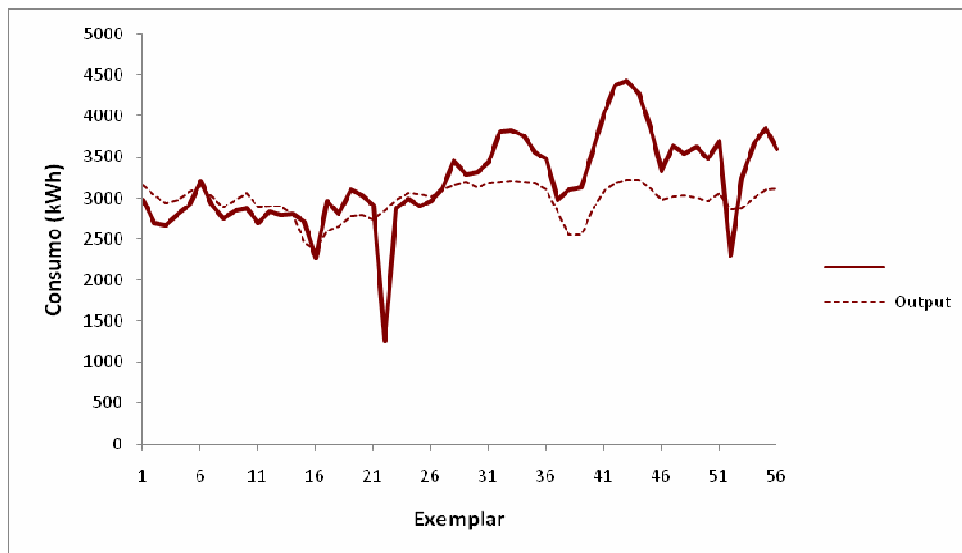


Figura 4.10. Rede Hopfield / temperatura–consumo
(Etapa de validação, Dias úteis)

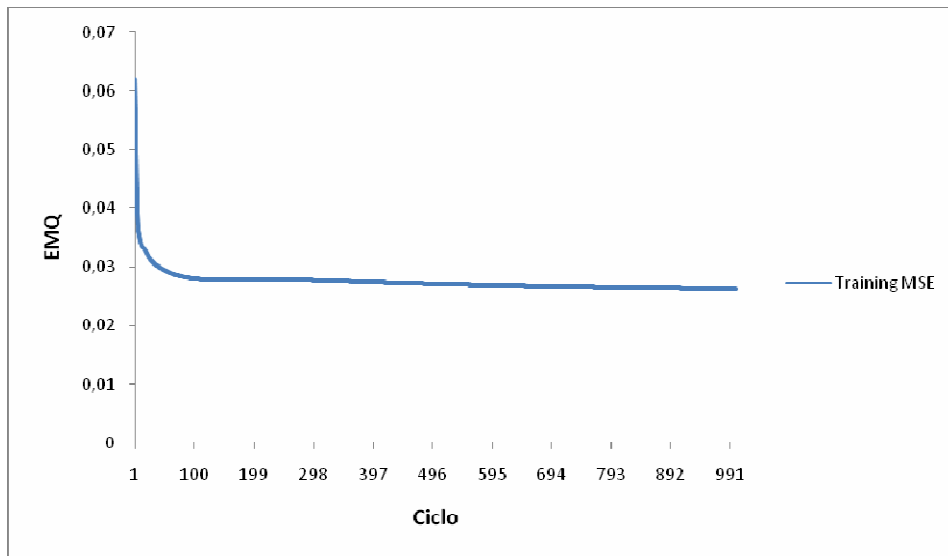


Figura 4.11. Rede Hopfield / temperatura–consumo
(Etapa de treino, Final de semana e Feriados)

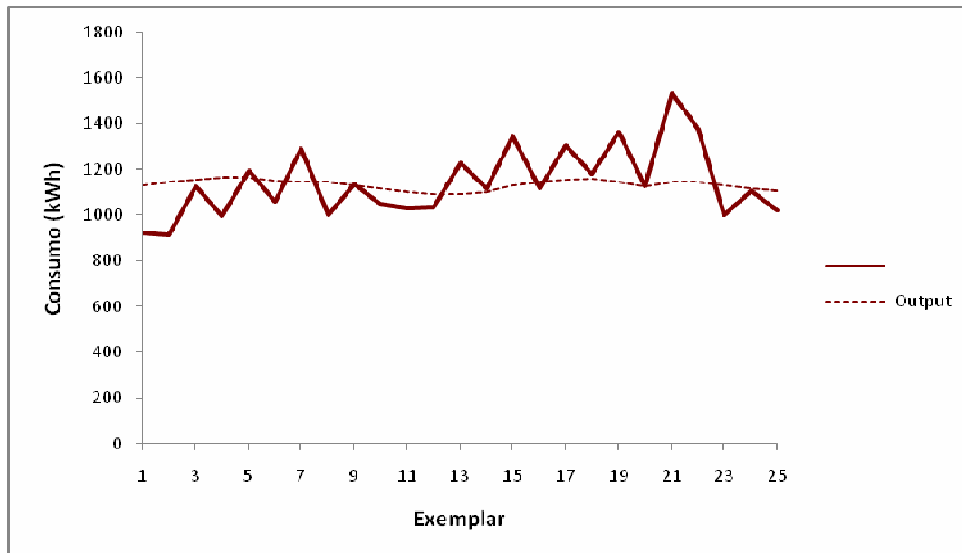


Figura 4.7. Rede Hopfield / temperatura–consumo
(Etapa de validação, Final de semana e Feriados)

4.3.3. Rede de Elman

Os parâmetros da rede Elman / temperatura–consumo são apresentados na Tab. 4.6 e seus resultados são resumidos na Tab. 4.7 e apresentados nas Figs. 4.13 e 4.14 para a rede “Todos os dias”, nas Figs. 4.15 e 4.16 para a rede “Dias úteis”, e nas Figs. 4.17 e 4.18 para a rede “Final de semana e Feriados”.

Tabela 4.6. Parâmetros da rede Elman / temperatura–consumo

Parâmetros	Todos os dias	Dias úteis	Finais de semana e feriados
Numero de camadas	3	3	3
Neurônios na camada de entrada	2	2	2
Neurônios na camada de saída	1	1	1
Neurônios na camada oculta	8	7	2
Número de ciclos de treino	1000	1000	1000

Tabela 4.7. Resultados da rede Elman / temperatura–consumo

Resultados	Todos os dias	Dias úteis	Finais de semana e Feriados
EMQ mínimo no treino	0,1172	0,0567	0,0264
EMQN	0,9165	0,5875	0,9463

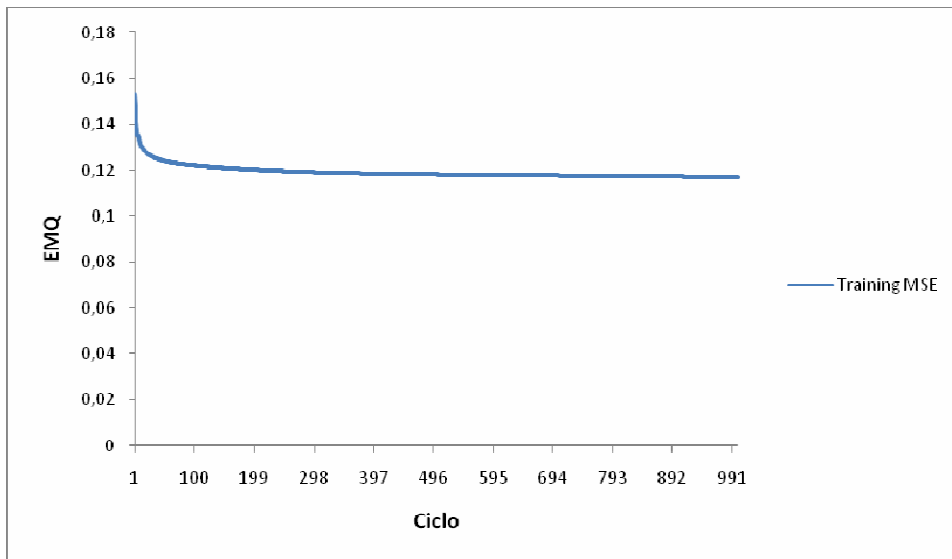


Figura 4.13. Rede Elman / temperatura–consumo
(Etapa de treino, Todos os dias)

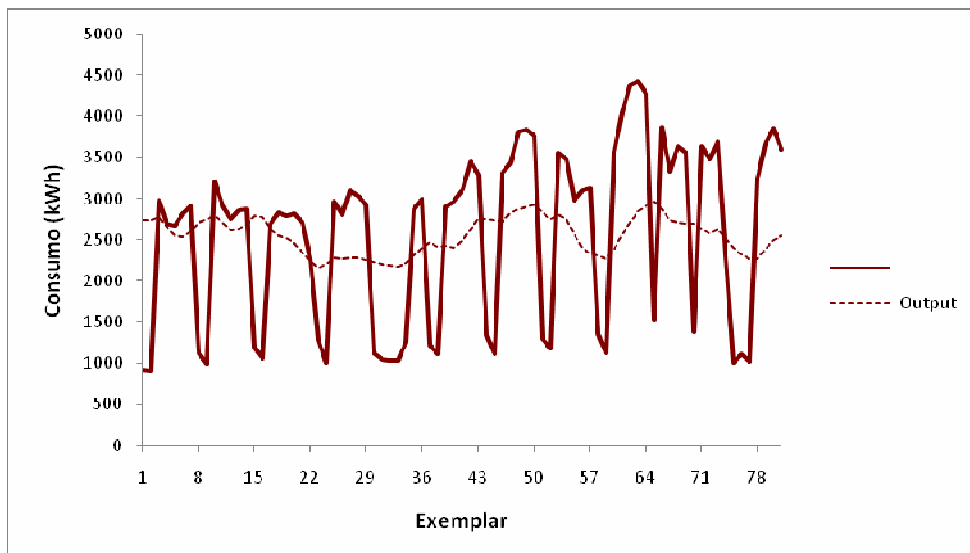


Figura 4.14. Rede Elman / temperatura–consumo
(Etapa de validação, Todos os dias)

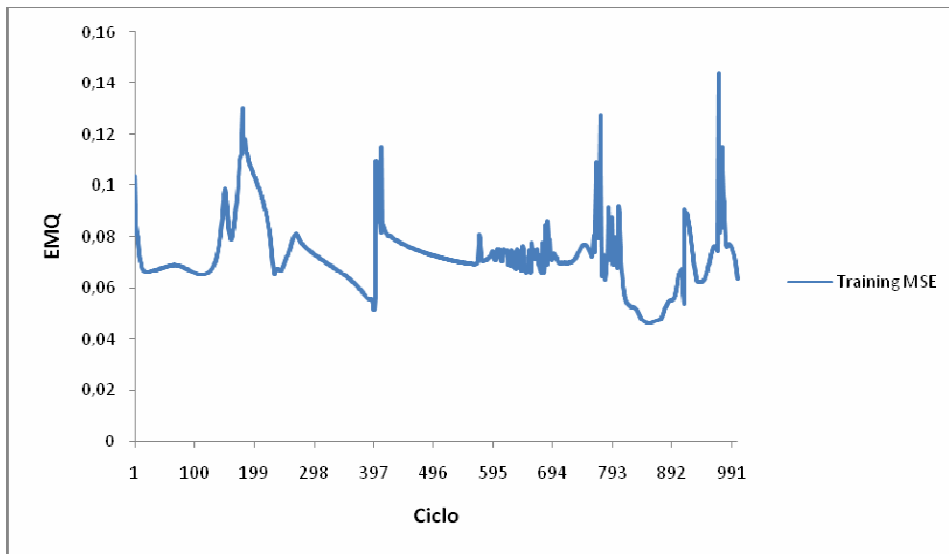


Figura 4.15. Rede Elman / temperatura–consumo
(Etapa de treino, Dias úteis)

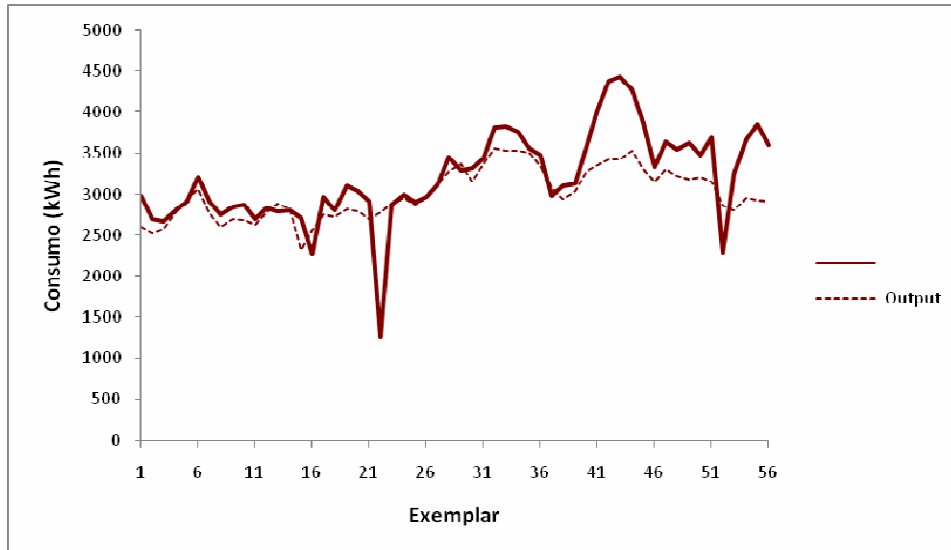


Figura 4.16. Rede Elman / temperatura–consumo
(Etapa de validação, Dias úteis)

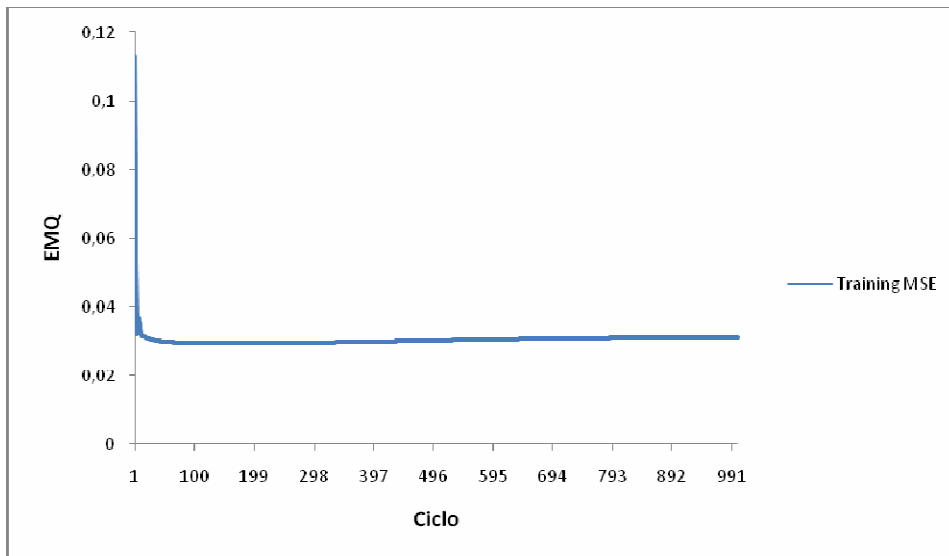


Figura 4.17. Rede Elman / temperatura–consumo
(Etapa de treino, Final de semana e Feriados)

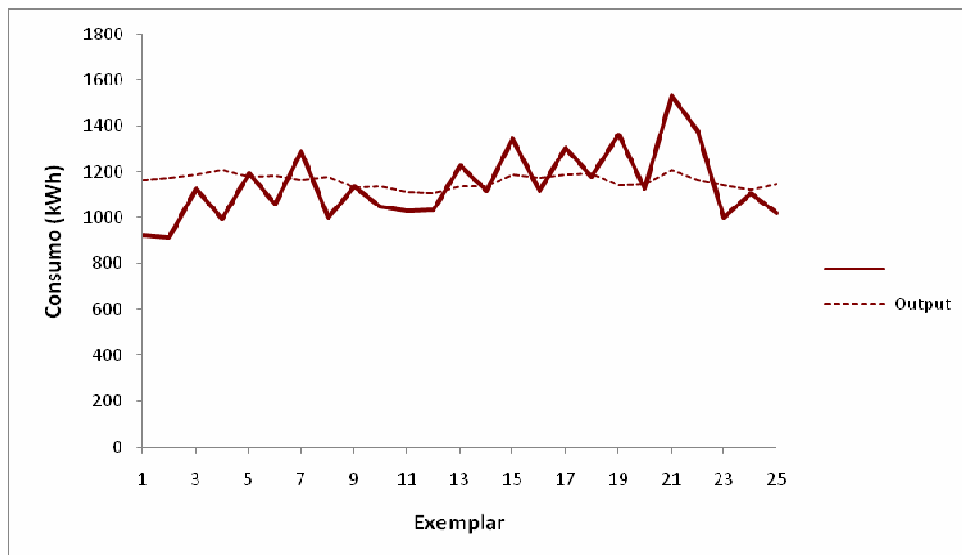


Figura 4.18. Rede Elman / temperatura–consumo
(Etapa de validação, Final de semana e Feriados)

4.3.4. Rede de Jordan

Os parâmetros da rede Jordan / temperatura–consumo são apresentados na Tab. 4.8 e seus resultados são resumidos na Tab. 4.9 e apresentados nas Figs. 4.19 e 4.20 para a rede “Todos os dias”, nas Figs. 4.21 e 4.22 para a rede “Dias úteis”, e nas Figs. 4.23 e 4.24 para a rede “Final de semana e Feriados”.

Tabela 4.8. Parâmetros da rede Jordan / temperatura–consumo

Parâmetros	Todos os dias	Dias úteis	Finais de semana e feriados
Numero de camadas	3	3	3
Neurônios na camada de entrada	2	2	2
Neurônios na camada de saída	1	1	1
Neurônios na camada oculta	8	4	4
Número de ciclos de treino	1000	1000	1000

Tabela 4.9. Resultados da rede Jordan / temperatura–consumo

Resultados	Todos os dias	Dias úteis	Finais de semana e Feriados
EMQ mínimo no treino	0,1219	0,0486	0,0296
EMQN	0,9433	0,8087	0,9492

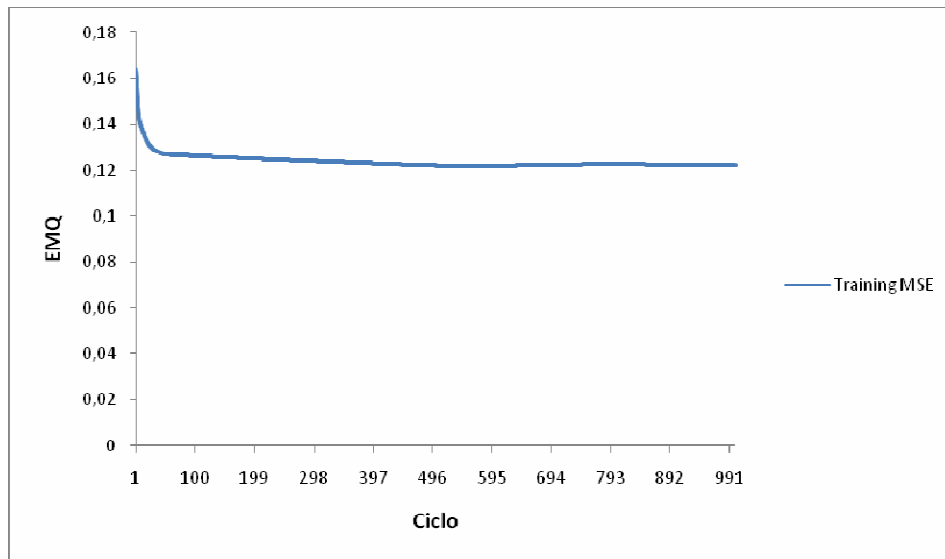


Figura 4.19. Rede Jordan / temperatura–consumo
(Etapa de treino, Todos os dias)

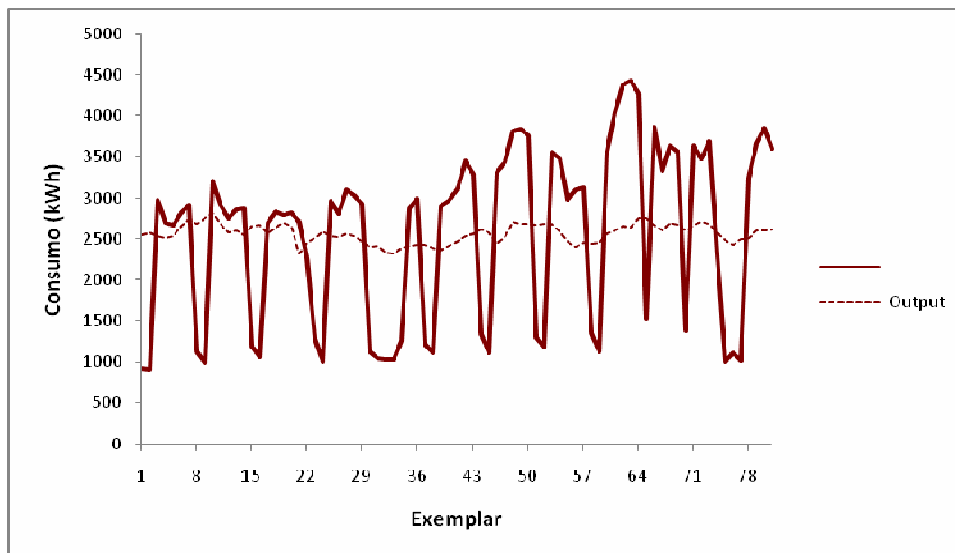


Figura 4.20. Rede Jordan / temperatura–consumo
(Etapa de validação, Todos os dias)

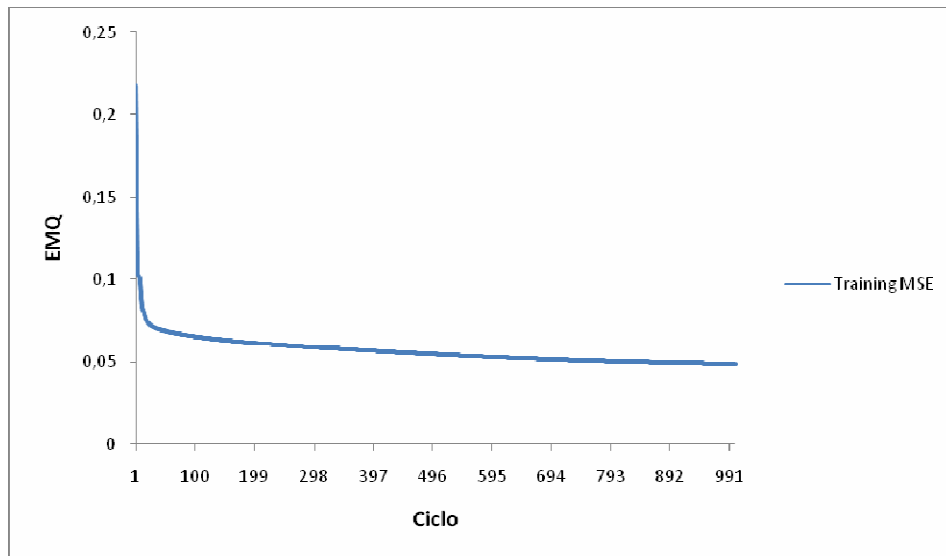


Figura 4.21. Rede Jordan / temperatura–consumo
(Etapa de treino, Dias úteis)



Figura 4.22. Rede Jordan / temperatura–consumo
(Etapa de validação, Dias úteis)

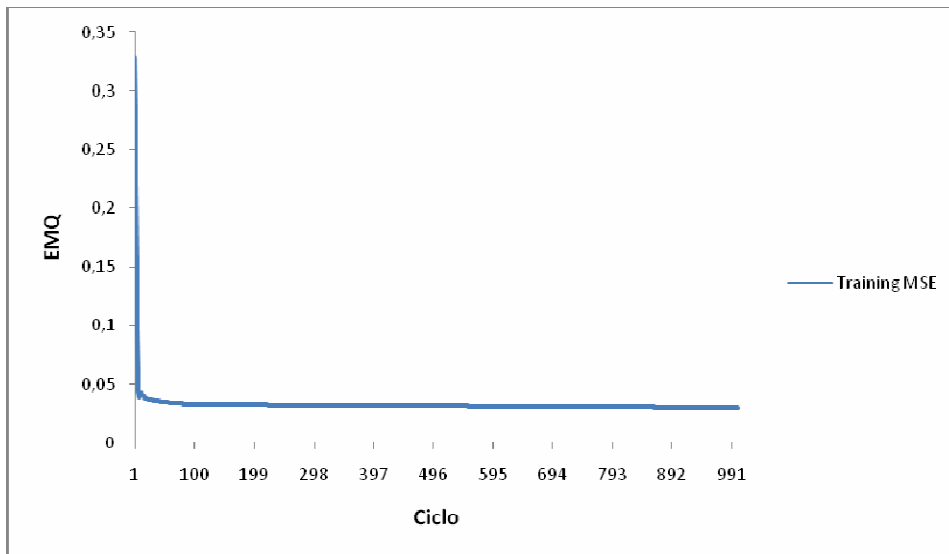


Figura 4.23. Rede Jordan / temperatura–consumo
(Etapa de treino, Final de semana e Feriados)

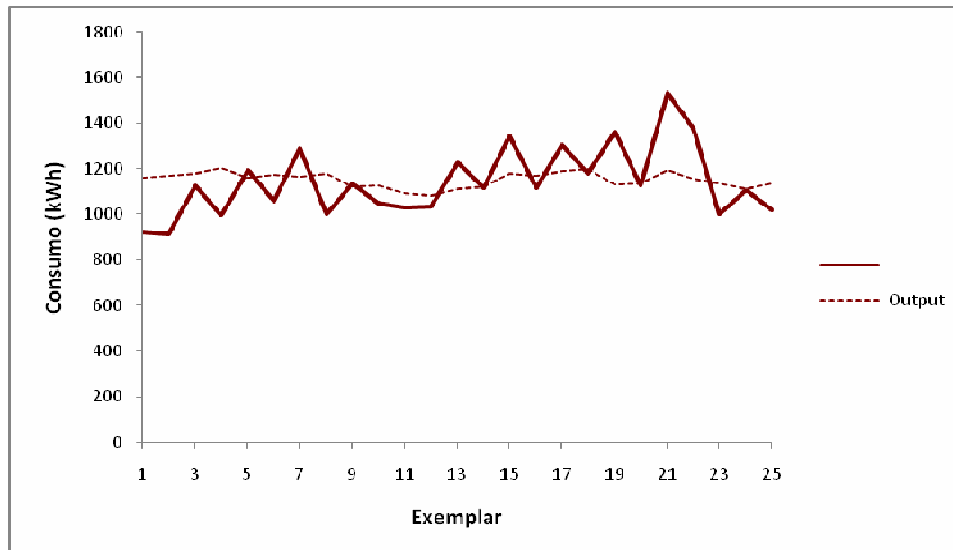


Figura 4.24. Rede Jordan / temperatura–consumo
(Etapa de validação, Final de semana e Feriados)

4.3.5. Mapas Auto-organizáveis (SOM)

Os parâmetros da rede SOM / temperatura–consumo são apresentados na Tab. 4.10 e seus resultados são resumidos na Tab. 4.11 e apresentados nas Figs. 4.25 e 4.26 para a rede “Todos os dias”, nas Figs. 4.27 e 4.28 para a rede “Dias úteis”, e nas Figs. 4.29 e 4.30 para a rede “Final de semana e Feriados”.

Tabela 4.10. Parâmetros da rede SOM / temperatura–consumo

Parâmetros	Todos os dias	Dias úteis	Finais de semana e feriados
Numero de camadas	4	3	3
Neurônios na camada de entrada	2	2	2
Neurônios na camada de saída	1	1	1
Neurônios na primeira camada oculta	8	4	2
Neurônios na segunda camada oculta	4	—	—
Dimensão do Plano de Neurônios	5 x 5	5 x 5	4 x 4
Número de ciclos de treino	1000	1000	1000

Tabela 4.11. Resultados da rede SOM / temperatura–consumo

Resultados	Todos os dias	Dias úteis	Finais de semana e Feriados
EMQ mínimo no treino	0,1169	0,0604	0,0292
EMQN	0,9332	0,6312	0,9514

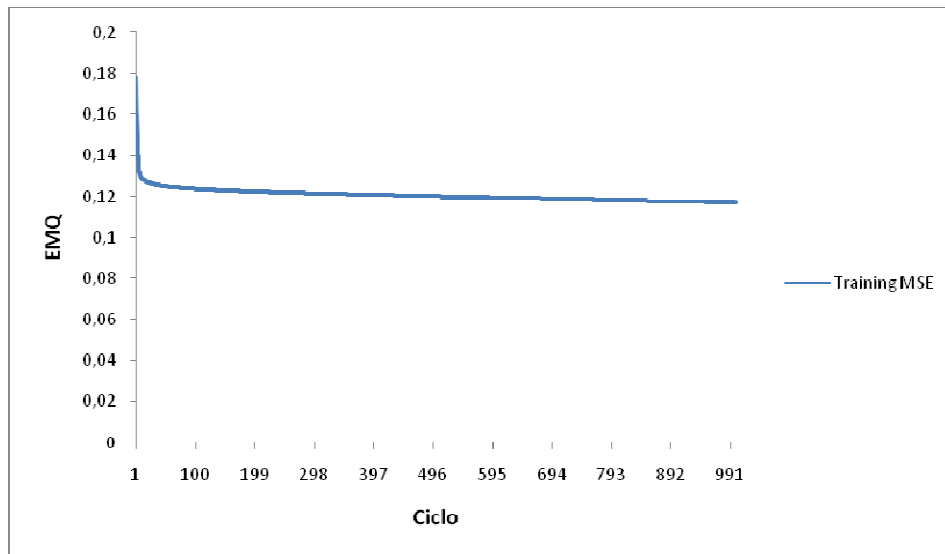


Figura 4.25. Rede SOM / temperatura–consumo
(Etapa de treino, Todos os dias)

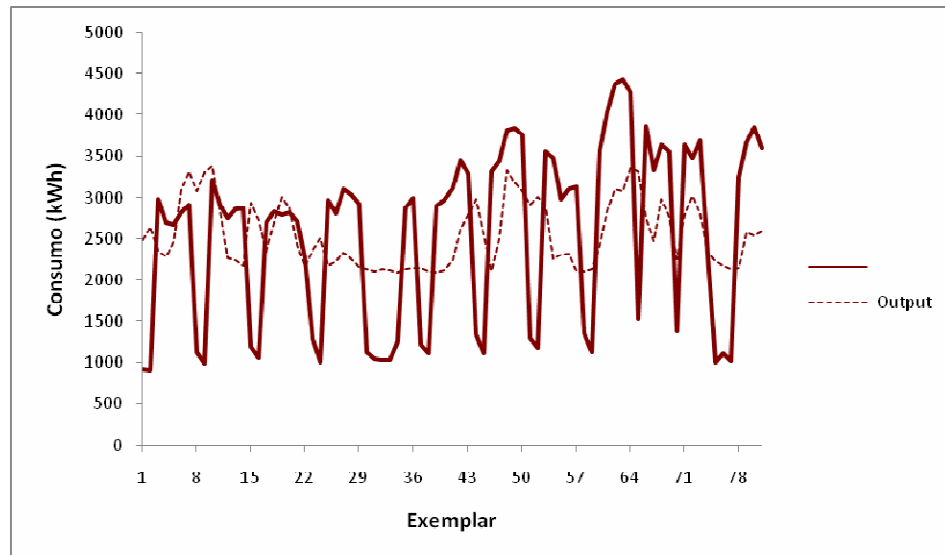


Figura 4.26. Rede SOM / temperatura–consumo
(Etapa de validação, Todos os dias)

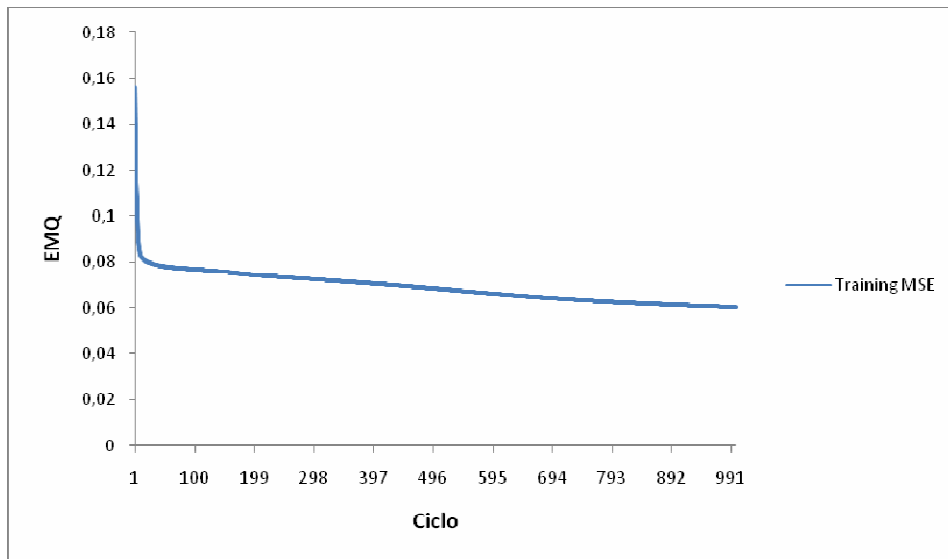


Figura 4.27. Rede SOM / temperatura–consumo
(Etapa de treino, Dias úteis)

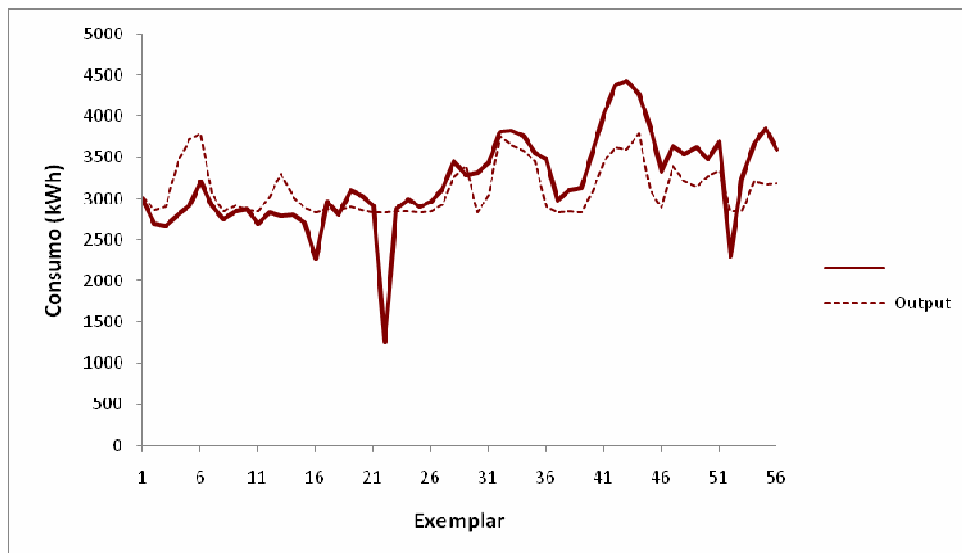


Figura 4.28. Rede SOM / temperatura–consumo
(Etapa de validação, Dias úteis)

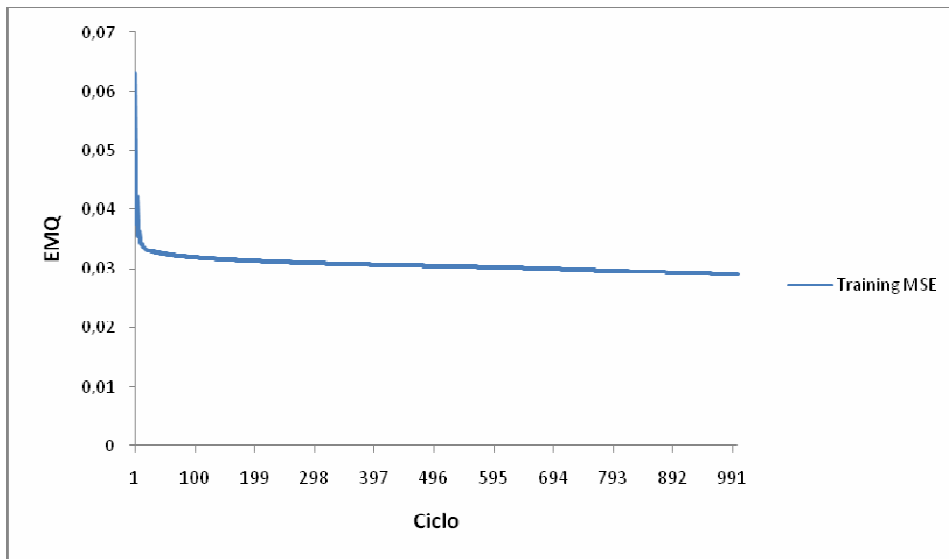


Figura 4.29. Rede SOM / temperatura–consumo
(Etapa de treino, Final de semana e Feriados)

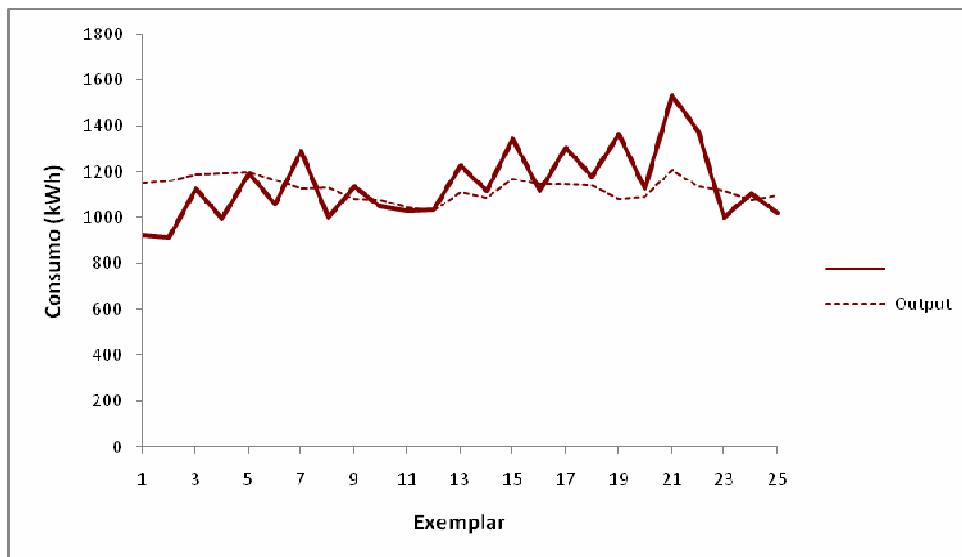


Figura 4.30. Rede SOM / temperatura–consumo
(Etapa de validação, Final de semana e Feriados)

4.3.6. Rede Modular

Os parâmetros da rede Modular / temperatura–consumo são apresentados na Tab. 4.12 e seus resultados são resumidos na Tab. 4.13 e apresentados nas Figs. 4.31 e 4.32 para a rede “Todos os dias”, nas Figs. 4.33 e 4.34 para a rede “Dias úteis”, e nas Figs. 4.35 e 4.36 para a rede “Final de semana e Feriados”.

Tabela 4.12. Parâmetros da rede Modular / temperatura–consumo

Parâmetros	Todos os dias	Dias úteis	Finais de semana e Feriados
Numero de camadas	4	4	4
Neurônios na camada de entrada	2	2	2
Neurônios na camada de saída	1	1	1
Número de camadas ocultas	2	2	2
Neurônios na primeira camada oculta	superior	5	4
	inferior	5	4
Neurônios na segunda camada oculta	superior	5	4
	inferior	5	4
Número de ciclos de treino	1000	1000	1000

Tabela 4.13. Resultados da rede Modular / temperatura–consumo

Resultados	Todos os dias	Dias úteis	Finais de semana e Feriados
EMQ mínimo no treino	0,1252	0,0781	0,0311
EMQN	0,9186	0,7432	0,9450

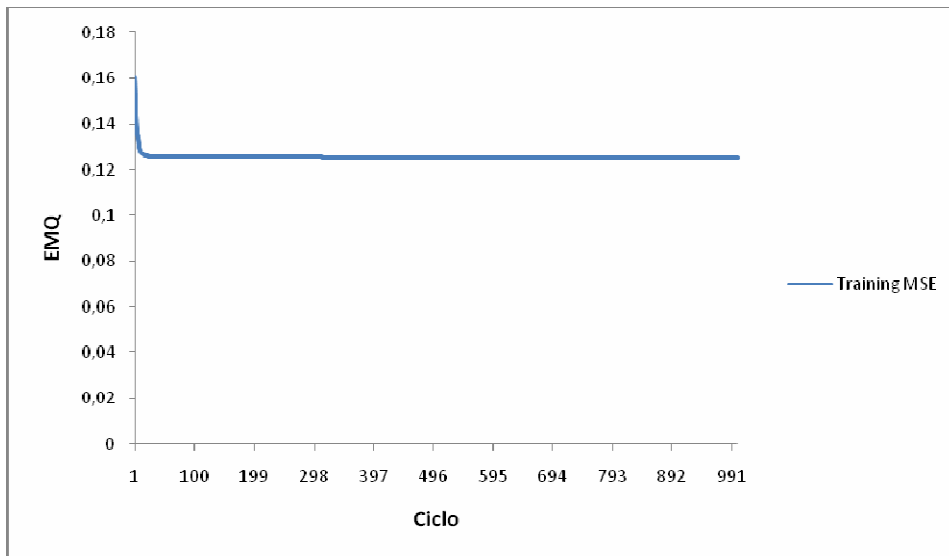


Figura 4.31. Rede Modular / temperatura–consumo
(Etapa de treino, Todos os dias)

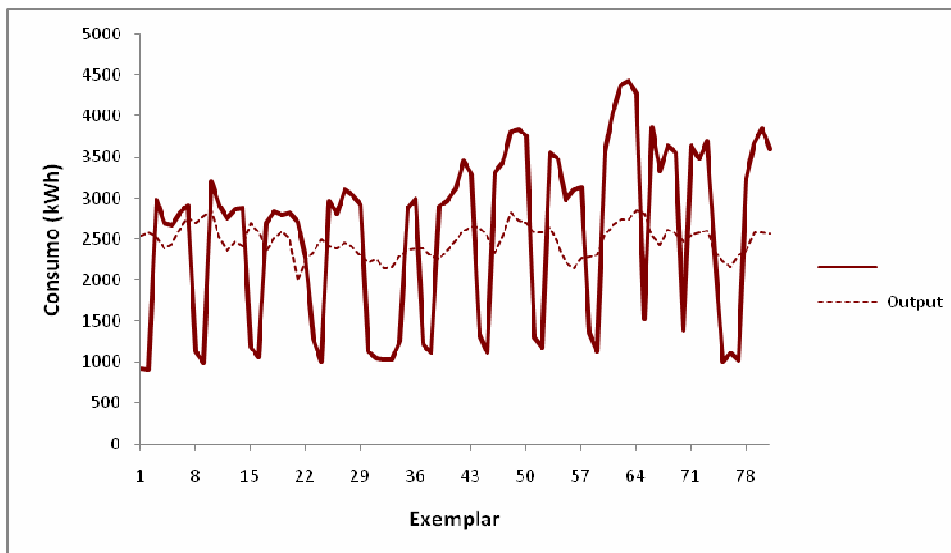


Figura 4.32. Rede Modular / temperatura–consumo
(Etapa de validação, Todos os dias)

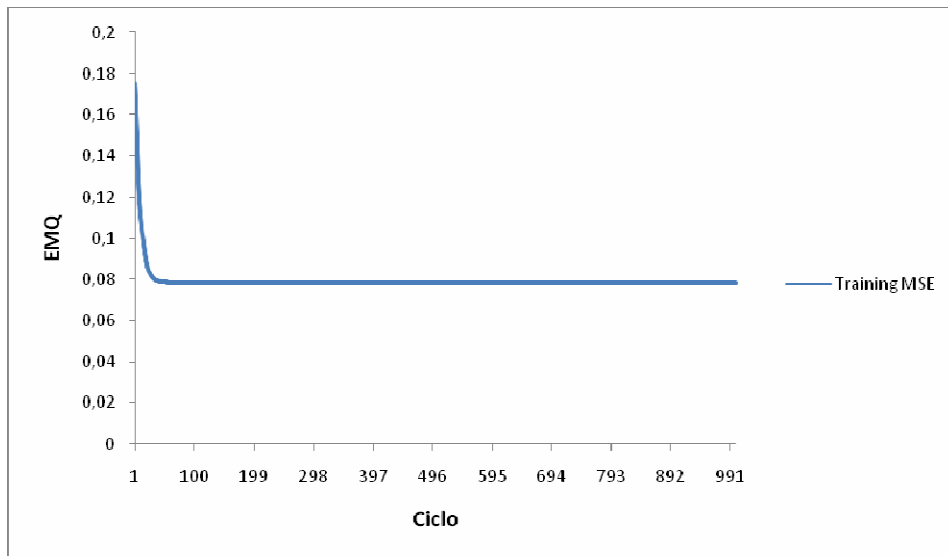


Figura 4.33. Rede Modular / temperatura–consumo
(Etapa de treino, Dias úteis)

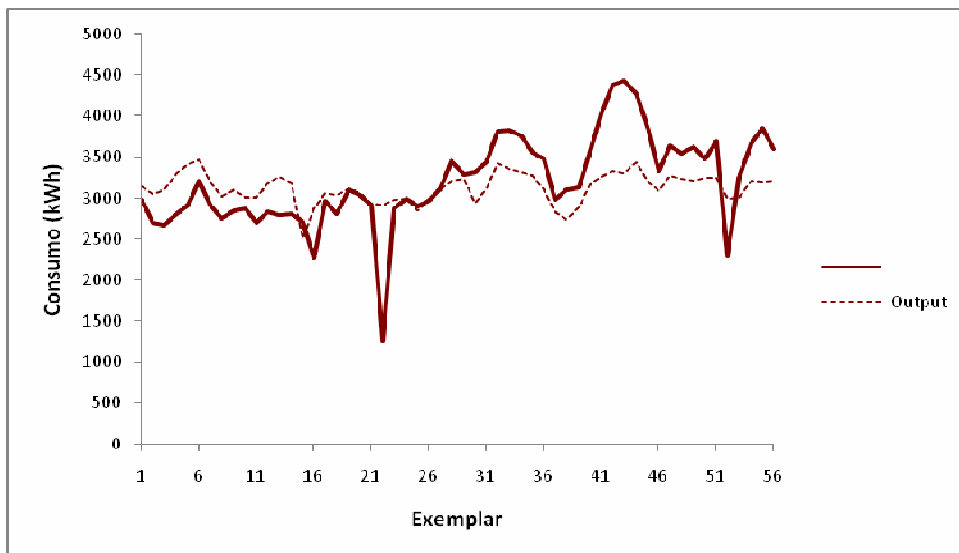


Figura 4.34. Rede Modular / temperatura–consumo
(Etapa de validação, Dias úteis)

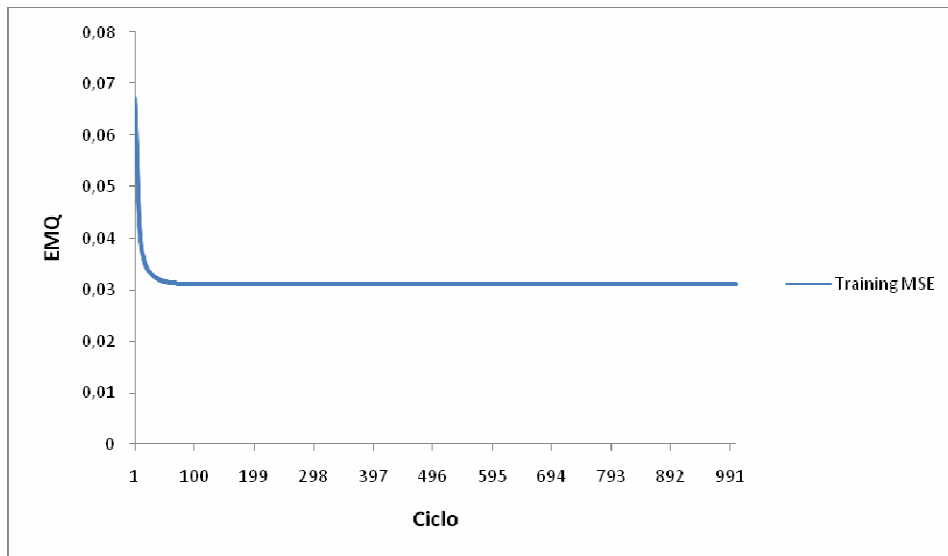


Figura 4.35. Rede Modular / temperatura–consumo
(Etapa de treino, Final de semana e Feriados)

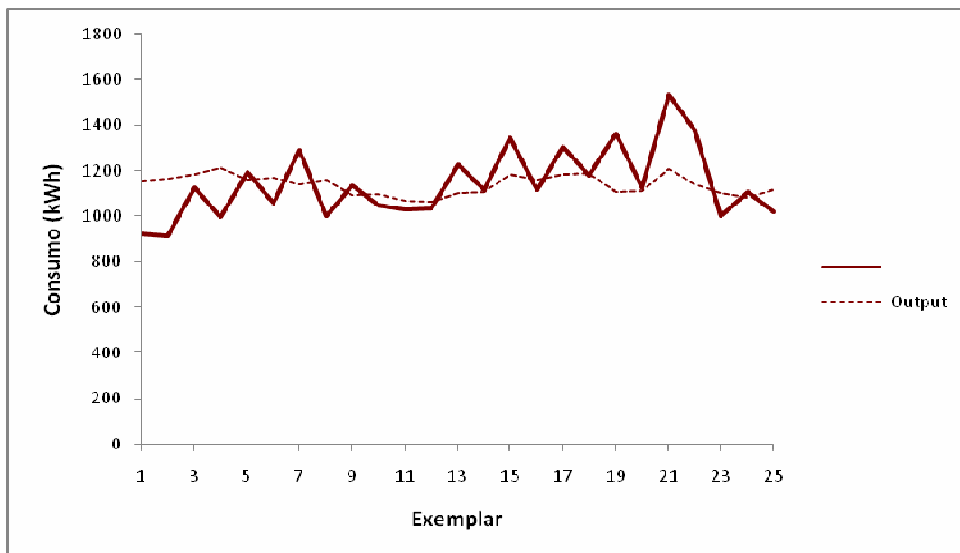


Figura 4.36. Rede Modular / temperatura–consumo
(Etapa de validação, Final de semana e Feriados)

4.3.7. Discussão dos Resultados

A Tab. 4.14 apresenta os valores dos erros quadráticos médios normalizados para as diversas arquiteturas de rede implementadas.

Tabela 4.14. Tabela comparativa dos valores de EQMN obtidos.

Rede	EQMN		
	Todos os dias	Dias úteis	Final de semana e Feriados
MLP	0,8897	0,6501	0,9372
Hopfield	0,9693	0,9053	0,9656
Elman	0,9165	0,5875	0,9463
Jordan	0,9433	0,8087	0,9492
SOM	0,9332	0,6312	0,9514
Modular	0,9187	0,7432	0,9450
Valor médio do EQMN	0,9284	0,7210	0,9491

As redes “Todos os Dias”, mesmo possuindo um grupo maior de dados para o treino e para a validação, apresentou variações bastante consideráveis, particularmente a rede de Hopfield. Uma das possíveis razões pela qual o nível de aprendizado foi baixo está relacionado com um padrão de dados de entrada muito disperso. O melhor resultado para esse tipo de rede foi obtido pela rede MLP.

As redes “Final de semana e Feriados” foram as que apresentaram maiores variações, mas isso é algo esperado, dada a baixa capacidade de consumo utilizada no prédio nesses dias, o que faz com que qualquer tipo de atividade fora do comum, seja refletida numa variação muito considerável na leitura de consumo energético, não necessariamente associada às condições climáticas.

Finalmente, as redes “Dias Úteis” apresentaram melhores resultados, com exceção da rede Hopfield. É importante destacar que a rede Hopfield é usualmente aplicada em campos como a percepção e reconhecimento de imagens e problemas de otimização, não sendo a priori adequada para esse tipo de aplicação, o que justifica o baixo desempenho dessa arquitetura de rede. Além disso, nesse caso em particular, em que se conta com uma amostra

relativamente pequena, a rede de Hopfield, considerada dentro do grupo de redes de memória associativa, não conseguiu achar algum tipo de padrão a seguir.

Por outro lado, os bons desempenhos obtidos pelas arquiteturas SOM e, em particular, Elman, indicam que elas são mais adequadas que a arquitetura MLP para esse tipo de rede, que seria a mais crítica em termos de previsão de consumo, uma vez que os maiores valores de consumo ocorrem nesse tipo de dia. No caso da arquitetura de Elman, o fato de a rede receber constantemente a informação de volta nos neurônios de entrada, desde as camadas intermediárias, lhe permite obter uma capacidade e velocidade de generalização maior do que uma MLP. Já para a arquitetura de Jordan, onde a informação só volta para as entradas da rede depois de ter chegado até a saída desta, a sua aprendizagem é mais lenta e a sua capacidade de generalização mais baixa.

Em relação às redes modulares, o fato de ser uma rede composta faz com que a sua capacidade de processamento seja muito maior em comparação a uma rede simples, já que a informação deve passar por mais estágios e elementos processadores, provavelmente essa mesma condição seja a que provoca um baixo desempenho em casos como o apresentado nesse trabalho.

Considerando alguns aspectos das configurações escolhidas, em particular a dos modelos que obtiveram mais baixo desempenho, uma recomendação seria usar um grupo de dados maior e mais compacto (sem “buracos”) em comparação aos usados nessa pesquisa, tanto para treino quanto para validação. Provavelmente esse aumento fará com que redes como a de Jordan ou a rede Modular consigam chegar a níveis de generalização aceitáveis.

Capítulo 5

Conclusões

Considerando as arquiteturas de rede implementadas, no caso particular da rede “Dias Úteis” a rede de Elman teve um desempenho consideravelmente melhor do que a rede MLP, com erros quadráticos médios aproximadamente 10% mais baixos, o que faz desse modelo de rede parcialmente recorrente uma melhor alternativa para trabalhar em problemas similares ao apresentado nesse trabalho. Contudo, a rede MLP não deixa de ser uma alternativa viável para trabalhar em situações similares, e em ambos casos deve se considerar o margem de erro correspondente. As demais arquiteturas, com exceção da SOM para a rede “Dias úteis”, apresentaram desempenhos piores que o da MLP, motivo pelo qual não se recomenda nesse caso.

A diferença significativa no desempenho das redes parcialmente recorrentes, em particular a rede de Elman, com a totalmente recorrente (Hopfield), ressalta que as diferentes configurações de redes possuem características particulares, tornando-as adequadas a um grupo específico de situações.

Por fim, dado que mesmo na etapa de treino, houve variações consideráveis nos erros médios, pode-se concluir que o aprendizado das redes apresentou dificuldades, provavelmente pela falta excessiva de dados, o que faz que a redes percam a sua capacidade de generalizar de forma eficiente.

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.

Aquino, R. 2005. **Gestão de manutenção de condicionadores de ar do tipo janela e split**. Trabalho de Conclusão de Curso, Escola Politécnica da USP, São Paulo.

Campoleoni, E.T. 2006. **Rede Neural Artificial para Previsão de Consumo de Energia**. Trabalho de Conclusão de Curso, Escola Politécnica da USP, São Paulo.

NEUROSOLUTIONS, **Neurosolutions Programa e Tutorial**. Disponível em: <http://www.neurosolutions.com/>. Último acesso 28/10/2008