

MODELOS NEURAI AUTÔNOMOS PARA PREVISÃO DE CARGA ELÉTRICA

VITOR HUGO FERREIRA, ALEXANDRE P. ALVES DA SILVA

Laboratório de Sistemas de Potência - LASPOT, Departamento de Engenharia Elétrica, Universidade Federal do Rio de Janeiro

COPPE/UFRJ, 21945-970, Rio de Janeiro, RJ, Brasil.

E-mails: vitor@vishnu.coep.ufrj.br, alex@coep.ufrj.br

Abstract— After 1991, the literature on load forecasting has been dominated by neural network based proposals. However, one major risk in using neural models is the possibility of excessive training data approximation, i.e., *overfitting*. The extent of nonlinearity provided by neural network based load forecasters, which depends on the input space representation, has been adjusted using heuristic procedures. The empirical nature of these procedures makes their application cumbersome and time consuming. This paper explores the most promising training methods proposed in recent years for avoiding the above mentioned drawbacks, considering automatic and simultaneous complexity control and input selection of neural network based load forecasters.

Keywords— Load forecasting, artificial neural networks, complexity control, input selection, Bayesian methods, support vector machines.

Resumo— Após 1991, a literatura sobre previsão de carga passou a ser dominada por propostas baseadas em modelos neurais. Entretanto, um empecilho na aplicação destes modelos reside na possibilidade do ajuste excessivo dos dados, i.e., *overfitting*. O excesso de não-linearidade disponibilizado pelos modelos neurais de previsão de carga, que depende da representação do espaço de entrada, vem sendo ajustado de maneira heurística. Este trabalho explora algumas das técnicas mais promissoras para abordagem das questões anteriores, considerando técnicas automáticas e simultâneas de controle de complexidade e seleção de variáveis de entrada de modelos neurais para previsão de carga.

Palavras-chave— Previsão de carga, redes neurais artificiais, controle de complexidade, seleção de entradas, métodos bayesianos, máquinas de vetor suporte.

1 Introdução

A tomada de decisão na operação de sistemas de potência, abrangendo desde o despacho econômico até a comercialização de energia, depende do conhecimento do comportamento futuro da carga (Debs, 1988). Vários métodos para previsão de carga vêm sendo propostos, baseados em técnicas como regressão múltipla, método de Box-Jenkins, redes neurais artificiais (RNAs), sistemas difusos e modelos híbridos. Entretanto, modelos autônomos de previsão são necessários, visando reduzir a necessidade de intervenção de especialistas e estender a aplicação destes modelos ao nível de barramento.

O relacionamento entre a carga e seus fatores exógenos é complexo e não-linear, dificultando a aplicação de técnicas como análise de séries temporais e regressão linear. Por outro lado, a experiência tem mostrado que as RNAs apresentam desempenho superior quando aplicadas a problemas multivariados envolvendo bases de dados de elevada cardinalidade, como é o caso da previsão de carga.

Mesmo sendo mais robustas, as RNAs vêm enfrentando alguns problemas antes de apresentarem sucesso comercial (Khotanzad et.al., 1998). Desde as primeiras propostas, empecilhos como elevado custo computacional, ausência de intervalos de confiança e baixa interpretabilidade vêm sendo abordados. Entre-

tanto, questões críticas como a representação do espaço de entrada e o controle de complexidade do modelo ainda não mereceram a devida atenção.

A seleção de entradas constitui uma das tarefas mais importantes na previsão de carga. Técnicas lineares de seleção de variáveis de entrada não são apropriadas a modelos não-lineares. Técnicas não-lineares de extração de características (Reis e Alves da Silva, 2005) utilizam somente informação da série em estudo, sendo necessária uma abordagem mais orientada a modelos neurais.

O controle de complexidade de RNAs visa adequar o nível de não-linearidade disponibilizado à regularidade apresentada pelos dados, evitando a modelagem indesejada do ruído e a conseqüente redução da capacidade de generalização do modelo. O procedimento mais popular de regularização de RNAs aplicadas à previsão de carga é a parada antecipada do treinamento. Além do excessivo empirismo, este tipo de método apresenta uma série de desvantagens teóricas, conforme destacam (Cataltepe et. al., 1999).

Este trabalho explora técnicas automáticas de controle de complexidade e seleção de entradas de modelos neurais de previsão de carga. Especificamente, são investigados o treinamento bayesiano de perceptrons de múltiplas camadas (MLPs) (Mackay, 1992) e a minimização de limites superiores do erro de generalização de máquinas de vetor suporte (SVMs) (Chang e Lin, 2005).

2 Redes neurais artificiais

Seja $\underline{x} \in \mathbb{R}^n$ o vetor contendo os sinais de entrada e $\underline{w} \in \mathbb{R}^M$ o vetor com todos os pesos e bias da RNA, sendo $M = mn + 2m + 1$, com m respondendo pelo número de neurônios na camada oculta. Os bias dos neurônios sigmoidais da camada oculta são representados por b_k , com b sendo o bias do neurônio linear da saída, dada por:

$$y = f(x, \underline{w}) = \sum_{k=1}^m \left[w_k \varphi \left(a_k \sum_{i=1}^n (w_{ik} x_i) + b_k \right) \right] + b \quad (1)$$

Usualmente, dado um conjunto U contendo N pares entrada/saída, $U = \{X, D\}$, $X = (x_1, \dots, x_N)$, $D = (d_1, \dots, d_N)$, $d_j \in \mathbb{R}$ sendo a saída desejada, o objetivo do treinamento de RNAs reside na estimação do vetor \underline{w} através da minimização do risco empírico dado por

$$\min_{\underline{w}} \left\{ E_s(\underline{w}, U) = \frac{1}{2} \sum_{j=1}^N [d_j - f(x_j, \underline{w})]^2 \right\} \quad (2)$$

Utilizando o algoritmo clássico de retropropagação do erro, ou até mesmo métodos de segunda ordem como Levenberg-Marquardt (Bishop, 1995), a principal desvantagem da minimização irrestrita de $E_s(\underline{w}, U)$ reside na ausência de preocupação com a complexidade do modelo estimado.

Existem duas abordagens gerais para controle de complexidade de RNAs. A primeira é conhecida como estabilização de estrutura, que busca determinar o número mínimo de neurônios na camada oculta que atenda as especificações de desempenho do problema. Esta técnica inclui algoritmos de poda de rede (Treadgold e Gedeon, 1999) e comparação analítica entre modelos (Murata et al., 1994).

O segundo método geral de controle de complexidade tem por base a teoria da regularização, onde técnicas analíticas buscam ajustar o grau de não-linearidade cedido pela RNA, conforme apresentado a seguir.

3 Teoria da regularização

Na teoria da regularização, o compromisso entre o ajuste dos dados de treinamento e a capacidade de generalização é relativizado através da minimização do risco total:

$$\min_{\underline{w}} \{ R(\underline{w}) = E_s(\underline{w}, U) + \lambda E_c(\underline{w}) \} \quad (3)$$

Na equação (3), $E_s(\underline{w}, U)$ denota o risco empírico, dado por (2), enquanto $E_c(\underline{w})$ responde pelo controle de complexidade do modelo. O fator λ é conhecido como parâmetro de regularização, que pode ser ajustado através de técnicas de re-amostragem ou utilizando inferência bayesiana.

3.1 Treinamento bayesiano de MLPs

O funcional $E_c(\underline{w})$ na equação (3) pode ser definido através da aplicação de inferência bayesiana. Através da regra de Bayes, a função densidade de probabilidade a posteriori de \underline{w} é dada por:

$$p(\underline{w}|D, X) = \frac{p(D|\underline{w}, X) p(\underline{w}|X)}{p(D|X)} \quad (4)$$

Visto que X condiciona todas as probabilidades em (4), esta variável será omitida a partir deste ponto. Assim, na equação (4), $p(D|\underline{w})$ é a verossimilhança de D dado \underline{w} , $p(\underline{w})$ a probabilidade a priori de \underline{w} e $p(D) = \int p(D|\underline{w}) p(\underline{w}) d\underline{w}$.

Assumindo que \underline{w} possui uma distribuição gaussiana com vetor média nulo e matriz de covariância diagonal igual a $\alpha^{-1} \underline{I}$, \underline{I} igual à matriz identidade de dimensão $M \times M$, e que as saídas desejadas estão corrompidas com ruído branco gaussiano com variância β^{-1} , ou seja, $d_j = f(x_j, \underline{w}) + \zeta_j$, a aplicação da equação (4) resulta em:

$$p(\underline{w}|D) = \frac{e^{[-S(\underline{w})]}}{\int e^{-S(\underline{w})} d\underline{w}} \quad (5)$$

onde

$$S(\underline{w}) = \frac{\beta}{2} \sum_{j=1}^N [d_j - f(x_j, \underline{w})]^2 + \frac{\alpha}{2} \sum_{i=1}^M w_i^2 \quad (6)$$

Desta forma, a maximização da probabilidade a posteriori de \underline{w} é equivalente à minimização de $S(\underline{w})$. Dividindo $S(\underline{w})$ por β e fazendo $\lambda = \alpha/\beta$,

$$E_c(\underline{w}) = \frac{1}{2} \|\underline{w}\|^2 \quad (7)$$

Uma das vantagens do treinamento bayesiano reside na estimativa do parâmetro de regularização λ , através de um processo iterativo de estimação dos hiperparâmetros α e β ao longo do treinamento. Maiores detalhes sobre o processo de estimação de α e β podem ser encontrados em (Mackay, 1992) e (Bishop, 1995).

Para problemas multivariados, a utilização de um único hiperparâmetro α para todos os pesos não é recomendável. Neste trabalho, os pesos que ligam cada entrada aos neurônios da camada oculta são agrupados, com cada grupo possuindo o seu respectivo α_i . A mesma idéia é aplicada aos demais pesos, sendo reunidos os bias dos neurônios da camada oculta, os pesos que ligam a camada oculta à saída e o bias do neurônio de saída, sendo utilizados $n+3$ grupos de pesos. Neste caso, $S(\underline{w})$ passa a ser dado por:

$$S(\underline{w}) = \frac{\beta}{2} \sum_{j=1}^N [d_j - f(x_j, \underline{w})]^2 + \frac{1}{2} \sum_{i=1}^{n+3} \alpha_i \sum_{l=1}^M w_{il}^2 \quad (8)$$

3.2.1 Determinação automática de relevância

A magnitude dos α_i 's relacionados às entradas pode ser utilizada para mensuração da relevância de cada sinal. Sob o ponto de vista de otimização, observando a equação (8), valores elevados para α_i resultam na redução da magnitude de $\underline{w}_i \in \mathbb{R}^{M_i}$, vetor representando o conjunto contendo os M_i pesos associados ao i -ésimo grupo. Desta forma, quanto maior α_i , menor a magnitude de \underline{w}_i e conseqüentemente menor a relevância da entrada no cálculo da saída. Esse procedimento é conhecido como determinação automática de relevância (DAR) (Bishop, 1995).

Para seleção de entradas, uma referência para α_i é necessária para determinação das variáveis que podem ser consideradas irrelevantes. Para previsão de carga, duas referências são necessárias, uma para as variáveis contínuas e outra para as binárias. Seguindo (Stoppiglia et. al., 2003), são inseridas no conjunto original duas variáveis irrelevantes. A primeira, contínua, obtida segundo uma distribuição uniforme definida no intervalo de normalização das variáveis originais, e a segunda, discreta, gerada através de uma distribuição uniforme binária. Após o treinamento do modelo com este conjunto estendido de entradas, as variáveis contínuas e discretas são ordenadas decrescentemente segundo as magnitudes de α_i , com as variáveis abaixo do respectivo nível de relevância sendo descartadas. O modelo alimentado somente com as entradas relevantes é então treinado finalmente.

3.2.2 Seleção de estrutura

Inferência bayesiana também pode ser utilizada para seleção da melhor estrutura em uma série de hipóteses $H = \{H_1, H_2, \dots, H_K\}$, com o conjunto de variáveis relevantes de cada hipótese previamente selecionado. Pela regra de Bayes, a probabilidade a posteriori de H_h é dada por:

$$P(H_h|D) = \frac{p(D|H_h)P(H_h)}{p(D)} \quad (9)$$

Como todas as hipóteses H_h são equiprováveis a priori, a evidência (Bishop, 1995) de cada hipótese, $p(D|H_h)$, pode ser utilizada para seleção de modelos. Utilizando uma aproximação gaussiana em torno dos hiperparâmetros obtidos no treinamento, é obtida a seguinte expressão para $\ln p(D|H_h)$:

$$\ln p(D|H_h) = -S(\underline{w}) - \frac{1}{2} \ln |A(\underline{w})| + \frac{1}{2} \sum_{i=1}^{n+3} M_i \alpha_i \quad (10)$$

$$+ \ln(\beta^2 m^2 m!) + \frac{1}{2} \sum_{i=1}^{n+3} \ln \left(\frac{2}{\gamma_i} \right) + \frac{1}{2} \ln \left(\frac{2}{N - \gamma} \right)$$

4 Máquina de Vetor Suporte

Desenvolvidas para problemas de classificação, as SVMs buscam a maximização da margem de separação. Os padrões de treinamento que definem a margem máxima são denominados vetores suporte.

Para problemas de regressão, o conceito de margem deve ser adaptado. Desta forma, durante o treinamento, erros de aproximação localizados dentro da banda definida por ε não são considerados. Os padrões localizados fora desta banda contribuem para a estimação do modelo, sendo estes os vetores suporte.

A saída de uma SVM é dada por:

$$y = \sum_{j=0}^m W_j \phi_j(\underline{x}) = \underline{W}' \underline{\phi}(\underline{x}) \quad (11)$$

Na equação (11), $\underline{\phi}(\underline{x}) = [1, \phi_1(\underline{x}), \dots, \phi_m(\underline{x})]'$ e

$\underline{W} = [b, W_1, \dots, W_m]'$. Para definição da banda, é utilizada a função quadrática de perda com tolerância ε :

$$L_\varepsilon(d, y) = \begin{cases} (d - y)^2 - \varepsilon, & \text{para } |d - y| - \varepsilon \geq 0 \\ 0, & \text{para } |d - y| - \varepsilon < 0 \end{cases} \quad (12)$$

SVMs que utilizam a equação (12) são conhecidas como L2-SVM (Chang e Lin, 2005), contrastando com as SVMs comumente utilizadas em previsão de carga, que utilizam a função linear de perda (Ferreira e Alves da Silva, 2005). L2-SVMs apresentam limites superiores diferenciáveis para o erro de generalização, motivando assim a sua utilização no trabalho.

Supondo que c_0 e ε são constantes definidas pelo usuário, o treinamento de SVMs visa à minimização restrita do risco empírico dado por:

$$\min_{\underline{W}} \left\{ E_\varepsilon(\underline{W}, D) = \frac{1}{N} \sum_{i=1}^N L_\varepsilon(d_i, y_i) \right\} \quad (13)$$

s.a.

$$\|\underline{W}\|^2 \leq c_0$$

A equação (13) é a formulação primal do treinamento de SVMs para regressão. A versão dual deste problema incorpora os kernels $K(\underline{x}_i, \underline{x}_j)$ do produto interno $\underline{\phi}(\underline{x}_i)' \underline{\phi}(\underline{x}_j)$, podendo ser formulada da maneira que segue (Vapnik, 1998):

$$\max \left\{ Q(\underline{\alpha}, \underline{\alpha}') = \sum_{i=1}^N d_i (\alpha_i - \alpha'_i) - \varepsilon \sum_{i=1}^N (\alpha_i + \alpha'_i) \right\} \quad (14)$$

$$- \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha'_i)(\alpha_j - \alpha'_j) \left[K(\underline{x}_i, \underline{x}_j) + \frac{\delta_{ij}}{C} \right]$$

s.a.

$$\sum_{i=1}^N (\alpha_i - \alpha'_i) = 0, \quad \alpha_i \geq 0, \quad \alpha'_i \geq 0, \quad i = 1, 2, \dots, N$$

Na equação acima, δ_{ij} é a função delta de Kronecker e C é o parâmetro de regularização, relacionado com c_0 e responsável pelo controle de complexidade

do modelo. Assim, a saída da SVM passa a ser dada por:

$$y = f(\underline{x}, \underline{W}) = \sum_{i=1}^N (\alpha_i - \alpha'_i) K(\underline{x}, \underline{x}_i) \quad (15)$$

Os padrões para os quais $\alpha_i \neq \alpha'_i$ definem a saída dada pela equação (15), sendo estes os chamados vetores suporte. Assim, SVMs podem ser vistas como RNAs alimentadas adiante com uma única camada oculta contendo neurônios definidos por $K(\underline{x}, \underline{x}_i)$.

4.1 Limites superiores do erro de generalização

Utilizando o conceito de extensão dos vetores suporte (Chapelle et. al., 2002), (Chang e Lin, 2005) desenvolveram um limite superior diferenciável para o erro de generalização de SVMs aplicadas a regressão, dado por

$$T_{SB} [f(\underline{x}, \underline{w})] = \sum_{i=1}^p (\alpha_i + \alpha'_i) \tilde{S}_i^2 + N\varepsilon \quad (16)$$

onde α_i e α'_i são os multiplicadores de Lagrange associados ao vetor suporte \underline{x}_i , p é igual ao número de vetores suporte e \tilde{S}_i^2 é dado por:

$$S_i^2 = \min_{\underline{\mu}} \left\| \tilde{\phi}(\underline{x}_i) - \sum_{j=1, j \neq i}^p \mu_j \tilde{\phi}(\underline{x}_j) \right\|^2 + \eta \sum_{j=1, j \neq i}^p \frac{\mu_j^2}{(\alpha_i + \alpha'_i)} \quad (17)$$

s.a

$$\sum_{j=1, j \neq i}^p \mu_j = 1, \text{ for } \mu_j \in \mathbb{R}$$

Em (17), η é um parâmetro não-nulo responsável pela diferenciabilidade de \tilde{S}_i^2 , e $\tilde{\phi}(\underline{x}_j) = [\phi(\underline{x}_j) \quad \underline{c}_j / \sqrt{C}]^t$ representa um mapeamento estendido do espaço de características, com $\underline{c}_j \in \mathbb{R}^N$ sendo um vetor nulo exceto pela sua j -ésima componente, feita igual a 1.

A solução do problema (17) está apresentada em (Chang e Lin, 2005), juntamente com as derivadas parciais do limite dado por (16) em relação a C , ε e aos parâmetros do kernel.

4.2 Seleção dos parâmetros da SVM

A minimização de (16) utilizando descida em gradiente é utilizada neste trabalho para seleção dos parâmetros da SVM, ou seja, C , ε e os parâmetros σ_i do kernel. Ponderadores σ_i das entradas podem ser utilizados para mensurar a significância de cada sinal na estimativa da saída. Isto pode ser verificado se o kernel gaussiano $K(\underline{x}, \underline{x}_i)$ for escrito da seguinte forma:

$$K(\underline{x}, \underline{y}) = e^{-\sum_{i=1}^n \sigma_i^2 (x_i - y_i)^2} = e^{-\sum_{i=1}^n (\sigma_i x_i - \sigma_i y_i)^2} \quad (18)$$

Variáveis com σ_i de pequena magnitude apresentam pequena contribuição para o cálculo da saída. Assim, um procedimento análogo ao desenvolvido para o treinamento bayesiano é utilizado para determinação das variáveis irrelevantes, com a SVM alimentada somente com as entradas relevantes sendo treinada posteriormente.

Em relação a C e ε , diante das dificuldades inerentes aos algoritmos baseados em gradiente, transformações logarítmicas são aplicadas a estes parâmetros (Chang e Lin, 2005). Para inicialização, são utilizadas as expressões (Cherkassky e Ma, 2004):

$$C_{ref} = \max \left(\left| \bar{d} + 3s_d \right|, \left| \bar{d} - 3s_d \right| \right) \quad (19)$$

$$\varepsilon_{ref} = 3s \sqrt{\frac{\ln N}{N}}$$

$$s = \sqrt{\frac{1}{N-n} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

Em (19), \bar{d} é a média das saídas desejadas, s_d o respectivo desvio padrão e s é o desvio padrão do erro do modelo de regressão. Neste trabalho, s é estimado através dos resíduos de um modelo linear utilizando o conjunto original de entradas. Os parâmetros σ_i são inicialmente feitos iguais a 0.1.

5 Resultados

Para avaliação dos modelos, são utilizadas três bases de dados. Vale ressaltar que todos os dados são reduzidos (média zero e variância unitária). Além disso, inicialmente foi escolhido um número elevado de entradas visando à verificação das técnicas propostas.

O primeiro conjunto apresenta dados horários de carga e temperatura para o período de 1 de janeiro de 1985 a 31 de março de 1991. Esta base de dados foi utilizada em uma competição entre modelos de previsão (Ramanathan et. al., 1997), podendo ser encontrada em <http://www.ee.washington.edu/class/555/elsharkawi/datafiles/forecasting.zip>. Deve ser prevista a carga horária de 16 até 40 passos à frente, para dias úteis, e de 16 até 80 passos à frente, para fins de semana, para o período de 1 de novembro de 1990 a 31 de março de 1991. Utilizando somente dados do mês onde são realizadas as previsões e dos dois meses anteriores, juntamente com dados do mesmo período no ano anterior, são treinados sete modelos, um para cada dia da semana. É utilizado o seguinte conjunto inicial de entradas: 24 variáveis binárias codificando hora do dia; atrasos $S(k-1), \dots, S(k-6), S(k-24), \dots, S(k-29), S(k-168), \dots, S(k-173)$ das séries de carga e temperatura; previsão de temperatura e o quadrado deste valor, $T(k)$ e $T^2(k)$; temperatura máxima prevista para o dia de previsão e seu quadrado, $T_{max}(d)$ e $T_{max}^2(d)$; temperatura máxima verificada no dia ante-

rior e seu quadrado, $T_{max}(d-1)$ e $T_{max}^2(d-1)$. Assim, um total de 84 entradas é apresentado inicialmente aos modelos. A saída representa a carga prevista $L(k)$. Como previsões de temperatura, são utilizados os próprios valores medidos, com as previsões até 80 passos a frente sendo obtidas via recursão. Resultados para esta base de dados podem ser encontrados em (Ramanathan et. al., 1997).

O segundo conjunto possui dados diários de carga e temperatura máxima para o período de 1 de janeiro de 1997 a 31 janeiro de 1999, disponíveis em <http://neuron.tuke.sk/competition>. Neste caso, são realizadas previsões para o período de 1 a 31 de janeiro de 1999. Evitando recursão, são desenvolvidos 31 modelos, um para cada passo à frente, utilizando todos os dados até 1 de janeiro de 1999. Para o j -ésimo modelo, o conjunto inicial de entradas apresenta $(33+j)$ entradas, consistindo nos últimos 7 valores medidos de carga máxima, $L(d-j), \dots, L(d-6-j)$, juntamente com os $(j+7)$ valores mais recentes de temperatura, $T(d), \dots, T(d-6-j)$, e 19 variáveis binárias, sendo 7 para codificação do dia da semana e 12 para o mês. A saída do modelo representa a carga máxima prevista $L(d)$. Como anteriormente, as temperaturas medidas são utilizadas como previsão. O modelo vencedor da competição tratando destes dados é apresentado em (Chen et. al., 2004).

A última base de dados, encontrada em www.nemco.com.au, apresenta dados de carga, preço da energia e temperatura verificados a cada meia-hora, para o período de 4 de dezembro de 2001 a 31 de dezembro de 2003. A tarefa consiste na previsão de carga horária, de 1 até 6 passos a frente, para diferentes semanas de 2003. Seguindo (Mandal et. al., 2005), onde podem ser encontrados os melhores resultados para este caso, a série em base horária é obtida através da média entre os dois valores verificados na respectiva hora. Para cada dia da semana, são desenvolvidos seis modelos, um para cada passo à frente. O modelo para o j -ésimo passo à frente apresenta um conjunto inicial com $(81-2j)$ entradas, a saber: $(19-j)$ atrasos de carga, preço e temperatura, $S(k-j), \dots, S(k-6), S(k-24), \dots, S(k-29), S(k-168), \dots, S(k-173)$; j previsões de temperatura, $T(k), \dots, T(k-j+1)$; e 24 variáveis binárias representando hora do dia. A saída do modelo é a carga prevista $L(k)$.

Para o MLP, são testadas as seguintes técnicas: retropropagação do erro tradicional, parada antecipada do treinamento, escalonamento do ganho da função de ativação (Reed et. al., 1995) e treinamento bayesiano. Os parâmetros da SVM são obtidos de duas formas, através da minimização do limite dado pela equação (16) e via validação cruzada. A tabela 1 apresenta o erro absoluto percentual médio (EAPM) obtido utilizando os diferentes métodos. A última linha desta tabela apresenta o ganho de desempenho da melhor metodologia em relação ao resultado encontrado na literatura.

O treinamento bayesiano apresentou os melhores resultados para todos os conjuntos de dados, exceto para o terceiro passo à frente do terceiro caso. Visto

que este trabalho não trata de maneira específica os feriados, ao contrário de (Mandal et. al., 2005), a comparação neste caso é injusta, visto que a semana de avaliação para este modelo inclui o Natal e o Ano Novo. Vale destacar que os métodos baseados em validação cruzada, tanto para SVM quanto para MLP, apresentaram resultados inferiores em relação às técnicas propostas, que exploram todo o conjunto de dados.

Tabela 1. Comparação entre os modelos (EAPM)

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
Retropropagação	21.8	4.3	2.7	5.4	9.0	6.0	6.4	4.1
Escalonamento do Ganho	21.4	4.1	2.7	4.3	7.9	5.6	4.8	4.6
Parada Antecipada	12.1	3.4	3.8	4.8	8.0	3.8	5.8	3.4
Treinamento Bayesiano	4.9	1.8	0.5	1.1	4.7	1.4	1.9	1.5
L2SVM Validação Cruzada	4.9	3.5	0.8	1.6	5.0	2.0	2.5	1.9
L2SVM Gradiente	8.7	2.1	0.8	1.8	7.5	1.9	2.0	1.9
Referência	4.7	2.0	0.6	1.5	2.8	1.4	2.2	1.5
Ganho (%)	-3.1	11.7	14.1	24.5	-65.5	3.5	15.2	0.5

Na tabela 2 é apresentado o número médio de entradas selecionadas. Esta tabela ilustra a eficiência das técnicas propostas em reduzir a dimensionalidade do espaço de entrada. A tabela 2 mostra reduções que variam de 10 a 28 %. Por fim, a tabela 3 apresenta o número médio de neurônios na camada oculta, como também o número médio de vetores suporte.

Tabela 2. Número médio de entradas utilizadas

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
Retropropagação	84	49	79	77	75	73	71	69
Escalonamento do Ganho	84	49	79	77	75	73	71	69
Parada Antecipada	84	49	79	77	75	73	71	69
Treinamento Bayesiano	70	40	62	57	60	53	55	56
L2SVM Validação Cruzada	84	49	79	77	75	73	71	69
L2SVM Gradiente	76	45	73	72	74	67	66	65
Redução (%)	17	19	22	26	19	28	23	19

6 Conclusão

Este trabalho investigou a utilização de inferência bayesiana e SVMs no desenvolvimento de modelos autônomos de previsão de carga, incluindo procedimentos automáticos de seleção de entradas e controle de complexidade do modelo. Neste sentido, as técnicas propostas são totalmente independentes, com a intervenção do usuário necessária apenas para determinação do conjunto inicial de entradas. Estes métodos podem atender ao problema de previsão de carga por barramento, onde a dinâmica específica de cada série não pode ser modelada manualmente em virtude do elevado número de barras a serem consideradas.

Tabela 3. Número médio de neurônios e vetores suporte

	Caso 1	Caso 2	Caso 3					
			1 passo	2 passos	3 passos	4 passos	5 passos	6 passos
Retropropagação	10	10	10	10	10	10	10	10
Escalonamento do Ganho	10	10	10	10	10	10	10	10
Parada Antecipada	10	10	10	10	10	10	10	10
Treinamento Bayesiano	8	7	6	4	6	4	6	6
L2SVM Validação Cruzada	428	464	344	316	375	319	317	347
L2SVM Gradiente	642	707	526	485	606	520	523	571

Comparando as técnicas, a otimização dos parâmetros da SVM não constitui uma tarefa simples, ao contrário da estimação dos hiperparâmetros do treinamento bayesiano. Independentemente dos pro-

blemas intrínsecos aos métodos baseados em gradiente, a convergência do algoritmo utilizado demanda várias iterações, elevando o custo computacional requerido. Apesar disso, esta técnica mostrou resultados comparáveis à escolha dos parâmetros via validação cruzada, evidenciando a aplicabilidade do método.

Diante dos resultados encorajadores obtidos pela inferência bayesiana aplicada a MLPs, a utilização desta técnica para obtenção dos parâmetros da SVM surge como alternativa promissora. Algumas idéias neste sentido vêm sendo pesquisadas (Wei et. al., 2004), sinalizando novos caminhos a serem seguidos pelos modelos de previsão de carga.

Agradecimentos

Os autores agradecem o apoio da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) e do CNPq (Conselho Nacional de Desenvolvimento Científico e Pesquisa) pelo suporte financeiro.

Referências Bibliográficas

- Debs, A.S. (1988). *Modern Power Systems Control and Operation*, Kluwer Academic Publishers.
- Khotanzad, A., Afkhami-Rohani, R., Maratukulam, D. (1998). ANNSTLF – Artificial Neural Network Short-Term Load Forecaster – Generation Three, *IEEE Transactions on Power Systems*, **13** (4), n.4, pp. 1413-1422.
- Reis, A.J.R., Alves da Silva, A.P. (2005). Feature Extraction Via Multi-Resolution Analysis for Short-Term Load Forecasting, *IEEE Transactions on Power Systems*, **20**(1), pp. 189-198.
- Cataltepe, Z., Abu-Mostafa, Y.S., Magdon-Ismail, M. (1999). No Free Lunch for Early Stopping, *Neural Computation*, **11**(4), pp. 995-1009.
- Mackay, D.J.C. (1992). *Bayesian Methods for Adaptive Models*, Ph.D. Dissertation, California Institute of Technology, Pasadena, USA.
- Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press.
- Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S. (2002). Choosing Multiple Parameters for Support Vector Machines, *Machine Learning*, **46**, pp. 131-159.
- Chang, M.-W., Lin, C.-J. (2005) Leave-One-Out Bounds for Support Vector Regression Model Selection, *Neural Computation*, **17**(5), pp. 1188-1222.
- Reed, R., Marks II, R.J., Oh, S. (1995). Similarities of Error Regularization, Sigmoid Gain Scaling, Target Smoothing and Training with Jitter, *IEEE Transactions on Neural Networks*, **6**(3), pp. 529-538.
- Ferreira, V.H., Alves da Silva, A.P. (2005). Técnicas de Regularização de Modelos Neurais Aplicadas à Previsão de Carga a Curto Prazo, *VII Congresso Brasileiro de Redes Neurais*, Natal, Rio Grande do Norte, Brasil.
- Treadgold, N.K., Gedeon, T.D. (1999). Exploring Constructive Cascade Networks, *IEEE Transactions on Neural Networks*, **10**(6), pp. 1335-1350.
- Murata, N., Yoshizawa, S., Amari, S.I. (1994). Network Information Criterion – Determining the Number of Hidden Units for an Artificial Neural Network Model, *IEEE Transactions on Neural Networks*, **5**(6), pp. 865-872.
- Stoppiglia, H., Dreyfus, G., Dubois, R., Oussar, Y. (2003). Ranking a Random Feature for Variable and Feature Selection, *Journal of Machine Learning Research*, **3**, pp. 1399-1414.
- Vapnik, V.N. (1998). *Statistical Learning Theory*, John Wiley & Sons.
- Chen, B.-J., Chang, M.-W., Lin, C.-J. (2004). Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001, *IEEE Trans. on Power Systems*, **19**(4), pp. 1821-1830.
- Cherkassky, V., Ma, Y. (2004). Practical Selection of SVM Parameters and Noise Estimation for SVM Regression, *Neural Networks*, **17**(1), pp. 113-126.
- Ramanathan, R., Engle, R., Granger, C.W.J., Vahid-Araghi, F., Brace, C. (1997). Short-Run Forecasts of Electricity Loads and Peaks, *International Journal of Forecasting*, **13**(2), pp. 161-174.
- Mandal, P., Senjyu, T., Uezato, K., Funabashi, T. (2005). Several-Hours-Ahead Electricity Price and Load Forecasting Using Neural Networks, *IEEE PES General Meeting, San Francisco, USA*.
- Wei, C., Keerthi, S.S., Chong, J.O. (2004). Bayesian Support Vector Regression Using a Unified Loss Function, *IEEE Transactions on Neural Networks*, **15**(1), pp. 29-44.