REDES NEURAIS E SUAS APLICAÇÕES EM CALIBRAÇÃO MULTIVARIADA

Eduardo O. de Cerqueira, João C. de Andrade e Ronei J. Poppi*
Departamento de Química Analítica, Instituto de Química, UNICAMP, CP 6154, 13083-970 Campinas - SP
Cesar Mello
Faculdade de Farmácia, Universidade de Franca, CP 82, 14404-400 Franca - SP

Recebido em 28/8/00; aceito em 13/3/01

NEURAL NETWORKS AND ITS APPLICATIONS IN MULTIVARIATE CALIBRATION. Neural Networks are a set of mathematical methods and computer programs designed to simulate the information process and the knowledge acquisition of the human brain. In last years its application in chemistry is increasing significantly, due the special characteristics for model complex systems. The basic principles of two types of neural networks, the multi-layer perceptrons and radial basis functions, are introduced, as well as, a pruning approach to architecture optimization. Two analytical applications based on near infrared spectroscopy are presented, the first one for determination of nitrogen content in wheat leaves using multi-layer perceptrons networks and second one for determination of BRIX in sugar cane juices using radial basis functions networks.

Keywords: neural networks; backpropagation; radial basis functions; multivariate calibration.

INTRODUÇÃO

Atualmente, tem-se observado um grande aumento na disponibilidade de informação em nosso cotidiano. A informação circula entre nós a uma velocidade que seria inconcebível até alguns anos atrás e tende aumentar ainda mais. Hoje em dia o volume de informação dobra a cada quatro anos e num futuro próximo passará a dobrar a cada dois anos. Assim, realmente vivemos no que se pode chamar de sociedade da informação. Grande parte deste aumento no volume de informações se deve ao aparecimento e uso disseminados de microcomputadores equipados com softwares, como Windows^{TM}, que tornam o seu uso extremamente simples, até mesmo para crianças em fase de alfabetização.

Uma das atividades humanas que mais se beneficiou com o uso de computadores foi a própria ciência, em particular a química analítica. Até algumas décadas atrás o principal problema enfrentado no desenvolvimento de um método analítico era como se obter os dados. Gastava-se muito tempo, dinheiro e esforço para se obter um pequeno conjunto de dados.

Graças a *computadorização* dos instrumentos o principal problema não é mais como obter os dados, mas como tratá-los de modo eficiente. Tipicamente tais instrumentos fornecem um grande conjunto de dados e para que tais dados sejam convertidos em informação útil deve-se usar métodos matemáticos e estatísticos modernos e eficientes. Dentre tais métodos destaca-se os métodos lineares e não lineares de calibração multivariada e mais recentemente as redes neurais[1].

Para uma melhor compreensão da álgebra envolvida nos modelos de tratamento de dados, foi utilizada a seguinte notação:

- Vetores - letras minúsculas em negrito (ex: x);
- Matrizes - letras maiúsculas em negrito (ex: X);
- Escalares - letras minúsculas sem negrito (ex: l);
- Matrizes diagonais - letras maiúsculas sem negrito (ex: S).

Redes Neurais

Recentes avanços em neurofisiologia têm desvendado vários mecanismos sobre o fluxo e o processamento de informações que ocorrem no cérebro humano. Alguns destes mecanismos foram modelados matematicamente permitindo a elaboração de algoritmos computacionais que simulam, ainda que de modo simplificado, a mais básica das estruturas cerebrais: *o neurônio*[2].

A capacidade de implementar computacionalmente versões simplificadas de neurônios biológicos deu origem a uma subespecialidade da inteligência artificial, conhecida como redes neurais. Existem várias definições para redes neurais, portanto adotaremos aquela que nos parece mais geral e aplicável a qualquer área da ciência.

> *"Redes neurais é o nome dado a um conjunto de métodos matemáticos e algoritmos computacionais especialmente projetados para simular o processamento de informações e aquisição de conhecimento do cérebro humano".*

Operacionalmente podemos considerar uma rede neural como uma "*caixa de processamento*" que pode ser treinada para que, a partir de um conjunto de dados de entrada (inputs), possa gerar uma ou mais saídas (outputs), conforme representado na Figura 1.

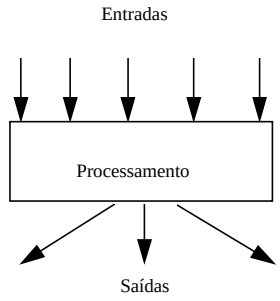

Figura 1. *Representação operacional de uma rede neural.*

Por exemplo, pode-se treinar uma rede neural para que a partir do espectro de RMN-H (input) de um composto orgânico ela apresente como saída (output) a estrutura molecular do mesmo, ou ainda os inputs podem ser dados clínicos de um paciente e os outputs o diagnóstico e tipo de evolução. Enfim, as aplicações são extremamente variadas e estão presentes em quase todas as áreas do conhecimento humano.

*e-mail: ronei@iqm.unicamp.br

Neste ponto do texto, já temos uma visão geral sobre redes neurais e, portanto, estamos aptos a aprofundar nossos conhecimentos sobre o assunto. Como veremos, esta é uma tarefa fácil visto que as redes neurais são elaboradas a partir de cinco conceitos básicos, extremamente simples. Estes elementos são:

① *Neurônios artificiais*
② *Sinapses e pesos*
③ *Funções de transferência*
④ *Arquitetura de redes neurais*
⑤ *Treinamento de redes neurais.*

Neurônios artificiais

Neurônios artificiais[1-3] são as unidades básicas de processamento da informação, projetadas para simular o comportamento de neurônios biológicos. Simplificadamente, um neurônio biológico pode ser descrito como um corpo celular contendo dois conjuntos de ramificações: *dendritos* e *axônio* (Figura 2A). Analogamente aos neurônios biológicos, os neurônios artificiais possuem um corpo de processamento de informação com duas ramificações: entradas (dendritos) e saída (axônio), conforme representado na Figura 2B.

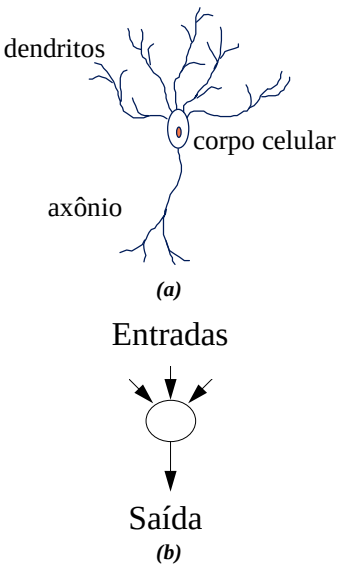

Figura 2. *(A) Neurônio biológico ; (B) Neurônio artificial.*

Sinapses e pesos: A transmissão de sinais pelos neurônios

Sem entrar na físico-química do processo, a transmissão de sinais entre neurônios[4,5] pode ser entendida do seguinte modo: O sinal entra no neurônio através dos dendritos, passa pelo corpo celular e em seguida é transmitido para outros neurônios, da rede neural, através do axônio. A passagem do sinal de um neurônio para os dendritos de um outro neurônio é chamada de sinapse.

Sinapses representam barreiras que modulam o sinal que é trocado através delas e a quantidade de sinal trocado em uma sinapse depende de um parâmetro chamado de *intensidade da sinapse*. Em um neurônio artificial a intensidade da sinapse é simulada por um fator de ponderação chamado *peso da sinapse* ou simplesmente peso, conforme ilustrado a Figura 3.

O sinal total que entra no corpo de processamento de um neurônio artificial é costumeiramente chamado Net, cujo valor é calculado através da simples multiplicação do sinal que chega ao neurônio pelo peso da sinapse em questão

Net = Sinal de entrada (input) x Peso.

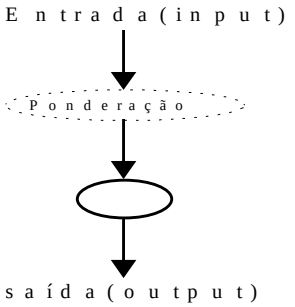

Figura 3. *Representação da sinapse artificial.*

Neurônios possuem um grande número de dendritos e portanto, podem estabelecer várias sinapses com outros neurônios da rede neural, simultaneamente (Figura 4). Assim, o cálculo do *net* deve ser generalizado para *n* entradas (inputs). Chamando os sinais de entrada de x_i e os pesos de w_i , o valor do *net* será dado por:

$$Net = \sum_{i=1}^{n} w_i x_i \qquad (1)$$

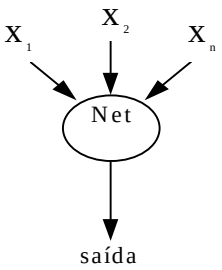

Figura 4. *Representação generalizada da sinapse.*

Até o momento vimos como calcular o sinal que entra em um neurônio, mas qual será o sinal de saída de um neurônio?

Funções de transferência

Como já poderíamos imaginar, a saída de um neurônio (output) será obtida em função do sinal de entrada (input), isto é, a saída será função do *net*

$$\text{saída} = f\,(net) \qquad (2)$$

A função $f\,(net)$ deve possuir as propriedades:

A função de transferência[1,6] não deve ser negativa, pois ou o neurônio troca sinal com outro neurônio ou não troca, não há como trocar sinal negativo.

A função de transferência deve ser contínua, pois um neurônio não pode trocar sinal com outro neurônio, indefinidamente.

Dentre as várias funções que preenchem os pré-requisitos anteriores, somente três são rotineiramente usadas, as quais sejam:

1) Função sigmoidal ou logística

Com certeza esta é a função de transferência mais utilizada em redes neurais. Esta função (Figura 5) também é a que mais se aproxima da saída de um neurônio biológico. Matematicamente, escrevemos a função sigmoidal como segue:

$$saída = f\,(net) = \frac{1}{1+e^{-(net)}} \qquad (3)$$

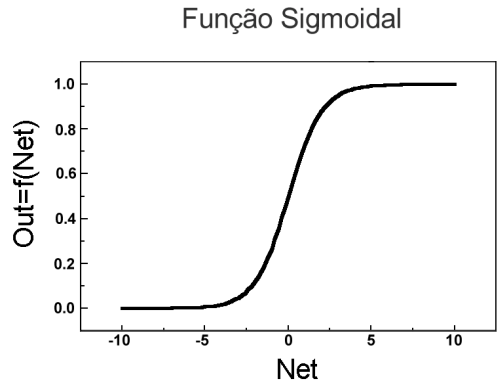

Figura 5. *Função sigmoidal.*

2) Função degrau de Heaviside

Como podemos observar na Figura 6, a função de heavside é muito simples e só pode assumir os valores 0 ou 1.

$$out = f(net) = \begin{cases} 1 \text{ se } net > \text{ valor limite} \\ 0 \text{ se } net \leq \text{ valor limite (arbitrário)} \end{cases}$$

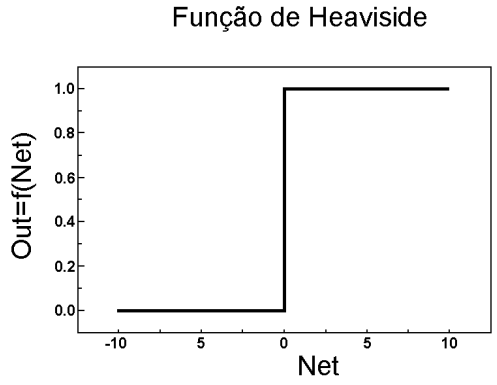

Figura 6. *Função degrau de Heaviside.*

3) Função Linear

Esta função, naturalmente, dispensa comentários. A saída do neurônio será dada pôr:

$$saída = a(net) + b \quad a,b \in R_+^* \tag{4}$$

Arquitetura de redes neurais

Uma rede neural artificial possui sempre uma *camada de entrada* e uma *camada de saída*. Entre a camada de entrada e a de saída, existe um número variável de *camadas intermediárias*. A esta disposição das camadas e número de neurônios por camada, dá-se o nome de *arquitetura* da rede neural. Na Figura 7 podemos observar a arquitetura de uma rede neural que foi utilizada para a determinação de nitrogênio em folhas de trigo a partir de espectroscopia de reflectância difusa no infravermelho próximo (NIRR).

Esta rede neural possui a seguinte arquitetura: 1 camada de entrada com 277 neurônios, uma camada intermediária com 5 neurônios e uma camada de saída com 1 neurônio. As funções de transferência utilizadas estão indicadas na própria Figura 7. As entradas (inputs) foram as intensidades, isto é, valores de (-log(R/Ro)) lidos em diferentes comprimentos de onda, em espectros de reflectância difusa na região do infravermelho próximo (NIRR) de folhas de trigo. A saída (output) foi a concentração de nitrogênio nestas folhas.

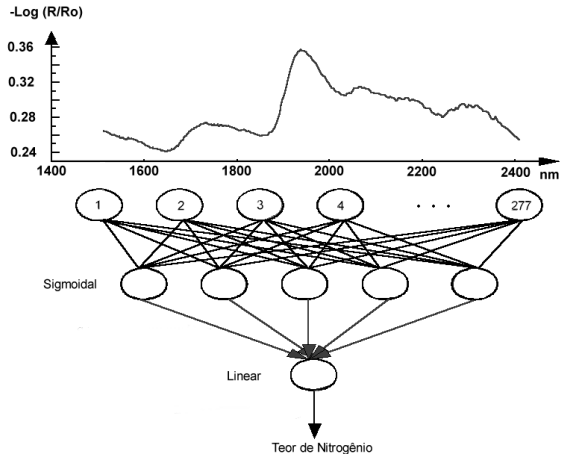

Figura 7. *Representação de uma rede neural utilizada para determinação do teor de nitrogênio em folhas de trigo, através de espectroscopia de reflectância difusa no infravermelho próximo (NIRR).*

Vale ressaltar que o número de camadas intermediárias e número de neurônios nestas, é um parâmetro ajustável conforme veremos adiante.

Cálculo do valor do Net em uma rede neural

Demonstramos como calcular o *Net* e a saída (output) para *um* neurônio. Vejamos agora como generalizar o cálculo do *Net* para uma rede neural.

É fácil imaginar que em uma rede neural com várias camadas, a saída de uma camada será a entrada da camada seguinte. Portanto, para calcular a saída final de uma rede neural basta calcular sequencialmente as entradas e saídas por toda a rede neural. Este cálculo que conceitualmente é muito simples, pode se tornar tedioso e complexo se a rede possuir várias camadas e muitos neurônios nestas camadas. Uma maneira mais simples e computacionalmente mais eficiente de se fazer isto, é "escrever" a rede neural de modo matricial. Assim, uma camada com *n* neurônios, cada um com *m* pesos, é representada por uma matriz *W* com *n* linhas e *m* colunas ($W_{n,m}$). Para uma rede neural com várias camadas, cada uma das camadas será representada por uma matriz W^l (em que *l* indica a camada). Essa matriz W é calculada na etapa de treinamento da rede neural mostrada a seguir.

Para a rede neural representada na Figura 7, temos $W^1_{277\times277}, W^5_{5\times1385}$ *e* $W^3_{1\times5}$:

① *Camada de entrada:*

$$W^1_{277\times277} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & \cdots & w_{1,277} \\ w_{2,1} & w_{2,2} & w_{2,3} & \cdots & w_{2,277} \\ w_{3,1} & w_{3,2} & w_{3,3} & \cdots & w_{3,277} \\ \vdots & \vdots & \vdots & & \vdots \\ w_{277,1} & w_{277,2} & w_{277,3} & \cdots & w_{277,277} \end{bmatrix}$$

② *Camada intermediária:*

$$W^2_{5\times1385} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,4} & w_{1,5} & \cdots & w_{1,1385} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,4} & w_{2,5} & \cdots & w_{2,1385} \\ w_{3,1} & w_{3,2} & w_{3,3} & w_{3,4} & w_{3,5} & \cdots & w_{3,1385} \\ w_{4,1} & w_{4,2} & w_{4,3} & w_{4,4} & w_{4,5} & \cdots & w_{4,1385} \\ w_{5,1} & w_{5,2} & w_{5,3} & w_{5,4} & w_{5,5} & \cdots & w_{5,1385} \end{bmatrix}$$

③ Camada de saída:

$$W^3_{1\times5} = [w_{1,1} \quad w_{1,2} \quad w_{1,3} \quad w_{1,4} \quad w_{1,5}]$$

As entradas, isto é, valores de (-log(R/Ro)) lidos em diferentes comprimentos de onda, no espectro de reflectância difusa no infravermelho próximo podem ser escritas vetorialmente do seguinte modo:

$$x_i^l = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad \cdots \quad x_{277}]$$

O cálculo do *net* para cada um dos neurônios, da camada de entrada, por exemplo, pode ser escrito como:

$$Net = [Net_1, Net_2, \ldots Net_{277}] = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & \cdots & w_{1,277} \\ w_{2,1} & w_{2,2} & w_{2,3} & \cdots & w_{2,277} \\ w_{3,1} & w_{3,2} & w_{3,3} & \cdots & w_{3,277} \\ \vdots & \vdots & \vdots & & \vdots \\ w_{277,1} & w_{277,2} & w_{277,3} & \cdots & w_{277,277} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{277} \end{bmatrix}$$

O cálculo do *Net* pode ser generalizado, para qualquer camada, através da seguinte equação:

$$Net_j^l = \sum_{i=1}^{m} W_i^l x_i^l \tag{5}$$

Uma vez que toda entrada de uma camada pode ser relacionada com a saída da camada anterior, podemos rescrever a Equação 5 como segue:

$$Net_j^l = \sum_{i=1}^{m} W_{ji}^l out_i^{l-1} \tag{6}$$

Conhecidos o *Net* e as funções de transferência pode-se "propagar" o sinal pela rede, desde a entrada (input) até a saída (output).

Treinamento de redes neurais

Para que uma rede neural execute tarefas como reconhecimento de padrões ou modelagem de dados (calibração), devemos treiná-la. Vejamos como se faz isto.

Os pesos w são escolhidos inicialmente de modo aleatório, a fim de se ter uma rede inicial a ser treinada, ou seja, otimizada para as entradas e saídas da etapa de treinamento.

Na etapa de treinamento[7] várias entradas são apresentadas e propagadas pela rede neural. Para a rede neural utilizada para determinação de nitrogênio em folhas de trigo, apresentada na Figura 7, os inputs foram intensidades lidas em diferentes comprimentos de onda, para vários espectros de reflectância difusa na região do infravermelho próximo (NIRR).

Uma vez propagados os inputs através da rede neural, podemos calcular a saída, isto é, o valor previsto da concentração de nitrogênio nestas folhas, para o exemplo representado na Figura 7. Obtidos os valores previstos pela rede neural, podemos calcular um erro de previsão (*E*), o qual pode ser definido, em princípio, como o somatório do quadrado da diferença entre o valor previsto pela rede (saída) e o valor real (target), como segue:

$$E = \sum_{k=1}^{n} (y_k - \hat{y}_k)^2 \tag{7}$$

em que é o valor real (target) e $\hat{y}$ o valor previsto (output) pela rede neural. Conhecido o erro de previsão, quais parâmetros da rede neural devemos corrigir para minimizá-lo? A resposta para esta pergunta é muito simples, pois uma vez fixada a arquitetura da rede neural, os únicos parâmetros passíveis de correções são os pesos, dado que não há outros parâmetros. Existem alguns métodos para se fazer tal correção, entretanto somente dois são usados com freqüência: a retropropagação de erros[8] e o método de Marquardt-Levenberg[9].

No método da retropropagação de erros, o processo de correção dos pesos é iniciado na última camada e prossegue em direção à primeira camada, daí o nome retropropagação. Este processo pode ser visualizado na Figura 8. Sem entrar em deduções matemáticas, a equação para a correção dos pesos pode ser escrita como:

$$\Delta W_{ji}^l = \eta . \delta_j^l out^{l-1} + \mu \Delta W_{ji}^{previo} \tag{8}$$

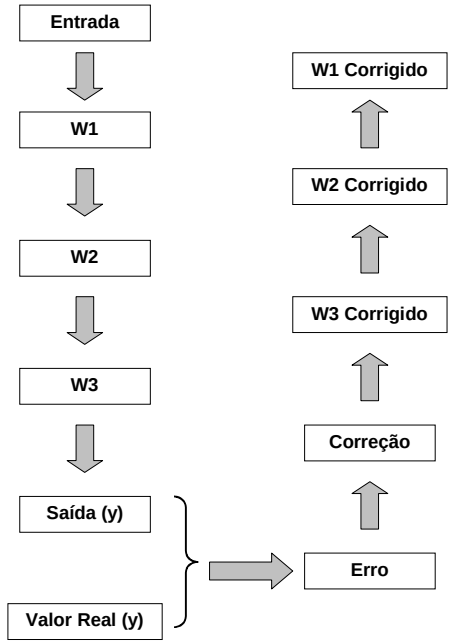

Figura 8. Representação da retropropagação para uma rede com uma camada de entrada (W1), uma intermediária (W2) e uma camada de saída (W3)

De acordo com a equação anterior, a correção dos pesos é composta pela soma de dois termos que atuam conjuntamente para minimizar o erro de previsão. Vejamos o que cada um representa:

- Primeiro termo:

$$\eta \delta_j^l out^{l-1}$$

O fator de correção δ_j^l é obtido utilizando-se o método do gradiente descendente, cuja essência é que o erro (*E*) deve apresentar um mínimo em função do parâmetro que o causa, ou melhor, δ_j^l fornecerá os pesos que minimizam o erro (*E*). O parâmetro η chamado de velocidade de treinamento é introduzido para ponderar as correções, de modo a evitar grandes correções principalmente no final do processo iterativo, quando boa parte dos pesos já foi corrigido e uma correção muito acentuada pode comprometer todas as correções efetuadas.

- Segundo termo:

$$\mu \Delta W_{ji}^{previo}$$

A constante m desloca o peso "ótimo" obtido pelo primeiro termo, forçando que o erro seja reavaliado neste novo peso. Este processo evita convergências para mínimos locais, pois um erro maior indica que o mínimo obtido pelo primeiro

termo era realmente global; caso contrário, o mínimo era local e a busca pelo mínimo prossegue. Resumindo, o segundo termo atua como dispositivo de segurança do primeiro termo, evitando mínimos locais.

No segundo método de treinamento as correções dos pesos são feitas de acordo com uma variante do método de Gauss-Newton[9], conhecida como método de Marquardt-Levenberg. Neste método as correções dos pesos são calculadas de acordo com a equação:

$$\Delta W=(J^{\tau} J+\lambda I)\text{-}1\ J^{\tau} E \quad (9)$$

onde J é a matriz jacobiana do erro para cada um dos pesos, λ é um escalar positivo, I é a matriz identidade e E é o vetor de erros dado pela Equação 7.

Em ambos os casos, a etapa de treinamento se encerra quando o erro atingir o critério de convergência previamente estabelecido. Neste ponto, a rede está treinada e podemos utilizar um conjunto de espectros de NIRR, diferente daquele utilizado no treinamento, para avaliar as propriedades de generalização da rede neural artificial.

As redes neurais geralmente apresentam uma boa capacidade de generalização, ou seja, apresentam resultados satisfatórios quando aplicada em amostras que não participaram do treinamento, especialmente em situações em que os dados apresentem não linearidades, visto que geralmente se utilizam funções e transferência não lineares. Entretanto, em alguns casos, as redes neurais artificiais, mesmo depois de terem sido treinadas, apresentam baixos erros de calibração e elevados erros de previsão, isto é, sobreajuste (overfitting), devido ao número excessivo de neurônios utilizados na camada intermediária. O número de neurônios na camada intermediária atua de modo análogo ao número de componentes principais utilizado na regressão de componentes principais[10,11] (PCR), ou à ordem do polinômio utilizado em regressão polinomial[12], por exemplo. Portanto, um procedimento fundamental para evitar-se sobreajuste (overfitting), isto é, pequenos erros de calibração e elevados erros de previsão, é a otimização do número de neurônios da camada intermediária.

A otimização da Arquitetura

A seleção do número ótimo de neurônios na camada intermediária ou otimização da arquitetura é feita variando-se o número destes neurônios e repetindo-se o processo de correção dos pesos. Escolhe-se então a configuração que apresentar os menores erros de calibração (%SEC) e previsão (% SEP), dados pelas equações 10 e 11

$$\%SEC = \frac{1}{\bar{c}_c}\sqrt{\frac{\sum_{i=1}^{n_c}(y_i-\hat{y}_i)^2}{n_c-k-1}}\times 100 \quad (10)$$

$$\%SEC = \frac{1}{\bar{c}_p}\sqrt{\frac{\sum_{i=1}^{n_p}(y_i-\hat{y}_i)^2}{n_p}}\times 100 \quad (11)$$

em que $\bar{c}_c$ e $\bar{c}_p$ são as concentrações médias da amostras utilizadas no conjunto de calibração e previsão respectivamente, n_c e n_p representam, respectivamente, o número de amostras usadas na calibração e na previsão, y_i é o valor real e $\hat{y}_i$ o valor previsto pelo modelo e k é o número de parâmetros usados no modelo. Para redes neurais artificiais o número de parâmetros utilizado no modelo não é conhecido e é utilizado como uma aproximação, para o denominador da Equação 10.

Aplicando-se o método das redes neurais anteriormente descrito para a determinação de nitrogênio em folhas de trigo a partir de espectros de refletância difusa no infravermelho próximo (NIRR) foi possível obter modelos que apresentaram baixos erros em amostras de calibração e amostras de previsão. Este fato pode ser verificado observando-se a Figura 9 onde estão representados os valores percentuais previstos contra os valores reais obtidos através do método Kjeldahl.

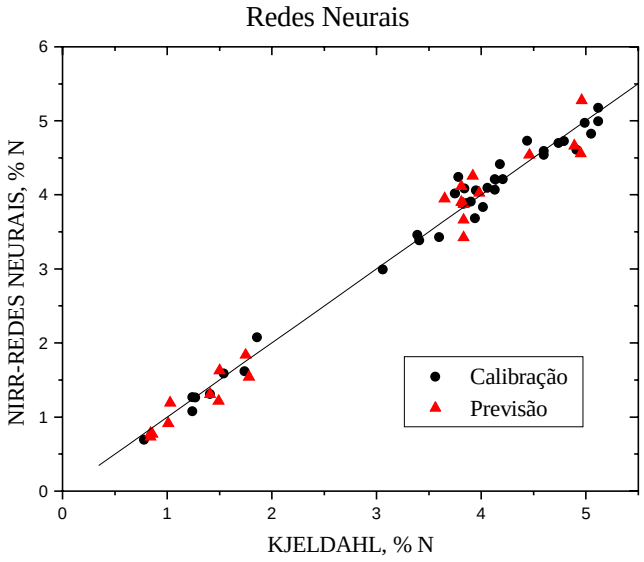

Figura 9. *Valor previsto, para o teor de nitrogênio, pelo modelo de calibração NIRR-RN versus o valor obtido com o método Kjeldahl.*

O leitor atento, a esta altura já deve ter observado que para construir modelos com redes neurais o conjunto de previsão também é utilizado na escolha da arquitetura ótima. Portanto, devemos ter um terceiro conjunto de espectros para realmente verificar as capacidades de previsão da rede neural otimizada. Afinado com este tipo de problema sempre utilizamos neste trabalho, três conjuntos de dados (calibração, validação e previsão) como veremos adiante no texto. Conforme descrevemos, a otimização da arquitetura é basicamente uma tarefa de tentativa e erro e um tanto tediosa, mas existe um modo de evitá-la, como veremos no próximo item.

REDES NEURAIS COM APODIZAÇÃO

A idéia básica deste método é iniciar a rede neural com um número razoável de neurônios na camada intermediária e, durante a etapa de treinamento cortar as conexões (ou pesos) dos neurônios que possuem pouca influência no erro E (Equação 7). Neurônios que tiverem todas as conexões cortadas serão eliminados e, portanto, ao final dos "cortes", sobrarão somente os neurônios realmente necessários à modelagem. A técnica de apodização[13,14] (pruning) reduz a complexidade da rede neural, melhorando sua capacidade de previsão, pois evita modelos sobre parametrizados (muitos neurônios) em que a possibilidade de sobreajuste (overfitting) é grande.

Existem basicamente dois métodos para a apodização de redes neurais: Optimal Brain Damage[15] (OBD) e Optimal Brain Surgeon[16-18] (OBS). Em ambos os métodos as conexões (ou pesos) são cortadas e a correspondente variação no erro E, chamada de saliência, é avaliada.

No método OBD as conexões são cortadas durante a etapa de treinamento e a rede neural não é retreinada após os cortes. No método OBS, as conexões são cortadas e, após o corte de uma conexão, a rede é retreinada, permitindo que um número maior de cortes seja efetuado. Além disso, no método OBS a rede neural é re-treinada, aproximando-se os erros de treinamento por uma função quadrática, de modo a garantir a existência de um mínimo.

Resumidamente, o que se faz no método OBS é adicionar um termo, chamado termo de regularização[19,20], à função erro (*E*) dada pela Equação 7, para obter a função *"custo"*:

$$C(W) = E(W) + \frac{1}{2} W^{\tau} D W \tag{12}$$

O termo de regularização simplesmente penaliza pesos com valores reduzidos, sendo esta penalização, ponderada pelo fator *D*. O fator de regularização auxilia no processo de pruning e assegura a estabilidade numérica do método. Em seguida, expande-se a função custo em série de Taylor até termos de segunda ordem, em torno de um possível mínimo W_0, como segue:

$$C(W) = C(W_0) + \frac{1}{2} \delta W^{\tau} H \delta W \tag{13}$$

em que $\delta W = W - W_0$ e $H = (A + D)$, é a matriz Hessiana da função custo, com termo de regularização. *A* é a matriz das derivadas segundas dos erros de treinamento $\left(\partial^2 E / \partial W^2 \right)$ contendo todas os termos de segunda ordem.

Após esta fase, pesos devem ser eliminados e a função custo minimizada. A eliminação do j-ésimo peso pode ser expressa como:

$$\delta W^{\tau} e_j = -W_0^T e_j \tag{14}$$

em que é o *j-ésimo* vetor unitário. O extremo vinculado pose ser obtido aplicando-se o método dos multiplicadores de Lagrange[21]:

$$\tilde{C}_j(W) = C(W) + \lambda (\delta W + W_0)^{\tau} e_j \tag{15}$$

em que λ é o multiplicador de Lagrange.

Resolvendo-se a Equação 15 obtém-se o extremo vinculado:

$$\delta W = -\lambda H^{-1} e_j \tag{16}$$

Neste ponto é possível voltar a Equação 12, explicitar *E(W)* obter a variação do erro em função dos pesos $\left(\frac{\partial E(W)}{\partial W} \right)$, substituir o valor de δW pelo valor obtido na Equação 16 e obter a seguinte equação para a saliência, isto é, a variação no erro de treinamento devido a eliminação de um determinado peso W_j

$$\delta E_j(W) = \lambda W_0 D H^{-1} e_j + \frac{1}{2} \lambda^2 e_j^T H^{-1} (\nabla^2 E + D) H^{-1} e_j \tag{17}$$

A equação da saliência fornece um critério matemático para que o corte de pesos possa ocorrer.

Aplicando-se o método das redes neurais com pruning para a determinação de nitrogênio em folhas de trigo foi possível obter modelos com erros padrões de previsão (%SEP) de determinação da ordem de 5%. Os gráficos dos valores previstos versus os valores de referência, obtidos com o método kjeldahl estão apresentados na Figura 10.

As técnicas de apodização de redes simplificam significativamente o processo de otimização da arquitetura e nos permite obter modelos com pequena possibilidade de sobreajuste (overfitting). Este fato pode ser observado comparando os resultados obtidos na determinação de nitrogênio utilizando-se redes neurais sem poda e com poda conforme mostrado na Tabela 1.

REDES COM FUNÇÕES RADIAIS DE BASE

Adicionalmente às redes neurais com múltiplas camadas, um

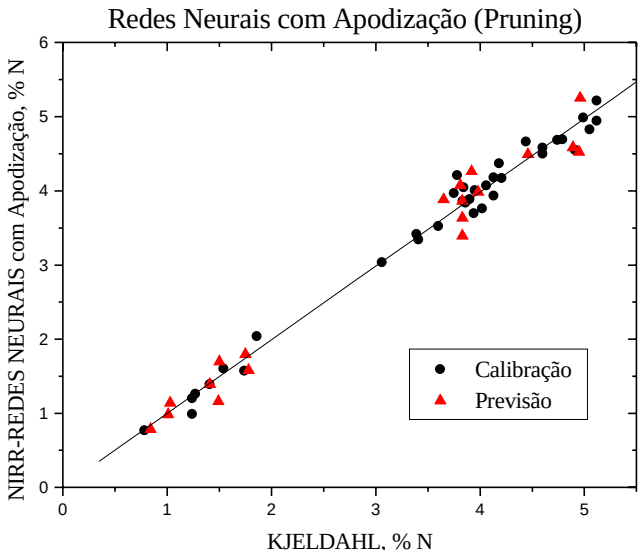

Figura 10. *Valor previsto, para o teor de nitrogênio, pelo modelo de calibração NIRR-RNP versus o valor obtido com o método Kjeldahl.*

Tabela 1. Comparação entre os modelos de redes neurais utilizados na determinações de nitrogênio em folhas de trigo.

	%RMSEC	%RMSEP
Redes neurais sem poda	4,78	6,53
Redes neurais com poda	4,8	6,29

tipo específico de rede que tem tido recente aplicação [22-24] é a rede com funções radiais de base (RBFN). Na RBFN os neurônios geralmente são funções radiais multivariadas, com vários argumentos de entrada, e com comportamento radial em relação a cada argumento, ou seja, as funções radiais decrescem (ou crescem) monotonicamente a medida que se distanciam do ponto central. Existe um grande número de funções radiais que podem ser utilizadas para esse fim, sendo mais comuns as funções com fórmula geral do tipo:

$$h(x) = \phi((x-c)^t R^{-1} (x-c)) \tag{18}$$

onde ϕ é o tipo de função usada, c é a posição em que a função está centrada e **R** é a métrica utilizada. O termo $z = ((x-c)^t R^{-1} (x-c))$ define a distância entre o vetor de entrada **x** e o vetor de centros c na métrica definida por **R** (cada entrada corresponde a uma RBF, que por sua vez possui um centro e um raio em relação a cada entrada).

Os tipos mais comuns de RBFs utilizadas são:

Gaussiana: $\phi(z) = e^{-2}$

Multiquádrica: $\phi(z) = (1+z)^{\frac{1}{2}}$

Multiquádrica inversa: $\phi(z) = (1+z)^{-\frac{1}{2}}$

Cauchy: $\phi(z) = (1+z)^{-1}$

Normalmente o sistema métrico utilizado é euclidiano, e nesse caso, $\mathbf{R} = r^2 \mathbf{I}$, onde r é o raio (dispersão) da RBF utilizada e I é a matriz identidade de ordem igual ao tamanho de **x**. Isso simplifica a RBF utilizada:

$$h(x) = \phi\left(\frac{(x-c)^t (x-c)}{r^2} \right) = \phi\left(\sum \frac{(x_j - c_i)^2}{r^2} \right) \tag{19}$$

caso haja apenas uma variável nos dados de entrada (cada amostra for um escalar, ao invés de um vetor) o somatório se restringe a um termo. A Figura 11 mostra o perfil das RBF mais comumente utilizadas. Nesse caso, trata-se de RBFs univariadas centradas na origem (c = 0) e com raio unitário (r = 1).

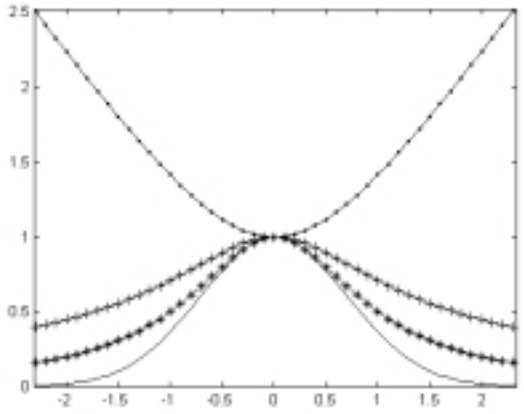

Figura 11. *Perfil dos tipos de RBF mais utilizados: gaussiana(—); multiquádrica (-·); multiquádrica inversa (-+) e cauchy (-*).*

No caso mais comum, em que as RBFs são multivariadas, as mesmas apresentam um perfil semelhante ao descrito na Figura 11 para cada variável do vetor de entrada (**x**), podendo possuir raios e centros iguais ou distintos em relação as mesmas. RBFs por si só, são apenas uma classe de funções, e podem ser utilizadas em rede sob qualquer arranjo. Porém, o termo RBFN (Radial Basis Function Network) ou rede de funções radiais de base, geralmente se refere a uma rede com uma camada, conforme mostrado na Figura 12.

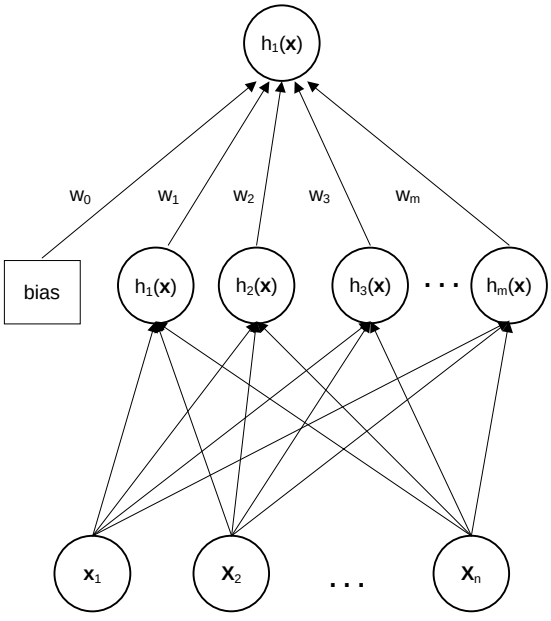

Figura 12. *RBFN tradicional com uma camada oculta: cada um das n variáveis dos dados são utilizados como argumento da m RBFs, cujas saídas são linearmente combinadas através dos pesos w gerando uma saída f(x).*

Cada variável da matriz de dados é usada como argumento de todas as RBFs, e a saída de cada RBF é ponderada e somada para se determinar uma saída. Adicionalmente pode-se acrescentar um termo independente dos dados de entrada para otimizar o ajuste ("bias").

Uma RBFN não é linear se as RBFs puderem mudar de posição (variação de c) ou tamanho (variação de r) ou caso a rede possua mais de uma camada oculta. Caso contrário, embora a RBFN seja formada por funções não lineares, ela é linear em relação aos parâmetros (w), ou seja, a saída da rede é uma combinação linear das saídas das funções. Em outras palavras, desde que não haja RBFs com raios e centros iguais simultaneamente, elas formam uma base para um espaço vetorial no qual serão modeladas as amostras em questão.

Devido a linearidade das RBFNs com apenas uma camada oculta, uma vez fixados o número, os raios e os centros das RBFs, os pesos (parâmetros de ponderação w) podem ser determinados pelo método dos mínimos quadrados. Isso possibilita o cálculo desses parâmetros de modo algébrico (matricial), o qual é muito mais rápido que os modelos iterativos de otimização de redes neurais.

Para a execução do método de mínimos quadrados, basicamente tem-se que diferenciar a expressão do somatório quadrático em relação a cada parâmetro a ser otimizado, igualar cada expressão a zero e resolver o sistema de equações resultante.

O modelo de RBFN gera uma resposta do tipo:

$$f(x) = \sum_{j=1}^{m} w_j h_j(x) \tag{20}$$

Para a determinação dos pesos é necessário minimizar o somatório quadrático

$$S = \sum_{i=1}^{p} (\hat{y}_i - f(x_i))^2 \tag{21}$$

Diferenciando o somatório quadrático em relação a cada parâmetro têm-se

$$\frac{\partial S}{\partial w_j} = 2\sum_{i=1}^{p} (f(x_i) - \hat{y}_i)\frac{\partial f}{\partial w_i}(x_i) \tag{22}$$

Devido à linearidade do sistema, as derivadas em relação a parâmetros distintos são independentes.

$$\frac{\partial f}{\partial w_i}(x_i) = h_j(x_i) \tag{23}$$

Substituindo a Equação 23 na derivada do somatório do erro quadrático (Equação 22), e igualando a zero têm-se:

$$\sum_{i=1}^{p} f(x_i)h_j(x_i) = \sum_{i=1}^{p} \hat{y}_i h_j(x_i) \tag{24}$$

Colocando em forma matricial, têm-se

$$\mathrm{h}_j^t \mathrm{f} = \mathrm{h}_j^t \hat{\mathrm{y}} \tag{25}$$

onde

$$\mathrm{h}_j = \begin{bmatrix} h_j(x_1) \\ h_j(x_2) \\ \vdots \\ h_j(x_p) \end{bmatrix}, \quad \mathrm{f}_j = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_p) \end{bmatrix}, \quad \mathrm{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_p \end{bmatrix}$$

ou seja,

$$\begin{bmatrix} \mathrm{h}_1^t \mathrm{f} \\ \mathrm{h}_2^t \mathrm{f} \\ \vdots \\ \mathrm{h}_m^t \mathrm{f} \end{bmatrix} = \begin{bmatrix} \mathrm{h}_1^t \hat{\mathrm{y}} \\ \mathrm{h}_2^t \hat{\mathrm{y}} \\ \vdots \\ \mathrm{h}_m^t \hat{\mathrm{y}} \end{bmatrix} \tag{26}$$

Utilizando as propriedades de multiplicação matricial,

$$H^t f = H^t \hat{y} \quad (27)$$

onde **H** é a matriz de "design" com os vetores $\{h_j\}_{j=1}^{m}$ em suas colunas

$$H = [h_1\ h_2\ ...\ h_m] \quad (28)$$

e tem p linhas relativas às amostras do conjunto de treinamento

$$\mathbf{H} = \begin{bmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \cdots & h_m(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \cdots & h_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_p) & h_2(\mathbf{x}_p) & \cdots & h_m(\mathbf{x}_p) \end{bmatrix} \quad (29)$$

O vetor **f** pode ser decomposto no produto de dois termos: A matriz de design e o vetor de pesos

$$f_i = f(x_i) = \sum_{j=1}^{m} \hat{w}_j h_j(x_i) = \bar{h}_i^t \hat{w} \quad (30)$$

onde

$$h_j = \begin{bmatrix} h_1(x_i) \\ h_2(x_i) \\ \vdots \\ h_m(x_i) \end{bmatrix} \quad (30)$$

É importante observar que enquanto é uma das colunas de H, é uma de suas linhas.

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_p \end{bmatrix} = \begin{bmatrix} \bar{h}_1^t w \\ \bar{h}_2^t w \\ \vdots \\ \bar{h}_m^t w \end{bmatrix} = H\hat{w} \quad (31)$$

Finalmente substituindo na Equação 27 tem-se:

$$H^t \hat{y} = H^t H \hat{w} \quad (32)$$

Multiplicando a Equação 32 à esquerda em ambos lados pela inversa de têm-se:

$$\hat{w} = (H^t H)^{-1} = H^t \hat{y} \quad (33)$$

Para adicionar o termo independente (bias) à rede em questão, basta adicionar um vetor coluna unitário na matriz de design (H), de modo a gerar uma RBF cuja resposta independe dos dados na matriz de design.

A fim de eliminar problemas numéricos no cálculo dos modelos de RBFN e acelerar os mesmos, usualmente utiliza-se um modelo de ortogonalização nos dados de entrada. Assim, elimina-se a correlação entre as variáveis de entrada e o número das mesmas. Esse processo condensa a informação relevante dos dados de entrada em uma matriz de menor tamanho, eliminando informações redundantes, facilitando assim, o cálculo com matrizes de menor tamanho.

Em uma rede com RBFs, expande-se o espaço de modelagem com uma base maior que o número de entradas (após a ortogonalização e consequente eliminação da correlação das variáveis de entrada). Nesse espaço com dimensão superior a separação dos padrões de informação é feita mais facilmente e posteriormente é feita uma regressão linear múltipla (método dos mínimos quadrados) entre os dados de entrada nessa nova base e a saída esperada (concentração do analito de interesse, por exemplo).

O principal problema em se trabalhar com RBFN, é definir o número, as posições, raios e tipos das RBFs utilizadas na rede, de modo otimizado, ou seja, definir a base de dimensão superior na qual será feita a separação dos padrões de informação. Não existe uma fórmula geral para definir esses parâmetros, porém, recentemente tem surgido uma série de artigos [25,26] com essa finalidade.

Uma estratégia alternativa de otimização dos parâmetros da RBFN é comparar modelos construídos com diferentes subconjuntos dentre um grupo definido de possíveis RBFs. Encontrar o melhor conjunto, por "tentativa e erro" é praticamente inviável, visto que existem $2^n - 1$ sub-conjuntos distintos em um conjunto com n RBFs, de modo que geralmente utiliza-se de métodos heurísticos para realizar essa tarefa. Um dos tipos mais simples dentre os métodos heurísticos de otimização é a "Forward Selection" [25,27].

O modelo de otimização por "forward selection" baseia-se em iniciar um sub-conjunto vazio de RBFs e ir adicionando RBFs de modo a diminuir (minimizar) o somatório do erro quadrático ou outro critério de minimização como a validação cruzada generalizada (por exemplo).

Outro modelo heurístico muito usado é fazer o oposto, ou seja, partir de um conjunto maior de RBFs e ir eliminando RBFs de modo a diminuir o erro (critério de minimização), conhecido como "backward elimination" .

É interessante comparar a seleção de subconjuntos de RBFs em uma RBFN com o modelo padrão para otimização em redes neurais artificiais, o qual envolve otimização por gradiente descendente de uma superfície não linear em um espaço multidimensional definido pelos parâmetros da rede [28]. Na otimização por subconjuntos ("foward selection" ou "backward selection") o algoritmo de otimização encontra em um espaço discreto de subconjuntos, um conjunto de RBFs com centros e raios fixos de modo a minimizar o erro de previsão. Os pesos não são selecionados, mas sim determinados em função do conjunto de RBFs definido pelo modelo de otimização. Por definição, "forward selection" é um algoritmo não linear de otimização para RBFNs, porém possui algumas vantagens:

- Não há necessidade de fixar o número máximo de RBFs;
- Os critérios de seleção dos modelos são tratáveis;
- Não há necessidade de poder computacional elevado.

O modelo de RBFN com otimização por "forward selection" foi aplicado na calibração de um conjunto de 367 espectros NIR de caldo de cana concentrado (xarope) para a determinação do BRIX (teor de sólidos em suspensão) em amostras cujo BRIX varia entre 64 e 74. As amostras foram divididas em dois grupos, sendo 267 amostras para a calibração e 100 para previsão, escolhidas de modo representativo, isto é, as amostras de cada grupo foram escolhidas de modo que cada grupo possuísse valores de BRIX em toda a faixa monitorada, sem extrapolação dos conjuntos de validação e previsão. Os espectros NIR das amostras de xarope de caldo de cana são mostrados na Figura 13.

Para a ortogonalização dos dados de entrada, utilizou-se um modelo de PCA[29] ("principal component analysis") usando os 6 primeiros componentes principais, os quais explicaram 99,995% da variância dos dados, ou seja, praticamente toda informação relevante.

Foram utilizadas 267 RBFs, tendo todas raios iguais a 9.6423×10^{-1}, 8.6465×10^{-1}, 7.9897×10^{-1}, 3.9774×10^{-1}, 4.1237×10^{-1} e 2.2724×10^{-1} em relação às 6 componentes principais (variáveis de entrada ortogonalizadas) respectivamente.

Os resultados obtidos pela modelagem com RBFN otimizada por "forward selection" nas amostras de calibração são mostrados na Figura 14 e para amostras de previsão são mostrados na Figura 15. Em ambas figuras (14 e 15), é mostrado o erro

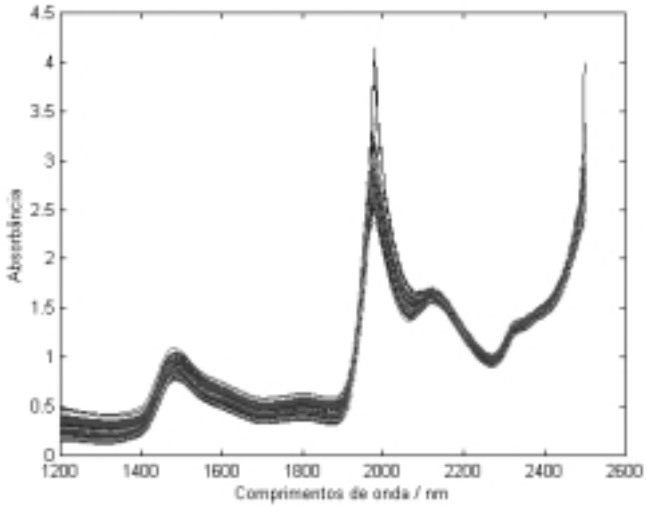

Figura 13. *Espectros NIR de xarope de caldo de cana.*

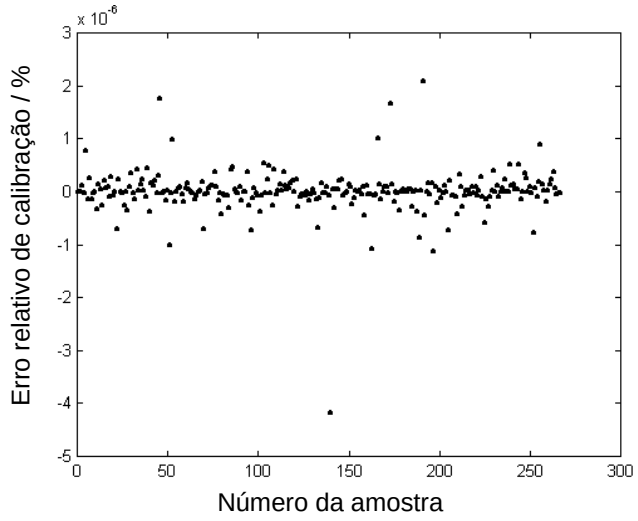

Figura 14. ***Erro relativo obtido com o modelo de RBFN otimizado por "foward selection" para amostras de calibração.***

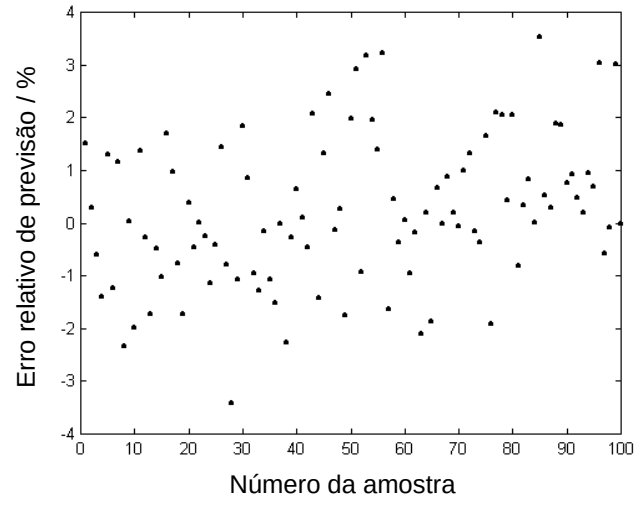

Figura 15. ***Erro relativo obtido com o modelo de RBFN otimizado por "foward selection" para amostras de previsão***

percentual relativo para as respectivas amostras. Observa-se valores de erros percentuais relativamente baixos, o que aumenta as possibilidades de utilização das redes com RBFs para a calibração multivariada em dados analíticos.

Adicionalmente, observa-se um melhor ajuste (erros percentuais relativos menores) para amostras de calibração que para as amostras de previsão, indicando um possível sobreajuste do modelo. Esse fato indica que para os dados em questão, embora os valores obtidos sejam aceitáveis, pode-se ainda otimizar a rede de RBFs para torná-la mais robusta, ou seja, fazer com que o modelo construído obtenha erros para as amostras de calibração e de previsão com mesma ordem de grandeza.

CONCLUSÕES

O uso de redes neurais, redes neurais com apodização e redes neurais com funções de base radial mostra-se como uma boa alternativa para a modelagem de sistemas em que a resolução dos sinais envolvidos é baixa ou a relação entre os sinais e a propriedade de interesse não é linear.

Especificamente em química analítica, a modelagem com redes neurais tem se mostrado eficaz na determinação quantitativa, simultânea ou não, de diversas espécies, através da análise com equipamentos de resposta multivariada (NIR, NIRR, UV-VIS, etc).

AGRADECIMENTOS

E. O. de Cerqueira agradece a FAPESP (proc. no. 97/06988-3) pela bolsa concedida.

REFERÊNCIAS

1. Zupan, J.; Gasteiger, J.; *Anal. Chim. Acta* **1991**, *248*, 1.
2. MacCulloch, W.; Pitts, W.; *Bull. Math. Biophy.* **1943**, *5*, 115.
3. Cartwright, H. M.; *Applications of Artificial Intelligence in Chemistry*; Oxford University Press, New York, 1995 p 13.
4. Katerman, G.; Smits, J. R.; *Anal. Chim. Acta.* **1993**, *277*, 179.
5. Gemperline, P. J.; *Chemom. Intell. Lab. Sys.* **1997**, *39*, 29.
6. Harrington, P. De B.; *Anal. Chem.* **1993**, *65*, 2167.
7. Long, J. R.; Gregoriou, V. G.; Gemperline, P. J.; *Anal. Chem.* **1990**, *62*, 1791.
8. Wythoff, B.; *Chemom. Intell. Lab. Sys.* **1993**, *18*, 115.
9. Jang, J. R. S.; Sun, C. T.; Mizutani, E.; *Neuro Fuzzy e soft computing*; Prentice Hall; Upper Saddle River, 1997; p 129.
10. Thomas, E. V.; Haale, D. M.; *Anal. Chem.* **1990**, *62*, 1091.
11. Otto, M.; *Statistics and Computer Application in Analytical Chemistry*; Wiley-VCH; Weinheim, DU; 1999; p 196.
12. De Barros Neto, B.; Scarminio, I. S.; Bruns, R. E.; *Planejamento e Otimização de Experimentos*; Editora da UNICAMP; Campinas; 1995; p 133.
13. Haykin, S.; *Neural Networks, a comprehensive foundation*; Prentice Hall; Upper Saddle River, New Jersey; 1999, p 218.
14. Hassibi, B.; Stork, D. G.*; In Advances in neural information processing systems 5*; Hanson, S. J.; Cowan, J. D.; Giles, C. L.; Ed.: Morgan Kaufmann, San Mateo, CA; 1993, p 164.
15. Cun, Y. Le; Boser, B.; Denker, J. S.; Solla, S. A.; *Optimal brain damage: Advances in Neural information processing systems, vol.2*; Morgan Kaufman; San Mateo, 1990, p 598
16. Poppi, R. J.; Massart, D. L.; *Anal. Chim. Acta* **1998**, *375*, 187.

17. Mello, C.; Poppi, R. J.; Andrade, J. C. De; Cantarella, H.; *Analyst* **1999**, *124*, 1669.
18. Fahlman, S. E.; Libiere, C.; *Advances in Neural Information Processing Systems, Vol. 2*; D. S. Touretzsky Ed.; Morgan Kaufmann, San Mateo; 1990; p 524.
19. Larsen, J.; Hansen, L. K.; *Generalization of regularized neural network models;* Proc. of the IEEE workshop on neural networks for signal Proc. IV; Piscataway; New Jersey; 1994; p 164.
20. Norgaard, M.*; Neural Network Based System Identification TOOLBOX for use with MATLAB™, version 1.1*;, Technical University of Denmark, DK; 1997.
21. Bishop C. M.*; Neural Networks for Pattern Recognition*; OXFORD University Press; New York, 1995; p. 448.
22. Giraud, F.; Salameh, Z. M.; *IEEE Transactions on Energy Conversion* 1999, *14*, 1572.
23. Wilkins M.F.; Boddy L.; Morris C. W.; Jonker R. R.; *Applied and Environmental Microbiology* **1999**, 65, 4404.
24. Walczak, B.; Massart, D.L.*; Anal. Chim. Acta* **1996**, *331*, 187.
25. Mark J. L. Orr; *Introduction to Radial Basis Function Networks*, Centre for Cognitive Science, University of Edinburg,2 Buccleuce Place, Edinburg EH8 9LW, Scotland, 1996.
26. Mark J. L. Orr; *Recent Advances in Radial Basis Function Network*, Institute for Adaptative and Neural computation Division of Informatics, Edinburgh University, Edinburg EH8 9LW, Scotland, 1999.
27. Stubbings, T.; Hutter, H. T.; *Chemmom. Intel. Labs. Syst.* **1999**, *49*, 163.
28. Hertz, J. Krough, A.; Palmer, R. G.; *Introduction to the Neural Network Computation*; Addison Wesley, Redwood City, CA, 1991.
29. Geladi, P.; Kowalski, B.; *Anal. Chim. Acta* **1986**, *185*, 1.