1° Congresso Brasileiro de Redes Neurais

Escola Federal de Engenharia de Itajubá
Itajubá, 24 a 27 de outubro de 1994

ALGORITMO RÁPIDO DE TREINAMENTO DE REDES NEURAIS PARA SISTEMAS DE CONTROLE EM TEMPO REAL

L.E. Borges da Silva
A.P. Alves da Silva
G. Lambert-Torres
E. R. Ribeiro

Grupo de Inteligência Artificial
Escola Federal de Engenharia de Itajubá

Abstract - The long time generally required for training ANN has been a critical problem for the utilization of this technology in real-time. The generalized delta rule with backward error propagation (backpropagation) has been the most used training algorithm for ANN in control applications. However, despite many attempts to improve the performance of this learning algorithm, there is still no efficient and reliable method to train a multilayer perceptron. The main problem is high nonlinearity of the error function. This paper presents an alternative method for load information into a multilayer perseptron. The idea is to use successive quadratic approximations for the error function, until getting into the proximity of the global minimum. This technique was applied to Control System in order to obtain an adaptive behaviour.

1- Introdução

O uso de Redes Neurais em sistemas de controle, tem se mostrado como uma alternativa muito interessante. Porém, problemas ainda existentes impedem sua utilização no controle adaptativo em tempo real, sendo que, o tempo de treinamento da rede é a restrição mais significativa. A técnica de treinamento mais popular, no momento, a retro-propagação do erro, embora muito simples é uma técnica que consome muito tempo.

Este trabalho apresenta um esquema de controle adaptativo, que usa Redes Neurais para identificação e controle de um acionamento de motor de corrente contínua alimentado por um conversor tiristorizado.

O algoritmo de adaptação e a estratégia de controle são apresentadas. A metodologia proposta foi simulada e os resultados são mostrados. Para alguns problemas detectados, como por exemplo o tempo de convergência, mais precisamente o número de iterações antes do algoritmo atingir um certo valor mínimo de erro, é proposta uma possível solução. Um novo método de carregamento da informação em um Perceptron Multicamadas é apresentado. A idéia é usar aproximações quadráticas sucessivas para a função erro, até que a proximidade de um mínimo global possa ser atingido.

Esta técnica permite o uso do algoritmo "Mínimos Quadrados Recursivos" para computar o mapeamento associativo ótimo. O algoritmo proposto de treinamento para perceptrons multicamadas resolve os dois principais problemas: tempo longo de treinamento e algoritmos complexos de treinamento.

2- Sistemas de Controle Adaptativo

Em muitas aplicações de controle se torna muito difícil representar a dinâmica do processo em equações matemáticas precisas. Processos reais sempre incluem no modelo incertezas tais como variações de parâmetros ou não linearidades, que não são conhecidas no momento do projeto dos controladores.

O controle adaptativo mostrado na figura.1, foi projetado de forma a obter um alto desempenho, independente da variação das características do processo. O problema do

controle é ajustar dinamicamente os parâmetros do controlador de tal forma que a saída da planta, c(k), siga o sinal de referência, r(k). Para isto, a identificação do modelo é utilizada pelo algoritmo adaptativo para o treinamento do controlador.

Muitos são os problemas associados com os controles adaptativos tradicionais; por exemplo, a não existência de uma metodologia aplicável a uma grande quantidade de problemas diferentes.

Neste trabalho apresentamos uma alternativa ao controle adaptativo tradicional para um acionamento de motor de corrente continua.

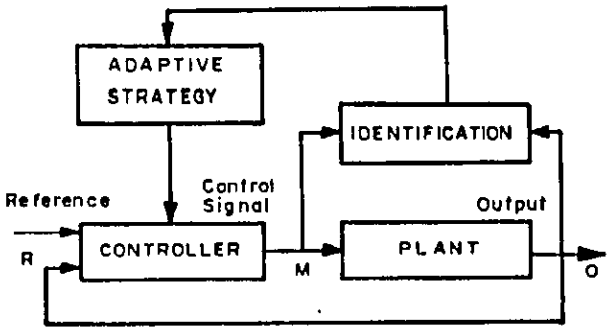

Fig.1 - Arquitetura Tradicional de Controle Adaptativo

3- Redes Neurais em Sistemas de Controle

O elemento básico de processamento de uma rede, o Neurônio, é representado pela equação:

$$h_{out,i} = f\left(\sum_{j=1}^{m} h_{in,j} \cdot w_{ji} - \theta_i\right)$$

O parâmetro θ_i serve como limite ou polarização, $h_{in,j}$ é o conjunto das variáveis de entrada e w_{ji} é o conjunto de pesos das interconecções partindo do elemento **j** para o elemento **i**.

A rede é definida pelo número e o modo como os neurônios são interconectados, pela função de ativação, e o algoritmo de treinamento.

Neste artigo, será considerado somente redes com encaminhamento para frente, também chamadas redes multicamadas (fig.2), devido sua capacidade de aprender características de sistemas através do "mapeamento" não-linear. Esta arquitetura de rede é a mais utilizada para aplicações em controle

A entrada dos neurônios monitoram os sinais externos que são convertidos em saídas, de acordo com os pesos e funções de ativação apropriadas. É sabido que as camadas internas da rede são responsáveis pelo mapeamento não-linear entre sinais de entrada e saída e supressão de padrões de ruídos que porventura possam existir.

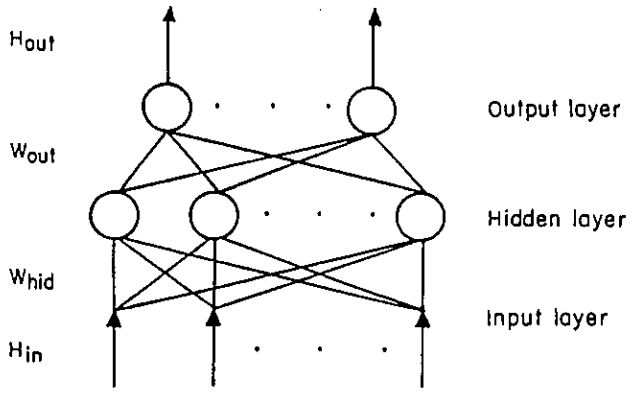

Fig.2 - Rede Neural Multicamadas

A entrada dos neurônios monitoram os sinais externos que são convertidos em saídas, de acordo com os pesos e funções de ativação apropriadas. É sabido que as camadas internas da rede são responsáveis pelo mapeamento não-linear entre sinais de entrada e saída e supressão de padrões de ruídos que porventura possam existir.

Durante o treinamento da rede, os pesos das conecções são ajustados em função da comparação entre o valor real saído da rede e o valor desejado, até que o erro caia dentro de limites especificados.

O método de treinamento mais utilizado, atualmente, é o algoritmo de retro-propagação do erro [1].

4- Identificação e Controle Adaptativo com Redes Neurais

O sistema proposto compreende três partes. A primeira é o treinamento da rede neural para que esta represente corretamente a resposta do sistema. A segunda usa do modelo do sistema identificado, pela rede, para o procedimento de ajuste do controlador. A terceira, quando o sistema estiver funcionando em operação normal, para garantir um bom desempenho, a rede é retreinada continuamente.

A especificação da estrutura da rede (número de entradas, número de saídas, número de camadas internas e número de neurônios), o padrão de treinamento e a escolha dos parâmetros do algoritmo de treinamento, são os principais problemas do projeto de um sistema de controle com redes neurais. Os parâmetros de treinamento possuem um efeito decisivo no desempenho do sistema.

A - Identificação do Processo

O esquema de identificação, mostrado na fig.3, ilustra a posição, em paralelo, da rede com o processo. A cada iteração o padrão de entrada é montado pelo sinal de referência e pelos valores anteriores das amostras da saída do processo. Assim, o problema de identificação consiste no ajuste adequado dos pesos da rede de forma a minimizar o erro E_i (fig.3), isto é, a diferença entre o valor correto do sinal c(k) e da saída da rede c^(k) [4].

Durante o treinamento da rede de identificação, a intervalos regulares, o erro global é calculado e comparado com o valor mínimo desejado que irá determinar o grau de aprendizado da rede.

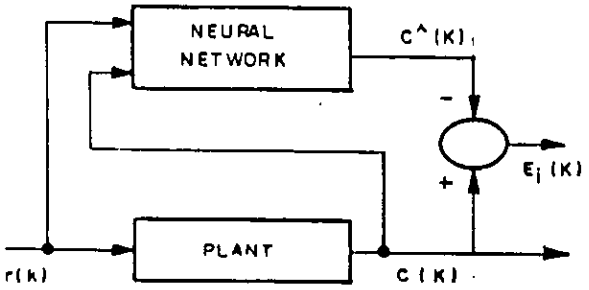

Fig.3 - Esquema de Identificação usando Redes Neurais

B - Implementação do Controle

O objetivo do controle é gerar um sinal m(k), que conduza o processo a um certo ponto de operação desejado. O treinamento do controlador, ilustrado na fig.5, necessita da informação sobre o erro na saída do controlador neural a cada instante. Porém, somente o erro final E_c, entre a saída do processo e a saída desejada, para uma dada referência, está disponível, deste modo, usamos a rede neural como modelo do processo, retro-propagando o erro E_c através da rede de identificação para conseguir um erro equivalente para o controlador.

O ajuste do controlador é similar ao procedimento de identificação, mas agora, o erro global, dado por E_c tem que atingir o valor mínimo desejado. De modo análogo, o sinal de referência tem que seguir as mesmas condições do procedimento de identificação.

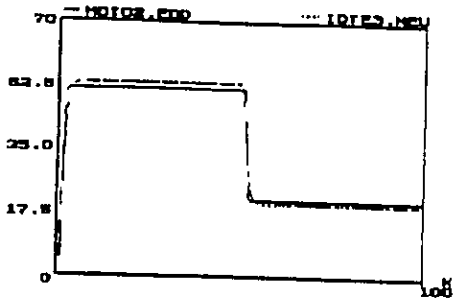

Fig.4 - Resultado da Identificação

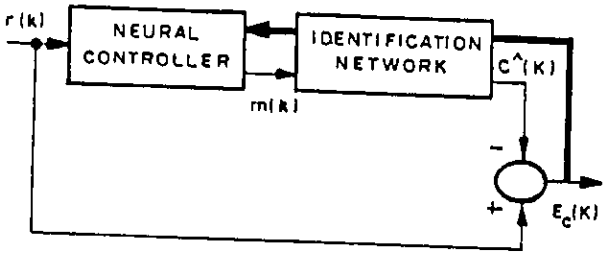

Fig.5 - Esquema de Treinamento do Controlador Neural

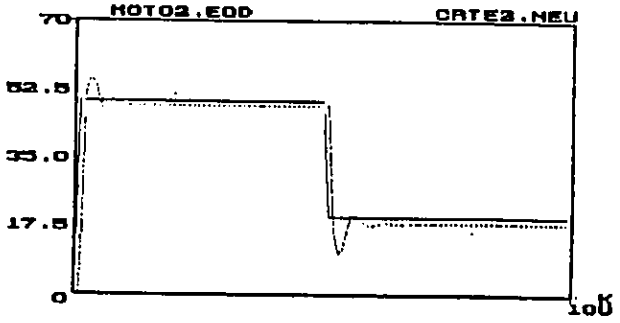

Fig.6 - Resultado do treinamento do Controlador

5- Algoritmo de Treinamento Proposto

O uso de Redes Neurais em Sistemas de Controle, como já mencionado anteriormente, tem se mostrado como

uma ferramenta poderosa para manipular não-linearidades e outros problemas. Porém, para aplicar o algoritmo em sistemas reais, o tempo de treinamento se mostra como uma restrição importante.

O algoritmo de retro-propagação, embora muito simples de ser implementado, facilitando a simulação do sistema de controle proposto, consome muito tempo invalidando sua aplicação em tempo real. Não possui escalamento adequado, isto é, o tempo de treinamento aumenta exponencialmente na medida que o número de neurônios aumenta. Requer uma escolha adequada dos parâmetros de treinamento: taxa de treinamento, constante de momento e inclinação da função de ativação. E como um método de passos descendentes, pode parar em mínimo local. Estes são alguns dos inúmeros problemas associados com este algoritmo.

A despeito dos esforços para melhorar o desempenho da Regra Delta Generalizada com retro-propagação do erro, não existe ainda um método eficiente e confiável para treinar Perceptrons Multicamadas. O principal problema é o alto grau de não linearidade da função de erro.

Para contornar os problemas mencionados acima, um novo método para carregar a informação em um Perceptron Multicamada será apresentado. A idéia é usar aproximações quadráticas sucessivas para a função de erro, até chegar nas proximidades de um mínimo global. O algoritmo dos mínimos quadrados poderá ser utilizado na computação do mapeamento associativo ótimo. Este esquema associativo permite a inclusão seqüencial de novos padrões sem recalcular a expressão pseudoinversa para o conjunto completo de treinamento.

Na próxima sessão, o algoritmo proposto para Perceptrons Multicamadas será apresentado, o qual contorna dois grandes problemas: longo tempo de treinamento e algoritmo complexo de treinamento.

6- Técnica de Aprendizado Supervisionado

O método apresentado está baseado na Estimativa Ótima de Treinamento (OET)[5]. OET é uma técnica de aprendizado supervisionado para perceptrons multicamadas. Resultados preliminares indicam que a técnica OET pode ser muito mais rápida e precisa que outras técnicas.

O conjunto de padrões de entrada e saída são representados na forma de matrizes. Um mapeamento não-linear que associa padrões de treinamento de entrada com padrões de treinamento de saída, é encontrado resolvendo sucessivamente um sistema linear usando métodos diretos, que é de fato um procedimento baseado no método mínimos quadrados não-linear (aproximação local da função erro por uma função quadrática, e a exata aproximação da mesma por uma função aproximada). A pseudoinversa de Moore-Penrose é solucionada explicitamente em diferentes passos da OET. Uma importante vantagem desta técnica é que o problema do escalamento é muito bem definido. Baseado no número de operações envolvidas ($[rm]^2$ onde **r** é o número de padrões de treinamento de entrada e **m** a dimensão deles), é fácil estimar a taxa de aprendizado da OET com relação ao tamanho da rede.

A. Estimativa Ótima de Treinamento

A idéia geral da OET pode ser resumida usando a fig.6 [3]. Nesta figura hin,j (j=0,1, ..., m) é o conjunto das variáveis de entrada (hin,0 representa uma entrada adicional com valor constante igual a 1).

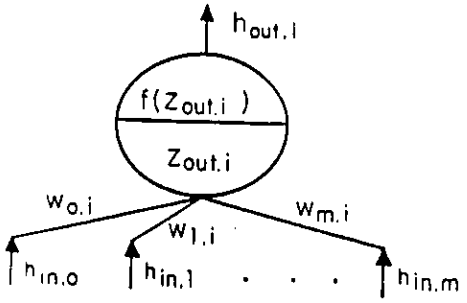

Fig 6 - Representação de um Neurônio

É bem conhecido que a propagação das entradas é feita pela soma das entradas ponderadas, isto é,

$$z_{out,i} = \sum_{j=0}^{m} h_{in,j} w_{ji}$$

O valor de hout,i é calculado pela aplicação da função de ativação hout,i = f(zout,i).

Para executar OET, um valor desejado (especificado) de saída hd-out,i, é retropropagado através da função de ativação e o resultado correspondente imediato desejado, zd-out,i , será obtido. Para executar esta operação, é necessario uma função contínua e inversível tal como a tangente hiperbólica. Esta função é adequada para OET, desde que as saídas desejadas possam ser positivas ou negativas. Assim, o conjunto das entradas e resultados intermendiários podem ser interrrelacionados para calcular a estimativa ótima, para um conjunto de interconecções ponderadas, no sentido dos mínimos quadrados.

A lista de símbolos utilizados para explicar o algoritmo será agora descrita (de acordo com a fig.2):

Hd-out -- matriz de saída desejada (r x n) ;
Hin -- matriz de entrada (r x m+1) ;
Zd-out -- **Hd-out** retro-propagado através da camada de saída (matriz r x n);
Wd-out -- matriz (m+1 x n) de interconecções ponderadas ligando a camada interna com a camada de saída;
Hd-hid -- matriz (r x m+1) ;
Zd-hid -- **Hd-hid** retro-propagado através da camada interna (matriz r x m+1);
Whid -- matriz (m+1 x m+1) de interconecções ponderadas ligando a camada de entrada com a camada interna ;
Zhid -- produzida pela multiplicação de **Hin** por **Whid** (matriz r x m+1);
Hhid -- **Zhid** propagado através da camada interna (matriz r x m+1);
Zout -- produzido pela multiplicação de **Hout** por **Wout** (matriz r x m+1);
Hout -- **Zout** propagado através da camada de saída (matriz r x n);
r -- número de padrões entrada/saída do conjunto de treinamento;
m+1 -- número de reurônios de entrada (que é iqual ao número de neurônios da camada interna); e
n -- número de neurônios de saída.

B. Passos de treinamento da Rede

O procedimento para treinamento de uma rede com uma camada interna será agora descrito. A extenção deste procedimento de treinamento para mais que uma camada interna é imediato.

1) Calcular o valor inicial de **Wout** (este deverá ser o valor final de **Wout** se camadas internas não são usadas):

$$W_{out} = H_{in}^{\Gamma} Z_{d-out}$$

onde

$$Z_{d-out} = \tanh^{-1}(H_{d-out})$$

e

$$H_{in}^{\Gamma} = (H_{in}^{\Gamma} H_{in})^{-1} H_{in}^{t}$$

assumindo r maior que m+1 (sistema sobre dimensionado). O símbolo Γ representa a inversa generalizada ou Moore-Penrose pseudo inversa.

2) Obter **Hd-hid** que irá produzir **Zd-out** dado **Wout** ,

- se n = m+1, $H_{d-hid}^{t} = (W_{out}^{t})^{-1} Z_{d-out}^{t}$

assumindo que W_{out}^{t} é não-singular

- se n > m+1 $H_{d-hid}^{t} = (W_{out}^{t})^{\Gamma} Z_{d-out}^{t}$,

$$(W_{out}^{t})^{\Gamma} = (W_{out} W_{out}^{t})^{-1} W_{out}$$

e

- se n < m+1 $H_{d-hid}^{t} = (W_{out}^{t})^{\Gamma} Z_{d-out}^{t}$

$$(W_{out}^{t})^{\Gamma} = W_{out} (W_{out}^{t} W_{out})^{-1}$$

caso indeterminado.

3) Normalizar os elementos de **Hd-hid** para garantir que eles irão se situar dentro do limite estipulado para as saídas das funções de ativação da camada interna,

Hd-hid = **Hd-hid** 0.99/ $|\lambda|$

onde λ é o elemento de **Hd-hid** com maior amplitude. Um fator de multiplicação, em torno de 0.99, é usado para obter valores finitos de **Zd-hid** , e tirar vantagem da não-linearidade da tangente hiperbólica.

4) Produzir **Zd-hid** usando a inversa da tangente hiperbólica,

$$Z_{d-hid} = \tanh^{-1}(H_{d-hid})$$

5) Dado **Hin** e **Zd-hid** , calcular **Whid** ,

$$W_{hid} = H_{in}^{\Gamma} Z_{d-hid}$$

6) Obter o valor da saída da camada interna;

$$Z_{hid} = H_{in} W_{hid}$$

e

$$H_{hid} = \tanh(Z_{hid})$$

7) Recalcular **Wout** ,

$$W_{out} = H_{hid}^{\Gamma} Z_{d-out}$$

onde

$$H_{hid}^{\Gamma} = (H_{hid}^{t} H_{hid})^{-1} H_{hid}^{t}$$

O treinamento da rede será finalizado no passo 7), isto é, uma vez as matrizes de interconecções ponderadas **Whid** e **Wout** estejam ótimamente calculadas usando o critério dos mínimos quadrados.

Para obter a saída do ANN para um dado conjunto de entradas (classificação/operação on-line), os seguintes passos são necessários:

1) Combinar **Hin** e **Whid** para produzir **Zhid** ,

Zhid = **Hin** **Whid**

2) Tratar **Zhid** com a função de ativação (tangente hiperbólica),

$$H_{hid} = \tanh(Z_{hid})$$

3) Combinar **Hhid** e **Wout** para produzir **Zout**

Zout = **Hhid** **Wout**

4) Tratar **Zout** com a função de ativação para gerar **Hout** (saída da rede),

$$H_{out} = K \tanh(Z_{out})$$

onde K=1.0101 (fator de normalização da saída (1/0.99)).

7- Conclusão

Rede Neural Artificial (ANN) possui importantes vantagens, as quais são de bastante utilidade na modelagem e controle de sistemas não-lineares[2]. A capacidade de aprendizado, generalização, paralelismo intrínceco, e capacidade de aproximar qualquer sistema não-linear, são algumas destas vantagens.

Devido aos avanços em ANN, é possível no momento o uso deste tipo de sistema inteligente em aplicações em tempo real. Uma das áreas da tecnologia que mais se beneficiaram disto é a de sistema de controle. O longo tempo, geralmente requerido, para o treinamento da ANN foi por muito tempo um problema crítico para utilização desta tecnologia em tempo real.

A Regra Delta Generalizada com retro-propagação do erro é o algoritmo mais usado em aplicações de controle. Embora muito trabalho tenha sido feito para melhorar o desempenho deste algoritmo de aprendizado, não existe ainda um método eficiente para treinamento de perceptrons multicamadas. O principal problema é o alto grau de não-linearidade da função de erro.

Inicialmente, a rede foi utilizada para reproduzir o comportamento entrada/saída do processo, isto é, um conversor tiristorizado controlando um motor DC com excitação independente. Então, uma segunda rede foi treinada para ser o controlador que irá manipular a energia fornecida ao motor para se obter a resposta desejada (representada pelo sinal de referência). Como o modelo foi identificado por uma rede neural, o erro final é retro-propagado através do modelo afim de se obter um erro equivalente para treinamento do controlador neural. Ambos, o controlador neural e a rede identificadora aprende continuamente durante a operação normal, pela monitoração da informação vinda dos sensores e relacionando com a saida desejada.

8- Agradecimentos

Este trabalho foi possível devido a ajuda do CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico e FAPEMIG - Fundação de Amparo a Pesquisa de Minas Gerais.

9- Referências

[1] D.E. Rumelhart, G.E. Hinton, and R.J. Williams - "Learning internal representations by error propagation in parallel distributed processing", Exploration in the Microestructure of Cognition, Vol.1, Foundations, MIT Press, 1986, pp.318-362.

[2] K.J. Hunt, D. Sbarbaro, R. Zbikowski, and P.J. Galwtrop - "Neural Networks for Control System - A Survey" Automatica, vol.28, no.6, pp. 1083-1113, 1992.

[3] A.P. Alves da Silva, A.M. Leite Silva, J.C.S. de Souza, and M.B. do Coutto Filho - "State Forecasting Based on Artificial Neural", 11th Power Systems Computation Conference, PSCC, Avignon, August 1993.

[4] L.E. Borges da Silva, G. Lambert-Torres, E.C. Saturno, A.P. Alves da Silva, and G. Olivier - "Neural Net Adaptive Schemes for DC Motor Drives", IEEE Industry Applications Society Conference, Toronto, October 1993.

[5] J.F. Shepanski- "Fast learning in Artificial Neural Systems: Multilayer Perseptron Training using Optimal Estimation ", Proc. IEEE 2nd Intern. Conf. Neural Nets, San Diego, Jul. 1988, Vol.1, pp.465-472.