

## Aceleração da Convergência do Algoritmo de Treinamento SMC - BP através do Ajuste da Superfície de Deslizamento

Gustavo G. Parma, Antônio P. Braga, Benjamim R. Menezes  
Departamento de Eletrônica  
Universidade Federal de Minas Gerais  
E-mails: {parma, apbraga,brm}@cpdee.ufmg.br

### Abstract

*An improved algorithm based on the SMC-BP[3] algorithm using the well known sliding mode theory is proposed. The sliding mode surface is obtained from the squared error of the training set and then the weight update equations were derived using the gradient's addition. The bounds for the gains (learning rates) are obtained from the sliding surface so that system stability is guaranteed. The proposed algorithm is applied to function approximation problems and results are compared with the standard backpropagation and Rprop algorithms in terms of speed of learning.*

### 1. Introdução

Algumas variações do algoritmo padrão de backpropagation são encontradas na literatura. Dentre elas pode-se citar o momentum [1], quickprop [2], Rprop [5] e vários algoritmos que utilizam algum sistema de ganho adaptativo, dentre outros. Os algoritmos propostos objetivam aumentar a velocidade de convergência do treinamento pois sabe-se que esta é uma das deficiências do algoritmo padrão de backpropagation.

Parma, Menezes e Braga, em 1998 [4], apresentaram um algoritmo de treinamento *on-line* baseado na teoria de Sistemas de Estrutura Variável (SEV). Em trabalho posterior [3], os autores propuseram a utilização da teoria de SEV para o treinamento *off-line*, determinando uma superfície de controle que possibilita controlar a direção de ajuste dos pesos da Rede Neural Artificial (RNA) a ser treinada. O algoritmo proposto foi denominado de SMC-BP. Neste trabalho é proposta uma nova superfície de deslizamento a qual, além de possibilitar o controle da direção de ajuste dos pesos, altera o módulo de atualização dos pesos em função do somatório dos gradientes dos erros durante a apresentação do conjunto de treinamento. Desta forma, possibilita-se um treinamento mais rápido do que aquele obtido com o algoritmo padrão de backpropagation.

O novo algoritmo proposto foi comparado com o algoritmo Rprop e backpropagation padrão em problemas de aproximação de funções. Os resultados mostram que a utilização desta nova superfície propicia um treinamento

da ordem de grandeza, ou mais rápido, do obtido com o algoritmo Rprop.

É realizado um estudo teórico dos limites da taxa de treinamento de forma a garantir a convergência do algoritmo proposto, formalizando a proposta.

Dividiu-se este artigo em quatro partes principais: inicialmente é feita a proposta do algoritmo, logo após é feito o estudo matemático utilizando a teoria de Sistemas de Estrutura Variável. São apresentados resultados obtidos no problema de aproximação de funções na terceira seção e, logo após, são feitas as conclusões finais sobre o trabalho.

### 2. Algoritmo Proposto

Durante o desenvolvimento matemático apresentado neste artigo, serão utilizadas as variáveis definidas na tabela 1.

Table 1: Variáveis utilizadas no desenvolvimento matemático

$R_i(k)$	Saída linear do neurônio $i$ da camada escondida
$Y_{Hi}(k)$	Saída do neurônio $i$ da camada escondida, após a aplicação da função de ativação
$V_j(k)$	Saída linear do neurônio $j$ da camada de saída
$Y_j(k)$	Saída do neurônio $j$ da camada de saída, após a aplicação da função de ativação
$f_H(\circ)$	Função de ativação da camada escondida
$f(\circ)$	Função de ativação da camada de saída
$f'_H(\circ)$	derivada da função de ativação da camada escondida em função da saída linear
$f'(\circ)$	derivada da função de ativação da camada de saída em função da saída linear
$W$	Matriz de pesos da camada de saída
$Z$	Matriz de pesos da camada escondida
$W_{ji}$	Peso que conecta a saída $i$ da camada escondida ao neurônio $j$ da camada de saída
$Z_{ih}$	Peso que conecta a entrada $h$ ao neurônio $i$ da camada escondida

O algoritmo proposto é, originalmente, desenvolvido

para uma Multilayer Perceptron (MLP) com uma camada escondida e uma camada de saída. Entretanto, ele pode ser facilmente estendido para mais camadas.

Como descrito anteriormente, o algoritmo proposto é baseado na teoria de Sistemas de Estrutura Variável (SEV) e controle por Modos Deslizantes (CMD) [8]. Desta forma, definem-se as seguintes superfícies de deslizamento:

- Camada de saída:

$$S_{ji(k)} = C \cdot X_{1ji(k)} + X_{2ji(k)} \quad (1)$$

onde,

$$X_{1ji(k)} = \sum_{l=1}^k \frac{\partial E_{(l)}}{\partial W_{ji(l)}} \quad (2)$$

$$X_{2ji(k)} = \frac{\partial E_{(k)}}{\partial W_{ji(k)}} \quad (3)$$

$$X_{1ji(k)} = 0/\forall k < 1$$

- Camada escondida:

$$S_{ih(k)} = C_H \cdot X_{1ih(k)} + X_{2ih(k)} \quad (4)$$

onde,

$$X_{1ih(k)} = \sum_{l=1}^k \frac{\partial E_{(l)}}{\partial Z_{ih(l)}} \quad (5)$$

$$X_{2ih(k)} = \frac{\partial E_{(k)}}{\partial Z_{ih(k)}} \quad (6)$$

$X_{2ih(k)} = 0/\forall k < 1$ .  $k$  representa o vetor apresentado dentro de uma *epoch*.

Com base nestas superfícies de deslizamento, pode-se definir as seguintes regras de atualização dos pesos:

- Camada de saída:

$$\Delta W_{ji(k)} = -\alpha \cdot |X_{1ji(k)}| \cdot \text{sgn}(S_{ji(k)}) \quad (7)$$

Camada Escondida:

$$\Delta Z_{ih(k)} = -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot f'_H(R_{ih(k)}) \quad (8)$$

Observa-se nas equações 7 e 8 que a direção de ajuste dos pesos é fornecida pelas superfícies de deslizamento definidas pelas equações 1 e 4, enquanto que o módulo do ajuste é obtido através das equações 2 e 5. Ou seja, o módulo do ajuste dos pesos é obtido através do somatório dos gradientes dos erros durante a apresentação do conjunto de treinamento à RNA. Desta forma, durante o treinamento, existe uma "memória" dos vetores já apresentados à rede, possibilitando um aumento da velocidade de treinamento enquanto que a superfície de deslizamento garante a convergência do sistema.

## 2.1. Estudo da Convergência

É necessário, agora, determinar os limites dos ganhos  $\alpha$  e  $\beta$ , utilizados nas equações 7 e 8, para garantir a estabilidade do sistema proposto.

A convergência do sistema, segundo Sapturk [6], pode ser estudada pela análise da seguinte equação:

$$|S_{(k)}| < |S_{(k-1)}| \quad (9)$$

onde  $S_{(k)}$  está definida pela equação 1 e 4 para a camada de saída e camada escondida, respectivamente.

Substituindo-se as equações 2 e 3 em 1:

$$S_{ji(k)} = (C + 1) \cdot \frac{\partial E_{(k)}}{\partial W_{ji(k)}} + C \cdot \sum_{l=1}^{k-1} \frac{\partial E_{(l)}}{\partial W_{ji(l)}}$$

utilizando-se a equação definida no algoritmo padrão de backpropagation e o fato de que  $Y_{j(k)} = \Delta Y_{j(k)} + Y_{j(k-1)}$ , tem-se:

$$S_{ji(k)} = (C + 1) \cdot \Delta Y_{j(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} - (C + 1) \cdot (Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + C \cdot \sum_{l=1}^{k-1} \frac{\partial E_{(l)}}{\partial W_{ji(l)}}$$

Considerando que a variável  $\Delta Y_{j(k)}$  pode ser calculada, através das equações básicas da RNA MLP, como  $\Delta Y_{j(k)} = f'(V_{j(k)}) \cdot Y_{Hi(k)} \cdot \Delta W_{ji(k)}$  e utilizando a equação 7, finalmente obtém-se:

$$S_{ji(k)} = -\alpha \cdot \text{sgn}(S_{ji(k)}) \cdot X_{(k)} + Y_{(k)} \quad (10)$$

onde  $X_{(k)}$  e  $Y_{(k)}$  são definidos como,

$$X_{(k)} = (C + 1) \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2 \cdot \left| \sum_{l=1}^k \frac{\partial E_{(l)}}{\partial W_{ji(l)}} \right| \quad (11)$$

$$Y_{(k)} = - (C + 1) \cdot (Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} \quad (12)$$

$$Y_{Hi(k)} + C \cdot \sum_{l=1}^{k-1} \frac{\partial E_{(l)}}{\partial W_{ji(l)}}$$

onde  $X_{(k)} > 0$  e  $\alpha > 0$ .

Considerando a equação 9 e a equação 10, a convergência do sistema será garantida se:

$$\alpha < \min \left\{ \frac{|Y_{(k)}|}{X_{(k)}}, \frac{|Y_{(k-1)}| - |Y_{(k)}|}{X_{(k-1)} - X_{(k)}} \right\} \quad (13)$$

Procedendo de maneira similar para a camada escondida, utilizando as equações 5, 6, 4 e a equação, obtida no algoritmo padrão de backpropagation, que define o gradiente do erro para a camada escondida, tem-se:

$$S_{ih(k)} = - (C_H + 1) \cdot f'_H(R_{ih(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k)}) \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E_{(l)}}{\partial Z_{ih(l)}}$$

Utilizando a definição de  $Y_{j(k)}$ , o fato de que  $\Delta Y_{j(k)} = f'(V_{j(k)}) \cdot f'_H(R_{i(k)}) \cdot W_{ji(k)} \cdot T_{h(k)} \cdot \Delta Z_{ih(k)}$  e a equação 8, obtém-se:

$$S_{ih(k)} = -\beta \cdot \text{sgn}(S_{ih(k)}) \cdot X_{(k)} + Y_{(k)}$$

onde as variáveis  $X_{(k)}$  e  $Y_{(k)}$  são, agora, definidas como:

$$X_{(k)} = |X_{1ih(k)}| \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \quad (14)$$

$$f'_H(R_{i(k)}) \cdot (C_H + 1) \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2$$

$$Y_{(k)} = -(C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \quad (15)$$

$$\sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot$$

$$W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial Z_{ih(l)}}$$

De maneira análoga à utilizada para a camada de saída, a convergência do sistema será garantida se:

$$\beta < \min \left\{ \frac{|Y_{(k)}|}{X_{(k)}}, \frac{|Y_{(k-1)}| - |Y_{(k)}|}{X_{(k-1)} - X_{(k)}} \right\} \quad (16)$$

### 3. Resultados de Simulação

O algoritmo proposto foi utilizado no problema de aproximação de função, tendo sido, também, implementados os algoritmos backpropagation padrão e Rprop. Foram utilizadas duas funções básicas para os testes:  $f_1 = -3.2 \cdot x^2 + 3.2 \cdot x$  e  $f_2 = x^4$ , onde  $x$  é considerada a entrada da RNA MLP e  $f_1$  ou  $f_2$  são as saídas. Foi utilizada uma RNA MLP com três nodos na camada escondida e um nodo de saída para cada uma das funções.

Optou-se por implementar o algoritmo Rprop por este apresentar alta velocidade de convergência [7]. Entretanto, não existe uma metodologia bem definida para o ajuste dos parâmetros envolvidos no algoritmo Rprop, bem como não é feita uma abordagem matemática sobre o estudo da convergência durante o treinamento. A nova proposta para o algoritmo SMC - BP, além da velocidade de convergência superior à obtida com o algoritmo Rprop, como pode ser observado nas figuras 1 e 3, possibilita um melhor ajuste dos parâmetros envolvidos no treinamento através de seu formalismo matemático.

Pode-se observar pelas figuras 1 e 3 que o algoritmo proposto (SMC-BP) é mais rápido do que o algoritmo RProp para os problemas apresentados. Nas figuras 2 e 4 é mostrada a característica de generalização da RNA treinada pelos três algoritmos. Pode-se observar que o comportamento final da rede é similar em todos os casos.

Além da inerente alta velocidade de convergência existente em sistemas de controle por modos deslizantes, a alta velocidade do treinamento é uma consequência da utilização do somatório dos gradientes do erro durante a apresentação do conjunto de treinamento à RNA. Desta

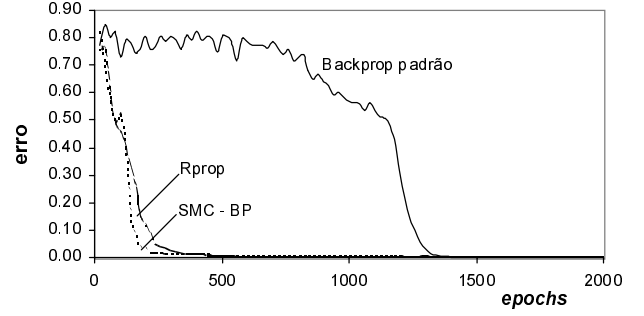


Figure 1: Erro durante o treinamento para a função  $f_1 = -3.2 \cdot x^2 + 3.2 \cdot x$ .

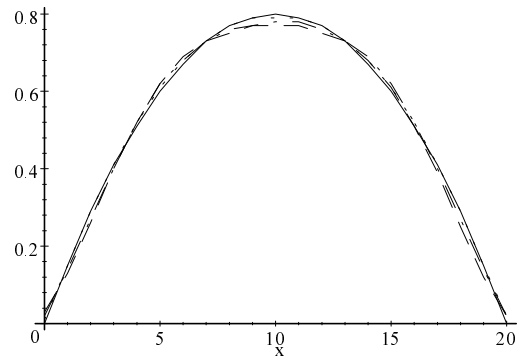


Figure 2: Teste de generalização para a função  $f_1 = -3.2 \cdot x^2 + 3.2 \cdot x$

forma, o treinamento mantém, sempre, uma memória dos vetores já apresentados, otimizando o treinamento.

Como mostrado anteriormente, a direção de atualização dos pesos é controlada pela superfície de deslizamento, composta pelo gradiente atual do erro e o seu somatório na *epoch* (equações 1 e 4). Desta forma, pelo ajuste das constantes que definem a inclinação da superfície ( $C$  e  $C_H$ ), controla-se a direção de atualização dos pesos.

### 4. Conclusões

Uma melhoria no algoritmo de treinamento SMC-BP, baseado na teoria de controle de sistemas de estrutura variável é proposta. Os resultados apresentados mostram que essa nova metodologia, além de possibilitar um estudo matemático da convergência do sistema, possibilita um acréscimo na velocidade de treinamento superior à obtida com o algoritmo de RProp. O aumento da velocidade de convergência desta nova proposta para o algoritmo SMC-BP padrão é decorrente da escolha das variáveis utilizadas na superfície de deslizamento. Nesta nova proposta, o módulo de variação dos pesos é função do somatório dos gradientes dos erros durante a apresentação do conjunto de treinamento, desta forma, o algo-

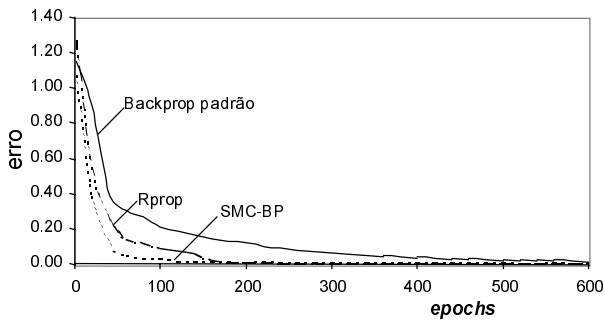


Figure 3: Erro durante o treinamento para a função  $f_2 = x^4$

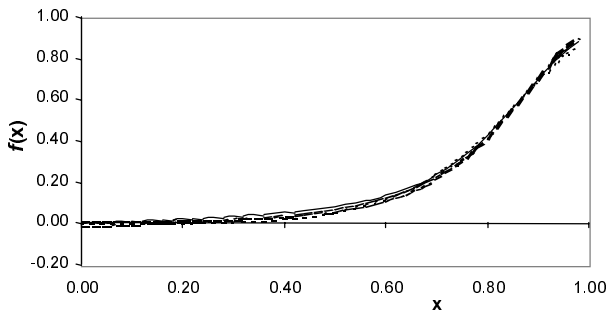


Figure 4: Teste de generalização para a função  $f_2 = x^4$

ritmo de treinamento mantém uma memória de todos os vetores já apresentados à RNA.

Garantidos os limites para os ganhos do treinamento ( $\alpha$  e  $\beta$ , equações 13 e 16), o treinamento da RNA MLP será garantido pelas superfícies de deslizamento para a camada de saída e escondida, definidas pelas equações 1 e 4, respectivamente.

## References

- [1] Bertsekas, Dimitri P. and John Tsitsiklis. *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.
- [2] Fahlman, S. E. "Faster-learning variations on backpropagation: an empirical study." *Proceedings of the 1988 Connectionist Models Summer School, Pittsburgh*, edited by D. Touretzky, et al. 38–51. San Mateo, CA: Morgan Kaufmann, 1988.
- [3] Parma, Gustavo G., et al. "Improving backpropagation with sliding mode control." *Proceedings of the Vth Brazilian Symposium on Neural Networks*. 8–13. Belo Horizonte, Brazil: IEEE Computer Society Press, 1998.
- [4] Parma, Gustavo G., et al. "Sliding mode algorithm for training multi-layer neural networks," *IEE Electronics Letters*, 38(1):97–98 (January 1998).
- [5] Riedmiller, Martin and Heinrich Braun. "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP algorithm." *Proc. of the IEEE Intl. Conf. on Neural Networks*. 586–591. April 1993.
- [6] Sarpturk, S. Z., et al. "On the stability of discrete time sliding mode control systems," *IEEE Transactions on Automatic Control*, 39(10):930–932 (1987).
- [7] Schiffmann, W., et al. *Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons*. Technical Report, Koblenz, Germany: Institute of Physics, University of Koblenz, 1992.
- [8] Utkin, V.I. *Sliding modes and their application in Variable Structure Systems*. Moscow: MIR, 1978.