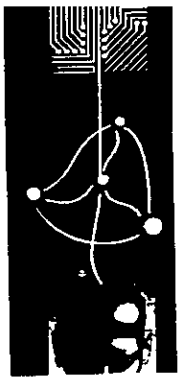

1° Congresso Brasileiro de Redes Neurais

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

CLASSIFICADORES NEURAIS USANDO BACKPROPAGATION COM ATENÇÃO SELETIVA

Marcelo C. Bossan[1], Luiz P. Calôba[2] e Jurandir Nadal[1]

[1]COPPE:Programa de Engenharia Biomédica e [2]COPPE/EE, Universidade Federal do Rio de Janeiro, Brazil

RESUMO

Uma das aplicações mais freqüentes das redes neurais é a classificação de padrões. Para tal, geralmente utiliza-se a estrutura *feedforward* com treinamento *backpropagation*, minimizando o erro médio quadrático (EMS) das saídas da rede. Este critério, razoável para a aproximação de funções, possui características que dificultam a obtenção da menor taxa possível de erros de classificação. A modificação no algoritmo *backpropagation* apresentada neste trabalho permite um melhor desempenho da rede neural em casos onde o espaço de padrões possui regiões pouco populosas juntamente com *clusters* com grande concentração de padrões, dirigindo a atenção do treinamento prioritariamente para os padrões ainda não aprendidos.

1.INTRODUÇÃO.

O algoritmo *backpropagation* geralmente é a primeira opção quando se deseja treinar uma rede neural *feedforward* multi-camadas. Sua característica de busca de descida segundo o gradiente, visando encontrar um ponto de mínimo na superfície de erro médio quadrático (EMQ) da saída, é muito desejável, na maioria dos casos, pois dá maior importância às entradas mais freqüentes e especial atenção às entradas mais deslocadas do centro da distribuição. Estas características tornam o algoritmo indicado para a aproximação de funções, onde um critério de mínimos quadrados é bastante razoável (Hecht-Nielsen, 1990).

Embora o problema da classificação de padrões possa ser entendido como o de encontrar uma função que associe padrões de entrada à uma classe, entre um número finito de classes (Lippman,1989), passos intermediários no sentido de reduzir o EMQ para o conjunto de padrões utilizado no treinamento da rede nem sempre implicam em redução instantânea do número de erros de classificação, podendo até mesmo causar um aumento no total de erros em algum instante durante o treinamento. Este algoritmo é reconhecidamente lento, sendo que muitos trabalhos vêm sendo publicados no sentido de reduzir o número de iterações

necessárias para a obtenção de um valor de EMQ satisfatório para o conjunto de treinamento e para o conjunto de avaliação (ver, p. ex., Hertz, Krogh e Palmer, 1990 e, Liguni et alii, 1992).

Uma alternativa possível para o treinamento de redes neurais *feedforward*, quando utilizadas para a classificação de padrões, é a modificação do algoritmo *backpropagation* com o objetivo de torna-lo mais sensível à quantidade de erros de classificação, durante os passos do treinamento da rede, ainda que em detrimento do EMQ para o conjunto de treinamento.

2.MODIFICAÇÃO NO ALGORITMO.

As redes neurais do tipo *feedforward* são as mais utilizadas atualmente para a classificação de padrões. Os padrões são apresentados à entrada desta rede que deve possuir uma saída correspondente à cada classe do espaço de padrões. As células utilizadas como exemplo terão funções de ativação sigmoidais. O objetivo do treinamento é fazer com que a rede tenha apenas uma de suas saídas ativada (igual a 1) para cada entrada aplicada e que esta saída corresponda à classe a que esta entrada pertença, ficando as demais saídas desativadas (iguais a 0). Como este comportamento ideal não é alcançável com um número finito de treinos, estabelece-se um limiar acima do qual uma saida é considerada ativada e abaixo do qual considerada desativada. Para células com saídas sigmoidais variando entre 0 e 1 utiliza-se normalmente o valor 0.5 como limiar. Quando estes valores são obtidos de forma correta para um determinado padrão, diz-se que a rede "aprendeu" a classificar este padrão.

O algoritmo normalmente aplicado para o treinamento da rede é o *backpropagation*, que utiliza um método de descida de gradiente para reduzir o erro médio quadrático para todo o conjunto de treinamento, expresso por

$$E[w] = \frac{1}{2}\sum_{u,i}\left[\zeta_i^u - O_i^u\right]^2$$

onde i e u representam respectivamente cada saída da rede e cada padrão aplicado à entrada da rede, ζ_i^u é a saída desejada (0 ou 1) e O_i^u a saída obtida .

O EMQ é fortemente dependente da freqüência de ocorrência de cada padrão. Qualquer tentativa de aproximar a O_i^u de um padrão muito freqüente de sua correspondente ζ_i^u, causará uma redução maior no EMQ, para todas as entradas, do que ocorreria com um padrão pouco freqüente. Como o algoritmo *backpropagation* sempre procura a redução mais rápida possível do EMQ, a partir da sua posição instantânea na curva, a rede tenderá a reduzir primeiro os erros devidos aos padrões mais freqüentes, até que estes erros sejam tão pequenos que os erros relacionados aos outros padrões se tornem significativos em relação aos erros dos padrões mais freqüentes. Só então os padrões menos freqüentes começarão a ser aprendidos.

Para um padrão já aprendido pela rede, qualquer tentativa de elevar a saída alta em direção ao seu valor ideal, 1, ou de reduzir as saídas baixas em direção à zero não contribuirá para a redução do número de erros da rede, mas este "reforço" será efetivo para a redução do EMQ da rede. Se este padrão for muito freqüente, em relação aos demais padrões do conjunto de treinamento, a rede se deterá na tentativa de realizar esta aproximação de valores, visando a redução do seu EMQ. Desta forma, a rede ficará por um longo número de seções de treinamento "presa" na tarefa de reduzir seu EMQ sem que isto cause qualquer redução no número de erros de classificação. Mais grave ainda, alguns padrões pouco freqüentes já aprendidos poderão até mesmo passar a ser classificados erroneamente durante esta aproximação, sendo necessário um grande número de seções de treinamento para que estes padrões possam voltar a ser classificados corretamente, o que pode resultar em um treinamento excessivamente lento.

Uma solução para o problema é tornar o algoritmo de treinamento da rede mais sensível às saídas correspondentes aos padrões erroneamente classificados e menos sensível aos erros das saídas correspondentes aos padrões já aprendidos. Definindo-se

$$e_i^u = \zeta_i^u - O_i^u$$

deseja-se que o algoritmo dê ênfase à redução dos erros onde $|e_i^u|>0.5$ e atenue os efeitos de valores de $|e_i^u|<0.5$ na busca do gradiente. Isto já é feito, de forma branda no algoritmo original, onde o erro "retropropagado" através das células da camada de saída é dado por

$$\delta_i^u = g'(h_i^u)\left[\zeta_i^u - O_i^u\right] = g'(h_i^u)\cdot f(e_i^u)$$

onde $g'(h_i^u)$ é a derivada da função de ativação da célula i da camada de saída quando o padrão u é aplicado à rede e $f(e_i^u)=e_i^u$. Neste caso, os valores de $f(e_i^u)=e_i^u$ para $|e_i^u|<0.5$, embora menores que para $|e_i^u|>0.5$, ainda permitem que a rede continue sendo treinada diminuindo o erro das entradas já aprendidas. Uma forma de se efetuar treinamento apenas quando ocorrer um erro de classificação é utilizar uma função $f(e_i^u)$ que seja não nula apenas para $|e_i^u|>0.5$, como a apresentada na Figura 1. O fato desta função não ser diferenciável impede seu uso em um algoritmo de gradiente, como o backpropagation. Para eliminar este problema, a solução desenvolvida foi a "suavização" da função

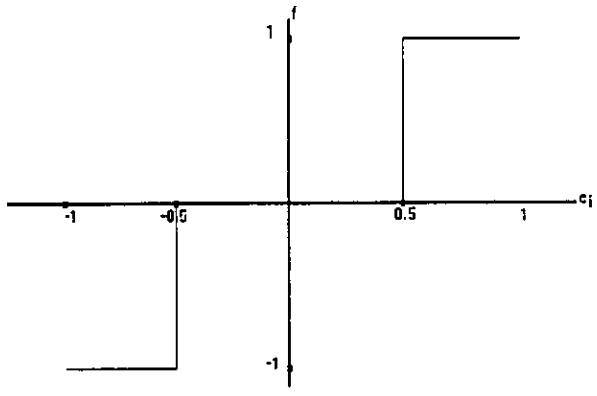

Figura 1 - $f(e_i)$ ideal para classificação de padrões.

através de uma aproximação sigmoidal, que pode ser obtida pela expressão

$$f(e_i^u) = \left(\frac{1}{1+e^{-\alpha(|e_i^u|-0.5)}} - a \right) \cdot b \cdot \text{sign}(e_i^u),$$

onde

$$a = \frac{1}{1+e^{\alpha/2}} \text{ , e } b = \frac{2+e^{\alpha/2}+e^{-\alpha/2}}{e^{\alpha/2}-e^{-\alpha/2}} .$$

Esta função tem como característica principal variar desde uma reta (como a do algoritmo original), para α próximo de 1, até a função idealizada da Figura 1 (semelhante à utilizada no treinamento do perceptron de Rosenblatt (1962)), para valores elevados de α. Para valores intermediários de α a função adquire um aspecto sigmoidal em cada um dos semi-planos. A Figura 2 apresenta o gráfico desta função para valores de α iguais a 1, 10 e 100.

Desta forma, aumentando-se α, $f(e_i^u)$ vai aos poucos reduzindo progressivamente seus valores para $|e_i^u|<0.5$ e aumentando seus valores para $|e_i^u|>0.5$.

A função custo F[w] associada ao algoritmo modificado tem mínimo em e_i^u igual a zero e é dada pela expressão

$$F[W] = \int f(x)dx$$
$$= \frac{2+e^{\alpha/2}+e^{-\alpha/2}}{e^{\alpha/2}-e^{-\alpha/2}} \cdot \sum_{iu} \left(\frac{1}{\alpha} \ln\left(1+e^{\alpha(|e_i^u|-0.5)}\right) - \frac{|e_i^u|}{1+e^{\alpha/2}} \right)$$

Com esta modificação o algoritmo torna-se capaz de reduzir a longa permanência do treinamento da rede na tentativa de

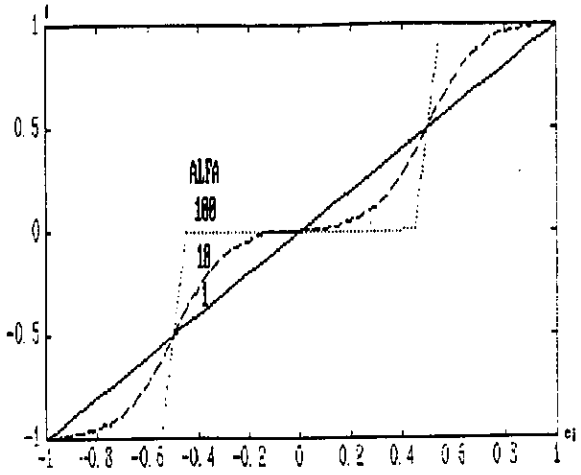

Figura 2 - Gráfico do $f(e_i)$ proposto para a modificação no algoritmo *backpropagation*.

aproximar de 0 ou 1 as saídas correspondentes aos padrões já aprendidos, liberando-a para o aprendizado de novos padrões. A redução no número de seções de treinamento aumenta com α, assim como também são aumentadas as regiões muito abruptas e muito planas da superfície de F[w] para o problema, elevando a possibilidade de paralisação da rede, como ocorre com os perceptrons de Rosenblatt (1962).

4.RESULTADOS

Dois diferentes conjuntos de padrões foram utilizados para a verificação do desempenho de redes neurais na classificação de padrões quando da introdução da alteração no algoritmo backpropagation proposta neste trabalho. No primeiro teste, foram considerados dois grupos de padrões de classes distintas, representados por cruzes e círculos cheios, cada um distribuído por uma região retangular composta por 1000 padrões. Acima da região à esquerda e

abaixo da região à esquerda, foi colocado 1 padrão de classe oposta à da região mais próxima (Figura 3). Treinando-se um único neurônio para a separação dos padrões deste espaço, utilizando-se o valor 0.1 para o passo de treinamento e 0.7 para o fator de momento, foram necessárias 200 apresentações de todo o conjunto de treinamento até que a rede fosse capaz de realizar corretamente o treinamento, obtendo resultado semelhante ao apresentado na Figura 3, onde a linha representa o separador gerado pelos pesos do neurônio. Utilizando-se a modificação proposta neste trabalho, com α igual a 100, foi possível a separação apresentada na Figura 3 com apenas 23 apresentações do conjunto de treinamento.

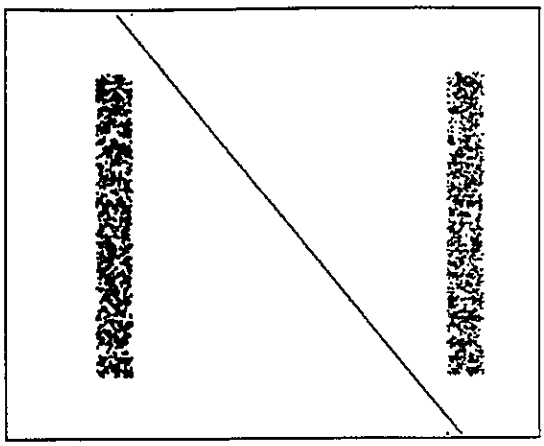

Figura 3 - Classificação do espaço de padrões do primeiro teste utilizando a modificação proposta neste trabalho.

Um segundo teste foi realizado utilizando uma rede de duas camadas (uma camada escondida), onde as vantagens da modificação proposta são ainda mais aparentes. Para tal, uma rede com 2 neurônios na primeira camada e 1 neurônio na segunda camada foi treinada com o algoritmo backpropagation para separar os padrões de um espaço formado por um padrão separados de outros 1000 de mesma classe (círculos cheios) por 1000 outros padrões de classe diferente (cruzes), conforme apresentado na Figura 4. Foram utilizados os valores 0.4 e 0.7 para, respectivamente, passo de treinamento e fator de momento. Mesmo após 10000 apresentações de todo o conjunto de treinamento, a rede não conseguiu classificar corretamente os padrões, separando apenas os dois grande blocos de dados. Utilizando-se a modificação proposta, com α igual a 100, foi possível a separação apresentada na Figura 4 com apenas 600 apresentações do conjunto de treinamento.

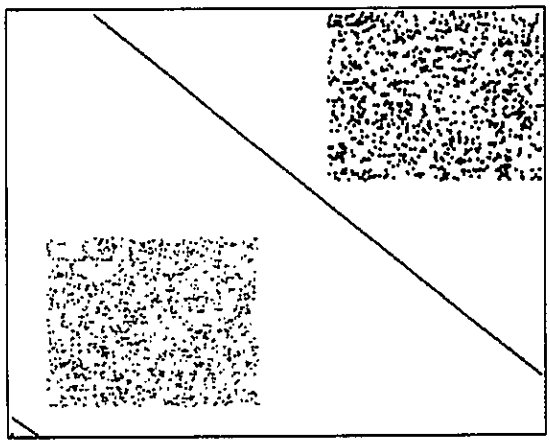

Figura 4 - Classificação do espaço de padrões do segundo teste utilizando a modificação proposta neste trabalho.

5.CONCLUSÕES.

Os testes apresentados mostram que a modificação proposta é capaz de resultar em aceleração do treinamento em casos onde existam padrões afastados de seus *clusters* principais, onde suas inclusões no grupo de padrões corretamente aprendidos pela rede não implicam em redução significativa de seu erro médio quadrático. Estes testes foram realizados em situações extremas, tanto no que concerne ao espaço de padrões quanto ao valor de α utilizado. Em casos reais, valores menores de α devem ser utilizados, podendo variar de 1, para um treinamento praticamente igual ao que se obtém ao utilizar o algoritmo backpropagation original, até valores tão elevados quanto 100, onde a rede comporta-se praticamente como um perceptron de Rosenblatt, com os seus conhecidos problemas de convergência.

O estudo deste possível novo algoritmo encontra-se ainda em fase inicial. Deve ainda ser definido um critério para o ajuste do valor de α em diferentes situações, ou de uma metodologia para a variação automática de α durante o treinamento. Uma alternativa aparentemente promissora é iniciar o treinamento com α igual a 1, permitindo um treinamento inicial praticamente idêntico ao do backpropagation original, elevando o valor de α quando for percebida uma redução significativa da velocidade de aprendizagem da rede. Porém, certamente o critério utilizado para a adaptação de α dependerá das características específicas do problema.

6.REFERÊNCIAS BIBLIOGRÁFICAS.

Hecht-Nielsen, R. (1990), *Neurocomputing*, Addison-Wesley Publishing Company, New York.

Hertz, J., Krogh, A. e Palmer, R.G. (1990), *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Company,New York.

Liguni, Y., Sakai, H., e Tokumaru,H. (1992), "A real Time Learning Algorithm for a Multilayered Neural Network Based on the Extended Kalman Filter", *IEEE Transactions on Signal Processing*, Volume 40, Páginas 959-966, Abril,1992.

Lippman, R.P. (1989), "Pattern Classification Using Neural Networks", *IEEE Comunications Magazine*, páginas 47-64, Novembro.1989.

Rosenblatt, F. (1962), *Principles of Neurodinamics*, Spartan, NewYork.