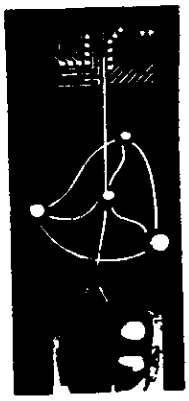


1° Congresso Brasileiro de Redes Neurais

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

UM ALGORÍTIMO DE TREINAMENTO CONTÍNUO PARA REDES MULTI CAMADAS

C.E. Pedreira e N.M.Roehl
Dept. de Eng. Elétrica PUC-Rio
C.P. 38063
22452-970 Rio de Janeiro
pedreira@ele.puc-rio.br

Resumo

Neste artigo se propõe um novo procedimento para treinamento contínuo de redes em camadas. Após a etapa tradicional de treinamento o algorítimo adaptativo ajusta, se e somente se necessário o conjunto de pesos, permitindo assim uma adaptação a um possível novo modelo. Trata-se portanto de metodologia especialmente interessante para sistemas variantes no tempo. Ao projetista é reservada em cada momento a decisão de que tolerância sobre o erro do último dado se está disposto a permitir. Tem-se um problema de compromisso entre casar o dado novo e perturbar o mínimo possível o conjuto de pesos gerado pelo treinamento anterior.

INTRODUÇÃO

Diversas aplicações de grande importância involvem sistemas variantes no tempo. A dificuldade intrínsica em lidar com esses tipos de sistemas induz muitas vezes à introdução de hipóteses artificiais que podem ser causadoras de resultados não satisfatórios. Redes neurais em camadas demonstraram-se uma ferramenta importante em uma variedade de problemas quando a hipótese de invariância do sistema é aceitável. Nossa contribuição consiste em propor um novo algoritmo de ajuste de pesos em redes multi camadas que se mostra conveniente à modelagem de sistemas variantes no tempo.
O principal objetivo é manter o erro relacionado ao dado novo dentro de uma tolerância pré estabelecida, minimizando a perda da informação incorporada pelo treinamento com os dados originais. Dessa forma, ao projetista é permitado julgar a relevância que se deseja atribuir à nova informação. Esforço neste sentido foi anteriormente feito por Park et al. [1], sem esta flexibilidade. Em [1], o último dado é obrigado a ter erro zero, em detrimento do treinamento anteriormente realizado. Esta falta de flexibilidade é especialmente pouco desejável em situações onde sabe-se de ante mão da possibilidade de mudanças bruscas, ou quando o sistema retorna ao modelo anterior com razoável rapidez. No primeiro caso, parece mais razoável informar a rede desta incerteza pré conhecida e controlar assim o erro que se pode permitir. No segundo, a metodologia proposta em [1] irá causar fortes danos ao conjuto de pesos original dificultando, deste modo, um retorno ao modelo original.

O PROBLEMA

Seja uma rede com uma camada oculta descrita da seguinte forma:

$$y = f_1[V^t u] \quad u = f_2[W^t x] \quad (1)$$

onde $y \in \Re^o$ é a saída da rede, $u \in \Re^h$ é o vetor de ativações da camada escondida e $x \in \Re^I$ é o vetor de dados de entrada. As

matrizes W e V contêm os pesos das ligações entre camadas de entrada e escondida e camadas escondida e de saída, respectivamente. As funções vetoriais $\mathbf{f_1} \in \Re^o$ e $\mathbf{f_2} \in \Re^h$ são tais que $f_1^i(.)$ e $f_2^i(.)$ são funções diferenciáveis não decrescentes. Para simplificar o desenvolvimento, vamos supor as funções sigmoidais. Essa formulação pode ser facilmente generalizada para redes com mais de uma camada embituda.

A partir de uma rede treinada, i.e., um conjunto de pesos (W,V) apropriado para os pares entrada-saída originais $((x_i,d_i)$, i = 1,...,n) e um dado novo (x_{n+1},d_{n+1}), o objetivo é determinar um novo conjunto de pesos (W',V') tal que a seguinte função energia é minimizada, sujeita à nova restrição imposta pelo último dado apresentado.

$$\text{Min } E = 1/2 \sum_{i=1}^{N+1} |d_i - y_i|^2 \qquad (2)$$

t.q.

$$-\varepsilon^l \le (d^l_{n+1} - y^l_{n+1}) \le \varepsilon^l, \; l = 1,...,o \qquad (3)$$

onde $\varepsilon^l \in (0, \min(d^l_{n+1}, 1- d^l_{n+1}), l = 1,...,o$ é a tolerância associada à nova restrição e $|.|$ é a norma Euclidiana. Pode-se provar que estes limites impostos sobre ε garantem a existência da função inversa $(f_1)^{-1}$.

A função objetivo (2) reflete o desejo de minimizar o erro relativo aos dados de treinamento originais enquanto a restrição (3) manterá o erro associado ao último dado dentro de uma tolerância pré estabelecida ε. Isso significa, que se está lidando com uma solução de compromisso entre a informação do sistema antigo e o dado novo, possivelmente refletindo uma variação na planta original. Essa abordagem permite ao projetista controlar a tolerância, ou seja, decidir o quão relevante o dado novo é comparado aos dados anteriormente utilizados na identificação do sistema.

De modo a simplificar o tratamento analítico e computacional do problema $P\beta$ é preciso linearizar a restrição (3). Desse modo, suponha-se que o novo conjunto (W',V') pode ser escrito como: V' = V+ΔV e W' = W+ΔW, onde os incrementos $\Delta V \in \Re^{hxo}$ e $\Delta W \in \Re^{Ixh}$. De modo a simplificar a notação, os índices de super escritos serão omitidos. Então, para (x_{n+1},d_{n+1}) das equações (1) e (3) temos

$$-\varepsilon \le d_{n+1} - f_1[(V+\Delta V)^t u] \le \varepsilon$$

logo,

$$d_{n+1} - \varepsilon \le f_1[(V+\Delta V)^t u] \le d_{n+1} + \varepsilon.$$

Como $(f_1^i)^{-1}$ é uma função monotonamente crescente $\forall$ i = 1,...,o:

$$(d_{n+1} - \varepsilon) \le (V+\Delta V)^t u \le f_1^{-1}(d_{n+1} + \varepsilon) \quad (4)$$

Por outro lado, aplicando-se a expansão de Taylor de primeira ordem obtem-se:

$$u \approx f_2(W^t x_{n+1}) + J f_2(W^t x_{n+1}) \Delta W^t x_{n+1}. \qquad (5)$$

E, substituindo-se a aproximação (5) na desigualdade (4), chega-se a:

$$f_1^{-1}(d_{n+1} - \varepsilon) + V^t f_2(W^t x_{n+1}) \le$$

$$\le V^t J f_2(W^t x_{n+1}) \Delta W^t x_{n+1} +$$

$$+ \Delta V^t f_2(W^t x_{n+1})$$

$$\le f_1^{-1}(d_{n+1} + \varepsilon) - V^t f_2(W^t x_{n+1}) \qquad (6)$$

Observe-se que essa aproximação é válida se e somente se H << 1, onde H é o Hessiano de $f_2(W^t x_{n+1})$, e $\Delta V_{ij} << V_{ij}$, $\forall$ i = 1...h e j = 1...o. Ao limitar as pertubações sobre W e V, definine-se uma região onde a hipótese de linearidade é válida. Rearrumando-se ΔV^t e Δ W em forma vetorial, através do operador vec, a restrição (6) pode ser escrita numa forma compacta como:

$$c_1^i(\varepsilon) \le (Az)^i \le c_2^i(\varepsilon) \quad i = 1...o \qquad (7)$$

onde $A = [A_p : A_q]$, $A \in \Re^{o(p+q)}$

$p = Ixh$ e $q = hxo$

$z = [\Delta W_{vec}{}^t : (\Delta V^t)_{vec}{}^t]^t$, $z \in \Re^{(p+q)}$

$A_p \in \Re^{oxp}$ é a solução do sistema de equações:

$$A_p \Delta W_{vec} = V^t J f_2(W^t x_{n+1}) \Delta W^t x_{n+1}$$

$A_q = [A_i] \in \Re^{oxq}$ onde

$A_i = [0...0\ f_2(W^t x_{n+1})^t\ 0...0]$, $i = 1...o$

$$c_1(\varepsilon) \equiv f_1^{-1}(d_{n+1} - \varepsilon) + V^t f_2(W^t x_{n+1}),$$

$c_1(\varepsilon) \in \Re^o$

$$c_2(\varepsilon) \equiv f_1^{-1}(d_{n+1} + \varepsilon) - V^t f_2(W^t x_{n+1}),$$

$c_2(\varepsilon) \in \Re^o$.

A região factível pode então ser definida como a intersecção entre a região de linearidade e o hiperespaço definido pela restrição (7).

A função energia pode também ser reescrita numa forma compacta como

$$J(z) = 1/2\ z^t K z$$

onde $K = S^tS$ e S é a matriz de sensibilidade de y às pequenas variações nos pesos, obtida após algum algebrismo. A variação de energia associada aos N+1 dados devido a variações nos pesos pode ser escrita como

$$\Delta E = [\Delta E_1 \ldots \Delta E_{N+1}]^t = Sz.$$

Após rearrumar (2) em termos de variação de energia, chega-se a:

$$J = 1/2 \sum_{i=1}^{N+1} (E_{i,W} - E_{i,W'})^2$$

onde $E_{i,W}$ e $E_{i,W'}$ são os erros de saída relacionados ao i-ésimo dado com osconjuntos de pesos (W,V) e (W',V'), respectivamente. Logo, tem-se que

$$J = 1/2 \sum_{i=1}^{N+1} \Delta E_i^2 = 1/2\ \Delta E^t \Delta E = 1/2\ z^t K z$$

O problema P_ε pode agora ser formulado da seguinte forma:

(P_β) Min $J(z) = 1/2\ z^t K z$

s.a $c_1(\varepsilon) \le Az \le c_2(\varepsilon)$

$z \in \beta$

onde $\beta = \{z \in \Re^{p+q}: |z_i| \le \delta_i,\ i = 1,...,p+q\}$, $\delta_i > 0$ é a região de linearidade da restrição (3). Utilizando-se propriedades da função inversa de f_1, a desigualdade de Shwartz e um valor arbitrariamente pequeno para o erro da aproximação linear de Taylor, determina-se valores bem definidos para δ_i, para todo i = 1,...,p+q.

Observa-se que simplificando-se a rede para uma única unidade de saída e ajustando-se $\varepsilon = 0$, a restrição (7) se tornará idêntica à restrição imposta em [1]. Dessa forma, P_β é de fato, uma generalização do problema proposto em [1], onde uma restrição de igualdade $d_{n+1} = f_1[(V+\Delta V)^t u]$ é utilizada para obrigar que a rede se ajuste perfeitamente ao último dado apresentado. Essa restrição é um hiperplano de simetria para o hiperespaço definido por (7), cuja largura é função do parâmetro ε (figura 1).

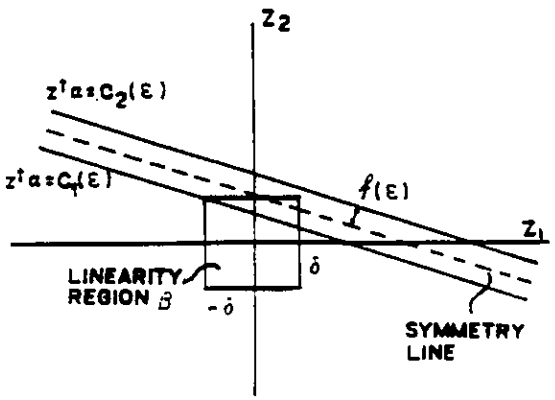


Figura 1

Outra característica introduzida nesse algoritmo é o tratamento de outliers nos dados. Foi implementada uma heurística simples que se baseia no erro de saída da rede e evita que os pesos se alterem caso haja uma alta probabilidade do dado novo se caracterizar com um outlier. Se, no entanto, os dados seguintes apresentarem o mesmo tipo de comportamento, interpreta-se que provavelmente estará ocorrendo uma mudança estrutural no sistema, devendo os pesos serem então modificados para captar essa nova tendência. Esse tipo de tratamento parece encontrar aplicação especialmente em problemas de previsão onde a ocorrência de outliers é frequente.

ANÁLISE DO PROBLEMA

Trata-se a seguir da análise teórica da questão de existência de solução para o problema P_β. Formulam-se duas proposições que estão relacionadas à essa análise, para o caso de uma única unidade de saída. Resultados análogos podem ser obtidos para o caso de redes com saídas múltiplas após algum algebrismo.

A proposição (P1) refere-se ao caso em que o hiperplano de simetria da região definida pela restrição (7) intercepta a região de linearidade β (figura 1). A proposição (P2) fornece às condições sobre o parâmetro ε que garantem a existência de solução quando esse hiperplano não intercepta β.

Proposição (P1)

Seja $r \equiv \{z: < z, a > = c\}$ onde $c = f_1^{-1}(d_{N+1}) - v^t f_2(W^t x_{N+1})$, o hiperplano de simetria da região definida pela restrição (7). Suponha-se que r intercepta a região de linearidade β, então o problema P_β tem solução para todo $\varepsilon \in (0, \min(d_{N+1}, (1-d_{N+1}))$.

Prova: cf. [2] e [3]

Proposição (P2)

Seja $\varepsilon_{max} = \min(d_{N+1}, (1-d_{N+1}))$. Assume-se que o hiperplano de simetria não intercepta a região de linearidade. Então, se $\varepsilon^* \leq \varepsilon_{max}$, o problema P_β tem solução $\forall \varepsilon \in [\varepsilon^*, \varepsilon_{max})$.

Prova: cf. [2] e [3]

Quando o hiperplano de simetria não intercepta a região de linearidade, o projetista deverá aumentar o valor do parâmetro ε de modo a garantir a existência de uma solução apropriada. Vale notar que se $\varepsilon > \varepsilon_{max}$ o problema não terá solução. Neste caso, ao usar o algorítimo proposto em [1] se estará violando fortemente a hipótese de linearidade, provocando assim um erro não previsível e não controlado.

A SOLUÇÃO DO PROBLEMA

O Problema P_β pode ser resolvido por uma variedade de algorítimos padrão de programação não linear. Embora, no caso geral, não se possa garantir otimização global para tais esquemas, no presente caso, tem-se convergência global uma vez que a função de custo J é uma função convexa sobre uma região viável também convexa [4]. Descreve-se a seguir o algorítimo proposto para solução do problema P_β. Um algorítimo do tipo projeção de gradiente [4] é usado para resolver o problema de programação quadrática que aparece no passo 4.

Algorítimo para solução de P_β:

Passo 0 - Verifique através de uma heurística se o dado novo tem alta probabilidade de ser um "outlier".

Passo 1 - Calcule a matriz de sensibilidade S (para maiores detalhes ver [2]), calcule $K \equiv S^t S$.

Passo 2 - Ache as restrições lineares

$$Az = c1 \quad e \quad Az = c2$$

Passo 3 - Ache o limitante β usando:

$$-\delta_i \leq z_i \leq \delta_i, \; i = 1...(p+q)$$

Passo 4 - Através do método de gradiente projetado ache z que minimiza a função de custo $J(z) = 1/2\ z^t\ K\ z$ sujeito às restrições $c_1(\varepsilon) \leq Az \leq c_2(\varepsilon)$ e $z \in \beta$.

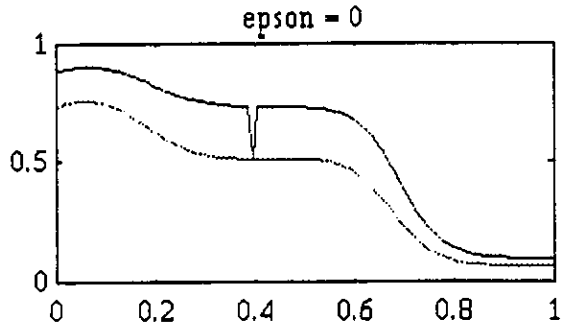
Passo 5 - Ache ΔW e ΔV a partir de z.

Passo 6 - Atualize W e V:

$$W' = W + \Delta W \text{ e } V' = V + \Delta V.$$

Uma característica importante do algorítimo proposto está relacionada com a não alteração dos pesos quando o dado novo não representa uma mudança do modelo. Deste modo, pode-se deixar o sistema adaptativo permanentemente ligado, com a garantia de que só serão feitas modificações em caso de necessidade. A prova desta propriedade se encontra em [2].

Como ilustração gráfica do algoritmo de treinamento contínuo proposto, apresentamos na figura 2 resultados relacionados a um exemplo simples de uma rede com uma unidade de entrada, uma unidade de saída e uma camada embutida. A rede foi inicialmente treinada através do algoritmo de retropropagação do erro para captar o mapeamento existente entre 100 pontos da curva. Provoca-se um desvio de 40% num dos pontos e aplica-se o algoritmo de treinamento contínuo para valores diferentes do parâmetro ε. Observa-se que a medida que o valor de ε aumenta, menor é o desvio da resposta da rede para os outros pontos da curva. Entretanto, pior é o ajuste para o dado novo.



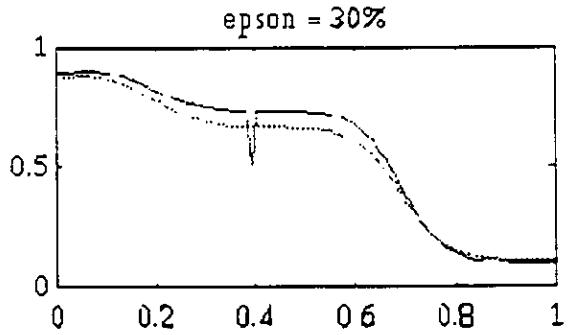


Figura 2

COMENTÁRIOS FINAIS

O esquema de treinamento contínuo aqui apresentado, permite que o projetista decida o quanto está disposto a pagar pela adaptação do dado novo. Esta metodologia parece ser bastante interessante especialmente quando se tem informação a respeito da velocidade de alteração do modelo, e também do nível de tolerância aceita pela aplicação em questão. Embora se tenha o campo das aplicações como meta, este artigo representa uma contribuição de cunho teórico. Pesquisa está no momento em andamento visando testar o desempenho numérico do algorítimo proposto, especialmente para previsão e classificação.

REFERÊNCIAS:

[1] Park D.C., El-Sharkawi M.A., e Marks II R.J., "An Adaptively Trained Neural Network", IEEE Trans. on Neural Networks, Vol 2, N. 3, Maio 1991.

[2] Pedreira C.E. e Roehl N.M., "On Adaptively Trained Neural Networks", Comunicação Interna CSC 36-93 PUC-Rio Dept. de Eng. Elétrica, Março de 1993.

[3] Pedreira C.E. e Roehl N.M., "On Adaptively Trained Neural Networks", Proceedings do 1993 Intern. Joint Conference on Neural Networks, Vol 1, pp 565-568, Nagoya, Japão, Outubro 1993.

[4] Gill P.E., Murray W. e Wright M.H., Practical Optimization, Academic Press 1981.