Proceedings of the V Brazilian Conference on Neural Networks - V Congresso Brasileiro de Redes Neurais
pp. 151–153, April 2–5, 2001 - Rio de Janeiro - RJ - Brazil

CUIDADOS NO USO DO ALGORITMO DE PODA EM REDES NEURAIS

Dilson G. Espenchitt [1], André G. G. Cunha [2], Gerson Lachtermacher [3]
[1] Centro de Análise de Sistemas Navais-CASNAV
[2] DEE/PUC-RIO
[3] FACC/UFRJ e Falculdades IBMEC-RJ
E-mails: espen@uol.com.br, andreggc@ele.puc-rio.br, glachter@ibmecrj.br

Abstract

In this paper is researched the relationship between weight elimination procedure, the initial application cycle, the final topology and generalization of a neural network prediction model. Keeping constant the initial parameters we developed several tests using different setups. We observed that the generalization of the different models were very different, sometimes assuming a different set of independent variables.

1. Introdução

Na procura da topologia ótima de uma rede neural que minimize os problemas de subparametrização / sobreparametrização surgiram os métodos de poda. Alguns destes iniciam o treinamento com uma topologia muito maior do que a necessária e no decorrer deste vão podando (removendo) partes da rede que não sejam necessárias, facilitando assim a resolução dos problemas de dimensionamento da rede e escolha de variáveis relevantes. Como inicialmente a rede é grande, esta possui graus de liberdade suficientes para acomodar rapidamente as características gerais dos dados de entrada, de uma forma pouco sensível às condições iniciais e aos mínimos locais. Após a acomodação inicial, a rede pode ser podada de forma a realizar um "trade-off" entre a complexidade da rede e seu poder de generalização, eliminando as características não relevantes do conjunto de treinamento favorecendo, desta forma, sua generalização.

2. Algoritmos de Poda

A idéia básica dos algoritmos de poda para simplificar a rede é cortar as conexões entre os nós. Pode-se simplesmente zerar o peso de uma conexão e verificar qual a influência disso na função objetivo. Se o erro aumentar muito então restaura-se o peso cortado, caso contrário, remove-se definitivamente o peso.

Naturalmente que esses algoritmos são demasiadamente lentos, principalmente para redes de maior porte. Assim, os algoritmos de poda que veremos a seguir adotam outras abordagens. [1], [2].

2.1. Métodos de Cálculo de Sensibilidade

Alguns métodos procuram estimar a sensibilidade da função de erro frente à remoção de uma ligação e então remover as ligações com menor influência. Em geral esses métodos atuam após o treinamento da rede. Ou seja, a rede é treinada para um tamanho maior que o necessário e posteriormente os pesos de menor sensibilidade são retirados.

Esses métodos não levam em consideração correlações entre os pesos da rede. A sensibilidade de cada peso wij é calculada como se ele fosse o único candidato a ser retirado. Quando um dos pesos é retirado, as demais sensibilidades calculadas não continuam necessariamente válidas para a nova configuração da rede.

2.2. Métodos com Termos de Penalização

A idéia central do algoritmo de "backpropagation" é minimizar uma função de custo, geralmente o erro quadrático, na saída da rede. Como a hipótese dos algoritmos de poda é que a menor rede estimada seja capaz de responder aos padrões de treinamento então a mesma apresentará a melhor generalização, deste fato surgiu a idéia de se acrescentar à função de custo tradicional um termo representando o custo devido à complexidade da rede.

Esse termo adicional é chamado de termo de penalização, na medida que representa uma penalização que a função de custo sofre devido à complexidade da rede. Os métodos que introduzem esse termo, procuram minimizar uma função de custo que inclui um termo de custo de complexidade. A lógica é que haja uma redução na complexidade da rede, forçada pela tendência que os pesos sejam levados para zero para atender à minimização do custo de complexidade.

Alguns métodos propõem a inclusão de termos na função de custo que levam em conta o somatório do quadrado dos pesos das ligações (Σw_{ij}^2), o somatório do módulo dos pesos ($\Sigma |w_{ij}|$) ou uma função logarítmica

$\log(1+w^2)$. Weigend [4] propôs um termo de penalização da forma :

$$\lambda \sum_{i \in C} \frac{w_i^2 / w_0^2}{1 + w_i^2 / w_0^2}$$

onde *C* é o conjunto de conexões e w_0 é uma constante que objetiva ponderar o valor dos pesos (fator de escala). Para $|w_i| >> w_0$ o custo de complexidade se aproxima de λ. Para $|w_i| << w_0$ o custo de complexidade se aproxima de zero. λ é um parâmetro que dá maior ou menor relevância ao custo de complexidade de acordo com uma heurística escolhida durante o treinamento [3].

2.3. Eliminação Dinâmica de Sinapses por Limiar

O algoritmo original de Weigend [4], força os pesos a valores insignificantes (próximos de zero) através da penalização da função de custo, como visto na equação acima. Para determinar a redução da topologia final da rede, é necessário fazer uma inspeção ao final do treinamento, retirar os nós que não tenham mais ligações e retreinar a rede para a nova configuração, de forma que as ligações restantes possam absorver os pesos das ligações que não eram exatamente zero e que se deseja eliminar. Para tal, Weigend [4], sugere iniciar o treinamento com $\lambda = 0$ (zero) e incrementando-o paulatinamente seguindo algumas heurísticas.

A abordagem utilizada em [1] e [2] foi ligeiramente diferente, onde o algoritmo estabelece um limiar para cada ligação sináptica. Quando o módulo do valor da sinapse fica abaixo desse limiar, a ligação é eliminada (tornada zero) durante o processo de treinamento. A tendência da rede é que, dinamicamente, as outras sinapses incorporem o resíduo do peso que deixou de fazer parte da rede (com a eliminação da sinapse que atingiu um valor em módulo abaixo do limiar) eliminando-se assim o último passo do procedimento sugerido por Weigend [4].

Entretanto, devido à aleatoriedade na escolha dos pesos iniciais, essa técnica [1] e [2] só pode ser empregada se garantirmos que a eliminação da sinapse, por estar abaixo do limiar, só será empregada após um período de estabilização inicial do valor dos pesos. Se aplicarmos esse critério de eliminação desde o início do treinamento, vista que os pesos iniciais são aleatoriamente escolhidos em um intervalo ao redor de zero, pode-se eliminar indevidamente um peso que esteja mudando de sinal e portanto passando por zero. Para que isto não ocorra, os treinamentos utilizando o algoritmo de "weigth-elimination" deverão ser feitos somente após uma quantidade inicial de ciclos de treinamento com "backpropagation" puro. Eliminando, desta forma, tal problema, uma vez que a acomodação inicial dos valores dos pesos é feita rapidamente ainda dentro dos ciclos de execução do "backpropagation" puro.

3. Resultados

Em [1], através da aplicação da metodologia do weight elimination modificada foi constatado que inicializando a aplicação desta a partir de ciclos diferentes (1.000, 2.000, 3.000, 4.000, 5.000, 10.000, 20.000), rodando 50.000 ciclos e utilizando os mesmos parâmetros de inicialização , o erro absoluto percentual médio (MAPE) variava de acordo com o ciclo inicial de aplicação do weight-elimination, conforme mostra o gráfico 1 abaixo:

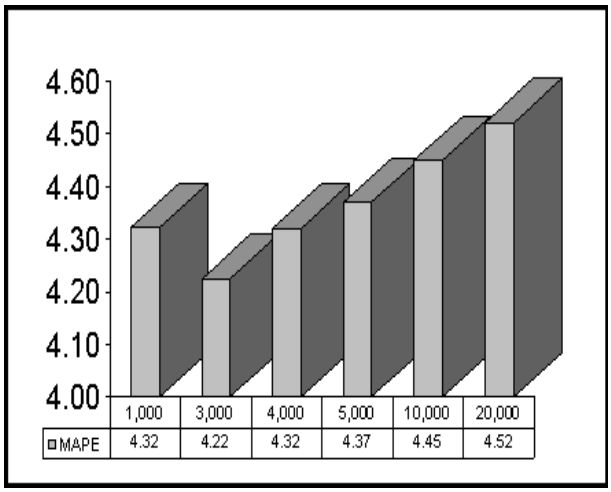

Gráfico 1

No estudo realizado em [5], o mesmo algoritmo modificado foi aplicado em diferentes ciclos iniciais (5.000, 10.000, 15.000, 20.000, 25.000, 30.000), utilizando os mesmos parâmetros de inicialização e rodando 85.000 ciclos, chegou-se a diferentes índices de acerto na previsão, mostrados no quadro 1 abaixo:

Rede	Ciclo	Rede	Não Falidas			Falidas		
Inicial	Inicial	Final	Corretas	Incorretas	Total	Corretas	Incorretas	Total
10x8x1	0	10x6x1	18	2	20	8	4	12
10x8x1	5.000	10x5x1	19	1	20	5	7	12
10x8x1	10.000	10x5x1	19	1	20	11	1	12
10x8x1	15.000	10x6x1	18	2	20	9	3	12
10x8x1	20.000	10x6x1	17	3	20	9	3	12
10x8x1	25.000	10x5x1	19	1	20	10	2	12
10x8x1	30.000	10x5x1	18	2	20	10	2	12

Quadro 1

Em outro estudo realizado de previsão de risco de hipertensão arterial [6], também aplicamos o mesmo algoritmo modificado onde, iniciamos o corte em diferentes ciclos (1.000, 5.000, 10.000, 15.000, 20.000, 25.000 ,45.000), inicializando a partir dos mesmos parâmetros (pesos e bias) e rodando 200.000 ciclos , observamos que a topologia final da rede é diferenciada

e algumas entradas foram eliminadas, conforme podemos observar no quadro 2 abaixo:

Rede Inicial	Ciclo Inicial	Rede Final	Entradas Eliminadas
14x14x1	1.000	8x1x1	1, 2, 4, 5, 7, 9
14x14x1	5.000	7x1x1	1, 2, 4, 5, 7, 9, 14
14x14x1	10.000	9x2x1	1, 4, 7, 9, 14
14x14x1	15.000	13x4x1	1
14x14x1	20.000	14x10x1	nenhuma
14x14x1	25.000	14x9x1	nenhuma
14x14x1	45.000	14x11x1	nenhuma

Quadro 2

4. Conclusão

Analisando os exemplos acima observamos que, além dos cuidados com a escolha da taxa de aprendizado, taxa de momento, topologia da rede, inicialização de pesos e bias, citados em [3], [7],[8], quando utilizamos o algoritmo weight-elimination temos um parâmetro a mais para especificarmos : o ciclo inicial de aplicação do weight-elimination o qual, conforme demonstramos, irá influenciar nos resultados de generalização da rede.

Tal fato, que é desconsiderado no algoritmo original de weight elimination sugerido por Weigend [4], tem relevância na performance do modelo e deve ser observado quando do treinamento de uma rede utilizando este algoritmo.

Referências

[1] Carvalho, J. G. Jr. , Algoritmo de Poda em Redes Neurais- Um Estudo Aplicado à Concentração de Ozônio , Dissertação de Mestrado, Departamento de Engenharia de Produção, Universidade Federal Fluminense, Niterói – RJ,1999.

[2] Oliveira, R. B., Redes Neurais Artificiais para Previsão de Séries Temporais, Dissertação de Mestrado, Departamento de Engenharia de Produção, Universidade Federal Fluminense, Niterói – RJ,1999.

[3] Haykin, S., Neural Networks – A Comprehensive Foundation. : Macmilan College Publishing Company, 2nd ed, New Jersey, 1999.

[4] Weigend, A. S., Rumelhart, D. E., Huberman, B.A., Back-propagation , weight- elimination and time series prediction, in Connectionist Models-proceedings of the 1990, Summer School, Edited by D. S. Touretzky, J.L. Elman, T. J. Sejnowski, G. E. Hinton. Morgan Kaufmann Publishers, Inc., 1991.

[5] Espenchitt, D. G., & Lachtermacher, G., Previsão de Falência de empresas, utilizando Redes Neurais com Algoritmo de Poda, Revista ANGRAD, vol. 1, nº1, pp 73-81, São Paulo, SP, jul/set 2000.

[6] Cunha, A.G.G. & Lachtermacher, G., Redes Neurais Aplicadas à Previsão de Riscos de Hipertensão Arterial, Relatório Técnico, Faculdades Ibmec-RJ, 2000.

[7] Hertz, J , Krogh A , Palmer R G., Introduction to Theory of Neural Computation, Perseus Publishing L.L.C., USA, 1991.

[8] Trippi, R. R. , Turbam, E., Neural Networks in Finance and Investing : using artificial intelligence to improve real-world performance, Mc Graw Hill, New York,1996.