

1º Congresso Brasileiro de Redes Neurais

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

Uso Cooperativo de Goal Seeking Neurons e Adaptive Logic Networks

W Martins and N M Allinson
Image Engineering Group, Electronics
University of York
York, United Kingdom

Resumo

A implementação física de Redes Neurais ainda é um dos grandes obstáculos para a sua utilização prática. Resultados significativos oriundos da cooperação entre dois modelos de fácil e econômica implementação reforçam a viabilidade de modelos que independem de variação de sinapses. A idéia básica deste artigo é o uso do treinamento iterativo de ALN em redes derivadas do treinamento (não-iterativo) de GSN.

1 Introdução

A popularidade de modelos cuja aprendizagem é desvinculada de variação de sinapses analógicas vem crescendo sensivelmente. O baixo custo e a facilidade da implementação física são fatores fundamentais neste fato. A deficiência de tais modelos está localizada freqüentemente no desempenho final. Alguns melhoramentos têm sido propostos para modelos específicos [1, 2, 3] com algumas exceções [4]. Este artigo apresenta resultados da combinação entre redes feed-forward de Goal Seeking Neurons [5] e Adaptive Logic Networks [6] e o algoritmo de conversão de topologias requerido.

A combinação entre os dois modelos envolvidos é feita através do uso de GSN como gerador de topologias a serem re-treinadas pelo algoritmo iterativo de ALN. A aprendizagem em redes feed-forward de GSN é extremamente rápida, porque utiliza cada exemplo somente uma única vez. Em outras palavras, o treinamento é não iterativo. Além disso, existe um claro privilégio a exemplos utilizados no início do treinamento porque não se admite que uma nova informação destrua outra já registrada. Quando um exemplo gera uma modificação interna em um neurônio, esta modificação é irrevogável. Os autores já demonstraram que o controle dinâmico da ordem de apresentação de exemplos é uma alternativa prática para resolver problemas onde várias redes são empregadas [7]. Neste artigo, resultados ainda melhores foram obtidos através do uso cooperativo de GSN e ALN.

Além do efeito positivo final na performance de classificadores empregados no reconhecimento de dígitos manuscritos, o produto final, redes de ALN, possui implementação (tanto em software quanto em hardware) mais fácil e econômica que a requerida por redes feed-forward de GSN. Por outro lado, o uso de redes de GSN resolve problemas de condições iniciais sofridos por ALN, pois possui maior funcionalidade a nível neuronal.

2 Breve Descrição Crítica dos Modelos

A nível neuronal, GSN são capazes de implementar qualquer função lógica, pois armazenam explicitamente a saída para cada possível situação de entrada. Neurônios de ALN, todavia, estão restritos às funções crescentes de duas variáveis: AND, OR.

LEFT e RIGHT (onde $RIGHT(x,y) = y$ e $LEFT(x,y) = x$). Ao contrário dos neurônios de ALN, GSN possuem um terceiro estado 'u' para assinalar posições internas ainda indefinidas e podem ter mais que duas entradas. O aumento do número de entradas é, contudo, acompanhado da necessidade de mais posições internas, obedecendo lei exponencial (2^n , onde n é o número de entradas¹). Apesar da melhoria no grau de saturação que o uso de mais posições internas resulta [8], este recurso é inconveniente sob ponto de vista econômico. Por outro lado, o estado final de neurônios de ALN ou são funções booleanas básicas, AND e OR, ou implementam simples cortes na topologia, RIGHT e LEFT. Ambos modelos utilizam uma única saída por neurônio, levando o uso de pirâmides como topologia natural (ALN usam árvores binárias mais especificamente). Novas abordagens foram recentemente publicadas para ultrapassar este obstáculo no caso de redes de GSN [2.3].

As limitações impostas na funcionalidade do neurônio em ALN são justificadas pelo aumento da insensibilidade das pirâmides a mudanças no nível de entrada [9], isto é, a saída final da rede não muda facilmente com pequenas variações na entrada. Como consequência, a implementação em hardware requer um simples circuito combinacional Booleano e a simulação por software pode ser acelerada sensivelmente através de *lazy evaluation*, já que tanto AND quanto OR podem ter a saída totalmente determinada pelo particular estado de uma única entrada.

Para aumentar a capacidade de mapeamento de ALN, entradas podem ser tomadas na sua forma normal ou complementada (previamente invertida). Mesmo assim, a dificuldade de implementar mapea-

mentos não-monotônicos pode ser demonstrada pelo valor 0,26% para a probabilidade de geração de topologias capazes de aprender a função XOR(x,y), assumindo eventos equiprováveis e quatro entradas (como aconselha o criador do modelo). A bem da verdade, ALN foram projetadas para implementar apenas funções monotônicas e esta característica é, às vezes, conveniente (como no controle de próteses para deficientes físicos [10], controle ativo de suspensão de automóveis [11]), pois possibilita a imposição de limites de segurança nas respostas das redes quando problemas com entradas e/ou saídas analógicas são enfrentados.

Assim como '0' e '1', o valor indefinido 'u' dos GSN pode ser armazenado, recebido e transmitido. Quando recebido, o valor indefinido acessa várias posições internas pois é entendido como '0' e '1' simultaneamente. Além de aproveitar melhor as posições internas já modificadas e, conseqüentemente, atrasar a ocorrência de saturação, a inclusão deste novo valor (com a mesma liberdade dos valores Booleanos originais) e a adoção de procedimentos determinísticos para manipulá-lo livra o sistema do caráter estocástico de seu predecessor. Os pontos negativos desta estratégia, além do maior número de bits requerido no estágio final, são o tempo gasto/complexidade final do chip para lidar com este valor indefinido no caso do uso de software/hardware e a necessidade de enviar dois bits pelos canais que interligam GSN no caso de implementação em hardware. O uso de redes de transputers é, portanto, adequado à implementação de redes de GSN.

Apesar de concordarem na escolha aleatória das topologias iniciais, as formas de treinamento adotadas por cada modelo são opostas: redes feed-forward de GSN são treinadas rapidamente de forma não-iterativa enquanto ALN procuram iterativamente um ponto de equilíbrio na aprendizagem de todos os exemplos. Enquanto redes de GSN mantêm a ordem de apresentação dos exemplos mesmo quando várias redes são utilizadas (como em classificadores), ALN modificam-na aleatoriamente antes de cada ciclo de

¹Na verdade, a implementação do terceiro estado de GSN faz com que cada posição interna tenha pelo menos dois bits. O requerimento em bits para um neurônio de n entradas é, portanto, 2^{n+1} .

aprendizagem para evitar efeitos temporais (que prejudicam redes feed-forward de GSN).

GSN apresentam três estados: validação, aprendizagem e avaliação. Um exemplo é aprendido se, e somente se, passa com sucesso pela fase de validação. Durante validação, neurônios são estimulados a responderem 'u'. Isto acontece quando não apenas um único valor Booleano é acessado. Se o neurônio da última camada responder 'u' ou o valor que se deseja ensinar, o exemplo foi validado com sucesso, isto é, um mapeamento entre o nível de entrada e o nível de saída pode ser implementado para cumprir o objetivo desejado. Na fase de aprendizagem, o neurônio irá modificar posições internas (se necessário). Durante o treinamento, a validação é feita através da propagação feed-forward de estímulos. Em caso positivo, os neurônios de cada nível entram em estado de aprendizagem logo após o mesmo processo ter terminado para o nível seguinte, já que precisam saber qual é o específico mapeamento requerido. A dinâmica é similar, portanto, ao algoritmo de back-propagation: uma fase feed-forward de propagação de estímulos seguida de outra fase em sentido contrário para ajustes na topologia. Durante avaliação, o procedimento típico é o uso do valor Booleano mais frequente dentre as posições acessadas.

O algoritmo de aprendizagem de ALN é, a nível neuronal, baseado na frequência de acesso, no sentido de respeitar a maioria das solicitações de mapeamento e destina-se a determinar a saída quando as entradas têm valores Booleanos diferentes. Um contador interno para cada entrada e um threshold são empregados durante treinamento. Se o contador está acima do threshold e a entrada associada é '1', a saída será também '1'. Para os casos onde as entradas são idênticas, a saída tem o mesmo valor e é independente de contadores. No início do treinamento, os contadores são aleatoriamente pré-definidos com valores próximos ao threshold para que possam mudar de comportamento rapidamente. A nível global da topologia, nem

todos os neurônios têm contadores atualizados a cada apresentação de um exemplo. A atualização, a exemplo das redes de GSN, também começa no nível de saída para mover em direção ao nível de entrada, mas é interrompida quando se encontra um neurônio cujas entradas têm valor idêntico ao da saída final. Ao contrário do algoritmo original de GSN, ALN impõe direta correlação entre saída de neurônios e saída final. Este é o ponto explorado pelos anteriormente citados novos algoritmos de treinamento para GSN que possibilitam o uso de topologias não-piramidais.

3 Vantagens do Uso Cooperativo

As vantagens do uso do algoritmo de ALN em árvores binárias oriundas da conversão de redes de GSN são claras. A única desvantagem, aumento do período de treinamento, é fortemente justificada pela melhor performance e redução na complexidade da implementação finais.

Devido à usual forma de escolher topologias (árvores binárias com entradas randomicamente selecionadas em forma normal/complementada), a probabilidade de ALN descobrirem relações não-monotônicas é virtualmente zero (particularmente em níveis superiores). Redes de GSN podem mostrar quais relações são necessárias e passar este conhecimento para ALN. GSN adotam entradas em forma complementada sem maiores esforços conforme a necessidade dos mapeamentos. ALN sofrem quando o número de entradas é pequeno porque necessitam descartar todas as sub-árvores que não foram escolhidas numa forma admissível. A solução é aumentar o tamanho da rede, contrariando a busca de baixo custo e alta eficiência. Por fim, o fato de que GSN podem utilizar neurônios com mais de duas entradas pode ser utilizado para descobrir relações dificilmente encontradas por ALN. Sob ponto de vista das redes feed-forward de GSN, o algoritmo iterativo de ALN corrige as distorções geradas pela ordem de apresentação dos exemplos.

4 Conversão de GSN para ALN

Primeiramente, suponha que, no final do treinamento, todas as posições internas dos neurônios que compõem a rede de GSN contêm um dos valores Booleanos, '0' ou '1'. Neste estado, a simples aplicação de Álgebra Booleana (em particular, dos Teoremas de Morgan), a partir do neurônio do nível de saída em direção ao nível de entrada, é suficiente para resultar numa topologia equivalente onde todas as inversões (uso de NOT) estão aplicadas às entradas e todos os neurônios comportam-se como AND ou OR (Fig. 1). A modificação para que todos os neurônios tenham apenas duas entradas é trivial através da inclusão de novos neurônios. A topologia equivalente gerada já obedece o formato de ALN.

Normalmente, contudo, várias posições permanecem indefinidas durante todo o treinamento. Como ALN não trabalham com este tipo de valor, todas posições indefinidas devem ser modificadas de modo a que o caso se reduza ao anteriormente descrito. No caso de GSN com duas entradas, a preocupação por aumentar o poder discriminatório do neurônio justifica a busca do equilíbrio entre a frequência de utilização de ambos valores Booleanos. No caso de GSN com mais entradas, estratégias que analisam os vizinhos de cada posição indefinida são plausíveis. Este último caso será reportado em um futuro artigo. Será interessante notar o efeito final causado pelo descobrimento de relações que envolvem mais do que duas variáveis.

5 Trabalho Experimental e Resultados

Muitas simulações foram conduzidas para testar a validade das idéias anteriores. Cada resultado da tabela 1 é a média de 100 simulações. De modo a fazer boas comparações, topologias iniciais idênticas (aleatoriamente escolhidas) foram utilizadas para as versões originais e combinada. Redes menores foram utilizadas como componentes de modo que redes de 64 entradas são resultado do acoplamento de novas redes de

32 entradas em redes pré-existentis. Esta estratégia parece apropriada, pois é usual a necessidade de saber quanto melhor uma rede se torna quando mais entradas são utilizadas.

O conjunto de exemplos é composto de 160 imagens preto-branco (16 por 16 pixels) de dígitos manuscritos. Para avaliar, 800 padrões foram empregados. A tarefa não é simples porque várias superposições entre classes acontece no espaço de entrada. Nenhum pré-processamento foi realizado, as variáveis de entradas são os pixels. Os dados foram coletados através de um digitalizador onde pequenos quadrados indicavam onde o usuário deveria escrever. Para códigos de saída, empregou-se a codificação de Hadamard, ou seja, códigos de comprimento igual a 15 bits onde cada par de códigos mantém uma distância Hamming de 8 bits entre seus componentes. O número de entrada por rede é de 32, 64 ou 128. O número de redes por cada bit dos códigos de saída é 3, 5 ou 11. Todas entradas foram aleatoriamente associadas às posições dos níveis de entrada das redes. Nenhum uso da natureza bidimensional da tarefa foi empregado. Os autores acreditam, portanto, que o mesmo método aqui descrito seja adequado para outras tarefas da área de reconhecimento de padrões.

Desempenhos de k -NN classificadores podem ser vistos na tabela 2. Nenhum outro tipo de classificador possui treinamento mais rápido, pois consiste simplesmente em armazenar os exemplos disponíveis. A avaliação, contudo, é lenta porque as distâncias entre o padrão corrente e todos os exemplos armazenados devem ser computadas de modo a revelar qual é a opinião da maioria dos k vizinhos mais próximos. Por outro lado, com relação a taxa de reconhecimento, estes classificadores são bem reputados.

A superioridade do sistema resultante da combinação entre os dois modelos é mostrada pelos resultados (veja tabela 1). É interessante notar como o desempenho das redes de GSN afeta o sistema final. Além disso, a taxa de reconhecimento final do

sistema em sua versão mais forte, que utiliza mais redes e onde as redes têm maior capacidade, é comparável ao melhor desempenho de classificadores k -NN. Convém lembrar que, devido ao fato de que o grau de robustez nestes classificadores é diretamente proporcional ao número de vizinhos consultados, não é usual a utilização de um único vizinho.

6 Conclusão

Este artigo apresentou e comparou dois modelos de redes neurais onde a aprendizagem é desvinculada da variação de sinapses: redes feed-forward de Goal Seeking Neurons e Adaptive Logic Networks. Demonstrou-se, através de trabalho experimental (reconhecimento de dígitos manuscritos), como a aplicação do treinamento iterativo pertencente a ALN pode melhorar o desempenho de topologias derivadas de redes de GSN.

As idéias principais do procedimento de conversão de redes de GSN para o formato de ALN foram sucintamente descritas.

O sistema resultante da cooperação entre os dois modelos mostrou-se nitidamente superior. Além do desempenho, vários outros pontos respaldam a estratégia desenvolvida sob ponto de vista de ambos modelos envolvidos, destacando-se: a descoberta de relações não-monotônicas e inversões de variáveis de entrada para ALN e a redução da complexidade final (implicando em menor custo e maior velocidade) para GSN.

Acknowledgement

W Martins goza de licença da Universidade Federal de Goiás e é bolsista do CNPq em fase de finalização de doutoramento na Universidade de York, Inglaterra.

Referências

[1] Bowmaker, R G, and Coghill, G G, 1992, "Improved Recognition Capabilities for Goal Seeking Neuron", in *Electronic Letters*, vol. 28, No. 3. 220-221.

- [2] Martins, W and Allinson, N M, 1993, "Two improvements for GSN neural networks", in *Proc. Weightless Neural Network Workshop '93*, 58-63, York, UK.
- [3] Carvalho, A de et al, 1994, "Progressive learning algorithm for GSN feedforward neural architectures", *Electronic Letters*, vol. 30, No. 6, 506-507.
- [4] Muselli, M. 1994, "Hamming clustering: improving generalization in binary neural networks", *Proc. of the Intl Conf on Artificial Neural Networks*, vol. 2, 1083-1086, Sorrento, Itália.
- [5] Bisset, D L et al, 1989, "A comparative study of neural network structures for practical application in a pattern recognition environment" in *Proc. First IEE Interl. Conf. on Artificial Neural Networks*, 378-382, London, UK.
- [6] Armstrong, W W and Gecsei, J, 1979, "Adaptation Algorithms for Binary Tree Networks", *IEEE Trans on System, Man and Cybernetics*, vol. 9, 276-285.
- [7] Martins, W and Allinson, N M, 1994, "Improving GSN-based classifiers by control of presentation order", in *Proc. do I Simpósio Brasileiro de Redes Neurais*, Caxambu, MG, artigo aceito.
- [8] Filho, E, 1990, "Investigation of boolean neural network based on a novel Goal-Seeking-Neuron", PhD Thesis, Electronic Engineering Laboratories, University of Kent, UK.
- [9] Armstrong, W W and Bochmann, G v, 1974, "Properties of Boolean Functions with a Tree Decomposition", *BIT*, vol. 13, 1-13.
- [10] Stein, R B et al, 1992, "Methods to Control Functional Electrical Simulation in Walking", *Proc. of the First Intl FES Symposium*, Sendai, Japan, 135-140.
- [11] Armstrong, W W and Thomas, M M, 1994, "Control of a Vehicle Active Suspension System Model using Adaptive Logic Networks", *Program Addendum of the World Congress on Neural Networks*, San Diego, USA, 9-14.

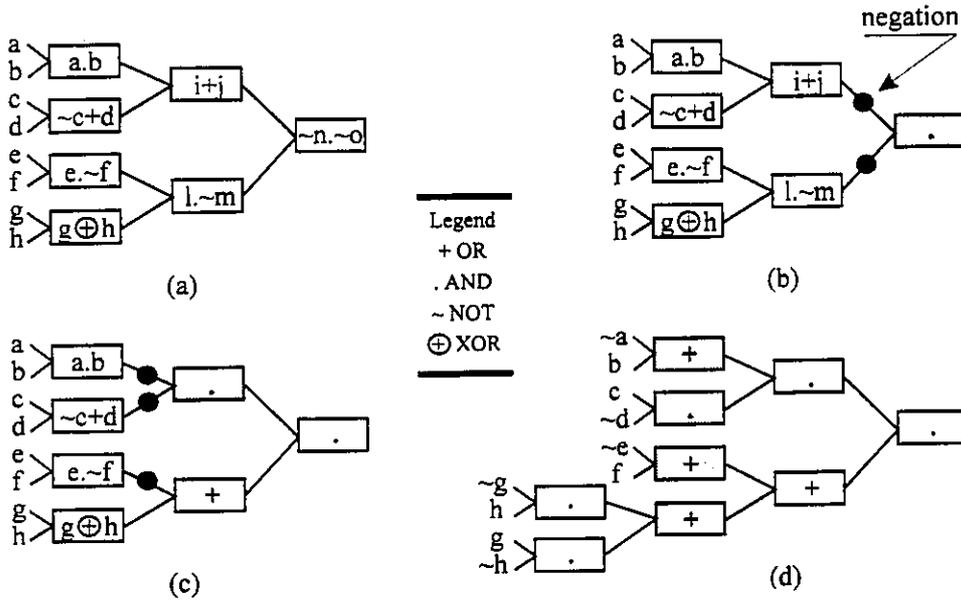


Figura 1: Exemplo de conversão de GSN para ALN: (a) estado inicial; (b) conversão do nível de saída; (c) conversão de nível intermediário; (d) estado final.

TREINAMENTO			SITUAÇÃO	AVALIAÇÃO		
redes per bit/entradas per rede				redes per bit/entradas per rede		
3 / 32	5 / 32	11 / 32		3 / 32	5 / 32	11 / 32
53.59	63.43	74.80	original GSN	39.57	47.21	56.95
73.46	80.76	85.08	original ALN	57.98	66.76	74.20
80.42	89.36	96.70	combinação	61.19	71.40	83.78
3 / 64	5 / 64	11 / 64		3 / 64	5 / 64	11 / 64
59.04	68.84	80.63	original GSN	43.78	52.25	64.35
81.40	84.79	87.76	original ALN	67.57	72.85	77.32
91.84	96.02	98.78	combinação	74.83	82.48	88.76
3 / 128	5 / 128	11 / 128		3 / 128	5 / 128	11 / 128
66.34	75.49	85.49	original GSN	47.57	56.69	68.65
85.12	87.68	88.94	original ALN	70.26	74.48	77.09
97.21	98.71	99.51	combinação	82.03	87.25	90.53

Tabela 2: Taxa de reconhecimento para o conjunto de exemplos do treinamento e o conjunto de padrões de teste.

# vizinhos k	reconhecimento treinamento (%)	reconhecimento avaliação (%)
1	100.00	91.50
2	96.88	84.75
3	91.88	79.88
4	92.50	81.75
5	93.13	78.00
15	85.63	69.87

Tabela 2: Performances de classificadores k-NN.