

Reconhecimento de Alvos utilizando Redes Neurais Construtivas

ERNESTO CUADROS VARGAS

HUMBERTO COSTA DE SOUSA

ANDRÉ CARLOS PONCE DE LEON FERREIRA DE CARVALHO

LABIC-- Laboratório de Inteligência Computacional

Instituto de Ciências Matemáticas e de Computação

Universidade de São Paulo

Caixa Postal 668, São Carlos, SP, Brasil

{ecvargas,betovs,andre}@icmsec.sc.usp.br

Abstract. This paper presents the use of Constructive Neural Networks for Patterns Recognition. The work presented here is part of the SAPRI project (System for Acquisition, Processing and Images Recognition) which is being developed for the Brazilian Navy. The objective of this paper is to investigate the behavior of Constructive Neural Networks in image recognition tasks, specially with images from the SAPRI project. Constructive Neural Networks differs from standard networks due to its ability to change its own number of elements, adding or removing units and connections. Four constructive algorithms are presented in this paper: Cascade Correlation, Tower, Pyramid and Upstart.

Keywords: Pattern Recognition, Neural Networks.

1 Introdução

Este trabalho é parte do desenvolvimento de um Sistema para Aquisição, Processamento e Reconhecimento de Imagens, SAPRI (1997). Este sistema será particularizado para a aplicação de tratamento de imagens de radar, fornecendo a especificação de um ambiente voltado para a segurança da navegação e para o controle de tráfego aeronaval.

O objetivo principal do projeto SAPRI é de contribuir para o desenvolvimento de um sistema de processamento de imagens que possa, baseado na utilização de avançados métodos e dispositivos computacionais, dar apoio ao processo de manipulação e sobretudo interpretação de imagens radar.

O projeto SAPRI é composto de 4 módulos que compreendem as etapas de aquisição, processamento e reconhecimento e interpretação de imagens, cujas informações são utilizadas no último módulo referente à análise tática.

Existem quatro abordagens, Parekh et al. (1997), para a escolha da topologia de uma Rede Neural. A abordagem Empírica consiste em testar várias topologias até encontrar uma que apresente um resultado satisfatório. A abordagem Genética consiste gerar variações de uma rede e combinar as melhores características das novas redes para melhorar o desempenho. A terceira abordagem é conhecida como "Pruning" ou Poda. Esta abordagem consiste em otimizar uma rede através da eliminação dos elementos que não influenciam no resultado. Finalmente a quarta

abordagem é a Construtiva; nesse caso as redes começam com uma topologia mínima e de acordo com as necessidades do problema vão inserindo novas unidades, melhorando o seu desempenho.

Este trabalho visa auxiliar o módulo de processamento e reconhecimento de imagens através da inspeção do comportamento de Redes Neurais Construtivas. Estas redes adaptaram o tratamento de redes perceptrons permitindo a inclusão de novos elementos durante a fase de treinamento. Assim, uma rede de topologia mínima é treinada, e cresce de acordo com a necessidade da aplicação.

A seção 2 introduz o conceito de reconhecimento de alvos como padrões, e a seção 3 apresenta a descrição de quatro algoritmos para o treinamento de Redes Neurais Construtivas. Experimentos práticos realizados com dados fornecidos pelo projeto SAPRI são apresentados na seção 4. Finalmente, na seção 5 as conclusões deste trabalho são discutidas.

2 Reconhecimento de Alvos como Padrões

Reconhecimento automático de alvos (ATR) refere-se à tarefa de identificar e classificar alvos. A execução desta tarefa depende de quatro fases distintas. A primeira fase diz respeito à escolha do modo de captura da imagem, isto é, o tipo de sensor mais adequado ao problema em questão. As qualidades das informações geradas pelo sensor, bem como sua fidelidade à imagem são características fundamentais para garantir o bom desempenho das fases posteriores. A captura da imagem deve ser criteriosa, afim de que nenhuma característica relevante ao problema passe despercebida.

A segunda fase é a de Pré-processamento e Segmentação da informação recebida. O pré-processamento é necessário para amenizar ruídos e eliminar informações incorretas da imagem. A segmentação tem por objetivo a separação de alvos que possivelmente tenham sido interligados na captura dos mesmos.

A seguir vem a fase de Extração e Seleção de Características. A extração de características tem por objetivo captar informações relevantes dos alvos. No fim destes processos é obtido um conjunto de dados mais compacto. Isto é feito para que não haja desperdício com processamento de informações desnecessárias.

Por fim a Classificação associa cada padrão gerado pelas fases descritas acima a uma classe. Esta fase é de suma importância para o sistema, pois um erro pode gerar muitos problemas em aplicações práticas.

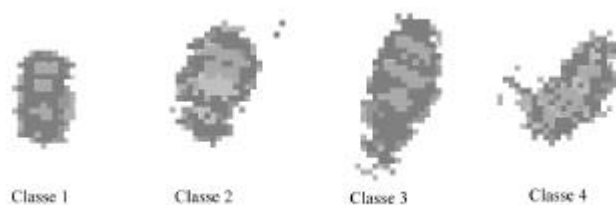


Figura 1 : Imagens de navios após o pré-processamento

No projeto SAPRI as imagens de navios capturadas por radar foram organizadas em quatro classes. A **Figura 1** apresenta um exemplo de cada classes.

3 Redes Neurais Construtivas

Redes Neurais Artificiais (RNA) apresentam um bom desempenho na classificação de padrões. Porém a topologia de uma RNA tradicional deve ser definida antes do início do treinamento, como é o caso de redes Perceptron Multi-Camadas. Como não existe regra para a determinação da topologia da rede, sua escolha depende basicamente de conhecimento empírico, ou seja, várias topologias precisam ser testadas antes da escolha da topologia definitiva.

Esta desvantagem dos modelos tradicionais de RNAs, pode se agravar se após a escolha da melhor topologia, o conjunto de padrões se alterar, tornando necessário uma nova escolha da topologia para o retreinamento da rede.

Redes Neurais Construtivas oferecem uma abordagem atraente para a construção gradativa de Redes Neurais a partir de uma topologia mínima. Ou seja, no início do treinamento uma rede mínima é criada, possuindo apenas camadas de entrada e saída com o número necessário de neurônios para modelar o problema em questão. A rede, então, altera sua topologia através da inserção ou remoção de elementos durante a fase de treinamento, evitando assim, a necessidade de se escolher a topologia empiricamente.

O aumento ou redução de elementos da rede é controlado pelo algoritmo de treinamento utilizado. Abaixo estão citados os algoritmos estudados para implementação deste trabalho.

3.1 Cascade Correlation

O algoritmo Cascade Correlation, Fahlman et al. (1991), utiliza uma técnica de aprendizado supervisionado para treinar RNAs. Este algoritmo começa com uma rede mínima (apenas camadas de entrada e saída) e, durante o treinamento, insere novas unidades intermediárias, uma por vez, criando uma estrutura de múltiplas camadas.

A nova unidade é conectada à entrada, saída e às unidades intermediárias existentes da rede. Quando uma nova unidade intermediária é inserida, os pesos de suas entradas (conexões) são congelados. Este neurônio começa então a influenciar permanentemente nas operações da rede, sendo utilizado para detectar novas características no grupo de padrões.

A unidade a ser incluída na rede pode ser selecionada de um conjunto de candidatos organizados em uma camada. Esta camada de candidatos é conectada à camada de entrada e às camadas intermediárias já existentes, mas não é conectada à saída, pois não deve influir diretamente no resultado da rede. O critério de seleção do candidato é a correlação que este candidato tem com a saída da rede. Quanto maior a correlação, mais influência este candidato tem no resultado da rede. Portanto, os pesos das conexões entre candidatos e as camadas de entrada e intermediárias devem ser definidos de maneira a maximizar a correlação entre os candidatos e a saída. Assim, o candidato que apresentar o maior valor de correlação será incluído na rede como uma nova camada intermediária, sendo conectado a todas as camadas da rede, como ilustra a **Figura 2**.

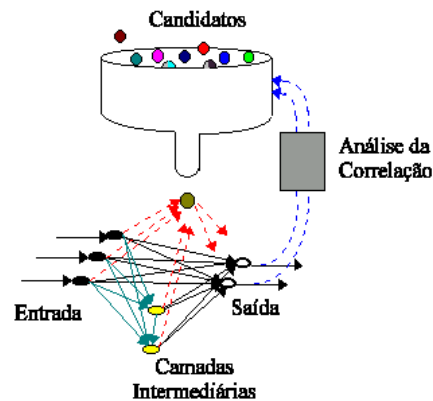


Figura 2: Seleção da uma nova unidade entre um conjunto de candidatos

Entre as vantagens das redes Cascade Correlation, está a velocidade de treinamento, que pode ser flexível dependendo do número de novas unidades incluídas. O treinamento destas redes pode ser observado mais claramente nas figuras abaixo. A **Figura 3** mostra o estado inicial da rede (topologia mínima), a **Figura 4** mostra a inserção de uma nova unidade intermediária, e a **Figura 5** mostra a inserção de uma segunda unidade.

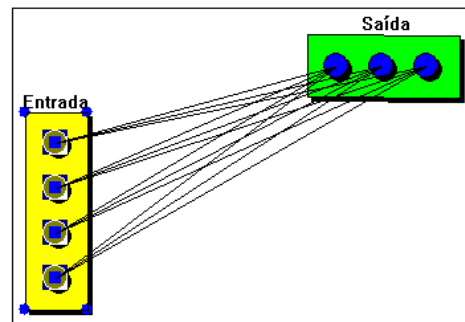


Figura 3: Estado inicial da rede Cascade Correlation

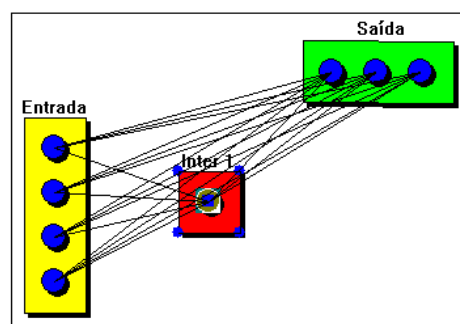


Figura 4: Estado da rede após inserção de um neurônio

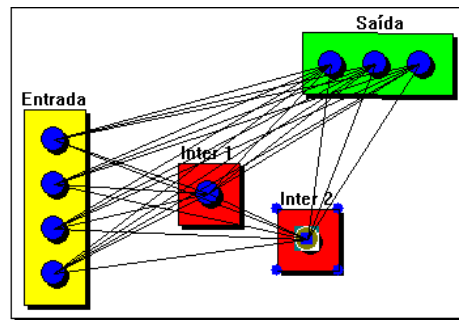


Figura 5: Estado da rede após inserção de um segundo neurônio

3.2 Tower

O algoritmo Tower, Parekh et al. (1997), constrói uma torre de unidades perceptron, também conhecidas como Threshold Logic Units (TLUs).

Ao contrário do algoritmo Cascade Correlation que insere uma unidade por vez durante o treinamento, este algoritmo insere uma camada com um número fixo de neurônios.

A nova camada inserida possui o mesmo número de neurônios que a camada de saída. Esta camada é conectada à camada de entrada e à camada de saída. A camada de saída se torna então uma camada intermediária, e a camada inserida se torna a nova camada de saída. Este procedimento é repetido até que a precisão de classificação desejada seja atingida. As figuras abaixo ilustram o processo de construção da rede. A **Figura 6** mostra o estado inicial da rede (topologia mínima). A **Figura 7** mostra a inserção de uma nova camada e a **Figura 8** mostra a inserção de uma segunda camada.

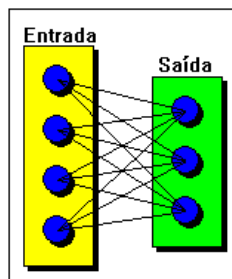


Figura 6: Estado inicial da rede Tower

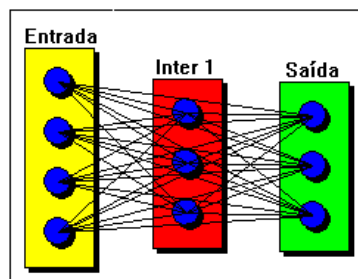


Figura 7: Inserção de uma nova camada, a camada de saída torna-se uma camada intermediária

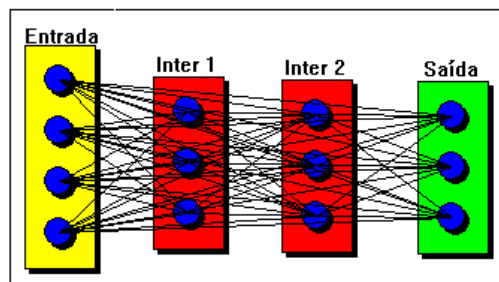


Figura 8: Inserção de um segunda camada, novamente, a camada de saída torna-se uma camada intermediária

3.3 Pyramid

O algoritmo Pyramid, Parekh et al. (1997), consiste numa extensão do algoritmo Tower citado anteriormente. Este algoritmo difere do algoritmo Tower apenas na maneira como são feitas as conexões da nova camada. A camada inserida, além de conectada à camada de entrada e à camada de saída, é também conectada a todas as camadas intermediárias existentes na rede, tornando as unidades da rede totalmente conectadas entre si. A **Figura 9** apresenta uma rede já treinada, com duas camadas intermediárias. Note que a camada **Inter 1** está conectada à nova camada de saída, o que não acontece no algoritmo Tower.

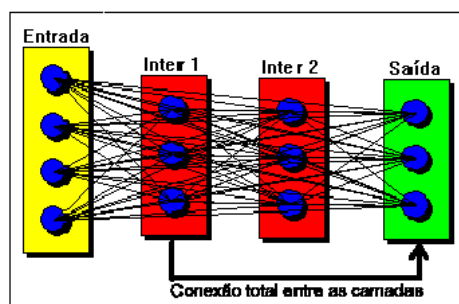


Figura 9: Rede Pyramid. A seta indica a conexão entre a camada Inter 1 e a camada de saída

3.4 Upstart

O algoritmo Upstart, Freat (1990), insere neurônios em pares (unidades X e Y), formando uma estrutura em forma de árvore binária a partir do neurônio da camada de saída. Cada par de neurônios é inserido para corrigir o erro da unidade à qual está ligado. O funcionamento destas unidades é descrito posteriormente. Uma vez treinados, os novos neurônios podem permitir a inserção de novos pares de neurônios, até que a rede apresente uma taxa de erro aceitável. A **Figura 10** e a **Figura 11** ilustram este algoritmo.

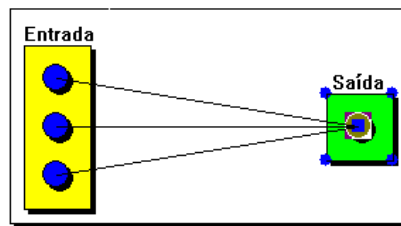


Figura 10: Estado inicial da rede Upstart, com apenas uma unidade de saída.

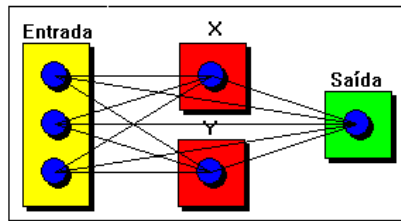


Figura 11: Rede Upstart após a inserção de unidades X e Y.

Ao final do treinamento, a rede resultante pode se tornar muito “profunda”, isto é, com um grande número de camadas intermediárias. Outra desvantagem deste algoritmo é a restrição da aplicação do mesmo para problemas de classificação com 2 classes.

Para suprir estas desvantagens, uma outra abordagem foi estudada e aplicada. Esta nova abordagem, Parekh et al. (1997), cria uma estrutura equivalente, porém os novos neurônios são incluídos em uma única camada. O processo de adição de novas unidades nesta camada intermediária pode ser descrito da seguinte forma:

- após o treinamento da rede mínima, o neurônio da camada de saída que tiver a maior taxa de erros é identificado;
- dependendo de qual erro é mais freqüente, erroneamente ativado (saída = 1, desejada = 0) ou desativado (saída = 0, desejada = 1), uma unidade X ou Y é criada para corrigir alguns dos erros desta unidade de saída;
- a unidade criada é conectada à entrada e seus pesos são treinados separadamente de acordo com o erro que deve corrigir;
- a unidade é então adicionada à camada intermediária e conectada à camada de saída para continuação do treinamento.

Esta abordagem permite também a utilização de mais de um neurônio na camada de saída. As figuras abaixo ilustram a inserção de uma nova unidade de acordo com esta abordagem do algoritmo Upstart. A **Figura 12** apresenta uma rede com três neurônios já inseridos anteriormente, e a **Figura 13** ilustra a inserção da nova unidade, representada pelo neurônio selecionado.

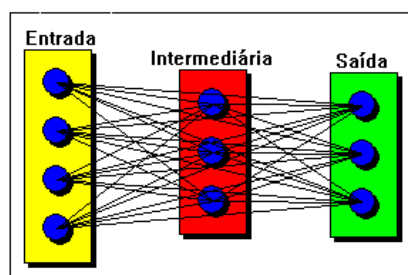


Figura 12: Rede Upstart com três neurônios já inseridos

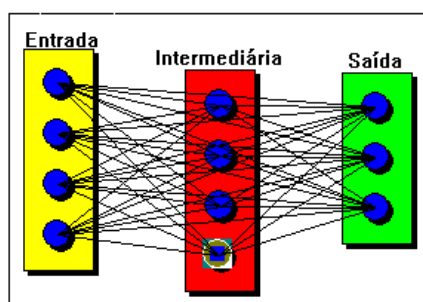


Figura 13: Inserção de um novo neurônio em uma rede Upstart

4 Experimentos

Foram realizados experimentos utilizando os algoritmos construtivos *Cascade Correlation* e *Tower*. Os resultados obtidos foram comparados com os resultados provenientes do algoritmo *BackPropagation*. Os dados utilizados foram imagens de navios (alvos de radar) fornecidas pelo Instituto de Pesquisa da Marinha do Brasil (IPqM).

Os padrões fornecidos pelo Projeto SAPRI são vetores de 1024 elementos com valores hexadecimais. Estes valores representam o nível de intensidade da cor de um pixel em uma determinada imagem capturada pelo radar.

Esses dados foram normalizados, ou seja, foram organizados em matrizes de ordem 32, e os valores convertidos em números reais no intervalo $[0; 1]$.

Ao todo, foram 800 padrões, sendo 200 destes para cada uma das quatro classes. Cada classe representa um tipo de embarcação, sendo que a escolha dos navios e a separação destes em classes foi realizada durante a captura das imagens.

Para a execução do experimento, os 800 padrões foram aleatoriamente separados em três conjuntos: treinamento, validação e teste, com proporções de 50%, 25% e 25%, respectivamente, seguindo as regras de **Proben1**, Prechelt (1994).

O conjunto de regras **Proben1** especifica algumas convenções com o intuito de eliminar falhas na validade do experimento.

A topologia da rede treinada com o algoritmo *Back Propagation* foi fixada em 1024-8-4 (neurônios nas camadas de entrada, intermediária e saída, respectivamente), obtida através de experimentação empírica com outras topologias. A seguir, na **Tabela 1**, podem ser observadas as

médias das proporções de padrões corretamente reconhecidos (%) acompanhado de seu respectivo desvio padrão.

	Treinamento	Validação	Teste
BackPropagation	99,70 \pm 0,326	98,30 \pm 1,151	98,40 \pm 1,557
Cascade Correlation	99,60 \pm 0,326	96,40 \pm 1,40	97,40 \pm 1,387
Tower	100,00 \pm 0,000	95,7 \pm 2,683	96,90 \pm 1,084

Tabela 1: Resultados do treinamento de redes com os dados de navios da Marinha do Brasil.

A tabela acima indica que todos os algoritmos apresentaram excelentes resultados. Todas as redes apresentaram taxas de reconhecimento satisfatórias e são capazes de generalizar. A rede treinada com o algoritmo *Back Propagation* apresentou os melhores resultados de validação e teste. Contudo, é necessário notar que a topologia escolhida para a rede treinada com o algoritmo *Back Propagation* foi determinada empiricamente, após a comparação com outras diversas topologias. A maior vantagem em se utilizar algoritmos construtivos, nesse caso, é a economia de tempo e esforço na obtenção de uma topologia que seja suficientemente eficiente na resolução do problema e ao mesmo tempo a mais compacta.

Os outros modelos apresentados neste trabalho estão atualmente em fase de desenvolvimento e teste, impossibilitando a execução dos experimentos. Entre os modelos implementados e os em desenvolvimento encontram-se os algoritmos: *BackPropagation*, *Cascade Correlation*, *Upstart*, *Tower* e *Pyramid*. Pretende-se implementar ainda os algoritmos RBF-DDA, Tiling e outros algoritmos construtivos, Parekh et al. (1997), que possam ser utilizados no projeto SAPRI.

5 Conclusão

Neste trabalho foram apresentados alguns modelos de RNC que foram utilizados para o Reconhecimento de Alvos. As RNC se caracterizam por melhorar o seu desempenho por meio da inserção de novas unidades, camadas ou conexões. A inserção de novas unidades na rede é responsabilidade do algoritmo que esteja sendo utilizado. Entre os algoritmos apresentados, *Cascade Correlation* e *UpStart* inserem uma unidade por vez. Os algoritmos *Tower* e *Pyramid* inserem uma camada por vez. Dessa forma, os algoritmos construtivos minimizam o erro e melhoram o seu desempenho.

Aplicando os algoritmos construtivos para o Reconhecimento de Alvos podemos observar um melhor desempenho em relação às redes neurais convencionais com topologia predeterminada. Ao contrário da abordagem empírica, as RNC se ajustam de acordo com o problema que está sendo resolvido e, dessa forma, evita ao usuário o desgaste em procurar a melhor topologia para cada problema.

Para realizar os experimentos citados na seção 4, foi utilizado um simulador para Redes Neurais Ontogênicas – **Kipu**, Cuadros (1998). Este simulador está sendo desenvolvido no Instituto de Ciências Matemáticas e de Computação com o apoio da CAPES e do CNPq. As

figuras de redes neurais contidas neste artigo também foram obtidas através da interface deste simulador.

Referências

- Cuadros, E.; Álvarez, M.; Carvalho, A.C.P.L.F. *Multi-Threads Object Oriented Simulator for Ontogenic Neural Networks*. In: Proceedings of the International Conference on Computational Intelligence and Multimedia Applications. pp. 276-281. World Scientific. 1998.
- Fahlman, S. E.; Lebiere, C. *The Cascade-Correlation Learning Architecture*. Technical Report. School of Computer Science. Carnegie Mellon university. 1991.
- Frean, M. *The Upstart Algorithm: a method for constructing and training feed-forward neural networks*. Technical Report. Department of Physics and Centre for Cognitive Science. Edinburgh University. 1990.
- Freeman, J.; Skapura, D. *Neural Networks, Principles, Algorithms, and Applications*. Addison Wesley. 1993.
- Parekh, R.; Yang, J.; Honavar, V. *Constructive Neural Network Learning Algorithms for Multi-Category Real-Valued Pattern Classification*. Technical Report. Department of Computer Science. Iowa State University. 1997.
- Prechelt, L. Proben1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules. Technical Report. 21/94. Fakultät für Informatik. Universität Karlsruhe. 1994.
- SAPRI. *Image Acquisition, Processing, and Recognition System(SAPRI)*.1997. Disponível em WWW. URL:<http://www.di.ufpe.br/~sapri>