

1º Congresso Brasileiro de Redes Neurais

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

RECONHECIMENTO DE PEÇAS UTILIZANDO REDES NEURAIAS

Idmilson H. Sepeda Filho*
Marcelo R. Stemmer

Laboratório de Controle e Microinformática
Universidade Federal de Santa Catarina
88040-900, Florianópolis, SC, Brasil,
fax:: (0482)31-9770
e-mail: idmilson@lcmi.ufsc.br
marcelo@lcmi.ufsc.br

RESUMO

O reconhecimento de objetos por computador é uma área de grande interesse. Novos paradigmas, como redes neurais, estão sendo utilizados com o propósito de conseguir sistemas que sejam robustos e que executem esta tarefa rapidamente. Este trabalho mostra a utilização de uma rede neural *feedforward* para reconhecimento de peças mecânicas.

Palavras chave: Reconhecimento de Objetos, Redes Neurais, visão

A System for Workpiece Recognition using Neural Networks

ABSTRACT

The object recognition by computer is an area of great interest. New paradigms as neural networks are now been used to obtain sytems that are robust and solve this task faster. This paper shows the use of a

feedforward neural network in mechanical workpiece recognition.

Key words: Object Recognition, Neural Networks, vision

1 INTRODUÇÃO

Visão por computador é uma área que vem recebendo muito interesse nos últimos anos. Já existem métodos para reconhecimento de padrões utilizando algoritmos convencionais de processamento de imagem.

As redes neurais surgem como um novo paradigma nesta área. Suas habilidades especiais como a capacidade de aprender por exemplos, tolerância a entradas corrompidas e reconhecimento de padrões mostram sua utilidade em aplicações de processamento de imagem.

Uma comparação entre técnicas convencionais de processamento de imagem e a técnica utilizando redes neurais é mostrada na tabela 1 [9].

As redes neurais são compostas de elementos que realizam muitas funções que são análogas as funções elementares do neurônio biológico

Além da semelhança superficial com a anatomia do cérebro, essas redes exibem algumas características deste. Por exemplo, elas aprendem por experiência. Mas apesar dessas similaridades funcionais, redes neurais artificiais estão longe de duplicar as funções do cérebro humano.

Redes neurais artificiais podem modificar seu comportamento em resposta a seu ambiente. Este fato, mais que qualquer outro, é responsável pelo interesse que vêm

* Bolsista do PICD-UFPa/CAPES

recebendo. Diz-se então que ela pode aprender (*learn*). Existe uma grande variedade de algoritmos de treinamento, todos com seus pontos fortes e fracos.

	Area	Boundary Following	Histograma	Redes Neurais
Tempo de processamento (segundos)	0.44	0.16	0.44	0.68 (0.28)
Precisão (%)	96	72	100	100
Orientação	-	-	Firme	Solta
Tempo de Ajuste (minutos)	10	15	10	60

Tabela 1 - Comparação de Técnicas de Processamento de imagens

Uma vez treinada, uma resposta da rede pode ser insensível à pequenas variações na sua entrada. Esta habilidade é essencial para o reconhecimento de padrões no mundo real, por causa de ruídos ou distorções de padrões (imperfeições) do mundo em que vivemos. É importante notar que os resultados das redes são obtidos a partir de sua estrutura e não pelo uso de inteligência humana embutida em alguma forma de programa de computador.

Propõe-se neste trabalho a utilização da técnica de redes neurais no reconhecimento de peças mecânicas a serem transportadas por uma esteira e identificadas por meio de uma câmera CCD (Charge Coupled Device). O sistema de reconhecimento de peças é parte de uma célula flexível de manufatura (FMC), mostrada na figura 1.

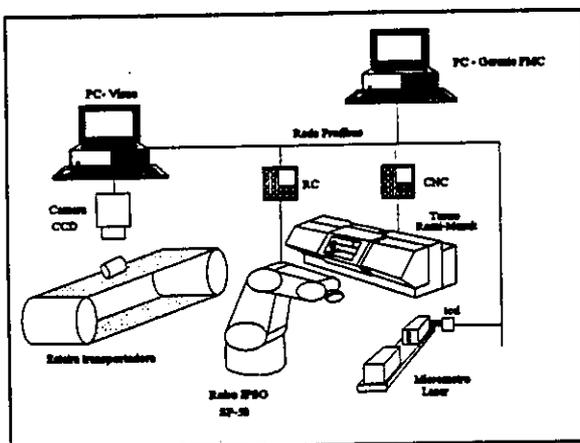


Figura 1 - Estrutura da FMC em implantação

2 ESCOLHA DE UMA ARQUITETURA DE REDE E ALGORITMO DE TREINAMENTO

Tendo em vista os objetivos propostos foi escolhida a rede counterpropagation, desenvolvida por Robert Hecht-Nielsen, para implementar o sistema de reconhecimentos de peças. Em comparação com Backpropagation, ela é de uso mais restrito, entretanto seu tempo de treinamento é bem menor. Esta característica pode ser importante, tendo em vista a tendência atual de maior diversificação e diminuição do tamanho dos lotes de peças produzidas, requerendo assim aprendizado frequente de novas imagens.

Counterpropagation é a combinação de dois algoritmos bem conhecidos: o *self-organizing map* de Kohonen e o *outstar* de Grossberg.

O processo de treinamento associa vetores de entrada com correspondentes vetores de saída. Esses vetores são números reais normalizados entre 0 e 1. Uma vez que a rede esta treinada, a aplicação de um vetor de entrada produz o desejado vetor de saída. A capacidade de generalização da rede permite que ela produza uma saída correta, mesmo quando é dado um vetor de entrada parcialmente incompleto ou parcialmente incorreto.

A figura 2 mostra uma versão simplificada da rede counterpropagation, que ilustra as características funcionais deste paradigma.

Os neurônios na camada 0 (mostrados como círculos) servem somente como *fan-out* e não efetuam nenhuma computação. Cada neurônio na camada 0 conecta-se com todos os neurônios na camada 1 (chamada camada Kohonen) através de um peso w_{mn} . Eles serão coletivamente referidos como a matriz de pesos W . Similarmente, cada neurônio na camada Kohonen (camada 1) conecta-se com todos os neurônios na camada Grossberg (camada 2) por um peso v_{np} , aqui referenciadas como a matriz de pesos V .

Camada Kohonen

A camada Kohonen foi treinada na sua forma mais simples, onde o "ganhador leva tudo" ("winner-take-all"), isto é, para um dado vetor de entrada, uma e somente uma saída dos neurônios Kohonen é logicamente 1; todas as outras são zero.

Associado com cada neurônio Kohonen esta um conjunto de pesos conectando-o à cada vetor de entrada. Como com neurônios em muitas redes, a saída NET de cada neurônio Kohonen é simplesmente a somatória das entradas ponderadas. Isto pode ser expressado como segue:

$$NET_j = w_{1j}x_1 + w_{2j}x_2 + \dots + w_{mj}x_m$$

onde NET_j é saída NET do neurônio Kohonen j.

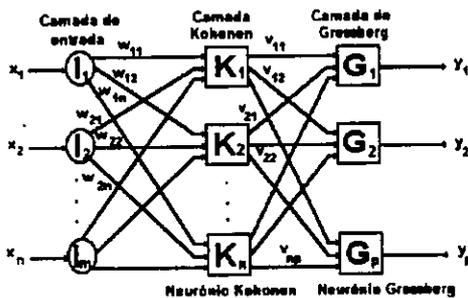


Figura 2 -Rede Counterpropagation Feedforward

O neurônio Kohonen com maior valor NET é o "ganhador". Sua saída é setada para 1; todas as outras são setadas para zero.

Camada Grossberg

A camada Grossberg funciona de maneira similar. Sua saída NET é a somatória ponderada das saídas da camada Kohonen K_1, K_2, \dots, K_n , formando o vetor K. A saída NET de cada neurônio Grossberg é então

$$NET_j = \sum_i k_i v_{ij}$$

onde NET_j é a saída do neurônio Grossberg j.

Como a camada Kohonen é operada tal que somente uma saída NET é 1 e todas as outras são zero, somente um elemento do vetor K não é zero, e o cálculo é simples. De fato, a única ação de cada neurônio na camada Grossberg é colocar na saída o valor do peso que conecta-o ao único neurônio Kohonen que não é zero.

Treinamento da Camada Kohonen

A camada Kohonen classifica os vetores de entrada dentro de grupos que são similares. Isto é conseguido com o ajuste dos pesos da camada Kohonen tal que vetores de entrada similares ativam o mesmo neurônio Kohonen. Daí em diante é responsabilidade da camada Grossberg produzir a saída desejada.

O treinamento Kohonen é um algoritmo *self-organizing* que opera em modo não supervisionado. Por esta razão, é difícil (e não necessário) prever qual neurônio Kohonen específico será ativado por um dado vetor de entrada. Só é necessário assegurar que o treinamento separe vetores de entrada que não sejam similares.

O neurônio com maior produto é declarado o "ganhador" e seus pesos são ajustados. Como a operação produto usada para calcular os valores de NET é uma medida de similaridade entre os vetores de entrada e peso, o processo de treinamento consiste na seleção do neurônio Kohonen cujo vetor peso é mais similar ao vetor de entrada, e torná-lo ainda mais similar, como mostra a equação abaixo.

$$w_{novo} = w_{velho} + \alpha(x - w_{velho})$$

onde:

w_{novo} = o novo valor de um peso conectando um componente de entrada x para o neurônio ganhador.

w_{velho} = o valor anterior deste peso.

A variável α é um coeficiente de taxa de treinamento que usualmente inicia próximo a 0.7 [1] e é reduzida durante o treinamento para 0.

Setando α para valores baixos reduz-se o efeito de cada passo do treinamento, fazendo o valor final uma média dos vetores de entrada para o qual ele foi treinado. Desta maneira, os pesos associados com um neurônio assumirão um valor próximo ao "centro" dos vetores de entrada para o qual aquele neurônio é o "ganhador".

Os pesos foram setados para valores iniciais antes do início do treinamento. É prática comum em redes neurais randomizar os pesos para pequenos valores.

Treinamento da Camada Grossberg

A camada Grossberg é relativamente simples de treinar. Um vetor de entrada é aplicado, as saídas Kohonen são estabelecidas, e a saída Grossberg é calculada como na operação normal. A seguir, cada peso é ajustado somente se ele está conectado a um neurônio Kohonen que tem uma saída diferente de zero. A quantidade de ajuste no peso é proporcional a diferença entre o peso e a saída desejada do neurônio Grossberg para o qual ele está conectado. Em símbolos

$$v_{ij\text{ novo}} = v_{ij\text{ velho}} + \beta(y_j - v_{ij\text{ velho}})k_i$$

onde

k_i = a saída do neurônio Kohonen i (somente os neurônios diferentes de zero).

y_j = componente j do vetor de saídas desejadas. Inicialmente β é setado em aproximadamente 0.1 [1] e é gradualmente reduzido com o progresso do treinamento.

Com isso pode-se ver que os pesos da camada Grossberg irão convergir para valores médios das saídas desejadas, considerando que os pesos da camada Kohonen são treinados para valores médios das entradas. O treinamento Grossberg é supervisionado; o algoritmo tem a saída desejada para o qual ele é treinado. A não

supervisionada operação de auto organização da camada Kohonen produz saídas em posições indeterminadas; elas são mapeadas para as saídas desejadas pela camada Grossberg.

3 PRÉ-PROCESSAMENTO DA IMAGEM

Foi utilizada uma câmera CCD para obter as imagens das peças a serem reconhecidas. Cada imagem compõe-se de 512 x 512 pixels, sendo que apenas 480 x 512 pixels são aproveitados. As imagens são em níveis de cinza que variam de 0 a 255.

Verificou-se nesta etapa que a iluminação feita sobre a peça é de grande importância. A iluminação deve ser feita de forma a não gerar sombras. A estrutura para obtenção da imagem está mostrada na figura 3.

Desde de que seria totalmente inviável mapear diretamente cada pixel da imagem para um neurônio de entrada da rede (seriam necessários 245.760 neurônios) existe a necessidade de pré-processar a imagem.

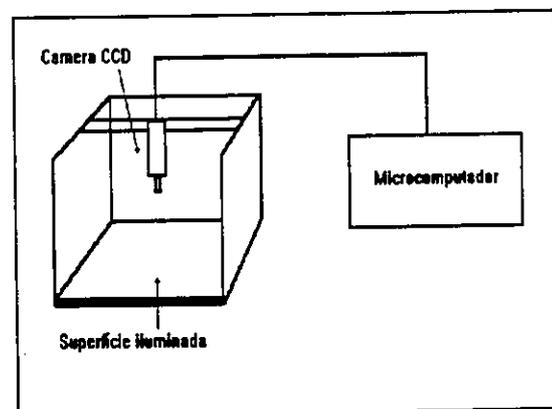


Figura 3 - Estrutura para obtenção da imagem

Para cada peça foram traçadas duas curvas de intensidade [10], uma no eixo X e outra no eixo Y, com 20 pontos cada uma. As curvas de intensidade são obtidas pela soma dos níveis de cinza em cada linha e coluna da imagem, respectivamente. Desse modo, cada peça teria uma identidade própria.

4 IMPLEMENTAÇÃO E RESULTADOS

Tanto o programa para pré-processamento da imagem quanto o que implementa a rede neural foram escritos em Turbo C.

A rede neural foi implementada com 40 neurônios na camada de entrada, 235 neurônios na camada Kohonen e 13 neurônios na camada Grossberg. Foram utilizadas 13 peças, das quais foram obtidas imagens de 20° em 20°. A imagem de uma das peças utilizadas, a curva de intensidade correspondente e o comportamento do erro médio quadrático são mostrados nas figuras 4, 5 e 6, respectivamente.

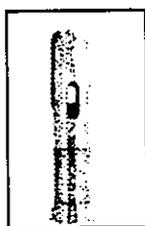


Figura 4 - Exemplo de uma peça utilizada

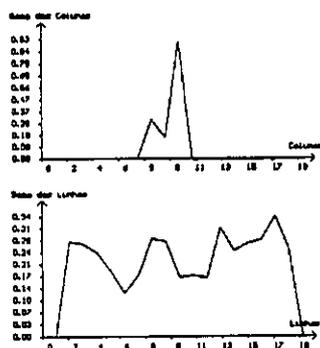


Figura 5 - Curvas de intensidade da peça exemplo

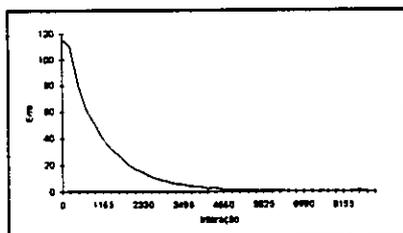


Figura 6 - Comportamento do erro durante o treinamento

Resultados experimentais obtidos com um conjunto de peças, conduziram a um

tempo de treinamento relativamente baixo (aproximadamente de 40 minutos) em IBM-PC 386DX40 com, com uma taxa de acerto no reconhecimento de cerca de 86% para a rede que reconhece as peças, que após o treinamento é feito em tempo muito reduzido (da ordem de alguns milissegundos, sem considerar o tempo de leitura e digitalização da imagem).

Além do reconhecimento, o sistema informa a translação e a rotação da peça em relação ao sistema de coordenadas da câmera. Para tanto, uma nova rede foi utilizada, possuindo 40 neurônios de entrada, 156 neurônios na camada intermediária e 2 neurônios na camada de saída. Como saída desejada são fornecidos o cosseno e o seno do ângulo. No treinamento desta rede foram usadas imagens de 30° em 30° de cada peça e no modo de operação normal foi utilizada uma interpolação linear para auxiliar na precisão dos resultados. Para o treinamento foram necessárias cerca de 8.000 iterações em aproximadamente 15 minutos. A curva do erro é bastante similar àquela mostrada para a rede que reconhece as peças.

Durante o treinamento, a rede era treinada de modo a existir somente um neurônio ganhador da competição; já durante o reconhecimento os dois neurônios com maior valor de saída eram declarados ganhadores da competição. Entretanto, a somatória das saídas desses dois neurônios não pode ultrapassar 1, porque se somente um neurônio fosse declarado ganhador sua saída seria levada a 1 e os restantes seriam levados a 0. Os valores de saída de cada neurônio obtidos no final do treinamento são salvos junto com a base de conhecimento. Esses valores são aqueles obtidos com a aplicação dos padrões de treinamento, ou seja, se um neurônio foi treinado para reconhecer uma determinada peça rotacionada de 30°, por exemplo, quando da aplicação deste padrão na entrada da rede, esse neurônio terá seu maior valor de saída. Esses valores são utilizados, para dizer o quão próximo deste

valor a saída de um determinado padrão esta.

Para que se pudesse associar cada neurônio da camada Kohonen com um determinado ângulo, o treinamento desta camada deixou de ser não supervisionado e passou a ser determinístico, ou seja, padrões rotacionados de um determinado ângulo ativam um único neurônio pré-especificado.

Com esta estrutura de rede conseguiu-se uma taxa de acerto de aproximadamente 74%, e precisão de $\pm 10^\circ$.

A translação da peça é obtida no momento da leitura da imagem, por um processo que verifica a rápida variação do nível de cinza do fundo (fundo branco) sobre o qual a peça está situada e o nível de cinza da peça.

5 CONCLUSÃO

O sistema aqui desenvolvido e implementado, possui uma taxa de acerto considerada boa, tendo em vista que outros métodos possuem taxas de acerto entre 72% e 100% [9]. Existem algumas razões para que esta taxa de acerto não seja maior. Foi utilizada uma iluminação não simétrica, ou seja, usou-se duas lâmpadas fluorescentes de cada lado da estrutura usada para obter as imagens. Como o algoritmo de pré-processamento de imagem é bastante simples e baseia-se no nível de cinza de cada pixel da imagem, esta deficiência na iluminação tende a mudar os níveis de cinza de alguns pixels que representam uma determinada área da peça sendo vista, dependendo da rotação e translação da mesma.

Outro motivo é a semelhança de algumas peças usadas no processo de treinamento, algumas delas diferem basicamente de tamanho, de maneira sutil.

Possíveis melhoramentos podem ser conseguidos, utilizando uma iluminação melhor e um algoritmo para pré-processamento que consiga extrair mais características da peça. Uma possibilidade seria a utilização de redes com inibição

lateral. Esse tipo de rede pode representar padrões mais complexos. Estudos estão sendo feitos neste sentido.

BIBLIOGRAFIA

- [1] - Wasserman, Philip D.: *Neural computing: theory and practice*, Van Nostrand Reinhold, 1989.
- [2] - Freeman, James A. e Skapura, David M.: *Neural networks: algorithms, applications, and programming techniques*, Addison-Wesley Publishing Company, Inc, 1991.
- [3] - Doyhoff, Judith.: *Neural network architectures: an introduction*, Van Nostrand Reinhold, 1990.
- [4] - Lawrence, Jeannette J.: *Untangling neural nets*, Dr. Dobb's Journal, Abril 1990.
- [5] - King, Todd: *Using neural networks for pattern recognition*, Dr. Dobb's Journal, Janeiro 1989.
- [6] - Sepeda Filho, Idmilson H. e Stemmer, Marcelo R. : *Um Sistema de Reconhecimento de Peças Baseado em Redes Neurais*, aceito para publicação nos anais do 10º Congresso Brasileiro de Automática, a realizar-se em Setembro de 1994.
- [7] - Jones, William P. e Hoskins, Josiah: *Back-propagation: a generalized delta learning rule*, Byte, Outubro 1987.
- [8] - Treleaven, Philip; Pacheco, Marco e Vellasco, Marley: *VLSI Architecture for Neural Networks*, IEE MICRO, Dezembro 1989.
- [9] - Chien-nam, Huang; Chin-Choon, Lim e Ming, Liu C.: *Comparison of Image Processing Algorithms and Neural Network in Machine Vision Inspection*, Proceedings of the 14th Annual Conference on Computer and Industrial Engineering, Novembro 1992.
- [10] - Wolf, Thomas: *Neuronen im Computer*, Revista MC, Alemanha, Abril 1990.