

# DETECÇÃO DE VOLUME DE TRÁFEGO DE VEÍCULOS PROPORCIONADA POR VISÃO COMPUTACIONAL VIA REDES NEURAIS

F. F. S. Ribeiro, A. C. de C. Lima

*Departamento de Engenharia Elétrica - Escola Politécnica - UFBA,  
R. Aristides Novis 2, CEP 40210-630 - Salvador - Bahia - Brasil  
E-mails: ffsr@hotmail.com, acdcl@ufba.br*

**Resumo** - Neste trabalho são associadas as ferramentas de Redes Neurais e de Tratamento de Imagens, na forma de técnicas de detecção de bordas, com o objetivo de identificar corretamente o número de carros em imagens obtidas a partir de um mesmo ponto fixo. As redes neurais artificiais, por serem inspiradas no neurônio biológico, possuem características muito úteis no reconhecimento de padrões. A detecção de bordas reduz a imagem original a uma imagem que apresenta apenas as bordas dos objetos contidos nela. Visto que experimentos realizados com seres humanos e outros animais mostraram que bordas são as mais importantes pistas para interpretar imagens, tal associação de ferramentas busca facilitar ou possibilitar a detecção do número de carros em imagens, o que possibilitaria, por exemplo, a determinação do tempo de atuação de semáforos em resultado da comparação do fluxo de suas diversas vertentes de tráfego. Como resultado, demonstra-se que tal associação é bem sucedida e que, apesar de se ter obtido um erro absoluto médio de 0,8 carros no derradeiro experimento, podem ser obtidos resultados melhores à medida que se aumenta o número de imagens utilizadas no treinamento da rede neural.

**Abstract** - This work associates Neural Networks with the Image Processing technique of edge detection, having the objective of correctly identifying the number of cars in images obtained from a fixed place. Inspired in the biological neuron, the artificial neural network is a very useful tool for on pattern recognition. The edge detection reduces the original image to a image containing only the edges of the objects. Since experiences made with humans and others animals proved that edges are of fundamental importance for interpreting images, such association is proposed to facilitate or to make possible the detection of the number of cars in images so that the traffic light commutation time can be determined according to the tracks flow. The simulation results show that this association is successful and, despite obtaining a 0.8 absolute average error on the cars detection in the last experiment, this work demonstrates that better results can be obtained by increasing the number of images used in the neural network training.

## 1 Introdução

O fluxo de carros nas cidades de médio e grande porte cada vez mais se torna um problema crítico. Diversos são os fatores que colaboram para isto, entre eles, o crescente número de carros, a falta de planejamento e manutenção das vias de tráfego e semáforos sem **adaptabilidade** às variáveis necessidades do dia. Este último fator mencionado será alvo deste estudo.

No semáforo pode ser selecionado o tempo de atuação, cedendo uma fração do tempo de livre tráfego para cada uma das direções de fluxo que se cruzam. Porém, a escolha deste tempo de atuação não é trivial, pois as necessidades de fluxo não são sempre as mesmas. Mesmo se for determinado este tempo de atuação a partir da necessidade média, ou por qualquer outro método, um único tempo de atuação fica claramente bem aquém da solução ideal.

A solução proposta vem a ser um mecanismo automatizado que possa equivar a um observador humano que manualmente determinasse o tempo de atuação conforme observasse a necessidade do fluxo mudar. Tal mecanismo teria a necessária **adaptabilidade**. Buscando tal solução, propõe-se o uso da ferramenta *Redes Neurais* pela sua característica central: capacidade de adaptação na classificação de padrões (Haykin, 1994) e mesmo no estabelecimento de tais padrões.

O sensor do sistema em questão será uma câmera filmadora que enviará a imagem que seria vista por um observador humano que estivesse na posição do semáforo. Para modelar o cérebro humano, será usada uma *Rede Neural*. Através da câmera, imagens do

tráfego serão obtidas e processadas por tal rede com o objetivo de ter sucesso na detecção automática do número de carros contidos em cada imagem, possibilitando um futuro controle do tempo de atuação de um conjunto de semáforos.

Entretanto, a eficiência deste sistema depende muito dos dados de entrada. Um tratamento prévio dos dados é muitas vezes essencial. Dentro das ferramentas existentes de *Tratamento de Imagens*, rotinas de detecção de bordas (Gose, 1974) serão propostas como valorizadoras da informação pertinente à detecção do número de carros, facilitando o trabalho da *Rede Neural*.

Neste interesse, este trabalho tem por objetivos determinar:

- a) o número de carros em imagens obtidas a partir de um ponto fixo de filmagem.
- b) a validade ou não de um pré-processamento destas imagens na obtenção correta do número de carros.

## 2 Material e Métodos

### 2.1 Equipamentos e Programas

Os equipamentos utilizados foram:

- Câmera Sony HandyCam, Vídeo 8, Modelo CCD-TRV11 NTSC;
- Computador PC K6-300-II, 256 Mb Ram, HD 6Gb;
- Placa de Captura de Vídeo LG, chipset Cirrus Logic CL-GD5446TV, 4Mb RAM.

Os programas utilizados foram:

- Sistema Operacional *Windows 95 OSR2*;
- *Adobe Premiere 4.2* com driver **intel**;

- *AVI Constructor 2.3* trial version, por Michael Caracena;
- *IrfanView 32 bit version 2.83*, por Irfan Skiljan, estudante da Universidade de Tecnologia de Viena;
- *MATLAB basic toolbox version 5.1*;
- *MATLAB Neural Network toolbox version 2.04*;
- *MATLAB Image Processing toolbox version 2.0*.

## 2.2 Geração das imagens digitais

O primeiro passo se deu na escolha de um local adequado à filmagem. Da posse de um filme de aproximadamente dez minutos, a câmera foi conectada à placa de captura de vídeo com o objetivo de obter as imagens digitais equivalentes. Para tanto, três programas gráficos foram utilizados.

Inicialmente, utilizou-se o Adobe Premiere para gerar um arquivo AVI (clipe de vídeo) que consiste numa conversão do sinal emitido pela câmera para uma seqüência de imagens digitais agrupadas na forma de um vídeo, incorporando inclusive o som, embora no presente caso desnecessário. Utilizando-se resolução 160x120 pixels e taxa de captura de 15 fps (frames por segundo) por opções de captura, resulta em 15 imagens geradas a cada segundo de filme, cada imagem equivalendo a uma matriz de 160 colunas por 120 linhas. Visto que quanto menor a resolução mais leve se tornará o processamento das imagens, uma resolução maior não foi escolhida por esta se mostrar suficiente para posterior aplicação da técnica de detecção de bordas.

A seguir, o programa AVI Constructor foi utilizado para desmembrar o arquivo de filme digital AVI em uma série de arquivos de imagens digitais. Isto gerou 8402 imagens em arquivos BMP (Imagem de Bitmap) distintos e autonumerados.

Das fotos obtidas, foram selecionadas aquelas que não continham imagens parciais de carros para permitir uma definição correta do número de carros em cada foto, possibilitando um treinamento bem definido e uma verificação exata dos erros resultantes do reconhecimento de imagens não utilizadas no treinamento. Imagens com ônibus, caminhões ou motos foram retiradas também por dificultarem a definição exata de sua equivalência em número de carros. Em resultado, das 8402 imagens originais, restaram 3200 fotos que se mostraram mais que suficientes para a realização deste experimento.

O derradeiro passo na obtenção das imagens digitais é a utilização do programa IrfanView para converter, através de um único comando, todas as imagens que até então são coloridas para imagens em tons de cinza. Isto foi feito devido ao fato de que as cores são claramente irrelevantes no processo de decisão da quantidade de carros, até mesmo dificultando tal decisão por acrescentar informação dispensável.

## 2.3 Pré-processamento das imagens

A partir deste ponto, de posse dos 3200 arquivos de imagens BMP em preto e branco no formato 160x120 pixels, todo o processo fica a cargo do programa MATLAB e seus *toolboxes* de Redes Neurais e de Processamento de Imagens, que acrescentam as rotinas necessárias ao MATLAB para a obtenção dos resultados desejados.

No presente caso, as imagens foram divididas em lotes de treinamento e lotes de verificação, sendo que os lotes de treinamento na quantidade de três e na forma de um primeiro lote com 400 imagens, um segundo com 200 imagens e um terceiro com 200 imagens. Já os lotes de verificação foram divididos na quantidade de seis lotes de 400 imagens cada. Esta divisão em vários lotes visa possibilitar a manipulação de tamanho volume de informações em vista da grande quantidade de memória RAM que este procedimento de treinamento necessitará.

As imagens foram então pré-processadas pelo MATLAB, que possui uma rotina de detecção de bordas implementada em seu *toolbox* de Tratamento de Imagens denominada **edge** (Gose, 1971; Gose, 1974). Esta rotina possui basicamente quatro métodos matemáticos para se detectar as bordas de uma imagem: *sobel*, *prewitt*, *roberts* e *log*. O método *log* foi apontado como o mais apropriado através da realização de um teste visual que pode ser observado em parte na Fig. 1.

Além disso, a área fora da pista de passagem de carros, ou seja, a área da calçada em diante, foi excluída no momento da entrada de dados na rede neural, resultando numa redução por imagem de anteriores 19200 pixels (160x120) para 12585 pixels, uma eliminação de 34,45% de informação claramente sem utilidade.

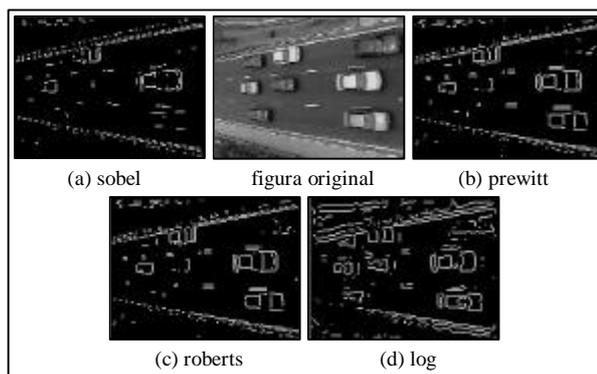


Figura 1 - Teste visual dos métodos de detecção de bordas.

No MATLAB, este pré-processamento definiu uma matriz de entrada com cada imagem na forma de uma de suas colunas, ou seja, uma matriz *Inp* (input) de 12585x400 pixels, visto que foram utilizadas 400 fotos para treinamento inicial. Também ficou definida a matriz de resposta correta do número de carros, ou seja, uma matriz *Tar* (target) de 1x400 elementos, que contém o número de carros de forma direta. Isto se dá por causa do fato de que a camada final da rede neural

consiste de um neurônio do tipo *log-sigmoidl*, cuja saída varia de 0 a 1. Devido ao fato de que o número de carros nas imagens variou de 0 a 8, utilizou-se a seguinte forma de adequar esta variação:

$$\text{Tar} = \text{Tar} / 10 + 0,1 \quad (1)$$

o que fez a faixa de respostas corretas variar de 0,1 a 0,9. De forma similar, foram definidos todos os nove lotes de figuras utilizados seja para treinamento como para verificação.

#### 2.4 Definição da rede neural

A definição da rede neural mais apropriada a determinado problema é essencialmente empírica (Haykin, 1994; Faggin, F., 1991). Na escolha do número de camadas, definiu-se duas como estritamente necessárias, sendo a primeira para receber diretamente cada pixel da imagem e a última para totalizar e emitir o resultado (o número de carros) na forma de um valor único e portanto possuindo um único neurônio. Uma camada adicional, intermediária, foi utilizada por se entender que esta traria a possibilidade de um reprocessamento do resultado da primeira camada, aumentando as chances de obtenção de uma regra geral que modele o reconhecimento de carros. Além disso, esta camada intermediária pesa muito pouco no tempo de processamento, pois ela só gera conexões com um pequeno número de neurônios, os da primeira camada, enquanto que na primeira camada cada neurônio tem uma conexão com cada pixel da imagem.

A quantidade de neurônios foi definida em função da quantidade de dados utilizada no treinamento. Embora para determinar tal quantidade não haja uma regra simples, pode-se observar que a necessidade de mais neurônios cresce à medida que se aumenta o universo de treinamento, pois com este aumento a regra geral a ser obtida tende a ser mais complexa e detalhada. Assim, em um treinamento realizado com 50 imagens, 8 neurônios podem ser suficientes para as duas primeiras camadas, enquanto que treinando com 200 imagens a necessidade de neurônios pode subir para 12. Um outro aspecto é o de que a regra geral obtida no primeiro treinamento não vai ser tão geral quanto a obtida com as 200 imagens no segundo exemplo. Apesar disso e de acordo com as necessidades de margem de erro, ambas as redes podem ser satisfatórias para solucionar o problema.

Tendo sido definidos o número de camadas e a quantidade neurônios em cada camada (20x20x1), foi escolhida a função de transferência *log-sigmoid* para os neurônios de todas as camadas. Isto foi feito por se desejar que a saída dos neurônios fosse limitada a uma pequena faixa, diminuindo a possibilidade de resultados muito espúrios, não atingindo também valores negativos por não serem resultados úteis para o problema.

No MATLAB, toda esta definição é realizada pela rotina **initff**, que define valores iniciais randômicos para os ganhos de cada conexão de cada neurônio de

uma rede *feed-forward*, ou seja, de redes que não possuem realimentação de dados.

#### 2.5 Treinamento da rede neural

Para realizar o treinamento da rede neural *feed-forward* foi escolhido o método *backpropagation*. Este método é implementado no MATLAB através de algumas rotinas cuja diferença está na velocidade de aprendizado dos algoritmos. Foi utilizado o método **trainbpx** por possuir tanto aprendizado adaptativo como *momentum*. A característica *momentum* diminui a sensibilidade do método a pequenos detalhes na superfície de erro, ajudando a rede a evitar mínimos locais que poderiam impedi-la de encontrar um ponto de erro satisfatório. O tempo de treinamento também pode ser diminuído por se utilizar aprendizado adaptativo, pois este tenta manter a taxa de aprendizado tão grande quanto possível, ao mesmo tempo que mantém o treinamento estável.

O treinamento da rede ocorre em etapas denominadas épocas. Estas épocas consistem em se aplicar todos os dados de entrada de treinamento na rede, verificar o erro de cada um desses dados e ajustar a rede para diminuir o erro médio. Isto evita que, ao ajustar a rede para uma entrada, se aumente o erro das outras. Assim, um vetor parâmetro de fundamental importância na rotina **trainbpx** é chamado de TP (training parameters), pois define os parâmetros que ajustam as necessidades do treinamento. Dois dos parâmetros úteis de TP definem o número máximo de épocas de treinamento e o erro mínimo desejado. O treinamento perdurará enquanto nenhuma destas duas condições for atingida. Também existem parâmetros em TP para ajustar as características de *momentum* e de aprendizado adaptativo.

#### 2.6 Simulação da rede neural

Após a obtenção da rede neural treinada, vem a necessidade de verificar se os resultados obtidos foram satisfatórios, ou seja, se a rede neural conseguiu definir uma regra suficientemente geral para reconhecer corretamente a quantidade de fotos não só das imagens utilizadas para treinamento mas também das imagens que lhe são até então desconhecidas. Este procedimento é chamado de simulação e é realizado com base na rotina **simuff** do MATLAB. Além de realizada a simulação, foram determinados os erros médios absolutos tanto do lote de treinamento como dos lotes de verificação, determinando também a distribuição do erro destes por faixas de variação.

#### 2.7 Estrutura geral do experimento

Tendo sido as imagens obtidas, pré-processadas, definidas no espaço de trabalho do MATLAB e salvas separadamente em nove arquivos definidos como lotes, foi realizada a inicialização da rede neural seguida do primeiro procedimento de treinamento com o lote denominado p1, possuidor de 400 imagens.

Após este processamento, obteve-se a rede neural treinada e pronta para ser submetida à simulação separadamente com os lotes denominados p4, p5, p6, p7, p8 e p9, cada um de 400 fotos.

Igual procedimento de definição e treinamento foi realizado com as imagens não pré-processadas pela técnica de detecção de bordas, para determinação da validade de se utilizar tal técnica. Neste caso, não se chegou a realizar nenhum procedimento de simulação.

Mantendo-se a linha de trabalho com detecção de bordas e partindo-se por ponto inicial do treinamento resultante do lote p1, acrescentou-se o lote p2, possuidor de 200 imagens, com o objetivo de obter uma rede neural com um treinamento mais aprimorado. Novamente, obteve-se a rede neural treinada e pronta para ser submetida à simulação separadamente com os lotes de p4 a p9.

Por fim, acrescentou-se o lote p3, possuidor de 200 imagens, com o objetivo de obter um aprimoramento ainda maior, determinado ou não uma tendência. Assim, obteve-se a rede neural treinada e pronta para ser submetida à simulação separadamente com os lotes de p4 a p9.

### 3 Resultados e Discussão

#### 3.1 Validação do uso de detecção de bordas

O treinamento do lote p1 de 400 imagens, sem utilização de técnicas de detecção de bordas foi realizado com limite de 1000 épocas ou até o momento em que o erro quadrático médio (Sum-Squared Error - SSE) atingisse o valor de 0,01 equivalente a um décimo de carro devido à anteriormente mencionada adequação das faixas de resposta (Equação 1).

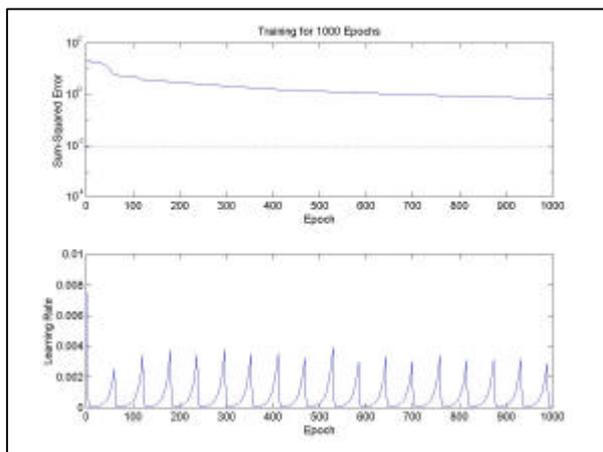


Figura 2 - Treinamento do lote p1 sem detecção de bordas.

Este treinamento levou cerca de seis horas de processamento e iniciou com  $SSE = 21,7737$  e, ao término das 1000 épocas sem atingir o erro desejado, obteve  $SSE = 0,6771$ , como pode ser visto na Fig. 2.

Cabe aqui uma explicação adicional sobre estes gráficos gerados pelo MATLAB, visto que seu tamanho reduzido dificulta a leitura no formato aqui

apresentado. Estes gráficos apresentam, primeiro, a variação do erro quadrático médio e, segundo, a variação da velocidade com que o erro foi aprimorado, ambos em função do tempo, no caso, do número de épocas de treinamento. O tracejado mais claro e retilíneo (e a princípio vermelho) do primeiro indica o erro que se deseja atingir ( $10^{-2}$ ).

No segundo gráfico, podem ser observados vários picos da velocidade de aprendizado (learning rate) que mostram momentos em que o passo de treinamento teve de ser diminuído por não haver mais progresso na diminuição do erro. Quando o erro volta a diminuir, este passo volta a crescer para acelerar o processo.

Em seqüência, treinamento similar foi realizado com a única diferença consistindo da utilização de técnicas de detecção de bordas. Este treinamento levou cerca de três horas de processamento e iniciou com  $SSE = 107,1609$  e, ao término de 372 épocas atingiu o erro desejado, obtendo-se  $SSE = 0,0097$ , como pode ser visto na Fig. 3.

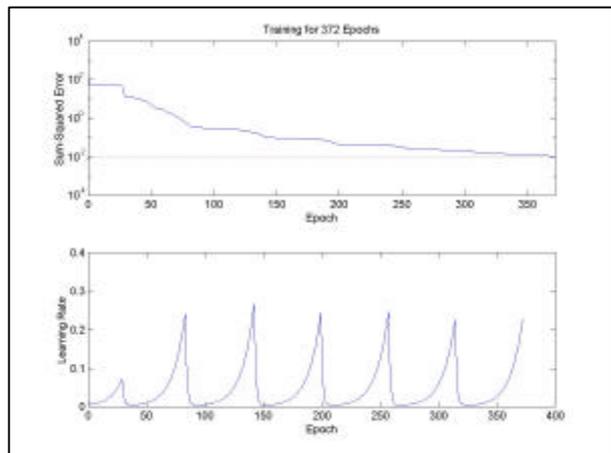


Figura 3 - Treinamento do lote p1 com detecção de bordas.

Comparando-se tais resultados, fica clara a superioridade do método que utilizou detecção de bordas. Testes adicionais mostraram que mesmo realizando o treinamento com as imagens na sua forma original com 4000 épocas de treinamento, o que demorou cerca de vinte horas de treinamento, não se pode reduzir o valor de SSE de 0,5. Este é um erro muito grande, da ordem de 5 carros por imagem!

Também fica clara a superioridade da velocidade de convergência do treinamento que utilizou detecção de bordas. Embora ambos os métodos mostrem tendência de queda do erro, a lentidão desta queda no experimento sem detecção de bordas demonstra que mesmo em 10000 épocas de treinamento é improvável que se consiga o resultado desejado, além de tal tentativa provavelmente vir a demorar dias!

Devido ao fato de que o método de treinamento utilizado possui técnicas de aprendizado adaptativo, podem ser vistos os valores que a taxa de aprendizado (Learning Rate) assumiu durante ambos os processos. Tal taxa sobe à medida que percebe que o erro continua

a diminuir e cai no momento em que verifica que o erro aumenta ou se mantém fixo. Comparando-se Os valores médios desta taxa visualizados nas Figuras 2 e 3, vê-se mais uma vez a superior velocidade de convergência do experimento com detecção de bordas.

### 3.2 Desempenho da rede neural

Mantendo a utilização de detecção de bordas, foram realizados os treinamentos adicionais descritos no tópico 2.7, a saber, um treinamento com 600 imagens (lotes p1 e p2) apresentado na Fig. 4 e outro com 800 imagens (lotes p1, p2 e p3) apresentado na Fig. 5. Ambos partiram do ponto inicial do experimento imediatamente anterior, ou seja, sem reinicializar a rede neural. Assim, obteve-se um total de três redes neurais diferentes, embora similares e possuidoras da mesma estrutura básica.

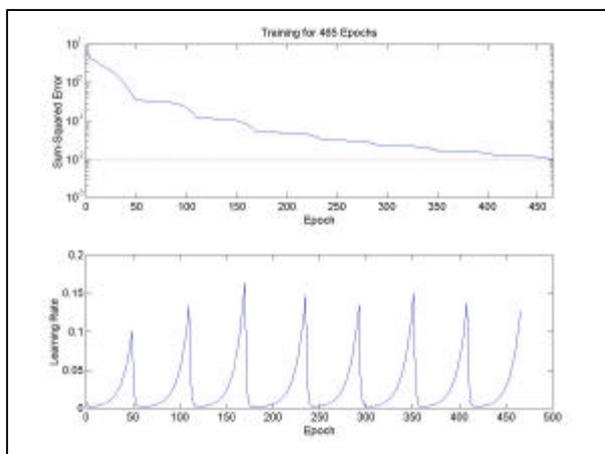


Figura 4 - Treinamento dos lotes p1+p2 com detecção de bordas.

O treinamento com 600 imagens (p1+p2) levou cerca de cinco horas de processamento e iniciou com SSE = 6,3689 e, ao término de 465 épocas, atingiu o erro desejado, obtendo SSE = 0,0100, como pode ser visto na Figura 5.3. O erro inicial aqui obtido é bem menor do que no experimento da Figura 5.2, tendo em vista que se utilizou por ponto inicial o resultado do treinamento do lote p1. Apesar disso, o número de épocas necessário para que se atingisse o erro desejado aumentou, pois o universo de imagens a ser modelado pela rede neural na forma de uma regra geral se tornou mais complexo. Também podem ser comparados os tempos de duração, levando em conta não a demora total mas o tempo gasto por época. Este deveria aumentar pois cada época teve de processar 200 figuras a mais e, de fato, aumentou.

Já o treinamento com 800 imagens (p1+p2+p3) demorou cerca de nove horas e meia de processamento, iniciando seu processo com SSE = 1,6093 e, ao término de 741 épocas, atingiu o erro desejado, obtendo SSE = 0,0100, como pode ser visto na Figura 5.3. O erro inicial aqui obtido é ainda menor do que no experimento da Figura 5.3, tendo novamente em vista

que se utilizou por ponto inicial o resultado do treinamento do lote p2, sendo novidade para a rede neural apenas um quarto das fotos apresentadas para treinamento. Mais uma vez, o número de épocas necessário para que se atingisse o erro desejado aumentou, bem como o tempo gasto por época de treinamento, por causa das mesmas razões mencionadas anteriormente quando se comparou o experimento de treinamento com os lotes “p1 e p2” e com apenas “p1”.

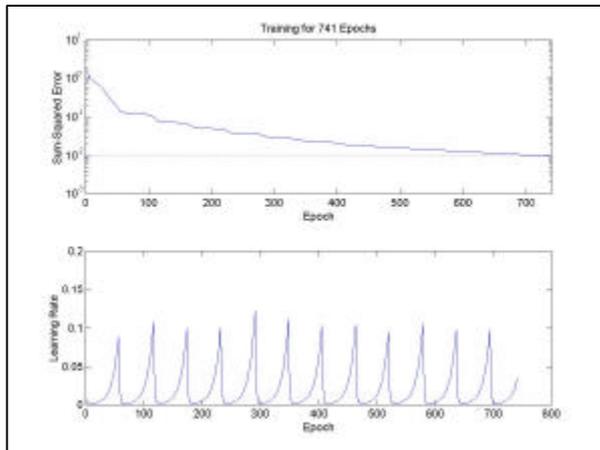


Figura 5 - Treinamento dos lotes p1+p2+p3 com detecção de bordas.

Para comparar os resultados obtidos com os três treinamentos realizados com o auxílio de detecção de bordas, foram realizadas as simulações com os seis lotes de verificação, cujos erros resultantes em **número de carros** por imagem podem ser vistos nas tabelas a seguir. Tanto o desempenho geral como de cada lote de verificação demonstram a clara tendência de melhora do desempenho da rede neural à medida que o treinamento é reforçado com um maior número de imagens. Portanto para obter resultados melhores, a depender da necessidade de reconhecimentos mais aprimorados, basta tornar mais numerosos os lotes de treinamento. O desempenho final obtido de um erro absoluto médio de 0,8 carros por imagem não conhecida já é suficiente para atingir o objetivo de comparar fluxos de duas ou mais vertentes de um semáforo.

Em tempo de execução, imagens geradas a intervalos relativamente longos podem dar a noção suficiente do fluxo real em cada vertente. Comparando o resultado visual da aplicação de detecção de bordas no experimento aqui desenvolvido com o resultado visual obtido na explanação teórica, pode-se ver que o resultado é espantoso tendo em vista o relativo baixo nível de bordas geradas pela rotina contida no MATLAB. De posse de rotinas superiores ou de maior resolução de imagens, este resultado certamente seria ainda melhor.

As tabelas apresentam os lotes e seus respectivos erros médios (em números de carros) após entrada na rede neural. Além disso, foram apresentadas faixas para ilustrar a distribuição do erro dentro dos lotes. A

primeira faixa mostra a quantidade de figuras que apresentaram erro de até 0,25 (um quarto de carro), a segunda faixa mostra a quantidade de figuras que apresentaram erro de 0,25 até 0,5 carros, e assim por diante. Ao final, uma linha totaliza os lotes de verificação e fornece o erro médio absoluto geral.

Lote Simulado	Erro Médio Absoluto	Erro absoluto distribuído em faixas						
		Até 0,25	≥0,25 a 0,5	≥0,5 a 1,0	≥1,0 a 2,0	≥2,0 a 3,0	≥3,0 a 4,0	≥ 4
p1	0,0492	-	-	-	-	-	-	-
p4	2,2503	17	18	37	105	149	74	0
p5	1,4315	40	43	78	175	56	8	0
p6	1,2476	58	53	100	156	31	2	0
p7	0,9820	116	65	118	83	12	4	2
p8	1,0009	62	66	137	118	16	1	0
p9	0,8189	126	85	117	57	14	1	0
<b>p4 a p9</b>	<b>1,2885</b>	419	330	587	694	278	90	2

**Tabela 1** - Desempenho da rede neural após treinamento com lote p1.

Lote Simulado	Erro Médio Absoluto	Erro absoluto distribuído em faixas						
		Até 0,25	≥0,25 a 0,5	≥0,5 a 1,0	≥1,0 a 2,0	≥2,0 a 3,0	≥3,0 a 4,0	≥ 4
p1+p2	0,0407	-	-	-	-	-	-	-
p4	1,3358	48	36	94	175	44	2	1
p5	1,0237	69	77	113	125	15	1	0
p6	0,9208	95	72	117	102	12	2	0
p7	0,8815	130	66	124	67	10	2	1
p8	0,8915	60	81	146	106	7	0	0
p9	0,7158	133	91	125	44	7	0	0
<b>p4 a p9</b>	<b>0,9615</b>	535	423	719	619	95	7	2

**Tabela 2** - Desempenho da rede neural após treinamento com lote p1+p2.

Lote Simulado	Erro Médio Absoluto	Erro absoluto distribuído em faixas						
		Até 0,25	≥0,25 a 0,5	≥0,5 a 1,0	≥1,0 a 2,0	≥2,0 a 3,0	≥3,0 a 4,0	≥ 4
p1+p2+p3	0,0353	-	-	-	-	-	-	-
p4	0,9347	109	86	111	78	13	2	1
p5	0,6784	101	125	134	35	4	1	0
p6	0,7648	99	88	145	61	7	0	0
p7	0,8674	122	79	120	66	10	3	0
p8	0,8453	71	90	144	89	6	0	0
p9	0,7096	133	86	120	57	4	0	0
<b>p4 a p9</b>	<b>0,8000</b>	635	554	774	386	44	6	1

**Tabela 3** - Desempenho da rede neural após treinamento com lote p1+p2+p3.

#### 4 Conclusão e Perspectivas Futuras

Com a associação das ferramentas de Tratamento de Imagem, na forma de detecção de bordas, e de Redes

Neurais, pode ser obtido um resultado que não seria possível somente com uma destas ferramentas. A comparação entre os resultados do treinamento de uma rede neural com fotos não pré-processadas e uma outra com fotos pré-processadas demonstrou no mínimo uma superioridade no desempenho desta última, superioridade esta que leva a uma economia de tempo de treinamento. Além disso, sequer houve sucesso comprovado da rede neural sem pré-processamento, sucesso este que talvez possa ser obtido com treinamentos excessivamente longos, com duração da ordem de dias.

Apesar de não se ter obtido um resultado com erro cem por cento inferior a um carro, ficou demonstrado que o erro relacionado ao reconhecimento correto do número de carros em uma imagem obtida a partir de um mesmo ponto fixo depende de quão bem o lote de imagens utilizado no treinamento represente a variação possível de todo o universo de imagens. Com o aumento do número de imagens utilizadas no treinamento, aumentam as chances de se aproximar de um lote cuja representatividade seja boa.

Estudos futuros podem ser realizados no sentido de comparar a eficiência de diferentes arquiteturas de redes neurais, bem como de diferentes rotinas de treinamento. Estudos a serem realizados com treinamentos envolvendo um número ainda maior de imagens e/ou de maior resolução devem experimentar outras ferramentas que utilizem variáveis inteiras ou reais de precisão simples. Por exemplo, no experimento aqui realizado, a matriz de entrada de dados só continha valores zero ou um, mas tiveram de ser definidas como reais de dupla precisão pois o MATLAB não multiplica variáveis inteiras.

Além disso, as rotinas de detecção de bordas possuem parâmetros que regulam sua sensibilidade automaticamente ou manualmente. Portanto, estudos sobre a influência desta sensibilidade no desempenho do sistema podem levar a uma mais rápida convergência no treinamento da rede neural.

#### Referências Bibliográficas

- Faggin, F., 1991. "VLSI implementation of neural networks," Tutorial Notes. *International Joint Conference on Neural Networks*, Seattle, WA.
- Gose, E. E., J. W. Bacus, and L. Ackerman, 1971. "A comparison of some computer-measured and human-measured pattern recognition properties". *Journal of Cybernetics*, 1.
- Gose, E. E., S. Werner, and R. G. Bickford, 1974. "Computadorized EEG spike detection". *Proceedings of the San Diego Biomedical Symposium*, February.
- Haykin, S., 1994. *Neural Networks: A Comprehensive Foundation*. New York: Macmillan Publishing.