

OCR

Optical Character Recognition

Computação Adaptativa (MEI)
CNSD (MIEB)

Alexandre Sousa (MIEB) - 2004107017

Marco Simões (MEI) - 2006125287

Sérgio Santos (MEI) - 2006125508

Introdução

O objectivo deste Trabalho Prático consiste no desenvolvimento de um programa de OCR: reconhecimento óptico de caracteres. A abordagem desejada passa pela aplicação dos conhecimentos teóricos adquiridos, relacionados com o desenvolvimento de redes neuronais, em software para a plataforma *MATLAB* (com a *Neural Networks Toolbox*).

Uma rede neuronal será responsável por um sistema de memória associativa para “melhorar” as entradas fornecidas e outra rede neuronal irá realizar a classificação em si dos caracteres. O software responsável pela visualização e inserção de caracteres é fornecido como ponto de partida, e a toolbox de redes neuronais da plataforma *MATLAB* auxiliará o desenvolvimento, treino e simulação das mesmas.

Desenvolvimento

Tendo em conta as funções de *MATLAB* já fornecidas à partida, desenvolvemos os seguintes módulos:

associativeMemory.m

Calcula os pesos com resultado óptimo para a memória associativa do algoritmo, através dos casos de treino e o respectivo output perfeito. Utiliza dois métodos: pseudo inversa e da matriz transposta.

Calcula também o erro obtido em relação ao resultado óptimo e ao esperado, através da soma dos quadrados dos erros.

trainNetwork.m

Cria uma rede neuronal utilizando a *Neural Networks Toolbox*, segundo os parâmetros configurados como argumento. De seguida, insere pesos aleatórios e treina-a segundo os casos de treinos fornecidos também como argumento, que vêm sempre com uma ordem específica, para facilitar a geração do resultado esperado.

classify.m

Classifica os casos de teste fornecidos utilizando a rede neuronal também fornecida como parâmetro. Após a classificação, ajusta o resultado ao formato necessário pela função *ocr_fun.m* fornecida inicialmente.

TP2.m

Esta função recorre as funções descritas acima para executar a aplicação como um todo: memória associativa + classificador. Permite efectuar testes utilizando várias configurações de redes neuronais especificadas inicialmente, com ou sem memória associativa, e com diferentes casos de treino e teste, e no fim apresentar os resultados obtidos.

Execução

Para executar/testar a aplicação desenvolvida existem duas formas:

Executar directamente a função **TP2.m**, que irá iniciar o teste de várias redes neuronais especificadas e apresentar os resultados do seu desempenho.

Executar a função **mpaper.m** já fornecida, que permite desenhar caracteres e posteriormente classificá-los. Foram feitas as alterações necessárias para que a função possa recorrer aos algoritmos desenvolvidos para classificar os caracteres desenhados, e a configuração que se encontra por defeito é a que obteve melhor performance nos resultados que obtivemos.

Resultados

Realizámos um levantamento extensivo de resultados para diversas configurações possíveis.

Os parâmetros que variámos foram:

- Número de casos de treino: 250 e 900;
- Casos de testes normais ou com caracteres defeituosos: PTest e PDefect;
- Utilização ou não de memória associativa;
- Método para o cálculo dos pesos da memória associativa: PINV (pseudo inversa) e transposed (matriz transposta);
- Algoritmo da rede neuronal:
 - newp + leanrp
 - newff + learnh
 - newff + learnhd
 - newff + learngd
- Função de transferência:
 - hardlim
 - purelin
 - logsig

Outros parâmetros de configuração definidos, mas estáticos foram:

```
net.performParam.ratio = 0.5;
```

```
net.trainParam.epochs = 1000;  
net.trainParam.show = 35;  
net.trainParam.goal = 1e-4;
```

As tabelas contendo todos os resultados obtidos encontram-se em **Anexo**.

Conclusões

Os resultados obtidos durante os nossos testes permitiram-nos chegar a várias conclusões.

Como exemplo, apresentamos um gráfico com várias configurações de teste distintas, que incluem a configuração que apresentou os melhores resultados (**Figura 1**).

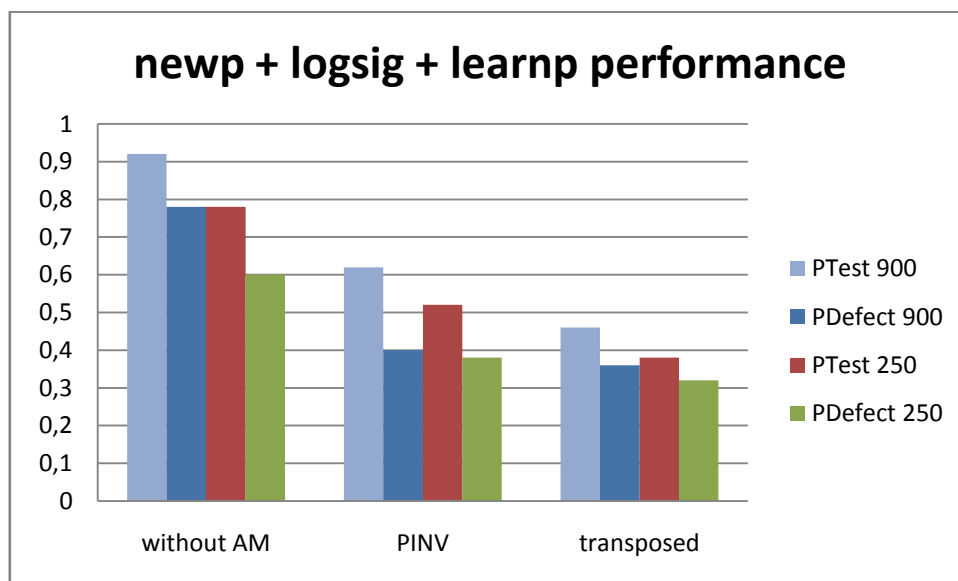


Figura 1 - newp + logsig + learnp performance

Casos de treino

É possível observar desde já a influência do treino com um maior número de casos (250 contra 900 casos de treino). Quando a rede é testada com casos normais (PTest) existe uma diferença de mais de 10% na performance (número de casos correctos identificados). Com casos de caracteres mais defeituosos e difíceis de classificar (PDefect), a diferença é menor, mas ainda visível. Numa experiência adicional, não adicionada à tabela de resultados, testámos o sistema com 500 casos de treino e verificámos uma performance muito semelhante à de 900 casos de treino. Este facto indica que o ganho de performance com o número de casos de treino é cada vez menor e que deverá tender para zero.

Estrutura das redes neuronais

Em termos de estrutura, na aplicação desenvolvida a utilização da memória associativa em conjunto com o classificador resultou sempre em piores resultados que a utilização apenas do classificador. No gráfico da [Figura 1](#), por exemplo, nota-se claramente a perda de performance do primeiro conjunto de valores para os dois restantes. Entre os diferentes métodos de cálculo dos pesos na memória associativa, como seria de esperar, o método da pseudo inversa apresenta melhores resultados que o da matriz transposta.

Em relação à função de activação, o gráfico da [Figura 2](#) apresenta as diferentes performances para 900 e 250 casos de treino, para três configurações:

- hardlim: newp + leanrp
- logsig: newp + learnp
- purelin: newff + learnh

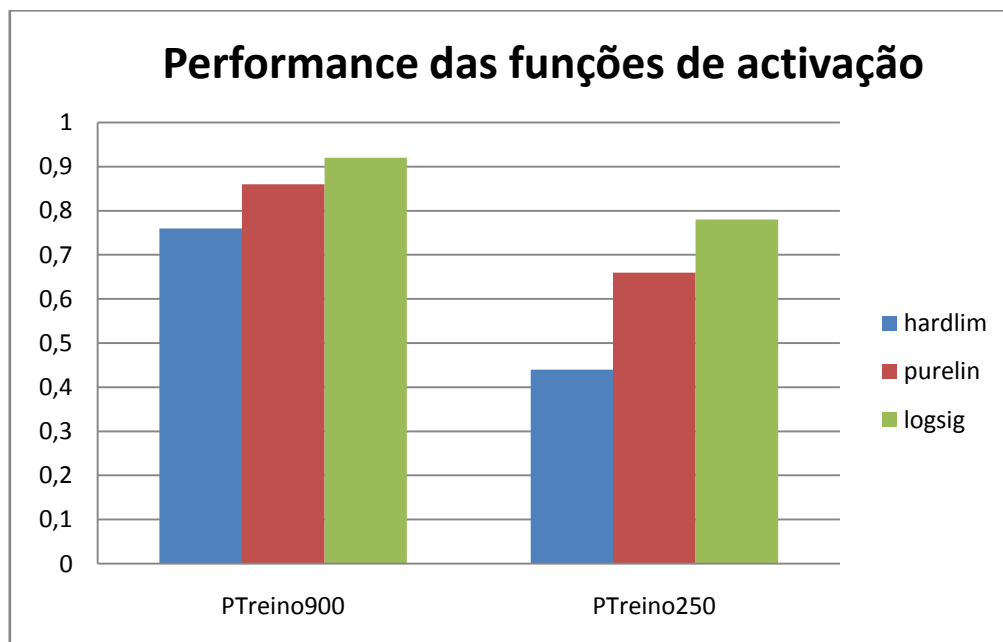


Figura 2 – Performance das funções de activação

É possível observar a melhor performance da função de activação logaritmica utilização a função sigmóide (*logsig*).

Em relação aos algoritmos de aprendizagem, os resultados obtidos encontram-se no gráfico da [Figura 3](#), para 900 casos de treino e uma função de activação *logsig*.

Os melhores resultados pertencem ao algoritmo de aprendizagem do perceptrão (*learnp*), que apresenta resultados marginalmente superiores aos restantes para casos de teste normais (PTest) e com diferenças mais significativas (até 10%) com casos mais defeituosos (PDefect). É também o algoritmo mais rápido em tempo de execução. A regra de Hebb apresenta uma performance ligeiramente inferior a esta.

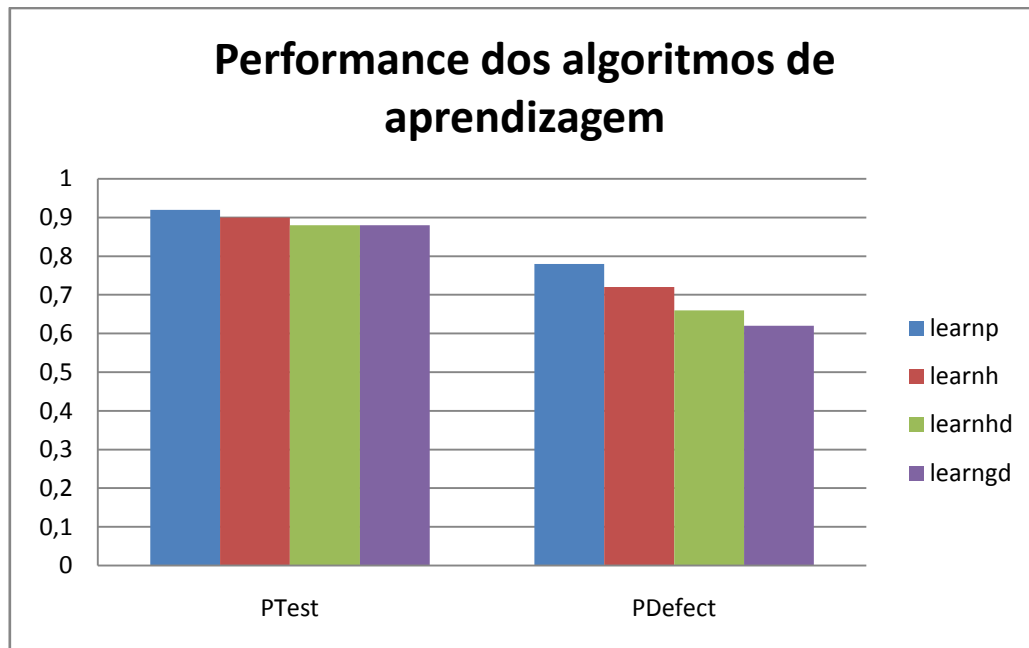


Figura 3 - Performance dos algoritmos de aprendizagem

Resultados

Em suma, o melhor resultado obtido nos testes efectuados foi de 92% de casos de teste correctamente identificados, 8% incorrectamente identificados e 0% não identificados. A configuração que obteve esses resultados foi:

- 900 casos de treino;
- Rede neuronal *newp* + *learnp*;
- Função de activação *logsig*;
- Função de performance *sse*: soma dos erros quadrados;
- 100 casos de teste normais.

Acreditamos que esta já é uma boa performance para uma aplicação deste tipo, facto que observámos empiricamente através da ferramenta *mpaper* e posterior correcta classificação dos caracteres desenhados.

Em termos de capacidade de generalização da rede, para caracteres mais defeituosos a performance diminui como seria de esperar. De qualquer forma atingiu 78% de caracteres correctamente identificados, 22% incorrectamente identificados e 0% não identificados, o que são valores bastante razoáveis tendo em conta a dificuldade da tarefa.

Anexo - Resultados

900 casos de treino

Test Case	Associative Memory				Classify					Result			without AM	PINV	transposed
	Train Case	Method	Transfer	Perform	Train Case	Net	Transfer	Learning	Perform	Correct	Incorrect	Undetected			
PTest					PTreino900	newp	hardlim	learnp	sse	76%	4%	20%	0,76		
PDefect					PTreino900	newp	hardlim	learnp	sse	62%	10%	28%	0,62		
PTest	PTreino900	pinv	purelin	sse	PerfectArial	newp	hardlim	learnp	sse	20%	26%	54%		0,2	
PDefect	PTreino900	pinv	purelin	sse	PerfectArial	newp	hardlim	learnp	sse	16%	26%	58%		0,16	
PTest	PTreino900	transposed	purelin	sse	AMO	newp	hardlim	learnp	sse	48%	4%	48%			0,48
PDefect	PTreino900	transposed	purelin	sse	AMO	newp	hardlim	learnp	sse	42%	6%	52%			0,42
PTest					PTreino900	newp	logsig	learnp	sse	92%	8%	0%	0,92		
PDefect					PTreino900	newp	logsig	learnp	sse	78%	22%	0%	0,78		
PTest	PTreino900	pinv	purelin	sse	PerfectArial	newp	logsig	learnp	sse	62%	38%	0%		0,62	
PDefect	PTreino900	pinv	purelin	sse	PerfectArial	newp	logsig	learnp	sse	40%	60%	0%		0,4	
PTest	PTreino900	transposed	purelin	sse	AMO	newp	logsig	learnp	sse	46%	18%	36%			0,46
PDefect	PTreino900	transposed	purelin	sse	AMO	newp	logsig	learnp	sse	36%	16%	48%			0,36
PTest					PTreino900	newff	purelim	learnh	sse	86%	14%	0%	0,86		
PDefect					PTreino900	newff	purelim	learnh	sse	64%	36%	0%	0,64		
PTest	PTreino900	pinv	purelin	sse	PerfectArial	newff	purelim	learnh	sse	76%	24%	0%		0,76	
PDefect	PTreino900	pinv	purelin	sse	PerfectArial	newff	purelim	learnh	sse	60%	40%	0%		0,6	
PTest	PTreino900	transposed	purelin	sse	AMO	newff	purelim	learnh	sse	74%	26%	0%			0,74
PDefect	PTreino900	transposed	purelin	sse	AMO	newff	purelim	learnh	sse	70%	30%	0%			0,7
PTest					PTreino900	newff	logsig	learnh	sse	90%	10%	0%	0,9		
PDefect					PTreino900	newff	logsig	learnh	sse	72%	28%	0%	0,72		
PTest	PTreino900	pinv	purelin	sse	PerfectArial	newff	logsig	learnh	sse	70%	30%	0%		0,7	

PDefect	PTreino900	pinv	purelin	sse	PerfectArial	newff	logsig	learnh	sse	46%	54%	0%	0,46	
PTest	PTreino900	transposed	purelin	sse	AMO	newff	logsig	learnh	sse	0%	0%	100%		0
PDefect	PTreino900	transposed	purelin	sse	AMO	newff	logsig	learnh	sse	0%	0%	100%		0
PTest					PTreino900	newff	purelim	learnhd	sse	86%	14%	0%	0,86	
PDefect					PTreino900	newff	purelim	learnhd	sse	64%	36%	0%	0,64	
PTest	PTreino900	pinv	purelin	sse	PerfectArial	newff	purelim	learnhd	sse	80%	20%	0%	0,8	
PDefect	PTreino900	pinv	purelin	sse	PerfectArial	newff	purelim	learnhd	sse	60%	40%	0%	0,6	
PTest	PTreino900	transposed	purelin	sse	AMO	newff	purelim	learnhd	sse	74%	26%	0%		0,74
PDefect	PTreino900	transposed	purelin	sse	AMO	newff	purelim	learnhd	sse	70%	30%	0%		0,7
PTest					PTreino900	newff	logsig	learnhd	sse	88%	10%	2%	0,88	
PDefect					PTreino900	newff	logsig	learnhd	sse	66%	32%	2%	0,66	
PTest	PTreino900	pinv	purelin	sse	PerfectArial	newff	logsig	learnhd	sse	66%	34%	0%	0,66	
PDefect	PTreino900	pinv	purelin	sse	PerfectArial	newff	logsig	learnhd	sse	42%	58%	0%	0,42	
PTest	PTreino900	transposed	purelin	sse	AMO	newff	logsig	learnhd	sse	0%	0%	100%		0
PDefect	PTreino900	transposed	purelin	sse	AMO	newff	logsig	learnhd	sse	0%	0%	100%		0
PTest					PTreino900	newff	purelim	learnhd	sse	86%	14%	0%	0,86	
PDefect					PTreino900	newff	purelim	learnhd	sse	64%	36%	0%	0,64	
PTest	PTreino900	pinv	purelin	sse	PerfectArial	newff	purelim	learnhd	sse	78%	22%	0%	0,78	
PDefect	PTreino900	pinv	purelin	sse	PerfectArial	newff	purelim	learnhd	sse	60%	40%	0%	0,6	
PTest	PTreino900	transposed	purelin	sse	AMO	newff	purelim	learnhd	sse	74%	26%	0%		0,74
PDefect	PTreino900	transposed	purelin	sse	AMO	newff	purelim	learnhd	sse	70%	30%	0%		0,7
PTest					PTreino900	newff	logsig	learnhd	sse	88%	12%	0%	0,88	
PDefect					PTreino900	newff	logsig	learnhd	sse	62%	38%	0%	0,62	
PTest	PTreino900	pinv	purelin	sse	PerfectArial	newff	logsig	learnhd	sse	54%	46%	0%	0,54	
PDefect	PTreino900	pinv	purelin	sse	PerfectArial	newff	logsig	learnhd	sse	44%	56%	0%	0,44	
PTest	PTreino900	transposed	purelin	sse	AMO	newff	logsig	learnhd	sse	0%	0%	100%		0
PDefect	PTreino900	transposed	purelin	sse	AMO	newff	logsig	learnhd	sse	0%	0%	100%		0

250 casos de treino

PTest					PTreino250	newp	hardlim	learnp	sse	44%	2%	54%	0,44		
PDefect					PTreino250	newp	hardlim	learnp	sse	42%	10%	48%	0,42		
PTest	PTreino250	pinv	purelin	sse	PerfectArial	newp	hardlim	learnp	sse	18%	12%	70%	0,18		
PDefect	PTreino250	pinv	purelin	sse	PerfectArial	newp	hardlim	learnp	sse	20%	24%	56%	0,2		
PTest	PTreino250	transposed	purelin	sse	AMO	newp	hardlim	learnp	sse	40%	10%	50%		0,4	
PDefect	PTreino250	transposed	purelin	sse	AMO	newp	hardlim	learnp	sse	32%	12%	56%		0,32	
PTest					PTreino250	newp	logsig	learnp	sse	78%	22%	0%	0,78		
PDefect					PTreino250	newp	logsig	learnp	sse	60%	40%	0%	0,6		
PTest	PTreino250	pinv	purelin	sse	PerfectArial	newp	logsig	learnp	sse	52%	46%	2%	0,52		
PDefect	PTreino250	pinv	purelin	sse	PerfectArial	newp	logsig	learnp	sse	38%	62%	0%	0,38		
PTest	PTreino250	transposed	purelin	sse	AMO	newp	logsig	learnp	sse	38%	10%	52%		0,38	
PDefect	PTreino250	transposed	purelin	sse	AMO	newp	logsig	learnp	sse	32%	12%	56%		0,32	
PTest					PTreino250	newff	purelim	learnh	sse	66%	34%	0%	0,66		
PDefect					PTreino250	newff	purelim	learnh	sse	40%	60%	0%	0,4		
PTest	PTreino250	pinv	purelin	sse	PerfectArial	newff	purelim	learnh	sse	58%	42%	0%	0,58		
PDefect	PTreino250	pinv	purelin	sse	PerfectArial	newff	purelim	learnh	sse	42%	58%	0%	0,42		
PTest	PTreino250	transposed	purelin	sse	AMO	newff	purelim	learnh	sse	82%	18%	0%		0,82	
PDefect	PTreino250	transposed	purelin	sse	AMO	newff	purelim	learnh	sse	62%	38%	0%		0,62	
PTest					PTreino250	newff	logsig	learnh	sse	58%	42%	0%	0,58		
PDefect					PTreino250	newff	logsig	learnh	sse	54%	46%	0%	0,54		
PTest	PTreino250	pinv	purelin	sse	PerfectArial	newff	logsig	learnh	sse	48%	52%	0%	0,48		
PDefect	PTreino250	pinv	purelin	sse	PerfectArial	newff	logsig	learnh	sse	38%	62%	0%	0,38		
PTest	PTreino250	transposed	purelin	sse	AMO	newff	logsig	learnh	sse	0%	0%	100%		0	

PDefect	PTreino250	transposed	purelin	sse	AMO	newff	logsig	learnh	sse	0%	0%	100%			0
PTest					PTreino250	newff	purelim	learnhd	sse	60%	40%	0%	0,6		
PDefect					PTreino250	newff	purelim	learnhd	sse	44%	56%	0%	0,44		
PTest	PTreino250	pinv	purelin	sse	PerfectAria	newff	purelim	learnhd	sse	60%	40%	0%		0,6	
PDefect	PTreino250	pinv	purelin	sse	PerfectAria	newff	purelim	learnhd	sse	42%	58%	0%		0,42	
PTest	PTreino250	transposed	purelin	sse	AMO	newff	purelim	learnhd	sse	82%	18%	0%			0,82
PDefect	PTreino250	transposed	purelin	sse	AMO	newff	purelim	learnhd	sse	62%	38%	0%			0,62
PTest					PTreino250	newff	logsig	learnhd	sse	62%	38%	0%	0,62		
PDefect					PTreino250	newff	logsig	learnhd	sse	48%	52%	0%	0,48		
PTest	PTreino250	pinv	purelin	sse	PerfectAria	newff	logsig	learnhd	sse	38%	62%	0%		0,38	
PDefect	PTreino250	pinv	purelin	sse	PerfectAria	newff	logsig	learnhd	sse	36%	64%	0%		0,36	
PTest	PTreino250	transposed	purelin	sse	AMO	newff	logsig	learnhd	sse	0%	0%	100%			0
PDefect	PTreino250	transposed	purelin	sse	AMO	newff	logsig	learnhd	sse	0%	0%	100%			0
PTest					PTreino250	newff	purelim	learnhd	sse	58%	42%	0%	0,58		
PDefect					PTreino250	newff	purelim	learnhd	sse	38%	62%	0%	0,38		
PTest	PTreino250	pinv	purelin	sse	PerfectAria	newff	purelim	learnhd	sse	60%	40%	0%		0,6	
PDefect	PTreino250	pinv	purelin	sse	PerfectAria	newff	purelim	learnhd	sse	42%	58%	0%		0,42	
PTest	PTreino250	transposed	purelin	sse	AMO	newff	purelim	learnhd	sse	82%	18%	0%			0,82
PDefect	PTreino250	transposed	purelin	sse	AMO	newff	purelim	learnhd	sse	62%	38%	0%			0,62
PTest					PTreino250	newff	logsig	learnhd	sse	62%	38%	0%	0,62		
PDefect					PTreino250	newff	logsig	learnhd	sse	48%	52%	0%	0,48		
PTest	PTreino250	pinv	purelin	sse	PerfectAria	newff	logsig	learnhd	sse	40%	60%	0%		0,4	
PDefect	PTreino250	pinv	purelin	sse	PerfectAria	newff	logsig	learnhd	sse	30%	70%	0%		0,3	
PTest	PTreino250	transposed	purelin	sse	AMO	newff	logsig	learnhd	sse	0%	0%	100%			0
PDefect	PTreino250	transposed	purelin	sse	AMO	newff	logsig	learnhd	sse	0%	0%	100%			0

AMO = Associative Memory Output