

AULA LABORATORIAL N.º 2
RECONHECIMENTO DE DÍGITOS - REDES NEURONAIS MLP

1. Objetivo

Pretende-se implementar em MATLAB redes não lineares de múltipla camada usando o algoritmo backpropagation para aprendizagem de parâmetros.

A metodologia deverá ser aplicada a um problema de classificação, nomeadamente reconhecimento de dígitos.

Uma rede perceptron multicamada é totalmente configurável através da definição do número de tipo de camadas, número de neurónios, funções de ativação e algoritmo de aprendizagem. Os algoritmos de treino principais baseiam-se:

- Gradiente descendente, com taxa de aprendizagem configurável, momentum e tamanho dos lotes (batch);
- Levenberg-Marquardt, com parâmetros configuráveis e regularização.

A evolução do treino pode ser analisada em termos de:

- Erro quadrado médio (MSE);
- Número de épocas;
- Relação entre os conjuntos de dados de treino e validação.

2. Arquitectura de Redes MLP (*Multi-layer Perceptron*)

A Figura 1 apresenta o esquema de uma rede neuronal MLP. A rede representada possui p entradas e uma camada de neurónios.

A função de activação usualmente aplicada consiste na função sigmóide combinada com funções lineares na camada de saída (Figura 2).

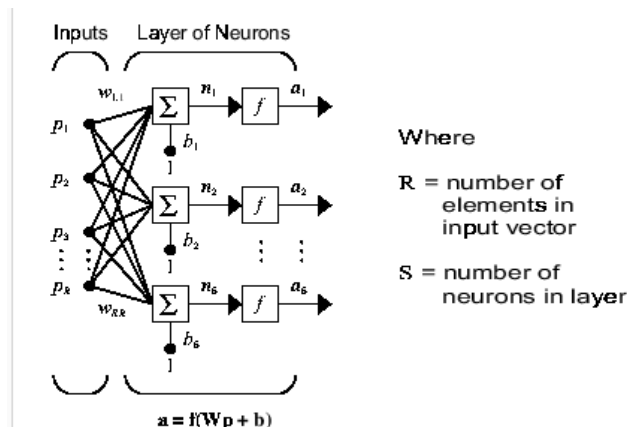


FIGURA 1. REDE MLP (MATLAB TOOLBOX)

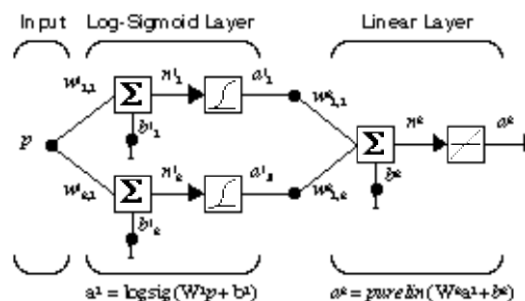


FIGURA 2. ARQUITECTURA TÍPICA.

O **algoritmo de aprendizagem** funciona de forma **supervisionada**: Os pesos são alterados de acordo com o princípio de minimização de um critério de erro, em particular o erro quadrático médio.

O algoritmo divide-se em duas fases:

- Apresenta-se a entrada à rede neuronal e calcula-se o valor de saída.
- Actualizam-se os pesos da rede neuronal baseado nos designados valores “delta”, que medem a contribuição de cada parâmetro para o valor do erro final (“Neural Networks Design”, Cap.11).

3. Aplicação

Pretende-se uma rede neuronal, MLP, capaz de classificar correctamente dígitos de 0 a 9. Cada dígito deve ser representando através de uma matriz de dimensão 7x5 (Figura 3).

A rede neuronal pode ser representada através de uma camada com 35 entradas (valores binários), de uma camada interna com funções sigmóide e um número de neurónios a configurar e 10 saídas. A activação da primeira saída (saída a um) representa o reconhecimento do número zero à entrada, a activação da segunda saída

representa o número um e assim sucessivamente. Em alternativa poderá apresentar apenas uma saída com função de activação linear, como representado na Figura 2.

0	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
0	1	1	1	0

FIGURA 3. REPRESENTAÇÃO DO DÍGITO ZERO, NUMA MATRIZ DE 7x5.

4. Exercícios

Resolva os seguintes exercícios recorrendo às funções disponibilizadas pela toolbox “deep learning” do MATLAB, nomeadamente da função **patternet**.

1. Construa um conjunto de padrões para treino da rede neuronal contendo dígitos de 0 a 9, para resolver um **problema de classificação binária – identificar o dígito “0”**.
 - a. **Construa um conjunto de treino.** Este conjunto deve incluir diversos exemplos referentes a cada dígito, contendo “ruído”, para aumentar a capacidade de generalização.
Considere 10 padrões (matrizes binárias), representativas de “5” e 10 padrões representativos de outros dígitos.
 - b. **Construa um conjunto de teste**, com apenas dois exemplos por classe.
 - c. Teste o desempenho de uma rede MLP no conjunto de treino e teste. Represente a matriz de confusão para o conjunto de teste e comente os resultados.
 - d. Varie o coeficiente de aprendizagem e o número de neurónios. Estude o efeito desses parâmetros na capacidade de classificação da rede neuronal.
 - e. Considera adequada a dimensão dos conjuntos de treino e teste? Justifique.
2. Treine uma rede neuronal MLP para o **dataset disponível na UCI designado de “optdigits”**:

<https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits>

- a. Analise o problema - Especifique as entradas e saídas da rede neuronal.
 - b. Treine uma rede MLP com os exemplos de treino “optdigits.tra”.
 - c. Apresente a matriz de confusão e valores de precisão e *recall* para o conjunto de teste “optdigits.tes”.
3. O dataset MNIST (Modified National Institute of Standards and Technology) – Figura 4, representa um conjunto de exemplos de dígitos manuscritos que é comumente usado para treinar e avaliar vários algoritmos de aprendizagem automática. Treine uma rede neuronal MLP para classificar corretamente este dataset.



FIGURA 4. EXEMPLOS DE IMAGENS NO DATASET MNIST.

- a. Analise o problema - Especifique as entradas e saídas da rede neuronal.

```
images = loadMNISTImages('train-images.idx3-ubyte');
labels = loadMNISTLabels('train-labels.idx1-ubyte');

colormap(gray)                                % set to grayscale
for i = 1:25                                   % preview first 25 samples
    subplot(5,5,i)                             % plot them in 5 x 5 grid
    digit = reshape(images(:,i),[28,28]); %row=28x28 image
    imagesc(digit)                             % show the image
end

disp(labels(1:25))
```

- b. Treine uma rede MLP e avalie para o conjunto de teste:

```
% conjunto de teste
imagesTeste = loadMNISTImages('t10k-images.idx3-ubyte');
labelsTeste = loadMNISTLabels('t10k-labels.idx1-ubyte');
```

Homework:

- Construa uma rede MLP com melhor AUC para o conjunto de teste.

Anexo - Métricas

Definição de uma Matriz de Confusão:

	<i>população</i>	<u>Real</u>	
		Condição Positiva	Condição Negativa
<u>Classificador</u>	Predição Positiva	TP	FP
	Predição Negativa	FN	TN

- **Métricas Comuns:**

Prevalência = Σ Condições positivas / Σ População Total

Taxa de Acerto (ACC-Accuracy) = $(TP+TN)/\text{Tamanho da População}$

Precisão ou PPV-“Positive Predicted Value” = $TP/(TP+FP)$

False discovery rate (FDR) = $FP/(TP+FP)$

False omission rate (FOR)= $FN/(FN+TN)$

Negative predictive value (NPV)= $TN/ (FN+TN)$

Sensibilidade ou Recall (TPR-True Positive Rate)= $TP/(TP+FN)$

False negative rate (FNR)= $FN/ (TP+FN)$

False positive rate (FPR)= $FP/(FP+TN)$

Especificidade (TNR-True Negative Rate)= $TN/FP+TN$

- **Métricas Combinadas:**

F-Measure= $2*(Precision*Recall)/(Precision+Recall)$

F1 score= $2*TP/(2*TP+FP+FN)$ – média harmónica entre precisão e recall

Positive likelihood ratio (LR+)= TPR/FPR

Negative likelihood ratio (LR-)= FNR/TNR