



Licenciatura em Engenharia Informática

Conhecimento e Raciocínio  
Trabalho Prático

*Neural Networks*

Prof.<sup>o</sup> Viriato António Pereira Marinho Marques

Prof.<sup>a</sup> Anabela Borges Simões

Pedro Gonçalo dos Reis Correia

2018020558 – Turma P4

## Índice

Introdução .....	3
Tratamento de imagens .....	4
Estudo do treino de uma rede neuronal .....	5
Estudo do treino de uma rede neuronal e o valor da precisão de teste .....	6
Estudo das precisões de teste com diferentes conjuntos de exemplos .....	7
Teste da pasta test sem re-treinar a rede .....	7
Treino da rede com a pasta test e teste de cada pasta individualmente .....	8
Simulação com formas geradas pelo utilizador .....	10
Aplicação Gráfica .....	11

## Introdução

Este trabalho prático foi realizado no âmbito da Unidade Curricular de Conhecimento e Raciocínio da Licenciatura em Engenharia Informática do Instituto Superior de Engenharia de Coimbra, tem como objetivo o desenvolvimento de redes neuronais, de modo a poder efetuar um estudo estatístico de diferentes arquiteturas de rede. Este projeto foi desenvolvido em MATLAB e encontra-se dividido nas diversas alíneas do enunciado. Foi também desenvolvida uma aplicação gráfica, recorrendo à ferramenta GUIDE do MATLAB.

## Tratamento de imagens

Dado que o objetivo do projeto consiste em treinar redes neuronais para a identificação de diferentes formas geométricas, é necessário começar por transformar estas imagens em matrizes binárias que possam ser interpretadas pelo MATLAB.

Para este efeito, foi criada uma função *read\_images*. Nesta função está definido um mapa que liga os valores binários de *target output* para cada forma, com o seu nome, por exemplo, *circle* tem um *target output* de 100000.

São então percorridas todas as imagens de cada pasta do conjunto indicado, onde cada imagem é lida para uma variável. De modo a conseguir uma melhor eficiência e maior velocidade de treino, estas imagens são redimensionadas para imagens de 25x25. Depois deste redimensionamento, as imagens passam por um tratamento de cor para ficarem a preto e branco. Assim, garantimos que a matriz de inputs gerada de seguida é apenas composta por bits 0 e 1.

Esta matriz gerada é transformada num vetor, de modo a gerar uma coluna na matriz de inputs. A esta matriz de inputs é correspondente o seu target associado.

## Estudo do treino de uma rede neuronal

Para a realização desta tarefa foram utilizados todos os exemplos da pasta *start* para treinar a rede neuronal. Utilizando diferentes tipos de configuração, verificamos que como são utilizados todos os exemplos no treino, a precisão do treino é sempre muito perto de 100%.

Mesmo assim, esta configuração de rede pode não ser a melhor, visto que os dados não são em quantidade suficiente para controlar a qualidade da saída.

## Estudo do treino de uma rede neuronal e o valor da precisão de teste

No ficheiro Excel em anexo, estão detalhadas as diferentes configurações de redes neuronais usadas para estudar o impacto de cada variável no treino de uma rede neuronal.

Assim, verificamos que a melhor rede neuronal tem a seguinte arquitetura:

- Número de *hidden layers*: 1
- Número de neurónios: 10
- Função de treino: trainbr
- Funções de ativação: tansig, purelin
- Divisão dos exemplos: 0.7, 0.15, 0.15

Esta rede obteve uma precisão global de 83.7% e de teste 77.8%.

Será esta a rede a utilizar no restante projeto.

## Estudo das precisões de teste com diferentes conjuntos de exemplos

Teste da pasta test sem re-treinar a rede

Confusion Matrix							
Output Class	1	2	3	4	5	6	
	8 13.3%	0 0.0%	0 0.0%	0 0.0%	5 8.3%	0 0.0%	61.5% 38.5%
	0 0.0%	9 15.0%	0 0.0%	0 0.0%	0 0.0%	2 3.3%	81.8% 18.2%
	0 0.0%	0 0.0%	4 6.7%	1 1.7%	2 3.3%	1 1.7%	50.0% 50.0%
	0 0.0%	0 0.0%	4 6.7%	9 15.0%	0 0.0%	0 0.0%	69.2% 30.8%
	2 3.3%	1 1.7%	0 0.0%	0 0.0%	2 3.3%	0 0.0%	40.0% 60.0%
	0 0.0%	0 0.0%	2 3.3%	0 0.0%	1 1.7%	7 11.7%	70.0% 30.0%

Figura 1 - Matriz de confusão da pasta test

Pela matriz de confusão apresentada acima podemos verificar que a precisão de teste da rede neuronal foi de 65%. Esta percentagem é aceitável, pois os exemplos dados para este teste, eram totalmente novos para a rede.

Treino da rede com a pasta test e teste de cada pasta individualmente

**Test for start - Confusion Matrix**

Output Class	1	2	3	4	5	6	
	5 16.7%	0 0.0%	0 0.0%	0 0.0%	1 3.3%	1 3.3%	71.4% 28.6%
	0 0.0%	3 10.0%	0 0.0%	0 0.0%	0 0.0%	1 3.3%	75.0% 25.0%
	0 0.0%	0 0.0%	4 13.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	1 3.3%	5 16.7%	1 3.3%	0 0.0%	71.4% 28.6%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 10.0%	1 3.3%	75.0% 25.0%
	0 0.0%	2 6.7%	0 0.0%	0 0.0%	0 0.0%	2 6.7%	50.0% 50.0%
	100% 0.0%	60.0% 40.0%	80.0% 20.0%	100% 0.0%	60.0% 40.0%	40.0% 60.0%	73.3% 26.7%
	1	2	3	4	5	6	
	Target Class						

Figura 2 - Matriz de confusão da pasta start

Depois de treinada a rede com os exemplos da pasta test, a simulação dos testes para a pasta start, teve uma precisão de 73.3%.



Test for train - Confusion Matrix								
Output Class	1	46 15.3%	2 0.7%	0 0.0%	0 0.0%	7 2.3%	0 0.0%	83.6% 16.4%
	2	0 0.0%	46 15.3%	1 0.3%	0 0.0%	0 0.0%	5 1.7%	88.5% 11.5%
	3	0 0.0%	0 0.0%	29 9.7%	2 0.7%	1 0.3%	2 0.7%	85.3% 14.7%
	4	1 0.3%	0 0.0%	13 4.3%	48 16.0%	2 0.7%	0 0.0%	75.0% 25.0%
	5	3 1.0%	0 0.0%	0 0.0%	0 0.0%	39 13.0%	0 0.0%	92.9% 7.1%
	6	0 0.0%	2 0.7%	7 2.3%	0 0.0%	1 0.3%	43 14.3%	81.1% 18.9%
			92.0% 8.0%	92.0% 8.0%	58.0% 42.0%	96.0% 4.0%	78.0% 22.0%	86.0% 14.0%
Target Class								

Figura 3 - Matriz de confusão da pasta train

Já para a pasta train, a percentagem de precisão foi de 83.7%.

Test for test - Confusion Matrix								
Output Class	1	8 13.3%	0 0.0%	0 0.0%	0 0.0%	5 8.3%	0 0.0%	61.5% 38.5%
	2	0 0.0%	9 15.0%	0 0.0%	0 0.0%	0 0.0%	2 3.3%	81.8% 18.2%
	3	0 0.0%	0 0.0%	4 6.7%	1 1.7%	2 3.3%	1 1.7%	50.0% 50.0%
	4	0 0.0%	0 0.0%	4 6.7%	9 15.0%	0 0.0%	0 0.0%	69.2% 30.8%
	5	2 3.3%	1 1.7%	0 0.0%	0 0.0%	2 3.3%	0 0.0%	40.0% 60.0%
	6	0 0.0%	0 0.0%	2 3.3%	0 0.0%	1 1.7%	7 11.7%	70.0% 30.0%
								80.0% 20.0%
								90.0% 10.0%
Target Class								
								40.0% 60.0%
								90.0% 10.0%
								20.0% 80.0%
								70.0% 30.0%
								65.0% 35.0%

Figura 4 - Matriz de confusão da pasta test

A pasta de test, obteve o mesmo resultado que anteriormente, 65%.

## Simulação com formas geradas pelo utilizador

Foram desenhadas 12 figuras, 2 de cada forma geométrica aprendida pela rede neuronal.

Foi efetuado um estudo da eficácia da rede na identificação das formas corretas, a qual não correu como esperado.

Como podemos verificar na tabela abaixo, apenas 2 formas foram corretamente identificadas, sendo ambas o círculo.

Mesmo assim, visto que as imagens desenhadas não têm o mesmo tamanho das imagens de treino, pode-se explicar algum deste falhanço.

Shape	Result
circle1	circle
kite1	circle
parallelogram1	trapezoid
square1	circle
trapezoid1	circle
triangle1	square
circle2	circle
kite2	circle
parallelogram2	trapezoid
square2	circle
trapezoid2	circle
triangle2	square

*Figura 5 - Resultado dos testes com desenhos*

## Aplicação Gráfica

De modo a generalizar o processo descrito anteriormente e utilizando algumas das funções desenvolvidas anteriormente, foi desenvolvida uma aplicação gráfica, utilizando a ferramenta GUIDE do MATLAB.

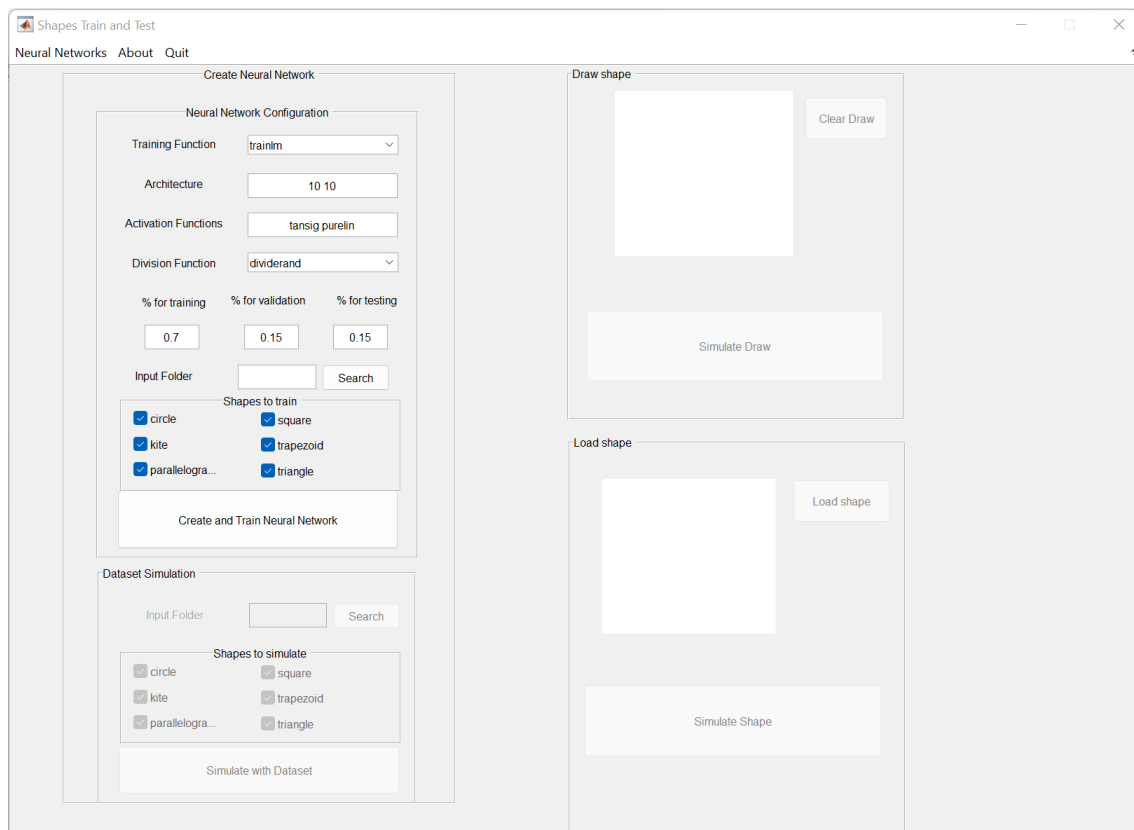


Figura 6 - Janela inicial da aplicação

Esta aplicação permite seleccionar diferentes funções de treino previamente definidas, assim como definir o número de neurónios e camadas escondidas. As funções de ativação devem ser definidas pelo utilizador, as percentagens de divisão também. A função de divisão dos inputs é também seleccionada de uma lista previamente carregada.

O utilizador tem a opção de seleccionar quais as formas que pretende treinar. Este deve também definir a pasta onde se encontram as imagens de exemplo.

Existem outros três modos que são desbloqueados depois do utilizador carregar uma rede neuronal através do menu, ou depois de efetuar o treino de uma rede: simulação com um Dataset, onde o utilizador pode configurar quais formas pretende simular; simulação com um desenho, onde o utilizador pode desenhar uma forma; simulação com um ficheiro, na qual o utilizador escolhe qual a imagem que pretende simular.