

RELATÓRIO DO TRABALHO
PRÁTICO CONHECIMENTO E
RACIOCÍNIO



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

ALUNO N.º 2019110035

Nuno Santos

ALUNO N.º 2020136741

Rafael Gil

ENTIDADE

Instituto Superior de Engenharia de Coimbra

Curso

Licenciatura em Engenharia Informática

COIMBRA – JUNHO – 2022

ÍNDICE

1 - INTRODUÇÃO	3
2 – Decisões tomadas	3
2.1 – Pré processamento das imagens:.....	3
2.2 – Targets	3
2.4 – Dados guardados	4
3 - Treino e Estudo Estatístico	4
3.1 – Alínea A.....	4
3.2 – Alínea B.....	5
3.3 – Alínea C.....	7
3.3.1 - 1ªFase:.....	7
3.3.2 - 2ªFase:.....	7
3.3.3 - 3ªFase:.....	8
5 - Aplicação	8
6 – Conclusões.....	10
7 – Bibliografia	10

1 - INTRODUÇÃO

Este trabalho foi realizado no âmbito da Unidade Curricular de Conhecimento e Raciocínio, tem como objetivo o desenvolvimento e estudo estatístico de redes neuronais para identificação de imagens consoante 6 classes de figuras geométricas (círculo, papagaio, paralelepípedo, quadrado, trapézio e triângulo) com recurso a uma rede neuronal do tipo feed forward.

2 – Decisões tomadas

2.1 – Pré processamento das imagens:

Para questões de otimização da aplicação optamos por redimensionar as imagens para uma resolução de 25x25. Passamos ainda as imagens para binário e fomos preenchendo a matriz dos inputs à medida que liamos as pastas pedidas.

Desenvolvemos quatro funções de processamento, três das quais processam as imagens de cada pasta específica (start, test e train) e a quarta função processa uma imagem individual, indicada através do caminho.

O processamento começa por redimensionar as imagens, transformando-as em 25x25 pixéis, e binarizando-as de maneira a termos cada uma das imagens representada por código binário.

No final, iremos ter uma matriz com os códigos binários de todas as imagens daquela pasta.

2.2 – Targets

Para os targets decidimos fazer uma matriz com vetores com os números 1,2,3,4,5,6 que servem para identificação da figura geométrica em que o 1 é o círculo e assim adiante.

2.4 – Dados guardados

Em vez de apenas guardarmos a matriz de confusão para os testes feitos, decidimos guardar as redes já treinadas na pasta “networks”.

3 Treino e Estudo Estatístico

3.1 – Alínea A

Como se pode ver no Excel, ao testar várias arquiteturas com diferente número de neurónios e camadas escondidas chegámos à conclusão de que os mesmos parâmetros alteram consideravelmente os resultados havendo algumas redes com resultados insatisfatórios.

A melhor rede que obtivemos foi com duas camadas escondidas com 10 neurónios cada uma, com uma precisão total de 76,667% e uma precisão de teste de 31,667% para a pasta test.

O número e dimensão das camadas escondidas influencia o desempenho?							
Conf1	2	5, 5	tansig, tansig	trainlm	dividerand = {0.7, 0.15, 0.15}	56,667	30
Conf2	2	10,10	tansig, tansig	trainlm	dividerand = {0.7, 0.15, 0.15}	76,667	31,667
Conf3	3	10,10,5	tansig, tansig	trainlm	dividerand = {0.7, 0.15, 0.15}	33,333	18,333
Conf4	3	10,10,10	tansig, tansig	trainlm	dividerand = {0.7, 0.15, 0.15}	30	21,667

3.2 – Alínea B

De maneira a conseguir melhores resultados na classificação dos caracteres fornecidos testámos várias funções de treino, ativação e segmentações de divisão.

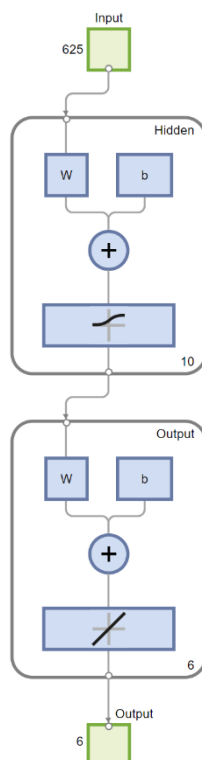
A função de treino influencia o desempenho?							
Conf1	1	10	tansig, purelin	traingd	dividerand = {0.7, 0.15, 0.15}	47,33	35
Conf2	1	10	tansig, purelin	traingdx	dividerand = {0.7, 0.15, 0.15}	53,667	36,667
Conf3	1	10	tansig, purelin	trainrp	dividerand = {0.7, 0.15, 0.15}	25,333	40
Conf4	1	10	tansig, purelin	traingcp	dividerand = {0.7, 0.15, 0.15}	36	30

De entre as funções de treino que testámos a que obtivemos melhores resultados foi a função traingdx, precisão global de 53.667% e uma precisão de teste de 36,667% para a pasta train.

Confusion Matrix							
Output Class	1	2	3	4	5	6	
	40 13.3%	3 1.0%	6 2.0%	6 2.0%	14 4.7%	10 3.3%	50.6% 49.4%
	0 0.0%	11 3.7%	0 0.0%	3 1.0%	0 0.0%	0 0.0%	78.6% 21.4%
	0 0.0%	3 1.0%	26 8.7%	4 1.3%	2 0.7%	7 2.3%	61.9% 38.1%
	2 0.7%	5 1.7%	9 3.0%	30 10.0%	3 1.0%	6 2.0%	54.5% 45.5%
	6 2.0%	2 0.7%	5 1.7%	6 2.0%	28 9.3%	1 0.3%	58.3% 41.7%
	2 0.7%	26 8.7%	4 1.3%	1 0.3%	3 1.0%	26 8.7%	41.9% 58.1%
Target Class							
	1	2	3	4	5	6	
	80.0% 20.0%	22.0% 78.0%	52.0% 48.0%	60.0% 40.0%	56.0% 44.0%	52.0% 48.0%	53.7% 46.3%

As funções de ativação influenciam o desempenho?							
Conf1	1	10	logsig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	95.3	71.6
Conf2	1	10	tansig, logsig	trainlm	dividerand = {0.7, 0.15, 0.15}	74	58.3
Conf3	1	10	compet, elliotsig	trainlm	dividerand = {0.7, 0.15, 0.15}	26.3	26.6
Conf4	1	10	elliotsig, compet	trainlm	dividerand = {0.7, 0.15, 0.15}	12.6	8.3
Conf5	1	10	hardlim, hardlims	trainlm	dividerand = {0.7, 0.15, 0.15}	17	11.6
Conf6	1	10	hardlims, hardlim	trainlm	dividerand = {0.7, 0.15, 0.15}	16.6	16.6

De entre as funções de ativação que testámos a que obtivemos melhores resultados foi a função logsig, com uma precisão global de 95.3% e uma precisão de teste de 71.6% para a pasta train.



A divisão de exemplos pelos conjuntos influencia o desempenho?						
Conf1	1	10	tansig, purelin	trainlm	dividerand = {0.33, 0.33, 0.33}	74.3 60
Conf2	1	10	tansig, purelin	trainlm	dividerand = {0.9, 0.05, 0.05}	94.6 76.6
Conf3	1	10	tansig, purelin	trainlm	dividerand = {0.80, 0.10, 0.10}	92.3 65
Conf4	1	10	tansig, purelin	trainlm	dividerand = {0.50, 0.25, 0.25}	77 60.1
Conf5	1	10	tansig, purelin	trainlm	dividerand = {0.60, 0.20, 0.20}	84 68.3
Conf6	1	10	tansig, purelin	trainlm	dividerand = {0.40, 0.30, 0.30}	74.6 66.6

No último teste que realizamos nesta alínea, alterámos as segmentações de divisão, obtendo o melhor resultado com os parâmetros de 90% para treino, 5% para validação e 5% para teste, com uma precisão total de 94,6% e precisão de teste 76,6%.

3.3 – Alínea C

3.3.1 - 1ªFase:

Sem treinar novamente a rede ao identificar as imagens da pasta “test”, a melhor rede neuronal da fase b) tem uma precisão de 36.667%.

Sem treinar novamente a rede ao identificar as imagens da pasta “train”, a melhor rede neuronal da fase b) tem uma precisão de 53.667%.

Sem treinar novamente a rede ao identificar as imagens da pasta “start”, a melhor rede neuronal da fase b) tem uma precisão de 23.333%.

3.3.2 - 2ªFase:

A melhor rede neuronal da fase b), treinada com a pasta “test”, tem uma precisão de 40% ao identificar as imagens da pasta “start”.

A melhor rede neuronal da fase b), treinada com a pasta “test”, tem uma precisão de 47.667% ao identificar as imagens da pasta “train”.

A melhor rede neuronal da fase b), treinada com a pasta “test”, tem uma precisão de 85.7% ao identificar as imagens da pasta test.

3.3.3 - 3ª Fase:

A melhor rede neuronal da fase b), treinadas com todas as pastas, tem uma precisão de 100% ao identificar as imagens da pasta “start”.

A melhor rede neuronal da fase b), treinadas com todas as pastas, tem uma precisão de 98.333% ao identificar as imagens da pasta “test”.

A melhor rede neuronal da fase b), treinadas com todas as pastas, tem uma precisão de 98.667% ao identificar as imagens da pasta “train”.

5 Aplicação

Início:

The image shows a MATLAB App window titled "MATLAB App" with a "Desenhar" (Design) tab. The app is divided into two main sections: "Criação de Neural Network" (Neural Network Creation) and "Simulação com Dataset" (Simulation with Dataset).

Criação de Neural Network:

- Configuração da Rede:**
 - Função de Treino:
 - Função de Ativação 1:
 - Função de Ativação 2:
 - Função de Divisão:
 - train: val: test:
- Inputs:**
 - Pasta para Treino:
 - Nome da rede:
 -
- Results:**

Simulação com Dataset:

- Inputs:**
 - Pasta para Simulação:
- Rede neuronal:**
 - Nome da rede:
-
- Results:**

Simulação com Ficheiro Único:

- Inputs:**
 - Ficheiro para simulação:
- Rede neuronal:**
 - Nome da rede:
-
- Results:**

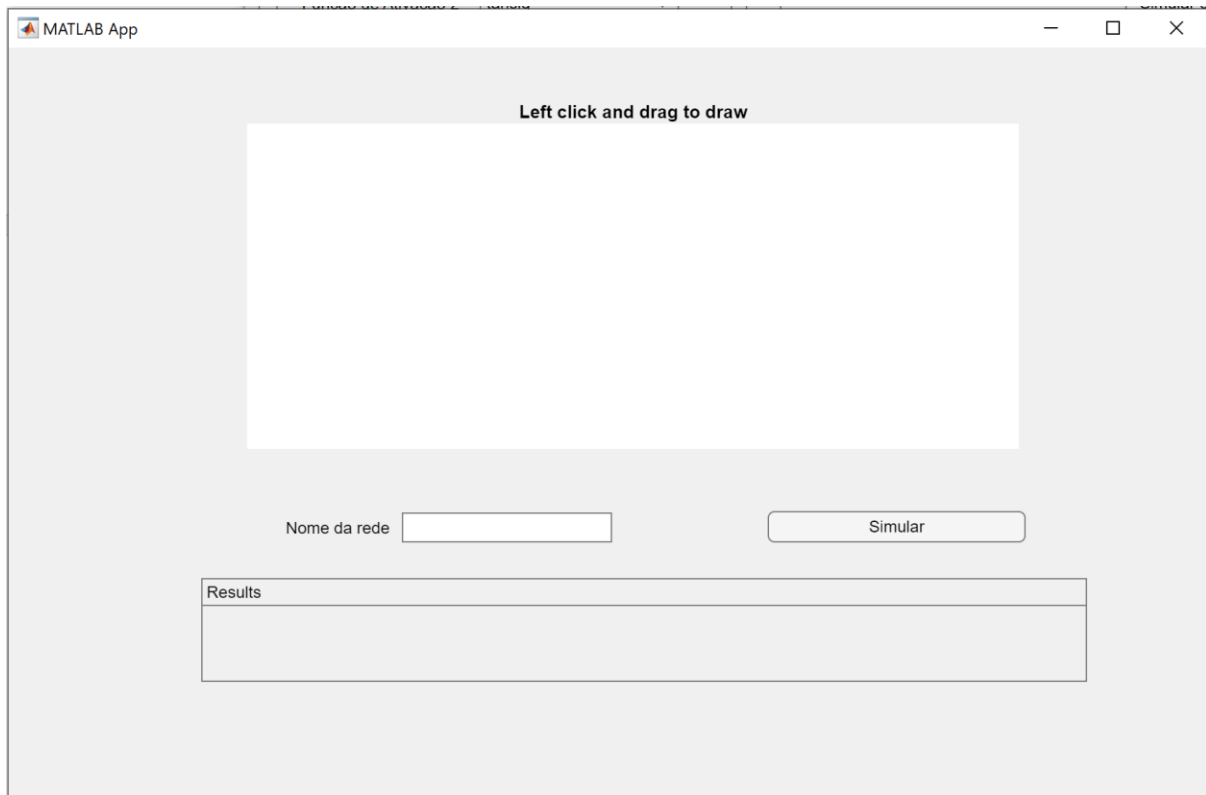
No painel “Criação de Neural Network”, podemos escolher a função de treino e de ativação e ainda as segmentações de divisão desejados para criar e treinar a rede, à qual também é possível nomear. Após realizar o treino da rede, vai ser apresentada a precisão da rede.

No painel “Simulação com DataSet”, temos que escolher a pasta com que queremos testar a rede. Temos também que fornecer o caminho para a rede que desejamos testar. Será apresentada a percentagem de imagens que a rede conseguiu reconhecer.

No painel “Simulação com Ficheiro Único”, temos que fornecer o caminho para a imagem que queremos que a rede reconheça, e à semelhança do painel anterior, fornecer o caminho para a rede com que queremos identificar a imagem. Será depois, apresentada a classificação que a rede atribuiu à imagem que foi fornecida.

Existe um botão no topo da aplicação, que ao clicar vai abrir uma outra aplicação que permite desenhar.

Desenhar:



Nesta aplicação, temos que desenhar para fornecer uma imagem à rede para que ela a consiga classificar. Para tal, é preciso desenhar no UI axes fornecido e introduzir o nome da rede que se quer usar para identificar a imagem desenhada. No final, será apresentada a classificação que a rede atribui à imagem desenhada.

6 – Conclusões

Observámos que a parametrização das redes é crucial no seu desempenho, sendo esta parametrização dependente do problema em questão. De alta importância é também o tratamento prévio das imagens, que permite à rede focar-se nas características que são mais relevantes ao problema, neste caso, a forma das figuras.

Concluimos, então, este trabalho sobre redes neuronais, que consideramos ter sido um sucesso, tanto em termos de aprendizagem e aplicação de conceitos como em termos de desenvolvimento de redes neuronais com desempenhos aceitáveis.

7 – Bibliografia

Como fontes utilizamos principalmente a Wikipédia, o moodle e os documentos da teórica:

Moodle: [Curso: 60023545 Conhecimento e Raciocínio LEI 2122 \(isec.pt\)](https://moodle.unl.pt/course/view.php?id=60023545)

Ibm: <https://www.ibm.com/br-pt/cloud/learn/neural-networks>

Linkedin: https://pt.linkedin.com/pulse/intelig%C3%A2ncia-artificial-o-que-%C3%A9-machine-learning-e-crivelari-costa?trk=articles_directory

Deep Learning: <https://www.deeplearningbook.com.br/o-perceptron-parte-1/>