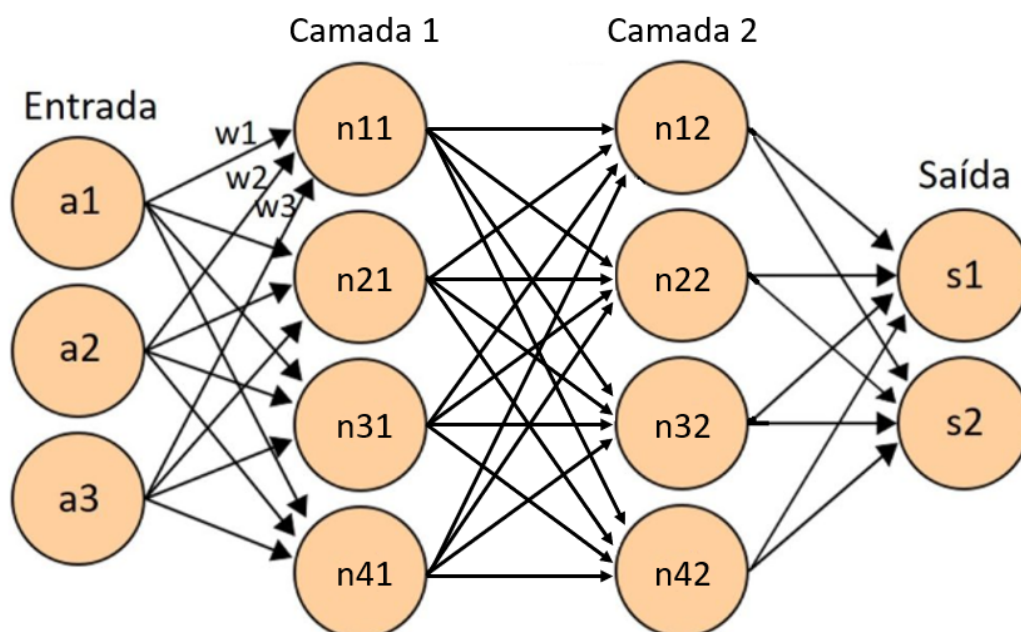


RELATÓRIO DO TRABALHO PRÁTICO

Tema 1: Redes Neurais



Índice

Alínea a)	2
preprocess.m.....	2
getTarget.m.....	3
A.m.....	4
Alínea b)	6
Alínea c)	8
Alínea d)	9

Alínea a)

preprocess.m

Esta função realiza a conversão das imagens presentes na diretoria escolhida para matrizes binárias, a fim de prepará-las para a rede neuronal. Recebe como parâmetros a diretoria das imagens e a resolução desejada para a conversão em matrizes binárias. Primeiramente, a função verifica a quantidade de imagens presentes no diretório. Em seguida, lê cada imagem, ajusta a sua resolução para o valor fornecido como entrada, converte-a para uma matriz binária e armazena essa matriz num array. Terminado o processo, a função devolve uma lista que contém todas as matrizes binárias produzidas.

```
function [input, size] = preprocess(folder, px)
    directory = dir(folder);
    size = length(directory);
    input = [];

    for i = 1 : size
        img = imread(append(directory(i).folder, "/", directory(i).name));
        img = img(:,:,1);

        img = imresize(img, [px, px]);

        img = imbinarize(img);

        img = img(:);

        input(:, i) = img;
    end
end
```

getTarget.m

Esta função faz uma matriz alvo para a rede neuronal que será treinada. Ela recebe o tamanho das imagens que foram convertidas para matriz binária e devolve a matriz alvo da rede neuronal.

```
function target = getTarget(tamanho, size)
    target = zeros(size, tamanho);
    n = floor(tamanho / size);

    linha = 1;
    for i = 1 : tamanho
        target(linha, i) = 1;
        if mod(i, n) == 0
            linha = linha + 1;
        end
    end
end
```

A.m

Neste ficheiro será feita a configuração da rede neuronal, o treino de 10 vezes e o cálculo da precisão média das 10 redes neurais treinadas. Em primeiro lugar são limpos os valores das variáveis em memória e também a consola. Logo após, ocorre o processamento das imagens com recurso à função preprocess e é feita a matriz alvo para os 10 treinos. Com o final de cada treino é feita a simulação de cada rede neuronal para fazer a avaliação da precisão.

```
clear all
close all
clc

folder = "../data/start/*//*.png";
[input, tamanho] = preprocess(folder, 28);
target = getTarget(tamanho);

epochs = 10;

accuracyFinal = 0;
nSim = 10;

for sim = 1 : nSim
    net = feedforwardnet(10);
    net.trainParam.epochs = epochs;

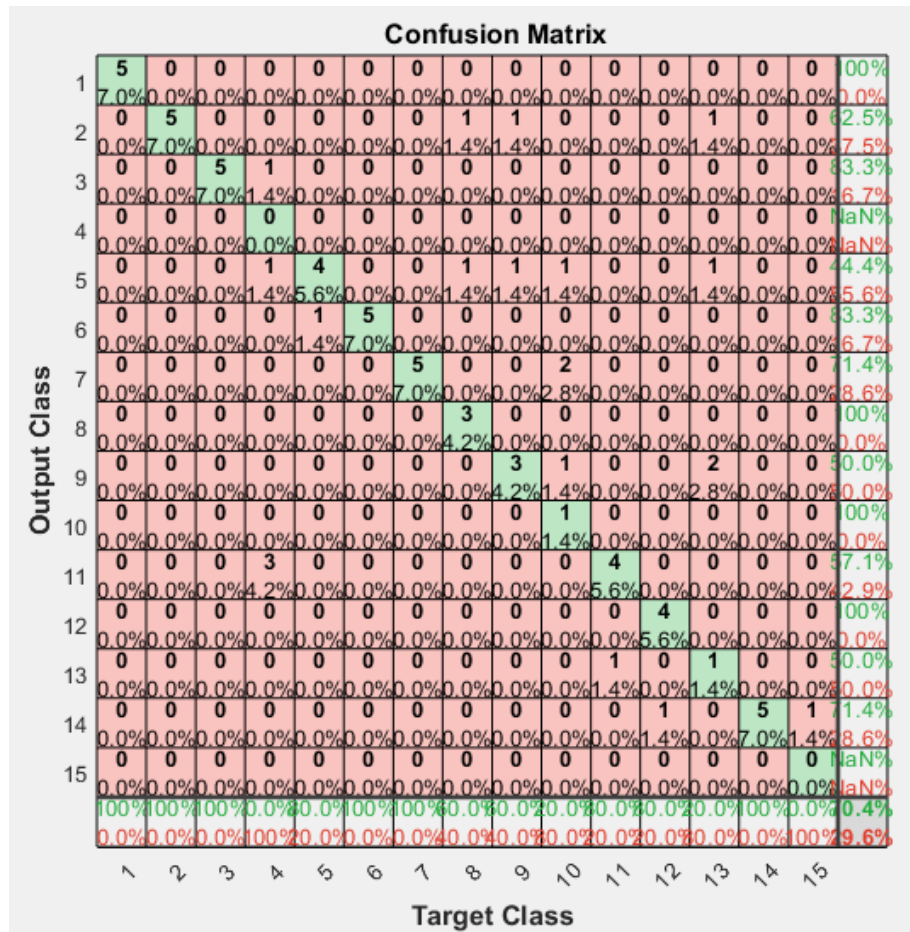
    [net, tr] = train(net, input, target);

    out = net(input);
    plotconfusion(target, out);

    r = 0;
    for i = 1:size(out, 2)
        [~, b] = max(out(:, i));
        [~, d] = max(target(:, i));
        if b == d
            r = r + 1;
        end
    end
    accuracy = r / size(target, 2) * 100;
    fprintf('Precisão na iteração %d: %.3f\n', sim, accuracy)
    accuracyFinal = accuracyFinal + accuracy;
end

fprintf('\nMédia da precisão depois de %i iterações: %.3f\n', nSim, accuracyFinal/nSim);
```

A matriz de confusão mostra-nos visualmente o desempenho da simulação da rede neuronal.



Alínea b)

Nesta alínea serão feitas várias configurações diferentes para ver qual seria a melhor configuração para o objetivo do nosso trabalho. Com isso, optámos por modificar as funções de ativação, a função de treino e a divisão dos exemplos.

As funções de ativação escolhidas foram *purelin*, *logsig* e *tansig*.

As funções de treino escolhidas foram *trainlm*, *trainbfg* e *traingd*. Como a função *traingb* demorava muito no treino de uma só rede e os seus valores eram muitos baixos, constatámos que não seria uma boa opção de configuração.

A divisão de exemplos seria 70% para treino, 15% para validação e 15% para teste.

Também era pedida a questão de separar as redes neuronais em duas partes distintas: uma para lidar com os números e outra para tratar as operações. Seguem na página seguinte os resultados dos testes realizados.

Rede Neural Números + Operações							
	Número de camadas	Número de neurónios	Funções de ativação	Função de treino	Divisão de exemplos	Média de precisão	Média do teste
Config1	10	10	purelin	trainlm	dividerand = { 70, 15, 15 }	70,914	54,19
Config2	10	10	logsig	trainlm	dividerand = { 70, 15, 15 }	20,486	18,095
Config3	10	10	tansig	trainlm	dividerand = { 70, 15, 15 }	76,157	46
Config4	10	10	purelin	trainbfg	dividerand = { 70, 15, 15 }	6,429	6,667
Config5	10	10	purelin	traingd	dividerand = { 70, 15, 15 }	8,286	5,714
Config6	10	10	logsig	traingd	dividerand = { 70, 15, 15 }	5,571	4,762
Config7	10	10	tansig	traingd	dividerand = { 70, 15, 15 }	5	2,857

Rede Neural Números							
	Número de camadas	Número de neurónios	Funções de ativação	Função de treino	Divisão de exemplos	Média de precisão	Média do teste
Config1	10	10	purelin	trainlm	dividerand = { 70, 15, 15 }	69,12	53,333
Config2	10	10	logsig	trainlm	dividerand = { 70, 15, 15 }	15,56	10,267
Config3	10	10	tansig	trainlm	dividerand = { 70, 15, 15 }	67,94	27,867
Config4	10	10	purelin	trainbfg	dividerand = { 70, 15, 15 }	7,2	8,5
Config5	10	10	purelin	traingd	dividerand = { 70, 15, 15 }	6,8	6,267
Config6	10	10	logsig	traingd	dividerand = { 70, 15, 15 }	6,28	6,933
Config7	10	10	tansig	traingd	dividerand = { 70, 15, 15 }	5,32	5,733

Rede Neural Operações							
	Número de camadas	Número de neurónios	Funções de ativação	Função de treino	Divisão de exemplos	Média de precisão	Média do teste
Config1	10	10	purelin	trainlm	dividerand = { 70, 15, 15 }	91,6	30,133
Config2	10	10	logsig	trainlm	dividerand = { 70, 15, 15 }	8,7	2,133
Config3	10	10	tansig	trainlm	dividerand = { 70, 15, 15 }	60,6	5,333
Config4	10	10	purelin	trainbfg	dividerand = { 70, 15, 15 }	6,667	2,667
Config5	10	10	purelin	traingd	dividerand = { 70, 15, 15 }	7,5	2,8
Config6	10	10	logsig	traingd	dividerand = { 70, 15, 15 }	6,25	2,533
Config7	10	10	tansig	traingd	dividerand = { 70, 15, 15 }	6,35	2,933

Acabámos por escolher a *Config1* para todas as 3 redes neuronais, já que tinham os melhores valores de precisão média e de teste médio.

Alínea c)

Esta alínea pedia para fazermos as nossas imagens para o teste. Portanto, fizemos 5 imagens para cada número e operação.

Os valores resultantes foram os seguintes:

- Média da precisão [da rede num + op] depois de 10 iterações: **11.940**
- Média da precisão [da rede num] depois de 10 iterações: **12.766**
- Média da precisão [da rede op] depois de 10 iterações: **65.000**

Os valores que tivemos não nos fizeram sentido, especialmente para a rede neuronal números + operações. A média de precisão desta, como as imagens são parecidas com aquelas usadas para o treino das redes, deveria ser bastante superior aos **11.940** obtidos. As outras duas redes aceitamos que tenham sido comprometidas pela nossa falha de configuração na parte do treino das redes neuronais.

Alínea d)

Por fim, era pedido para fazer um interface que permitisse desenhar uma conta matemática e calcular o valor que a rede(s) neuronal(is) leu(ram).

A interface deveria, então, ser feita com duas opções:

- utilizar a rede neuronal com treino para números + operações
- utilizar a rede neuronal dedicada aos números + rede neuronal dedicada às operações

Devido a alguns problemas, não foi concluída a interface na questão de ser possível desenhar números e símbolos na mesma. Com isso, a interface fica por terminar neste trabalho

Redes Neurais
Trabalho Prático de CR

Quantas redes

Carregar

Testar