

Conhecimento e Raciocínio

Tema 1 – Redes Neurais



A *e*
i
O *U*

Licenciatura Eng. Informática

2º Ano/2º Semestre

2019/20

Carolina Ferreira, nº 2018018459

Marco Domingues, nº 2018016632

Índice

Introdução	3
Enquadramento Geral.....	4
Testes e Resultados	7
Pasta_1	7
Variar a função de treino.....	7
Variar o número de neurónios em cada função de treino.....	9
Variar o número de camadas e de neurónios com várias combinações de funções de ativação	13
Variar o número de épocas.....	14
Alterar o tipo de rede.....	14
Pasta_2	15
Variar o número de neurónios	16
Variar a função de treino.....	18
Variar a função de ativação.....	19
Conclusão.....	20

Introdução

O presente trabalho insere-se na unidade curricular de Conhecimento e Raciocínio. Foram propostos, pelos docentes responsáveis pela unidade curricular, quatro temas principais: Redes Neurais, Sistemas Periciais, Lógica Difusa e CBR, sendo o primeiro o eleito por nós.

O objetivo deste trabalho consiste na implementação de uma rede neuronal capaz de classificar corretamente caracteres alfabéticos – as cinco vogais (A, E, I, O, U). De modo a possibilitar todo esse processo, foram-nos fornecidos os ficheiros de imagens a preto e branco, separadas por **três** pastas diferentes, que devem ser usadas nas tarefas pedidas no enunciado. A Pasta_1 contém um exemplar de cada letra, ou seja, um total de apenas 5 imagens, a Pasta_2 contém 5 subpastas – uma para cada letra – e cada uma dessas subpastas contém 200 imagens com representação da respetiva letra e, por fim, a Pasta_3, é em tudo semelhante à pasta 2, mas com 100 imagens para representar cada letra.

Ao longo da implementação fomos realizando alguns testes com o intuito de verificar se as redes treinadas conseguiam classificar, ou não, corretamente e, para além disso, fomos também avaliando várias topologias de redes neurais.

Enquadramento Geral

O presente trabalho divide-se essencialmente em 4 pontos principais:

1.

1.1. Converter as imagens fornecidas em matrizes binárias:

```
input = zeros(784,5);  
i = 1;  
for n = 1 : size(d)  
    folder = getfield(d, {i}, 'folder'); % folder é a pasta  
    nome = getfield(d, {i}, 'name'); % vai buscar cada imagem  
    img = strcat(folder, '/', nome); % cria localização final  
    imagem_arr = imbinarize(imread(img));  
    vector = imagem_arr(:);  
    input(:, i) = vector; % cada coluna é uma imagem  
    i = i + 1;  
end
```

NOTA: Como as imagens são de 28x28 pixels, o que se traduz em 784 inputs, não considerámos necessário fazer um redimensionamento prévio às imagens.

1.2. Começar por uma rede neuronal de uma camada com 10 neurónios e usar essa rede para treinar o reconhecimento dos caracteres da Pasta_1 e, posteriormente, testar outras topologias, funções de ativação e de treino, registando e comparando os resultados obtidos:

```
target = [1 0 0 0 0;  
          0 1 0 0 0;  
          0 0 1 0 0;  
          0 0 0 1 0;  
          0 0 0 0 1];  
  
% Criação de rede neuronal com 10 neurónios com 1 camada escondida  
%net = feedforwardnet(10);  
%net = feedforwardnet([3 2 2 1]);  
net = patternnet(10);  
  
% FUNCAO DE ATIVACAO DA CAMADA DE SAIDA  
%net.layers{1}.transferFcn = 'tansig';  
% FUNCAO DE ATIVACAO DA CAMADA DE SAIDA  
net.layers{1}.transferFcn = 'purelin';  
% FUNCAO DE ATIVACAO DA CAMADA DE SAIDA  
%net.layers{3}.transferFcn = 'purelin';  
  
% FUNCAO DE TREINO  
net.trainFcn = 'trainlm';  
% NUMERO DE EPOCAS DE TREINO  
net.trainParam.epochs=100;  
  
net.divideFcn = 'dividerand';  
net.divideParam.trainRatio = 1;  
net.divideParam.valRatio = 0;  
net.divideParam.testRatio = 0;
```

```

% |treinar a rede
[net,tr] = train(net, input, target);
view(net);
disp(tr)

% % SIMULAR
out = sim(net, input);

% % Treinar a rede
%net2 = net;

% %VISUALIZAR DESEMPENHO

plotconfusion(target, out) % Matriz de confusao

plotperf(tr) % Grafico com o desempenho da rede nos 3 conjuntos

```

2. Fazer as alterações necessárias para implementar e testar várias topologias e parametrizações de RN de forma a obter um bom desempenho para a classificação das vogais fornecidas na pasta Pasta_2.

- 2.1. Começar por usar uma segmentação do dataset de 70%, 15%, 15% para treino, validação e teste;

```

% NUMERO DE EPOCAS DE TREINO
net.trainParam.epochs=100;

```

```

net.divideFcn = 'dividerand';
net.divideParam.trainRatio = 0.7;
net.divideParam.valRatio = 0.15;
net.divideParam.testRatio = 0.15;

```

- 2.2. Observar a matriz de confusão, erros de treino e teste;
(anexada na secção “Testes e Resultados”).)

- 2.3. Explorar e comparar várias configurações da rede;

```

net = feedforwardnet(10);
%net = patternnet(10);

```

2.4. Testar diferentes funções de treino/ativação, diferentes segmentações na divisão dos exemplos;

```
% FUNCAO DE ATIVACAO DA CAMADA DE SAIDA
net.layers{1}.transferFcn = 'tansig';
% FUNCAO DE ATIVACAO DA CAMADA DE SAIDA
%net.layers{2}.transferFcn = 'purelin';
% FUNCAO DE ATIVACAO DA CAMADA DE SAIDA
%net.layers{3}.transferFcn = 'purelin';

% FUNCAO DE TREINO
net.trainFcn = 'trainlm';
% FUNCAO DE TREINO
% net.trainFcn = 'traingdx';
```

2.5. Gravar a(s) rede(s) neuronal(ais) com melhor(es) desempenho(s) no ficheiro excel.

3. Utilizar as imagens da Pasta_3 que não foram usadas no treino anterior e, sem treinar a rede, verificar se a classificação dada pela RN é correta.

(resultados a observar no capítulo “Testes e Resultados”).

3.1. Voltar a treinar a rede só com a Pasta_3. Testar a rede separadamente para as imagens da Pasta_1, Pasta_2 e Pasta_3. Comparar e registar os resultados obtidos.

3.2. Voltar a treinar a rede com todas as imagens fornecidas (Pasta1 + Pasta_2 +Pasta_3). Testar a rede para as imagens da Pasta_1, Pasta_2 e Pasta_3 em separado. Comparar e registar os resultados obtidos.

4. Desenhar manualmente algumas vogais com semelhanças com os exemplos usados no treino da rede. Transcrever os desenhos para matrizes binárias. Desenvolver um pequeno programa para ler um ficheiro correspondente a uma destas imagens e aplicá-lo à melhor rede neuronal obtida em c)

Testes e Resultados

Todos os resultados dos testes efetuados são apresentados neste capítulo.

Alínea a): Pasta_1

Configuração padrão:

Tipo de rede: feedforwardnet

Número de camadas: 1

Número de neurónios: 10

Função de treino: trainlm

Função de ativação: purelin

net.divideFcn: dividerand

TrainRatio: 1.0

ValRatio: 0

TestRatio: 0

Épocas: 100

Variar a função de treino

Função de treino: trainlm

<i>Nº de execuções</i>	1	2	3	4	5	6	7	8	9	10	<i>Média</i>
Precisão Global	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Precisão de treino	NaN										

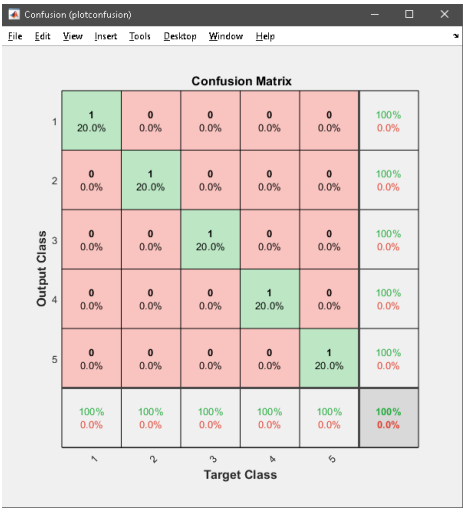


Figura 1. Melhor matriz de confusão

Função de treino: traingd

<i>Nº de execuções</i>	1	2	3	4	5	6	7	8	9	10	<i>Média</i>
Precisão Global	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Precisão de treino	NaN										

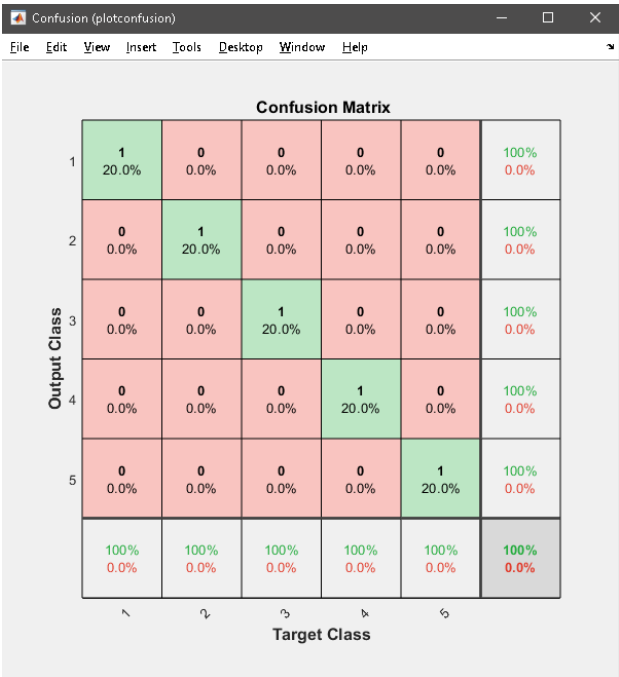


Figura 2. Melhor matriz de confusão

Função de treino: trainbr

<i>Nº de execuções</i>	1	2	3	4	5	6	7	8	9	10	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Precisão de treino</i>	NaN										

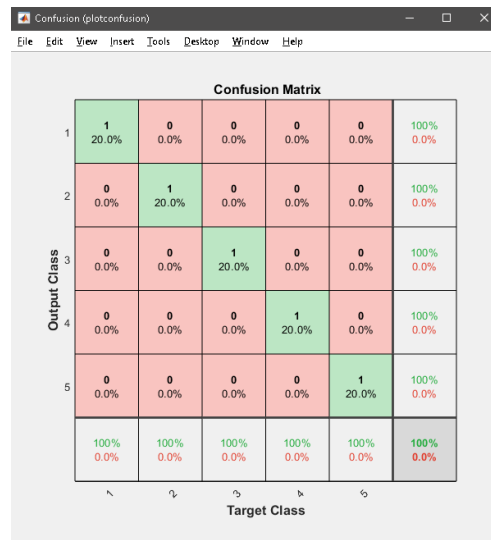


Figura 3. Melhor matriz de confusão

Função de treino: traingd

<i>Nº de execuções</i>	1	2	3	4	5	6	7	8	9	10	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Precisão de treino</i>	NaN										

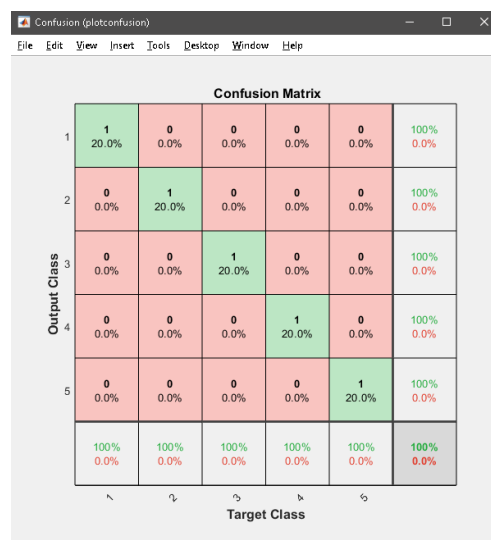


Figura 4. Melhor matriz de confusão

Variar o número de neurónios em cada função de treino

Função de treino: trainml
Nº de neurónios: 5

Nº de execuções	1	2	3	4	5	6	7	8	9	10	Média
Precisão Global	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Precisão de treino	NaN										

Função de treino: trainml
Nº de neurónios: 20

Nº de execuções	1	2	3	4	5	6	7	8	9	10	Média
Precisão Global	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Precisão de treino	NaN										

Função de treino: trainml
Nº de neurónios: 50

Nº de execuções	1	2	3	4	5	6	7	8	9	10	Média
Precisão Global	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Precisão de treino	NaN										

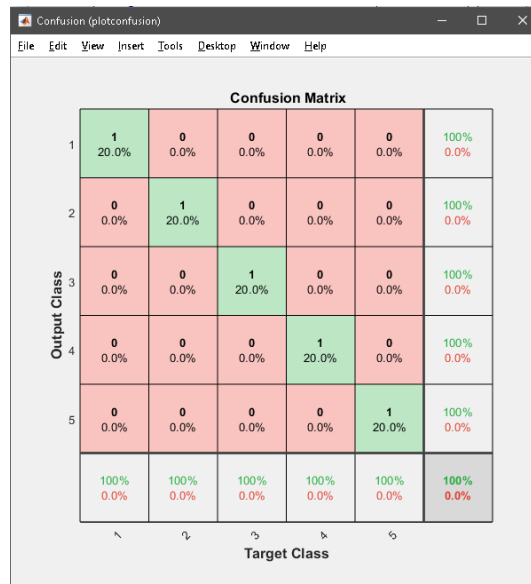


Figura 5. Melhor matriz de confusão

Função de treino: traincgf
Nº de neurónios: 5

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Precisão de treino</i>	NaN										

Função de treino: traincgf
Nº de neurónios: 20

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Precisão de treino</i>	NaN										

Função de treino: traincgf
Nº de neurónios: 50

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Precisão de treino</i>	NaN										

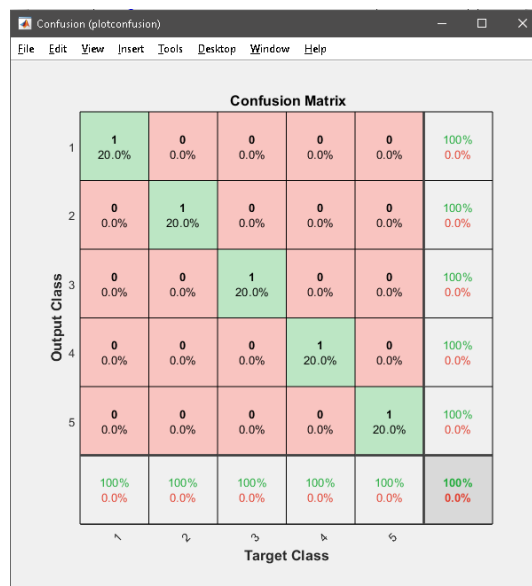


Figura 6. Melhor matriz de confusão

Função de treino: traincgp
Nº de neurónios: 50

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Precisão de treino</i>	NaN										

Função de treino: traincgp
Nº de neurónios: 100

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Precisão de treino</i>	NaN										

Função de treino: traincgp
Nº de neurónios: 150

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Precisão de treino</i>	NaN										

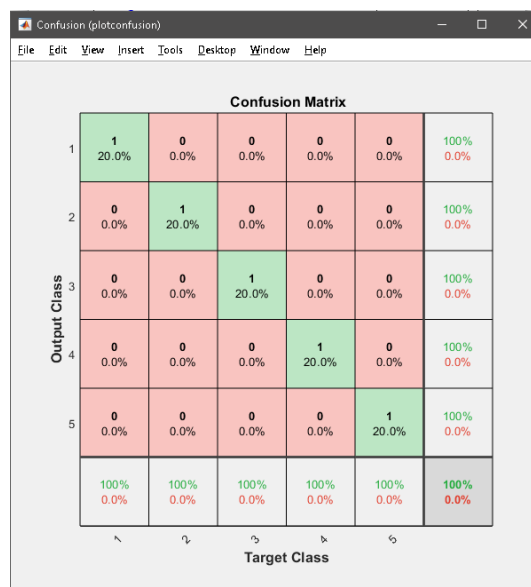


Figura 7. Melhor matriz de confusão

Variar o número de camadas e de neurónios com várias combinações de funções de ativação

<i>Nº de execuções</i>	<i>Nº de camadas</i>	<i>Nº de neurónios</i>	<i>Funções ativação</i>	<i>Precisão do treino</i>	<i>Média</i>
1	1	5	tansig	100%	100%
2	1	5	tansig	100%	
3	1	5	tansig	100%	
4	1	5	tansig	100%	
5	1	5	tansig	100%	
1	1	5	purelin	100%	100%
2	1	5	purelin	100%	
3	1	5	purelin	100%	
4	1	5	purelin	100%	
5	1	5	purelin	100%	
1	2	5, 5	tansig, purelin	100%	100%
2	2	5, 5	tansig, purelin	100%	
3	2	5, 5	tansig, purelin	100%	
4	2	5, 5	tansig, purelin	100%	
5	2	5, 5	tansig, purelin	100%	
1	2	5, 5	purelin, tansig	80%	76%
2	2	5, 5	purelin, tansig	80%	
3	2	5, 5	purelin, tansig	100%	
4	2	5, 5	purelin, tansig	60%	
5	2	5, 5	purelin, tansig	60%	
1	3	2, 2, 1	tansig, purelin, tansig	100%	100%
2	3	2, 2, 1	tansig, purelin, tansig	100%	
3	3	2, 2, 1	tansig, purelin, tansig	100%	
4	3	2, 2, 1	tansig, purelin, tansig	100%	
5	3	2, 2, 1	tansig, purelin, tansig	100%	
1	3	2, 2, 1	purelin, purelin, purelin	100%	100%
2	3	2, 2, 1	purelin, purelin, purelin	100%	
3	3	2, 2, 1	purelin, purelin, purelin	100%	
4	3	2, 2, 1	purelin, purelin, purelin	100%	
5	3	2, 2, 1	purelin, purelin, purelin	100%	

Variar o número de épocas

Nº de épocas: 10

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Épocas Realizadas</i>	10	10	10	10	10	10	10	10	10	10	

Nº de épocas: 50

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Épocas Realizadas</i>	35	28	23	30	27	14	22	24	29	25	

Nº de épocas: 100

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Épocas Realizadas</i>	29	53	45	23	35	32	32	32	26	30	

Nº de épocas: 1000

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
<i>Épocas Realizadas</i>	23	32	30	39	21	47	30	29	32	31	

Alterar o tipo de rede

Configuração padrão:

Tipo de rede: Patternnet

Nº de camadas: 1

Função de treino: trainlm

Função de ativação: purelin

Net.divideFcn: dividerand

Nº de neurónios: 10

TrainRatio: 1.0

ValRatio: 0

TestRatio: 0

Épocas: 100

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
Precisão Global	60%	100%	100%	40%	80%	100%	100%	60%	100%	100%	84%
Épocas Realizadas	42	14	18	13	8	11	20	11	16	16	

Alínea b): Pasta_2

Variar tipo de rede

Tipo de rede: patternnet

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Global</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}		

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
Precisão Global	94.4	88.0	93.2	91.7	94.1	92.8	93.3	90.6	90.5	94.9	92.35
Precisão de treino	88.7	81.4	80.7	87.4	83.4	81.4	83.4	82.1	76.1	84.1	82.87

Tipo de rede: feedforwardnet

	Camadas	Nº de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste
Configuração	1	10	tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}		

Nº de execuções	1	2	3	4	5	6	7	8	9	10	Média
Precisão Global	92.9	92.4	93.3	92.4	93.0	93.1	90.0	92.2	94.9	87.8	92,2
Precisão de treino	84.7	81.4	77.4	81.4	79.4	86.0	73.5	76.1	89.4	68.2	79,75

Variar o número de neurónios

Nº de execuções	Camadas	Nº de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste	Média
1	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	94.2	82.7	Precisão Global: 90,88 Precisão Teste: 77,02
2	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	91.4	80.7	
3	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	93,0	81.4	
4	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	82.2	63.5	
5	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	93.6	76.8	
1	2	10,10	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	93.0	82.7	

2	2	10,10	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	94.4	84.7	Precisão Global: 89,66
3	2	10,10	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	76.4	70.8	
4	2	10,10	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	93.7	82.7	
5	2	10,10	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	90.8	81.4	Precisão Teste: 80,46
1	2	20,20	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	94.8	86.0	Precisão Global: 94,74
2	2	20,20	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	93.8	82.7	
3	2	20,20	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	94.9	86.7	
4	2	20,20	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	95.5	88.0	
5	2	20,20	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	94.7	82.7	
1	2	50,50	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	91.4	74.8	Precisão Teste: 85,22

Variar a função de treino

Nº de execuções	Camadas	Nº de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste	Média
1	1	10	tansig, purelin	traingd	dividerand={0.7,0.15,0.15}	19.9	19.8	Precisão Global: 22,22
2	1	10	tansig, purelin	traingd	dividerand={0.7,0.15,0.15}	31.6	29.8	
3	1	10	tansig, purelin	traingd	dividerand={0.7,0.15,0.15}	19.9	15.2	
4	1	10	tansig, purelin	traingd	dividerand={0.7,0.15,0.15}	19.8	15.2	
5	1	10	tansig, purelin	traingd	dividerand={0.7,0.15,0.15}	19.9	22.5	
								Precisão Teste: 20,5
1	1	10	tansig, purelin	trainbfg	dividerand={0.7,0.15,0.15}	19.9	13.2	Precisão Global: 19,9
2	1	10	tansig, purelin	trainbfg	dividerand={0.7,0.15,0.15}	19.9	19.8	
3	1	10	tansig, purelin	trainbfg	dividerand={0.7,0.15,0.15}	19.9	14.6	
4	1	10	tansig, purelin	trainbfg	dividerand={0.7,0.15,0.15}	19.9	31.1	
5	1	10	tansig, purelin	trainbfg	dividerand={0.7,0.15,0.15}	19.9	22.5	
								Precisão Teste: 20,24
1	1	10	tansig, tansig	traincgf	dividerand={0.7,0.15,0.15}	20.3	24.5	Precisão Global: 20,68
2	1	10	tansig, tansig	traincgf	dividerand={0.7,0.15,0.15}	20.3	24.5	
3	1	10	tansig, tansig	traincgf	dividerand={0.7,0.15,0.15}	20.3	20.5	
4	1	10	tansig, tansig	traincgf	dividerand={0.7,0.15,0.15}	20.3	17.8	
								Precisão Teste: 21,56

5	1	10	tansig, tansig	traincgf	dividerand= {0.7,0.15,0.15}	22.2	20.5	
1	1	10	purelin, purelin	traincgp	dividerand= {0.7,0.15,0.15}	20.3	18.5	Precisão Global: 21,56
2	1	10	purelin, purelin	traincgp	dividerand= {0.7,0.15,0.15}	19.9	23.1	
3	1	10	purelin, purelin	traincgp	dividerand= {0.7,0.15,0.15}	19.9	19.8	
4	1	10	purelin, purelin	traincgp	dividerand= {0.7,0.15,0.15}	20.3	20.5	Precisão Teste: 21,28
5	1	10	purelin, purelin	traincgp	dividerand= {0.7,0.15,0.15}	27.4	27.1	

Variar a função de ativação

Nº de execuções	Camadas	Nº de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste	Média
1	1	10	logsig, purelin	trainlm	dividerand= {0.7,0.15,0.15}	91.9	76.8	Precisão Global: 91,4
2	1	10	logsig, purelin	trainlm	dividerand= {0.7,0.15,0.15}	94.2	87.4	
3	1	10	logsig, purelin	trainlm	dividerand= {0.7,0.15,0.15}	85.3	80.1	
4	1	10	logsig, purelin	trainlm	dividerand= {0.7,0.15,0.15}	89.6	76.1	Precisão Teste: 82,22
5	1	10	logsig, purelin	trainlm	dividerand= {0.7,0.15,0.15}	94.2	90.7	
1	1	10	tansig, logsig	trainlm	dividerand= {0.7,0.15,0.15}	91.4	80.1	
2	1	10	tansig, logsig	trainlm	dividerand= {0.7,0.15,0.15}	92.8	80.7	

3	1	10	tansig, logsig	trainlm	dividerand= {0.7,0.15,0.15}	88.5	79.4	Precisão Global: 92,26
4	1	10	tansig, logsig	trainlm	dividerand= {0.7,0.15,0.15}	94.3	82.7	
5	1	10	tansig, logsig	trainlm	dividerand= {0.7,0.15,0.15}	94.3	79.4	
1	1	10	hardlim, purelin	trainlm	dividerand= {0.7,0.15,0.15}	26.8	29.1	Precisão Global: 37,16
2	1	10	hardlim, purelin	trainlm	dividerand= {0.7,0.15,0.15}	40.9	38.4	
3	1	10	hardlim, purelin	trainlm	dividerand= {0.7,0.15,0.15}	36.0	34.4	
4	1	10	hardlim, purelin	trainlm	dividerand= {0.7,0.15,0.15}	39.6	37.7	
5	1	10	hardlim, purelin	trainlm	dividerand= {0.7,0.15,0.15}	42.5	43.0	

Alínea c): Parte 1

Utilizando as imagens da Pasta_3, sem treinar a rede, e usando a melhor configuração resultante dos testes anteriores, obtivemos uma precisão total média de 84,98%.

Configuração utilizada:

4	2	20,20	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	95.5	88.0
---	---	-------	--	--------------------	---	------	------

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Global</i>	88.0	79.4	87.8	85.6	83.4	88.0	86.8	87.0	79.8	84.0	84,98%

Alínea c): Parte 2

TESTES PASTA_1:

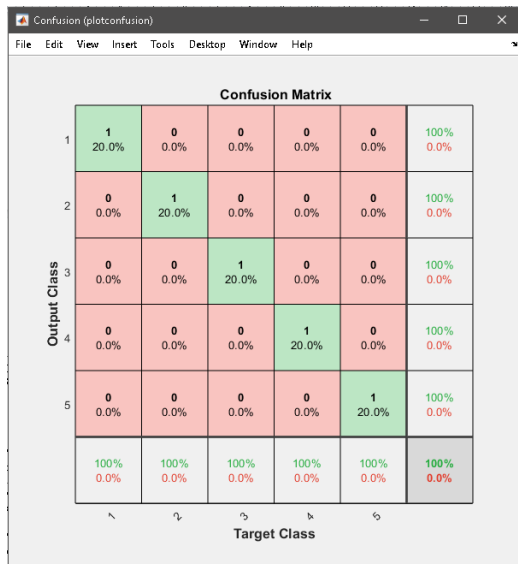
Tipo de rede: feedforwardnet

Treino: Pasta_3

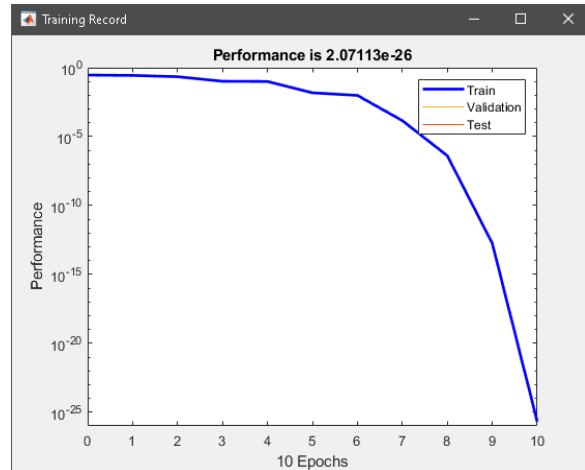
Teste: Pasta_1

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}	

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Teste</i>	100	80	40	40	20	40	80	20	80	80	58%



Melhor matriz de confusão.



Melhor performance.

	Camadas	Nº de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Teste
Configuração	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	

Nº de execuções	1	2	3	4	5	6	7	8	9	10	Média
Precisão Teste	80	40	80	60	40	60	60	60	100	60	64%

Tipo de rede: patternnet

Treino: Pasta_3

Teste: Pasta_1

	Camadas	Nº de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Teste
Configuração	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}	

Nº de execuções 1 2 3 4 5 6 7 8 9 10 Média

Precisão Teste

60 40 40 60 100 100 40 40 40 60 58%

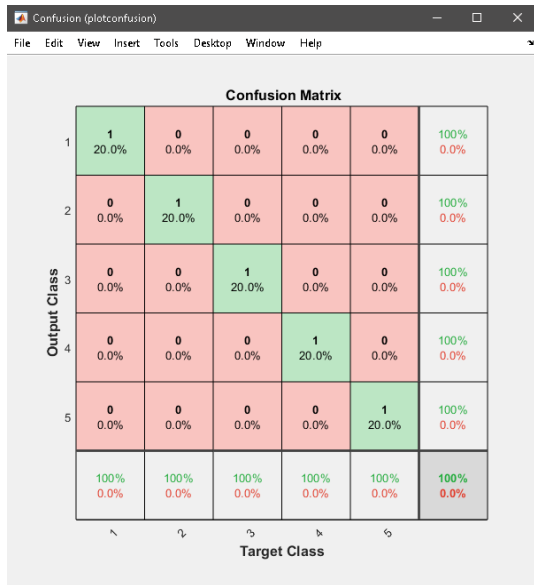


Figura 8. Melhor matriz de confusão

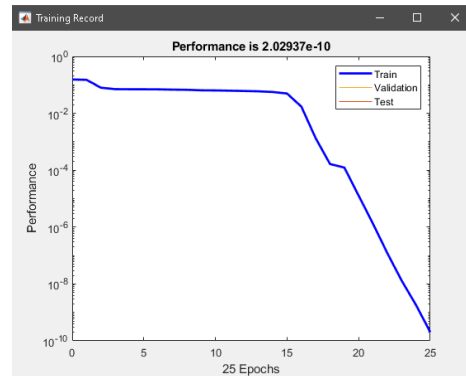


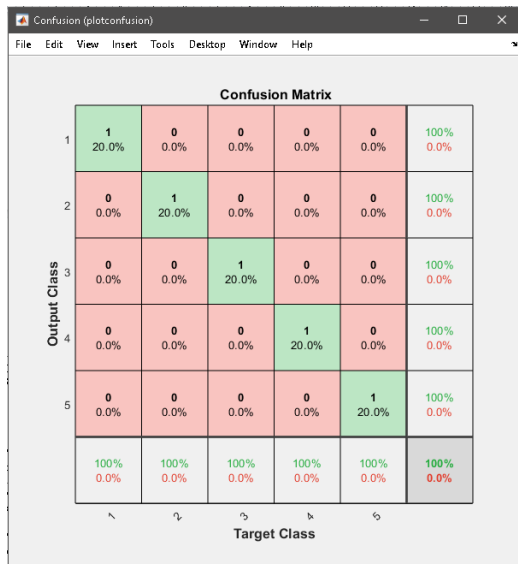
Figura 9. Melhor performance

	Camadas	Nº de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste
Configuração	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {1, 0, 0}		

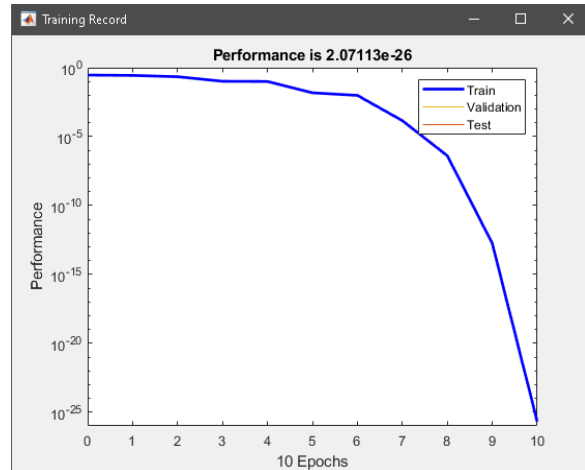
Nº de execuções 1 2 3 4 5 6 7 8 9 10 Média

Precisão Teste

60 60 80 40 40 80 60 40 60 40 56%



Melhor matriz de confusão.



Melhor performance.

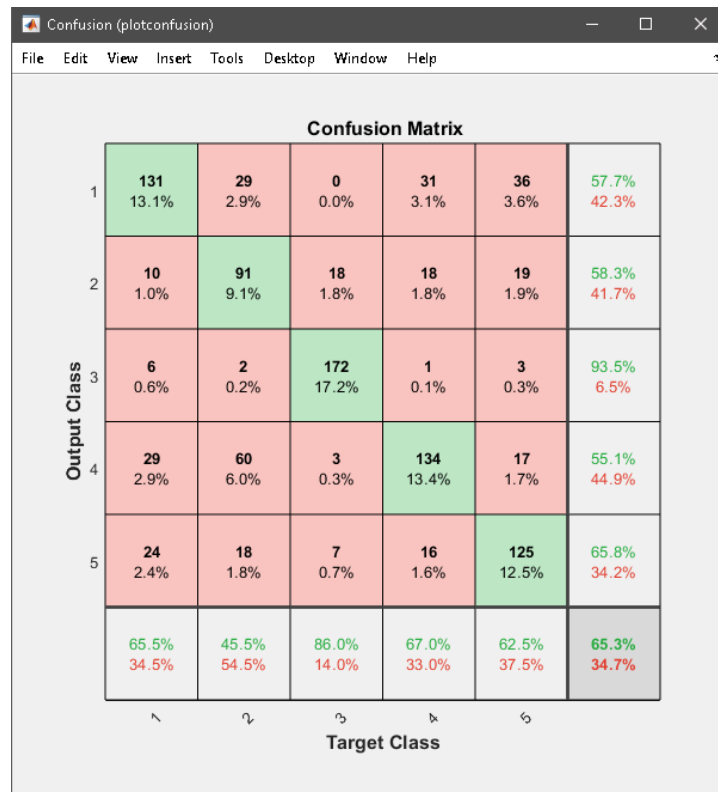
TESTES PASTA_2:

Tipo de rede: feedforwardnet

Treino: Pasta_3

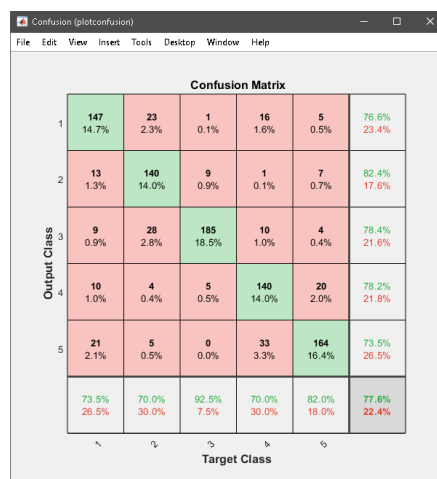
Teste: Pasta_2

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>			<i>Função de treino</i>		<i>Divisão dos exemplos</i>		<i>Precisão Teste</i>	
<i>Configuração</i>	1	10	tansig, purelin			trainlm		dividerand = {1,0,0}			
<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Teste</i>	65.3	72.9	74.3	71.3	81.1	69.0	75.4	77.5	78.3	72.9	73.8%



Matriz exemplar para a precisão de 65,3%

	Camadas	Nº de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Teste					
Configuração	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {1,0,0}						
Nº de execuções	1	2	3	4	5	6	7	8	9	10	Média
Precisão Teste	77.6	78.1	74.3	65.4	75.7	73.2	78.3	79.8	74.3	76.5	75.32%



Matriz exemplar para a precisão de 77,6%

Tipo de rede: patternnet

Treino: Pasta_3

Teste: Pasta_2

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}	

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Teste</i>	71.0	76.2	70.7	71.9	82.4	75.3	74.6	71.2	69.2	79	74,15%

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {1,0,0}	

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Teste</i>	80.2	76.4	70.9	71.3	76.6	74.5	77.2	74.9	75.1	72.1	74,92%

TESTES PASTA_3:

Tipo de rede: feedforwardnet

Treino: Pasta_3

Teste: Pasta_3

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}	

Nº de execuções 1 2 3 4 5 6 7 8 9 10 *Média*

<i>Precisão Teste</i>	100	100	100	100	100	100	100	100	100	100	100	100%
-----------------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	

Nº de execuções 1 2 3 4 5 6 7 8 9 10 *Média*

<i>Precisão Teste</i>	100	100	100	100	99.6	100	99.8	100	100	100	100	99,94%
-----------------------	-----	-----	-----	-----	------	-----	------	-----	-----	-----	-----	--------

Tipo de rede: patternnet

Treino: Pasta_3

Teste: Pasta_3

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}	

Nº de execuções 1 2 3 4 5 6 7 8 9 10 *Média*

<i>Precisão Teste</i>	100	100	99.8	100	100	100	100	100	100	100	99,98%
-----------------------	-----	-----	------	-----	-----	-----	-----	-----	-----	-----	--------

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {1,0,0}	

Nº de execuções 1 2 3 4 5 6 7 8 9 10 *Média*

<i>Precisão Teste</i>	100	100	100	100	99.8	100	79.8	99.4	100	100	97,9%
-----------------------	-----	-----	-----	-----	------	-----	------	------	-----	-----	-------

alínea c): Parte 3

Pasta_1:

Tipo de rede: patternet

Treino: Pasta_1+Pasta_2+Pasta_3

Teste: Pasta_1

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}	

Nº de execuções 1 2 3 4 5 6 7 8 9 10 *Média*

<i>Precisão Teste</i>	100	100	100	100	100	100	100	100	100	100	100%
-----------------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {1,0,0}	

Nº de execuções 1 2 3 4 5 6 7 8 9 10 *Média*

<i>Precisão Teste</i>	100	100	100	100	100	100	100	100	100	100	100%
-----------------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

Tipo de rede: feedforwardnet

Treino: Pasta_1+Pasta_2+Pasta_3

Teste: Pasta_1

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}	

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Teste</i>	100	100	100	100	100	100	100	100	100	100	100%

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Teste</i>	100	100	100	100	100	100	100	100	100	100	100%

Tipo de rede: patternnet

Treino: Pasta_1+Pasta_2+Pasta_3

Teste: Pasta_2

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}	

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
------------------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	--------------

<i>Precisão Teste</i>	92.9	90.6	90.3	95.7	93.4	90.9	94.1	95.0	93.8	91.1	92.78
-----------------------	------	------	------	------	------	------	------	------	------	------	-------

<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
----------------	------------------------	----------------------------	-------------------------	-----------------------------	-----------------------

<i>Configuração</i>	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {1,0,0}
---------------------	---	-----	----------------------------	---------	-------------------------

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
------------------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	--------------

<i>Precisão Teste</i>	95.8	91.6	90.7	97.9	94.4	94.7	92.3	94.3	95.1	91.3	93.81
-----------------------	------	------	------	------	------	------	------	------	------	------	-------

Tipo de rede: feedforwardnet

Treino: Pasta_1+Pasta_2+Pasta_3

Teste: Pasta_2

<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
----------------	------------------------	----------------------------	-------------------------	-----------------------------	-----------------------

<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}
---------------------	---	----	-----------------	---------	-------------------------

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
------------------------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	--------------

<i>Precisão Teste</i>	89.8	95.8	93.3	93.0	94.5	90.3	94.7	92.1	90.8	92.4	92.67
-----------------------	------	------	------	------	------	------	------	------	------	------	-------

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	

Nº de execuções 1 2 3 4 5 6 7 8 9 10 *Média*

<i>Precisão Teste</i>	91.9	89.7	93.3	97.6	97.5	90.3	92.7	95.0	97.1	93.4	93.85
-----------------------	------	------	------	------	------	------	------	------	------	------	-------

Tipo de rede: patternnet

Treino: Pasta_1+Pasta_2+Pasta_3

Teste: Pasta_3

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}	

Nº de execuções 1 2 3 4 5 6 7 8 9 10 *Média*

<i>Precisão Teste</i>	96.0	93.0	93.8	95.6	94.3	95.0	92.8	94.8	94.4	93.2	94.30
-----------------------	------	------	------	------	------	------	------	------	------	------	-------

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {1,0,0}	

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Teste</i>	96.0	94.8	96.6	94.0	94.6	94,3	96,4	94,9	96,2	94,8	95,26

Tipo de rede: feedforward
Treino: Pasta_1+Pasta_2+Pasta_3
Teste: Pasta_3

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}	

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Teste</i>	93.0	94.4	92.1	92.8	95.2	94.1	93.8	92.1	90.7	94.6	93.28

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Teste</i>
<i>Configuração</i>	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}	

<i>Nº de execuções</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Média</i>
<i>Precisão Teste</i>	96.4	95.2	94.4	92.8	95.0	92.8	93.4	95.4	95.9	94.2	94,55%

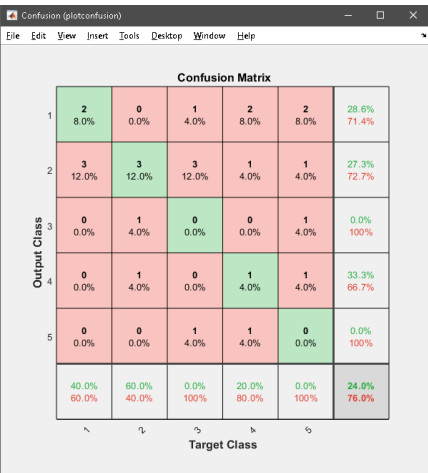
Desenho das Vogais

PATTERNET

	Camadas	Nº de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste
Configuração	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}		

Nº de execuções 1 2 3 4 5 6 7 8 9 10 Média

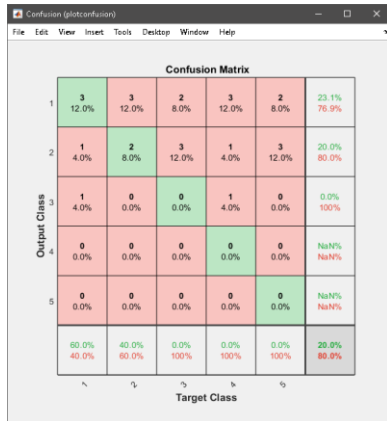
Precisão Global	28.0	28.0	24.0	32.0	24.0	12.0	12.0	28.0	34.0	24.0	24.6
Precisão de treino	NaN										



	Camadas	Nº de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste
Configuração	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {1,0,0}		

Nº de execuções 1 2 3 4 5 6 7 8 9 10 Média

<i>Precisão Global</i>	20.0	24.0	28.0	20.0	32.0	24.0	12.0	24.0	22.0	22.0	22.8
<i>Precisão de treino</i>	NaN										



feedforward

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Global</i>	<i>Precisão Teste</i>
<i>Configuração</i>	1	10	tansig, purelin	trainlm	dividerand = {1,0,0}		

<i>Nº de execuções</i>	1	2	3	4	5	6	7	8	9	10	<i>Média</i>
<i>Precisão Teste</i>	24.0	22.0	28.0	28.0	32.0	12.0	24.0	16.0	18.0	20.0	22.402
<i>Precisão de treino</i>	NaN										

	<i>Camadas</i>	<i>Nº de neurónios</i>	<i>Funções de ativação</i>	<i>Função de treino</i>	<i>Divisão dos exemplos</i>	<i>Precisão Global</i>	<i>Precisão Teste</i>
<i>Configuração</i>	2	5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7,0.15,0.15}		

Nº de execuções 1 2 3 4 5 6 7 8 9 10 *Média*

<i>Precisão Teste</i>	24.0	36.0	16.0	24.0	8.0	28.0	32.0	24.0	28.0	20.0	24.00
<i>Precisão de treino</i>	NaN										

Conclusão

Conclusões para a alínea a): Pasta_1:

Para as diferentes funções de treino testadas, com a mesma configuração (restantes parâmetros em igualdade), verificou-se que a precisão se manteve constante em 100%. Variando o número de neurónios, o número de épocas e o tipo de rede, verificámos que não ocorreu nenhuma alteração, logo, não influencia as funções de ativação. Notámos ainda que, ao testar para número de épocas de treino 5, 10, 50, 100 e 1000, o programa nunca alcançou realmente o valor máximo de épocas de treino, atingindo um valor máximo de 53.

Por sua vez, ao variar o número de neurónios e o número de camadas com diferentes combinações de funções de ativação, obtivemos resultados diferentes de precisão de treino máxima: com a combinação “purelin, tansig” a precisão baixou de 100% para um valor médio de 76%.

Alterando o tipo de rede, mas mantendo a restante configuração padrão inicial - (detalhada no início do capítulo “Testes e Resultados”) – a precisão baixou para um valor médio de 84%. Concluimos então que o melhor tipo de rede para o caso da Pasta_1 é a feedforwardnet.

Conclusões para a alínea b): Pasta_2:

Mantendo os restantes parâmetros em igualdade e alternando o tipo de rede entre patternnet e feedforwardnet, concluimos que a diferença não é significativa para esta amostra uma vez que a precisão global média da primeira foi de 92,35% e a precisão média da segunda foi de 92,2%.

Concluimos que a função de treino mais adequada à situação é a trainml uma vez que para as restantes – traincgf, traincgp, trainbfg e traingd – a precisão rondou os 20% e na trainml, usada nos testes iniciais, passou os 90%. Assim, os testes para as restantes variáveis mantiveram constantes o tipo de rede patternnet e a função de treino trainml.

Comparando os resultados obtidos ao variar o número de neurónios em 10, 20 e 50, notámos que a melhor precisão obtida foi na configuração com duas camadas com

número de neurónios configurado em 20,20. Em relação ao número de camadas, observámos que para um mesmo número de neurónios, numa única camada, a precisão foi ligeiramente superior à registada para duas camadas. A primeira, rondou os 92% e, a segunda, os 90%.

Por fim, a conclusão sobre a função de ativação (ou combinação de funções) mais adequada foi que esta seria a que combina tansig e purelin, atingindo na sua melhor configuração os 94,74%. Em contrapartida, observámos que a combinação mais desadequada foi a que combinou handlim com purelin.

Conclusões para a alínea c):

Parte 1: Partindo de um treino anterior, sem treinar novamente a rede para testar os exemplos contidos na Pasta_3, e usando a melhor configuração obtida da alínea anterior para a Pasta_2 (precisão média de 10 execuções = 94,74%), era previsível que a precisão baixasse, o que se verificou. O valor obtido foi de 84,98% que, não sendo perfeito ou sequer ideal, pode ser considerado razoavelmente aceitável.

Parte 2: Os melhores resultados obtidos foram observados para a pasta_3 como seria de esperar, uma vez que o treino da rede foi feito precisamente para essa pasta. Assim, a precisão atingiu os 100% quando usado o tipo de rede feedforward para uma camada com 10 neurónios. No entanto, para a pasta_3, com outras configurações, os valores foram sempre superiores a 99%. Para a pasta_1, o valor de precisão máximo atingido foi de apenas 64% e, para a pasta_2, 75%.

Parte 3: Nos primeiros testes a precisão de 100% foi facilmente atingida e seria de esperar que esse valor se mantivesse constante (ou próximo disso) nos restantes testes. Isso não se verificou porque os nossos testes tiveram apenas 20 épocas como default. Se esse valor fosse superior, a precisão seria a esperada.