



Trabalho Prático

*Redes Neurais*

*Relatório*

*Unidade Curricular: Conhecimento e Raciocínio*

ANO LETIVO 2024/2025

Ana Luísa Silva Guedes - 2023141564

Gabriel Matos Dias - 2023133076

## Índice

Introdução.....	3
Decisões tomadas para a implementação.....	4
Pré-processamento das imagens.....	4
Targets.....	4
Dados Guardados .....	4
Treino e estudos estatísticos .....	5
Alínea A .....	5
Alínea B .....	6
Análise das camadas escondidas .....	7
Análise da Função de Treino.....	8
Análise da Função de Ativação.....	9
Análise da divisão de exemplos pelos conjuntos .....	10
Junção das melhores configurações.....	11
Alínea C .....	12
Tarefa i) .....	12
Tarefa ii).....	13
Tarefa iii).....	14
Tarefa iv).....	14
Alínea D .....	15
Aplicação .....	16
Conclusão.....	18
Bibliografia .....	19

## Introdução

As redes neuronais artificiais têm desempenhado um papel fundamental no avanço da inteligência artificial e no desenvolvimento de sistemas inteligentes capazes de aprender e tomar decisões a partir de dados. Entre os diversos tipos de redes neuronais, as redes feedforward destacam-se pela sua simplicidade e eficácia em tarefas de classificação e regressão.

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, paralelogramo, quadrado, trapézio e triângulo) com recurso a uma rede neuronal do tipo feedforward.

# Decisões tomadas para a implementação

## Pré-processamento das imagens

Para uma otimização da aplicação, foi realizado um redimensionamento das imagens para uma resolução de 32x32 pixels, utilizando funções da Image Processing Toolbox do MATLAB.

As imagens foram convertidas para tons de cinzento, caso estivessem a cores, e posteriormente binarizadas, resultando numa representação onde os pixels pretos assumem o valor 0 e os brancos o valor 1 (de acordo com o comportamento padrão da função `imbinarize`).

Após a binarização, cada imagem foi transformada num vetor coluna e inserida na matriz de dados de entrada. Em paralelo, foi criada uma matriz de dados de saída (alvo), com codificação *one-hot*, representando a classe de cada imagem.

## Targets

Para cada imagem processada, foi gerado um vetor de saída (*target*) utilizando codificação *one-hot*, em que cada classe é representada por um vetor binário com dimensão 6x1. Este vetor contém o valor 1 na posição correspondente à classe da imagem e 0 nas restantes, permitindo assim treinar a rede neuronal para distinguir corretamente entre as seis classes geométricas distintas.

## Dados Guardados

Após o treino da rede neuronal, é realizada uma avaliação da sua precisão tanto no conjunto total de dados como no subconjunto de teste. Caso a precisão global obtida seja superior a 93%, a rede neuronal treinada é automaticamente guardada num ficheiro.mat, cujo nome inclui a percentagem de precisão global e de teste, facilitando a identificação do seu desempenho. Este critério dos 93% foi definido por nós como referência para selecionar redes com desempenho elevado. A partir dos ficheiros guardados, é feita manualmente uma seleção das três melhores redes, eliminando-se as restantes que não se encontram no top 3.

# Treino e estudos estatísticos

## Alínea A

A rede neuronal utilizada é uma rede feedforward, onde os dados de entrada são processados através de uma ou mais camadas ocultas antes de chegar à camada de saída. No caso da implementação, foi definida uma rede com uma camada escondida contendo 10 neurónios, conforme recomendado no enunciado.

A função de ativação da camada escondida foi definida como tansig, uma função sigmoide que mapeia os valores de entrada para o intervalo entre -1 e 1. Na camada de saída, foi utilizada a função purelin, que é linear, adequada para tarefas de classificação multiclasse.

```
net = feedforwardnet(10, 'trainlm');           % x camada escondida com y neurónios
net.layers{1}.transferFcn = 'tansig';          % Função de ativação da camada escondida
net.layers{2}.transferFcn = 'purelin';         % Função de ativação da camada de saída

net.divideFcn = '';

net.trainParam.epochs = 100;                  % Número máximo de épocas
```

A função de divisão de dados (divideFcn) foi desativada, de forma que todo o conjunto de dados seja usado exclusivamente para treino. Esta opção foi adotada dada a reduzida dimensão do conjunto de imagens.

O número de épocas, ou iterações de treino, não foi especificado no enunciado, sendo uma escolha definida no código. No entanto, foi estabelecido um número máximo de 100 épocas para garantir que a rede tivesse tempo suficiente para aprender e ajustar seus pesos durante o treino.

Em várias execuções, a precisão da rede foi quase sempre de 100%, demonstrando que a rede conseguiu aprender a classificar corretamente as imagens geométricas.

Após o treino, foi realizada a avaliação global do desempenho da rede utilizando a função plotconfusion. Esta função gera uma matriz de confusão, que é uma ferramenta fundamental para analisar o desempenho da rede em tarefas de classificação. A matriz de confusão permite visualizar as classificações corretas e incorretas da rede para cada classe.

Além disso, foi calculada a precisão global da rede utilizando a fórmula:

$$\text{Precisão} = \frac{\text{Número de acertos}}{\text{Número total de imagens}} * 100$$

A precisão global foi calculada através da contagem dos acertos totais e da sua proporção em relação ao número de amostras.

Os resultados mostram que a rede feedforward, com a configuração inicial proposta e utilizando a função de treino trainlm, obteve uma precisão quase sempre de 100%. Este desempenho indica que a rede foi bem-sucedida em aprender as características das imagens geométricas e realizar a classificação corretamente.

## Alínea B

Para tentar obter os melhores resultados na classificação das imagens fornecidas, foram testadas várias funções e parâmetros diferentes. Podendo assim comparar com a configuração inicial.

A configuração inicial:

	Número de épocas usadas	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Média Precisão Global	Média Precisão Teste
<b>Configuração por defeito</b>	100	1	10	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	91	71,5

  

Rep. 1	Rep. 2	Rep. 3	Rep. 4	Rep. 5	Rep. 6	Rep. 7	Rep. 8	Rep. 9	Rep. 10
Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste
92	73	92	69	92	71	92	76	92	73
82	53	93	69	92	82	93	87	90	62

Decidimos realizar 10 repetições por cada configuração de rede neuronal, com o objetivo de obter valores médios mais fiáveis e representativos. Como o processo envolve divisão aleatória dos dados (treino, validação e teste), uma única execução poderia fornecer resultados enviesados ou não generalizáveis. Ao repetir o treino e teste várias vezes, conseguimos reduzir o impacto da aleatoriedade e observar o comportamento médio e a variabilidade do modelo, garantindo uma avaliação mais rigorosa do seu desempenho.

## Análise das camadas escondidas

### O número e dimensão das camadas escondidas influencia o desempenho?

Conf1	100	1	20	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	89,8	64,8
Conf2	100	2	10, 10	tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	89,0	69,1
Conf3	100	2	5, 5	tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	83,5	65,2
Conf4	100	3	10, 10, 10	tansig, tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	90,4	72,1

Rep. 1		Rep. 2		Rep. 3		Rep. 4		Rep. 5		Rep. 6		Rep. 7		Rep. 8		Rep. 9		Rep. 10	
Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste
91	67	89	69	88	62	90	60	88	60	92	69	90	71	90	67	91	67	89	56
91	62	71	53	93	76	92	69	95	84	92	76	92	67	90	67	83	64	91	73
67	51	83	67	76	56	91	71	82	62	88	64	88	67	84	73	87	72	89	69
86	71	92	73	91	69	92	76	87	64	87	62	92	69	91	71	94	84	92	82

Os resultados mostram que o número e a dimensão das camadas escondidas influenciam o desempenho da rede neuronal. A configuração inicial, com uma única camada de 10 neurónios, apresentou bons resultados (71,5% de precisão média no teste). No entanto, quando se aumentou o número de neurónios para 20 (Configuração 1), a precisão de teste diminuiu, o que indica sobreajuste. Configurações mais profundas, como a Configuração 2 e Configuração 3, tiveram desempenhos variados, dependendo do número de neurónios. A Configuração 4, com três camadas de 10 neurónios cada, foi a que melhor generalizou (72,1% de precisão média no teste). Conclui-se que a arquitetura da rede tem impacto direto no desempenho e que o equilíbrio entre profundidade e dimensão é fundamental para obter bons resultados.

## Análise da Função de Treino

A função de treino influencia o desempenho?																			
Conf1		100		1		10		tansig, purelin		traingdx		dividerand = {0.7, 0.15, 0.15}		57,4		45,5			
Conf2		100		1		10		tansig, purelin		trainscg		dividerand = {0.7, 0.15, 0.15}		64,2		53,2			
Conf3		100		1		10		tansig, purelin		trainbr		dividerand = {0.7, 0.15, 0.15}		96,9		79,4			
Conf4		100		1		10		tansig, purelin		trainr		dividerand = {0.7, 0.15, 0.15}		75,7		59,7			
Rep. 1		Rep. 2		Rep. 3		Rep. 4		Rep. 5		Rep. 6		Rep. 7		Rep. 8		Rep. 9		Rep. 10	
Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste
21	31	58	47	56	44	56	44	58	40	20	18	65	51	78	60	85	62	77	58
88	76	66	56	80	58	47	49	75	60	68	60	60	51	86	69	36	22	36	31
99	93	96	73	97	78	96	78	96	73	96	76	98	76	98	89	96	76	97	82
90	64	84	67	86	78	71	49	67	53	64	60	62	53	71	62	76	51	86	60

Os resultados demonstram inequivocamente que a função de treino tem um impacto significativo no desempenho do sistema de classificação. O algoritmo trainbr revela-se como a escolha mais adequada para este problema específico, proporcionando não só a melhor precisão global como também a melhor capacidade de generalização, evidenciada pelos valores superiores de precisão no conjunto de teste.

Esta análise sugere que, em aplicações práticas de reconhecimento de formas geométricas utilizando redes neuronais, a seleção criteriosa do algoritmo de treino pode ser um fator determinante para o sucesso do sistema.



## Análise da Função de Ativação

As funções de ativação influenciam o desempenho?																			
Conf1		100		1		10		logsig, purelin		trainlm		dividerand = {0.7, 0.15, 0.15}		89,4		66,4			
Conf2		100		1		10		tansig, logsig		trainlm		dividerand = {0.7, 0.15, 0.15}		58,0		50,6			
Conf3		100		1		10		tansig, softmax		trainlm		dividerand = {0.7, 0.15, 0.15}		89,2		69,7			
Conf4		100		1		10		softmax, purelin		trainlm		dividerand = {0.7, 0.15, 0.15}		62,2		49,0			
Conf5		100		1		10		logsig, softmax		trainlm		dividerand = {0.7, 0.15, 0.15}		88,3		69,2			
Rep. 1		Rep. 2		Rep. 3		Rep. 4		Rep. 5		Rep. 6		Rep. 7		Rep. 8		Rep. 9		Rep. 10	
Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste
87	60	90	62	91	69	85	69	92	71	90	69	89	62	93	73	86	60	91	69
68	53	39	40	64	56	59	49	88	76	37	44	58	42	66	58	24	24	77	64
90	69	92	87	91	78	89	64	87	67	90	73	83	64	94	73	84	53	92	69
43	38	74	53	76	69	75	67	57	38	63	36	66	56	38	29	58	44	72	60
91	76	84	56	88	60	93	73	86	58	92	78	86	78	91	73	84	71	88	69

A análise dos resultados demonstra claramente que as funções de ativação influenciam significativamente o desempenho do sistema de classificação.

Após analisar os resultados, decidimos que a configuração 1 (logsig e purelin) é a melhor, pois obteve a maior precisão global (89,4%) e uma precisão de teste sólida (66,4%). No entanto, a configuração com a função de ativação softmax, como na configuração 3 (tansig e softmax) e na configuração 5 (logsig e softmax), mostrou um desempenho muito próximo, com precisões de teste de 69,7% e 69,2%, respetivamente. Isso indica que, embora a configuração 1 tenha alcançado melhores resultados gerais, a softmax oferece um desempenho competitivo, ficando apenas ligeiramente atrás, e pode ser uma boa alternativa dependendo do objetivo específico do sistema.

## Análise da divisão de exemplos pelos conjuntos

A divisão de exemplos pelos conjuntos influencia o desempenho?								
Conf1	100	1	10	tansig, purelin	trainlm	dividerand = {0.4, 0.35, 0.25}	78,1	63,9
Conf2	100	1	10	tansig, purelin	trainlm	dividerand = {0.33, 0.33, 0.33}	74,6	62,5
Conf3	100	1	10	tansig, purelin	trainlm	dividerand = {0.9, 0.05, 0.05}	93,3	68,6
Conf4	100	1	10	tansig, purelin	trainlm	dividerand = {0.2, 0.5, 0.3}	63,5	52,7
Conf5	100	1	10	tansig, purelin	trainlm	dividerand = {0.8, 0.1, 0.1}	93,6	70,4

Rep. 1		Rep. 2		Rep. 3		Rep. 4		Rep. 5		Rep. 6		Rep. 7		Rep. 8		Rep. 9		Rep. 10	
Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste
81	69	80	63	79	60	69	53	78	61	79	68	80	69	76	67	78	64	81	65
76	63	74	64	77	64	77	62	74	63	75	64	72	62	75	62	73	64	73	57
97	67	98	73	97	60	71	67	93	73	94	73	97	73	96	67	97	73	93	60
63	54	62	51	62	48	66	50	60	48	65	52	65	53	67	61	60	54	65	56
89	50	96	83	93	77	95	70	95	57	95	70	95	80	91	67	94	70	93	80

A análise da divisão de exemplos pelos conjuntos de treino, validação e teste revela um impacto crucial deste parâmetro no desempenho do sistema. As configurações que privilegiam uma maior proporção de dados para treino (80% a 90%) apresentam resultados significativamente superiores, com precisões globais acima de 93% e precisões de teste acima de 68%. Em particular, a configuração com divisão [0.8, 0.1, 0.1] proporciona o melhor equilíbrio entre aprendizagem e generalização.

Estes resultados indicam claramente que, para sistemas de classificação de formas geométricas com conjuntos de dados limitados, é fundamental alocar uma proporção substancial dos dados para o processo de aprendizagem, mantendo simultaneamente um equilíbrio adequado entre os conjuntos de validação e teste. Configurações com percentagens reduzidas para treino (como 20% ou 40%) comprometem significativamente o desempenho global do sistema. Esta análise fornece orientações práticas valiosas para a implementação de sistemas semelhantes, sugerindo que a divisão [0.8, 0.1, 0.1] deve ser considerada como referência para a distribuição ótima dos dados.

## Junção das melhores configurações

Junção das melhores configurações para obter a melhor precisão																	
Conf1	100		3	10, 10, 10		logsig, purelin		trainlm		dividerand = {0.8, 0.1, 0.1}		92,2	70,9				
Rep. 1	Rep. 2		Rep. 3		Rep. 4		Rep. 5		Rep. 6		Rep. 7		Rep. 8		Rep. 9		Rep. 10
Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste	Precisão Global	Precisão Teste
93	53	84	60	94	67	89	70	89	70	92	67	96	83	96	83	93	63
38	27																

A junção das configurações otimizadas demonstra que a combinação dos parâmetros identificados como ótimos nas análises individuais resulta efetivamente num sistema de classificação de elevado desempenho. A configuração que incorpora três camadas escondidas (10, 10, 10), funções de ativação logsig e purelin, algoritmo de treino trainlm e divisão [0.8, 0.1, 0.1] alcança uma impressionante precisão global de 92,2% e precisão de teste de 70,9%.

Apesar de a nossa melhor função de treino ser a trainbr, decidimos utilizar a trainlm, pois, ao combinar a trainbr com as outras configurações ideais, a média da precisão global foi de apenas 38% e a da precisão de teste de 27%, o que representou um desempenho muito baixo. Isso pode ter acontecido porque a trainbr tende a se ajustar demasiado aos dados de treino, o que faz com que o modelo tenha dificuldade em generalizar para os dados de teste. Por isso, optámos pela trainlm, que teve um desempenho melhor.

Esses resultados mostram que a otimização paramétrica foi eficaz e que, mesmo com a rede mais complexa (com várias camadas escondidas), os benefícios de desempenho valem a pena. A consistência dos resultados ao longo das repetições também mostra que a configuração é robusta. Para sistemas práticos de reconhecimento de formas geométricas, esta configuração é uma boa base, pois combina as vantagens de cada componente analisado.

As nossas 3 melhores redes foram testadas com a função de ativação trainbr, mas as precisões obtidas foram bastante baixas. Por esse motivo decidimos usar outra função de ativação para a alínea seguinte.

Nas imagens abaixo é possível ver a matriz de confusão das 3 melhores redes treinadas com a função de ativação trainlm, estas serão usadas no treino e testes da alínea seguinte.

Matriz de Confusão - Total

1	48 16.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.7%	96.0% 4.0%
2	0 0.0%	48 16.0%	0 0.0%	0 0.0%	0 0.0%	2 0.7%	96.0% 4.0%
3	0 0.0%	1 0.3%	50 16.7%	2 0.7%	0 0.0%	2 0.7%	90.9% 9.1%
4	1 0.3%	0 0.0%	0 0.0%	48 16.0%	0 0.0%	1 0.3%	96.0% 4.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 16.7%	0 0.0%	100% 0.0%
6	1 0.3%	1 0.3%	0 0.0%	0 0.0%	0 0.0%	43 14.3%	95.6% 4.4%
	96.0% 4.0%	96.0% 4.0%	100% 0.0%	96.0% 4.0%	100% 0.0%	86.0% 14.0%	95.7% 4.3%
	1	2	3	4	5	6	

Output Class

Target Class

Matriz de Confusão - Total

1	48 16.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	98.0% 2.0%
2	0 0.0%	49 16.3%	0 0.0%	1 0.3%	0 0.0%	0 0.0%	98.0% 2.0%
3	0 0.0%	0 0.0%	50 16.7%	1 0.3%	1 0.3%	2 0.7%	92.6% 7.4%
4	2 0.7%	0 0.0%	0 0.0%	48 16.0%	0 0.0%	0 0.0%	96.0% 4.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	49 16.3%	1 0.3%	98.0% 2.0%
6	0 0.0%	1 0.3%	0 0.0%	0 0.0%	0 0.0%	46 15.3%	97.9% 2.1%
	96.0% 4.0%	98.0% 2.0%	100% 0.0%	96.0% 4.0%	98.0% 2.0%	92.0% 8.0%	96.7% 3.3%
	1	2	3	4	5	6	

Output Class

Target Class

Matriz de Confusão - Total

1	50 16.7%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	2 0.7%	94.3% 5.7%
2	0 0.0%	50 16.7%	2 0.7%	0 0.0%	0 0.0%	1 0.3%	94.3% 5.7%
3	0 0.0%	0 0.0%	48 16.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	50 16.7%	0 0.0%	1 0.3%	98.0% 2.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	49 16.3%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	46 15.3%	100% 0.0%
	100% 0.0%	100% 0.0%	96.0% 4.0%	100% 0.0%	98.0% 2.0%	92.0% 8.0%	97.7% 2.3%
	1	2	3	4	5	6	

Output Class

Target Class

## Alínea C

### Tarefa i)

Neste passo, o objetivo foi testar as redes neuronais que apresentaram melhor desempenho na alínea b), mas agora utilizando apenas as imagens da pasta test, sem qualquer novo treino. Carregámos as três redes previamente guardadas e avaliámo-las com os dados de teste, registando a precisão de classificação e gerando as respetivas matrizes de confusão.

Os resultados mostraram que todas as redes mantiveram um desempenho menos satisfatório, com precisões da pasta test inferiores às registadas anteriormente:

```
--- c.i) Avaliar redes de b) com imagens da pasta test ---  
  
Rede rede_com_96_global_83_teste.mat  
Precisão em test: 70.00%  
  
Rede rede_com_97_global_73_teste.mat  
Precisão em test: 73.33%  
  
Rede rede_com_98_global_73_teste.mat  
Precisão em test: 73.33%
```

Estes resultados indicam que as redes apresentaram níveis variados de generalização face a novos dados, sendo que se verificou uma ligeira descida no desempenho. Uma possível explicação para esta diminuição prende-se com o facto de os dados da pasta test poderem não estar tão bem distribuídos ou representativos para estas redes, em comparação com o conjunto de validação utilizado durante o treino.

## Tarefa ii)

Neste passo da alínea c.ii), o objetivo foi treinar novamente as melhores redes da alínea b), mas agora usando apenas as imagens da pasta test como dados de treino. Após o treino, cada rede foi testada com imagens das pastas start, train e test, de forma a avaliar a sua capacidade de generalização para diferentes conjuntos de dados. Para cada rede, registámos as precisões obtidas em cada pasta e gerámos também as respetivas matrizes de confusão.

Os resultados foram os seguintes:

```
--- c.ii) Treinar redes com imagens da pasta test/ ---  
  
Treinar nova rede com dados de test...  
Precisões: start=33.33% | train=47.00% | test=80.00%  
  
Treinar nova rede com dados de test...  
Precisões: start=36.67% | train=42.00% | test=81.67%  
  
Treinar nova rede com dados de test...  
Precisões: start=30.00% | train=45.33% | test=80.00%
```

Observa-se que todas as redes obtiveram bons resultados ao classificar imagens da própria pasta test, com precisões entre 80% e 81,67%, o que é esperado por terem sido treinadas com essas imagens. No entanto, a precisão nas pastas start e train foi consideravelmente mais baixa, especialmente no caso da Rede 3, que obteve apenas 30% na pasta start, indicando fraca generalização.

Estas diferenças de desempenho sugerem que, embora o treino com dados da pasta test produza bons resultados nesse mesmo conjunto, ele não garante uma boa generalização para outros conjuntos de dados. Isso pode dever-se à limitação da diversidade das imagens da pasta test, que não representa totalmente as variações presentes nas outras pastas.

As matrizes de confusão permitiram identificar que formas como o círculo e o quadrado foram geralmente bem classificadas, enquanto formas como o paralelogramo e o trapézio apresentaram mais confusões com outras classes, refletindo as suas maiores semelhanças visuais com outras figuras.

Concluimos, assim, que o desempenho de uma rede treinada num único conjunto pode ser enganador e que é importante treinar com dados variados para garantir uma boa capacidade de generalização.

### Tarefa iii)

Na alínea c.iii), treinámos novamente as melhores redes da alínea b), desta vez utilizando todas as imagens disponíveis — ou seja, das pastas start, train e test. O objetivo foi criar redes com uma base de treino mais completa e diversa, e depois avaliar o seu desempenho em cada uma das três pastas separadamente.

Os resultados obtidos foram bastante positivos:

```
--- c.iii) Treinar com TODAS as imagens ---  
  
Treinar rede completa com todos os dados...  
Todas: start=80.00% | train=92.33% | test=95.00%  
  
Treinar rede completa com todos os dados...  
Todas: start=93.33% | train=92.67% | test=86.67%  
  
Treinar rede completa com todos os dados...  
Todas: start=93.33% | train=91.67% | test=96.67%
```

Estas redes apresentaram precisões elevadas e consistentes nos três conjuntos de dados, demonstrando que o treino com todas as imagens aumentou significativamente a capacidade de generalização. As diferenças entre as redes são pequenas, e todas foram capazes de classificar corretamente a maioria das imagens, independentemente da sua origem.

As matrizes de confusão mostraram que as formas geométricas como o círculo e o quadrado continuaram a ser bem reconhecidas, enquanto figuras como o paralelogramo ou o trapézio ainda apresentaram algumas confusões com classes semelhantes.

Em conclusão, treinar com o conjunto completo de imagens revelou-se a abordagem mais eficaz, permitindo obter redes robustas e bem generalizadas para o problema de reconhecimento de formas.

### Tarefa iv)

Na alínea c.iv), o objetivo foi simplesmente guardar as três melhores redes obtidas durante a experimentação da alínea c.iii), em que cada rede foi treinada com todas as imagens disponíveis (das pastas start, train e test).

As redes representam as versões finais mais robustas, resultantes de um treino com dados diversificados e completos, e estão agora prontas para serem reutilizadas ou integradas em sistemas de classificação de formas geométricas. Esta etapa finaliza a experimentação garantindo que os melhores modelos ficam guardados para futuras utilizações.

```
--- c.iv) As 3 melhores redes foram treinadas e guardadas com prefixo rede_c)iii)*.mat ---
```

## Alínea D

Na última experiência (alínea d), foram testadas as três melhores redes neuronais treinadas previamente, utilizando imagens desenhadas manualmente que representavam cada uma das seis formas geométricas. Estas imagens, embora semelhantes às usadas durante o treino, eram inéditas para as redes e permitiram avaliar a sua capacidade de generalização.

Os resultados mostraram precisões globais bastante reduzidas: 13,33% para a primeira, para a segunda de 23,33% e apenas 6,67% para a terceira. Cada rede teve desempenho positivo apenas numa ou duas classes — por exemplo, a primeira identificou corretamente os trapézio, enquanto a segunda classificou quase todos paralelogramas. No entanto, todas as outras classes foram praticamente ignoradas. Estes resultados indicam que as redes, apesar do bom desempenho em dados previamente vistos, apresentam dificuldade em lidar com variações não treinadas, evidenciando uma possível sobre adaptação (overfitting) aos dados originais. Este comportamento destaca a importância de utilizar conjuntos de dados mais variados e diversificados durante o treino, de modo a aumentar a robustez das redes em cenários reais e não supervisionados.

```
---REDE: rede_c)iii)_1.mat---
Precisão GLOBAL (total): 13.33% (4/30)

Precisão por classe:
circle      : 0.00% (0/5)
kite        : 0.00% (0/5)
parallelogram : 0.00% (0/5)
square      : 0.00% (0/5)
trapezoid   : 80.00% (4/5)
triangle    : 0.00% (0/5)

---REDE: rede_c)iii)_2.mat---
Precisão GLOBAL (total): 23.33% (7/30)

Precisão por classe:
circle      : 40.00% (2/5)
kite        : 0.00% (0/5)
parallelogram : 60.00% (3/5)
square      : 0.00% (0/5)
trapezoid   : 20.00% (1/5)
triangle    : 20.00% (1/5)

---REDE: rede_c)iii)_3.mat---
Precisão GLOBAL (total): 6.67% (2/30)

Precisão por classe:
circle      : 0.00% (0/5)
kite        : 0.00% (0/5)
parallelogram : 0.00% (0/5)
square      : 0.00% (0/5)
trapezoid   : 40.00% (2/5)
triangle    : 0.00% (0/5)
```

## Aplicação

Aplicação Gráfica de Treino de Redes

Sair Relatório Autores

Treinar Rede Carregar Rede Criar Imagem

Função de Treino:  Perc. Treino:

Função de Ativação:  Perc. Teste:

Func. Ativação Saída:  Perc. Validação:

Número de Neurónios:  Pasta:

Número de Épocas:  ☐ Deseja guardar a rede?

Nome da Rede:

O trabalho inclui também uma interface gráfica, que permite treinar, carregar e guardar redes neuronais, assim como simular esta com um conjunto de figuras geométricas.

Esta está preparada para validar todos os dados inseridos de modo a mostrar mensagens de erro ao utilizador e impedir comportamentos inesperados.

Optámos por uma interface simples e fácil de navegar, em que no primeiro painel, “Treinar Rede”, é onde é feito o treino da rede, onde o utilizador tem oportunidade de escolher todos os parâmetros para configurar a rede neuronal e no final pode optar por guardar a rede.

Aplicação Gráfica de Treino de Redes

Sair Relatório Autores

Treinar Rede Carregar Rede Criar Imagem

Pasta:

☐ Deseja guardar os resultados?

Nome:

Resultados:

Rede utilizada: rede\_com\_98\_global\_73\_teste.mat  
Pasta analisada: train  
Total de imagens: 300  
Precisão: 97.67%

No segundo painel “Carregar Rede”, o utilizador pode usar uma rede criada e treinada para percorrer todas as figuras geométricas da pasta. Assim que finalizar, mostra os seus resultados, a pasta que foi analisada, o número total de imagens e a percentagem de imagens que a rede conseguiu reconhecer.





No terceiro painel “Criar Imagem”, o utilizador carrega uma rede e classifica uma imagem escolhida por ele. A imagem pode pertencer à pasta das figuras geométricas facultadas para este trabalho, ou pode ser o utilizador a desenhar a imagem no UI axes fornecido. No final, será apresentada a classe prevista que a rede atribui à imagem.

Achamos relevante para a App, termos também um submenu onde guia o utilizador para o Relatório desempenhado para este trabalho e uma breve apresentação dos autores do mesmo.

## Conclusão

Ao longo deste trabalho foi possível explorar e compreender, de forma prática, o funcionamento e o desempenho de redes neuronais na tarefa de reconhecimento de formas geométricas. Foram testadas diversas configurações, avaliando o impacto de parâmetros como o número de camadas escondidas, neurónios, funções de ativação e função de treino. Os resultados demonstraram que a escolha adequada destes parâmetros influencia diretamente a precisão da rede, tanto em treino como em teste.

Durante a experimentação, observou-se que redes treinadas com conjuntos de dados completos (start, train e test) apresentaram os melhores desempenhos, com precisões superiores a 90% em alguns casos. No entanto, quando testadas com imagens desenhadas manualmente, as redes demonstraram fraca capacidade de generalização, revelando limitações face a dados diferentes dos usados no treino.

Conclui-se, assim, que embora as redes possam atingir elevados níveis de acerto em cenários controlados, é essencial garantir diversidade e representatividade no conjunto de dados de treino. Só dessa forma será possível desenvolver modelos robustos e eficazes em ambientes reais, onde as variações nos dados são inevitáveis. Este trabalho reforça também a importância da validação cruzada e da análise crítica dos resultados obtidos, não apenas em termos quantitativos, mas também qualitativos.

## Bibliografia

Foram utilizados para a realização deste trabalho os seguintes sites:

<https://moodle.isec.pt/moodle> - Slides das Teóricas, e fichas práticas

<https://www.mathworks.com/help/matlab/> - Informações sobre funções MatLab