



# Instituto Superior de Engenharia

Politécnico de Coimbra

DEPARTAMENTO DE / DEPARTMENT OF  
INFORMÁTICA E SISTEMAS

## Licenciatura de Engenharia Informática

Conhecimento e Raciocínio - 2024/2025

### **Classificação de Formas Geométricas Usando Redes Neuronais *Feedforward* em MATLAB**

Autores / Authors

**Diogo Silva**                      **a2023139070**

**Paulo Soares**                    **a2023146252**



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra



# ÍNDICE

1. Introdução .....	3
2. Redes Neurais Feedforward.....	4
3. Estrutura das Tarefas.....	4
4. a) Classificação com Imagens da Pasta start.....	5
4.1 Pré-processamento das Imagens .....	5
4.2 Rede com uma camada oculta de 10 neurónios .....	5
4.3 Testes com diferentes topologias .....	5
4.4 Conclusões da Tarefa a) .....	6
5. b) Classificação com Imagens da Pasta train.....	6
5.1 Estratégia e Configurações Testadas .....	6
5.2 Análise de Resultados .....	7
6. c) Estudo com Conjuntos Completos de Imagens.....	8
6.1 Aplicação das Melhores Redes ao Conjunto test (sem novo treino).....	8
6.1.1 Análise de resultados.....	8
6.2 Treino com o Conjunto test e Avaliação nas Três Pastas.....	9
6.2.1 Análise de resultados.....	9
6.3 Treino com o Conjunto Completo de Imagens (start + train + test) .....	10
6.3.1 Análise de resultados.....	11
7. d) Avaliação com Imagens Desenhadas Manualmente.....	12
7.1 Análise de resultados .....	12
8. E) Aplicação Gráfica desenvolvida em MATLAB .....	13
8.1 Funcionalidades Implementadas .....	16
9. Conclusão .....	19

## Índice de Figuras

Figura 1 - Formas feitas manualmente.....	12
Figura 2 - Interface com Run e Params .....	14
Figura 3 - Interface com Dataset.....	14
Figura 4 - Interface com Results.....	15
Figura 5 - Matriz de Confusão.....	15

## Índice de Tabelas

Tabela 1 - Dados alínea a) .....	5
Tabela 2 - Dados alínea b) .....	7
Tabela 3 - Dados alínea c) i) .....	8
Tabela 4 - Dados alínea c) ii).....	9
Tabela 5 - Dados alínea c) iii).....	11
Tabela 6 - Dados alínea d) .....	12

## 1. INTRODUÇÃO

O presente trabalho prático, desenvolvido no âmbito da unidade curricular de **Conhecimento e Raciocínio**, tem como objetivo principal a exploração e aplicação de redes neuronais do tipo *feedforward* para a classificação automática de imagens contendo formas geométricas simples. Esta tarefa visa consolidar conhecimentos teóricos através da implementação prática com recurso ao MATLAB e à sua toolbox de Deep Learning.

O problema proposto consiste no reconhecimento de figuras pertencentes a seis classes distintas: círculo, papagaio (*kite*), paralelogramo, quadrado, trapezoide e triângulo. Para tal, foram fornecidos conjuntos de imagens organizados em diferentes pastas (*start*, *train* e *test*), que serviram como base para o treino, validação e teste das redes neuronais.

Ao longo deste trabalho foram testadas várias configurações de redes neuronais, avaliando-se o impacto de diferentes topologias, funções de ativação, algoritmos de treino e rácios de divisão dos dados. Pretendeu-se não apenas alcançar boas taxas de classificação, mas também compreender de que forma cada um destes parâmetros influencia o desempenho da rede.

Para além da análise quantitativa dos resultados, foi ainda desenvolvida uma aplicação gráfica que permite ao utilizador treinar redes, carregar conjuntos de dados e visualizar os resultados de classificação, tornando o sistema mais acessível e interativo.

Este relatório documenta todo o processo seguido, desde o pré-processamento das imagens até à avaliação final das redes treinadas, incluindo as decisões tomadas, os resultados obtidos e as conclusões retiradas.

## 2. REDES NEURONAIS FEEDFORWARD

As redes neurais utilizadas neste trabalho são do tipo *feedforward*, um modelo simples e bastante comum em tarefas de classificação. Neste tipo de rede, os dados percorrem a rede num único sentido: da entrada até à saída, passando por uma ou mais camadas intermédias chamadas de camadas ocultas.

Cada camada é composta por neurónios, que recebem valores da camada anterior, aplicam uma operação matemática (através de uma função de ativação) e transmitem o resultado à camada seguinte. A camada final da rede dá uma saída que representa a classe prevista para a imagem processada.

Neste trabalho, as imagens analisadas foram transformadas em **matrizes binárias de 28x28 píxeis**, permitindo que cada imagem fosse representada por um vetor de 784 valores ( $28 \times 28$ ), utilizado como entrada para a rede neuronal. Esta conversão garantiu que todas as imagens tivessem o mesmo tamanho e formato, facilitando o processamento.

Estas redes são criadas com diferentes **números de camadas ocultas e neurónios por camada**, dependendo da complexidade pretendida. Além disso, é possível escolher diferentes tipos de **funções de ativação**, que determinam como cada neurónio responde às entradas, e diferentes **funções de treino**, que definem o método usado para ajustar os pesos da rede durante o processo de aprendizagem.

A configuração da rede (número de camadas, neurónios, funções utilizadas, entre outros) influencia diretamente a sua capacidade de aprender e generalizar, sendo por isso importante testar várias combinações e avaliar o desempenho obtido.

## 3. ESTRUTURA DAS TAREFAS

A realização deste trabalho foi dividida em várias etapas, de acordo com as alíneas indicadas no enunciado. Em cada uma das tarefas foram implementadas redes neurais com diferentes configurações, avaliando o seu desempenho na classificação de imagens geométricas.

Nas secções seguintes apresenta-se, para cada alínea, o processo seguido, os parâmetros usados, os resultados obtidos e a análise respetiva.

## 4. A) CLASSIFICAÇÃO COM IMAGENS DA PASTA START

### 4.1 Pré-processamento das Imagens

As imagens da pasta start estão organizadas em seis subpastas, uma para cada forma geométrica: círculo, papagaio (*kite*), paralelogramo, quadrado, trapezoide e triângulo. Cada classe inclui 5 imagens, totalizando 30 imagens.

Estas imagens foram convertidas em **matrizes binárias de 28x28 píxeis**, após serem convertidas para tons de cinzento e redimensionadas. Cada imagem foi depois transformada num vetor de 784 elementos ( $28 \times 28$ ), formando o conjunto de dados de entrada para a rede neuronal. As etiquetas foram convertidas para vetores *one-hot*, de forma a serem compatíveis com a saída da rede.

### 4.2 Rede com uma camada oculta de 10 neurónios

A primeira abordagem consistiu na utilização de uma rede *feedforward* com apenas **uma camada oculta de 10 neurónios**, usando os parâmetros por defeito do MATLAB, sem divisão em validação ou teste (100% dos dados usados no treino).

Esta rede obteve uma **precisão média de 100%** ao longo de 10 repetições c, revelando ser já totalmente eficaz na tarefa de classificação.

### 4.3 Testes com diferentes topologias

Foram depois testadas outras configurações da rede, com o objetivo de comparar o desempenho. As topologias testadas foram:

- 1 camada oculta com 5 neurónios
- 1 camada oculta com 20 neurónios
- 2 camadas ocultas com 10 neurónios cada ([10 10])
- 2 camadas ocultas com 20 neurónios cada ([20 20])

Tabela 1 - Dados alínea a)

Topologia	Média de Precisão
5	99,(3)
10	100
20	100
[10 10]	100
[20 20]	100

#### 4.4 Conclusões da Tarefa a)

- Os resultados mostram que é possível alcançar uma **precisão de 100%** tanto com redes de maior capacidade (20 neurónios ou 2 camadas ocultas) como com algumas mais reduzidas, mesmo com um conjunto de treino reduzido.
- O pré-processamento simples das imagens (binarização e redimensionamento para 28x28) foi suficiente para garantir bons resultados.
- As topologias mais complexas não trouxeram ganhos significativos nesta fase, mas serão úteis nas próximas tarefas com mais dados.
- Todos os exemplos foram usados apenas para treino, como exigido no enunciado.

### 5. B) CLASSIFICAÇÃO COM IMAGENS DA PASTA TRAIN

#### 5.1 Estratégia e Configurações Testadas

Nesta tarefa, foram utilizadas as imagens da pasta train, contendo um conjunto alargado de exemplos para cada uma das seis classes geométricas. As imagens foram tratadas da mesma forma que na alínea anterior: convertidas para escala de cinzento, binarizadas e redimensionadas para **28x28 píxeis**.

O objetivo foi testar e comparar o desempenho de várias configurações de redes neurais *feedforward*. As combinações avaliadas incluíram variações nos seguintes parâmetros:

- **Topologias:**  
Redes com uma ou duas camadas ocultas, contendo 5, 10 ou 20 neurónios em cada camada (5, 10, 20, [10 10], [20 20])
- **Funções de ativação:**
  - Nas camadas ocultas: tansig, logsig
  - Na camada de saída: purelin, logsig, softmax
- **Funções de treino:**
  - trainlm (Levenberg-Marquardt)
  - trainbfg (quase-Newton)
  - traingd (gradiente descendente)
- **Divisão dos dados** (treino, validação e teste):
  - 70% / 15% / 15%
  - 60% / 20% / 20%
  - 50% / 25% / 25%

Cada combinação foi executada **10 vezes**, sendo registadas as **precisões globais** (sobre todos os dados) e as **precisões de teste** (apenas sobre o conjunto de teste). Foi ainda guardada a **matriz de confusão** da melhor rede de cada combinação.



A abordagem de teste seguiu uma lógica **sequencial e controlada**, começando com uma configuração base e, posteriormente, ajustando um parâmetro de cada vez. Inicialmente foram testadas apenas diferentes **topologias**, mantendo os restantes parâmetros com valores por defeito (tansig, purelin, trainlm, divisão 70/15/15). Após escolher a melhor topologia, essa foi fixada e passaram a ser testadas, por ordem, as diferentes funções de ativação, de treino e, por fim, os rácios de divisão dos dados.

## 5.2 Análise de Resultados

A análise dos resultados obtidos permitiu observar o impacto de cada parâmetro no desempenho das redes neuronais treinadas com o conjunto de dados da pasta train.

De forma geral, as redes com **topologia [20 20]** demonstraram melhor desempenho global, sendo as únicas a ultrapassar os **95% de precisão no conjunto de teste**. As topologias mais simples, como 5 ou 10, apresentaram resultados aceitáveis, mas menos consistentes.

No que respeita às **funções de ativação**, logsig obteve ligeiras vantagens face a tansig, especialmente na camada oculta. Já na camada de saída, purelin e softmax demonstraram desempenhos bastante equilibrados, sendo que logsig nesta camada provocou um colapso no desempenho (com precisão a rondar os 24%), revelando-se inadequada para esta tarefa de classificação multiclasse.

Entre as **funções de treino**, trainlm (Levenberg-Marquardt) destacou-se como a mais eficaz, alcançando as melhores precisões com diferentes combinações de parâmetros. trainbfg revelou-se aceitável, embora ligeiramente inferior. Já traingd, baseado em gradiente descendente simples, obteve resultados bastante fracos, ficando aquém dos 25% de precisão.

A análise dos **rácios de divisão dos dados** indicou que a configuração 70% treino / 15% validação / 15% teste proporcionou melhores resultados do que os outros dois rácios testados, mostrando ser uma distribuição equilibrada entre treino e validação, com dados de teste suficientes para avaliar o desempenho.

Com base na média de precisão de teste, foram seleccionadas as **três melhores redes** desta fase, apresentadas abaixo:

Tabela 2 - Dados alínea b)

Rede	Topologia	FuncAtivOculta	FuncAtivSaida	FuncTreino	Ratio	PrecisaoGlobal	PrecisaoTeste
1	[20 20]	logsig	purelin	trainlm	[0.7 0.15 0.15]	96	95,1
2	[20 20]	tansig	purelin	trainlm	[0.7 0.15 0.15]	94,2	93,1
3	[20 20]	logsig	softmax	trainlm	[0.7 0.15 0.15]	96,3	94

Estas redes foram guardadas e serão utilizadas nas próximas tarefas para avaliar a sua capacidade de generalização face a outros conjuntos de dados.

## 6. c) ESTUDO COM CONJUNTOS COMPLETOS DE IMAGENS

Esta alínea tem como objetivo avaliar a capacidade de generalização das redes treinadas anteriormente (1, 2 e 3), utilizando conjuntos de dados distintos: test, start, train e o conjunto total (start + train + test). A análise incide nas melhores redes obtidas na alínea **b)**, testando o seu desempenho em cenários novos e mais abrangentes.

### 6.1 Aplicação das Melhores Redes ao Conjunto test (sem novo treino)

Nesta primeira fase, foram carregadas as **três melhores redes** treinadas na alínea **b)**. Sem qualquer novo treino, estas redes foram utilizadas para **classificar diretamente as imagens do conjunto test**, de forma a avaliar a sua capacidade de generalização para dados nunca vistos. Para cada rede, foram registadas as seguintes métricas:

- **Precisão de teste:** percentagem de classificações corretas nas imagens da pasta test.
- **Matriz de confusão:** identificação das classes melhor e pior reconhecidas.

#### 6.1.1 Análise de resultados

A aplicação direta das três melhores redes ao conjunto test permitiu avaliar o seu desempenho em dados nunca antes vistos. Esta abordagem teve como objetivo testar a **capacidade de generalização** das redes treinadas exclusivamente com imagens da pasta train.

Tabela 3 - Dados alínea c) i)

RedeID	PrecisaoTeste	MatrizConfusao						
1	83,3	10	0	0	0	0	0	0
		0	10	0	0	0	0	0
		0	0	8	2	0	0	0
		0	0	0	10	0	0	0
		3	0	2	0	5	0	0
		0	0	2	1	0	0	7
		0	0	0	0	0	0	0
2	65	10	0	0	0	0	0	0
		0	5	2	0	1	2	0
		0	0	6	3	1	0	0
		0	0	1	9	0	0	0
		3	1	1	0	5	0	0
		0	3	1	2	0	0	4
		0	0	0	0	0	0	0
3	83,3	10	0	0	0	0	0	0
		0	10	0	0	0	0	0
		0	0	7	0	1	2	0
		0	0	1	9	0	0	0
		2	0	2	0	6	0	0
		0	0	2	0	0	0	8
		0	0	0	0	0	0	0

As **redes 1 e 3** apresentaram desempenhos idênticos e significativamente superiores à rede 2. A **rede 2** teve uma precisão mais baixa (65%), o que indica uma menor capacidade de adaptação a novos dados, apesar de ter tido um bom desempenho na fase de treino.

As **matrizes de confusão** revelam que:

- As formas **círculo, quadrado e kite** foram consistentemente bem classificadas em todas as redes.
- As maiores dificuldades surgiram na distinção entre **paralelogramo, trapezoide e triângulo**, com alguns casos de confusão entre estas classes.
- A **rede 1**, por exemplo, confundiu dois **paralelogramos** com **triângulos**, e classificou incorretamente três **trapezoides** como outras formas.

Estas falhas podem ser atribuídas a **semelhanças visuais entre as formas geométricas e a variações no estilo das imagens da pasta test**, que podem não estar suficientemente representadas no conjunto de treino.

De forma geral, a **rede 1** mantém-se como a mais robusta, tendo sido também a que apresentou o melhor desempenho na alínea **b**. A **rede 3**, com ativação softmax na camada de saída, também demonstrou bom potencial de generalização.

## 6.2 Treino com o Conjunto test e Avaliação nas Três Pastas

Nesta fase, as três melhores redes selecionadas na alínea **b)** foram **novamente treinadas**, desta vez **usando apenas as imagens da pasta test**. Após o treino, cada rede foi aplicada às imagens das três pastas (start, train e test) para avaliar a sua capacidade de adaptação e generalização após treino com um conjunto de dados distinto.

Para cada aplicação, foram registadas:

- A **precisão global** (percentagem de classificações corretas)
- A **matriz de confusão** correspondente
- A **precisão por classe**
- A **forma mais bem classificada** (maior precisão por classe)
- A **forma com pior desempenho**

### 6.2.1 Análise de resultados

Os melhores resultados obtidos para cada uma das pastas foram os seguintes:

Tabela 4 - Dados alínea c) ii)

Pasta	Melhor Rede	Precisão (%)	Melhor Forma	Precisão Melhor Forma (%)	Pior Forma	Precisão Pior Forma (%)
start	Rede 3	76,67	Square	100	Triangle	40
train	Rede 1	94,33	Kite	100	Parallelogram	86
test	Rede 1 ou 2	100	Circle	100	—	—

As três redes atingiram **100% de precisão** no conjunto test, o que era esperado, dado que foram **reutilizadas e re-treinadas com estas mesmas imagens**. No entanto, os resultados mais relevantes surgem nos conjuntos start e train, onde se pode avaliar a **capacidade de generalização** para dados **não utilizados no novo treino**.

- **Conjunto start:**

A **rede 3** obteve o melhor desempenho, com uma precisão de **76,67%**, destacando-se face às redes 1 (60%) e 2 (70%). Isto sugere uma **melhor adaptação a dados novos**, mesmo tendo sido treinada apenas com test.

- **Conjunto train:**

A **rede 1** foi a mais eficaz, com **94,33% de precisão**, superando as restantes. Este resultado reforça a **capacidade da rede em reconhecer exemplos já explorados anteriormente na alínea b)**.

- **Conjunto test:**

As **redes 1 e 2** atingiram **100% de precisão**, o que é consistente com o facto de terem sido treinadas com este conjunto. A **rede 3** obteve **96,67%**, valor ainda elevado, mas ligeiramente inferior, o que poderá refletir uma menor tendência para overfitting.

É possível retirar as seguintes conclusões da análise:

- Nenhuma rede foi a melhor em todos os conjuntos.
- A **rede 3** demonstrou a **melhor generalização** para dados não vistos, com destaque no conjunto start.
- A **rede 1** revelou-se **mais sólida na classificação de dados conhecidos**, nomeadamente no conjunto train.
- A **rede 2**, apesar de igualar a rede 1 no test, ficou **consistentemente atrás nos restantes conjuntos**.

Estas diferenças mostram que o desempenho das redes está intimamente ligado à **diversidade dos dados de treino** e reforçam a importância de testes em **múltiplos contextos** para avaliar a robustez de modelos de classificação.

### 6.3 Treino com o Conjunto Completo de Imagens (start + train + test)

Nesta última fase da alínea c), as três melhores redes da alínea b) foram novamente treinadas, desta vez utilizando **todas as imagens disponíveis** (pastas start, train e test). O objetivo foi avaliar o desempenho das redes quando expostas à totalidade do conjunto de dados, maximizando a diversidade de exemplos durante o treino.

Após o treino com o conjunto completo, cada rede foi aplicada **individualmente a cada uma das três pastas** para avaliar a capacidade de generalização por subconjunto.

### 6.3.1 Análise de resultados

Os resultados obtidos foram os seguintes:

Tabela 5 - Dados alínea c) iii)

Rede	Precisão Global (%)	Pasta	Precisão na Pasta (%)	Melhor Forma	Melhor Acc (%)	Pior Forma	Pior Acc (%)
1	100	start	100	Circle	100	Circle	100
		train	100	Circle	100	Circle	100
		test	100	Circle	100	Circle	100
2	100	start	100	Circle	100	Circle	100
		train	100	Circle	100	Circle	100
		test	100	Circle	100	Circle	100
3	98,72	start	96,67	Circle	100	Kite	80
		train	99,33	Circle	100	Trapezoid	98
		test	96,67	Circle	100	Trapezoid	80

As **redes 1 e 2** atingiram **precisão total (100%)** tanto no treino como na classificação de todas as pastas, o que demonstra que foram capazes de memorizar todos os exemplos corretamente — o que é esperado dado que foram treinadas com todo o conjunto de dados. Ainda assim, tal desempenho deve ser interpretado com cuidado, pois pode refletir overfitting e não necessariamente boa generalização para novos dados.

A **rede 3**, apesar de não ter atingido 100% de precisão, manteve valores muito elevados (entre 96,67% e 99,33%), demonstrando um comportamento mais próximo de uma rede com **boa capacidade de generalização**.

Nas três redes, a **forma "circle"** voltou a destacar-se como a mais facilmente reconhecida. Por outro lado, as formas com desempenho mais fraco continuaram a ser **"kite"** e **"trapezoid"**, especialmente na rede 3, embora mesmo aí os valores de precisão se tenham mantido altos (mínimo de 80%).

É possível retirar as seguintes conclusões da análise:

- As redes 1 e 2 apresentaram resultados perfeitos, mas indicam possível **memorização exata dos dados** (overfitting).
- A rede 3, apesar de ligeiramente inferior em precisão, revelou-se **mais robusta**, especialmente nos testes anteriores (c.i e c.ii), demonstrando **melhor equilíbrio entre desempenho e generalização**.
- Esta fase confirmou que as redes neuronais treinadas com mais dados tendem a ser mais estáveis, mas é essencial avaliar o seu comportamento também com dados externos ou novos para validar a sua eficácia real que é o que será feito na alínea seguinte.

## 7. D) AVALIAÇÃO COM IMAGENS DESENHADAS MANUALMENTE

Nesta etapa, avaliou-se a robustez das redes neurais perante imagens **desenhadas manualmente**, criadas fora dos conjuntos originais (start, train, test). Estas imagens foram preparadas seguindo os mesmos procedimentos de pré-processamento: **conversão para tons de cinzento, binarização e redimensionamento para 28×28 píxeis**. Estas imagens foram desenhadas **manualmente num tablet com caneta digital**, de forma a simular entradas reais e não uniformizadas, distintas das imagens dos conjuntos de treino.

Foram utilizadas as **três redes treinadas com o conjunto completo** na alínea c.iii. Cada rede foi testada **10 vezes** sobre o conjunto de desenhos, e foram calculadas:

- A **precisão média de classificação**
- A **matriz de confusão média**
- A **precisão por classe (forma geométrica)**
- A forma com **melhor desempenho médio**
- A forma com **pior desempenho médio**

De seguida encontra-se um exemplo das imagens de cada um dos tipos em análise.

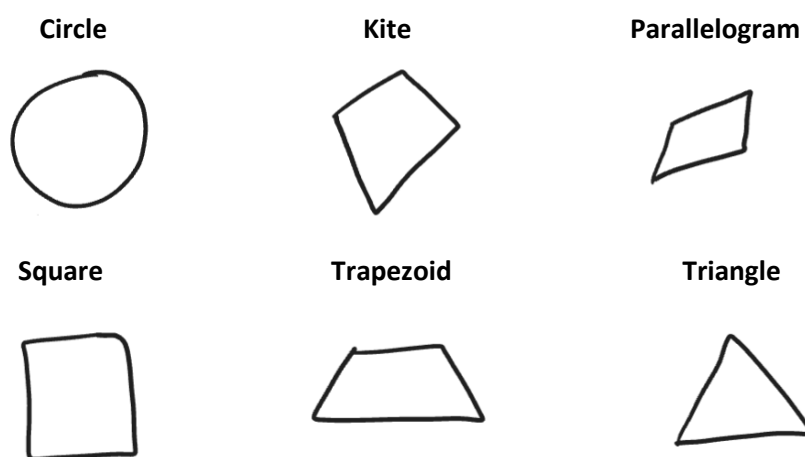


Figura 1 - Formas feitas manualmente

### 7.1 Análise de resultados

Os resultados obtidos foram os seguintes:

Tabela 6 - Dados alínea d)

Rede	Precisão Média (%)	Melhor Forma	Precisão Melhor Forma (%)	Pior Forma	Precisão Pior Forma (%)
1	20	Triangle	60	Parallelogram	0
2	30	Kite	60	Parallelogram	0
3	40	Triangle	100	Parallelogram	0

As redes mostraram uma **forte queda de desempenho** face aos testes anteriores, o que é natural tendo em conta que as imagens manuais introduzem **variações significativas** que não estavam presentes nos dados de treino.

A **rede 3** foi a que apresentou **melhor desempenho global**, atingindo uma precisão média de **40%**, e classificando corretamente **100% das imagens da classe “triangle”** em média. Por outro lado, a forma **“parallelogram”** foi consistentemente mal classificada por todas as redes, com **0% de precisão média**, indicando dificuldades claras na sua distinção.

É possível retirar as seguintes conclusões da análise:

- As redes mostraram **dificuldades significativas** na generalização para dados **completamente externos** aos conjuntos usados em treino.
- A **rede 3** demonstrou novamente **maior capacidade de adaptação**, mesmo que com valores modestos.

Esta fase evidencia a importância de **testar modelos com dados reais ou não convencionais**, especialmente em cenários de aplicação prática.

## 8. E) APLICAÇÃO GRÁFICA DESENVOLVIDA EM MATLAB

Nesta etapa foi desenvolvida uma aplicação gráfica em MATLAB, utilizando a ferramenta *App Designer*, com o objetivo de facilitar a execução das principais tarefas envolvidas no treino, teste e avaliação das redes neuronais implementadas no projeto.

A interface foi organizada em duas secções principais: ***Params/Dataset*** e ***Run/Results***, permitindo uma utilização intuitiva por parte do utilizador. As figuras seguintes mostram a interface da aplicação.

## Reconhecimento de Formas Geométricas com Redes Neurais

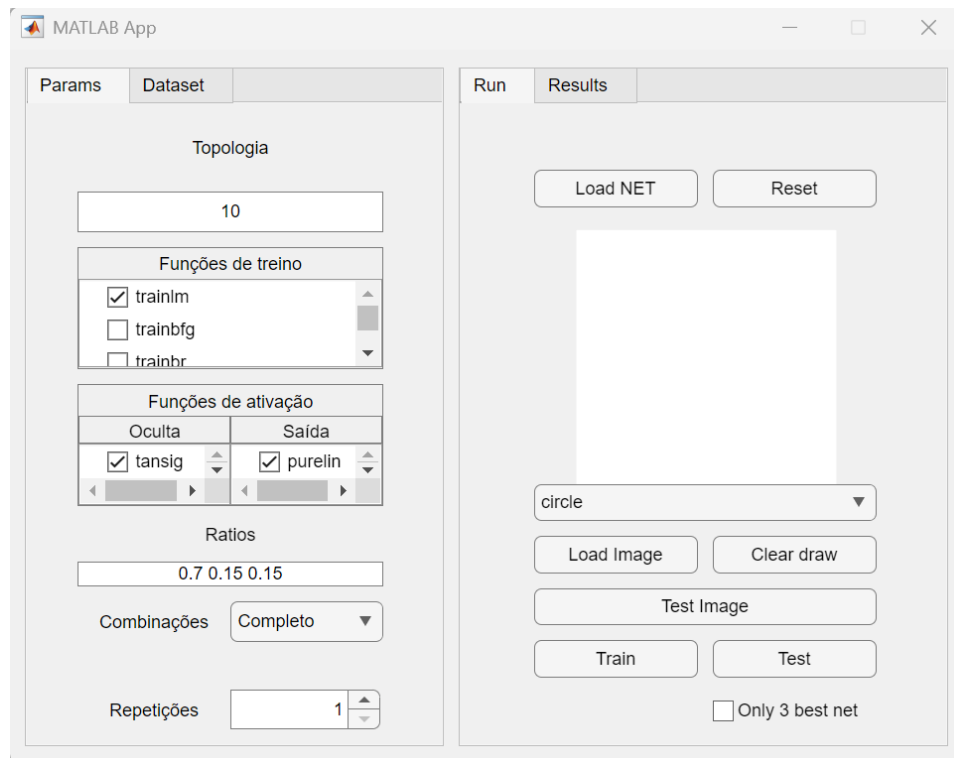


Figura 2 - Interface com Run e Params

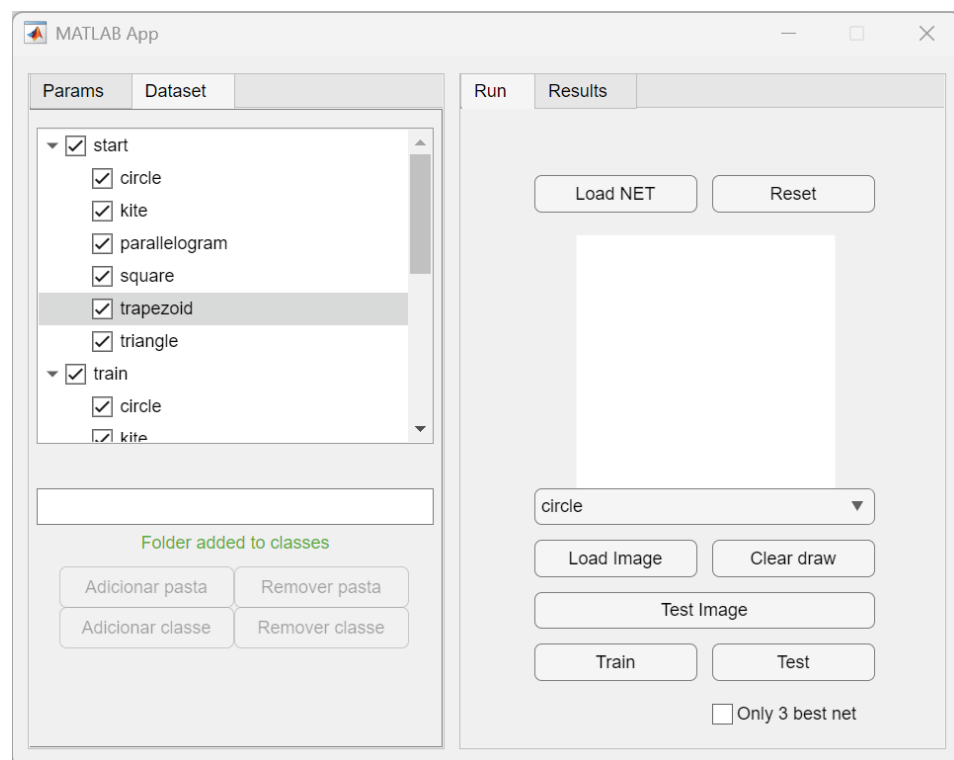


Figura 3 - Interface com Dataset



## Trabalho Prático

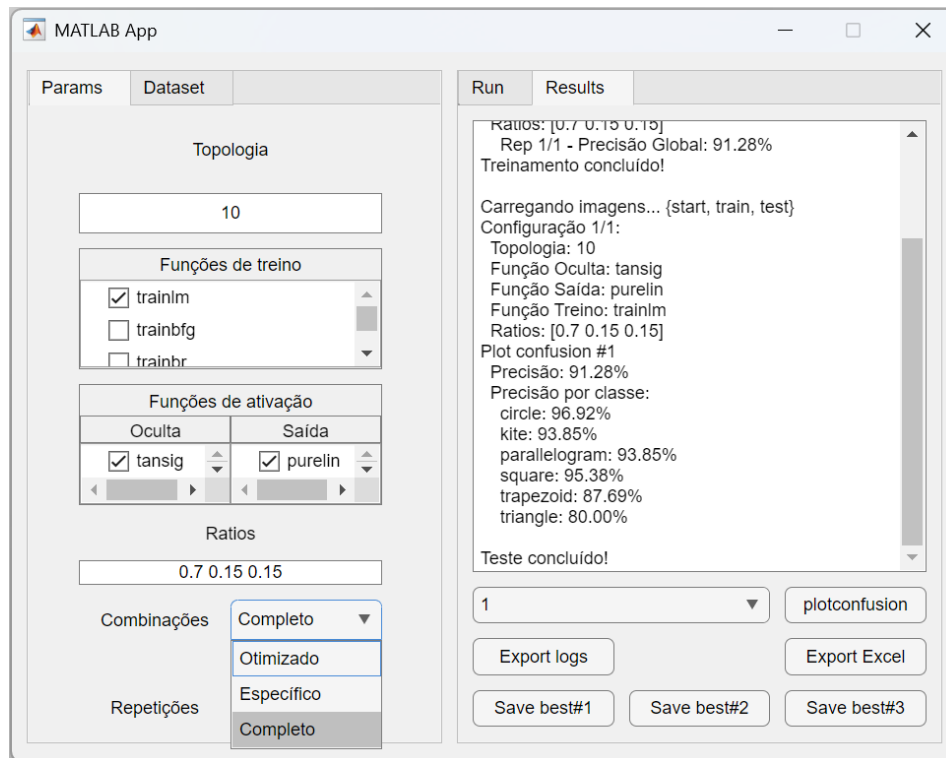


Figura 4 - Interface com Results

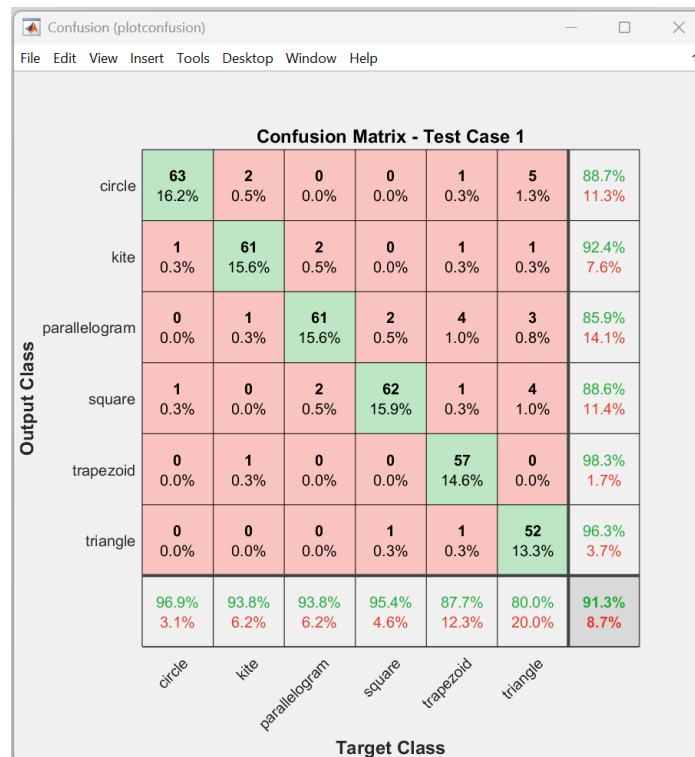


Figura 5 - Matriz de Confusão

## 8.1 Funcionalidades Implementadas

### Configuração de Parâmetros (Params):

A aplicação permite ao utilizador configurar livremente os principais parâmetros das redes neuronais, de forma a facilitar a experimentação com diferentes combinações. As opções disponíveis são:

- **Topologia:** Campo de entrada para definição do número de neurónios da(s) camada(s) oculta(s).
- **Funções de treino:** Lista de seleção múltipla com as seguintes opções: trainlm, trainbfg, trainbr, trainc, traincgb, traincgf, traincgp, traingd, traingda, traingdm, traingdx, trainoss, trainr, trainrp, trains, trainscg.
- **Funções de ativação:**
  - *Camada oculta:* transig, compet, harlim, harlims, logsig, netinv, poslin, purelin, radbas, radbasn, satlin, satlins, softmax.
  - *Camada de saída:* purelin, compet, harlim, harlims, logsig, netinv, poslin, radbas, radbasn, satlin, satlins, softmax, transig.
- **Ratios:** Campo de entrada para definição dos rácios de divisão do dataset em treino, validação e teste (por exemplo: 0.7, 0.15, 0.15).
- **Combinações e repetições:** Menus suspensos para escolha do tipo de execução (Específico, Otimizado, Completo) e do número de repetições a efetuar.

- **Comportamento consoante o tipo de execução:**

- **Específico:**

Nesta opção, a aplicação executa as combinações de parâmetros definidos manualmente pelo utilizador. As listas de seleção permitem escolher múltiplas topologias, funções de treino, funções de ativação (camada oculta e de saída) e rácios de divisão. Com base nas seleções efetuadas, é calculado o **número total de combinações possíveis**, e esse valor é apresentado automaticamente numa **caixa de texto informativa**. No entanto, o utilizador pode também **definir manualmente quantas dessas combinações deseja executar**, introduzindo esse número numa **caixa de texto editável**. Por exemplo, mesmo que existam 5 combinações possíveis, o utilizador pode optar por executar apenas 2 de forma aleatória. Esta funcionalidade permite ajustar a carga computacional e testar apenas subconjuntos das configurações escolhidas, tornando o processo mais flexível e controlado.

- **Otimizado:**

A aplicação segue uma lógica de otimização sequencial, idêntica à utilizada na alínea **b)** do relatório. A abordagem começa com uma configuração base padrão (ex.: topologia simples, funções tansig, purelin, treino com trainlm, divisão 70/15/15). A partir daí, são ajustados os parâmetros um de cada vez, por ordem:

1. Primeiramente, são testadas diferentes **topologias**, mantendo os restantes parâmetros por defeito.

2. A melhor topologia é então fixada, e passam a ser testadas as diferentes **funções de ativação**.
3. Segue-se a comparação entre várias **funções de treino**.
4. Por fim, são analisados diferentes **rácios de divisão** dos dados. A melhor configuração final é determinada com base na precisão global e nos resultados médios das repetições. Esta opção permite encontrar boas configurações com menor custo computacional que a execução completa.

▪ **Completo:**

A aplicação realiza uma varredura total sobre todas as combinações possíveis de topologias, funções de treino, funções de ativação e rácios de treino/validação/teste, consoante os valores definidos no sistema. É a opção mais abrangente e computacionalmente intensiva, usada para estudo exaustivo e comparação global de resultados.

**Execução da Rede (Run):**

- **Load NET / Reset:** Carregamento de redes previamente treinadas e reposição da interface.
- **Desenho ou carregamento de imagem:** Possibilidade de desenhar manualmente (botão direito do rato para desenhar e botão esquerdo para apagar) uma forma ou carregar uma imagem à escolha do utilizador.
- **Classificação de imagens:** A imagem introduzida pode ser testada diretamente usando as redes carregadas (“Test Image”).
- **Train/Test:** Botões para iniciar o processo de treino ou teste, consoante a configuração escolhida.
- **Seleção de classe esperada:** Menu para indicar a forma desenhada (círculo, trapézio, etc.).
- **Opções adicionais:** Possibilidade de limitar a execução às 3 melhores redes anteriores.

**Gestão de Conjuntos de Dados (Dataset):**

Esta secção da aplicação permite ao utilizador configurar e gerir os conjuntos de dados a utilizar nas tarefas de treino, teste e validação. A interface apresenta:

- **Seleção de conjuntos:** Listagem com caixas de seleção para incluir/excluir os diretórios start, train e test da execução atual. Estes diretórios correspondem às pastas com os conjuntos de imagens fornecidos.
- **Gestão de pastas/classes adicionais:** Campo de introdução de caminho manual, que pode ser usado para adicionar novos diretórios à análise. Caso a pasta introduzida não corresponda a uma categoria válida, é apresentada uma mensagem de erro.
- **Botões auxiliares (Adicionar/Remover pasta e classe):** Estes botões estão funcionalmente ativos e o seu comportamento depende do conteúdo introduzido no campo de texto.
  - Se for introduzido o caminho de uma **nova pasta válida**, o botão “Adicionar pasta” é ativado e permite incluir esse diretório no conjunto de dados.
  - Se o caminho já corresponder a uma pasta previamente incluída, o botão “Remover pasta” é ativado, permitindo removê-la.

- Caso o texto introduzido corresponda a uma **classe existente** (ex.: “circle”, “triangle”, etc.) numa pasta, os botões de adicionar ou remover classe também são ativados em conformidade.

### Visualização de Resultados (Results):

A secção **Results** permite visualizar os resultados das execuções de treino e teste realizadas na aplicação. As funcionalidades disponíveis incluem:

- **Área de visualização de resultados:** Painel central onde são apresentados os registos e métricas das execuções, como precisão global, matrizes de confusão, entre outros dados relevantes definidos pelo utilizador.
- **Exportação de dados:**
  - **Exportar logs:** Guarda os registos das execuções num ficheiro de texto, permitindo consulta posterior.
  - **Exportar Excel:** Exporta os principais resultados (precisões, matrizes, etc.) para um ficheiro Excel.
- **Guardar redes treinadas:**
  - **Guardar melhor #1, #2, #3:** Guarda as três melhores redes obtidas com base no desempenho médio nas repetições. Estas redes podem ser reutilizadas ou testadas posteriormente na aplicação.
- **Seleção de Configuração e Matriz de Confusão:**
  - A interface inclui um menu de seleção que permite ao utilizador escolher a configuração específica para a qual deseja gerar a matriz de confusão.
  - Após seleccionar a configuração pretendida, ao clicar no botão **plotconfusion**, a matriz de confusão correspondente é gerada e exibida.

## 9. CONCLUSÃO

O presente trabalho prático abordou a classificação de formas geométricas utilizando redes neurais feedforward implementadas no MATLAB, no âmbito da unidade curricular de Conhecimento e Raciocínio. A exploração de diferentes configurações e parâmetros permitiu aprofundar o conhecimento sobre redes neurais, bem como avaliar a sua capacidade de generalização e precisão na identificação de figuras geométricas simples.

Ao longo do projeto, foi realizada uma análise comparativa das arquiteturas de redes, variando topologias, funções de ativação, métodos de treino e rácios de divisão dos dados. A partir das experiências realizadas, constatou-se que redes com maior capacidade (como as com duas camadas ocultas de 20 neurónios) apresentaram um desempenho superior em cenários mais complexos, especialmente quando utilizadas funções de ativação e métodos de treino adequados, como logsig na camada oculta e trainlm como algoritmo de aprendizagem.

Os resultados demonstraram que, apesar de algumas redes alcançarem 100% de precisão nos conjuntos de treino e teste, essas mesmas redes mostraram fragilidades quando expostas a dados não previamente vistos, especialmente no contexto das imagens desenhadas manualmente. Este facto evidencia a importância de evitar o overfitting e de garantir a robustez do modelo frente a novas entradas.

A implementação de uma aplicação gráfica interativa foi um passo relevante para tornar o sistema mais acessível e intuitivo. Através desta aplicação, os utilizadores podem configurar os parâmetros da rede, realizar o treino, testar com novos dados e visualizar os resultados de forma prática e eficiente. Esta ferramenta não só facilita a experimentação, mas também contribui para a aplicação prática dos conceitos explorados no trabalho.

Concluindo, o trabalho prático cumpriu os objetivos propostos, consolidando os conhecimentos sobre redes neurais feedforward e suas aplicações em classificação de imagens geométricas. A análise dos resultados evidenciou a necessidade de equilibrar a capacidade da rede com a diversidade dos dados de treino para garantir um desempenho generalizável. A aplicação gráfica desenvolvida complementa este estudo, permitindo futuras explorações e validações com novos conjuntos de dados, reforçando a aplicabilidade dos conhecimentos adquiridos.



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra