



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

Conhecimento e Raciocínio

Redes neuronais feedforward

Trabalho realizado por:

Francisco Carreira Ruivo – 2021142024 – P3

Sebastian Vigas – 2023155905 – P3



INSTITUTO POLITÉCNICO DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, 05/2025

1. Introdução

O presente relatório insere-se no âmbito da unidade curricular de *Conhecimento e Raciocínio*, do 2.º ano da Licenciatura em Engenharia Informática, e tem como objetivo documentar o trabalho prático desenvolvido ao longo do semestre letivo. Este trabalho incidiu sobre a exploração e implementação de redes neurais feedforward, utilizando a toolbox de *Deep Learning* do Matlab.

Com este projeto, pretendeu-se aprofundar os conhecimentos teóricos e práticos sobre redes neurais, nomeadamente no que respeita à sua construção, treino e avaliação em tarefas de classificação de imagens geométricas. Foram disponibilizadas imagens pertencentes a seis classes distintas — círculo, kite, paralelogramo, quadrado, trapezoide e triângulo — com o intuito de treinar modelos capazes de as identificar com precisão.

Ao longo do trabalho, foram implementadas várias configurações de redes neurais, testadas diferentes funções de ativação e treino, e analisado o desempenho dos modelos em múltiplos conjuntos de dados (start, train e test). Adicionalmente, foi desenvolvida uma aplicação gráfica em Matlab que permite ao utilizador configurar, treinar e aplicar redes neurais de forma intuitiva.

2. Tratamento das Imagens

O tratamento das imagens é uma etapa crucial neste trabalho prático, pois a qualidade dos dados de entrada influencia diretamente o desempenho das redes neuronais. O conjunto de dados disponibilizado contém 390 imagens distribuídas por seis classes distintas: círculo, kite, paralelogramo, quadrado, trapézio e triângulo, organizadas em três pastas: "**start**", "**train**" e "**test**".

Para processar estas imagens de forma eficiente, criámos a função `carregarImagens.m`. Esta função recebe três parâmetros principais: o caminho base onde se encontram as imagens, as classes a processar e o tamanho desejado para o redimensionamento.

O processamento das imagens é feito em três passos principais. Primeiro, convertemos as imagens para escala de cinza, simplificando a representação das formas geométricas. Em seguida, aplicamos uma binarização, transformando as imagens em matrizes binárias onde cada pixel é representado por 0 ou 1.

Por último, redimensionamos todas as imagens para um tamanho padrão, garantindo que todas as entradas da rede neuronal tenham a mesma dimensão.

A função `carregarImagens.m` processa cada imagem individualmente, aplicando estas transformações e organizando os dados em duas matrizes: X e Y. A matriz X contém as imagens processadas, enquanto a matriz Y contém os rótulos em formato one-hot encoding. Esta estrutura de dados foi escolhida por ser adequada para o treino de redes neuronais e permitir uma manipulação eficiente dos dados.

3. Treino e Estudo Estatístico

Alinea a)

Para tentar atingir 100% de precisão global, mantivemos a função de treino "*trainlm*", uma vez que é a mais adequada para redes treinadas com conjuntos de dados pequenos (<1000 amostras). Utilizámos uma topologia fixa de **10 neurónios numa única camada** e aumentámos o número de *epochs* do valor predefinido (1000) para **5000** na primeira tentativa.

Configurações testadas e resultados:

1. Função de ativação "*logsig*" na camada oculta:

- Obteve-se uma precisão de **76,67%**.

Conf1	1	10	logsig	trainlm	dividerand = {0.7, 0.15, 0.15}	5000	76.67%
-------	---	----	--------	---------	--------------------------------	------	--------

2. Modificação das funções de ativação:

- Camada oculta: "*tansig*"
- Camada de saída: "*purelin*"
- Resultado: **Melhoria de 3,33%** (total de **80,00%**).
- Testes posteriores com outras funções de ativação não trouxeram melhorias adicionais.

Conf2	1	10	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	5000	80.00%
Conf3	1	10	logsig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	5000	80.00%

3. Ajuste na divisão dos dados (treino/teste):

- **1ª tentativa:** Aumento do conjunto de treino e redução do de teste → **Precisão diminuiu 40%**.
- **2ª tentativa:** Aumento do conjunto de teste e redução do de treino → **Melhoria de 10%** (atingindo **90%** no total).

Conf4	1	10	logsig, purelin	trainlm	dividerand = {0.85, 0.15, 0.05}	5000	50.00%
Conf5	1	10	logsig, purelin	trainlm	dividerand = {0.75, 0.05, 0.25}	5000	90.00%

○

Possível explicação para os resultados:

- A estagnação nos 90% pode estar relacionada com o **pré-processamento das imagens**, embora esta seja apenas uma hipótese que carece de validação.

Relativamente aos resultados do estudo sobre o número e dimensão da topologia de redes neuronais treinadas com a pasta 'start' utilizando os valores predefinidos, é possível concluir sobre:

1. Número de neurónios por camada

Tem um impacto decisivo na precisão, devido a que quando aumentamos de 5 para 10 neurónios a precisão aumenta. Como é possível ver elevou a precisão de **61,0% para 85,0%** (Conf1 → Conf2) nas redes de duas camadas e nas redes de 3 camadas repete-se o mesmo padrão, **60,0% (5 neurónios/camada) vs. 77,3% (10 neurónios/camada)**, mas já se alocamos ainda mais neurónios a percentagem não melhora **85,0% (10 neurónios/camada) vs. 83,66% (15 neurónios/camada)** (diferencia insignificante).

2. Número de camadas

Não tem impacto definido ou até pode piorar a precisão da rede se não são definidos um número suficiente de neurónios para ela funcionar eficientemente como é evidenciado nos dados:

- Para a mesma dimensão de neurónios (10 por camada), aumentar de 2 (Conf2: 85%) para 3 camadas (Conf4: 77,3%) **reduz a precisão**.
- O mesmo acontece com 5 neurónios por camada (Conf1: 61% vs. Conf3: 60%).

Alinea b)

Influência da Topologia

	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste
Configuração por defeito	1	10	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	79,20%	60,67%
O número e dimensão das camadas escondidas influencia o desempenho?							
Conf1	2	5, 5	tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	80,17%	58,89%
Conf2	2	10,10	tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	88,70%	67,11%
Conf3	3	5,5,5	tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	76,67%	55,33%
Conf4	3	10,10,10	tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	89,10%	61,33%

Ao testarmos várias topologias, verificámos comportamentos muito interessantes:

- A rede com apenas uma camada escondida e 10 neurónios mostrou um desempenho razoável, com 79,20% de precisão global e 60,67% nos testes.
- Quando adicionámos uma segunda camada, os resultados melhoraram bastante. A topologia [10, 10] foi a que mais se destacou, com 88,70% de precisão global e 67,11% nos testes, mostrando boa capacidade para lidar com novos exemplos.
- Apesar da rede com três camadas [10, 10, 10] ter conseguido a melhor precisão global (89,10%), a sua precisão nos testes foi mais baixa (61,33%) que a rede [10, 10], o que nos leva a pensar que a rede começou a "decorar" em vez de "aprender".
- As redes com mais camadas mas menos neurónios, como a [5, 5, 5], tiveram pior desempenho (76,67% global, 55,33% nos testes), o que nos leva a concluir que para estas formas geométricas, o número de neurónios por camada é mais importante que ter muitas camadas.

Influência das Funções de Treino

	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste
A função de treino influencia o desempenho?							
Conf1	1	10	tansig, purelin	traingd	dividerand = {0.7, 0.15, 0.15}	28,70%	26,89%
Conf2	1	10	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	90,90%	70,67%
Conf3	1	10	tansig, purelin	trainbr	dividerand = {0.7, 0.15, 0.15}	16,67%	10,44%
Conf4	1	10	tansig, purelin	traingcf	dividerand = {0.7, 0.15, 0.15}	40,20%	34,89%

Notámos grandes diferenças entre os vários algoritmos de treino:

- O algoritmo 'trainlm' (Levenberg-Marquardt) foi claramente o melhor, com 90,90% de precisão global e 70,67% nos testes. A sua capacidade de chegar rapidamente a boas soluções torna-o perfeito para este problema.
- O algoritmo 'traingd' (Gradiente Descendente) teve um desempenho muito pior, com apenas 28,70% de precisão global e 26,89% nos testes, provavelmente porque converge muito lentamente.
- O algoritmo 'trainbr' (Regularização Bayesiana) foi o pior de todos com apenas 16,67% de precisão global e 10,44% nos testes, talvez por aplicar uma regularização demasiado forte para o nosso conjunto de dados. Foi o com pior resultados e o que demorou mais a processar
- O 'traingcf' (Gradiente Conjugado) ficou a meio caminho (40,20% global, 34,89% nos testes), mas ainda assim muito abaixo do 'trainlm'.

Influência das Funções de Ativação

	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste
As funções de ativação influenciam o desempenho?							
Conf1	1	10	logsig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	87,23%	64,00%
Conf2	1	10	tansig, logsig	trainlm	dividerand = {0.7, 0.15, 0.15}	89,33%	65,78%
Conf3	1	10	logsig,radbas	trainlm	dividerand = {0.7, 0.15, 0.15}	85,63%	60,00%
Conf4	1	10	radbas, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	38,47%	24,00%

Quanto às várias funções de ativação que experimentámos:

- A combinação 'tansig'/'logsig' deu-nos os melhores resultados: 89,33% de precisão global e 65,78% nos testes. Esta combinação parece captar melhor as características das formas geométricas.
- A combinação 'logsig'/'purelin' também funcionou bem (87,23% global, 64,00% nos testes), o que mostra que diferentes funções sigmoidais são boas para este problema.
- Por outro lado, a combinação 'radbas'/'softmax' teve um desempenho muito fraco (38,47% global, 24,00% nos testes), o que prova que nem todas as funções são adequadas para este tipo de classificação.

A escolha das funções de ativação é determinante para o desempenho da rede, sendo as combinações sigmoidais (logsig e tansig) significativamente superiores para a classificação de formas geométricas, enquanto a combinação radbas/softmax revelou-se inadequada para este problema específico

Influência da Divisão dos Dados

	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste
A divisão de exemplos pelos conjuntos influencia o desempenho?							
Conf1	1	10	tansig, purelin	trainlm	dividerand = [0.7, 0.15, 0.15]	87,67%	60,44%
Conf2	1	10	tansig, purelin	trainlm	dividerand = [0.8, 0.1, 0.1]	91,10%	71,00%
Conf3	1	10	tansig, purelin	trainlm	dividerand = [0.6, 0.2, 0.2]	85,27%	64,83%
Conf4	1	10	tansig, purelin	trainlm	dividerand = [0.5, 0.25, 0.25]	82,93%	82,93%
Conf5	1	10	tansig, purelin	trainlm	dividerand = [0.9, 0.05, 0.05]	89,93%	64,00%

A forma como dividimos os dados também afetou os resultados:

- A divisão 80/10/10 (treino/validação/teste) deu-nos os melhores resultados (91,10% global, 71,00% teste).
- A divisão 90/05/05 teve boa precisão global (89,93%), mas precisão de teste mais baixa (64,00%), o que provavelmente indica que a rede começou a decorar os exemplos.
- As divisões com menos dados de treino (50/25/25 e 60/20/20) tiveram desempenhos mais fracos, mostrando que precisamos de bastantes exemplos para treinar bem a rede.

Concluimos que a divisão adequada dos dados é crucial para o desempenho da rede neural, sendo 80/10/10 a proporção ideal para este problema específico de classificação de formas geométricas.

As três redes neurais com melhores desempenhos

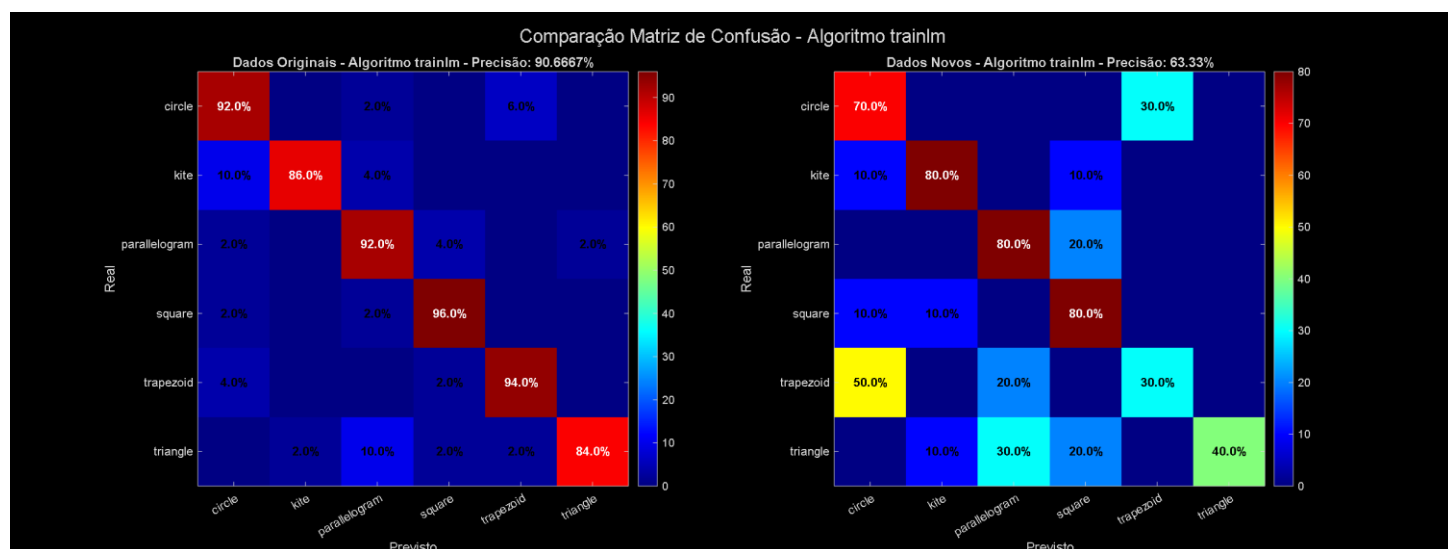
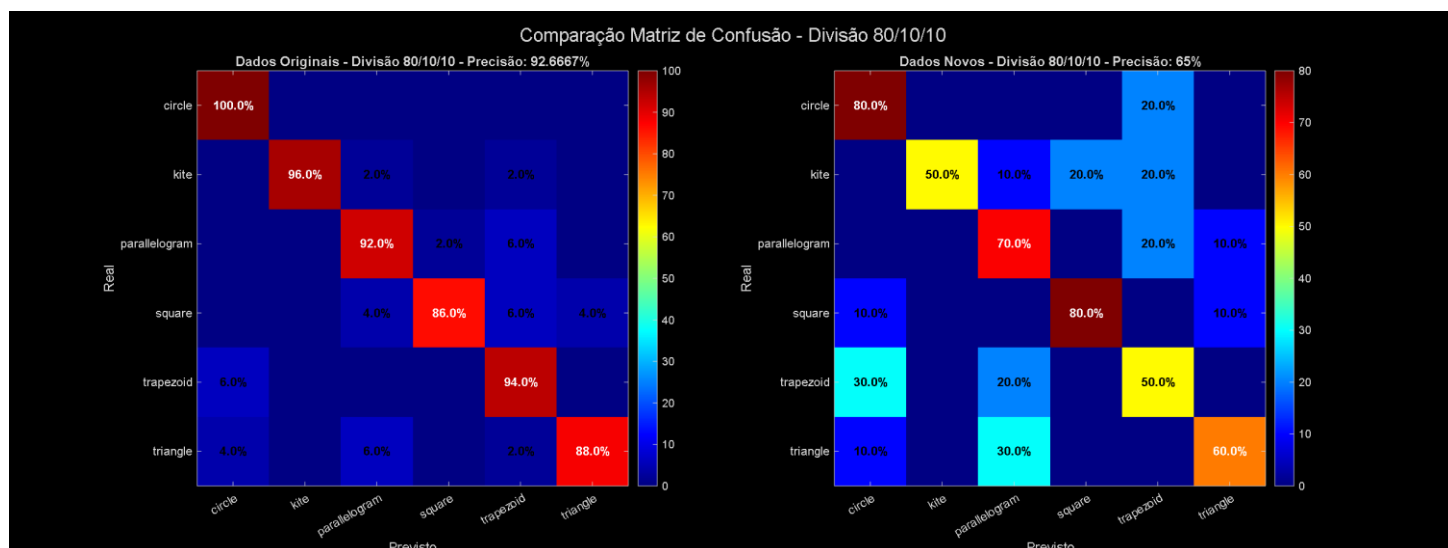
	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos	Precisão Global	Precisão Teste
As três redes neurais com melhores desempenhos?							
Conf1	2	10 , 10	tansig, tansig, logsig	trainlm	dividerand = [0.8, 0.1, 0.1]	91,10%	71,00%
Conf2	2	10 , 10	tansig, tansig, logsig	trainlm	dividerand = [0.7, 0.15, 0.15]	90,90%	70,67%
Conf3	2	10 , 10	tansig, tansig, logsig	trainlm	dividerand = [0.7, 0.15, 0.15]	88,70%	67,11%

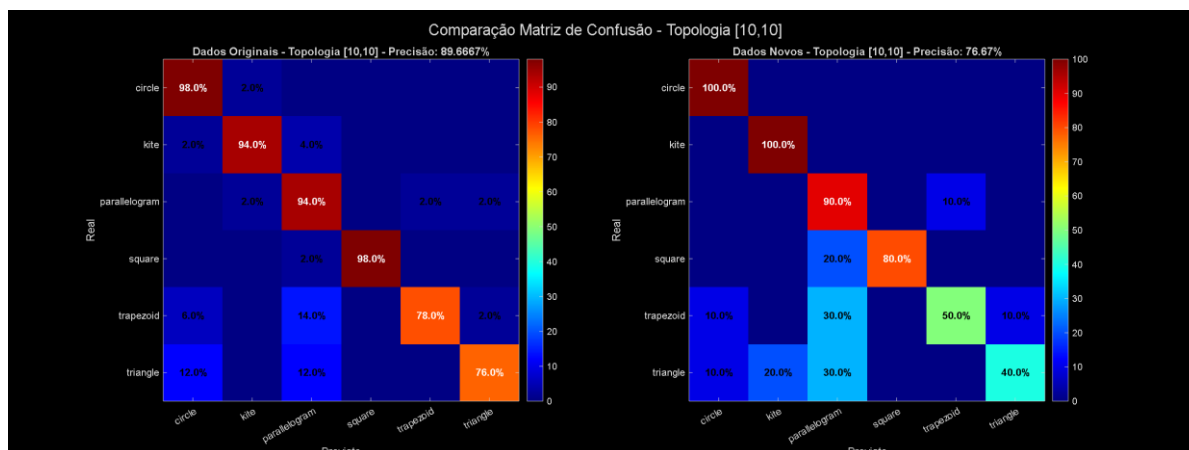
As três redes com melhor desempenho partilham a mesma estrutura base: duas camadas de 10 neurónios, treino “trainlm” e funções de ativação “tansig/tansig/logsig”. O que as diferencia é a divisão dos dados. A primeira rede, utilizando 80% para treino (em vez de 70%), alcançou a melhor precisão (71% nos testes). As restantes redes obtiveram 70,67% e 67,11%, demonstrando que mais dados de treino beneficiam a aprendizagem neste caso de classificação de formas geométricas.

Alinea c)

i.) Avaliação e análise das redes treinadas com imagens novas

Para esta primeira parte da tarefa, recuperámos as três melhores redes neuronais obtidas na alínea anterior b) e avaliámos o seu desempenho utilizando apenas as imagens da pasta "test", que não foram utilizadas durante o treino. O objetivo foi verificar a capacidade de generalização das redes para imagens completamente novas, sem qualquer ajuste adicional dos parâmetros. Analisámos a precisão global, a precisão por classe e as matrizes de confusão para cada rede, procurando identificar padrões e comparar estes resultados com os obtidos anteriormente.





Ao comparar os resultados obtidos com as três redes nas alíneas b) e c) i), observaram-se algumas diferenças interessantes no seu desempenho. A Rede 1, que utilizou a divisão 80/10/10, baixou de 71,00% para 65,00% na precisão. De forma semelhante, a Rede 2 (algoritmo trainlm) também apresentou uma redução, passando de 70,67% para 63,33%.

No entanto, a Rede 3, com topologia [10,10], surpreendeu ao melhorar o seu desempenho de 67,11% para 76,67%. Analisando as matrizes de confusão, notou-se que:

- As formas 'circle' e 'square' são geralmente bem reconhecidas por todas as redes
- Existe alguma confusão entre 'trapezoid' e 'triangle', provavelmente por serem formas semelhantes
- A Rede 3 conseguiu reconhecer perfeitamente as formas 'circle' e 'kite' (100%)

A média geral teve apenas uma pequena redução de 1,26%, o que sugere que as redes mantiveram um desempenho razoável com as novas imagens.

A Rede 3 destacou-se positivamente, mostrando que a sua estrutura com duas camadas escondidas de 10 neurónios cada pode ser mais adequada para este tipo de classificação

.As principais dificuldades concentraram-se no reconhecimento de 'trapezoid' e 'triangle', que são formas que podem ser facilmente confundidas dependendo da sua orientação

Nota: As imagens das matrizes de confusão foram geradas por um script separado que não incluímos na entrega, pois serviu apenas para criar as visualizações de apoio, não sendo essencial para o funcionamento do sistema de classificação

ii.) Treino Especializado com Dados de Teste e análise e classificação (start,train,test)

Nesta alínea, as três melhores redes da alínea b) foram treinadas utilizando apenas as imagens da pasta 'test' e posteriormente testadas com os três conjuntos de dados disponíveis

Rede 1 (divisão 80/10/10):	Precisao_Global	Precisao_circle	Precisao_kite	Precisao_parallelogram	Precisao_square	Precisao_trapezoid	Precisao_triangle
start	40	80	100	40	0	20	0
train	55,67	74	72	56	50	54	28
test	90	100	100	70	90	100	80
Rede 2 (algoritmo trainlm):	Precisao_Global	Precisao_circle	Precisao_kite	Precisao_parallelogram	Precisao_square	Precisao_trapezoid	Precisao_triangle
start	46,67	100	40	80	20	20	20
train	52	80	44	36	70	48	34
test	81,670	100	60	60	90	100	80
Rede 3 (topologia [10 10]):	Precisao_Global	Precisao_circle	Precisao_kite	Precisao_parallelogram	Precisao_square	Precisao_trapezoid	Precisao_triangle
start	20	60	0	40	0	20	0
train	46	46	50	48	62	46	24
test	86,66666667	100	80	90	100	90	60

Ao analisar os resultados das três redes, verificamos que todas apresentam um bom desempenho no conjunto de teste, com precisões globais entre 81,67% e 90%.

A Rede 1 (80/10/10) conseguiu os melhores resultados no conjunto de teste, com 90% de precisão global. Destaca-se principalmente no reconhecimento de círculos, kites e trapezóides, onde atingiu 100% de precisão.

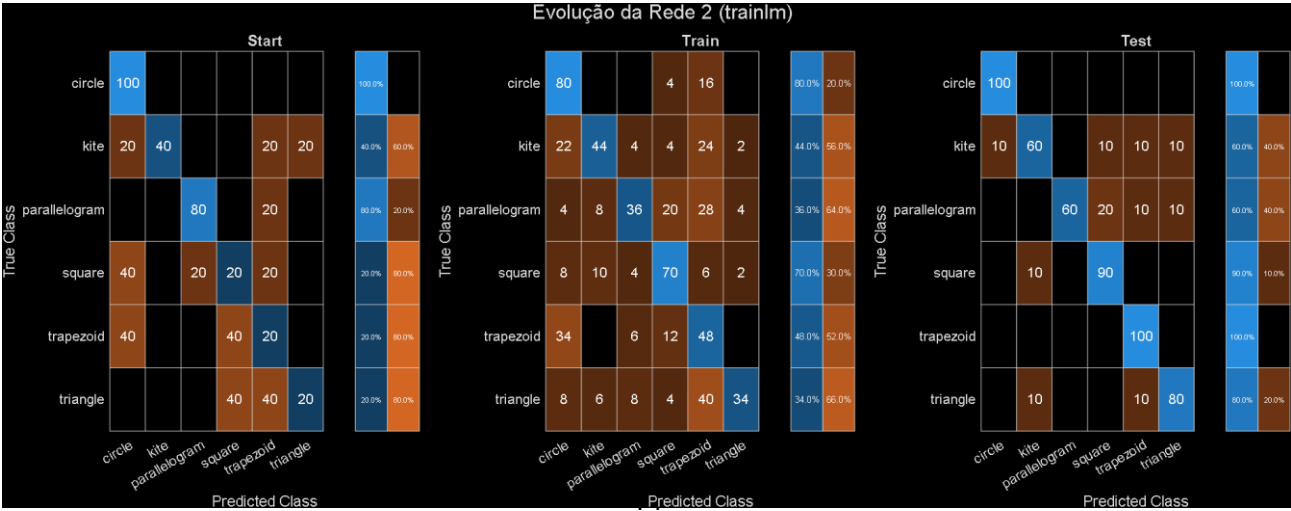
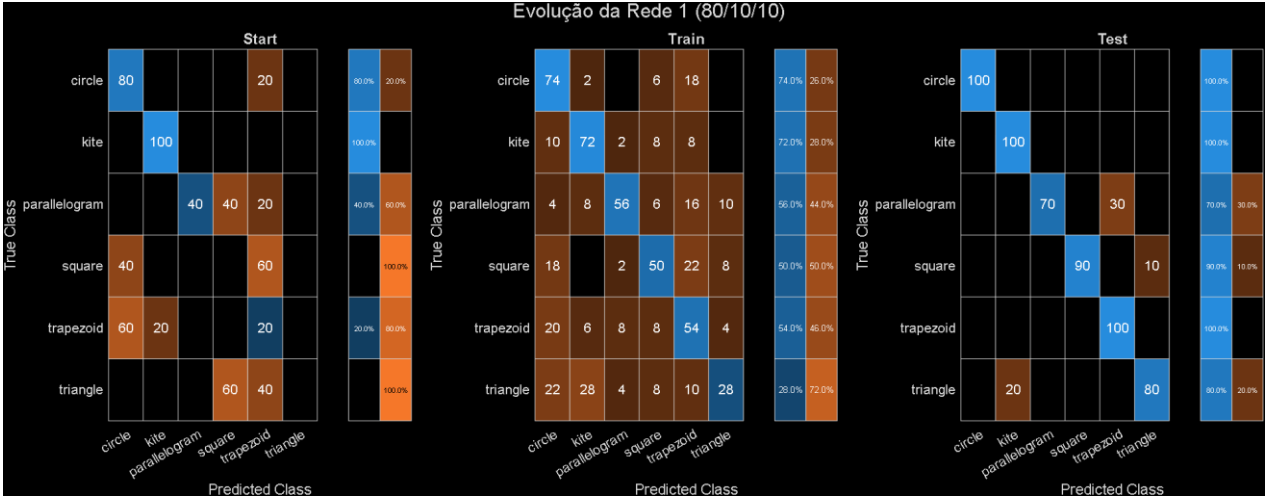
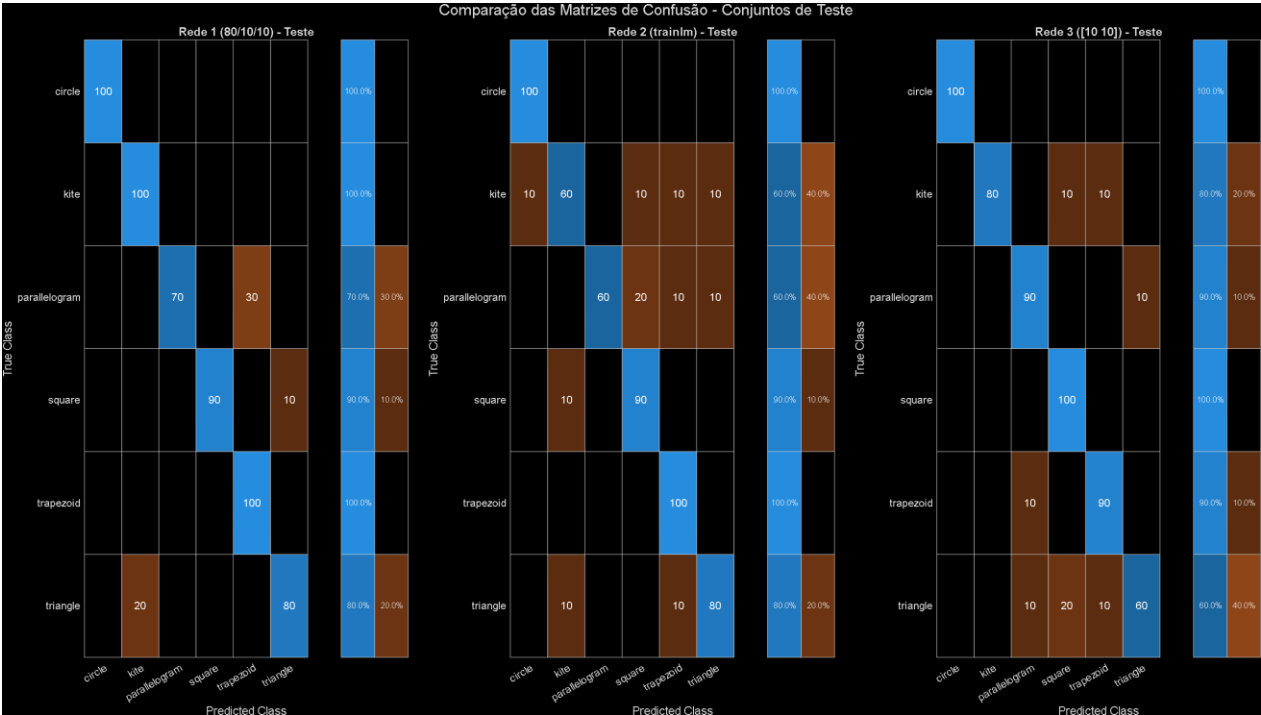
A Rede 2 (trainlm) mostrou um comportamento mais uniforme entre as diferentes formas, com 81,67% de precisão global no teste. Esta rede teve particular sucesso com círculos, mantendo também bons resultados nas restantes formas.

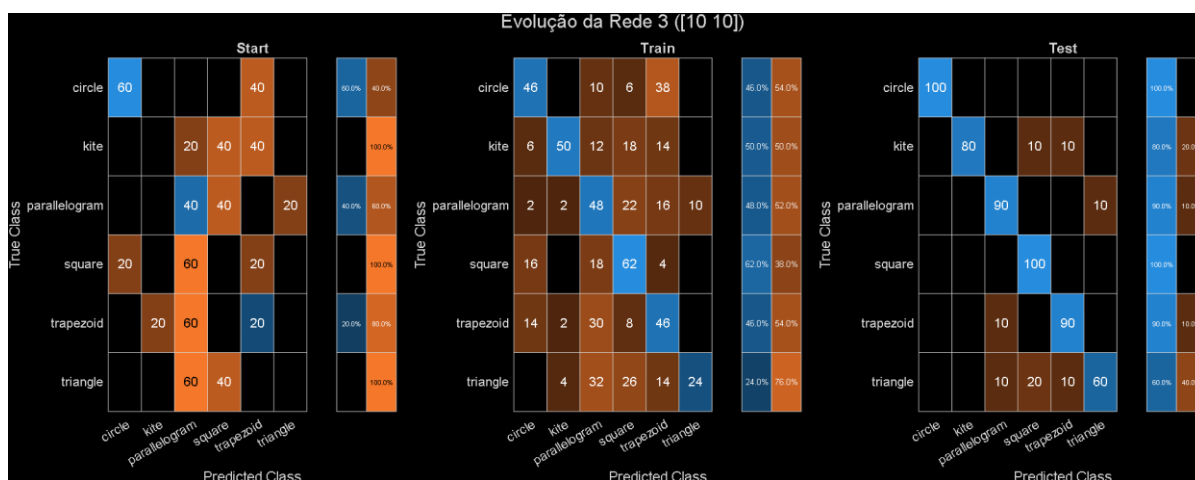
A Rede 3 ([10 10]) alcançou 86,67% de precisão global no teste, com excelente desempenho em círculos e quadrados (100%).

Conclusões a tirar:

- Todas as redes reconhecem círculos com grande eficácia
- Os triângulos são geralmente mais difíceis de identificar
- Existe uma tendência clara de melhoria do desempenho no conjunto de teste
- Cada rede tem os seus pontos fortes específicos

Em suma, os resultados mostram que as redes conseguem classificar as formas geométricas com boa precisão, especialmente no conjunto de teste.





iii.) Otimização Global com Treino em Todos os Conjuntos de Dados

Rede_Conjunto	Precisao_Global	Precisao_circle	Precisao_kite	Precisao_parallelogram	Precisao_square	Precisao_trapezoid	Precisao_triangle
Rede 1: start	100	100	100	100	100	100	100
Rede 1: train	96	100	94	90	98	100	94
Rede 1: test	95	100	100	100	90	80	100
Rede 2: start	90	100	80	100	100	100	60
Rede 2: train	92,66	96	98	86	96	90	90
Rede 2: test	93,33	100	100	80	100	90	90
Rede 3: start	96,66	100	100	100	100	100	80
Rede 3: train	93,66	88	100	94	100	94	86
Rede 3: test	91,66	80	100	100	100	80	90

Rede 1 (80/10/10):

- Melhor desempenho global no teste (95%)
- Perfeita no conjunto start (100% em todas as formas)
- Boa classificação de circles, kites, parallelograms e triangles

Rede 2 (trainlm):

- Boa precisão no teste (93,33%)
- Dificuldades com triangles no início (60%)
- Excelente com circles, kites e squares

Rede 3 ([10 10]):

- Precisão de 91,66% no teste
- Perfeita com kites, parallelograms e squares
- Algumas dificuldades com circles e trapezoids

Todas as redes treinadas com as imagens combinadas mostraram resultados bastante bons, com precisões acima de 90%. Obviamente que se conclui que ao usar mais dados para treino melhora o reconhecimento das formas geométricas.

- A Rede 1 (80/10/10) foi perfeita a reconhecer todas as formas no conjunto start
- A Rede 2 (trainlm) teve o melhor resultado global no teste (93,33%)
- Os círculos foram sempre bem reconhecidos em quase todas as redes
- Os triângulos causaram mais problemas, especialmente na Rede 2 no start (só 60% corretos)

Formas mais fáceis/difíceis

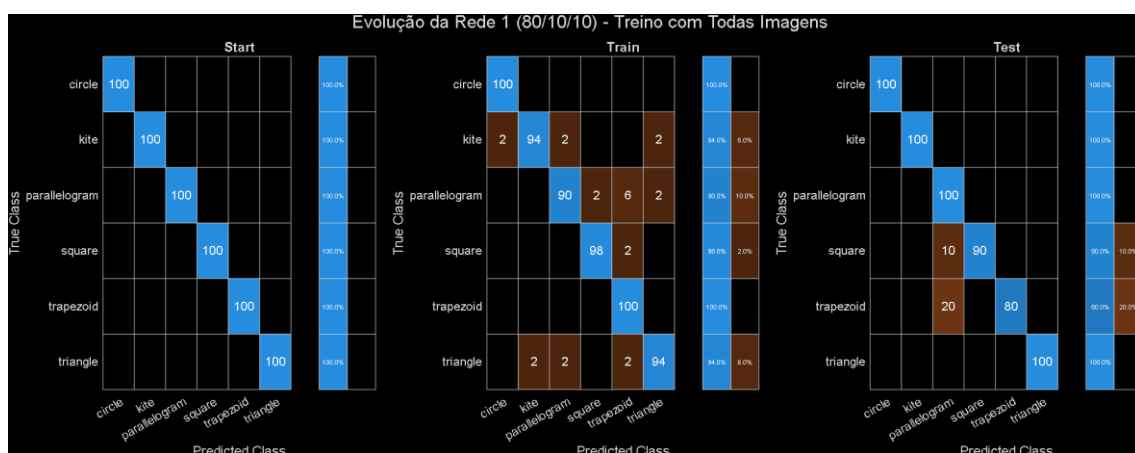
Mais fáceis de reconhecer:

- Papagaios (100% em todas as redes no teste)
- Círculos (100% nas Redes 1 e 2)
- Paralelogramos (100% nas Redes 1 e 3)

Mais difíceis:

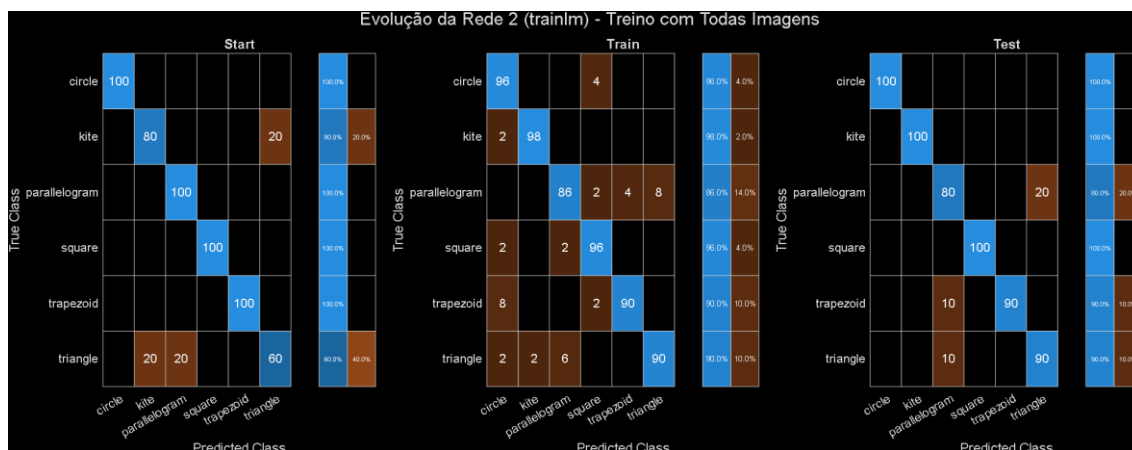
- Triângulos (especialmente no início do treino)
- Trapézios (80% nas Redes 1 e 3)

Evolução das Redes



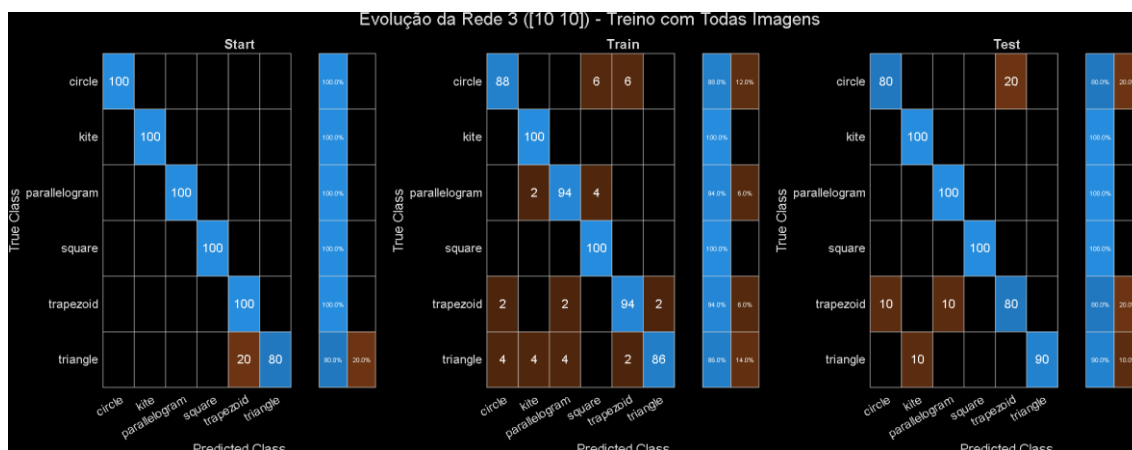
Rede 1 (80/10/10):

- No início (Start): Perfeita a reconhecer todas as formas (100%)
- No treino: Muito boa (96%), com pequenas falhas nos papagaios e paralelogramos
- No teste: Ótima generalização, com quadrados e trapézios a mostrarem alguns erros



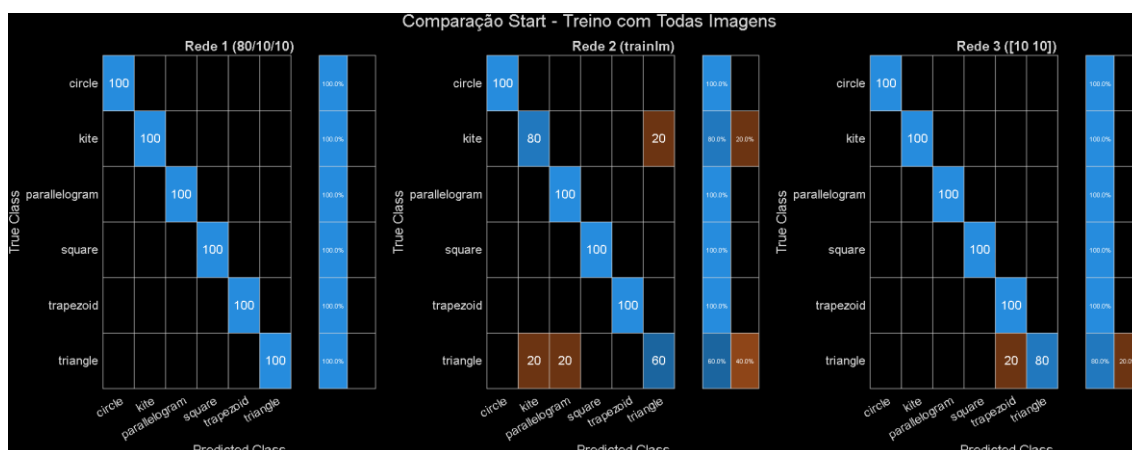
Rede 2 (trainlm):

- No início: Boa no geral (90%), mas com problemas a identificar triângulos (40% de erro)
- No treino: Melhorou bastante, especialmente nos triângulos
- No teste: Excelente nos círculos, papagaios e quadrados (100%), alguma confusão entre paralelogramos e triângulos

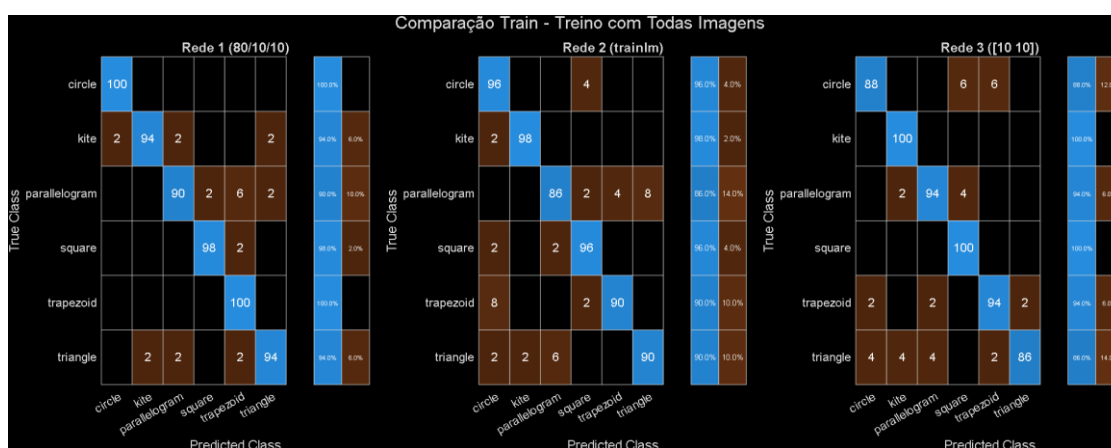


Rede 3 ([10 10])

A rede começou muito bem, com pequena dificuldade apenas nos triângulos. No treino, manteve bom desempenho geral. No teste final, mostrou reconhecimento perfeito para três formas, mas teve mais dificuldade com círculos e trapézios. Particularmente, os círculos pioraram no teste comparado ao início.



A Rede 1 (80/10/10) destacou-se por reconhecer perfeitamente todas as formas no conjunto Start (100% em todas as classes), enquanto as Redes 2 e 3 apresentaram dificuldades significativas com triângulos (apenas 60% e 80% de precisão, respetivamente) e a Rede 2 também teve problemas com papagaios (80%).



No conjunto de treino, todas as redes mostraram bom desempenho na classificação das formas (acima de 93% no global), com a Rede 1 destacando-se nos círculos (100%) e trapézios (100%), a Rede 2 nos papagaios (98%) e a Rede 3 nos papagaios e quadrados (ambos 100%).

Os triângulos continuaram a ser a classe mais problemática em todas as redes, especialmente na Rede 3 (86%), enquanto os paralelogramos apresentaram confusão significativa na Rede 2 (86%).

iv.) As três melhores redes obtidas nesta experimentação foram guardadas na pasta “Melhores Redes Finais”

Alinea d)

O nosso script da desta alinha não está a funcionar, devido a escolhas do framework anteriores e a falta de tempo para as mudar, o que submetemos é o que conseguimos fazer com informação que encontramos online.

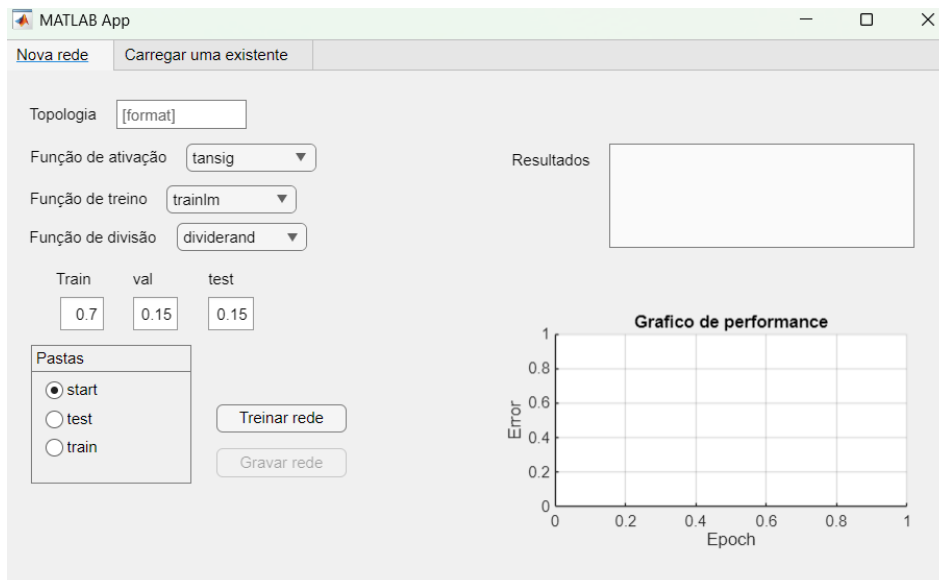
Aplicação

A aplicação está organizada em duas janelas principais:

- **Janela de Nova Rede**

Destina-se à criação e treino de novas redes neuronais. Nesta interface, o utilizador pode:

- Definir os parâmetros da rede (funções de ativação, valores de divisão dos dados, etc.).
- Visualizar a **precisão** e o **erro** da rede após o treino.

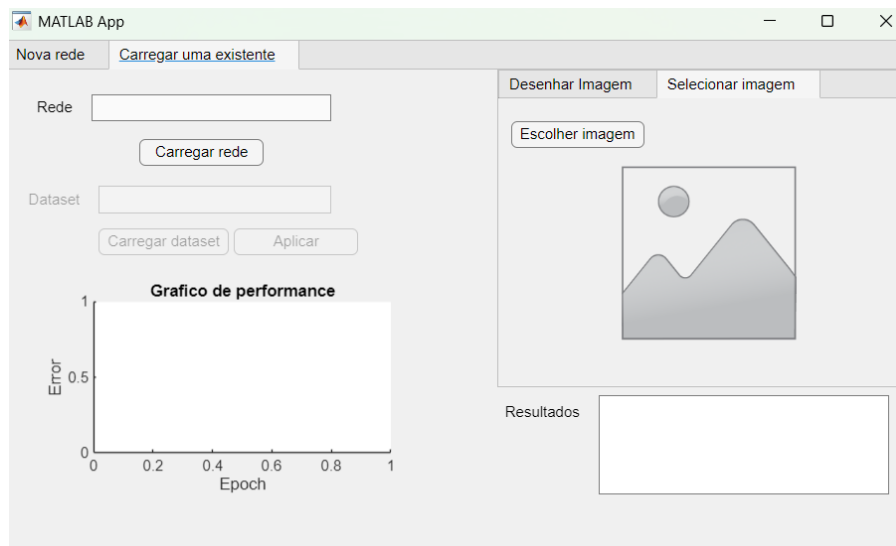


- **Janela de Carregar Rede Existente**

Permite carregar uma rede previamente treinada e aplicá-la a um *dataset*.

Contudo, esta funcionalidade apresenta atualmente **limitações**:

- Problema técnico:** A falta de um cálculo genérico para os *inputs* está a causar instabilidades (devido a restrições de tempo no desenvolvimento).
- Funcionalidade incompleta:** Inclui um *sub-tab group* onde seria possível desenhar/selecionar uma imagem e processá-la com a rede. Esta parte também não foi finalizada por falta de tempo.



5. Conclusões

Este trabalho demonstrou a importância de uma abordagem estruturada no desenvolvimento de redes neurais, mostrando como pequenas alterações nos parâmetros podem ter um impacto significativo nos resultados e que para conseguir atingir uma rede neuronal precisa não é necessariamente uma rede com muitos recursos como é provado na alínea a. Embora tenham sido alcançados bons níveis de precisão nas tarefas feitas, há margem para melhorias, especialmente na aplicação e na alínea d.