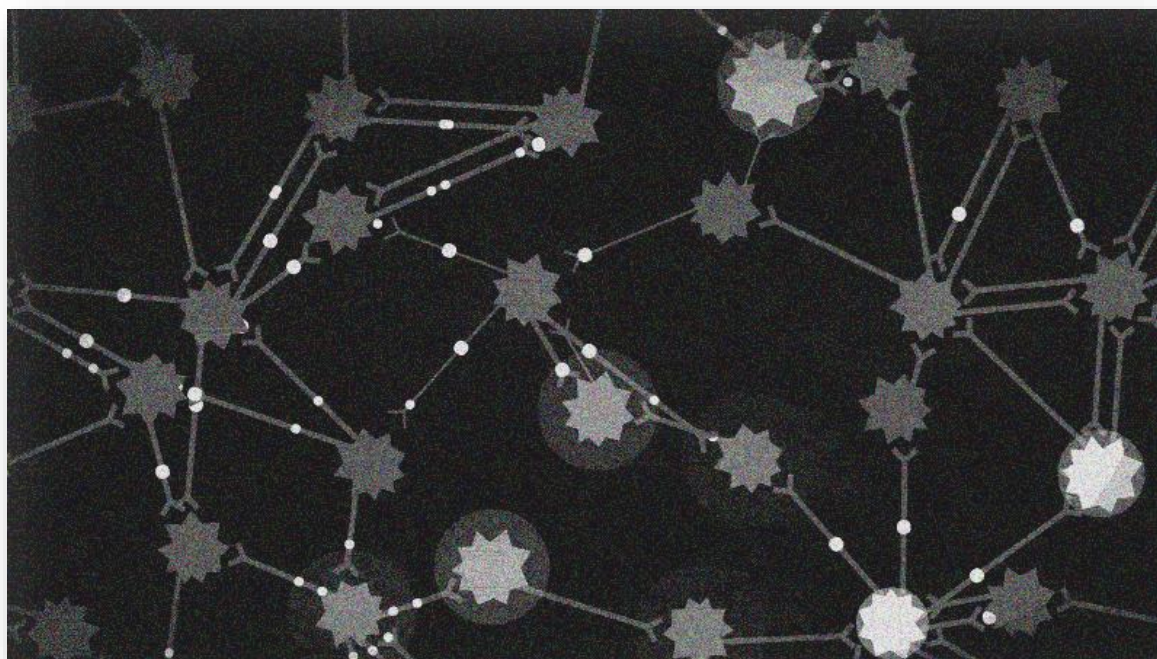


# Conhecimento e Raciocínio

## 2021/22 | LEICE

### Redes Neurais



Docente: Viriato Marinho Marques

Docente: Anabela Simões (prática)

Discentes: Pedro Praça 2020130980

## Índice

---

Índice .....	2
1. Introdução .....	3
2. Conversão e tratamento das imagens .....	4
3. Treino e Estudo Estatístico .....	5
3.1 Pasta start.....	6
3.2 Pasta train .....	7
3.3 Pasta test.....	8
3.4 Pasta start_train_test.....	8
3.5 Pasta images_draws.....	8
4. Interface Gráfica.....	9
5. Conclusão .....	10
6. Anexos.....	11
Médias.....	11
Resultados Obtidos .....	11
Script obtemImagens .....	11

## 1. Introdução

---

Este trabalho, foi desenvolvido no âmbito da Unidade Curricular de Conhecimento e Raciocínio.

Tem como objetivo o desenvolvimento e estudo estatístico de redes neuronais para classificar algumas formas geométricas. A linguagem e tecnologia usada para este efeito é o Matlab.

## 2. Conversão e tratamento das imagens

---



O *dataset* usado no trabalho é constituído por 5 pastas, 3 delas fornecidas com o enunciado, cada qual com várias imagens de formas geométricas.

De modo a diminuir os custos de tempo e memória de treino, as imagens, ao serem lidas pelo programa, são redimensionadas para 32x32 através das funções da *toolbox* de *image processing* do Matlab.

De seguida, são colocadas em matrizes binárias e convertidas em uma única coluna, pois as redes neuronais recebem os seus *inputs* na forma de colunas de uma matriz.

As imagens no trabalho são lidas em grupos de 6 classes (cada classe é uma forma geométrica diferente).

Os *targets* da rede neuronal são criados usando matrizes de 6 por 30 (no caso da pasta *start*, sendo que se fosse da pasta *train* seria 6 por 300) . A matriz de *targets* teve de ser invertida (utilizando a transposta “ ’ ”) pois ficava com tamanhos diferentes tornando-se incompatível.

Cada coluna da matriz de *targets* terá 6 linhas, ou seja, uma para cada classe (forma geométrica), esta preenchida com o valor 0 ou 1.

Caso seja 1, quer dizer que as imagens correspondentes contêm a forma geométrica que a linha representa, caso contrário fica com o valor 0.

### 3. Treino e Estudo Estatístico

---

O treino para cada pasta foi feito de maneira minuciosa, respeitando o requisito da alínea b), referente à pasta ***train***.

Estes testes foram feitos 5 vezes para cada configuração, calculando-se a média logo a seguir, tendo o cuidado de assegurar a melhor rede para um próximo estudo, garantido assim a precisão elevada dos estudos.

Todos os resultados obtidos e as médias para cada configuração encontram-se em anexo, acessíveis através das ligações [resultados obtidos](#) (testes) e [médias](#).

Estes estudos foram feitos sempre em relação à configuração por defeito que tem o seguinte aspeto:

	Número de camadas escondidas	Número de neurónios	Funções de ativação	Função de treino	Divisão dos exemplos
<b>Configuração por defeito</b>	1	10	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}

### 3.1 Pasta start

Durante o processo de estudo da rede neuronal, foram testadas várias topologias e funções de ativação, bem como diferentes tipos de rede e de treino.

Relativamente à configuração por defeito, o número de dimensão das camadas escondidas afetou ligeiramente o desempenho negativamente no que toca ao tempo de treino, o qual aumentou quando se acrescentavam mais camadas. Ainda assim, a precisão global foi sempre de 100.00%.

O mesmo aconteceu para as funções de treino, nomeadamente na ***trainbfg***, que teve um tempo de treino muito elevado.

Nas funções de ativação verificou-se que, quando se usou a função ***purelin*** na camada de output, obtiveram-se melhores resultados, tendo o desempenho aumentado significativamente (um aumento de 80%).

No entanto, estes testes efetuados nesta pasta não são muito significativos pois as imagens são em menor número e estão a ter acesso a todos os exemplos da pasta, logo os valores de desempenho são ótimos.

### 3.2 Pasta train

Esta pasta teve o mesmo processo que a pasta **start**, tendo sido efetuados testes ainda mais minuciosos, diferentes segmentações na divisão dos exemplos e também várias arquiteturas de rede (número de camadas/nº de neurónios).

O número e dimensão das camadas escondidas tiveram um impacto positivo no desempenho da rede, até um certo ponto. Como aconteceu com a pasta **start**, também houve um elevado custo temporal associado.

Nenhuma função de treino conseguiu alterar os bons resultados da **trainlm**, pois todas elas produziram resultados inferiores a essa configuração *default*, portanto concluí que, neste caso, nenhuma destas funções de treino foram boas para o desempenho.

Já as funções de ativação produziram resultados positivos, superando aqueles resultantes da configuração por *default*. Mas tal e qual como na pasta **start**, a função **purelin**, na camada *output*, trouxe os melhores resultados.

A divisão de exemplos pelos conjuntos tornou-se possível devido ao maior tamanho do *dataset* da pasta **train**. Depois de dividir os exemplos pelos três conjuntos (treino, validação e teste) obtive o melhor desempenho nesta pasta com uma **precisão global** de **96,66%** e uma **precisão de teste** de **70,66%**. Isto quando se utilizou a seguinte parametrização: *dividerand* = {0.9, 0.05, 0.05}.

As outras parametrizações revelaram-se bastante positivas superando assim a configuração *default* na maior parte dos casos.

### 3.3 Pasta test

Este *dataset* foi usado para testar as melhores redes obtidas até ao momento. Estas tiveram uma *accuracy* de pelo menos 70%, ou seja, acertaram no mínimo 70% das imagens das três pastas. Sendo que houve uma rede que acertou 99% das imagens da pasta **train**.

De seguida treinei novamente a rede, mas apenas para os exemplos da pasta **test**. Após isso, a rede foi testada para as imagens de cada pasta. Observei um bom desempenho, sempre acima dos 70%, sem muita discrepância nos resultados.

### 3.4 Pasta start\_train\_test

Esta pasta continha todas as imagens fornecidas, perfazendo 390 imagens de formas geométricas.

A rede voltou a ser treinada com este novo *dataset*, e mostrou resultados de pelo menos 80% de *accuracy*. Sendo que a rede que se destacou das restantes foi a rede 5 ("melhor\_net\_alinea\_c\_3ponto.net"), que atingiu os 90%.

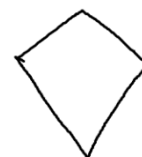
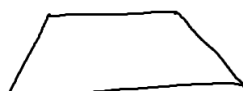
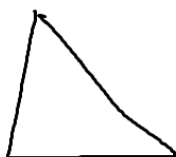
Devido ao maior tamanho do *dataset* disponível, concluí que, quanto maior for o número de exemplos para treino disponíveis à rede, melhor o seu desempenho.

### 3.5 Pasta images\_draws

A última avaliação das redes neuronais, onde foi usada a melhor rede obtida até ao momento na identificação das formas geométricas, teve como *dataset*, 30 imagens desenhadas por mim.

Este *dataset* é composto por várias formas geométricas, desenhadas no *Paint*, umas mais perceptíveis que outras.

A rede conseguiu identificar 9 de 30 formas geométricas. Não teve um desempenho muito bom, apesar que nem todas essas imagens têm o mesmo tamanho e não estão perfeitos comparativamente às formas geométricas dadas.





## 4. Interface Gráfica

The screenshot shows the MATLAB App interface for a neural network. The interface is organized into several sections:

- Configuration Section:**
  - Pasta de teste:** A dropdown menu set to 'start'.
  - Tipo de Rede:** A dropdown menu set to 'Feed Forward Net'.
  - Nº de Camadas Escondidas:** A dropdown menu set to '1', with three input fields below it containing the value '5'.
  - Função de Divisão:** A dropdown menu set to 'sem\_funcao'.
  - Função de Ativação da Camada Escondida:** Three dropdown menus, all set to 'logsig'.
  - Função de Ativação da Camada de Output:** A dropdown menu set to 'tansig'.
  - Função de Treino da Rede:** A dropdown menu set to 'trainlm'.
  - train, val, test:** Three input fields with values '0.15', '0.15', and '0.3' respectively.
  - Nº max de epochs:** An input field with the value '400'.
- Buttons and File Fields:**
  - Guardar Rede:** A button with a text field 'Nome do Ficheiro' below it.
  - Carregar Rede:** A button with a text field 'Nome do Ficheiro' below it.
  - Treinar Rede:** A button.
- Simulation and Results Section:**
  - Precisão Total Treino:** An input field.
  - Precisão Testes Treino:** An input field.
  - Epochs de Treino:** An input field.
  - Pasta de teste:** A dropdown menu set to 'start'.
  - Precisão Total do Teste:** An input field.
  - Simular:** A button.
  - Nome do Ficheiro:** A text field with a **Guardar Rede** button below it.

O trabalho contém uma interface gráfica que permite criar, treinar, carregar, simular e guardar as redes neuronais.

Foi criado um *script* para que a aplicação conseguisse ter acesso às imagens de cada pasta, contendo uma função denominada "[obtemImagens](#)", que recebe o nome da pasta ("nomePasta") e devolve o *target* e o seu *input*.

No entanto, não foi concretizada a parte de avaliação das formas geométricas desenhadas nesta aplicação.

## 5. Conclusão

---

Com este trabalho concluí que os dados inseridos nas parametrizações são cruciais no seu desempenho, assumindo grande importância.

O correto tratamento de imagens também é algo que afeta o desempenho do mesmo.

No geral, a implementação deste trabalho prático permitiu-me adquirir vários conhecimentos importantes para esta área.

Globalmente, creio ter cumprido com a maior parte dos requisitos do enunciado, obtendo um trabalho consistente e de qualidade.

## 6. Anexos

---

### Médias



medias\_alinea\_b.xls  
x

### Resultados Obtidos



testes\_alinea\_a.xlsx



testes\_alinea\_b.xlsx



testes\_alinea\_c.xlsx



testes\_alinea\_c\_3po  
nto.xlsx

### Script obtemImagens



obtemImagens.m