

---

# Sistemas Baseados em Inteligência Computacional

## Trabalho 3 - Redes Neurais

Eduardo Gonçalves - up201706230

Nuno Minhoto - up201604509

Maio 2021

### 1 Exercício 2

#### 1.1 Qual a necessidade de usar uma rede neuronal multi-camadas?

As redes neuronais constituídas por apenas uma camada apenas podem ser utilizadas para dados que sejam linearmente separáveis. No caso dos dados fornecidos para este exercício 2, esta separação não pode ser feita de maneira linear, pelo que tivemos de recorrer a uma rede neuronal com mais do que uma camada.

Com a visualização dos dados disponibilizados, chegamos à conclusão de que a sua separação só poderia ser feita através de uma curva que seria composta por, no mínimo, 5 rectas que se intersectam 4 vezes.

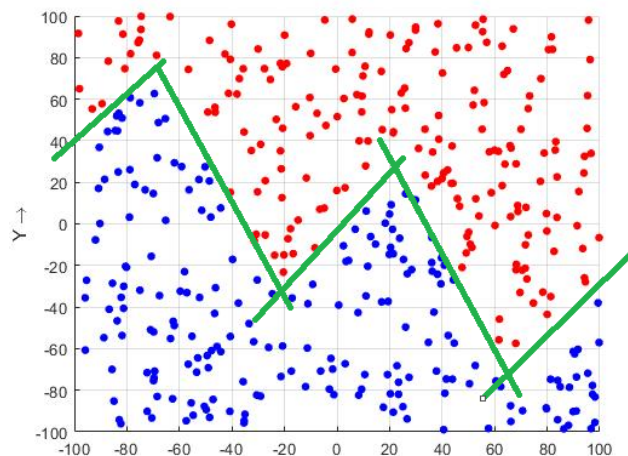


Figura 1: Separação possível dos dados B

Após a análise dos dados e pesquisa sobre o tema, chegamos à conclusão de que a melhor solução seria usar duas camadas escondidas, tendo a primeira camada 5 neurónios (igual ao número de rectas) e a segunda 3 que juntando ao neurónio de saída fazem as 4 intersecções evidenciadas na figura 1

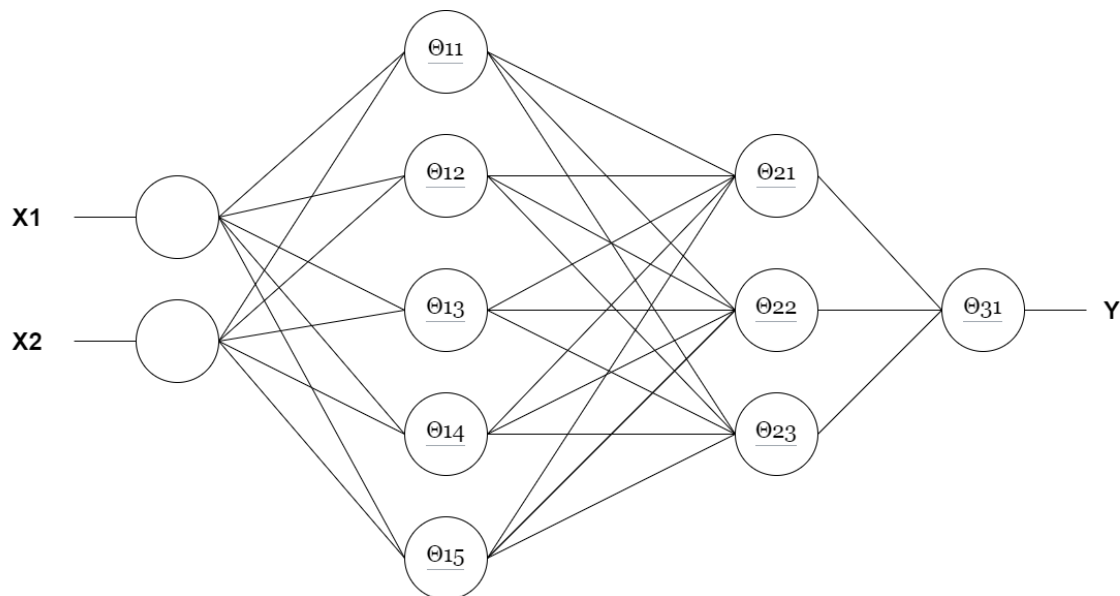


Figura 2: rede neural utilizada

A rede fica completa com o neurónio de saída, que representa a última intersecção ( $3+1=4$ ).

## 1.2 Algoritmo

### Leitura e processamento inicial dos dados

Começamos por fazer a leitura dos dados, normalizando-os consoante a função de ativação escolhida (Tangente Hiperbólica ou Sigmóide). Assim, para o caso da Tangente Hiperbólica, consideramos que a entrada deveria assumir valores entre -1 e 1. Para a função Sigmoid, a normalização foi feita para valores entre 0 e 1. É ainda feita uma inicialização aleatório dos pesos.

### Pesos, Bias, Taxa de aprendizagem e número máximo de iterações de treino

Os pesos e bias a utilizar são gerados aleatoriamente pelo script podendo estes estar dentro de um intervalo dependendo da função de activação a utilizar. Para a função Tangente Hiperbólica o intervalo de valores é  $[-0.25, 0.25]$  e para a função Sigmoid o intervalo é  $[0, 0.25]$ . Quanto à taxa de aprendizagem e ao número máximo de iterações são parâmetros definidos no início do script e que podem ser alterados pelo utilizador. Os valores que foram escolhidos e utilizados nos testes, tanto para estes parâmetros como para os intervalos dos pesos e bias, foram obtidos através de várias tentativas e experiências feitas de maneira a encontrar valores que permitiam obter os melhores resultados.

### Forward Pass e Backpropagation

Após as leituras e normalizações dos dados, utilizamos um ciclo que representa o cálculo dos valores à saída da rede consoante as entradas (Forward Pass) e ao ajuste dos pesos das ligações entre neurónios e Biases (Backpropagation). Também procedemos, dentro desse ciclo, ao cálculo da derivada da função de ativação. No final de cada ciclo a saída é ainda ajustada para que o seu valor seja -1 ou 1 e é calculado o número de erros do treino.

### Teste e resultados

Posteriormente ao treino da rede, avançamos com o teste utilizando os inputs fornecidos no ficheiro de dados. De seguida comparamos os valores obtidos na saída da rede com os valores correctos fornecidos para a saída, de onde obtivemos os seguintes resultados:

Tabela 1: Resultados obtidos com a função de ativação TANH

Dados	Iterações	Erros no treino	Erros no teste	Taxa de acerto
A	193	0	0	100%
B	3000	1	0	100%
C	50	0	0	100%

Tabela 2: Resultados obtidos com a função de ativação SIGMOID

Dados	Iterações	Erros no treino	Erros no teste	Taxa de acerto
A	1972	0	0	100%
B	3000	38	2	96%
C	1377	0	0	100%

## 2 Exercício 3

### Procedimento Geral

Utilizamos, neste exercício, a mesma rede que foi projetada para o exercício 2, visto que, nesta situação, era mais difícil analisar previamente os dados (as saídas eram contínuas). As diferenças no código encontram-se apenas na normalização dos dados (cujos limites nos eram fornecidos no enunciado), no cálculo do erro quadrático médio (MSE) e, obviamente, na leitura dos dados. Implementamos também uma função responsável pela "desnormalização" dos dados da saída, ajustando-os, assim, aos dados de saída do treino.

Com cerca de 6000 iterações e um learning rate de 0,1 conseguimos erros consistentemente abaixo de 15%, mas muito raramente abaixo de 10%.

### Teste e Resultados

Numa das iterações obtivemos os seguintes resultados, com  $MSE = 7,2\%$ :

Tabela 3: Resultados obtidos com a função de ativação TANH

Saída Esperada	Resultados	Erro Relativo
16,96	17,06	0,56%
12,04	12,81	6,40%
17,08	17,13	0,29%
19,20	18,95	1,32%
16,98	17,07	0,52%
16,50	16,55	0,29%
17,07	17,12	0,31%
18,05	18,57	2,85%
17,02	17,10	0,46%

## Anexos

### Dispersão dos dados usados no exercício 2

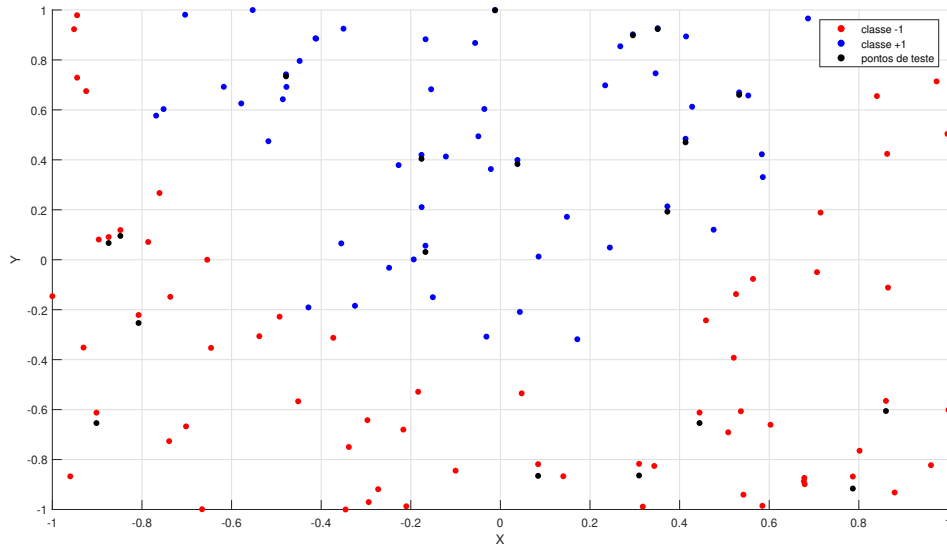


Figura 3: Plot dos dados de entrada 11A

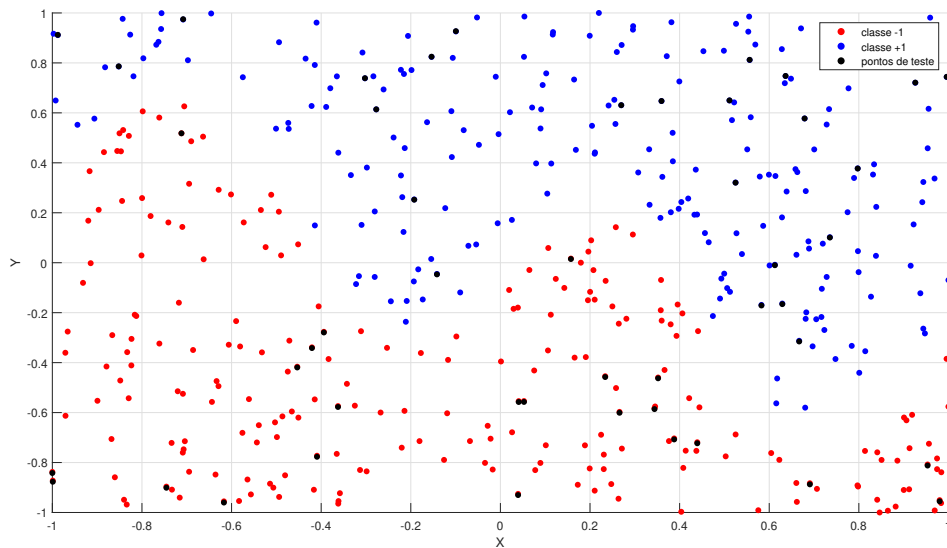


Figura 4: Plot dos dados de entrada 11B

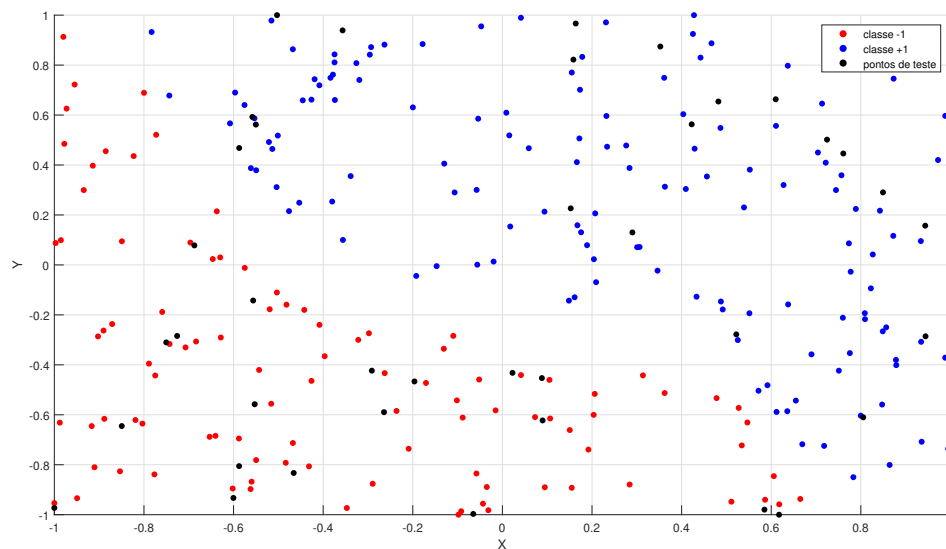


Figura 5: Plot dos dados de entrada 11C