

## APVC – Exercícios – Redes neuronais

### Exercício 1 (Rede neuronal para classificação multiclasse)

Neste exercício vai tomar um contacto maior com a construção, treino e visualização dos resultados de um classificador implementado com uma rede neuronal. Para tal irá explorar o script utilizado durante a demonstração feita na aula: `wineNet.py`.

Recorde que neste problema pretende-se identificar o fabricante de vinho com base em características químicas do próprio vinho. O *dataset* utilizado é o WineData<sup>1</sup>, que consiste em amostras de 178 vinhos produzidos por 3 fabricantes diferentes.



Neste exercício, para além do script `wineNet.py`, irá precisar de colocar o ficheiro `wineData.csv` na mesma localização em que for executado o script.

- 1) Verifique as dimensões dos conjuntos `x_train` e `y_train`, usando o atributo `shape` das matrizes `numpy`. Visualize o conteúdo das 5 primeiras linhas de cada uma das matrizes. Interprete a organização das matrizes.
- 2) Visualize o sumário do modelo. Interprete a informação que é apresentada e justifique com cálculos a razão para haver 280 parâmetros para treinar na camada “escondida” e 63 na camada de output (note que estes parâmetros são os pesos das ligações aos neurónios).
- 3) Experimente treinar com variações aos seguintes parâmetros, verificando o que acontece aos resultados, assim como à estabilidade e rapidez de convergência no processo de treino:
  - a. Taxa de aprendizagem do algoritmo de otimização (`learning_rate`) – utilize por exemplo 0.01. e 0.0001.
  - b. Parâmetro `batch_size` na função `fit(...)`. Use um valor mais baixo (4) e um valor mais alto (64).
  - c. Número de neurónios da camada escondida da rede – altere para 3 neurónios e para um número exageradamente alto (e.g., 300).
- 4) Será que são necessárias 500 épocas para treinar o modelo? Justifique com base nos gráficos.

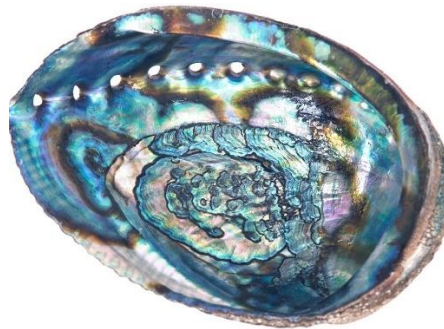
---

<sup>1</sup> Informação adicional em: <https://archive.ics.uci.edu/dataset/109/wine>

## Exercício 2 (Rede neuronal para regressão)

Neste exercício pretende-se desenvolver uma rede neuronal para regressão. O *dataset* que será utilizado é o *Abalone*<sup>2</sup>.

O objetivo é estimar a idade de um molusco da espécie *Abalone*, com base em medidas simples como a altura e diâmetro da concha, peso, etc. Para estimar a idade de uma forma mais precisa seria necessário serrar a concha e examinar o número de camadas (anéis) ao microscópio, processo esse que danificaria a concha e que apenas poderia ser feito após a morte do animal.



- 1) Copie os conjuntos de treino e teste (*abalone\_train.csv* e *abalone\_test.csv*) assim como o script *abaloneNet.py* para o seu projeto.
- 2) Verifique que os conjuntos de treino e teste estão a ser bem carregados, correndo o script *abaloneNet.py*. Caso tenham sido bem carregados os conjuntos de treino e teste deverão conter 3320 e 850 amostras, respetivamente.
- 3) Complete o script *abaloneNet.py* de forma a:
  - a. Definir a arquitetura de uma rede neuronal simples para regressão;
  - b. Compilar o modelo, utilizando o erro quadrático médio na função de perda – `keras.losses.MeanSquaredError()` – pois esta será adequada para uma regressão;
  - c. Treinar o modelo, ajustando parâmetros de forma a conseguir chegar a valores de *loss* abaixo de 5.0 no conjunto de treino.
- 4) Calcule as predições com base nas amostras do conjunto de teste e visualize-as num gráfico onde apareçam os valores verdadeiros da idade do *Abalone* vs. os valores da idade que foram estimados pela sua rede.

Nota: para construir o gráfico pode utilizar uma instrução idêntica a:

```
plt.plot(y, y_pred, 'bo')
```

*y* e *y\_pred* são os valores da idade verdadeiros e estimados, respetivamente; o parâmetro **'bo'** faz com que os pontos do gráfico sejam desenhados como pequenos círculos azuis.

---

<sup>2</sup> Informação adicional em: <https://archive.ics.uci.edu/dataset/1/abalone>