

APVC – Exercícios

Modelos pré-treinados e transferência de conhecimento

Exercício 1 (Rede CNN pré-treinada – VGG16 e MobileNet)

Neste exercício pretende-se que utilize diretamente um modelo de CNN pré-treinado no conjunto IMAGENET, mas para classificar um conjunto de imagens escolhidas por si.



Utilize como base o script `pretrained.py` (ou o *notebook*) que contém uma importação do modelo VGG16.

- 1) Consulte <https://deeplearning.cms.waikato.ac.nz/user-guide/class-maps/IMAGENET/> e veja a lista de 1000 classes associadas ao *dataset* ImageNet. Procure na internet algumas imagens (sugere-se à volta de 5) que encaixem nessas classes e descarregue-as.
- 2) Verifique o resultado da classificação das imagens que descarregou usando a rede pré-treinada VGG16. Atenção ao formato dos dados de entrada – é necessário realizar um passo de pré-processamento para que estes fiquem no formato que o modelo é suposto receber.
- 3) Altere a rede pré-treinada para uma das versões do modelo MobileNet. Consulte a documentação em https://www.tensorflow.org/api_docs/python/tf/keras/applications para verificar os argumentos que pode passar quando se cria o modelo e para verificar se a versão que escolheu necessita ou não de algum tipo de pré-processamento aos dados.
- 4) Repita o ponto 2, usando agora a versão da MobileNet que escolheu. Verificou diferenças significativas?

Exercício 2 (Transferência de conhecimento)

Neste exercício pretende-se treinar uma rede neuronal convolucional, usando transferência de conhecimento, para distinguir entre imagens de flores.



- 1) Descarregue o *dataset flower_photos* (disponibilizado no Moodle, no módulo da semana anterior) para o seu sistema de ficheiros. Este conjunto de dados é composto por 5 classes de imagens de flores, com um número de imagens por classe que varia entre 631 e 898, perfazendo um total de 3670 imagens.
- 2) Utilize a função `keras.utils.image_dataset_from_directory(...)` para construir dois datasets: um para treino e outro para validação, com base nas imagens em disco. Pode usar como base o script `flowerNet.py` (ou o *notebook*) da semana anterior.
- 3) Implemente uma rede neuronal baseada em transferência de conhecimento, utilizando a MobileNet como base. Treine a rede e verifique os resultados. Deverá conseguir uma melhoria bastante significativa face à rede neuronal originalmente implementada no script `flowerNet.py`.
- 4) Experimente utilizar um outro modelo de rede pré-treinada – sugere-se por exemplo uma das versões mais simples da EfficientNet. O Tensorflow disponibiliza várias arquiteturas (ver https://www.tensorflow.org/api_docs/python/tf/keras/applications). Note que poderá ser necessário realizar algumas adaptações ao seu código, nomeadamente ajustar a dimensão das imagens. Atenção que o tempo de treino poderá ser significativamente maior em algumas arquiteturas – evite por isso modelos que conduzam a tempos de treino por época superiores a 2 minutos.