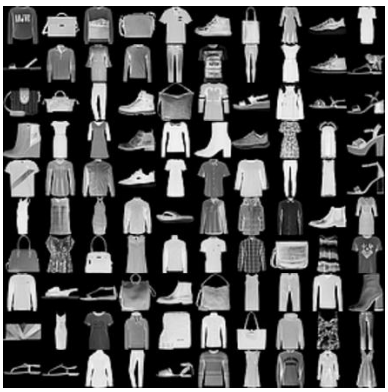
APVC – Exercícios – Redes neuronais convolucionais

Exercício 1 (CNNs vs. rede *shallow*)

Neste exercício irá comparar os resultados obtidos em classificação de imagens usando uma redes neuronal clássica (*shallow*) e uma rede neuronal convolucional.

Para tal será novamente utilizado o dataset FASHION_MNIST do Desafio 1. Relembre que este conjunto consiste em 70.000 mini-imagens de 28x28 pixels que representam roupas, sapatos e acessórios (10 classes), originalmente divido em 60.000 imagens para treino e 10.000 para teste.



FASHION_MNIST

1) Esta alínea refere-se à rede neuronal "*shallow*" multiclasse que desenvolveu no Desafio 1. Caso não a tenha disponível, poderá utilizar como base o script `fashionNet_v2.py`, que implementa uma rede neuronal simples com 128 neurónios na camada "escondida" e 10 na camada de saída. Treine usando os conjuntos de treino (e validação) e depois avalie o desempenho obtido no conjunto de teste (e anote esse desempenho). Caso tenha "à mão" os resultados que obteve no Desafio 1, não necessita de realizar esta alínea.

2) Pretende-se agora substituir a rede neuronal "*shallow*" por uma rede neuronal convolucional (CNN). As características da CNN a implementar são as seguintes: tem duas camadas convolucionais intercaladas por camadas de subamostragem *Max Pooling* (2->1), 128 neurónios na camada densa e 10 na camada de output. Na primeira camada convolucional deverá ter 16 filtros 3x3 e na segunda 32 filtros 3x3.
    a. Com base na topologia da rede convolucional, estime o número de parâmetros (pesos) da rede que precisam de ser otimizados. Primeiro faça as contas! E depois confirme através do método `summary()` do modelo da rede neuronal.
    b. Implemente a rede neuronal descrita.
    c. Treine a rede e verifique o desempenho no conjunto de teste. Melhorou o desempenho em relação ao que foi obtido com a rede neuronal mais simples?

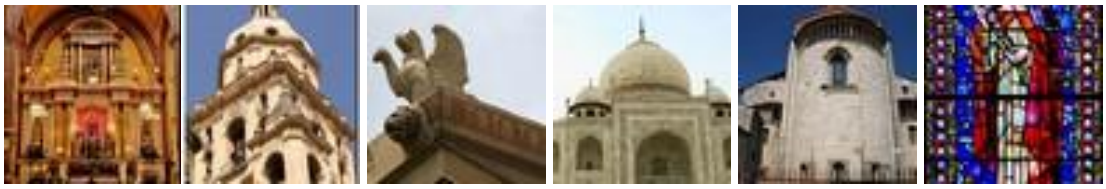
3) Tente melhorar o desempenho da sua CNN introduzindo *Dropout* e de *Data Augmentation*. Poderá também experimentar algumas alterações estruturais à rede tais como variar a dimensão e número de filtros das camadas convolucionais, introduzir camadas convolucionais adicionais antes de cada camada de *pooling*, alterar o número de neurónios na camada densa, etc.

Exercício 2 (CNN para classificação de imagens de monumentos)

Neste exercício pretende-se classificar mini-fotografias que mostram pormenores de monumentos.

Para realizar o exercício necessita de descarregar o *dataset* `cultural_heritage_images64`[1], disponibilizado na página de APVC no moodle. Este *dataset* consiste em 10 classes (desequilibradas) de imagens que correspondem a elementos que fazem parte de monumentos ou outros edifícios com valor cultural, como por exemplo campanários, altares, cúpulas, gárgulas, vitrais, etc.

A dimensão das imagens na versão disponibilizada é de 64x64 pixels (são "mini-imagens"). A organização original do *dataset* inclui um conjunto de treino (que se poderá dividir posteriormente em treino e validação) e um conjunto de teste.



1. Carregue os *datasets* e processe-os de forma a construir conjuntos para treino, validação e teste.
2. Desenvolva e treine uma rede neuronal convolucional para classificar as imagens. A topologia da rede fica ao seu critério.
3. Calcule a taxa de acertos e mostre a matriz de confusão referentes ao conjunto de teste.

Notas/dicas:

- Como neste caso as imagens são carregadas a partir do sistema de ficheiros, sugere-se que utilize como base o script `flowerNet.py`, onde o procedimento de construção automática do *dataset* é feito de uma forma idêntica ao que necessita de ser feito neste caso.

- O conjunto de teste deve ser carregado com a opção `shuffle=False`, e sem definir as opções `validation_split`, `seed` e `subset`.

- Este conjunto de imagens é mais difícil de classificar do que os anteriores, por isso não coloque as expectativas muito altas (uma taxa de acertos no conjunto de teste perto dos 75% já não é nada má!).

[1] https://www.kaggle.com/datasets/ikobzev/architectural-heritage-elements-image64-dataset