



Universidade do Minho

Departamento de Informática

Mestrado em Matemática e Computação

Perfil de Machine Learning: Fundamentos e Aplicações

Classificadores e Sistemas Conexionistas

1º Ano, 2º Semestre

Ano letivo 2023/2024

Enunciado Prático nº 4

14 de março de 2024

Tema

Redes Neurais Artificiais usando *tf.Keras*.

Enunciado

Pretende-se, com esta ficha, que seja implementada uma *Multi-Layer Perceptron* recorrendo à API de alto-nível do *TensorFlow*, *tf.Keras*, para previsão e classificação de peças de roupa, como sapatos, camisas ou vestidos.

Tarefas

Esta ficha encontra-se dividida em duas partes distintas.

1. A primeira parte desta ficha prática consiste em adicionar, à script do Enunciado Prático nº 3, um *callback* que permita, aquando da execução de alto-nível, guardar ficheiros checkpoints a cada 5 epochs;
2. A segunda parte desta ficha prática consiste em prever e classificar peças de roupa disponibilizadas pelo *dataset* Fashion MNIST. Este *dataset* contém 70000 imagens, em tons de cinza, de 10 categorias diferentes. Estas categorias são, ordenadamente, as seguintes: ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']. Cada imagem tem uma resolução de 28 por 28 pixels. Posto isto, o modelo a desenvolver deve:
 - a) Carregar os dados recorrendo à função *tf.keras.datasets.fashion_mnist.load_data()* cuja assinatura é semelhante à utilizada para o *dataset* MNIST;
 - b) Implementar uma função que permita visualizar as primeiras 16 imagens do *dataset* a trabalhar (semelhante ao apresentado na Figura 1);
 - c) Normalizar os dados de treino (x_{train}) e teste (x_{test}) de forma que passem de uma matriz de 28x28 para um vetor normalizado de 784 pixels (função *reshape*);
 - d) Implementar uma rede *Multi-Layer Perceptron* recorrendo à API sequencial do *tf.Keras* - a rede deverá possuir, no total, 3 camadas;
 - e) Compilar o modelo usando o otimizador Adam, *sparse categorical crossentropy* como função de perda, e *accuracy* como métrica. Deverão treinar o modelo durante 5 epochs e utilizar 10% dos dados de treino para validação. Deverão também avaliar a *accuracy* do modelo em dados desconhecidos;
 - f) Correr o modelo numa ferramenta *cloud* como o Google Colab ou o Kaggle;
 - g) Implementar a mesma rede *Multi-Layer Perceptron* recorrendo à API funcional do *tf.Keras*;
 - h) Fazer o *tuning* manual da rede, experimentado e documentando várias combinações de hiper-parâmetros e os resultados obtidos.



Figura 1. Peças de roupa do *dataset* Fashion MNIST.