Mestrado em Engenharia Informática

Análise de Dados
2022/2023

Worksheet 5

CNN for Image Classification
Part II

The main objective of this class is to implement a CNN from scratch, understanding the impact that its main components have on network performance. The definition of the model will be performed using Keras Functional API. Specifically, the following topics will be addressed:

1. Definition, training and validation of a CNN architecture
2. Using Keras Functional API
3. Strategies to combat overfitting: Data Augmentation

Flowers Dataset
Image Classification

In this class, a dataset with color images of flowers will be used. The dataset has 3670 images divided into 5 classes: ['daisy', 'dandelion', 'roses', 'sunflowers', 'tulips']. You can find some information about the dataset here:
https://www.tensorflow.org/tutorials/images/classification

The base code for this example is available here:
https://github.com/FranciscoBPereira/AnaliseDados-MEI-ISEC-2223

1. Dataset fetching and loading

The operations for obtaining and loading the Dataset are identical to the previous class. The usual processing operations are also applied. The original file can be found at this link: https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz

Five folders are created, one for each of the classes. After the original downloading, there is no division between training, validation and test sets. The code provided creates training (80%) and validation (20%) sets from the original images. No test set is used in this class.

2. Creating the baseline version of the Neural Network using the Keras Functional API

a) Create a baseline CNN for this problem using the **Keras functional API**. This is a more flexible method, which allows the definition of graph structures, with multiple inputs and outputs. Nevertheless, in this class, the model architecture should be sequential.

b) When defining the architecture, you should use combinations of the layers that were presented in the last class. The network size is constrained to 5 million trainable parameters. At this point, you should not use data augmentation.

c) Verify the details of the model's architecture and the specific characteristics of each layer, including their weights. Check that the restriction on the number of parameters is met.

d) Compile the model, using the following components:
   a. *Optimizer*: Adam
   b. *Metrics*: Accuracy
   c. Choose the *loss function* you consider most appropriate

e) Train the model for 20 epochs, using the train and validation sets previously defined. Apply any callbacks that you consider as appropriate.

f) Create charts to visualize, analyze and interpret the results

3. Fighting Overfitting: Data Augmentation

You've probably noticed that your CNN tends to overfit. One of the main problems with this dataset is the reduced number of examples. Using *data augmentation techniques,* we can artificially increase the number of images used during training, which will make the process more robust.

In the tutorial available at the following link, 2 options for using data augmentation are discussed. These 2 approaches were also presented in the theoretical class: https://www.tensorflow.org/tutorials/images/data_augmentation

In this class we will adopt the strategy of adding pre-processing layers to the beginning of the CNN model. During training, these layers will ensure that the network never finds 2 images exactly the same. There is a wide variety of layers that can be parameterized and used (new data augmentation operations can also be created). The list of image augmentation layers can be consulted here: https://keras.io/api/layers/preprocessing_layers/image_augmentation/

a) Create a pre-processing layer consisting of 3 augmentation layers: Flip, Rotation and Zoom.

b) Visualize an example of the changes that are produced by the preprocessing layer

c) Add the pre-processing layer to the beginning of the previously implemented CNN.

d) Compile and train the model, keeping the previous parameterization.

e) Visualize, analyze and interpret the results

What is the impact of Data Augmentation in the performance of a CNN?

This the goal of the second Lab Assignment.