Homework 1: Image classification

Introduction

To tackle the problem we decided to adopt two different approaches:

1. Classical Hand-made Convolutional Neural Network
2. Transfer Learning

Network Design

The first approach involved complete design for both the convolutional part and the fully connected one; the second was based on the InceptionResnetv2 architecture as features extractor, fully designing the FC layer on top and re-training weights of some InceptionResnetv2 convolutional layers.
Fully convolutional method was discarded after a few trials due to the impossibility to reach high levels of accuracy, focusing anyway on the best one.

Data Preparation

Training data were firstly splitted into three different subfolders, according to their labels *(see attached code)*, in order to use the categorical class mode of the data generator.
Input images were colored, rectangular and size-variable: input images were resized to meet the standard format 256x256 rgb and rescaled to get pixel values between 0 and 1.
Validation sets were extracted randomly from this structure (pc=0.15), with fixed random seed in order to compare different experiments.
After trying to analyze just raw data, we chose to apply the following data augmentation techniques in order to achieve a better generalization and reduce overfitting:

- rotation range of 30°
- shift range and height shift equal to 10
- zoom range 0.2
- horizontal flip
- nearest fill mode

Pipeline

In both approaches, the network design follows the same pipeline:

- *achieve good performances* (metrics used: accuracy) on the training set, working on the setting of the hyperparameters. In this stage we analyzed different combinations of layers, neurons per layer (number of filters in convolutional ones) and batch size.

- *reduce overfitting*. This was achieved mainly through the application of kernel regularizers (l2) and dropout layers with high rates (placed in-between fully connected ones). Early stopping was enabled in all significant experiments, based on the validation loss evolution.

Strategies

1. FOR BOTH APPROACHES

    - after careful considerations we reduced the batch size from an initial value of 32 to 8, also testing intermediate values.
    - we tried different optimizers like SGD, Adadelta, Adam with various learning rates and exponential rates. Finally we settled for Adam with exponential rate 0.9 and learning rate 1e-4
    - we tweaked the patience of the early stopping criterion to achieve more interesting results

2. CLASSICAL CASE

    - we tuned the number of initial filters as well as the depth of the convolutional part
    - we also exploited an AveragePooling layer

3. TRANSFER LEARNING

    - we considered a variety of bases such as VGG16, VGG19, MobileNetv2, InceptionResnetV2
    - weights trained on imageNet database were used as starting values in all ‘frozen’ layers
    - two final Dense layers were implemented
    - we adopted a triple fine tuning approach, incrementing the number of trainable layers of the model but reducing subsequently Adam’s learning rate

Conclusions

Almost immediately we realized that transfer learning models outplayed our fully designed CNN when it came to performances, thus we focused our efforts on them.
Our best model was built upon InceptionResnetv2, achieving a score on the test set of 0.94666 based on classification accuracy.