AN2DL Homework 1 - Report

Gabriele Bozzetto - Lorenzo Castiglia - Davide Mantegazza

Leaderboard: bozzetto-castiglia-mantegazza

Introduction

We approached the task by building different convolutional neural networks models following the outlines provided during lectures. In this short document we are going to highlight and present the most notable models and experiments we made: starting from our first and raw model, describing our thought process to justify the changes and tweaks done in order to enhance it.

Models

First experiments

We began our work by creating a simple model without any optimization, in order to have an idea about the mean accuracy in a potential worst-case scenario. First, we generated the training set by the means of an **ImageDataGenerator** and its `flow_from_directory` function (in these first experiments we only normalized the training set images without using any data augmentation). The dataset was then fed to a simple convolutional neural network composed by five 3*3 convolutional layers (respectively by 16, 32, 64 and 128 filters), followed by a flattening layer and a single 512 neurons dense layer. All the internal layers used the ReLU activation function, while the output layer utilized the softmax activation, since the network is solving a multiclass classification problem. The model was then compiled with the categorical cross-entropy loss and the Adam optimizer, and finally fitted on 100 epochs, each of them split in batches of 8 images. Without any further optimization, this returned a 13% mean accuracy; this result was expected, since our goal was to find a simple starting model that overfitted the training data.

In order to improve the results and deal with overfitting we decided to use 20% of the original dataset as a validation set, and we employed some regularization techniques like Early Stopping and Dropout, which brought the accuracy up to 24%.

Data Augmentation

To further improve the mean accuracy on the test set, and to better tackle overfitting we started to work with data augmentation, which we performed by enriching the **ImageDataGenerator** with the usual random transformations, such as rotation, shift and flip. After some trials and errors, we found that having three hidden layers in the fully connected part of the network (respectively of 512, 256 and 128 neurons), and using L2 norm regularization on each of them, would boost the mean accuracy of our model to around 66%.

Transfer Learning and Fine Tuning

Since we felt satisfied with the results obtained by the aforementioned model, we decided to move on and started doing experiments with the Transfer Learning and Fine Tuning techniques, in which we didn't change the structure of the fully connected part of the network, keeping as a reference our best model with data augmentation, except for some minor tweaks.
We extensively experimented with the VGG16 feature extractor and subsequently VGG19, InceptionV3 and Xception as well.
We observed that using the globalAvgPooling layer instead of Flatten provided slightly better results in terms of accuracy.
Considering the nature of the Fine Tuning technique, we expected to obtain better results by decreasing the learning rate and indeed our best results were obtained with a value of 1e-5, significantly lower than the default value (1e-3) of the Adam optimizer, which we were previously using.
We then tried to train the models by keeping the feature extractor's layers frozen for a few initial epochs and then allowing fine tuning for the rest of the epochs, however this only led to marginal improvements.
During the first phase of the challenge the VGG16 model was our best submission accuracy-wise, exceeding 92%, while in the second phase it scored slightly less, about 90%.
On the other hand the Xception model saw an improvement of around 1% in the second phase, reaching 91%, our best result.

Notebooks

Together with this report, we are also including the most significant notebooks that we used for the duration of the competition:

- Model0: this was the first tested model
- Es-Rescaling-Dropout: this notebook returned the best model without data augmentation
- 6CL-3HL-l2Norm: this notebook returned the best model with data augmentation and without transfer learning and/or fine tuning
- VGG16-3HL-GlobAvPool-unfreeze: our best performing model using VGG16 and the “freeze-unfreeze” method
- Xception: the best model overall

Inside the notebooks there are plots of the models' structures and the accuracy on the training set and the validation set over the epochs.

Notes

- Notebooks have been executed via Kaggle, therefore it might be necessary to modify the cells in order to correctly load the dataset

- It is important to note that we set the seed to make the experiments reproducible but, since the evaluation with a GPU introduces a stochastic element, the experiments may not be repeatable with perfect accuracy