Artificial Neural Network and Deep Learning

Homework 1 (A.A. 2020/2021) – Image Classification

Diego Savoia, Matr. 944508 (10535515) – Francesco Emanuele Stradi, Matr. 944616 (10538326)

For this Image Classification problem, we decided to set up the model seen during the practical session of the course; this includes a batch size equal to 8, the image rescaling to 256x256, the Adam optimizer with a learning rate equal to 1e-4, the training and validation sets setup and the network architecture, composed by 5 blocks of convolutional layer, activation function (ReLU) and MaxPooling, followed by the fully connected part. On the opposite, we decided not to include the data augmentation part, considering the number of images available → *Test_Accuracy ≈ 58%*.

We tried to improve the accuracy increasing the model depth, obtaining the best performance with a depth equal to 7 → *Test_Accuracy ≈ 63% (first model in the notebook)*.

Working on the number of filters, the best results have been obtained reducing the depth of the model and the initial number of filters to 5, and at the same time incrementing more (x4) the latter after each block → *Test_Accuracy ≈ 66% (second model in the notebook)*.

To further improve the performance, we decided to try a **Transfer Learning** approach starting with the standard **VGG-16** architecture with fine-tuning and using our custom fully connected part → *Test_Accuracy ≈ 70%*.

A batch size hyperparameter tuning showed worse performance while increasing it, and better performance decreasing it: the best results were observed using a batch size equal to 2 → *Test_Accuracy of the first model in the notebook ≈ 66%;* → *Test_Accuracy of the second model in the notebook ≈ 70%*.

Inserting **data augmentation** in the training set, the results were very similar, so we decided to keep it anyway.

We introduced other architectures, always using a Transfer Learning approach: we tried networks like DenseNet, EfficientNet, Xception, **InceptionV3** and **ResNet**. Only the last two showed good results, so we decided to work on them. Of course, the proper pre-processing on the test images has been done where necessary.

We tried different freeze levels for the fine tuning of these three Transfer Learning models, obtaining worse results when decreasing them down to 5, and observing the best performance when setting them to 15 for VGG-16 and to 13 for InceptionV3 and ResNet → *Test_Accuracy of InceptionV3 ≈ 87%;* → *Validation_Accuracy of ResNet ≈ 83%*.

However, VGG-16 was showing a highly stochastic behaviour, remaining stuck on low accuracy values during training (in completely unpredictable situations): for this reason, we decided to change the optimizer only for this network, using **Stochastic Gradient Descent** with a learning rate to 1e-3. This is because SGD works better when dealing with a very high number of parameters like in the VGG-16 architecture. Furthermore, to keep the architecture more similar to the original VGG-16 one, we added two more layers in the FC part of our model → *Test_Accuracy of VGG-16 ≈ 81%*.

The same SGD optimizer with different learning rate has been tried also to the other models, without observing any improvement.

Further tunings were made using different dataset splits between training set and validation set, but the best results were observed keeping 20% of the samples for validation.

We tried some modifications in the fully connected parts of our models, introducing Dropout and/or Regularization layers, but without significatively improvements. In general, to contrast overfitting, we relied on **Early Stopping** (using a patience equal to 5) and we saved the best trained model through callbacks.

Changing the input image size to **512x512**, we noticed some important improvements especially in the first model in the notebook → *Validation_Accuracy ≈ 76%* and in the fourth model (InceptionV3) → *Validation_Accuracy ≈ 92%;* → *Test_Accuracy ≈ 91%*. The fifth model (ResNet) also showed an improvement → *Validation_Accuracy ≈ 86%*; → *Test_Accuracy ≈ 80%*. Instead, the first architecture and the third one (VGG-16) did not change much their results.

We also tried to push the input image size up to **768x768**, obtaining a slight improvement in the first model → *Validation_Accuracy ≈ 77%* and in the fourth one (InceptionV3) → *Validation_Accuracy ≈ 93%*. However, we decided not to continue this path because the accuracy on the test set started to decrease → *Test_Accuracy ≈ 88%* and the computational effort was becoming excessive.

Having all these different Transfer Learning models, we tried an **ensemble method** to combine the results of the single models to get the mode of the class predictions for each test sample. The performance was decent, but still lower than InceptionV3, which resulted as our best model → *Test_Accuracy of the ensemble method ≈ 87%*.