Machine Learning
Università della Svizzera italiana

# Assignment 2

De Castelli Fabrizio

June 5, 2023

## 1 Image Classification with Fully Connected Feed Forward Neural Networks (FFNN)

6. As we can see from figure 2, the performance of this model is poor. This is related to the fact that we are not extracting good features from the images, and we are just classifying images. However, the classifier is slightly better than a trivial (*"stupid"*) classifier, because the score is better than 14%, which is the average classification error in the case of uniform distribution. Here is the plot:

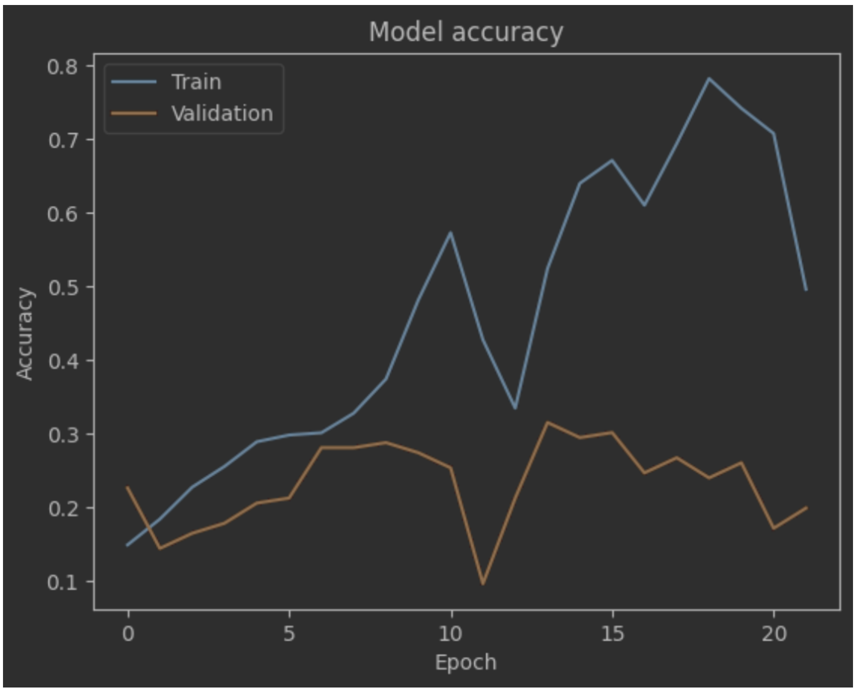


Figure 1: Plot 1 FFNN

It might not be very clear, but from the plot we can deduce that the training has been done with the help of Early Stopping, that restore the best weights configuration, found at epoch number 13. It is deducible because the validation error is at its peak, and then it starts decreasing for 7 epochs.
The test accuracy for this model is around 40%, after many runs on different dataset configurations. Even if it is not very coherent with respect to the plot, the saved model has some parameters configuration that ensures this performance, shown in figure 2.


Figure 2: Test accuracy T1

The accuracy of this model on new unseen data should be between 28% and 30%. Anyway, I got an intuition of this value by running 5-fold cross validation on permutation of the dataset (not available in the solution).

7. The results obtained from this task were very poor, even if the model was performing quite well for T1. I chose histograms as feature extraction, but they were not effective (the plot is very similar to figure 1). As before the Early Stopping callback were triggered at epoch number 11. The main difference between the previous plots is that in this last one, the model is not overfitting.

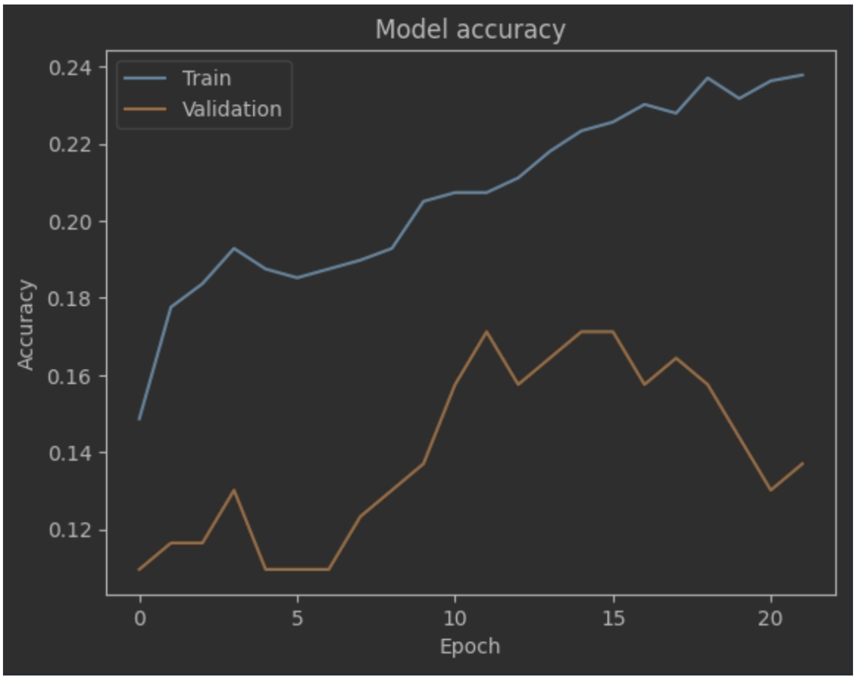


Figure 3: Plot T1 Bonus

The accuracy on test set was:

```
Accuracy: 17.2840%
```

Figure 4: Test accuracy T1 Bonus

## 2 Image Classification with Convolutional Neural Networks (CNN)

4. The performance of this model is better than the best model found in task 1. We can see it from the plot in figure 5:

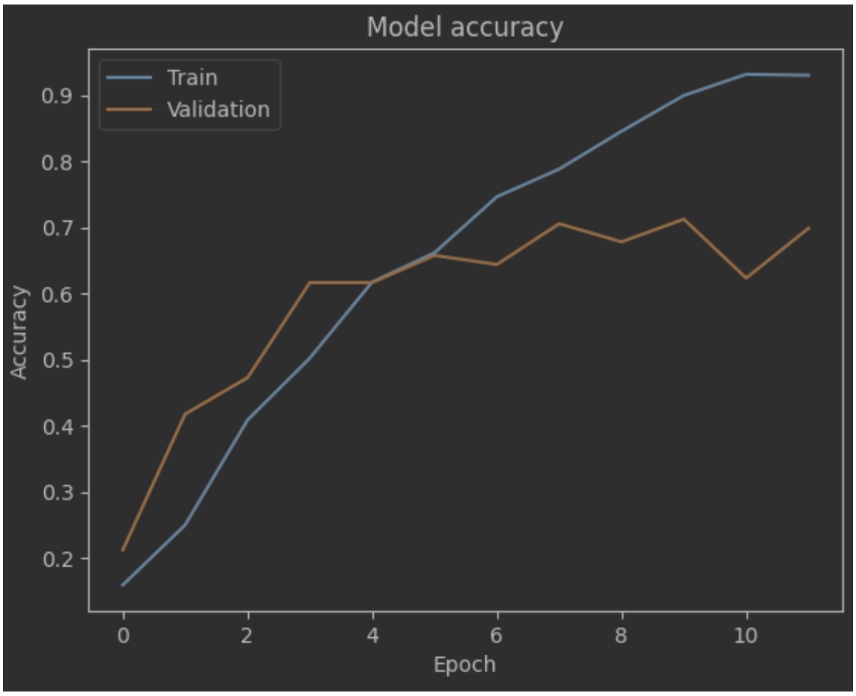


Figure 5: Plot T2.1

We can see that the validation accuracy and the training accuracy curves are more similar than the ones found in T1. This is a sign that the model can correctly extract the right features from the images and classify them better, as overall.

```
Test accuracy without image augmentation: 69.1358%
```

Figure 6: Test accuracy T2.1

5. To compare better the models, I ran a **Paired T-Test** and the outcome was that, very likely, the second model was better than the best found in the first task, as the T value was around 9.

```
T = 9.1289: T2 model is better than T1
```

Figure 7: T-Test T2.1

6. Applying image manipulation techniques to shift, shear (and other...) images was a good way to improve the model performance. From the following plot it might not be clear that the model perform very better than the previous one. Surely, image manipulation is very powerful: for instance, if we translate an image of one pixel, the network will not understand that it is the same image but translated, as all values in pixels will differ from each other.

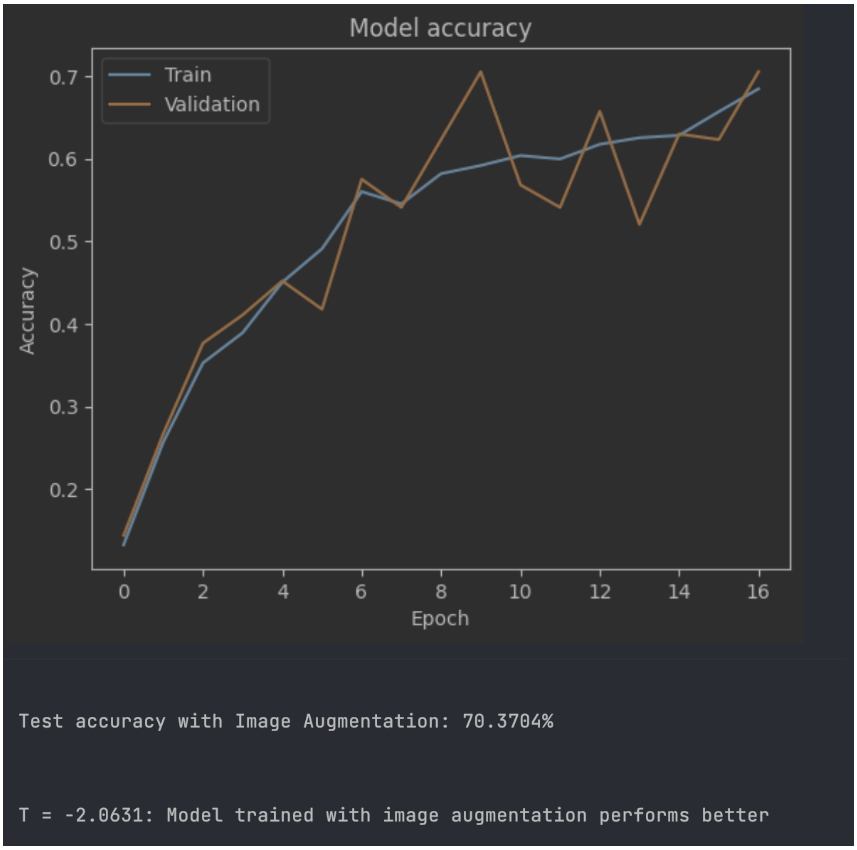


Figure 8: Accuracy T2.2

As we can see from the t-test, the model after image augmentation is slightly better than without.

7. In this task I performed grid search to fine tune the following hyperparameters:
   - Learning rate;
   - Pooling size;
   - Number of neurons in the first Dense layer;
   - Number of neurons in the second Dense layer.

As a result, the optimal configuration was:

- Learning rate = 0.001;
- Pooling size = (4, 4)
- Layer 1 = 64 neurons;
- Layer 2 = 64 neurons.

```
Best Parameters: {'pool_size': (4, 4), 'learning_rate': 0.001,
'neurons_dense1': 64}
Best Score: 78.7671%
```

Figure 9: Grid Search

The algorithm was implemented with for loops as **Tensorflow** was not supporting rthe utility *GridSearchCV*. The code available in the submission is not provided with for loops as I first optimized the neurons and then the other parameters (I started from 128, 64 neurons to arrive to 64, 64).
The T-Test stated that the improved model might be better on overall than the not fine-tuned model ($T = 2\%$).

```
T2 Bonus accuracy: 88.2716%
T2 accuracy: 83.3333%

T = 2.0094: T2 bonus model performs better
```

Figure 10: T-Test T2 Bonus