Master 2 AIC : Image Mining
Lab 2 : Deep learning for image classification

Corentin Leloup & Wafa Bouzouita

Abstract

The aim of this session is to use deep convolutional neural networks to classify images.

1 Data

The data we are trying to classify are images issued from the cifar 10 database. Their quality is low, the definition is 30x30 pixels. The images can belong to 10 different classes, which are visible on Figure 1.

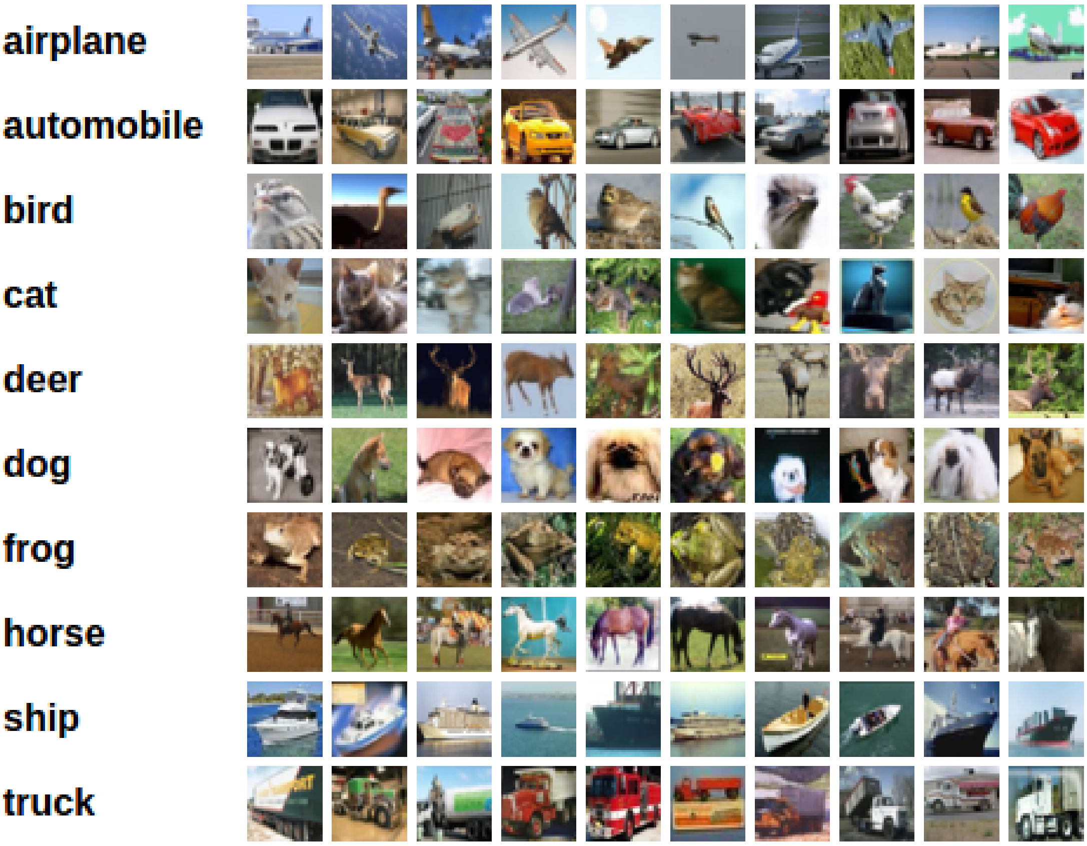

Figure 1: cifar 10 database

50 000 examples are available in this set and the 10 classes are well balanced, all of them contain the same amount of examples.

2 First model

We are given a baseline model composed of one convolutional layer with masks of size 3x3 and 18 output channels followed by a pooling layer where we take the max over 2x2 squares. We then have 2 dense layers with ReLu between each, to finally obtain 10 weights.

We train this model over 10 epochs with a training set of size 20 000, the results are visible on Figure 2.

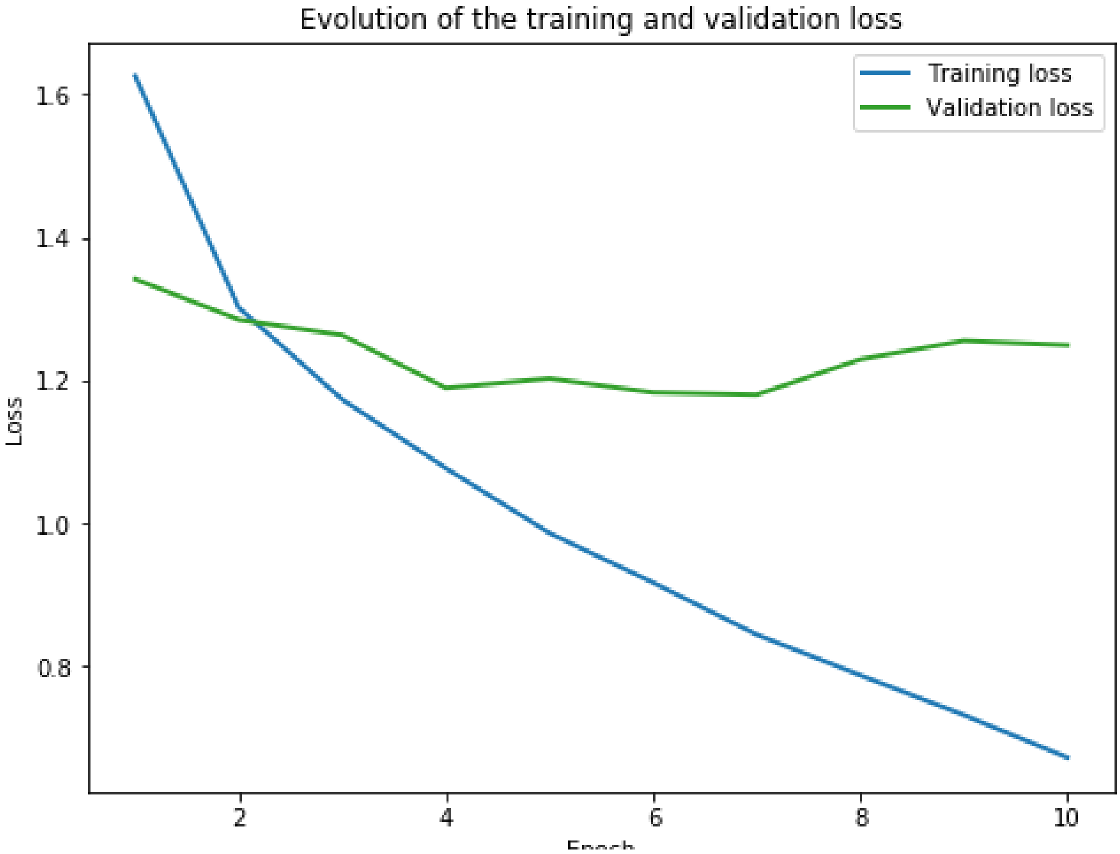

Figure 2: Training of the baseline model

3 Results

With this first simple model, we obtain a loss of 1.18 and an accuracy of 60%. Not all classes deliver the same performance, some are harder to classify than others, in particular animals that are often mistaken with one another. On the other hand, vehicles are easier to distinguish, cars in particular. The performance of each class is visible on Figure 3.

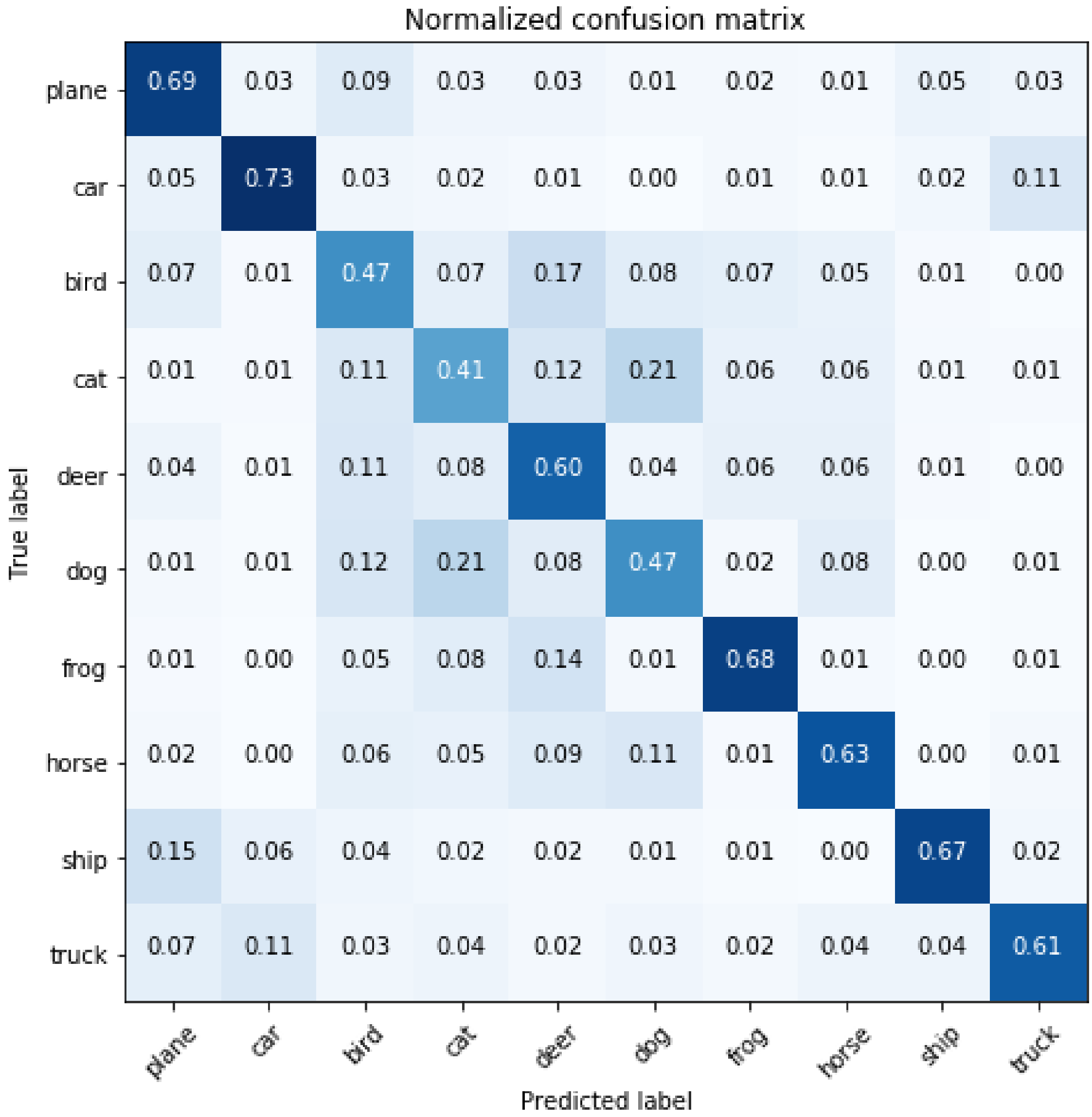

Figure 3: Confusion matrix

4 Improvements

To improve the performance, we first increased the size of the training set to get 40 000 training examples. It increased the accuracy to 65%.

Then we tried to modify the architecture of the network, we increased the size of the mask of the first convolutional layer to 5x5. Then we added a second convolutional layer after the first one with masks of size 3x3 and with only 10 output channels, followed by a max pooling layer similar to the first one. As the complexity of the network increased, we needed to add more training epochs. The results of the training are visible on figure 4.

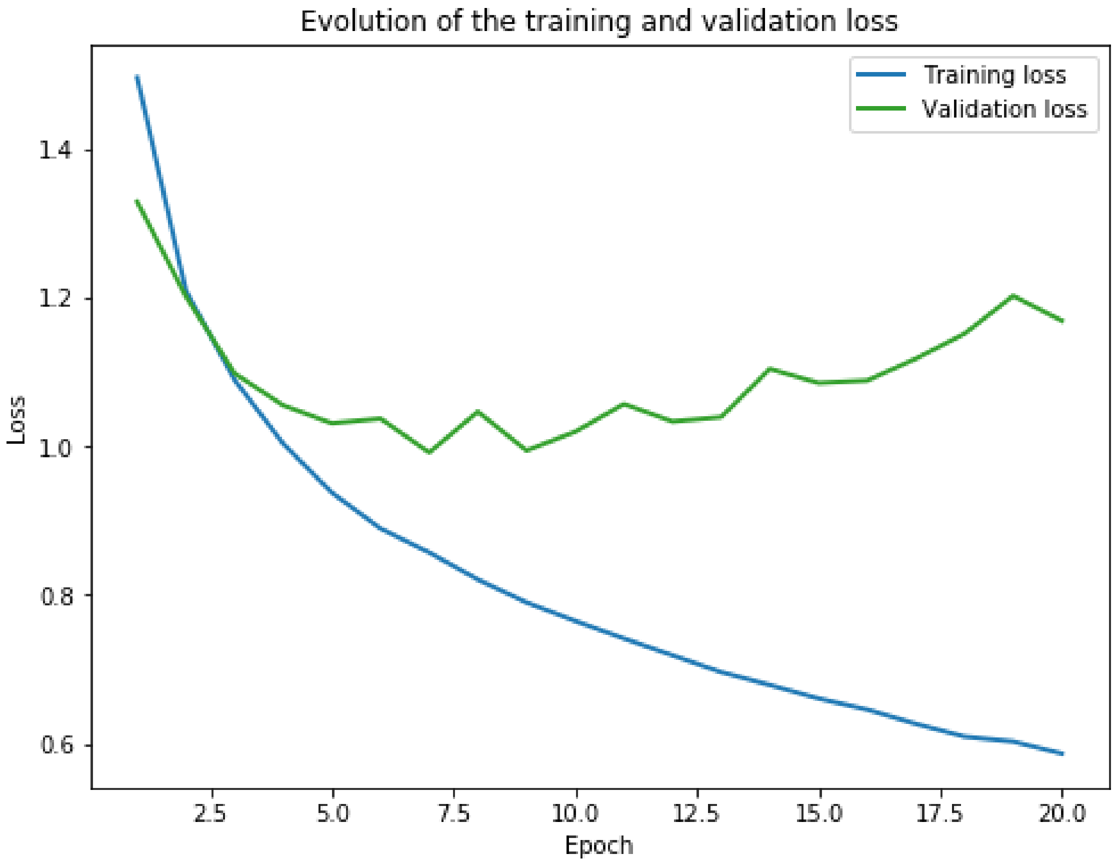

Figure 4: Training losses of the final model

With this final model, we obtained an accuracy of 67%. We were not able to obtain significantly better results, this is probably due to the low quality of the images.

Changing the learning rate was no use since we used the Adam optimizer.