Assignment 3: Bird image classification competition

Tamim EL AHMAD
ENS Paris-Saclay
elahmad.tamim@gmail.com

Abstract

This short report is about my approach and use of Deep Learning to classify the Caltech-UCSD Birds-200-2011 bird dataset.

1. Introduction

First, the Caltech-USCD Birds-200-2011 bird dataset is a dataset containing pictures of 20 different species of birds, each picture containing one bird. There is a training set which contains 1082 pictures, a validation set containing 103 pictures and a test set of 517 pictures, which is the final set on which predictions are performed to produce a test accuracy score and compete on Kaggle.

2. Data preprocessing

Before getting into the CNN model, a preprocessing on data should be performed. Training and test sets are not treated in the same way. Indeed, in order to have a robust and efficient model, and avoid over-fitting, it is useful to do some random rotations, cropping, flipping, change the brightness, contrast and saturation of the training images.

Then, it is necessary to resize all images to the same dimensions, transform it to tensors and normalize its pixel values before putting it in the CNN, to have some homogeneous data semantically. Same kind of transformations are done on the validation set except the random resizing and cropping of the images. Finally, for the test set, I only resize it, crop the center of the images to have 224*224 images, transform it to tensor and normalize its pixel values.

3. Model

As time and number of submissions for this assignment were limited, as well as my computing capabilities (I use a GPU on Google Colab), I used pretrained models. I first used a pretrained VGG16 model, in which I froze all the layers and only replaced the last one by a linear one with an output of size 256, a ReLU activation, a 0.2-rate dropout layer and a final linear layer with an output of size 20 (as the number of classes) and a logSoftmax activation.

Then, in my latest submission, I changed this model a little and stacked in parallel an other pretrained model: a ResNet152 one. First, in the VGG16 part, I unfroze the last 2 layers to let it train on the training set and I only added a liner layer with an output of size 2048 and a ReLU activation and 0.2-rate dropout layer. For the ResNet152 part, I froze all layers except the last 3, removed the Softmax layer and added a ReLU one and an Average Pool one of dimension 2. Finally, the output of both models in parallel are 2048 each, so at the end of the network I put a linear layer with an input size of 4096 and an output size of obviously 20, the number of classes.

4. Results

First, with the model containing only a VGG16 pretrained model, my best test accuracy score was 67%, and with a batch size of 64, a learning rate of 0.001 and 15 epochs, I reached around 80% of accuracy on the validation set.

Then, with the second stacked model with ResNet152, I reached a 78% test accuracy score, with a batch size of 32, a learning rate of 0.001 still and 20 epochs. The network performs regularly around 85% of accuracy on validation set and a loss of 0.1 approximately for each training batch.

Reducing the learning rate makes it too slow to train and not more performing, as well as increasing the number of epochs and reducing batch size. With a learning rate of 0.0001, the test accuracy score was 72%.

5. Conclusion

The use of pretrained model like VGG16 or ResNet152 is a good thing as these models are good performing on images and modeling your own network is very hard and long. But to get better results, I should have performed more preprocessing on data and use other networks to point out images' features and especially birds' features.

Finally, I should have test more times my network to fine-tune it, find the optimal hyper-parameters and find a better architecture.