Object Recognition and Computer Vision : Assignment n°3

Edimah SONGO
MASH
Paris Dauphine — ENS Ulm
edimah.songo@dauphine.eu

Abstract

The goal of this assignment is to classify several species of birds using the Caltech-UCSD Birds-200-2011dataset, and to maximise the accuracy of the model. A baseline model was provided, and I approached this exercise by first working on the data itself, then by fine-tuning the base model, and finally by tuning hyperparameters. Here I will explain my approach and conclude.

1. Original framework

Caltech-UCSD Birds-200-2011 is a dataset containing 200 species of birds and 11,788 images. However the dataset provided for the assignment seems to be a subset of the original, containing 20 categories of birds, 1082 pictures in the training set, 517 in the test set, and 103 in the validation set.
We will build a model architecture taking the baseline model as a departure point.

2. Data processing

The data in the first 'data.py' file had already been normalised and rised to 64×64 pictures. Resizing them to 300×300 yielded worse results, so I padded to keep them square and to make sure birds were preserved, but it didn't have much of an impact. So I kept the original resize (64×64) and chose instead to apply various transformations.

2.1. Data Augmentation

Most birds are placed at the center of the image but not all of them, this could bring in biais. In order to train the model on all possible positions, I have modified the training set by randomly cropping, horizontally flipping, and rotating some images by 25°. I have also added some gaussian blur and colour jitters. I preserved the validation set by only applying minor transformations : resizing and normalisation.

3. Model Architecture

Now that we have processed data and applied various transformation, we now fine-tune our baseline model to truly optimise our accuracy. Firstly, to avoid overfitting due to the small size of the training data sample. In order to remedy this, I turned to pre-training.

3.1. Transfer Learning : Using pre-trained models

Here we use the weight initialisation of ResNet152, a model pre-trained on ImageNet, and I applied a sigmoid to the output of the network.

3.2. Training and Validation

From this new base model I froze the layers preceding the fully connected one in order to only train the latter for 10 epochs. Afterwards, I unfroze the rest and trained the whole model for another 10 epochs.
As I kept running into RAM limitation issues, I reduced the default batch size to 32.

3.3. Hyperparameter tuning

In order to increase the performance, I turned to hyperparameter tuning for the learning rate and the number of epochs.
I used a learning rate scheduler to find the optimal learning rate for the model by considering its architecture and parameters. To that end, I picked StepLR, based on step sizes for each iteration of the model parameters. This yielded a maximum test accuracy of 69%
At last, I raised the number of epochs from 10 to 20, which didn't improve the performance on the test set. Setting it to 15 seemed to slightly improve it.

4. Conclusion

After a data augmentation stage, I used and fine-tuned a pretrained model, then used hyperparameter tuning methods to improve performance on the testing set from 30% initially, to 69%, and finally to 79%.