HW3: Kaggle comptetition

Tamames Louis
louis.tamames@ens-paris-saclay.fr

Abstract

The assignment is to participate in a Kaggle competition with the rest of the class. We were given a subset of the Caltech-UCSD Birds-200-2011 bird dataset and we had to implement a Neural Network with the best performances.

1. Introduction

Our subset contains 20 species of birds out of the 200 of the original dataset. Our objective was to produce a model that gives the highest possible accuracy on a test dataset containing the same categories. To do so, we were given a train set of 1082 images and a validation set of 103.

2. Development

First, I tried a hand made model based on the one given in the instruction, but the results were very bad. Indeed, I was not able to go beyond 50 $\%$ accuracy on the validation set. At this point, I understood that with our small dataset, a handmade model couldn't be very efficient, so we need to import pretrained model with more complex architecture, and also use data augmentation techniques to increase the diversity of our train set.

In this purpose, I tried differents data augmentation methods, and I kept a $RandomRotation$, $RandomResizedCrop$ and $RandomHorizontalFlip$ which made our models much more efficient (it made my accuracy jump from 60$\%$ to 80$\%$).

Then, I learned how to do transfer learning with the help of pytorch tutorial. First I downloaded pretrained model and then I changed the last fully connected layer to output one of our 20 classes of birds. It permitted me to train different models to see which one had the best results. My first trial was on resnet18, it was quite good but I found more complexe model with higher accuracy:

Transfer learning			
	Train	Validation	Kaggle
Resnet18	91	88	70
VGG16	84	89	74
densenet161	93	89	75
Resnet152	90	88	76

In the four models above, I retrained all the parameters with our dataset. I realized that the most efficient models were Resnet152 and densenet161, so I focused my attention in those ones, and tried to improve their results.

3. Improvement

In order to improve the result of those models, I resampled the train and Validation set, which had, as effect, to increase the accuracy of around 2 $\%$.

I also tried to fix two third of the parameters in the Resnet152, and only trained the last third of them. The training was faster and gave me 3$\%$ of accuracy more in kaggle. Doing this permit the model to less overfit our train and validation set which resulted in generalizing better on the kaggle test set.

Resnet152 with			
	Train	Validation	Kaggle
Resample train/val	93	92	77.4
+ Fixing parameters	90	92	80.6
densenet161 with			
	Train	Validation	Kaggle
Resample train/val	94	95	76.1

During my work I searched for optimal parameters, testing different values for the batch size, the learning rate and momentum. As optimal parameters, I kept a batch of 16 images, a decreasing learning rate and a momentum of 0.8 for the first epochs, and then 0.95.

4. Possible improvement

I thought about two other methods to improve the performance of my models, but I didn't manage to implement them well because of a lack of knowledge and time.

The first was to assemble methods. To do so, I wanted to implement Bagging or Boosting function to give the best results out of different models, either by taking the mean of the outputs or by a voting system.

The secund was to, first, find a bounding box around the bird, and then, make the box go through the Neural Network, to eliminate the possible interferances created by the background.