MVA - Object Recognition and Computer Vision - HW3

Jean-Baptiste Sevestre
ENS Paris-Saclay
61 Avenue du Président Wilson, 94230 Cachan

Abstract

In the low-data framework, many strategies can be adopted to face overfitting. In this article, we will see how we improve our accurary score on test set, and all the experiments that we've done for this.

At first, we've implemented different models with different parameters to choose the best ones (at least the best trade-off). So we will keep these parameters set for the rest of our experiments (you can find them in the code).

1. Transfert Learning

Transfer Learning is very convenient in a setting where there is not a lot of data. In a first time, we've used Resnet152 pretrained on ImageNet. We have replaced the last layer of Resnet152 with a dense layer of size 20, adapted to our classification problem. Then, for the training step, we haven't trained the entire network, but only its last dense layer. That give us an accuracy of 0.70322. Unfortunately, ImageNet images do not look like birds, so Transfer Learning is here limited. To overcome this, we've decided in a second time to train, in addition to the last layer, the second to last layer of the model. Logically, the scores were then improved to 0.81290 (cf Table 1.). In a third time, We've decided to train also the third to last layer of the model. However the score did not increase this time (0.74838). Indeed, the number of parameters is then too great for the number of data that one has.

2. Data Augmentation

To avoid overfitting, we can try to increase the number of data. The number of images for learning (train-images) is only 1082. So we've decided to use a Data Augmentation technique to increase the number of data. A tool we've used is called Augmentor. It allows to distort the initial images according to certain probabilities. Another ambitious Data Augmentation technique is generating images through a Generative Adversarial Network (GAN) to generate new data. So We've implemented a DCGAN, but the results were not conclusive. Finally, there is something surprising ; the performances aren't better for a train set of 2082 or 21000 images than for a train set of 1082. The images of "deformed" birds must therefore induce the model in error, hence the lowest score. Despite that, there's something interesting to note : if we pretrained and trained recursively a Resnet152 on the 2082 images, that give us the third time an accuracy (test set) of 0.81290, as much as Resnet152 with 1082 images.

Model	Accuracy (test set)
3-best Resnet152 vote	0.83225
3-best Resnet152 vote (weighted)	0.82580
best Resnet152	0.81290
3-bagging-Resnet152 vote	0.79354
1-bagging-Resnet152 vote	0.78709

Table 1. optimizer=SGD, lr=0.001, batchsize=8, epochs=10

3. Ensemble Methods

Ensemble Methods are powerful in machine learning. They improve accuracy of predictions using many models. First of all, we've assembled the best models we've trained, the two Resnet152 each train initially with 1082 and 2082 images with accuracy of 0.81290, and an other one, based on Resnet152 train with 1082 images too. For each prediction, the algorithm will ask all the models their prediction, and each model has one vote. So we use majority vote to decide which label it is. Combining the three best of our previous algorithms give us an accuracy (on the test set) of 0.83225 (cf Table 1.). Output of the model's last layer give us a probability of "believe" that is the real class ; it is information that we can use. So in a second time we've implemented the same algorithm, but here, in the final voting process, each vote is weighted by the probability. Combining the best three of our previous algorithms give us an accuracy now (on the test set) of 0.82580, which is not an improvement (cf Table 1.). Something interesting is that changing the vote mode or combining more models (the four, five,... seven best) does not improve the accuracy than the previous increase We note that the two best algorithms were trained on different training set : one has the original training set (1082 images), and the other has distorted images (2082 images). And the other best models are trained on this sets. So we've supposed that the improvement of our accuracy score comes from the fact that models were trained on different sets. After this finding, we've tried Bagging method to get models built on different training set. The results show that add others models to the final vote does not improve accuracy, even if the models are built on a little different sets thanks to bagging method (cf Table 1.).