Precedure for Kaggle Competition in Object Recognition and Computer Vision

Franck Laborde
f.laborde6540@gmail.com

Abstract

This paper explains the approach and the results made to the kaggle competition that aims at classifying the images of the data ImageNet Sketch.

1. Introduction

The ImageNet-Sketch dataset is a derived version of the original ImageNet dataset, consisting of sketch-style drawings corresponding to 1,000 categories. For this competition, the dataset includes only 500 categories.The availability of pre-written code for training, testing, and generating files significantly facilitated the workflow.

2. Models

2.1. Baseline CNN

As a starting point, I explored the structure of the provided codebase to understand how new models could be integrated. I began by running the script with the baseline CNN already implemented. This model, being shallow and simplistic, achieved a test accuracy of only 4%, which was expected given its limited capacity.

2.2. Pretrained Models

2.2.1. Fully Trained Models

Recognizing the limitations of the baseline CNN, I moved to pretrained models. The first model I tested was VGG16, training it end-to-end. Initially, the results were unsatisfactory, with test accuracy around 30%. Upon investigation, I found that the default learning rate of 0.1 was too high, causing poor convergence. Reducing the learning rate to 0.01 yielded significantly better results, 75% accuracy when submitted to Kaggle.

Following this, I experimented with other popular architectures, such as AlexNet and ResNet. Using the same approach, I obtained Kaggle submission accuracies of 70% for AlexNet and 83% for ResNet.

2.2.2. Optimizer Selection and freezing layers

With ResNet identified as the best-performing model, I refined its optimization process. Initially, I tested the Adam optimizer, expecting improvements. Surprisingly, Adam produced significantly lower accuracy, achieving only 13% on Kaggle (This result is particularly low though). This aligns with known issues of Adam's performance on CNNs for certain tasks (this paper for example Improving Generalization Performance by Switching from Adam to SGD). I reverted to Stochastic Gradient Descent (SGD) with momentum, which consistently delivered better results.

To further fine-tune ResNet, I froze the early layers of the model and trained only the later layers. However, this approach led to a drop in accuracy (about 30% on Kaggle). This can be attributed to the significant differences between ImageNet-Sketch and ImageNet, where the early layers' features were insufficiently representative of the new dataset.

2.2.3. Momentum and Image Transformation

Encouraged by the results with ResNet, I explored additional architectures, including EfficientNet. This model initially achieved 84% accuracy on Kaggle so I used this model for the following tests. Noticing that I had used an SGD momentum of 0.5 thus far, I increased it to the commonly recommended value of 0.9. This adjustment yielded my highest accuracy in the competition: 88%.

I also experimented with data augmentations to further improve the model's robustness. I added transformations such as random rotations and Gaussian noise, but these changes did not improve accuracy. On the last epoch, the training loss oscillates but it did not improve the accuracy. Lastly, I tried a learning rate scheduler as the models seemed to oscillates on the last epochs but it did not improve the accuracy.

3. Conclusion

While the experiments achieved promising results, with EfficientNet reaching 88% accuracy, there remains room for further improvement. One unexplored way is the use of Vision Transformers (ViT), which have shown state-of-the-art performance in various vision tasks.