HW1: Image Classification

XIU-RONG HUANG
Institute of Information Management
National Yang Ming Chiao Tung University
Jejin0327@gmail.com

Github : https://github.com/jiejin0327/bird-species-classification

1. Introduction

In this assignment, I implement the techniques of image classification trained in deep learning network. To classify the bird species classification, I use PyTorch framework for building my own ResNet model on this task. For the high accuracy score, I utilize the data augmentation, proper hyperparameter setting, data preprocess (resize in proper size) and learning rate schedule.

2. Implementation Procedure

2.1 Data pre-process

In order to select the proper image size, I caculated the average width and height of images on training data. As a result, the average size of training data is 375 and 500 pixels in height and width respectively. Then, I refer to the this github [1] on padding strategy code. The author utilized the max padding method to modify the image size into 500x500 pixels. After that, the images will be cropped into the average size of the majority data 350x350 pixels on data augmentation.

2.2 data augmentation

Deep learning model usually requires a lot of data for training. In general, the more data numbers, the higher performance model is. In addition, more transformation on images before feeding into model can reduce the overfitting effects. Therefore, I implemented the data augmentation technique. I cropped image into the 350x350 pixels. Guessing the bird will appear in any position, I utilized the random-crop, rotation and horizontal transformation in random order. Besides, I also normalize images by mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225] because I will use the pretrained model ResNet50 and it is recommended for using its own pretrained mean and std.

2.3 Model architecture

For training a bird classification model, I chose the ResNet50 model. Since the pytorch provides pre-trained model which is already trained on ImageNet dataset, I utilized the pre-trained model and fine-tune the weights. Fitting the output class number of fully connected layers, I set the argument to 200 given the class numbers of bird classification.

2.4 hyperparameter setting

Getting the better performance of model, I set the hyperparameter following:

- batch size : 24
- epoch numbers:30
- criterion : nn.CrossEntropyLoss()
- optimizer : Adam
- learning rate : initiate 1e-4 with learning rate schedule

This task is multi-class classifier, the cross-entropy is proper. The choice of optimizer is Adam as optimizer with initial learning rates 1e-4. Then, I set the learning rate schedule with gamma factor to adjust the learning rate during the training model steps. Therefore, the learning rates will decay by gamma in each epoch.

3. Summary

Through the analysis of experiment results, I realize that the pre-trained model provides higher performance even feeding to another data. In this assignment, I implemented the pre-trained model ResNet50 and used the data augmentation to increase the complexity of training data. In addition, I referred to the author's padding strategy on preprocessing the data into a proper size to higher the model performance. Adding this padding step, my scores increase from 64 to 67. After a few days on tuning model, I get the 0.676888 scores on codalab in the end.

References

[1] https://github.com/slipnitskaya/caltech-birds-advanced-classification