

11785 Homework 2 Part 2, README

1. Code overview

This code is an implementation of image classification and verification using Residual Network (ResNet). The “main_classification.py” and “classification.sh” are used for classification job and “main_verification.py” and “verification.sh” are used for image verification. The “models” folder contains ResNet architecture models. The two main python files have a lot in common but differ in the data loader and training.

2. Classification

2.1 Data loader

In the data loader part, I chose not to use ImageFolder method since I met with hidden files at the beginning. Using the customized data loader(torch.utils.data.dataset/dataloader), I created loader for train/validation and test data.

2.2 Model selection

I used the architecture of ResNet but did some minor changes in the fully connected layer since our output label is different from the original ResNet network. Also, I implemented ResNet18, 34, 50, 101 and 152, but eventually I chose the ResNet 50 for the classification and verification due to the GPU memory.

2.3 Initialization, Optimization and Parameter tuning

For initialization, I applied He initialization according to the ResNet paper. I also used Adam optimizer with learning rate decay. In the training process I began with learning rate 0.01, and decay with gamma 0.1 every 7 epochs. For the mini-batch size, I tried from 32 to 256 and found 256 has the best performance.

2.4 Training, validating and testing

The main process of each epoch is in the train_epoch, val_epoch and test_epoch function. For each epoch I will do the training, validation, testing and save models.

2.5 Result

After about 15 epochs I got nearly 99% training accuracy and 83% validation accuracy. The final Kaggle score is about 76%.

3. Verification

The verification process is similar to classification, although I had to rewrite my data loader and training_epoch function. I applied nn.cosinesimilarity here to compute the similarity score.