

IE 534 Homework 5 Report - Image Ranking

Hanwen Hu

October 22, 2018

This homework asks us to train an Image Ranking network with ResNet, with dataset tiny-imagenet-200.

As for the final result, the train accuracy has achieved 52.13%, and test accuracy has achieved 50.35%.

1 Hyperparameter Settings and Time Cost

For this network, I choose pretrained ResNet18 as initialization of the whole network. It runs 19 epochs in total, costing almost 12 hours, and SGD beginning from $1e-3$ with momentum 0.9 is applied during training.

Input images are shuffled, then divided into mini batches, with batch size as 64. This is the empirical largest value for batch size, otherwise overflow errors may occur on BlueWaters.

2 Dataset Preprocess

The dataset to use is stipulated that each epoch should be trained with a brand new triplet, each training image being the query image for exactly one time. To realize this, I sample the paths of each triplet and write 30 txt files for a maximum of 30 epochs, and read the paths at the beginning of each epoch to load all image triplets.

In addition, I notice the images are repeated loaded and going through the same transformation in the whole training process, which actually costs much time. Therefore, I transform all images to be (224×224) and save them in the 'RGB' mode.

Last but not least, to conveniently load the validation image set, I divide them into 200 folders according to the classes they belong (recorded in the annotation text file).

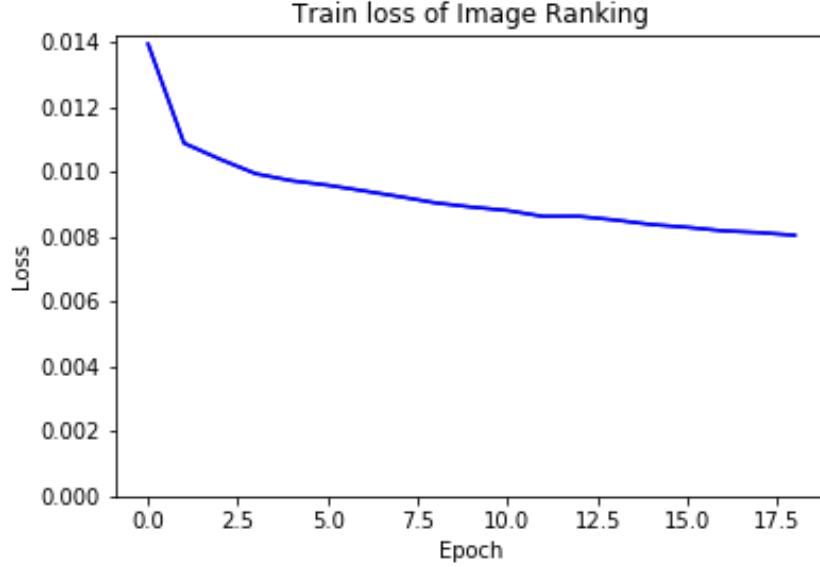


Figure 1: Train Loss graph of Image Ranking ResNet18

3 Loss Graph

The training loss graph is given in Figure 1. As is observed, the loss keeps decreasing during the training process.

4 Improvement

One improvement that's possible to make is to advance the sampling strategy. For now, triplet sampling is done by sampling the query image uniformly overall (or just take each train image one time), the positive image uniformly in class, and the negative uniformly out of class. While in the original paper, the authors first compute the relevance scores between each pair of images in every class, as well as the total relevance score of each train image; then sample the triplets in the following way:

- (i) Sample the query image with probability proportional to total relevance score;
- (ii) Sample the positive image with probability proportional to its relevance score between query image;
- (iii) Sample the negative image with same distribution as the positive image, but only accept it when its relevance score is lower enough than that of the positive image.