

CS 547 Homework 4 Report

Name: Jiashuo Tong

Date: 10/02/2019

Code Structure

The code is composed of 5 parts:

Part 1: This part defines data augmentation techniques that are imposed on the data. First, the `torchvision.transforms` module is used for data augmentation purpose. Images are flipped horizontally and vertically to generate a lot of new data. The data is also converted to tensor data type, and normalized before being fed to the network.

Part 2: This part defines the CNN Model. Convolution layers, Batch Normalization layers, Dropouts and Maximum Pooling layers are built in to the model. The order of the layers is same as the one proposed in Lecture 6 Note. This model structure proves to be effective as discussed in Results section. With the power of PyTorch, editing a network becomes much simpler than previous HWs.

Part 3: This part prepares for the following training process as it sends model to GPU, defines necessary hyperparameters and set the model to training mode.

Part 4: This part trains the model in multiple epochs. It performs forward and backward propagation, updates the optimizer and reports training accuracy of each epoch.

Part 5: This part is for extra credit, as it compares the accuracies of heuristic method and Monte Carlo simulation.

Results

The program reaches a training accuracy of~ 90% at the 14th epoch. The trained model achieves a test accuracy of around 85% for both heuristic and Monte Carlo method with 40 samples. For heuristic method, we need only one feeding process without dropout. For Monte Carlo, we want to dropout neurons as we do in the training process. We want to feed in each piece of data multiple times and average the raw scores for making predictions. It is worth pointing out that the Monte Carlo simulation with small sample size (<5) is not as good as the heuristic method in making predictions. Only when the sample size is increased to over 10 does Monte Carlo have a comparable accuracy with the heuristic method.