Assignment 5
Due May 24, 2020 at 11.59 PM via ilearn Assignment-5 Submission folder

Independent Reading.

1. ImageNet Classification with Deep Convolutional Neural Networks (papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf)
2. Generative Adversarial Networks (https://arxiv.org/pdf/1406.2661.pdf)

Note: Sample codes for this assignment are in PyTorch. You are free to convert this code in the framework of your choice in python.

Problem [10 pts] In this assignment, you will be training a Convolutional Neural Network (CNN) from scratch. Recall, in Assignment-4, you used pre-trained weights of ResNet50 for feature extraction and trained just a logistic regression model, which is essentially the final layer of a CNN used for classification. In this assignment, you will be using the CIFAR-10 dataset, which contains 32×32 images divided into 10 categories. The training set contains $50,000$ images and the test set contains $10,000$ images.

The starter code for this problem is `train.py`.

You need to fill in the following.

[5 pts] Fill in object class named `ConvNet` in `cnn_model.py`. You'll define the network layers in the method `__init__` and the forward pass in the method `forward`. We create an instance `net` of the class `inference` in `train.py` which takes a tensor with dimension $(B, 3, 32, 32)$ as an input, where B is the batch size. The output should be a matrix of dimension $(B, 10)$ representing the confidence (before applying the softmax function) for the 10 categories. Your CNN should have 4 convolutional layers with ReLU, Dropout, Batch Normalization and Max pooling operations in each (if required), followed by a linear layer and then a classifier layer at the end. You can use the function `_initialize_weight` in `cnn_model.py` to initialize weights of the convolutional layers. You are allowed to choose stride, weight decay, dropout rate, filter kernel size, number of output feature maps for each layer and other hyper-parameters. FYI: A 4 layer network (written properly) should be able to obtain at least an accuracy of 65% on the test set.

[0.5 pts] Fill in loss function with cross-entropy loss.

[1 pts] Fill in to obtain accuracy of current batch of data.

[1.5 pts] Fill in to obtain test accuracy over the entire test set and append it to the test_accuracy variable.

[2 pts] Plot training loss, train accuracy and test accuracy from the saved variables. Can you infer based on the plots, whether the model is overfitted, under-fitted or perfectly fitted ?

Submission Protocol. You should submit codes as well as explanations to each problem (if required). You should add comments to your codes to make them reader friendly. If you may require to call functions for a problem, you may do so, but include them in your submission. You MUST also include a report (in pdf) written electronically (using the likes of LaTeXor MS Word). It should contain explanations, images, etc (as required).

Each student must do the assignment independently, although you may discuss prior to that. While discussion is allowed, we will be particularly careful about any plagiarism, whether from each other or from other sources.