COMS W4995-2: Deep Learning October 16, 2017

Assignment 2. Due on October 30, 11:59pm

Instructor: *Iddo Drori*, Assignment CAs: *Ching-Hui Hsu, Surabhi Bhargava*

In this assignment, you are required to create a Convolutional Neural Network using TensorFlow. Your network will be trained on the CIFAR10 dataset and will learn to classify images into 10 categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.
More information about setting up and using TensorFlow can be found here: https://www.tensorflow.org/

1 Part 1 (60%)

Using TensorFlow, implement a Convolutional Neural Network to classify images in CIFAR10 provided in assignment1. Your model must have at least the following:

- At least two Convolutional Layers followed by normalization and pooling layers.
- Activation function ReLU.
- Optimizer: Gradient Descent
- At least one fully connected layer followed by softmax transformation.

2 Part 2 (40%)

There are several techniques to improve the performance and generalization of the Convolutional Neural Network. The objective in this problem is to try different methods to tweak the performance of your model and observe how the performance changes. Following are a few suggestions which you may try for the purpose of this problem.

- Modify architecture :
  - Change number of Convolution layers
  - Try a different kernel size
  - Change number of fully connected Layers
- Change the activation function
- Try a different optimizer
- Data Augmentation
- Add dropout

Your techniques need not be limited to just these. You must try at least two different methods and document any significant observations. To receive full credit, the performance should be better than Part1.

3 Extra Credit (10%)

The 3 teams reporting the maximum test accuracies will receive an extra credit of 10%.

Submission details:
Submit HW2-uni.zip file that contains the following files named in the specified way:

- Submit an iPython notebook ”q1-uni.ipynb” showing the training process, particularly training loss and training accuracy for Part1. Similarly, submit an iPython notebook ”q2-uni.ipynb” for Part 2.
- Save the predictions *y_pred* for both part 1 and part 2 to files ”ans1-uni.npy” and ”ans2-uni.npy”. The numpy array should contain the unnormalized scores from the network, so the shape should be (num_classes, test_size). For generating the y-pred npy files, you are required to follow the order of test data as mentioned in test_list.txt file present on courseworks. This will be the same order as produced by the helper function ”def get_files(folders)” in assignment1.
- Submit a report consisting of the following:
  - Name and Uni of your group members
  - Part 1
    - The architecture, training and test accuracies of the CNN.
  - Part 2
    - The architecture, training and test accuracies of the CNN.
    - Duration of training.
    - Size of training data (especially if you perform data augmentation.)
    - Documentation of all the steps you performed for Part 2 along with the observations made.

  For your best performing model (Highest Test Accuracy), include in your report the TensorBoard screen shots of the following:

  - Total Training loss
  - Training Accuracy
  - Histogram of weight gradient of the convolution layers.
  - Histogram of weight gradient of the last fully connected layer.

  Please make sure to label them correctly.

Notes:

- Please ensure to normalize the input data before feeding it into the neural network.
- The expected training time for Part 1 is approximately one hour with an expected training accuracy of 70% using a standard CPU.