Assignment-4: Convolutional Neural Network

Problem Statement:

A social media company wants to design a method which differentiates images uploaded by different users in the social media into one of several classes. The purpose is to identify and block malicious images uploaded by any user in the social media before it spreads. The different classes are: (a) malicious, (b) acceptable, and (c) completely safe. While collecting and labelling the dataset, the engineers want to try their method on a toy dataset (e.g., MNIST data) to analyse different practical issues. Develop a CNN and study the effect of various aspects of the network to help the team.

Implementation: [4+4=8]

- Implementation of a CNN (named, CNN-Vanilla).
- Implementation of a Resnet based CNN (CNN-Resnet). Note that the CNN-Vanilla must have the same architecture of CNN-Resnet, but with the residual connection removed.

**You may make use of the Pytorch library for implementation and to process and handle data.

** You may use a two level Resnet block with two fully-connected layers (each Resnet block has two CNN layers – refer to CNN slides). For the baseline network, use spatial size of CNN filters as 3X3. You may use the number of feature maps in each level that suit your system's computation power (e.g., see CNN tutorial) — but once selected, it must be kept the same for different analysis.

** For the baseline network, use ADAM optimizer, and mean and standard normalization of data (as done in CNN Tutorial).

**For training [CNN-Vanilla] and [CNN-Resnet], use a learning rate of 0.001, minibatch size greater than 256 and total no of epochs greater than 25.

**Performance Metric to be used for evaluating the models is Percentage Accuracy on Test-set.

Experiments: [4+3+3+3+2=15]

Use the standard train and test split of MNIST (refer Dataset details and CNN Tutorial).

1. **Experiment 1:** Report and compare the performances (percentage accuracy on the test set) of [CNN-Vanilla] and [CNN-Resnet]. Also, compare the number of parameters of the two networks. Find which one is better for the given problem.

2. **Experiment 2:** *Study the Effect of Data Normalization*. For the best network found in the earlier experiment, train the network with data normalization and *without* data normalization. Compare the performances of the two cases.

3. **Experiment 3:** *Study the Effect of Different Optimizers.* Report the effect of varying the optimizers for the best network and best choice of data normalization (from Experiments1-2).

Consider the following optimizers: (a) Mini-batch gradient descent with no momentum, (b) Mini-batch gradient descent with momentum 0.9, and (c) ADAM optimizer. For (b-c), use a mini-batch size greater than 256.
Optional: Consider Stochastic Gradient Descent (this may take much time to train).

4. **Experiment 4:** *Study the Effect of Network Depth:* For the Resnet network (CNN-Resnet) and best choice of normalization and Optimizer (from Experiments 2-3), change the depth of the network (i.e., from two level Resnet blocks with two fully-connected layers) as follows: (a) *Three* level Resnet block with two fully-connected layers; (b) Two level Resnet blocks with *four* fully-connected layers (Note that a & b has effectively an increase of two layers as one Resnet block has two Conv layers). Compare the change in the number of parameters and performance differences of the three networks. Next, comment on which one is better among (a-b): i.e, increasing the conv layers or increasing the fully-connected layers.

5. **Experiment 5:** *Study the Effect of Different Regularizers.* Report the effect of adding the regularizers for the best network, best choice of data normalization and best optimizer (from Experiments1-3). Consider the following regularizers: (a) Batch Normalization, (b) Dropout, (c) Both Batch Normalization and Dropout. Consider the performance on *testing datasets*.

Datasets:

The MNIST database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples. The classes belong to numbers 0-9. Refer to the CNN Tutorials to understand loading datasets.

Problem: Predict the class in which a particular image belongs.

Submission:

A .zip file containing the python source code and a PDF report file. The final name should follow the template: <Assign-No>_<Your Roll No>.zip. For example, if your roll no is 15CE30021, the filename for Assignment 4 will be: `Assign-4_15ce30021.zip`

1. A single python code (.py) containing the implementations of the models and experiments with comments at function level. The first two lines should contain your name and roll no. Also provide the python notebook for evaluation.

2. A report [PDF] containing **[2 points]**
   a. Experiment 1: A plot of training accuracy vs epochs for CNN-Vanilla and CNN-Resnet (similar to the plot in the CNN Tutorial). Also, mention the best choice among the two.
   b. Experiment 2: A plot of training accuracy vs epochs for the two cases of data normalisation. Also, mention the best choice among the two.
   c. Experiment 3: A plot of training accuracy vs epochs for the three cases of optimizers. Also, mention the best choice among the three.
   d. Experiment 4: A plot of training accuracy vs epochs for the two cases of network depth. Also, the best choice among the three. Also, comment on whether increasing the conv layers or FCL is better.

e. Experiment 5: A plot of **testing** accuracy vs epochs for the three cases of optimizers. Also, mention the best choice among the three.

Responsible TAs:

Please write to the following TAs for any doubt or clarification regarding Assignment 4.
Sumanta Mishra: sumantamishra22@gmail.com

Deadline:

The deadline for submission is **31st March (Thursday), 11:55PM, IST**. Irrespective of the time in your device, once submission in moodle is closed, no request for submission post-deadline will be entertained. No email submission will be considered. So, it is suggested that you start submitting the solution at least one hour before the deadline.