CSE 591 - Introduction to Deep Learning in Visual Computing
Mini Project - 2
Due Date : Feb 16, 2018 before 11:59 PM

Baoxin Li
Hemanth Venkateswara

Spring 2018

Submission

Submissions will be handled through asu blackboard. If there are programming tasks in the mini projects, the code must be written by you. Although you may seek advice from online forums, you shall not copy or borrow code from any sources. When in doubt, email the instructor/TA for clarifications on borrowing code.

Academic Integrity

You are expected to maintain the utmost level of academic integrity in the project. Any violation of the code of academic integrity will be reported to the dean for official actions. It is an academic violation to copy, to include text from other sources, including online sources for both material and code, without proper citation and licensing. To get a better idea of what constitutes plagiarism, consult the ASU policy on student obligations. This is a serious violation and evidence of plagiarism or academic dishonesty, will likely result in failing the course and at worse can lead to disqualification from your degree program. Please contact the TA before borrowing material when unsure. Refer to the section on licensing in the code submission requirements.

1 Mini Project 2

For this project you are going to implement a two layer neural network for a binary classifier and a multi layer neural network for a multiclass classifier. Some starer code is provided on blackboard(content/mini projects/mini project2).

1.1 Two layer neural network for a binary classifier (40%)

Dataset: MNIST - digits 2 and 3
Data can be downloaded using ('download_mnist.sh'). Train a two layer network (1 hidden layer dimension=500) for binary classification. Train with the following parameters.
learning_rate = 0.1
num_iterations = 1000
Filename - `twoLayerBinary_starter.py`
Outputs: (1) The training and testing accuracies. (2) Plot of train error vs iterations

1.2 Multi-layer neural network for multi-class classifier(60%)

Dataset: MNIST - digits 0 to 9
Data can be downloaded using ('download_mnist.sh'). Train a multi-layer neural network to classify MNIST. The MNIST dataset has 60,000 images which may be too large for batch gradient descent. Therefore, train with merely 6000 samples and test with 1000 samples. The starter code has this set up.
Modify the num_iterations, learning_rate and decay_rate to improve training.
The program must be able to create and train a multilayer network based on command line arguments.

```
To create a network with 1 hidden layer of dimensions 800 Run the program as:
python deepMultiClassNetwork_starter.py "[784,800]"
 The network will have the dimensions [784,800,10]
784 is the input size of digit images (28pix x 28pix = 784)
10 is the number of digits


To create a network with 2 hidden layers of dimensions 800 and 500 Run the
program as:
python deepMultiClassNetwork_starter.py "[784,800,500]"
 The network will have the dimensions [784,800,500,10]
784 is the input size of digit images (28pix x 28pix = 784)
10 is the number of digits
```

Filename - `deepMultiClassNetwork_starter.py`
Outputs: (3) The training and testing accuracies. (4) Plot of train error vs iterations

Notes

- Your code should be written in Python (either 2.x or 3.x), remember to indicate which version you are using at the beginning of the code.
- You can use packages like numpy, scipy, etc., but write your own code for the gradient descent.

- The plots should be included in a single pdf file. Title the plots to avoid confusion. You should submit a zip file, which contains the .pdf and .py files.
- The zip file should be named as ”(your student id)_(your last name)_mp1.zip”. (For example, 1246813579_Li_mp2.zip)