ECSE 4965/6965

Introduction to Deep Learning

Program Assignment 2

Due date: 11:59 pm, Feb. 23th

In this programming assignment, you will implement techniques to learn a multi-class logistic regressor for image digit classification on the MNIST dataset (https://en.wikipedia.org/wiki/MNIST_database). The classifier will take an image of a hand-written numerical digit between 1 and 5 as input and classify it into one of 5 classes corresponding to digit 1 to 5 respectively. Given the training data, you will follow the equations in the lecture notes to train a multi-class logistic regressor using the gradient descent method and then evaluate its performance on the given testing data. For training, you will learn the discriminant function regression parameters $\Theta=(\mathbf{W_k},W_{k,0})^t$, where k=1,2,3, 4, 5, $\mathbf{W_k}$ is a vector for kth discriminant function, and $W_{k,0}$ is its bias.

Specifically, given the training data $\mathbf{D}=\{\mathbf{x}[m], \boldsymbol{y}[m]\}$, m=1,2, …, M, where $\mathbf{x}[m]$ is a grayscale image of 28x28 (a 784x1 vector) and $\boldsymbol{y}[m]$ is output vector that follows the 1 of K encoding, with kth element of being 1 and the rest being 0. You will implement the following methods to learn the parameters Θ

1. For students taking this class at 4000 level, implement in Tensorflow the gradient descent method to solve for Θ iteratively using all data in $\mathbf{D}$. Initialize Θ to small values and literately update Θ with appropriate learning rate until convergence. Save Θ and plot the $(\mathbf{W}_k)$ for each class using matplotlib.pyplot functions (see below for details).

2. For students taking this class at 6000 level, implement in Tensorflow the Stochastic Gradient Descent method with L2 regularization. Initialize Θ to small values and literately update Θ with appropriate learning rate and regularization parameter λ until convergence. Save Θ and plot the $(\mathbf{W}_k)$ for each class using matplotlib.pyplot functions (see below for details).

3. Using the given testing dataset $\mathbf{T}$, evaluate the performance of your trained classifier by computing the classification error for each digit as well as the average classification errors for all five digits. The classification error for each digit is computed as the ratio of incorrect classification to the total number images for that digit

Do not use Tensorflow's existing gradient descent and stochastic gradient descent functions. Submit your Tensorflow code via LMS, along with the required classification errors, weight plots, and the saved weights $\mathbf{W}$.

Data format

The training data **D** contains 25112 training images (00001.jpg – 25112.jpg) in train_data folder, and their labels are stored in train_label.txt with each raw being the label of the corresponding image.

The testing data **T** contains 4982 testing images (00000.jpg – 04982.jpg) in test_data folder, and their labels are stored in test_label.txt.

Load images and normalize

Load the images in the order

Convert it to vector and normalize it to [0,1] by dividing the intensity of each pixel by 255

The following link teaches how to load images:

http://learningtensorflow.com/lesson3/

Output

Use the code below to save the learned parameters matrix $\mathbf{W}$ in the following format

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 & \mathbf{W}_3 & \mathbf{W}_4 & \mathbf{W}_5 \\ W_{1,0} & W_{2,0} & W_{3,0} & W_{4,0} & W_{5,0} \end{bmatrix}$$

$\mathbf{W}_k$ is the learnt weights for the kth digit and $W_{k,0}$ is the corresponding bias.

```
import pickle
filehandler = open("multiclass_parameters.txt","wb")
pickle.dump(W, filehandler)
filehandler.close()
```

Plot

Use the following function to plot $\mathbf{W}_k$ as an image for each digit:

```
import matplotlib.pyplot as plt
Img = W_k.reshape(28,28)
plt.imshow(img)
plt.colorbar()
plt.show()
```
