

Assignment 2: Perceptron Learning

Deep Learning — Spring 2017
Information Technology University (ITU), Punjab



Figure 1: Randomly selected hundred training examples from MNIST dataset

Brief

- Due date: March 7, 2017
- Implement / edit relevant sections of the starter code (*perceptron.m*)
- Required files: Report document, code in a zip file, and a Readme.txt file explaining how to run your code.
- Submission: Google Class

Overview

The goal of this assignment is to implement the perceptron algorithm discussed in the class. Specifically, we will implement a single layer perceptron for binary classification. The dataset we are going to use is MNIST database of handwritten digits. It can be downloaded from the following link (<http://yann.lecun.com/exdb/mnist/>). The learning objective of this assignment is to understand challenges is designing a classifier with higher accuracy.

Details and Starter Code

The starter code (*perceptron.m*) loads the data into Matlab. It also setup the training and testing examples with their labels depending upon which “pair” of digits you’ve chosen. Furthermore, it randomly selects 100 examples from the training set, reshape and arrange them into a 10x10 grid, and then show them for visualization.

- Download the following four files and extract them to a folder i.e. "MNIST".
 1. train-images.idx3-ubyte
 2. train-labels.idx1-ubyte
 3. t10k-images.idx3-ubyte
 4. t10k-labels.idx1-ubyte
- Set path of your folder in the *perceptron.m* file e.g. `train_data_dir = '../MNIST/';`
- Implement the perceptron training code in the "Training" section of the *perceptron.m*. It should ideally be a function that takes training examples and their labels as input and return the trained model.
- Implement the testing function that takes the learned model and the test example(s) as input and return the predictions (classification result for each test example).
- Lastly, write the evaluation function that takes the predictions and ground truth labels of the test set as input and compute the following quantities.
 - True positive
 - False positive
 - True negative
 - False negative
 - Precision
 - Recall
 - Accuracy

Write up

For this assignment, and all other assignments and projects, you must write a report. In the report you will describe your algorithm, critical decision you made, important things you learned, or any decisions you made to write your algorithm a particular way. You are required to report the accuracy you achieved. You are also required to report the details about your experiments and environment setup, so that the results can be recreated.

Extra Credit

For all extra credit, be sure to include quantitative analysis showing the impact of the particular method you've implemented. Each item is "up to" some amount of points because trivial implementations may not be worthy of full extra credit.

- **up to 20 points** — Implement logistic regression and compare its results with your single layer perceptron classifier.
- **up to 20 points** — Use off the shelf SVM (Matlab or VLFeat) and implement one vs all SVM for multi-class classification. Report accuracy and confusion matrix. Talk to me before you start implementing it.