CS 419 - Project Abstract

HANDWRITING RECOGNITION BY MACHINE LEARNING

**Aim** -: To code various classifiers from scratch, to perform digit recognition using the MNIST data set. We then wish to the performances of various classifiers on the data set. The Image can be of handwritten document or printed document. It can be used as a form of data entry from printed records.

**Background -:** The MNIST database is a large database of handwritten digits that is used to train various image processing systems and machine learning classification systems. It was created using the samples from NIST's original datasets. The black and white images like the one shown below,from NIST were normalized to fit into a 28x28 = 728 pixels bounding box and anti-aliased, which introduced grayscale levels, as can be clearly seen in the image. Each pixel is provided as a number between 0-255 indicating its density.

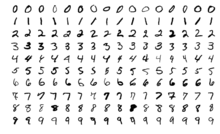
The MNIST database contains 70,000 standardized images of handwritten digits and consists of 4 files:

(1) A training set of 60,000 images

(2) The labels (correct answers) for the training set

(3) A testing set of 10,000 images

(4) The labels (correct answers) for the testing set

****

**Project Details -:** We plan to implement the following classifiers from scratch in the project -:

1. Multilayer perceptron
2. Multiclass Logistic regression
3. SVMs
4. Neural Networks

There will be emphasis on preprocessing the MNIST dataset also, to obtain better results, or to see dependence of the classifier performance on different pre-processing techniques, like feeding the fourier transforms of the MNIST image to the neural network.

1. Perceptron - The perceptron consists of a core which is connected to at least two inputs and one output. It receives inputs, does some processing and spits out an output. The input connections of the perceptron are assigned different weights. These weights are values between 0 to 1 and can be thought of as an expression of priority or importance of the respective input. In the beginning, i.e. for our first guess, we use random numbers for these weights.The perceptron’s output is simply the sum of the weighted inputs.This perceptron’s output is the neural network’s guess for the desired result. In the scenario of supervised learning, as described above, this output is then compared to the desired output or target and the difference between both (the error) is calculated.Now the network not only knows that its guess was wrong but also by how much (error). We now want the network to try again, but not using the same but new values. Since the input to the system is a given and fix, the only values we can change are the weights of these inputs.The learning rate defines the interval for changing the weights, i.e. a high rate mean faster but likely less accurate change, a low rate mean more accurate but slower change.
2. Multiclass logistic regression - Logistic regression is a well-known method in statistics that is used to predict the probability of an outcome, and is particularly popular for classification tasks. The algorithm predicts the probability of occurrence of an event by fitting data to a logistic function. In multiclass logistic regression, the classifier can be used to predict multiple outcomes.
3. Support Vector Machines - An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.
4. Neural Networks - We are going to use a very simple 1-layer neural network to recognize the handwritten digits in the MNIST database. A neural network represents a different computing approach to solving a problem. Instead of receiving an exact method how to calculate the output, the computer receives lots of reference data and starts guessing. Every time it guesses the output, we’re letting it know whether its prediction was correct or not. If it was incorrect, we’re telling it know by how much the prediction was off the desired result, so the network can adjust accordingly and try a slightly different guess next time. This process is repeated a large number of times. With every guess the network’s predicted result will move closer to the actual desired one.

**Extension to the project -:** To go one step further, if time permits, we also wish to try person specific handwriting classification on a very primitive scale, i.e. to identify whether a digit is written by the same person or not, using a neural network. A very important use case for this can be to detect fake signatures

|  |  |
| --- | --- |
| GROUP MEMBERS | ROLL NUMBERS |
| CHINMAY TALEGAONKAR | 15D070046 |
| KAVYA PRUDHVI | 15D070014 |
| DARSHAN TANK | 150020012 |
| TANYA GUPTA | 150040010 |