HANDWRITTEN DIGITS RECOGNITION SYSTEM

Report submitted in partial fulfillment of the requirement for the degree

of

Bachelor of Technology
In
Computer Science & Engineering

By
Shivansh Bhasin
To
Ms. Shaily Malik, Assistant Professor, C.S.E

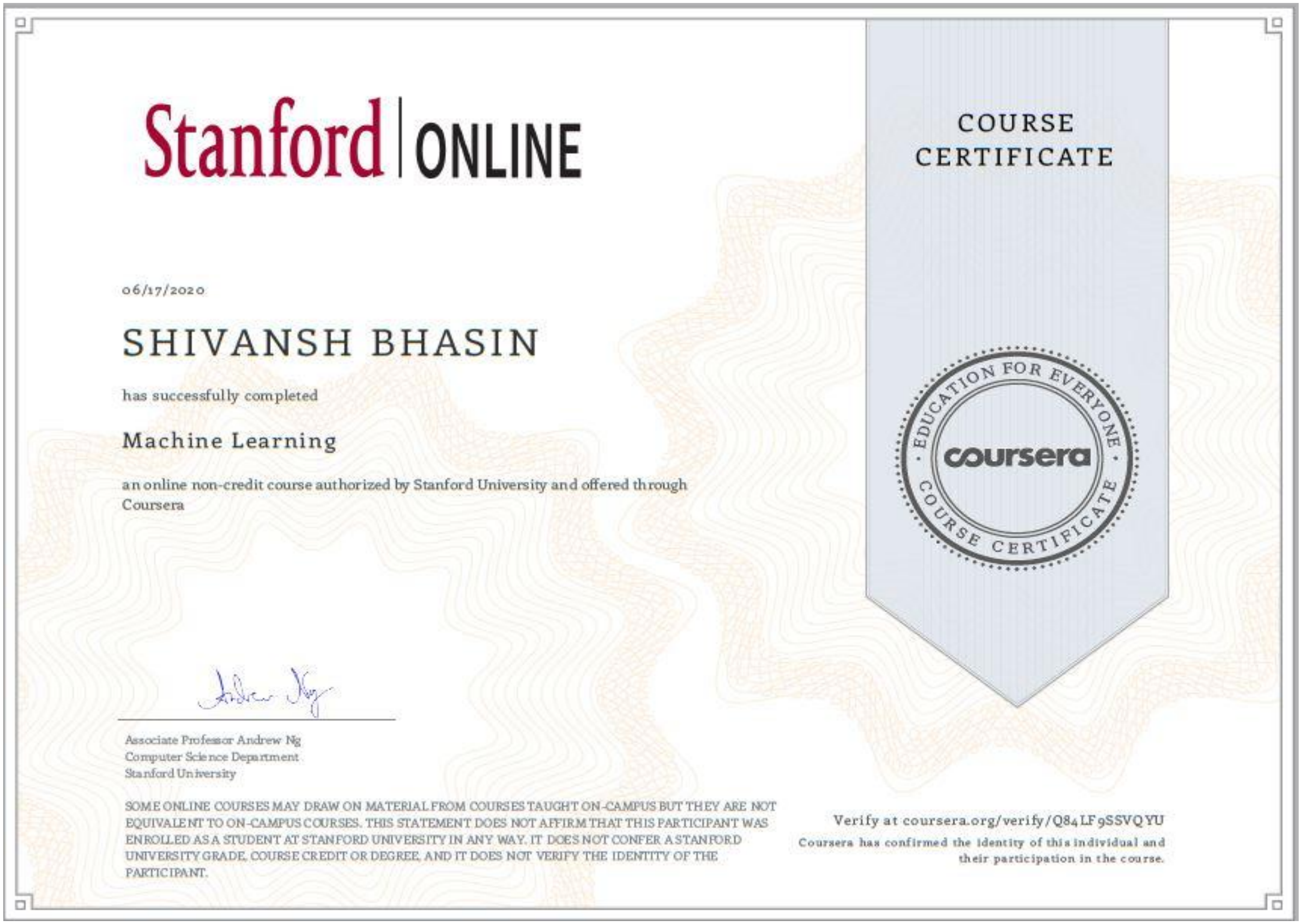
Maharaja Surajmal Institute of Technology
Affiliated to Guru Gobind Singh Indraprastha University
Janakpuri, New Delhi-58

2018-22

CERTIFICATE

This is to certify that Project Report entitled “**Handwritten Digit Recognition System**” which is submitted by **SHIVANSH BHASIN**, Enroll No:**75515002718** of Maharaja Surajmal Institute of Technology in partial fulfillment of the requirement for the award of degree BTech (C.S.E) to GGSIP University, Dwarka, Delhi is a record of the candidate own work carried out by him under my/our supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Supervisor’s Signature



Stanford | ONLINE

COURSE CERTIFICATE

06/17/2020

SHIVANSH BHASIN

has successfully completed

Machine Learning

an online non-credit course authorized by Stanford University and offered through Coursera

[illegible]

Associate Professor Andrew Ng
Computer Science Department
Stanford University

SOME ONLINE COURSES MAY DRAW ON MATERIAL FROM COURSES TAUGHT ON-CAMPUS BUT THEY ARE NOT EQUIVALENT TO ON-CAMPUS COURSES. THIS STATEMENT DOES NOT AFFIRM THAT THIS PARTICIPANT WAS ENROLLED AS A STUDENT AT STANFORD UNIVERSITY IN ANY WAY. IT DOES NOT CONFER A STANFORD UNIVERSITY GRADE, COURSE CREDIT OR DEGREE, AND IT DOES NOT VERIFY THE IDENTITY OF THE PARTICIPANT.

Verify at coursera.org/verify/Q84LF9SSVQYU
Coursera has confirmed the identity of this individual and their participation in the course.

ACKNOWLEDGEMENT

I offer my sincere thanks and humble regards to Mr. Andrew Ng Associate Professor Computer Science Department, Stanford University for imparting us very valuable professional course in Machine Learning. I am thankful to him as he has been a constant sourceof advice, motivation and inspiration. I am also thankful to him for giving his suggestions and encouragement throughout the course.I also want to thank Coursera Support Team for constantly helping me in this 11 week long course . I take the opportunity to express my gratitude and thanks to Coursera for providing me opportunity to utilize their resources for the completion of the course and project. I am also thankful to my family and friends for constantly motivating me to complete the project and providing me an environment which enhanced my knowledge .

Student’s Signature

ABSTRACT

The aim of this project is to implement one-vs-all logistic regression and neural networks to recognize handwritten digits (0-9). The Dataset contains 5000 training examples of handwritten digits that was taken from standard MNIST database. Each pixel is represented by a floating point number indicating the grayscale intensity at that location. The 20 by 20 grid of pixels is ‘unrolled’ into a 400-dimensional vector. The second part of the training set is a 5000-dimensional vector y that contains labels for the training set. To make things more compatible with Octave indexing, where there is no zero index, we have mapped the digit zero to the value ten. Therefore, a 0 digit is labeled as 10, while the digits 1 to 9 are labeled as 1 to 9 in their natural order.

We have used multiple one-vs-all logistic regression models to build a multi-class classifier. After training the one-vs-all classifier, we have used it to predict the digit contained in a given image. While training one-vs-all classifier we have seen that the training set accuracy is around 94%.

Logistic regression cannot form more complex hypotheses as it is only a linear classifier.So in order to overcome this we have also implemented Neural Networks on the same data set, in this project. The neural network will be able to represent complex models that form non-linear hypotheses.

So after implementing one-vs-all logistic regression model we have implemented the feedforward propagation algorithm of Neural Networks to use our weights for prediction. Here our neural network has 3 layers { an input layer, a hidden layer and an output layer. Recall that our inputs are pixel values of digit images. Since the images are of size 20 by 20, this gives us 400 input layer units (excluding the extra bias unit which always outputs +1).After we have implemented the feedforward propagation algorithm. We have calculated the training set accuracy again and that comes around 97%, which is more than previous one.

So for the smooth functioning of our recognition system,we have implemented above mentioned algorithm step by step. And tried best to predict every digit more precisely.

Keywords: Handwritten Digits, Image Correlation, Matrix Matching, Machine Learning, Logistic Regression , one-vs-all classifier, Neural Networks

Table of Content

List of Figures

Chapter 1

1. Introduction-Need & Objective

The aim of this project is to recognize handwritten digits (0-9). It has been shown in pattern recognition that no single classifier performs the best for all pattern classification problems consistently. Hence, the scope of the project also included the elementary study the different classifiers and combination methods, and evaluate the caveats around their performance in this particular problem of handwritten digit recognition.
It is easy for the human brain to process images and analyse them. When the eye sees a certain image, the brain can easily segment it and recognize its different elements. The brain automatically goes through that process, which involves not only the analysis of this images, but also the comparison of their different characteristics with what it already knows in order to be able to recognize these elements. There is a field in computer science that tries to do the same thing for machines, which is Image Processing.
One of the narrow fields of image processing is recognizing characters from an image, which is referred to as Optical Character Recognition (OCR). This method is about reading an image containing one or more characters, or reading a scanned text of typed or handwritten characters and be able to recognize them. A lot of research has been done in this field in order to find optimal techniques with a high accuracy and correctness. The most used algorithms that proved a very high performance are machine learning algorithms like Neural Networks and Support Vector Machine.
One of the main applications of OCR is recognizing handwritten characters. In this project, we will focus on building a mechanism that will recognize handwritten digits. Automated handwritten digit recognition is widely used today - from recognizing zip codes (postal codes) on mail envelopes to recognizing amounts written on bank checks. We will be reading images containing handwritten digits extracted from the MNIST database and try to recognize which digit is represented by that image. For that we will use basic Image Correlation techniques, also referred to as Matrix Matching. This approach is based on matrices manipulations, as it reads the images as matrices in which each element is a pixel. It overlaps the image with all the images in the reference set and find the correlation between them in order to be able to determine the digit it represents.
In order to make our recognition system we have majorly used two broader concept of Machine Learning in this project i.e Logistic Regression and Neural Networks.We have used one-vs-all classifier model from Logistic Regression and FeedForward Propagation Algorithm from Neural Networks.

After Moving ahead in this project report, first see this brief introduction to Machine Learning.

Introduction to Machine Learning

Machine Learning is used anywhere from automating mundane tasks to offering intelligent insights, industries in every sector try to benefit from it. You may already be using a device that utilizes it. For example, a wearable fitness tracker like Fitbit, or an intelligent home assistant like Google Home. But there are much more examples of ML in use.

- Prediction — Machine learning can also be used in the prediction systems. Considering the loan example, to compute the probability of a fault, the system will need to classify the available data in groups.
- Image recognition — Machine learning can be used for face detection in an image as well. There is a separate category for each person in a database of several people.
- Speech Recognition — It is the translation of spoken words into the text. It is used in voice searches and more. Voice user interfaces include voice dialing, call routing, and appliance control. It can also be used a simple data entry and the preparation of structured documents.
- Medical diagnoses — ML is trained to recognize cancerous tissues.
- Financial industry and trading — companies use ML in fraud investigations and credit checks.

2. Methodology

2.1 Getting Familiar with the Tools

The first step we had to go through while working on this project was getting familiar with the tools used, i.e., Octave and the MNIST dataset. We have used Octave Version 5.2. 0 .After setting up the environment for Octave to work perfectly and downloading the dataset, I have started experimenting with both in order to get familiar with them and know how to use them easily in the future.
Since all the programming is mainly done in Octave, we had to download it along with its Graphical User Interface into the computer, and learn a little bit about its functions and how to use it. Octave is a free software which makes it very easy to work with matrices and vectors and is very efficient in performing calculations on them. I have started learning how to use it and looking for its main functions that I will be using in the implementation of the project. For that, I have used some random images of digits to see how they can be read and modified as well as how to apply some computations on them.
Moreover, I had to investigate the format of the MNIST dataset and get familiar with its representation. The MNIST dataset, which was used to create our test set, contains thousands of handwritten digits, represented as matrices. It has been used in the development of several programs and projects with the same aim as ours. After downloading the file which contains the handwritten digits, I have loaded it on Octave in order to visualize the images and figure out how to use and manipulate them.

2.2 Creating the reference and test set

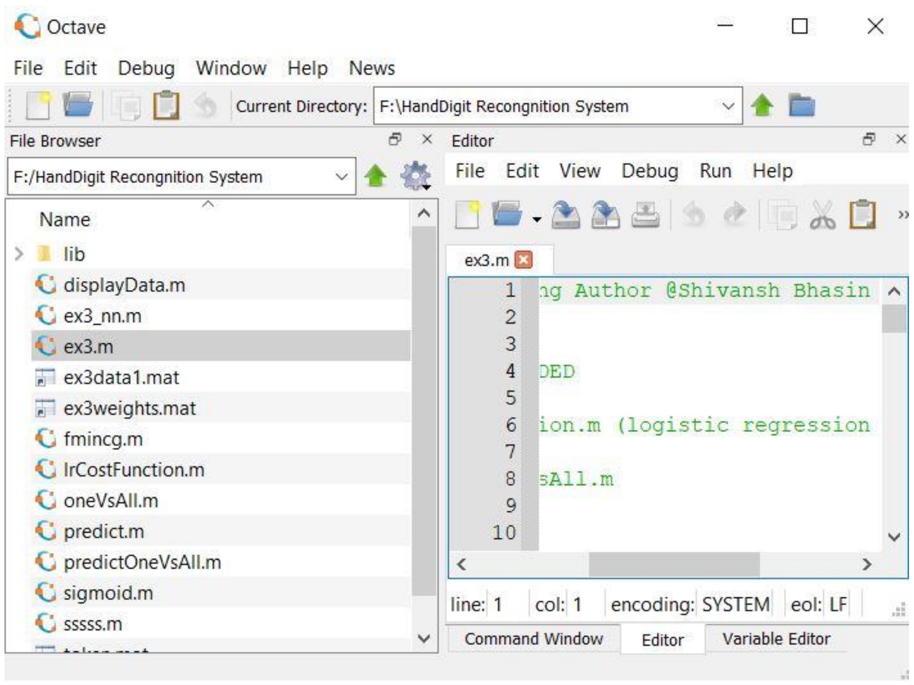
One of the main steps in the project is creating the reference and the test set that will both be used in the implementation phase.
The test set is to be used in order to assess the performance of the program and evaluate its success or error rate. It is to be taken from the MNIST dataset, since it contains the handwritten digits that we intend to recognize and identify. We have taken data set in file name “ex3data1.mat” that contains 5000 training examples of handwritten digits. The .mat format means that that the data has been saved in a native Octave matrix format, instead of a text (ASCII) format like a csv-file. These matrices can be read directly into our program by using the load command of Octave. After loading, matrices of the correct dimensions and values will appear in our program's memory.
We have taken 5000 training examples in our file “ex3data1.mat”, where each training example is a 20 pixel by 20 pixel grayscale image of the digit. Each pixel is represented by a floating point number indicating the grayscale intensity at that location. The 20 by 20 grid of pixels is unrolled into a 400-dimensional vector. Each of these training examples becomes a single row in our data matrix X. This gives us a 5000 by 400 matrix X where every row is a training example for a handwritten digit image.
The second part of the training set is a 5000-dimensional vector y that contains labels for the training set. To make things more compatible with Octave indexing, where there is no zero index, we have mapped the digit zero to the value ten. Therefore, a 0 digit is labeled as 10, while the digits 1 to 9 are labeled as 1 to 9 in their natural order.

3. Software Used

For making this Handwritten Digits Recognition System we have used Octave language. So In order to implement this we have used GNU OCTAVE software. We have used Octave Version 5.2. 0

About The Software: GNU Octave is software featuring a high-level programming language, primarily intended for numerical computations. Octave helps in solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with MATLAB. It may also be used as a batch-oriented language. Since it is part of the GNU Project, it is free software under the terms of the GNU General Public License.

Fig 1-Official Logo of GNU Octave


Octave
File Edit Debug Window Help News
Current Directory: F:\HandDigit Recongnition System
File Browser
F:/HandDigit Recongnition System
Name
Editor
File Edit View Debug Run Help
ex3.m
line: 1 col: 1 encoding: SYSTEM eol: LF
Command Window Editor Variable Editor

lib
displayData.m
ex3_nn.m
ex3.m
ex3data1.mat
ex3weights.mat
fmincg.m
lrCostFunction.m
oneVsAll.m
predict.m
predictOneVsAll.m
sigmoid.m
sssss.m

```
ng Author @Shivansh Bhasin


DED

ion.m (logistic regression

sAll.m
```

Fig 2-Glimpse of Octave GUI window

4. About Organization

This Project was made after the completion of 11 week Machine Learning Course from Coursera which was taught by Mr. Andrew Ng, Associate Professor Computer Science Department, Stanford University. Also he the founder of Coursera.

About Coursera : Coursera is an American massive open online course (MOOC) provider founded in 2012 by Stanford University's computer science professors Andrew Ng and Daphne Koller that offers massive open online courses (MOOC), specializations, degrees, professional and mastertrack courses.

Coursera works with universities and other organizations to offer online courses, certifications, and degrees in a variety of subjects, such as engineering, data science, machine learning, mathematics, business, financing, computer science, digital marketing, humanities, medicine, biology, social sciences, 3000 plus variety of courses giving students a very broad range of information & experience in different fields.

About the ML Course : Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level AI. In this course, I have learnt about the most effective machine learning techniques, and gain practice implementing them and getting them to work for myself. More importantly, I have learnt not only the theoretical underpinnings of learning, but also gain the practical know-how needed to quickly and powerfully apply these techniques to new problems. Finally, I have learnt about some of Silicon Valley's best practices in innovation as it pertains to machine learning and AI.

This course provides a broad introduction to machine learning, data mining, and statistical pattern recognition. Topics include: (i) Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks). (ii) Unsupervised learning (clustering, dimensionality reduction, recommender systems, deep learning). (iii) Best practices in machine learning (bias/variance theory; innovation process in machine learning and AI). The course will also draw from numerous case studies and applications, so that you'll also learn how to apply learning algorithms to building smart robots (perception, control), text understanding (web search, anti-spam), computer vision, medical informatics, audio, database mining, and other areas.

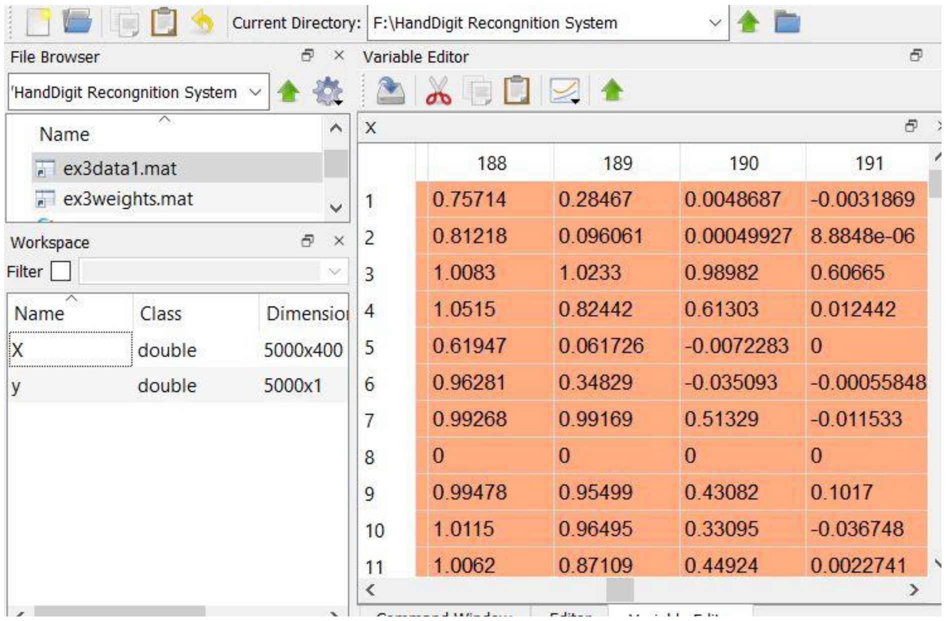
Chapter-2

1. Project Design

In order to make this Handwritten Digit Recognition System Various steps comes into play like selecting the correct dataset, converting it into correct format. Then comes selection of appropriate machine learning algorithm as said above we have used Logistic Regression and Neural Network in this System. So we will discuss the blueprint of this project one by one.

1.1 About Dataset

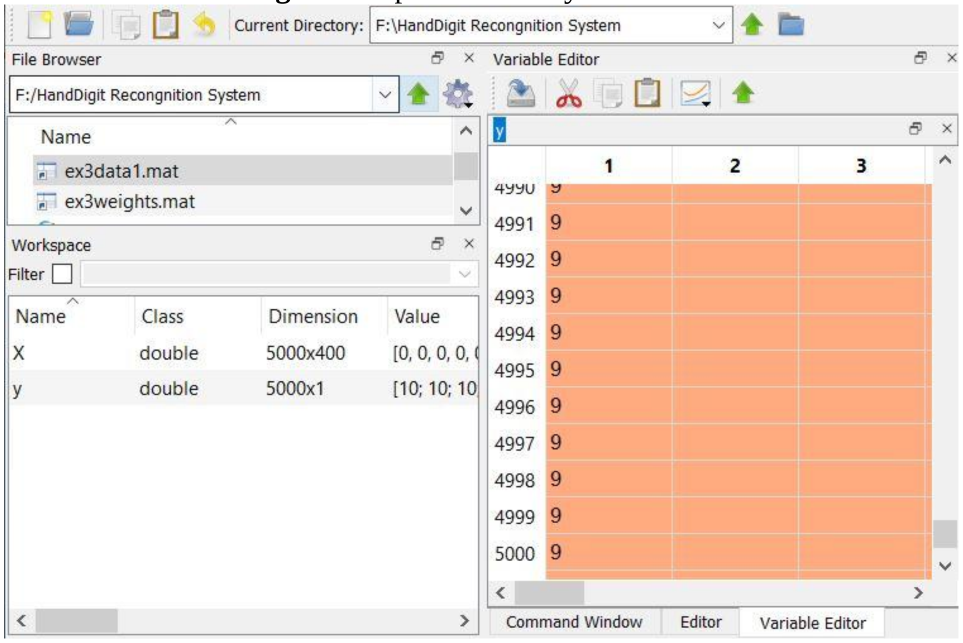
It is very necessary to know the kind of data we are using before we start the design and the implementation of the program. That is why we had to have a look at its format to understand how it is represented before creating the reference and the test set.
The dataset that we have downloaded from the MNIST database contains contains 5000 training examples of handwritten digit. We have stored the data of images in file name “ex3data1.mat”. The .mat format means that that the data has been saved in a native Octave matrix format, instead of a text (ASCII) format like a csv-file. These matrices can be read directly into our program by using the load command of Octave. After loading, matrices of the correct dimensions and values will appear in our program's memory.
We have taken 5000 training examples in our file “ex3data1.mat”, where each training example is a 20 pixel by 20 pixel grayscale image of the digit. Each pixel is represented by a floating point number indicating the grayscale intensity at that location. The 20 by 20 grid of pixels is unrolled into a 400-dimensional vector. Each of these training examples becomes a single row in our data matrix X. This gives us a 5000 by 400 matrix X where every row is a training example for a handwritten digit image.
The second part of the training set is a 5000-dimensional vector y that contains labels for the training set. To make things more compatible with Octave indexing, where there is no zero index, we have mapped the digit zero to the value ten. Therefore, a 0 digit is labeled as 10, while the digits 1 to 9 are labeled as 1 to 9 in their natural order.


Current Directory: F:\HandDigit Recongnition System
File Browser
'HandDigit Recongnition System
Name
ex3data1.mat
ex3weights.mat
Workspace
Filter
Variable Editor
X

Name	Class	Dimensio
X	double	5000x400
y	double	5000x1

	188	189	190	191
1	0.75714	0.28467	0.0048687	-0.0031869
2	0.81218	0.096061	0.00049927	8.8848e-06
3	1.0083	1.0233	0.98982	0.60665
4	1.0515	0.82442	0.61303	0.012442
5	0.61947	0.061726	-0.0072283	0
6	0.96281	0.34829	-0.035093	-0.00055848
7	0.99268	0.99169	0.51329	-0.011533
8	0	0	0	0
9	0.99478	0.95499	0.43082	0.1017
10	1.0115	0.96495	0.33095	-0.036748
11	1.0062	0.87109	0.44924	0.0022741

Fig 3: Glimpse of 5000 by 400 matrix X



Name	Class	Dimension	Value
X	double	5000x400	[0, 0, 0, 0, (
y	double	5000x1	[10; 10; 10

	1	2	3
4990	9		
4991	9		
4992	9		
4993	9		
4994	9		
4995	9		
4996	9		
4997	9		
4998	9		
4999	9		
5000	9		

Fig 4: Glimpse of 5000-dimensional vector y

As said above X is 5000 by 400 matrix,so it is not possible to display the entire matrix .So our code randomly selects 100 rows from X and passes those rows to the 'displayData' function. This function maps each row to a 20 pixel by20 pixel grayscale image and displays the images together.
After we have run this step, we have see an image like Figure 5

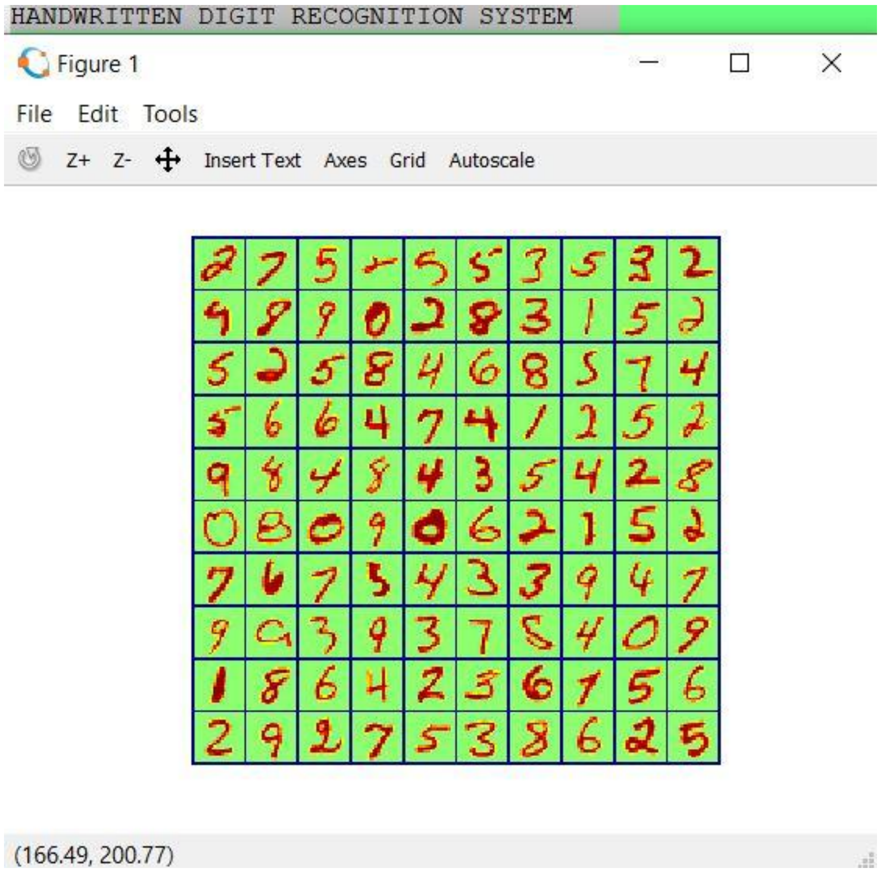
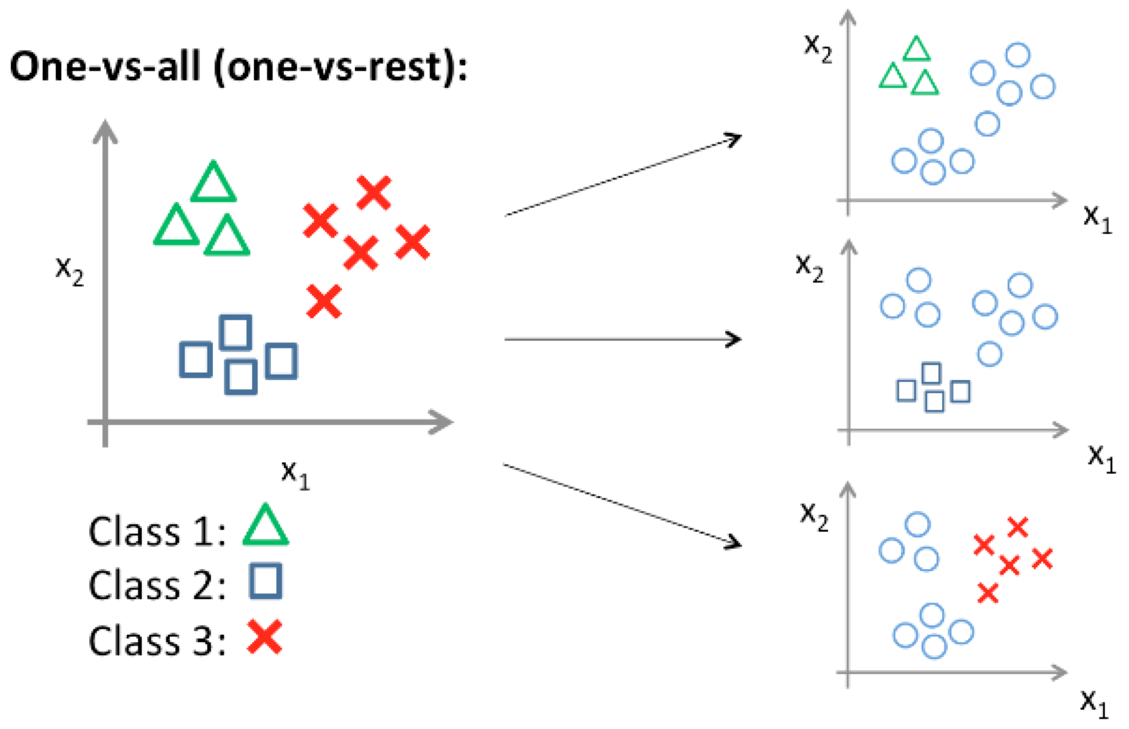

HANDWRITTEN DIGIT RECOGNITION SYSTEM
Figure 1
File Edit Tools
Z+ Z- Insert Text Axes Grid Autoscale
(166.49, 200.77)

Fig 5: Randomly Selected 100 grayscale imaged

1.2 Logistic Regression

Introduction to Logistic Regression:- Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.
In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).
Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc**.**

One-vs-all Logistic Regression:-As said above we have used one-vs all classification method.So In one-vs-All classification, for the N-class instances dataset, we have to generate the N-binary classifier models. The number of class labels present in the dataset and the number of generated binary classifiers must be the same.We can understand the one-vs-all classification below



1.2 Vectorizing Logistic Regression

We have used multiple one-vs-all logistic regression models to build a multi-class classier. Since there are 10 classes,so we **need to train 10 separate logistic regression classifiers**. To make this training efficient, it is important to ensure that your code is well vectorized. So we have implemented a vectorized version of logistic regression that does not employ any for loops. Now we have to write to write the unregularized cost function.So the formulae that was applied to compute the cost is:

J = ((1 / m) * sum(-y'*log(sigmoid(X*theta)) - (1-y)'*log(1 - sigmoid(X*theta)))) + (lambda/(2*m))*sum(theta(2:length(theta)).*theta(2:length(theta))) ;

Sigmoid Function:-

It is a mathematical function having a characteristic that can take any real value and map it to between 0 to 1 shaped like the letter “S”. The sigmoid function also called a logistic function.

$$Y = 1 / 1+e^{-z}$$

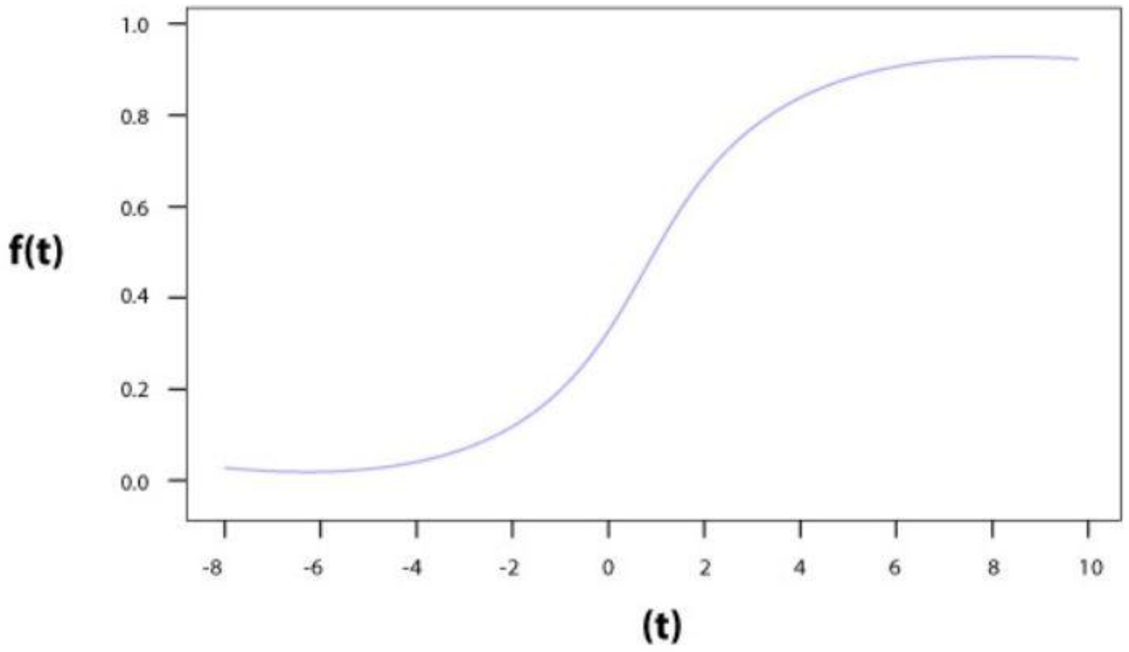


Fig-6 : Sigmoid Function

So, if the value of z goes to positive infinity then the predicted value of y will become 1 and if it goes to negative infinity then the predicted value of y will become 0. And if the outcome of the sigmoid function is more than 0.5 then we classify that label as class 1 or positive class and if it is less than 0.5 than we can classify it to negative class or label as class 0.

1.3 One-vs-all Classifcation

After vectorization we have implemented one-vs-all classification by training multiple regularized logistic regression classifiers, one for each of the K classes in our dataset. In the handwritten digits dataset, K = 10, but our code should work for any value of K.
We have made a file ‘ oneVsAll.m’ to train one classifier for each class. In particular,our code has returned all the classier parameters in a matrix , where each row of matrix corresponds to the learned logistic regression parameters for one class. We have done this with a for-loop from 1 to K, training each classier independently.The y argument to this function is a vector of labels from 1 to10, where we have mapped the digit 0 to the label 10 (to avoid confusions with indexing).
When training the classifieer for class k , we want a multi-dimensional vector of labels y, where y belongs to 0; 1 indicates whether the j-th training instance belongs to class k (yj = 1), or if it belongs to a different class.We have used logical arrays in order to implement this task.

1.3.1 One-vs-all Prediction

After training our one-vs-all classier, we have used it to predict the digit contained in a given image. For each input,we have computed the probability that it belongs to each class using the trained logistic regression classifiers. Our one-vs-all prediction function will pick the class for which the corresponding logistic regression classifier outputs the highest probability and return the class label (1, 2,..., or K) as the prediction for the input example.
Once we have ,we have called the predict function and saw that the training set accuracy is about 94% (i.e., it classifies 94% of the examples in the training set correctly).

1.4 Neural Networks

Our Handwritten Digit Recognition System is now working ,but in order to make it more efficient we implemented the concept of Neural Networks too.
We have implemented a neural network to recognize handwritten digits using the same training set as before. The neural network will be able to represent complex models that form non-linear hypotheses. We have used parameters from a neural network that we have already trained. We have implemented the feedforward propagation algorithm to use our weights for prediction.

More About Neural Network:

Neural Networks mimics the working of how our brain works. They have emerged a lot in the era of advancements in computational power.

Deep learning is the acronym for Neural Networks, the network connected with multilayers. The layers are composited form nodes. A node is just a perception which takes an input performs some computation and then passed through a node's activation function, to show that up to what context signal progress proceeds through the network to perform classification.

Algorithm for basic Neural Network
1) Initializing the weights randomly (not by keeping them zero)

2) Implementing forward propagation to achieve $h\theta(x(i))$.

3) Compute cost

4) Now evaluate backpropagation to compute partial derivatives and use gradient checking to confirm that backpropagation is working fine. Then disable gradient checking.

5) Use gradient descent or any built-in optimization function to minimize the cost function with weights of theta

1.4.1 Neural Networks Model Representation

Our neural network is shown in Figure 7. It has 3 layers { an input layer, ahidden layer and an output layer. Recall that our inputs are pixel values ofdigit images. Since the images are of size 20   20, this gives us 400 input layer units (excluding the extra bias unit which always outputs +1). As before, the training data will be loaded into the variables X and y.

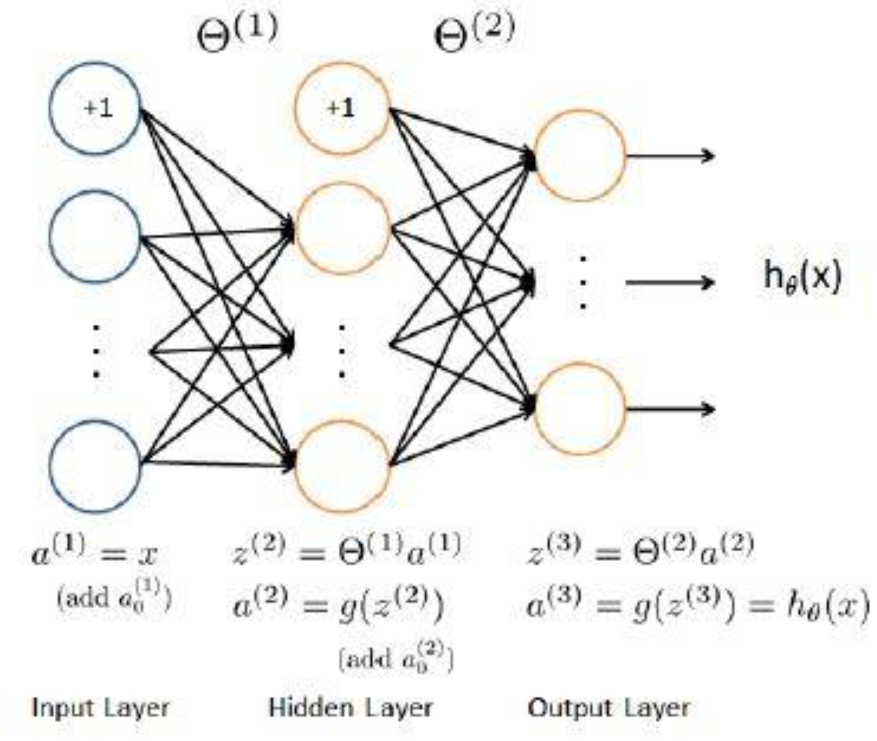


Fig 7:- Neural Networks Model Representation

1.4.2 FeedForward Propagation Algorithm

In order to generate some output, the input data should be fed in the forward direction only. The data should not flow in reverse direction during output generation otherwise it would form a cycle and the output could never be generated. Such network configurations are known as feed-forward network. The feed-forward network helps in forward propagation.
For a feed-forward neural network, the gradient can be efficiently evaluated by means of error backpropagation. The key idea of backpropagation algorithm is to propagate errors from the output layer back to the input layer by a chain rule.

Chapter 3

1. Implementation

In this project we aim at building a mechanism that would recognize handwritten digits from the MNIST dataset. We have opted for one-vs-all classification and Neural Networks for the smooth functioning. The goal of this project is to use the basic and simple concepts of this methods and see how good can we make the accuracy rate without using complex techniques such as machine learning.
In this report we are providing source code of our main file that is ‘ex3.m’.This file is blue print of our system. The code in this file has call to four main functions that are responsible for the smooth functioning of the system. Those functions are:-

1. lrCostFunction.m (logistic regression cost function)
2. oneVsAll.m
3. predictOneVsAll.m
4. predict.m

Source Code of File ‘ex3.m’

```
-------------------------------------------------------------------------------------------------------------
%% Machine Learning Author @Shivansh Bhasin


% FUNCTIONS INCLUDED
%
%     lrCostFunction.m (logistic regression cost function)
%     oneVsAll.m
%     predictOneVsAll.m
%     predict.m

clear ; close all; clc

%% Setup the parameters you will use for this part of the exercise
input_layer_size  = 400;  % 20x20 Input Images of Digits
num_labels = 10;          % 10 labels, from 1 to 10
                          % (note that we have mapped "0" to label 10)

fprintf('HANDWRITTEN DIGIT RECOGNITION SYSTEM\n')
fprintf('author@Shivansh_Bhasin\n\n\n')
%% =========== Part 1: Loading and Visualizing Data =============

% Load Training Data
fprintf('Loading and Visualizing Data ...\n')

load('ex3data1.mat'); % training data stored in arrays X,
m = size(X, 1);

% Randomly select 100 data points to display
rand_indices = randperm(m);
sel = X(rand_indices(1:100), :);

displayData(sel);

fprintf('Program paused. Press enter to continue.\n');
pause;

%% ============ Part 2a: Vectorize Logistic Regression ============
% Test case for lrCostFunction
fprintf('\nTesting lrCostFunction() with regularization');

theta_t = [-2; -1; 1; 2];
X_t = [ones(5,1) reshape(1:15,5,3)/10];
y_t = ([1;0;1;0;1] >= 0.5);
lambda_t = 3;
[J grad] = lrCostFunction(theta_t, X_t, y_t, lambda_t);

fprintf('\n Calculated Cost: %f\n', J);
% fprintf('Expected cost: 2.534819\n');
fprintf('Gradients:\n');
fprintf(' %f \n', grad);
% fprintf('Expected gradients:\n');
% fprintf(' 0.146561\n -0.548558\n 0.724722\n 1.398003\n');

fprintf('Program paused. Press enter to continue.\n');
pause;
%% ============ Part 2b: One-vs-All Training ============
fprintf('\nTraining One-vs-All Logistic Regression...\n')

lambda = 0.1;
[all_theta] = oneVsAll(X, y, num_labels, lambda);

fprintf('Program paused. Press enter to continue.\n');
pause;


%% ================ Part 3: Predict for One-Vs-All ================

pred = predictOneVsAll(all_theta, X);

fprintf('\nTraining Set Accuracy for for one-vs-all classifier: %f\n', mean(double(pred == y)) *
100);

%-----------------------------
fprintf('\nLoading Saved Neural Network Parameters ...\n')
```

```
% Load the weights into variables Theta1 and Theta2
load('ex3weights.mat');

%% ================= Part 3: Implement Predict =================
%  After training the neural network, we would like to use it to predict
%  the labels. You will now implement the "predict" function to use the
%  neural network to predict the labels of the training set. This lets
%  you compute the training set accuracy.

pred = predict(Theta1, Theta2, X);
fprintf('Predicting Accuracy For the Training Set...\n\n')
fprintf('\nTraining Set Accuracy for Neural Network Prediction is: %f\n', mean(double(pred ==
y)) * 100);

fprintf('Program paused. Press enter to continue.\n');pause;

%  To give you an idea of the network's output, you can also run
%  through the examples one at the a time to see what it is predicting.

%  Randomly permute examples
rp = randperm(m);

for i = 1:m
    % Display
    fprintf('\nDisplaying Example Image\n');
    displayData(X(rp(i), :));

    pred = predict(Theta1, Theta2, X(rp(i),:));
  fprintf('Running HandDigit Recognition System...\n')
   fprintf('\nNeural Network Prediction:The Predicted Digit is  %d (digit %d)\n', pred, mod(pred,
10));

    % Pause with quit option
    s = input('Paused - Press Enter to continue and q to exit:','s');
    if s == 'q'

      fprintf('\nThank You for using HandDigit Recognition System\n\n\n\n\n\n')
      fprintf('Copyright
@ShivanshBhasin@ShivanshBhasin@ShivanshBhasin@ShivanshBhasin@ShivanshBhasin@Sh
ivanshBhasin@ShivanshBhasin')
      fprintf('\n\n\n\nFor Any Suggestions Email me :- shivanshbhasin0@gmail.com')
      fprintf('\n\nNote:-This is Summer Project of Shivansh Bhasin,Enrollment No:-
75515002718,College:-M.S.I.T')
      break
    end
end
```

1.1 Explanation of Different Function

1.1.1 DisplayData function

We have Displayed 2D data in a nice grid of 10 by 10

[h, display_array] = DISPLAYDATA(X, example_width)

Above is the prototype of DisplayData Function. It displays 2D data stored in X in a nice grid. It returns the figure handle h and the displayed array if requested.

1.1.2 lrCostFunction.m (logistic regression cost function)

This is unregularized cost function.So the formulae that was applied to compute the cost is:

J = ((1 / m) * sum(-y'*log(sigmoid(X*theta)) - (1-y)'*log(1 - sigmoid(X*theta)))) + (lambda/(2*m))*sum(theta(2:length(theta)).*theta(2:length(theta))) ;

The prototype of this function is

function [J, grad] = lrCostFunction(theta, X, y, lambda)

Here J = LRCOSTFUNCTION(theta, X, y, lambda) computes the cost of using theta as the parameter for regularized logistic regression and thegradient of the cost w.r.t. to the parameters.

1.1.3 oneVsAll.m

ONEVSALL trains multiple logistic regression classifiers and returns all the classifiers in a matrix all_theta, where the i-th row of all_theta corresponds to the classifier for label i
The protype of the function is

[all_theta] = ONEVSALL(X, y, num_labels, lambda)

It trains num_labels   logistic regression classifiers and returns each of these classifiers in a matrix all_theta, where the i-th row of all_theta corresponds to the classifier for label i.

1.1.4 predictOneVsAll.m

Predict the label for a trained one-vs-all classifier. The labels are in the range 1..K, where K = size(all_theta, 1). The prototype for this function is :

p = PREDICTONEVSALL(all_theta, X)

This will return a vector of predictions for each example in the matrix X. Note that X contains the examples in rows. all_theta is a matrix where the i-th row is a trained logistic regression theta vector for the i-th class. You should set p to a vector of values from 1..K (e.g., p = [1; 3; 1; 2] predicts classes 1, 3, 1, 2 for 4 examples).

1.1.5 predict.m

Predict the label of an input given a trained neural network. The prototype of this function is

p = PREDICT(Theta1, Theta2, X)

This outputs the predicted label of X given the trained weights of a neural network (Theta1, Theta2).

1.1.6 fmincg.m function

This function is basically a standard function that is Copyright (C) 2001 and 2002 by Carl Edward Rasmussen. Date 2002-02-13
(C) Copyright 1999, 2000 & 2001, Carl Edward Rasmussen

We have modified the use of this function in lrcostfunction by applying vectorization.
Description of function:
Minimize a continuous differentialble multivariate function. Starting point is given by "X" (D by 1), and the function named in the string "f", must return a function value and a vector of partial derivatives. The Polack- Ribiere flavour of conjugate gradients is used to compute search directions, and a line search using quadratic and cubic polynomial approximations and the Wolfe-Powell stopping criteria is used together with the slope ratio methodfor guessing initial step sizes. Additionally a bunch of checks are made to make sure that exploration is taking place and that extrapolation will not be unboundedly large. The "length" gives the length of the run: if it is positive, it gives the maximum number of line searches, if negative its absolute gives the maximum allowed number of function evaluations. You can (optionally) give "length" a second component, which will indicate the reduction in function value to be expected in the first line-search (defaults to 1.0). The function returns when either its length is up, or if no further progress can be made (ie, we are at a minimum, or so close that due to numerical problems, we cannot get any closer). If the function terminates within a few iterations, it could be an indication that the function value and derivatives are not consistent (ie, there may be a bug in the implementation of your "f" function). The function returns the found solution "X", a vector of function values "fX" indicating the progress made and "i" the number of iterations (line searches or function evaluations,depending on the sign of "length") used.

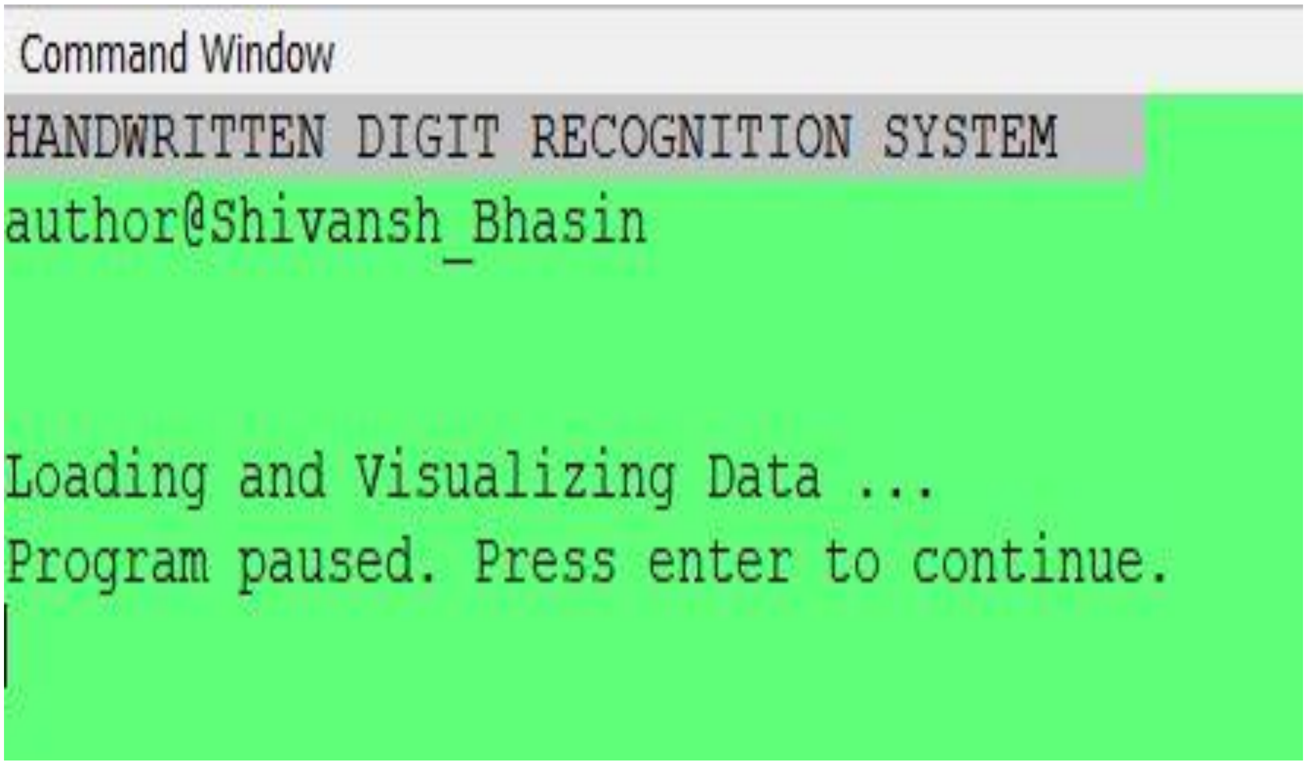
Usage: [X, fX, i] = fmincg(f, X, options, P1, P2, P3, P4, P5)

1.1.7 Sigmoid Function:

It is a mathematical function having a characteristic that can take any real value and map it to between 0 to 1 shaped like the letter “S”. The sigmoid function also called a logistic function.
The Code we have written to compute this is

```
function g = sigmoid(z)
SIGMOID Compute sigmoid functoon
J = SIGMOID(z) computes the sigmoid of z.

g = 1.0 ./ (1.0 + exp(-z));
end
```

Above set of functions are used for the successful implementation of our HandWritten Digit Recognition System. So when we run our main file ‘ex3.m’ we get the following message ,which depicts that our System is successfully executed.



```
HANDWRITTEN DIGIT RECOGNITION SYSTEM
author@Shivansh_Bhasin


Loading and Visualizing Data ...
Program paused. Press enter to continue.
```

Fig 8: Successful Execution of HDRS

Chapter-4

1. Result & Discussion

After successful compilation of all files when we run the file ‘ex3.m’.We get the following series of output.

1.1

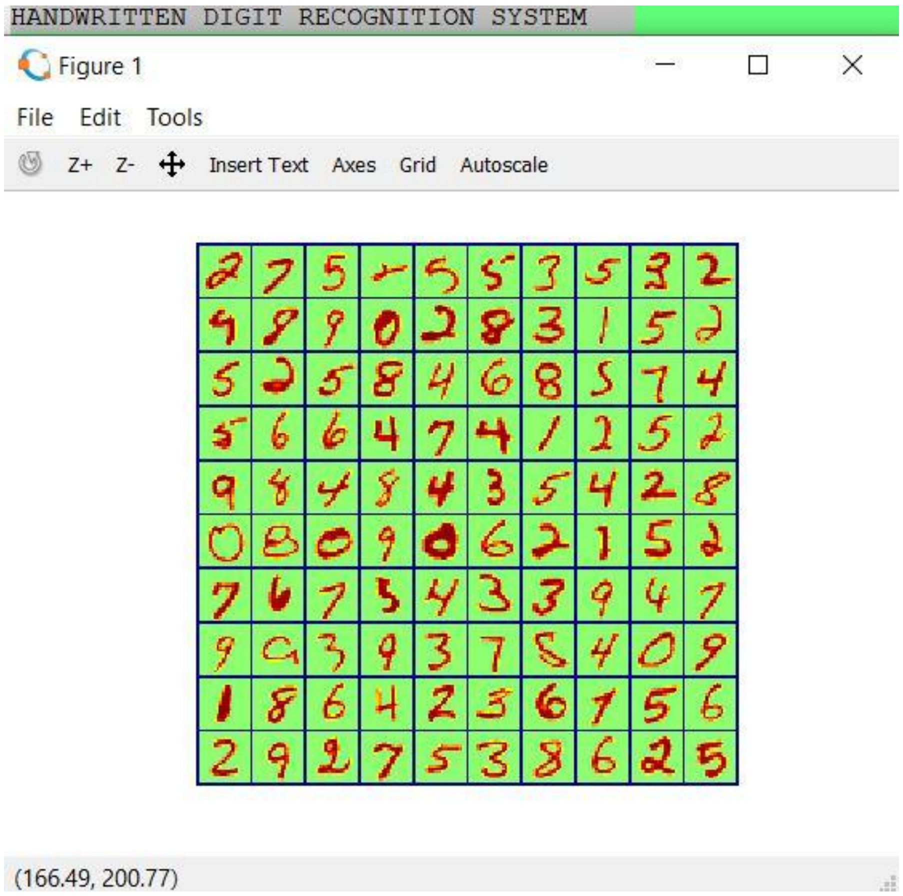
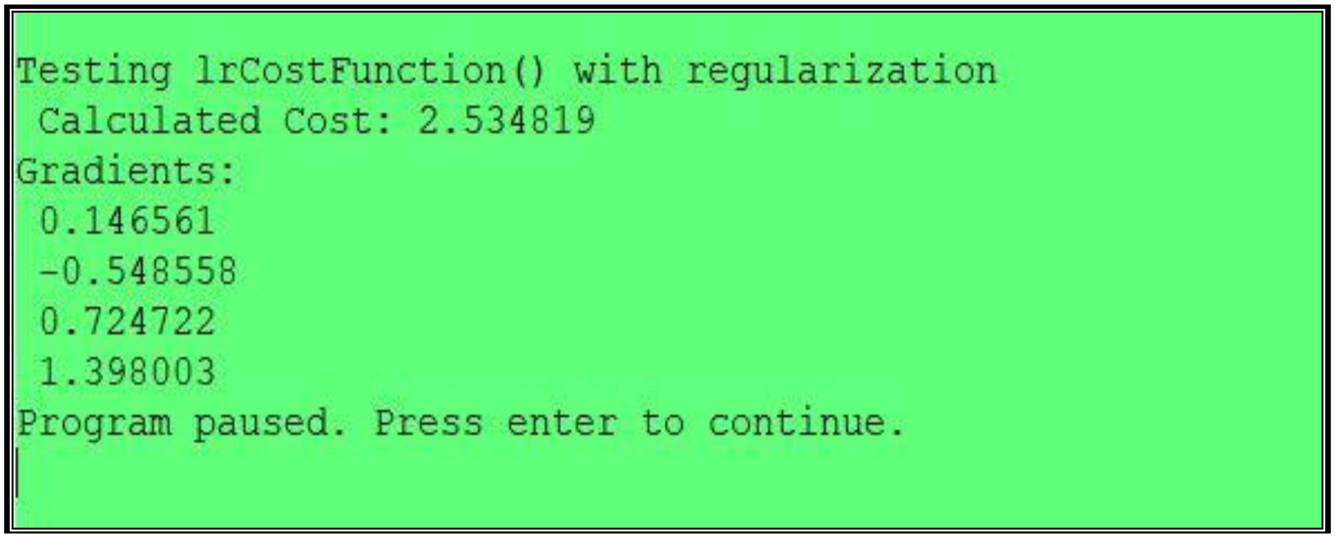


Fig 9 : Examples from the dataset

This is the first output figure that was appeared this contains example of 100 images This image was formed by DisplayData Function.

1.2

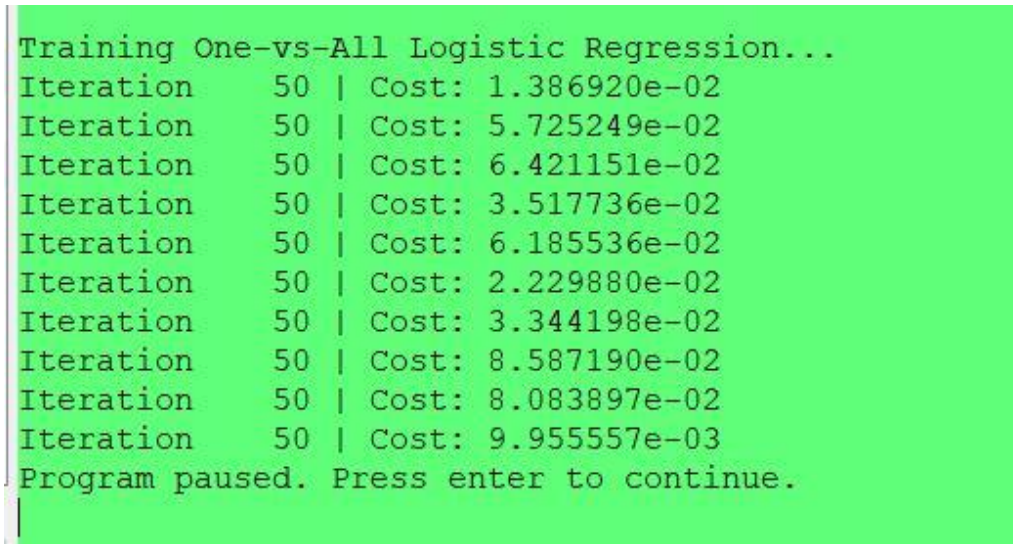


```
Testing lrCostFunction() with regularization
 Calculated Cost: 2.534819
Gradients:
 0.146561
 -0.548558
 0.724722
 1.398003
Program paused. Press enter to continue.
```

Fig 10: Calculated Cost and Gradients

The above image contains Calculated cost and Gradients. This was calculated using lrCostFuction using regularization.

1.3

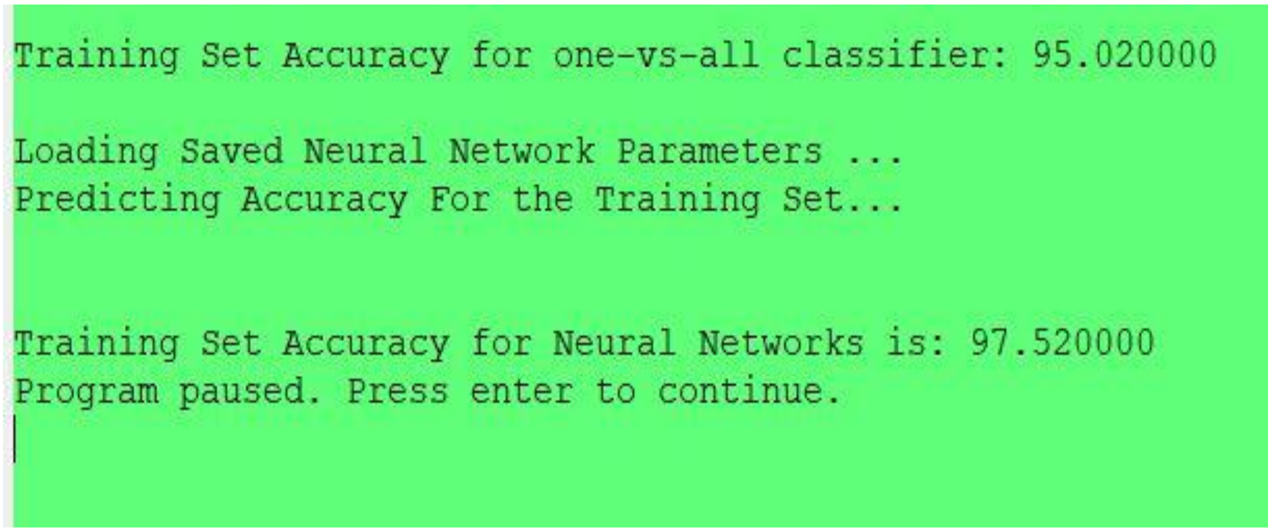


```
Training One-vs-All Logistic Regression...
Iteration    50 | Cost: 1.386920e-02
Iteration    50 | Cost: 5.725249e-02
Iteration    50 | Cost: 6.421151e-02
Iteration    50 | Cost: 3.517736e-02
Iteration    50 | Cost: 6.185536e-02
Iteration    50 | Cost: 2.229880e-02
Iteration    50 | Cost: 3.344198e-02
Iteration    50 | Cost: 8.587190e-02
Iteration    50 | Cost: 8.083897e-02
Iteration    50 | Cost: 9.955557e-03
Program paused. Press enter to continue.
```

Fig 11: Training of One-vs-All Logistic Regression

The above image is basically the training of our one-vs-all Logistic Regression model,for value k=10.Each step has performed 50 iterations and calculated the corresponding cost.
So totally we have performed 500 iterations.

1.4

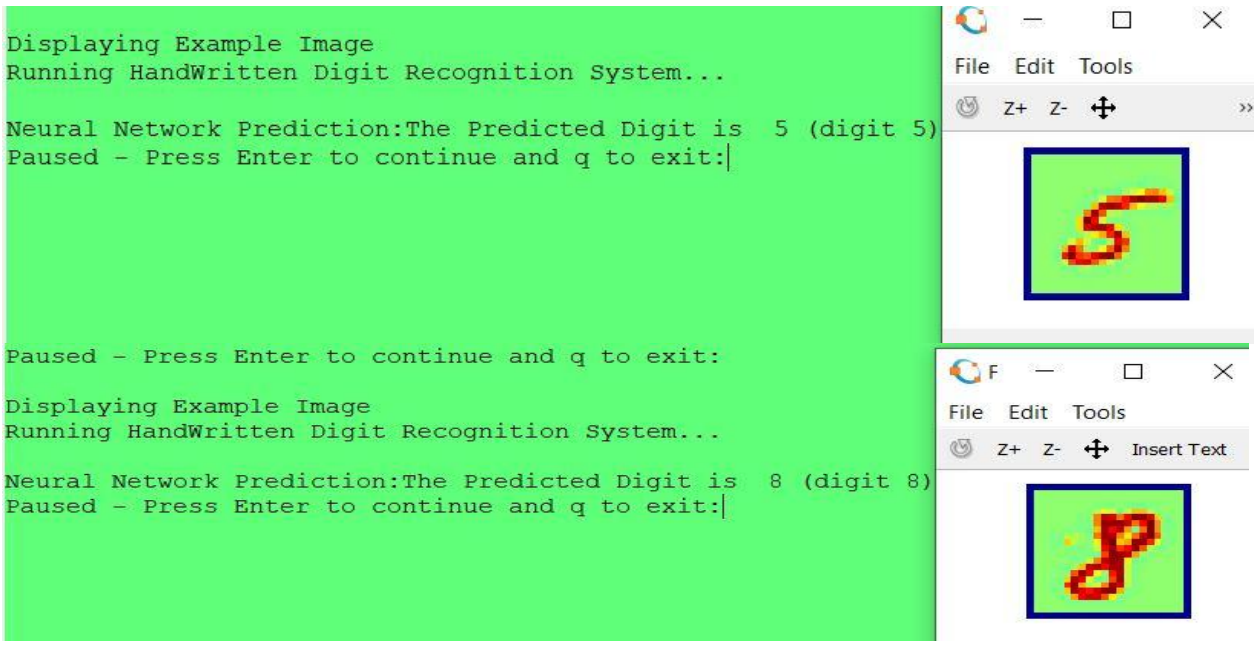


```
Training Set Accuracy for one-vs-all classifier: 95.020000

Loading Saved Neural Network Parameters ...
Predicting Accuracy For the Training Set...


Training Set Accuracy for Neural Networks is: 97.520000
Program paused. Press enter to continue.
```

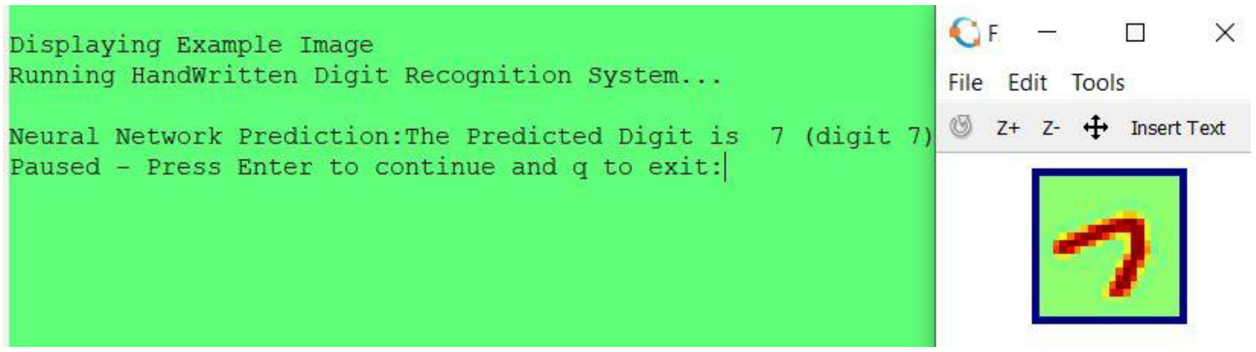
Fig 12: Training Set Accuracy

The above image contains calculated Training set accuracy for both neural networks and one-vs-all classifier.We found that the accuracy for Neural Network is greater than the one-vs-all classifier.

1.5



```
Displaying Example Image
Running HandWritten Digit Recognition System...

Neural Network Prediction:The Predicted Digit is  5 (digit 5)
Paused - Press Enter to continue and q to exit:




Paused - Press Enter to continue and q to exit:

Displaying Example Image
Running HandWritten Digit Recognition System...

Neural Network Prediction:The Predicted Digit is  8 (digit 8)
Paused - Press Enter to continue and q to exit:
```



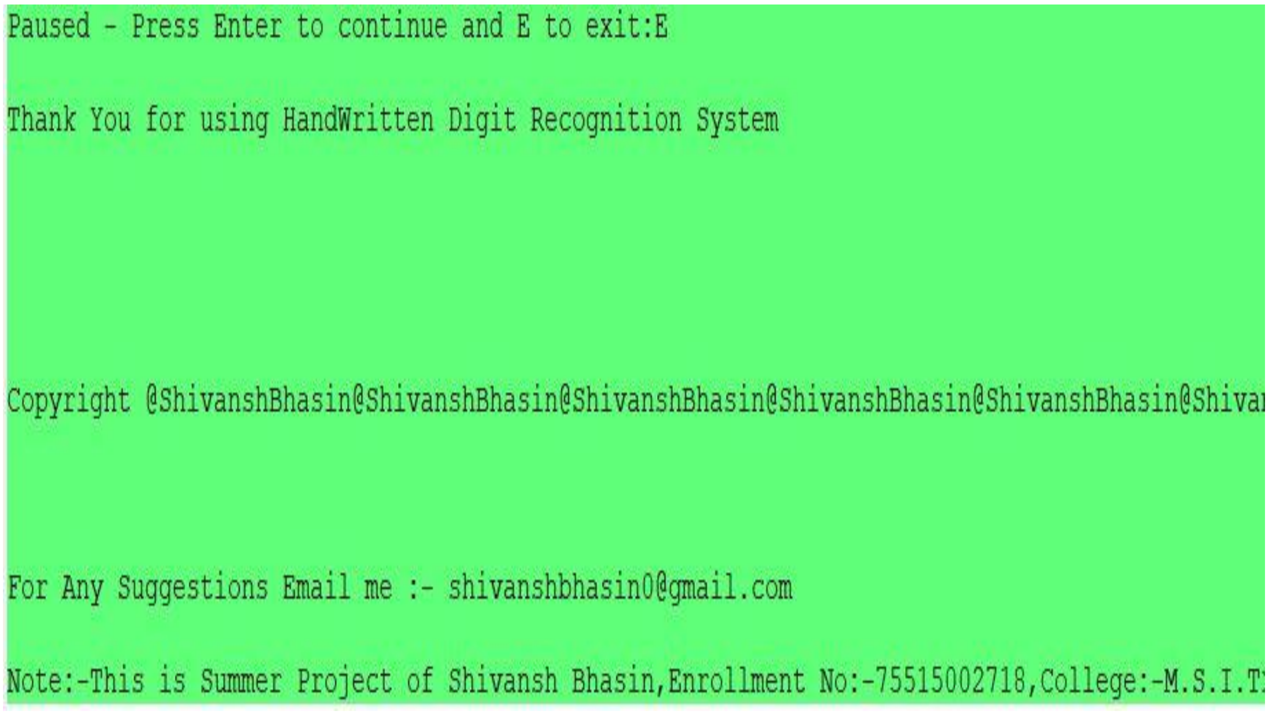
```
Displaying Example Image
Running HandWritten Digit Recognition System...

Neural Network Prediction:The Predicted Digit is  7 (digit 7)
Paused - Press Enter to continue and q to exit:
```

Fig 13: Output of Prediction System

In the above set of images, the prediction system is running. Example image is displaying on right hand side and on left side text we see the recognition of the digit by our handwritten Digit Recognition System.

1.6

In order to terminate the Handwritten Digit Recognition System Press E, and the program terminates and displays following message.



```
Paused - Press Enter to continue and E to exit:E

Thank You for using HandWritten Digit Recognition System



Copyright @ShivanshBhasin@ShivanshBhasin@ShivanshBhasin@ShivanshBhasin@ShivanshBhasin@Shiva


For Any Suggestions Email me :- shivanshbhasin0@gmail.com

Note:-This is Summer Project of Shivansh Bhasin,Enrollment No:-7551500271 8,College:-M.S.I.T
```

Fig 14:-ThankYou Message of our HWDRS

Chapter 5

Future Scope & Conclusion:

Optical Character Recognition is a very broad field concerned with turning an image or a scanned document containing a set of characters into an encoded text that could be read by machines. In this project, we have attempted to build a recognizer for handwritten digits using the MNIST dataset. The challenge of this project was to be able to come up with some basic image correlation techniques, instead of some sophisticated algorithms, and see to what extent we can make this mechanism accurate. We have tried several versions and kept trying to improve each one in order to reach a higher performance rate. The last version has reached a rate of 57% accuracy. Unfortunately, we could not compare the performance of the mechanism we have built to some others that have already been designed and/or implemented before because we did not find any academic paper that tackles this method. The performance we have reached is far less than that of machine learning, which reaches a performance rate of 99.3%; however, it could be further improved and made into a better one. The goal of this project was to explore the field of OCR and try to come up with some techniques that could be used without going into deep computations, and even if the final result is not very reliable, it still provides an accuracy way better than random.

The future steps that to go for would be having a closer look at the results of all the versions in order to find new rules. By extracting and implementing them, we will be able to enhance the performance of these versions. Moreover, it would be good if we could make some modifications to both the reference set and the rules in order to make our program more general and able to identify both typed and handwritten digits.

Furthermore, in the future, we could make a great use of the matrices that indicate the first maximum overlap of each test image with the reference images, along with the number of pixels left out from both. These matrices could be used with some clustering algorithms to build a program able to recognize handwritten digits with a very high efficiency.

Last but not least, we thought about using linear or high level regression in the versions we have developed in order to create more rules. As regression could be used for binary classification and is not very suitable to classify a digit out of ten, this technique could be used in order to tell which digit is the most suitable, the first maximum or second maximum, which will enable us to generate more rules; thus, reach a higher efficiency.

Challenges and Limitations

This Handwritten Digit Recognition System project was my first encounter with Optical Character Recognition (OCR). That is why, while working on this project I was faced with many challenges and issues. First of all, it took me a long while to understand all the concepts and get familiar with them, from image processing to OCR, to all of the techniques and algorithms used in it. Furthermore, the data we were dealing with was very problematic in terms of the way the digits are written. Since some of the digits were rotated by an angle, some of them were thicker or thinner than the rest, and some of the digits were not well centered or were written in confusing ways. In addition to that, understanding neural networks was difficult for me. But with constant efforts I am able to get the concept of neural network. We have tried to maximize the success rate, but still there is a lot of scope for improvement .Our future plan is to improvise this project into larger databases and try to implement this with other complex techniques of Machine Learning and Artificial Intelligence. As in this field of Ml and AI there is always a scope of improvement .

Our Future Plan

Now our next plan is to implement CNN that is Convolution Neural Network in this project, by which this system will become more optimized .Hope we will release version 2 of this project very soon. We have planned to use CNN because Convolutional Neural Networks, CNN has become famous among the recent times. CNN is part of deep, feed forward artificial neural networks that can perform a variety of task with even better time and accuracy than other classifiers, in different applications of image and video recognition, recommender system and natural language processing.

References–

1] MNIST Database of Handwritten digits: http://yann.lecun.com/exdb/mnist/

[2] Claude Chouinard, Rejean Plamondon "Thinning and Segmenting Handwritten Characters by Line Following", Ecole Polytechnique de Montreal, Machine Vision and Applications, 1992

[3] M. Turk and A. Pentland (1991) ``Eigenfaces for recognition", Journal of Cognitive Neuroscience, 3(1)

[4] By Kardi Teknomo, PhD. "K-Nearest Neighbors Tutorial"
http://people.revoledu.com/kardi/tutorial/KNN/index.html

[5] Yann LeCun, Leon Bottou, Yoshua Bengio, Patrick Haffner "Gradient-Based Learning Applied to Document Recognition", IEEE, November 1998

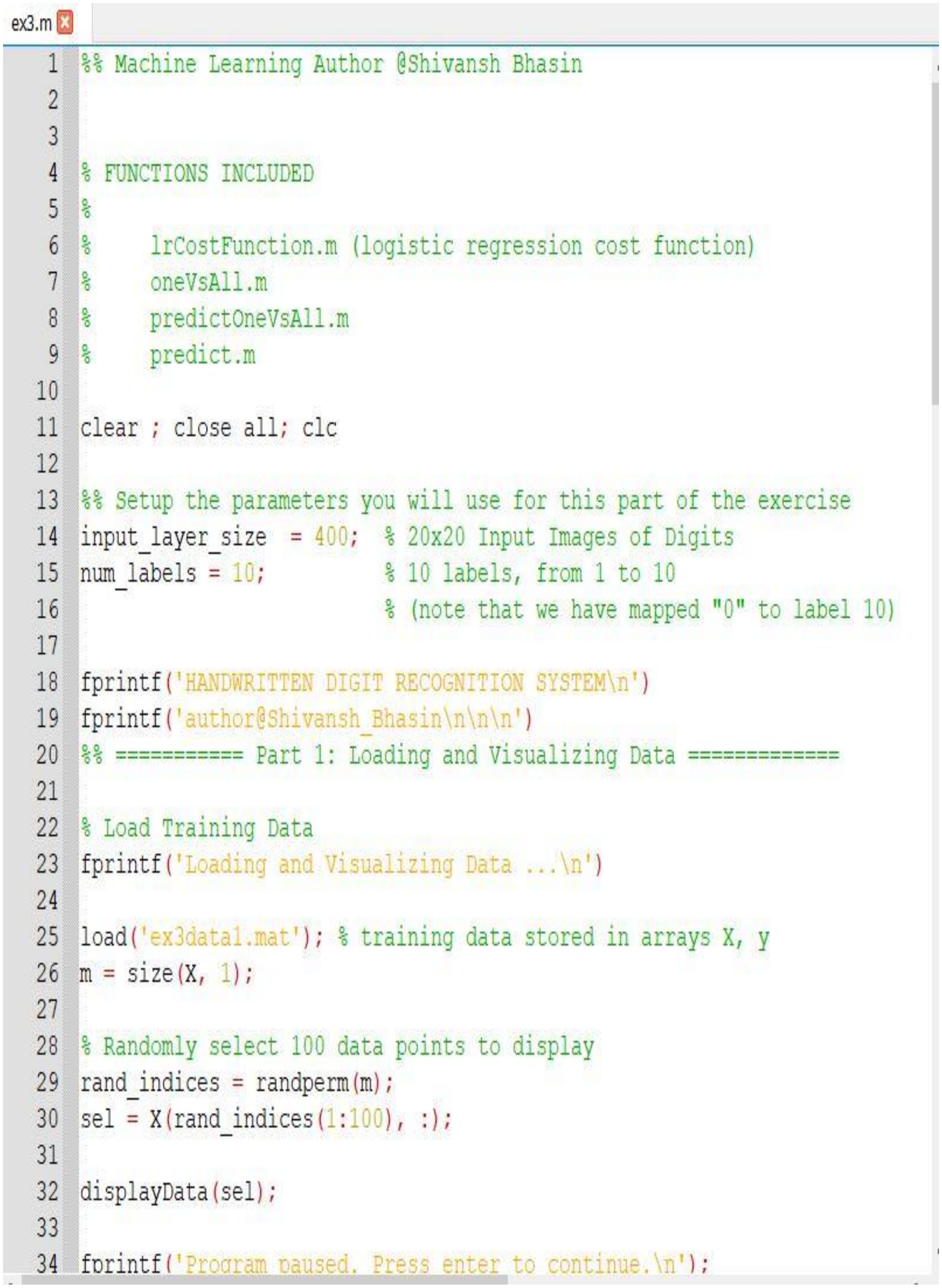
[6] Decong Yu, Lihong Ma and Hanqing Lu "Lottery Digit Recognition Based Multi-feature"

[7] Wen-Li Jiang, Zheng-Xing Sun, Bo Yuan, Wen-Tao Zheng and Wen-Hui Xu "User-Independent Online Handwritten Digit Recognition", Nanjing University, 13-16 August 2006

[8] Geoffrey F Hinton, Michael Revow and Peter Dayan "Recongnizing Handwritten Digits Using Mixtures of Linear Models", http://www.cs.utoronto.ca/~hinton/absps/pancake.pdf

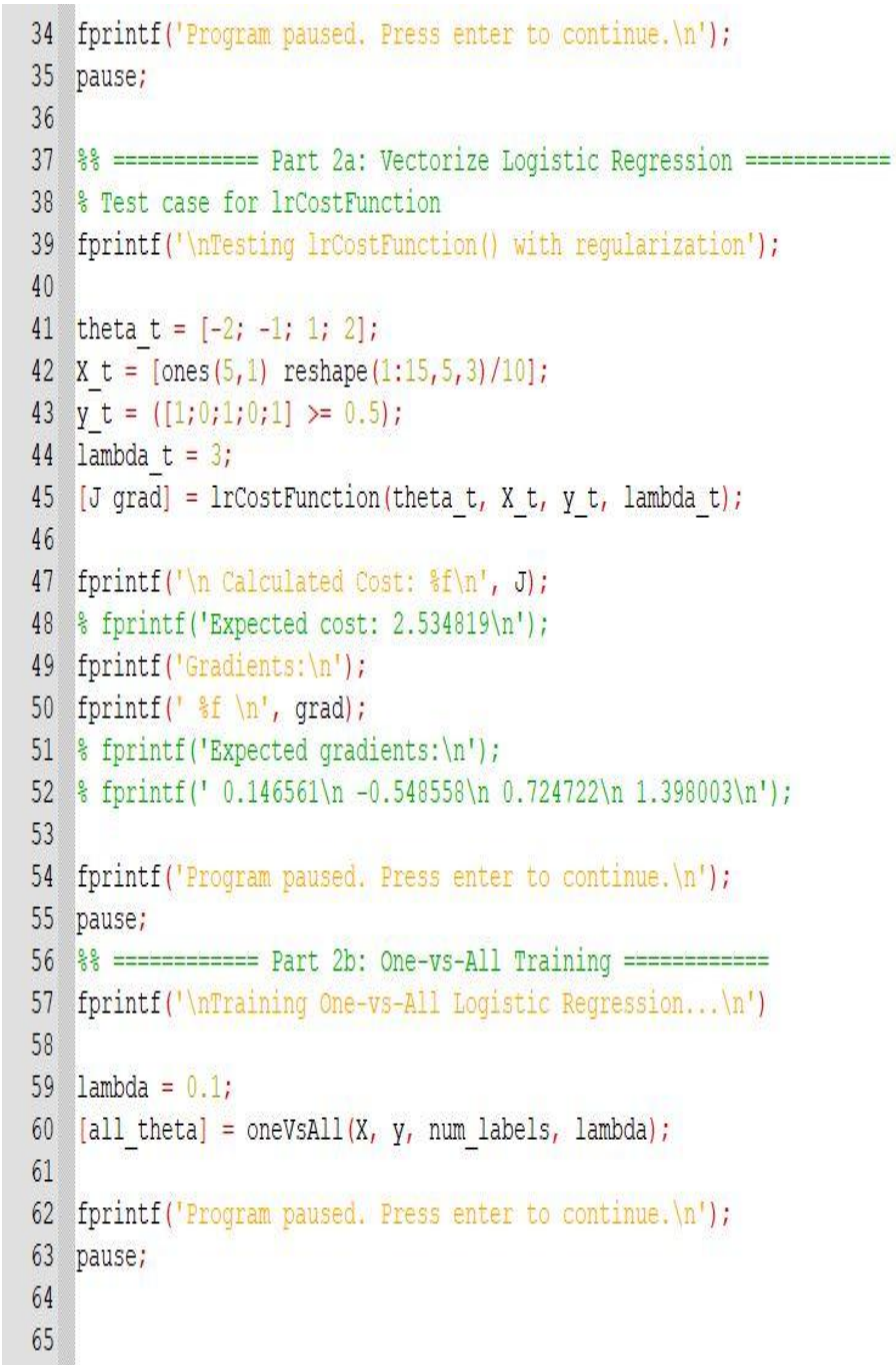
[9] Coursera ML Course Potral ,
https://www.coursera.org/learn/machinelearning/home/welcome

APPENDIX A-SOURCE CODE
(File Name:- ‘ex3.m’)

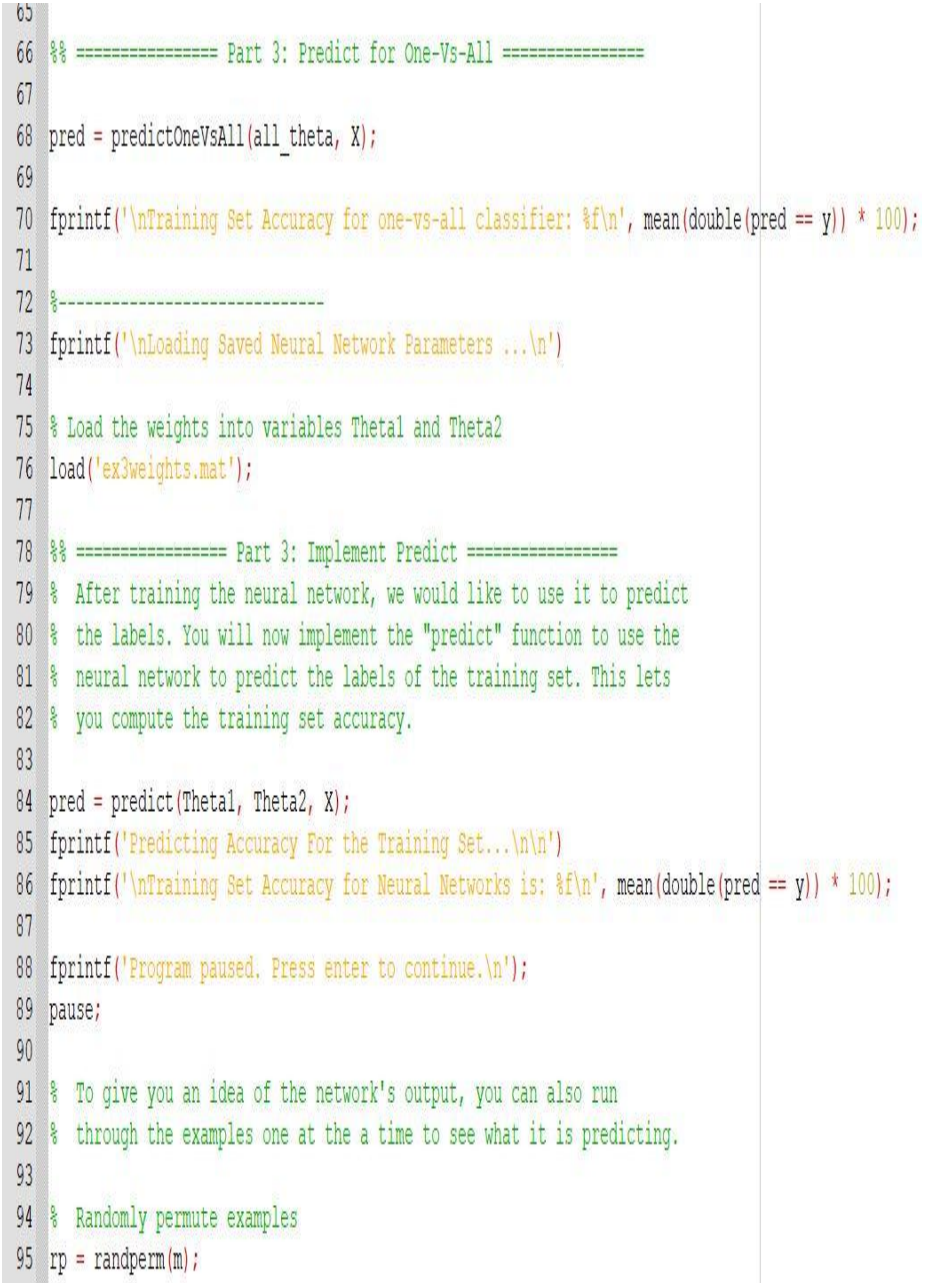


ex3.m

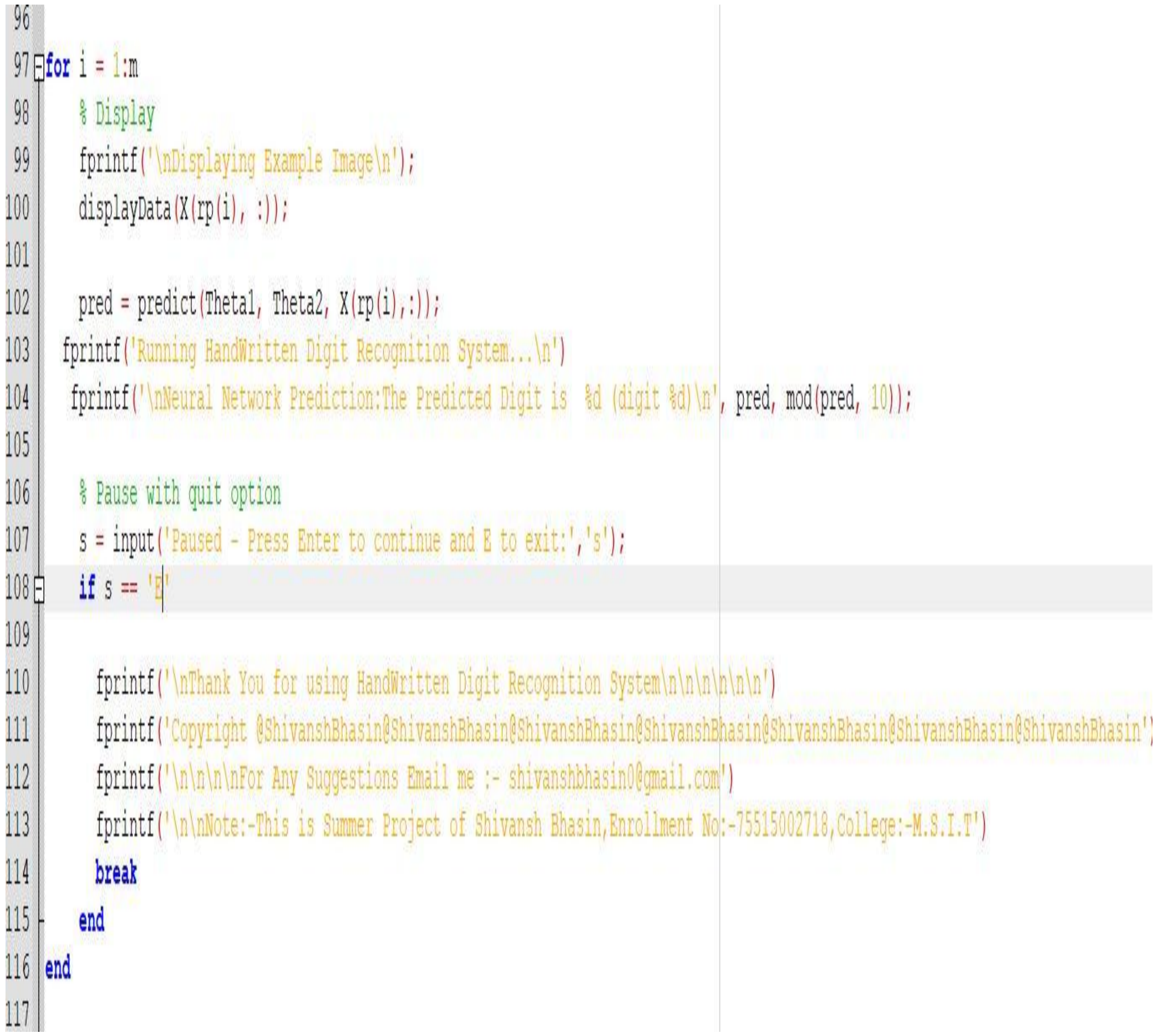
```
%% Machine Learning Author @Shivansh Bhasin


% FUNCTIONS INCLUDED
%
%     lrCostFunction.m (logistic regression cost function)
%     oneVsAll.m
%     predictOneVsAll.m
%     predict.m

clear ; close all; clc

%% Setup the parameters you will use for this part of the exercise
input_layer_size  = 400;  % 20x20 Input Images of Digits
num_labels = 10;          % 10 labels, from 1 to 10
                          % (note that we have mapped "0" to label 10)

fprintf('HANDWRITTEN DIGIT RECOGNITION SYSTEM\n')
fprintf('author@Shivansh_Bhasin\n\n\n')
%% =========== Part 1: Loading and Visualizing Data =============

% Load Training Data
fprintf('Loading and Visualizing Data ...\n')

load('ex3data1.mat'); % training data stored in arrays X, y
m = size(X, 1);

% Randomly select 100 data points to display
rand_indices = randperm(m);
sel = X(rand_indices(1:100), :);

displayData(sel);

fprintf('Program paused. Press enter to continue.\n');
```



```
fprintf('Program paused. Press enter to continue.\n');
pause;

%% ============ Part 2a: Vectorize Logistic Regression ============
% Test case for lrCostFunction
fprintf('\nTesting lrCostFunction() with regularization');

theta_t = [-2; -1; 1; 2];
X_t = [ones(5,1) reshape(1:15,5,3)/10];
y_t = ([1;0;1;0;1] >= 0.5);
lambda_t = 3;
[J grad] = lrCostFunction(theta_t, X_t, y_t, lambda_t);

fprintf('\n Calculated Cost: %f\n', J);
% fprintf('Expected cost: 2.534819\n');
fprintf('Gradients:\n');
fprintf(' %f \n', grad);
% fprintf('Expected gradients:\n');
% fprintf(' 0.146561\n -0.548558\n 0.724722\n 1.398003\n');

fprintf('Program paused. Press enter to continue.\n');
pause;
%% ============ Part 2b: One-vs-All Training ============
fprintf('\nTraining One-vs-All Logistic Regression...\n')

lambda = 0.1;
[all_theta] = oneVsAll(X, y, num_labels, lambda);

fprintf('Program paused. Press enter to continue.\n');
pause;


```

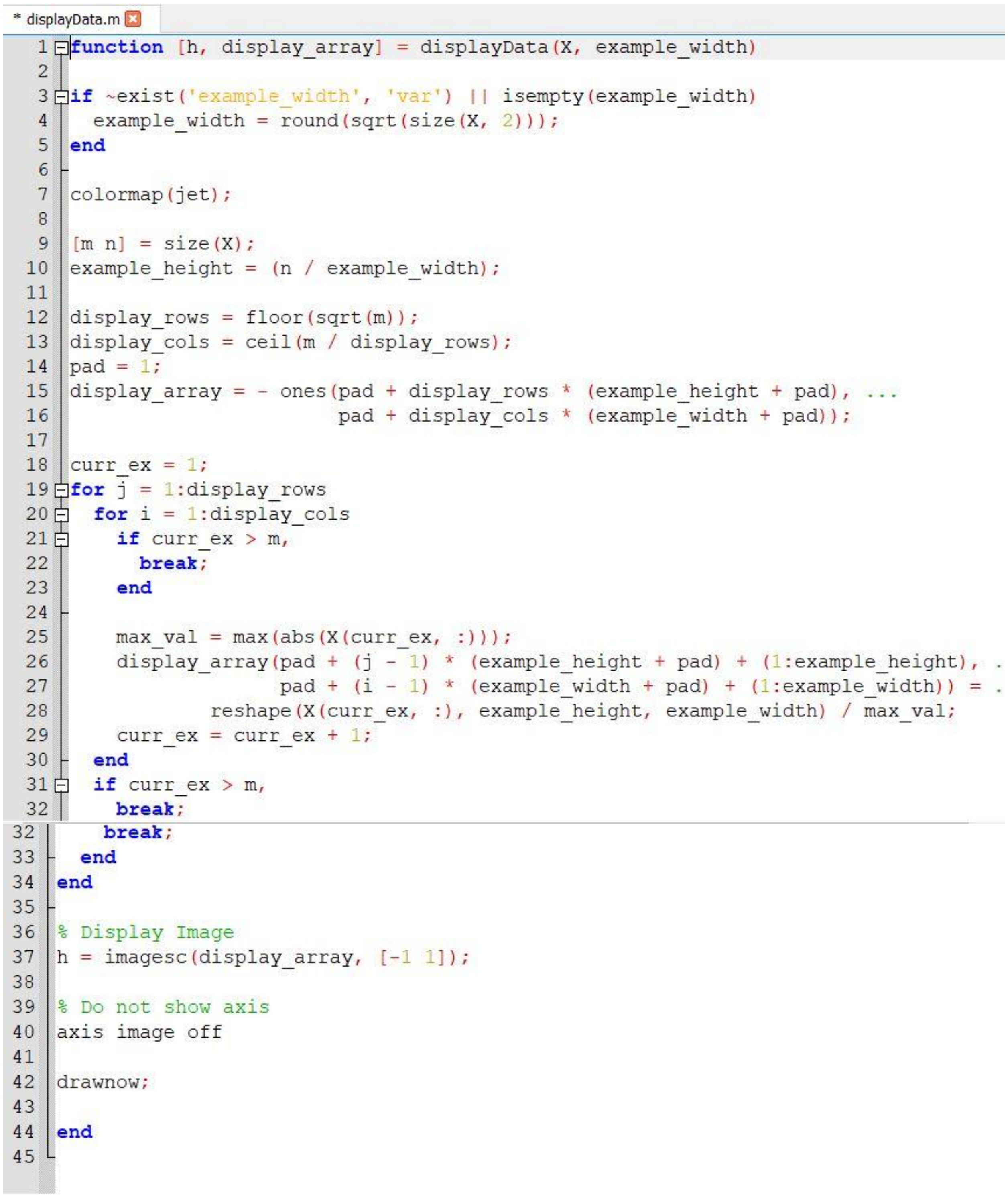


```
%% ================ Part 3: Predict for One-Vs-All ================

pred = predictOneVsAll(all_theta, X);

fprintf('\nTraining Set Accuracy for one-vs-all classifier: %f\n', mean(double(pred == y)) * 100);

%-----------------------------
fprintf('\nLoading Saved Neural Network Parameters ...\n')

% Load the weights into variables Theta1 and Theta2
load('ex3weights.mat');

%% ================= Part 3: Implement Predict =================
%  After training the neural network, we would like to use it to predict
%  the labels. You will now implement the "predict" function to use the
%  neural network to predict the labels of the training set. This lets
%  you compute the training set accuracy.

pred = predict(Theta1, Theta2, X);
fprintf('Predicting Accuracy For the Training Set...\n\n')
fprintf('\nTraining Set Accuracy for Neural Networks is: %f\n', mean(double(pred == y)) * 100);

fprintf('Program paused. Press enter to continue.\n');
pause;

%  To give you an idea of the network's output, you can also run
%  through the examples one at the a time to see what it is predicting.

%  Randomly permute examples
rp = randperm(m);
```



```
for i = 1:m
    % Display
    fprintf('\nDisplaying Example Image\n');
    displayData(X(rp(i), :));

    pred = predict(Theta1, Theta2, X(rp(i),:));
  fprintf('Running HandWritten Digit Recognition System...\n')
   fprintf('\nNeural Network Prediction:The Predicted Digit is  %d (digit %d)\n', pred, mod(pred, 10));

    % Pause with quit option
    s = input('Paused - Press Enter to continue and E to exit:','s');
    if s == 'E'

      fprintf('\nThank You for using HandWritten Digit Recognition System\n\n\n\n\n\n')
      fprintf('Copyright @ShivanshBhasin@ShivanshBhasin@ShivanshBhasin@ShivanshBhasin@ShivanshBhasin@ShivanshBhasin@ShivanshBhasin')
      fprintf('\n\n\n\nFor Any Suggestions Email me :- shivanshbhasin0@gmail.com')
      fprintf('\n\nNote:-This is Summer Project of Shivansh Bhasin,Enrollment No:-75515002718,College:-M.S.I.T')
      break
    end
end
```

APPENDIX B-SOURCE CODE
(File Name:- ‘displaydata.m’)



```
function [h, display_array] = displayData(X, example_width)

if ~exist('example_width', 'var') || isempty(example_width)
  example_width = round(sqrt(size(X, 2)));
end

colormap(jet);

[m n] = size(X);
example_height = (n / example_width);

display_rows = floor(sqrt(m));
display_cols = ceil(m / display_rows);
pad = 1;
display_array = - ones(pad + display_rows * (example_height + pad), ...
                       pad + display_cols * (example_width + pad));

curr_ex = 1;
for j = 1:display_rows
  for i = 1:display_cols
    if curr_ex > m,
      break;
    end

    max_val = max(abs(X(curr_ex, :)));
    display_array(pad + (j - 1) * (example_height + pad) + (1:example_height), .
                  pad + (i - 1) * (example_width + pad) + (1:example_width)) = .
            reshape(X(curr_ex, :), example_height, example_width) / max_val;
    curr_ex = curr_ex + 1;
  end
  if curr_ex > m,
    break;
  end
end

% Display Image
h = imagesc(display_array, [-1 1]);

% Do not show axis
axis image off

drawnow;

end
```