**EX3 – Multi-Class Classification**

input\_layer\_size = 400; % 20x20 Input Images of Digits

num\_labels = 10; % 10 labels, from 1 to 10; 0->10

%% =========== Part 1: Loading and Visualizing Data =============

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

%% ============ Part 2: Vectorize Logistic Regression ============

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: %f\n', mean(double(pred == y)) \* 100);

function g = **sigmoid**(z)

%SIGMOID Compute sigmoid functoon

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

end

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

%LRCOSTFUNCTION Compute cost and gradient for logistic regression with

%regularization

m = length(y); % number of training examples

J = 0;

grad = zeros(size(theta));

h\_theta = sigmoid(X\*theta);

J = -(1./m)\*sum(y.\*log(h\_theta) + (1-y).\*log(1-h\_theta)) + ...

lambda/(2\*m)\*(theta(2:end)'\*theta(2:end));

temp = theta;

temp(1) = 0;

grad = (1./m)\*X'\*(h\_theta-y) + lambda/m \* temp;

grad = grad(:);

end

function [all\_theta] = **oneVsAll**(X, y, num\_labels, lambda)

%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

m = size(X, 1);

n = size(X, 2);

all\_theta = zeros(num\_labels, n + 1);

X = [ones(m, 1) X];

for c=1:num\_labels

initial\_theta = zeros(n + 1, 1);

options = optimset('GradObj', 'on', 'MaxIter', 50);

[theta] = ...

fmincg (@(t)(lrCostFunction(t, X, (y == c), lambda)), ...

initial\_theta, options);

all\_theta(c,:) = theta';

end

end

function p = **predictOneVsAll**(all\_theta, X)

%PREDICT Predict the label for a trained one-vs-all classifier. The labels

%are in the range 1..K, where K = size(all\_theta, 1).

m = size(X, 1);

num\_labels = size(all\_theta, 1);

p = zeros(size(X, 1), 1);

X = [ones(m, 1) X];

[high,p] = max(X\*all\_theta',[],2);

end