mat_orig.m

```
%% Initialization
%clear ; close all; clc

%% Setup the parameters you will use for this code
input_layer_size  = 400;  % 20x20 Input Images of Digits
hidden_layer_size = 25;   % 25 hidden units
num_labels = 10;          % 10 labels, from 1 to 10
                          % (note that we have mapped "0" to label 10)

%% =========== Loading and Visualizing Data =============

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

load('ex4data1.mat');
m = size(X, 1);



%% ================ Loading Parameters ================

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

% Load the weights into variables Theta1 and Theta2
load('ex4weights.mat');

% Unroll parameters
nn_params = [Theta1(:) ; Theta2(:)];






%% ================ Initializing Pameters ================


fprintf('\nInitializing Neural Network Parameters ...\n')

initial_Theta1 = randInitializeWeights(input_layer_size, hidden_layer_size);
initial_Theta2 = randInitializeWeights(hidden_layer_size, num_labels);

% Unroll parameters
initial_nn_params = [initial_Theta1(:) ; initial_Theta2(:)];
```

```
%% ==================  Training NN ==================

%
fprintf('\nTraining Neural Network... \n')


options = optimset('MaxIter', 10);


lambda = 1;

% Create "short hand" for the cost function to be minimized
costFunction = @(p) nnCostFunction(p, ...
                                   input_layer_size, ...
                                   hidden_layer_size, ...
                                   num_labels, X, y, lambda);

% Now, costFunction is a function that takes in only one argument (the
% neural network parameters)
[nn_params, cost] = fmincg(costFunction, initial_nn_params, options);

% Obtain Theta1 and Theta2 back from nn_params
Theta1 = reshape(nn_params(1:hidden_layer_size * (input_layer_size + 1)), ...
                 hidden_layer_size, (input_layer_size + 1));

Theta2 = reshape(nn_params((1 + (hidden_layer_size * (input_layer_size + 1))):end), ...
                 num_labels, (hidden_layer_size + 1));

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


%% ================ Visualize Weights ================

%fprintf('\nVisualizing Neural Network... \n')

%displayData(Theta1(:, 2:end));

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

%% ================ Implement Predict ================


pred = predict(Theta1, Theta2, X);

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

%50 iter --- 95.36 %
%20 iter --- 88 %
```

%15 iter --- 84- 86%
%10 iter --- 76-77%