```
clear ; close all; clc
%% Setup the parameters you will use for this exercise
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)

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

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

```
Loading and Visualizing Data ...
```

```

load('ex4data1.mat');

ind=randperm(5000,500);
X_test=X(ind,:);
y_test=y(ind);

X(ind,:)=[];
y(ind)=[];

m = size(X, 1);

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

displayData(X(sel, :));


%% ================ Part 2: Initializing Pameters ================

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

```
Initializing Neural Network Parameters ...
```

```

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(:)];


%% =================== Part 3: Training NN ===================

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

```
Training Neural Network...
```

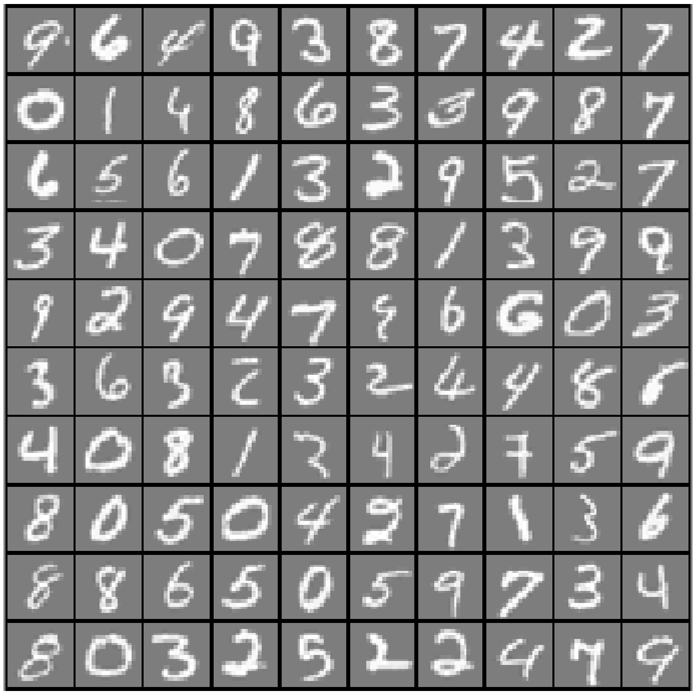
```
options = optimset('MaxIter', 50);

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);
```

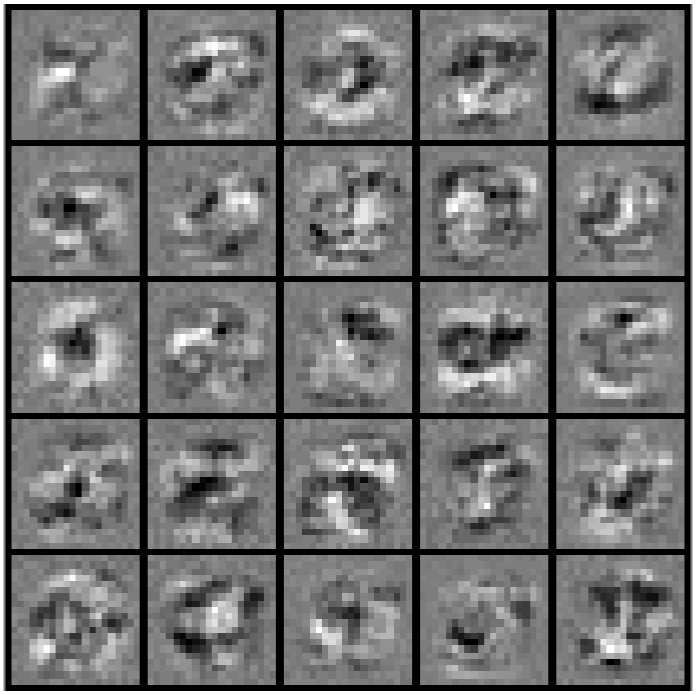
```
Iteration     1 | Cost: 3.294229e+00
Iteration     2 | Cost: 3.242612e+00
Iteration     3 | Cost: 3.207131e+00
Iteration     4 | Cost: 3.132181e+00
Iteration     5 | Cost: 2.500892e+00
Iteration     6 | Cost: 2.223493e+00
Iteration     7 | Cost: 1.951644e+00
Iteration     8 | Cost: 1.702434e+00
Iteration     9 | Cost: 1.581736e+00
Iteration    10 | Cost: 1.461215e+00
Iteration    11 | Cost: 1.303768e+00
Iteration    12 | Cost: 1.238879e+00
Iteration    13 | Cost: 1.144979e+00
Iteration    14 | Cost: 1.059829e+00
Iteration    15 | Cost: 1.019579e+00
Iteration    16 | Cost: 1.004322e+00
Iteration    17 | Cost: 8.928782e-01
Iteration    18 | Cost: 8.561143e-01
Iteration    19 | Cost: 8.303510e-01
Iteration    20 | Cost: 8.025858e-01
Iteration    21 | Cost: 7.606790e-01
Iteration    22 | Cost: 7.404188e-01
Iteration    23 | Cost: 7.200020e-01
Iteration    24 | Cost: 7.071093e-01
Iteration    25 | Cost: 6.912218e-01
Iteration    26 | Cost: 6.600585e-01
Iteration    27 | Cost: 6.340816e-01
Iteration    28 | Cost: 6.184044e-01
Iteration    29 | Cost: 6.065556e-01
Iteration    30 | Cost: 5.760490e-01
Iteration    31 | Cost: 5.672711e-01
Iteration    32 | Cost: 5.581652e-01
Iteration    33 | Cost: 5.535956e-01
Iteration    34 | Cost: 5.506406e-01
Iteration    35 | Cost: 5.385262e-01
Iteration    36 | Cost: 5.341459e-01
Iteration    37 | Cost: 5.314589e-01
Iteration    38 | Cost: 5.284160e-01
Iteration    39 | Cost: 5.254961e-01
Iteration    40 | Cost: 5.139658e-01
Iteration    41 | Cost: 5.045724e-01
Iteration    42 | Cost: 4.961391e-01
Iteration    43 | Cost: 4.853823e-01
Iteration    44 | Cost: 4.780329e-01
Iteration    45 | Cost: 4.735209e-01
Iteration    46 | Cost: 4.689992e-01
Iteration    47 | Cost: 4.653184e-01
Iteration    48 | Cost: 4.628102e-01
Iteration    49 | Cost: 4.566603e-01
Iteration    50 | Cost: 4.507753e-01
```

```
% 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));


%% ================= Part 4: Visualize Weights =================

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

```
Visualizing Neural Network...
```

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





```
%% ================= Part 5: Implement Predict =================

pred = predict(Theta1, Theta2, X);

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

```
Training Set Accuracy: 96.377778
```

```
pred = predict(Theta1, Theta2, X_test);

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

```
Training Set Accuracy: 92.000000
```