lrCostFunction

```
function [J, grad] = lrCostFunction(theta, X, y, lambda)
m = length(y); % number of training examples

htheta=sigmoid(X*theta);
J=(-y'*log(htheta)-(1-y)'*log(1-htheta))/m + lambda/2/m*...
   (theta'*theta-theta(1)^2);
%Old version
% d = size(theta,1);
% grad(1) = ones(1,m)*((sigmoid(X*theta)-y).*X(:,1))/m;
% for j=2:d
%     grad(j) = ones(1,m)*((sigmoid(X*theta)-y).*X(:,j))/m + lambda/m*theta(j);
% end

%New version: Utilize the idea of X'
temp = theta;
temp(1) = 0;
grad = 1/m*X'*(sigmoid(X*theta)-y)+temp*lambda/m;
grad = grad(:);
```

sigmoid

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

oneVsAll

```
function [all_theta] = oneVsAll(X, y, num_labels, lambda)
% y ~ m*K  all_theta~K*(d+1)        X~m*d       d=n
m = size(X, 1);
n = size(X, 2);
all_theta = zeros(num_labels, n + 1);
Y=zeros(m,num_labels);
for i=1:num_labels
   Y(:,i) = (y==i);
end
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
```

predictOneVsAll

```
function p = predictOneVsAll(all_theta, X)
m = size(X, 1);
num_labels = size(all_theta, 1);
p = zeros(size(X, 1), 1);
X = [ones(m, 1) X];
htheta_est = sigmoid(all_theta * X');
[r,pp] = max(htheta_est);
p = pp';
```

displayData

```
function [h, display_array] = displayData(X, example_width)
%DISPLAYDATA Display 2D data in a nice grid
%   [h, display_array] = DISPLAYDATA(X, example_width) displays 2D data
%   stored in X in a nice grid. It returns the figure handle h and the
%   displayed array if requested.

% Set example_width automatically if not passed in
if ~exist('example_width', 'var') || isempty(example_width)
        example_width = round(sqrt(size(X, 2)));
end

% Gray Image
colormap(gray);

% Compute rows, cols
[m n] = size(X);
example_height = (n / example_width);

% Compute number of items to display
display_rows = floor(sqrt(m));
display_cols = ceil(m / display_rows);

% Between images padding
pad = 1;

% Setup blank display
display_array = - ones(pad + display_rows * (example_height + pad), ...
                       pad + display_cols * (example_width + pad));

% Copy each example into a patch on the display array
curr_ex = 1;
for j = 1:display_rows
```

```
    for i = 1:display_cols
        if curr_ex > m,
            break;
        end
        % Copy the patch

        % Get the max value of the patch
        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
```

```
function p = predict(Theta1, Theta2, X)
%PREDICT Predict the label of an input given a trained neural network
%   p = PREDICT(Theta1, Theta2, X) outputs the predicted label of X
given the
%   trained weights of a neural network (Theta1, Theta2)

% Useful values
m = size(X, 1);
num_labels = size(Theta2, 1);

% You need to return the following variables correctly
p = zeros(size(X, 1), 1);

% ====================== YOUR CODE HERE
======================
% Instructions: Complete the following code to make predictions using
%               your learned neural network. You should set p to a
%               vector containing labels between 1 to num_labels.
%
% Hint: The max function might come in useful. In particular, the max
%       function can also return the index of the max element, for more
%       information see 'help max'. If your examples are in rows, then, you
%       can use max(A, [], 2) to obtain the max for each row.
%
X=[ones(m,1) X];
z2 = X*Theta1';
a2 = sigmoid(z2);
a2 = [ones(size(a2,1),1) a2];
z3 = a2*Theta2';
a3 = sigmoid(z3);
[ti,p] = max(a3,[],2);
```

```
%
=======================================================
==================


end
```