Machine Learning week 4 programming exercise One vs All and Neural network

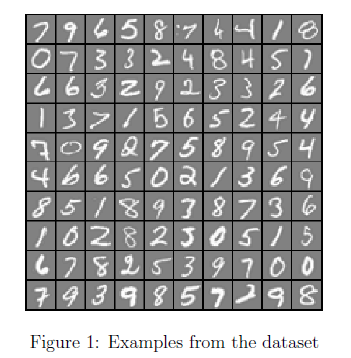
2013年05月22日 23:55:00

**Multi-class Classification**

这一次的任务是识别手写数字0-9，共10个数字，那么这里的分类就会有10种。手上的数据时5000个训练数据，每个数据都是来自于一个20x20像素的图片的每个像素点的灰度信息，构成一个1x400的向量。y就是对应的数字，这里0会被映射成为10.防止0造成octave index的麻烦

**Visualizing the data**

程序会随机选择一些training example 显示出来。如下

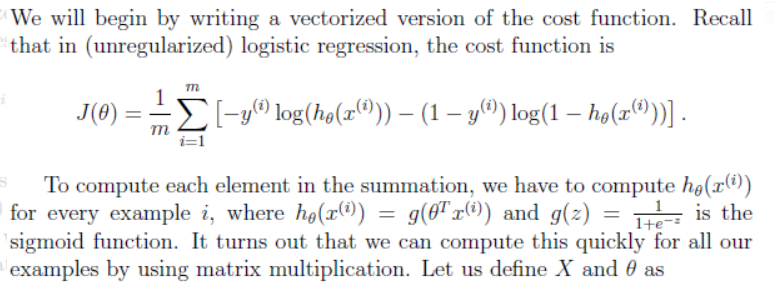


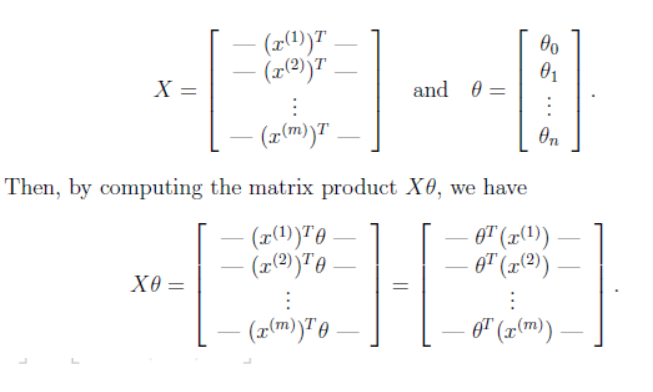
displayData.m

1. function [h, display\_array] = displayData(X, example\_width)
2. %DISPLAYDATA Display 2D data in a nice grid
3. % [h, display\_array] = DISPLAYDATA(X, example\_width) displays 2D data
4. % stored in X in a nice grid. It returns the figure handle h and the
5. % displayed array if requested.
7. % Set example\_width automatically if not passed in
8. if ~exist('example\_width', 'var') || isempty(example\_width)
9. example\_width = round(sqrt(size(X, 2)));
10. end
12. % Gray Image
13. colormap(gray);
15. % Compute rows, cols
16. [m n] = size(X);
17. example\_height = (n / example\_width);
19. % Compute number of items to display
20. display\_rows = floor(sqrt(m));
21. display\_cols = ceil(m / display\_rows);
23. % Between images padding
24. pad = 1;
26. % Setup blank display
27. display\_array = - ones(pad + display\_rows \* (example\_height + pad), ...
28. pad + display\_cols \* (example\_width + pad));
30. % Copy each example into a patch on the display array
31. curr\_ex = 1;
32. for j = 1:display\_rows
33. for i = 1:display\_cols
34. if curr\_ex > m,
35. break;
36. end
37. % Copy the patch
39. % Get the max value of the patch
40. max\_val = max(abs(X(curr\_ex, :)));
41. display\_array(pad + (j - 1) \* (example\_height + pad) + (1:example\_height), ...
42. pad + (i - 1) \* (example\_width + pad) + (1:example\_width)) = ...
43. reshape(X(curr\_ex, :), example\_height, example\_width) / max\_val;
44. curr\_ex = curr\_ex + 1;
45. end
46. if curr\_ex > m,
47. break;
48. end
49. end
51. % Display Image
52. h = imagesc(display\_array, [-1 1]);
54. % Do not show axis
55. axis image off
57. drawnow;
59. end

**Vectorizing Logistic Regression**

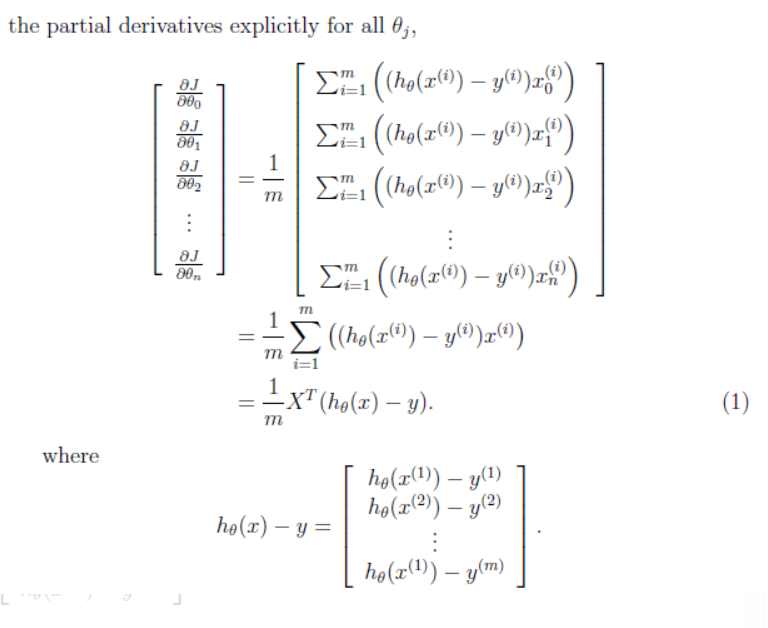
**Cost Function**





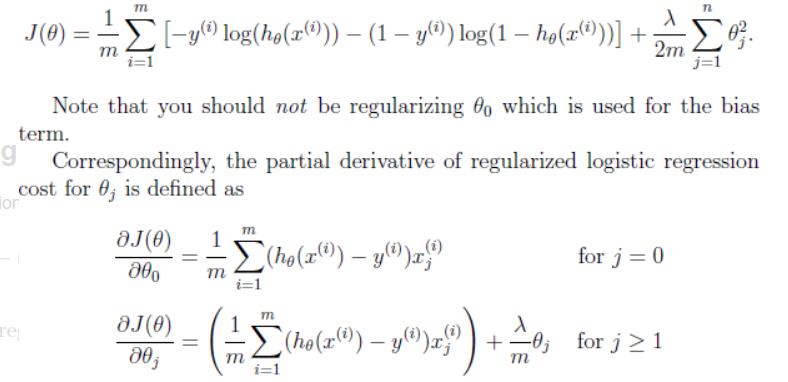
为什么这里costfunction变成了除以m了，以前不都是除以2m的么？？？？

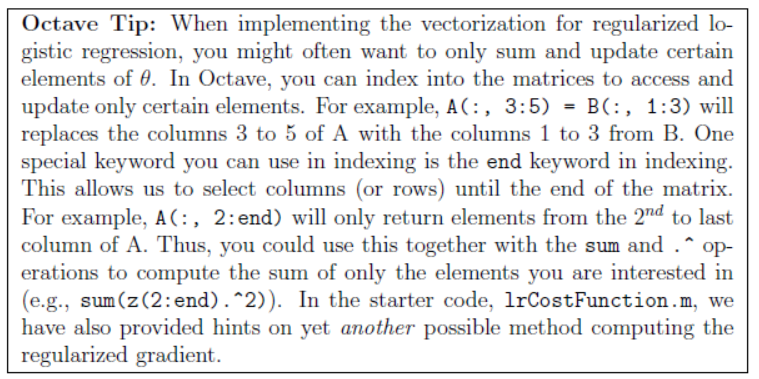
**Gradient**



**Vectorizing regularized logistic regression**

为什么这里用的是theta的平方做regularization？？？？？？





上面的tips在计算regularization项的时候会用到，因为只有theta0会被排除，也就是从2：end都要进行.^2，然后sum up

lrCostFunction.m

1. function [J, grad] = lrCostFunction(theta, X, y, lambda)
2. %LRCOSTFUNCTION Compute cost and gradient for logistic regression with
3. %regularization
4. % J = LRCOSTFUNCTION(theta, X, y, lambda) computes the cost of using
5. % theta as the parameter for regularized logistic regression and the
6. % gradient of the cost w.r.t. to the parameters.
8. % Initialize some useful values
9. m = length(y); % number of training examples
11. % You need to return the following variables correctly
12. J = 0;
13. grad = zeros(size(theta));
15. % ====================== YOUR CODE HERE ======================
16. % Instructions: Compute the cost of a particular choice of theta.
17. % You should set J to the cost.
18. % Compute the partial derivatives and set grad to the partial
19. % derivatives of the cost w.r.t. each parameter in theta
20. %
21. % Hint: The computation of the cost function and gradients can be
22. % efficiently vectorized. For example, consider the computation
23. %
24. % sigmoid(X \* theta)
25. %
26. % Each row of the resulting matrix will contain the value of the
27. % prediction for that example. You can make use of this to vectorize
28. % the cost function and gradient computations.
29. %
30. % Hint: When computing the gradient of the regularized cost function,
31. % there're many possible vectorized solutions, but one solution
32. % looks like:
33. % grad = (unregularized gradient for logistic regression)
34. % temp = theta;
35. % temp(1) = 0; % because we don't add anything for j = 0
36. % grad = grad + YOUR\_CODE\_HERE (using the temp variable)
37. %
38. sigm = sigmoid(X\*theta);
39. J = sum(-y.\*log(sigm) - (1-y).\*log(1-sigm))/m + lambda \* sum(theta(2:end).^2)/(2\*m);
40. grad = X' \* (sigm-y)/m;
41. grad0 = grad(1);
42. grad = grad + (lambda/m)\*theta;
43. grad(1) = grad0;

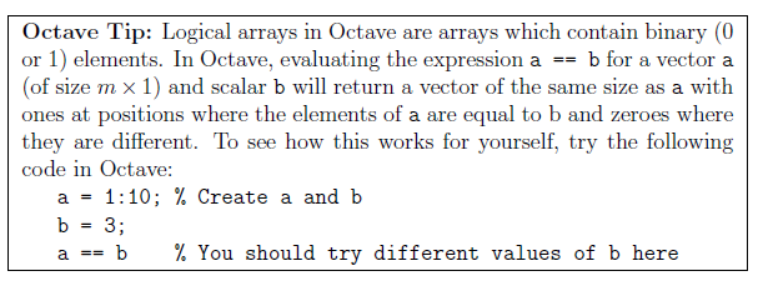




50. % =============================================================
52. grad = grad(:);
54. end

**One-vs-All Classification**

针对每个class分别进行训练，当训练的时候，其他class的y需要被影射成0，只有当前class的y设置为1，下面的tip可以很快的完成这种转化。



oneVsAll.m

1. function [all\_theta] = oneVsAll(X, y, num\_labels, lambda)
2. %ONEVSALL trains multiple logistic regression classifiers and returns all
3. %the classifiers in a matrix all\_theta, where the i-th row of all\_theta
4. %corresponds to the classifier for label i
5. % [all\_theta] = ONEVSALL(X, y, num\_labels, lambda) trains num\_labels
6. % logisitc regression classifiers and returns each of these classifiers
7. % in a matrix all\_theta, where the i-th row of all\_theta corresponds
8. % to the classifier for label i
10. % Some useful variables
11. m = size(X, 1);
12. n = size(X, 2);
14. % You need to return the following variables correctly
15. all\_theta = zeros(num\_labels, n + 1);
17. % Add ones to the X data matrix
18. X = [ones(m, 1) X];
20. % ====================== YOUR CODE HERE ======================
21. % Instructions: You should complete the following code to train num\_labels
22. % logistic regression classifiers with regularization
23. % parameter lambda.
24. %
25. % Hint: theta(:) will return a column vector.
26. %
27. % Hint: You can use y == c to obtain a vector of 1's and 0's that tell use
28. % whether the ground truth is true/false for this class.
29. %
30. % Note: For this assignment, we recommend using fmincg to optimize the cost
31. % function. It is okay to use a for-loop (for c = 1:num\_labels) to
32. % loop over the different classes.
33. %
34. % fmincg works similarly to fminunc, but is more efficient when we
35. % are dealing with large number of parameters.
36. %
37. % Example Code for fmincg:
38. %
39. % % Set Initial theta
40. % initial\_theta = zeros(n + 1, 1);
41. %
42. % % Set options for fminunc
43. % options = optimset('GradObj', 'on', 'MaxIter', 50);
44. %
45. % % Run fmincg to obtain the optimal theta
46. % % This function will return theta and the cost
47. % [theta] = ...
48. % fmincg (@(t)(lrCostFunction(t, X, (y == c), lambda)), ...
49. % initial\_theta, options);
50. %
51. for k=1:num\_labels
52. initial\_theta = zeros(n + 1, 1);
53. options = optimset('GradObj', 'on', 'MaxIter', 50);
54. [theta] = fmincg (@(t)(lrCostFunction(t, X, (y == k), lambda)),initial\_theta, options);
55. all\_theta(k,:) = theta';
56. end








66. % =========================================================================

69. end

Predict One-vs-All

注意下面max的用法，直接返回了每行最大值所在的index，而这个index其实就是我们预测的值。

1. function p = predictOneVsAll(all\_theta, X)
2. %PREDICT Predict the label for a trained one-vs-all classifier. The labels
3. %are in the range 1..K, where K = size(all\_theta, 1).
4. % p = PREDICTONEVSALL(all\_theta, X) will return a vector of predictions
5. % for each example in the matrix X. Note that X contains the examples in
6. % rows. all\_theta is a matrix where the i-th row is a trained logistic
7. % regression theta vector for the i-th class. You should set p to a vector
8. % of values from 1..K (e.g., p = [1; 3; 1; 2] predicts classes 1, 3, 1, 2
9. % for 4 examples)
11. m = size(X, 1);
12. num\_labels = size(all\_theta, 1);
14. % You need to return the following variables correctly
15. p = zeros(size(X, 1), 1);
17. % Add ones to the X data matrix
18. X = [ones(m, 1) X];
20. % ====================== YOUR CODE HERE ======================
21. % Instructions: Complete the following code to make predictions using
22. % your learned logistic regression parameters (one-vs-all).
23. % You should set p to a vector of predictions (from 1 to
24. % num\_labels).
25. %
26. % Hint: This code can be done all vectorized using the max function.
27. % In particular, the max function can also return the index of the
28. % max element, for more information see 'help max'. If your examples
29. % are in rows, then, you can use max(A, [], 2) to obtain the max
30. % for each row.
31. %
32. [c,i] = max(sigmoid(X \* all\_theta'), [], 2);
33. p = i;

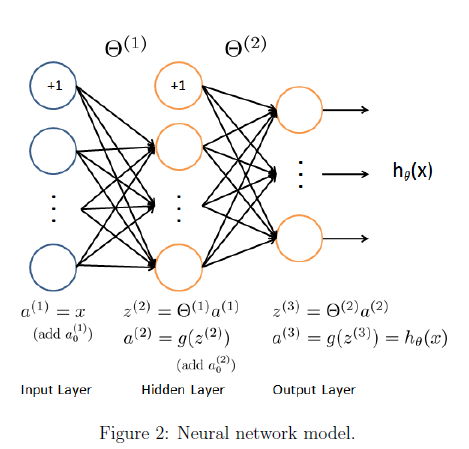


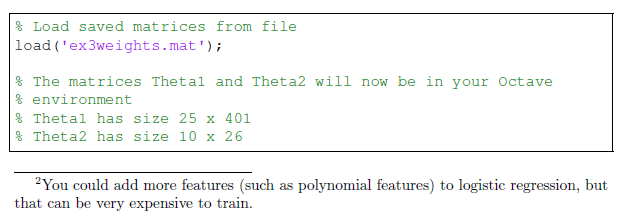
38. % =========================================================================

41. end

**Neural Network**

**Model Representation**





**Feedforward Propagation and Prediction**

1. function p = predict(Theta1, Theta2, X)
2. %PREDICT Predict the label of an input given a trained neural network
3. % p = PREDICT(Theta1, Theta2, X) outputs the predicted label of X given the
4. % trained weights of a neural network (Theta1, Theta2)
6. % Useful values
7. m = size(X, 1);
8. num\_labels = size(Theta2, 1);
10. % You need to return the following variables correctly
11. p = zeros(size(X, 1), 1);
13. % ====================== YOUR CODE HERE ======================
14. % Instructions: Complete the following code to make predictions using
15. % your learned neural network. You should set p to a
16. % vector containing labels between 1 to num\_labels.
17. %
18. % Hint: The max function might come in useful. In particular, the max
19. % function can also return the index of the max element, for more
20. % information see 'help max'. If your examples are in rows, then, you
21. % can use max(A, [], 2) to obtain the max for each row.
22. %
24. X = [ones(m, 1) X];
25. z2 = Theta1 \* X';
26. a2 = sigmoid(z2);
27. a2 = [ones(1, m);a2];
28. z3 = Theta2 \* a2;
29. a3 = sigmoid(z3);
30. output =a3';
31. [c,i] = max(output, [], 2);
32. p = i;
34. % =========================================================================

37. end
38. %% Machine Learning Online Class - Exercise 3 | Part 1: One-vs-all
40. % Instructions
41. % ------------
42. %
43. % This file contains code that helps you get started on the
44. % linear exercise. You will need to complete the following functions
45. % in this exericse:
46. %
47. % lrCostFunction.m (logistic regression cost function)
48. % oneVsAll.m
49. % predictOneVsAll.m
50. % predict.m
51. %
52. % For this exercise, you will not need to change any code in this file,
53. % or any other files other than those mentioned above.
54. %
56. %% Initialization
57. clear ; close all; clc
59. %% Setup the parameters you will use for this part of the exercise
60. input\_layer\_size = 400; % 20x20 Input Images of Digits
61. num\_labels = 10; % 10 labels, from 1 to 10
62. % (note that we have mapped "0" to label 10)
64. %% =========== Part 1: Loading and Visualizing Data =============
65. % We start the exercise by first loading and visualizing the dataset.
66. % You will be working with a dataset that contains handwritten digits.
67. %
69. % Load Training Data
70. fprintf('Loading and Visualizing Data ...\n')
72. load('ex3data1.mat'); % training data stored in arrays X, y
73. m = size(X, 1);
75. % Randomly select 100 data points to display
76. rand\_indices = randperm(m);
77. sel = X(rand\_indices(1:100), :);
79. displayData(sel);
81. fprintf('Program paused. Press enter to continue.\n');
82. pause;
84. %% ============ Part 2: Vectorize Logistic Regression ============
85. % In this part of the exercise, you will reuse your logistic regression
86. % code from the last exercise. You task here is to make sure that your
87. % regularized logistic regression implementation is vectorized. After
88. % that, you will implement one-vs-all classification for the handwritten
89. % digit dataset.
90. %
92. fprintf('\nTraining One-vs-All Logistic Regression...\n')
94. lambda = 0.1;
95. [all\_theta] = oneVsAll(X, y, num\_labels, lambda);
97. fprintf('Program paused. Press enter to continue.\n');
98. pause;

101. %% ================ Part 3: Predict for One-Vs-All ================
102. % After ...
103. pred = predictOneVsAll(all\_theta, X);
105. fprintf('\nTraining Set Accuracy: %f\n', mean(double(pred == y)) \* 100);
106. %% Machine Learning Online Class - Exercise 3 | Part 2: Neural Networks
108. % Instructions
109. % ------------
110. %
111. % This file contains code that helps you get started on the
112. % linear exercise. You will need to complete the following functions
113. % in this exericse:
114. %
115. % lrCostFunction.m (logistic regression cost function)
116. % oneVsAll.m
117. % predictOneVsAll.m
118. % predict.m
119. %
120. % For this exercise, you will not need to change any code in this file,
121. % or any other files other than those mentioned above.
122. %
124. %% Initialization
125. clear ; close all; clc
127. %% Setup the parameters you will use for this exercise
128. input\_layer\_size = 400; % 20x20 Input Images of Digits
129. hidden\_layer\_size = 25; % 25 hidden units
130. num\_labels = 10; % 10 labels, from 1 to 10
131. % (note that we have mapped "0" to label 10)
133. %% =========== Part 1: Loading and Visualizing Data =============
134. % We start the exercise by first loading and visualizing the dataset.
135. % You will be working with a dataset that contains handwritten digits.
136. %
138. % Load Training Data
139. fprintf('Loading and Visualizing Data ...\n')
141. load('ex3data1.mat');
142. m = size(X, 1);
144. % Randomly select 100 data points to display
145. sel = randperm(size(X, 1));
146. sel = sel(1:100);
148. displayData(X(sel, :));
150. fprintf('Program paused. Press enter to continue.\n');
151. pause;
153. %% ================ Part 2: Loading Pameters ================
154. % In this part of the exercise, we load some pre-initialized
155. % neural network parameters.
157. fprintf('\nLoading Saved Neural Network Parameters ...\n')
159. % Load the weights into variables Theta1 and Theta2
160. load('ex3weights.mat');
162. %% ================= Part 3: Implement Predict =================
163. % After training the neural network, we would like to use it to predict
164. % the labels. You will now implement the "predict" function to use the
165. % neural network to predict the labels of the training set. This lets
166. % you compute the training set accuracy.
168. pred = predict(Theta1, Theta2, X);
170. fprintf('\nTraining Set Accuracy: %f\n', mean(double(pred == y)) \* 100);
172. fprintf('Program paused. Press enter to continue.\n');
173. pause;
175. % To give you an idea of the network's output, you can also run
176. % through the examples one at the a time to see what it is predicting.
178. % Randomly permute examples
179. rp = randperm(m);
181. for i = 1:m
182. % Display
183. fprintf('\nDisplaying Example Image\n');
184. displayData(X(rp(i), :));
186. pred = predict(Theta1, Theta2, X(rp(i),:));
187. fprintf('\nNeural Network Prediction: %d (digit %d)\n', pred, mod(pred, 10));
189. % Pause
190. fprintf('Program paused. Press enter to continue.\n');
191. pause;
192. end