

Ex 3 Tutorial

ex3 oneVsAll()

Tom MosherMentorAssignment: Multi-class Classification and Neural Networks · 2 years ago ·

Edited

all_theta is a matrix, where there is a row for each of the trained thetas. In the exercise example, there are 10 rows, of 401 elements each. You know this because that's how all_theta was initialized in line 15 of the script template. (note that the submit grader's test case doesn't have 401 elements or 10 rows - your function must work for any size data set - so use the "num_labels" variable).

Each call to fmincg() returns a theta vector. Be sure you use the lambda value provided in the function header.

You then need to copy that vector into a row of all_theta.

The oneVsAll.m script template contains several Hints and a code example to guide your work.

Type these commands in your workspace to see how to copy a vector into a matrix:

1
2
3

```
Q = zeros(5,3)    % create a test matrix of all-zeros
v = [1 2 3]'      % create a column vector
Q(2,:) = v        % copy v into the 2nd row of Q
```

XX

The syntax "(2,:)" means "use all columns of the 2nd row".

=====

keywords: tutorial onevsall

#####

ex3: tutorial for predictOneVsAll()

Tom MosherMentorWeek 4 · 2 years ago · Edited

The code you add to predictOneVsAll.m can be as little as two lines:

1. one line to calculate the sigmoid() of the product of X and all_theta. X is (m x n), and all_theta is (num_labels x n), so you'll need a transposition to get a result of (m x num_labels)

2. one line to return the classifier which has the max value. The size will be (m x 1). Use the "help max" command in your workspace to learn how the max() function returns two values.

Note that your function must return the predictions as a column vector - size (m x 1). If you return a row vector, the script will not compute the accuracy correctly.

#####

ex3 tutorial for predict()

Tom MosherMentorWeek 4 · 2 years ago · Edited

Here is an outline for forward propagation using the vectorized method. This is an implementation of the formula in Figure 2 on Page 11 of ex3.pdf.

1. Add a column of 1's to X (the first column), and it becomes 'a1'.
2. Multiply by Theta1 and you have 'z2'.
3. Compute the sigmoid() of 'z2', then add a column of 1's, and it becomes 'a2'.
4. Multiply by Theta2, compute the sigmoid() and it becomes 'a3'.
5. Now use the max(a3, [], 2) function to return two vectors - one of the highest value for each row, and one with its index. Ignore the highest values. Keep the vector of the indexes where the highest values were found. These are your predictions.

Note: When you multiply by the Theta matrices, you'll have to use transposition to get a result that is the correct size.

Note: The predictions must be returned as a column vector - size (m x 1). If you return a row vector, the script will not compute the accuracy correctly.

Note: Not getting the correct results? In the hidden layer, be sure you use sigmoid() first, then add the bias unit.

----- dimensions of the variables -----

a1 is (m x n), where 'n' is the number of features including the bias unit

Theta1 is (h x n) where 'h' is the number of hidden units

a2 is (m x (h + 1))

Theta2 is (c x (h + 1)), where 'c' is the number of labels.

a3 is (m x c)

p is a vector of size (m x 1)

#####