



Week 5

Discuss this week's module: Neural Networks: Learning.

← Week 5



Test case for ex4 nnCostFunction()



Tom Mosher Mentor Week 5 · 9 months ago · Edited

Here is a test case for the nnCostFunction() with (and without) regularization:

Enter these values in your console workspace, compare your results with those given.

Test Case with regularization:

```
1 il = 2;           % input layer
2 hl = 2;           % hidden layer
3 nl = 4;           % number of labels
4 nn = [ 1:18 ] / 10; % nn_params
5 X = cos([1 2 ; 3 4 ; 5 6]);
6 y = [4; 2; 3];
7 lambda = 4;
8 [J grad] = nnCostFunction(nn, il, hl, nl, X, y, lambda)
```

output:

```
1 J = 19.474
2 grad =
3 0.76614
4 0.97990
5 0.37246
6 0.49749
7 0.64174
8 0.74614
9 0.88342
10 0.56876
11 0.58467
12 0.59814
13 1.92598
14 1.94462
15 1.98965
16 2.17855
17 2.47834
18 2.50225
19 2.52644
20 2.72233
```

Here are the values for all internal variables for the regularized test case:

```

1  d2 =
2      0.79393    1.05281
3      0.73674    0.95128
4      0.76775    0.93560
5
6  d3 =
7      0.888659    0.907427    0.923305   -0.063351
8      0.838178   -0.139718    0.879800    0.896918
9      0.923414    0.938578   -0.049102    0.960851
10
11 Delta1 =
12      2.298415   -0.082619   -0.074786
13      2.939691   -0.107533   -0.161585
14
15 Delta2 =
16      2.65025    1.37794    1.43501
17      1.70629    1.03385    1.10676
18      1.75400    0.76894    0.77931
19      1.79442    0.93566    0.96699
20
21 z2 =
22      0.054017    0.166433
23     -0.523820   -0.588183
24      0.665184    0.889567
25
26 sigmoidGradient(z2)
27 ans =
28      0.24982    0.24828
29      0.23361    0.22957
30      0.22426    0.20640
31
32 a2 =
33      1.00000    0.51350    0.54151
34      1.00000    0.37196    0.35705
35      1.00000    0.66042    0.70880
36
37 a3 =
38      0.88866    0.90743    0.92330    0.93665
39      0.83818    0.86028    0.87980    0.89692
40      0.92341    0.93858    0.95090    0.96085

```

Test case without regularization (uses same data, but 0 for lambda):

```

1  >> [J grad] = nnCostFunction(nn, il, hl, nl, X, y, 0)
2  J = 7.4070
3  grad =
4      0.766138
5      0.979897
6     -0.027540
7     -0.035844
8     -0.024929
9     -0.053862
10     0.883417
11     0.568762
12     0.584668
13     0.598139
14     0.459314
15     0.344618
16     0.256313
17     0.311885
18     0.478337
19     0.368920
20     0.259771
21     0.322331
22

```

=====

Values for Delta1 and Delta2 (the unregularized gradient, from tutorial Step 5 and Step 6) - truncated to 3 decimal places, prior to scaling by $1/m$.

```
1 Delta1 =
2   2.298 -0.082 -0.074
3   2.939 -0.107 -0.161
4
5 Delta2 =
6   2.650  1.377  1.435
7   1.706  1.033  1.106
8   1.754  0.768  0.779
9   1.794  0.935  0.966
10
11
```

👍 50 Upvote · Follow 12 · Reply to Tom Mosher

🔒 This thread is closed. You cannot add any more responses.

Earliest

Top

Most Recent

No Replies Yet