Discussion Forums

Get help and discuss course material with the community.

THIS WEEK'S FORUM

Week 7

Discuss and ask questions about Week 7.

2161 threads · Last post 2 days ago

Go to forum

Forums | All Threads

Search

All Course Discussions

ex4 tutorial for nnCostFunction and backpropagation

Tom Mosher Mentor Week 5 · 3 years ago · Edited

Keywords: ex4 tutorial backpropagation nnCostFunction

(note: if you have a question about this tutorial, please start a new thread. This one is full and is closed to additional replies)

==========

You can design your code for backpropagation based on analysis of the dimensions of all of the data objects. This tutorial uses the vectorized method, for easy comprehension and speed of execution.

Reference the four steps outlined on Page 9 of ex4.pdf.

Let:

m = the number of training examples

n = the number of training features, including the initial bias unit.

h = the number of units in the hidden layer - NOT including the bias unit

r = the number of output classifications

1: Perform forward propagation, see the separate tutorial if necessary.

2: δ_3 or d3 is the difference between a3 and the y_matrix. The dimensions are the same as both, (m x r).

3: z2 came from the forward propagation process - it's the product of a1 and Theta1, prior to applying the sigmoid() function. Dimensions are (m x n) · (n x h) --> (m x h)

4: δ_2 or d2 is tricky. It uses the (:,2:end) columns of Theta2. d2 is the product of d3 and Theta2(no bias), then element-wise scaled by sigmoid gradient of z2. The size is (m x r) · (r x h) --> (m x h). The size is the same as z2, as must be.

5: Δ_1 or Delta1 is the product of d2 and a1. The size is (h x m) · (m x n) --> (h x n)

6: Δ_2 or Delta2 is the product of d3 and a2. The size is (r x m) · (m x [h+1]) --> (r x [h+1])

7: Theta1_grad and Theta2_grad are the same size as their respective Deltas, just scaled by 1/m.

Now you have the unregularized gradients. Check your results using ex4.m, and submit this portion to the grader.

===== Regularization of the gradient ==========

Since Theta1 and Theta2 are local copies, and we've already computed our hypothesis value during forward-propagation, we're free to modify them to make the gradient regularization easy to compute.

8: So, set the first column of Theta1 and Theta2 to all-zeros. Here's a method you can try in your workspace console:

```
Q = rand(3,4)       % create a test matrix
Q(:,1) = 0          % set the 1st column of all rows to 0
```

9: Scale each Theta matrix by λ/m. Use enough parenthesis so the operation is correct.

10: Add each of these modified-and-scaled Theta matrices to the un-regularized Theta gradients that you computed earlier.

You're done. Use the test case (from the Resources menu) to test your code, and the ex4 script, then run the submit script.

The test case for ex4 include the values of the internal variables discussed in the tutorial.

Appendix:

Here are the sizes for the Ex4 digit recognition example, using the method described in this tutorial.

NOTE: The submit grader, the gradient checking process, and the additional test case all use different sized data sets.

a1: 5000x401

z2: 5000x25

a2: 5000x26

a3: 5000x10

d3: 5000x10

d2: 5000x25

Theta1, Delta1 and Theta1_grad: 25x401

Theta2, Delta2 and Theta2_grad: 10x26

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

Earliest **Top** **Most Recent**

Y

Yuki · 2 years ago

Hi I'm pretty confused.

From the ex4 tutorial sheet: "Concretely, you should implement a for-loop for t = 1:m and

place steps 1-4 below inside the for-loop, with the tth iteration performing

the calculation on the tth training example (x(t); y(t))." I assume you're supposed to run each training example forwards + backwards individually? i.e. we have 5000 training example, with 400 inputs x(i). so I tried passing X(m,:) in the loop.

Its not really working at the moment for me. I do the forward prop for each training example m. Then try to pass through back prop by using d3 the difference of a3 and y_matrix(m,:). Then d2 using the equation given. I get an error for tring to compute D2.

Also, confused as the test case doesnt have a theta value. how will we get the same values?

Probably I misunderstood something fundamental

6 Upvotes Hide 2 Replies

Y

Yuki · 2 years ago · Edited

Also I don't understand the equation for D1. my d2 and a1 dimensions are non comformant, a 1x25 and a 1x401 matrix..

4 Upvotes

Tom Mosher Mentor · 2 years ago

This thread is about the tutorial in the OP that teaches the vectorized method. I am not sure what "ex4 tutorial sheet" you are referring to.

The method in the ex4.pdf file is an iterative for-loop version. That's not discussed here. The iterative method is very difficult to get working, and even if it does work, it runs about 50x slower than the vectorized method.

I have closed this thread to additional comments, as the number of replies has grown so large that Coursera's forum software doesn't handle it well. If you want to continue this discussion, please start a new thread in the Week 5 forum area.

3 Upvotes

SG

Shikha Gupta · 2 years ago · Edited by moderator

Hi,

I have implemented backprop algorithm usinjg a for loop,the code has paased for the cases of Feedforward and Cost Function,Regularized Cost Function and Sigmoid Gradient

But if fails for BackPropagation. I am unable to understand what is wrong with my code:

{Mentor edit: Code removed due to Honor Code violation}

Can someone help me in understanding have I misunderstood any step in this algorithm?

0 Upvotes Hide 2 Replies

Tom Mosher Mentor · 2 years ago

Sorry, but students are not allowed to post their program scripts on the Forum. That is a violation of the course Honor Code.

I have edited your post.

I recommend you read the tutorial for this exercise, and compare it with your code.

1 Upvote

Tom Mosher Mentor · 2 years ago

Tutorials can be found here:

https://www.coursera.org/learn/machine-learning/discussions/iyd75Nz_EeWBhgpcuSIffw

1 Upvote

Ingrid Nieuwenhuis · 2 years ago

Hi, I've implemented backprop with regularization, submit says 100/100, also all the values in ex4 match the numbers the exercise says they should be. However, when I train the network the training set accuracy is 33.400000, and when I visualize the hidden units they all look identical. I remember from the lectures that maybe the weights are not set up correctly, but that's unlikely since they were given to us in the file. I'm puzzled about what could have happened... Thanks!

1 Upvote Hide 1 Reply

Tom Mosher Mentor · 2 years ago · Edited

I suspect that maybe you overlooked completing the function that randomly initializes the Theta values. That's not a graded function, so there's no automatic check that you have completed it.

See the bottom of page 7 of ex4.pdf.

The weight values we were provided with were only for testing the forward propagation part of the cost function. But the second half of ex4.m uses your cost function to train the NN - so at that point we're no longer using the provided weights.

3 Upvotes

XL

Xinghou Liu · 2 years ago

I have been trying to work this out for 6 hours!!!!!!

It turn out that in step 4, I used 'sigmoid(z2)' but this should be 'sigmoidGradient(z2)'!!!!!

All is fine now and I submit the code and pass!

Tom Thanks for your help!

17 Upvotes Reply

Tim Everett · 2 years ago

Hi,

I am having problems getting the correct values for d2. J, d3, sigmoidgradient, and Delta2 are all matching the values from the test case, but d2 doesn't come close, which leads to the gradient being out. I have used (:,2:end) to remove the bias column from Theta2, and multiplied this - d3*Theta2. I then did an element wise multiplication of the result by the sigmoid gradient of z2. The results that I got from the test case were

d2 =

0.95385 1.68470

0.94793 1.60165

0.95742 1.53805

All the dimensions match up, and I have spent the last couple of hours trying to figure out where I am going wrong. Any help or suggestions would be greatly appreciated.

0 Upvotes Hide 2 Replies

Tim Everett · 2 years ago

Please ignore this post - I had forgotten that I had done an element wise square of Theta2 for the regularised cost function and saved it over the original Theta2. All working fine now, as it would have been for the last two hours if everything had been in the right order...

4 Upvotes

AB

Alok Bhargava · 2 years ago

Excellent! I did such debugging many times. And it is frustrating but a great learning experience. All the best with the rest of the course!

1 Upvote

TN

Tyler Nigon · 2 years ago

I have a question about what fmincg() is doing in regards to updating the neural network Thetas. My understanding is that fmincg() simply optimizes our parameters (i.e., Theta1 and Theta2) so that our model is well suited to the training data. So with every execution of nnCostFunction(), we return the cost J and the partial derivatives on each neural network node (grad). Am I correct in my understanding that grad contains the new nn_params that will be used as input into the next iteration of nnCostFunction() (via fmincg())? I'm trying to implement my own rather simple gradient descent code to see how the nn_params change during convergence, but my cost is actually increasing rather than decreasing when I use grad output as the nn_params input in the next iteration of nnCostFunction().

0 Upvotes Hide 6 Replies

Tom Mosher Mentor · 2 years ago

fmincg() does the same job that your gradientDescentMulti() code did back in ex1 - except more efficiently, because it isn't limited to a fixed learning rate and fixed number of iterations.

fmincg() essentially applies the gradients - returned by your cost function - to update the Theta values. It does this as many times as necessary to get a stable solution. It uses both the cost J and the gradients - your ex1 gradient descent method used only the gradients for guidance.

If you wish to experiment, you should be able to adapt your ex1 gradient descent method and have it call your NN cost function.

0 Upvotes

TN

Tyler Nigon · 2 years ago

So if I use the gradientDescentMulti() code, for every iteration, I should:

1) execute nnCostFunction() - the first time will be with random nn_params

2) use my "grad" variable that was output from nnCostFunction() as my "nn_params" input for the next iteration (i.e., nn_params = grad;)

3) check if I've reached my maximum number of iterations

Then the output of gradientDescentMulti() would be nn_params, which is equivalent to my last grad output from nnCostFunction()?

This approach just isn't working for me - when I run the code using fmincg(), it works great, but when I use my gradientDescentMulti() code, the cost increases from iteration to iteration and the final predicted labels are all the same, which makes me think I'm doing something wrong with updating Theta1 and Theta2 from iteration to iteration.

0 Upvotes

Tom Mosher Mentor · 2 years ago

Have you tried modifying the learning rate you are using?

0 Upvotes

Tom Mosher Mentor · 2 years ago · Edited

One further detail (edited...)

The only reason this method should work is that the gradient descent method works with any set of gradients.

It might be a good idea to start with a simpler experiment - see if you can get the same results on ex2 (which is logistic regression) using your ex1 gradient descent method, as you do for ex2 with fminunc().

0 Upvotes

TN

Tyler Nigon · 2 years ago

I have not adjusted the learning rate. Just to be clear, the learning rate could correspond to 'lambda' in this exercise (not 'alpha' as it did in ex2)? Or should it be a separate variable?

0 Upvotes

Tom Mosher Mentor · 2 years ago · Edited

Lambda is the regularization parameter.

Alpha is the learning rate.

Totally different things.

0 Upvotes

xiang zhou · 2 years ago

One more question Tom,

so when we calculate d2 we need to remove the first biased column:d2(:,2:end), but when we calculate d1 , we still keep the first column.

so why we skip the biased column for d2 but not d1?

thanks!

erik

1 Upvote Hide 4 Replies

Tom Mosher Mentor · 2 years ago

There is no d1.

0 Upvotes

xiang zhou · 2 years ago

sorry Tom, my question is why should we exclude the biased units when we calculating the previous layer grad?

Is it because the error of the biased units(namely theta0) would only affect the the error of the next layer which would be used to calculate the derivative of theta0 of the current layer and have nothing to do with the previous layer so when we go back we should exclude it?

thanks!

Erik

0 Upvotes

Tom Mosher Mentor · 2 years ago

The bias unit in the hidden layer does not connect back to the input layer. So we do not need to perform backpropagation for it into the Theta1 matrix.

2 Upvotes

xiang zhou · 2 years ago

Thank you Tom!

0 Upvotes

xiang zhou · 2 years ago · Edited

Hi Tom,

I checked every step of the tutorial and don't know where got wrong.

My grad shows all zeros.

what possibly could be wrong?

Many thanks!

Edit:

Do worry about it Tom , the Theta1 is CASE sensitive and that's why it is not working.

Took me 4 hours!!! to find this tiny bug!!!

Thank you for your help throughout out the course Tom!

You are the hero ;)

erik

0 Upvotes Hide 1 Reply

Tom Mosher Mentor · 2 years ago

I'm glad you fixed it.

0 Upvotes

coco · 2 years ago

pay attention to step 4 "then element-wise scaled by sigmoid gradient of z2 ", I neglected this sentence and spent plenty of time debugging

2 Upvotes Reply

Herman Autore · 2 years ago

What's the size of z3?

I'm trying to troubleshoot my function. I noticed z3 is not in your Appendix.

0 Upvotes Hide 1 Reply

Tom Mosher Mentor · 2 years ago

Since we don't add a bias unit to the output, z3 and a3 are the same size.

0 Upvotes

Sunil Skanda · 2 years ago

My J is computing correctly, but the grad values are off more than the tom's test case answer. What could go wrong ?

0 Upvotes Hide 1 Reply

Tom Mosher Mentor · 2 years ago

If you implement the equations wrong, you will get the wrong answers.

If you include the first column of Theta in regularization, you will get the wrong answers.

The most common issue is how you handle the first column of Theta2 when you backpropagate to compute Theta1_grad.

I believe the test case includes the values of all of the variables inside the cost funtion. Those should be helpful. Test cases are here:

https://www.coursera.org/learn/machine-learning/discussions/iyd75Nz_EeWBhgpcuSIffw

0 Upvotes

Shivam · 2 years ago

I am getting error "not enough inputs" in sigmoidGradient .I have tried almost .But I can not figure out what s wrong with this?

0 Upvotes Hide 4 Replies

Tom Mosher Mentor · 2 years ago

The sigmoidGradient() function only requires one parameter.

0 Upvotes

Shivam · 2 years ago

But I have checked it and still getting same error again and again.It does not recognise z neither sigmoid nor sigmoidGradient

0 Upvotes

Tom Mosher Mentor · 2 years ago · Edited

The function template script for sigmoid() and sigmoidGradient() defines a single parameter 'z' to be passed to the function. Hopefully you have not changed this definition.

When you use the function, you must pass it a parameter. Otherwise the script will complain that you have not provided enough inputs.

0 Upvotes

Tom Mosher Mentor · 2 years ago

Maybe it would help if you posted a screen capture that shows the commands you entered and the error message.

0 Upvotes

Sankaranarayanan P N · 2 years ago

Extremely thankful for this tutorial. It made the work easier

0 Upvotes Reply

subhojit · 2 years ago

Hi Tom,

I am getting an error while submitting the code for this exercise which is related to some proxy settings. More specifically , this is the error that i am getting .

!! Submission failed: unexpected error: Error using urlreadwrite (line 98)

Error downloading URL. Your network connection may be down or your proxy settings improperly configured.

!! Please try again later.

Can you suggest a way around it.

Thanks ,

Subhojit

0 Upvotes Hide 1 Reply

Tom Mosher Mentor · 2 years ago · Edited

If your computer is behind a strong firewall, proxy server, or anti-virus protection, then you need access through those features before you can submit your work. See your system administrator if necessary.

There's nothing on Coursera's end of the system that can help with this.

1 Upvote

A

AlainH · 2 years ago

Thanks again for this. I have a question. d3 starts out at 5000x25 (so m x r) but in Step 6 you refer to it as (r x m) - the formula calls for the product d3 x transpose of a2 - I'm a bit confused about why you transposed d3 (which makes mathematical sense but does not match the formula from the ex4.pdf page 9 step 4). It's been the source of many headaches for me when the formula in the exercise doesn't match the actual implementation.

0 Upvotes Hide 2 Replies

Tom Mosher Mentor · 2 years ago

The method in the ex4.pdf file is for an iterative solution that works on one training example at a time. The tutorial is for the vectorized solution. The implementation of the code is different for the two methods.

One big benefit of the vectorized method is that it runs about 50x faster than the iterative method.

3 Upvotes

Tom Mosher Mentor · 2 years ago · Edited

I find the iterative method given in ex4.pdf hopelessly confusing. I was unable to complete this exercise using that method when I was a student of the course.

5 Upvotes

1 2 3 4 … 8

DESCRIPTION

Welcome to the course discussion forums! Ask questions, debate ideas, and find classmates who share your goals. Browse popular threads below or other forums in the sidebar.

MODERATORS

AC KK VA KS LK SL MS FL PS MR CU JN JD S T AT

Learn more about becoming a Mentor

Forum guidelines