Programming Assignment: Neural Network Learning

Deadline The assignment was due on August 14, 11:59 PM PDT
You can still pass this assignment before the course ends.

Instructions

My submission

Discussions

Discussions

ex4 Tutorial for forward propagation and cost

Tom Mosher Mentor Week 5 · a year ago · Edited

Note: this thread is closed to comments. If you have a question, please post it in the Week 5 Discussion Forum area.

This tutorial uses the vectorized method. If you're using a for-loop over the training examples, you're doing it the hard way, and you're on your own.

A note on Errata: The cost and gradient equations in the ex4.pdf file are correct. There may be some errata in the video lectures. Check the Course Wiki to be sure.

I'll use the less-than-helpful greek letters and math notation from the video lectures in this tutorial, though I'll start off with a glossary so we can agree on what they are. I will also suggest some common variable names, so students can more easily get help on the Forum.

It is left to the reader to convert these descriptions into program statements. You will need to determine the correct order and transpositions for each matrix multiplication, so that the result has the correct size.

Glossary:

Each of these variables will have a subscript, noting which NN layer it is associated with.

Θ: A Theta matrix of weights to compute the inner values of the neural network. When we used a vector theta, it was noted with the lower-case theta character θ.

z : is the result of multiplying a data vector with a Θ matrix. A typical variable name would be "z2".

a : The "activation" output from a neural layer. This is always generated using a sigmoid function $g()$ on a z value. A typical variable name would be "a2".

δ : lower-case delta is used for the "error" term in each layer. A typical variable name would be "d2".

Δ : upper-case delta is used to hold the sum of the product of a δ value with the previous layer's a value. In the vectorized solution, these sums are calculated automatically though the magic of matrix algebra. A typical variable name would be "Delta2".

Θ_gradient : This is the thing we're solving for, the partial derivative of theta. There is one of these variables associated with each Δ. These values are returned by nnCostFunction(), so the variable names must be "Theta1_grad" and "Theta2_grad".

$g()$ is the sigmoid function.

$g'()$ is the sigmoid gradient function.

Tip: One handy method for excluding a column of bias units is to use the notation SomeMatrix(:,2:end). This selects all of the rows of a matrix, and omits the entire first column.

See the Appendix at the bottom of the tutorial for information on the sizes of the data objects.

A note regarding bias units, regularization, and back-propagation:

There are two methods for handing exclusion of the bias units in the Theta matrices in the back-propagation and gradient calculations. I've described only one of them here, it's the one that I understood the best. Both methods work, choose the one that makes sense to you and avoids dimension errors. It matters not a whit whether the bias unit is excluded before or after it is calculated - both methods give the same results, though the order of operations and transpositions required may be different. Those with contrary opinions are welcome to write their own tutorial.

Forward Propagation:

We'll start by outlining the forward propagation process. Though this was already accomplished once during Exercise 3, you'll need to duplicate some of that work because computing the gradients requires some of the intermediate results from forward propagation. Also, the y values in ex4 are a matrix, instead of a vector. This changes the method for computing the cost J.

1 - Expand the 'y' output values into a matrix of single values (see ex4.pdf Page 5). This is most easily done using an eye() matrix of size num_labels, with vectorized indexing by 'y'. A useful variable name would be "y_matrix", as this...

```
1  y_matrix = eye(num_labels)(y,:)
```

Note: For MATLAB users, this expression must be split into two lines, such as...

```
1  eye_matrix = eye(num_labels)
2  y_matrix = eye_matrix(y,:)
```

Discussions of other methods are available in the Course Wiki - Programming Exercises section.

2 - Perform the forward propagation:

a_1 equals the X input matrix with a column of 1's added (bias units) as the first column.

z_2 equals the product of a_1 and Θ_1

a_2 is the result of passing z_2 through $g()$

Then add a column of bias units to a_2 (as the first column).

NOTE: Be sure you DON'T add the bias units as a new row of Theta.

z_3 equals the product of a_2 and Θ_2

a_3 is the result of passing z_3 through $g()$

Cost Function, non-regularized:

3 - Compute the unregularized cost according to ex4.pdf (top of Page 5), using a_3, your y_matrix, and m (the number of training examples). Note that the 'h' argument inside the log() function is exactly a3. Cost should be a scalar value. Since y_matrix and a3 are both matrices, you need to compute the double-sum.

Remember to use element-wise multiplication with the log() function. Also, we're using the natural log, not log10().

Now you can run ex4.m to check the unregularized cost is correct, then you can submit this portion to the grader.

Cost Regularization:

4 - Compute the regularized component of the cost according to ex4.pdf Page 6, using Θ_1 and Θ_2 (excluding the Theta columns for the bias units), along with λ, and m. The easiest method to do this is to compute the regularization terms separately, then add them to the unregularized cost from Step 3.

You can run ex4.m to check the regularized cost, then you can submit this portion to the grader.

Appendix:

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

NOTE: The submit grader (and the gradient checking process) uses a different test case; these sizes are NOT for the submit grader or for gradient checking.

a1: 5000x401

z2: 5000x25

a2: 5000x26

a3: 5000x10

d3: 5000x10

d2: 5000x25

Theta1, Delta1 and Theta1_grad: 25x401

Theta2, Delta2 and Theta2_grad: 10x26

=========

Here is a link to the test cases, so you can check your work:

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

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

=========

keywords: ex4 tutorial nncostfunction forward propagation

93 Upvote · Follow 79 · Reply to Tom Mosher

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

Earliest

Top

Most Recent

J jonney · a year ago

super helpful. I guess most of it was covered in the lectures, but was good to have it all in one place thanks!

2 Upvote · Hide 3 Replies

 Smith Sen · a year ago

In this function how to make Theta ,as our input is theta only?

1 Upvote

 Tom Mosher Mentor · a year ago

See the first line of nnCostFunction.m (the "function" statement) to see what values are input to the function, and what are returned. The nn_parameters variable is a vector that contains both Theta matrices unrolled.

1 Upvote

 Richard Wheatley · 4 months ago

Tom, why is this tutorial not using MathJax?

0 Upvote

YH Yuli Hassine · a year ago · Edited

I followed the steps but still get 89 instead of 0.28 for the cost without regularization.

I multiply a 10 by 10 eye matrix with the logs, and sum twice ... dont understand what Im missing ...

3 Upvote · Hide 37 Replies

See earlier replies

 Tom Mosher Mentor · a year ago

Indeed. Octave and MATLAB are optimized for multiplying matrices.

0 Upvote

AR Alexei Ramotar · a year ago

Hi Tom, first I apologize for placing this in the wrong discussion group, kind of lazy. In ex4 they gave us the Theta1 and Theta2. How do they get these values for the Thetas? Are they using gradient descent treating each layer as a logarithmic regression? Or will they tell us later?

1 Upvote

 Tom Mosher Mentor · a year ago

The Theta matrices come from having a trained neural network. We can't generate those unit we have a working backpropagation/learning method - so for ex3 we're given the trained Theta values.

In ex4, we'll learn how to generate them ourselves.

The method is covered in the Week 5 videos.

0 Upvote

M Machete · a year ago

I don't get how to create y. Is it simply

```
y = 1:10;
```

?

0 Upvote

Tom Mosher Mentor · a year ago

y is provided for you when ex4.m calls your nnCostFunction().

0 Upvote

M

Machete · a year ago

Oh I landed here because I searched for some hints for solving ex3. This explains a lot, as it's ex4 :) Discussion search should be improved...

0 Upvote

Tom Mosher Mentor · a year ago

Agreed, the search tools are awful, and we've asked Coursera to make some improvements.

It's also a problem that ex3 and ex4 both have "predict.m" functions, but they each work totally differently.

2 Upvote

HA

Hania El Ayoubi · a year ago

Tom, for calculating unregularized J in nnCostFunction in ex4, I am puzzled why we need to use element multiplication when I did the same function in ex3 and did not use element multiplication.Is it because y is a matrix in ex4?

3 Upvote

Tom Mosher Mentor · a year ago

Yes.

1 Upvote

SS

surjit kumar singh · a year ago

Hi Tom, I am stuck with forward propagation Cost calculation, I am following below steps:

1. getting X adding 1 to column making it 5000*401 matrix and treating it as a1.

2. getting z2 and a2 as per theta1 * a1 and sigmoid on z2 respectfully.

3. adding bias 1 column to z2 and getting a3 by multiplying z2 and theta2.

4. using a3 to do matrix multiplication with log of y_matrix as per cost calculation.

I am able to get J of 10*5000 matrix but many value in matrix is either NaN or Inf. I am not sure whether it is expected or I am missing something. What would be final Cost matrix size?

0 Upvote

Tom Mosher Mentor · a year ago · Edited

Your step 2 is incorrect, it gives the wrong shape for z2. You cannot multiply Theta1 and a1 as the sizes are (25x401) and (5000 x 401). The inner dimensions don' t match. Try a1 * Theta1'.

Your step 3 is also incorrect. You add the bias units AFTER using the sigmoid function.

You need to use the sigmoid() function to get a3 also.

Here's the stepwise process:

- Add a column of bias units to X. This gives you the a1 matrix.
- Multiply a1 and Theta1. This gives you the z2 matrix. Its size should be 5000 x 25.
- Take the sigmoid() of the z2 matrix. Tip: Save this as 'u'. You can use it later when you need to compute the sigmoid gradient of z2.
- Add a column of bias units. This gives you a2.
- Multiply a2 and Theta2. This gives you z3. Its size should be 5000 x 10,
- Take the sigmoid() of z3. This gives you a3.

The shape of a3 should be (5000 x 10).

2 Upvote

Fadwa Fawzy · a year ago

Another way to build the y matrix is to use **sub2ind** function.

0 Upvote

Jackie Tang · 10 months ago

I seems that I am getting the cost of 89 problem as well. I am not understanding what you mean by not using element-wise multiplication within the log, since the log of a3 is not multiplied by anything within.

Also, before applying the double sum, J is a 10 by 10 matrix, yes? I converted my ones column to be a 5000 by 10 matrix or 10 by 5000 depending on how it subtracted from the y_matrix or a3.

I am not sure what I am doing wrong, and I tried the test cases and am not getting the right answer. I double sum only after the entire 10 by 10 matrix has been created though such that I end up with a 1 by 1 scalar value.

0 Upvote

Tom Mosher Mentor · 10 months ago

Do NOT use matrix multiplication for computing the cost. Use element-wise multiplication, then use a double sum over the 5000 x 10 cost values.

3 Upvote

Jackie Tang · 10 months ago

Okay thanks I think I get it now. If you do not use element-wise multiplication, you end up with a 5000 by 5000 matrix or 10 by 10 matrix even before the double summation is applied.

0 Upvote

Tom Mosher Mentor · 10 months ago

Indeed. And if you do a double-summation of either of those matrices, you get the wrong answer.

0 Upvote

GK

Gautam Karmakar · 8 months ago

Hi Tom,

I am getting error for z2 = a1 * Theta1 as 25x401 * 401X5000 any idea?

0 Upvote

SG

shreya gupta · 7 months ago

Hello sir,

I have been receiving the same "i got 89 " error, however when i try using element-wise multiplication with the log function MATLAB shows the following error : "Error using .* Matrix dimensions must agree."

whereas i confirmed that my

y_matrix= 5000 x 10

and a3= 5000 x 10

why do i get this error?

1 Upvote

Tom Mosher Mentor · 7 months ago

Sorry, I am confused. What do you mean "i got 89" error?

0 Upvote

Tom Mosher Mentor · 7 months ago

How do you know that the log() calculation is where the error "matrix dimensions" occurs?

0 Upvote

JG

Jaimie Gillette · 6 months ago

I've been able to get the right answer for the cost function with the test case and via submission, but I'm confused about why we are using element wise multiplication between the y matrix and the log term. In all previous calculations of the cost function, I've been using normal matrix multiplication, and getting the right results. Why did this change?

0 Upvote

Tom Mosher Mentor · 6 months ago

Matrix multiplication works fine when you're multiplying two vectors - you get the sum automatically. That's for linear and logistic regression, where the hypothesis is a vector.

But in a NN, the hypothesis is a matrix also - and if you use matrix multiplication, the only elements you want to sum for the cost are the ones on the diagonal. You'll have to work out a small example by hand to see why this is true.

6 Upvote

AR

Anjanibhargavi Ragothaman · 5 months ago

When I run the ex4 nnCostFunction, I'm getting 0.287629 for the feedforward without regularization, but when I try the test case example, I'm not getting J = 19.474, instead I'm getting 7.4070. Any idea what's going wrong?

0 Upvote

Tom Mosher Mentor · 5 months ago

The test case has a lot of matrices that are (unfortunately) square - so it doesn't detect problems such as adding bias units as rows, instead of columns, that sort of thing.

The tutorial has a list of the correct sizes of the variables. Put a breakpoint in your code and verify that everything is the correct size. That is a good starting point.

0 Upvote

AR

Anjanibhargavi Ragothaman · 5 months ago

Ah! I was looking at the tutorial which has results for regularized version (lambda=4), the test case revised link has both lambda=0 and lambda=4. lambda=0 (unregularized version) has the same result as I do. Thanks a lot Tom!

0 Upvote

Tom Mosher Mentor · 5 months ago

I'll update the tutorial so that's less confusing.

1 Upvote

JR

Jonathan Roman · 3 months ago · Edited

After completing Step 3 above, for the unregularized test case values, I get J = 7.4070 but the grad vector is all 0's. Your test case shows grad has non-zero values.

When I submit, it starts, then fails. (Submission failed: unexpected error: horizontal dimensions mismatch (3x1 vs 16x4))

Is the error related to the grad vector being all 0's or something else in my code?

Were we supposed to do anything with grad at this point?

2 Upvote

Tom Mosher Mentor · 3 months ago

I think you've got a couple of problems.

The grad vector should not be all zeros. So fix that first.

The submit issue is that the submit grader uses a different sized data set, and your code isn't able to handle it.

0 Upvote

SH

Syeda Hashmi · 2 months ago

the same problem is occuring with me and i am unable to see the 35 replies on this post . Please make it visible to me!!

0 Upvote

Tom Mosher Mentor · 2 months ago

If you can't see the replies in this thread, then please start a new thread.

Which "same problem" are you having? Please be specific.

0 Upvote

AH

Ashley Hutchings · a year ago

Tom,

For Step 3, computing the un-regularized cost function, can you explain how you code a double sum? Will sum(sum(equation code)) work? Thanks.

4 Upvote · Hide 4 Replies

Tom Mosher Mentor · a year ago · Edited

Yes. Try this in your workspace:

```
Q = magic(3)
Q1 = sum(Q)      % gives a vector result
Q2 = sum(Q1)     % gives a scalar result
sum(sum(Q))      % gives the same result
```

15 Upvote

James Newton · a year ago

You can also sum any n dimensional matrix by first converting the matrix to a vector. In Octave Q(:) is a vector with all the elements of Q. So sum(sum(Q)) is the same as sum(Q(:)) if Q is 1 or 2 dimensions. But it is possible to have matrixes with more than 2 dimensions, in which case sum(Q(:)) works and sum(sum(Q)) does not. Doesn't matter here, but sum(Q(:)) strikes me as a "best practice" in general.

6 Upvote

Tom Mosher Mentor · a year ago · Edited

Indeed, although it is not obvious to the novice what the (:) notation does.

1 Upvote

AB

ABDUL BASEER BURIRO · a year ago

Hi Tom,

I implemented nnCostFunction with the help of procedure outlined in ex4.pdf. For the very last part I used

J=-sum((y.*log(a3(:,i)))+((1-y).*log(1-a3(:,i))))/m;

within a "FOR LOOP" and did not get the correct answer. while for matrix, I got it.

The question is as why there is a dire need of converting vector y to a matrix while rare use of for loop previously given correct results.

0 Upvote

AH

Ashley Hutchings · a year ago

I'm on step 4, nnCostFunction with regularization but I'm not quite getting the correct cost. Not sure what's wrong. Here's what I've done. First I removed the first column from Theta1 and Theta2 (I call them t1 and t2). Then to get t1^2 I take t1'*t1 and get the double sum, then I do that for t2, plug them in appropriately to the cost function... not sure what I'm missing...

2 Upvote · Hide 15 Replies

Tom Mosher Mentor · a year ago

When you've got a matrix, multiplying by its inverse does not give you a double sum of the squares.

3 Upvote

Tom Mosher Mentor · a year ago

Better to use sum(sum(Q.^2))

4 Upvote

AH

Ashley Hutchings · a year ago

t1'=transpose of t1

0 Upvote

Tom Mosher Mentor · a year ago · Edited

Understood.

With a vector, you can compute the sum of the squares of the elements using a vector transposition and multiplication.

```
1  v = [1 ; 2 ; 3]       % a column vector
2  v' * v                % vector multiplication
3  sum(v.^2)             % same answer
```

With a matrix, the same trick does not work.

```
1  Q = [1 2 3;4 5 6]      % a non-square matrix
2  Q'*Q                   % gives another matrix!
3  sum(sum(Q'*Q))         % not the sum you're looking for
4  sum(sum(Q.^2))         % probably what you want
```

However, there is a function that will sum the elements of the diagonal of a matrix - trace().

```
1  trace(Q'*Q)
```

If you work through the details of the matrix multiplication, the elements that are are included in computing the sum of the squares of the elements all all lie along the primary diagonal.

When I evaluated this dilemma a few months ago, I found that trace() wasn't the best choice from a performance standpoint, at least the way I was using it.

You can instrument your code by putting the 'tic' and 'toc" statements around a set of statements. When executed, the 'toc' statement will print out the elapsed execution time since the previous 'tic' statement. It's pretty handy for investigating this sort of performance question.

10 Upvote

AA

Alan A. Alexander · a year ago

Hi Tom, can I use this method to exclude the bias units for the theta colums: (sum(sum(Theta1(2:end).^2))) ? . I'm getting cost value of 0.384430 instead of 0.383770...

5 Upvote

Tom Mosher Mentor · a year ago

With a matrix, you always need to provide two index values. Your notation only gives one. Try these in your console to see the difference:

```
1  Q = magic(3)
2  Q(2:end)
3  Q(:,2:end)
```

For your purpose, you'll want the latter.

16 Upvote

AA

Alan A. Alexander · a year ago

Oh.... Thanks Tom!

0 Upvote

Harry Lewin · a year ago

Trace() seems to be 25% to 37% slower than a sum(sum()) statement in the couple of cases I tested. But for me, the Trace() statement is much easier to write and debug.

Thanks for your good work in putting together this tutorial. I'd be lost without it.

0 Upvote

Tom Mosher Mentor · a year ago

I agree, once I figured out how trace() worked for this exercise, I found it was fairly slow, and abandoned it.

0 Upvote

GK

Gautam Karmakar · 8 months ago

Hi, This is what I have done for non regularized cost and not sure I got the sum right. Please let me know if I am doing anything wrong:

calculate a3, y_matrix then

doing a

j = 1/m[sum(sum(((-ylog(a3) - (1-y)log(1-a3)))]

0 Upvote

Tom Mosher Mentor · 8 months ago · Edited

Use element-wise multiplication with the log() function.

Be sure your 1/m has enough parenthesis around it to give the correct scaling factor.

0 Upvote

GK

Gautam Karmakar · 8 months ago

corrected but while running it getting mismatch error for z2 = a1 * Theta1;

0 Upvote

Tom Mosher Mentor · 8 months ago

You've got to transpose one of them, so you get a result of the correct size.

0 Upvote

Tom Mosher Mentor · 8 months ago

Note that you're getting very close to violating the course Honor Code. Don't post your lines of code.

0 Upvote

GK

Gautam Karmakar · 8 months ago

Oops I am sorry.

0 Upvote

RB Rees Boffey · a year ago · Edited

Hi Tom,

I've just been reading your tutorial and was wondering if there is there any reason why we can't use the included predict.m function to calculate h theta of X when calculating the cost?

0 Upvote · Hide 5 Replies

Tom Mosher Mentor · a year ago

The backpropagation process needs some internal values from the forward prediction process, and the predict.m function does not return those.

0 Upvote

Tom Mosher Mentor · a year ago

Also note that we're not interested in predictions at this point in the cost function. We need all the activation levels for the output layer, not the actual predictions.

0 Upvote

Min Yang · 9 months ago

Hi,

Why am I getting Cost Function w/ Regularization) 0.379046, not 0.383770?

1 Upvote

S Shubha · 8 months ago

Hi,

I get the correct answer for the cost function (w/o regularization) but with regularization I get an answer of

0.384488 which is not correct. So I tried to test the double sum on a test matrix

A = [1, 2, 3; 4, 5, 6]

and the double sum gives me an answer of 91 as expected. So I know I have added the terms correctly. What could be going wrong?

1 Upvote

S Shubha · 8 months ago

Sorry, I think I figured it out.

0 Upvote

JK Jordan A. Kodner · a year ago

Given matrices with sizes

z2: 5000x25

d3: 5000x10

Theta2: 10x26

Can you please explain how step 3 on page 9 (δ(2) = □Θ(2)□T δ(3). ∗ g′(z(2))) yields

d2: 5000x25

?

It's the .* operation that is confusing me, since it's a by-element operation and the matrices on either side don't have the same dimensions.

Thanks

0 Upvote · Reply

Tom Mosher Mentor · a year ago

Your question is about backpropagation. You've posted in a thread that discusses forward propagation.

There's a different thread that addresses your question.

https://www.coursera.org/learn/machine-learning/discussions/i2u9QezvEeSQaSIACtiO2Q/replies/XpcX6-0PEeS0tyIAC9RBcw

0 Upvote · Reply

SA

Sekhar Attaluri · a year ago

Tom,

Thank you for putting this together. I was completely lost and this helped me understand how to implement the concepts. Couldnt have managed without this.

Regards,

Sekhar Attaluri

3 Upvote · Hide 3 Replies

Tom Mosher Mentor · a year ago

Cool.

2 Upvote

SAIFUL RIZAL BIN MUHD RAMLI · 9 months ago

Hi Tom,

Phew your solution is really elegant and I agree with you, this is far more intuitive than trying to make it work with a for loop!

Thanks!!

0 Upvote

SAIFUL RIZAL BIN MUHD RAMLI · 9 months ago

I forgot to mention that the y_matrix method is the one that helped me to understand how to vectorize this.

0 Upvote

J

Jianbin · a year ago

Dear Tom

I have problems in computing the J.

I found y given is 5000X1 and the first value inside was 10 and then 1 and 2... I have computed h as a 5000X10 matrix. So, I am trying to do transpose y and one column of h in a for-loop to get J. But I got J which was about 300.

p.s. Your method looks much easier. But I couldn't get what "eye(num_labels)(y,:)" means.

Thank you

0 Upvote · Hide 8 Replies

JP

Jorge Miguel Santos Pinto · a year ago

What "eye(num_labels) (y,:)" is permutation over identity matrix of size num_labels!

If you search for those words you will have a clear idea about what it does. In MATLAB you can't use that code directly, you need to divide it into two code lines and, in my opinion, you stay with a more clear idea about what it does.

```
1  Id=eye(N);
```

- creates an identity matrix (Id) with size (N x N)

```
1  A=( Id (y,:) );
```

- creates a row vector that corresponds to the row 'y' of 'Id'. Being 'y' also a vector, the returned result will be a vector of vectors (matrix)

Play a while with both instructions and you will figure it out pretty fast!

Hope it helps!

4 Upvote

Tom Mosher Mentor · a year ago

Thanks for the explanation.

You don't need the outer set of parenthesis in your "A = " formula.

1 Upvote

PS

pramesh shakya · a year ago

y is a 5000x1 vector so why is it stated("Also, the y values in ex4 are a matrix, instead of a vector.") that it's a matrix? are we supposed to convert this y vector into a 5000x10 matrix of ones and zeroes?

0 Upvote

Tom Mosher Mentor · a year ago

Yes, you first have to convert the 'y' vector in to a 'y_matrix' of logical vectors.

2 Upvote

T

Tarneem · 10 months ago

I could not figure it out how to do it, I wrote it like this:

p= eye(num_labels);

y= p(y,:);

but still getting y of size 5000x1

please help!

0 Upvote

T

Tarneem · 9 months ago

Hi Tom,

do I need to save the y matrix in other variable or I can over-write the original one as i did in the above code?

0 Upvote

Alex Furnica · 5 months ago

Why does y have to be a matrix? Why does the code for creating the y_matrix that you posted give us what we want? Wouldn't it just create a random identity matrix?

I tried googling permutation and such but I really still don't understand what this piece of code does. I also played around with the example code you gave and still I don't understand what is going on. It seems like magic.

0 Upvote

Tom Mosher Mentor · 5 months ago

We're training multiple classifiers, so we want the output for every training example to be a set of values, where one is a '1' and all the others are '0's

Try those commands that create the y_matrix in your console, to see how they work.

0 Upvote

JP

Jorge Miguel Santos Pinto · a year ago

Hi Tom!

After spending some hours looking for errors on my program I've decided to submit it anyway despite of my FP algorithm return a value of 6.931472 instead of the predicted one: 0.287629.

I've got full score despite of the returned values difference... Any clue why this happened?

BR,

JP

2 Upvote · Reply

JP

Jorge Miguel Santos Pinto · a year ago

After applying regularization I've got the right result!

1 Upvote · Reply

J

Jianbin · a year ago

Hello Tom

Can I also do vector product to get D2 and D1? I did it and used sigmoidGradient to get Theta1_grad and Theta2_grad. But I found some errors raised.

I got all the values under "Sigmoid gradient evaluated at [1 -0.5 0 0.5 1]:"

and then I saw these things

Initializing Neural Network Parameters ...

Checking Backpropagation...

Error using horzcat

0 Upvote · Hide 1 Reply

Tom Mosher Mentor · a year ago

The "horzcat" error is usually due to a problem in how you're calculating the gradients in the cost function.

0 Upvote

James Newton · a year ago · Edited

This (very VERY valuable) thread does not appear to be listed in your list of tutorials:

https://www.coursera.org/learn/machine-learning/discussions/m0ZdvjSrEeWddiIAC9pDDA

0 Upvote · Hide 1 Reply

Tom Mosher Mentor · a year ago

That's the thread that is used to find all of the tutorials. We've recently added that thread (it's now our logical "top level") since students have a very hard time using the search capabilities.

0 Upvote

LK

Larry Kline · a year ago

Although my program gets the correct cost for the first step, I get an error in the backprop step:

Checking Backpropagation...

error: nnCostFunction: product: nonconformant arguments (op1 is 5x3, op2 is 5x10)

And therefore my submission fails. Do I need to implement everything before I can submit?

BTW, I implemented a completely vectorized approach, no for loop.

0 Upvote · Hide 1 Reply

Tom Mosher Mentor · a year ago

Seems like you need to transpose something before you try to multiply them.

0 Upvote

Bryan Lim · a year ago

Hey how do you input latex in this forum?

0 Upvote · Hide 23 Replies

See earlier replies

Agostino Sturaro · 8 months ago · Edited

I wrote lambda\(2*m) instead of lambda/(2*m). The backslash did not produce any error, but it skewed the results. Terrible!

2 Upvote

Tom Mosher Mentor · 8 months ago · Edited

The forward-slash character is a different operator, but it is still a syntactically valid so doesn't generate an error.

0 Upvote

Felipe Moreira · 7 months ago

Tom, I just wanted to say that you're the man.

That's all.

2 Upvote

Xavi Lizarraga · 5 months ago

Tom, thanks a lot for your thorough explanation. It was crucial for me. Best wishes!

0 Upvote

GG

Gulshan K Garg · 4 months ago

Hi Tom, I am stuck with forward propagation getting the h matrix right before Cost calculation, I am following below steps:

1. getting X adding 1 to matrix and treating it as a1.

2. getting z2 as per a1 * theta1'

3. Getting a2 by using sigmoid on z2 .

4. adding bias 1 column to a2

5. getting z3 by multiplying a2 and theta2'.

6. Getting a3 by using sigmoid on z3 .

7. Converting a3 to 1s and 0s by using a3>=0.5 command and getting a3

8. Converting a3 to h_matrix by using the formula provided by you for eye_matrix

all steps pass but while executing step 8, I get the following error :

Index exceeds matrix dimensions.

Error in nnCostFunction (line 87)

h_matrix = eye_matrix(a3,:);

On further investigation I found out that multiple entries in a3 are 1 when scanned rowwise.

what am I doing wrong?

0 Upvote

Tom Mosher Mentor · 4 months ago

Your steps 7 and 8 are not needed, and are incorrect.

0 Upvote

Tom Mosher Mentor · 4 months ago

Step 7 is not needed.

Step 8 should convert y into y_matrix. a3 does not need to be converted.

0 Upvote

ZH

ziqi hu · 3 months ago

i get the right answer of 0.287629 in ex4, but when i submit the answer it says:

!! Submission failed: unexpected error: Error using horzcat

Dimensions of matrices being concatenated are not consistent.

may i ask what's the problem?

0 Upvote

Paul T Mielke Mentor · 3 months ago · Edited

You must have mismatching matrix dimensions. As usual, the dimensions of the test cases used by submit are different than those in ex4. Your best bet is to use Tom's test cases for this exercise to give your code some other dimensions to handle. Also with the test cases, it's easier to get the line number at which the error is happening. Here are the test cases for the nnCostFunction in ex4.

0 Upvote

GV

George Vincent · 3 months ago

Sorry I'm stuck with this assignment. I'm trying to grasp what each step is doing to compute the cost function. Im OK with finding activation parameters; but didn't really understand what

```
1  y_matrix = eye(num_labels)(y,:)
```

statement does . I searched *help* in octave for '*eye*' that takes 2 parameters but couldn't find any.

0 Upvote

Tom Mosher Mentor · 3 months ago

It makes slightly more sense if you break it into two statements:

```
1  Q = eye(num_labels)
2  y_matrix = Q(y,:)
3
```

The (y,:) part is a vector index. Each 'y' value in the vector causes a row of the Q matrix to be copied into y_matrix.

1 Upvote

GV

George Vincent · 3 months ago

Thanks for the quick reply Tom,

Considering num_labels = 3

So Q will be [1, 0, 0; 0, 1, 0; 0, 0, 1]

and let y = [1;1;2;2;3;3]

what would y_matrix look like and how.

0 Upvote

Tom Mosher Mentor · 3 months ago

I think you can enter those commands in your workspace and see the result for yourself.

0 Upvote

L

L.L. · 2 months ago

you said the size ofa3 is 5000x10,so I guess h θ (x (i)) k = a (3)k , whose size is 5000x1;but the size of y(i) is 10x1;

so when I calculate J,how can I calculate −y (i)k*log((h θ (x (i))) k) − (1 − y (i)k)*log(1 − (h θ (x (i))) k)

0 Upvote

Tom Mosher Mentor · 2 months ago

You must convert 'y' from a vector into a matrix of logical values.

0 Upvote