Back to Week 5 Lessons Prev Next

Programming Assignment: Neural Network Learning

Deadline The assignment was due on February 12, 11:59 PM PST
You can still pass this assignment before the course ends.

Instructions

My submission

Discussions

Assignment: Neural Network Learning

ex4 Tutorial for forward propagation and cost

Tom Mosher Mentor · 2 years 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...

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

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

```
eye_matrix = eye(num_labels)
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

70 Upvote · Follow 80 · Reply to Tom Mosher

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

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

Krishnang Dalal · 7 months ago

Hello Tom,

Can you please briefly explain what is double sum doing based on the cost function formula? Also where is the formula to calculate 'grad'?

0 Upvote · Hide 1 Reply

Tom Mosher Mentor · 7 months ago

The double-sum() is summing over the rows and columns of the cost values.

The grad formula is in the exercise PDF file.

1 Upvote

victor yampolsky · 7 months ago · Edited by moderator

I'm so frustrated...

I worked all night and morning on this and I still cant submit succesfully.

even when I regularize the cost function J in nnCostfunction the submit. only successful (partly) when I mark the regularization as comment this works:

```
{code removed}
```

please HELP. this is really frustrating!!!

in general went through the tutorial, there is some issues with the matrices sizes in the appendix.

I really need help:

this is the full code:

why on earh this is wrong?

```
{code removed}
```

I anyone can help I'll be greatfull!!!

0 Upvote · Hide 2 Replies

Paul T Mielke Mentor · 7 months ago

Hi, Victor.

I'm sorry that you are frustrated by this, but we are not allowed to post source code for solutions in this course. It probably will not be any consolation, but this assignment is by far the most challenging in the entire course. When you say that your code is not generating the correct results, do you mean that you are using Tom's test cases? Or just that submit is failing.

Also I'm not understanding your comment about setting lambda = 1 making everything work. I notice that you have the value of lamdba hard-wired to 1 and are ignoring the passed in argument in your code as posted. That's got to be a big clue as to what is wrong.

0 Upvote

Tom Mosher Mentor · 7 months ago

Sorry, students are not allowed to post their code, per the course Honor Code.

The matrix sizes in the appendix are correct.

0 Upvote

AK

Ayodeji Kuponiyi · 8 months ago · Edited by moderator

Hi Tom, thanks a lot for you time and help. Up until this point everything worked well.

I'm not sure what I am doing wrong with the FeedForward program, could you kindly help look through my pseudocode:

I followed the following steps:

{Mentor edit: detailed line-by-line code description removed}

It runs without error but no marks.

Thanks

0 Upvote · Hide 2 Replies

Tom Mosher Mentor · 8 months ago

Your description is just as detailed as actually posting the code itself, which is not allowed under the course Honor Code.

So I cannot comment on it.

You may be able to debug your code by using the additional test cases, from the Resources menu.

0 Upvote

AK

Ayodeji Kuponiyi · 8 months ago

Apologies Tom and to the forum, it's my first time of posting and was trying to present my steps without being too vague. I promise that won't happen next time. I'd try the additional test cases.

Thanks

0 Upvote

AS

Andrey Shkabko · 8 months ago · Edited

About regularization term writtent in ex4.pdf p.6.

The sum for k should be from 1 to 401 and from 1 to 26 to get "correct" J = 0.383770 after biases (k=1) removal (and sum for k = 2 to 401 and k = 2 to 26).

If you do it according to the formula written in ex4.pdf p.6 than ....Theta1(:,2:400).^2)... and 0.381095 is a correct J

1 Upvote · Reply

JF

Justin Francis · 8 months ago

Sorry if this has been asked before already Tom, but exactly how is

y_matrix = eye(num_labels)(y,:)

Creating the indexed matrix of the values of y. I understand what it creates but how is it doing this?

0 Upvote · Hide 3 Replies

Tom Mosher Mentor · 8 months ago · Edited

Reading left to right, it creates an eye matrix of num_labels size. Then it uses 'y' as a vectorized index. For each value in y, it copies that row of the eye matrix into y_matrix. That's what the (y,:) does - the colon means "use all columns".

Note that MATLAB doesn't support this syntax, so it must be split into two lines of code.

0 Upvote

Paul T Mielke Mentor · 8 months ago · Edited

It's easier to understand if you split it into two lines:

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

So what that says is we index the rows of A using the vector y and then the ":" in the column position means "Use all corresponding columns of that row". So the first row of the output matrix will be determined by using $y(1)$ to select one of the rows of A (which is eye(n)). And so on through the y vector. So each row of the output is the row of $I(n)$ that corresponds to the value of $y(i)$.

3 Upvote

JF

Justin Francis · 8 months ago

It finally clicks in my brain, my logic in explaining y_matrix = A(y,:) follows:

For each value of y it creates a new matrix with the value of the y'th row in A

Seems so simple now but I could not wrap my mind around it earlier.

1 Upvote

H

Harpreet · 8 months ago

For calculating the delta_2 correctly I have to do delta_2 := delta_2(:,2:end) in step 3 and skip this in step 4. If I dont do it my matrix multiplications and element multiplication with sigmoid doesnt work correctly. I do it after doing Theta2'*delta_3 and before element multiplication with sigmoid.

0 Upvote · Reply

SB

Suraj Bhandari · 8 months ago

Hi Tom,

In the videos, Andrew says that we don't calculate error term for input layer. So there isn't any ∂1 (smaller delta-1). So for a 3 layer network, we will have just ∂3 and ∂2. But when we calculate Δ (bigger delta) for the same 3 layer network, we don't have Δ3 (bigger delta-3) and all we can have is Δ1 and Δ2. If Δ (bigger delta) is to accumulate the error, why do we have it for layer 1 and not for 3? Or am I missing something here? Please help.

0 Upvote · Hide 2 Replies

Tom Mosher Mentor · 8 months ago

There are a lot of errors in the video lectures - especially for the NN exercise.

I'm not familar with this particular question. Maybe check the Course Wiki and see if there are some better notes on this lecture.

0 Upvote

SB

Suraj Bhandari · 8 months ago

Sure Tom. I'll go through the wiki.

0 Upvote

JR

Jeffrey Robinson · 9 months ago

The neural network in my brain couldn't manage the very first task of this assignment...until I read Tom's tutorial. Thanks, Tom. There was a huge difference in complexity between what I was trying to do and what the final solution ended up being -- such a simple and elegant line of code, whereas I was trying all sorts of for loops and other nonsense. Wow!

1 Upvote · Hide 9 Replies

Tom Mosher Mentor · 9 months ago

Once you start thinking like a matrix, it all becomes clear.

0 Upvote

MS

MAHATHI SRINIVASAN · 9 months ago

hello,

i have followed the steps of calculating the non-regularized cost function using vector implementation. My a3 is of the correct dimension(5000 X 10) .i have performed element-wise multiplication of the two matrices(log of a3 and y_matrix(which is also5000 X 10)) but my answer comes out as

Cost at parameters (loaded from ex4weights): -66.727017

(this value should be about 0.287629).

i have used the correct formula. I am unable to figure out the error.How to rectify this and get the expected answer?

Also, are all the entries of matrices Theta1 and Theta2 from ex4weights.mat zero?

thanks,

1 Upvote

Tom Mosher Mentor · 9 months ago

No, they are not all zero.

Where did you obtain the correct formula?

0 Upvote

MS

MAHATHI SRINIVASAN · 9 months ago

i used the formula given in the ex4 pdf.

0 Upvote

MS

MAHATHI SRINIVASAN · 9 months ago

Also,when i finish computing the cost(which displays -66.727107), i find that Theta1 and Theta2 turn out to be null matrices(i.e after the non-regularized cost computation)

0 Upvote

Tom Mosher Mentor · 9 months ago

What do you mean by "null matrix"?

A matrix full of zeros?

Or a matrix with no size?

Neither of those are correct, I'm just trying to understand what you're seeing.

Where in the workspace do you see this problem with the matrices? Inside your cost function, or within the scope of ex4.m?

0 Upvote

MS

MAHATHI SRINIVASAN · 9 months ago

after performing the cost function ,both Theta1 and Theta2 come out as matrices full of zeroes.This happens after running ex4.m

0 Upvote

MS

MAHATHI SRINIVASAN · 9 months ago

i have figured it.My cost function was missing a few brackets.

Thanks,Tom.

0 Upvote

Tom Mosher Mentor · 9 months ago

Good news.

0 Upvote

M

manish · 9 months ago · Edited

Hi,

I'm following correct steps, dimensions of matrix are also correct(printing at each step for debug) but still I'm getting cost as -25.484421 instead of 0.287629. Plz help.

In last step I'm doing matrix multiplication for a3 and y.(transpose of y). which gives 10x10 matrix. Then double sum over the result dividing by input size. I'm not clear where I'm going wrong. Signs are used as per PDF.

TIA.

0 Upvote · Hide 9 Replies

Tom Mosher Mentor · 9 months ago

The tutorial says to use element-wise multiplication.

0 Upvote

M

manish · 9 months ago

element wise multiplication gives me same answer. :(

0 Upvote

M

manish · 9 months ago

Hi Tom,

I figured out that I was passing a3 again through sigmoid function. But when I removed that part of code I'm getting NaN instead of cost. Any clue where I'm going wrong?

TIA.

0 Upvote

Tom Mosher Mentor · 9 months ago · Edited

Im your post, you said "In the last step...". Can you be specific which "last step" you're referring to?

When you element-wise multiply using the log() of a3 and y_matrix, the results should be size (m x k); that is, 5000 x 10, not 10 x 10. So I don't really understand what you're doing well enough to offer any advice.

0 Upvote

Tom Mosher Mentor · 9 months ago

Also, be sure that you're using the formulas from the ex4.pdf file. The formulas in the video lectures are incorrect in some places.

0 Upvote

M

manish · 9 months ago

Earlier one was typo... I'm doing element wise multiplication of log() of a3 and y_matrix. size is 5000x10.

Result1: Earlier result, i.e., -25.484421 was coming when I was passing a3 through sigmoid in turn calculating z3 and passing z3 for calculation, which of course is incorrect.

Result2: I rectified the result1 mistake and did element wise multiplication of log() of a3 and y_matrix which gave me 5000x10 matrix. Putting this result in formula I got NaN. P.S.- I'm using formula in pdf.

0 Upvote

M

manish · 9 months ago

I got it. I was missing a loop in sigmoid calculation. Thanks Mat for all your time. :)

0 Upvote

M

manish · 9 months ago

Hi Mat,

For regularized cost I'm getting 0.327271 instead of 0.383770. Is it ok? or should I rectify it?

TIA.

0 Upvote

M

manish · 9 months ago

Resolved. Mistake: Took 1 st column of theta matrices. :)

0 Upvote

RLi · 9 months ago

Tks Tom

0 Upvote · Reply

Michael Hesse · 9 months ago

Thanks Tom. Was really scratching my head about an efficient way to compute y_matrix from y. :-)

0 Upvote · Reply

Julie Lorin · 10 months ago

Hi,

I have another question about the computation of the cost function. I'd just like to check if I am doing it the correct way of if there is a better way. (It is annoying not to be able to just copy paste my code in this situation :D)

I took the cost "formula" from the previous exercice, but as I am doing it with matrix instead of vector, my result is a 10*10 matrix, instead of a scalar. In this 10*10 matrix, I only keep the diagonal (using a .* eye(10)) and I sum and sum again this result, which gives me a correct cost (well after my division by m and the regularization if I need it). I am kind of sceptic with this method, because I compute thing that I don't use after (everything that is not on the diagonal), so I am thinking that maybe there is a better way.

I had to put all my matrices on paper to find this logic, and it works both for the cost and the regularization, but is this the better way ?

Thanks !

1 Upvote · Hide 4 Replies

Tom Mosher Mentor · 10 months ago

I presume you've read the tutorial at the top of this thread.

A modification to your method would be using the trace() function, instead of using element-wise multiplication by an eye matrix.

Which method is better depends on how you define "better".

0 Upvote

Julie Lorin · 10 months ago

Thanks for the answer ! I didn't know the trace function but it seems way easier than my multiplication by the eye matrix and then my double sum. It does exactly what I wanted (sum the diagonal elements). At least it conforts me that I was not totally wrong with my method and that I have to calculate useless numbers (the ones that are not in the diagonal). I just wanted to know if there was a "cleaner" method where we doesn't need to compute those numbers not on the diagonal.

(I am only a beginner in this, I am trying to do it the best way each time but matrix calculations are not natural to me and I am not too good with the computation cost of my method ! So maybe it is the best way :))

0 Upvote

Tom Mosher Mentor · 10 months ago

The knock against trace() is that it is reputed to run rather slowly.

You can instrument your code by framing it with the "tic" and "toc" statements. When "tic" is executed, a timer is started. When "toc" is executed, the timer is stopped and the runtime is displayed. It's one of several ways of measuring relative code performance.

0 Upvote

Julie Lorin · 10 months ago

Thanks for the info, I'll try that !

0 Upvote

Darwin R.C. · 10 months ago

Good day, all

I know I'm missing something here, I followed the tutorial but I'm getting the following results with the test case without regularization:

```
J = -19.281
grad =

-0.56599
-0.56063
-0.55609
-0.55228
-0.77765
.
.
.

```

I also don't understand what is meant by "NOTE: Be sure you DON'T add the bias units as a new row of Theta."

Maybe my issue lies there.

Can anyone give me a hint about it?

0 Upvote · Hide 4 Replies

Tom Mosher Mentor · 10 months ago

The note is a reminder that bias units are columns, not rows. It is frightfully easy to make some drastic errors in the cost function and still end up with a NN that almost works.

The cost J should always be a positive value. Be sure you haven't made any sign errors. The first thing to check is that you're using the equations from the ex4.pdf file, and not the ones you may have seen in the videos.

0 Upvote

Darwin R.C. · 10 months ago

Thanks for your quick answer Tom, I will take a look at my equation, surely the problem is there then.

0 Upvote

Darwin R.C. · 10 months ago

Just found my mistake... I was using y instead of y_matrix in the formula. I can't believe I spent 3 hours struggling with this :(Thanks for the insight Tom!

0 Upvote

Tom Mosher Mentor · 10 months ago

Good catch. As I said, "frightfully easy...".

0 Upvote

Julie Lorin · 10 months ago · Edited

Hi,

I hope the question hasn't been already asked, but I didn't find it. For the first step, to have the y matrix, I don't understand why it works.

I tested the code in command line, but I don't understand. Working with matrix, and selecting them is not the most natural thing for me yet.

First, I don't understand why we need the "eye" matrix, and not one full of zeros (but I may understand this part if I understood the next, I guess)

I don't understand why selecting eye_matrix(y,:) gives the good result. y is a vector of values between 1 and 10, and for me, the y here must be used to select the rows, and then saying all columns. And the result is supposed to have 5000 rows, but we only select from a 10 rows matrix. I guess I totally misunderstood what this does, but I can't figure out myself. So what happens here ?

When I first did it without the tutorial, I had to use a for loop as I was unable to do it in a vectorized way like this !

Thanks ! (sorry for the bad english !)

0 Upvote · Hide 7 Replies

Tom Mosher Mentor · 10 months ago · Edited

Good questions.

The 'y' vector we're provided in this exercise is valued from 1 to 10, just as it was in ex3. But to train a logistic classifier, we need the output values to be 0's and 1's only.

In ex3, we accomplished this by using the code "y == c" inside the for loop from c=1:num_labels. That gives us 0's and 1's like this (try the commands in your console):

```
y = [1 3 4 2]
c = 2
y == c
```

On each iteration of the loop, the statement "y == c" returns a different vector, which is passed to our cost function as the 'y' values it expects. Each pass through the for-loop give us back a theta vector, which is trained to recognize one class only.

In ex4, we also need to train on 0's and 1's. But since we're training all 10 classifiers at once (the NN has 10 outputs), we can't use 'y' as a vector. 'y' must be a matrix, and since this is a classifier, the values must be limited to only 0's and 1's.

That's where the eye matrix trick comes in.

The statement "eye(num_labels)" returns an identity matrix. For each row, there is a '1' in a single column. That's perfect for training all of our classifiers at once. There is one column for each label.

The syntax "y_matrix = eye(num_labels)(y,:)" is a vector addressing trick. For each row of 'y', the statement is executed once. The 'y' value is an index into the identity matrix. Essentially, 'y,:' causes the appropriate row of the identity matrix to be copied entirely and placed in y_matrix.

4 Upvote

Tom Mosher Mentor · 10 months ago

Example for eye_matrix:

```
y = [1 3 4 2]
eye_matrix(4)
y_matrix = eye_matrix(y,:)

```

1 Upvote

Tom Mosher Mentor · 10 months ago

Also note, I've edited my previous posts, so please refresh your browser to get the latest version.

0 Upvote

Julie Lorin · 10 months ago

Hi ! Thanks for the answer ! I think that I understand this way better ! As y is full of numbers between 1 and 10, the (y,:) selects the row with the same index as the number in y (as the first parameter is the row selection). (That's what I understood, I may not be explaining it very well, but I was trying to tell it in my own words so that if I totally misunderstood, someone can tell me :D)

Anyway, I think that I got it, it makes sense ! Thanks !

0 Upvote

June Yu · 8 months ago

This makes alot more sense to me now as well! Thanks, Tom!

0 Upvote

JR

Jonathan Roman · 8 months ago

Super dumb questions: What does "train a logistic classifier" mean?

Is num_labels the number of classifiers?

(I successfully submitted this assignment- just asking a dumb question)

0 Upvote

Tom Mosher Mentor · 8 months ago · Edited

Is num_labels the number of classifiers?

Yes. It can be called the number of labels, the number of classes, or the number of NN outputs.

0 Upvote

LL

lennie leong · a year ago

HI Tom and mentors,

i am working on my first assignment in ex4, nnCostFunction.m.

I am curenttly getting " J = NaN 1.6054 0.5897 3.0771 2.0912" & I am not able to get the correct scores. Is there any areas where i could look at to solve the problem?

(On a side note: i had used the test cases. My results for J were: J = NaN 5.7023 6.2358 6.7824 7.3402, and my "grad" were all zeros.)

please help mentors. Thank you very much in advanced!

0 Upvote · Hide 8 Replies

Tom Mosher Mentor · a year ago · Edited

The cost "J" should be a scalar. Yours appears to be a vector. Fix that first.

Follow the tutorial. Use the test cases. Really, everything we know about this exercise is right there.

There's not much more help we can offer than that.

0 Upvote

LL lennie leong · a year ago

Thanks Tom,

I had relooked at my codes. I think I am getting close. Currently, my

J = 2.4593 0.5791 1.2843 3.6999 1.4549

It is still a vector though. I would like to check, scalar is just one number. How do I turn my J into scalar?

I was searching "scalar" in the search bar but it isn't returning much discussions around "scalar". Is there any link / test cases you can direct me for this insight please?

Thank you very much in advanced!

0 Upvote

Tom Mosher Mentor · a year ago

Please re-read the last two lines of my previous post.

0 Upvote

LL lennie leong · a year ago · Edited

I had re-read it. Using the test cases, it seems that my z3 (i.e product a2 and theta2) is having error: "Error using * Inner matrix dimensions must agree", "Error in nnCostFunction"

I had printed both a2 and Theta 2. a2 is showing 16x4 matrix. while Theta2 is showing 4x5 matrix. And i took product of the 2 which prints out 16x5. But when i did a test case, it is showing this error. (No transposition were used)

this seems odd given i am very sure i didn't make a error. more specifically, i understand that if Matrix A = h x r, and Matrix B = r x t. Multiplying A*B gives us a h x t matrix.

Can mentors kindly help please?

0 Upvote

LL lennie leong · a year ago · Edited

I was also re-reading the tutorials. Tom mentioned about:

"Then add a column of bias units to a2 (as the first column).

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

What does it mean to say dont add ... as a new row of Theta? I had added a column of 1's in a2. ie. Initially a2 = 16x4. Now, a2 = 16x4 (but with 1's on first column). I used someMatrix(:,1) = 1. Not sure this might be the hiccup - as i don't quite understand this part after re-reading it again. Just wanted to be sure i am working them on the right track.

Using the test-case, it has showed:

a2 = [1.0000 0.5415 ; 1.0000 0.3571 ; 1.0000 0.7088]

J = 0

Grad = column_of_zeros

0 Upvote

Tom Mosher Mentor · a year ago · Edited

Don't do what it says to not do, and you'll be fine.

0 Upvote

LL

lennie leong · a year ago · Edited

My a2 is a matric of 16x4 now. Should it remain 16x4 after adding 1's?

I'm just curious why is Theta in the picture when we are working on a2.

0 Upvote

Paul T Mielke Mentor · a year ago · Edited

If you are adding a column of bias units, then the matrix should have one more column after the operation. From your comments a couple of posts ago it sounds like you overwrote one of the existing columns of a2 with the bias units, which is why the matrix doesn't change size.

0 Upvote

1 2 3 4 ... 7