MITx 6.86x

Machine Learning with Python-From Linear Models to Deep Learning

Help smitha_kannur

Course Progress Dates Discussion Resources

Course / Unit 3 Neural networks (2.5 weeks) / Homework 3

< Previous ✎ ✓ ✎ ✓ ✎ ✓ ✎ ✓ Next >

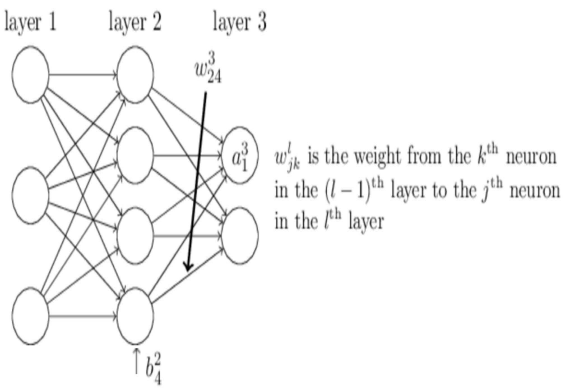
3. Backpropagation

Bookmark this page

Homework due Oct 28, 2020 05:29 IST *Completed*

One of the key steps for training multi-layer neural networks is stochastic gradient descent. We will use the back-propagation algorithm to compute the gradient of the loss function with respect to the model parameters.

Consider the L-layer neural network below:


layer 1
layer 2
layer 3
w^3_{24}
a^3_1
b^2_4
w^l_{jk} is the weight from the k^{th} neuron in the $(l-1)^{\text{th}}$ layer to the j^{th} neuron in the l^{th} layer

In the following problems, we will the following notation: b^l_j is the bias of the j^{th} neuron in the l^{th} layer, a^l_j is the activation of j^{th} neuron in the l^{th} layer, and w^l_{jk} is the weight for the connection from the k^{th} neuron in the $(l-1)^{th}$ layer to the j^{th} neuron in the l^{th} layer.

If the activation function is f and the loss function we are minimizing is C, then the equations describing the network are:

$$\begin{aligned} a^l_j &= f\left(\sum_k w^l_{jk} a^{l-1}_k + b^l_j\right) \\ \text{Loss} &= C\left(a^L\right) \end{aligned}$$

Note that notations without subscript denote the corresponding vector or matrix, so that a^l is activation vector of the l^{th} layer, and w^l is the weights matrix in l^{th} layer.

For $l = 1, \ldots, L$.

Computing the Error

2/2 points (graded)

Let the weighted inputs to the d neurons in layer l be defined as $z^l \equiv w^l a^{l-1} + b^l$, where $z^l \in \mathbb{R}^d$. As a result, we can also write the activation of layer l as $a^l \equiv f\left(z^l\right)$, and the “error" of neuron j in layer l as $\delta^l_j \equiv \frac{\partial C}{\partial z^l_j}$. Let $\delta^l \in \mathbb{R}^d$ denote the full vector of errors associated with layer l.

Back-propagation will give us a way of computing δ^l for every layer.

Assume there are d outputs from the last layer (i.e. $a^L \in \mathbb{R}^d$). What is δ^L_j for the last layer?

● $\frac{\partial C}{\partial a^L_j} f'\left(z^L_j\right)$

○ $\sum_{k=1}^{d} \frac{\partial C}{\partial a^L_k} f'\left(z^L_j\right)$

○ $\frac{\partial C}{\partial a^L_j}$

○ $f'\left(z_j^L\right)$

✓

What is δ_j^l for all $l \neq L$?

● $\sum_k w_{kj}^{l+1} \delta_k^{l+1} f'\left(z_j^l\right)$

○ $\delta_k^{l+1} f'\left(z_j^l\right)$

○ $\sum_k w_{jk}^{l-1} \delta_j^{l-1} f'\left(z_j^l\right)$

○ $\sum_k w_{kj}^{l+1} \delta_k^{l+1} f\left(z_j^l\right)$

✓

Solution:

We make use of the chain rule.

1. By definition, $\delta_j^L = \frac{\partial C}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L} f'\left(z_j^L\right)$.

2. We have:

$$\begin{aligned} \delta_j^l &= \frac{\partial C}{\partial z_j^l} \\ &= \sum_k \frac{\partial C}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_j^l} \\ &= \sum_k \frac{\partial z_k^{l+1}}{\partial z_j^l} \delta_k^{l+1} \end{aligned}$$

Then we have $z_k^{l+1} = \sum_j w_{kj}^{l+1} a_j^l + b_k^{l+1} = \sum_j w_{kj}^{l+1} f\left(z_j^l\right) + b_k^{l+1}$. Taking the derivative of this with respect to z_j^l gives $w_{kj}^{l+1} f'\left(z_j^l\right)$.

Combining the two gives the final answer: $\delta_j^l = \sum_k w_{kj}^{l+1} \delta_k^{l+1} f'\left(z_j^l\right)$.

Submit You have used 2 of 2 attempts

ℹ Answers are displayed within the problem

Parameter Derivatives

2/2 points (graded)

During SGD we are interested in relating the errors computed by back-propagation to the quantities of real interest: the partial derivatives of the loss with respect to our parameters. Here that is $\frac{\partial C}{\partial w_{jk}^l}$ and $\frac{\partial C}{\partial b_j^l}$.

What is $\frac{\partial C}{\partial w_{jk}^l}$? Write in terms of the variables a_k^{l-1}, w_j^l, b_j^l, and δ_j^l if necessary.

Example of writing superscripts and subscripts:

`delta_j^l` for δ_j^l

`w_(jk)^l` for w^l_{jk}

$\frac{\partial C}{\partial w^l_{jk}} =$ a_k^(l-1)*delta_j^l ✓ **Answer:** a_k^(l-1)*delta_j^l

$a^{l-1}_k \cdot \delta^l_j$

What is $\frac{\partial C}{\partial b^l_j}$? Write in terms of the variables a^{l-1}_k, w^l_j, b^l_j, and δ^l_j if necessary.

$\frac{\partial C}{\partial b^l_j} =$ delta_j^l ✓ **Answer:** delta_j^l

δ^l_j

STANDARD NOTATION

Solution:

1. $\frac{\partial C}{\partial w^l_{jk}} = \frac{\partial C}{\partial z^l_j}\frac{\partial z^l_j}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j$

2. $\frac{\partial C}{\partial b^l_j} = \frac{\partial C}{\partial z^l_j}\frac{\partial z^l_j}{\partial b^l_j} = 1 * \delta^l_j$

Submit You have used 1 of 5 attempts

Answers are displayed within the problem

Activation Functions: Sigmoid

4/4 points (graded)

Recall that there are several different possible choices of activation functions f. Let's get more familiar with them and their gradients.

What is the derivative of the sigmoid function, $\sigma(z) = \frac{1}{1+e^{-z}}$? Please write your answer in terms of e and z:

(1-1/(1+e^(-z)))*(1/(1+e^(✓ **Answer:** e^(-z) / (1 + e^(-z))^2

$\left(1 - \frac{1}{1+e^{-z}}\right) \cdot \left(\frac{1}{1+e^{-z}}\right)$

Which of the following is true of $\sigma'(z)$ as $||z||$ gets large?

○ Its magnitude becomes large.

● Its magnitude becomes small.

○ It suffers from high variance.

✓

What is the derivative of the ReLU function, $\text{ReLU}(z) = \max(0, z)$ for $z > 0$?

1 ✓ **Answer:** 1

1

For $z < 0$?

0 ✓ **Answer:** 0

0

STANDARD NOTATION

Solution:

$\sigma'(z) = \sigma(z)(1 - \sigma(z))$. As z gets large in magnitude, the sigmoid function saturates, and the gradient approaches zero.

ReLU is a simple activation function. Above zero, it has a constant gradient of 1. Below zero, it is always zero.

Submit You have used 1 of 5 attempts

Answers are displayed within the problem

Simple Network

0/4 points (graded)

Consider a simple 2-layer neural network with a single neuron in each layer. The loss function is the quadratic loss: $C = \frac{1}{2}(y-t)^2$, where y is the prediction and t is the target.

Starting with input x we have:

- $z_1 = w_1 x$
- $a_1 = \text{ReLU}(z_1)$
- $z_2 = w_2 a_1 + b$
- $y = \sigma(z_2)$
- $C = \frac{1}{2}(y-t)^2$

Consider a target value $t = 1$ and input value $x = 3$. The weights and bias are $w_1 = 0.01$, $w_2 = -5$, and $b = -1$.

Please provide numerical answers accurate to at least three decimal places.

What is the loss?

0.26722332269 ✗ **Answer:** 0.28842841648243966

What are the derivatives with respect to the parameters?

$\frac{\partial C}{\partial w_1} =$ 0.1966 ✗ **Answer:** 2.0809165621704553

$\frac{\partial C}{\partial w_2} =$ ✗ **Answer:** -0.00416183312434091

$\frac{\partial C}{\partial b} =$ ✗ **Answer:** -0.13872777081136367

STANDARD NOTATION

Solution:

Using the chain rule, we have:

- $\frac{\partial C}{\partial w_1} = \frac{\partial C}{\partial y}\frac{\partial y}{\partial z_2}\frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1} = (y-t)\,y\,(1-y)\,w_2\mathbf{1}\{z_1 > 0\}x$
- $\frac{\partial C}{\partial w_2} = \frac{\partial C}{\partial y}\frac{\partial y}{\partial z_2}\frac{\partial z_2}{\partial w_2} = (y-t)\,y\,(1-y)\,a_1$
- $\frac{\partial C}{\partial b} = (y-t)\,y\,(1-y)$

Submit You have used 1 of 5 attempts

Answers are displayed within the problem

SGD

1/1 point (graded)

Referring to the previous problem, what is the update rule for w_1 in the SGD algorithm with step size η? Write in terms of w_1, η, and $\frac{\partial C}{\partial w_1}$; enter the latter as `(partialC)/(partialw_1)`, noting the lack of space in the variable names:

Next w_1 = w_1-eta*(partialC)/(parti ✓ **Answer:** w_1 - eta * (partialC)/(partialw_1)

STANDARD NOTATION

Solution:

The definition of the simple SGD update rule is new_parameter = old_parameter - learning_rate * derivative of loss w.r.t old parameter.

Submit You have used 1 of 5 attempts

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: Unit 3 Neural networks (2.5 weeks):Homework 3 / 3. Backpropagation

Add a Post

Show all posts by recent activity

Another approach - Simple Network 12
Pinned Community TA

Backpropagation resource - Simple network 7
I found this well done and helpful to apply backpropagation, also in the context of the assignment "Simple Network": https://www.y...
Pinned

On the notation 4
For clarity, I believe the weight sub-indexes could be switched to make it consistent with regular matrix notation. For example, W_L,i,...

? What is the point of doing an online course in deep learning when you have to derive and understand the equations on your own? 25
I am probably missing the point of the course structure but I don't understand what is the point of basically assigning to you the tas...

Staff:about 'Sorry, couldn't parse formula' 7
even if i am entering the formula as per given technique, it says 'Sorry, couldn't parse formula'.

Simple Network chain derivatives answer input 3
Hi everyone, Once again I find myself loosing points due to not understanding the right way to input the solution (happened already ...

[STAFF] Simple Network Help
[STAFF] Im stuck on this question. I tried to start by taking the derivatives for y, but y is defined in terms of sigmoid, wich is defined i... 2

Hint for Parameter Derivatives
I found it a lot easier to do the Simple Network question first, and then I went back and did Parameter Derivatives. I think I could hav... 4

Simple Network
I am not able to get the derivative of the loss with respect to w_1 even though i could get all the other answers correct. Not sure whe... 17

Simple Network: does it make sense to expand? 6

SGD: how to represent η in standard notation?
SGD: how to represent η in standard notation? 3

On the 'simple network' section
One way and less confusing for me is just to use the chain rule on differentating the individual expressions given for z1, a1, z2, y and... 1

[STAFF] "Parameter Derivatives": question formatting isssue delta_i\^l 4

‹ Previous

Next ›

Accessibility Policy
Trademark Policy
Sitemap

Connect

Blog
Contact Us
Help Center
Media Kit
Donate

GET IT ON Google play

Download on the App Store