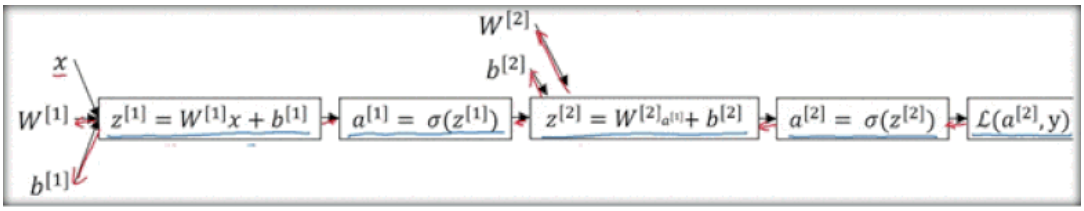
All the Backpropagation derivatives

Patrick David
Jun 7, 2018 · 11 min read



$$\frac{\partial L}{\partial a}$$

All

So you've completed Andrew Ng's Deep Learning course on Coursera,

You know that ForwardProp looks like this:

$$\begin{aligned}
Z^{[1]} &= W^{[1]}X + b^{[1]} \\
A^{[1]} &= g^{[1]}(Z^{[1]}) \\
Z^{[2]} &= W^{[2]}A^{[1]} + b^{[2]} \\
A^{[2]} &= g^{[2]}(Z^{[2]}) \\
&\vdots \\
A^{[L]} &= g^{[L]}(Z^{[L]}) = \hat{Y}
\end{aligned}$$

Forwardpropagation Equations

And you know that Backprop looks like this:

$$\begin{aligned}
dZ^{[L]} &= A^{[L]} - Y \\
dW^{[L]} &= \frac{1}{m} dZ^{[L]} A^{[L]^T} \\
db^{[L]} &= \frac{1}{m} np.\text{sum}(\text{d}Z^{[L]}, axis = 1, keepdims = True) \\
dZ^{[L-1]} &= W^{[L]^T} dZ^{[L]} g'^{[L]}(Z^{[L-1]}) \\
&\vdots \\
dZ^{[1]} &= W^{[L]^T} dZ^{[2]} g'^{[1]}(Z^{[1]}) \\
dW^{[1]} &= \frac{1}{m} dZ^{[1]} A^{[1]^T} \\
db^{[1]} &= \frac{1}{m} np.\text{sum}(\text{d}Z^{[1]}, axis = 1, keepdims = True)
\end{aligned}$$

Backprop Equations

But do you know how to derive these formulas?

But do you know how to derive these formulas?

TL;DR

Full derivations of all Backpropagation derivatives used in Coursera Deep Learning, using both chain rule and direct computation.

If you've been through backpropagation and not understood how results such as

$$A - Y$$

The derivative of our linear function - dz

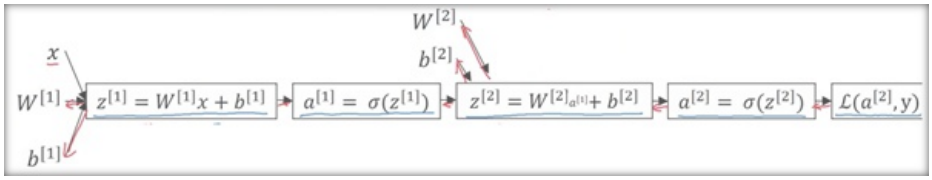
and

$$\frac{\partial L}{\partial a} = [\frac{-y}{a} + \frac{(1-y)}{(1-a)}]$$

derivative of Cost w.r.t activation 'a'

are derived, if you want to understand the direct computation as well as simply using chain rule, then read on…

Our Neural Network



Neural Net taken from Coursera Deep Learning.

This is the simple Neural Net we will be working with, where x,W and b are our inputs, the "z's" are the linear function of our inputs, the "a's" are the (sigmoid) activation functions and the final

$$L(a, y)$$

Cross Entropy cost function

is our Cross Entropy or Negative Log Likelihood cost function.

So here's the plan, we will work backwards from our cost function

$$L(a,y) = -\frac{1}{m}\left[\sum_{i=1}^{m} y ln(a) + (1-y) ln(1-a)\right]$$

Our cost function

and compute directly, the derivative of

$$L(a,y)$$

Cross Entropy cost function

with respect to (*w.r.t*) each of the preceding elements in our Neural Network:

$$\frac{\partial L}{\partial a}\frac{\partial L}{\partial z}\frac{\partial L}{\partial w}\frac{\partial L}{\partial b}$$

The derivatives of L(a,y) w.r.t each element in our NN

As well as computing these values *directly*, we will also show the *chain rule* derivation as well.

 # Note: we don't differentiate our input 'X' because these are fixed values that we are given and therefore don't optimize over.

[1] Derivative w.r.t activation function

$$\frac{\partial L}{\partial a}$$

[1] derivative of our activation function

So to start we will take the derivative of our cost function

$$L(a,y)$$

w.r.t the activation function

$$a^{[2]}$$

Activation function 2

So we are taking the derivative of the Negative log likelihood function (Cross Entropy) , which when expanded looks like this:

$$\frac{\partial L}{\partial a} - [y ln(a) + (1 - y) ln(1 - a)]$$

Taking derivative of our cost function

First lets move the minus sign on the left of the brackets and distribute it inside the brackets, so we get:

$$[-y ln(a) - (1 - y) ln(1 - a)]$$

distribute minus sign

Next we differentiate the left hand side:

$$\frac{-y}{ln(a)}$$

l.h.s

The right hand side is more complex as the derivative of ln(1-a) is not simply 1/(1-a), we must use chain rule to multiply the derivative of the inner function by the outer.

$$\frac{d}{dx}[\ln g(x)] = \frac{1}{g(x)} g'(x)$$

the derivative of a log

The derivative of (1-a) = -1, this gives the final result:

$$\frac{\partial L}{\partial a} = [\frac{-y}{a} - (-)\frac{(1 - y)}{(1 - a)}]$$

derivative of L w.r.t activation ‘a’

$$\frac{\partial L}{\partial a} = [\frac{-y}{a} + \frac{(1 - y)}{(1 - a)}]$$

final result

And the proof of the derivative of a log being the inverse is as follows:

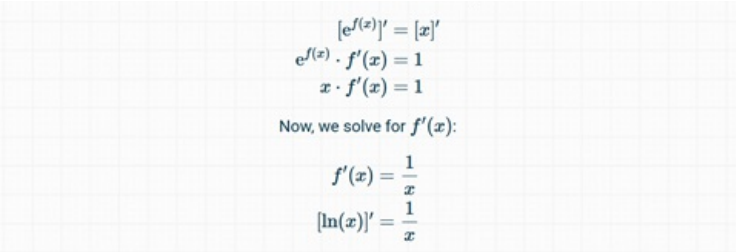
Proof for the derivative of the natural logarithm:

Let $f(x) = \ln(x)$. To be found: $f'(x) = [\ln(x)]'$.

The natural logarithm is the inverse function of the exponential function e^x, and vice versa, therefore:

$$e^{f(x)} = x$$

We differentiate both sides of the equation:



proof for the derivative of a log

[2] Derivative of sigmoid

$$\frac{\partial a}{\partial z}$$

[2] derivative of sigmoid

It is useful at this stage to compute the derivative of the sigmoid activation function, as we will need it later on.

our logistic function (sigmoid) is given as:

$$\frac{1}{1+e^{-z}}$$

Sigmoid (Logistic) function

First is is convenient to rearrange this function to the following form, as it allows us to use the chain rule to differentiate:

$$(1 + e^{-z})^{-1}$$

Rearranged sigmoid function

Now using chain rule: multiplying the outer derivative by the inner, gives

$$-(1 + e^{-z})^{-2} \cdot \left[e^{-z} \cdot -1\right]$$

outer derivative x inner derivative

which rearranged gives

$$\frac{e^{-z}}{(1+e^{-z})^2}$$

put RHS over LHS

Here's the clever part. We can then separate this into the product of two fractions and with a bit of algebraic magic, we add a '1' to the second numerator and immediately take it away again:

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

add a ‘1’ and subtract a ‘1’ on RHS

The RHS then simplifies to

$$\left[1 - \frac{1}{1+e^{-z}}\right]$$

Which is nothing more than

$$(1 - sig(z))$$

1 minus our sigmoid

Which gives a final result of

$$sig(z) \cdot (1 - sig(z))$$

Or alternatively:

$$a(1 - a)$$

This notation will be easier

[3] Derivative w.r.t linear function

$$\frac{\partial L}{\partial z}$$

[3] derivative of our linear function (z = wX + b)

To get this result we can use chain rule by multiplying the two results we’ve already calculated [1] and [2]

$$\frac{\partial L}{\partial a} * \frac{\partial a}{\partial z}$$

multiply derivative [1] by derivative [2]

$$\left[y \quad (1 - y)\right]$$

$$-\left[\frac{}{a} - \frac{}{(1-a)}\right] \cdot a(1-a)$$

der[1] x der[2]

So if we can get a common denominator in the left hand of the equation, then we can simplify the equation, so lets add '(1-a)' to the first fraction and 'a' to the second fraction

$$-\left[\frac{y(1-a)}{a(1-a)} - \frac{(a)(1-y)}{(a)(1-a)}\right] \cdot a(1-a)$$

add '(1–a)' and 'a' to get common denominator

with a common denominator we can simplify to

$$-\left[\frac{y(1-a)+(a)(1-y)}{a(1-a)}\right] \cdot a(1-a)$$

common denominator

now we multiply LHS by RHS, the a(1-a) terms cancel out and we are left with just the numerator from the LHS!

$$-y(1-a)+(a)(1-y)$$

the remaining numerator

which if we expand out gives:

$$-y+ya+a-ay$$

expanded out

note that 'ya' is the same as 'ay', so they cancel to give

$$-y+a$$

which rearranges to give our final result of the derivative

$$\frac{\partial L}{\partial z}$$

[3]

our final result is

$$a-y$$

derivative of our linear function (z = wX +b)

[4] Derivative w.r.t weights

$$\frac{\partial z}{\partial w}$$

[4] derivative of linear func ‘z’ w.r.t weights ‘w’

This derivative is trivial to compute, as z is simply

$$W^T X + b$$

linear function ‘z’

and the derivative simply evaluates to

$$x$$

derivative of ‘z’ w.r.t ‘w’

[5] Derivative w.r.t weights (2)

$$\frac{\partial L}{\partial w}$$

[5] derivative of cost func w.r.t weights ‘w’

This derivative can be computed **two different ways!** We can use **chain rule** or **compute directly**. We will do both as it provides a great intuition behind backprop calculation.

To use chain rule to get derivative [5] we note that we have already computed the following

$$\frac{\partial L}{\partial a} \quad \frac{\partial a}{\partial z} \quad \frac{\partial z}{\partial w}$$

previously computed

Noting that the product of the first two equations gives us

$$\frac{\partial L}{\partial z}$$

$$\partial z$$

if we then continue using the chain rule and multiply this result by

$$\frac{\partial z}{\partial w}$$

then we get

$$\frac{\partial L}{\partial z} \cdot x$$

which is nothing more than

$$x(a - y)$$

The final result for ‘dw’

or written out long hand

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w}$$

chain rule result for ‘dw’

So that’s the ‘*chain rule way*’. Now lets compute ‘dw’ *directly*:

To compute **directly**, we first take our cost function

$$-\sum[y ln(a) + (1 - y) ln(1 - a)]$$

Cross Entropy cost function

We can notice that the first log term ‘ln(a)’ can be expanded to

$$ln(1 + e^{-z})^{-1}$$

expanding ‘ln(a)’

Which simplifies to:

$$-ln(1 + e^{-z})$$

And if we take the second log function ‘ln(1-a)’ which can be shown as

$$e^{-z}$$

$$ln\frac{e}{(1+e^{-z})}$$

ln(1-a)

taking the log of the numerator (we will leave the denominator) we get

$$(-z - ln(1+e^{-z}))$$

log of the numerator

This result comes from the rule of logs, which states: log(p/q) = log(p)—log(q).

Plugging these formula back into our original cost function we get

$$\sum -yln(1+e^{-z})\left[(1-y)(-z-ln(1+e^{-z}))\right]$$

plugged back into cost function

Expanding the term in the square brackets we get

$$\sum -yln(1+e^{-z}) - z - ln(1+e^{-z}) + yz + yln(1+e^{-z})$$

terms inside bracket expanded

The first and last terms ‘yln(1+e^-z)’ cancel out leaving:

$$-z - ln(1+e^{-z}) + yz$$

Which we can rearrange by pulling the ‘yz’ term to the outside to give

$$yz - \left[z + ln(1+e^{-z})\right]$$

Here’s where it gets interesting, by adding an exp term to the ‘z’ inside the square brackets and then immediately taking its log

$$yz - \left[ln(e^{z}) + ln(1+e^{-z})\right]$$

we exponentiate ‘e^z’ then take its log

next we can take advantage of the rule of sum of logs: ln(a) + ln(b) = ln(a.b) combined with rule of exp products:e^a * e^b = e^(a+b) to get

$$yz - [ln((e^{z})\cdot(1+e^{-z}))]$$

ln(a) + ln(b) = ln(a.b)

followed by

$$yz - [ln(1 + e^z)]$$

add e^(z +-z)

Pulling the ‘yz’ term inside the brackets we get :

$$[yz - ln(1 + e^z)]$$

Finally we note that z = Wx+b therefore taking the derivative w.r.t W:

$$\frac{\partial L}{\partial w}[yz - ln(1 + e^z)]$$

take derivative w.r.t W

The first term ‘yz ’becomes ‘yx ’and the second term becomes :

$$yx - \frac{x \cdot e^{wx}}{1+e^{wx}}$$

taking derivative of logs again

Note that the 2nd term is nothing but

$$x \cdot sig(z)$$

Which gives a final result of

$$-[yx - x \cdot sig(z)]$$

We can rearrange by pulling ‘x’ out to give

$$-x[y - sig(z)]$$

which gives

$$\frac{\partial L}{\partial w} = x[a - y]$$

final result

[6] derivative w.r.t bias

$$\frac{\partial L}{\partial b}$$

Again we could use **chain rule** which would be

$$\frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial b}$$

chain rule for ‘db’

This is easy to solve as we already computed ‘dz’ and the second term is simply the derivative of ‘z’ which is ‘wX +b’ w.r.t ‘b’ which is simply 1!

so the derivative w.r.t b is simply

$$\frac{\partial L}{\partial z}$$

which we already calculated earlier as

$$a - y$$

derivative of our linear function (z = wX +b)

For completeness we will also show how to calculate ‘db’ **directly**. To calculate this we will take a step from the above calculation for ‘dw’, (from just before we did the differentiation)

$$[yz - ln(1 + e^z)]$$

note: z = wX + b

remembering that z = wX +b and we are trying to find derivative of the function w.r.t b, if we take the derivative w.r.t b from both terms ‘yz’ and ‘ln(1+e^z)’ we get

$$\frac{\partial L}{\partial b} y(wX + b) - \frac{\partial L}{\partial b} ln(1 + e^{wX+b})$$

note the parenthesis

its important to note the parenthesis here, as it clarifies how we get our derivative.

Taking the LHS first, the derivative of ‘wX’ w.r.t ‘b’ is zero as it doesn’t contain b! The derivative of ‘b’ is simply 1, so we are just left with the ‘y’ outside the parenthesis.

for the RHS, we do the same as we did when calculating ‘dw’, except this time when taking derivative of the inner function ‘e^wX+b’ we take it w.r.t ‘b’ (instead of ‘w’) which gives the following result (this is because the derivative w.r.t in the exponent evaluates to 1)

$$\frac{e^{wX+b}}{1+e^{wX+b}}$$

derivative of ln(1+e^wX+b)

this term is simply our original

$$a$$

so putting the whole thing together we get

$$[a-y]$$

final result

which we have already show is simply ‘dz’!

$$\frac{\partial L}{\partial z}$$

‘db’ = ‘dz’

. . .

So that concludes all the derivatives of our Neural Network. **We have calculated all of the following:**

$$\frac{\partial L}{\partial a}\frac{\partial L}{\partial z}\frac{\partial L}{\partial w}\frac{\partial L}{\partial b}$$

The derivatives of L(a,y) w.r.t each element in our NN

. . .

Wrapping up

And what about the result:

$$\frac{\partial L^{[L-1]}}{\partial Z} = dW^{[L]T} \cdot dZ^{[L]} * g'^{[L]}(Z^{[L-1]})$$

well, we can unpack the chain rule to explain:

$$= \frac{\partial L}{\partial A^{[L]}} \cdot \frac{\partial A^{[L]}}{\partial Z^{[L]}} \cdot \frac{\partial Z^{[L]}}{\partial A^{[L-1]}} \cdot \frac{\partial A^{[L-1]}}{\partial Z^{[L-1]}}$$

‘dz’ using chain rule

Note that the term

$$\frac{\partial L}{\partial A^{[L]}} \cdot \frac{\partial A^{[L]}}{\partial Z^{[L]}}$$

is simply ‘dz’ the term we calculated earlier:

$$\frac{\partial L}{\partial z}$$

and the term

$$\frac{\partial Z^{[L]}}{\partial A^{[L-1]}}$$

evaluates to W[l] or in other words, the derivative of our linear function Z =’Wa +b’ w.r.t ‘a’ equals ‘W’.

and finally the term in blue

$$\frac{\partial A^{[L-1]}}{\partial Z^{[L-1]}}$$

is simply

$$\frac{\partial a}{\partial z}$$

[2] derivative of sigmoid

‘da/dz’ the derivative of the the sigmoid function that we calculated earlier!

As a final note on the notation used in the Coursera Deep Learning course, in the result

$$\frac{\partial L^{[L-1]}}{\partial Z} = \quad W^{[L]T} \cdot dZ^{[L]} * g'^{[L]}(Z^{[L-1]})$$

we perform element wise multiplication between DZ and g’(Z), this is to ensure that all the dimensions of our matrix multiplications match up as expected.

So there we have it…

… all the derivatives required for backprop as shown in Andrew Ng’s Deep Learning course.

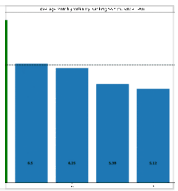
Simply reading through these calculus calculations (*or any others for that matter*) won’t be enough to make it stick in your mind. The best way to learn is to lock yourself in a

room and **practice, practice, practice!**

What next?

> *If you got something out of this post, please share with others who may benefit, follow me Patrick David for more ML posts or on twitter @pdquant and give it a cynical/pity/genuine round of **applause**!*

Stocks Significance Testing & p-Hacking



Stocks, Significance Testing & p-Hacking: How volatile is volatile?

October is historically the most volatile month for stocks, but is this a persistent signal or just noise in the data?

medium.com

Build a Bit(Cointegration) Backtester

Build a BitCoin(tegration) Backtester

Learn the statistical technique of Cointegration and build your own crypto backtester to create and test a quantitative…

medium.com

Machine Learning | Deep Learning | Neural Networks | Calculus | AI

774 claps

10

Patrick David

Machine Learning , Deep Learning, AI, Python, Quant Trading -ML Researcher - follow for walkthroughs, tutorials, proofs, research etc. Also on twitter @pdquant

Follow