Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

Fall 2022

X.19 The loss derivative

Finally, we apply this to our common derivatives in section 7.5.

$$\overbrace{\frac{\partial \mathcal{L}}{\partial A^L}}^{(n^L \times 1)} \tag{1}$$

Loss is not given, so we can't compute it. But, we can get the shape: we have a scalar/vector derivative, so the shape matches A^L.

Notation 1

Our derivative

$$\frac{\partial \mathcal{L}}{\partial A^L} \tag{2}$$

Is a scalar/vector derivative, and thus the shape $(n^L \times 1)$.

X.20 The weight derivative

$$\overbrace{\frac{\partial Z^\ell}{\partial W^\ell}}^{(m^\ell \times 1)?} \tag{3}$$

This derivative is difficult - it's a derivative in the form vector/matrix. With **three** axes, we might imagine representing as a 3-tensor.

In fact, this can be manipulated into multiple different interesting **shapes** based on your **interpretation**: as we mentioned, there's no consistent rule for these variables.

But, our goal is to use this for the **chain rule**: so, we need to make the shapes **match**. This is why we do that strange transposing for our complete derivative.

$$\frac{\partial \mathcal{L}}{\partial W^\ell} = \overbrace{\frac{\partial Z^\ell}{\partial W^\ell}}^{\text{Weight link}} \cdot \overbrace{\left(\frac{\partial \mathcal{L}}{\partial Z^\ell}\right)^T}^{\text{Other layers}} \tag{4}$$

Our problem is we have **too many axes**: the easiest way to resolve this to **break up** our matrix. So, for now, we focus on only **one neuron** at a time: it has a column vector W_i.

For simplicity, we're gonna ignore the ℓ notation: just be careful, because Z and A are from two different layers!

$$W = \begin{bmatrix} W_1 & W_2 & \cdots & W_n \end{bmatrix} \tag{5}$$

Notice that, this time, we broke it into **column vectors**, rather than row vectors: each neuron's **weights** are represented by a column vector.

We'll ignore everything except W_i.

$$W_i = \begin{bmatrix} w_{1i} \\ w_{2i} \\ \vdots \\ w_{mi} \end{bmatrix} \tag{6}$$

Finally, we get into our equation: notice that a **single** neuron has only **one** pre-activation z_i, so we don't need the whole vector.

$$z_i = W_i^T A \tag{7}$$

Wait: there's something to notice, right off the bat. z_i is **only** a function of W_i: that means the derivative for every other term $\partial/\partial W_k$ is **zero**!

For example, changing W_2 would have **no** effect on z_1.

Concept 2

The i^{th} neuron's **weights**, W_i, have **no effect** on a different neuron's **pre-activation** z_j.

So, if the **neurons** don't match, then our derivative is zero:

- i is the neuron for pre-activation z_i
- j is the j^{th} weight in a neuron.
- k is the neuron for weight vector W_k

$$\frac{\partial z_i}{\partial W_{jk}} = 0 \qquad \text{if } i \neq k$$

So, our only nonzero derivatives are

$$\frac{\partial z_i}{\partial W_{ji}}$$

With that done, let's substitute in our values:

$$z_i = \begin{bmatrix} w_{1i} & w_{2i} & \cdots & w_{mi} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \tag{8}$$

And we'll do our **matrix multiplication**:

$$z_i = \sum_{j=1}^{n} W_{ji} a_j \tag{9}$$

Finally, we can get our derivatives:

$$\frac{\partial z_i}{\partial W_{ji}} = a_j \tag{10}$$

So, if we combine that into a vector, we get:

$$\frac{\partial z_i}{\partial W_i} = \begin{bmatrix} \frac{\partial z_i}{\partial W_{1i}} \\ \frac{\partial z_i}{\partial W_{2i}} \\ \vdots \\ \frac{\partial z_i}{\partial W_{mi}} \end{bmatrix} \tag{11}$$

We can use our equation:

$$\frac{\partial z_i}{\partial W_i} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = A \tag{12}$$

We get a result!

What if the pre-activation z_i and weights W_k don't match? We've already seen: the derivative is 0: weights don't affect different neurons.

$$\frac{\partial z_i}{\partial W_{jk}} = 0 \qquad \text{if } i \neq k \tag{13}$$

We can combine these into a **zero vector**:

$$\frac{\partial z_i}{\partial W_k} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \vec{0} \qquad \text{if } i \neq k \tag{14}$$

So, now, we can describe all of our vector components:

$$\frac{\partial z_i}{\partial W_k} = \begin{cases} A & \text{if } i = k \\ \vec{0} & \text{if } i \neq k \end{cases} \tag{15}$$

These are all the elements of our matrix $\partial z_i / \partial W_k$: so, we can get our result.

$$\frac{\partial Z}{\partial W} = \begin{bmatrix} A & \vec{0} & \cdots & \vec{0} \\ \vec{0} & A & \cdots & \vec{0} \\ \vdots & \vdots & \ddots & \vec{0} \\ \vec{0} & \vec{0} & \vec{0} & A \end{bmatrix} \tag{16}$$

We have our result: it turns out, despite being stored in a **matrix**-like format, this is actually a **3-tensor**! Each entry of our **matrix** is a **vector**: 3 axes.

But, we don't really... *want* a tensor. It doesn't have the right shape, and we can't do matrix multiplication.

We'll solve this by **simplifying**, without losing key information.

Concept 3

For many of our "tensors" resulting from matrix derivatives, they contain **empty** rows or **redundant** information.

Based on this, we can **simplify** our tensor into a fewer-dimensional (fewer axes) object.

We can see two types of **redundancy** above:

- Every element **off** the diagonal is 0.
- Every element **on** the diagonal is the same.

Let's fix the first one: we'll go from a diagonal matrix to a column vector.

$$\begin{bmatrix} A & \vec{0} & \cdots & \vec{0} \\ \vec{0} & A & \cdots & \vec{0} \\ \vdots & \vdots & \ddots & \vec{0} \\ \vec{0} & \vec{0} & \vec{0} & A \end{bmatrix} \longrightarrow \begin{bmatrix} A \\ A \\ \vdots \\ A \end{bmatrix} \tag{17}$$

Then, we'll combine all of our redundant A values.

$$\begin{bmatrix} A \\ A \\ \vdots \\ A \end{bmatrix} \longrightarrow A \tag{18}$$

We have our big result!

Notation 4

Our derivative

$$\overbrace{\frac{\partial Z^\ell}{\partial W^\ell}}^{(m^\ell \times 1)} = A^{\ell-1}$$

Is a vector/matrix derivative, and thus should be a 3-tensor.

But, we have turned it into the shape $(m^\ell \times 1)$.

This is as **condensed** as we can get our information: if we compress to a scalar, we lose some of our elements.

Even with this derivative, we still have to do some clever **reshaping** to get the result we need (transposing, changing derivative order, etc.)

However, at the end, we get the right shape for our chain rule!

X.21 Linking Layers

$$\frac{\partial Z^\ell}{\partial A^{\ell-1}} \tag{19}$$

This derivative is much more manageable: it's just the derivative between a vector and a vector. Let's look at our equation again:

Ignoring superscripts ℓ, as before.

$$Z = W^T A \tag{20}$$

We'll use the same approach we did last time: W is a vector, and we'll focus on W_i. This will allow us to break it up **element-wise**, and get all of our **derivatives**.

We could treat W as a whole matrix, but this will give us our results without as much clutter: the only **difference** is that we would have to depict every W_i at **once**.

$$W = \begin{bmatrix} W_1 & W_2 & \cdots & W_n \end{bmatrix} \qquad W_i = \begin{bmatrix} w_{1i} \\ w_{2i} \\ \vdots \\ w_{mi} \end{bmatrix} \tag{21}$$

Here's our equation:

$$z_i = \begin{bmatrix} w_{1i} & w_{2i} & \cdots & w_{mi} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \tag{22}$$

We matrix multiply:

$$z_i = \sum_{j=1}^{n} W_{ji} a_j \tag{23}$$

The derivative can be gotten from here -

$$\frac{\partial z_i}{\partial a_j} = W_{ji} \tag{24}$$

We look at our whole matrix derivative:

This notation looks a bit weird, but it's just a way to represent that all of our elements follow this pattern.

$$\frac{\partial Z}{\partial A} = \left.\overbrace{\begin{bmatrix} \ddots & \vdots & \iddots \\ \cdots & \frac{\partial z_i}{\partial a_j} & \cdots \\ \iddots & \vdots & \ddots \end{bmatrix}}^{\text{Column } j \text{ matches } z_i}\right\} \text{Row } i \text{ matches } a_j \tag{25}$$

Wait.

- The derivative $\partial z_i / \partial a_j$ is in the j^{th} row, i^{th} column.
- W_{ji} represents the element in the j^{th} row, i^{th} column.

They're the same matrix!

If two matrices have exactly the same shape and elements, they're the same matrix.

We get our final result:

Notation 5

Our derivative

$$\overbrace{\frac{\partial Z^{\ell}}{\partial A^{\ell-1}}}^{(m^{\ell} \times n^{\ell})} = W^{\ell}$$

Is a vector/vector derivative, and thus a matrix.

But, we have turned it into the shape $(m^{\ell} \times n^{\ell})$.

X.22 Activation Function

$$\frac{\partial A^\ell}{\partial Z^\ell} \tag{26}$$

The last derivative is less unusual than it looks.

$$A^\ell = f(Z^\ell) \longrightarrow \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = f\left(\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \right) \tag{27}$$

We can apply our function element-wise:

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(z_1) \\ f(z_2) \\ \vdots \\ f(z_n) \end{bmatrix} \tag{28}$$

As we can see, each activation is a function of only **one** pre-activation.

Concept 6

Each **activation** is only affected by the **pre-activation** in the **same neuron**.

So, if the **neurons** don't match, then our derivative is zero:

- i is the neuron for pre-activation z_i
- j is the neuron for activation a_j

$$\frac{\partial a_j}{\partial z_i} = 0 \qquad \text{if } i \neq j$$

So, our only nonzero derivatives are

$$\frac{\partial a_j}{\partial z_i}$$

As for our remaining term, we'll describe any row of the above vectors:

$$a_i = f(z_i) \tag{29}$$

Our derivative is:

$$\frac{\partial a_i}{\partial z_i} = f'(z_i) \tag{30}$$

In general, including the non-diagonals:

$$\frac{\partial a_i}{\partial z_i} = \begin{cases} f'(z_i) & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{31}$$

This gives us our result:

Notation 7

Our derivative

$$\overbrace{\frac{\partial A^\ell}{\partial Z^\ell}}^{(n^\ell \times n^\ell)} = \begin{array}{c} \text{Column } j \text{ matches } a_j \\ \overbrace{\begin{bmatrix} f'(z_1^\ell) & 0 & 0 & \cdots & 0 \\ 0 & f'(z_2^\ell) & 0 & \cdots & 0 \\ 0 & 0 & f'(z_3^\ell) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & f'(z_n^\ell) \end{bmatrix}} \end{array} \left.\vphantom{\begin{bmatrix} a \\ a \\ a \\ a \\ a \end{bmatrix}}\right\} \text{Row } i \text{ matches } z_i \tag{32}$$

Is a vector/vector derivative, and thus a matrix.

But, we have turned it into the shape $(n^\ell \times n^\ell)$.

X.23 Element-wise multiplication

Notice that, in the previous section, we would've compressed this matrix down to remove the unnecessary 0's:

$$\begin{bmatrix} f'(z_1^\ell) \\ f'(z_2^\ell) \\ \vdots \\ f'(z_n^\ell) \end{bmatrix} \tag{33}$$

This is a valid way to interpret this matrix! The only thing we need to be careful of: if we were to use this in a chain rule, we couldn't do normal matrix multiplication.

However, because of how this matrix works, you can just do **element-wise** multiplication instead!

You can check it for yourself: each index is separately scaled.

Concept 8

When multiplying two vectors R and Q, if they take the form

$$R = \begin{bmatrix} r_1 & 0 & 0 & \cdots & 0 \\ 0 & r_2 & 0 & \cdots & 0 \\ 0 & 0 & r_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & r_n \end{bmatrix} \qquad Q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_n \end{bmatrix}$$

Then we can write their product each of these ways:

$$RQ = \overbrace{R * Q}^{\text{Element-wise multiplication}} = \begin{bmatrix} r_1 q_1 \\ r_2 q_2 \\ r_3 q_3 \\ \vdots \\ r_n q_n \end{bmatrix} \tag{34}$$

So, we can substitute the chain rule this way.