Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

Fall 2022

7.X.20 The weight derivative

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

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.

$$\frac{\partial \mathcal{L}}{\partial W^{\ell}} = \overbrace{\frac{\partial Z^{\ell}}{\partial W^{\ell}}}^{\text{Weight link}} \cdot \overbrace{\left(\frac{\partial \mathcal{L}}{\partial Z^{\ell}}\right)^{T}}^{\text{Other layers}} \tag{2}$$

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!

$$W = \begin{bmatrix} W_1 & W_2 & \cdots & W_n \end{bmatrix} \tag{3}$$

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.

$$W_i = \begin{bmatrix} w_{1i} \\ w_{2i} \\ \vdots \\ w_{mi} \end{bmatrix} \tag{4}$$

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.

$$z_i = W_i^T A \tag{5}$$

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 1

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:

$$z_i = \begin{bmatrix} w_{1i} & w_{2i} & \cdots & w_{mi} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \tag{6}$$

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

$$z_i = \sum_{j=1}^{n} W_{ji} a_j \tag{7}$$

Finally, we can get our derivatives:

$$\frac{\partial z_i}{\partial W_{ji}} = a_j \tag{8}$$

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

$$\frac{\partial z_i}{\partial W_i} = \begin{bmatrix} \frac{\partial z_i}{\partial W_{1i}} \\ \frac{\partial z_i}{\partial W_{2i}} \\ \vdots \\ \frac{\partial z_i}{\partial W_{mi}} \end{bmatrix} \tag{9}$$

We can use our equation:

$$\frac{\partial z_i}{\partial W_i} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = A \tag{10}$$

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.

$$\frac{\partial z_i}{\partial W_{jk}} = 0 \qquad \text{if } i \neq k \tag{11}$$

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

$$\frac{\partial z_i}{\partial W_k} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \vec{0} \qquad \text{if } i \neq k \tag{12}$$

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

$$\frac{\partial z_i}{\partial W_k} = \begin{cases} A & \text{if } i = k \\ \vec{0} & \text{if } i \neq k \end{cases} \tag{13}$$

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

$$\frac{\partial Z}{\partial W} = \begin{bmatrix} A & \vec{0} & \cdots & \vec{0} \\ \vec{0} & A & \cdots & \vec{0} \\ \vdots & \vdots & \ddots & \vec{0} \\ \vec{0} & \vec{0} & \vec{0} & A \end{bmatrix} \tag{14}$$

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 2

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.

$$\begin{bmatrix} A & \vec{0} & \cdots & \vec{0} \\ \vec{0} & A & \cdots & \vec{0} \\ \vdots & \vdots & \ddots & \vec{0} \\ \vec{0} & \vec{0} & \vec{0} & A \end{bmatrix} \longrightarrow \begin{bmatrix} A \\ A \\ \vdots \\ A \end{bmatrix} \tag{15}$$

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

$$\begin{bmatrix} A \\ A \\ \vdots \\ A \end{bmatrix} \longrightarrow A \tag{16}$$

We have our big result!

Notation 3

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!