Why is $\frac{\partial J(W)}{\partial W_{i,j}^{(out)}} = (A^{(h)})^T \delta^{(out)}$?

Kassi Bertrand

January 2023

1 Introduction

In this LaTeX document, I want to show my step-by-step process to derive the partial derivative of all the weights in $W^{(out)}$.

I'll start simple with a 2-2-2 MLP, which is a Multi-Layer Perceptron with 2 inputs, 2 hidden units, and 2 outputs, and then generalize what we learn to a general $m - d - t$ MLP.

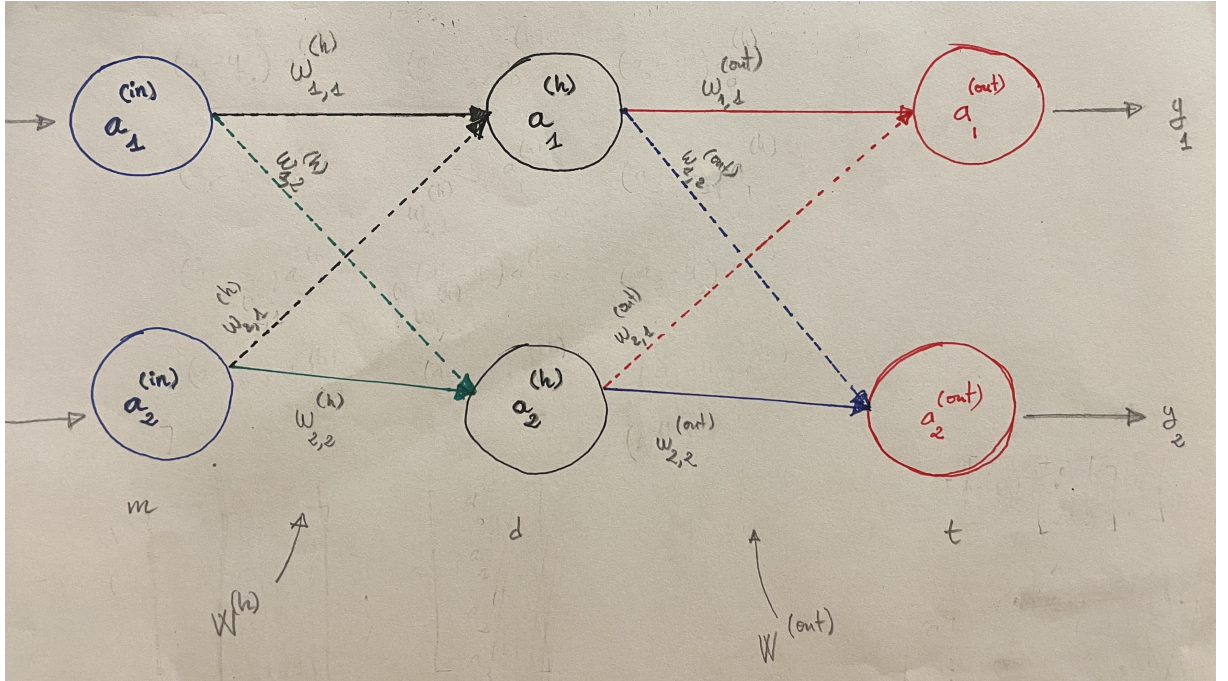In both cases, I will use the following loss/error function:

$$J(w) = -\sum_{i=1}^{n} \sum_{j=1}^{t} y_j^{[i]} ln(a_j^{[i]}) + (1 - y_j^{[i]}) ln(1 - a_j^{[i]})$$

Where the superscript $[i]$ is an index for training examples, and j is the number of output units.

Ready? Let's go!

2 With a 2-2-2 MLP

For this example, we will consider the following Neural network:

We'll ignore biase units in the input and hidden layers, and consider only ONE training example for simplicity purposes.

Since one training example is considered and the network has two output units, then $n = 1$ and $t = 2$. So, the loss function becomes like this:

$$J(w) = -\sum_{j=1}^{t} y_j^{[1]} ln(a_j^{[1]}) + (1 - y_j^{[1]}) ln(1 - a_j^{[1]})$$

The journey of a SINGLE training example from the input layer to the output layer of our network goes like this:

$$[a_1^{(in)}, a_2^{(in)}]$$

$$\downarrow$$

$$W^{(h)}$$

$$\downarrow$$

$$[z_1^{(h)}, z_2^{(h)}]$$

$$\downarrow$$

$$\phi(\bullet)$$

$$\downarrow$$

$$[a_1^{(h)}, a_2^{(h)}]$$

$$\downarrow$$

$$W^{(out)}$$

$$\downarrow$$

$$[z_1^{(out)}, z_2^{(out)}]$$
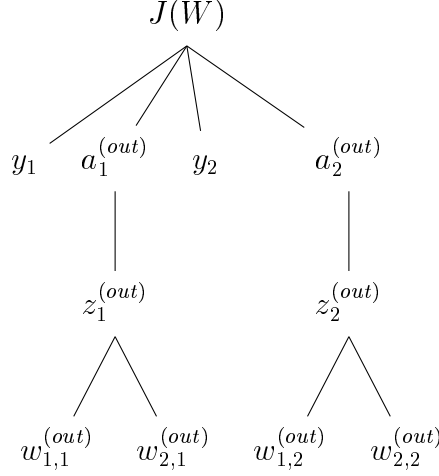
$$\downarrow$$

$$\phi(\bullet)$$

$$\downarrow$$

$$[a_1^{(out)}, a_2^{(out)}]$$

$$\downarrow$$

$$[y_1, y_2]$$

Now, let's compute the derivative of each weights in the $W^{(out)}$ matrix. To help myself, I drew the following tree diagram to see how the variables in $J(W)$ relate to the weights in $W^{(out)}$:

$$J(W)$$

$$y_1 \qquad a_1^{(out)} \qquad y_2 \qquad\qquad a_2^{(out)}$$

$$z_1^{(out)} \qquad\qquad z_2^{(out)}$$

$$w_{1,1}^{(out)} \qquad w_{2,1}^{(out)} \qquad w_{1,2}^{(out)} \qquad w_{2,2}^{(out)}$$

Our MLP has two outputs. If you look at $J(W)$ expression closely, we compute the cost for each output, then add the results together. And that's how we get the cost for the current training example. With that in mind, we can now compute the partial derivatives:

$$\frac{\partial J(W)}{\partial w_{1,1}^{(out)}} = \frac{\partial J(W)}{\partial a_1^{(out)}} \times \frac{\partial a_1^{(out)}}{\partial z_1^{(out)}} \times \frac{\partial z_1^{(out)}}{\partial w_{1,1}^{(out)}}$$

$$\frac{\partial J(W)}{\partial w_{2,1}^{(out)}} = \frac{\partial J(W)}{\partial a_1^{(out)}} \times \frac{\partial a_1^{(out)}}{\partial z_1^{(out)}} \times \frac{\partial z_1^{(out)}}{\partial w_{2,1}^{(out)}}$$

$$\frac{\partial J(W)}{\partial w_{1,2}^{(out)}} = \frac{\partial J(W)}{\partial a_2^{(out)}} \times \frac{\partial a_2^{(out)}}{\partial z_2^{(out)}} \times \frac{\partial z_2^{(out)}}{\partial w_{1,2}^{(out)}}$$

$$\frac{\partial J(W)}{\partial w_{2,2}^{(out)}} = \frac{\partial J(W)}{\partial a_2^{(out)}} \times \frac{\partial a_2^{(out)}}{\partial z_2^{(out)}} \times \frac{\partial z_2^{(out)}}{\partial w_{2,2}^{(out)}}$$

Notice, the following expressions are common in the first two and last two equations. Let's evaluate them:

$$\frac{\partial J(W)}{\partial a_1^{(out)}} \times \frac{\partial a_1^{(out)}}{\partial z_1^{(out)}} = (a_1^{(out)} - y_1) = \delta_1^{(out)}$$

$$\frac{\partial J(W)}{\partial a_2^{(out)}} \times \frac{\partial a_2^{(out)}}{\partial z_2^{(out)}} = (a_2^{(out)} - y_2) = \delta_2^{(out)}$$

$(a_1^{(out)} - y_1)$ and $(a_2^{(out)} - y_2)$ are the "**error**" terms in the output layer. Since our MLP has two outputs, we have two "error" terms as well. We use "δ" to denote that "error". $\delta_1^{(out)}$ for instance, is the error in the first activation unit in the output layer.

With this new knowledge, we can re-write the partial derivatives:

$$\frac{\partial J(W)}{\partial w_{1,1}^{(out)}} = \delta_1^{(out)} a_1^{(h)}$$

$$\frac{\partial J(W)}{\partial w_{2,1}^{(out)}} = \delta_1^{(out)} a_2^{(h)}$$

$$\frac{\partial J(W)}{\partial w_{1,2}^{(out)}} = \delta_2^{(out)} a_1^{(h)}$$

$$\frac{\partial J(W)}{\partial w_{2,2}^{(out)}} = \delta_2^{(out)} a_2^{(h)}$$

From the above, I can now introduce the $\delta^{(out)}$ matrix. It is a $n \times t$ matrix; where n is the number of training examples and t the number of activation units in the output layer. Since we are dealing with <u>ONE</u> example and our MLP has <u>2</u> outputs, so the $\delta^{(out)}$ matrix is of the shape (1×2), and looks like this:

$$\delta^{(out)} = [\delta_1^{(out)}, \delta_2^{(out)}] = [(a_1^{(out)} - y_1), (a_2^{(out)} - y_2)]$$

Also, remember the $A^{(h)}$ matrix, obtained after training examples are forward propagated from the input to the hidden layer. Since we have <u>ONE</u> training example and $\underline{2}$ units in the hidden layer. So $A^{(h)}$, is also of the shape (1×2), and looks like this:

$$A^{(h)} = [a_1^{[1]}, a_2^{[1]}]$$

3   With a general $m - d - t$ MLP