Solutions 16: Training Neural Networks

Exercises

Assume that we have a neural network with one hidden layer, defined as (where x is one row of the input matrix and y is the corresponding output value):

$$z_k = \alpha_k + \sum_{j=1}^{P} B_{j,k} \cdot x_j \tag{11.1}$$

$$a_k = \sigma(z_k) \tag{11.2}$$

$$w = c + \sum_{k} \gamma_k a_k \tag{11.3}$$

Where σ is a differentiable activation function. The terms c, γ_k, and $B_{j,k}$ are the parameters that define the model. We want to minimize the quantity (the loss function):

$$L(w, y) = \frac{1}{2} \cdot (w - y)^2. \tag{11.4}$$

We need to compute a number of partial derivatives, which we will do using the chain rule. It is important that you don't jump ahead and plug things in before I ask you to.

Step 1: Compute the partial derivative of:

$$\frac{\partial z_k}{\partial B_{j,k}} = x_j$$

Note that z_k does not depend on $B_{j,m}$ if $m \neq k$, so we do not need to worry about those terms.

Step 2: Compute the partial derivative of (yes, this is easy):

$$\frac{\partial z_k}{\partial \alpha_k} = 1$$

Step 3: Write down a formula for the following using the notation $\sigma'(\cdot)$ to denote the derivative of σ.

$$\frac{\partial a_k}{\partial z_k} = \sigma'(z_k)$$

Step 4: Write down a formula for:

$$\frac{\partial w}{\partial \gamma_k} = a_k$$

Step 5: What is the following (yes, this is easy also):

$$\frac{\partial w}{\partial c} = 1$$

Step 6: Finally, what is the derivative of the loss function with respect to w:

$$\frac{\partial L}{\partial w} = (w - y)$$

Step 7: Notice that I can use the chain rule to write the following, the derivative with respect to each tunable parameter in the second layer of the model:

$$\frac{\partial L}{\partial c} = \frac{\partial L}{\partial w} \cdot \frac{\partial w}{\partial c} \tag{11.5}$$

$$\frac{\partial L}{\partial \gamma_k} = \frac{\partial L}{\partial w} \cdot \frac{\partial w}{\partial \gamma_k} \tag{11.6}$$

Now, plug in the values that you know to compute each of these:

$$\frac{\partial L}{\partial c} = (w - y) \cdot 1 = w - y$$

$$\frac{\partial L}{\partial \gamma_k} = (w - y) \cdot a_k$$

Step 8: What about the terms $B_{j,k}$ and α_k? They are in the hidden layer and require one more step:

$$\frac{\partial L}{\partial B_{j,k}} = \frac{\partial L}{\partial z_k} \cdot \frac{\partial z_k}{\partial B_{j,k}} \tag{11.7}$$

$$= \frac{\partial L}{\partial w} \cdot \frac{\partial w}{\partial a_k} \cdot \frac{\partial a_k}{\partial z_k} \cdot \frac{\partial z_k}{\partial B_{j,k}} \tag{11.8}$$

And:

$$\frac{\partial L}{\partial \alpha_k} = \frac{\partial L}{\partial z_k} \cdot \frac{\partial z_k}{\partial B_{j,k}} \tag{11.9}$$

$$= \frac{\partial L}{\partial w} \cdot \frac{\partial w}{\partial a_k} \cdot \frac{\partial a_k}{\partial z_k} \cdot \frac{\partial z_k}{\partial \alpha_k} \tag{11.10}$$

But, you do know all of these terms. Plug them in to get the partial derivative with respect to $B_{j,k}$ and α_k.

$$\frac{\partial L}{\partial B_{j,k}} = (w - y) \cdot \gamma_k \cdot \sigma'(z_k) \cdot x_j$$

$$\frac{\partial L}{\partial \alpha_k} = (w - y) \cdot \gamma_k \cdot \sigma'(z_k)$$

Summary

The most important lines to understand here are Equations 11.7 and 11.9. They show the core back propagation logic: decomposing the influence of a parameter to (i) how it influences the output of that layer and (ii) how that layer influences the loss. This make it possible, with just a little bit more notation, to compute gradients for the deepest of neural networks.