Backpropagation

Peder Nørving Viken

February 2020

1 Backpropagation

Backpropagation is the goto method for evaluating the gradient of the loss function in a neural network [LBH15]. The derivation in this document is for a fully connected, feed forward neural network such as in figure 1. This was inpsired by the video series [Dee].

1.1 Deriving the backpropagation algorithm

First a few notes on the notation used here.

- C_0 is the loss function, e.g. L_2-loss.
- $a_j^{(L)}$ is the output of node j in layer L, where layer (L) is the output layer.
- $w_{jk}^{(L)}$ is the weight from node k in layer $(L-1)$ to node j in layer (L).
- $g()^{(L)}$ is the activation function in layer (L).
- $z_j^{(L)}$ is the weighted sum defined as $z_j^{(L)} = \sum_i w_{ji}^{(L)} a_i^{(L)}$, and $a_j^{(L)} = g(z_j^{(L)})$

1.1.1 The derivative w.r.t. a weight between layer (L-1) and (L).

Here the weight only effects the final (output) layer. We also only take into account the effect of one training sample. Consider the weight e.g. $w_{21}^{(L)}$.

$$\frac{\partial C_0}{\partial w_{21}^{(L)}} = \frac{\partial C_0}{\partial a_2^{(L)}} \frac{\partial a_2^{(L)}}{\partial z_2^{(L)}} \frac{\partial z_2^{(L)}}{\partial w_{21}^{(L)}} \tag{1}$$

In equation 1, we have the following terms

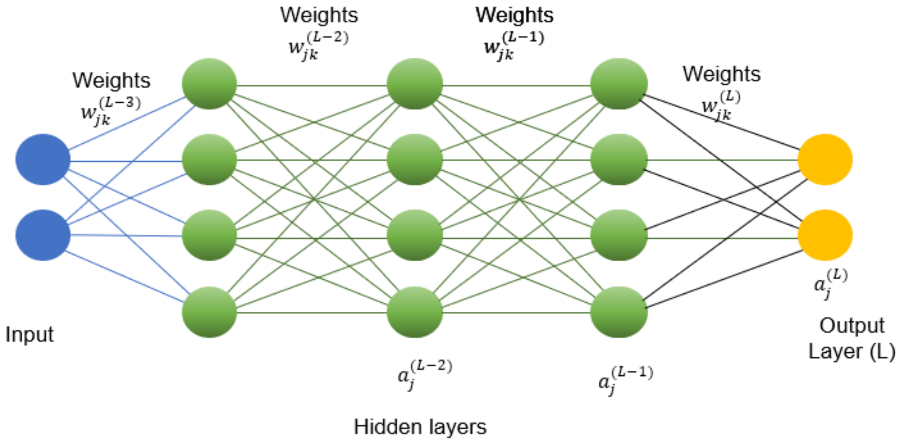

Figure 1: Basic neural network

$$\frac{\partial C_0}{\partial a_j^{(L)}} = \text{The derivative of the loss function} \tag{2}$$

$$\frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} = g^{(L)'}(z_j^{(L)}) \text{ Derivative of the activation function} \tag{3}$$

$$\frac{\partial z_j^{(L)}}{\partial w_{jk}^{(L)}} = \frac{\partial}{\partial w_k^{(L)}} \sum_i w_{ji}^{(L)} a_i^{(L-1)} = a_j^{(L-1)} \tag{4}$$

1.1.2 The derivative w.r.t. a weight between layer (L-2) and (L-1).

The equation becomes similar as the one above

$$\frac{\partial C_0}{\partial w_{21}^{(L-1)}} = \frac{\partial C_0}{\partial a_2^{(L-1)}} \frac{\partial a_2^{(L-1)}}{\partial z_2^{(L-1)}} \frac{\partial z_2^{(L-1)}}{\partial w_{21}^{(L-1)}} \tag{5}$$

The final two derivatives can be found in the same manner as in equation 1. Now computing the term that is different.

$$\frac{\partial C_0}{\partial a_2^{(L-1)}} = \sum_j \frac{\partial C_0}{\partial a_j^{(L)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial z_j^{(L)}}{\partial a_2^{(L-1)}} \tag{6}$$

Here j runs over all the nodes in the final (output) layer, and we have already computed the first two factors in equation 1, the final factor is

$$\frac{\partial z_j^{(L)}}{\partial a_2^{(L-1)}} = \frac{\partial}{\partial a_2^{(L-1)}} \sum_i w_{ji}^{(L)} a_i^{(L)} = w_{j2}^{(L)} \tag{7}$$

1.1.3 The derivative w.r.t. a weight between layer (L-3) and (L-2).

We can continue the process in a similar manner for the next set of weights.

$$\frac{\partial C_0}{\partial w_{21}^{(L-2)}} = \frac{\partial C_0}{\partial a_2^{(L-2)}} \frac{\partial a_2^{(L-2)}}{\partial z_2^{(L-2)}} \frac{\partial z_2^{(L-2)}}{\partial w_{21}^{(L-2)}} \tag{8}$$

Again we see that almost everything is similar to the sections above. The only "tricky" part is in computing the first factor. Here we must apply the chain rule for partial derivatives.

$$\frac{\partial C_0}{\partial a_2^{(L-2)}} = \sum_k \frac{\partial C_0}{\partial a_k^{(L-1)}} \frac{\partial a_k^{(L-1)}}{\partial z_k^{(L-1)}} \frac{\partial z_k^{(L-1)}}{\partial a_2^{(L-2)}} \tag{9}$$

Where we see again most of the terms have already been computed in previous steps or are easy to evaluate, and the sum is over all nodes in layer $(L-1)$.

1.1.4 The derivative w.r.t. a weight between layer (L-m-1) and (L-m).

Finally we can also do this on the weights between any two layers. Note the this is different when we are at the final layer.

$$\frac{\partial C_0}{\partial w_{jk}^{(L-m)}} = \frac{\partial C_0}{\partial a_j^{(L-m)}} \frac{\partial a_j^{(L-m)}}{\partial z_j^{(L-m)}} \frac{\partial z_j^{(L-m)}}{\partial w_{jk}^{(L-m)}} \tag{10}$$

$$\frac{\partial C_0}{\partial a_j^{(L-m)}} = \sum_k \frac{\partial C_0}{\partial a_k^{(L-m+1)}} \frac{\partial a_k^{(L-m+1)}}{\partial z_k^{(L-m+1)}} \frac{\partial z_k^{(L-m+1)}}{\partial a_j^{(L-m)}} \tag{11}$$

References

[Dee] DeepLizard. *Deep learning video series*. URL: https://deeplizard.com/learn/video/Zr5viAZGndE (visited on 02/19/2020).

[LBH15] Lecun, Y., Bengio, Y., and Hinton, G. “Deep learning”. In: *Nature* vol. 521, no. 7553 (2015), p. 436.