Backpropagation. A Peek into the Mathematics of Optimization

365 DataScience

1 Motivation

In order to get a truly deep understanding of deep neural networks, one must look at the mathematics of it. As backpropagation is at the core of the optimization process, we wanted to introduce you to it. This is definitely not a necessary part of the course, as in TensorFlow, sk-learn, or any other machine learning package (as opposed to simply NumPy), will have backpropagation methods incorporated.

2 The specific net and notation we will examine

Here's our simple network:

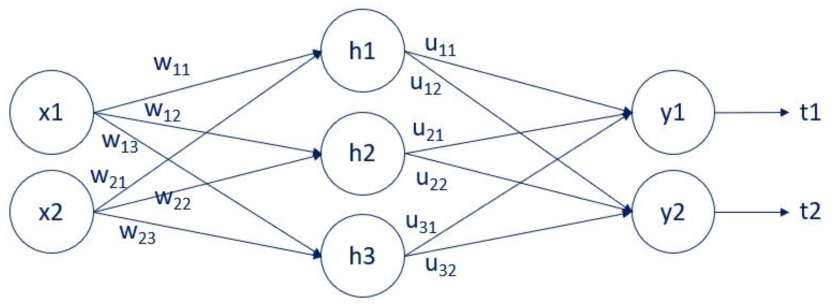

Figure 1: Backpropagation

We have two inputs: x_1 and x_2. There is a single hidden layer with 3 units (nodes): h_1, h_2, and h_3. Finally, there are two outputs: y_1 and y_2. The arrows that connect them are the weights. There are two weights matrices: $\mathbf{w}$, and $\mathbf{u}$. The $\mathbf{w}$ weights connect the input layer and the hidden layer. The $\mathbf{u}$ weights connect the hidden layer and the output layer. We have employed the letters $\mathbf{w}$, and $\mathbf{u}$, so it is easier to follow the computation to follow.
You can also see that we compare the outputs y_1 and y_2 with the targets t_1 and t_2.

There is one last letter we need to introduce before we can get to the computations. Let a be the linear combination prior to activation. Thus, we have: $\mathbf{a}^{(1)} = \mathbf{x}\mathbf{w} + \mathbf{b}^{(1)}$ and $\mathbf{a}^{(2)} = \mathbf{h}\mathbf{u} + \mathbf{b}^{(2)}$.
Since we cannot exhaust all activation functions and all loss functions, we will focus on two of the most common. A **sigmoid** activation and an **L2-norm loss**.
With this new information and the new notation, the output y is equal to the activated linear combination. Therefore, for the output layer, we have $\mathbf{y} = \sigma(\mathbf{a}^{(2)})$, while for the hidden layer: $\mathbf{h} = \sigma(\mathbf{a}^{(1)})$.
We will examine backpropagation for the output layer and the hidden layer separately, as the methodologies differ.

3 Useful formulas

I would like to remind you that:

$$\text{L2-norm loss: } L = \frac{1}{2}\sum_i (y_i - t_i)^2$$

The sigmoid function is:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

and its derivative is:

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

4 Backpropagation for the output layer

In order to obtain the update rule:

$$\mathbf{u} \leftarrow \mathbf{u} - \eta \nabla_{\mathbf{u}} L(\mathbf{u})$$

we must calculate

$$\nabla_{\mathbf{u}} L(\mathbf{u})$$

Let's take a single weight u_{ij}. The partial derivative of the loss w.r.t. u_{ij} equals:

$$\frac{\partial L}{\partial u_{ij}} = \frac{\partial L}{\partial y_j} \frac{\partial y_j}{\partial a_j^{(2)}} \frac{\partial a_j^{(2)}}{\partial u_{ij}}$$

where i corresponds to the previous layer (input layer for this transformation) and j corresponds to the next layer (output layer of the transformation). The partial derivatives were computed simply following the chain rule.

$$\frac{\partial L}{\partial y_j} = (y_j - t_j)$$

following the L2-norm loss derivative.

$$\frac{\partial y_j}{\partial a_j^{(2)}} = \sigma(a_j^{(2)})(1-\sigma(a_j^{(2)})) = y_j(1-y_j)$$

following the sigmoid derivative.
Finally, the third partial derivative is simply the derivative of $\mathbf{a}^{(2)} = \mathbf{hu} + \mathbf{b}^{(2)}$. So,

$$\frac{\partial a_j^{(2)}}{\partial u_{ij}} = h_i$$

Replacing the partial derivatives in the expression above, we get:

$$\frac{\partial L}{\partial u_{ij}} = \frac{\partial L}{\partial y_j}\frac{\partial y_j}{\partial a_j^{(2)}}\frac{\partial a_j^{(2)}}{\partial u_{ij}} = (y_j - t_j)y_j(1-y_j)h_i = \delta_j h_i$$

Therefore, the update rule for a single weight for the output layer is given by:

$$u_{ij} \leftarrow u_{ij} - \eta\delta_j h_i$$

5 Backpropagation of a hidden layer

Similarly to the backpropagation of the output layer, the update rule for a single weight, w_{ij} would depend on:

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial h_j}\frac{\partial h_j}{\partial a_j^{(1)}}\frac{\partial a_j^{(1)}}{\partial w_{ij}}$$

following the chain rule.
Taking advantage of the results we have so far for transformation using the sigmoid activation and the linear model, we get:

$$\frac{\partial h_j}{\partial a_j^{(1)}} = \sigma(a_j^{(1)})(1-\sigma(a_j^{(1)})) = h_j(1-h_j)$$

and

$$\frac{\partial a_j^{(1)}}{\partial w_{ij}} = x_i$$

The actual problem for backpropagation comes from the term $\frac{\partial L}{\partial h_j}$. That's due to the fact that there is no "hidden" target. You can follow the solution for weight w_{11} below. It is advisable to also check Figure 1, while going through the computations.

$$
\begin{aligned}
\frac{\partial L}{\partial h_1} &= \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial a_1^{(2)}}\frac{\partial a_1^{(2)}}{\partial h_1} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial a_2^{(2)}}\frac{\partial a_2^{(2)}}{\partial h_1} = \\
&= (y_1 - t_1)y_1(1-y_1)u_{11} + (y_2 - t_2)y_2(1-y_2)u_{12}
\end{aligned}
$$

From here, we can calculate $\frac{\partial L}{\partial w_{11}}$, which was what we wanted. The final expression is:

$$\frac{\partial L}{\partial w_{11}} = \left[(y_1 - t_1)y_1(1-y_1)u_{11} + (y_2 - t_2)y_2(1-y_2)u_{12}\right] h_1(1-h_1)x_1$$

The generalized form of this equation is:

$$\frac{\partial L}{\partial w_{ij}} = \sum_k (y_k - t_k)y_k(1-y_k)u_{jk}h_j(1-h_j)x_i$$

6 Backpropagation generalization

Using the results for backpropagation for the output layer and the hidden layer, we can put them together in one formula, summarizing backpropagation, in the presence of L2-norm loss and sigmoid activations.

$$\frac{\partial L}{\partial w_{ij}} = \delta_j x_i$$

where for a hidden layer

$$\delta_j = \sum_k \delta_k w_{jk} y_j(1-y_j)x_i$$

Kudos to those of you who got to the end.

Thanks for reading.