A

Backpropagation with Binary Cross-Entropy

Let's consider a simple binary classification task. It is common to use a network with a single logistic output with the binary cross-entropy loss function and for the sake of simplicity, let's assume that there is only one hidden layer.

$$BCE = -\sum_{i=1}^{nout} \left(y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right) \log\left(1 - \hat{y}_i\right)\right)$$

Where y is the ground truth and $\hat{y}$ is the output of the network. After having the loss function, let's continue with the forward pass.

$$a_k = h_{k-1} w_k + b_k$$

$$h_k = f(a_k)$$

Where, w_k is the weight, b_k is the bias term, h_k is the output of the layer (which means that $h_0 = X$ and $h_2 = \hat{y}$) and f is the non linear function. Please note that for last layer logistic function is used whereas for hidden layer reLU is used as non linear functions.

We can compute the derivative of the weights by using the chain rule.

$$\frac{\partial BCE}{\partial w_2} = \frac{\partial BCE}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_2} \frac{\partial a_2}{\partial w_2}$$

A. Backpropagation with Binary Cross-Entropy

Computing each factor in the term, we have:

$$\begin{aligned}\frac{\partial BCE}{\partial \hat{y}} &= \frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}} \\ &= \frac{\hat{y}-y}{\hat{y}\,(1-\hat{y})} \\ \frac{\partial \hat{y}}{\partial a_2} &= \hat{y}\,(1-\hat{y}) \\ \frac{\partial a_2}{\partial w_2} &= h_1^T\end{aligned}$$

Which gives us:

$$\frac{\partial BCE}{\partial w_2} = h_1^T\,(\hat{y}-y)$$

Derivative of the w_1 concerning loss function can be calculated as the following:

$$\frac{\partial BCE}{\partial w_1} = \frac{\partial BCE}{\partial h_1}\frac{\partial h_1}{\partial a_1}\frac{\partial a_1}{\partial w_1}$$

Compute each factor in the term again, we have:

$$\begin{aligned}\frac{\partial BCE}{\partial h_1} &= \frac{\partial BCE}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial a_2}\frac{\partial a_2}{\partial h_1} \\ &= (\hat{y}-y)\,w_2^T \\ \frac{\partial h_1}{\partial a_1} &= f'(a_1) \\ \frac{\partial a_1}{\partial h_1} &= X^T\end{aligned}$$

Which gives us:

$$\frac{\partial BCE}{\partial w_1} = (X)^T\,(\hat{y}-y)\left(w_2^T\right) \odot f'(a_1)$$

Where $\odot$ is element-wise multiplication, similarly, bias terms can be calculated by following:

$$\begin{aligned}\frac{\partial BCE}{\partial b_2} &= \frac{\partial BCE}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial a_2}\frac{\partial a_2}{\partial b_2} \\ &= (\hat{y}-y) \\ \frac{\partial BCE}{\partial b_1} &= \frac{\partial BCE}{\partial h_1}\frac{\partial h_1}{\partial a_1}\frac{\partial a_1}{\partial b_1} \\ &= (\hat{y}-y)\left(w_2^T\right) \odot f'(a_1)\end{aligned}$$

After having all these results, we can update the parameters (weights and biases) using gradient descent and its variants.