NN Gradients

Archim Jhunjhunwala

March 27, 2024

1 Goal

The goal is to efficiently compute the gradient of the cost function - C with respect to the weights and biases of a neural networks - $W^l_{i,j}$ denoting the weight from node j in layer $l-1$ to node i in layer l. This piece concerns only feed-forward neural networks. Each layer has an activation function $f_l(x)$, as well as the activated a^l_i and unactivated u^l_i values associated with each node i.

2 Single Weights / Biases

Ignoring matrices, let's find $\frac{\partial C}{\partial b^l_i}$, and $\frac{\partial C}{\partial W^l_{i,j}}$ for some layer l in our network of L layers.

2.1 Bias Gradient

$\frac{\partial C}{\partial b^l_i} = \frac{\partial C}{\partial u^l_i}\frac{\partial u^l_i}{\partial b^l_i} = \frac{\partial C}{\partial a^l_i}\frac{\partial a^l_i}{\partial u^l_i}\frac{\partial u^l_i}{\partial b^l_i}$ by the chain rule
$\frac{\partial a^l_i}{\partial u^l_i} = \frac{\partial}{\partial u^l_i} f(u^l_i) = f'(u^l_i)$

$\frac{\partial u_i^l}{\partial b_i^l} = \frac{\partial}{\partial b_i^l}(b_i^l + ...) = 1$
$\frac{\partial C}{\partial b_i^l} = \frac{\partial C}{\partial a_i^l} f'(u_i^l)$

2.2 Weight Gradient

$\frac{\partial C}{\partial W_{i,j}^l} = \frac{\partial C}{\partial u_i^l} \frac{\partial u_i^l}{\partial W_{i,j}^l} = \frac{\partial C}{\partial a_i^l} \frac{\partial a_i^l}{\partial u_i^l} \frac{\partial u_i^l}{\partial W_{i,j}^l}$ by the chain rule
$\frac{\partial u_i^l}{\partial W_{i,j}^l} = \frac{\partial}{\partial W_{i,j}^l}(W_{i,j}^l a_{l-1,j} + ...) = a_j^{l-1}$
$\frac{\partial C}{\partial W_{i,j}^l} = \frac{\partial C}{\partial a_i^l} f'(u_i^l) a_j^{l-1}$

2.3 Backpropagation

We must find 2 more values before we are done: $\frac{\partial C}{\partial a_i^L}$ (this value depends on the specific cost function used, and I will ignore it here). $\frac{\partial C}{\partial a_i^{l-1}}$ as an expression of $\frac{\partial C}{\partial a_i^l}$. In this way, we can propagate backwards through the network calculating all the necessary gradients.
$\frac{\partial C}{\partial a_i^{l-1}} = \sum_{j=1}^n \frac{\partial C}{\partial u_j^l} \frac{\partial u_j^l}{\partial a_i^{l-1}} = \sum_{j=1}^n \frac{\partial C}{\partial a_j^l} \frac{\partial a_j^l}{\partial u_j^l} \frac{\partial u_j^l}{\partial a_i^{l-1}}$ by the chain rule (n is the number of nodes in layer l)
$\frac{\partial u_j^l}{\partial a_i^{l-1}} = \frac{\partial}{\partial a_i^{l-1}}(W_{j,i}^l a_i^{l-1} + ...) = W_{j,i}^l$
$\frac{\partial C}{\partial a_i^{l-1}} = \sum_{j=1}^n \frac{\partial C}{\partial a_j^l} f'(u_j^l) W_{j,i}^l$
Observe that the bias gradient is present in both the weight gradient, and the derivative cost with respect to the previous layers nodes.

3 Results / Matrix Notation

$\frac{\partial C}{\partial b^l} = \begin{bmatrix} \frac{\partial C}{\partial a_1^l} f'(u_1^l) \\ ... \\ \frac{\partial C}{\partial a_n^l} f'(u_n^l) \end{bmatrix}$

$\frac{\partial C}{\partial W_{i,j}^l} = \frac{\partial C}{\partial b^l} (a^{l-1})^T$

$\frac{\partial C}{\partial a_i^{l-1}} = \frac{\partial C}{\partial b^l}^T W^l$ (this will be a row vector)