1 5. Neural Networks - Principles

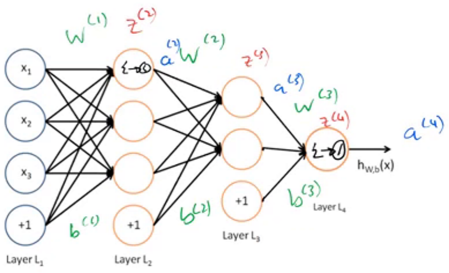

Figure 1: 4-layer NN representation

1.1 Backprop

1.1.1 Defining the layers and their error contributions

Consider a NN for regression with sigmoid activation functions

$$a^{(l)} = f(z^{(l)}) \tag{1}$$
$$z^{(l)} = W^{(l-1)} \cdot a^{(l-1)} + b^{(l-1)} \tag{2}$$

The cost function is:

$$J(\theta) = \frac{1}{2}(y - h(x))^2 \tag{3}$$
$$= \frac{1}{2}(y - a^{(nl)})^2 \tag{4}$$

where nl is the output layer. To calculate the weight updates of each layer we need to find out how much error each node in those layers contributes. For the output layer we can denote it's error fraction as δ^{nl} and derive it:

$$\delta^{(nl)} = \frac{\partial J(\theta)}{\partial a^{(nl)}} \odot \frac{\partial a^{(nl)}}{\partial z^{(nl)}} \tag{5}$$

For a hidden layer l, we find the error contribution term (gradient) by taking the upstream layer gradient $(l+1)$ and multiply it by the local gradient. In this way the error portions propagate back through the graph

$$\delta^{(l)} = (\delta^{(nl)} \cdot \frac{\partial z^{(l+1)}}{\partial a^{(l)}}) \odot \frac{\partial a^{(l)}}{\partial z^{(l)}} \tag{6}$$

$$\delta^{(l)} = (\delta^{(nl)} \cdot (W^{(l)})^T) \odot \frac{\partial a^{(l)}}{\partial z^{(l)}} \tag{7}$$

Remembering that the gradient with respect to a variable should have the same shape as the variable (as each element of that variable is quantifying how much error is being contributed)

1.1.2 Algorithm (Batch GD)

1. Compute a forward pass of the network with the input data, storing the intermediate node values $a^{(l)}$.
2. Calculate the error term at the output $\delta^{(nl)}$
3. For each layer l calculate its error term $\delta^{(l)}$
4. Updating the weights:
   (a) Find the partials: $\Delta W^l = \delta^{(l+1)} \cdot \frac{\partial z^{(l+1)}}{\partial w^{(l)}}$
   (b) Update weights: $W^{(l)} = W^{(l)} - \eta(\frac{1}{N})\Delta W^l$
5. Updating the biases:
   (a) Find the partials: $\Delta b^l = \delta^{(l+1)} \cdot 1$
   (b) Update bias: $b^{(l)} = b^{(l)} - \eta(\frac{1}{N})\Delta b^l$

This can be converted to stochastic GD by adjusting the $\frac{1}{N}$ term in the updates