Note on Backpropagation

John Hull

Backpropagation is a way of using the chain rule to calculate derivatives of the mean squared error (or other objective function) with respect to the parameter values. For convenience we assume a single target. The mean squared error is given by:

$$E = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

where there are n observations, $\hat{y}_i$ is the value of the target for the ith observation, and y_i is the estimate of the target's value given by the neural network. If θ is the value of a parameter

$$\frac{\partial E}{\partial \theta} = -\frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\frac{\partial y_i}{\partial \theta}$$

We can therefore consider each observation separately, calculating $\partial y_i/\partial\theta$, and at the end use this equation to get the partial derivative we are interested in.

We start with the values of θ used to calculate the target y_i at the end of the network and work back through the network considering other θ values. As in Chapter 6, we define L as the number of layers and K as the number of neurons per layer. The value at the kth neuron for layer l will be denoted by V_{lk}($1 \le k \le K$; $1 \le l \le L$).

First we note that if θ is a parameter relating the output to the final layer, $\partial y_i/\partial\theta$ can be calculated without difficulty. If θ is a parameter relating the value at neuron k of the final layer to a neuron in the penultimate layer, we have from the chain rule

$$\frac{\partial y_i}{\partial \theta} = \frac{\partial y_i}{\partial V_{Lk}} \frac{\partial V_{Lk}}{\partial \theta}$$

Both $\partial y_i / \partial V_{Lk}$ and $\partial V_{Lk} / \partial \theta$ can be calculated without difficulty.

Now let us consider the situation where the parameter θ relates the value at neuron k of layer l to a neuron in layer l−1 ($l < L$). Then

$$\frac{\partial y_i}{\partial \theta} = \frac{\partial y_i}{\partial V_{lk}} \frac{\partial V_{lk}}{\partial \theta}$$

The partial derivative $\partial V_{lk} / \partial \theta$ can be calculated without difficulty. We have to do a little more work to calculate $\partial y_i / \partial V_{lk}$. An application of the chain rule gives

$$\frac{\partial y_i}{\partial V_{lk}} = \sum_{k^*=1}^{K} \frac{\partial y_i}{\partial V_{l+1,k^*}} \frac{\partial V_{l+1,k^*}}{\partial V_{lk}}$$

The partial derivative, $\partial V_{l+1,k^*} / \partial V_{lk}$, can be calculated without difficulty for all k and k^*. Because calculations start at the end of the network and work back, we have already calculated the values of $\partial y_i / \partial V_{l+1,k^*}$ for all k^* by the time that we consider a θ that relates layer l−1 to layer l.

Taken together, the equations we have presented provide a fast way to calculate all the partial derivatives necessary for the gradient descent algorithm.