Back propagation algorithm:

We have a neural network consisting of L layers, each labeled by l for $l = 1, 2, \ldots, L$.

The layer l consists of N_l nodes, each with N_{l-1} weights.

We label the weights by w_{ij}^l where the subscript i indicates what neuron in layer l it belongs to, while the subscript j indicates that it is multiplied by the input received from neuron j in layer $l-1$.

We label the biases by b_i^l where the subscript i again indicates what neuron in layer l it belongs to.

We label the activations a_i^l and the activities $z_i^l = \sum_j w_{ij}^l a_j^{l-1} + b_i^l$.

Finally we label the desired outputs $\mathbf{y}_k$ for $k = 1, \ldots, K$ where K is the amount of training data.

The cost function is C which is a direct function of the activities a_i^L of the last layer.

The output function is f and the activity function is σ.

We need to find the weights w_{ij}^l and the biases b_i^l such that the cost function is minimized, i.e we want

$$\frac{\partial C}{\partial w_{ij}^l} = \frac{\partial C}{\partial b_i} = 0$$

for all weights and biases.

The cost function can be written as a sum over the data set

$$C = \sum_{k=1}^{K} C_k$$

where a possible choice of a cost function is one where each term C_k are on the form

$$C_k = \frac{1}{2}\left(\widetilde{\mathbf{y}}_k - \mathbf{y}_k\right)^2$$

We start by considering only a single term C_k and pretend that there is only a single desired output $\mathbf{y}$ and a single prediction $\widetilde{\mathbf{y}}$.

$$C_k = \frac{1}{2}\left(\widetilde{\mathbf{y}} - \mathbf{y}\right)^2 = \frac{1}{2}\sum_i \left(a_i^L - y_i\right)^2$$

We will use this to derive an algorithm which can be extended by simply looping over the data set $\{\mathbf{y}_k\}$. Note that this is just a possible choice of the cost function. In the following derivation the cost function is completely general.

Since C_k is a direct function of the outputs a_i^L we start by finding the derivatives with respect to the weights w_{ij}^L of the output layer.

$$\frac{\partial C_k}{\partial w_{ij}^L} = \frac{\partial C_k}{\partial a_i^L}\frac{\partial a_i^L}{\partial z_i^L}\frac{\partial z_i^L}{\partial w_{ij}^L}$$

Here $\partial a_i^L \,/\, \partial z_i^L = f'\left(z_i^L\right)$ and $\partial z_i^L \,/\, \partial w_{ij}^L = a_i^{L-1}$. Thus

$$\frac{\partial C_k}{\partial w_{ij}^L} = \frac{\partial C_k}{\partial a_i^L} f'\left(z_i^L\right) a_i^{L-1}$$

The first couple of factors will appear a lot, so we define

$$\delta_i^L \equiv \frac{\partial C_k}{\partial a_i^L} f'\left(z_i^L\right)$$

to write

$$\frac{\partial C_k}{\partial w_{ij}^L} = \delta_i^L a_i^{L-1}$$

Now we find the derivatives of C_k with respect to the biases b_i^L

$$\frac{\partial C_k}{\partial b_i^L} = \frac{\partial C_k}{\partial a_i^L}\frac{\partial a_i^L}{\partial z_i^L}\frac{\partial z_i^L}{\partial b_i^L}$$

Here $\partial a_i^L \,/\, \partial z_i^L = f'\left(z_i^L\right)$ and $\partial z_i^L \,/\, \partial b_i^L = 1$, so

$$\frac{\partial C_k}{\partial b_i^L} = \frac{\partial C_k}{\partial a_i^L} f'\left(z_i^L\right) = \delta_i^L$$

Now even though C_k is a direct function of the activations a_i^L of the last layer, we could just as well (if we wanted to) write C_k as a direct function of the activities z_i^L of the last layer as well. The derivatives of C_k with respect to the activities of the last layer are

$$\frac{\partial C_k}{\partial z_i^L} = \frac{\partial C_k}{\partial a_i^L}\frac{\partial a_i^L}{\partial z_i^L} = \frac{\partial C_k}{\partial a_i^L} f'\left(z_i^L\right) = \delta_i^L$$

We can use these to find an expression for the total derivative of C_k, which will help us to find the derivatives of C_k with respect to the weights and biases of the next-to-last layer.

$$dC_k = \sum_i \frac{\partial C_k}{\partial z_i^L} dz_i^L = \sum_i \delta_i^L dz_i^L$$

Now, since

$$\frac{\partial z_i^L}{\partial w_{jn}^{L-1}} = \frac{\partial z_i^L}{\partial a_j^{L-1}}\frac{\partial a_j^{L-1}}{\partial z_j^{L-1}}\frac{\partial z_j^{L-1}}{\partial w_{jn}^{L-1}} = w_{ij}^L \sigma'\left(z_j^{L-1}\right) a_n^{L-2}$$

and

$$\frac{\partial z_i^L}{\partial b_j^{L-1}} = \frac{\partial z_i^L}{\partial a_j^{L-1}}\frac{\partial a_j^{L-1}}{\partial z_j^{L-1}}\frac{\partial z_j^{L-1}}{\partial b_j^{L-1}} = w_{ij}^L \sigma'\left(z_j^{L-1}\right)$$

the derivatives of C_k with respect to the weights and biases of the next-to-last layer are

$$\frac{\partial C_k}{\partial w_{jn}^{L-1}} = \sum_i \delta_i^L \frac{\partial z_i^L}{\partial w_{jn}^{L-1}} = \sum_i \delta_i^L w_{ij}^L \sigma'\left(z_j^{L-1}\right) a_n^{L-2}$$

and

$$\frac{\partial C_k}{\partial b_j^{L-1}} = \sum_i \delta_i^L \frac{\partial z_i^L}{\partial b_j^{L-1}} = \sum_i \delta_i^L w_{ij}^L \sigma'\left(z_j^{L-1}\right)$$

Defining

$$\delta_j^{L-1} \equiv \sum_i \delta_i^L w_{ij}^L \sigma'\left(z_j^{L-1}\right)$$

we can write

$$\frac{\partial C_k}{\partial w_{jn}^{L-1}} = \delta_j^{L-1} a_n^{L-2}$$

and

$$\frac{\partial C_k}{\partial b_j^{L-1}} = \delta_j^{L-1}$$

So the derivatives of the cost function with respect to the weights and biases of the next-to-last layer can be written on the same form as the derivatives with respect to the weights and biases of the outer layer. In fact, if we write the total differential of C_k in terms of derivatives of C_k with respect to the activities z_i^{L-1} of the next-to-last layer, we can find the derivatives of C_k with respect to the weights w_{jn}^{L-2} and the biases b_j^{L-2}, and they are on the same form as well. Repeating this process, with a general δ_j^{l-1} given by

$$\delta_j^{l-1} \equiv \sum_i \delta_i^l w_{ij}^l \sigma'\left(z_j^{l-1}\right)$$

we can find the derivatives of C_k with respect to *all* the weights and biases in the neural network. Therefore, we can use the following algorithm:

$$\delta_i^L = \frac{\partial C_k}{\partial a_i^L} f'\left(z_i^L\right)$$

for $l = L,\ L-1,\ \ldots,\ 2$:

$$\frac{\partial C_k}{\partial w_{ij}^L} = \delta_i^L a_i^{L-1}$$

$$\frac{\partial C_k}{\partial b_i^l} = \delta_i^l$$

$$\delta_j^{l-1} \equiv \sum_i \delta_i^l w_{ij}^l \sigma'\left(z_j^{l-1}\right)$$