Back Propagation

w^l_{ji} Weight from the i neuron in the l-1 layer to the j neuron in the l layer.
b^l_j Bias from the j neuron in the l layer.

a^l_j Activation from the j neuron in the l layer.
$a^l_j = \sum_i w^l_{ji} a^{l-1}_i + b^l_j = \sigma(z^l_j)$
$a^l = w^l a^{l-1} + b^l = \sigma(z^l)$

Cost function : $C = \frac{1}{2n} \sum_i (target_i - a^L_i)^2$
Cost function derivative : $\frac{\partial C}{\partial a^L_i} = (a^L_i - target_i)$

$\frac{\partial C}{\partial w}$ and $\frac{\partial C}{\partial b}$ represent how quickly Cost function changes with respect to weight w and bias b. Back Propagation is to relate δ^l_i to $\frac{\partial C}{\partial w}$ and $\frac{\partial C}{\partial b}$.

$delta^l_i$ Error in the i neuron in the l layer.

BP1 : $\delta^L_i = \frac{\partial C}{\partial a^L_i} \sigma'(z^L_i) = (a^L_i - target_i)\sigma'(z^L_i)$
Matrix : $\delta^L = \nabla_a C \odot \sigma'(z^L) = (a^L - target) \odot \sigma'(z^L)$

BP2 : $\delta^l_i = \frac{\partial C}{\partial z^l_i} = \sum_j \frac{\partial C}{\partial z^{l+1}_j} \frac{\partial z^{l+1}_j}{\partial z^l_i} = \sum_j \delta^{l+1}_j \frac{\partial z^{l+1}_j}{\partial z^l_i} = \sum_j \delta^{l+1}_j w^{l+1}_{ji} \sigma'(z^l_i)$
$(\frac{\partial z^{l+1}_j}{\partial z^l_i} = w^{l+1}_{ji} \sigma'(z^l_j))$

BP1 compute the error in L layer, then apply BP2 to compute the layers back through.

BP3 : $\frac{\partial C}{\partial b^l_i} = \delta^l_i$
Matrix : $\frac{\partial C}{\partial b} = \delta$

BP4 : $\frac{\partial C}{\partial w^l_{ji}} = a^{l-1}_i \delta^l_j$
Matrix : $\frac{\partial C}{\partial w} = a_{in} \delta_{out}$
a_{in} is the activation of the neuron input to the weight, and δ_{out} is the error of the neuron output from the weight.

update weights according to $w^l = w^l - \frac{\partial C}{\partial w{i^l_j}}$.
update biases according to $b^l = b^l - \frac{\partial C}{\partial b^l_i}$.