The Generalised Delta Rule and Gradient Descent

The change to the weight on a connection from a unit i to a unit j when a pattern p is presented is given by the delta rule -

$$\Delta^p \mathbf{w}_{ji} = \eta \delta_j^p \mathbf{o}_i^p$$

where,

p	is the input pattern
$\Delta^p w_{ji}$	is the change to the weight on the connection from unit i to unit j
η	is the learning rate
δ_j^p	is the error at unit j
o_i^p	is the activation of unit i

For gradient descent we need –

$$\Delta^p \mathbf{w}_{ji} \propto -\frac{\partial \mathbf{E}^p}{\partial \mathbf{w}_{ji}}$$

where E^p is the error on presentation of pattern p

From the chain rule we have –

$$\frac{\partial \mathbf{E}^p}{\partial \mathbf{w}_{ji}} = \frac{\partial \mathbf{E}^p}{\partial \mathbf{net}_j^p} \times \frac{\partial \mathbf{net}_j^p}{\partial \mathbf{w}_{ji}}$$

where net_j^p is the total input to node j

Now,

$$\mathbf{net}_j^p = \sum_i \mathbf{w}_{ji} \mathbf{o}_i^p$$

where o_i is the output of unit i

$$\therefore \frac{\partial \mathbf{net}_j^p}{\partial \mathbf{w}_{ji}} = \frac{\partial \left(\sum_k \mathbf{w}_{jk} \mathbf{o}_k^p \right)}{\partial \mathbf{w}_{ji}} = \mathbf{o}_i^p$$

So setting

$$\delta_j^p = -\frac{\partial \mathbf{E}^p}{\partial \mathbf{net}_j^p}$$

will yield ...

$$\Delta^p \mathbf{w}_{ji} = \eta \delta_j^p \mathbf{o}_i^p$$

****

Now, applying the chain rule to δ_j^p, we have –

$$-\frac{\partial \mathbf{E}^{\mathbf{p}}}{\partial \mathbf{net}_{\mathbf{j}}^{\mathbf{p}}} = -\frac{\partial \mathbf{E}^{\mathbf{p}}}{\partial \mathbf{o}_{\mathbf{j}}^{\mathbf{p}}} \times \frac{\partial \mathbf{o}_{\mathbf{j}}^{\mathbf{p}}}{\partial \mathbf{net}_{\mathbf{j}}^{\mathbf{p}}}$$

and,

$$\mathbf{o}_{\mathbf{j}}^{\mathbf{p}} = \mathbf{f}_{\mathbf{j}}\left(\mathbf{net}_{\mathbf{j}}^{\mathbf{p}}\right)$$

where f_j is the activation function for node j

$$\therefore \frac{\partial \mathbf{o}_{\mathbf{j}}^{\mathbf{p}}}{\partial \mathbf{net}_{\mathbf{j}}^{\mathbf{p}}} = \mathbf{f}_{\mathbf{j}}'\left(\mathbf{net}_{\mathbf{j}}^{\mathbf{p}}\right)$$

So,

$$\delta_{\mathbf{j}}^{\mathbf{p}} = -\frac{\partial \mathbf{E}^{\mathbf{p}}}{\partial \mathbf{o}_{\mathbf{j}}^{\mathbf{p}}} \times \mathbf{f}_{\mathbf{j}}'\left(\mathbf{net}_{\mathbf{j}}^{\mathbf{p}}\right)$$

****

The remaining partial differential is readily determined for an output node, j, since we can measure the error at such a node –

$$\frac{\partial E^p}{\partial o_j^p} = -\left(T_j^p - o_j^p\right)$$

where T_j^p is the desired output for node j when pattern p is presented.

Note that this also ensures that a weight change of zero is produced when the error is zero and that the sign of the weight change is reversed when the error is negative.

Therefore, for an output node,

$$\delta_j^p = \left(T_j^p - o_j^p\right) \cdot f_j'\left(net_j^p\right)$$

****

However, for a hidden node, j, the partial differential is not so easily determined. We need to apply the chain rule once again so that we can obtain δ_j^p in terms of the δ_k^p of the nodes, k, in the next layer up -

$$\frac{\partial \mathbf{E^p}}{\partial \mathbf{o_j^p}} = \sum_{\mathbf{k}} \frac{\partial \mathbf{E^p}}{\partial \mathbf{net_k^p}} \times \frac{\partial \mathbf{net_k^p}}{\partial \mathbf{o_j^p}}$$

$$= \sum_{\mathbf{k}} \left(\frac{\partial \mathbf{E^p}}{\partial \mathbf{net_k^p}} \times \frac{\partial \left(\sum_{\mathbf{i}} \mathbf{w_{ki} o_i^p} \right)}{\partial \mathbf{o_j^p}} \right)$$

$$= \sum_{\mathbf{k}} \frac{\partial \mathbf{E^p}}{\partial \mathbf{net_k^p}} \times \mathbf{w_{kj}}$$

$$= -\sum_{\mathbf{k}} \delta_{\mathbf{k}}^{\mathbf{p}} \times \mathbf{w_{kj}}$$

Therefore, for a hidden node,

$$\delta_{\mathbf{j}}^{\mathbf{p}} = \mathbf{f_j'}\left(\mathbf{net_j^p}\right) \cdot \sum_{\mathbf{k}} \delta_{\mathbf{k}}^{\mathbf{p}} \mathbf{w_{kj}}$$

****

So, to summarise, the Generalised Delta Rule is -

$$\Delta^{\mathbf{p}} \mathbf{w_{ji}} = \eta \delta_{\mathbf{j}}^{\mathbf{p}} \mathbf{o_i^p}$$

where

$$\delta_j^p = \left(T_j^p - o_j^p\right) \cdot f_j'\left(net_j^p\right)$$

for an output node

and

$$\delta_{\mathbf{j}}^{\mathbf{p}} = \mathbf{f_j'}\left(\mathbf{net_j^p}\right) \cdot \sum_{\mathbf{k}} \delta_{\mathbf{k}}^{\mathbf{p}} \mathbf{w_{kj}}$$

for a hidden node

Problems with Gradient Descent Learning

There are three main problems with gradient descent. They are the inverses of the classical hill-climbing problems –

i). **Local Minima**
Sub-optimal solutions arise when the algorithm gets stuck in a local minimum and cannot proceed to uncover the global minimum. It is fairly easy to detect when this has happened – the error remains unacceptably high but the weights are no longer being modified.
Solutions include changing the learning rate, adding a momentum term, changing the initial weights.

ii). **Plateaux**
Expanses of the error surface which are totally flat mean that the algorithm has no directional information to guide it towards the global minimum. It is fairly easy to detect when this has happened – the weights keep changing but the error remains constant.
Solutions include increasing the learning rate or changing the initial weights.

iii). **Crevasses**
The algorithm follows a path which is gradually spiralling downwards but the end-point is not the global minimum. The path is running along the bottom of a crevasse and reaching the global minimum would require climbing out of it first. This is similar to the local minimum problem except that the error keeps decreasing. This is a very difficult situation to detect.
If detected (a big "if") re-initialising the weights is the only solution.