$$E_n \triangleq \frac{1}{2}\sum_{k=1}^{K}(y_k - t_k)^2$$

$$\delta_k \triangleq \frac{\partial E_n}{\partial a_k} = \frac{1}{2}\cdot 2\,(y_k - t_k)\frac{\partial y_k}{\partial a_k}$$

output layer activation

$\frac{\partial y_k}{\partial a_k} = 1$ as $y_k = a_k$ [$\sigma(\cdot) =$ unit fn.]

$$\delta_k = y_k - t_k$$

(Else, according to the specific activation function $\sigma(\cdot)$ at the output layer)

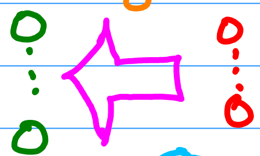
③ <u>Backpropagate these to obtain δ_j's for the hidden layer units</u>

$$\delta_j = (1 - z_j^2)\sum_{k=1}^{K} w_{kj}\,\delta_k$$

What is this, and how?

[hidden layer]

previous step (step #2) [output layer]



$$\delta_j \triangleq \frac{\partial E_n}{\partial a_j} = \sum_{\forall k} \frac{\partial E_n}{\partial a_k}\,\frac{\partial a_k}{\partial a_j}$$

$\frac{\partial E_n}{\partial a_k} = \delta_k$ [step #2]

$$a_k = \underline{w}^{(2)T}\underline{z} = \sum_{j=0}^{M} w_{kj}^{(2)} z_j = \sum_{j=0}^{M} w_{kj}^{(2)} h(a_j)$$

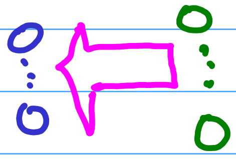
$$\Rightarrow \frac{\partial a_k}{\partial a_j} = w_{kj}^{(2)}\frac{\partial h(a_j)}{\partial a_j} = w_{kj}^{(2)} h'(a_j)$$
$$= w_{kj}^{(2)}(1 - z_j^2)$$

$$\delta_j = \sum_{\forall k} \delta_k w_{kj}^{(2)}(1 - z_j^2) = (1 - z_j^2)\sum_{\forall k} w_{kj}^{(2)}\delta_k$$

④ Use the chain rule to evaluate the gradient

$$\frac{\partial E_n}{\partial w_{ji}^{(1)}} = \delta_j x_i \,, \quad \frac{\partial E_n}{\partial w_{kj}^{(2)}} = \delta_k z_j$$

What is this, and how?



$$\bar{\nabla} E = \begin{bmatrix} \vdots \\ \partial E_n / \partial w \\ \vdots \end{bmatrix}$$

all weights (1st & 2nd layers of connections)

$$\frac{\partial E_n}{\partial w_{ji}^{(1)}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}^{(1)}} \; ; \quad a_j = \underline{w}_j^{(1)T} \underline{x} = \sum_{i=0}^{D} w_{ji}^{(1)} x_i$$

δ_j (previous step: step#3)

$$\Rightarrow \frac{\partial a_j}{\partial w_{ji}^{(1)}} = x_i$$

$$\Rightarrow \frac{\partial E_n}{\partial w_{ji}^{(1)}} = \delta_j x_i$$

$$\frac{\partial E_n}{\partial w_{kj}^{(2)}} = \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial w_{kj}} \; ; \quad a_k = \underline{w}_k^{(2)T} \underline{z} = \sum_{j=0}^{M} w_{kj}^{(2)} z_j$$

δ_k (step#2)

$$\Rightarrow \frac{\partial a_k}{\partial w_{kj}^{(2)}} = z_j$$

$$\Rightarrow \frac{\partial E_n}{\partial w_{kj}^{(2)}} = \delta_k z_j$$

Side topic: Numerical Evaluation of the gradient

Empirically, all of these alternative methods are numerically not as good as Backpropagation

$$(*) \quad \frac{\partial E_n}{\partial \omega_{ji}} = \frac{E_n(\omega_{ji}+\epsilon) - E_n(\omega_{ji})}{\epsilon} + o(\epsilon)$$

$\epsilon \rightarrow (\omega_{ji}+\epsilon) - (\omega_{ji})$

OR: symmetrical central differences

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{E_n(w_{ji}+\epsilon) - E_n(\omega_{ji}-\epsilon)}{2\epsilon} + o(\epsilon)$$

Why? The first direct formula?

$$E_n(w_{ji}+\epsilon) = E_n(\omega_{ji}) + \epsilon \frac{\partial E_n}{\partial \omega_{ji}} + \frac{1}{2}\epsilon H \epsilon + \text{higher order terms}$$

$$\Rightarrow \frac{E_n(w_{ji}+\epsilon) - E_n(w_{ji})}{\epsilon} = \frac{\partial E_n}{\partial w_{ji}} + \frac{1}{2}\epsilon H$$

Why? The symmetrical central differences formula:

$$E_n(w_{ji}+\epsilon) = E_n(\omega_{ji}) + \epsilon \frac{\partial E_n}{\partial w_{ji}} + \frac{1}{2}\epsilon H \epsilon + o(\epsilon^3)$$

$$E_n(w_{ji}-\epsilon) = E_n(w_{ji}) - \epsilon \frac{\partial E_n}{\partial w_{ji}} + \frac{1}{2}\epsilon H \epsilon - o(\epsilon^3)$$

$- \quad - \quad + \quad - \quad +$

$$E_n(\omega_{ji}+\epsilon) - E_n(w_{ji}-\epsilon) = 2\epsilon \frac{\partial E_n}{\partial \omega_{ji}} + 2\,o(\epsilon^3)$$

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{E_n(w_{ji}+\epsilon) - E_n(w_{ji}-\epsilon)}{2\epsilon} + O(\epsilon^2)$$

Physical Significance: NN-based solution vis-a-vis the linear (or restricted linear) method done before.

(*) NN: the weight parameters in the first layer are shared between the outputs
linear: each classification is performed independently.

(*) The first layer of the network can be viewed as performing a non-linear feature extraction, and sharing the features between different outputs can lead to savings on computation and also lead to improved generalisation.