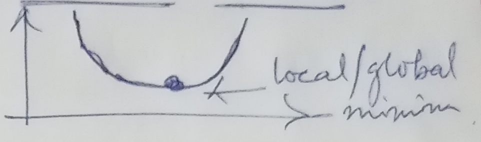
Backpropagation → justification

Cost function / Lossfunction

$$J = \frac{1}{2}\sum (T_j - O_j)^2 \Rightarrow \text{sequared error.}$$

Minimize the Cost function using Gradient Decent.

$$\boxed{\Delta W = -l \frac{\partial J}{\partial W}}$$



From ANN model we are getting →

$$Net_j = \sum x_j w_j + \theta_j$$

$$O_j = \frac{1}{1+e^{-Net_j}}$$

Hence →

$$\frac{\partial J}{\partial W} = \frac{\partial J}{\partial O_j} \cdot \frac{\partial O_j}{\partial Net_j} \cdot \frac{\partial Net_j}{\partial w_j}$$

$$J = \frac{1}{2} \cdot \sum (T_j - O_j)^2$$

$$\therefore \frac{\partial J}{\partial O_j} = \frac{1}{2} \cdot 2 \cdot (T_j - O_j)(-1) = -(T_j - O_j)$$

As we know,

$$O_j = \frac{1}{1+e^{-Net_j}}$$

Assume that

$$y = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x}.$$

$$\therefore \frac{\partial y}{\partial x} = \frac{(1+e^x) \cdot e^x - e^x \cdot e^x}{(1+e^x)^2} = \frac{e^x}{(1+e^x)^2} = \frac{e^x}{(1+e^x)}\left[1 - \frac{e^x}{1+e^x}\right]$$

$$= y(1-y)$$

Therefore,

$$\frac{\partial y}{\partial x} = y(1-y)$$

likewise,

$$\frac{\partial O_j}{\partial Net_j} = O_j(1-O_j)$$

Moreover, $$\frac{\partial Net_j}{\partial w_j} = x_j$$

Therefore,

$$\frac{\partial J}{\partial w_j} = \frac{\partial J}{\partial O_j} \cdot \frac{\partial O_j}{\partial Net_j} \cdot \frac{\partial Net_j}{\partial w_j}$$

$$= -(T_j - O_j)\, O_j(1-O_j)\, x_j$$

Now, $x_j = O_i$ that output of i^{th} Neuron.

$$\therefore \boxed{\frac{\partial J}{\partial w_j} = -(T_j - O_j)\, O_j(1-O_j)\, O_i}$$

From perceptron learning rule we are getting—

$$\Delta W = l.\underbrace{(T_j - O_j)}_{Err_j}.\, O_i = \underline{l.\, Err_j .\, O_i} \quad \text{---} (1)$$

from gradient decent we are getting →

$$\Delta W = -l.\frac{\partial J}{\partial W} = l.(T_j - O_j)\, O_j(1-O_j)\, O_i \quad \text{---} (2)$$

from (1) and (2) we are getting.

$$Err_j = (T_j - O_j)\, O_j(1-O_j)$$

or, $$\boxed{Err_j = O_j(1-O_j)(T_j - O_j)}$$