DERIVATION OF THE BACKPROPAGATION RULE

CHAIN RULE

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$$

(1) $$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$$

E_d : Error on the training example d

$$E_d(\vec{w}) = \frac{1}{2} \sum_{k \in \underbrace{outputs}_{\text{Set of output units in the network}}} (\overbrace{t_k}^{\text{target value of the unit k}} - \underbrace{O_k}_{\text{Output value of unit k.}})^2$$

ANISHA.C.D. | AP | CSE

x_{ji} → the ith input to unit j

w_{ji} → the weight associated with the ith input to unit j

net_j → $\sum_i w_{ji} x_{ji}$ (the weighted sum of inputs for unit j)

o_j → the output computed by unit j

σ → the Sigmoid function

Outputs → the set of units in the final layer of the network

Downstream (j) ⇒ the set of units whose immediate inputs the output of unit j

Derive an Expression $\frac{\partial E_d}{\partial w_{ji}}$

(w_{ji} can influence the rest of the network only through net_j)

<u>Chain Rule</u>

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}}$$

$[\because net_j = \sum_i w_{ji} x_{ji}]$

$$= \boxed{\frac{\partial E_d}{\partial net_j} x_{ji}}$$

Further derivation

Focuses on deriving an expression for $\frac{\partial E_d}{\partial net_j}$

CASE 1 : TRAINING RULE FOR OUTPUT UNIT WEIGHS.

Just as w_{ji} can Influence the rest of the network only via net_j

net_j can Influence the network only through o_j

<u>chain Rule</u>

$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\underbrace{\partial o_j}} \frac{\partial o_j}{\partial net_j} \rightarrow (1)$$

Consider the 1st term in (1)

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2$$

The derivatives $\frac{\partial}{\partial o_j} (t_k - o_k)^2$ will be zero

for all output units k except $k = j$.

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} (t_j - o_j)^2$$

$$= \frac{1}{2} 2 (t_j - o_j) \frac{\partial (t_j - o_j)}{\partial o_j}$$

$$\partial E_d / \partial o_j = -(t_j - o_j) \longrightarrow (2)$$

Consider 2nd term in (1) of Case 1.

Since $\boxed{o_j = \sigma(net_j)}$

$$\frac{\partial o_j}{\partial net_j} \rightarrow \text{derivative of the Sigmoid function.}$$

Sigmoid function → useful property

"Its derivative is easily expressed in terms of its output)

$$\frac{d\,\{\sigma(y)}{dy} = \sigma(y) \cdot (1 - \sigma(y))$$

$$\frac{\partial o_j}{\partial net_j} = \frac{\partial \, \sigma(net_j)}{\partial net_j}$$

$$\frac{\partial o_j}{\partial net_j} = o_j(1 - o_j) \longrightarrow (3)$$

$$\left[\frac{d \, \sigma(\overset{o_j}{\overrightarrow{net_j}})}{d\, net_j} = \underline{\sigma(net_j)}(1 - \overset{o_j}{\overrightarrow{\sigma(net_j)}}) \right]$$

Substituting (2), (3) in (1)

$$\frac{\partial E_d}{\partial net_j} = -(t_j - o_j)\, o_j (1 - o_j)$$

ANISHA.C.D|AP|CSE

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$$

$$\Delta w_{ji} = \eta \, (t_j - o_j) \, o_j \, (1 - o_j) \, x_{ji}$$

$$\delta_K = - \frac{\partial E_d}{\partial net_K}$$

CASE 2 : Training Rule for Hidden unit weights

Downstream (j) = outputs

$$\delta_j = o_j (1 - o_j)$$

$$\Delta w_{ji} = \eta \, \delta_j \, x_j$$

ANISHA. C. D | AP | CSE