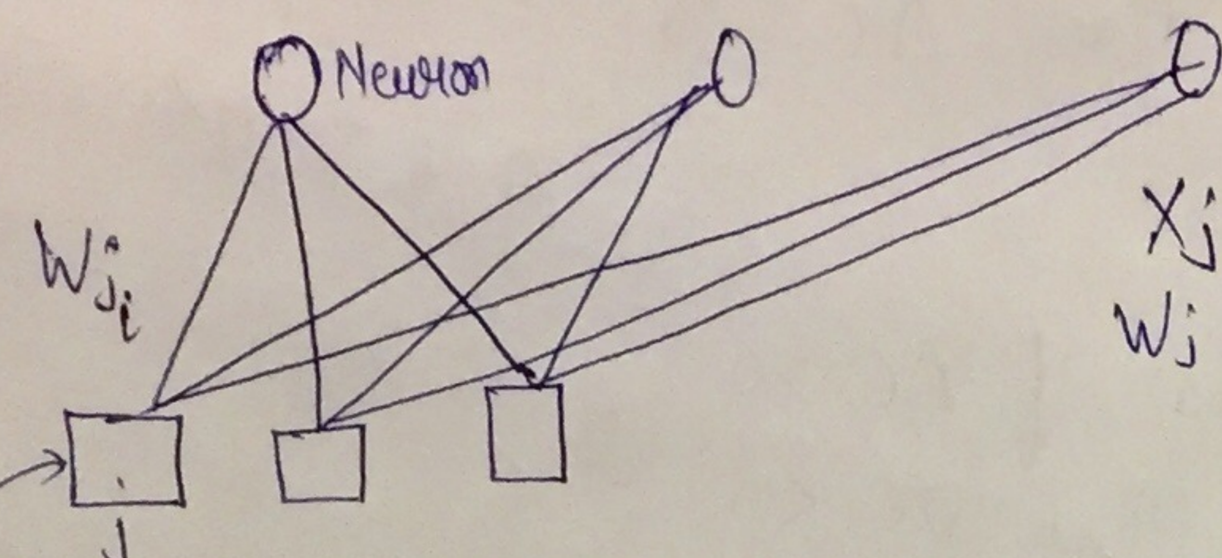
Single layer n/w is called perceptron.



X_j = i/p = Scalar equal to value j

W_j = weight

Activation f^n $\Rightarrow g\left(\sum_j W_{j,i} I_j\right)$

g: can be Sigmoid f^n
can be threshold f^n

⇒ activation value of neuron A for i/p it receives

⇒ goal: g gives correct o/p

⇒ Error $= \frac{1}{2} E_{RMS}^2 = \frac{1}{2}\left(y - h_W(x)\right)^2$ y = actual o/p = constant

$$a_i = h_W(x) = g\left(\sum_j W_j x_j\right)$$

Bring down error by adjusting weights

$\frac{\partial E}{\partial W_j}$ → change in error w.r.t. change in W_j

weight updating f^n - For one

$$\frac{\partial E}{\partial W_j} = \frac{\partial}{\partial W_j}\left\{\frac{1}{2} E_{RMS}^2\right\}$$

$$= E_{RMS}\left\{\frac{\partial E_{RMS}}{\partial W_j}\right\}$$

$$= E_{RMS}\left\{-\frac{\partial}{\partial W_j} g\left(\sum W_j x_j\right)\right\}$$

$$= E_{RMS} * g'(in) * x_j$$

To reduce error we can;

$$W_j \leftarrow W_j + (\alpha E_{RM} * g'(in) * x_j)$$

= Learning Rate

For Sigmoid function: derivative is $g(1-g)$

For threshold function: derivative is not possible.

Majority function → If major i/p's are 1 then 1 else 0.
→ So half of variables are to be examined to know the o/p

Note: Perceptrons can learn majority function better. Because of linearity function interpretation; we have all yes cases at one side of hyperplane & no cases at other side. Individually if we move towards dimensions of these planes then the error is going to go down & gradient descent will decrease.
If o/p's are not independent then perceptrons fail.

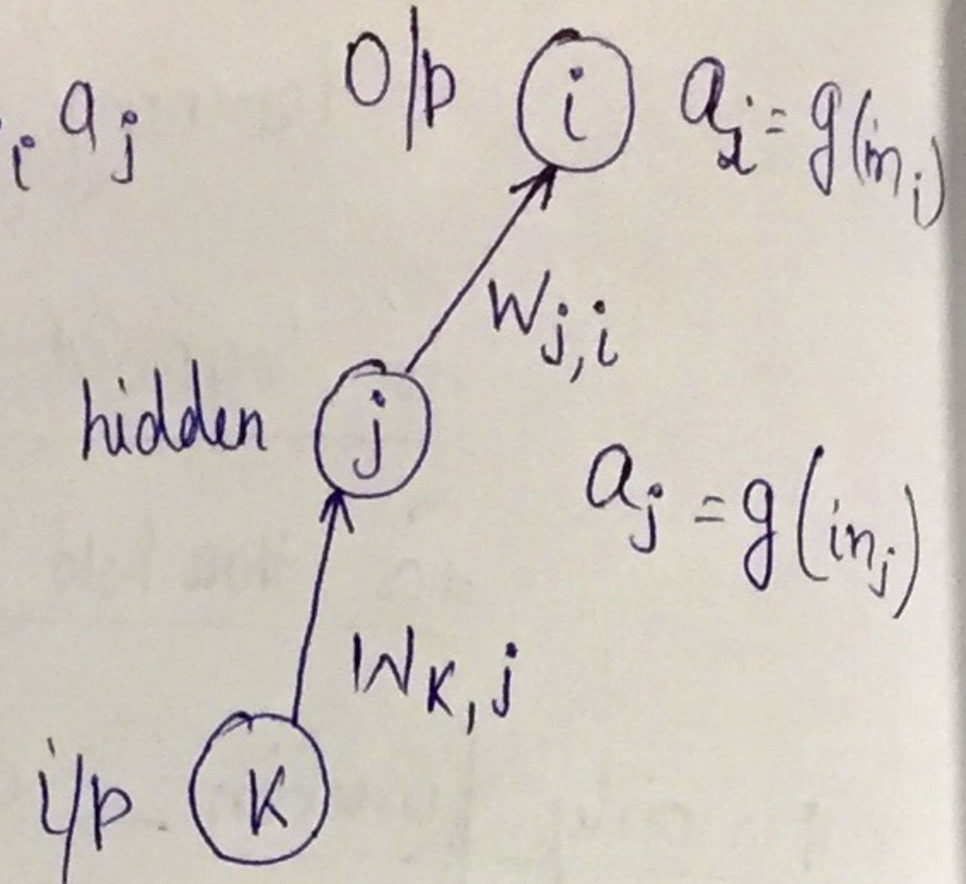
Compute the errors in o/p layer & propagate it back to hidden layer. So we take a fraction of errors from o/p layer & feed it to hidden layer.

$$W_{j,i} \leftarrow W_{j,i} + \alpha a_j \Delta_i$$

where $\Delta_i = Err_i * g'(in_i)$

$$\Delta_j = g'(in_j) \underbrace{\sum_i W_{j,i} \Delta_i}_{Err_j}$$

$$\sum_j W_{j,i} a_j$$



$$W_{k,j} \leftarrow W_{k,j} + \alpha a_k \Delta_j$$

<u>Derivation:</u>

$$\frac{\partial E}{\partial W_{k,j}} = -\sum_i (y_i - a_i) \frac{\partial a_i}{\partial W_{k,j}}$$

$\because W_{k,j}$ affects all the links so nothing can be eliminated

$$= -\sum_i (y_i - a_i) \frac{\partial g(in_i)}{\partial W_{k,j}}$$

$$= -\sum_i \underbrace{(y_i - a_i)\,(g'(in_i))}_{\Delta_i} \frac{\partial (in_i)}{\partial W_{k,j}}$$

$$= -\sum_i \Delta_i \frac{\partial}{\partial W_{k,j}} \left(\sum_{j'} W_{j,i} a_j\right)$$

$$= -\sum_i \Delta_i W_{j,i} \frac{\partial a_j}{\partial W_{k,j}}$$

$$= -\sum_i \Delta_i W_{j,i} g'(in_j) \frac{\partial in_j}{\partial W_{k,j}}$$

$$= -\sum_i \Delta_i W_{j,i} g'(in_j) \frac{\partial}{\partial W_{k,j}} \left(\sum_{k'} W_{k',j} a_{k'}\right)$$

$$= -\sum_i \Delta_i W_{j,i} g'(in_j) a_k$$

$$= -a_k \Delta_j$$