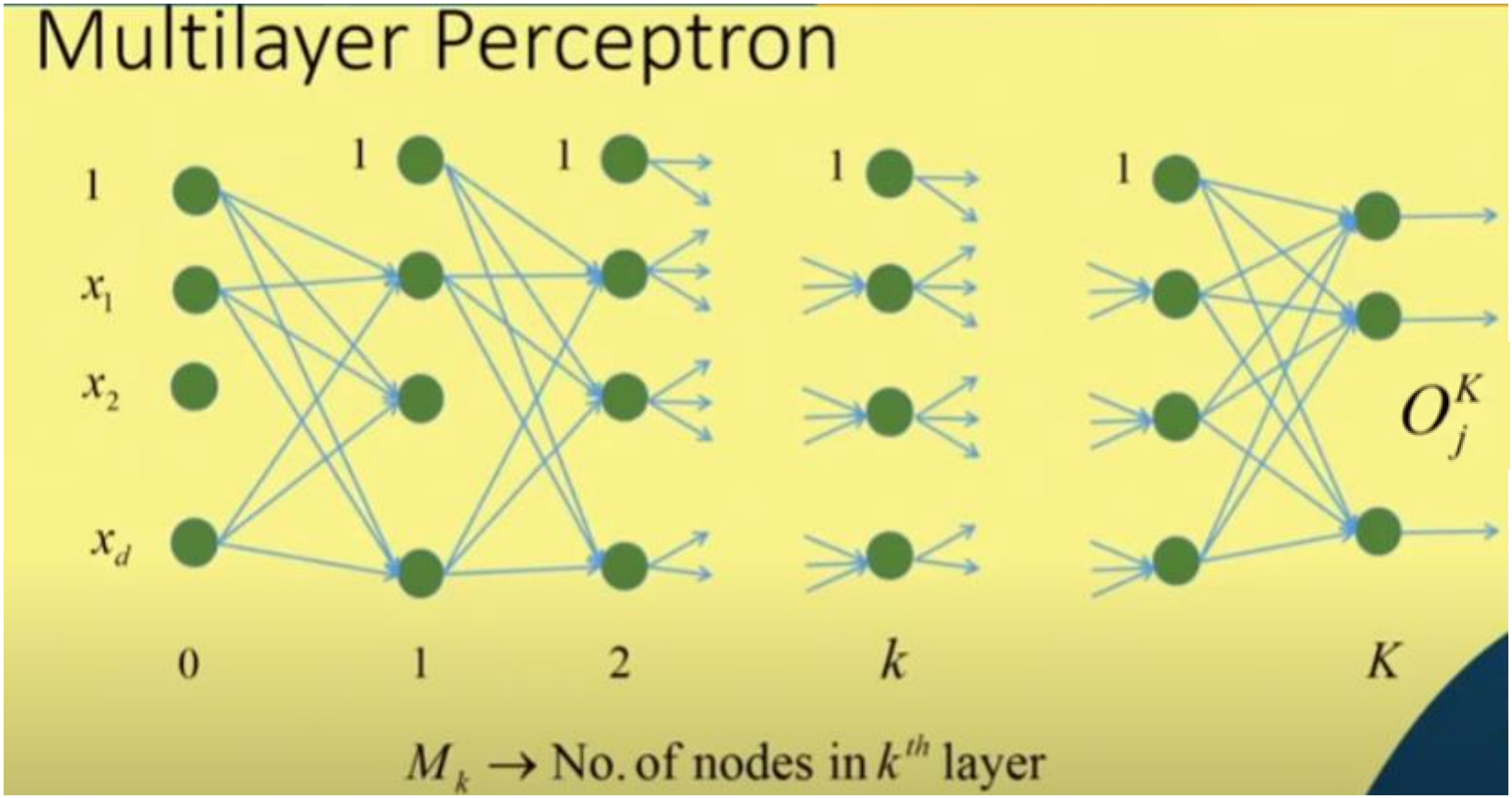
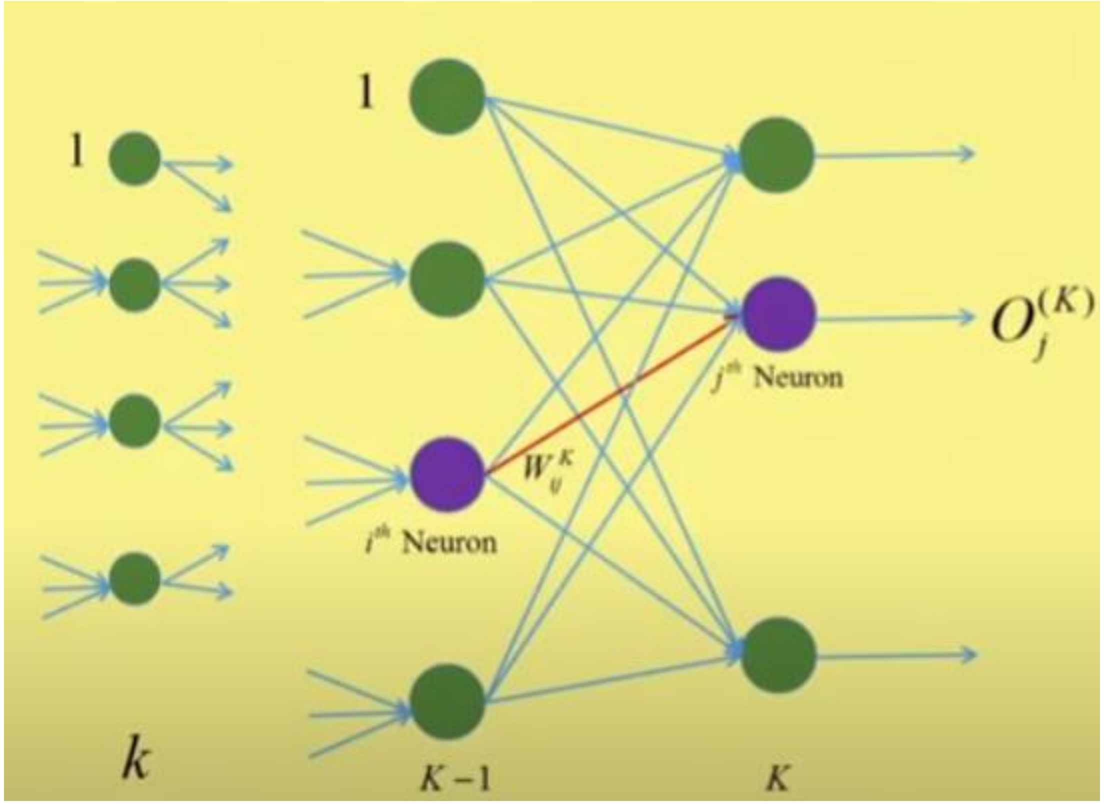
MULTILAYER PERCEPTRON

Umarani Jayaraman

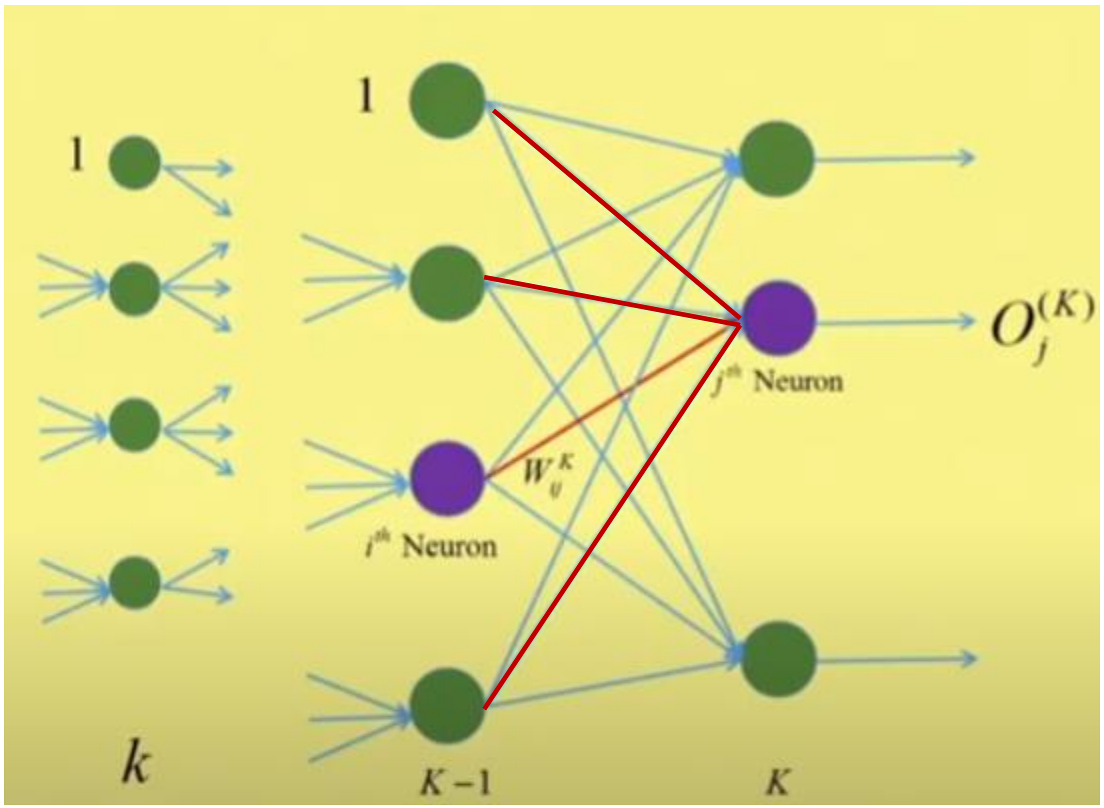
Multilayer Perceptron



Back Propagation Learning: Output layer



Back Propagation Learning: Output layer



$$O_j^K = \frac{1}{1+e^{-\theta_j^K}} \qquad \theta_j^K = \sum_{i=1}^{M_{K-1}} W_{ij}^K x_i^{K-1}$$

$$E = \frac{1}{2}\sum_{j=1}^{M_K}\left(O_j^K - t_j\right)^2$$

Back Propagation Learning: Output layer

$$\frac{\partial E}{\partial W_{ij}^K} = \frac{\partial E}{\partial O_j^K} \cdot \frac{\partial O_j^K}{\partial \theta_j^K} \cdot \frac{\partial \theta_j^K}{\partial W_{ij}^K}$$

$$= (O_j^K - t_j) O_j^K (1 - O_j^K) O_i^{K-1}$$

Let $\delta_j^K = O_j^K (1 - O_j^K)(O_j^K - t_j)$

$$\Rightarrow \frac{\partial E}{\partial W_{ij}^K} = \delta_j^K O_i^{K-1}$$

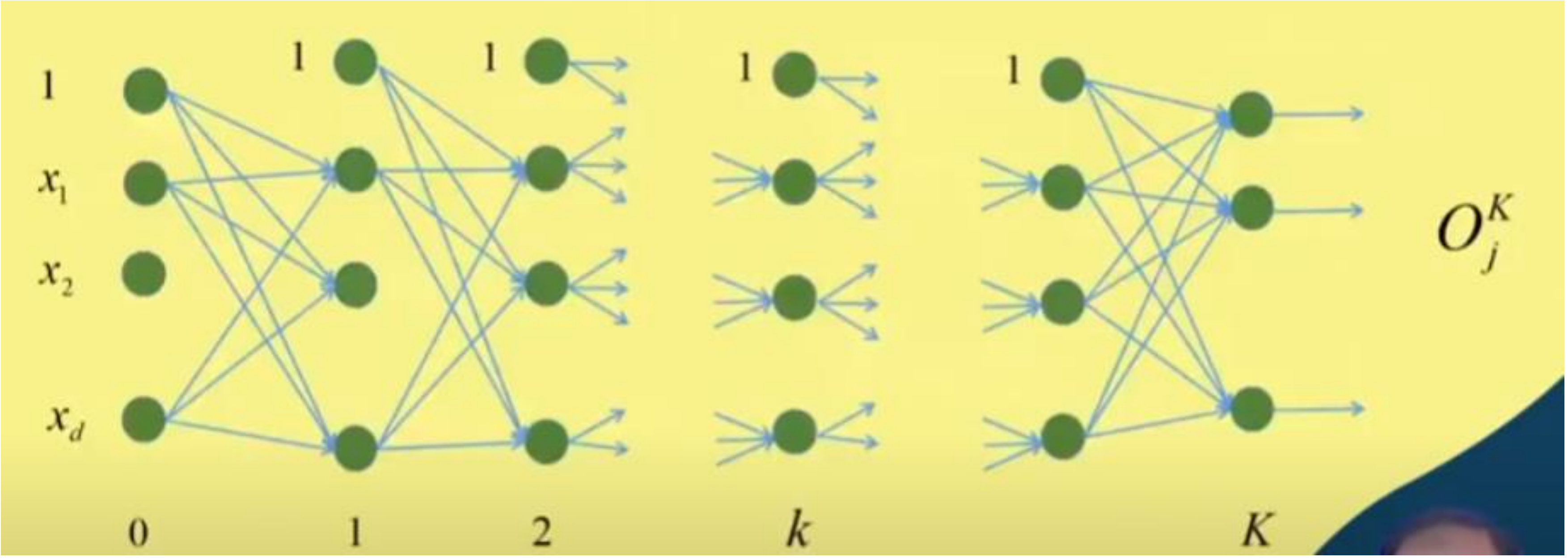
$$O_j^K = \frac{1}{1 + e^{-\theta_j^K}} \qquad \theta_j^K = \sum_{i=1}^{M_{K-1}} W_{ij}^K O_i^{K-1}$$

Weight updation rule

Output Layer

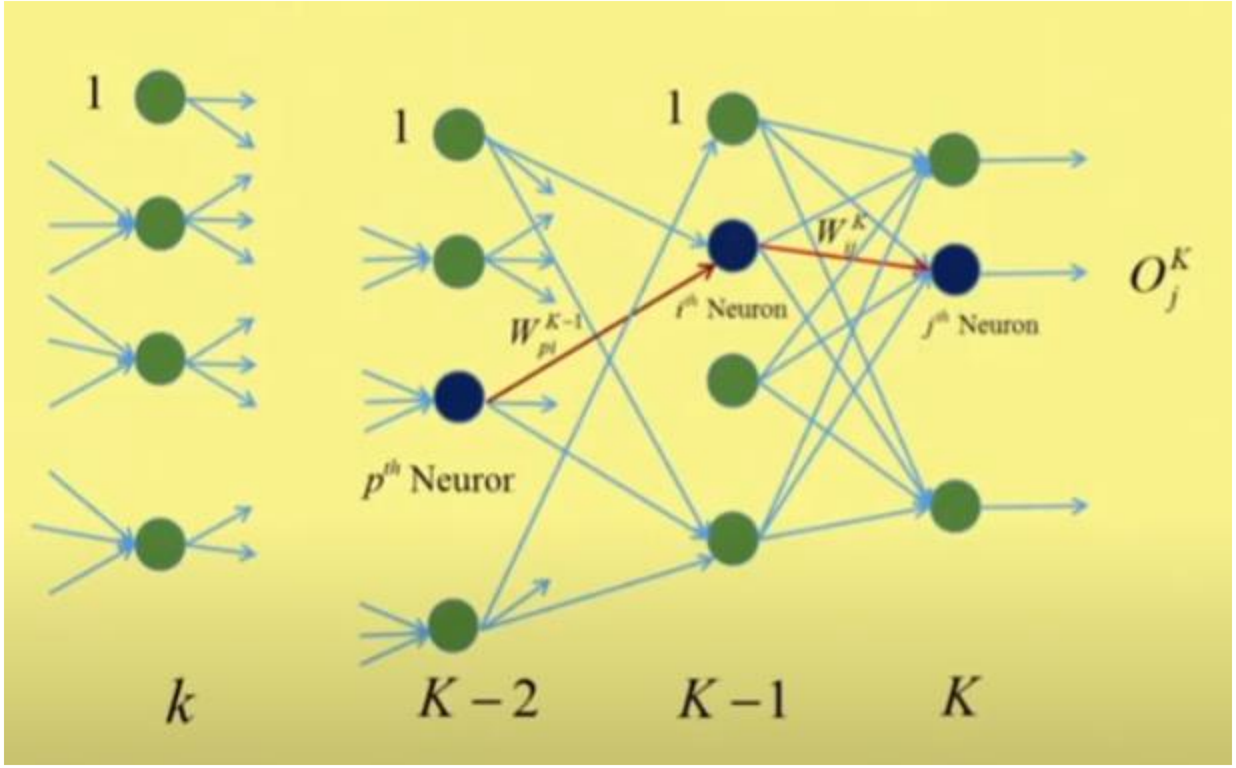
$$W_{ij}^K \leftarrow W_{ij}^K - \eta \delta_j^K O_i^{K-1}$$

Update the weights at the hidden layer

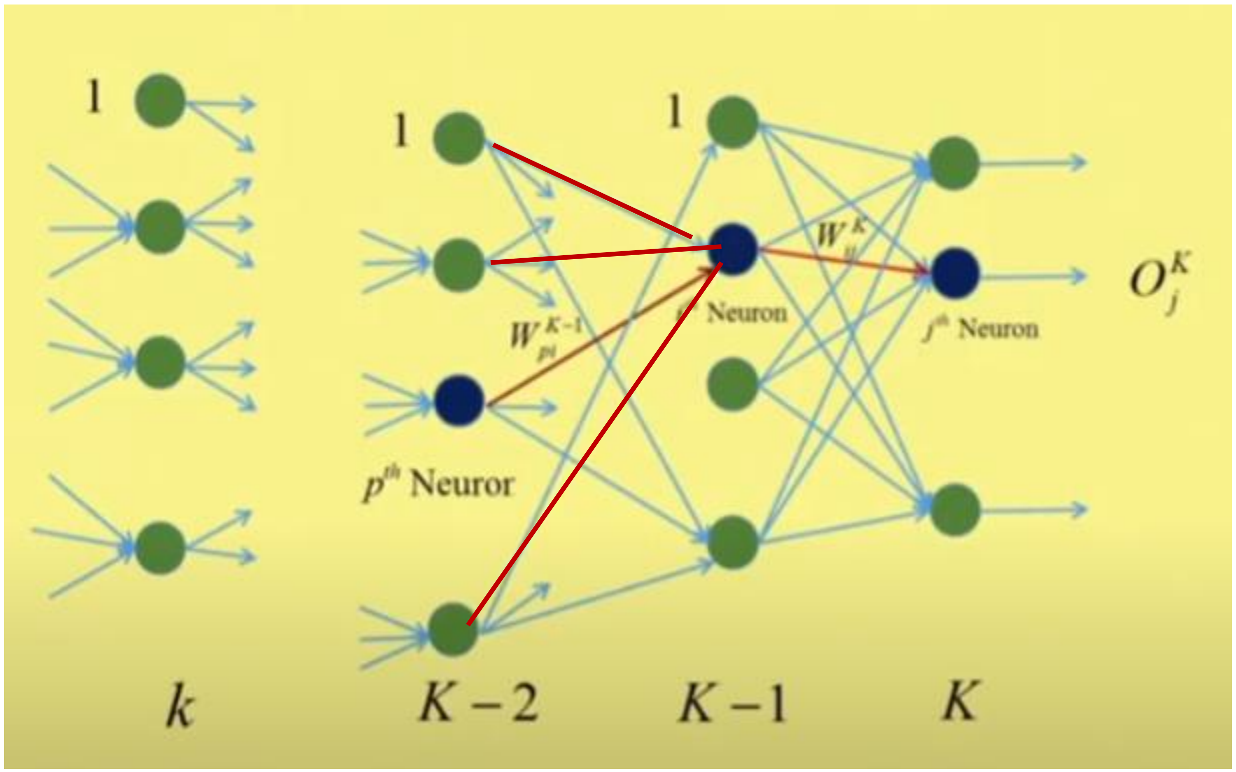


$M_k \rightarrow$ No. of nodes in k^{th} layer

Back Propagation Layer- Hidden Layer



Back Propagation Layer- Hidden Layer



$$E = \frac{1}{2}\sum_{j=1}^{M_K}\left(O_j^K - t_j\right)^2$$

Back Propagation Learning- Hidden Layer

Find W_{pi}^{K-1} that minimizes $E = \frac{1}{2}\sum_{j=1}^{M_K}\left(O_j^K - t_j\right)^2$

Gradient Descent $\Rightarrow \frac{\partial E}{\partial W_{pi}^{K-1}}$

Thank you