NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas

Department : E & ECE, IIT Kharagpur

Topic

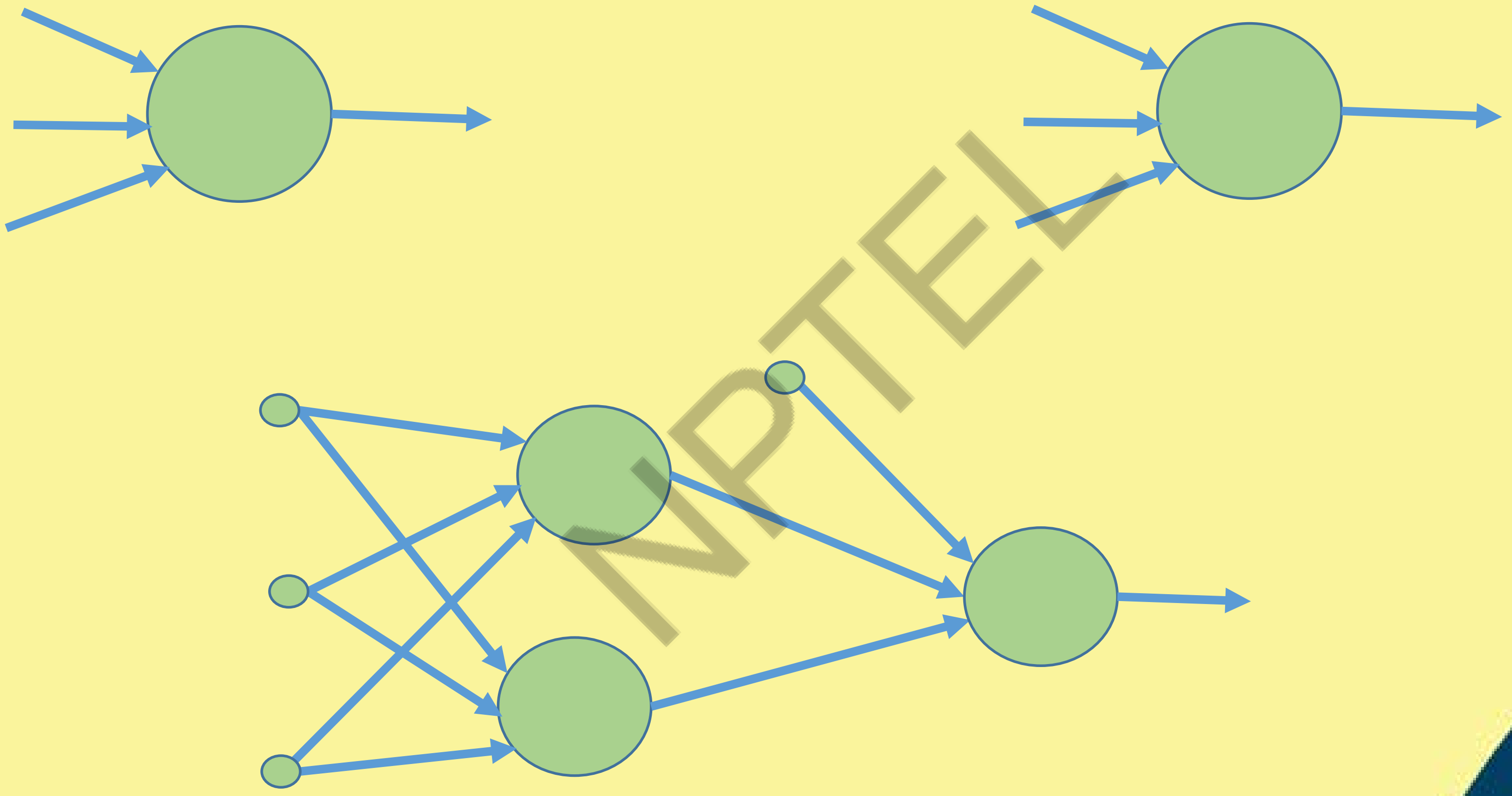
Lecture 21: Multilayer Perceptron

CONCEPTS COVERED

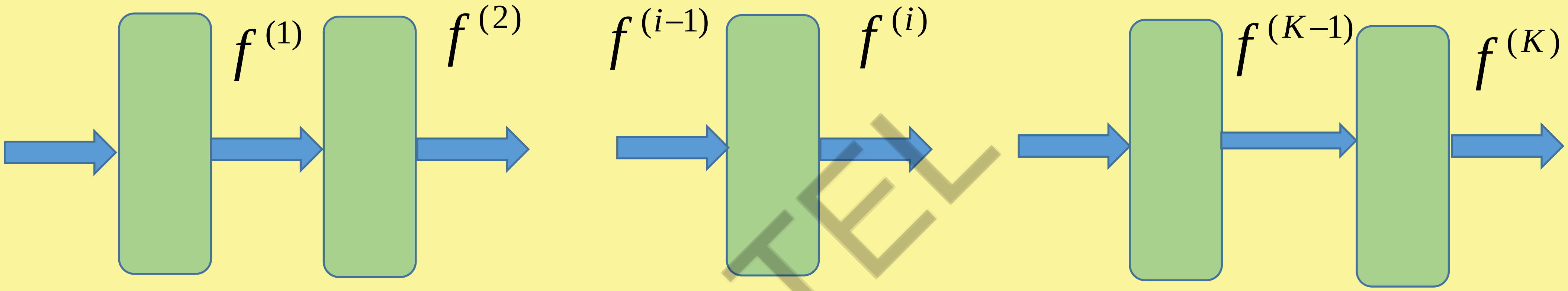
Concepts Covered:

- Neural Network
  - AND Logic
  - OR Logic
  - XOR Logic
- Feed Forward NN
- Back Propagation Learning

AND/ OR/ XOR

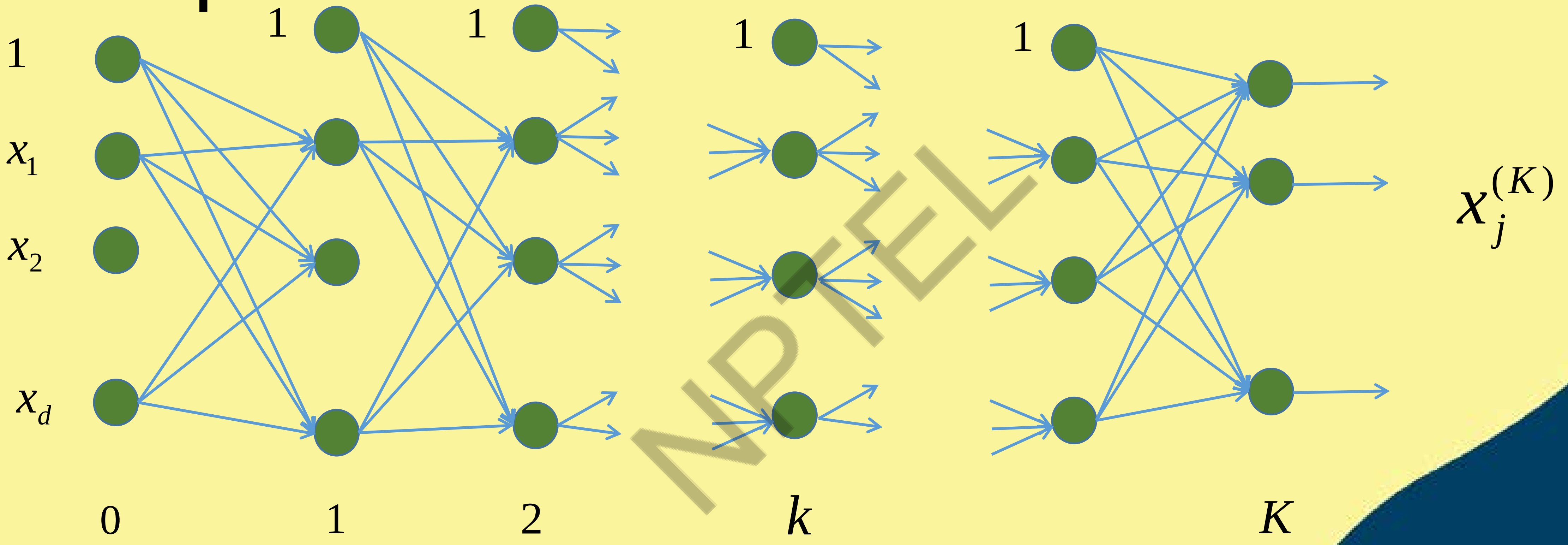


Neural Network Function



$$f^{(K)}(f^{(K-1)}\ldots\ldots(f^{(i)}\ldots.(f^{(2)}(f^{(1)}(X)))))$$

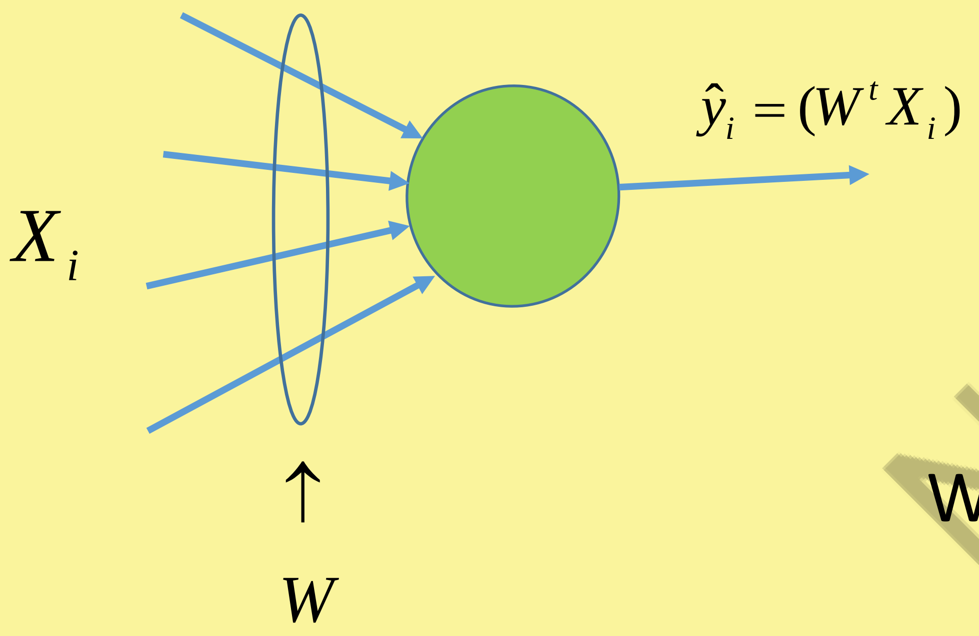
Multilayer Perceptron



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

Back Propagation Learning

Single Layer Network- Single Output without nonlinearity



$$E = \frac{1}{2}\sum_{i=1}^{N}(W^t X_i - y_i)^2 = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

$$\nabla_W E = \sum_{i=1}^{N}(\hat{y}_i - y_i)X_i$$

Weight updation rule

$$W \leftarrow W - \eta\sum_{i=1}^{N}(\hat{y}_i - y_i)X_i$$

NPTEL ONLINE CERTIFICATION COURSES

Thank you

NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas
Department : E & ECE, IIT Kharagpur

Topic

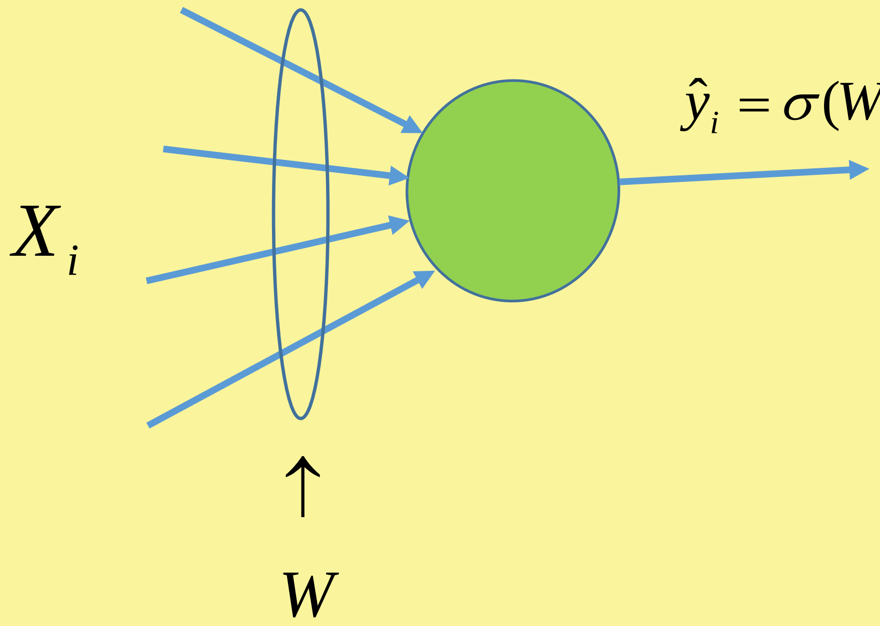
Lecture 22: Multilayer Perceptron -II

CONCEPTS COVERED

Concepts Covered:

- Neural Network
- Feed Forward NN
- Back Propagation Learning

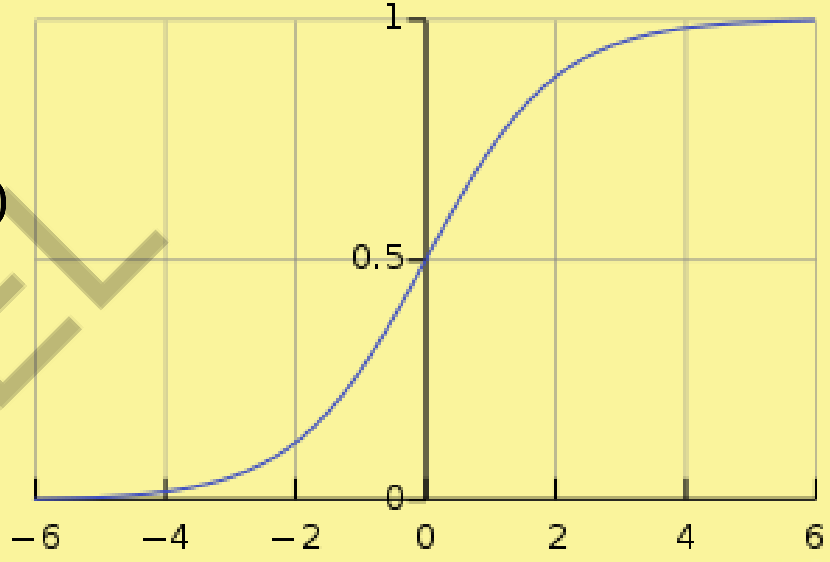
Single Layer Network- Single Output with nonlinearity



$\hat{y}_i = \sigma(W^t X)$

$\uparrow$

$\sigma(W^t X)$



$W^t X \rightarrow$

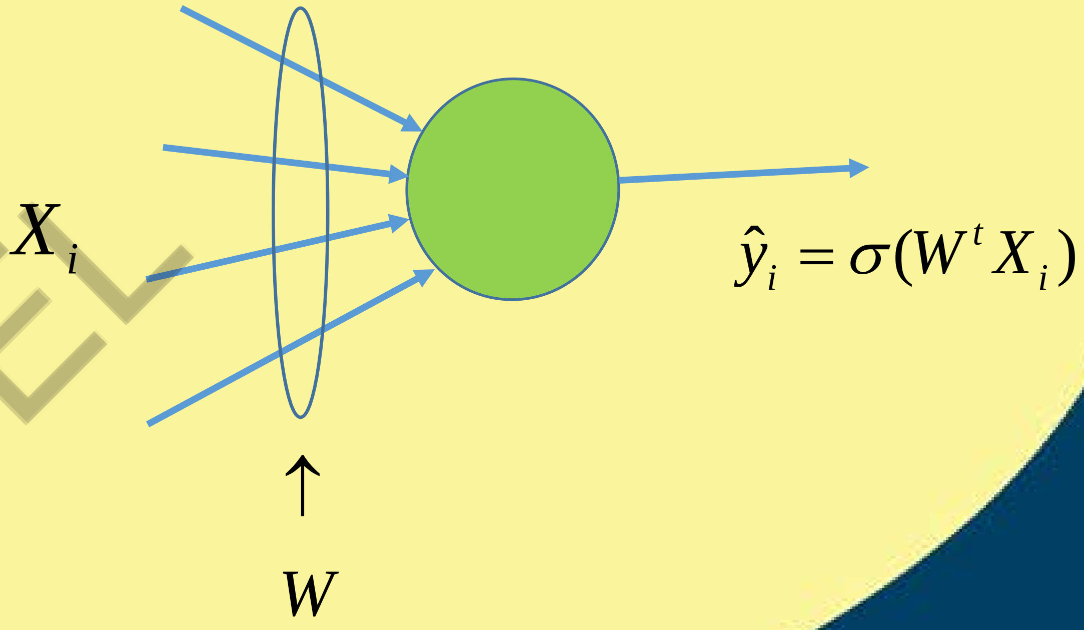
Single Layer Network- Single Output with nonlinearity

$$E = \frac{1}{2}(\hat{y}_i - y_i)^2 = \frac{1}{2}(\sigma(W^t X_i) - y_i)^2$$

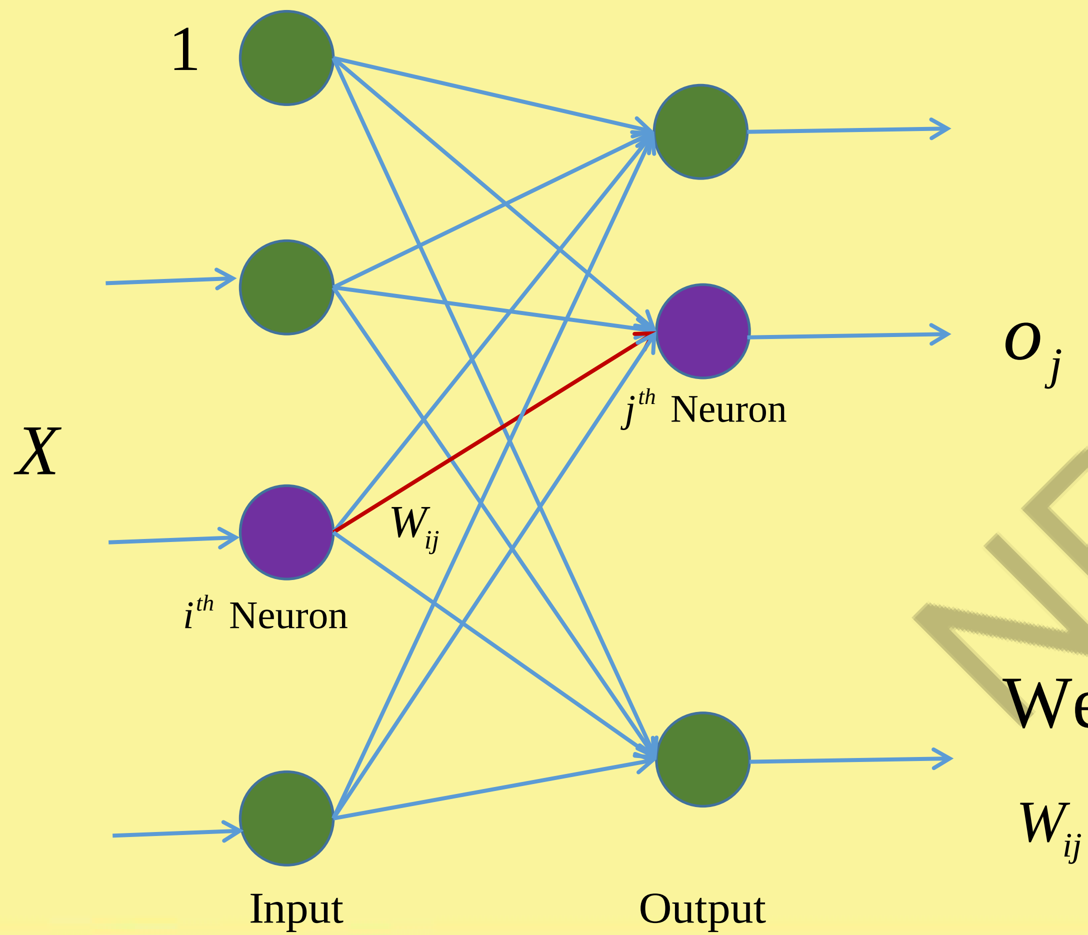
$$\nabla_W E = \hat{y}_i(1-\hat{y}_i)(\hat{y}_i - y_i)X_i$$

Weight updation rule $\Rightarrow$

$$W \leftarrow W - \eta\hat{y}_i(1-\hat{y}_i)(\hat{y}_i - y_i)X_i$$



Back Propagation Learning:- Single Layer Multiple Output



$$o_j = \frac{1}{1+e^{-\theta_j}} \qquad \theta_j = \sum_{i=1}^{D} W_{ij} x_i$$

$$E = \frac{1}{2}\sum_{j=1}^{M}\left(o_j - t_j\right)^2$$

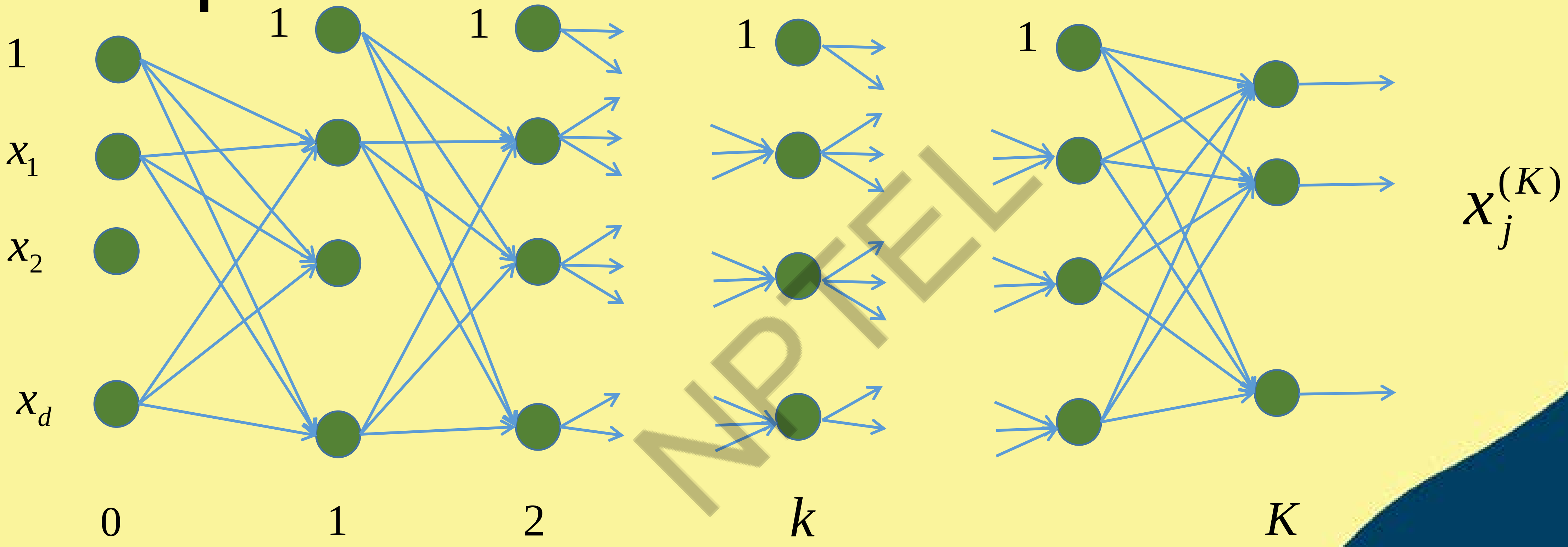
$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial o_j}\cdot\frac{\partial o_j}{\partial \theta_j}\cdot\frac{\partial \theta_j}{\partial W_{ij}}$$

$$= (o_j - t_j) o_j (1 - o_j) x_i$$

Weight updation rule $\Rightarrow$

$$W_{ij} \leftarrow W_{ij} - \eta (o_j - t_j) o_j (1 - o_j) x_i$$

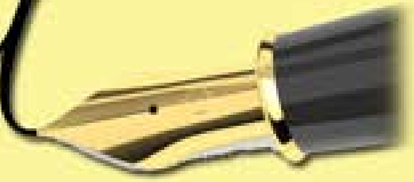
Multilayer Perceptron



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

NPTEL ONLINE CERTIFICATION COURSES

Thank you



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas

Department : E & ECE, IIT Kharagpur

Topic

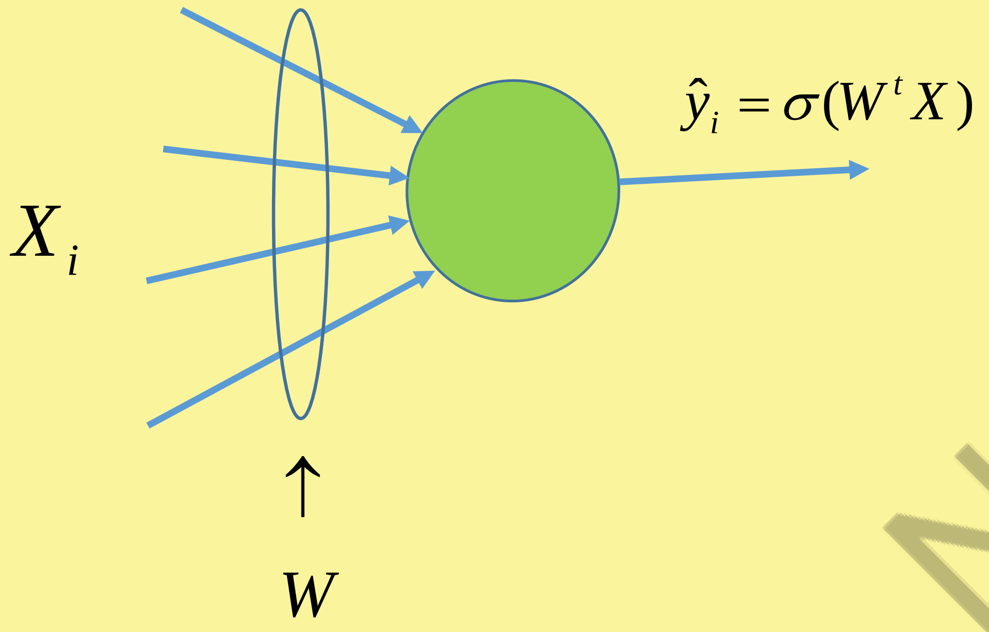
Lecture 23: Back Propagation Learning

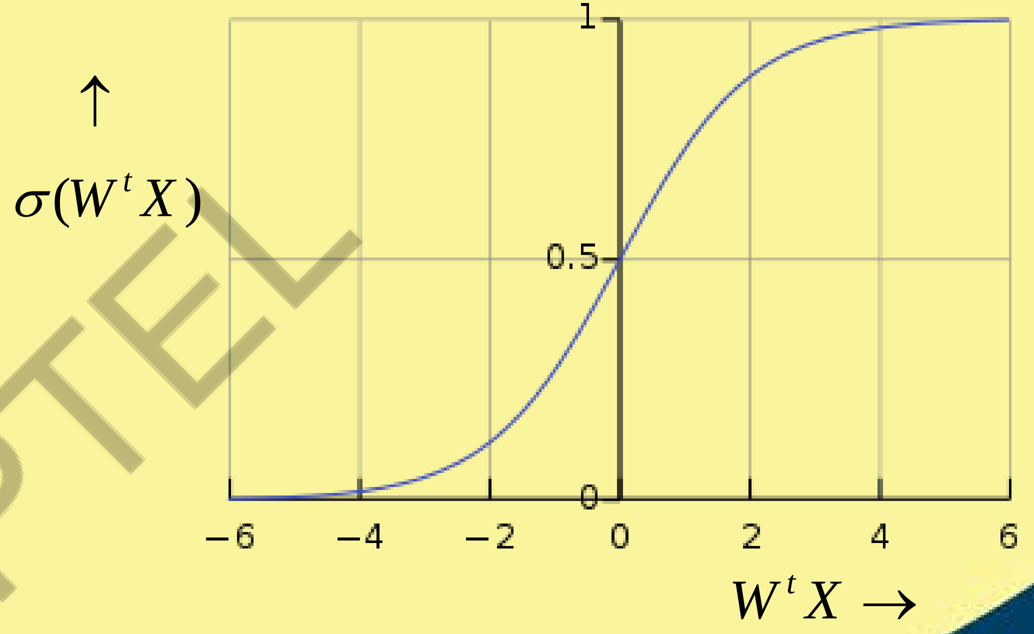
CONCEPTS COVERED

Concepts Covered:

- ❑ Learning in Single Layer Perceptron
- ❑ Back Propagation Learning in MLP

Single Layer Network- Single Output with nonlinearity

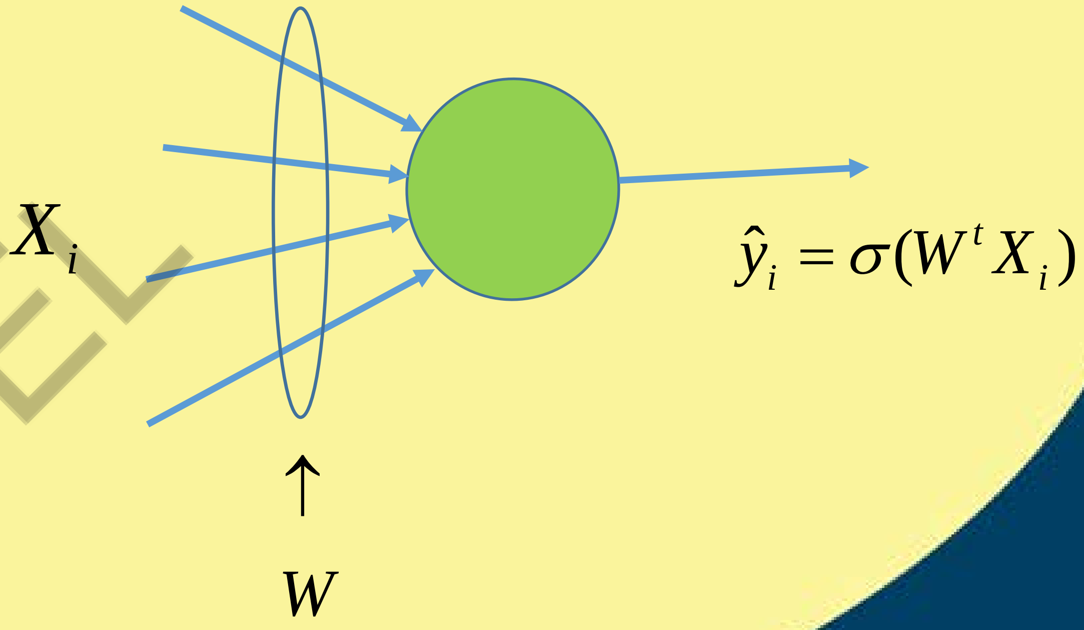




Single Layer Network- Single Output with nonlinearity

$$E = \frac{1}{2}(\hat{y}_i - y_i)^2 = \frac{1}{2}(\sigma(W^t X_i) - y_i)^2$$

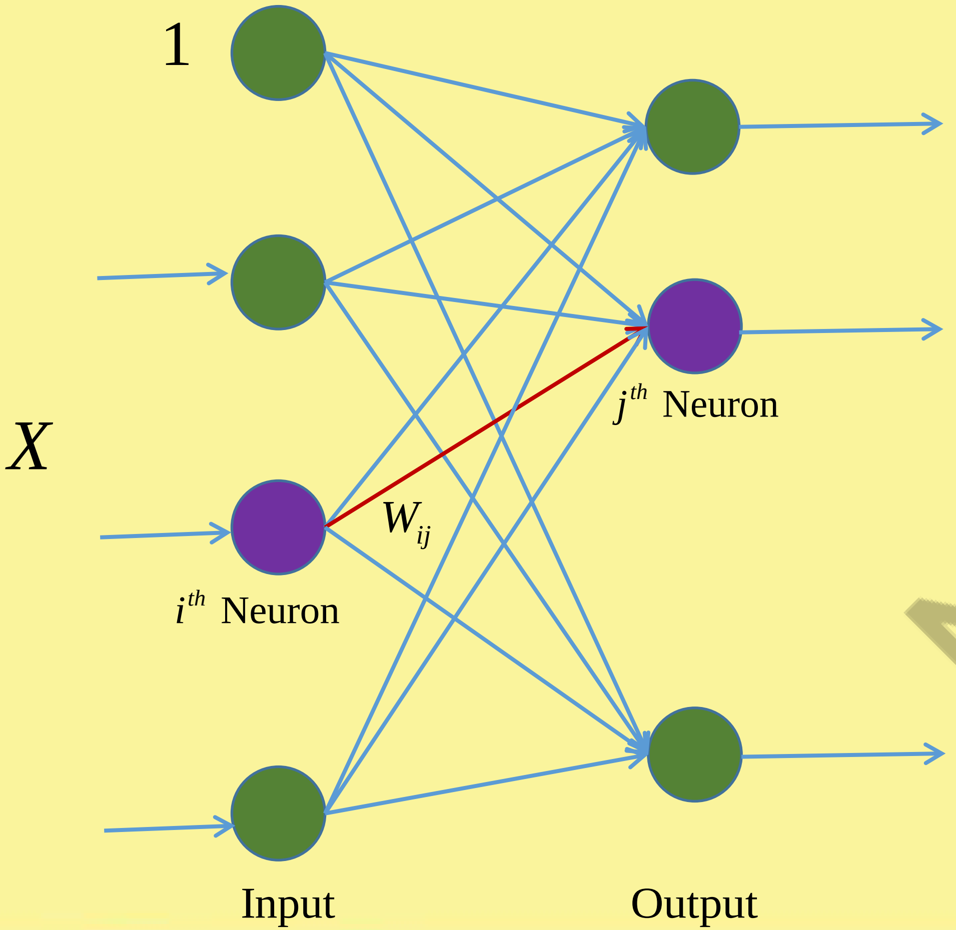
$$\nabla_W E = \hat{y}_i(1-\hat{y}_i)(\hat{y}_i - y_i)X_i$$



Weight updation rule $\Rightarrow$

$$W \leftarrow W - \eta \hat{y}_i(1-\hat{y}_i)(\hat{y}_i - y_i)X_i$$

Back Propagation Learning:- Single Layer Multiple Output



$$o_j = \frac{1}{1+e^{-\theta_j}} \qquad \theta_j = \sum_{i=1}^{D} W_{ij} x_i$$

$$E = \frac{1}{2}\sum_{j=1}^{M}\left(o_j - t_j\right)^2$$

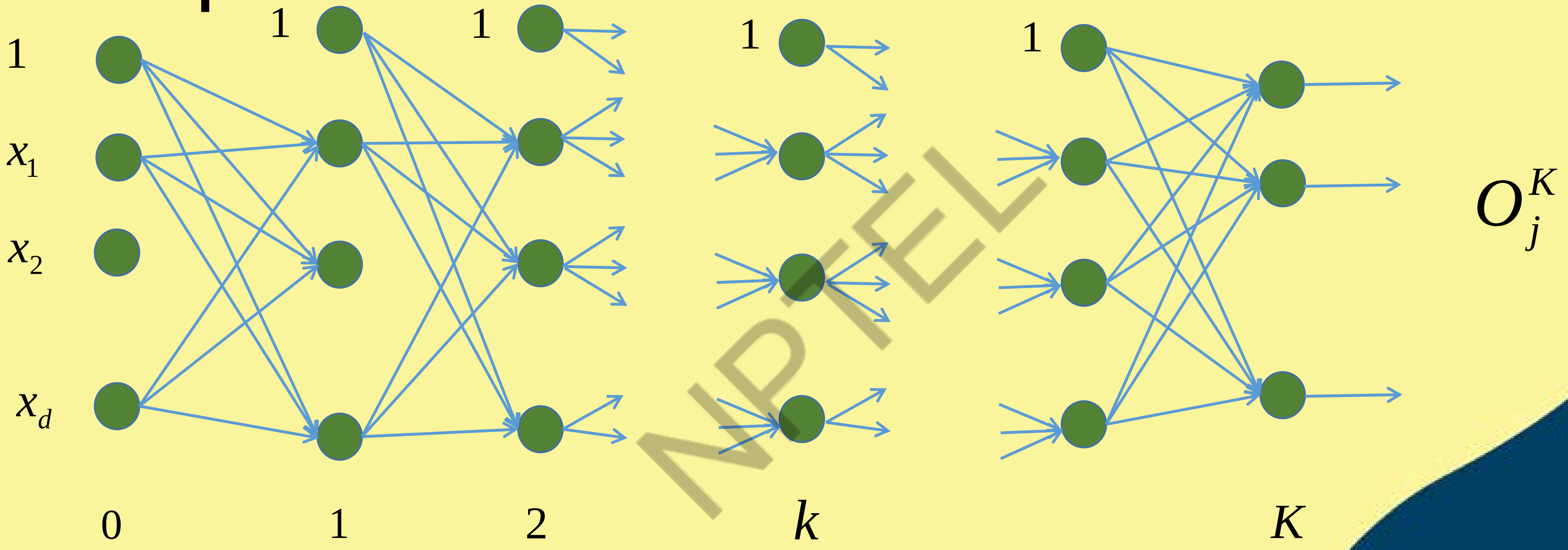
$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial o_j}\cdot\frac{\partial o_j}{\partial \theta_j}\cdot\frac{\partial \theta_j}{\partial W_{ij}}$$

$$= (o_j - t_j)o_j(1-o_j)x_i$$

Weight updation rule $\Rightarrow$

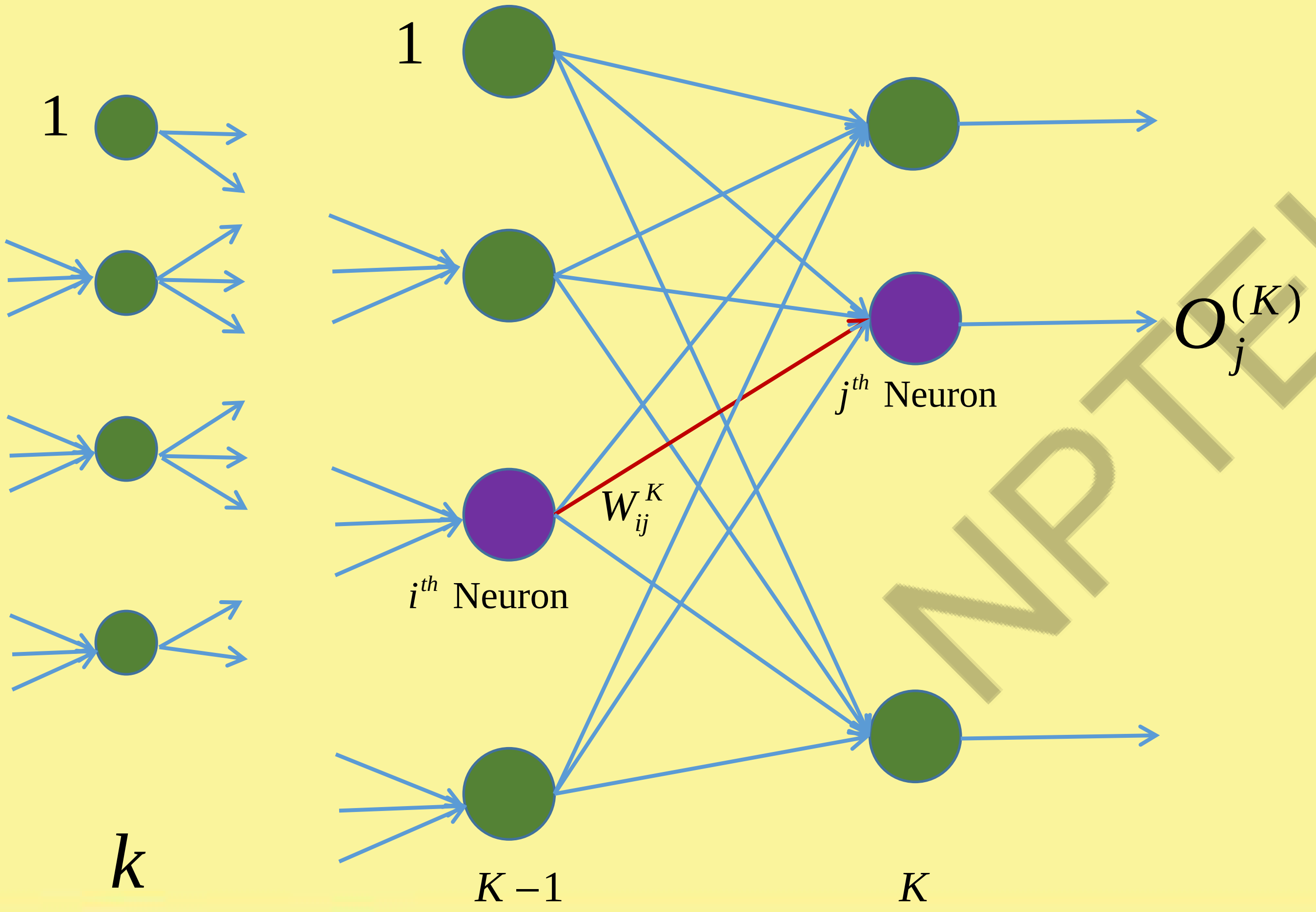
$$W_{ij} \leftarrow W_{ij} - \eta(o_j - t_j)o_j(1-o_j)x_i$$

Multilayer Perceptron



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

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

Gradient Descent $\frac{\partial E}{\partial W_{ij}^{K}}$

Back Propagation Learning:- Output Layer

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

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

$$\text{Let} \quad \delta_j^K = O_j^K (1 - O_j^K)(O_j^K - t_j)$$

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

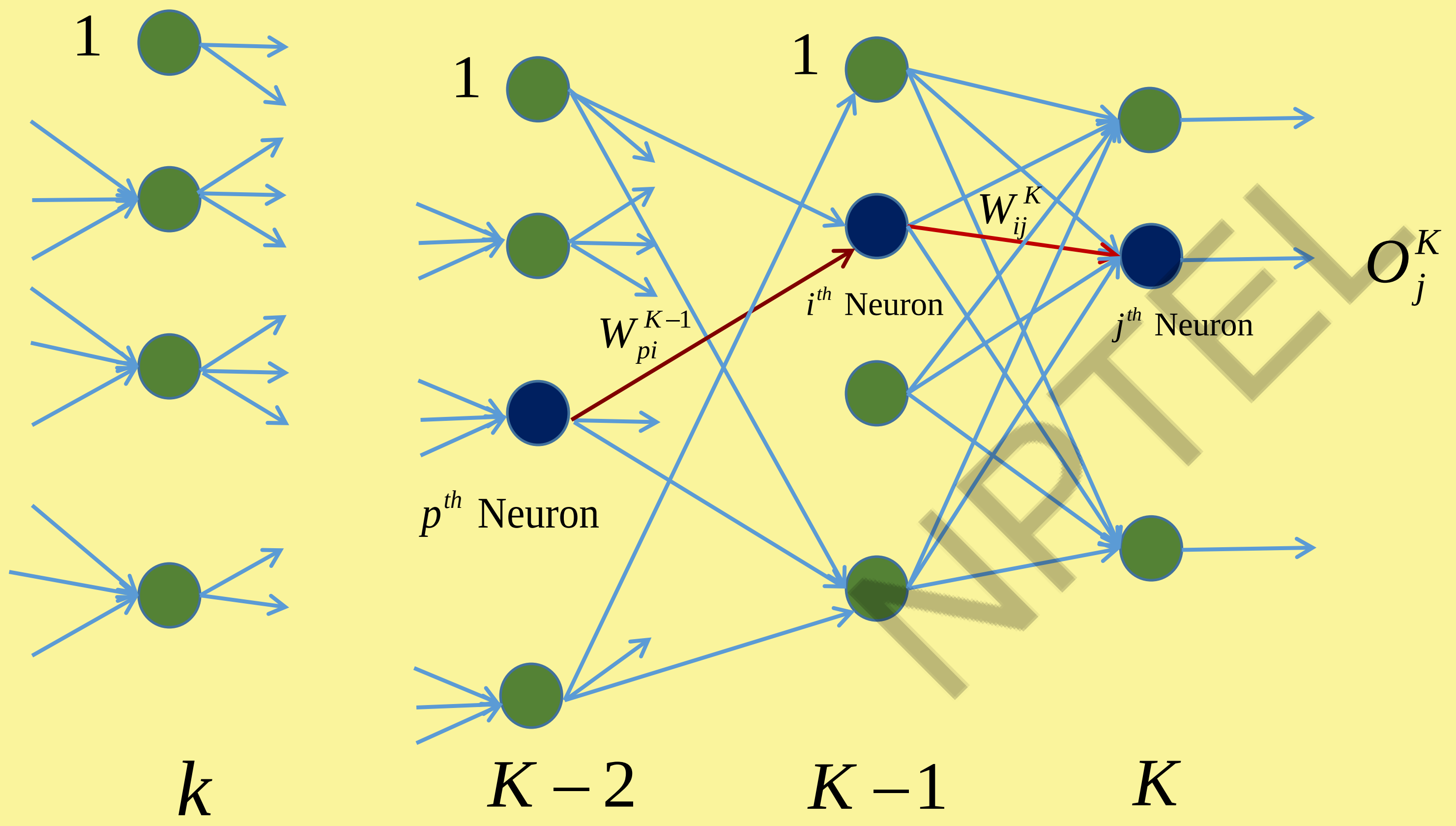
$$O_j^K = \frac{1}{1 + e^{-\theta_j^K}} \qquad \theta_j^K = \sum_{i=1}^{M_{K-1}} W_{ij}^K O_i^{K-1}$$

Weight updation rule

Output Layer

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

Back Propagation Learning:- Hidden Layer



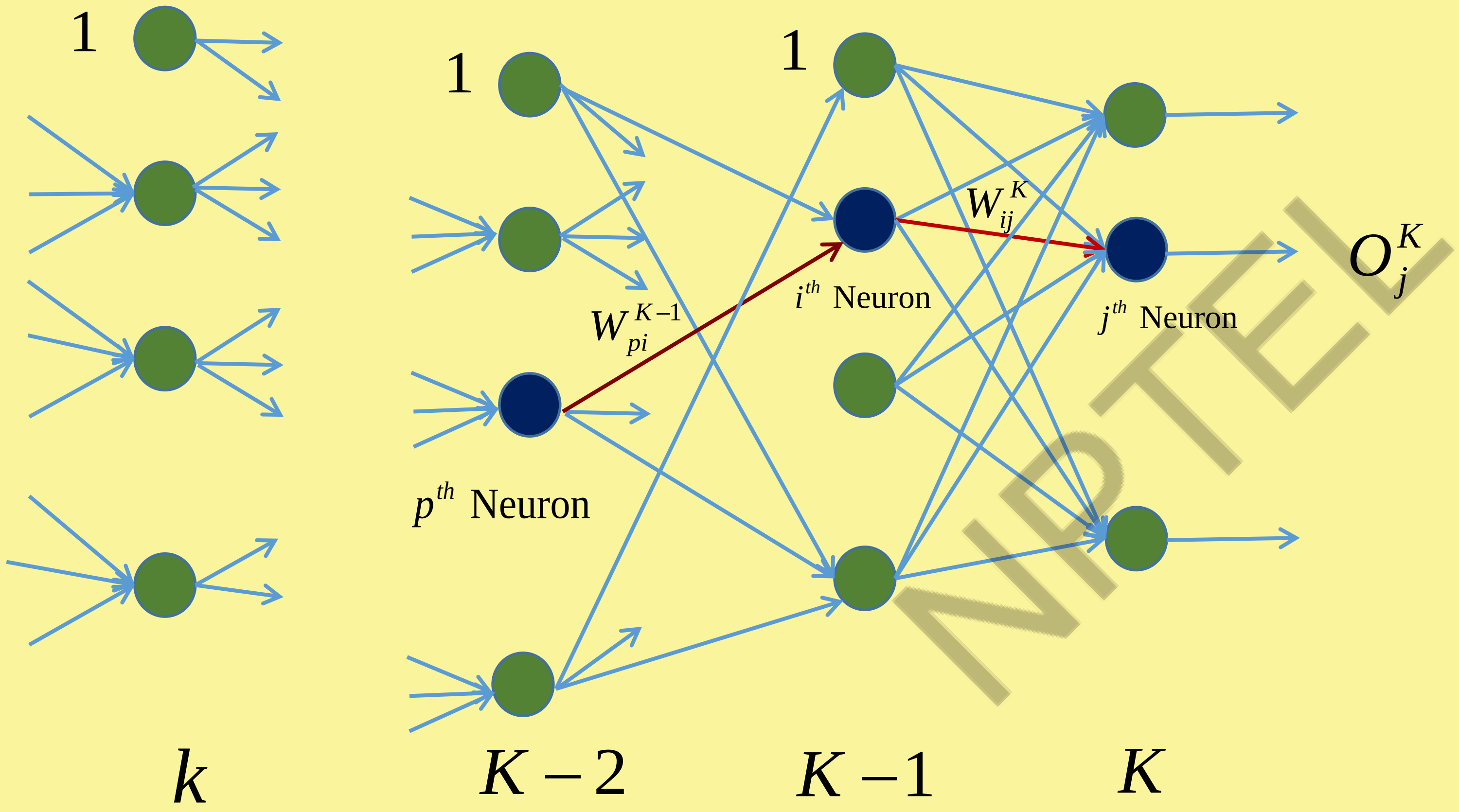
$$E = \frac{1}{2}\sum_{j=1}^{M_K}\left(O_j^K - t_j\right)^2$$

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 $\Rightarrow \dfrac{\partial E}{\partial W_{pi}^{K-1}}$

Back Propagation Learning:- Hidden Layer



$$O_i^{K-1} = \frac{1}{1+e^{-\theta_i^{K-1}}}$$

$$\theta_i^{K-1} = \sum_{p=1}^{M_{K-2}} W_{pi}^{K-1} O_p^{K-2}$$

Back Propagation Learning:- Hidden Layer

$$\frac{\partial E}{\partial W_{pi}^{K-1}} = \frac{\partial E}{\partial O_i^{K-1}} \cdot \frac{\partial O_i^{K-1}}{\partial W_{pi}^{K-1}}$$

$$= \frac{\partial E}{\partial O_i^{K-1}} \cdot \frac{\partial O_i^{K-1}}{\partial \theta_i^{K-1}} \cdot \frac{\partial \theta_i^{K-1}}{\partial W_{pi}^{K-1}}$$

$$= \frac{\partial E}{\partial O_i^{K-1}} . O_i^{K-1}(1 - O_i^{K-1}) . O_p^{K-2}$$

$$O_i^{K-1} = \frac{1}{1 + e^{-\theta_i^{K-1}}}$$

$$\theta_i^{K-1} = \sum_{p=1}^{M_{K-2}} W_{pi}^{K-1} O_p^{K-2}$$

Back Propagation Learning:- Hidden Layer

$$\frac{\partial E}{\partial O_i^{K-1}} = \frac{\partial E}{\partial O_j^K} \cdot \frac{\partial O_j^K}{\partial \theta_j^K} \cdot \frac{\partial \theta_j^K}{\partial O_i^{K-1}}$$

$$= \sum_{j=1}^{M_K} (O_j^k - t_j) O_j^K (1 - O_j^K) W_{ij}^K$$

$$= \sum_{j=1}^{M_K} \partial_j^K W_{ij}^K$$

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

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

$$\boxed{\delta_j^K = O_j^K (1 - O_j^K)(O_j^K - t_j)}$$

Back Propagation Learning:- Hidden Layer

$$\frac{\partial E}{\partial W_{pi}^{K-1}} = \frac{\partial E}{\partial x_i^{K-1}} . O_i^{K-1}(1-O_i^{K-1}).O_p^{K-2} = O_i^{K-1}(1-O_i^{K-1}).O_p^{K-2}\sum_{j=1}^{M_K}\partial_j^K W_{ij}^K$$

Putting $\delta_i^{K-1} = O_i^{K-1}(1-O_i^{K-1})\sum_{j=1}^{M_K}\partial_j^K W_{ij}^K$

Weight updation rule

Last but Output Layer

$$W_{pi}^{K-1} \leftarrow W_{pi}^{K-1} - \eta\delta_i^{K-1} O_p^{K-2}$$

Back Propagation Learning:- any Hidden Layer

For any hidden layer weight W_{ij}^{k}

Putting $\delta_i^k = O_i^k(1 - O_i^k)\sum_{j=1}^{M_{k+1}} \partial_j^{k+1} W_{ij}^{k+1}$

Weight updation rule

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

NPTEL ONLINE CERTIFICATION COURSES

Thank you

NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas

Department : E & ECE, IIT Kharagpur

Topic

Lecture 24: Cross Entropy Loss

CONCEPTS COVERED

Concepts Covered:

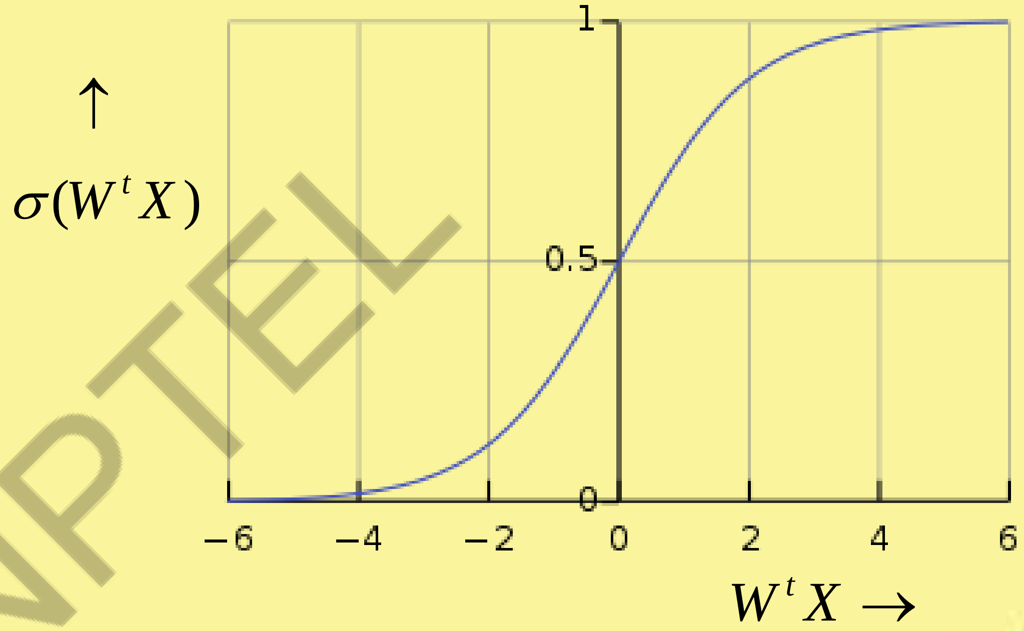
- Back Propagation Learning in MLP
  - Squared Error
- Cross Entropy Loss

Problem with Quadratic Loss Function

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

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

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

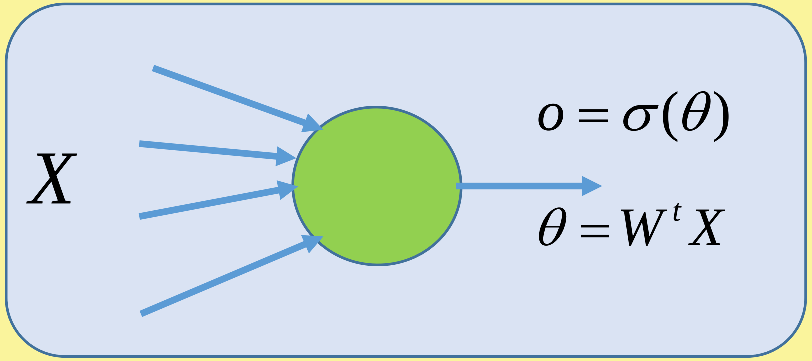


Cross Entropy Loss

Cross Entropy Loss- Two Class Problem

$o \Rightarrow$ likelihood that y is 1

$(1-o) \Rightarrow$ likelihood that y is 0



Likelihood that is to be maximized $\Rightarrow o^y (1-o)^{(1-y)}$

Loglikelihood $\Rightarrow y \log o + (1-y)\log(1-o)$

Cross Entropy Loss

$$\text{Minimize} \Rightarrow C = -\frac{1}{N}\sum_{\forall X}\left[y\log o + (1-y)\log(1-o)\right]$$

$$\frac{\partial C}{\partial W_i} = -\frac{1}{N}\sum_{\forall X}\left[\frac{y}{\sigma(\theta)} - \frac{(1-y)}{1-\sigma(\theta)}\right]\frac{\partial \sigma(\theta)}{\partial W_i}$$

$$= -\frac{1}{N}\sum_{\forall X}\left[\frac{y}{\sigma(\theta)} - \frac{(1-y)}{1-\sigma(\theta)}\right]\frac{\partial \sigma(\theta)}{\partial \theta}\cdot\frac{\partial \theta}{\partial W_i}$$

Cross Entropy Loss

$$\frac{\partial C}{\partial W_i} = -\frac{1}{N}\sum_{\forall X}\left[\frac{y}{\sigma(\theta)} - \frac{(1-y)}{1-\sigma(\theta)}\right]\frac{\partial \sigma(\theta)}{\partial \theta}\cdot\frac{\partial \theta}{\partial W_i}$$

$$= -\frac{1}{N}\sum_{\forall X}\left[\frac{y}{\sigma(\theta)} - \frac{(1-y)}{1-\sigma(\theta)}\right]\frac{\partial \sigma(\theta)}{\partial \theta}\cdot\frac{\partial \theta}{\partial W_i}$$

$$= -\frac{1}{N}\sum_{\forall X}\left[\frac{y-\sigma(\theta)}{\sigma(\theta)(1-\sigma(\theta))}\right]\sigma(\theta)(1-\sigma(\theta).x_i$$

$$= \frac{1}{N}\sum_{\forall X} x_i(\sigma(\theta)-y)$$

$$= \frac{1}{N}\sum_{\forall X} x_i(o-y)$$

Cross Entropy Loss- Multiclass Problem

$$C = -\frac{1}{N}\sum_{\forall X}\sum_{j}\left[y_j \log o_j^K + (1-y_i)\log(1-o_j^K)\right]$$

$$\frac{\partial C}{\partial W_{ij}^K} = \frac{1}{N}\sum_{\forall X} o_i^{K-1}(o_j^K - y_j)$$

$$W_{ij}^K \leftarrow W_{ij}^K - \eta\frac{1}{N}\sum_{\forall X} o_i^{K-1}(o_j^K - y_j)$$

NPTEL ONLINE CERTIFICATION COURSES

Thank you

NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas

Department : E & ECE, IIT Kharagpur

Topic

Lecture 25: Back propagation Learning – Examples

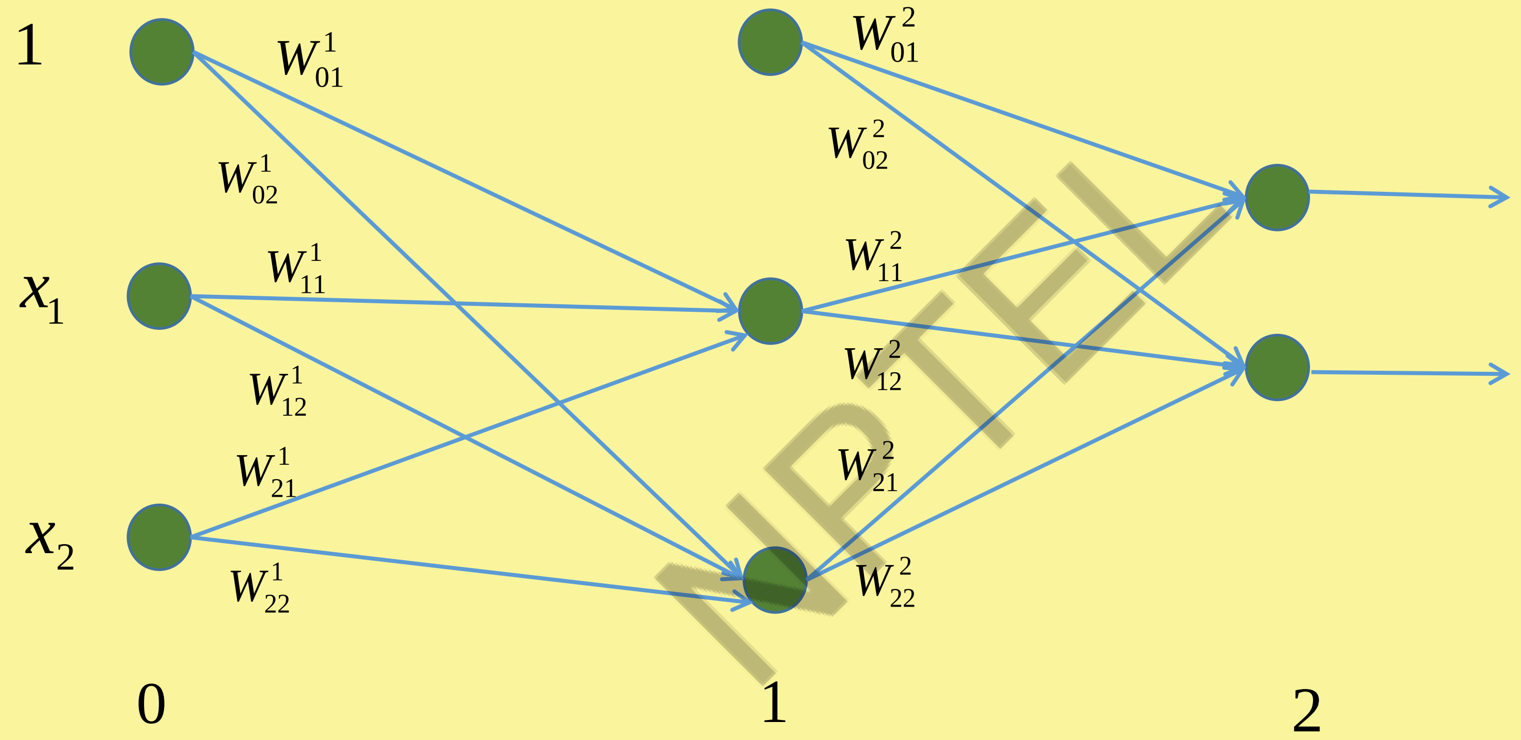
CONCEPTS COVERED

Concepts Covered:

- Back Propagation Learning in MLP
- Different Loss Functions
- Back Propagation Learning - Example
- Back Propagation – Node Level

Back Propagation Learning an Example

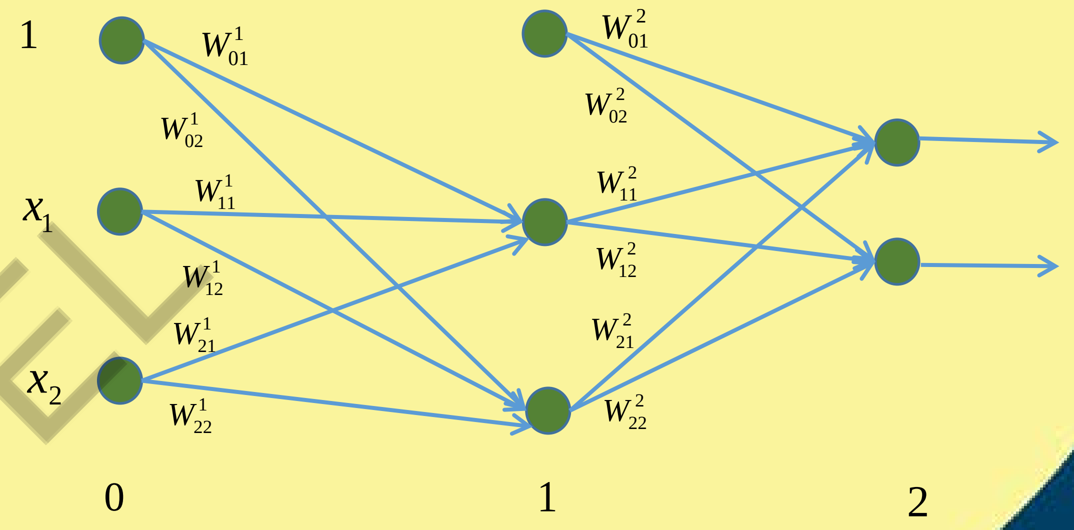
Multilayer Perceptron



Multilayer Perceptron

W^1_{01}	W^1_{11}	W^1_{21}
0.5	1.5	0.8
W^1_{02}	W^1_{12}	W^1_{22}
0.8	0.2	-1.6

W^2_{01}	W^2_{11}	W^2_{21}
0.9	-1.7	1.6
W^2_{02}	W^2_{12}	W^2_{22}
1.2	2.1	-0.2



$$X = \begin{bmatrix} 0.7 \\ 1.2 \end{bmatrix} \text{ from category } 1 \Rightarrow t = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

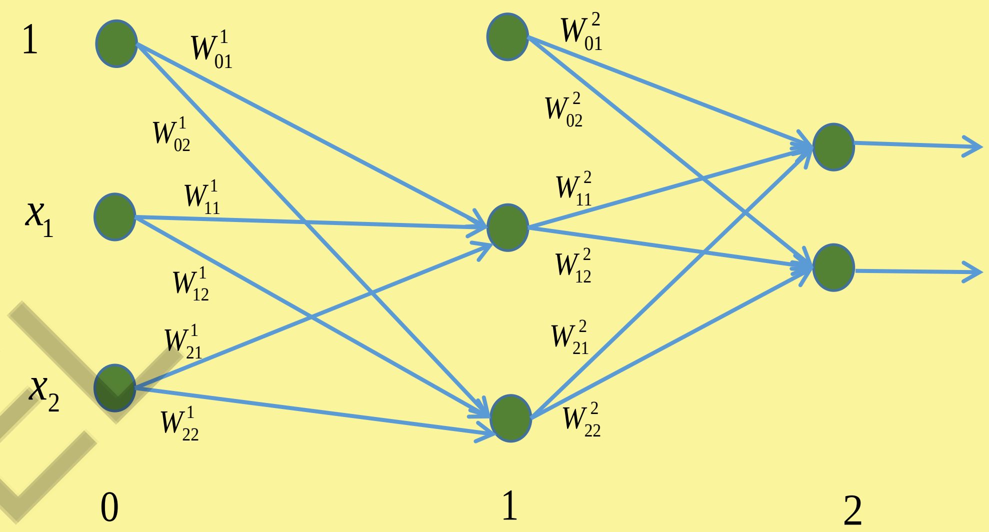
Feed Forward Pass

$$W^1 \qquad x_i^0 \qquad \theta_j^1 = \sum W_{ij}^1 x_i^0 \qquad x_j^1 = \frac{1}{1+e^{-\theta_j^1}}$$

$$\begin{bmatrix} 0.5 & 1.5 & 0.8 \\ 0.8 & 0.2 & -1.6 \end{bmatrix} \begin{bmatrix} 1 \\ 0.7 \\ 1.2 \end{bmatrix} = \begin{bmatrix} 2.51 \\ -9.8 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.92 \\ 0.27 \end{bmatrix}$$

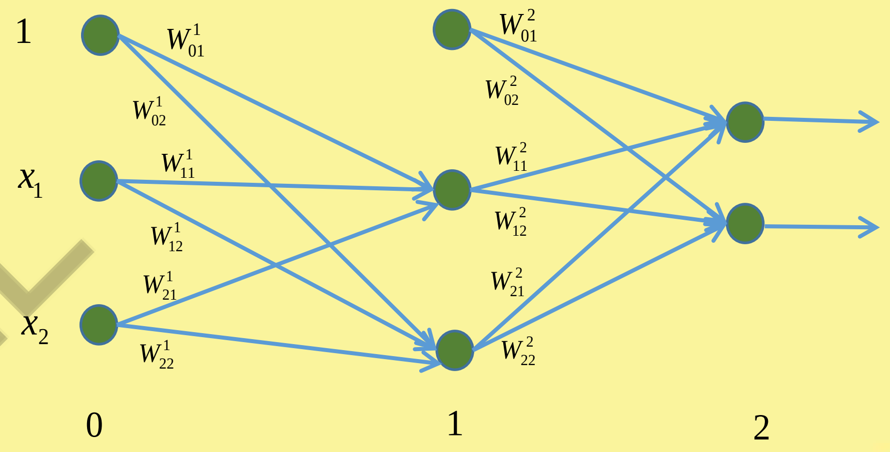
$$W^2 \qquad x_i^1 \qquad \theta_j^2 = \sum_i W_{ij}^2 x_i^1 \qquad x_j^2 = \frac{1}{1+e^{-\theta_j^2}}$$

$$\begin{bmatrix} 0.9 & -1.7 & 1.6 \\ 1.2 & 2.1 & -1.-0.26 \end{bmatrix} \begin{bmatrix} 1 \\ 0.92 \\ 0.27 \end{bmatrix} = \begin{bmatrix} -0.232 \\ 3.057 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.44 \\ 0.95 \end{bmatrix}$$



Back Propagation Learning:- Output Layer

$$E = \frac{1}{2}\sum_{j=1}^{2}\left(x_j^2 - t_j\right)^2 \qquad x_j^2 = \frac{1}{1+e^{-\theta_j^2}} \qquad \theta_j^2 = \sum_{i=0}^{2} W_{ij}^2 x_i^1$$



$$\frac{\partial E}{\partial W_{ij}^2} = \frac{\partial E}{\partial x_j^2}\cdot\frac{\partial x_j^2}{\partial \theta_j^2}\cdot\frac{\partial \theta_j^2}{\partial W_{ij}^2} = (x_j^2 - t_j)x_j^2(1-x_j^2)x_i^1$$

We set $\boxed{\delta_j^2 = x_j^2(1-x_j^2)(x_j^2 - t_j)} \Rightarrow \frac{\partial E}{\partial W_{ij}^2} = \delta_j^2 x_i^1$

$$\boxed{W_{ij}^2 \leftarrow W_{ij}^2 - \eta\frac{\partial E}{\partial W_{ij}^2}}$$

NPTEL ONLINE CERTIFICATION COURSES

Thank you