NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

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

Topic

Lecture 16: Optimization

CONCEPTS COVERED

Concepts Covered:

- Multiclass SVM Loss Function
- Optimization
- Stochastic Gradient descent
- Batch Optimization
- Mini-Batch Optimization

Optimizing Loss Function

$$L = \frac{1}{N}\sum_{i}\sum_{j\neq y_i}\left[\max(0, W_j^t X_i - W_{y_i}^t X_i + \nabla\right] + \lambda \sum_{k}\sum_{l} W_{kl}^2$$

$$\nabla_{W_{y_i}} = -\frac{1}{N}\sum_{i}\sum_{j\neq y_i}[X_i \mid (W_j^t X_i - W_{y_i}^t X_i + \nabla > 0)] + \eta W_{y_i}$$

$$\nabla_{W_j} = \frac{1}{N}\sum_{i}\sum_{j\neq y_i}[X_i \mid (W_j^t X_i - W_{y_i}^t X_i + \nabla > 0)] + \xi W_j$$

Optimizing Loss Function

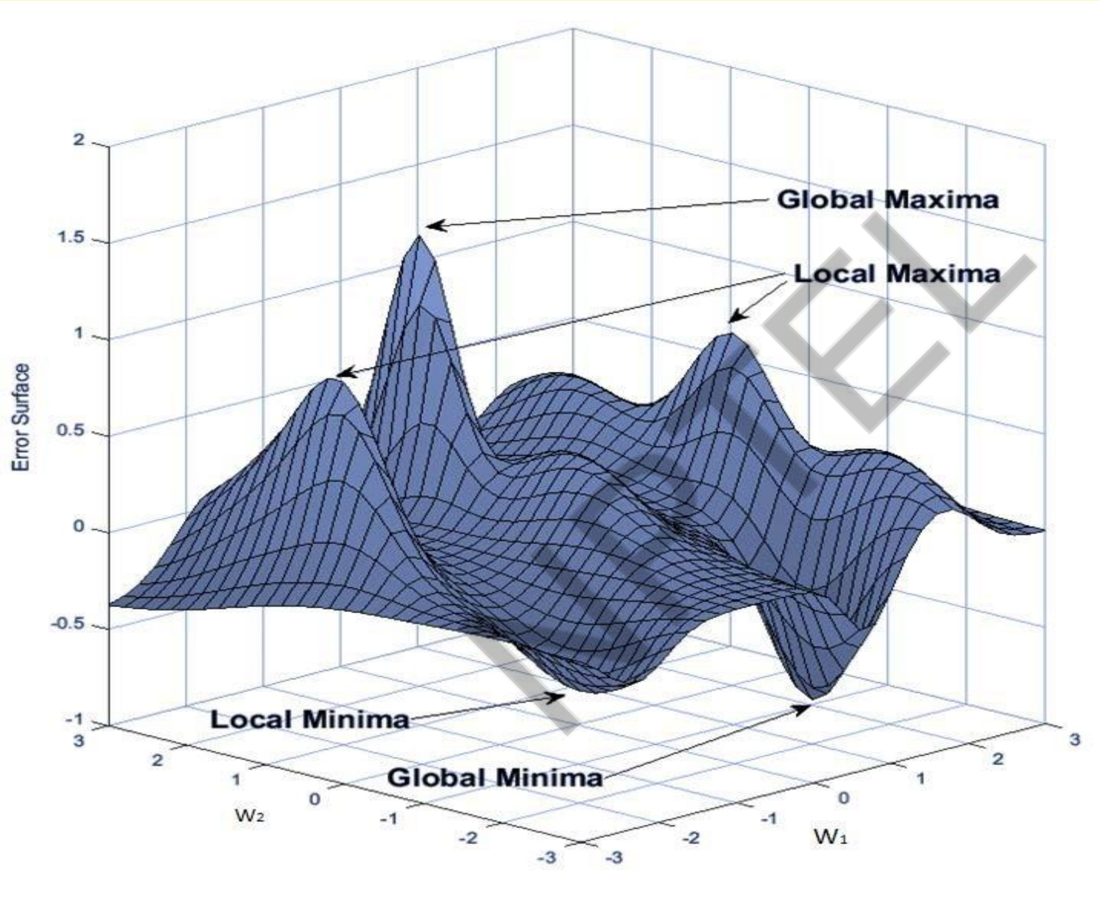
$$\nabla_{W_{y_i}} = -\frac{1}{N}\sum_{i}\sum_{j\neq y_i}[X_i \mid (W_j^t X_i - W_{y_i}^t X_i + \nabla > 0)] + \eta W_{y_i} \qquad \nabla_{W_j} = \frac{1}{N}\sum_{i}\sum_{j\neq y_i}[X_i \mid (W_j^t X_i - W_{y_i}^t X_i + \nabla > 0)] + \xi W_j$$

Gradient descent

$$W_{y_i}(k+1) \leftarrow (1-\eta)W_{y_i}(k) + \frac{1}{N}\sum_{i}\sum_{j\neq y_i}[X_i \mid (W_j^t X_i - W_{y_i}^t X_i + \nabla > 0)]$$

$$W_j(k+1) = (1-\xi)W_j(k) - \frac{1}{N}\sum_{i}\sum_{j\neq y_i}[X_i \mid (W_j^t X_i - W_{y_i}^t X_i + \nabla > 0)]$$

Local and Global Minima



Stochastic/ Batch/ Mini batch Optimization

Stochastic Gradient Descent

Upsides

- The frequent updates immediately give an insight into the performance of the model and the rate of improvement.
- This variant of gradient descent may be the simplest to understand and implement.
- The increased model update frequency can result in faster learning on some problems.
- The noisy update process can allow the model to avoid local minima (e.g. premature convergence).

Stochastic Gradient Descent

Downsides

- Updating the model so frequently is more computationally expensive than other configurations of gradient descent, taking significantly longer to train models on large datasets.

- The frequent updates can result in a noisy gradient signal, which may cause the model parameters and in turn the model error to jump around (have a higher variance over training epochs).

- The noisy learning process down the error gradient can also make it hard for the algorithm to settle on an error minimum for the model.

Batch Gradient Descent

Upsides

- Fewer updates to the model means this variant of gradient descent is more computationally efficient than stochastic gradient descent.
- The decreased update frequency results in a more stable error gradient and may result in a more stable convergence on some problems.
- The separation of the calculation of prediction errors and the model update lends the algorithm to parallel processing based implementations.

Batch Gradient Descent

Downsides

- The more stable error gradient may result in premature convergence of the model to a less optimal set of parameters.
- The updates at the end of the training epoch require the additional complexity of accumulating prediction errors across all training examples.
- It requires the entire training dataset in memory and available to the algorithm.
- Model updates, and in turn training speed, may become very slow for large datasets.

Mini-Batch Gradient Descent

Upsides

- The model update frequency is higher than batch gradient descent which allows for a more robust convergence, avoiding local minima.

- The batched updates provide a computationally more efficient process than stochastic gradient descent.

- The batching allows both the efficiency of not having all training data in memory and algorithm implementations.

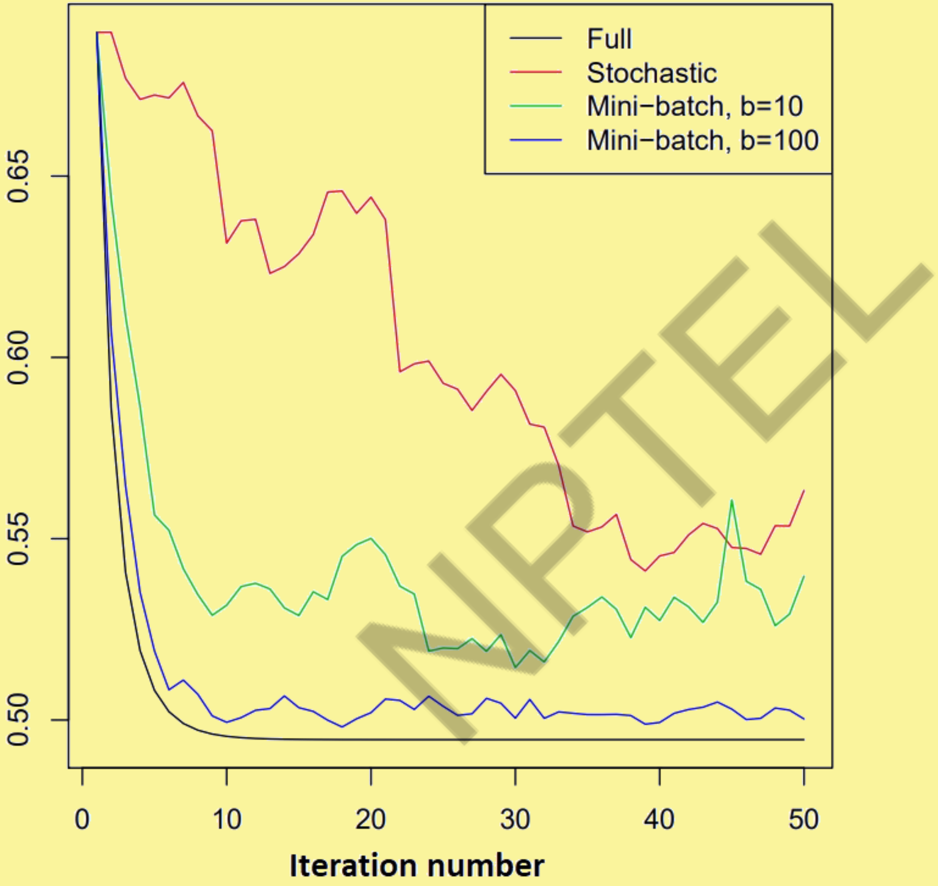
Mini-Batch Gradient Descent

Downsides

- ❑ Mini-batch requires the configuration of an additional “mini-batch size” hyper parameter for the learning algorithm.

- ❑ Error information must be accumulated across mini-batches of training examples like batch gradient descent.

Error minimization with iterations




NPTEL ONLINE CERTIFICATION COURSES
Thank you
NPTEL

NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

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

Topic

Lecture 17: Optimization in ML

CONCEPTS COVERED

Concepts Covered:

- Optimization
  - Stochastic Gradient Descent
  - Batch Optimization
  - Mini-batch optimization
- Optimization in ML
- Linear and Logistic Regression
- Softmax classifier
- Nonlinearity

Optimization in Machine Learning

Optimization in Machine Learning

- Goal of optimization is to reduce a cost function *J(W)* to optimize some performance measure *P*.
- In pure optimization minimizing *J* is the goal in and of itself.
- In Machine Learning *J(W)* is minimized w.r.t parameter *W* on training data (training error), and we the error to be low on unforeseen (test) data.
- Test error (generalization error) should be low.

Optimization in Machine Learning

Assumptions

- Test and Training data are generated by a probability distribution: Data generating process.
- Data samples in each data set are independent.
- Training set and Test set are identically distributed.

Performance of ML is its ability to

- Make the training error small.
- Reduce the gap between training and test error.

Underfitting and Overfitting

- **Underfitting**: Model is not able to obtain sufficiently low training error.
- **Overfitting**: The gap between training and test error is too large.

We can control Overfitting/ Underfitting by altering its Capacity

Set of functions the learning algorithm can select as being the solution

Linear and Logistic Regression

Linear & Logistic Regression- Binary Classification

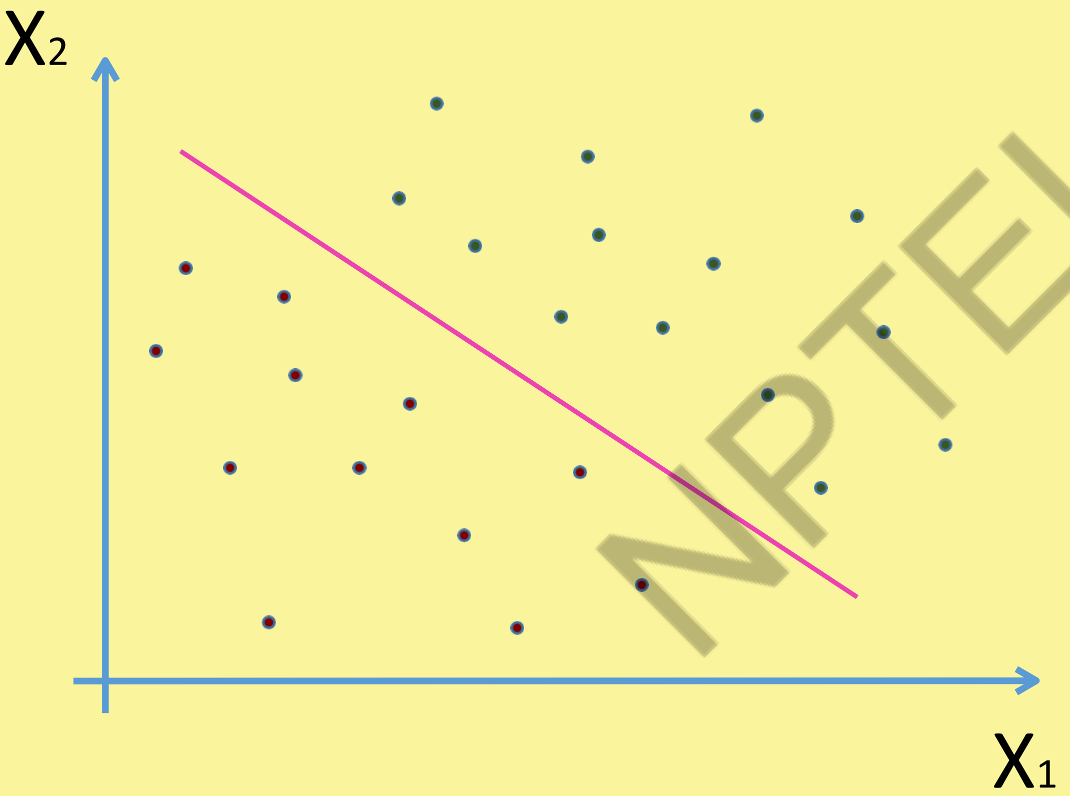
Linear Regression

$$f : X \in R^d \rightarrow y \in R \qquad \hat{y} = W^t X$$

Logistic Regression

$$p(y | X; W) = \sigma(W^t X)$$

Linear Regression

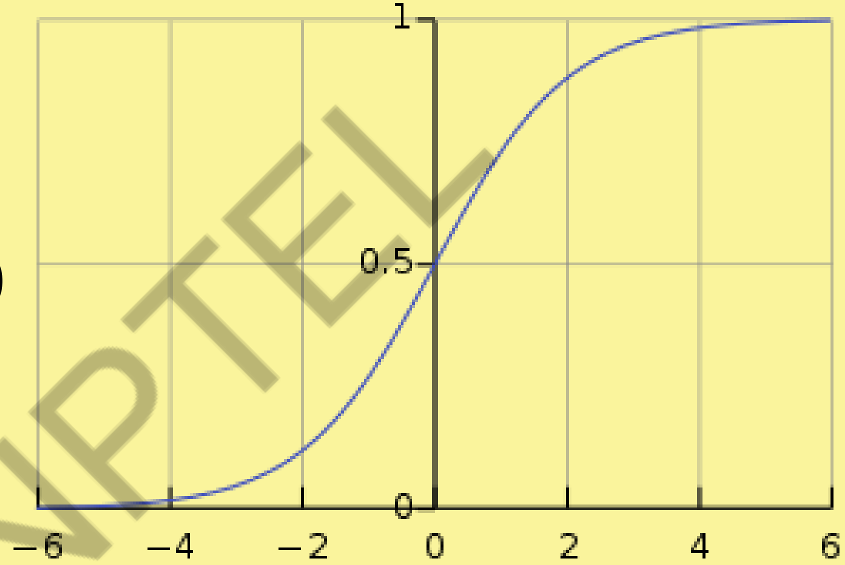


Logistic Regression

$$\sigma(W^t X) = \frac{1}{1+e^{-W^t X}} \Rightarrow$$

$\uparrow$

$\sigma(W^t X)$



$W^t X \rightarrow$

Softmax Classifier

- Generalization of Binary Logistic Classifier to Multiple Classes

$$s_{y_i} = f(X_i, W)_{y_i} = (WX_i)_{y_i} = W_{y_i}^t X_i$$

- Softmax Classifier

$$p(y_i \mid X_i; W) = \frac{e^{s_{y_i}}}{\sum_{\forall j} e^{s_j}}$$

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 18: Nonlinearity

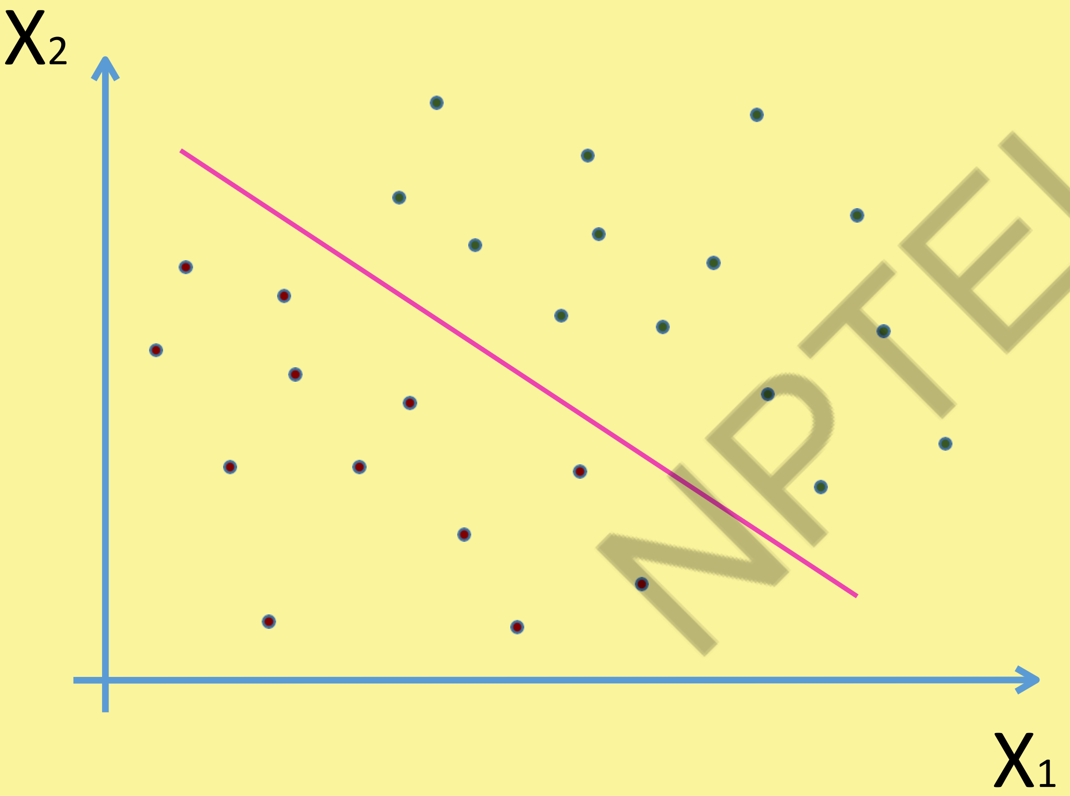
CONCEPTS COVERED

Concepts Covered:

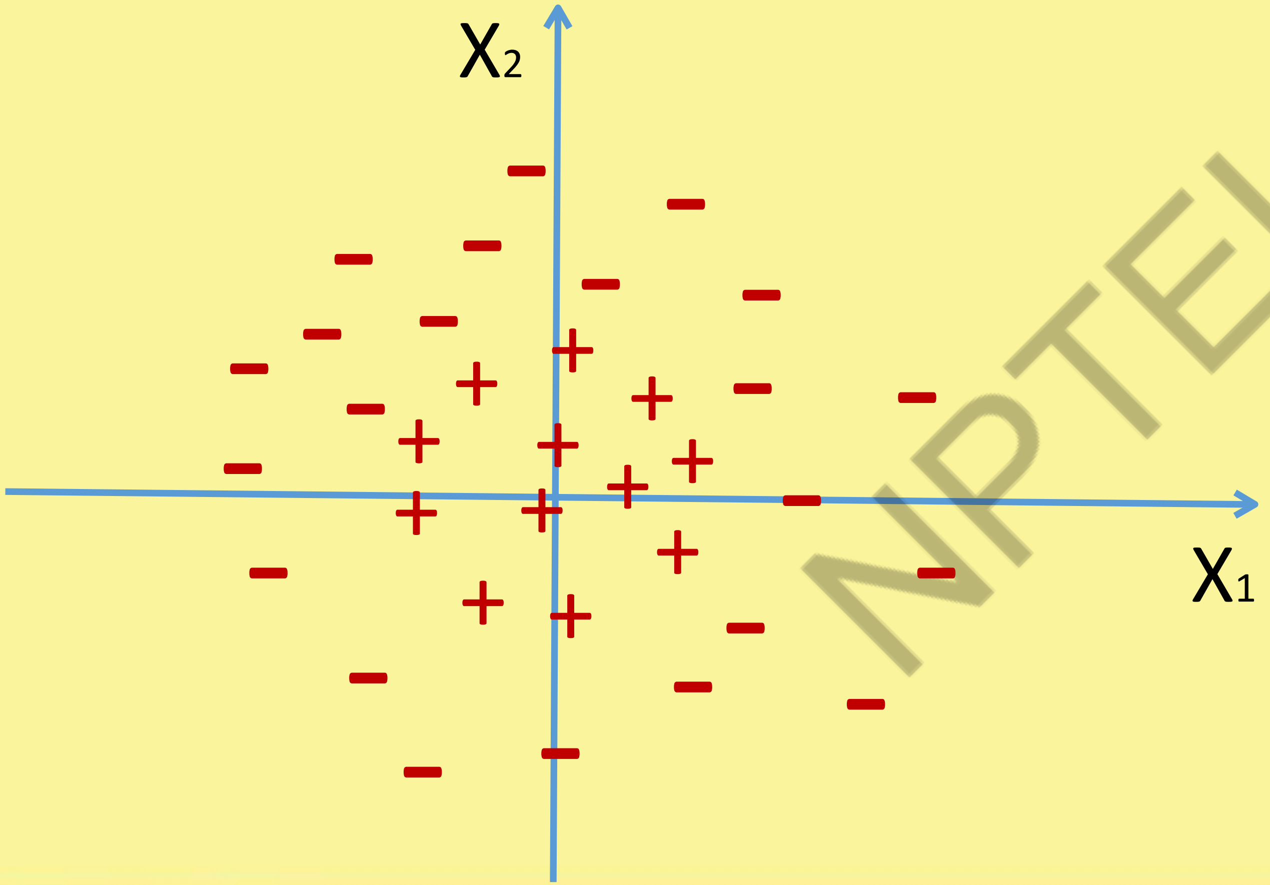
- ❑ Optimization in ML
- ❑ Linear and Logistic Regression
- ❑ Softmax classifier
- ❑ Nonlinearity
- ❑ Neural Network

Nonlinearity

Linear Seperability



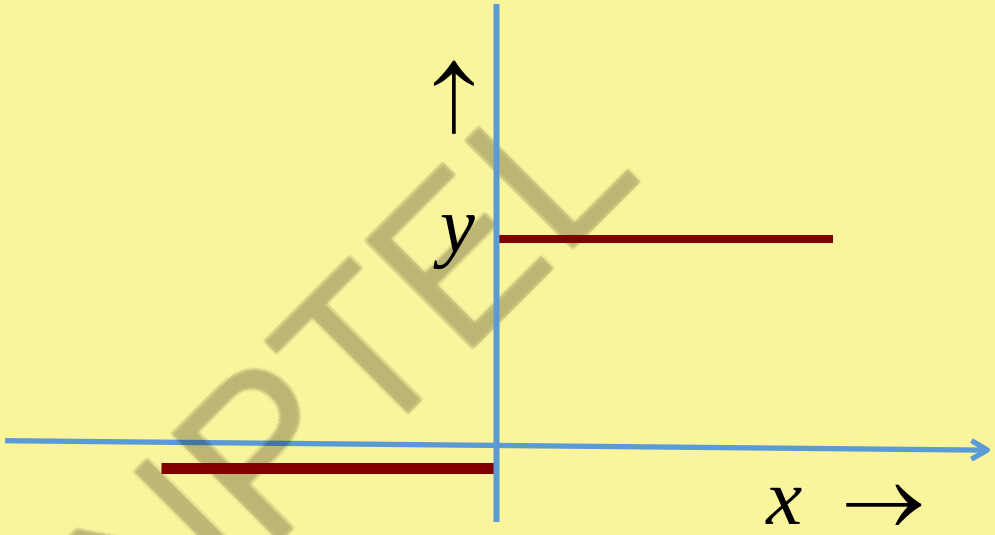
Nonlinearity



Nonlinearity

Threshold

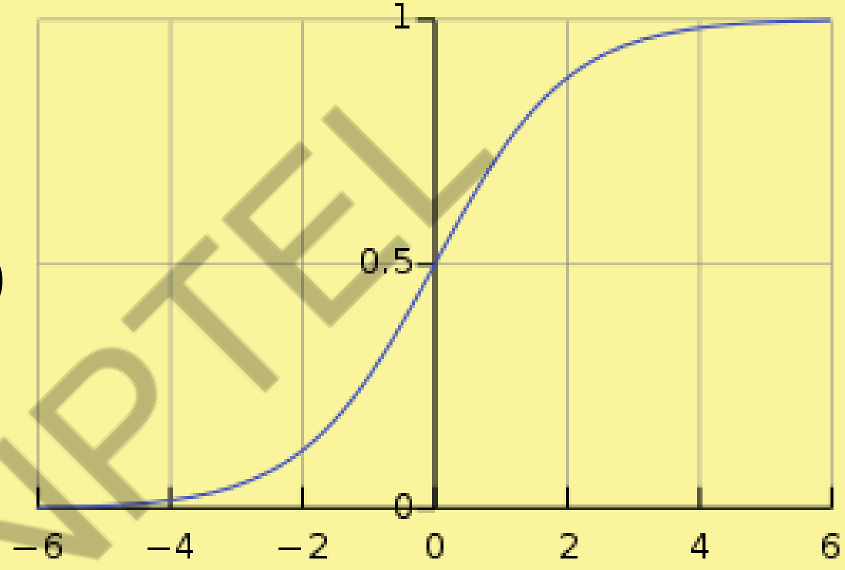
$$y = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$



Logistic Regression

$$\sigma(W^tX) = \frac{1}{1+e^{-W^tX}} \Rightarrow$$

$\uparrow$
$\sigma(W^tX)$

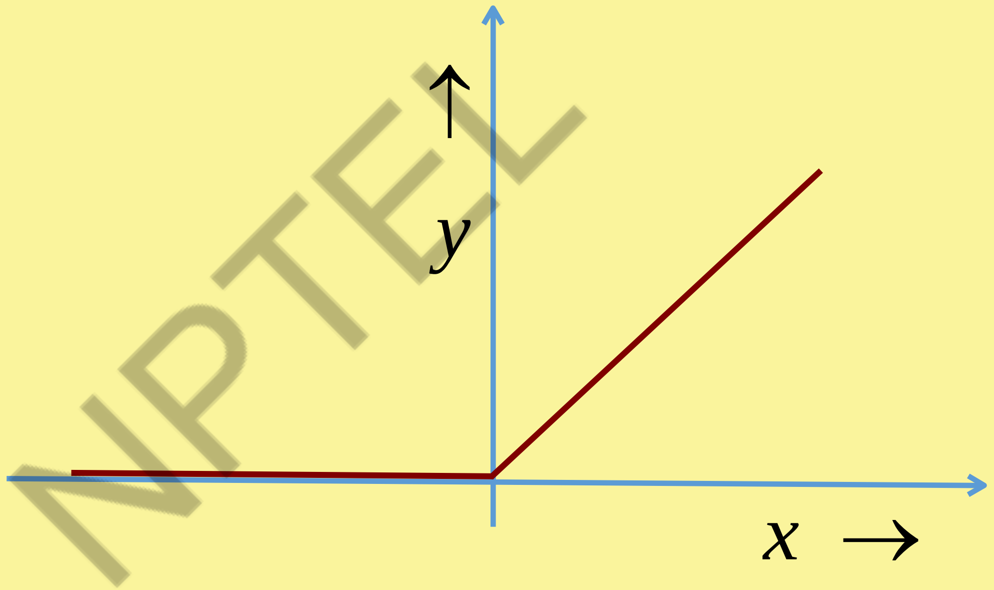


$W^tX \rightarrow$

Nonlinearity

ReLU : Rectified Linear Unit

$$y = \max(0, x) \;\Rightarrow$$



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 19: Neural Network

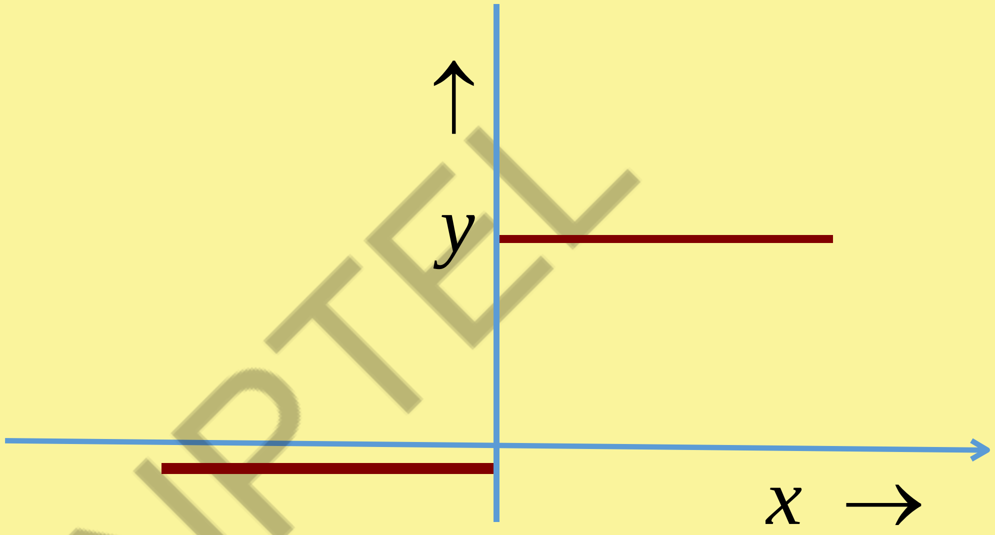
CONCEPTS COVERED

Concepts Covered:

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

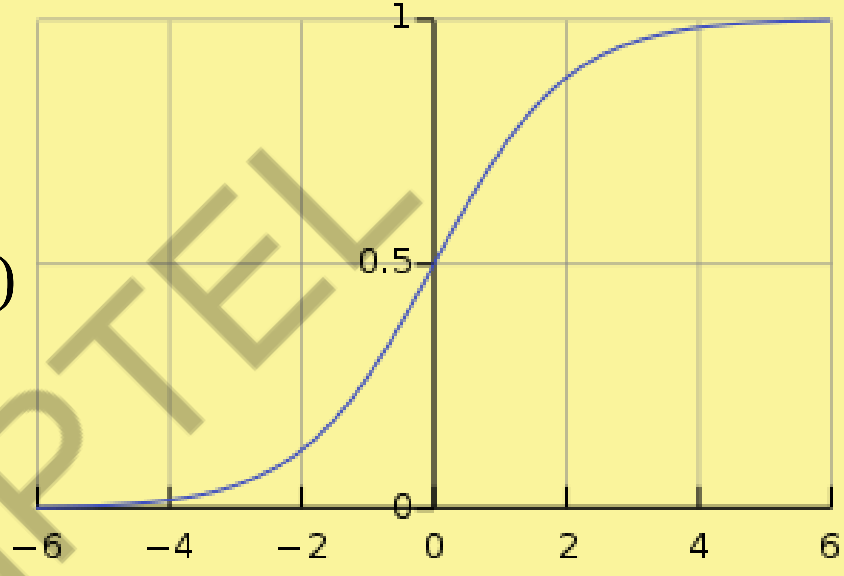
Threshold

$$y = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$



Logistic Regression

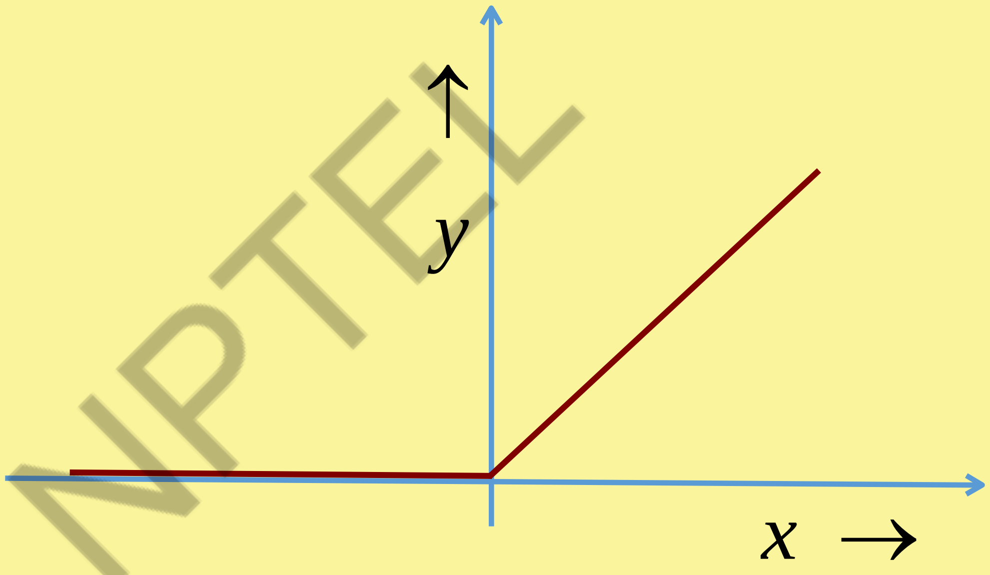
$$\sigma(W^t X) = \frac{1}{1+e^{-W^t X}} \Rightarrow \sigma(W^t X)$$



Nonlinearity

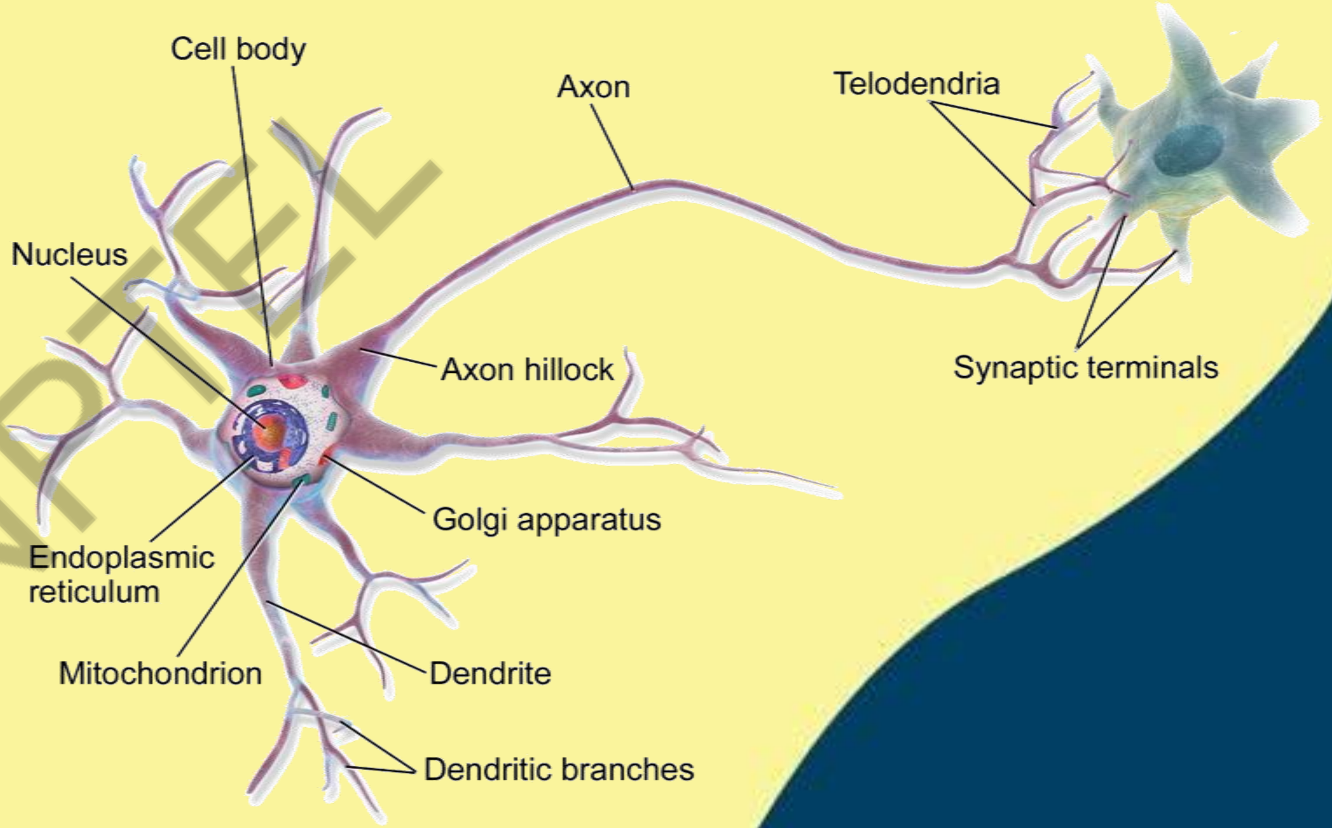
ReLU : Rectified Linear Unit

$$y = \max(0, x) \Rightarrow$$

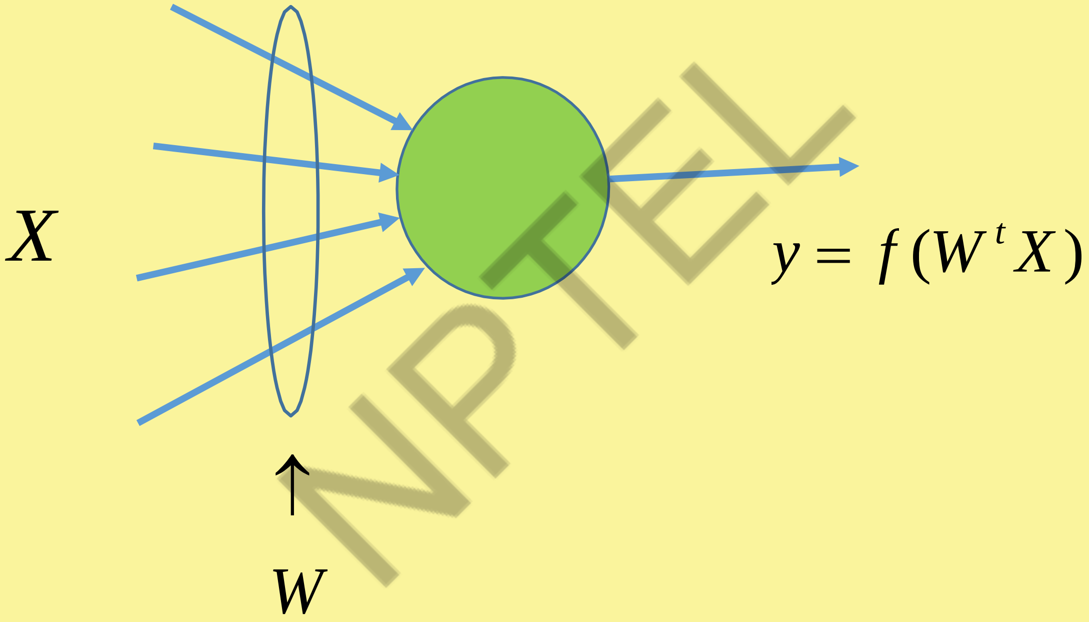


Neuron

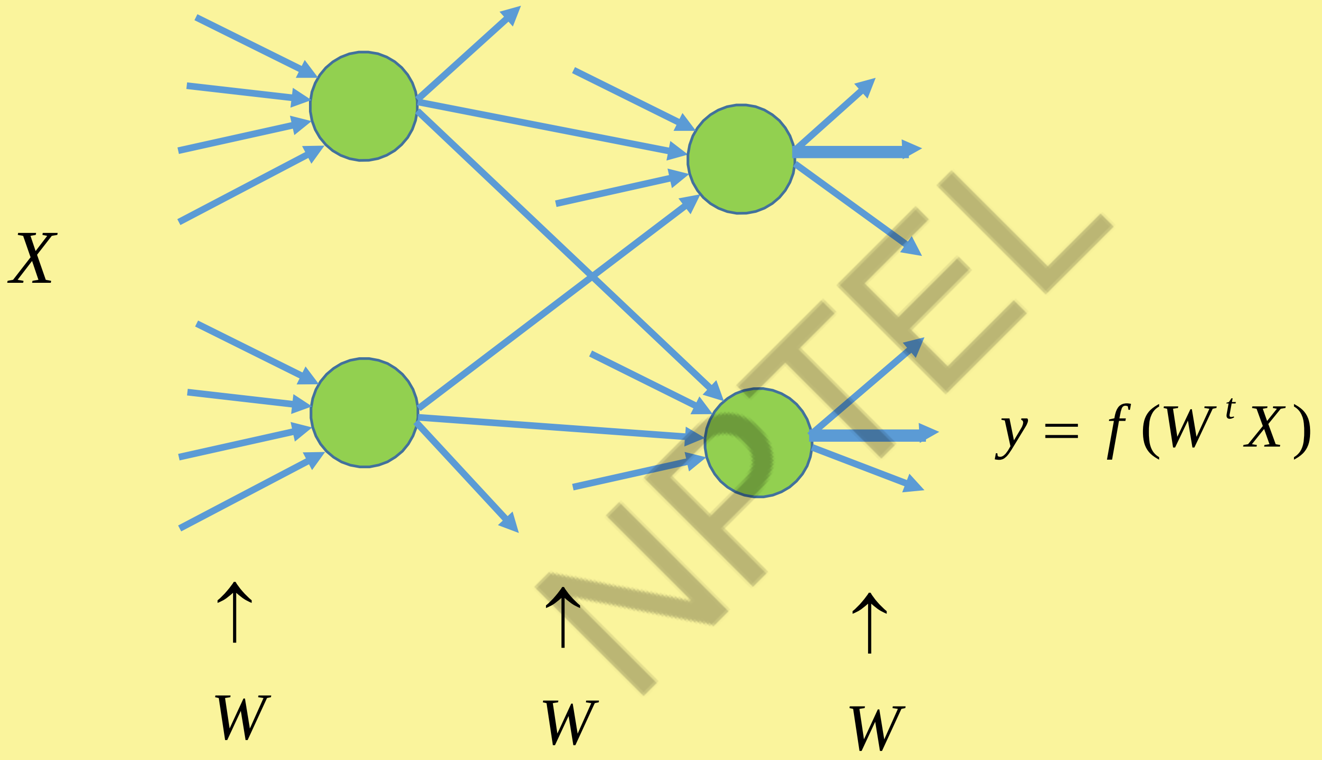
- **Dendrite:** receives signals from other neurons
- **Synapse :** point of connection to other neurons
- **Soma :** processes the information
- **Axon :** transmits the output of this neuron.



Neuron

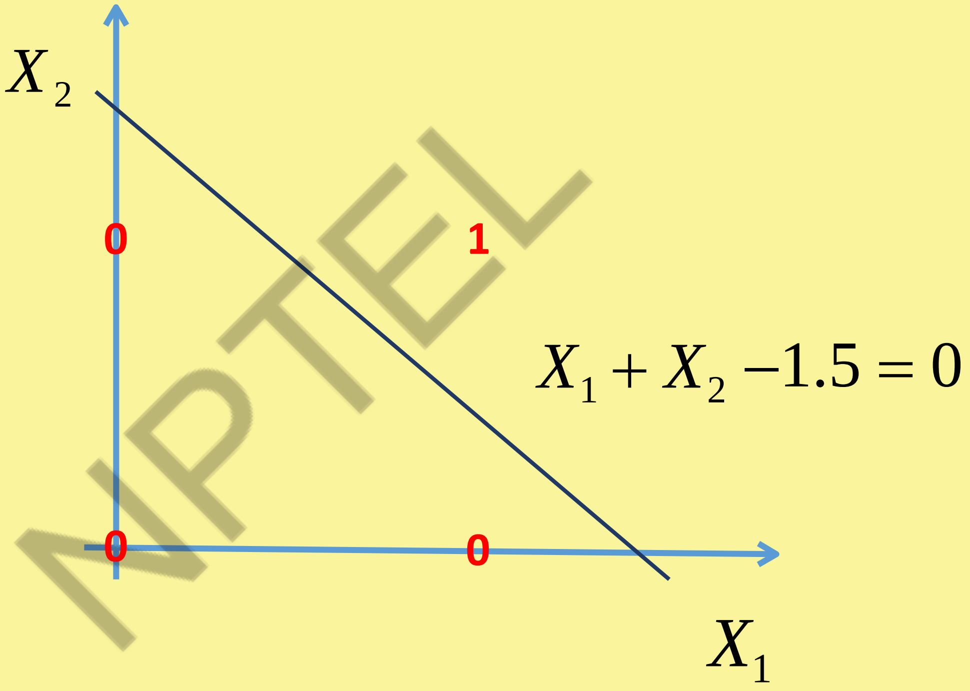


Neural Network



AND Function

X_1	X_2	y
0	0	0
0	1	0
1	0	0
1	1	1

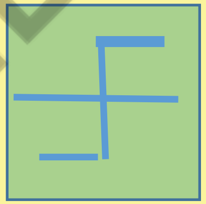


AND Function

$$W = \begin{bmatrix} -1.5 \\ 1 \\ 1 \end{bmatrix} \qquad X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

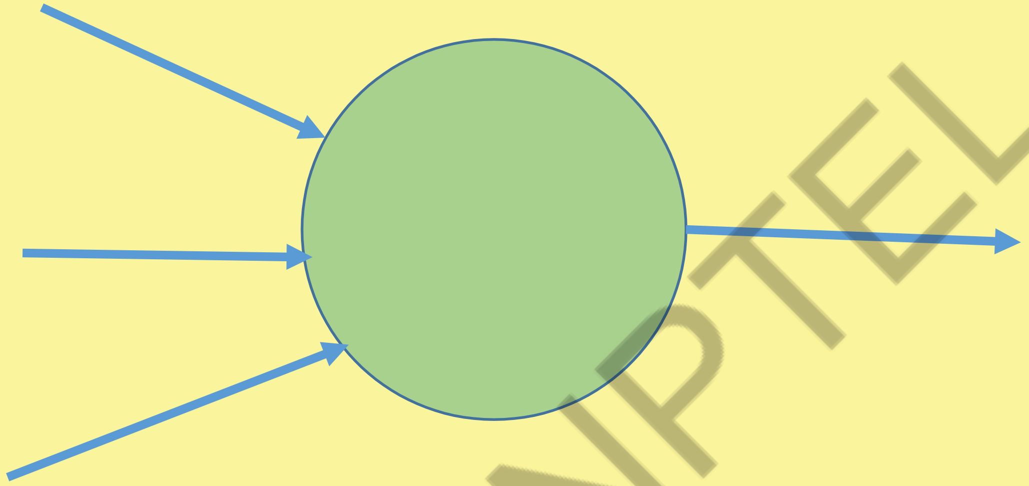
AND Function

$$X^{t}W = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1.5 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1.5 \\ -0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \Rightarrow$$



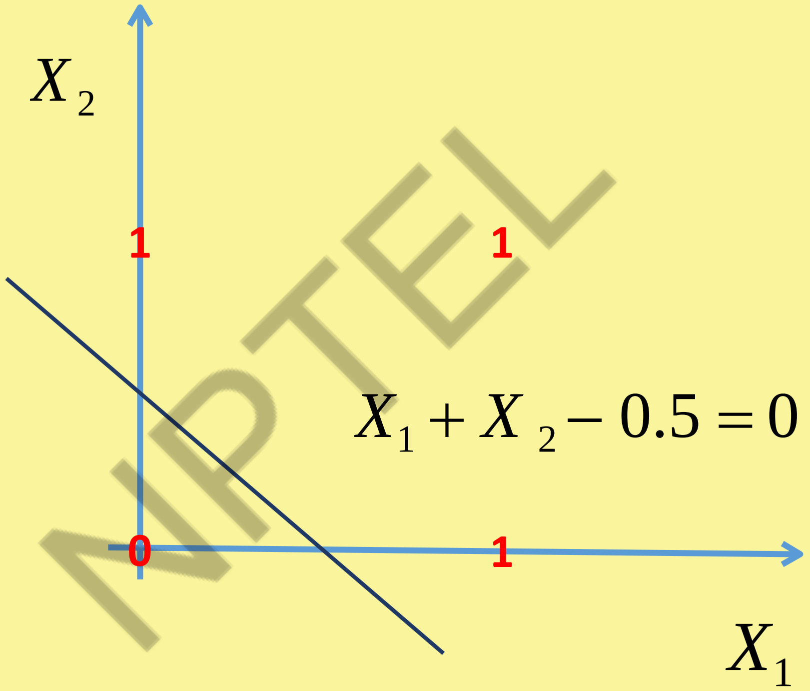
$$\Rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

AND Function



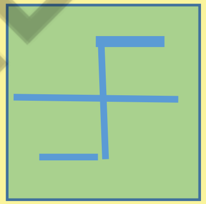
OR Function

X_1	X_2	y
0	0	0
0	1	1
1	0	1
1	1	1



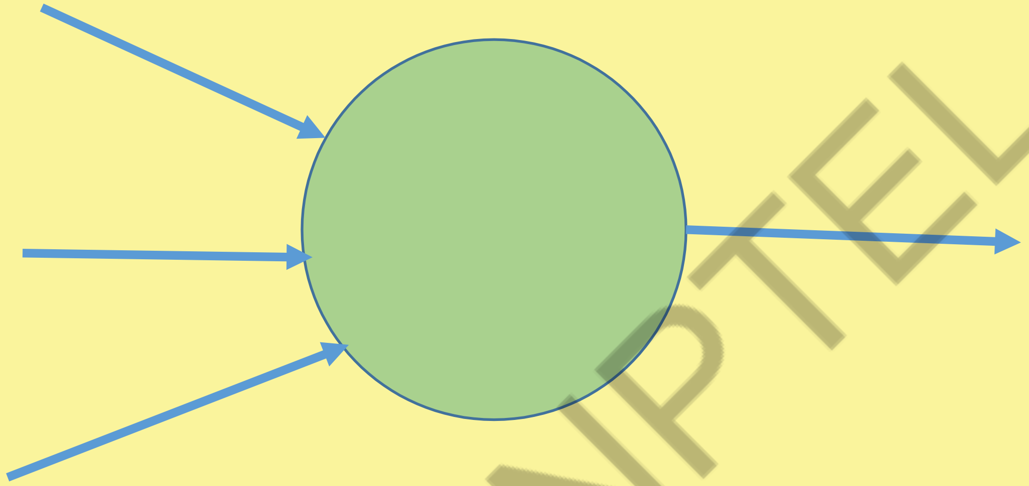
OR Function

$$X^{t}W = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.5 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.5 \\ 0.5 \\ 1.5 \end{bmatrix} \Rightarrow$$



$$\Rightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

OR Function



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

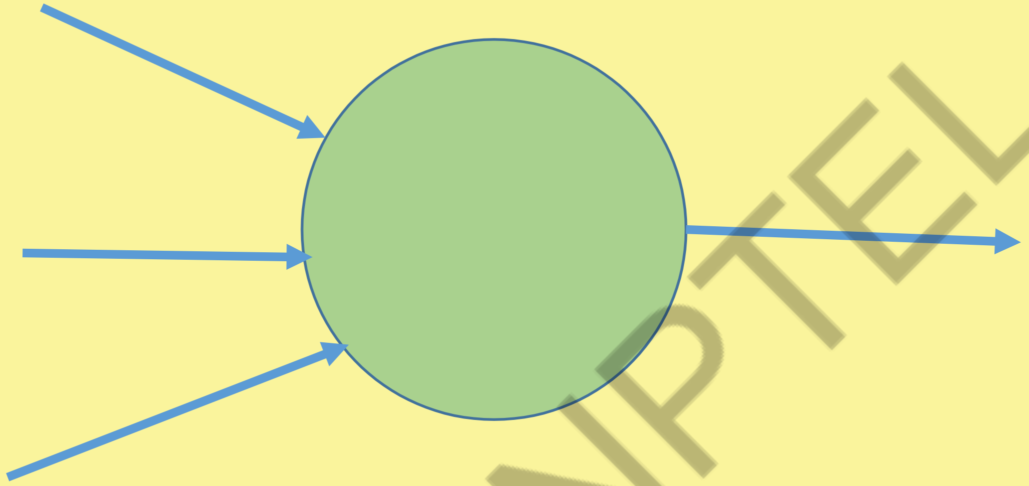
Lecture 20: Neural Network - II

CONCEPTS COVERED

Concepts Covered:

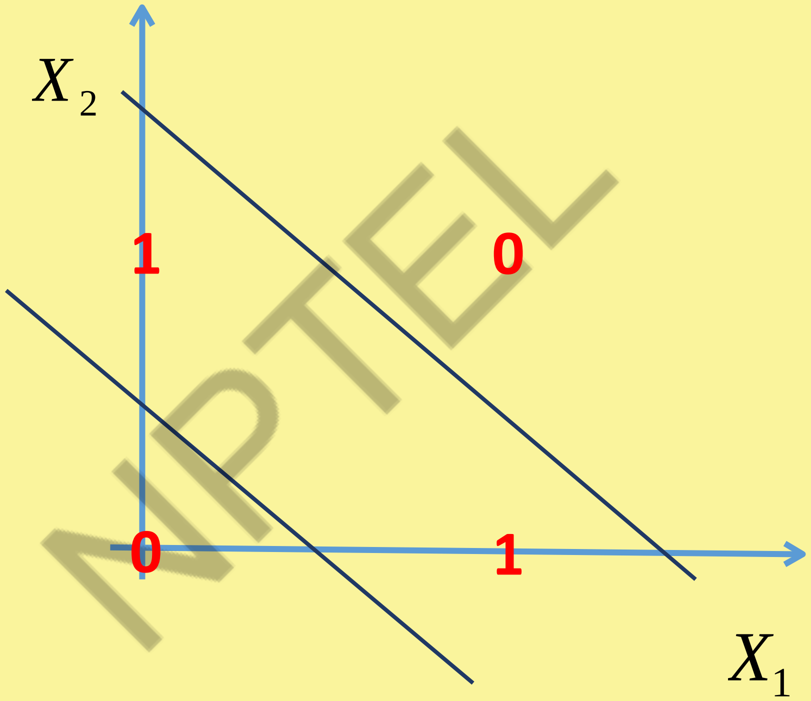
- Neural Network
  - AND Logic
  - OR Logic
  - XOR Logic
- Feed Forward NN
- Back Propagation Learning

AND/ OR Function



XOR Function

X_1	X_2	y
0	0	0
0	1	1
1	0	1
1	1	0



XOR Function

$$X_1 \oplus X_2 = (X_1 + X_2).(\overline{X}_1 + \overline{X}_2)$$

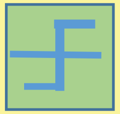
X_1	X_2	$h_1 = X_1 + X_2$	$h_2 = \overline{X}_1 + \overline{X}_2$	$h_1.h_2 = X_1 \oplus X_2$
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

XOR Function

$$\underbrace{\begin{bmatrix} -0.5 & 1 & 1 \\ 1.5 & -1 & -1 \end{bmatrix}}_{W_1^t} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{X} = \begin{bmatrix} -0.5 & 0.5 & 0.5 & 1.5 \\ 1.5 & 0.5 & 0.5 & -0.5 \end{bmatrix} \Rightarrow$$

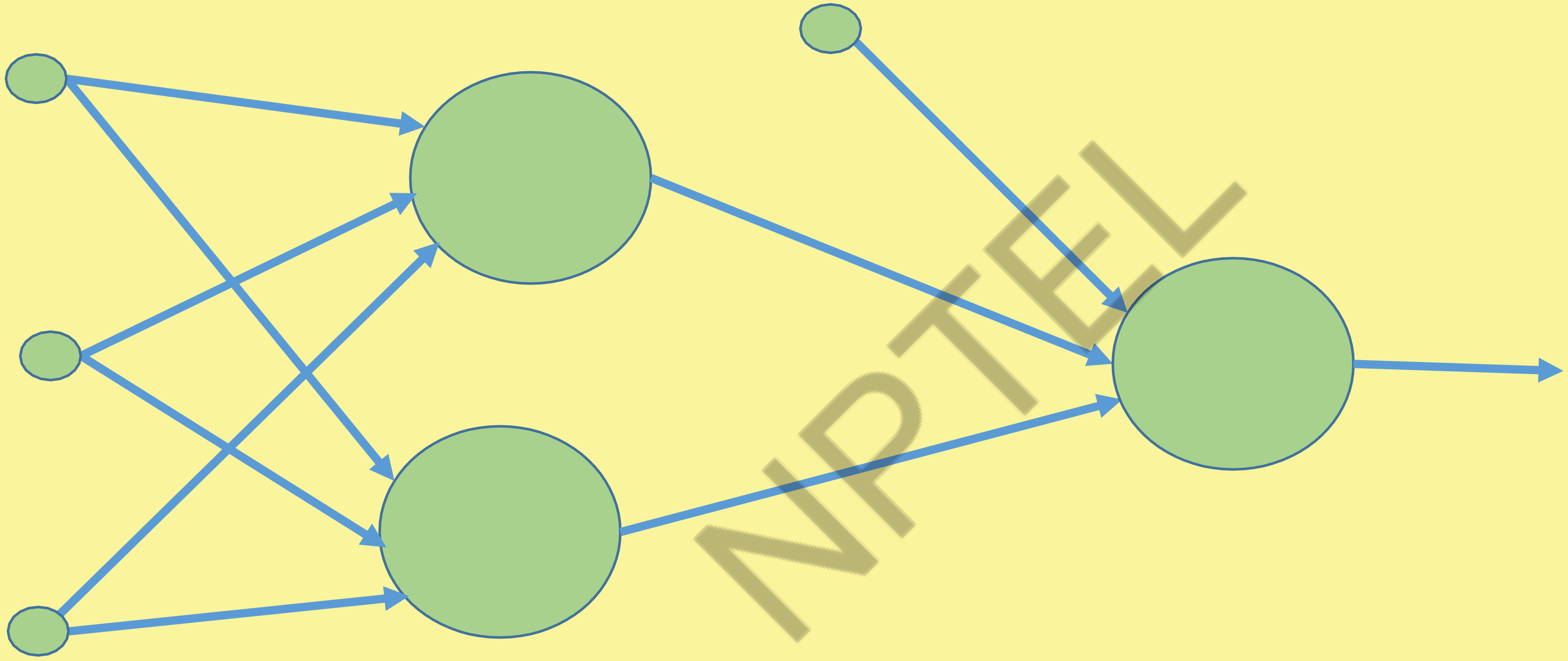
$$\Rightarrow \underbrace{\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}}_{h}$$

$$h^t W_2 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1.5 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.5 \\ 0.5 \\ -0.5 \end{bmatrix} \Rightarrow$$

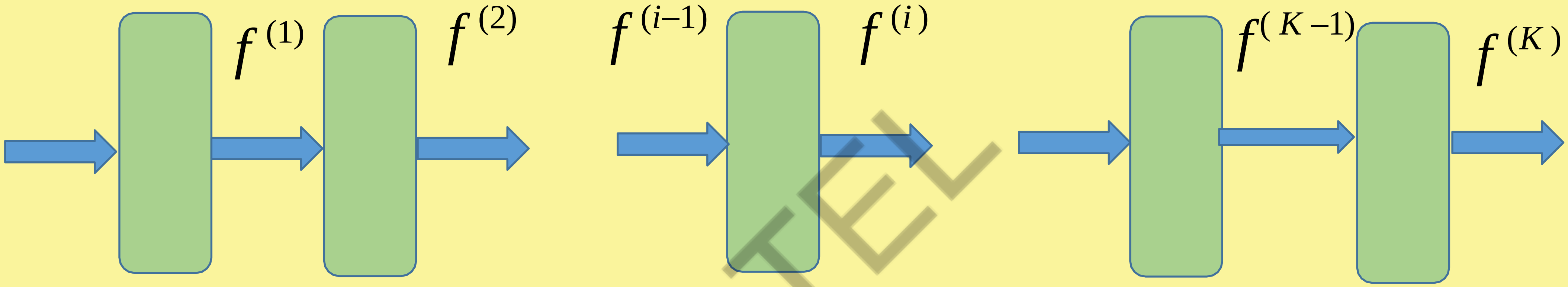


$$\Rightarrow \underbrace{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}}_{X_1 \oplus X_2}$$

XOR Function



Neural Network Function



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

NPTEL ONLINE CERTIFICATION COURSES

Thank you