BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Deep Neural Networks

The author of this deck, Prof. Seetha Parameswaran, is gratefully acknowledging the authors who made their course materials freely available online.

Module 2

Human Brain

How do Humans Learn?

- It begins with human brain.
- Humans learn, solve problems, recognize patterns, create, think deeply about something, meditate and many many more.....
- Humans learn through **association**. [Refer to Associationism for more details.]

Observation: The Brain

- The brain is a mass of interconnected neurons.
- Number of neurons is approximately 10^(10).
- Connections per neuron is approximately 10^(4 to 5).
- Neuron switching time is approximately 0.001 second.
- Scene recognition time is 1 second.
- 100 inference steps doesn't seem like enough. Lot of parallel computation.

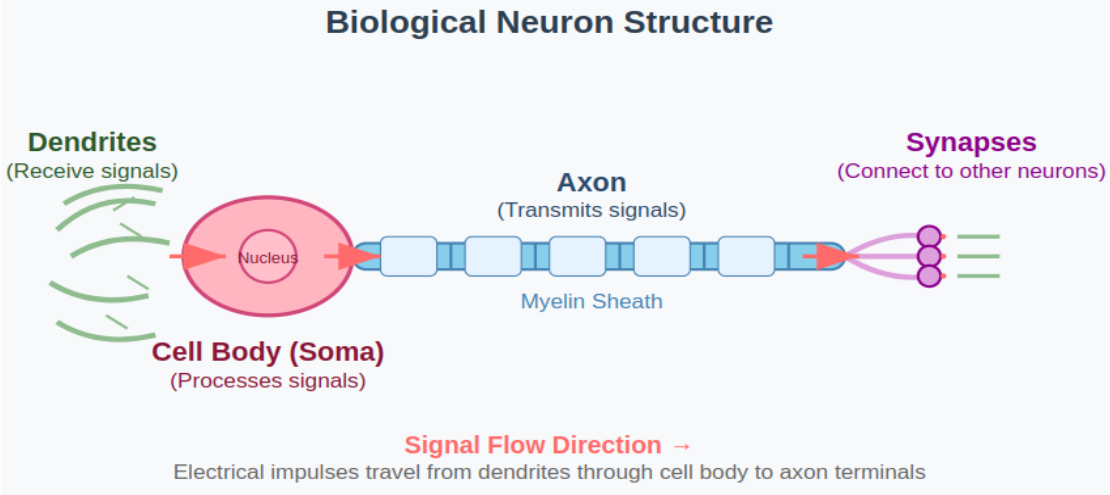
Brain: Interconnected Neurons

- Many neurons connect **in** to each neuron.
- Each neuron connects **out** to many neurons.

Biological Neuron

Key Components:

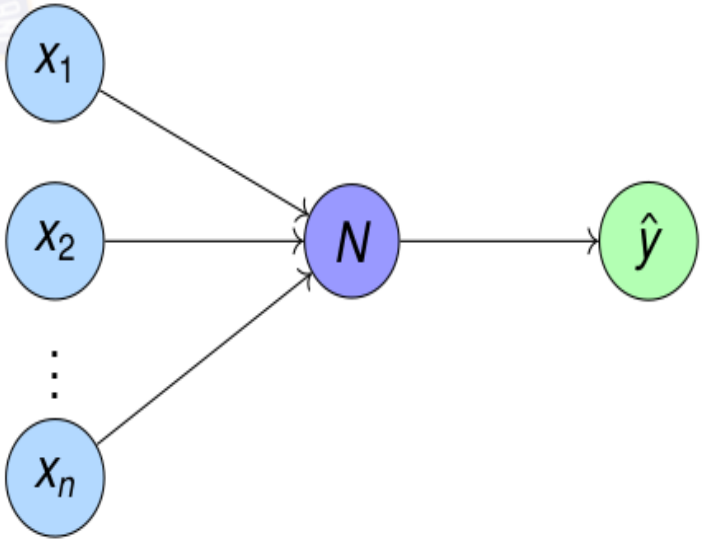
- **Dendrites:** Receive signals from other neurons
- **Cell Body (Soma):** Processes incoming signals
- **Axon:** Transmits output signal
- **Synapses:** Connection points between neurons



Artificial Neuron

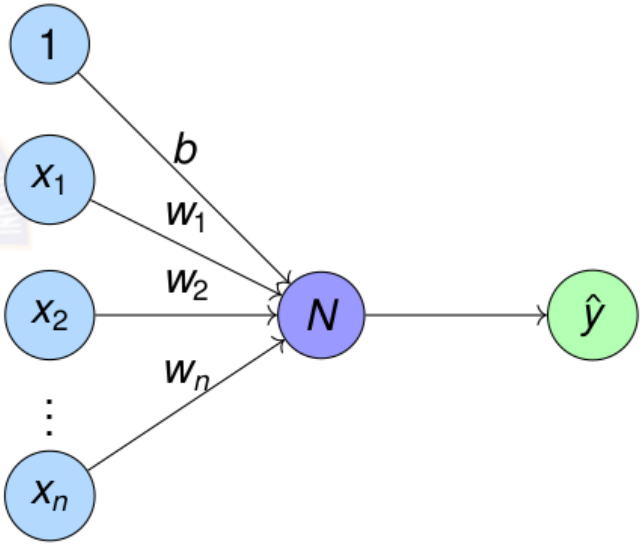
What are Artificial Neurons?

- Neuron is a processing element inspired by how the brain works.
- Similar to biological neuron, each artificial neuron will be do some computation. Each neuron is interconnected to other neurons.
- Similar to brain, the interconnections between neurons store the knowledge it learns. The knowledge is stored as parameters.



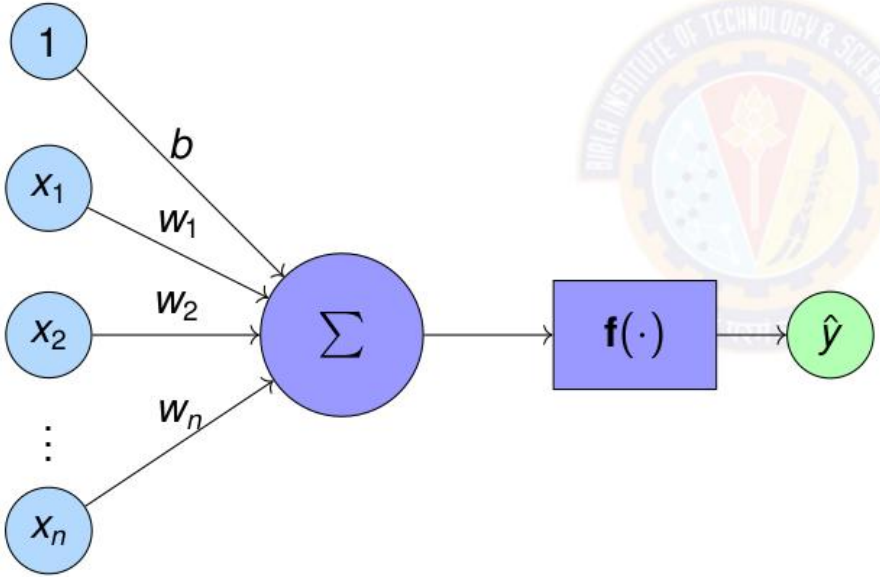
Artificial Neuron N

- Processing neuron N performs a very basic computation.
- For each feature x_i , weight w_i shows the importance to the feature
- N generates one numerical output $\hat{y}$



Artificial Neuron – Mathematical Computation

- Sum of weights w_i times corresponding features x_i
- Apply activation function f () on the sum.



$$z = \sum_{i=1}^{n} w_i x_i + b \qquad (1)$$

$$\hat{y} = \mathbf{f}\left(\sum_{i=1}^{n} w_i x_i + b \right) \qquad (2)$$

Comparison: Biological vs Artificial Neurons

Biological Neuron

- Complex biochemical processes
- Analog signal processing
- Adaptive synaptic strengths
- Parallel processing
- Learning through experience
- Fault tolerant

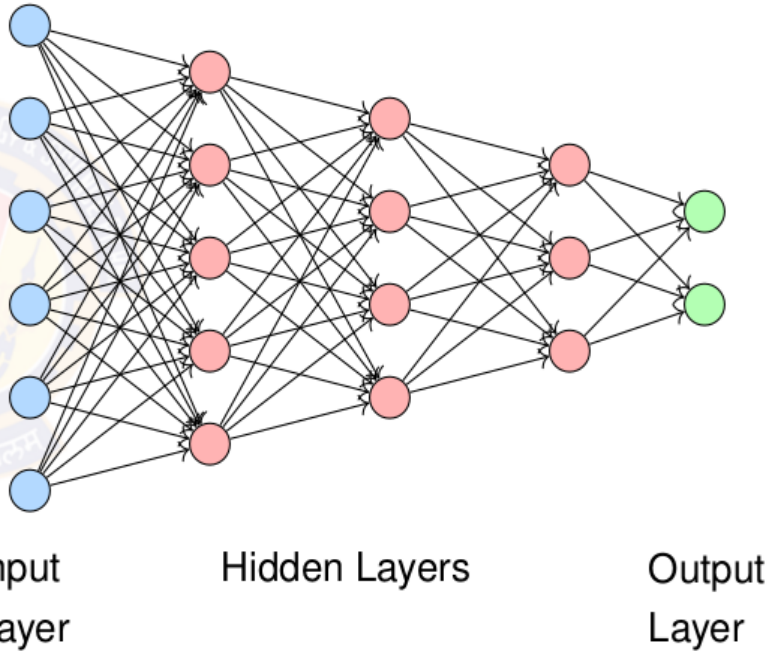
Artificial Neuron

- Simple mathematical model
- Digital computation
- Adjustable weights
- Parallel computation possible
- Learning through algorithms
- Deterministic behavior

Artificial Neural Network (ANN)

What are Artificial Neural Networks?

- Neural Network is a computational model inspired by human brain
- Interconnected neurons process information
- Interconnection can be forward, recursive or self loop
- Learn patterns from data through training
- Capable of approximating complex functions



Properties of Artificial Neural Network

- Many neuron-like threshold switching units.
- Many weighted interconnections among units.
- Highly parallel, distributed process.
- Emphasis on tuning parameters or weights automatically.

When to consider Artificial Neural Networks?

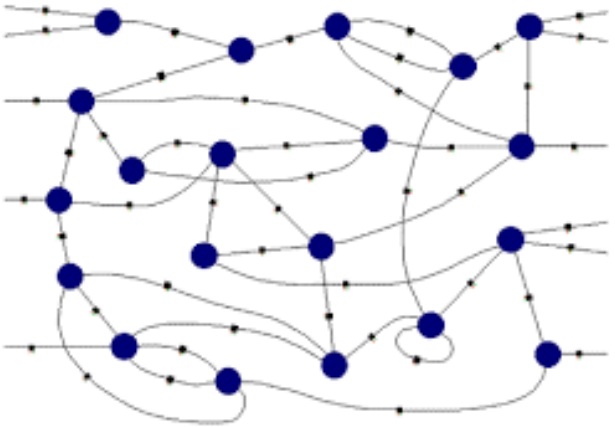
- Input is high-dimensional discrete or real-valued (e.g. raw sensor input).
- Possibly noisy data. Data has lots of errors.
- Output is discrete or real valued or a vector of values.
- Form of target function is unknown.
- Human readability, in other words, explainability , of result is unimportant.

Examples:

  - Speech phoneme recognition
  - Image classification
  - Financial prediction

Connectionist Machines

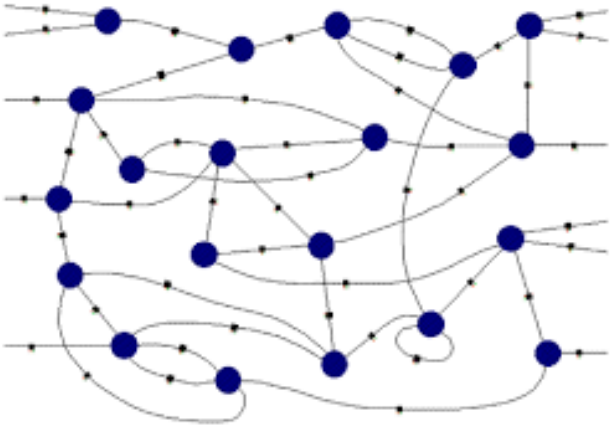
- Network of processing elements, called artificial neurons
- The neurons are interconnected to form a network.
- Interconnection can be forward, recursive or self loop
- All world knowledge is stored in the connections between these elements.
- Neural networks are connectionist machines.



Connectionism Principles

Core Ideas:

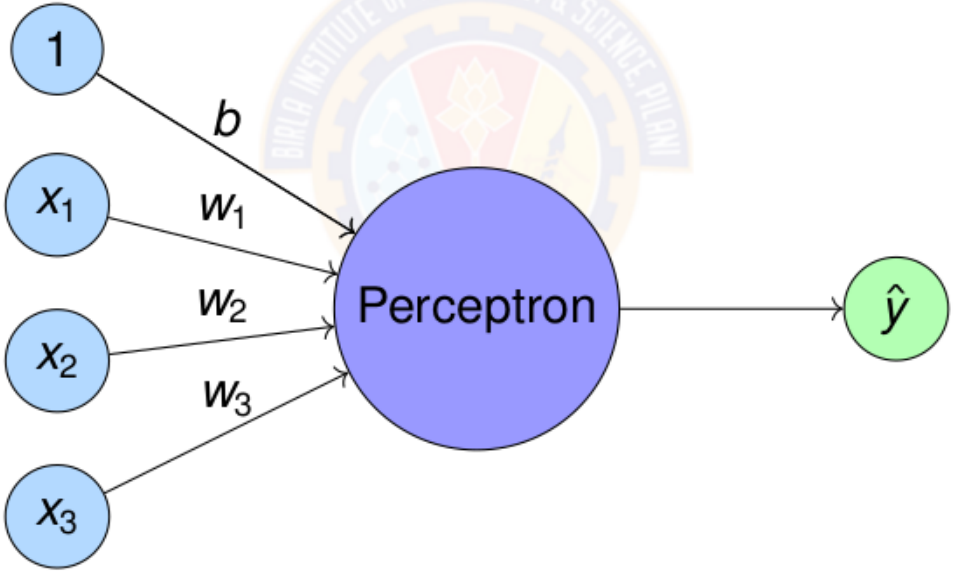
- Intelligence emerges from simple processing neurons
- Knowledge is stored in connection weights
- Learning modifies connection strengths
- Parallel distributed processing



Perceptron

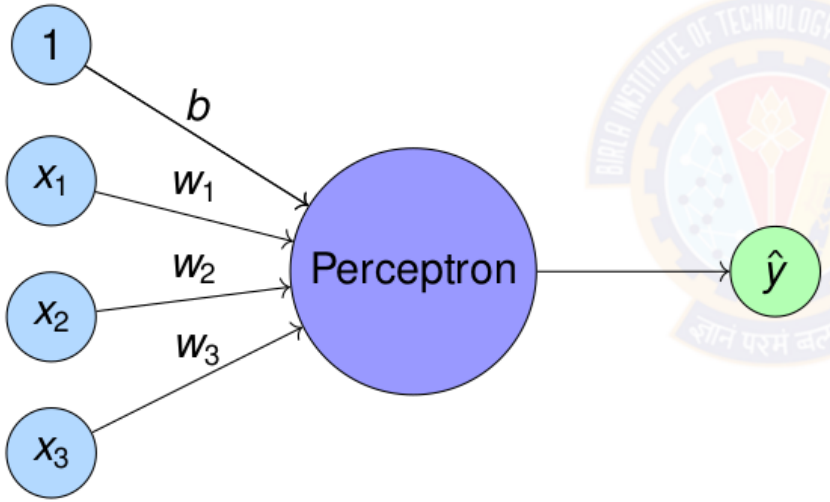
Perceptron

- Perceptron = Simplest form of artificial neuron
- Perceptron was introduced by Rosenblatt in 1958.



Perceptron

- **Perceptron = Simplest form of artificial neuron**



$$\hat{y} = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_i + b \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Perceptron & Logic Gates

Perceptron

- Any linear decision can be represented by a Boolean equation.
- The quest is to find out how to represent the above using Perceptron.
- For this we test whether each of the logic gates can be represented by a Perceptron.

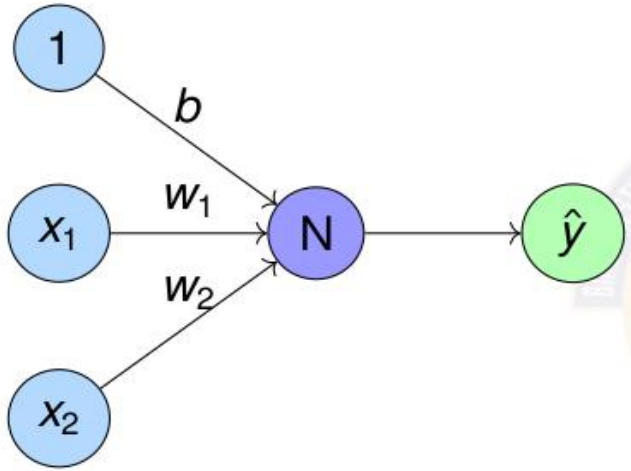
AND Logic Gate

Question:

- How to represent AND gate using a perceptron?
- What are the parameters for the AND perceptron?
- Data is given below.

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

Perceptron for AND gate



Goal: Find parameters w_0, w_1, w_2

x_1	x_2	t	h
0	0	0	$w_0 + 0 + 0 < 0$
0	1	0	$w_0 + 0 + w_2 < 0$
1	0	0	$w_0 + w_1 + 0 < 0$
1	1	1	$w_0 + w_1 + w_2 \geq 0$

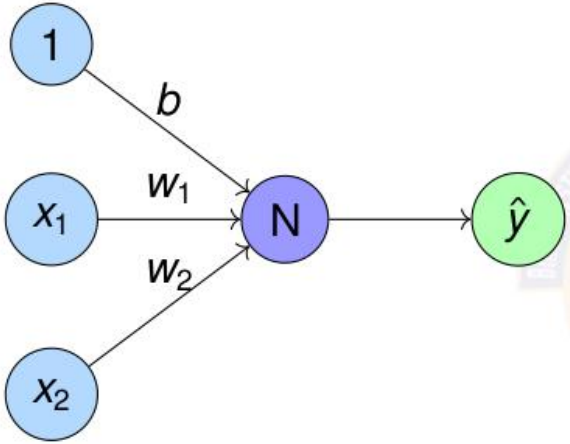
One solution is

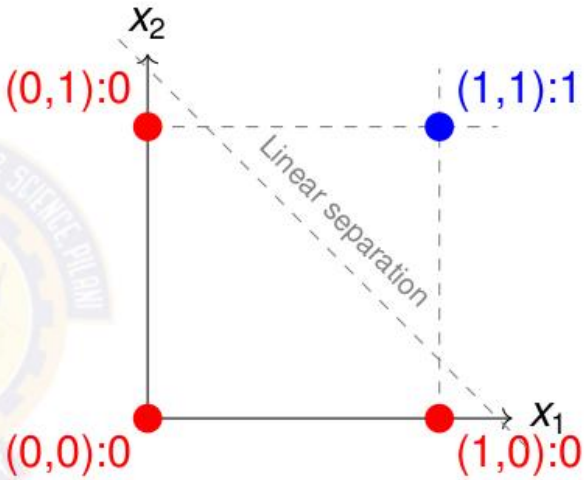
Perceptron equation is

$$\hat{y} = w_0 x_0 + w_1 x_1 + w_2 x_2$$

$$w_0 = -1$$
$$w_1 = 0.75$$
$$w_2 = 0.75$$

Perceptron for AND gate





Perceptron equation is

$$\hat{y} = w_0 x_0 + w_1 x_1 + w_2 x_2$$

One solution is

$$w_0 = -1$$
$$w_1 = 0.75$$
$$w_2 = 0.75$$

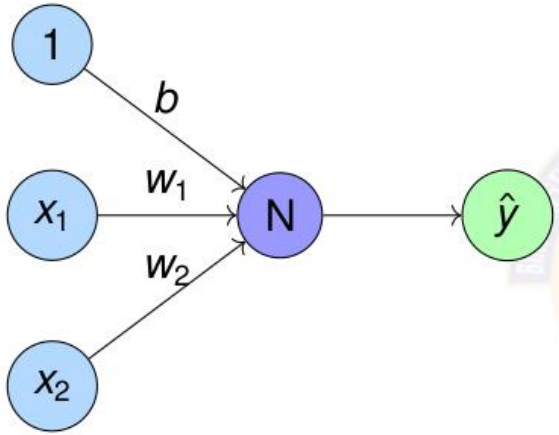
OR Logic Gate

Question:

- How to represent OR gate using a perceptron?
- What are the parameters for the OR perceptron?
- Data is given below.

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

Perceptron for OR gate



Goal: Find parameters w_0, w_1, w_2

x_1	x_2	t	h
0	0	0	$w_0 + 0 + 0 < 0$
0	1	1	$w_0 + 0 + w_2 \geq 0$
1	0	1	$w_0 + w_1 + 0 \geq 0$
1	1	1	$w_0 + w_1 + w_2 \geq 0$

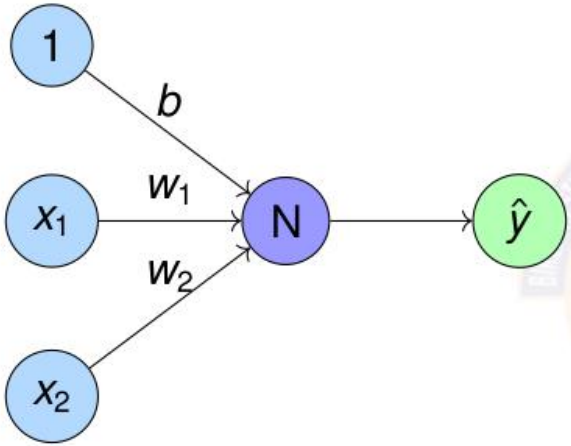
One solution is

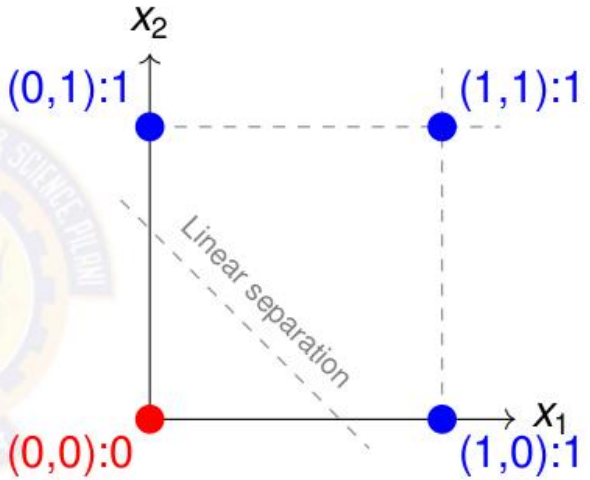
Perceptron equation is

$$\hat{y} = w_0 x_0 + w_1 x_1 + w_2 x_2$$

$$w_0 = -1$$
$$w_1 = 2$$
$$w_2 = 2$$

Perceptron for OR gate





Perceptron equation is

$$\hat{y} = w_0 x_0 + w_1 x_1 + w_2 x_2$$

One solution is $w_0 = -1$

$w_1 = 2$

$w_2 = 2$

Exercise

1. Represent OR gate using Perceptron. Compute the parameters of the perceptron.
2. Represent NOR gate using Perceptron. Compute the parameters of the perceptron.
3. Represent NAND gate using Perceptron. Compute the parameters of the perceptron.

Perceptron Learning Algorithm

Perceptron Learning Algorithm

Goal: Find weights that correctly classify training data (input, output) pair (x, t).

Algorithm:

1. Initialize learning rate $\eta = 0.1$
2. Initialize weights randomly: $w_i = \text{random}$
3. For each training example (x, t):
   - Calculate output: $\hat{y} = sign(\sum_i w_i x_i + b)$
   - If $\hat{y} \neq t$, update weights:

$$w_i^{new} = w_i^{old} + \eta(t - \hat{y})x_i$$

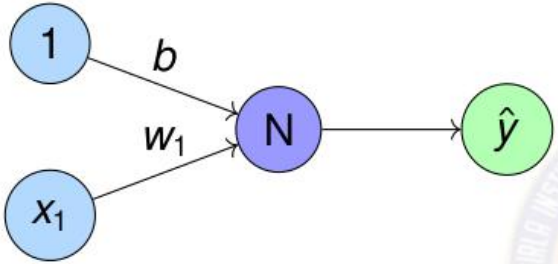
$$b^{new} = b^{old} + \eta(t - \hat{y})$$

4. Repeat until convergence or max iterations

Convergence Theorem:

If linearly separable data, Perceptron learning algorithm will converge in finite steps.

Perceptron Learning for NOT gate

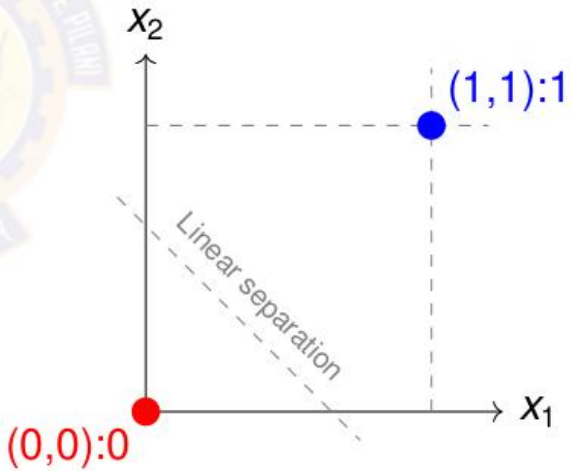


Example	x_1	t
Eg:1	-1	1
Eg:2	1	-1

Perceptron equation

$$h = sign(w_0 x_0 + w_1 x_1)$$

$$\hat{y} = \begin{cases} 1 & \text{if } h \geq 0 \\ -1 & \text{if } h < 0 \end{cases}$$



Perceptron Learning for NOT gate

Assume $w_0 = w_1 = 0$. Let the learning rate$= \eta = 1$.

- Epoch 1 Example 1 pair (1,-1)

$$\hat{y} = sign(w_1 x_1 + w_0) = sign(0 \cdot -1 + 0) = -1$$

Is $\hat{y}$ equal to t? NO. Hence update weights.

$$w_i^{(new)} = w_i(old) + \eta(t - \hat{y})x_i$$
$$w_1^{(new)} = w_1(old) + \eta(t - \hat{y})x_1 = 0 + 1(1 - (-1)) \cdot (-1) = (-2)$$
$$w_0^{(new)} = w_0(old) + \eta(t - \hat{y})x_0 = 0 + 1(1 - (-1)) = (2)$$

Perceptron Learning for NOT gate

- Epoch 1 Example 2 pair (1,-1)

$$\hat{y} = sign(w_1 x_1 + w_0) = sign((-2) \cdot 1 + 2) = sign(0) = -1$$

  Is $\hat{y}$ equal to t? YES. NO update weights.
- Epoch 2 Example 1 pair (-1,1)

$$\hat{y} = sign(w_1 x_1 + w_0) = sign((-2) \cdot (-1) + 2) = sign(4) = 1$$

  Is $\hat{y}$ equal to t? YES. NO update weights.
- Epoch 2 Example 2 pair (1,1)

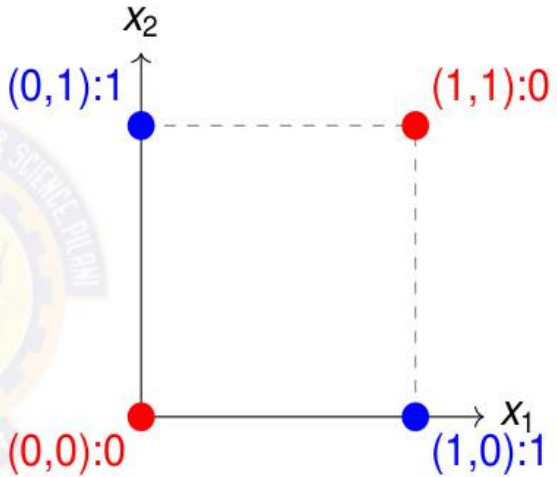
$$\hat{y} = sign(w_1 x_1 + w_0) = sign((-2) \cdot 1 + 2) = sign(0) = -1$$

  Is $\hat{y}$ equal to t? YES. NO update weights.
- Algorithm Converges

The XOR Problem

Truth Table for XOR:

x_1	x_2	XOR Output
0	0	0
0	1	1
1	0	1
1	1	0



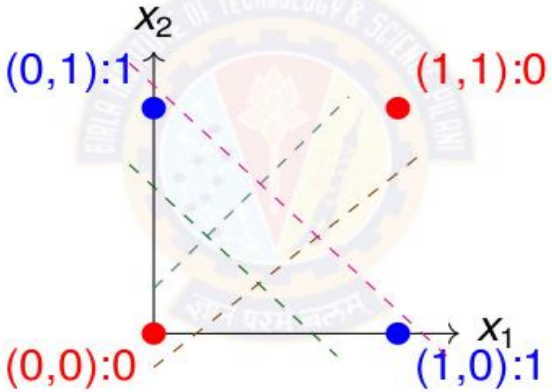
PROBLEM

XOR is not linearly separable! Single perceptron cannot solve XOR.

Why Single Perceptron Fails on XOR

Linear Decision Boundary Limitation:

Any line $w_1 x_1 + w_2 x_2 + b = 0$ cannot separate XOR classes.



No single line works!

Solution: Need multiple layers (Multilayer Perceptron)

Perceptron Demo Python Code

https://colab.research.google.com/drive/1DUVcOoUIWhl8GQKc6AWR1wi0LaMeNkgD?usp=sharing

Student pl note:

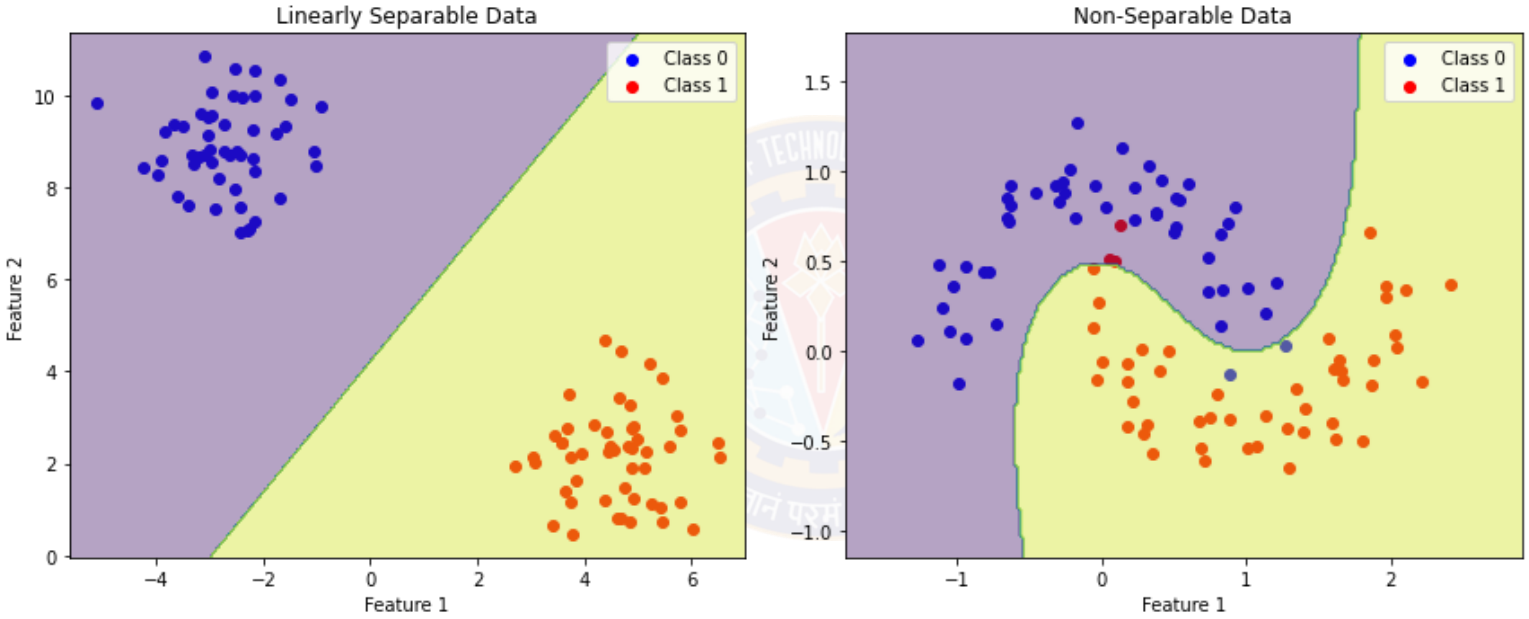
The Python notebook is shared for anyone who has access to the link and the access is restricted to use BITS email id. So pl do not access from non-BITS email id and send requests for access.

Linearly Separable Data

Linearly Separable Data

- **Two sets data points are separable in a n-dimensional space if they can be separated by an (n-1) dimensional hyperplane.**
- A straight line can be drawn to separate all the data examples belonging to class +1 from all the examples belonging to the class -1. Then the two-dimensional data are clearly linearly separable.
- An infinite number of straight lines can be drawn to separate the class +1 from the class -1.

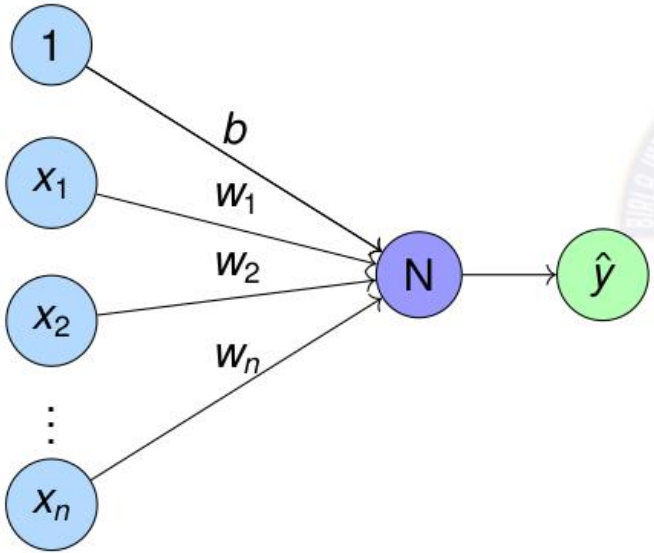
Linearly vs Non-Linearly Separable Data



Perceptron for Linearly Separable Data

- Data
  - Truth tables or set of examples
- Model
  - Perceptron
  - Inputs x are the features or columns
  - Target output t is the required label
- Objective Function
  - Deviation of desired output or target t and the computed output $\hat{y}$
- Learning Algorithm
  - Perceptron Learning algorithm
- Challenge
  - How to learn these n + 1 parameters w_0 , w_1 . w_2 . . . w_n ?
- Solution
  - Use Perceptron learning algorithm.

Perceptron for Linearly Separable Data



$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

$$h = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$
$$= w^\top x$$

$$\hat{y} = \begin{cases} 1 & \text{if } h \geq 0 \\ -1 & \text{if } h < 0 \end{cases}$$

Exercise

- Represent OR gate using Perceptron. Compute the parameters of the perceptron using perceptron learning algorithm.
- Represent AND gate using Perceptron. Compute the parameters of the perceptron using perceptron learning algorithm.
- Represent NOR gate using Perceptron. Compute the parameters of the perceptron using perceptron learning algorithm.
- Represent NAND gate using Perceptron. Compute the parameters of the perceptron using perceptron learning algorithm.
- Implement a perceptron for AND and OR gates
- Sketch decision boundaries for Single perceptron

Key Concepts

- **Neural Network**: Biological inspiration and mathematical models
- **Perceptron**: Simplest neural unit, linear classification, learning algorithm
- **XOR Problem**: Limitation of linear models, need for hidden layers
- **Learning Paradigms**: Supervised, unsupervised, reinforcement learning

References

1. Zhang et al. "Dive into Deep Learning" - Chapter 1 https://d2l.ai/
2. http://mlsp.cs.cmu.edu/people/rsingh/docs/Chapter1_Introduction.pdf
3. Goodfellow, Bengio, Courville - "Deep Learning" Chapters 1-6
4. Rosenblatt, F. (1958) - "The Perceptron: A Probabilistic Model”
5. Minsky & Papert (1969) - "Perceptrons" (Historical perspective)

Thanks