Deep Learning

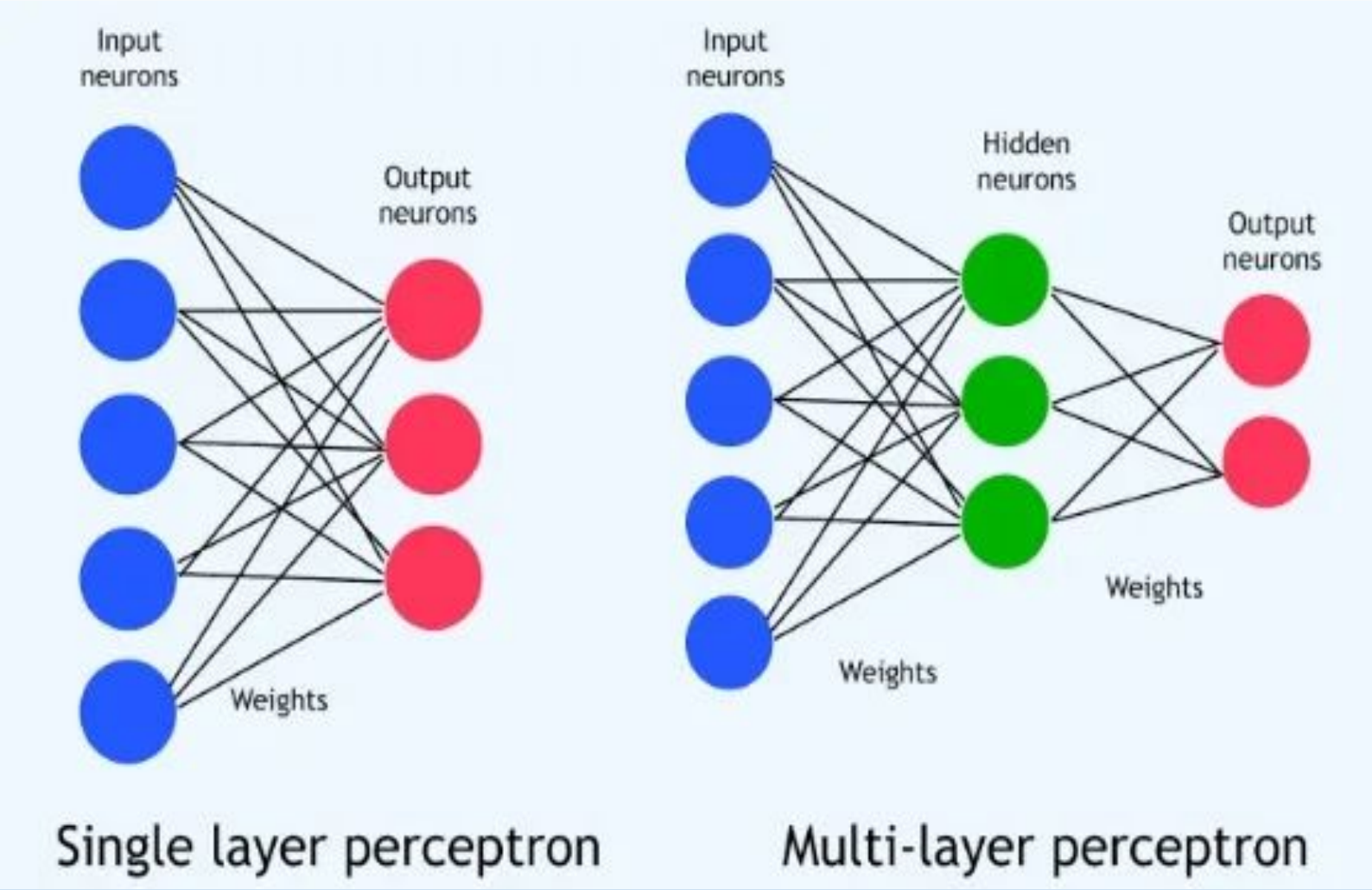
Feed Forward Network

Gananath Bhuyan
Assistant Professor
School of Computer Engineering
KIIT DU, Bhubaneswar

Deep Feed Forward Network

- The goal of a feedforward network is to approximate some function f^* without any looping back to the previous layer.
- Directed acyclic graph.
- $y = f^*(x)$ ⇨ if y is an image then y is the class of it (dog/cat)
- $y = f(x;\theta)$ ⇨ $y = f(x;w,b)$ ⇨ $y = wx + b$
- $y = (f^n \ldots\ldots (f^2 (f^1 (x)))\ldots.)$
- Write the equation for 3 layer FFN.

Single and Multi Layer Perceptrons

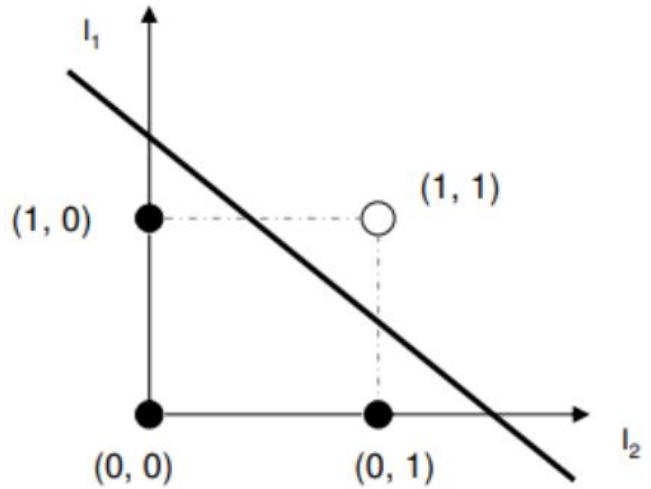


- perceptron

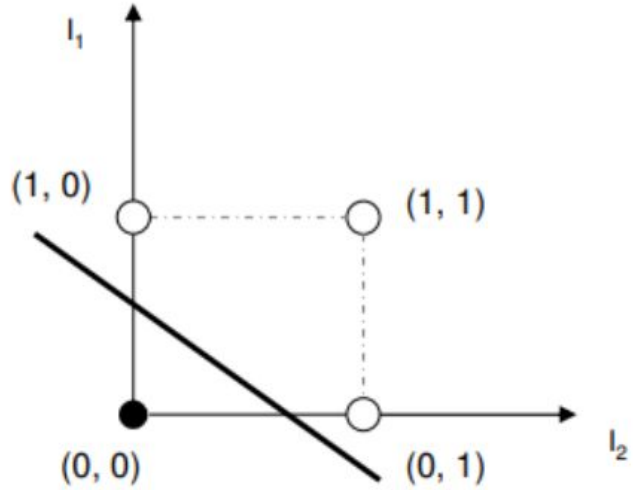
$$y = \begin{cases} 0, & \text{if } w \cdot x + b \leq 0 \\ 1, & \text{if } w \cdot x + b > 0 \end{cases}$$

AND and OR gate implementation

AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1

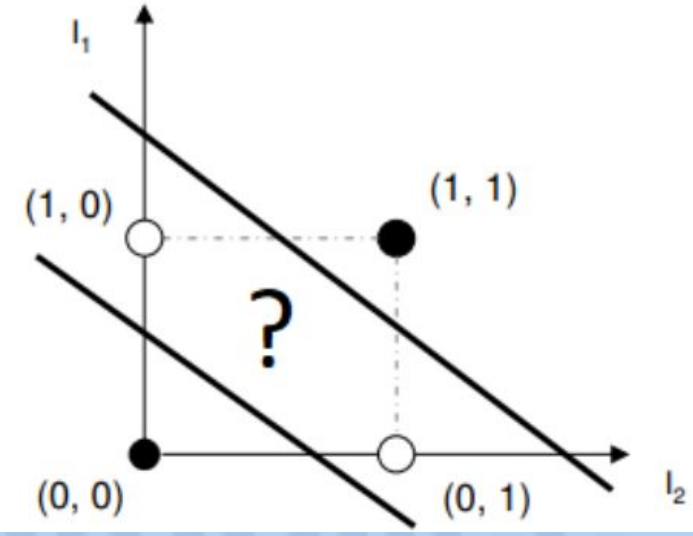


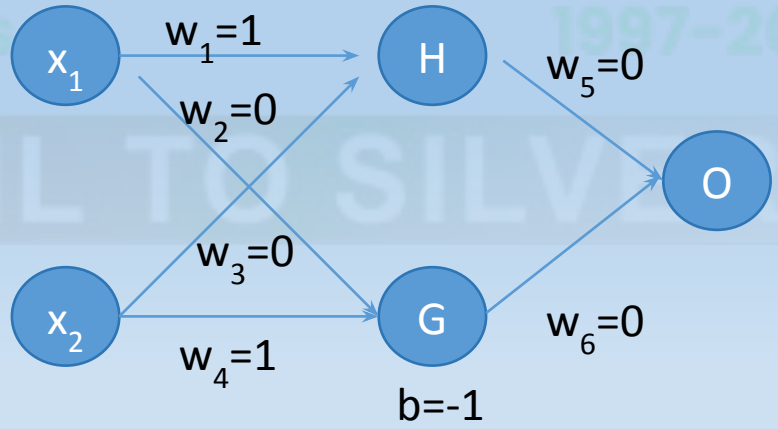
XOR gate implementation

- Why Single-Layer Perceptrons Fail?

- Single-layer perceptrons can model only linearly separable functions (like AND, OR).
- XOR's output requires a non-linear decision boundary.
- Solution: Add hidden layers with non-linear activations.
- Inputs: A and B as coordinates (0 or 1).
- XOR outputs 1 for (0,1) and (1,0) — diagonally opposite points.
- No single straight line can separate outputs of 0 and 1.
- Implies need for transformation via hidden H_1 neurons.

XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0





Mathematical Flow of Computation

- Input: x=[A,B]
- Hidden layer:

$$z = \sigma(w_1 \cdot x + b_1)$$

$$z = \sigma(w_2 \cdot x + b_2)$$

- Output layer:

$$y = \sigma(w_0 \cdot [z_1, z_2] + b_0)$$

- Output
- y approximates A XOR B