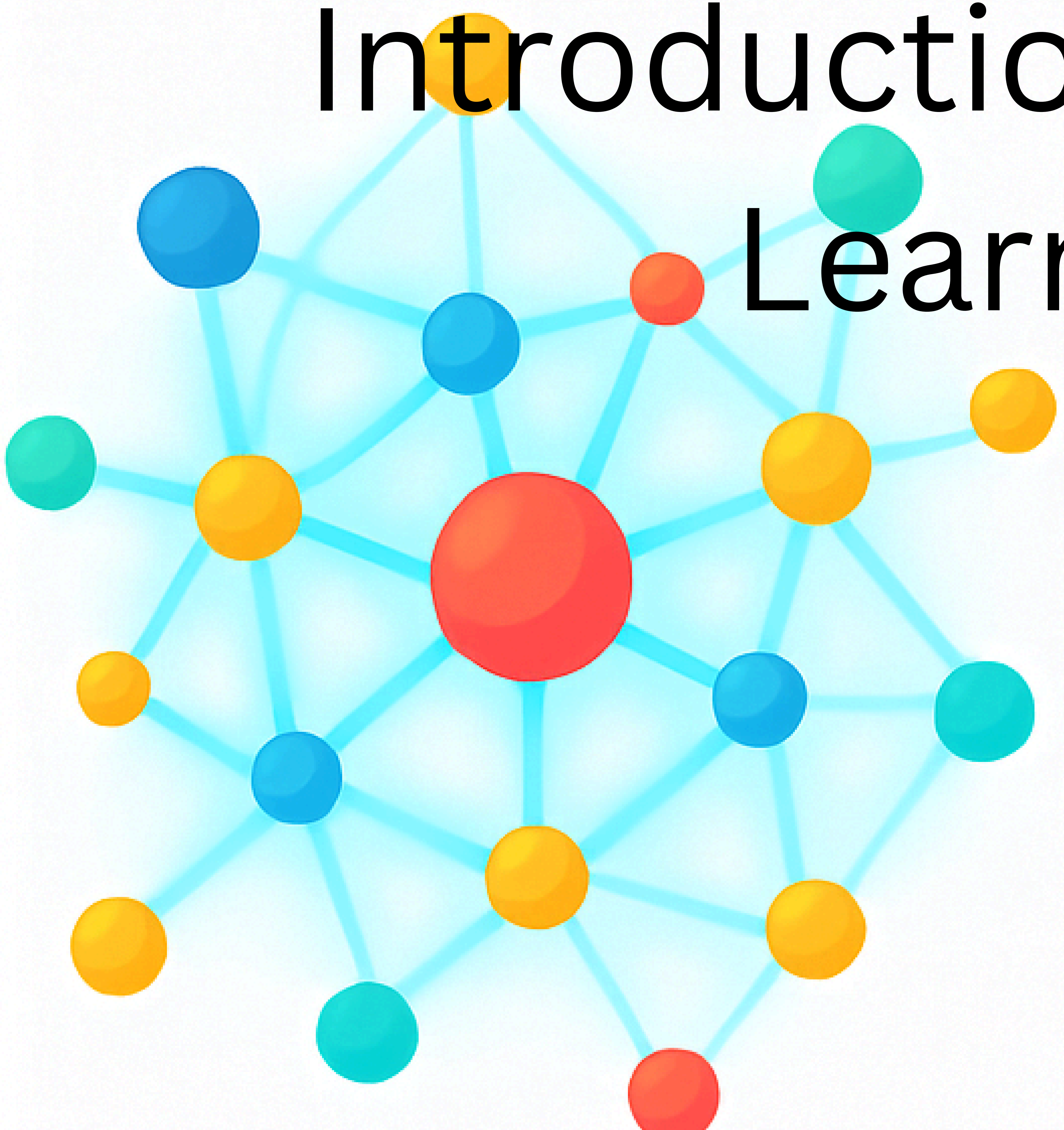
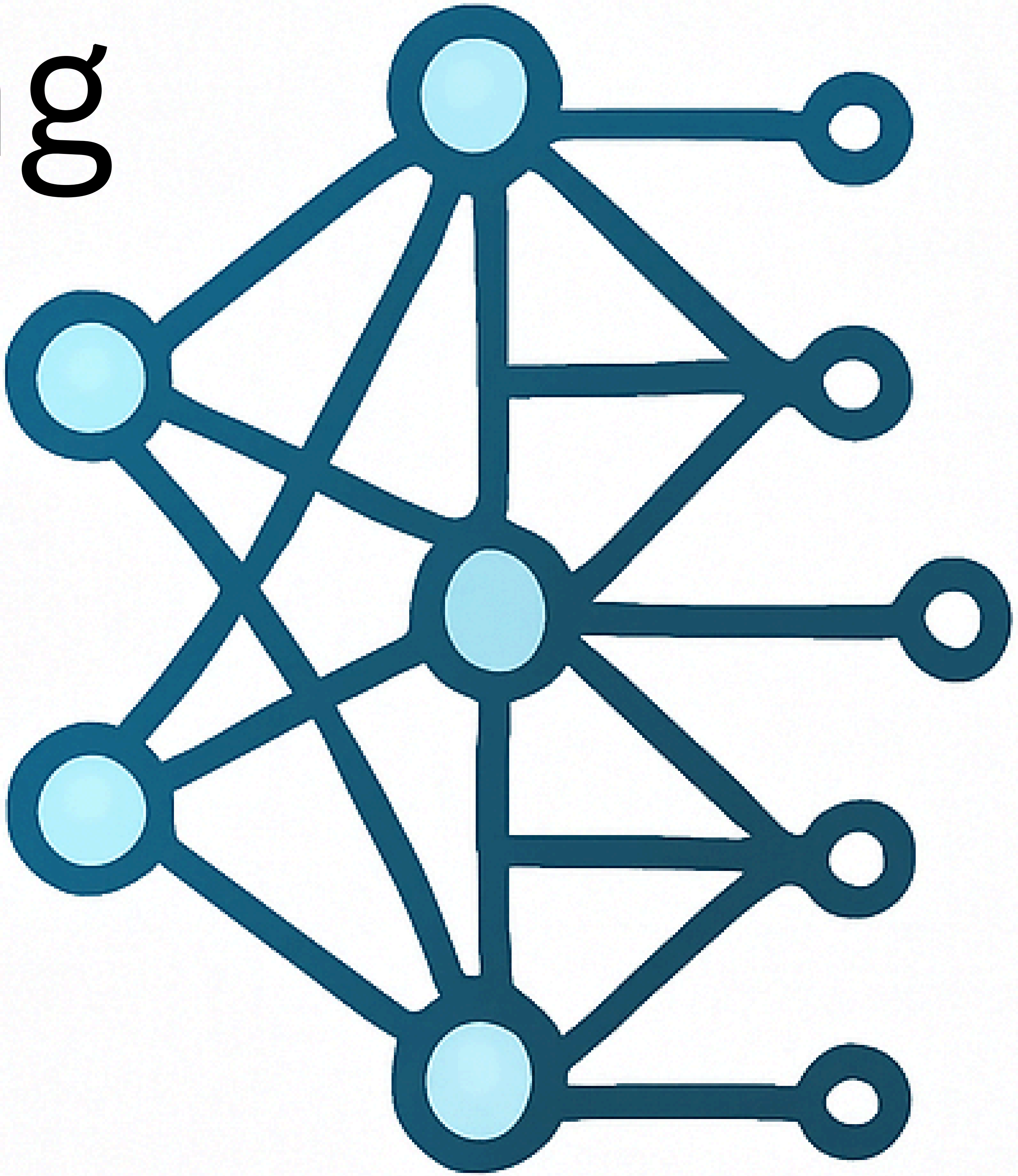
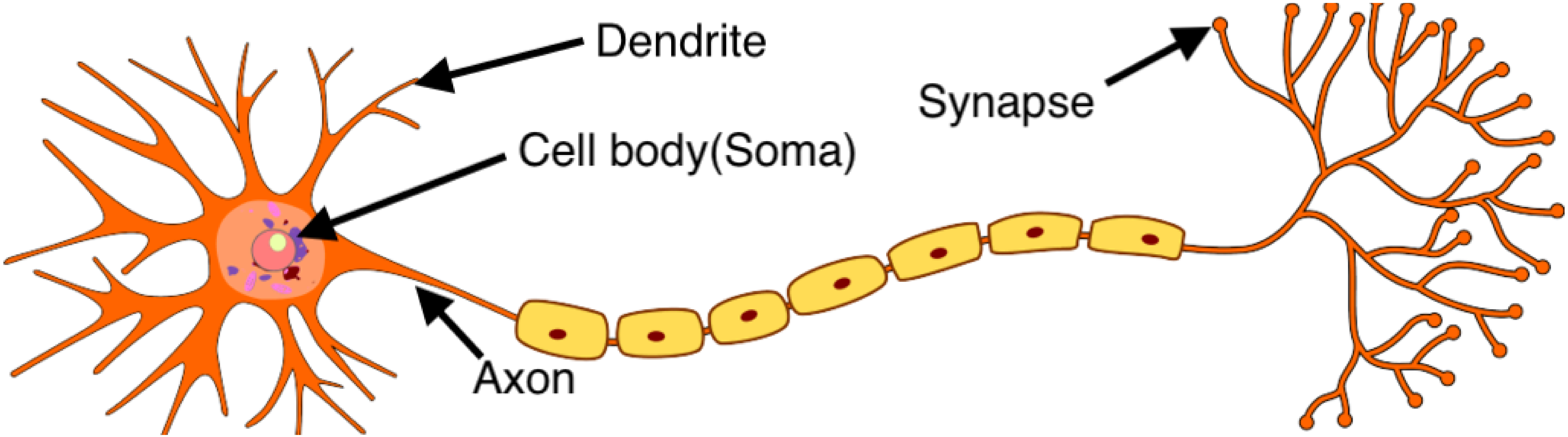
Introduction to Deep Learning



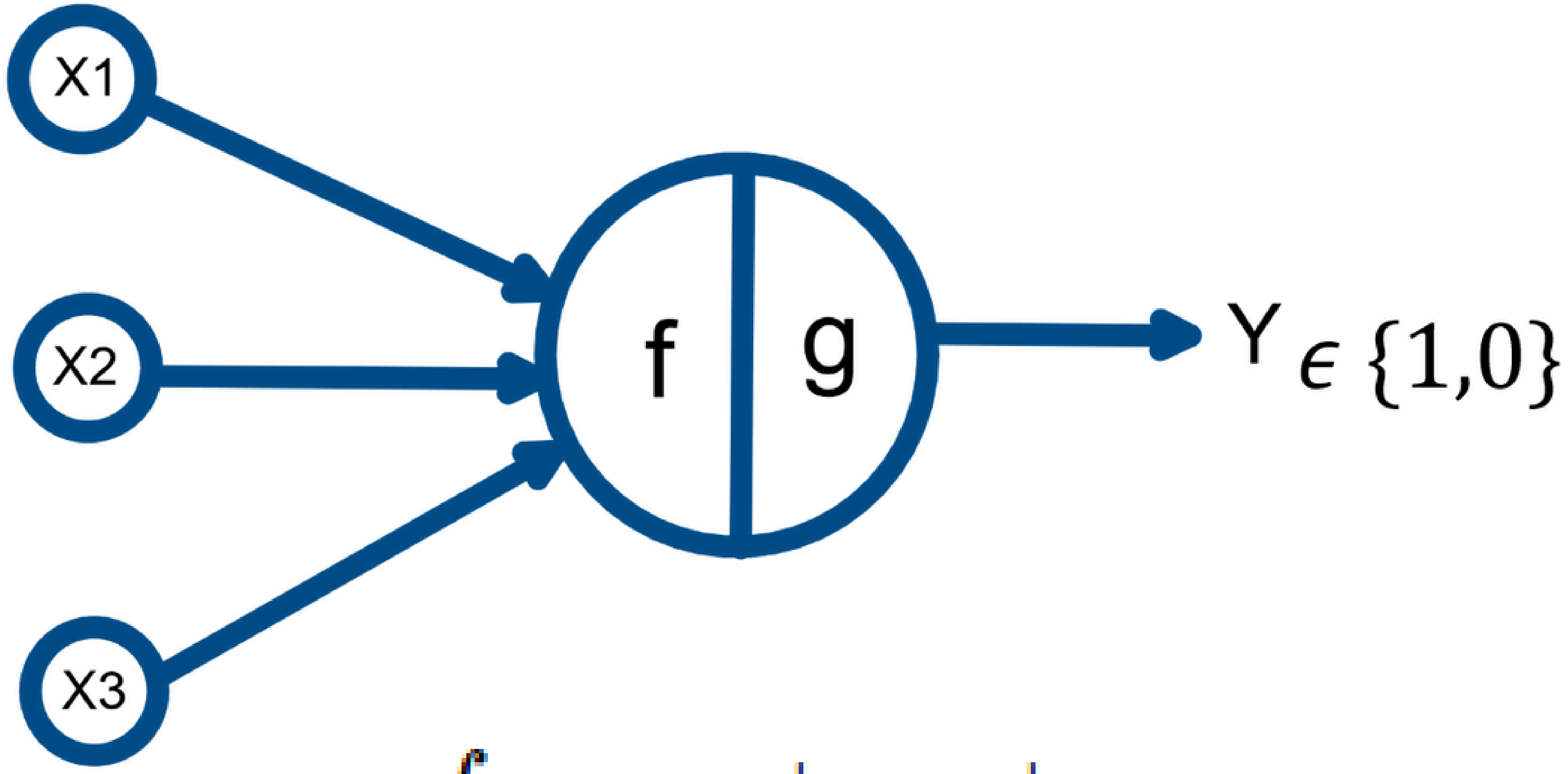


Biological Neuron



McCulloch Pitts Neuron

$x_1,\ x_2,\ x_3 \in \{0,1\}$

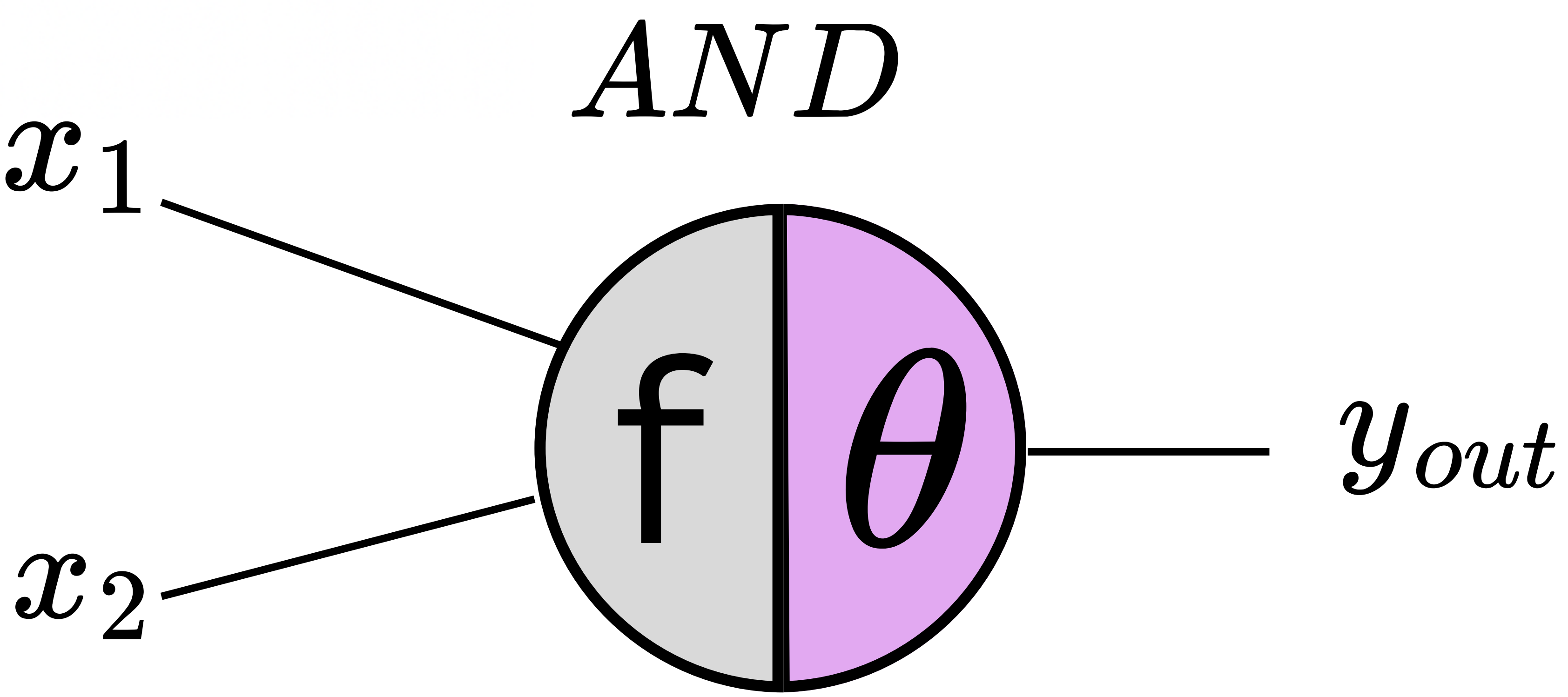


$$f = x_1 + x_2 + x_3$$

$$g(f) = \begin{cases} 1, & f \geq T \\ 0, & f < T \end{cases}$$

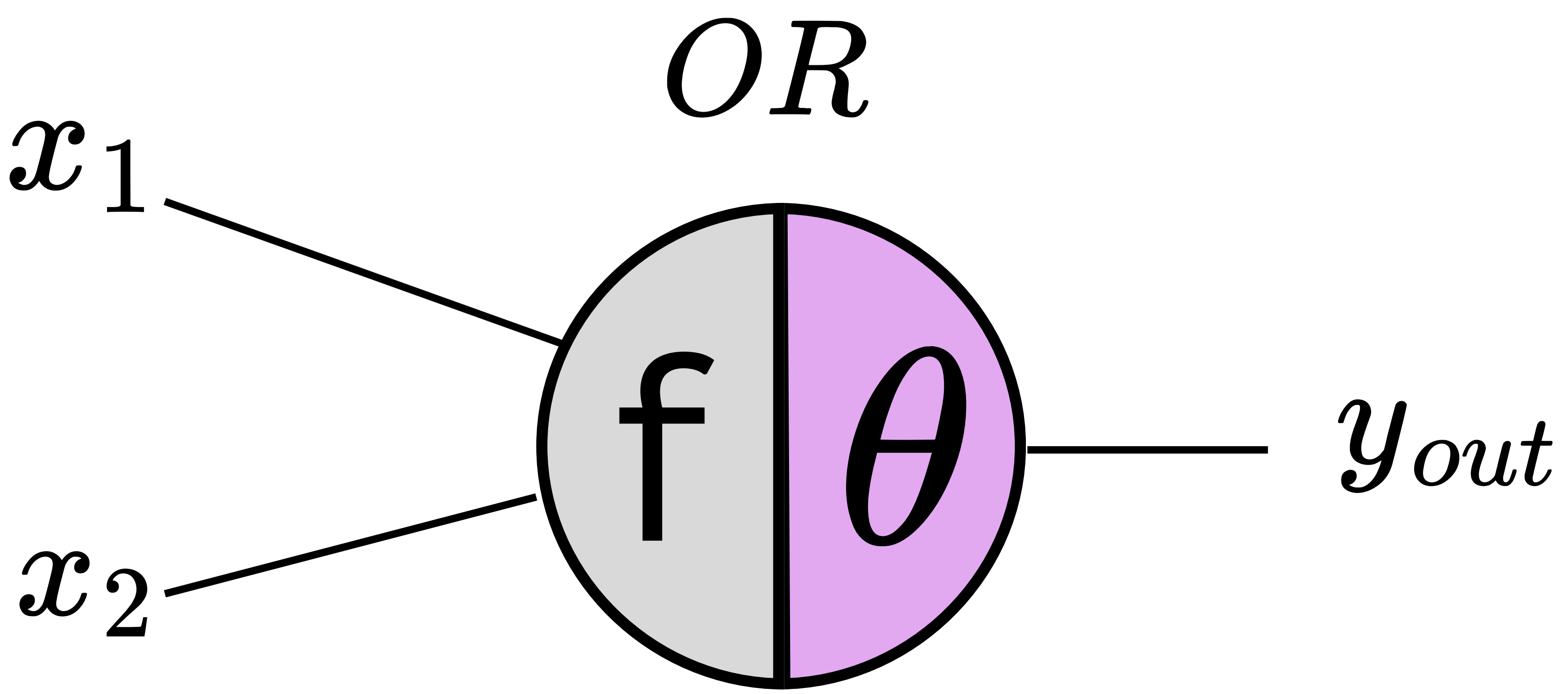
Threshold logic

Logic Gates With MCP neuron



$$\theta \geqslant 2$$

x_1	x_2	$x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1

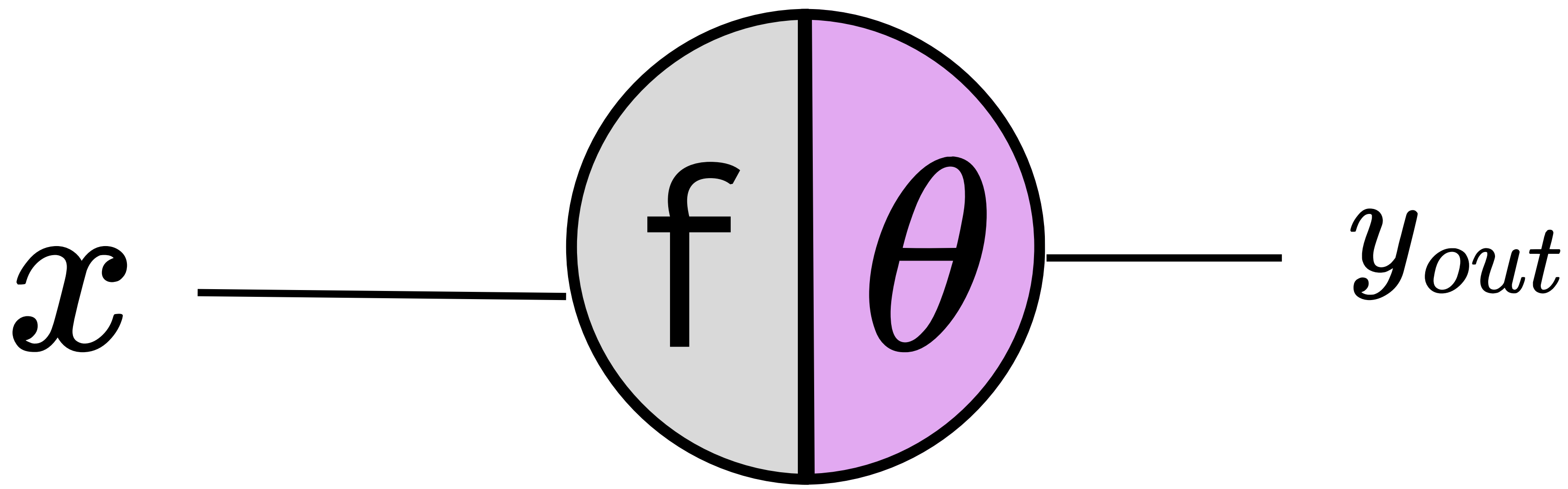


$$\theta \geqslant 1$$

x_1	x_2	$x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1

$$x_i \in \{0, 1\}$$

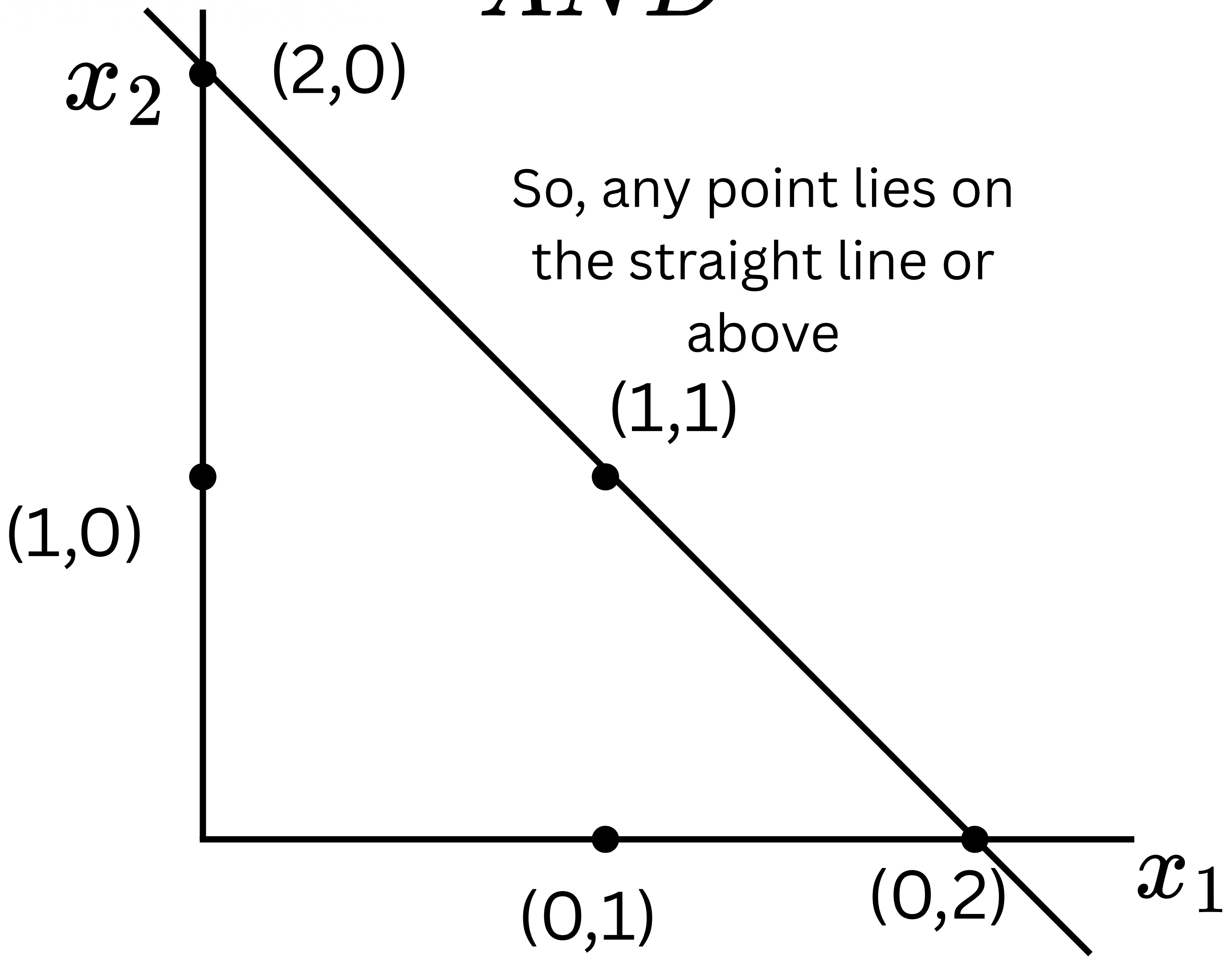
Logic Gates With MCP neuron



$$x \in \{0, 1\}$$

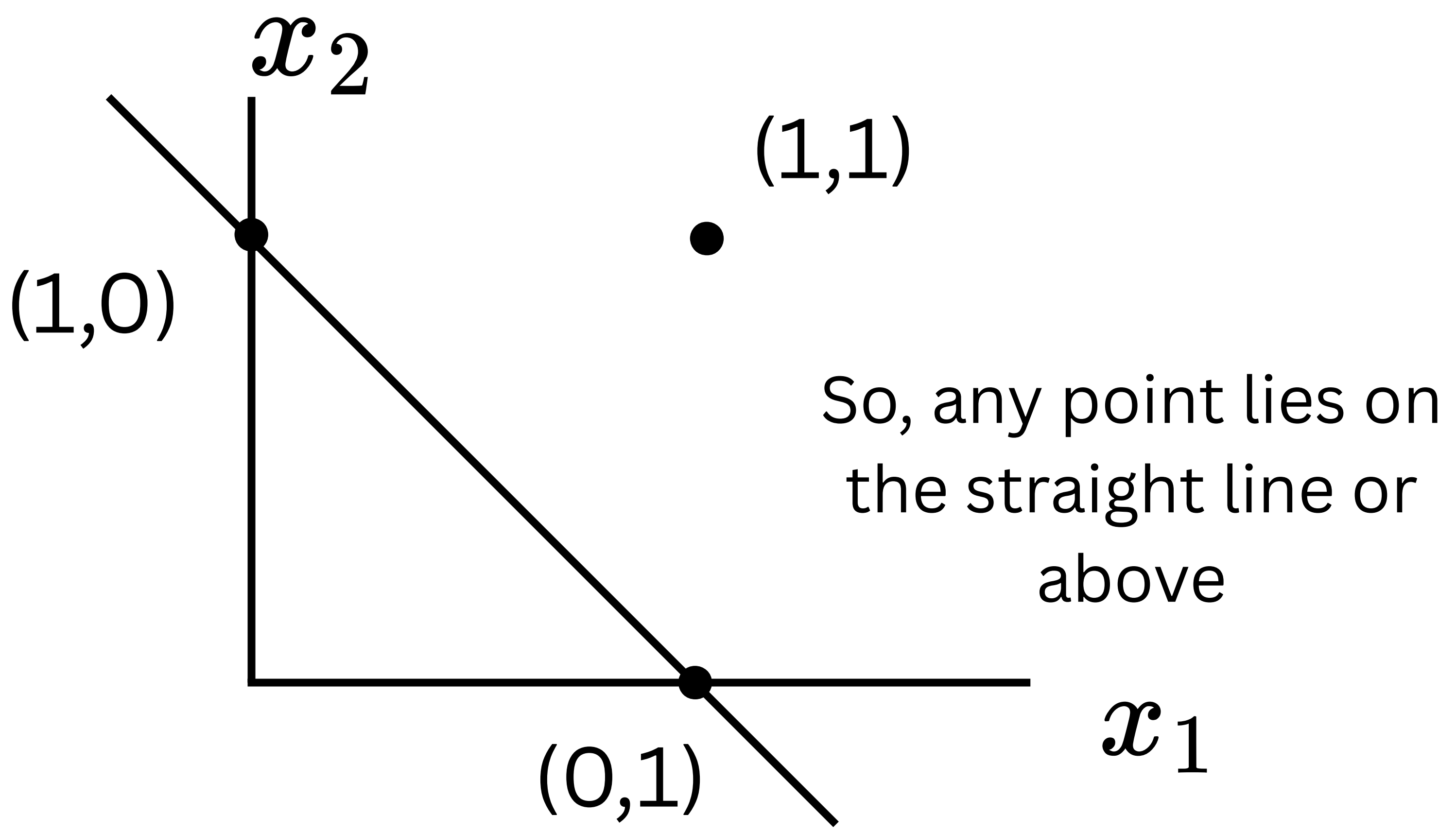
Logic Gates With MCP neuron

AND



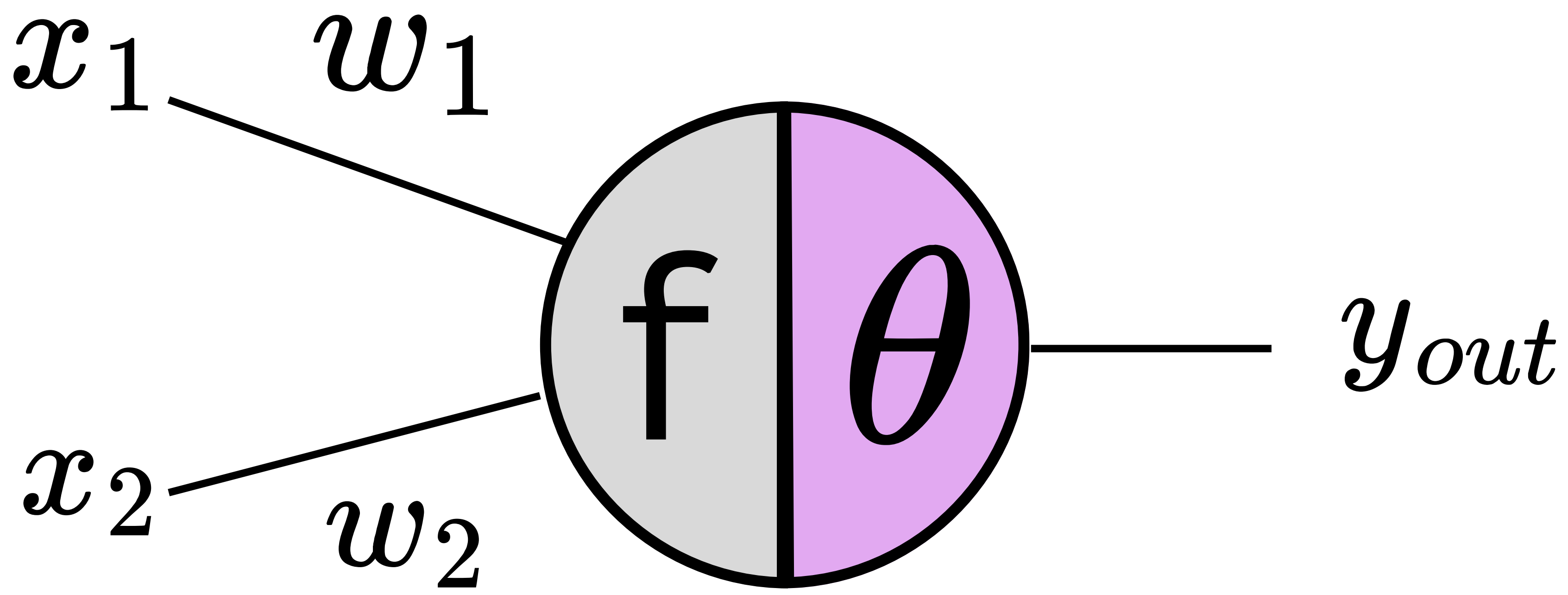
$$\theta \geqslant 2$$

OR



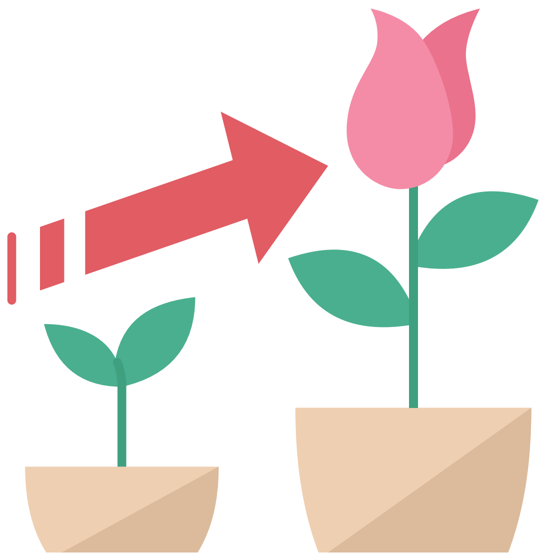
$$\theta \geqslant 1$$

Perceptron

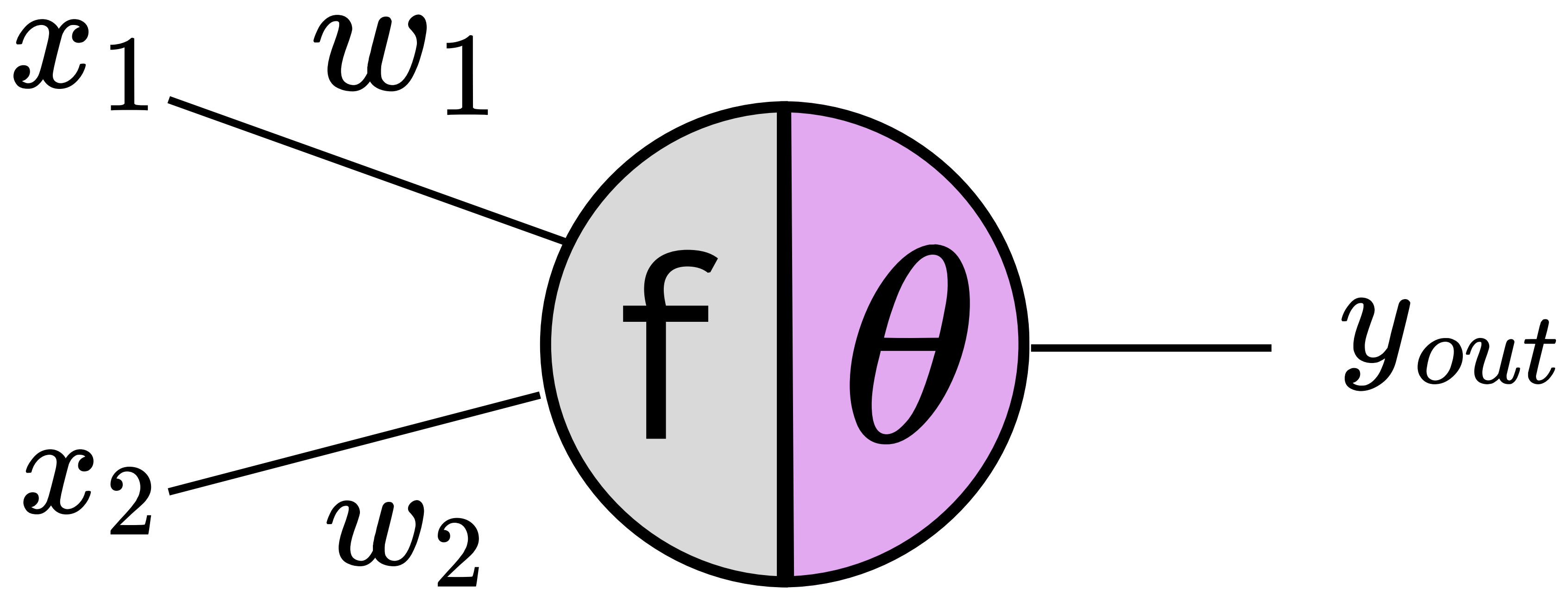


$$f = \sum_{1}^{2} w_i x_i$$

- It is the same idea followed by MCP neuron architecture except the introduction of weights
- The weight gives power to keep decision boundary at any place

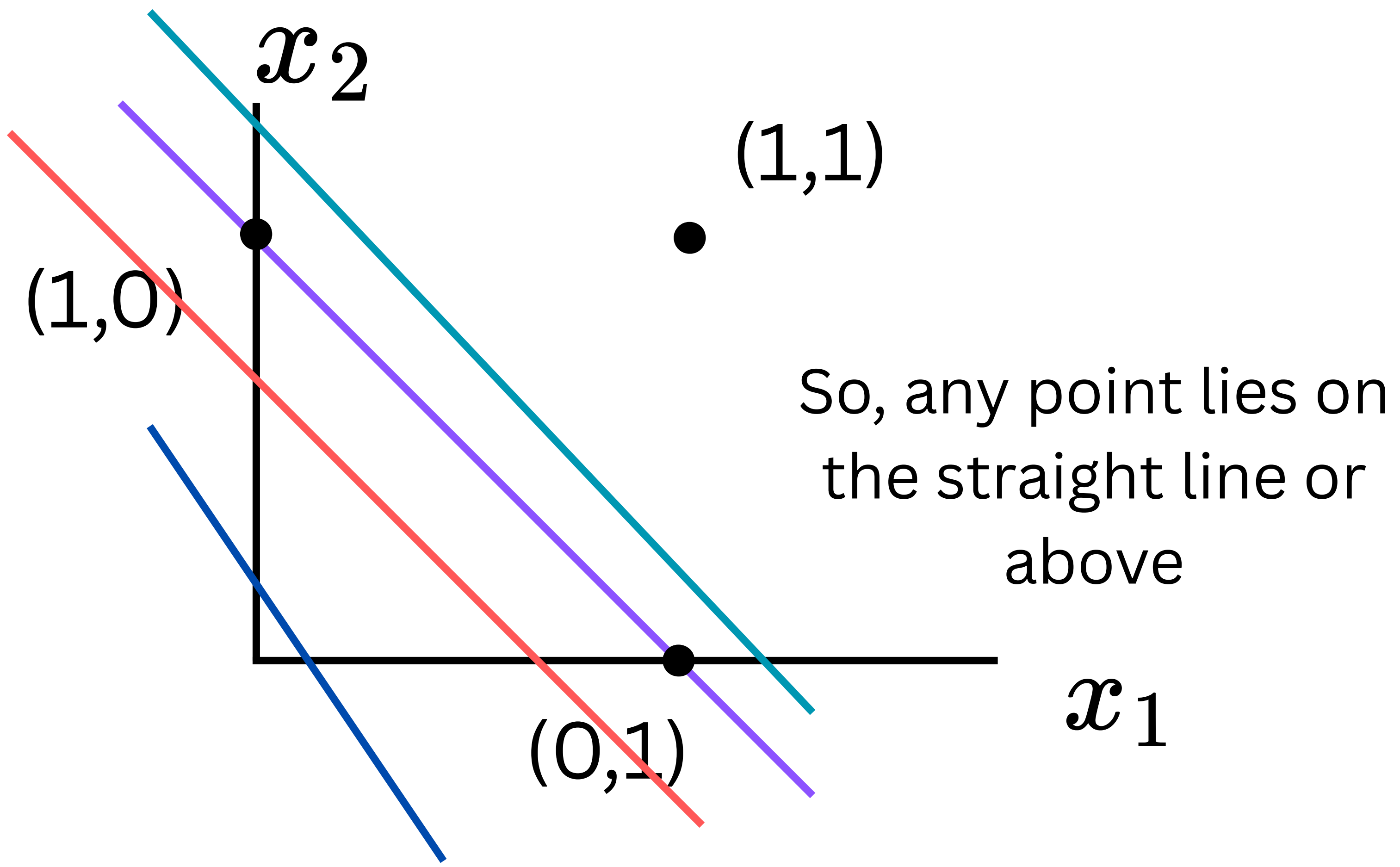


Perceptron

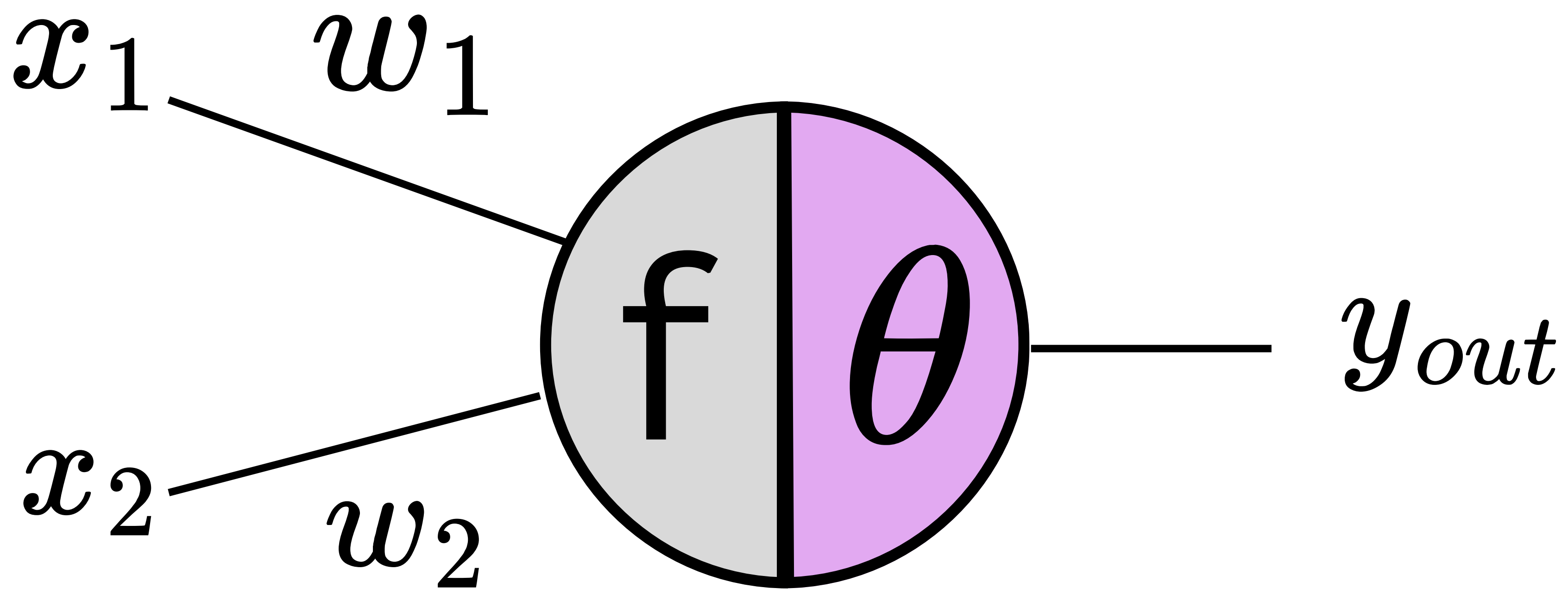


$$f = \sum_{1}^{2} w_i x_i$$

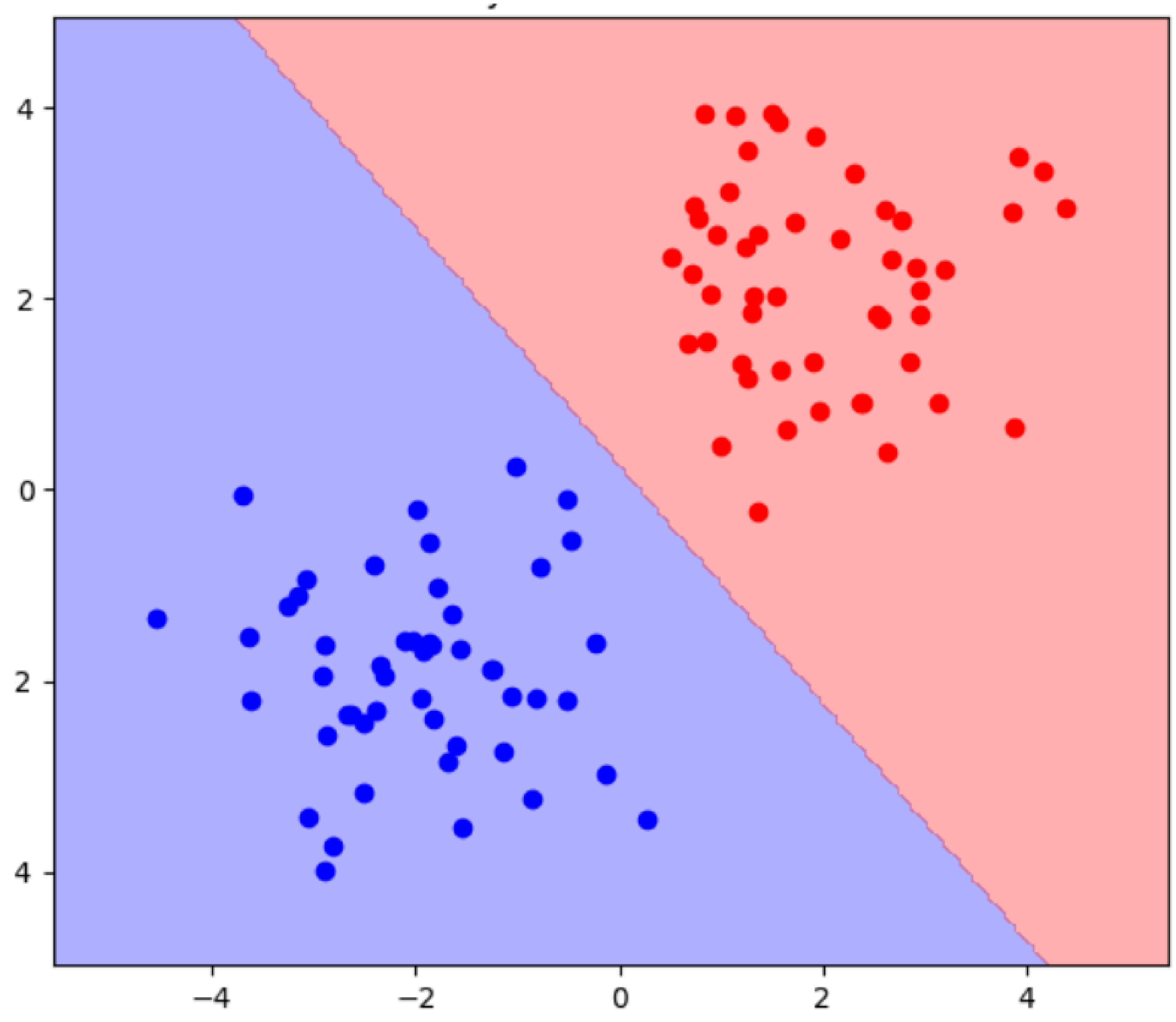
Decision Boundary



Perceptron



$$f = \sum_{1}^{2} w_i x_i$$



Perceptron Learning Algorithm

Algorithm: Perceptron Learning Algorithm

$P \leftarrow inputs \quad with \quad label \quad 1;$

$N \leftarrow inputs \quad with \quad label \quad 0;$

Initialize $\mathbf{w}$ randomly;

while !*convergence* **do**

  Pick random $\mathbf{x} \in P \cup N$;

  if $\mathbf{x} \in P \quad and \quad \mathbf{w.x} < 0$ **then**

    $\mathbf{w} = \mathbf{w} + \mathbf{x}$;

  end

  if $\mathbf{x} \in N \quad and \quad \mathbf{w.x} \geq 0$ **then**

    $\mathbf{w} = \mathbf{w} - \mathbf{x}$;

  end

end

//the algorithm converges when all the inputs are classified correctly

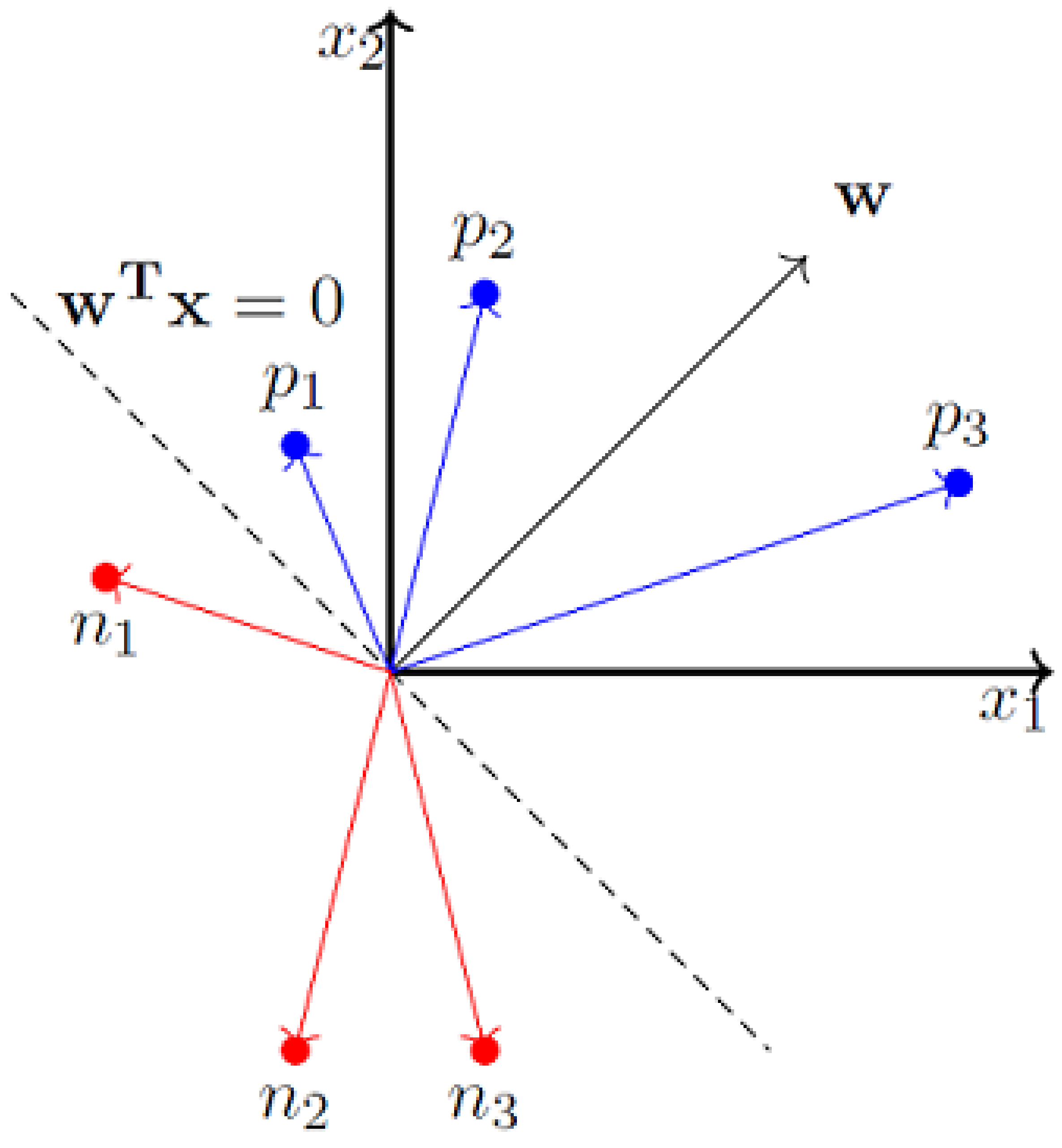
---

Perceptron Learning Algorithm

$$cos\alpha = \frac{\mathbf{w}^T\mathbf{x}}{||\mathbf{w}||||\mathbf{x}||} \quad \Big| \quad cos\alpha \propto \mathbf{w}^T\mathbf{x}$$

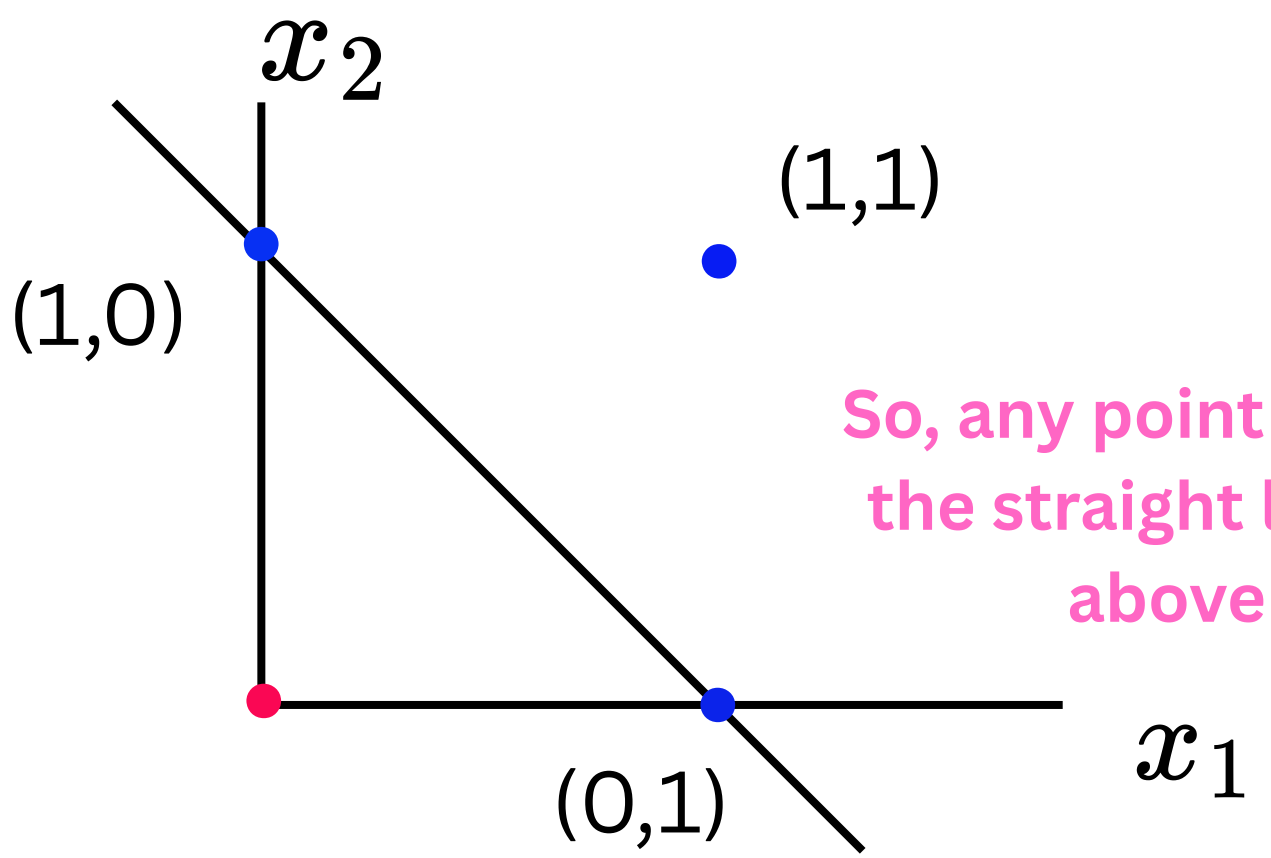
So if $\mathbf{w}^T\mathbf{x} > 0 \;\Rightarrow cos\alpha > 0 \;\Rightarrow \alpha < 90$

Similarly, if $\mathbf{w}^T\mathbf{x} < 0 \;\Rightarrow cos\alpha < 0 \;\Rightarrow \alpha > 90$



Separability and Non-Separable problem

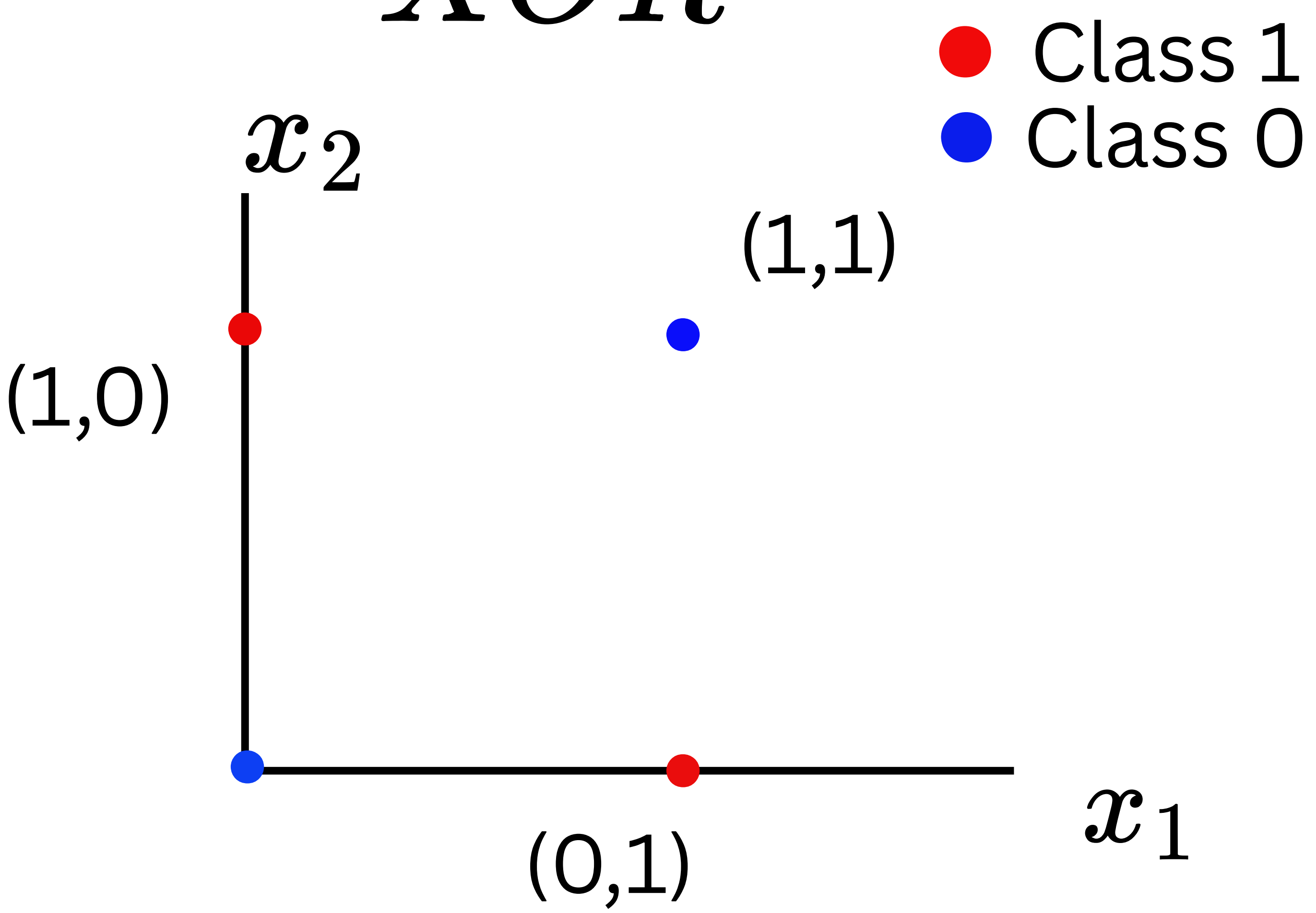
$OR \quad \theta \geqslant 1$



So, any point lies on the straight line or above

- *In above scenario we can easily draw a line to separate these two classes*

XOR



- *In above scenario there is no way that we can draw a line to separate these two classes*
- The problem is called XOR problem

Sigmoid Neurons

- The **XOR problem** has prepared the platform for complex problems which cannot be solved with simple linear **Perceptron.**
- TO address complex problems we need non-linear relationship between input and output.
- It has been introduced in terms of Activation functions.
- Activation functions converts linear inputs to a non-linear form.
- Activation function has many forms, Sigmoid is very popular choice as activation function.