# Artificial Neural Networks (ANN)

Dr. Partha Pratim Sarangi
School of Computer Engineering

# What is Neural Network?

- When we say Neural Network in this course, commonly referred to as Artificial Neural Network (ANN).
- ANN is essentially a computational model inspired by the human brain's network of neurons.
- The human brain is a highly complex, nonlinear, and parallel computer (information-processing system). It comprises nearly 100 billion neurons and 100 trillion interconnections.
- ANN is not an exact representation of the human brain; instead, it consists of a limited set of neurons and their interconnections designed to perform a specific task.
- Today artificial neural networks are used in a myriad of real-life tasks due to their ability to solve problems previously considered impossible to solve such as language translation, video and audio synthesis and autonomous vehicle driving.

# Neural Network: Definition

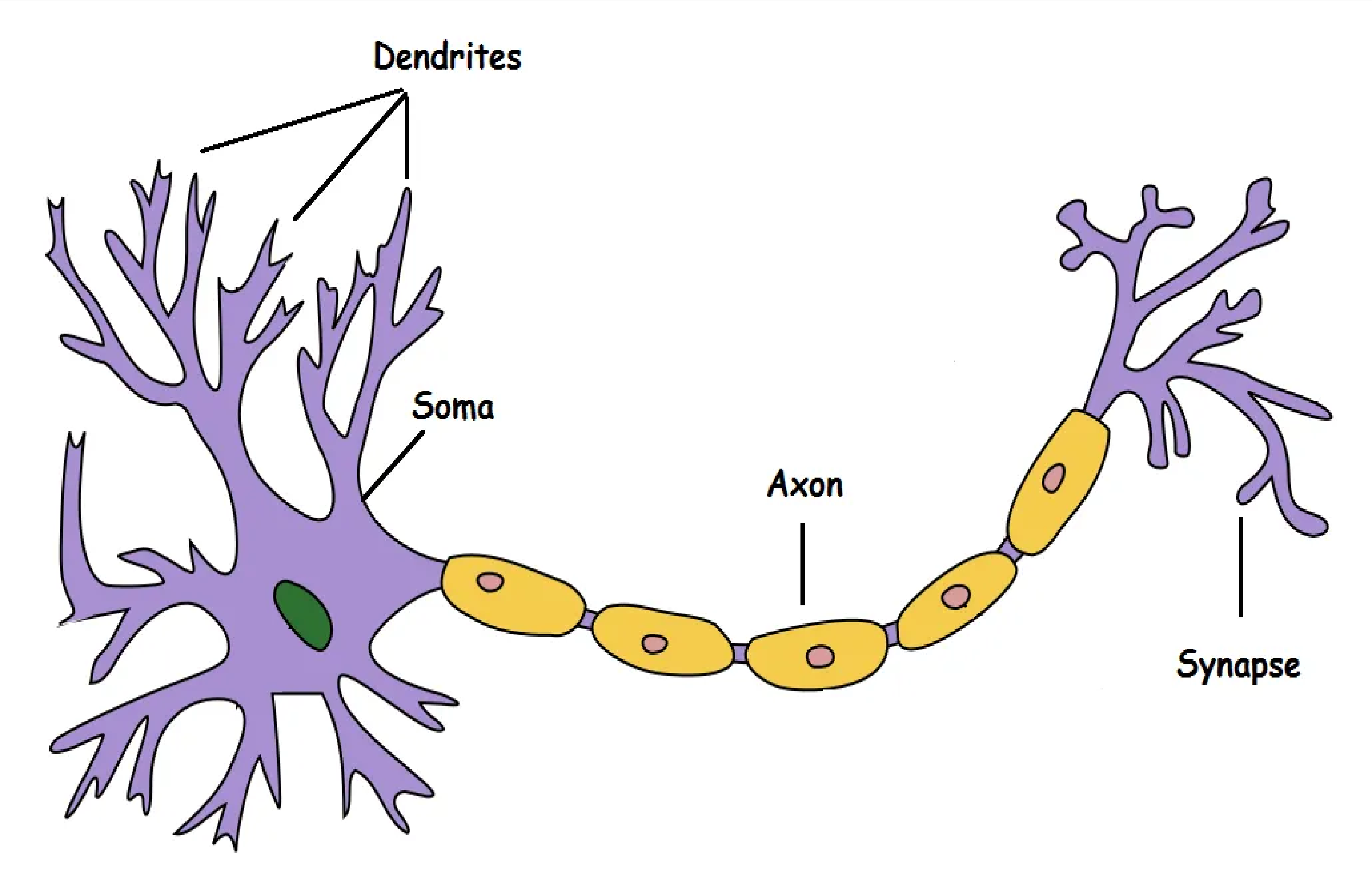
- In order to achieve good performance, neural networks employ a massive interconnection of simple computing cells referred to as “neurons” or “processing units.”

- A neural network is a massively parallel distributed processor made up of simple processing units (neurons), which has a natural propensity for acquiring experiential knowledge from its environment through a learning process and making it available for use.

# Biological Neural Network

- Neural Networks are composed of
  - Cell bodies (Neurons)
  - Dendrites (Inputs)
  - Axons (Outputs)
  - Synapse (Interconnection)
- A neuron (also known as a nerve cell) is a cell that carries electric impulses. Neurons are the basic units of the nervous system and its most important part in the brain.
- Every neuron is made of a cell body (also called a soma), dendrites, and an axon.
- Neurons are connected to one another. They do not touch each other; instead form tiny gaps fills with chemical fluid called synapse that can carry electrical impulses and pass the electric signal from one neuron to the other neurons.

# A Biological Neuron



# A Biological Neuron

- Dendrite: Receives signals from other neurons
- Soma: Processes the information
- Axon: Transmits the output of this neuron
- Synapse: Point of connection to other neurons

# ANN vs. BNN

## Artificial NN

- **Nodes**
  - input
  - output
  - node function
- **Connections**
  - connection strength

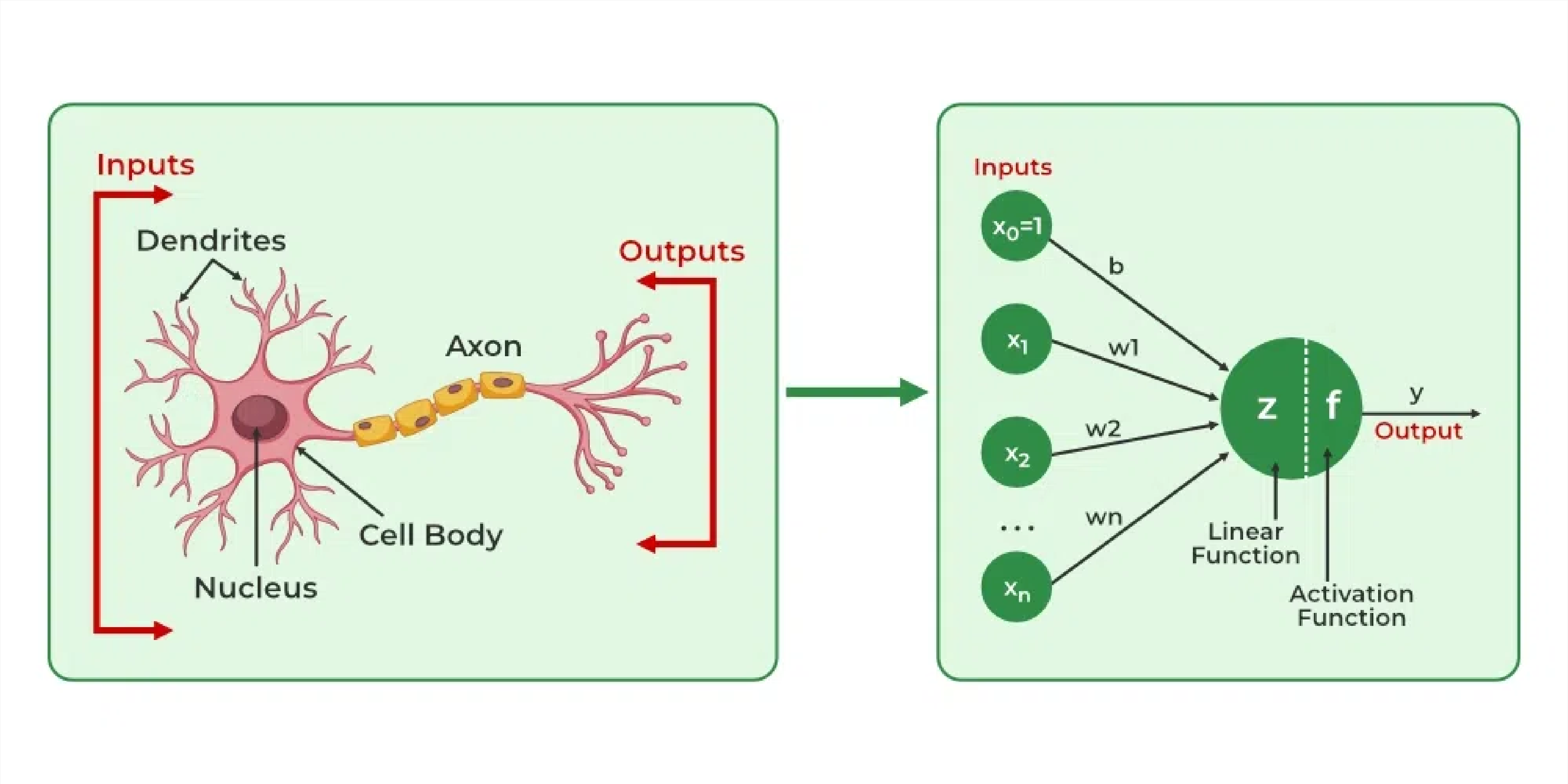
## Biological NN

- **Cell body**
  - signal from other neurons
  - firing frequency
  - firing mechanism
- **Synapses**
  - synaptic strength

# Analogous to BNN and ANN

| Biological Neural Network (BNN) | Artificial Neural Network (ANN) |
|---|---|
| Soma | Node or Neuron |
| Dendrites | Inputs |
| Synapse | Weights or Interconnections |
| Axon | Output |

# Analogy of ANN with BNN



# Artificial Neuron resembles the human brain

- Two aspects make the neuron resembles the brain as follows:
  1. Knowledge is acquired by the network from its environment through a learning process.
  2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.
- The procedure used to perform the learning process is called a learning algorithm, function of which is to modify the synaptic weights of the network in an orderly fashion to obtain a desired model.

# What ANNs do?

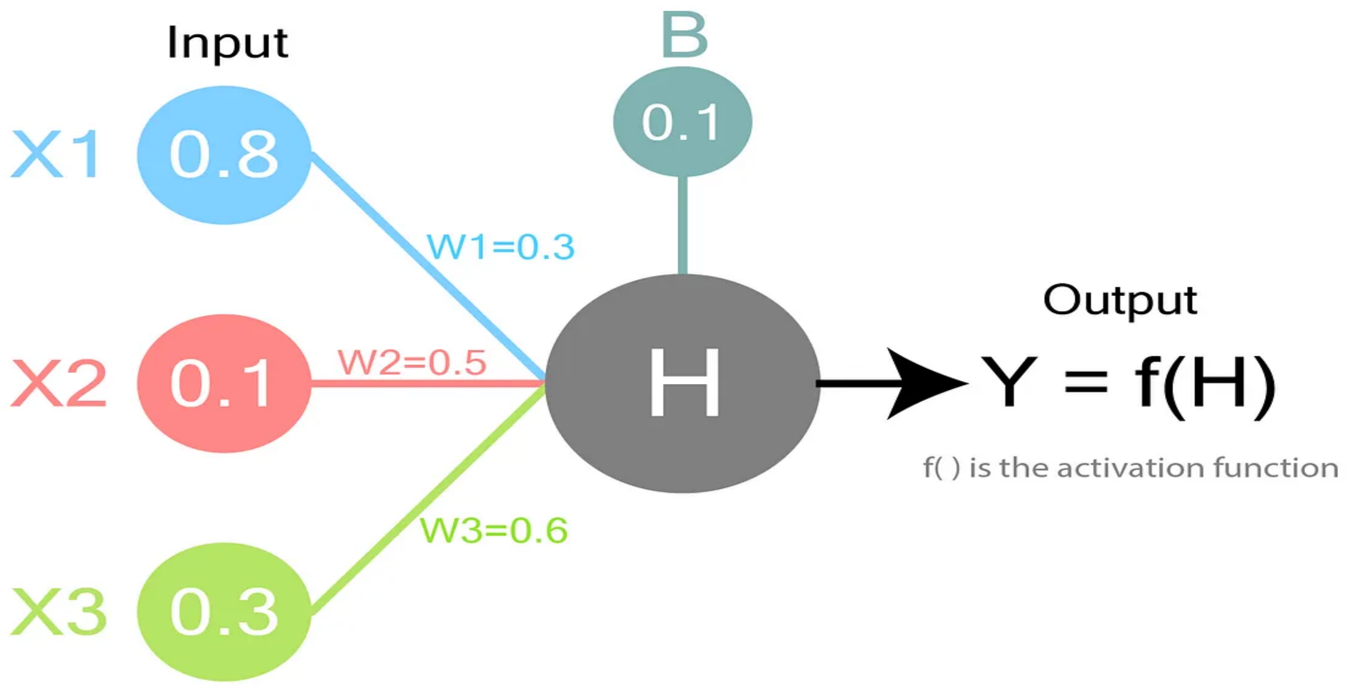
- **Receive Input Signals**: ANNs take in input signals from the dataset, which are typically numerical values.
- **Determine Signal Strength**: The network evaluates the strength of each input signal, often by multiplying each input by a corresponding weight.
- **Calculate Linear Combination:** The weighted inputs are summed to form a total linear combination.
- **Apply Activation Function**: The linear combination is then passed through an activation function to introduce non-linearity into the model. This helps the network learn more complex patterns.
- **Generate Output:** The network generates an output based on the activation function's result.
- **Compute Error:** The difference between the actual output and the desired output (target value) is calculated. This difference is known as the error.
- **Adjust Weights:** In response to the error, the network adjusts the weights till error is below the tolerance level.

# Benefits of Neural Networks

1. Nonlinearity: An artificial neuron can be either linear or nonlinear. However, biological neurons are nonlinear, enabling complex and intelligent decision-making. Consequently, most Artificial Neural Network (ANN) models adopt nonlinearity to emulate these sophisticated biological behaviours.
2. Input-Output Mapping: The learning capability of ANN makes them robust for mapping input objects to class labels, especially in supervised learning paradigms.

4. Adaptability: ANN have a built-in capability to adapt their weights to changes in the surrounding environment.
5. Contextual Information: Neural networks can leverage contextual information to understand data more comprehensively.
6. Fault Tolerance: ANN implemented in hardware form, has the potential to be inherently fault tolerant, or capable of robust computation irrespective of damage of some neurons.
7. VLSI Implementability: The massively parallel nature of a neural network makes it ponentially fast for the computation of certain task which can be implementable in IC chip.

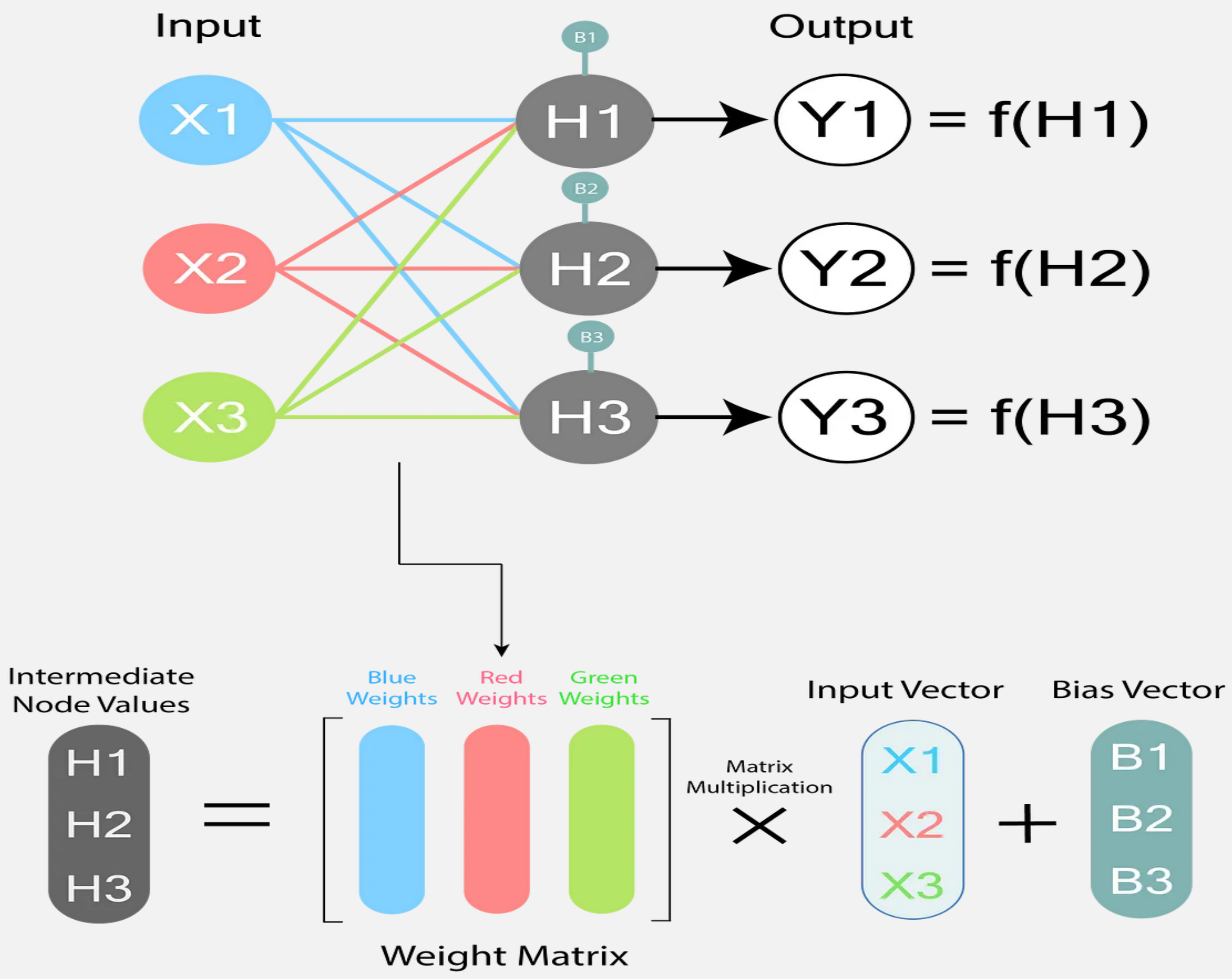
# Single Neuron Functionality

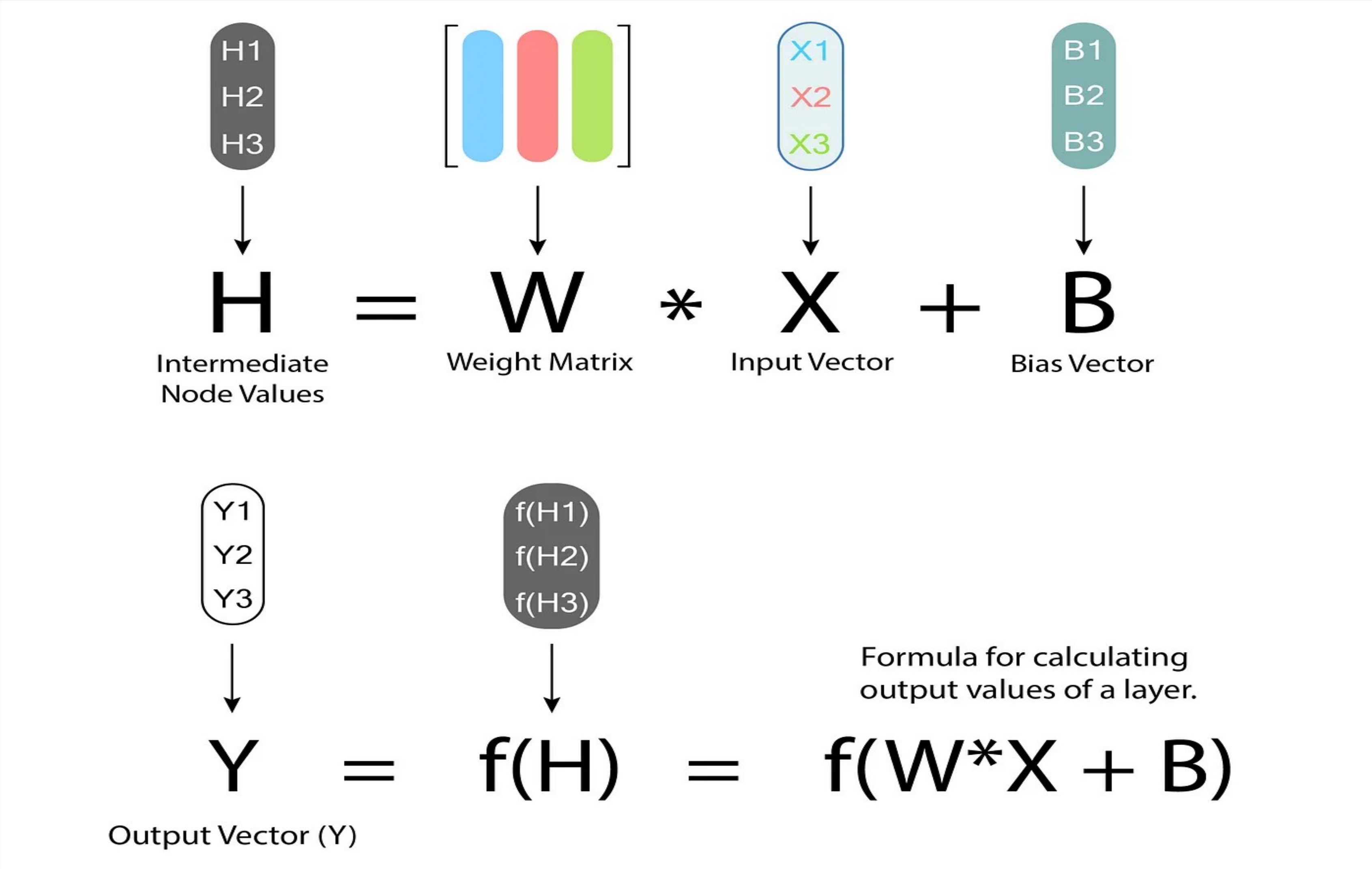



$$H = X1*W1 + X2*W2 + X3*W2 + B$$

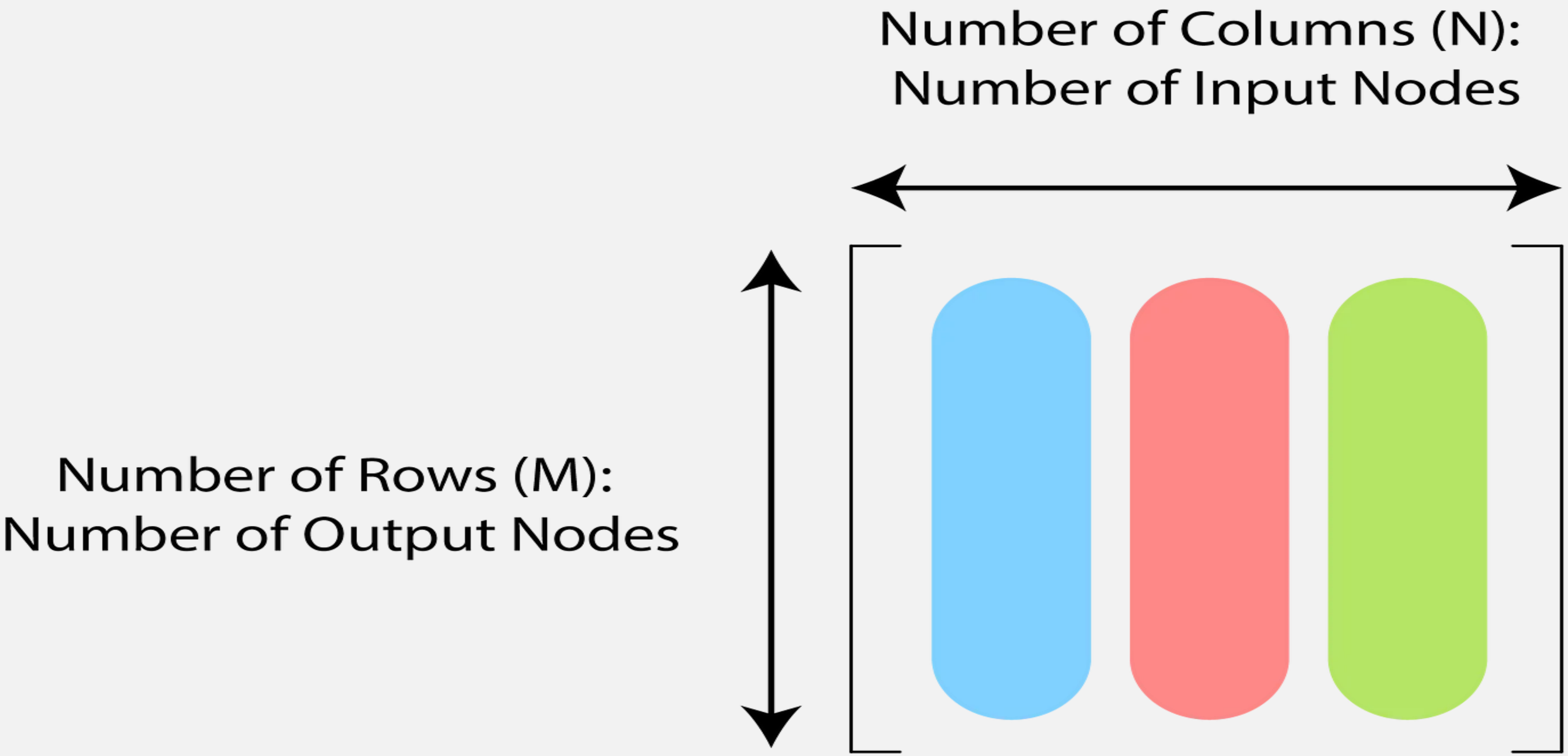
$$Y = f(H)$$

# Multiple neurons with multiple outputs in the output layer



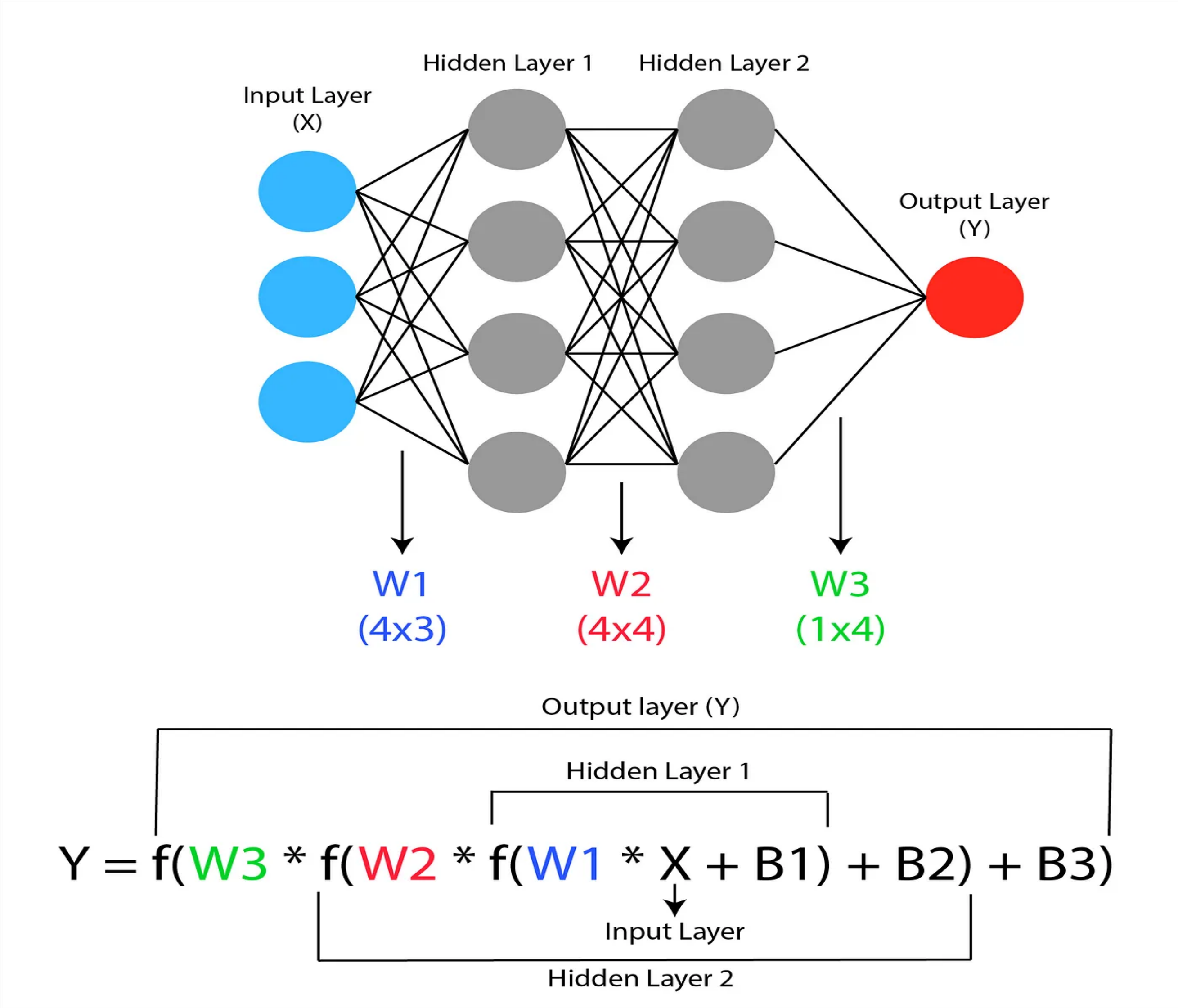

H1
H2
H3
X1
X2
X3
B1
B2
B3
H = W * X + B
Intermediate Node Values
Weight Matrix
Input Vector
Bias Vector
Y1
Y2
Y3
f(H1)
f(H2)
f(H3)
Formula for calculating output values of a layer.
Y = f(H) = f(W*X + B)
Output Vector (Y)

# M*N weight matrix: Interconnections between output layer and input layer



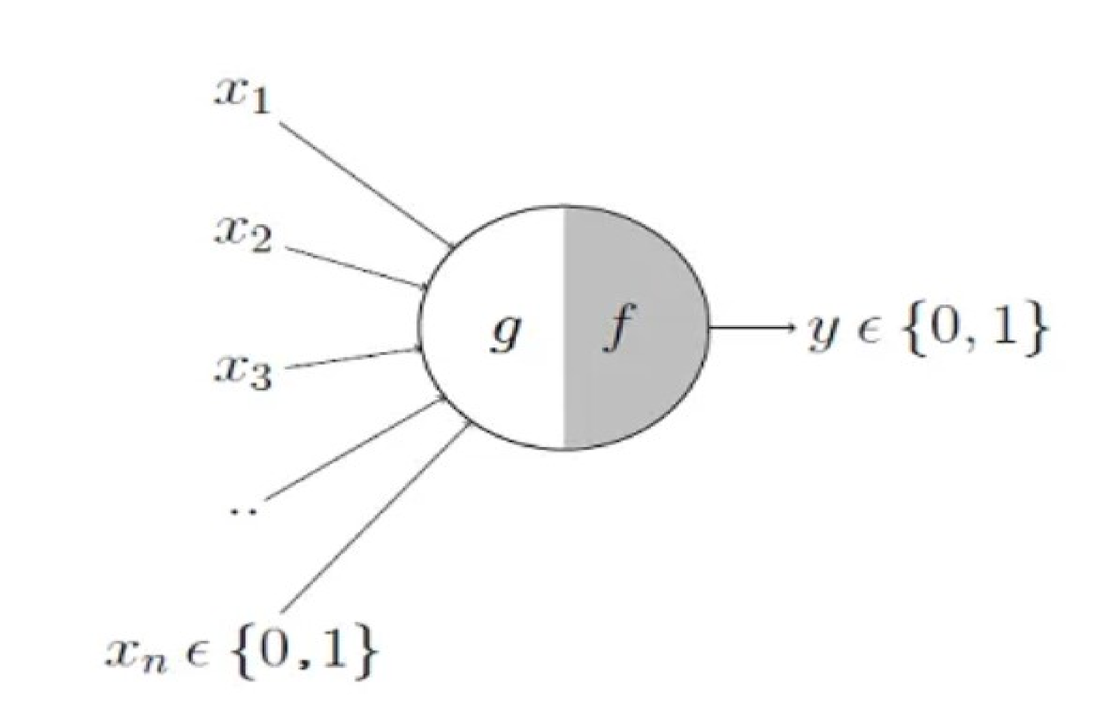

Weight Matrix (MxN)

# Multi-layers neural network



# McCulloch-Pitts Neuron

- The first computational model of a neuron was proposed by Warren MuCulloch (neuroscientist) and Walter Pitts (logician) in 1943.



# McCulloch-Pitts Model

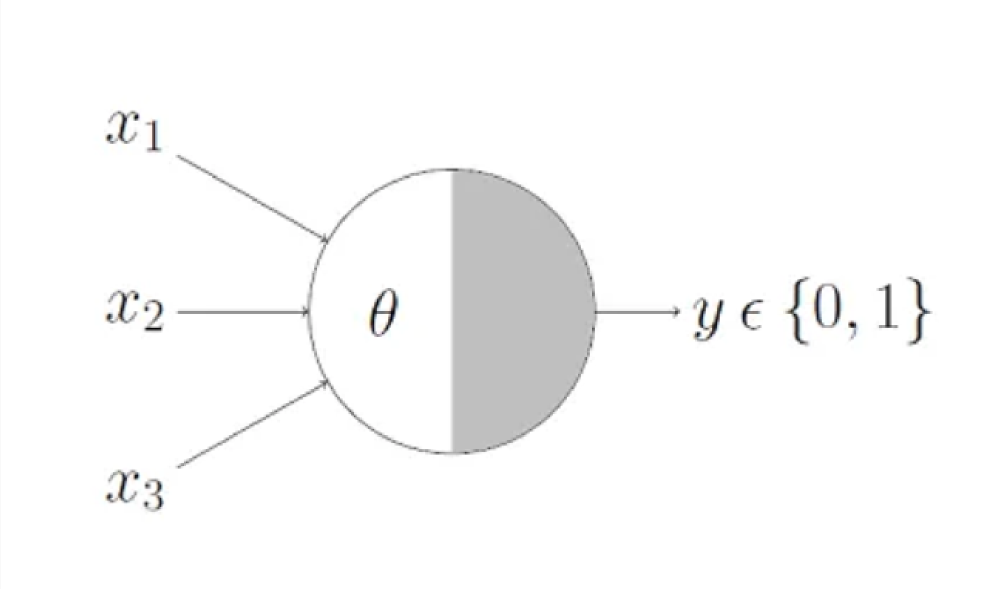
- McCulloch-Pitts (M-P) Model is divided into two parts.
- The first part, g takes an input performs an aggregation and based on the aggregated value the second part, f makes a decision.
- These inputs can either be excitatory or inhibitory.
- Inhibitory inputs are those that have maximum effect on the decision making irrespective of other inputs.
- Excitatory inputs are NOT the ones that will make the neuron fire on their own but they might fire it when combined together.
- Finally, the M-P Neuron fires only when the sum of the inputs is greater than the threshold value ($\theta$).

$$g(x_1, x_2, x_3, ..., x_n) = g(\mathbf{x}) = \sum_{i=1}^{n} x_i$$

$$\begin{aligned} y = f(g(\mathbf{x})) &= 1 \quad if \quad g(\mathbf{x}) \geq \theta \\ &= 0 \quad if \quad g(\mathbf{x}) < \theta \end{aligned}$$

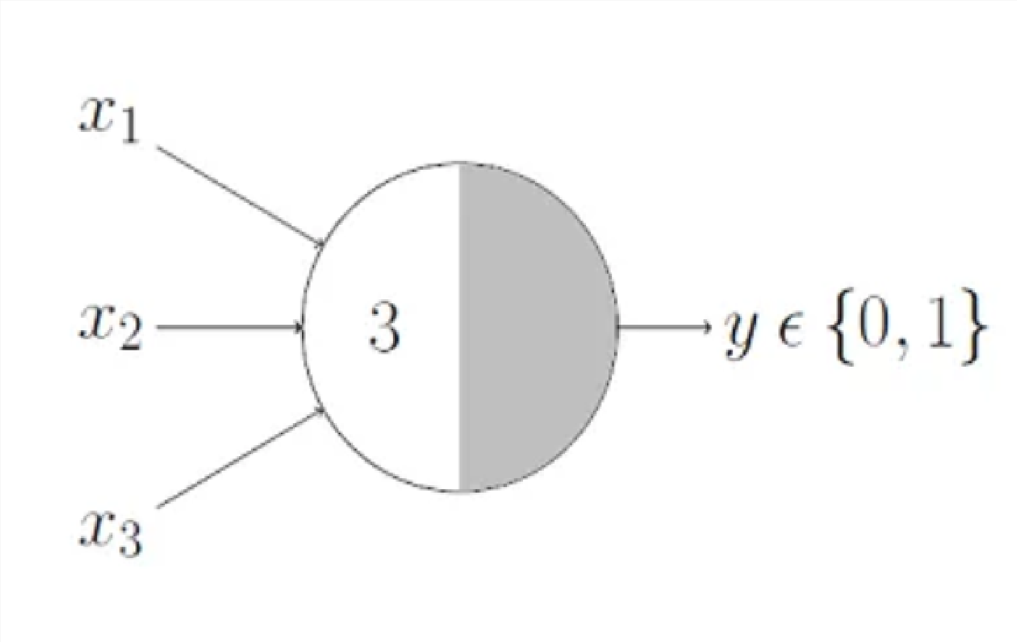
- We can see that g(x) is just doing a sum of the inputs — a simple aggregation.
- Here the function *f* is called threshold activation (θ). Theta decides the binary out; it is 1 when g(x) greater that equal to θ, otherwise output is 0.
- Note that Boolean output decision is taken based on appropriate input variables.

# M-P Neuron: A Concise Representation



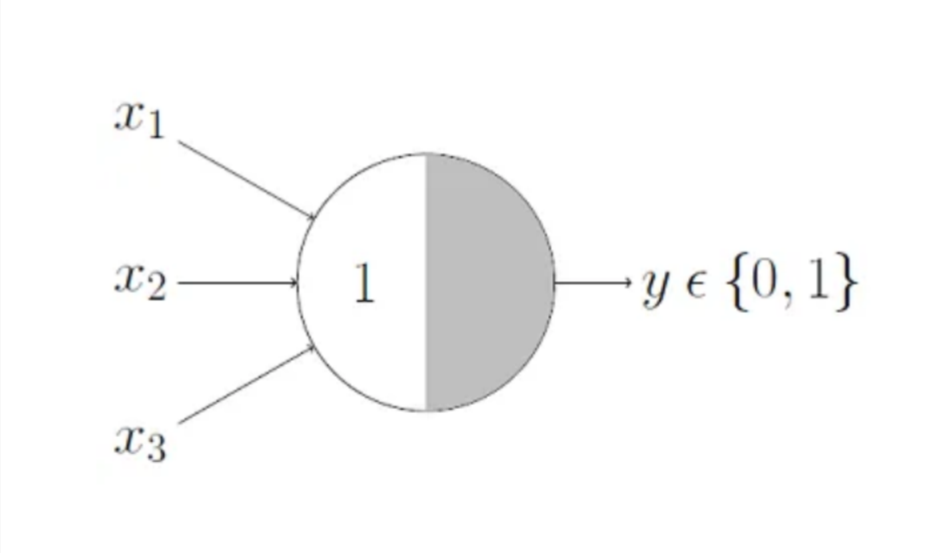

- This representation just denotes that, for the boolean inputs $x_1$, $x_2$ and $x_3$ if the g($x$) i.e., sum ≥ θ, the neuron will fire otherwise, it won't.

# AND Function using M-P Model



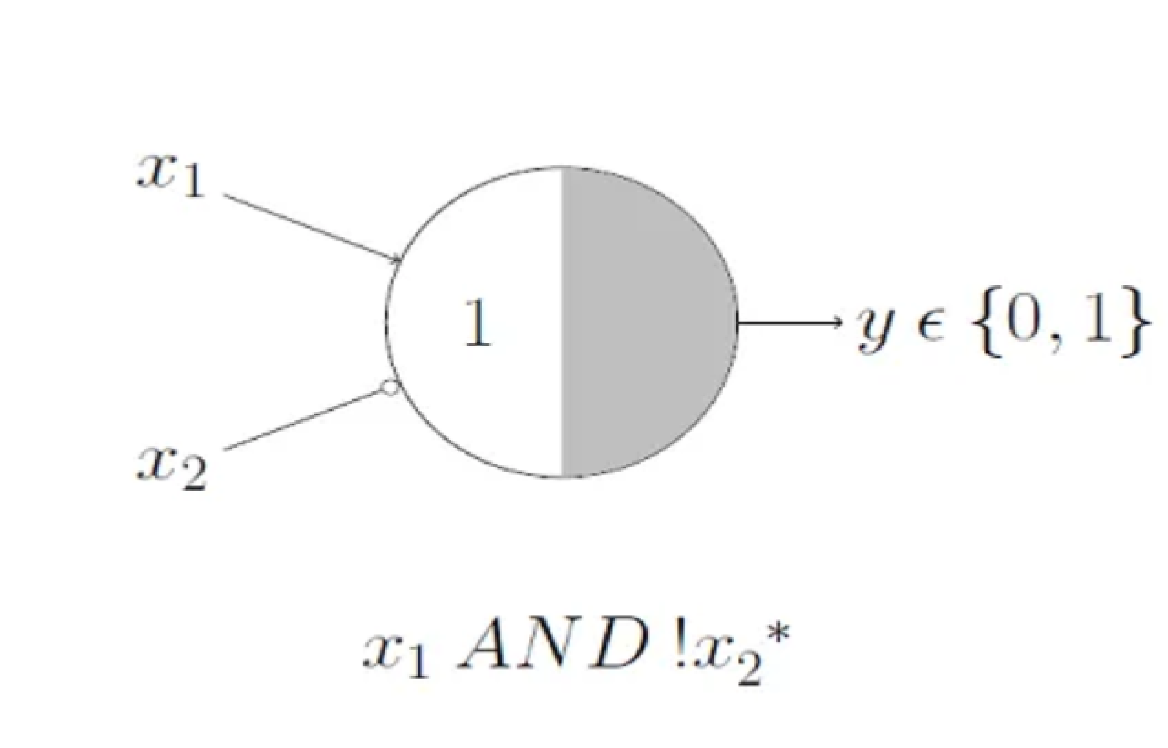

- An AND function neuron would only fire when ALL the inputs are ON i.e., $g(x) \geq 3$ here.

# OR Function using M-P Model



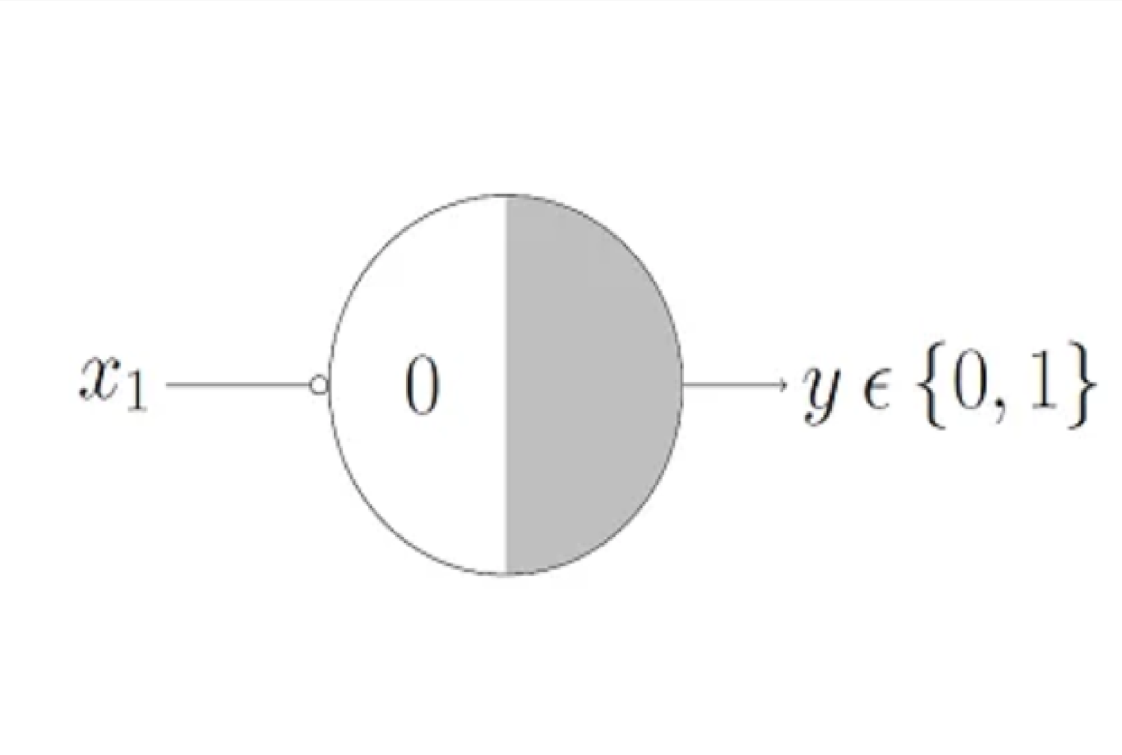

- An OR function neuron would fire if ANY of the inputs is ON i.e., $g(x) \geq 1$ here.

# A Function With An Inhibitory Input




- When, we have an inhibitory input i.e., $x_2$ so whenever $x_2$ is 1, the output will be 0. Keeping that in mind, we know that $x_1$ AND $!x_2$ would output 1 only when $x_1$ is 1 and $x_2$ is 0 so it is obvious that the threshold parameter should be 1.

# NOT Function using M-P Model




- For a NOT neuron, 1 outputs 0 and 0 outputs 1. So, we take the input as an inhibitory input and set the thresholding parameter to 0.

- Can any boolean function be represented using the M-P neuron?

- Ans: M-P neuron is able to conveniently represent the boolean functions which are linearly separable.

# Limitations Of M-P Neuron

- What about non-boolean (say, real) inputs?
- Do we always need to hand code the threshold?
- What about functions which are not linearly separable? Say XOR function.
- To overcome the limitations of the M-P neuron, Frank Rosenblatt, an American psychologist, proposed the classical perception model, the mighty artificial neuron, in 1958.
- It is more generalized computational model than the McCulloch-Pitts neuron.