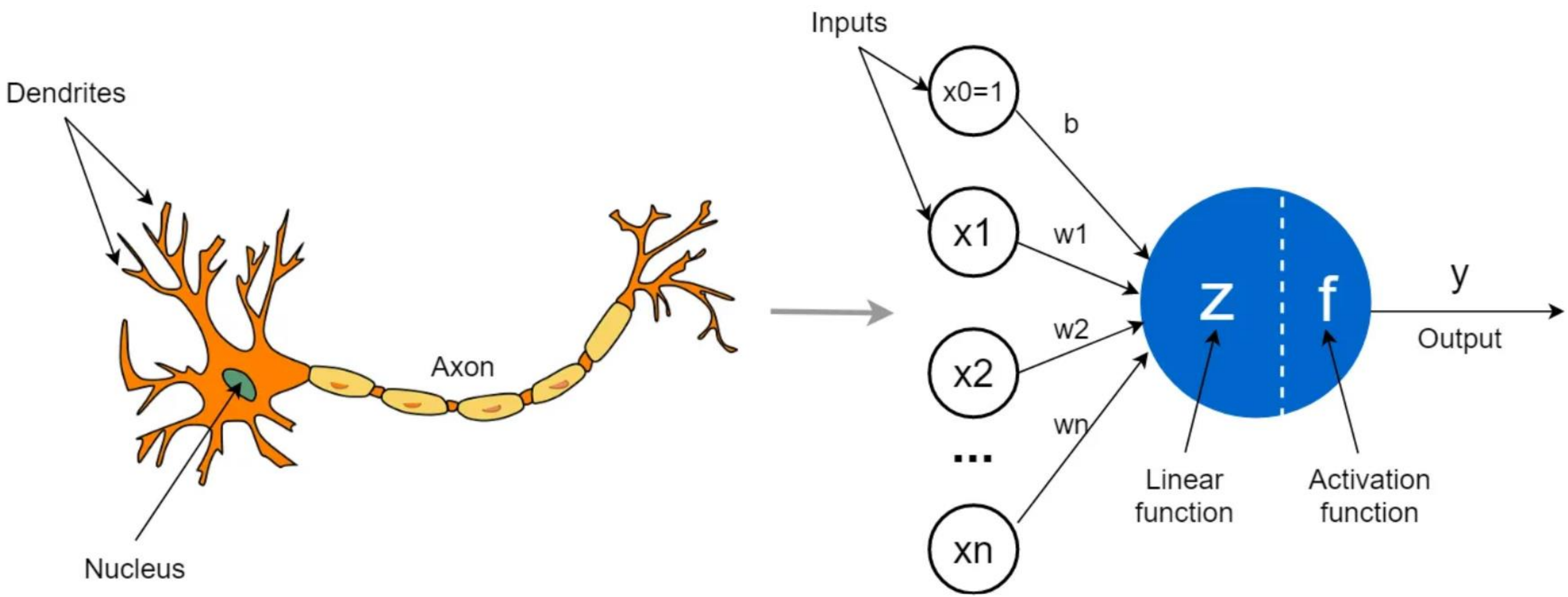
# Module 3: Neurons, Neural Networks, and Linear Discriminants

By
Ms. Sarika Dharangaonkar
Assistant Professor,
KJSCE

# Contents: Module3

# Neurons and Neural Network



# Neurons

- Neuron is a fundamental building block that emulates the behavior of neurons in the human brain.
- It's also referred to as a node or a perceptron.
- Neurons play a crucial role in processing and transmitting information within a neural network, enabling the network to learn patterns and make predictions.

# Hebb’s Rule

- Hebb’s rule says that the changes in the strength of synaptic connections are proportional to the correlation in the firing of the two connecting neurons.
- So if two neurons consistently fire simultaneously, then any connection between them will change in strength, becoming stronger.
- However, if the two neurons never fire simultaneously, the connection between them will die away.
- The idea is that if two neurons both respond to something, then they should be connected.

***"Hebb's Rule states that when two neurons are activated at the same time, the connection between them gets stronger."***

# Hebb’s Rule

- "Neurons" are the cells in our brain that transmit information.
- "Activated at the same time" means when both neurons are firing or sending signals simultaneously.

**Example:**

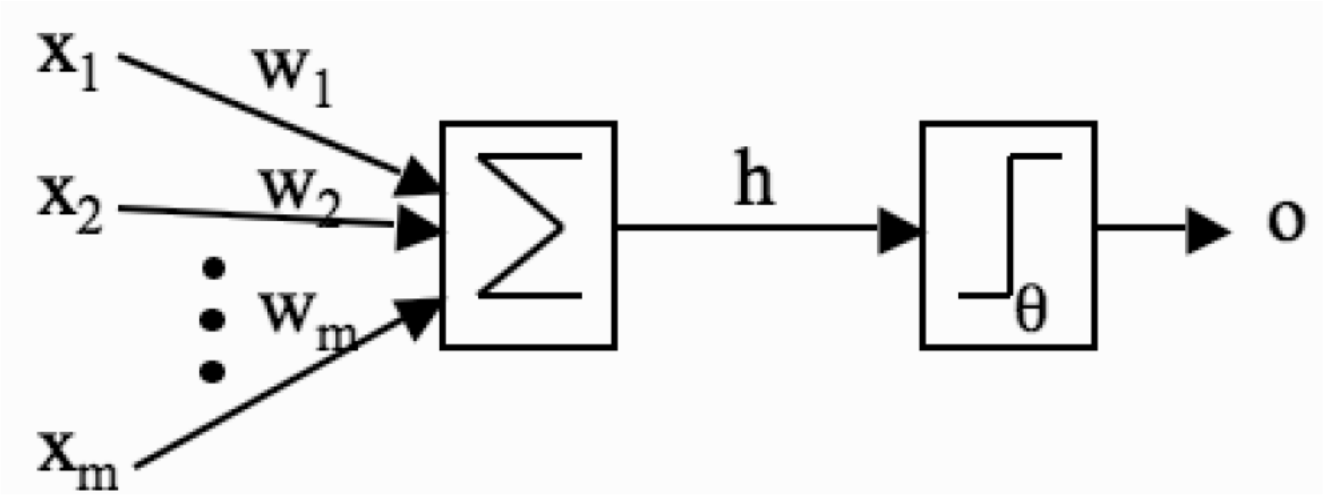
- *"Think of it like remembering people's names. Imagine you meet someone new, and you say their name while looking at their face. When you see that person's face again, your brain automatically remembers their name. This is because your brain strengthens the connection between their face and their name when you say them together."*

# Hebb’s Rule

**Example:**

*Suppose that you have a neuron somewhere that recognizes your grandmother. Now if your grandmother always gives you a chocolate bar when she comes to visit, then some neurons, which are happy because you like the taste of chocolate, will also be stimulated. Since these neurons fire at the same time, they will be connected together, and the connection will get stronger over time. So eventually, the sight of your grandmother, even in a photo, will be enough to make you think of chocolate.*

# McCulloch and Pitts Neurons

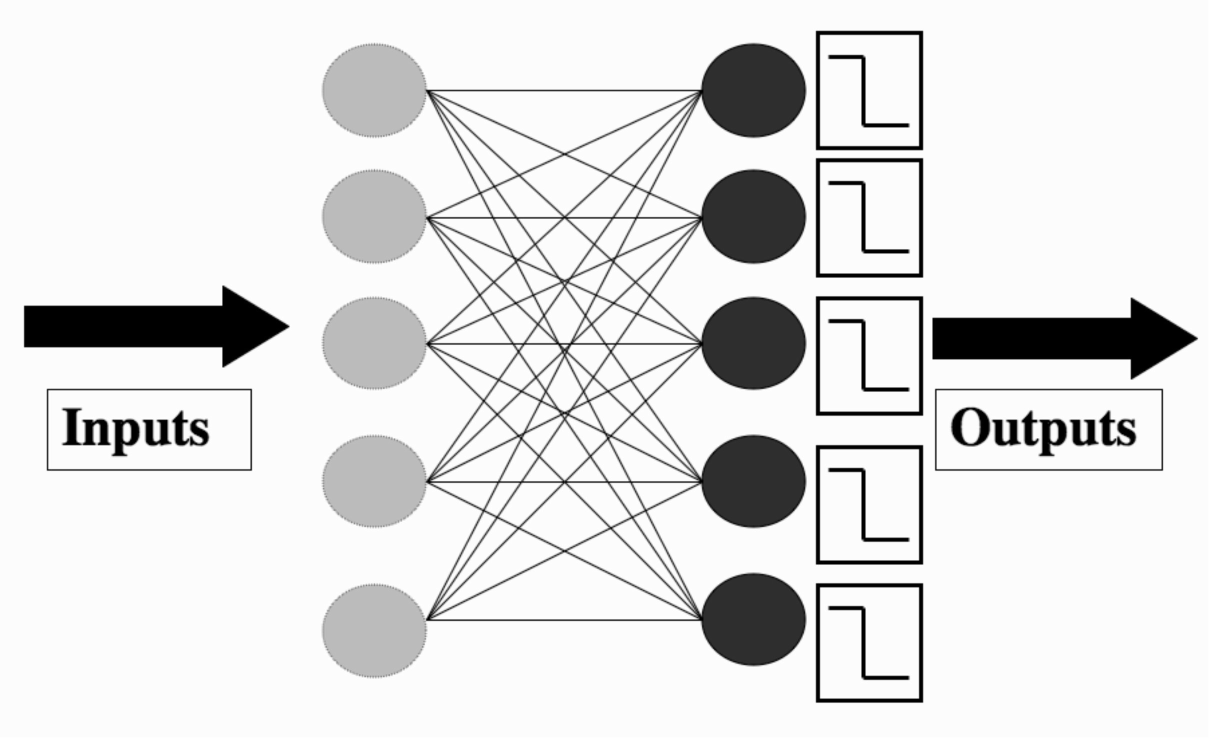



A picture of McCulloch and Pitts' mathematical model of a neuron. The inputs *xi* are multiplied by the weights *wi*, and the neurons sum their values. If this sum is greater than the threshold then the neuron fires; otherwise it does not.

# Limitations

- **Binary Activation:** The model uses a simple on/off activation, unlike the gradual activation in real neurons.
- **No Weights:** It doesn't consider the strengths of connections between inputs and the neuron.
- **Fixed Threshold:** The threshold for activation is fixed, not adaptable.
- **Limited Complexity:** It can only handle linear decision boundaries, not complex patterns.
- **Can't Learn:** The model can't learn from data; real neural networks can.
- **No Adaptability:** It can't adapt to changes or new experiences.
- **Oversimplified Biology:** It doesn't fully represent how real neurons work.
- **Single Output:** It produces only one output; real problems often need multiple outputs.
- **No Hidden Layers:** It lacks the concept of hidden layers for complex tasks.
- **Limited Generalization:** It struggles to apply learning to new, unseen data.

# The Perceptron




- The Perceptron is nothing more than a collection of McCulloch and Pitts neurons together with a set of inputs and some weights to fasten the inputs to the neurons.
- The Perceptron network, consisting of a set of input nodes (left) connected to McCulloch and Pitts neurons using weighted connections.
- The neurons in the Perceptron are completely independent of each other.

# The Perceptron

**Perceptron Structure:**

A perceptron consists of three main components:

- Inputs ($x_1, x_2, ..., x_n$): These represent the features of the input data.
- Weights ($w_1, w_2, ..., w_n$): Each input is associated with a weight that represents its importance.
- Activation Function: This function determines whether the neuron should fire or not based on the weighted sum of inputs.

**Perceptron Operation:**

The operation of a perceptron involves these steps:

- Multiply each input by its corresponding weight.
- Sum up the weighted inputs.
- Pass the sum through an activation function.

# The Perceptron

**Activation Function:**

The activation function is typically a step function or a similar function that converts the sum of weighted inputs into a binary output (0 or 1), representing a decision.

**Use in Binary Classification:**

Perceptron's are often used for binary classification tasks, where the output represents a class prediction (e.g., 0 or 1, yes or no).

**Learning and Adaptation:**

Perceptron's can learn from data using a learning algorithm like the perceptron learning rule. This algorithm adjusts the weights based on misclassifications to improve the accuracy of predictions.

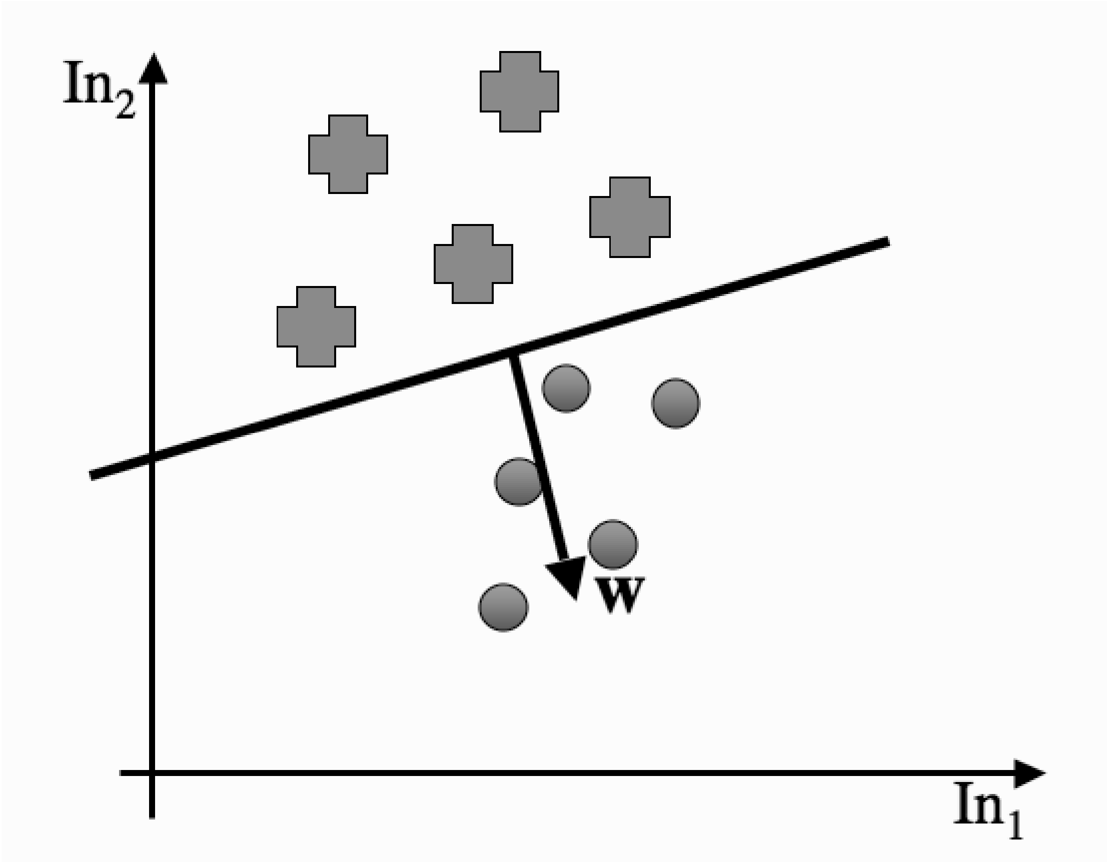
# The Perceptron

**Neural Network:**

A perceptron is a single neuron, and multiple perceptron's are typically combined in layers to create a neural network capable of solving more complex problems.

a perceptron is a fundamental unit in neural networks used for binary classification. It takes inputs, applies weights, calculates a weighted sum, and produces an output through an activation function.

# Linear Seperability




- Linear seperability refers to the property of data points being separable by a straight line (or hyperplane in higher dimensions).
- In other words, if you can draw a line (or hyperplane) on a graph that completely separates data points of one class from those of another class, then the data is said to be linearly separable.

# Linear Seperability

- Linear seperability is most commonly associated with binary classification problems, where you're trying to separate data points into two classes (e.g., positive and negative, yes and no).
- Imagine you have a scatter plot of data points, where each point has two features (variables) plotted on the axes. If you can draw a straight line that perfectly divides the points of one class from the points of the other class, the data is linearly separable.
- Mathematically, linear seperability means that there exists a set of weights (coefficients) and a bias term that, when used in a linear equation, can correctly classify all the data points into their respective classes.

# Linear Seperability

## Linear Decision Boundary:

- The dividing line (or hyperplane) is called a linear decision boundary. It's determined by the weights and bias in the linear equation.

## Non-Linear Seperability:

- If it's not possible to draw a straight line to separate the data points, the data is not linearly separable. In this case, you might need more complex models or techniques to classify the data accurately.

## Example:

- Consider the following scenario: you have data points representing exam scores of students, and you want to classify them into two classes: pass and fail. If you can draw a line that clearly separates the students who passed from those who failed based on their scores, then the data is linearly separable.

# Linear Seperability

- **Handling Non-Linear Data:**
  - For data that is not linearly separable, more advanced techniques, such as kernel methods, neural networks with non-linear activation functions, and decision trees, can be used to capture complex patterns and classify the data accurately.

# Linear Seperability

- **Linear Equation for a Decision Boundary:**
  - In a binary classification problem, you're trying to classify data points into two classes: positive (1) and negative (0). A linear equation for a decision boundary can be written as:

$$w_0 + w_1 * x_1 + w_2 * x_2 = 0$$

Where:

  - $w_0$, $w_1$, and $w_2$ are weights or coefficients.
  - $(x_1, x_2)$ represents the features of a data point.

# Linear Seperability

- **Determining Classes:**
  - Given the equation, you can determine the class of a data point based on where it lies with respect to the decision boundary:
  - If $w_0 + w_1 * x_1 + w_2 * x_2$ is positive or zero, the data point is classified as positive (1).
  - If $w_0 + w_1 * x_1 + w_2 * x_2$ is negative, the data point is classified as negative (0).
- **Visual Representation:**
  - In a 2D space, the decision boundary is a straight line. If the data points can be perfectly separated by this line, the data is linearly separable.

# Linear Seperability

- **Linearly Separable Data:**
  - For linearly separable data, there exists a set of weights ($w_0$, $w_1$, $w_2$) that allows you to draw a line that separates the positive and negative classes without any misclassifications.
- **Example:**
  - Let's say you have data points in a 2D space and want to classify them as either blue (positive) or red (negative). If you find weights $w_0$, $w_1$, and $w_2$ that satisfy the linear equation for a decision boundary and the equation perfectly separates the blue and red points, then your data is linearly separable.
- In higher dimensions, the concept remains the same, but the decision boundary becomes a hyperplane. The equation generalizes to:

$$w_0 + w_1 * x_1 + w_2 * x_2 + \ldots + w_n * x_n = 0$$

Where ($x_1$, $x_2$, ..., $x_n$) represents the features of a data point in n-dimensional space.

Thank You