**Intro to Neural Networks**

**What is a Neural Network?**  
 A neural network is a computational model inspired by the human brain. It’s designed to recognize patterns and make decisions based on input data. Neural networks are the backbone of **deep learning**.

**Key idea:**

* They take inputs, process them through layers of neurons, and produce outputs.
* Each neuron applies a weight to inputs, sums them, and passes them through an activation function.

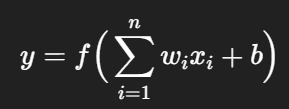
## **Why use Neural Networks?**

* Can model non-linear relationships (simple models like linear regression cannot).
* Can handle complex tasks: image recognition, speech recognition, NLP, etc.

## **Basic structure of a neural network:**

1. **Input Layer** – receives data (features).
2. **Hidden Layer(s)** – process data using neurons.
3. **Output Layer** – produces the prediction/result.

**Mathematical view:**  
 For a neuron:



Where:

* ***xi*** = input features
* ***wi*** = weights
* **b** = bias
* **f** = activation function (e.g., step, sigmoid, ReLU)
* **y** = output

## **Basic Components of Artificial Neural Networks:**

* **Neurons** = Basic computing units of a neural network that process inputs to produce an output.
* **Layers** = Groups of neurons arranged in levels that transform data step by step.
* **Connections** = Links between neurons that carry weighted signals from one layer to another.
* **Activation Function**= A rule that decides whether a neuron should activate, adding non-linearity to the model.

**The Perceptron**

The Perceptron is the simplest type of neural network. It was introduced by Frank Rosenblatt in 1958.

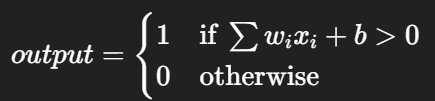
**What is it used for?**

Binary classification problems: output is either 0 or 1.

Example: Is an email spam (1) or not spam (0)?

**Structure of a Perceptron:**

* Inputs ***x******1,x******2,...,xn***
* Weights ***w1, w******2,...,wn***
* Bias ***b***
* Step activation function:



**How it learns:**

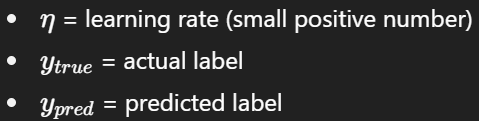
* Using the **Perceptron Learning Algorithm**:

1. Initialize weights randomly.
2. For each training sample:

* Compute the output.
* Compare with the actual label.
* Update weights if prediction is wrong:



Where:



**Limitations of a Perceptron:**

* Can only solve linearly separable problems.
* Cannot solve problems like XOR.

**Example Use Case:**

* Predict if a student passes or fails based on study hours and sleep hours (binary output).

**Summary:**

* Neural networks are inspired by the human brain and are used for pattern recognition.
* Perceptron = simplest neural network for binary classification.
* Learns by adjusting weights based on errors.
* Limitation: only works for linearly separable problems.