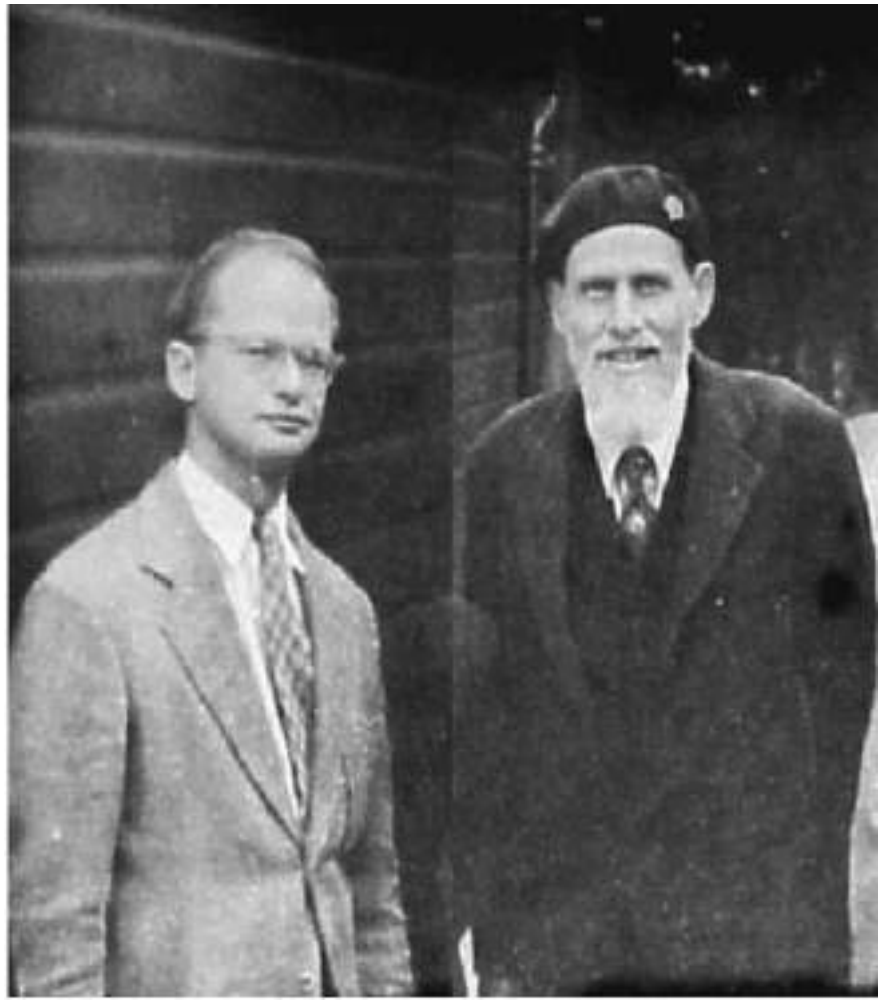
Introduction Neural Network

Walter Pitts

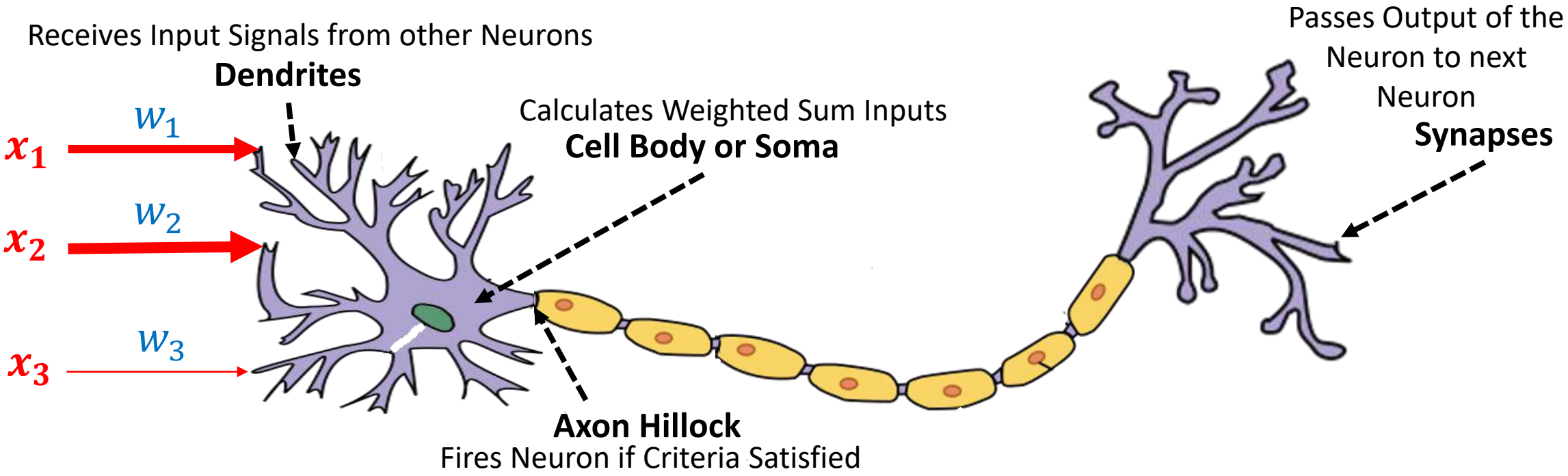


Warren McCulloch

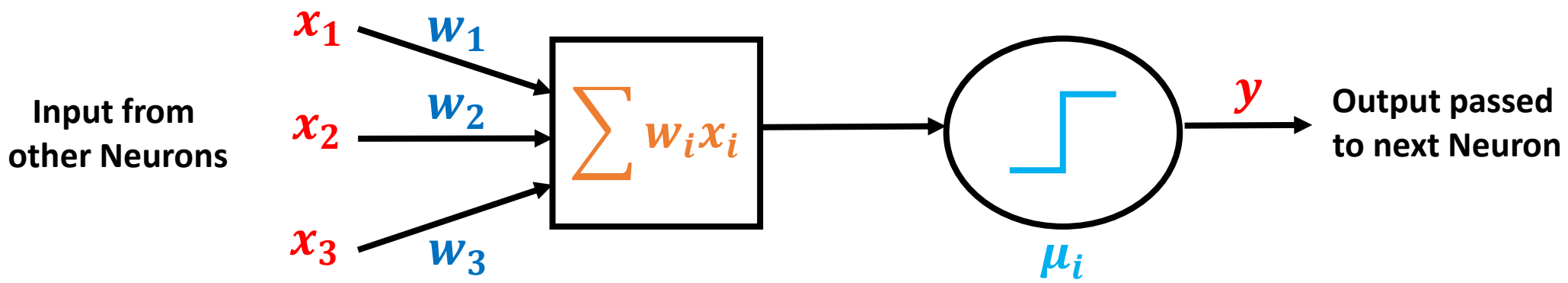
They Proposed the first Mathematical Model for a Neural Network in 1943

Image Source: www.semanticscholar.org

Introduction Neural Network



Structure of Biological **"Neuron"**

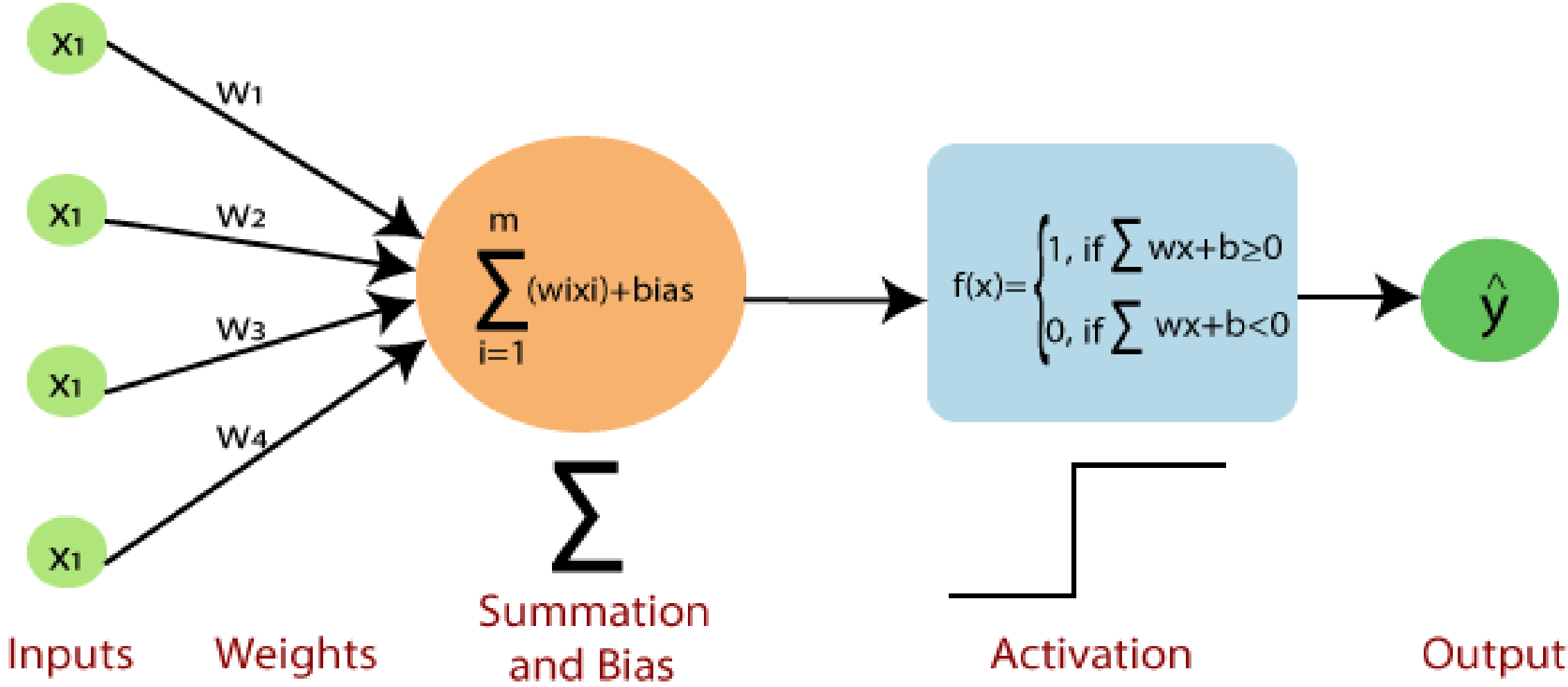


Threshold Logic Unit (TLU): The First Artificial **"Neuron"**

Artificial Neural Network (ANN)

Neural Networks are complex systems with artificial neuron.

Artificial neurons or perceptron consist of:



Introduction Neural Network

Mathematical Representation of TLU

$$y = \begin{cases} 1 & if \sum_{k=1}^{n} w_i x_i \geq T \\ 0 & if \sum_{k=1}^{n} w_i x_i < T \end{cases}$$

where,

It has a binary output $y \in \{1, 0\}$, $y = 1$ indicates that the neuron fires and $y = 0$ that it is at rest.

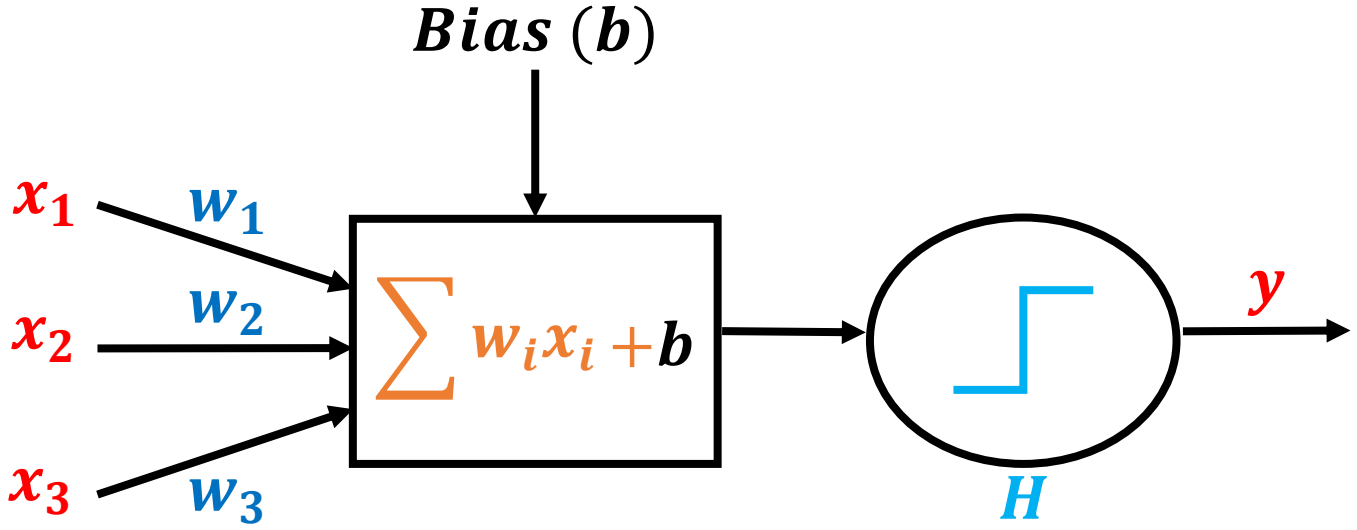
If weighted sum of inputs is greater than threshold value $\boldsymbol{T}$, the neuron fires.

If weighted sum of inputs is less than or equal to threshold value $\boldsymbol{T}$, the neuron stays at rest.

Introduction Neural Network

Frank Rosenblatt

He Proposed the concept of Perceptron based on TLU in 1958



Artificial **"Neuron"** used by "**Perceptron**"

$$\Delta w = \eta \times d \times x$$

where,

x Input, η Learning rate, and

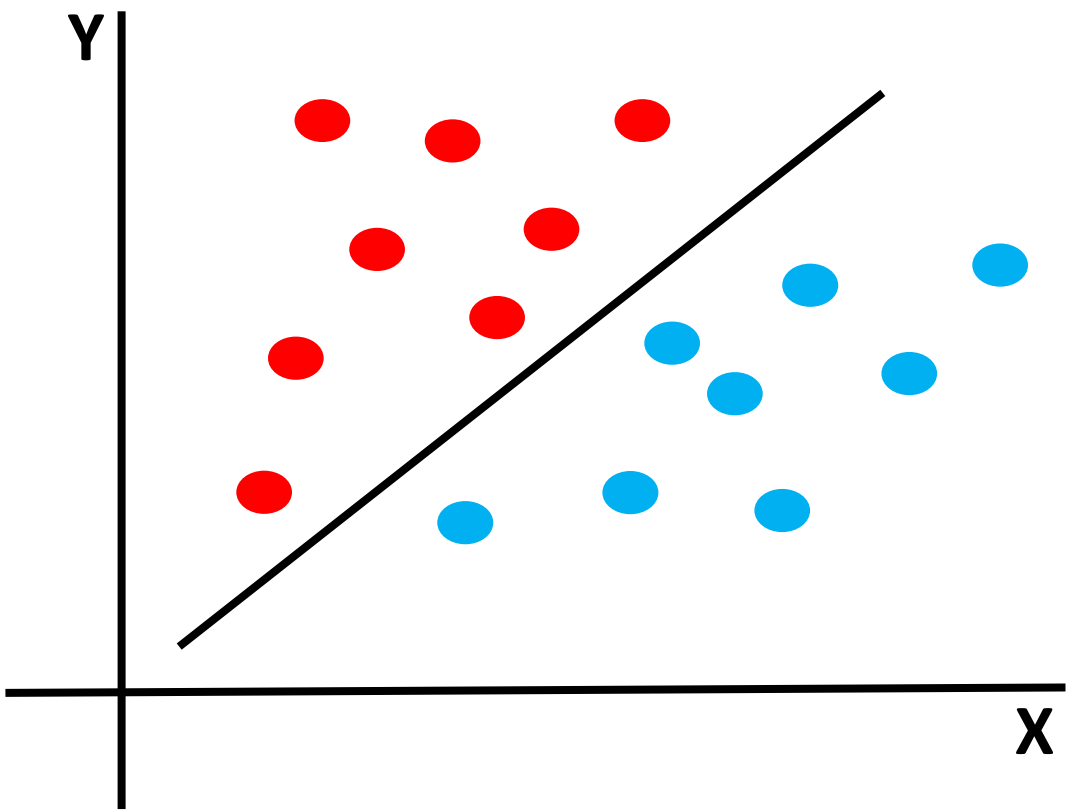
d = Predicted Output − Desired Output

$$y = \begin{cases} 1 & if \sum_{k=1}^{n} w_i x_i + b \geq H \\ 0 & if \sum_{k=1}^{n} w_i x_i + b < H \end{cases}$$

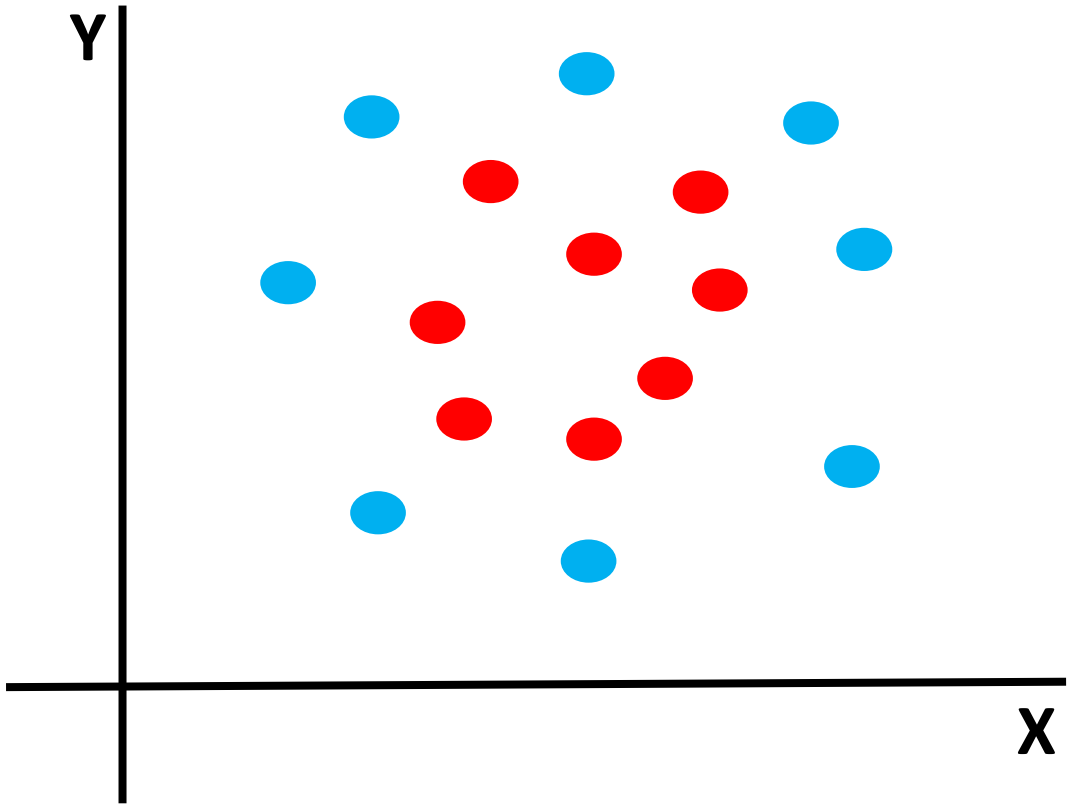
Image Source: www.semanticscholar.org

Introduction Neural Network

Marvin Minsky and Seymour Papert proposed a Book “Perceptrons: An Introduction to Computational Geometry” in 1969



Linearly Separable Data Points



Linearly Non-Separable Data Points

Must Visit https://playground.tensorflow.org

Artificial Neural Network (ANN)

- Neural networks are comprised of layers of neurons.

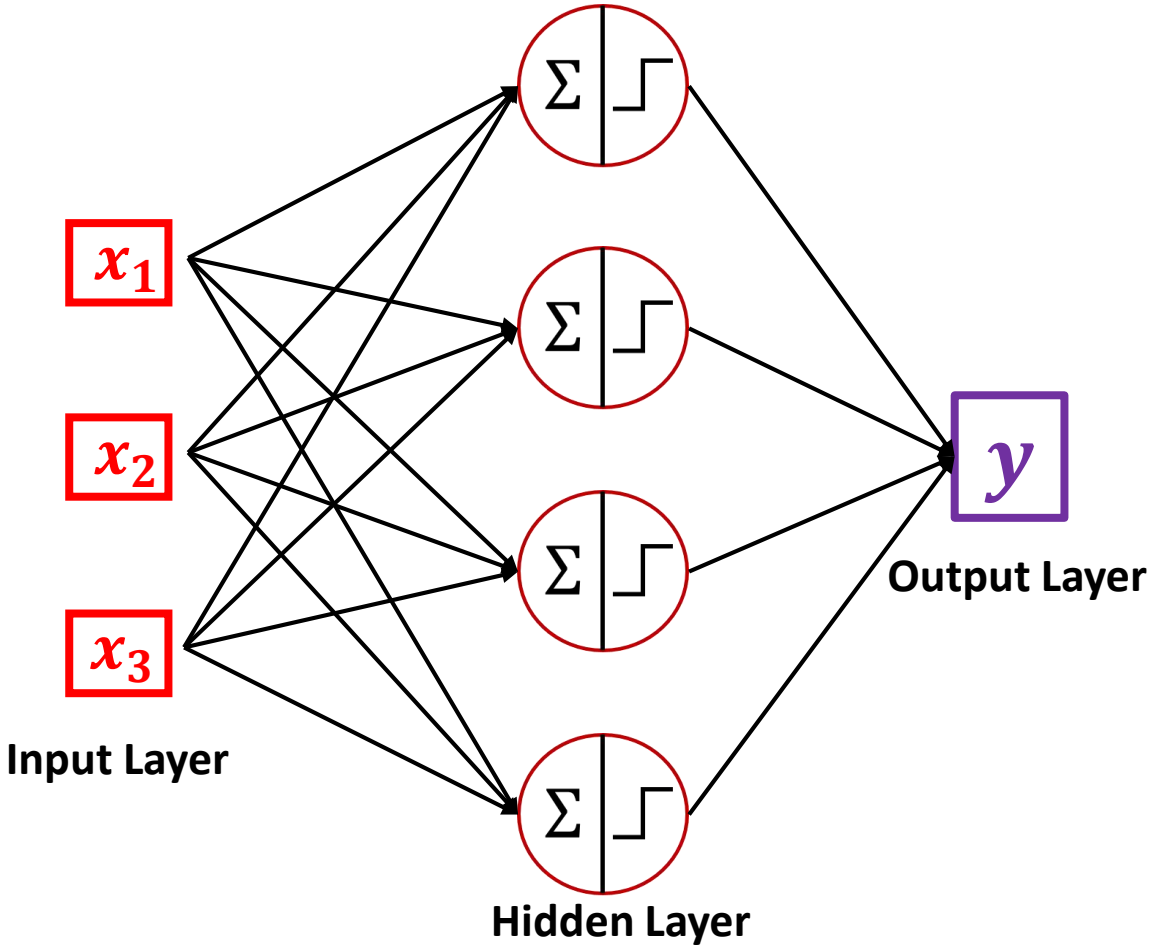
These layers consist of the following:

- Input layer
- Multiple hidden layers
- Output layer

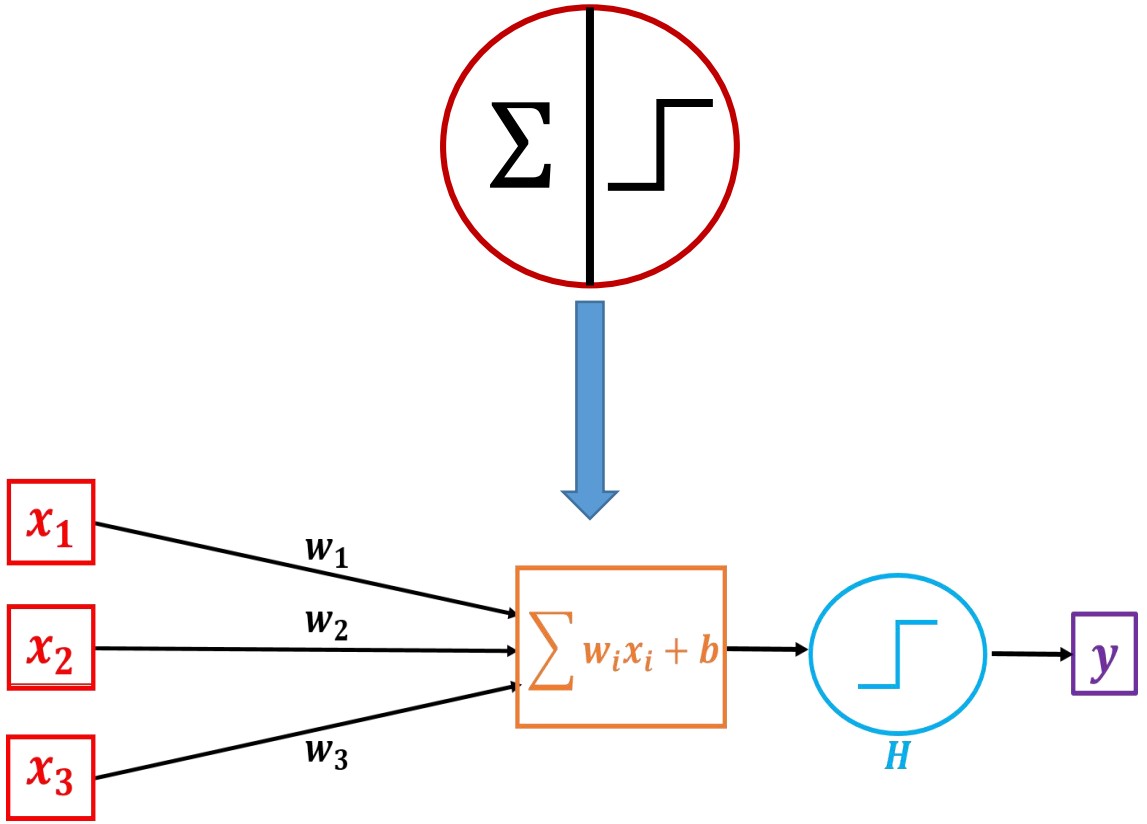
- The input layer receives data represented by a numeric value. Hidden layers perform the most computations required by the network. Finally, the output layer predicts the output.

Introduction Neural Network

David Rumelhart, Geoffrey Hinton and Ronald Williams published "**Learning Internal Representations by Error Propagation**" in 1986.

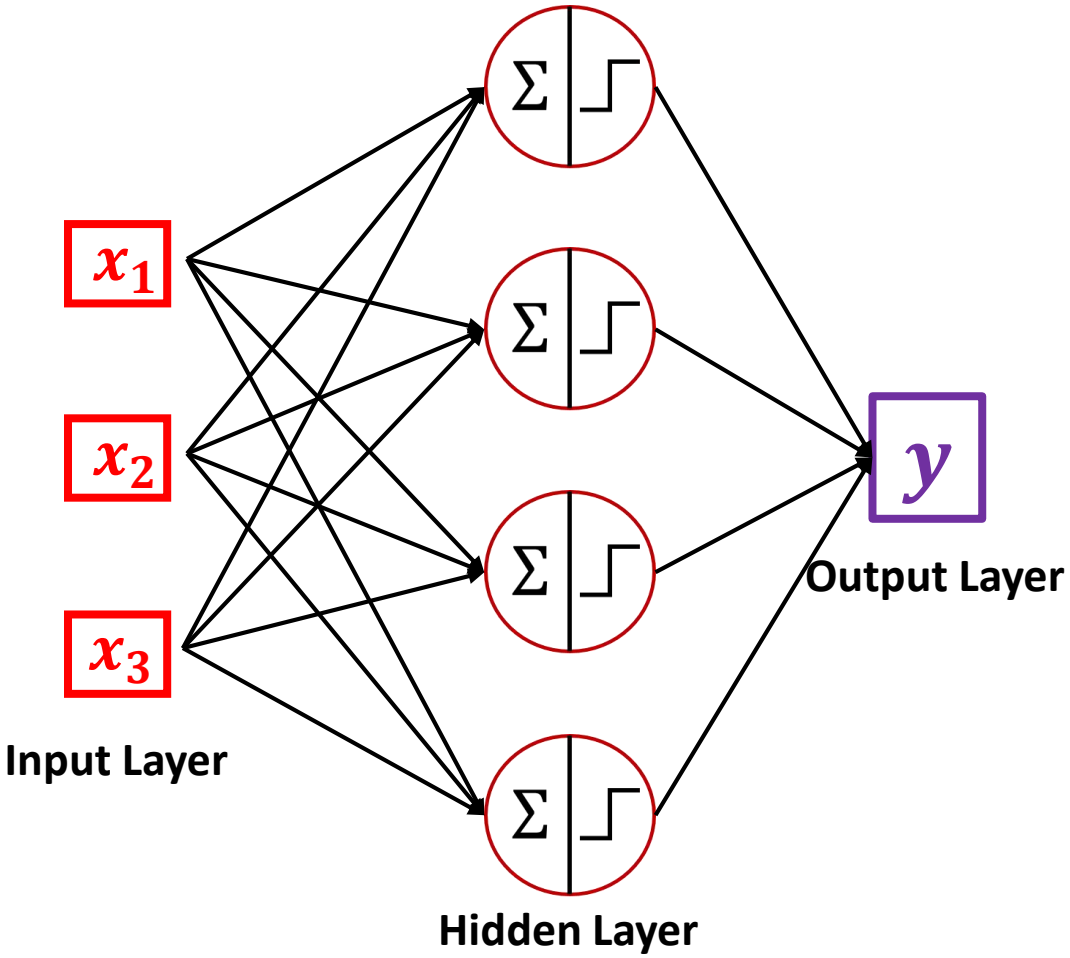


MLP with One Hidden Layer

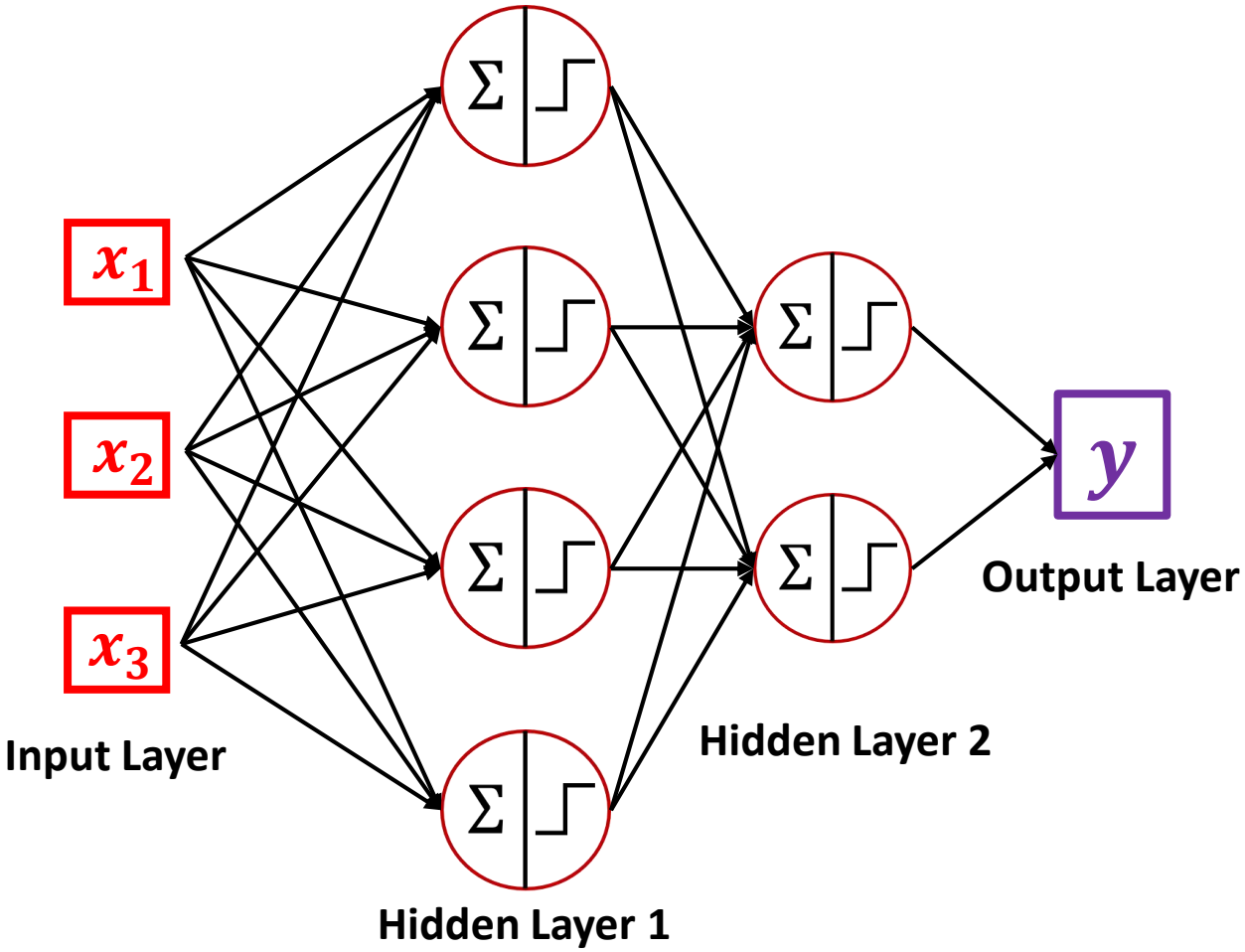


Single Layer Perceptron

Introduction Neural Network



MLP with One Hidden Layer

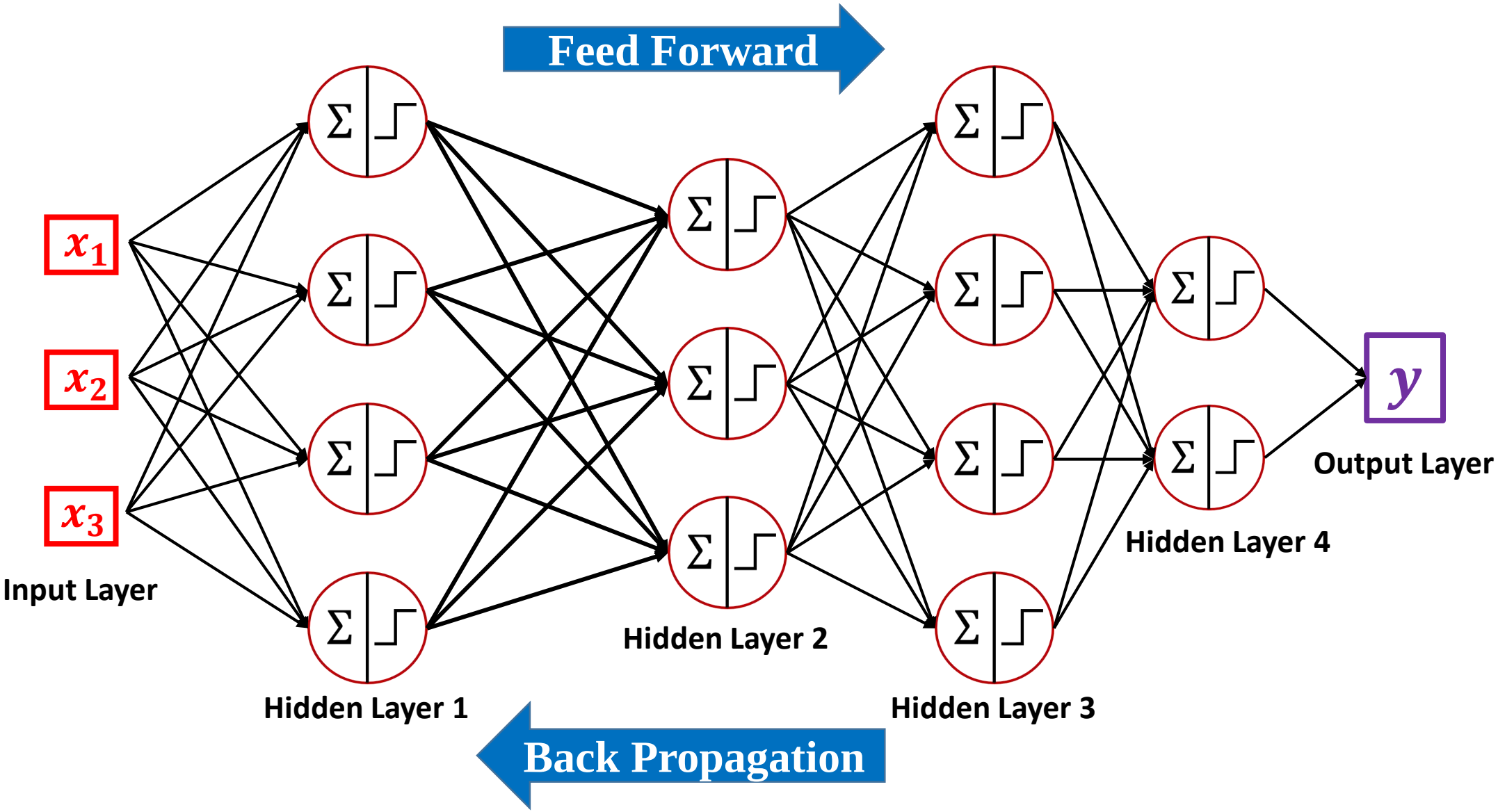


MLP with Two Hidden Layers

Must Visit https://playground.tensorflow.org

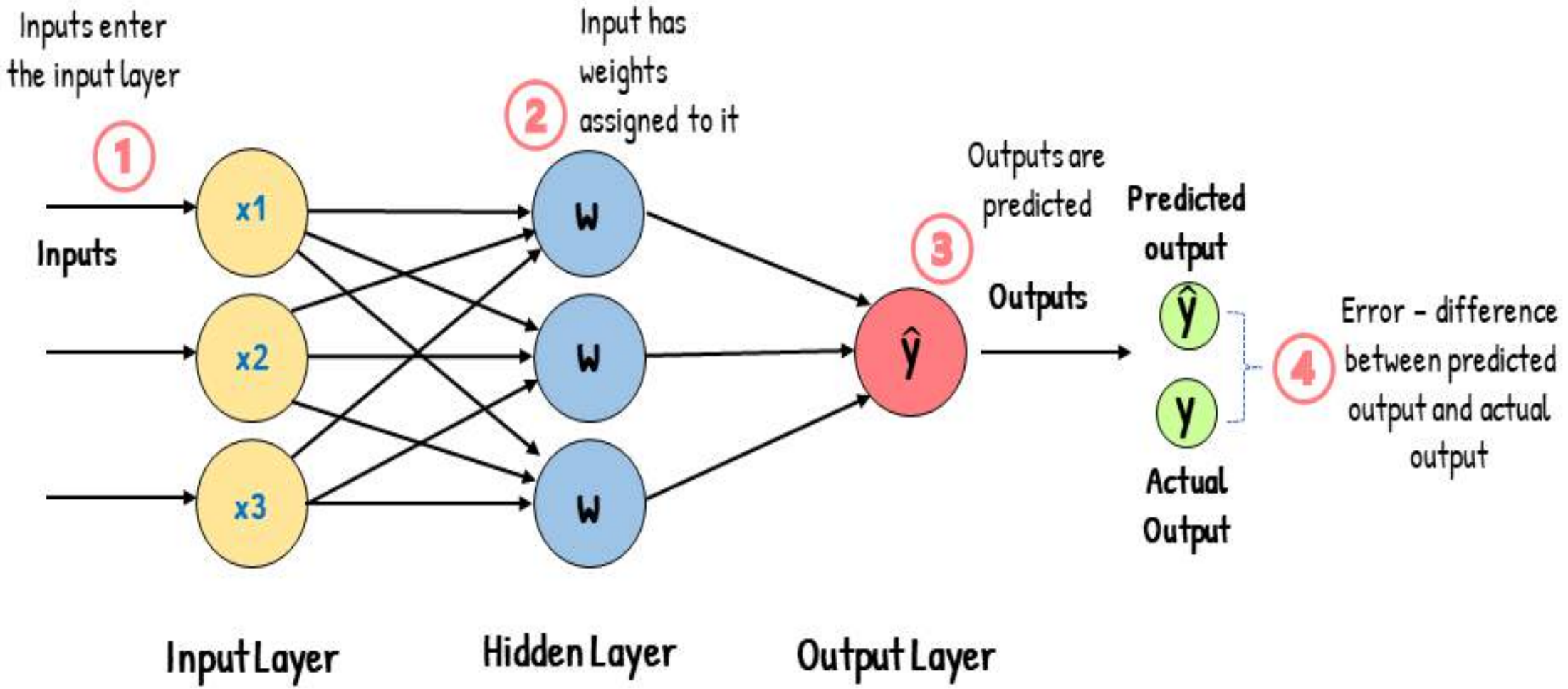
Introduction Neural Network

David Rumelhart, Geoffrey Hinton and Ronald Williams published "**Learning Internal Representations by Error Propagation**" in 1986.



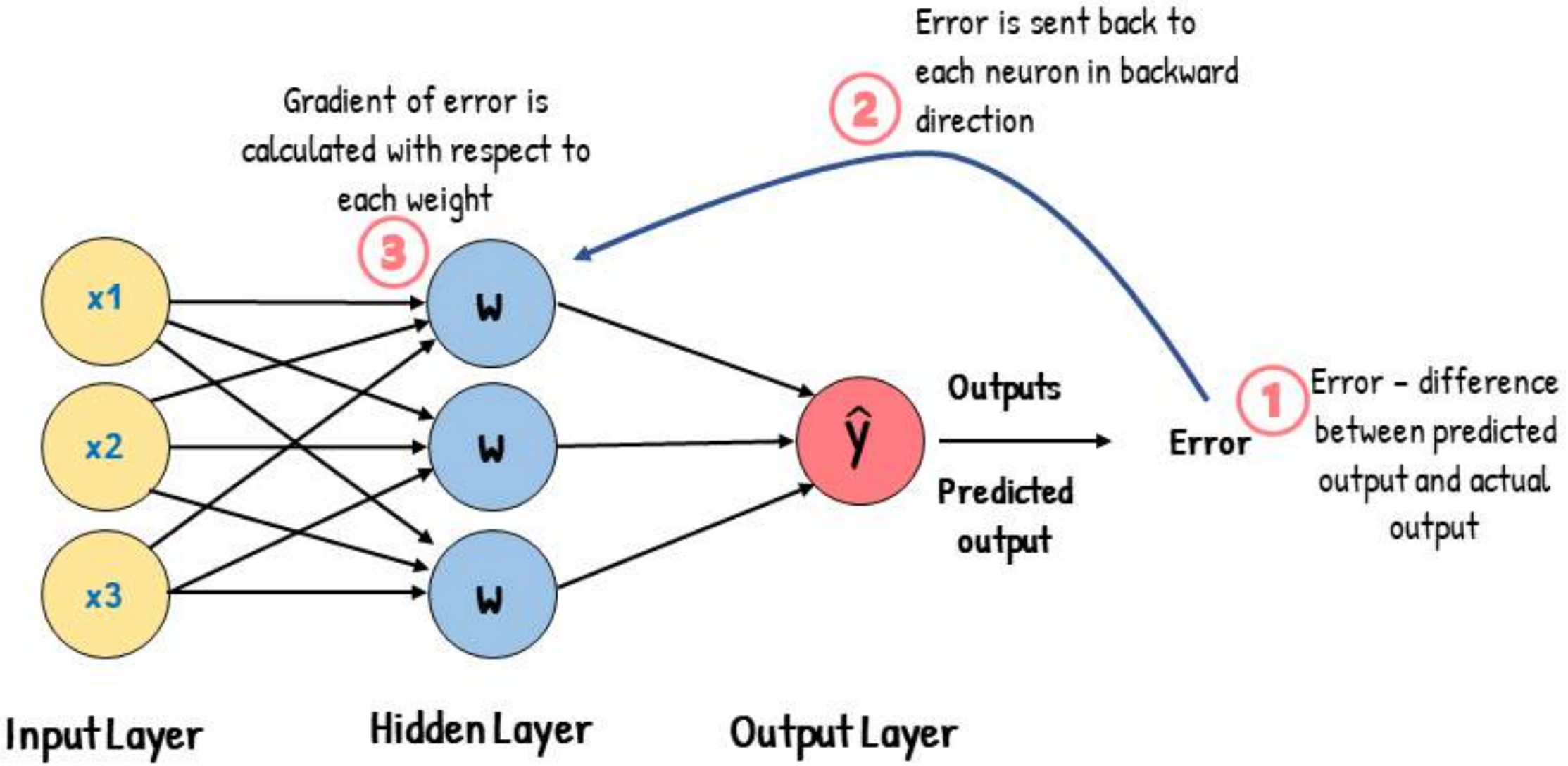
Introduction Neural Network

David Rumelhart, Geoffrey Hinton and Ronald Williams published "**Learning Internal Representations by Error Propagation**" in 1986.



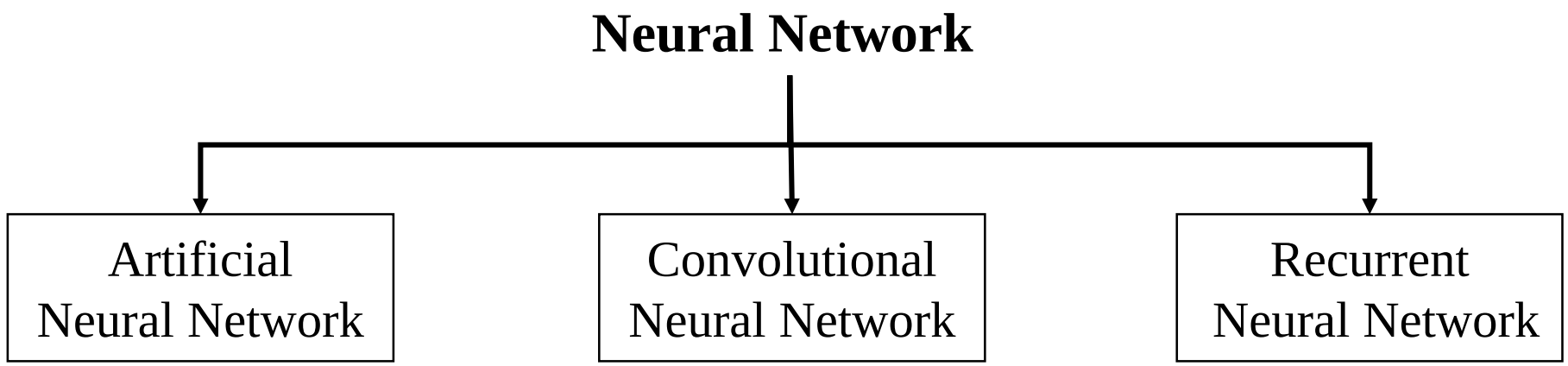
Introduction Neural Network

David Rumelhart, Geoffrey Hinton and Ronald Williams published "**Learning Internal Representations by Error Propagation**" in 1986.



Introduction Neural Network

Types of Neural Network



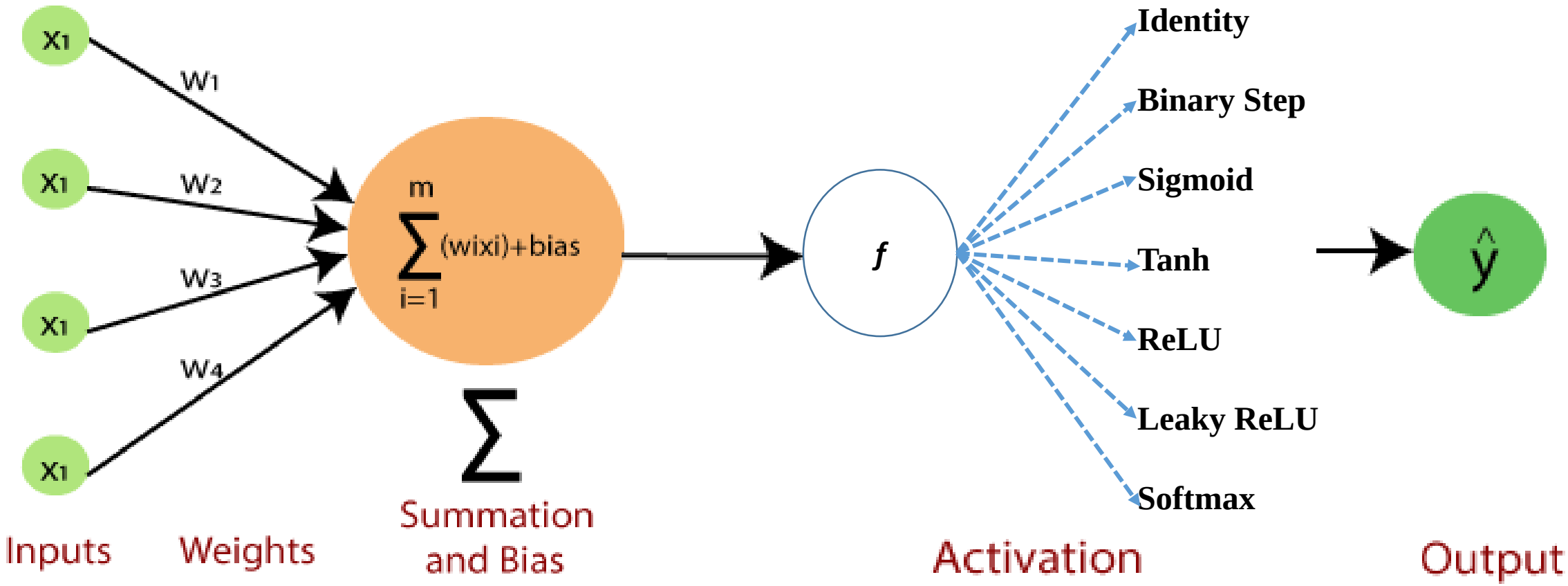
- Numerical Data
- Tabular Data
- Images and Videos

- Images and Videos

- Sequential or Time Series Data

Activation Function

In **Neural Networks**, the **Activation Function (Mathematical Expression)** of a node defines the output of that node given an input or set of inputs.



Activation Function

In **Neural Networks**, the **Activation Function (Mathematical Expression)** of a node defines the output of that node given an input or set of inputs.

Linear Models

$$y = \sum_{k=1}^{n} w_i x_i + b$$

where, *x* = input, *w* = weight, and *b* = bias

Non-Linear Models

$$y = \alpha\,(w_1\,x_1 + w_2\,x_2 + \ldots + w_n\,x_n + b)$$

$$y = \alpha\,(\sum_{k=1}^{n} w_i x_i + b) = w_1\,x_1 + w_2\,x_2 + \ldots + wn\,xn + b$$

where α is the activation function

Activation Functions are used:

1. **Convert linear output into non-linear output.**
2. **To learn pattern in the data.**

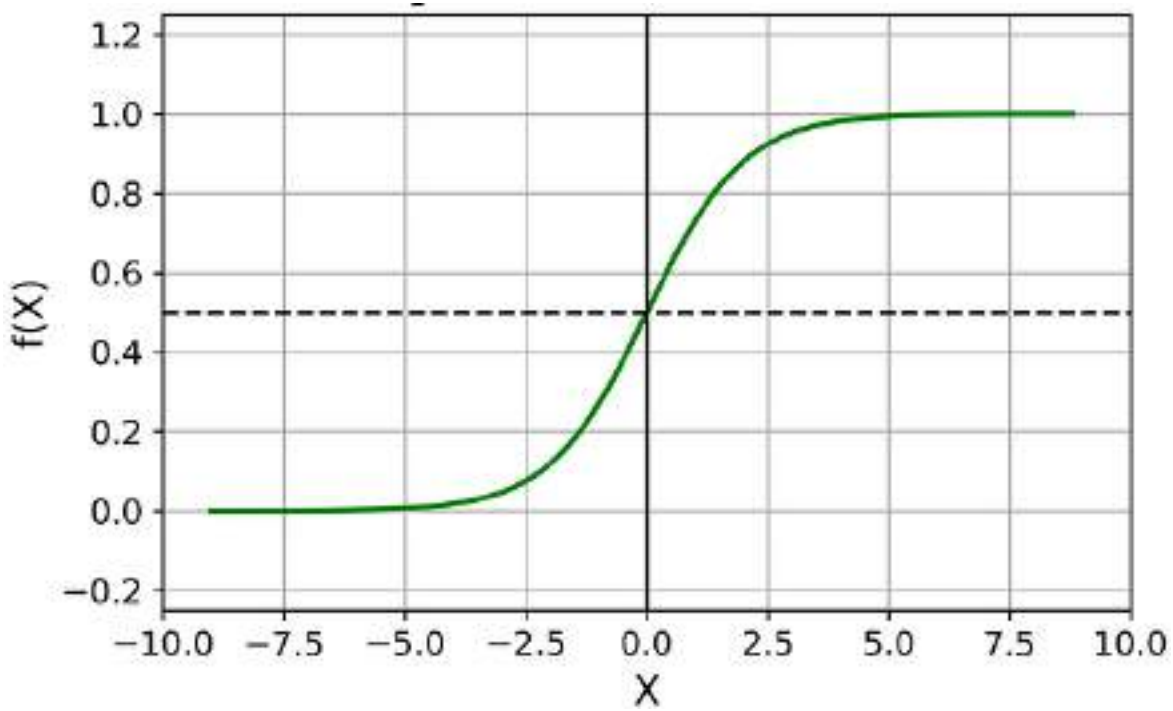
Activation Function

In **Neural Networks**, the **Activation Function (Mathematical Expression)** of a node defines the output of that node given an input or set of inputs.

Sigmoid Activation Function

- The sigmoid function is a non-linear AF used primarily in feed forward neural networks.
- The sigmoid function appears in the output layer of the deep learning models and is used for predicting probability-based outputs.

Equation	$\boldsymbol{f(x) = \frac{1}{1 + e^{-x}}}$
Range	**0 to 1**
Nature	$\boldsymbol{Non - Linear}$



Activation Function

In **Neural Networks**, the **Activation Function (Mathematical Expression)** of a node defines the output of that node given an input or set of inputs.

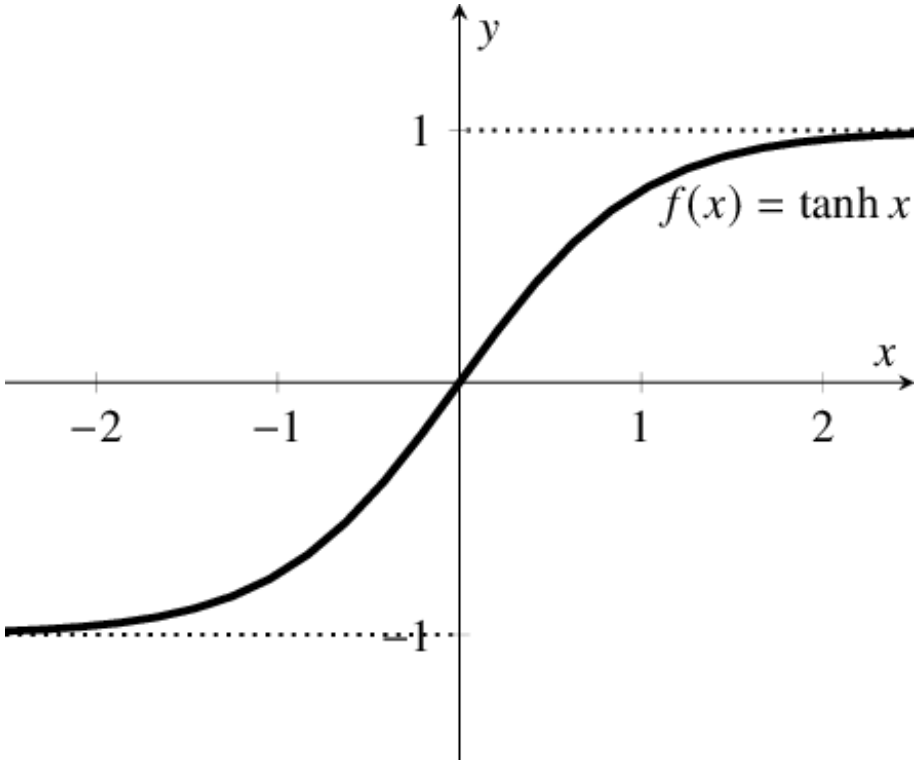
Tanh Activation Function (Tangent Hyperbolic Function)

- The tanh function is a non-linear AF used primarily in feed forward neural networks.
- The sigmoid function appears in the output layer of the deep learning models and is used for predicting probability-based outputs.
- It is 0-centric in nature.

Equation $f(x) = \dfrac{2}{1 + e^{-2x}} - 1$

Range -1 **to** 1

Nature $Non - Linear$



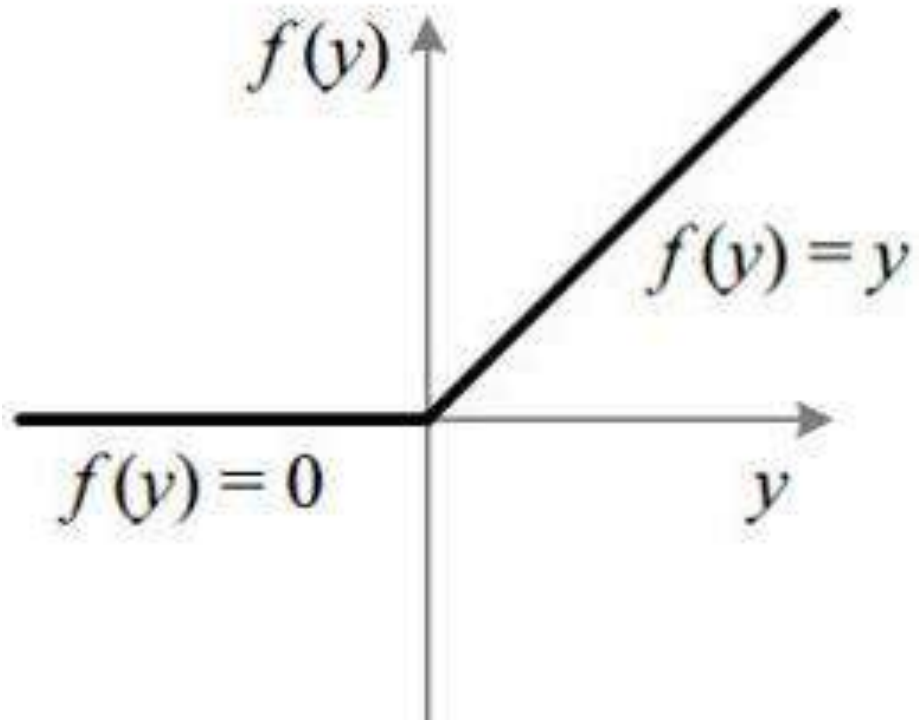
Activation Function

In Neural Networks, the **Activation Function (Mathematical Expression)** of a node defines the output of that node given an input or set of inputs.

ReLU Activation Function (Rectified Linear Unit)

- The **ReLU** function is a non-linear AF used primarily in feed forward neural networks.
- It is 0-centric in nature.
- The calculation speed is much **faster** then sigmoid and tanh activation function.
- **Dead ReLU Problem.**

Equation	$\boldsymbol{f(x) = max(0, x)}$
Range	**0 to** $\boldsymbol{infinity}$
Nature	$\boldsymbol{Non - Linear}$



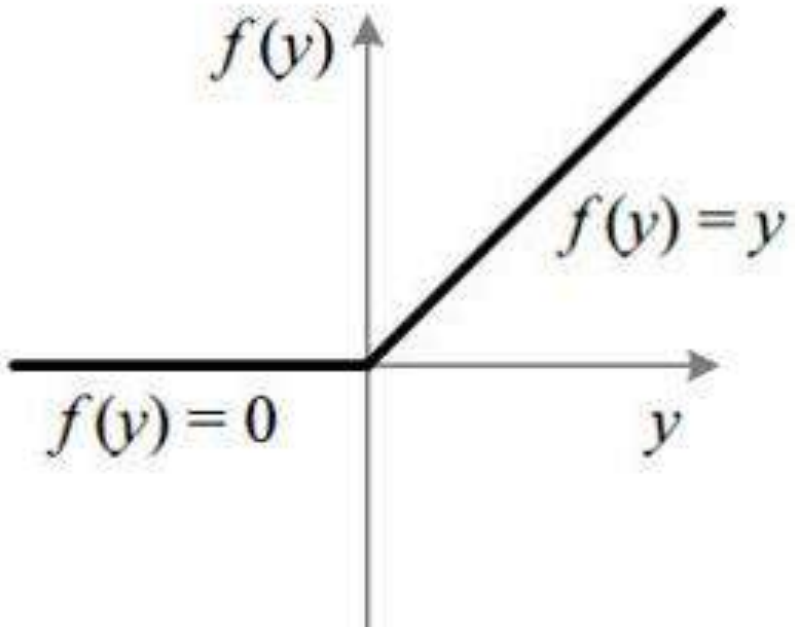
Activation Function

In **Neural Networks**, the **Activation Function (Mathematical Expression)** of a node defines the output of that node given an input or set of inputs.

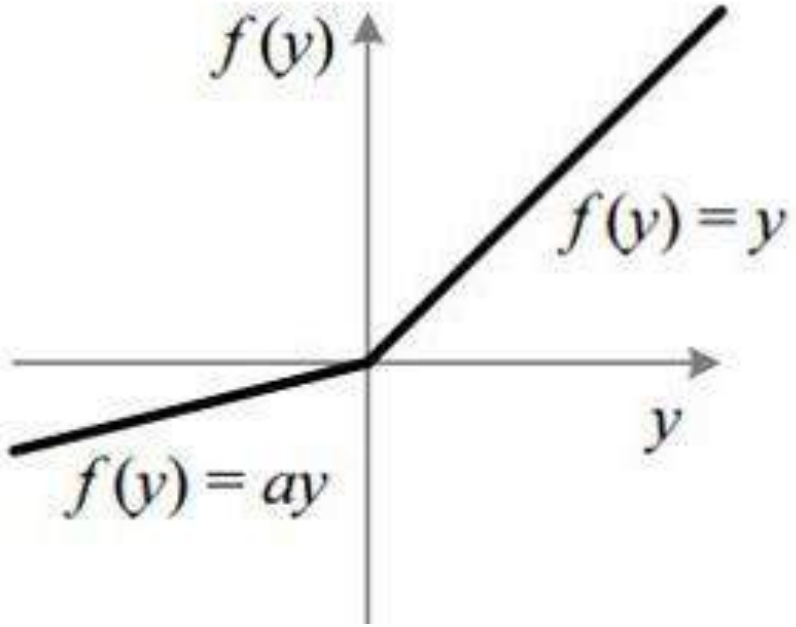
Leaky ReLU Activation Function

- **Specifically designed to compensate** for the dying ReLU problem.
- It is 0-centric in nature.

Equation $f(x) = max(\alpha x, x)$ **Range** $-infinity$ **to** $infinity$



ReLU



Leaky ReLU

Activation Function

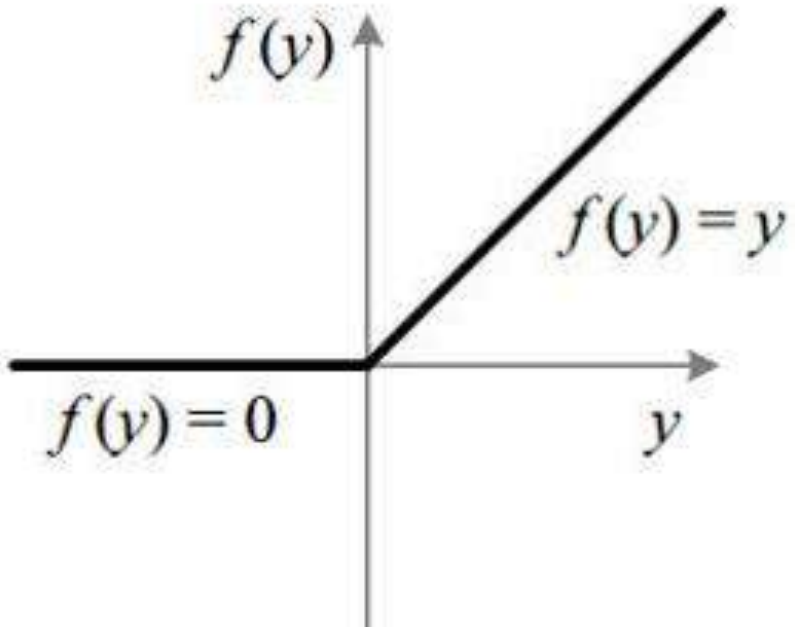
In **Neural Networks**, the **Activation Function (Mathematical Expression)** of a node defines the output of that node given an input or set of inputs.

Softmax Activation Function

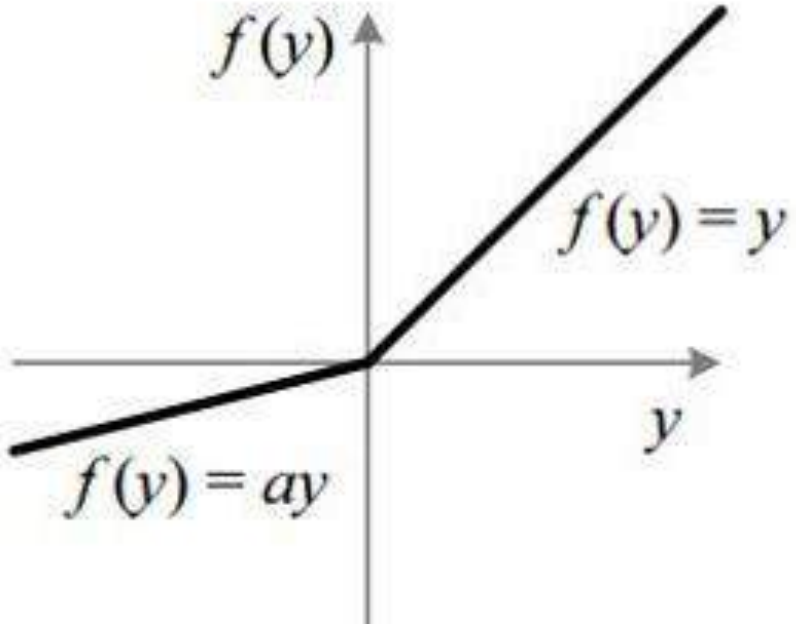
- **Softmax** is used as the **activation function** for multi-class classification problems.

Equation $f(x) = max(\alpha x, x)$

Range $-infinity$ **to** $infinity$



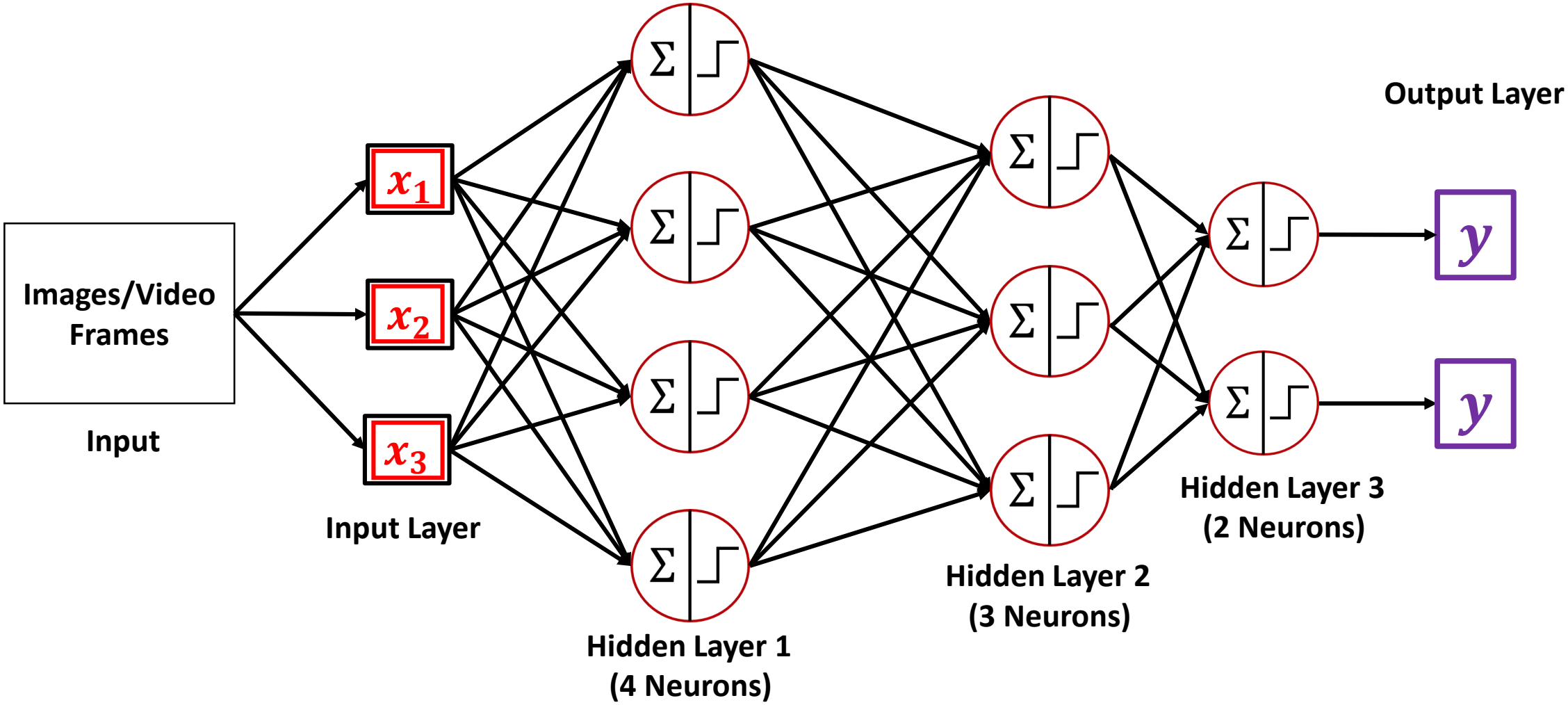
ReLU



Leaky ReLU

Convolutional Neural Network

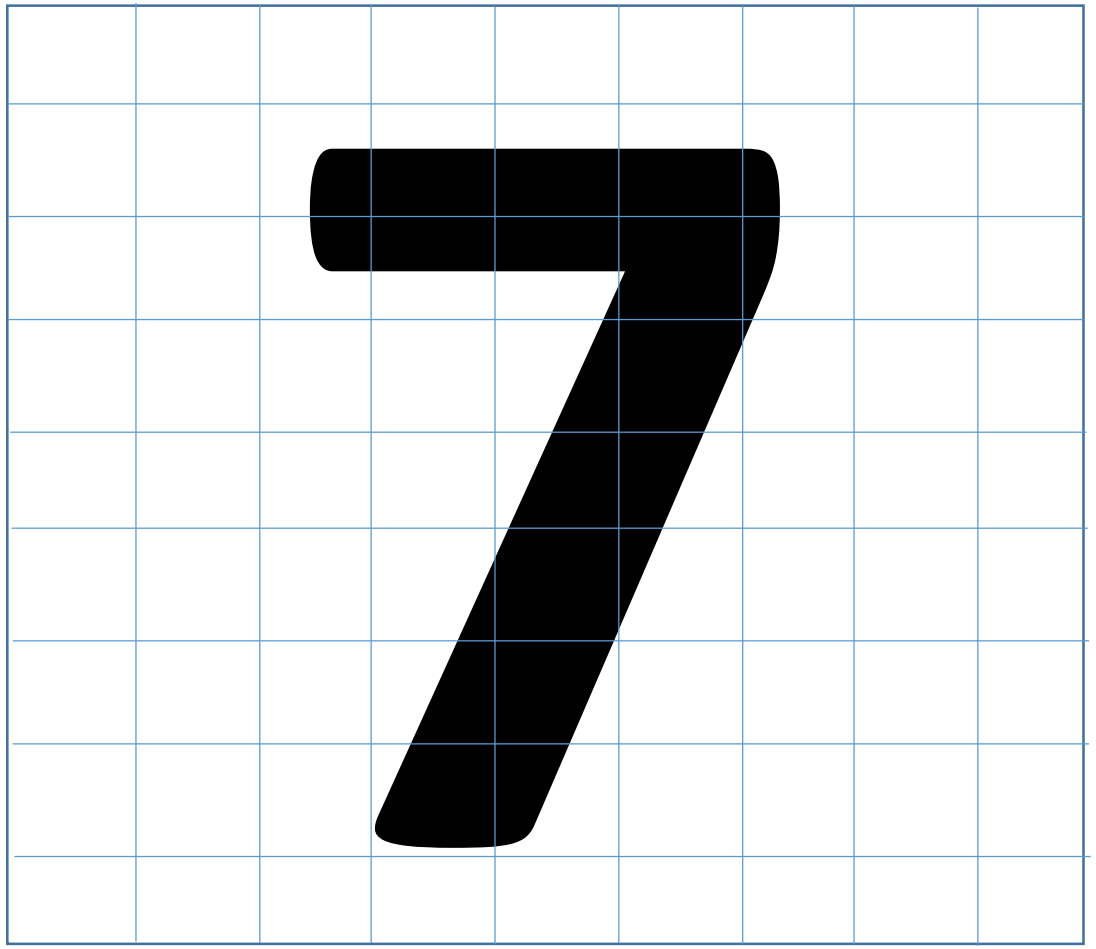
In **Convolutional Neural Networks**, works with Images and Videos as Input.



Convolutional Neural Network

In **Convolutional Neural Networks**, works with Images and Videos as Input.

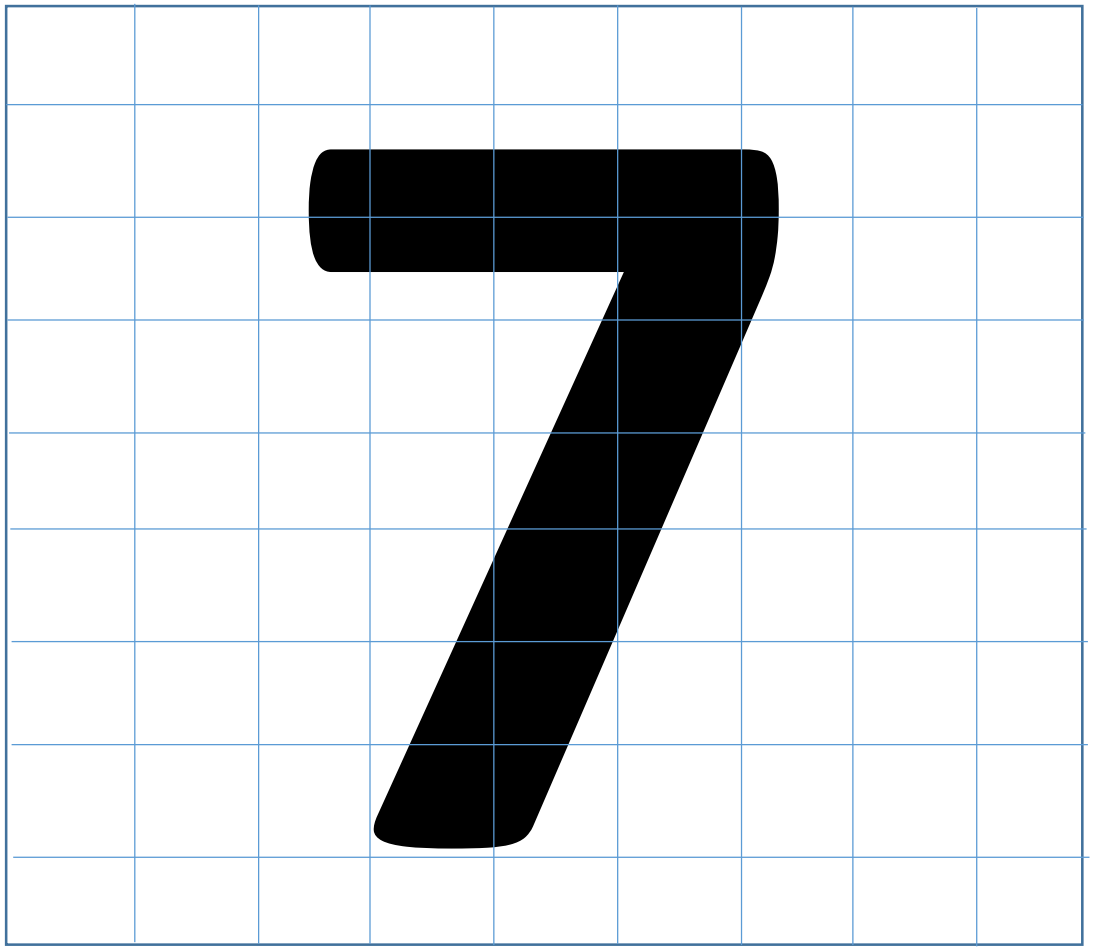
Input Image



Matrix Representation

Convolutional Neural Network

In **Convolutional Neural Networks**, works with Images and Videos as Input.



Matrix Representation

0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Binary Image

Convolutional Neural Network

In **Convolutional Neural Networks**, works with Images and Videos as Input.

Layers in Convolutional Neural Networks:

1. **Convolutional Layer:** Extract Valuable Features from the input image known as Feature Maps.

2. **Activation Layer:** Converts Linear output to non-linear output.

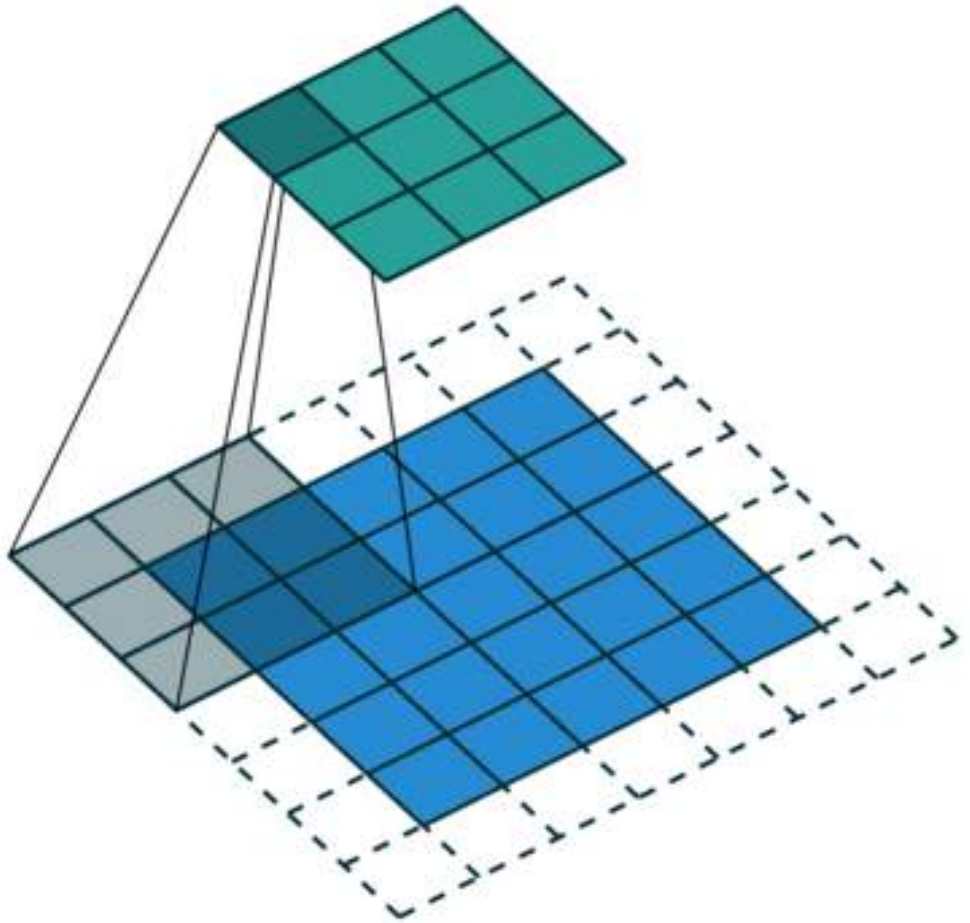
3. **Pooling Layer:** Pooling layers, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input.

4. **Fully Connected Layer (FC Layer):** The task of classification based on the features extracted through the previous layers

Convolutional Neural Network

Layers in Convolutional Neural Networks:

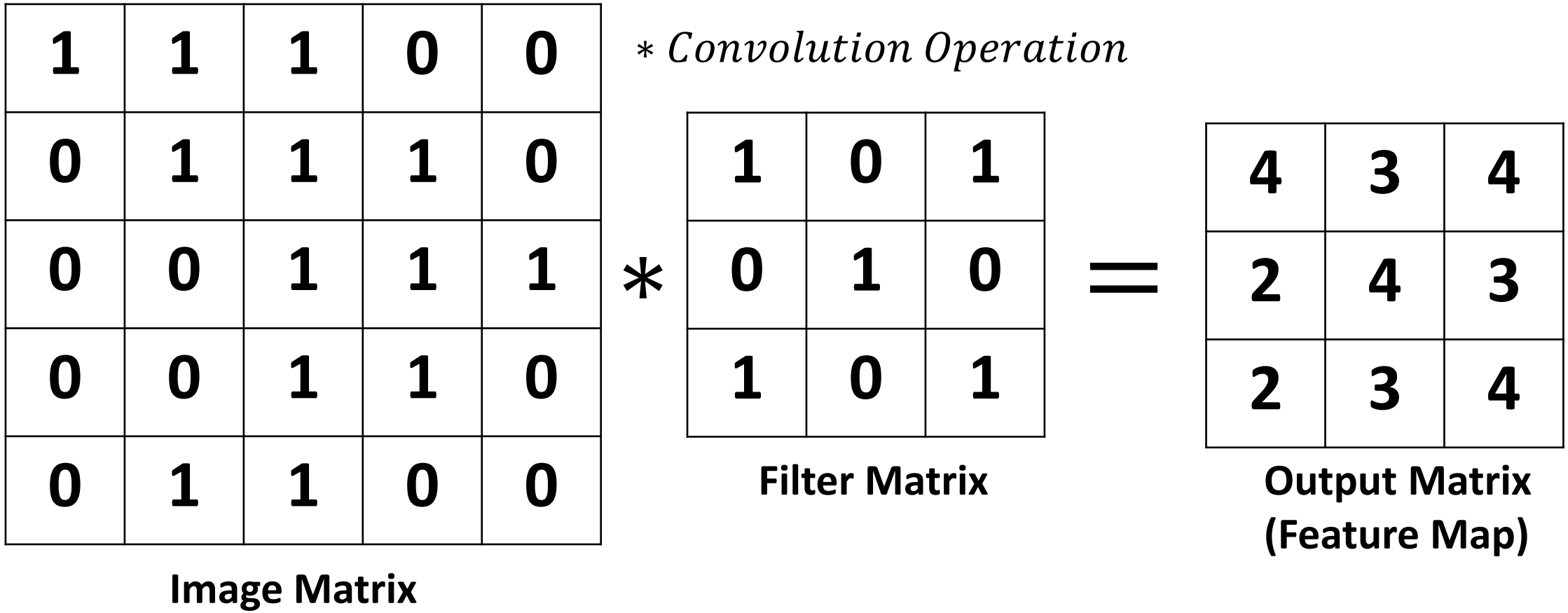
1. **Convolutional Layer:** Extract Valuable Features from the input image known as Feature Maps.



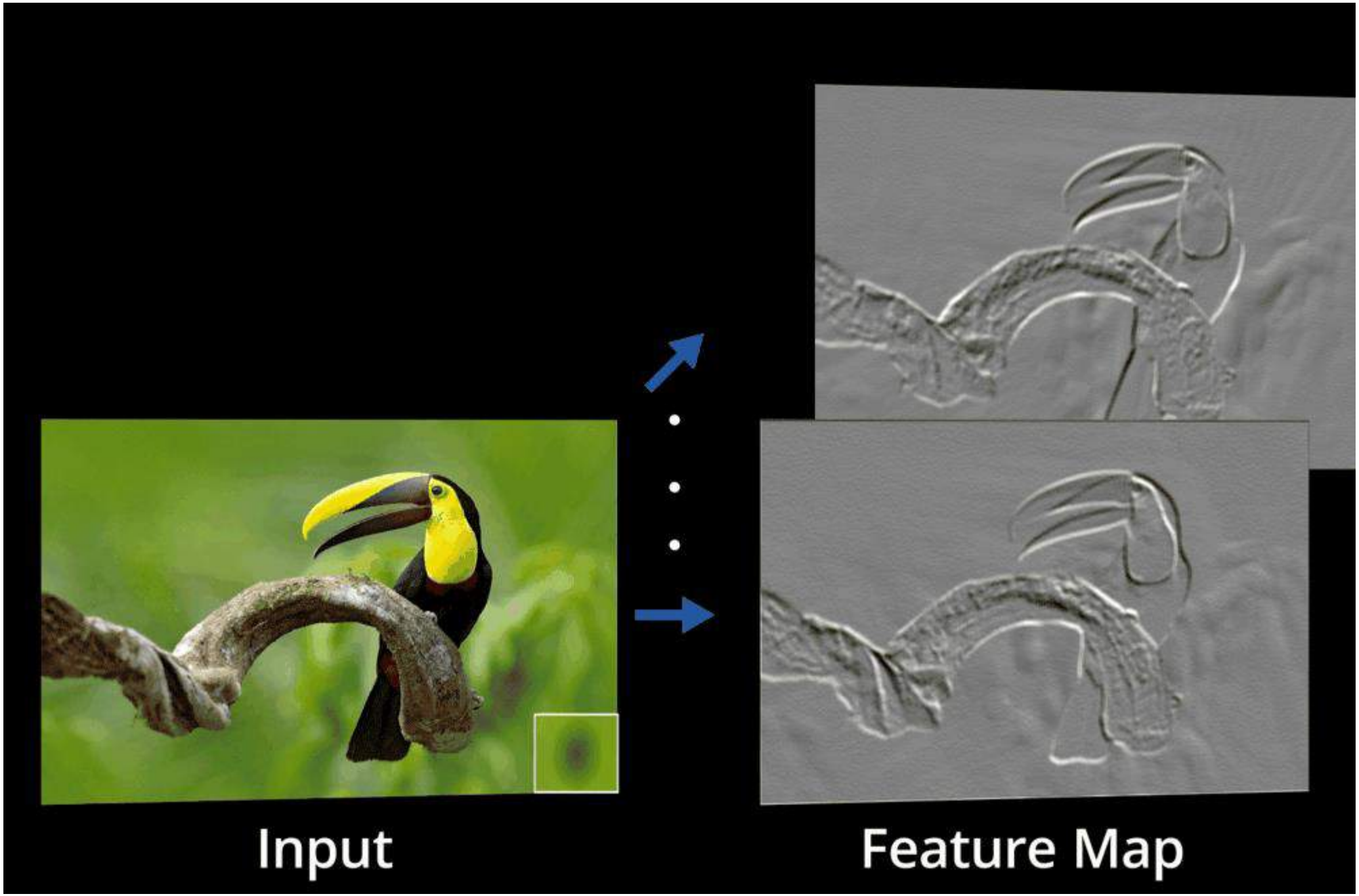
Convolutional Neural Network

Layers in Convolutional Neural Networks:

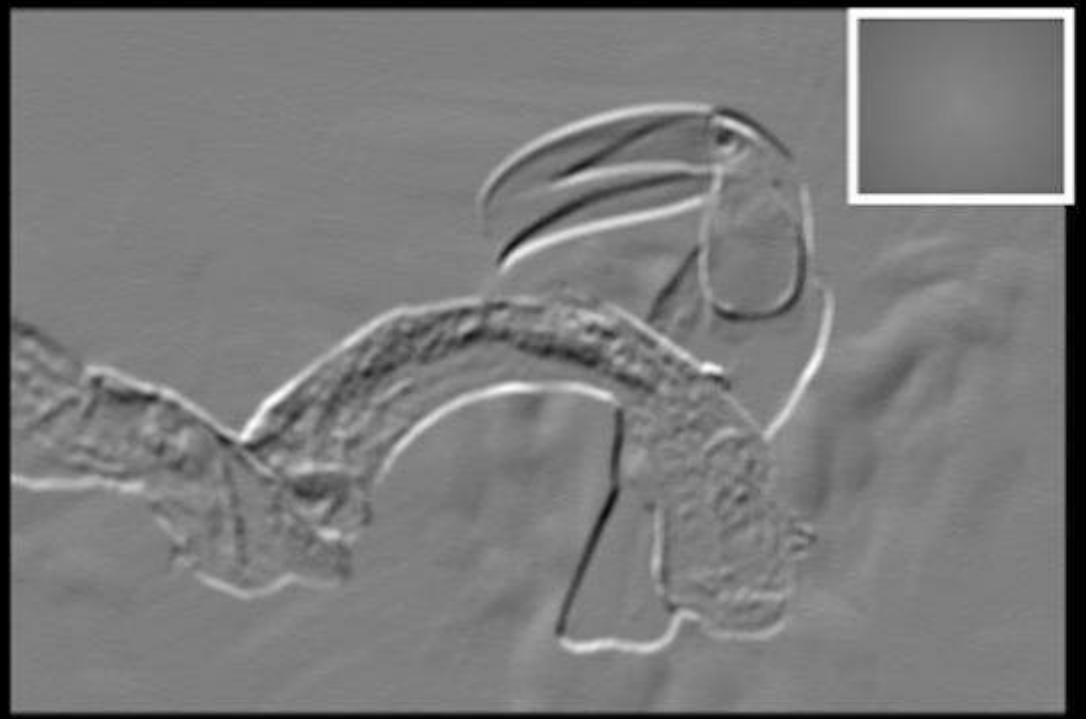
1. **Convolutional Layer:** Extract Valuable Features from the input image known as Feature Maps.



conv2D(filters = 6, kernel_size = 3x3, activation = ‘relu’, padding= ‘same’)


Input
Feature Map

Input Feature Map

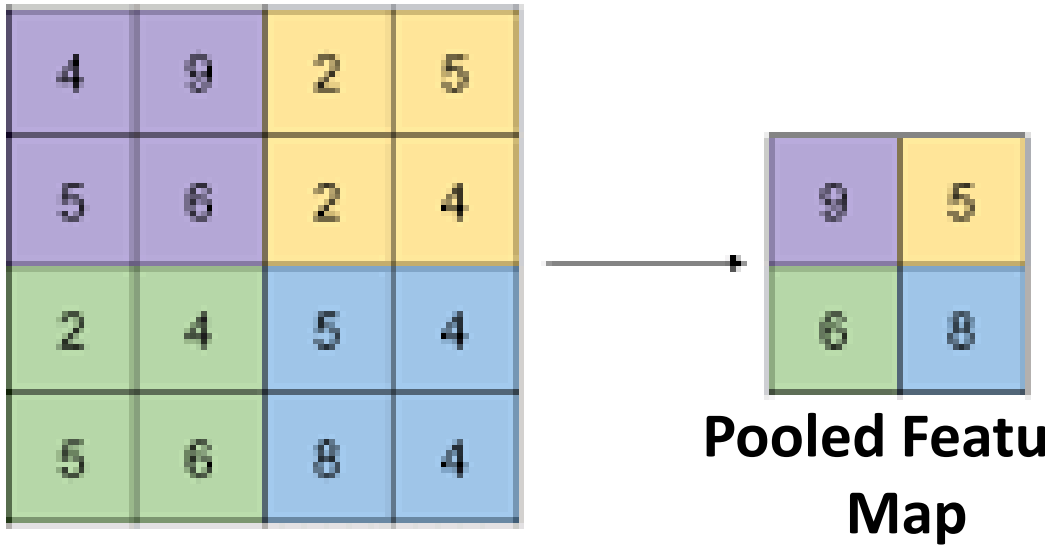


Convolutional Neural Network

Layers in Convolutional Neural Networks:

3. Pooling Layer: Pooling layers, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input.

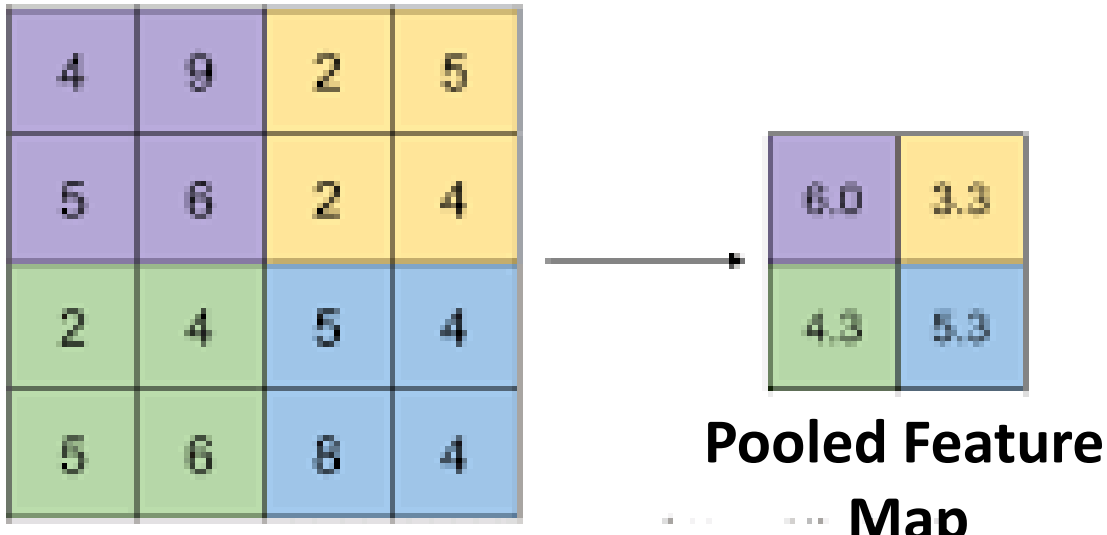
Max Pooling



Output Matrix (Feature Map)

Pooled Feature Map

Average Pooling



Output Matrix (Feature Map)

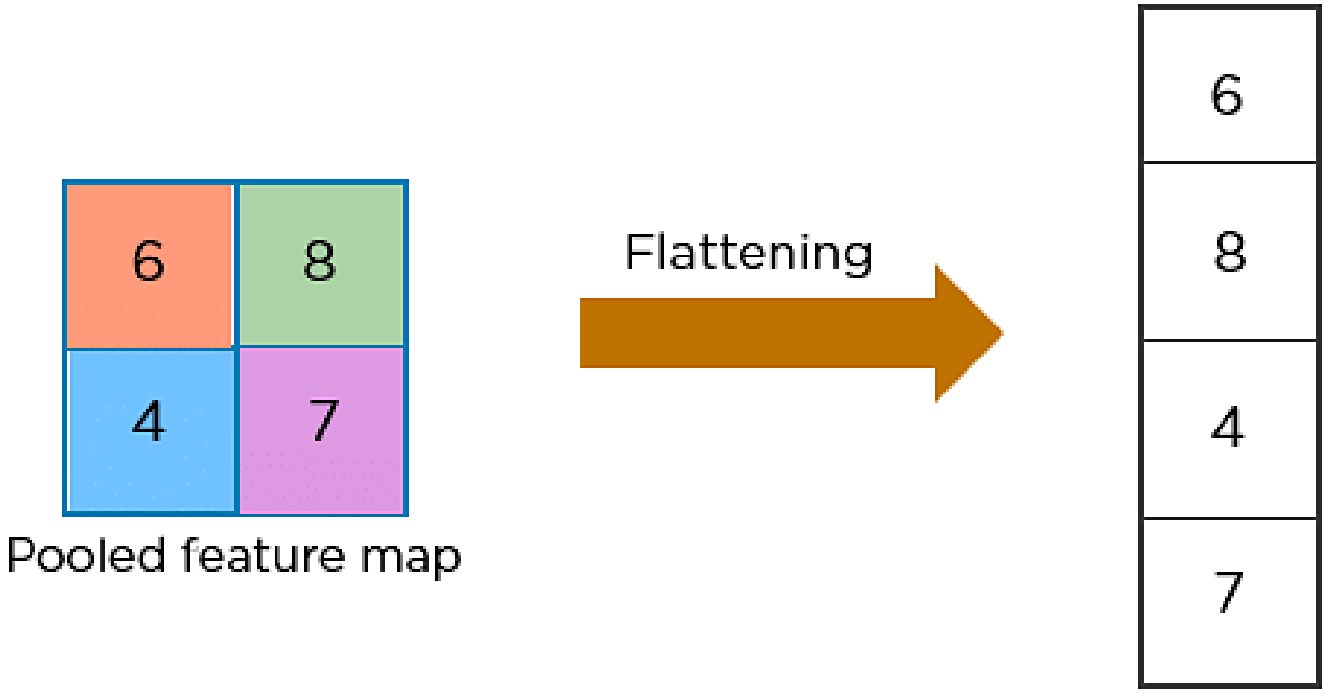
Pooled Feature Map

Convolutional Neural Network

Layers in Convolutional Neural Networks:

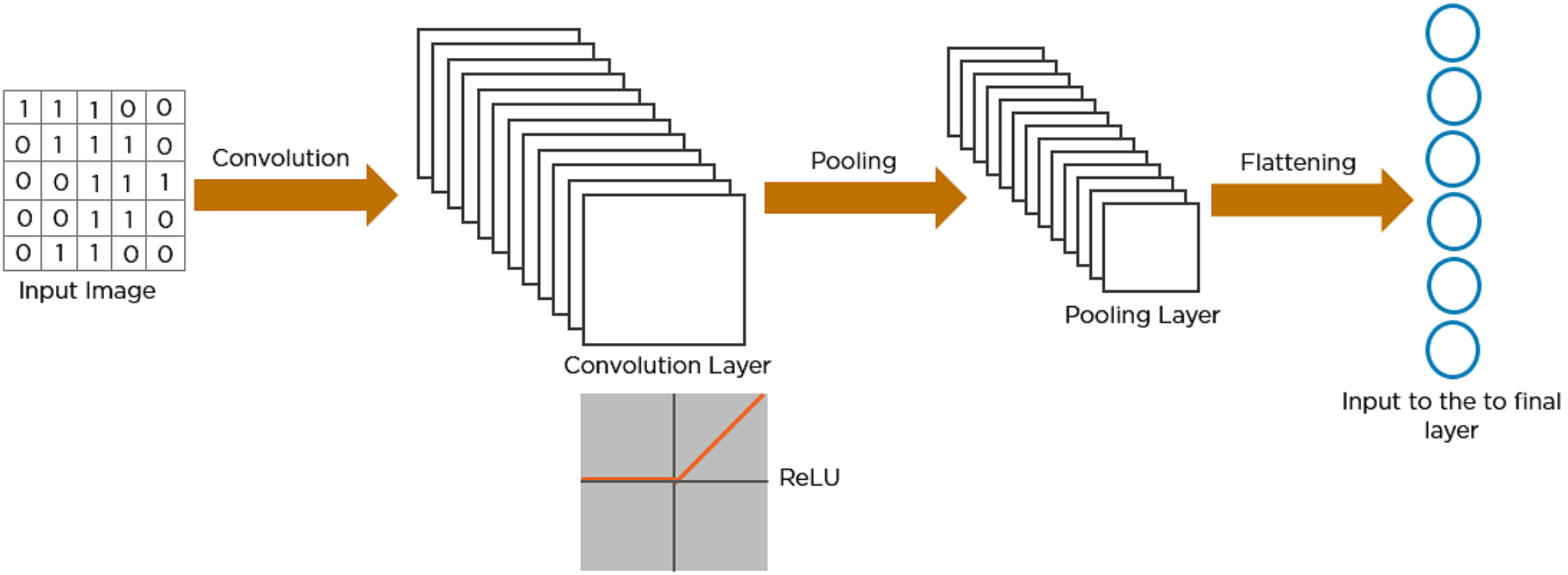
4. **Fully Connected Layer (FC Layer):** The task of classification based on the features extracted through the previous layers

Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector.



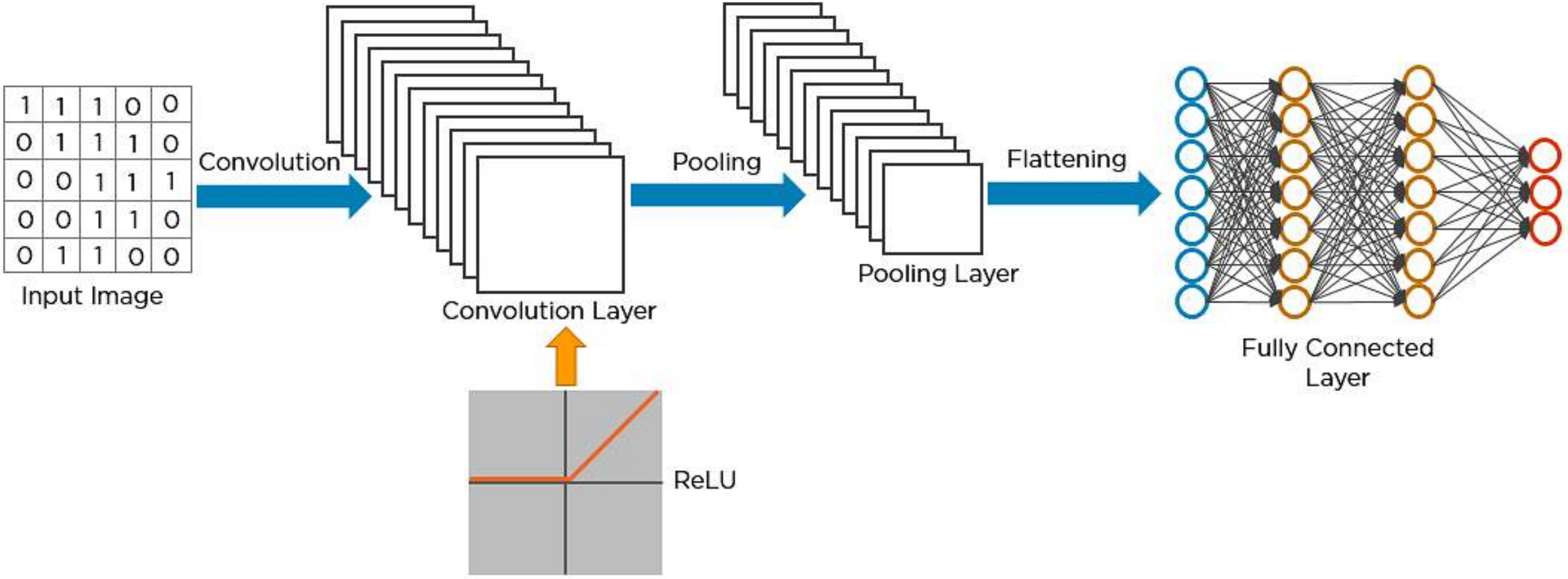
Convolutional Neural Network

Layers in Convolutional Neural Networks:



Convolutional Neural Network

Layers in Convolutional Neural Networks:

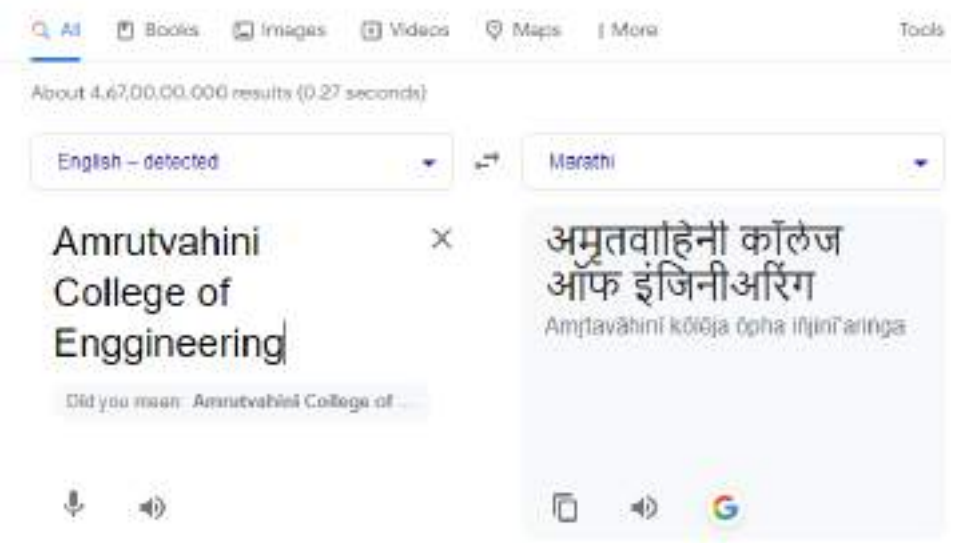


Recurrent Neural Network

I really like the color for my new phone

I didn’t really like the color for my new phone

Sentiment Classification



Language Translation

Image Captioning

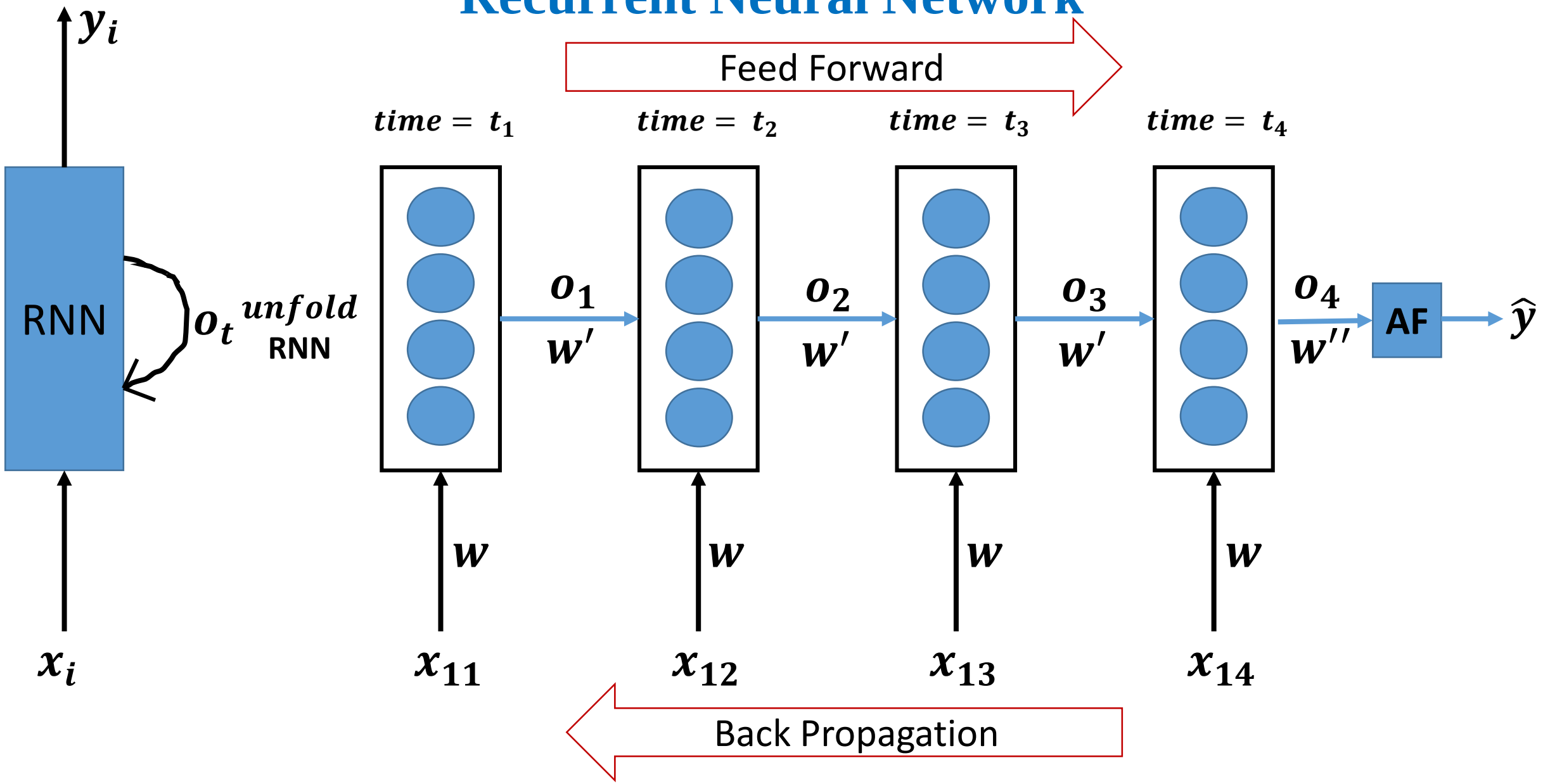
A person riding a motorcycle on a dirt road.

Two dogs play in the grass.

A group of young people playing a game of frisbee.

Two hockey players are fighting over the puck.

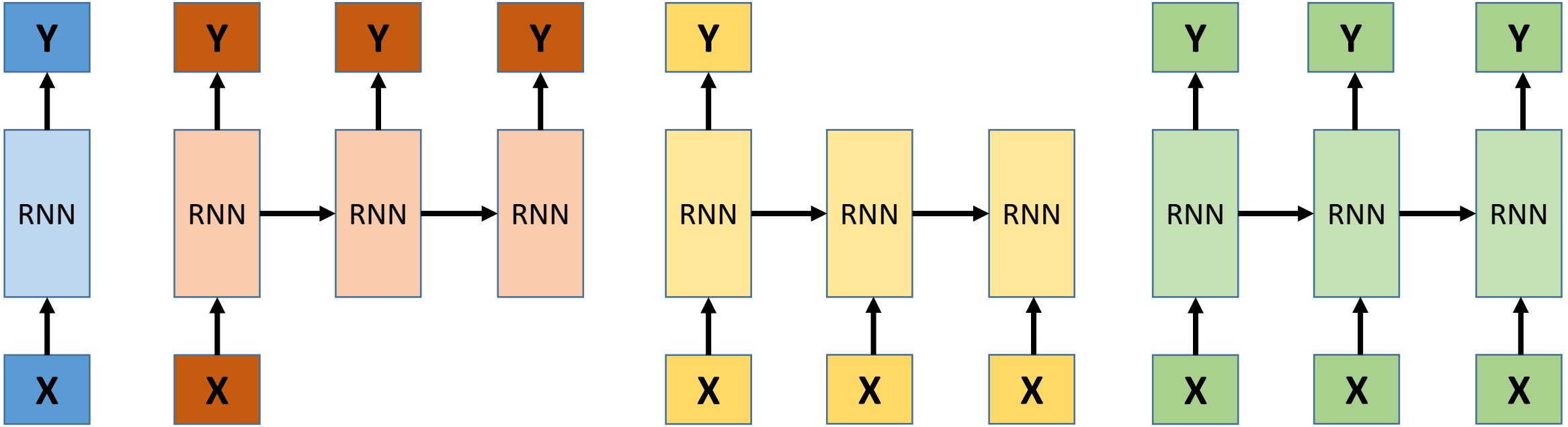
Recurrent Neural Network



$Neuron$ $x_i = input$ $o_i = output$ $w = weight$ $\hat{y} = output$ AF = Activation Function

Recurrent Neural Network

Types of RNN



One to One

One to Many

Many to One

Many to Many

Assignment VI

Q1. Explain the Working of Biological Neuron and Artificial Neuron?

Q2. Explain CNN in detail?

Q3. Explain Feed Forward in RNN? List out various applications?

Check your assignment on or before Saturday (31/10/2023)

References

- McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity.“ *The bulletin of mathematical biophysics* 5 (1943): 115-133.
- Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65, no. 6 (1958): 386.
- Minsky, Marvin, and Seymour Papert. "An introduction to computational geometry." *Cambridge tiass., HIT* 479 (1969): 480.
- Fukushima, Kunihiko. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position." *Biological cybernetics* 36, no. 4 (1980): 193-202.
- Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science, 1986.