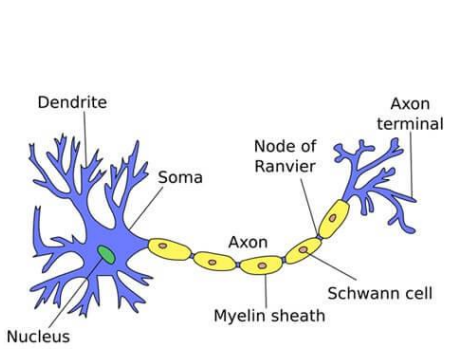
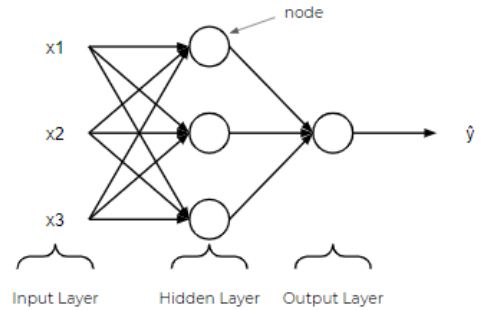
6 Neural Networks and Deep Learning

Dr. Sultan Alfarhood

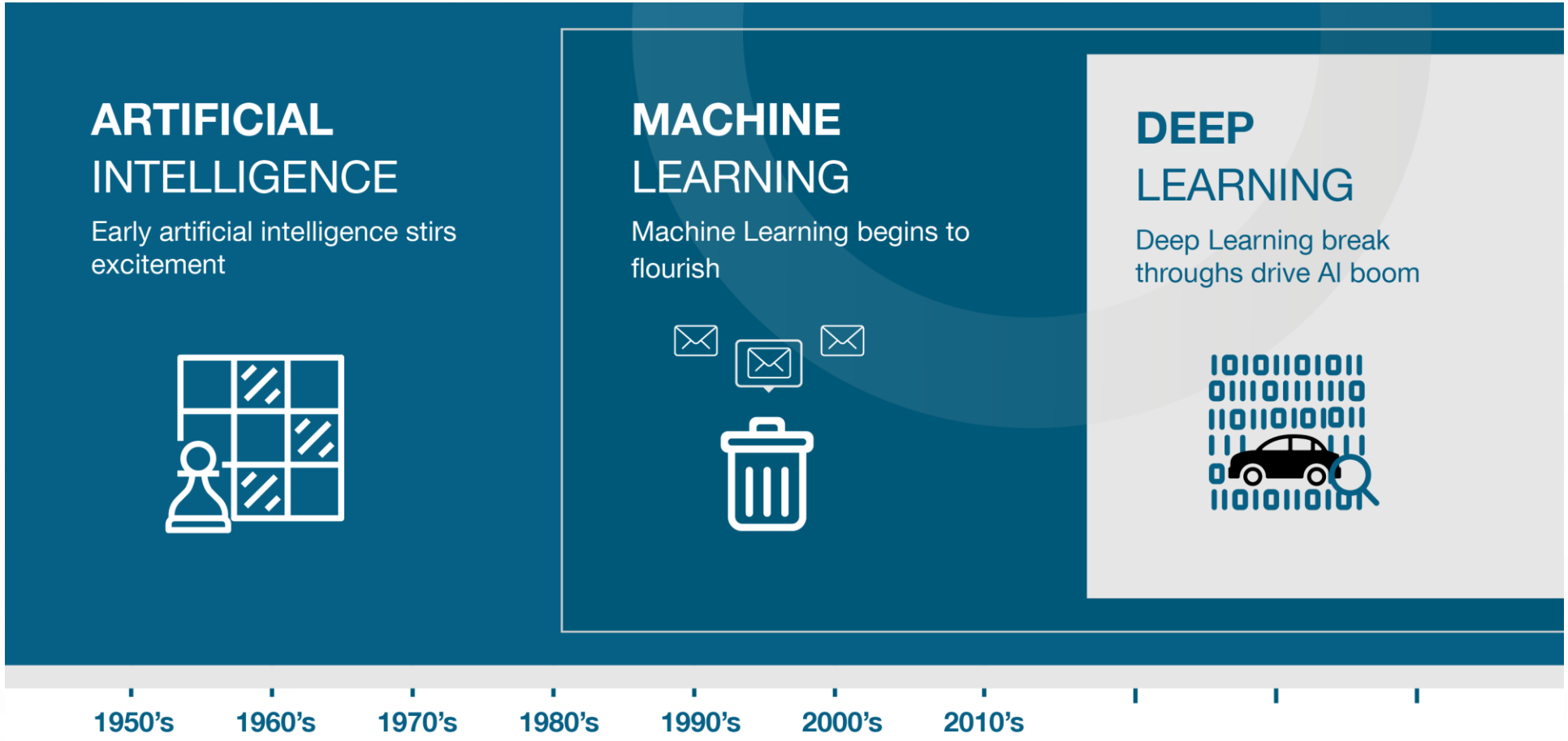
Neural Networks (NN)

- A neural network is a method that teaches computers to process data in a way that is inspired by the human brain.
- It uses interconnected **nodes** or **neurons** in a layered structure that resembles the human brain.
- A neural network that consists of more than three layers can be considered a **deep learning** algorithm.





Deep Learning



Architecture of Neural Networks

Input Layer

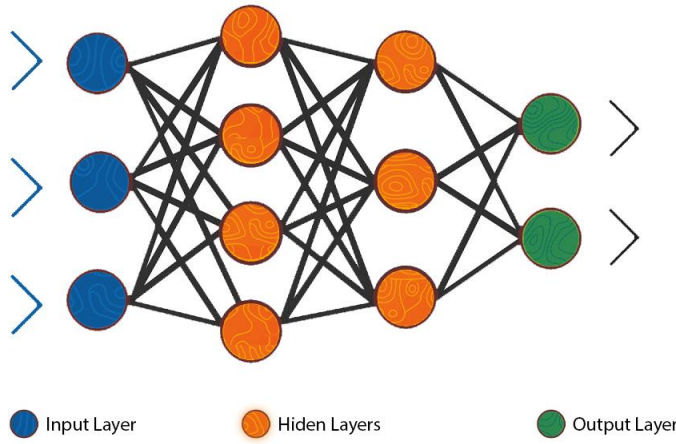
- Information from the outside world enters the artificial neural network from the input layer.
- Input nodes process the data, analyze or categorize it, and pass it on to the next layer.

Hidden Layers

- Hidden layers take their input from the input layer or other hidden layers.
- Artificial neural networks can have many hidden layers.
- Each hidden layer analyzes the output from the previous layer, processes it further, and passes it on to the next layer.

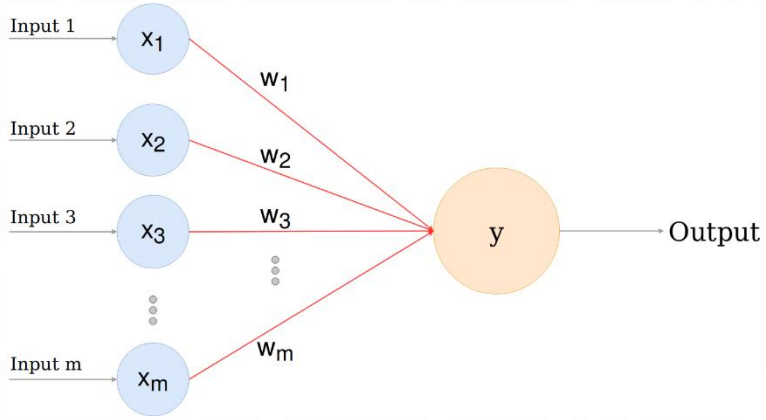
Output Layer

- The output layer gives the final result of all the data processing by the artificial neural network.
- It can have single or multiple nodes.

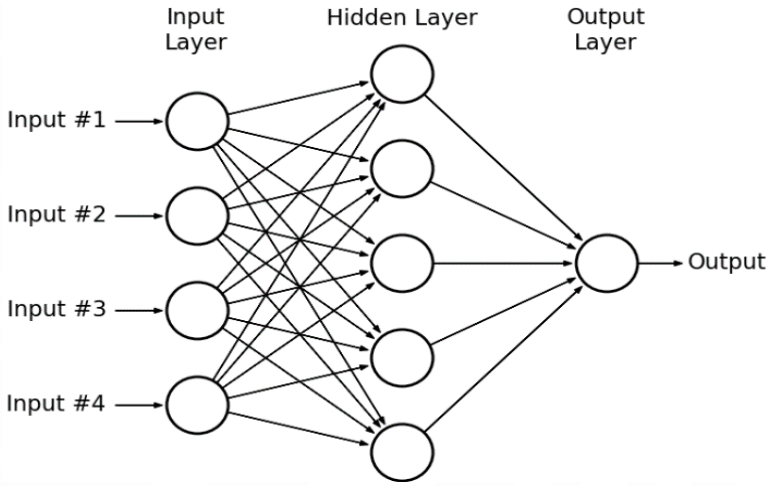


Simple Neural Networks

- **Perceptron** is a simple form of Neural Network
  - Consists of a single layer where all the mathematical computations are performed.
  - The perceptron was invented in 1943 by Warren McCulloch and Walter Pitts.
  - The first implementation was a machine built in 1958



- **Multilayer Perceptron** also known as Artificial Neural Networks or Vanilla Neural Network
  - Consists of more than one perception which is grouped together to form a multiple layer neural network.



Neural Networks (NN)

- A neural network (NN) is a function:

$$y = f_{NN}(x)$$

- The function f_{NN} is a nested function.

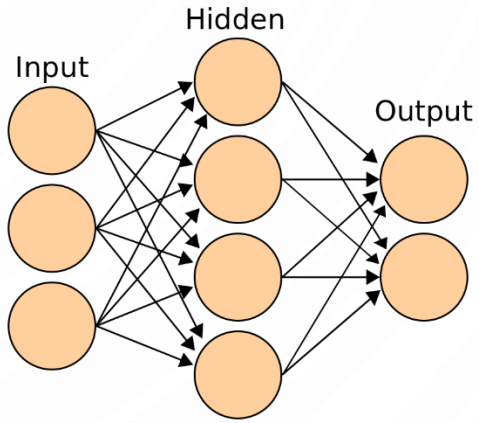
Example: a 3-layer neural network that returns a scalar, f_{NN} looks like this:

$$y = f_{NN}(x) = f_3\left(f_2\left(f_1(x)\right)\right)$$

In the above equation, f_1 and f_2 are vector functions of the following form:

$$f_l(z) = g_l(W_l z + b_l)$$

- l is called the layer index and can span from 1 to any number of layers.
- g_l is an activation function.
- W_l , b_l are the parameters learned at each layer.



A Multilayer Perceptron Example

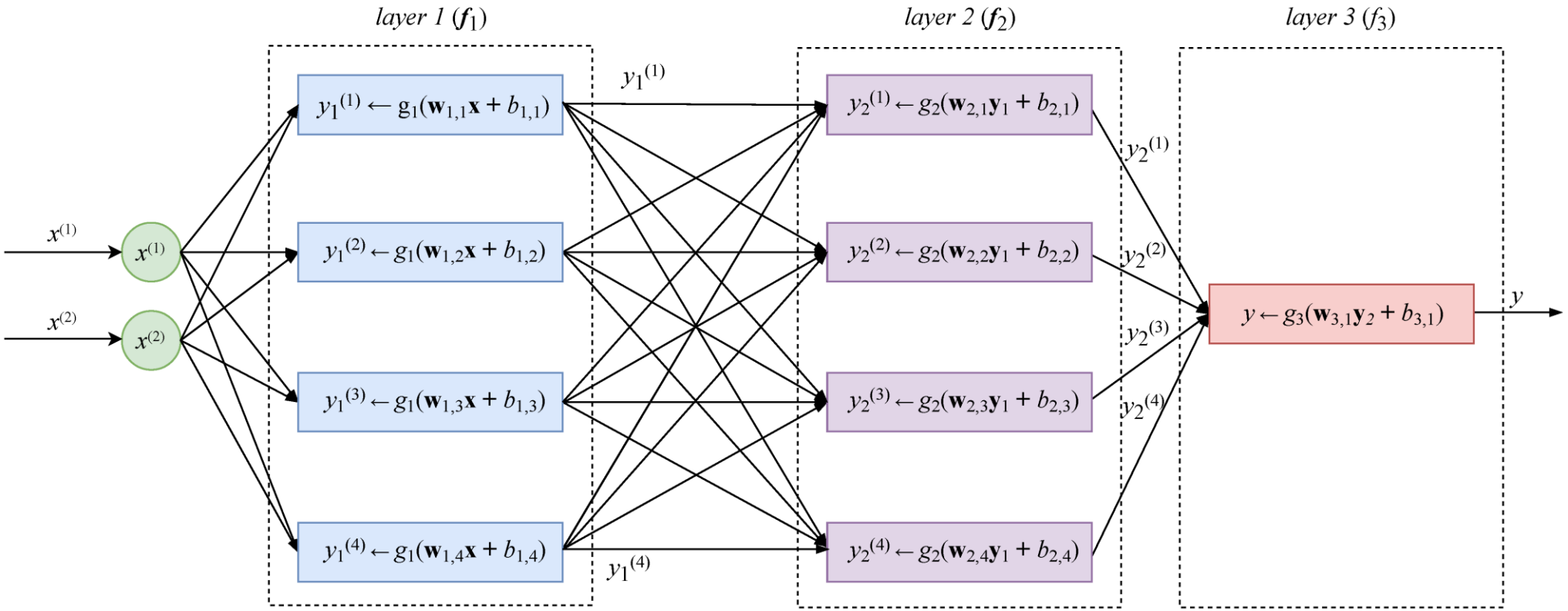


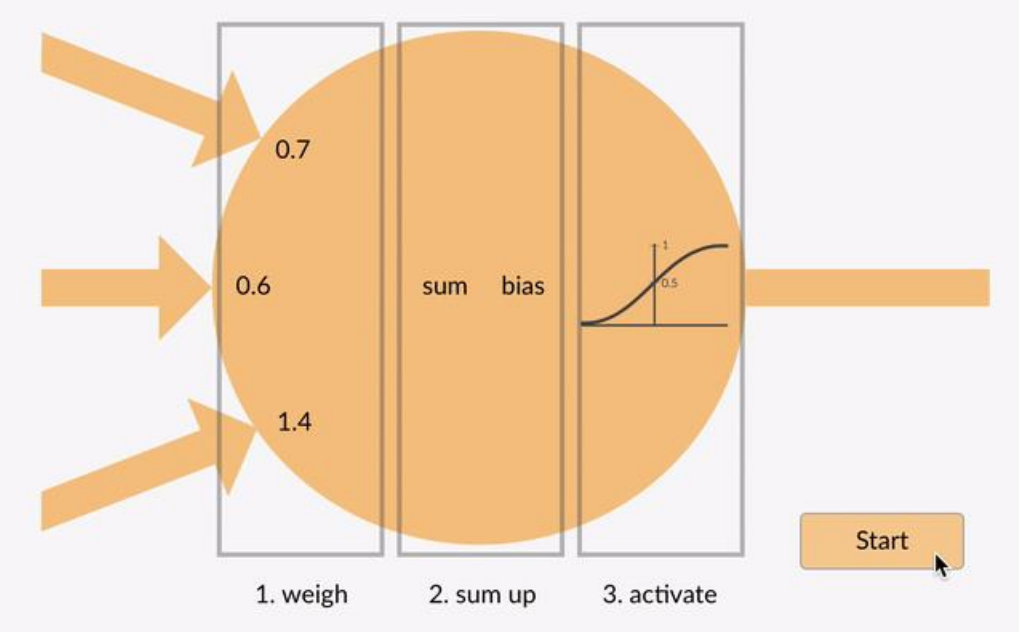
Figure 1: A multilayer perceptron with two-dimensional input, two layers with four units and one output layer with one unit.

Ingredients of Neural Network

Data	Model	Objective Function	Optimization Algorithm
• Tabular Data • Images • Text • Sounds • …	• Nodes Parameters • Activation Functions	• e.g., MSE	• e.g., SGD

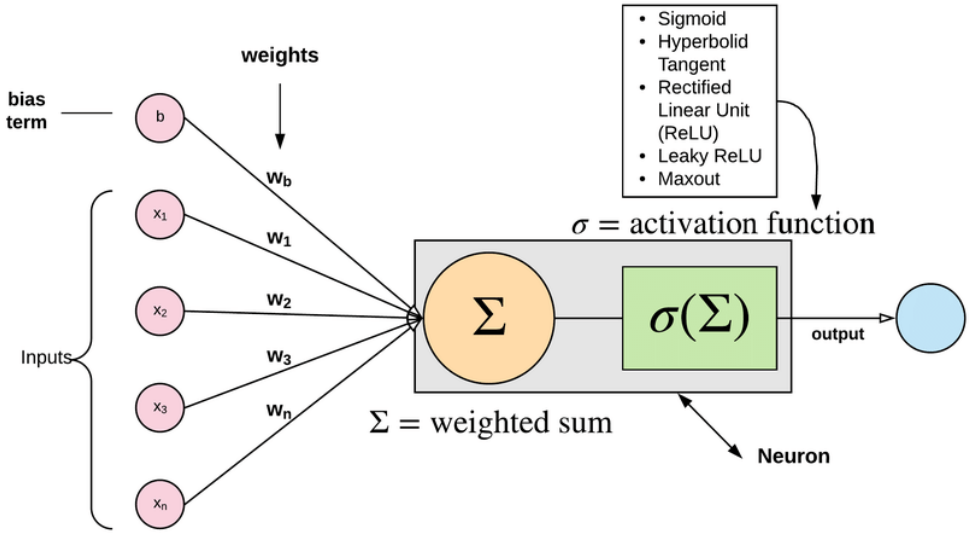
Hidden Layers Computations

1. All the inputs are multiplied by their weights.
2. The activation function is applied.
   - The activation function is a nonlinear transformation that is applied to the input before sending it to the next layer of neurons.

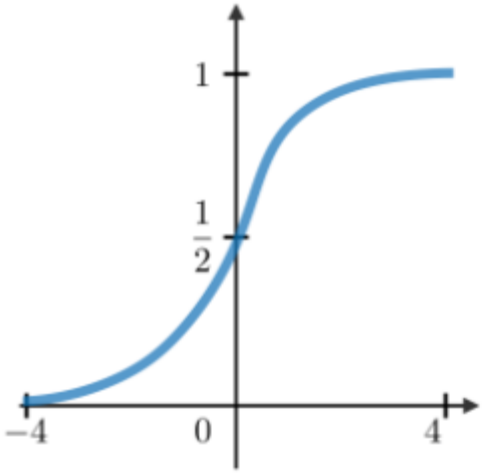
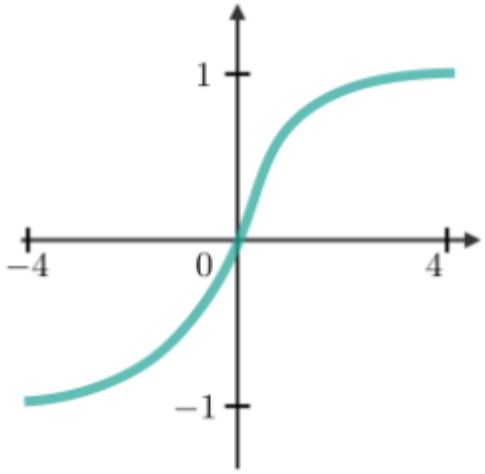
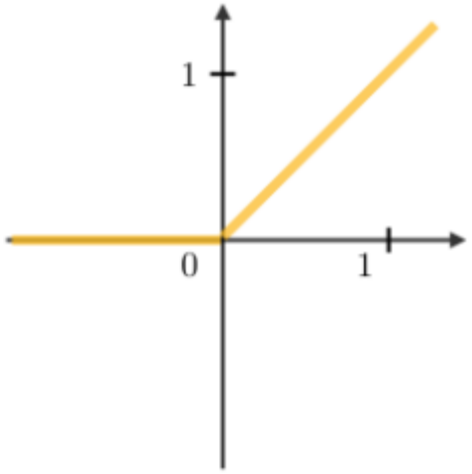


Activation Functions

- Activation functions are attached to each neuron and are mathematical equations that determine **whether a neuron should be activated or not** based on whether the neuron's input is relevant for the model's prediction or not.
  - The purpose of the activation function is to introduce the nonlinearity in the data.
- Examples:
  - **Sigmoid**
  - **Tanh**
  - Rectified Linear Unit Function (**ReLU**)
  - **Softmax**

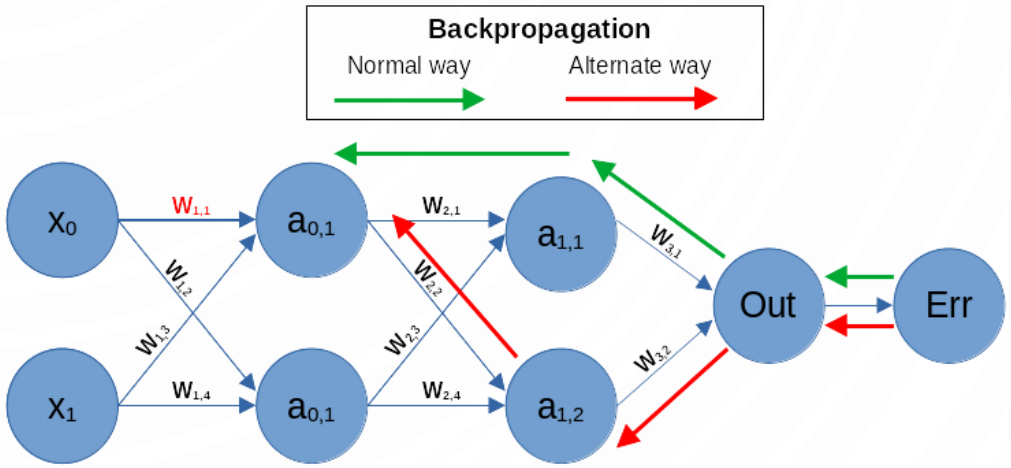


Activation Function Examples

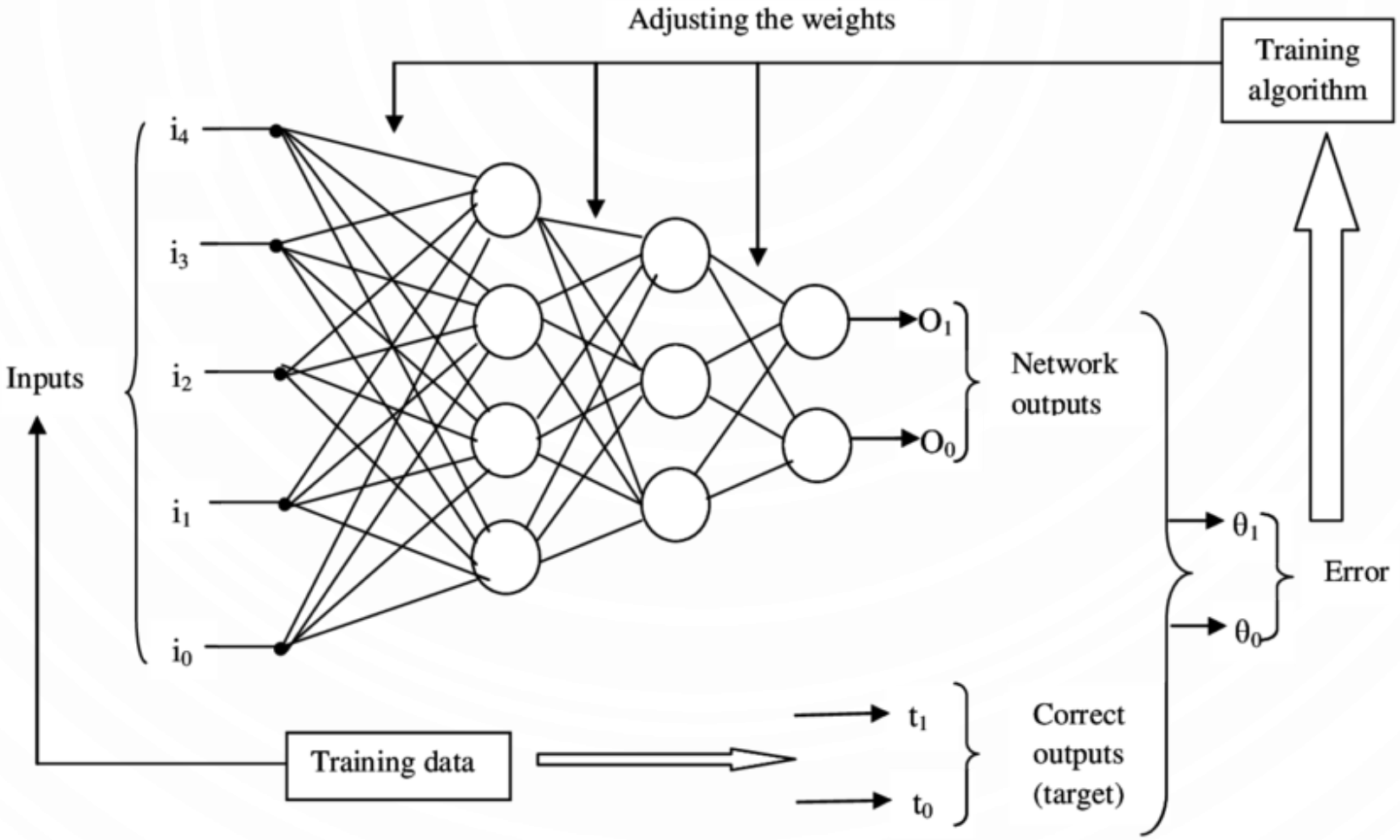
Sigmoid	Tanh	ReLU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$
		

Backpropagation

- **Backpropagation**, short for "backward propagation of errors," is a process involved in training a neural network.
- It involves taking the error rate of a forward propagation and feeding this loss **backward** through the neural network layers to **fine-tune the weights**.
- Given an artificial neural network and an error function, the method calculates the gradient of the error function with respect to the neural network's weights.
  - The "backwards" part of the name stems from the fact that calculation of the gradient proceeds backwards through the network, with the gradient of the final layer of weights being calculated first and the gradient of the first layer of weights being calculated last.
  - Partial computations of the gradient from one layer are reused in the computation of the gradient for the previous layer.



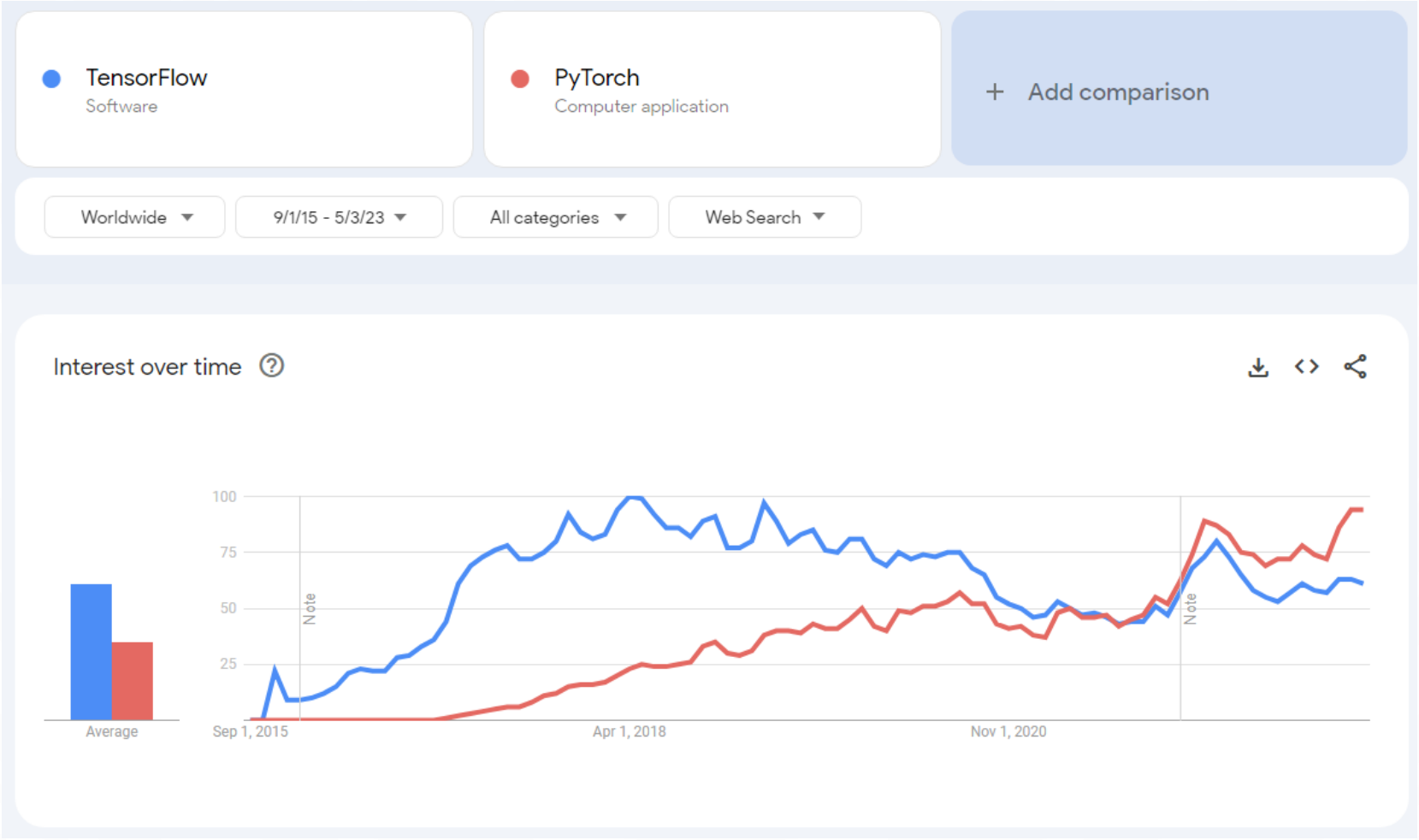
How a neural network works



Deep Learning Libraries

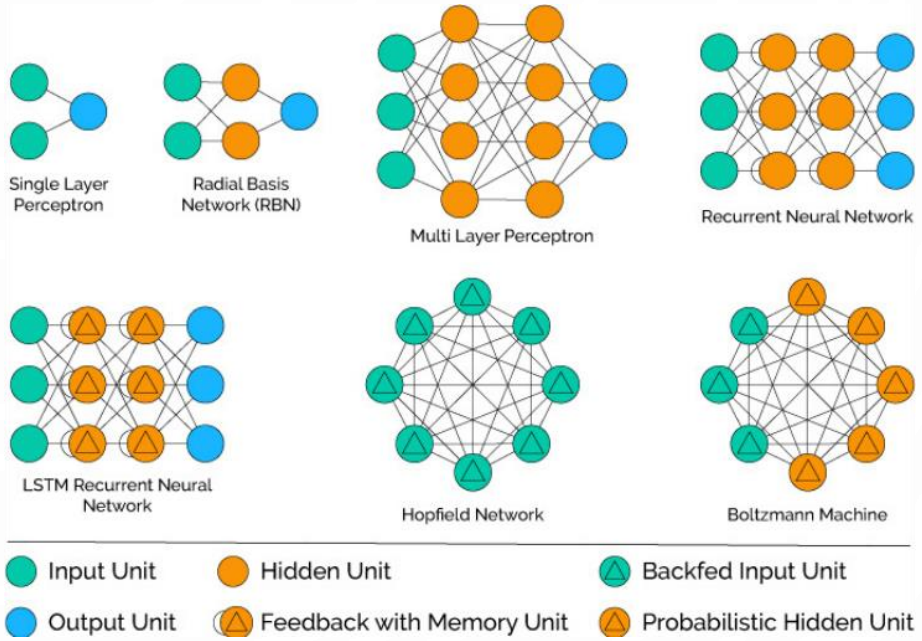
- **TensorFlow** is one of the Python libraries for deep learning applications.
  - Developed by the Google Brain Team
  - Provides a wide range of flexible tools, libraries, and community resources.
- **PyTorch** is a machine learning framework used for applications such as computer vision and natural language processing
  - Originally developed by Meta AI and now part of the Linux Foundation umbrella.
  - It is free and open-source software released under the modified BSD license.
  - The name of the library is derived from Torch, which is a deep learning framework written in the Lua programming language.

Deep Learning Libraries



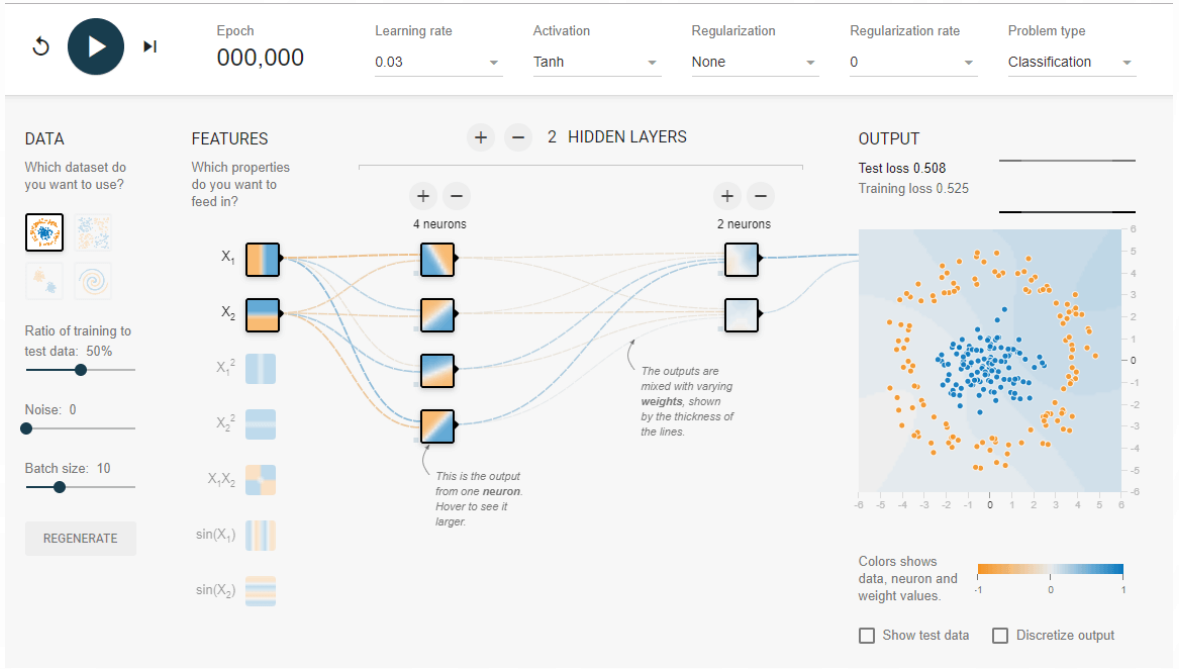
Types of Neural Networks

- Perceptron
- Multilayer Perceptron
- Feed Forward Neural Network
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- …



TensorFlow Playground

- TensorFlow Playground is a web app that allows users to test the artificial intelligence (AI) algorithm with TensorFlow machine learning library for learning purposes.
  - http://playground.tensorflow.org



TensorFlow Example

- https://colab.research.google.com/drive/1Vs5BArfH0r8DuNtPDHiK4KsnBb2V2dIf?usp=sharing