Artificial Neural Network

1. Introduction to ANN

1.1. Definition

An Artificial Neural Network (ANN) is a computational model that mimics the way of nerve cells work in the human brain.

1.2. History

In 1949, Donald Hebb published "The Organization of Behavior," which illustrated a law for synaptic neuron learning. This law, later known as Hebbian Learning in honor of Donald Hebb, is one of the most straight-forward and simple learning rules for artificial neural networks.

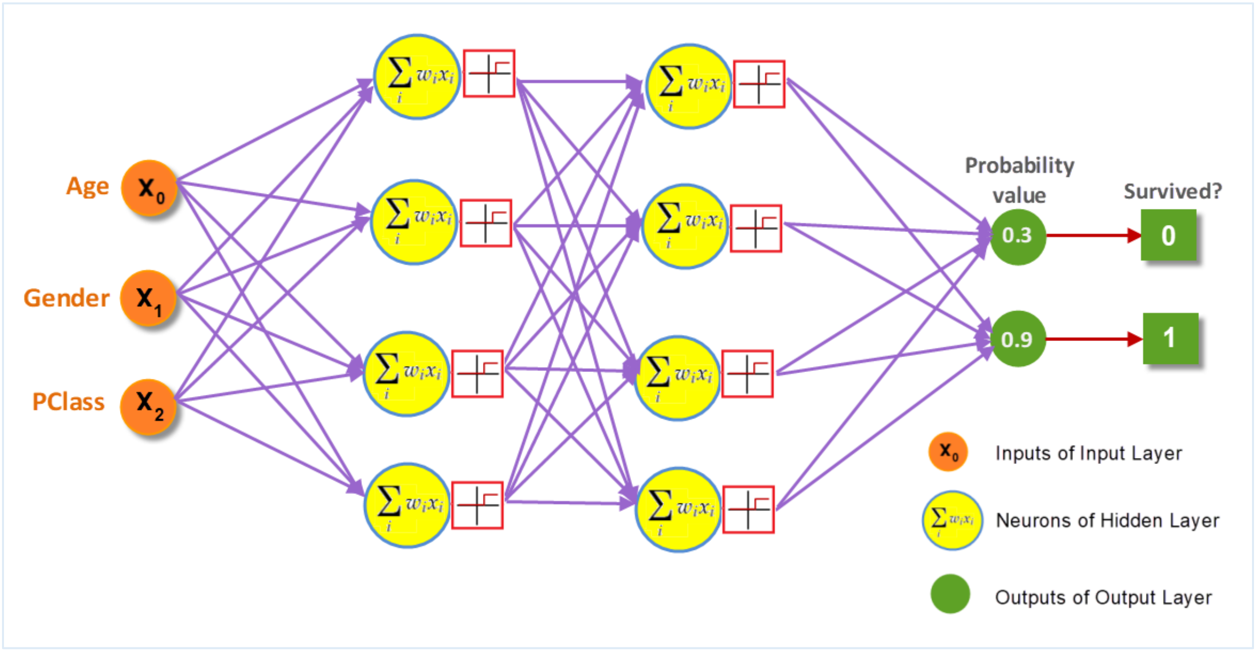
In 1951, Narvin Minsky made the first Artificial Neural Network (ANN) while working at Princeton.

In 1958, "The Computer and the Brain" were published, a year after Jhon von Neumann's death. In that book, von Neumann proposed numerous extreme changes to how analysts had been modeling the brain.

2. ANN architecture

- Artificial Neural Networks (ANNs) are a fundamental building block of deep learning.
- ANNs are inspired by the structure and function of the human brain, with interconnected nodes that mimic neurons and layers that perform specific tasks.

Basic Architecture of an Artificial Neural Network



2.1. Key Components:

1. Neurons (Nodes):

- Neuron is a fundamental processing unit, inspired by biological neurons.
- Neuron taking input signals, perform calculations, and produce output signals.
- Neurons are arranged in layers.

2. Layers:

- Groups of interconnected neurons are called layer.
- Common types of layers in ANN:
  - ✓ Input layer: Receives raw input data.
  - ✓ Hidden layers: Intermediate layers for processing the data and perform feature extraction.
  - ✓ Output layer: Produces final results (predictions, classifications).

3. Connections (Weights):

- Numerical values (weights) representing the strength of connections between neurons.
- These weights adjusted during training to learn patterns in data.

4. Activation Function:

- Determines whether a neuron "fires" or not, based on its input.
- Introduces non-linearity, enabling complex patterns to be captured.
- Common examples: ReLU, sigmoid, tanh.

5. Training:

- Process of adjusting weights to minimize errors and achieve desired outcomes.

- Involves feeding training data through the network and updating weights using backpropagation.

2.2. Key Characteristics:

- Deep: Multiple hidden layers enabled to learning the complex representations.
- Non-linear: Activation functions allow for modeling the non-linear relationships.
- Adaptive: Weights are adjusted during training to fit the data.

2.3. Additional Considerations:

- Deeper Networks: Can learn more complex patterns, but require more data and computational resources.
- Regularization: Techniques to prevent overfitting and improve generalization.
- Optimization Algorithms: Used to efficiently update weights during training.
- Hyperparameter Tuning: Finding optimal settings for network architecture and training process.