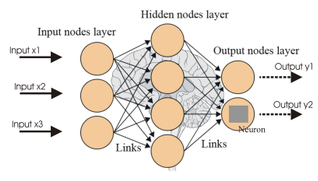
# Introduction to Neural Networks

## What?

- Subset of machine learning
- Heart of Deep Learning Algorithms of algorithms
- Inspired by the human brain structure




- consists of layers of interconnected nodes, or neurons
- each connection represents a weight
- network learns by adjusting these weights based on the errors it makes in predictions, using a process called backpropagation
- Input Layer: Receives the initial data.
- Hidden Layers: Perform computations and feature extraction.
- Output Layer: Produces the final prediction."

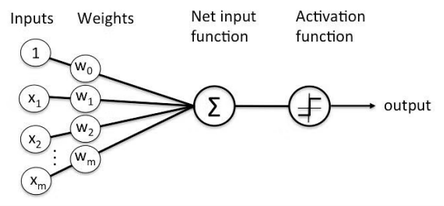
## Perceptrons

- Simplest type of artificial neural network
- Consists of a single neuron and is used for binary classification tasks.
- The perceptron can be mathematically represented as:

$$y = f\left(\sum_{i=1}^{n} w_i \cdot x_i + b\right)$$

where:

- $w_i$ are the weights
- $x_i$ are the input features
- $b$ is the bias
- $f$ is the activation function, typically a step function."




## Activation Functions

- crucial in neural networks as they introduce non-linearity
- allow neural networks to learn complex patterns

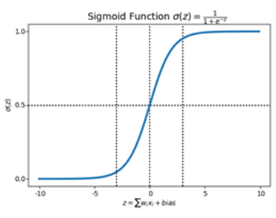
### Types

#### Sigmoid

The sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- input values: (0, 1)
- suitable for binary classification



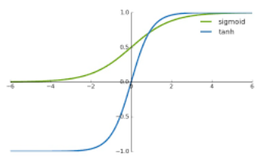

- Graph

#### Tanh

The tanh function is defined as

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- input values to the range (-1, 1)
- useful for hidden layers.




#### ReLU

The ReLU (Rectified Linear Unit) function is defined as:

$$\mathrm{ReLU}(x) = \max(0, x)$$

- outputs the input directly if it is positive; otherwise, it outputs zero
- used in deep learning due to its computational efficiency

## Backpropagation

- algorithm used to train neural networks
- calculates the gradient of the loss function
- updates the weights to minimize the loss.
- Forward Pass: "In the forward pass, we compute the output of the network."
- In the backward pass, we compute the gradient of the loss

Mathematical Formulas

- **Loss Function**:

$$L(y, \hat{y}) = \text{loss between actual and predicted values}$$

- **Gradient Calculation**:

$$\frac{\partial L}{\partial w_i}$$

- **Weight Update**:

$$w_i = w_i - \eta \cdot \frac{\partial L}{\partial w_i}$$

- $\eta$: Learning rate




## Project: MNIST Classification with Neural Network

- Data Loading and Preprocessing
- Model Building
- Model Compilation
- Model Training
- Model Evaluation

Presented with xmind