Soft Computing - Module 3

Neural Networks

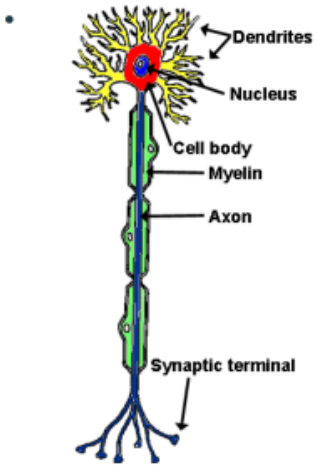
Introduction

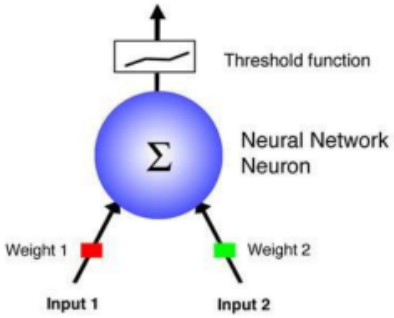
- A neural network is a type of machine learning algorithm that is modelled after the structure and function of the human brain.
- It is a collection of interconnected nodes, called neurons, that work together to process and classify data.
- At the core of a neural network is a mathematical model that is used to make predictions or decisions based on input data.
- This model is typically composed of several layers of neurons, each of which performs a specific function.
- The first layer of neurons is responsible for receiving input data, and subsequent layers process this data to produce an output.
- The connections between neurons are weighted, allowing the network to learn and adapt to new information.
- These weights are initially set randomly, but are adjusted during training using an optimization algorithm such as gradient descent.
- The goal of training is to find the set of weights that produce the most accurate predictions on a given dataset.
- There are several types of neural networks, including feedforward neural networks, convolutional neural networks, and recurrent neural networks. Each type is designed to handle a specific type of data, such as images or time-series data.

(refer ANN from mod 1)

Similarities between Biological Neural Network and Artificial Neural Network

Neurons



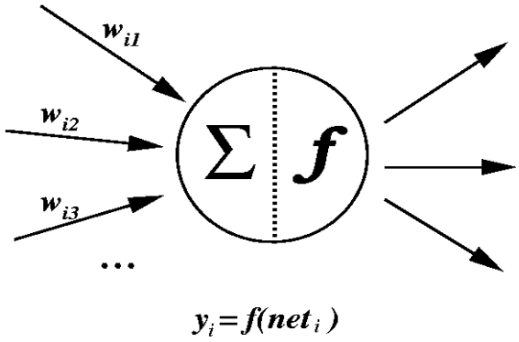


Criteria	BNN	ANN
Processing	Massively parallel, slow but superior than ANN	Massively parallel, fast but inferior than BNN
Size	1011 neurons and 1015 interconnections	102 to 104 nodes (mainly depends on the type of application and network designer)
Learning	They can tolerate ambiguity	Very precise, structured and formatted data is required to tolerate ambiguity
Fault tolerance	Performance degrades with even partial damage	It is capable of robust performance, hence has the potential to be fault tolerant
Storage capacity	Stores the information in the synapse	Stores the information in continuous memory locations

Types of learning in Neural Networks

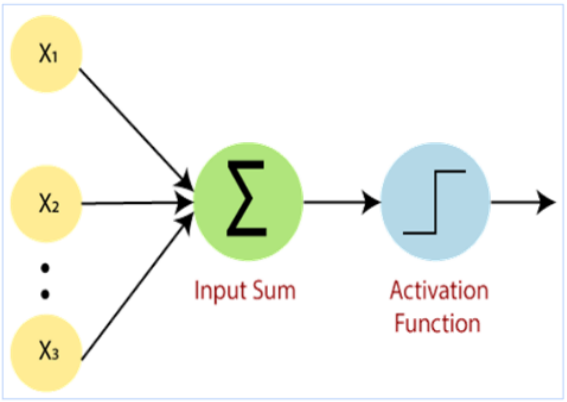
- Supervised
- Unsupervised
- Reinforcement

Model of an ANN:



- Receives n-inputs
- Multiplies each input by its weight
- Applies activation function to the sum of results
- Outputs result

Single Layer Perceptron



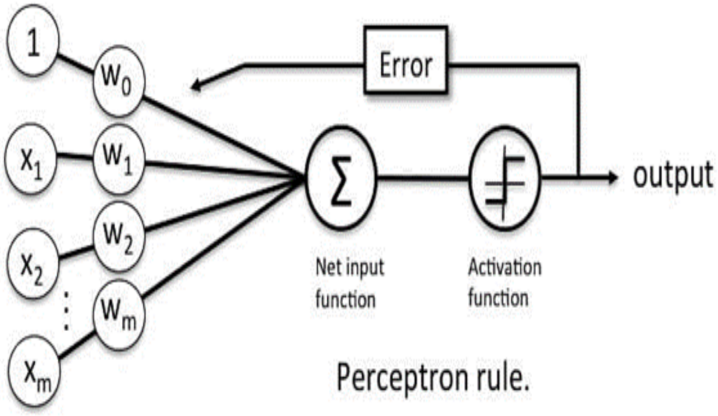
- A single-layer perceptron is a type of artificial neural network that consists of only one layer of artificial neurons.
- It is the simplest type of neural network
- Single layer perceptron has been used in various applications, including: Pattern Recognition, Binary Classification, , Control Systems, Medical Diagnosis, Financial Forecasting

- The perceptron consists of 4 parts
  - **Input value or One input layer:** The input layer of the perceptron is made of artificial input neurons and takes the initial data into the system for further processing.
  - **Weights and Bias:**
    - **Weight**: It represents the dimension or strength of the connection between units.
    - **Bias**: It is the same as the intercept added in a linear equation. bias is a tunable parameter in neural networks that can help improve the

accuracy and flexibility of the model by allowing it to learn more complex decision boundaries.

- **Net sum**: It calculates the total sum.
- **Activation Function**: A neuron can be activated or not, is determined by an activation function. The activation function calculates a weighted sum and further adds bias with it to give the result.

The Perceptron Learning Rule



1. **Initialize the weights**: Start with random weights for each input.
2. **Input the training data**: Input the features into the perceptron and calculate the output.
3. **Calculate the error**: Compare the predicted output with the desired output to calculate the error.
4. **Update the weights**: Adjust the weights of the inputs based on the error. If the predicted output is less than the desired output, increase the weights of the inputs. If the predicted output is greater than the desired output, decrease the weights of the inputs. The magnitude of the weight adjustment is proportional to the error and the input value.

Repeat: Repeat steps 2 to 4 until the error is minimized or a maximum number of iterations is reached

McQuillans Model v/s Single-layer perceptron

McQuillans Model :

- McQuillan's model, also known as the cascade-correlation architecture, is a type of feedforward neural network that is designed to automatically learn the structure of the network during training.
- The model starts with a small number of hidden units and then adds additional hidden units one at a time as needed to improve the accuracy of the network.
- This allows the model to automatically adapt to the complexity of the data, and it can often achieve higher accuracy than fixed-topology networks.
- Another important difference between McQuillan's model and a single-layer perceptron is that McQuillan's model uses a nonlinear activation function, such as the sigmoid or ReLU function, in the hidden units. Nonlinear activation functions allow the network to learn more complex relationships between the inputs and outputs, and they are essential for modeling nonlinear problems.

Single Layer Perceptron

- In contrast, a single-layer perceptron is a simple feedforward neural network architecture that consists of a single layer of neurons connected to an output layer.
- It is designed to learn linear decision boundaries for classification tasks, and it can be trained using the perceptron learning algorithm.
- However, it has limited representational power and cannot learn more complex decision boundaries.
- a single-layer perceptron uses a linear activation function.

Overall, while both McQuillan's model and single-layer perceptron are neural network architectures, they differ in terms of their ability to learn complex representations and adapt to the complexity of the data. McQuillan's model is a more flexible and powerful architecture, while a single-layer perceptron is a simpler architecture that is better suited for simpler classification tasks.

Multi-Layer Perceptron (MLP)

- A Multi-Layer Perceptron (MLP) is a type of neural network that consists of multiple layers of artificial neurons.
- MLPs are also known as feedforward neural networks.
- The architecture of an MLP consists of an input layer, one or more hidden layers, and an output layer.
- Each layer is composed of multiple artificial neurons that compute a weighted sum of the input signals and apply an activation function to produce an output signal.
- The hidden layers in an MLP are responsible for extracting features from the input data and transforming them into a format that is suitable for the output layer.
- The output layer produces the final output of the network, which can be binary or continuous.
- The learning process of an MLP involves adjusting the weights of the input signals using backpropagation.
- Backpropagation allows the MLP to learn from the training data and improve its performance over time.

Comparing Single Layer and Multilayer perceptron

	Single Layer Perceptrons	Multi Layer Perceptrons
Architecture	Only one layer of neurons that directly connects to input data	Multiple layers of neurons, including one or more hidden layers that lie between the input and output layers.
Capabilities	Limited to linearly seperable problems, meaning they can only learn and classify data that can be separated by a single straight line.	Can learn and classify non-linearly separable problems by using hidden layers to solve complex problems
Training	Uses a simple learning rule called the **Perceptron Learning Algorithm**, which adjusts the weights of the input signals to minimize the error between the predicted and actual output	Uses a more complex learning algorithm called **backpropagation**, which iteratively adjusts the weights of all the neurons in the network to minimize the error between the predicted and actual output.
Applications	Simple binary classification problems, such as predicting whether an email is binary or not	Wide range of applications, including image and speech recognition, NLP and financial forecasting

Back Propogation

Need?

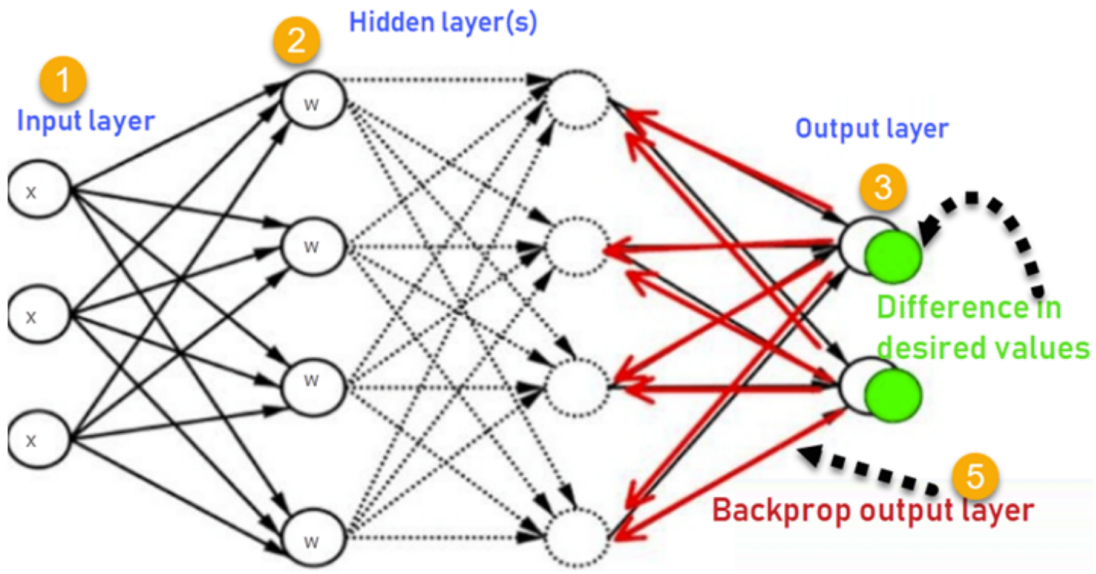
- Standard method that generally works well
- Doesnot require prior knowledge about the network
- Does not need special mention of the features
- Very flexible
- No parameters to tune apart from the numbers of the input
- fast, simple and easy to program

How does it work?

1. Inputs X, arrive through the preconnected path.

2. Input is modeled using real weights W. The weights are usually randomly selected.
3. Calculate the output for every neuron
4. Calculate the error in the outputs

   Error= Actual Output – Desired Output
5. Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.



- Back Propagation are supervised learning algorithms used for training neural networks.
- The basic structure of a backpropagation network consists of an input layer, one or more hidden layers, and an output layer.
- Each layer is composed of one or more neurons, which receive inputs, process them, and pass the outputs to the next layer.
- The connections between the neurons are weighted, and these weights are adjusted during training to improve the accuracy of the network's predictions.
- During the training process, the network is fed a set of input-output pairs, and the output of the network is compared to the desired output.
- The error between the actual output and the desired output is then back propagated through the network, and the weights are adjusted to reduce the

error.

- This process is repeated many times, with the hope that the network will eventually converge to a set of weights that produces accurate predictions for new input data.
- If the predicted output is less than the desired output, increase the weights of the inputs.
- If the predicted output is greater than the desired output, decrease the weights of the inputs.
- The magnitude of the weight adjustment is proportional to the error and the input value.

Examples of Neural Networks

1. Character Recognition Application

- Character recognition is a common application of neural networks, and **can be achieved using various types of neural networks**, including Feedforward neural networks, convolutional neural networks, and recurrent neural networks
- The network must be **trained on a dataset of labeled character** images in order to learn to recognize characters.
- During training, **the network adjusts its weights based on the difference between its predicted output and the true label of the input image**.
- Once the network is trained, it can be used to make predictions on new, unlabeled character images.

Optical Character Recognition

- OCR is a technology that **analyzes the text of a page and turns the letters into code** that may be used to process information.
- OCR is a technique for **detecting printed or handwritten text characters** inside digital
- OCR systems **turn physical documents into machine-readable text.**
- These digital versions can be highly beneficial to children and young adults who struggle to read.

- The essential application of OCR is to convert hard copy legal or historical documents into PDFs.
- Steps of OCR
    - **Image Preprocessing**

      Pre-processing covers all those functions of feature extraction to produce a original image.

      The steps in pre-processing involves

        - Size normalization: Bicubic interpolation is used for standard sized image.
        - Binarization: it is process of converting a gray scale image into binary image by thresholding
        - Smoothing: the erosion and dilation smooth the Boundaries of objects.
    - **Text recognition**
    - **Pattern Matching**
    - **Feature Extraction**
    - **Post processing**
        - After analysis, the system converts the extracted text data into a computerized file. Some OCR systems can create annotated PDF files that include both the before and after versions of the scanned document.

2. Kohonen's Algorithm

- It is a type of unsupervised neural network that is used for clustering and visualization of high-dimensional data.
- Self Organising Map: A self organising map (SOM) **is an unsupervised neural network** that reduces the input dimensionality in order to represent its distribution as a map.
- SOMs can be used for various applications, such as data visualization, feature extraction, and clustering.
- They have been applied in fields such as image processing, speech recognition, and bioinformatics.
- Self Organising map has two layers one is input layer and the output layer.

- Unlike other ANN types, SOM doesn’t have activation function in neurons, we directly pass weights to output layer without doing anything.