MPSSDEGB
कौशल से रोजगार, समृद्धि अपार

FITT
Foundation For Innovation And Technology Transfer

SANKALP
Ministry of Skill Development
& Entrepreneurship

Fundamentals of Neural Networks (NN)

Contents

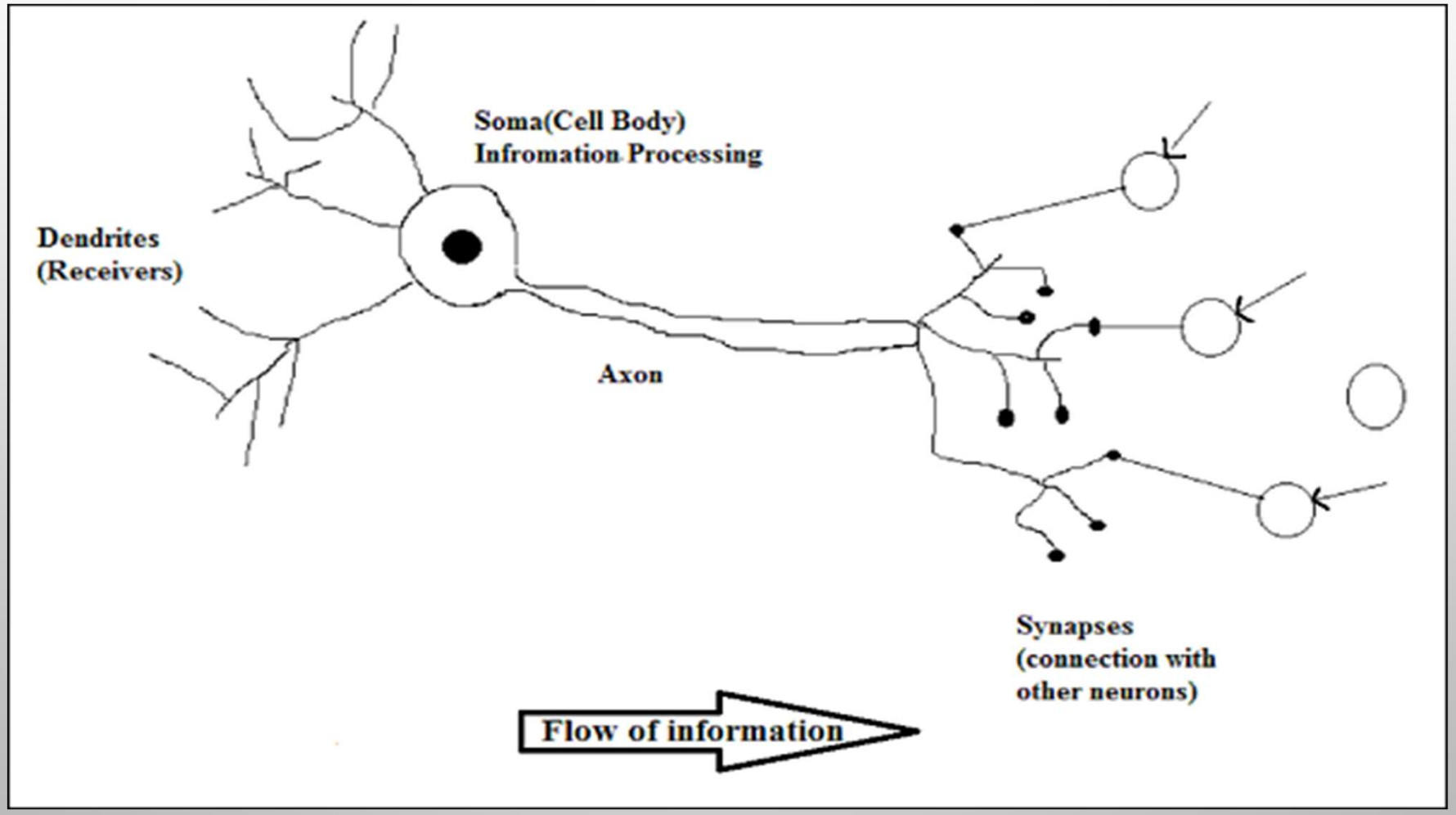
- Neural Network Concepts
- Neural Network Types
- Neural Network Applications
- Perceptrons in Neural Network
- Feedforward Neural Networks
- Feedforward NN structure and its Functionality
- Q&A and discussion.

Homework:

- Reading on the history and development of neural networks.

Neural Network Concepts

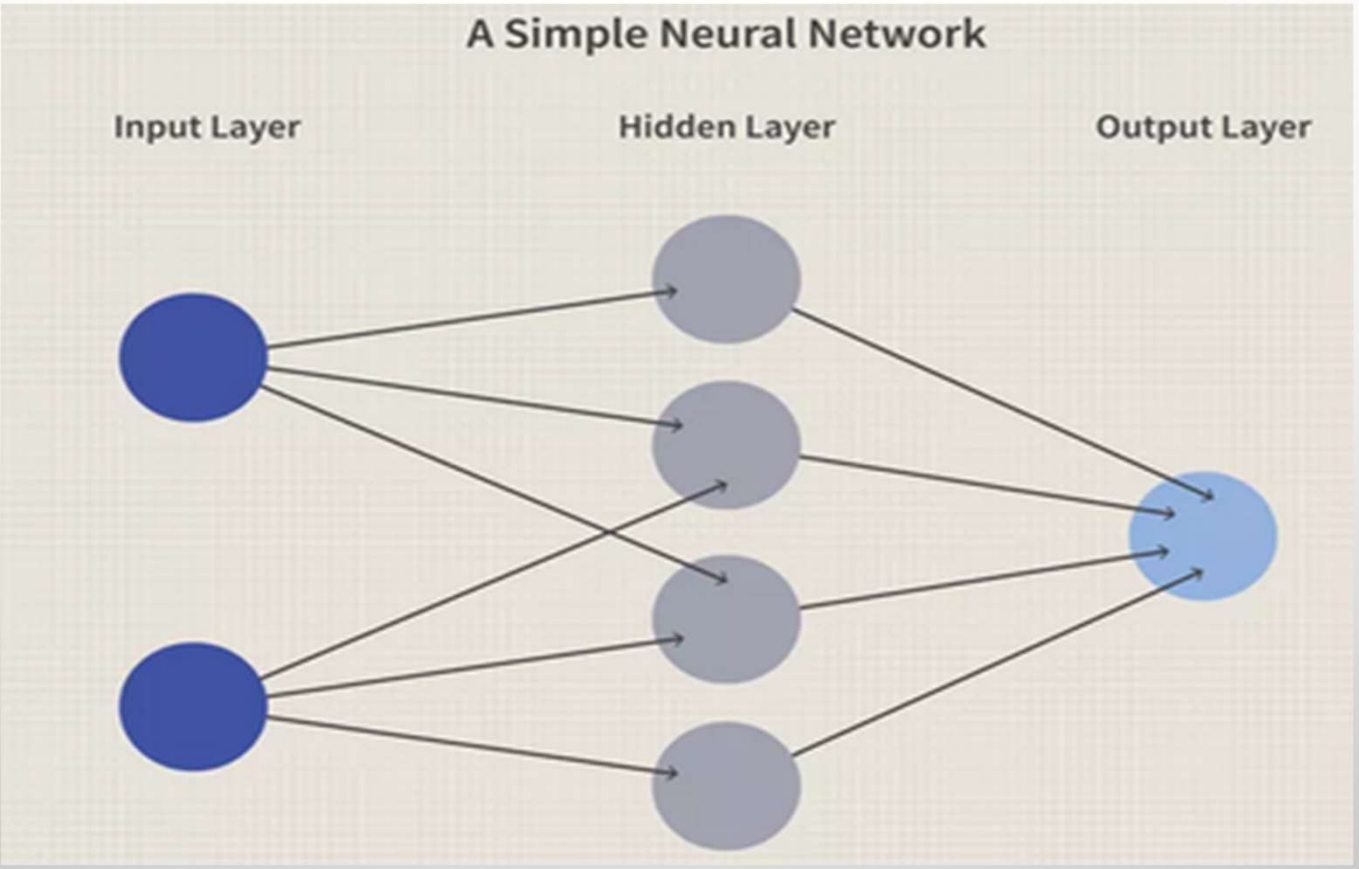
- ANN is an efficient computing system whose central theme is borrowed from the analogy of biological neural networks [2].



Neural Network Concepts

- ANN is an integral part of Artificial Intelligence (AI) and the foundation of Deep Learning (DL).
- ANN is the computational architecture consisting of neurons that mathematically represent how a biological neural network operates to identify and recognize relationships within the data.
- The main objective of ANN is to develop a system to perform various computational tasks faster than traditional systems.
- These tasks include pattern recognition, classification, optimization, data clustering, forecasting, marketing, fraud detection and risk assessment.

Neural Network Concepts



Neural Network Concepts

Biological Neural Network BNN	Artificial Neural Network ANN
Soma	Node
Dendrites	Input
Synapse	Weights or Interconnections
Axon	Output

BNN Vs ANN

Criteria	BNN	ANN
Processing	Massively parallel, slow but superior than ANN	Massively parallel, fast but inferior than BNN
Size	10^{11} neurons and 10^{15} interconnections	10^{2} to 10^{4} nodes *mainlydependsonthetypeofapplicationandnetworkdesigner*
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

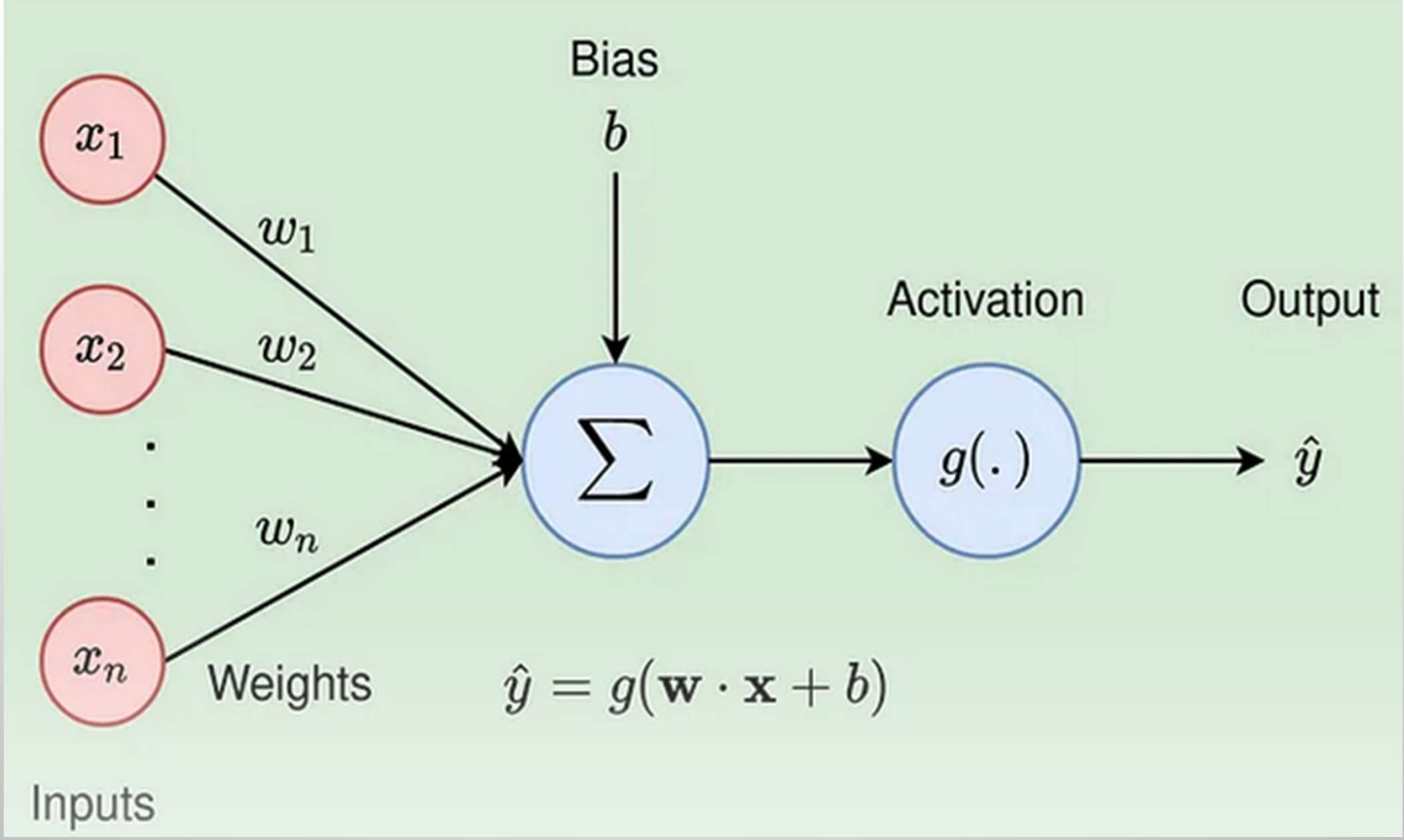
ANN during 1960s to 1980s

- **1961** − Rosenblatt made an unsuccessful attempt but proposed the "backpropagation" scheme for multilayer networks.
- **1964** − Taylor constructed a winner-take-all circuit with inhibitions among output units.
- **1969** − Multilayer perceptron MLP was invented by Minsky and Papert.
- **1971** − Kohonen developed Associative memories.
- **1976** − Stephen Grossberg and Gail Carpenter developed Adaptive resonance theory.

ANN during 1980s to 1990s

- **1982** – The major development was Hopfield's Energy approach.
- **1985** – Boltzmann machine was developed by Ackley, Hinton, and Sejnowski.
- **1986** – Rumelhart, Hinton, and Williams introduced Generalised Delta Rule.
- **1988** – Kosko developed Binary Associative Memory BAM and also gave the concept of Fuzzy Logic in ANN.

Mathematical Operations on one Neuron / Perceptron



Definitions

Neuron —It takes weighted values, performs mathematical calculation and produce output. It is also called a unit, node or **perceptron**.

Input — This is the data/values passed to the neurons.

Deep Neural Network (DNN) — This is an ANN with many hidden layers.

Weights — These values explain the strength (degree of importance) of the connection between any two neurons.

Definitions

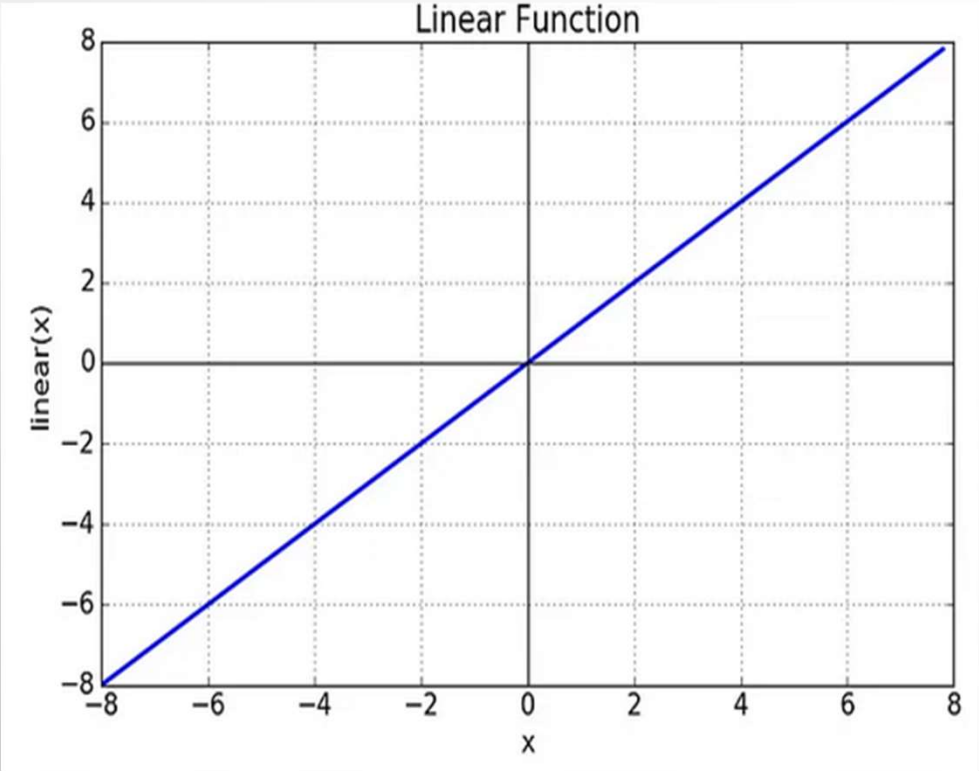
Bias — is a constant value added to each layer. It implies that even if there are no inputs, the model will be activated with a default value of the bias.

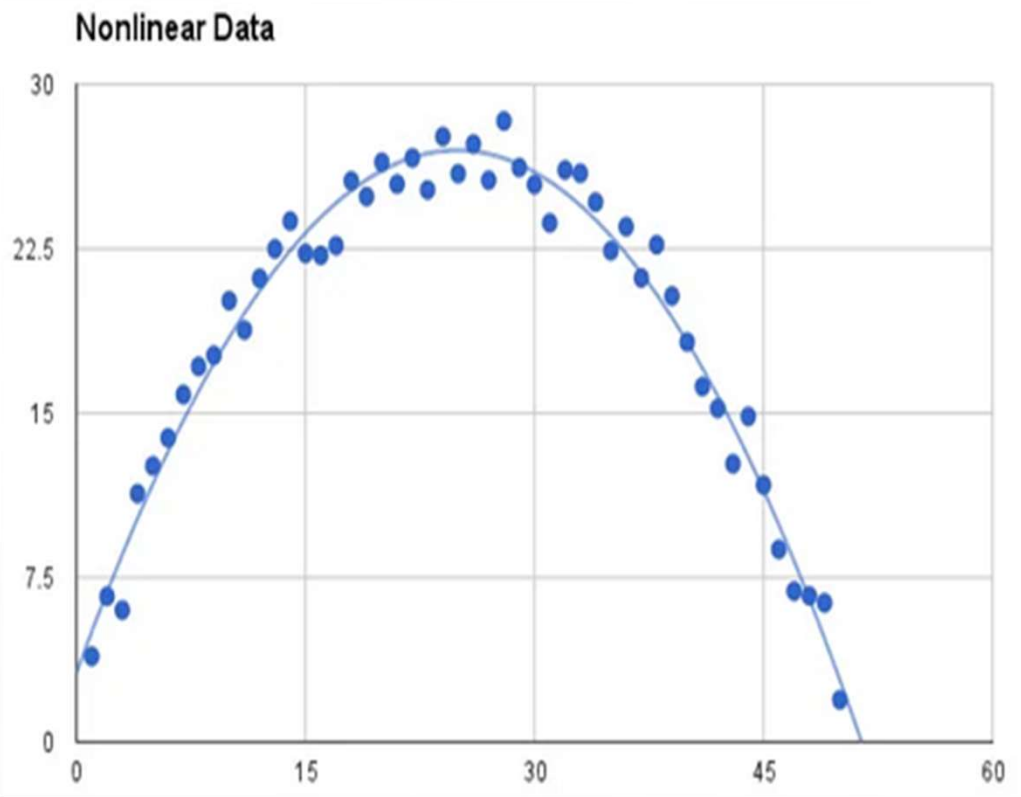
Activation function — It is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1 etc..

Note: weights and biases are the trainable parameters in NN, that is, the network learns patterns by tuning these parameters to get the best predictions.

Activation Functions [3]

- These can be basically divided into 2 types- [3]
  - Linear Activation Function
  - Non-linear Activation Functions





Activation Functions [3]

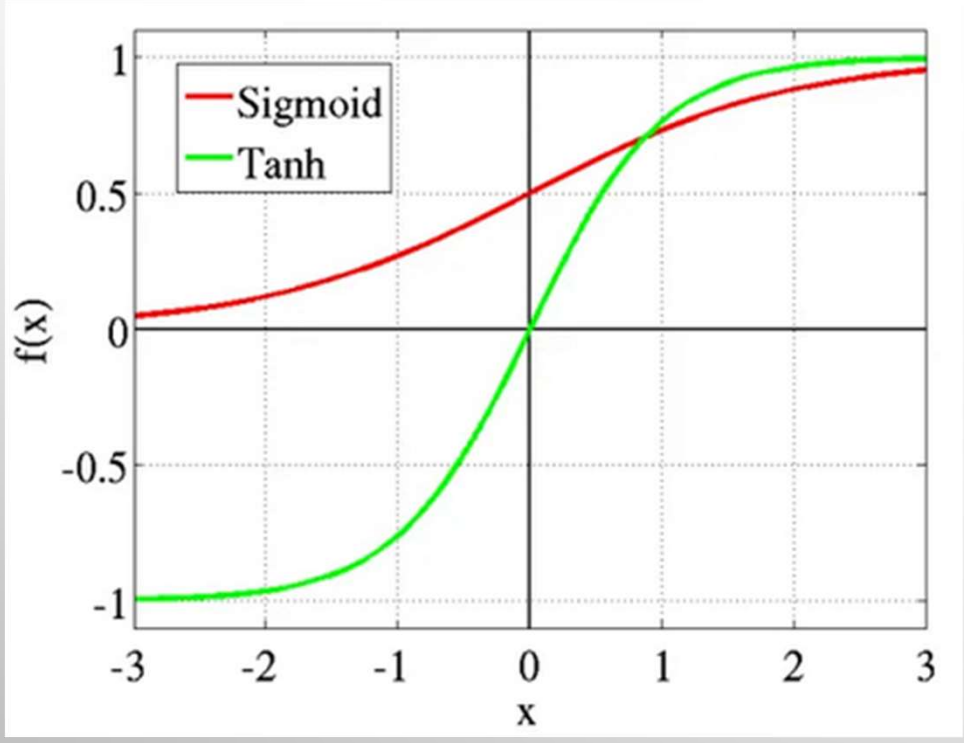
- The Nonlinear Activation Functions are mainly divided on the basis of their **range or curves**.

1. **Sigmoid or Logistic Activation Function**
2. **Tanh or hyperbolic tangent Activation Function**
3. **ReLU (Rectified Linear Unit) Activation Function**
4. **Leaky ReLU**
5. **Softmax**

- The range of the sigmoid function is from (0 to 1).
- The range of the tanh function is from (-1 to 1).
- The range of the ReLU function is from (0 to infinity).
- The range of the Leaky ReLU function is from (-infinity to infinity).
- The range of the softmax function is probability values from (0 to 1).

- **NOTE:** The ReLU is the most used activation function in the world right now. Since, it is used in almost all convolutional neural networks or DL.

Activation Functions [3]



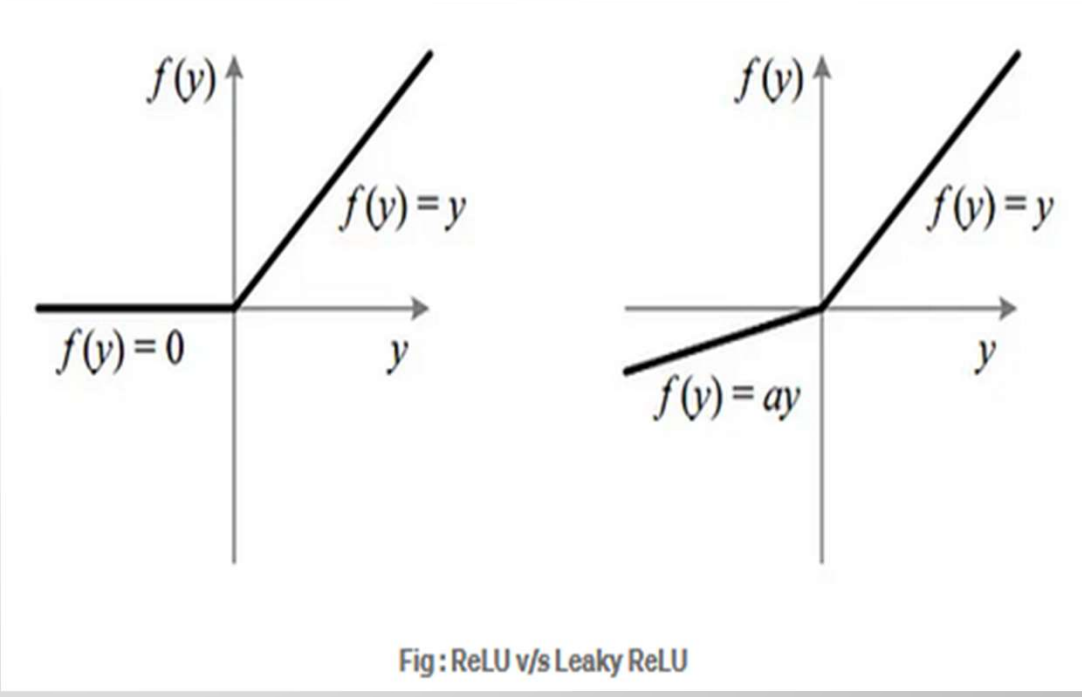
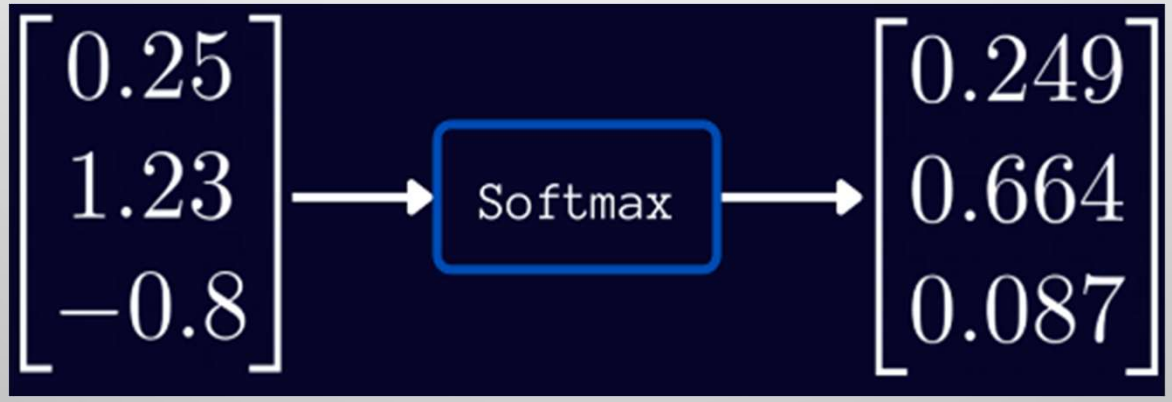


Fig : ReLU v/s Leaky ReLU

Softmax Activation Function [10]

- The softmax activation function takes in a vector of **raw outputs** of the neural network and returns a vector of **probability scores**.
- All entries in the softmax output vector are between 0 and 1.
- Notice how the entries of the softmax output sum up to 1: 0.664 + 0.249 + 0.087 = 1.
- This is useful for classification tasks.



Activation Functions [3]

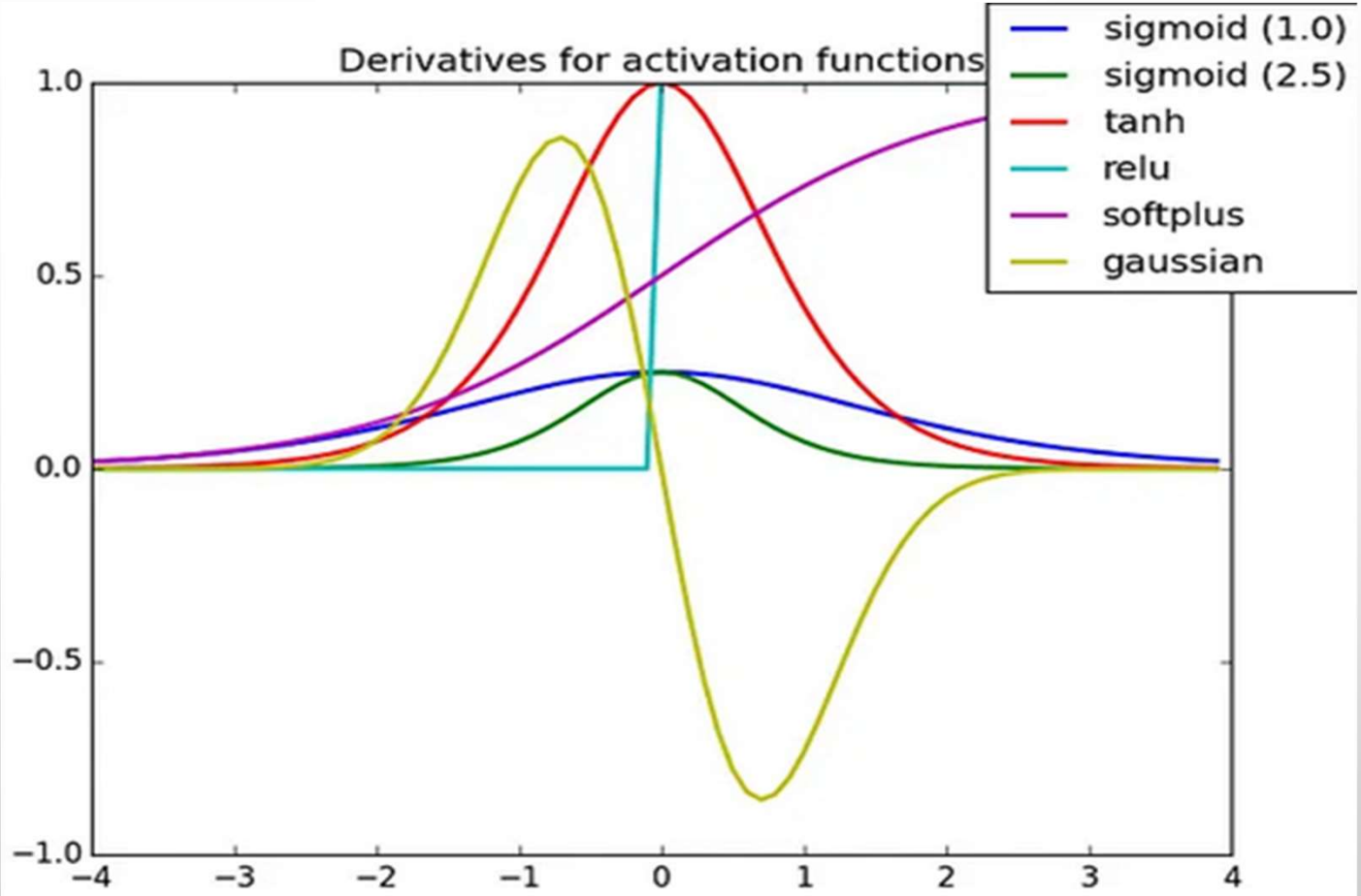


Fig: Derivative of Activation Functions

Neural Network Concepts [1]

Neural Networks

Pros

- Can often work more efficiently and for longer than humans
- Can be programmed to learn from prior outcomes to strive to make smarter future calculations
- Often leverage online services that reduce (but do not eliminate) systematic risk
- Are continually being expanded in new fields with more difficult problems

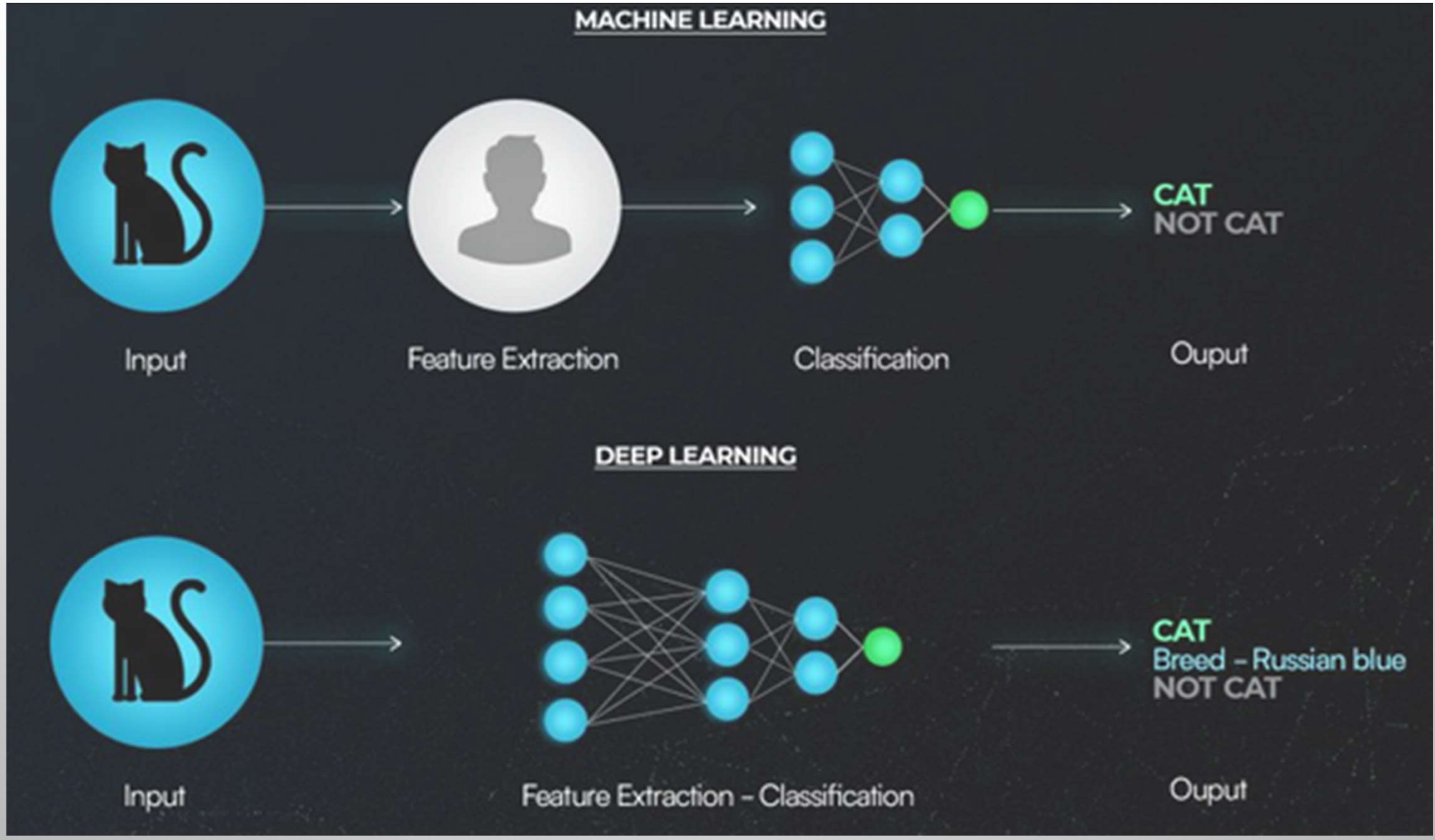
Cons

- Still rely on hardware that may require labor and expertise to maintain
- May take long periods of time to develop the code and algorithms
- May be difficult to assess errors or adaptions to the assumptions if the system is self-learning but lacks transparency
- Usually report an estimated range or estimated amount that may not actualize

Deep Learning (DL) [4]

- Deep Learning has neural networks which perform unsupervised learning from unstructured data.
- The learning can be either unsupervised, supervised, or semi-supervised in DL.
- The learning in this field is using the complex structure of artificial neural networks, which comprise multiple layers, making the learning 'deep' for solving specific problems.
- The DL models work on their own and automatically uncover patterns / features, unlike the ML models which need to manually extract features.

DL Versus ML [4]



DL Versus ML [4]

- To learn better and extract features from the data the DL models require large amounts of data.
- The DL nets have very high computing power and require GPU (Graphics Processing Unit) to run upon.

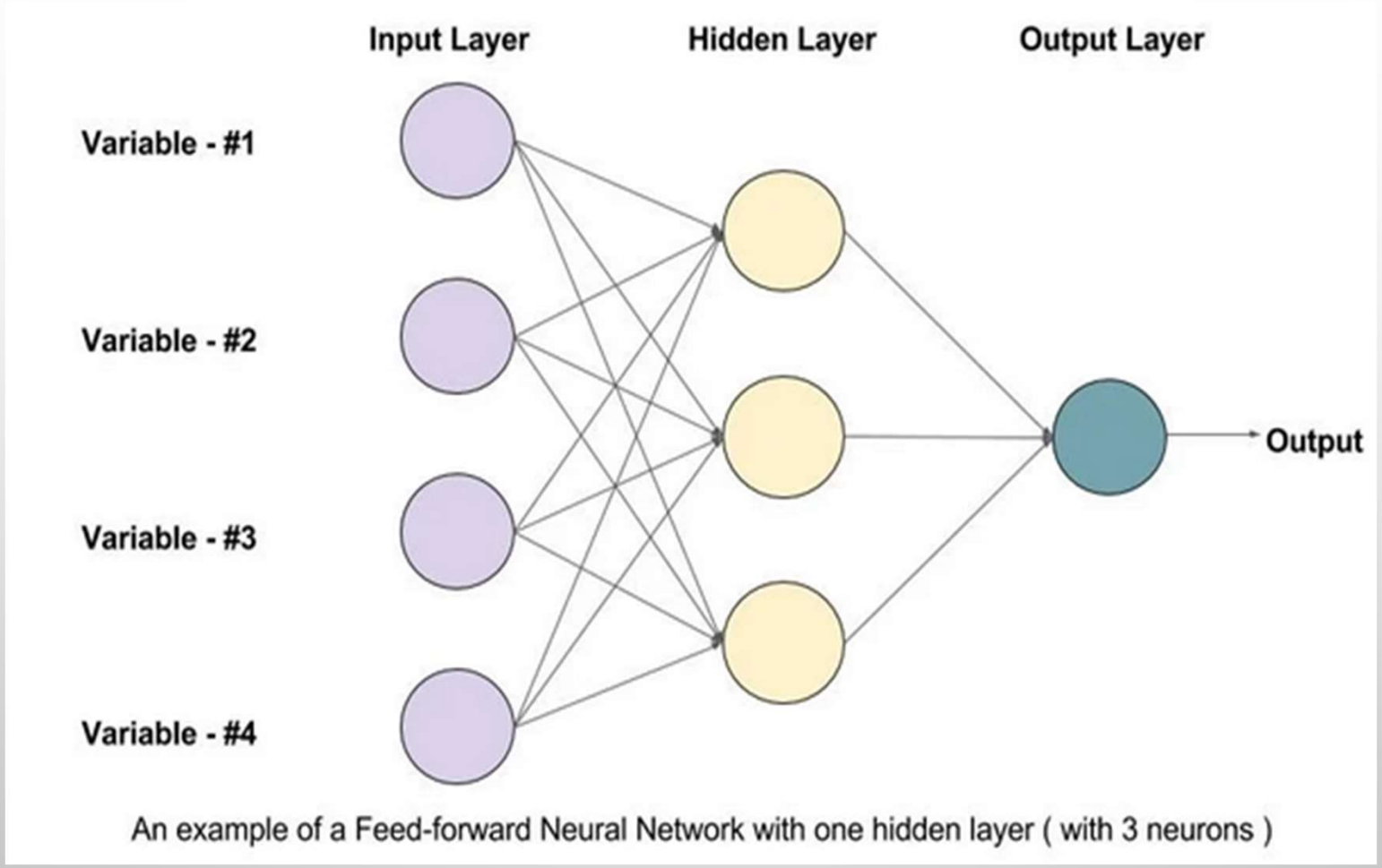
Types of Neural Networks [4]

- Feedforward Neural Network (FNN)
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Autoencoders

Feedforward Neural Network (FNN) [4]

- In FNN, the information travels only in one direction from the input to the output layer.
- Each of the layers is fully connected with all the nodes in the layer before and after.
- In this network, there may or may not be hidden layers.
- The network here does not form a loop.
- **Applications:** classification, face recognition, speech recognition, computer vision.

Feedforward Neural Network (FNN) [7]



An example of a Feed-forward Neural Network with one hidden layer (with 3 neurons)

Back Propagation Algorithm

It is the most widely used algorithm for training artificial neural networks.

The architecture of a neural network consists of some sequential layers, where the layer numbered i is connected to the layer numbered i+1.

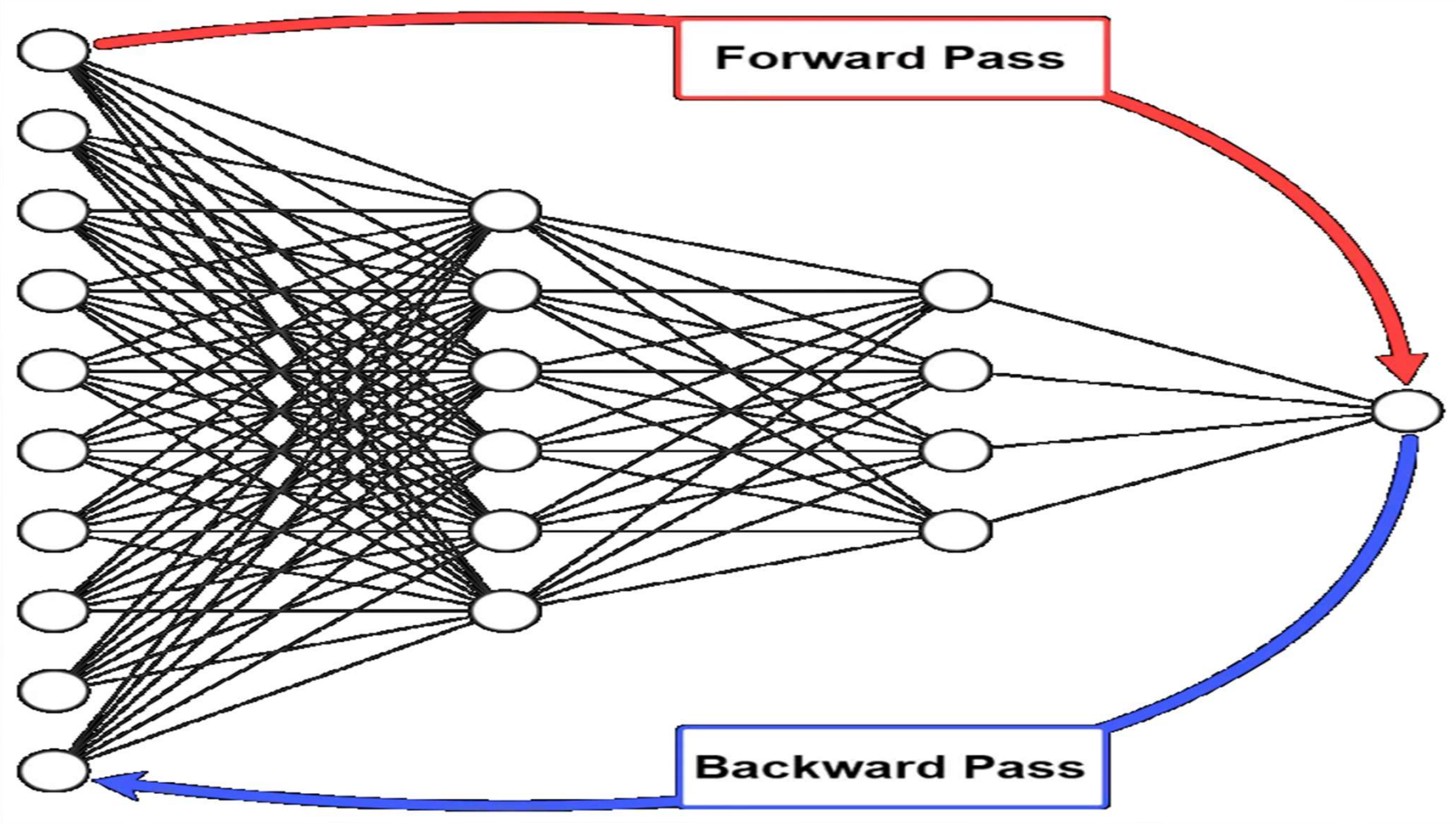
To train a neural network, there are 2 passes (phases):

- Forward
- Backward

The forward and backward phases are repeated from some epochs. In each epoch, the following occurs:

1. The inputs are propagated from the input to the output layer.
2. The network error is calculated.
3. The error is propagated from the output layer to the input layer.

Back Propagation Algorithm



Back Propagation Algorithm

Knowing that there's an error, what should we do? We should minimize the error.

To minimize network error, we must change something in the network.

Remember that the only parameters we can change are the weights and biases.

We can try different weights and biases, and then test our network.

Convolutional Neural Network (CNN) [4]

- The CNN is a variation of FNN, it uses minimal amounts of preprocessing.
- CNN is commonly used to extract information from unstructured data such as image and video data.
- **Applications:** video, signal, image recognition, visual imagery analysis, and recommender systems.
- It takes the inputs in batches like a filter, assigns relative importance to the weights and biases to various objects in the image, and differentiates one from the other.
- The network can remember the images in parts and compute the operations.

Recurrent Neural Network (RNN) [4]

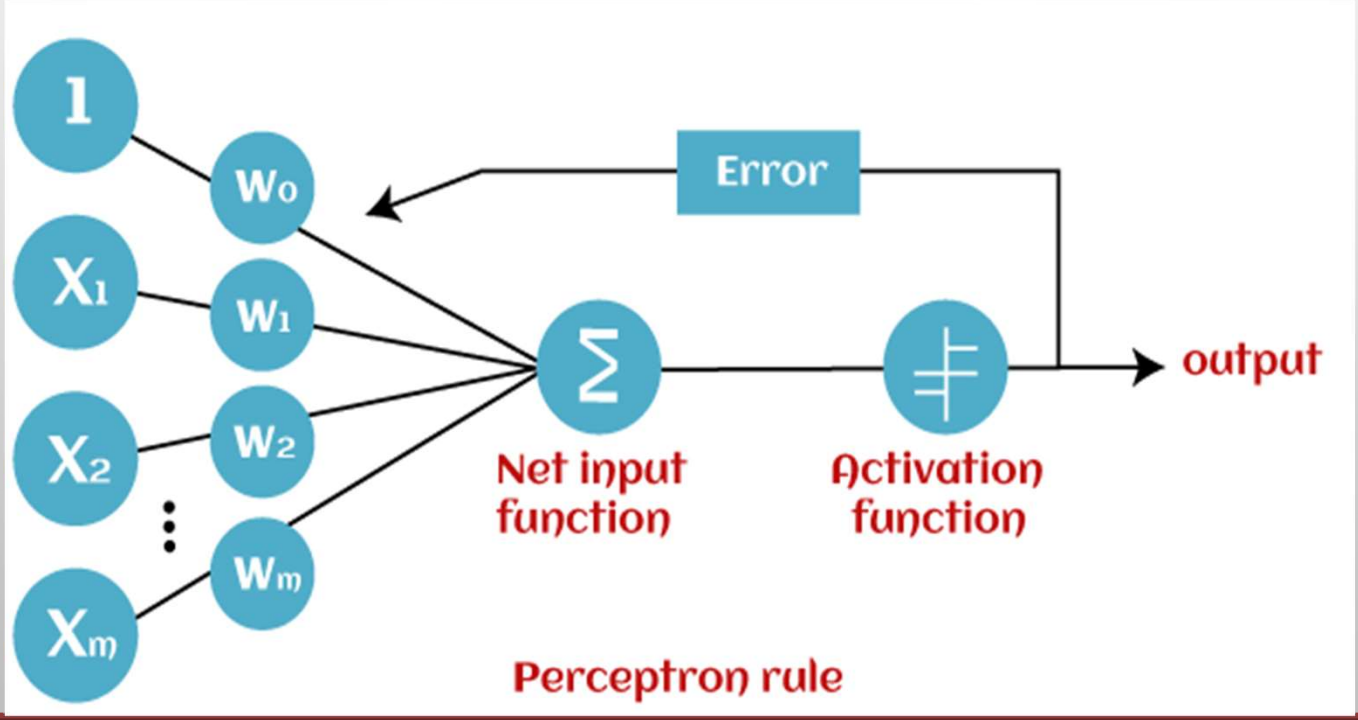
- The RNN stores the output of a layer and feeds it back to the input layer to predict the layer's outcome. (Forms the loop)
- This network uses the internal state (or memory) to process the variable-length sequences of inputs and hence can handle the arbitrary input or output lengths.
- **Applications:** To perform complex tasks, including providing image captions, time series forecasting, learning handwriting, language recognition, chatbots, fraud detection, detection of anomalies, and are most suitable to process sequences of inputs.

Autoencoders [4]

- The objective of an autoencoder is to learn a representation or encoding for a set of data in an unsupervised manner.
- These are mostly applicable for dimensionality reduction by training the network to ignore signal “noise” and learning generative data models.
- **Applications:** In image reconstruction and image colourization.

Perceptron [8]

- Perceptron is a building block of an Artificial Neural Network.
- The **Perceptron** defines the first step into **Neural Networks.**
- Mr. Frank Rosenblatt invented the perceptron model as a binary classifier which contains three main components. These are Input Nodes, (Weights, Bias), and Activation function.



The Perceptron Algorithm [9]

Frank Rosenblatt suggested this algorithm:

1. Set a threshold value
2. Multiply all inputs with its weights
3. Sum all the results
4. Activate the output

1. Set a threshold value:

- Threshold = 1.5

2. Multiply all inputs with its weights:

- x1 * w1 = 1 * 0.7 = 0.7
- x2 * w2 = 0 * 0.6 = 0
- x3 * w3 = 1 * 0.5 = 0.5
- x4 * w4 = 0 * 0.3 = 0
- x5 * w5 = 1 * 0.4 = 0.4

3. Sum all the results:

- 0.7 + 0 + 0.5 + 0 + 0.4 = 1.6 (The Weighted Sum)

4. Activate the Output:

- Return true if the sum > 1.5 ("Yes I will go to the Concert")

Types of Perceptron Models [8]

Based on the layers, Perceptron models are divided into two types.

Single-layer Perceptron Model
Multi-layer Perceptron model

The main objective of the single-layer perceptron model is to analyze the linearly separable objects with binary outcomes.

Multi-layer perceptron model also has the same model structure but has a greater number of hidden layers.

Single-Layer Perceptron Limitations [8]

.

A perceptron model has limitations as follows:

- The output of a perceptron can only be a binary number (0 or 1) due to the hard limit transfer function.
- Perceptron can only be used to classify the linearly separable sets of input vectors.
- If input vectors are non-linear, it is difficult to classify them properly.

Multi-Layer Perceptron Models [8]

.

The multi-layer perceptron model is also known as the Backpropagation algorithm, which executes in two stages as follows:

Forward Stage: Activation functions start from the input layer in the forward stage and terminate on the output layer.

Backward Stage: In the backward stage, weight and bias values are modified as per the model's requirement.

In this stage, the error (difference between actual output and calculated output) will be propagated back from the output layer to the hidden layers or input layer.

Multi-Layer Perceptron Model [8]

Advantages:

- Used to solve complex non-linear problems.
- It works well with both small and large input data.
- It helps us to obtain quick predictions after the training.
- It helps to obtain the same accuracy ratio with large as well as small data.

Disadvantages:

- Computations are difficult and time-consuming.
- It is difficult to predict how much the dependent variable affects each independent variable.
- The model's functioning depends on the quality of the training.

Day-6 Quiz-Link (10 Minuits)

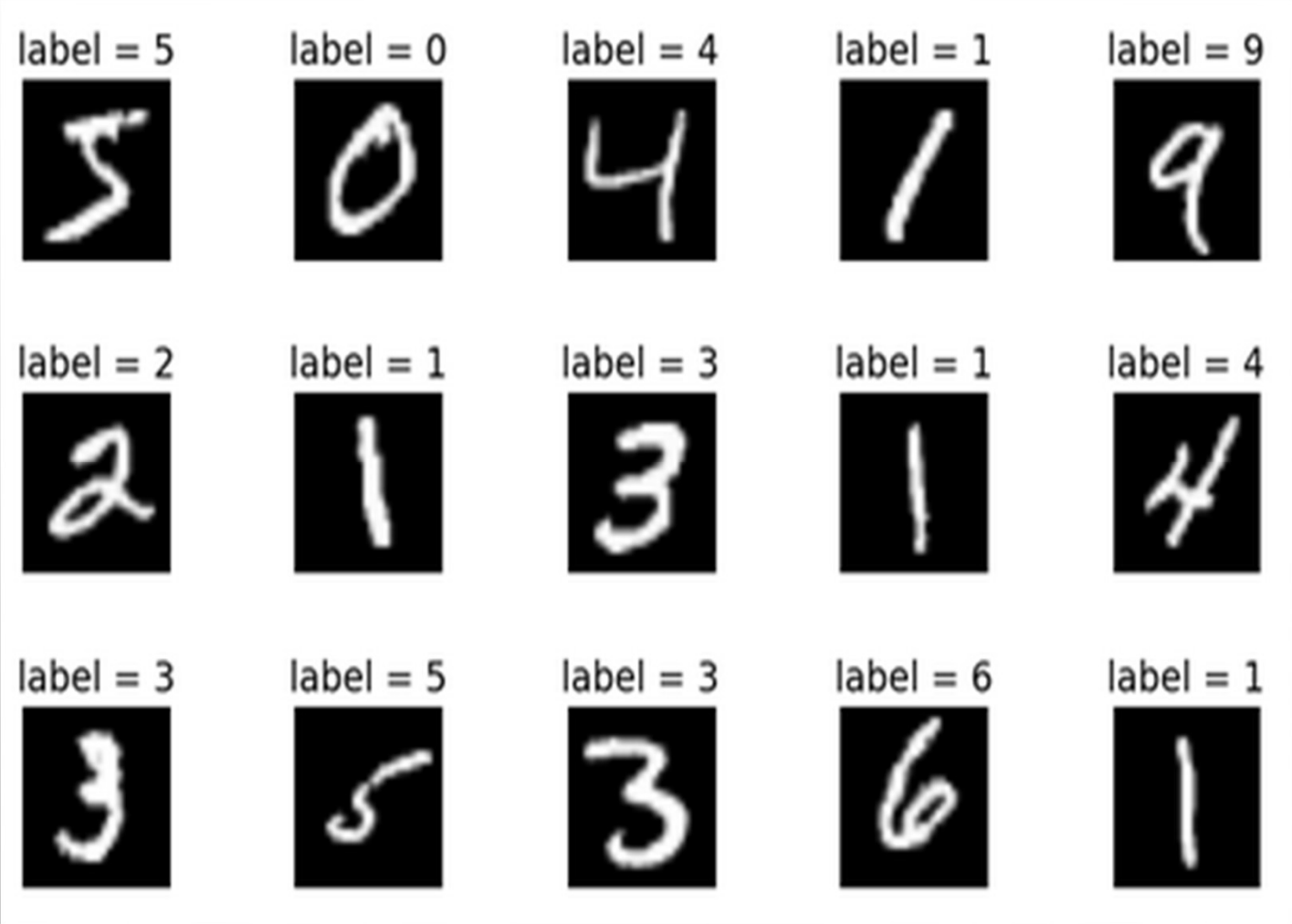
https://forms.gle/kircfneGTWxw6hjB7

Developing a Simple FNN Model

Now let's develop a simple Feedforward neural network model using Tensorflow.

- Initially, we declare the variable and assign it to the type of architecture which is a "Sequential()" in our case.
- Next, we directly add layers in a sequential manner using the model.add() method.
- The type of layer can be imported from tf. layers as shown in the code.
- We use adam function as an optimizer and crossentropy function as parameters for our architecture.
- Once the model is defined, the next step is to start the training process for which we will be using the model.fit() method.
- The evaluation will be done on the test dataset which can be called using model.evaluate() method.

MNIST Dataset for FNN



MNIST Dataset for FNN

- MNIST **contains 60,000 training images and 10,000 testing images of handwritten digits**.
- MNIST: (Modified National Institute of Standards and Technology database)
- The images are normalized to fit into a 28x28 pixel bounding.
- It is a standard dataset used in computer vision and deep learning.
- It was created as training dataset was collected from American Census Bureau employees, while the testing dataset was taken from American high school students.

Developing a Simple FNN Model [7]

https://colab.research.google.com/drive/1QsD5xBxcYlhx-Z28UMsLiKMXhbsROkeh?usp=sharing

Backpropagation Program

https://colab.research.google.com/drive/1sfsiO80o9zmDWjQZDcXq9vGv9egUQYcS?usp=sharing

Homework: Reading

History and Development of Neural Networks

1. https://www.youtube.com/watch?v=6fXNiJXUheI

2. https://www.youtube.com/watch?v=jz8hf5i_qDU

3. https://www.youtube.com/watch?v=YuZRiHWld-g

4. https://www.researchgate.net/publication/374723059_27-Artificial-Neural-Networks-History-and-State-of-the-Art

References

[1] https://www.investopedia.com/terms/n/neuralnetwork.asp
[2] https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_basic_concepts.htm
[3] https://towardsdatascience.com/the-basics-of-neural-networks-neural-network-series-part-1-4419e343b2b
[4] https://www.analytixlabs.co.in/blog/fundamentals-of-neural-networks/
[5] https://www.turing.com/kb/mathematical-formulation-of-feed-forward-neural-network
[6] https://pyimagesearch.com/2021/05/06/implementing-feedforward-neural-networks-with-keras-and-tensorflow/
[7] https://www.codingninjas.com/studio/library/intro-to-feedforward-neural-networks
[8] https://www.javatpoint.com/perceptron-in-machine-learning
[9] https://www.w3schools.com/ai/ai_perceptrons.asp
[10] https://www.pinecone.io/learn/softmax-activation/

THANK YOU