Deep Learning

Module – 7

Deep Learning

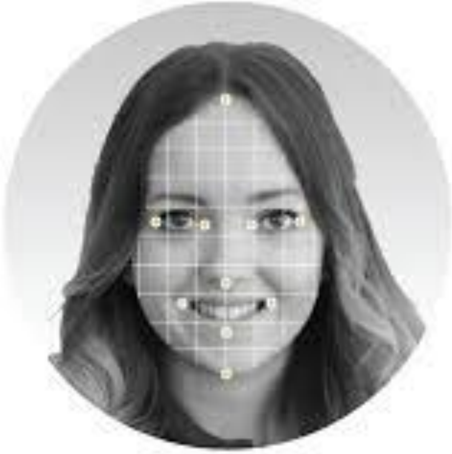
- Simple Feed Forward Networks
- Computational graphs for Deep Learning
- Convolutional Networks
- Recurrent Neural Networks
- Kernel Machines
- Hidden Markov Models.

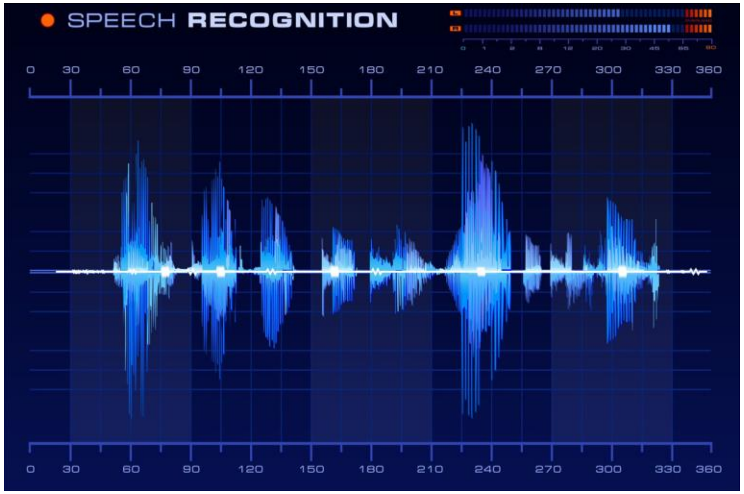
INTRODUCTION

What is Deep Learning?

Deep learning is a branch of machine learning that uses data, loads and loads of data, to teach computers how to do things only humans were capable of before.

For example, how do machines solve the problems of perception?





HISTORY

1958: Frank Rosenblatt creates the perceptron, an algorithm for pattern recognition.

1989: Scientists were able to create algorithms that used deep neural networks.

2000's: The term “deep learning” begins to gain popularity after a paper by Geoffrey Hinton.

2012: Artificial pattern-recognition algorithms achieve human-level performance on certain tasks.

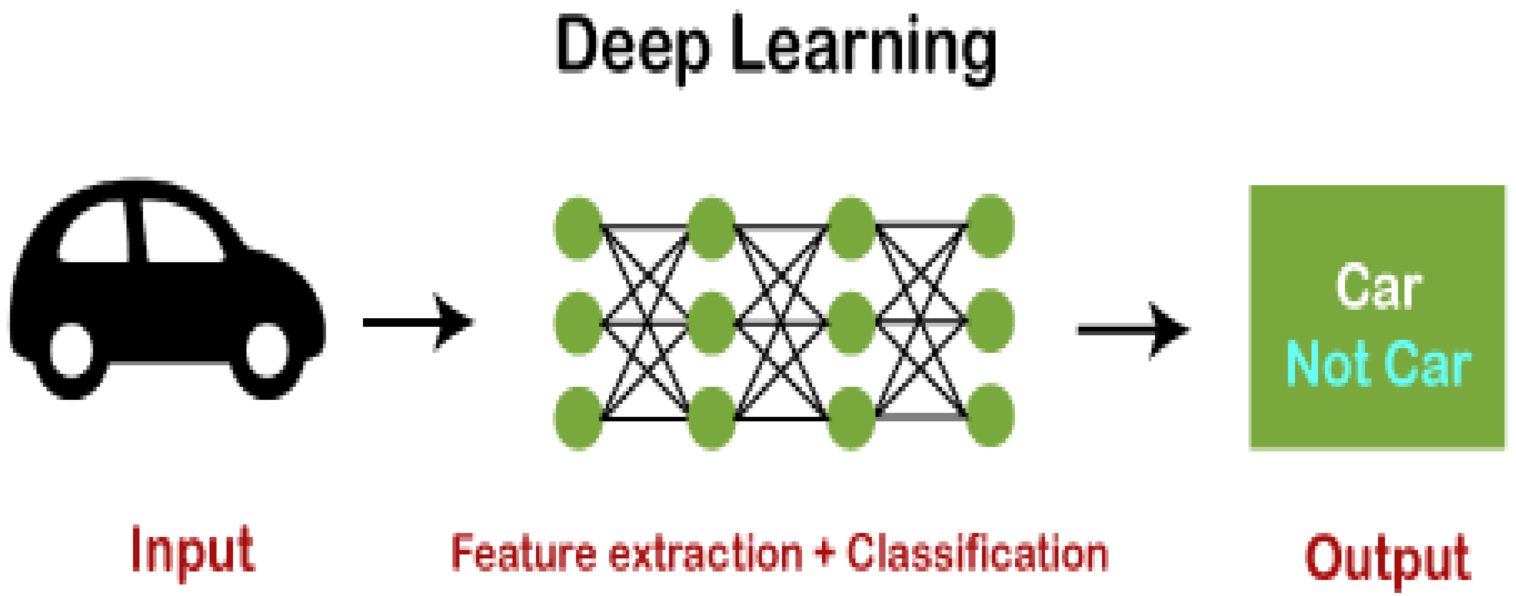
What is Deep Learning?

- *Deep Learning is the subset of machine learning or can be said as a special kind of machine learning*.
- It works technically in the same way as machine learning does, but with different capabilities and approaches.
- It is inspired by the functionality of human brain cells, which are called neurons, and leads to the concept of artificial neural networks.
- It is also called a deep neural network or deep neural learning.

Some popular deep learning models are:

- **Feed Forward network**
- **Convolutional Neural Network**
- **Recurrent Neural Network**
- **Autoencoders**
- **Classic Neural Networks, etc.**

- Consider the below image:



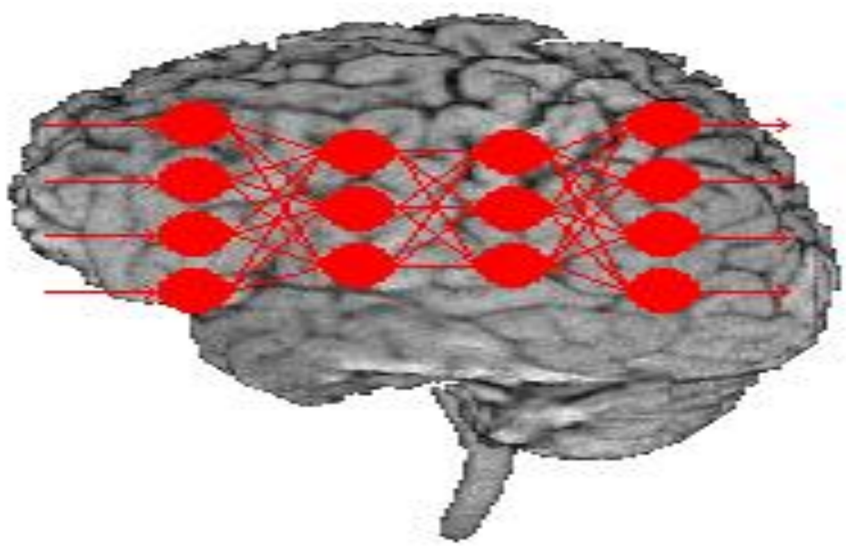
Key comparisons between Machine Learning and Deep Learning

Parameter	Machine Learning	Deep Learning
Data Dependency	Although machine learning depends on the huge amount of data, it can work with a smaller amount of data.	Deep Learning algorithms highly depend on a large amount of data, so we need to feed a large amount of data for good performance.
Execution time	Machine learning algorithm takes less time to train the model than deep learning, but it takes a long-time duration to test the model.	Deep Learning takes a long execution time to train the model, but less time to test the model.

Parameter	Machine Learning	Deep Learning
Hardware Dependencies	Since machine learning models do not need much amount of data, so they can work on low-end machines.	The deep learning model needs a huge amount of data to work efficiently, so they need GPU's and hence the high-end machine.
Feature Engineering	Machine learning models need a step of feature extraction by the expert, and then it proceeds further.	Deep learning is the enhanced version of machine learning, so it does not need to develop the feature extractor for each problem; instead, it tries to learn high-level features from the data on its own.
Problem-solving approach	To solve a given problem, the traditional ML model breaks the problem in sub-parts, and after solving each part, produces the final result.	The problem-solving approach of a deep learning model is different from the traditional ML model, as it takes input for a given problem, and produce the end result. Hence it follows the end-to-end approach.

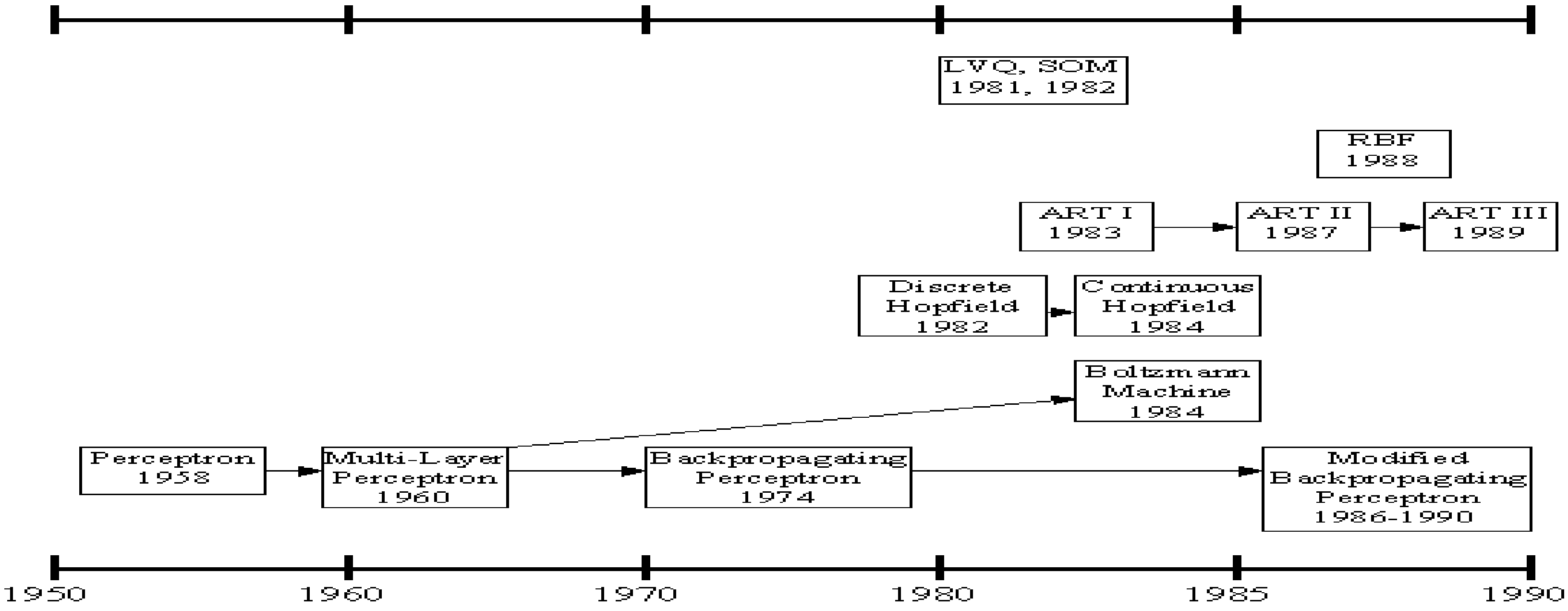
Parameter	Machine Learning	Deep Learning
Interpretation of result	The interpretation of the result for a given problem is easy. As when we work with machine learning, we can interpret the result easily, it means why this result occur, what was the process.	The interpretation of the result for a given problem is very difficult. As when we work with the deep learning model, we may get a better result for a given problem than the machine learning model, but we cannot find why this particular outcome occurred, and the reasoning.
Type of data	Machine learning models mostly require data in a structured form.	Deep Learning models can work with structured and unstructured data both as they rely on the layers of the Artificial neural network.
Suitable for	Machine learning models are suitable for solving simple or bit-complex problems.	Deep learning models are suitable for solving complex problems.

Artificial Neural Networks

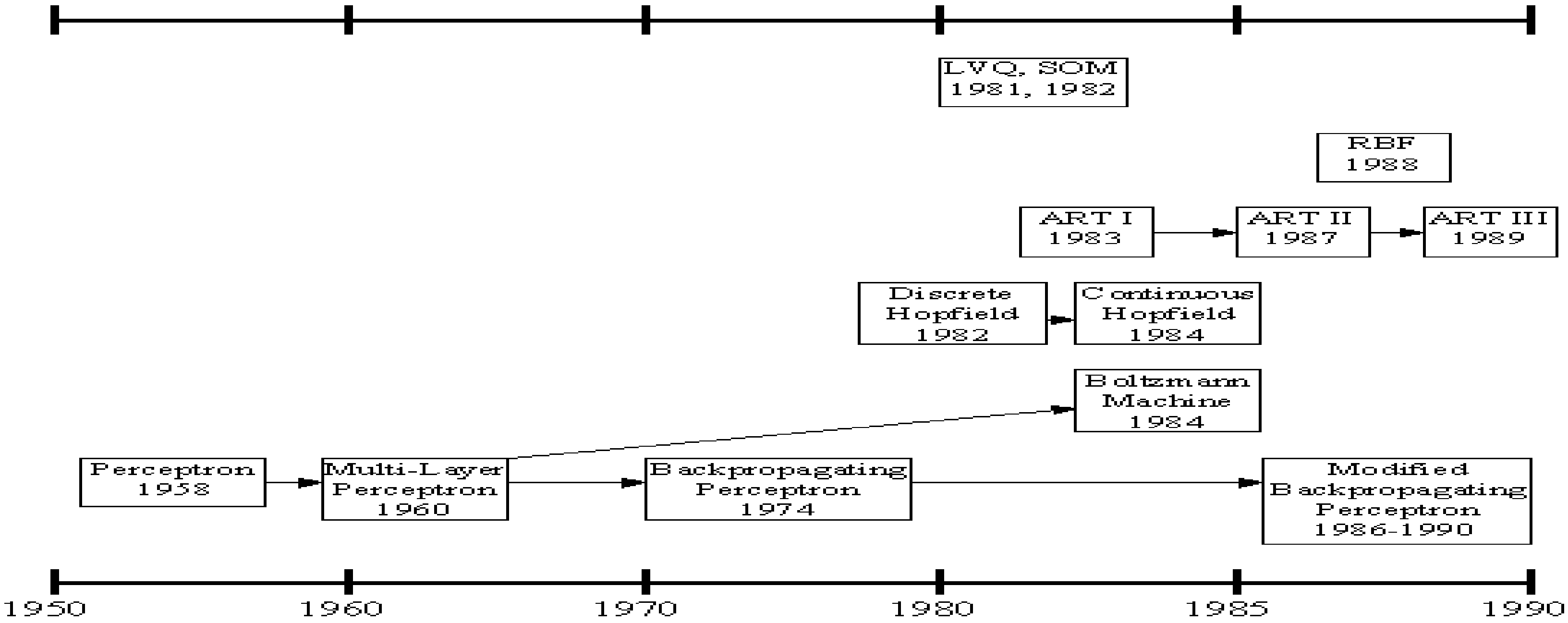


History of the ANN

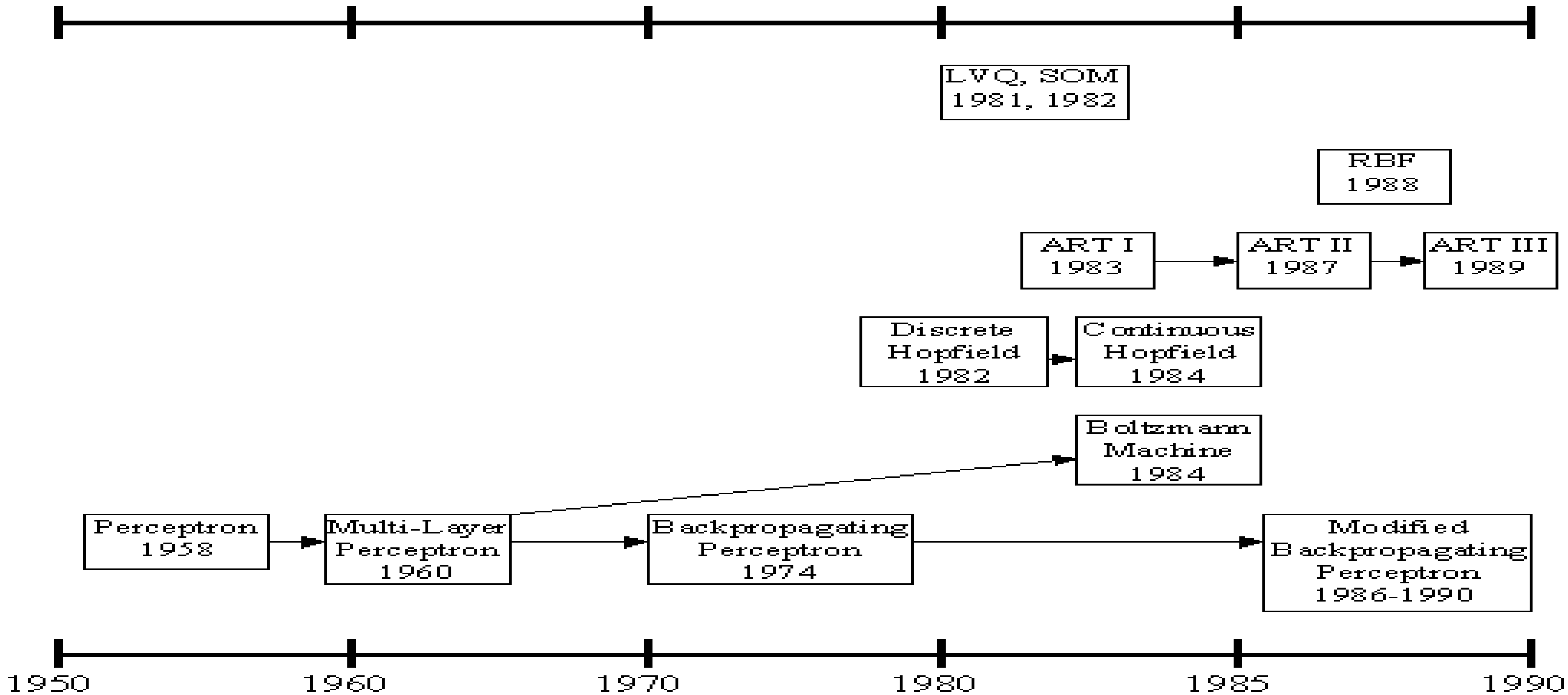
- history of the ANNs stems from the 1940s, the decade of the first electronic computer.
- However, the first important step took place in 1957 when Rosenblatt introduced the first concrete neural model, the perceptron. Rosenblatt also took part in constructing the first successful neurocomputer, the Mark I Perceptron. After this, the development of ANNs has proceeded as described in *Figure*.



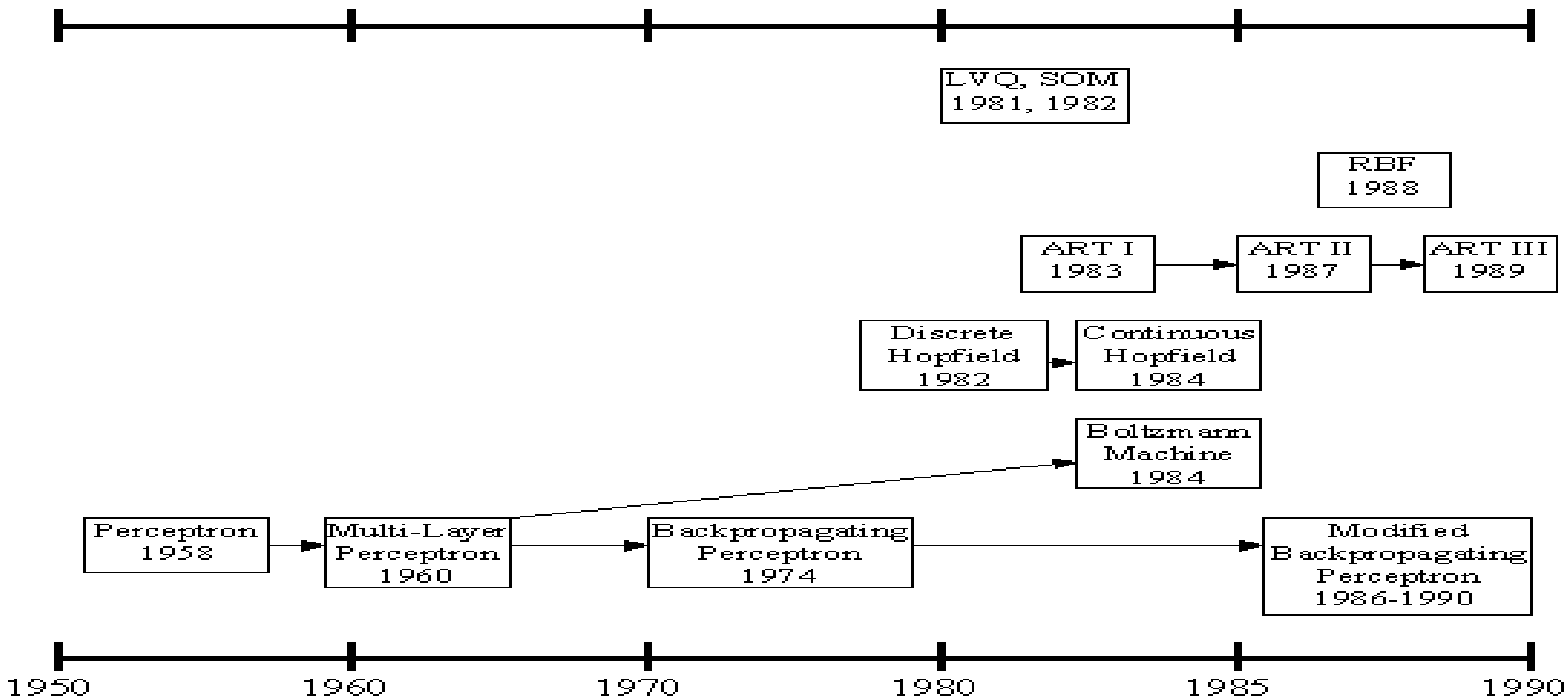
- Rosenblatt's original perceptron model contained only one layer. From this, a multi-layered model was derived in 1960. At first, the use of the multi-layer perceptron (MLP) was complicated by the lack of a appropriate learning algorithm.
- In 1974, Werbos came to introduce a so-called backpropagation algorithm for the three-layered perceptron network.



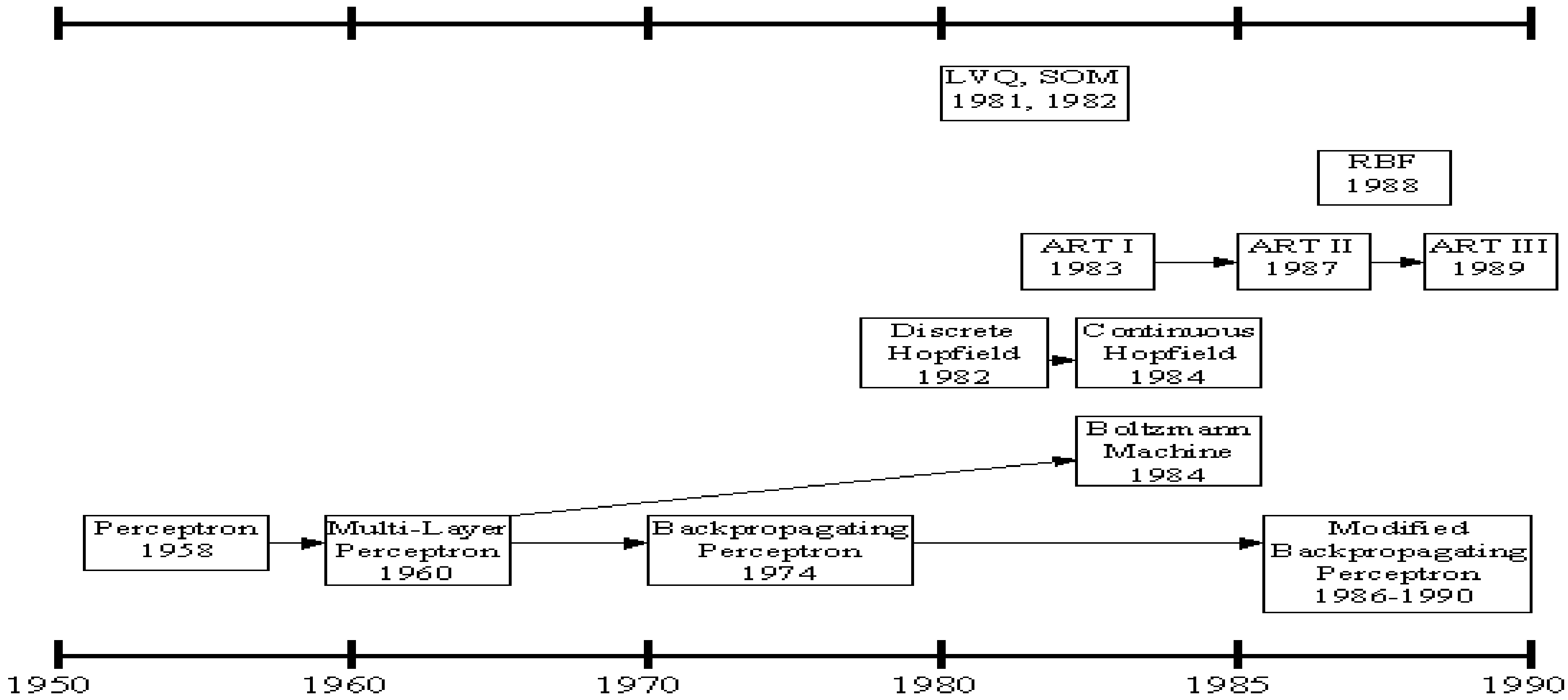
- in 1986, The application area of the MLP networks remained rather limited until the breakthrough when a general back propagation algorithm for a multi-layered perceptron was introduced by Rummelhart and Mclelland.
- in 1982, Hopfield brought out his idea of a neural network. Unlike the neurons in MLP, the Hopfield network consists of only one layer whose neurons are fully connected with each other.



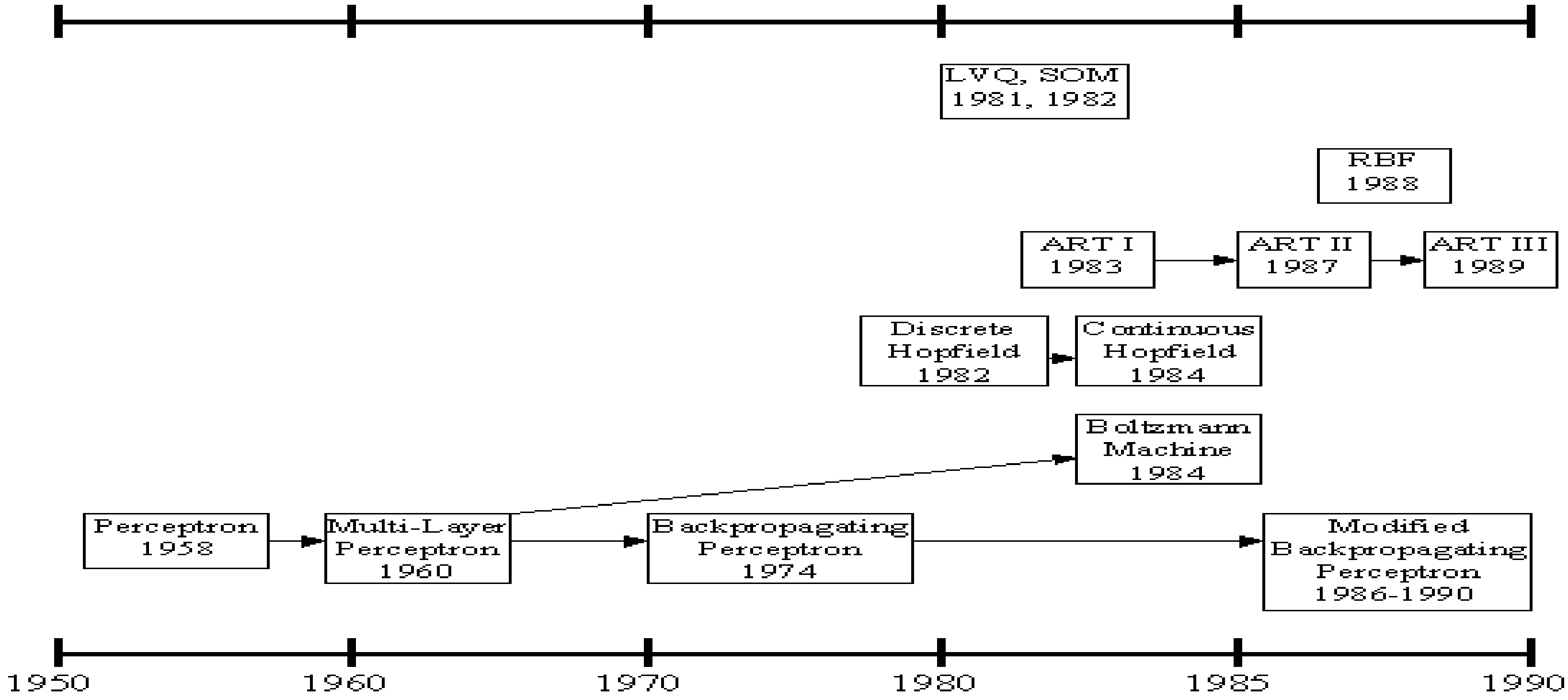
- Since then, new versions of the Hopfield network have been developed. The Boltzmann machine has been influenced by both the Hopfield network and the MLP.



- in 1988, Radial Basis Function (RBF) networks were first introduced by Broomhead & Lowe. Although the basic idea of RBF was developed 30 years ago under the name method of potential function, the work by Broomhead & Lowe opened a new frontier in the neural network community.

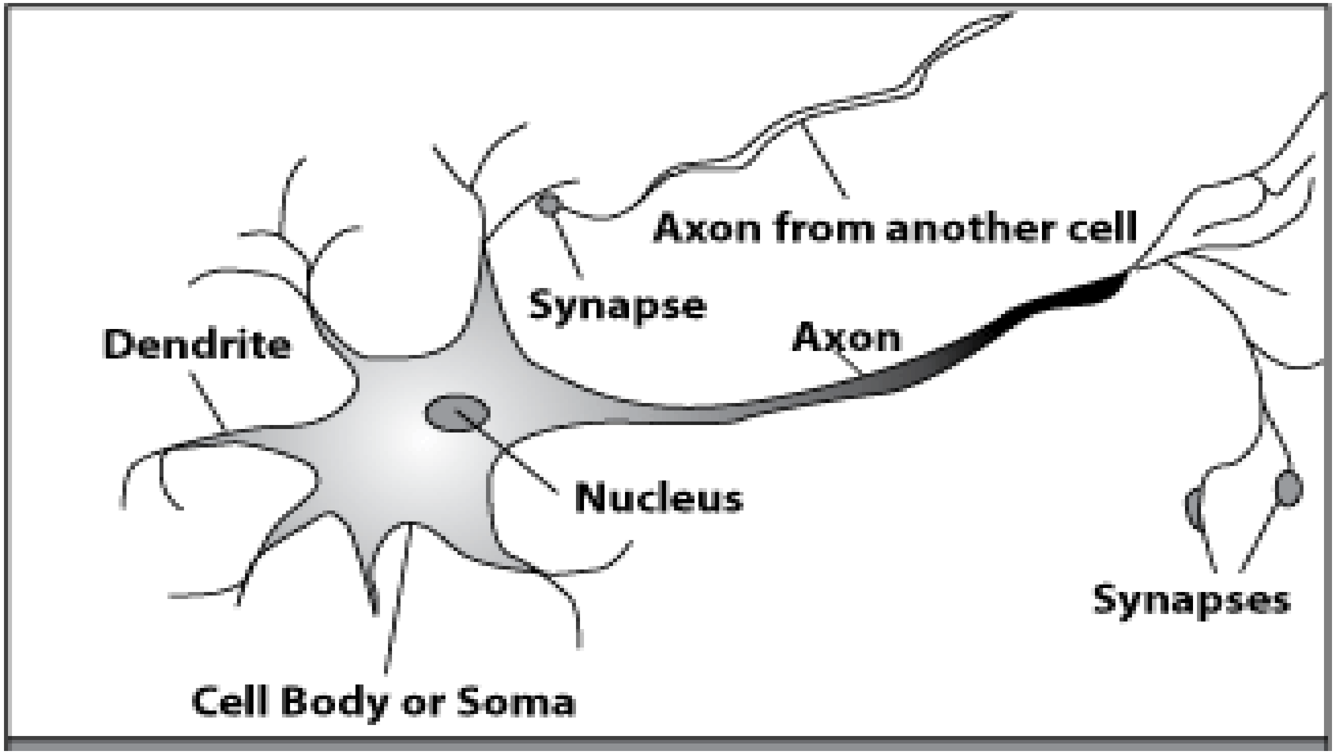


- in 1982, A totally unique kind of network model is the Self-Organizing Map (SOM) introduced by Kohonen. SOM is a certain kind of topological map which organizes itself based on the input patterns that it is trained with. The SOM originated from the LVQ (Learning Vector Quantization) network the underlying idea of which was also Kohonen's in 1972.



How do our brains work?

- A processing element



Dendrites: Input
Cell body: Processor
Synaptic: Link
Axon: Output

- A neuron is connected to other neurons through about *10,000 synapses*
- A neuron receives input from other neurons. Inputs are combined.
- Once input exceeds a critical level, the neuron discharges a spike - an electrical pulse that travels from the body, down the axon, to the next neuron(s)
- The axon endings almost touch the dendrites or cell body of the next neuron.

- Transmission of an electrical signal from one neuron to the next is effected by neurotransmitters.
- Neurotransmitters are chemicals which are released from the first neuron and which bind to the second.
- This link is called a synapse. The strength of the signal that reaches the next neuron depends on factors such as the amount of neurotransmitter available.

Feed Forward Neural Networks

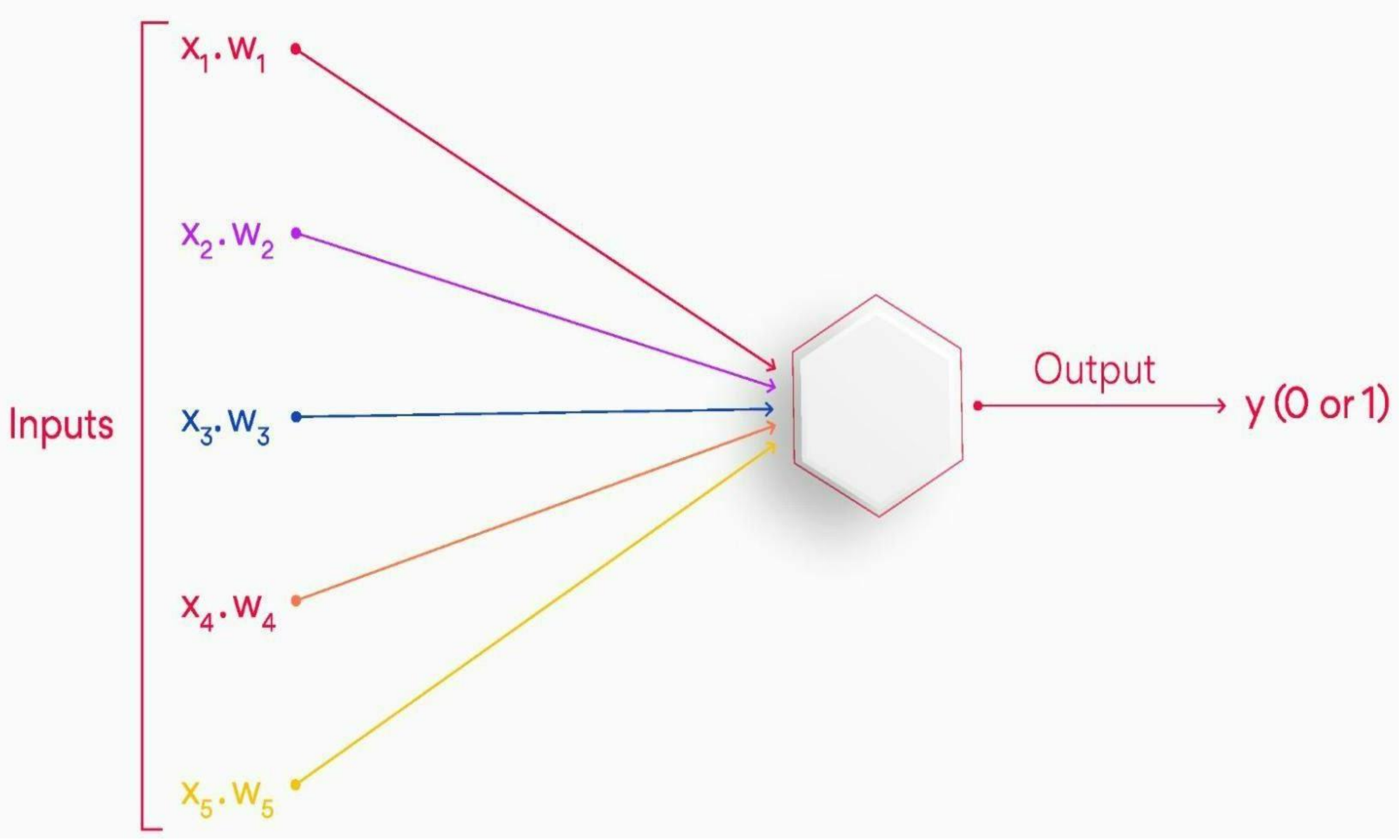
Feed forward neural network?

- Feed forward neural networks are artificial neural networks in which nodes do not form loops.
- This type of neural network is also known as a multi-layer neural network as all information is only passed forward.
- During data flow, input nodes receive data, which travel through hidden layers, and exit output nodes.
- No links exist in the network that could get used to by sending information back from the output node.

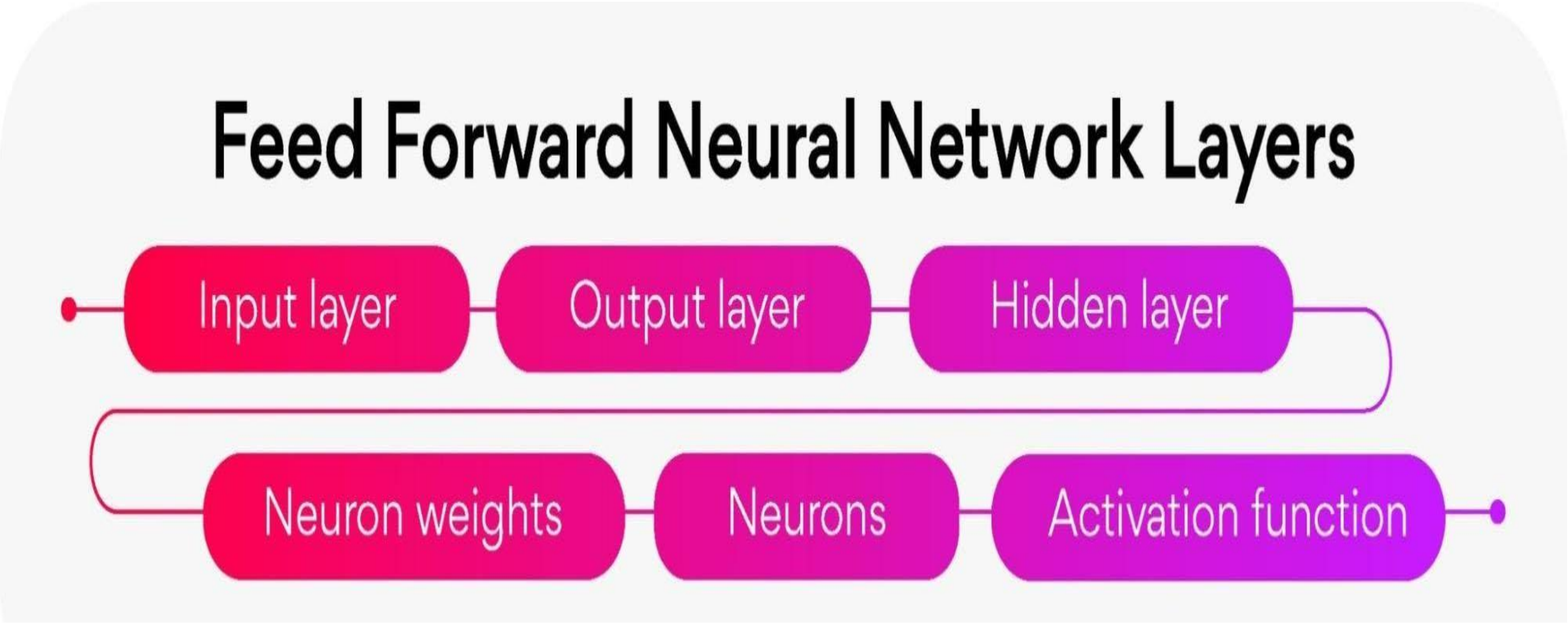
A feed forward neural network approximates functions in the following way:

- An algorithm calculates classifiers by using the formula $y = f^*(x)$.
- Input x is therefore assigned to category y.
- According to the feed forward model, $y = f(x; \theta)$. This value determines the closest approximation of the function.

working principle



- When the feed forward neural network gets simplified, it can appear as a single layer perceptron.
- This model multiplies inputs with weights as they enter the layer. Afterward, the weighted input values get added together to get the sum. As long as the sum of the values rises above a certain threshold, set at zero, the output value is usually 1, while if it falls below the threshold, it is usually -1.
- As a feed forward neural network model, the single-layer perceptron often gets used for classification. Machine learning can also get integrated into single-layer perceptrons.
- Through training, neural networks can adjust their weights based on a property called the delta rule, which helps them compare their outputs with the intended values.
- As a result of training and learning, gradient descent occurs. Similarly, multi-layered perceptrons update their weights. But, this process gets known as back-propagation. If this is the case, the network's hidden layers will get adjusted according to the output values produced by the final layer.


Feed Forward Neural Network Layers
Input layer
Output layer
Hidden layer
Neuron weights
Neurons
Activation function

- **Input layer:** The neurons of this layer receive input and pass it on to the other layers of the network. Feature or attribute numbers in the dataset must match the number of neurons in the input layer.
- **Output layer:** According to the type of model getting built, this layer represents the forecasted feature.
- **Hidden layer:** Input and output layers get separated by hidden layers. Depending on the type of model, there may be several hidden layers.
- There are several neurons in hidden layers that transform the input before actually transferring it to the next layer. This network gets constantly updated with weights in order to make it easier to predict.

- **Neuron weights:** Neurons get connected by a weight, which measures their strength or magnitude. Similar to linear regression coefficients, input weights can also get compared.
  - Weight is normally between 0 and 1, with a value between 0 and 1.
- **Neurons:** Artificial neurons get used in feed forward networks, which later get adapted from biological neurons. A neural network consists of artificial neurons.
  - Neurons function in two ways: first, they create weighted input sums, and second, they activate the sums to make them normal.
- Activation functions can either be linear or nonlinear. Neurons have weights based on their inputs. During the learning phase, the network studies these weights.

- **Activation Function:** Neurons are responsible for making decisions in this area.
  - According to the activation function, the neurons determine whether to make a linear or nonlinear decision. Since it passes through so many layers, it prevents the cascading effect from increasing neuron outputs.
  - An activation function can be classified into three major categories: **sigmoid, Tanh,** and **Rectified Linear Unit (ReLu).**
- **Sigmoid:** Input values between 0 and 1 get mapped to the output values.
- **Tanh:** A value between -1 and 1 gets mapped to the input values.
- **Rectified linear Unit(ReLu):** Only positive values are allowed to flow through this function. Negative values get mapped to 0.

Feed Forward Neural Network Functions

1. Cost function
2. Loss function
3. Gradient learning algorithm
4. Output units

Cost function

- In a feed forward neural network, the cost function plays an important role.
- The categorized data points are little affected by minor adjustments to weights and biases.
- Thus, a smooth cost function can get used to determine a method of adjusting weights and biases to improve performance.
- Following is a definition of the mean square error cost function:

$$C(w, b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2.$$

Where,

w = the weights gathered in the network

b = biases

n = number of inputs for training

a = output vectors

x = input

$||v||$ = vector v's normal length

Loss function

- The loss function of a neural network gets used to determine if an adjustment needs to be made in the learning process.
- Neurons in the output layer are equal to the number of classes. Showing the differences between predicted and actual probability distributions.
- Following is the cross-entropy loss for binary classification.

Cross Entropy Loss:

$$L(\Theta) = \begin{cases} -log(\hat{y}) & \text{if } y = 1 \\ -log(1-\hat{y}) & \text{if } y = 0 \end{cases}$$

As a result of multiclass categorization, a cross-entropy loss occurs:

Cross Entropy Loss:

$$L(\Theta) = -\sum_{i=1}^{k} y_i \log\left(\hat{y}_i\right)$$

Gradient learning algorithm

- In the gradient descent algorithm, the next point gets calculated by scaling the gradient at the current position by a learning rate. Then subtracted from the current position by the achieved value.
- To decrease the function, it subtracts the value (to increase, it would add). As an example, here is how to write this procedure:

$$p_{n+1} = p_n - \eta \nabla f(p_n)$$

- The gradient gets adjusted by the parameter η, which also determines the step size. Performance is significantly affected by the learning rate in machine learning.

Output units

- In the output layer, output units are those units that provide the desired output or prediction, thereby fulfilling the task that the neural network needs to complete.
- There is a close relationship between the choice of output units and the cost function. Any unit that can serve as a hidden unit can also serve as an output unit in a neural network.

Advantages of feed forward Neural Networks

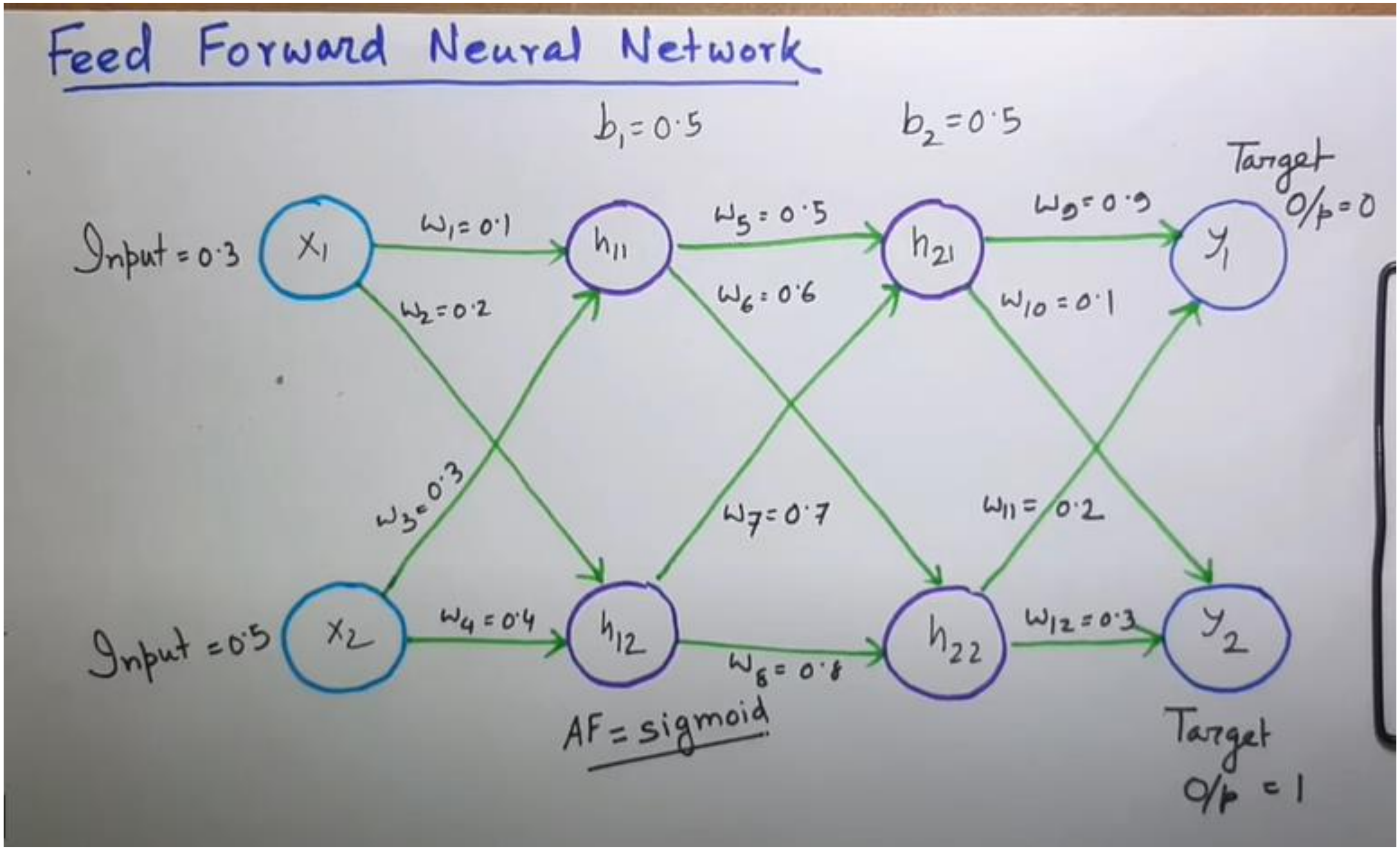
- Machine learning can be boosted with feed forward neural networks' simplified architecture.
- Multi-network in the feed forward networks operate independently, with a moderated intermediary.
- Complex tasks need several neurons in the network.
- Neural networks can handle and process nonlinear data easily compared to perceptrons and sigmoid neurons, which are otherwise complex.
- A neural network deals with the complicated problem of decision boundaries.
- Depending on the data, the neural network architecture can vary. For example, convolutional neural networks (CNNs) perform exceptionally well in image processing, whereas recurrent neural networks (RNNs) perform well in text and voice processing.
- Neural networks need graphics processing units (GPUs) to handle large datasets for massive computational and hardware performance. Several GPUs get used widely in the market, including Kaggle Notebooks and Google Collab Notebooks.

Applications of feed forward neural networks

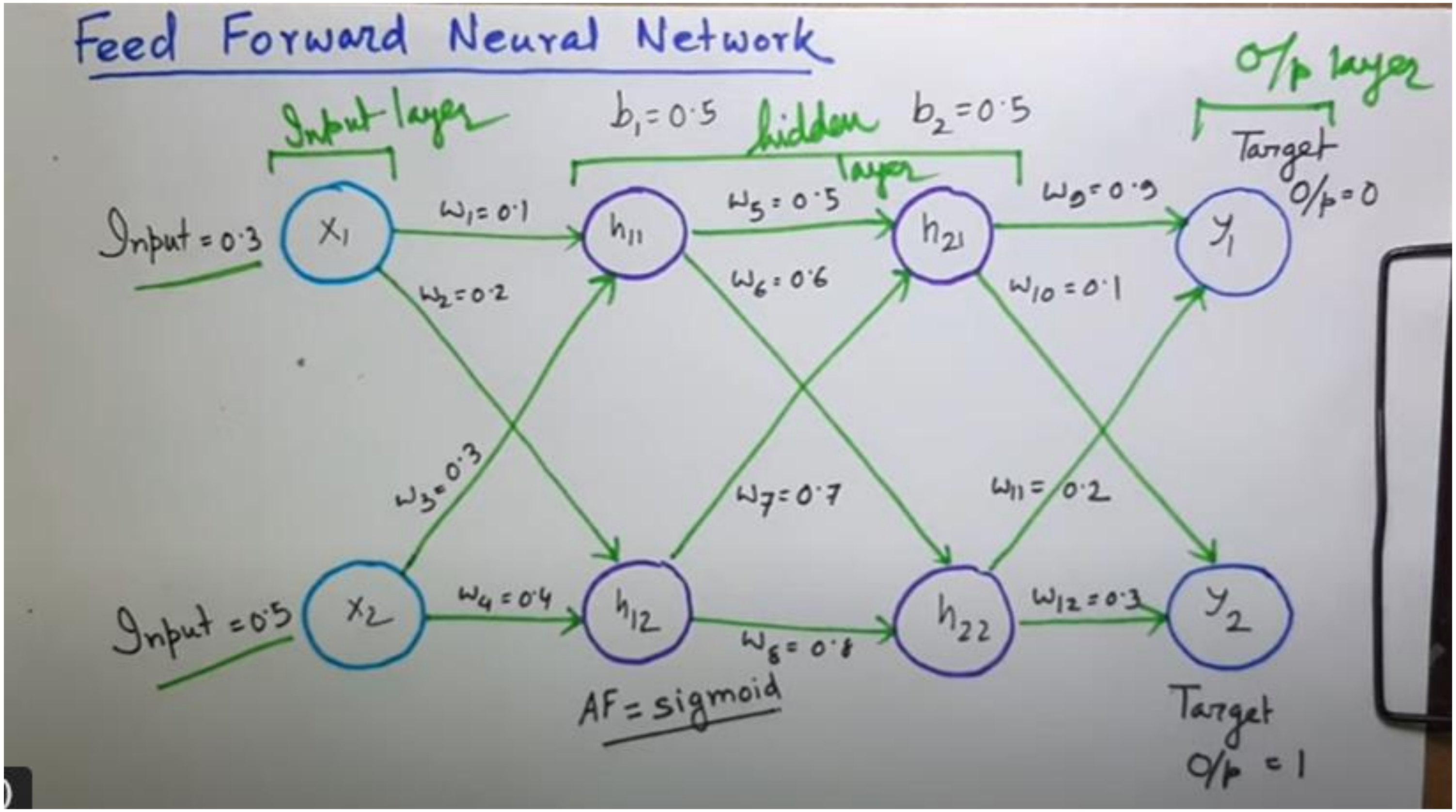
Applications

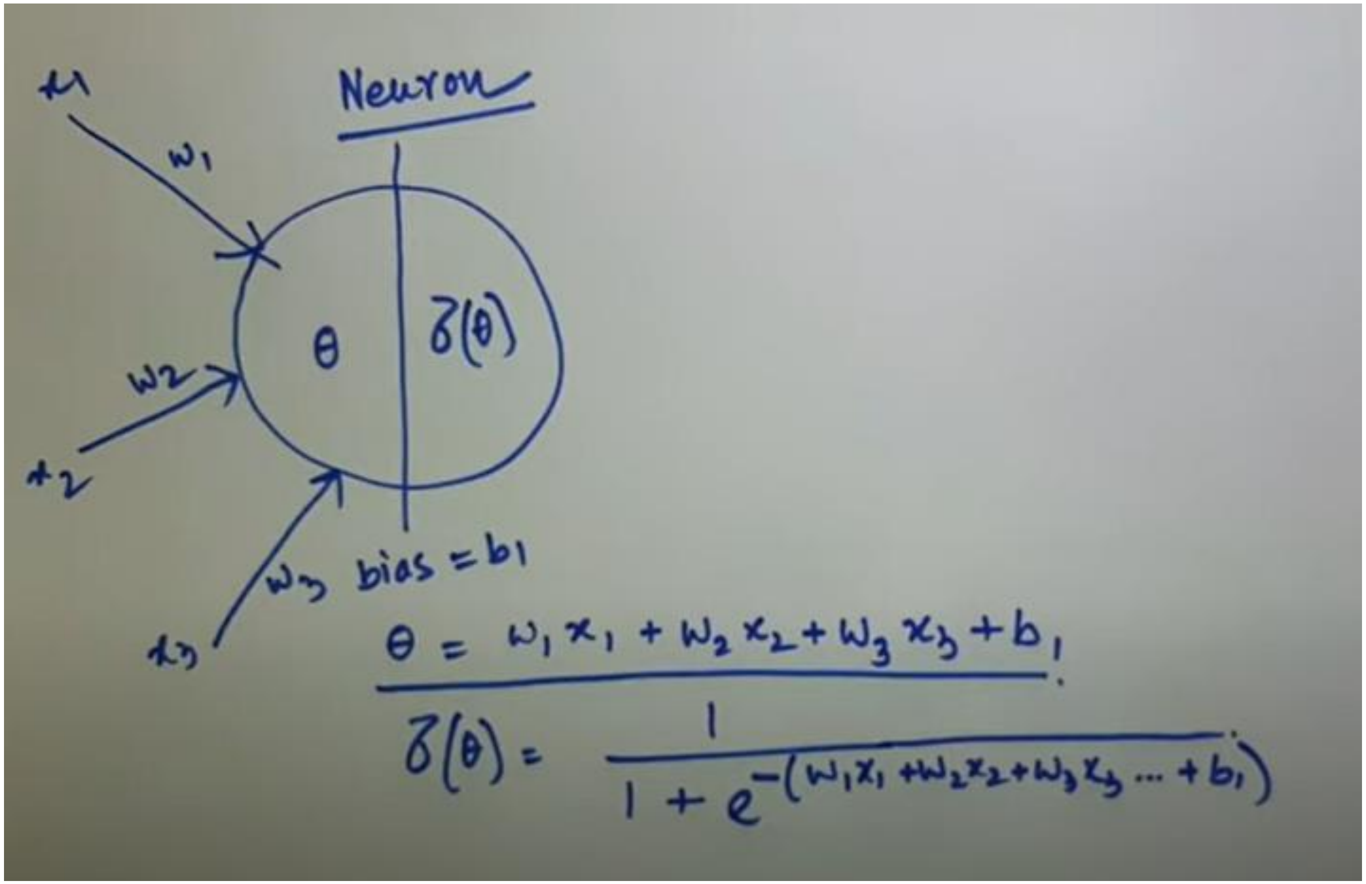
1. Physiological feed forward system
2. Gene regulation and feed forward
3. Automation and machine management
4. Parallel feed forward compensation with derivative

Problem



Feed Forward Neural Network
$b_1 = 0.5$
$b_2 = 0.5$
Target O/p = 0
Input = 0.3
x_1
$w_1 = 0.1$
h_{11}
$w_5 = 0.5$
h_{21}
$w_9 = 0.9$
y_1
$w_2 = 0.2$
$w_6 = 0.6$
$w_{10} = 0.1$
$w_3 = 0.3$
$w_7 = 0.7$
$w_{11} = 0.2$
Input = 0.5
x_2
$w_4 = 0.4$
h_{12}
$w_8 = 0.8$
h_{22}
$w_{12} = 0.3$
y_2
AF = sigmoid
Target O/p = 1


Feed Forward Neural Network
Input layer
$b_1 = 0.5$
hidden layer
$b_2 = 0.5$
O/p layer
Target O/p = 0
Input = 0.3
x_1
$w_1 = 0.1$
h_{11}
$w_5 = 0.5$
h_{21}
$w_9 = 0.9$
y_1
$w_2 = 0.2$
$w_6 = 0.6$
$w_{10} = 0.1$
$w_3 = 0.3$
$w_7 = 0.7$
$w_{11} = 0.2$
Input = 0.5
x_2
$w_4 = 0.4$
h_{12}
$w_8 = 0.8$
h_{22}
$w_{12} = 0.3$
y_2
AF = sigmoid
Target O/p = 1


Neuron
x1
w1
θ
δ(θ)
w2
x2
w3
bias = b1
x3
θ = w1 x1 + w2 x2 + w3 x3 + b1
δ(θ) = 1 / (1 + e^−(w1x1 + w2x2 + w3x3 ... + b1))

Computing hidden layer value

$$h_{11} = w_1 * x_1 + w_3 * x_2 + b_1$$

$$= 0.1*0.3+0.3*0.5 + 0.5$$

$$= 0.03 + 0.15 + 0.5$$

$$= 0.68$$

$$\delta(h_{11})= \delta(0.68) = 1/(1+e^{-0.68}) = 0.66$$

$$h_{12} = w_2 * x_1 + w_4 * x_2 + b_1$$

$$= 0.2*0.3+0.4*0.5 + 0.5$$

$$= 0.06 + 0.20 + 0.5$$

$$= 0.76$$

$$\delta(h_{12}) = \delta(0.76) = 1/(1+e^{-0.76}) = 0.68$$

Computing hidden layer value

$$h_{21} = w_5 * \delta(h_{11}) + w_7 * \delta(h_{12}) + b_1$$

$$= 0.5*0.66+0.7*0.68 + 0.5$$

$$= 0.33 + 0.476 + 0.5$$

$$= 1.306$$

$$\delta(h_{21}) = \delta(1.306) = 1/(\ 1+e^{-1.306}) = 0.786$$

$$h_{22} = w_6 * \delta(h_{11}) + w_8 * \delta(h_{12}) + b_1$$

$$= 0.6*0.66+0.8*0.68 + 0.5$$

$$= 1.504$$

$$\delta(h_{22}) = \delta(1.504) = 1/(\ 1+e^{-1.504}) = 0.818$$

Computing hidden layer value

$y_1 = w_9 * \delta(h_{21}) + w_{11} * \delta(h_{22})$

$= 0.9*0.786+0.2*0.818$

$= 0.1636$

$\delta(y_1)= \delta(0.1636) = 1/(1+e^{-.1636}) = 0.54$

$y_2 = w10 * \delta(h_{21}) + w_{12} * \delta(h_{22})$

$= 0.1*0.786+0.3*0.818$

$= 0.324$

$\delta(y_2) = \delta(0.324) = 1/(1+e^{-0.324}) = 0.58$

Error

E/L = Mean square Error

$= 1/2[(y_A' - y_T')^2 + (y_A - y_T)^2]$

$= 1/2[(0.54 - 0)^2 + (0.58 - 1)^2]$

$= 0.234$

$$y_1 = w_9 \cdot \sigma(h_{21}) + w_{11}\,\sigma(h_{22})$$
$$= 0.9 \times 0.786 + 0.2 \times 0.818$$
$$= 0.1636$$
$$\therefore \sigma(y_1) = 0.54$$

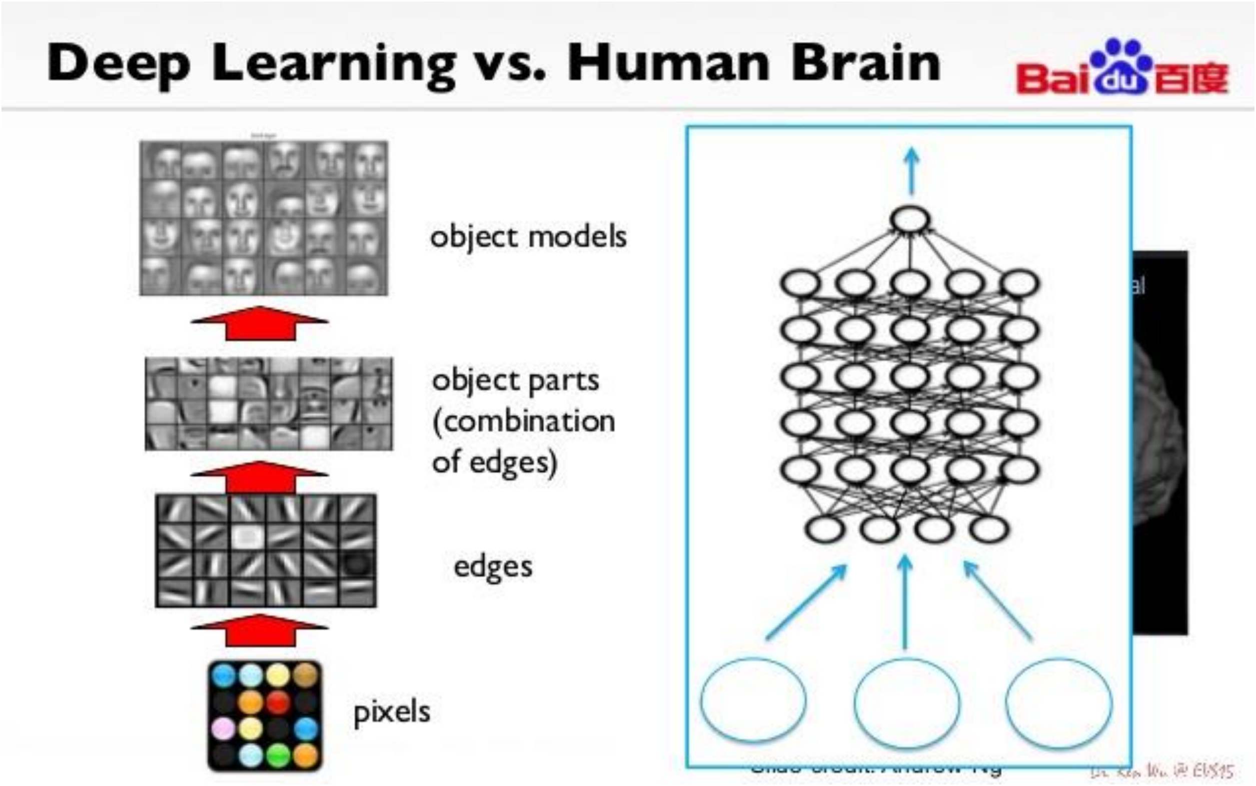
$$y_2 = w_{10}\,\sigma(h_{21}) + w_{12}\,\sigma(h_{22})$$
$$= 0.1 \times 0.786 + 0.3 \times 0.818$$
$$= 0.324$$
$$\therefore \sigma(y_2) = 0.58$$

E/L = Mean Squared Error

$$= \frac{1}{2}\left[\left(y_A^1 - y_T^1\right)^2 + \left(y_A^2 - y_T^2\right)^2\right]$$
$$= \frac{1}{2}\left[\left(0.54 - 0\right)^2 + \left(0.58 - 1\right)^2\right]$$
$$= \frac{1}{2}\left(0.2916 + 0.1764\right)$$
$$= \frac{0.468}{2} = 0.234$$

PRINCIPLE

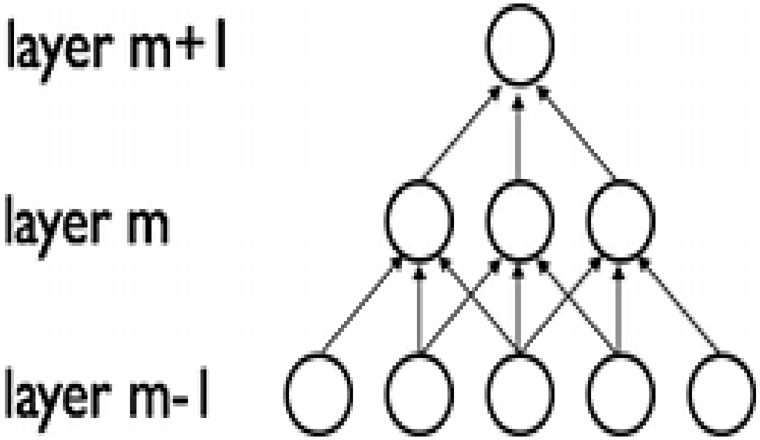
Deep learning is based on the concept of artificial neural networks, or computational systems that mimic the way the human brain functions.



Deep learning is a fast-growing field, and new architectures, variants appear every few weeks. We'll see discuss the major three:

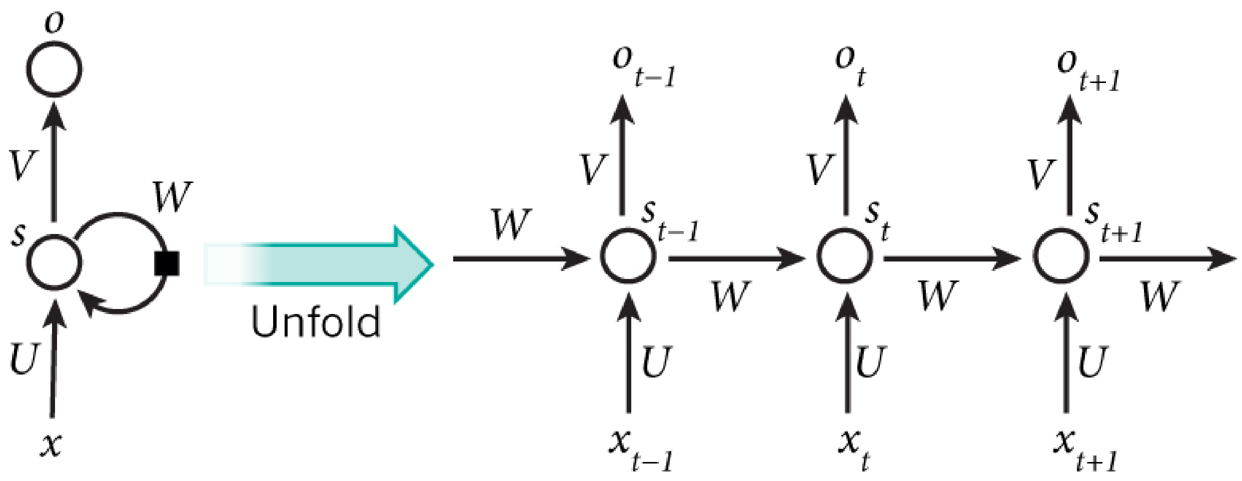
1. Convolution Neural Network (CNN)

CNNs exploit spatially-local correlation by enforcing a local connectivity pattern between neurons of adjacent layers.



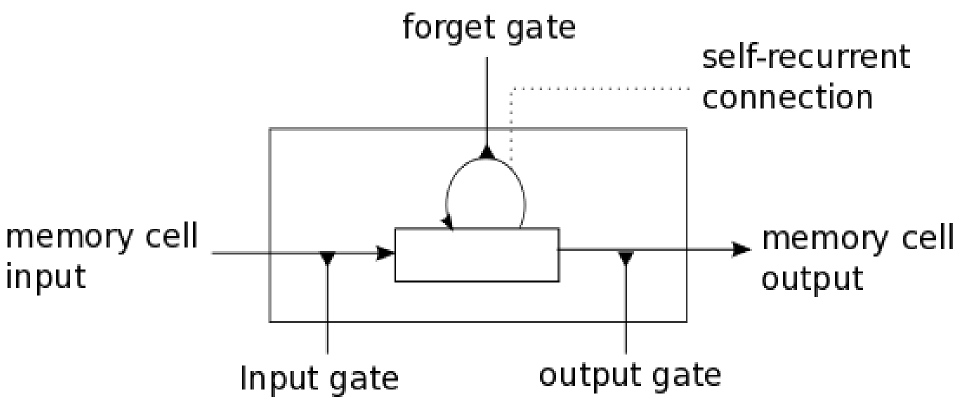
2. Recurrent Neural Network (RNN)

RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations. Or RNNs have a “memory” which captures information about what has been calculated so far.



3. Long-Short Term Memory

LSTM can learn "Very Deep Learning" tasks that require memories of events that happened thousands or even millions of discrete time steps ago.LSTM works even when there are long delays, and it can handle signals that have a mix of low and high frequency components.



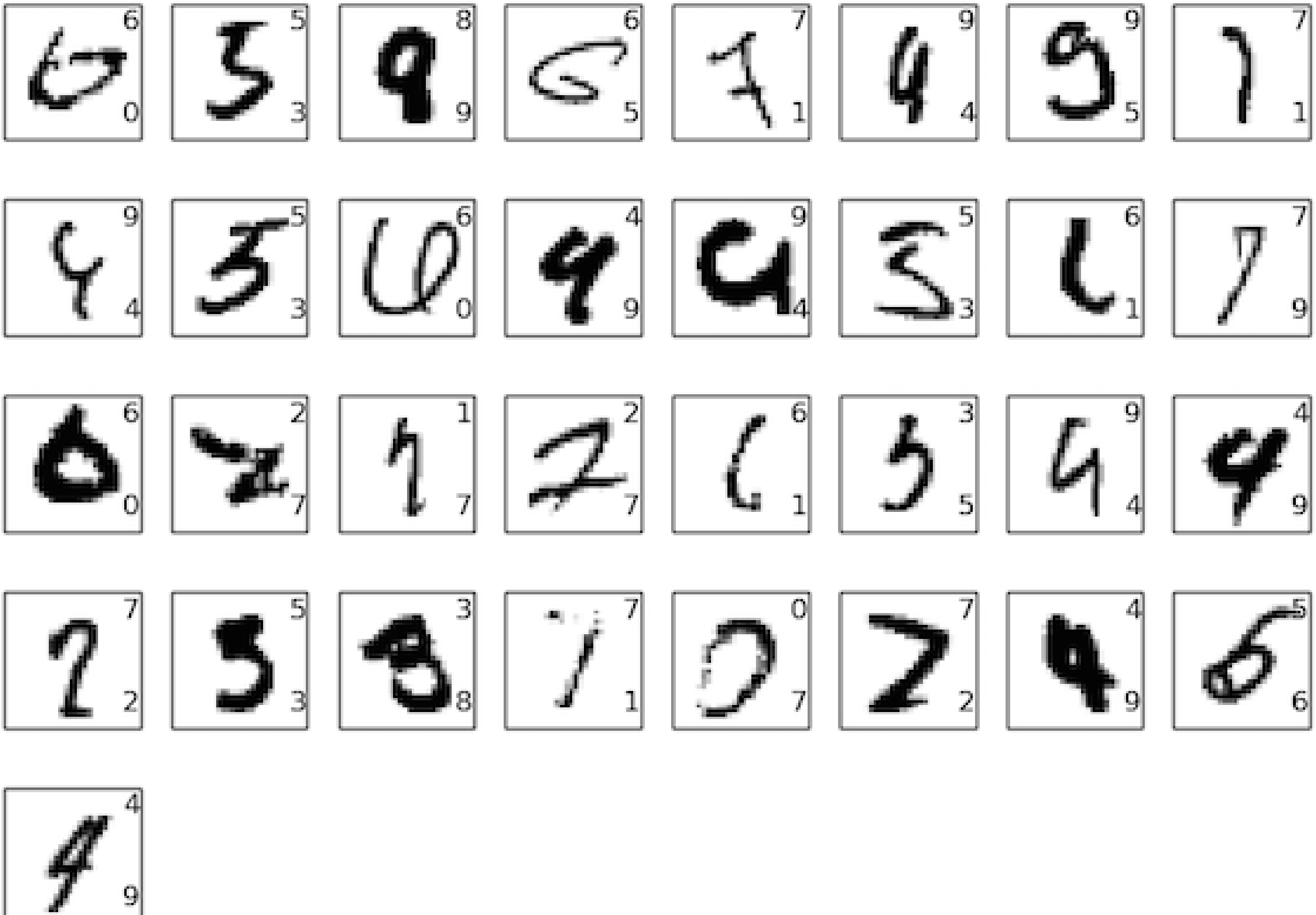
WORKING

Consider the following handwritten sequence:

Most people effortlessly recognize those digits as 504192. That ease is deceptive.

The difficulty of visual pattern recognition becomes apparent if you attempt to write a computer program to recognize digits like those above.

WORKING

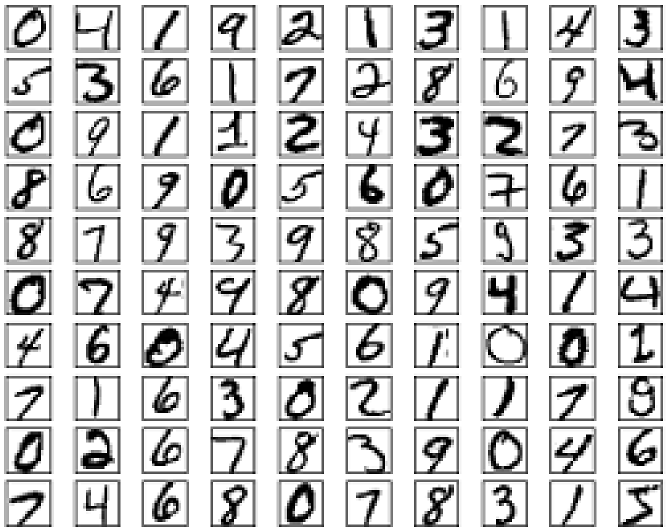


6 0
5 3
8 9
6 5
7 1
9 4
9 5
7 1
9 4
5 3
6 0
4 9
9 4
5 3
6 1
7 9
6 0
2 7
1 7
2 7
6 1
3 5
9 4
4 9
7 2
5 3
3 8
7 1
0 7
7 2
4 9
5 6
4 9

Vellore Institute of Technology

The idea of neural network is to develop a system which can learn from these large training examples.

Each neuron assigns a weighting to its input — how correct or incorrect it is relative to the task being performed. The final output is then determined by the total of those weightings



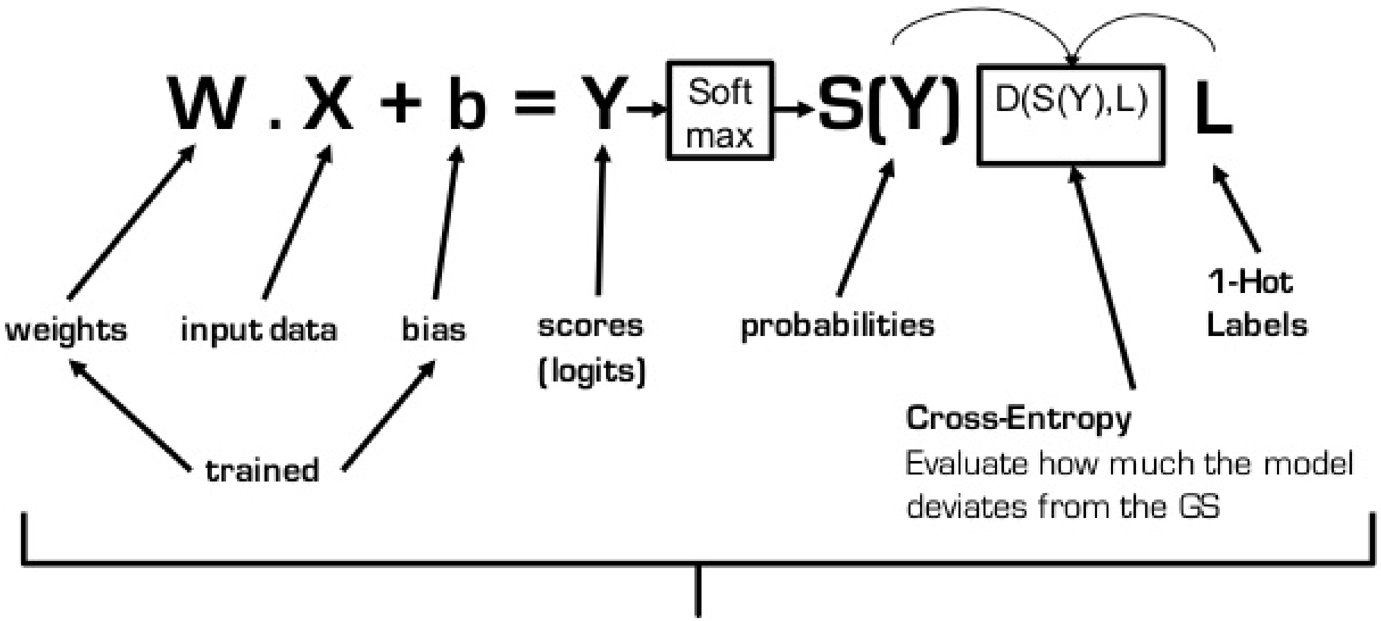
A training Sample

A very basic approach: Binary Classifier

FORMULATIONS

The basis of deep learning is classification which can be further used for detection, ranking, regression, etc.

- Logistic Classifier => Linear Classifier



ADVANTAGES

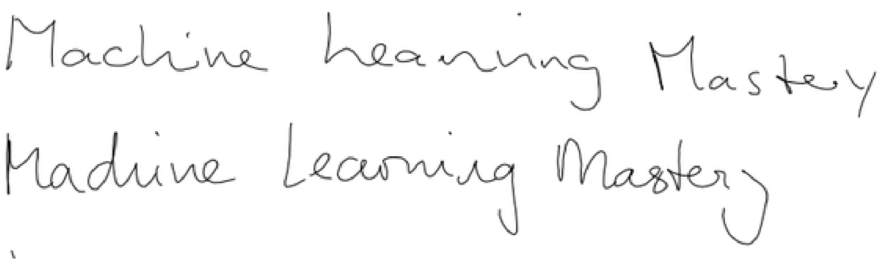
1. It does feature extraction, no need for engineering features
2. Moving towards raw features
3. Better optimization
4. A new level of noise robustness
5. Multi-task and transfer learning
6. Better Architectures

CHALLENGES

1. Need a large dataset
2. Because you need a large dataset, training time is usually significant.
3. The scale of a net's weights is important for performance. When the features are of the same type this is not a problem. However, when the features are heterogeneous, it is.
4. Parameters are hard to interpret--although there is progress being made.
5. Hyperparamter tuning is non-trivial.

1. Automatic Colorization of Black and White Images

2. Automatically Adding Sounds To Silent Movies
3. Automatic Machine Translation
4. Object Classification and Detection in Photographs
5. Automatic Handwriting Generation



REAL TIME APPLICATIONS

6. Automatic Text Generation
7. Automatic Image Caption Generation
8. Automatic Game Playing

"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

Thank You!