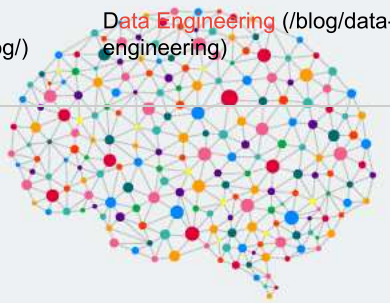
Overview of Artificial Neural Networks and its Applications

by **Jagreet** | May 05, 2017 | Categories - Artificial Neural Networks, (/blog/category/Artificial-Neural-Networks) Machine Learning, (/blog/category/Machine-Learning) Deep Learning (/blog/category/Deep-Learning)



Overview of Artificial Neural Networks and it's Applications

What is Neural Network?

The term 'Neural' is derived from the human (animal) nervous system's basic functional unit 'neuron' or nerve cells which are present in the brain and other parts of the human (animal) body.

Structure of Neurons in Brain

The typical nerve cell of human brain comprises of four parts -

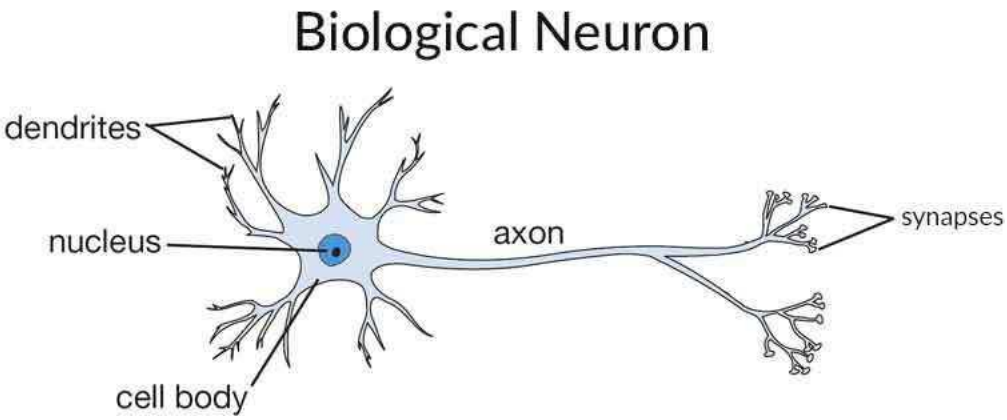


Image Source - cs231n.github.io

Dendrite - It receives signals from other neurons.

Soma (cell body) - It sums all the incoming signals to generate input.

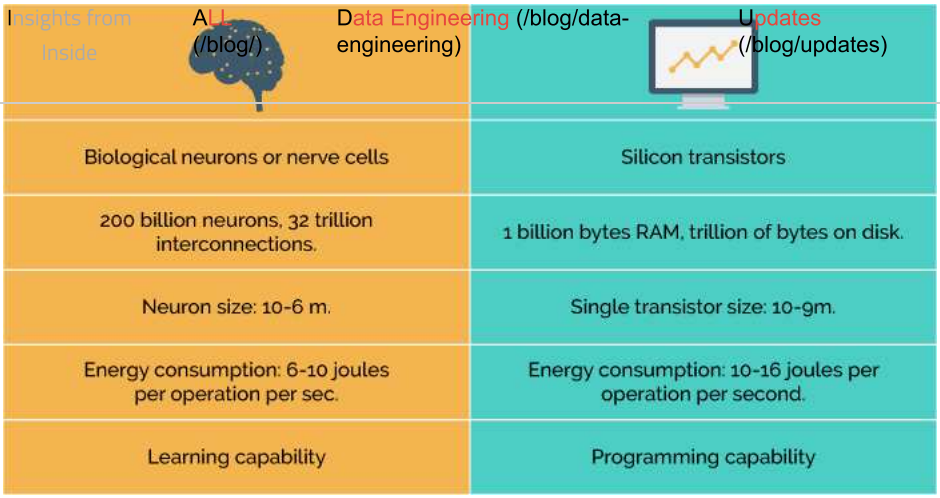
Axon - When the sum reaches a threshold value, neuron fires and the signal travels down the axon to the other neurons.

Synapses - The point of interconnection of one neuron with other neurons. The amount of signal transmitted depend upon the strength (synaptic weights) of the connections.

The connections can be inhibitory (decreasing strength) or excitatory (increasing strength) in nature.

So, neural network, in general, is a highly interconnected network of billions of neuron with trillion of interconnections between them.

How is Brain Different from Computers?



What is Artificial Neural Network?

Artificial Neural Networks are the biologically inspired simulations performed on the computer to perform certain specific tasks like clustering, classification, pattern recognition, etc.

Artificial Neural Networks, in general - is a biologically inspired network of artificial neurons configured to perform specific tasks.

What is Artificial Neural Networks

CLICK TO TWEET

(http://ctt.ec/acz1V)

Biological (Real) vs Artificial Neural Network

Characteristics	Artificial Neural Network	Biological(Real) Neural Network
Speed	Faster in processing information. Response time is in nanoseconds.	Slower in processing information. The response time is in milliseconds.
Processing	Serial processing.	Massively parallel processing.
Size & Complexity	Less size & complexity. It does not perform complex pattern recognition tasks.	Highly complex and dense network of interconnected neurons containing neurons of the order of 10^{11} with 10^{15} of interconnections.
Storage	Information storage is replaceable means new data can be added by deleting an old one.	Highly complex and dense network of interconnected neurons containing neurons of the order of 10^{11} with 10^{15} of interconnections.

Similarity of ANN with Biological Neural Network

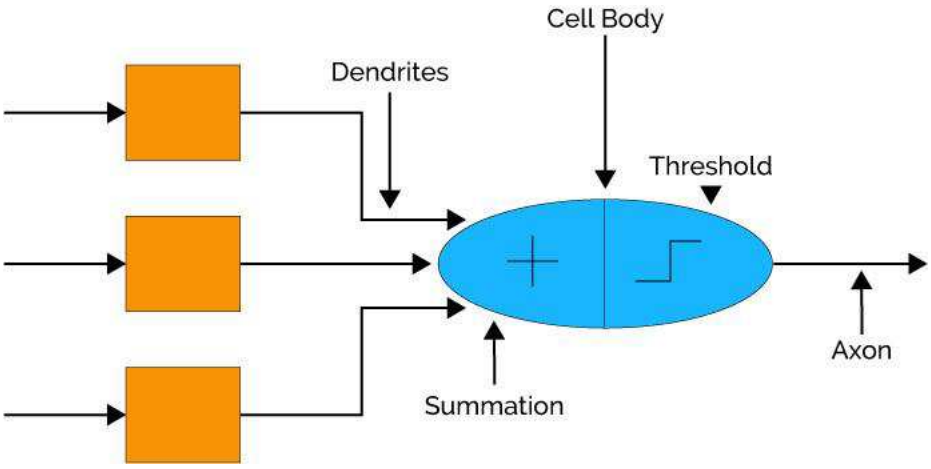
Neural Networks resemble the human brain in the following two ways -

- A neural network acquires knowledge through learning.
- A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

Fault tolerance	Fault intolerant. Information once corrupted cannot be retrieved in case of failure of the system.	Information storage is adaptable means new information is added by adjusting the interconnection strengths without destroying old information
Control Mechanism	There is a control unit for controlling computing activities	No specific control mechanism external to the computing task.

VON NEUMANN ARCHITECTURE BASED COMPUTING	ANN BASED COMPUTING
Serial processing - processing instruction and problem rule one at time (sequential)	Parallel processing - several processors perform simultaneously (multitasking)
Function logically with set of if & else rules - rule-based approach	Function by learning pattern from given input (image, text or video, etc.)
Programmable by higher level languages such as C, Java (https://www.xenonstack.com/blog/building-serverless-microservices-with-java), C++, etc.	ANN is in essence program themselves.
Requires either big or error prone parallel processors	Use of application specific multi-chips.

Analogy of Artificial Neural Network With Biological Neural Network

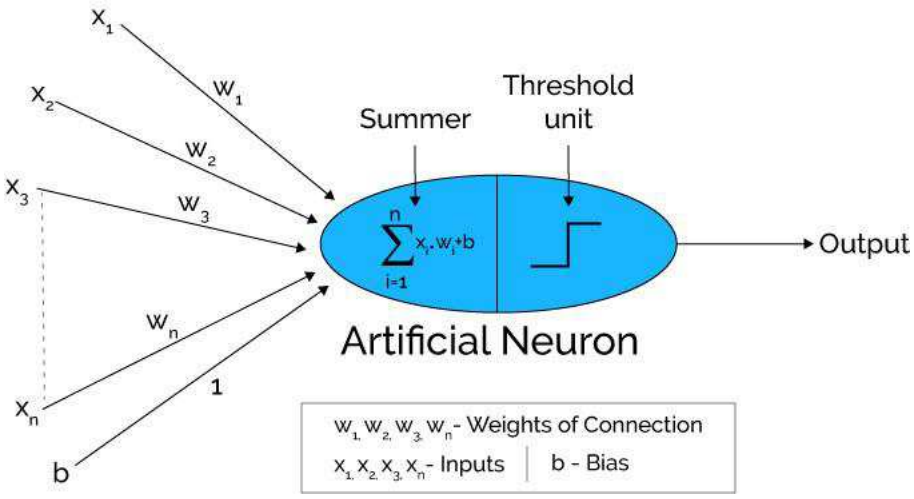


The dendrites in the **Biological Neural Network** are analogous to the weighted inputs based on their synaptic interconnection in the **Artificial Neural Network**.

The cell body is comparable to the artificial neuron unit in the Artificial Neural Network which also comprises of summation and threshold unit.

Axon carries output that is analogous to the output unit in case of Artificial Neural Network. So, ANN is modeled using the working of basic biological neurons.

How Does Artificial Neural Network Works?



Artificial (https://www.xenonstack.com/blog/overview-of-artificial-intelligence-and-role-of-natural-language-processing-in-big-data) Neural Networks can be viewed as weighted directed graphs in which artificial neurons are nodes, and directed edges with weights are connections between neuron outputs and neuron inputs.

The Artificial Neural Network receives information from the external world in the form of pattern and image in vector form. These inputs are mathematically designated by the notation x(n) for n number of inputs.

Each input is multiplied by its corresponding weights. Weights are the information used by the neural network to solve a problem. Typically weight represents the strength of the interconnection between neurons inside the **Neural Network**.

The weighted inputs are all summed up inside computing unit (artificial neuron). In case the weighted sum is zero, bias is added to make the output not- zero or to scale up the system response. Bias has the weight and input always equal to '1'.

The sum corresponds to any numerical value ranging from 0 to infinity. To limit the response to arrive at the desired value, the threshold value is set up. For this, the sum is passed through activation function.

The activation function is set to the transfer function used to get the desired output. There are linear as well as the nonlinear activation function.

Working of Artificial Neural Networks

Some of the commonly used activation function is - binary, sigmoidal (linear) and tan hyperbolic sigmoidal functions(nonlinear).

Binary - The output has only two values either 0 and 1. For this, the threshold value is set up. If the net weighted input is greater than 1, an output is assumed one otherwise zero.

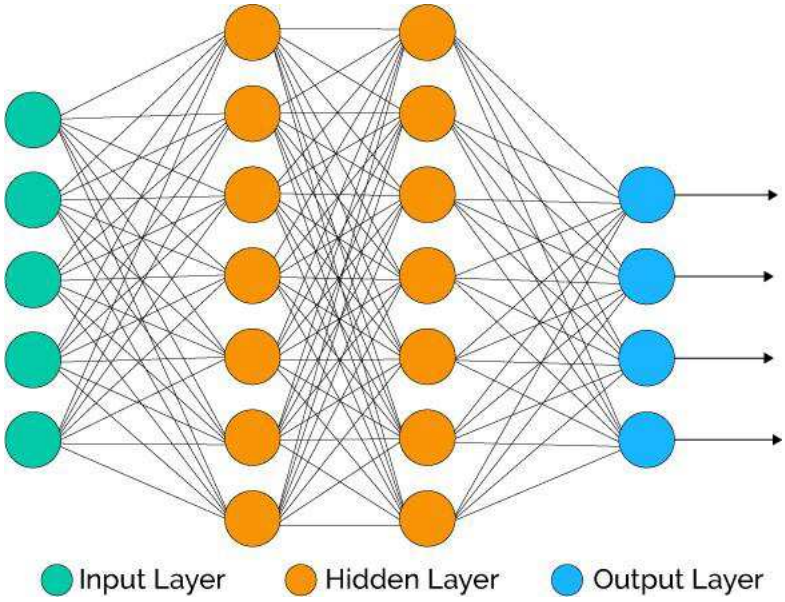
Sigmoidal Hyperbolic - This function has 'S' shaped curve. Here tan hyperbolic function is used to approximate output from net input. The function is defined as - f (x) = (1/1+ exp(-σx)) where σ - steepness parameter.

You May also Love to Read Overview of Artificial Intelligence & Role of Natural Language Processing in Big Data (https://www.xenonstack.com/blog/overview-of-artificial-intelligence-and-role-of-natural-language-processing-in-big-data)

Architecture of Artificial Neural Networks

A typical **Neural Network** contains a large number of artificial neurons called units arranged in a series of layers.

In typical Artificial Neural Network, comprises different layers -



- **Input layer** - It contains those units (**Artificial Neurons**) which receive input from the outside world on which network will learn, recognize about or otherwise process.
- **Output layer** - It contains units that respond to the information about how it's learned any task.
- **Hidden layer** - These units are in between input and output layers. The job of hidden layer is to transform the input into something that output unit can use in some way.

Most Neural Networks are fully connected that means to say each hidden neuron is fully linked to the every neuron in its previous layer(input) and to the next layer (output) layer.

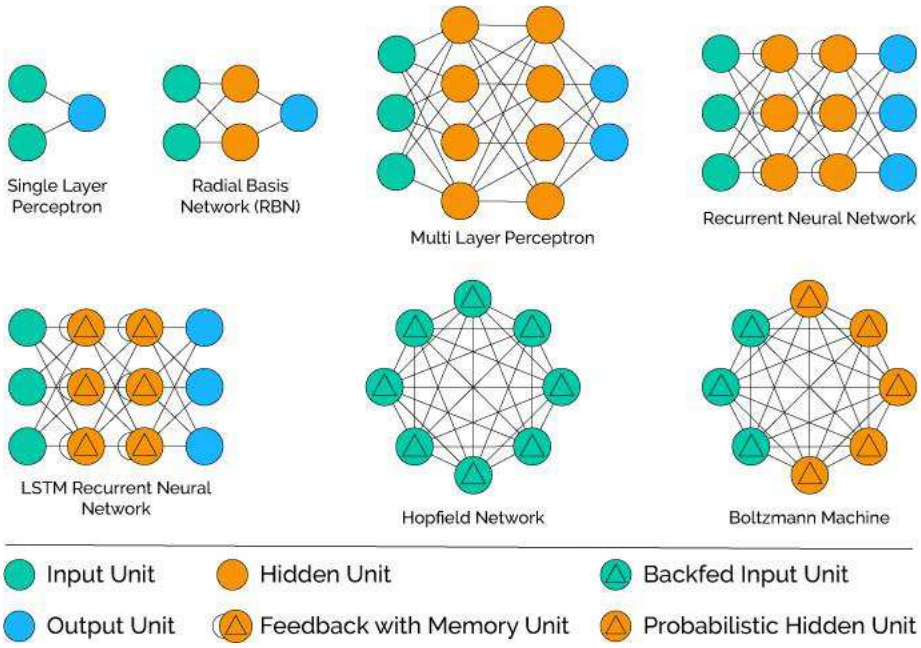
Architecture of Artificial Neural Networks

CLICK TO TWEET

(http://ctt.ec/e4fAa)

Parameter	Types	Description
Based on connection pattern	FeedForward, Recurrent	• **Feedforward** - In which graphs have no loops. • **Recurrent** - Loops occur because of feedback.
Based on the number of hidden layer	Single layer, Multi-Layer	• **Single Layer** - Having one hidden layer. E.g. , Single Perceptron • **Multilayer** - Having multiple hidden layers. Multilayer Perceptron
Based on nature of weights	Fixed, Adaptive	• **Fixed** - Weights are fixed a priori and not changed at all. • **Adaptive** - Weights are updated and changed during training.
Based on Memory unit	Static, Dynamic	• **Static** - Memoryless unit. The current output depends on the current input. E.g. , Feedforward network • **Dynamic** - Memory unit - The output depends upon the current input as well as the current output. E.g. , Recurrent Neural Network

Popular Neural Network Architectures



Perceptron - Neural Network is having two input units and one output units with no hidden layers. These are also known as single layer perceptrons.

Radial Basis Function Network - These networks are similar to the feed forward Neural Network except radial basis function is used as activation function of these neurons.

Multilayer Perceptron - These networks use more than one hidden layer of neurons, unlike single layer perceptron. These are also known as **Deep Feedforward Neural Networks**.

Recurrent Neural Network - Type of Neural Network in which hidden layer neurons has self-connections. **Recurrent Neural Networks** possess memory. At any instance, hidden layer neuron receives activation from the lower layer as well as it previous activation value.

Long /Short Term Memory Network (LSTM) - Type of Neural Network in which memory cell is incorporated inside hidden layer neurons is called LSTM network.

Hopfield Network - A fully interconnected network of neurons in which each neuron is connected to every other neuron. The network is trained with input pattern by setting a value of neurons to the desired pattern. Then its weights are computed. The weights are not changed. Once trained for one or more patterns, the network will converge to the learned patterns. It is different from other Neural Networks.

Boltzmann Machine Network - These networks are similar to Hopfield network except some neurons are input, while other are hidden in nature. The weights are initialized randomly and learn through back propagation algorithm.

Convolutional Neural Network - Get a complete overview of Convolutional Neural Networks through our blog Log Analytics with Machine Learning and Deep Learning (https://www.xenonstack.com/blog/log-analytics-with-deep-learning-and-machine-learning).

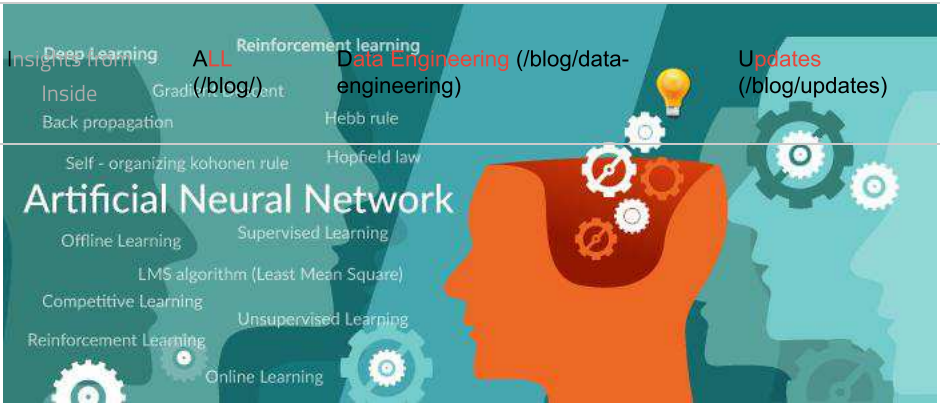
Popular Neural Network Architectures

Other types of Neural Network

Modular Neural Network - It is the combined structure of different types of the neural network like multilayer perceptron, **Hopfield Network**, **Recurrent Neural Network**, etc. which are incorporated as a single module into the network to perform independent subtask of whole complete Neural Networks.

Physical Neural Network - In this type of Artificial Neural Network, electrically adjustable resistance material is used to emulate the function of synapse instead of software simulations performed in the neural network.

Learning in Artificial Neural Networks



The neural network learns by adjusting its weights and bias (threshold) iteratively to yield desired output. These are also called free parameters. For learning to take place, the Neural Network is trained first. The training is performed using a defined set of rules also known as the learning algorithm.

Popular Learning Algorithms used in Neural Network

Gradient Descent - This is the simplest training algorithm used in case of supervised training model. In case, the actual output is different from target output, the difference or error is find out. The gradient descent algorithm changes the weights of the network in such a manner to minimize this mistake.

Back propagation - It is an extension of gradient based delta learning rule. Here, after finding an error (the difference between desired and target), the error is propagated backward from output layer to the input layer via hidden layer. It is used in case of **Multilayer Neural Network**.

Other learning algorithms

- Hebb Rule
- Self - Organizing Kohonen Rule
- Hopfield law
- LMS algorithm (Least Mean Square)
- Competitive Learning

Popular Learning Algorithms Used in Neural Networks

CLICK TO TWEET

(http://ctt.ec/kfQc0)

Types of Learning in Neural Network

Supervised Learning - In supervised learning, the training data is input to the network, and the desired output is known weights are adjusted until production yields desired value.

Unsupervised Learning - The input data is used to train the network whose output is known. The network classifies the input data and adjusts the weight by feature extraction in input data.

Reinforcement Learning - Here the value of the output is unknown, but the network provides the feedback whether the output is right or wrong. It is **Semi-Supervised Learning**.

Offline Learning - The adjustment of the weight vector and threshold is made only after all the training set is presented to the network. It is also called **Batch Learning.**

Online Learning - The adjustment of the weight and threshold is made after presenting each training sample to the network.

Learning Data Sets in Artificial Neural Networks

Training Set - A set of examples used for learning, that is to fit the parameters [i.e., weights] of the network. **One Epoch comprises of one full training cycle on the training set.**

Validation Set - A set of examples used to tune the parameters [i.e., architecture] of the network. For example to choose the number of hidden units in a Neural Network.

Test Set - A set of examples used only to assess the performance [generalization] of a fully specified network or to apply successfully in predicting output whose input is known.

How does Learning happen in Neural Network?

Learning occurs when the weights inside the network get updated after many iterations.

For example - Suppose we have inputs in the form of patterns for two different class of patterns - I & O as shown and b -bias and y as the desired output.

Pattern	y	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	b
I	1	1	1	1	-1	1	-1	1	1	1	1
O	-1	1	1	1	1	-1	1	1	1	1	1

We want to classify input patterns into either pattern 'I' & 'O.'

Following are the steps performed:

- Nine inputs from x_1 - x_9 along with bias b (input having weight value 1) is fed to the network for the first pattern.
- Initially, weights are initialized to zero.
- Then weights are updated for each neuron using the formulae: $\Delta w_i = x_i y$ for i = 1 to 9 (Hebb's Rule)
- Finally, new weights are found using the formulae:
- $w_i(new) = w_i(old) + \Delta w_i$
- $W_i(new)$ = [111-11-1 1111]
- The second pattern is input to the network. This time, weights are not initialized to zero. The initial weights used here are the final weights obtained after presenting the first pattern. By doing so, the network
- The steps from 1 - 4 are repeated for second inputs.
- The new weights are $W_i(new)$ = [0 0 0 -2 -2 -2 000]

So, these weights correspond to the learning ability of the network to classify the input patterns successfully.

Four Different Uses of Neural Networks

- **Classification -** A Neural Network can be trained to classify given pattern or data set into predefined class. It uses **Feedforward Networks**.

- **Prediction -** A Neural Network can be trained to produce outputs that are expected from given input. E.g., - Stock market prediction.

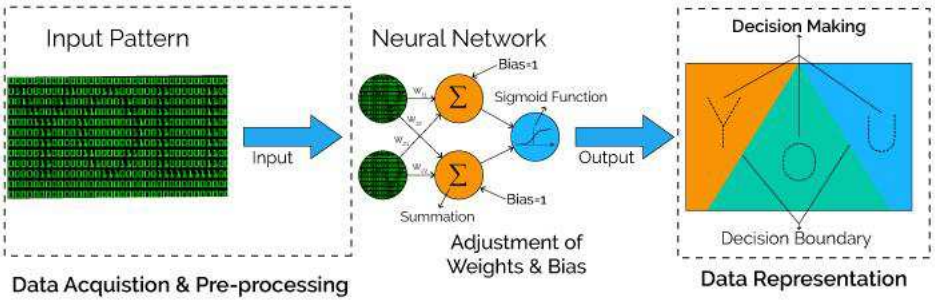
- **Clustering -** The Neural network can be used to identify a unique feature of the data and classify them into different categories without any prior knowledge of the data.

Following networks are used for clustering -

- Competitive networks
- Adaptive Resonance Theory Networks
- Kohonen Self-Organizing Maps.

- **Association -** A Neural Network can be trained to remember the particular pattern, so that when the noise pattern is presented to the network, the network associates it with the closest one in the memory or discard it. E.g. , **Hopfield Networks** which performs recognition, classification, and clustering, etc.

Neural Network for Pattern Recognition



Pattern recognition is the study of how machines can observe the environment, learn to distinguish patterns of interest from their background, and make sound and reasonable decisions about the categories of the patterns.

Some examples of the pattern are - fingerprint image, a handwritten word, human face or speech signal.

Given an input pattern, its recognition involves the following task -

- **Supervised classification** - Given input pattern is identified as the member of a predefined class.
- **Unsupervised classification** - Pattern is assigned to a hitherto unknown class.

So, the recognition problem here is essentially classification or categorized task.

The design of pattern recognition systems usually involve the following three aspects-

- Data acquisition and preprocessing
- Data representation
- Decision Making

Approaches Used For Pattern Recognition

- Template Matching

- Statistical
- Syntactic Matching
- Artificial Neural Networks

Following Neural Network architectures used for Pattern Recognition -

- Multilayer Perceptron
- Kohonen SOM (Self Organizing Map)
- Radial Basis Function Network (RBF)

Neural Networks For Pattern Recognition

CLICK TO TWEET

(http://ctt.ec/wlb2s)

Neural Network for Machine Learning

- Multilayer Perceptron (supervised classification)
- Back Propagation Network (supervised classification)
- Hopfield Network (for pattern association)
- Deep Neural Networks (unsupervised clustering)

Neural Network for Deep Learning

Following Neural Network, architectures are used in Deep Learning (https://www.xenonstack.com/data-science-machine-deep-learning-services) -

- Feed-forward neural networks
- Recurrent neural network
- Multi-layer perceptrons (MLP)
- Convolutional neural networks
- Recursive neural networks
- Deep belief networks
- Convolutional deep belief networks
- Self-Organizing Maps
- Deep Boltzmann machines
- Stacked de-noising auto-encoders

You May also Love to Read Log Analytics With Deep Learning & Machine Learning (https://www.xenonstack.com/blog/log-analytics-with-deep-learning-and-machine-learning)

Fuzzy Logic in Neural Network

Fuzzy logic refers to the logic developed to express the degree of truthiness by assigning values in between 0 and 1, unlike traditional boolean logic that represents 0 and 1.

Fuzzy logic and Neural networks have one thing in common. They can be used to solve problems of pattern recognition and other which do not involve any mathematical model.

Systems combining both fuzzy logic and neural networks are neuro - fuzzy systems.

These systems (Hybrid) can combine advantages of both neural networks and fuzzy logic to perform in a better way.

Fuzzy logic and Neural Networks have been integrated to use in following applications -

- Automotive engineering
- Applicant screening of jobs
- Control of crane
- Monitoring of glaucoma

In a hybrid (neuro-fuzzy) model, **Neural Networks Learning Algorithms** are fused with the fuzzy reasoning of fuzzy logic.

Neural network determines the values of parameters, while if-then rules are handled by fuzzy logic.

Hardware for Neural Networks

Two types of methods are used for implementing hardware for Neural Networks.

- Software simulation in conventional computer
- Special hardware solution for decreasing execution time.

When Neural Networks are used with a fewer number of processing units and weights, software simulation is performed on the computer directly. E.g. , voice recognition, etc.

When **Neural Network Algorithms** developed to the point where useful things can be done with 1000's of neurons and 10000's of synapses, high-performance Neural Networks hardware will become essential for practical operation.

E.g., GPU (Graphical processing unit) in the case of Deep learning (https://www.xenonstack.com/blog/log-analytics-with-deep-learning-and-machine-learning) algorithms in the event of object recognition, image classification, etc.

The performance of implementation is measured by connection per the second number (cps), i.e., the number of the data chunk is transported through the edges of the neural network.

While the performance of the learning algorithm is measured in the connection updates per second (cups)

Applications of Neural Network

Neural networks have been successfully applied to the broad spectrum of data-intensive applications, such as:

Application	Architecture / Algorithm	Activation Function

Process modeling and control	Radial Basis Network	Radial Basis
Machine Diagnostics	Multilayer Perceptron	Tan-Sigmoid Function
Portfolio Management	Classification Supervised Algorithm	Tan-Sigmoid Function
Target Recognition	Modular Neural Network	Tan-Sigmoid Function
Medical Diagnosis	Multilayer Perceptron	Tan-Sigmoid Function
Credit Rating	Logistic Discriminant Analysis with ANN, Support Vector Machine	Logistic function
Targeted Marketing	Back Propagation Algorithm	Logistic function
Voice recognition	Multilayer Perceptron, Deep Neural Networks(Convolutional Neural Networks)	Logistic function
Financial Forecasting	Backpropagation Algorithm	Logistic function
Intelligent searching	Deep Neural Network	Logistic function
Fraud detection	Gradient - Descent Algorithm and Least Mean Square (LMS) algorithm.	Logistic function

Advantages of Neural Network

- A neural network can perform tasks that a linear program can not.
- When an element of the neural network fails, it can continue without any problem by their parallel nature.
- A neural network learns and does not need to be reprogrammed.
- It can be implemented in any application.
- It can be performed without any problem.

Limitations of Neural Network

- The neural network needs the training to operate.
- The architecture of a neural network is different from the architecture of microprocessors, therefore, needs to be emulated.
- Requires high processing time for large neural networks.

How Can XenonStack Help You?

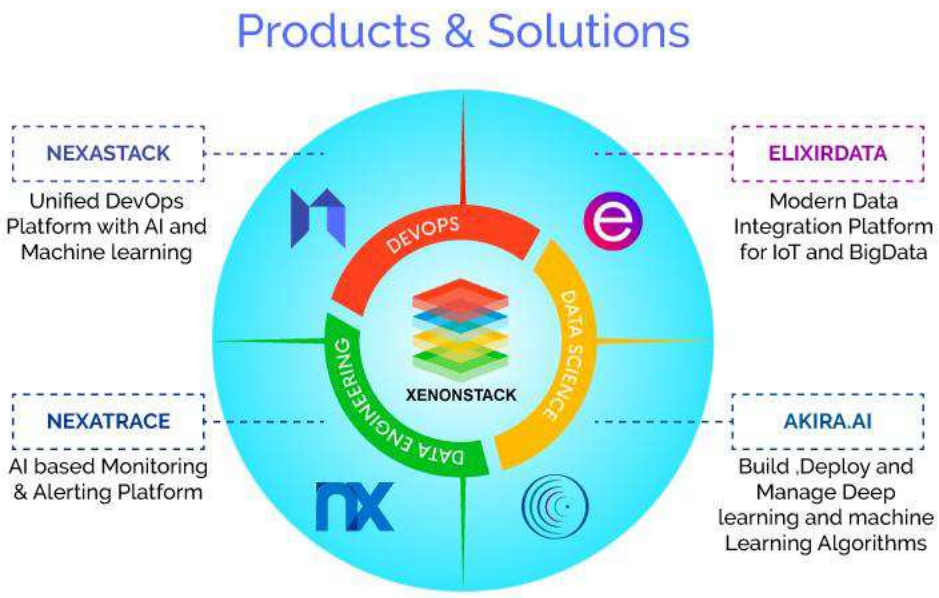
XenonStack can help you develop and deploy your model solutions based on Neural Networks. Whatever kind of problem you face - Prediction, Classification or Pattern Recognition - XenonStack has a solution for you.

Contact us With your Data Analysis, Classification and Forecasting Requirements. XenonStack Data Science Solutions (https://www.xenonstack.com/data-science-machine-deep-learning-services) provides a Platform for Data Scientists and Researchers to Build, Deploy Machine Learning and Deep Learning Algorithms at a scale with automated On-Premises and Hybrid Cloud Infrastructure.

Get in Touch (https://www.xenonstack.com/contact-for-data-science-solutions) with us for Proof of Concept, Consulting & Building Data & Artificial Intelligence Products. Talk to our Data Scientist for assessment and consulting for your Industry/Solution/Products.

About XenonStack

XenonStack is a leading Software Company in Product Development (https://www.xenonstack.com/products?utm_source=blog) and Solution Provider for DevOps (https://www.xenonstack.com/devops-consulting-services?utm_source=blog), Big Data Integration (https://www.xenonstack.com/big-data-services?utm_source=blog), Real Time Analytics (https://www.xenonstack.com/streaming-real-time-analytics-solutions?utm_source=blog), Cloud Infrastructure (https://www.xenonstack.com/cloud-hybrid-infrastructure-solutions) & Data Science (https://www.xenonstack.com/data-science-machine-deep-learning-services?utm_source=blog).



Get Free Assessment for DevOps, Big Data Strategy, Cloud, and Data Science. CONTACT US NOW (https://www.xenonstack.com/talk-with-our-experts?utm_source=blog)

(http://www.facebook.com/share.php?u=https://xenonstack.com/blog/overview-of-artificial-neural-networks-and-its-applications&title=Overview of Artificial Neural Networks and its Applications)

(http://twitter.com/home?status=Overview of Artificial Neural Networks and its Applications+https://xenonstack.com/blog/overview-of-artificial-neural-networks-and-its-applications)

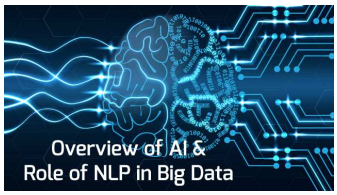
(http://www.linkedin.com/shareArticle?mini=true&url=https://xenonstack.com/blog/overview-of-artificial-neural-networks-and-its-applications&title=Overview of Artificial Neural Networks and its Applications&source=overview-of-artificial-neural-networks-and-its-applications)

(https://plus.google.com/share?url=https://xenonstack.com/blog/overview-of-artificial-neural-networks-and-its-applications)

Related Posts

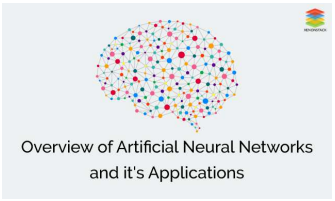
Ingestion and Processing of Data For Big Data and IoT Solutions (/blog/ingestion-processing-data-for-big-data-iot-solutions)

March 03, 2017



Overview of Artificial Intelligence and Role of Natural Language Processing in Big Data (/blog/overview-of-artificial-intelligence-and-role-of-natural-language-processing-in-big-data)

April 21, 2017



Overview of Artificial Neural Networks and its Applications (/blog/overview-of-artificial-neural-networks-and-its-applications)

May 05, 2017

Overview of Kotlin & Comparison With Java (/blog/overview-of-kotlin-comparison-between-kotlin-java)

May 17, 2017

Data Preproc
Wrangling in I
Learning and
(/blog/data-p
data-wranglir
learning-deep

May 23, 2017

XenonStack

XenonStack is a Software Company developing Productivity, Monitoring, Intelligence Platforms for DevOps, Data Integration, Analysis & Security.

Company

About Us (/about-us)

Blog (/blog)

Engagement (/)

Careers (/careers)

Events & Trainings (/events)

Services

DevOps Services (/devops-consulting-services)

Decision Support (/data-science-machine-deep-learning-services)

Big Data Analytics (/big-data-services)

IoT Platform (/xenonify-iot-platform)

Center Of Excellence (/stack-labs)

Enterprise

DataOps (/dataops)

Migration (/migration)

Log Analytics (/log-analytics)

Managed Services (/managed-services)

Streaming Analytics (/streaming-analytics)

Contact Us (/contact-us)

Email us at

business@xenonstack.com (mailto:business@xenonstack.com)

Call us at

+91 803-317-2899

ANALYTICS BY HEAP (https://heapanalytics.com/?utm_source=badge)