A Comprehensive Study on Various Neural Network Frameworks

Khadija Jabeen[1], K.Ishthaq Ahamed[2]

[1] CSE Department, G.Pulla Reddy, College of Engineering and Technology, Kurnool

[2] CSE Department, G.Pulla Reddy College of Engineering and Technology, Kurnool

***Abstract*— The traditional computation techniques of programming were not capable enough to solve "hard" problems like pattern recognition, prediction, compression, optimization, classification and machine learning. In order to solve such problems, an interest towards developing intelligent computation systems became stronger. To develop such intelligent systems, innumerable advances have been made by the researchers. Inspired by the human brain neural networks, researchers from various disciplines designed the Artificial Neural Networks (ANN). These artificial neurons are characterized on the basis of architecture, training or learning method and activation function. The neural network architecture is the arrangement of neurons to form layers and connection scheme formed in between and within the layers. Neural network architectures are broadly classified into feed-forward and feedback architectures that further contain single and multiple layers. The feed-forward networks provide a unidirectional signal flow whereas in the feedback networks the signals can flow in both the directions. These neural network architectures are trained through various learning algorithms for producing most efficient solutions to computation problems. In this paper, we present neural network architectures that play a crucial role in modeling the intelligent systems.**

***Keywords*— ANN, Multi-Layer Perceptron, Recurrent Neural Network, neuron, adaptive learning**

I.INTRODUCTION

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem [2]. Neural networks learn by example [1]. They have emerged in the past few years as an area of unusual opportunity for research, development and application to a variety of real world problems. Neural network consists of processing elements that form networks with weighted functions for every input. These elements are generally arranged into a sequence of layers with several connections between them. The structure of neural networks has three types of layers: an **input layer** that receives data from external sources, **hidden layer** that performs computation on the basis of function provided, and an **output layer** that generates output based on the input provided.

These elements usually run in parallel and are connected to each other using connection links. The connection links have weights that contain data related to input signal. Neurons use this data to solve a specific problem. The behavior of a neural network is described by their capability of learning, analyzing and generalizing training data having resemblance to that of a human brain. These networks consist of largely interconnected processing neurons that are inherently parallel in nature. The below figure represents the difference between a biological and artificial neuron:

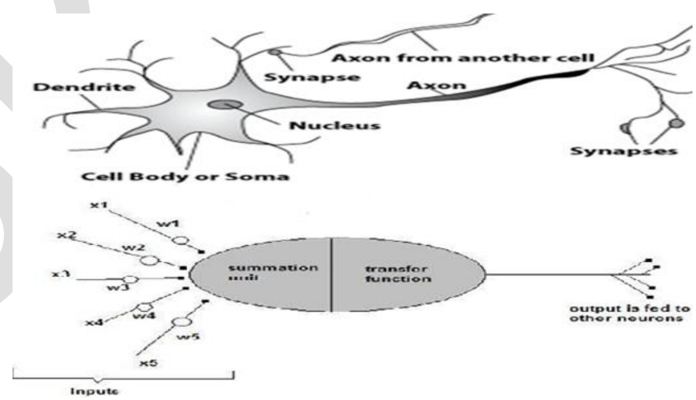


Fig 1 A biological versus artificial neuron

II. PROPERTIES OF ANN

The two major characteristics which differentiate neural networks from artificial intelligence and traditional computing or processing are **1) learning by example 2) distributed associative memory.**

A. *Learning by Example*

The neural networks have the ability to learn by an example. They cannot be organized to perform a particular task. The examples that are used for learning should be selected carefully such that it doesn't lead to wastage of time or incorrect working of the network. Instead of programming the system is developed through learning.

B. *Distributed associative memory*

The neural network structure can be largely distributed and associated in parallel. A unit of knowledge is distributed across all the weighted links in the network. Upon training a network it is provided with an input, the network then chooses the nearest match to that input in its memory and generates an output that is similar to the full output.

These characteristics enable Artificial Neural Networks to generate solutions for typical problems that are difficult to manage by the traditional ways of problem solving. Using these characteristics an ANN has the following benefits:

1) Adaptive Learning: An ability to learn how to do tasks based on the data given for training or initial experience. The ANN is capable of determine the relationship between the different examples which are presented to it, or to identify the kind to which belong, without requiring a previous model. [3]

2) Self-Organisation: An ANN can create its own organisation or representation of the information it receives during learning time. This property allows the ANN to distribute the knowledge in the entire network structure; there is no element with specific stored information. [3]

3) Fault Tolerance: This characteristic is shown in two aspects: The first is related to the samples shown to the network, in which case it answers correctly even when the examples exhibit variability or noise; the second, appears when in any of the elements of the network occurs a failure, which does not impossibilitate its functioning due to the way in which it stores information.

4) Flexibility: Neural networks are flexible in a changing environment. Although neural networks may take some time to learn a sudden drastic change they are excellent at adapting to constantly changing information. [1]

5) *Performance:* Performance of neural networks is at least as good as classical statistical modeling, and better on most problems. The neural networks build models that are more reflective of the structure of the data in significantly less time.

III. Real Time Applications of ANN

A few of the recent applications of neural networks are given below to illustrate the wide spectrum of applications to which neural networks have been applied.

A. Complex system modeling

A system with multiple inputs and outputs can be modeled using a neural network by applying the system inputs to the network and using the system outputs as the desired outputs of the neural network. Such modeling can be used on physical systems, business and financial systems, or social systems. Current applications include the use of a neural network to determine whether loan applications should be approved using the previous five years experience of that bank as the input training data.

B. Image or data compression

It involves the transforming of image data to a different representation that requires less memory. Then the image must be reconstructed from this new representation in such a way that there is an imperceptible difference from the original. [4]

C. Character recognition

It is the process of visually interpreting and classifying symbols. Neural networks were the first systems to efficiently read Japanese Kanji characters. This it effectively broke the input barrier for computers used in Japan.

D. Noise filtering

Neural networks are able to filter noisy data and preserve a greater depth of structure and detail than any of the traditional filters while still removing the noise. Applications include removal of background noise from voice communications and separation of the fetal heart beat from a mother's heart beat. [4]

E. Servo-control systems

These are the systems used in robots, which should compensate for physical variations in the system introduced by misalignments in the axes, or any kind of changes in robots due to bending and stretching. These are extremely difficult to describe analytically. A neural network can be trained to predict and respond to these errors in the final position of a robot.

F. Text-to-speech conversion

In this application the printed symbols or letters in a text were converted into the spoken language using a neural network that taught itself to translate written text into speech in the same way that a human child learns to read.

IV. Neural Network Frameworks

The major architectures that play a crucial role in neural networks are classified as below:

A. Single input neuron

A single-input neuron is shown in Fig. 2. The scalar input p is multiplied by the scalar weight W to form Wp, one of the terms that is sent to the summer. The other input, 1, is multiplied by a bias b and then passed to the summer. The summer output n often referred to as the net input, goes into a transfer function f which produces the scalar neuron output a (sometimes "activation function" is used rather than transfer function and offset rather than bias). [5]

From Fig. 2, both w and b are both adjustable scalar parameters of the neuron. Typically the transfer function is chosen by the designer and then the parameters w and b will be adjusted by some learning rule so that the neuron input/output relationship meet some specific goal.

The transfer function in Fig.2 may be a linear or nonlinear function of n. A particular transfer function is chosen to satisfy some specification of the problem that the neuron is attempting to solve. [5] If we relate the below structure to biological neuron, the weight **w** is similar to strength of synapse, summation represents the cell body, the signal on axon is represented by neuron output **"a"** and the activation function.

The output of the neuron is calculated using:

$$\mathbf{a = f\ (wp + b)}$$

For example, let w=4, p=3 and b= -1.2, then:

$$\mathbf{a = f\ (4(3)-1.2) = f\ (10.8)}$$

The output value depends on the chosen activation function. The bias is similar to the weight and its constant value is 1. Bias and weight are the adjustable parameters of the neuron. These parameters can be adjusted using learning rules.

The below Fig.2 shows architecture of single-input neuron:

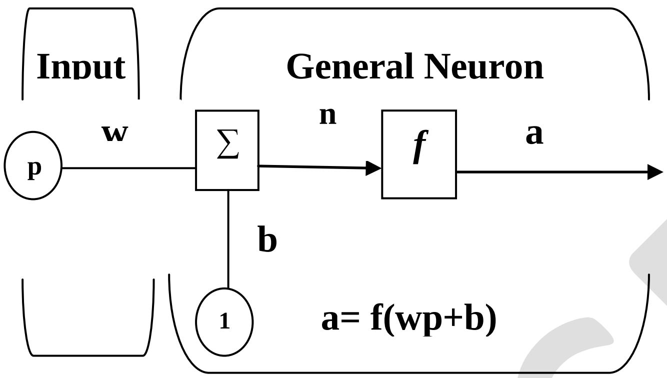


Fig. 2 Single-Input Neuron

B. Multiple inputs neuron

Usually, a neuron can have more than one input. The input neurons P_1, P_2, P_3,….,P_R are associated with weights $w_{1,1}$,$w_{1,2}$,…,$w_{1,R}$ of the weight matrix W. A particular convention in assigning the indices of the elements of the weight matrix has been adopted. The first index indicates the particular neuron destination for the weight. The second index indicates the source of the signal fed to the neuron. Thus, the indices in W1,2 say that this weight represents the connection to the first (and only) neuron from the second source. [3]

The below Fig.3 shows a multiple input neuron with R inputs:

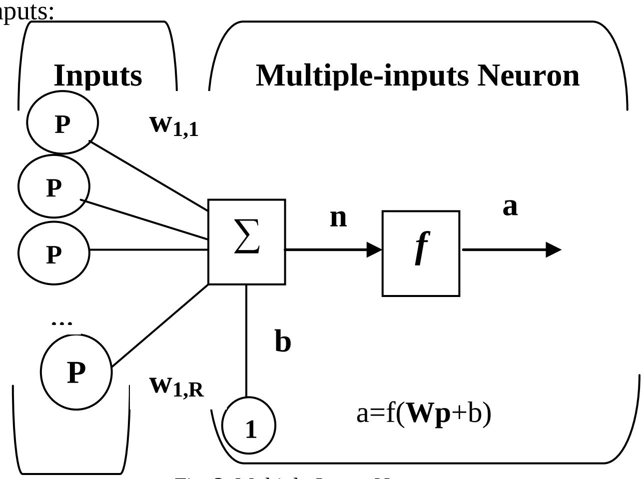


Fig. 3 Multiple-Inputs Neuron

C. Multi-Layer Feed-forward Network

The Multi-Layer Perceptron or feed-forward neural network is perhaps the most popular network architecture in use today. The units each perform a biased weighted sum of their inputs and pass this activation level through an activation function to produce their output, and the units are arranged in a layered feed forward topology. The network thus has a simple interpretation as a form of input-output model, with the weights and thresholds (biases) the free parameters of the model. [6]

Generally it may not be sufficient to use one neuron, even with many inputs. We might need five to ten neurons that operate in parallel, which is called as a "layer". The following Fig.4 represents Multi-layer feed-forward architecture:

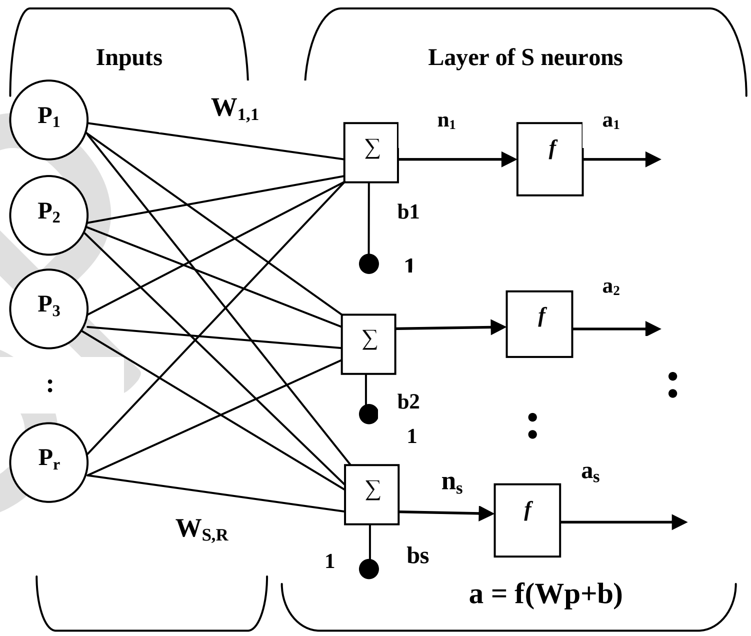


Fig. 4 Multi -Layer Feed-forward Network (Multi-Layer Perceptron)

The layer consists of summing units, activation functions, bias **b,** weight matrix **W**, and output vector **a**. Each component of the input **P** is connected to each neuron through weight matrix **W**. Each neuron has an activation function ***f***, bias $\mathbf{b}_S$ and an output $\mathbf{a}_S$. The number of inputs to a layer can be different from the number of neurons. The neurons in a layer can have different activation functions by combining two of networks each network can generate some outputs. The input components enter the network through the weight matrix **W**:

$$\mathbf{W}= \begin{matrix} w1,1 & w1,2 & w1,R \\ w2,1 & w2,2 & w2,R \\ wS,1 & wS,2 & wS,R \end{matrix}$$

The column of weight matrix represent source of the input and the row of weight matrix represent destination neuron associated with weight.

D. Multi-Layer Feedback network: (Recurrent Network)

In the neural network literature, neural networks with one or more feedback loops are referred to as recurrent networks. A recurrent network distinguishes itself from a feed forward neural network in that it has at least one feedback loop. Such a system has very rich temporal and spatial behaviors, such as stable and unstable fixed points and limit cycles, and chaotic behaviors. These behaviors can be utilized to model certain cognitive functions, such as associative memory, unsupervised learning, self-organizing maps, and temporal reasoning [7]. Recurrent networks are more powerful than feed forward networks in terms of their potential and can depict temporal behavior.

In this network the initial conditions are supplied by input vector **p**, a(0)= p. Then further outputs of the network are computed from previous outputs:

a(1) = satlins(Wa(0)+b) a(2)=satlins(Wa(1)+b)

1) Symmetric Recurrent Network: In symmetric recurrent network, the connections are symmetric, that is, the connection weights from unit i to unit j and from unit j to unit i are identical for all i and j. The widely known Hopfield networks are a kind of symmetric recurrent networks.

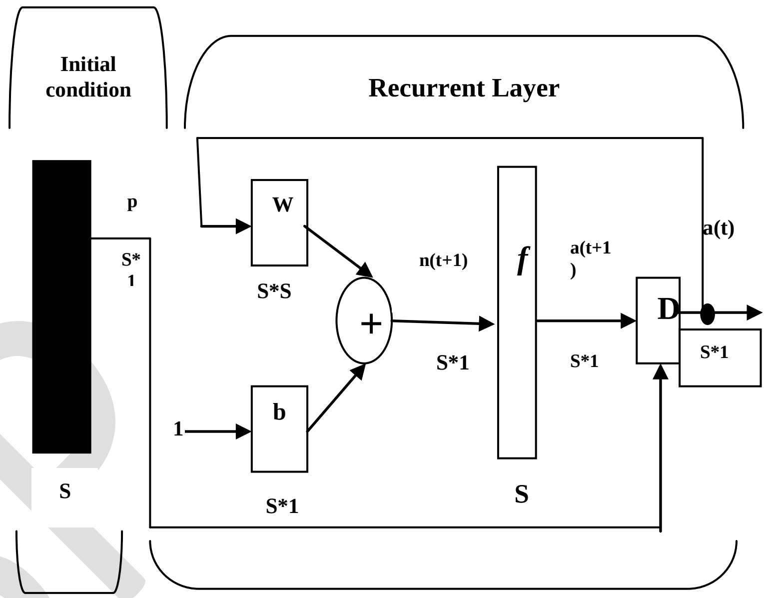
2) Asymmetric Recurrent Network: The dynamic behavior of asymmetric networks includes *limit cycles* and *chaos*, and these networks are capable of storing or generating temporal sequences of spatial patterns. Chaos in a recurrent neural network is characterized by a time evolution that progresses through a set of distorted patterns in a notably irregular manner.

3) Fully Recurrent Network: The main example of implementation of feedback is the classical fully recurrent neural network, i.e. a single layer of neurons fully interconnected with each other or several such layers [10]. They are very general architectures which can model a large class of dynamical systems, but on specific problems simpler dynamic neural networks which make use of available prior knowledge can be better.

4) Locally Recurrent Network: The newest approach to the temporal processing by neural networks is realized by Locally Recurrent Neural Networks (LRNNs) or Local Feedback Multi-Layer Networks (LF-MLN).

Diagonal Recurrent Neural Networks (DRNN), have been proposed for dynamic systems control, claiming relevant results [8]. This architecture is also a particular case of output feedback MLN since DRNN is a two layer network with static linear output neurons and dynamic hidden neurons with static synapses but with one delay feedback from the output. RNNs constitute a very powerful class of computational models, capable of instantiating almost arbitrary dynamics. RNNs are especially promising for tasks that require to learn how to use memory. The ability to map real-valued input- sequences to real-valued output sequences, making use of their internal state to incorporate past context, makes them remarkably general process sequencing devices [9]. Potential applications are: time series prediction, time series production and time series classification or labelling.

A discrete-time recurrent network is shown in Figure 5



a(0)=p a(t+1)=satlins(Wa(t)+b)

Fig. 5 Multi-Layer feedback Network (Recurrent Network)

V. CONCLUSIONS

Thus, from this paper it is evident that the neural network architectures play a very important role in modeling the intelligent systems. The feed-forward and feedback architectures make use of various learning algorithms and paradigms for obtaining these outputs. They act as a backbone of the entire learning process which provides the end user with the desired result. For this purpose the neural network architectures are trained through various learning algorithms for producing most efficient solutions to computation problems.

REFERENCES

[1]. Koushal Kumar, Gour Sundar Mitra Thakur, (2012). Advanced Applications of Neural Networks and Artificial Intelligence: A Review. *I.J. Information Technology and Computer Science, 2012, 6, 57-58.*

[2]. Sivanandam SN, Deepa SN., (2007). Principles of Soft Computing. Wiley India.

[3]. Haykin, S., (1999). Neural Networks: A Comprehensive Foundation, 2nd Edn., New Jersey: Prentice-Hall.

[4]. Robert E. Uhrig., (1995). "Introduction to Artificial Neural Networks", IEEE Transactions on Industrial Electronics, Control, and Instrumentation, vol. 1, pp. 33-37.

[5]. Anabtawi, M.Z. and jamal M. Nazzal., (1994). Multilayer Perceptron Neural Networks (MLPs) For Analyzing the Properties of Jordan Oil, pp: 175-178.

[6]. A.B.Karthick Anand Babu., (2012). Design and Development of Artificial Neural Network Based Tamil Unicode Symbols Identification System. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, January 2012.

[7]. Qin He., (1999). Neural Network and Its Application in IR. University of Illinois at Urbana-Champaign Spring.

[8]. Ku C.-C. and Lee K.Y., 1995, "*Diagonal Recurrent Neural Networks for Dynamic Systems Control*", IEEE Trans. on Neural Networks vol. 6, no.1, January 1995.

[9]. Bengio, y., Frasconi, P., Gori, M., & G.Soda. (1993). Recurrent neural networks for adaptive temporal processing. In Proceedings of the 6th Italian workshop on parallel architectures and neural networks wirn93 (pp. 85-117). Vietri (Italy): World Scientific Pub.

[10]. Williams R.J. and Zipser D., 1989a, "*A Learning Algorithm for Continually Running Fully Recurrent Neural Networks*", Neural Computation 1: 270-280.