Neural Networks – State of Art, Brief History, Basic Models and Architecture

Bohdan Macukow

Neural Networks – State of Art, Brief History, Basic Models and Architecture

Bohdan Macukow

Warsaw University of Technology, Faculty of Applied Mathematics and Information Science, 00-662 Warsaw, Koszykowa 75, Poland, B.Macukow@mini.pw.edu.pl

Abstract. The history of neural networks can be traced back to the work of trying to model the neuron. Today, neural networks discussions are occurring everywhere. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A brief history of the neural networks research is presented and some more popular models are briefly discussed. The major attention is on the feed-forward networks and specially to the topology of such the network and method of building the multi-layer perceptrons.

Keywords: neural networks, history, building feedforward net

1 What is a Neural Network?

An Artificial Neural Network (ANN) is an information or signal processing system composed of a large number of simple processing elements which are interconnected by direct links and which cooperate to perform parallel distributed processing in order to solve a desired computational task. Neural networks process information in a similar way the human brain does. ANN is inspired by the way the biological nervous systems, such as the brain works - neural networks learn by example.

ANN takes a different approach to problem solving than that of conventional computing. Conventional computer systems use an algorithmic approach i.e. follow a set of instructions in order to solve a problem. That limits the problem solving capability to problems that we already understand and know how to solve. However, neural networks and conventional algorithmic computing are not in competition but complement each other. There are the tasks that are more suited to an algorithmic approach like arithmetic operations and the tasks that are more suited to neural networks approach.

2 History

The history of neural networks can be divided into several periods.

2.1 First Attempts

The first step toward artificial neural networks came in 1943 when Warren McCulloch, a neurophysiologist and a young mathematician, Walter Pitts, developed the first models of neural networks. They wrote a paper *The Logical Calculus of the Ideas Immanent in Nervous Activity* on how neurons might work [1]. Their networks were based on simple elements which were considered to be binary devices with fixed thresholds. The results of their model were simple logic functions with "all-or-none" character of nervous activity.

In 1944 Joseph Erlanger together with Herbert Spencer Gasser identified several varieties of nerve fiber and established the relationship between action potential velocity and fiber diameter.

In 1949, Donald Hebb a psychologist, wrote *The Organization of Behavior* [2], a work which pointed out the fact that neural pathways are strengthened each time they are used, a concept fundamentally essential to the ways humans learn.

2.2 Promising & Emerging Technology

In 1958, Frank Rosenblatt a psychologist, conducted an early work on perceptrons [3]. The Perceptron was an electronic device that was constructed in accordance with biological principles and showed an ability to learn. He also wrote an early book on neurocomputing, *Principles of Neurodynamics* [4].

Another system was the ADALINE (ADAptive LInear Element) which was developed in 1960 by two electrical engineers Bernard Widrow and Marcian Hoff [5]. The method used for learning was different to that of the Perceptron, it employed the Least-Mean-Squares learning rule. In 1962, Widrow & Hoff developed a learning procedure that examines the value before the weight adjusts it.

2.3 Period of Frustration & Disgrace

Following an initial period of enthusiasm, the field survived a period of frustration and disgrace.

In 1969 Marvin Minsky and Seymour Papert wrote a *book Perceptrons: An Introduction to Computational Geometry* [6]. It was a part of a campaign to discredit neural network research showing a number of fundamental problems, and in which they generalized the limitations of single layer perceptron. Although the authors were well aware that powerful perceptrons have multiple layers and Rosenblatt's basic feedforward perceptrons have three layers, they defined a perceptron as a two-layer machine that can handle only linearly separable problems and, for example, cannot solve the exclusive-OR problem.

Because the public interest and available funding becoming minimal, only several researchers continued working on the problems such as pattern recognition. But, during this period several paradigms were generated which modern work continues to enhance.

Henry Klopf in 1972, developed a basis for learning in artificial neurons based on a biological principle [7]. Paul Werbos in 1974 developed the back-propagation learning method although its importance wasn't fully appreciated until a 1986.

Kunihiko Fukushima developed a stepwise trained multilayered neural network for the interpretation of handwritten characters. The original work *Cognitron: A self-organizing multilayered neural network* [8] was published in 1975.

In 1976 Stephen Grossberg in the paper *Adaptive pattern classification and universal recoding* [9] introduced the adaptive resonance as a theory of human cognitive information processing.

2.4   Innovation

In 1980s several events caused a renewed interest. Teuvo Kohonen has made many contributions to the field of artificial neural networks. He introduced the artificial neural network sometimes called a Kohonen map or network [10].

John Joseph Hopfield of Caltech in 1982 presented a paper *Neural Networks and Physical Systems with Emergent Collective Computational Abilities* [11]. Hopfield describe the recurrent artificial neural network serving as content-addressable memory system. His works persuaded hundreds of highly qualified scientists, mathematicians, and technologists to join the emerging field of neural networks.

The backpropagation algorithm, originally discovered by Werbos in 1974 was rediscovered in 1986 with the book *Learning Internal Representation by Error Propagation* by Rumelhart, Hinton and Williams [12]. Backpropagation is a form of a gradient descent algorithm used with artificial neural networks for error minimization.

By 1985 the American Institute of Physics began what has become an annual meeting - *Neural Networks for Computing*. In 1987, the first open conference on neural networks in modern times; the *IEEE International Conference on Neural Networks* was held in San Diego, and the *International Neural Network Society (INNS)* was formed. In 1988 the INNS journal *Neural Networks* was founded, followed by *Neural Computation* in 1989 and the *IEEE Transactions on Neural Networks* in 1990.

Gail Carpenter and Steve Grossberg in 1987 in *A massively parallel architecture for a self-organizing neural pattern recognition machine* [13] described the ART1 an unsupervised learning model specially designed for recognizing binary patterns.

2.5   Today

Significant progress has been made in the field of neural networks - enough to attract a great deal of attention and fund further research. Today, neural networks discussions are occurring everywhere. Advancement beyond the current commercial applications appears to be possible, and research is advancing the field on many fronts. Chips based on the neural theory are emerging and applications to complex problems developing. Clearly, today is a period of transition for neural network technology.

Between 2009 and 2012, the recurrent neural networks and deep feedforward neural networks were developed in the research group of Jürgen Schmidhuber [14].

In 2014 the scientists from IBM introduced the processor (TrueNorth), with the architecture similar to that existing in the brain. IBM presented the integrated circuit the size of postage stamp able to simulate the work of the millions of neurons and 256 million of synapses in a real time. The system is able to execute from 46 to 400 billion synaptic operations per second.

3 Different models of neural networks

A neural network can be thought of as a network of "neurons" organized in layers. The number of types of Artificial Neural Networks (ANNs) and their uses can potentially be very high. Since the first neural model by McCulloch and Pitts there have been developed hundreds of different models considered as ANNs. The differences in them might be the functions, the accepted values, the topology, the learning algorithms, etc. Also there are many hybrid models. Since the function of ANNs is to process information, they are used mainly in fields related to it. An ANN is formed from single units, (artificial neurons or processing elements - PE), connected with coefficients (weights), which constitute the neural structure and are organized in layers. The power of neural computations comes from connecting neurons in a network. Each PE has weighted inputs, transfer function and one output. The behavior of a neural network is determined by the transfer functions of its neurons, by the learning rule, and by the architecture itself. The weights are the adjustable parameters and, in that sense, a neural network is a parameterized system. The weighed sum of the inputs constitutes the activation of the neuron.

An ANN is typically defined by three types of parameters:

1. The interconnection pattern between the different layers of neurons.
2. The learning process for updating the weights of the weights.
3. The activation function that converts a neuron's weighted input to its output activation.

How should the neurons be connected together? If a network is to be of any use, there must be inputs and outputs. However, there also can be hidden neurons that play an internal role in the network. The input, hidden and output neurons need to be connected together. A simple network has a feedforward structure: signals flow from inputs, forwards through any hidden units, eventually reaching the output units. However, if the network is recurrent (contains connections back from later to earlier neurons) it can be unstable, and has a very complex dynamics. Recurrent networks are very interesting to researchers in neural networks, but so far it is the feedforward structures that have proved most useful in solving real problems.

3.1 Multilayer Perceptrons

This is perhaps the most popular network architecture in use today (Fig.1). The units each perform a biased weighted sum of their inputs and pass this activation level through a transfer function to produce their output, and the units are arranged in a layered feedforward topology.

3.2 ADALINE

Adaptive Linear Neuron or later Adaptive Linear Element (Fig.2) is an early single-layer artificial neural network and the name of the physical device that implemented this network. It was developed by Bernard Widrow and Ted Hoff of Stanford Univer-

sity in 1960. It is based on the McCulloch–Pitts neuron. It consists of a weight, a bias and a summation function. The difference between Adaline and the standard (McCulloch–Pitts) perceptron is that in the learning phase the weights are adjusted according to the weighted sum of the inputs (the net). In the standard perceptron, the net is passed to the activation (transfer) function and the function's output is used for adjusting the weights. There also exists an extension known as Madaline.
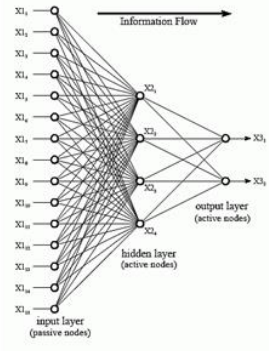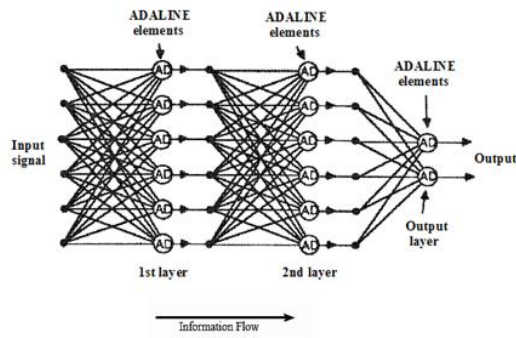

Fig.1

Fig.2

3.3 ART

The primary intuition behind the ART model (Fig.3) is that object identification and recognition generally occur as a result of the interaction of 'top-down' observer expectations with 'bottom-up' sensory information. The model postulates that 'top-down' expectations take the form of a memory template or prototype that is then compared with the actual features of an object as detected by the senses. This comparison gives rise to a measure of category belongingness. As long as this difference between sensation and expectation does not exceed a set threshold called the 'vigilance parameter', the sensed object will be considered a member of the expected class. The system thus offers a solution to the 'plasticity/stability' problem, i.e. the problem of acquiring new knowledge without disrupting existing knowledge.
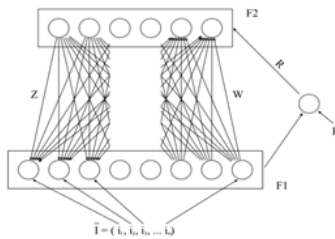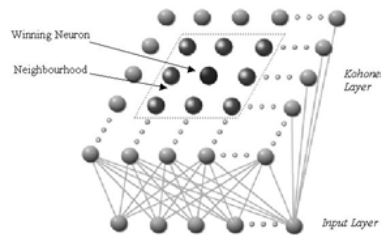

Fig.3

Fig.4

3.4 Self-Organizing Feature Map

SOFM or Kohonen networks (Fig.4) are used quite differently. Whereas most of other networks are designed for supervised learning tasks, SOFM networks are designed primarily for unsupervised learning. Whereas in supervised learning the training data set contains cases featuring input variables together with the associated outputs (and the network must infer a mapping from the inputs to the outputs), in unsupervised learning the training data set contains only input variables.

3.5 A Hopfield network

A Hopfield network (Fig.5) is a form of a recurrent artificial neural network popularized by John Hopfield in 1982. Hopfield nets serve as content-addressable memory systems with binary threshold nodes. They are guaranteed to converge to a local minimum, but convergence to a false pattern (wrong local minimum) rather than the stored pattern (expected local minimum) can occur. Hopfield networks also provide a model for understanding human memory.

3.6 The Simple Recurrent Network

SRN or Elman network (Fig.6) it is really just a three-layer, feed-forward back propagation network. The only proviso is that one of the two parts of the input to the net work is the pattern of activation over the network's own hidden units at the previous time step.
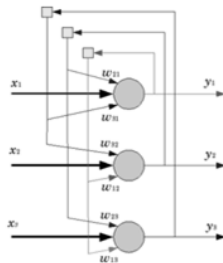

Fig.5   Fig.6

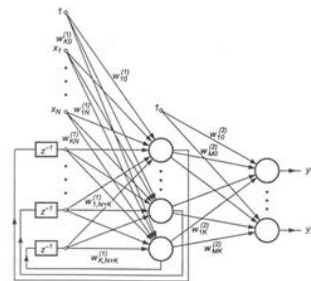### 3.7 CNN

Cellular neural networks (Fig.7) are a parallel computing paradigm similar to neural networks, with the difference that communication is allowed between neighbouring units only. CNN main characteristic is the locality of the connections between the units. Each cell has one output, by which it communicates its state with both other cells and external devices.
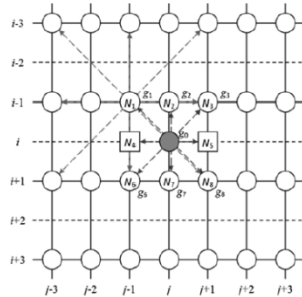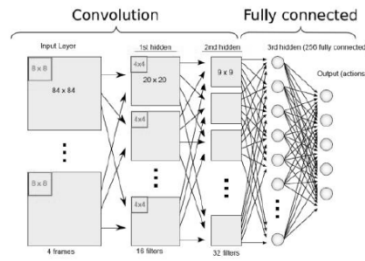
Fig.7   Fig.8

3.8   Convolutional Neural Network

A convolutional neural network (Fig.8) is a type of feed-forward artificial neural network whose individual neurons are arranged in such a way that they respond to overlapping regions tiling the visual field. Convolutional neural networks consist of multiple layers of small neuron collections which process portions of the input image. The outputs of these collections are then tiled so that their input regions overlap, to obtain a better representation of the original image; this is repeated for every such layer.

4   Topology, in other words – network architecture

In the Feed-Forward Artificial Neural Networks scheme, the data moves from the input to the output units in a strictly feed-forward manner. Data processing may spawn multiple layers, but no feedback connections are implemented. Examples of feed-forward ANN's would be a Perceptron (Rosenblatt) or an Adaline (Adaptive Linear Neuron) based net.

Recurrent ANN's. These types of ANN's incorporate feedback connections. Compared to feed-forward ANN's, the dynamic properties of the network are paramount. In some circumstances, the activation values of the units undergo a relaxation process so that the network evolves into a stable state where these activation values remain unchanged. Examples of recurrent ANN's would be a Kohonen (SOM) or a Hopfield based solution.

4.1   Inconsistency in nomenclature

What is a layer? Some authors refer to the number of layers of variable weights but some authors describe the number of layers of nodes. Usually, the nodes in the first layer, the input layer, merely distribute the inputs to subsequent layers, and do not perform any operations (summation or thresholding). NB.. some authors miss out these nodes.

A layer - it is the part of network structure which contains active elements performing some operation.

A multilayer network (Fig.9) receives a number of inputs. These are distributed by a layer of input nodes that do not perform any operation – these inputs are then passed along the first layer of adaptive weights to a layer of perceptron-like units, which do sum and threshold their inputs. This layer is able to produce classification lines in pattern space. The output from the first hidden layer is passed to the second hidden layer. This layer is able to produce classification convex area etc.
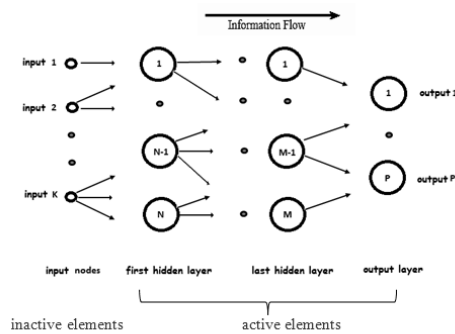

Fig.9

5 How to build a feedforward net?

The main problems faced building a feedforward network (without feedback loops) are:.

- linear or nonlinear network?
- how many layers is necessary for the proper network`s work?
- how many elements have to be in these layers?

A linear network it is a network where input signals are multiplied by the weights, added, and the result follows to the axon as the output signal of the neuron. Eventually some threshold can be used. Typical examples of a linear network are a simple perceptron and an Adeline network.

In a nonlinear network the output signal is calculated by a nonlinear function f( ). The function f(?) is called neuron transfer function and its operations have to be similar to the operations of a biological neuron. Typical example of a nonlinear network is a sigmoidal network.

5.1 How many layers?

The simplest feedforward network has at least two layers – an input and an output (NB. such a networks are called single layer networks – active neurons are located

only in an output layer). Usually between these layers there are multiple intermediate or hidden layers.

Hidden layers are very important they are considered to be categorizers or feature detectors. The output layer is considered a collector of the features detected and producer of the response

5.1.1 The Input Layer

With respect to the number of neurons comprising this layer, this parameter is completely and uniquely determined once you know the shape of your training data. Specifically, the number of neurons comprising that layer is equal to the number of features (columns) in your data. Some neural networks configurations add one additional node for a bias term.

5.1.2 The Output Layer

Like the input layer, every neural network has exactly one output layer. Determining its size (number of neurons) is simple; it is completely determined by the chosen model configuration. The interesting solution is called „one out of N". Unfortunately, because of limited accuracy in network operation the non-zero signal can occur on each out elements. It is necessary to implement the special criteria for results post-processing and threshold of acceptance and rejection.

5.2   How to build the network

Too small network without hidden layer or too few neurons is unable to solve a problem and even the very long learning time will not help.

Too big network will cheat the teacher. Too many hidden layers or too many elements in the hidden layers yields to the simplification of task. The network will learn whole set of the learning patterns. It learns very a fast and precisely but is completely useless for solving any similar problem.

5.3   How many hidden layers

Too many hidden layers yield to a significant deterioration of learning. There is a consensus as to the performance difference due to additional hidden layers: the situations in which performance improves with a second (or third, etc.) hidden layer are relatively infrequent. One hidden layer is sufficient for the large majority of problems. An additional layer yields the instability of gradient, and increases the number of false minima. Two hidden layer are necessary only if the learning refers the function with points of discontinuity. Too many neurons in the hidden layers may result in overfitting. Overfitting occurs when the neural network has so much information processing capacity that the limited amount of information contained in the training set is not enough to train all of the neurons in the hidden layers. Another problem can occur even when the training data is sufficient. An inordinately large number of neurons in the hidden layer may increase the time it takes to train the network and may lead to the increase of errors (Fig.10). Using too few neurons in the hidden layers will, in turn, result in something called underfitting.

A rough prerequisite for the number of hidden neurons (for most of typical problems) is the rule of a geometric pyramid. The number of neurons in the consecutive layers has a shape of a pyramid and decrease from the direction of input to the output. The numbers of neurons in a consecutive layers are forming a geometric sequence
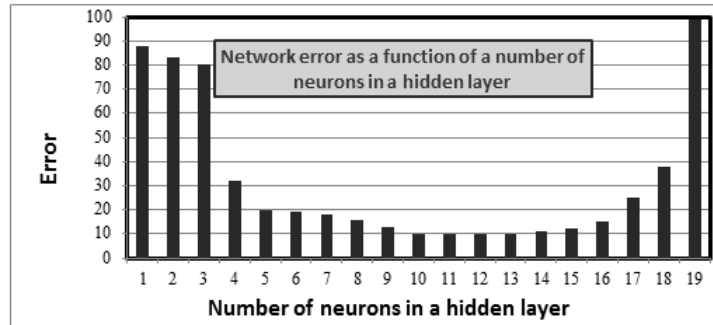

Fig.10

For example, for the network with one hidden layer with n-neurons in the input layer and m-neurons in the output layer, the numbers of neurons in the hidden layer should be NHN = $\sqrt{n * m}$. For the network with two hidden layers NHN1 = m*r2 and NHN2 = m*r where r = $\sqrt[3]{\dfrac{n}{m}}$.

The hidden neuron can influence the error in the nodes to which its output is connected. The stability of neural network is estimated by error. The minimal error denotes better stability, and higher error indicates worst stability. During the training, the network adapts in order to decrease the error emerging from the training patterns. Many researchers have fixed number of hidden neurons based on trial rule.

The estimation theory was proposed to find a number of hidden units in the higher order feedforward neural network. This theory is applied to the time series prediction. The determination of an optimal number of hidden neurons is obtained when the sufficient number of hidden neurons is assumed. According to the estimation theory, the sufficient number of hidden units in the second-order neural network and the first-order neural networks are 4 and 7 respectively.

6   Reviews of methods how to fix a number of hidden neurons

To festablish the optimal(?) number of hidden neurons, for the past 20 years more than 100 various criteria have been tested based on the statistical errors. The very good review was done by K. Gnana Sheela and S. N. Deepa in [15]. Below there is a short review of some endeavours:

- **1991:** Sartori and Antsaklis proposed a method to find the number of hidden neurons in multilayer neural network for an arbitrary training set with P training patterns.

- **1993**: Arai proposed two parallel hyperplane methods for finding the number of hidden neurons
- **1995:** Li et al. investigated the estimation theory to find the number of hidden units in the higher order feedforward neural network
- **1997:** Tamura and Tateishi developed a method to fix the number of hidden neuron. The number of hidden neurons in three layer neural network is *N -1* and four-layer neural network is *N/2+3* where *N* is the input-target relation.
- **1998:** Fujita proposed a statistical estimation for the number of hidden neurons. The merits of this method are speed learning. The number of hidden neurons mainly depends on the output error.
- **2001:** Onoda presented a statistical approach to find the optimal number of hidden units in prediction applications. The minimal errors are obtained by the increase of number of hidden units. Md. Islam and Murase proposed a large number of hidden nodes in weight freezing of single hidden layer networks.
- **2003:** Zhang et al. implemented a set covering algorithm (SCA) in three-layer neural network. The SCA is based on unit sphere covering (USC) of hamming space. This methodology is based on the number of inputs.
- **2006:** Choi et al. developed a separate learning algorithm which includes a deterministic and heuristic approach. In this algorithm, hidden-to-output and input-to-hidden nodes are separately trained. It solved the local minima in two-layered feedforward network. The achievement here is the best convergence speed.
- **2008:** Jiang et al. presented the lower bound of the number of hidden neurons. The necessary numbers of hidden neurons approximated in hidden layer using multi-layer perceptron (MLP) were found by Trenn. The key points are simplicity, scalability, and adaptivity. The number of hidden neurons is $N_h = n + n_0 - 0.5$ where n is the number of inputs and $n0$ is the number of outputs. Xu and Chen developed a novel approach for determining the optimum number of hidden neurons in data mining. The best number of hidden neurons leads to minimum root means Squared Error.
- **2009:** Shibata and Ikeda investigated the effect of learning stability and hidden neurons in neural network. The simulation results show that the hidden output connection weight becomes small as number of hidden neurons increases.
- **2010:** Doukim et al. proposed a technique to find the number of hidden neurons in MLP network using coarse-to-fine search technique which is applied in skin detection. This technique includes binary search and sequential search. Yuan et al. proposed a method for estimation of hidden neuron based on information entropy. This method is based on decision tree algorithm. Wu and Hong proposed the learning algorithms for determination of the number of hidden neurons.
- **2011:** Panchal et al. proposed a methodology to analyse the behaviour of MLP. The number of hidden layers is inversely proportional to the minimal error.

References

1. McCulloch, W. S., Pitts, W.: A Logical Calculus of the Ideas Immanent in Nervous Activity. Bull. Math. Bioph., 5, 115--133 (1943)
2. Hebb, D.: The Organization of Behavior. New York: Wiley & Sons, (1949)
3. Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, 65, No. 6, 386--408 (1958)
4. Rosenblatt, F.: Principles of Neurodynamics. Washington, Spartan Books, (1962)
5. Widrow, B., Hoff, M.: Adaptive switching circuits, Tech.Rep.1553-1, Stanford Electron. Labs., Stanford, CA, June (1960)
6. Minsky, M.L., Papert, S.: Perceptrons: An Introduction to Computational Geometry. Cambridge, MA: MIT Press (1969)
7. Klopf, A. H.: Brain Function and Adaptive Systems - A Heterostatic Theory. Air Force Research Laboratories Technical Report, AFCRL-72-0164 (1972)
8. Fukushima, K.: Cognitron: A self-organizing multilayered neural network. Biol. Cyber. 20, 121--136 (1975)
9. Grossberg, S.: Adaptive Pattern Classification and Universal Recoding, Biol. Cyber. 23/3, 121--134 (1976)
10. Kohonen, T.: Self-Organized Formation of Topologically Correct feature Maps. Biol. Cyber. 43, pp. 59--69 (1982)
11. Hopfield J. J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proc. Natl. Acad. Sci. USA, 79 (1982)
12. Rumelhart, D. E., Hinton, G. E., Williams, R. J.: Learning internal representations by error propagation. in: Parallel distributed processing: explorations in the microstructure of cognition. vol. 1, pp.318--362, MIT Press, Cambridge, MA, USA (1986)
13. Carpenter, G.A., Grossberg, S.: A Massively Parallel Architecture For A Self-Organizing Neural Pattern Recognition Machine. Comp. Vision, Graph. and Image Proc. Vol. 37, pp. 54--115 (1987)
14. Schmidhuber, J.: Deep Learning in Neural Networks: An Overview. Neural Networks. 61, pp. 85--117 (2014)
15. Gnana Sheela, K., Deepa, S. N.: Review on Methods to Fix Number of Hidden Neurons in Neural Networks. Anna University, Regional Centre, Coimbatore 641047, India (2013)