

# Analytical Study of Neural Network Techniques: SOM, MLP and Classifier-A Survey

Sonali B. Wankhede

4<sup>th</sup> Sem ME (Computer Engineering) Thadomal Shahani Engineering College, Mumbai, India

---

**Abstract:** This paper focuses on the various neural network techniques including Multilayer perceptron (MLP) neural network, classifier and self-organizing maps (SOMs). Various aspects of neural network techniques are mentioned in this paper along with the advantages and drawbacks. The neural network models are trained with measured values of the field strength at arbitrary points. The back propagation training algorithm is used for the learning process in MLP. The mechanism of supervised and unsupervised learning are also specified.

**Keywords:** MLP, Self-organizing maps (SOM), Neural network, Classifier, Applications

---

## I. INTRODUCTION

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do. 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. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to be solved must be known and stated in small unambiguous instructions. These instructions are then converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency. The neural networks are commonly categorized in terms of their corresponding training algorithms: fixed-weights networks, unsupervised networks, and supervised networks. There is no learning required for the fixed-weight networks, so a learning mode is supervised or unsupervised. Supervised learning networks have been the mainstream of neural model development. The training data consist of many pairs of input/output training patterns. For an unsupervised learning rule, the training set consists of input training patterns only. Therefore, the network is trained without benefit of any teacher. The network learns to adapt based on the experiences collected through the previous training patterns.

## II. NEURAL NETWORK

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

### A. Multilayer Perceptron (MLP) network

The most common neural network model is the multilayer perceptron (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the

model can then be used to produce the output when desired output is unknown. A graphical representation of an MLP is as shown below:

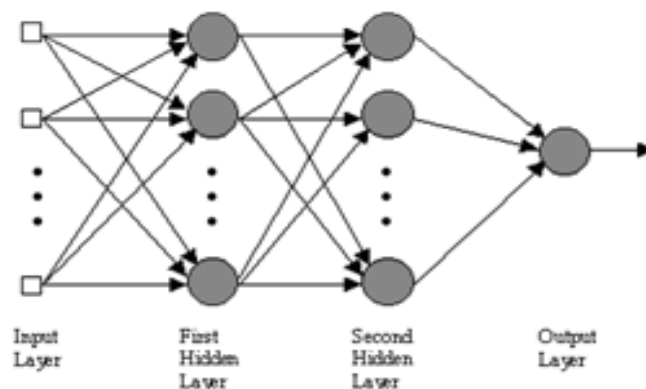


Fig 1. Architecture of an MLP neural network

This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as an activation function [12].

Multi-layer networks use a variety of learning techniques, the most popular being back-propagation. Here, the output values are compared with the correct answer to compute the value of some predefined error-function. By various techniques, the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles, the network will usually converge to some state where the error of the calculations is small. In this case, one would say that the network has *learned* a certain target function. To adjust weights properly, one applies a general method for nonlinear optimization that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated, and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function). For this reason, back-propagation can only be applied on networks with differentiable activation functions.

### B. Classifier

In the field of machine learning, the goal of statistical classification is to use an object's characteristics to identify which class (or group) it belongs to. A classifier is a system that performs a mapping from a feature space  $X$  to a set of labels  $Y$ . Basically what a classifier does is assign a pre-defined class label to a sample. For example, if you are building a spam classifier then the feature space contains a representation of an email and the label is either "Spam" or "Non-Spam".

#### Types of Models

**Decision-Region Boundaries:** This type of model defines decision regions by explicitly constructing boundaries in the input space. These models attempt to minimize the number of expected misclassifications by placing boundaries appropriately in the input space.

**Probability Density Functions:** The models of this type attempt to construct a *probability density function*,  $p(x|C)$ , that maps a point  $x$  in the input space to class  $C$ . *Prior probabilities*,  $p(C)$ , is to be estimated from the given database. This model assigns the most probable class to an input vector  $x$  by selecting the class *maximizing*  $p(C)p(x|C)$ .

**Posterior Probabilities:** Let there be  $m$  possible classes denoted  $C_1, C_2, \dots, C_m$ . This type of models attempts to generate  $m$  posterior probabilities  $p(C_i|x)$ ,  $i=1, 2, \dots, m$  for any input vector  $x$ . The classification is made in the way that the input vector is assigned to the class associated with maximal  $p(C_i|x)$

Commonly Used Algorithms

Parametric	Nonparametric
Linear regression	Backpropagation
Logistic regression	Radial basis function
Unimodal Gaussian	K nearest neighbor
	Gaussian mixture
	Nearest clustering
	Binary/Linear decision tree
	Projection pursuit
	Estimate-Maximize clustering
	Multivariate Adaptive Regression Spline (MARS)
	Group Method of Data Handling (GMDH)

Training Time

Algorithm	Training Time
Linear / Logistic regression	Fast-Medium
Backpropagation	Slow
Radial basis function	Medium
K nearest neighbor	No training required
Gaussian mixture	Medium-Slow
Nearest clustering	Medium
Estimate-Maximize clustering	Medium
GMDH	Fast-Medium
LVQ	Slow

Classification Time

Algorithm	Classification time
Linear / Logistic regression	Very fast
Backpropagation	Very fast
Radial basis function	Medium
K nearest neighbor	Slow
Gaussian mixture	Medium
Nearest clustering	Fast-medium
Estimate-Maximize clustering	Medium
GMDH	Fast
LVQ	Medium

Algorithm	Memory Usage
Linear / Logistic regression	Very low
Backpropagation	Low
Radial basis function	Medium
K nearest neighbor	High
Gaussian mixture	Medium
Nearest clustering	Medium
Estimate-Maximize clustering	Medium
GMDH	Low
LVQ	Medium

Memory Usage

### C. Self-organizing maps

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map[17]. Self-organizing maps are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space. This makes SOMs useful for visualizing low-dimensional views of high-dimensional data. Like most artificial neural networks, SOMs operate in two modes: training and mapping. "Training" builds the map using input examples (a competitive process, also called vector quantization), while "mapping" automatically classifies a new input vector. A self-organizing map consists of components called nodes or neurons. Associated with each node is a weight vector of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a two-dimensional regular spacing in a hexagonal or rectangular grid.

The self-organizing map describes a mapping from a higher dimensional input space to a lower dimensional mapspace. The procedure for placing a vector from data space onto the map is to find the node with the closest (smallest distance metric) weight vector to the data space vector. It has been shown that while self-organizing maps with a small number of nodes behave in a way that is similar to K-means, larger self-organizing maps rearrange data in a way that is fundamentally topological in character. It is also common to use the U-Matrix. The U-Matrix value of a particular node is the average distance between the node and its closest neighbors. In a square grid, for instance, we might consider the closest 4 or 8 nodes or six nodes in a hexagonal grid.

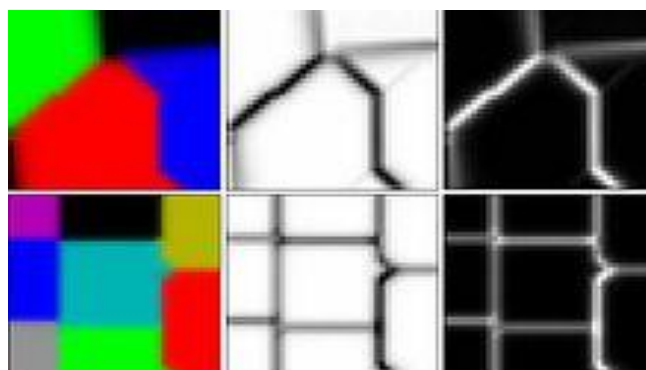


Fig 2. Self organizing maps (SOM) of three and eight colors with U-Matrix.

Figure 3 shows an example of a SOM trained to recognize the eight different colors shown on the right. The colors have been presented to the network as 3D vectors - one dimension for each of the color components - and the network has learnt to represent them in the 2D space you can see. Notice that in addition to clustering the colors into distinct regions, regions of similar properties are usually found adjacent to each other.

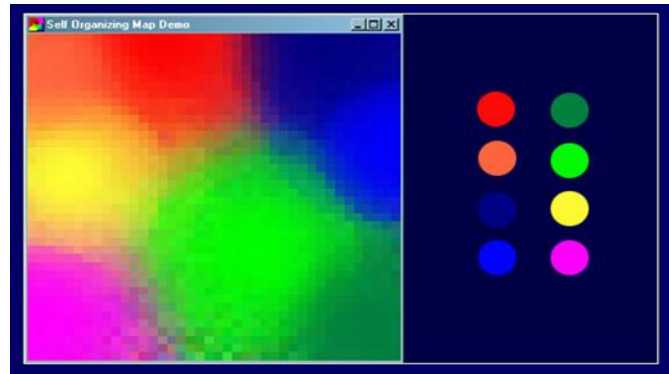


Fig 3. Screenshots of the demo program (left) and the color it has described (right)

One of the most interesting aspects of SOMs is that they learn to classify data *without supervision*. You may already be aware of supervised training techniques such as backpropagation where the training data consists of vector pairs - an input vector and a target vector. With this approach an input vector is presented to the network (typically a multilayer feedforward network) and the output is compared with the target vector. If they differ, the weights of the network are altered slightly to reduce the error in the output. This is repeated many times and with many sets of vector pairs until the network gives the desired output. Training a SOM however, requires no target vector.

**Learning algorithm:**

A SOM does not need a target output to be specified unlike many other types of network. Instead, where the node weights match the input vector, that area of the lattice is selectively optimized to more closely resemble the data for the class the input vector is a member of. From an initial distribution of random weights, and over many iterations, the SOM eventually settles into a map of stable zones. Each zone is effectively a feature classifier, so you can think of the graphical output as a type of feature map of the input space. If you take another look at the trained network shown in figure 3, the blocks of similar color represent the individual zones. Any new, previously unseen input vectors presented to the network will stimulate nodes in the zone with similar weight vectors.

Training occurs in several steps and over many iterations:

1. Each node's weights are initialized.
2. A vector is chosen at random from the set of training data and presented to the lattice.
3. Every node is examined to calculate which one's weights are most like the input vector. The winning node is commonly known as the Best Matching Unit (BMU).
4. The radius of the neighborhood of the BMU is now calculated. This is a value that starts large, typically set to the 'radius' of the lattice, but diminishes each time-step. Any nodes found within this radius are deemed to be inside the BMU's neighborhood.
5. Each neighboring node's (the nodes found in step 4) weights are adjusted to make them more like the input vector. The closer a node is to the BMU, the more its weights get altered.
6. Repeat step 2 for N iterations.

Winner take-all learning rule :

$$\Delta \bar{w}_k^q(t) = + \eta(t) \Lambda(k, i^q, t) (\bar{x}^q - \bar{w}_k^q(t)) \text{ for all units } k$$

$$\text{Neighborhood function } : \Lambda(k, i^q, t) = \exp\left(\frac{\|k - r_{i^q}\|^2}{2\sigma(t)^2}\right)$$

$$\text{Quantization Error } Q = \frac{1}{m} \sum_{q=1}^m \|\bar{x}^q - \bar{w}_{i^q}^q\|^2$$

$$\text{Average Distortion } D = \frac{1}{m} \sum_{q=1}^m \Lambda(k, i^q, t) \|\bar{x}^q - \bar{w}_{i^q}^q\|^2$$

**III. ADVANTAGES AND DISADVANTAGES OF MLP AND SOMS**

*Benefits of Multilayer Perceptron*

1. Connectionist: used as a metaphor for biological neural networks
2. Computationally efficient: Can easily be parallelized
3. Universal computing machines

Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. One of the preferred techniques for gesture recognition.
3. They yield the required decision function directly via training.
4. A two layer backpropagation network with sufficient hidden nodes has been proven to be a universal approximator.

#### *Drawbacks*

1. Convergence can be slow
2. Local minima can affect the training process
3. Hard to scale

#### *Advantages of SOM*

1. The main advantage of using a SOM is that the data is easily interpreted and understood.
2. SOMs are capable of handling several types of classification problems while providing a useful, interactive, and intelligible summary of the data. For example, a SOM could be used in place of a Bayesian spam filter. Using a weight vector similar to the one used in the WEBSOM project, a SOM should be able to map e-mails onto a grid with clusters representing spam and not spam. Training would occur whenever the user marked an e-mail as spam or not spam. The user would be presented with a graphical map of e-mail clusters, allowing him/her to graphically view where an e-mail fell compared to other e-mails they had previously marked as spam / not spam.
3. A SOM can be trained in a short amount of time. Furthermore, the algorithm is simple enough to make the training process easy to understand and alter as needed.

#### *Disadvantages*

1. The major disadvantage of a SOM is that it requires necessary and sufficient data in order to develop meaningful clusters. The weight vectors must be based on data that can successfully group and distinguish inputs. Lack of data or extraneous data in the weight vectors will add randomness to the groupings. Finding the correct data involves determining which factors are relevant and can be a difficult or even impossible task in several problems. The ability to determine a good data set is a deciding factor in determining whether to use a SOM or not.
2. Another problem with SOMs is that it is often difficult to obtain a perfect mapping where groupings are unique within the map. Clusters will often get divided into smaller clusters, creating several areas of similar neurons. This can be prevented by initializing the map well.
3. Finally, SOMs require that nearby data points behave similarly. Parity-like problems such as the XOR gate do not have this property and would not converge to a stable mapping in a SOM.

## **IV. APPLICATIONS OF SOM, MLP AND CLASSIFIERS**

### **a) Aerospace**

High performance aircraft autopilots, flight path simulations, aircraft control systems, autopilot enhancements, aircraft component simulations, aircraft component fault detectors.

### **b) Automotive**

Automobile automatic guidance systems, warranty activity analyzers.

### **c) Banking**

Check and other document readers, credit application evaluators.

### **d) Defense**

Weapon steering, target tracking, object discrimination, facial recognition, new kinds of sensors, sonar, radar and image signal processing including data compression, feature extraction and noise suppression, signal/image identification.

### **e) Electronics**

Code sequence prediction, integrated circuit chip layout, process control, chip failure analysis, machine vision, voice synthesis, nonlinear modeling.



**f) Financial**

Real estate appraisal, loan advisor, mortgage screening, corporate bond rating, credit line use analysis, portfolio trading program, corporate financial analysis, currency price prediction.

**g) Manufacturing**

Manufacturing process control, product design and analysis, process and machine diagnosis, real-time particle identification, visual quality inspection systems, welding quality analysis, paper quality prediction, computer chip quality analysis, chemical product design analysis, machine maintenance analysis, project bidding, dynamic modeling of chemical process systems.

**h) Medical**

Breast cancer cell analysis, EEG and ECG analysis, prosthesis design, optimization of transplant, hospital quality improvement, emergency room test advisement.

**i) Robotics**

Trajectory control, forklift robot, manipulator controllers, vision systems.

**j) Speech**

Speech recognition, speech compression, vowel classification and text to speech synthesis.

**k) Securities**

Market analysis, automatic bond rating, stock trading advisory systems.

## V. CONCLUSION

In this paper a detailed study on the Multilayer perceptron neural network (MLP), self-organizing maps (SOMs) and classifiers are presented. The advantages and drawbacks related to these computing models are highlighted. The applications in various fields are briefly discussed. MLP constructs global approximations to nonlinear input and output mapping. Consequently they are capable of generalizing in those regions of input space where little or no training data is available. On the other side SOMs follow unsupervised training which is based on competitive learning, in which the output neurons compete amongst themselves to be activated. This activated neuron is called a winner-takes-all neuron or simply winner neuron. Such competition can be induced by having lateral inhibition connections (negative feedback paths). Hence there is a scope for further research in these areas.

## REFERENCES

- [1] J Harun, N.; Woo, W.L. ; Dlay, S.S. ;” Performance of keystroke biometrics Authentication system using Artificial Neural Network (ANN) and Distance Classifier Method " International Conference on Computer and Communication Engineering (ICCCCE), 2010
- [2] Schelar, A.; Rokach, L.; Abramson, A.; Elovici Y, "User Authentication Based on Representative Users" IEEE Transactions on systems, man and cybernetics part C, Vol 42, NO. 6, pp.1669-1678,( NOVEMBER 2012)
- [3] BioPassword: History and Science of keystroke dynamics. <http://www.biopassword.com/resources/>
- [4] D.Shanmugapriya and G.Padmavathi, "An Efficient Feature Selection Technique for User Authentication using Keystroke Dynamics," IJCSNS, vol. 11, no.10, 2011.
- [5] L. K. Maisuria , C. S. Ong and W. K. Lai, " A comparison of artificial neural network and cluster analysis for typing biometrics authentication", International Joint Conference on Neural Network, IJCNN'99, vol.5, pp 3295-3299, (1999).
- [6] S. Modi and S. J. Elliott, "Keystroke dynamics verification using a spontaneously generated password password," in *Proc. 40th Annu. Int. Carnahan Conf. Security Technol.*, Lexington, KY, Oct. 2006, pp. 116–121.
- [7] A. Peacock, X. Ke, and M. Wilkerson, "Typing patterns: A key to user identification," *IEEE Security Privacy*, vol. 2, no. 5, pp. 40–47, Sep./Oct. 2004.
- [8] N. Capuano, M. Marsella, S. Miranda and S. Salerno, "User Authentication with Neural networks", University of Salerno Italy. [http://www.capuano.biz/Papers/EANN\\_99.pdf](http://www.capuano.biz/Papers/EANN_99.pdf)
- [9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*, 2<sup>nd</sup> edition, John Wiley & Sons, Inc., 2001
- [10] S R Shorrock, D J Atkinson, S S Dlay, Biometric verification of computer users with probabilistic and cascade forward neural networks.
- [11] Ross, A., Shah, J., & Jain, A. K. March 2005. Towards reconstructing fingerprints from minutiae points. *Proceedings of SPIE Conference on Biometric Technology for Human Identification II*, 5779, 68–80.
- [12] M.S. Obaidat and D.T Macchiarolo, "A multilayer neural network for computer access security", IEEE transactions on Systems, Machine and Cybernetics, vol 24(5), (1994).
- [13] S. Bleha and M. obaidat. Computer users verification using the perceptron algorithm. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3):900 -902, May. 1993
- [14] D. T. Lin, "Computer-access authentication with neural network based keystroke identity verification," in *Proc. Int. Conf. Neural Networks*, vol.1, 1997, pp. 174–178.
- [15] A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: Personal Identification in Networked Society*. Norwell, MA: Kluwer, 1999.
- [16] Kohonen T. Analysis of A Simple Self-Organizing Process[M]. *Biol. cybern.* 1982(44):135-140.
- [17] Flanagan J A. Self-organisation in Kohonen's SOM[J]. *Neural Networks*, 1996, 9(7):1185-1197.