International Journal of Information and Computation Technology
ISSN 0974-2239 Volume 2, Number 3 (2012), pp. 189–195

http://www.ripublication.com/

ARTIFICIAL NEURAL NETWORK FOR PATTERN REORGANIZATION

[1.] Dara Eshwar, [2.] Desh Pal Singh

[1.] Research Scholar , CMJ University ,Shillong , Meghalaya, India
[2.] Professor and Director , MIT, Ghaziabad, India

Abstract

Brain organizes this huge number of neurons (also referred as cells or units), each with weak computing power, into a massively parallel complex network, which these neurons interact with each other dynamically to produce a powerful information processor. Neurons are five to six orders of magnitude slower than current silicon gates. The modern computer easily outperforms the human in pre-programmable, repetitive computations. However, speech understanding and visual perception are still beyond the reach of serial digital computers even after allowing for an increase of speed by several orders of magnitude. Many of the tasks are difficult for the digital computers either because of computational load requires speed and storage not realizable with existing technology or because of possible inherent intractability of some problems including their complete and accurate symbolic descriptions. In comparison to digital computers, human beings as well as many other living creatures tackle the practical problems without much effort. These human capabilities motivate Artificial Neural Network (ANN) research.

1 .INTRODUCTION

Neural networks can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. Neural network models attempt to use some organizational principles (such as learning, generalization, adaptivity, fault tolerance and distributed representation, and computation) in a network of weighted directed graphs in which the nodes are artificial neurons and directed edges (with weights) are connections between neuron outputs and neuron inputs. The main characteristics of neural networks are that they have the ability to learn complex nonlinear input-output relationships, use sequential training procedures, and adapt themselves to the data. The most commonly used family of neural networks for pattern classification tasks [13] is the feed-forward network, which includes multilayer perception and Radial-Basis Function (RBF) networks. These networks are organized into layers and have unidirectional connections between the layers. Another popular network is the Self-Organizing Map (SOM), or Kohonen-Network [14], which is mainly used for data clustering and feature mapping. The learning process involves updating network architecture and connection weights so that a network can efficiently perform a specific classification/clustering task. The increasing popularity of neural network models to solve pattern recognition problems has been primarily due to their seemingly low dependence on domain-specific knowledge (relative to model-based and rule-based approaches) and due to the availability of efficient learning algorithms for practitioners to use.

2 HUMAN BRAIN AS NEURAL NETWORK

Human brain, made up of a vast network of computing elements called *neurons*, is coupled with sensory receptors (affecters) and effectors. A neuron is a special cell that conducts an electrical signal. There are about 10 billion neurons in the human brain. Neurons interact through contacts called synapses. Each synapse spans a gap about a millionth of an inch wide. On an average each neuron receives signals via thousands of synapses.

Brain organizes this huge number of neurons (also referred as cells or units), each with weak computing power, into a massively parallel complex network, which these neurons interact with each other dynamically to produce a powerful information processor. Neurons are five to six orders of magnitude slower than current silicon gates. The modern computer easily outperforms the human in pre-programmable, repetitive computations. However, speech understanding and visual perception are still beyond the reach of serial digital computers even after allowing for an increase of speed by several orders of magnitude. Many of the tasks are difficult for the digital computers either because of computational load requires speed and storage not realizable with existing technology or because of possible inherent intractability of some problems including their complete and accurate symbolic descriptions. In comparison to digital computers, human beings as well as many other living creatures tackle the practical problems without much effort. These human capabilities motivate Artificial Neural Network (ANN) research. Neurobiologists want to understand the stimulus-response characteristics of a single neuron and the interconnections of neurons that form either sub-regions of the brain or smaller subdivisions of the nervous system. Such lower-level models help us to understand the properties of neurons that are important for higher-level functions. Psychologists attempt to understand the brain functions from the behavioral and cognitive levels.

2.3 HUMAN BRAIN AND NEURAL NETWORK: A COMPARISON

Many studies suggest that humans may use less than 10 percent of their brains' potential power. While this anecdotal evidence has not been scientifically proven, it is one of the many mysteries of the human brain. Some scientists state that human memory cells are located in certain areas of the brain. Others state that memory is distributed throughout the brain and there is no specific memory location. Of course, nothing is clear. This article compares the similarities between human and neural networks. Our interest in this topic stems from our research on using neural networks to recognize fingerprints.

Comparison

Now the question remains, what is the difference between human and neural networks? Both can learn and become expert in an area and both are mortal. *The main difference is, humans can forget but neural networks cannot.* Once fully trained, a neural net will not forget. Whatever a neural network learns is hard-coded and becomes permanent. A human's knowledge is volatile and may not become permanent. There are several factors that cause our brain cells to die and if they do, the information that is stored in that part is lost and we start to forget.

The other difference is accuracy. Once a particular application or process is automated through a neural network, the results are repeatable and accurate. Whether the process is replicated one thousand times or one million times, the results will be the same and will be as accurate as calculated the first time. Human beings are not like that. The first 10 processes may be accurate, but later we may start to make mistakes in the process. Another key difference is speed. Neural networks can be hardware or software. It is obvious that neural networks are much faster than humans in processing data and information.

2.4 ARTIFICIAL NEURAL NETWORK

We consider an ANN as a highly simplified model of the structure of the biological neural network. An ANN consists of the following:

a. ***Processing Unit:*** This is the summing part of which receives n input values, weighs each value, and performs a weighted sum. The weighted sum is called *activation value*. The sign of weight may be negative (*inhibitory* input) or positive (*excitatory* input). The inputs could be discrete or continuous data values, and likewise the outputs also could be discrete or continuous.

b. ***Interconnections:*** In an artificial neural network several processing units are interconnected according to some topology to accomplish a pattern recognition task. Therefore the inputs to a processing unit may come from outputs of the other processing units, and/or from an external source. The output of each unit may be given to several units including it. The amount of the output of one unit received by another unit depends on the strength of the connection between the units, and it is reflected in the *weight* value associated with the connecting link. If there are N units in a given ANN then at any instant of time each unit will have a unique activation value and a unique output value. The set of the N activation value defines the *activation state* of the network at that instant. Likewise, the set of the N output values of the network defines the *output state* of the network at that instant.

c. Operations: In operation, each unit of an ANN receives inputs from other connected units and /or from an external source. A weighted sum of the inputs is computed at a given instant of time. The resulting activation value determines the actual output from the output function unit i.e. the output state of the unit. The activation values of the units (activation state) of the network as a function of time are referred to as *activation dynamics*. The activation dynamics also determine the dynamics of the output state of the network. For a given network, defined by the units and their interconnections with appropriate weights, the activation state refers to the *short term memory* function of the network. Generally the activation dynamics is followed to *recall* a pattern stored in a network.

d. Update: In implementation, there are several options available for both activation and synaptic dynamics. In particular, the updating of the output states of all units could be performed *synchronously*. In this case, the activation values of all units are computed at the same time assuming a given output state throughout. From these activation values the new output state of the network is derived. In an *asynchronous* update, on the other hand, each unit is updated sequentially, taking the current output state of the network into account each time. For each unit, the output state can be determined from the activation value either deterministically or *stochastically*.

2.4 ARTIFICIAL NEURAL NETWORK TOPOLOGIES

Artificial neural networks are useful only when the processing units are organized in a suitable manner to accomplish a given pattern recognition task. The arrangement of the processing units, connections, and pattern input/output is referred to as *topology*. In ANN these processing units are normally organized into layers. Connections are either *interlayer* (i.e. from units of one layer to units of another) or *intralayer* (i.e. from the units within the layer) or both. Further, the connections among the layers and among the units within a layer can be made either in *feedforward* manner or in *feedback* manner. In a feedback network the same processing unit may be visited more than once.

Assuming two layers F_1 and F_2 with N and M processing units respectively, let us define few basic topologies. By providing connections to the j^{th} unit in F_2 from all the units in F1, as shown in **figure 2.4a** and **2.4b**, we get two network structures- *instar* and *outstar*. The units in F_1 layer are linear, so that for each unit I in this layer the input (a_i) = activation (x_i) = output signal (s). In instar, during learning, the weight vector $\mathbf{w_j}$ (w_{j1}, $w_{j2} \ldots w_{jN}$) is adjusted so as to approach the given input vector **a** at F_1 layer. Therefore whenever the input is given to F_1, then the jth unit of F_2 will be activated to the maximum extent. Thus the operation of instar can be viewed as *content addressing the memory*. In the case of outstar, during learning, the weight vector for the connections from the jth unit in F_2 approaches the activity pattern in F_1 when input vector **a** is present at F_1. During recall, whenever the unit j is activated, the signal pattern ($s_j w_{1j}$, sjw_{2j},…..$s_j w_{Nj}$) will be transmitted to F_1, which then produces the original activity pattern corresponding to the input vector **a**, although the point is absent. Thus the operation of outstar can be viewed as *memory addressing the contents*.

When all the connections from units in F_1 and F_2 are made as in figure 2.4c and d, then we obtain a heteroassociation network. This network can be viewed either as group of instars (**figure 2.4c**) or group of outstars (**figure 2.4d**).

When the flow is bi-directional and the weight are symmetric (w_{ij} = w_{ji}), then we get a bi-directional associative memory (**fig. 2.4e**), where either of the layers can be used as input/output.

If the two layers coincide, then we obtain an autoasociative memory in which each unit is connected to every other unit and to itself (**figure 2.4f**).

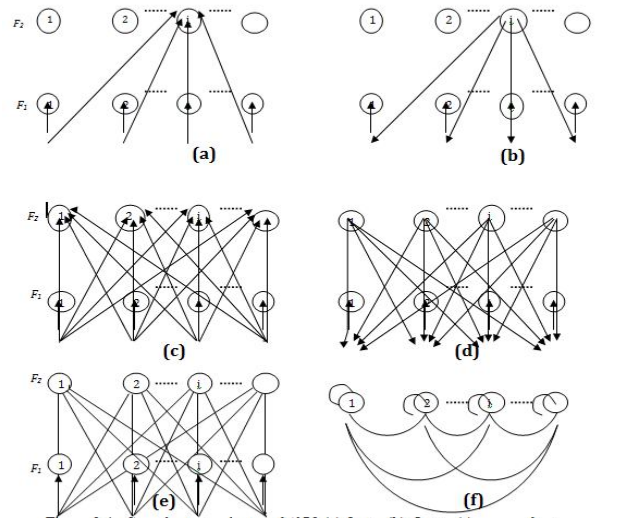

Figure 2.4: *Some basic topologies of ANN.* **(a)** *Instar* **(b)** *Outstar* **(c)** *group of instars* **(d)** *group of outstars* **(e)** *bi-directional associative memory* **(f)** *auto associative memory*

2.5 NEURONAL DYNAMICS

Artificial Neural Networks can be considered as trainable nonlinear dynamical systems. For a network consisting of N processing units, the activation state of the network at any given instant corresponds to a point in the N dimensional state space. The dynamics of the neural network, after tracing a trajectory in the state space, ends at an *equilibrium state* of the system in the normal course. An equilibrium state is one at which small perturbations around it due to neuronal dynamics will not perturb the state.

Neuronal dynamics consists of two parts: one corresponding to the dynamics of activation states and the other corresponding to the dynamics of synaptic weights. The *activation dynamics* determines the time evolution of the neuronal activations, and it is described by a system of first order differential equations which are first derivatives of the activation state i.e. dx_i/dt. Likewise, *synaptic dynamics* determines the changes in the synaptic weights. The equations governing the dynamics are described in terms of the first derivatives of the synaptic weights, i.e. dw_{ij}/dt, where w_{ij} is the strength of the connecting link from the jth unit to the ith unit.

Synaptic weights change gradually, whereas the neuronal activations fluctuate rapidly. Therefore, while computing the activation dynamics, the synaptic weights are assumed to be constant. The synaptic dynamics dictates the *learning process*. The *short-term memory* (STM) in neural networks is modeled by the activation state of the network. The *long-term memory* (LTM) corresponds to the encoded pattern information in the synaptic weights due to learning.

2.6 LEARNING LAWS

Synaptic dynamics, as discussed earlier, is described in terms of expressions for the first derivative of the weights. They are called learning equations. Typical learning involves adjustment of the weight vector such that

$$\Delta \mathbf{w}_i(t) = \eta g[\mathbf{w}_i(t), \mathbf{a}(t), b_i(t)]\mathbf{a}(t) \quad (2.11)$$

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \Delta \mathbf{w}_i(t) \quad (2.12)$$

where η = learning rate parameter,

$\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots w_{iN}]^T$ weight vector with components w_{ij},

w_{ij}= weight connecting the jth input unit to the ith processing unit,

$\mathbf{a}$ = input vector with components a_i, i = 1, 2, …N,

$\mathbf{b}$= desired output vector with components b_i, i = 1, 2, …M.

Input units are assumed linear. Hence $\mathbf{a} = \mathbf{x}$ (unit activation) = $\mathbf{s}$ (unit output)

Output units are in general non-linear. Hence $s_i = f(\mathbf{w}_i^T \mathbf{a})$.

The function *g* may be viewed as a learning function that depends on the type of learning adopted. There are different methods for implementing the synaptic dynamics. These methods are called *learning laws*.

Supervised Vs Unsupervised learning

There are several learning laws in use, and new are being developed to suit a given application and architecture. There are some general categories that these laws fall into. Most common is out of those is supervised or unsupervised. In supervised learning the weight changes are determined by the difference between the desired output and the actual output. Some of the supervised learning laws are: error correction learning or delta rule, stochastic learning, and hardwired learning [29]. Unsupervised learning discovers features in a given set of patterns and organizes the patterns accordingly. There is no externally specified desired output as in the case of supervised learning. Examples of this category include Hebbian learning, differential Hebbian learning, principle component learning and competitive learning [26]. Unsupervised learning uses mostly local information to update the weights.

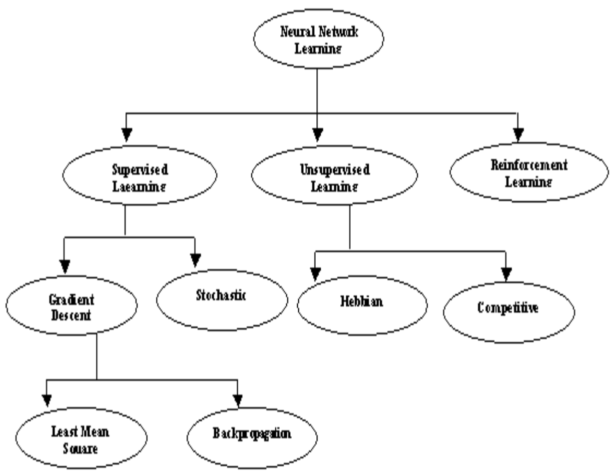

Figure 2.5: *Classification of Learning algorithms*

Some of common discrete learning laws are discussed as follows:

2.6.1 Hebb's law:

Here the change in the weight vector is given by

$\Delta \mathbf{w}_i = \eta f(\mathbf{w}_i^T \mathbf{a})\ \mathbf{a}$ (2.13)

Therefore, the jth component of $\Delta \mathbf{w}_i$ is given by

$\Delta w_{ij} = \eta f(\mathbf{w}_i^T \mathbf{a})\ a_j = \eta s_i a_j$,

for $j = 1,2,\ldots,M$ (2.14)

where s_i is the output signal of ith unit. This law requires weight initialization $\mathbf{w}_i \approx \mathbf{0}$ prior to learning and represent unsupervised learning.

2.6.2 Perceptron learning law [13] :

Here the change in the weight vector is given by

$\Delta \mathbf{w}_i = \eta[b_i - \mathrm{sgn}(\mathbf{w}_i^T \mathbf{a})]\mathbf{a}$ (2.15)

Wheresgn(x) is sign of x. Therefore, we have

$\Delta w_{ij} = \eta[b_i - \mathrm{sgn}(\mathbf{w}_i^T \mathbf{a})]\ a_j = \eta(b_i - s_i)a_j$ for $j = 1,2,\ldots,M$ (2.16)

This rule is applicable for bipolar output functions. The weights can be initialized to any random value prior to learning. This law is a supervised learning law because it requires a desired output for each input.

2.6.3 Delta learning law:

Here the change in the weight vector is given by

$\Delta \mathbf{w}_i = \eta\ [b_i - f(\mathbf{w}_i^T \mathbf{a})]\ f(\mathbf{w}_i^T \mathbf{a})\ \mathbf{a}$ (2.17)

where $f(x)$ is the derivative with respect to x. Hence,

$\Delta w_{ij} = \eta\ [b_i - f(\mathbf{w}_i^T \mathbf{a})]\ f(\mathbf{w}_i^T \mathbf{a})\ a_j = \eta[bi - s_i]\ f(x_i) a_j$, for $j = 1,2,\ldots,M$ (2.18)

This law is valid only for a differentiable output function, as it depends on the derivative of the output function. It is a supervised learning law since the change in the weight is based on the error between the desired and the actual output values for the given input. The weights can be initialized to any random value in the beginning.

2.6.4 Widrow- Hoff LMS learning law :

Here the change in the weight vector is given by

$\Delta \mathbf{w}_i = \eta\ [b_i - \mathbf{w}_i^T \mathbf{a}]\ \mathbf{a}$ (2.19)

Hence

$\Delta w_{ij} = \eta\ [b_i - \mathbf{w}_i^T \mathbf{a}]\ a_j$,

for $j = 1,2,\ldots,M$ (2.20)

This is a supervised learning law and is a special case of the delta learning law, where the output function is assumed linear. The weights may be initialized to any values.

2.6.5 Correlation learning law:

Here the change in the weight vector is given by

$\Delta \mathbf{w}_i = \eta b_i \mathbf{a}$ (2.21)

Therefore,

$\Delta w_{ij} = \eta b_i a_j$, for $j = 1,2,\ldots,M$ (2.22)

This is the special case of the Hebbian learning with the output signal (s_i) being replaced by the desired signal (b_i). But the Hebbian learning is an unsupervised learning, whereas the correlation learning is a supervised learning, since it uses the desired output value to adjust the weights. The weights are initialized to zero prior to learning.

2.6.6 Instar (winner-take-all) learning law:

This is relevant for a collection of neurons organized in an instar topology.
Here,
$\Delta w_{mj} = \eta(a_j - w_{mj})$,
for $j = 1,2\ldots N$ (2.23)

where $\mathbf{w}_m^T\mathbf{a} = \max(\mathbf{w}_i^T\mathbf{a})$. Here the weights are initialized to random values prior to learning and their lengths are normalized during learning.

2.7.7 Outstar learning law:

This is relevant for a collection of neurons organized in an outstar topology.
Here,
$\Delta w_{kj} = \eta(b_k - w_{kj})$,
for $k = 1, 2\ldots K$ (2.24)
Where **b** is the desired response from the layer of K neurons. The weights are initialized to zero before learning.

2.8 TYPES OF ARTIFICIAL NEURAL NETWORKS

There are three types artificial neural networks. They are:

- Feed-forward
- Feedback
- Combination of both

The simplest networks of each of these types form the basic functional units. They are functional because they can perform by themselves some simple pattern recognition tasks. They are basic because they form building blocks for developing neural network architectures for complex pattern recognition tasks.

In multi-layer feed forward networks, the processing elements are arranged in layers and only the elements in adjacent layers are connected with each other. It has a minimum of three layers of elements (i) the input layer (ii) the middle or hidden layer and (iii) the output layer. The information propagation is only in the forward direction and there no feedback loops. Even it does not have feedback connections, errors are propagated during training.
If the connection among the layer is such that the layers may be visited more than once, the network is called feedback networks.

3. CONCLUSION

The present chapter carried us through the fundamentals of neural networks. These two approaches have been emerged following different paths but both are inspired by natural world. Substantial work is in progress to combine these two to evolve hybrid systems, which can be applied to solve different problems in more efficient manner. Since we are applying this approach to solve pattern recognition task (storage and recalling of patterns in specific).

4. REFERENCES

[1] Simpson P K, “Foundations of Neural Networks”, *Artificial Neural Networks: Paradigms, Applications and Hardware Implementations* (E. Sanchez-Sinencio and C. Lau, eds.), New York: IEEE Press, pp. 3-24, 1992

[2] Moller M F, “A scale conjugate gradient algorithm for supervised learning”, *Neural Networks*, vol 6, number 4, pp 525-533, 1993

[3] Knerr S, Personnaz L, and Dreyfus G, “Handwritten digit recognition by neural networks with single-layer training”, *IEEE Transactions on Neural Networks*, vol.3, pp. 962-968, November 1992

[4] Whitley D, Starkweather T, and Bogart C, “Genetic algorithms and neural networks: Optimizing connections and connectivity”, *Parallel Computing*, vol. 14, number 3, pp. 347-361, 1990

[5] Mangal, M and Singh M P, “Analysis of Multidimensional X-OR Classification Problem with Evolutionary Feed-forward Neural Networks”. *“International Journal on Artificial Intelligence Tools”*, vol. 16, No.1, pp-111-120, World Scientific, 2007

[6] Mangal M and Singh, M P, “Analysis of Classification for the Multidimensional Parity-Bit-Checking Problem with Hybrid Evolutionary Feed-forward Neural Network.” *Neurocomputing*, vol. 70, pp-1511-1524, 2007

[7] Breiman L and Friedman J H, “Estimating optimal transformations for multiple regression and correlation”, *Journal of the*

American Statistical Association, vol. 80, pp. 580-619, 1985

[8] Wright A H, “Genetic Algorithms for Real Parameter Optimization”, *Foundations of Genetic Algorithms*, Rawlins G J E, Eds., CA: Morgan Kaufmann Publishers, pp. 205-218, 1991

[9] Rudolph, G, “Convergence Properties of Canonical Genetic Algorithms”, *IEEE Tans. Neural Networks*, vol. 5, number 1, 1994

[10] Davis L., “Mapping Classifier Systems into Neural Networks”, *Proceedings of the 1988 Conference on Neural Information Processing Systems*, Morgan Kaufmann, 1988

b. Branke J, “Evolutionary algorithms for neural network design and training”, *First Nordic Workshop on Genetic Algorithms and its Applications*, Vaasa, Finland, 1995

[11] Kosko, B., “Neural Networks for Computing”, *Proc. American Inst. Of Physics*, pp. 277-282, April 1986

[12] Desai M S, *Noisy pattern retrieval using associative memories*, MSEE thesis, University of Louisville, Kentucky, 1990

[13] McCulloch W. S. and Pitts W., “A logical calculus of the ideas immanent in nervous activity”, *Bull. Math. Biophy.*, vol. 5, pp. 115-133, 1943

[14] Zurada J.M., *Introduction to Artificial Neural Systems*, Singapore: Info Access and Distribution, 1992

[15] Rosenblatt F, “A Probalistic model for information storage and organization in the brain”, *Psychol. Rev.* vol 65, pp 386-408, 1958