Chapter 1

Introduction

Artificial Neural Networks are abstract representations of brain information processing. The hope to reproduce at least some of the flexibility and power of human brain by artificial means has lead to the subject of study known as Neural Networks, Neural Computation or Brain like computation [Anderson 92].

The recent resurgence of interest in neural networks has its roots in the recognition that the brain performs computations in a different manner than do conventional computers. Conventional computers are extremely fast and precise at executing sequence of instructions that have been precisely formulated for them. But human brain is more efficient than conventional computers at computationally complex tasks such as vision, speech, information retrieval and pattern recognition, though human brain is composed of neurons which are million times slower than computer gates.

Unfortunately the understanding of biological neural systems is not developed sufficiently enough to address the functional similarities that may exist between the biological and man made neural systems. As a result, many major potential gains derived from such functional similarities, if they exist have yet to be exploited. It is not necessary that the architecture of brain is copied as it is to the extent it is understood. Implementation of the functionality of the brain in any manner is the guiding force in neurocomputing.

Neurocomputing, a functionally new and different approach for information processing,
is concerned with parallel, distributed and adaptive information processing systems that develop information processing capabilities in adaptive response to an information environment [Hecht-Nielsen 91]. In other words Artificial Neural Systems (ANS) are physical cellular systems which can acquire, store and utilize experiential knowledge.

1.1 Basic Concepts of ANN

The potential of Artificial Neural Networks (ANN) relies on massively parallel architecture composed of many simple computational elements connected by edges called weights. The basic computational element in an artificial neural network is called neuron and is also known as node or processing element. Brain researchers have identified over 100 different kinds of biological neurons. Processing elements also come in a variety of types. McCulloch and Pitts [McCulloch 43] proposed a binary threshold unit as computational model for a neuron. As a unit, the neural network can be represented with threshold and weight functions.

In an artificial neural network, inputs are fed to neurons through synapses (connection weights). Basically, the output of a neuron in a neural network is a weighted sum of its inputs, but a threshold function is also used to determine the final value or the output. For a detailed exposure for threshold functions, refer [Kosko 92].

The state of a neuron is nothing but the activity of a neuron at any time instance. This attribute may be discrete or continuous valued. An update rule or transition rule or activation rule is a rule for evaluating the state of one or more neurons under the existing conditions and changing them if necessary.

Topology of ANN is the pattern of connectivity in such a way that it can be viewed as weighted directed graph in which artificial neurons are nodes and directed edges (with
weights) are connections between neuron outputs and neuron inputs. Basing on the connection pattern (architecture), ANNs may be grouped into two categories: *feed-forward networks (non-recurrent)* are those in which graphs have no loops, while *feed back (recurrent)* networks are those in which loops occur because of feedback connections. Broadly speaking, feed forward networks are static, whereas recurrent or feedback networks are dynamic systems. When a new input pattern is presented, the neuron outputs are computed. Because of the feedback paths, the inputs to each neuron are then modified, which leads the network to enter a new state.

The state of all neurons in a feedback network at any instance of time is the *state of the neural network*. *Stable states* of a feedback network are the states of the network which do not change under usual disturbances in the states of the neurons of the network. *Learning* in a neural network is the process of making certain set of states of network as stable states. The function in which one can substitute the state values of the neurons that represents the energy of the network (at that instance) is called its *energy function*. During the updation of the network, the value of the energy function decreases and eventually reaches a minimum. In view of the above definition, this state of the network is referred to as *stable state* and this minimum is called *local minimum*. When this state is reached, the network is said to be stable.

Since this thesis addresses a new neural network architecture with a new learning rule, some of the learning rules known in the literature are recalled in the next section.

### 1.2 Learning Rules

The ability to learn is a fundamental trait of intelligence. ANNs' ability to automatically learn from examples makes them attractive and exciting. A learning process in the ANN
context can be viewed as the problem of updating network architecture and connection weights so that a network can efficiently perform a specific task [Jain 96].

A learning algorithm refers to a procedure in which learning rules are used for adjusting the weights. There are three main learning paradigms: supervised, unsupervised and hybrid.

In supervised learning, or learning with a "teacher", the network is provided with a correct answer (output) for every input pattern. Weights are determined to allow the network to produce answers as close as possible to the known correct answers.

In contrast, unsupervised learning, or learning without a teacher, does not require a correct answer associated with each input pattern in the training data set. It explores the underlying structure in the data, or correlates patterns in the data and organizes patterns into categories from these correlations.

Hybrid learning combines supervised and unsupervised learning.

There are four basic types of learning rules: error-correction, Boltzmann, competitive learning and Hebbian learning.

Error-correction learning [Rosenblatt 58]: In this learning paradigm, the network is given a desired output for each input pattern. During the learning process the actual output $y$ generated by the network may not be equal to the desired output $d$. The basic principle of error-correction learning rules is to use the error signal $(d - y)$ to modify the connection weights to gradually reduce this error.

Boltzmann learning [Anderson 88]: It is a stochastic learning rule derived from information theoretic and thermodynamic principles. The objective of Boltzmann learning is to adjust the connection weights so that the states of visible units satisfy a particular
desired probability distribution.

Competitive learning [Haykin 94]: In this method of learning output units compete among themselves for activation. As a result, only one output unit is active at any given time. This phenomenon is known as winner-take-all. Competitive learning is found to exist in biological neural networks.

Hebbian learning [Hebb 49]: The oldest learning rule is Hebb's postulate of learning. Hebb's rule is based on the following observation from neurobiological experiments: If neurons on both sides of a synapse are activated synchronously and repeatedly, the synapse's strength is selectively increased. Mathematically, the Hebbian rule can be described as

$$w_{ij}(T + 1) = w_{ij}(r) + ny_i(T) x_i(r)$$

(1.2.1)

where $x_i$ and $y_j$ are the output values of neurons i and j, respectively, which are connected by synapse $w_{ij}$, and $\eta$ is the learning rate. Note that $x_i$ is the input to the synapse. The important property of this rule is that learning is done locally, that is, the change in the synaptic weight depends only on the activities of the two neurons connected by it. This significantly simplifies the complexity of the learning circuit in a VLSI implementation.

Some of these learning rules are recalled as and when required in subsequent chapters.

1.3 Models of Neural Networks

Researchers from many scientific disciplines are designing artificial neural networks to solve a variety of problems in pattern recognition, prediction, optimization, associative memory and control. In this direction the premier work is due to McCulloch and Pitts, later known as Perception. Afterwards many models like Multilayer feedforward, Kohonen's self-organizing map, ART and Hopfield have played vital role in the development.
This section deals with different trendsetting models of neural networks.

Perceptron [McCulloch 43; Rosenblatt 58; Rumelhart 86]: Perceptron is the first precisely specified, computationally oriented neural network. In this network processing units may be of binary or of continuous type. Simple reinforcement kind of learning rules are used.

Multilayer perceptron [Rumelhart 86; Minsky 69]: The most popular class of multi-layer feed-forward networks is multilayer perceptrons with one or more layers of nodes between the input and the output nodes. The first layer is the input layer and the last the output layer. The layers that are placed between the first and the last layers are the hidden layers. Multilayer perceptrons can form arbitrarily complex decision boundaries and can represent any Boolean function. The development of the back-propagation learning algorithm [Rumelhart 86] for determining weights in a multilayer perceptron has made these networks the most popular among researchers and users of neural networks. The back-propagation algorithm enables one to determine the errors in the hidden layer outputs which are used as a basis for adjustment of connection weights between input and hidden layers.

Kohonen Self-organizing Map [Kohonen 84]: Many parts of the brain are organized in such a way that the aspects of the sensory environment can be represented in the form of two dimensional maps. Self-organizing network is an attempt to construct this aspect by artificial means. The synaptic weights to a neuron represent prototype vectors defining the classes or clusters. The fundamental fact that must hold true for a topographically organized system is that nearby neurons respond similarly. The essential mechanism of this model is to cause the neural network to modify itself to achieve this character. Kohonen's self-organizing map can be used for projection of multivariant data, density
approximation and clustering.

Adaptive Resonance Theory model (ART) [Carpenter 88]: ART is introduced to solve the stability-plasticity dilemma i.e., learning new things (plasticity) and yet retain the existing knowledge (stability). The networks designed by this approach are better suited to adapt to unexpected changes as biological neural networks are geared to do. Carpenter and Grossberg [Carpenter 91] contend that adaptive resonance, defined to be a state of collective activity of the behavioral system as a whole, arises when feedforward and feedback computations are consonant. Accordingly, they propose the ART1 and ART2 networks to deal with the stability and plasticity dilemma.

The network has a sufficient supply of output units, but they are not used until deemed necessary. A unit is said to be committed (uncommitted) if it is (is not) being used. The learning algorithm updates the stored prototypes of a category only if the input vector is sufficiently similar to them. An input vector and a stored prototype are said to resonate when they are sufficiently similar. The extent of similarity is controlled by a vigilance parameter, $l$ with $0 < l < 1$, which also determines the number of categories. When the input vector is not sufficiently similar to any existing prototype in the network, a new category is created and an uncommitted unit is assigned to it with the input vector as the initial prototype.

Hopfield network [Hopfield 82, 84]: Hopfield used the network energy function as a tool for designing recurrent networks and for understanding their dynamic behavior. Hopfield's formulation made explicit the principle of storing information as dynamically stable attractors and popularized the use of recurrent networks for associative memory and for solving combinatorial optimization problems. A Hopfield network with $n$ units has two versions: binary and continuous valued. Let $v_i$ be the state or output of the $i^{th}$
neuron. For binary networks \( v_i \) is 1 or 0, for bipolar \( v_i \) is +1 or -1, but for continuous networks \( v_i \) could be any real value. Let \( w_{ij} \) be the synaptic weight connecting neurons \( i \) and \( j \). The network dynamics for the binary Hopfield network are as follows:

If the state of \( i \)th neuron at time \( t \) is denoted by \( x_i(t) \), then the neuron at the next time step \( t + 1 \) is computed as

\[
x_i(\tau + 1) = \text{sgn} \left( \sum_{j=1}^{n} w_{ij} x_j(\tau) \right)
\]

where the \( \text{sgn}(x) \) is signum function that produces 1 if \( x > 0 \) and 0 otherwise. The central feature of the Hopfield network is that each state can be associated with a quantity called energy \( E \). The energy of the network at a particular state is given by [Hopfield 82].

\[
E = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} x_i x_j + \sum_{i=1}^{n} t_i x_i
\]

The energy function is of Lyapnov type which maps system state variables to real numbers and monotonically decreases with time [Kosko 92]. The central property of the energy function is that as network state evolves according to the network dynamics, the network energy always decreases and eventually reaches a local minimum point (attractor) where the network stays with a constant energy.

Associative memory [Hopfield 82; Kohonen 72]: When a set of patterns is stored as local minima (attractors or stable states) in a network, it can be used as an associative memory. Any pattern present in the basin of attraction of stored pattern can be used as index to retrieve it. The set of all initial state vectors, that converges to a stable state is called its basin of attraction. An associative memory usually operates in two phases: memory storage and information retrieval. In the storage phase, the weights in the network are determined (using a learning rule like Hebb rule) so that the attractors of the network memorize a set of patterns to be stored. In the retrieval phase, the input pattern is used as the initial state of the network, and the network evolves according to
its dynamics. A pattern is produced or retrieved when the network reaches equilibrium.

The number of patterns stored in a network is called capacity of the network. It is finite because a network with n binary neurons has a maximum of \(2^n\) distinct states and not all of them are attractors. Some attractors are called spurious attractors (or spurious states) when they store patterns different from those in the training input.

1.4 Overview of the Present Work

Associative memory concept is related to the association of stored information for a given input patterns. High capacity and accurate recall are the most desired properties of a neural network, which is being classified as an associative memory network. Models for improving and characterizing the theoretical storage capacity of associative memories have been suggested by many researchers in the recent past. The most significant work related to associative memory networks is attributed to Hopfield model [Hopfield 82, 84]. However, there are certain limitations in the models known in the literature including Hopfield, namely the practical storage capacity is 0.15n, n is the number of neurons, the stability of the patterns decreases as the number of patterns stored in the memory approaches the maximum capacity [Muller 90] and the accuracy of recall falls with the increase in the number of stored patterns [Han 89]. Various learning rules including the popular Hebb rule proposed for Hopfield network, suffer from the presence of spurious states.

These limitations have prompted us for an alternative, which has resulted in the new network termed here as Dynamical Neural Network (DNN for short) with the properties such as associative memory, pruning and order sensitive learning. It is called Dynamical Neural Network in the sense that its architecture gets modified dynamically over time as training progresses. The proposed network is a massively parallel and distributed
prompted us to propose this new learning algorithm. This has not yet been explored in the context of artificial neural networks. Another novel capability of the network, allows pruning of basic nodes in binary order as it progressively carries out associative memory retrieval and enables a part of the network parallely reusable for another task. Mathematical proofs and experimental results are presented to substantiate the above inferences.

The efficiency of the network is demonstrated with the presentation of applications in various fields. These different application areas vary from Library Database, Protein Structure Database to Natural Language Understanding.

1.5 Layout of the Thesis

This thesis is organized as follows:

Chapter 1, this chapter deals with the introduction and background for this work. The motivation for the present work is also explained in detail.

In Chapter 2, a detailed description of the proposed model is presented. The architecture of the proposed network has a composite structure wherein each node of the network is a Hopfield network by itself. The Hopfield network employs the new order-sensitive learning technique and converges to user-specified stable states but not to any spurious states. The survey on multiple neural network architectures and on learning rules is presented. The mathematical tools required for the learning rule (synaptic matrix) is presented in detail. The correctness of the learning rule is validated with the help of energy function of the network. Experimental results are presented in this chapter, which confirm the superiority of this learning rule in recall over Hebb rule. The recall efficiency of this DNN is found to be better than other single and multiple neural networks (of same configuration) for some known learning strategies.
In Chapter 3, the special characteristics of DNN, i.e., order-sensitivity and relative pruning are described. Let X and Y be two distinct training patterns. A learning rule is said to be order-sensitive if network behaves differently for different order, of presentation of the training patterns. In the present context of Hopfield network (the basic block of DNN), it is identified that the learning is order-sensitive, as the learning rule generates a different synaptic matrix when X precedes Y in the training compared to the matrix when X follows Y. It is emphasized here that order-sensitivity is one of the important features in human learning process and has not yet been explored in the context of artificial neural networks. Moreover, the natural instinct of human learning is to have longer impression of patterns that are learnt earlier.

Basing on another biological phenomenon a new notion called relative pruning is introduced for the network. Relative pruning is different from the standard notion of pruning of neural networks, which aims at removing the neurons or weights that are not participating in training and without loss of generality of the training algorithm. On the other hand, in our proposed network, the network gets pruned relatively to an extent that half the number of basic nodes are relieved in each iteration and this finally results in leaving one node in the network. The advantage of this relative pruning is that the network structure employs the hardware most optimally when it is implemented and the relatively pruned nodes can be used for parallel processing of the next series of data. In other words this capability enables a part of the network parallely reusable for another task.

Chapter 4, describes the applicability of the DNN to close-proximity match in large databases like library and protein databases. In a library retrieval system it is required to retrieve a book using inexact keywords. Similarly, in the protein sequence database determining approximate match is required to discover the relationship between newly
sequenced protein that resulted due to evolution and the various classes of proteins already available in the database. In both the cases it is required to perform approximate retrieval from large databases. The DNN is not trained by the actual data, rather with signatures which are obtained by superimposed coding on partitions of the input data. With the proposed network, this problem can be solved with an accuracy of 95% and above.

Word sense disambiguation (WSD) is the task of assigning sense labels to the occurrence of an ambiguous word. WSD is one of the hard tasks to be accomplished in Natural Language Understanding. Our premise is that the context in which the word is used, plays a major role in disambiguation and the context is largely determined by co-occurring words that are present in the sentence. In Chapter 5, it is demonstrated that DNN is suitable to handle WSD. Manually sense-tagged corpora is used for obtaining training sentences and an algorithm is used to extract the context of a particular sense of the word. The disambiguation of the word takes place by matching (associating) the signatures of the contexts of training and test sentences using the DNN.

Though the Hopfield model in its conventional form is suitable for associative memory, it is not suitable for word-sense disambiguation due to the existence of spurious stable states. The spurious stable states may lead to an invalid word-sense of a given word. But the DNN with a new learning rule to avoid any spurious states and with an enhanced retrieval capacity is suitable for word sense disambiguation. The experimental results reported confirms the theoretical findings. The disambiguator is tested on Telugu words. The performance of the disambiguator is quite good and it has an average accuracy of 83%.

In Chapter 6, a primitive hardware implementation of DNN is presented along with the
concluding remarks for the thesis.

In a nutshell, the results embodied in this thesis explain about a new architecture. This architecture betters in many aspects over its predecessors like, avoiding spurious states, converging only to user specified states and associative memory with 100% perfect recall. The advantage in this architecture is that it embeds relative pruning, a new interesting feature. The learning rule is order-sensitive, also relatively a very recent phenomenon in neural networks. The claims are corroborated with substantial experimental data.