International Journal of Current Engineering and Technology
ISSN 2277 - 4106

General Article

Neural Networks

Ritu[a*], Sangita[a] and Vinod Saroha[a]

[a]School of Engg. & science,BPSMV Khanpur Kalan,India

Accepted 24 May 2013, Available online 1June 2013, **Vol.3, No.2 (June 2013)**

Abstract

This report is an introduction to Artificial Neural Networks. The various types of neural networks are explained and demonstrated, applications of neural networks like ANNs in medicine are described, and a detailed historical background is provided. The connection between the artificial and the real thing is also investigated and explained. Finally, the mathematical models involved are presented and demonstrate.

***Keyword** :-Novel Structure, Highly Interconnected Processing element*

1. Introduction

1.1 What is a 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 (neurones) 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 neurones.

1.2 Historical background

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.

Many importand advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few reserchers. These pioneers were able to develop convincing technology which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to do too much.

1.3 Why use neural networks?

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 trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.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. Self-Organisation: An ANN can create its own organisation or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

1.4 Neural networks versus conventional computers

*Corresounding author: Ritu

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(neurones) 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 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 are 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.

Neural networks do not perform miracles. But if used sensibly they can produce some amazing results.

2. Human and Artificial Neurones - investigating the similarities

2.1 How the Human Brain Learns?

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites*. The neuron sends out spikes of electrical activity through a long, thin stand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurones. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

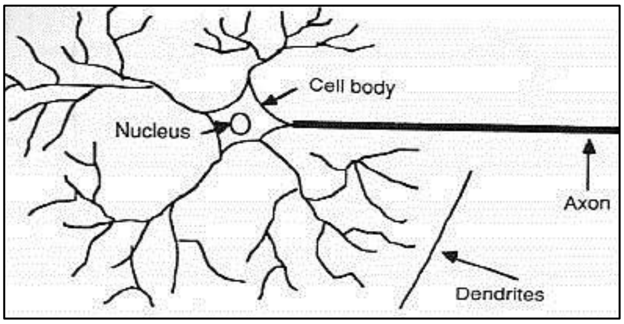

Fig. 2.1 Learning process of brain

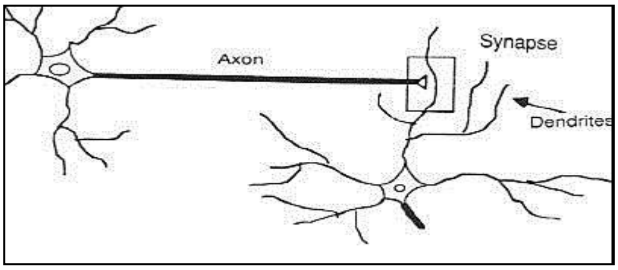

Fig.2.2 The synapse Components of a neuron

3 Architecture of neural networks

3.1 Feed-forward networks

Feed-forward ANNs (figure 1) allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organisation is also referred to as bottom-up or top-down.

3.2 Feedback networks

Feedback networks (figure 3.1) can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organisations.

3.3 Network layers

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "**input**" units is connected to a layer of "**hidden**" units, which is connected to a layer of **"output"** units. (see Figure 3.2)

- The activity of the input units represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

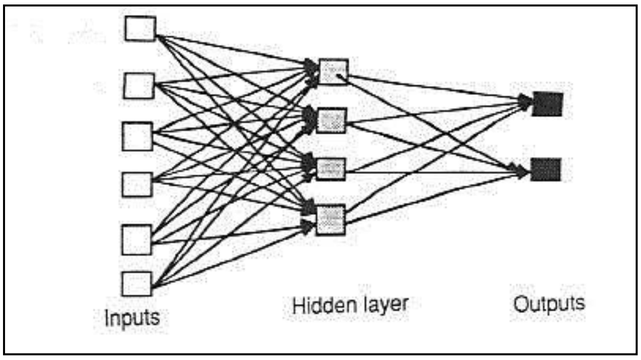

Fig. 3.1 An example of a simple feed forward network

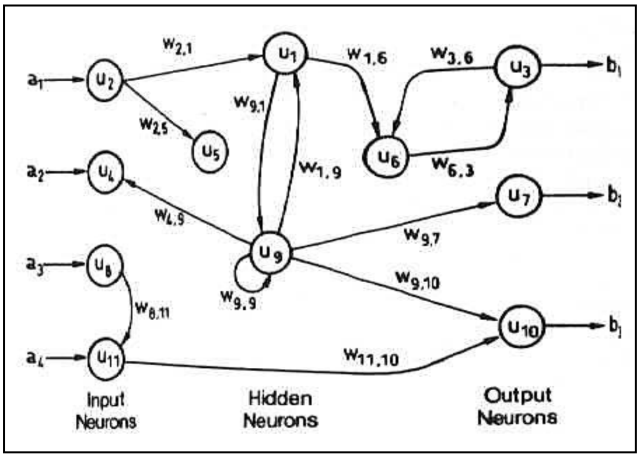

Fig. 3.2 An example of a complicated network

3.4 Perceptrons

The most influential work on neural nets in the 60's went under the heading of 'perceptrons' a term coined by Frank Rosenblatt. The perceptron (figure 3.3) turns out to be an MCP model (neuron with weighted inputs) with some additional, fixed, pre--processing. Units labelled A1, A2, Aj , Ap are called association units and their task is to extract specific, localised featured from the input images. Perceptrons mimic the basic idea behind the mammalian visual system. They were mainly used in pattern recognition even though their capabilities extended a lot more.

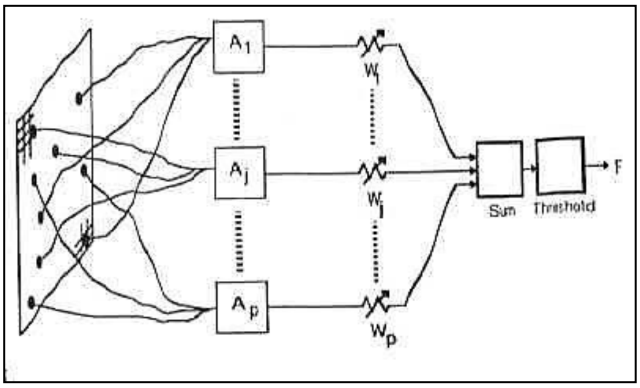

Fig.3.3 Perception

In 1969 Minsky and Papert wrote a book in which they described the limitations of single layer Perceptrons. The impact that the book had was tremendous and caused a lot of neural network researchers to loose their interest. The book was very well written and showed mathematically that *single layer* perceptrons could not do some basic pattern recognition operations like determining the parity of a shape or determining whether a shape is connected or not. What they did not realised, until the 80's, is that given the appropriate training, multilevel perceptrons can do these operations.

4. Transfer Function of neural networks

The behaviour of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (transfer function) that is specified for the units. This function typically falls into one of three categories:

- linear (or ramp)
- threshold
- sigmoid

For **linear units**, the output activity is proportional to the total weighted output.

For **threshold units**, the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

For **sigmoid units**, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurones than do linear or threshold units, but all three must be considered rough approximation.

5. Applications of neural networks

5.1 Neural Networks in Practice

Given this description of neural networks and how they work, what real world applications are they suited for? Neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries. They including

- Sales forecasting
- Industrial process control
- Customer research
- Data validation
- Risk management
- Target marketing

But to give you some more specific examples; ANN are also used in the following specific paradigms: recognition of speakers in communications; diagnosis of hepatitis; recovery of telecommunications from faulty software; interpretation of multimeaning Chinese words; undersea mine detection; texture analysis; three-dimensional object recognition; hand-written word recognition; and facial recognition.

6.Recent Neural Networks Articles

6.1 Bayesian ARTMAP for regression

Bayesian ARTMAP (BA) is a recently introduced neural architecture which uses a combination of Fuzzy ARTMAP competitive learning and Bayesian learning. Training is generally performed online, in a single-epoch. During training, BA creates input data clusters as Gaussian categories, and also infers the conditional probabilities between input patterns and categories, and between categories and classes. During prediction, BA uses Bayesian posterior probability estimation. So far, BA was used only for classification. The goal of this paper is to analyze the efficiency of BA for regression problems. Our contributions are: (i) we generalize the BA algorithm using the clustering functionality of both ART modules, and name it BA for Regression (BAR); (ii) we prove that BAR is a universal approximator with the best approximation property. (iii) we experimentally compare the online trained BAR with several neural models, on the following standard regression benchmarks: CPU Computer Hardware, Boston Housing, Wisconsin Breast Cancer, and Communities and Crime. .

6.2 Bifurcation and control in a neural network with small and large delays

A neural network modeled by a scalar delay differential equation. The focus is placed upon the Hopf bifurcation generated by varying the interaction parameter. Then, our results are tested in the two limits of small and large delays. For small delays, it is shown that a Hopf bifurcation to sinusoidal oscillations emerges as long as the interaction parameter is large enough (bifurcation from infinity) (Rosenblat & Davis, 1979). For large delays, it is pointed out that the oscillation progressively changes from sine to square-wave (Chow, Hale, & Huang, 1992; Hale & Huang, 1994). Moreover, a time delayed feedback control algorithm is introduced to generate the Hopf bifurcation at a desired bifurcation point for our neural network model.

6.3 An improved analysis of the Rademacher data-dependent bound using its self bounding property

The problem of assessing the performance of a classifier, in the finite-sample setting, has been addressed by Vapnik in his seminal work by using data-independent measures of complexity. Recently, several authors have addressed the same problem by proposing data-dependent measures, which tighten previous results by taking in account the actual data distribution. In this framework, we derive some data-dependent bounds on the generalization ability of a classifier by exploiting the Rademacher Complexity and recent concentration results: in addition of being appealing for practical purposes, as they exploit empirical quantities only, these bounds improve previously known results.

6.4 Complete synchronization of temporal Boolean networks

complete synchronization of two temporal Boolean networks coupled in the drive-response configuration. Necessary and sufficient conditions are provided based on the algebraic representation of Boolean networks. Moreover, the upper bound to check the criterion is given. Finally, an illustrative example shows the efficiency of the proposed results.

6.5 Convergence rate of the semi-supervised greedy algorithm

It proposes a new greedy algorithm combining the semi-supervised learning and the sparse representation with the data-dependent hypothesis spaces. The proposed greedy algorithm is able to use a small portion of the labeled and unlabeled data to represent the target function, and to efficiently reduce the computational burden of the semi-supervised learning. We establish the estimation of the generalization error based on the empirical covering numbers. A detailed analysis shows that the error has $O(n-1)$ decay. Our theoretical result illustrates that the unlabeled data is useful to improve the learning performance under mild conditions.

6.6 A recurrent neural network for solving a class of generalized convex optimization problems

In this,we propose a penalty-based recurrent neural network for solving a class of constrained optimization problems with generalized convex objective functions. The model has a simple structure described by using a differential inclusion. It is also applicable for any nonsmooth optimization problem with affine equality and convex inequality constraints, provided that the objective function is regular and pseudoconvex on feasible region of the problem. It is proven herein that the state vector of the proposed neural network globally converges to and stays thereafter in the feasible region in finite time, and converges to the optimal solution set of the problem.

6.7 Contractivity of a Markov operator on the space of normalised positive distributions

The goal of this article is to establish the contractivity, on the space of normalized positive distributions, of a certain class of Markov operators defined by stochastic kernels. The motivation for this work is the promising use of stationary densities in characterizing convergence properties of a certain class of discrete-time random algorithms, especially when the so-called associated ordinary differential equation has multiple asymptotically stable equilibrium, and no other stable structures.

Conclusion

- Artificial neural networks have advanced in leaps and bounds since their discovery in 1943 and their first implementation to tackle real world problems in 1958. The aim of this project was to bring these fascinating developments and the world of neural networks to users and readers of all ages in a simplistic, informative and interactive web interface. These aspects include: the history of neural networks, a general description of neural networks, the different types of architectures and the networks associated with them as well as their applications. The latest developments in the research of neural networks, are providing society with new and improved methods of tackling complex problems and tedious tasks. It can be concluded that the future of artificial neural networks and artificial intelligence looks very promising!
- The ThinkQuest 2000 Internet Challenge was both a daunting and exciting challenge to take up, and our team members and coaches have benefited and learnt a great deal about things other than neural networks like time management, group work and meeting deadlines. We hope that you have, by now, an understanding of this seemingly daunting topic.

References

An introduction to neural computing .Aleksander,i.and morton,H. nd edition

http://www.research.ibm.com

http://www.emsl.pnl.gov\

DARPA Neural Network Study.MIT Lincoln La

Neural networks,Eric Davalo and Patrick naim

http://en.wikipedia.org/wiki/neural