University of Salzburg, Institute for Computer Science

PS Introduction to Bioinformatics
SS 1999

Applications of Hopfield Networks (HN)

Mayer August	ags@gmx.net
Wiesbauer Gerald	gerald.wiesbauer@sbg.ac.at
Spitzlinger Markus	mspitzl@cosy.sbg.ac.at
Leader:	
Dipl.Ing. Mag. Roland Schwaiger	rschwaiger@cosy.sbg.ac.at

Version from March 23, 1999

Abstract

This document gives an overview of the structure of Hopfield Networks (HN) and describes some applications in detail.

Contents

Chapter 1

Introduction

1.1 About Neural Networks

The beginning of neural computing was in 1936, when Alan Turing, a British mathematician laid down the principles for defining what is and what is not computable. He also wrote about the way in which the brain did its computations through cycles of activity in neural nets. Another mathmatician, John von Neumann, also talked about neural nets, seeing them as providing a way of doing computations that emerges from the structure of the net itself.
Norbert Wiener, whose name is associated with the definition of cybernetics, thought that the fundamental structure to a mathematical understanding of computation in both brains and machines is the interplay of logic and network structure. But the algorithmic method has always been seen as the well-behaved, predictable kind of computing and so neural networks laid largely dormant for a long time.
Years of definition (1943 - 1969) followed, where a model of the fundamental cell of the living brain, the neuron, was suggested from the neurophysiologist Warren Mc Culloch and the logician Walter Pitts. Another important idea was put forward in 1949 by Donald Hebb, who suggested that a group of neurons could reverberate in different patterns, each of these patterns relating to the recall of a differnet experience. This was the first time a model of dynamic memory had been used by a neuropsychologist.
With this and a few other theories the capabilities of the nets where understood and they could be simulated on contemporary computers. This empirical approach was severely challenged in 1969 by Marvin Minsky and Seymour Papert. They wrote that neural nets are able to learn to perform tasks that requires internal represantations not explicity specified by the training data. The net must somehow work out these representation for itself and this training turns out to be a difficult task and so it is very much part of current debate and most modern learning algorithmus and net structures are being designed with the possibility of solving this difficulty in mind.
Today most nets are studied as programs running on serial computers, but the future turns upon the ability of engineers to make physically parallel systems. There is a little doubt that John Hopfield of the California Institute of Technoloy was responsible for the breakthrough of neurals nets, because his orientation was very practical. He thought that neural nets are a good way of making advanced computer memories.
Neurals nets are used in a wide range of applications. One of the most successful and proven demonstrations of the ability of neural nets is to take symbolic representation of words as letters and generate phoneme (sound segment) code that may be used to drive a voice synthesizer. The opposite problem is the speech recognition and another thing is that a net could be made to follow conversations aubout simple scences and paraphrase simple stories. But probably the major application of neural nets lies in the recognition of visual patterns.
The Future of neural computing are hybrid systems, which find out from the communication between the user and the machine what the user wants.

1.2 About Hopfield Networks

In short[1], Hopfield networks belong to the recurrent neural networks. Recurrent means that the output values are feeded back to the input. of the network. They are mainly used in the fields of pattern recognition and optimization. Learning is performed by modifying the strength of the connections between the neurons. The network consists of a layer where the output of each neuron is connected to the input of all other.

[1] see [Bäu97]

The neurons of the network have only two states: 0 (not firing) or 1 (firing). Whether a neuron i is firing or not is a result of the values at its input, i.e. if these values surpass a specific threshold U_i, which is distinct for every neuron, the neuron's state will change to firing. Input values is caused by other neurons firing, and the strength or weight of a neuron-to-neuron connection is defined during the learning process.
A simple connection of neurons is shown in picture 1.2. In this example, the state of each neuron in this network is initially 0. This state causes neuron 1 to fire. Both of the other neurons have the same probability to fire. After some cycles the state 0/1/1 will be reached; in this case the network will not change its state any more. This leads us to the supposition that there are stabile states in the network. If one of these states is reached the network is "jailed", which means that there is no way for the network, to leave this state called an attractor.

1.2.1 Learning in Hopfield networks

Neural networks, in particular HNs, can learn their stable states from examples[2]. Rules which can be used for this are Hebb's learning rule (1949) and a development of this rule, the Widrow-Hoff rule (1960). Hebb's rule states that the connection strength between two neurons changes proportionally to the frequency of the neurons firing together. The Widrow-Hoff rule or Delta learning rule says that the greatest sources of error are from active lines, so that

$$\text{Weight Change} = (\text{Pre-Weight Line Value}) * \frac{(\text{Error})}{(\text{Number of Inputs})}.$$

The error signal is measured by calculating the activation of a neuron and then comparing this to the required signal, i.e.

$$\text{Error} = (\text{Desired Output}) - (\text{Actual Output}),$$

where only 'active' neurons' weights are recalculated.
So the network performs learning by feeding the desired state to the network; if this state isn't stable, the values of the thresholds and inter-neuron connection strengths are changed until the desired stability is reached.
With this system it is theoretically possible to understand the network by making a step-for-step calculation and evaluation of the network states and the resulting error.
Hopfield himself thought more in terms of an energy-based analysis. The energy of a neuron i is given by

$$E_i = -\left[V_i\left(\sum_j T_{ij}V_j - U_i\right)\right] = -\left[V_i \times (\text{activation})\right]$$

where V_i is the input state, U_i is the state of a neuron and T_{ij} is the weight of the connection between neuron i and neuron j.
There should be also noted that if an unit has to fire, but its activation is negative, the Widrow-Hoff rule causes it to be made positive, so that it decreases the total energy. If a unit is firing and should not, the activation is made negative along with the value of V_i, which again leads to a reduction of energy[3].
The effect of the Widrow-Hoff learning rule is that the energy of the appropriate state is lowered in comparison with its 'neighbouring' states (i.e. states differing by only one node).
Another aspect is that such networks can be considered as inequalities, and training the former provides an automatic way of solving such equations, which would be relatively time-consuming using standard methods.

[2]For a detailed step-by-step example, see [Ale93, pp.105–109]. Also derived from [Olm98].
[3]For an example, again see [Ale93, p.109]

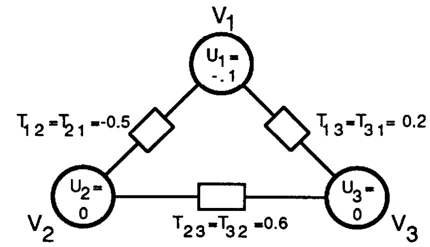

Figure 1.1: A simple example of a Hopfield network.

Appendix A

Records

A.1 Record of March 10th, 1999

Leader: DI Mag.Roland Schwaiger
Recorder: Mayer August
Participants: Spitzlinger Markus
Wiesbauer Gerald
Place: Salzburg
Time: 12^{30}

- General discussion
- Website under http://www.cosy.sbg.ac.at on Mr. Schwaigers Homepage
- Requirements for the paper
  - Structure
  - Records in the appendix
  - Goal: Understanding of the topic, (detailed) description of some applications
- LaTeX-file until Di 13^{00} as TeX-Source
- Possible Applications:
  - TSP (Travelling Salesman Problem)
  - Associative Memory: Picture searching (IBM), denoising of pictures
- Should programs be written?
  - not required, but appreciated
  - also: search ready programs in the Internet
  - utilization of Java
  - would be good for presentations of the institute
- Are there ready-made simulators?
  - SNNS (Stuttgarter Neuronale Netze Simulator) is/will be installed
  - optional: search in the net
- Literature
  - Igor Aleksander, "'Introduction to Neural Computing"'
- Activities until next week:
  - Compressed overview of Hopfield networks
  - Later: search applications in the Internet

A.2 Record of March 17th, 1999

Leader: DI Mag. Roland Schwaiger
Recorder: Spitzlinger Markus
Participants: Mayer August
Wiesbauer Gerald
Place: Salzburg
Time: 12^{30}

- Old points:
  - Title-page
  - Photos (Wiesbauer, Mayer)
  - Picture

- New points:
  - Glossary
  - Studying Igor Aleksander, "'Introduction to Neural Computing"'
  - Detailed description of Hopfield Networks
    - Example
    - How learns a HN?
  - Simulate a HN with SNNS

Appendix B

Glossary

cybernetics: The comparative study of the internal workings of organic and machine processes in order to understand their similarities and differences. Cybernetics often refers to machines that imitate human behaviour.

firing: The word comes from the language of the biologist interested in the real neurons of the brain. A neuron is said to fire when it emits a buzz of electrical pulses at the rate of about 100 per second.

firing rule: It determines how one calculates whether the node should fire for any input pattern.

Neural Computing: Is the study of networks of adaptable nodes which, through a process of learning from task examples, store experiential knowledge and make it available for use.

neuron: Is the adaptive node of the brain.

nodes: Are adaptable of the nodes of the brain, they acquire knowledge through changes in the function of the node by being exposed to examples.

strength: Firing rules are couched in terms of some strength (or weight) associated with each connection in the net.

wells: Wells are stable states.

weight: Firing rules are couched in terms of some strength (or weight) associated with each connection in the net.

Bibliography

[Ale93] I[gor] Aleksander. *An Introduction to Neural Computing*. Chapman&Hall, 1993.

[Bäu97] Ch[ristian] Bäumer. Hopfield netze. Vortrag im Seminar Neuronale Netze und Fuzzy-Systeme im Sommersemester 1997, 1997.

[Olm98] D[avid] D. Olmsted. History and principles of neural networks. . http://www.neurocomputing.org/history.htm, 1998.