Savitribai Phule Pune University
Gokhale Education Society's

R. H. Sapat College of Engineering, Management Studies and Research,
Nashik - 422 005, (M.S.), INDIA

DEPARTMENT OF COMPUTER ENGINEERING
Third Year Computer Engineering
Year 2022– 2023

Roll No: 41

Name of Student: Wafiyah Muzffarali Sayyad

Mobile No.: 8698093881

official-Mail ID: 21_wafiyah.sayyad@ges-coengg.org

Seminar Title : Neuromorphic Hopfield Network

Seminar Guide : Ms.R.D.Narwade

Area of the Seminar: Artificial Intelligence and Machine Learning

Abstract

A neuromorphic Hopfield network combines the principles of Hopfield networks, a type of recurrent artificial neural network, with neuromorphic computing, which aims to replicate the architecture and functionality of the human brain in hardware or software.

Neuromorphic computing is a method used in engineering in which the elements of a computer are modeled as the human brain or the nervous system of the human. Neuromorphic systems are inspired by Biology. Neuromorphic systems are based on highly connected synthetic neurons and synapses inspired by biological networks.

The requirements for a neuromorphic architecture are high connection density, parallelism, low power consumption and memory colocation and processing capabilities. The network topologies like feed forward neural networks, recurrent neural networks, spiking neural networks are employed. These systems have faster computing speed.

Hopfield neural network is a single layer feedback neural network proposed by physicist John Hopfield in 1982. It is a recurrent neural network with feedback connections between input and output layers. Types of Hopfield neural network include discrete hopfield neural network and continuous hopfield neural network. Hopfield neural network has been used to solve travelling salesman problem optimistically.

Designing a Neuromorphic Hopfield Computing system involves creating a recurrent neural network with binary neurons and symmetric connections. Train the network to store and retrieve patterns using the Hopfield update rule. Implement, fine-tune, and test the network for your specific problem, ensuring efficient and reliable operation.

Neuromorphic Hopfield Networks offer energy-efficient, fault-tolerant memory and pattern recognition. They excel in applications like content retrieval, image recognition, and optimization, making them ideal for resource-constrained environments. However, they have limitations, like storage capacity constraints and susceptibility to spurious states, which should be considered in their application.

Introduction

A Neuromorphic Hopfield Network (NHN) is a specialized type of recurrent artificial neural network that takes its inspiration from the biological Hopfield network, which was originally proposed by John Hopfield in 1982. NHNs are designed to replicate certain aspects of the human brain's associative memory and pattern recognition capabilities. These networks are characterized by their binary neuron states, symmetric connections, and energy-based computation. The NHN architecture aims to provide a content-addressable memory system, enabling the network to retrieve information based on its content rather than specific addresses, much like the human brain's ability to recall memories based on context.

One of the fundamental features of NHNs is their symmetric connectivity pattern, where each neuron is connected to every other neuron. These symmetric connections are crucial for their associative memory capabilities. The connections, or weights, between neurons are typically initialized randomly, and the network learns to store patterns by adjusting these weights during a training phase. The Hopfield update rule is often used for this training process, allowing NHNs to converge to stable states that correspond to stored patterns.

In practical terms, NHNs have found applications in various domains, including content-addressable memory systems, content retrieval, optimization problems, and even in the field of neuromorphic computing, where researchers aim to create hardware systems that mimic the brain's behavior. However, it's important to note that NHNs have their limitations, such as a limited storage capacity and the potential for spurious states, which can affect their applicability to certain tasks. Despite these limitations, they remain a valuable tool in the toolbox of neural computing, particularly for tasks requiring associative memory and content-based retrieval.

Combining neuromorphic computing principles with the Hopfield neural network framework in designing a Neuromorphic Hopfield Network aims to capitalize on the energy-efficient and brain-inspired aspects of neuromorphic computing while harnessing the associative memory and pattern recognition capabilities of Hopfield networks. This fusion can result in a powerful computational tool with potential applications in various domains, particularly those requiring efficient content-based memory retrieval and pattern recognition.

Method/Algorithm

Creating a Neuromorphic Hopfield Network (NHN) involves training it to store and retrieve patterns using specific algorithms. One of the most common algorithms for NHNs is the **Hopfield update rule**.

Step 1 : Initialization

- Initialize the NHN with a set of binary neurons. Each neuron has a binary state, which can be +1 or -1 (or 0 or 1, depending on your representation).

Step 2 : Weight Initialization

- Initialize the connection weights between neurons. These weights represent the strengths of connections and can be initialized randomly or using some other scheme.

Step 3 : Pattern Storage

- Train the NHN to store patterns by updating the connection weights. For each pattern you want to store:
  - Convert the pattern into a binary format (e.g., 0s and 1s).
  - Update the connection weights based on the outer product of the pattern with itself and subtract the identity matrix. This can be represented as:

$W = W + (p * p') - I$

Where:
- `W` is the weight matrix.
- `p` is the binary pattern.
- `p'` is the transpose of `p`.
- `I` is the identity matrix.

- Repeat this process for all patterns you want to store.

Step 4 : Pattern Retrieval

- To retrieve a pattern, provide an initial input pattern to the NHN. This input pattern can be a complete pattern or a noisy/incomplete version of a stored pattern.

- Update the neuron states iteratively using the following steps until the network stabilizes (reaches a fixed point) or until a maximum number of iterations is reached:
  - Calculate the new state of each neuron using the following equation for each neuron `i`:

$S\_i(t+1) = \text{sign}(W\_i * S(t))$

Where:
- `S_i(t+1)` is the state of neuron `i` at the next time step.

  - `W_i` is the row of the weight matrix corresponding to neuron `i`.
  - `S(t)` is the current state of all neurons at time step `t`.

Step 5 : Pattern Convergence and Output

- The NHN will eventually converge to a stable state. This stable state represents the retrieved pattern.
- The output pattern can be the result of the converged neuron states.

Step 6 : Spurious States Handling

- Check if the converged state is indeed one of the stored patterns. If not, it may be a spurious state.
- Employ techniques like adding noise to the input pattern or using energy-based measures to handle and correct spurious states.

Step 7 : Repeat for Other Patterns

- You can repeat the retrieval process for multiple patterns by providing different initial inputs.

Step 8 : Evaluation and Termination

- Evaluate the accuracy and reliability of pattern retrieval using appropriate metrics.
- Terminate the retrieval process when the desired pattern is successfully retrieved or when a predefined stopping criterion is met.

This Hopfield update rule algorithm allows the NHN to learn and retrieve patterns in a content-addressable manner, making it a valuable tool for associative memory and pattern recognition tasks.

There are other algorithms such as **Hebb Algorithm**, **Training Algorithm** for **Discrete Hopfield Neural Network (DHNN)** that are used.

References

1] Zheqi Yu , Amir M. Abdulghani, Adnan Zahid , Hadi Heidari, Muhammad Ali Imran , James Watt, Qammer H. Abbasi on “An Overview of Neuromorphic Computing for Artificial Intelligence Enabled Hardware-Based Hopfield Neural Network ” IEEE VOLUME 8, no.: 67097, April 21, 2020.

2]https://en.wikipedia.org/wiki/Hopfield_network#:~:text=Updates%20in%20the%20Hopfield%20network,updated%20at%20the%20same%20time.

3]https://www.geeksforgeeks.org/hopfield-neural-network/

4]https://www.javatpoint.com/artificial-neural-network-hopfield-network#:~:text=If%20Wij%20%3E%200%2C%20the,its%20value%20Xj%20%3D%20%2D1