Constructing an Associative Memory System Using Spiking Neural Network

A Seminar Report

submitted by

SANKAR VINAYAK E P

PKD19CS046

to

the APJ Abdul Kalam Technological University

in partial fulfillment of requirements for the award of degree

of

Bachelor of Technology

in

Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT ENGINEERING COLLEGE PALAKKAD

SREEKRISHNAPURAM 678 633

DECEMBER 2022

DEPT. OF COMPUTER SCIENCE ENGINEERING GOVERNMENT ENGINEERING COLLEGE PALAKKAD

2022 - 23

CERTIFICATE

This is to certify that the report entitled **Constructing an Associative Memory System Using Spiking Neural Network** submitted by **SANKAR VINAYAK E P** (PKD19CS046), to the APJ Abdul Kalam Technological University in partial fulfillment of the B.Tech. degree in Computer Science and Engineering is a bonafide record of the seminar work carried out by him under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Liji L Dominic
(Seminar Guide)
Assistant Professor
Dept.of CSE
GOVERNMENT ENGINEERING
COLLEGE PALAKKAD

Dr. Swaraj K P
(Seminar Coordinator)
Associate Professor
Dept.of CSE
GOVERNMENT ENGINEERING
COLLEGE PALAKKAD

Dr. Sabitha S
Professor and Head
Dept.of CSE
GOVERNMENT ENGINEERING COLLEGE
PALAKKAD

DECLARATION

I SANKAR VINAYAK E P hereby declare that the seminar report **Constructing an Associative Memory System Using Spiking Neural Network**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Liji L Dominic

This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources.

I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Sreekrishnapuram

15-12-2022

SANKAR VINAYAK E P

Acknowledgement

I take this opportunity to express my deepest sense of gratitude and sincere thanks to everyone who helped me to complete this work successfully. I express my sincere thanks to **Dr. Sabitha S**, Head of Department, Computer Science and Engineering, Government Engineering College Sreekrishnapuram for providing me with all the necessary facilities and support.

I would like to express my sincere gratitude to **Dr. Swaraj K P** and **Ali Akbar N**, department of Computer Science and Engineering, Government Engineering College Sreekrishnapuram for their support and co-operation.

I would like to place on record my sincere gratitude to my seminar guide **Liji L Dominic**, Assistant Professor, Computer Science and Engineering, Government Engineering College for the guidance and mentorship throughout the course.

Finally I thank my family, and friends who contributed to the succesful fulfilment of this seminar work.

SANKAR VINAYAK E P

Abstract

An associative memory system is a type of artificial neural network that can learn and store associations between input and output patterns. Spiking neural networks, on the other hand, are a type of neural network that models the behaviour of neurons in the brain by using discrete time steps to simulate the firing of individual neurons. Combining these two concepts can result in an effective memory representation technique in which the contents can be accessed with speed and efficiency. The report provides an overview of the principles of associative memory and spiking neural networks, and then describe the architecture and training procedure for the system. The results show that spiking neural networks can be effective for implementing associative memory systems, and have potential applications in a range of computational neuroscience and machine learning problems.

Contents

List of Figures

Abbreviation

Abbreviation	Description
ANN	Artificial neural network
NN	Neural network
SNN	Spiking neural network
RNN	Recurrent neural network
STDP	Spiking-time-dipendent plasticity
LIF	Leaky integrate and fire

Chapter 1

Introduction

The ability to store and retrieve associations between different stimuli is a fundamental component of many cognitive processes, including perception, learning, and memory. Associative memory is a type of memory system that allows for the storage and retrieval of information based on the relationships between different items in memory. It is a key component of a good deal of artificial intelligence and machine learning systems and has been extensively studied in both neuroscience and computer science. According to research done by google for fast contextual adaptation of speech [3] associative memory system using ANN are efficient in case of contextual adaptation of information.

Spiking neural networks (SNNs) are a type of neural network that can simulate the dynamics of individual neurons and synapses in the brain. They are effective for modelling a range of cognitive and sensory processing tasks and have potential applications in a variety of fields, including computational neuroscience and machine learning.

This work presents a study on the construction of an associative memory system using a spiking neural network. We describe the architecture and training procedure for our system and evaluate its performance on a variety of associative memory tasks. We discuss the implications of our results for the use of SNNs in implementing associative memory systems and highlight their potential applications in computational neuroscience and machine learning.

Chapter 2

Literature Review

2.1 Associative Memory

Associative memory also known as content addressable memory is a type of memory which is specially optimized for access to memory locations without using the memory address of the location that needs to be accessed. Its electronic circuit will have extra connections which enable it to parallelly search through the contents in a single clock pulse. It is widely used in applications like database management systems which require searching through the data as fast as possible figure2.1 shows one such circuit.

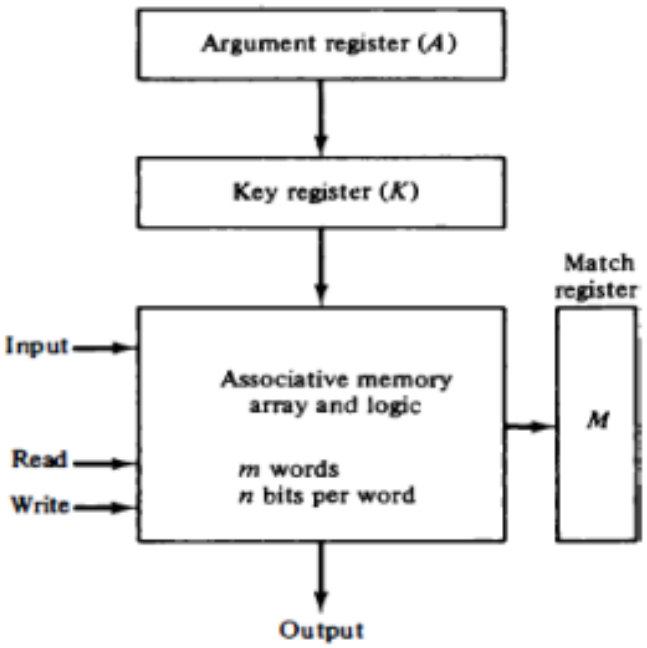

Figure 2.1: Associative memory circuit

2.2 Neural associative memory

Neural associative memory is also known as the associative network, which works based on pattern association. It can store different patterns and produce the output which closely matches the already given patterns. They are implemented using the artificial neural network, which tries to mimic the working of the brain. They are commonly used in applications like pattern recognition, data storage and information retrieval. There are two types of associative networks

Auto associative memory A single layer NN in which the number of training vectors and the number of output vectors are the same. The weights are determined by the stored patterns

Hetero associative memory A single layer NN in which the number of input training vectors and output are different. Weights are determined by the pattern stored in the network. It is static in nature and hence there would be no linear or delay operations

2.2.1 Hopfield model

The Hopfield model [2] is a type of RNN. The model is designed to mimic the behaviour of neurons in the brain. It is a type of associative memory system and hence it can store and recall information based on the relationship between the data stored. It has application including pattern recognition, optimization, and error correction.

It is a fully connected NN and weights between connections determine the strength of the connections. When the network is supplied with an input these weights are updated sequentially until the network reaches a stable state, at which the output is determined. By adjusting the weights in a specific way, the the network is able to recall the information as a set of stable states

The main drawbacks of the model are that it may not converge to correct output state when it is supplied with a pattern which is only partially similar to the stored patterns or when it is not trained on sufficiently distinct data.

2.3 Spiking neural network

Spiking neural networks are a type of neural network that models the behaviour of biological neurons by using spikes or pulses to encode and transmit information. They are a relatively new type of neural network that has the potential to improve the performance and efficiency of artificial intelligence systems. The use of spiking neural networks for building associative memory systems is a relatively new area of research that has only recently started to gain attention.

2.3.1 Spike time dependent plasticity

Spike-timing-dependent plasticity (STDP) [6] is an unsupervised learning rule based on the functioning of neurons in the brain for neuromorphic computing, which is the study inspired by the structure and functioning of the brain. In the process strength of the connection between neurons change based on the relative timing of spikes or impulses

The basic idea behind STDP is that if two N_{pre} and N_{suc} neurons are connected and their spike time are t_1 and t_2 respectively according to STDP

Weight of connection from N_{pre} to N_{suc} should increase, if $\boldsymbol{t_1 > t_2}$

Weight of connection from N_{pre} to N_{suc} should decrease, if $\boldsymbol{t_1 < t_2}$

Weight of connection from N_{pre} to N_{suc} should remain same, if $\boldsymbol{t_1 = t_2}$

This process allows the neurons to adjust connection in a way which reflects the relationship between input and output spikes signals.

2.3.2 SpikeProp

SpikeProp [4] is an unsupervised learning algorithm used in the field of neuromorphic computing used to train SNN based on the principle of STDP. It is similar to the gradient descent algorithm used in conventional deep neural networks. It is computationally efficient and well-suited for real-time applications. The issue with this algorithm is the need for a large amount of data to achieve good results and only applicable to SNN.

Chapter 3

Methodology

Construction of associative memory using SNN [1] in this method consist of four phases

1. Initialization: Initialization of SNN and the input spiking signals
2. Structure formation: New connection with neighboring neuron are formed
3. Parameter training: Optimize weight of synapse based on STDP
4. Pruning: Removing unnecessary connection to improve efficiency

3.1 Initialization

It involves two sub process in which the data is preprocessed and converted into spiking signals and the network is initialized

3.1.1 Initialization input spiking signals

The input to the SNN are spiking signals for that the input values need to be converted into spiking signals. For the example purpose here MNIST dataset is used which contains handwritten characters on digits. The figure3.1 show the steps involved

Four convolutional kernels of size 4×4 shown in figure3.2 is used to extract the features from the image pixel values. The input image into the kernel is of size 28×28 and the convolutional kernels reduce the shape into 24×24. Next these values are passed through a max pooling layer of size 2×2. It reduces the size of image to 12×12. The conversion layer convert these values into the spiking encoding. Pixel

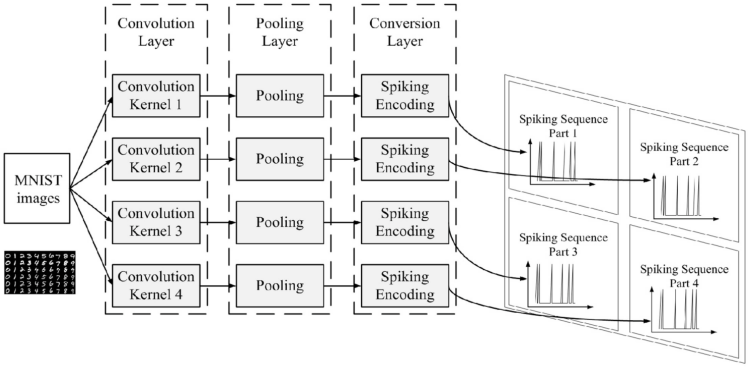

Figure 3.1: Data preprocessing

value in the range of [0,255] converted to delay in spike from [0,100]ms. The higher the value, shorter the delay. In order to cover the values first min-max normalization is

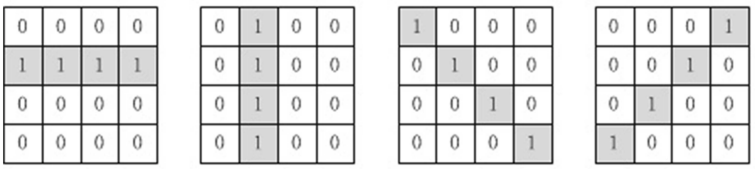

Figure 3.2: Kernels used

used in which, if the value of a pixel is d then,

$$R(d) = \frac{d - d_{\min}}{d_{\max} - d_{\min}}$$

Where $d_{\min}$ and $d_{\max}$ are maximum and minimum pixel values. Power encoding is used to get the spike time of pixel d,

$$S(d) = (R(d) - 1)^2 \times (T_{\max} - T_{\min}) + T\min$$

where $T_{\min}$ and $T_{\max}$ are starting and stopping time of spike.

3.1.2 Initialization of Spiking neural network

The memory NN in this method consists of three layers input, memory and output as shown in figure3.3. The spiking signal input is fed into the input layer. For the neuron, the LIF model is used. The memory layer grows new connections to remember them. The output layer is responsible for generating the output. The number of neurons in both the input and memory layers is the same as the number of input spiking signals which is 576. The output layer consists of 10 neurons which are equal to the number

of output classes.

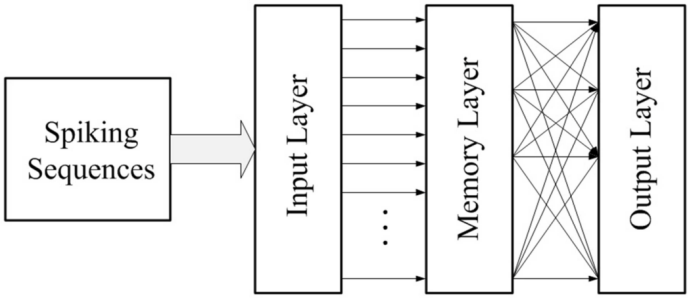

Figure 3.3: Structure of network

The connection from the input layer to the memory layer is one-to-one style. The weight of the synapse is set to 50 as an initial value. It helps in provoking enough response in the memory layer based on Hebb's learning rule [7] to take place.

Each neuron in layers is assigned a coordinate value. By using a spatial to temporal mechanism which encodes the spatial information of pixel values into delay of connection from the input layer to the memory layer as shown in the figure3.4. The delay of a connection from a neuron $i(x, y)$ where x and y are coordinates of the pixel values in a $p \times q$ input layer to corresponding neuron in the memory layer is calculated as

$$delay_{im(x,y)} = x * p + y + 1$$

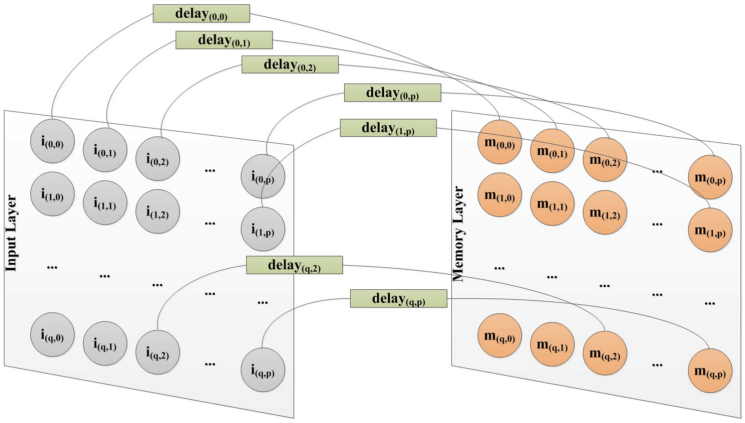

Figure 3.4: Delay in neuron connections

3.2 Structure formation

During the structure formation phase are fed into the network. The behavior of neurons in the memory layer is recorded, and new connections are made within the memory layer based on Hebb's learning rule [7]. New connection conditions are based on their threshold values of distance and time difference in firing two neurons to avoid explosive growth of connections. If there are two neurons which satisfy the threshold conditions on delay and distance in firing and there is no connection between them a new connection is established between them.The smaller the threshold lesser the number of connections that will be created.

It is repeated until the stopping condition is satisfied. Due to the leaking characteristics of the LIF model used the connection from the memory layer to output layer implements a spatial-to-temporal mechanism discussed earlier is used. The delay of connection from the memory layer to the output layer is calculated as

$$delay_{mo(x,y)} = [N_m - delay_{im(x,y)}] + 1$$

3.3 Parameter training

This phase is based on the ideas of STDP and reinforcement learning. This phase checks the recall ability of the network for the set of inputs. This phase does not change the weights of the connection between layers but rather changes the weight of the connection within the memory layer itself. For this process when a spiking sequence is fed into the network the most frequently fired neuron in the output layer is considered as output. If the network could correctly recall then no optimization need to be done else the weights need to be adjusted. It works based on the following algorithm

Step 1:Pick an input image

Step 2:Feed input to the network

Step 3:Pick If the result of the output layer is correct go to 1 else 4

Step 4:Identify incorrectly firing set of neurons in output layer S_O memory layer S_M

Step 5:If i is a neuron in S_M and and j is a neuron in S_O and weight of connection between them is $W_{i,j}$, then $W_{i,j} = W_{i,j} * Shrink\_Coeff$

This process repeated for all the images.The value of $Shrink\_Coeff$ is constant between 0 and 1.figure3.5 shows the firing behavior of the memory layer when it is supplied with an input image corresponding to the number six. Colours indicate the time at which the spike occurred and lines indicate the connection between neurons.

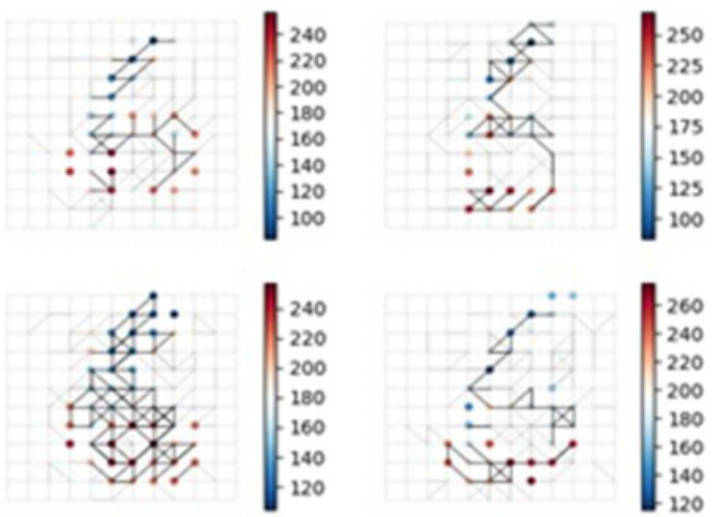

Figure 3.5: Recall response for number 6

3.4 Pruning

This phase help in improving the efficiency of the network. In this phase if the weight of a connection is less than the threshold (here 3), connection will be removed. Also, if there is a neuron in memory layer in memory layer which does not have any connections to output layer, the connection from input layer to that neuron will also be deleted.

Chapter 4

Results and Discussion

4.1 Growing process of memory layer

During the initialization phase, there was no connection between neurons in the memory layer, but during the structure formation phase, new connections were grown in the memory layer of the network. This network could grow different connections in the memory layer to memorize different patterns. When the threshold was set to smaller values the connection was more sparse and the memory could remember more information with more availability of free space.

4.2 Recall process

To check the recall process all images are used in the structure formation phase. Then used the same images in both parameter training phase and pruning phase The memory layer generated observed. There is different response based on the four different kernels used show by four images.When an image is fed into the network, different parts of the image provoke different firing responses.As described previously an image is fed into the network and the firing sequence in output layer decides the result using majority voting technique. The results show that the network could recall the images it memorized When it was supplied with an unseen but similar image the network could show some association ability with the new data and the previously memorized data

Chapter 5

Conclusion

This study presented a method for constructing an associative memory system using a SNN. This approach combines the principles of associative memory and SNNs to create a neural network architecture that can store and retrieve associations between different stimuli.

There are active research going on to develop neoromorphic hardware which improve the efficiency of different operation involved in processing of information using an SNN. The hardware Lohi and software for that Lava are developed by intel to this application. The software like **N**EST and SpikeTorch are open source tools for the SNN. As more and more research is done in this field, it could be the next generation machine learning algorithm.

References

[1] He, Hu, et al. *"Constructing an Associative Memory System Using Spiking Neural Network."* Frontiers in Neuroscience, vol. 13, 2019, https://doi.org/10.3389/fnins.2019.00650.

[2] Hopfield J. J. (1982). *Neural networks and physical systems with emergent collective computational abilities.* Proceedings of the National Academy of Sciences of the United States of America, 79(8), 2554–2558. https://doi.org/10.1073/pnas.79.8.2554

[3] Munkhdalai, Tsendsuren, et al. *"Fast Contextual Adaptation with Neural Associative Memory for On-Device Personalized Speech Recognition."* arXiv, 2021, https://doi.org/10.48550/arXiv.2110.02220.

[4] Bohté, Sander M. et al. *"SpikeProp: backpropagation for networks of spiking neurons."* The European Symposium on Artificial Neural Networks (2000).

[5] G. Indiveri, *"A low-power adaptive integrate-and-fire neuron circuit,"* Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03., 2003, pp. IV-IV, doi: 10.1109/ISCAS.2003.1206342.

[6] Wade, J. J., McDaid, L. J., Santos, J. A., & Sayers, H. M. (2010). *SWAT: a spiking neural network training algorithm for classification problems.* IEEE transactions on neural networks, 21(11), 1817–1830. https://doi.org/10.1109/TNN.2010.2074212

[7] Morris R. G. (1999). D.O. emphHebb: The Organization of Behavior, Wiley: New York; 1949. Brain research bulletin, 50(5-6), 437. https://doi.org/10.1016/s0361-9230(99)00182-3

Appendices

Hebb's learning rule: [7] When an axon of a cell A is sufficiently close to excite a cell B, and repeatedly and or persistently takes part in firing it some growth related process or metabolic changes takes place in one or both of the cell such that A's efficiency, as one of the cells firing B is increased.

Leaky integrate and fire [5]

$$V(t) = \begin{cases} \beta.V(t-1) + V_{in}(t) \ \ when \ \ V < V_{th} \\ V_{reset} and set spike \ \ when \ \ V \geq V_{th} \end{cases}$$