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

2022 - 23

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 fulfilment 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

31-12-2022

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**, **Prof Ali Akbar N** and **Prof. Irshad M**, 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 Sreekrishnapuram for the guidance and mentorship throughout the course.

Finally, I thank my family, and friends who contributed to the successful 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 describes 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
CAM	Content Addressable Memory
ANN	Artificial neural network
NN	Neural network
SNN	Spiking neural network
RNN	Recurrent neural network
STDP	Spiking-time-dependent 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 [1] 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.

Presented in this work is a study on the construction of an associative memory system using a spiking neural network. This study describes the architecture and training procedure of the system and evaluates its performance as a associative memory . It also discusses the implications of the results for the use of SNNs in implementing associative memory systems and highlights 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 was first proposed in the early 1960s by researchers at IBM, including Richard W. Harker, Kenneth C. Thompson, and Robert D. Denny. CAM is a type of computer memory that allows for the rapid searching and retrieval of data by using the content of the data as the address.

In traditional random access memory (RAM), data is stored in a specific location based on a numerical address and the data can be retrieved by accessing the corresponding address. In contrast, CAM stores data in a specific location based on the content of the data and the data can be retrieved by searching for the specific content rather than the numerical address.

CAM has several advantages over traditional RAM, including faster search and retrieval times and the ability to store and retrieve data based on the content rather than a numerical address. These features make CAM particularly useful in applications where rapid searching and retrieval of data is important, such as database management and pattern recognition.

The concept of CAM has had a significant impact on the field of computer science. The work of Harker, Thompson and Denny has contributed to the development of efficient and effective methods for storing and retrieving data in computers. Its applications include database management systems which require searching through the data as fast as possible figure2.1 shows one such circuit.

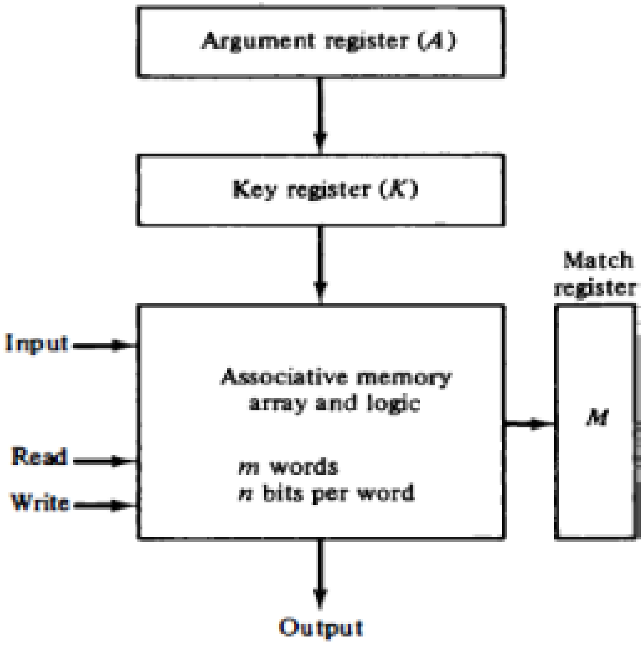

Figure 2.1: Associative memory circuit

2.2 Associative network

An associative network is a type of artificial neural network that is designed to store and retrieve information based on the strength of the connections between neurons. Associative networks are often used for tasks that require the recall of specific patterns or relationships, such as image recognition, natural language processing, and recommendation systems.

2.2.1 An Interactive Activation Model of Context Effects in Letter Perception

Introduction

The interactive activation model [2] is a computational model that attempts to explain how the brain processes and interprets written language. It suggests that the brain stores and retrieves information about letters and words based on the strength of the connections between neurons, and that these connections are influenced by both bottom-up and top-down processes.

The model proposes that the brain has a hierarchical structure, with lower-level processing units representing features such as lines and curves, and higher-level processing units representing letters and words. According to the model, the activation of a processing unit depends on both the activation of its inputs and the activation of its outputs.

Proposed system

The authors propose a hierarchical structure for the model, with lower-level processing units representing features such as lines and curves, and higher-level processing units representing letters and words. They suggest that the activation of a processing unit depends on both the activation of its inputs and the activation of its outputs. The main two concepts in this system are

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.

Advantages

One advantage of the model is that it provides a computational framework for understanding how the brain processes and interprets written language. It suggests a hierarchical structure for the brain, with lower-level processing units representing features such as lines and curves, and higher-level processing units representing letters and words. Another advantage of the model is that it is based on the concept of auto-associative and heteroassociative memory, which are thought to play a role in perception and language processing in the human brain. This allows the model to capture the complex relationships between letters and words, and to explain a wide range of phenomena in written language processing.

Disadvantages

A disadvantage of the model is that it is a simplified model of the brain, and it may not capture all the complexity and nuance of language processing. It is also based on a set of assumptions and simplifications, and it may not accurately reflect the underlying mechanisms of the brain. In addition, the model is based on a set of equations and learning rules that are used to simulate the activation and deactivation of processing units, and these equations may not accurately reflect the underlying mechanisms of the brain.

2.2.2 Neural networks and physical systems with emergent collective computational abilities.

Introduction

The Hopfield model [3] is a type of RNN that 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 applications including pattern recognition, optimization, and error correction.

Methodology

This paper discusses the use of computational models and simulations to study the behaviour of neural networks and other distributed systems. He proposed an associative memory network type called the Hopfield network. A Hopfield network consists of a set of interconnected neurons that are arranged in a single layer. Each neuron is fully connected in the network, and the connections between neurons are adjusted based on the strength of the relationships between the inputs and outputs. The behaviour of a Hopfield network is determined by a set of equations that describe the activation and deactivation of the neurons over time. These equations are based on the concept of auto-associative memory, in which the output is used to retrieve the original input.

Advantages

One advantage of the Hopfield model is that it is relatively simple and easy to implement. It consists of a single layer of interconnected neurons and the connections between neurons are adjusted based on the strength of the relationships between the inputs and outputs. This simplicity makes the Hopfield model easy to understand and implement. Another advantage of the Hopfield model is that it is capable of storing and retrieving patterns or sequences of data based on the strength of the connections between neurons. This makes it useful for tasks that require the recall of specific patterns or sequences of data, such as image recognition, natural language processing, and recommendation systems. Also, this model is relatively robust and resistant to noise. These advantages make it useful in applications like image or speech recognition systems.

Disadvantages

One disadvantage of the Hopfield model is that it is relatively simple and may not be able to capture the complexity and nuance of more advanced neural networks. Another drawback is that it can retrieve the information only with the original input. A third disadvantage of the Hopfield model is that it can be sensitive to initialization and may not always converge to a stable state when used with a large amount or insufficiently distinct data. This can make it difficult to use the model for certain types of tasks, such as optimization or decision-making.

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 SWAT: a spiking neural network training algorithm for classification problems.

Introduction

SWAT [6](Spiking Weight association Training) is a method for training spiking neural networks for classification tasks. It uses a combination of supervised and unsupervised learning to improve the performance of SNN. It is based on the idea of adjusting the weights of the connections between neurons in the network based on the input and output patterns of the network. The weights are adjusted using a learning rule that takes into account the error between the actual output of the network and the desired output.

Methodology

They propose a method which merges Bienenstock-Cooper-Munro learning rule (BCM) [9] with Spike Time Dependent plasticity Spike-timing-dependent plasticity (STDP) [8]. It yields an unimodal weight distribution where height is associated with STDP.

The BCM learning rule is based on the idea that the strength of the connections between neurons in the network should be adjusted based on the activity of the neurons and the error between the actual output of the network and the desired output. The rule specifies that the weights of the connections between neurons should be increased if the activity of the neurons is high and the error is also high, and the weights should be decreased if the activity of the neurons is low and the error is also low.

STDP 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 changes 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 $\boldsymbol{t_1} = \boldsymbol{t_2}$

Overall SWAT is based on the idea of adjusting the weights of the connections between neurons in the network based on the input and output patterns of the network. The weights are adjusted using a learning rule that takes into account the error between the actual output of the network and the desired output.

Advantages

One of the main advantages of SWAT is that it can be used to train spiking neural networks for classification tasks. This is important because spiking neural networks are a promising type of artificial neural network that is thought to have the potential to improve the performance of artificial intelligence systems. It uses a combination of supervised learning and unsupervised learning to train the network. This allows the network to learn from labelled examples as well as identify patterns and features in the data on its own. Since it is based on the simple learning rules of adjusting the weights of the connections between neurons in the network based on the input and output patterns of the network it is relatively easy to implement and understand. Overall it has the potential to enable the use of SNN in a wide range of applications.

Disadvantages

Sometimes this may become computationally intensive, especially for large networks with many neurons and connections. This could make it difficult to use SWAT on systems with limited computational resources. Another potential disadvantage is that it is based on supervised and unsupervised learning, in which supervised learning requires a large amount of labelled data to train the network effectively. This may be difficult to obtain in some cases. It also relies on the initial weights of the connections between neurons in the network, and these weights may have a significant impact on the performance of the network. This means that SWAT may be sensitive to initialization and may require careful tuning of the initial weights to achieve good performance.

2.3.2 SpikeProp: backpropagation for networks of spiking neurons.

Introduction

Spiking neurons may be more biologically realistic and efficient than other types of artificial neurons, and they may be more suitable for certain tasks such as pattern recognition and natural language processing. However, they also note that training networks of spiking neurons can be difficult due to the non-differentiable nature of the spiking function. To address this problem, the authors propose the use of SpikeProp [4], which is a method for training networks of spiking neurons using backpropagation.

Methodology

It is a method for training networks of spiking neurons using backpropagation. The spiking function, which describes the relationship between the input and the output of a spiking neuron, is not differentiable. This means that traditional methods for training artificial neural networks, such as backpropagation, cannot be directly applied to networks of spiking neurons.This can be avoided by surrogate gradient function that approximates the gradient of the spiking function, which allows the use of backpropagation to train networks of spiking neurons.

Advantages

It allows the use of backpropagation to train networks of spiking neurons. This makes it possible to leverage the simplicity and efficiency of backpropagation for training spiking neural networks. With spikeprop, it is possible to create artificial neural networks that are more biologically realistic and may be more suitable for certain tasks. It is also more efficient in training SNN than other techniques. It makes it effective in applications like image recognition and language translation.

Disadvantages

It is not effective in all applications of spiking neural networks. Also, due to the approximations used in surrogate gradient function, it may not be always accurate which may affect its performance.

Chapter 3

Methodology

Construction of associative memory using SNN [10] in this method consist of four phases

1. Initialization: Initialization of SNN and the input spiking signals
2. Structure formation: New connections with neighbouring neurons are formed
3. Parameter training: Optimize weight of synapse based on STDP
4. Pruning: Removing unnecessary connections 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 is 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 figure 3.1 shows the steps involved.

Four convolutional kernels of size 4×4 shown in figure 3.2 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 the image to 12×12. The conversion layer converts these values into spiking encoding. Pixel value

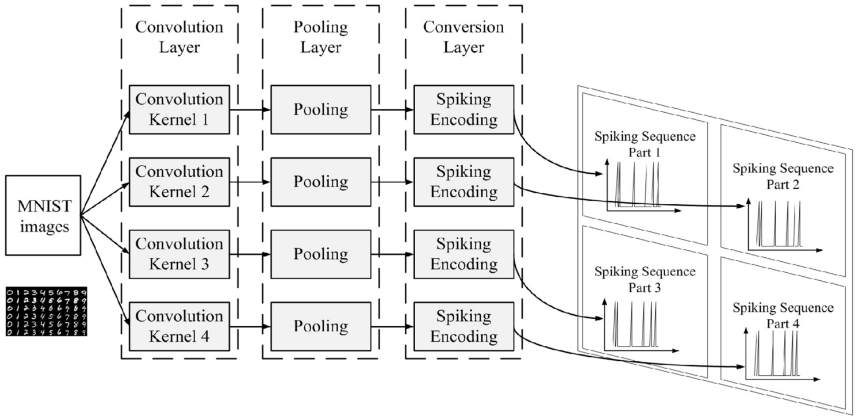

Figure 3.1: Data preprocessing

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

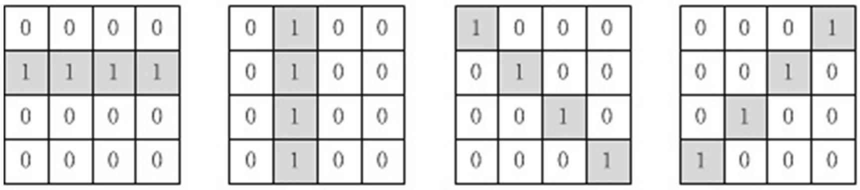

Figure 3.2: Kernels used

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 figure 3.3. 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.

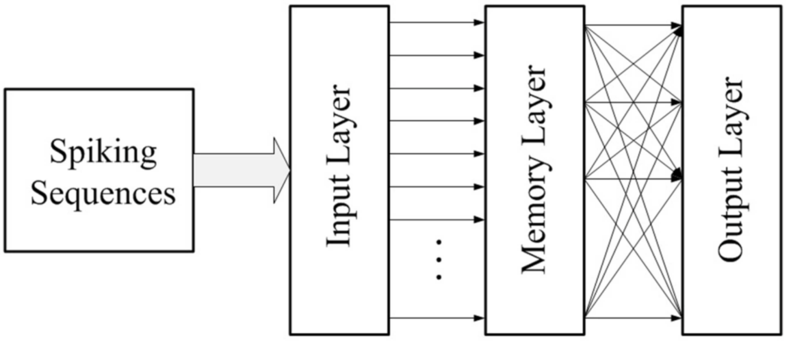

Figure 3.3: Structure of network

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 the 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 figure 3.4. 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 the the corresponding neuron in the memory layer is calculated as

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

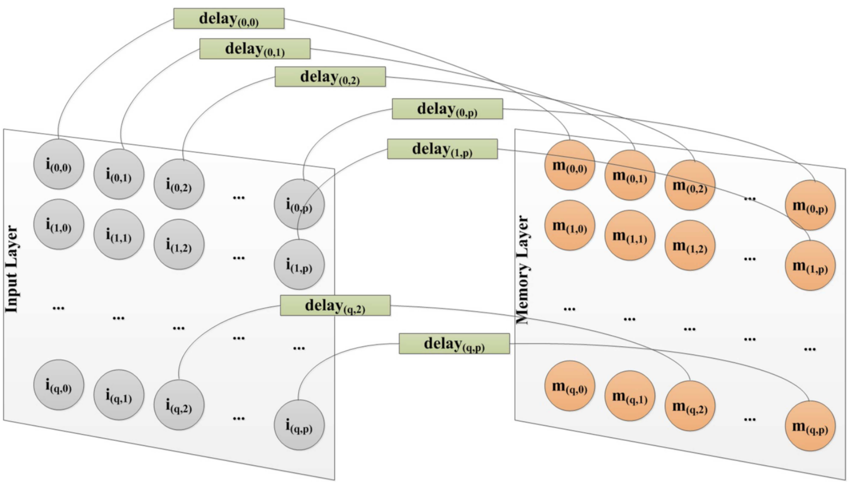

Figure 3.4: Delay in neuron connections

3.2 Structure formation

During the structure formation phase are fed into the network. The behaviour 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. It 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 needs 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 j is a neuron in S_O and the weight of the 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. Figure 3.5 shows the firing behaviour 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.

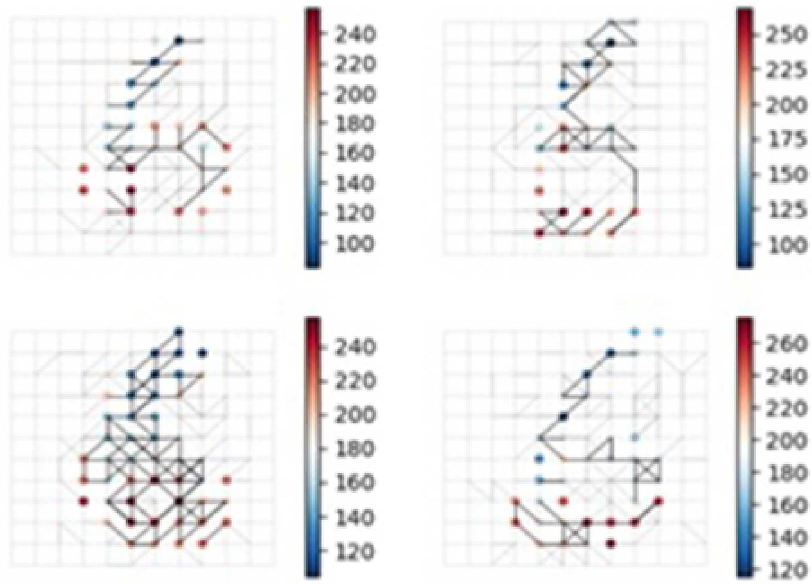

Figure 3.5: Recall response for number 6

3.4 Pruning

This phase helps in improving the efficiency of the network. In this phase, if the weight of a connection is less than the threshold (here 3), the connection will be removed. If there is a neuron in the memory layer which does not have any connections to the output layer, the connection from the 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 an 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 neuromorphic hardware which improves the efficiency of different operation involved in the processing of information using an SNN. The hardware Lohi [11] and software for that Lava [12] are developed by intel for this application. Software like NEST [13] and SpikeTorch [14] is open source tool for the SNN. As more and more research is done in this field, it could be the next-generation machine learning algorithm.

References

[1] 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.

[2] McClelland, J. L., & Rumelhart, D. E. (1981). *An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. Psychological Review*, 88(3), 375-407.

[3] 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

[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. *Hebb: 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

[8] Shouval, H. Z., Wang, S. S., & Wittenberg, G. M. (2010). *Spike timing dependent plasticity: a consequence of more fundamental learning rules.* Frontiers in computational neuroscience, 4, 19. https://doi.org/10.3389/fncom.2010.00019

[9] Bienenstock, E. L., Cooper, L. N., & Munro, P. W. (1982). *Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex.* The Journal of neuroscience : the official journal of the Society for Neuroscience, 2(1), 32–48. https://doi.org/10.1523/JNEUROSCI.02-01-00032.1982

[10] 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.

[11] Intel Corporation. (n.d.). LOHI: Low Overhead High Throughput Inference. Retrieved from `https://www.intel.com/content/www/us/en/newsroom/news/intel-unveils-neuromorphic-loihi-2-lava-software.html`

[12] Intel Corporation. (n.d.). Lava: Low Overhead Accelerator for Vision Algorithms. Retrieved from `https://www.intel.com/content/www/us/en/artificial-intelligence/solutions/inference/lava.html`

[13] Eppler, J. M., Helias, M., Muller, E., Diesmann, M., & Gewaltig, M.-O. (2008). NEST (NEural Simulation Tool). Scholarpedia, 3(4), 1430.

[14] Cao, Y., & Paine, T. (2020). SpikeTorch: A PyTorch Library for Spiking Neural Networks. arXiv preprint arXiv:2012.13350.

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]

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