

Multi-Image/Sequence learning

Gaurav Honnavara Manjunath
gaurav.honnnavaramanjunath@stud.fra-uas.de

Harish Palanivel
harish.palanivel@stud.fra-uas.de

Athkar Praveen Prajwal
praveen.athkar@stud.fra-uas.de

Abstract— In real-life situations, the prediction and perception of temporal sequences for sensory inputs are helpful/critical. Based on multiple known features of neurons, a theoretical framework known as HTM (hierarchical temporal memory) is used for sequence learning. The HTM model replicates the neocortex's operating principles which handle learning of sequences and storing the trained data in the memory and can perform prediction Operations until the right match. This paper evaluates the HTM Prediction Engine when applied to a sequence of Numbers, Sequence of Alphabets (Anti – Cancer Peptide Cells) and Set of Images to predict the sequences of numbers, Sequence of Alphabets, and Images. The primary objective is to examine the HTM prediction Engine and understand Multi sequence Learning for Sequence of Numbers, upon which, A prediction algorithm is developed for a Sequence of Alphabets for predicting a set of alphabets that belong to a particular sequence (To Predict Anti–Cancer Peptide Cells) and also to develop Multi Sequence Learning for Image Data sets can predict whether a specific Image belongs to the Training Image data Sets.

Keywords—*Hierarchical Temporal Memory (HTM), Image Prediction, Homeostatic Plasticity Controller (HPA), Prediction code, Local Area Density, Potential Radius, Local/Global Inhibition, HTM Prediction Engine).*

I. INTRODUCTION

The ability to perceive and predict temporal sequences of sensory inputs is critical for survival. Hierarchical temporal memory (HTM) sequence memory has recently been proposed as a theoretical framework for sequence learning in the cortex, based on numerous known features of cortical neurons. The model's sparse temporal codes can robustly handle branching temporal sequences by keeping numerous predictions until enough disambiguating evidence is available.

The medical sciences have advanced to provide us with a significant understanding of the working of the cortex. Investigations have concluded that many cortical regions are part of the temporal sequence processing [1] [2]. On the other hand, ML engineers have been researching sequential memory, leading to several temporal pattern recognition models [3].

Scientists have gained insights by working on the cortex that sequence learning has large invariant changing series of inputs. The exact neural mechanism of sequence memory is still unknown, but models that give a reading of the neurons are used to study. These models show significant capabilities

to recollect and recognize the sequence of inputs using rules. These ML models do not match the real-world issues

Hierarchical Temporal Memory (HTM) is a Biomimetics model based on the principles of memory predictions developed by scientists to capture the architectural and algorithmic features of the neocortex [4] [5]. HTM has given promising results in pattern recognition, and This can learn the temporal sequences and spatial flow of sensory inputs as data.

II. LITERATURE SURVEY

A. SDRs

HTM's language is Sparse Distributed representations (SDRs) of input patterns. It internally generates SDRs, given a fixed number of active bits. These bits have semantic meaning. Hence two inputs with similar semantic meaning must have equal active bits representation in SDR, which plays a significant role in HTM's learning.

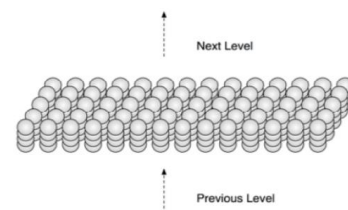


Figure.1: HTM divides artificial cells into 2,048 columns, each with 32 cells. The columns are placed in a two-dimensional array conceptually [6]

B. Proximal Dendrite Segments

A proximal dendrite connects the cells in a column, with synapses represented by small black circles. A solid circle represents a valid synaptic connection with a permanence value over the connection threshold. In contrast, a possible synapse connection with a permanence value below the connection threshold is represented by an empty circle. Feedforward input activates a column after a local inhibition step if enough valid synapses are coupled to active input bits.

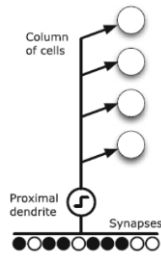


Figure.2: Proximal Dendrite Representation

C. Distal Dendrite Segments

A cell has over 130 distal dendrite segments with about 40 synapses and a single proximal dendrite segment. Nearby cells provide lateral input to the distal segments. Within an area determined by a "learning radius," the set of possible synapses connects to a subset of other cells. A dendritic segment creates connections with cells that were active together in a previous period, allowing it to remember the activation state of nearby cells. If one of its segments reencounters the same cellular activation pattern, that is, if the number of active synapses on any segment exceeds a threshold, the cell will enter a predictive state, signalling that feedforward input will likely result in column activation. Feedforward input through the proximal dendrite or lateral connections through the distal dendrite segments keeps a cell active.

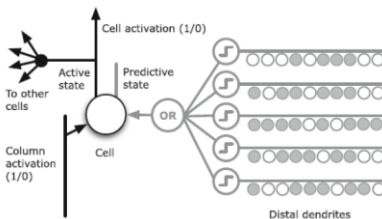


Figure.3: Distal Dendrite Segment at the cellular level

D. Neocortex

The neocortex is defined as the part of the cerebral cortex that serves mental functions for humans. It also contains billions of cells and some millions of meters. The cells are layered within which different regions are dedicated to vision, hearing, touch, movement, sensory balance, stimulus, etc.

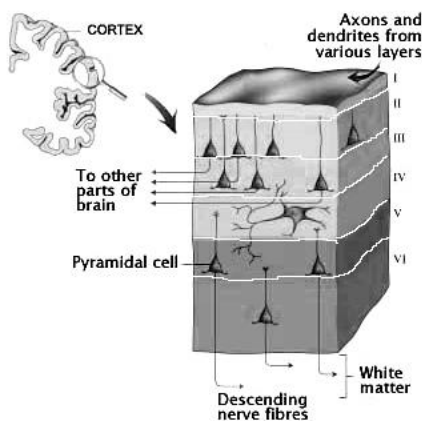


Figure.4: Neocortex Layers [6]

HTM is a working model that is inspired and designed to replicate the functionality of the biological neocortex in the

brain. Its part is to learn the input data fed as sensory input. HTM uses different approaches to replicate the neuron model until the functionality framework is defined to accept the respective sensory information [7]. The study has also confirmed that biological neurons perform more complicated functions.

E. Connection

HTM follows a different approach for the neuron model, which is inspired by cortical neurons. In contrast, the classical ANN neuron model is a weighted summation of inputs followed by a non-linear operation on the sum [7]. From advancements in neurosciences, it is confirmed that biological neurons perform much more complex functions. Communication between neurons takes place via electrical and chemical signals. These signals from the base for memory and learning within the brain.

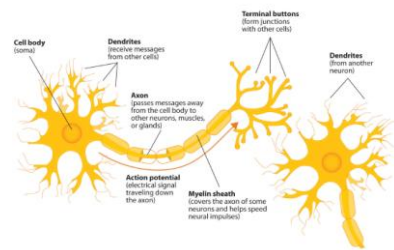


Figure.5: Neocortex Layers

The signalling process is similar: Neuron A becomes electrically charged with the surrounding fluid outside its membrane when it receives a chemical signal from another neuron. The electrical charge travels down the axon, away from A's soma [7]. A set of storage sites, known as vesicles, are located within the synapse and hold substances produced by the soma. When an electrical charge reaches the synapse, these vesicles fuse with the cell membrane of the synapse, releasing substances known as neurotransmitters into the synaptic cleft. The neurotransmitters go through the synaptic cleft to one of neuron B's dendrites, binding to receptor sites in the membrane. Neuron B generates an electrical charge, which travels down its axon and then repeats the process.

F. Memory

The cortex is not the same as parallel computers. Many computations are carried out on the input patterns in parallel computers to produce contrasting output patterns. By using this, the cortex can recover the output from its immense memory at a faster rate. These sequential patterns are stored and associated automatically with regular patterns in hierarchies [7]. These associated memories can fetch complete patterns from partial input patterns in spatial and temporal memory.

G. Prediction

Prediction is the primary function of the cortex and the foundation for intelligence [8]. The neocortex merges the invariant representation with new input data to provide a prediction about real-world life.

H. Hierarchical Temporal Memory (HTM)

The HTM model learns the procedure that occurs in one layer of the cortex. HTM works on the continuous streams of input patterns and tries to build infrequent and constant representations of input sequences based on the repeated pattern of the input stream.

HTM's ability to predict future patterns from the trained patterns of data. In a few cycles, HTM receives a unique pattern that compares the previous patterns with the new pattern. Input patterns should not repeat, and the uniqueness of the pattern is important to train different sequences of input patterns which provides a wide variety of sequences to be predicted.

III. METHODOLOGY

The project Multi sequence Learning developed using C# .Net Core in Microsoft Visual Studio 2022 IDE (Integrated Development Environment) is used as a reference model to understand the functioning of Multi sequence learning, which uses HTM Prediction Engine and HPA controller for stability.

The objective of this project is to understand Multi sequence learning for the sequence of Numbers and develop Multi sequence learning for Sequence of Alphabets and Image Datasets which includes learning of a sequence of data and images resulting in the prediction of any sequence of alphabets or input images based on the learning in HTM.

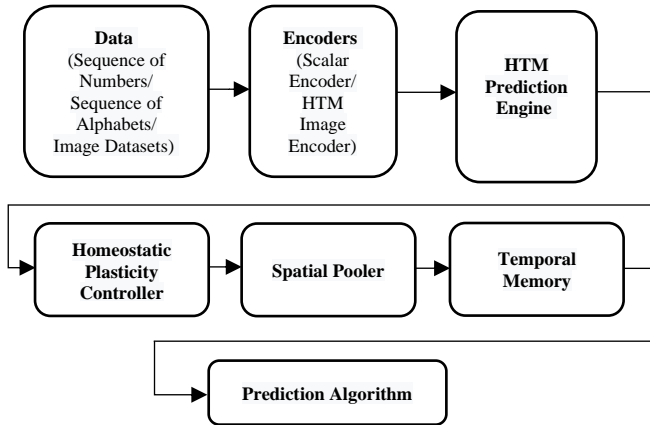


Figure.6: Overview of Multi sequence Learning

A. Datasets

This section describes the reference to the datasets which are used for Multi Sequence Learning.

- Sequence of Numbers: Includes Multiples of Numbers

Labels	Sequence
Multiples of 2	{2,4,6,8,10,12,14,16}
Multiples of 3	{3,6,9,12,15,18,21,24}
Multiples of 5	{5,10,15,20,25,30,35,40}
Multiples of 7	{7,14,21,28,35,42,49}
Multiples of 11	{11,22,33,44}

Table.1: Date Sets (Numbers) – Multi sequence Learning

- Sequence of Alphabets: Anticancer peptides Data Set

Sequence	Labels
AIGKFLHSAKKFGKAFVGEIMNS	mod. active
FAKIIAKIAKIAKKIL	inactive - exp
FAKKLAKKLKKLAKKLAKKWKL	mod. active
FAKLLAKALKKLL	very active
....

Table.2: Date Sets (Alphabets) – Multi sequence Learning [9]

- Image Datasets: Set of Fruits (Viz. Apple, Avocado, Banana)

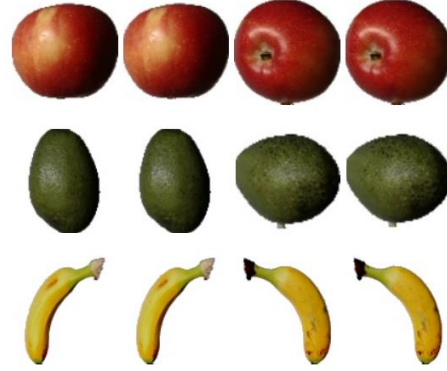


Figure.7: Datasets (Image) – Multi sequence Learning

B. Encoders

An HTM develops from the data that is given to it, as well as the presentation of that data. An encoder converts arbitrary input into a format that an HTM can understand. This input must always be an SDR. Each bit in the SDR represents the activation state ('0' or '1' for inactive and active, respectively) of the columns from the previous area in the HTM. This input is then used as the feedforward input for the HTM's next area.

The below example shows the binarization of an Image data set using HTM Image Encoder.

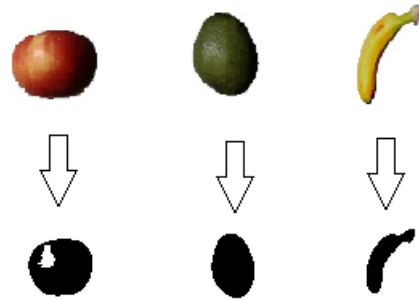


Figure.8: Image Binarization

C. Spatial Pooler

Spatial Pooler creates SDR input, during which the cells of the active columns are mapped. Each column has a network of connections with the next region of input bits via synapses. Many columns would look the same, but these columns are unique from each other. Different patterns produce different activation levels, and the more robust activation restricts lower activation of the columns. The area of columns is adjustable and can range from small regions to the entire area. The inhibitory mechanism is implemented to give a limited representation of the input. An identical pattern produces identical activated columns. HTM trains from the input and unforms connections between cells. Updating synapse permanence leads to learning. The active columns increase the persistence value with active bits while the other columns decrease it. Columns that are not active do not learn. The inactive columns are boosted to ensure that all the columns participate in the training. The spatial pooler implies pools or clusters of data in the spatial dimension. Each pattern that appears at the input during the spatial pooler's learning process is compared to the database of other patterns.

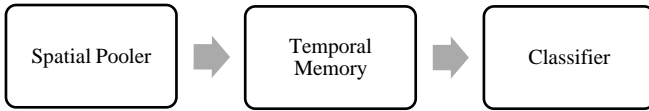


Figure. 9: HTM Algorithm Flow

D. Sparse Distributed Representation

In the HTM, SDR is an effective information organization system. Sparse means that only a tiny percentage of the big, interconnected cells are active at any given time. "Distributed" denotes that active cell are dispersed throughout the region and will be used to depict the region's activity. Because the binary representation is more biologically reasonable and highly computationally efficient, HTM considers the binary SDR converted from a specific encoder. Even though the number of possible inputs exceeds the number of possible representations, the binary SDR does not result in a functional loss of information due to the following critical features of the SDR.

The HTM methods can be set with various parameters and picking the proper parameters for investigations is crucial. The table shows a list of Spatial Pooler parameters with default values widely utilized in HTM investigations. Each of these variables influences the performance of HTM on its own; however, we will concentrate on the effect of potential radius and local area density, Global/Local Inhibition, and NumActiveColumns PerInhArea.

The initialization of numerous parameters defined by class htmconfig (HTM configuration) is the first and most crucial step in using any HTM configuration. Below is the table consisting of all the HTM parameters that affect the images' classification.

Parameters	Default value
inputBits	100
numColumns	1024
CellsPerColumn	25
GlobalInhibition	true
LocalAreaDensity	-1
NumActiveColumnsPerInhArea	$0.02 * \text{numColumns}$
PotentialRadius	$0.15 * \text{inputBits}$
MaxBoost	10
InhibitionRadius	15
DutyCyclePeriod	25
MinPctOverlapDutyCycles	0.75
MaxSynapsesPerSegment	$0.02 * \text{numColumns}$
ActivationThreshold	15
ConnectedPermanence	0.5
PermanenceDecrement	0.25
PermanenceIncrement	0.15
PredictedSegmentDecrement	0.1

Table.3: HTM Config parameters

IV. IMPLEMENTATION

This section explains how Multi Sequence Learning Experiment has been carried out. We have analyzed how multi-sequence learning for a sequence of numbers works and worked on the accuracy of the HTM prediction engine. Further, we have developed Multi Sequence learning that can train the Sequence of Alphabets (Anticancer Peptide Cells) and predict cancer sequences. Multi Sequence Learning for Image data sets includes Training Image data sets using HTM Image Encoder and Predicting Images from the input given by the user.

A. Learning Phase

The learning phase includes fetching Datasets from the solution directory and Train using a Spatial pooler using a Homeostatic Plasticity Controller for stability. Training of Different datasets for Multi Sequence Learning is explained as follows.

i. Training of Sequence of Numbers:

Training of Sequence of Numbers Includes Initialization of Datasets, including Label and the Sequence. The Sequence is then used to train the spatial pooler with HTM configuration parameters for several iterations. After several iterations, the spatial pooler enters a stable state.

The Figure below illustrates how training of Sequence of Numbers is carried out in the Multi Sequence Model.

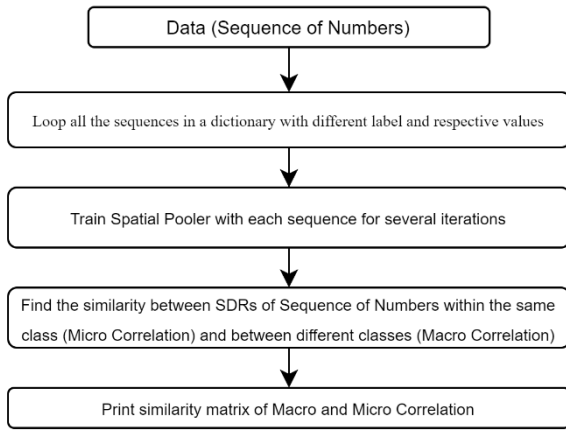


Figure. 10: Training Model – Sequence of Numbers

ii. Training of Sequence of Alphabets

Training of Sequence of Alphabets Includes fetching the Sequence of Datasets (Anticancer Peptide Datasets) from the solution directory stored as a .csv file. The Sequence includes Label and the Anticancer Peptide Sequences. The Sequence is then used to train the spatial pooler with HTM configuration parameters for several iterations. Each cancer sequence is treated as a row of a single element associated with a label which helps in classification. After several iterations, the spatial pooler enters a stable state.

The Figure below illustrates how training of Sequence of Alphabets is carried out in the Multi Sequence Model.

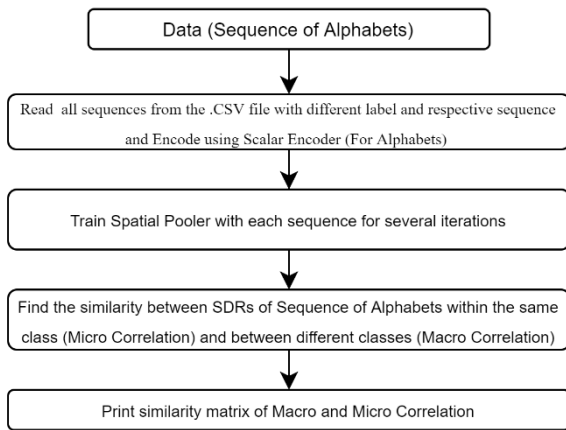


Figure. 11: Training Model – Sequence of Alphabets

iii. Training of Image Datasets

Training of Image data sets Includes fetching all the image data sets from the solution directory stored as a .jpg file. The respective folder name is considered as the label and is then used to train the spatial pooler with HTM configuration parameters for several iterations. After several iterations, the spatial pooler enters a stable state.

The Figure below illustrates how training of Image Datasets is carried out in the Multi Sequence Model.

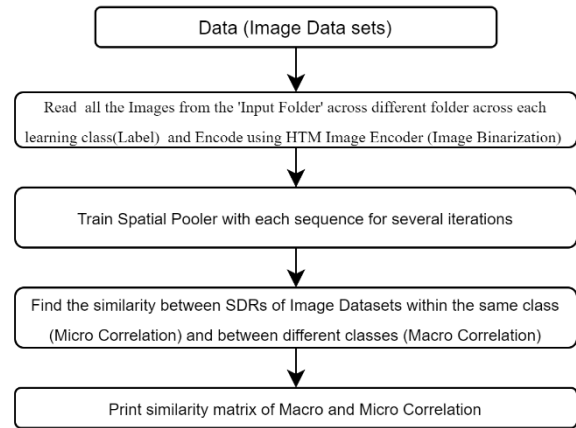


Figure. 12: Training Model – Image Datasets

We used Images of dimensions 40x40 pixels for Training with default parameters in htmconfig. The Training runs for several iterations until the spatial pooler enters a stable state controlled by the HomeostaticPlasticityController (HPC) class. The goal is to place the Spatial Pooler in a newborn state at the start of the learning process. The boosting is very active, but the spatial pooler is unstable. Once the SDR generated for each input becomes stable, the HPC will fire an event that notifies the code that the spatial pooler is stable.

B. Cancer Sequence Classification

The experiment uses Anticancer peptides cells represented using a sequence of alphabets. All the alphabets are treated as cells of the sequence. We use HTM to train multiple sequences. Each cancer sequence is treated as a row of a single element associated with a label which helps in classification later.

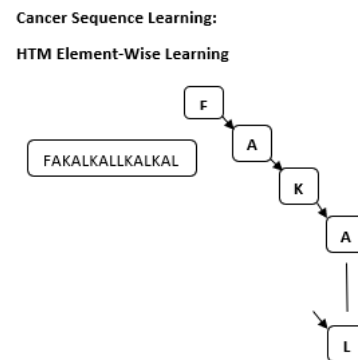


Figure 13: Cancer Sequence Classification

C. Prediction

After the learning/training phase, the training model generates the similarity matrix for all the classes. The SDRs computed for numbers, alphabets, and images compare with the SDRs of the respective Sequence learned during the

training phase and compute accuracy based total number of matches and sequence count. Further, the input sequence (Number, Alphabets, Images) is computed as SDR input and compared with each of the SDRs of the Sequence learned during the training phase; the best match is searched in the correlation matrix and based on the accuracy and observation class (Label), the Sequence is predicted.

V. RESULTS

In this project, we have used sequence of Numbers for Multi sequence learning of Numbers, Sequence of alphabets (Anticancer Peptide Sequence) for Multi Sequence learning of Alphabets and Image data sets for Muti Sequence learning of Images. The experiment runs individually when the user selects the respective options.

Figure 14 shows the flow chart for Multi sequence learning for the experiment carried out.

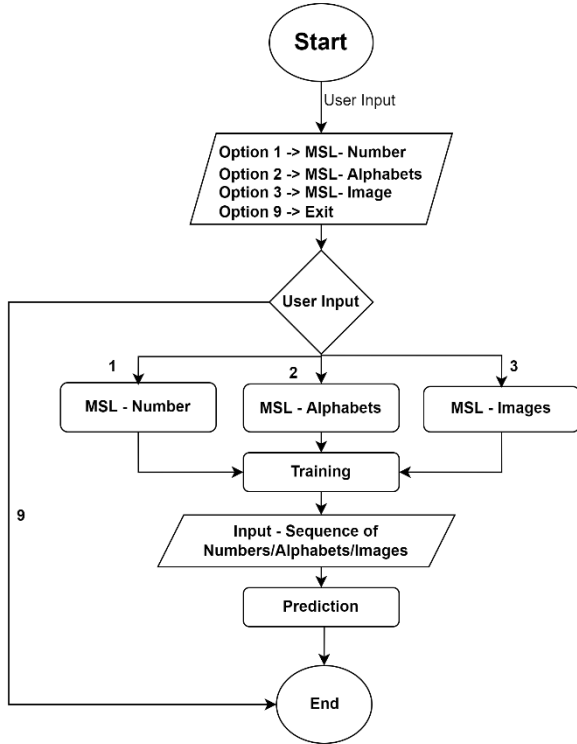


Figure 14: Flow chart for Multi Sequence Learning Experiment

From all the experiments carried out in the training phase and prediction phase, the similarities between sequences of the same class different classes have explained our findings in the below-given cases:

A. Multi Sequence Learning – Sequence of Numbers

In Multi Sequence learning for a sequence of numbers, the sequence is defined as shown in Table.1 and trained with HTM parameters shown in Table.4 below. The HTM uses Scalar Encoder for encoding, and the Spatial Pooler creates SDR input, during which the cells of the active columns are

mapped. HTM trains from the input and unforms connections between cells. The spatial pooler implies pools or clusters of data in the spatial dimension. Each pattern that appears at the input during the spatial pooler's learning process is compared to the database of other patterns.

Parameters	value
inputBits	100
numColumns	1024
CellsPerColumn	25
GlobalInhibition	true
LocalAreaDensity	-1
NumActiveColumnsPerInhArea	$0.02 * \text{numColumns}$
PotentialRadius	$0.15 * \text{inputBits}$
MaxBoost	10
InhibitionRadius	15
DutyCyclePeriod	25
MinPctOverlapDutyCycles	0.75
MaxSynapsesPerSegment	$0.02 * \text{numColumns}$
ActivationThreshold	15
ConnectedPermanence	0.5
PermanenceDecrement	0.25
PermanenceIncrement	0.15
PredictedSegmentDecrement	0.1

Table 4: HTM Config

Figure 15. Shows the training accuracy for a sequence of numbers for five sequences, Table 5 shows the sequence which was trained during the experiment.

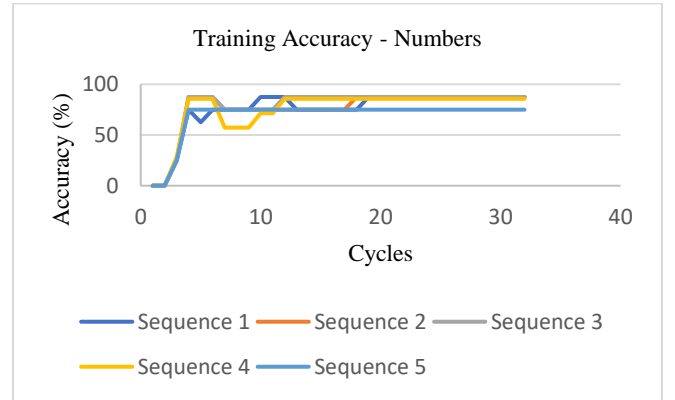


Figure 15: Training Accuracy – Sequence of Numbers

Figure 16 below shows the prediction for the sequence of Numbers for the trained data sequence as in Table 1.

```

8
9
11
13
17
19
1
Entered Number are :
2 5 8 3 9 11 13 17 19 1
-----Start of PredictNextElement Function-----
Predicted Sequence: TwoMultiple, predicted next element 8
Predicted Sequence: ThreeMultiple, predicted next element 15
Invalid Match.....
Predicted Sequence: ThreeMultiple, predicted next element 12
Invalid Match.....
Predicted Sequence: ElevenMultiple, predicted next element 1.0
Invalid Match.....
Invalid Match.....
Invalid Match.....
Invalid Match.....
-----End of PredictNextElement -----
F:\Frankfurt University Of Applied Sciences\Software Engineering(VS)\Final Project\source\MySEProject\src\Main.java:24584: exited with code 0.

```

Figure 16: Prediction – Sequence of Numbers

B. Multi Sequence Learning – Sequence of Alphabets

In Multi Sequence learning for a sequence of Alphabets (Anticancer Peptide Sequences), the sequence is defined [9] and trained with HTM parameters shown in Table.5 below. Anticancer peptides cells are represented using a sequence of alphabets. All the alphabets are treated as cells of the sequence. We use HTM to train multiple sequences, each cancer sequence is treated as a row of a single element associated with a label which helps to classify.

Parameters	value
inputBits	31
numColumns	1024
CellsPerColumn	32
GlobalInhibition	true
LocalAreaDensity	-1
NumActiveColumnsPerInhArea	0.02 * numColumns
PotentialRadius	65
MaxBoost	10
InhibitionRadius	15
DutyCyclePeriod	25
MinPctOverlapDutyCycles	0.75
MaxSynapsesPerSegment	128
ActivationThreshold	15
ConnectedPermanence	0.5
PermanenceDecrement	0.25
PermanenceIncrement	0.15
PredictedSegmentDecrement	0.1

Table 5: HTM Config (Alphabets)

Figure 17. Shows the training accuracy for a sequence of alphabets (Anticancer Peptide Sequence).

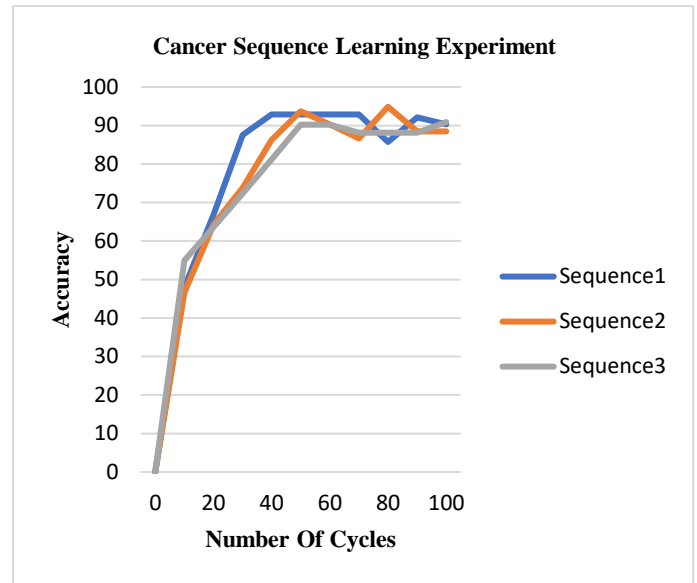


Figure 17: Training Accuracy – Sequence of Alphabets

Figure 18 below shows the prediction for a particular sequence that is entered by the user.

```

F:\Frankfurt University Of Applied Sciences\Software Engineering(VS)\Final Project\source\MySEProject\src\Main.java:24584: exited with code 0.
----- D,inactive - virtual -----
----- P,inactive - virtual -----
----- Q,inactive - virtual -----
----- K,inactive - virtual -----
----- A,inactive - virtual -----
----- D,272,inactive - virtual -----
----- A,273,inactive - virtual -----
----- V,inactive - virtual -----
----- A,275,inactive - virtual -----
----- V,276,inactive - virtual -----
----- N,inactive - virtual -----
----- D,inactive - virtual -----
----- P,inactive - virtual -----
----- Q,inactive - virtual -----
----- K,inactive - virtual -----
----- R,inactive - virtual -----
----- A,inactive - virtual -----
Ready to Predict.....
Enter Cancer Sequence: *note format->AAAAVVV {AlphabeticSequence}
NDPQKRA
Enter Next Sequence :
GLFDVIAKVASVTKKL
Predicted Class : mod. active votes: 6
Predicted Class : inactive - exp votes: 8
Enter Next Sequence :
GLFDIVKKVWGAFGSL
Enter Next Sequence :
GLFDIVKKIAGHIVSST
Enter Next Sequence :
GLFDIVKKIAGHIASSI
Enter Next Sequence :
KWKSFAKTFKSAKKTVLHTALKAISS
Predicted Class : mod. active votes: 10
Predicted Class : very active votes: 2
Predicted Class : inactive - exp votes: 2
Enter Next Sequence :
KWKLFKKIKFLHSAKKF
Predicted Class : mod. active votes: 10
Predicted Class : very active votes: 2
Predicted Class : inactive - exp votes: 2
Enter Next Sequence :
DPTKLNHVAAGL
Enter Next Sequence :
KWKLFKKIGIKFKLAKKF
Predicted Class : mod. active votes: 10
Predicted Class : very active votes: 2
Predicted Class : inactive - exp votes: 2
Enter Next Sequence :

```

Figure 18: Prediction – Sequence of Alphabets

C. Multi Sequence Learning – Image Data Sets

In Multi Sequence learning for Image Datasets, they are defined in the "Input Folder" in the solution directory, and image data sets are used as shown in *Figure .7* with HTM parameters shown in *Table.6* below. The HTM uses HTM image Encoder for encoding. The images are binarized and encoded, stored as array elements of zeros and ones, and are computed as SDR input, during which the cells of the active columns are mapped. HTM trains from the input and unforms connections between cells. The spatial pooler implies pools or clusters of data in the spatial dimension. Each pattern that appears at the input during the spatial pooler's learning process is compared to the database of other patterns.

Parameters	value
Image height	40
Image Width	40
inputBits	Image height*Width
numColumns	1024
CellsPerColumn	25
GlobalInhibition	true
LocalAreaDensity	-1
NumActiveColumnsPerInhArea	0.02 * numColumns
PotentialRadius	0.15 * inputBits
MaxBoost	10
InhibitionRadius	15
DutyCyclePeriod	25
MinPctOverlapDutyCycles	0.75
MaxSynapsesPerSegment	128
ActivationThreshold	15
ConnectedPermanence	0.5
PermanenceDecrement	0.25
PermanenceIncrement	0.15
PredictedSegmentDecrement	0.1

Table 6: HTM Config (Image Datasets)

Figure 19. Shows the training accuracy for Image Data Sets.

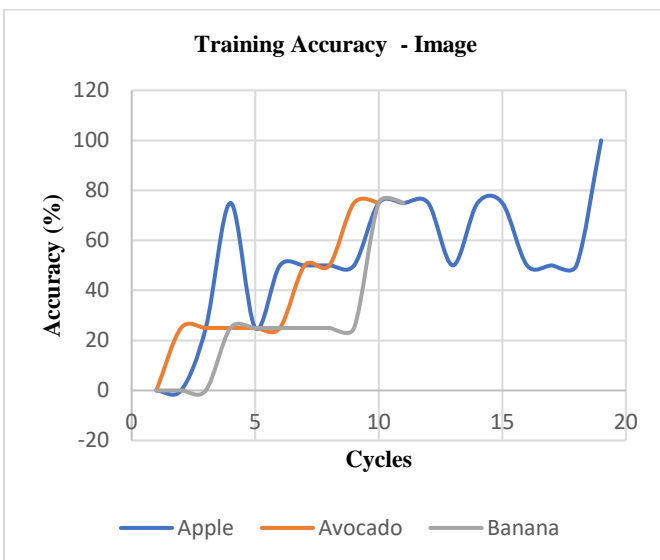


Figure 19. Training Accuracy – Image datasets

Figure 20 shows the prediction for the input Images for the trained Image data as in *Figure.7*.

```

F:\Frankfurt University Of Applied Sciences\Software Engineering(VS)\Final Project\source\
"F:\Frankfurt University Of Applied Sciences\Software Engineering(VS)

Predicted Sequence: Apple

Input Image to Predict :
"\SimpleMultiSequenceLearning\Testing Files\Avocado_1.jpg"
Predicted Sequence: Avocado

Input Image to Predict :
"\SimpleMultiSequenceLearning\Testing Files\Banana_3.jpg"
Predicted Sequence: Banana

Input Image to Predict :
"\SimpleMultiSequenceLearning\Testing Files\Apple_1.jpg"
Predicted Sequence: Apple

Input Image to Predict :
"\SimpleMultiSequenceLearning\Testing Files\Banana_4.jpg"
Predicted Sequence: Banana

Input Image to Predict :
"\SimpleMultiSequenceLearning\Testing Files\Cauliflower.jpg"
Invalid Match....

Input Image to Predict :
"\SimpleMultiSequenceLearning\Testing Files\Ginger_root.jpg"
Invalid Match....

Input Image to Predict :

```

Figure 20. Prediction – Image Data Sets

VI. CONCLUSION

Multi Sequence learning for Sequence of Numbers which uses Neocortex API is used as a reference model to develop a solution for Multi Sequence learning - Sequence of Alphabets and Multi Sequence learning- Image data sets. HTM Prediction Engine was modified with different parameters to match the respective training process. The Sequence of Alphabets (Anticancer Peptide Sequence) Stored as a CSV file was modified and stored as an encoded value in the dictionary using Scalar Encoder and SDR input for the Training process. A prediction algorithm was developed to predict the trained sequences where the similarity matrix generated is compared with each of the SDRs of the Sequence learned during the training phase and based on the accuracy and observation class (Label), the Sequence is predicted.

HTM Image Encoder was incorporated to develop a solution that could train multiple Image data sets and a prediction algorithm that could predict input images. The HTM Image Encoder binarizes the input image and stores as array elements of zeros and ones used as SDR Input for training. Similar to the Prediction algorithm for Sequence of Numbers and Alphabets, the Prediction of Image algorithm was developed, and the input image was predicted by comparing with the trained data sets and returning the prediction output based on accuracy and Observation class (Label).

We performed Multi Sequence Learning for a different sequence of data sets and could achieve up to 87.5% of accuracy in the Training Phase.

The experiments carried out helped us understand different types of encoders, such as scalar encoders and HTM Image encoders, how the Spatial pooler creates SDR inputs and computes the learning phase, and how the Homeostatic Plasticity controller helps in stabilizing the learning phase in NeoCortex API.

VII. REFERENCES

- [1] Keele, "“Sequence learning,” - B. A. C. G. J. D. S. W.," 1998. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/21227209..>
- [2] "D. V. B. Michael D Mauk, “Sequence learning,”," 2004. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/15217335>.
- [3] "B. H. J. L. R. Rabiner, “An introduction to hidden markov models,”," 1986. [Online]. Available: http://ai.stanford.edu/~pabbeel/depth_qual/Rabiner_Juang_hmms.pdf.
- [4] Hawkins, "J. C. S. A. J. , “Continuous online sequence learning with an unsupervised neural network model,”," 2016. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/27626963>.
- [5] S. A. J. Hawkins, "“Why neurons have thousands of synapses, a theory of sequence memory in neocortex,”," 2016. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fncir.2016.00023/full>.
- [6] "Neocortexlayersimageref," [Online]. Available: <http://brainmind.com/neocortex.html>.
- [7] K. J. Hole, "The htm learning algorithm," 2016. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-30070-2_11.
- [8] S. A. J. Hawkins, "Why neurons have thousands of synapses, a theory of sequence memory in neocortex," 2016. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fncir.2016.00023/full>.
- [9] M. L. Repository, "AntiCancer Peptides Data sets," 2019. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Anticancer+peptides..>
- [10] B. N. a. I. N. S. G. Eason, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," London, Phil. Trans. Roy. Soc, pp. vol. A247, pp. 529–551.
- [11] J. C. Maxwell, "A Treatise on Electricity and Magnetism," Oxford: Clarendon, 1892, pp. 3rd ed., vol. 2. pp.68–73.
- [12] I. S. J. a. C. P. Bean, "Fine particles, thin films and exchange anisotropy in Magnetism," G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. vol. III, 271–350.