Approve Prediction of Multisequence Learning

Faiz Mohammad Khan
faiz.khan@stud.fra-uas.de

Shiva Kumar Biru
shiva.biru@stud.fra-uas.de

Mohan Sai Ram Sarnala
mohan.sarnala@stud.fra-uas.de

***Abstract*— this paper focuses on Multisequence Learning, which is a technique for learning and predicting sequences. The existing implementation of Multisequence Learning is examined to understand how sequences are learned and predicted. The paper then proposes a new method that improves upon the existing implementation. The new method automates the process of reading learning sequences from a file and testing subsequence's from another file to calculate the percentage prediction accuracy which is then stored in a result file. This makes the process more efficient and less error-prone than manually inputting sequences. The method can be applied to a variety of industrial solutions, such as recognizing songs and classifying cancer peptides.**

The paper highlights the importance of sequence learning and prediction in various industries and demonstrates the effectiveness of the proposed method in accurately predicting sequences. The results show that the proposed method can be used to improve various applications that involve sequence learning and prediction. Overall, the paper provides insights into Multisequence Learning and presents a new method that can be used to improve the efficiency of sequence learning and prediction and stores the prediction accuracy in the file.

Keywords—Hierarchical Temporal Memory (HTM), Prediction code, Local Area Density, Potential Radius, Local/Global Inhibition, HTM Prediction Engine).

I. INTRODUCTION

Survival requires the ability to comprehend and predict temporal sequences of sensory inputs. Based on multiple known properties of cortical neurons, hierarchical temporal memory (HTM) sequence memory has recently been proposed as a theoretical framework for sequence learning in the cortex. The sparse temporal codes of the model can handle branching temporal sequences effectively by retaining several predictions until enough disambiguating evidence is present. [1] [2]

The medical sciences have improved to the point that we have a good grasp of how the cortex works. According to research, multiple cortical areas are involved in temporal sequence processing.ML engineers, on the other hand, have been exploring sequential memory, which has resulted in various temporal pattern recognition models.

Working on the cortex, scientists discovered that sequence learning has a huge invariant changing series of inputs. The precise neurological process of sequence memory is yet understood, however models that provide a reading of the neurons are being studied. These models demonstrate great ability to remember and recognize the sequence of inputs utilizing rules. These ML models do not correspond to real-world challenges.

Hierarchical Temporal Memory (HTM) is a Biomimetic model that was built by scientists to replicate the architectural and algorithmic elements of the neocortex. HTM has demonstrated promising pattern recognition results, and it can learn the temporal sequences and spatial flow of sensory inputs as data. [3]

II. LITERATURE SURVEY

A. SDRs

In HTM, Sparse Distributed representations (SDRs) are used to represent input patterns. The algorithm generates these SDRs by selecting a fixed number of active bits from Fig 1, [4]each of which carries a semantic meaning. As a result, inputs with similar semantic meaning will have equivalent SDRs with the same number of active bits, which is crucial for the learning process in HTM.

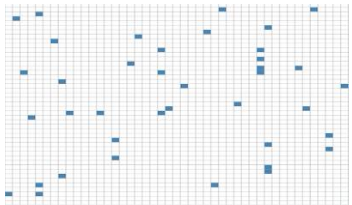

Figure.1: represents a sparse distributed representation

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. From Fig 2. [5]

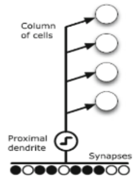

Figure.2: Proximal Dendrite Representation

C. Distal Dendrite Segments

A single cell in the brain typically has more than 130 distal dendrite segments, each of which contains approximately 40 synapses. In addition, it has a solitary proximal dendrite segment. Nearby cells provide lateral input to the distal segments, and within a specific "learning radius," a set of possible synapses connect to a subset of other cells. The dendritic segment forms connections with cells that were previously active together, allowing it to remember the activation state of neighbouring cells. If a dendritic segment recognizes the same cellular activation pattern, i.e., if the number of active synapses on any segment exceeds a particular threshold, the cell enters a predictive state, indicating that feedforward input is likely to result in column activation. Feedforward input through the proximal dendrite or lateral connections through the distal dendrite segments maintains cell activity.

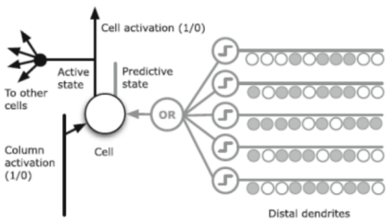

Figure.3: Distal Dendrite Segment at the cellular level

D. Neocortex

The neocortex, a component of the cerebral cortex, is responsible for carrying out various mental functions in humans. With billions of cells and millions of meters, it is a highly complex and intricate structure. The cells are arranged in layers, with specific regions dedicated to different functions such as vision, hearing, touch, movement, sensory balance.

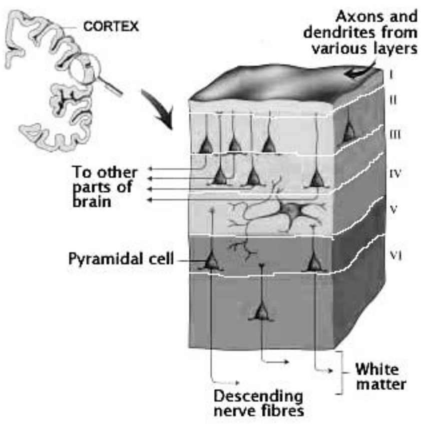

Figure.4: Neocortex Layers

HTM is a functioning model that is created with the inspiration of replicating the functionality of the neocortex in the human brain. Its primary purpose is to learn the input data that is fed as sensory input. To replicate the neuron model accurately, HTM employs various techniques until the functional framework is defined to accept the relevant sensory information. However, it has been confirmed through research that biological neurons perform more intricate functions than those performed by HTM.

E. Connection

The HTM neuron model is inspired by cortical neurons and differs from the classical ANN neuron model From Fig 5, [6] which involves a weighted summation of inputs and a subsequent non-linear operation on the sum. Recent developments in neuroscience indicate that biological neurons perform more complex functions and rely on both electrical and chemical signals for communication, which form the basis of memory and learning in the brain.

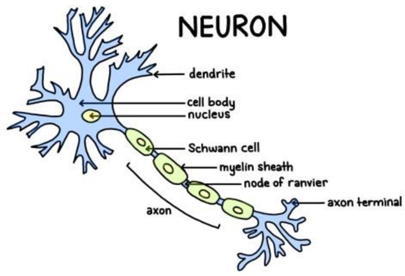

Figure.5

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. 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

Parallel computers and the cortex are not identical. While parallel computers perform numerous computations on input patterns concurrently to generate distinct output patterns, the cortex utilises this approach to retrieve output from its vast memory more swiftly. The cortex automatically stores and links sequential patterns with regular patterns in a hierarchical manner. These linked memories can retrieve complete patterns from partial input patterns in both spatial and temporal memory.

G. Prediction

The neocortex is primarily responsible for prediction, which forms the basis for intelligence. It combines unchanging representations with fresh input information to generate predictions about the actual world. [7]

H. Hierarchical Temporal Memory (HTM)

The HTM model focuses on learning the process that takes place in a single layer of the cortex and operates on uninterrupted streams of input patterns to generate rare and consistent representations of input sequences based on the recurring patterns of the input stream. Additionally, HTM has the capability to anticipate future patterns by utilising the trained data patterns. In a few iterations, HTM analyses a distinct pattern and compares it to the preceding ones. To enable the training of diverse input pattern sequences that can be predicted, it is crucial that the input patterns are unique and not repetitive.

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.

The objective of this project is to understand Multi sequence learning for the sequence of Numbers and develop a new method automates the process of reading learning sequences from a file and testing subsequence's from another file to calculate the percentage prediction accuracy based on the learning in HTM.

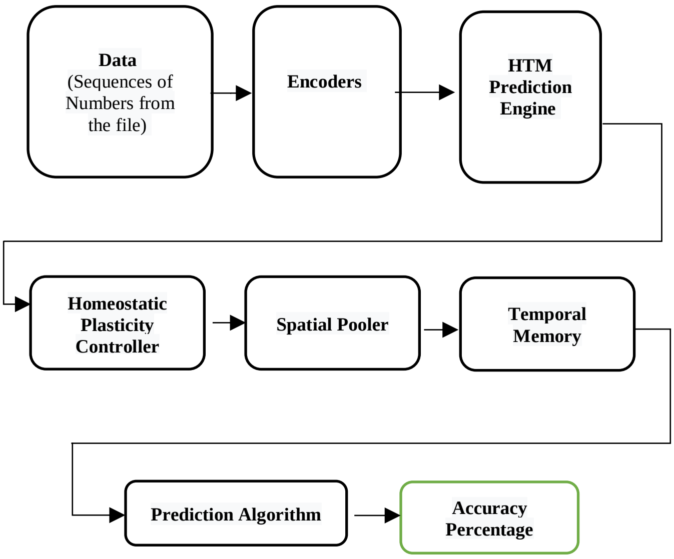

Figure.6: Overview of Multi sequence Learning

A. Datasets

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

Multiple Sequence of Numbers in csv-file:

seq1: 2, 4,6,8,10,12,14,16,18
seq2: 4, 8, 12, 16,20,5,3,2,1
seq3: 1, 2,3,4,5,6,7,8,9,10
seq4: 1,2,3,7,8,4,5,6,8,9
.
.
.

Seqn: 3,5,7,3,2,1,6,7,8,9

B. Encoders

The development of an HTM is dependent on the data it receives, as well as how that data is presented. To enable an HTM to interpret the input, an encoder is used to convert arbitrary input into a format that the HTM can understand. This format always consists of a Sparse Distributed Representation (SDR), where each bit represents the activation state of columns in the previous area of the HTM. The SDR is then utilized as the feedforward input for the next area of the HTM.

C. Spatial Pooler

The Spatial Pooler is responsible for generating a Sparse Distributed Representation (SDR) input by mapping active cells to columns. Each column has connections with the next region of input bits through synapses, and although many columns may appear similar, they are unique from one another as shown in Fig 7, [8] Different patterns produce varying levels of activation, and the more robust activation suppresses lower activation levels in the columns. The area of columns can be adjusted to cover small or large regions. An inhibitory mechanism is implemented to limit the representation of input. The HTM trains from the input and forms connections between cells. Updating synapse permanence is a form of learning. The persistence value of active columns is increased while that of inactive columns is decreased. Inactive columns do not learn, and they are boosted to ensure they participate in training. The Spatial Pooler groups or clusters data in the spatial dimension, and each pattern presented during learning is compared to the database of other patterns.

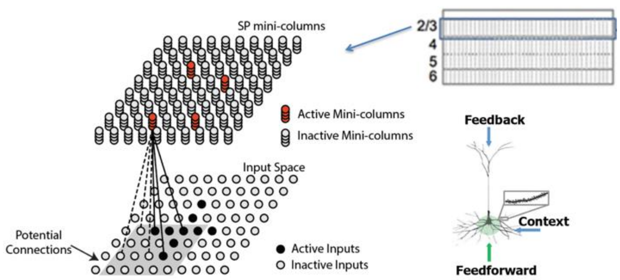

Figure. 7:Representation of spatial pooler

D. Sparse Distributed Representation

In HTM, Sparse Distributed Representation (SDR) is an effective system for organizing information. The term "sparse" refers to the fact that only a small percentage of the large, interconnected cells are active at any given time. The term "distributed" implies that the active cells are spread out across the region and are used to represent the region's activity. HTM uses a binary SDR, which is more biologically plausible and computationally efficient, and is obtained 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 loss of functional information due to the critical features of the SDR.

In HTM, it is important to select appropriate parameters for the various methods, and the table provides a list of Spatial Pooler parameters with default values that are commonly used in HTM studies. Each of these parameters has a separate impact on the performance of HTM. However, we will focus on the effects of certain parameters, such as potential radius and local area density, global/local inhibition, and the number of active columns per area.

The initial step in utilizing any HTM configuration is to define numerous parameters using the htmconfig class. This is a critical step in the process. The table below lists all the HTM parameters that affect image classification

Parameters	Default value
inputBits	100
numColumns	1024
CellsPerColumn	25
GlobalInhibition	true
LocalAreaDensity	-1
NumActiveColumnsPerInhArea	0.02 * numColumns
PotentialRadius	0.15 * inputBits
MaxBoost	10.0
InhibitionRadius	15
DutyCyclePeriod	25
MinPctOverlapDutyCycles	0.75
MaxSynapsesPerSegment	0.02 * numColumns
Activation Threshold	15
Connected Permanence	0.5
Permanence Decrement	0.25
Permanence Increment	0.15
Predicted Segment Decrement	0.1

Table.1: 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 automates the process of reading learning sequences from a file and testing subsequence’s from another file to calculate the percentage prediction accuracy, then store the results in a result file at the end of the program.

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 datasets for Multi Sequence Learning is explained as follows.

Training of Multi-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.

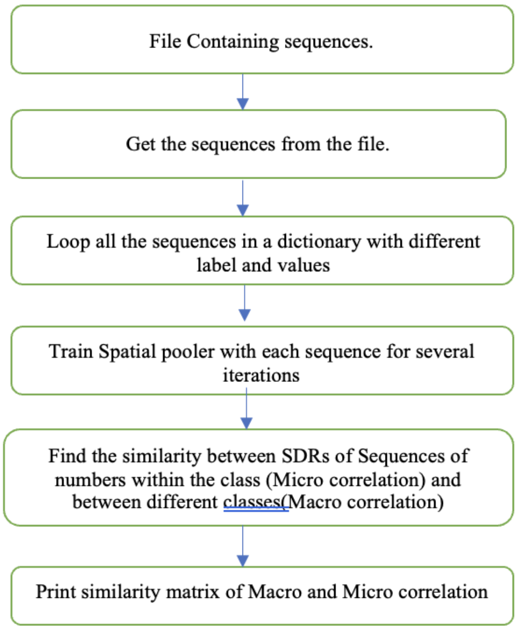

Figure. 8: Training Model – Sequences of Numbers

B. Prediction Method

Once the learning/training phase is complete, the model produces a similarity matrix for all the classes. The SDRs computed for the input numbers are then compared with the SDRs of the corresponding sequence learned during training to calculate accuracy based on the total number of matches and sequence count. Additionally, the input sequence is converted to an SDR and compared with each of the SDRs of the learned sequences during training. The correlation matrix is then used to search for the best match, and the predicted sequence is assigned an observation class (label) based on accuracy and classification and prediction Engine shows the percentage accuracy of sequences which it belongs to.

V. RESULTS

In this project, we have used sequence of Numbers for Multi sequence learning of Numbers Sequence for Multi Sequence

Figure 9: shows the flow chart for Multi sequence learning for the experiment carried out.

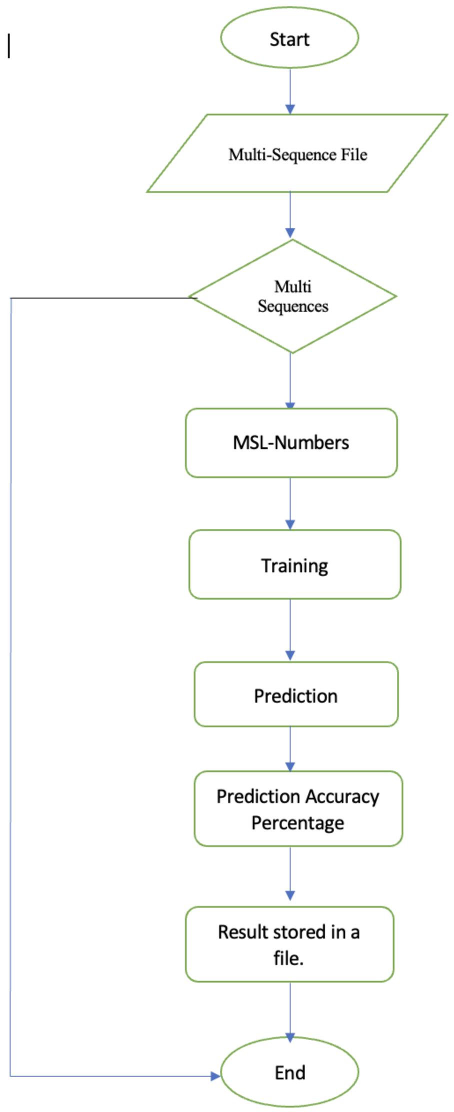

Figure 9: Flow chart for Multi Sequence Learning Experiment

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 trained with HTM parameters shown in *Table.2* 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 * numColumns
PotentialRadius	0.15 * inputBits
MaxBoost	10
InhibitionRadius	15
DutyCyclePeriod	25
MinPctOverlapDutyCycles	0.75
MaxSynapsesPerSegment	0.02 * numColumns
ActivationThreshold	15
ConnectedPermanence	0.5
PermanenceDecrement	0.25
PermanenceIncrement	0.15
PredictedSegmentDecrement	0.1

Table 2: HTM Config

Figure 10. Shows the training accuracy for a sequence of numbers for five sequences.

Figure 11 below shows the prediction for the sequence of Numbers for the trained data sequence as in file.

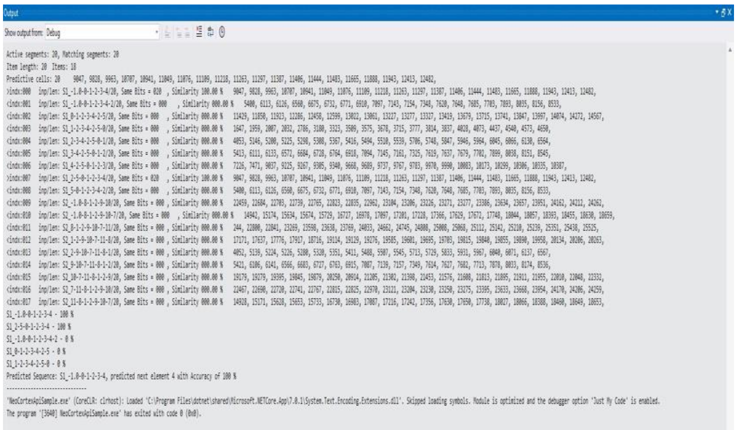

Figure 11: Prediction – Sequence of Numbers

Accuracy is defined as the ratio of the matched sequence count where is predicts the next element in an iteration to the total number of iterations.

VI. Conclusion

A solution for Multi Sequence learning of a Sequence of numbers was developed using the Neocortex API's Multi Sequence learning reference model. The HTM Prediction Engine was adjusted with various parameters to suit the training process. The Multi-Sequence of Numbers was saved and then transformed into an encoded value and stored in a dictionary using an Encoder and SDR input for the training process. The existing algorithm was improved to predict the trained sequences by reading from a file, which involved comparing the generated similarity matrix with each of the SDRs of the learned Sequence from the training phase. The Sequence was then predicted based on the accuracy and observation class (Label), and the accuracy percentage of the predicted sequences was calculated and stored in a file.

We performed Multi Sequence Learning for a sequence of numerical data sets and achieved the prediction accuracy of sequences.

The experiments conducted enabled us to gain insights into various aspects of the Neocortex API, including the functioning of encoders, how the Spatial Pooler generates SDR inputs and performs the learning phase, and the role of the Homeostatic Plasticity controller in stabilizing the learning process

VII. References

[1] D. G. K. S. Clegg BA, "Sequence learning. Trends Cogn Sci.," 1998. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/21227209/.

[2] B. D. Mauk MD, "The neural basis of temporal processing," 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] subutai, "Sparse Distributed Representations (SDRs)," 2017. [Online]. Available: https://discourse.numenta.org/t/sparse-distributed-representations/2150.

[5] K. Hole, "The HTM Learning Algorithm. In: Anti-fragile ICT Systems. Simula SpringerBriefs on Computing, vol 1," 2016. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-30070-2_11.

[6] O. Guy-Evans, "Neurons (Nerve Cells) Structure, Function & Types," 2023. [Online]. Available: https://simplypsychology.org/neuron.html.

[7] 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.

[8] Y. Cui, "HTM Spatial Pooler," 2017. [Online]. Available: https://numenta.github.io/numenta-web/assets/pdf/spatial-pooling-algorithm/HTM-Spatial-Pooler-Overview.pdf.