Multi-Sequence learning

Faiz  
[gaurav.honnavaramanjunath@stud.fra-uas.de](mailto:gaurav.honnavaramanjunath@stud.fra-uas.de)

Shiva Kumar Biru   
[harish.palanivel@stud.fra-uas.de](mailto:harish.palanivel@stud.fra-uas.de)

Ram Moham Sarnala   
[praveen.athkar@stud.fra-uas.de](mailto:praveen.athkar@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 subsequences from another file to calculate the percentage prediction accuracy. 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 accuracy and efficiency of sequence learning and prediction.**

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

# **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.

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.

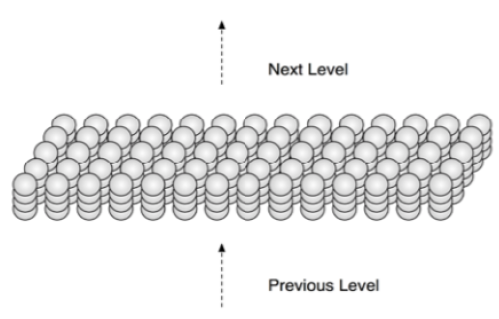
Working on the cortex, scientists discovered that sequence learning has a huge invariant changing series of inputs. The precise neurological process of sequence memory is yet understood, however models that provide a reading of the neurons are being studied. These models demonstrate great ability to 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 Biomimetics 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.

# **LITERATURE** **SURVEY**

## **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]

## **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*

## **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*

## **Neocortex**

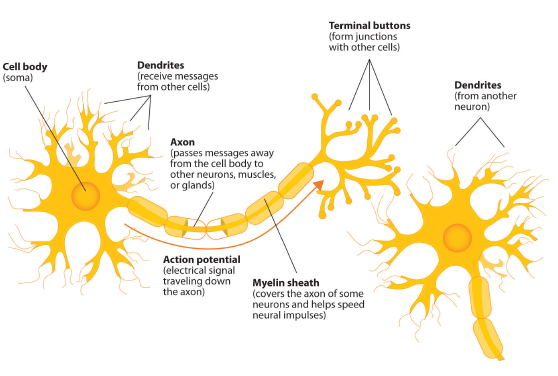
The neocortex is defined as the part of the cerebral cortex that serves mental functions for humans. It also contains billions of cells and some millions of meters. The cells are layered within which different regions are dedicated to vision, hearing, touch, movement, sensory balance, stimulus, etc.

*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.

## 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.

## **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.

## **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.

## **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.

# **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 accuracybased on the learning in HTM.

**Encoders**(Scalar Encoder/  
HTM Image Encoder)

**Data**

(Sequences of Numbers from the file)

**HTM Prediction Engine**

**Temporal Memory**

**Spatial Pooler**

**Accuracy Percentage**

**Prediction Algorithm**

**Homeostatic Plasticity Controller**

*Figure.6: Overview of Multi sequence Learning*

## **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

## **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.

## **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. 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. 9: HTM Algorithm Flow*

## **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 |
| 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.3: HTM Config parameters*

# **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.

## **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 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.

Graphical user interface, text, application

Description automatically generated

*Figure. 10: Training Model – Sequence of Numbers*

## **Cancer Sequence Classification**

In the experiment, Anticancer peptides cells are represented using a sequence of alphabets, with each alphabet being considered as a cell in the sequence. To train multiple sequences, the HTM method is employed. Each cancer sequence is treated as a single element with its own label, which facilitates classification at a later stage.

Diagram

Description automatically generated

*Figure 13: Cancer Sequence Classification*

## **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.

# **RESULTS**

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

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

Diagram

Description automatically generated

*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:

## **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.

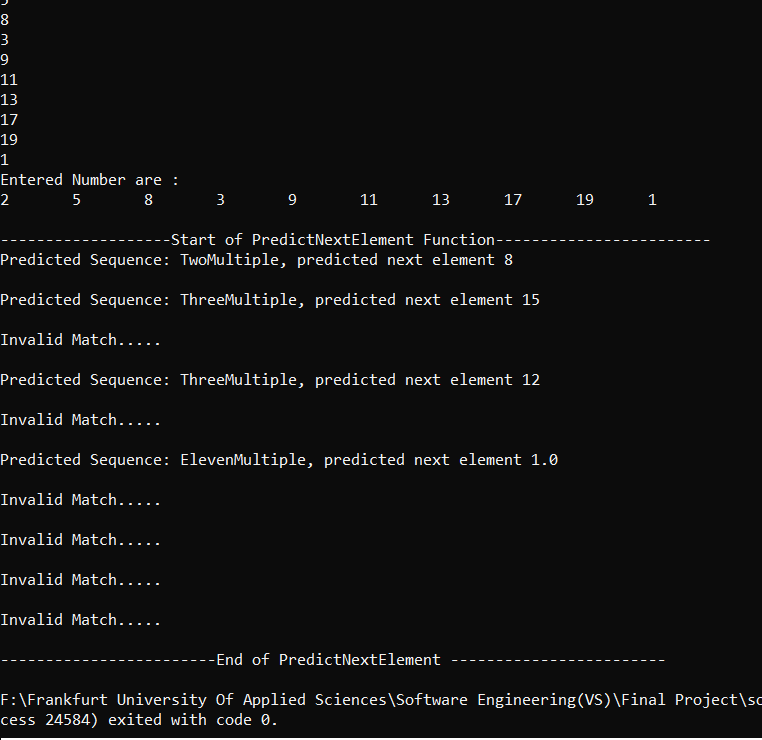
|  |  |
| --- | --- |
| **Parameters** | **value** |
| inputBits | 100 |
| numColumns | 1024 |
| CellsPerColumn | 25 |
| GlobalInhibition | true |
| LocalAreaDensity | -1 |
| NumActiveColumnsPerInhArea | 0.02 \* numColumns |
| PotentialRadius | 0.15 \* inputBits |
| MaxBoost | 10 |
| InhibitionRadius | 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 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.*



*Figure 16: Prediction – Sequence of Numbers*

# **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.

# **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. |