Multi-Image/Sequence learning

Gaurav Honnavara Manjunath  
[gaurav.honnavaramanjunath@stud.fra-uas.de](mailto:gaurav.honnavaramanjunath@stud.fra-uas.de)

Harish Palanivel  
[harish.palanivel@stud.fra-uas.de](mailto:harish.palanivel@stud.fra-uas.de)

Prajwal Praveen Athkar  
[praveen.athkar@stud.fra-uas.de](mailto: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 handles 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 for 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 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).

# **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].

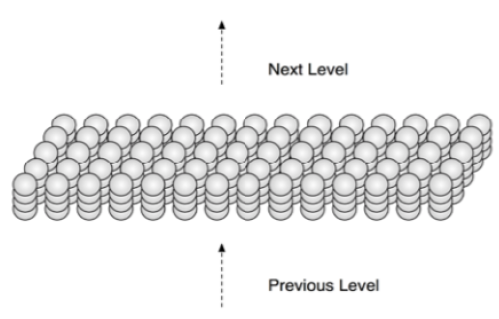
Scientists have gained insights by working on the cortex that sequence learning has large invariant changing series of inputs. The exact neural mechanism of sequence memory is still unknown, but models that give a reading of the neurons are used to study. These models show significant capabilities to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

# **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 at 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

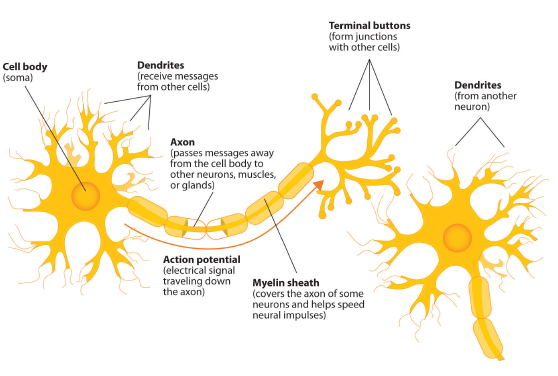
The neocortex is defined as the part of the cerebral cortex that serves as mental functions for humans. It also contains billions of cells and some millions of meters. The cells are layered within which different regions are dedicated for vision, hearing, touch, movement, sensory balance, stimulus, etc.

*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 advancement 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 concerning 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 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.

# **Methodolody**

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.

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

**Data**

(Sequence of Numbers/  
Sequence of Alphabets/  
Image Datasets)

**HTM Prediction Engine**

**Temporal Memory**

**Spatial Pooler**

**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.

1. 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} |

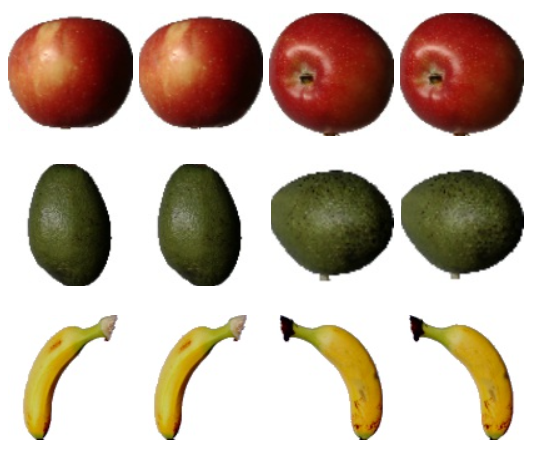
*Table.1: Date Sets (Numbers) – Multi sequence Learning*

1. 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]

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

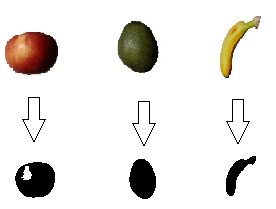


*Figure.7: Datasets (Image) – Multi sequence Learning*

## 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.7: Image Binarization*

## 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 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. 8: HTM Algorithm Flow*

## 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. Serialize the default configuration settings to the htmconfig.json file after initializing. The changed settings are loaded every time the user launches the HTM image categorization application. 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 \* 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 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.

## Learning Phase

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

1. 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 Number is carried out in the Multi Sequence Model.

Graphical user interface, text, application

Description automatically generated

*Figure. 9: Training Model – Sequence of Numbers*

1. 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.

Graphical user interface, text, application, chat or text message

Description automatically generated

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

1. 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.

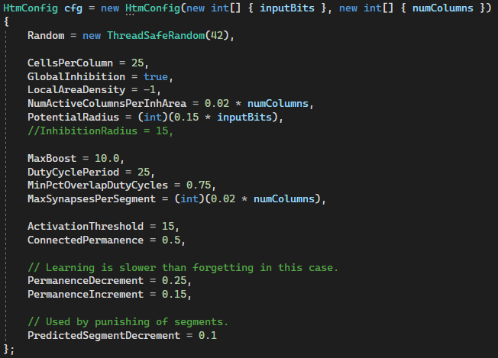
Graphical user interface, text

Description automatically generated

*Figure. 11: Training Model – Image Datasets*

## HTM Configuration

HTM configurations contain different parameters which are defined to control the behaviour such as permanence increment and decrement, max number of cells, and max cycles to train the input dataset.



*Figure. 12: HTM Configurations*

We are using Spatial Pooler with HomoPlasticityController. Spatial Pooler is used to training individually to combine SP and TM training. The structure of the spatial pooler is shown in the below figure which is taken from NeocortexApi.



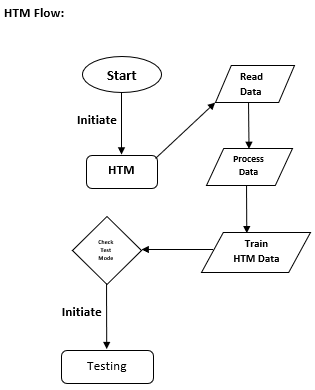
*Figure. 13: Spatial Pooler Implementation*



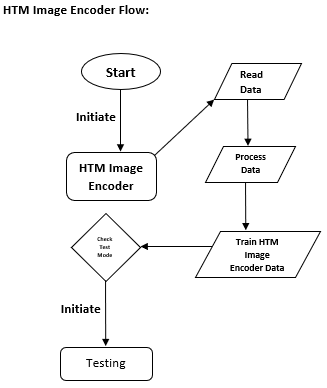
*Figure. 14: HPA configurations*

## HTM Flow

There are four steps in this HTM Flow i.e., reading data, processing the data, training the HTM network, and predicting the using the processed data. During data processing, for cancer cell sequence we use the scalar encoder and for the image dataset, we use HTM Image encoder.



*Figure. 15: HTM Process Flow Chart*



*Figure. 16: HTM Image Encoder Process Flow Chart*

##### results (IN PROGRESS)

TBD

## A. Cancer Sequence Classification

*HTM Training and Accuracy*

HTM gives very good performance and helps in classifying the cancer sequence. During the training phase, the next elements are predicted and are then connected to their labels of sequence. We have used around 1000 sequences in the experiment for better accuracy.

For testing, we are using 70 percent of the training data.

## B. Image Dataset Classification.

TBD

## Cancer Sequence Classification

Here multiple images are categorized which are identical from each other and considered as a sequence of images. Each image is binarized using the HTM image encoder to generate a binarized image (1,0). The classification of the images is done by encoding and saving them to respective folders and the folder name here is considered as the label of the image.

Diagram

Description automatically generated

*Figure. 10: Image Sequence Learning*

## Cancer Sequence Classification

*Figure. 9: Cancer Sequence Learning*

# **Conclusion**

TBD

# **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. |