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*— **The evolution of life on Earth has developed the human brain containing the cortex which in the environment processes a series of sensory information and builds temporary memory. This leads to identifying and foretelling information about the previously visualized functions like speech recognition, natural vision, and sequences of information. In real-life situations, the prediction and perception of temporal sequences for sensory inputs are useful/critical. Based on multiple known features of neurons, a theoretical framework has been proposed for sequence learning in the cortex is known as Hierarchical Temporal Memory (HTM) sequence learning. The model using temporal memory can handle varying the sequences by keeping the predictions until valid evidence is available.** **HTM’s ability to predict future patterns arrive from learning which patterns are likely to follow each other. In scenarios when HTM receives a unique pattern, it compares historically received patterns with the new pattern. As input never repeats in the same fashion, the uniqueness of sequence is critical for recognizing inputs. HTM uses shifting order memory, which enables it to predict using variable-length sequences.**

Keywords— HTM, sequence learning, temporal memory, prediction, cortex.

# 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.

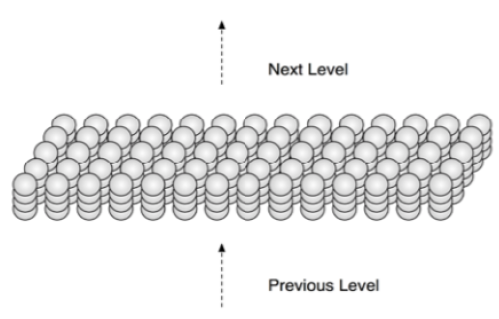
The medical sciences have advanced to provide us with a major understanding of the working of the cortex. Investigations have concluded that many cortical regions are part of the temporal sequence processing [1][2]. On the other hand, ML engineers have been researching sequential memory which led to several models for temporal pattern recognition [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 a sequence of inputs using rules. These ML models do not match the real-world issues.

Hierarchical Temporal Memory (HTM) is a Biomimetics model which is 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 results that are promising in pattern recognition. This can learn the temporal sequences and spatial flow of sensory inputs as data.

* SDRs

HTM’s language is Sparse Distributed representations (SDRs) of input patterns. It internally generates SDRs which are given a fixed number of active bits. These bits have semantic meaning, hence two inputs with similar semantic meaning must have similar active bits representation in SDR which plays a major 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 valid synaptic connection with a permanence value over the connection threshold is represented by a solid circle, while a possible synapse connection with a permanence value below the connection threshold is represented by an empty circle. If enough valid synapses are coupled to active input bits, feedforward input activates a column after a local inhibition step.



*Figure.2: Proximal Dendrite Representation [7]*

* Distal Dendrite Segments

A cell has over 130 distal dendrite segments, each with about 40 synapses, in addition to the 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 encounters the same cellular activation pattern again, that is, if the number of active synapses on any segment exceeds a threshold, the cell will enter a predictive state, signaling that feed-forward input will likely result in column activation shortly. Feedforward input through the proximal dendrite or lateral connections through the distal dendrite segments keeps a cell active.



*Figure.3: Distal Dendrite Segment at cellular level [7]*

# Literature Survey

## 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.

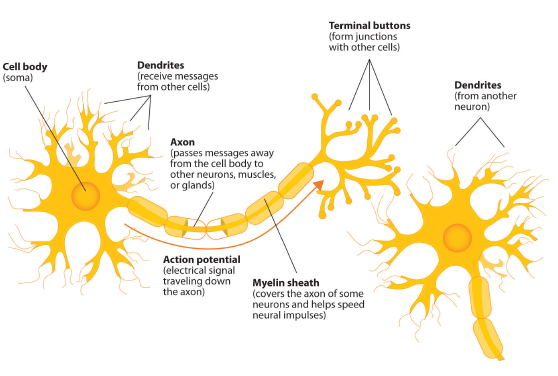


*Figure.4: Neocortex Layers [6]*

HTM is a working model which is inspired and designed trying to replicate the functionality of the biological neocortex in the brain. Its part is to learn the input data that is fed as sensory input. HTM uses different approaches to replicate the neuron model until the framework of the functionality is defined to accept the respective sensory inputs [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 form the base for memory and learning within the brain.



*Figure.5: Neocortex Layers [6]*

The signaling process is something like this: Neuron A becomes electrically charged in relation to the surrounding fluid outside its membrane when it receives a chemical signal from another neuron. Until it reaches a synapse, 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, where they bind 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. In parallel computers, many computations are carried out on the input patterns 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 unchanging patterns in hierarchies [7]. These associated memories can fetch complete patterns from partial input patterns in both 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 pattern 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 implemented here is divided into three stages. Stage 1 - Input Dataset is prepared and processed using the HTM algorithm. Stage 2- The dataset is trained using the HTM network respectively. We are using Cancer cells and Images as datasets. Stage 3 – results are evaluated and visualized using a prediction algorithm.

## Dataset preprocessing

We have used two datasets i.e., 1) Sequence of Numbers 2) Cancer Peptides sequences [9] 3) Sequence of Images

## Number Sequence DataSet Processing

In this sequence learning experiment, we have used a sequence of numbers in the pattern – multiples of two, three, four, and so on and repeated a few numbers in the same row for better training of the HTM Model. This is a self-implemented Dataset.

## Cancer Peptides cells dataset processing

In this sequence learning experiment, we have used Cancer Peptides cells as a dataset. While training the dataset, we are learning the sequences with different sequence labels. For HTM, we have used a scalar encoder to encode alphabet values into SDR.

Two peptide datasets targeting breast and lung cancer cells were assembled and curated manually from Cancer PPD. EC50, IC50, LD50, and LC50 annotations on breast and lung cancer cells were retained (breast cell lines: MCF7â€‰=â€‰57%, MDA-MB-361â€‰=â€‰11%, MT-1â€‰=â€‰9%; lung cell lines: H-1299â€‰=â€‰45%, A-549â€‰=â€‰17.7%); mg mlâˆ’1 values were converted to Î¼M units. Linear and l-chiral peptides were retained, while cyclic, mixed, or d-chiral peptides were discarded. In the presence of both amidated and non-amidated data for the same sequence, only the value referred to the amidated peptide was retained. Peptides were split into three classes for model training: (1) very active (EC/IC/LD/LC50â€‰â‰¤â€‰5 Î¼M), (2) moderately active (EC/IC/LD/LC50 values up to 50 Î¼M), and (3) inactive (EC/IC/LD/LC50â€‰>â€‰50 Î¼M) peptides. Duplicates with conflicting class annotations were compared manually to the original sources, and, if necessary, corrected. If multiple class annotations were present for the same sequence, the most frequently represented class was chosen; in the case of ties, the less active class was chosen. Since the Cancer PPD is biased towards the annotation of active peptides, we built a set of presumably inactive peptides by randomly extracting 750 alpha-helical sequences from crystal structures deposited in the Protein Data Bank (7â€“30 amino acids).

Attribute Information:

The dataset contains three attributes:

1. Peptide ID

2. One-letter amino-acid sequence

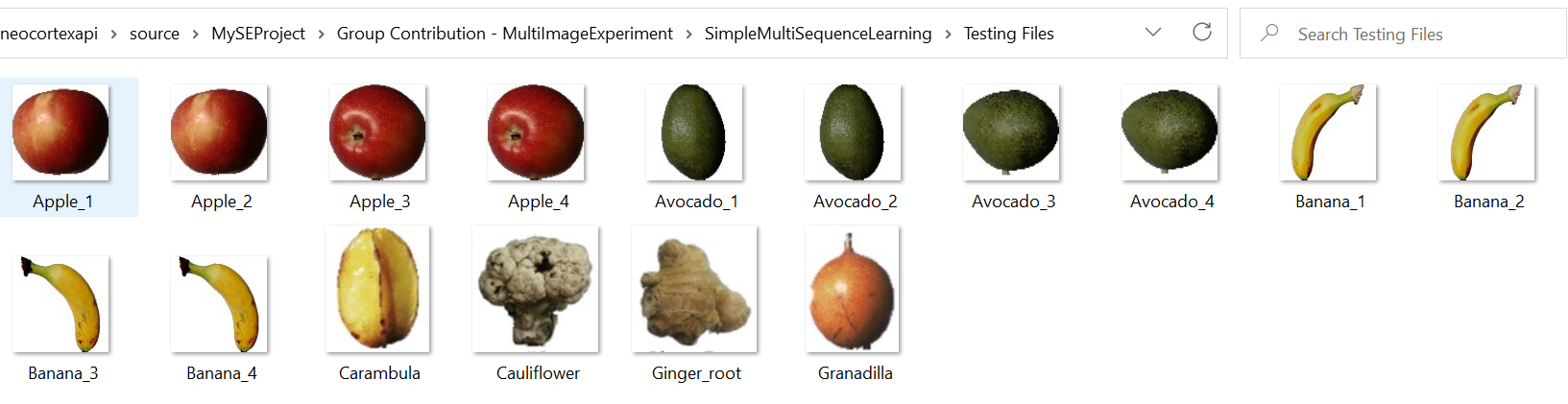
3. Class (active, moderately active, experimental inactive, virtual inactive)



*Figure. 6: Cancer Peptide Cell Dataset Preview*

## Image Dataset processing

For Image sequence classification, we have used sequences of images as the dataset. Here we have Apple, Avocado, and Banana series of other images in the dataset. The series of images are different from each other. While training the dataset, we are learning different series of images to identify the category. For HTM, we have used an Image encoder to binarize the image and encode it into SDR.



*Figure. 7: Image Dataset Preview*

## HTM Network Training

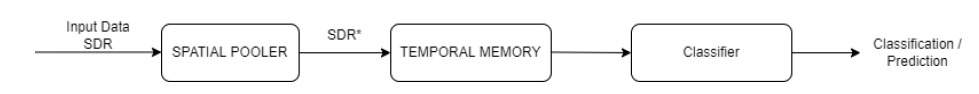
At each difference time, HTM executes three steps on the input. The steps are described below:

Step 1: Create an SDR of the input by activating the whole columns.

Step 2: Place the input by selecting the cells from active columns.

Step 3: Predict the next patterns from the trained sequences from SDRs.

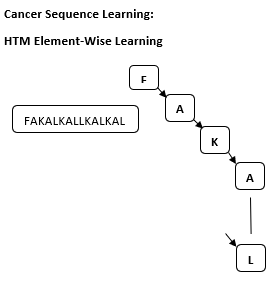
Spatial Pooler creates SDR input, during which the cells of the active columns are mapped. Each column has a network of connections with the next region of input bits via synapses. Many columns would look the same but these columns are unique from each other. Different patterns produce different levels of activation, the stronger 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.



*Figure. 8: HTM Algorithm Flow*

## Cancer Sequence Classification

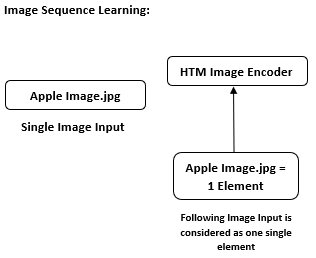
The experiment here uses cancer peptides cells which are represented using a sequence of alphabets. All the alphabet are treated as cells of the sequence. We use HTM to train multiple sequences. Each cancer sequence is treated as a row of a single element associated with a label which helps in classification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



*Figure. 9: Cancer Sequence Learning*

## Image dataset 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.



*Figure. 10: Image Sequence Learning*

# implementation

Implementation of HTM network is performed. The dataset preparation and configuration of the components are detailed below.

## HTM Implementation

Using NeocortexApi, the HTM .Net implementation is carried out. The NuGet package is available in [10]. This package is an implementation of the Hierarchical Temporal Memory Cortical Learning Algorithm (HTM CLA) in C#/.NET Core. It includes the Spatial Pooler, Temporal Memory, various encoders, and Cortical Network algorithms.

## Number Sequence Dataset Preparation

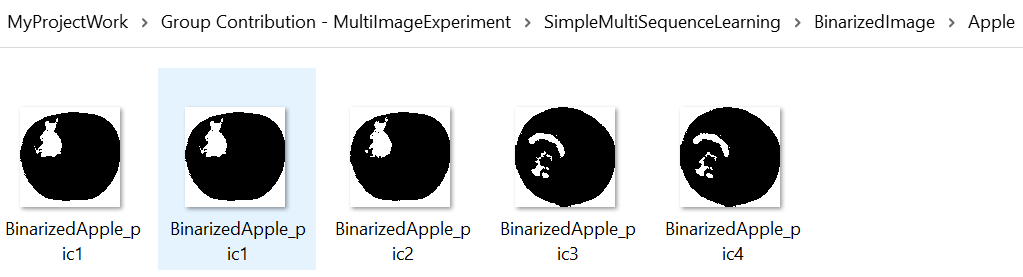
For Number sequence classification, we are providing the sequence of numbers as input data and encoding it using a Scalar encoder to process the dataset. In this experiment, no further process is required in dataset preparation.

## Cancer Sequence Dataset Preparation

For Cancer sequence classification, we are fetching the .csv file that contains the sequence of input data and encoding it using a Scalar encoder to process the dataset. In this experiment, no further process is required in dataset preparation.

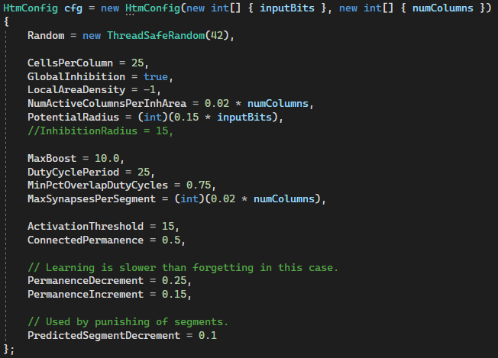
## Image Dataset Preparation

For Image sequence classification, we are fetching the image from each folder which contains a series of images and is encoded using an HTM Image encoder which is initially binarized. The binarized image is the processed data.

 *Figure. 11: Binarized Image Output*

## HTM Configuration

HTM configurations contain different parameters which are defined to control the behavior 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.



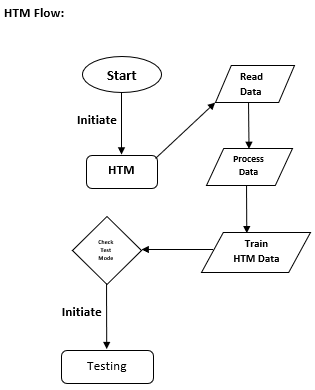
*Figure. 13: Spatial Pooler Implementation*



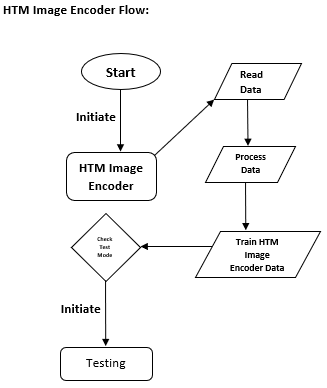
*Figure. 14: HPA configurations*

## 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

## 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.



Fig: Confusion Matrix

## B. Image Dataset Classification.

Need to work on it

##### Conclusion

Group discussion required.

##### References

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.