Investigate influence of parameter MaxNewSynapseCount

Akash Saha
akash.saha@stud.fra-uas.de

Md Fathir Ahmed Shishir
md.shishir@stud.fra-uas.de

***Abstract*—Hierarchical Temporal Memory (HTM) is the consider as new era implementation of machine learning algorithm. This algorithm is based on neocortex of human brain. Our human brain is very complex system which learns and understand in different way as compared any other computation learning algorithm which have been implemented. HTM is specifically designed to learn as if brain is computing input from a sensory organ. These sensory organs take real world input and encode it in sparse distributed representation (SDR). This representation is later feed to HTM algorithm. A single SDR represents multiple properties of single input. While learning happens we do not want to give rise to problem which traditional machine learning algorithms have such as overfitting, zero filling null values, working with mean which misleads outliers in the input and hence HTM algorithm have different sort of parameter such as sparsity, learning rate, inhibition rate and so on. One of the parameters called as MaxNewSynapseCount is responsible for growth of synapses while learning takes place. We have tried to analyse the influence of MaxNewSynapseCount parameter in MultiSequenceLearning Algorithm.**

Keywords— Machine Learning, neural network, artificial intellegence, hierarchical temporal memory, sparse distributed representation, encoders, spatial pooler, sequence learning, temporal memory, MaxNewSynapseCount.

I. Introduction

Hierarchical Temporal Memory (HTM) stands at the forefront of bio-inspired artificial intelligence, mimicking the human neocortex's architecture to achieve remarkable learning and prediction capabilities. At the heart of HTM's efficacy is its Temporal Memory (TM) phase, responsible for learning sequences and making predictions based on spatial and temporal patterns. Among the various parameters governing TM's functionality, MaxNewSynapseCount emerges as pivotal, dictating the maximum number of new synaptic connections that can be formed to previously unconnected neurons during the learning process.

The significance of MaxNewSynapseCount extends beyond mere numerical value; it encapsulates the balance between network plasticity and stability. High values foster an environment ripe for rapid learning, allowing the network to adapt swiftly to new patterns. Conversely, lower values constrain the network's adaptive capabilities, potentially leading to slower learning rates but potentially enhancing the specificity and stability of the learned representations.

This research is motivated by the hypothesis that there exists an optimal MaxNewSynapseCount setting that harmonizes learning speed and prediction accuracy, thereby maximizing the HTM network's performance for various tasks. To investigate this, we embarked on a comprehensive study, methodically varying MaxNewSynapseCount and meticulously observing its effects on network behavior. This endeavor not only seeks to illuminate the intricate dynamics of HTM networks but also aims to contribute practical guidelines for configuring HTM models, enhancing their applicability across diverse domains.

II. Literature Survey

A. *Hierarchical Temporal Memory*

Hierarchical Temporal Memory (HTM) is a modern world machine learning model. Its adaptable and biologically faithful framework makes it suitable for solving a wide range of data-driven problems, including prediction, classification, and anomaly detection. To feed data into HTM systems, Sparse Distributed Representations (SDRs) are required, which differ significantly from conventional computer representations like ASCII for text. SDRs are encoded directly into the representation and comprise mostly zeros, with only a few ones. Each one-bit holds a unique semantic meaning, and if two SDRs have several overlapping one-bits, they are considered to have similar meanings [5].

HTM can be considered as a type of neural network, with interconnected biological neurons forming layers of nodes or neurons. The output from one layer is passed on to the next layer, and the connections between them can be adjusted using a bias. However, HTM is structured hierarchically, much like the neocortex of the human brain. Each column in HTM contains several cells that can either be active or inactive, depending on the input received in the form of SDRs.

B. *Sparse Distributed Representation*

Sparse distributed representation (SDR) is a method of data encoding that aims to represent information using sparse patterns of activity in a high-dimensional space. In SDR, only a small subset of the elements in a vector are active, while the rest are inactive or zero. This type of encoding has been shown to be efficient, flexible, and robust, and has applications in various fields, including neuroscience, artificial intelligence, and information theory [2]. SDRs can handle missing and noisy data and can generalize to new patterns that are similar to learned patterns.

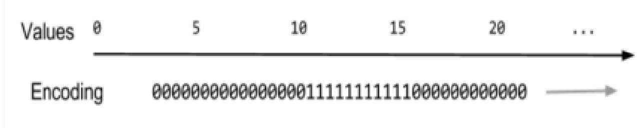

Figure 1: SDR of a number

C. *Spatial Pooler*

The Spatial Pooler is responsible for creating a sparse distributed representation of the input data, which is then fed into the Temporal Memory component for further processing.

The Spatial Pooler works by creating a set of columns, each of which represents a potential input feature [3]. The input data is then mapped onto these columns, with each input feature activating a subset of the columns. The Spatial Pooler then selects a small subset of the active columns to represent the input data, using a process known as inhibition.

The inhibition process involves selecting a fixed percentage of the most highly activated columns and suppressing the activation of all other columns. This results in a sparse distributed representation, where only a small subset of the columns is active at any given time.

D. *Temporal Memory*

The temporal memory component of HTM is responsible for learning and predicting sequences in time-varying input data.

Temporal Memory works by maintaining a set of cells that represent patterns of activity in the input data. These cells are connected to each other through synapses, which can strengthen or weaken over time based on the input data patterns. As the input data sequences are presented to the temporal memory, the cells become active in a sparse and distributed manner, forming a representation of the input sequence.

When a new input sequence is presented to the temporal memory, it compares the sequence to the patterns it has learned and predicts the next likely pattern based on the sequence's context.

E. *Encoder*

An encoder is a technique that converts unprocessed input data into a sparsely dispersed depiction, which can be handled by the Spatial Pooler component [4]. The encoder's task is to transform continuous input data into a collection of distinct categories, forming a representation that is resistant to noise and can adapt to novel patterns.

In HTM, various encoder types are available, which include scalar encoders, category encoders, and datetime encoders. Among these, scalar encoders are the most frequently employed type and are utilized to encode continuous data like temperature values, sensor readings, and audio signals. On the other hand, category encoders are employed to encode discrete data such as words, colors, or symbols.

III. METHODOLOGY

Most of the time in this experiment was spent on understanding and experimenting on Multisequence Learning [5]. The Multisequence Learning is based on HTM Classifier taken from the NeoCortexApi [1].

Our investigation into the MaxNewSynapseCount parameter's influence employs a robust experimental framework designed to yield precise, actionable insights. Central to our approach is the modification of the MultiSequenceLearning[5] class, which we adapted to dynamically accept different MaxNewSynapseCount values. This flexibility is crucial for our experimental design, allowing us to systematically explore a wide range of settings and their impacts on the network's learning and predictive performance.

Our investigation into the MaxNewSynapseCount parameter's influence employs a robust experimental framework designed to yield precise, actionable insights. Central to our approach is the modification of the MultiSequenceLearning class, which we adapted to dynamically accept different MaxNewSynapseCount values. This flexibility is crucial for our experimental design, allowing us to systematically explore a wide range of settings and their impacts on the network's learning and predictive performance.Experimental Setup

We initiated our study by establishing a baseline HTM network configuration, utilizing the NeoCortexApi framework for its proven reliability and flexibility. The network was structured to mirror key aspects of the human neocortex, with a focus on the Temporal Memory phase to closely examine sequence learning and prediction dynamics.

Parameter Variation and Data Collection: With the experimental framework in place, we conducted a series of runs, each time varying the MaxNewSynapseCount within a predetermined range. This methodical variation aimed to span a broad spectrum of potential network behaviors, from highly plastic and rapidly adapting networks to more stable and slowly learning configurations.

For each experimental run, we meticulously logged a wealth of data, including learning cycles, prediction accuracy rates, and the specific MaxNewSynapseCount employed. This detailed data collection was instrumental in our analysis, providing a rich dataset for assessing the parameter's impact on network performance.

Analysis Techniques: Our analysis leveraged both quantitative metrics, such as learning speed and accuracy rates, and qualitative observations of network behavior over time. By applying statistical methods and visualizations, we aimed to extract meaningful patterns and relationships from the collected data, with a particular focus on identifying the MaxNewSynapseCount settings that yielded optimal network performance.

- Prediction Accuracy Test 1: To rigorously evaluate the impact of the MaxNewSynapseCount parameter on the Hierarchical Temporal Memory (HTM) network's prediction accuracy, we designed a specific test, PredictionAccuracyTest1[6], within our experimental framework. This test aims to compare the prediction accuracy of HTM networks configured with distinct MaxNewSynapseCount settings, thereby elucidating the parameter's influence on the network's ability to generalize from learned sequences. The test leverages a set of predefined sequences provided by the GetTestSequences1 [7] method. This method returns a dictionary of sequences, each identified by a unique key and comprising a list of double values representing the sequence elements. The sequences are carefully chosen to include a variety of patterns and lengths, challenging the network's learning and prediction capabilities. For instance, S1 and S2 consist of extended numerical sequences with varied repetition and progression patterns, aimed at simulating complex temporal patterns that the HTM network must learn to predict. Test Execution of PredictionAccuracyTest1 initiates by fetching the test sequences through the GetTestSequences1 method. It then conducts two separate experiment runs with differing MaxNewSynapseCount values—5 and 20—to explore how a lower versus a higher limit on new synapse formation affects the network's predictive accuracy. The method calculates the prediction accuracy for each MaxNewSynapseCount setting by evaluating the network's performance in forecasting subsequent elements of the test sequences. Accuracy is defined by the proportion of correctly predicted sequence elements relative to the total number of predictions made. Following the completion of both runs, the test compares the achieved accuracies to determine which MaxNewSynapseCount setting yielded superior predictive performance. The comparison results, including the best accuracy obtained and the corresponding MaxNewSynapseCount value, are then logged for analysis. We know from this test is that 20 is better than 5 in terms of Accuracy.

- Prediction Accuracy Test 2: Building upon our foundational assessment of how MaxNewSynapseCount affects HTM network performance, the PredictionAccuracyTest2 [8] method introduces a nuanced layer of investigation. This test is designed to probe deeper into the parameter’s impact, utilizing a more complex and extended set of input sequences to challenge the network's learning and predictive capabilities.The method leverages the GetTestSequences2 [9] function to procure a sophisticated collection of test sequences, each embodying a greater depth of pattern complexity and sequence length compared to the initial test set. This deliberate choice aims to simulate more challenging learning scenarios, thereby providing a rigorous testing ground for evaluating the HTM model's adaptability and accuracy under varied MaxNewSynapseCount settings. In this iteration of accuracy testing, MaxNewSynapseCount values are set to 20 and 40 for the low and high configurations, respectively. These specific values are chosen to represent a moderate and a relatively high capacity for synaptic creation, enabling a comparative analysis that sheds light on the balance between synaptic flexibility and overextension in relation to prediction accuracy. For each configured MaxNewSynapseCount value, the HTM network is trained on the provided sequences, after which its prediction accuracy is meticulously quantified. The test method then compares these accuracy metrics to identify the MaxNewSynapseCount setting that yields the highest predictive success, offering insights into optimal synaptic creation thresholds for handling complex sequence predictions. The pivotal outcome of this method is the determination of the MaxNewSynapseCount value that achieves the best prediction accuracy, underpinning the parameter's critical role in tuning the HTM network for enhanced performance. By asserting that a positive accuracy was achieved, the test not only validates the experimental approach but also emphasizes the HTM model's potential in learning from and predicting complex sequences effectively.

- Compare Learning Speed Test 1: The CompareLearningSpeedWithDifferentSynapseCounts [10] test method plays a pivotal role in assessing the influence of the MaxNewSynapseCount parameter on the learning efficiency of Hierarchical Temporal Memory (HTM) networks. This methodological approach aims to quantify the temporal efficiency of learning processes under varying synaptic thresholds, providing critical insights into optimizing HTM configurations for enhanced performance. The primary objective of this test is to evaluate how the number of learning cycles required to reach a predetermined accuracy threshold varies with different MaxNewSynapseCount settings. This approach facilitates a direct comparison between lower and higher synaptic count configurations, illuminating their respective impacts on the network's learning speed. The method begins by fetching a set of sequences using the GetTestSequences1 function, designed to serve as standardized input data for the learning experiments. These sequences represent varied patterns that the HTM network attempts to learn, providing a consistent basis for evaluating learning speed across different experimental conditions.The experiment is structured to test the HTM network's learning speed at two distinct MaxNewSynapseCount values: a lower setting of 5 and a higher setting of 20. This design choice reflects the intent to explore the spectrum of synaptic creation capacity, from restrictive to expansive, assessing its effect on the network's ability to rapidly assimilate and predict sequence patterns. For each MaxNewSynapseCount setting, the GetCyclesToReachAccuracy [11] function is invoked to ascertain the number of learning cycles necessary to achieve an accuracy threshold of 90%. This metric serves as a quantifiable measure of learning efficiency, enabling an objective comparison between the different synaptic configurations. Upon obtaining the learning cycle counts for the low and high MaxNewSynapseCount configurations, the method compares these values to identify which setting demonstrates superior learning efficiency. The comparison hinges on identifying the minimum cycle count required to meet the accuracy benchmark, signifying the most efficient synaptic count configuration in terms of learning speed. The concluding assertion verifies the attainment of a positive cycle count indicative of successful learning, ensuring the validity of the experimental outcomes. This step underscores the method's flexibility in acknowledging synaptic count setting as potentially superior, contingent upon empirical evidence of learning efficiency. This methodology offers a systematic and analytical framework for dissecting the role of MaxNewSynapseCount in modulating the learning dynamics of HTM networks. By focusing on the relationship between synaptic creation capacity and learning speed, the CompareLearningSpeedWithDifferentSynapseCounts test method contributes valuable insights into the optimization of HTM models for rapid and efficient learning. This approach not only enhances our understanding of HTM's operational parameters but also guides the tuning of these networks for optimal performance across a range of tasks and applications. The validity of the experimental outcomes is verified through an assertion that checks for a positive accuracy value, ensuring that the HTM network was capable of learning and making successful predictions from the provided sequences. This assertion serves as a basic check for the experiment's success, confirming that the network achieved a meaningful level of prediction accuracy under at least one of the tested MaxNewSynapseCount configurations.

- Compare Learning Speed Test 2: In the CompareLearningSpeedWithDifferentSynapseCounts2 [12] test method, the focus shifts towards evaluating the HTM network's learning speed using a more intricate set of sequences, aiming to discern the impact of higher MaxNewSynapseCount values on the network's efficiency. This evaluation is predicated on reaching a 90% accuracy threshold, serving as a benchmark for successful learning. The method systematically compares the learning speed at two distinct MaxNewSynapseCount settings: 20 and 40. This comparison is designed to investigate whether a higher capacity for synaptic creation enables the network to learn complex sequences more rapidly, thereby enhancing overall learning efficiency. The procedure involves calculating the number of learning cycles required to meet or exceed the accuracy threshold at each specified MaxNewSynapseCount setting. By doing so, it provides valuable insights into how varying the limit on new synapse formation influences the time it takes for the HTM network to adapt to and accurately predict complex sequences. The determination of the "best" MaxNewSynapseCount, in terms of learning efficiency, is made by identifying the setting that achieves the desired level of accuracy with the fewest number of cycles. This methodological approach not only sheds light on the optimal synaptic creation capacity for accelerating the learning process in the face of challenging sequence data but also underscores the nuanced relationship between MaxNewSynapseCount and the HTM network's learning dynamics. The goal is to refine our understanding of HTM parameter tuning, ensuring that networks are optimally configured to tackle a broad array of learning tasks with enhanced speed and accuracy. The assertion that a valid cycle count was achieved further validates the experimental design, confirming that the HTM model is capable of effective learning within the tested parameter ranges.

IV. RESULTS

We conducted a series of manual and automated unit test in order to investigate the influence of parameter MaxNewSynapseCount on MultiSequenceLearning algorithm. We have analyzed the generated logs manually to determine which MaxNewSynapseCount parameter is optimal for learning and predicting. The tests we have created, we must set the MaxNewSynapseCount manually. We could have also used the DataRow attribute but as the execution time is much longer than usual, we didn't use the attribute. The results we got from manual and automated testing is given below in the table,

Input to learn	Value
S1	0.0, 1.0, 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 5.0, 4.0, 3.0, 7.0, 1.0, 9.0, 12.0, 11.0, 12.0, 13.0, 14.0, 11.0, 12.0, 14.0
S2	0.8, 2.0, 0.0, 3.0, 3.0, 4.0, 5.0, 6.0, 5.0, 7.0, 2.0, 7.0, 1.0, 9.0, 11.0, 11.0, 10.0, 13.0, 14.0, 11.0, 7.0, 6.0

Table 1: Input of unit tests set 1

MaxNewSynapseCount	Cycles to stabilize
10	Learning didn't happen
20	230
30	267
40	301
50	234

Table 2: Unit tests set 1 cycles results

MaxNewSynapseCount	Accuracy (%)
10	Learning didn't happen
20	85.7
30	83.2
40	85.7
50	85.7

Table 3: Unit tests set 1 accuracy results

Here, we can see in terms of Cycle count and accuracy, MaxNewSynapseCount value 20 performance better. For given input of S1, S2 values changed we get different results.

Input to learn	Value
S1	0.0, 1.0, 0.0, 2.0, 3.0, 4.0, 5.0
S2	8.0, 1.0, 2.0, 9.0, 10.0, 7.0, 11.00

Table 4: Input of unit tests set 2

MaxNewSynapseCount	Cycles to stabilize
10	256
20	238
30	216
40	234
50	227

Table 5: Unit tests set 2 cycles results

MaxNewSynapseCount	Accuracy (%)
10	93.3
20	93.7
30	90.2
40	93.7
50	93.7

Table 6: Unit tests set 2 accuracy results

Here from the above result, we can see that almost all the parameters perform identically when given a small input to learn. So, in terms of the optimal parameter value of MaxNewSynapseCount 50 looks better.

V. Conclusion & Future Work

All the manual and automated unit test we performed with varying MaxNewSyanpseCount in order to investigate its influence on learning capabilities of temporal memory led us to believe that there are optimal MaxNewSyanpseCount values for different learning scenarios. Also, at the time of development and running the test we realized that this project is much more suitable for cloud implementation. Because in our local machine as the resource is very much limited it led to higher execution time and lower automation of dynamic input values and MaxNewSyanpseCount values.

VI. References

[1] D. Dobric, *et el*, "NeoCortexApi, : https://github.com/ddobric/neocortexapi".

[2] Jeff Hawkins, "On Intelligence" (2004)

[3] Subutai Ahmad and Jeff Hawkins, "A Review of Hierarchical Temporal Memory", 2016

[4] Matthew K. Graham, William C. Gross, Subutai Ahmad, and Jeff Hawkins, "Real-Time Anomaly Detection in Streaming Sensor Data Using Hierarchical Temporal Memory", 2019

[5] MD.F.A.Shishir, Akash Saha, *MultiSequenceLearning*, 2024, Source code

[6] MD.F.A.Shishir, Akash Saha, *PredictionAccuracyTest1*, 2024, Source code

[7] MD.F.A.Shishir, Akash Saha, *GetTestSequences1*, 2024, Source code

[8] MD.F.A.Shishir, Akash Saha, *PredictionAccuracyTest2*, 2024, Source code

[9] MD.F.A.Shishir, Akash Saha, *Compare Learning Speed With Different Synapse Counts*, 2024, Source code

[10] MD.F.A.Shishir, Akash Saha, *Compare Learning Speed With Different Synapse Counts 2*, 2024, Source code

[11] MD.F.A.Shishir, Akash Saha, *Get Cycles to Reach accuracy*, 2024, Source code

[12] MD.F.A.Shishir, Akash Saha, *Test with MaxNewSynapseCount*, 2024, Source code