Test and Investigation of Video Learning Project

Tuan Nghia Nguyen
ntuan@fra-uas.de

Khaled kandil
khaled.kandil@fra-uas.de

***Abstract*-The main purpose of the project is to test and investigate the Video Learning project. In order to do that, different video inputs were provided to the program and is applied with various parameters in the configuration. Afterwards, the results are carefully documented for each experiment, which are demonstrated as data tables. It is observed that the project performed decently in average, along with some great exceptions under specific conditions. Overall, the Video Learning process has been developed properly, that only a few adjustments to the configuration in the code of the program are required so that the trained model can achieve an even better performance.**

Keywords—parameters, HTM, Video Learning, frame, HTM configurations

I. INTRODUCTION

Hierarchical temporal memory (HTM) is a biologically constrained machine intelligence technology developed by Numenta. Originally described in the 2004 book On Intelligence by Jeff Hawkins with Sandra Blakeslee, HTM is primarily used today for anomaly detection in streaming data. The technology is based on neuroscience and the physiology and interaction of pyramidal neurons in the neocortex of the mammalian (in particular, human) brain[1]. Regarding the concept, this project takes on the already developed framework of HTM, in which a model is trained to learn and remember videos. Although the Video Learning program is already completed and fully functional, it's performance has not been thoroughly tested under different settings. Therefore, it is necessary to not only experiment the software with various input and several modified parameters, but also to record the result in detail. More over, these settings can be implemented later that results in a much more optimal video learning, i.e, higher accuracy with lower elapsed time.

II. METHODS

The aforementioned project of Video Learning with HTM functions by using Temporal Memory to learn binary representation of videos (sequence of bit-arrays, with each bit-array represents 1 frame). This whole experiment was conducted with the input of randomly chosen videos to ensure the project's objectivity in order to evaluate the video learning process's performance accurately. Firstly, the source code of the Video Learning project[2] was executed and the video's input folder path was required to be dragged into the command prompt of the program. After a period of run time, the result of the learning process was determined by how well the trained model could predict the next frame of the video based on what it had learned, when a specific frame was assigned to the model. The model's performance was assessed by two aspects: the accuracy and the elapsed time of the learning process, which were obtained by experimenting with following two types of input videos.

A. Simple Shape Video Inputs

The first experiment was running the program with many set of simple videos, in which only basic shapes (a black circle, rectangle and triangle) moving around on a white background[2]. These videos are generated via Python code. All of them have the same duration length of 2 seconds. The difference between these sets is the angle that these shape moves. By changing the Angle parameter in the python code[3], different set of videos with different moving angle (varies from 0 to 360 degree for this experiment) can be generated. After that, these video inputs were tested with Run1 and Run2. These are different ways how the program will process the inputs, which would be specifically explained later.

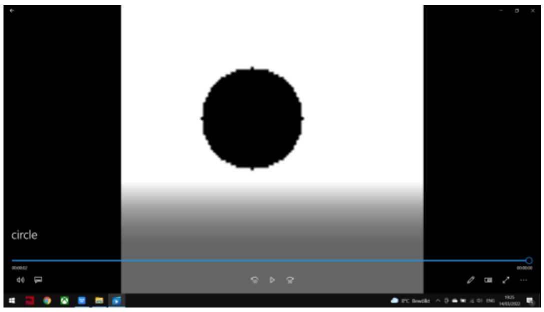

Figure 1a: Circle Video

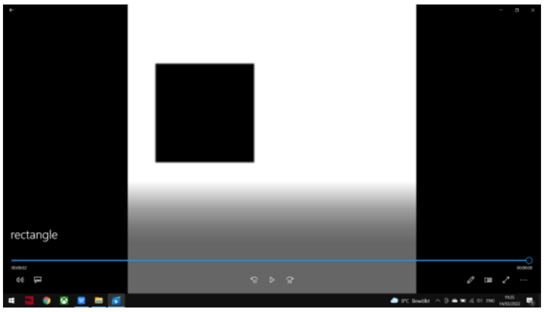

Figure 1b: Rectangle Video

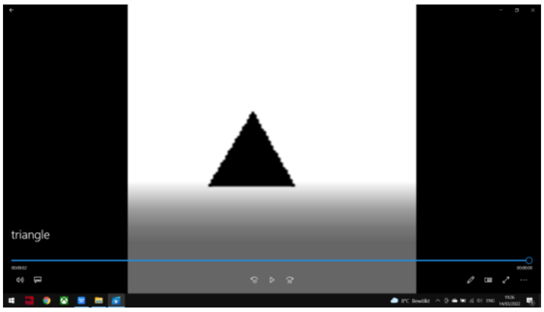

Figure 1c: Triangle Video

B. *Random Video Inputs*

A random music video on Youtube[4] is edited down to a small 5-second video which is used as an input for the learning process of the program. The editing is done by an online editor called Kapwing[5]. This short-version video[6] was put under many different conditions, such as by changing the video's resolutions or the HTM configurations of the program, including Cells per column, Global Inhibition, Max Boost, etc. Another Video input was used ("Frankfurt Video") edited first into 11 seconds with 184 frames which was a lot of patterns to take care of though for Run1 testing resolution was done in a reasonable time, the video was edited down again into 5 seconds with 85 frames, but it took a very long time to learn such video so Youtube Video was carried on for testing on Run2.

Figure 2a: Frankfurt Video

Figure 2b: Youtube Video

These Configuration was used for these inputs: Video's configuration[7], Default HTM configuration[8]. For the HTM configuration experiment, each below parameter is modified independently, while the others remained the same as the default HTM Configuration. The value of N/A indicates the experiment required too long an elapsed time without reaching a specific result, so they were cancel. Additionally, only Run2 was applied for this testing. Because Run1 (with Max Cycle = 1000 so the model could reach proper accuracy) required too much run time and there would not be enough time to experiment with various HTM configuration parameters. Additionally, Run2 is superior since the key used for learning is generated from the FrameKey List previousInputs of the current Video, in which one frame information is associated with the information of the whole frame sequence (Video). Compared to Run1 that predicts the frame one by one, the error of the learning process is reduced. Therefore, testing with only Run2 would be a much more optimal choice.

Additionally, the parameters of the HTM configuration that were used to experiment the software's performance include:

- Cells per column: indicates the number of input cells in each column cell.
- GlobalInhibition: If the value is TRUE, global inhibition algorithm will be used. If it is FALSE, local inhibition algorithm will be used.
- LocalAreaDensity: Density of active columns inside of local inhibition radius. If set on value < 0, explicit NumActiveColumnsPerInhArea will be used.
- NumActiveColumnsPerInhArea: An alternate way to control the density of the active columns. If this value is specified then LocalAreaDensity must be less than 0, and vice versa.
- PotentialRadius: Defines the radius in number of input cells visible to column cells. It is important to choose this value, so every input neuron is connected to at least a single column. For example, if the input has 50000 bits and the column topology is 500, then you must choose some value larger than 50000/500 > 100.
- InhibitionRadius: Defines neighbourhood radius of a column.
- MaxBoost: Maximum boost factor of a column.
- DutyCyclePeriod: Number of iterations. The period used to calculate duty cycles. Higher values make it take longer to respond to changes in boost. Shorter values make it more unstable and likely to oscillate.

- MinimumPctOverlapDutyCycles
- MaxSynapsesperSegment: Defines the maximum number of Synapses for each Segment.
- ActivationThreshold: One mini-column is active if its overlap exceeds overlap threshold θo of connected synapses.
- ConnectedPermanence: Defines Connected Permanence Threshold θp, which is a float value, which must be exceeded to declare synapse as connected.
- PermanenceDecrement: Decrement step of synapse permanence value within every inactive cycle. It defines how fast the NeoCortex will forget learned patterns.
- PermanenceIncrement: Increment step of connected synapse during learning process.

III. Results

A. Angle Experiments with Simple Shape inputs

Run1 was only run with Max Cycle = 10 to see the difference between each case of the experiment, since elapsed time of Run1 with Max Cycle = 1000 took much longer for each case than Run2. This is also the reason that the average accuracy for Run1 is much lower than Run2. The accuracy and the elapsed time of the Video Learning process for each angles, in which the shape was moving in these sets of videos were recorded as in the following tables:

Run1: Max Cycle = 10

Angle	36	72	108	144	180
Accuracy	0.73	0.61	0.73	0.75	0.64
Elapsed time	7min	5min	7min	6min	15min

Angle	216	252	288	324	360
Accuracy	0.65	0.69	0.69	0.64	0.6
Elapsed time	5min	6min	6min	7min	17min

Run2: Max Cycle = 1000

Angle	36	72	108	144	180
Accuracy	0.93	0.91	0.88	0.88	0.88
Elapsed time	7min	10min	7min	11min	11min

Angle	216	252	288	324	360
Accuracy	0.75	0.71	0.77	0.74	0.8
Elapsed time	5min	6min	6min	7min	12min

B. Resolution Experiments with Youtbe Video inputs

As shown below are the table results of the performance of the trained model, which also consists of the accuracy and the elapsed time of the learning process. However, for these experiments, Video's configuration[5] is slightly modified for the resolution section. This parameter represents the height and the width of the input video after it is binarized by the ColorMode and converted into frames for the learning process. The number of frames depends on the frameRate parameter.

Run1: Max Cycle = 10

Resolution	18x18	30x30	40x40	50x50	50x25
Accuracy	0.58	0.68	0.78	0.69	0.59
Elapsed time	4min	10min	21min	37min	16min

Run2: Max Cycle = 1000

Resolution	18x18	30x30	40x40	50x50	50x25
Accuracy	0.92	0.98	0.90	0.90	0.98
Elapsed time	6min	7min	12min	7min	8min

C. Resolution Experiments with Frankfurt Video inputs

Run1: Max Cycle = 10

Resolution	18x18	30x30	40x40	30x40
Accuracy	0.28	0.33	0.31	0.25
Elapsed time	19min	29min	46min	36min

Run2: Max Cycle = 1000

Resolution	18x18	30x30	40x40	30x40
Accuracy	0.86	0.5	0.43	0.82
Elapsed time	35min	45min	48min	50min

D. HTM Configuration Experiments with Youtube Video input

The final experiment was to test the Youtube Video with different settings in the HTM. As mentioned before, every single parameter is adjusted independently, while the others are maintained with the exact values as in the Default HTM configuration[6]. Therefore, the performance of the program is recorded specifically for each of the parameter in order to produce the comparison between the features in the HTM more clearly.

Cells per column:

Parameter	30	40	60	80	100
Accuracy	0.81	0.88	0.82	0.90	0.84

Elapsed time	6min	7min	12min	7min	8min

Switch Global Inhibition = false: Accuracy is 0.88 and Elapsed Time is 40min.

Enable LocalAreaDensity

Parameter	-1	0	1
Accuracy	0.85	0.91	N/A
Elapsed time	14min	22min	N/A

NumActiveColumnsPerInhArea with adjusted constant

Constant	0.01	0.02	0.03	0.04
Accuracy	0.95	0.86	1	1
Elapsed time	16min	10min	25min	30min

PotentialRadius

Constant	0.15	0.2	0.3	0.1
Accuracy	0.86	0.81	0.99	0.99
Elapsed time	10min	11min	32min	14min

Enable InhibitionRadius

Parameter	40	50	60	70	80
Accuracy	0.92	0.82	0.85	0.90	0.84
Elapsed time	18min	13min	21min	16min	15min

MaxBoost

Parameter	10	20	30	40	50
Accuracy	0.81	0.88	0.81	0.81	0.85
Elapsed time	6min	22min	20min	15min	10min

Enable DutyCyclePeriod

Parameter	50	60	70	80	90
Accuracy	0.86	0.89	0.89	0.81	0.96
Elapsed time	10min	38min	19min	12min	19min

Enable MinPctOverlapDutyCycles

Parameter	0.5	0.75	1.0	1.25	1.5
Accuracy	0.86	0.82	0.81	0.86	0.85
Elapsed time	10min	21min	20min	8min	13min

MaxSynapsesPerSegment with adjusted constant

Constant	0.02	0.03	0.04	0.05	0.1
Accuracy	0.82	0.88	0.84	0.81	0.82
Elapsed time	7min	13min	28min	12min	18min

Enable ActivationThreshold = 10: N/A

Enable ActivationThreshold = 20: N/A

->Only with ActivationThreshold = 15: Accuracy is 0.81 and Elapsed Time is 6min.

Enable ConnectedPermanence

Parameter	0.5	0.75	1.5
Accuracy	0.97	0.89	0.84
Elapsed time	25min	36min	27min

Enable PermanenceDecrement = 0.15: Accuracy is 0.84, Elapsed Time is 11min.

Enable PermanenceIncrement = 0.15: Accuracy is 0.81, Elapsed Time is 14min.

Both PermanenceIncrement and PermanenceDecrement

Parameters	0.15	0.2	0.25	0.3
Accuracy	0.96	0.97	0.95	0.95
Elapsed time	17min	31min	12min	13min

IV. Discussions

Overall, the experiments produced satisfactory results. Their accuracy has the mean value of approximately 0.88, with the lowest value was 0.81, which is decent. Noticeably, a number of modifications in certain parameters results in exceptional accuracy values, some of which were even able to reach 0.99. The elapsed time for these experiments varies from 4 minutes to 46 minutes. Unconventionally, it is observed that high accuracy was not always accompanied by the long elapsed time and vice versa.

Attached below is the list of the best results from the experiments of Run2 exclusively, with the accuracy of at least 0.9 under different settings of the video's resolution and the HTM configuration, since these parameters contribute directly to the performance of the learning process of the trained model. More over, after careful consideration, experiments with angles will not included in this list since these parameters are only relevant for these specific input videos. Generally, the results from the experiment must be applicable for every random video so that they can be implemented in the near future for the program in order to properly improve the quality of the Video Learning project.

- Resolution 18x18: 92% with elapsed time of 6 minutes.
- Resolution 30x30: 98% with elapsed time of 7 minutes.
- Resolution 40x40: 90% with elapsed time of 12 minutes.
- Resolution 50x50: 90% with elapsed time of 7 minutes.
- Resolution 50x25: 98% with elapsed time of 8 minutes.
- Cells per Column = 80: 90% with elapsed time of 7 minutes.
- .LocalAreaDensity = 0: 91% with elapsed time of 22 minutes.
- NumActiveColumnsPerInhArea with constant 0.01: 95% with elapsed time of 16 minutes.
- NumActiveColumnsPerInhArea with constant 0.03: 100% with elapsed time of 25 minutes.
- NumActiveColumnsPerInhArea with constant 0.04: 100% with elapsed time of 30 minutes.
- PotentialRadius = 0.1: 99% with elapsed time of 14 minutes.
- PotentialRadius = 0.3: 99% with elapsed time of 32 minutes.
- InhibitionRadius = 40: 92% with elapsed time of 18 minutes.
- InhibitionRadius = 70: 90% with elapsed time of 16 minutes.
- DutyCyclePeriod = 90: 96% with elapsed time of 19 minutes.
- ConnectedPermanence = 0.5: 97% with elapsed time of 25 minutes.
- PermanenceIncrement = PermanenceDecrement = 0.15: 96% with elapsed time of 17 minutes.
- PermanenceIncrement = PermanenceDecrement = 0.2: 97% with elapsed time of 31 minutes.
- PermanenceIncrement = PermanenceDecrement = 0.25: 95% with elapsed time of 12 minutes.
- PermanenceIncrement = PermanenceDecrement = 0.3: 95% with elapsed time of 13 minutes.

ACKNOWLEDGMENT

We would like to express our special thanks of gratitude to Professor Dobric for his patient guidance, enthusiastic encouragement and giving us the chance to do this project on the Video Learning, which also helped us to get to know about neuroscience and how spatial pooler designed for learning any pattern in iteration steps and generating the Sparse Distributed Representation (SDR) of the input. We would like also to extend our deepest thanks to Sabin and Toan for their valuable and constructive suggestions during the planning and development of this project.

REFERENCES

[1] https://en.wikipedia.org/wiki/Hierarchical_temporal_memory.

[2] Repository: neocortexapi\source\GroupSEProject\HTMVideoLearning\HTMVideoLearning.sln

[3] Repository: neocortexapi\source\GroupSEProject\HTMVideoLearning\VideoLibrary\AngleLibrary.

[4] Repository: Line17, neocortexapi\DataGeneration\circle.py.

[5] https://www.youtube.com/watch?v=dQw4w9WgXcQ

[6] https://www.youtube.com/watch?v=V2wofDTu48g&t=112s&ab_channel=Exploropia

[7] https://www.kapwing.com/studio/editor.

[8] https://express.adobe.com/tools/trim-video/#

[9] Repository: neocortexapi\source\GroupSEProject\HTMVideoLearning\VideoLibrary\YoutubeVideo\Video 1.

[10] Repository: Line43-47, neocortexapi\source\GroupSEProject\HTMVideoLearning\HTMVideoLearning\VideoLearning.cs.

[11] Repository: Line779-808, neocortexapi\source\GroupSEProject\HTMVideoLearning\HTMVideoLearning\VideoLearning.cs.