*Approve prediction for multisequence learning*

Poonam Dashrath Paraskar   
poonam.paraskar@stud.fra-uas.de

Pratik Prakash Desai  
pratik.desai@stud.fra-uas.de

Ankita Talande  
ankitatalande294@gmail.com

*Abstract*— Multisequence learning is the approach used to investigate implicit learning, where the model learns temporal patterns of sequences one by one during the experiment and provide the matching sequences as the predicted output. Model also predict the next element of the predicted sequence. List of sequences with double data-type are stored in an excel file, which worked as an input sequence file for model to train itself by storing these values in temporal memory. This experiment demonstrates how to learn sequences using Predictor's own method and save all the sequences as a dictionary object which is then considered as training data.

After that temporal memory will quickly learn cells for patterns and memorized the sequences. By using HtmClassifier's prediction method, where we must pass one test data sequence, by comparing each value of test data sequence with the trained data, model will predict the which sequence data is matching with test data, it also predicts the next element of that sequence. This experiment also focuses on calculating the accuracy of matched sequences and writing the resulted accuracy into CSV file.

In this experiment the additional implemented functionality helps user to pass both input sequences and test data sequences through excel file, which helps user to modify the sequence values directly through an external file instead of adding code changes

Keywords—: HtmClassifier, Temporal Memory, Prediction.

# **Introduction**

Hierarchical temporal memory (HTM) is a type of machine learning model inspired by the structure and

function of the neocortex in the human brain. Its purpose is to learn sequences of information and

make predictions based on them. The HTM model consists of several components, including an encoder, a spatial pooler, a temporal memory, and an HTM classifier. Each element in a sequence is represented by a Sparse Distributed Representation (SDR). The temporal memory learns the sequences of these SDRs and makes predictions about future input SDRs.

During the learning phase, the temporal memory predicts the next element in the sequence at every step.

After the last prediction, a learning cycle ends and the temporal memory starts again with the same sequence.

The goal is for every cycle to have correct predictions for every element, resulting in an accuracy of 100%.

The aim is to achieve 100% accuracy for as many consecutive cycles as possible.

Sequence-to-sequence learning is a general framework used for machine translation, where the encoder maps

the input sequence to a fixed-length vector.

# **Methods**

In the existing method inputs were hardcoded so if user wants to change the input, then it’s necessary to change the input sequences from the code. so, to resolve the issue we have tested different methods to take the inputs from the file.

## GetInputFromTextFile( ):

Team has implemented *GetInputFromTextFile()* method to take the inputs from the Text file. We have tried 2 approaches to split the multiple input sequences by using comma ‘,’ to separate each digit of the input sequence and using special character at the end of each sequence for splitting it from other input sequences. In this case we used semi-colon ‘;’ to split. The significant issue we faced by using this approach is we had to add both comma ‘,’ and semi-colon ‘;’ at the end of each input sequence, which is not a feasible solution and by which text file also looks inappropriate.

To resolve issue, we faced in the first approach we used regular expression to split multiple sequences based on detecting the enter ‘/r/n’. Using this approach wherever we added enter for next input, is getting detected by our regular expression logic. For this we had to read all the rows together using *reader.ReadToEnd( )* method and then split it by detecting the enter keyword. This can cause an issue in real time working environment.

## GetInputFromCsvFile( ):

Team has implemented *GetInputFromCsvFile( )* method to take the inputs from the CSV file. CSV stands for "Comma-Separated Values". It is a file format used for storing and exchanging tabular data, such as spreadsheets or databases. In a CSV file, each line represents a row of data and each field within a row is separated by a comma. CSV files are simple and widely supported, making them a popular choice for data exchange between different systems and applications.

The problem with CSV file is we need to add one non double character at end of each row to terminate the row/sequence and take the next sequence. This can cause an issue in real time working environment.

## GetInputFromExcelFile( ):

In the *GetInputFromExcelFile()* method we are using .xlsx file type to take the input sequences. Which are referred as training data sequences. Here we overcame the issues of the previous methods *GetInputFromCsvFile()* and *GetInputFromTextFile()* where we need to add any non-double value to terminate the row/sequence and to jump to the next row/sequence and any special in case of text file to jump over the next input sequence. To implement this feature we used the *string.IsNullOrWhiteSpace( )* property.

## GetSubSequencesInputFromExcelFile( ):

We have implmented the *GetSubSequencesInputFromExcelFile()* method to take the subsequence test input from the .xlsx file. We are passing the TestSubSequences to the SubSequences list of type double. After reading all the TestSubSequences we are returning SubSequences.

## Accuracy Logs:

We have used *StreamWriter*() class to create the file. If the file exists, it can be overwritten or appended to. If the file does not exist, this constructor creates a new file. The true flag appends to the file instead of overwriting it. Here we are generating logs for *sequenceKeyPair.Key* and *accuracy*. *Ex. Sequence: 1 is having accuracy 30%*

## Encoder Settings:

For encoder settings we have modified the value from 0-99. We have added input validation for the same in our *program.cs* file.

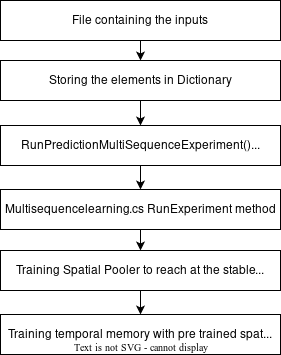
# **Implementation**

This part of the report explains stages of the experiment. This experiment broadly carried out into two stages.

1. Learning/Training phase
2. Prediction and Accuracy calculation phase.

**Learning phase:** In learning phase input data sequences are getting passed to *RunExperiment()* method. In *RunExperiment()* method training of input sequences is done using Cortex Layer, Spatial Pooler, Homeostatic Plasticity Controller which checks the stability of spatial pooler. Training of input sequences is required to get the stable state of Spatial pooler. Newborn cycles are generated for each input sequence till the time Spatial pooler reach the stable state. In newborn cycle, compute method of Cortex Layer is getting executed. Once Spatial pooler reaches to stable state the Temporal Memory algorithm is getting activated. At this stage, Spatial Pooler is trained completely. With pretrained SP and HPC, the TM learn cells for patterns.

The figure 1.1 explains how training phase is carried out:



*Figure 1.1 Training Phase*

**Prediction and Accuracy calculation phase:** *PredictNextElement()* method and Predictor class is used for prediction.

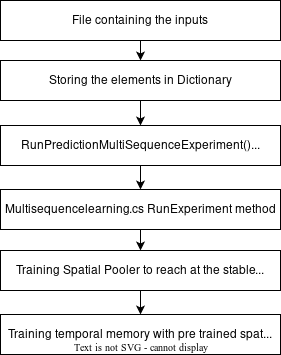
After the learning process, the algorithm returns the instance of Predictor class. This class provides *Predict()* method with a list of input elements.

For every presented input element, the predictor tries to predict the next element.

The more element provided in a sequence the predictor returns with the higher score then model produces a similarity matrix for all the classes.

After prediction of test sequence and next element, accuracy calculation is done.

The figure 1.2 explains how prediction phase is carried out:



*Figure 1.2 Prediction Phase*

# **Results**

This Part of the text describes results of your works. There can only be mentioned references, MUST point back to Methods and Intro chapter. No more external references.

Code examples must be provided to demonstrate how to use the algorithm/module. Provide a reference to more unit tests, which show the same in more detail. Also provide all diagrams with comments and reference to unit tests, which generate diagrams.

# Discussion

Conclusion of your work should be precise and concise. How was the project, what is done, what is the result... There can be discussion on further work and direction.

# Ease of Use

## Selecting a Template (Heading 2)

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the Microsoft Word, Letter file.

## Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

# Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

## Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

## Units

* Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
* Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
* Do not mix complete spellings and abbreviations of units: “Wb/m2” or “webers per square meter”, not “webers/m2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.

may use the solidus ( / ), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

*a**b* 

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

## Some Common Mistakes

* The word “data” is plural, not singular.
* The subscript for the permeability of vacuum **0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
* In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
* A graph within a graph is an “inset”, not an “insert”. The word alternatively is preferred to the word “alternately” (unless you really mean something that alternates).
* Do not use the word “essentially” to mean “approximately” or “effectively”.
* In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
* Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
* Do not confuse “imply” and “infer”.
* The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
* There is no period after the “et” in the Latin abbreviation “et al.”.
* The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

# Using the Template

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, three authors.** A minimum of one author is required for all report articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than three authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than three authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

For adding object other than text (tables, equations, graphs, figures, code…), **there must be at least one cross reference** to it. Figure 1 is an example

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)



Figure Example Figure Caption

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

## Code References:

Referencing Code in your text should be avoided unless necessary. In such cases it can be inserted as a listing as shown in **Error! Reference source not found.**

Listing Code Reference Example

Console.WriteLine(“Referencing code”, var);

// using tab can be replaced with 4 spaces

Do not pass code as image. When referring to variable in **Error! Reference source not found.**, italics should be used for example *var.* Code flows and logic should be presented better as Graph or Diagram instead of words.

Code Block which is too big to put in the textbox can be reference as Listing 2.

Listing Unit Test [EncodeDateTimeTest](https://github.com/ddobric/neocortexapi/blob/0348ffb99739ddf8c8c3a875f8162a18073938ca/source/UnitTestsProject/EncoderTests/DateTimeEncoderExperimentalTests.cs#L34-L49)

public void EncodeDateTimeTest(int w, double r, …)

{

…

DateTimeEncoderExperimental encoder = new…

var result = encoder.Encode(input);

…

Assert.IsTrue(result.SequenceEqual(expected…

}

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

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.

**This report template contains guidance text for composing and formatting technical reports. Please ensure that all template text is removed from your report prior to submission to the examination office. Failure to remove template text from your paper may result in your paper being degraded.**

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.