Scalar Encoder with Buckets

Sahith Kumar Singari  
[sahith.singari@stud.fra-uas.de](mailto:sahith.singari@stud.fra-uas.de)

Anil Kumar Gadiraju  
[anil.gadiraju@stud.fra-uas.de](mailto:anil.gadiraju@stud.fra-uas.d)

Vinay Kumar Bandaru

[vinay.bandaru@stud.fra-uas.de](mailto:vinay.bandaru@stud.fra-uas.de)

*Abstract*— In real-life situations, Scalar encoders with buckets are useful in a variety of real-world scenarios where continuous quantities must be represented as discrete values. This is frequently the case in machine learning and data analysis, when you may want to convert a continuous characteristic such as age, temperature, or height into a categorical feature that may be used in a model or algorithm. Sensor data analysis is one common application for scalar encoders with buckets. Encoders convert various data types into sparse distributed representations. They accept external inputs and convert them into a binary representation known by the CLA, similar to how the retina or cochlea convert external information into binary neuronal representations. This paper evaluates Investigate the implementation and performance of scalar encoding using buckets in AI and machine learning applications. We will examine the literature on best practices for determining the number and breadth of buckets, compare scalar encoding using buckets to other encoding approaches, and discuss the technique's drawbacks.

# **Introduction**

Using a scalar encoder with buckets, continuous data can be transformed into a set of discrete values that can then be utilized for analysis, modeling, or machine learning. With this method, the range of continuous values is divided into a number of discrete intervals, or "buckets," and each value is then assigned to the bucket that it belongs in. The encoding procedure can be accomplished in a variety of ways and customized to the unique requirements of the application, for as by employing a binary or multi-level encoding scheme. In applications including sensor data analysis, natural language processing, and picture classification, scalar encoder with buckets are frequently employed and have shown to be effective tools for transforming continuous data into discrete data.

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**

Sparse Distributed representations (SDRs) of input patterns are used in HTM's language. With a set amount of active bits, it produces SDRs internally. These bits have semantic value. As a result, two inputs with equivalent semantic meaning must have equal active bit representation in SDR, which plays an important role in HTM learning.

Hierarchical Temporal Memory (HTM) technique is based on the concept of SDRs, which are high-dimensional binary vectors with only a small fraction of the bits set to 1. SDRs are a natural way for the brain to represent patterns because they allow for the efficient storage and processing of large amounts of information.

SDR can be used in a wide range of applications using HTM systems. Consequently, the first step of using an HTM system is to convert a data source into an SDR using what we call an encoder. The encoder converts the native format of the data into an SDR that can be fed into an HTM system. The encoder is responsible for determining which output bits should be ones, and which should be zeros, for a given input value in such a way as to capture the important semantic characteristics of the data. Similar input values should produce highly overlapping SDRs.

Diagram

Description automatically generated

*Figure.1: Cochlear hair cells stimulate a set of neurons based on the frequency of the sound.*

## **Neocortex**

The neocortex is defined as the part of the cerebral cortex that serves 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 to 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.

## **Encoders**

Encoders should generate SDRs with a constant number of bits 'N' and a fixed number of active (1's) bits 'W', regardless of what they represent. What do you know about the ideal values for N and W?

To preserve the features of sparsity, we cannot be a large proportion of N. But, if W is too small, we lose the features of a distributed representation.

While encoding data, there are several unique aspects to consider:

1. Semantically related data can trigger SDRs with overlapping active bits.

2. The same input should always provide the same SDR output.

3. The result will have the same dimensionality as all of the inputs (total number of bits).

## **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.

# **Methodology**

The project Scalar Encoder with Buckets 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 Scalar Encoder which uses

## **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 of 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. 9: 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.

# **implementation**

# **RESULTS**

In this project,

# **Conclusion**

Multi Sequence learning for Sequence of Numbers which uses Neocortex API is used as a reference model to develop a solution for Multi Sequence learning - Sequence of Alphabets and Multi Sequence learning- Image data sets. HTM Prediction Engine was modified with different parameters to match the respective training process. The Sequence of Alphabets (Anticancer Peptide Sequence) Stored as a CSV file was modified and stored as an encoded value in the dictionary using Scalar Encoder and SDR input for the Training process. A prediction algorithm was developed to predict the trained sequences where the similarity matrix generated is compared with each of the SDRs of the Sequence learned during the training phase and based on the accuracy and observation class (Label), the Sequence is predicted.

HTM Image Encoder was incorporated to develop a solution that could train multiple Image data sets and a prediction algorithm that could predict input images. The HTM Image Encoder binarizes the input image and stores as array elements of zeros and ones used as SDR Input for training. Similar to the Prediction algorithm for Sequence of Numbers and Alphabets, the Prediction of Image algorithm was developed, and the input image was predicted by comparing with the trained data sets and returning the prediction output based on accuracy and Observation class (Label).

We performed Multi Sequence Learning for a different sequence of data sets and could achieve up to 87.5% of accuracy in the Training Phase.

The experiments carried out helped us understand different types of encoders, such as scalar encoders and HTM Image encoders, how the Spatial pooler creates SDR inputs and computes the learning phase, and how the Homeostatic Plasticity controller helps in stabilizing the learning phase in NeoCortex API.

# **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. |