

# *Improve Samples and Documentation for SDR representation*

Md Tanzeem Hasan Mahmud  
[mahmud.mdtanzeemhasan@stud.fra-  
uas.de](mailto:mahmud.mdtanzeemhasan@stud.fra-<br/>uas.de)

Khan Mushfiquir Rahman  
[khan.rahman@stud.fra-uas.de](mailto:khan.rahman@stud.fra-uas.de)

**Abstract**—This paper presents a method for visualizing Sparse Distributed Representations (SDRs) as bitmap images, leveraging the DrawBitmap utility within Hierarchical Temporal Memory (HTM) systems. SDRs, akin to the brain's data processing technique, offer a robust and efficient way to handle diverse information types. By converting SDRs into bitmap images, we can visually interpret the encoded patterns and assess the functionality of encoders and spatial pooling processes in HTM models. This approach facilitates a deeper understanding of HTM's capabilities in pattern recognition and data encoding.

**Keywords**—HTM, SDR, Bitmap, DrawBitMap

## I. INTRODUCTION

Sparse Distributed Representations (SDRs) [1] are pivotal in computational models that mimic the human brain's processing, such as Hierarchical Temporal Memory (HTM) [2]. Representing data in a sparse and distributed manner allows for efficient and robust information processing. However, understanding and interpreting the complex patterns encoded within SDRs can be challenging. This paper introduces a visualization method that transforms SDRs into bitmap images, providing an intuitive way to analyze and comprehend the information encoded by HTM systems. Through the customization of DrawBitMap [3] methods parameters such as dimensions, colors, and additional text, this approach allows users to personalize visualizations to suit their unique analysis requirements, promoting a more profound comprehension of SDR properties and behaviors. This paper demonstrates the practicality and flexibility of bitmap representations in understanding the complexities of SDRs, using examples from Date Time, Scalar, and Geospatial Encoders, Binary Encoders, Spatial Pooler etc.

## II. METHODS

In this section, we will describe the approach we followed to improve samples and documentation for SDR representation. Our objective was to visualize Sparse Distributed Representations (SDRs) as bitmap images, leveraging the DrawBitmap utility within Hierarchical Temporal Memory (HTM) systems. To achieve this, we performed DrawBitmap method. The DrawBitmap method transforms a two-dimensional array representing an SDR into a visual bitmap image. By specifying the dimensions, colours, and additional text, users can customize the visualization to suit their analysis needs. The method scales

the SDR array to fit the specified bitmap dimensions, allowing for a clear and adjustable representation of the SDR's structure.

Turning an SDR into a visual bitmap involves a few straightforward steps:

1. *Setting the Scene:*
  - Determine the width and height for the bitmap, scaling the image to fit your needs.
  - Choose colors for the active and inactive cells to make the SDR's structure clear.
2. *Calculating the Scale:*
  - A scale factor is calculated based on the ratio of the bitmap's width to the SDR array's width. This helps adjust the cell sizes in the bitmap to fit the entire SDR.
3. *Drawing the Bitmap:*
  - Go through each cell in the SDR:
    - Color it with the active cell color if it's active (1).
    - Use the inactive cell color if it's inactive (0).
  - The scale factor ensures each cell in the bitmap represents the SDR accurately.
4. *Saving the drawn bitmaps:*
  - Once every cell is colored, the bitmap is saved to the location specified in file Path.
  - This method simplifies analyzing and understanding SDR patterns by providing a visual representation.

Our methodology encompasses several scenarios, including basic SDR examples, encoder generated SDRs, and SDRs resulting from spatial pooling processes. We discuss the settings and configurations for various encoders, such as Scalar and Geospatial Encoders, and how they contribute to generating meaningful SDRs for visualization.

Here, we are presenting some examples that will help to understand how it works,

#### A. Basic SDR Examples with binary encoders

- Example of visualizing a number: This simple example will help to understand the visualization process, taking a simple value like 40148. Now, we must encode this data to visualize with the help of SDR. Here is the full example [4].

Then, we can get the below image from the DrawBitMap method,



Fig 1: Example of visualizing a number 40148

- Another example of visualizing a number: For this example, we will take a random integer value 50149. We will encode the value by setting up the SDR encoder settings and encode the value with those settings. and set up the two-dimensional array. Here is the full example [5].

It returns the below image,



Fig 2: Example of visualizing an integer number 50149

#### B. DrawBitMap example with Binary Encoder 1D image

We can also use DrawBitMap method for generating 1D images by getting the binary encoded value from our input. Here is the full example [6] and The Draw1DBitMap method can be found [7].



Fig 3: Example of DrawBitMap with Binary Encoder 1D image

#### C. DrawBitmap example for DateTime Encoder

A DateTime encoder is a type of encoder that transforms datetime information such as dates and times into Sparse Distributed Representations (SDRs). We can make and use different types of encoders to visualize that particular type of data in order to visualize them. We can take datetime data and encode them with DateTime encoder and visualize them with DrawBitMaps method. For this example, we can

take "08/03/2024 21:58:07" and send it through datetime encoder to get SDR. The full example can be found [8].

The generated image is this,



Fig 4: DrawBitmap example for DateTime Encoder

#### D. Drawing AQI Values with Scalar Encoder

The Scalar Encoder converts AQI levels into SDRs, capturing the essence of air quality in a binary format. For instance, the AQI levels are segmented into: 0-49: Good 50-149: Moderate 150-249: Unhealthy for Sensitive Groups 250-349: Unhealthy 350-449: Very Unhealthy 450-500: Hazardous Generating Bitmaps To visualize the encoded AQI values. The full example can be found here [9].

The generated bitmap are as follows:

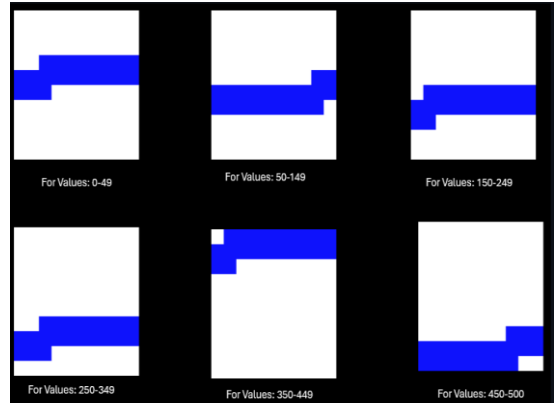


Fig 5: Drawing AQI Values with Scalar Encoder

#### E. DrawBitmap sample for Geospatial Encoder

Geospatial encoder is used to encode latitude or longitude values into Sparse Distributed Representations (SDRs) for Hierarchical Temporal Memory (HTM) systems. In the exploration of geospatial data through Sparse Distributed Representations (SDRs), we utilize the DrawBitmap method to translate encoded geographical coordinates into visually interpretable bitmap images. This approach allows for the visualization of spatial information encoded within SDRs, offering insights into the encoded geographical regions. The full example can be found here [10].

The generated bitmap are as follows:



Fig 6: DrawBitmap sample for Geospatial Encoder

The bitmap images generated for geographical coordinates offer a unique view of the spatial patterns encoded within the SDRs. Now if we change the encoder settings and provide the below settings: The bitmap images generated for geographical coordinates offer a unique view of the spatial patterns encoded within the SDRs. Now if we change the encoder settings and provide the below settings:

```
encoderSettings.Add("W", 21);
```

```
encoderSettings.Add("N", 40);
```

The output will be different for the same value. The bitmaps generated in this case are:

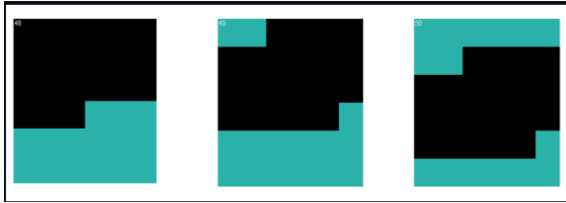


Fig 7: DrawBitmap sample for Geospatial Encoder after encoder changed.

#### F. Bitmap representation of Image using Spatial Pooler

In the context of Hierarchical Temporal Memory (HTM) theory, the Spatial Pooler (SP) plays a crucial role in transforming input data into Sparse Distributed Representations (SDRs). These representations capture the essential features of the input data in a way that emphasizes structural and semantic similarities. To visually understand the transformation process and the output of the Spatial Pooler, we can represent SDRs as bitmaps. This example elaborates on how active arrays generated by the SpatialPooler are converted into bitmap images, further enhancing our understanding of HTM's processing capabilities. The full example can be found here [11].

The DrawBitmaps utility function generates bitmap images that visually represent the active columns as a result of the SP's processing. The generated Images looks like this.

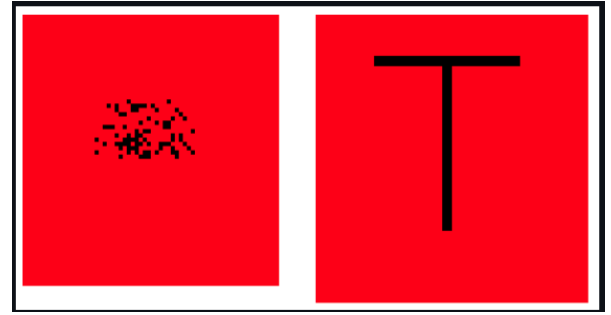


Fig 8: Bitmap representation of Image using Spatial Pooler

On the right side we can see the images generated from the binarized file of the input training image, on the left we have the SDR representation by SpatialPooler after feeding the training image.

Example representing Overlap (Intersection), Difference and Union for Alphabet T and Numeric 3 in Bitmap after computing in spatial pooler:

Here we can see the Alphabet T and Numeric 3

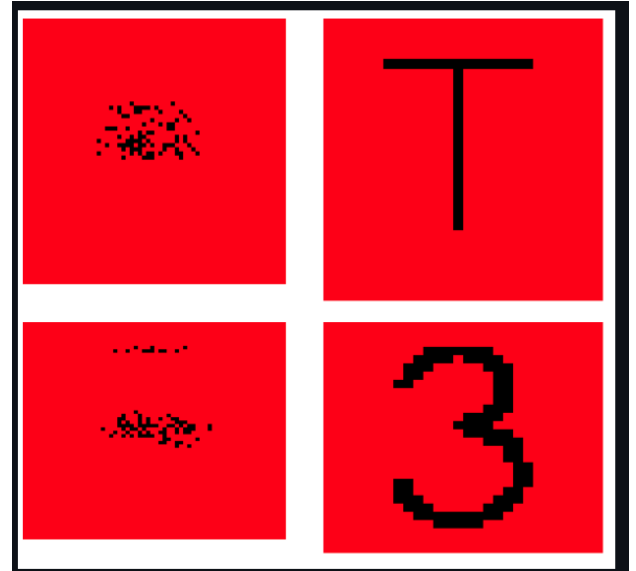


Fig 9: Example representing Overlap (Intersection), Difference and Union for Alphabet T and Numeric 3 in Bitmap

Below is the representation for Overlap, Difference and Union:

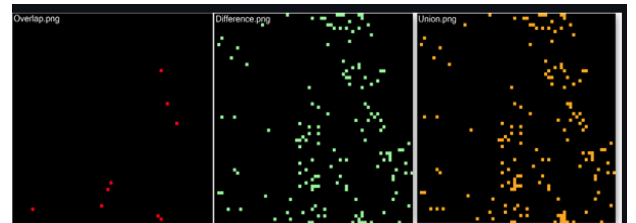


Fig 10: Representation for Overlap, Difference and Union

Two SDRs are compared with the help of UnionSDRfun(), DiffSDRfun() and OverlapSDRfun() functions. The overlap.png shows very few intersections as

the SDRs are very different from each other. Basically, the combination of Overlap and Difference gives us Union.

Example representing Overlap (Intersection), Difference and Union for Alphabet T and 1 in Bitmap after computing in spatial pooler.

Generated Bitmap representation of T and 1 is shown below:

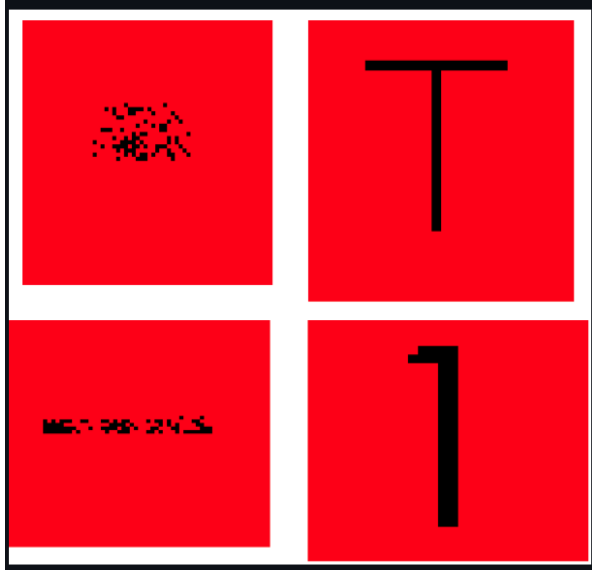


Fig 11: Bitmap representation of T and 1

Below is the representation for Overlap, Difference and Union:

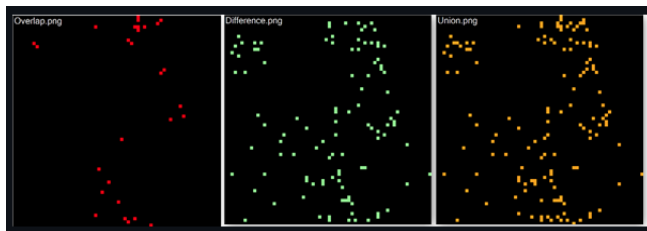


Fig 12: Representation for Overlap, Difference and Union

On comparison between two SDRs, the overlap.png shows more overlaps/intersections in comparison to the above example. SDRs are similar to each other.

### III. RESULTS

The paper presents a series of bitmap images generated from encoded SDRs, showcasing how different data types, including numbers, geographical coordinates, and temporal information, can be visually represented. We illustrate the efficacy of the DrawBitmap method in highlighting the active and inactive bits within SDRs, providing insights into the encoding mechanisms of HTM models. Additionally, we explore the visualization of SDRs generated by the spatial pooling process, demonstrating how input patterns are captured and represented in the model.

### IV. DISCUSSION

In conclusion, our project aims to interpret the encoded patterns and assess the functionality of encoders and spatial

pooling processes in HTM models by converting SDRs into bitmap images.

Visualizing SDRs as bitmap images offers a powerful tool for understanding and interpreting the complex patterns encoded in HTM systems. This approach not only aids in the analysis of HTM models but also paves the way for developing more intuitive interfaces for interacting with these models. Overall, our project is able to generate new examples and samples and all together a new guidance to run our experiment with BitMap generation of SDRs whether it be 1D or 2D.

### REFERENCES

- [1] Sparse Distributed Representation-numenta(no date) [www.numenta.com](http://www.numenta.com) Available at: <https://www.numenta.com/assets/pdf/biological-and-machine-intelligence/BaMI-SDR.pdf> (Accessed: March 22, 2024).
- [2] Taylor, M. (2016) HTM School, YouTube. YouTube. Available at: <https://www.youtube.com/playlist?list=PL3yXMgtrZmDqhsFQzwUC9V8MeeVOQ7eZ9> (Accessed: March 22, 2024).
- [3] D. Dobric, et al, DrawBitMap, 2023, [Source Code](#)
- [4] MD. T. H. Mahmud, K.M. Rahman, EncodeAndVisualizeSingleValueTest1, [Source Code](#)
- [5] MD. T. H. Mahmud, K.M. Rahman, EncodeAndVisualizeSingleValueTest2, [Source Code](#)
- [6] MD. T. H. Mahmud, K.M. Rahman, EncodeAndVisualizeSingleValueTest3, [Source Code](#)
- [7] MD. T. H. Mahmud, K.M. Rahman, Draw1DBitMap, [Source Code](#)
- [8] MD. T. H. Mahmud, K.M. Rahman, DateTimeEncoderExample, [Source Code](#)
- [9] MD. T. H. Mahmud, K.M. Rahman, ScalarEncoderExample, [Source Code](#)
- [10] MD. T. H. Mahmud, K.M. Rahman, GeoSpatialEncoderExample, [Source Code](#)
- [11] MD. T. H. Mahmud, K.M. Rahman, SpatialSpatialEncoderExample, [Source Code](#)