**Scalar Encoder with Buckets**

**Names of** [Team\_ScalarEncoder](https://github.com/UniversityOfAppliedSciencesFrankfurt/se-cloud-2022-2023/tree/Team_ScalarEncoder) **members:**

* Aqib Javed
* Haris Abbas Qureshi
* Shahab Uddin
* Saad Jamil

getBucketIndices(*inputData*)

Returns an array containing the sub-field bucket indices for each sub-field of the inputData. To get the associated field names for each of the buckets, call getScalarNames().

|  |  |
| --- | --- |
| Parameters: | inputData – The data from the source. This is typically a object with members. |
| Returns: | array of bucket indices |

getBucketInfo(*buckets*)

Returns a list of EncoderResult namedtuples describing the inputs for each sub-field that correspond to the bucket indices passed in ‘buckets’. To get the associated field names for each of the values, call getScalarNames().

|  |  |
| --- | --- |
| Parameters: | buckets – The list of bucket indices, one for each sub-field encoder. These bucket indices for example may have been retrieved from the getBucketIndices() call. |

**Hierarchical Temporal Memory**

Encoder -

Encoder is chosen according to the type of the inputs. There are some encoders available for popular input type:

- Scalar Encoder

- Datetime Encoder

- Boolean Encoder

- Category Encoder

- Geo-Spatial Encoder

In this project we are using - Scalar Encoder

Scalar Encoder is one of the encoding techniques and is a part of Hierarchical Temporal Memory (HTM). HTM is a machine intelligence technology which is trying to imitate the process and architecture of neocortex. The main purpose for scalar encoder is to encode numeric or floating-point value into an array of bits, where the output has 0’s with an adjacent block of 1’s. The location of the block of 1’s varies continuously depending on the input value.

The scalar representation of value (e.g. for categories, this is the internal index used by the encoder). This number is consistent with what is returned by **[getScalars()](https://nupic.docs.numenta.org/1.0.3/api/algorithms/encoders.html" \l "nupic.encoders.base.Encoder.getScalars" \o "nupic.encoders.base.Encoder.getScalars)**. This value is always an int or float, and can be used for numeric comparisons.

HTM consists of 2 different components: Spatial Pooler and Temporal Memory.

1. Spatial Pooler -

Encoder produces output to be fed into Spatial Pooler algorithm. Type of Spatial Pooler (SP) that is used in this example is the multithreaded version that utilize multicore of the machine to run the spatial pooler algorithm.

SpatialPoolerMT spatialPooler = new SpatialPoolerMT(hpa);

patialPooler.Init(memory, UnitTestHelpers.GetMemory());

1. Temporal Memory -

The output of Spatial Pooler (SDR) is used as the input of Temporal Memory.Temporal memory algorithm will then learn the temporal pattern from spatial pattern.

TemporalMemory temporalMemory = new TemporalMemory();

temporalMemory.Init(mem);