Approve Prediction of Multisequence Learning

Multisequence Learning

In the endeavor to implement multi-sequence learning with HTM, the initial step involves encoding input data into Sparse Distributed Representations (SDRs) using a scalar encoder. These SDRs are then processed by the spatial pooler, generating sparse representations of the input sequences. Subsequently, these representations are fed into the temporal memory component for learning and prediction. This approach is highly effective for recognizing and predicting patterns across multiple input sequences.

In my project, I have introduced novel methods to enhance the Multi-sequence Learning algorithm . These methods facilitate automatic dataset retrieval from a specified location. Additionally, we've incorporated separate test data, which will be utilized for evaluating subsequences during testing. The Multi-sequence Learning algorithm operates by analyzing multiple sequences and testing subsequences for learning purposes. Once learning is finalized, the accuracy of predicted elements is computed for evaluation.

## Hierarchical Temporal Memory

The objective of HTM is to emulate the hierarchical structure and learning process of the brain. It is comprised of a network of nodes organized hierarchically, where each node corresponds to a group of neurons in the neocortex. These nodes acquire the ability to identify patterns in sensory information and generate predictions based on their prior experiences. Subsequently, the predictions are evaluated against the input data to refine the node's model and enhance its predictive precision.

As per Hawkins, the neocortex learns and makes predictions by forming a hierarchical structure of columns, each containing a set of neurons that recognize patterns in sensory input. These columns communicate with each other in a hierarchical manner, with higher-level columns representing more abstract concepts.

## Sparse Distributed Representation

In the context of HTM, SDRs are used to represent patterns of activity in the network. Each input to the network is transformed into an SDR, which is then processed by the network's hierarchy of nodes to make predictions about future input.

Hawkins and Ahmad proposed that SDRs, which are binary vectors with a small number of active bits (ones) out of a large number of total bits, are a natural way to represent sparse, distributed patterns of activity in the neocortex.

## Encoder

The encoder is designed to take in raw data inputs and convert them into SDRs that represent the relevant features of the input data. The encoding process involves several steps, including dimensionality reduction, noise reduction, and normalization. The encoder also learns to recognize patterns in the input data and adapts to changes in the input distribution over time.

The SP encoder is designed to handle temporal data and uses a sliding window approach to capture temporal patterns in the input data. It first converts the input data into a continuous stream of binary values, which are then fed into the HTM network as a sequence of SDRs [5].

## Spatial Pooler

The Spatial Pooler is a type of unsupervised learning algorithm that takes in high-dimensional input data and creates a lower-dimensional SDR that represents the relevant features of the input data. It does this by first assigning a random set of weights to each input feature and then computing the overlap between each input and the set of weights. The features with the highest overlap are then selected and included in the SDR [6].

## Temporal Memory

The functioning of Temporal Memory involves preserving a collection of active cells that embody the present context of the input data. Upon receiving fresh input patterns, the active cells undergo modifications according to the similarity between the input and the current context. Additionally, the algorithm manages a set of connections between cells that represent the sequence of patterns that were encountered previously [6].

Using these connections, the Temporal Memory can predict the next likely input pattern based on the current context. If the prediction is correct, the algorithm reinforces the connections between the cells that were active during the predicted sequence. If the prediction is incorrect, the algorithm adjusts the connections to reduce the likelihood of that sequence occurring in the future.

## Multisequence Learning

Multisequence learning is a HTM based algorithm that involves learning and predicting multiple sequences of patterns simultaneously. Multisequence learning is achieved by using separate Temporal Memory modules to learn and predict each sequence of patterns.

The key idea behind this approach by Ahmad [7] is to use a hierarchical structure of nodes to learn and predict sequences of patterns at different levels of abstraction. At the lowest level, each Temporal Memory module learns and predicts the raw sensory input from a single modality. At higher levels, the nodes learn and predict sequences of patterns that combine information from multiple modalities.

## Sequence

Sequence is data model which consists of sequence name and numeric sequence. The object of list of sequence data model is nothing but multiple sequences. The subsequence is subset of sequence and hence used same data model.

public class Sequence

{

public String name { get; set; }

public int[] data { get; set; }

}