Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Effect of Spatial Pooler Initialization on Column Activity in Hierarchical Temporal Memory

Mackenzie Leake
Scripps College
Claremont, CA
mleake8625@scrippscollege.edu

Liyu Xia
University of Chicago
Chicago, IL
jimmyxia@uchicago.edu

Kamil Rocki
IBM Research
San Jose, CA
kmrocki@us.ibm.com

Wayne Imaino
IBM Research
San Jose, CA
wayne1@us.ibm.com

Abstract

In the Hierarchical Temporal Memory (HTM) paradigm the effect of overlap between inputs on the activation of columns in the spatial pooler is studied. Numerical results suggest that similar inputs are represented by similar sets of columns and dissimilar inputs are represented by dissimilar sets of columns. It is shown that the spatial pooler produces these results under certain conditions for the connectivity and proximal thresholds at initialization. Qualitative arguments about the learning dynamics of the spatial pooler are then discussed.

Introduction

Inspired by the importance of time-dependent behavior and prediction in the brain, Hierarchical Temporal Memory (HTM) (Hawkins, Ahmad, and Dubinsky 2011), a machine learning paradigm, enables real-time learning of sequences and demonstrates predictive capabilities. Prior works have explored the use of HTM for pattern recognition (Maltoni 2011), speech-based learning tasks (van Doremalen and Boves 2008), and stock trading (Gabrielsson, Konig, and Johansson 2013). However, the mathematics of HTM has not yet been studied widely.

As outlined in the HTM white paper (Hawkins, Ahmad, and Dubinsky 2011), the spatial pooler, the first phase of HTM, demonstrates efficient dimensionality reduction and adaptation. The spatial pooler comprises a set of *columns*, analogous to biological cortical columns, the fundamental units of computation in the neocortex (Mountcastle 1997). Each *column* has a set of synapses. Each synapse has an associated permanence value, the magnitude of which decides whether the synapse is connected.

The spatial pooler is responsible for converting each input into an internal representation called a sparse distributed representation (SDR), which is a set of *columns* that are activated by the input vector. In order to group similar inputs and distinguish between differing inputs, the spatial pooler should exhibit the following behavior: similar inputs map to similar SDR's, and dissimilar inputs map to dissimilar SDR's. The quality of these SDR's is critical to the later predictive and temporal HTM phases, which predict and recognize sequences of SDR's.

We consider inputs to be either overlapping or non-overlapping. We use "overlapping" to describe inputs whose binary representations share at least one on-bit and "non-overlapping" to describe inputs whose binary representations share no on-bits. Here we study the specific initial conditions for parameters to pool overlapping inputs into similar sets of active *columns* and non-overlapping inputs into dissimilar sets of active *columns*.

Formulation / Notation

We use the notation below to describe the status and behavior of the spatial pooler. In our formulation synaptic activity for *columns* is stored in a permanence matrix, P. Each row of P is a vector of permanence values associated with a certain *column* in the spatial pooler. For example, the value at P_{ij} denotes the strength of the connection between the i-th *column* and the j-th bit of the binary input vector. Another matrix called the connectivity matrix is a binary version of P. This connectivity matrix denoted by C can be described by $C = P > \tau_C$, where τ_C is the synaptic threshold. This notation indicates entry C_{ij} is set to 1 if $P_{ij} > \tau_C$ and 0 otherwise. We use C_j, the j-th row of C, to denote the connectivity of the j-th *column* in the spatial pooler.

If we consider the spatial pooler to be a function that maps input vectors to SDR's, the function can be formally characterized as the following:

$$\texttt{ov} = Cx$$
$$c = \texttt{ov} > \tau_o$$

where x is the input vector, $\texttt{ov}$ is the vector of overlap scores between the input x and C, and τ_o is the proximal threshold. Here c is the SDR of x, and the proximal threshold, τ_o, determines whether there is sufficient overlap between an input x and a particular *column* C_j to activate C_j.

Initialization

From our numerical experiments we find the initialization of parameters influences the learning dynamics of the spatial pooler once inputs are introduced. Therefore, it is useful to find potential restrictions on the parameters upon initialization so that certain intended behavior is achieved. One intended behavior of the spatial pooler is to map non-overlapping inputs to distinct *columns*. Using probabilistic methods we can derive a relationship between the connectivity threshold, τ_C, the proximal threshold, τ_o, and the number of on-bits, d, in the input vector.

Assume we have two non-overlapping inputs, x^a and x^b. Assume x^a activates *column* C_j, i.e. ov_{ac} is greater than τ_o, and assume all inputs are equally likely. Without loss of generality, we assume $x^a = [1, \ldots, 1, 0, \ldots, 0]$. Denote these assumptions as event Z. Using the linearity of expectation, we compute the expected value, E_n, for the overlap between x^b and C_j given Z. This gives:

$$E_n = \mathbb{E}[\sum_{k=1}^{L} x_k^b C_{jk} | Z] = \sum_{k=1}^{L} \mathbb{P}[x_k^b = 1, C_{jk} = 1 | Z]$$
$$= \sum_{k=d+1}^{L} \mathbb{P}[x_k^b = 1, C_{jk} = 1 | Z].$$

Using Bayes' theorem we compute:

$$E_n = \sum_{k=d+1}^{L} \mathbb{P}[x_k^b = 1 | C_{jk} = 1, Z] \times \mathbb{P}[C_{jk} = 1 | Z].$$

Considering all possible x^b, the probability for each entry of x^b to be 1 is $\frac{d}{L-d}$ resulting in:

$$E_n = \sum_{k=d+1}^{L} \frac{d}{L-d} \times \mathbb{P}[C_{jk} = 1 | Z] = d \times \mathbb{P}[C_{jk} = 1 | Z].$$

The threshold for connectivity is τ_C so the expected number of 1's within a row of the connectivity matrix is $L(1-\tau_C)$. The number of 1's in the first d entries of *column* C_j is equal to $\mathtt{ov}_{\mathtt{ac}}$ because we assumed $x^a = [1, \ldots, 1, 0, \ldots, 0]$. Therefore, the number of 1's left for *column* C_j to overlap with x^b is equal to $L(1-\tau_C) - \mathtt{ov}_{\mathtt{ac}}$. This results in:

$$E_n = d \times \frac{L(1-\tau_C) - \mathtt{ov}_{\mathtt{ac}}}{L-d}.$$

The proximal threshold τ_o can provide a loose upper bound for E_n to ensure on average x^b will not activate C_j. We can substitute $\mathtt{ov}_{\mathtt{ac}}$ by τ_o, since x^b should not activate C_j as long as x^a activates C_j. We then find:

$$d \times \frac{L(1-\tau_C) - \tau_o}{L-d} \leq \tau_o$$
$$1 - \tau_C \leq \frac{1}{d}\tau_o.$$

This preceding relationship provides guidelines for the initialization of τ_C and τ_o that on average will prohibit input x^b from activating *column* C_j. We can also perform a similar analysis for two overlapping inputs to obtain numerical relationships between parameters upon initialization.

Incremental learning

Moving beyond the initialization phase, we can make observations about the interaction between non-overlapping inputs and the spatial pooler during the learning phase. If the parameters are initialized such that each non-overlapping input activates a distinct *column*, we claim this behavior will continue during the learning phase. Suppose input x^a activates *column* C_j and no other input activates C_j initially. Once learning starts, x^a will readily activate C_j, and $\mathtt{ov}_{\mathtt{ac}}$ will increase incrementally. However, another non-overlapping input x^b will not activate C_j. Thus, x^a will continue to activate C_j during the learning phase, while all other inputs will have no effect on C_j. We can extend this reasoning to all other non-overlapping inputs and *columns* to ensure non-overlapping inputs activate distinct columns.

We can also make qualitative arguments about the behavior of overlapping inputs during the learning phase. Let us examine a specific example for two overlapping inputs. Suppose we have two inputs, x^a and x^b, which have sufficient overlap to activate the same *column* C_j, and suppose the values of P corresponding to this column are not so close to the connectivity threshold, τ_C, that one update of P will result in a change of C. If x^a and x^b are the only inputs and they are presented to the spatial pooler the same number of times in an alternating fashion, C_j will reinforce the common parts of x^a and x^b. During learning when x^a activates C_j, the entries in P that are associated with C_j and correspond to the on-bits in x^a will be incremented, while the entries of P that are associated with C_j and are unique to x^b will be decremented. Similar changes in P will occur when x^b activates C_j. The entries of P corresponding to the shared on-bits of x^a and x^b will be incremented twice, while the entries of P associated with the on-bits that are unique to either x^a or x^b will be incremented for one input and decremented for the other. This results in the reinforcement of the common parts of x^a and x^b. Under these assumptions we see a mechanism for how the spatial pooler groups similar inputs.

Summary

We show that with careful initialization of parameters the spatial pooler maps distinct inputs into distinct SDR's as indicated by the *column* activity. Assuming appropriate parameter selection, we make observations about the learning dynamics for both non-overlapping and overlapping inputs.

References

Gabrielsson, P.; Konig, R.; and Johansson, U. 2013. Evolving hierarchical temporal memory-based trading models. In *EvoApplications 2013*, 213–222.

Hawkins, J.; Ahmad, S.; and Dubinsky, D. 2011. Hierarchical temporal memory including HTM cortical learning algorithms. Technical Report 0.2.1, Numenta, Redwood City, Calif.

Maltoni, D. 2011. Pattern recognition by hierarchical temporal memory. Technical report, DEIS Univ. of Bologna.

Mountcastle, V. 1997. The columnar organization of the neocortex. *Brain* 120(4):701–722.

van Doremalen, J., and Boves, L. 2008. Spoken digit recognition using a hierarchical temporal memory. In *Interspeech*, 2566–2569.