bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

The HTM Spatial Pooler – a neocortical algorithm for online sparse distributed coding

Yuwei Cui*, Subutai Ahmad, and Jeff Hawkins
Numenta, Inc,
Redwood City, California, USA

*Corresponding author
Emails: ycui@numenta.com, sahmad@numenta.com, jhawkins@numenta.com

A version of this manuscript has been submitted for publication on Feb 3, 2017. This is a draft preprint. The final paper will be different from this version.

We welcome comments. Please contact the authors with any questions or comments.

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

The HTM Spatial Pooler – a neocortical algorithm for online sparse distributed coding

Yuwei Cui, Subutai Ahmad, and Jeff Hawkins
Numenta, Inc, Redwood City, California, United States of America

1. Abstract

Hierarchical temporal memory (HTM) provides a theoretical framework that models several key computational principles of the neocortex. In this paper we analyze an important component of HTM, the HTM spatial pooler (SP). The SP models how neurons learn feedforward connections and form efficient representations of the input. It converts arbitrary binary input patterns into sparse distributed representations (SDRs) using a combination of competitive Hebbian learning rules and homeostatic excitability control. We describe a number of key properties of the spatial pooler, including fast adaptation to changing input statistics, improved noise robustness through learning, efficient use of cells and robustness to cell death. In order to quantify these properties we develop a set of metrics that can be directly computed from the spatial pooler outputs. We show how the properties are met using these metrics and targeted artificial simulations. We then demonstrate the value of the spatial pooler in a complete end-to-end real-world HTM system. We discuss the relationship with neuroscience and previous studies of sparse coding. The HTM spatial pooler represents a neurally inspired algorithm for learning sparse representations from noisy data streams in an online fashion.

2. Introduction

Our brain continuously receives vast amounts of information about the external world through peripheral sensors that transform changes in light luminance, sound pressure, and skin deformations into millions of spike trains. Each cortical neuron has to make sense of a flood of time-varying inputs by forming synaptic connections to a subset of the presynaptic neurons. The collective activation pattern of populations of neurons contributes to our perception and behavior. A central problem in neuroscience is to understand how individual cortical neurons learn to respond to specific input spike patterns, and how a population of neurons collectively represents features of the inputs in a flexible, dynamic, yet robust way.

Hierarchical temporal memory (HTM) is a theoretical framework that models a number of structural and algorithmic properties of the neocortex (Hawkins et al., 2011). HTM networks can learn time-based sequences in a continuous online fashion using realistic neuron models that incorporate nonlinear active dendrites (Antic et al., 2010; Major et al., 2013) with thousands of synapses (Hawkins and Ahmad, 2016). When applied to streaming data, HTM networks achieve state of the art performance on anomaly detection (Lavin and Ahmad, 2015; Ahmad and Purdy, 2016) and sequence prediction tasks (Cui et al., 2016a).

The success of HTM relies on the use of sparse distributed representations (SDRs) (Ahmad and Hawkins, 2016). Such sparse codes represent a favorable compromise between local codes and dense codes (Földiák, 2002). It allows simultaneous representation of distinct items with little interference, while still has a large representational capacity (Kanerva, 1988; Ahmad and Hawkins, 2015). Existence of SDRs have been documented in auditory, visual and somatosensory cortical areas (Vinje and Gallant, 2000; Weliky et al., 2003; Hromádka et al., 2008; Crochet et al., 2011). HTM spatial pooler (SP) is a key component of HTM networks that continuously encodes streams of sensory inputs into SDRs. Originally described in (Hawkins et al., 2011), the term "spatial pooler" is used because input patterns that share a large number of co-active neurons (i.e. that are spatially similar) are grouped together into a common output representation. Recently there has been increasing interest in the mathematical properties of the HTM spatial pooler (Pietroń et al., 2016; Mnatzaganian et al., 2017) and machine learning applications based on it (Thornton and Srbic, 2011; Ibrayev et al., 2016). In this paper we explore several functional properties of the HTM spatial pooler that have not yet been systematically analyzed.

The HTM spatial pooler incorporates several computational principles of the cortex. It relies on competitive Hebbian learning (Hebb, 1949), homeostatic excitability control (Davis, 2006), topology of connections in sensory cortices (Udin and Fawcett, 1988; Kaas, 1997) and activity-dependent structural plasticity (Zito and Svoboda, 2002). The HTM spatial pooler is designed to achieve a set of computational properties that support further downstream computations with SDRs. These properties include (1) preserving the semantic similarity of the input space by mapping similar inputs to similar outputs, (2) continuously adapting to changing statistics of the input stream, (3) forming fixed sparsity representations, (4) being robust to noise, and (5) being fault tolerant. As an integral component of HTM, the outputs of the SP can be easily recognized by downstream neurons and contribute to improved performance in an end-to-end HTM system.

The primary goal of this paper is to provide a thorough discussion of the computational properties of the HTM spatial pooler and demonstrate its value in end-to-end HTM systems. The paper is organized as follows. We first introduce the HTM spatial pooler algorithm. We discuss the computational properties in detail and then describe a set of metrics to quantify them. We demonstrate how these properties are satisfied, first using specific isolated simulations, and then in the context of an end-to-end HTM system. In the discussion, we propose potential neural mechanisms and discuss the relationship to existing sparse coding techniques.

3. Model

The spatial pooler is a core component of HTM networks (Fig. 1A). In an end-to-end HTM system, the spatial pooler transforms input patterns into SDRs in a continuous online fashion. The HTM temporal memory learns temporal sequences of these SDRs and makes predictions for future inputs (Cui et al., 2016a; Hawkins and Ahmad, 2016). A single layer in an HTM network is structured as a set of mini-columns, each with a set of cells (Fig. 1B). The HTM neuron model incorporates dendritic properties of pyramidal cells in neocortex (Spruston, 2008), where proximal and distal

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

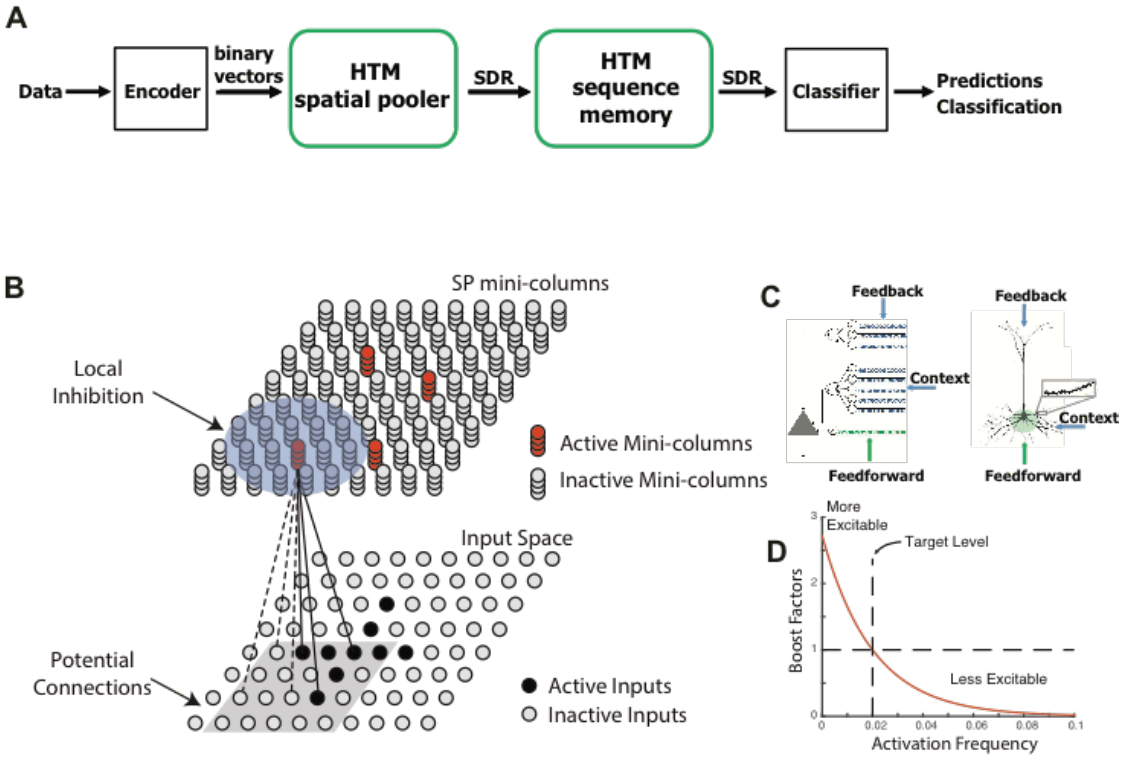

Figure 1 HTM Spatial Pooler. A. An end-to-end HTM system consists of an encoder, the HTM spatial pooler, the HTM temporal memory and an SDR classifier. **B.** The HTM spatial pooler converts inputs (*bottom*) to SDRs (*top*). Each SP mini-column forms synaptic connections to a subset of the input space (gray square, potential connections). A local inhibition mechanism ensures that a small fraction of the SP mini-columns that receive most of the inputs are active within the local inhibition radius (shaded blue circle). Synaptic permanences are adjusted according to the Hebbian rule: for each active SP mini-column, active inputs (black lines) are reinforced and inactive inputs (dashed lines) are punished. **C.** An HTM neuron (left) has three distinct dendritic integration zones, corresponding to different parts of the dendritic tree of pyramidal neurons (right). The SP models the feedforward connections onto the proximal dendrite. **D**. The excitability of a SP mini-column depends on its past activation frequency.

dendritic segments on HTM neurons have different functions (Fig. 1C) (Hawkins and Ahmad, 2016). Patterns detected on proximal dendrites lead to action potentials and define the classic receptive field of the neuron. Patterns recognized by a neuron's distal synapses act as predictions by depolarizing the cell without directly causing an action potential.

In HTM theory different cells within a mini-column represent this feedforward input in different temporal contexts. The spatial pooler models synaptic growth in the proximal dendritic segments. Since cells in a mini-column share the same feedforward classical receptive field (Buxhoeveden, 2002), the SP models how this common receptive field is learned from the input. The SP output represents the activation of mini-columns in response to feedforward inputs. The HTM temporal memory models a cell's distal dendritic segments and learns transitions of SDRs by activating different sets of cells depending on the temporal context (Hawkins and Ahmad, 2016). The output of HTM temporal memory represents the activation of individual cells across all mini-columns.

The SP models local inhibition among neighboring mini-columns. This inhibition implements a *k*-winners-take-all computation (Majani et al., 1988; Makhzani and Frey, 2015). At any time, only a small fraction of the mini-columns with the most active inputs become active. Feedforward connections onto active cells are modified according to Hebbian learning rules at each time step. A homeostatic excitatory control mechanism operates on a slower time scale. The mechanism is called "boosting" in (Hawkins et al., 2011), because it increases the relative excitability of mini-columns that are not active enough. Boosting encourages neurons with insufficient connections to become active and participate in representing the input.

Each SP mini-column forms synaptic connections to a population of input neurons. We assume that the input neurons are arranged topologically in an input space. The location of the *j*th input neuron is denoted as $\mathbf{x}_j$. The dimensionality of the input space depends on applications. For example, the input space is 2-dimensional if the inputs are images and 1-dimensional if the inputs are scalar numbers. A variety of encoders are available to deal with different data types (Purdy, 2016). The output neurons are also arranged topologically in a different space; we denote the location of the *i*th SP mini-column as $\mathbf{y}_i$.

We use HTM neuron models in the SP (Fig. 1B). A complete description of the motivation and supporting evidence for this model can be found in (Hawkins and Ahmad, 2016). In this model, the learning rule is inspired by neuroscience studies of activity-dependent synaptogenesis (Zito and Svoboda, 2002). The synapses for the *i*th SP mini-column are located in a

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

hypercube of the input space centered at $\mathbf{x}_i^c$ with an edge length of γ. Each SP mini-column has potential connections to a fraction of the inputs in this region. We call these "potential" connections because a synapse is connected only if its synaptic permanence is above the connection threshold. The set of potential input connections for the *i*th mini-column is,

$$PotentialInput(i) = \{ j \mid \mathrm{I}(\mathbf{x}_\mathrm{j}; \mathbf{x}_i^c, \gamma) and \; Z_{ij} < p \} \quad (1)$$

$\mathrm{I}(\mathbf{x}_\mathrm{j}; \mathbf{x}_i^c, \gamma)$ is an indicator function that returns one only if $\mathbf{x}_\mathrm{j}$ is located with a hypercube centered at $\mathbf{x}_i^c$ with an edge length of γ. $Z_{ij} \sim U(0,1)$ is a random number uniformly distributed in [0, 1], p is the fraction of the inputs within the hypercube that are potential connections.

We model each synapse with a scalar permanence value and consider a synapse connected if its permanence value is above a connection threshold. We denote the set of connected synapses with a binary matrix $\mathbf{W}$,

$$W_\mathrm{ij} = \begin{cases} 1 \; if \; D_\mathrm{ij} \geq \theta_c \\ 0 \; otherwise \end{cases} \quad (2)$$

where D_ij gives the synaptic permanence from the *j*th input to the *i*th SP mini-column. The synaptic permanences are scalar values between 0 and 1, which are initialized to be independent and identically distributed according to a uniform distribution between 0 and 1 for potential synapses.

$$D_\mathrm{ij} = \begin{cases} \mathrm{U}(0,1) \; if \; j \in PotentialInput(i) \\ 0 \qquad otherwise \end{cases} \quad (3)$$

The connection threshold θ_c is set to be 0.5 for all experiments, such that initially 50% of the potential synapses are connected. Performance of the SP is not sensitive to the connection threshold parameter.

Neighboring SP mini-columns inhibit each other via a local inhibition mechanism. We define the neighborhood of the *ith* SP mini-column $\mathbf{y}_\mathrm{i}$ as

$$Neighbor(i) = \{ j \mid \left\| \mathbf{y}_\mathrm{i} - \mathbf{y}_\mathrm{j} \right\| < \phi, j \neq i \} \quad (4)$$

where $\left\| \mathbf{y}_\mathrm{i} - \mathbf{y}_\mathrm{j} \right\|$ is the Euclidean distance between the mini-column *i* and *j*. Since local inhibition occurs among neighboring mini-columns, the parameter ϕ controls the inhibition radius. Local inhibition is important when the input space has topology, that is, when neighboring input neurons represent information from similar subregions of the input space. The inhibition radius is dynamically adjusted to ensure local inhibition affects mini-columns with inputs from the same region of the input space. That is, ϕ increases if the average receptive field size increases. Specifically, ϕ is determined by the product of the average connected input spans of all SP mini-columns and the number of mini-columns per input. If SP inputs and mini-columns have the same dimensionality, $\phi = \gamma$ initially. In practice we also deal with input spaces that have no natural topology, such as categorical information (Purdy, 2016). In this case there is no natural ordering of inputs and we use an infinitely large ϕ to implement global inhibition.

Given an input pattern $\mathbf{z}$, the activation of SP mini-columns is determined by first calculating the feedforward input to each mini-column, which we call the input overlap

$$o_i = b_i \sum_j W_\mathrm{ij} z_j \quad (5)$$

b_i is a positive boost factor that controls the excitability of each SP mini-column.

A SP mini-column becomes active if the feedforward input is above a stimulus threshold θ_{stim} and is among the top *s* percent of its neighborhood,

$$a_i = 1 \; if \; \mathrm{o}_i \geq \mathrm{prctile}(NeighborOverlap(i), 1 - s) \; and \; \mathrm{o}_i \geq \theta_{stim} \quad (6$$

We typically set θ_{stim} to be a small positive number to prevent mini-columns without sufficient input to become active. $\mathrm{prctile}(\cdot, \cdot)$ is the percentile function. $NeighborOverlap(i)$ is the overlap values for all neighboring mini-columns of the *i*th mini-column.

$$NeighborOverlap(i) = \{ o_j | j \in Neighbor(i) \} \quad (7)$$

s is the target activation density (we typically use *s*=0.02). The activation rule (Eq. 6-7) implements *k*-winners-take-all computation within a local neighborhood. It has been previously shown that such computation can be realized by integrate-and-fire neuron models with precise spike timings (Billaudelle and Ahmad, 2015a). In this study we use discrete time steps to speed up simulation.

The feedforward connections are learned using a Hebbian rule. For each active SP mini-column, we reinforce active input connections by increasing the synaptic permanence by p^+, and punish inactive connections by decreasing the synaptic permanence by p^-. The synaptic permanences are clipped at the boundaries of 0 and 1.

To update the boost factors, we compare the recent activity of each mini-column to the recent activity of its neighbors. We calculate the time-averaged activation level for the each mini-column over the last *T* inputs as

$$\bar{a}_i(t) = \frac{(T-1) * \bar{a}_i(t-1) + a_i(t)}{T} \quad (8)$$

where $a_i(t)$ is the current activity of the *i*th mini-column at time *t*. *T* controls how fast the boost factors are updated. Because the activity is sparse it requires many steps before we can get a meaningful estimate of the activation level. Typically we choose *T* to be 1000. The time-averaged activation level in Eq. 8 can be approximated by low-pass filtering of the voltage signal or intracellular calcium concentration. Similar calculations have been used in previous models of homeostatic synaptic plasticity (Clopath et al., 2010; Habenschuss et al., 2013).

The recent activity in the mini-column's neighborhood is calculated as

$$< \bar{a}_i(t) >= \frac{1}{|Neighbor(i)|} \sum_{j \in Neighbor(i)} \bar{a}_j(t) \quad (9)$$

The boost factor b_i is then updated based on the difference between $a_i(t)$ and $< \bar{a}_i(t) >$ as shown in Fig. 1D.

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

$$b_i = e^{-\beta(\bar{a}_i(t) - <\bar{a}_i(t)>)} \quad (10)$$

Here β is a positive parameter that controls the strength of the adaptation effect. The above boosting mechanism is inspired by numerous studies of homeostatic regulation of neuronal excitability (see (Davis, 2006) for a review). The mechanism encourages efficient use of mini-columns. The exact formula is not critical; we chose Eq. (10) due to its simplicity.

4. Properties of the HTM spatial pooler

In this section we describe a set of desirable properties for the HTM spatial pooler. These properties ensure flexible and robust representations of input streams with changing statistics, and are important for downstream neural computation.

The first property of the SP is to form fixed-sparsity representations of the input. To contribute to further neural computation, the outputs of the SP have to be recognized by downstream neurons. A cortical neuron recognizes presynaptic input patterns by initiating nonlinear dendritic spikes (Major et al., 2013) or somatic action potentials (Bean, 2007), with thresholds depending on intrinsic cellular properties. It has been previously shown that recognition of presynaptic activation patterns is robust and reliable if the presynaptic inputs have a fixed level of sparsity (Ahmad and Hawkins, 2016). However, if the sparsity is highly variable, input patterns with high activation densities would be more likely to cause dendritic spikes or action potentials in downstream neurons, whereas patterns with low activation densities would be much harder to detect. This will contribute to high false positive error for high density patterns and false negative error for low density patterns. A fixed sparsity is desirable because it ensures all input patterns can be equally detected.

A second desirable property is that the system should utilize all available resources to learn optimal representations of the inputs. From an information theoretic perspective, neurons that are almost always active and neurons that do not respond to any of the input patterns convey little information about the inputs. Given a limited number of neurons, it is preferable to ensure every neuron responds to a fraction of the inputs such that all neurons participate in representing the input space. The boosting mechanism in the SP (Eq. 8-10) is designed to achieve this goal. We quantify this property using an entropy metric (see details below).

A third desirable property is that output representations should be robust to noise in the inputs. Real-world problems often deal with noisy data sources where sensor noise, data transmission errors, and inherent device limitations frequently result in inaccurate or missing data. In the brain, the responses of sensory neurons to a given stimulus can vary significantly (Tolhurst et al., 1983; Faisal et al., 2008; Masquelier, 2013; Cui et al., 2016c). It is important for the SP to have good noise robustness, such that the output representation is relatively insensitive to small changes in the input.

A fourth property is that the system should be flexible and able to adapt to changing input statistics. The cortex is highly flexible and plastic. Regions of the cortex can learn to represent different inputs in reaction to changes in the input data. If the statistics of the input data changes, the spatial pooler should quickly adapt to the new data by adjusting its synaptic connections. This property is particularly important for applications with continuous data streams that has fast-changing statistics (Cui et al., 2016b).

Finally, a fifth property is that the system should be fault tolerant. If part of the cortex is damaged, as might occur in stroke or traumatic brain injury, there is often an initial deficit in perceptual abilities and motor functions which is followed by substantial recovery that occurs in the weeks to months following injury (Nudo, 2013). It has also been documented that the receptive fields of sensory neurons reorganize following restricted lesions of afferent inputs, such as retinal lesions (Gilbert and Wiesel, 1992; Baker et al., 2005). The spatial pooler should continue to function in the event of system faults such as loss of input or output neurons in the network.

5. Spatial pooler metrics

In addition to gauging performance in end-to-end HTM systems, we would like to quantify the performance of spatial pooler as a standalone component. Since the spatial pooler is an unsupervised algorithm designed to achieve multiple properties, we describe several statistical metrics that can be directly calculated based on the inputs and outputs of the spatial pooler. Such metrics are particularly useful if configurations of the spatial pooler caused the end-to-end HTM system to have poor performance.

Metric 1: sparseness. We define the population sparseness as

$$s^t = \frac{1}{N}\sum_{i=1}^{N} a_i^t \quad (11)$$

a_i^t is the activity of the *i*th mini-column at time step t, N is the number of SP mini-columns. This metric reflects the percentage of active neurons at each time step. Since we consider binary activations (Eq. 6), the sparsity is straightforward to calculate. This metric has the same spirit as other population sparseness metrics for scalar value activations (Willmore and Tolhurst, 2001). We can quantify how well the spatial pooler achieves a fixed sparsity by looking at the standard deviation of the sparseness across time.

Metric 2: entropy. Given a dataset of M inputs, the average activation frequency of each SP mini-column is

$$P(a_i) = \frac{1}{M}\sum_{t=1}^{M} a_i^t \quad (12)$$

The entropy of the *i*th SP mini-column is given by the binary entropy function

$$S_i = -P(a_i)\log_2 P(a_i) - (1 - P(a_i))\log_2(1 - P(a_i)) \quad (13)$$

If $P(a_i)$ equals zero or one, we set S_i to zero following convention. We define the entropy of the spatial pooler as

$$Entropy = \sum_{i=1}^{N} S_i \quad (14)$$

Since the average sparseness is almost constant in SP, the entropy is maximized when every SP mini-column has equal

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

activation frequency. The spatial pooler will have low entropy if a small number of the SP mini-columns are active very frequently and the rest are inactive. Therefore, the entropy metric quantifies whether the SP efficiently utilizes all mini-columns.

Metric 3: noise robustness. We test noise robustness by measuring the sensitivity of the SP representation to varying amount of input noise. We denote a clean input and a noise contaminated input as $\mathbf{z}_i$ and $\mathbf{z}_i'(k)$ respectively, where k denotes the amount of noise added to the input. The corresponding SP outputs are denoted as $\mathbf{a}_i$ and $\mathbf{a}_i'(k)$ respectively. In our simulations, we randomly flip k percent of the active input bits to inactive, and flip the corresponding number of inactive input bits to active. This procedure randomizes inputs while maintaining constant input sparsity. We vary the amount of noise between 0 and 100%, and measure the fraction of shared active mini-columns in the SP output, averaged over a set of M inputs. The noise robustness index is defined as

$$NoiseRobustness = \frac{1}{M}\sum_{i=1}^{M}\int_{k=0}^{1}\frac{\|\mathbf{a}_i \circ \mathbf{a}_i'(k)\|_0}{\|\mathbf{a}_i\|_0}\mathrm{d}k \quad (15)$$

The L0-norm $\|\cdot\|_0$ gives the number of non-zero bits in a binary vector; the $\circ$ operator represents element-wise multiplication. Note that with binary vectors the L0-norm is identical to the L1-norm. The noise robustness index measures the area under the output overlap curve in Fig. 2C. The fraction of shared active mini-columns start at 100% when noise is zero, and decreases towards 0 as the amount of noise increases. The noise robustness thus lies between 0 and 1.

Metric 4: stability. Since the spatial pooler is a continuously learning system, it is possible that the representation for a given input changes over time or becomes unstable. Instability without changes in the input statistics could negatively impact downstream processes. We measure stability by periodically disabling learning and presenting a fixed random subset of the input data. The stability index is the average fraction of active mini-columns that remained constant for each input.

Denote the SP output to the ith test input at the jth test point as $\mathbf{a}_i^j$, the stability index at test point j is given as,

$$Stability(j) = \frac{1}{M}\sum_{i=1}^{M}\frac{\left\|\mathbf{a}_i^j \circ \mathbf{a}_i^{j-1}\right\|_0}{\left\|\mathbf{a}_i^{j-1}\right\|_0} \quad (16)$$

M is the number of inputs tested. The stability lies between 0 and 1, and equals 1 for a perfectly stable spatial pooler. Note that the superscript j is the index for test points instead of time steps in Eq. 16. We compute the percentage overlap between the SP outputs to the same test inputs across consecutive test points. We train the SP on the entire set of training data between test points.

6. Simulation details

We ran a number of different simulations (datasets described below). We used either a 2-dimensional spatial pooler with 32x32 mini-columns for experiments with topology, or a dimensionless spatial pooler with 1024 mini-columns for experiments without topology. The complete set of spatial pooler parameters is given in Table 1. The source code for all experiments are openly available at:

https://github.com/numenta/nupic.research/tree/master/projects/sp_paper

Table 1 Parameters for the HTM spatial pooler

Common parameters	Value
Activation density	2%
Connection threshold for synaptic permanence θ_c	0.5
Synaptic permanence increment p^+	0.1
Synaptic permanence decrement p^-	0.02
Boosting strength β	100
Activation frequency duty cycle T	1000
Stimulus threshold θ_{stim}	1
Fraction of potential inputs p	1
Parameters for the 1-dimensional spatial pooler (no topology)	
Column dimensions	1024x1
Potential input radius γ	∞
Parameters for the 2-dimensional spatial pooler (with topology)	
Column dimensions	32x32
Potential input radius γ	5

We presented each dataset in a streaming online fashion. Each dataset is presented to the SP in one or more epochs. We define an epoch as a single pass through the entire dataset in random order. Note that this definition of epoch is different from batch training paradigm because the SP receives one input pattern at a time and does not maintain any buffer of the entire dataset. We measured SP metrics between epochs on a random subset of the input data with learning turned off. In practice, the metrics could also be monitored continuously during learning.

We used the following datasets:

Random Sparse Inputs. In this experiment we created a set of 100 random inputs with varying sparsity levels. Each input is a 32x32 image where a small fraction of the bits are active. The fraction of active inputs is uniformly chosen between 2% to 20%. This dataset employs a spatial pooler with topology and is used in Figs. 2-3.

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Random Bars and Crosses. The random bar dataset consists of 100 pairs of random bars. Each random bar pair stimuli is a 10x10 image, with a horizontal bar and a vertical bar placed at random locations. The bars have a length of 5 pixels. Each random cross stimuli is a 10x10 image with a single cross. The random cross dataset consists of 100 cross patterns at random locations, where each cross consists of a horizontal bar and a vertical bar that intersect at the center. Each bar has a length of 5 pixels. This dataset is used in Fig. 4A-B.

MNIST. We trained a spatial pooler without topology on the MNIST database of handwritten digits (Lecun et al., 1998). We present the full training set of 60,000 examples in a single epoch to the spatial pooler. We visually examined the receptive field structures of a subset of randomly selected SP mini-columns after training. This dataset is used in Fig. 4C.

Fault tolerance with topology. For the fault tolerance experiment (Fig. 5), we used images of random bar sets as input. The input space has dimensionality of 32x32 and each input contains 6 randomly located horizontal or vertical bars. Each bar has a length of 7. We used an SP with a two-dimensional topology and 32x32 mini-columns. We first trained the intact SP on the random bars input until it stabilized (after 18000 inputs). We then tested two different types of trauma: simulated stroke or simulated input lesion. During the simulated stroke experiment, we permanently eliminated 121 SP mini-columns that lie in an 11x11 region at the center of the receptive field. For the simulated input lesion experiment, we did not change the SP during the trauma. Instead, we permanently blocked the center portion of the input space (121 inputs that lie in an 11x11 region at the center of the input space). For both experiments, we monitored the recovery of the SP for another 42000 steps.

NYC taxi passenger count prediction. In addition to the above artificial datasets, we also tested the SP in an end-to-end real-world HTM system. We chose the problem of demand prediction for New York City taxis. The dataset is publicly available via the New York City Metropolitan Authority. Full details are described in our previous paper (Cui et al., 2016a). As described in Fig. 1, the input data stream is first converted to binary representations using a set of encoders (Purdy, 2016). The spatial pooler takes the output of the encoders as input and forms sparse distributed representations. The HTM sequence memory then learns sequences of SDRs and represents sequences with a sparse temporal code. Finally, we use a single layer feedforward classification network to map outputs of HTM sequence memory into real-time predictions for future inputs. The task is to model a continuous stream within the context of a real-time application. As such the spatial pooler, temporal memory, and classifier all learn continuously. It is important for the spatial pooler to output robust and efficient representations in order for the downstream components to learn.

Following (Cui et al., 2016a), we aggregated the passenger counts in New York City taxi rides at 30-minute intervals. We encoded the current passenger count, time of day and day of week into binary vectors using scalar and date-time encoders (Purdy, 2016). A spatial pooler with global inhibition was trained continuously on the outputs of encoders and provided input to the HTM sequence memory (Cui et al., 2016a). To evaluate the role of learning and boosting in SP, we compared the prediction accuracy in three scenarios (1) SP without learning or boosting, (2) SP with learning but not boosting, and (3) SP with both learning and boosting.

7. Results

We first discuss results on the Random Sparse Inputs dataset with respect to the metrics (Figs. 2 and 3). The input patterns are presented repeatedly to the HTM spatial pooler in a streaming fashion. In our simulation, the population sparsity of the spatial pooler is always close to the target level of 2%, even though the input sparsity varies widely in the range of 2%~20% (Fig. 2A). This is an inherent property of the network due to the use of local *k*-winners-take-all activation rules.

We measured the average entropy across all mini-columns (see Spatial Pooler Metrics). Since the overall activation sparsity is fixed in our network, the entropy is maximized if all mini-columns have the same activation probability. In this experiment, the entropy increases from 0.1221±0.0013 bits/mini-column to 0.1320±0.0007 bits/mini-column with training. The difference is highly significant across repeated experiments with different set of random inputs ($p<10^{-8}$, n=10, paired t-test). As a reference, the maximum possible entropy is 0.1345 bits/mini-column with the same sparsity levels. The increase of entropy is due to efficient use of all mini-columns. Before learning, a significant fraction of the mini-columns (~30%) were not active for any of the input, whereas a small fraction of the mini-columns were much more active than others. After learning, almost every mini-column was active for 2% of the time.

Figs. 2D and 2E demonstrate noise robustness as a function of SP learning. Before learning, a small change in the input will cause a large change in the SP output, suggesting high noise sensitivity (Fig. 2D, blue). After learning, the noise robustness gradually improves. After 40 repetitions of the dataset, there is no change on the SP output even if 40% of the active input bits are changed. The average noise robustness index (Eq. 15) correspondingly improves from 0.254±0.004 to 0.652±0.007 (Fig. 2E, $p<10^{-16}$, n=10, paired t-test). The improved noise robustness is due to the Hebbian learning rules. A set of SP mini-columns forms reliable connections to active input neurons during learning. The same set of SP mini-columns can be activated even if some of the input neurons are affected by noise after learning.

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

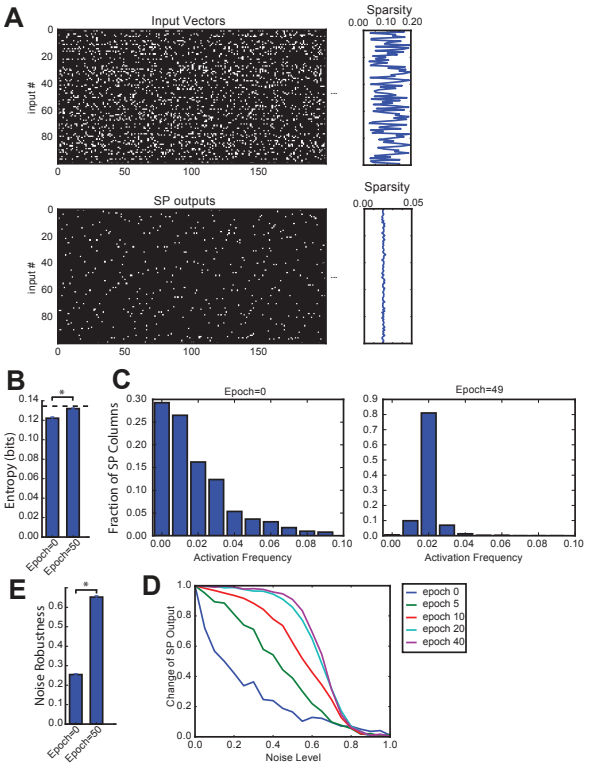

Figure 2 Spatial pooler forms SDRs with fixed sparsity and good noise robustness. A. We trained SP on a set of 100 randomly generated inputs (*top*). The input sparsity varies between 2% to 20%. The sparsity of the SP output lies close to 2% (*bottom*), despite the large variation of input sparsity. **B.** Entropy before and after learning, averaged across 10 repeated experiments (* $p<10^{-16}$, n=10, paired t-test). The maximum possible entropy is shown as the black dashed line. **C.** The distribution of activation frequency of SP mini-columns. Before learning (*left*), a significant fraction of the SP mini-columns (~30%) are not being used at all, while other mini-columns are active much more frequently. After learning (*right*), almost every mini-column is active for 2% of the time, suggesting every SP mini-column participate in representing the input. As a result, the entropy of the distribution is much higher. **D-E** Noise robustness of SP. We tested SP on noisy inputs during learning. **D**. The change of the SP outputs is plotted as a function of the noise level. Before learning, a small amount of noise will lead to significant change in the SP output (*blue*), whereas after learning, there is almost no change in the SP output when 50% of the input bits changed. **E.** Average noise robustness before and after learning (* $p<10^{-16}$, n=10, paired t-test).

To test whether the spatial pooler can adapt to changing inputs, we train the spatial pooler until it stabilizes on one set of inputs. We then present a completely different input dataset (Fig. 3A). Right after we switch to a new dataset, the entropy and noise robustness drops sharply (Fig. 3B, 3D). At this point a large fraction of the SP mini-columns are not responsive to any input in the new dataset (Fig. 3C, left). Once learning resumes the spatial pooler quickly adapts to the new input dataset and the performance metrics recover back to the levels before the change (Fig. 3B, 3D). The SP adapts to the new dataset by first forming many more new synapses (Fig. 3A, 4th row) and then pruning unnecessary connections later (Fig. 3A, 5th row).

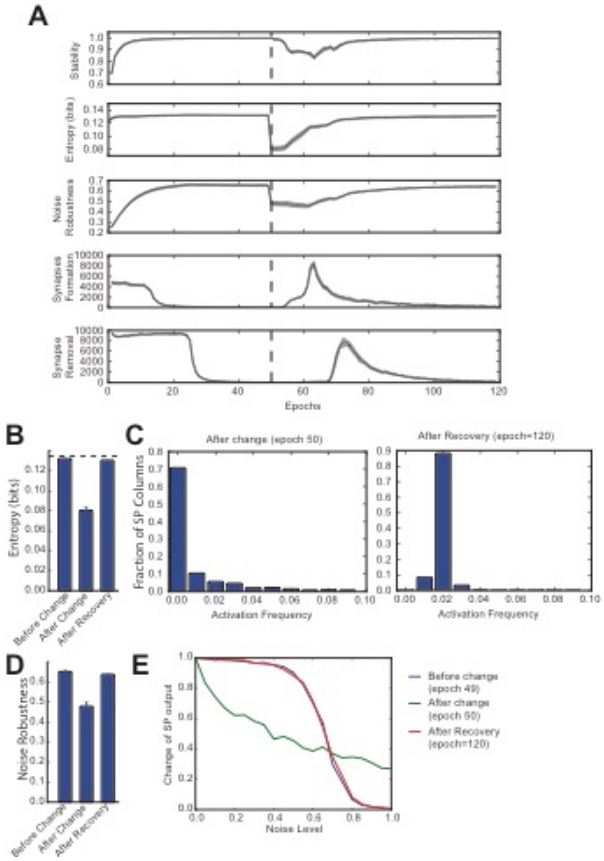

Figure 3 Continuous learning with HTM spatial pooler. SP continuously adapts to the statistics of the input data. SP is trained on a set of random inputs (described in Fig. 2) until it stabilizes. We then switch to a new set of inputs (black dashed line) and monitor the continuous adaptation of SP to the new dataset. **A.** Statistical metrics on SP during continuous learning: top: stability, 2nd row: entropy; 3rd row: noise robustness; 4th row: formation of new synapses; 5th row: removal of synapses. **B.** The entropy decreases right after the change of the input dataset (epoch=50), and recovers completely after the SP is trained on the new dataset for long enough time (epoch=120). The black dashed line showed the theoretical limit for entropy given the sparsity constraint. **C.** Distribution of activation frequency of SP mini-columns right after the change in dataset (left) and after recovery (right). **D**. Noise robustness before change (epoch=49), right after change (epoch=50) and after recovery (epoch=120). **E.** The noise robustness decreases right after the change of the input dataset (blue vs. green), and recovers completely after the SP is trained on the new dataset for long enough time (red).

The spatial pooler achieves these properties by continuously adapting feedforward connections to the input data. To illustrate how receptive field structures of SP are shaped by the input data, we trained the spatial pooler on the Random Bars dataset. In this experiment, the training data consists of

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

either pairs of randomly generated bars, or a single cross at a random location.

We plot example receptive fields from a random subset of SP mini-columns in Fig. 4. For this dataset the receptive field typically contains horizontal and vertical bar structures at random locations (Fig. 4A). Each SP mini-column responds to more than one bar in order to achieve the target activation frequency of 2%. When trained on the random cross data stream, the resulting receptive field consists of cross structures that resemble statistics of the inputs (Fig. 4B). Note that the resulting receptive field also depends on the ratio of synaptic permanence decrement and increment (p^-/p^+). If we increase p^- to 0.05, most of the receptive field contains single bar segment when trained on the random bar pairs dataset (data not shown).

We also trained the spatial pooler on the MNIST dataset. In this case the receptive field contains digit-like structures (Fig. 4C). Although some receptive fields clearly detect single digits, others are responsive to multiple digits. This is because individual SP mini-columns do not behave like "grandmother cells" -- they are not meant to detect single instances of the inputs. Instead a single input is collectively encoded by a set of SP mini-columns. When trained on complex natural datasets, we expect to see a diversity of receptive structures within the spatial pooler, which may not resemble specific input instances.

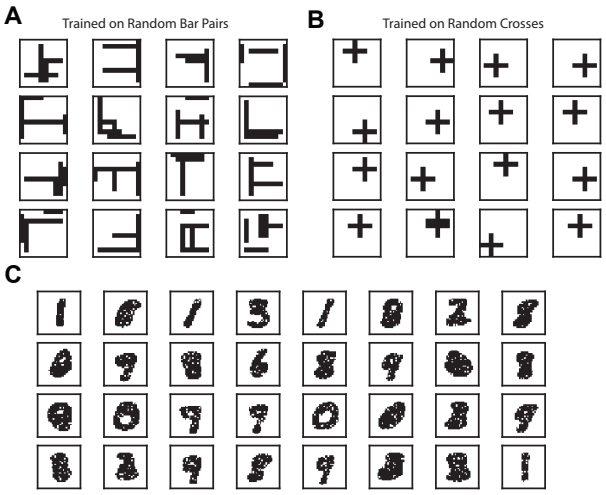

Figure 4 Example receptive fields of SP. The receptive fields of SP mini-columns capture statistics of the input data. We define receptive field as the set of inputs that are connected to a mini-column. **A.** Example SP Receptive fields trained on random bar pairs. **B.** Example SP receptive fields trained on random crosses. **C.** Example SP receptive fields trained on MNIST dataset.

7.1. Fault tolerance

We evaluated whether the HTM spatial pooler has the ability to recover from lesion of the afferent inputs (input lesion experiment) or damage to a subset of the SP mini-columns (stroke experiment).

In the stroke experiment, the center portion of the input space becomes much less represented right after the trauma because the corresponding SP mini-columns are eliminated. The network partially recovers from the trauma after a few hundred epochs, with each epoch consisting of 100 inputs. During the recovery process, SP mini-columns near the trauma region shift their receptive field toward the trauma region and start to represent stimuli near the center (Fig. 5C, supplemental movie 1). The network forms many more new synapses in the center, which is accompanied by loss of synapses in the non-trauma region (Fig. 5B, top). There is a clear recovery on the coverage of the trauma region (Fig. 5D, bottom).

In the input lesion experiment, the center portion of the input space is blocked. Since there is no change on the SP mini-columns or the associated synaptic connections, there is no immediate change on the input space coverage (Fig. 5A, bottom) or the receptive field center distributions (Fig. 5C, bottom). However, the SP mini-columns quickly reorganize their receptive field within a few epochs. The SP mini-columns that respond to the center inputs starts to respond to inputs on the surrounding, non-damaged areas (Fig. 5C, supplemental movie 2). Almost all the connections to the lesion region are lost after the reorganization (Fig. 5B, bottom).

These demonstrate the fault tolerance and flexibility of the SP. The fixed sparsity and the homeostasis excitability control mechanism of the SP ensure that the input space is efficiently represented by all SP (undamaged) mini-columns. It is interesting to note that the different recovery speeds from the two simulations coincide with experimental studies. It has been reported that after focal binocular retinal lesions, the receptive field sizes increases within a few minutes for cortical neurons that lie near the edge of the retinal scotoma (Gilbert and Wiesel, 1992). In contrast, if part of the cortex is damaged, the recovery is partial and occurs on a much slower time scale (Nudo, 2013).

7.2. The spatial pooler in a real-world streaming analytics task

In this section we evaluate the role of the SP in an end-to-end real-world HTM system. We consider the problem of real-time prediction of the number of taxi passenger in New York City (Fig. 6A, see Methods). We have previously shown that HTM systems with a fixed pre-trained spatial pooler achieves state-of-the-art performance on this task (Cui et al., 2016b). Here we consider the role of learning in the spatial pooler and evaluate three scenarios: using a randomly initialized SP without learning, allowing SP learning but without boosting, and using a SP with both continuous learning and boosting.

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

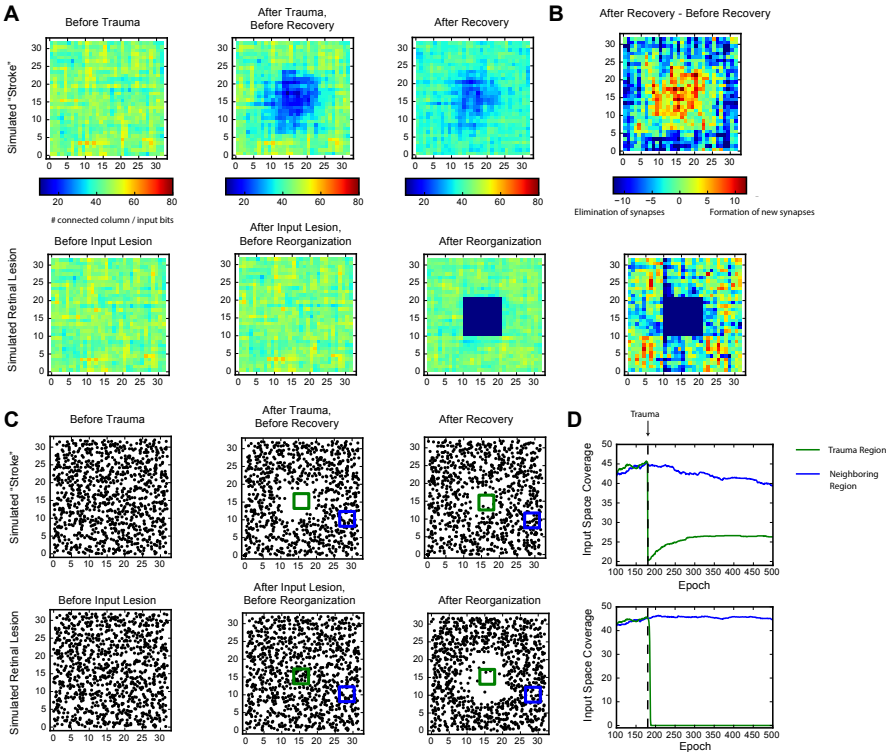

Figure 5 Recovery of HTM spatial pooler after damage and input lesion. During the simulated stroke, a fraction of SP mini-columns that are connected to the center region of the input space is killed. During the simulated retinal lesion, the center portion of the input space is blocked while the spatial pooler and its feedforward inputs are kept intact. **A.** The number of SP mini-columns connected to each input bits before trauma (*left*), right after trauma (*middle*) and after recovery (*right*). The simulated stroke experiment is shown at the top and the simulated retinal lesion experiment is shown at the bottom. **B.** Growth and elimination of synapses during the recovery process for the simulated stroke (*top*) and retinal lesion (*bottom*) experiment. **C.** Receptive field centers of all SP mini-columns before trauma (*left*), right after trauma (*middle*) and after recovery (*right*). **D.** Number of mini-columns connected to the center region (green square in Fig. 5C) and a neighboring region (blue square) during the recovery process. The recovery is very fast for the retinal lesion experiment (bottom), and slower for the simulated stroke experiment (top).

We fed the inputs in a single pass to mimic the scenario of real-time online prediction. The time-averaged prediction error for the three cases is plotted as a function of training time (Fig. 6B). At the beginning of learning, the prediction error rapidly decreases, representing the initial learning phase of the system. The occasional increases in error reflect real world changes that correspond to events and holidays (e.g., Thanksgiving, Christmas, New York City Marathon, etc.).

The spatial pooler with both learning and boosting achieves the best performance throughout the prediction task. The spatial pooler with learning but without boosting is roughly comparable to a random static SP. This suggests the importance of both continuous Hebbian learning and homeostatic excitability control. The difference in performance can be understood by observing the distribution of activation frequency across SP mini-columns. Without the homeostatic excitability control, a large fraction of the SP mini-columns are not being used at all (Fig. 6C). It is more error-prone for the HTM sequence memory to learn transitions of such ill-behaved SDRs.

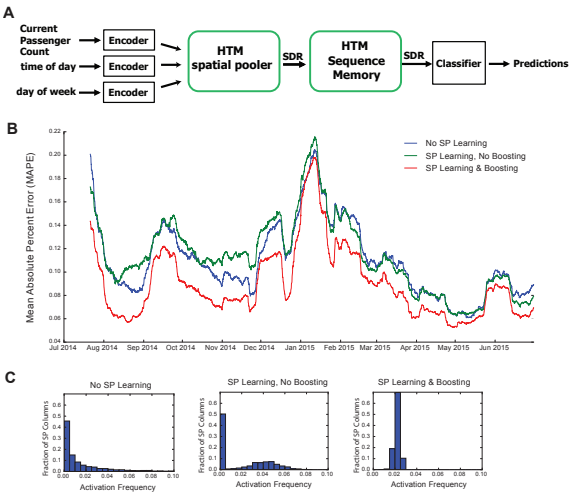

Figure 6 The role of HTM spatial pooler in prediction task. A. A complete HTM system of encoder -> spatial pooler ->

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

sequence memory -> classifier is used for predicting the NYC taxi passenger count. B. Prediction error with an untrained random SP (blue), a SP with continuous learning but without boosting (green), and a SP with both continuous learning and boosting (red). C. Distribution of activation frequency of SP mini-columns. A large fraction of SP mini-columns are not being used for an untrained SP or for a SP without boosting. In contrast, almost all mini-columns are active for about 2% of the time when boosting is enabled.

8. Discussion and Conclusions

In this paper, we described properties of the HTM spatial pooler, a neurally inspired algorithm for learning sparse distributed representations online. Inspired by computational principles of the neocortex, the goal of the HTM spatial pooler is to create SDRs and support essential neural computations such as sequence learning and memory. The model satisfies a set of important properties, including tight control of output sparsity, efficient use of mini-columns, preserving similarity among inputs, noise robustness, fault tolerance and fast adaptation to changes. These properties are achieved using competitive Hebbian learning rules and homeostatic excitability control mechanisms. We demonstrate the effectiveness of SP in an end-to-end HTM system on the task of streaming data prediction. The HTM spatial pooler leads to a flexible sparse coding scheme that can be used in practical machine learning applications.

8.1. Relationship with other sparse coding techniques

The spatial pooler learns sparse distributed representations for inputs. It is related to the broad class of sparse coding techniques, which uses activation of a small set of neurons to encode each item. One theory of sparse coding suggests that sparse activations in sensory cortices reduce energy consumption of the brain while preserving most of the information (Földiák, 2002; Olshausen and Field, 2004). Early studies of sparse coding explicitly optimize a cost function that combines low reconstruction error and high sparseness (Olshausen and Field, 1996a, 1997). When applied to natural images, these techniques lead to receptive fields that resemble those of V1 neurons (Olshausen and Field, 1996a; Lee et al., 2006), suggesting that the functionality of early sensory neurons can be explained by the sparse coding framework. Sparse coding has been implemented previously with biologically plausible local learning rules. Földiák showed that a neural network could learn a sparse code using Hebbian forward connections combined with a local threshold control mechanism (Földiák, 1990). It has been recently shown that such learning rules can be derived analytically (Hu et al., 2014).

Many of the properties we analyzed in this paper have also been discussed in previous studies of sparse coding. It has been shown that sparse representations are naturally robust to noise and can be used for robust speech recognition (Sivaram et al., 2010; Gemmeke et al., 2011), robust face recognition (Wright et al., 2009) and super resolution image reconstruction (Jianchao Yang et al., 2010). Online sparse coding and dictionary learning techniques have been proposed in previous studies in order to handle dynamic datasets (Mairal et al., 2010). It is known that representations learned from traditional sparse coding techniques have low entropy, as the probability distribution of activity of an output unit is peaked around zero with heavy tails (Olshausen and Field, 1996b). In this study we found that although the entropy is low compared to dense representations, it increases with training in the HTM spatial pooler. This is because the homeostatic excitability control mechanism encourages neurons in the SP to have similar activation frequencies, thus increasing the representational power of the network.

Most previous studies propose the goal of sparse coding is to avoid information loss, reduce energy consumption, and form associative memory with minimum cross talk (Olshausen and Field, 2004). A commonly used criterion is how well one can reconstruct the inputs given the sparse activations and a set of learned basis vectors. Although these studies explain receptive structure in primary visual cortex and lead to practical machine learning algorithms for feature selection (Gui et al., 2016) and data compression (Pati et al., 2015), the purpose of neural computation is more than preserving information. In this paper, we take a different perspective and ask how computational properties of the HTM spatial pooler contribute to downstream cortical processing in the context of HTM systems. Instead of reconstruction error, we define the performance of SP in terms of a set of properties, including population entropy, noise robustness, stability, and fault tolerance. It is important to perform a multi-dimensional assessment of the SP in order to ensure that it forms robust sparse distributed representations that capture semantic similarity of the inputs. We demonstrated that the HTM spatial pooler achieves these properties and that these properties contribute to improved performance in an end-to-end system. It is important to note that no single metric is sufficient to ensure SP is behaving properly. For example, one can achieve good noise robustness by always using a small set of SP mini-columns, but that will give bad entropy. It is easy to achieve high entropy by using a random output at each time step, but that will cause bad stability. It is important to consider all the metrics together. As a result, the learning algorithm of SP cannot be easily derived by optimizing a single objective function.

The Hebbian learning rules of HTM spatial pooler resemble many previous sparse coding algorithms (Földiák, 1990; Zylberberg et al., 2011; Hu et al., 2014) and associative memory models (Willshaw et al., 1969; Hecht-Nielsen, 1990; Bibbig et al., 1995). There are several differences. First, we include homeostatic excitability control as a gain modulation mechanism. The role of homeostasis is to make sure that the distribution of neural activity is homogeneous. It has been previously proposed that homeostasis is crucial in providing an efficient solution when learning sparse representations (Perrinet, 2010). Some models of synaptic plasticity do include homeostatic components in the learning rule that control the amount of synaptic weight change (Clopath et al., 2010; Habenschuss et al., 2013). The homeostatic excitability regulation mechanism in the SP achieves a similar effect without directly affecting the synapse modification process. Second, we use a local inhibition circuit that implements *k*-winners-take-all computation to have tight control over the output sparseness. This is important when SP activations are used by downstream neurons with dendrites that have

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

threshold nonlinearities. Finally, we use binary synapses and learning via synaptogenesis (Zito and Svoboda, 2002). The use of binary synapses can dramatically speed up the computation. Overall the HTM spatial pooler algorithm is a suitable candidate for learning sparse representations online from streaming data.

8.2. Potential Neural Mechanisms of spatial pooler

A layer in a HTM system contains a set of mini-columns. Each mini-column contains cells with the same feedforward receptive field. The mini-column hypothesis has been proposed for several decades (Mountcastle, 1997; Buxhoeveden, 2002), but the utility of mini-columns remains controversial due to a lack of theoretical benefit (Horton and Adams, 2005). According to HTM theory cells within the same mini-column have the same feedforward connections but different lateral connections, thus representing the same feedforward input in different temporal contexts (Hawkins and Ahmad, 2016). We propose the spatial pooler models feedforward receptive field learning at the mini-column level. Experimental studies have shown that neurons within the same mini-column have almost identical receptive field locations, sizes and shapes, whereas RFs of neurons in neighboring mini-columns can differ significantly (Jones, 2000). This variability cannot be explained by the difference in feedforward inputs, because the extent of arborization of single thalamic afferent fibers, which can be as much as 900 µm in cats (Jones, 2000), is significantly more extensive than the dimensions of minicolumns, which typically have diameters around 20-60 µm (Buxhoeveden, 2002). (Favorov and Kelly, 1996; Jones, 2000). It requires dedicated circuitry mechanism to ensure that cells in the same mini-column acquire the same receptive field.

We propose two possible neural circuit mechanisms for the spatial pooler and discuss their anatomical support. In the first proposal (Fig. 7B, left), the feedforward thalamic inputs innervate both excitatory pyramidal neurons as well as an inhibitory neuron (*green*). This inhibitory neuron can indirectly activate the pyramidal neurons through a disynaptic dis-inhibition circuit. It acts as a "teacher" cell that guides the receptive field formation of excitatory neurons. There are many distinct classes of inhibitory neurons in the cortex. Some classes, such as bipolar cells and double bouquet cells exclusively innervate cells within a cortical mini-column (Markram et al., 2004; Wonders and Anderson, 2006). It is well documented that feedforward thalamacortical input strongly activates specific subtypes of inhibitory neurons (Gibson et al., 1999; Porter et al., 2001; Swadlow, 2002; Kremkow et al., 2016). It is possible these inhibitory neurons participate in defining and maintaining the feedforward receptive field of cortical mini-columns.

In the second proposal, a single excitatory cell receives thalamic inputs and innervates all excitatory cells in a mini-column. This excitatory neuron guides the receptive field formation of other excitatory cells in the mini-column. A similar circuit has been observed during early development. Subplate neurons, a transient population of neurons, receive synaptic inputs from thalamic axons, establishing a temporary link between thalamic axons and their final targets in layer IV (Friauf et al., 1990; Ghosh and Shatz, 1992; Kanold et al., 2003). It remains to be tested whether a similar circuit exists in adult brain.

The spatial pooler relies on several other neural mechanisms. The learning rule is based on competitive Hebbian learning. Such learning can be achieved in the brain via synaptic plasticity rules such as long-term potentiation (Teyler and DiScenna, 1987), long-term depression (Ito, 1989), or spike-time dependent plasticity (Song et al., 2000). Homeostatic excitability control mechanisms, analogous to the spatial pooler's boosting rule, have been observed in cortical neurons (Davis, 2006). Finally, the *k*-winners-take-all computation in the SP can be implemented using leaky integrate-and-fire neuron models (Billaudelle and Ahmad, 2015b).

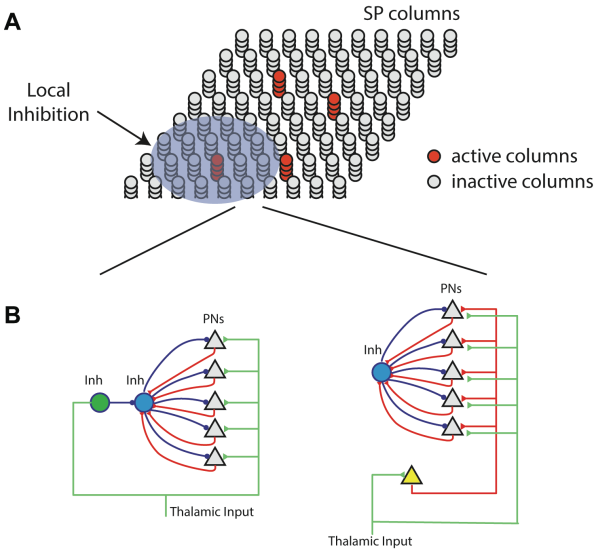

Figure 7 Neural mechanism of HTM Spatial pooler. **A.** Spatial pooler requires local across mini-column inhibition to ensure that a small fraction of the mini-columns are active at any time. **B.** Potential mechanisms to ensure neurons within the same mini-column share the same feedforward receptive field. *left*: the green inhibitory neuron controls the receptive fields of excitatory pyramidal neurons (gray triangles) through a dis-inhibition circuit. *right*: A single (or small number of) excitatory neurons (yellow) controls the receptive field of excitatory neurons. In both cases, PNs indirectly inhibit other PNs in the same mini-column through the within mini-column inhibition (blue).

9. Acknowledgement

We thank the editor and the reviewers for their constructive comments and suggestions. We thank Mirko Klukas for suggestions on the entropy metric. We also thank the NuPIC open source community (numenta.org) for continuous support and enthusiasm about HTM.

10. REFERENCES

Ahmad S, Hawkins J (2015) Properties of sparse distributed representations and their application to hierarchical temporal memory. arXiv:1503.07469 Available at:

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

http://arxiv.org/abs/1503.07469.

Ahmad S, Hawkins J (2016) How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons and active dendrites. arXiv:1601.00720 [q-NC] Available at: http://arxiv.org/abs/1601.00720.

Ahmad S, Purdy S (2016) Real-Time Anomaly Detection for Streaming Analytics. arXiv:1607.02480 Available at: http://arxiv.org/abs/1607.02480 [Accessed October 18, 2016].

Antic SD, Zhou WL, Moore AR, Short SM, Ikonomu KD (2010) The decade of the dendritic NMDA spike. J Neurosci Res 88:2991–3001.

Baker CI, Peli E, Knouf N, Kanwisher NG (2005) Reorganization of visual processing in macular degeneration. J Neurosci 25:614–618 Available at: http://www.ncbi.nlm.nih.gov/pubmed/15659597 [Accessed October 31, 2016].

Bean BP (2007) The action potential in mammalian central neurons. Nat Rev Neurosci 8:451–465 Available at: http://www.ncbi.nlm.nih.gov/pubmed/17514198 [Accessed October 22, 2016].

Bibbig A, Wennekers T, Palm G (1995) A neural network model of the cortico-hippocampal interplay and the representation of contexts. Behav Brain Res 66:169–175 Available at: http://www.ncbi.nlm.nih.gov/pubmed/7755887 [Accessed February 2, 2017].

Billaudelle S, Ahmad S (2015a) Porting HTM models to the Heidelberg neuromorphic computing platform. arXiv:1505.02142 [q-NC] Available at: http://arxiv.org/abs/1505.02142 [Accessed October 23, 2015].

Billaudelle S, Ahmad S (2015b) Porting HTM Models to the Heidelberg Neuromorphic Computing Platform. arXiv:1505.02142 [q-NC] Available at: http://arxiv.org/abs/1505.02142 [Accessed October 23, 2015].

Buxhoeveden DP (2002) The minicolumn hypothesis in neuroscience. Brain 125:935–951 Available at: http://brain.oxfordjournals.org/content/125/5/935.long [Accessed September 2, 2014].

Clopath C, Büsing L, Vasilaki E, Gerstner W (2010) Connectivity reflects coding: a model of voltage-based STDP with homeostasis. Nat Neurosci 13:344–352 Available at: http://www.nature.com/doifinder/10.1038/nn.2479 [Accessed January 30, 2017].

Crochet S, Poulet JFA, Kremer Y, Petersen CCH (2011) Synaptic mechanisms underlying sparse coding of active touch. Neuron 69:1160–1175 Available at: http://www.ncbi.nlm.nih.gov/pubmed/21435560 [Accessed October 21, 2016].

Cui Y, Ahmad S, Hawkins J (2016a) Continuous online sequence learning with an unsupervised neural network model. Neural Comput 28:2474–2504 Available at: http://arxiv.org/abs/1512.05463.

Cui Y, Ahmad S, Hawkins J (2016b) Continuous Online Sequence Learning with an Unsupervised Neural Network Model. Neural Comput 28:2474–2504 Available at: http://www.mitpressjournals.org/doi/10.1162/NECO_a_00893 [Accessed November 13, 2016].

Cui Y, Liu L, McFarland J, Pack C, Butts D (2016c) Inferring Cortical Variability from Local Field Potentials. J Neurosci 36:4121–4135.

Davis GW (2006) Homeostatic control of neural activity: from phenomenology to molecular design. Annu Rev Neurosci 29:307–323 Available at: http://www.ncbi.nlm.nih.gov/pubmed/16776588 [Accessed October 18, 2016].

Faisal AA, Selen LPJ, Wolpert DM (2008) Noise in the nervous system. Nat Rev Neurosci 9:292–303 Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2631351&tool=pmcentrez&rendertype=abstract [Accessed April 28, 2014].

Favorov O V, Kelly DG (1996) Stimulus-response diversity in local neuronal populations of the cerebral cortex. Neuroreport 7:2293–2301 Available at: http://www.ncbi.nlm.nih.gov/pubmed/8951842 [Accessed October 18, 2016].

Földiák P (1990) Forming sparse representations by local anti-Hebbian learning. Biol Cybern 64:165–170 Available at: http://www.ncbi.nlm.nih.gov/pubmed/2291903 [Accessed October 18, 2016].

Földiák P (2002) Sparse coding in the primate cortex. In: The handbook of brain theory and neural networks, Second Edi. (Arbib M, ed), pp 1064–1068. MIT Press.

Friauf E, McConnell SK, Shatz CJ (1990) Functional synaptic circuits in the subplate during fetal and early postnatal development of cat visual cortex. J Neurosci 10:2601–2613 Available at: http://www.ncbi.nlm.nih.gov/pubmed/2388080 [Accessed October 18, 2016].

Gemmeke JF, Virtanen T, Hurmalainen A (2011) Exemplar-Based Sparse Representations for Noise Robust Automatic Speech Recognition. IEEE Trans Audio Speech Lang Processing 19:2067–2080 Available at: http://ieeexplore.ieee.org/document/5710402/ [Accessed January 27, 2017].

Ghosh A, Shatz CJ (1992) Involvement of subplate neurons in the formation of ocular dominance columns. Science 255:1441–1443 Available at: http://www.ncbi.nlm.nih.gov/pubmed/1542795 [Accessed October 18, 2016].

Gibson JR, Beierlein M, Connors BW (1999) Two networks of electrically coupled inhibitory neurons in neocortex. Nature 402:75–79 Available at: http://www.ncbi.nlm.nih.gov/pubmed/10573419 [Accessed October 29, 2016].

Gilbert CD, Wiesel TN (1992) Receptive field dynamics in adult primary visual cortex. Nature 356:150–152 Available at:

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

http://www.ncbi.nlm.nih.gov/pubmed/1545866 [Accessed October 22, 2016].

Gui J, Sun Z, Ji S, Tao D, Tan T (2016) Feature Selection Based on Structured Sparsity: A Comprehensive Study. IEEE Trans neural networks Learn Syst Available at: http://www.ncbi.nlm.nih.gov/pubmed/27116754 [Accessed October 22, 2016].

Habenschuss S, Puhr H, Maass W (2013) Emergence of optimal decoding of population codes through STDP. Neural Comput 25:1371–1407 Available at: http://www.mitpressjournals.org/doi/10.1162/NECO_a_00446 [Accessed January 30, 2017].

Hawkins J, Ahmad S (2016) Why neurons have thousands of synapses, a theory of sequence memory in neocortex. Front Neural Circuits 10 Available at: http://journal.frontiersin.org/article/10.3389/fncir.2016.00023/abstract [Accessed March 30, 2016].

Hawkins J, Ahmad S, Dubinsky D (2011) Cortical learning algorithm and hierarchical temporal memory. Numenta Whitepaper:1–68 Available at: http://numenta.org/resources/HTM_CorticalLearningAlgorithms.pdf.

Hebb D (1949) The organization of behavior: a neuropsychological theory. Psychology Press. Available at: http://www.amazon.com/dp/0805843000%5Cnhttp://www.ncbi.nlm.nih.gov/pmc/articles/PMC1279737/%5Cnhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:The+organization+of+behavior:+A+neuropsychological+theory#0%5Cnhttp://scholar.google.com/scholar?h.

Hecht-Nielsen R (1990) Neurocomputing. Addison-Wesley Pub. Co.

Horton JC, Adams DL (2005) The cortical column: a structure without a function. Philos Trans R Soc Lond B Biol Sci 360:837–862.

Hromádka T, Deweese MR, Zador AM (2008) Sparse representation of sounds in the unanesthetized auditory cortex. PLoS Biol 6:e16 Available at: http://www.ncbi.nlm.nih.gov/pubmed/18232737 [Accessed October 21, 2016].

Hu T, Pehlevan C, Chklovskii DB (2014) A Hebbian/Anti-Hebbian network for online sparse dictionary learning derived from symmetric matrix factorization. In: 2014 48th Asilomar Conference on Signals, Systems and Computers, pp 613–619. IEEE. Available at: http://ieeexplore.ieee.org/document/7094519/ [Accessed October 18, 2016].

Ibrayev T, James A, Merkel C, Kudithipudi D (2016) A design of HTM spatial pooler for face recognition using Memristor-CMOS hybrid circuits. In: IEEE International Symposium on Circuits and Systems.

Ito M (1989) Long-term depression. Annu Rev Neurosci 12:85–102 Available at: http://www.ncbi.nlm.nih.gov/pubmed/2648961 [Accessed October 29, 2016].

Jianchao Yang, Wright J, Huang TS, Yi Ma (2010) Image Super-Resolution Via Sparse Representation. IEEE Trans Image Process 19:2861–2873 Available at: http://ieeexplore.ieee.org/document/5466111/ [Accessed January 27, 2017].

Jones EG (2000) Microcolumns in the cerebral cortex. Proc Natl Acad Sci U S A 97:5019–5021 Available at: http://www.ncbi.nlm.nih.gov/pubmed/10805761 [Accessed October 18, 2016].

Kaas JH (1997) Topographic maps are fundamental to sensory processing. Brain Res Bull 44:107–112 Available at: http://www.ncbi.nlm.nih.gov/pubmed/9292198 [Accessed October 21, 2016].

Kanerva P (1988) Sparse Distributed Memory. The MIT Press.

Kanold PO, Kara P, Reid RC, Shatz CJ (2003) Role of subplate neurons in functional maturation of visual cortical columns. Science 301:521–525 Available at: http://www.ncbi.nlm.nih.gov/pubmed/12881571 [Accessed October 18, 2016].

Kremkow J, Perrinet LU, Monier C, Alonso J-M, Aertsen A, Frégnac Y, Masson GS (2016) Push-Pull Receptive Field Organization and Synaptic Depression: Mechanisms for Reliably Encoding Naturalistic Stimuli in V1. Front Neural Circuits 10:37 Available at: http://www.ncbi.nlm.nih.gov/pubmed/27242445 [Accessed January 26, 2017].

Lavin A, Ahmad S (2015) Evaluating real-time anomaly detection algorithms - the numenta anomaly benchmark. In: 14th International Conference on Machine Learning and Applications (IEEE ICMLA) Available at: http://arxiv.org/abs/1510.03336 [Accessed November 21, 2015].

Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2324 Available at: http://ieeexplore.ieee.org/document/726791/ [Accessed October 18, 2016].

Lee H, Battle A, Raina R, Ng AY (2006) Efficient sparse coding algorithms. In: Advances in Neural Information Processing Systems, pp 801–808.

Mairal J, Bach F, Ponce J, Sapiro G (2010) Online Learning for Matrix Factorization and Sparse Coding. J Mach Learn Res 11:19–60.

Majani E, Erlanson R, Abu-Mostafa Y (1988) On the K-winners-take-all network. Proc 1st Int Conf Neural Inf Process Syst:634–642.

Major G, Larkum ME, Schiller J (2013) Active properties of neocortical pyramidal neuron dendrites. Annu Rev Neurosci 36:1–24 Available at: http://www.ncbi.nlm.nih.gov/pubmed/23841837.

Makhzani A, Frey B (2015) k-Sparse Autoencoders. In: Advances in Neural Information Processing Systems 28, pp 2791--2799.

Markram H, Toledo-Rodriguez M, Wang Y, Gupta A, Silberberg G, Wu C (2004) Interneurons of the neocortical inhibitory system. Nat Rev Neurosci

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

5:793–807 Available at: http://www.ncbi.nlm.nih.gov/pubmed/15378039 [Accessed October 18, 2016].

Masquelier T (2013) Neural variability, or lack thereof. Front Comput Neurosci 7:7 Available at: http://www.frontiersin.org/Journal/10.3389/fncom.2013.00007/abstract [Accessed April 4, 2014].

Mnatzaganian J, Fokoué E, Kudithipudi D (2017) A mathematical formalization of hierarchical temporal memory's spatial pooler. Front Robot AI:in press.

Mountcastle VB (1997) The columnar organization of the neocortex. Brain 120:701–722 Available at: http://www.ncbi.nlm.nih.gov/pubmed/9153131 [Accessed October 7, 2011].

Nudo RJ (2013) Recovery after brain injury: mechanisms and principles. Front Hum Neurosci 7:887 Available at: http://www.ncbi.nlm.nih.gov/pubmed/24399951 [Accessed October 18, 2016].

Olshausen BA, Field DJ (1996a) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381:607–609 Available at: http://www.ncbi.nlm.nih.gov/pubmed/8637596 [Accessed December 8, 2014].

Olshausen BA, Field DJ (1996b) Natural image statistics and efficient coding. Netw Comput Neural Syst 7:333–339.

Olshausen BA, Field DJ (1997) Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Res 37:3311–3325.

Olshausen BA, Field DJ (2004) Sparse coding of sensory inputs. Curr Opin Neurobiol 14:481–487.

Pati N, Pradhan A, Kanoje LK, Das TK (2015) An Approach to Image Compression by Using Sparse Approximation Technique. Procedia Comput Sci 48:769–775.

Perrinet LU (2010) Role of homeostasis in learning sparse representations. Neural Comput 22:1812–1836 Available at: http://www.ncbi.nlm.nih.gov/pubmed/20235818 [Accessed October 22, 2016].

Pietroń M, Wielgosz M, Wiatr K (2016) Formal analysis of HTM spatial pooler performance under predefined operation conditions. In, pp 396–405. Springer International Publishing. Available at: http://link.springer.com/10.1007/978-3-319-47160-0_36 [Accessed October 21, 2016].

Porter JT, Johnson CK, Agmon A (2001) Diverse types of interneurons generate thalamus-evoked feedforward inhibition in the mouse barrel cortex. J Neurosci 21:2699–2710 Available at: http://www.ncbi.nlm.nih.gov/pubmed/11306623 [Accessed October 29, 2016].

Purdy S (2016) Encoding Data for HTM Systems. arXiv:1602.05925.

Sivaram GSVS, Nemala SK, Elhilali M, Tran TD, Hermansky H (2010) Sparse coding for speech recognition. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, pp 4346–4349. IEEE. Available at: http://ieeexplore.ieee.org/document/5495649/ [Accessed January 27, 2017].

Song S, Miller KD, Abbott LF (2000) Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. Nat Neurosci 3:919–926 Available at: http://www.ncbi.nlm.nih.gov/pubmed/10966623 [Accessed October 29, 2016].

Spruston N (2008) Pyramidal neurons: dendritic structure and synaptic integration. Nat Rev Neurosci 9:206–221 Available at: http://www.ncbi.nlm.nih.gov/pubmed/18270515 [Accessed July 10, 2014].

Swadlow HA (2002) Thalamocortical control of feed-forward inhibition in awake somatosensory "barrel" cortex. Philos Trans R Soc Lond B Biol Sci 357:1717–1727 Available at: http://www.ncbi.nlm.nih.gov/pubmed/12626006 [Accessed October 29, 2016].

Teyler TJ, DiScenna P (1987) Long-term potentiation. Annu Rev Neurosci 10:131–161 Available at: http://www.ncbi.nlm.nih.gov/pubmed/3032063 [Accessed October 29, 2016].

Thornton J, Srbic A (2011) Spatial pooling for greyscale images. Int J Mach Learn Cybern 2:1–10.

Tolhurst DJ, Movshon JA, Dean AF (1983) The statistical reliability of signals in single neurons in cat and monkey visual cortex. Vis Res 23:775–785 Available at: http://www.ncbi.nlm.nih.gov/pubmed/6623937 [Accessed February 11, 2015].

Udin SB, Fawcett JW (1988) Formation of topographic maps. Annu Rev Neurosci 11:289–327 Available at: http://www.ncbi.nlm.nih.gov/pubmed/3284443 [Accessed October 21, 2016].

Vinje WE, Gallant JL (2000) Sparse coding and decorrelation in primary visual cortex during natural vision. Science 287:1273–1276 Available at: http://www.ncbi.nlm.nih.gov/pubmed/10678835 [Accessed October 21, 2016].

Weliky M, Fiser J, Hunt RH, Wagner DN (2003) Coding of natural scenes in primary visual cortex. Neuron 37:703–718 Available at: http://www.ncbi.nlm.nih.gov/pubmed/12597866 [Accessed October 21, 2016].

Willmore B, Tolhurst DJ (2001) Characterizing the sparseness of neural codes. Network 12:255–270 Available at: http://www.ncbi.nlm.nih.gov/pubmed/11563529 [Accessed October 18, 2016].

Willshaw DJ, Buneman OP, Longuet-Higgins HC (1969) Non-holographic associative memory. Nature 222:960–962 Available at: http://www.ncbi.nlm.nih.gov/pubmed/5789326 [Accessed January 30, 2017].

Wonders CP, Anderson SA (2006) The origin and specification of cortical interneurons. Nat Rev Neurosci 7:687–696 Available at: http://www.ncbi.nlm.nih.gov/pubmed/16883309 [Accessed October 18, 2016].

bioRxiv preprint doi: https://doi.org/10.1101/085035; this version posted February 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Wright J, Yang AY, Ganesh A, Sastry SS, Yi Ma (2009) Robust Face Recognition via Sparse Representation. IEEE Trans Pattern Anal Mach Intell 31:210–227 Available at: http://ieeexplore.ieee.org/document/4483511/ [Accessed January 27, 2017].

Zito K, Svoboda K (2002) Activity-dependent synaptogenesis in the adult Mammalian cortex. Neuron 35:1015–1017 Available at: http://www.ncbi.nlm.nih.gov/pubmed/12354392 [Accessed December 15, 2014].

Zylberberg J, Murphy JT, DeWeese MR (2011) A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of V1 simple cell receptive fields. PLoS Comput Biol 7:e1002250 Available at: http://www.ncbi.nlm.nih.gov/pubmed/22046123 [Accessed October 22, 2016].