Sparse distributed associative memory

Yuwei Cui
ycui@numenta.com

Numenta

Outline

Associative memory

Hopfield network / Boltzmann machine

Sparse distributed associative network
- Capacity experiment
- Simultaneously hold multiple items given ambiguous input
- Narrowing down the union with dynamic sparsity
- Matches listed criterion for single Module in L2 pooling

Ensemble of sparse associative network
- Matches some listed criterion for multiple modules in L2 pooling

Associative memory

Generic term that refers to all memories that enable one to retrieve a piece of data from only partial information.

Autoassociative memory:
Retrieving a piece of data upon presentation of only partial information from *that* piece of data
Also known as content-addressable memory, associative storage
Example: Hopfield network, Boltzmann Machine

Heteroassociative memory:
Recall an associated piece of datum from one category upon presentation of data from another category
Example: Artificial neural network

"We propose a simple duality between this dense associative memory and neural networks commonly used in deep learning." (Krotov and Hopfield 2016)

Hopfield network

Fully connected, single layer, recurrent neural network

Binary threshold neurons with discrete time

States of the network is represented by a binary vector (-1/1 or 0/1)

Memory corresponds to specific states of the network (local minima/attractor)

Memory retrieval begins from a clue – an initial state describing partial information, and settles down to a stable state (fixed point attractor).

“The network responds to an ambiguous starting state by a statistical choice between the memory states it most resembles” (but we want a union!)

Hopfield network

Activation rule:

$$\begin{matrix} V_i \to 1 \\ V_i \to 0 \end{matrix} \quad \text{if} \quad \sum_{j \neq i} T_{ij} V_j \quad \begin{matrix} > U_i \\ < U_i \end{matrix} \; . \qquad [1]$$

Neurons can be updated asynchronously or synchronously
Gradient descent of the "energy" function when weights are symmetric

Learning rule:

$$\Delta T_{ij} = [V_i(t)V_j(t)]_{\text{average}} \qquad [6]$$

The learning rule decreases the "energy" of the memory states, making them local minima in the state space

Note*: Boltzmann machine is a stochastic, generative counterpart of Hopfield nets*

Limitation of Hopfield network

Severe limit to the number of stored pattern (M~0.138N)

Spurious patterns may exist (local minimum that are not memory), which seems to grow exponentially with M

Not designed to simultaneously represent multiple items

Hopfield network used dense coding.

Could the use of SDRs alleviate the limitations?

"Sparseness of the stored patterns is most important for an effective use of NAMs for information storage and retrieval" (Willshaw 1971; Palm, 88, 90; Tsodyks & Feigelman 88)

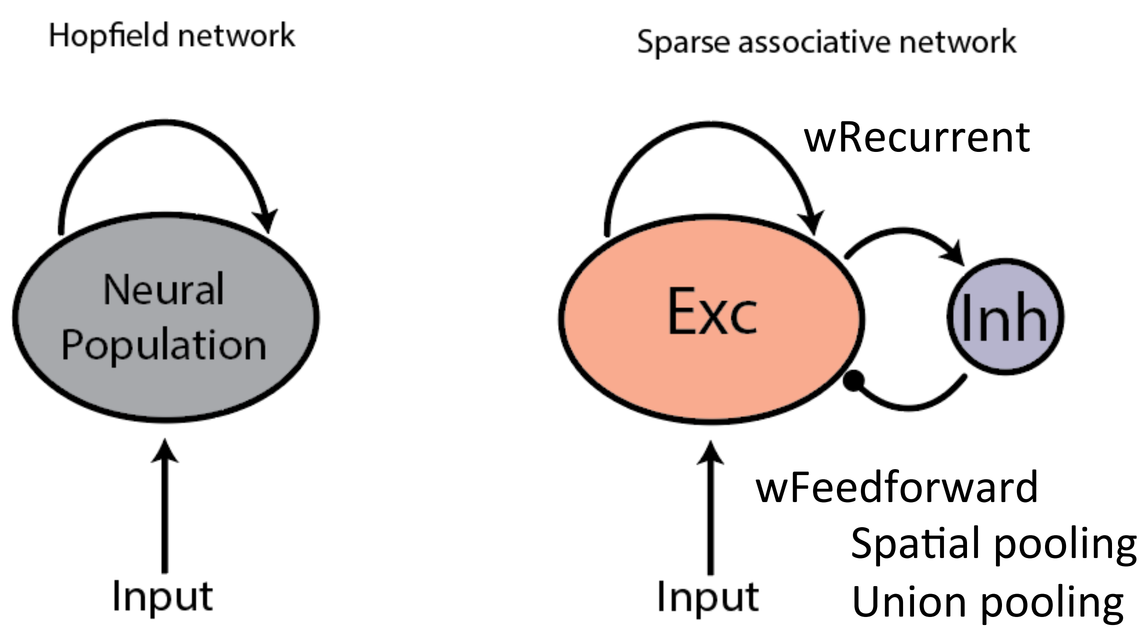
Sparse distributed associative network

An neuron will become active if

its total input > minThreshold, and

it's among the top *w* neurons with the most inputs

Sparsity *w* is regulated by a population of inhibitory neurons



Capacity experiment

Experiment 1

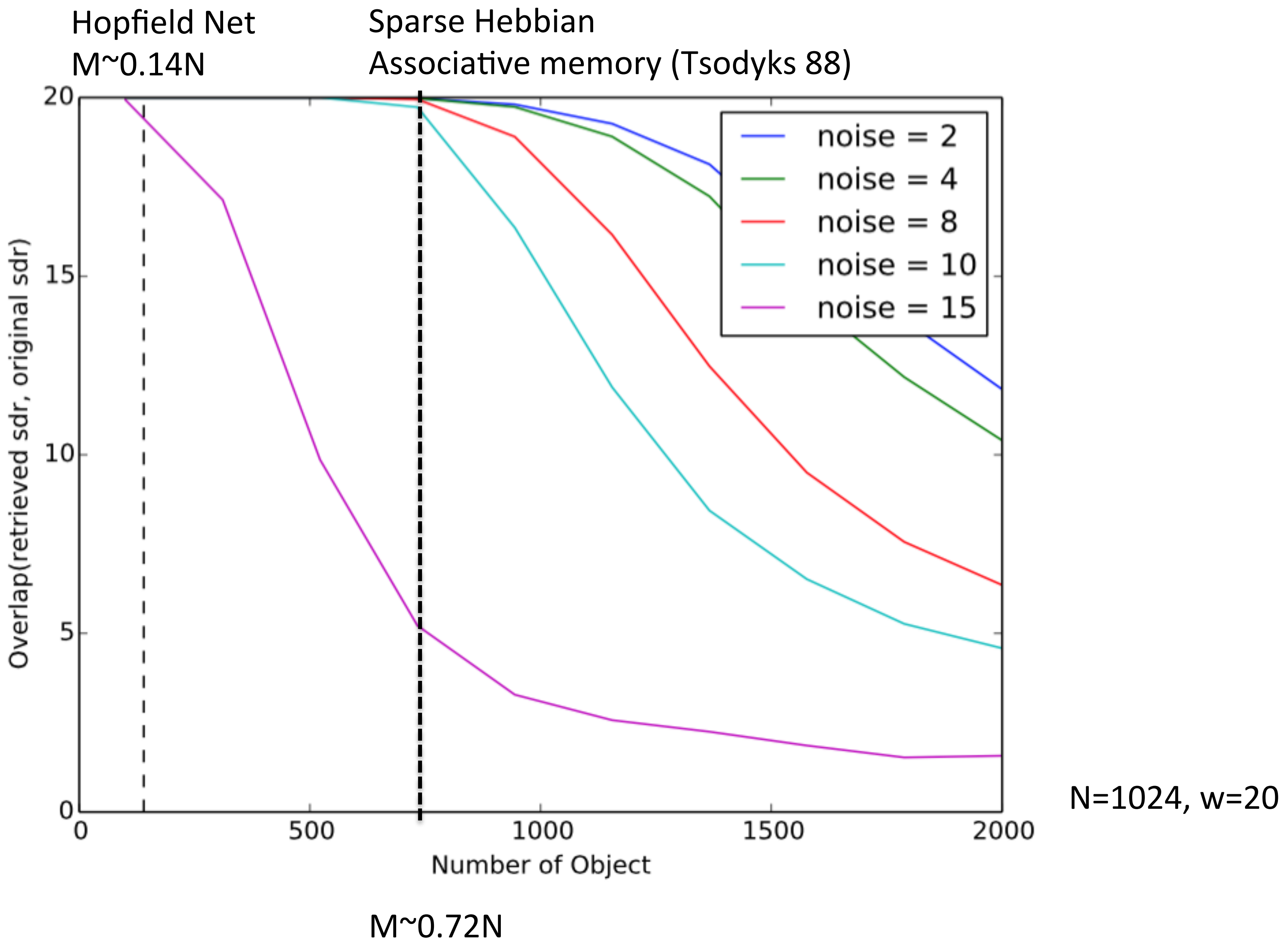
Store M object SDRs using the Hebbian learning rule

Each object is represented by w active bits out of N neurons

Provide a test object SDR that is corrupted by E noise bits

Does the network settle into the original object SDR?

How does the retrieval performance change as a function of M and E?


Hopfield Net
M~0.14N
Sparse Hebbian
Associative memory (Tsodyks 88)
noise = 2
noise = 4
noise = 8
noise = 10
noise = 15
Overlap(retrieved sdr, original sdr)
Number of Object
0
5
10
15
20
0
500
1000
1500
2000
M~0.72N
N=1024, w=20

Simultaneously retrieve multiple elements

Experiment 2

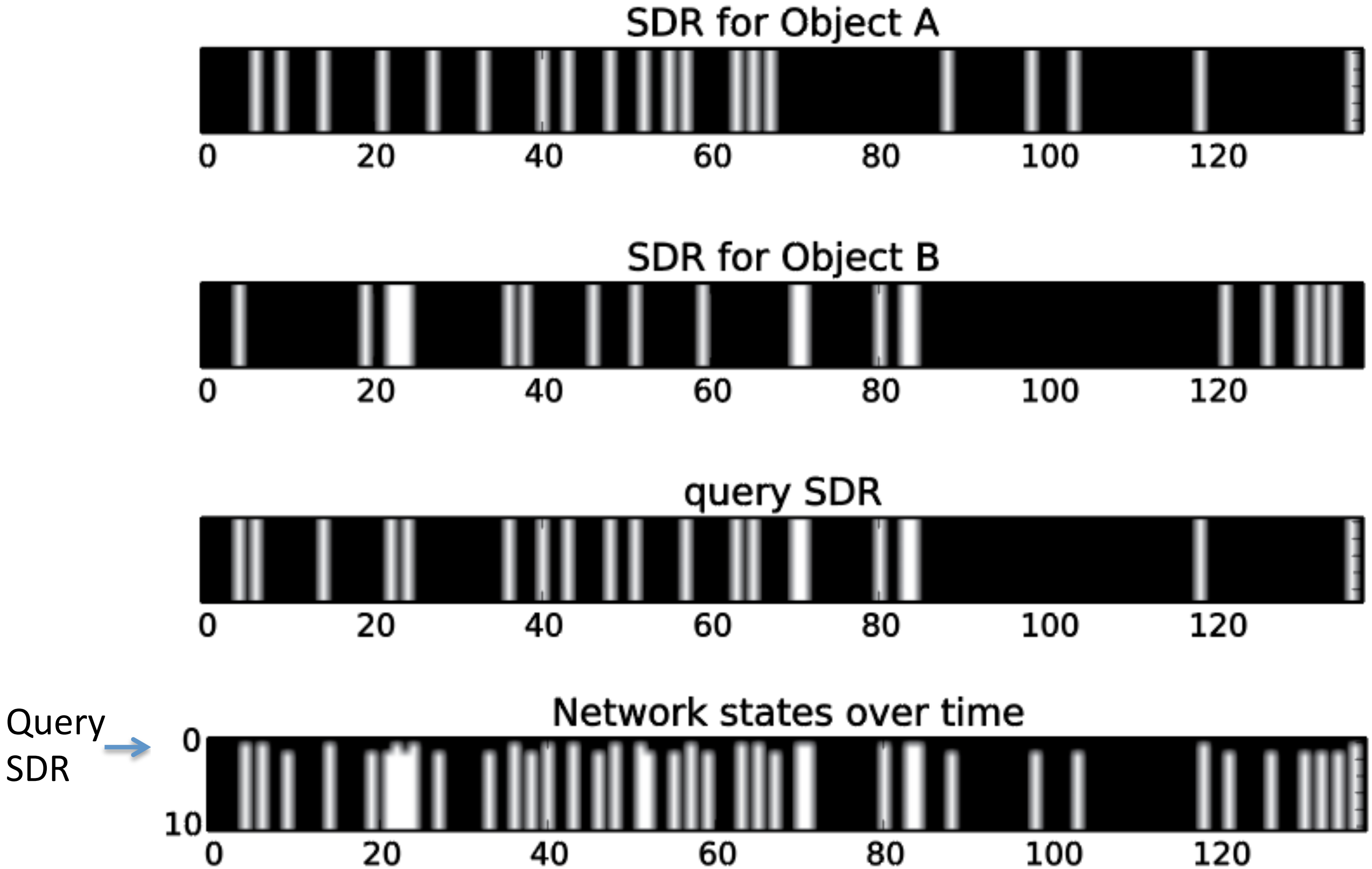
Store M object SDRs using the Hebbian learning rule

Each object is represented by w active bits out of N neurons

Provide a test SDR that is partially consistent with multiple objects

(e.g., 10 bits of object A and 10 bits of object B).

Can we retrieve multiple memory items simultaneously?



Note:

SDRs are sampled for display purpose (really SDRs are much more sparse)

Query SDR is present only at t=0. The network state are persistent due to recurrent excitatory connections, not due to “hysteresis”

Narrow down union with more sensory inputs

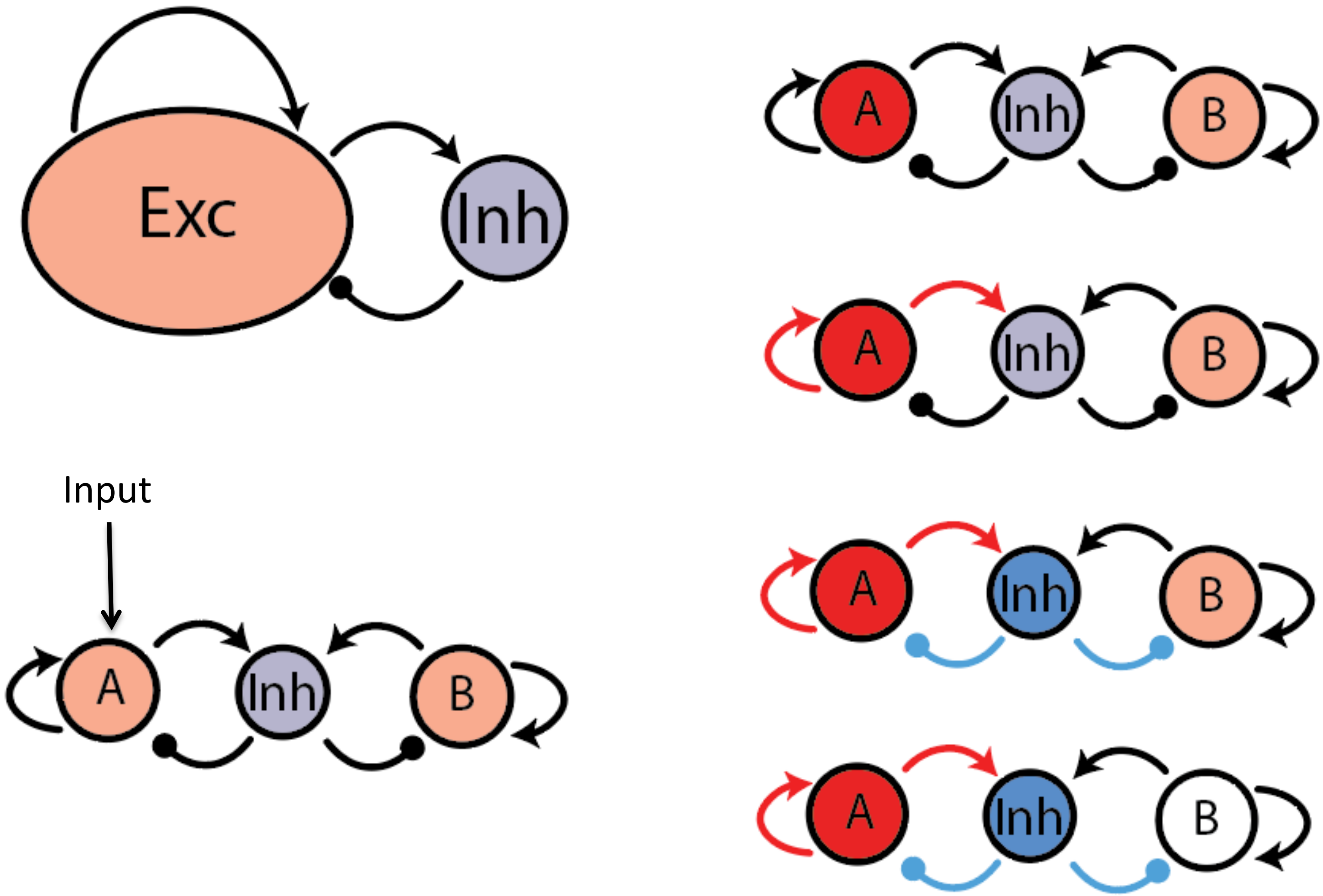
Experiment 3

The network state represents multiple memory items

Given further sensory inputs, can we narrow down the union?

Idea: the "match" between existing network states and feedforward inputs caused stronger activation, which tightens the sparsity

Narrow down union with more sensory inputs



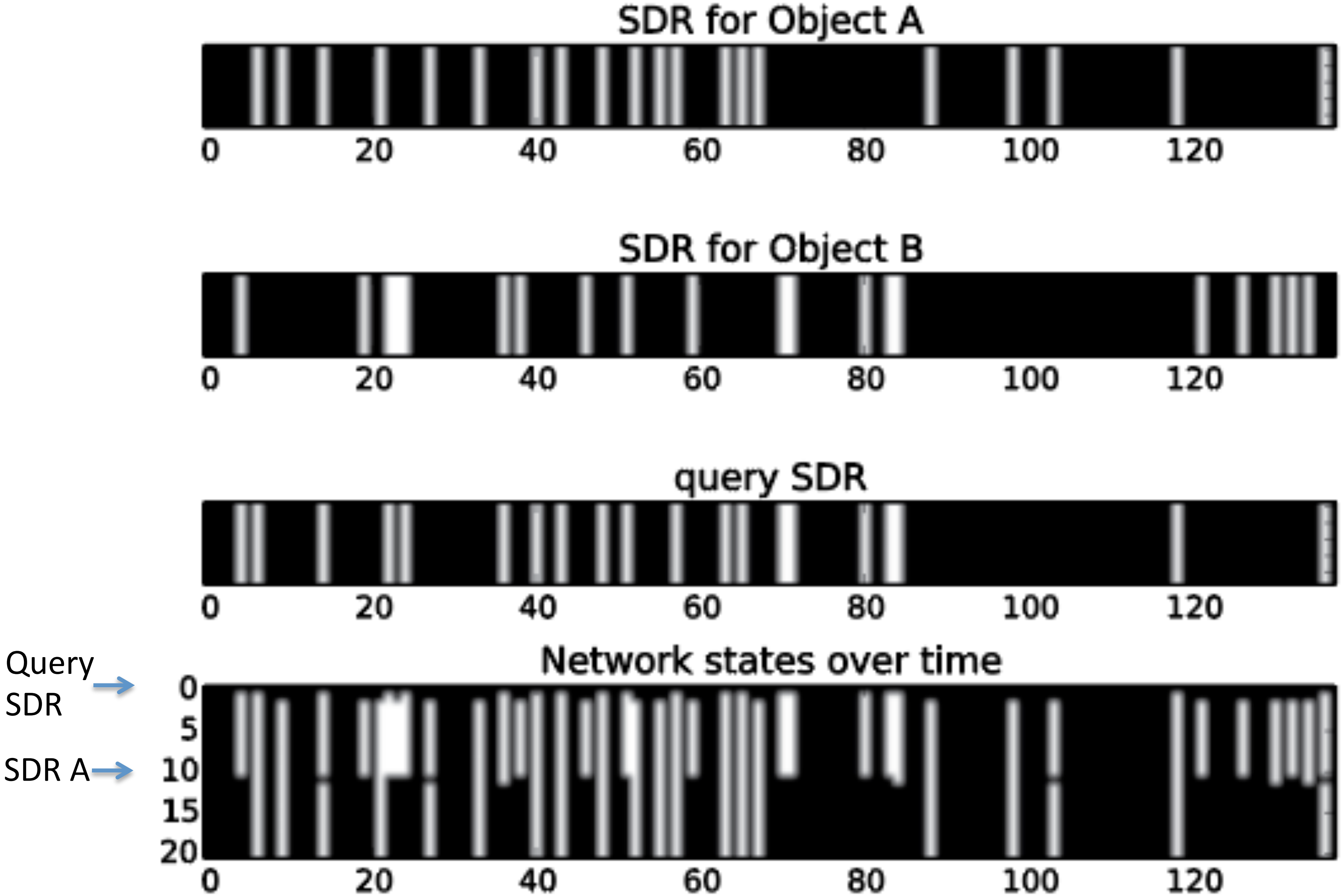
Dynamic sparsity

Sparsity w depends on the number of strongly activated neurons

Strongly activated neurons: previously active neurons that also receives strong feedforward sensory inputs (> minThreshold)

w decreases as the number of strongly activated neurons increases

w recovers if the number of strongly activated neurons decreases

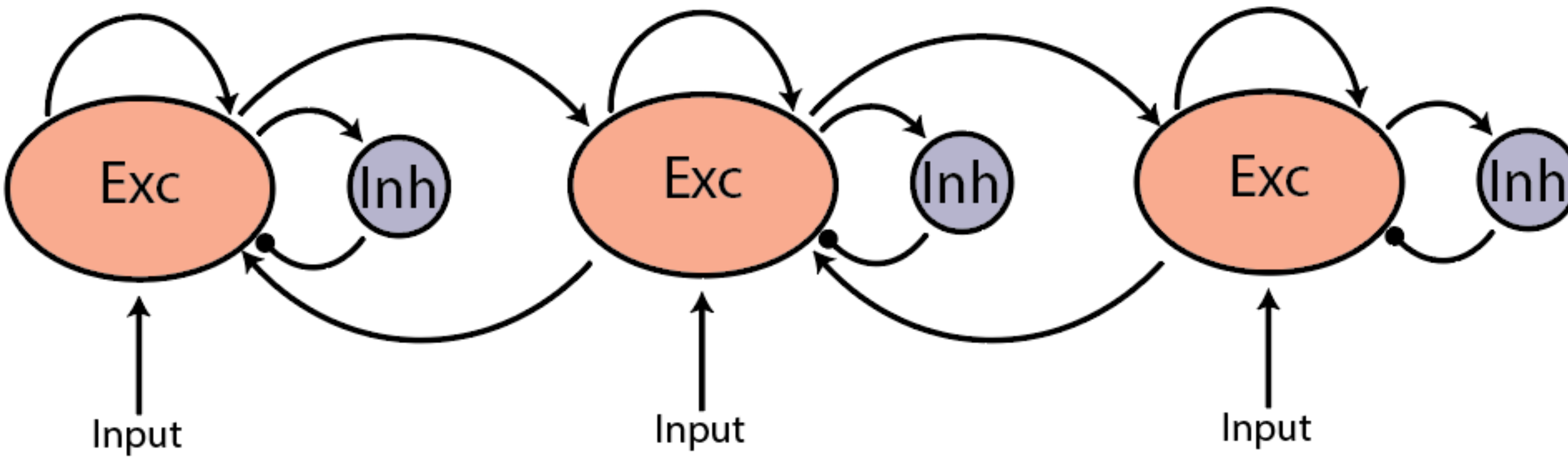


Note: the sparsity w will recover if the feedforward input for A stops. If we feed "B" in at t=20 The network will represent a union of A & B.

Single HC.

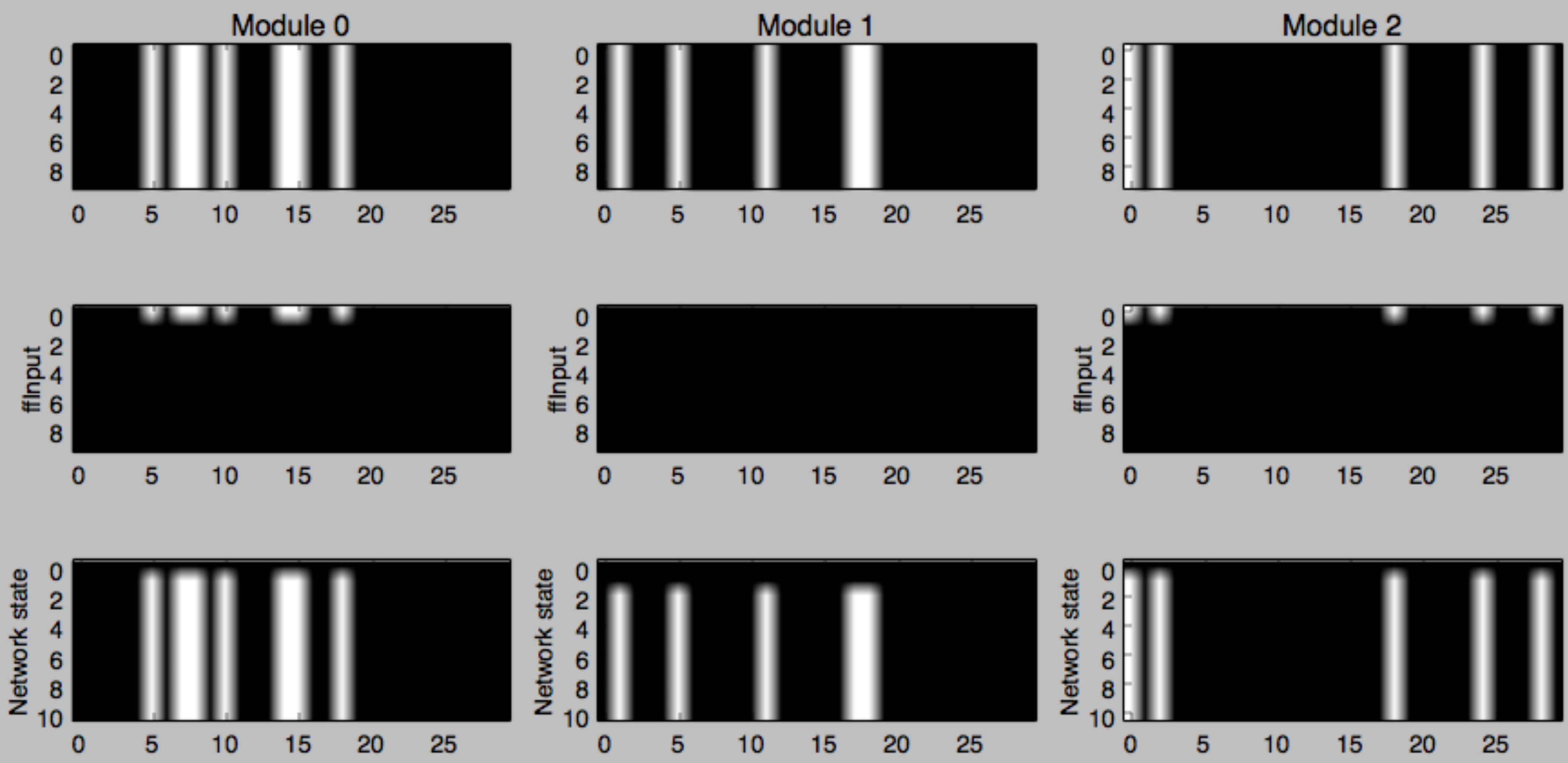
1. learn L4 Rep. of obj+loc pairs (GTM)
2. learn a pooled Rep of L4 in L2
3. learn a bunch of objects w/some shared features+loc's.
4. Given one loc+feature invoke a union of L4+L2 repr. that are consistent.
5. As you get sequence, narrow down to unique Repr. in L2
6. with no input maintain L2 for awhile.
7. Feedback L2 to L4 to improve L4(??)

Ensemble of sparse associative memory



Ensemble of sparse associative memory

A	**?**	**A**
A	A	A



Multiple HC's

1. HC's learn L2 Repr's of nearby HC's for same object using lateral connections.

2. Given single sensation HC's w/ input invoke union rep in L2. HC's w/o input also invoke union

3. Most consistent representations win

   ABC, A; A → AB, A, A

4. With multiple sensations quickly converge on unique repr. across all L2 HC's.

5. System should tolerate some noise

   A, noise, A → A, A, A

Boltzmann network

Stochastic, generative counterpart of Hopfield nets

Activation rule:

$$prob(s_i = 1) \;=\; \frac{1}{1+e^{-z_i}} \qquad\qquad z_i = b_i + \sum_j s_j w_{ij}$$

Learning rule:

$$\left\langle \frac{\partial \log P(\mathbf{v})}{\partial w_{ij}} \right\rangle_{\text{data}} = \langle s_i s_j \rangle_{\text{data}} - \langle s_i s_j \rangle_{\text{model}}$$

The learning rule decreases the “energy” of the memory states, making them local minima in the state space