HIERARCHICAL TEMPORAL MEMORY

Fred Rotbart

Pycon Israel 2019

rotbart@softbart.com
https://www.linkedin.com/in/fcrotbart
https://github.com/fcr
@rotbartfc

Deep Neural Nets

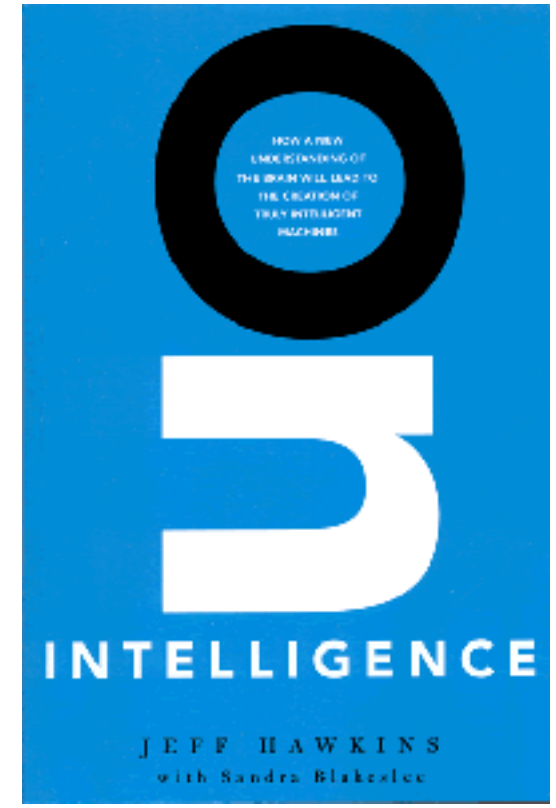
- Deep Neural Nets (DNN) are frameworks for ~~**A**rtificial **I**ntelligence~~
  Assistive **I**ntelligence
- Deep Neural Nets have clocked up incredible successes in many areas, (and continue to do so), however,
  - DNN need thousands if not millions of samples to train on
  - DNN find it hard to adapt to continually changing data and surprises
  - DNN are susceptible to noise and can easily be fooled
- DNN don’t work as our brains do and cannot lead to true AI

HTM

- Hierarchical Temporal memory (HTM) is a theoretical framework for both biological and generalised machine intelligence.
- Based on the latest understanding of the Neocortex
- Only requires a few hundred samples to learn
- Learns unsupervised as it goes and easily handles changing data and surprises
- Immune to up to 40% noise
- Opens the way for truly intelligent systems

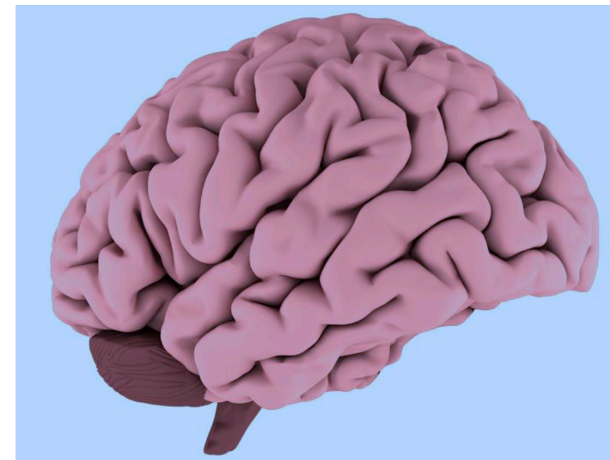
History

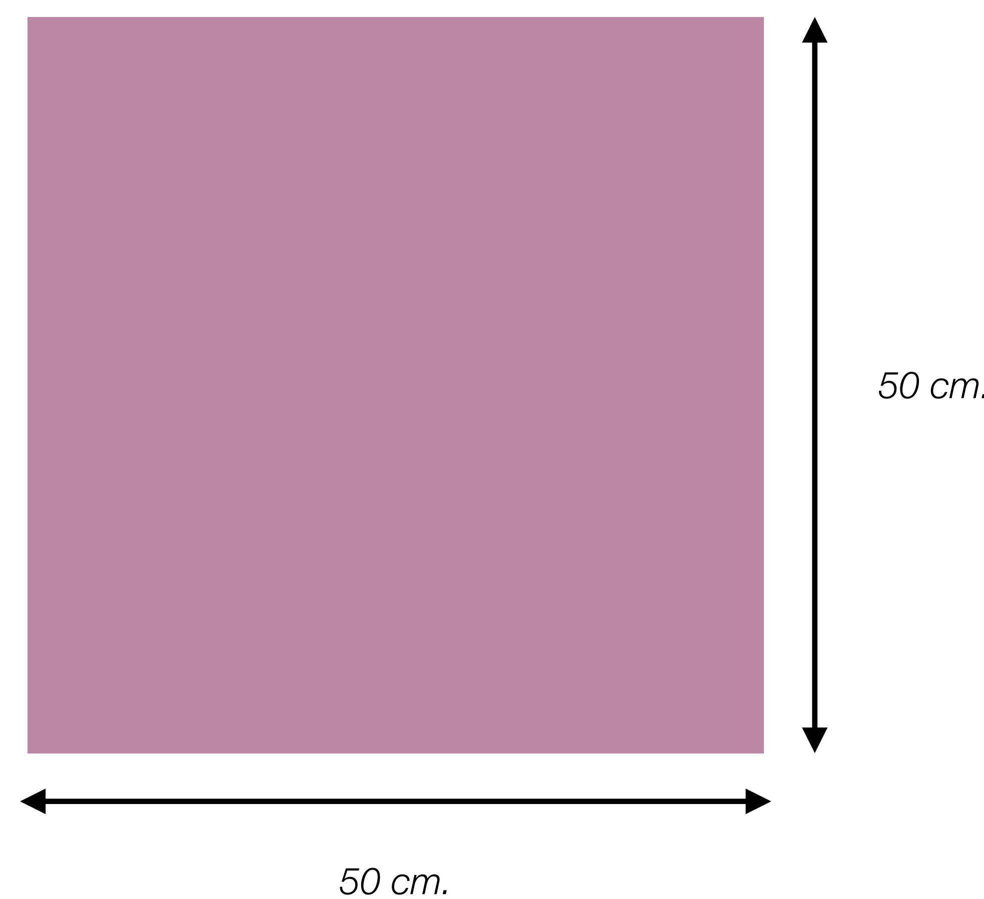
- 2004 “On Intelligence” by Jeff Hawkins & Sandra Blakeslee
  - The core concepts in Hierarchical Temporal Memory (HTM) theory were first described in this book
- 2005 Numenta was established in Redwood City, CA to
  - **Reverse engineer the neocortex**
  - **Apply neocortical theory to AI**
- 2014 NuPIC (**Nu**menta **P**latform for **I**ntelligent **C**omputing) was open sourced under the AGPLv3 license
  - API in Python 2.7, and C++
- Community port to Python 3 (in progress)
- Third Party Implementations in Java, Closure, C#, Javascript…
- HTM is constantly evolving with Numenta’s open research



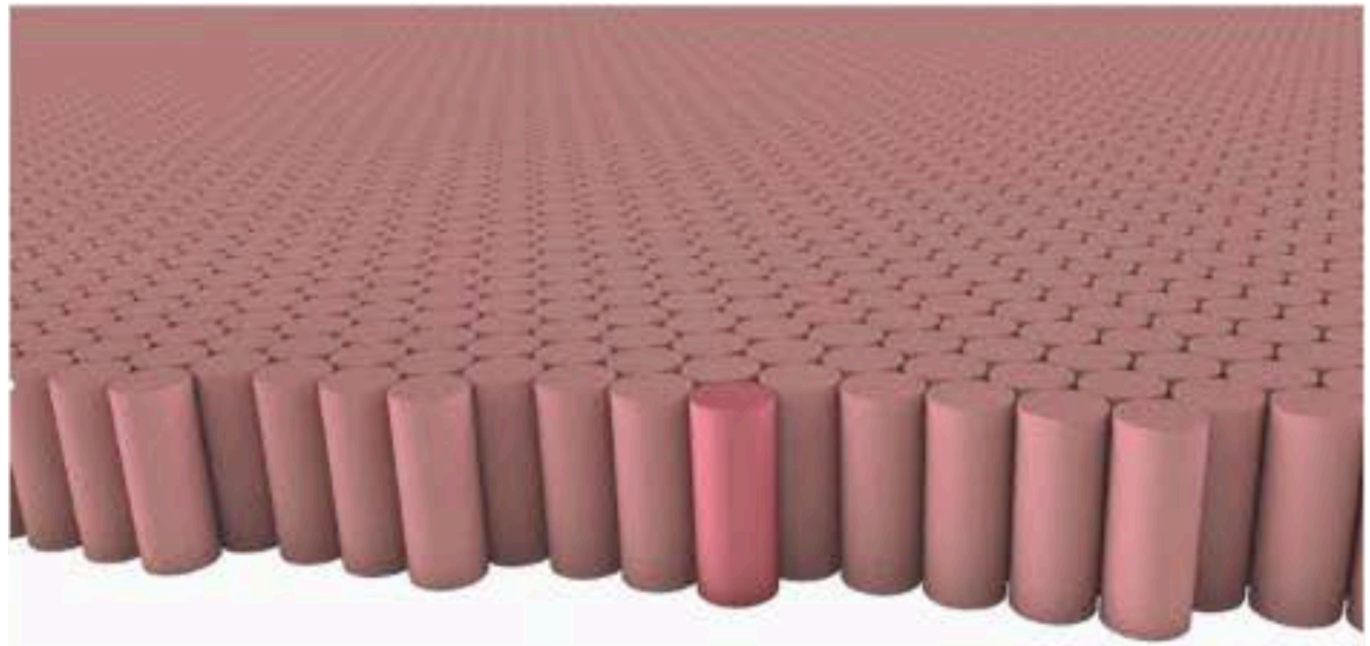
Neocortex

- Size of a large table napkin (50x50 square centimetres)
- 75% of brain's volume
- 2.5 mm thick
- 30 billion neurons
- Tens of thousands of synapses per neuron
- Sparsely active
  - Only ~ 2% spiking at any one time
- Constantly predicting its inputs
- Learns a model of the world
- Creates behaviours



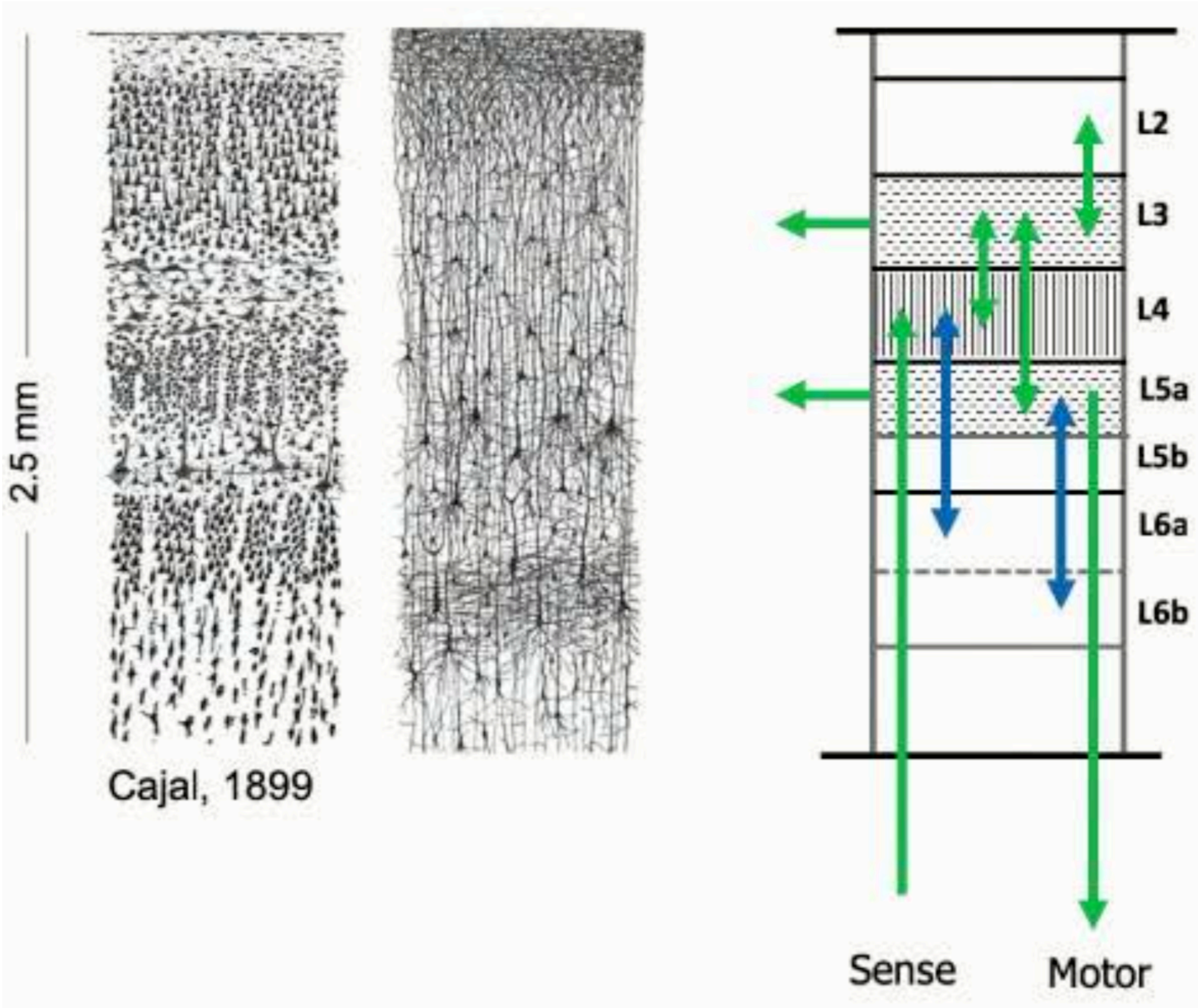


Neocortex Structure



- All areas in the neocortex look the same, so they must perform the same basic function
- What makes one region visual and an another touch is what is connected to them
- The basic unit of replication is a cortical column (1mm^2)
  - About 2,000,000 of them
  - So logically the cortical column is the basic unit of **computation**

Cortical Column



Dozens of neuron types

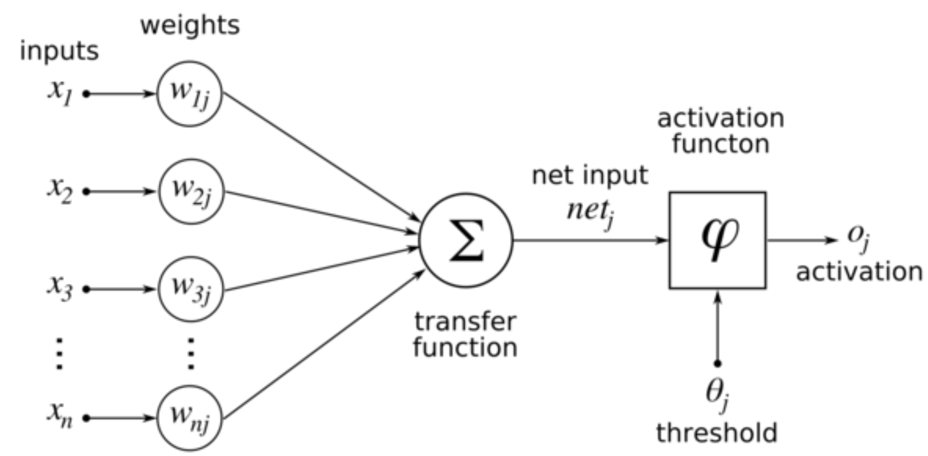
Organised into layers

Vertical local projections cross all layers

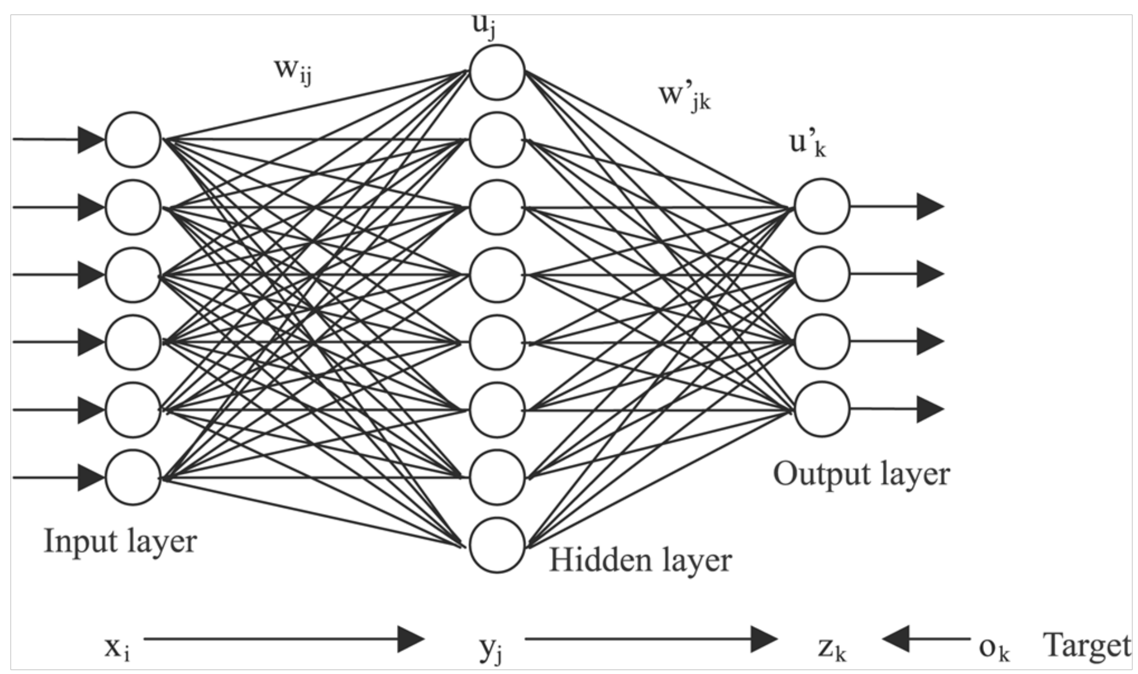
Horizontal inter-column long distance projections in some layers

- Cortical Columns are complex
- So whatever the column does must also be complex
- And whatever the column does, so does the neocortex.

Deep Neural Net Neuron



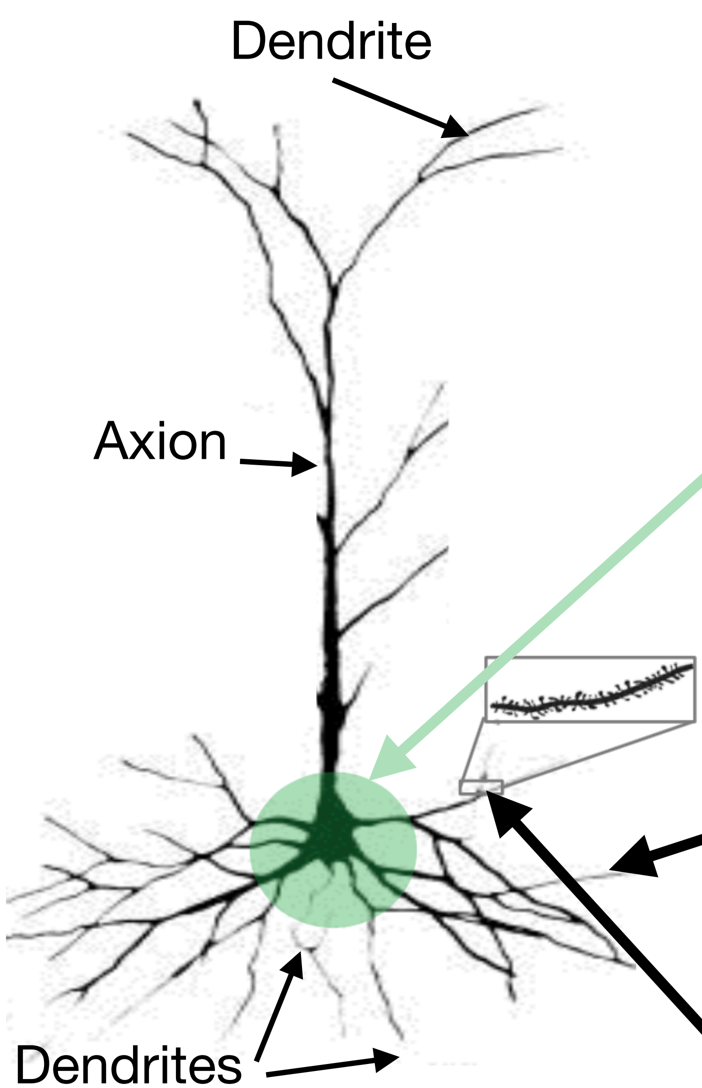
- Based on the 1957 concept of the Perceptron



- Learning is by adjusting the synaptic weights

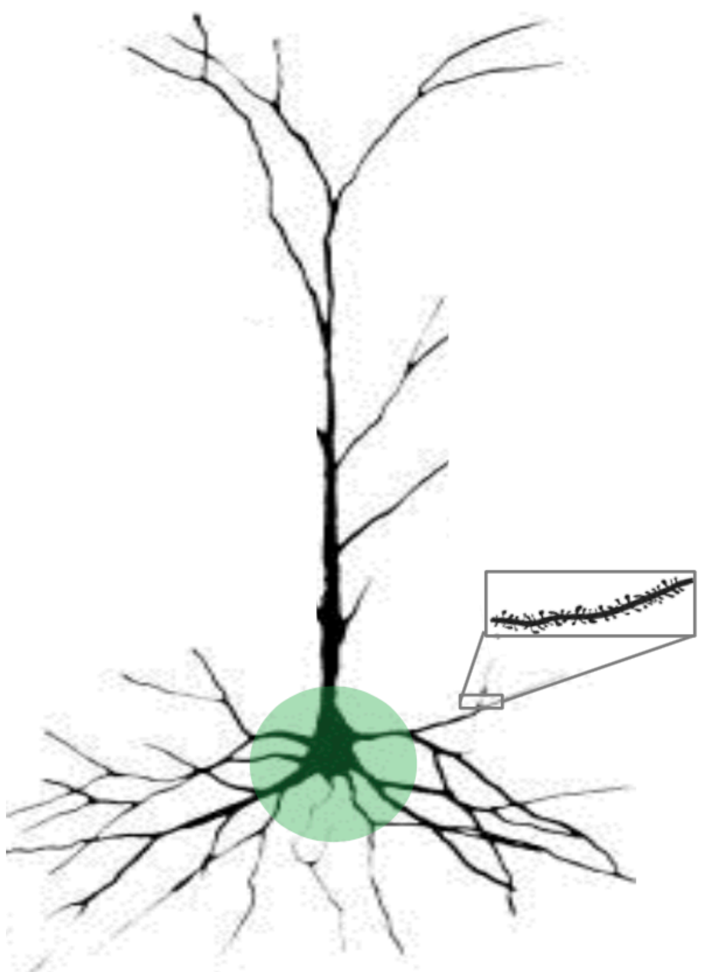
- Real neurons are not like this

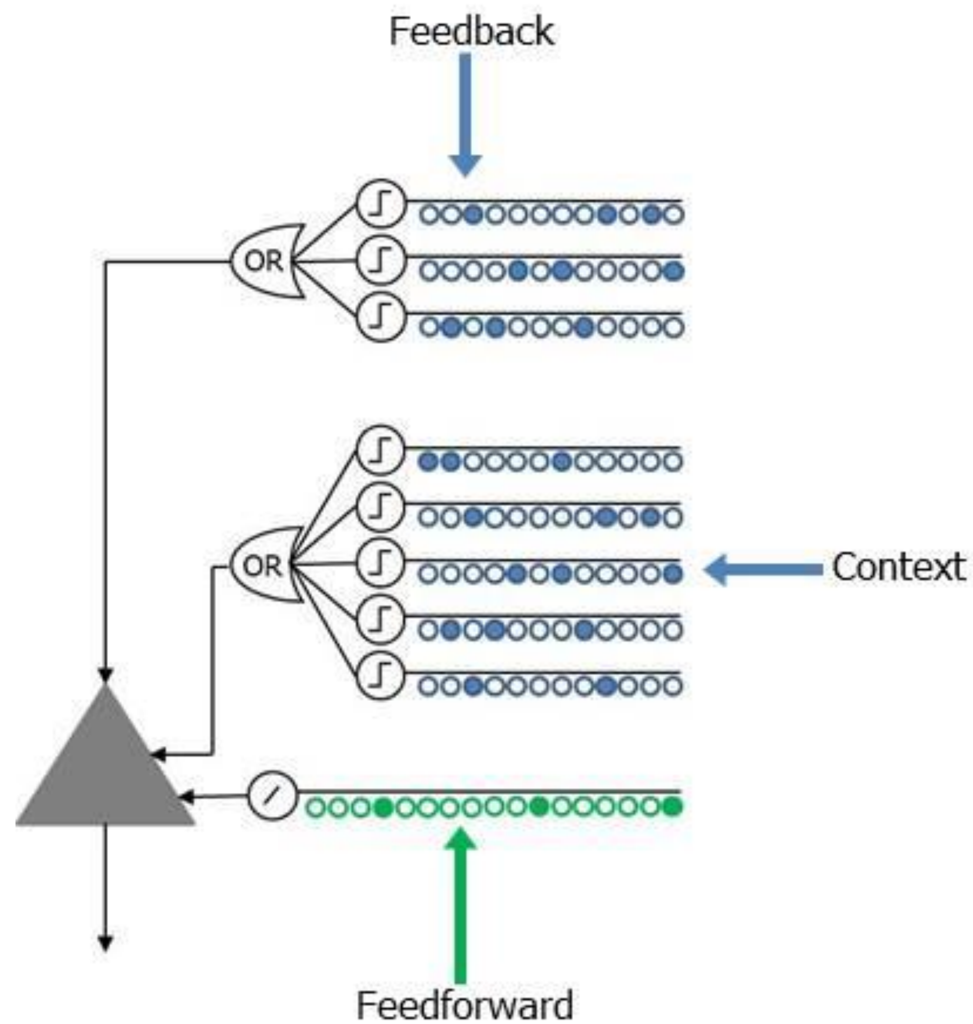
Real Neuron



- 5K to 30K excitatory synapses on the dendrites
    - 10% proximal can cause **neural** spike
    - 90% distal cannot cause neural spike
- Distal dendrites are pattern detectors
    - 8-15 co-active, co-located synapses will generate a **dendritic** spike
        - this puts the cell into a depolarised, or “predictive” state
    - Depolarised neurons fire sooner, inhibiting nearby neurons.

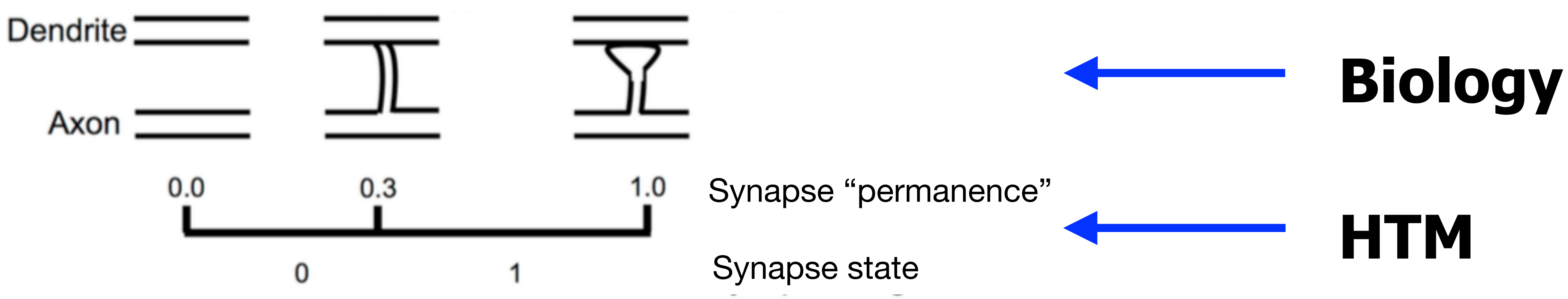
HTM Neuron





- HTM neurons don’t attempt to model all aspects of biological neurons
- Only those that are essential for the informational aspects of the neocortex.
- HTM neuron state depends on the position and number of activated synapses - **not on a sum of weights**

Neural Learning



- In HTM neurons, learning is modelled by the growth of new synapses or removal of unused synapses as in biological neurons

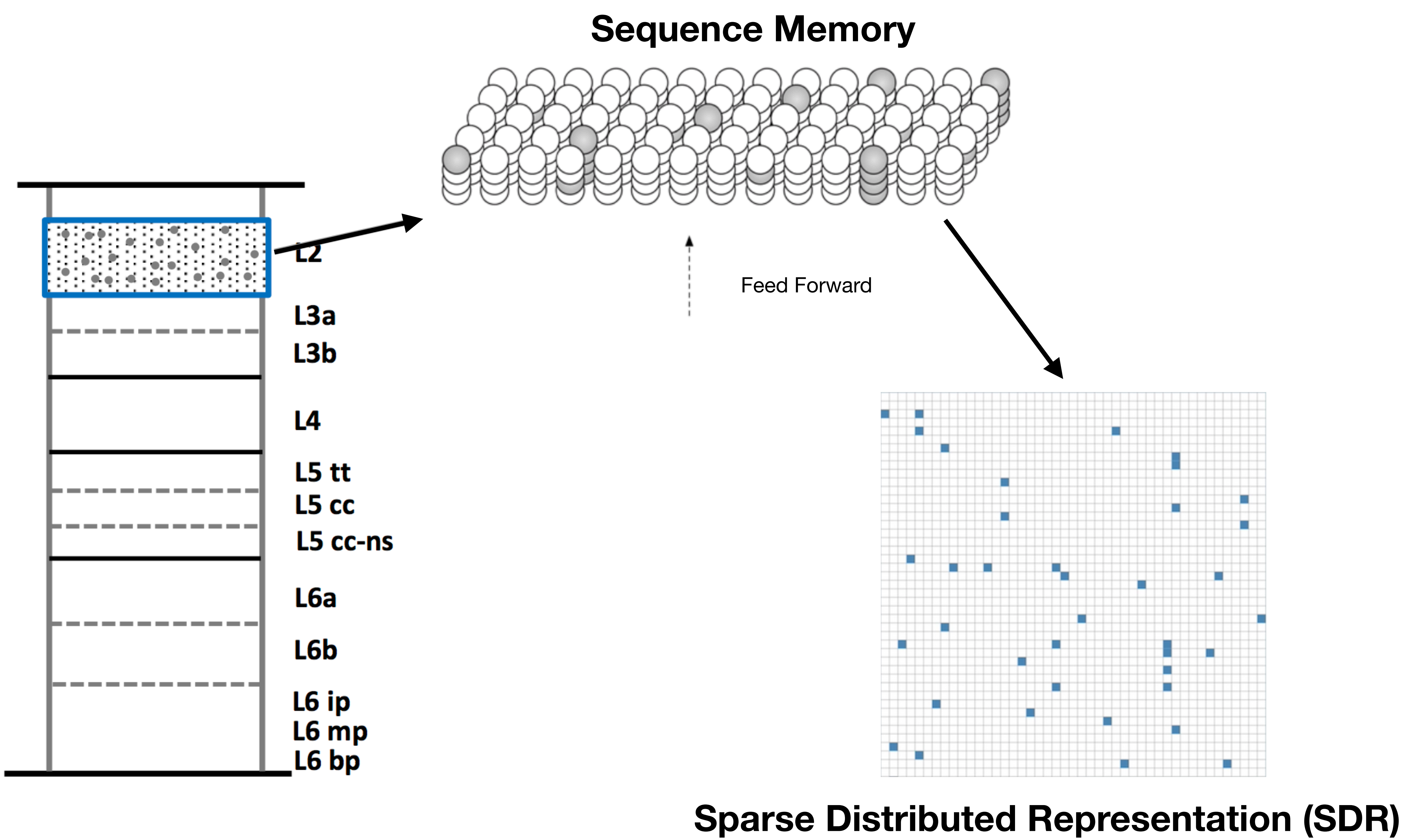
- This learning occurs by incrementing or decrementing the synapse “permanence”.

- A synapse is **disconnected** for a permanence **under** the threshold

- A synapse is **connected** for a permanence **over** the threshold

- **Learning is making or breaking synapses, not adjusting synaptic weights as in DNNs**

HTM Cortical Column



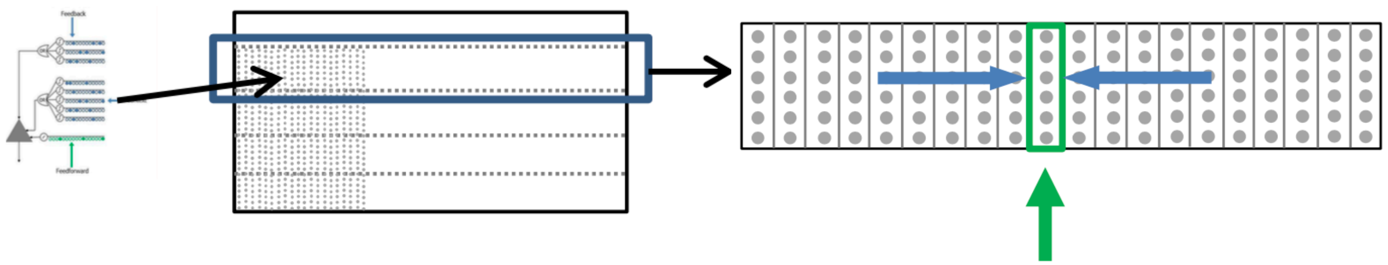
Sparse Distributed Representations

- SDRs are how brains solve the problem of representing knowledge
- Each bit has semantic meaning
- Extremely high capacity. For 2048 bit vector and 2% are set, we have >> 10^{84} unique patterns
- Fixed sparseness

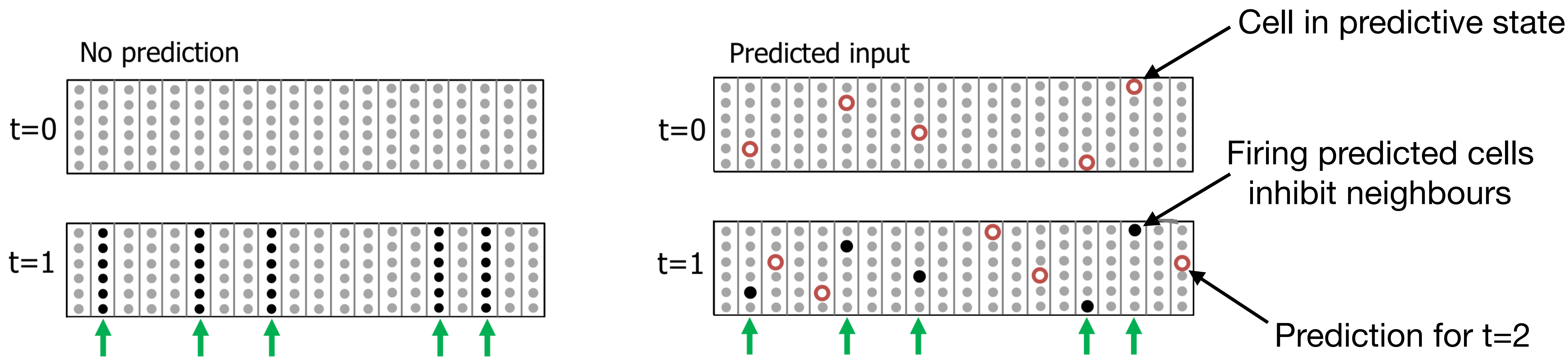
```
0100000000000000000000010000000000000000000000000000000000000000010000..........01000
0000000010000000000000010000000000000000000000000010000000000000010000..........00000
                       ↑                                         ↑
```

- Two representations with shared bits have some shared semantic information
- Comparing two representations is as simple as taking the intersection of the two indices sets.
- SDRs are inherently fault-tolerant and noise tolerant.
- Can check for existence of an SDR in a union

Sequence (formally Temporal) Memory



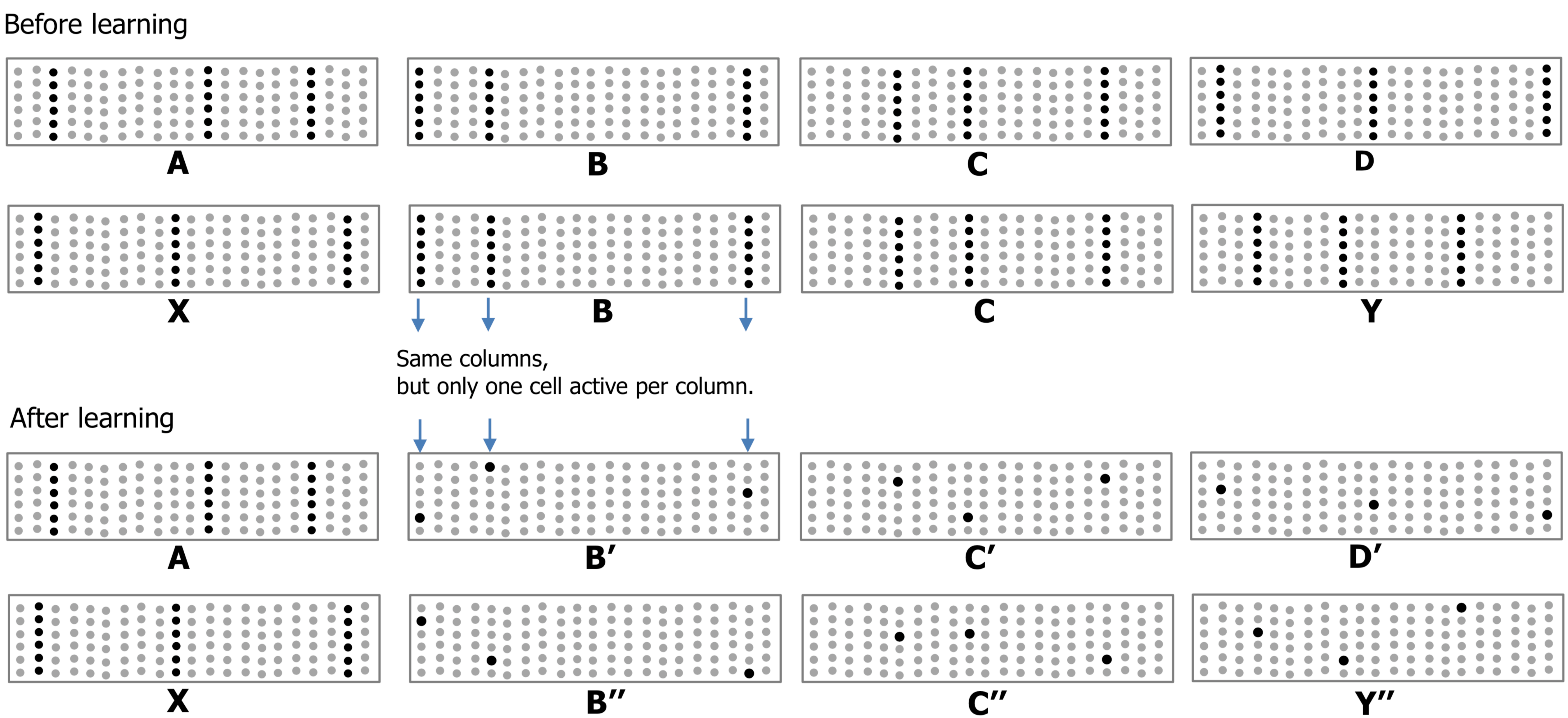
- Neurons in a mini-column learn same FF receptive field.
- Neurons forms distal connections to nearby cells.



- Learns sequences of SDRs and makes predictions of what the next input SDR will be
- Extremely robust (40% noise and fault tolerant)
- Learning is unsupervised and continuous
- Learns higher order sequences: “ABCD” vs “XBCY”

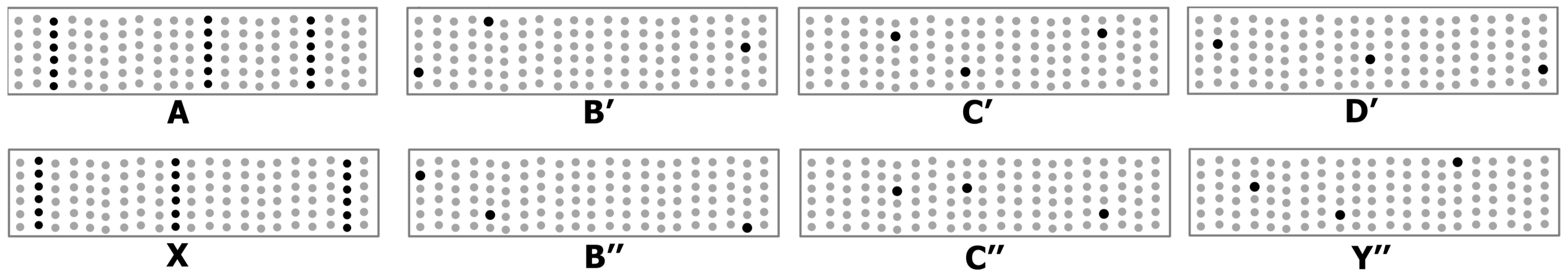
High Order Sequence Prediction

Sequences A-B-C-D vs. **X-B-C-Y**

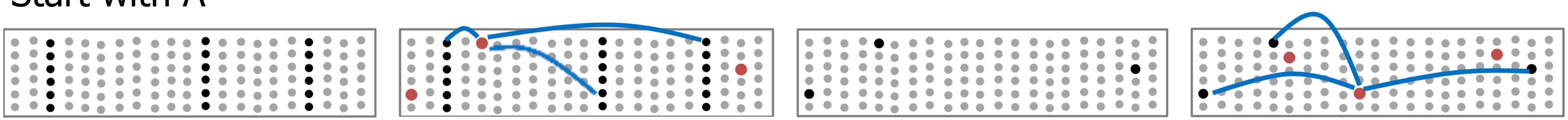


Sequence Prediction Step by Step

Trained Two Sequences A-B-C-D and **X-B-C-Y**

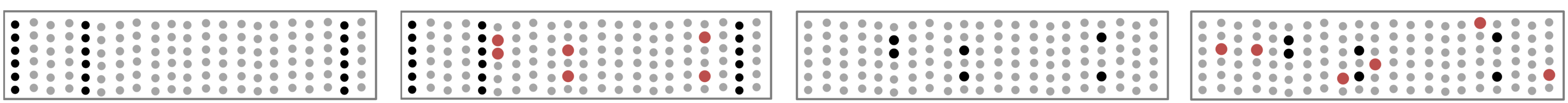


Start with A



A input | **B′ predicted** | **B input** | **C′ predicted**

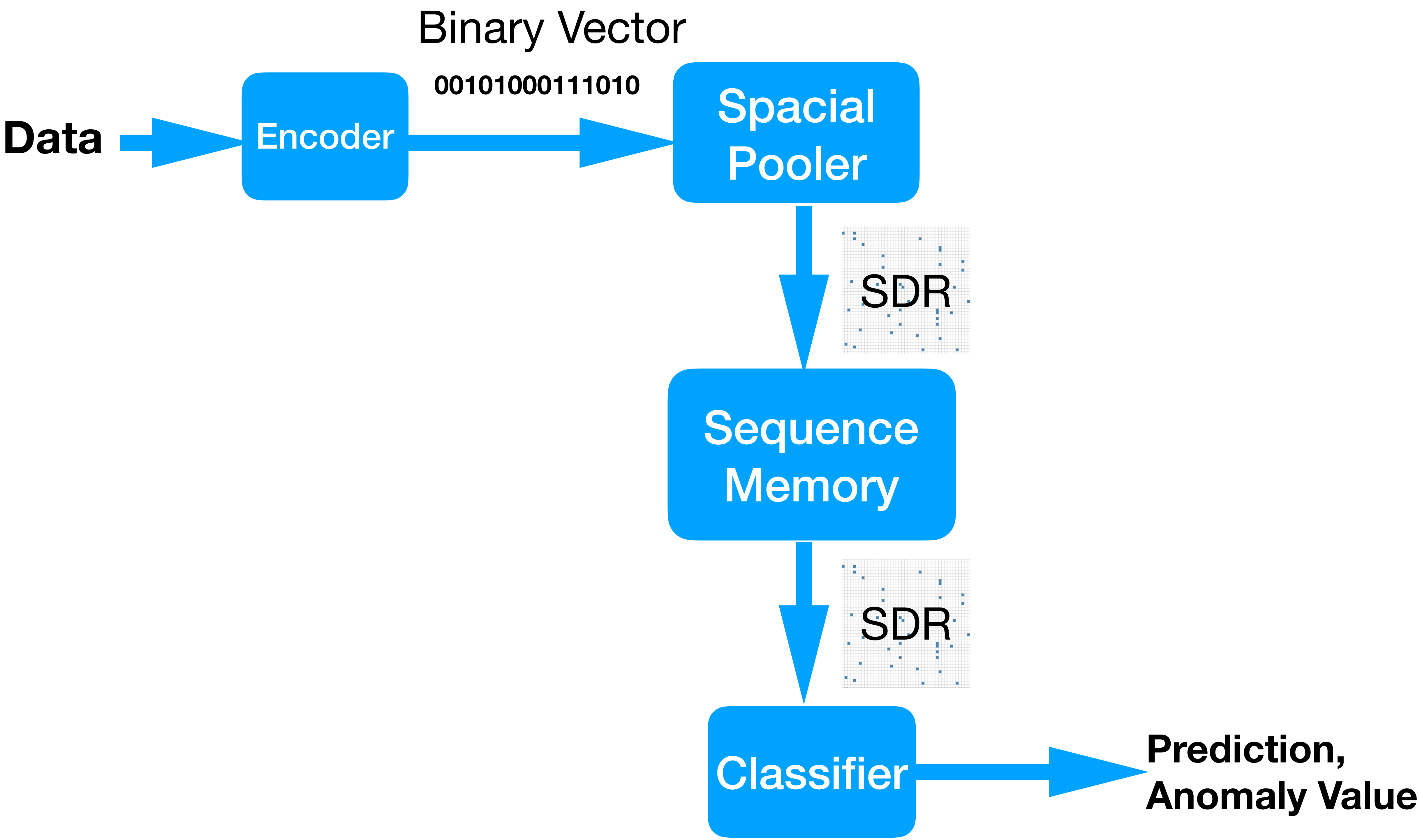
Start with B



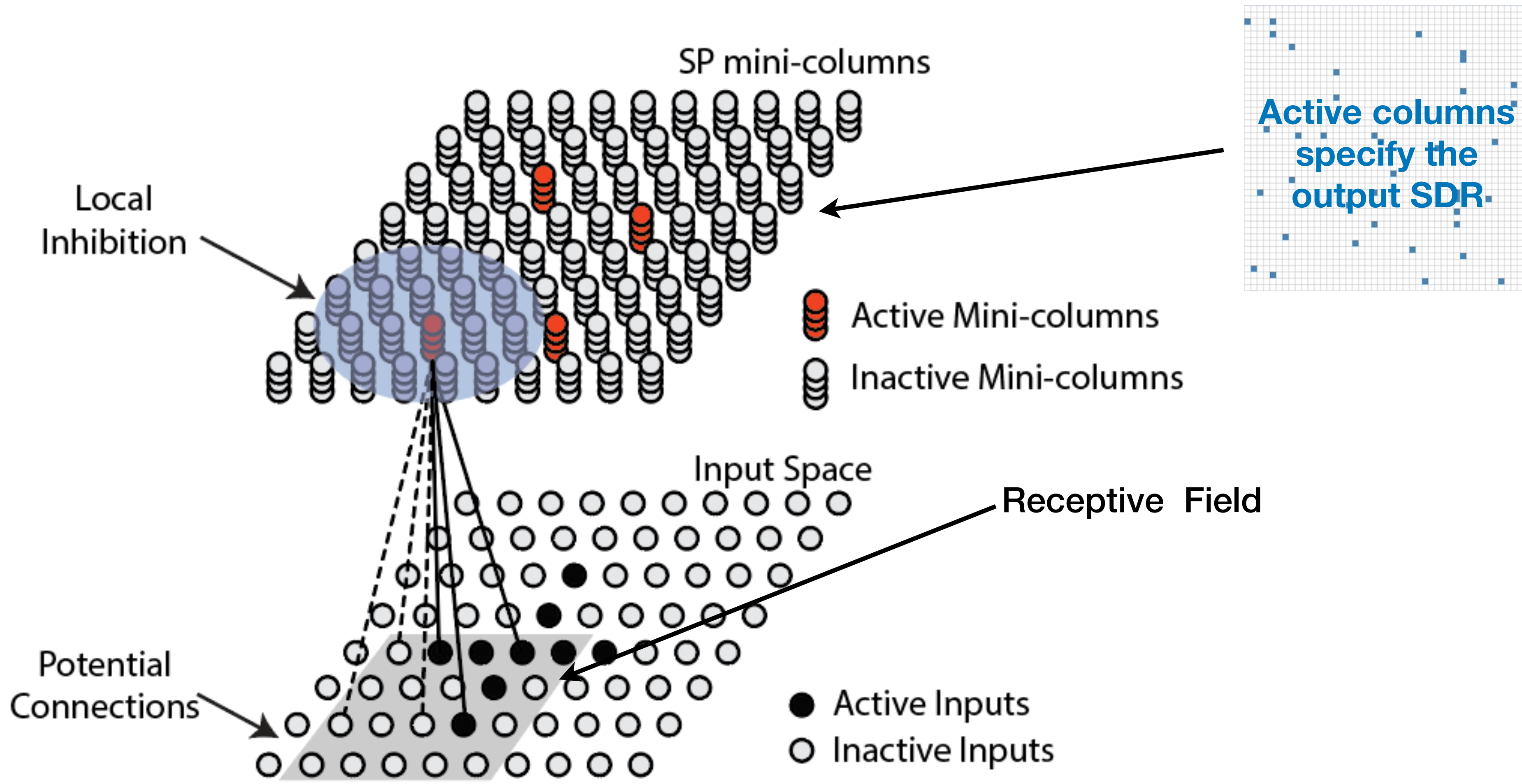
B input | **C′ and C″ predicted** | **C input** | **D′ and Y″ predicted**

- So starting with B and inputing C predicts both D and Y
- Hence Sequence Memory handles surprise and multiple simultaneous predictions

Anomaly Detection

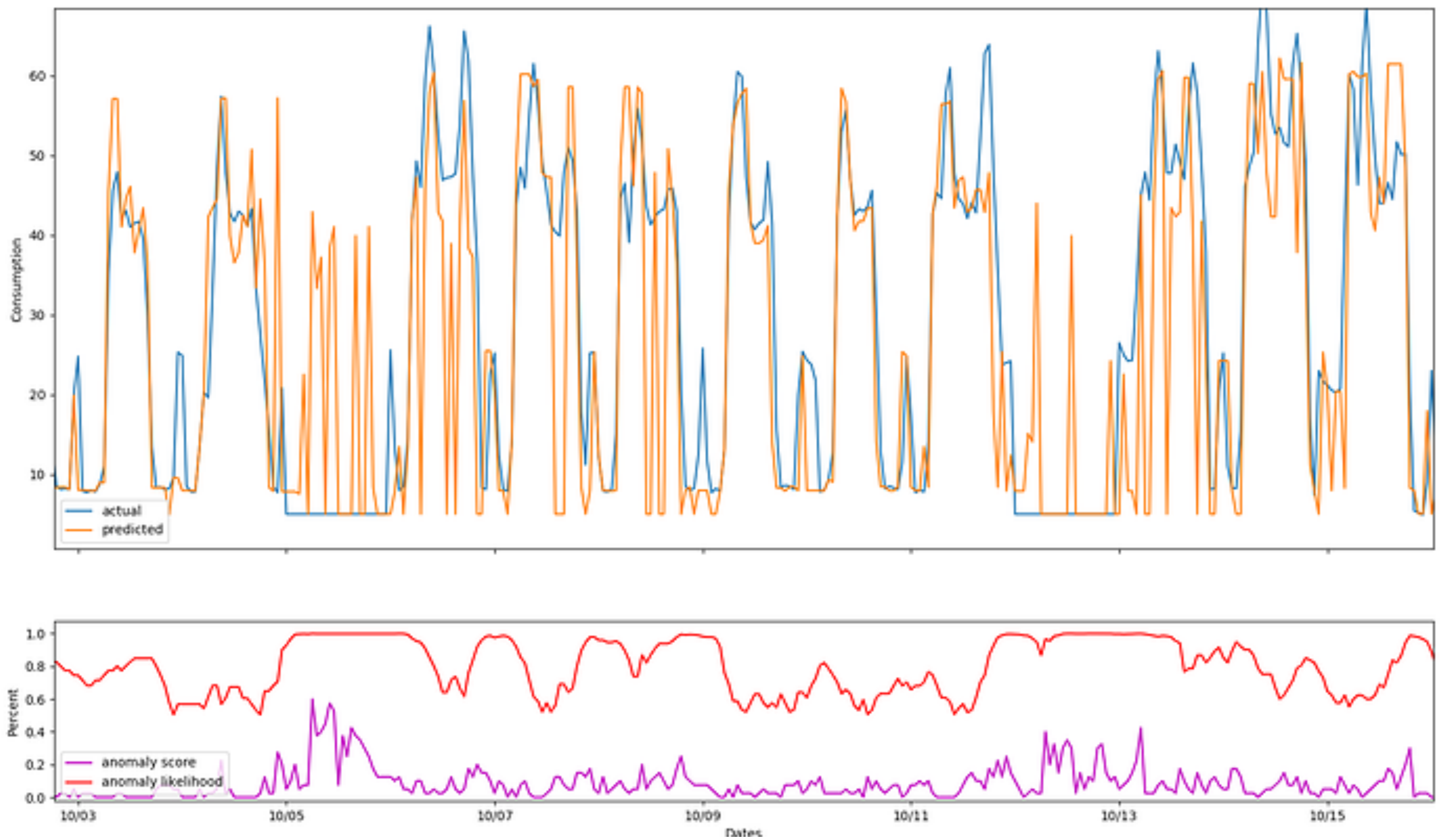


Spacial Pooler

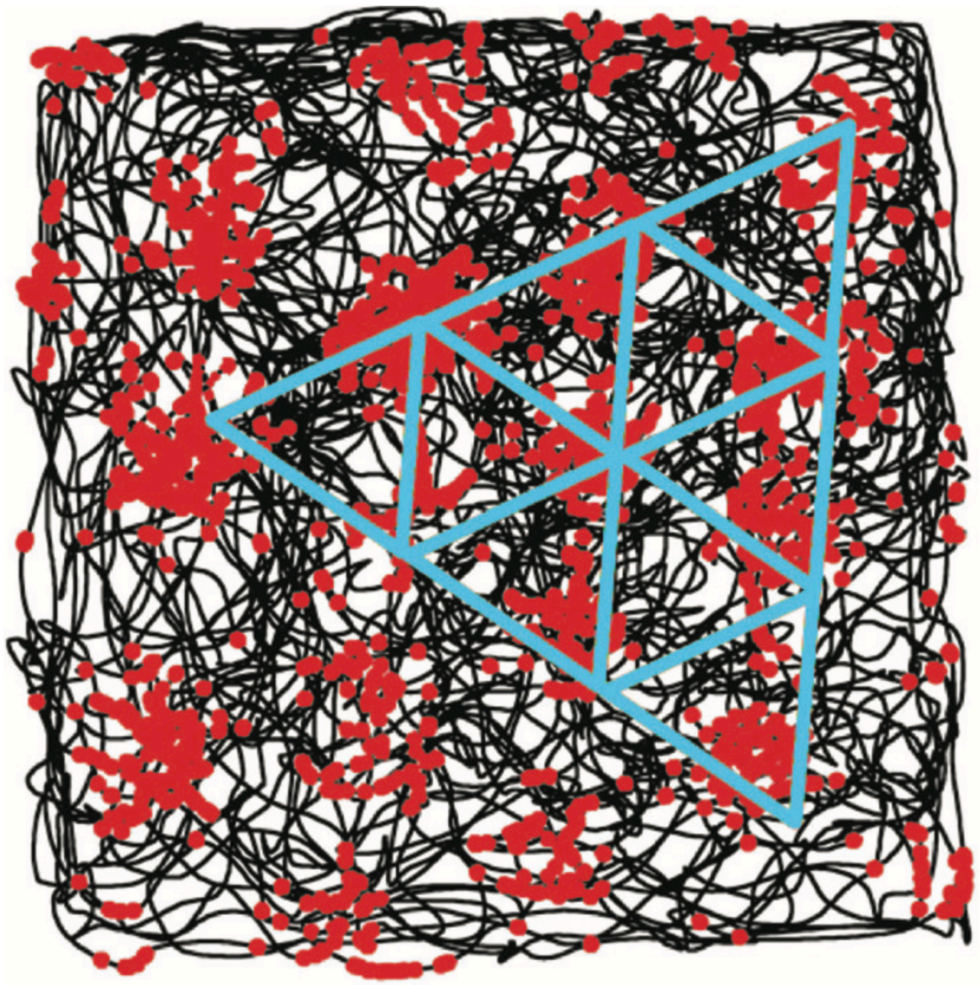


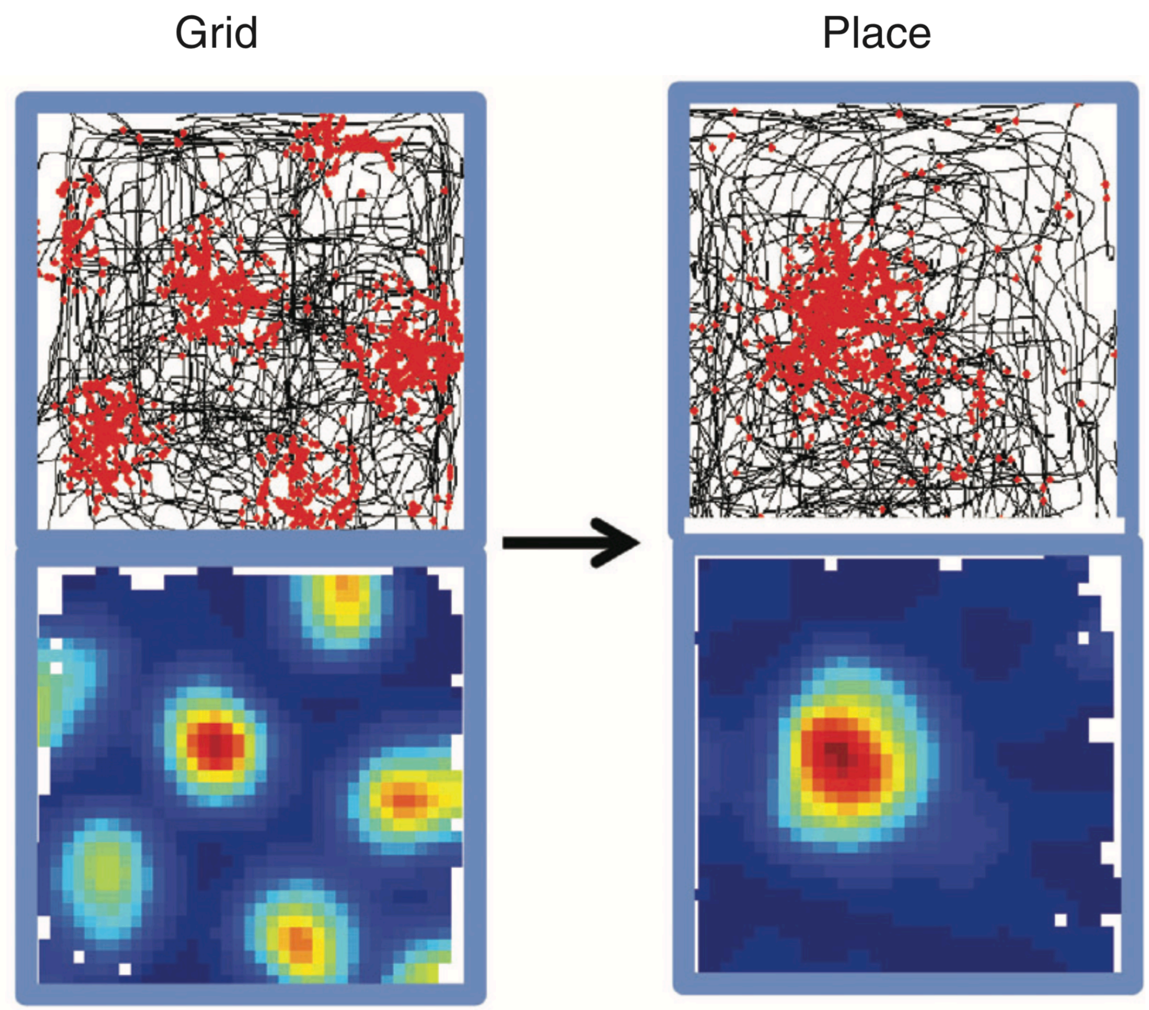
- Creates SDRs from N dimensional bit array inputs using Hebbian learning
- Learning is continuous
- Same input will generate the same unique SDR
- Preserves semantic similarity

Demo - Anomaly Detection

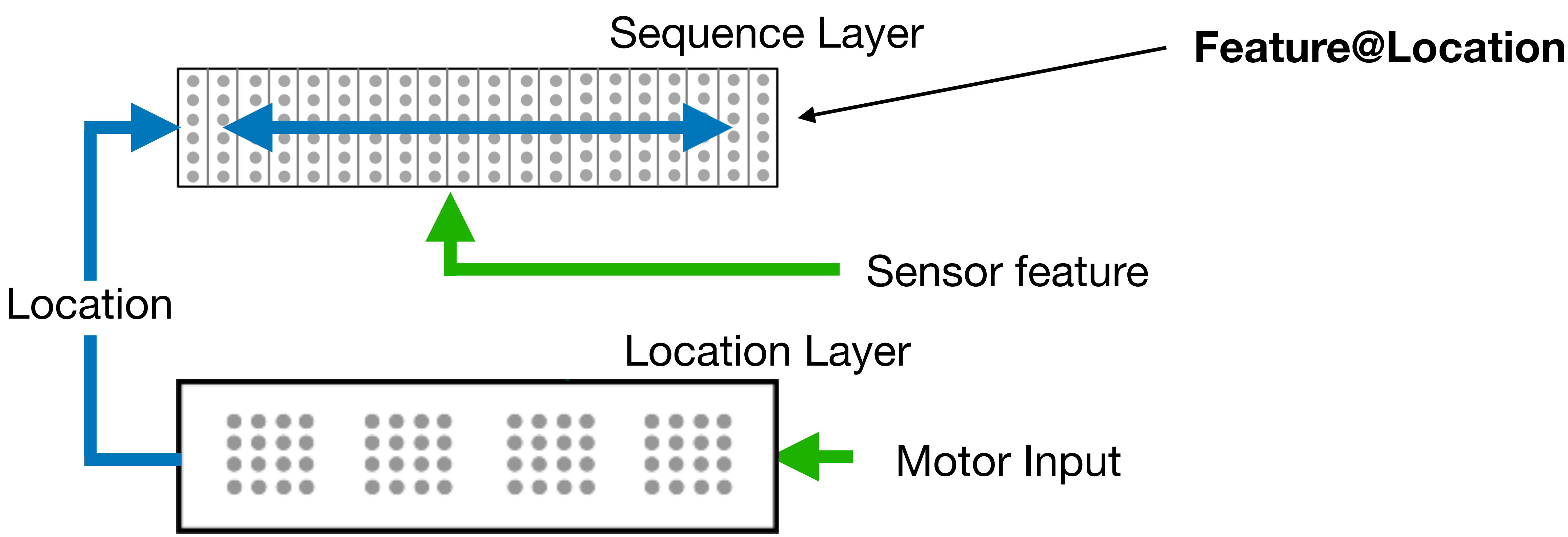


Grid and Place Cells



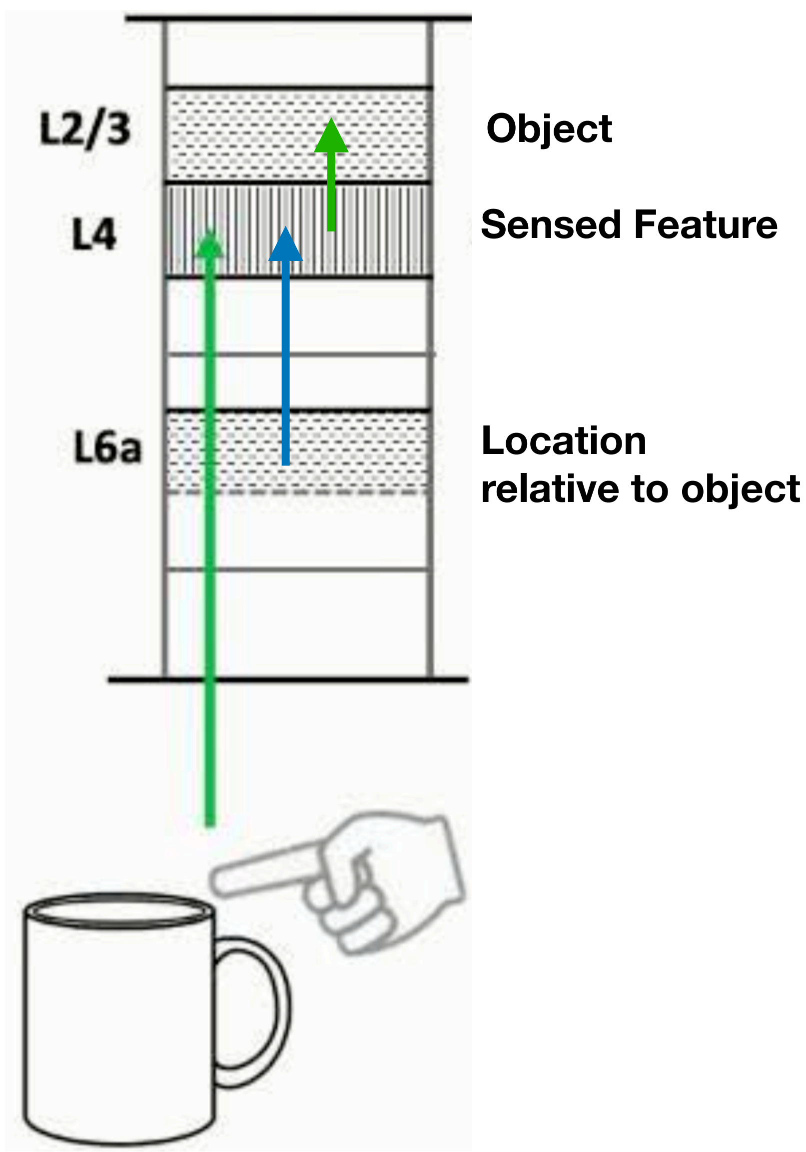


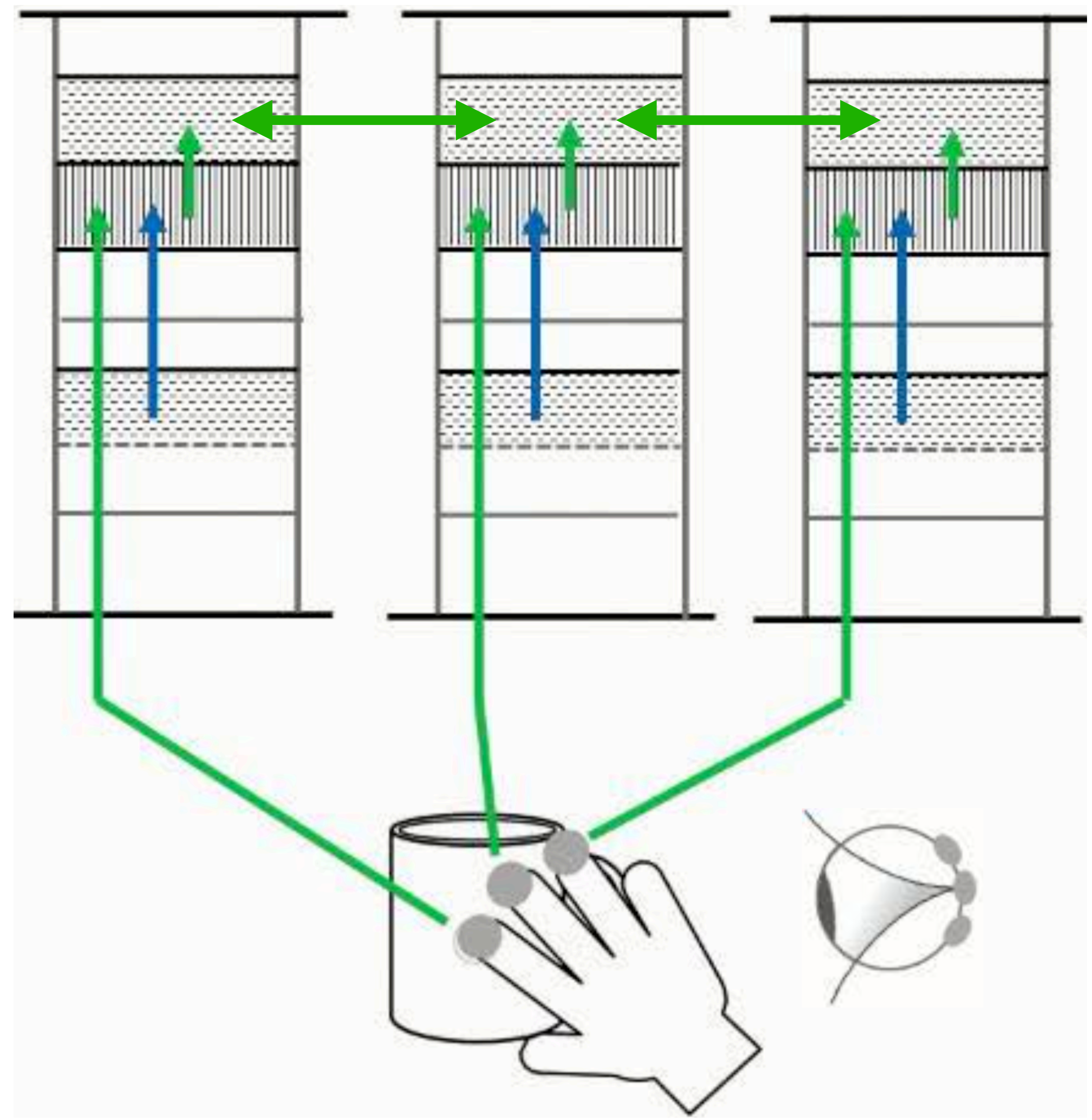
Model of Sensorimotor Sequence Memory



- With a motor-related context, a sequence memory can predict its input as the motor sensor moves.

Column Learns Objects





A single column learns complete models of objects by integrating feature and locations over time

Multiple columns can infer objects in a single sensation by “voting” on object identity

Commercial Applications

GROK

http://grokstream.com/

cortical.io

http://www.cortical.io/

INTELLETIC CORPORATION

https://intelletic.com/

Links

- Numenta: https://numenta.com/
- HTM Community: https://numenta.org/
- HTM School: https://www.youtube.com/channel/UC8-ttzWLgXZOGuhUyrPlUuA

Slides and Demos

- https://github.com/fcr/pyconis_2019_htm_slides

rotbart@softbart.com
https://www.linkedin.com/in/fcrotbart
https://github.com/fcr
@rotbartfc

Questions?