Sensory-Motor Integration in HTM Theory

Numenta

Catalyst for machine intelligence

Research

Theory, Algorithms

→

NuPIC

Open source community

→

GROK

Products for streaming analytics
Based on cortical algorithms

San Jose Meetup 3/14/14

Jeff Hawkins

jhawkins@Numenta.com

What is the Problem?

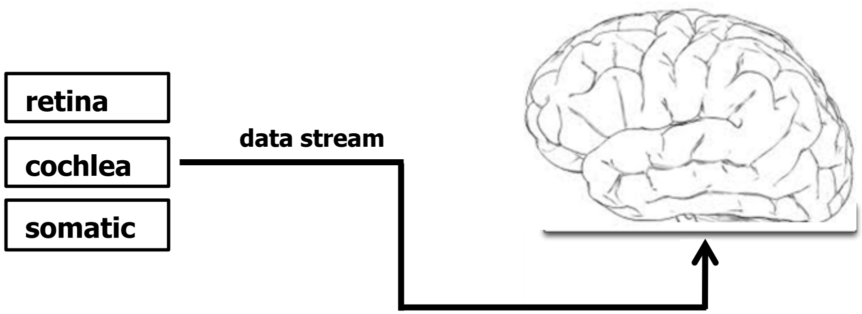
Sensory-motor integration
Sensorimotor contingencies
Embodied A.I.
Symbol grounding

- Most changes in sensor data are due to our own actions.
- Our model of the world is a “sensory-motor model”

1) How does the cortex learn this sensory-motor model?

2) How does the cortex generate behavior?

Hierarchical Temporal Memory (Review)



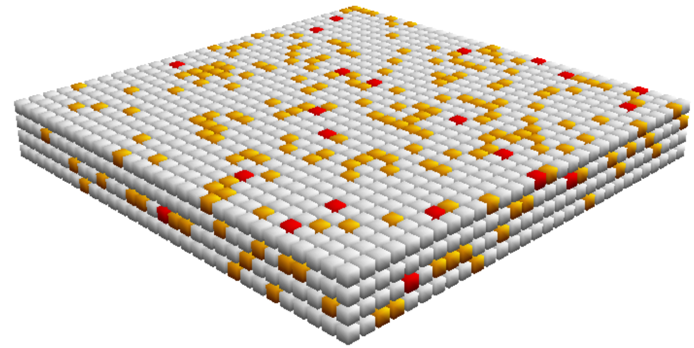
1) Hierarchy of nearly identical regions
- across species
- across modalities

2) Regions are mostly sequence memory
- inference
- motor

3) Feedforward: Temporal stability
- inference
- the world seems stable

Feedback: Unfold sequences
- motor/prediction

Cortical Learning Algorithm (Review)

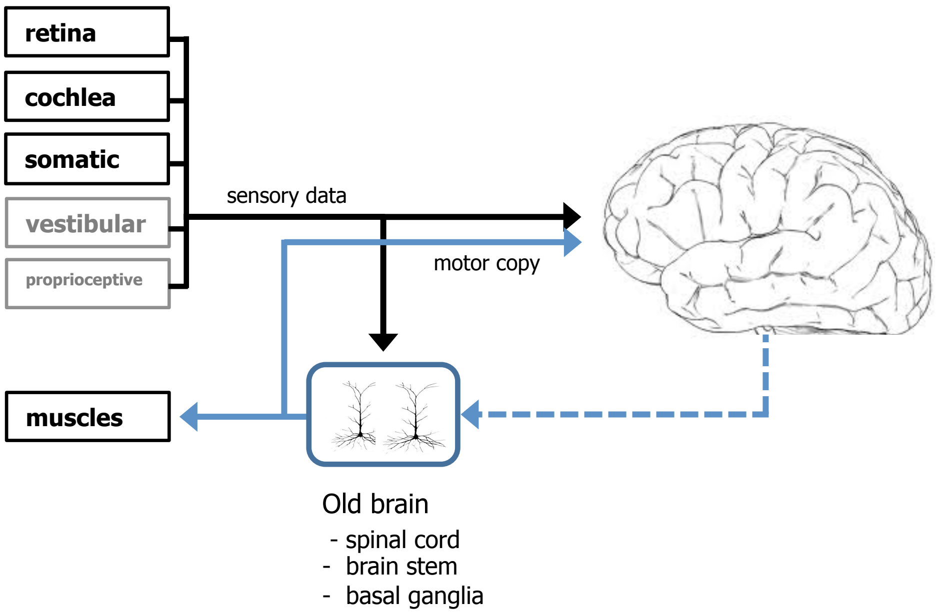


A Model of a Layer of Cells in Cortex
Sequence Memory

- Converts input to sparse representations
- Learns sequences
- Makes predictions and detects anomalies

Capabilities
- On-line learning
- Large capacity
- High-order
- Simple local learning rules
- Fault tolerant

Cortex and Behavior (The big picture)

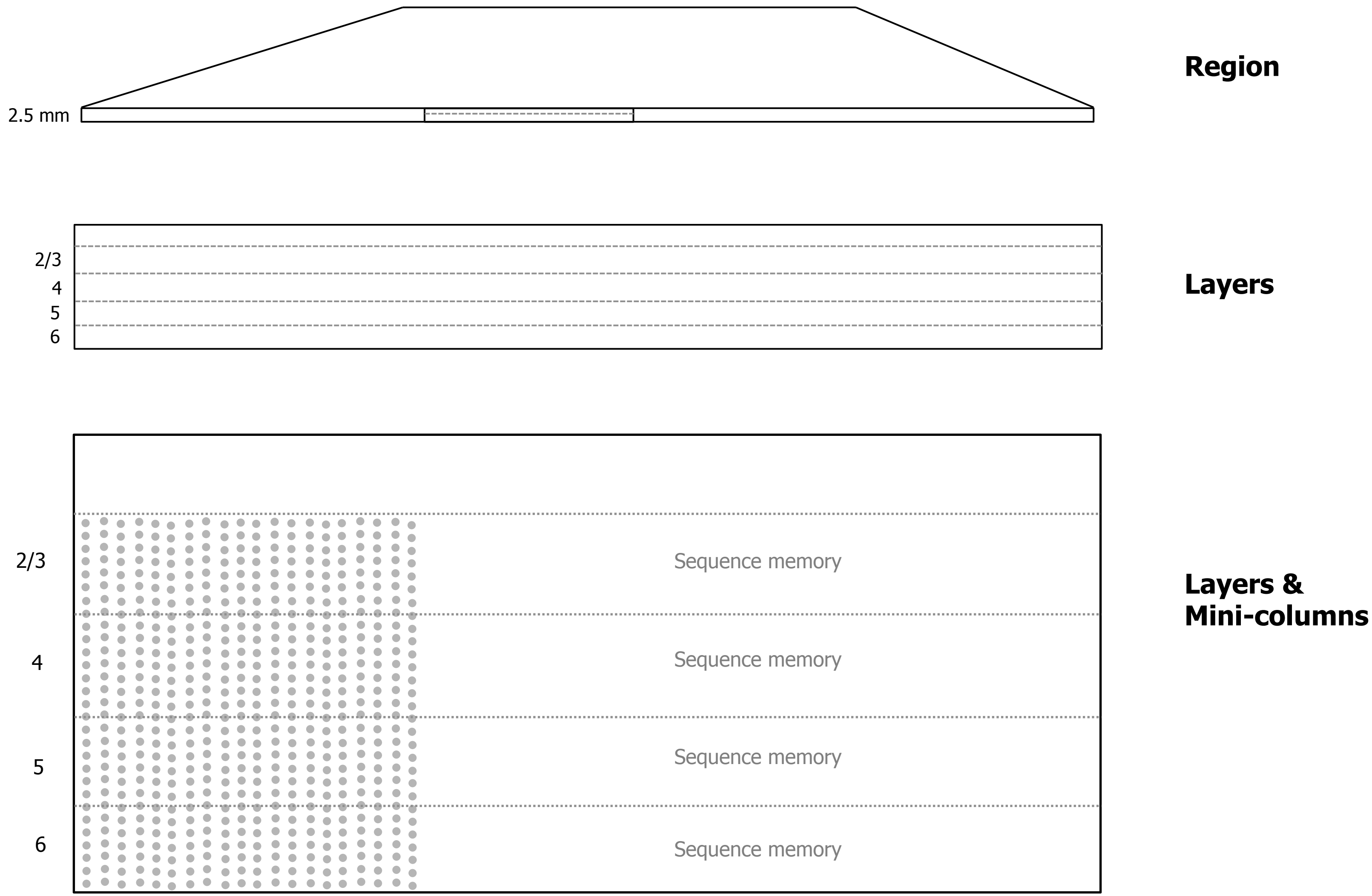


Old brain has complex behaviors.

Cortex evolved "on top" of old brain.

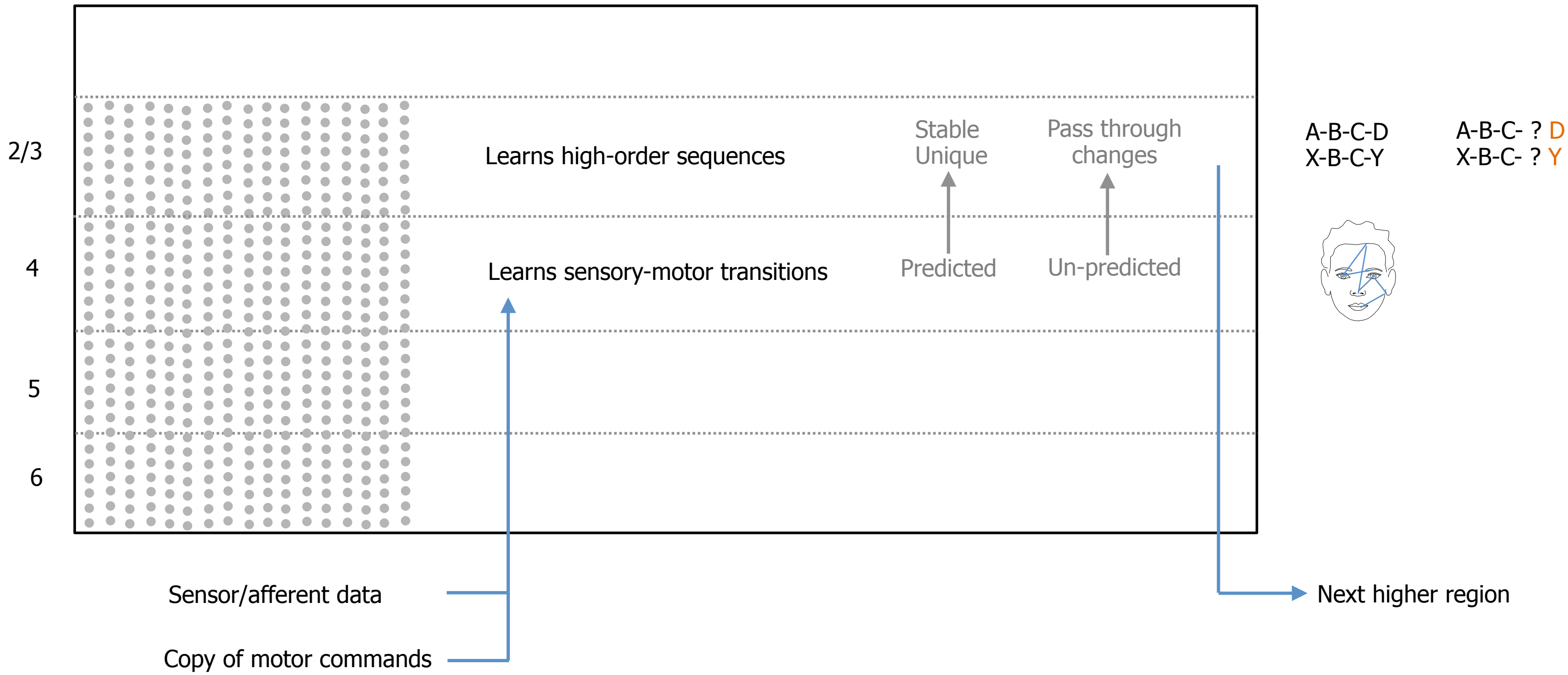
Cortex receives both sensory data and copies of motor commands.
- It has the information needed for sensory-motor inference.

Cortex learns to "control" old brain (via associative linking).
- It has the connections needed to generate goal-oriented behavior.



1) **The cellular layer is unit of cortical computation.**
2) **Each layer implements a variant of the CLA.**
3) **Mini-columns play critical role in how layers learn sequences.**

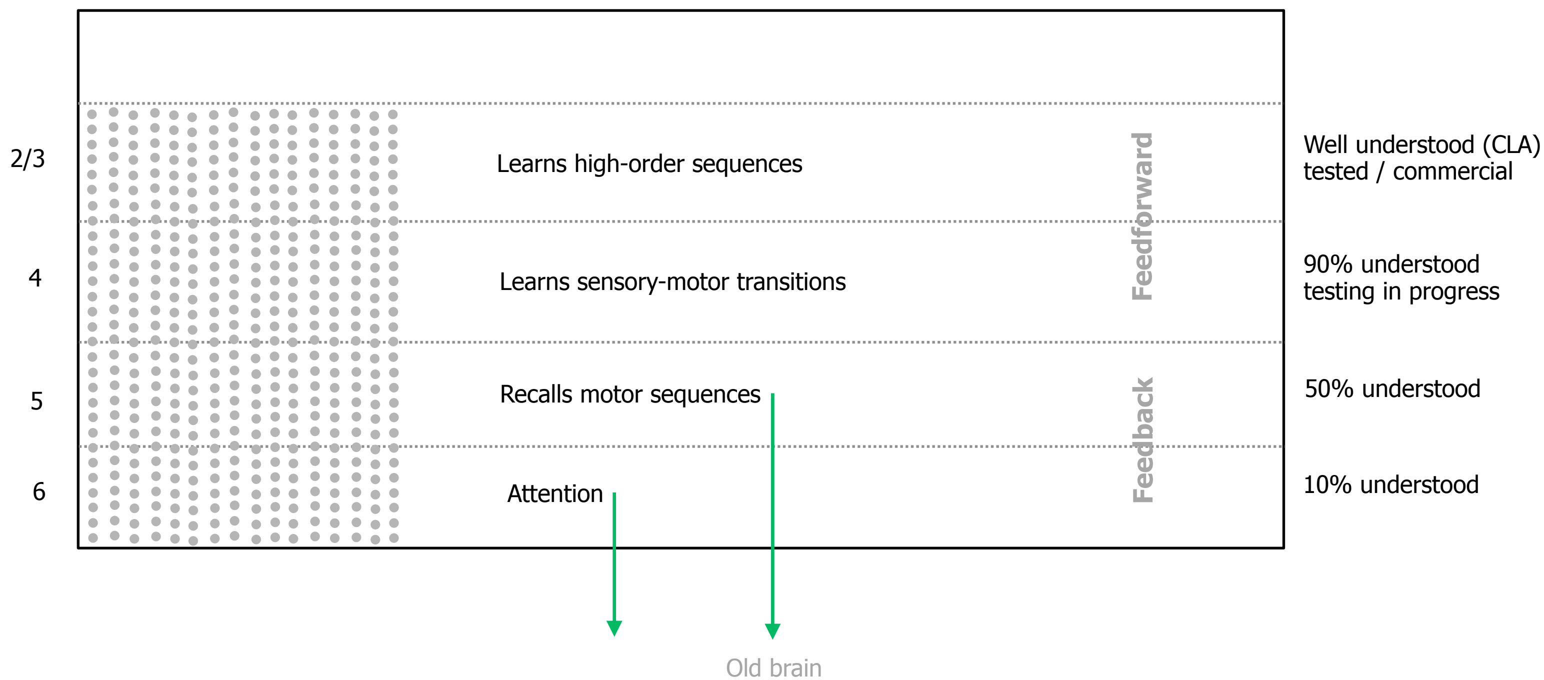
L4 and L2/3: Feedforward Inference



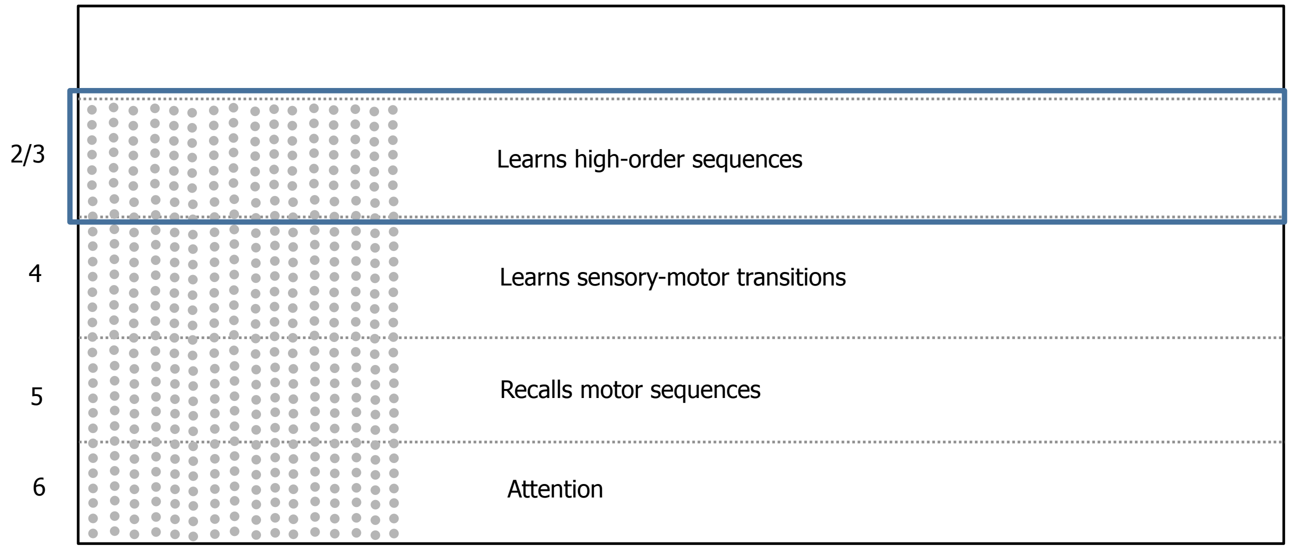
Layer 4 learns changes in input due to behavior.
Layer 3 learns sequences of L4 remainders.

These are universal inference steps.
They apply to all sensory modalities.

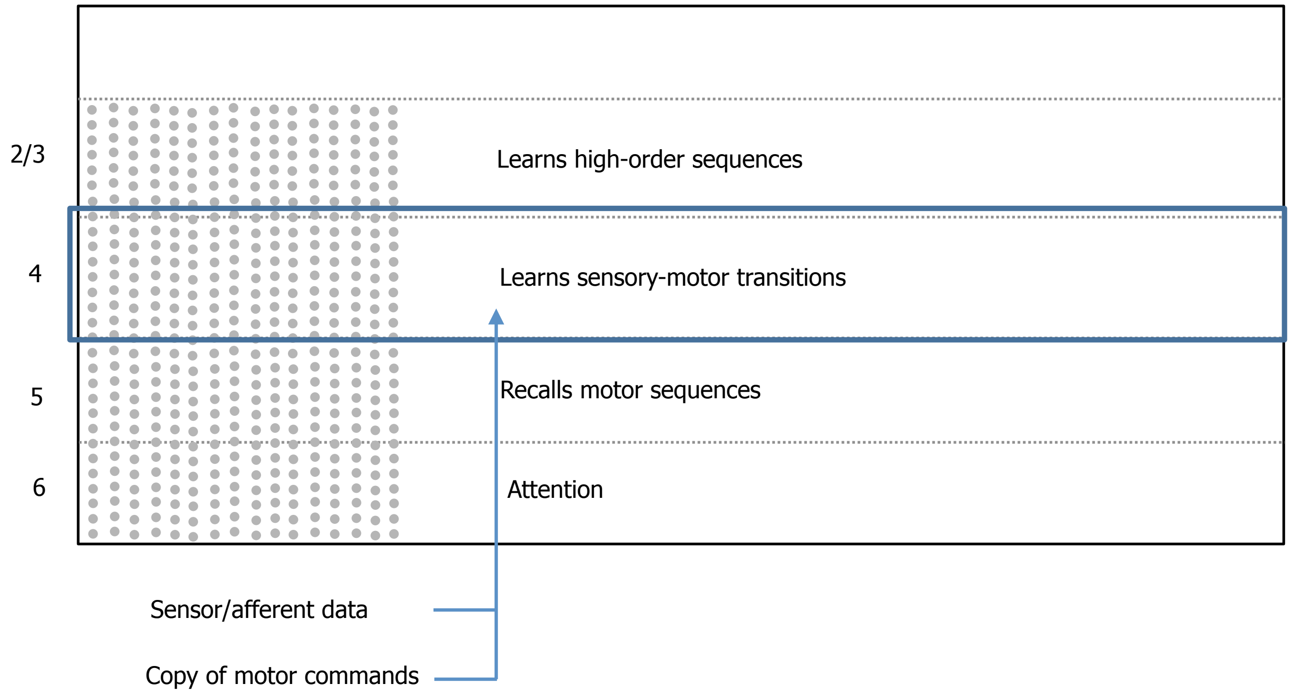
L5 and L6: Feedback, Behavior and Attention

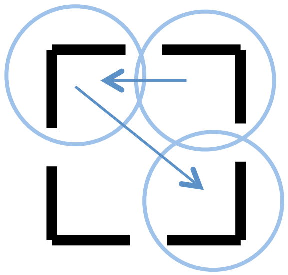


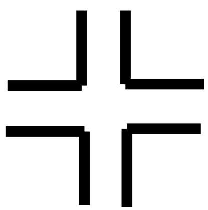
L3: Existing CLA Learns High-order Sequences

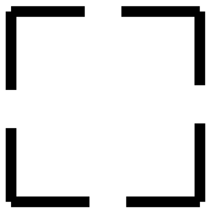


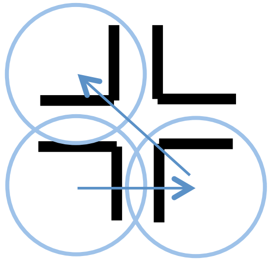
L4: CLA With Motor Command Input
Learns Sensory-motor Transitions

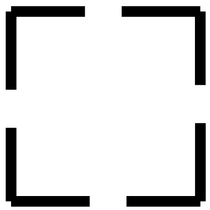


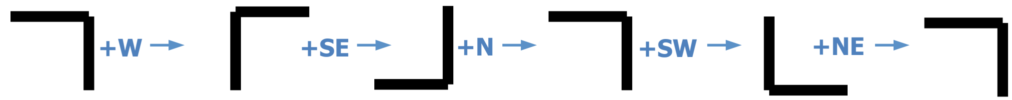




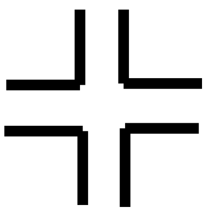


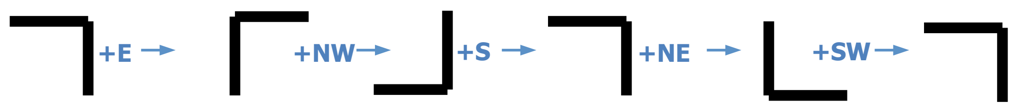






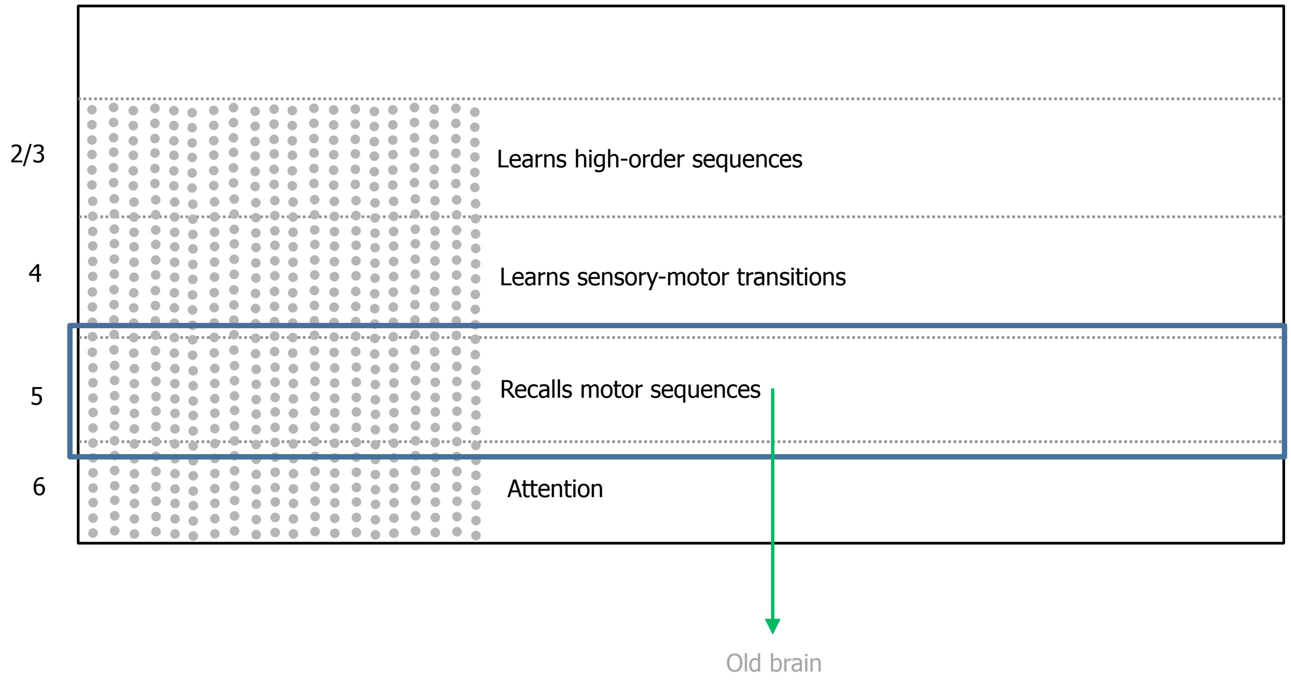
With motor context, CLA will form <u>unique</u> representations.



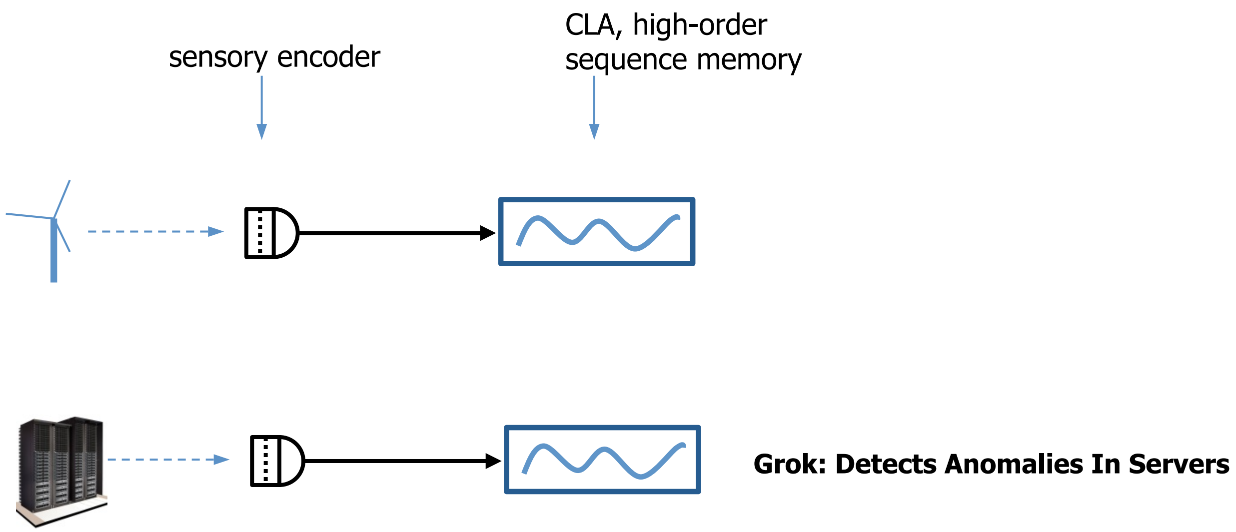


With motor context, CLA will form <u>unique</u> representations.

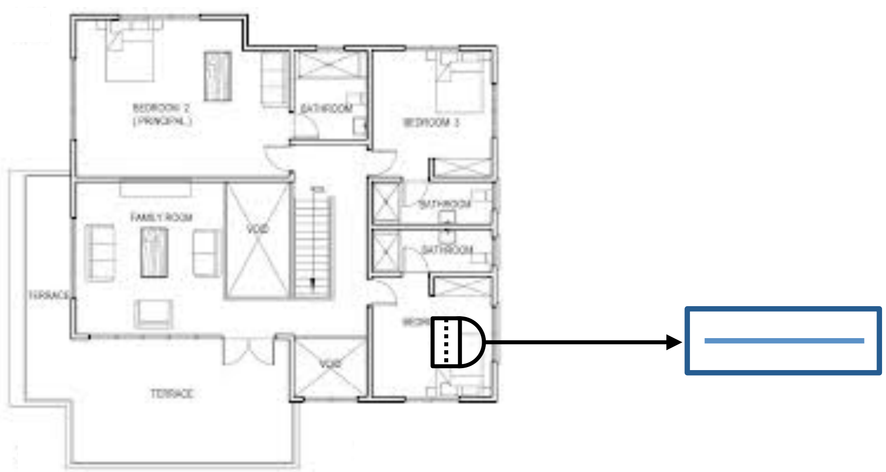
How Does L5 Generate Motor Commands?



One extreme: Dynamic world no behavior



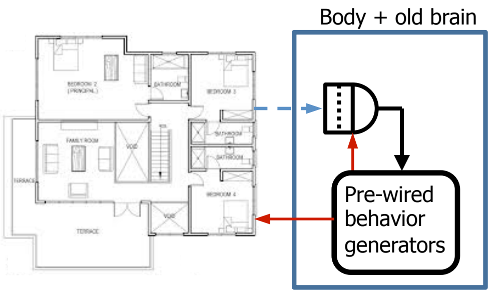
Other extreme: Static world



Behavior is needed for any change in sensory stream

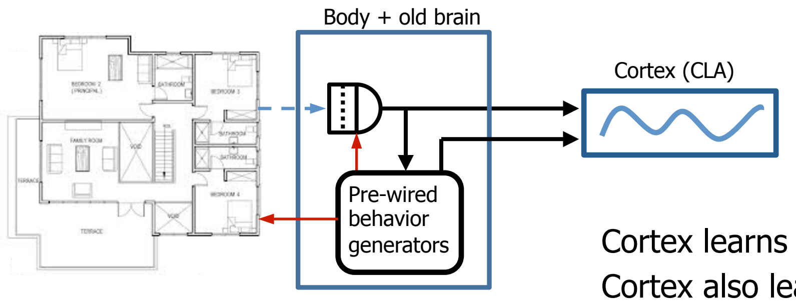
1) **Move sensor:** saccade, turn head, walk
2) **Interact with world:** push, lift, open, talk

Review: Mammals have pre-wired, sub-cortical behaviors



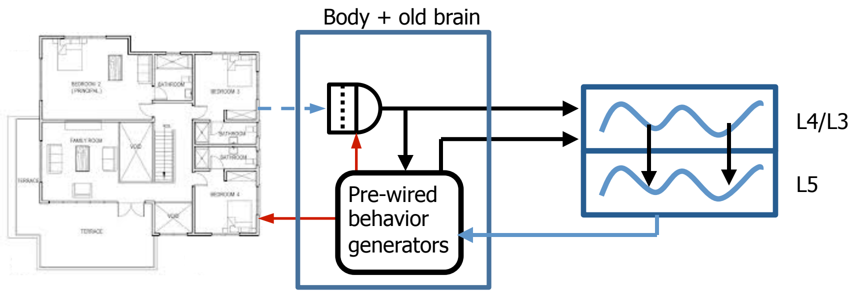
walking
running
eye movements
head turning
grasping
reaching
breathing
blinking, etc.

Review: Cortex models behavior of body and old brain



Cortex learns sensory patterns.
Cortex also learns sensory-motor patterns.
Makes much better predictions knowing behavior.

How does cortex control behavior?



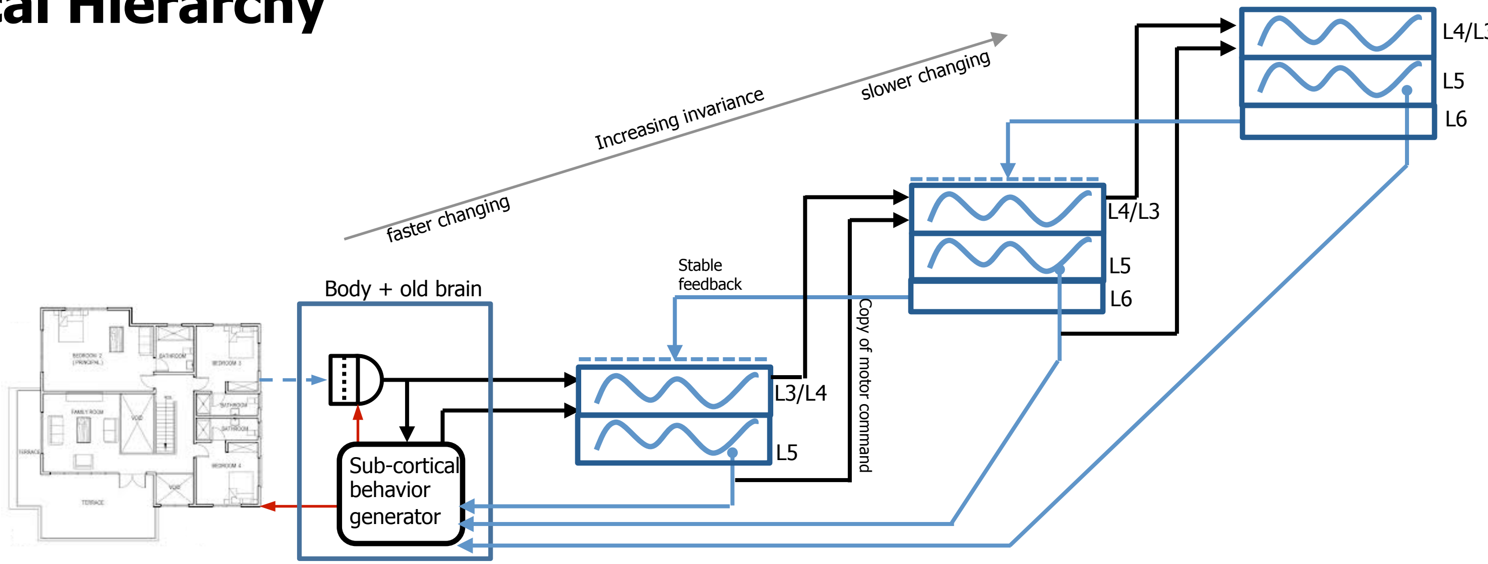
L4/L3 is a sensory-motor model of the world
L3 projects to L5 and they share column activations
Think of L5 as a copy of L3

L5 associatively links to sub-cortical behavior
Cortex can now invoke behaviors

Q. Why L5 in addition to L3?
A. To separate inference from behavior.

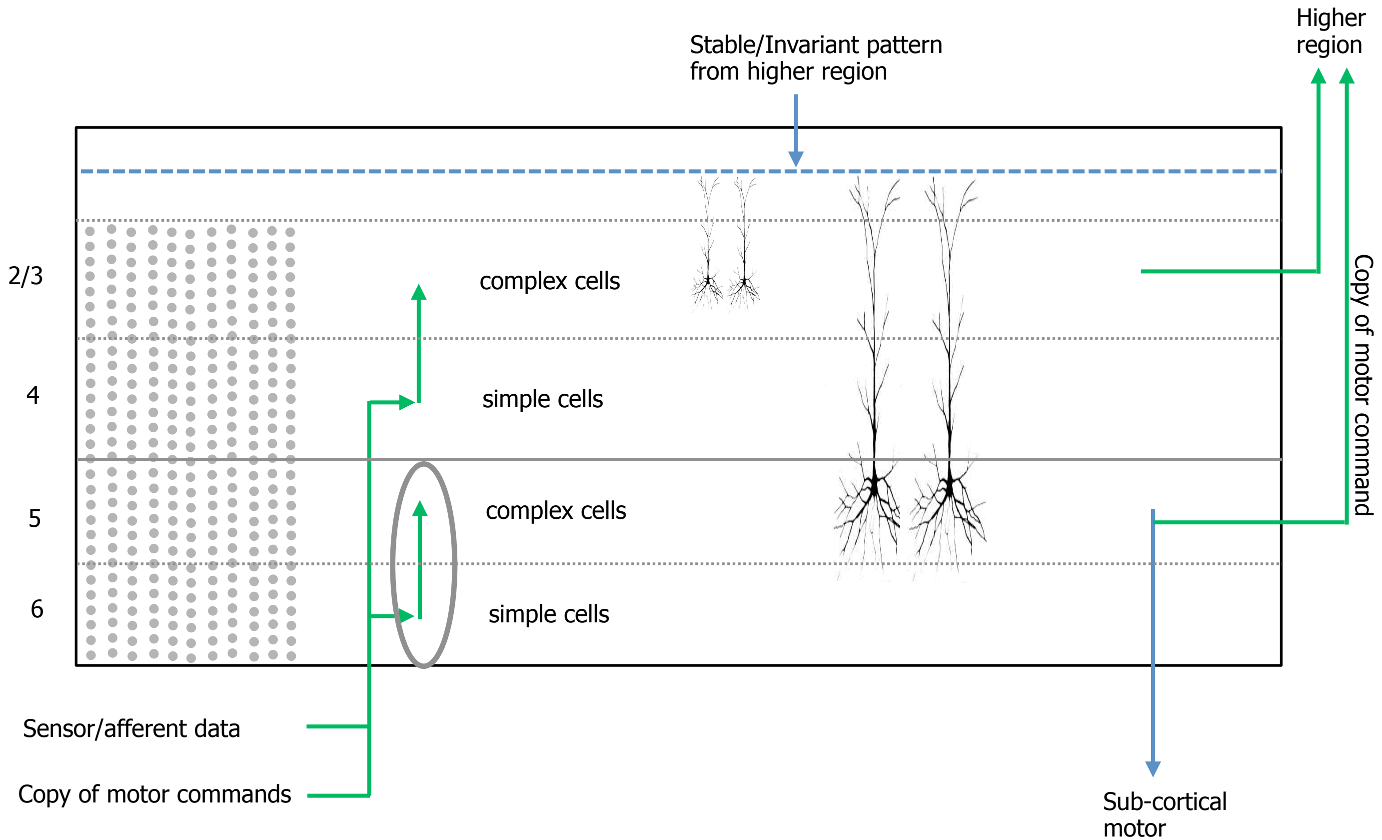
- Inference: L4/L3 (L3->L5->old brain learns associative link)
- Behavior: L5 (L5 generates behavior <u>before</u> feedforward inference)

Cortical Hierarchy



Each cortical region treats the region below it in the hierarchy as the first region treats the pre-wired areas.

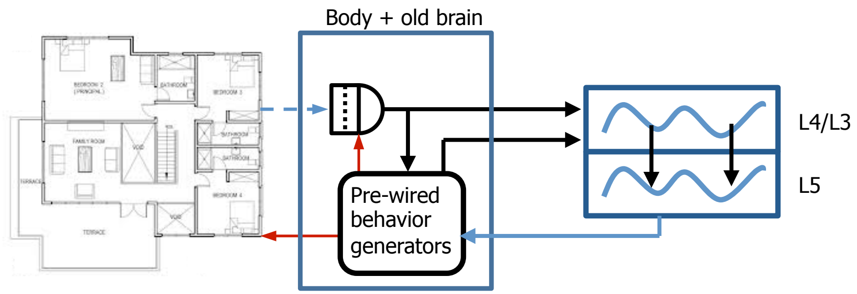
Goal Oriented Behavior via Feedback



Stable feedback invokes union of sparse states in multiple sequences. ("Goal")

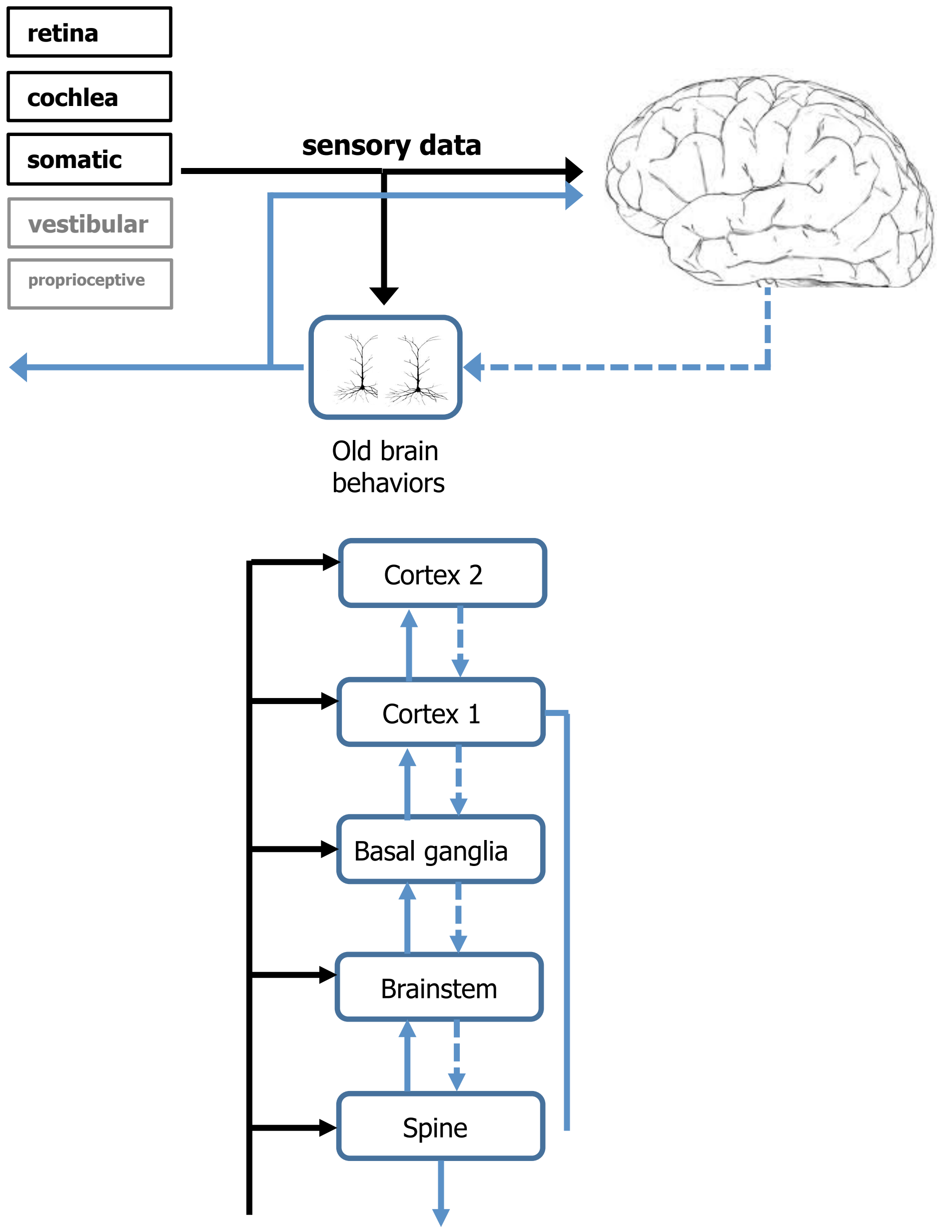
Feedforward input selects one state and plays back motor sequence from there.

Short term goal: Build a one region sensorimotor system

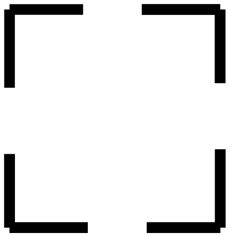


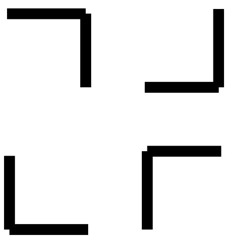
Hypothesis:

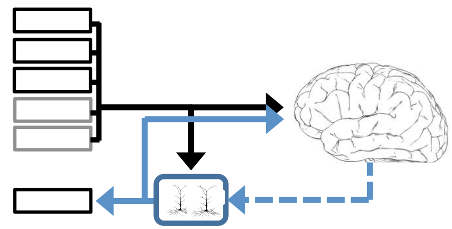
The only way to build machines with intelligent goal oriented behavior is to use the same principles as the neocortex.

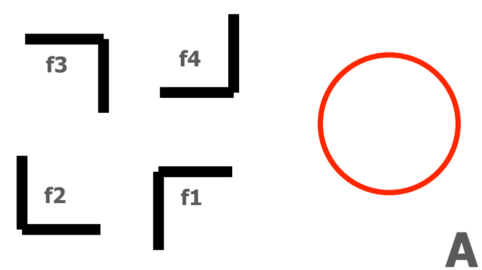


retina
cochlea
somatic
vestibular
proprioceptive
sensory data
Old brain behaviors
Cortex 2
Cortex 1
Basal ganglia
Brainstem
Spine



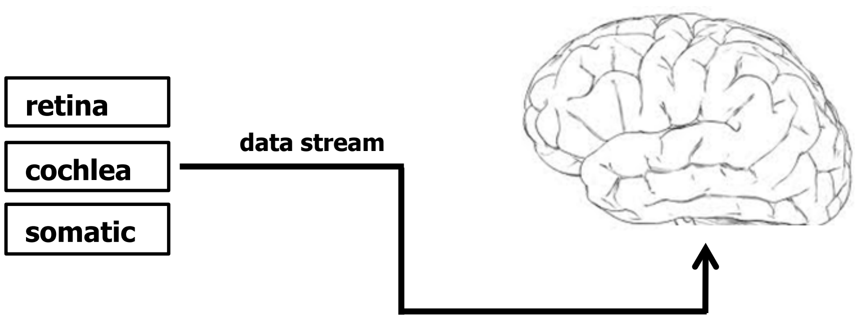




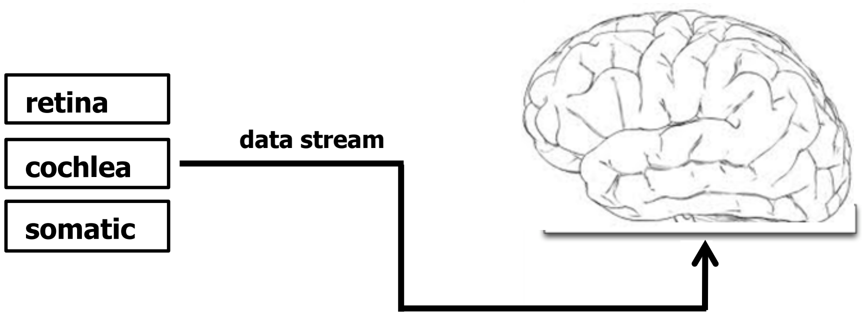

f3
f4
f2
f1
A

Cortical Principles

1) On-line learning from streaming data



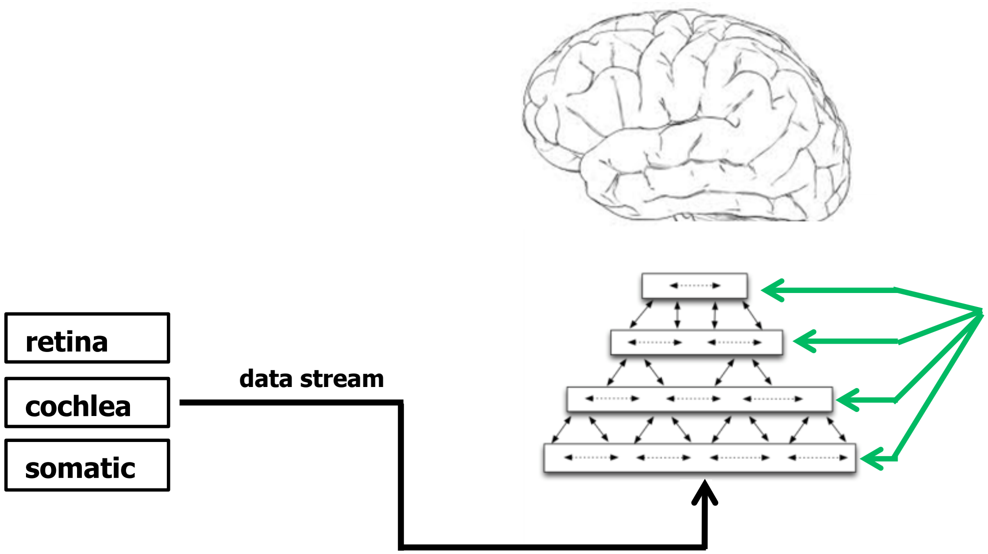
Cortical Principles



1) On-line learning from streaming data

2) Hierarchy of memory regions

Cortical Principles

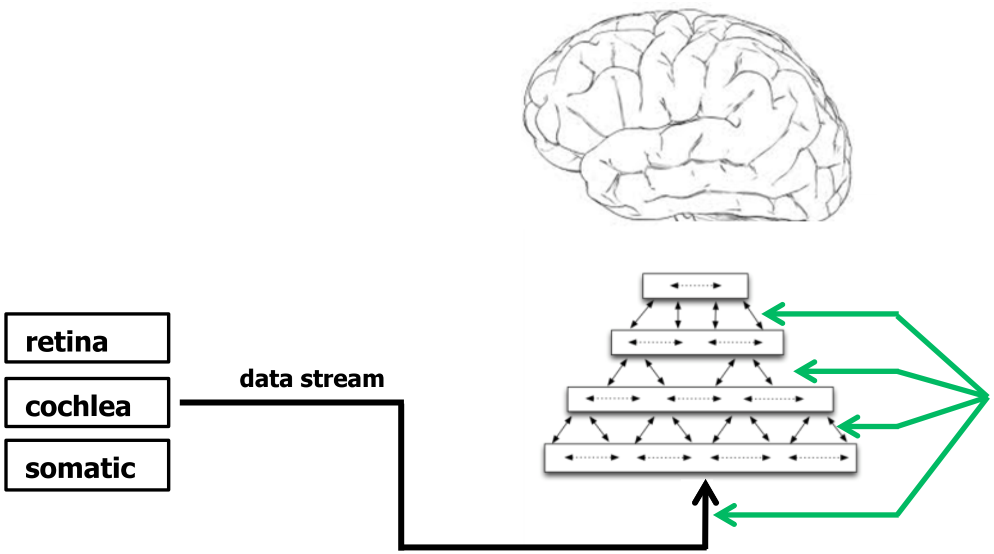


1) On-line learning from streaming data

2) Hierarchy of memory regions

3) Sequence memory
- inference
- motor

Cortical Principles



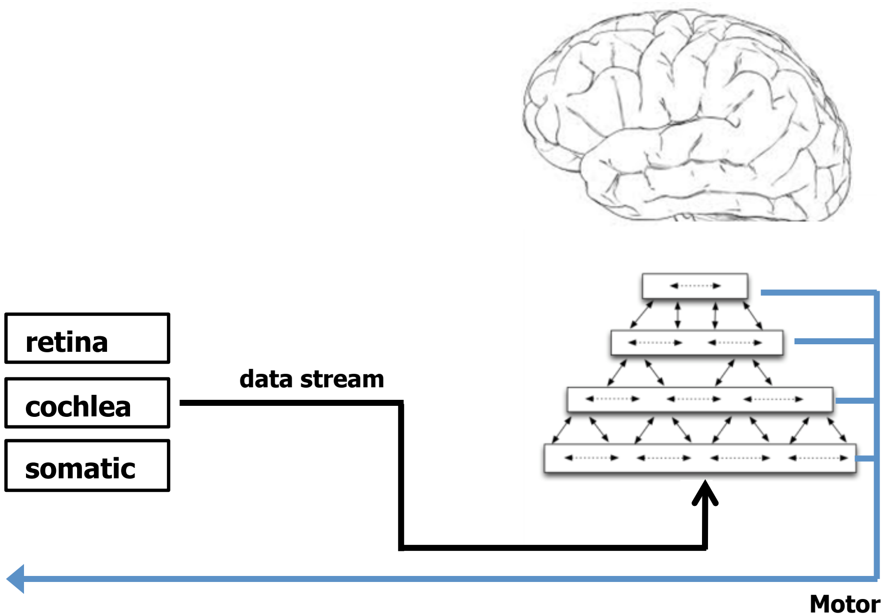
1) On-line learning from streaming data

2) Hierarchy of memory regions

3) Sequence memory

4) Sparse Distributed Representations

Cortical Principles



1) On-line learning from streaming data

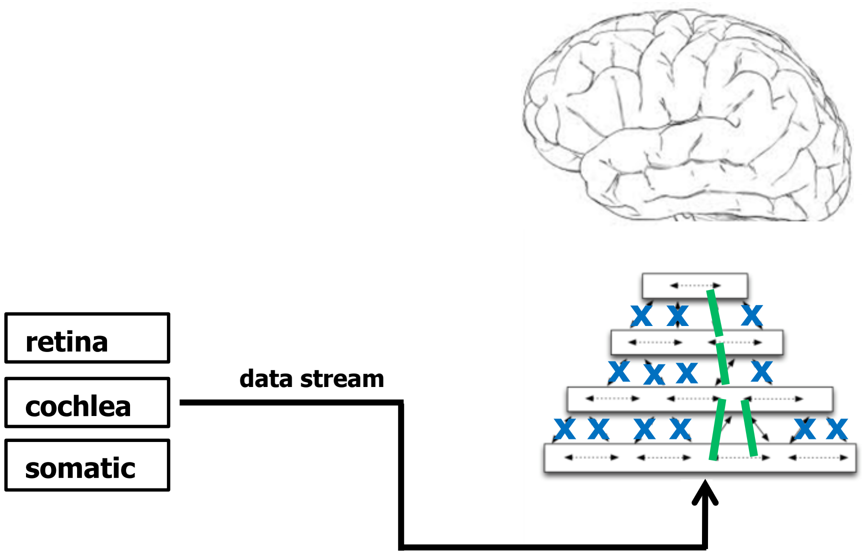
2) Hierarchy of memory regions

3) Sequence memory

4) Sparse Distributed Representations

5) All regions are sensory and motor

Cortical Principles



1) On-line learning from streaming data

2) Hierarchy of memory regions

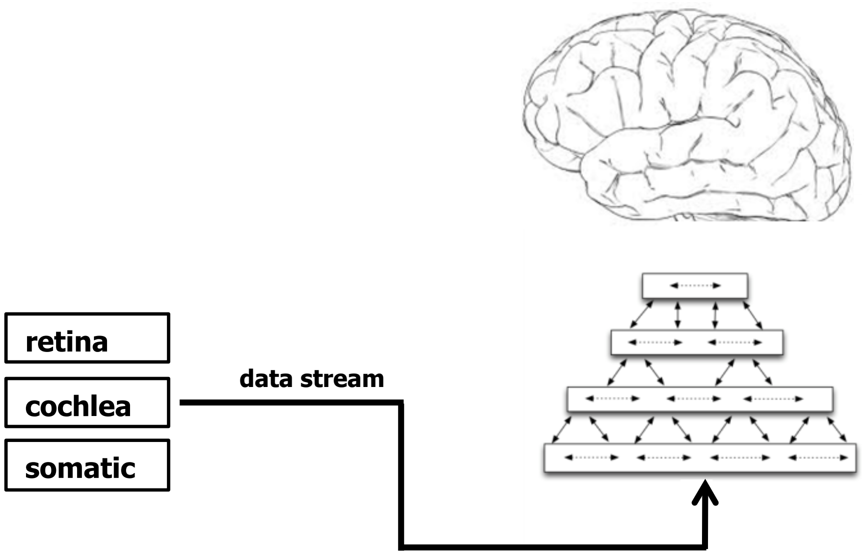
3) Sequence memory

4) Sparse Distributed Representations

5) All regions are sensory and motor

6) Attention

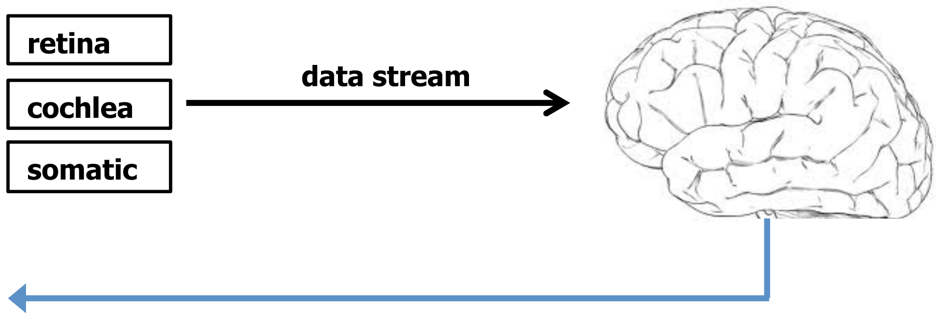
Cortical Principles



1) On-line learning from streaming data
2) Hierarchy of memory regions
3) Sequence memory
4) Sparse Distributed Representations
5) All regions are sensory and motor
6) Attention

These six principles are necessary and sufficient for biological and machine intelligence.

What the Cortex Does



The neocortex learns a model from sensory data

- predictions
- anomalies
- actions

The neocortex learns a sensory-motor model of the world

Attributes of Solution

Independent of sensory modality and motor modality

Should work with biological and non-biological senses

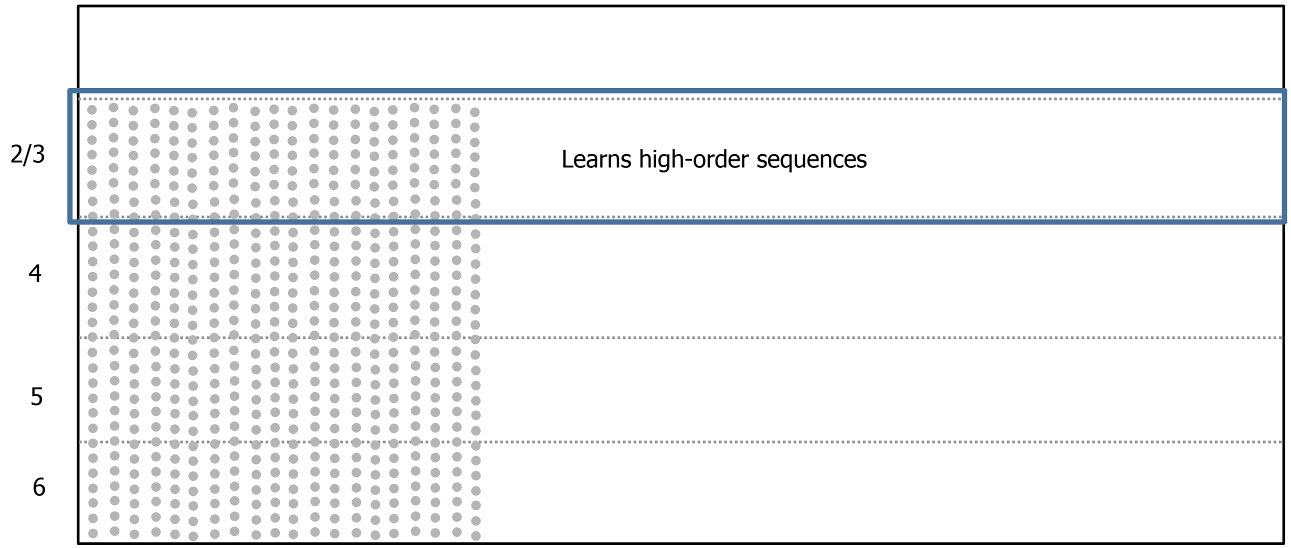
- Biological: Vision, audition, touch, proprioception
- Non-biological: M2M data, IT data, financial data, etc.

Can be understood in a single region

Must work in a hierarchy

Will be based on CLA principles

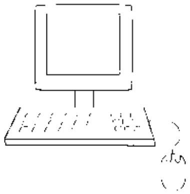
How does a layer of neurons learn sequences?



First:

- Sparse Distributed Representations

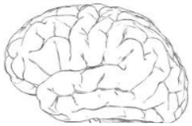
- Neurons



Dense Representations

- Few bits (8 to 128)
- All combinations of 1’s and 0’s
- Example: 8 bit ASCII

  01101101 = m
- Bits have no inherent meaning
  Arbitrarily assigned by programmer



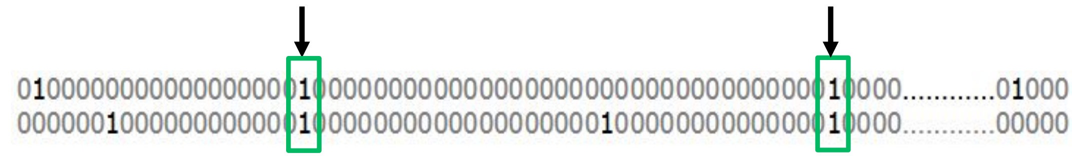
Sparse Distributed Representations (SDRs)

- Many bits (thousands)
- Few 1’s mostly 0’s
- Example: 2,000 bits, 2% active

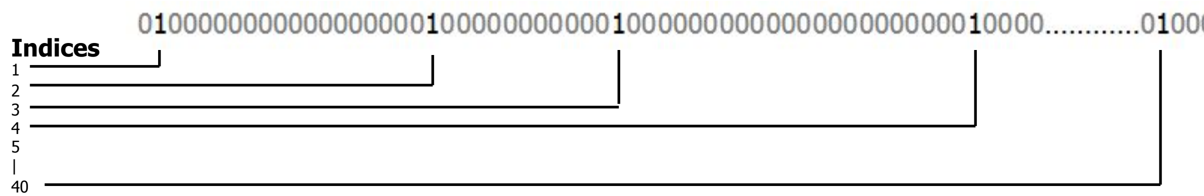
  0100000000000000000100000000000000000000000000000000000010000...........01000
- Each bit has semantic meaning
  Learned

SDR Properties

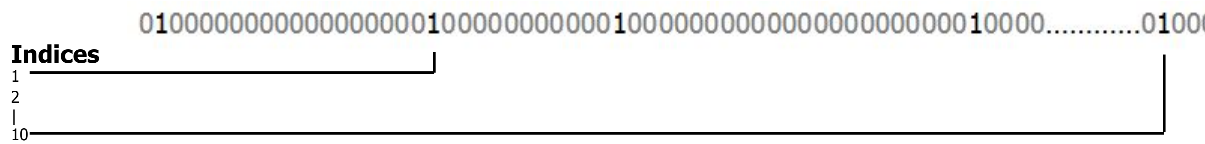
1) **Similarity**:
shared bits = semantic similarity



2) **Store and Compare**:
store indices of active bits



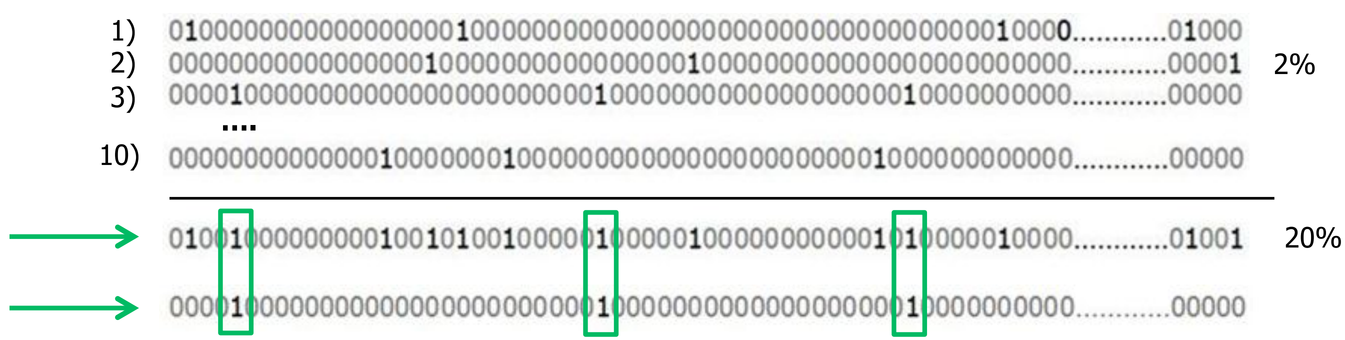
subsampling is OK



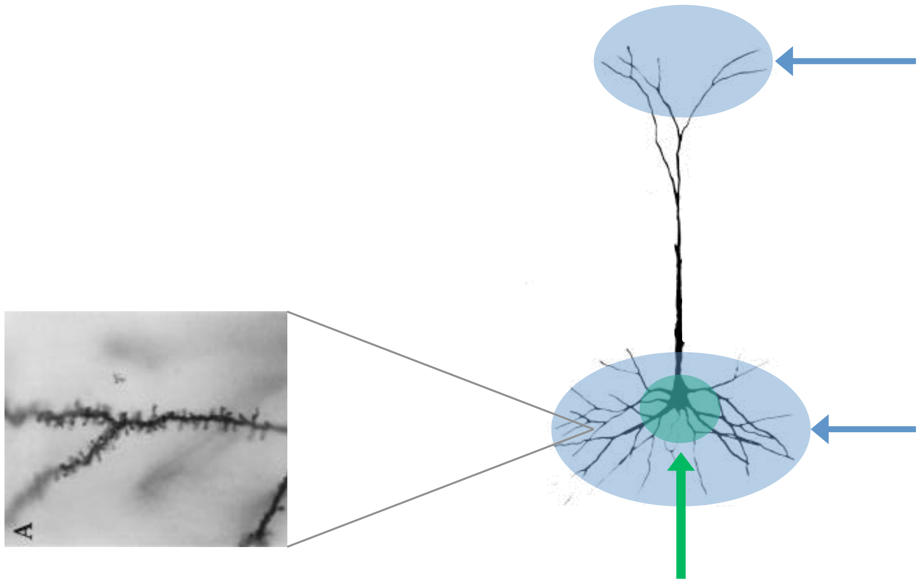
3) **Union membership**:

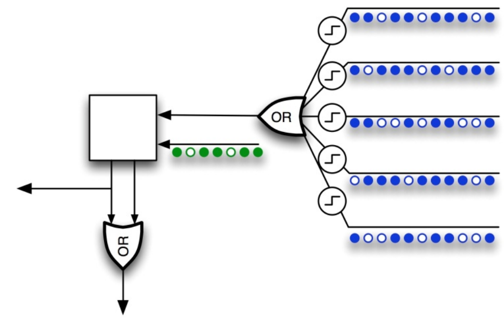
Union

Is this SDR
a member?



Neurons





Active Dendrites

Pattern detectors
Each cell can recognize
100's of unique patterns

Feedforward

100's of synapses
"Classic" receptive field

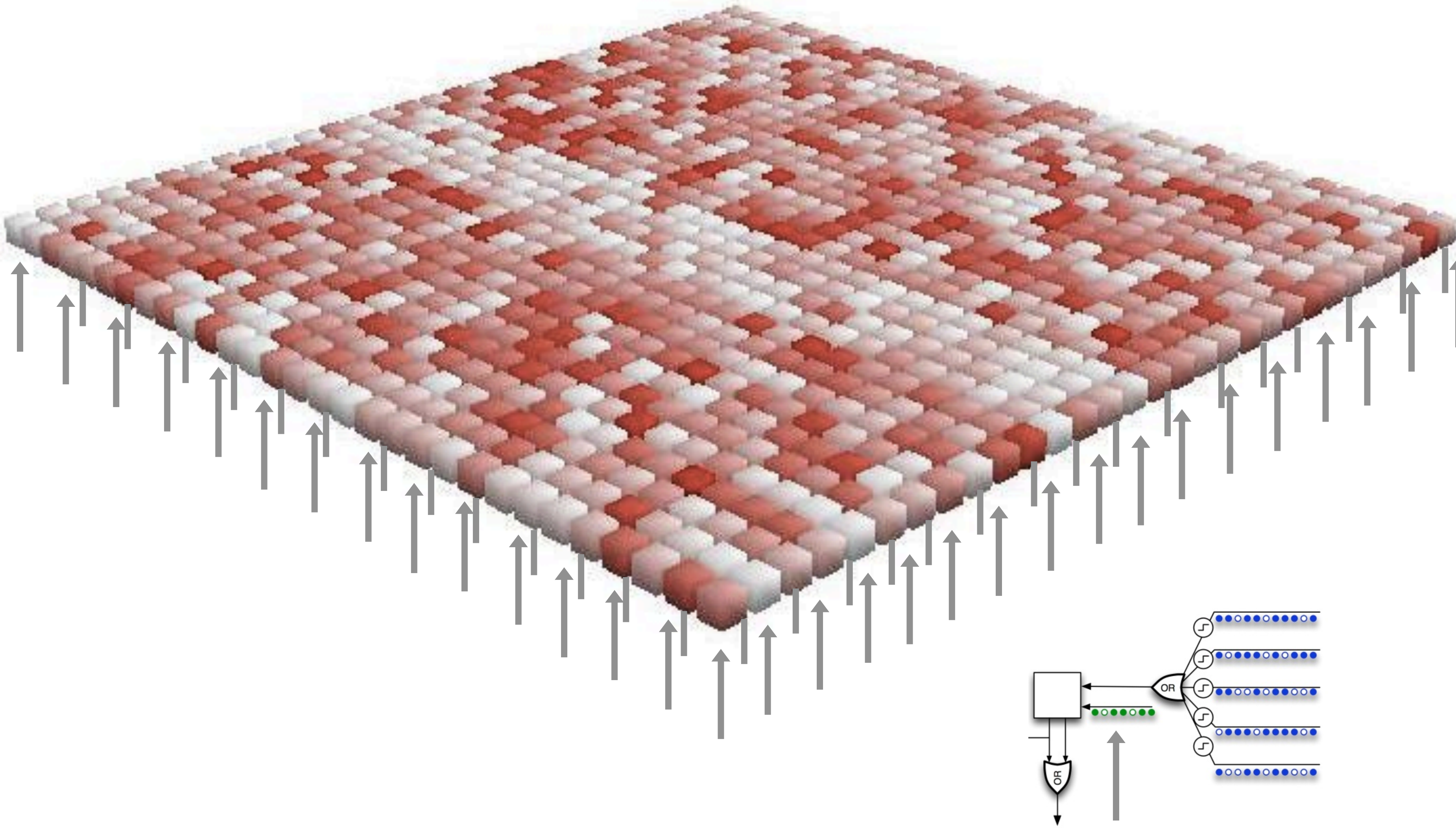
Context

1,000's of synapses
Depolarize neuron
"Predicted state"

Model neuron

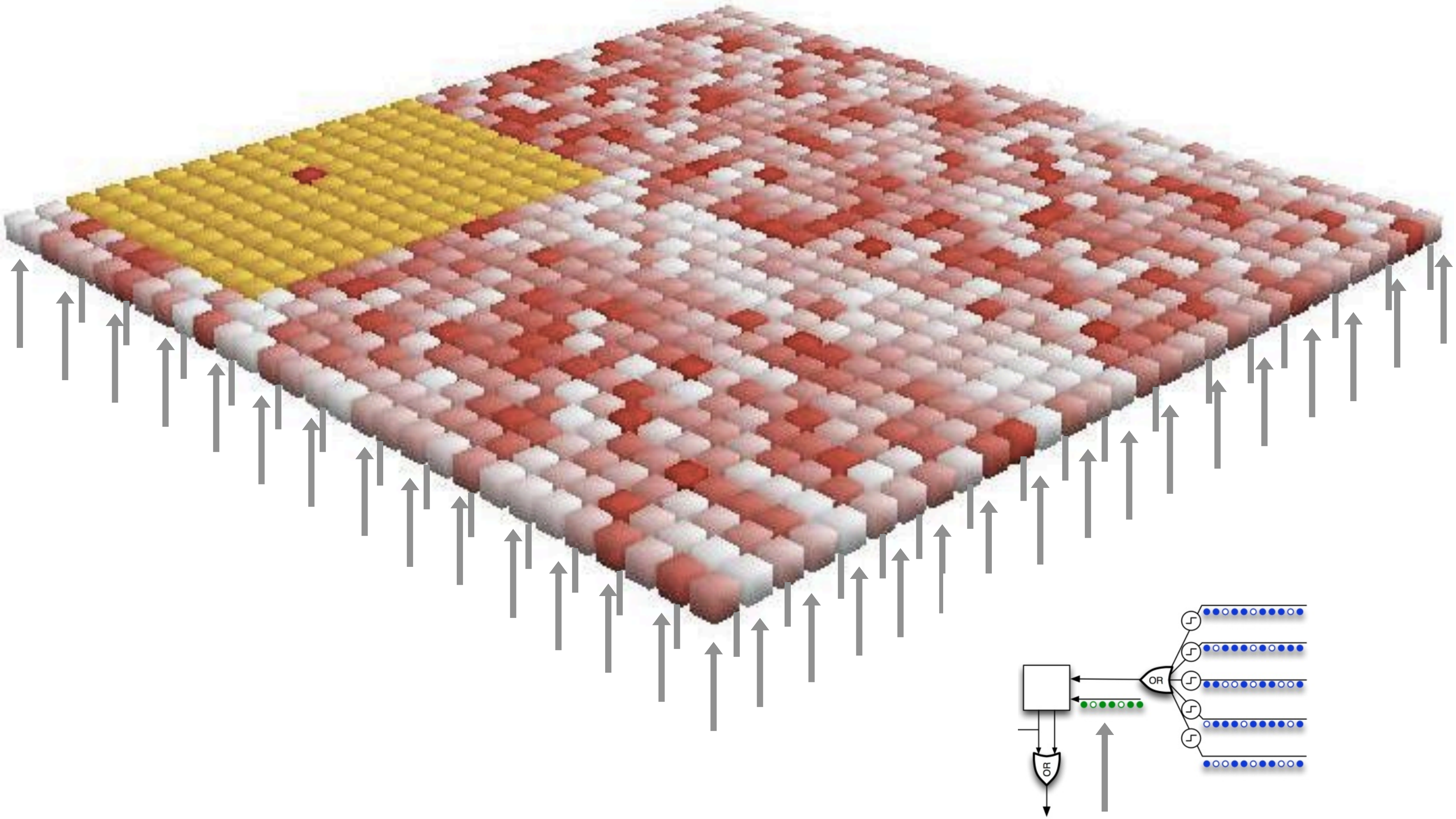
Learning Transitions

Feedforward activation



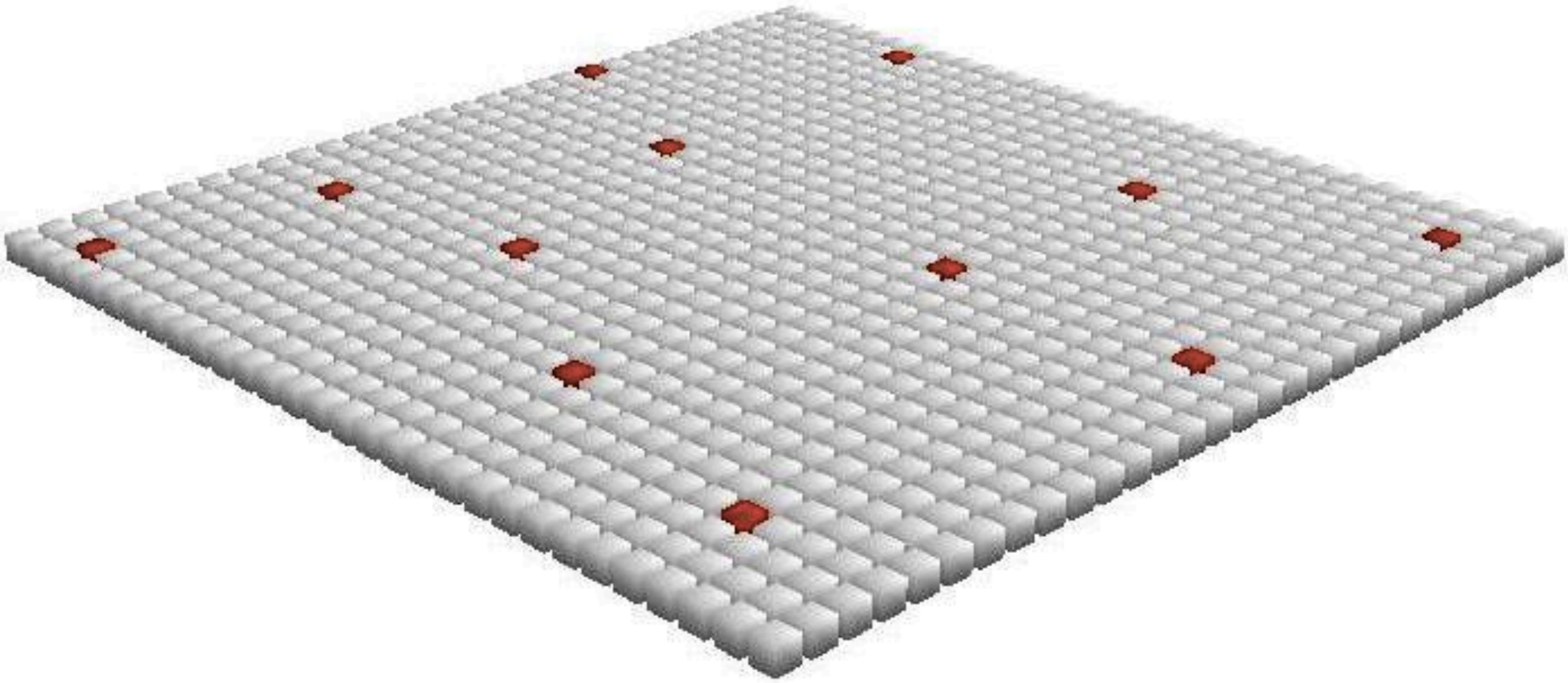
Learning Transitions

Inhibition



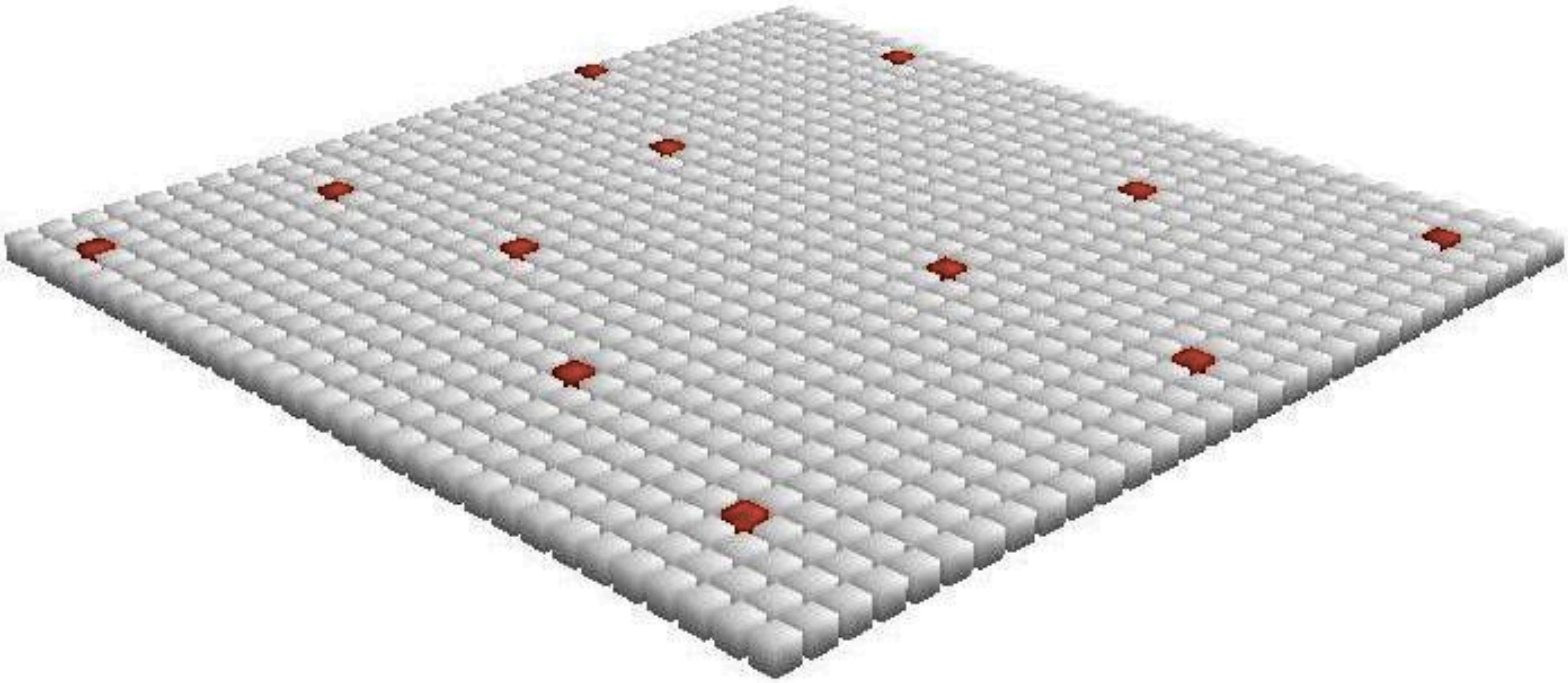
Learning Transitions

Sparse cell activation

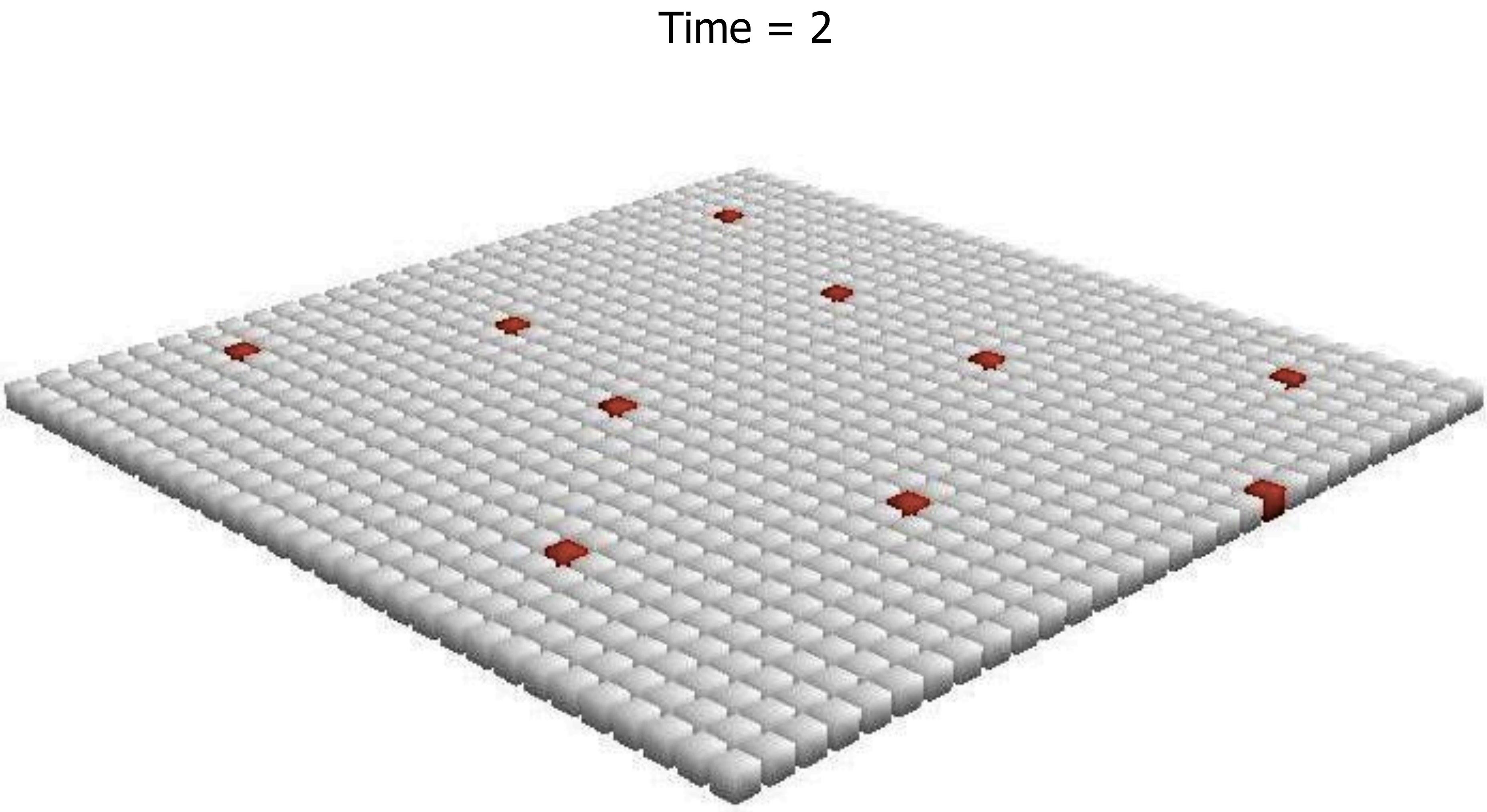


Learning Transitions

Time = 1



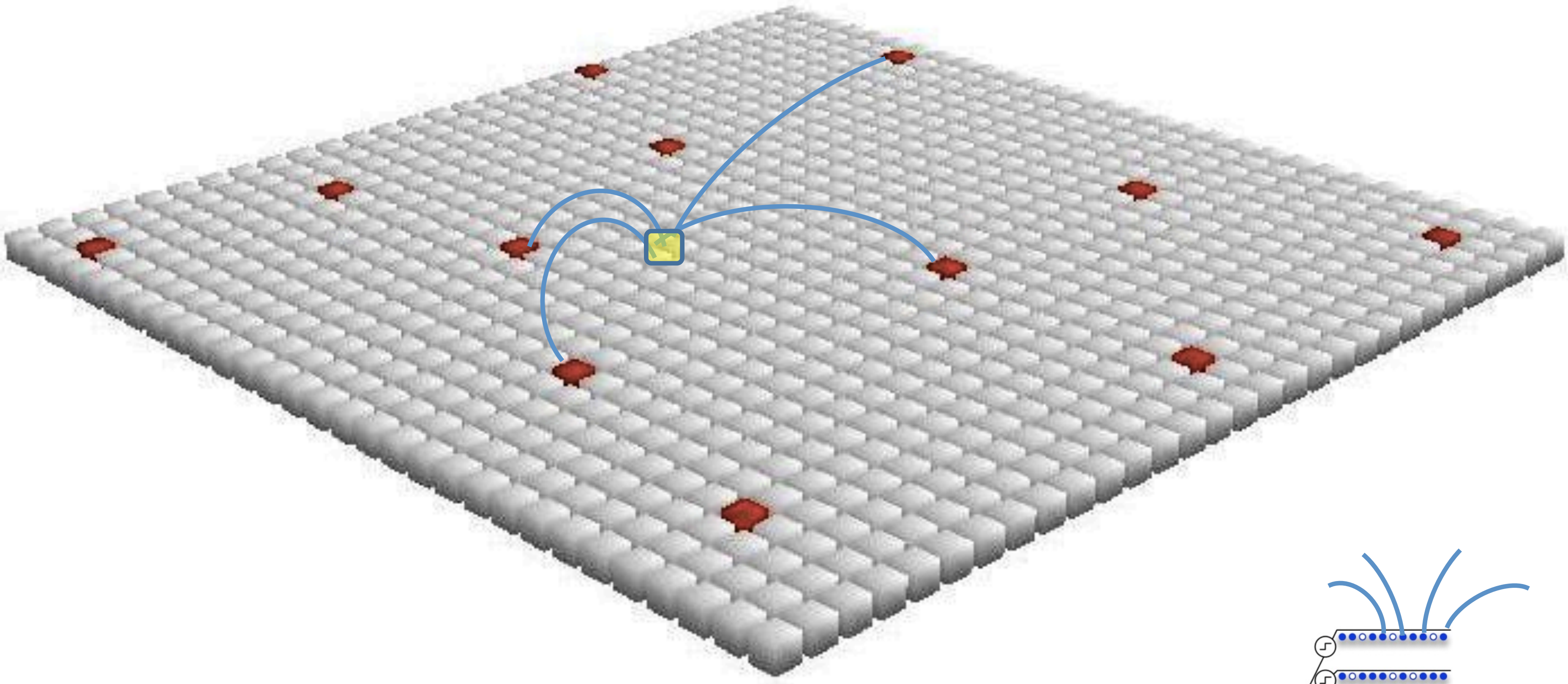
Learning Transitions

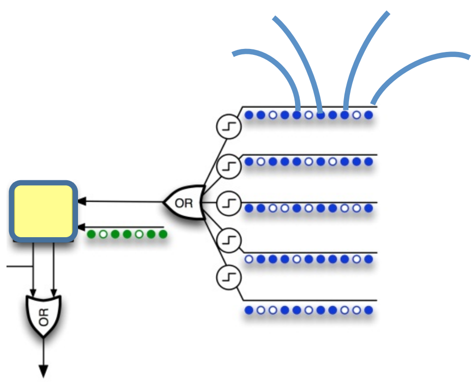


Learning Transitions

Form connections to previously active cells.

Predict future activity.

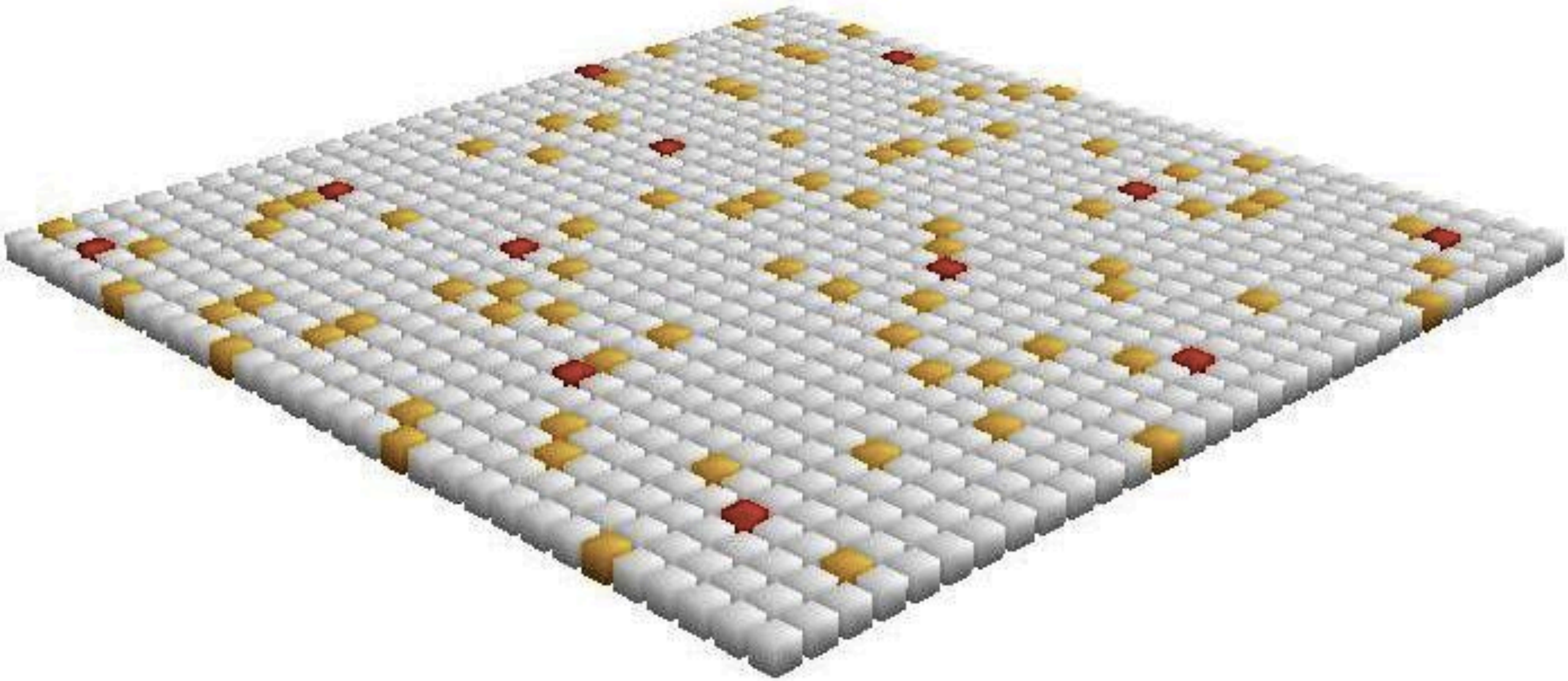




Learning Transitions

Multiple predictions can occur at once.

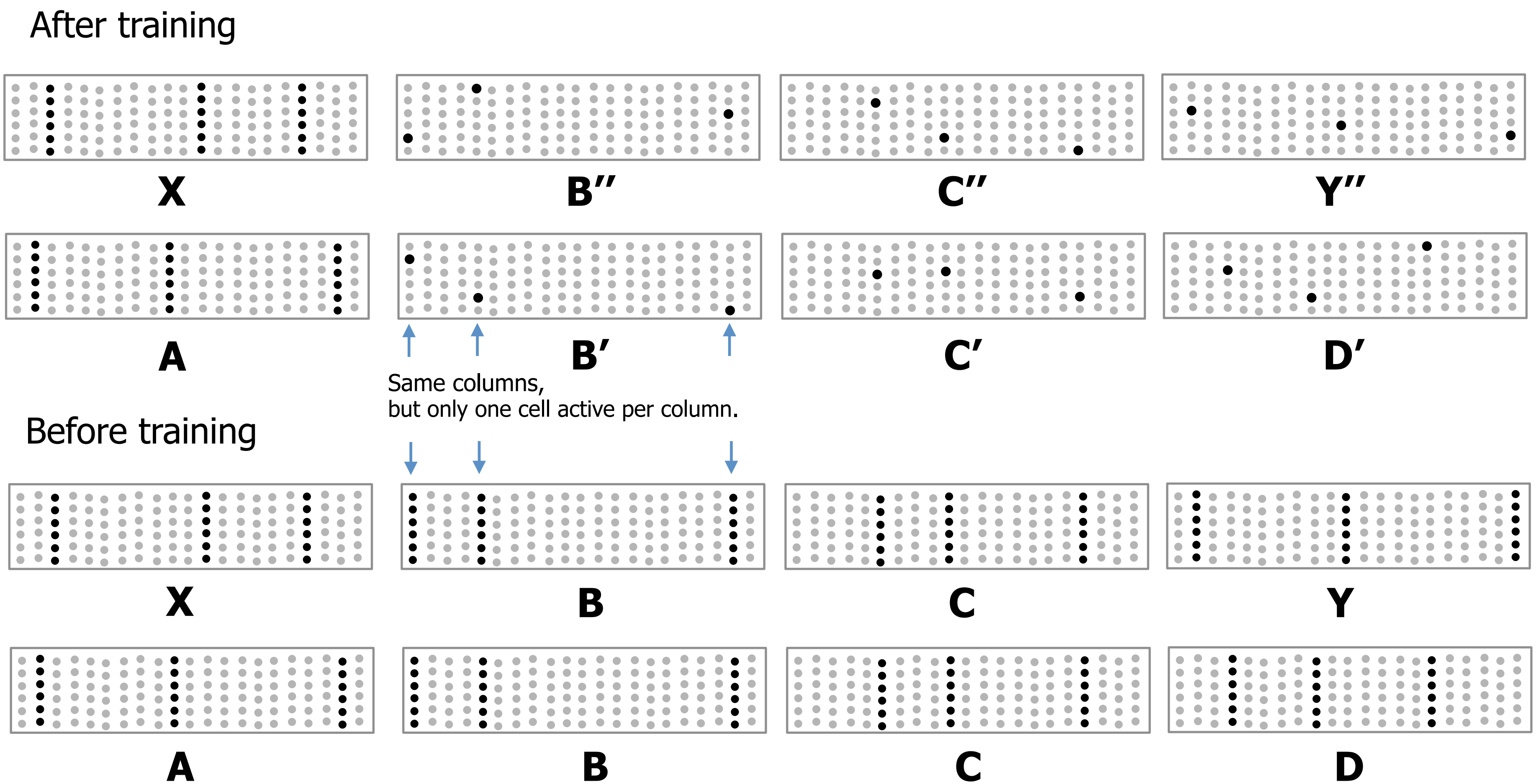
A-B A-C A-D



This is a first order sequence memory.
It cannot learn A-B-C-D vs. X-B-C-Y.

High-order Sequences Require Mini-columns

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

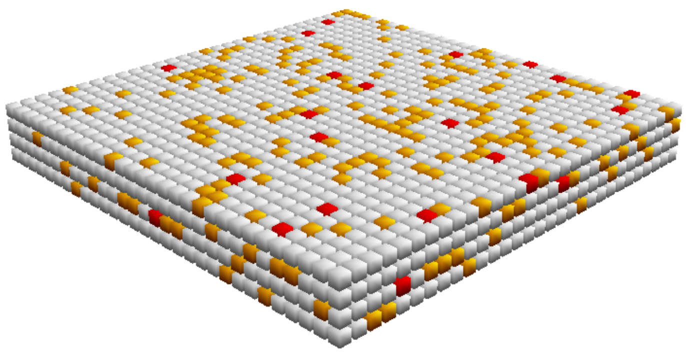


IF **40 active columns, 10 cells per column**

THEN **<u>10^{40}</u> ways to represent the same input in different contexts**

Cortical Learning Algorithm (CLA)
aka Cellular Layer

Converts input to sparse representations in columns
Learns transitions
Makes predictions and detects anomalies



Applications

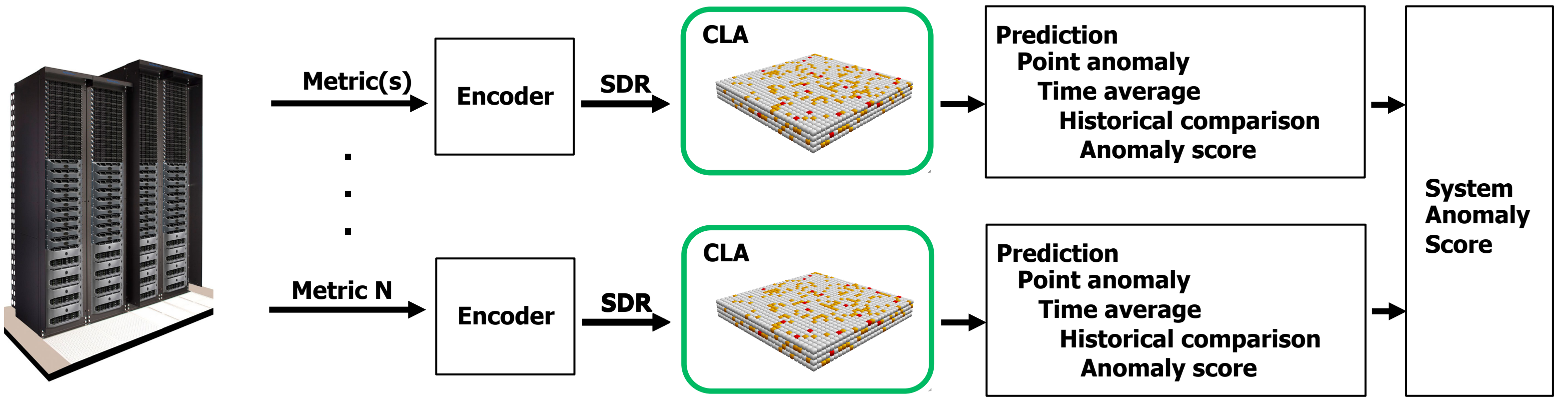
1) High-order sequence inference	L2/3
2) Sensory-motor inference	L4
3) Motor sequence recall	L5

Capabilities

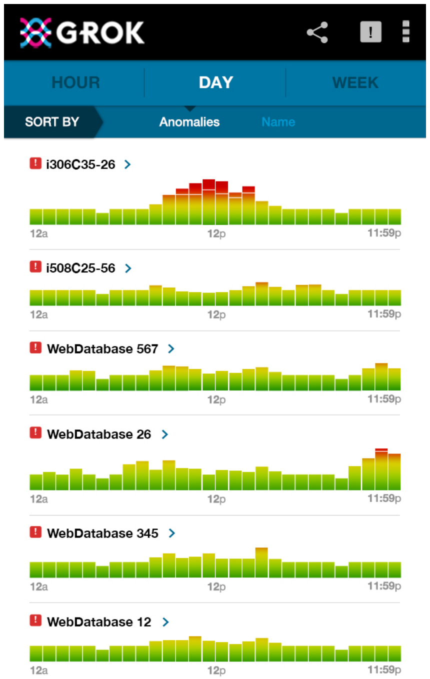
- On-line learning
- High capacity
- Simple local learning rules
- Fault tolerant
- No sensitive parameters

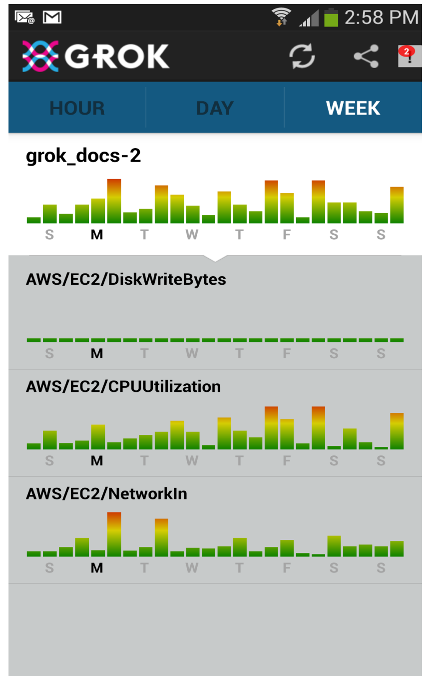
Basic building block of neocortex/Machine Intelligence

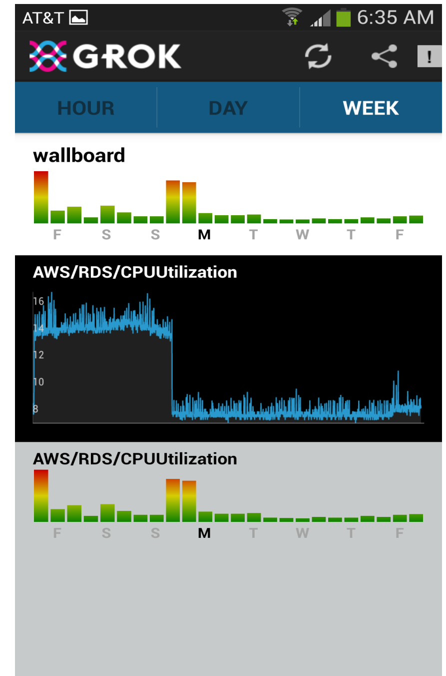
Anomaly Detection Using CLA



Grok for Amazon AWS
“Breakthrough Science for Anomaly Detection”

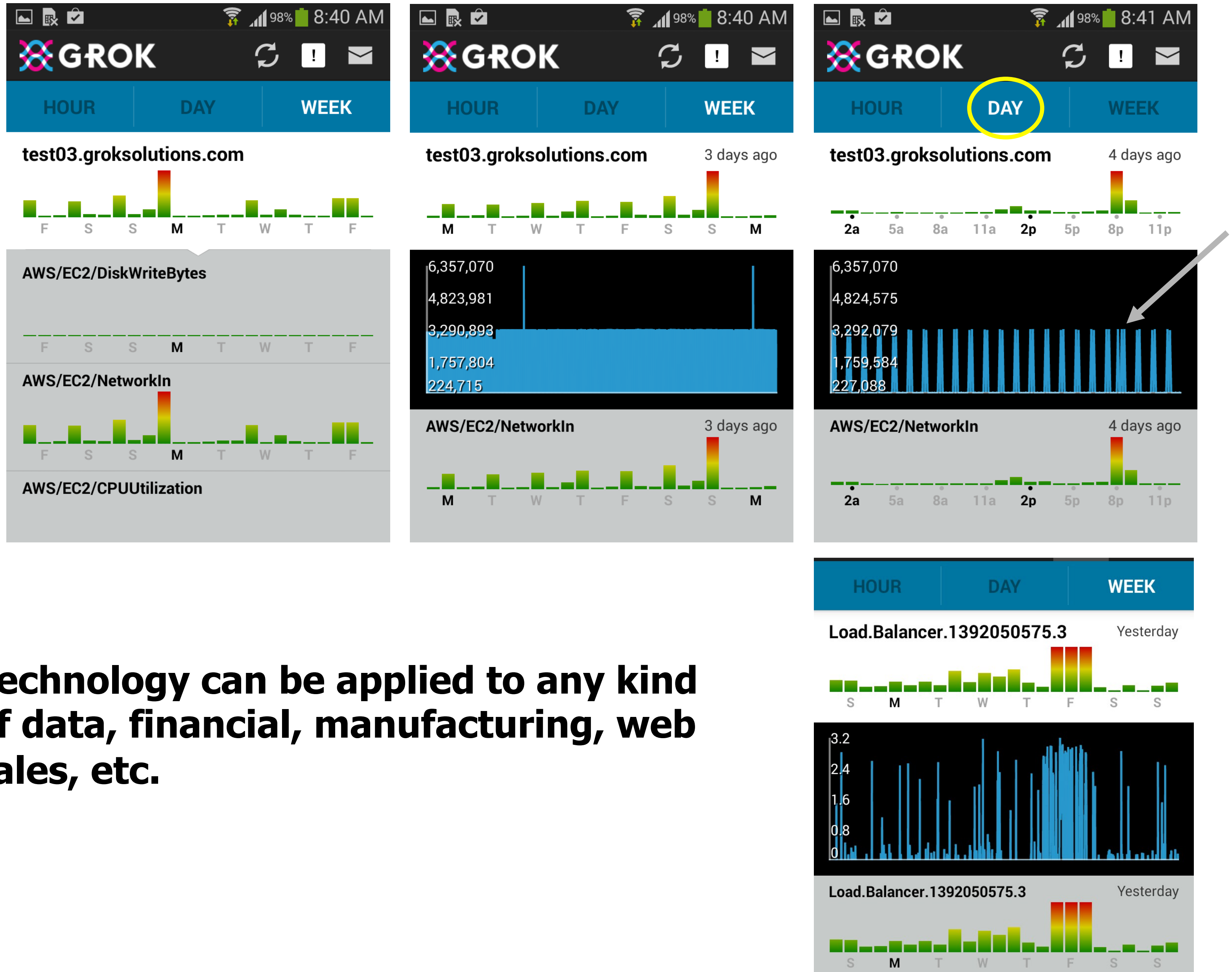






- **Automated model creation**
- **Ranks anomalous instances**
- **Continuously updated**
- **Continuously learning**

Unique Value of Cortical Algorithms



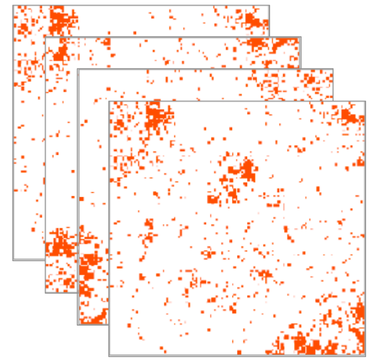
Technology can be applied to any kind of data, financial, manufacturing, web sales, etc.

Application: CEPT Systems

Document corpus
(e.g. Wikipedia)

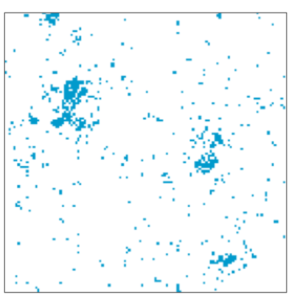
→

100K "Word SDRs"



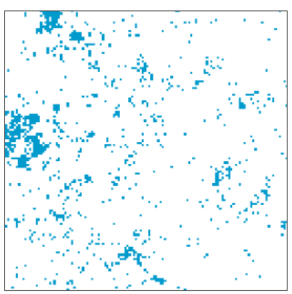
128 x 128

Apple



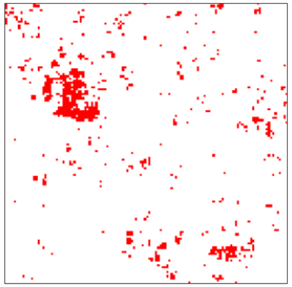
–

Fruit



=

Computer

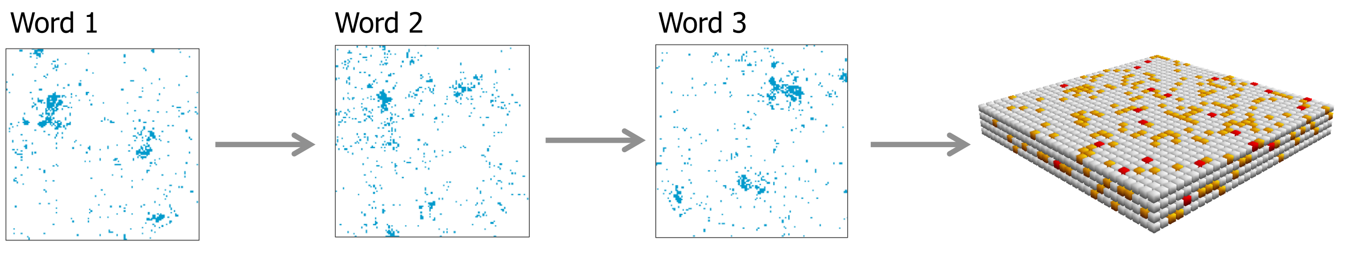


Macintosh
Microsoft
Mac
Linux
Operating system
....

Sequences of Word SDRs

Training set

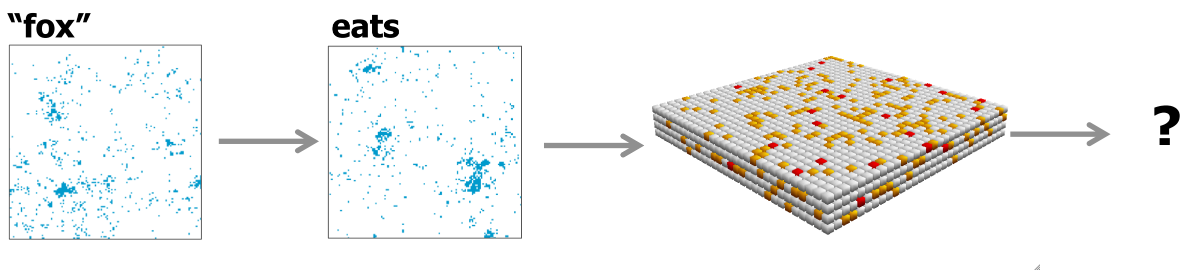
frog	eats	flies
cow	eats	grain
elephant	eats	leaves
goat	eats	grass
wolf	eats	rabbit
cat	likes	ball
elephant	likes	water
sheep	eats	grass
cat	eats	salmon
wolf	eats	mice
lion	eats	cow
dog	likes	sleep
elephant	likes	water
cat	likes	ball
coyote	eats	rodent
coyote	eats	rabbit
wolf	eats	squirrel
dog	likes	sleep
cat	likes	ball
----	----	-----



Sequences of Word SDRs

Training set

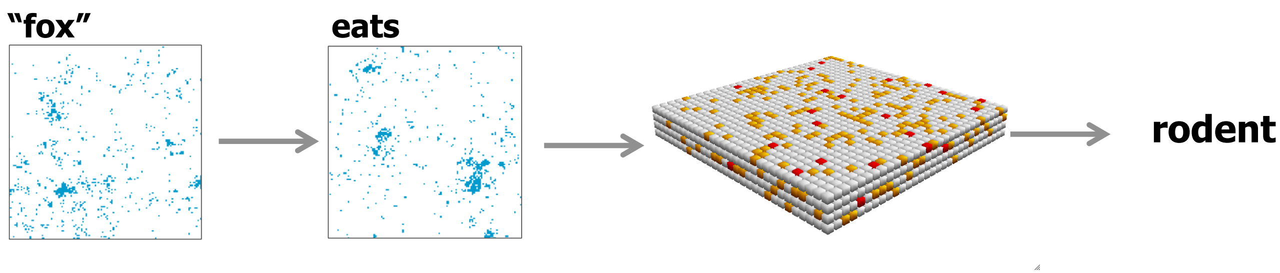
frog	eats	flies
cow	eats	grain
elephant	eats	leaves
goat	eats	grass
wolf	eats	rabbit
cat	likes	ball
elephant	likes	water
sheep	eats	grass
cat	eats	salmon
wolf	eats	mice
lion	eats	cow
dog	likes	sleep
elephant	likes	water
cat	likes	ball
coyote	eats	rodent
coyote	eats	rabbit
wolf	eats	squirrel
dog	likes	sleep
cat	likes	ball
----	----	-----



Sequences of Word SDRs

Training set

frog	eats	flies
cow	eats	grain
elephant	eats	leaves
goat	eats	grass
wolf	eats	rabbit
cat	likes	ball
elephant	likes	water
sheep	eats	grass
cat	eats	salmon
wolf	eats	mice
lion	eats	cow
dog	likes	sleep
elephant	likes	water
cat	likes	ball
coyote	eats	rodent
coyote	eats	rabbit
wolf	eats	squirrel
dog	likes	sleep
cat	likes	ball
----	----	-----



1) Word SDRs created unsupervised

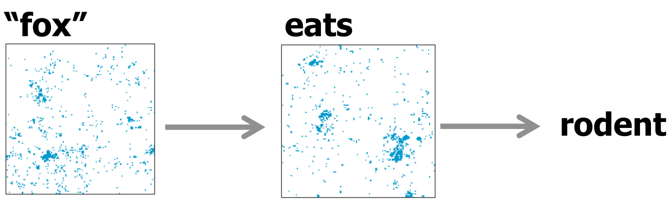
2) Semantic generalization

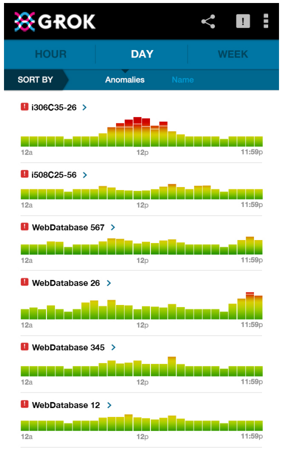
SDR: lexical
CLA: grammatic

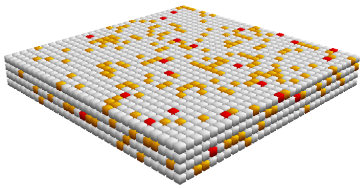
3) Commercial applications

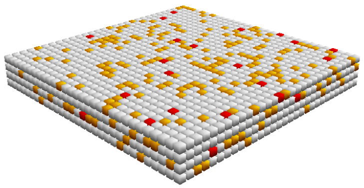
Sentiment analysis
Abstraction
Improved text to speech
Dialog, Reporting, etc.
www.Cept.at

Cept and Grok use exact same code base









NuPIC Open Source Project (created July 2013)

Source code for:

- Cortical Learning Algorithm
- Encoders
- Support libraries

Single source tree (used by GROK), GPLV3

Active and growing community

- 80 contributors as of 2/2014
- 533 mailing list subscribers

Education Resources / Hackathons

www.Numenta.org

Goals For 2014

<u>Research</u>

- Implement and test L4 sensory/motor inference
- Introduce hierarchy
- Publish

<u>NuPIC</u>

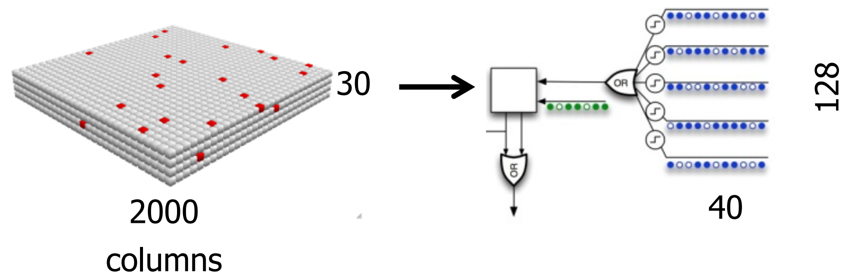
- Grow open source community
- Support partners, e.g. IBM Almaden, CEPT

<u>Grok</u>

- Demonstrate commercial value for CLA
Attract resources (people and money)
Provide a "target" and market for HW

- Explore new application areas

The Limits of Software



- One layer, no hierarchy
- 2000 mini-columns
- 30 cells per mini-column
- Up to 128 dendrite segments per cell
- Up to 40 active synapses per segment
- Thousands of potential synapses per segment

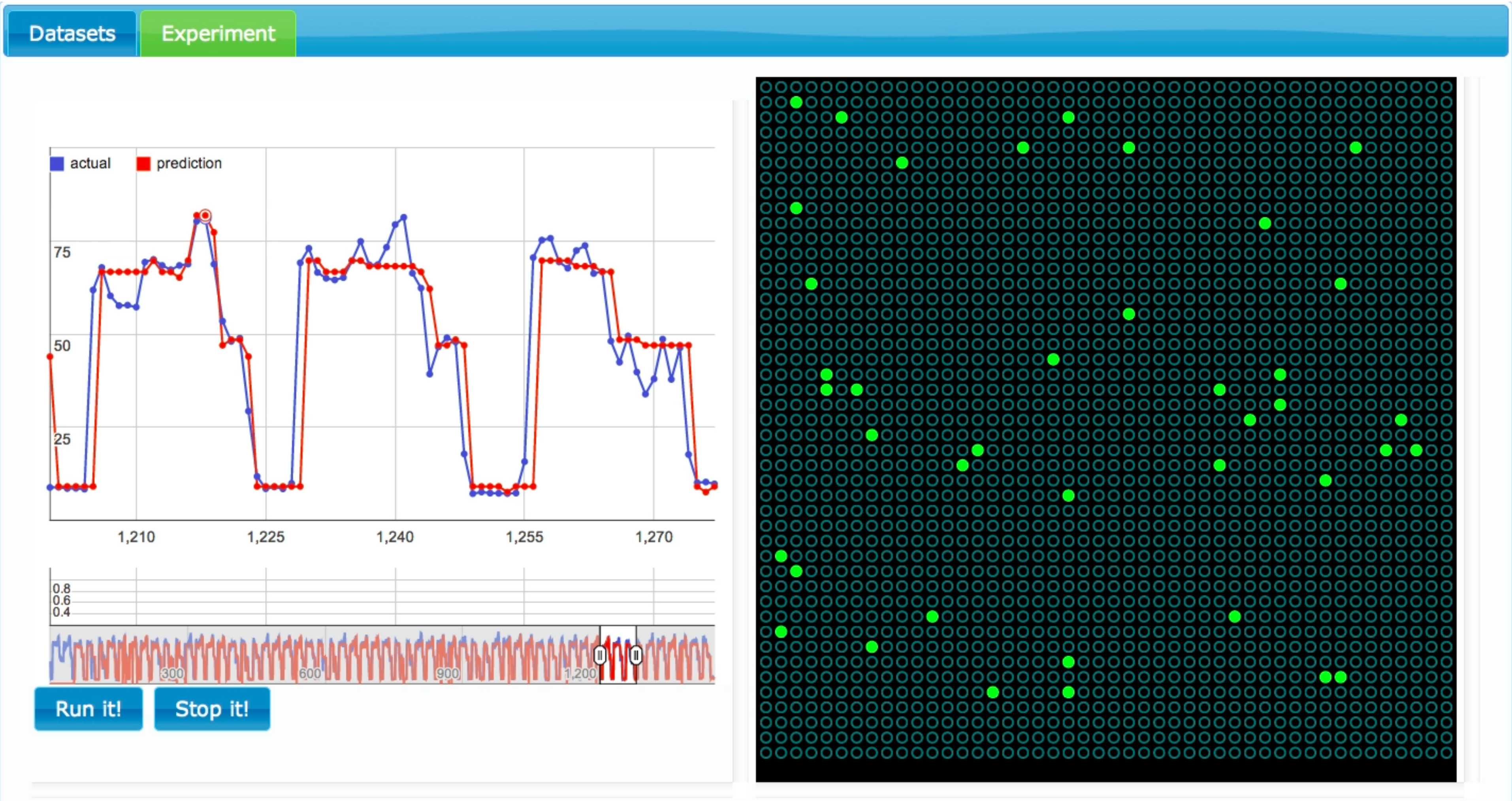
One millionth human cortex

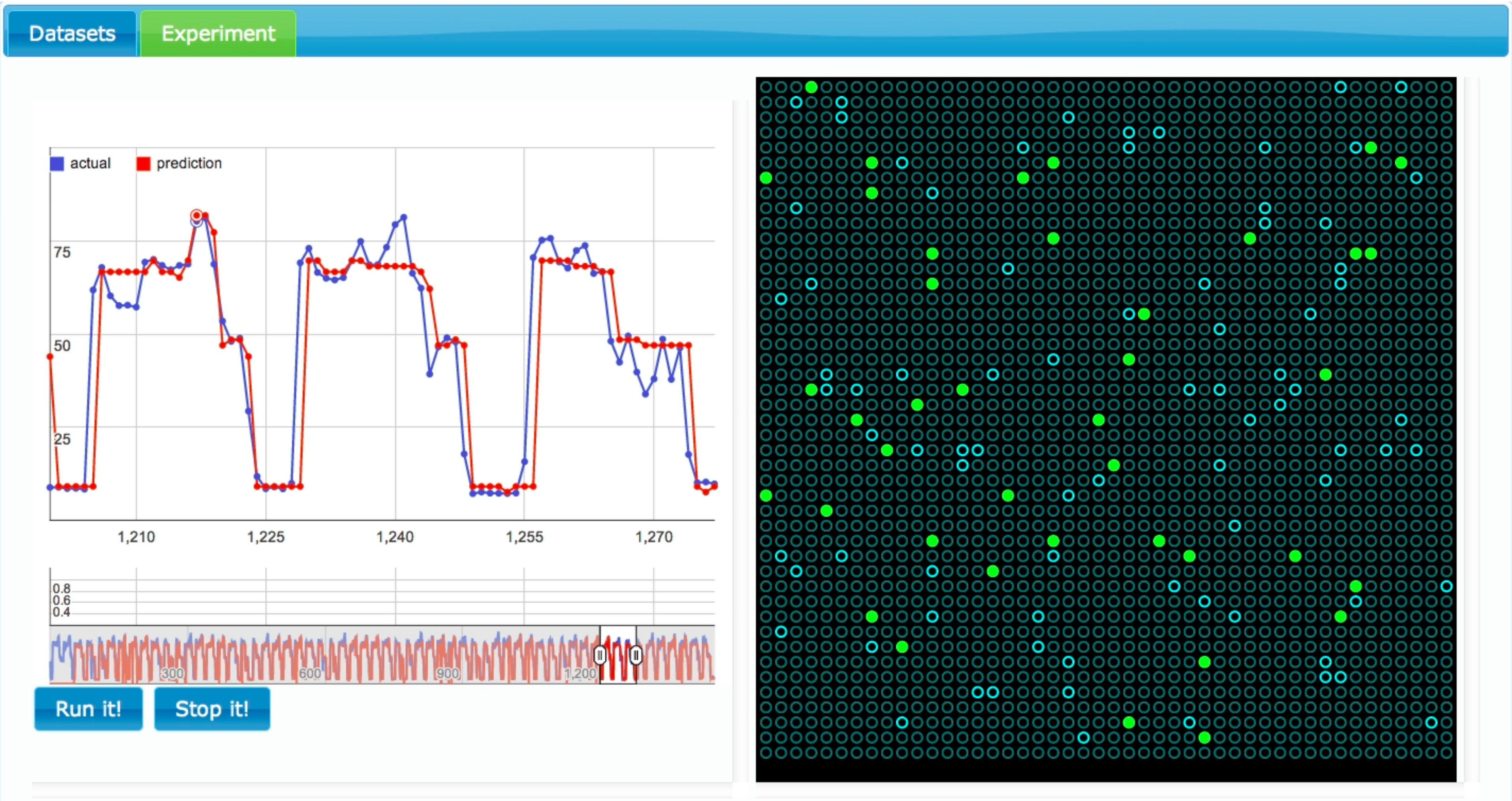
50 msec per inference/training step **(highly optimized)**

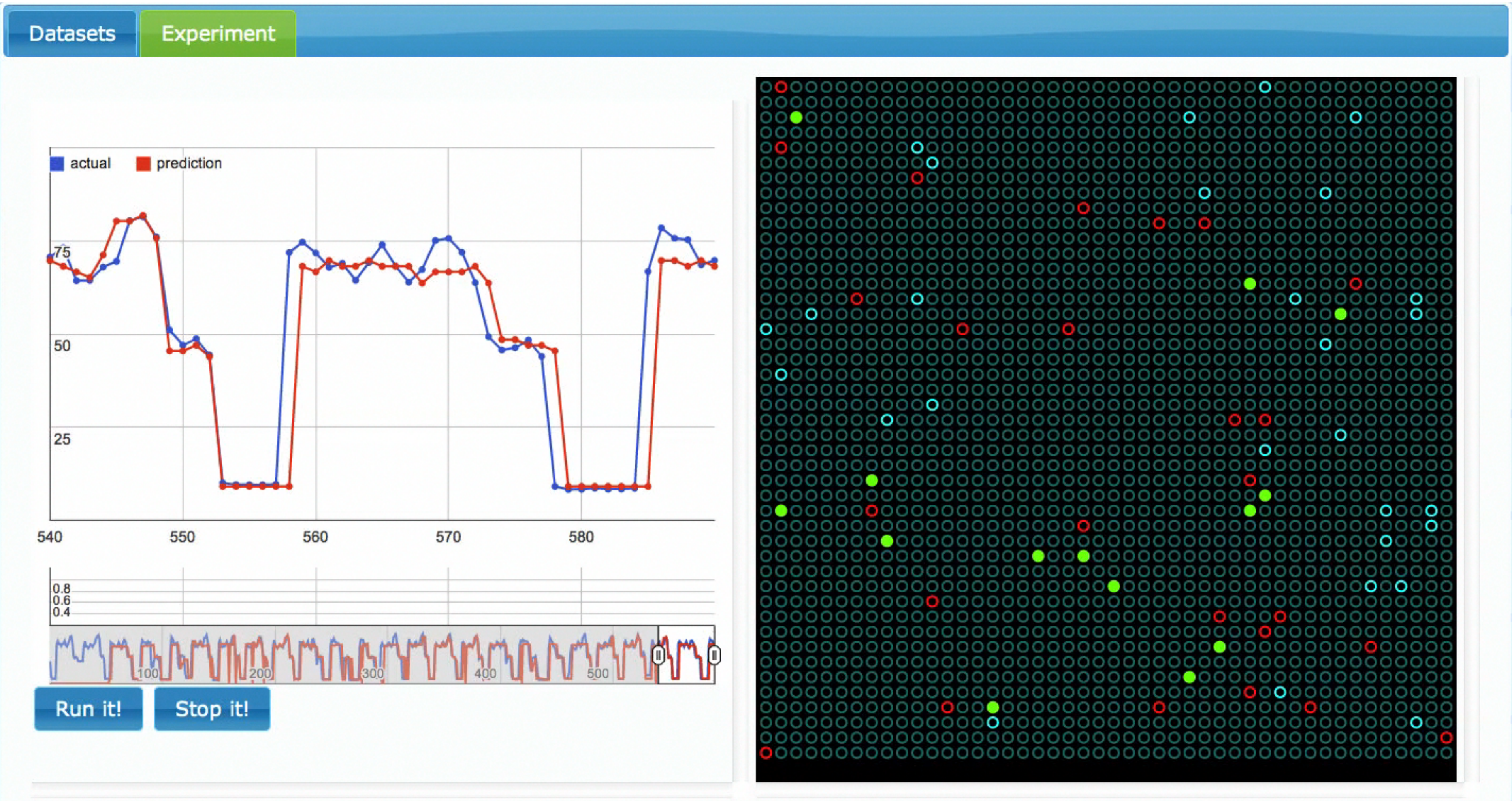
We need HW for the usual reasons.

- **Speed**
- **Power**
- **Volume**

Start of supplemental material

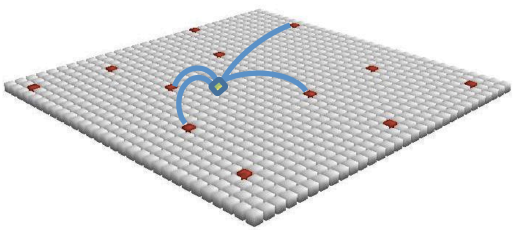

Datasets
Experiment
actual
prediction
75
50
25
1,210
1,225
1,240
1,255
1,270
0.8
0.6
0.4
300
600
900
1,200
Run it!
Stop it!

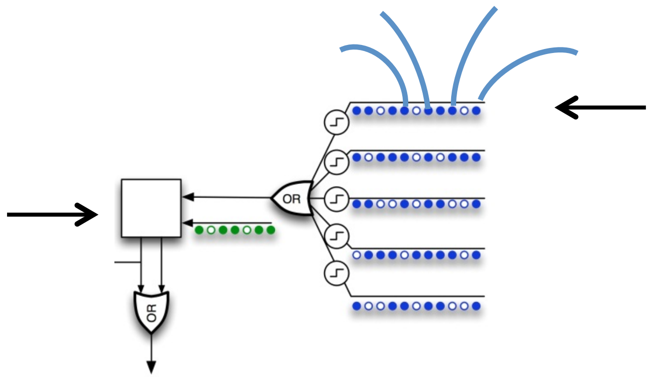

Datasets
Experiment
actual
prediction
75
50
25
1,210
1,225
1,240
1,255
1,270
0.8
0.6
0.4
300
600
900
1,200
Run it!
Stop it!

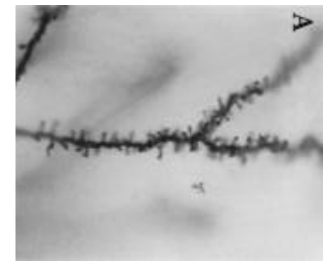

Datasets
Experiment
actual
prediction
75
50
25
540
550
560
570
580
0.8
0.6
0.4
100
200
300
400
500
Run it!
Stop it!

Requirements for Online learning

- Train on every new input
- If pattern does not repeat, forget it
- If pattern repeats, reinforce it







Learning is the formation of connections

Connection permanence

unconnected | connected

0 — 0.2 — 1

Connection strength/weight is binary

Connection permanence is a scalar

Training changes permanence

If permanence > threshold then connected

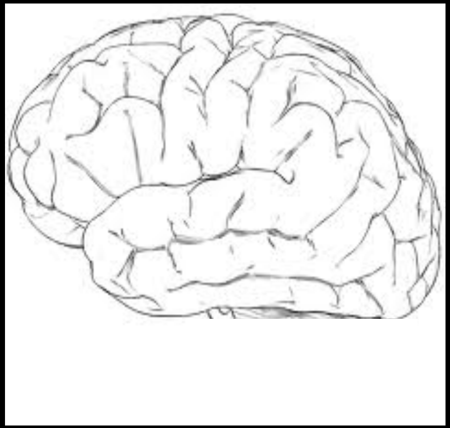
“If you invent a breakthrough so computers can learn, that is worth 10 Microsofts”

Machine Intelligence

computers that learn

1) What principles will we use to build intelligent machines?

2) What applications will drive adoption in the near and long term?



Machine intelligence will be built on the principles of the neocortex

1) Flexible (universal learning machine)

2) Robust

3) If we knew how the neocortex worked, we would be in a race to build them.