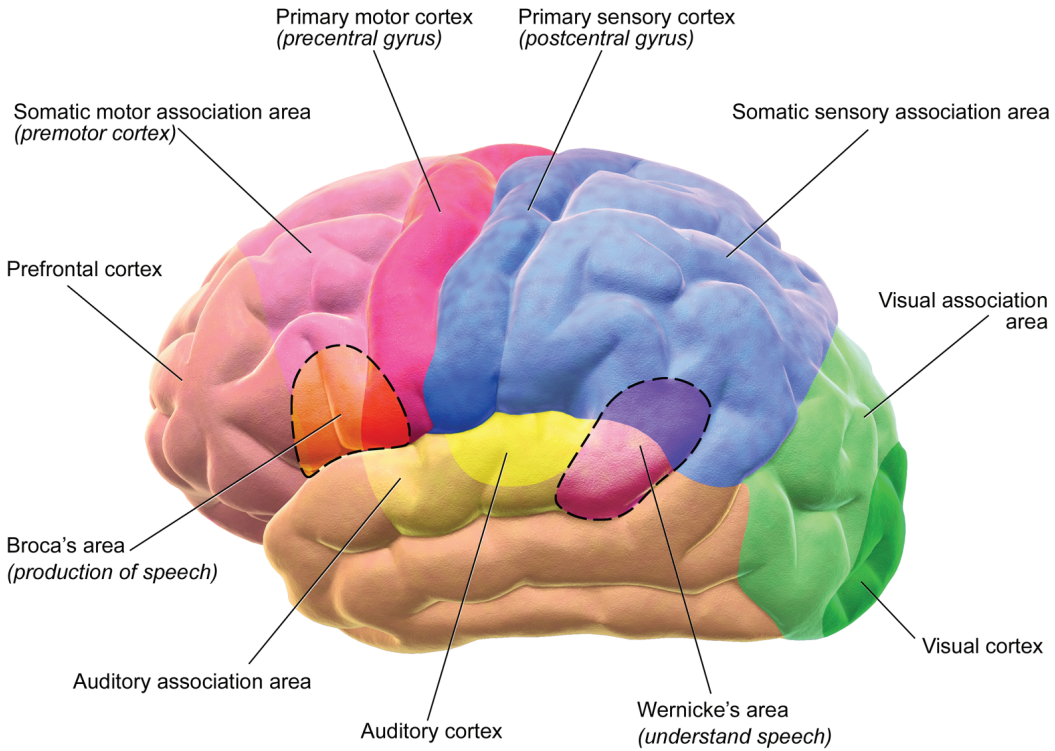
HTM

Michoel Snow

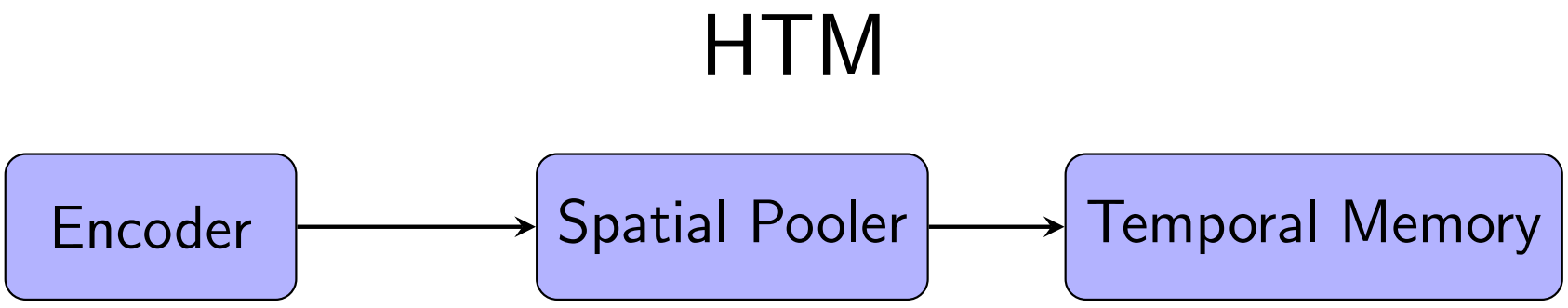
June 16, 2017

Cerebral Cortex and the HTM

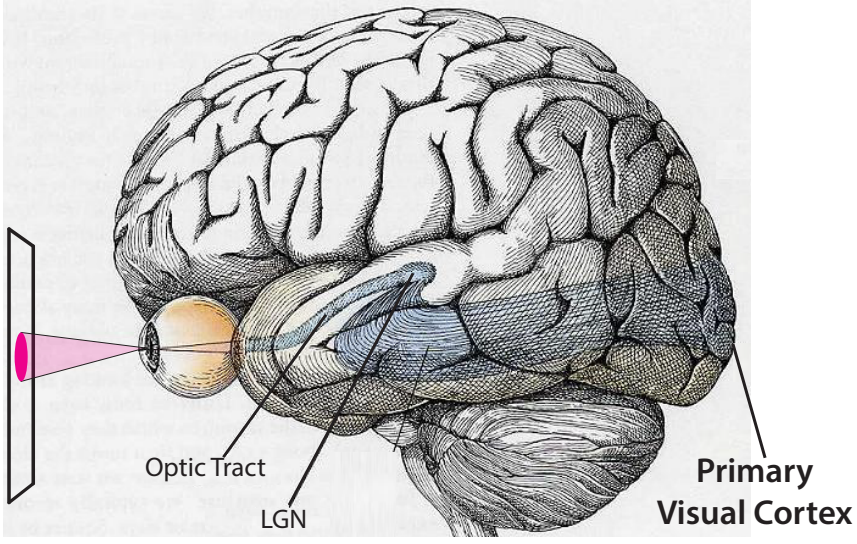


- The cortex processes incoming sensory and motor data
- HTM aims to mimic the structural and algorithmic properties of the "neocortex" to build better machine learning tools

Hierarchical Temporal Memory (HTM)



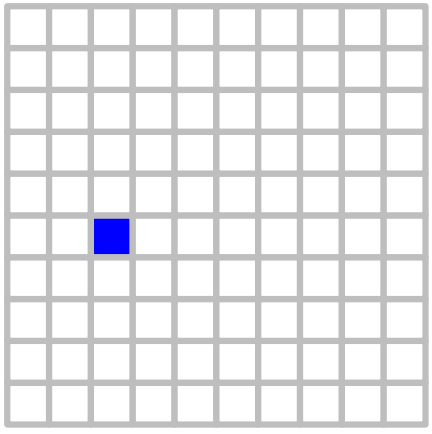
Neural Encoding



- Neurons respond to analog sensory input in a binary manner, either firing an action potential or not

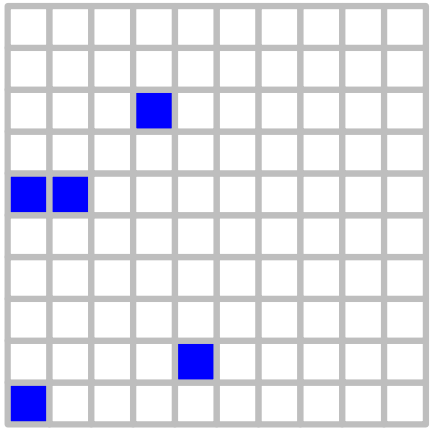
Sparse Coding

Local

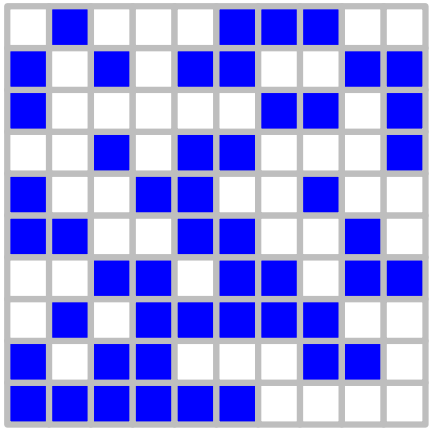


- N possible states
- Linearly separable learning and decoding
- Simultaneous transmission

Sparse



Dense



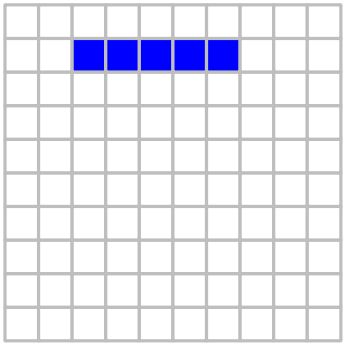
- 2^N possible states
- Fault tolerant
- Robust

HTM Encoding

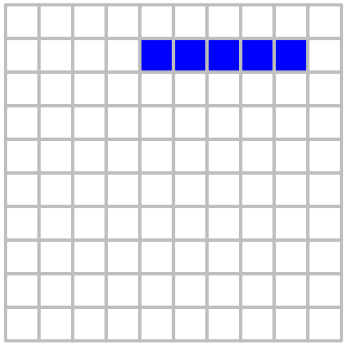
- Semantic similarity in inputs results in overlapping Sparse Distributed Representation (SDR) representations
- Same input should always give the same SDR output
- Output should always have the same dimensionality (total number of bits) for all inputs
- Output should have similar density (sparsity) for all inputs (with enough active bits to handle noise and subsampling)

Scalar Encoders

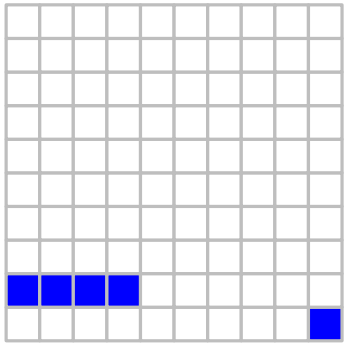
85°



87°



12°



Design Goals

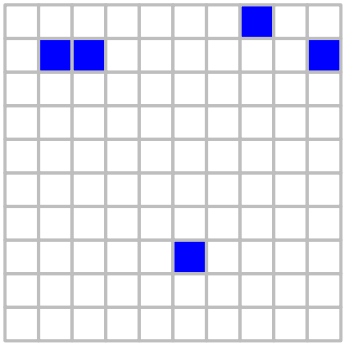
- Semantic similarity
- Input $\rightarrow$ output
- Dimensionality
- Sparsity

Settings

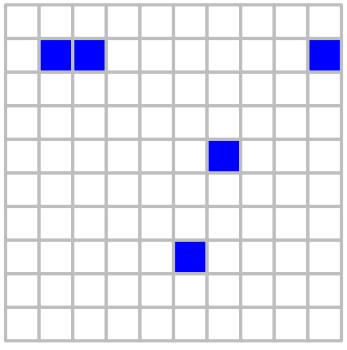
- Number of bits
- Sparsity
- Resolution
- Min - Max

Random Distributed Scalar Encoder

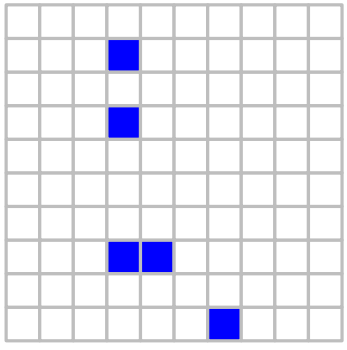
85°



87°



12°



Design Goals

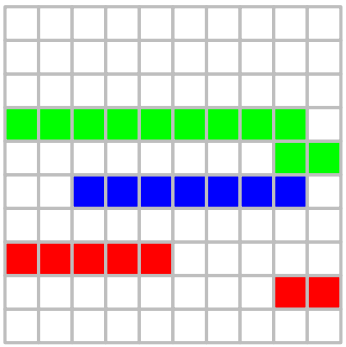
- Semantic similarity
- Input $\rightarrow$ output
- Dimensionality
- Sparsity

Settings

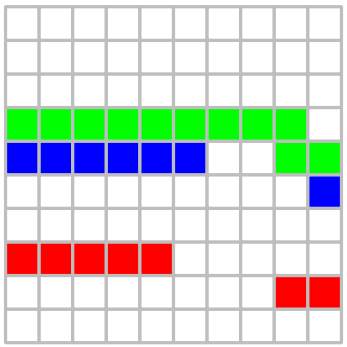
- Number of bits
- Sparsity
- Resolution

Date Encoder

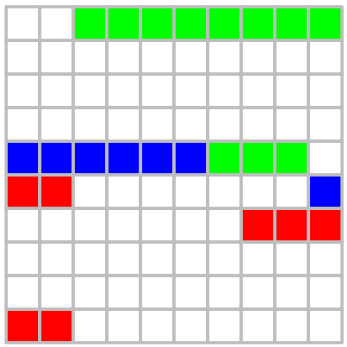
3/1/17 12:00 pm



3/4/17 12:00 pm



12/15/17 11:00 pm



Design Goals

- Semantic similarity
- Input $\rightarrow$ output
- Dimensionality
- Sparsity

Settings

- Sparsity
- Resolution
- Wrap around

Options

- Time of Day
- Day of Week
- Weekend
- Season

Summary

HTM Encoders

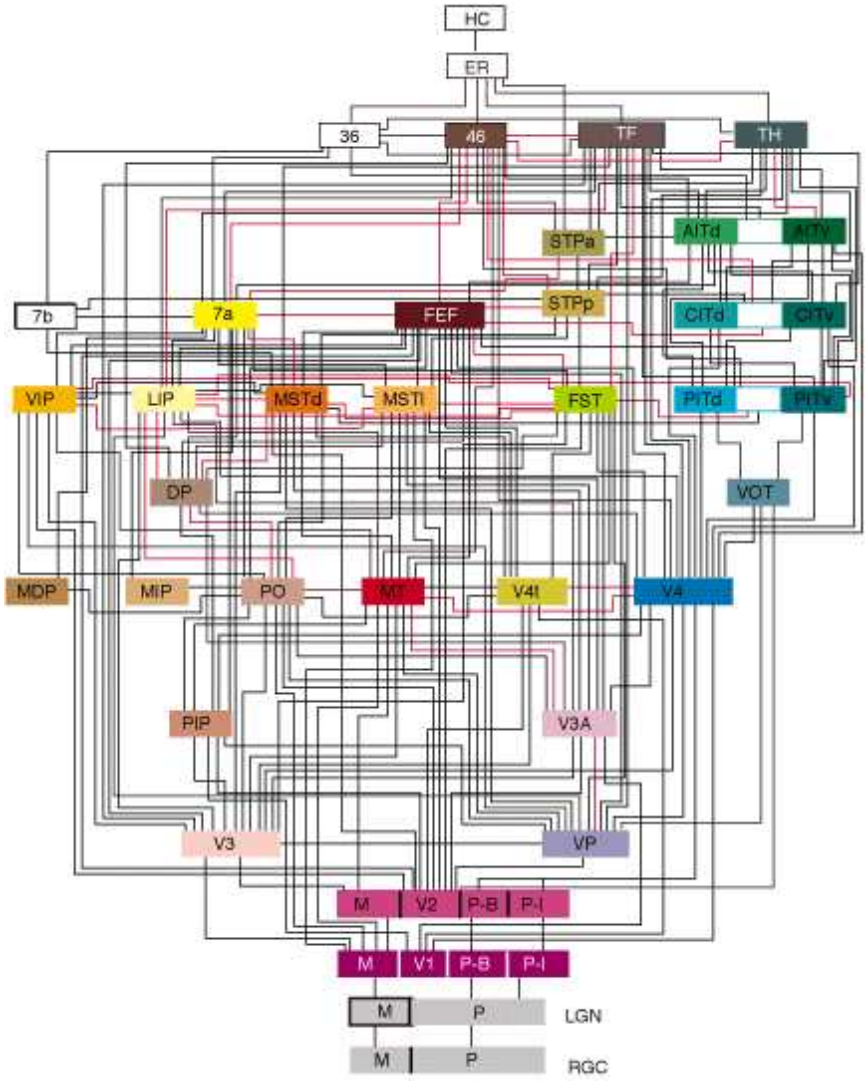
- Scalar Encoder
- RDSE
- Date Encoder
- Category Encoder
- Coordinate Encoder

Our Encoders

- Date encoder with Federal Holidays

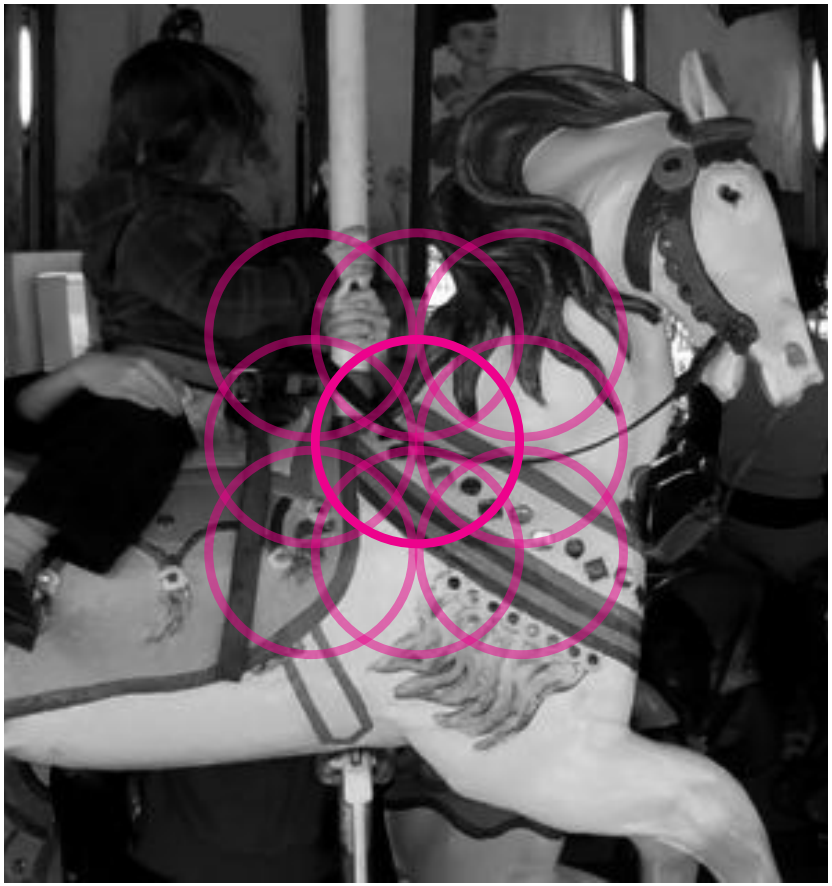
Questions?

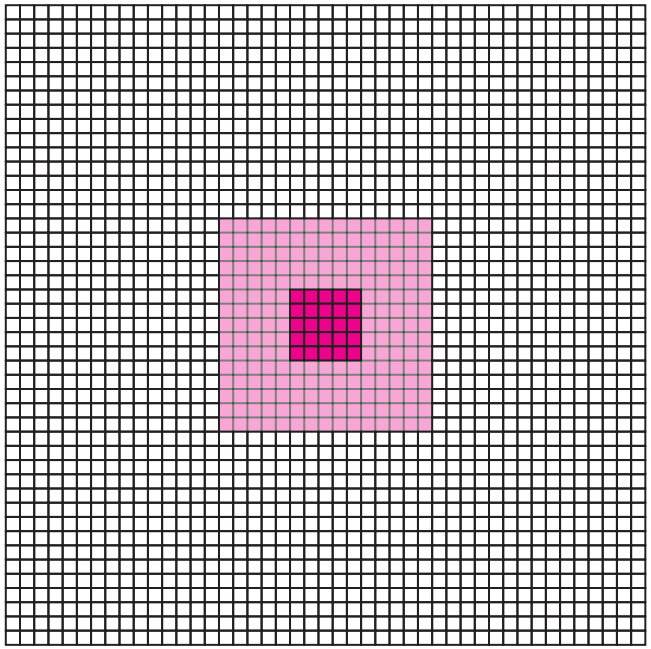
Developmental Hierarchy



- The visual hierarchy has developed on an evolutionary scale
- The visual cortex is attuned to the properties of the viusal environment
- Higher cortical areas respond to progressively more complex signals

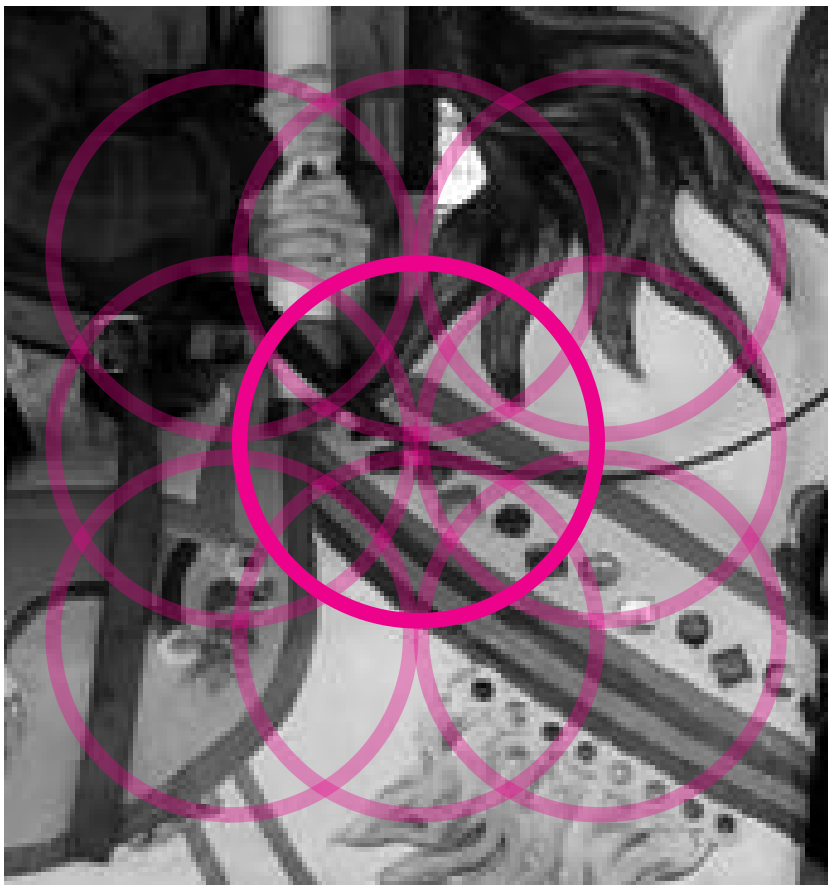
Receptive Fields

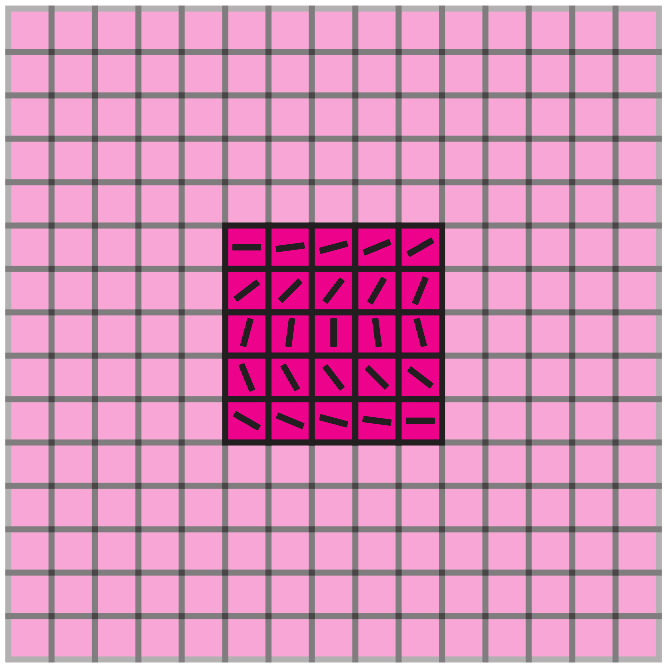




- A neuron's receptive field (RF) is the region of sensory space in which a stimulus will modify the response pattern of that neuron

Receptive Fields

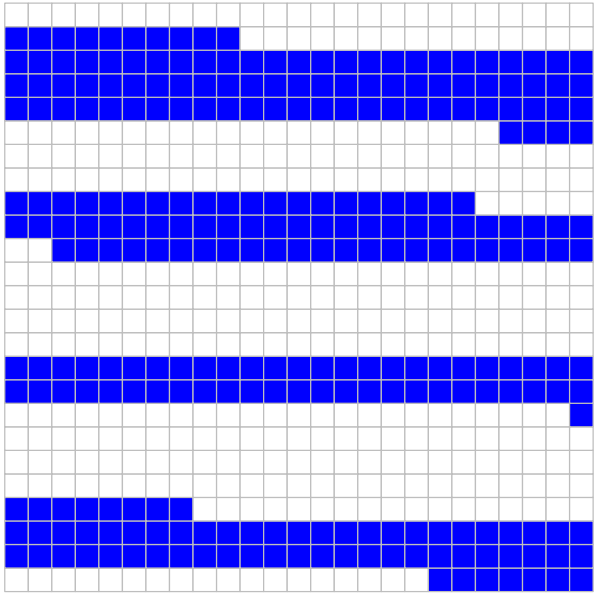




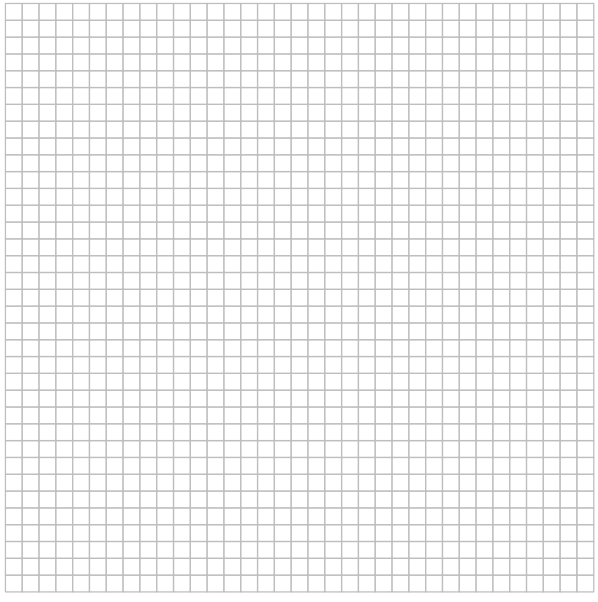
- A neuron's receptive field (RF) is the region of sensory space in which a stimulus will modify the response pattern of that neuron

Spatial Pooler

Input Space (encoder)



Spatial Pooler



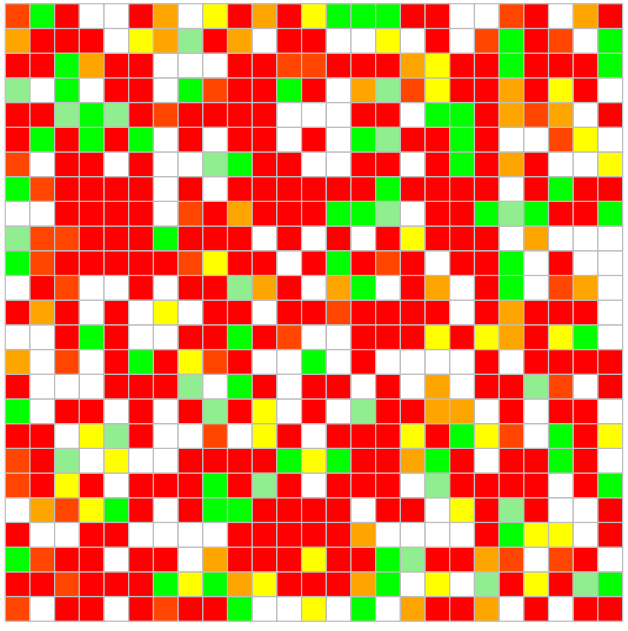
- SP is made up of columns which have connections to the input space

Spatial Pooler Design Goals

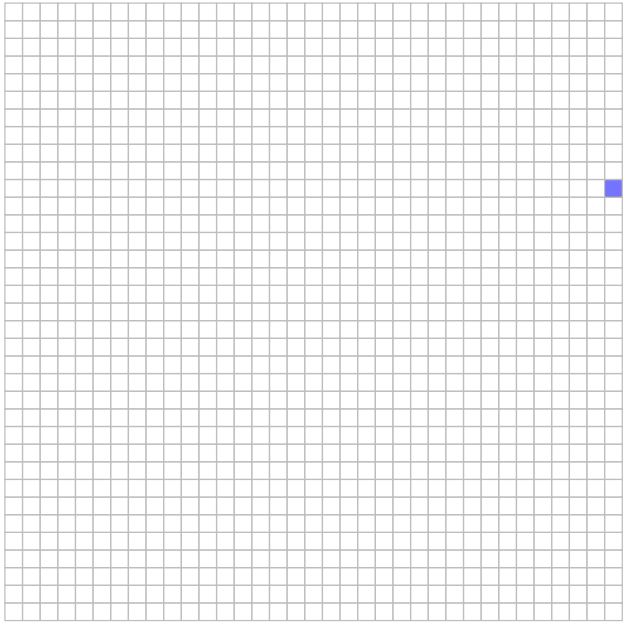
- Maintain SP active column sparsity
- Avoid trivial patterns
- All columns should learn to represent something useful
- Avoid extra connections

Connection Permanence

Input Space (encoder)



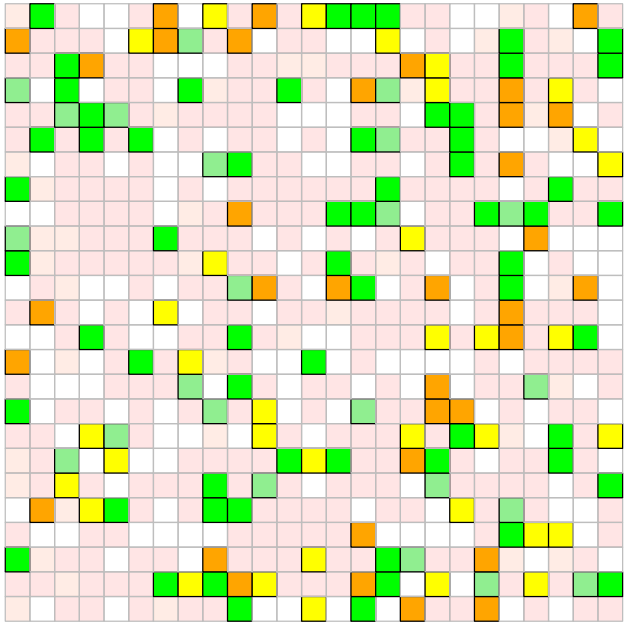
Spatial Pooler



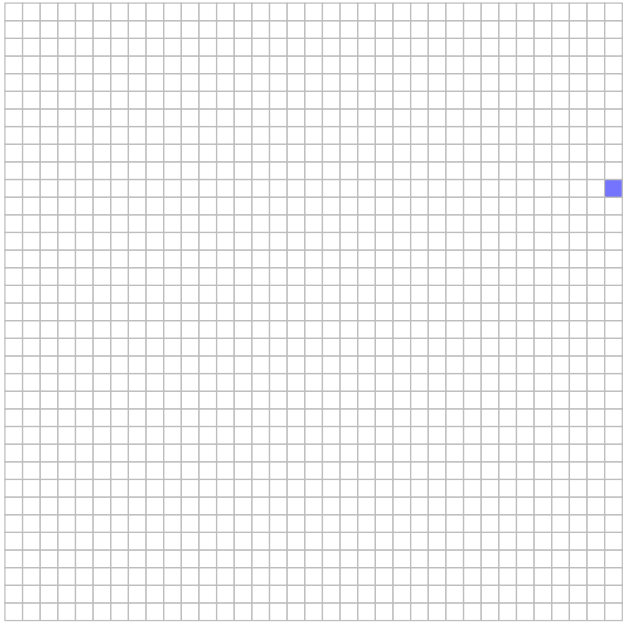
- Each SP column is connected to a percentage of the input space and each potential connection has an associated strength or permanence.

Connection Permanence

Input Space (encoder)



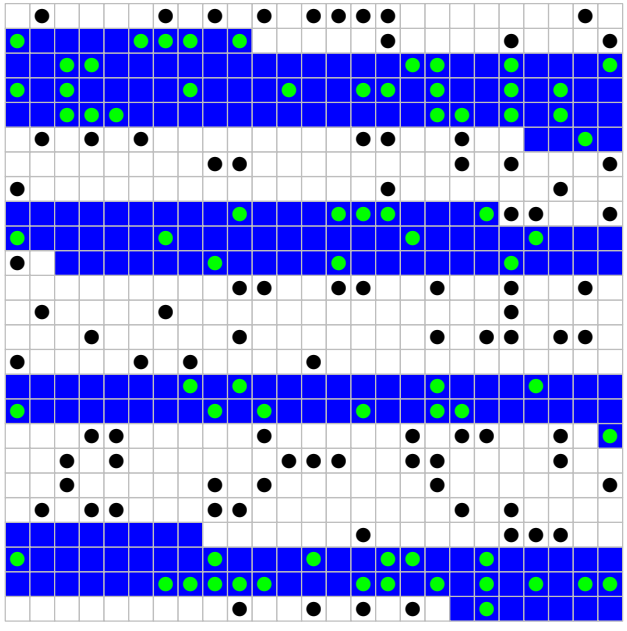
Spatial Pooler



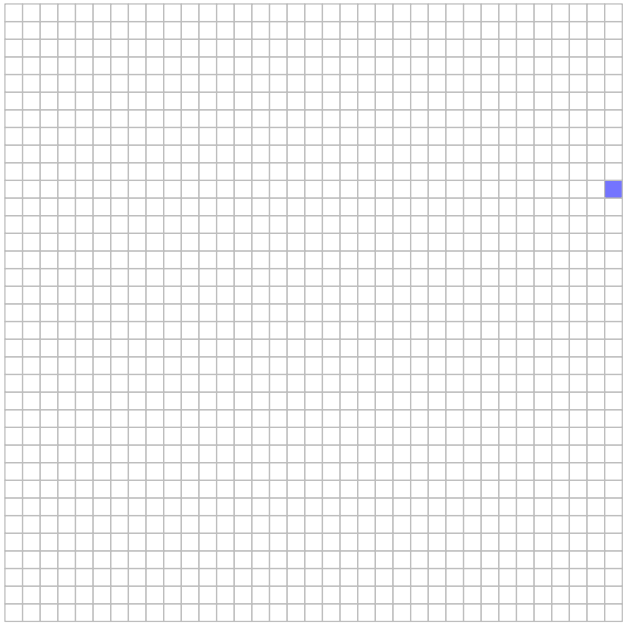
- Only potential connections with a strength above threshold are considered active connections

Active Columns

Input Space (encoder)



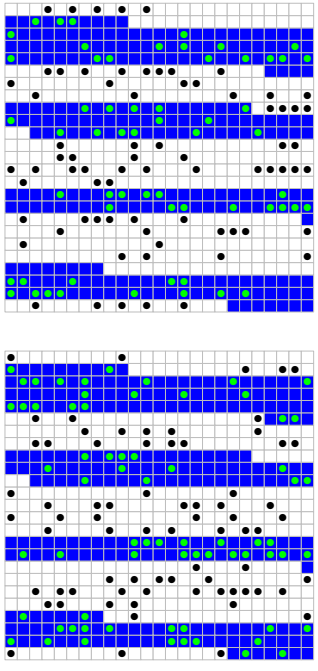
Spatial Pooler



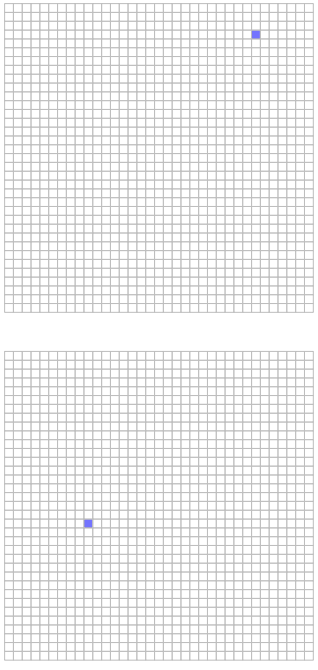
- A column's overlap score is the intersection of SP connections and encoder signal

Active Columns

Input Space (encoder)



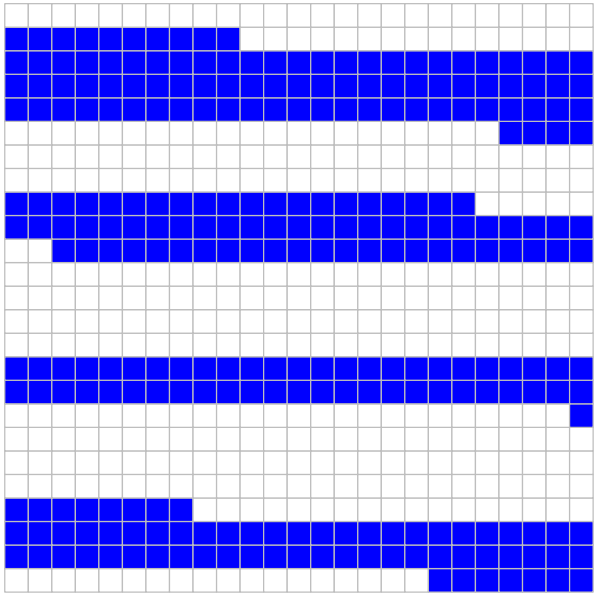
Spatial Pooler



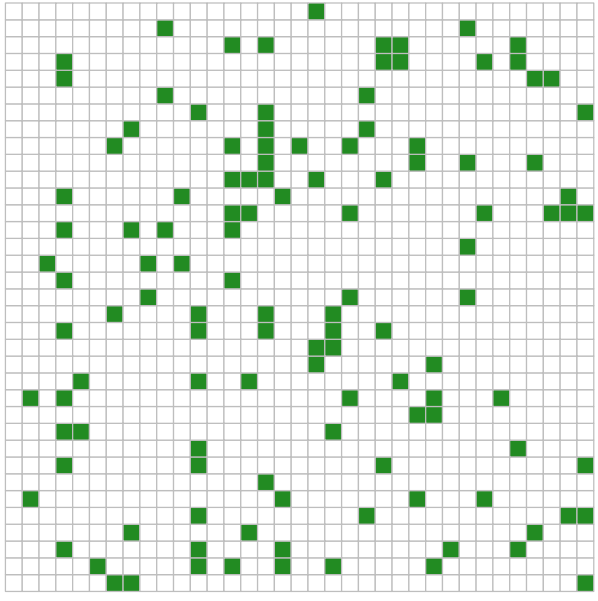
- Each column will have a different intersection and as a result a different overlap score

Active Columns

Input Space (encoder)



Spatial Pooler



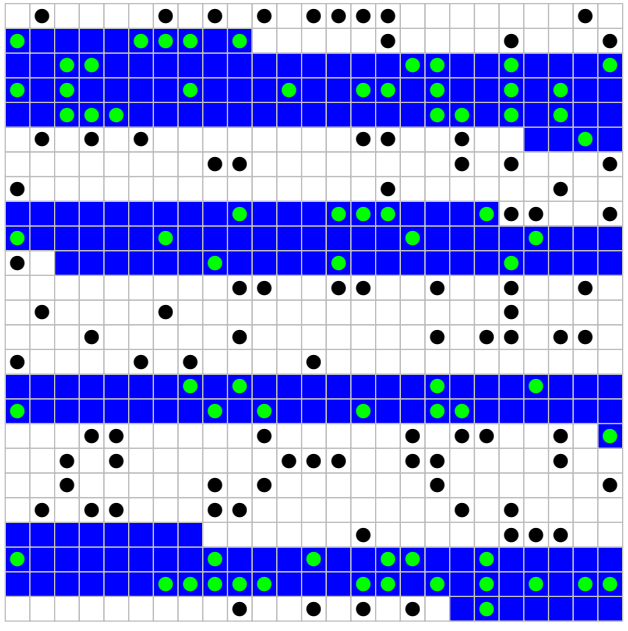
- Columns with greatest overlap score are considered active/winning

Spatial Pooler Design Goals

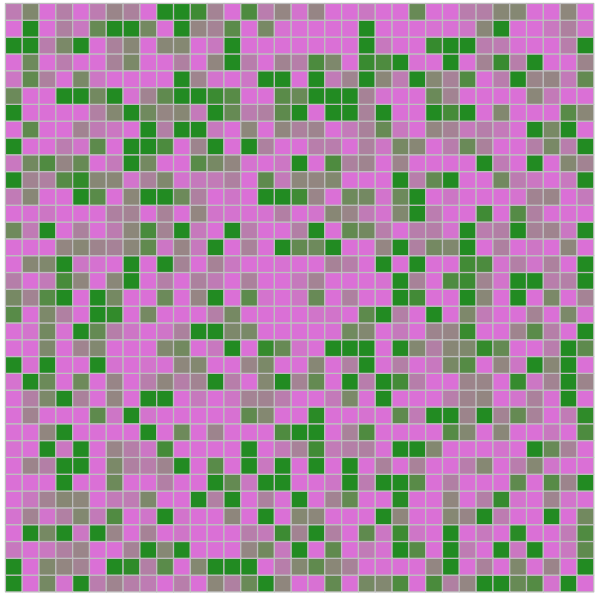
- Maintain SP active column sparsity **by limiting number of winning columns**
- Avoid trivial patterns **via a minimum overlap score**
- All columns should learn to represent something useful
- Avoid extra connections

Boosting

Input Space (encoder)



Spatial Pooler



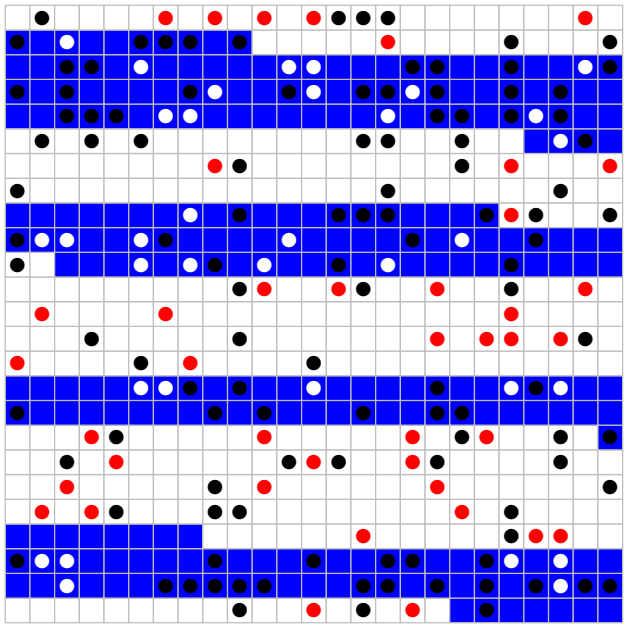
- To prevent a small subset of columns from always winning the overlap scores are boosted before the winners are determined

Spatial Pooler Design Goals

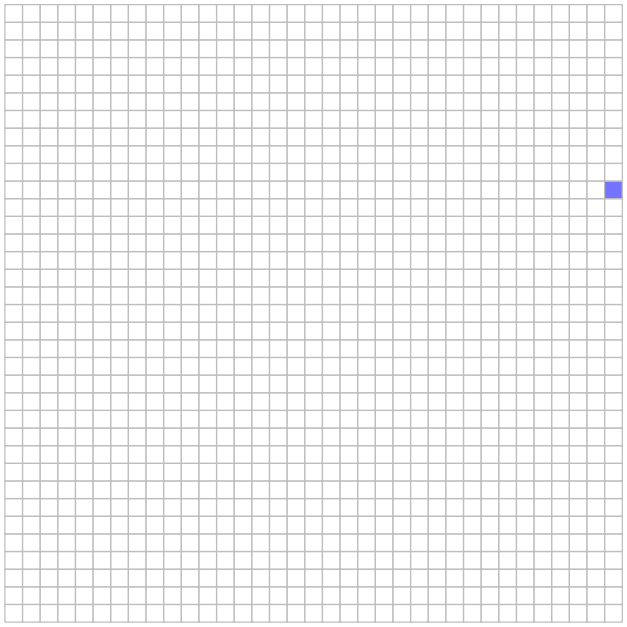
- Maintain SP active column sparsity by limiting number of winning columns
- Avoid trivial patterns via a minimum overlap score
- All columns should learn to represent something useful **by boosting**
- Avoid extra connections

Learning

Input Space (encoder)



Spatial Pooler



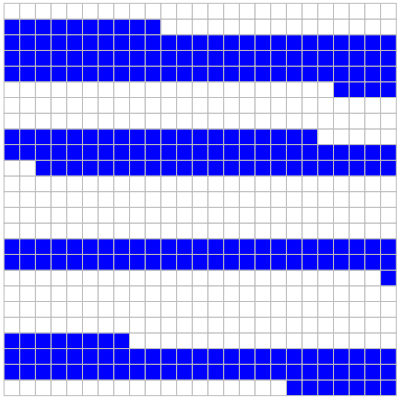
- In the active/winning columns, all connections to active bits are incremented and all connections to inactive bits are decremented. This creates and destroys active connections.

Spatial Pooler Design Goals

- Maintain SP active column sparsity by limiting number of winning columns
- Avoid trivial patterns via a minimum overlap score
- All columns should learn to represent something useful by boosting
- Avoid extra connections **by decrementing non active connections of winning columns**

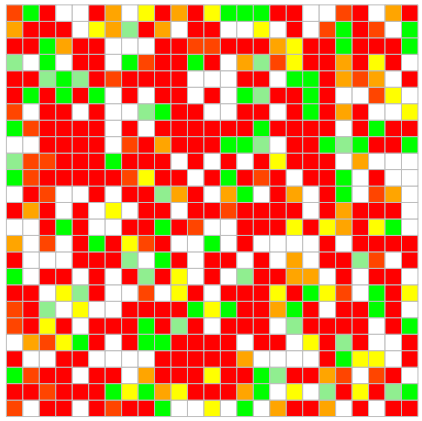
Spatial Pooler Algorithm

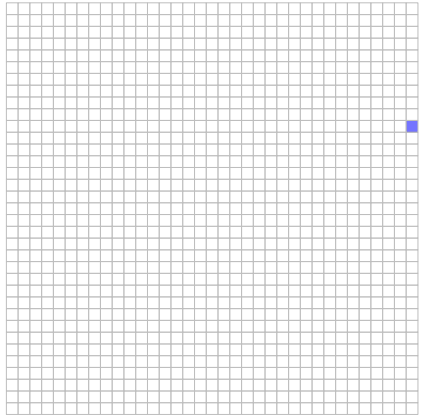
1. Start with an input consisting of a fixed number of bits



Spatial Pooler Algorithm

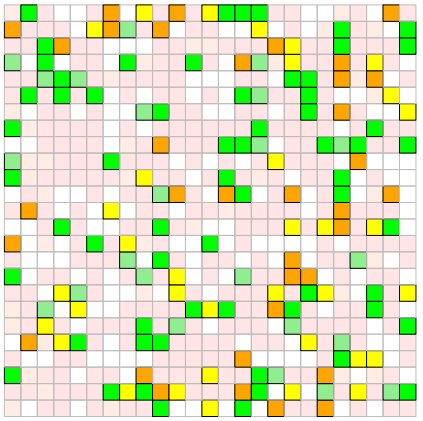
1. Start with an input consisting of a fixed number of bits
2. Assign a fixed number of columns to the input bits, wherein each column has potential connections to a subset of bits.

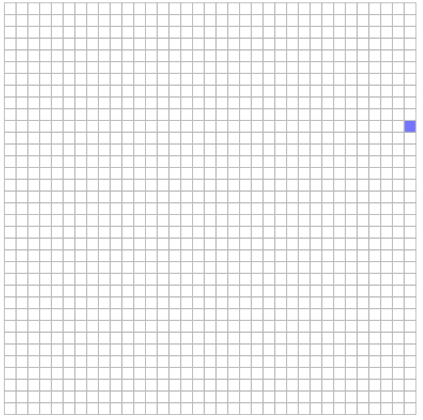




Spatial Pooler Algorithm

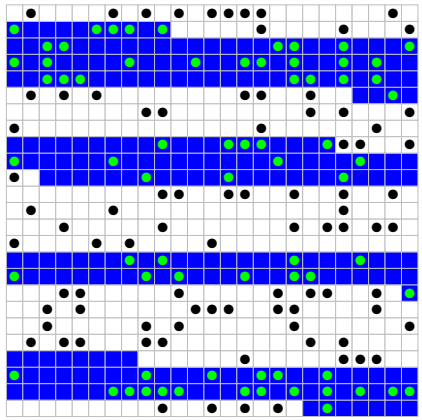
1. Start with an input consisting of a fixed number of bits
2. Assign columns to the input bits
3. Some of the potential connections will become active based on their permanence values

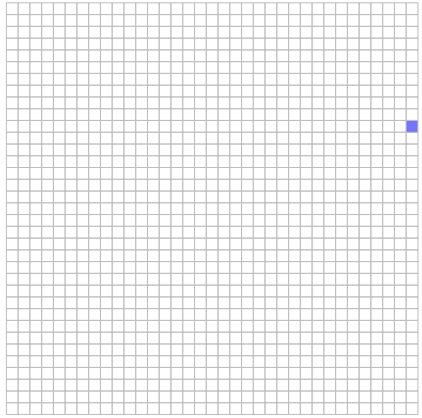




Spatial Pooler Algorithm

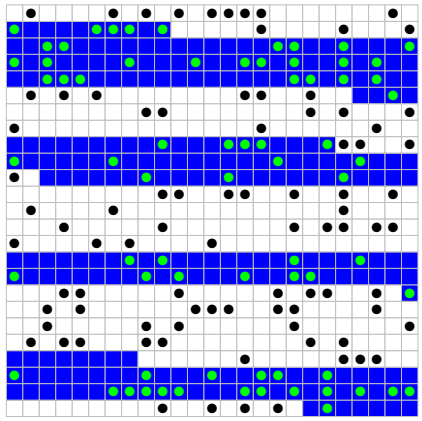
1. Start with an input consisting of a fixed number of bits
2. Assign columns to the input bits
3. Potential columns become active based on permanences
4. Determine the number of active connections for each column connected to active input bits

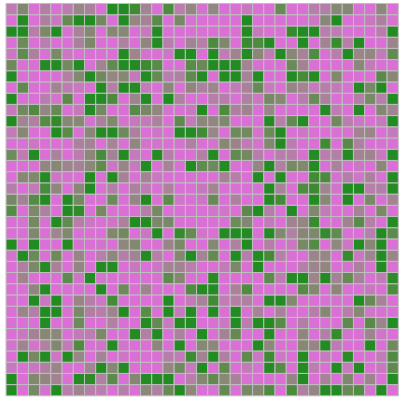




Spatial Pooler Algorithm

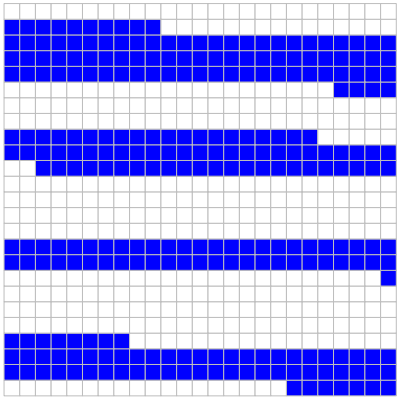
1. Start with an input consisting of a fixed number of bits
2. Assign columns to the input bits
3. Potential columns become active based on permanences
4. Number of active connections to active input bits
5. The number of active connections for each column is modified by a "boosting" factor determined by how often it is active relative to its neighbors

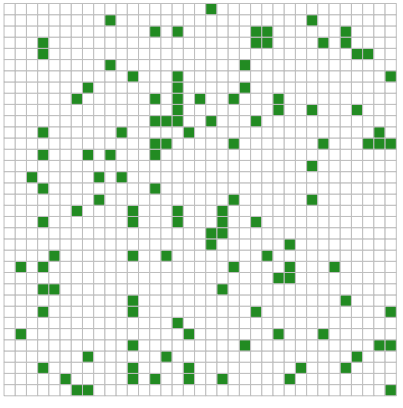




Spatial Pooler Algorithm

1. Start with an input consisting of a fixed number of bits
2. Assign columns to the input bits
3. Potential columns become active based on permanences
4. Number of active connections to active input bits
5. Boost inactive and too active columns
6. A fixed number of columns with the highest overlaps become active and the rest of the columns are disabled, maintaining sparsity

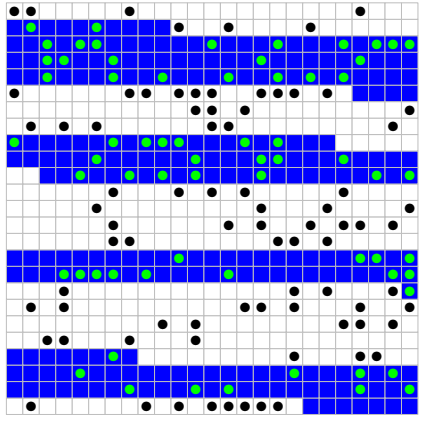




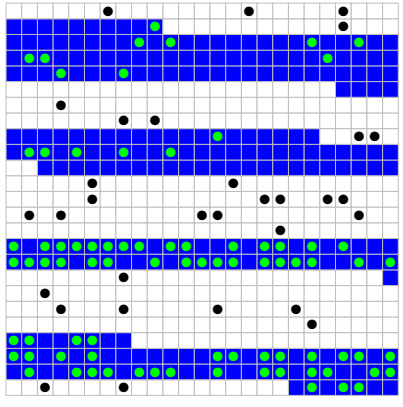
Spatial Pooler Algorithm

1. Start with an input consisting of a fixed number of bits
2. Assign columns to the input bits
3. Potential columns become active based on permanences
4. Number of active connections to active input bits
5. Boost inactive and too active columns
6. Columns with highest overlap become active
7. For each active column, the permanence values of connections to active bits are incremented and those to non-active bits are decremented. This may result in new active connections and loss or previously active connections

Start



End

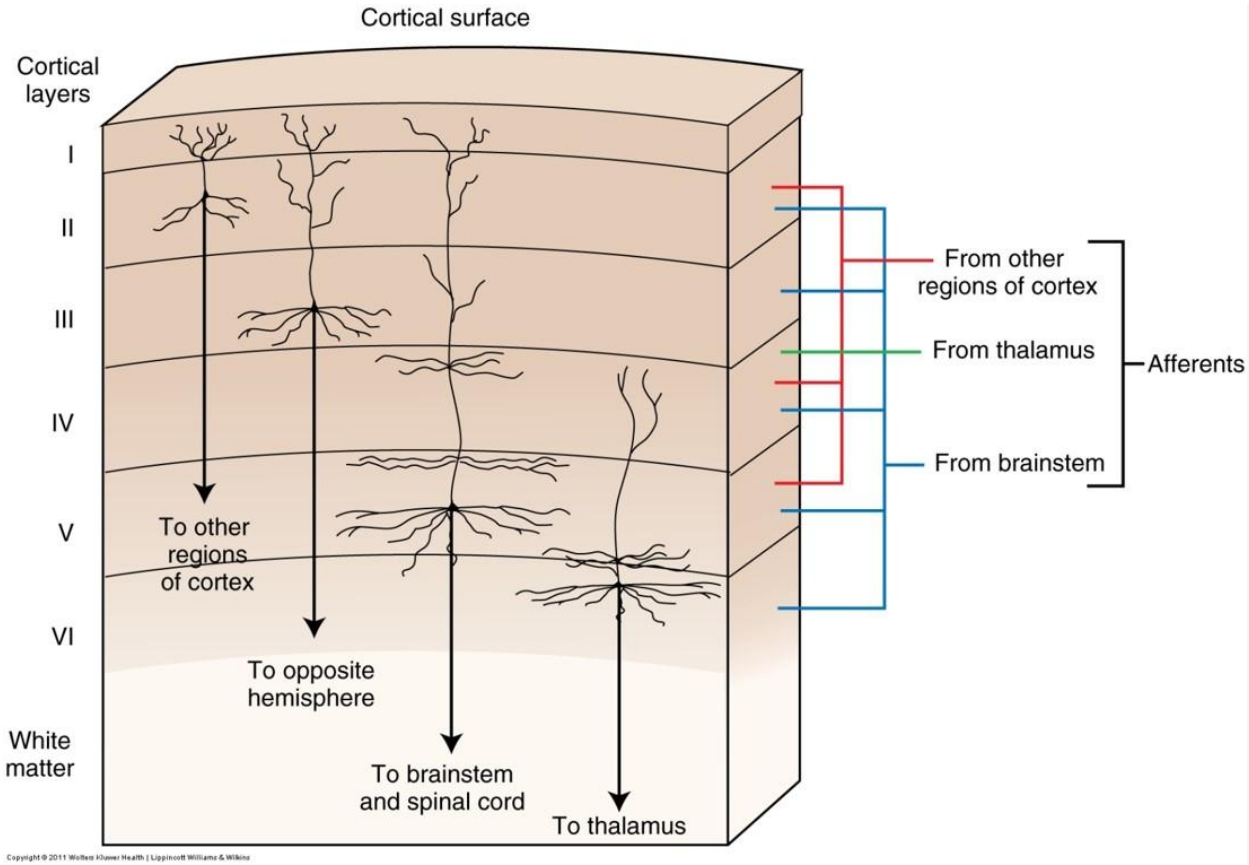


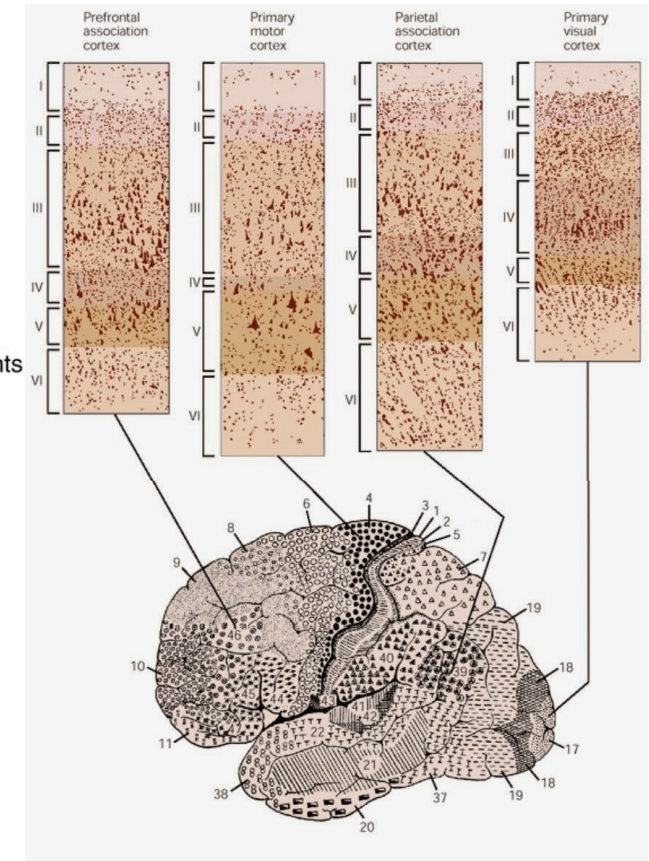
Spatial Pooler Parameters

Local Area Density	Controls the density (sparsity) for any SP area
Stimulus Threshold	Minimum overlap score for a column to become active
Permanence Increment	Amount a connection to an active bit is incremented when connected to a winning column
Permanence Decrement	Amount a connection to a non-active bit is decremented when connected to a winning column
Overlap Duty Cycle	How often to check a column should have at least **Stimulus Threshold** active inputs
Boost Strength	Strength of boosting

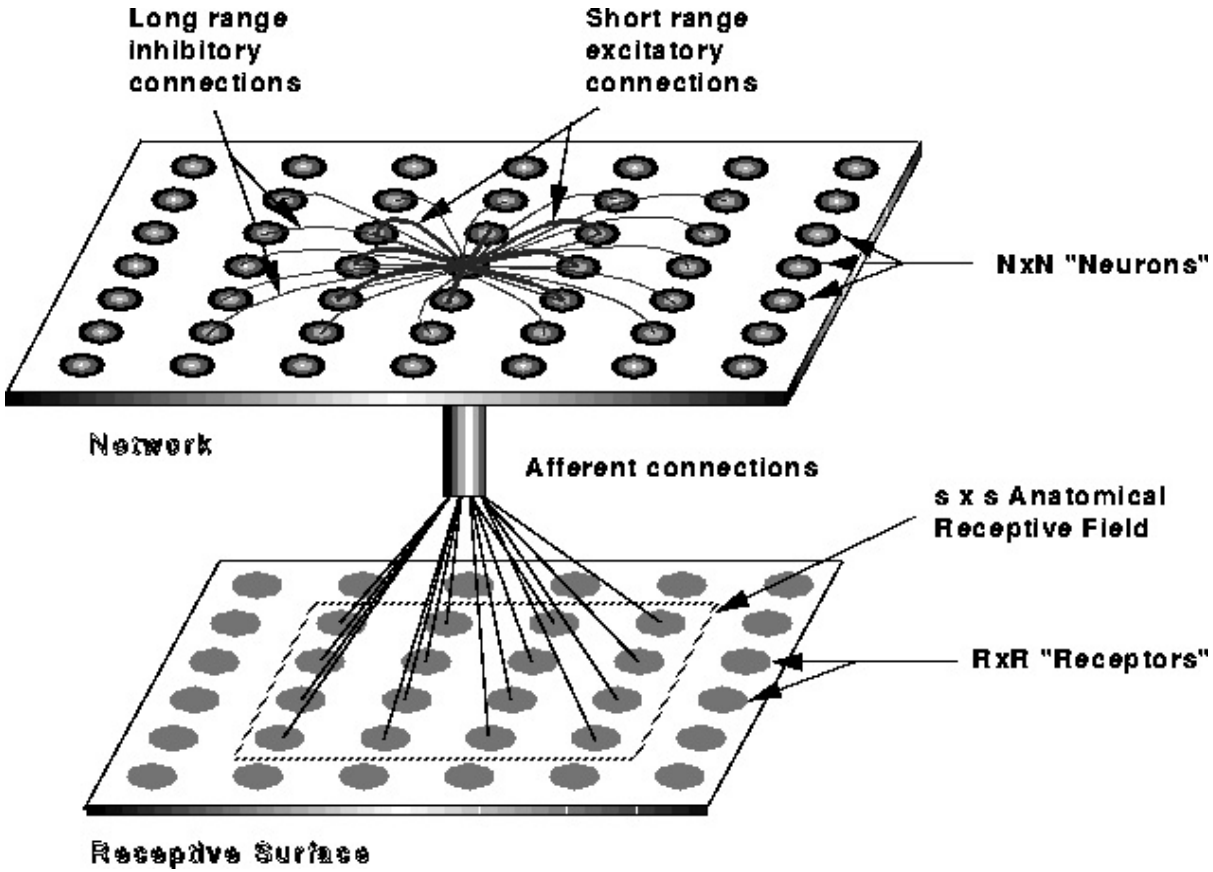
Questions?

Cortical Layers





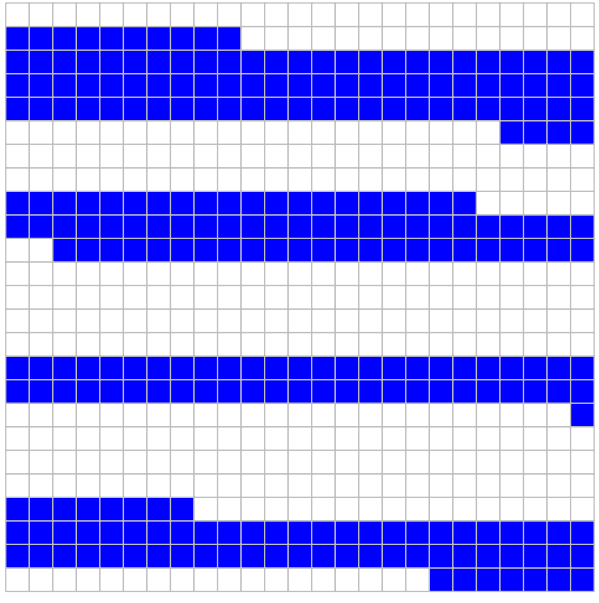
Cortical Connections



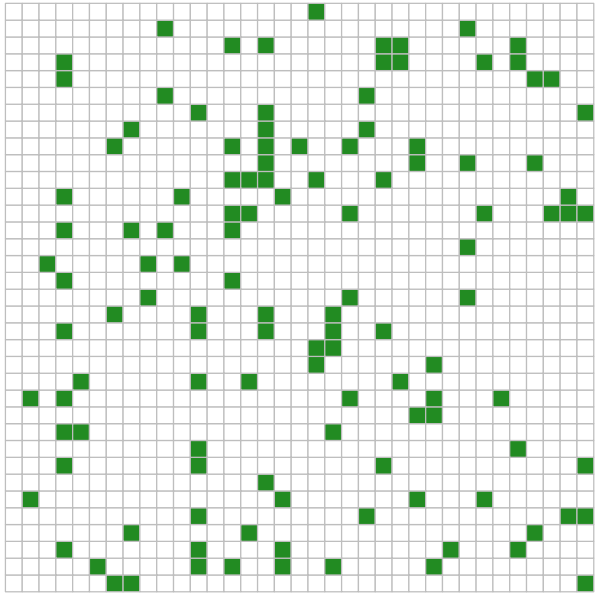
- The HTM equivalent for receptive surface is the input (encoder), the afferent connection to the network is the spatial pooler and the lateral connections are the temporal memory.

Temporal Memory Structure

Input Space (encoder)



Spatial Pooler



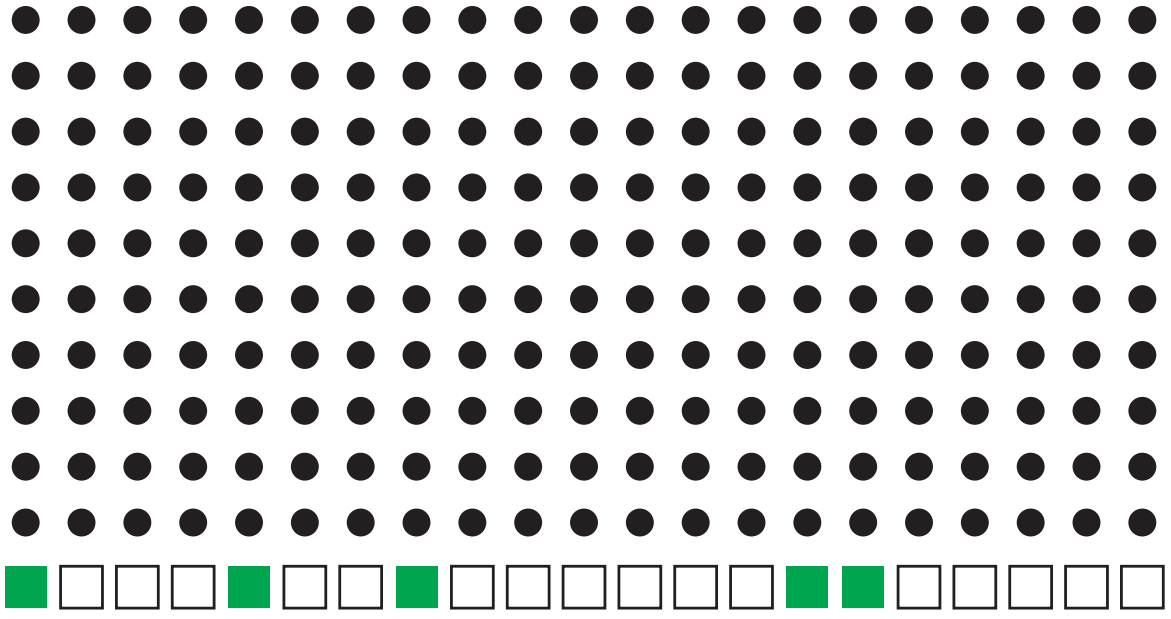
- These are two dimensional representations of vectors

Temporal Memory Structure

Input Space (encoder)

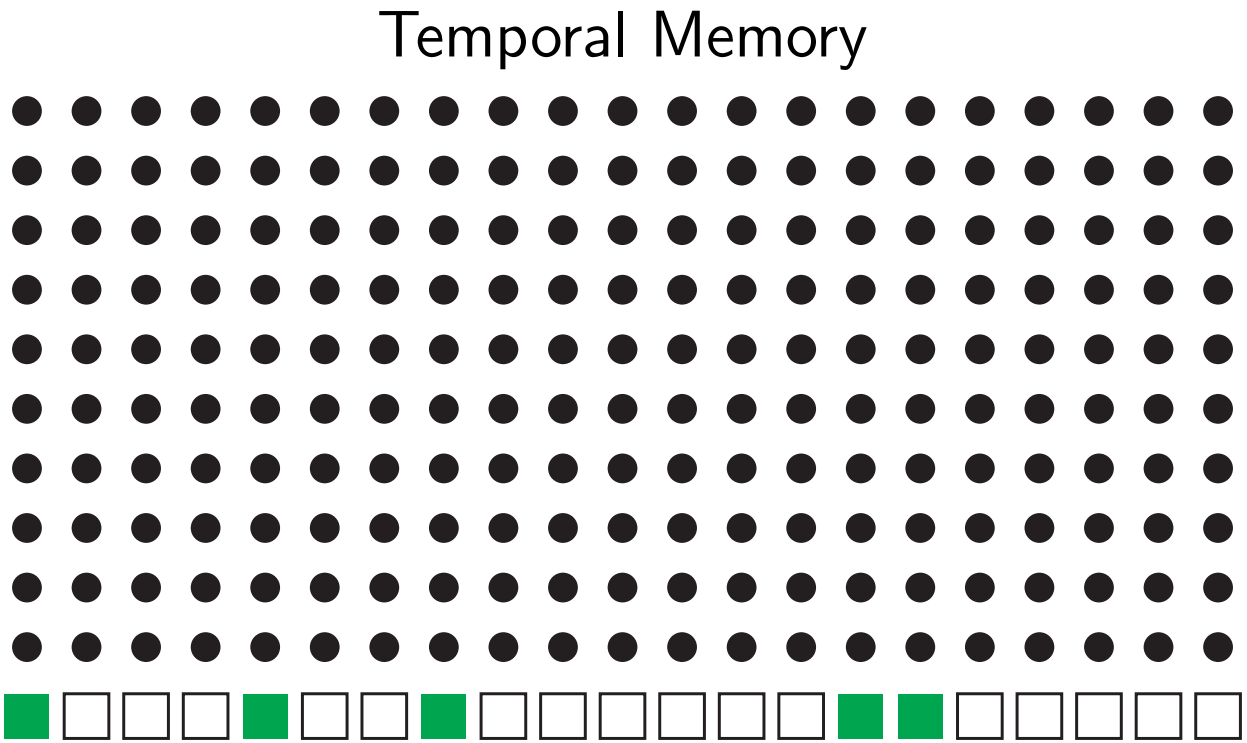
EGA	β-**hCG**
3.0	40 mIU/mL

Temporal Memory



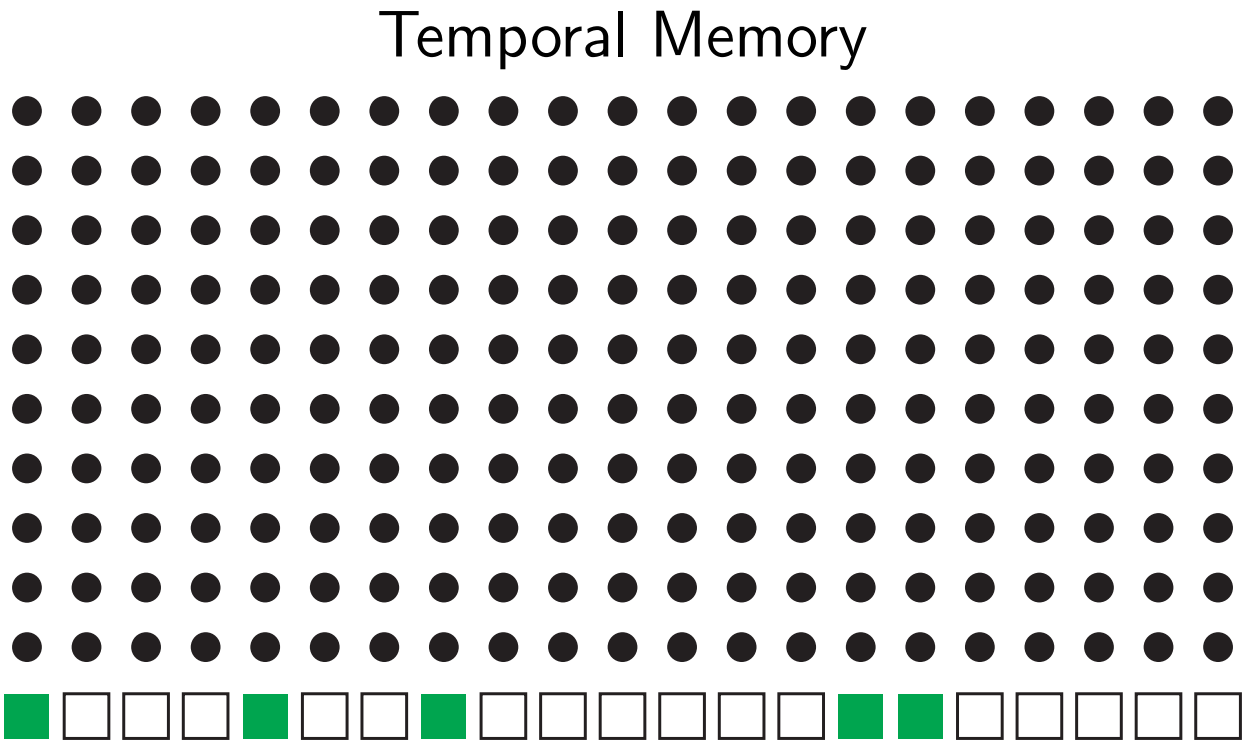
- Each active column is made up of a number of cells

Temporal Memory Phases



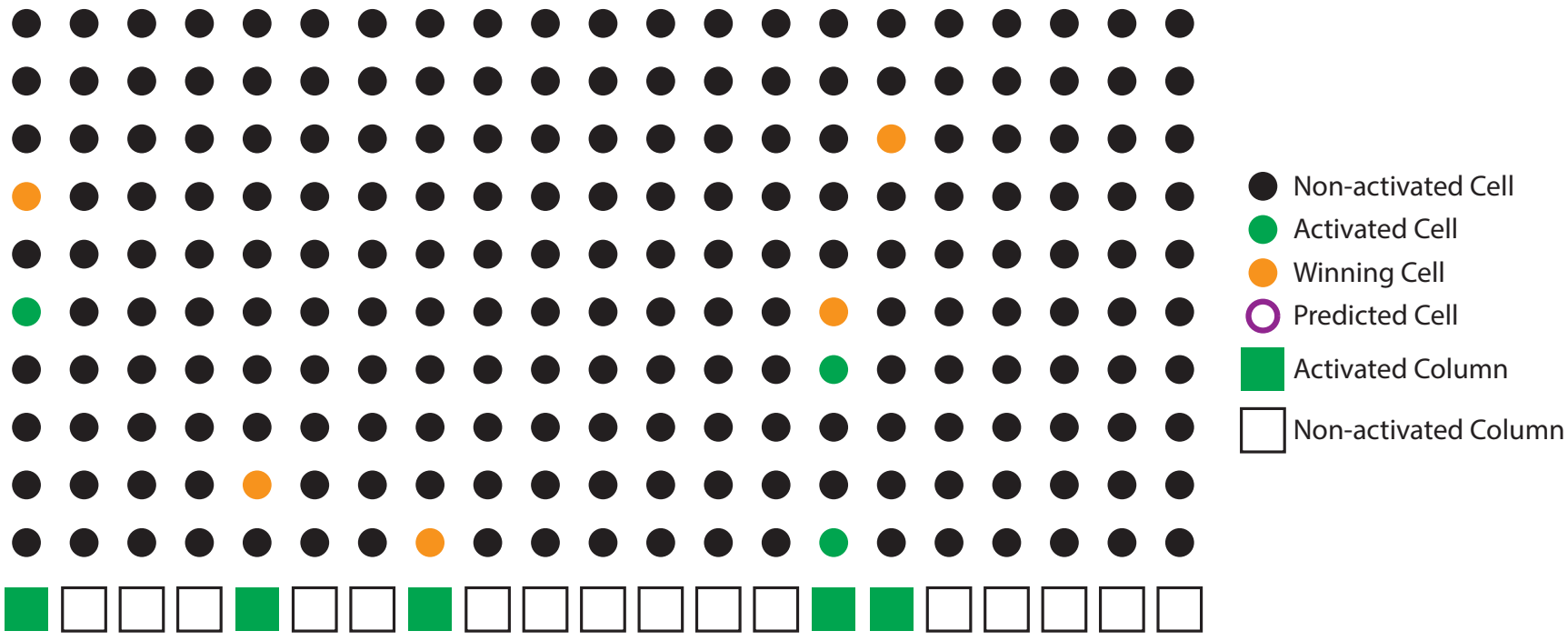
1. Active state for each cell
2. Predictive state for each cell
3. Update cell connections

Temporal Memory Intuition



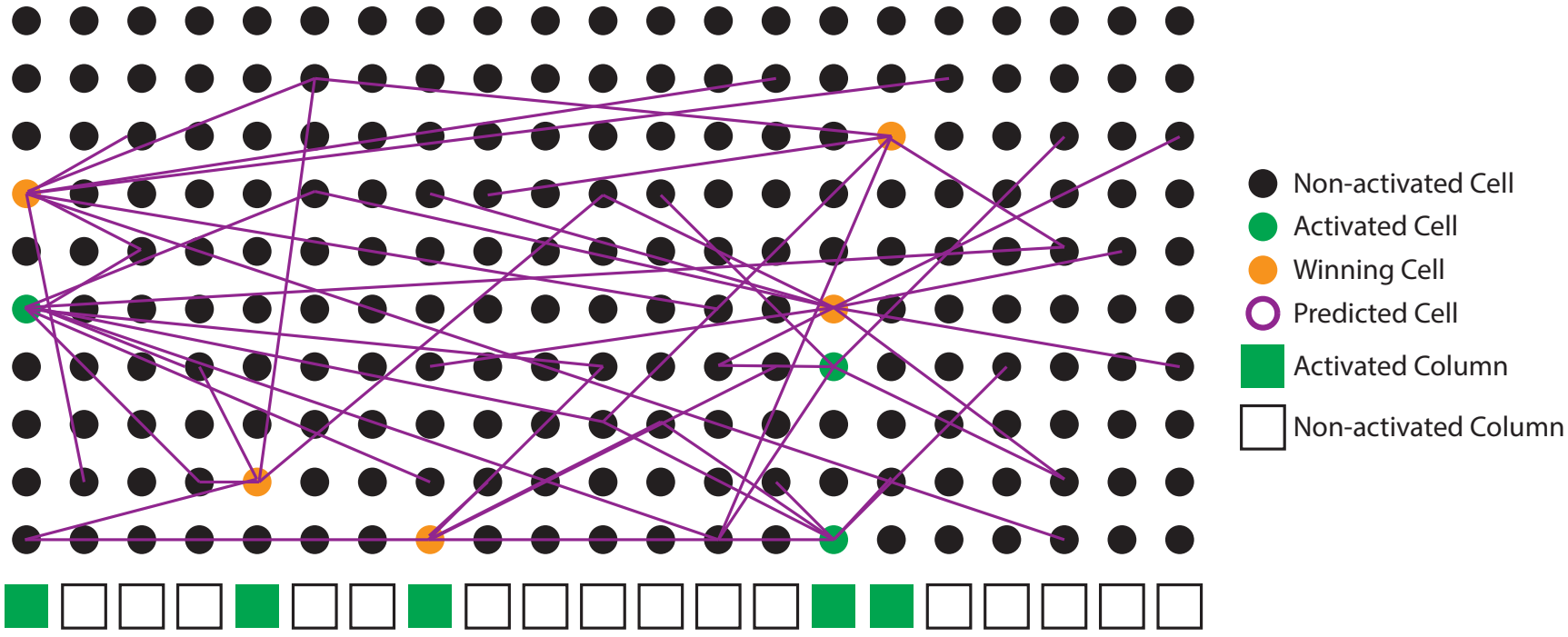
1. Predictive state for each cell
2. Active state for each cell
3. Update cell connections

Temporal Memory Phases



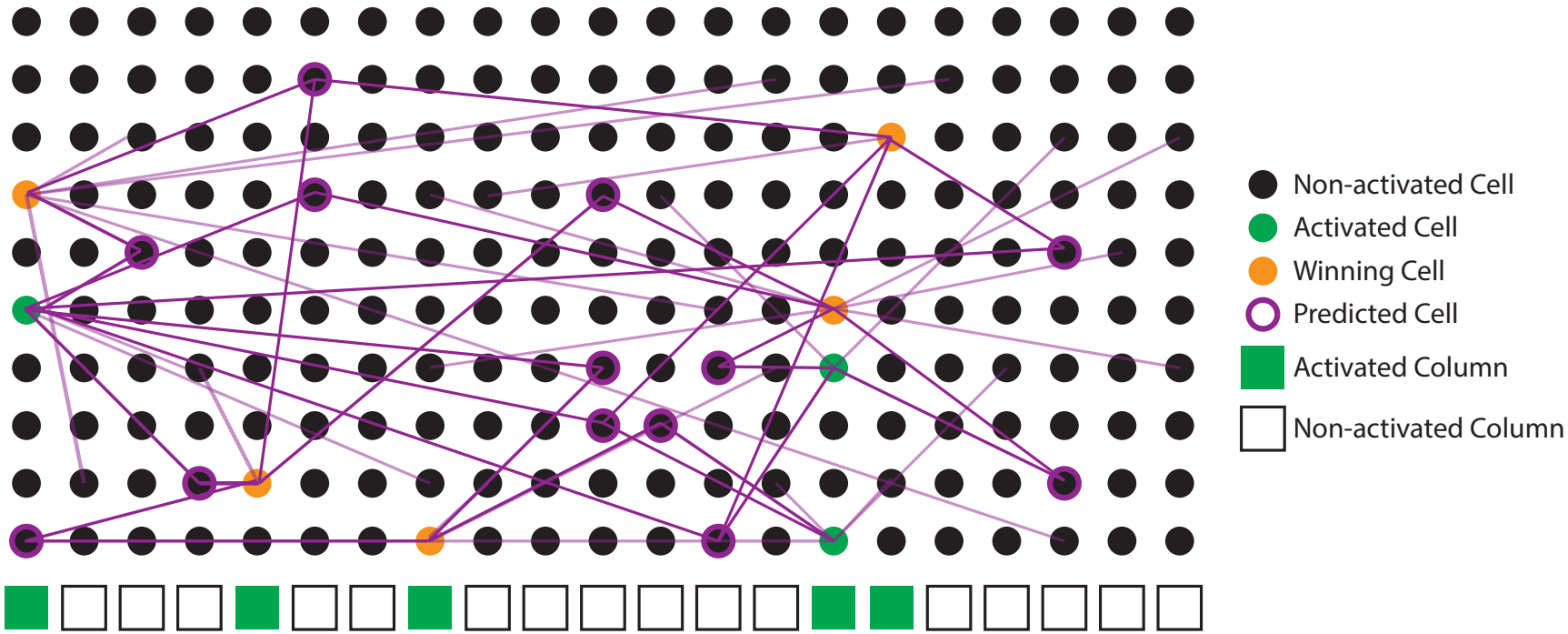
- Predictive state for each cell

Temporal Memory Phases



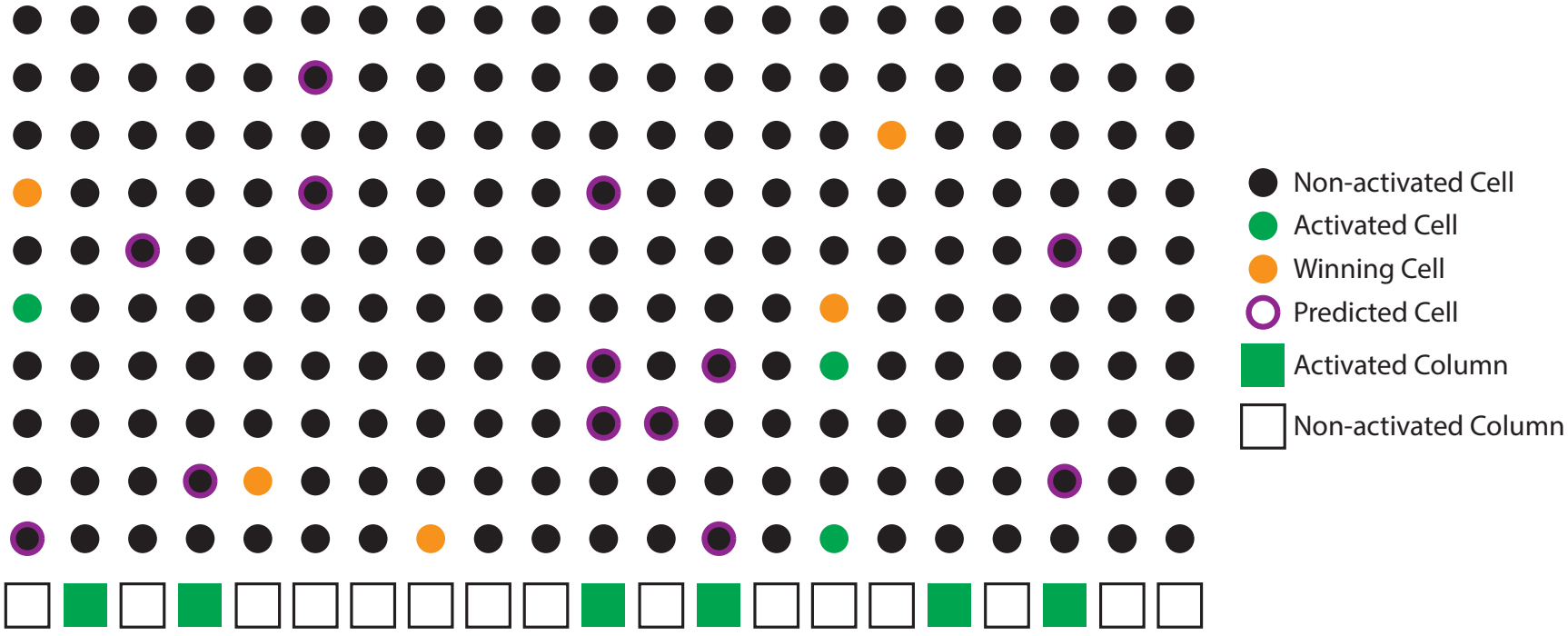
- Predictive state for each cell
  - Trace the predictive connections for all active cells in the current time step

Temporal Memory Phases



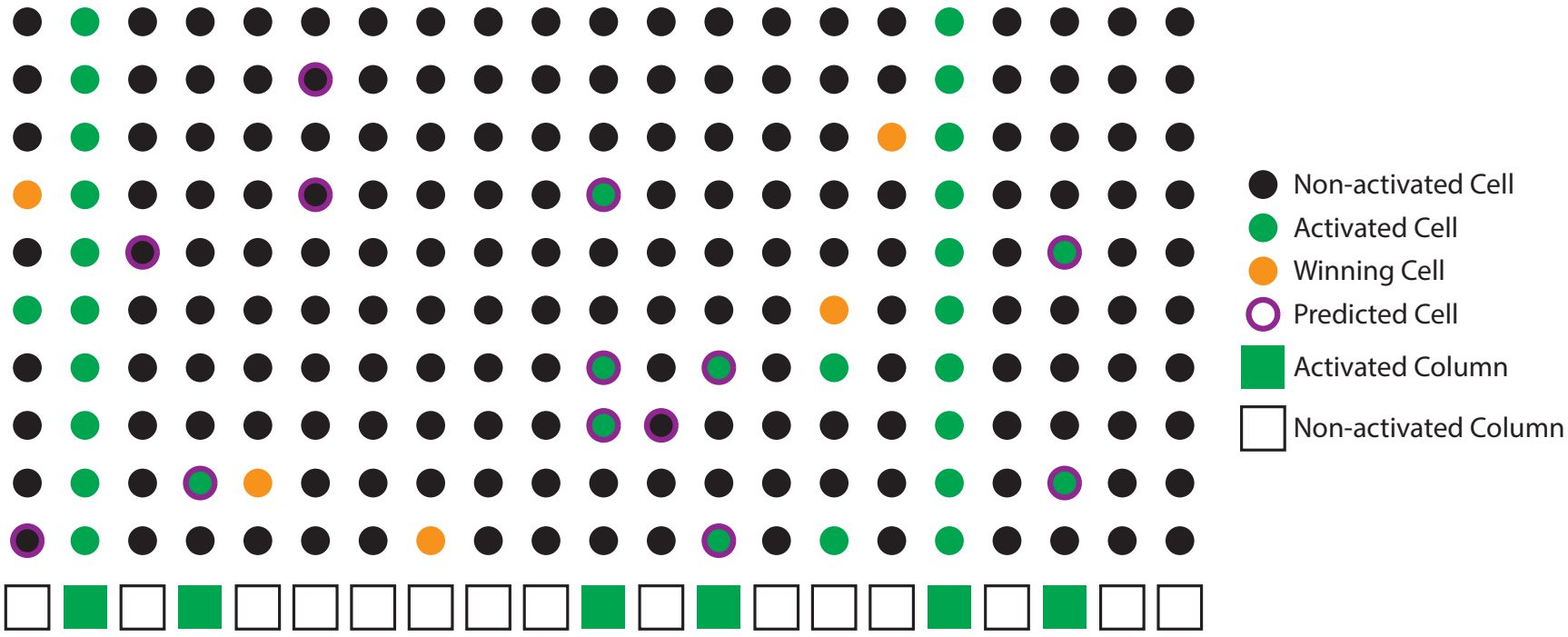
- Predictive state for each cell
  - Trace the predictive connections for all active cells in the current time step
  - Cells with prediction values above threshold are predicted for the next time step

Temporal Memory Phases



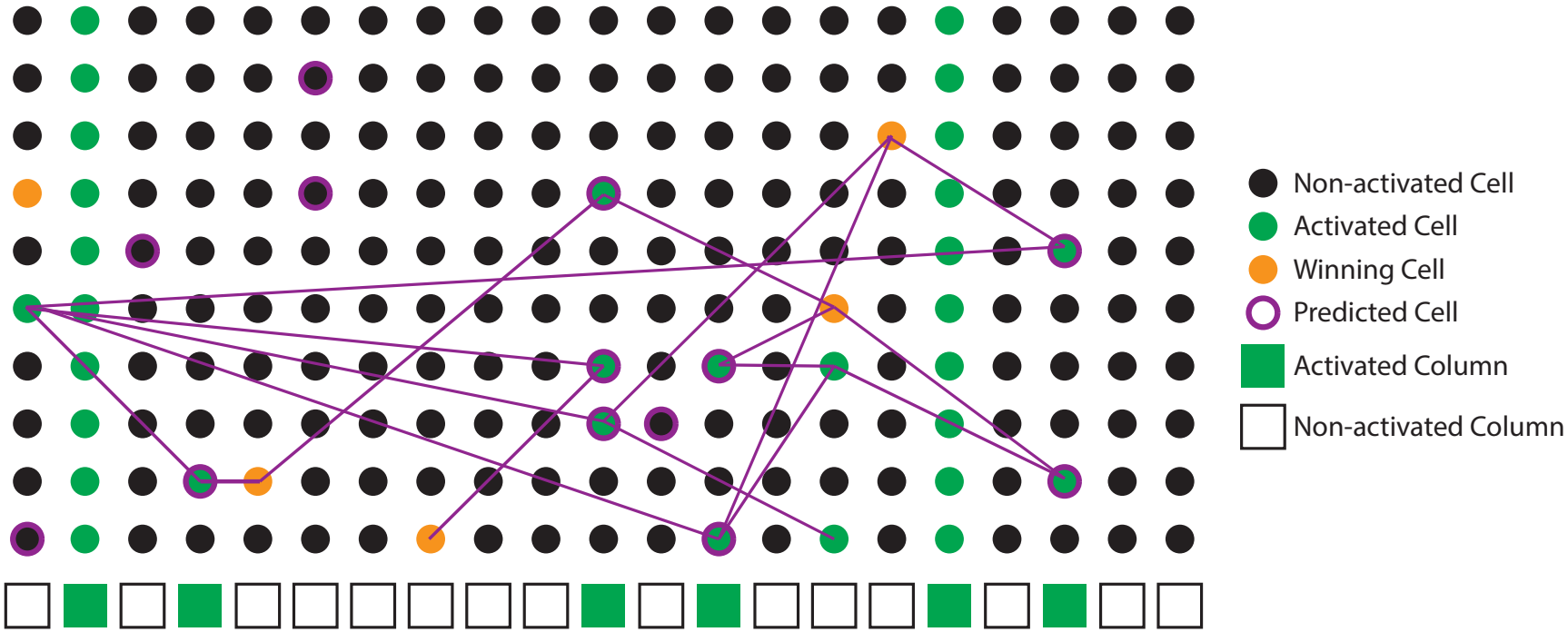
- Active state for each cell
  - Activate cells
    - Determine the winning columns from the Spatial Pooler

Temporal Memory Phases



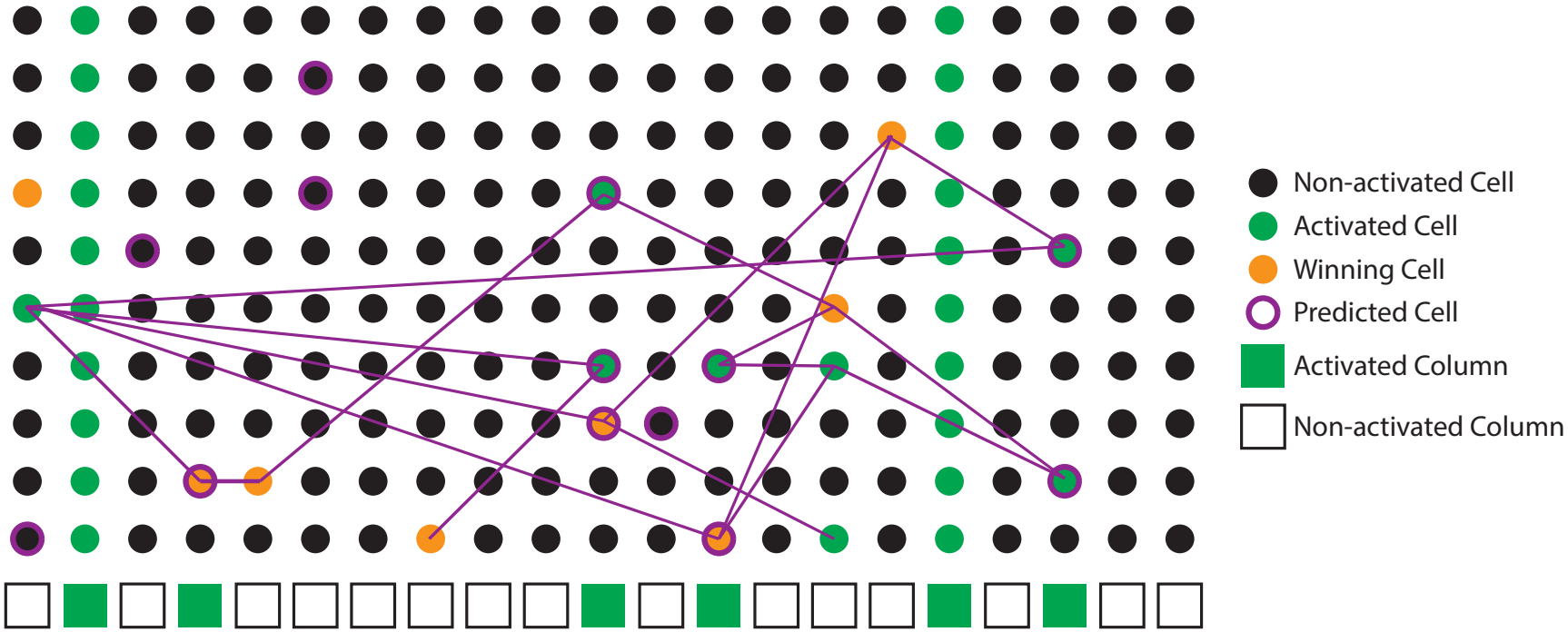
- Active state for each cell
  - Activate cells
    - Determine the winning columns from the Spatial Pooler
    - Activate all predicted cells in the activated columns
    - If no cells are predicted, activate all cells (Burst)

Temporal Memory Phases



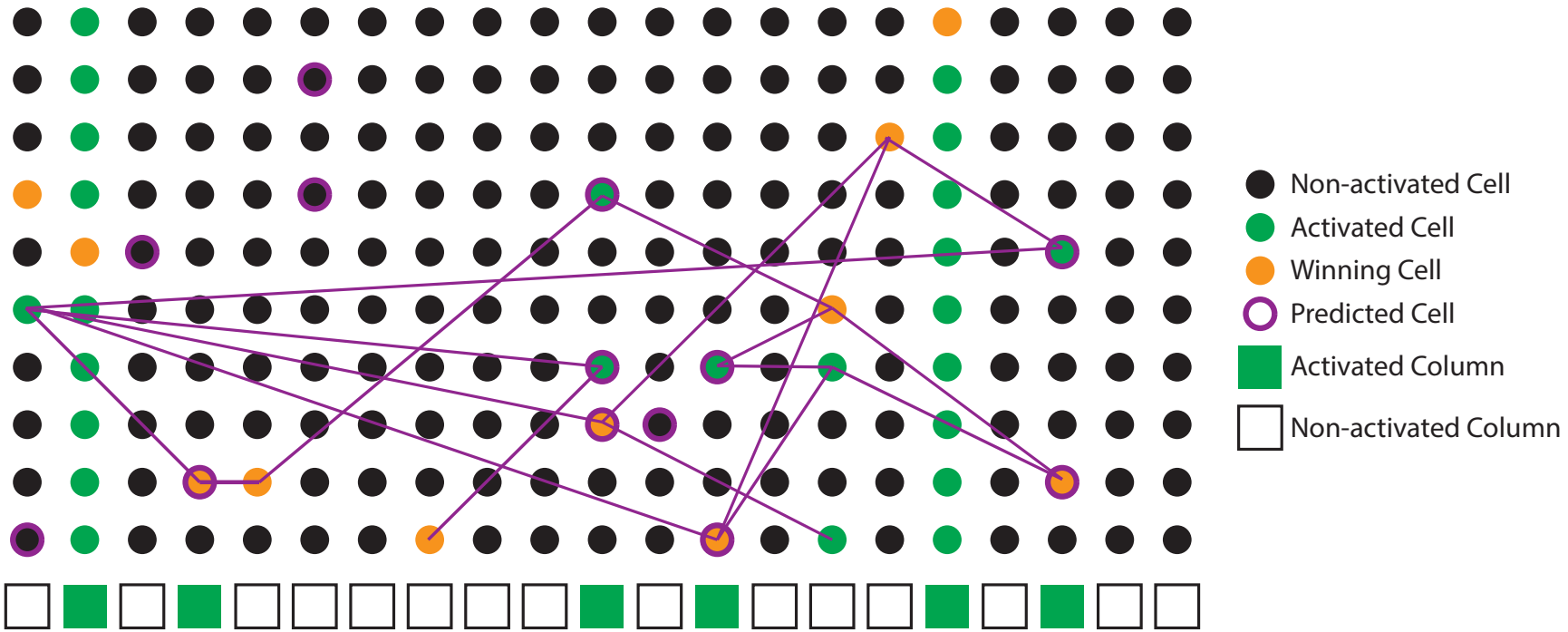
- Active state for each cell
  - Activate cells
  - Winning cells
    - Look at the number of predictions per activated cell

Temporal Memory Phases



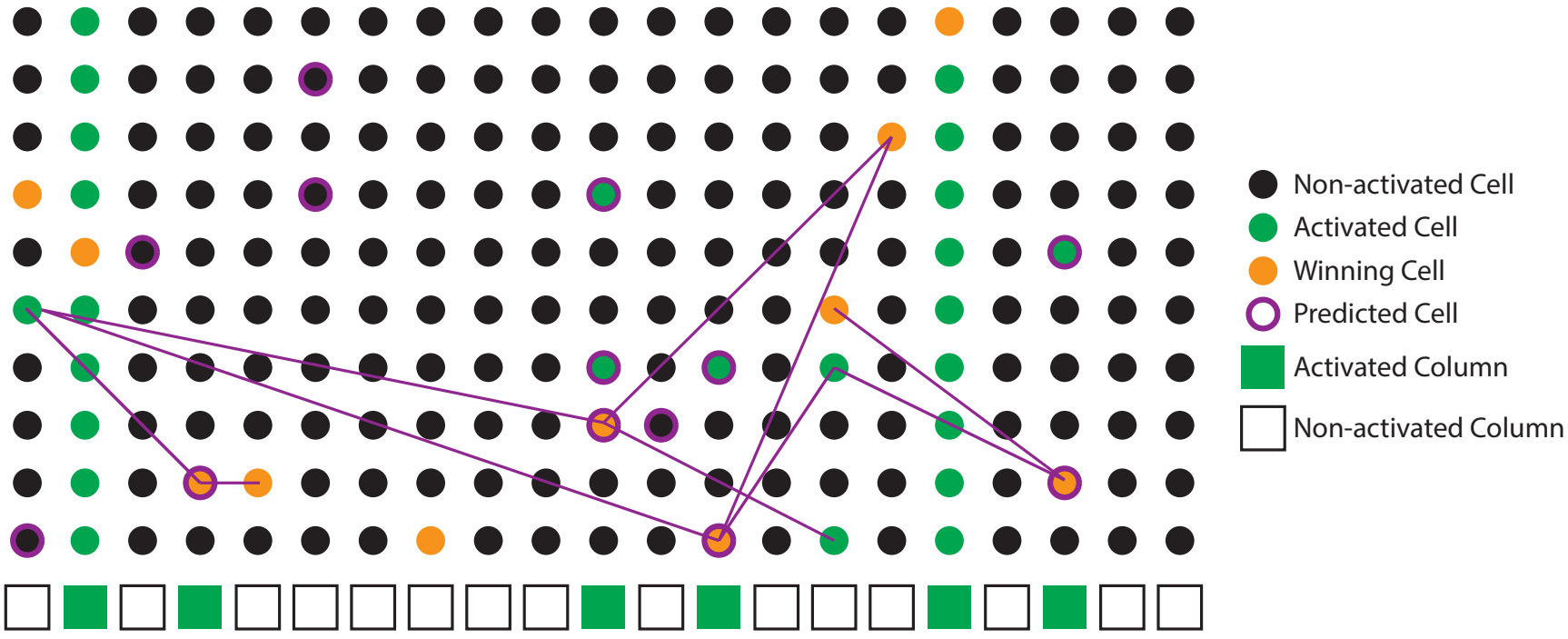
- Active state for each cell
  - Activate cells
  - Winning cells
    - Look at the number of predictions per activated cell
    - Choose the cell, per column, with the most predictions

Temporal Memory Phases



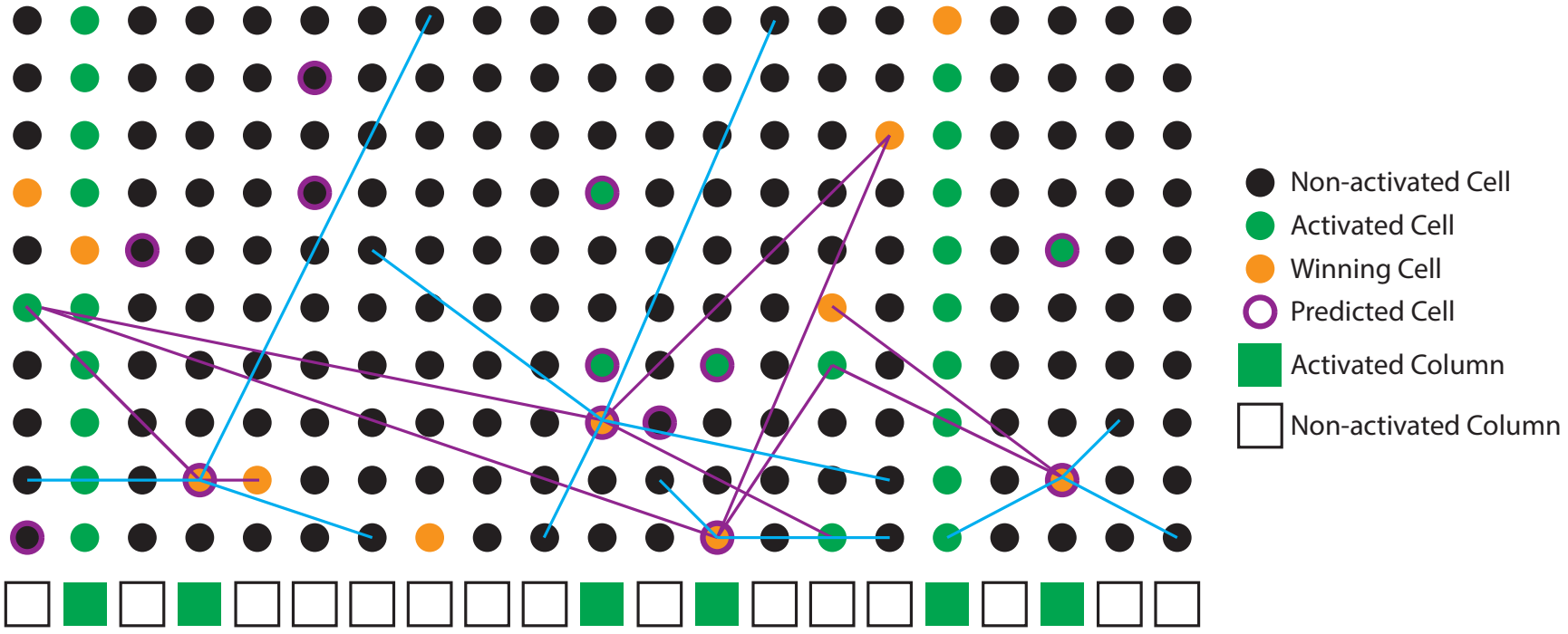
- Active state for each cell
  - Activate cells
  - Winning cells
    - Look at the number of predictions per activated cell
    - Choose the cell, per column, with the most predictions
    - If there are multiple with same number, choose one at random

Temporal Memory Phases



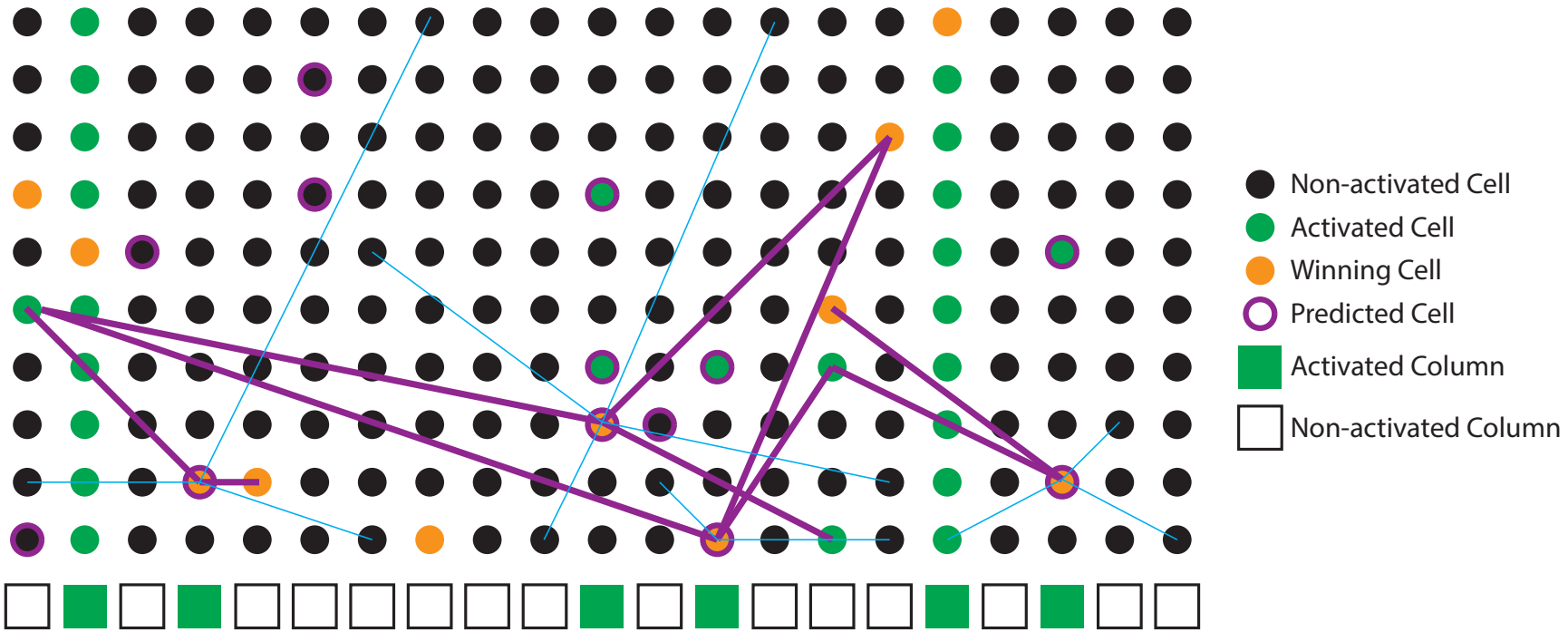
- Update cell connections
  - Non-winning cells
    - Nothing happens to their connections (like in the SP)

Temporal Memory Phases



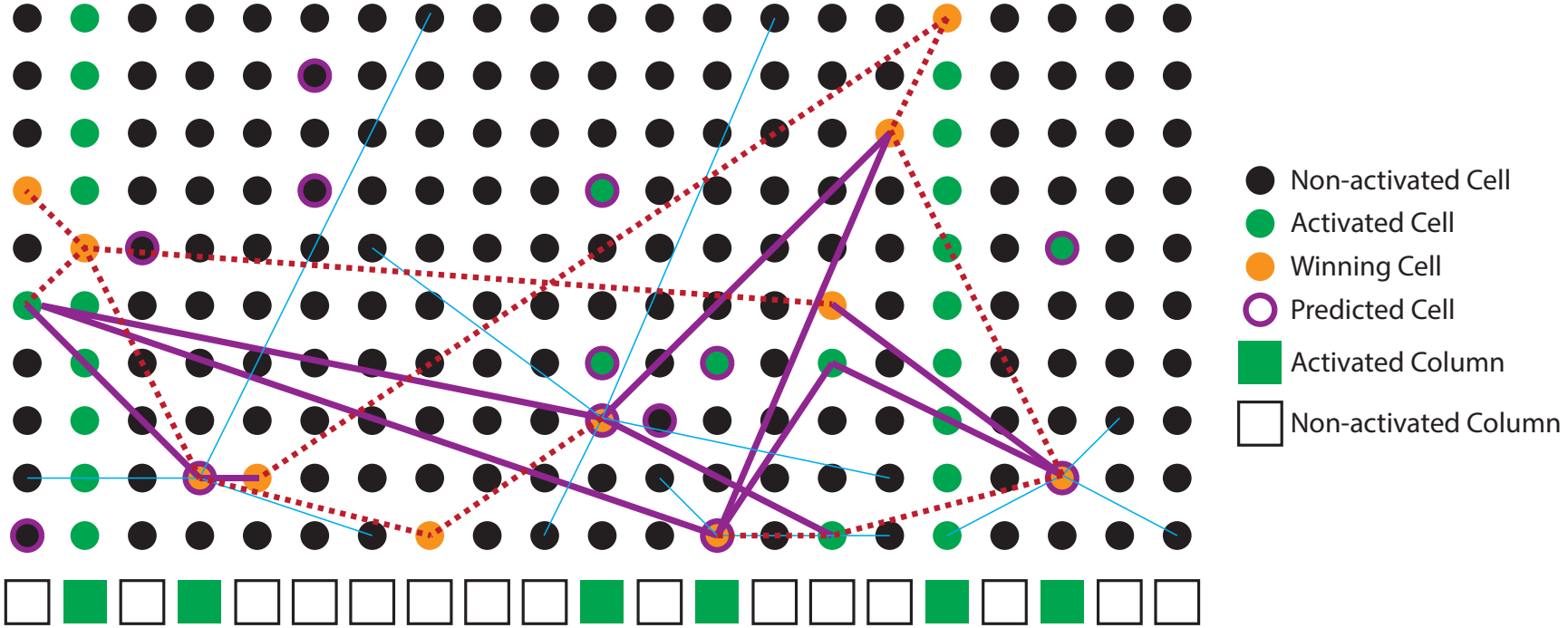
- Update cell connections
  - Non-winning cells
  - Predicted winning cells
    - Look at all their connections

Temporal Memory Phases



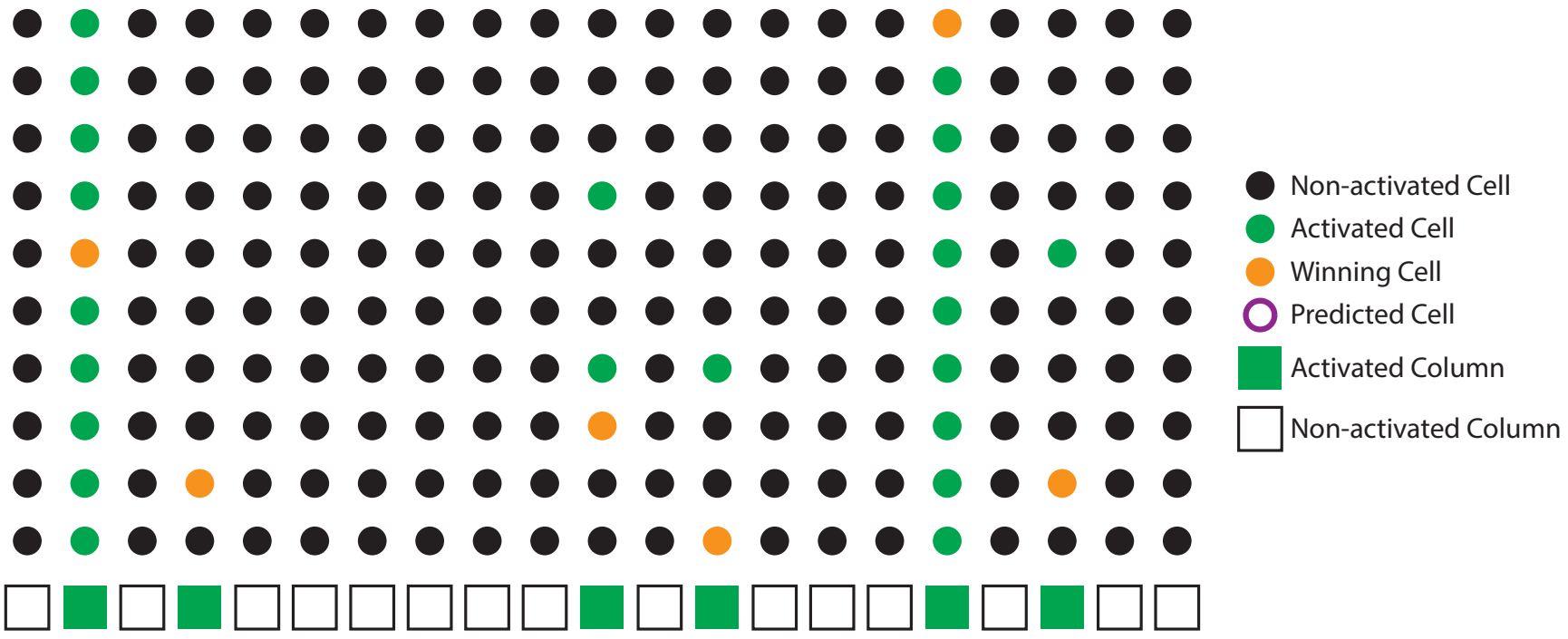
- Update cell connections
  - Non-winning cells
  - Predicted winning cells
    - Look at all their connections
    - Strengthen their connections to cells active in prior step
    - Weaken their connections to cells not active in prior step

Temporal Memory Phases



- Update cell connections
  - Non-winning cells
  - Predicted winning cells
  - All winning cells
    - Connect to a small subset of cells active in the prior time step

Temporal Memory Phases



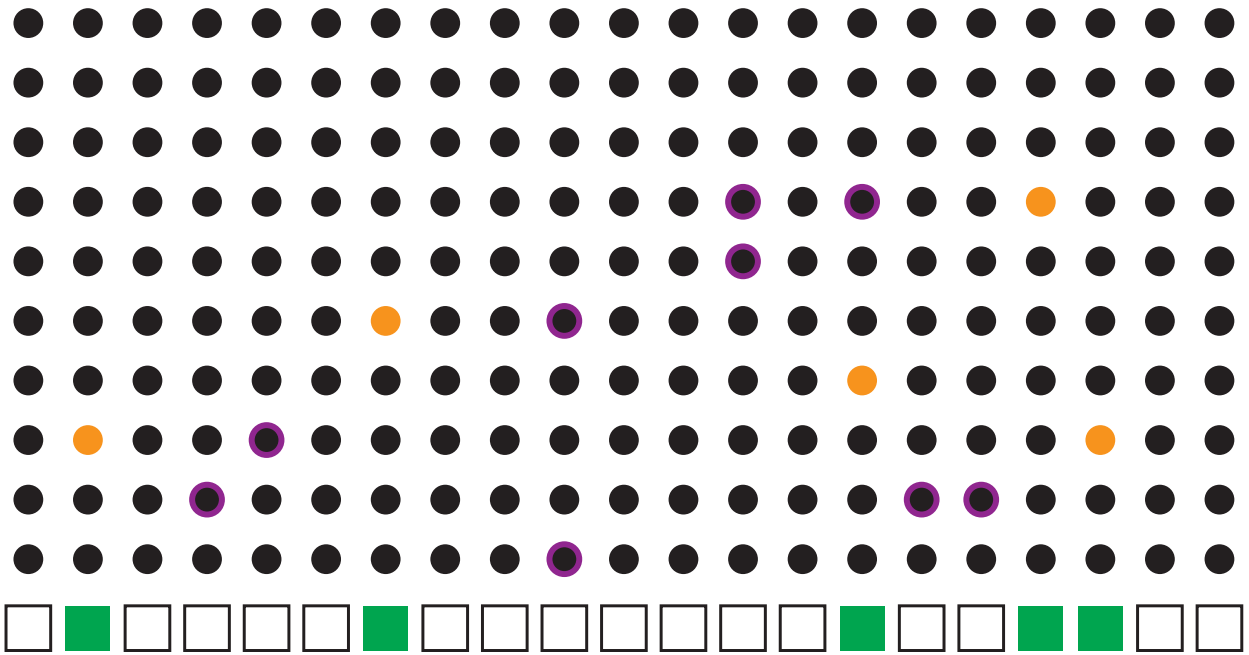
- Predictive state for each cell

Temporal Memory Example

Input Space

EGA	β-**hCG**
3.0	40 mIU/mL

Temporal Memory

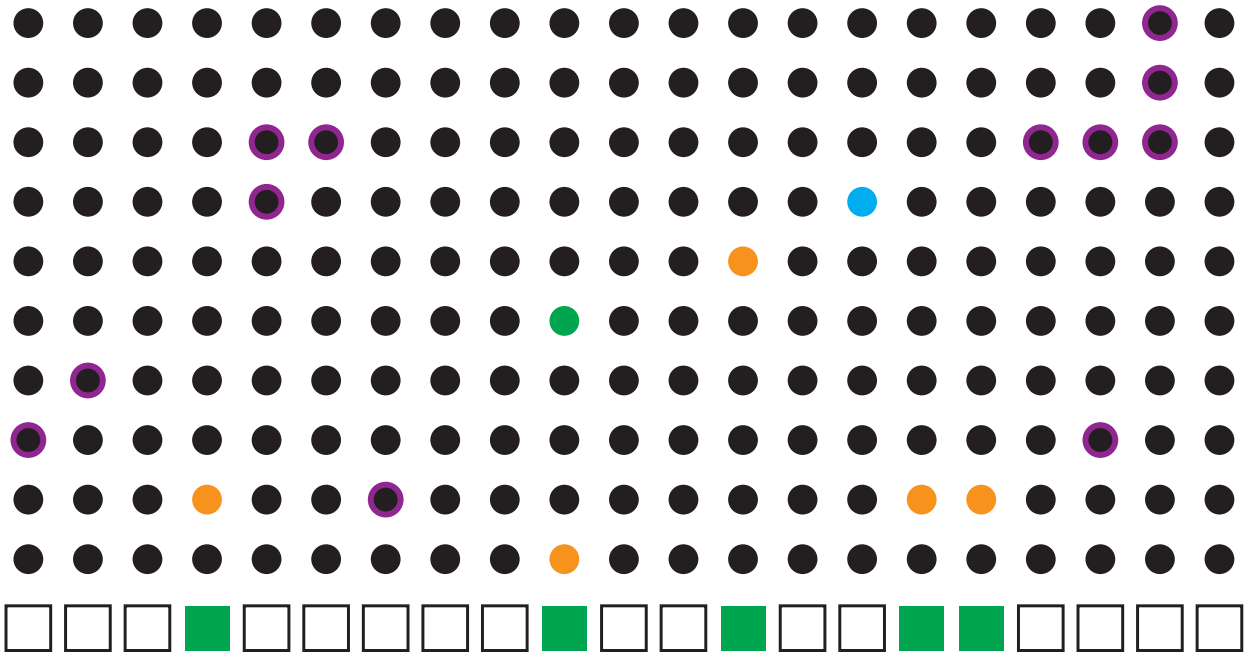


Temporal Memory Example

Input Space

EGA	β-hCG
3.0	40 mIU/mL
3.2	76 mIU/mL

Temporal Memory

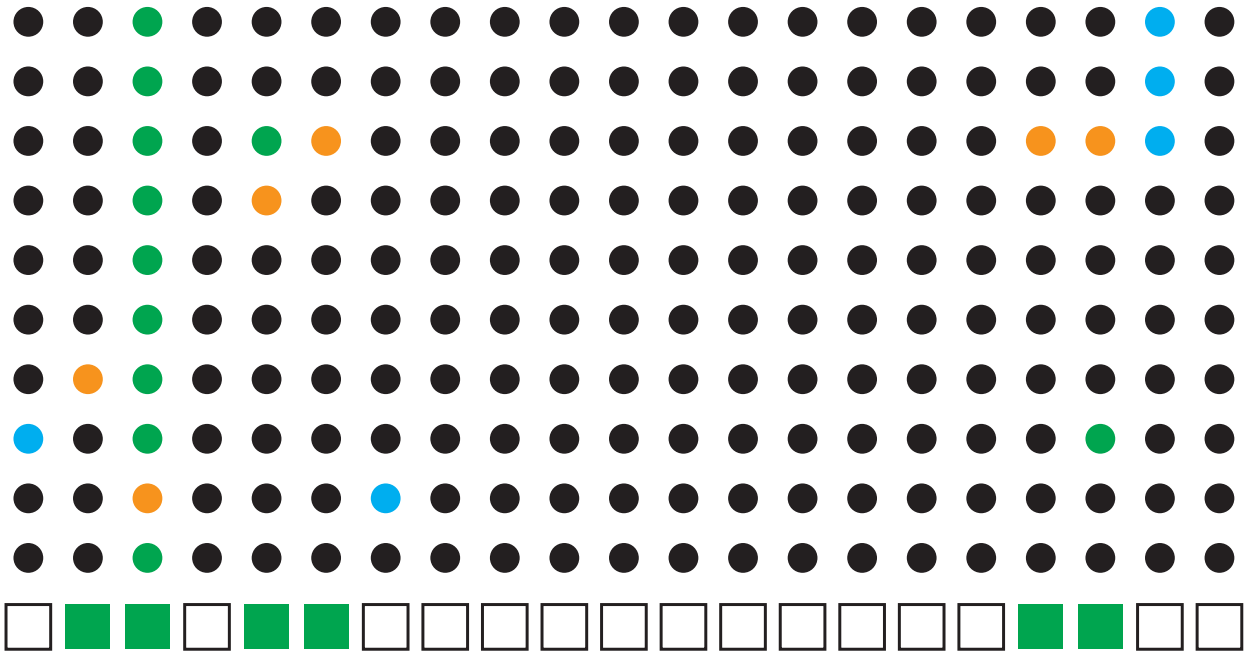


Temporal Memory Example

Input Space

EGA	β-hCG
3.0	40 mIU/mL
3.2	76 mIU/mL
3.4	130 mIU/mL

Temporal Memory



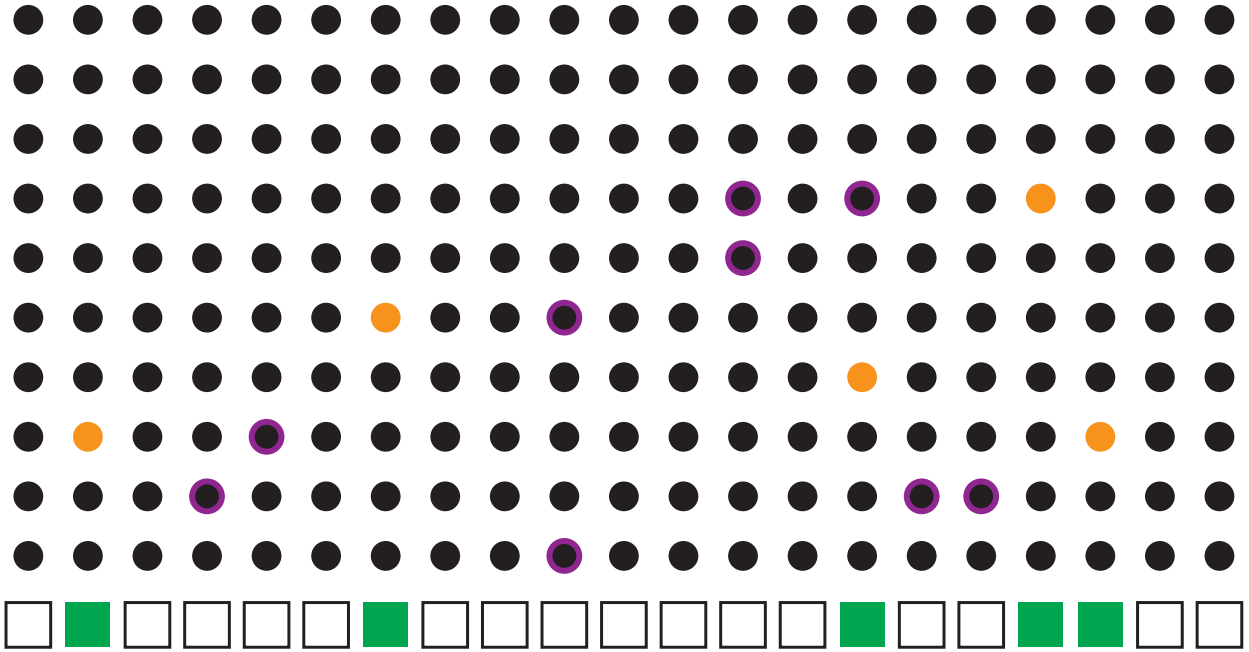
$$AnomalyScore = \frac{|A_t - (P_{t-1} \cap A_t)|}{|A_t|}$$

Temporal Memory Example

Input Space

EGA	β-**hCG**
3.0	40 mIU/mL

Temporal Memory



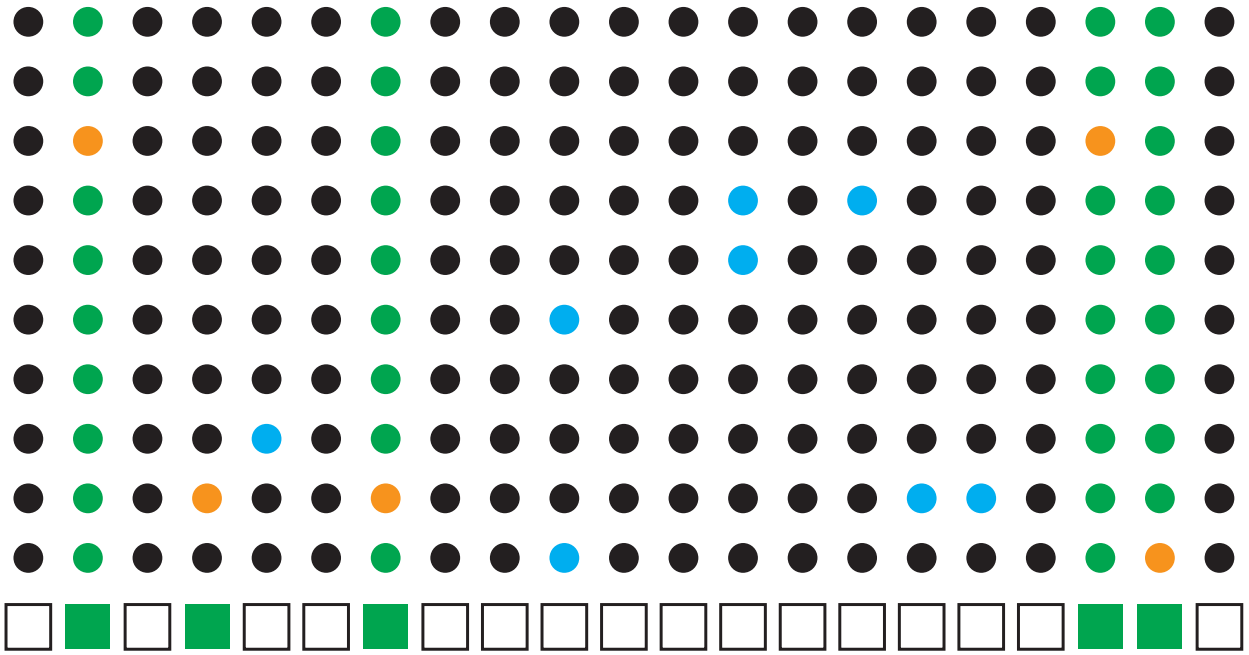
$$AnomalyScore = \frac{|A_t - (P_{t-1} \cap A_t)|}{|A_t|}$$

Temporal Memory Example

Input Space

EGA	β-hCG
3.0	40 mIU/mL
3.2	35 mIU/mL

Temporal Memory



$$AnomalyScore = \frac{|A_t - (P_{t-1} \cap A_t)|}{|A_t|}$$

Temporal Memory Parameters

Activation Threshold	Required active predictions to predict a cell
Synapse Increment	Amount connection is incremented between active cells and predicted active cells
Synapse Decrement	Amount connection is decrement between non-active cells and predicted active cells
Synapse Size	Total number of connections from a cell, a winning cell will grow connections to reach this number
Max New Synapse	Maximum number of connection added to a winning cell
Global Decay	Value to decrease connection strength for all connections at **Max Age** frequency (forget infrequent connections)
Max Age	The frequency at which to apply **Global Decay**
Max Sequence Length	Maximum number of inputs that can be learned as one sequence

Questions?