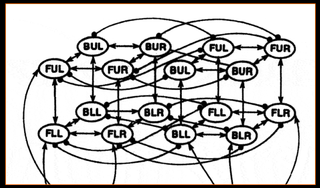
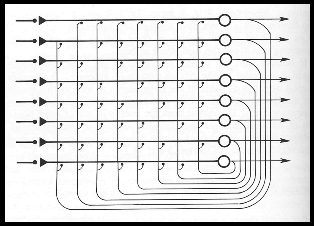
Associative Learning and Feature Maps

Learning

Learning

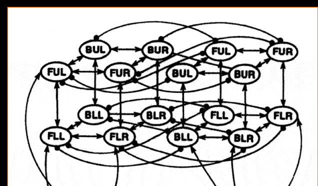
- So far, we've focused on **processing:**

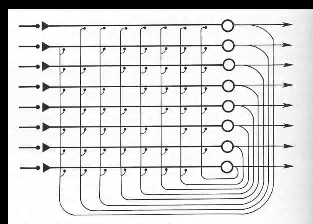


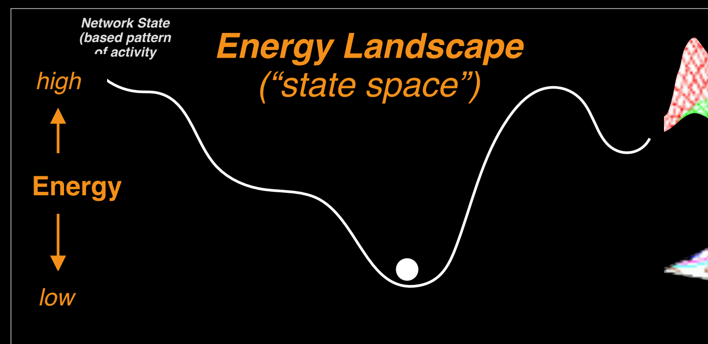


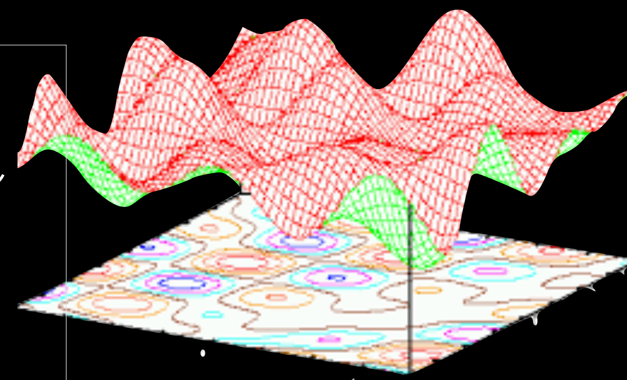
Learning

- So far, we've focused on processing:
  - dynamics of ***processing***: *encoding* and *representation* information



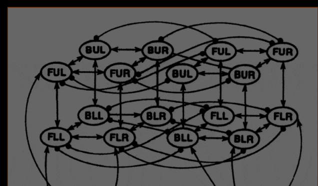


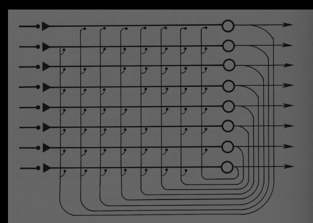


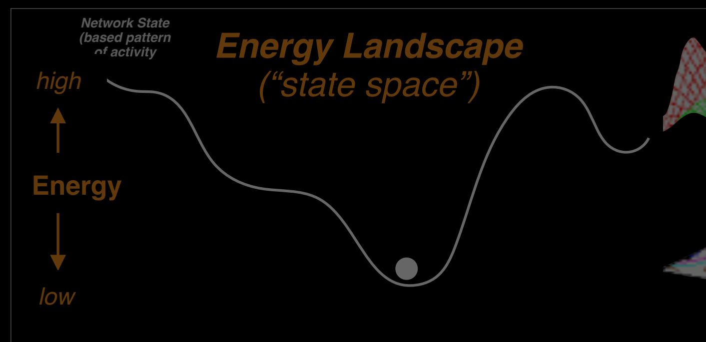


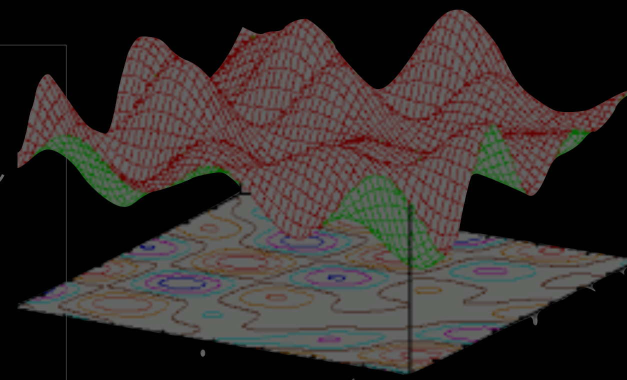
Learning

- So far, we've focused on processing:
  - dynamics of ***processing***: *encoding* and *representation* information









- What about learning?
  - how is the landscape shaped?
  - dynamics of ***acquisition***

Learning

- Unsupervised Learning
  - Hebbian Learning Rule
  - Self-organized maps
  - Topographic structure
  - Pattern associator
  - Pattern detectors
- Supervised Learning
  - Scalar Learning
    - Classical and Instrumental Conditioning
    - Sequential learning and Prediction
  - Vector-Based Learning
    - Generalized Delta Rule
    - Backpropagation
    - Deep Learning

Learning

- Unsupervised Learning
  - Hebbian Learning Rule
  - Self-organized maps
  - Topographic structure
  - Pattern associator
  - Pattern detectors
- Supervised Learning
  - Scalar Learning
    - Classical and Instrumental Conditioning
    - Sequential learning and Prediction
  - Vector-Based Learning
    - Generalized Delta Rule
    - Backpropagation
    - Deep Learning

Hebbian Learning

Hebbian Learning

- **D. O. Hebb:** *(1949)*
  "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

Hebbian Learning

- **D. O. Hebb:** *(1949)*

  "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

- **Critical factor**
  - **concurrent presynatptic and postsynaptic activity:** ***correlation***
  - **units that *"fire together wire together"***

Hebbian Learning

- **D. O. Hebb:** *(1949)*
  "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

- **Critical factor**
  - **concurrent presynatptic and postsynaptic activity:** ***correlation***
  - **units that *"fire together wire together"***

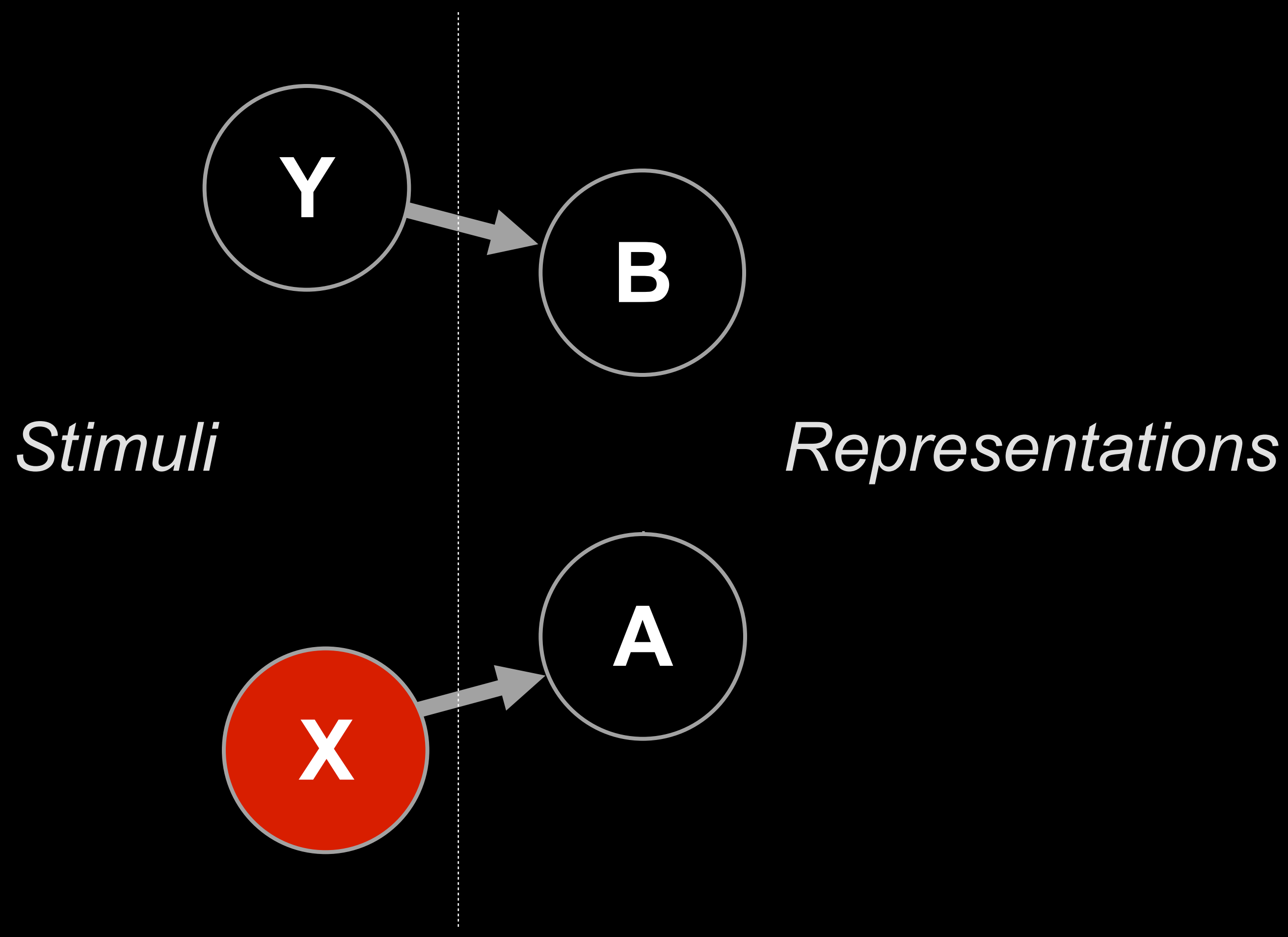
- **Fundamental learning mechanism**
  - **responsible for much of how we gain our knowledge**

Hebbian Learning

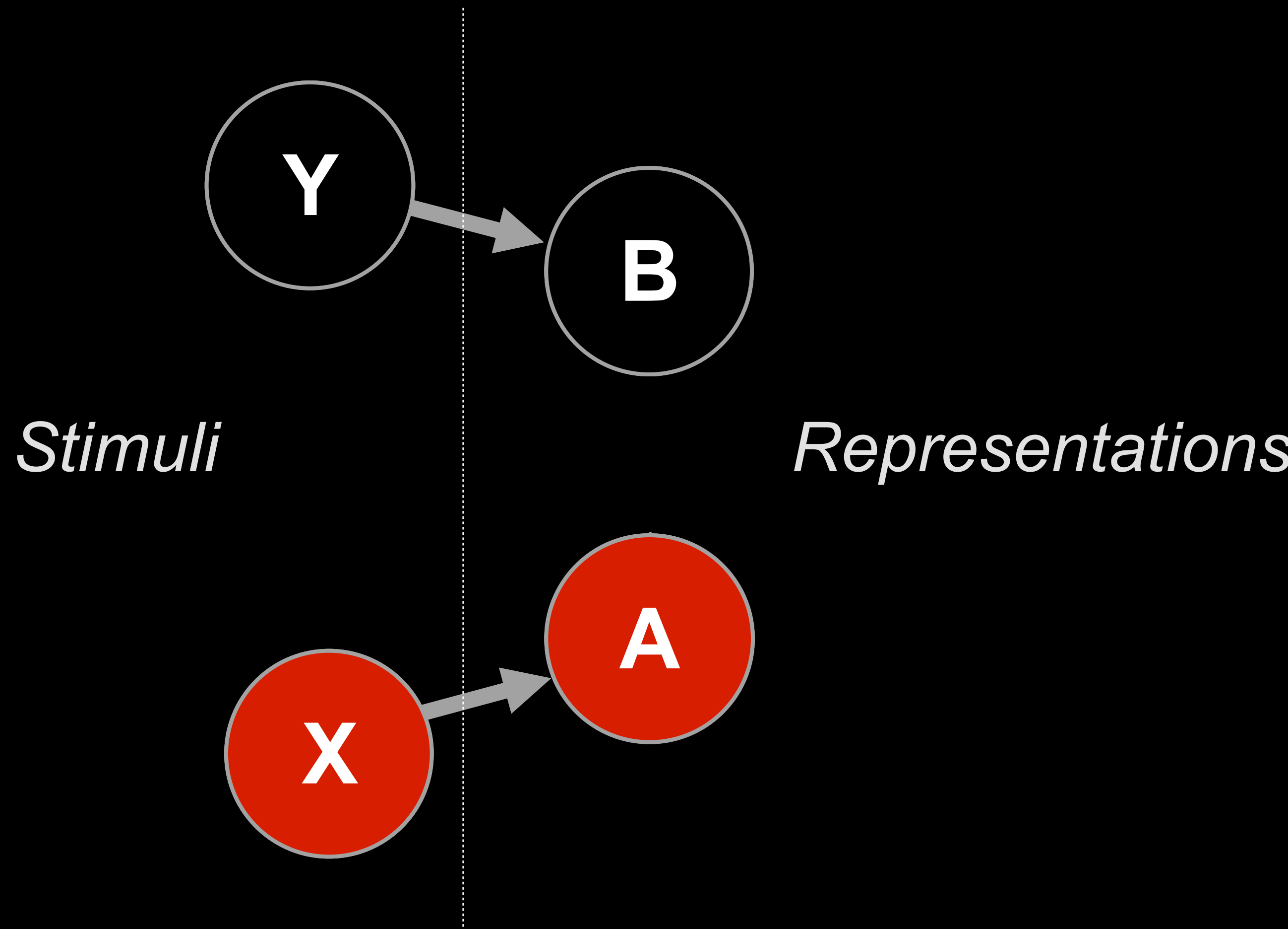
Stimuli

Representations

Hebbian Learning

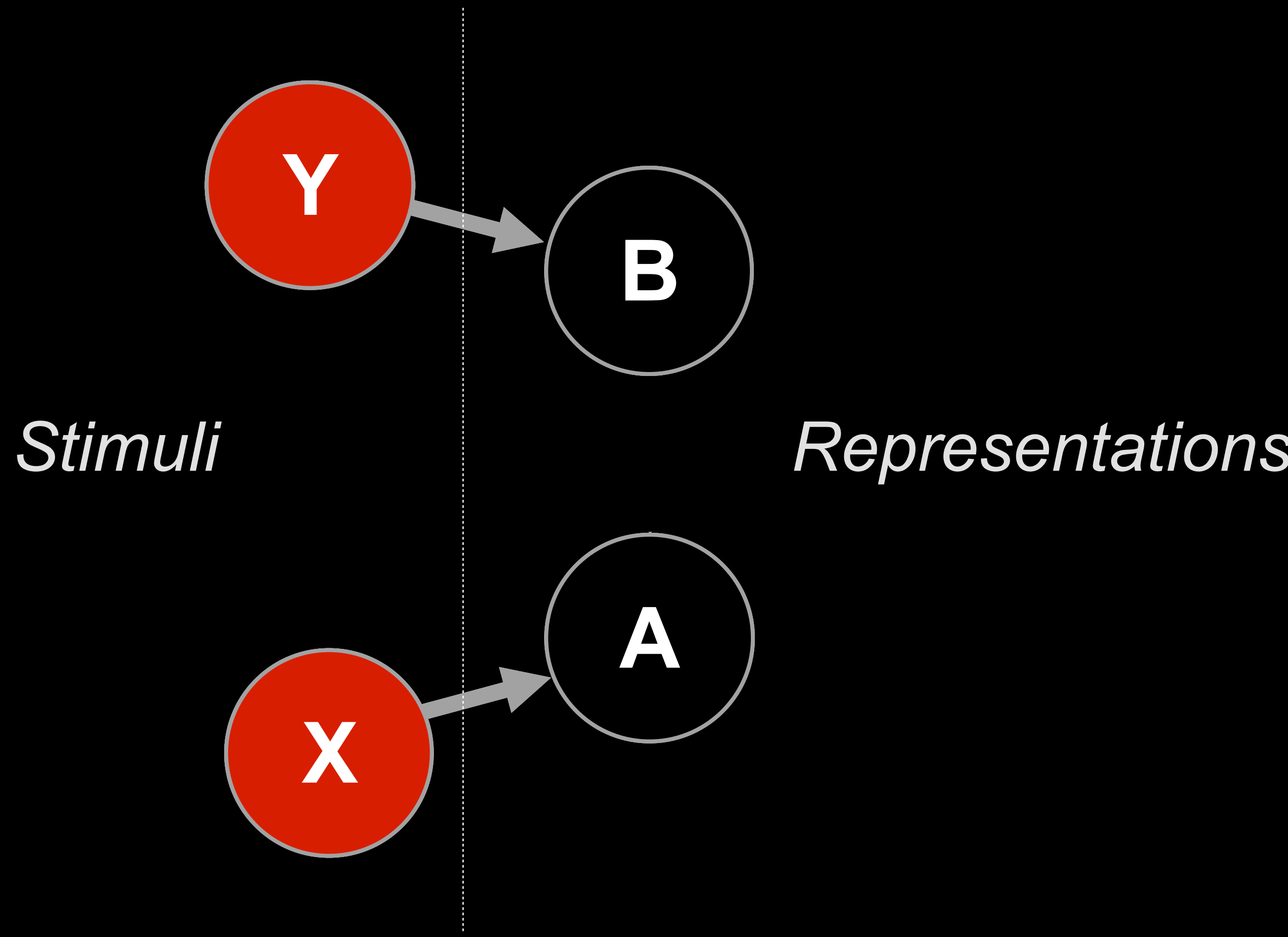


Hebbian Learning



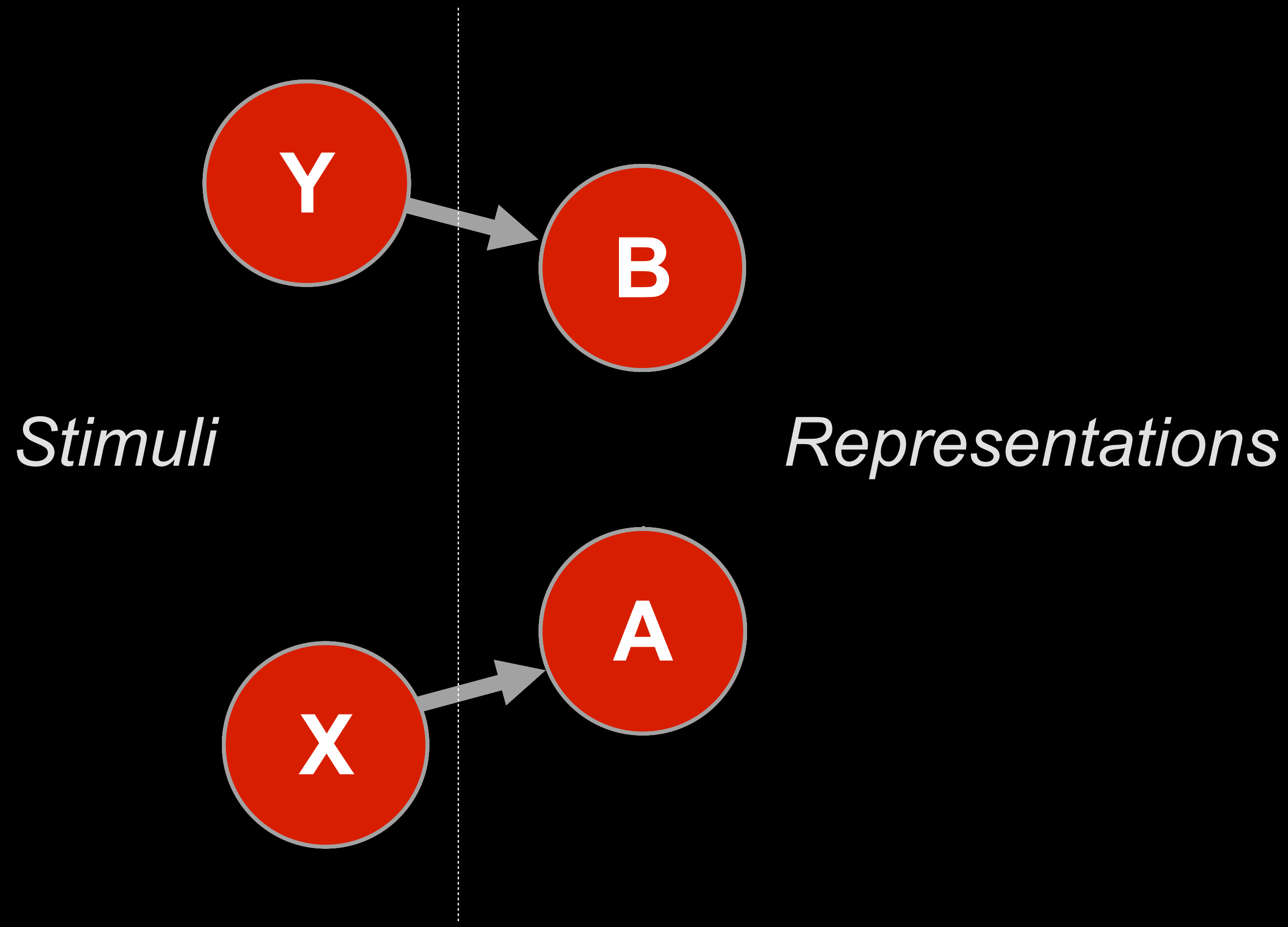
X activates A but not B

Hebbian Learning



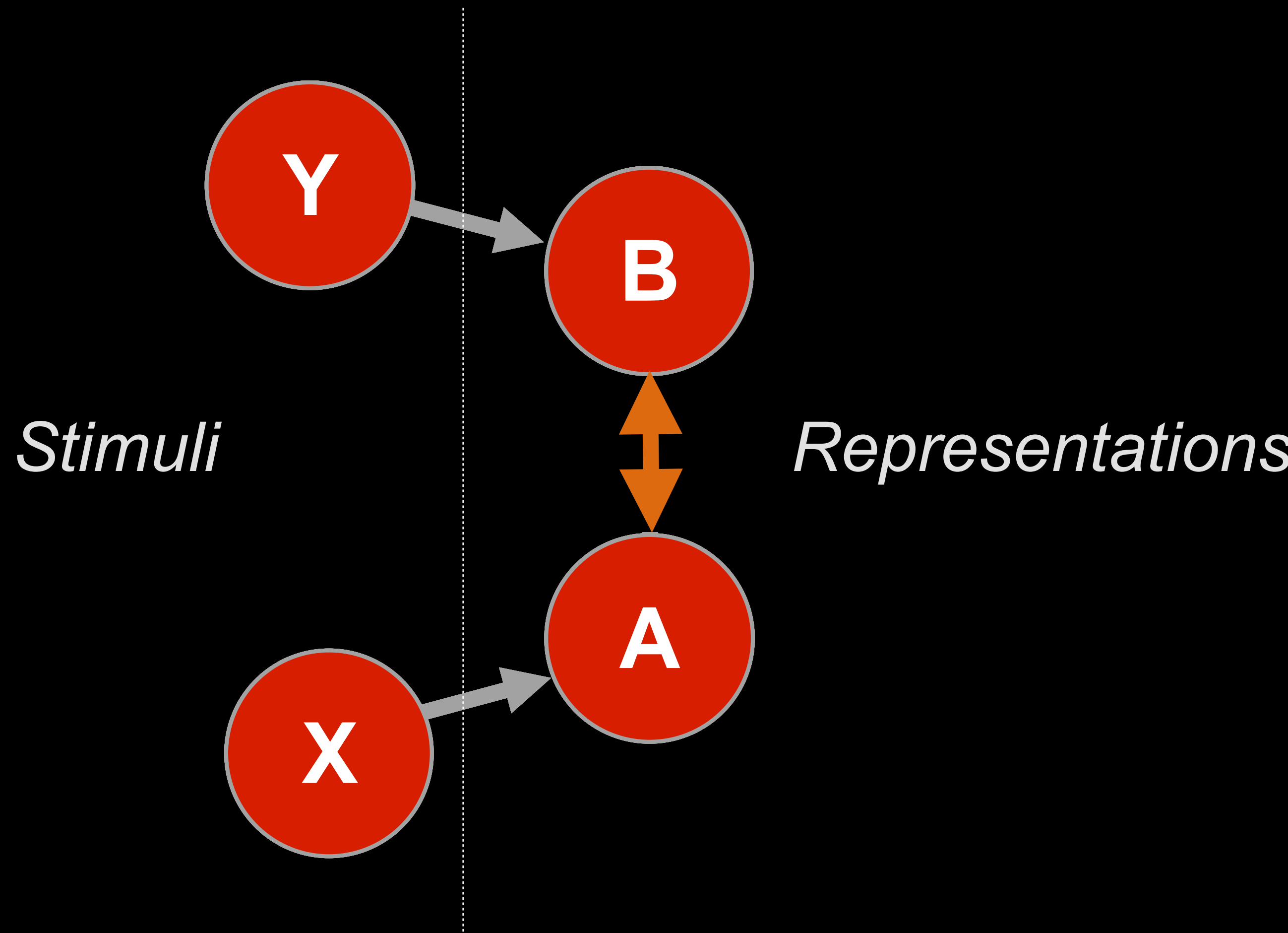
X and Y co-activate A and B

Hebbian Learning



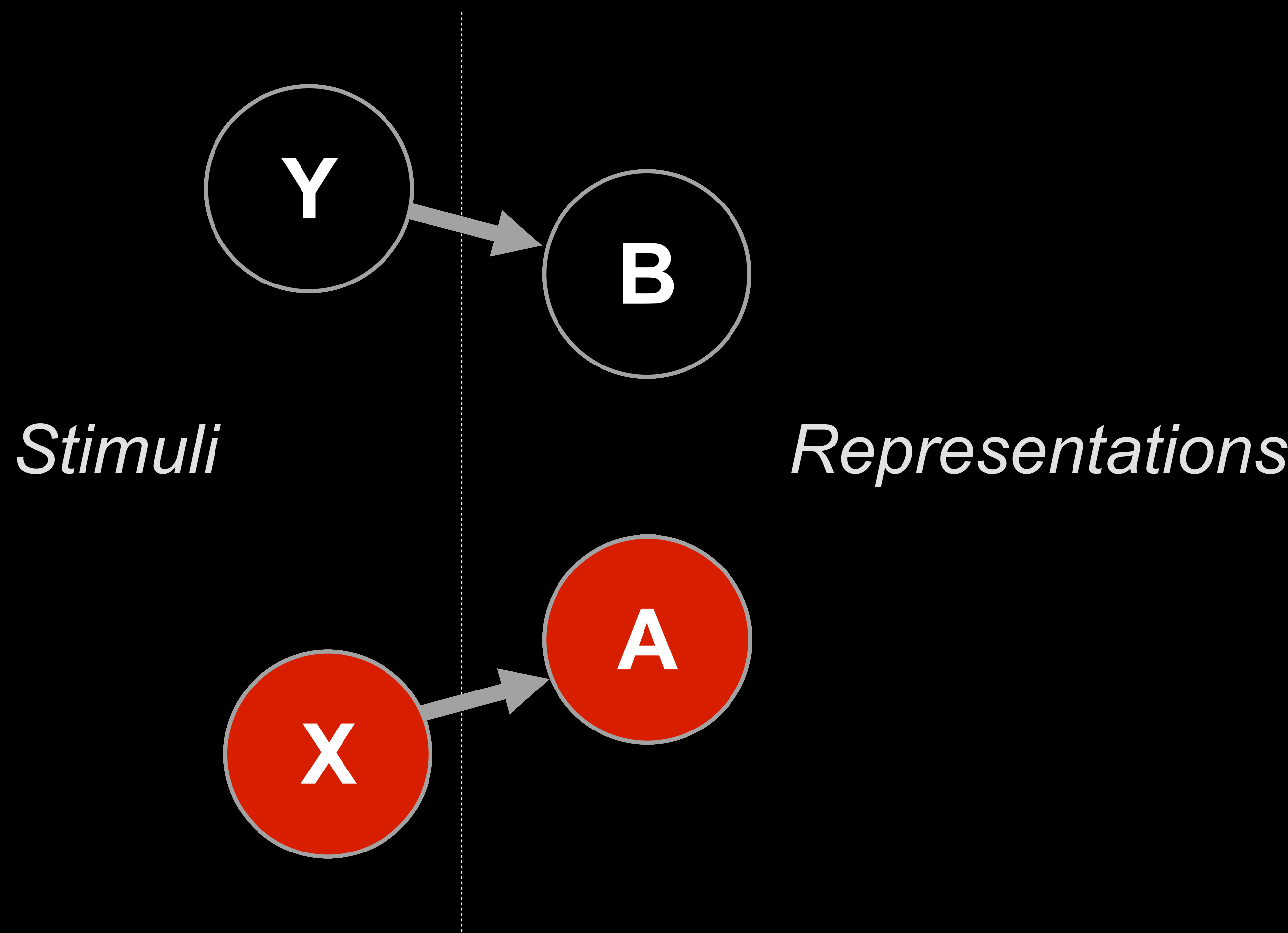
X and Y co-activate A and B

Hebbian Learning



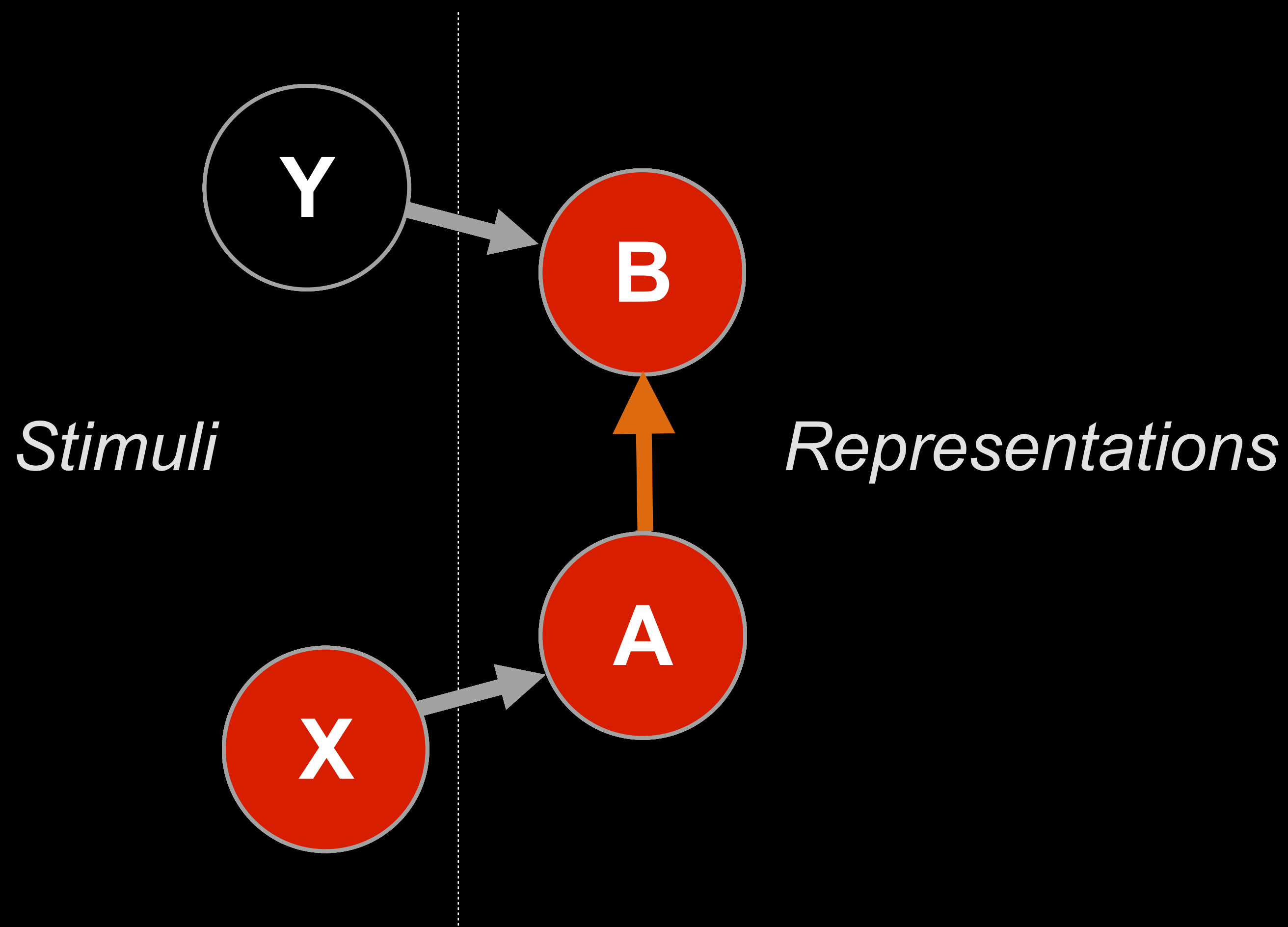
…which strengthens connection between A and B

Hebbian Learning



Activating X activates A *which now activates B*

Hebbian Learning



Activating X activates A *which now activates B*

Hebbian Learning

Hebbian Learning

Formalism:

$$\Delta w_{ij} = \alpha \, a_i a_j$$

where α is the learning rate and a can be any real number

Hebbian Learning

Formalism:

$$\Delta w_{ij} = \alpha\, a_i a_j$$

where α is the learning rate and a can be any real number

After n "experiences:"

$$w_{ij} = \alpha \sum_n a_{in} a_{jn}$$

Hebbian Learning

Formalism:

$$\Delta w_{ij} = \alpha\, a_i a_j$$

where α is the learning rate and a can be any real number

After n "experiences:"

$$w_{ij} = \alpha \textstyle\sum_n a_{in} a_{jn}$$

"Correlational Learning:

$w_{ij} \equiv$ correlation of a_i and a_j over time *(patterns)*

if a_i and a_j vary linearly from -1 to +1 (i.e., mean=0 and unit variance)

Hebbian Learning

Formalism:

$$\Delta w_{ij} = \alpha \, a_i a_j$$

where α is the learning rate and a can be any real number

After n "experiences:"

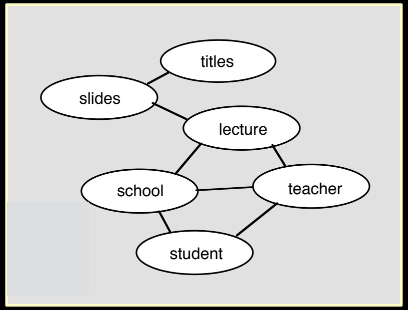
$$w_{ij} = \alpha \, \Sigma_n a_{in} a_{jn}$$

"Correlational Learning:

w_{ij} ≡ correlation of a_i and a_j over time *(patterns)*

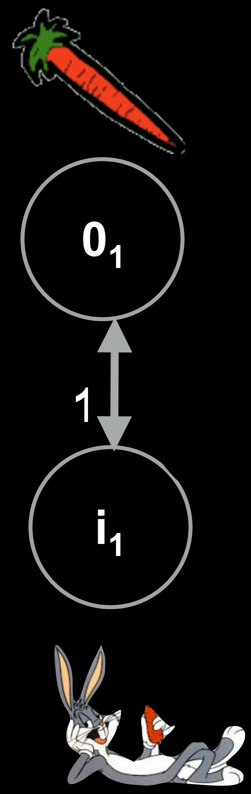
if a_i and a_j vary linearly from -1 to +1 (i.e., mean=0 and unit variance)

Captures statistical relationship among co-occurring features



Multiple Associations

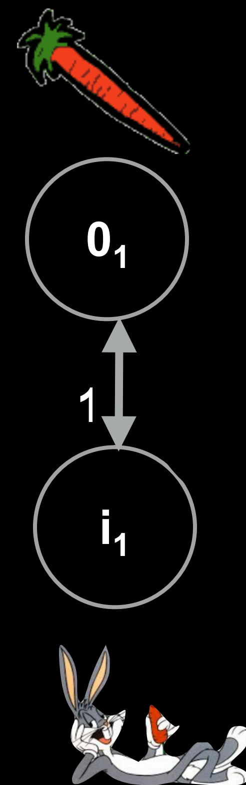
	 x1	 x2
trial 1	+	+
trial 2	+	+
trial 3	−	−
trial 4	−	−



Multiple Associations

	 x1	 x2
trial 1	+	+
trial 2	+	+
trial 3	−	−
trial 4	−	−

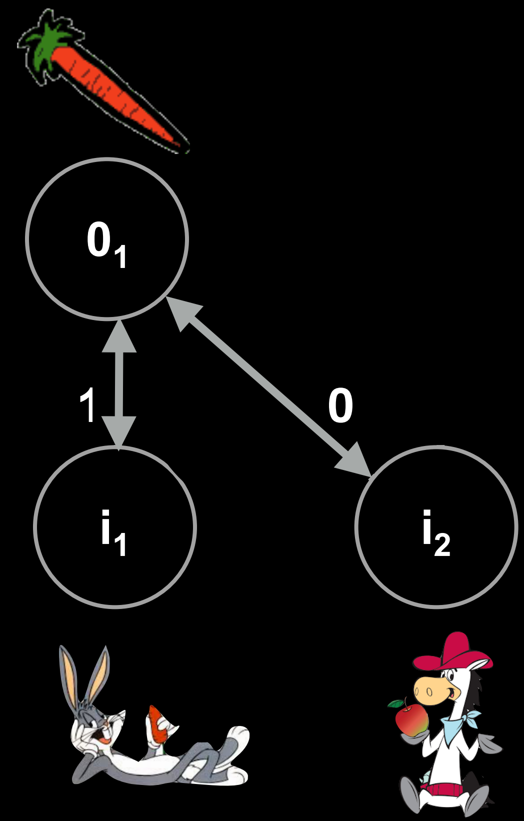
Correlated



Multiple Associations

	x1	i_2	x2
trial 1	+	+	+
trial 2	+	−	+
trial 3	−	+	−
trial 4	−	−	−

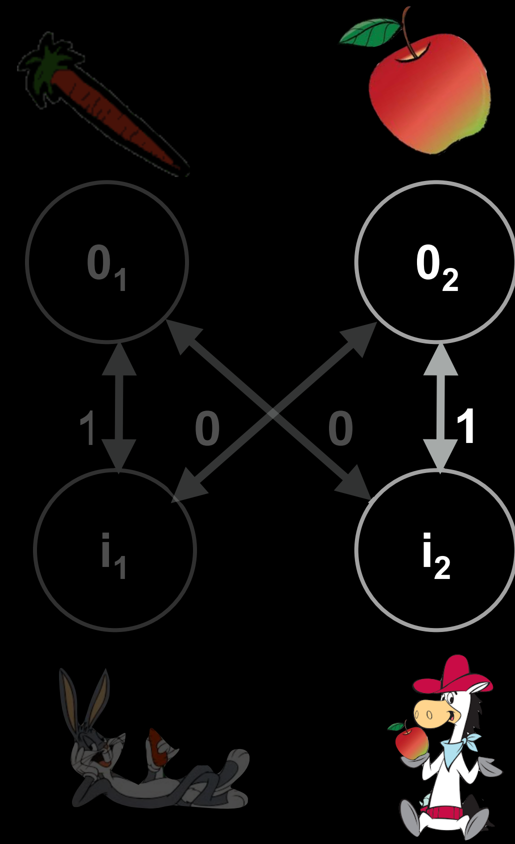
Correlated *Uncorrelated*



Multiple Associations

	$x1$	i_2	$x2$	o_2
trial 1	+	+	+	+
trial 2	+	−	+	−
trial 3	−	+	−	+
trial 4	−	−	−	−

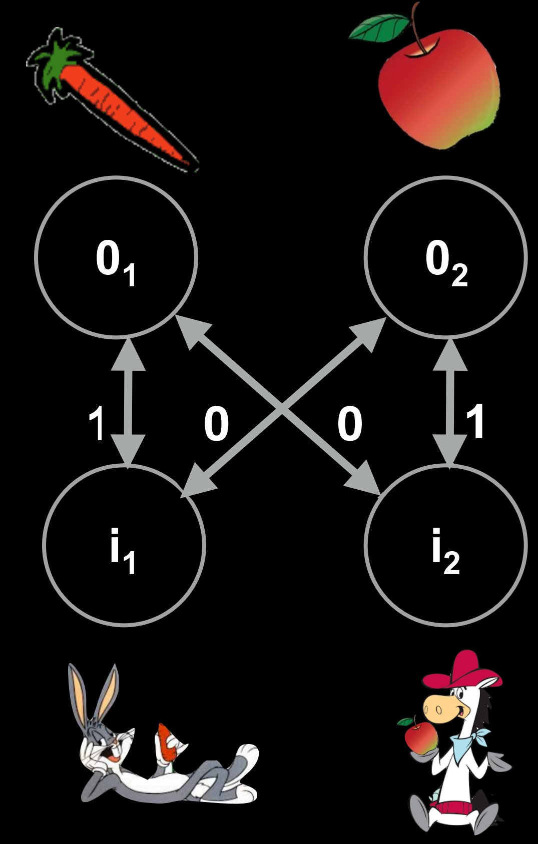
Correlated *Uncorrelated* *Correlated*



Multiple Associations

	x1	i_2	x2	o_2
trial 1	+	+	+	+
trial 2	+	−	+	−
trial 3	−	+	−	+
trial 4	−	−	−	−

Correlated *Uncorrelated* *Correlated*



A Bit O' Math

A Bit O' Math

- Patterns can be considered as vectors (lists of activation values) and relationships between them described using linear algebra

A Bit O' Math

- Patterns can be considered as vectors (lists of activation values) and relationships between them described using linear algebra

- Normalized Dot Product (NDP) of two patterns a and b over n units:

  $$\mathbf{a} \cdot \mathbf{b} = \left(\Sigma_i a_i b_i\right) / n$$

A Bit O' Math

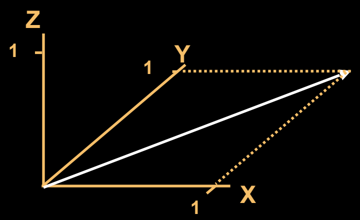
- Patterns can be considered as vectors (lists of activation values) and relationships between them described using linear algebra

- Normalized Dot Product (NDP) of two patterns *a* and *b* over *n* units:

  $a \cdot b = (\Sigma_i a_i b_i) / n$

- NDP combines measure of strength and similarity

  - Strength of pattern: vector length, normalized for # of elements
    - *Tip: this is the Euclidean distance from the origin to the point defined by the vector; (≈ hypotenuse of the triangle defined by the vector and its distance along each axis*

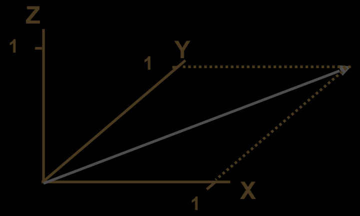


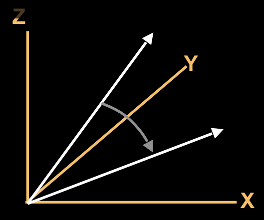
A Bit O' Math

- Patterns can be considered as vectors (lists of activation values) and relationships between them described using linear algebra
- Normalized Dot Product (NDP) of two patterns *a* and *b* over *n* units:

  $a \cdot b = (\Sigma_i a_i b_i) / n$

- NDP combines measure of strength and similarity
  - Strength of pattern: vector length, normalized for # of elements
    - *Tip: this is the Euclidean distance from the origin to the point defined by the vector; (≈ hypotenuse of the triangle defined by the vector and its distance along each axis*
  - Similarity: correlation, independent of length
    - *Tip: this is the cosine of the angle between the two vectors*
      0° = similar (+ correlation)
      90° = unrelated (0 correlation)
      180° = opposite (- correlation)





A Bit O' Math

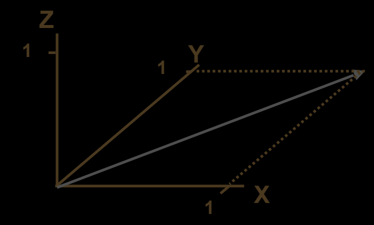
- Patterns can be considered as vectors (lists of activation values) and relationships between them described using linear algebra

- Normalized Dot Product (NDP) of two patterns *a* and *b* over *n* units:

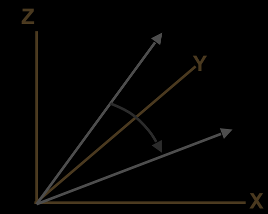
  $a \cdot b = (\Sigma_i a_i b_i) / n$

- NDP combines measure of strength and similarity

  - Strength of pattern: vector length, normalized for # of elements
    - *Tip: this is the Euclidean distance from the origin to the point defined by the vector; (≈ hypotenuse of the triangle defined by the vector and its distance along each axis*



  - Similarity: correlation, independent of length
    - *Tip: this is the cosine of the angle between the two vectors*
      0° = similar (+ correlation)
      90° = unrelated (0 correlation)
      180° = opposite (- correlation)



- **Two patterns whose NDP = 0 are said to be "orthogonal"**
  - ***Tip: Vectors that are "perpendicular" in 3D space are orthogonal (compute the NDP for the x axis against the y axis); this is because they are uncorrelated***

Associative Learning and
Internal Representations / Model Building

Associative Learning and Internal Representations / Model Building

- The role of associative learning in model building
  - Correlations are important for building internal models of the world:
    - the world is inhabited by objects and agents with features that are in consistent relationship to one another

Associative Learning and Internal Representations / Model Building

- The role of associative learning in model building
  - Correlations are important for building internal models of the world:
    - ♦ the world is inhabited by objects and agents with features that are in consistent relationship to one another
    - ♦ **these regularities are useful for identification and prediction** (predators have fangs; when it is warm fruit will be available; types of faces)

Associative Learning and Internal Representations / Model Building

- The role of associative learning in model building
  - Correlations are important for building internal models of the world:
    - the world is inhabited by objects and agents with features that are in consistent relationship to one another
    - these regularities are useful for identification and prediction (predators have fangs; when it is warm fruit will be available; types of faces)
    - correlations among features define dimensions that are relevant for and efficient at describing and understanding the world

Associative Learning and Internal Representations / Model Building

- The role of associative learning in model building
  - Correlations are important for building internal models of the world:
    - the world is inhabited by objects and agents with features that are in consistent relationship to one another
    - these regularities are useful for identification and prediction (predators have fangs; when it is warm fruit will be available; types of faces)
    - correlations among features define dimensions that are relevant for and efficient at describing and understanding the world
- Extracting regularities is a fundamental job of cognition:
  - Parsimony/ Abstraction: can describe a complex world with finite resources

Associative Learning and Internal Representations / Model Building

- The role of associative learning in model building
  - Correlations are important for building internal models of the world:
    - the world is inhabited by objects and agents with features that are in consistent relationship to one another
    - these regularities are useful for identification and prediction
      (predators have fangs; when it is warm fruit will be available; types of faces)
    - correlations among features define dimensions that are relevant for and efficient at describing and understanding the world
- Extracting regularities is a fundamental job of cognition:
  - Parsimony/ Abstraction: can describe a complex world with finite resources
  - Generalization: infer properties of the world in novel circumstances

Associative Learning and Internal Representations / Model Building

- The role of associative learning in model building
  - Correlations are important for building internal models of the world:
    - the world is inhabited by objects and agents with features that are in consistent relationship to one another
    - these regularities are useful for identification and prediction (predators have fangs; when it is warm fruit will be available; types of faces)
    - correlations among features define dimensions that are relevant for and efficient at describing and understanding the world
- Extracting regularities is a fundamental job of cognition:
  - Parsimony/ Abstraction: can describe a complex world with finite resources
  - Generalization: infer properties of the world in novel circumstances
  - Efficiency of learning: can represent novel items with existing codes

More Generally…

More Generally…

- Can think of associative networks as implementing "exploratory" analysis of environment

More Generally…

- Can think of associative networks as implementing "exploratory" analysis of environment

- Parameterization implements different classes of statistical functions

Various Approaches

- **Pattern associator**
  - Inputs + detector units -> network implementation of Principal Components Analysis (PCA)
- **Linsker's Information Maximization**
  - Multiple detector units, similar to PCA network:
    maximizing variance in output units ≈ maximizing information
    (in limit not useful, since no dimension reduction ∴ no generalization)
- **Kohonen Network**
  - Multiple detector units with structured local connections among them:
    captures neighborhood relationships among features; topographic maps
    (Ken Miller's simulations of ocular dominance columns)
- **Competitive Learning (winner-take-all)**
  - Multiple detector units but only one allowed to be active;
    forces different detectors to identify different correlations among input units
- **Minimum Description Length (K-winners-take-all)**
  - Similar to competitive learning, but a small set of detectors can be active;
    trades off maximizing information against minimizing complexity
- **LEABRA**
  - Combines K-winners-take-all competition with error-driven learning

Various Approaches

- Pattern associator
  - Inputs + detector units -> network implementation of Principal Components Analysis (PCA)
- Linsker's Information Maximization
  - Multiple detector units, similar to PCA network:
    maximizing variance in output units ≈ maximizing information
    (in limit not useful, since no dimension reduction ∴ no generalization)
- **Kohonen Network**
  - **Multiple detector units with structured local connections among them: captures neighborhood relationships among features; topographic maps**
    (Ken Miller's simulations of ocular dominance columns)
- Competitive Learning (winner-take-all)
  - Multiple detector units but only one allowed to be active;
    forces different detectors to identify different correlations among input units
- Minimum Description Length (K-winners-take-all)
  - Similar to competitive learning, but a small set of detectors can be active;
    trades off maximizing information against minimizing complexity
- LEABRA
  - Combines K-winners-take-all competition with error-driven learning

Topographic Organization

Topographic Organization

- **Associative learning can extract structure in the world, and represent it *structurally* (topographically)**

Topographic Organization

- Associative learning can extract structure in the world, and represent it *structurally* (topographically)
- There is (lots of) topographic organization in the nervous system:
  - Retina (spatiotopic), inner ear (tonotopic), sensory and motor cortex
  - Exploited for imaging *(e.g., retinotopic mapping of primary visual cortex)*
  - Even as it gets more complex, some topography is maintained:
    - Occular dominance columns *(Miller, 1989)*
    - Orientation and retinotopic position "pinwheels" *(Durbin & Mitchison, 1990)*

Topographic Organization

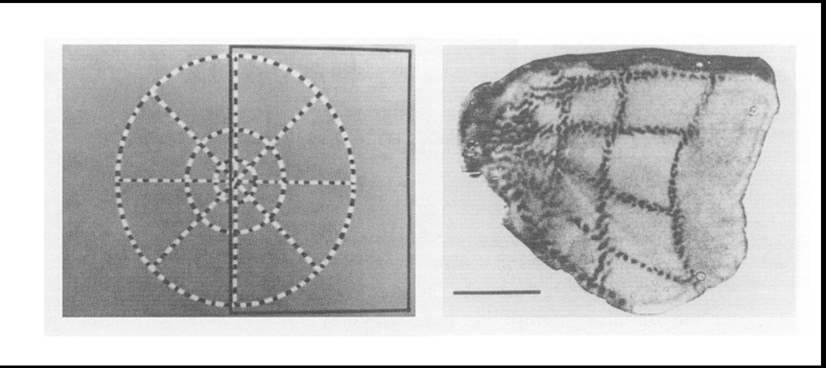
- Associative learning can extract structure in the world, and represent it *structurally* (topographically)

- There is (lots of) topographic organization in the nervous system:
  - Retina (spatiotopic), inner ear (tonotopic), sensory and motor cortex
  - Exploited for imaging *(e.g., retinotopic mapping of primary visual cortex)*
  - Even as it gets more complex, some topography is maintained:
    - Occular dominance columns *(Miller, 1989)*
    - Orientation and retinotopic position "pinwheels" *(Durbin & Mitchison, 1990)*

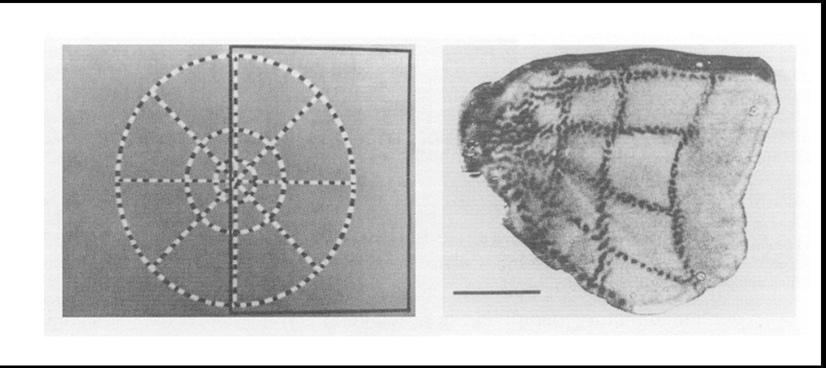
- These may reflect meaningful relationships that exist in the "data" *(i.e., the "real world")*

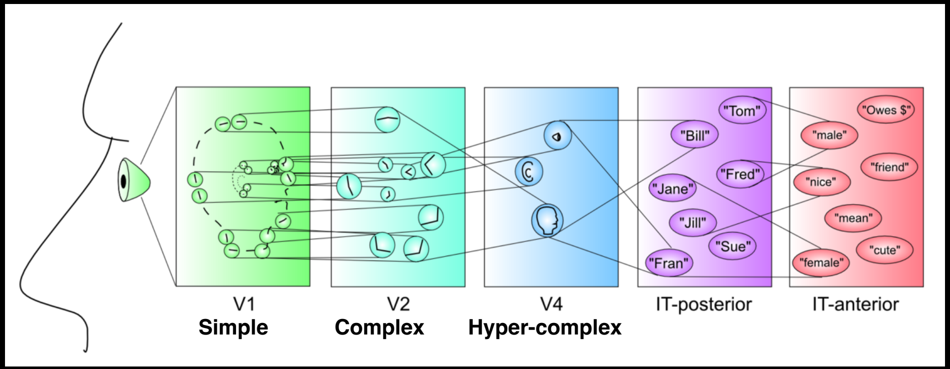
Topographic Organization

Topographic Organization

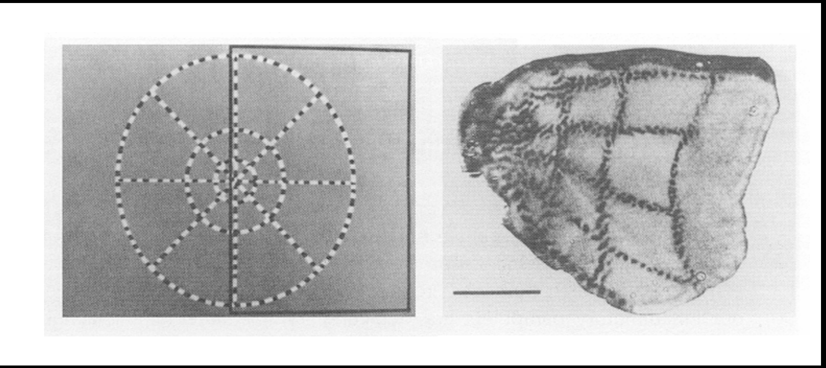


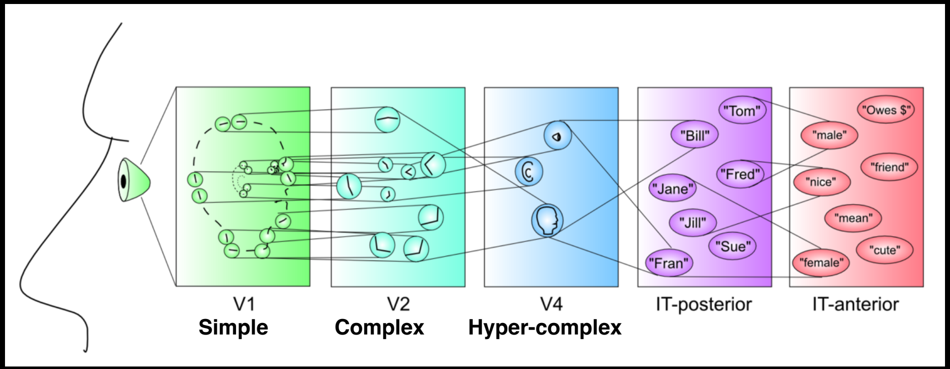
Topographic Organization

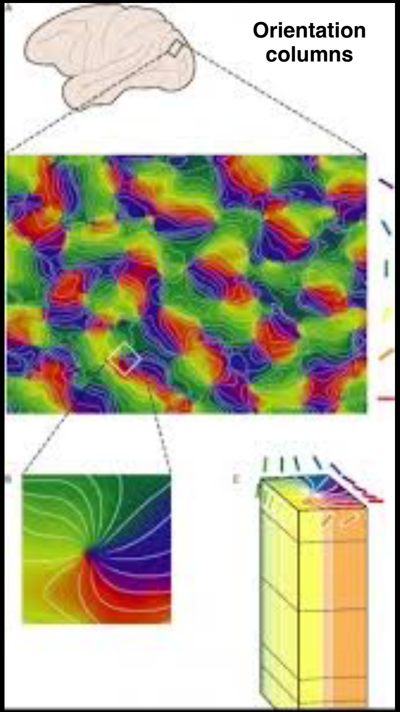




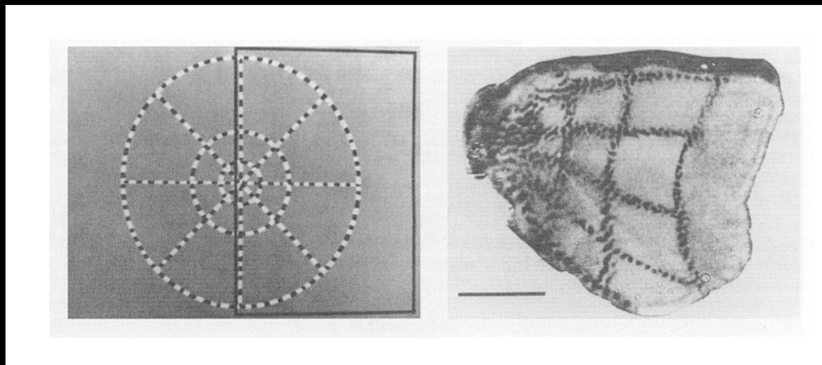
Topographic Organization

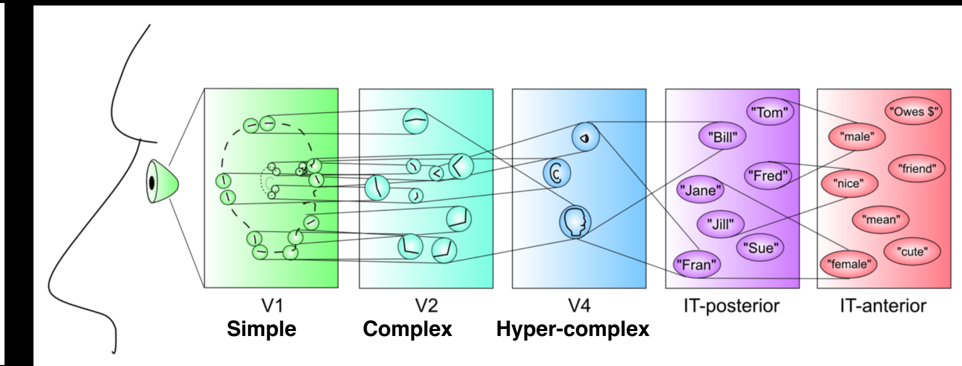


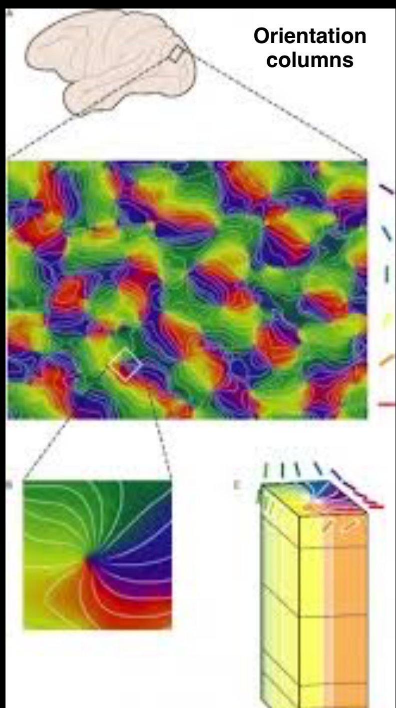


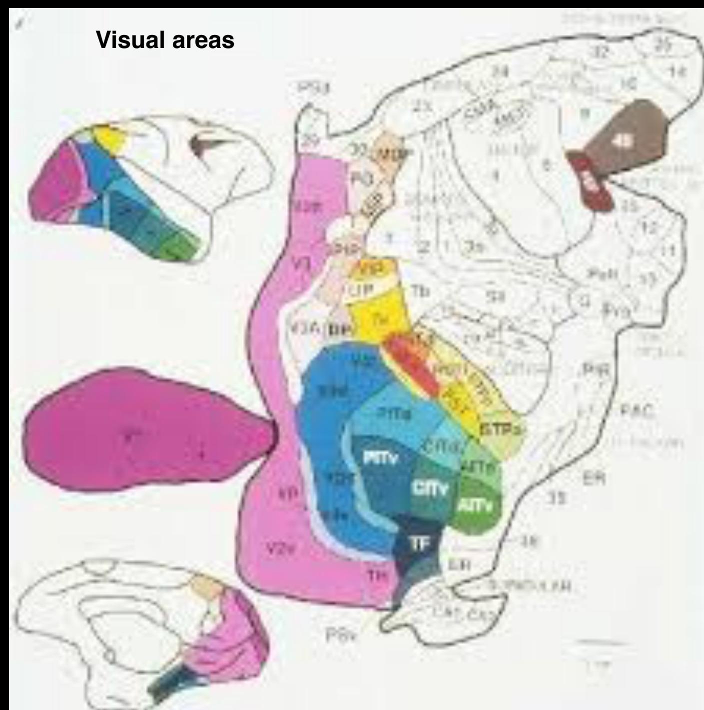


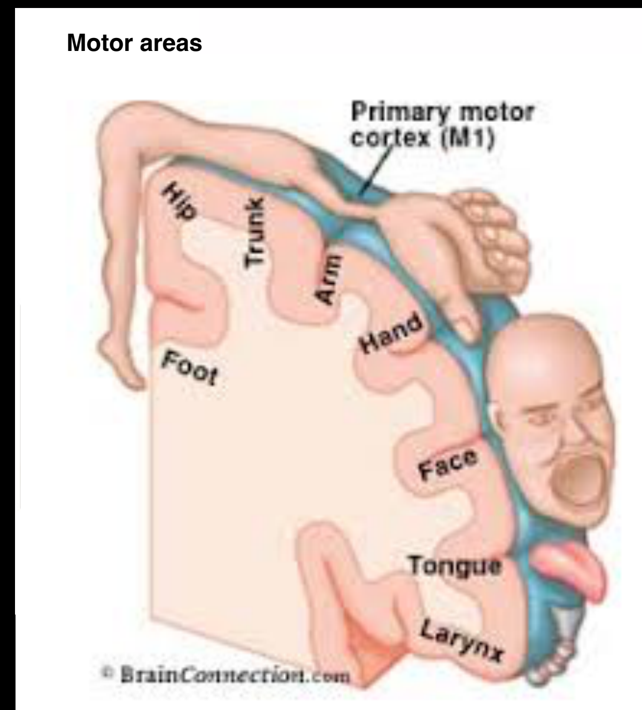
Topographic Organization











“Dimension Reduction”

“Dimension Reduction”

- How does this structure arise?

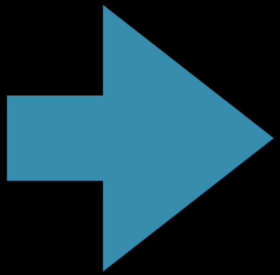
“Dimension Reduction”

- How does this structure arise?
- Challenge:

“Dimension Reduction”

- How does this structure arise?
- Challenge:

3D



"Dimension Reduction"

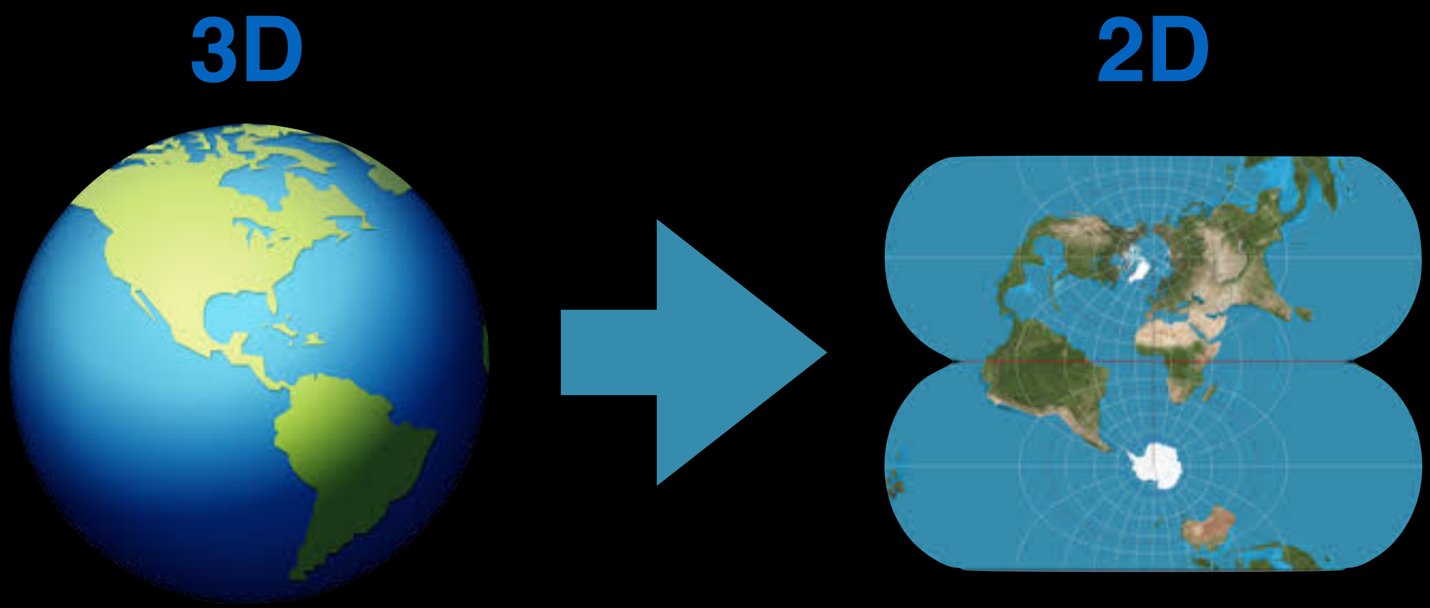
- How does this structure arise?
- Challenge:

3D

2D

"Dimension Reduction"

- How does this structure arise?
- Challenge:



- What about even higher dimensional data?

Self-Organizing Maps (SOMs)

Kohonen Network (1982)

Self-Organizing Maps (SOMs)

Kohonen Network (1982)

- **Objectives:**
  - **Map input vectors (patterns) of dimension *N* onto a map with 1 or 2 dimensions.**

Self-Organizing Maps (SOMs)

Kohonen Network (1982)

- **Objectives:**
  - **Map input vectors (patterns) of dimension *N* onto a map with 1 or 2 dimensions.**
  - **Patterns *close* to one another in the input space should project to *nearby* units ("map" should be *topographically ordered*)**

Self-Organizing Maps (SOMs)

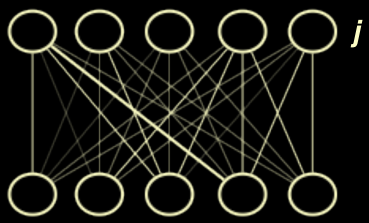
Kohonen Network (1982)

- **Objectives:**
  - **Map input vectors (patterns) of dimension *N* onto a map with 1 or 2 dimensions.**
  - **Patterns *close* to one another in the input space should project to *nearby* units ("map" should be *topographically ordered*)**
- **Network architecture and input environment *("training")***
  - **Input layer:**
    - **units that code a space of *vectors with structure*, but *not spatially arranged***

Self-Organizing Maps (SOMs)

Kohonen Network (1982)

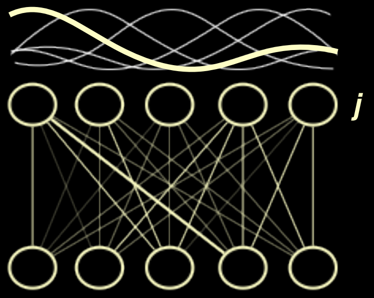
- **Objectives:**
  - Map input vectors (patterns) of dimension *N* onto a map with 1 or 2 dimensions.
  - Patterns *close* to one another in the input space should project to *nearby* units ("map" should be *topographically ordered*)
- **Network architecture and input environment *("training")***
  - Input layer:
    - units that code a space of *vectors with structure*, but *not spatially arranged*
  - **Output layer:**
    - each unit *j* has *weights from all units* in *input layer*



Self-Organizing Maps (SOMs)

Kohonen Network (1982)

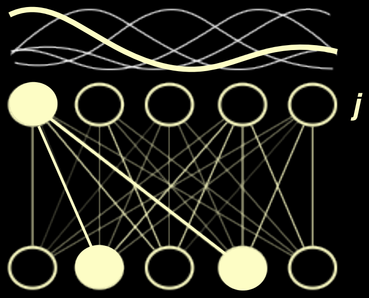
- **Objectives:**
  - Map input vectors (patterns) of dimension *N* onto a map with 1 or 2 dimensions.
  - Patterns *close* to one another in the input space should project to *nearby* units ("map" should be *topographically ordered*)
- **Network architecture and input environment *("training")***
  - Input layer:
    - units that code a space of *vectors with structure*, but *not spatially arranged*
  - Output layer:
    - each unit *j* has *weights from all units* in *input layer*
    - each unit *j* has a *defined distance* from all *other units* in the *output layer*



Self-Organizing Maps (SOMs)

Kohonen Network (1982)

- **Objectives:**
  - Map input vectors (patterns) of dimension *N* onto a map with 1 or 2 dimensions.
  - Patterns *close* to one another in the input space should project to *nearby* units ("map" should be *topographically ordered*)
- **Network architecture and input environment *("training")***
  - Input layer:
    - units that code a space of *vectors with structure*, but *not spatially arranged*
  - Output layer:
    - each unit *j* has *weights from all units* in *input layer*
    - each unit *j* has a *defined distance* from all *other units* in the *output layer*

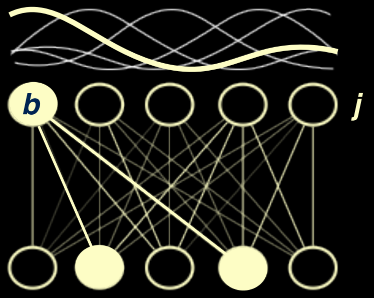


- **Learning rule:**
  - present input pattern, and identify *best matching* (most active) *output unit:*
    - one with input *current weights closest to input pattern* *("winner" of lateral competition)*

Self-Organizing Maps (SOMs)

Kohonen Network (1982)

- **Objectives:**
  - **Map input vectors (patterns) of dimension *N* onto a map with 1 or 2 dimensions.**
  - **Patterns *close* to one another in the input space should project to *nearby* units ("map" should be *topographically ordered*)**
- **Network architecture and input environment *("training")***
  - **Input layer:**
    - units that code a space of ***vectors with structure***, but ***not spatially arranged***
  - **Output layer:**
    - each unit *j* has ***weights from all units*** in ***input layer***
    - each unit *j* has a ***defined distance*** from all ***other units*** in the ***output layer***



- **Learning rule:**
  - **present input pattern, and identify *best matching* (most active) *output unit:***
    - one with input ***current weights closest to input pattern*** *("winner" of lateral competition)*
  - **adjust weights for that unit using following rule:**

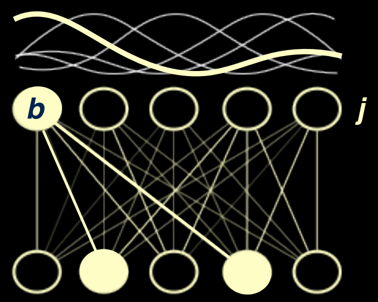
$$W_{b(t+1)} = W_{b(t)} + \mathbf{c}_{wb(t)} \cdot \mathbf{g}(t) \cdot (I - W_{b(t)})$$

*change in weights to **b*** — *closeness to **b*** — *gain* — *difference from Input pattern*

Self-Organizing Maps (SOMs)

Kohonen Network (1982)

- **Objectives:**
  - **Map input vectors (patterns) of dimension *N* onto a map with 1 or 2 dimensions.**
  - **Patterns *close* to one another in the input space should project to *nearby* units ("map" should be *topographically ordered*)**
- **Network architecture and input environment *("training")***
  - **Input layer:**
    - **units that code a space of *vectors with structure*, but *not <u>spatially</u> arranged***
  - **Output layer:**
    - **each unit *j* has *weights from all units* in *<u>input layer</u>***
    - **each unit *j* has a *defined distance* from all *other units* in the *<u>output layer</u>***



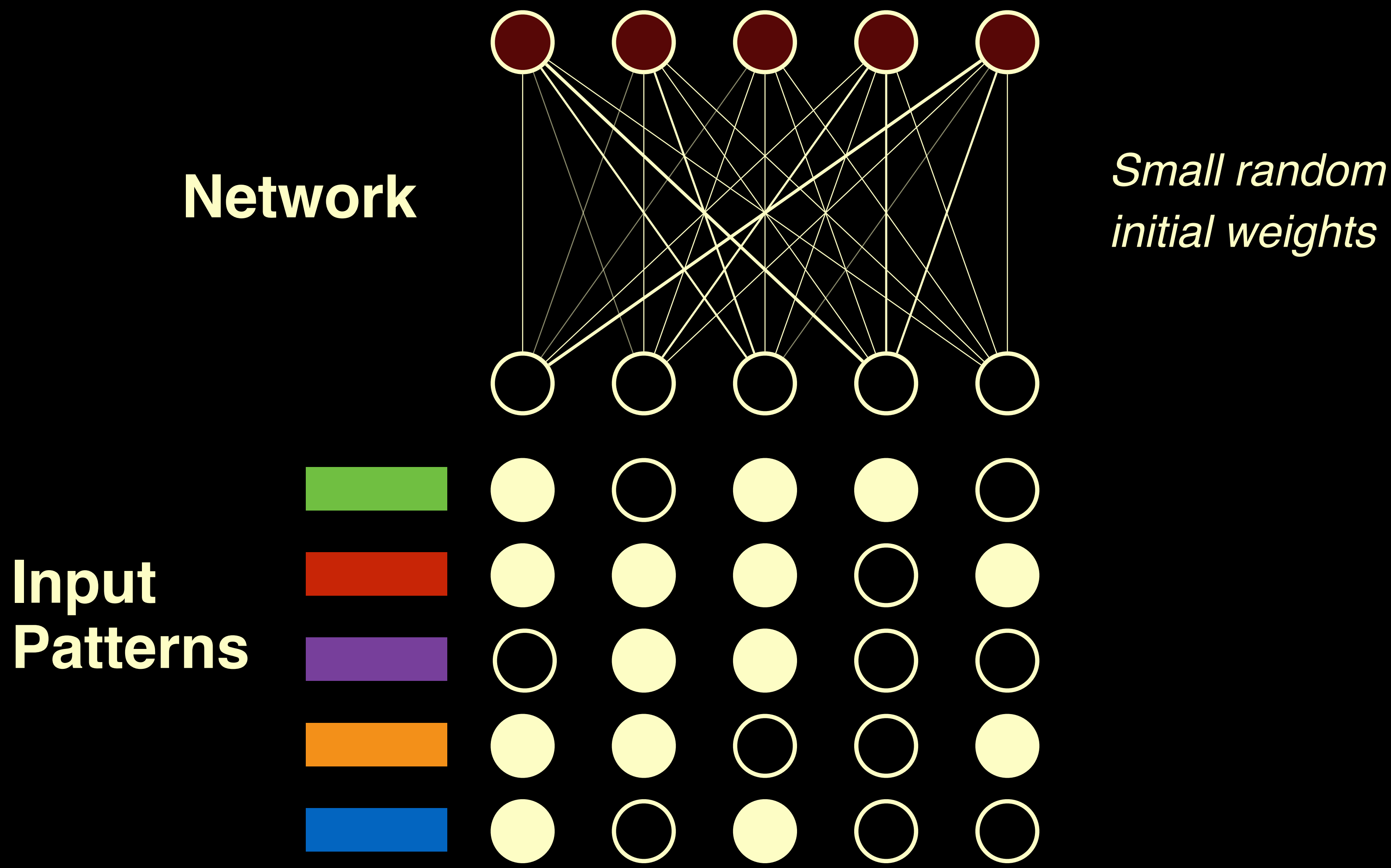
- **Learning rule:**
  - **present input pattern, and identify *best matching* (most active) *output unit:***
    - **one with input *current weights closest to input pattern*** *("winner" of lateral competition)*
  - **adjust weights for that unit using following rule:**

$$W_b(t+1) = W_b(t) + c_{wb(t)} \cdot g(t) \cdot (I - W_b(t))$$

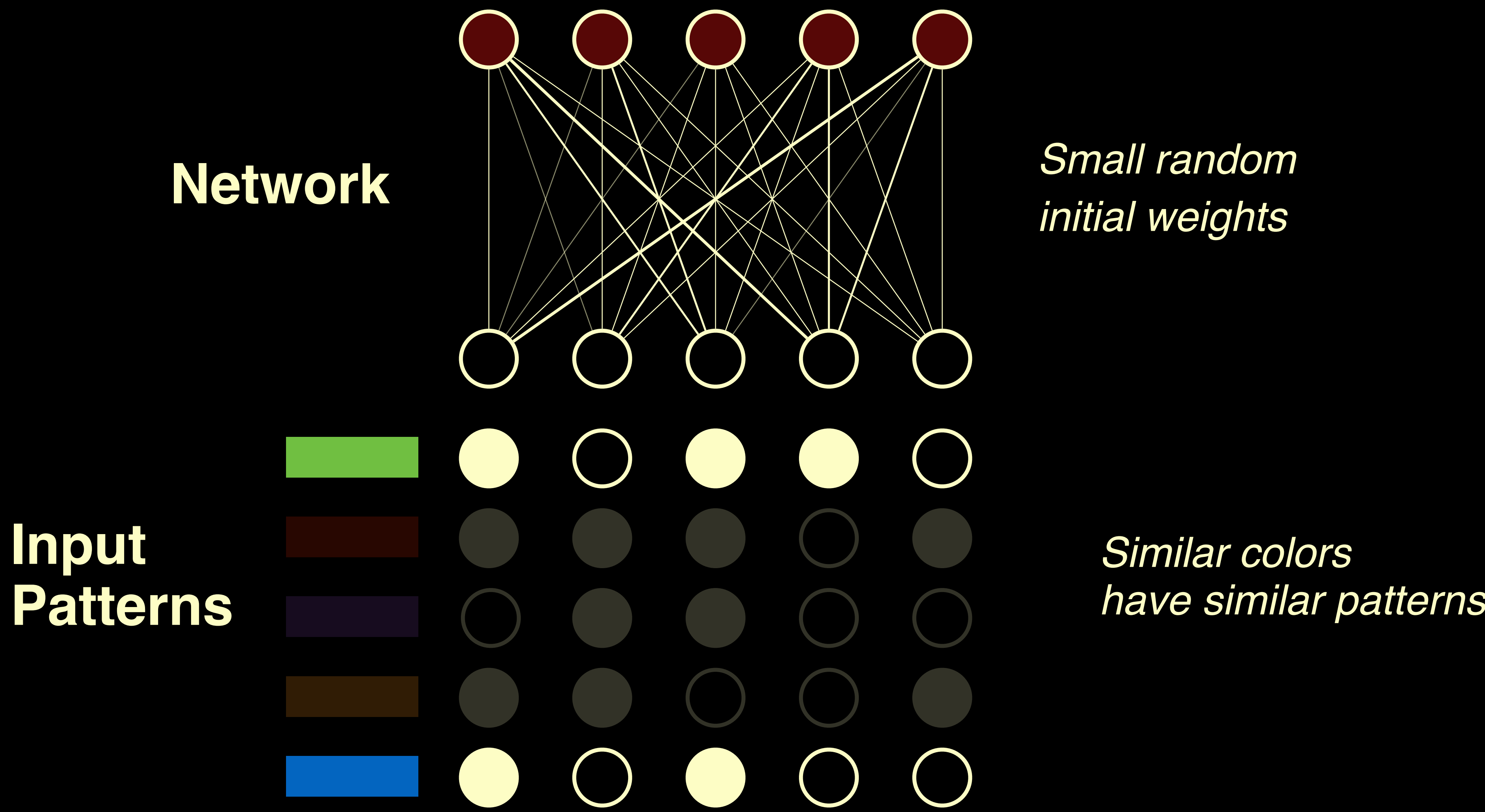
*change in weights to **b*** — *closeness to **b*** — *gain* — *difference from Input pattern*

$\propto$ correlation of output unit with pattern of activity over input units

Self-Organizing Maps

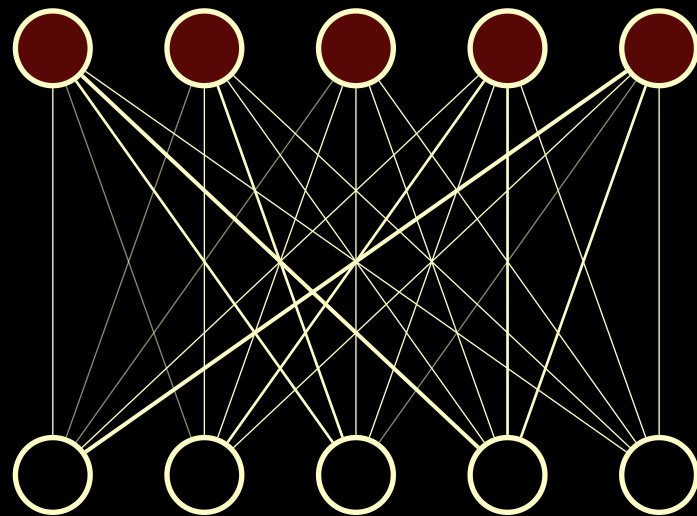


Self-Organizing Maps



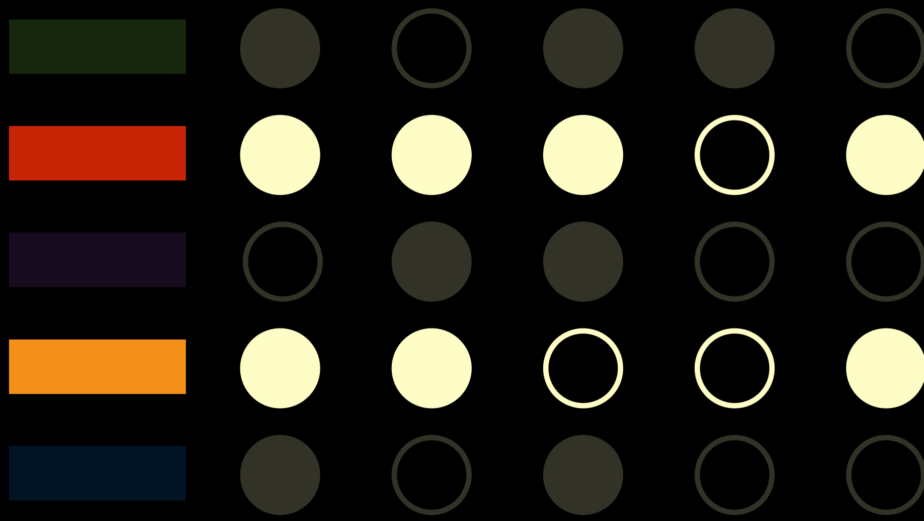
Self-Organizing Maps

Network



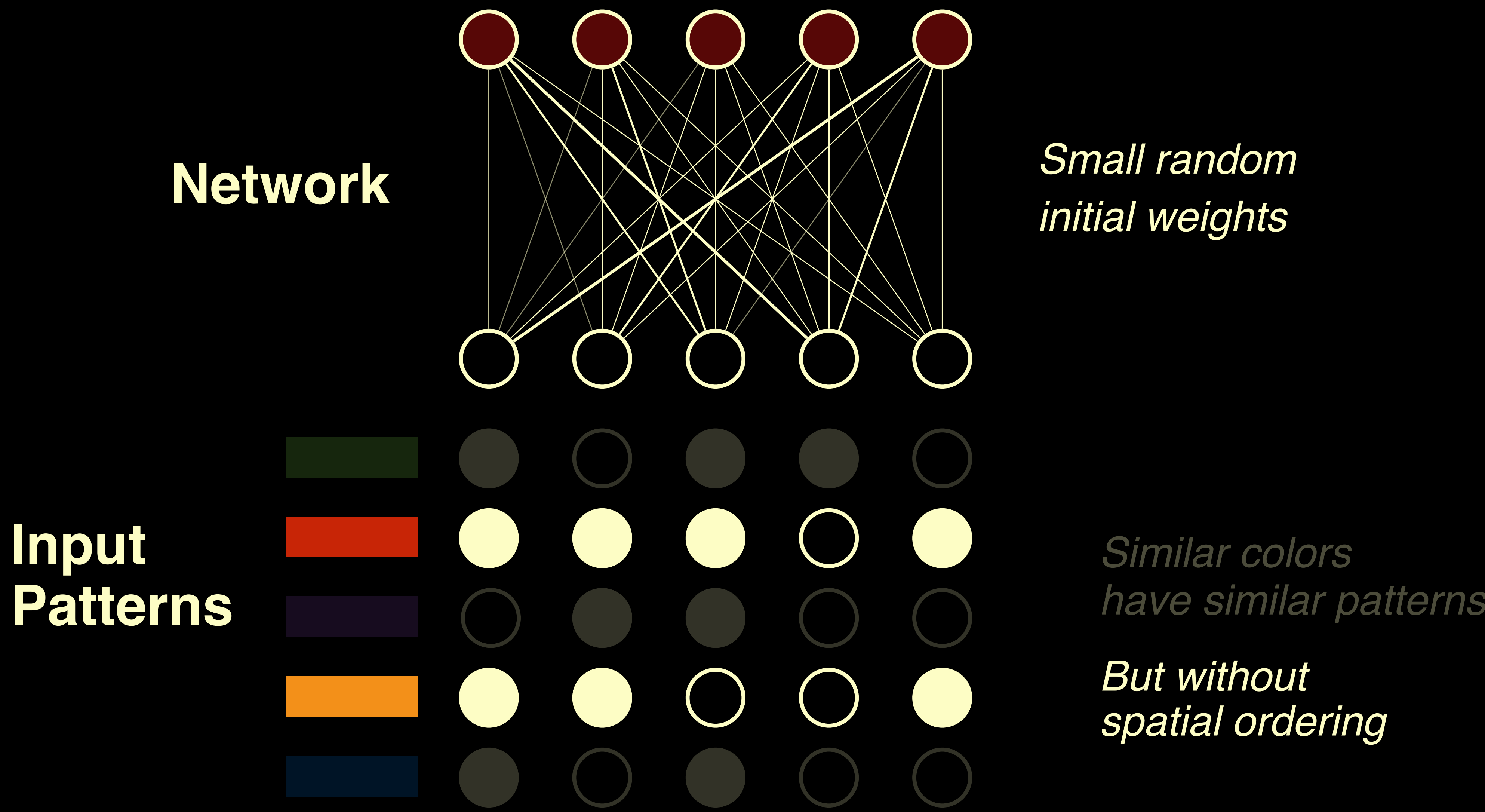
Small random initial weights

Input Patterns



Similar colors have similar patterns

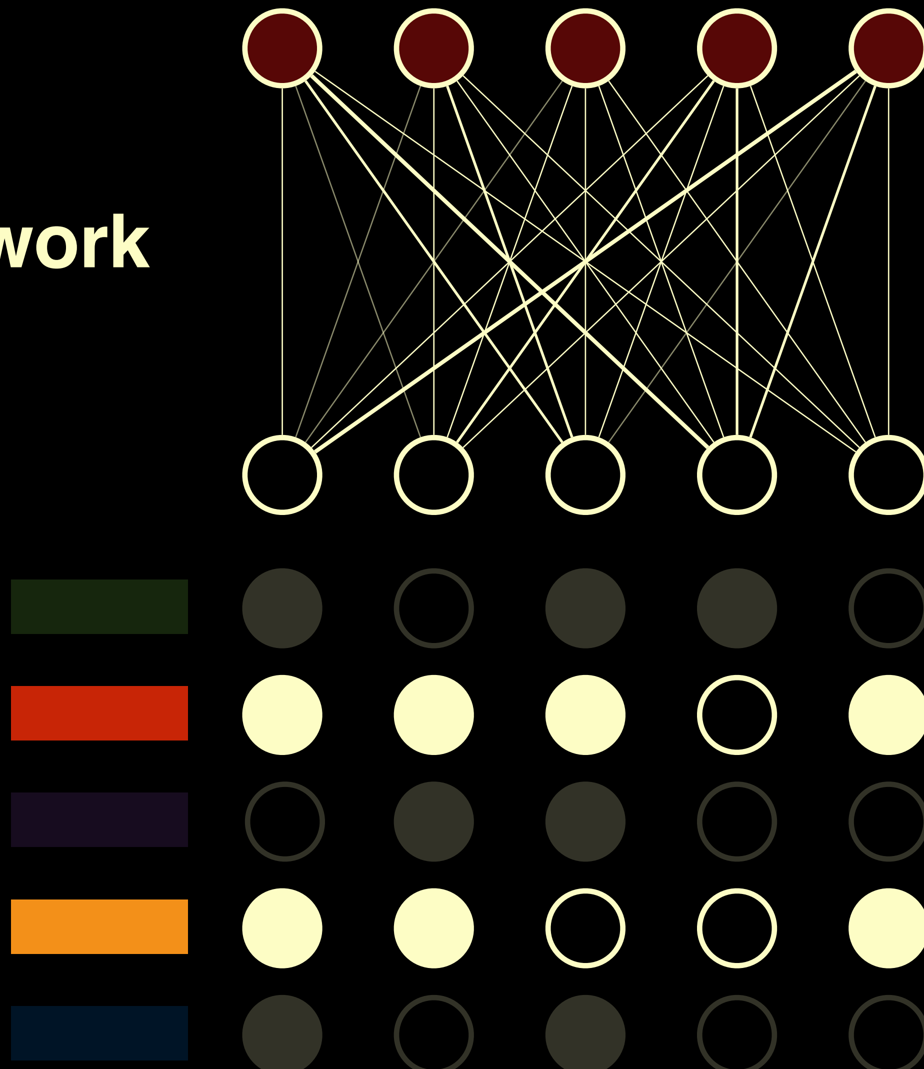
Self-Organizing Maps



Self-Organizing Maps

Network

Small random initial weights



Input Patterns

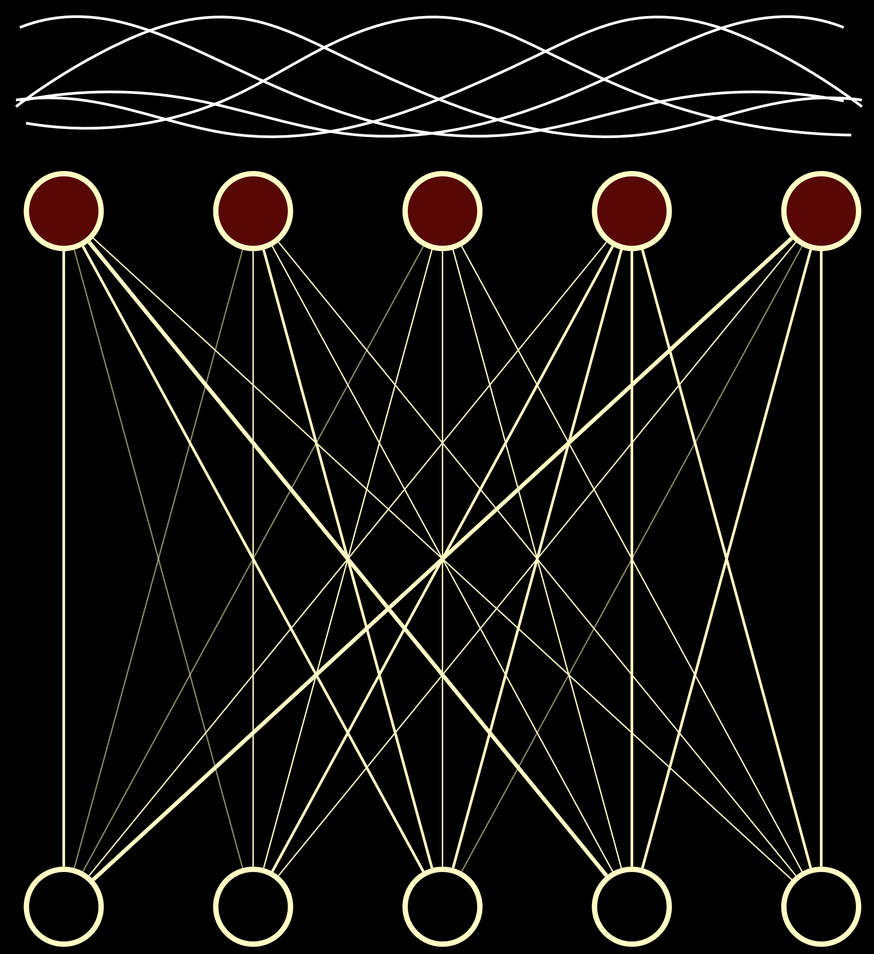
Similar colors have similar patterns

But without spatial ordering

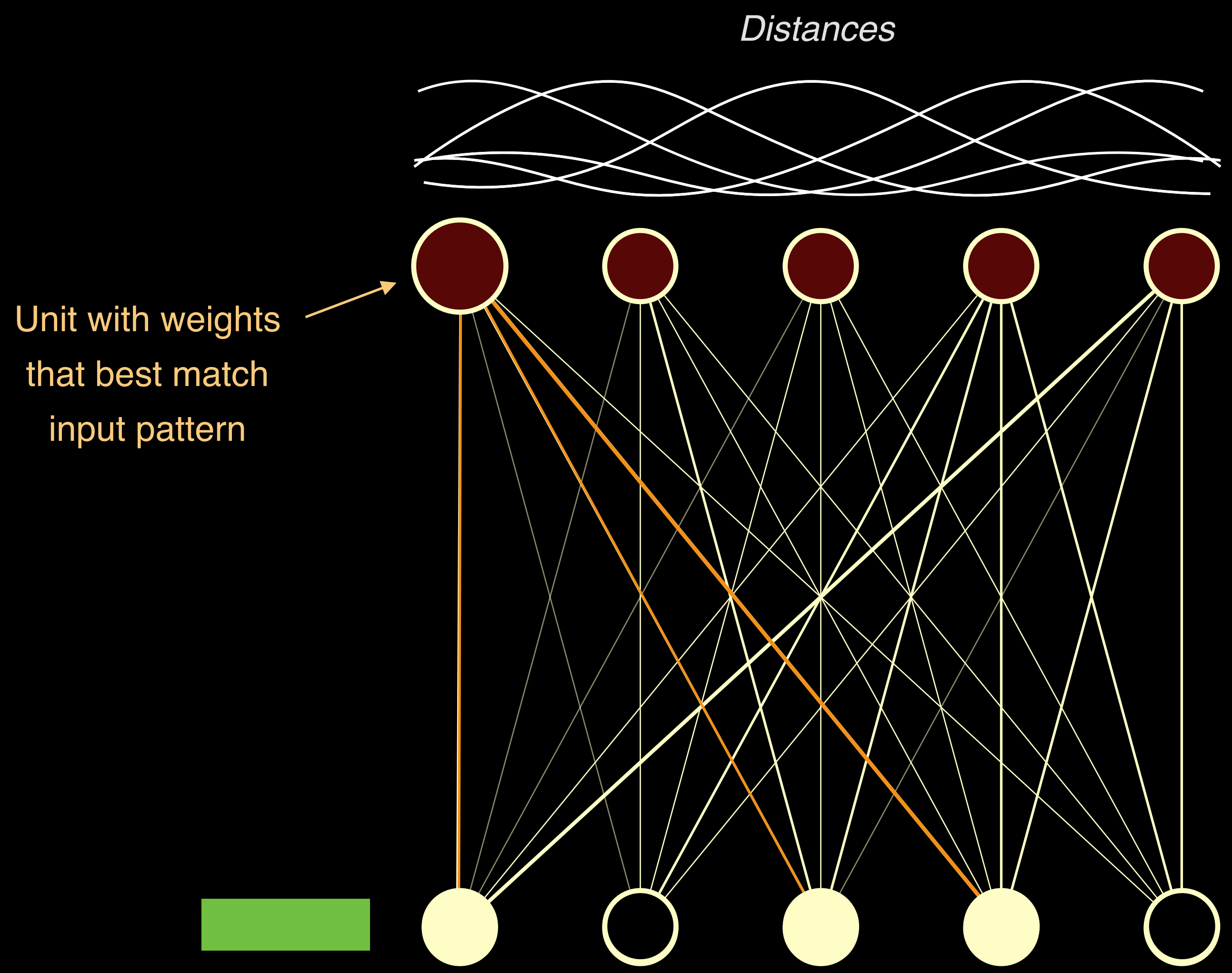
Demo

Self-Organizing Maps

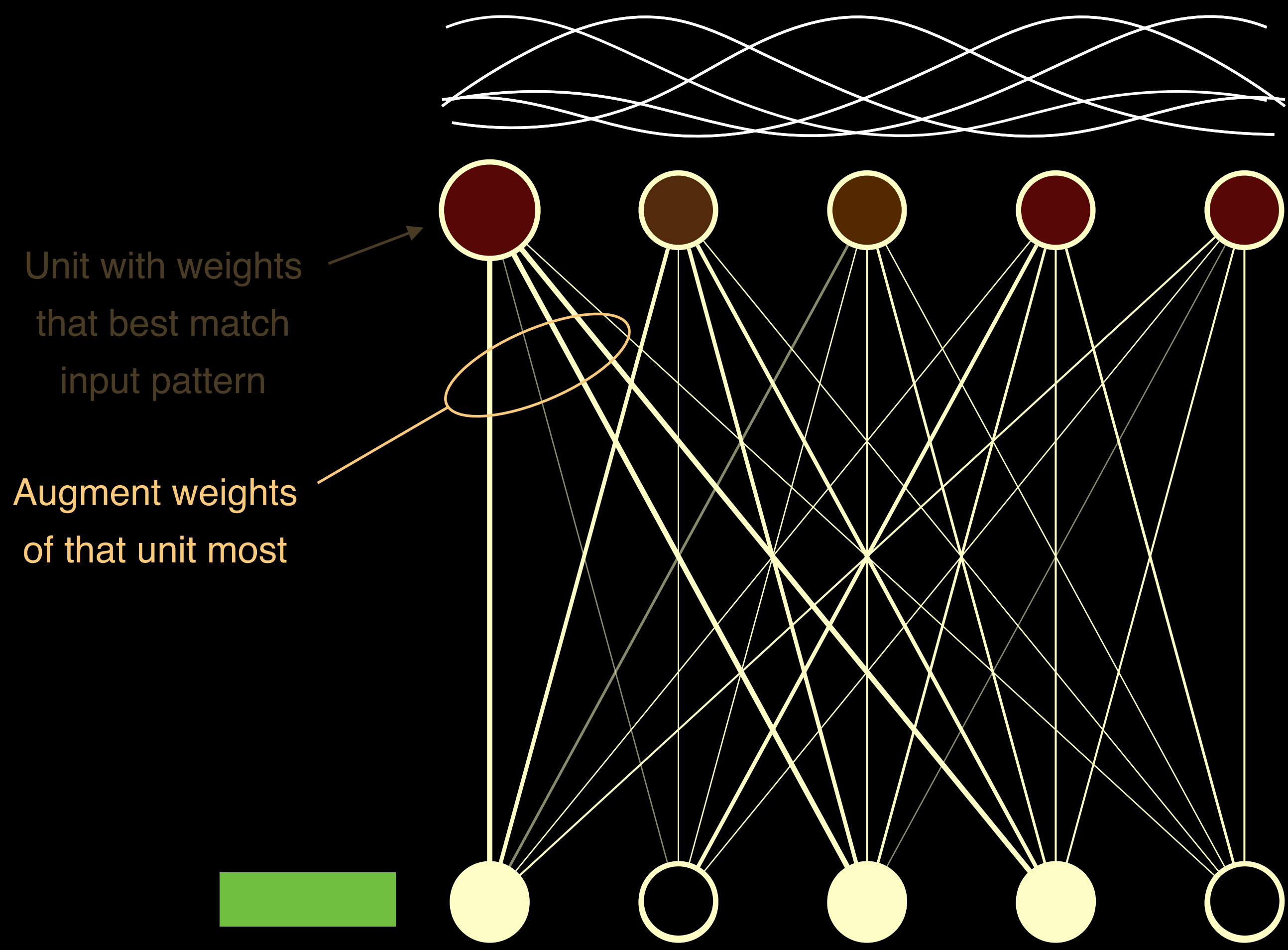
Distances



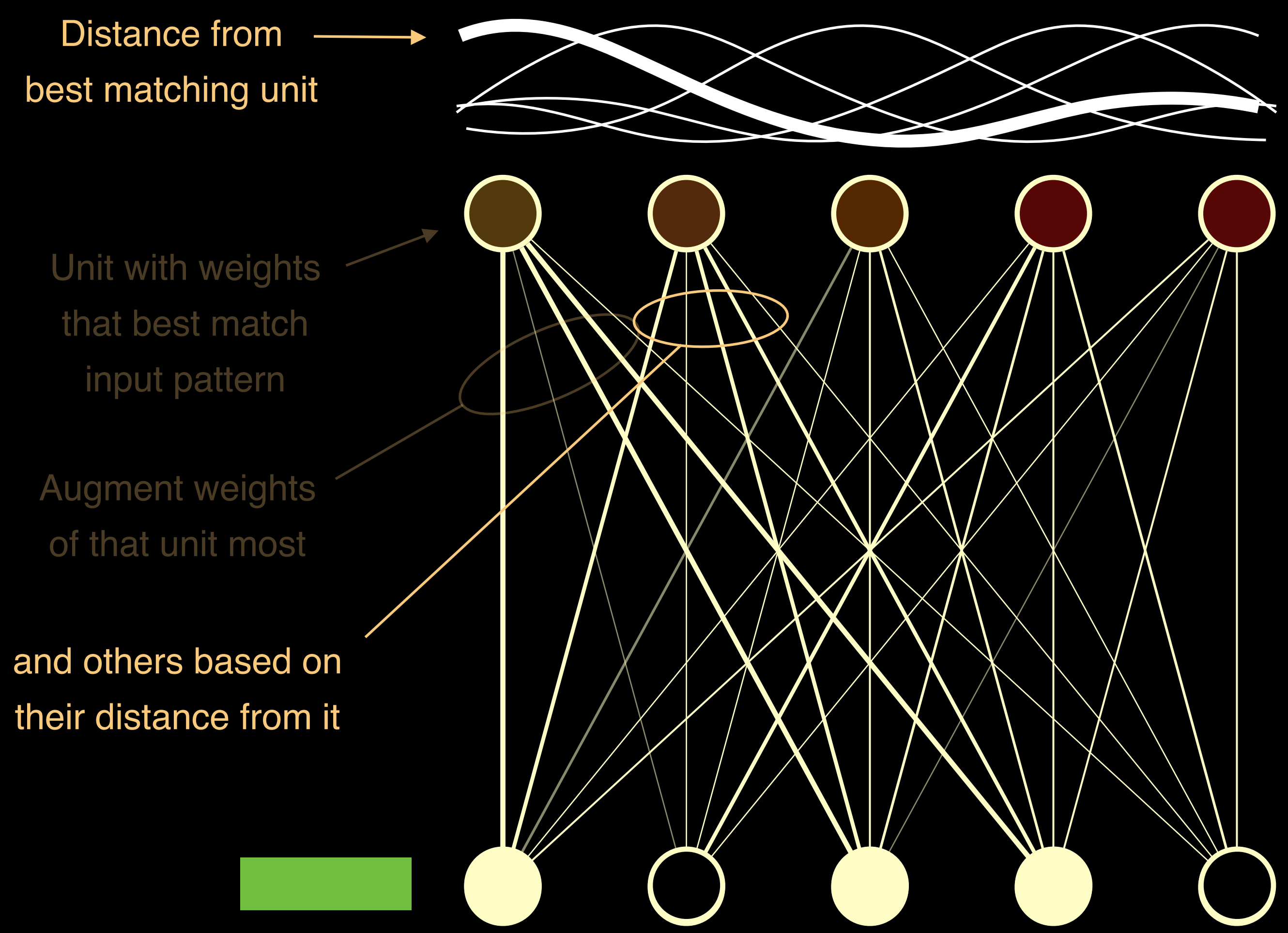
Self-Organizing Maps



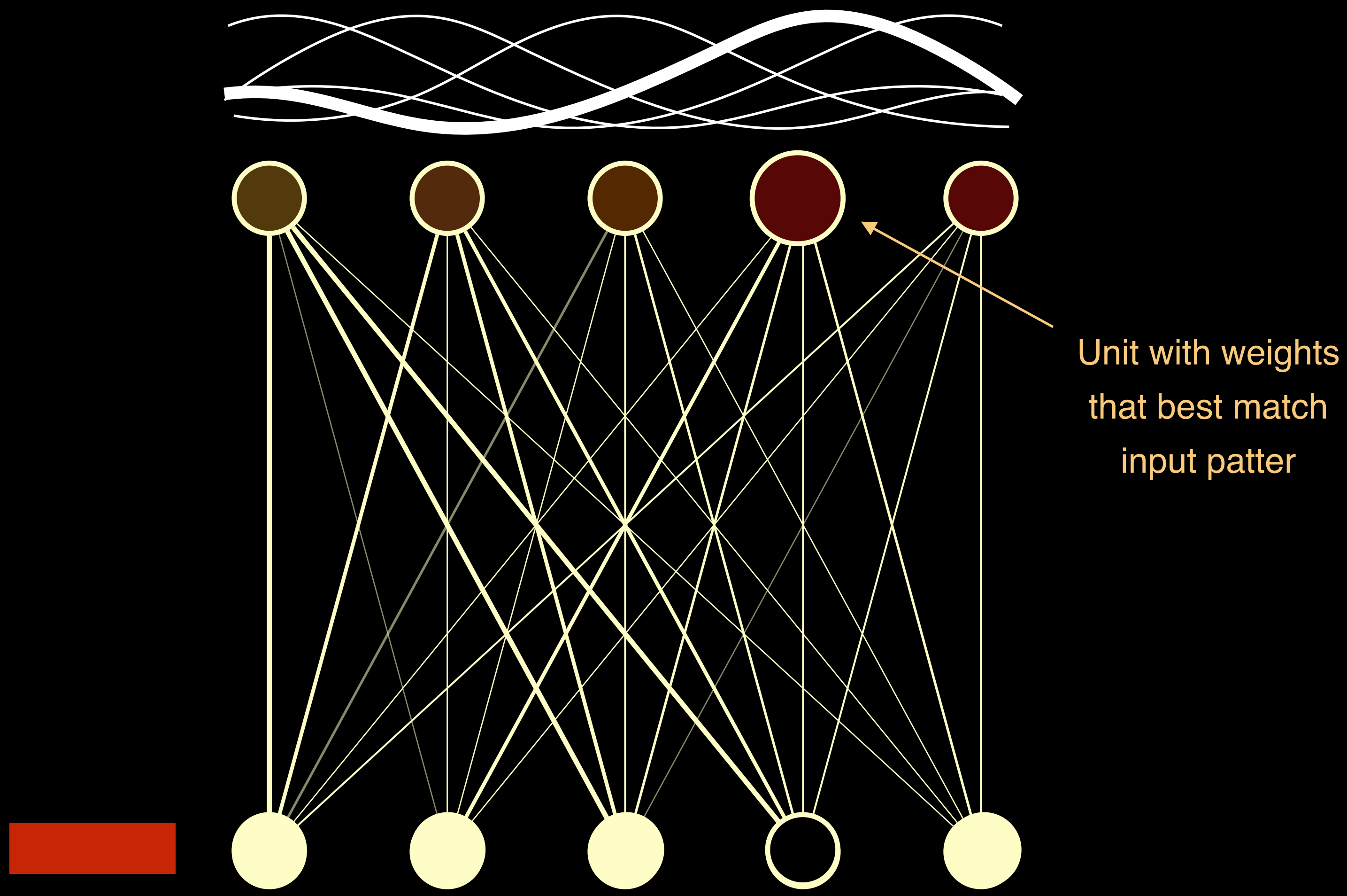
Self-Organizing Maps



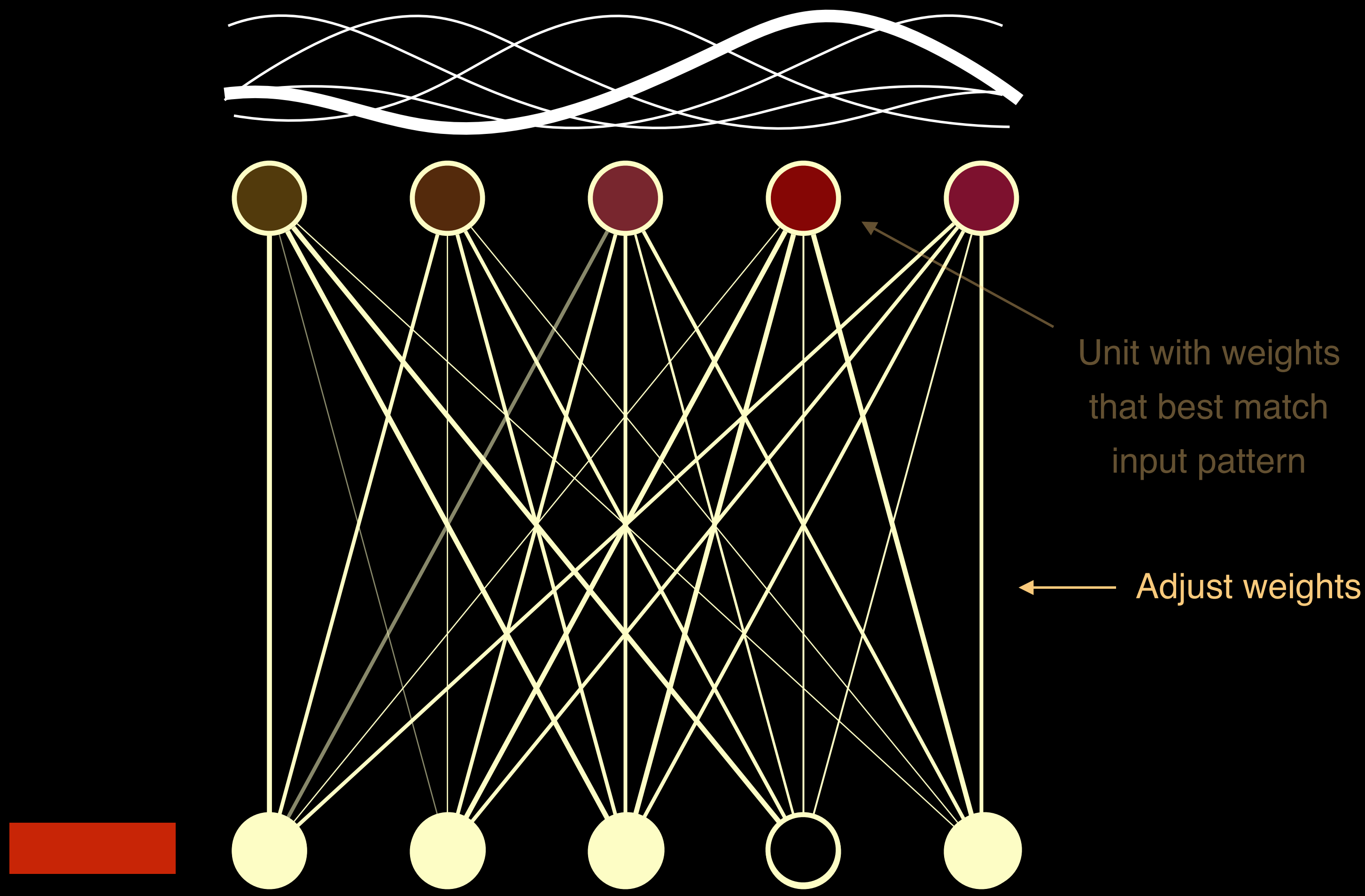
Self-Organizing Maps



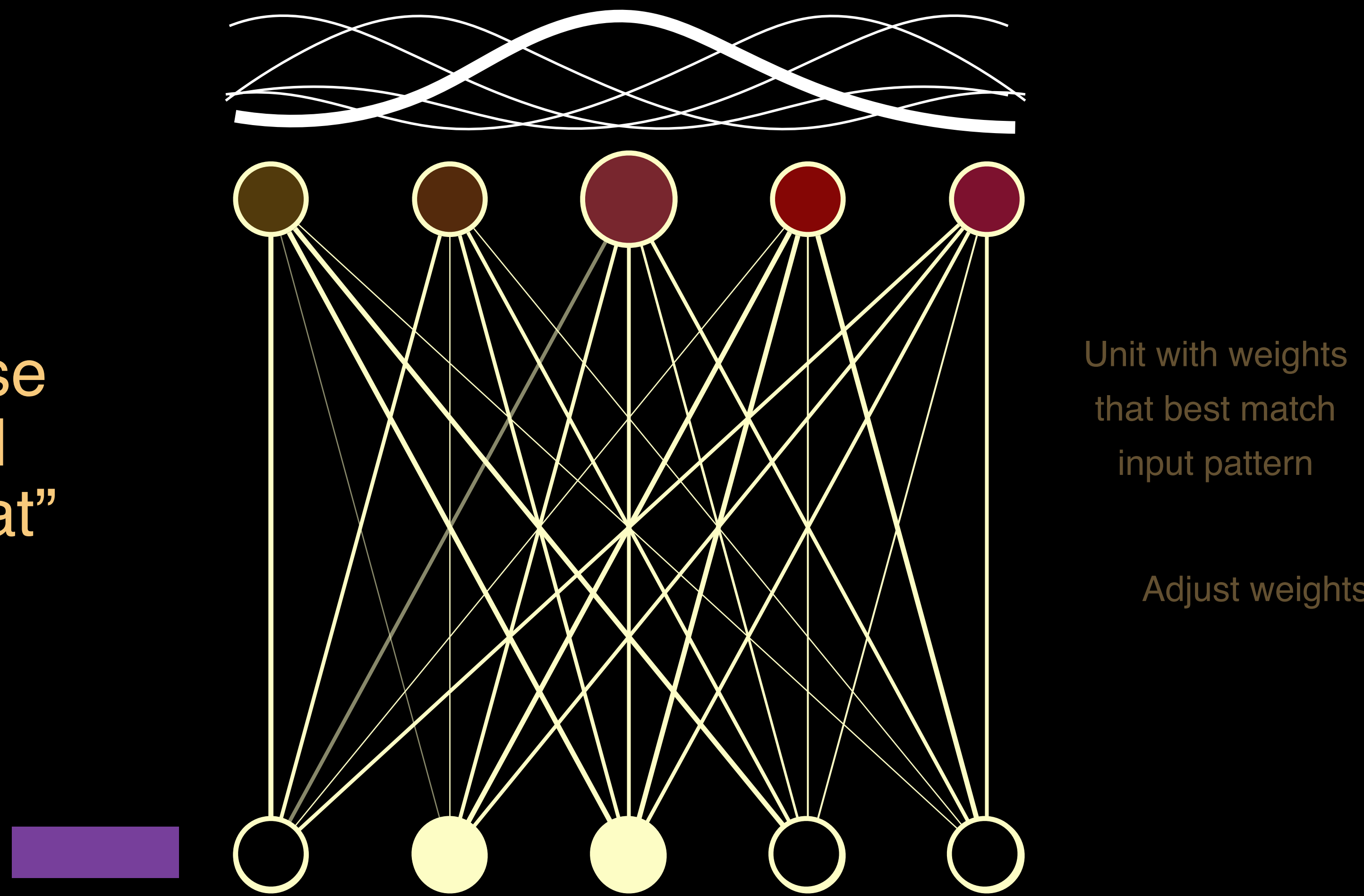
Self-Organizing Maps



Self-Organizing Maps

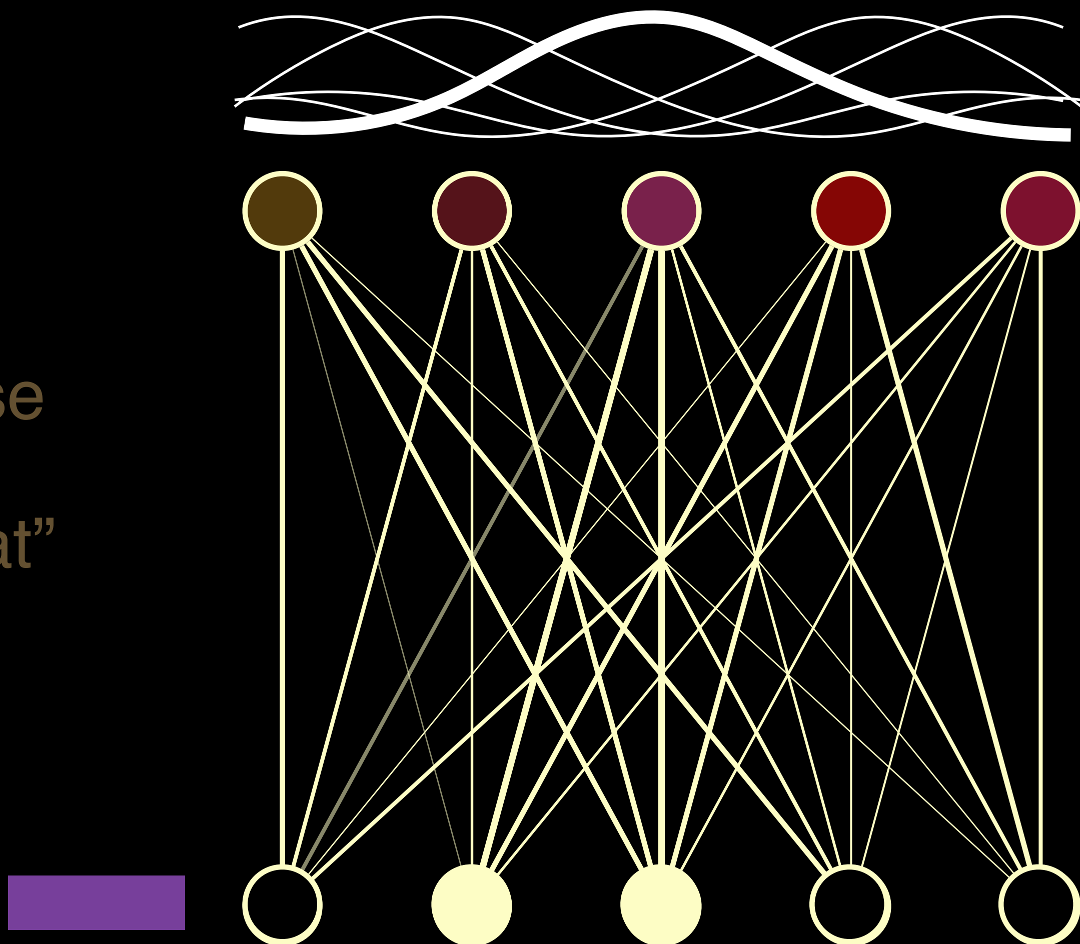


Self-Organizing Maps



"Rinse and repeat"

Self-Organizing Maps

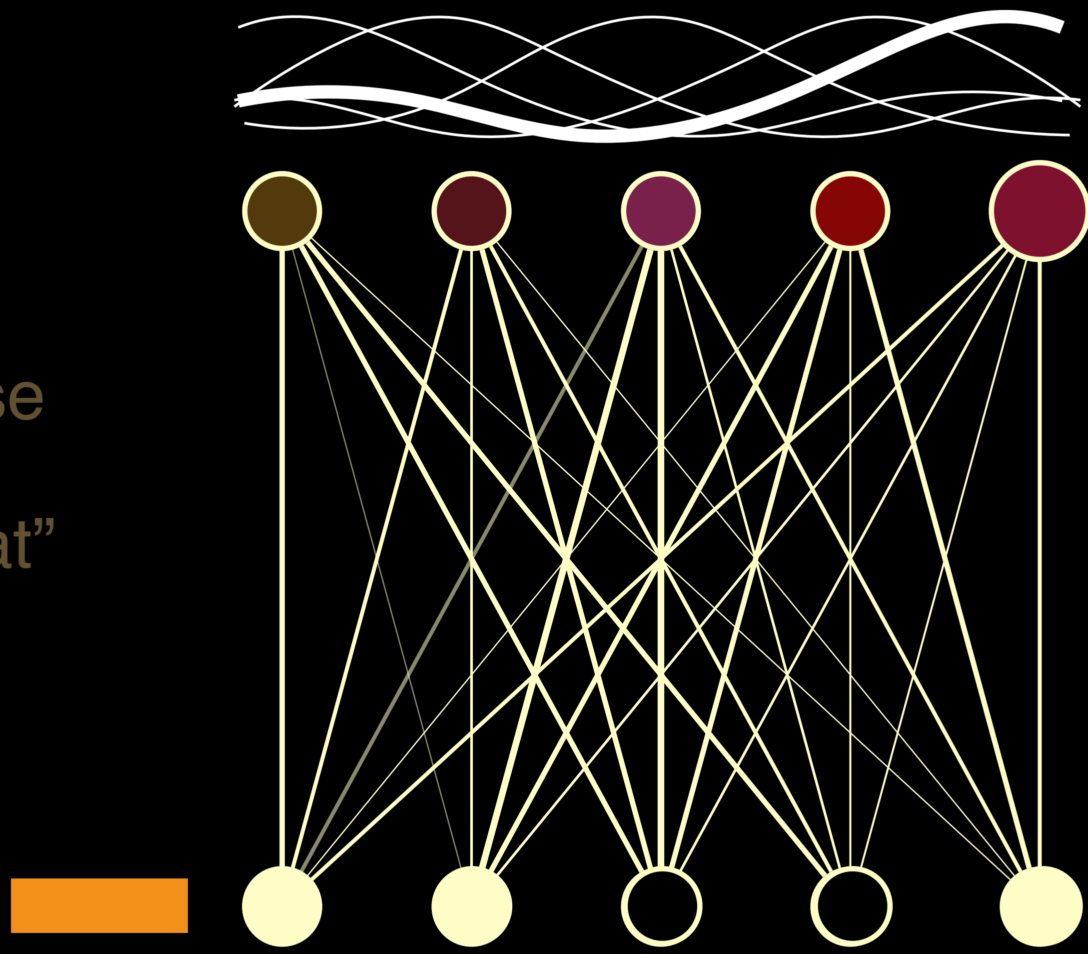


“Rinse and repeat”

Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps

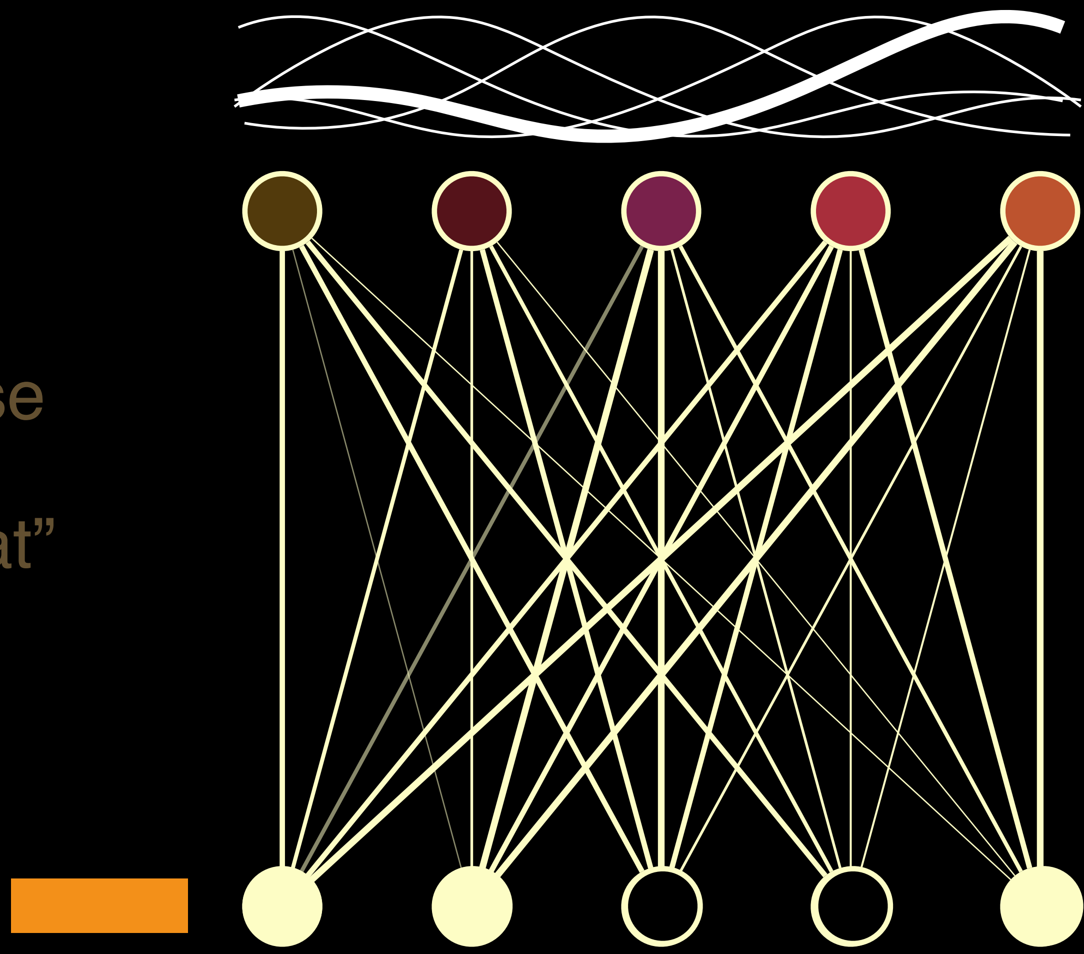


“Rinse and repeat”

Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps



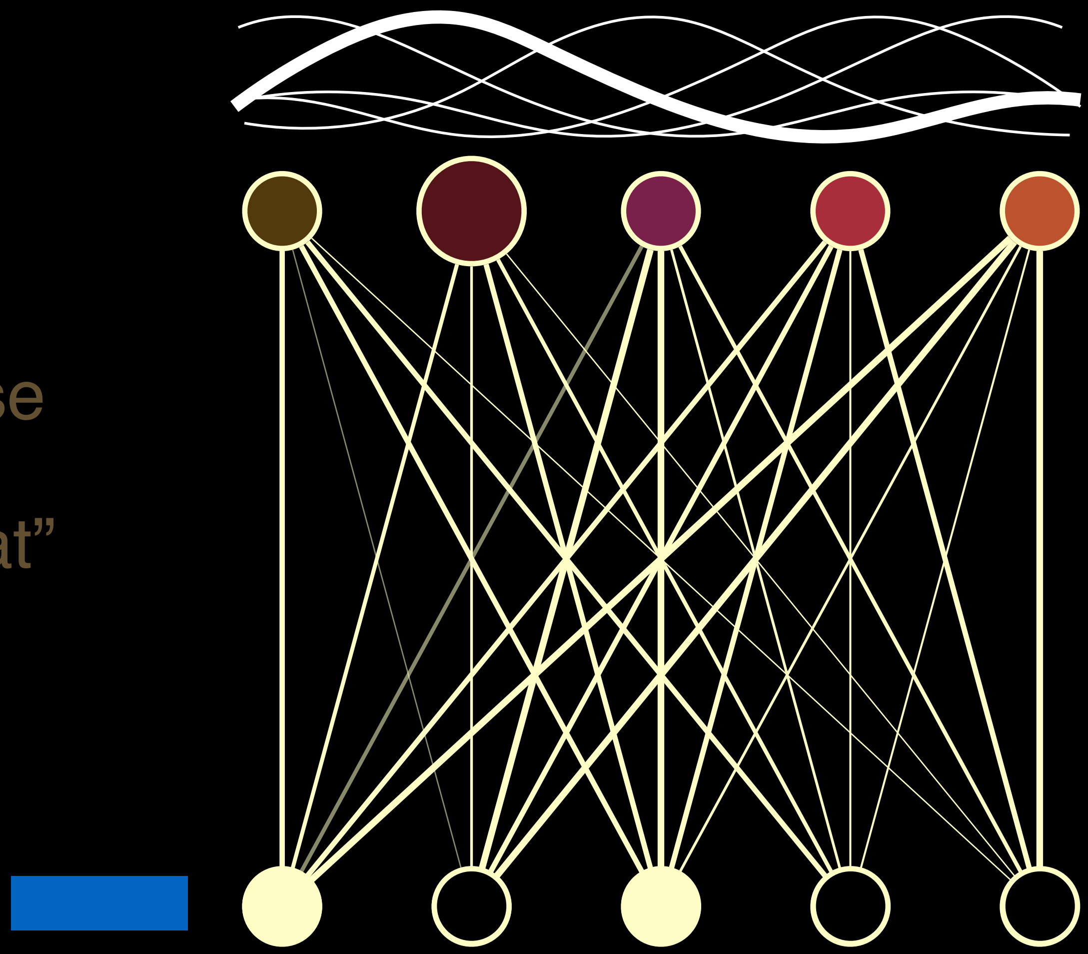
“Rinse and repeat”

Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps

“Rinse and repeat”

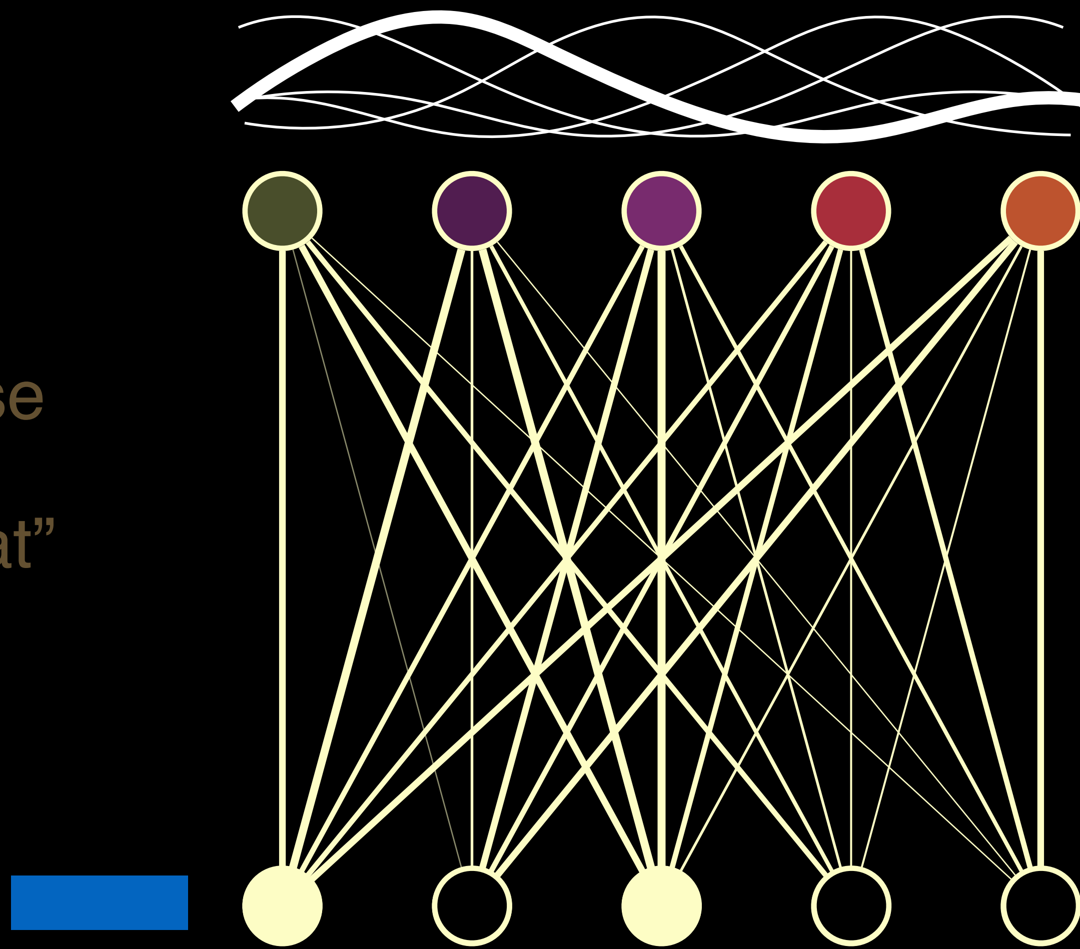


Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps

"Rinse and repeat"

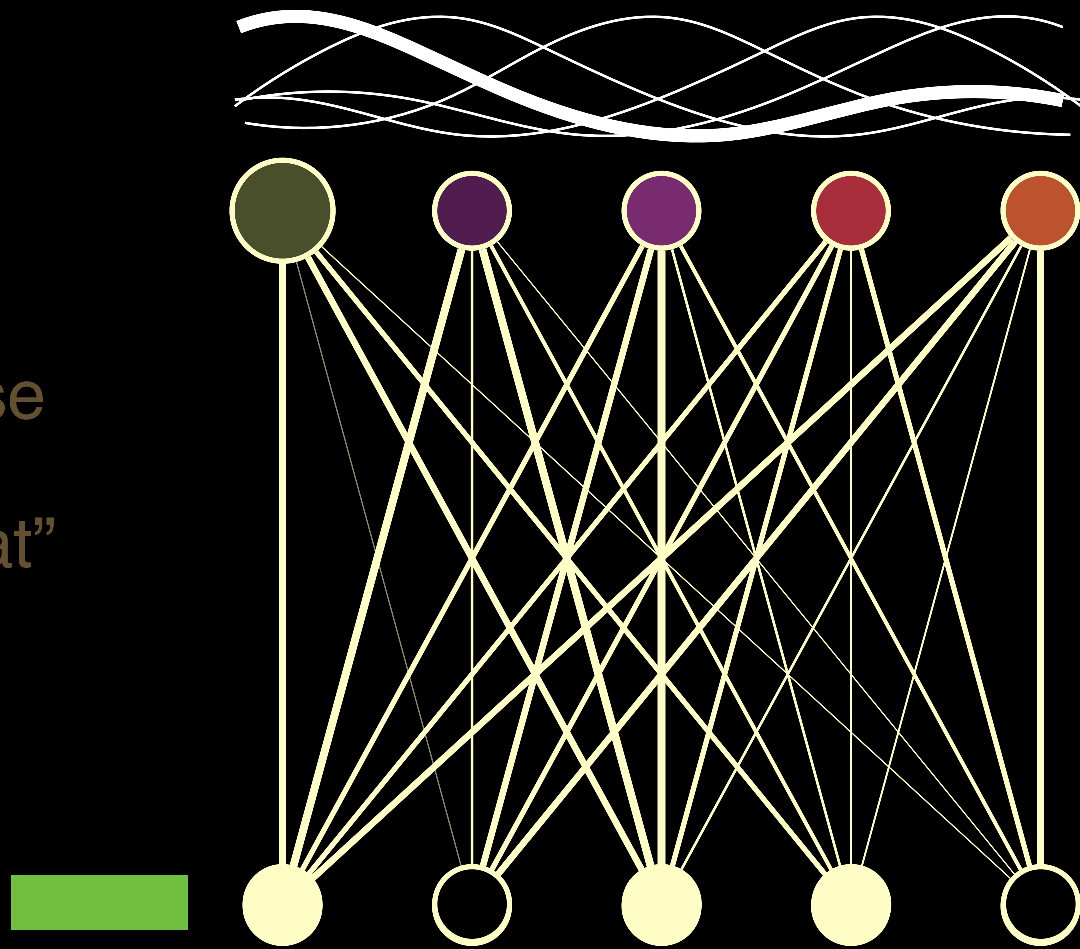


Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps

“Rinse and repeat”

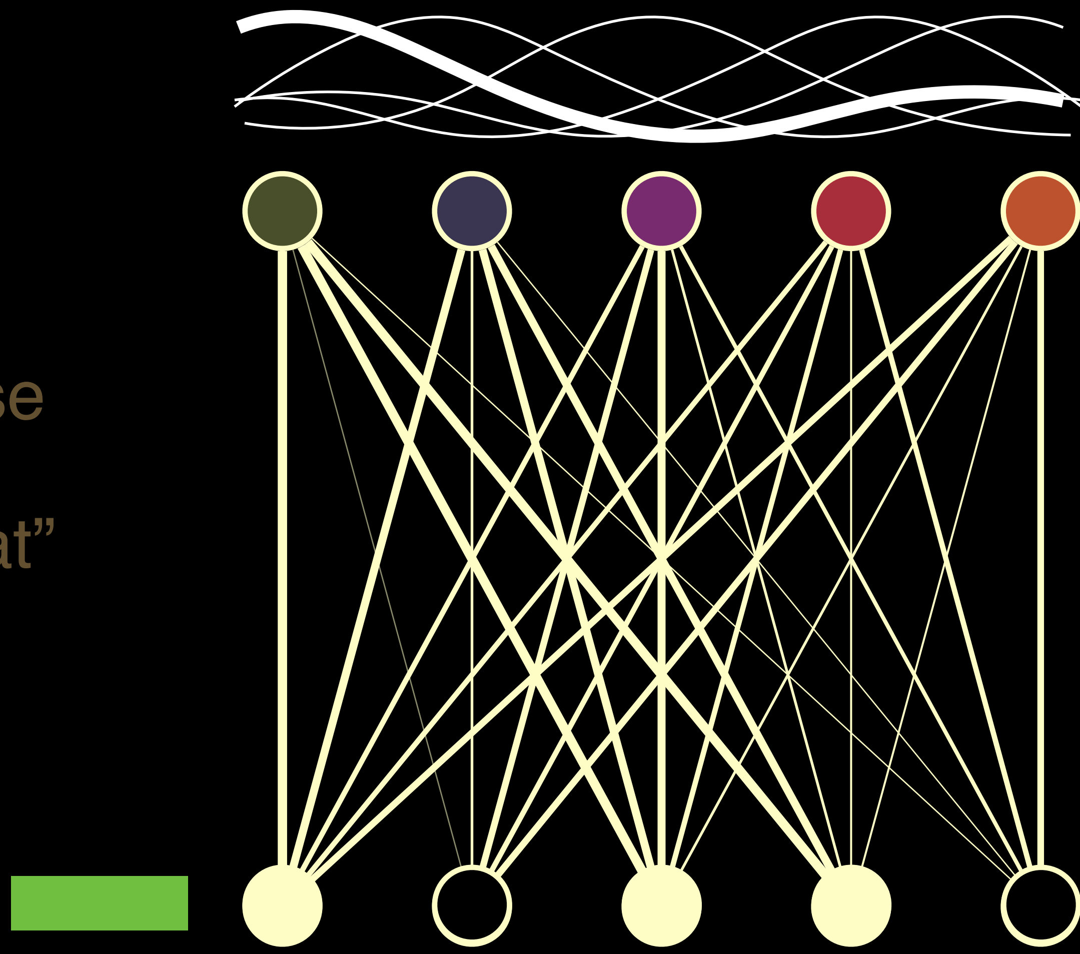


Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps

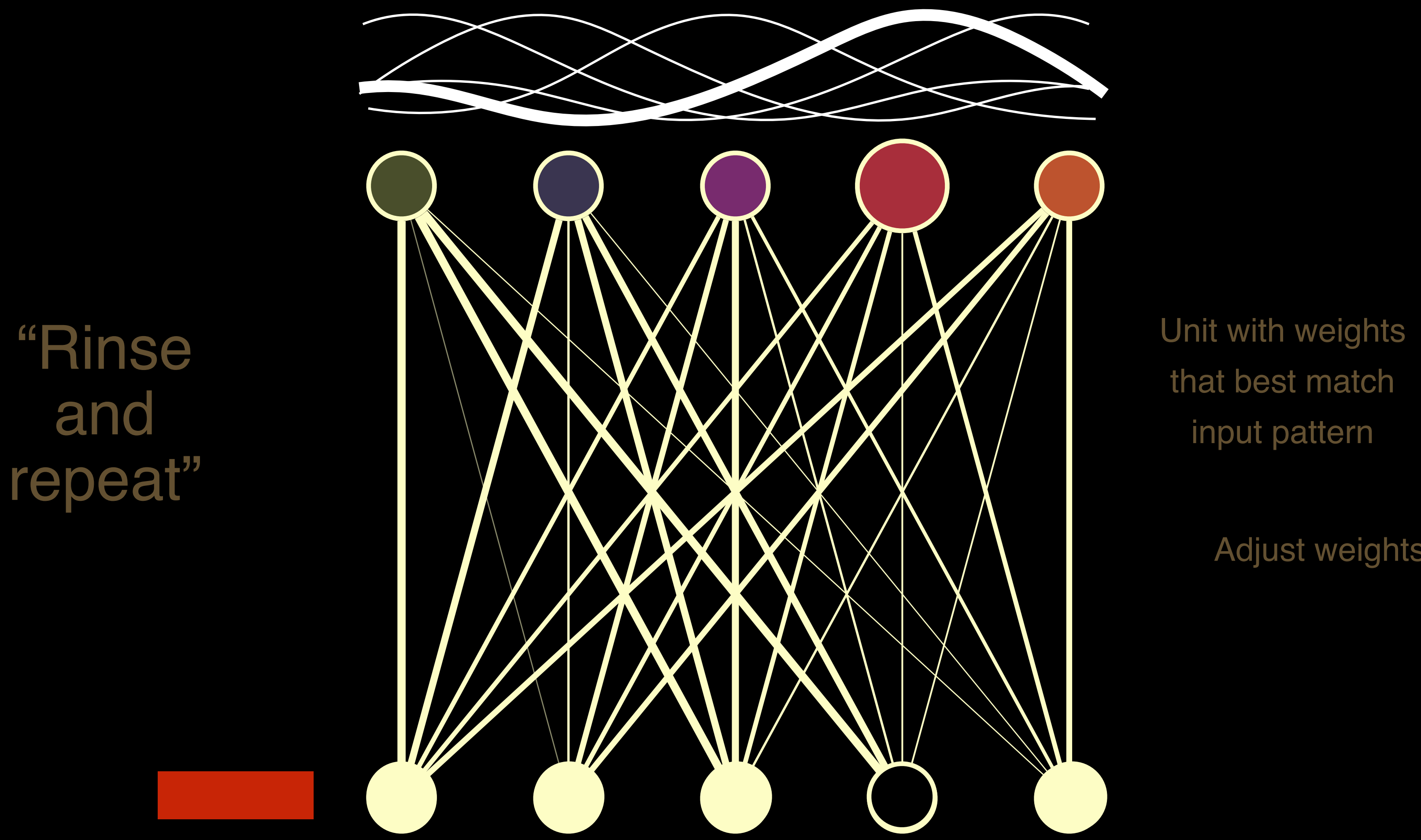
“Rinse and repeat”



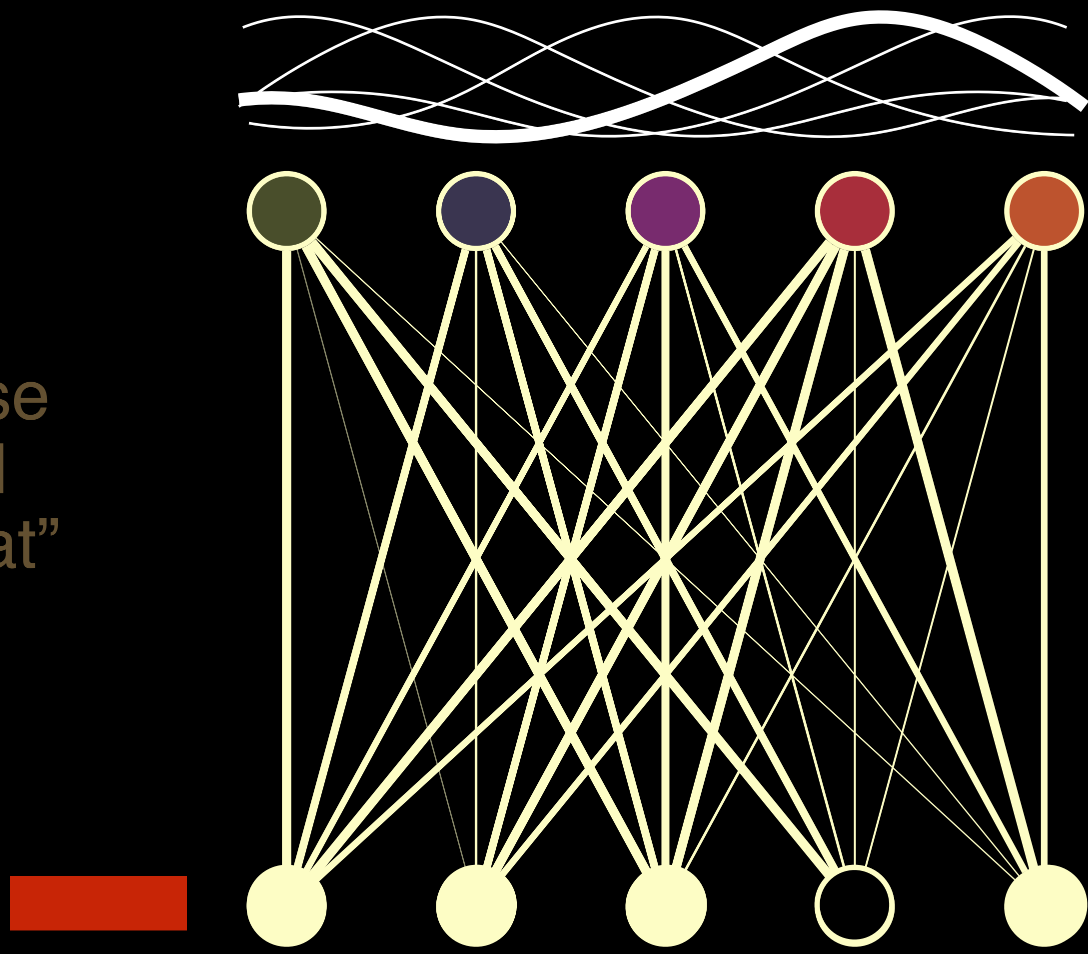
Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps



Self-Organizing Maps



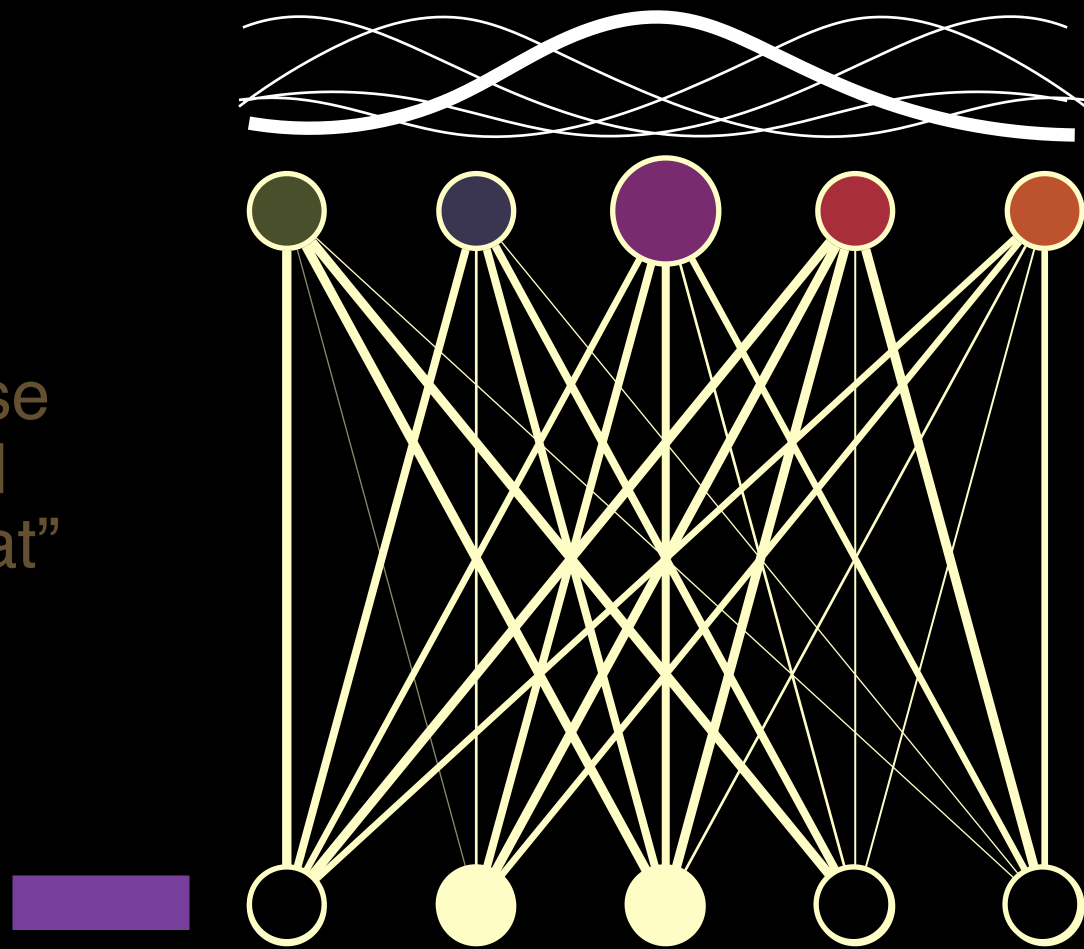
“Rinse and repeat”

Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps

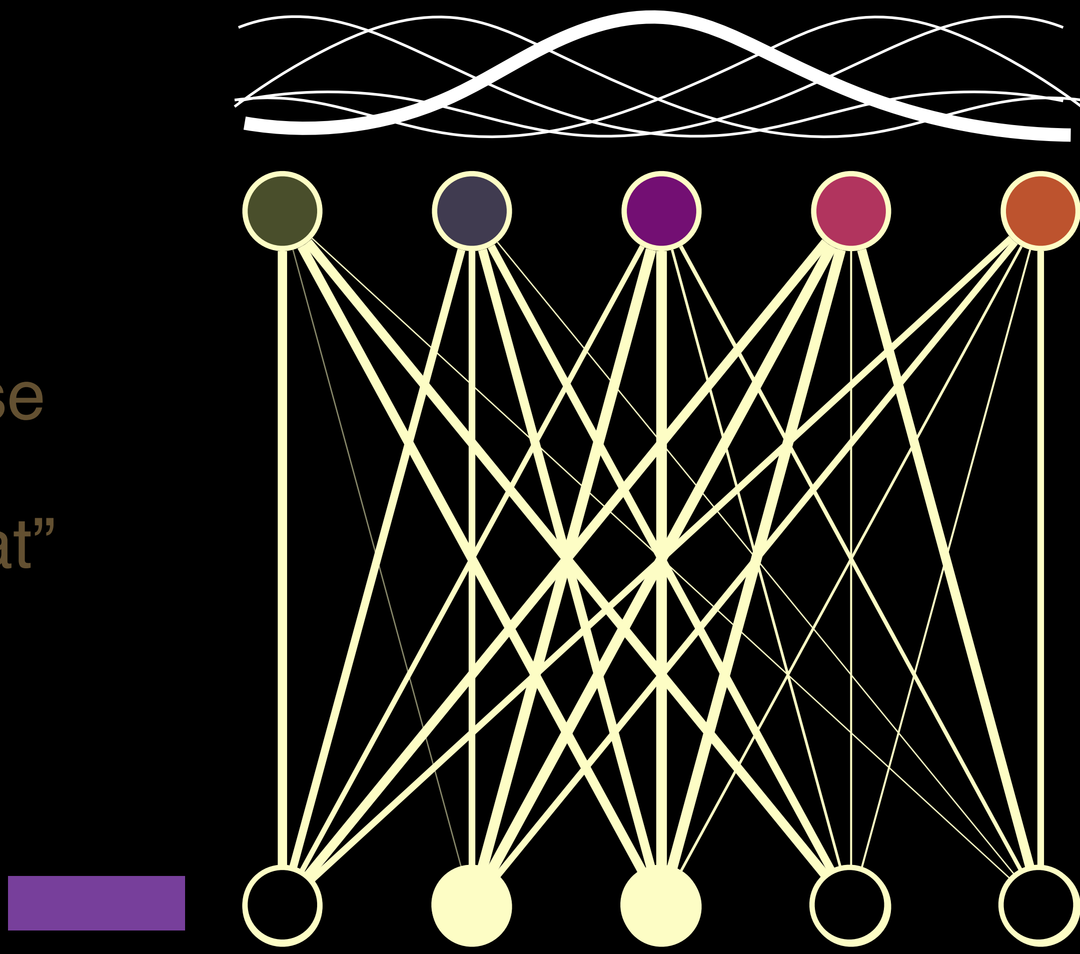
“Rinse and repeat”



Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps

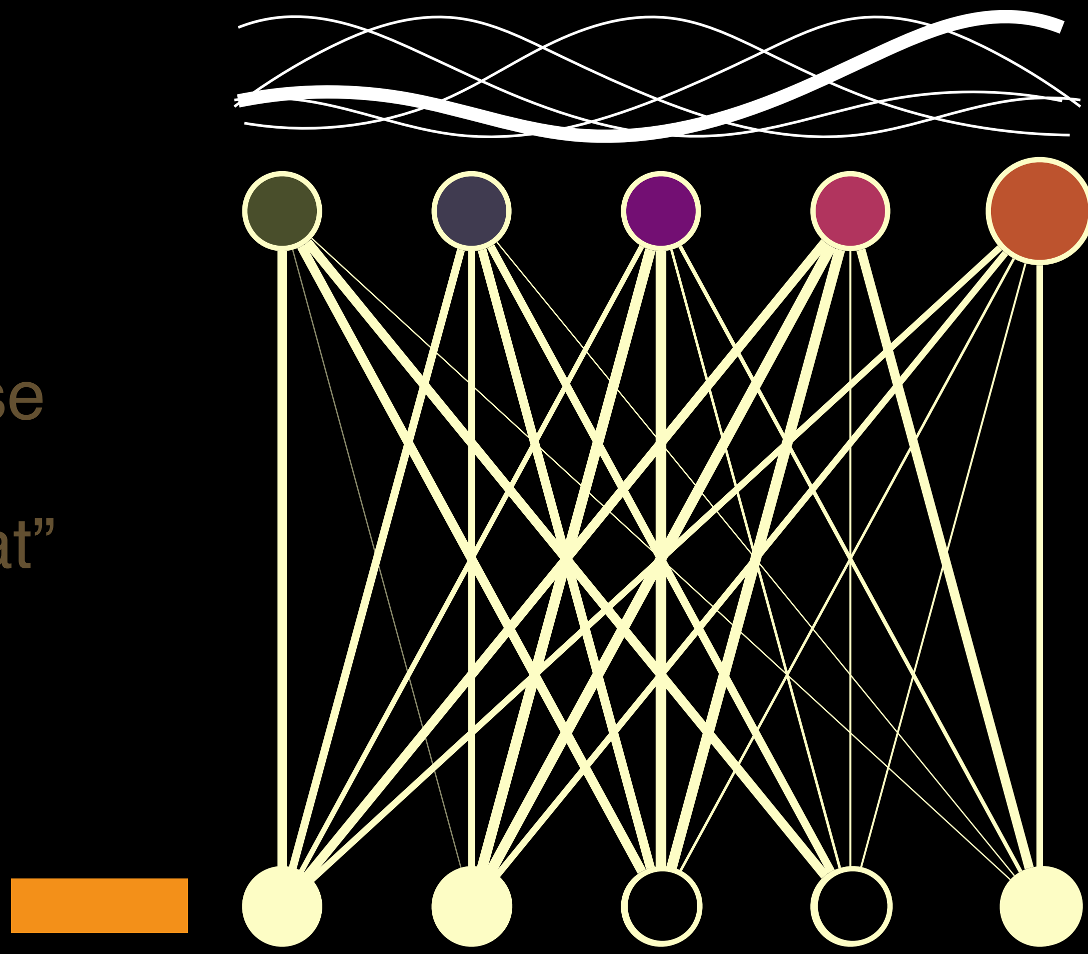


“Rinse and repeat”

Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps

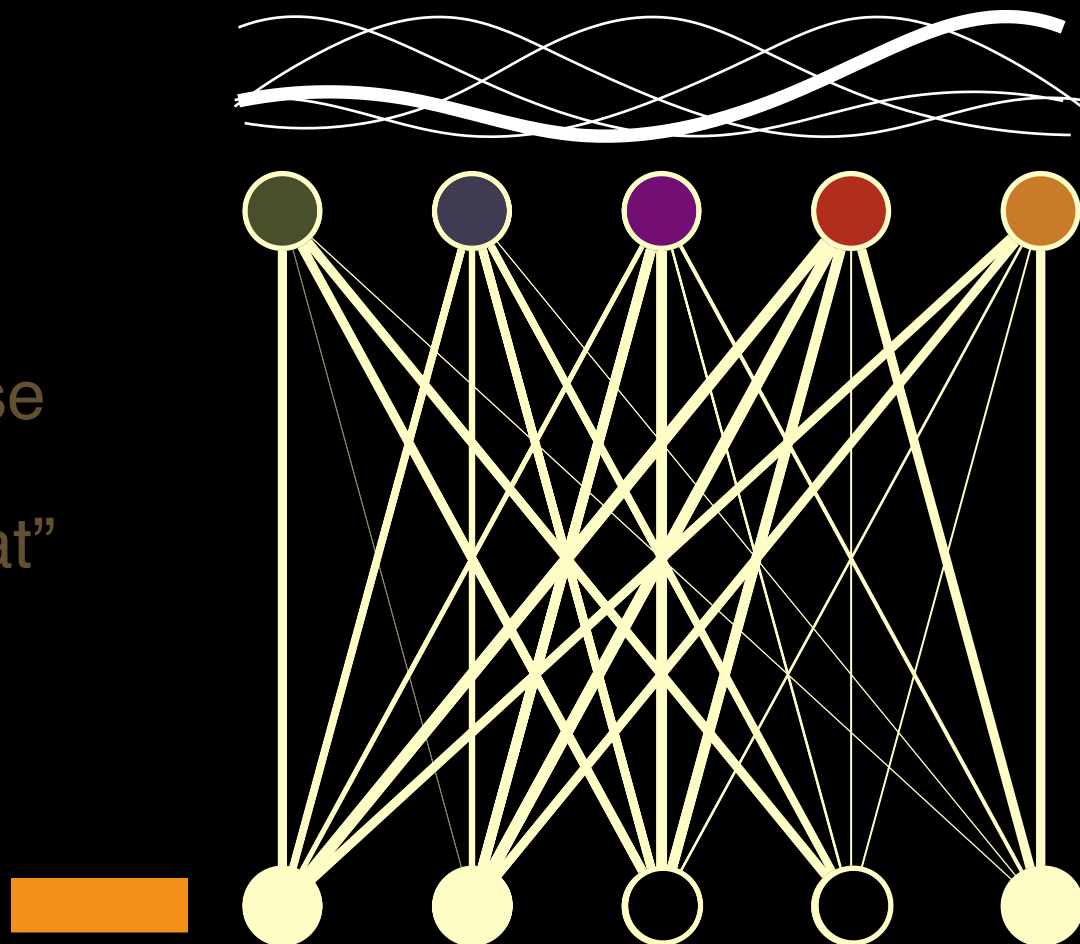


“Rinse and repeat”

Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps



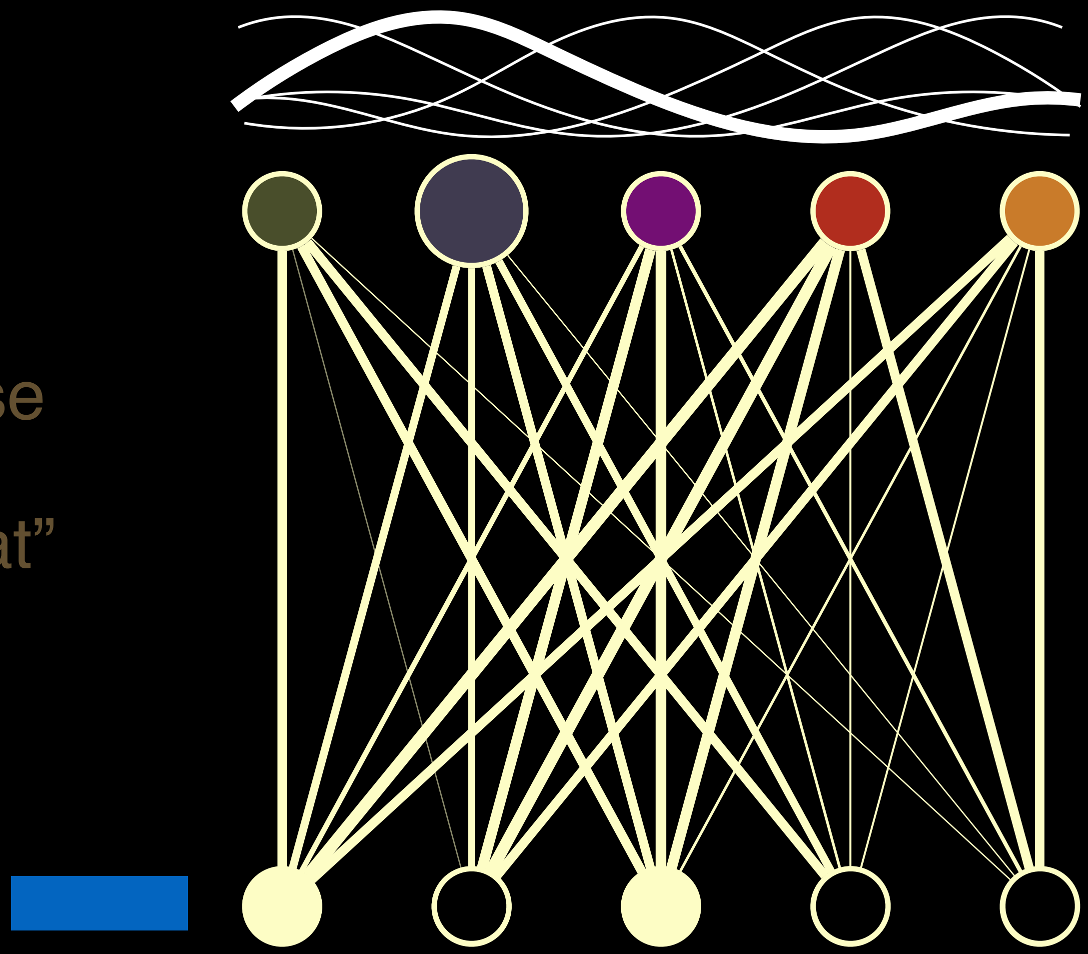
“Rinse and repeat”

Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps

“Rinse and repeat”



Unit with weights that best match input pattern

Adjust weights

Self-Organizing Maps

Spectrally arranged!