Longterm Memory: Distributed Representation and Semantics

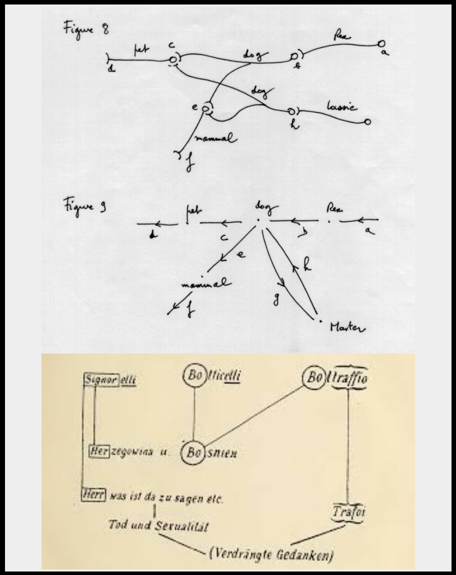
Longterm (Semantic) Memory

Longterm (Semantic) Memory

- Processing in longterm memory:
  - activation of representations in a semantic network

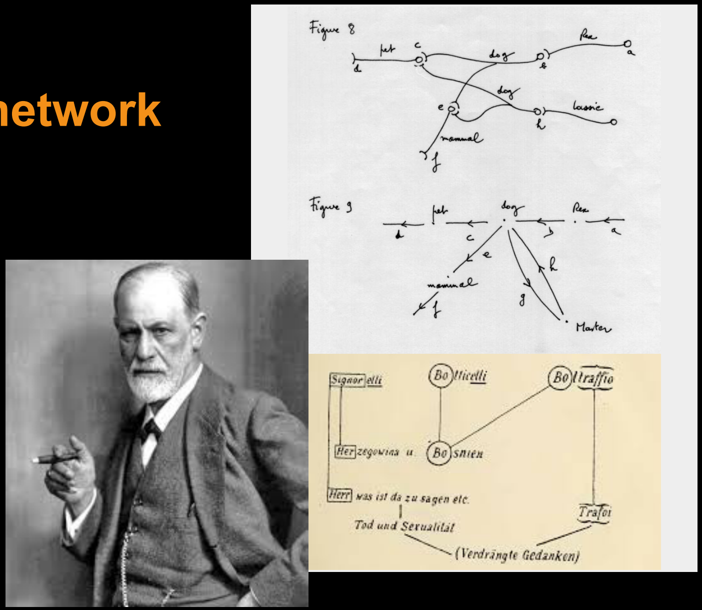
Longterm (Semantic) Memory

- **Processing in longterm memory:**
  - **activation of representations in a semantic network**



Longterm (Semantic) Memory

- **Processing in longterm memory:**
  - **– activation of representations in a semantic network**

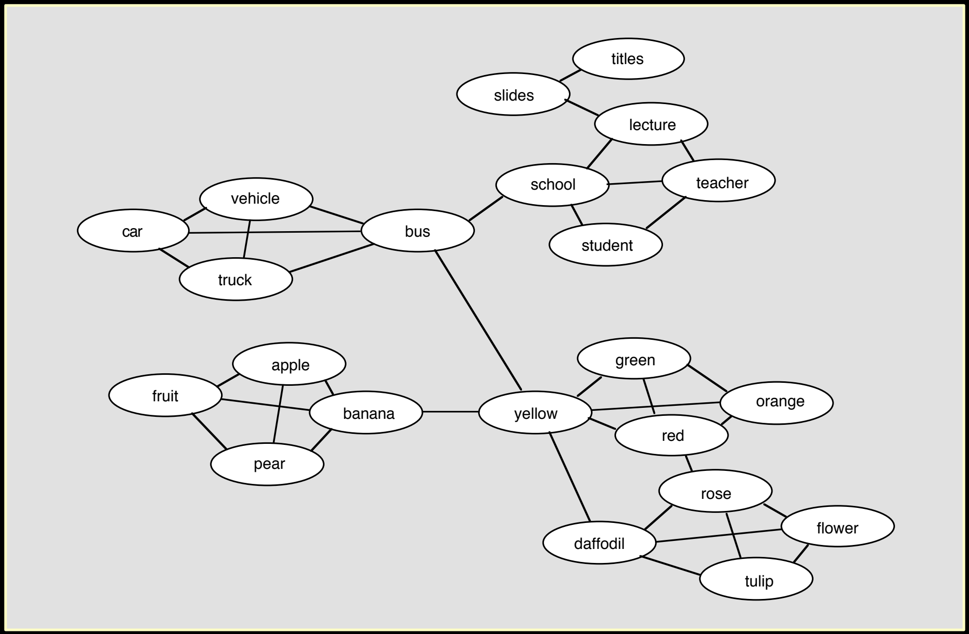


Project for a Scientific Psychology, 1895

Longterm (Semantic) Memory

- **Processing in longterm memory:**
  - **activation of representations in a semantic network**
  - **spread of activation to related items**

    (Quillian, 1967; Collins and Loftus, 1975)



Longterm (Semantic) Memory

- Processing in longterm memory:
  - activation of representations in a semantic network
  - spread of activation to related items
    (Quillian, 1967; Collins and Loftus, 1975)

- Evidence: lexical priming studies…

Lexical Priming

Lexical Priming

- **Meyer & Schvaneveldt** *(1973)*:
  - **Present a pair of words in sequence: PRIME…. TARGET**

Lexical Priming

- **Meyer & Schvaneveldt** *(1973)*:
  - Present a pair of words in sequence: PRIME…. TARGET
  - Observe the prime, respond to target:

Lexical Priming

- **Meyer & Schvaneveldt** *(1973)*:
  - Present a pair of words in sequence: PRIME…. TARGET
  - Observe the prime, respond to target:
    - lexical decision task: decide if target is a word (sometimes it is not, e.g. FLURX)

Lexical Priming

- **Meyer & Schvaneveldt** *(1973)*:
  - Present a pair of words in sequence: PRIME…. TARGET
  - **Observe the prime, respond to target:**
    - lexical decision task: decide if target is a word (sometimes it is not, e.g. FLURX)
    - **word naming task: just say the target out loud**

Lexical Priming

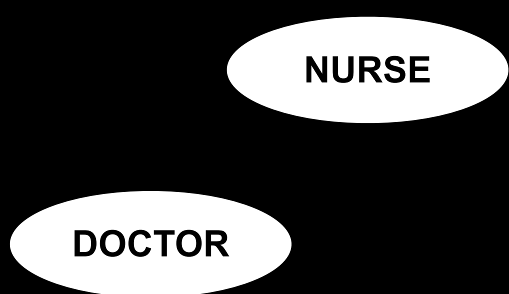
- **Meyer & Schvaneveldt** *(1973)*:
  - Present a pair of words in sequence: PRIME…. TARGET
  - Observe the prime, respond to target:
    - lexical decision task: decide if target is a word (sometimes it is not, e.g. FLURX)
    - word naming task: just say the target out loud
  - Manipulate relatedness of cue and probe

Lexical Priming

- **Meyer & Schvaneveldt** *(1973)*:
  - Present a pair of words in sequence: PRIME…. TARGET
  - Observe the prime, respond to target:
    - lexical decision task: decide if target is a word (sometimes it is not, e.g. FLURX)
    - word naming task: just say the target out loud
  - Manipulate relatedness of cue and probe
  - **Result: faster response if prime is *related* to target: DOCTOR…. NURSE**

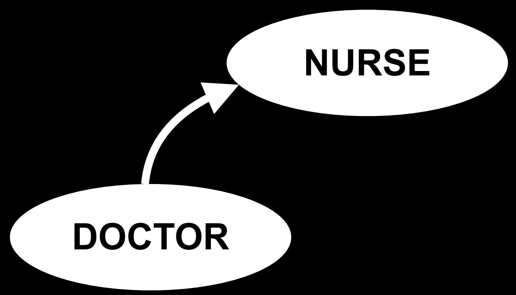
Lexical Priming

- **Meyer & Schvaneveldt** *(1973)*:
  - Present a pair of words in sequence: PRIME…. TARGET
  - Observe the prime, respond to target:
    - lexical decision task: decide if target is a word (sometimes it is not, e.g. FLURX)
    - word naming task: just say the target out loud
  - Manipulate relatedness of cue and probe
  - Result: faster response if prime is *related* to target: DOCTOR…. NURSE



Lexical Priming

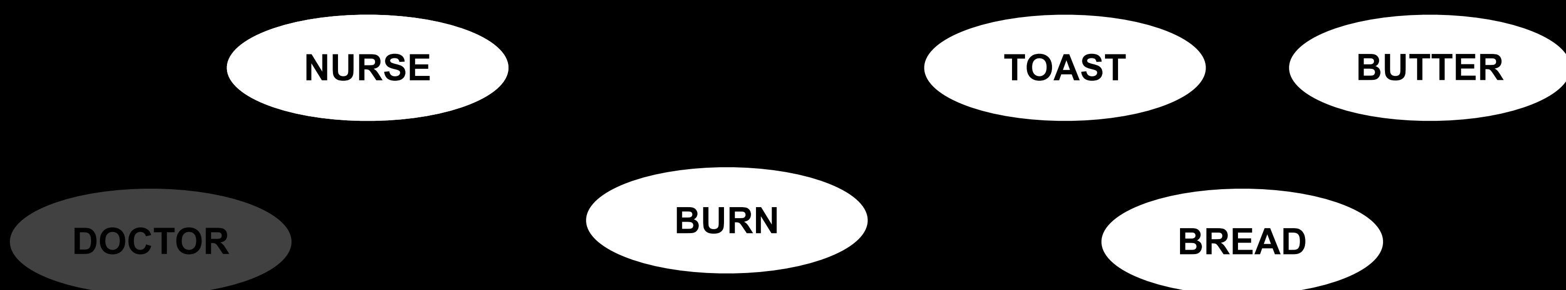
- **Meyer & Schvaneveldt** *(1973)*:
  - Present a pair of words in sequence: PRIME…. TARGET
  - Observe the prime, respond to target:
    - lexical decision task: decide if target is a word (sometimes it is not, e.g. FLURX)
    - word naming task: just say the target out loud
  - Manipulate relatedness of cue and probe
  - Result: faster response if prime is *related* to target: DOCTOR…. NURSE



Lexical Priming

- **Meyer & Schvaneveldt** *(1973)*:
  - Present a pair of words in sequence: PRIME…. TARGET
  - Observe the prime, respond to target:
    - lexical decision task: decide if target is a word (sometimes it is not, e.g. FLURX)
    - word naming task: just say the target out loud
  - Manipulate relatedness of cue and probe
  - **Result:** faster response if prime is *related* to target: DOCTOR…. NURSE

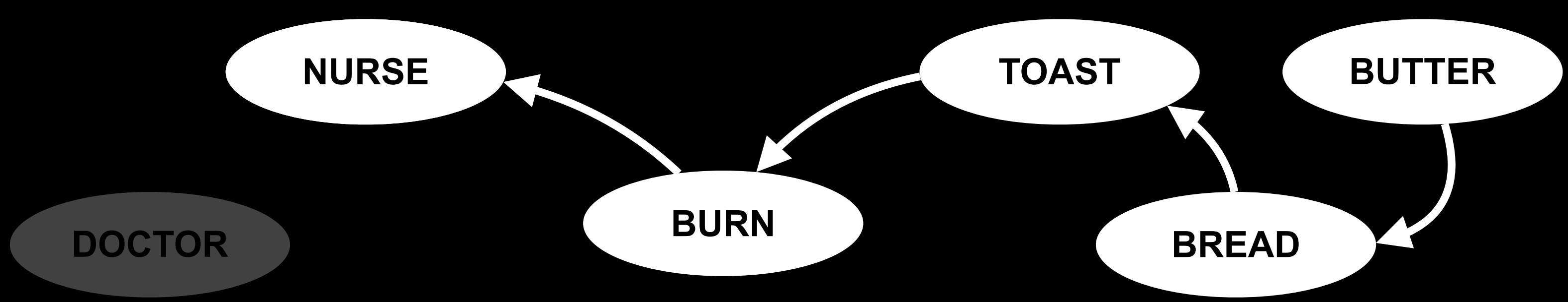
    slower if prime is *not* related to the target: BUTTER…. NURSE



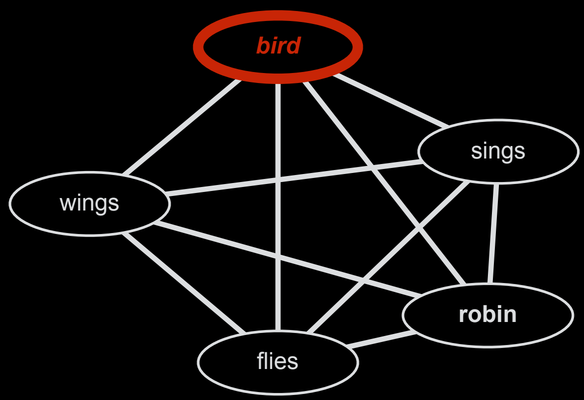
Lexical Priming

- **Meyer & Schvaneveldt** *(1973)*:
  - Present a pair of words in sequence: PRIME…. TARGET
  - Observe the prime, respond to target:
    - lexical decision task: decide if target is a word (sometimes it is not, e.g. FLURX)
    - word naming task: just say the target out loud
  - Manipulate relatedness of cue and probe
  - **Result:** faster response if prime is *related* to target: DOCTOR…. NURSE

    slower if prime is *not* related to the target: BUTTER….   NURSE

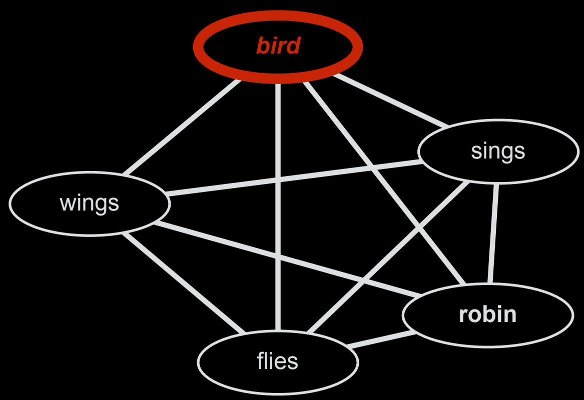


Additional Empirical Support



Additional Empirical Support

- Semantic priming
  - subjects respond faster to a word following a related vs. unrelated word

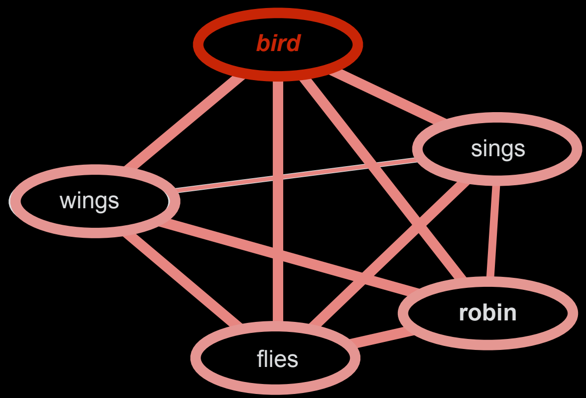


Additional Empirical Support

- **Semantic priming**
  - subjects respond faster to a word following a related vs. unrelated word
  - mechanism:

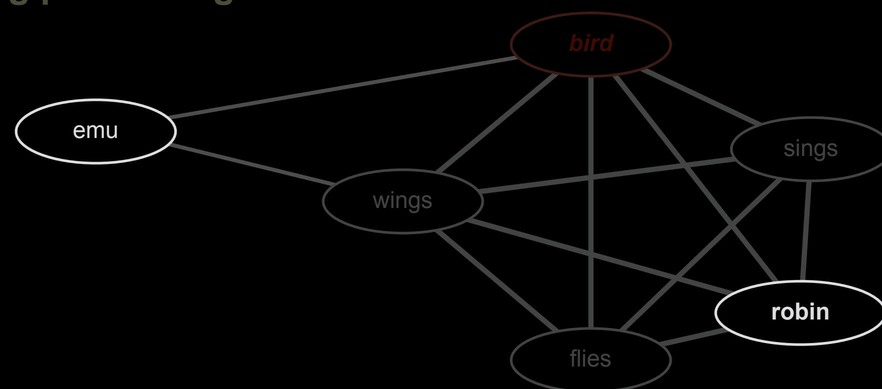
    activation spreads from activated node to related nodes

    pre-activated nodes are *primed*, facilitating processing



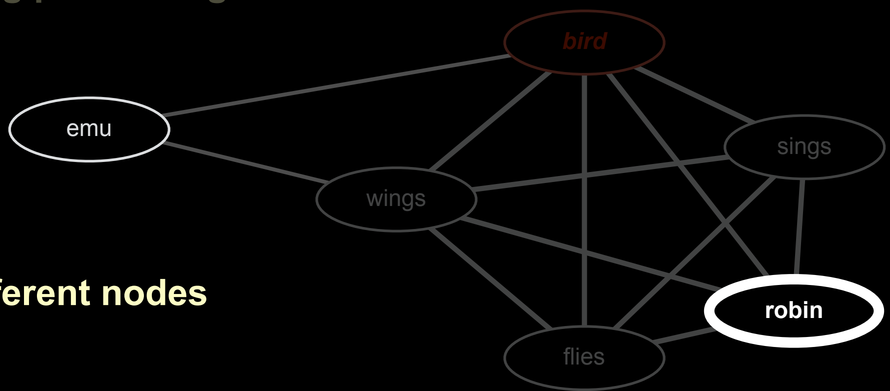
Additional Empirical Support

- Semantic priming
  - subjects respond faster to a word following a related vs. unrelated word
  - mechanism:
    - activation spreads from activated node to related nodes
    - pre-activated nodes are *primed*, facilitating processing
- Frequency effects:
  - faster to say “robin” than “emu”



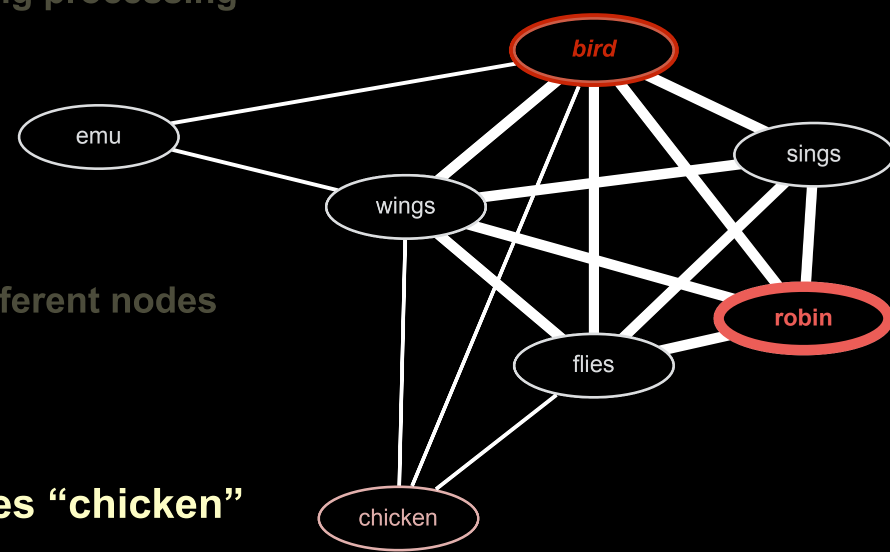
Additional Empirical Support

- Semantic priming
  - subjects respond faster to a word following a related vs. unrelated word
  - mechanism:
    - activation spreads from activated node to related nodes
    - pre-activated nodes are *primed*, facilitating processing
- Frequency effects:
  - faster to say "robin" than "emu"
  - mechanism:
    - base level of activation is different for different nodes



Additional Empirical Support

- Semantic priming
  - subjects respond faster to a word following a related vs. unrelated word
  - mechanism:
    - activation spreads from activated node to related nodes
    - pre-activated nodes are *primed*, facilitating processing
- Frequency effects:
  - faster to say “robin” than “emu”
  - mechanism:
    - base level of activation is different for different nodes
- Typicality effects:
  - “bird” primes “robin” more than it does “chicken”



Additional Empirical Support

- Semantic priming
  - subjects respond faster to a word following a related vs. unrelated word
  - mechanism:

    activation spreads from activated node to related nodes

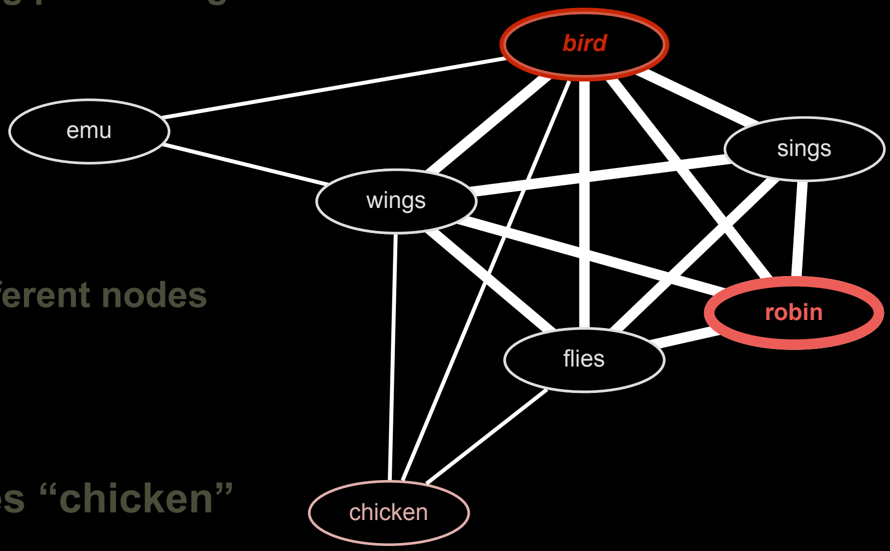
    pre-activated nodes are *primed*, facilitating processing

- Frequency effects:
  - faster to say "robin" than "emu"
  - mechanism:

    base level of activation is different for different nodes

- Typicality effects:
  - "bird" primes "robin" more than it does "chicken"
  - mechanism:

    typical members of a category *(prototypes)* are more centrally placed than others so they are more likely to get activated, and to get more activated than others…



Success and Failures of *Simple* Spreading Activation Models

Success and Failures of *Simple* Spreading Activation Models

- Successes
  - explicitly specify structure and mechanisms of long term (semantic) memory
  - can account for much relevant data

Success and Failures of *Simple* Spreading Activation Models

- **Successes**
  - explicitly specify structure and mechanisms of long term (semantic) memory
  - can account for much relevant data
- **Problems:**
  - don’t specify how the structure (nodes and links) got there:
    - presumably experience and learning (we’ll get to that)

Success and Failures of *Simple* Spreading Activation Models

- Successes
  - explicitly specify structure and mechanisms of long term (semantic) memory
  - can account for much relevant data
- Problems:
  - don’t specify how the structure (nodes and links) got there:
    - presumably experience and learning (we’ll get to that)
    - but what about nodes for things that were never experienced?

Success and Failures of *Simple* Spreading Activation Models

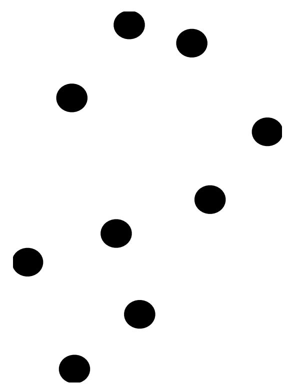
- Successes
  - explicitly specify structure and mechanisms of long term (semantic) memory
  - can account for much relevant data

- Problems:
  - don’t specify how the structure (nodes and links) got there:
    - presumably experience and learning (we’ll get to that)
    - but what about nodes for things that were never experienced?

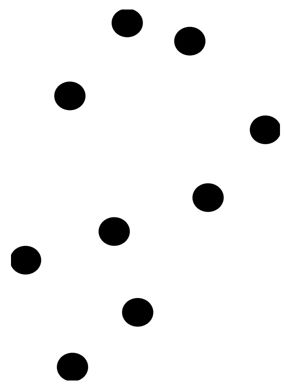
    - Example: people respond to faster to prototypes they’ve never seen…

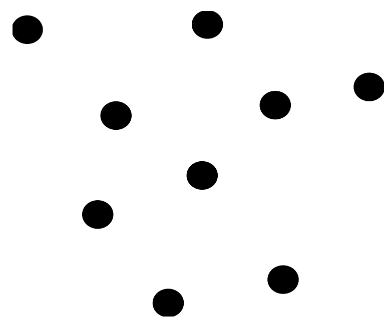
Posner & Keele Demo

- You will see dot patterns
- Judge whether each belongs to category A or category B
- Guess at first,
  but will get better with feedback…

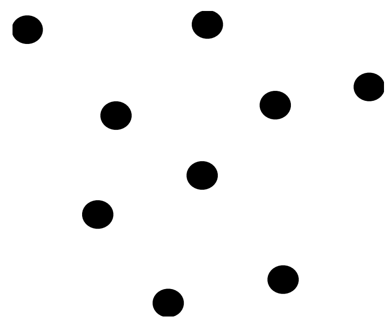


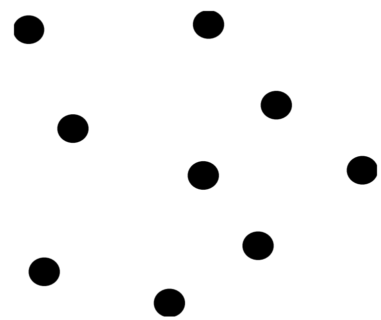
That was Category B.



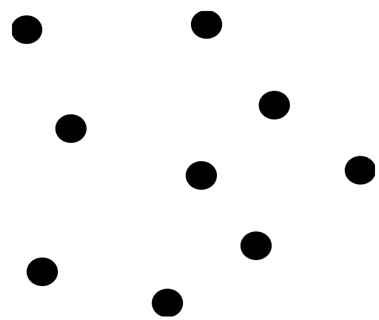


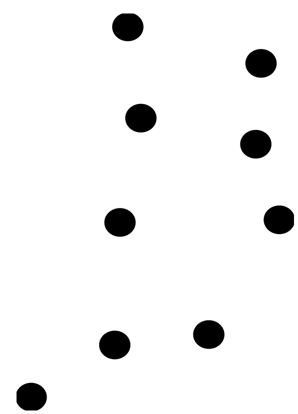
That was Category A.



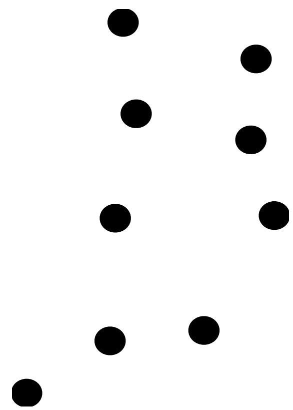


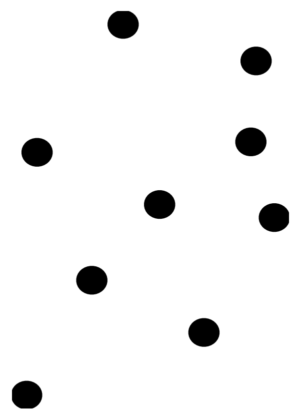
That was Category A.



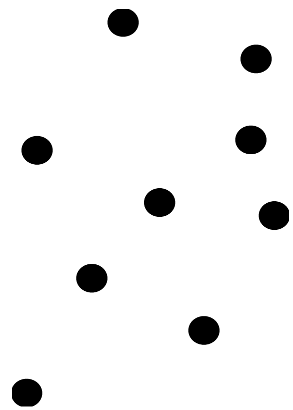


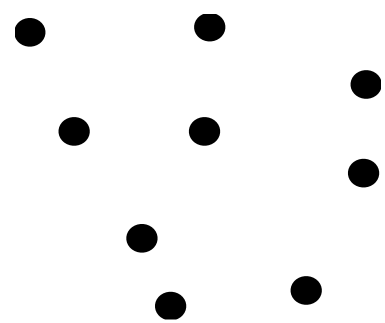
That was Category B.



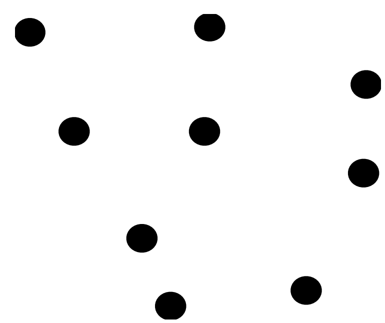


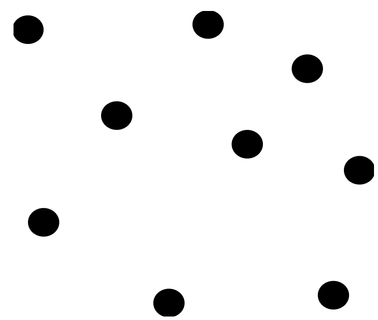
That was Category B.



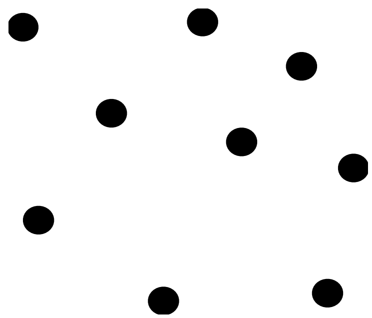


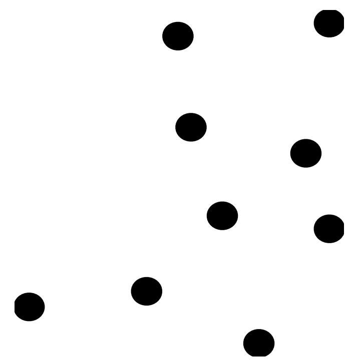
That was Category A.



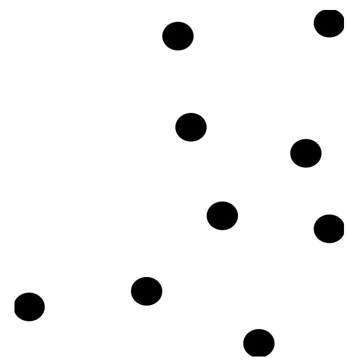


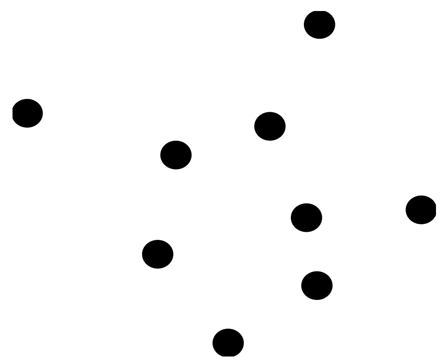
That was Category A.



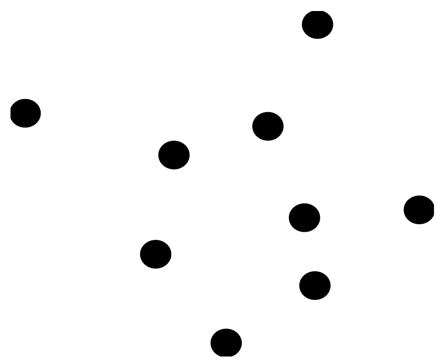


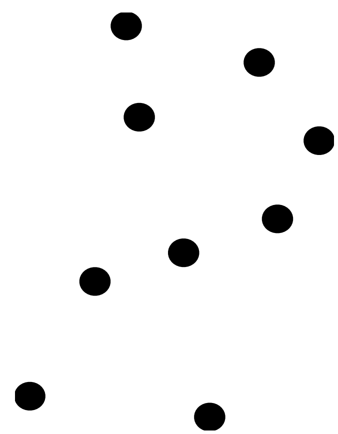
That was Category B.



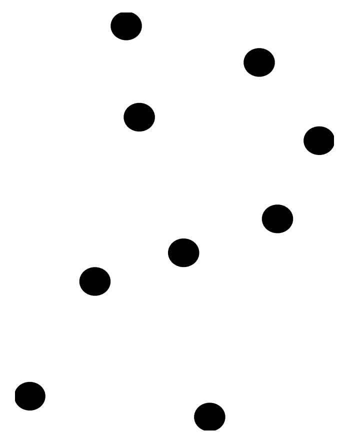


That was Category A.

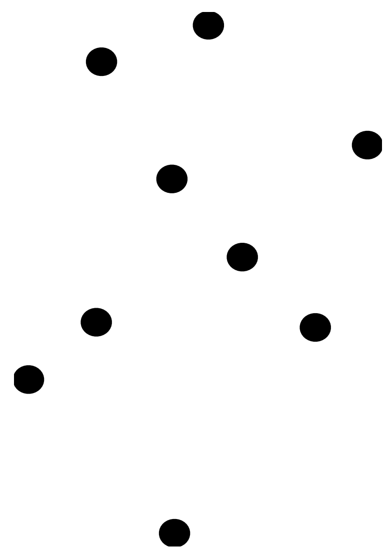




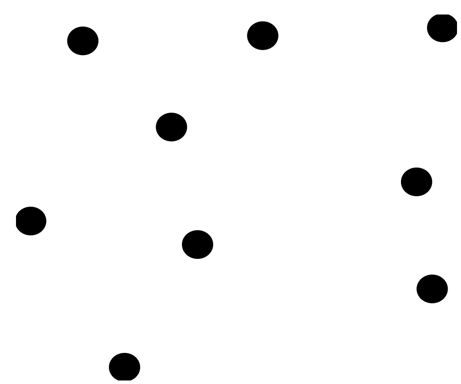
That was Category B.



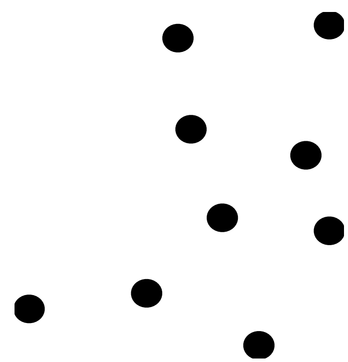
Now, judge whether or not
you have seen the following dot patterns…



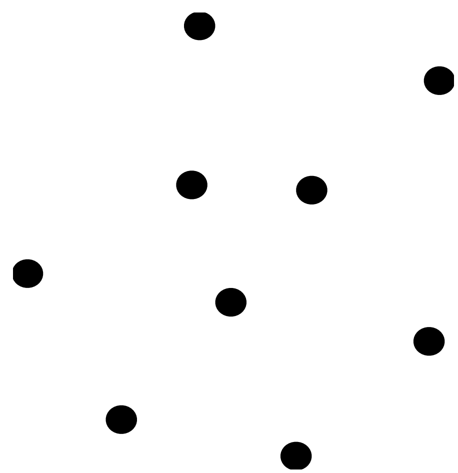
Old or New



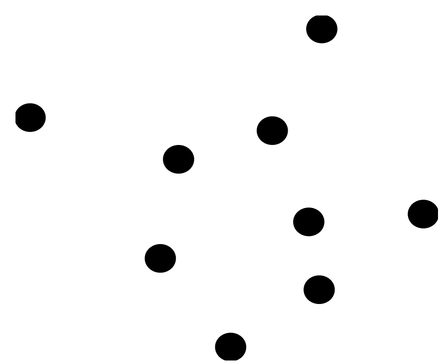
Old or New



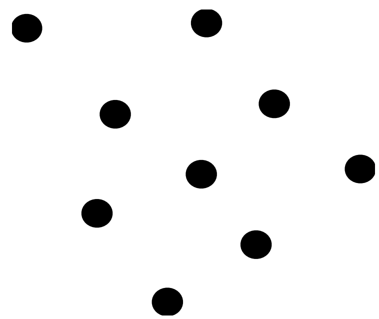
Old or New



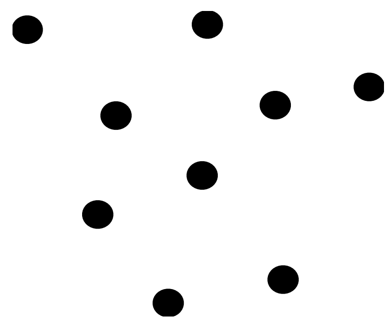
Old or New



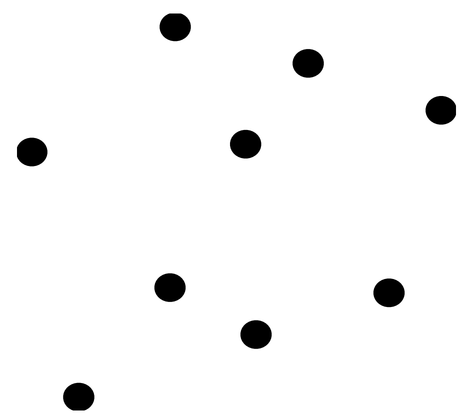
Old or New



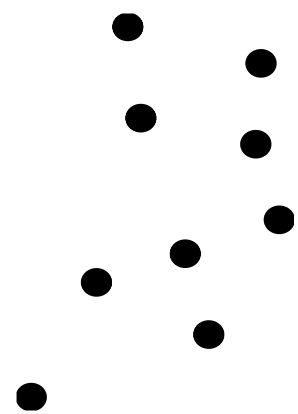
Old or New



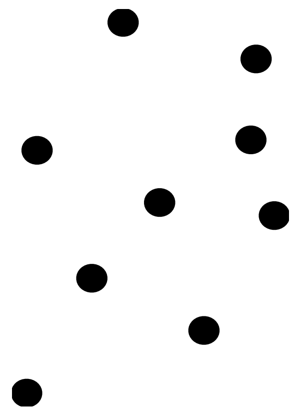
Old or New



Old or New

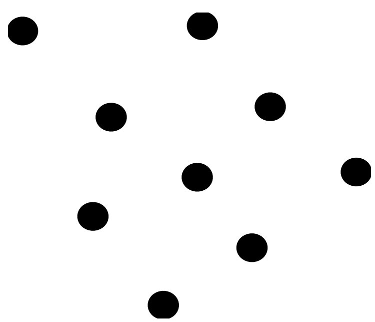


Old or New



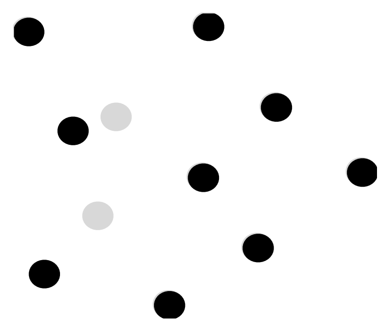
Old or New

Category A Members

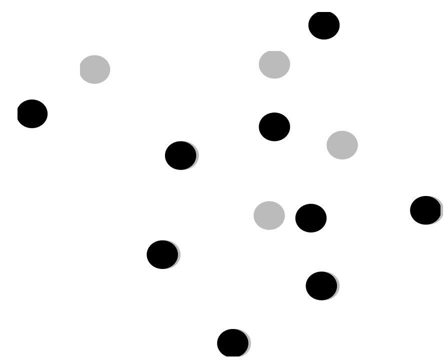


PROTOTYPE-A

6th stimulus

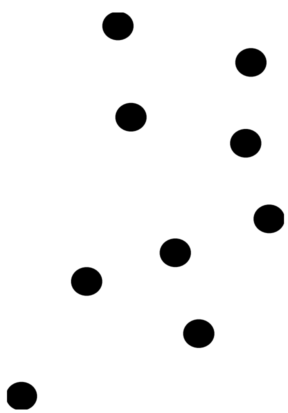


low distortion



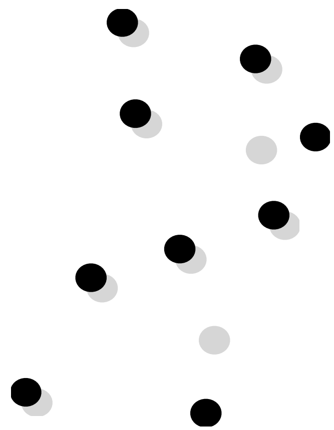
high distortion

Category B Members

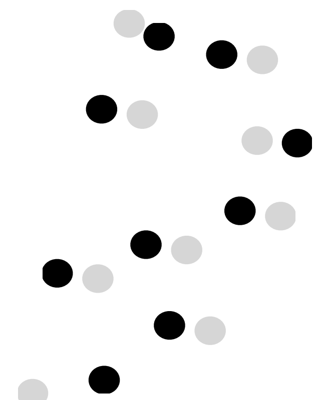


PROTOTYPE-B

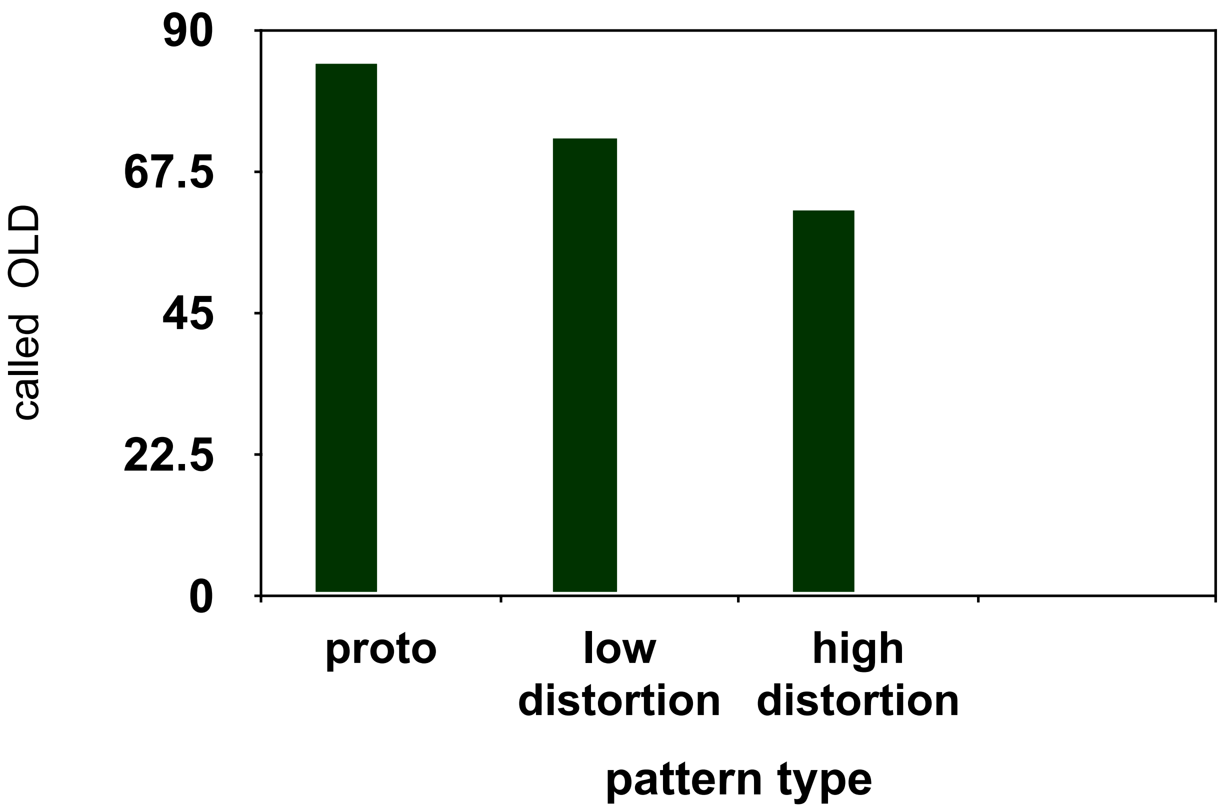
9th stimulus



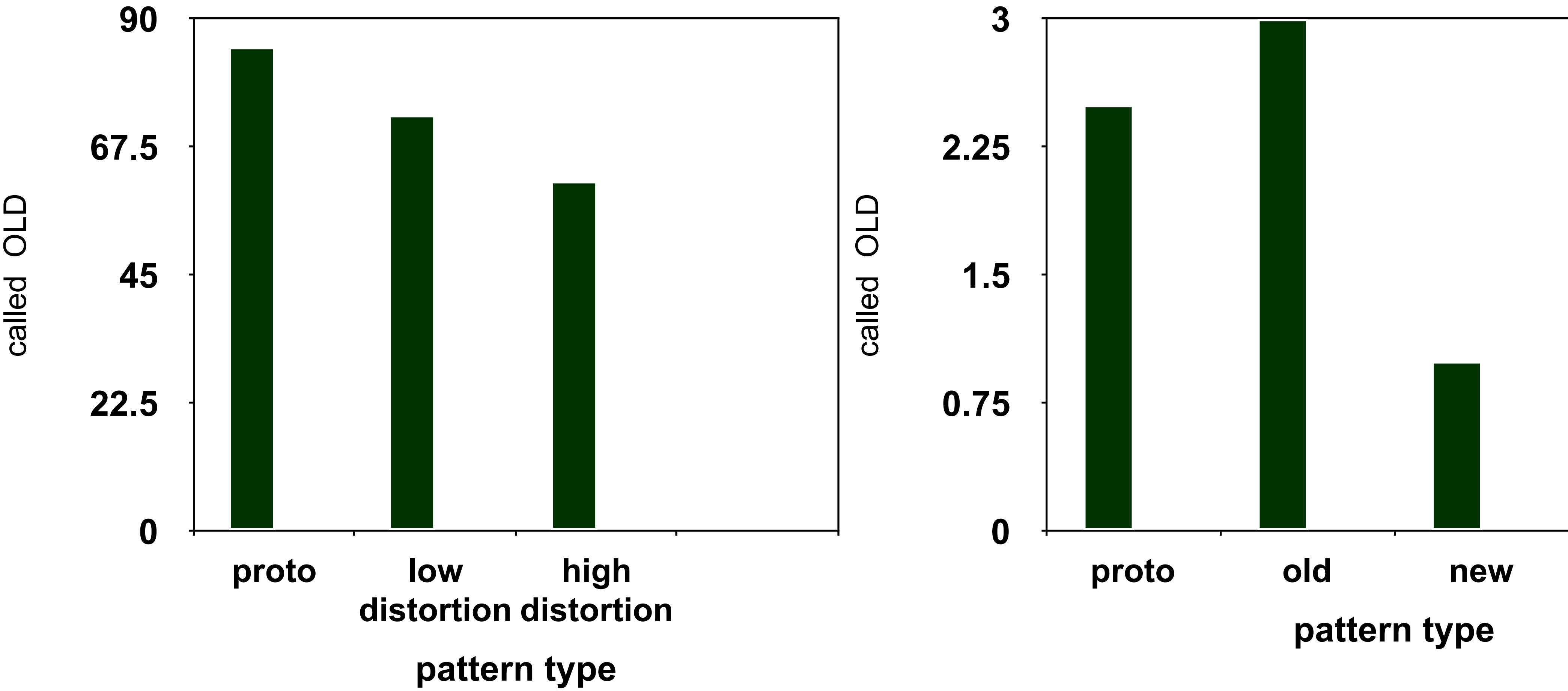
low distortion



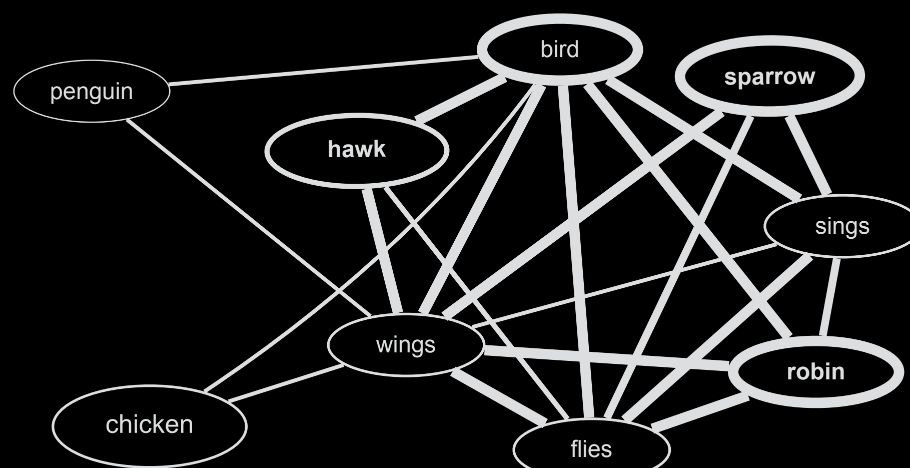
high distortion



90
67.5
45
22.5
0
called OLD
proto
low distortion
high distortion
pattern type

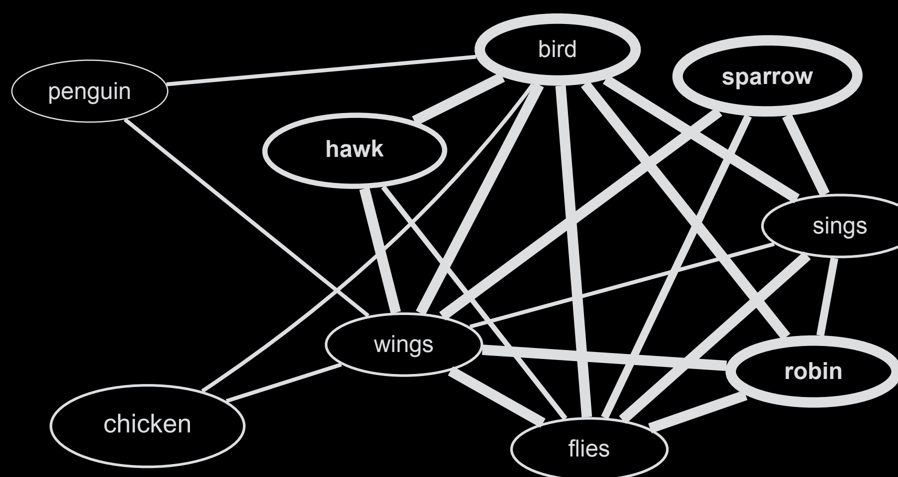

90
67.5
45
22.5
0
called OLD
proto
low distortion
high distortion
pattern type
3
2.25
1.5
0.75
0
called OLD
proto
old
new
pattern type

Exemplars vs. Prototypes



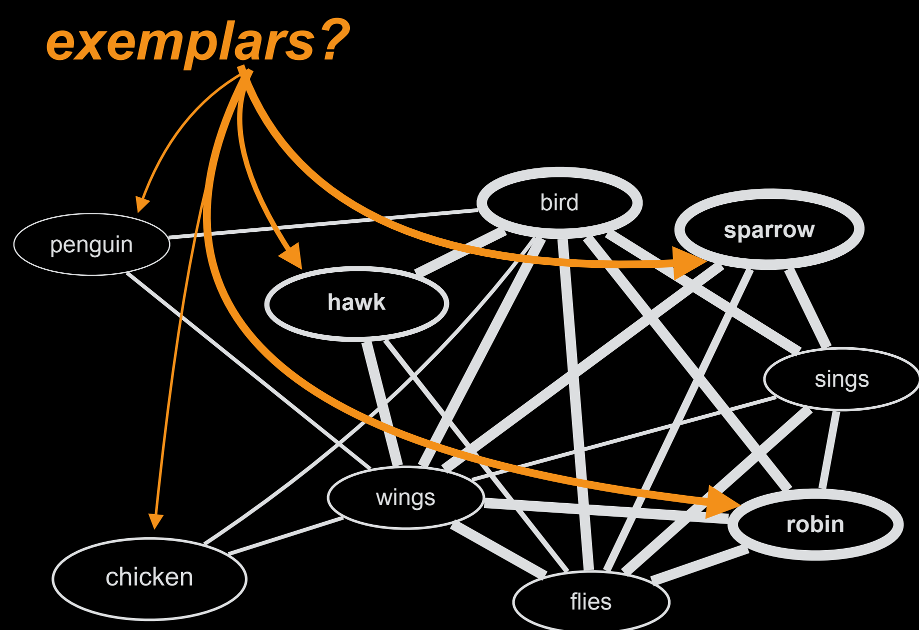
Exemplars vs. Prototypes

- How do we represent both individual instances and categories?



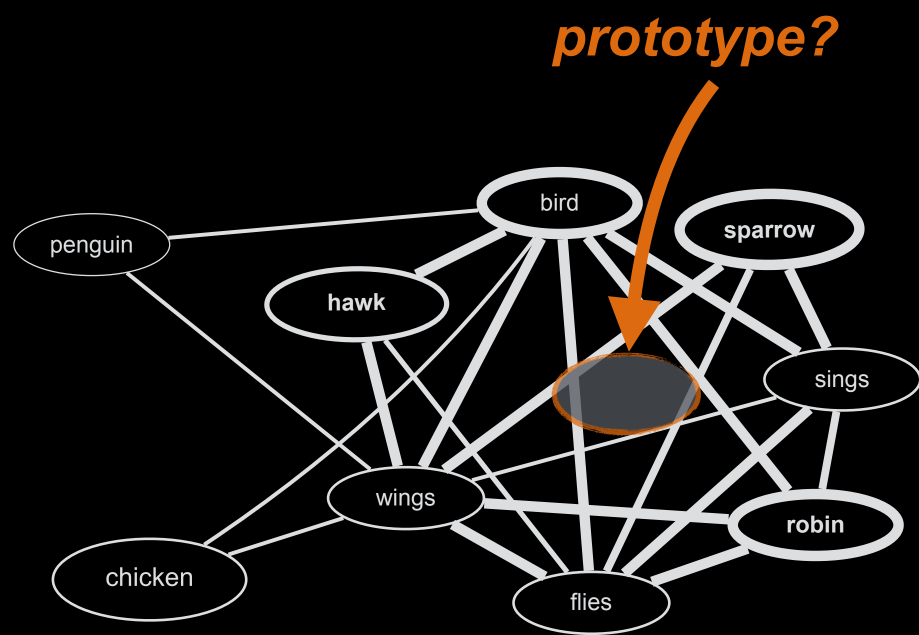
Exemplars vs. Prototypes

- **How do we represent both individual instances and categories?**
  - **collection of exemplars?** *(e.g., Atkinson & Shiffrin, 1968)*



Exemplars vs. Prototypes

- **How do we represent both individual instances and categories?**
  - **– collection of exemplars?** *(e.g., Atkinson & Shiffrin, 1968)*
  - **– abstraction of a prototype?** *(Rosch, 1983)*



Model of Distributed Memory (MDM)

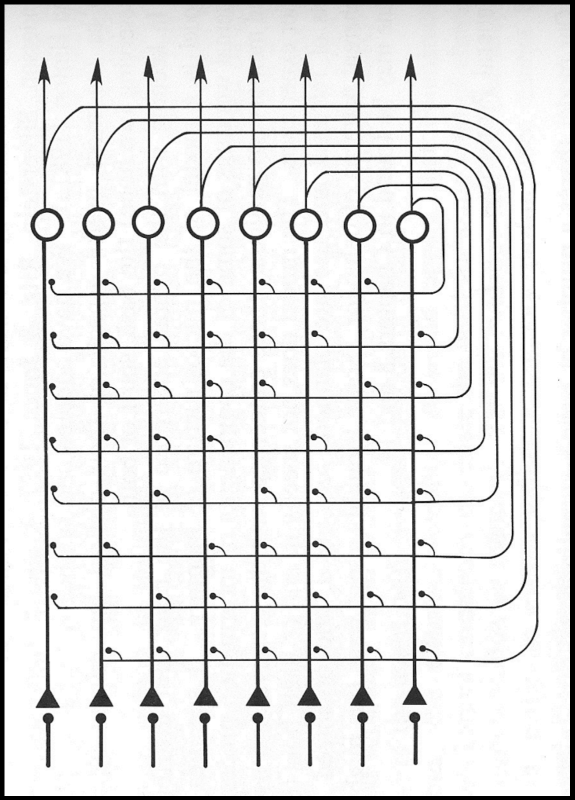
McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

- **Key features of the model:**
  - ***Recurrent*** **(attractor) network**

$$i_i = \Sigma a_i w_{ij}$$
$$net_i = i_i + e_i$$



Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

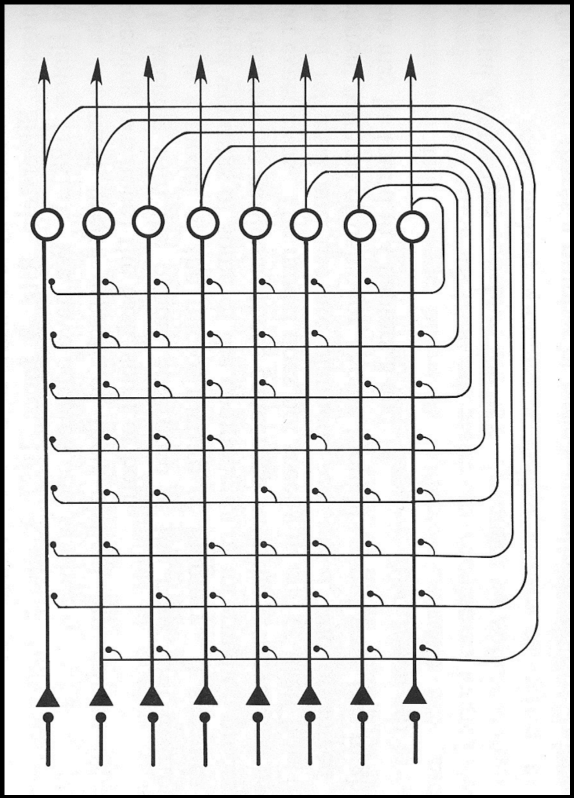
- **Key features of the model:**
  - ***Recurrent*** **(attractor) network**

$$i_i = \Sigma a_i w_{ij}$$
$$net_i = i_i + e_i$$

  - ***Trained*** **using Rescorla-Wagner (delta) rule to represent input as a** ***pattern of activity*** **over units in network (auto-associator)**

$$\Delta w_{ij} = \eta(e_i - i_i)\ a_j$$

(constraint: target value of each unit must be predictable from a linear combination of the other units, since no hidden units)



Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

- **Key features of the model:**
  - ***Recurrent*** **(attractor) network**

$$i_i = \Sigma a_i w_{ij}$$
$$net_i = i_i + e_i$$

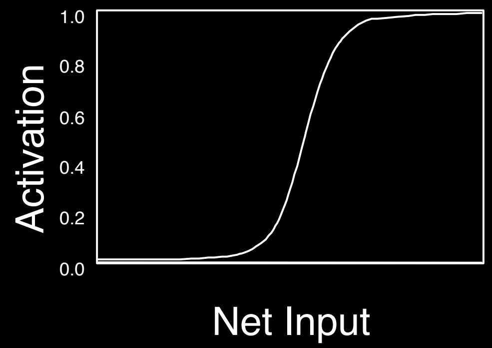
  - ***Trained*** **using Rescorla-Wagner (delta) rule to represent input as a** ***pattern of activity*** **over units in network (auto-associator)**

$$\Delta w_{ij} = \eta(e_i - i_i)\ a_j$$

(constraint: target value of each unit must be predictable from a linear combination of the other units, since no hidden units)

  - ***Non-linear*** **(sigmoid) activation function**

$$\Delta a_i = a_i + net_i \cdot (a_{max} - a_i)\ if\ net_i > 0$$
$$\Delta a_i = a_i - net_i \cdot (a_i - a_{min})\ if\ net_i \leq 0$$



Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

- **Key features of the model:**
  - ***Recurrent*** **(attractor) network**

$$i_i = \Sigma a_i w_{ij}$$
$$net_i = i_i + e_i$$

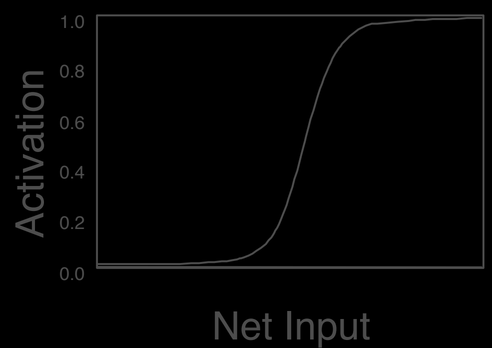
  - ***Trained*** **using Rescorla-Wagner (delta) rule to represent input as a** ***pattern of activity*** **over units in network (auto-associator)**

$$\Delta w_{ij} = \eta(e_i - i_i)\ a_j$$

(constraint: target value of each unit must be predictable from a linear combination of the other units, since no hidden units)

  - ***Non-linear*** **(sigmoid) activation function**

$$\Delta a_i = a_i + net_i \cdot (a_{max} - a_i)\ if\ net_i > 0$$
$$\Delta a_i = a_i - net_i \cdot (a_i - a_{min})\ if\ net_i \leq 0$$



  - **Exponential** ***weight decay*** **during learning (emphasizes recent experiences)**

Prototype learning

	Name Pattern	Visual Pattern
Prototype (never shown)	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Pattern 1	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Pattern 2	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −

Prototype learning

- Training patterns made up of two parts:

	Name Pattern	Visual Pattern
Prototype (never shown)	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Pattern 1	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Pattern 2	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −

Prototype learning

- Training patterns made up of two parts:
  - name part: random vectors (e.g, dog names)

	Name Pattern	Visual Pattern
Prototype (never shown)	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Pattern 1	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Pattern 2	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −

Weakly correlated

Prototype learning

- Training patterns made up of two parts:
  - name part: random vectors (e.g, dog names)
  - visual part: variants of a prototype (e.g., different dogs) (bit flipped in each element with $p = 0.2$)

	Name Pattern	Visual Pattern
Prototype (never shown)	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Pattern 1	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Pattern 2	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −

Strongly Correlated

Prototype learning

- Training patterns made up of two parts:
  - name part: random vectors (e.g, dog names)
  - visual part: variants of a prototype (e.g., different dogs)
    (bit flipped in each element with $p = 0.2$)

	Name Pattern	Visual Pattern
Prototype (never shown)	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Pattern 1	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Pattern 2	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −

- Model trained on 50 such patterns

Prototype learning

- Training patterns made up of two parts:
  - name part: random vectors (e.g, dog names)
  - visual part: variants of a prototype (e.g., different dogs)
    (bit flipped in each element with $p = 0.2$)

	Name Pattern	Visual Pattern
Prototype (never shown)	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Pattern 1	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Pattern 2	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −

  - Model trained on 50 such patterns

- Learns to recognize the prototype
  even though it was never explicitly presented
  - Responds to prototype more strongly than any given exemplar
  - Doesn’t retrieve individual names very well,
    although it can remember the most recent ones

Prototype learning

- Can learn (and keep separate) multiple, non-orthogonal prototypes: *implicit categorization*
  - increasing similarity produces increased confusability *initially*, but resolved by further *training*

RESULTS OF TESTS AFTER LEARNING THE DOG, CAT, AND BAGEL PATTERNS

	Name Pattern	Visual Pattern
Pattern for dog prototype	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Response to dog name		
Response to dog visual pattern		
Pattern for cat prototype	+ + − − + + − −	+ − + + − − − − + − + − + + − +
Response to cat name		
Response to cat visual pattern		
Pattern for bagel prototype	+ − − + + − − +	+ + − + − + + − + − − + + + + −
Response to bagel name		
Response to bagel visual pattern		

Prototype learning

- Can learn (and keep separate) multiple, non-orthogonal prototypes: *implicit categorization*
  - increasing similarity produces increased confusability *initially*, but resolved by further *training*

RESULTS OF TESTS AFTER LEARNING THE DOG, CAT, AND BAGEL PATTERNS

	Name Pattern	Visual Pattern
Pattern for dog prototype	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Response to dog name		+3 −4 +4 +4 −4 −4 −4 −4 +4 +4 +4 +3 +4 −4 −4 −3
Response to dog visual pattern		
Pattern for cat prototype	+ + − − + + − −	+ − + + − − − − + − + − + + − +
Response to cat name		+4 −3 +4 +4 −4 −3 −3 −4 +4 −4 +4 −4 +4 +4 −4 +4
Response to cat visual pattern		
Pattern for bagel prototype	+ − − + + − − +	+ + − + − + + − + − − + + + + −
Response to bagel name		+3 +4 −4 +4 −4 +4 +4 −4 +4 −4 −4 +4 +3 +4 +4 −4
Response to bagel visual pattern		

Prototype learning

- Can learn (and keep separate) multiple, non-orthogonal prototypes: *implicit categorization*
  - increasing similarity produces increased confusability *initially*, but resolved by further *training*

RESULTS OF TESTS AFTER LEARNING THE DOG, CAT, AND BAGEL PATTERNS

	Name Pattern	Visual Pattern
Pattern for dog prototype	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Response to dog name		
Response to dog visual pattern	+5 −4 +4 −5 +5 −4 +4 −4	
Pattern for cat prototype	+ + − − + + − −	+ − + + − − − − + − + − + + − +
Response to cat name		
Response to cat visual pattern	+5 +4 −4 −5 +4 +4 −4 −4	
Pattern for bagel prototype	+ − − + + − − +	+ + − + − + + − + − − + + + + −
Response to bagel name		
Response to bagel visual pattern	+4 −4 −4 +4 +4 −4 −4 +4	

Co-existence of Prototype and Exemplars

RESULTS OF TESTS WITH PROTOTYPE AND SPECIFIC EXEMPLAR PATTERNS

	Name Pattern	Visual Pattern
Pattern for dog prototype	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Response to prototype name		
Response to prototype visual pattern		
Pattern for Fido exemplar	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −
Response to Fido name		
Response to Fido visual pattern		
Pattern for Rover exemplar	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Response to Rover name		
Response to Rover visual pattern		

Co-existence of Prototype and Exemplars

- Train on two *specific* exemplars with names,
  and many other random distortions with category label

RESULTS OF TESTS WITH PROTOTYPE AND SPECIFIC EXEMPLAR PATTERNS

	Name Pattern	Visual Pattern
Pattern for dog prototype	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Response to prototype name		
Response to prototype visual pattern		
Pattern for Fido exemplar	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −
Response to Fido name		
Response to Fido visual pattern		
Pattern for Rover exemplar	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Response to Rover name		
Response to Rover visual pattern		

Co-existence of Prototype and Exemplars

- Train on two *specific* exemplars with names,
  and many other random distortions with category label

RESULTS OF TESTS WITH PROTOTYPE AND SPECIFIC EXEMPLAR PATTERNS

	Name Pattern	Visual Pattern
Pattern for dog prototype	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Response to prototype name		
Response to prototype visual pattern		
Pattern for Fido exemplar	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −
Response to Fido name		
Response to Fido visual pattern		
Pattern for Rover exemplar	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Response to Rover name		
Response to Rover visual pattern		

funny eyes and tail

Co-existence of Prototype and Exemplars

- Train on two *specific* exemplars with names,
  and many other random distortions with category label

RESULTS OF TESTS WITH PROTOTYPE AND SPECIFIC EXEMPLAR PATTERNS

	Name Pattern	Visual Pattern
Pattern for dog prototype	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Response to prototype name		
Response to prototype visual pattern		
Pattern for Fido exemplar	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −
Response to Fido name		
Response to Fido visual pattern		
Pattern for Rover exemplar	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Response to Rover name		
Response to Rover visual pattern		

funny eyes and tail

funny ears

Co-existence of Prototype and Exemplars

- Train on two *specific* exemplars with names, and many other random distortions with category label
  - Can retrieve:
    - features of the labeled exemplars

RESULTS OF TESTS WITH PROTOTYPE AND SPECIFIC EXEMPLAR PATTERNS

	Name Pattern	Visual Pattern
Pattern for dog prototype	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Response to prototype name		
Response to prototype visual pattern		
Pattern for Fido exemplar	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −
Response to Fido name		+4 −4 −4 +4 −4 −4 −4 −4 +4 +4 +4 +4 +4 +4 −4 −4
Response to Fido visual pattern		
Pattern for Rover exemplar	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Response to Rover name		+4 +5 +4 +4 −4 −4 −4 −4 +4 +4 +4 +4 +4 −4 −4 −4
Response to Rover visual pattern		

funny eyes and tail

funny ears

Co-existence of Prototype and Exemplars

- Train on two *specific* exemplars with names, and many other random distortions with category label
  - Can retrieve:
    - features of the labeled exemplars
    - the names of each exemplar from each features

RESULTS OF TESTS WITH PROTOTYPE AND SPECIFIC EXEMPLAR PATTERNS

	Name Pattern	Visual Pattern
Pattern for dog prototype	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Response to prototype name		
Response to prototype visual pattern		
Pattern for Fido exemplar	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −
Response to Fido name		
Response to Fido visual pattern	+5 −5 −3 −5 +4 −5 −3 −5	
Pattern for Rover exemplar	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Response to Rover name		
Response to Rover visual pattern	+4 −4 −2 +4 +4 +4 −2 +4	

funny eyes and tail

funny ears

Co-existence of Prototype and Exemplars

- Train on two *specific* exemplars with names,
  and many other random distortions with category label
  - Can retrieve:
    - features of the labeled exemplars
    - the names of each exemplar from each features
  - Retrieves prototype for all others

RESULTS OF TESTS WITH PROTOTYPE AND SPECIFIC EXEMPLAR PATTERNS

	Name Pattern	Visual Pattern
Pattern for dog prototype	+ − + − + − + −	+ − + + − − − − + + + + + − − −
Response to prototype name		+4 −5 +3 +3 −4 −3 −3 −3 +3 +3 +4 +3 +4 −3 −4 −4
Response to prototype visual pattern	+5 −4 +4 −4 +5 −4 +4 −4	
Pattern for Fido exemplar	+ − − − + − − −	+ − (−) + − − − − + + + + + (+) − −
Response to Fido name		+4 −4 −4 +4 −4 −4 −4 −4 +4 +4 +4 +4 +4 +4 −4 −4
Response to Fido visual pattern	+5 −5 −3 −5 +4 −5 −3 −5	
Pattern for Rover exemplar	+ − − + + + − +	+ (+) + + − − − − + + + + + − − −
Response to Rover name		+4 +5 +4 +4 −4 −4 −4 −4 +4 +4 +4 +4 +4 −4 −4 −4
Response to Rover visual pattern	+4 −4 −2 +4 +4 +4 −2 +4	

prototypical eyes, tail and ears

Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

- **The model can extract the prototype (central tendency) from a set of patterns (series of exemplars)**

Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

- It can do this for several different prototypes using the same set of connections

Model of Distributed Memory (MDM)

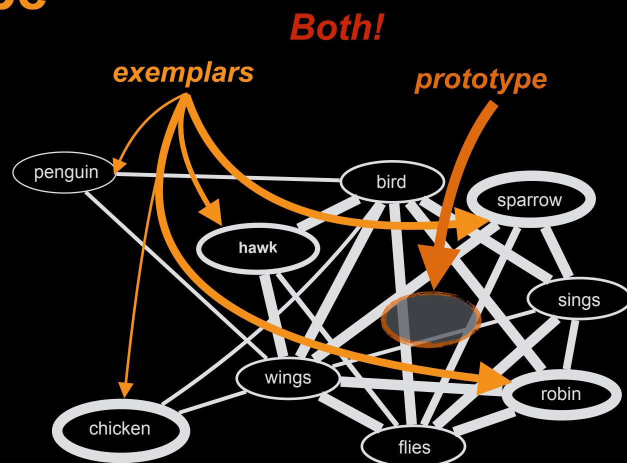
McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

- **It does not depend on being presented with labels**

Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

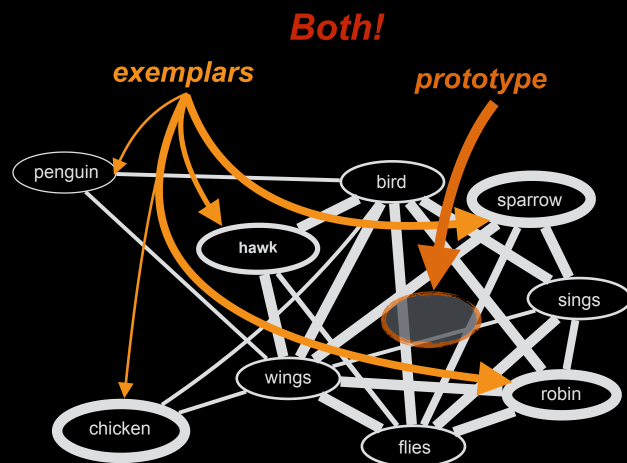
- **Representations of specific distinguishable instances can co-exist with knowledge of the prototype**



Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

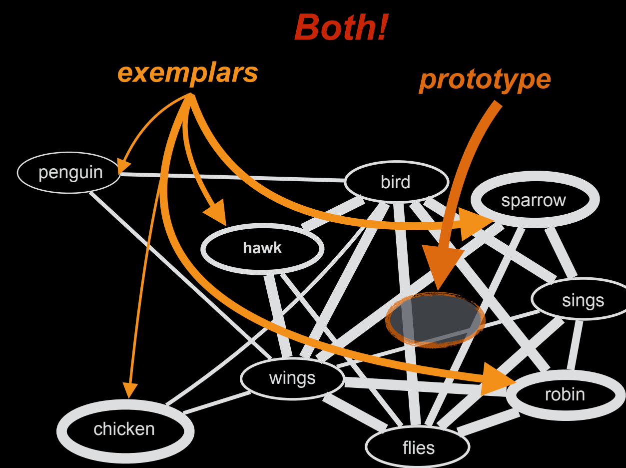
- **Fits empirical data:**



Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

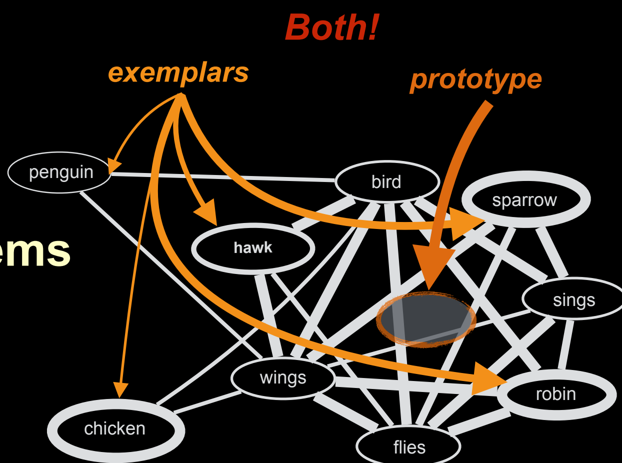
– best (fastest) response to prototype



Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

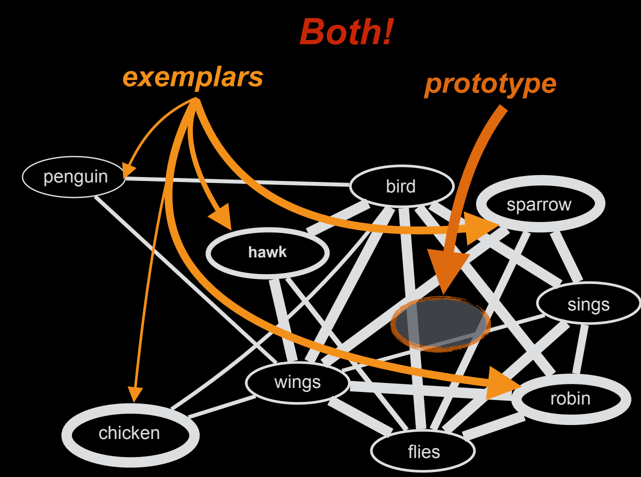
– fastest and most accurate response to familiar items



Model of Distributed Memory (MDM)

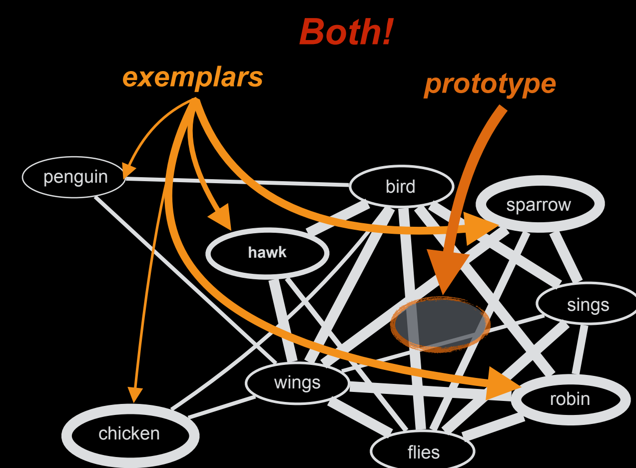
McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

– **priming effects (identity > similarity > none)**



Model of Distributed Memory (MDM)

McClelland & Rumelhart (1985, JEP:General, 114(2), 159-188)

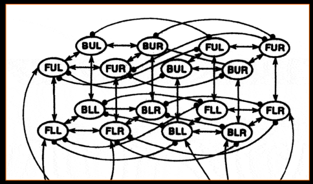


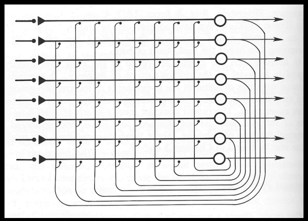
– **interaction of priming and familiarity effects**
(priming greater for unfamiliar than familiar items)

Learning

Learning

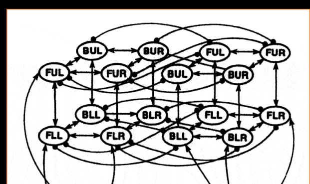
- So far, we've focused on processing:

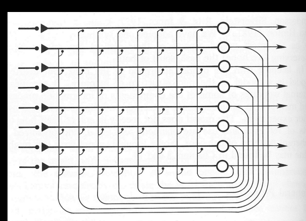


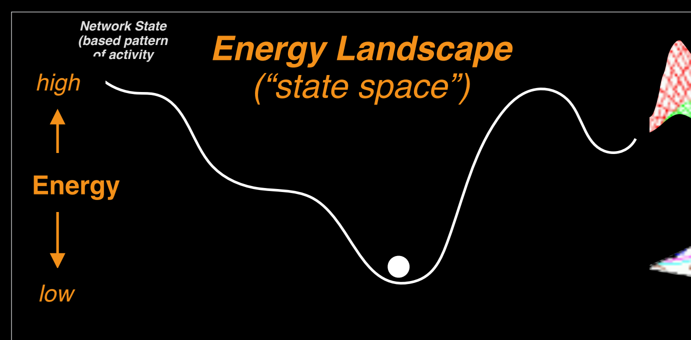


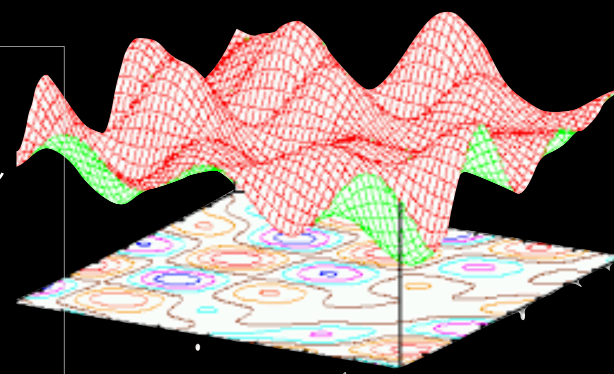
Learning

- **So far, we've focused on processing:**
  - **– dynamics of *encoding* and *representation* information** *(≈ weather)*



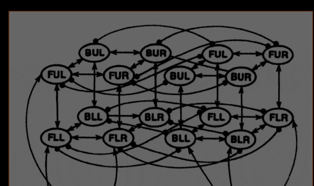


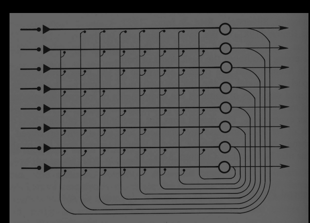


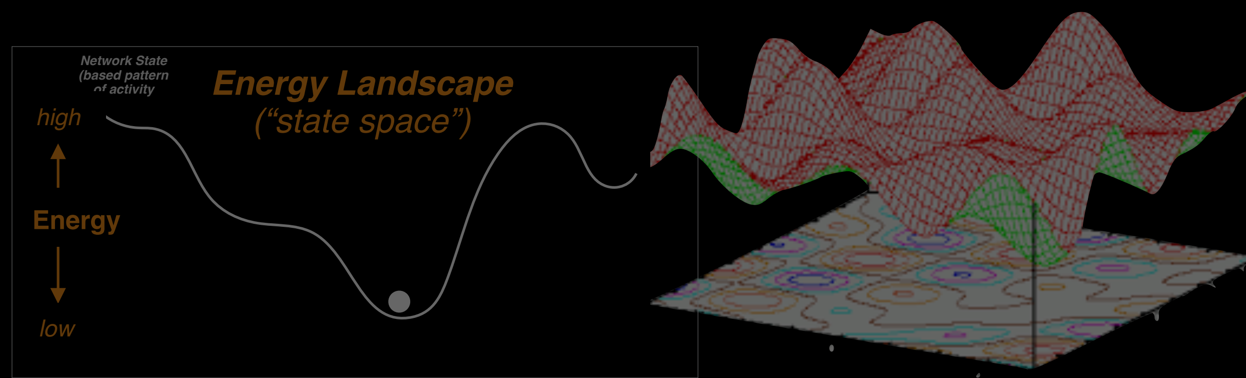


Learning

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





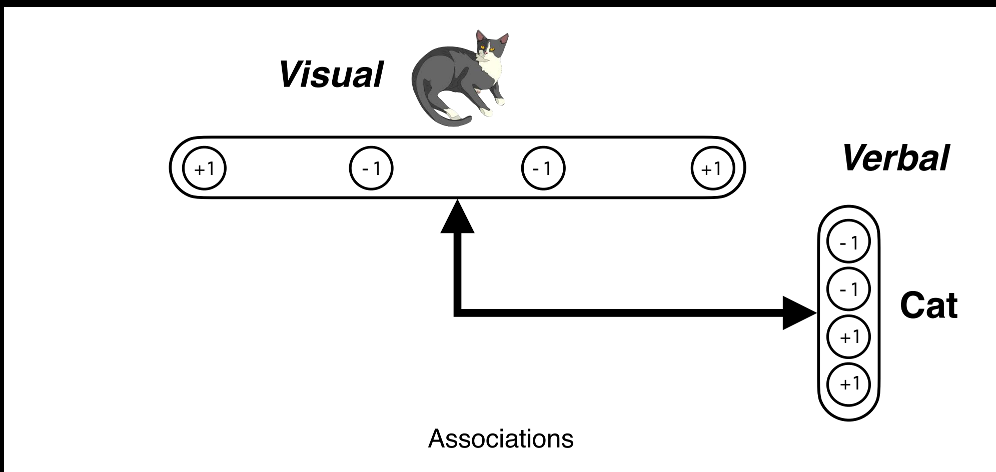


- What about learning?
  - how is the landscape shaped? *(≈geology)*
  - dynamics of *acquisition*

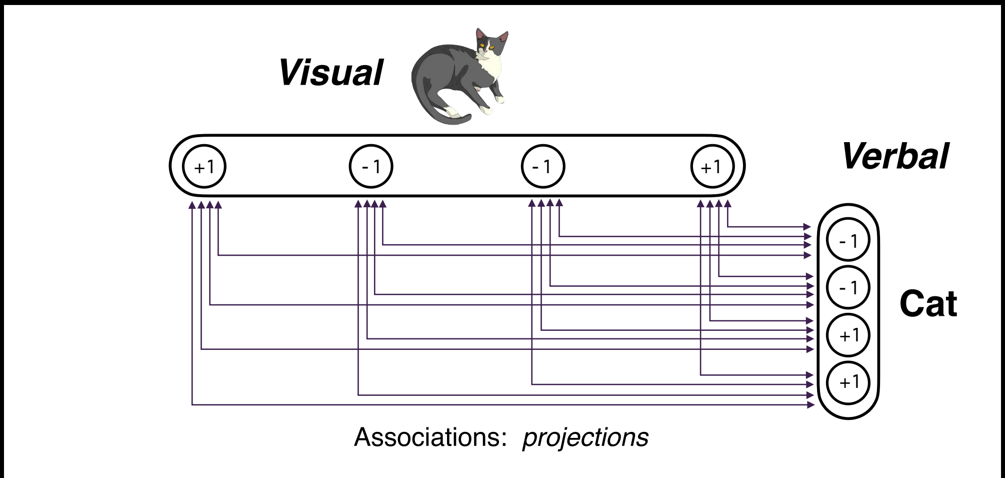
Simple Pattern Associator

- "Association":
  - Network that learns associations (correlations) between input and output patterns; given an input, it can generate the output…

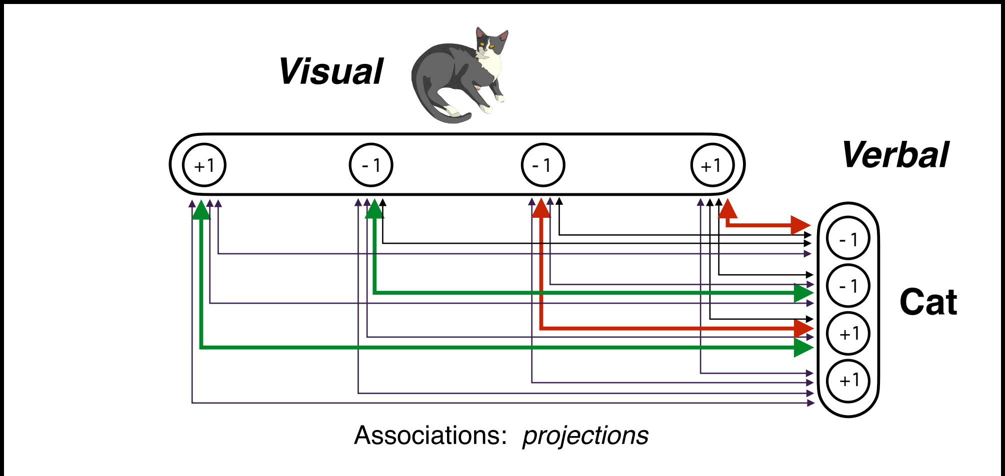
Pattern Associator



Pattern Associator



Pattern Associator



Pattern Associator

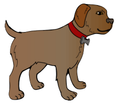
+1	-1	-1	+1		
-.25	+.25	+.25	-.25	-1	Cat
-.25	+.25	+.25	-.25	-1	
+.25	-.25	-.25	+.25	+1	
+.25	-.25	-.25	+.25	+1	

Weight matrix

Pattern Associator

$$
\begin{array}{cccc|c}
+1 & -1 & -1 & +1 & \\
\hline
-.25 & +.25 & +.25 & -.25 & -1 \\
-.25 & +.25 & +.25 & -.25 & -1 \\
+.25 & -.25 & -.25 & +.25 & +1 \\
+.25 & -.25 & -.25 & +.25 & +1
\end{array}
\quad \textbf{Cat}
$$

Weight matrix



$$
\begin{array}{cccc|c}
-1 & +1 & -1 & +1 & \\
\hline
+.25 & -.25 & +.25 & -.25 & -1 \\
-.25 & +.25 & -.25 & +.25 & +1 \\
-.25 & +.25 & -.25 & +.25 & +1 \\
+.25 & -.25 & +.25 & -.25 & -1
\end{array}
\quad \textbf{Dog}
$$

Weight matrix

Pattern Associator

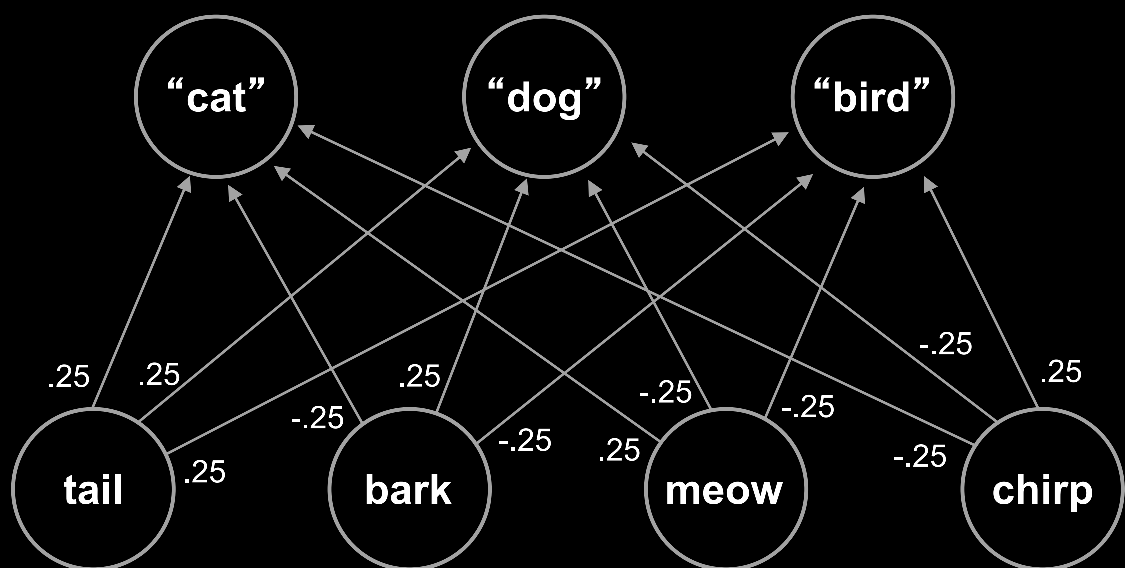
Mathemagic!

 +1	-1	-1	+1			
 -1	+1	-1	+1			
					Cat	**Dog**
0	0	+.5	-.5		-1	-1
-.5	+.5	0	0		-1	+1
0	0	-.5	+.5		+1	+1
+.5	-.5	0	0		+1	-1

Pattern Associator

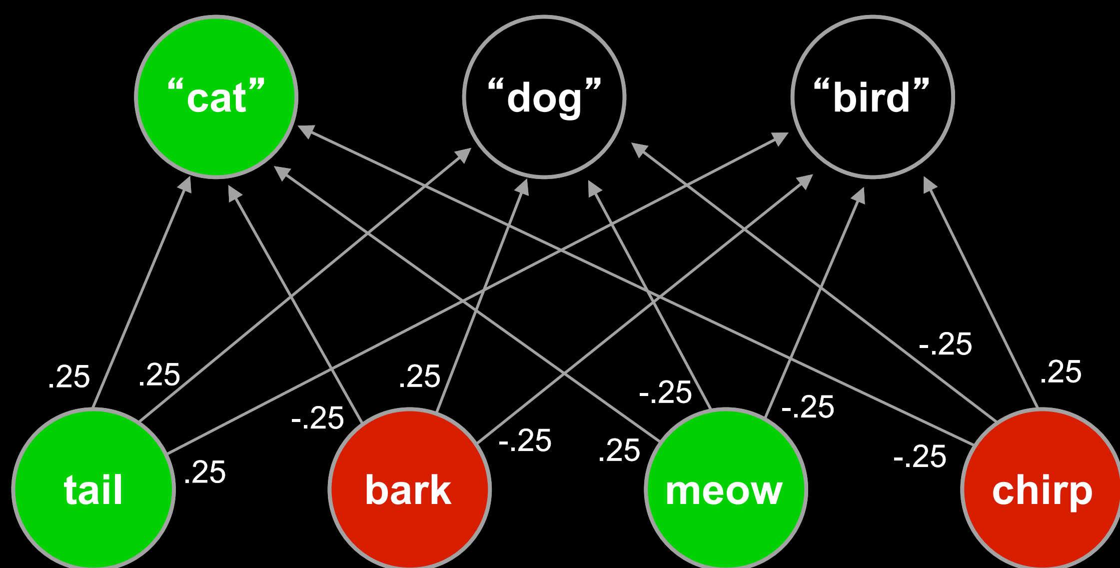
Pattern Associator

ε=.25	Tail	Bark	Meow	Chirp	“Cat”	“Dog”	“Bird”
Cat	+1	–1	+1	–1	+1	0	0
Dog	+1	+1	–1	–1	0	+1	0
Bird	+1	–1	–1	+1	0	0	+1



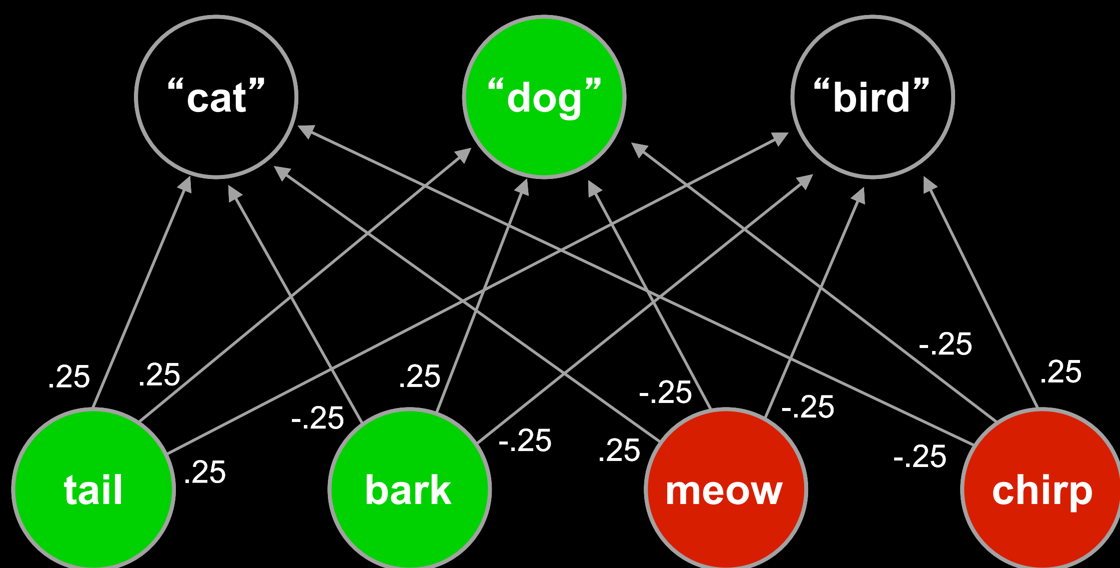
Pattern Associator

ε=.25	*Input* Tail	Bark	Meow	Chirp	*Output* "Cat"	"Dog"	"Bird"
Cat	+1	−1	+1	−1	+1	0	0
Dog	+1	+1	−1	−1	0	+1	0
Bird	+1	−1	−1	+1	0	0	+1



Pattern Associator

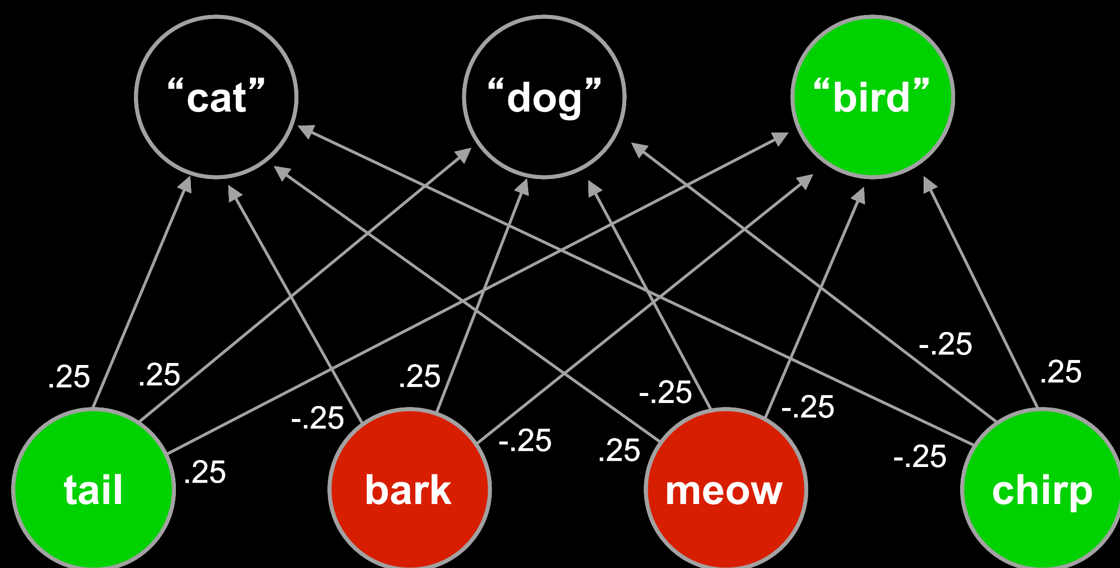
	Input				*Output*		
ε=.25	Tail	Bark	Meow	Chirp	"Cat"	"Dog"	"Bird"
Cat	+1	−1	+1	−1	+1	0	0
Dog	**+1**	**+1**	**−1**	**−1**	**0**	**+1**	**0**
Bird	+1	−1	−1	+1	0	0	+1



Pattern Associator

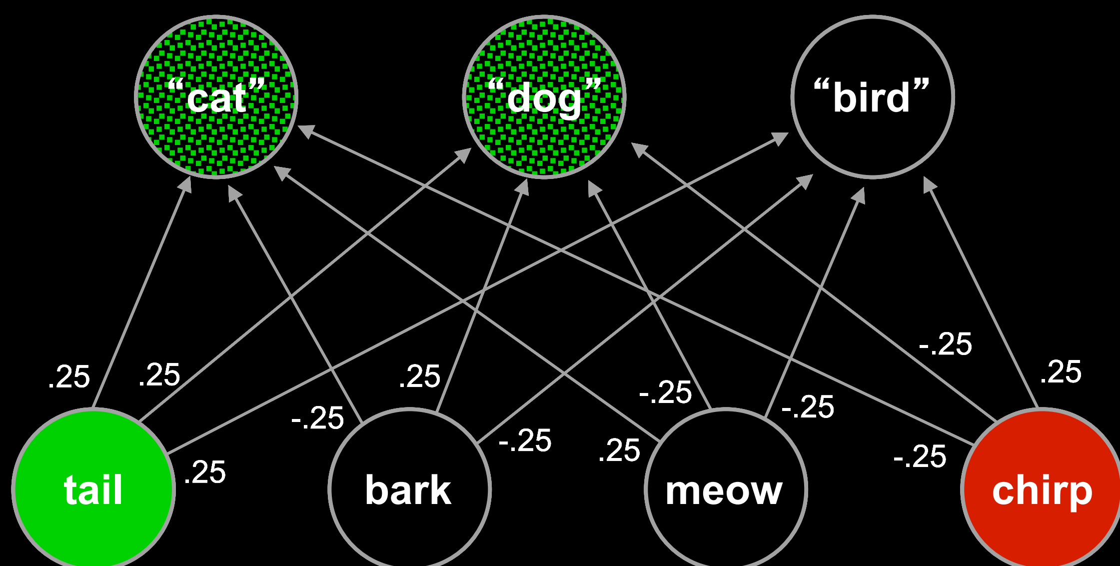
Input *Output*

ε=.25	Tail	Bark	Meow	Chirp	"Cat"	"Dog"	"Bird"
Cat	+1	−1	+1	−1	+1	0	0
Dog	+1	+1	−1	−1	0	+1	0
Bird	**+1**	**−1**	**−1**	**+1**	**0**	**0**	**+1**



Pattern Associator

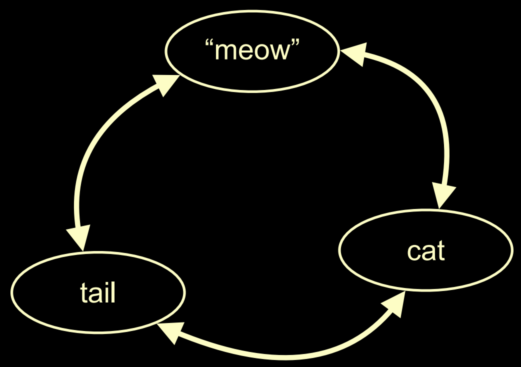
	Input				*Output*		
ε=.25	Tail	Bark	Meow	Chirp	“Cat”	“Dog”	“Bird”
Cat	+1	-1	+1	-1	+1	0	0
Dog	+1	+1	-1	-1	0	+1	0
Bird	+1	-1	-1	+1	0	0	+1
Fox	**+1**	**0**	**0**	**-1**	**+.5**	**+.5**	**0**
	Half-way between Cat and Dog				Output is blend		



Auto-associator

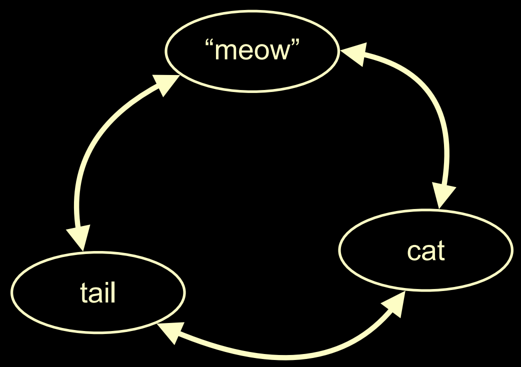
Auto-associator

- **Auto-associator: pattern completion:**



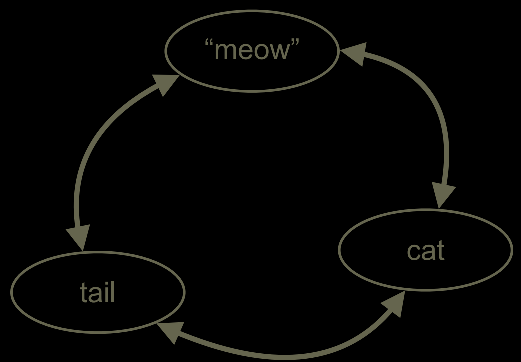
Auto-associator

- **Auto-associator: pattern completion:**
  - **Network that learns associations among parts given a partial pattern, it can complete the pattern**

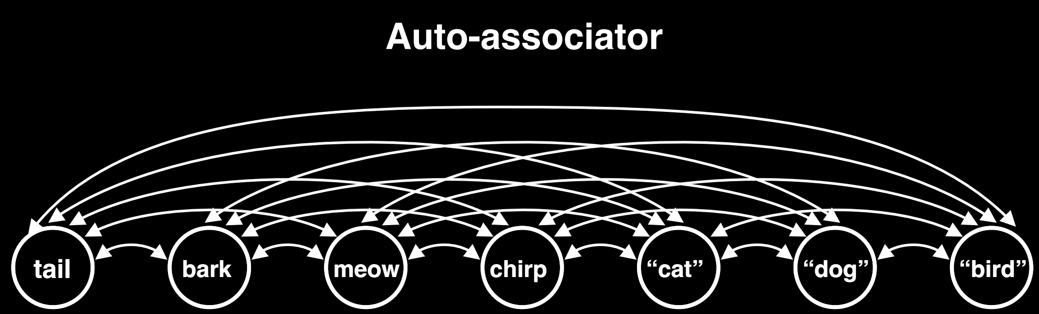


Auto-associator

- **Auto-associator: pattern completion:**
  - Network that learns associations among parts given a partial pattern, it can complete the pattern

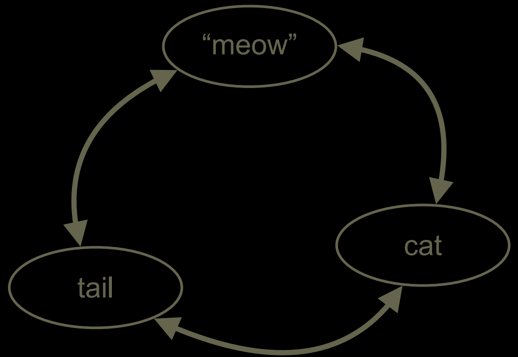


- **Pattern associators are really just a special case of "auto-associators":**

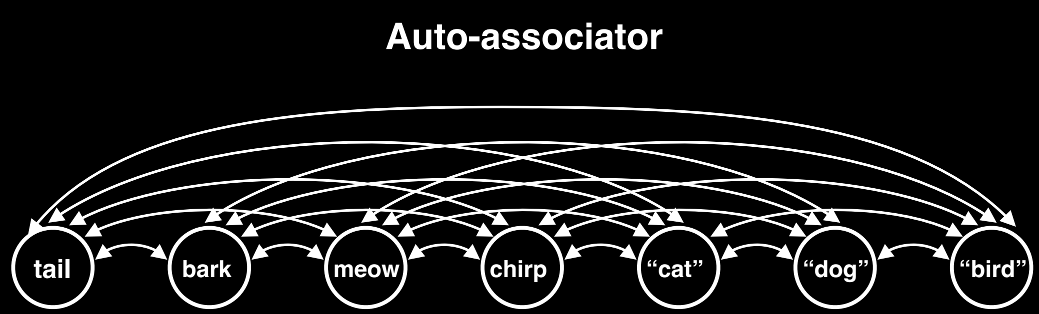


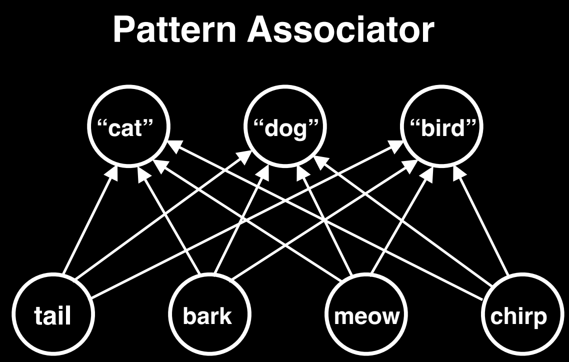
Auto-associator

- **Auto-associator: pattern completion:**
  - **Network that learns associations among parts given a partial pattern, it can complete the pattern**



- **Pattern associators are really just a special case of “auto-associators”:**
  - **have uni-directional connections from inputs and outputs some units have been labeled as “input” and some units have been labeled as “output”**





Associative Learning and Memory

Associative Learning and Memory

- **You only know what you know:**
  - **If a test input pattern *doesn't overlap* with** *(is orthogonal to)* **to all trained patterns *no patterns* will become active**
    (since test pattern is not similar to any of the trained patterns)

Associative Learning and Memory

- **You only know what you know:**
  - **If a test input pattern *doesn't overlap* with** *(is orthogonal to)* **to all trained patterns *no patterns* will become active**
    (since test pattern is not similar to any of the trained patterns)

- **Total recall:**
  - **If a test input is similar to *one* of the trained patterns and doesn't overlap with any of the others, the output will be a *scaled version* of the output for the one to which it is similar**
    (scaled by the similarity to that pattern)

Associative Learning and Memory

- **You only know what you know:**
  - **If a test input pattern *doesn't overlap* with** *(is orthogonal to)* **to all trained patterns *no patterns* will become active**
    (since test pattern is not similar to any of the trained patterns)

- **Total recall:**
  - **If a test input is similar to *one* of the trained patterns and doesn't overlap with any of the others, the output will be a *scaled version* of the output for the one to which it is similar**
    (scaled by the similarity to that pattern)

- **Reconstructive memory:**
  - **If a test input is similar to *two or more* of the trained patterns the output will be a *blend* of the training patterns to which it is similar**
    (with the contribution of each weighted by the similarity)

Associative Learning and Memory

- **You only know what you know:**
  - **If a test input pattern *doesn't overlap* with** *(is orthogonal to)* **to all trained patterns *no patterns* will become active**
    (since test pattern is not similar to any of the trained patterns)

- **Total recall:**
  - **If a test input is similar to *one* of the trained patterns and doesn't overlap with any of the others, the output will be a *scaled version* of the output for the one to which it is similar**
    (scaled by the similarity to that pattern)

- **Reconstructive memory:**
  - **If a test input is similar to *two or more* of the trained patterns the output will be a *blend* of the training patterns to which it is similar**
    (with the contribution of each weighted by the similarity)

- **But… correlation is not causation:**
  - **Correlations are not always sufficient to learn *meaningful* associations between patterns of activity**