Memory

- Memory = any persistent effect of experience (not just memorization of facts, events, names, etc.)
- Weights vs activations
- Gradual, integrative cortical learning and priming effects
- Rapid memorization: The hippocampus
- Active memory: prefrontal cortex

Memory: Weights vs Activations

Despite appearances, memory is not unitary.

(shoes; breakfast; sentence)

Memory: Weights vs Activations

Despite appearances, memory is not unitary.
(shoes; breakfast; sentence)

Weights:

- Long-lasting.
- Requires re-activation.
- Wts in diff’t brain systems store different types of memories!

Activations:

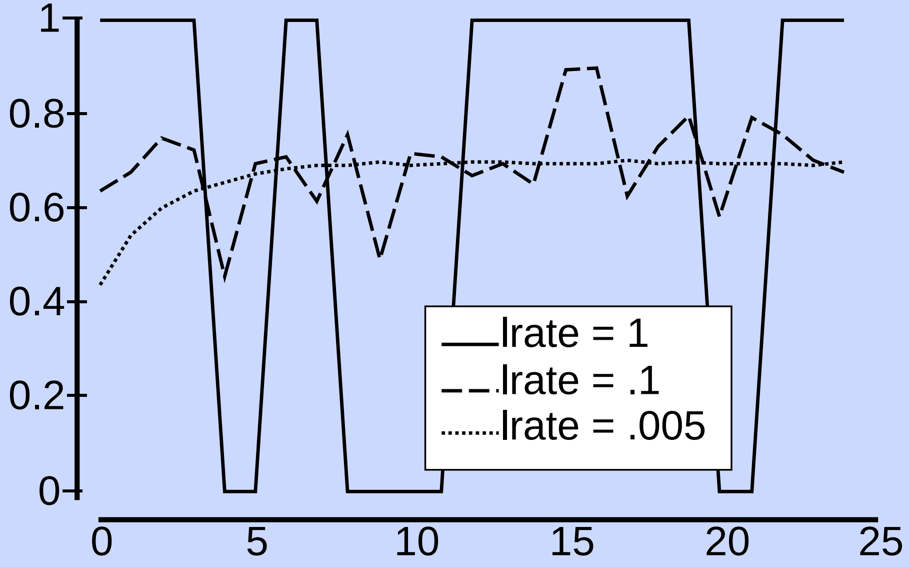
- Short-term.
- Already active, can influence processing.

Weight-based Memories

- Cortex does gradual, integrative learning
- Cortex can learn arbitrary input-output mappings given:
  - multiple passes through the training set
  - a relatively small learning rate

Weight-based Memories

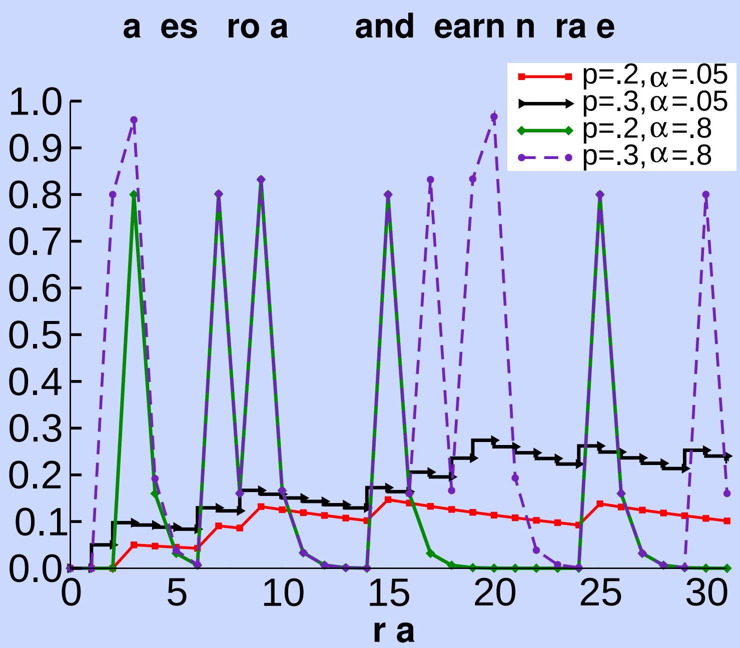
Rapid weight changes causes interference:



Two systems needed:

- Slow learning cortex.
- Rapid learning hippocampus (pattern sep avoids interference).

b. Slow *vs* Fast [Reinforcement] Learning

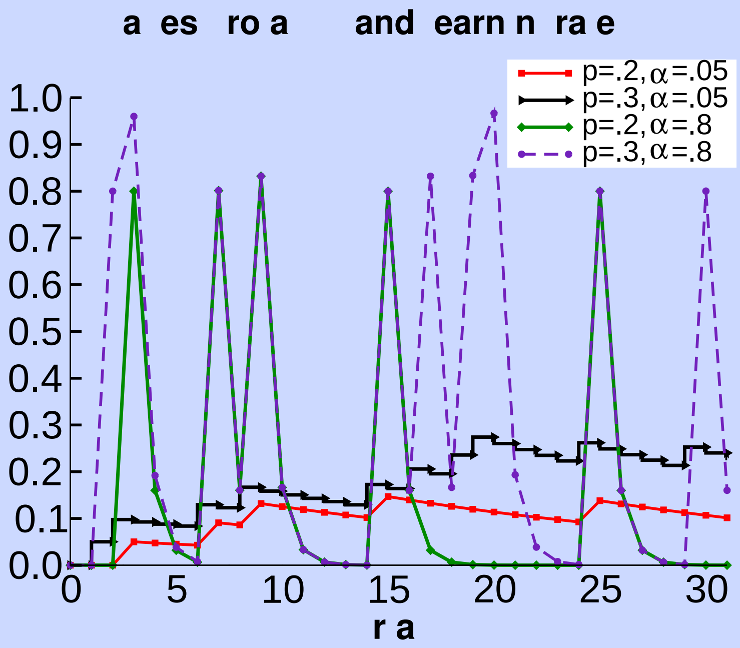


[Reinforcement] Learning must be *slow* to capture best actions that work on average.

But you also have to be able to sensitive to rapid changes in value (e.g., stock market).

Tradeoff solved by 2 systems:
BG learns slowly, PFC relies on (flexible updating of) *activation-based memory*, and can override habitual choices.

b. Slow *vs* Fast [Reinforcement] Learning



[Reinforcement] Learning must be *slow* to capture best actions that work on average.

But you also have to be able to sensitive to rapid changes in value (e.g., stock market).

Tradeoff solved by 2 systems:
BG learns slowly, PFC relies on (flexible updating of) *activation-based memory*, and can override habitual choices.

→ lots of evidence for differential BG and PFC contributions to habitual and rapid action-outcome learning, across species, methods.

Memory: Rapid Learning, Interference, & The Hippocampus

1. AB-AC List Learning

2. The Hippocampus.

AB-AC List Learning

Humans can rapidly learn overlapping associations without too much interference.

Example: learn one set of paired associates (the A-B list):
window-reason

AB-AC List Learning

Humans can rapidly learn overlapping associations without too much interference.

Example: learn one set of paired associates (the A-B list):
window-reason
bicycle-garbage

AB-AC List Learning

Humans can rapidly learn overlapping associations without too much interference.

Example: learn one set of paired associates (the A-B list):
window-reason
bicycle-garbage
... Then, learn overlapping set (the A-C list):
window-locomotive

AB-AC List Learning

Humans can rapidly learn overlapping associations without too much interference.

Example: learn one set of paired associates (the A-B list):
window-reason
bicycle-garbage
... Then, learn overlapping set (the A-C list):
window-locomotive
bicycle-dishtowel

AB-AC List Learning

Then test on AB list:
window- ?

AB-AC List Learning

Then test on AB list:
window- ?
bicycle- ?

AB-AC List Learning

Then test on AB list:
window- ?
bicycle- ?

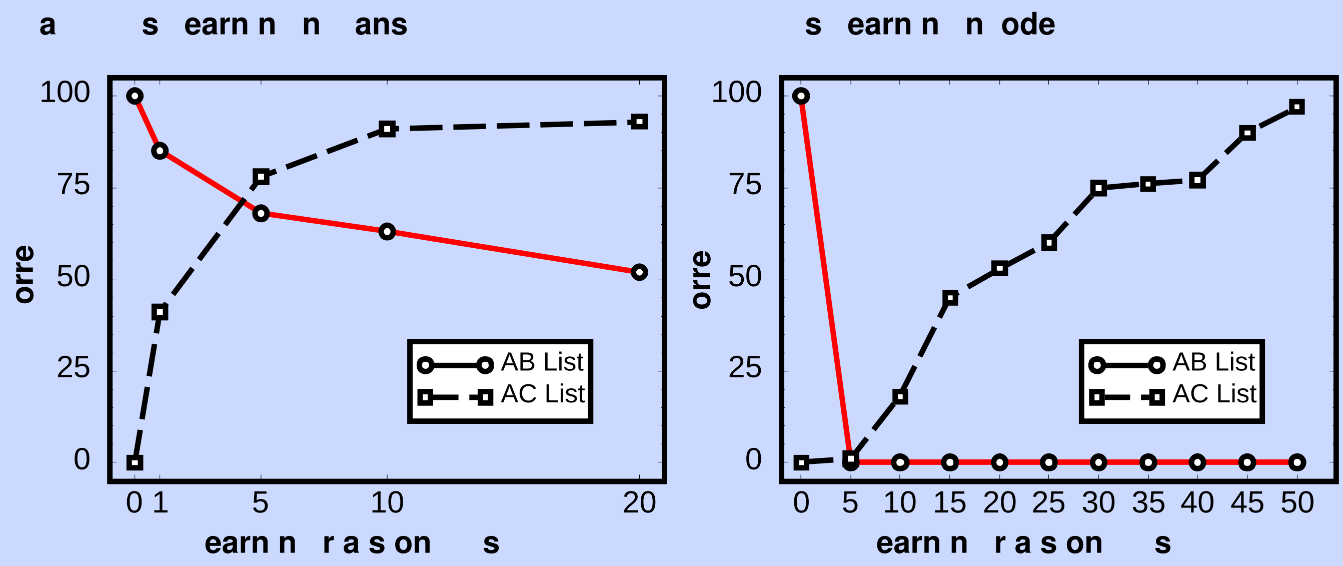
and on AC list:
window- ?

AB-AC List Learning

Then test on AB list:
window- ?
bicycle- ?

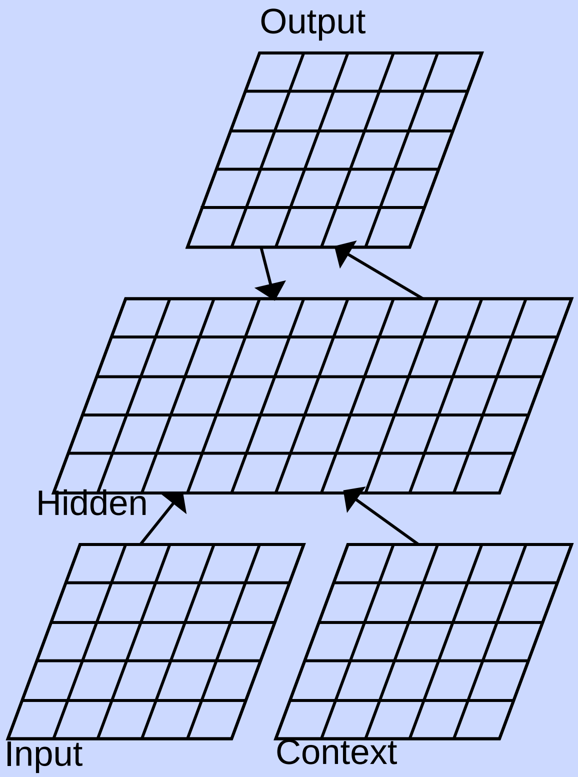
and on AC list:
window- ?
bicycle- ?

AB-AC List Learning



Standard network shows *catastrophic* interference (McCloskey & Cohen, 1989).

AB-AC Exploration

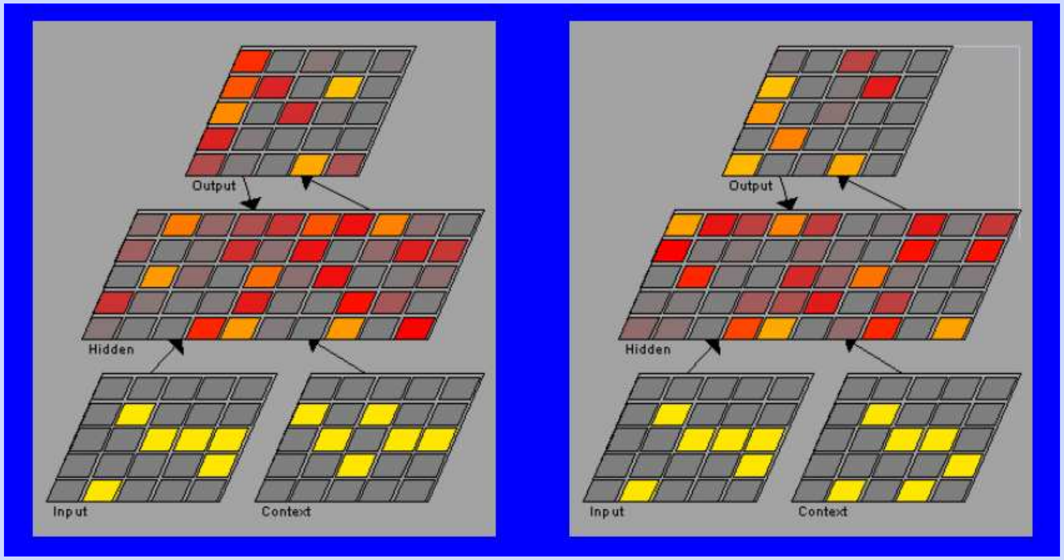


Input = A, Output = B,C

Context differentiates the lists: Each list is associated with a different context pattern

[ab_ac_interference.proj]

AB-AC Simulations: Summary



- There is overlap between the hidden units activated by an input pattern (“window”) in the AB context and units activated by that same pattern in the AC context.
- This causes *interference* (changing weights for one changes weights for the other)
- Can this be fixed?

How to reduce Interference?

- How can we reduce overlap between hidden units activated by patterns in the AB and AC contexts?

How to reduce Interference?

- How can we reduce overlap between hidden units activated by patterns in the AB and AC contexts?

- → Lower the number of units that are activated → increase inhibition (increase g_i)...

How to reduce Interference?

- How can we reduce overlap between hidden units activated by patterns in the AB and AC contexts?
- $\rightarrow$ Lower the number of units that are activated $\rightarrow$ increase inhibition (increase g_i)...

But still need ***different*** units to be active for AB and AC inputs...

How to reduce Interference?

- How can we reduce overlap between hidden units activated by patterns in the AB and AC contexts?

- → Lower the number of units that are activated → increase inhibition (increase g_i)...

But still need ***different*** units to be active for AB and AC inputs...

- → Increase relative weight scale of the context layer so that hidden units "pay more attention" to it

- → Also increase initial weight *variance*: Lowers the odds that a unit will "like" both the AB and AC version of a pattern

AB-AC Exploration: Summary

- Note that ***even with all these changes***, interference gets only slightly better...
- Also network learns much slower than people do...

AB-AC Exploration: Summary

- Note that ***even with all these changes***, interference gets only slightly better...

- Also network learns much slower than people do...

  → Increase learning rate?

AB-AC Exploration: Summary

- Note that ***even with all these changes***, interference gets only slightly better...

- Also network learns much slower than people do...

  → Increase learning rate?

- This speeds up learning, but makes interference **worse**!

- Also, by changing all these parameters, cortex can no longer generalize (requires overlapping distributed representations)

AB-AC Exploration: Summary

- Note that ***even with all these changes***, interference gets only slightly better...

- Also network learns much slower than people do...

  → Increase learning rate?

- This speeds up learning, but makes interference **worse**!

- Also, by changing all these parameters, cortex can no longer generalize (requires overlapping distributed representations)

→ *Trade-off:* Must need another brain system!

Memory

Memory

Memory is not unitary.

1. Weights versus activations.

2. Specialized neural systems: computational tradeoffs.

Memory

Memory is not unitary.

1. Weights (long-lasting, requires re-activation) versus activations (short-term, already active, can influence processing).

2. Specialized neural systems: computational tradeoffs. Cortex shows priming, but suffers catastrophic interference.

Memory

Memory is not unitary.

1. Weights (long-lasting, requires re-activation) versus activations (short-term, already active, can influence processing).

2. Specialized neural systems: computational tradeoffs. Cortex shows priming, but suffers catastrophic interference.

   Abandon neural network models?

Hippo To the Rescue

Two specialized, complementary systems resolve fundamental tradeoff:

The hippocampus can learn rapidly without interference by using sparse, pattern-separated representations!

Meanwhile, cortex slowly learns overlapping representations of similarity structure & regularities, semantic knowledge.

Hippo To the Rescue

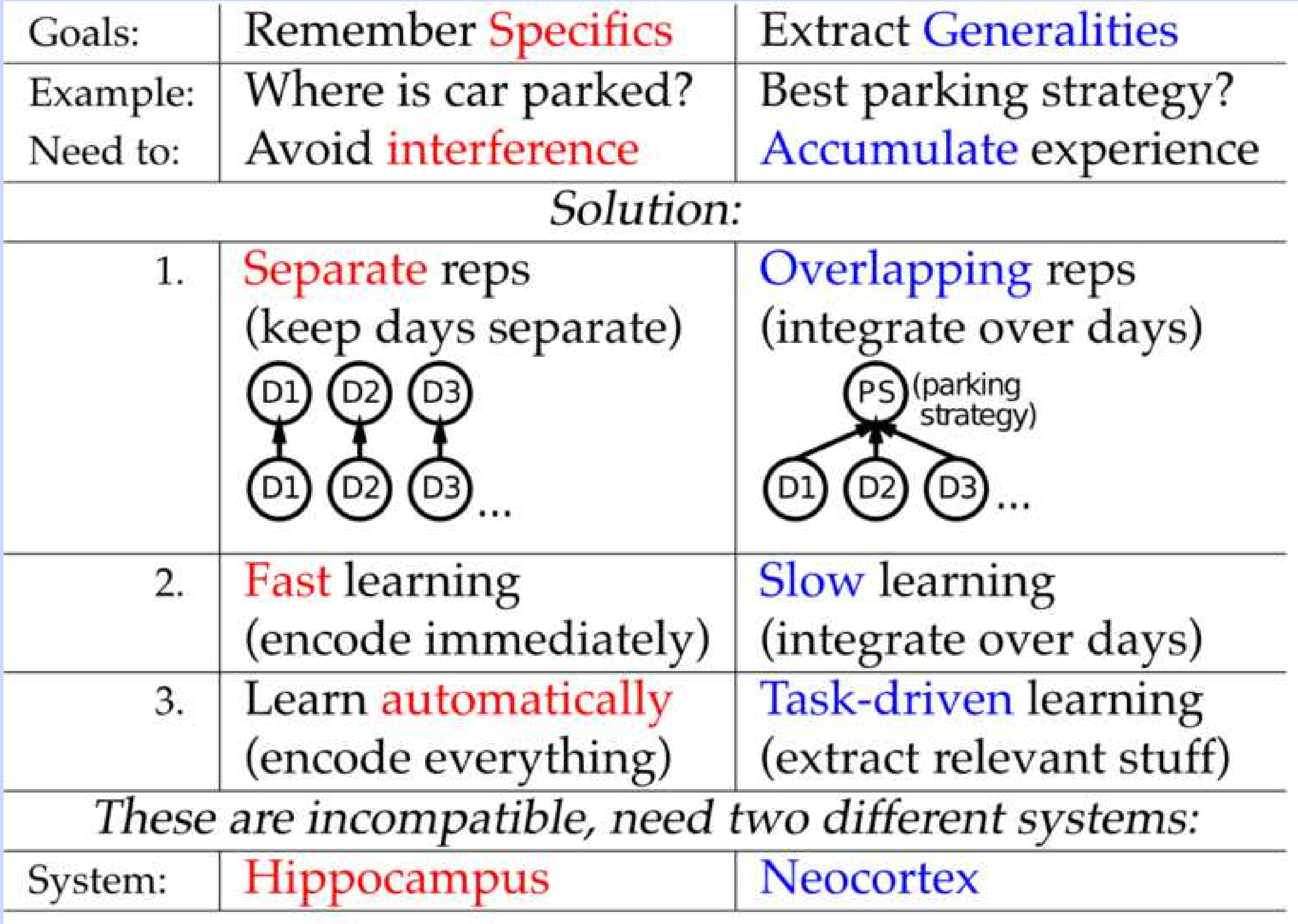
Two specialized, complementary systems resolve fundamental tradeoff:

The hippocampus can learn rapidly without interference by using sparse, pattern-separated representations!

Meanwhile, cortex slowly learns overlapping representations of similarity structure & regularities, semantic knowledge.

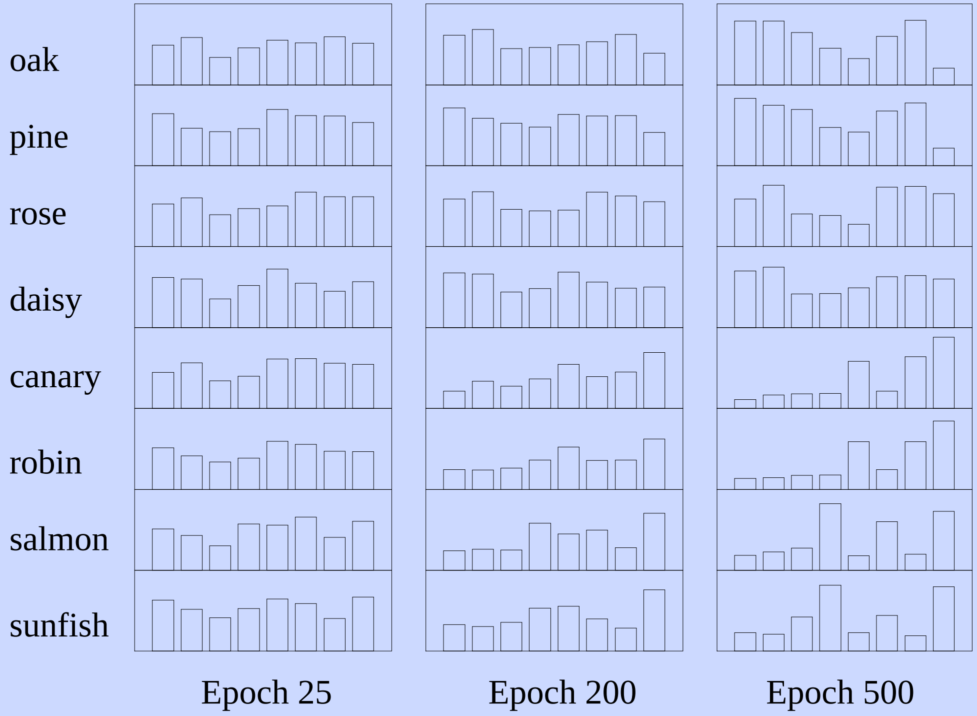
e.g. “one small step for man” 9/11, etc

Complementary Learning Systems



Goals:	Remember Specifics	Extract Generalities
Example: Need to:	Where is car parked? Avoid interference	Best parking strategy? Accumulate experience
	Solution:	
1.	Separate reps (keep days separate)	Overlapping reps (integrate over days)
2.	Fast learning (encode immediately)	Slow learning (integrate over days)
3.	Learn automatically (encode everything)	Task-driven learning (extract relevant stuff)
	These are incompatible, need two different systems:	
System:	Hippocampus	Neocortex

Systematic Overlap Develops by Slowly Integrating over Experience



Effects of hippocampal damage in Amnesia

Amnesics show:

- spared implicit memory, skill learning (without recall)

Effects of hippocampal damage in Amnesia

Amnesics show:

- spared implicit memory, skill learning (without recall)
  small adaptive adjustments in synaptic weights

Effects of hippocampal damage in Amnesia

Amnesics show:

- spared implicit memory, skill learning (without recall)
  small adaptive adjustments in synaptic weights

- intact repetition priming for existing associations (table-chair) but not for arbitrary novel pairs of words (locomotive-spoon)

Effects of hippocampal damage in Amnesia

Amnesics show:

- spared implicit memory, skill learning (without recall)

  small adaptive adjustments in synaptic weights

- intact repetition priming for existing associations (table-chair) but not for arbitrary novel pairs of words (locomotive-spoon)

  small cortical adjustments can prime existing reps but not novel conjunction

Effects of hippocampal damage in Amnesia

Amnesics show:

- spared implicit memory, skill learning (without recall)

  small adaptive adjustments in synaptic weights

- intact repetition priming for existing associations (table-chair) but not for arbitrary novel pairs of words (locomotive-spoon)

  small cortical adjustments can prime existing reps but not novel conjunction

- remote memories spared but recent ones completely forgotten

Effects of hippocampal damage in Amnesia

Amnesics show:

- spared implicit memory, skill learning (without recall)

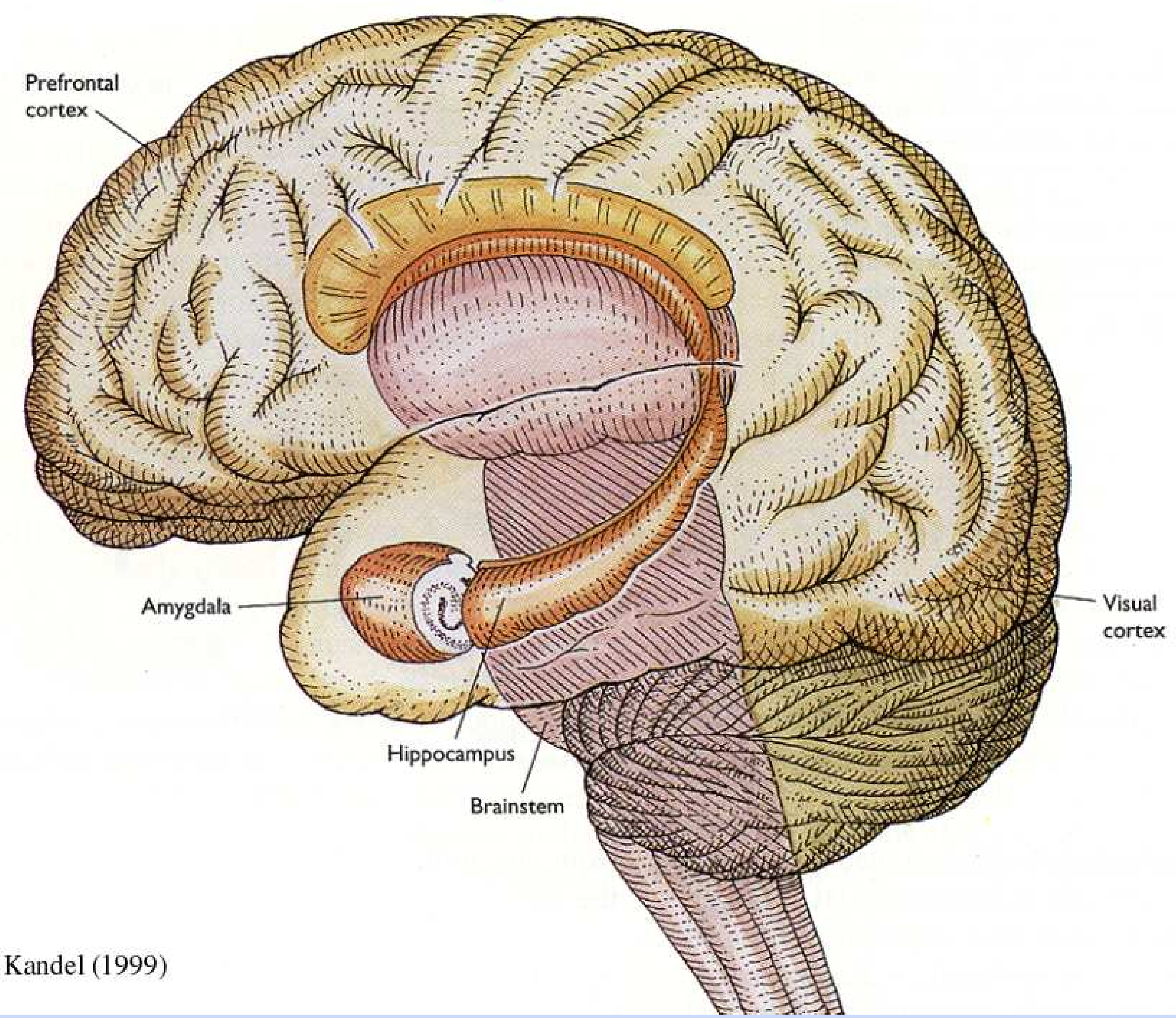
  small adaptive adjustments in synaptic weights

- intact repetition priming for existing associations (table-chair) but not for arbitrary novel pairs of words (locomotive-spoon)

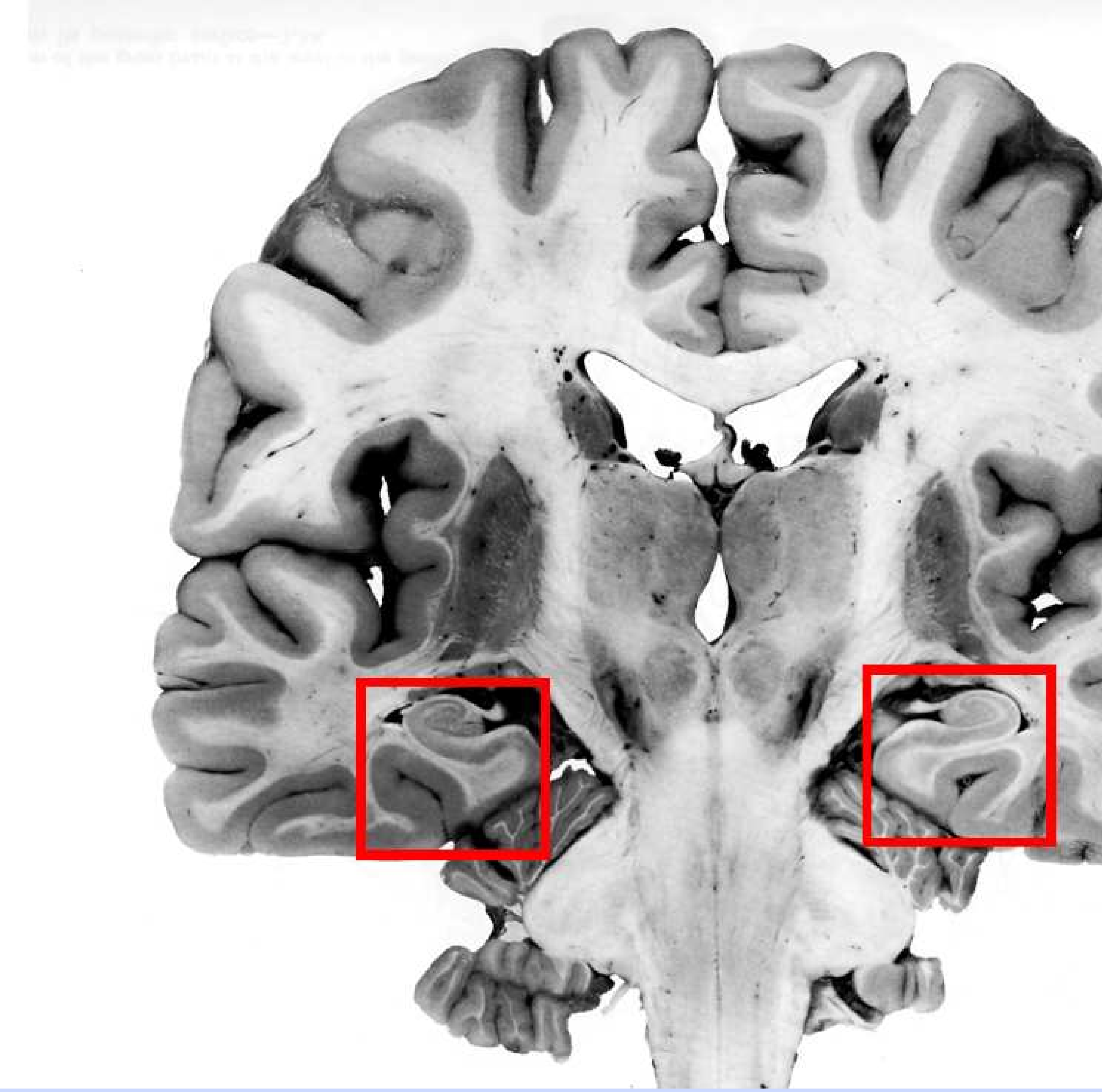
  small cortical adjustments can prime existing reps but not novel conjunction

- remote memories spared but recent ones completely forgotten

  "Consolidation": reactivation of memories across multiple contexts, sleep, etc



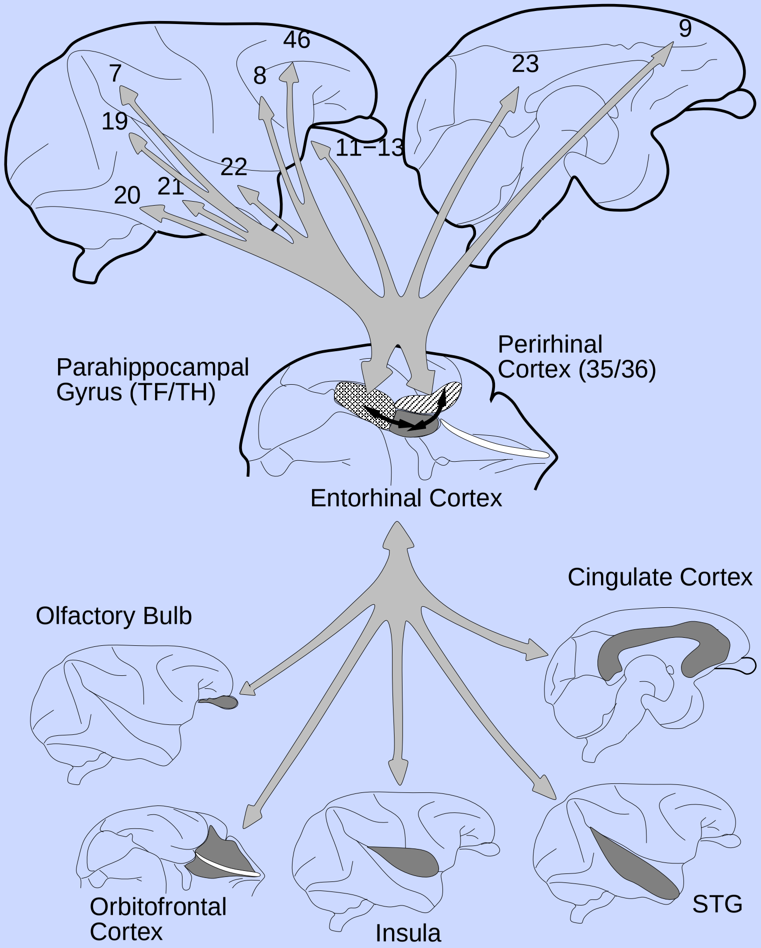
from Squire & Kandel (1999)



from DeArmond,
Fusco, & Dewey
(1989)

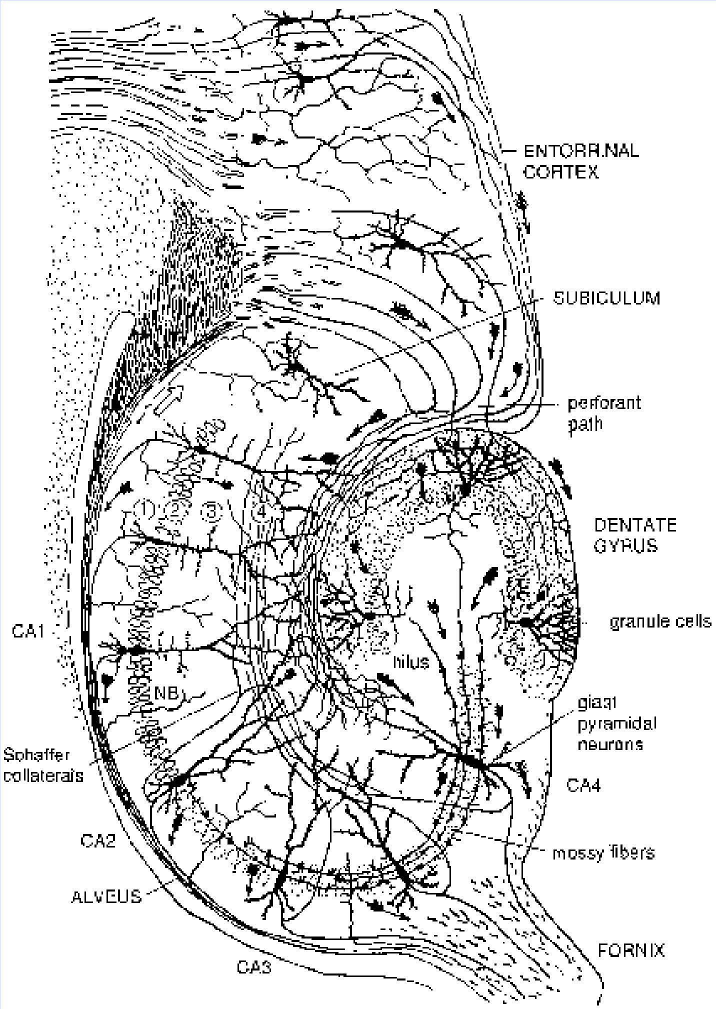
(Greek: hippo=horse, kampos=sea monster).

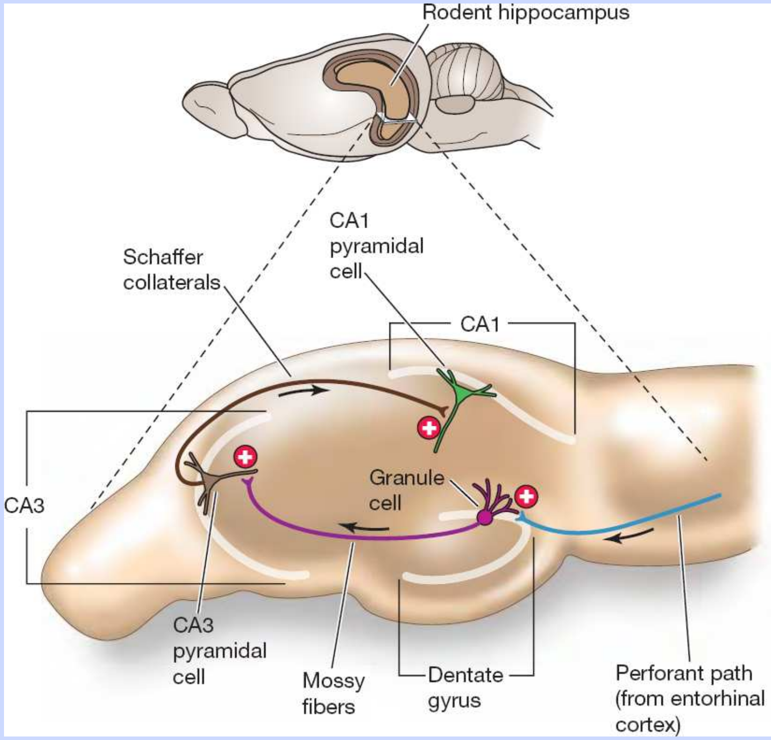
Hippo = King-of-the-Cortex

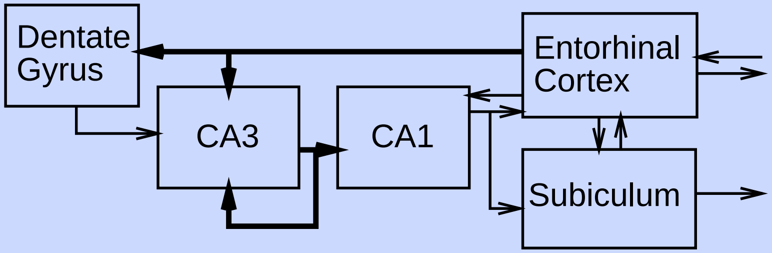


Hippo ***binds*** together multiple cortical representations into one coherent memory

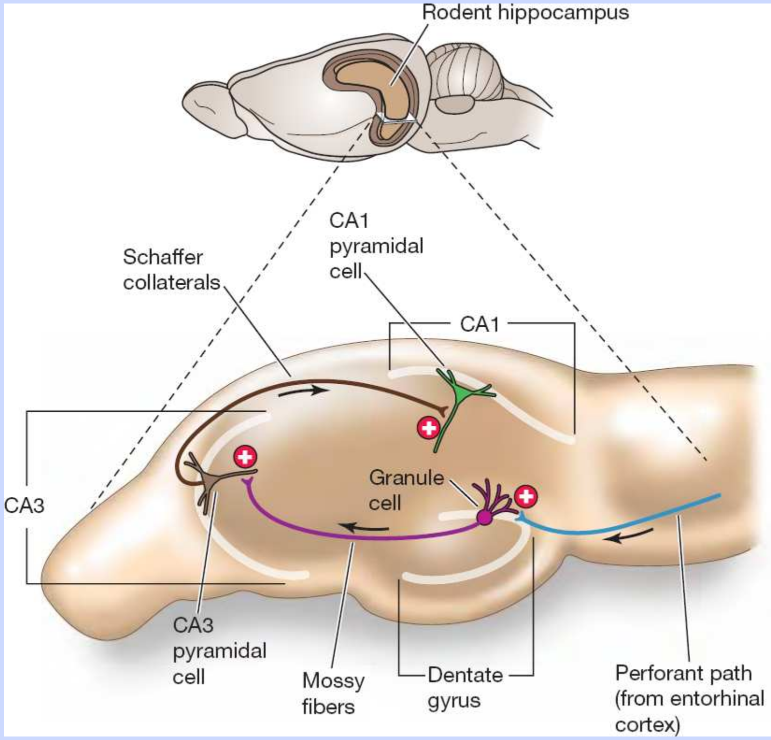
Hippocampal Anatomy

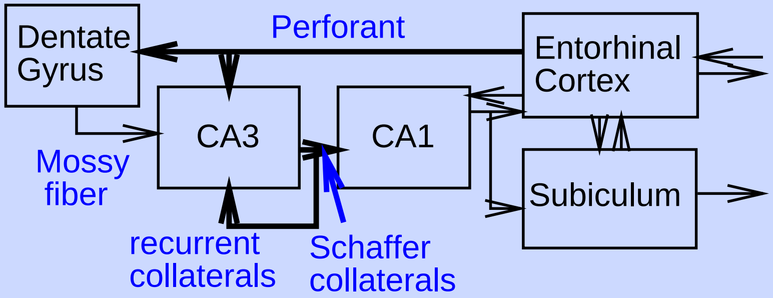






Hippocampal Anatomy




Dentate Gyrus

Perforant

Entorhinal Cortex

CA3

CA1

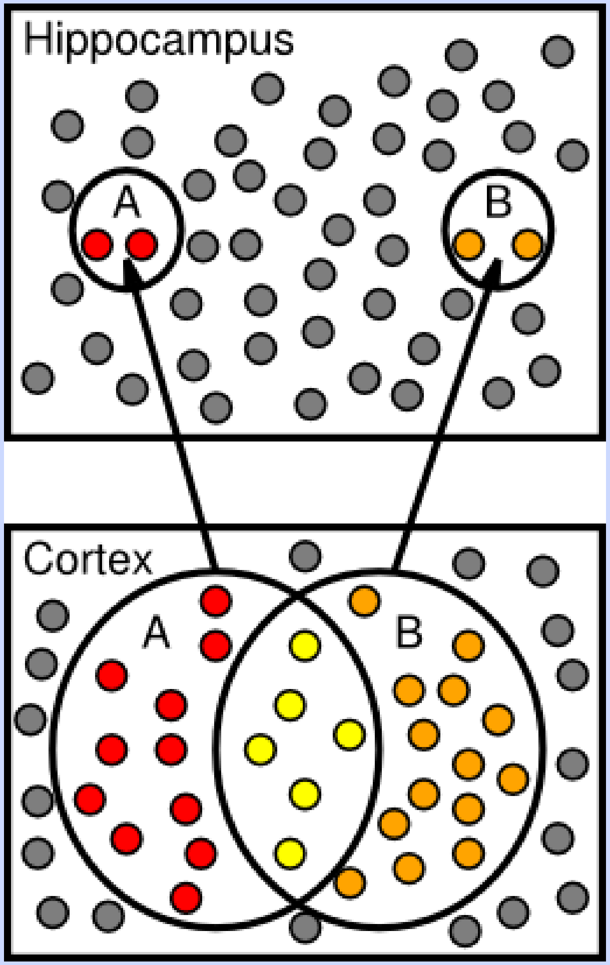
Subiculum

Mossy fiber

recurrent collaterals

Schaffer collaterals

Pattern Separation & Conjunctions



Explaining Pattern Separation

How does the hippocampus assign distinct representations to similar inputs?

Explaining Pattern Separation

How does the hippocampus assign distinct representations to similar inputs?

- *Partial connectivity*: units are specialized for responding to a particular set of input features

- *Sparse activity*: fierce inhibitory competition

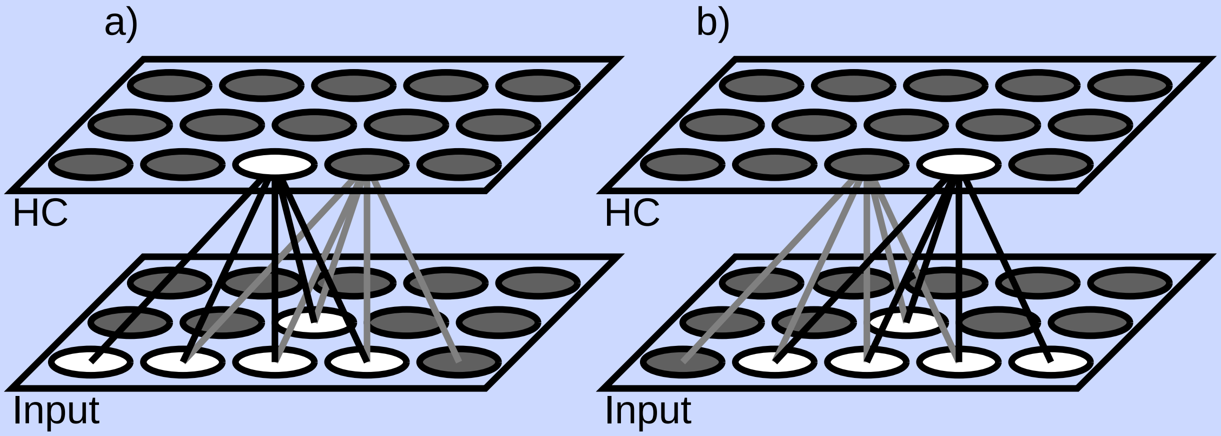
- Units only survive this competition if they receive a very large amount of excitatory input

- Units only fire if all features they detect are present in the input

→ Units represent *conjunctions* of features

Pattern Separation & Conjunctions: Space and episodes

Pattern Separation & Conjunctions: Space and episodes



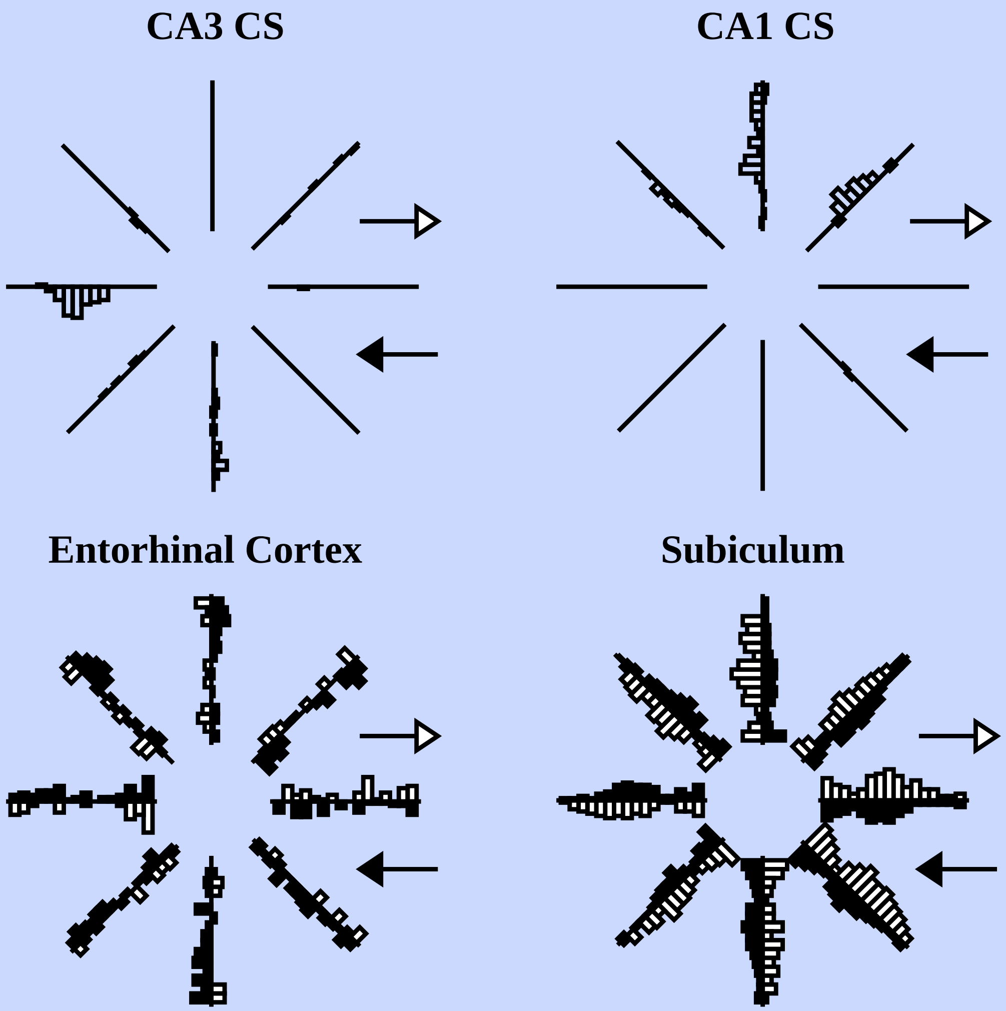
- Here each HC unit connected to 5 inputs; $k = 1$

- Changing one input unit causes a different HC unit to win!

Sparse Activity

	Rat		Model	
Area	Neurons	Pct Act	Units	Pct Act
EC	200,000	7.0	144	25.0
DG	1,000,00	0.5	625	1.0
CA3	160,000	2.5	240	5.0
CA1	250,000	2.5	384	9.4

Sparse Activity



CA3 CS

CA1 CS

Entorhinal Cortex

Subiculum

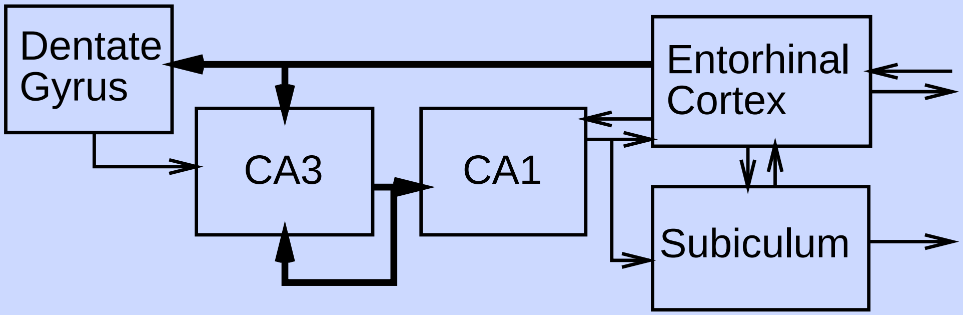
The Flip Side of Separation: Pattern Completion

College friend example: “This one time, at this one party...”

The Flip Side of Separation: Pattern Completion

College friend example: “This one time, at this one party...”

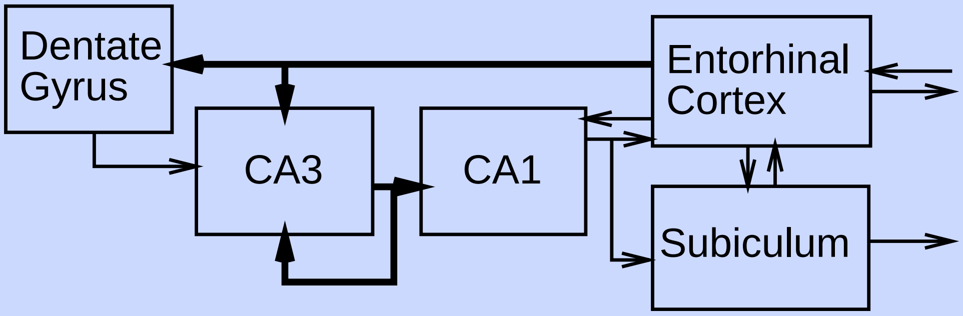
Pattern completion in CA3 activates corresponding CA1 rep, which reinstates original EC pattern...
→ “You told me this already!”.



The Flip Side of Separation: Pattern Completion

College friend example: "This one time, at this one party..."

Pattern completion in CA3 activates corresponding CA1 rep, which reinstates original EC pattern...
→ "You told me this already!".

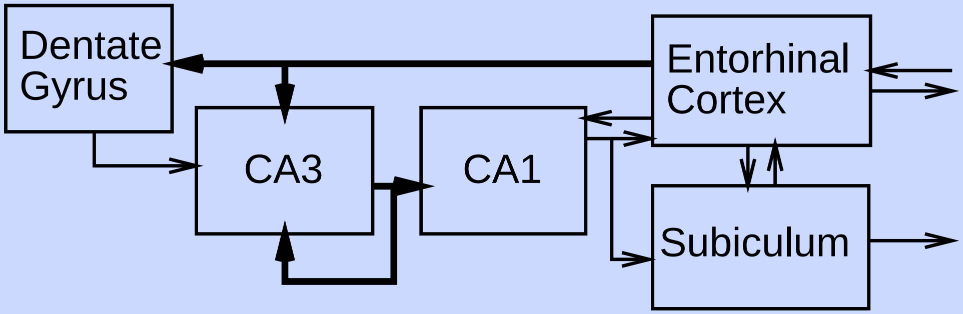


How does your hippo 'know' whether to store new memory and keep it separate, or instead complete to an existing memory?

The Flip Side of Separation: Pattern Completion

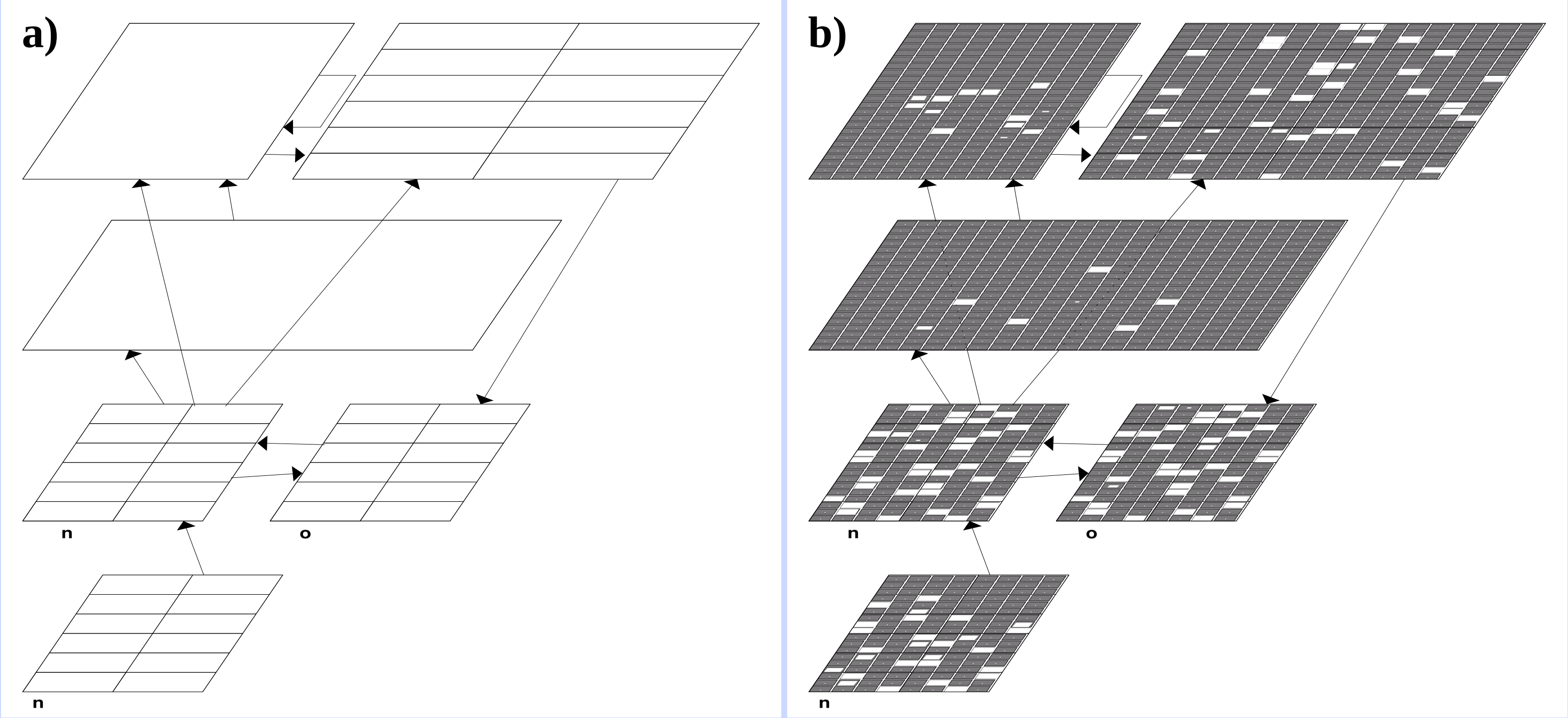
College friend example: “This one time, at this one party...”

Pattern completion in CA3 activates corresponding CA1 rep,
which reinstates original EC pattern...
→ “You told me this already!”.

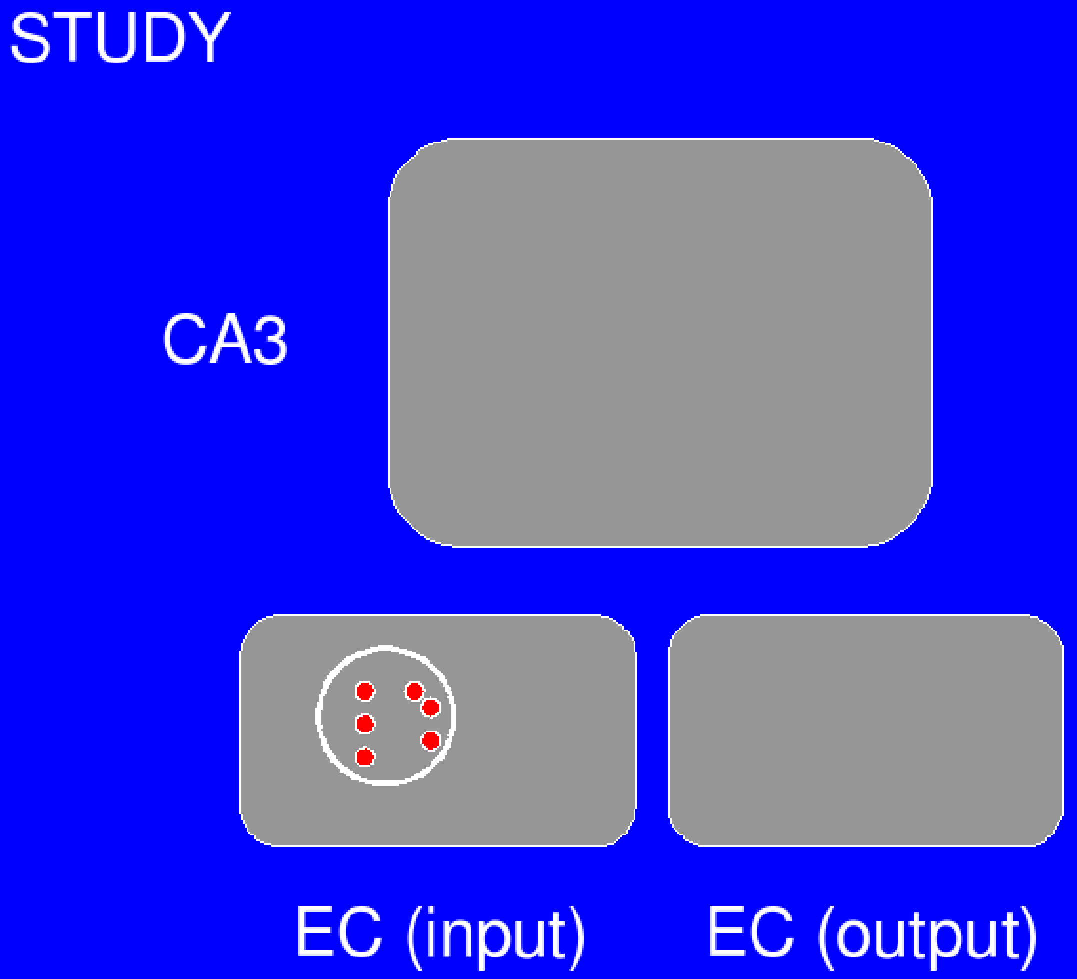


How does your hippo ’know’ whether to store new memory and keep it separate, or instead complete to an existing memory?
→ hippo designed to minimize this tradeoff (LTP in CA3 supports pat complet while LTD supports pat sep; O’Reilly & McLelland ’94).

The Model

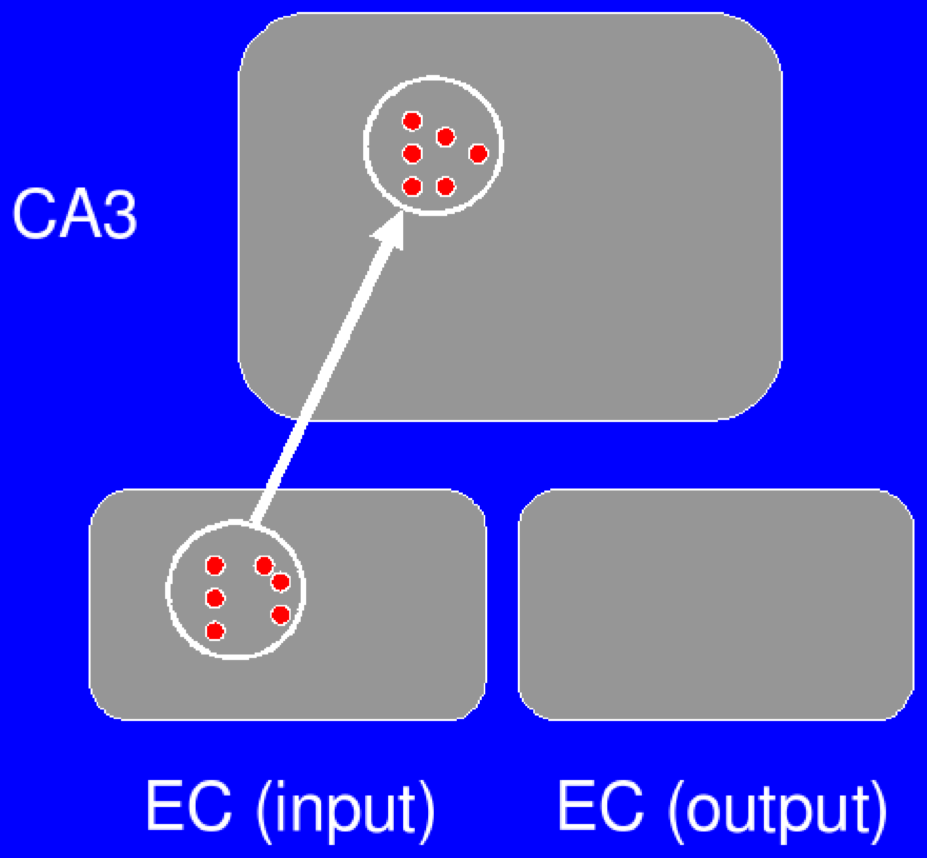


Hippo binding



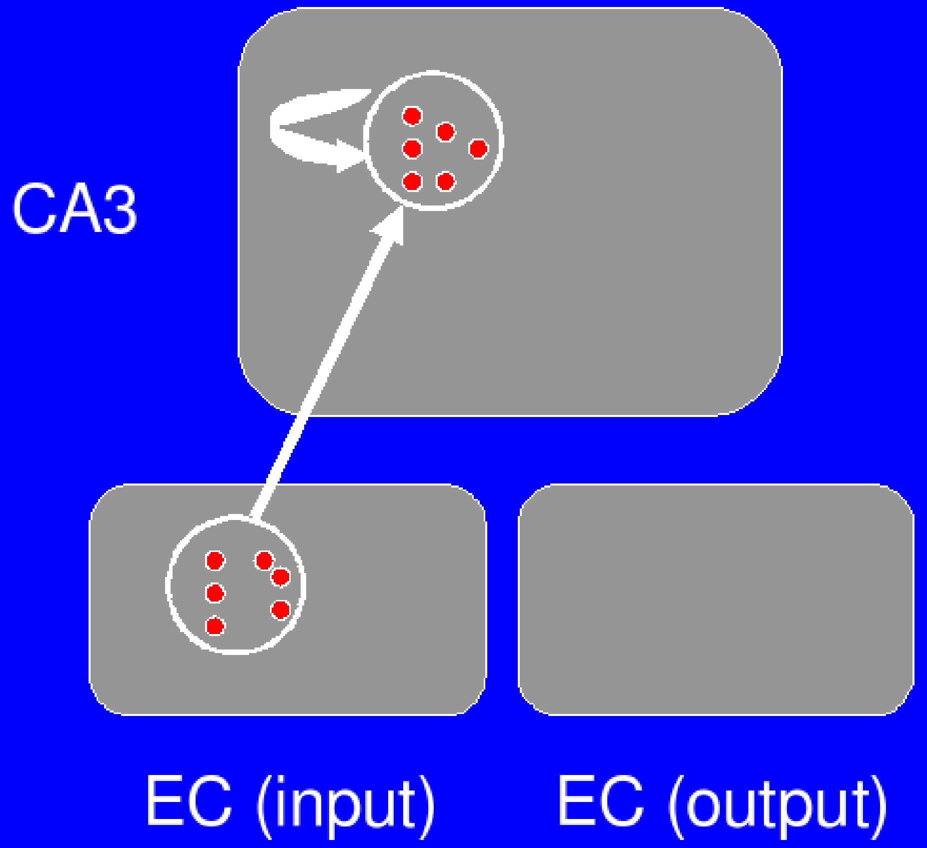
Hippo binding

STUDY



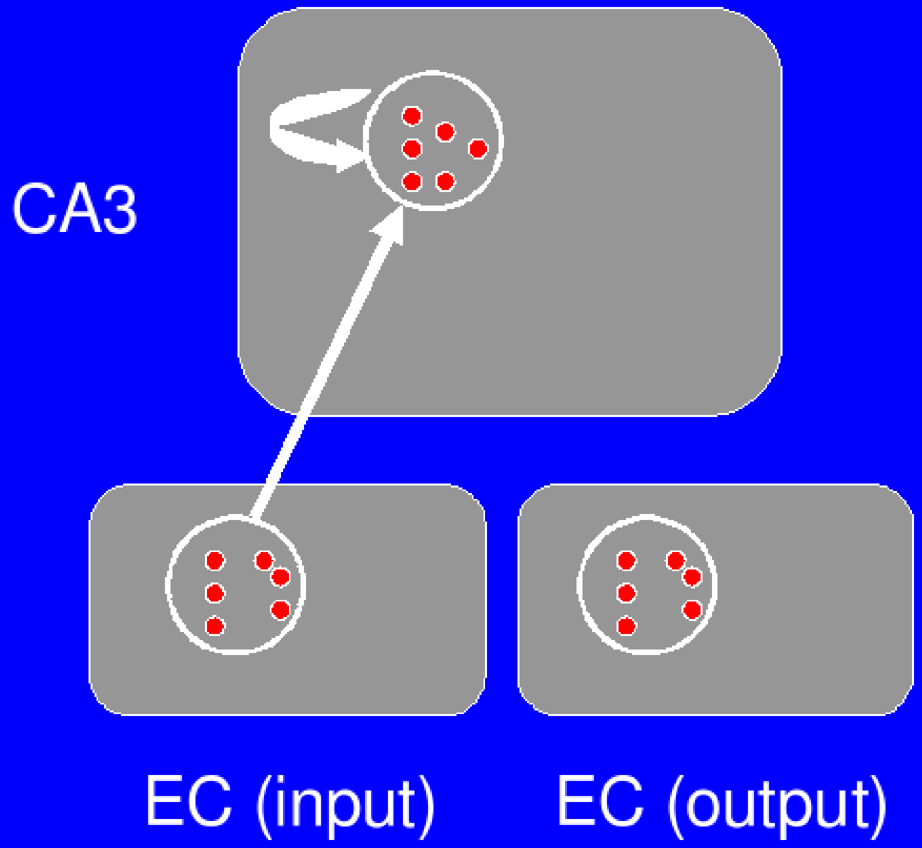
Hippo binding

STUDY

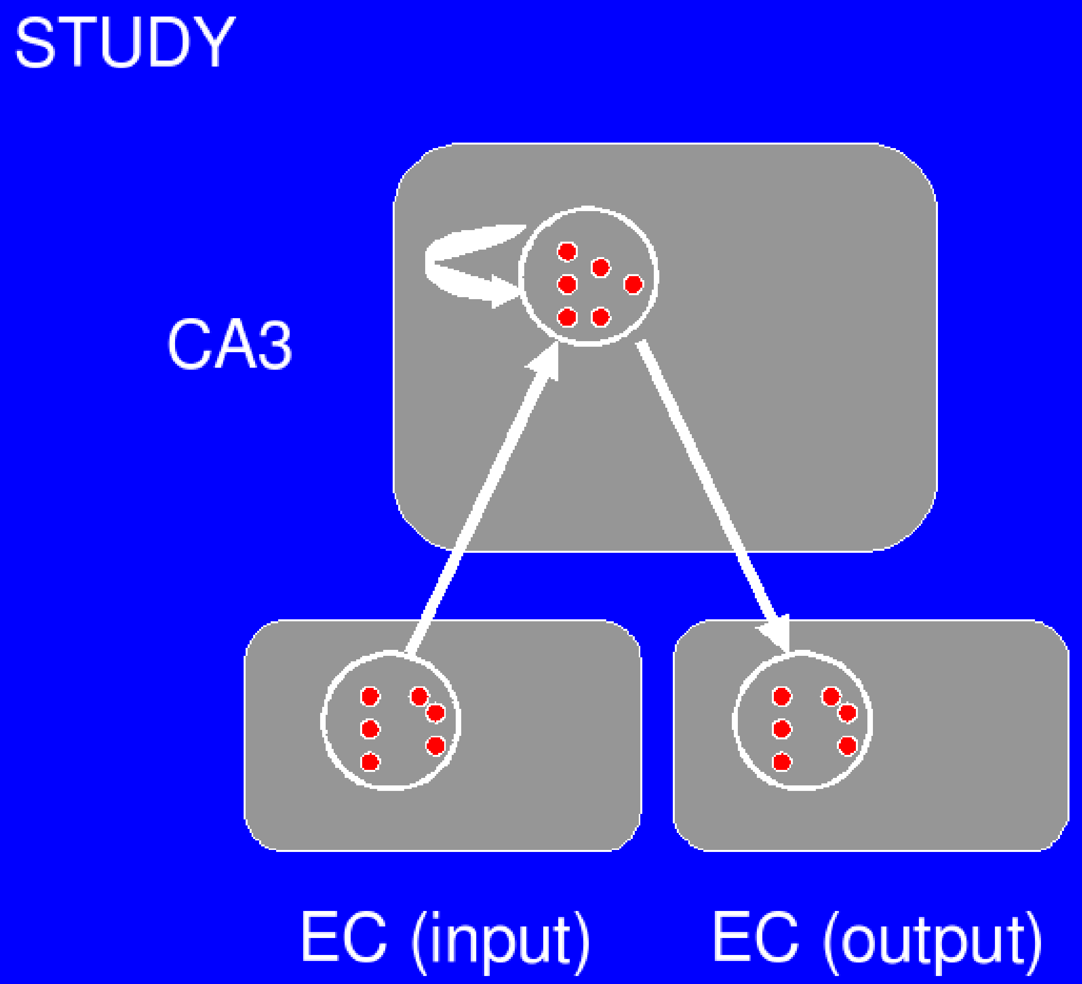


Hippo binding

STUDY

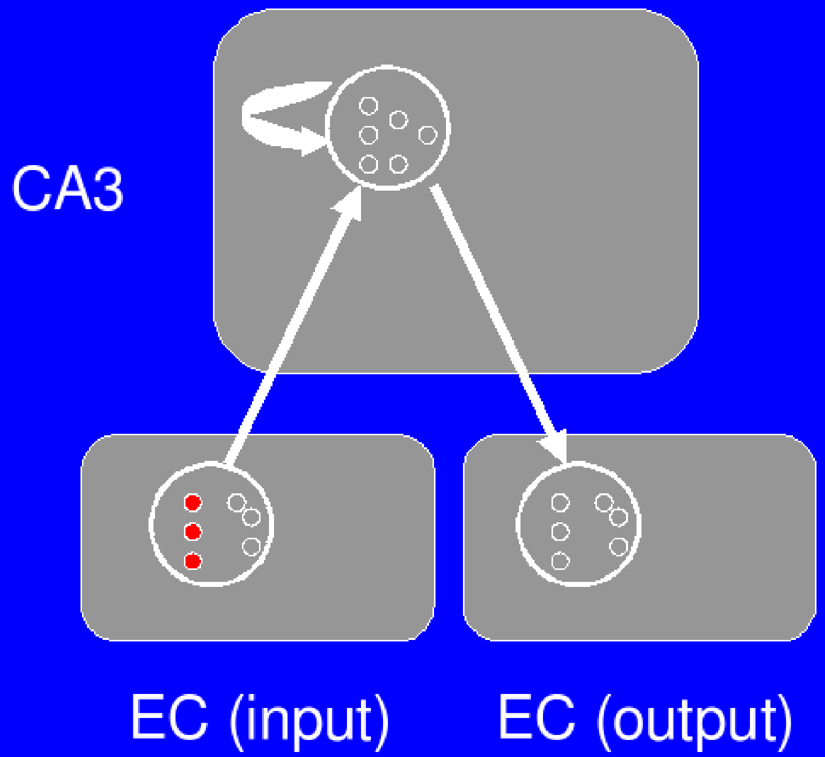


Hippo binding



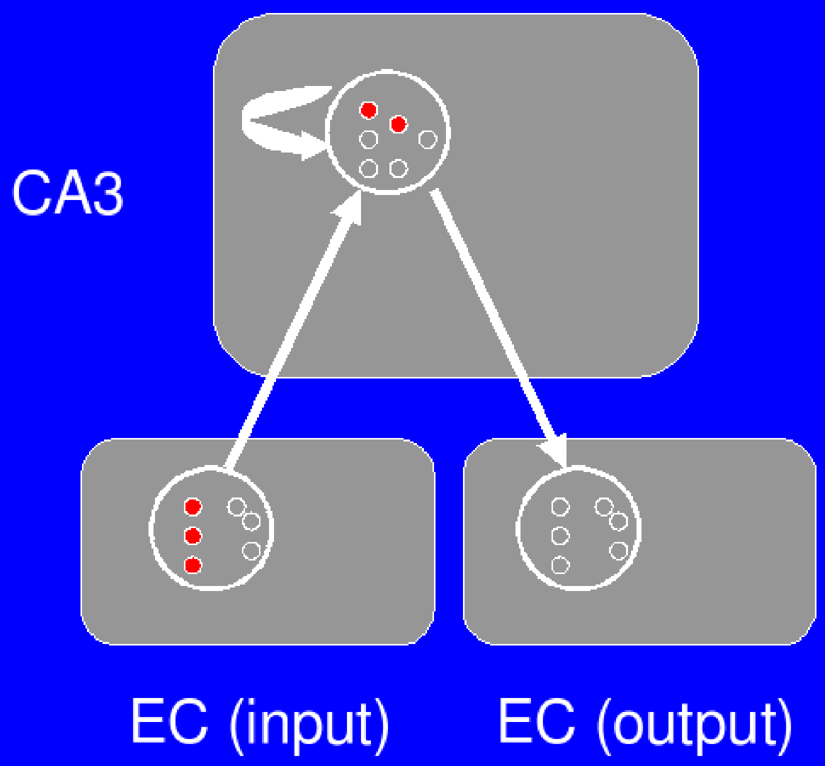
Pattern Completion

TEST



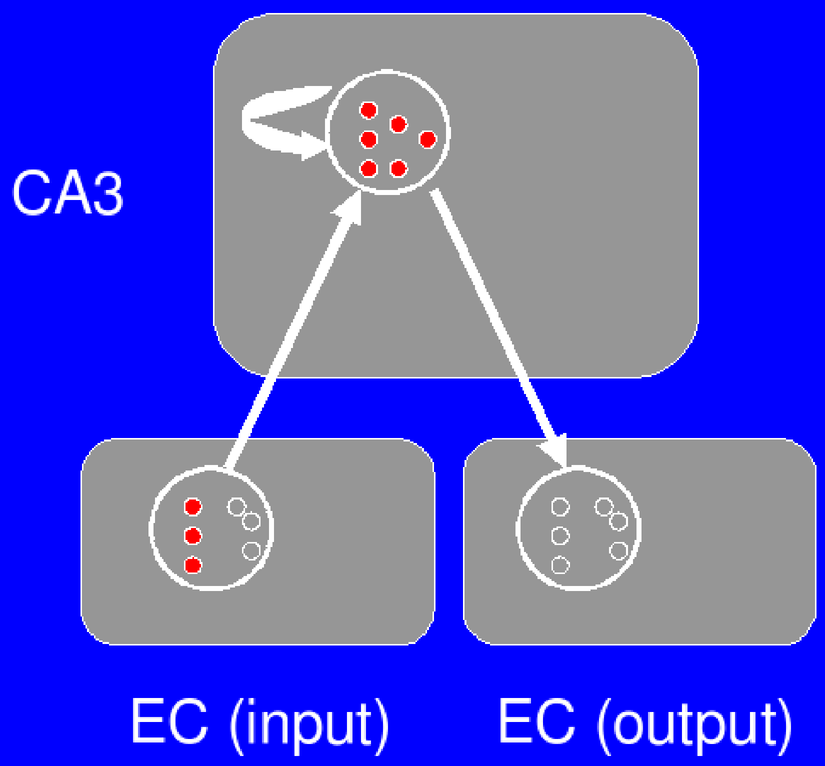
Pattern Completion

TEST



Pattern Completion

TEST



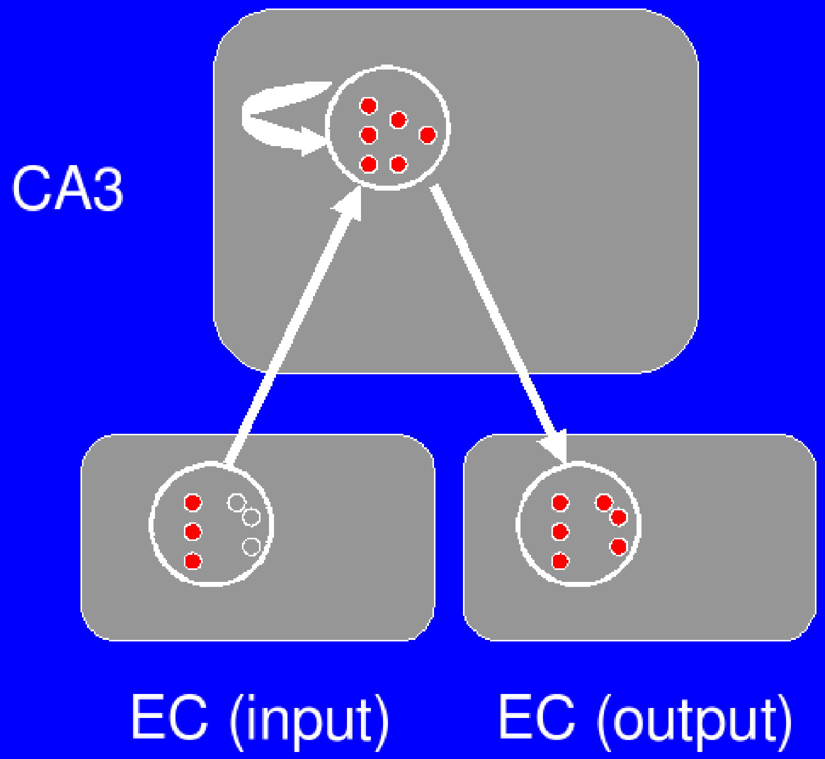
CA3

EC (input)

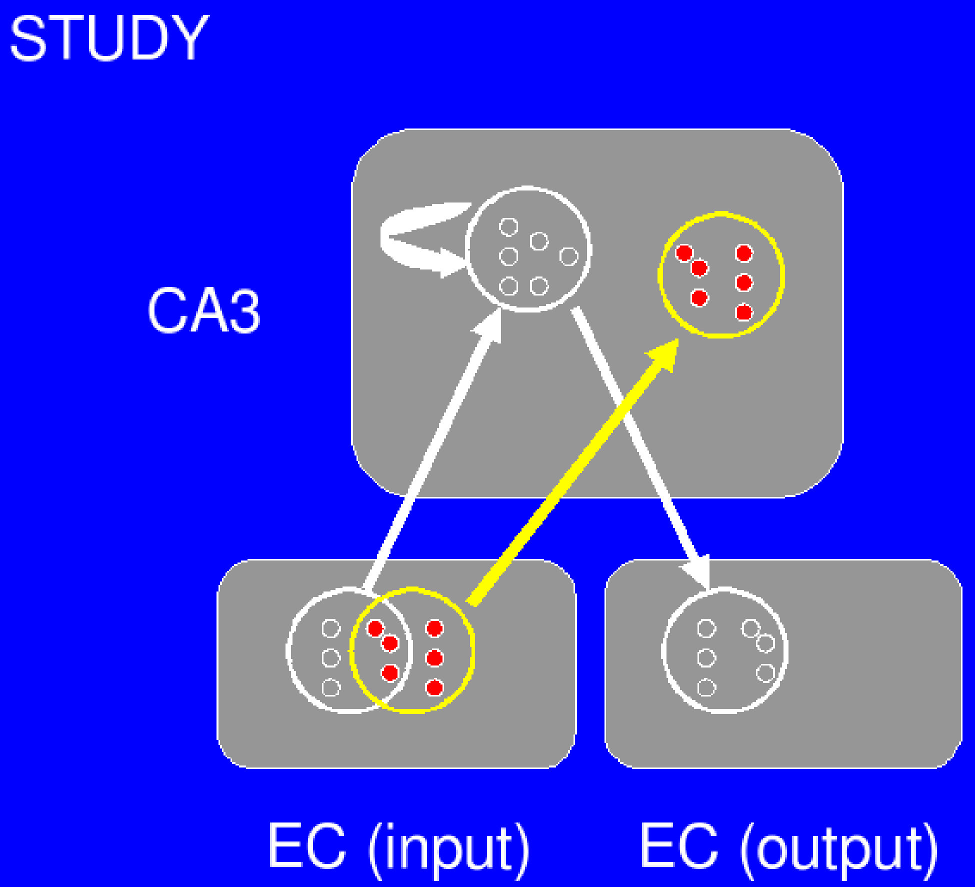
EC (output)

Pattern Completion

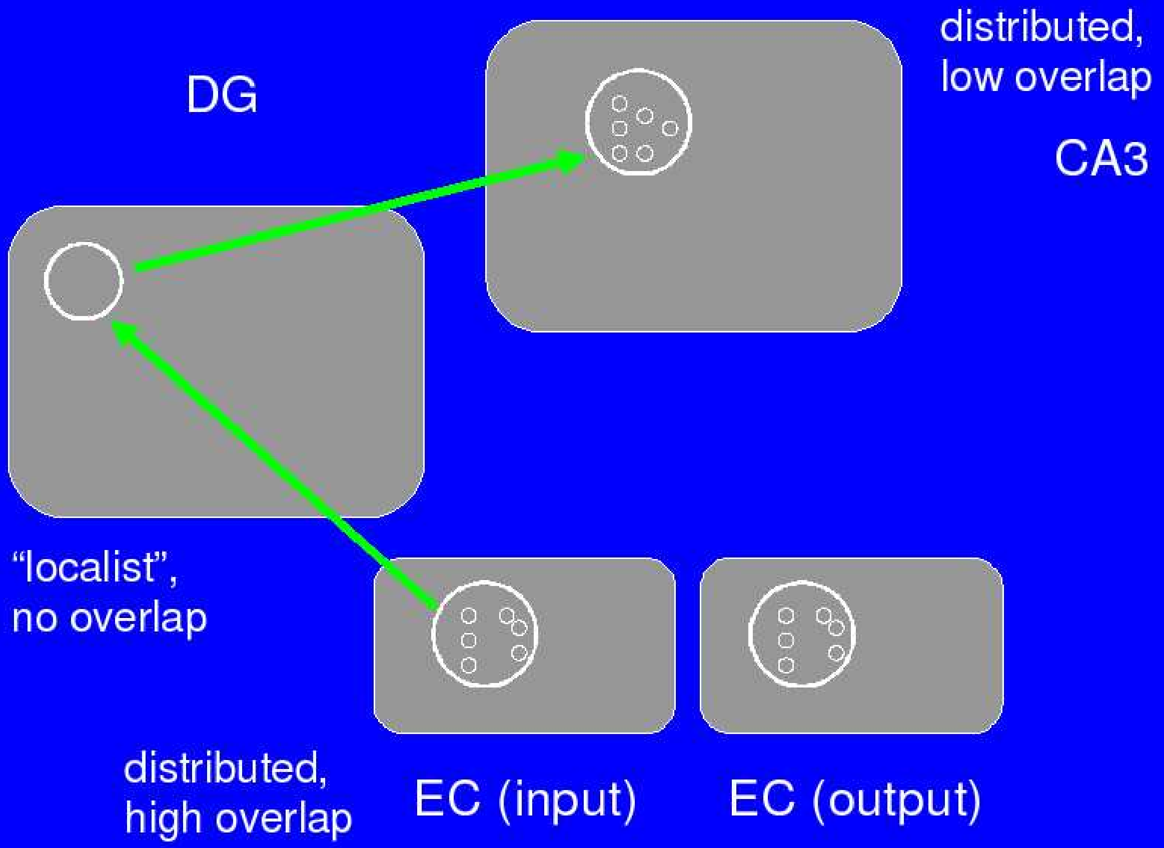
TEST



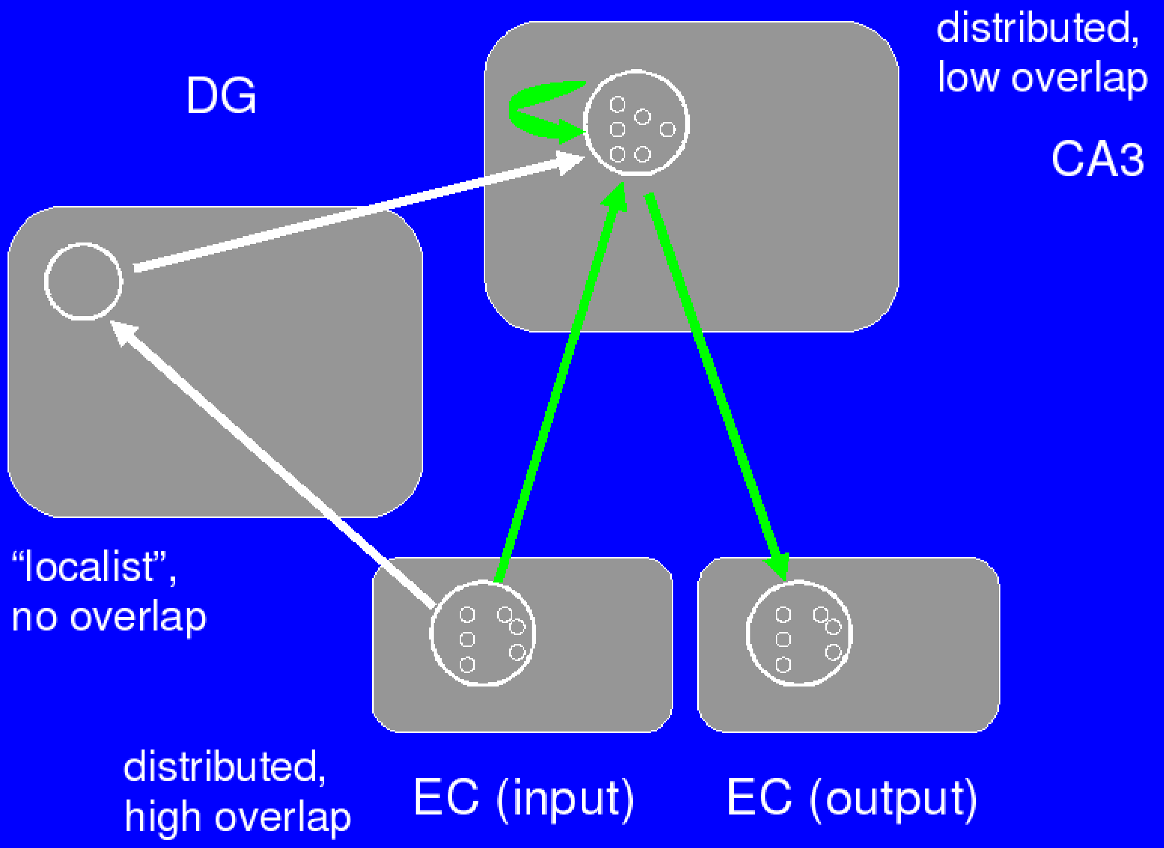
Pattern Separation



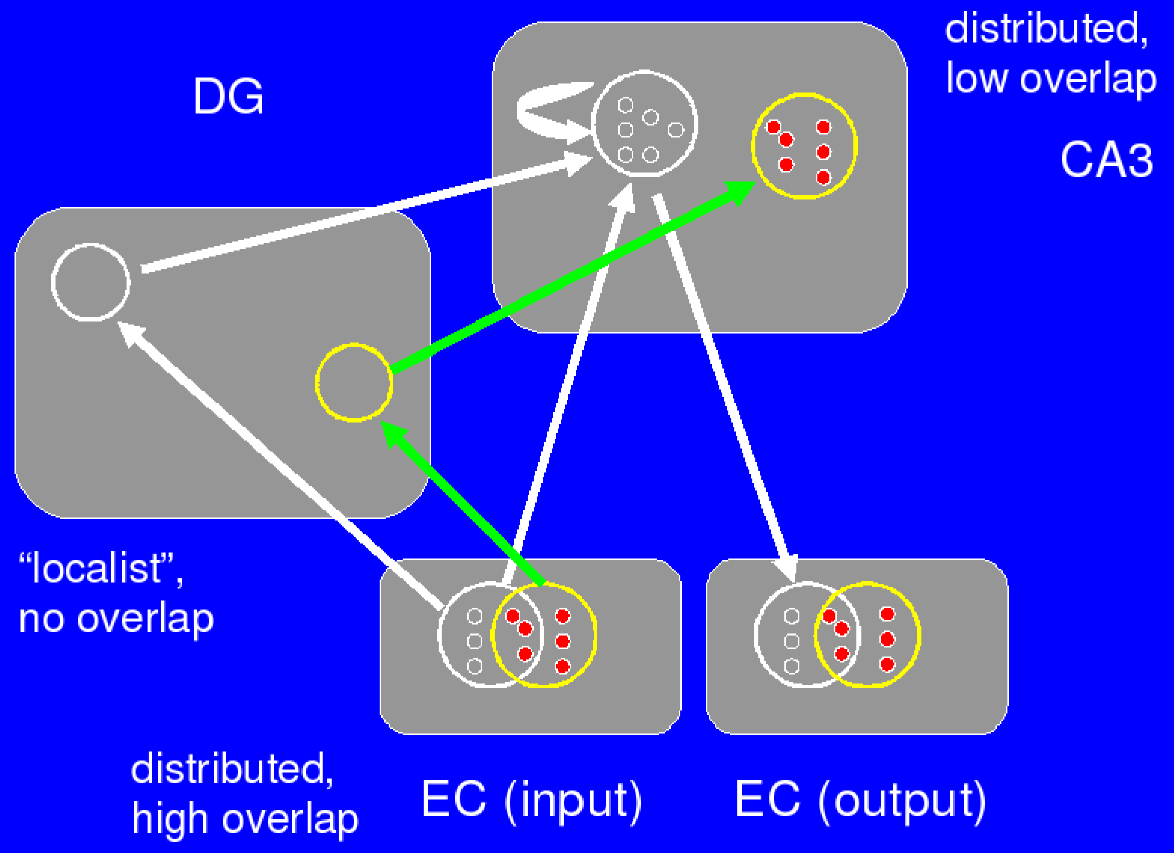
DG = Pattern Separation Turbocharger



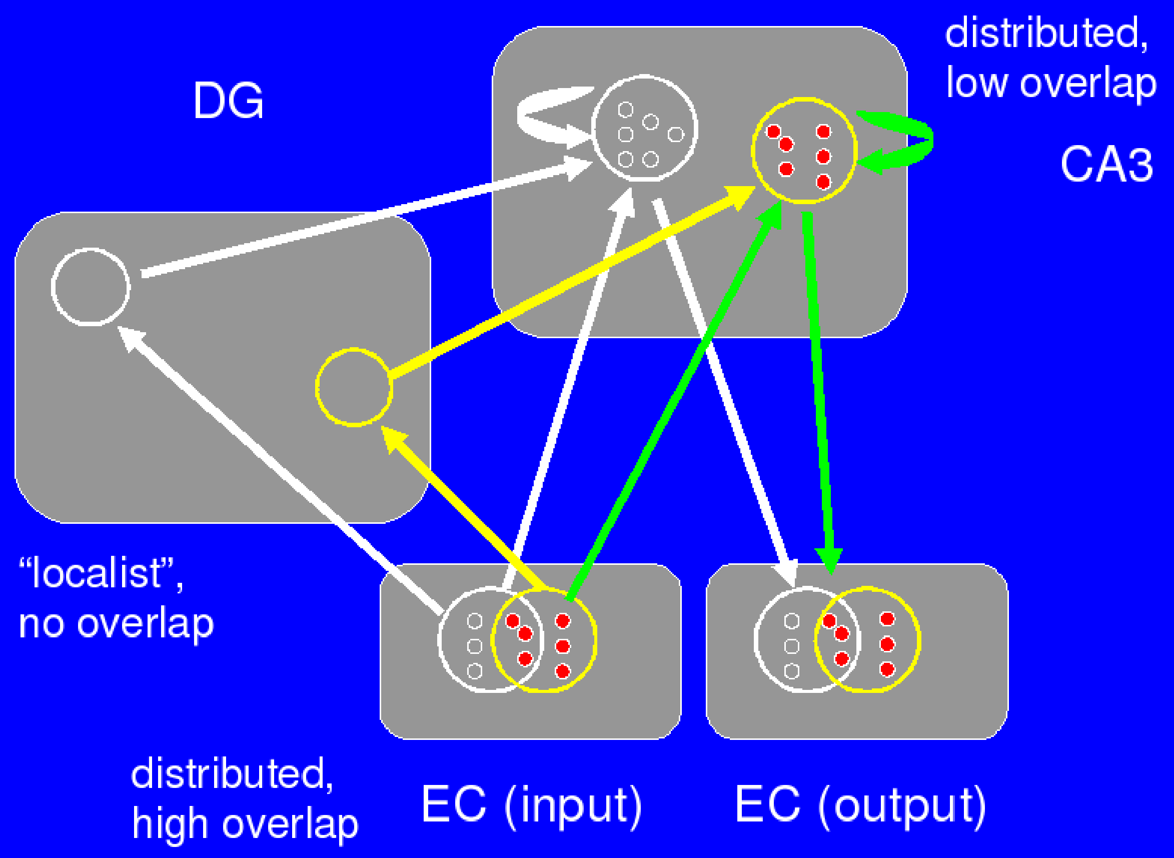
DG = Pattern Separation Turbocharger



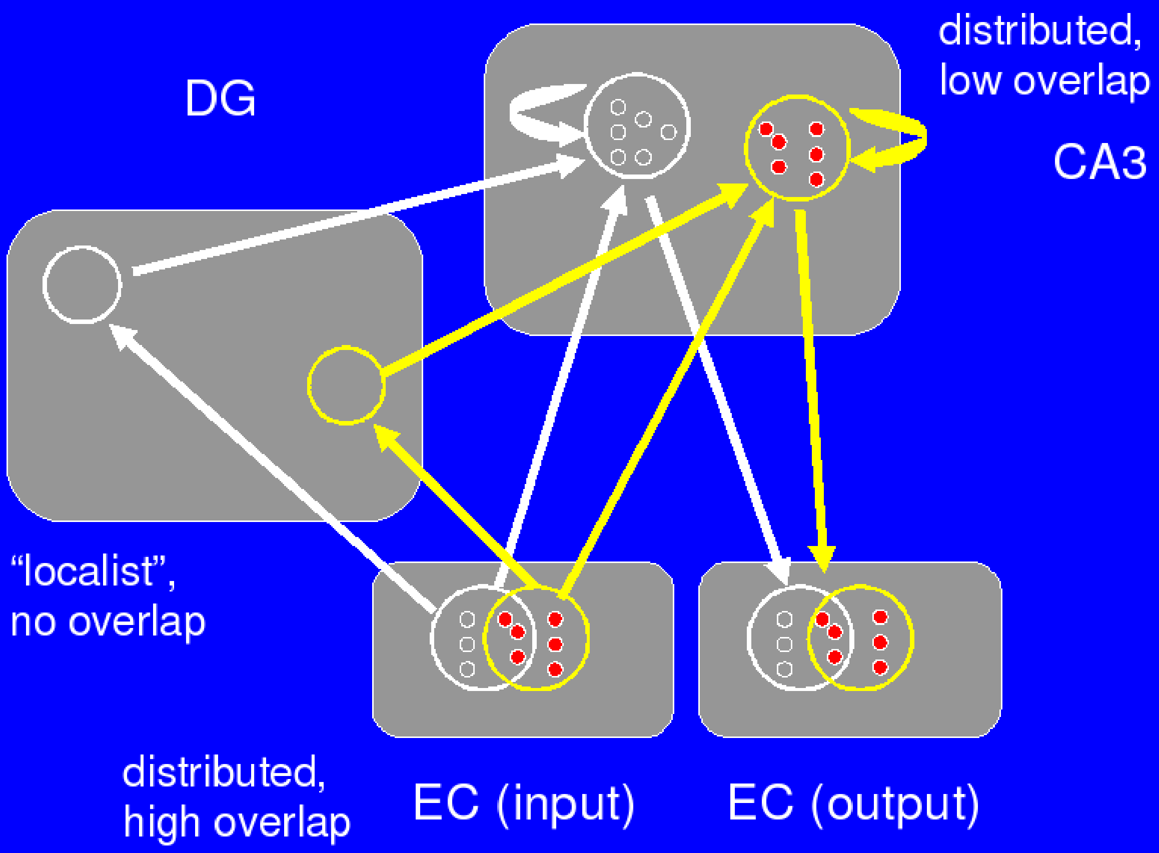
DG = Pattern Separation Turbocharger



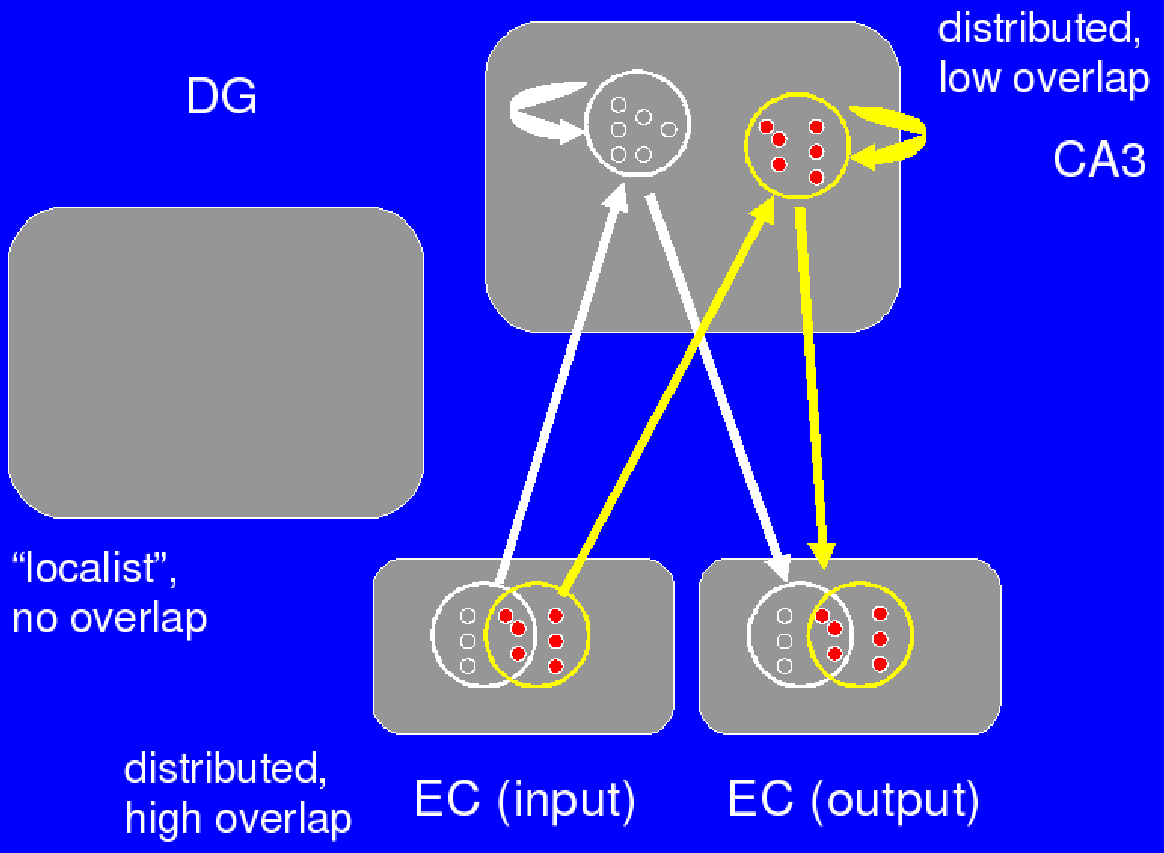
DG = Pattern Separation Turbocharger



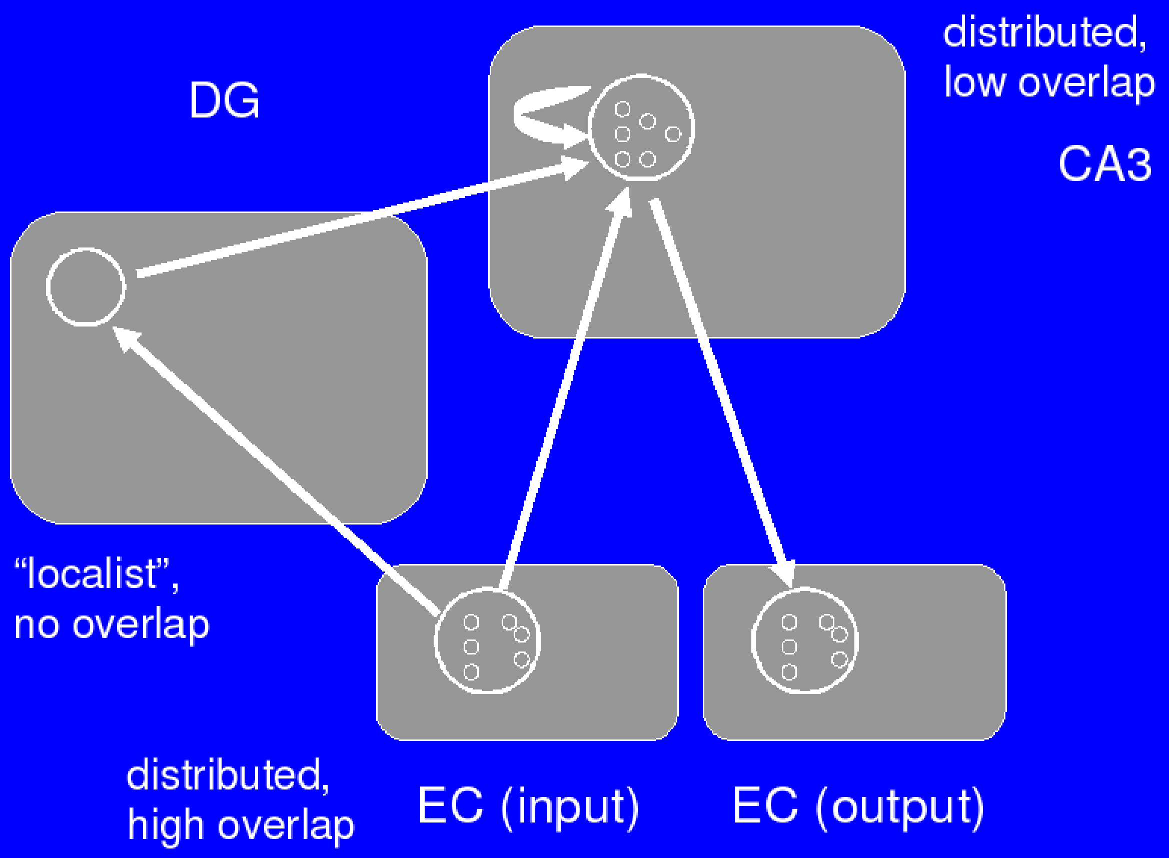
DG = Pattern Separation Turbocharger



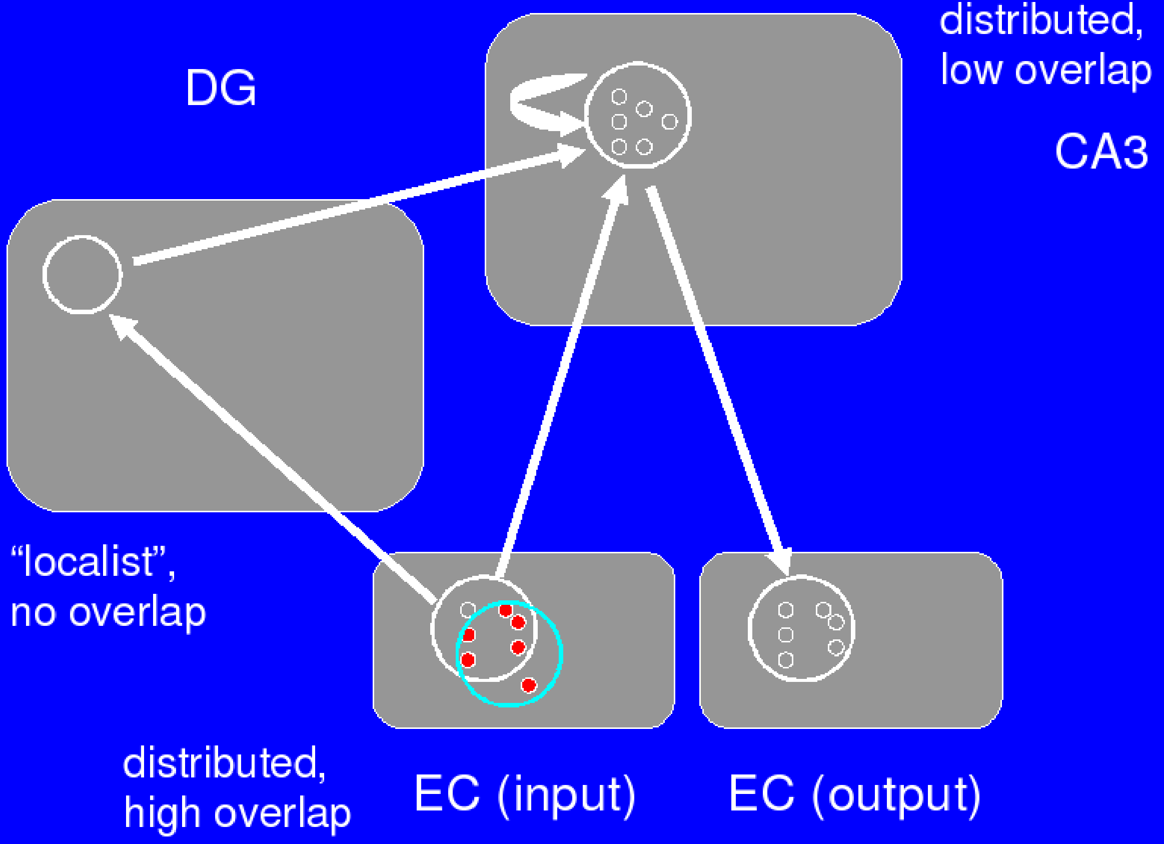
DG = Pattern Separation Turbocharger



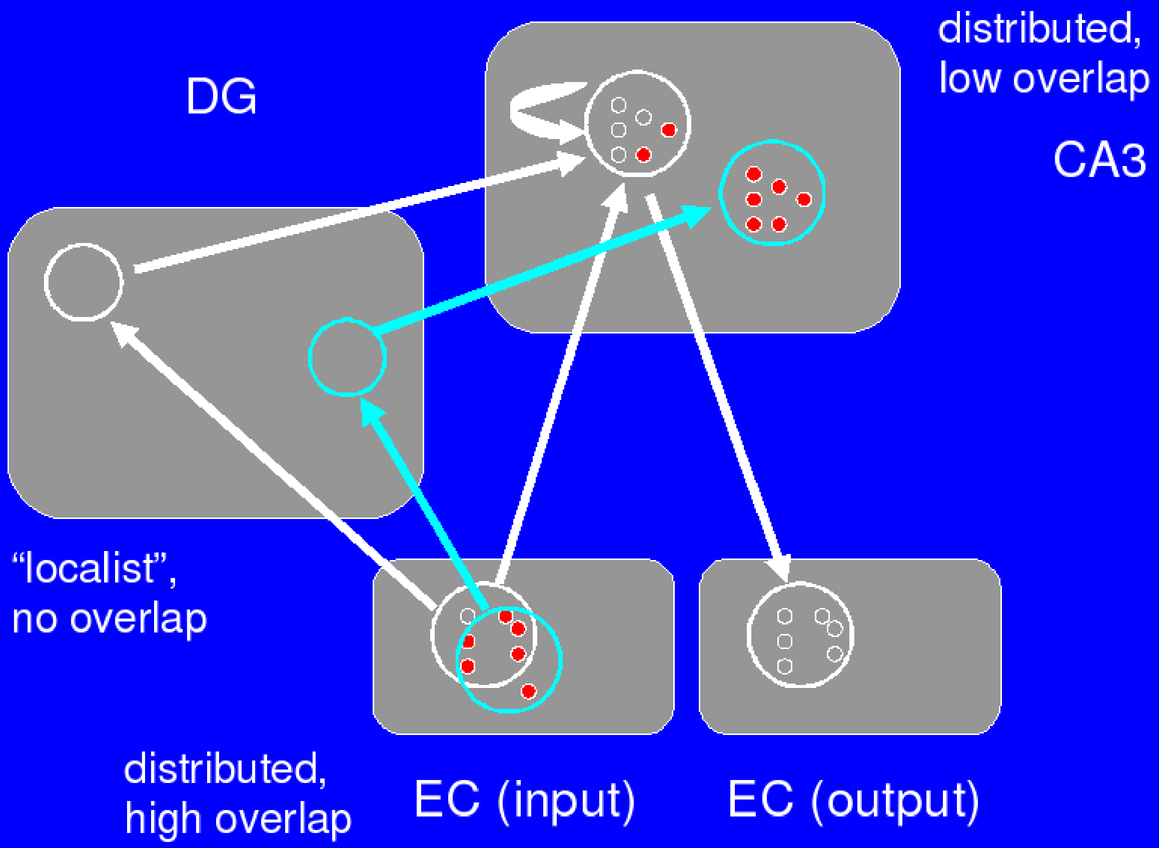
DG is good for pattern separation, but it's bad for recall



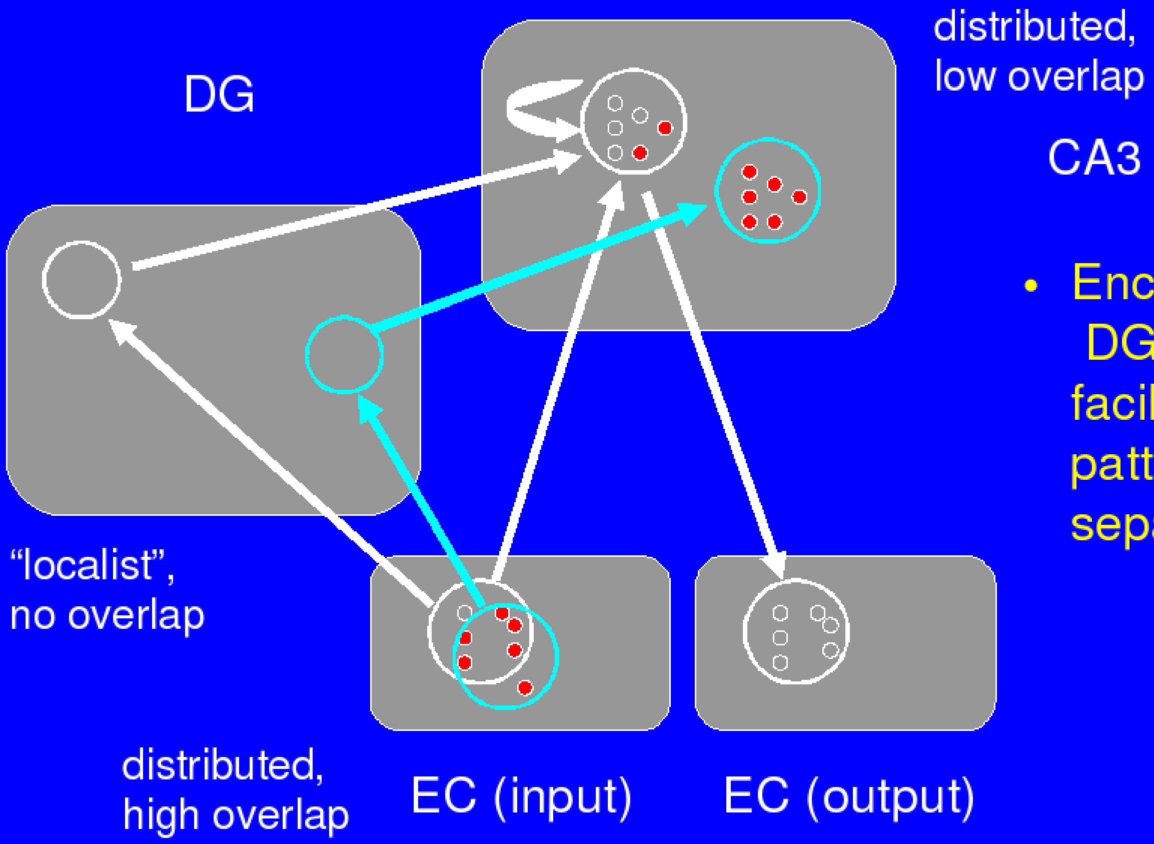
DG is good for pattern separation, but it's bad for recall



DG is good for pattern separation, but it's bad for recall

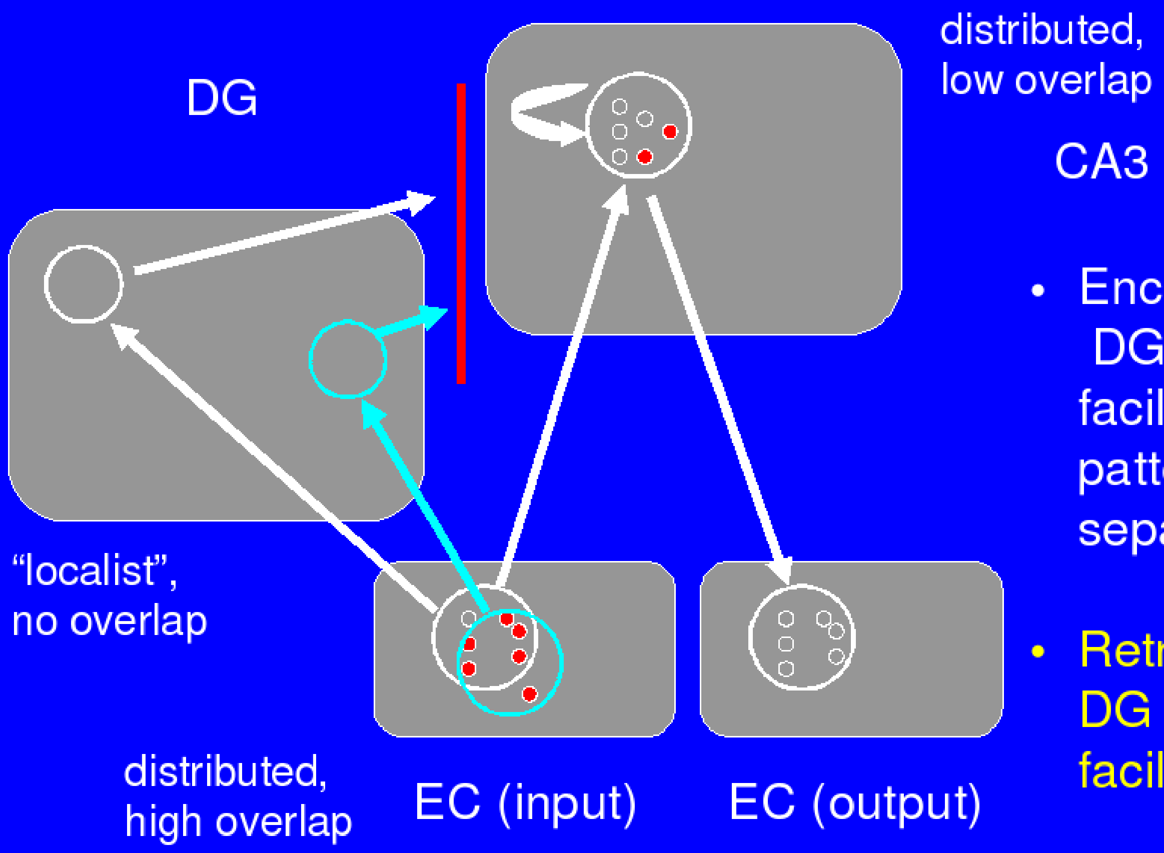


Possible Solution: Two Modes



- Encoding mode: DG strong, facilitates pattern separation

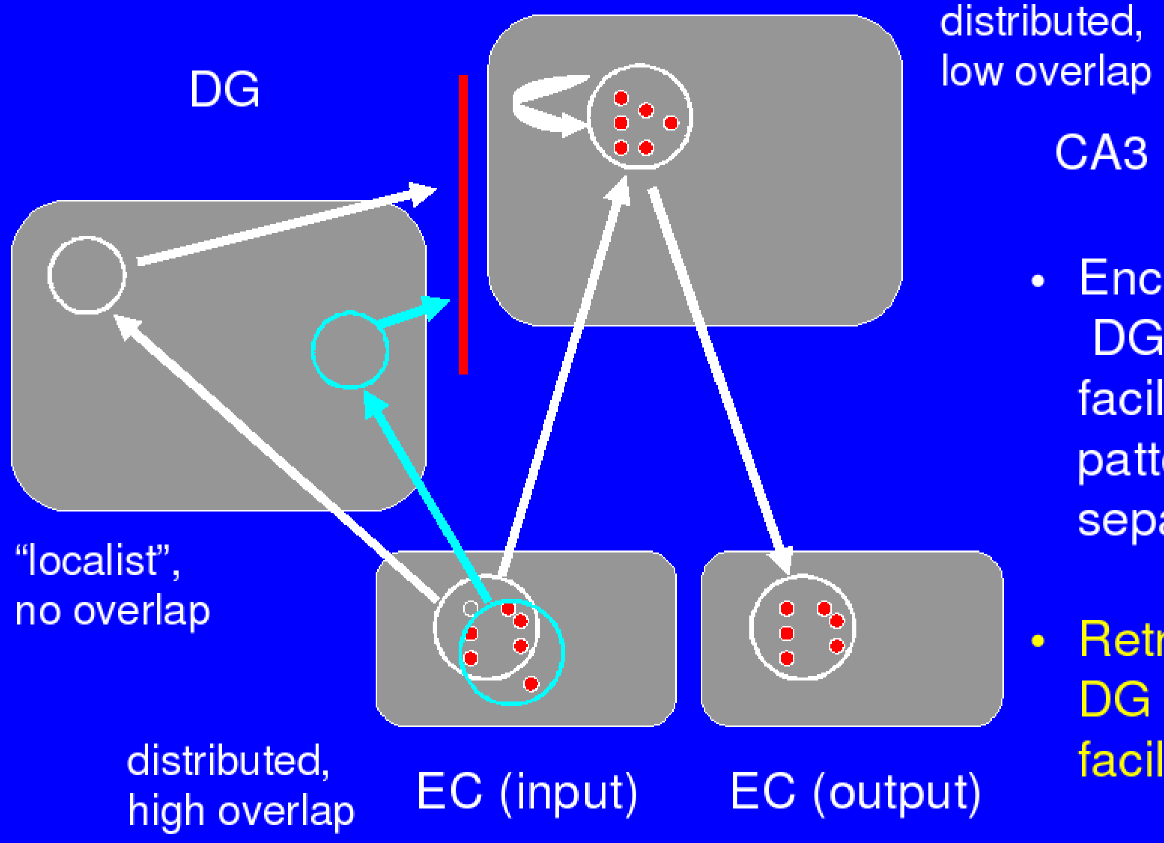
Possible Solution: Two Modes



- Encoding mode:
  DG strong,
  facilitates
  pattern
  separation

- Retrieval mode:
  DG weak,
  facilitates recall

Possible Solution: Two Modes

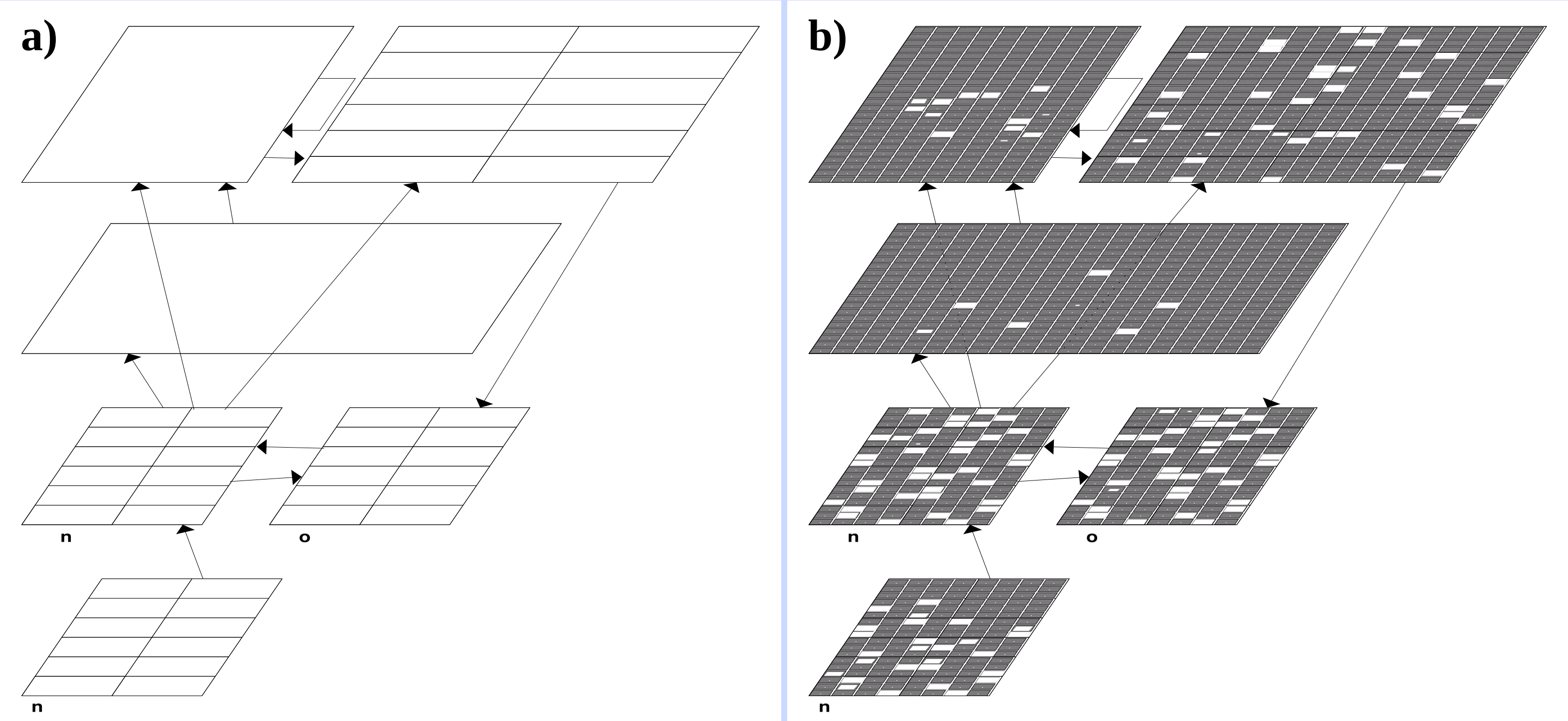


- Encoding mode: DG strong, facilitates pattern separation
- Retrieval mode: DG weak, facilitates recall

Hippocampus: Summary

- CA3 stores sparse, pattern-separated representations of cortical input patterns
- Recurrent self projections in CA3 facilitate recall (pattern completion)
- Dentate Gyrus (DG) acts as a *removable pattern separation turbocharger*
  - DG uses super-sparse representations, helps increase pat separation at encoding
  - DG “steps aside” at retrieval
  - Evidence for two modes: theta cycle (eg. Hasselmo et al, 2002); neuromodulatory control over rel DG effect on CA3

The Model



An example of how Modeling informs Science

Model makes clear predictions about how different regions contribute to memory (not directly evident in experiments before)

Many of these have been subsequently confirmed!

An example of how Modeling informs Science

Model makes clear predictions about how different regions contribute to memory (not directly evident in experiments before)

Many of these have been subsequently confirmed!

(note that model itself is incremental synthesis of many ideas in a coherent framework, ranging from Hebb to Marr to Nadel, McNaughton, O'Reilly...)

It has been applied to explain many different learning and memory phenomena in rats and humans.

An example of how Modeling informs Science

Pattern separation in Rat DG (*Leutgeb et al, 2007,* Science)

(change environment ever so slightly see new populations of correlated act)

An example of how Modeling informs Science

Pattern separation in Rat DG (*Leutgeb et al, 2007,* Science)

(change environment ever so slightly see new populations of correlated act)

Pattern separation in Human DG (*Bakker et al, 2008,* Science)

(encode new stims, some are similar to old but slightly diff).

An example of how Modeling informs Science

Pattern separation in Rat DG (*Leutgeb et al, 2007,* Science)
(change environment ever so slightly see new populations of correlated act)

Pattern separation in Human DG (*Bakker et al, 2008,* Science)
(encode new stims, some are similar to old but slightly diff).

Mouse genetic knockout of DG NMDA receptors impairs pat separation behaviorally; also CA3 becomes more biased toward completion than separation (*McHugh et al, 2007,* Science)

An example of how Modeling informs Science

Pattern separation in Rat DG (*Leutgeb et al, 2007,* Science)
(change environment ever so slightly see new populations of correlated act)

Pattern separation in Human DG (*Bakker et al, 2008,* Science)
(encode new stims, some are similar to old but slightly diff).

Mouse genetic knockout of DG NMDA receptors impairs pat separation behaviorally; also CA3 becomes more biased toward completion than separation (*McHugh et al, 2007,* Science)

Monosynaptic route (EC→CA1→EC) sufficient on its own for incremental spatial learning, *but*
Trisynaptic route (EC→DG→CA3→CA1→EC) required for rapid, one-trial conjunctive learning, and for pat completion.
(Transgenic mouse *Nakashiba et al, 2008,* Science)

An example of how Modeling informs Science

Pattern separation in Rat DG (*Leutgeb et al, 2007,* Science)
(change environment ever so slightly see new populations of correlated act)

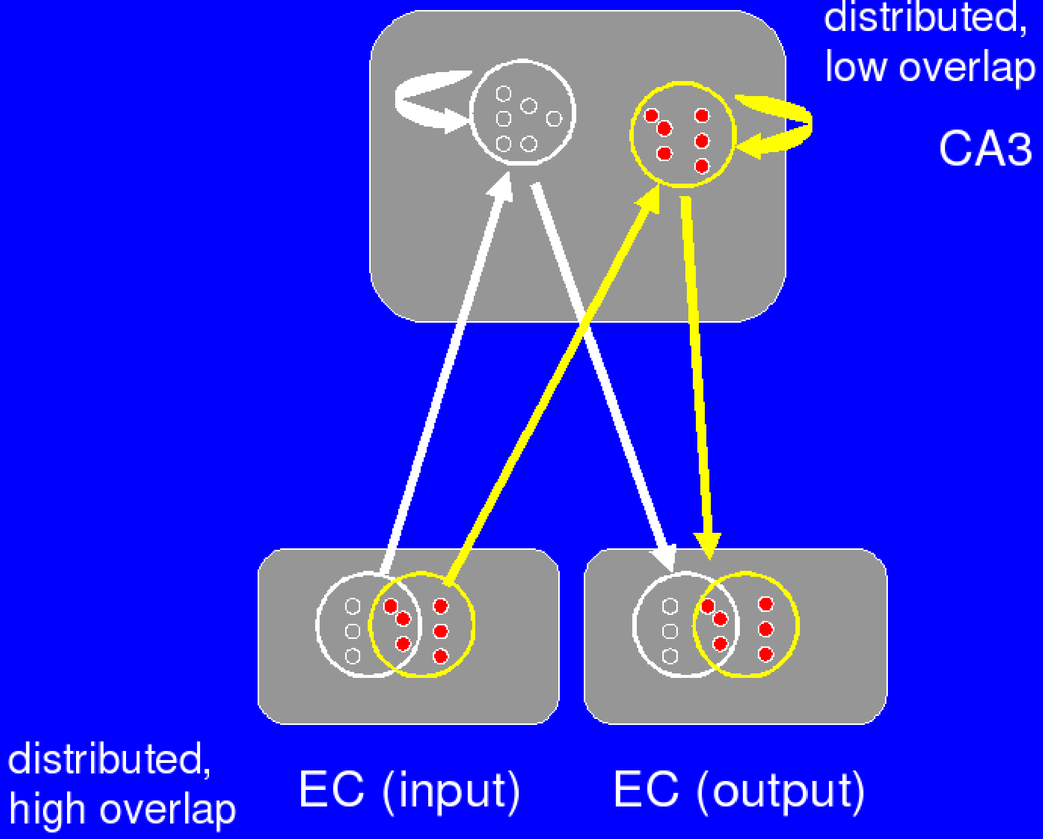
Pattern separation in Human DG (*Bakker et al, 2008,* Science)
(encode new stims, some are similar to old but slightly diff).

Mouse genetic knockout of DG NMDA receptors impairs pat separation behaviorally; also CA3 becomes more biased toward completion than separation (*McHugh et al, 2007,* Science)

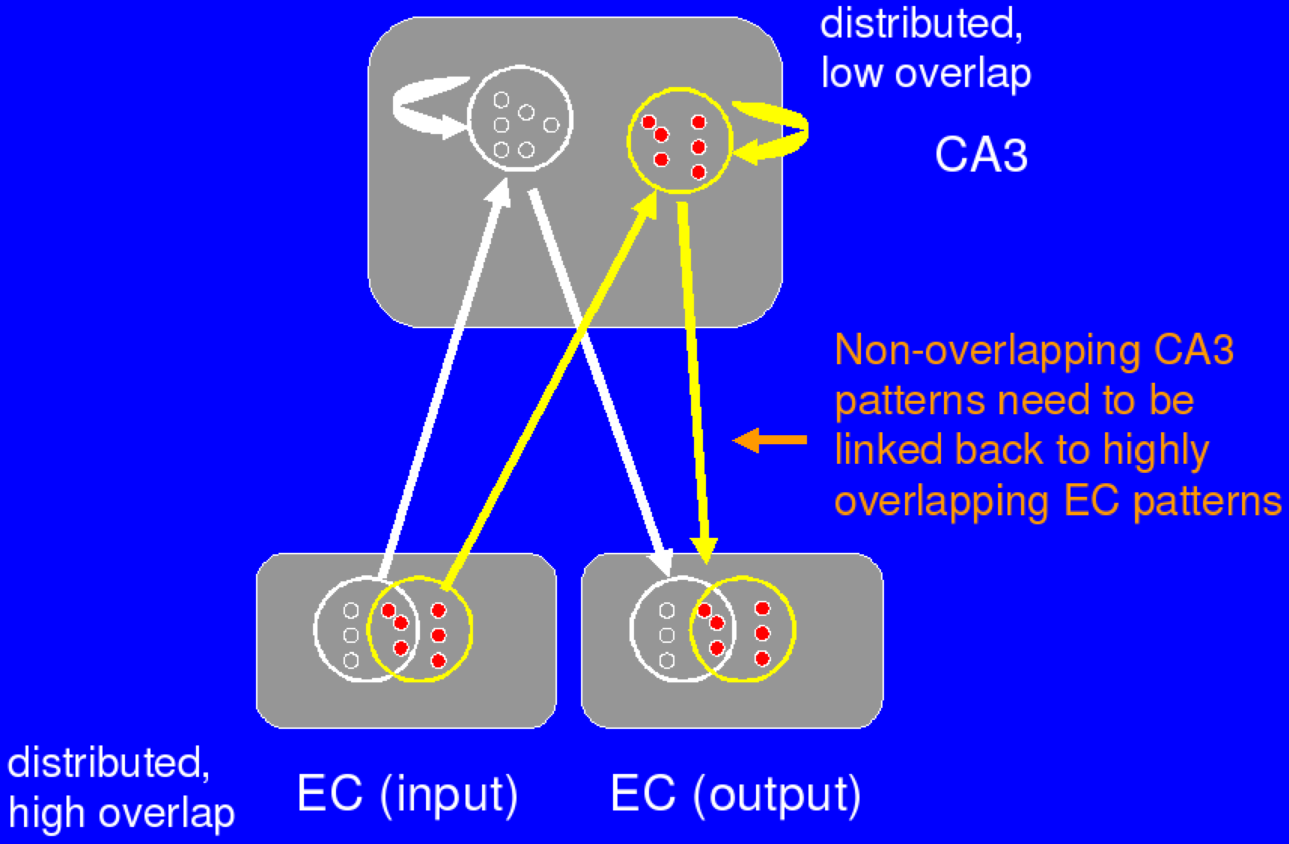
Monosynaptic route (EC→CA1→EC) sufficient on its own for incremental spatial learning, *but*
Trisynaptic route (EC→DG→CA3→CA1→EC) required for rapid, one-trial conjunctive learning, and for pat completion.
(Transgenic mouse *Nakashiba et al, 2008,* Science)

Neurogenesis in DG supports behavioral pat sep *Clellan et al '09,* Science*; Nahay et al '11* Nature (discriminate between items with overlapping contexts)

Role of CA1

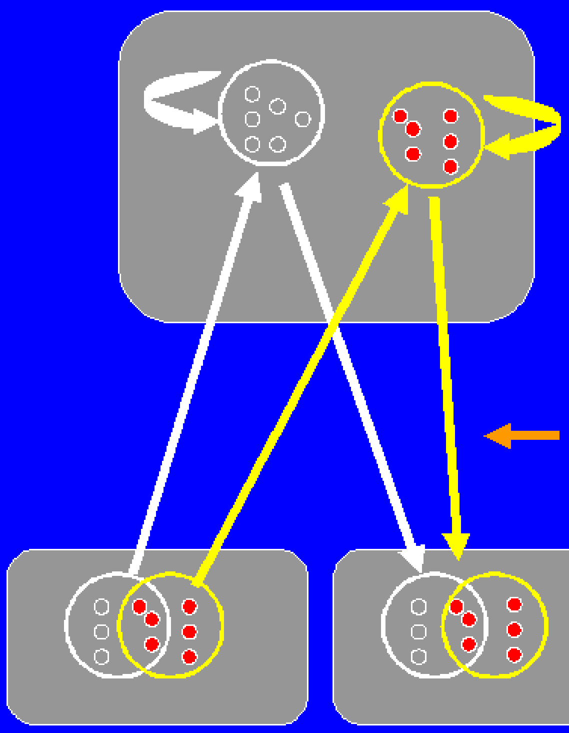

distributed,
low overlap
CA3
distributed,
high overlap
EC (input)
EC (output)

Role of CA1



Non-overlapping CA3 patterns need to be linked back to highly overlapping EC patterns

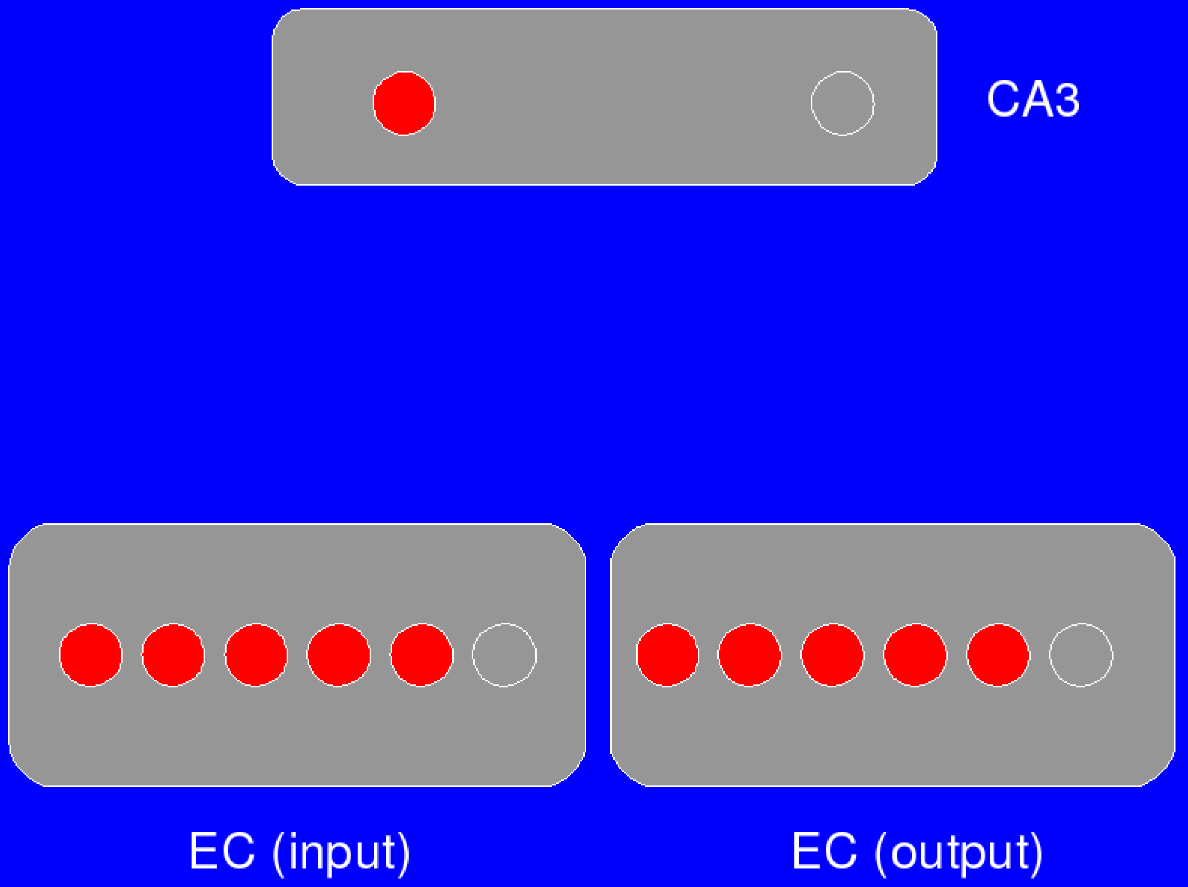
Role of CA1



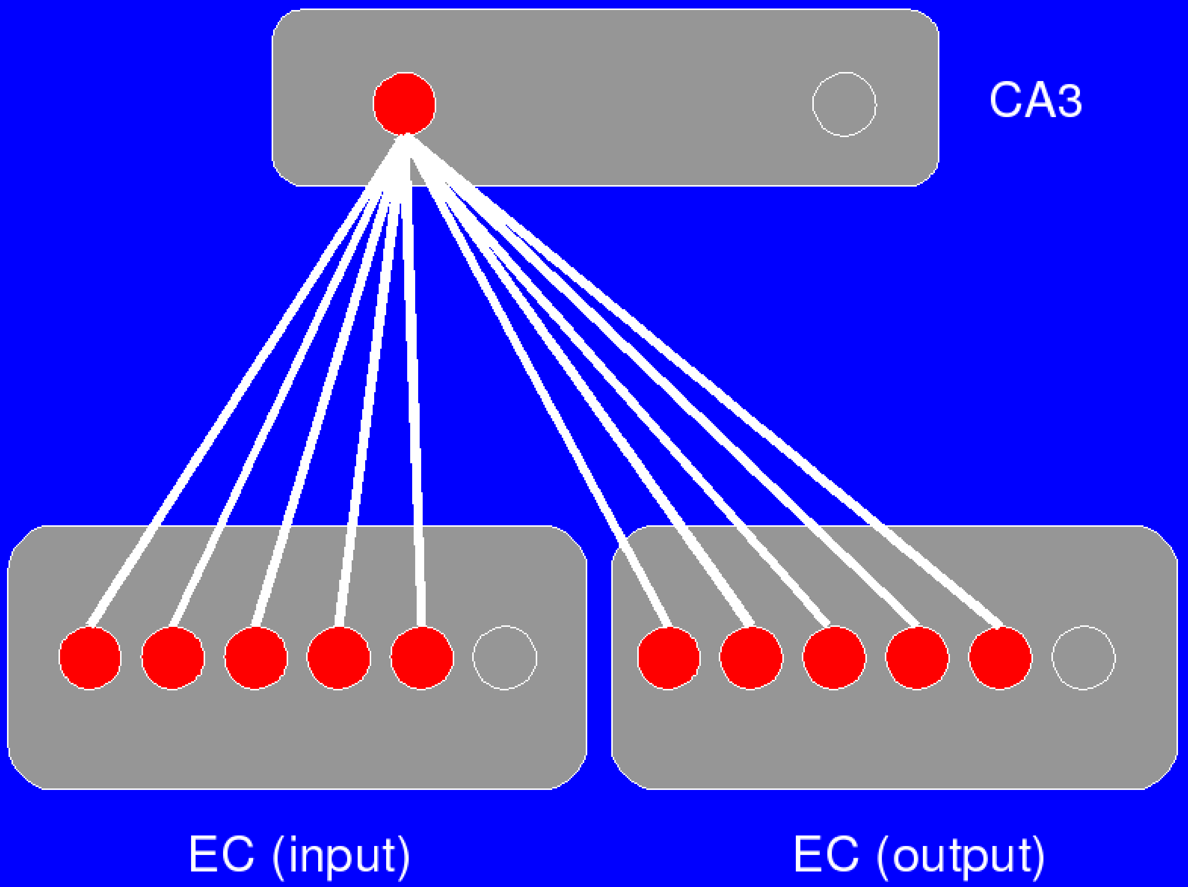
Non-overlapping CA3 patterns need to be linked back to highly overlapping EC patterns

This mapping is highly prone to interference!

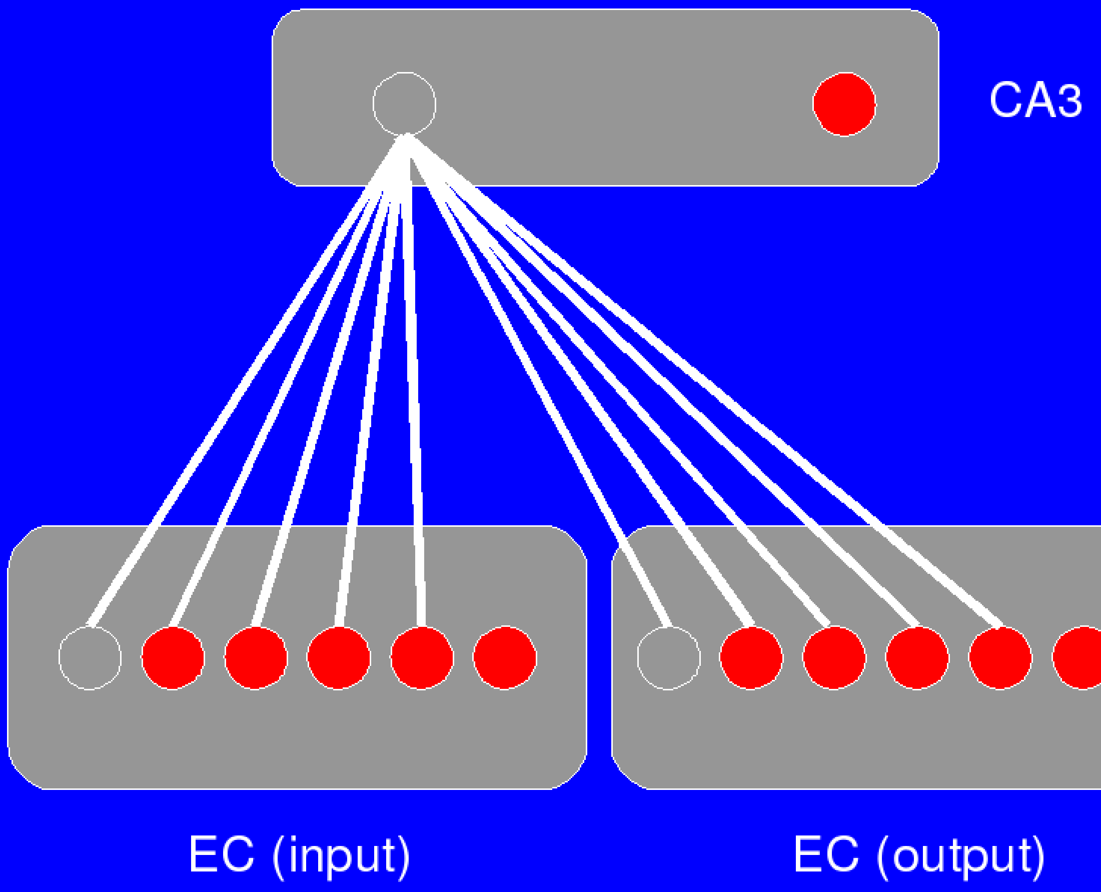
Role of CA1



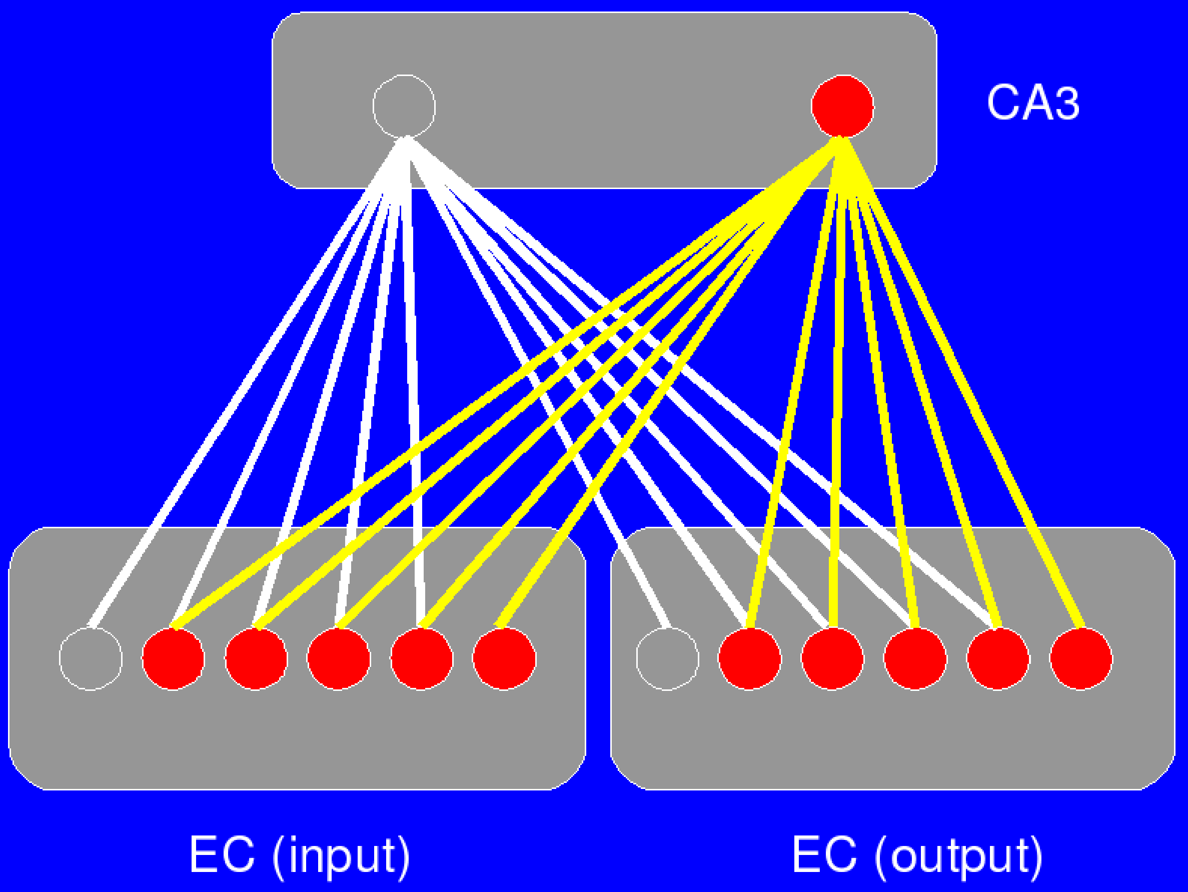
Role of CA1



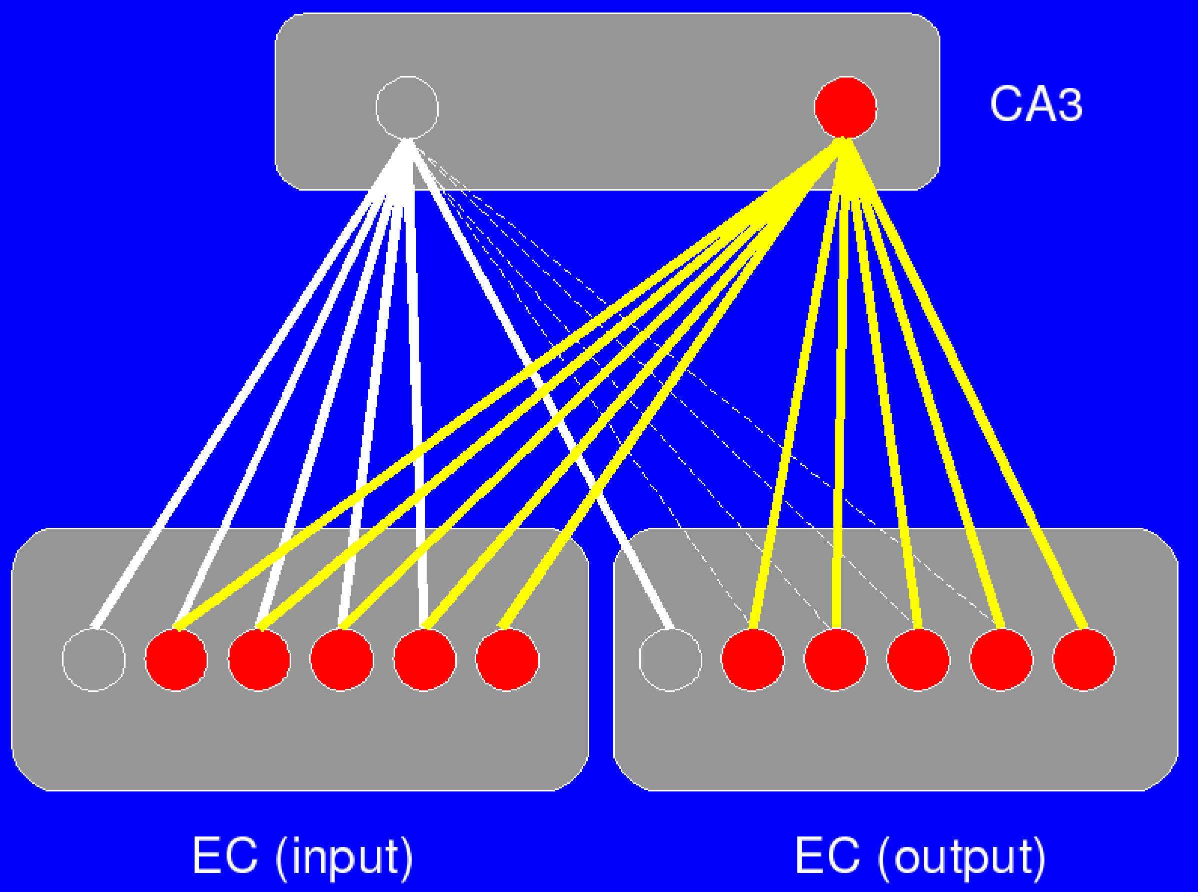
Role of CA1



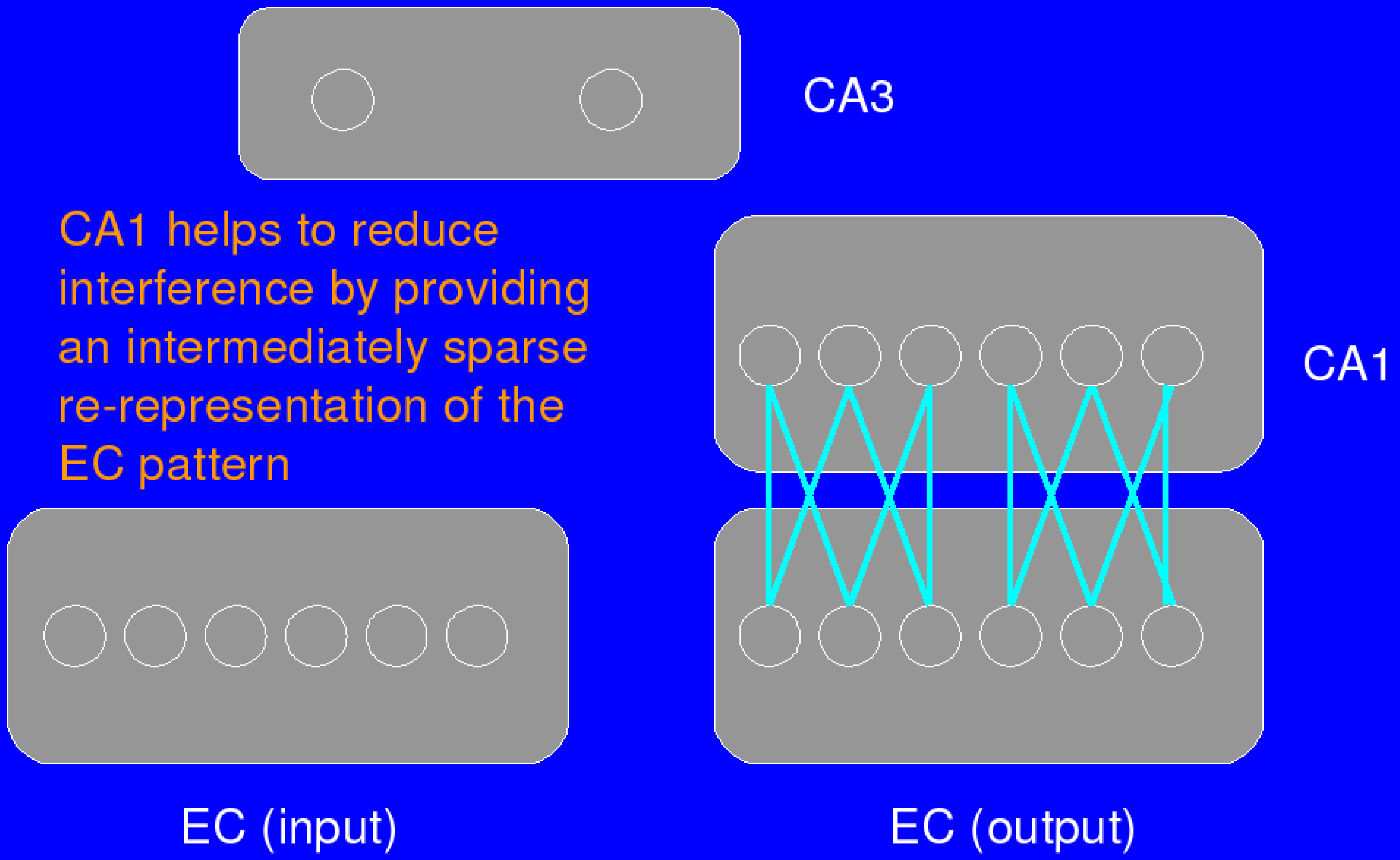
Role of CA1



Role of CA1

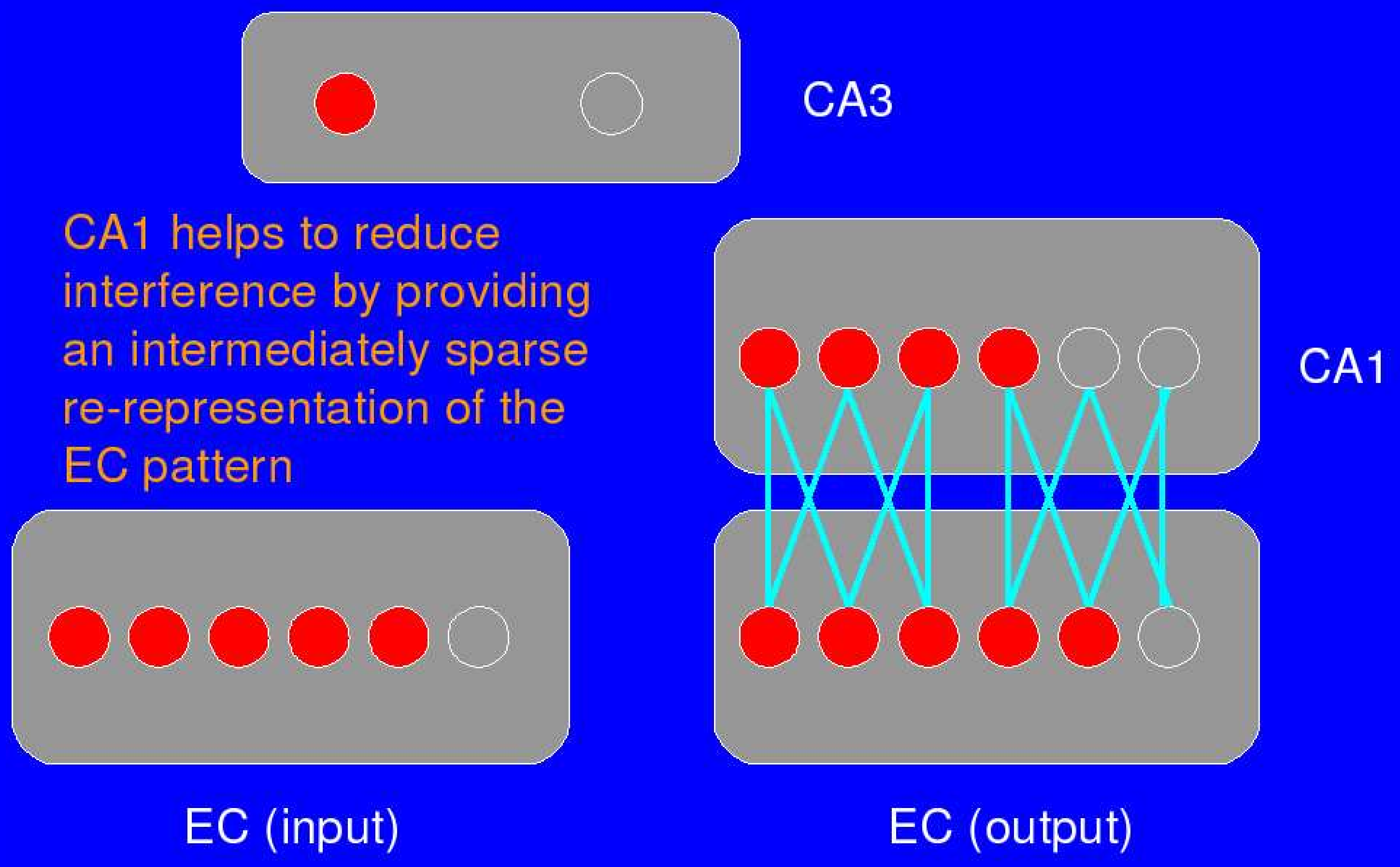


Role of CA1



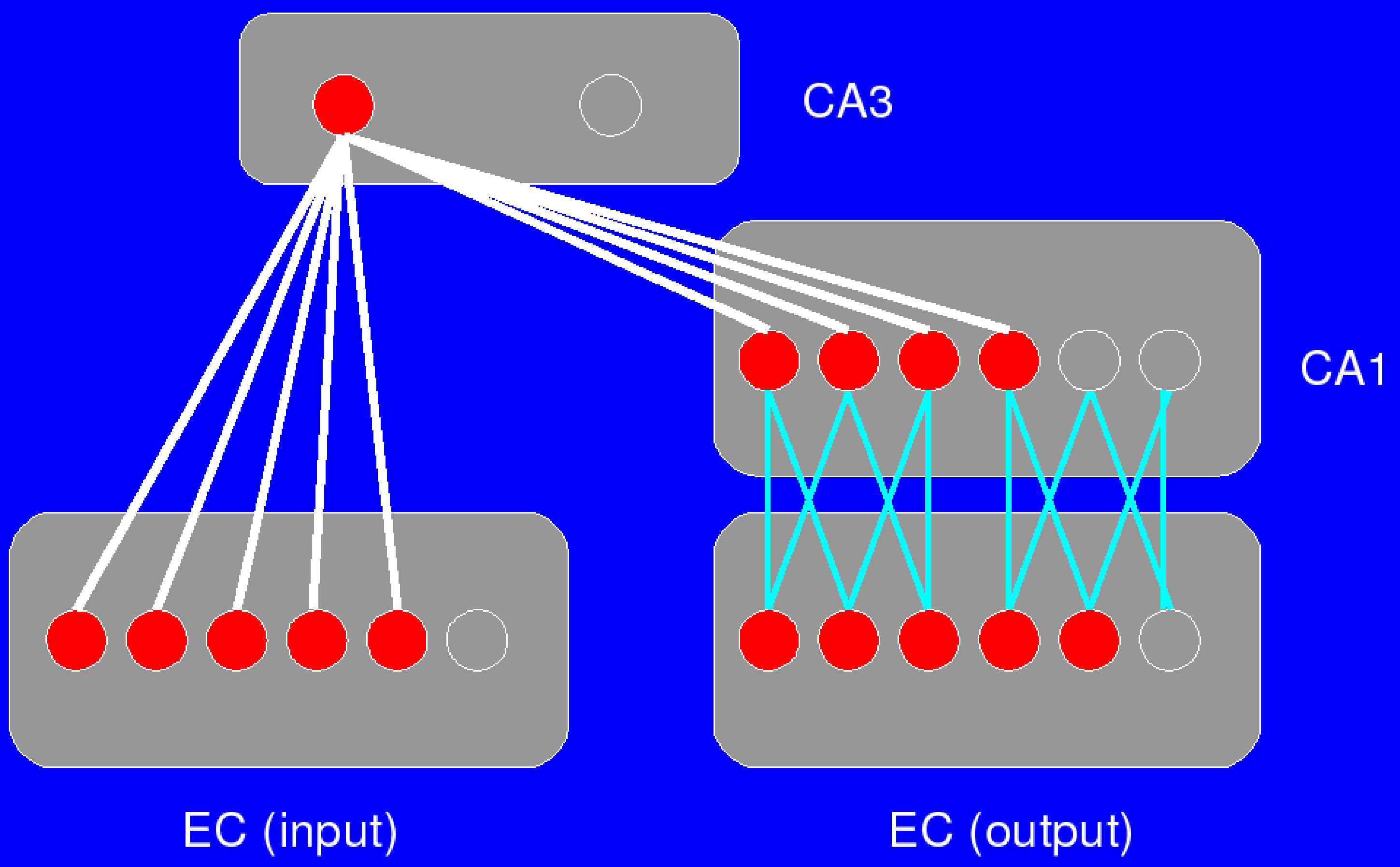
CA1 helps to reduce interference by providing an intermediately sparse re-representation of the EC pattern

Role of CA1

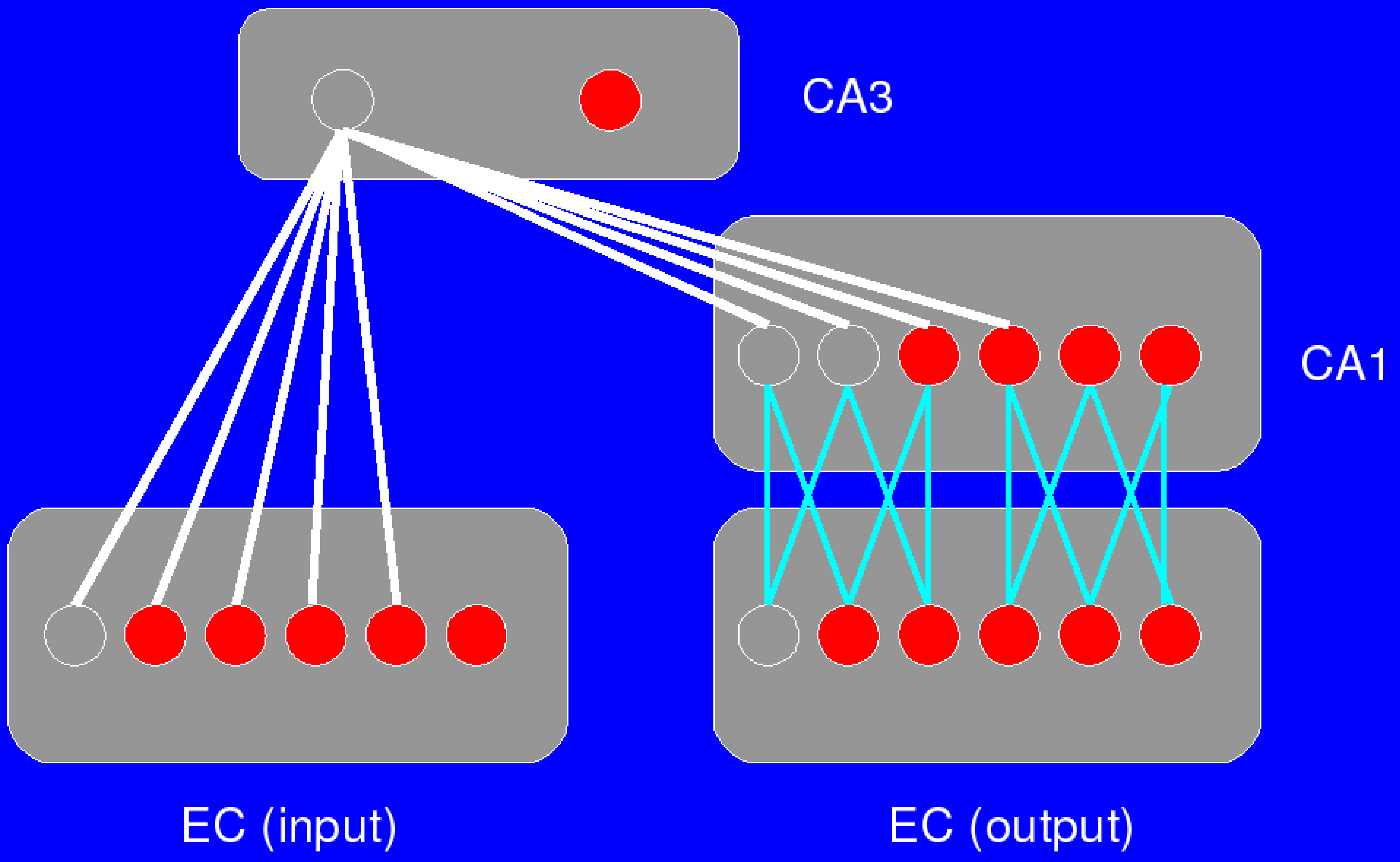


CA1 helps to reduce interference by providing an intermediately sparse re-representation of the EC pattern

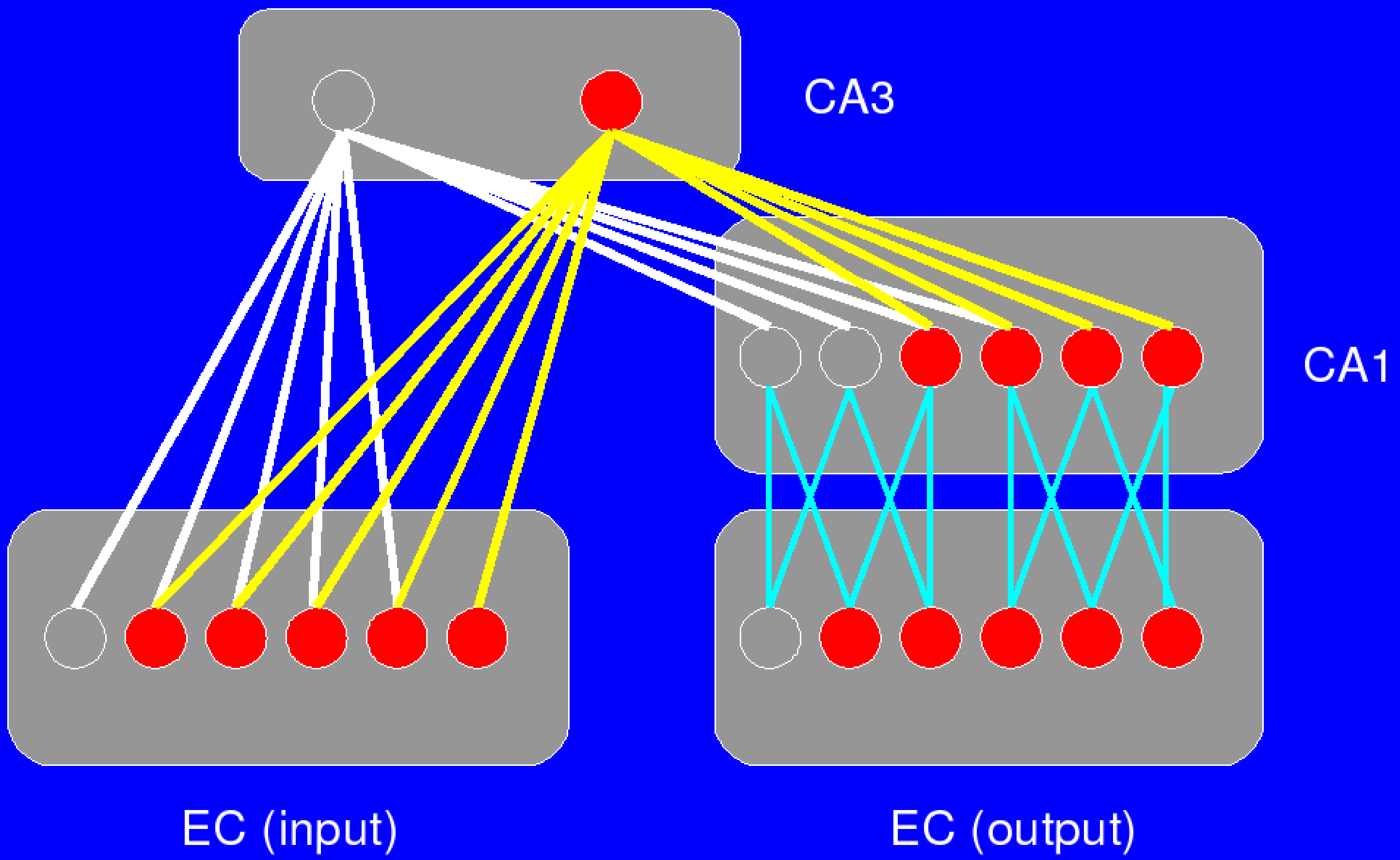
Role of CA1



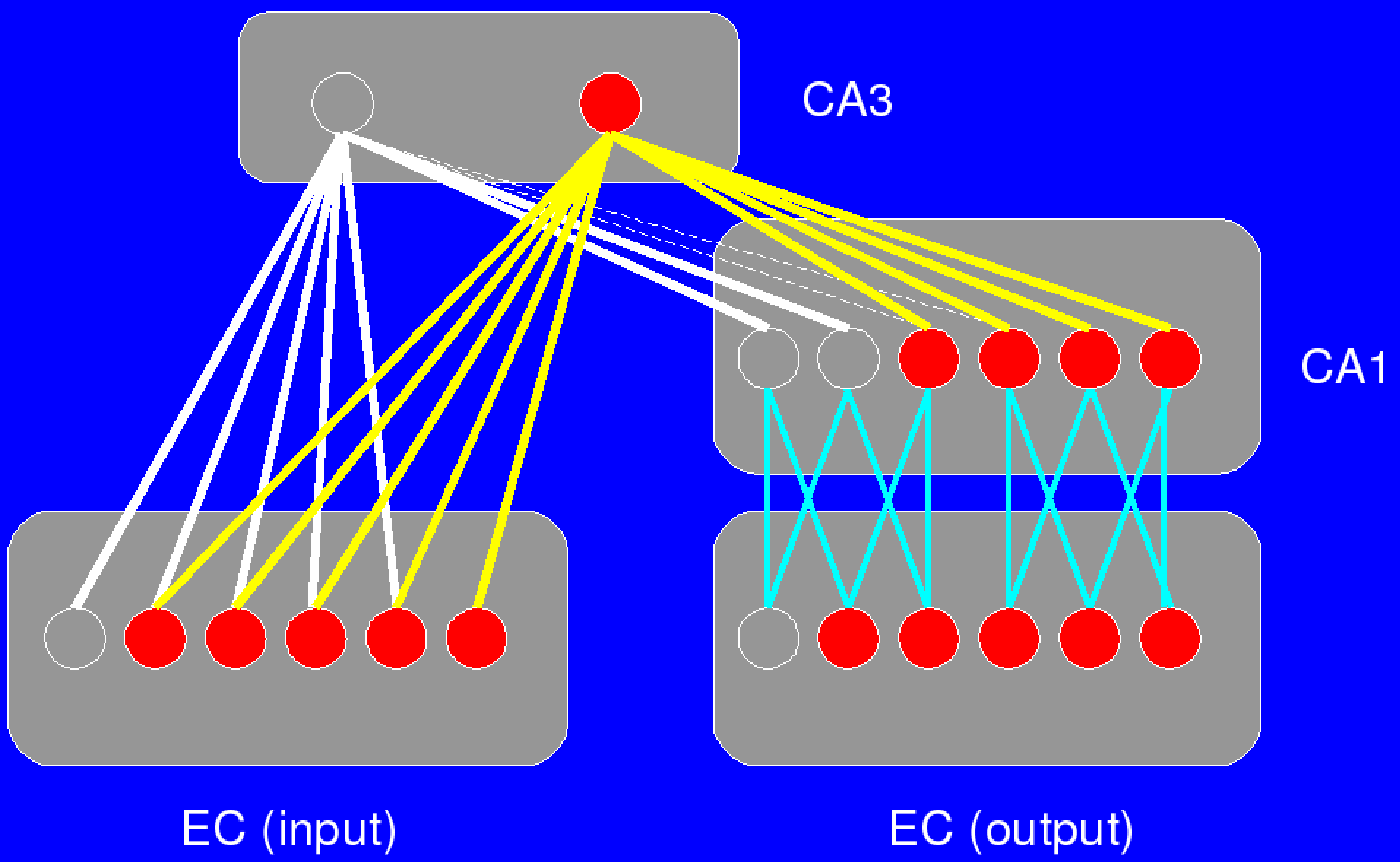
Role of CA1



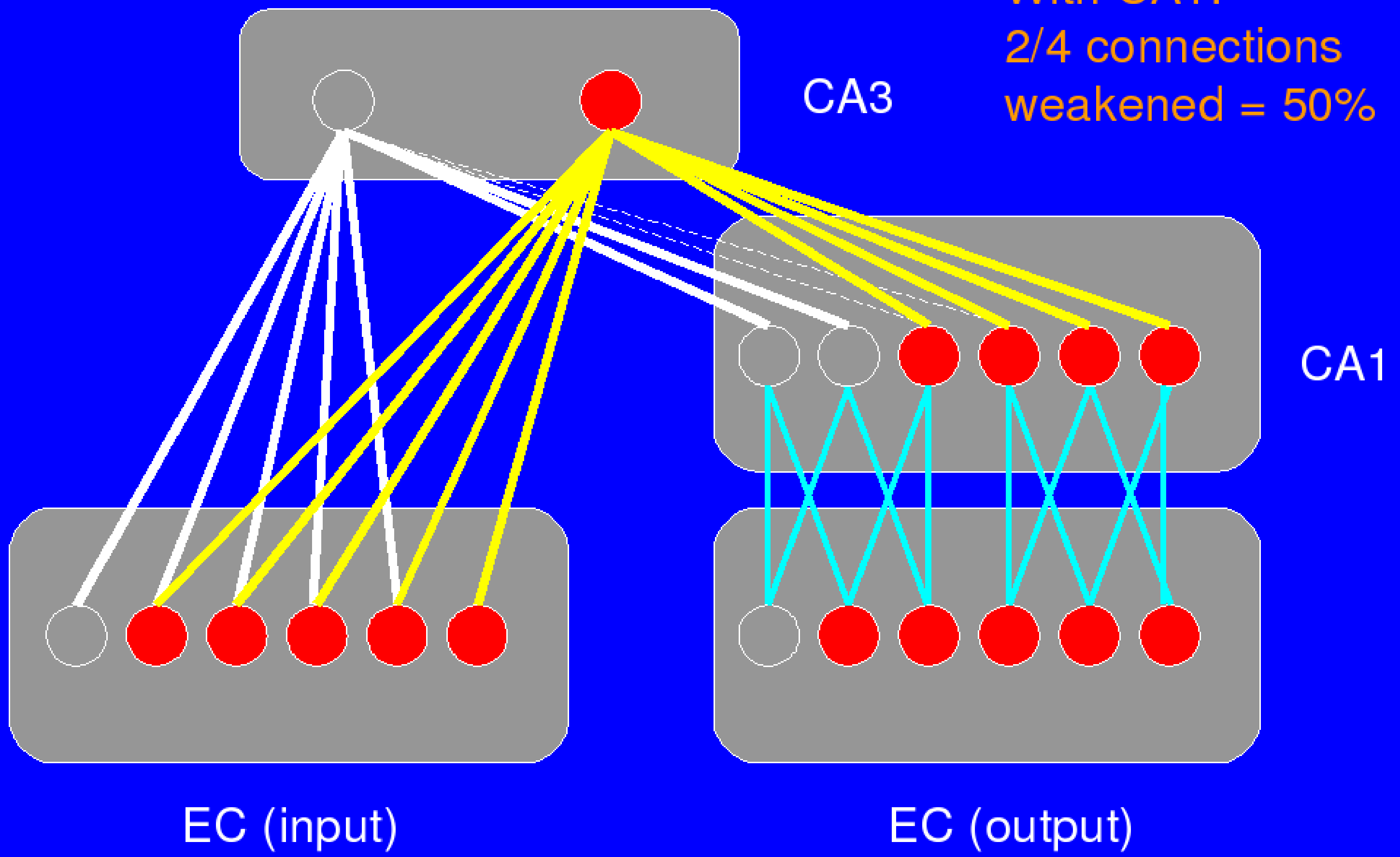
Role of CA1



Role of CA1

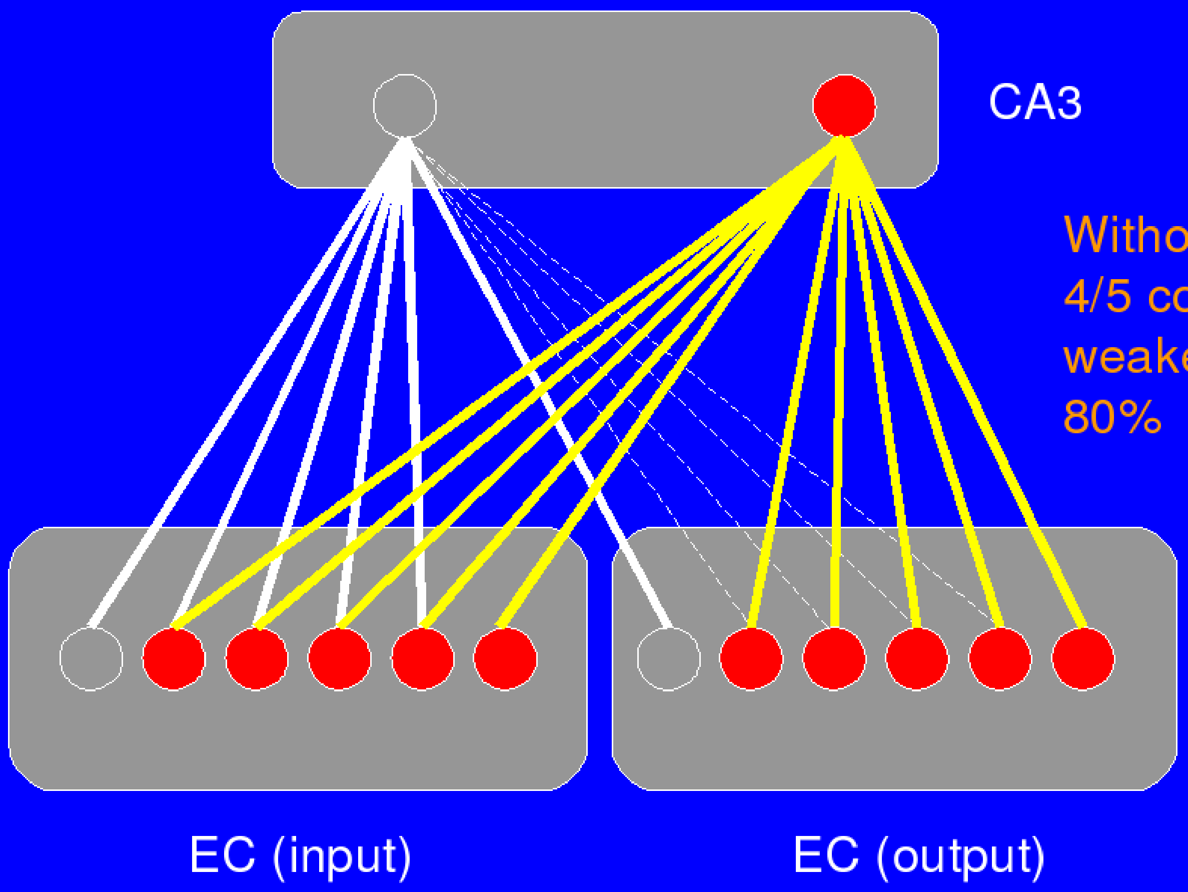


Role of CA1



With CA1:
2/4 connections
weakened = 50%

Role of CA1



Without CA1:
4/5 cons
weakened =
80%

Hippocampus: Summary

- CA3 stores sparse, pattern-separated representations of cortical input patterns

- Recurrent self projections in CA3 facilitate recall (pattern completion)

- Dentate Gyrus (DG) acts as a *removable pattern separation turbocharger*

Hippocampus: Summary

- CA3 stores sparse, pattern-separated representations of cortical input patterns
- Recurrent self projections in CA3 facilitate recall (pattern completion)
- Dentate Gyrus (DG) acts as a *removable pattern separation turbocharger*
- *CA1 helps "translate" sparse, non-overlapping CA3 representations back into overlapping EC reps, by providing an* intermediately *sparse representation*

AB-AC Learning in the Hippo Model

[hip.proj]

AB-AC Learning in the Hippo Model

[hip.proj]

- *Unlike cortical model, Hippocampus can rapidly and sequentially learn arbitrary information (AB-AC lists) without huge amounts of interference.*

- Cortex still critical for slow learning of overlapping, distributed representations, supporting *generalized* knowledge, semantic information, and similarity.

AB-AC Learning in the Hippo Model

[hip.proj]

- *Unlike cortical model, Hippocampus can rapidly and sequentially learn arbitrary information (AB-AC lists) without huge amounts of interference.*

- Cortex still critical for slow learning of overlapping, distributed representations, supporting *generalized* knowledge, semantic information, and similarity.

- Later: How learning/memory capacity can be enhanced with theta waves (Ken Norman)

Memory

Memory is not unitary.

1. Weights (long-lasting, requires re-activation) versus activations (short-term, already active, can influence processing).

2. Specialized neural systems: computational tradeoffs. Cortex shows priming, but suffers catastrophic interference. Abandon neural network models? *No, hippocampus can learn rapidly without interference using sparse, pattern-separated representations.*

Memory

Memory is not unitary.

1. Weights (long-lasting, requires re-activation) versus activations (short-term, already active, can influence processing).

2. Specialized neural systems: computational tradeoffs. Cortex shows priming, but suffers catastrophic interference. Abandon neural network models? *No, hippocampus can learn rapidly without interference using sparse, pattern-separated representations.*

3. Next time: Activation-based memory and activation-weight-based interactions.

Memory

Memory is not unitary.

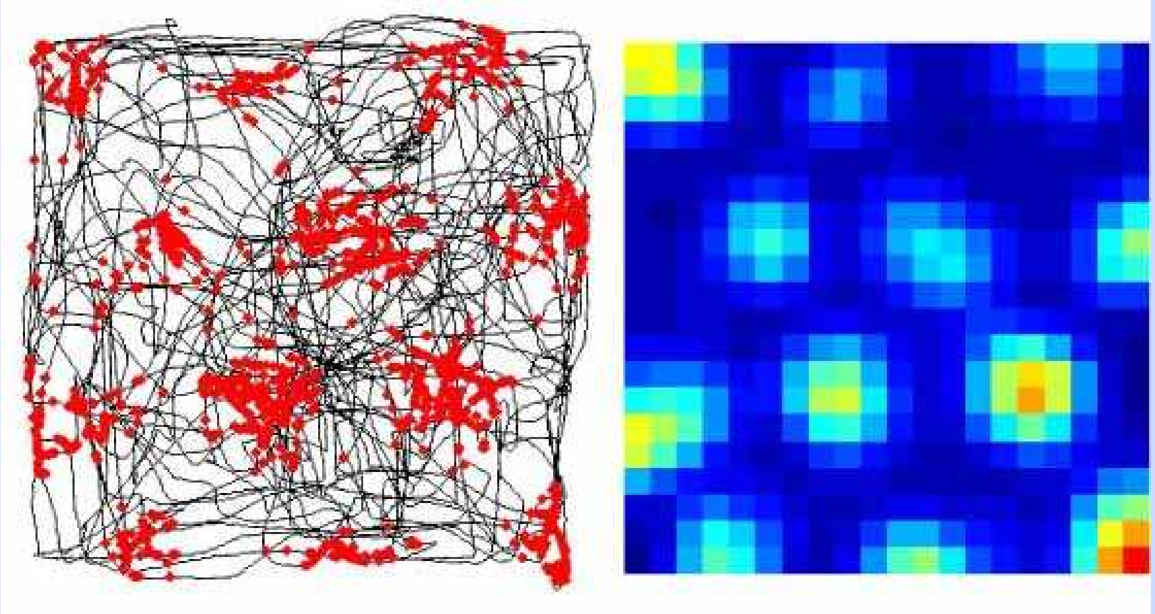
1. Weights (long-lasting, requires re-activation) versus activations (short-term, already active, can influence processing).

2. Specialized neural systems: computational tradeoffs. Cortex shows priming, but suffers catastrophic interference. Abandon neural network models? *No, hippocampus can learn rapidly without interference using sparse, pattern-separated representations.*

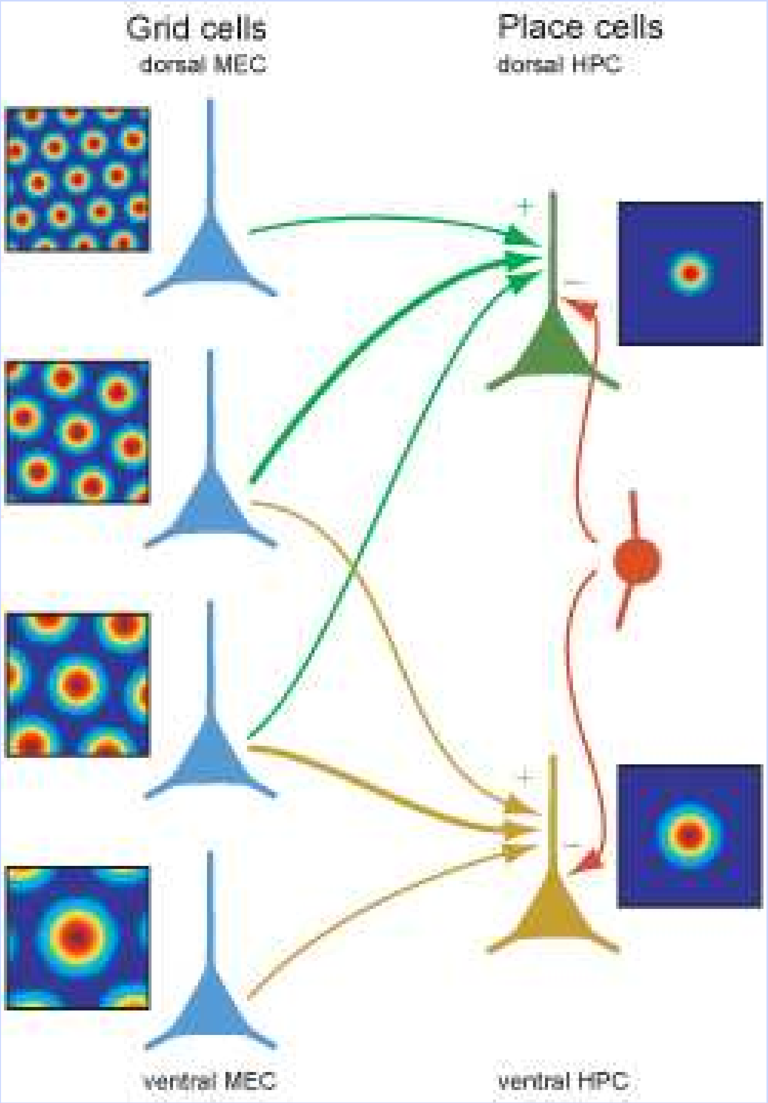
3. Next time: Activation-based memory and activation-weight-based interactions.

Hippo and spatial topography: what about “grid cells”?

Hippo and spatial topography: what about "grid cells"?



- Grid cells are in medial entorhinal cortex (Hafting et al, 2005), not hippo proper
- Hippo might integrate location with speed and direction ("head direction cells") to perform *path integration*
- This can be recast as just another example of conjunctive, pattern-separate representations



Solstad et al, 2006

Memory

Memory is not unitary.

1. Weights (long-lasting, requires re-activation) versus activations (short-term, already active, can influence processing).
2. Weight-based: Cortex shows priming, but suffers catastrophic interference. Hippocampus can learn rapidly without interference using sparse, pattern-separated representations.
3. Activation-based: Cortex shows priming, but can’t do working memory.
4. Activation- and weight-based interactions.

Cortical Priming

Even slow cortical weight changes can yield one-trial learning effects..

win____

Cortical Priming

Even slow cortical weight changes can yield one-trial learning effects..

win____

handle

Cortical Priming

Even slow cortical weight changes can yield one-trial learning effects..

win____

handle
winter

Cortical Priming

Even slow cortical weight changes can yield one-trial learning effects..

win____

handle
winter
shower...

Cortical Priming

Even slow cortical weight changes can yield one-trial learning effects..

win____

handle
winter
shower...
win____

Cortical Priming

Even slow cortical weight changes can yield one-trial learning effects..

win____

handle
winter
shower...
win____

Spell $/r\bar{e}d/$.

Cortical Priming

Even slow cortical weight changes can yield one-trial learning effects..

win____

handle
winter
shower...
win____

Spell $/r\bar{e}d/$.
Name a musical instrument that uses a reed.

Cortical Priming

Even slow cortical weight changes can yield one-trial learning effects..

win____

handle
winter
shower...
win____

Spell $/r\bar{e}d/$.
Name a musical instrument that uses a reed.
Spell $/r\bar{e}d/$.

Cortical Priming

- There are many, many types of priming effects:
  - Stem-completion & phonetic priming

Cortical Priming

- There are many, many types of priming effects:
  - Stem-completion & phonetic priming
  - Perceptual identification → faster, more accurate detection after recent exposure to words (even hours later)

Cortical Priming

- There are many, many types of priming effects:
  - Stem-completion & phonetic priming
  - Perceptual identification → faster, more accurate detection after recent exposure to words (even hours later)
  - Category generation priming: *"peach, kiwi"*; <many hrs later> → *"name some fruits"* (in absence of recall)

Cortical Priming

- There are many, many types of priming effects:
  - Stem-completion & phonetic priming
  - Perceptual identification → faster, more accurate detection after recent exposure to words (even hours later)
  - Category generation priming: *"peach, kiwi"*;
    <many hrs later> → *"name some fruits"* (in absence of recall)
- Cortex is the key substrate for these priming effects
- Patients with hippocampus damage (sparing cortex) show impaired recall but intact priming
- These priming effects are long-lasting
  - This indicates that a weight change is involved
    (unlikely for activations to persist for long periods)

Simulations of Cortical Priming

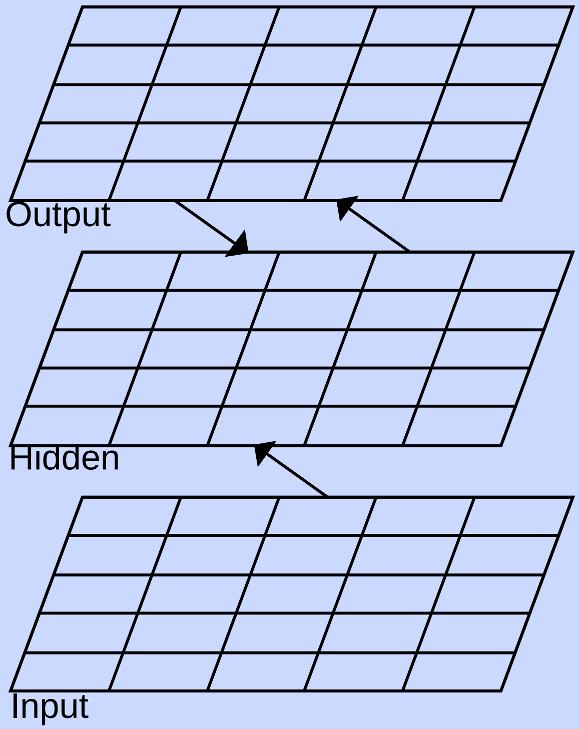
- Train a network to learn input-output mappings

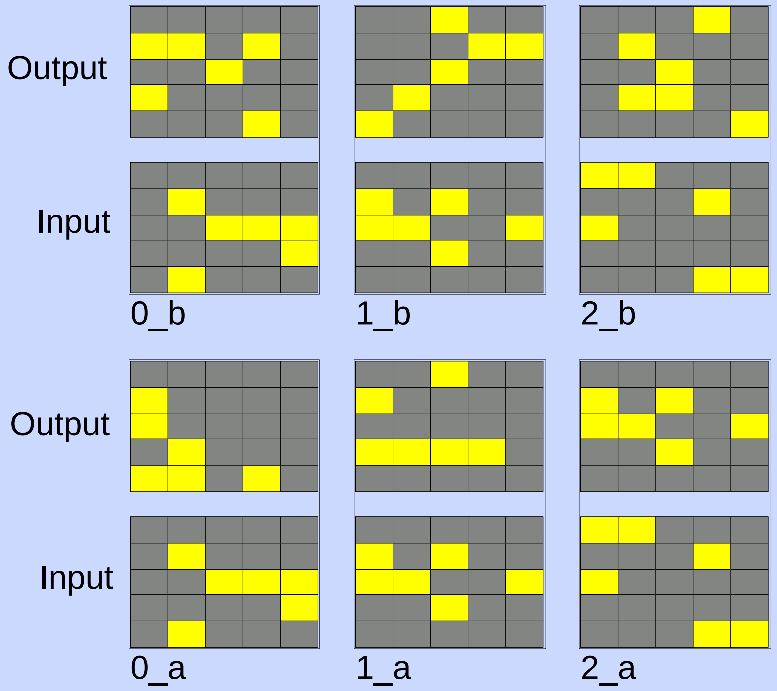
- Each input is associated with two valid outputs

- Analogous to:

win____ → window
win____ → winter
$/r\bar{e}d/$ → “read”
$/r\bar{e}d/$ → “reed”

Weight-based Priming Model



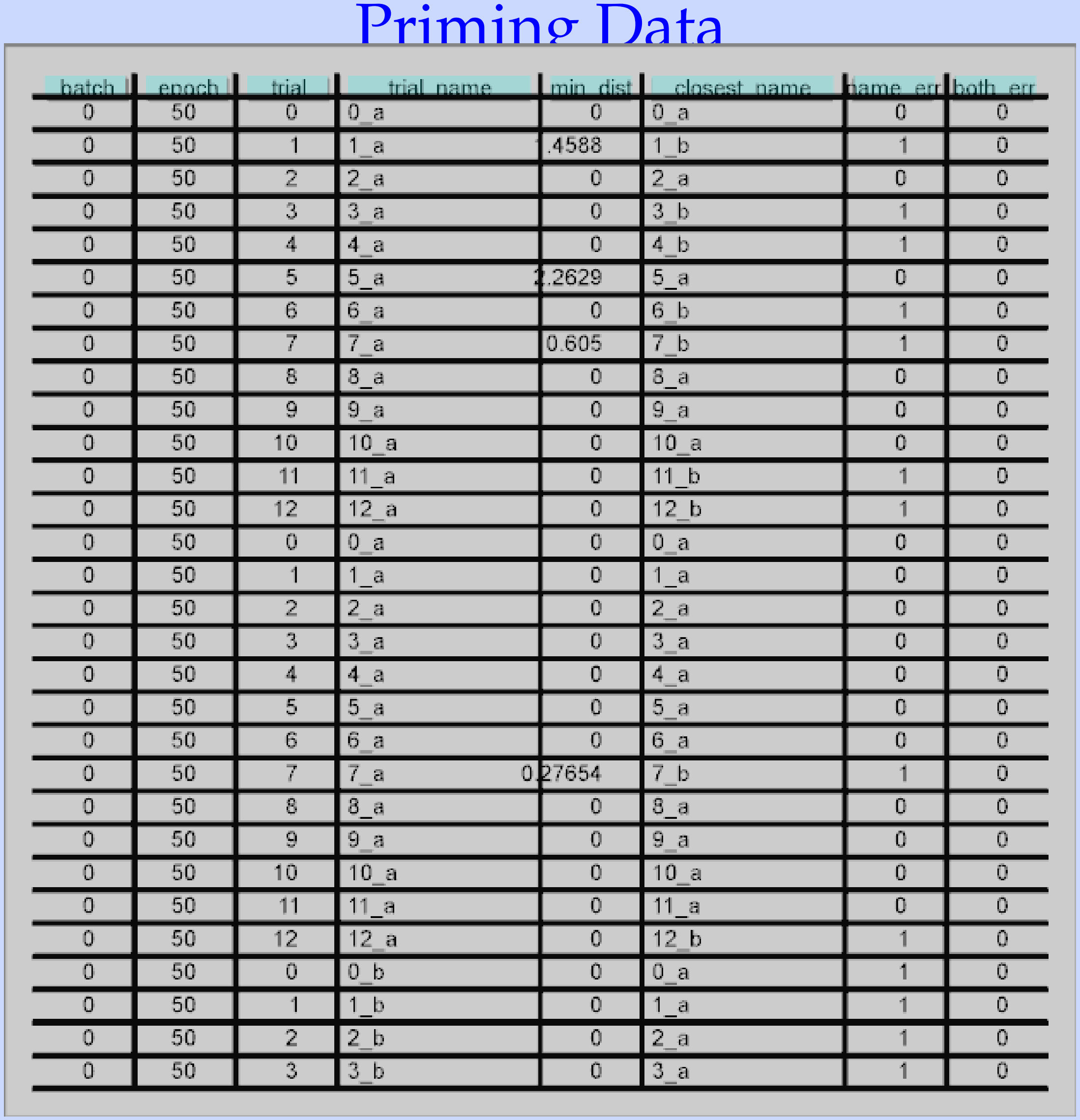


[wt_priming.proj]

Priming Simulations

- After training, the network is equally likely to produce the “a” or “b” output in response to a cue...
- Does not “blend” the two, but instead settles into one of the two valid attractors
- How does one additional study trial with the “a” input affect performance?
- Small weight changes (resulting from a single study trial) can “tip the balance” in favor of the recently studied response...

Priming Data



batch	epoch	trial	trial_name	min_dist	closest_name	name_err	both_err
0	50	0	0_a	0	0_a	0	0
0	50	1	1_a	1.4588	1_b	1	0
0	50	2	2_a	0	2_a	0	0
0	50	3	3_a	0	3_b	1	0
0	50	4	4_a	0	4_b	1	0
0	50	5	5_a	2.2629	5_a	0	0
0	50	6	6_a	0	6_b	1	0
0	50	7	7_a	0.605	7_b	1	0
0	50	8	8_a	0	8_a	0	0
0	50	9	9_a	0	9_a	0	0
0	50	10	10_a	0	10_a	0	0
0	50	11	11_a	0	11_b	1	0
0	50	12	12_a	0	12_b	1	0
0	50	0	0_a	0	0_a	0	0
0	50	1	1_a	0	1_a	0	0
0	50	2	2_a	0	2_a	0	0
0	50	3	3_a	0	3_a	0	0
0	50	4	4_a	0	4_a	0	0
0	50	5	5_a	0	5_a	0	0
0	50	6	6_a	0	6_a	0	0
0	50	7	7_a	0.27654	7_b	1	0
0	50	8	8_a	0	8_a	0	0
0	50	9	9_a	0	9_a	0	0
0	50	10	10_a	0	10_a	0	0
0	50	11	11_a	0	11_a	0	0
0	50	12	12_a	0	12_b	1	0
0	50	0	0_b	0	0_a	1	0
0	50	1	1_b	0	1_a	1	0
0	50	2	2_b	0	2_a	1	0
0	50	3	3_b	0	3_a	1	0

Cortical Priming

Residual activation can also result in priming.
(Activation-based priming: later)

Three factors:

- Duration (short-term activations vs long-term weights).
- Content (visual, semantic, etc.)
- Similarity (repetition, semantic relation, etc).

Remember Weight-Based Priming?

r a	en	s se	d s	e n	s n	o err
0	0_a	5.22935	0	0_b	1	0
1	1_a	6.48608	0	1_b	1	0
2	2_a	7.77501	0.273233	2_b	1	0
3	3_a	7.64788	0	3_b	1	0
4	4_a	5.41569	0.551383	4_b	1	0
5	5_a	0	0	5_a	0	0
6	6_a	10.2454	0	6_b	1	0
7	7_a	8.33851	0	7_b	1	0
8	8_a	5.64973	2.61438	8_b	1	0
9	9_a	10.2408	0	9_b	1	0
10	10_a	3.21385	1.06278	10_b	1	0
11	11_a	2.82117	2.42077	11_b	1	0
12	12_a	4.69916	0.253711	12_b	1	0
13	0_b	6.68981	0	0_a	1	0
14	1_b	5.40769	0.330821	1_a	1	0
15	2_b	7.51547	0	2_a	1	0
16	3_b	7.73557	0	3_a	1	0
17	4_b	1.94789	1.94789	4_b	0	0
18	5_b	0.414954	0.414954	5_b	0	0
19	6_b	10.5514	0	6_a	1	0
20	7_b	8.79166	0	7_a	1	0
21	8_b	9.64561	0	8_a	1	0
22	9_b	10.2245	0	9_a	1	0
23	10_b	3.53423	0.766472	10_a	1	0
24	11_b	7.46935	0	11_a	1	0
25	12_b	5.72054	0	12_a	1	0

Activation-Based Priming

Residual activation can also result in priming: *act_priming.proj*

No learning (wt changes), to see effects of activation alone.

Activation-based Priming: Residual Activation

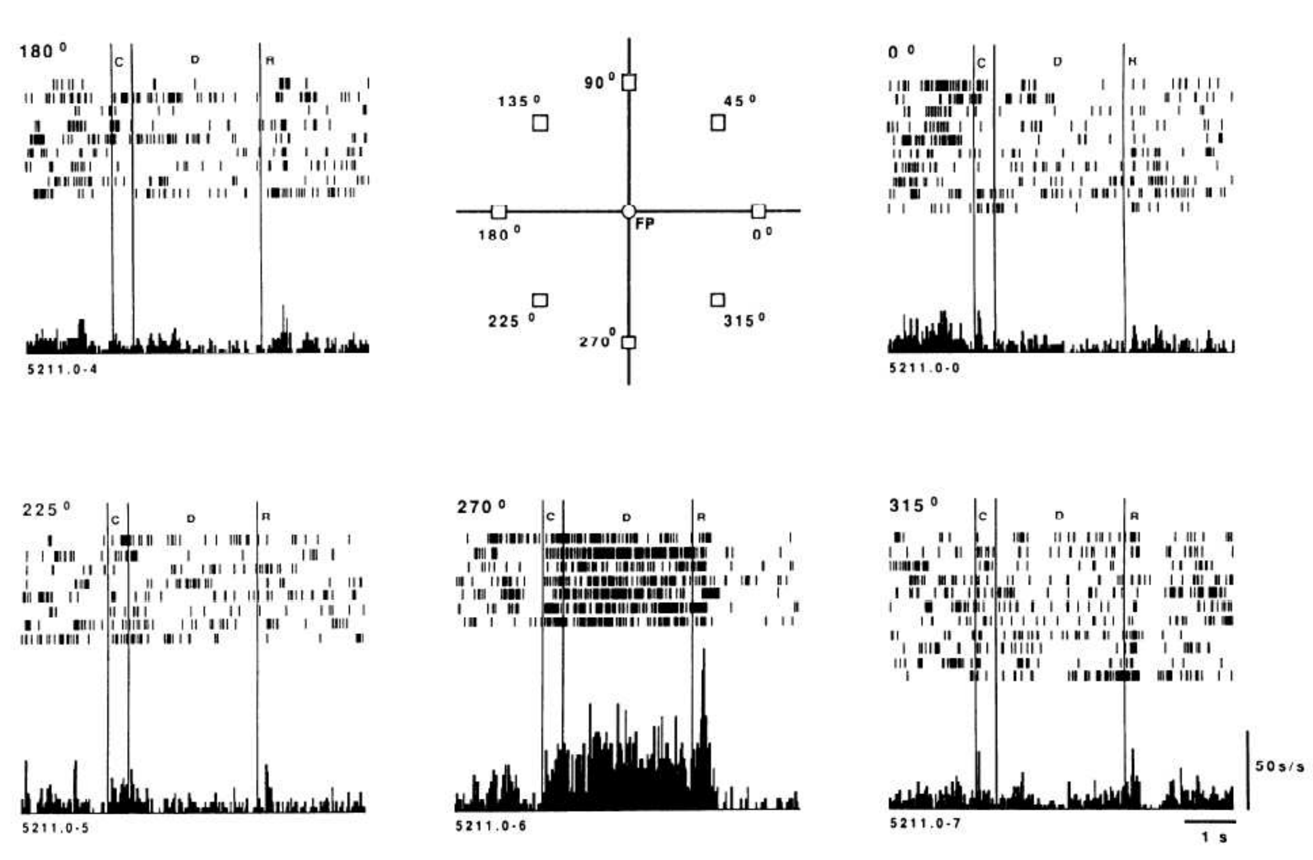
r a	en	s se	d s	e n	s n	o err
0	0_a	0	0	0_a	0	0
1	0_b	1.7529	1.7529	0_b	0	0
2	1_a	0	0	1_a	0	0
3	1_b	2.18947	2.06997	1_a	1	0
4	2_a	0	0	2_a	0	0
5	2_b	5.43822	0.467382	2_a	1	0
6	3_a	0	0	3_a	0	0
7	3_b	1.05335	1.05335	3_b	0	0
8	4_a	0	0	4_a	0	0
9	4_b	6.26163	0.663053	4_a	1	0
10	5_a	0	0	5_a	0	0
11	5_b	4.02698	2.36882	5_a	1	0
12	6_a	0	0	6_a	0	0
13	6_b	5.74102	2.00435	6_a	1	0
14	7_a	0	0	7_a	0	0
15	7_b	8.85609	0	7_a	1	0
16	8_a	0	0	8_a	0	0
17	8_b	9.4205	0.444151	8_a	1	0
18	9_a	0	0	9_a	0	0
19	9_b	7.888	1.64196	9_a	1	0
20	10_a	0	0	10_a	0	0
21	10_b	5.20613	0.337607	10_a	1	0
22	11_a	0	0	11_a	0	0
23	11_b	6.4702	1.40431	11_a	1	0
24	12_a	0	0	12_a	0	0
25	12_b	5.32969	0.33391	12_a	1	0

Activation-based Priming: Residual Activation

r a	en	s se	d s	e n	s n	o err
0	0_a	0	0	0_a	0	0
1	0_b	1.7529	1.7529	0_b	0	0
2	1_a	0	0	1_a	0	0
3	1_b	2.18947	2.06997	1_a	1	0
4	2_a	0	0	2_a	0	0
5	2_b	5.43822	0.467382	2_a	1	0
6	3_a	0	0	3_a	0	0
7	3_b	1.05335	1.05335	3_b	0	0
8	4_a	0	0	4_a	0	0
9	4_b	6.26163	0.663053	4_a	1	0
10	5_a	0	0	5_a	0	0
11	5_b	4.02698	2.36882	5_a	1	0
12	6_a	0	0	6_a	0	0
13	6_b	5.74102	2.00435	6_a	1	0
14	7_a	0	0	7_a	0	0
15	7_b	8.85609	0	7_a	1	0
16	8_a	0	0	8_a	0	0
17	8_b	9.4205	0.444151	8_a	1	0
18	9_a	0	0	9_a	0	0
19	9_b	7.888	1.64196	9_a	1	0
20	10_a	0	0	10_a	0	0
21	10_b	5.20613	0.337607	10_a	1	0
22	11_a	0	0	11_a	0	0
23	11_b	6.4702	1.40431	11_a	1	0
24	12_a	0	0	12_a	0	0
25	12_b	5.32969	0.33391	12_a	1	0

But what about when need to maintain over longer delays (working memory)??

Prefrontal Cortex: Delay-related activity



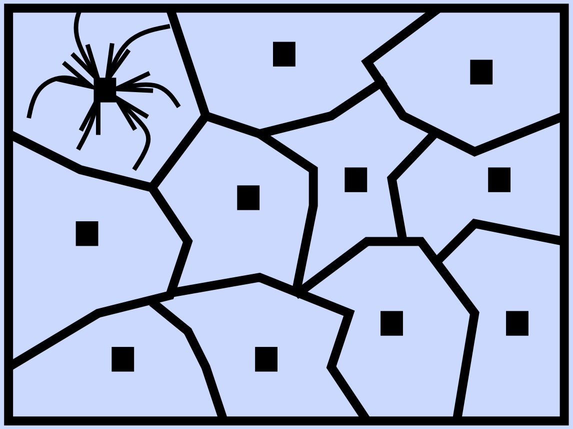
Spatial delayed-response task; Funahashi et al, 1989

Active Maintenance

Maintaining information in active form over longer time periods.

Can be used for *working memory* (e.g., in mental arithmetic).

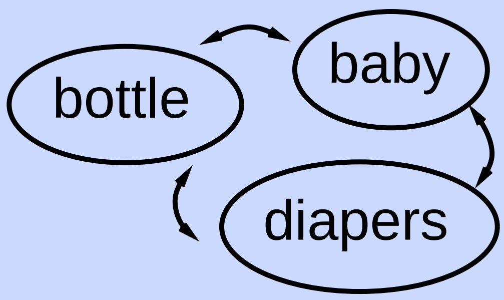
Attractor = stable activation state:



(don’t want activity to spread)

Prefrontal vs. Posterior Cortex

Posterior cortex: interactive reps w/spreading activation



Advantages	*Disadvantages*
Semantic associations	Memory confusion
Inference (diapers → baby)	
Schema (parenting)	

Prefrontal: isolated reps, maintenance w/out activation spread

Attractors: Summary

- To get robustness from noise, you need *isolated* representations with strong recurrent connections

- This prevents activity from spreading

- Tradeoff #1: Preventing spreading activation (active maintenance) vs. allowing spreading activation (inference)

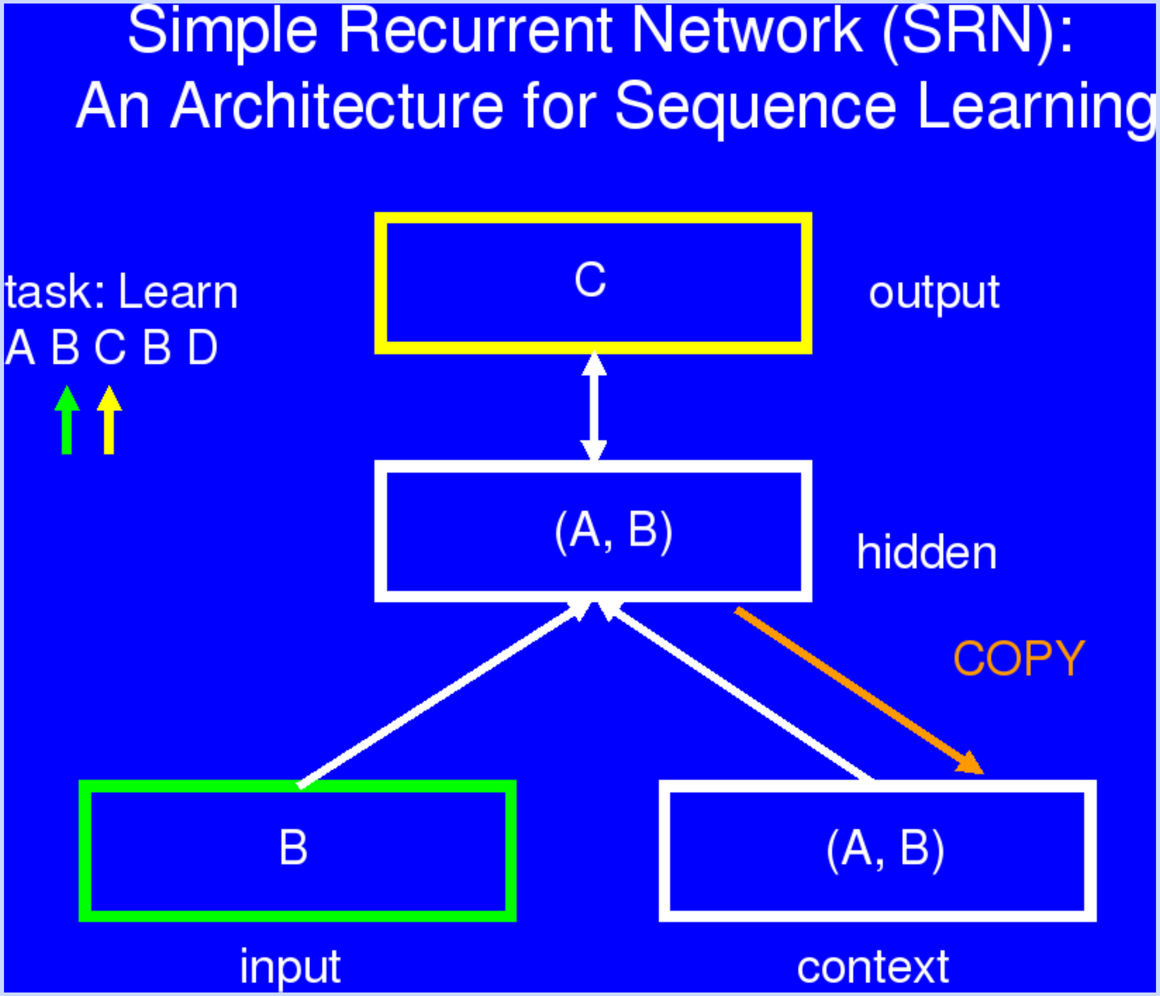
- Solution: Posterior cortex uses interconnected representations → spreading activation; prefrontal cortex (PFC) uses isolated reps → prevents spreading activation

- Evidence for isolated *stripes* in PFC (Levitt et al, 93; Pucak et al, 96)

Attractors: Summary

- Tradeoff #2: Within PFC, need for robust maintenance vs. need to update PFC activation when appropriate
- Strong recurrents (weak inputs) = robust maintenance
- Weak recurrents (strong inputs) = rapid updating
- We need a mechanism for switching PFC between the two modes
- Also, how to *learn* when to update?

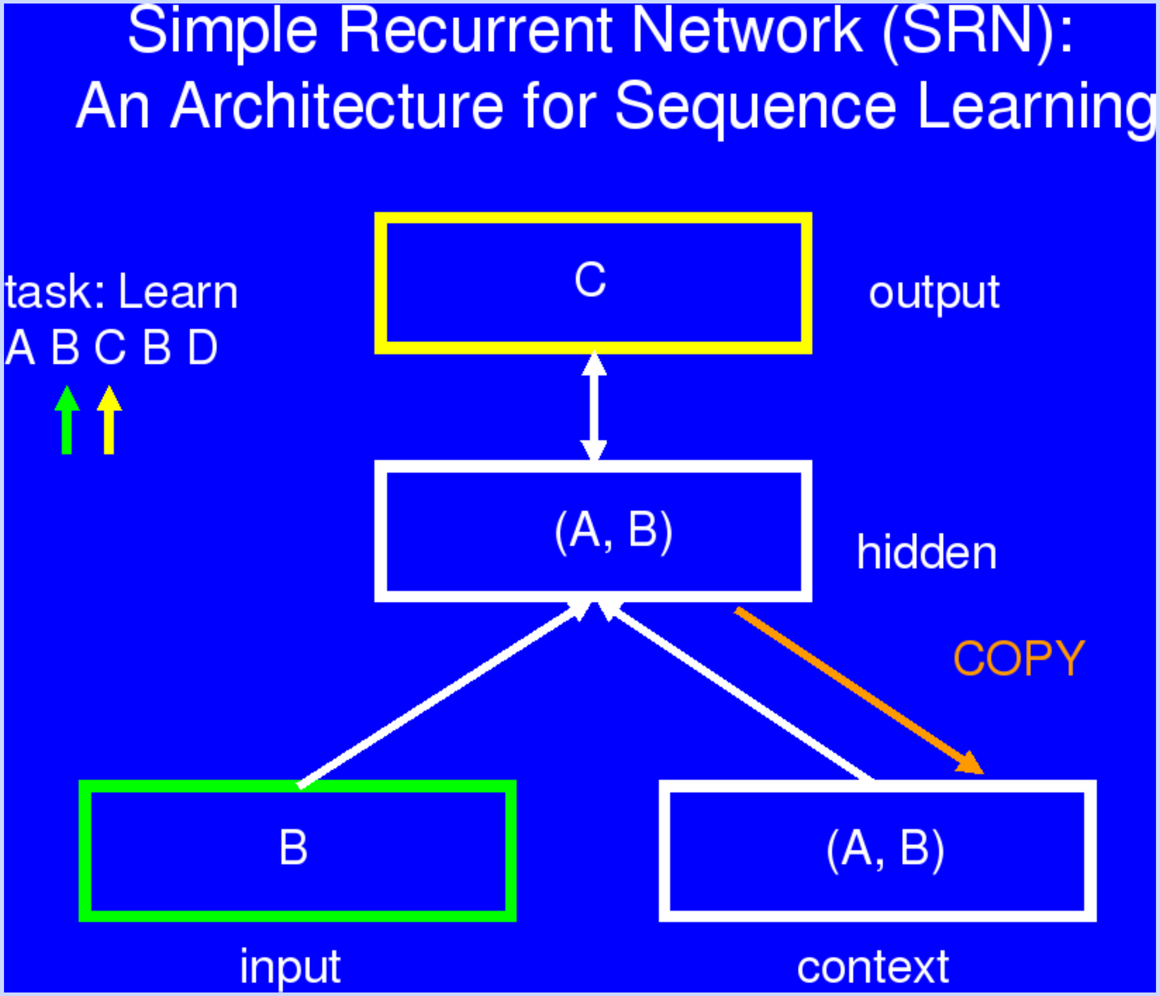
Remember the SRN? (chap 6)



Simple Recurrent Network (SRN):
An Architecture for Sequence Learning

task: Learn
A B C B D

Remember the SRN? (chap 6)



Simple Recurrent Network (SRN):
An Architecture for Sequence Learning

task: Learn
A B C B D

this is a *gating* network: context only updated at discrete timepoints

Simple SRN story is not flawless

- How is hidden→ “copy” function implemented biologically?

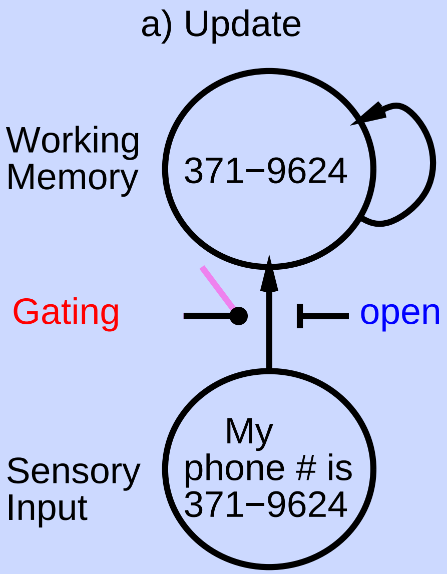
- During settling, context must be *actively maintained* (ongoing hidden activity has no effect on context).

- Assumes all context is relevant: What if distracting information presented in middle of sequence? Want to only hold on to *relevant* context.

- What if want to hold on to more than one piece of information in WM at a time?? Or to selectively update one part of WM while continuing to robustly maintain others?

- And what if the decision of whether or not to update information *depends* on currently internal WM state?

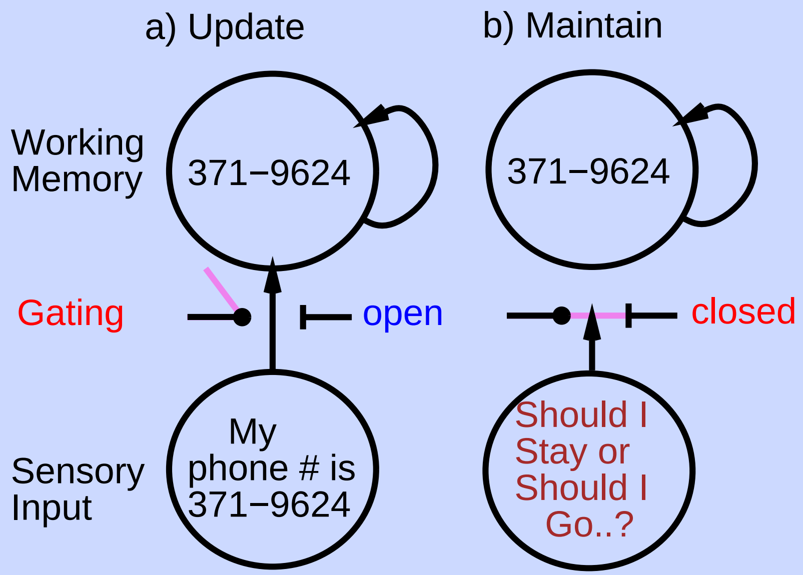
Working Memory Demands: Updating & Maintenance



- Working memory: robust maintenance of information, but must also have ability to be rapidly updated — requires *gating*.

- You've got to know when to hold 'em, know when to fold 'em.

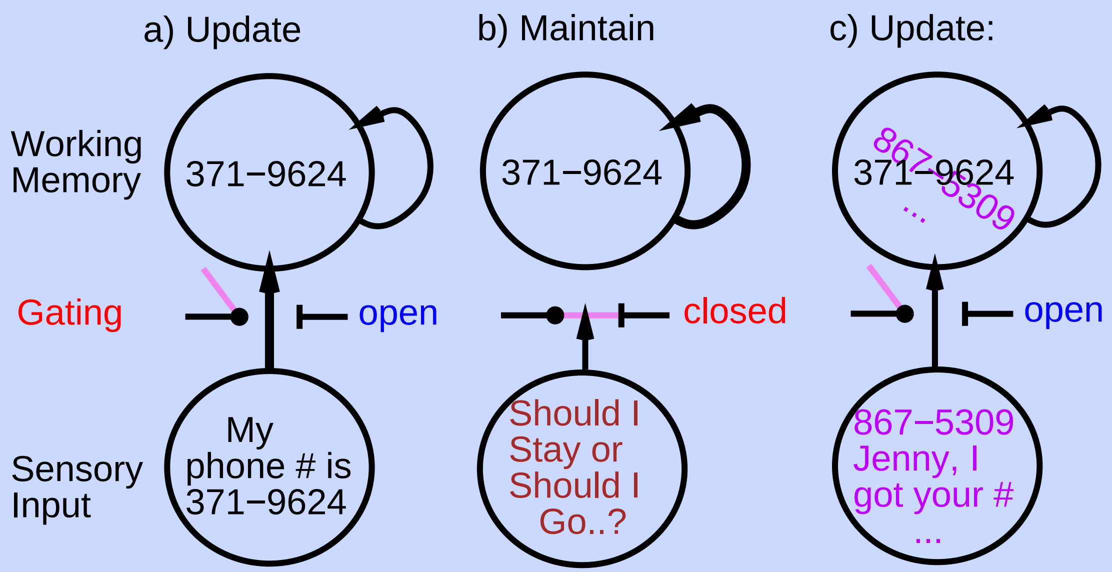
Working Memory Demands: Updating & Maintenance



- Working memory: robust maintenance of information, but must also have ability to be rapidly updated — requires *gating*.

- You’ve got to know when to hold ’em, know when to fold ’em.

Working Memory Demands: Updating & Maintenance

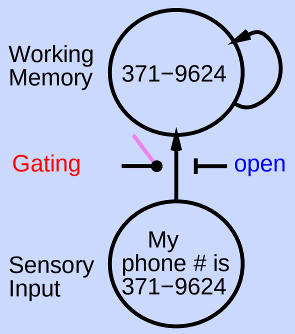


- Working memory: robust maintenance of information, but must also have ability to be rapidly updated — requires *gating*.

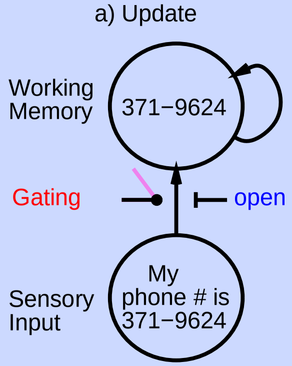
- You've got to know when to hold 'em, know when to fold 'em.

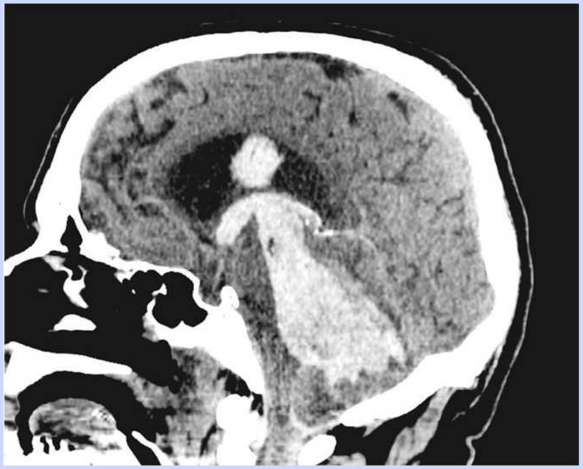
But who controls the controller??

a) Update



But who controls the controller??

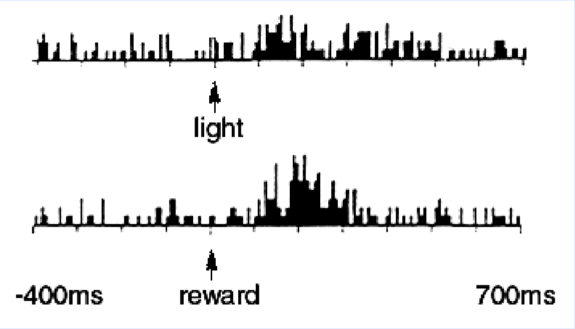




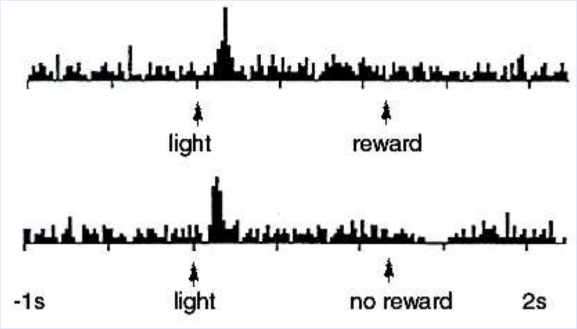
But who controls the controller??

But who controls the controller??

Before:



After:



[First pass story (Braver & Cohen, ’00 and text):]

Dopamine provides dynamic *gating* mechanism:

- Positive TD δ (reward) = DA burst = update PFC.
- No TD δ = constant DA = maintain PFC.
- Negative TD δ (error) = DA dip = clear PFC.

The *same* DA signal that learns to predict reward can be used to drive updating of PFC states!

DA solves part of the problem

- Learning signal for gating.

- But DA is very global signal projecting to all of PFC – sufficent for updating and maintaining one item at a time.

- How to *selectively* update some aspects of WM but not others?

- Also prev DA-PFC model had awkward catch-22 problem: the stimulus is only predictive of reward if it is maintained (ie in PFC). But then stim needs to be gated into PFC in the first place to generate DA!

DA solves part of the problem

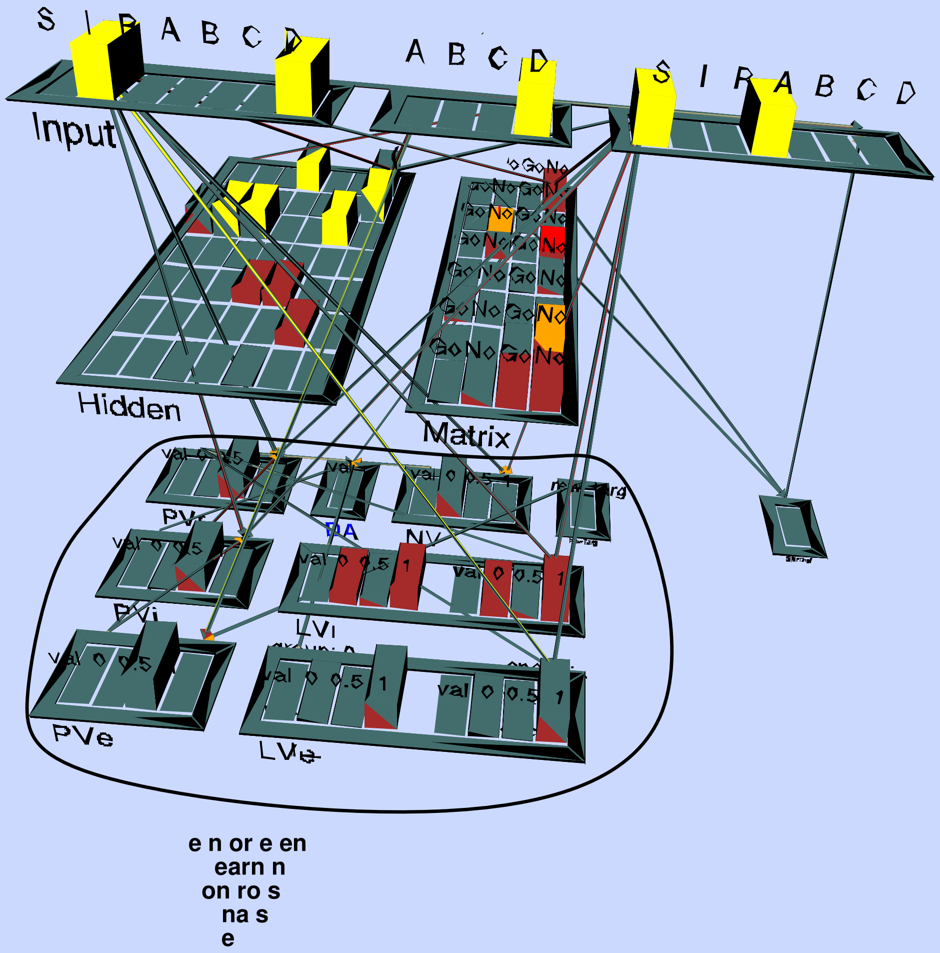
- Learning signal for gating.
- But DA is very global signal projecting to all of PFC – sufficent for updating and maintaining one item at a time.
- How to *selectively* update some aspects of WM but not others?
- Also prev DA-PFC model had awkward catch-22 problem: the stimulus is only predictive of reward if it is maintained (ie in PFC). But then stim needs to be gated into PFC in the first place to generate DA!
- Solution: separate learning from gating... and link to now well established role of basal ganglia-thalamus in gating.

Dynamic Gating: Current Story

- DA signals are important for learning/knowing when to gate
- But actual gating signals are implemented via more complex circuit interactions with the Basal Ganglia Go/NoGo system
- DA used to train Go/NoGo system exactly like in the motor and simple decision making domains...
- BG-gating solves multiple computational and biological plausibility issues that are problematic with pure-DA based gating
- Goto BG_PFC_WM1.pdf slides for more info and evidence

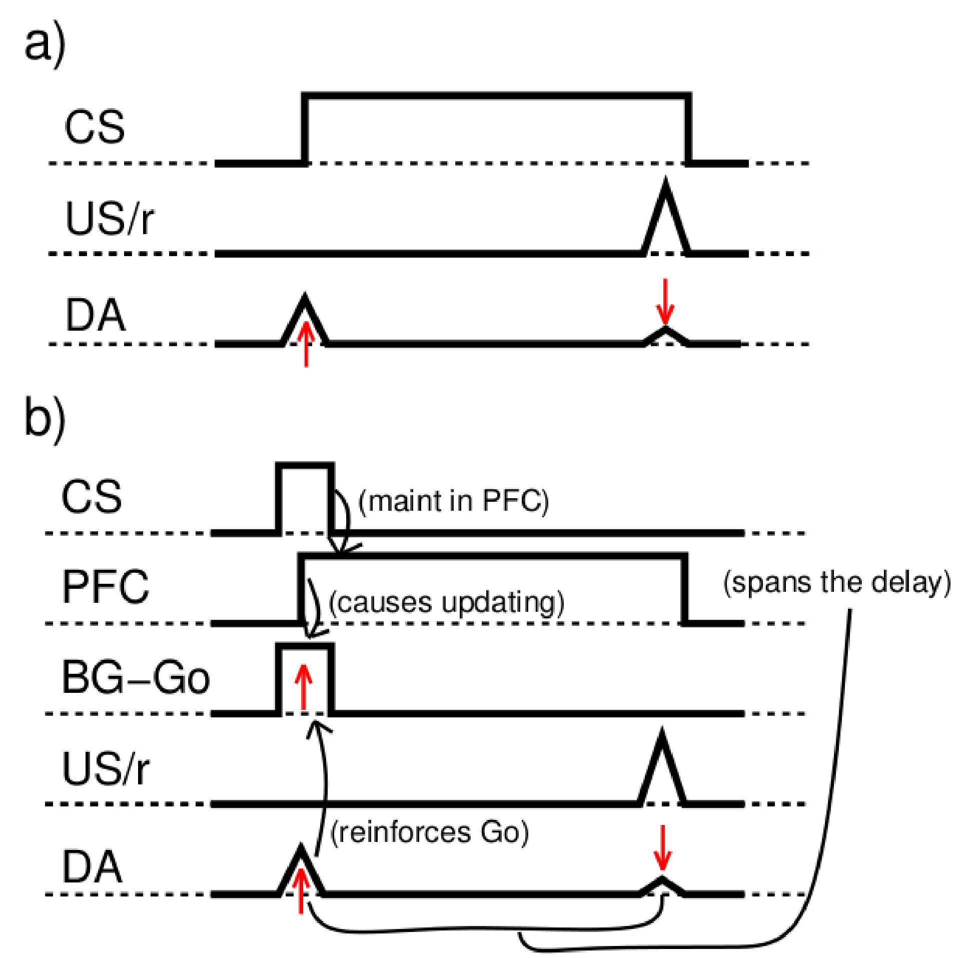
A Simple WM Task

Trial	Input	Maint	Output
1	STORE-A	A	A
2	IGNORE-B	A	B
3	IGNORE-C	A	C
4	IGNORE-D	A	D
5	RECALL	A	A



PFC/BG Model: sir.proj

PFC maintenance enables RL to train BG



Reinforcement learning and WM gating

- Network learns to associate stimuli with rewards via PVLV / DA system (like TD)

- PVLV gets information not only from outside world, but also PFC state

- *Desired outcome:* Network learns that having the STORE pattern in PFC leads to rewards, but having the IGNORE pattern does not

Reinforcement learning and WM gating

- Bursts and dips of DA train the basal ganglia Go/NoGo gating system

- If BG system gates an input into PFC **and** that PFC pattern had been associated with reward → DA burst (DA system recognizes this new PFC pattern as rewarding)

- This DA burst reinforces *Go* activity in the BG units that caused the gating in the first place, making it even more likely that the BG will gate this pattern into PFC on *future trials*. (phasic DA does *not* directly drive updating itself, but is a learning signal)

- *Desired outcome:* Networks learns “Go” to gate STORE into PFC, but learns “NoGo” to IGNORE

Sketch of how the network learns

- Begins with trial-and-error learning (both at response output and in BG gating system)
- Explore different gating "policies" and reinforce ones that work. (some amount noise helpful!)
- If correct response happens to occur when STORE is active in PFC (initially due to guessing) ⇒ Reward
- Resulting DA burst trains PVLV (or TD) system to learn that having "STORE" in PFC is a good thing

Sketch of how the network learns

- Next time STORE is represented in PFC, PVLV system triggers a DA burst, based on its learned PFC-reward association (without needing external reward)

- This DA burst drives BG Go learning so that good stimuli are more likely to be gated

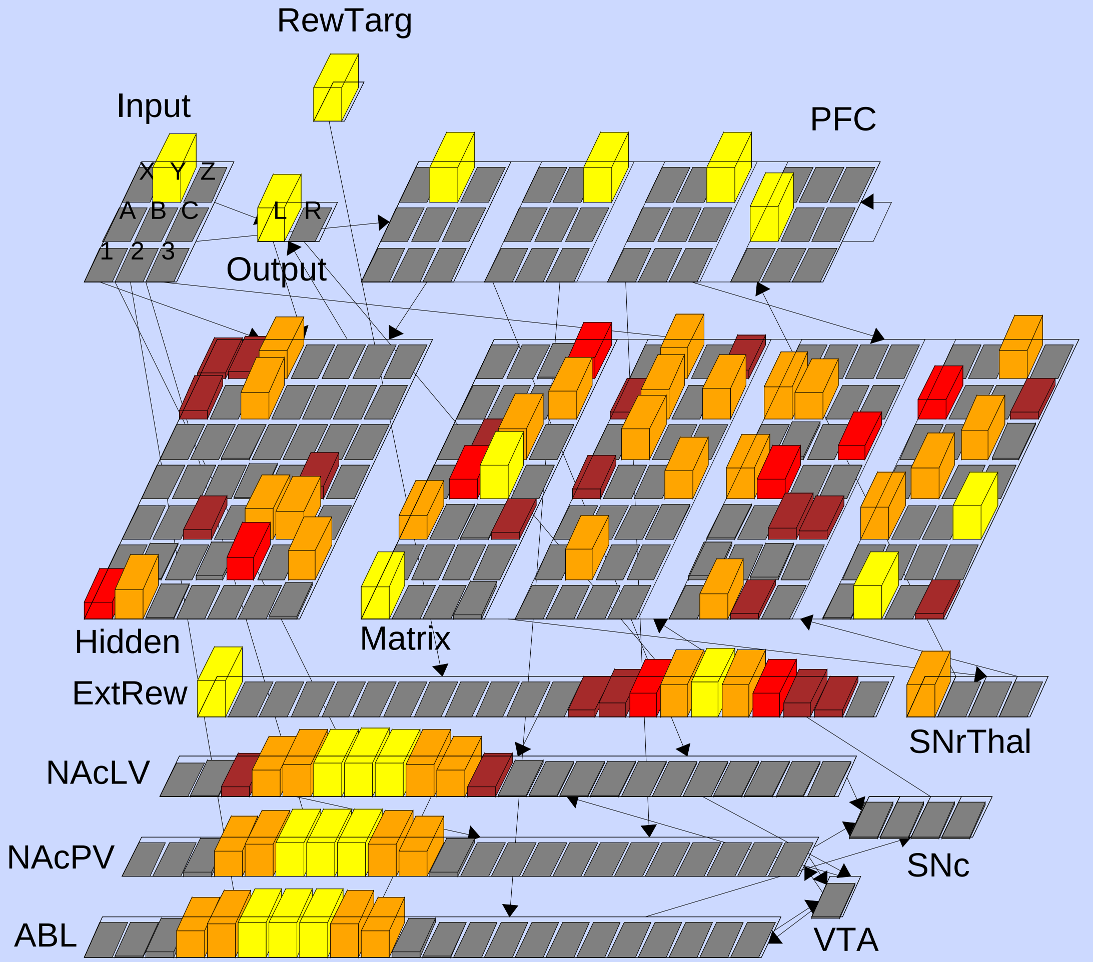
- In turn, stored information is more likely to be present in PFC during RECALL trial.

- At this point, Hidden layer simply has to learn to map PFC representation of stored stimulus to the Output response.

- This leads to increased rewards, further training gating system, and leading to stable state.

Four "Stripe" PFC/BG Model, Learns with DA

(O'Reilly & Frank, 2006)

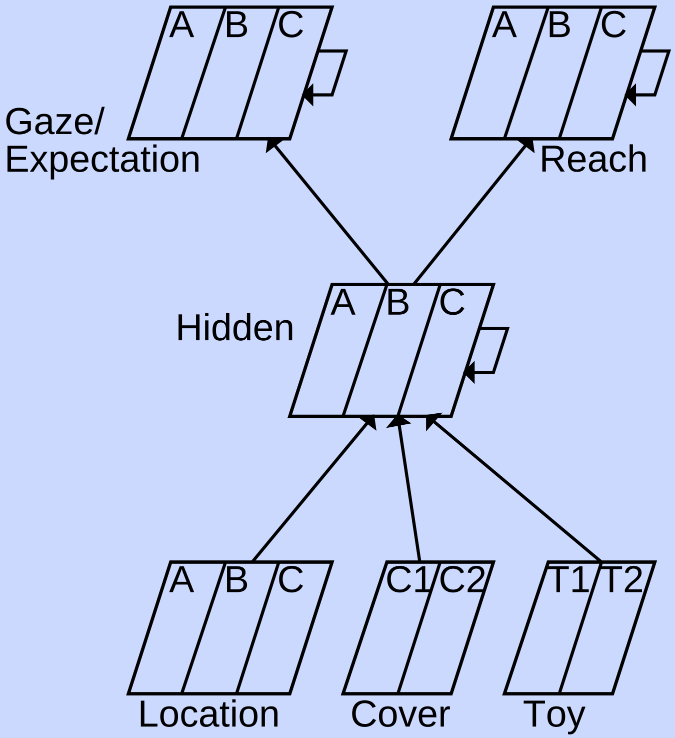


Weight- and Activation-Based Memory Interactions

A-not-B task

- Perseverative searching at A – also seen in patients with PFC damage
- Better peformance in gaze/expectation
- Inhibition problem?
- Model demonstrates maintenance problem.
- Same model accounts for various effects in different versions of A-not-B task not explained by any other unified theory (Munakata, 1998).

A-not-B Model


A
B
C
Gaze/
Expectation
A
B
C
Reach
Hidden
A
B
C
A
B
C
Location
C1
C2
Cover
T1
T2
Toy

Knowledge-action dissociations in card-sort task

- Kids can tell you where trucks go in the shape game, even after sorting according to color!

- But if you ask “where do red trucks go in the shape game” they still fail! (Morton & Munakata, 2002)

- Explained by different levels of conflict experienced when faced with multiple stimuli-response associations..