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

Weights-based Memories

- Cortex does gradual, integrative learning
- Given:
  - Multiple passes through the input
  - Relatively small learning rate

Activation-based Memories

- Spreading activation causes interference
- Two systems needed:
  - Interconnected posterior cortex
  - Isolated prefrontal cortex

Cortical Priming

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

Memory: Weights vs Activations

- Despite appearances, memory is not unitary
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
/red/ → "read"
/red/ → "reed"

Weight-based Priming Model

<table>
<thead>
<tr>
<th>Input</th>
<th>Hidden</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0_a</td>
<td>1_a</td>
<td>2_a</td>
</tr>
<tr>
<td>0_b</td>
<td>1_b</td>
<td>2_b</td>
</tr>
</tbody>
</table>

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

| Trial | 0_a | 1_a | 2_a | 3_a | 4_a | 5_a | 6_a | 7_a | 8_a | 9_a | 10_a | 11_a | 12_a | 0_b | 1_b | 2_b | 3_b | 4_b | 5_b | 6_b | 7_b | 8_b | 9_b | 10_b | 11_b | 12_b |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|-----|

Outp_dist

sm_nm
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).

Memory: Rapid Learning, Interference, & The Hippocampus

1. 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

Then test on AB list:

- window-
- bicycle-

Then test on AC list:

- window-
- bicycle-

AB-AC List Learning

% Correct

0 5 10 15 20 25 30 35 40 45 50

Learning Trials on AC List

0 25 50 75 100

AB List

AC List

b) AB−AC List Learning in Model

Standard network shows catastrophic interference

(Rescorla & Cohen, 1986).

Standard network shows catastrophic interference

Standard network shows catastrophic interference

Standard network shows catastrophic interference

Humans can rapidly learn overlapping associations without too much interference.

2. The Hippocampus

1. AB-AC List Learning

3. How the Hippocampus

4. The Hippocampus

- memory: rapid learning, interference, & the
- The hippocampus

- Memory: Rapid Learning, Interference, & The Hippocampus

- Similarity (repetition, semantic relation, etc)
- Complementary (visual, semantic, etc)
- Duration (short-term associations vs long-term weights)
- Residual addiction can also result in priming

Correction Priming
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?

→ Lower the number of units that are activated → increase inhibition (decrease k)....

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 the context when they see 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...

→ 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 use another brain system!

Memory is not unitary.

1. Weights versus activations.

2. Specialized neural systems: computational tradeoffs.

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.

Memory is not unitary.

1. Weights versus activations.

2. Specialized neural systems: computational tradeoffs.

Memory is not unitary.

1. Weights versus activations.

2. Specialized neural systems: computational tradeoffs.

Memory is not unitary.

1. Weights versus activations.

2. Specialized neural systems: computational tradeoffs.

Memory is not unitary.

1. Weights versus activations.

2. Specialized neural systems: computational tradeoffs.

Memory is not unitary.
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:

Where is car parked?

Best parking strategy?

Need to:

Avoid interference

Accumulate experience

Solution:

1. Separate reps

(keep days separate)

2. Fast learning

(encode immediately)

3. Learn automatically

(Task-driven learning)

These are incompatible, need two different systems:

System:

Hippocampus

Neocortex

Systematic Overlap Develops by Slowly Integrating over Experience

Effects of hippocampal damage in Amnesia

Effects of hippocampal damage in Amnesia

Complementary Learning Systems

<table>
<thead>
<tr>
<th>Neocortex</th>
<th>Hippocampus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn new facts (encode immediately)</td>
<td>Learn new memories (encode immediately)</td>
</tr>
<tr>
<td>Learn new responses (encode immediately)</td>
<td>Learn new concepts (encode immediately)</td>
</tr>
<tr>
<td>Slow learning</td>
<td>Fast learning</td>
</tr>
</tbody>
</table>

Effects of hippocampal damage in Amnesia

Amnesics show:

• spared implicit memory, skill learning (without recall)

• intact repetition priming for existing associations (table-chair)

• remote memories spared but recent ones completely forgotten

“Consolidation”: reactivation of memories across multiple contexts, sleep, etc.
Hippo = King of the Cortex

Hippo binds together multiple cortical representations into one coherent memory.

Hippocampal Anatomy

Pattern Separation

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 fire if all features they detect are present in the input
• Units represent conjunctions of features

Pattern Separation & Conjunctions

Pattern Separation & Conjunctions: Space and episodes

a) b)

How are 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

1,000,000 0.5

625 1.0

CA3

160,000 2.5

240 5.0

CA1

250,000 2.5

384 9.4

The Model

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

par complete with LTD supports pat sep; O'Reilly & McLelland

→ hippocampus to minimize this tradeoff (LTP in CA3 supports

keep pac complete to an existing memory).

How does your hippo know whether to store new memory and

LTD supports pat sep; O'Reilly & McLelland

- “You told me this already!”

- Which maintains original EC pattern?

Pattern completion in CA3 activates corresponding CA1 rep,

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

The Hippo Model of Separation: Pattern Completion

Sparse Activity

Sparse Activity
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 pattern separation at encoding
  - DG "steps aside" at retrieval
  - Evidence for two modes: theta cycle (e.g., Hasselmo et al., 2002); neuromodulatory control over DG effect on CA3

The Model

An example of how Modeling informs Science

The 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 an 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.

Possible Solutions: Two Modes
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 activity).

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

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

Monosynaptic route (EC → CA1) 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 pattern completion.

Transgenic mouse Nashashiba et al, 2008, Science (Neurogenesis in DG critical for behavioral pattern separation; discriminate between items spatially close together but not far apart).

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

- 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

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.

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

- Grid cells are in medial entorhinal cortex (Hafting et al, 2005), not hippocampus proper.
- Hippocampus 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-separated representations.
But first: Why does this happen?

Prefrontal Cortex: Delay-related activity

Activation-based Priming: Residual Activation

Remember Weight-Based Priming?

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

Spatial delayed-response task; Funahashi et al, 1989

But what about when need to maintain over longer delays?

Activation-based Priming: Residual activation

Outp_dist

Both_err

ev_nm

Outp_dist

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

Both_err

But first: Why does this happen?

Prefrontal Cortex: Delay-related activity

But what about when need to maintain over longer delays?

Activation-based Priming: Residual activation

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

Activation-based Priming

Remember Weight-Based Priming?

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
bottle
baby
diapers

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, 1993; Pucak et al, 1996)

Exploration: [active maint.proj]
synthesizer
television
terminal
keyboard
speaker
monitor

Input
Monitor Speakers Keyboard

Weight- and Activation-Based Memory Interactions
A-not-B task
Perseverative searching at A – also seen in patients with PFC damage
Better performance 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

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.

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 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

(a) Update
- My phone # is 371−9624

(b) Maintain
- My phone # is 371−9624

- 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.

Abstract: 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.

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 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

(a) Update
- My phone # is 371−9624

(b) Maintain
- My phone # is 371−9624

- 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

Sensory Input

Working Memory Gating

a) Update

Should I Stay or Should I Go...

b) Maintain

Jenny, I got your #... My phone # is 371-9624... 371-9624... 867-5309...

... 371-9624... My phone # is 371-9624

• 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??

DA solves part of the problem

• Learning signal for gating.

• DA is very global signal projecting to all of PFC — sufficient for updating and maintaining one item at a time.

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

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

• Learning signal for gating.

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 signals are important for learning/knowing when to gate

Dryamine: Cause & Current Story

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 up dating of PFC states!

DA solves part of the problem

• Learning signal for gating.

• But DA is very global signal projecting to all of PFC — sufficient for updating and maintaining one item at a time.

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

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

• Learning signal for 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...

• DA signals are important for learning/knowing when to gate

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 up dating of PFC states!

DA solves part of the problem

• Learning signal for gating.

• But DA is very global signal projecting to all of PFC — sufficient for updating and maintaining one item at a time.

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

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

• Learning signal for 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...

• DA signals are important for learning/knowing when to gate

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 up dating of PFC states!

DA solves part of the problem

• Learning signal for gating.

• But DA is very global signal projecting to all of PFC — sufficient for updating and maintaining one item at a time.

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

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

• Learning signal for 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...

• DA signals are important for learning/knowing when to gate

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 up dating of PFC states!
Reinforcement learning and WM gating

- Network learns to associate stimuli with rewards via DA system
- DA 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

- Burst and dips of DA train the basal ganglia Go/NoGo gating system: DA burst → Go; DA dip → NoGo
- 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 more likely that the BG will gate this pattern into PFC on future trials. (ie 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 Go learning in BG 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 learning to access the state