Sequential & Temporally-Delayed Learning

1. The Problem.

2. Sequential Learning & Context.

3. Temporally-delayed Learning & Reinforcement.

The Problem

Error-driven + Hebbian: Solve tasks, learn systematic representations, generalize to new stimuli.

What's left?...

The Problem

Error-driven + Hebbian: Solve tasks, learn systematic representations, generalize to new stimuli.
What's left?...

Time!

The Problem

Error-driven + Hebbian: Solve tasks, learn systematic representations, generalize to new stimuli.
What's left?...

Time!

Currently: networks learn *immediate* consequence of a given input.

The Problem

Error-driven + Hebbian: Solve tasks, learn systematic representations, generalize to new stimuli.
What's left?...

Time!

Currently: networks learn *immediate* consequence of a given input.

- What if current input only makes sense as part of a *sequence* of inputs (e.g., language, social interactions)?

The Problem

Error-driven + Hebbian: Solve tasks, learn systematic representations, generalize to new stimuli.
What's left?...

Time!

Currently: networks learn *immediate* consequence of a given input.

- What if current input only makes sense as part of a *sequence* of inputs (e.g., language, social interactions)?
- What if the consequence of this input comes *later* (e.g., school/work, life)?

Sequence Learning

How do we do it?

For example:

Sequence Learning

How do we do it?

For example:

My favorite color is purple.

Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.

Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.

Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Is purple my color favorite.

Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Is purple my color favorite.

The girl picked up the pen.

Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Is purple my color favorite.

The girl picked up the pen.
The pig raced around the pen.

Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Is purple my color favorite.

The girl picked up the pen.
The pig raced around the pen.

We represent the *context,* not just the current input.

Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Is purple my color favorite.

The girl picked up the pen.
The pig raced around the pen.

We represent the *context*, not just the current input.

in language, social interactions, driving (who goes at a 4-way stop?)

Representing Context for Sequence Learning

How does the brain do it?

How would we get our models to do it?

Representing Context for Sequence Learning

How does the brain do it?

How would we get our models to do it?

Add layers to keep track of context (prefrontal cortex; hippocampus...).

An Example Task

BTXSE

BPVPSE

BTSXXTVVE

BPTVPSE

An Example Task

BTXSE
BPVPSE
BTSXXTVVE
BPTVPSE

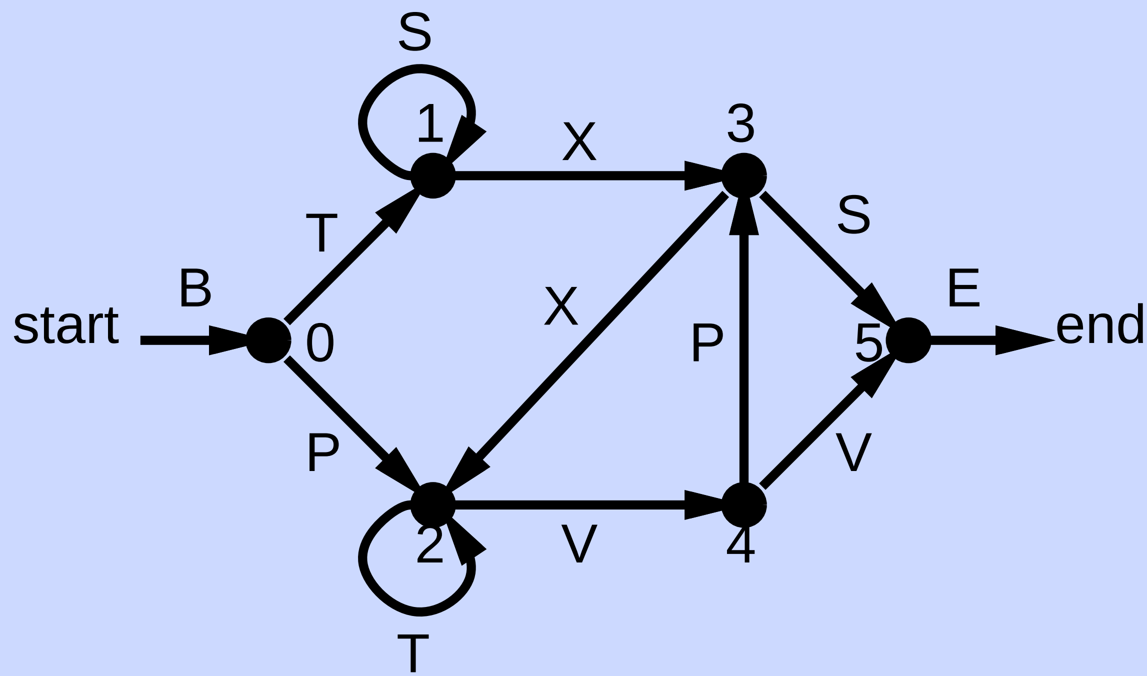
Which of the following sequences are allowed?:
BTXXTTVVE
TSXSE
VVSXE
BSSXSE

An Example Task

BTXSE, BPVPSE, BTSXXTVVE, BPTVPSE, BTXXTTVVE
TSXSE, VVSXE, BSSXSE

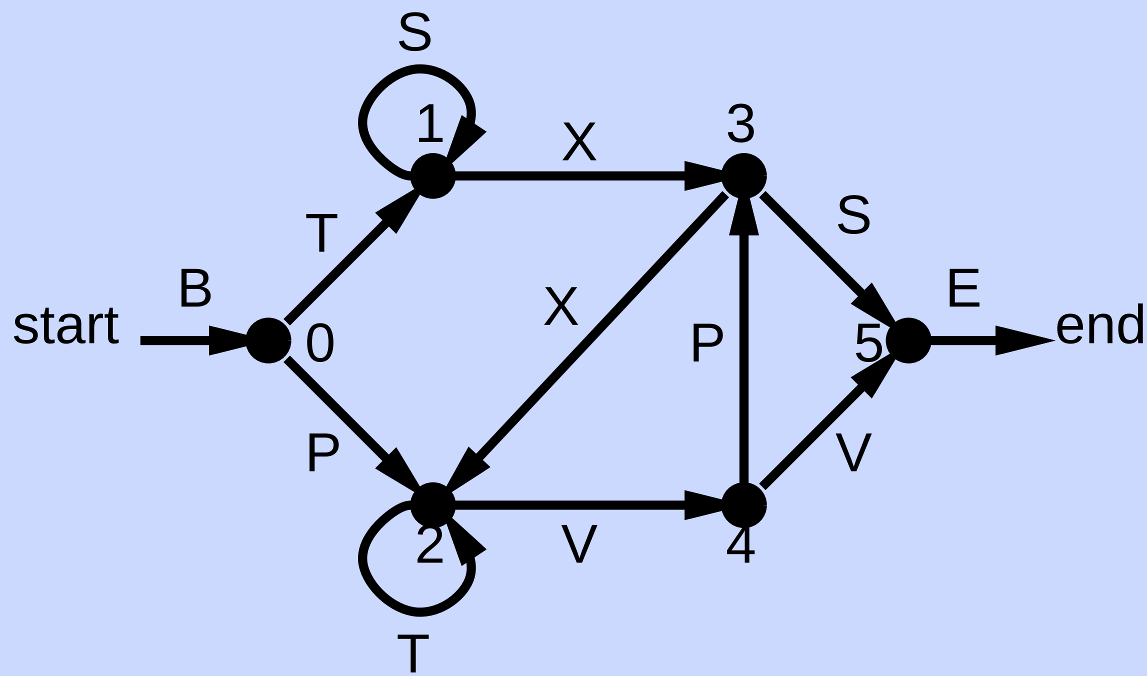
An Example Task

BTXSE, BPVPSE, BTSXXTVVE, BPTVPSE, BTXXTTVVE
TSXSE, VVSXE, BSSXSE



An Example Task

BTXSE, BPVPSE, BTSXXTVVE, BPTVPSE, BTXXTTVVE
TSXSE, VVSXE, BSSXSE



We implicitly learn such grammars (e.g., pressing buttons faster to letters that follow grammar).

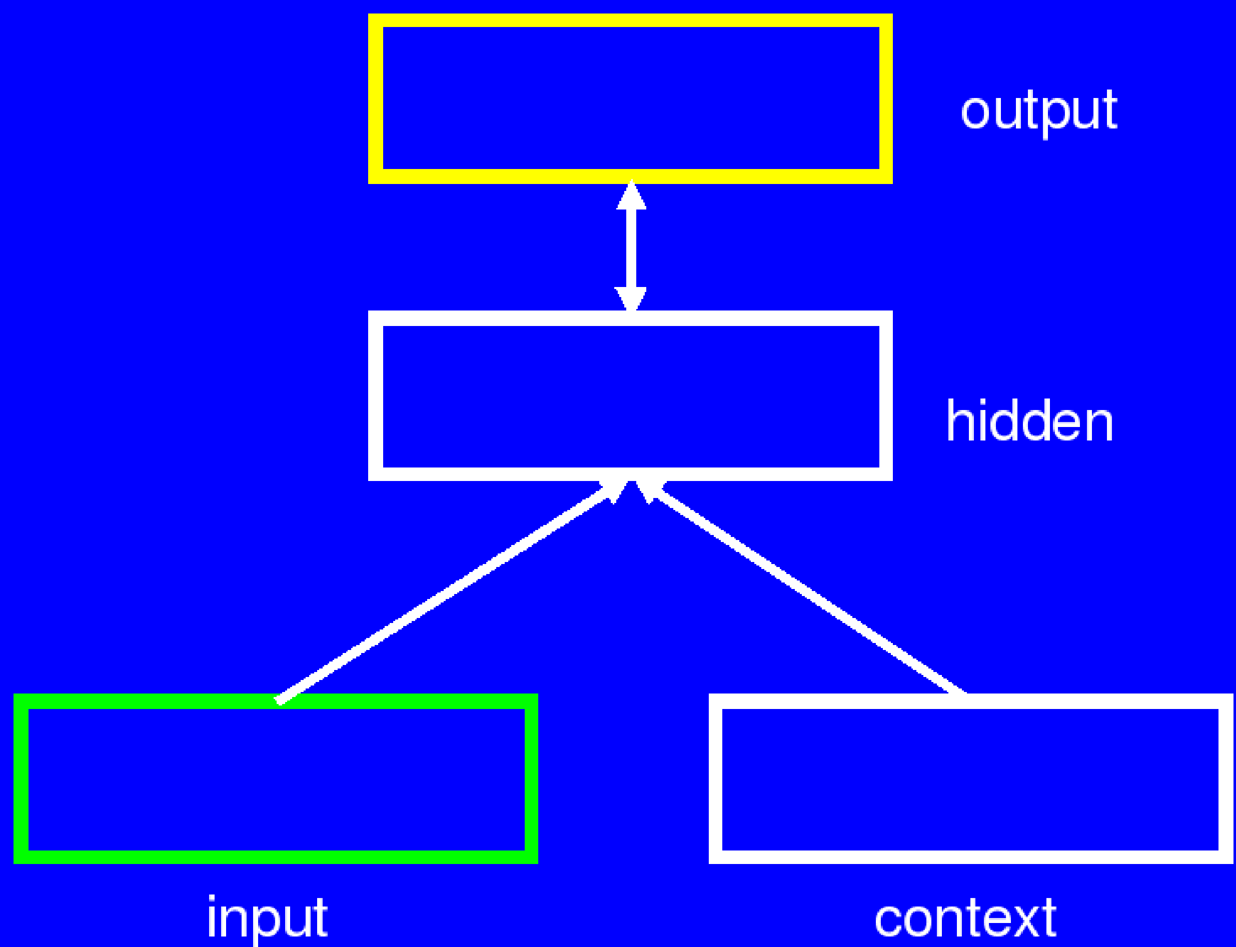
Time & Sequences

Currently: networks learn *immediate* consequence of a given input.

What if current input only makes sense as part of a temporally- extended *sequence* of inputs? (*context*)

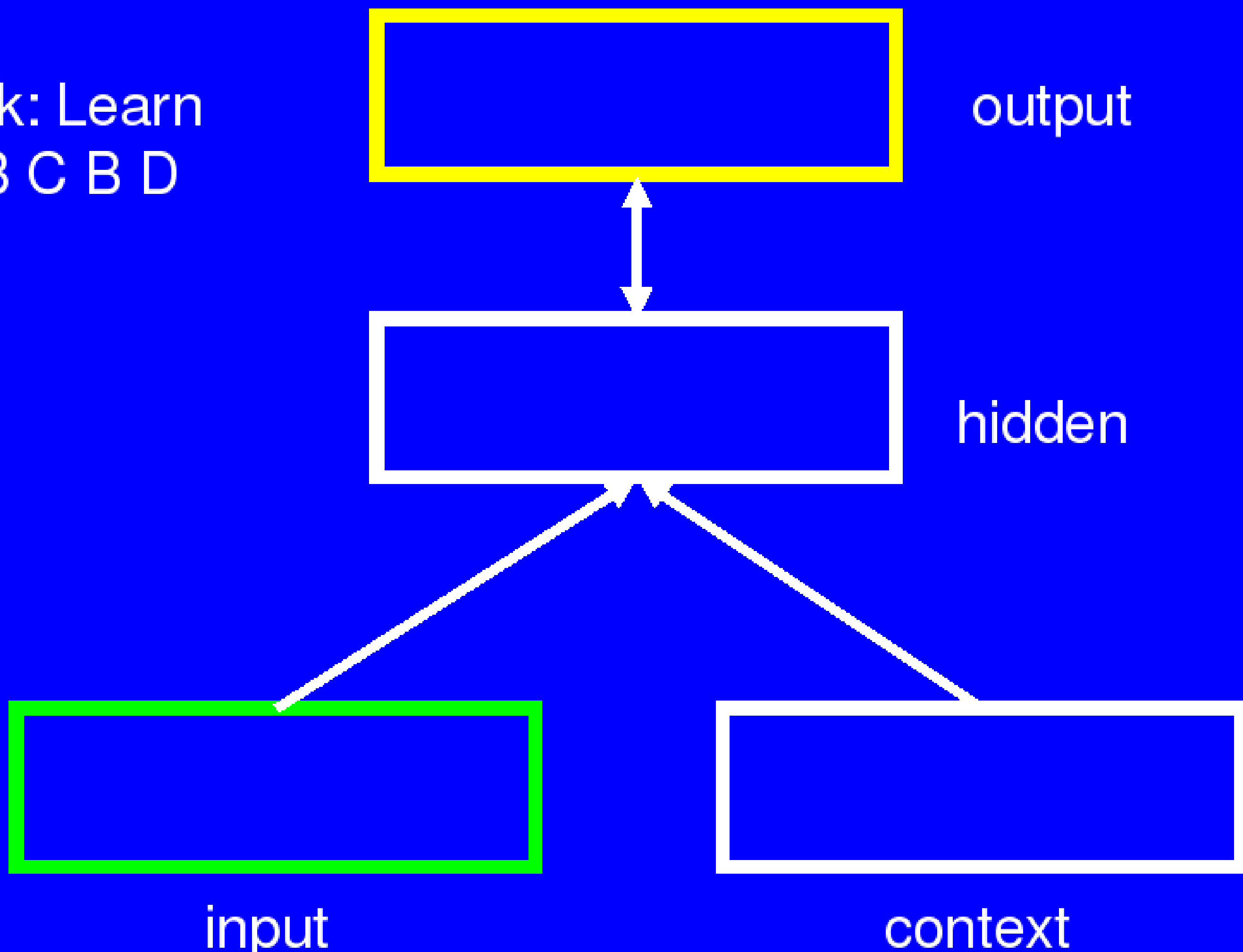
What if the consequence of this input comes *later* in time? (next week)

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



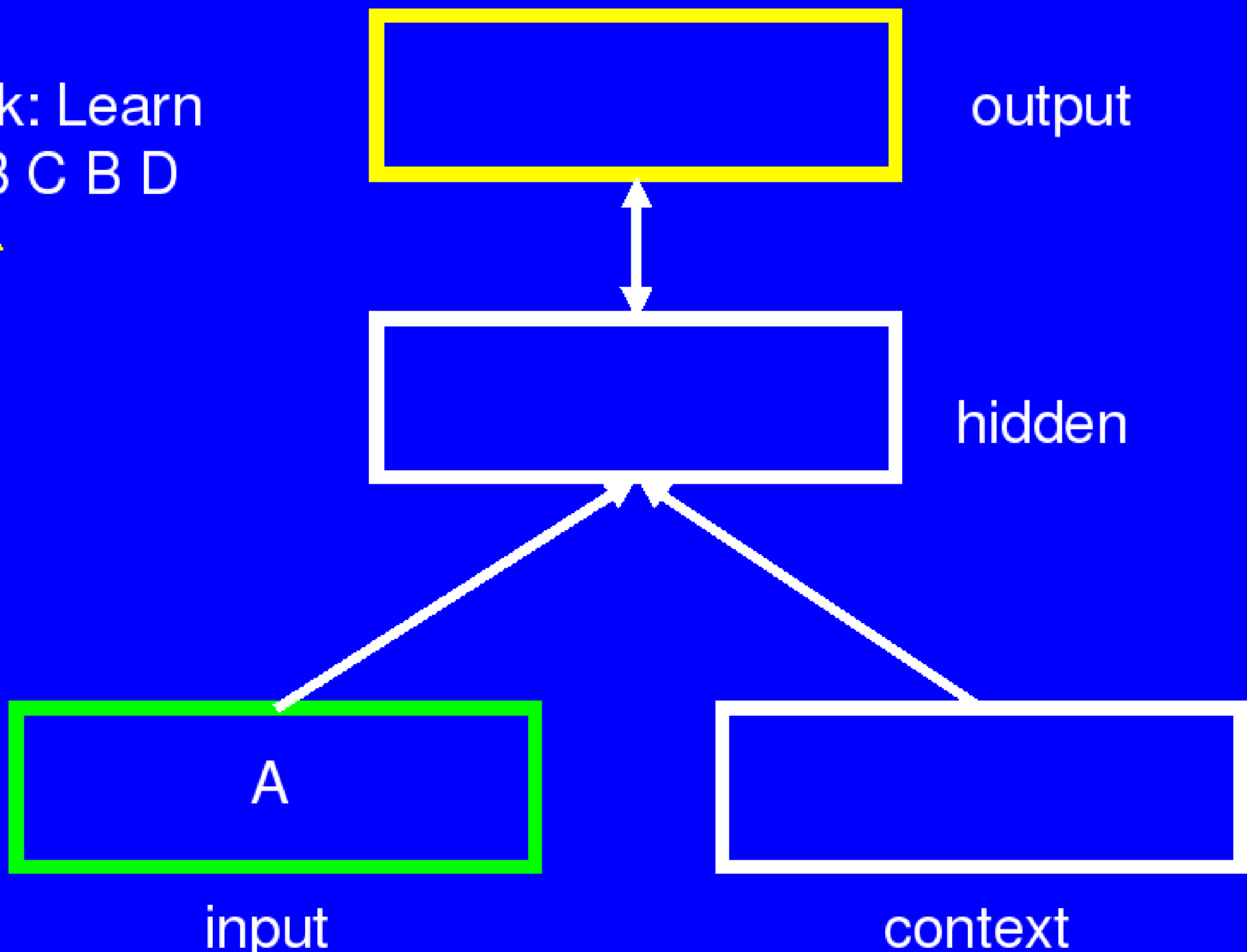
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



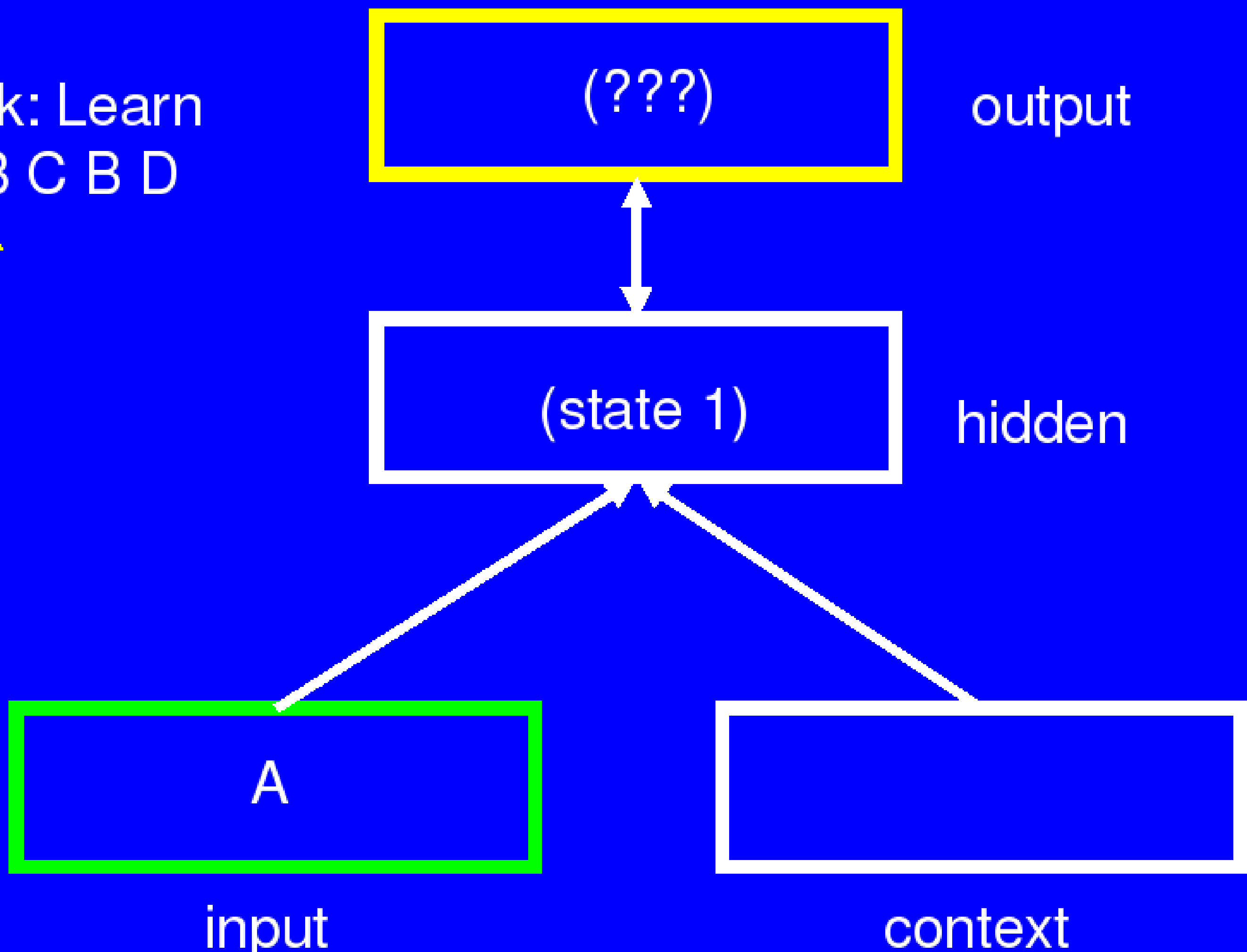
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D

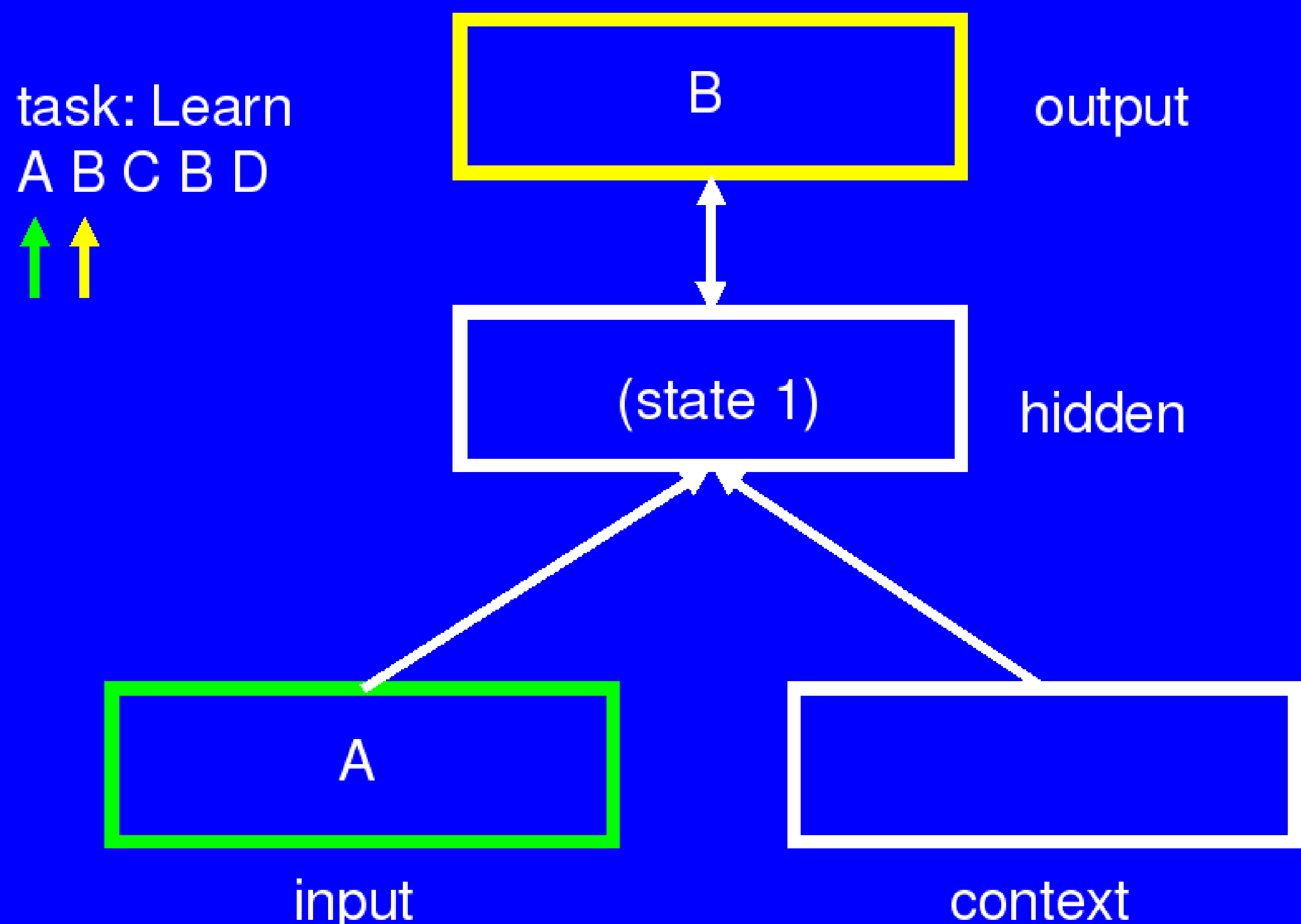


Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D

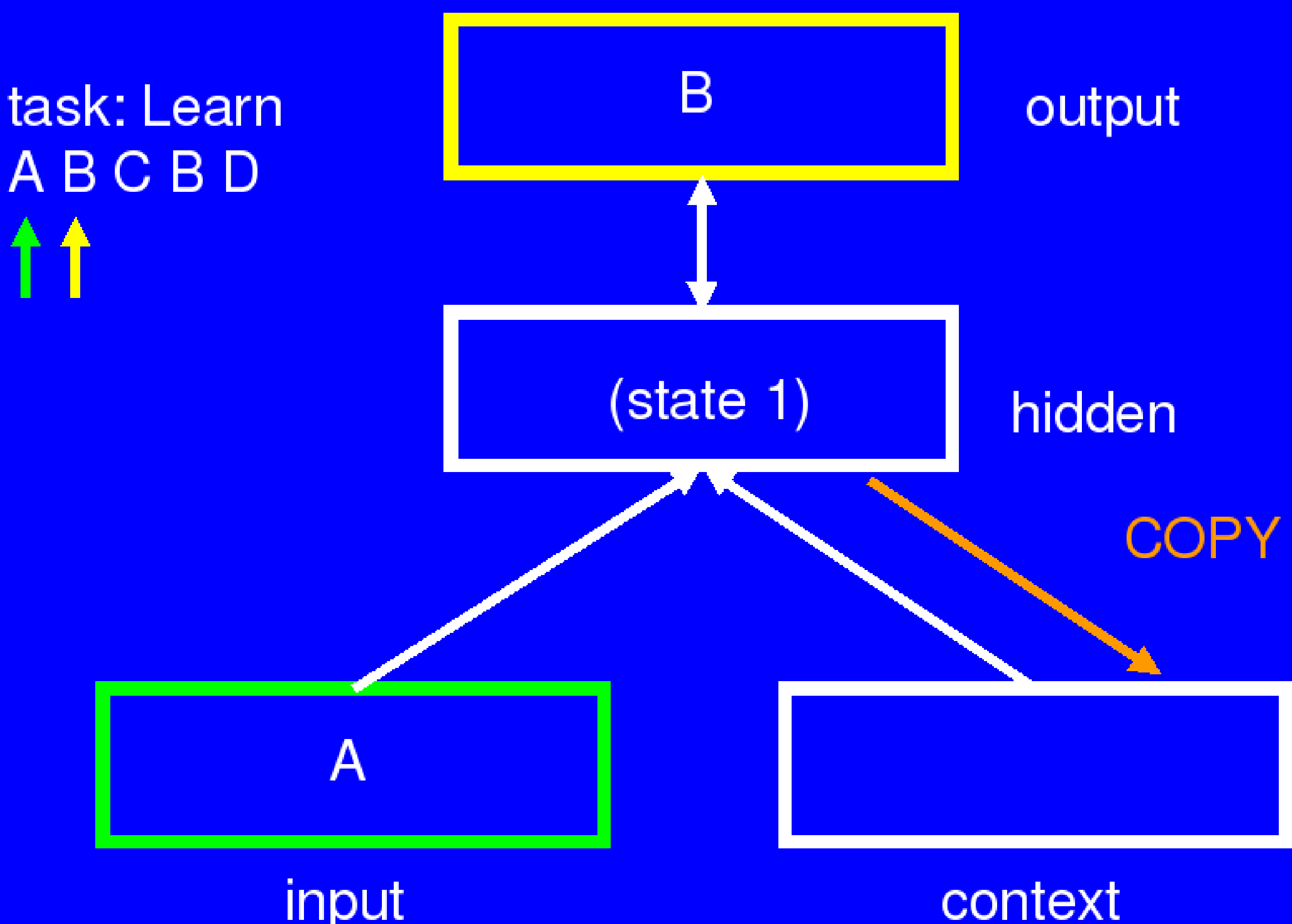


Simple Recurrent Network (SRN): An Architecture for Sequence Learning



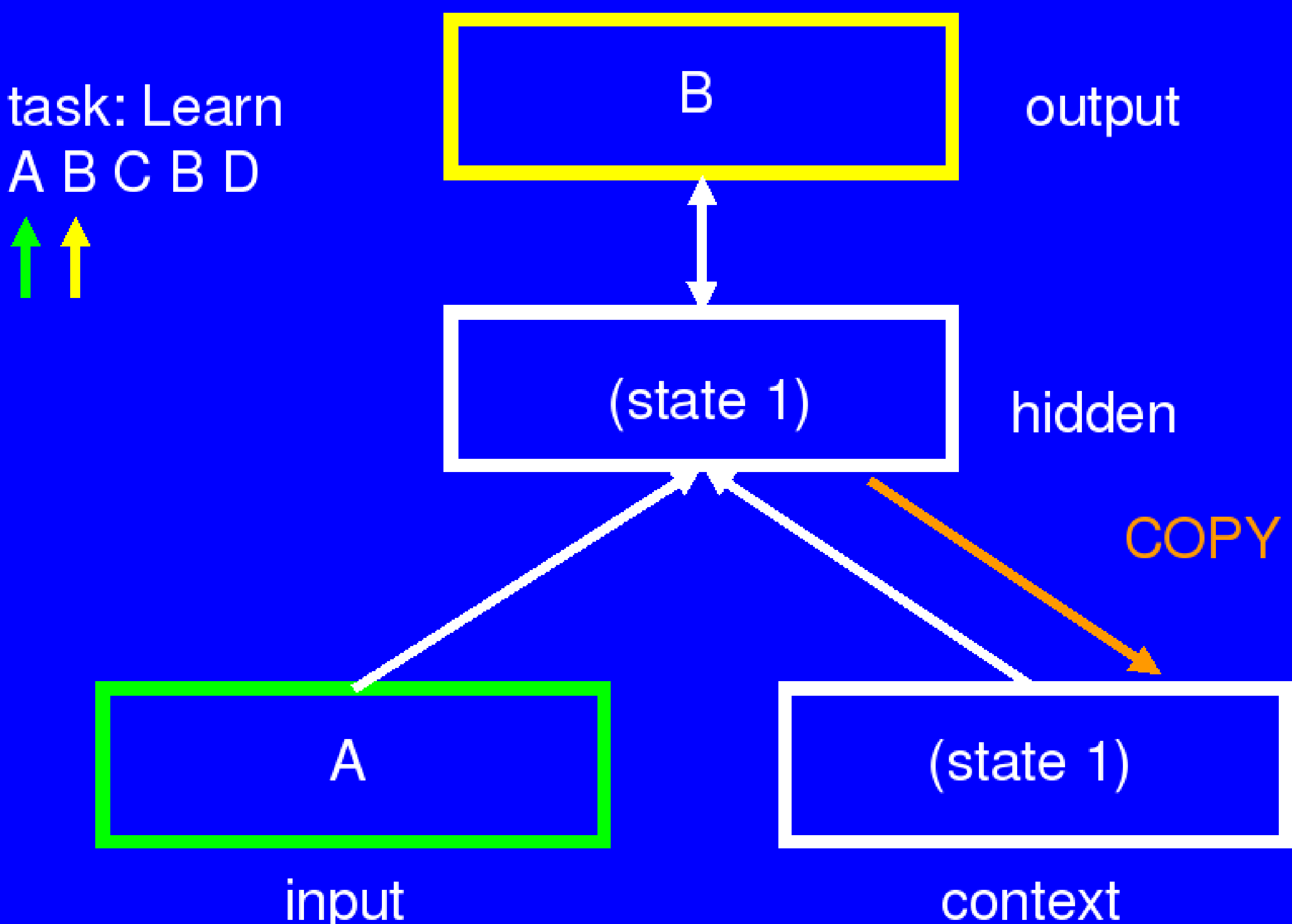
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



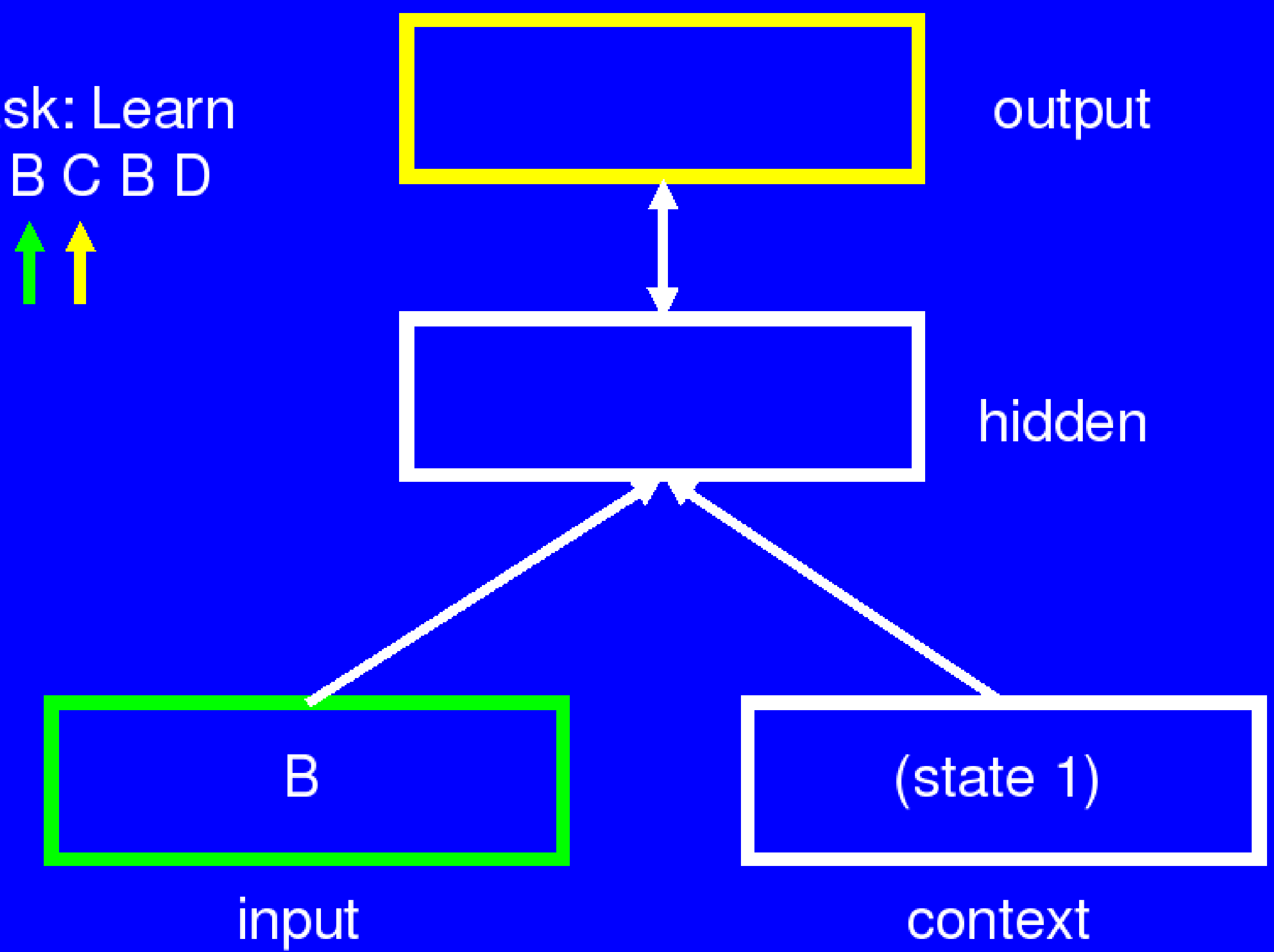
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D

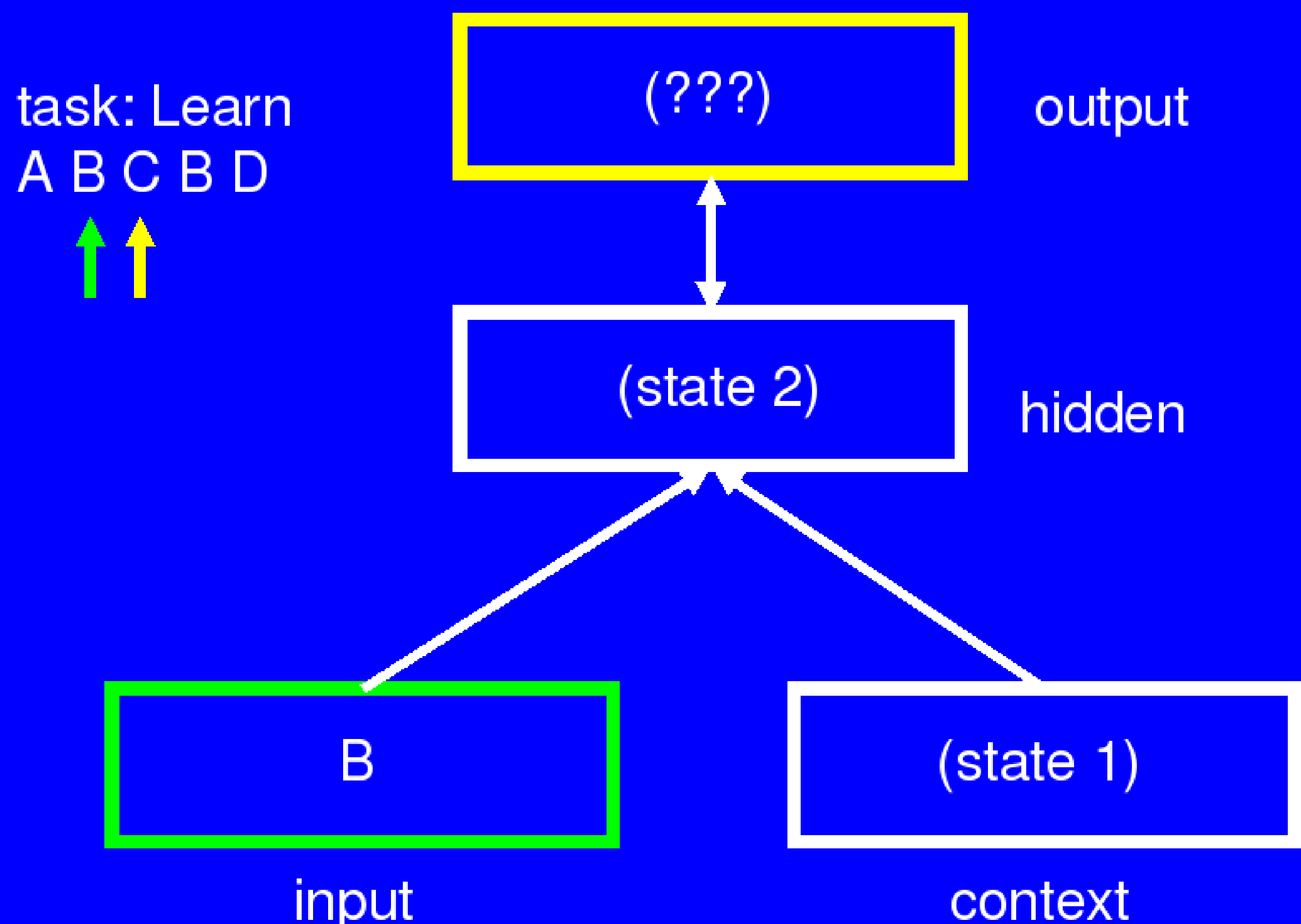


Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D

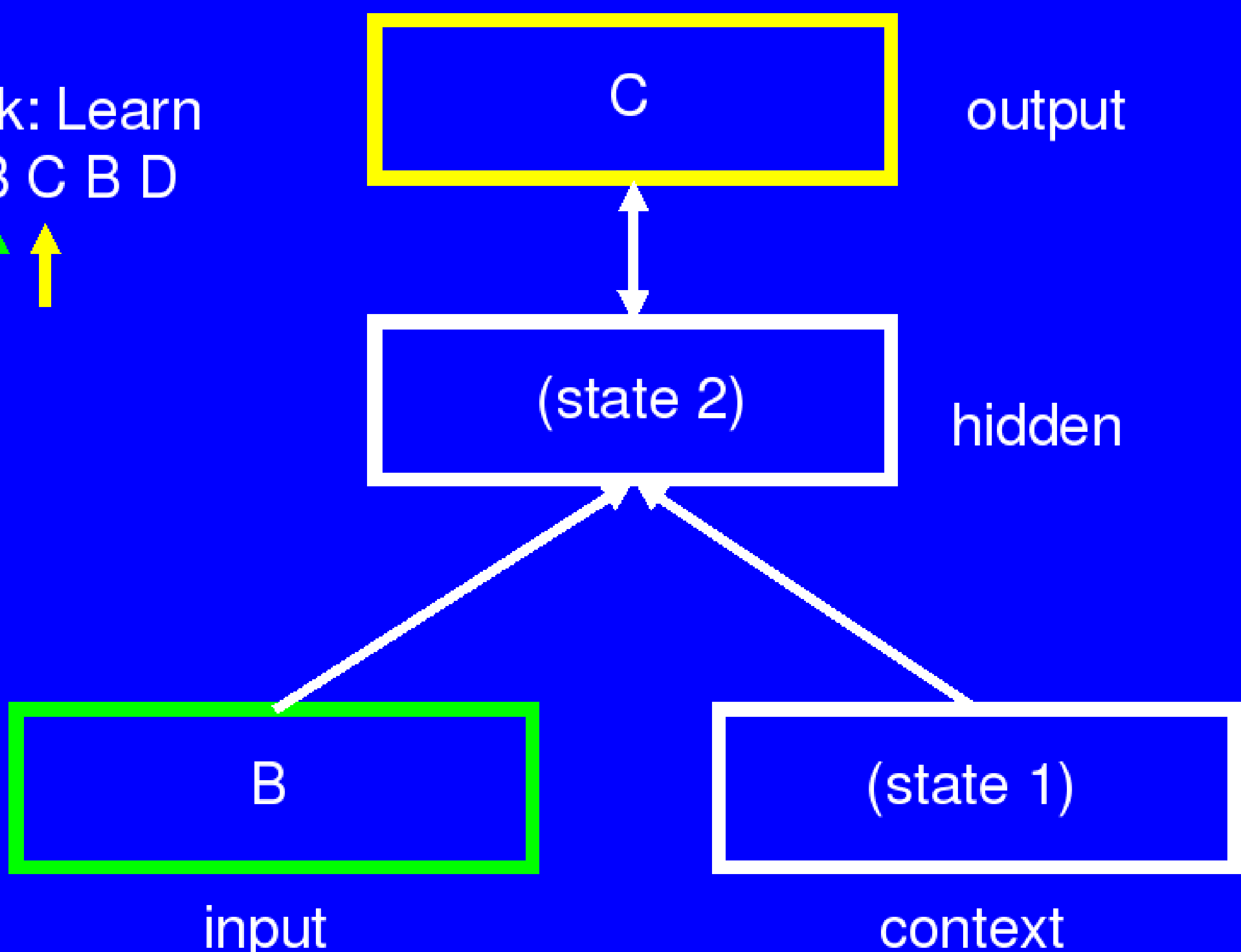


Simple Recurrent Network (SRN): An Architecture for Sequence Learning



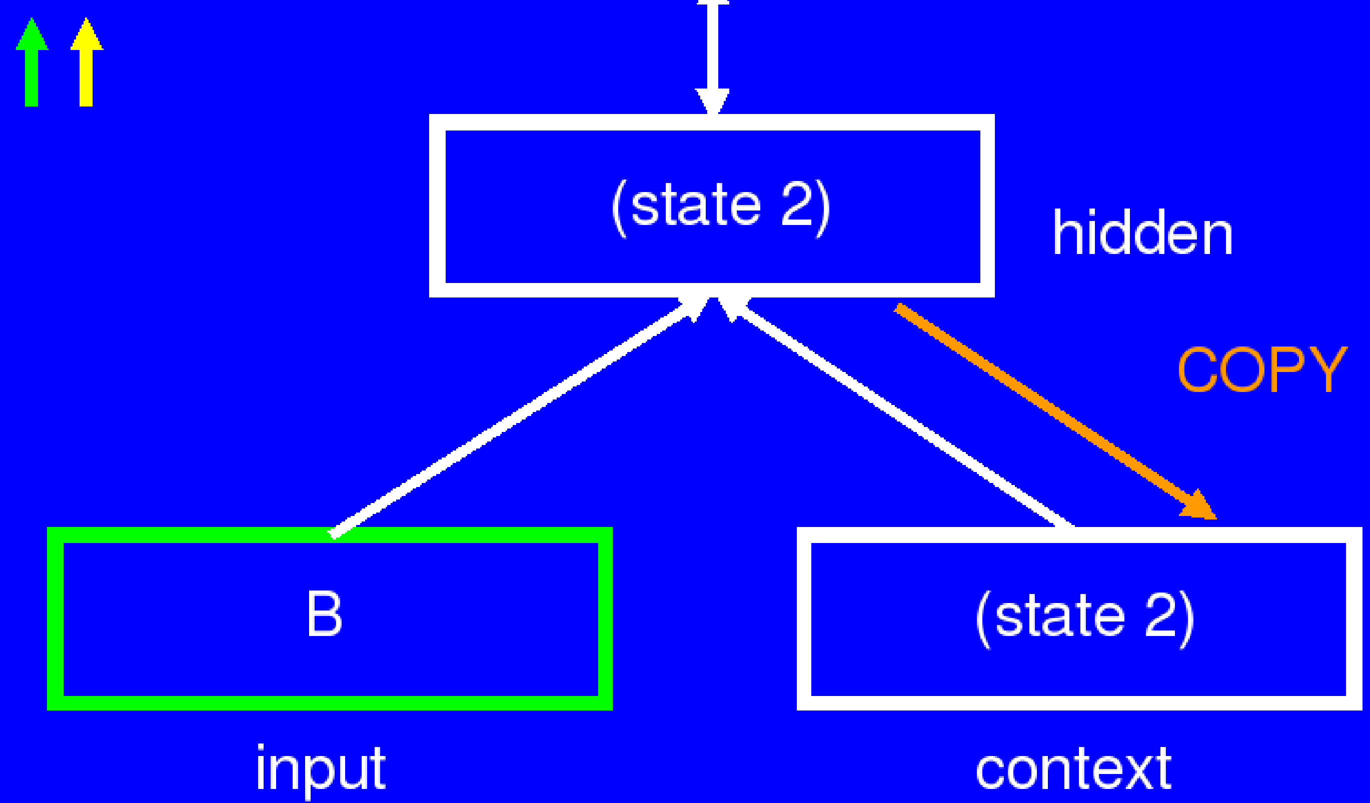
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



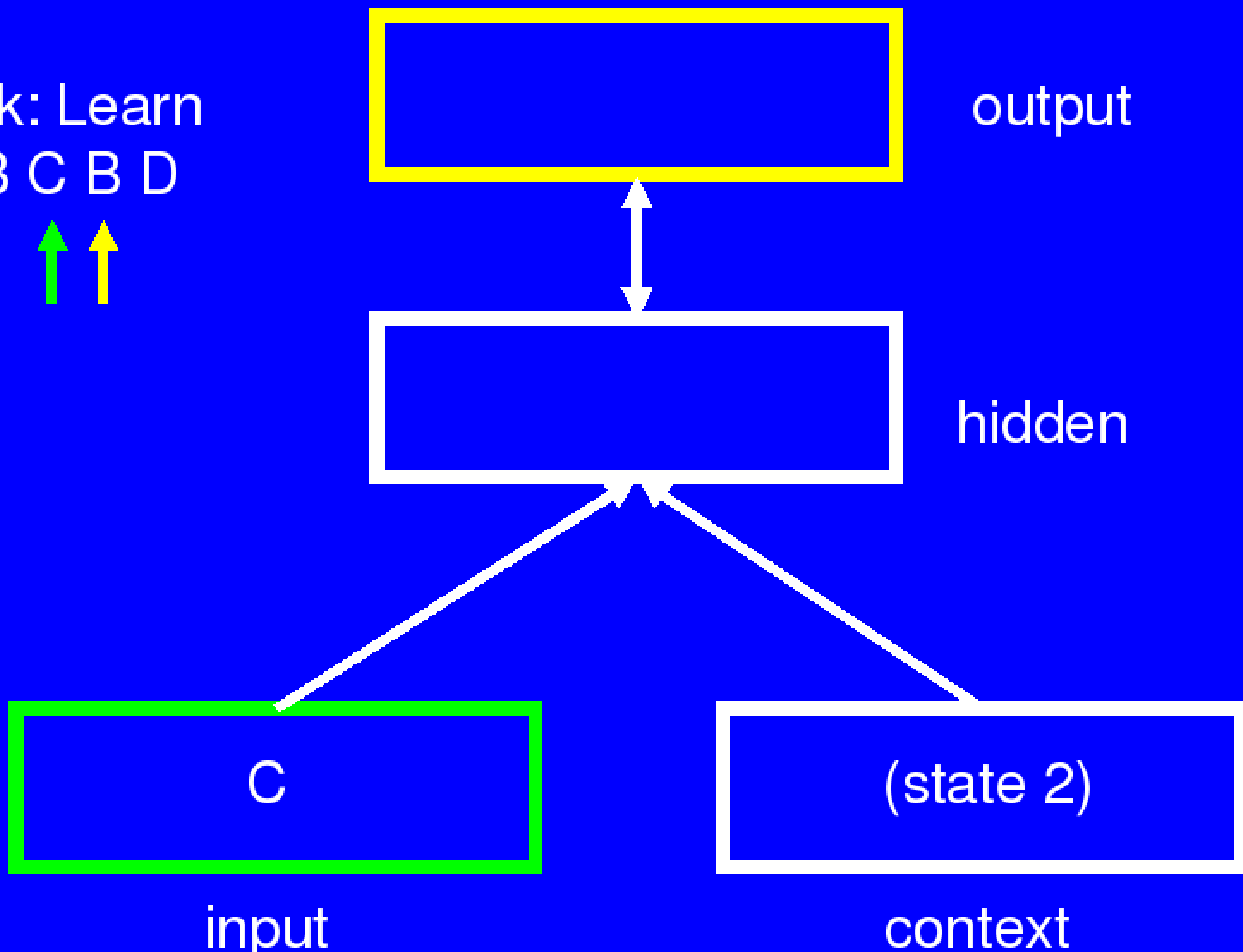
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



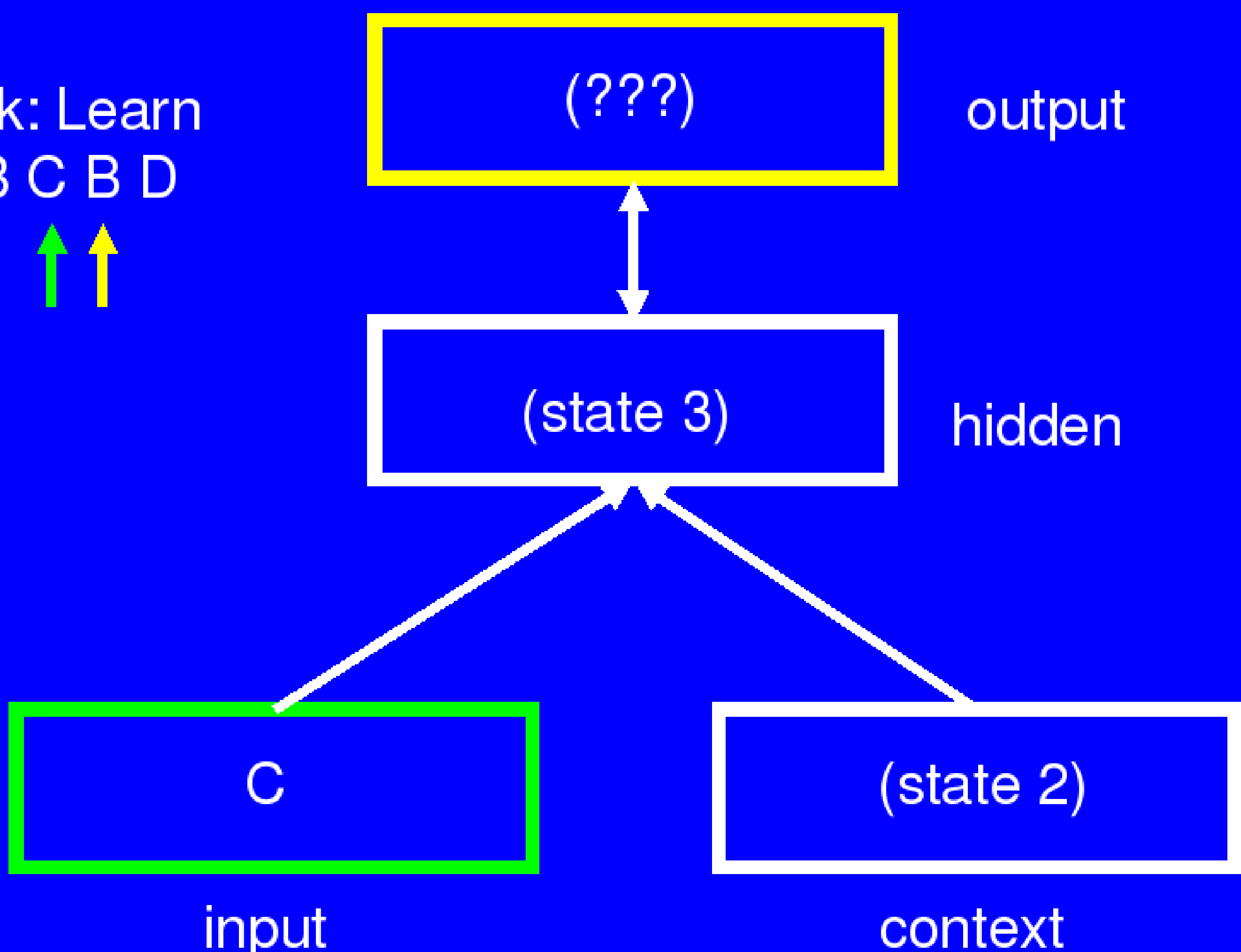
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



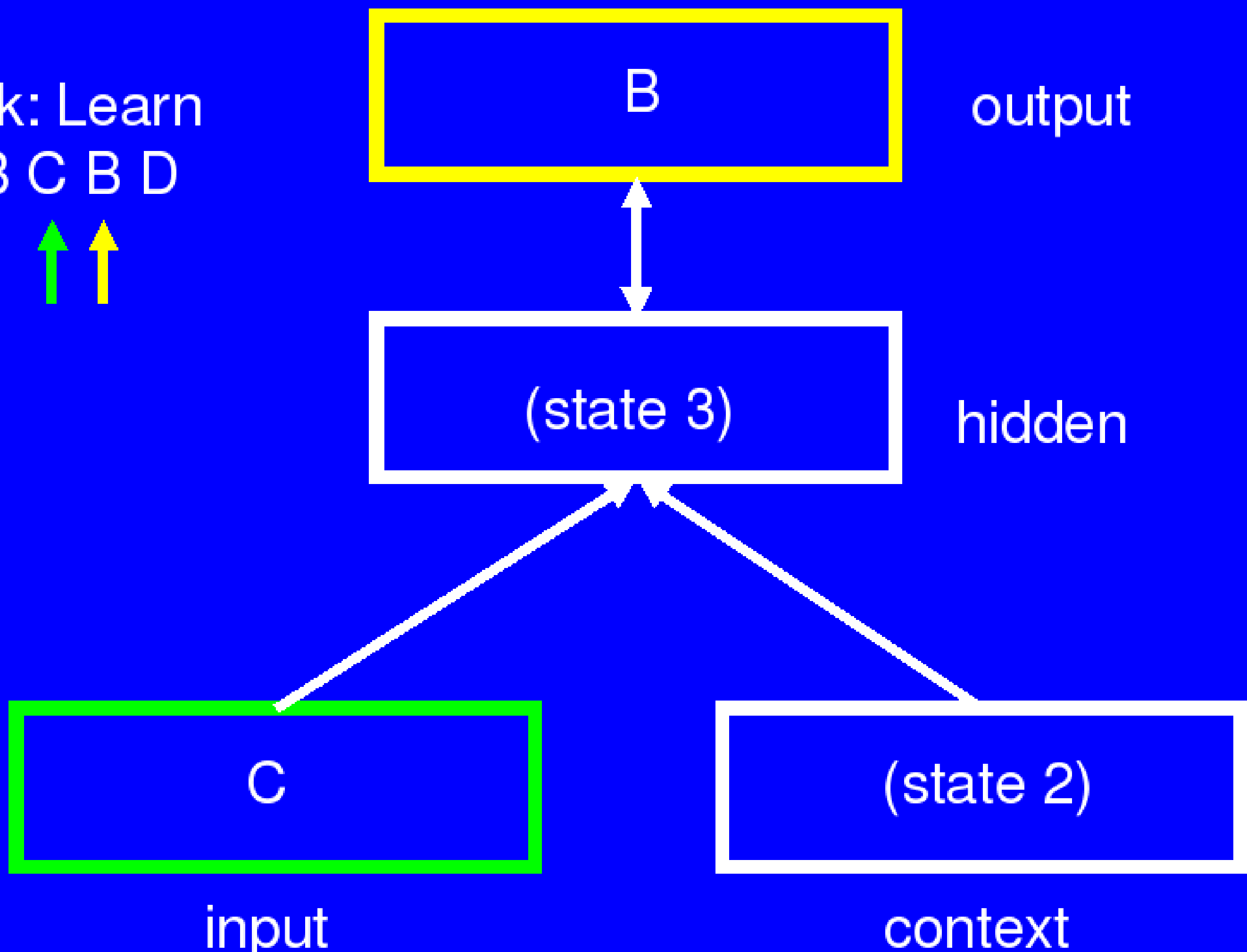
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



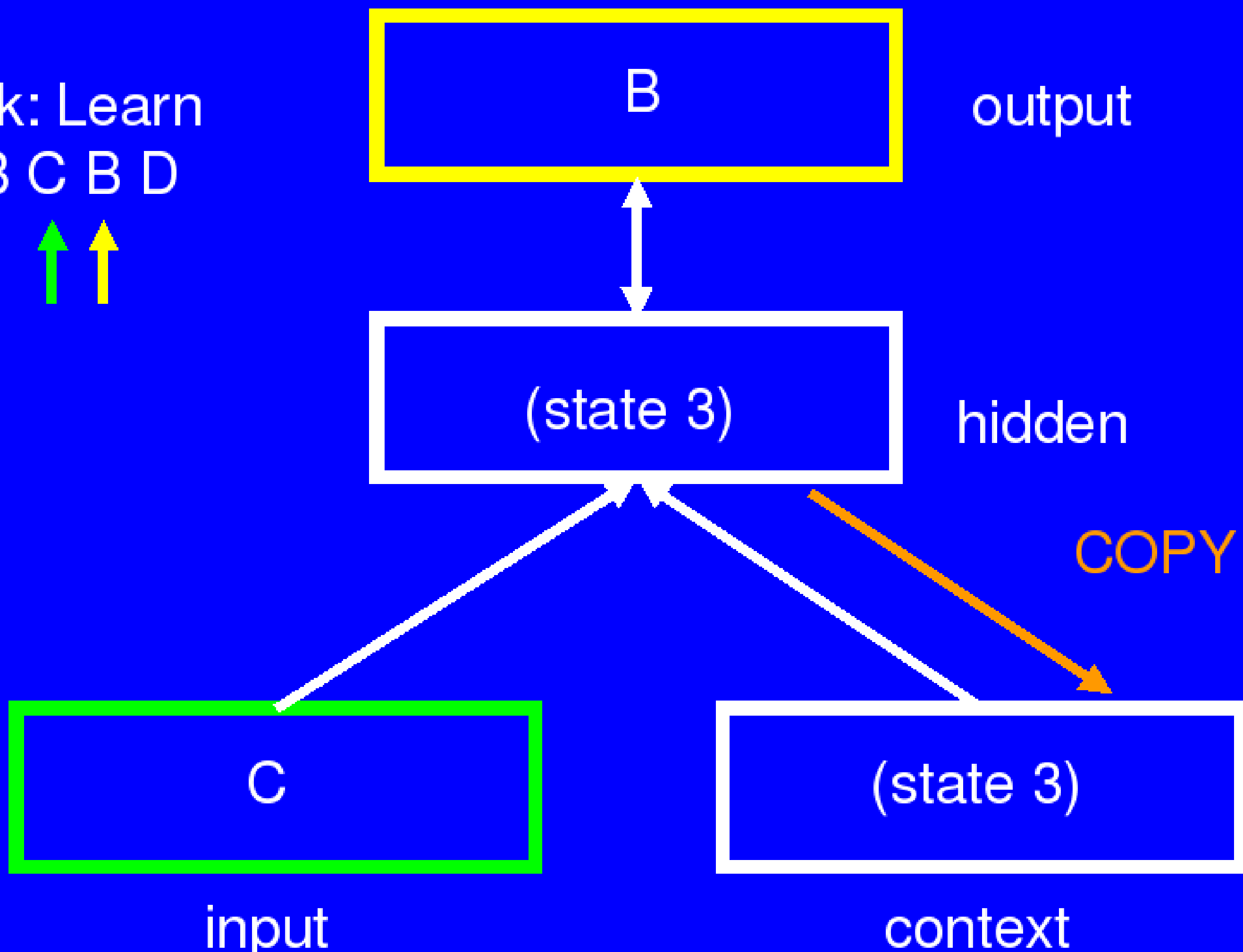
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



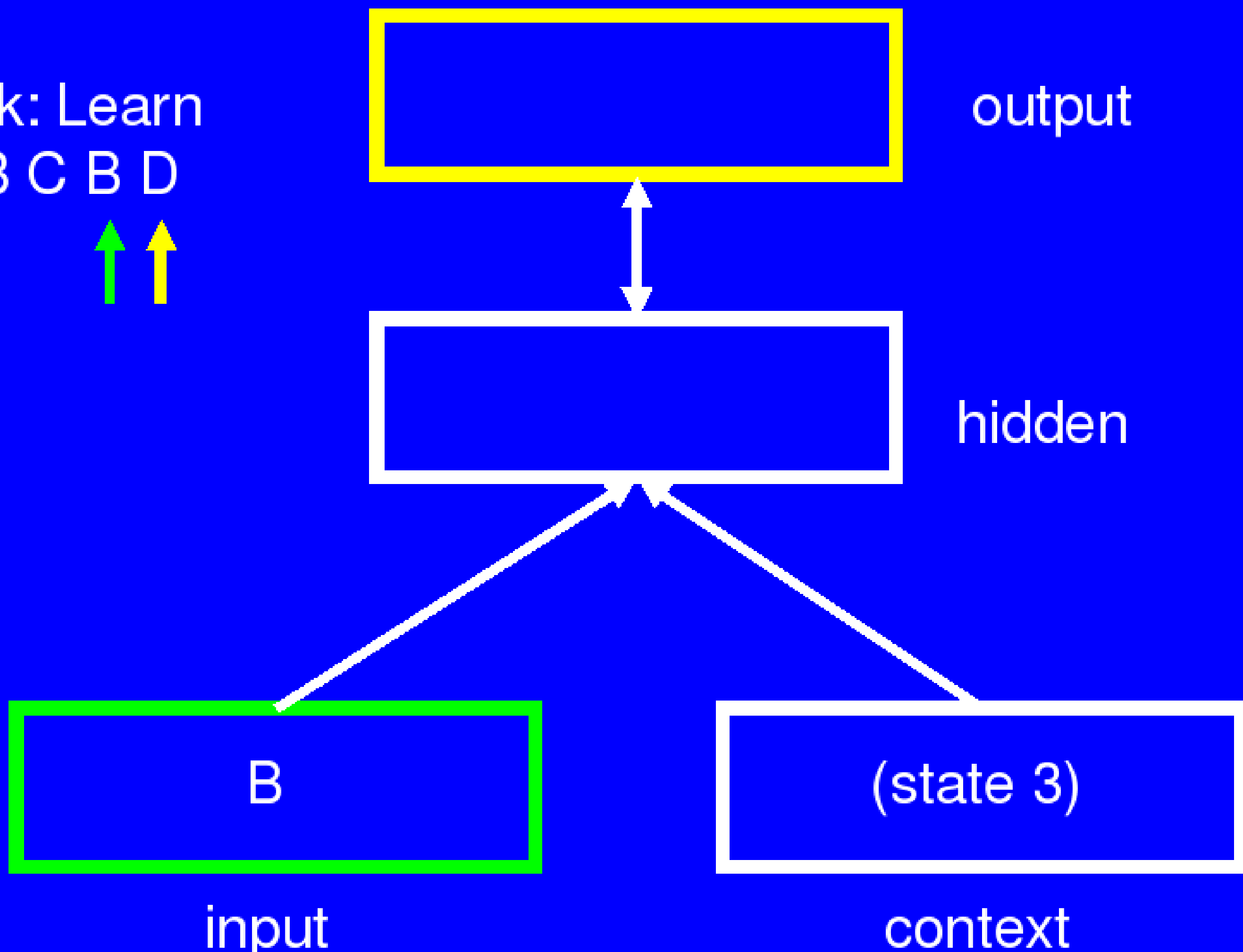
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



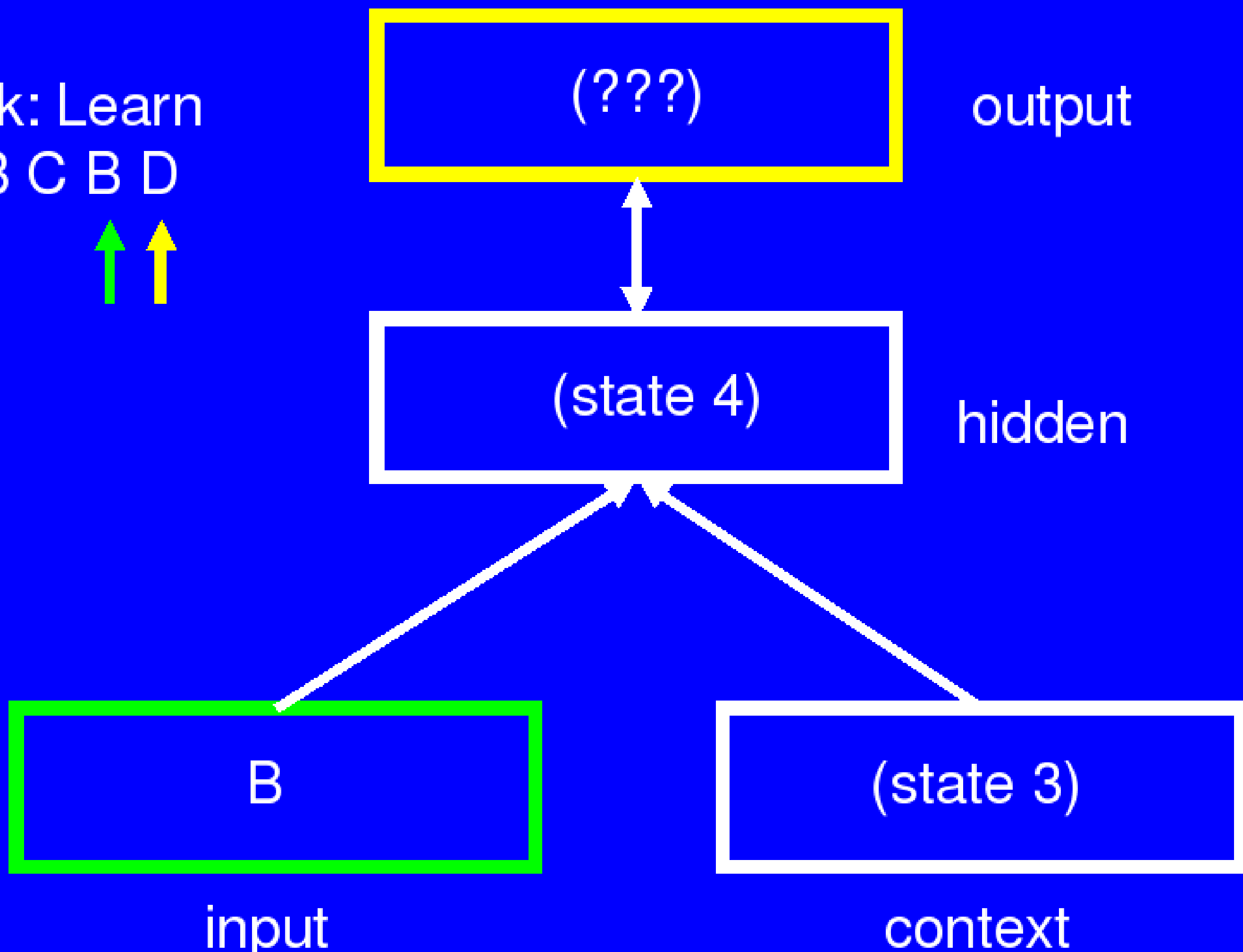
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



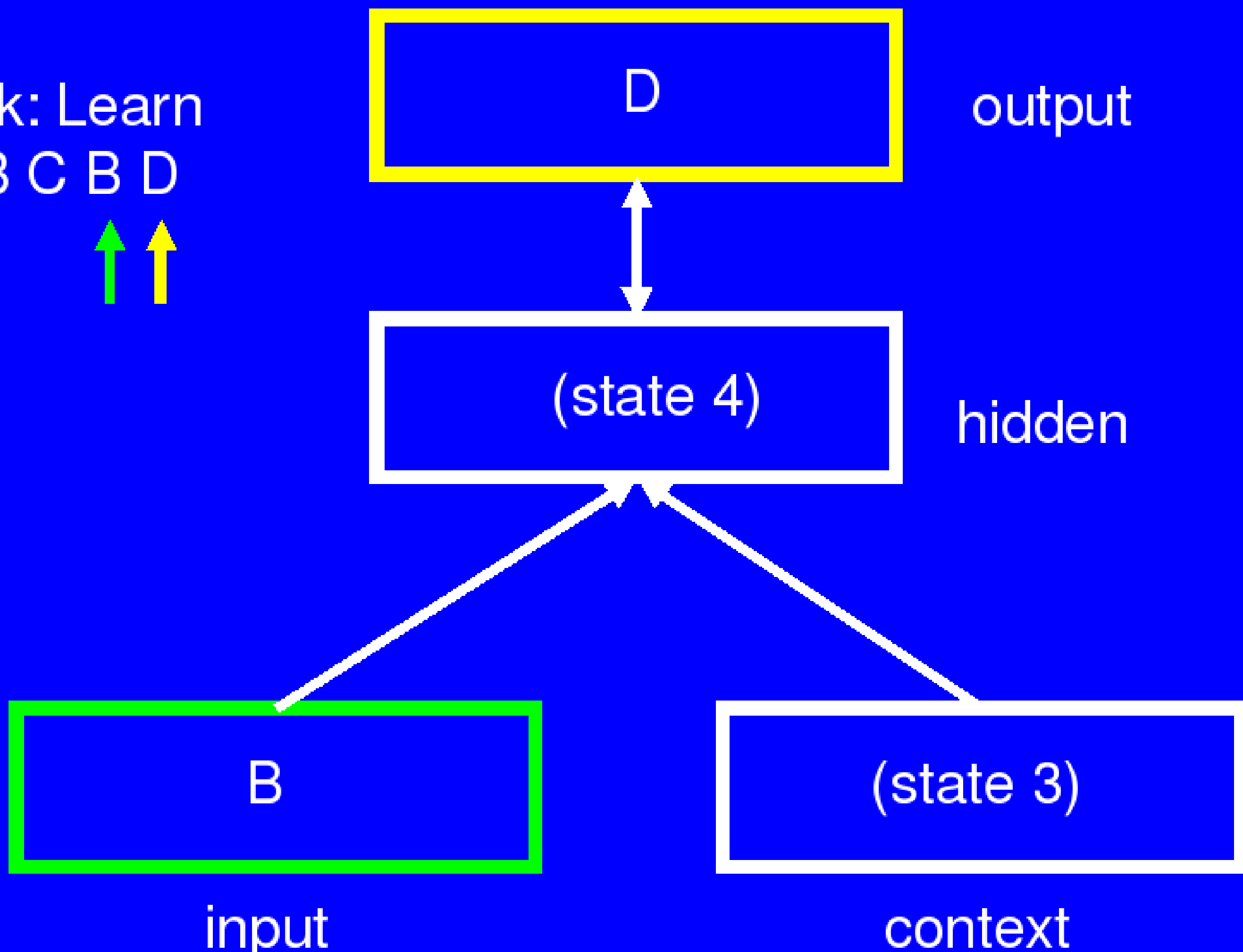
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



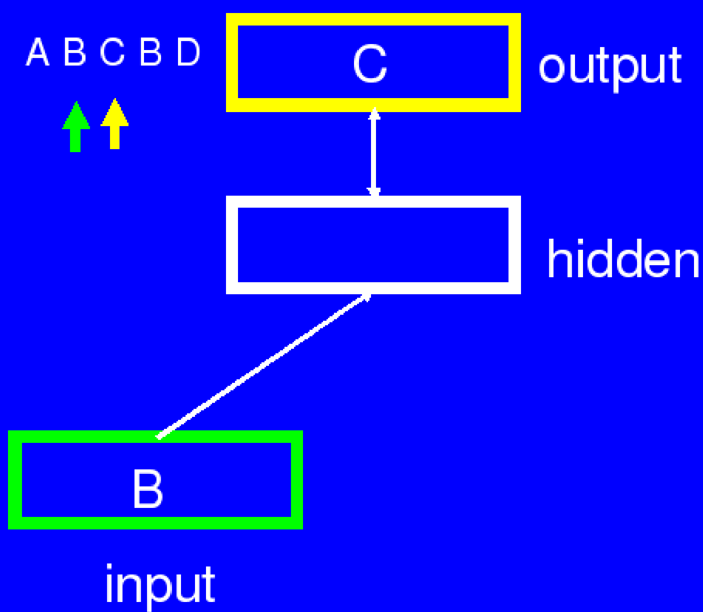
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

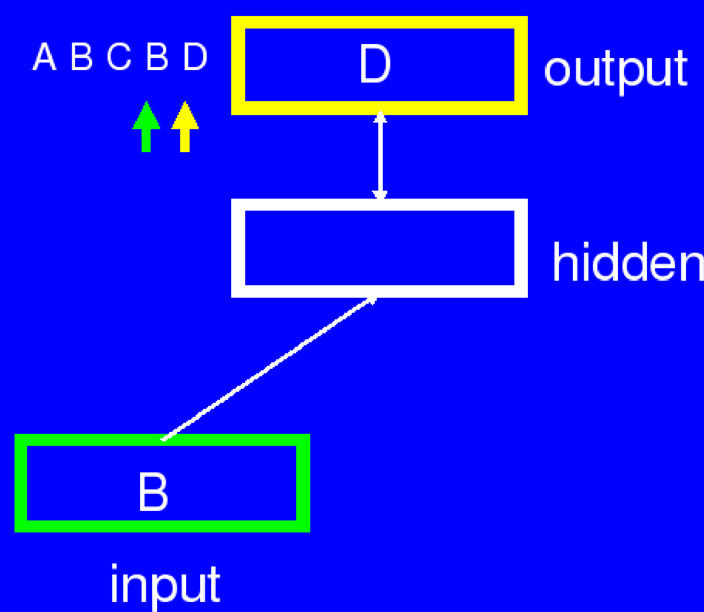
task: Learn
A B C B D



Simple Recurrent Network: Summary

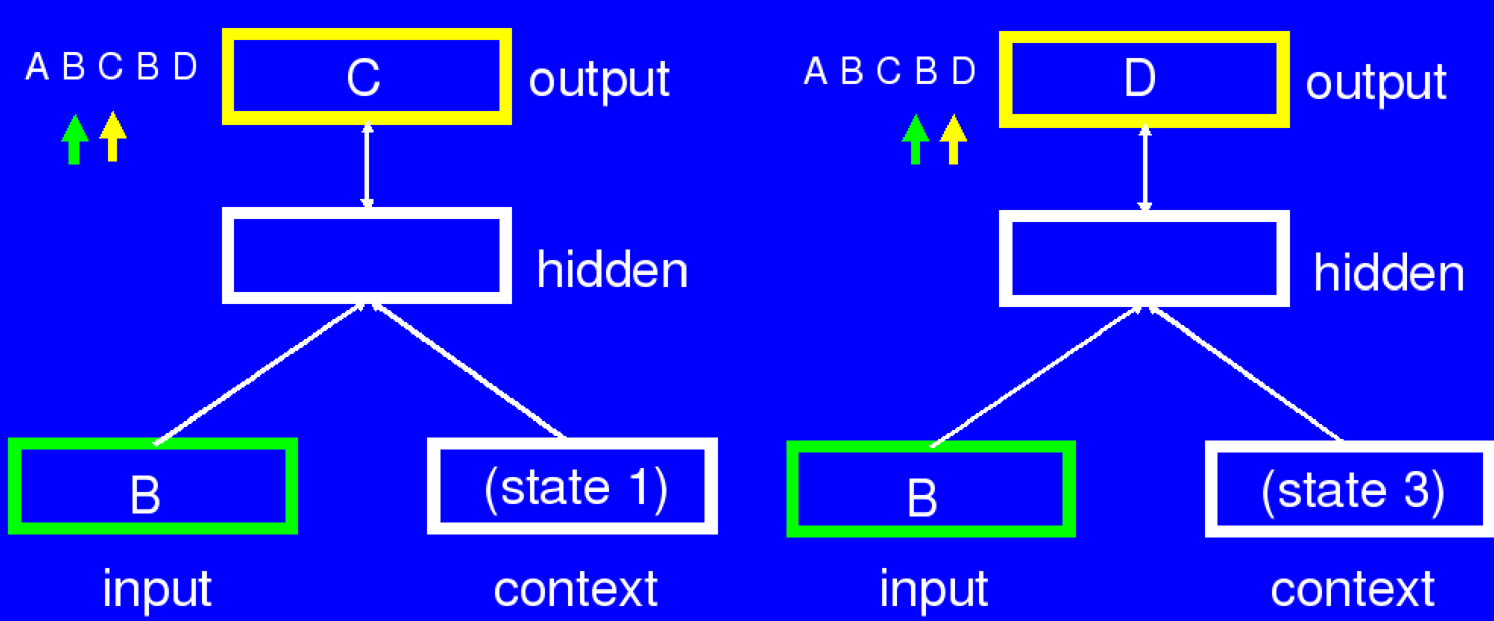
- Carries forward information by means of a **context** layer that contains the hidden representation from the previous time step.
- This hidden representation serves to disambiguate the input





Simple Recurrent Network: Summary

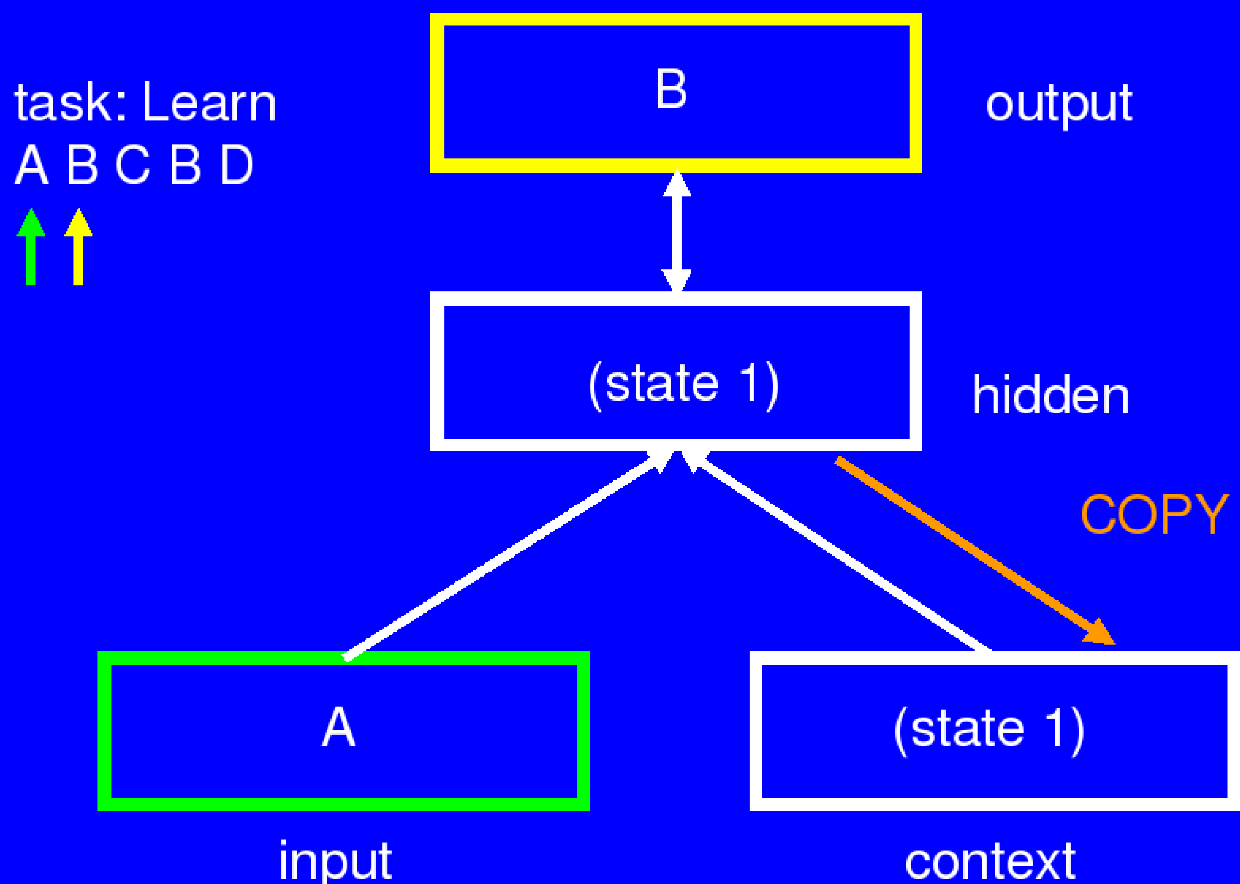
- Carries forward information by means of a **context** layer that contains the hidden representation from the previous time step.
- This hidden representation serves to disambiguate the input



Why Copy the Hidden Representation?

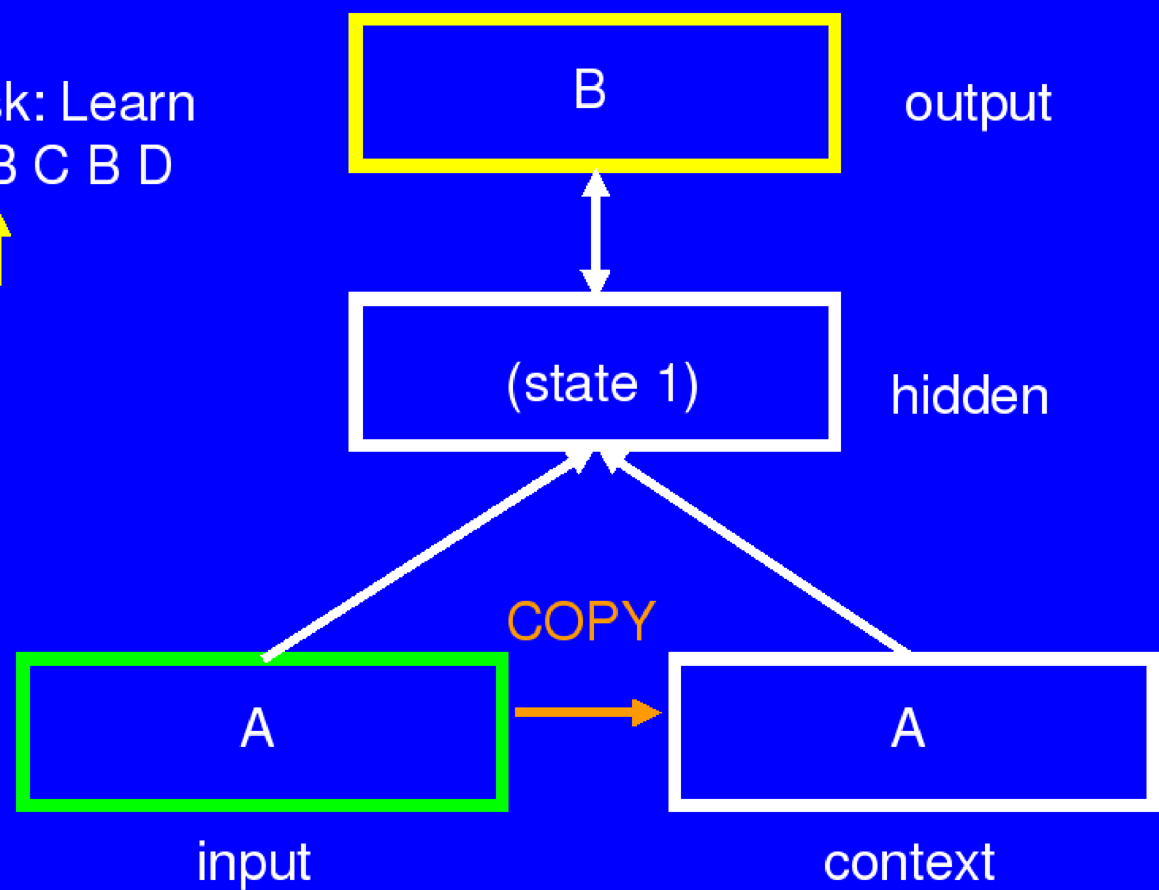
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



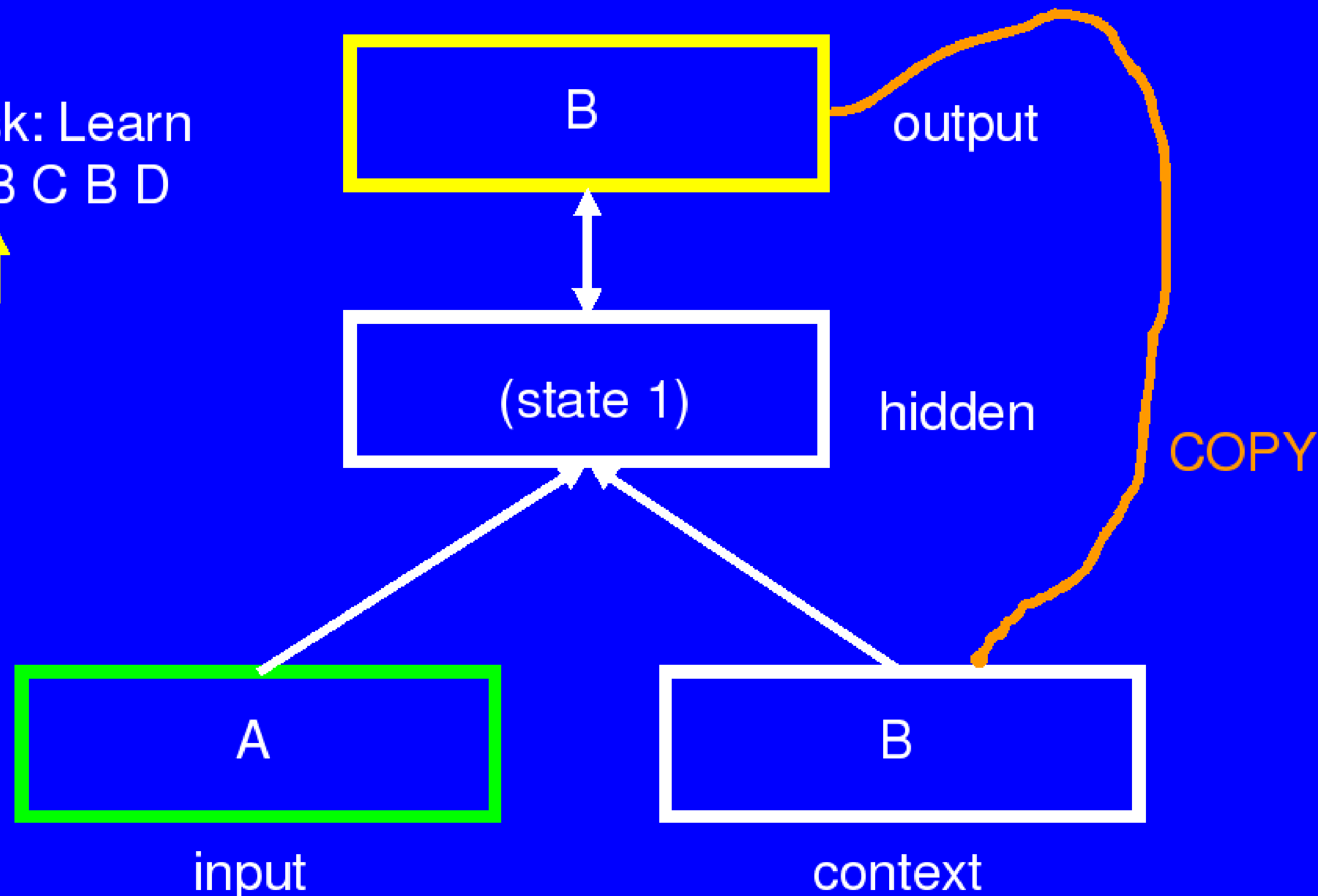
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



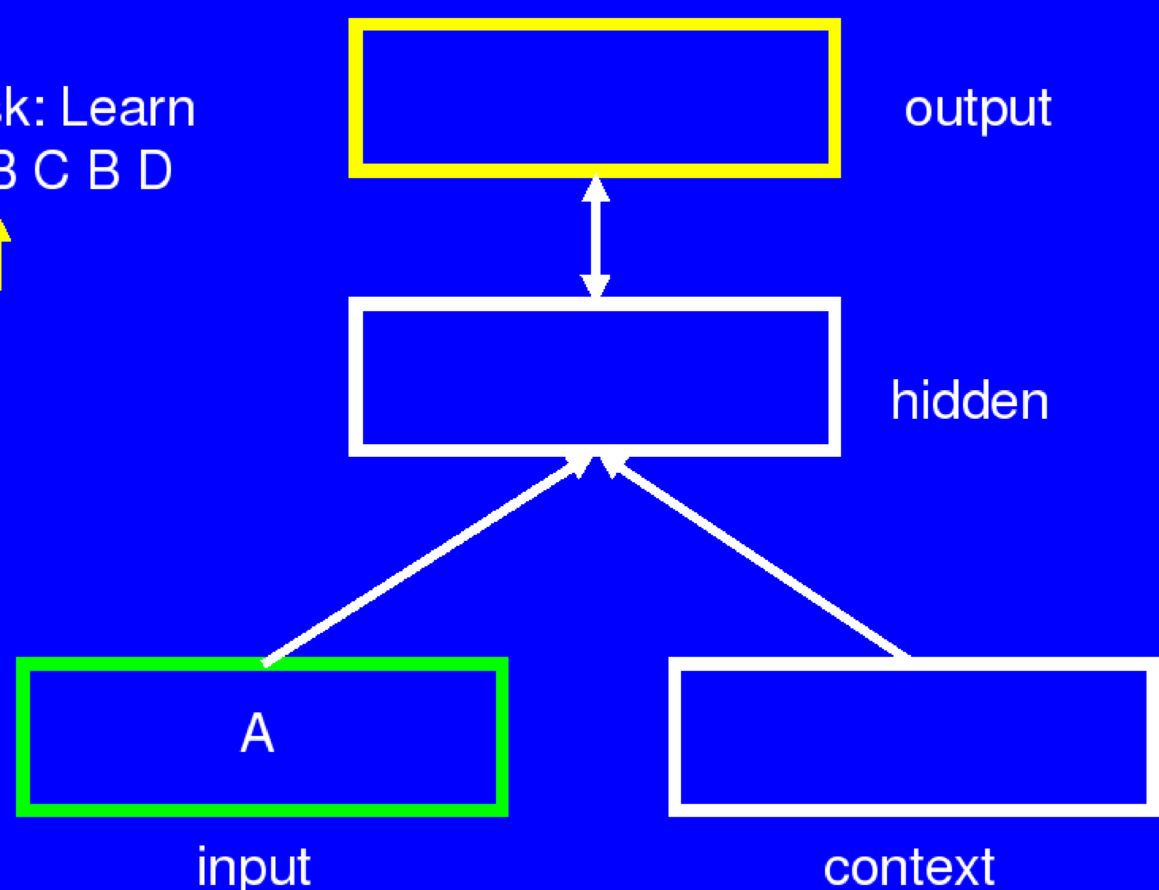
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



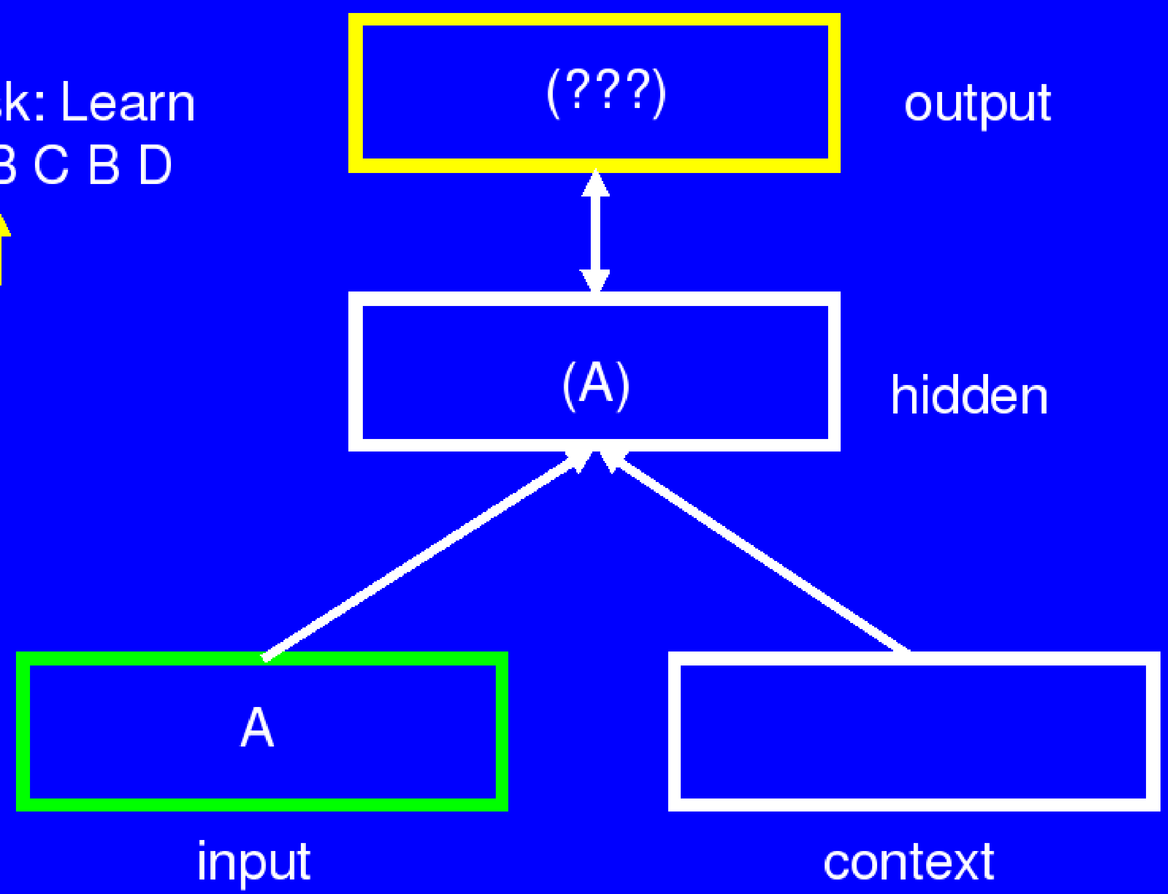
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



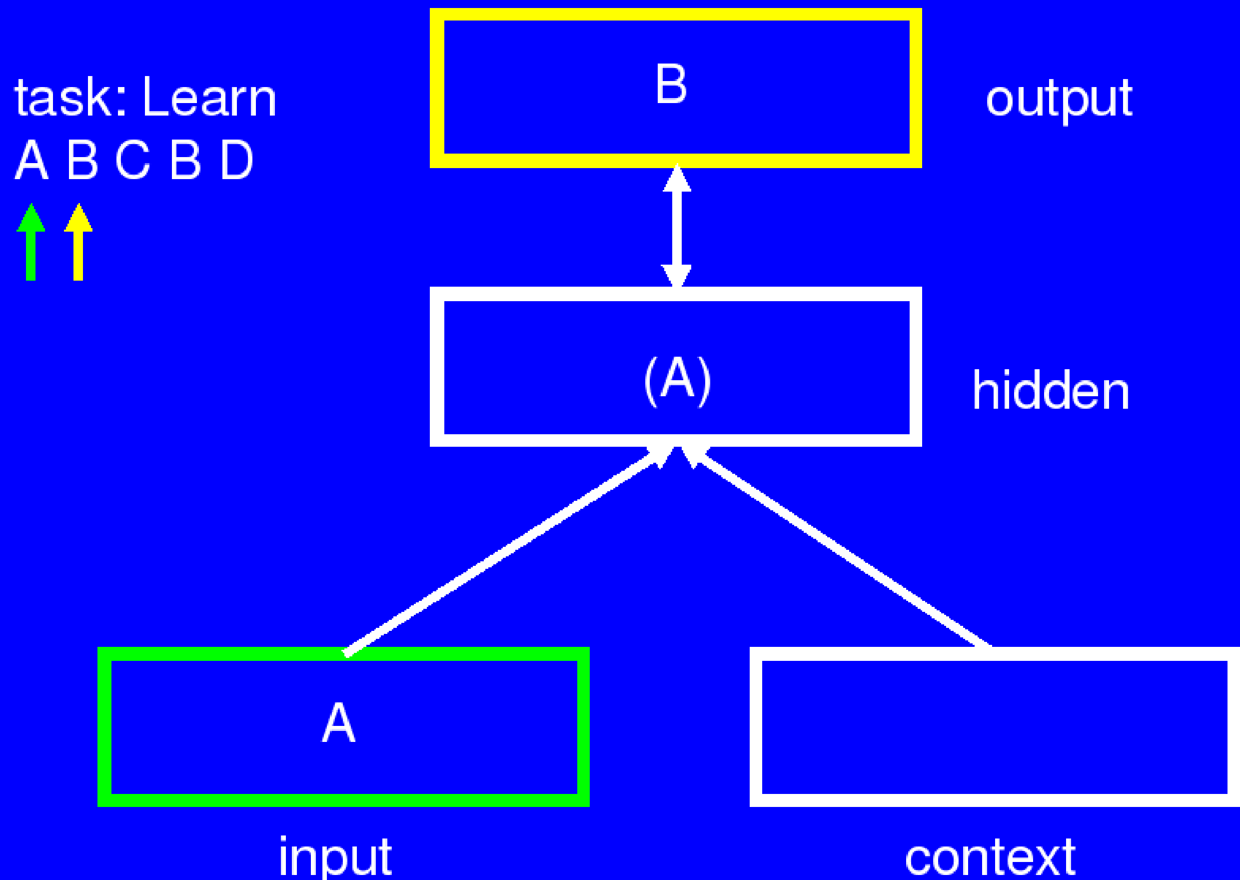
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



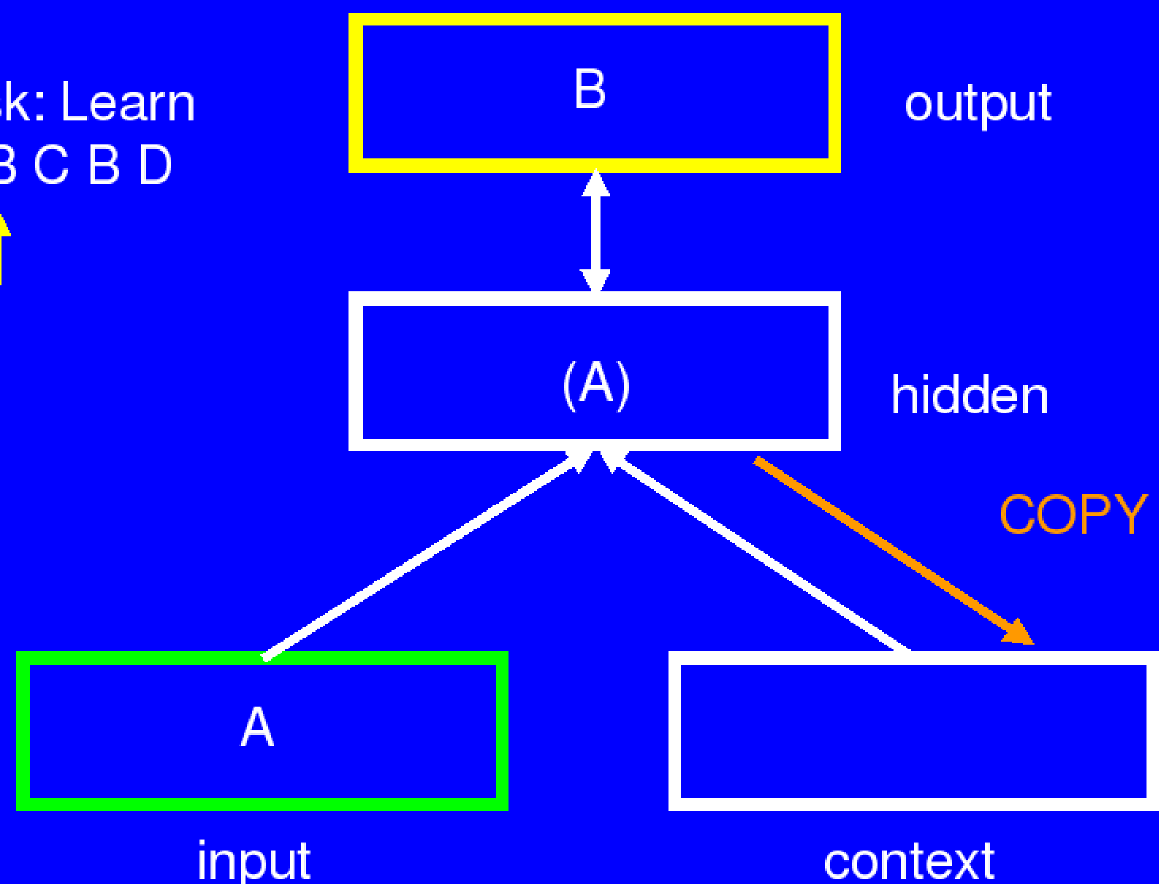
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



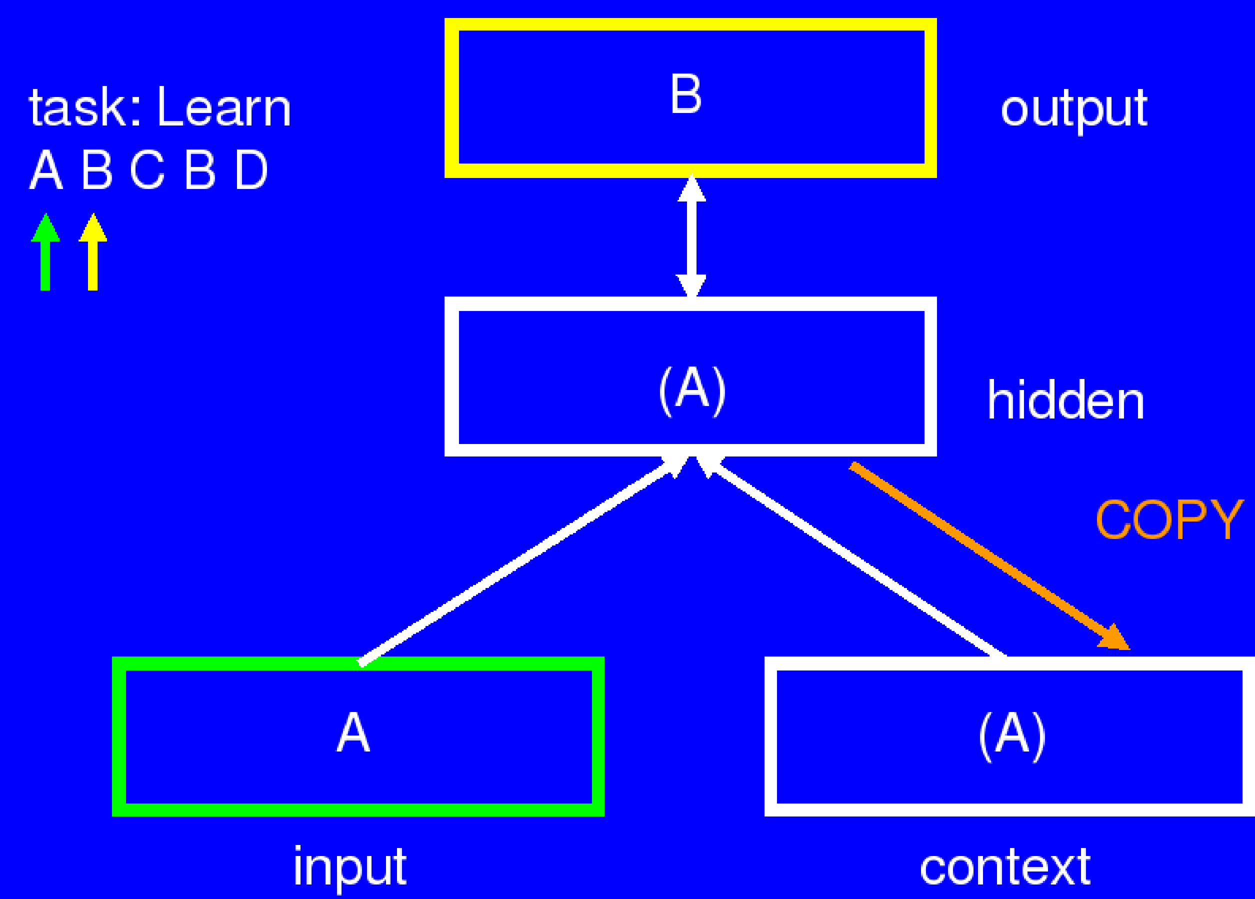
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



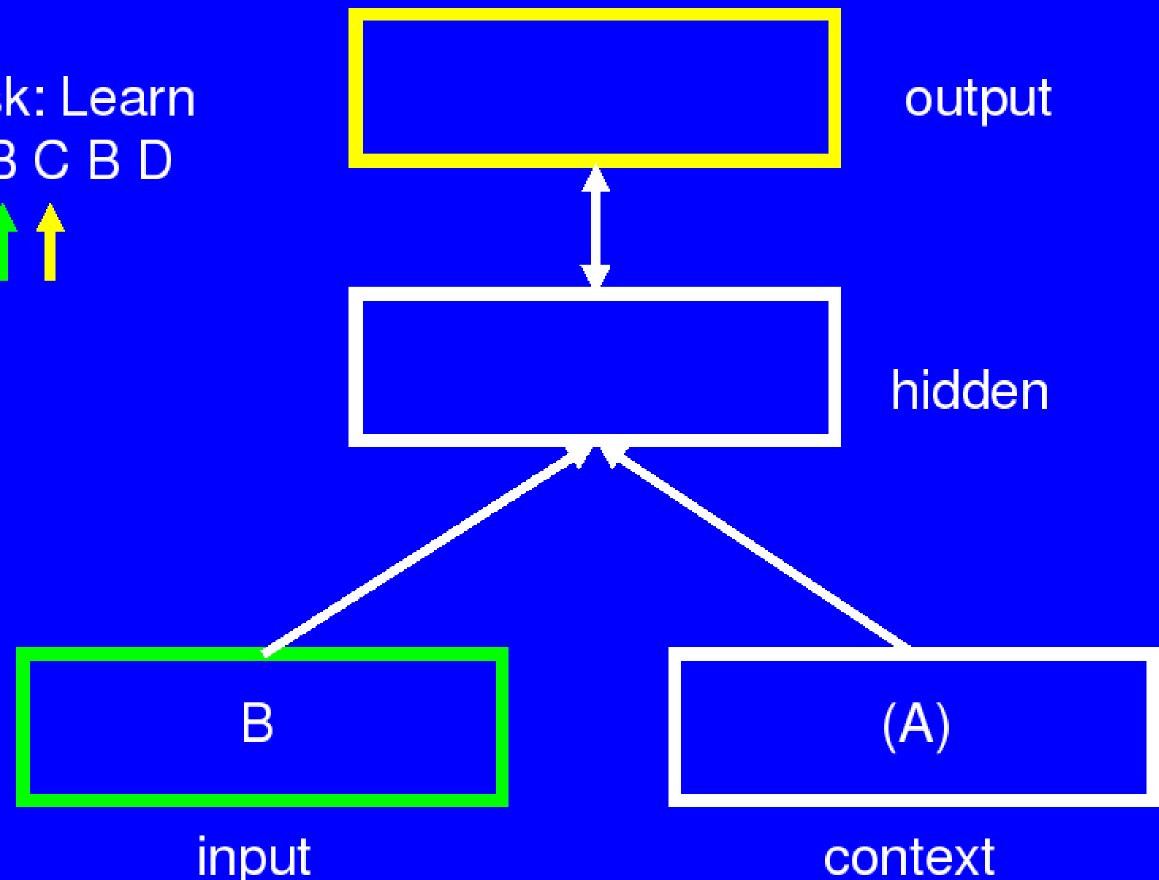
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D

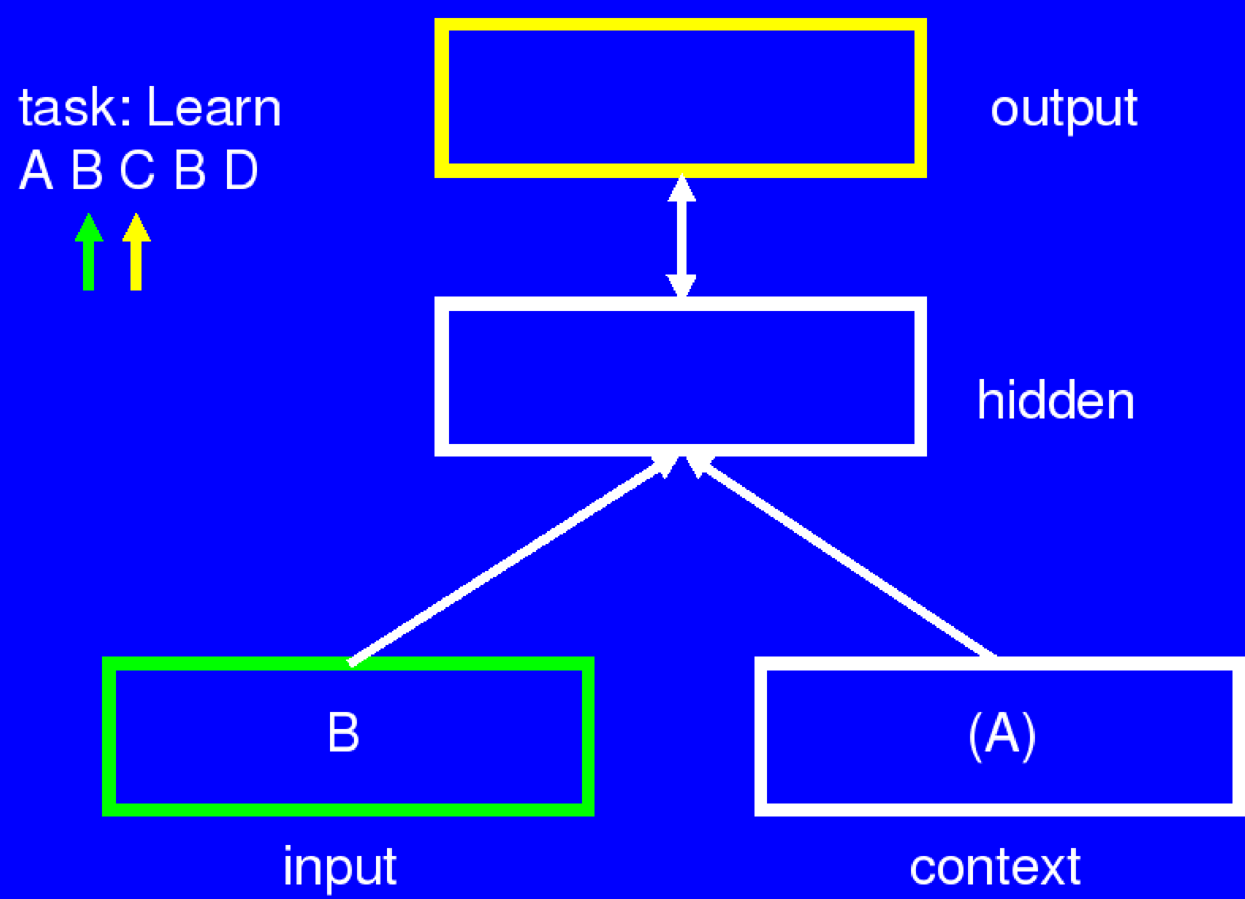


Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D

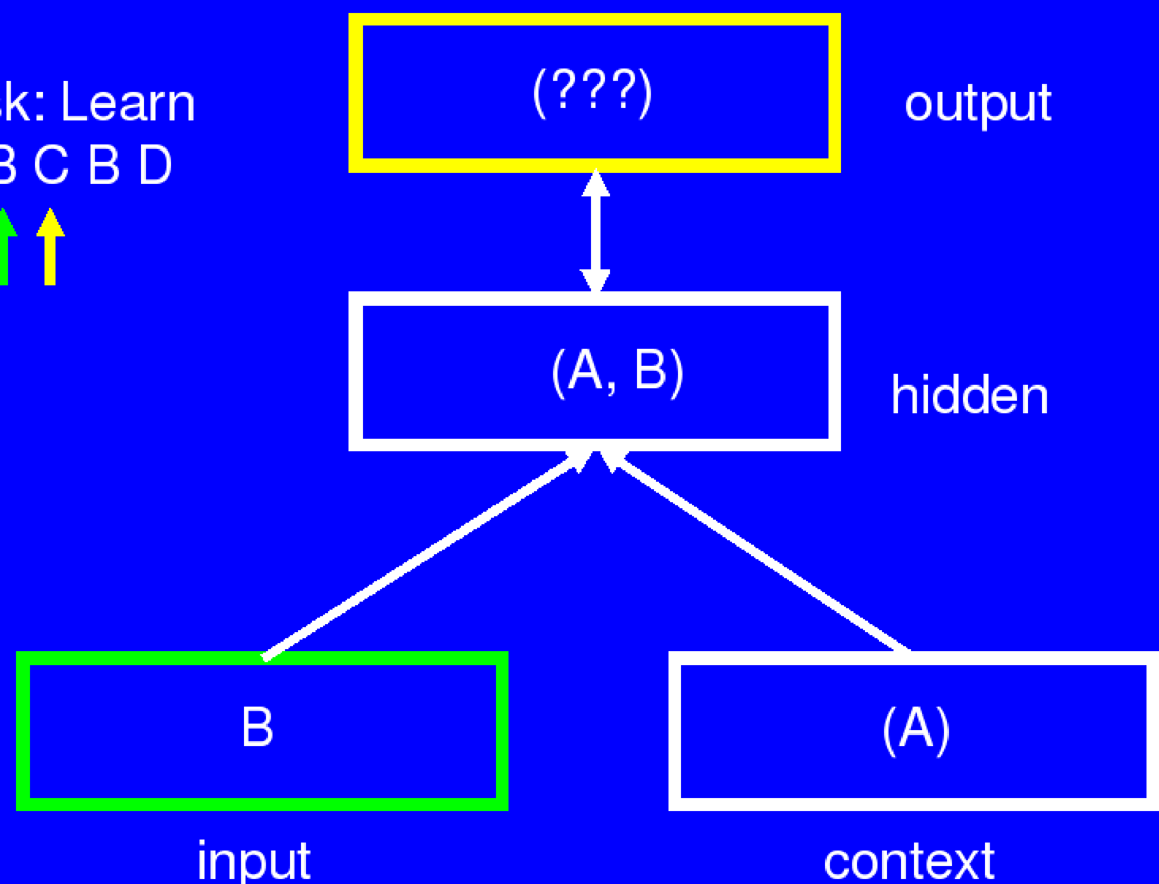


Simple Recurrent Network (SRN): An Architecture for Sequence Learning



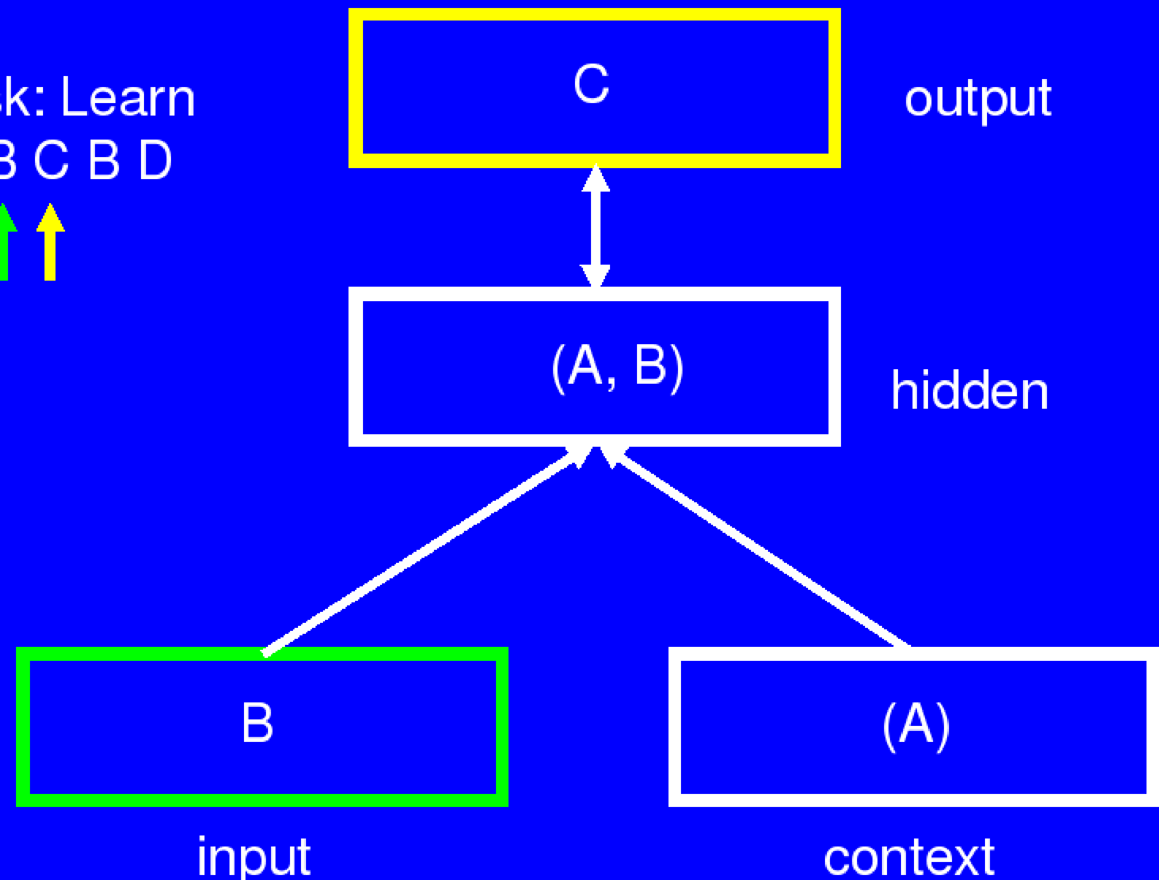
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D

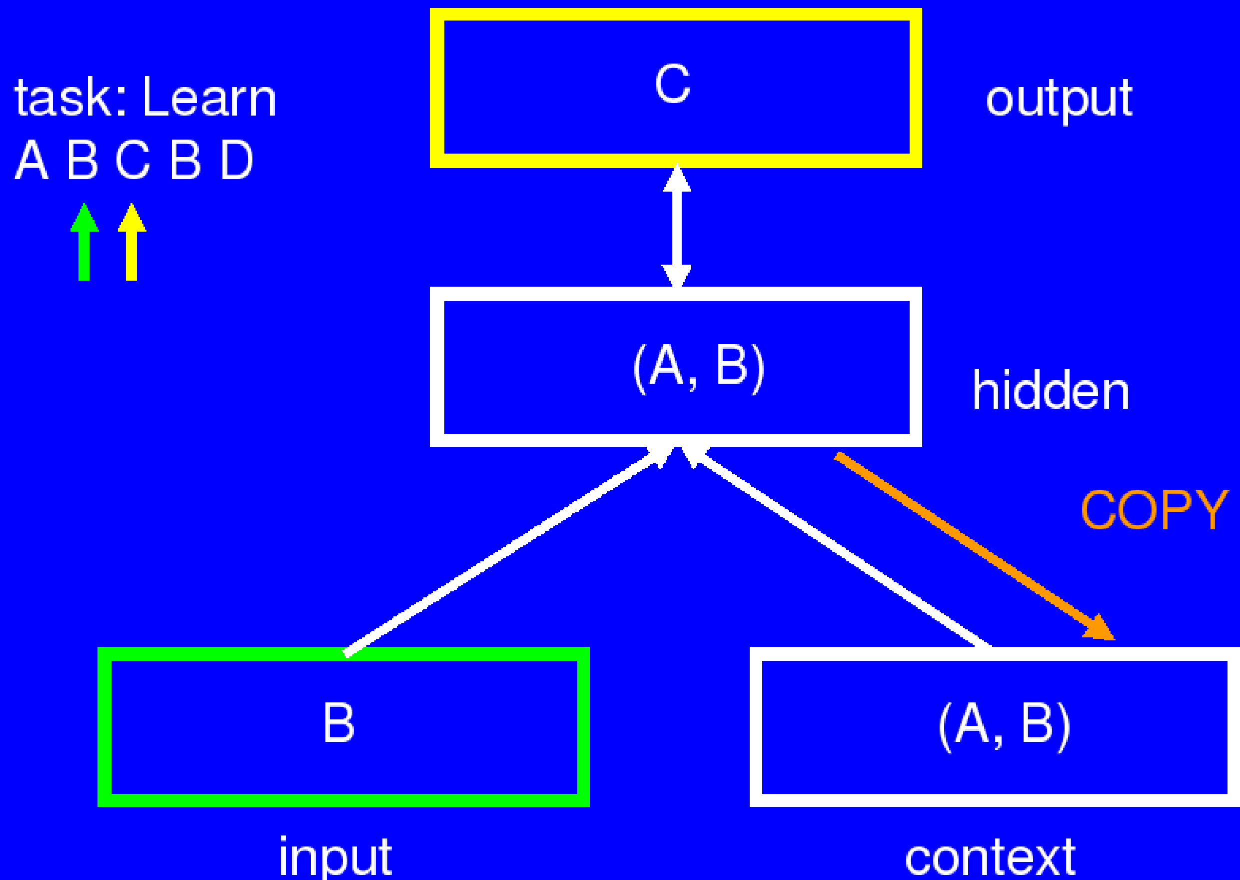


Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D

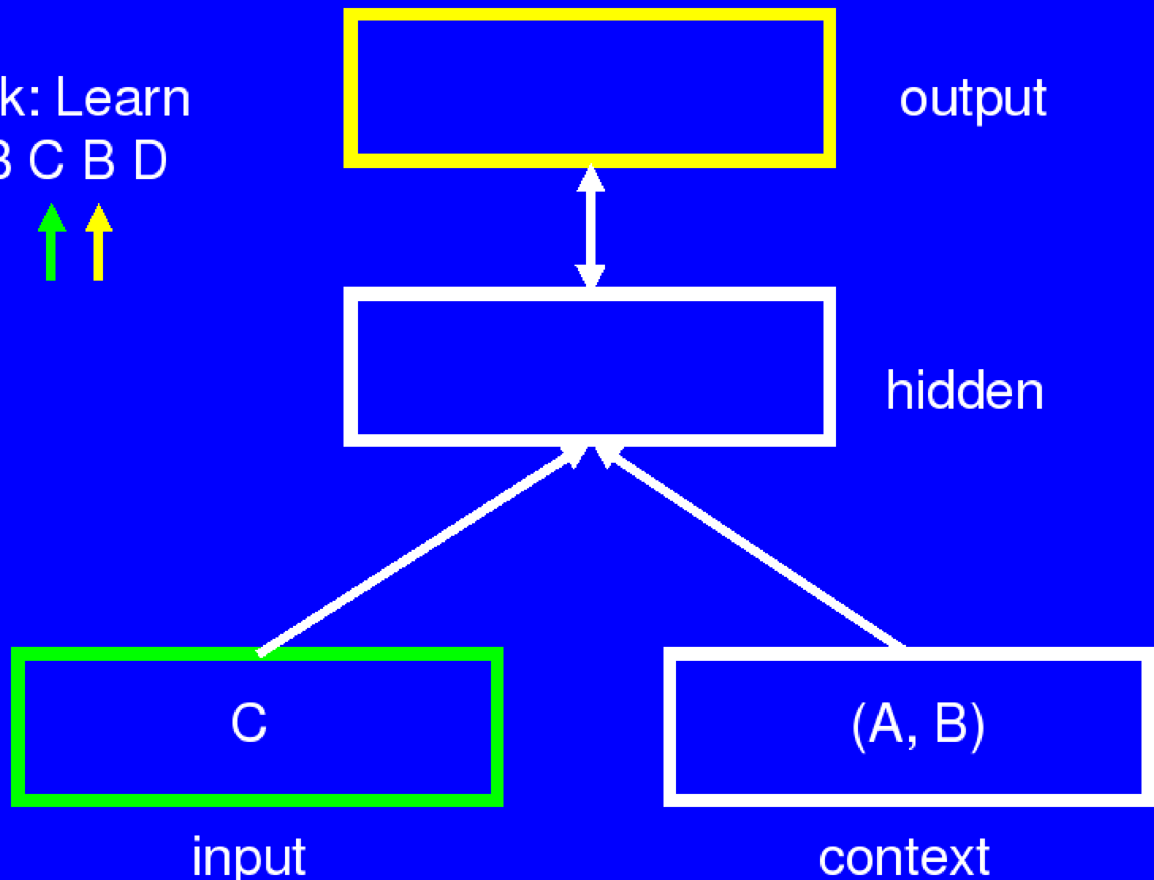


Simple Recurrent Network (SRN): An Architecture for Sequence Learning

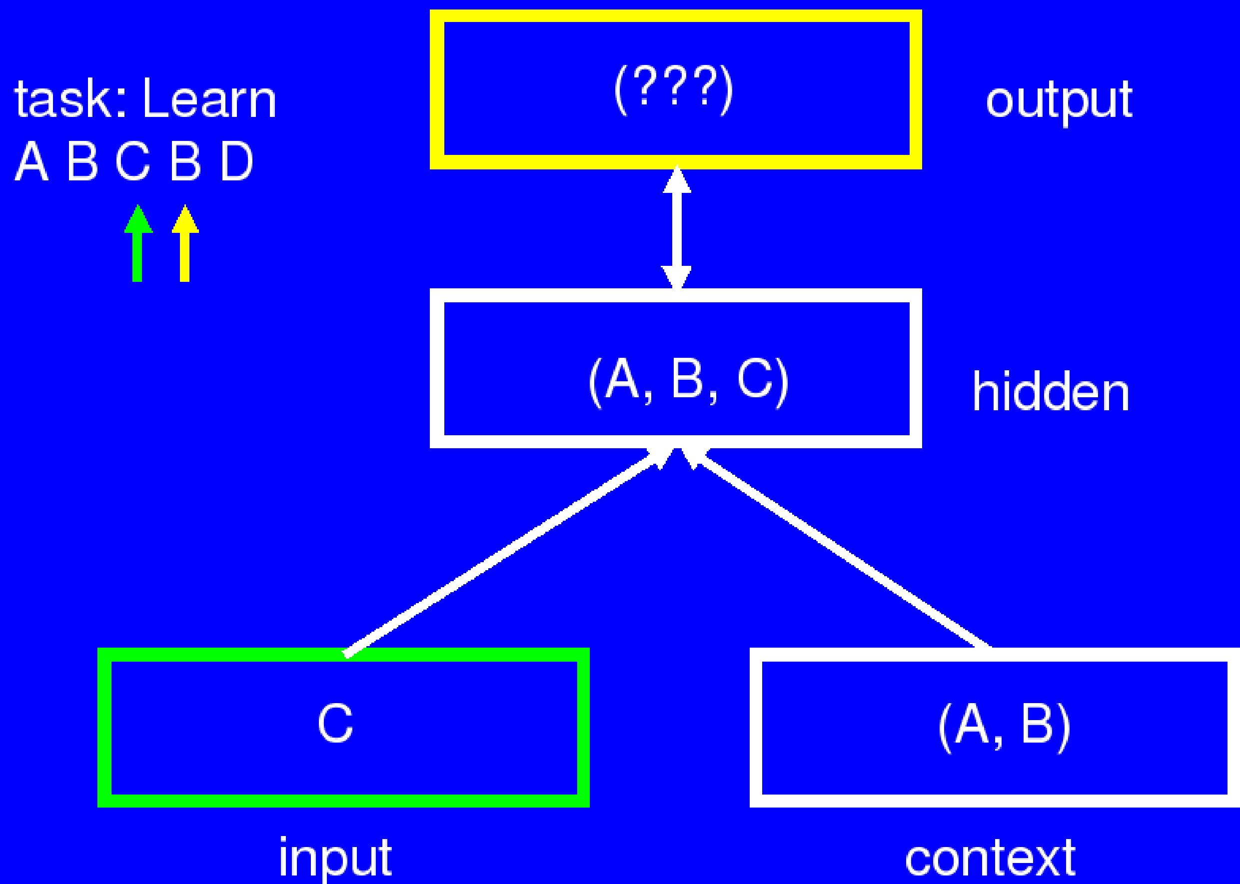


Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D

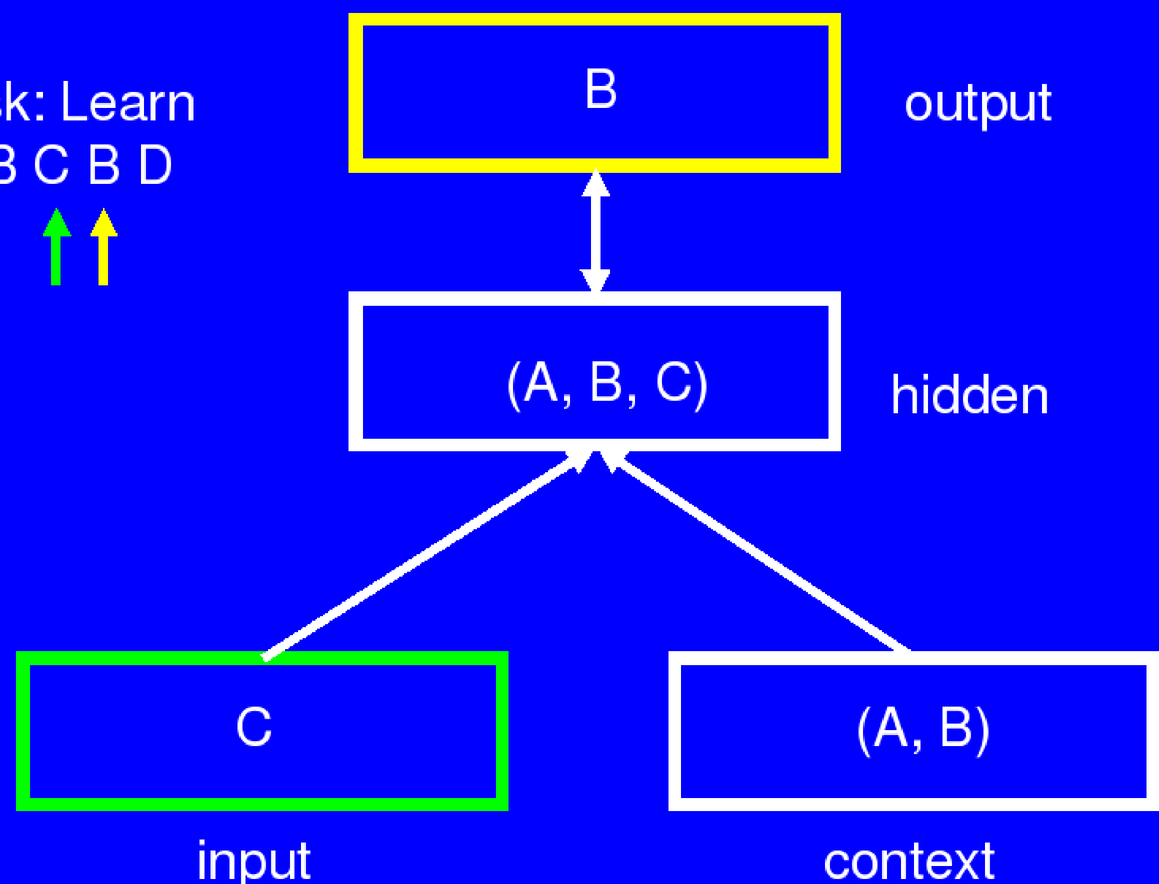


Simple Recurrent Network (SRN): An Architecture for Sequence Learning

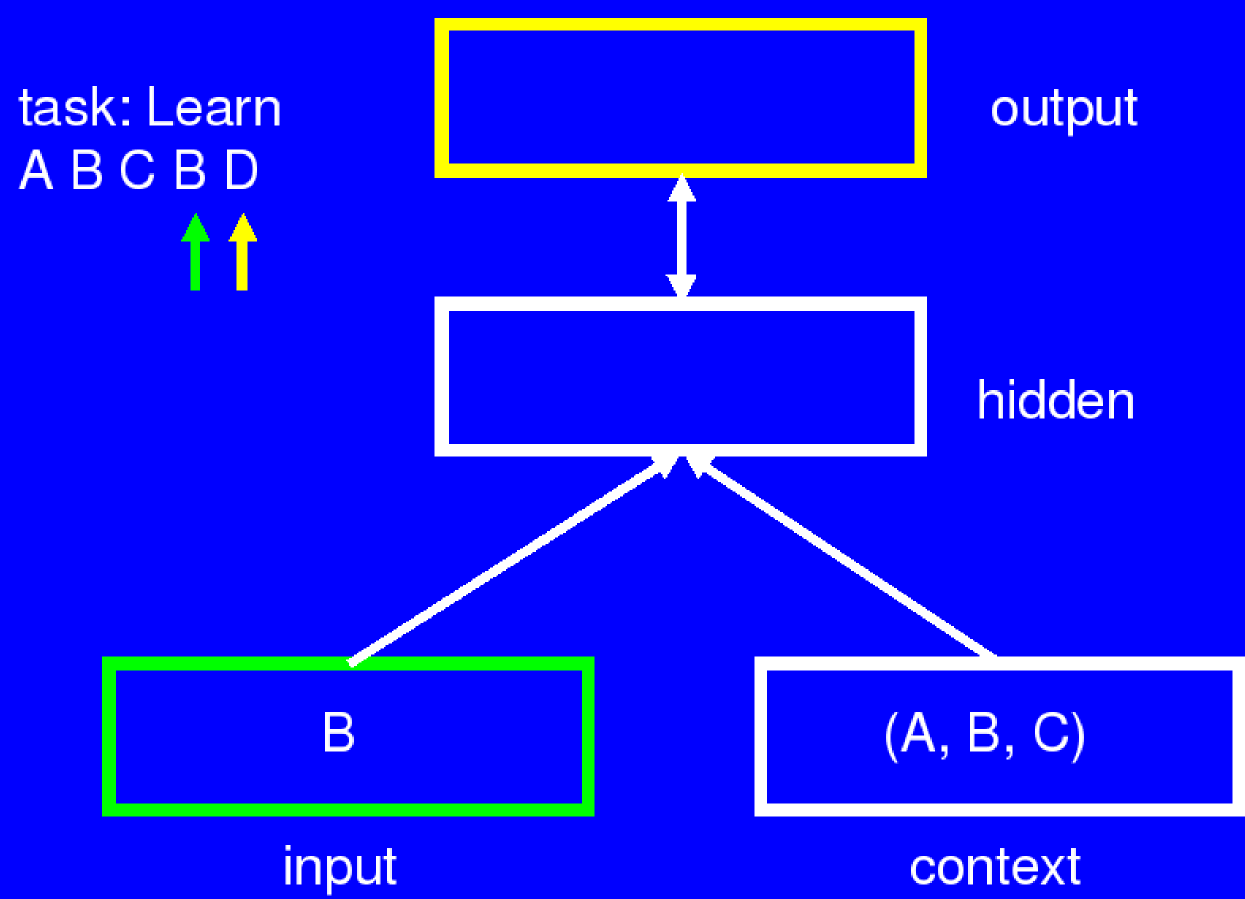


Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D



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

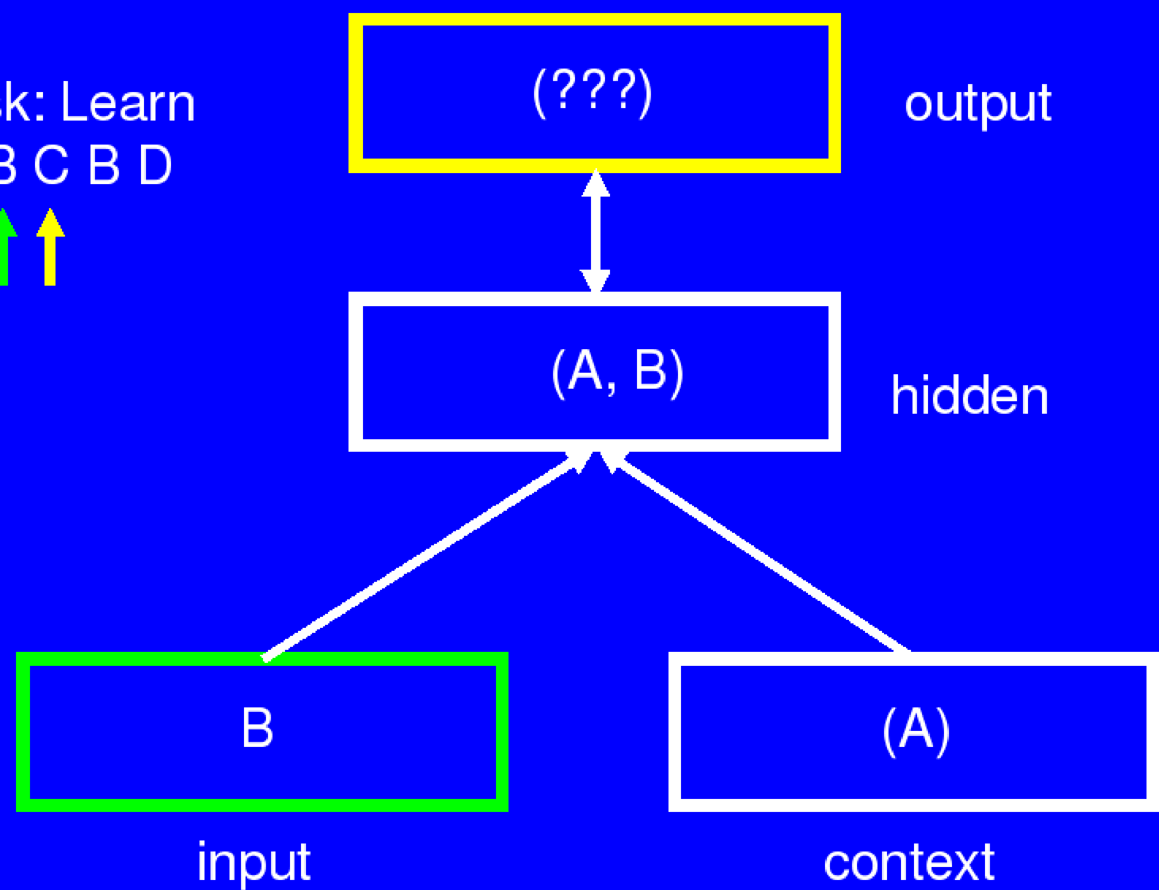


Why Copy the Hidden Representation?

- Copying input or output only lets the network hold on to one previous item
- Copying the hidden layer lets the network hold on to an arbitrarily large number of items – **even though it is always just copying last hidden state at time t-1.**
- The network *learns* how strongly to hold on to past items

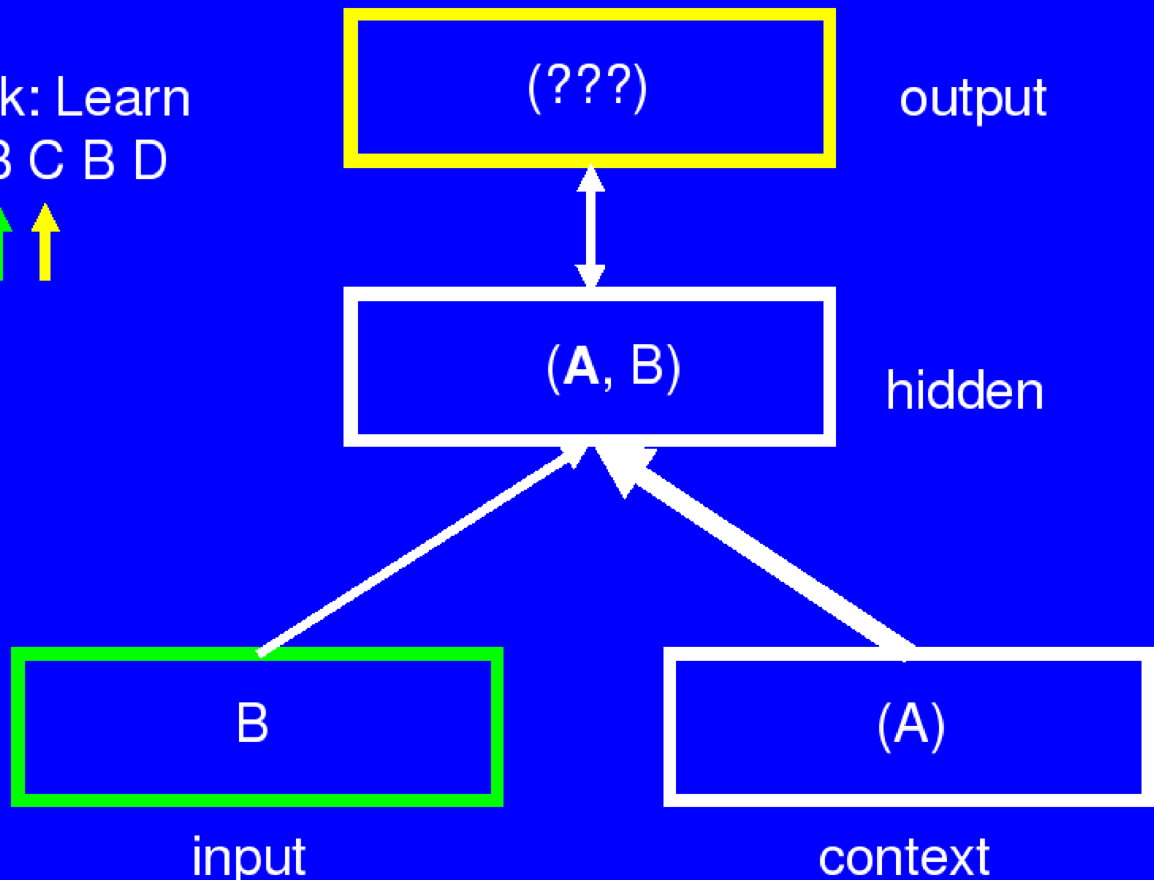
Simple Recurrent Network (SRN): An Architecture for Sequence Learning

task: Learn
A B C B D

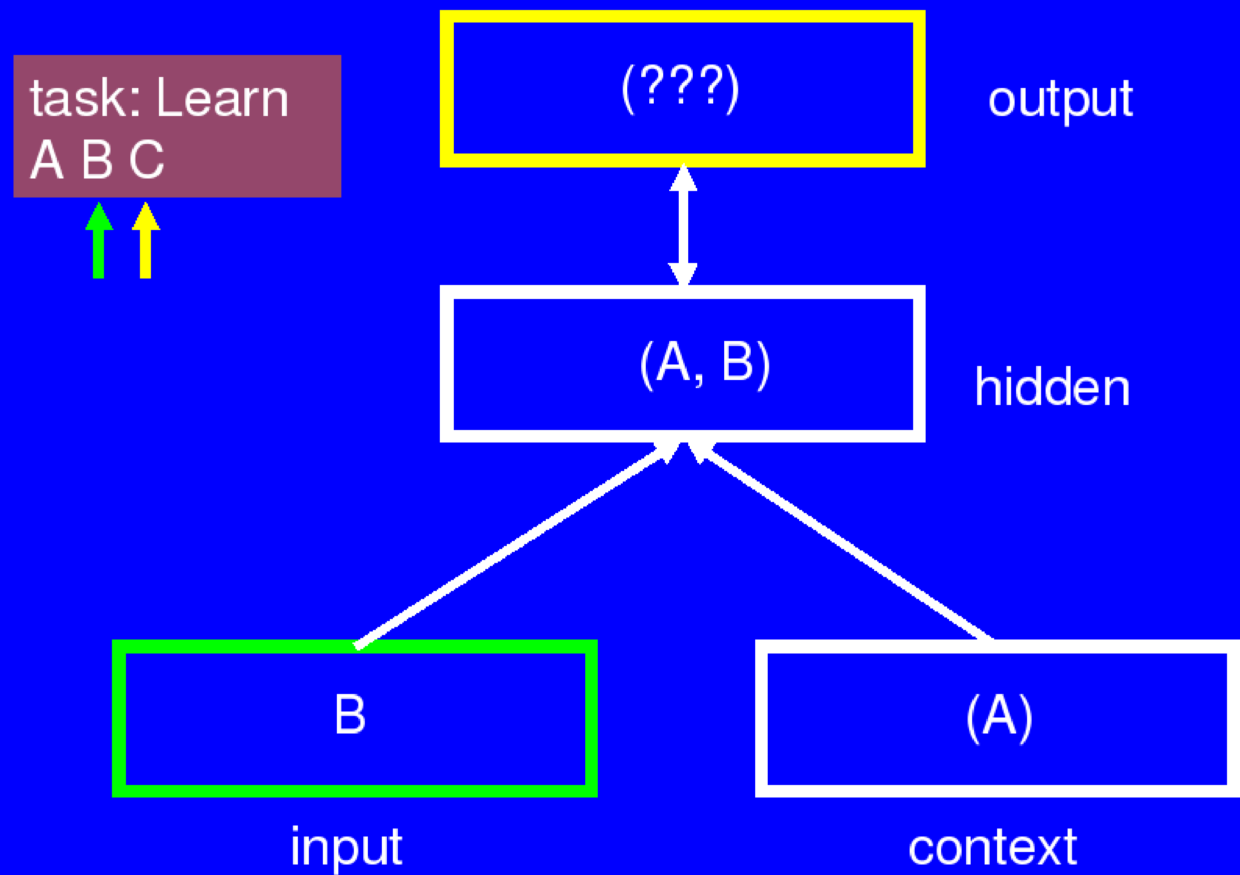


Simple Recurrent Network (SRN): An Architecture for Sequence Learning

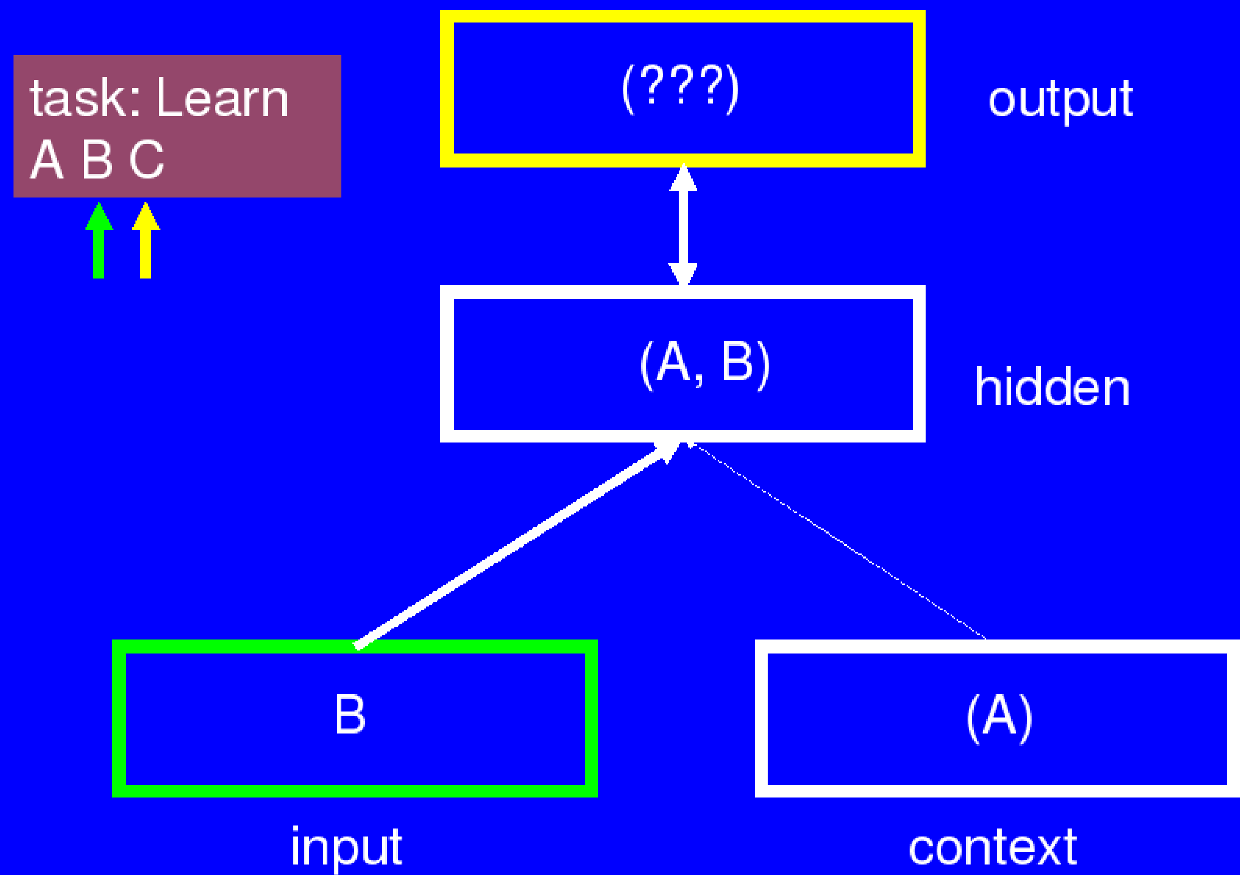
task: Learn
A B C B D



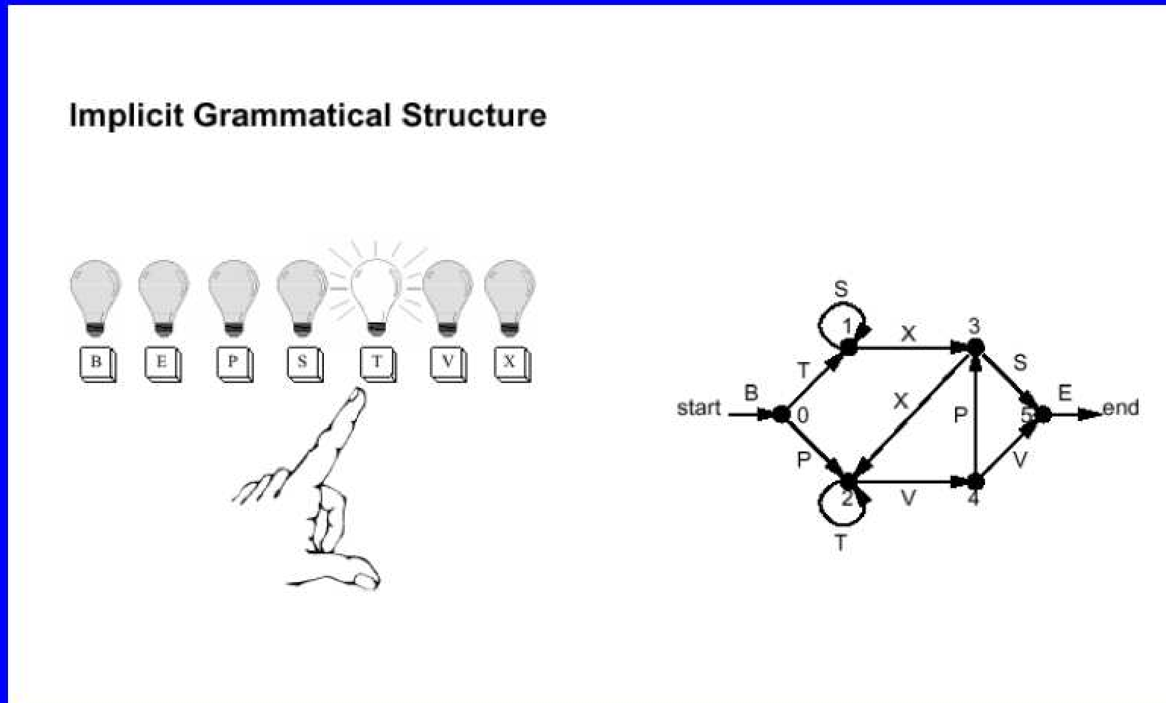
Simple Recurrent Network (SRN): An Architecture for Sequence Learning



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



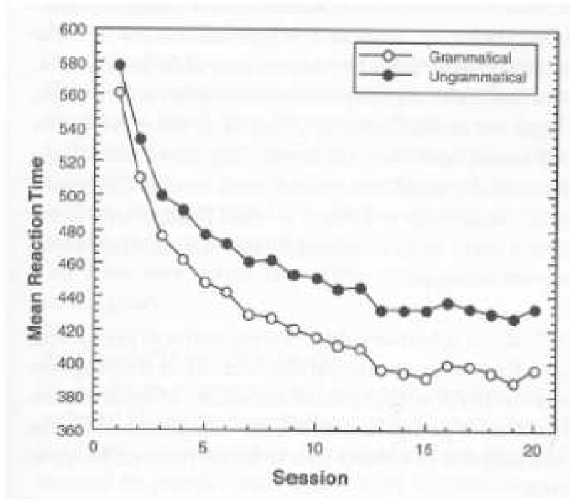
Artificial Grammar Learning



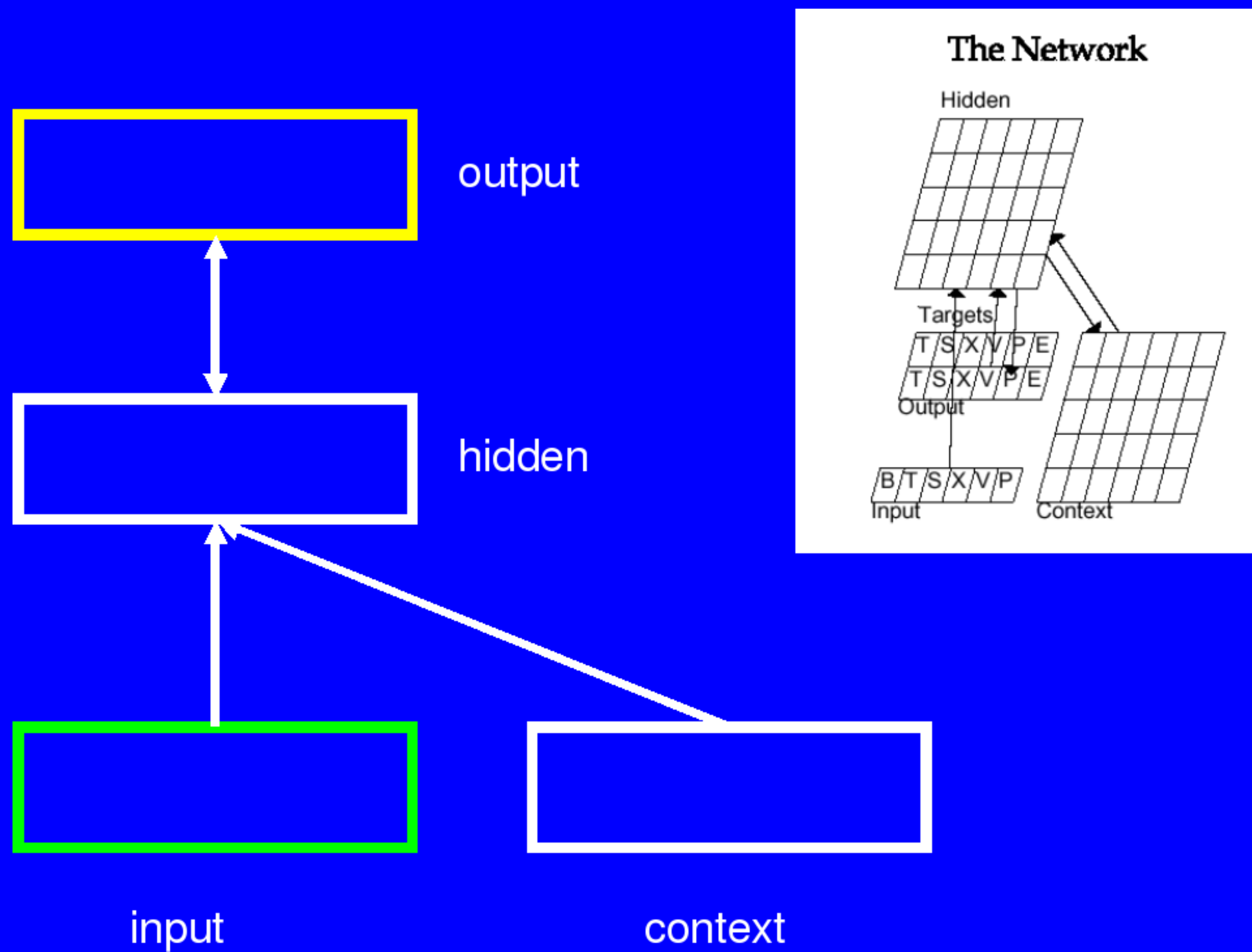
Task: Press buttons that implicitly follow the grammar

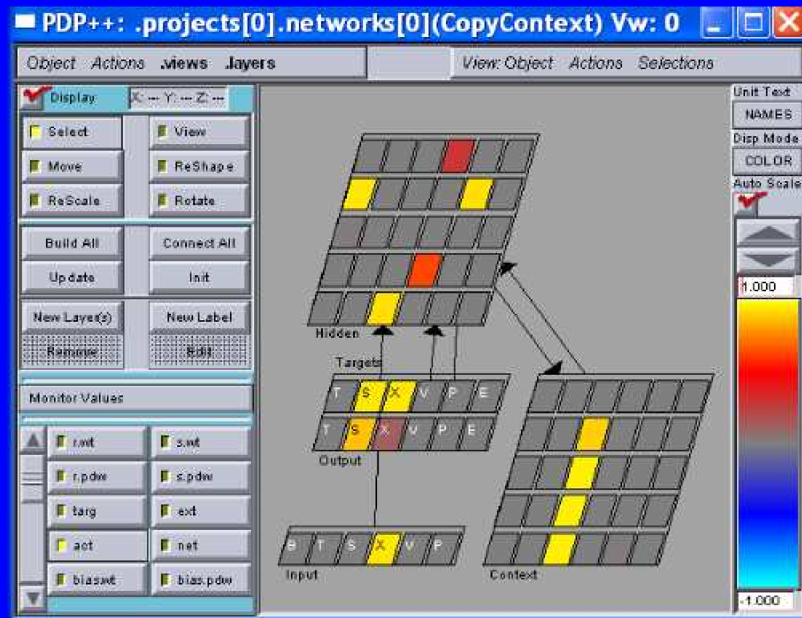
Note: Subjects do not know about the grammar!

Evidence For Implicit Learning



- ➤ Response times are lower for grammatical sequences.
- ➤ Learners report no knowledge of a sequential pattern.
- ➤ Learners are at chance when asked to predict the next light.


output
hidden
input
context
The Network
Hidden
Targets
T S X V P E
T S X V P E
Output
B T S X V P
Input
Context



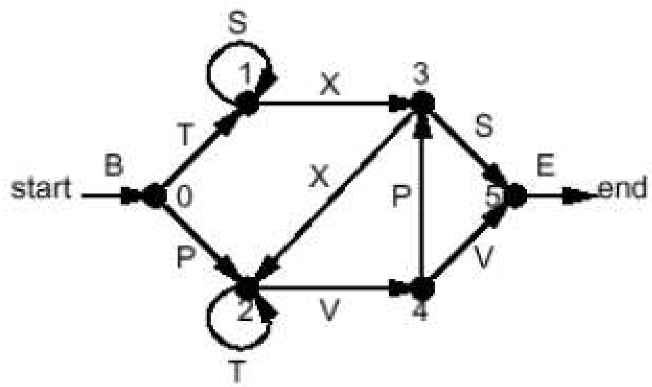
Some details:

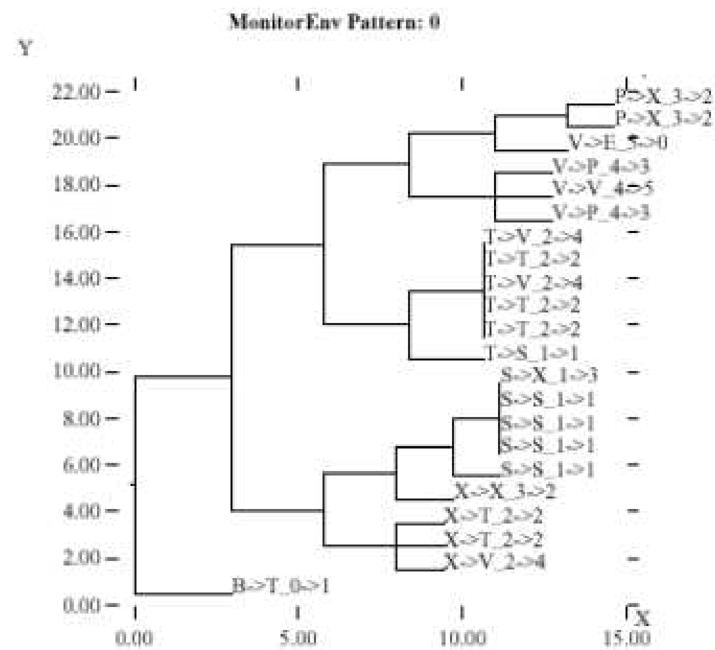
- Input patterns: Generated (randomly) using the finite state grammar

- Sometimes there are multiple “correct” answers for what comes next. kWTA forces the network to guess one answer

Hidden Activity Reflects State Information

Grammatical state information, along with information concerning the last input, is available in the pattern of hidden layer activity.





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.

→ Stay tuned for specialized biological/computational mechanisms for *updating / gating* vs. robust *maintenance* of context.

Motivating Motivation

Why does anyone go to university?

Motivating Motivation

Why does anyone go to university?

(or, why do we ever do anything besides eat, sleep, have sex, etc)?

Motivating Motivation

Why does anyone go to university?

(or, why do we ever do anything besides eat, sleep, have sex, etc)?

e.g., Why am I here today, instead of lying on a beach in Mexico, drinking mojitos and reading a good book?

Motivating Motivation

Why does anyone go to university?

(or, why do we ever do anything besides eat, sleep, have sex, etc)?

e.g., Why am I here today, instead of lying on a beach in Mexico, drinking mojitos and reading a good book?

Challenge: make a responsible neural network!

The Motivational Bootstrap

- Some motivations must be built-in (else we would die)
- Where do art/science come from?
  - Need to learn on top of built-in drives

The Motivational Bootstrap

- Some motivations must be built-in (else we would die)
- Where do art/science come from?
  - Need to learn on top of built-in drives

Culture & social drives provide cumulative shaping of learning.

The Motivational Bootstrap

- Some motivations must be built-in (else we would die)
- Where do art/science come from?
  - Need to learn on top of built-in drives

Culture & social drives provide cumulative shaping of learning.

So, why does anyone go to university?

- Socially-mediated standards of success.
- Strong built-in desire to share w/others.
- Strong built-in desire to learn (dopamine?)

What I’m Actually Talking About

Skinnerian learning

The basic stuff that every mammal has in common:

Neural mechanisms of Pavlovian conditioning
(from a computational perspective).

What I'm Actually Talking About

Skinnerian learning

The basic stuff that every mammal has in common:

Neural mechanisms of Pavlovian conditioning
(from a computational perspective).

No supervised target signal available: only good/bad outcomes
Enables *bootstrap* of new stimuli (CS's) onto built-in desires (US's):

CS (money) $\rightarrow$ US (food, etc)

What I'm Actually Talking About

Skinnerian learning

The basic stuff that every mammal has in common:

Neural mechanisms of Pavlovian conditioning
(from a computational perspective).

No supervised target signal available: only good/bad outcomes
Enables *bootstrap* of new stimuli (CS's) onto built-in desires (US's):

CS (money) $\rightarrow$ US (food, etc)

But what if consequence of given input comes *later* in time?

Temporally-Extended Tasks with Sparse Rewards

Temporally-delayed Learning & Reinforcement

Reinforcement often delayed from the events that lead to it: need to "span the gap".

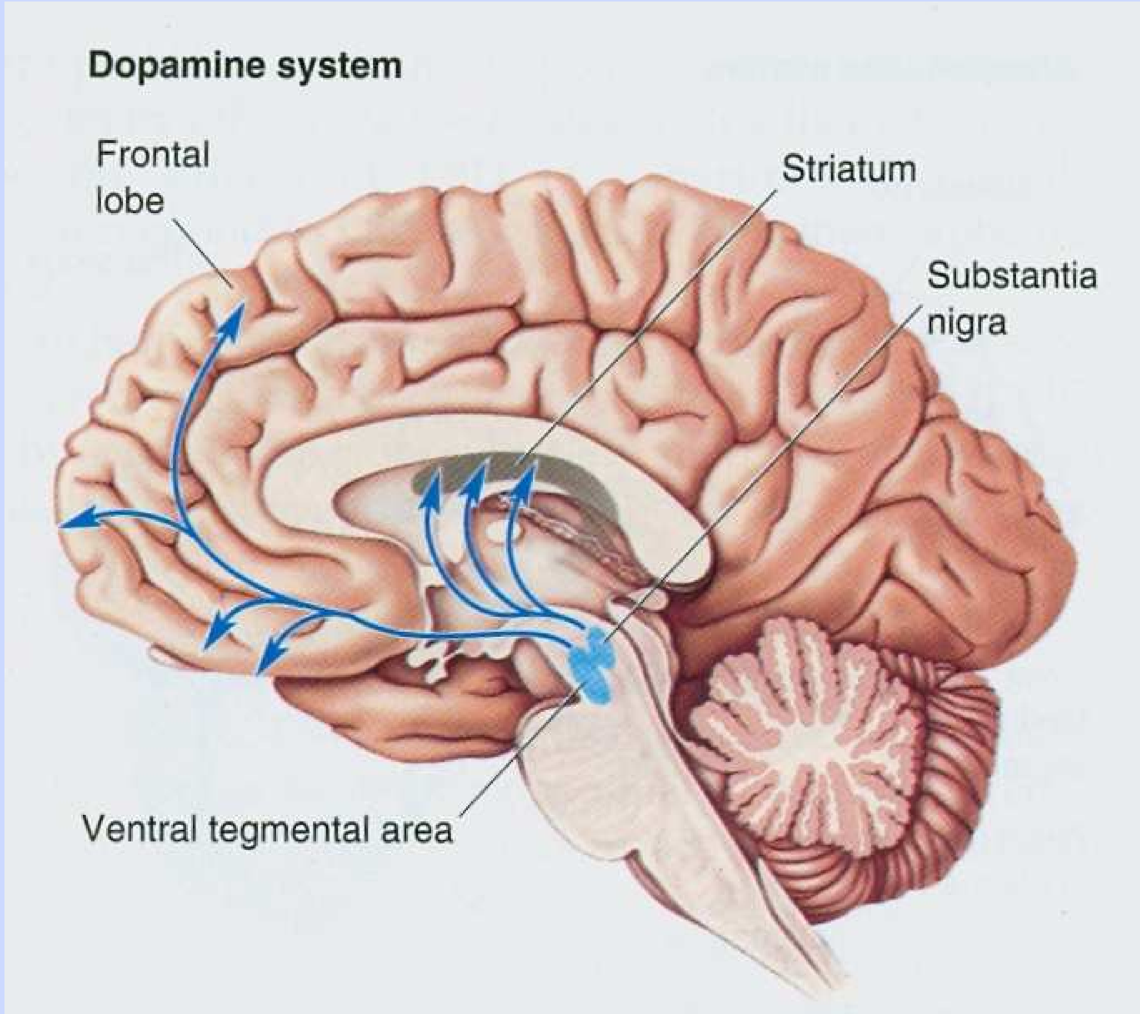
Temporally-delayed Learning & Reinforcement

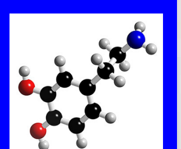
Reinforcement often delayed from the events that lead to it:
need to "span the gap".

Key idea:

- We want to predict future rewards consistently over time.
- This allow us to learn what events are associated with rewards, earlier and earlier back in time.

We use the Temporal Differences (TD) algorithm (Sutton & Barto).

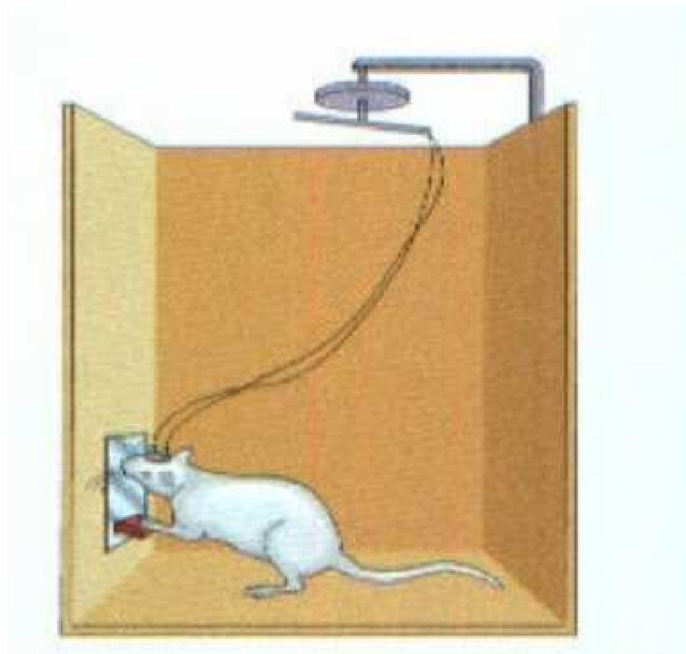

Dopamine system
Frontal lobe
Striatum
Substantia nigra
Ventral tegmental area




sundance
CHANNEL
HOME ENTERTAINMENT
LOVE.
REAL OR JUST A CHEMICAL REACTION?
John Livingston Sabrina Lloyd
DOPAMINE
"A sweet, smart romance."
- Kevin Crust, Los Angeles Times

What is Dopamine Doing?

Dopamine carries the brain's reward signal

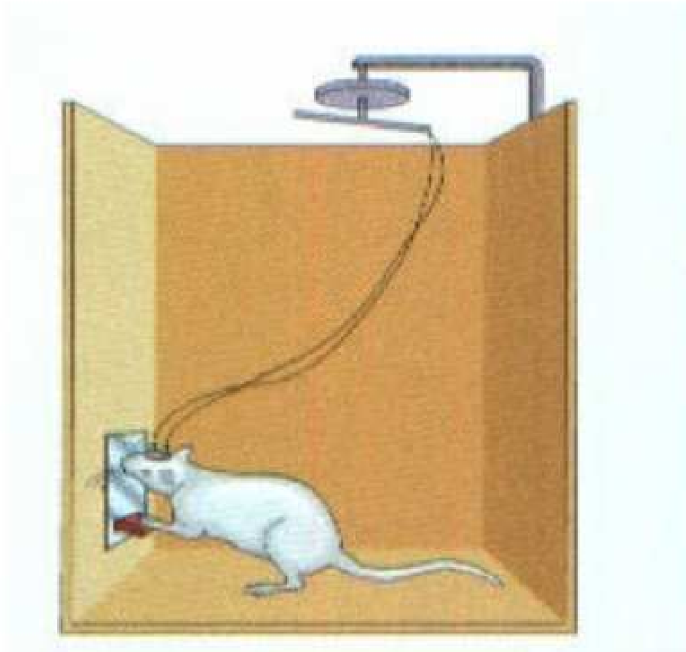


Wise & Romper, 89

Schultz et. al, 98

What is Dopamine Doing?

Dopamine carries the brain's reward signal

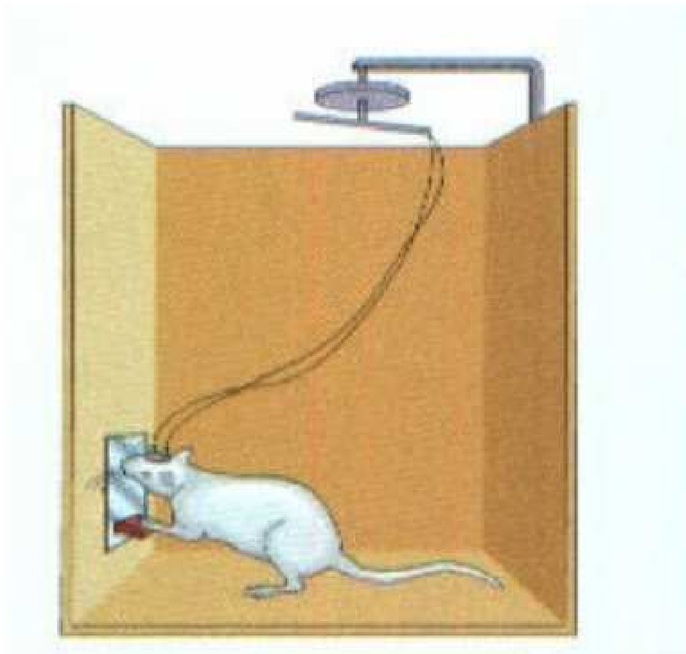


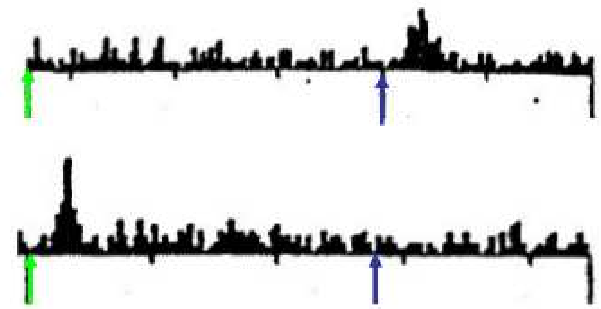
Wise & Romper, 89

Schultz et. al, 98

What is Dopamine Doing?

Dopamine carries the brain's reward signal



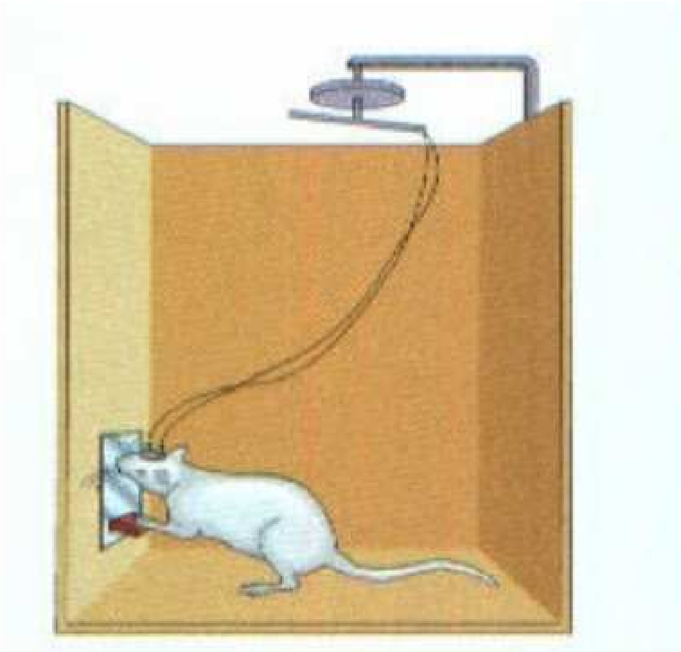


Wise & Romper, 89

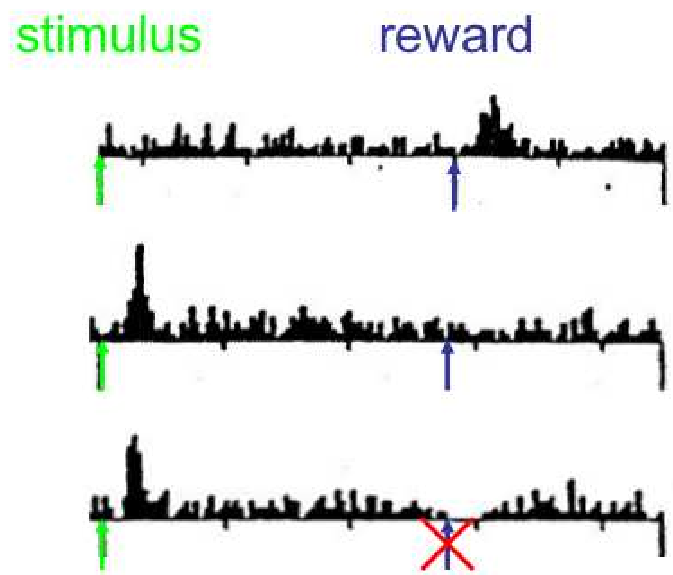
Schultz et. al, 98

What is Dopamine Doing?

Dopamine carries the brain's reward signal



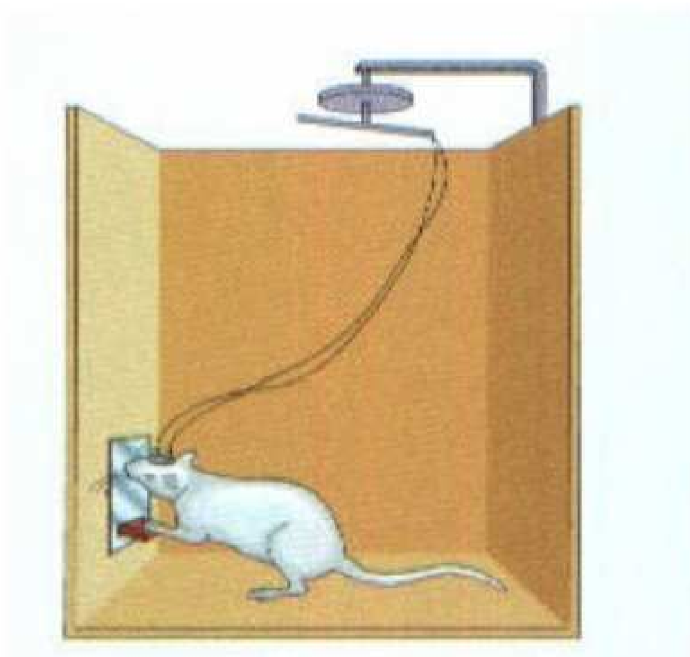
Wise & Romper, 89



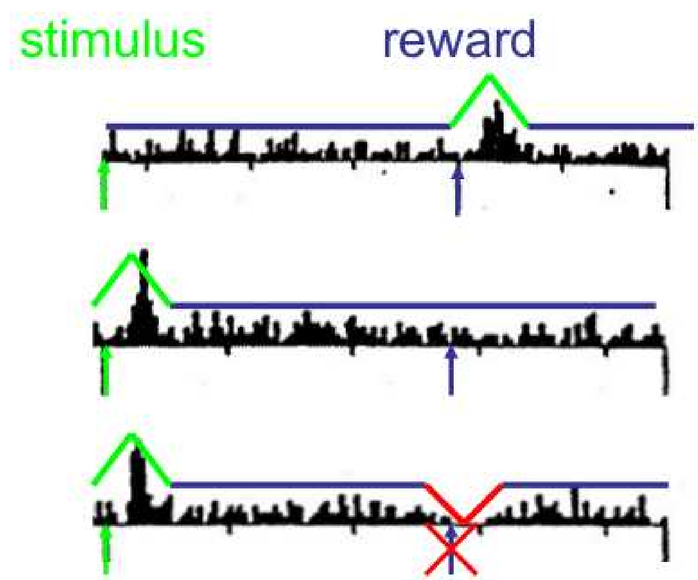
Schultz et. al, 98

What is Dopamine Doing?

Dopamine carries the brain's reward signal



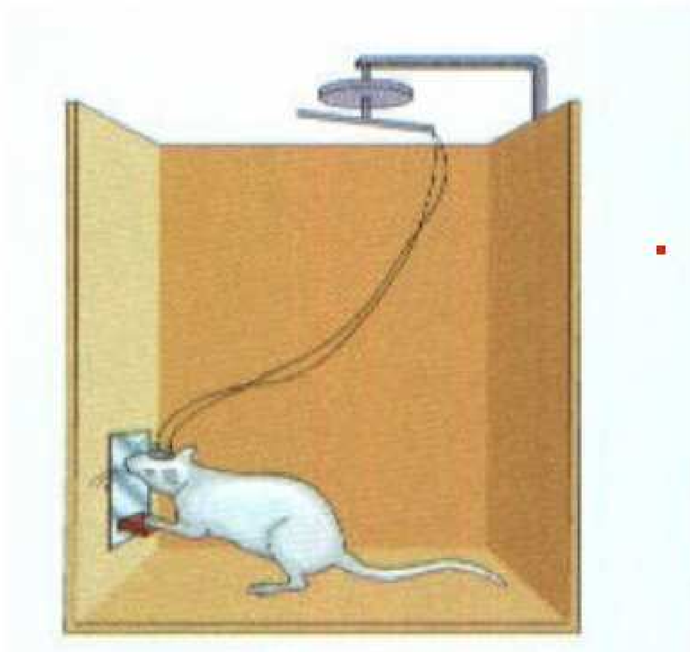
Wise & Romper, 89



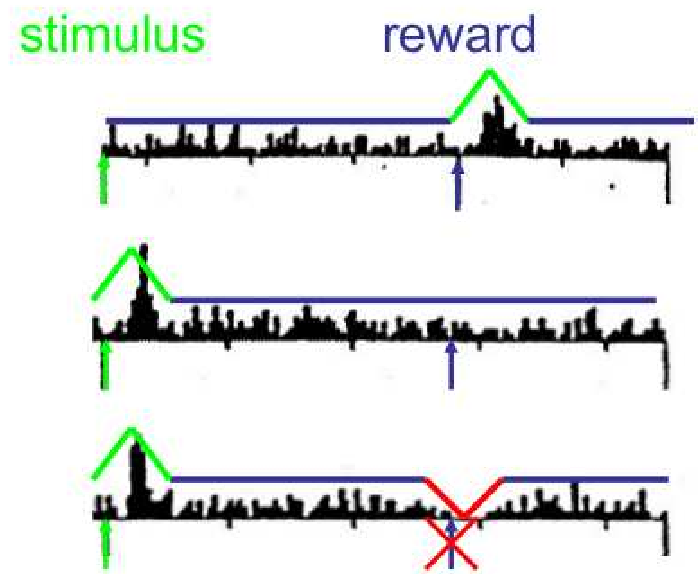
Schultz et. al, 98

What is Dopamine Doing?

Dopamine carries the brain's ~~reward~~ signal
reward prediction error



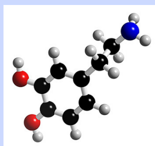
Wise & Romper, 89



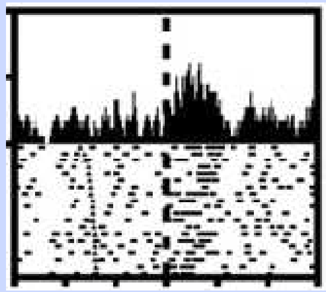
Schultz et. al, 98

Reinforcement learning and dopamine: prediction errors

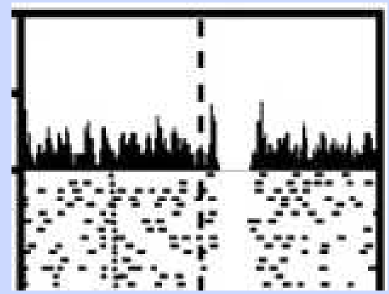
dopamine:



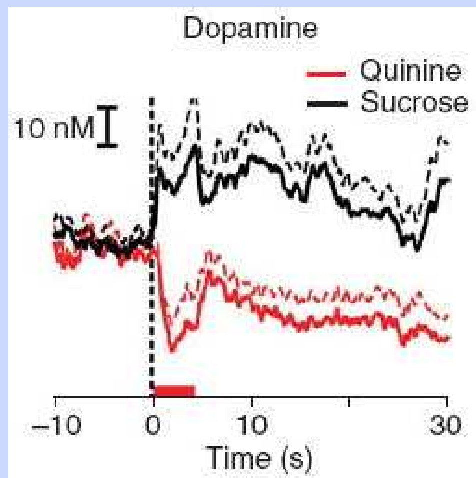
Positive PE:



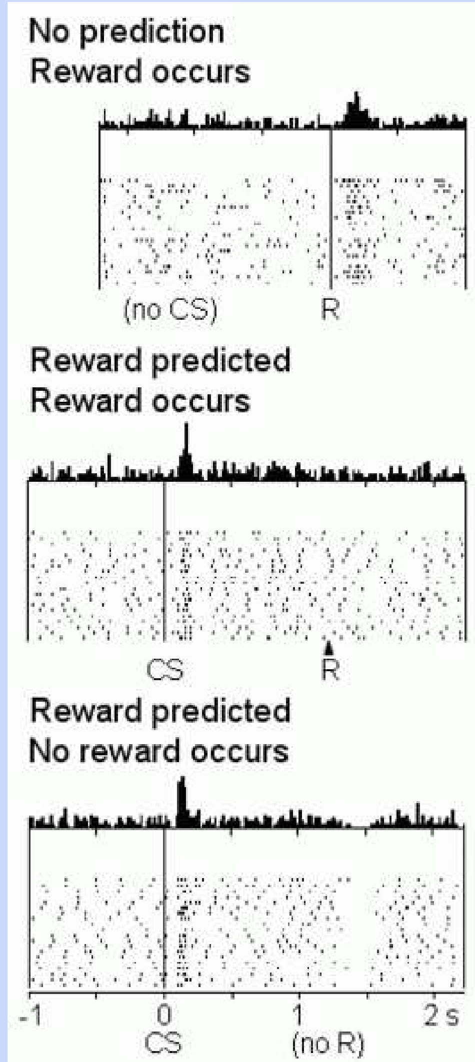
Negative PE:



Schultz, Satoh, Roesch, Zaghoul, Glimcher, Hyland.. and many more

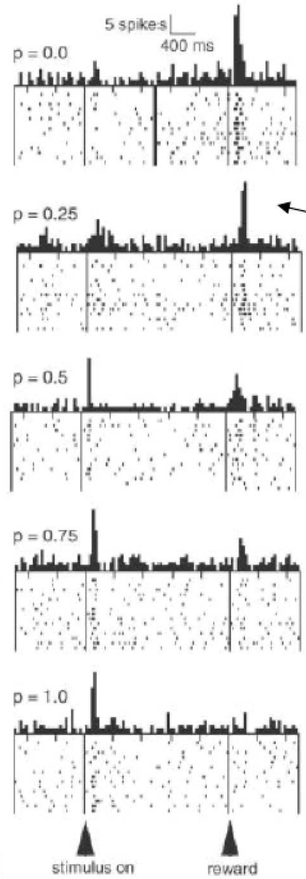


Basic Data: VTA dopamine firing in Conditioning



Schultz, Montague & Dayan, 2007

Dopamine and Reward Probability

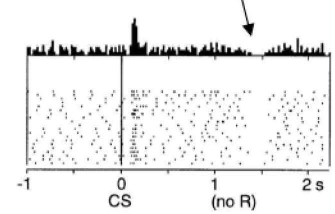


reward following 0% predictive cue

reward following 50% predictive cue

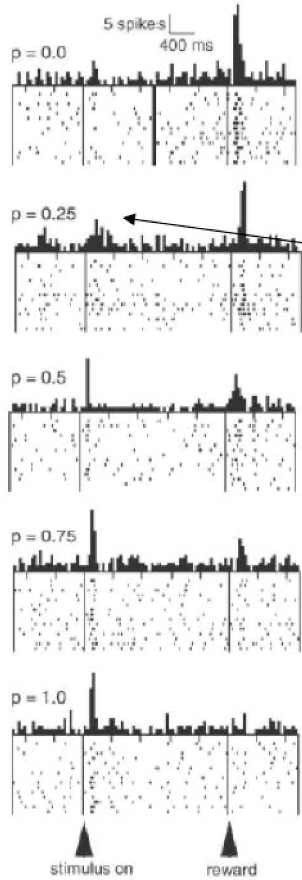
reward following 100% predictive cue

Prediction error:
$r_t - EV_t$



no reward following 100% predictive cue

Dopamine and Reward Probability

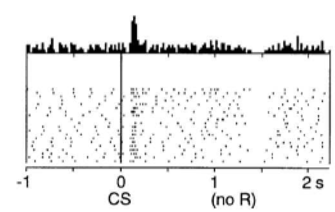


reward following 0% predictive cue

reward following 50% predictive cue

reward following 100% predictive cue

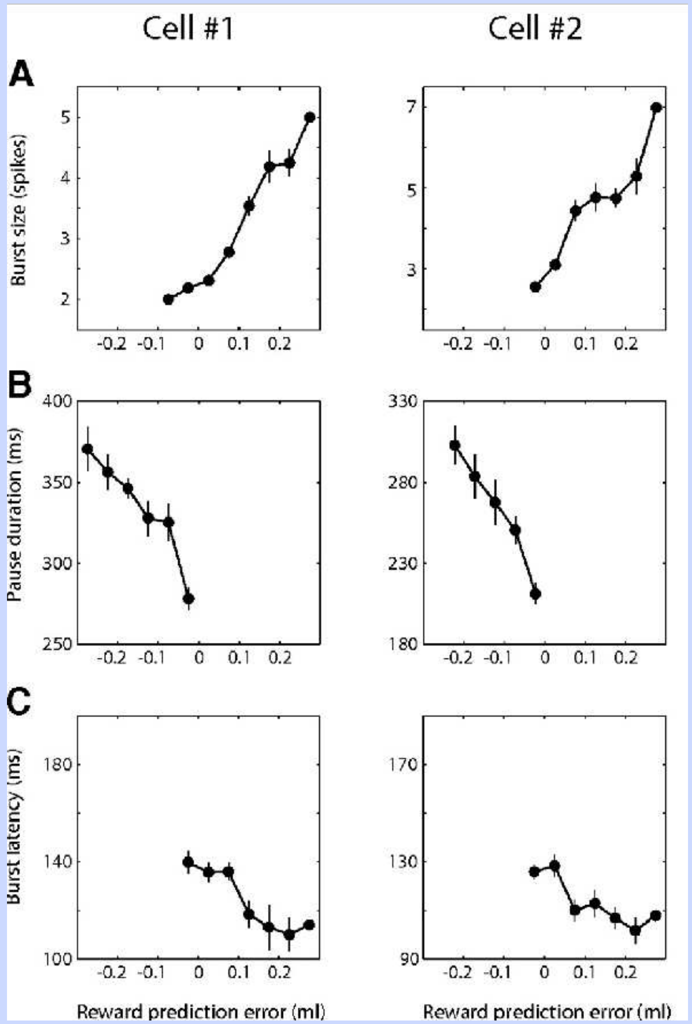
These are prediction errors too!
$[r_t + EV_{t+1}] - EV_t$



no reward following 100% predictive cue

Fiorillo et al 2003

Burst/Pause correlations with Rew Prediction Errors



Bayer et al, 2007 JNeurophys

Temporal Difference Learning: Equations

Value function, sum of discounted future rewards:

$$V(t) = \langle \gamma^0 r(t) + \gamma^1 r(t+1) + \gamma^2 r(t+2) \ldots \rangle \tag{1}$$

Temporal Difference Learning: Equations

Value function, sum of discounted future rewards:

$$V(t) = \langle \gamma^0 r(t) + \gamma^1 r(t+1) + \gamma^2 r(t+2) \ldots \rangle \tag{1}$$

Recursive definition:

$$V(t) = \langle r(t) + \gamma V(t+1) \rangle \tag{2}$$

Temporal Difference Learning: Equations

Value function, sum of discounted future rewards:

$$V(t) = \langle \gamma^0 r(t) + \gamma^1 r(t+1) + \gamma^2 r(t+2) ... \rangle \tag{1}$$

Recursive definition:

$$V(t) = \langle r(t) + \gamma V(t+1) \rangle \tag{2}$$

Error in predicted reward (from previous to next time-step):

$$\delta(t) = \Big(r(t) + \gamma \hat{V}(t+1) \Big) - \hat{V}(t) \tag{3}$$

Temporal Difference Learning: Equations

Value function, sum of discounted future rewards:

$$V(t) = \langle \gamma^0 r(t) + \gamma^1 r(t+1) + \gamma^2 r(t+2) \ldots \rangle \tag{1}$$

Recursive definition:

$$V(t) = \langle r(t) + \gamma V(t+1) \rangle \tag{2}$$

Error in predicted reward (from previous to next time-step):

$$\delta(t) = \Big(r(t) + \gamma \widehat{V}(t+1) \Big) - \widehat{V}(t) \tag{3}$$

Update value estimate:

$$\widehat{V}(t) \leftarrow \widehat{V}(t) + \alpha \delta(t) \tag{4}$$

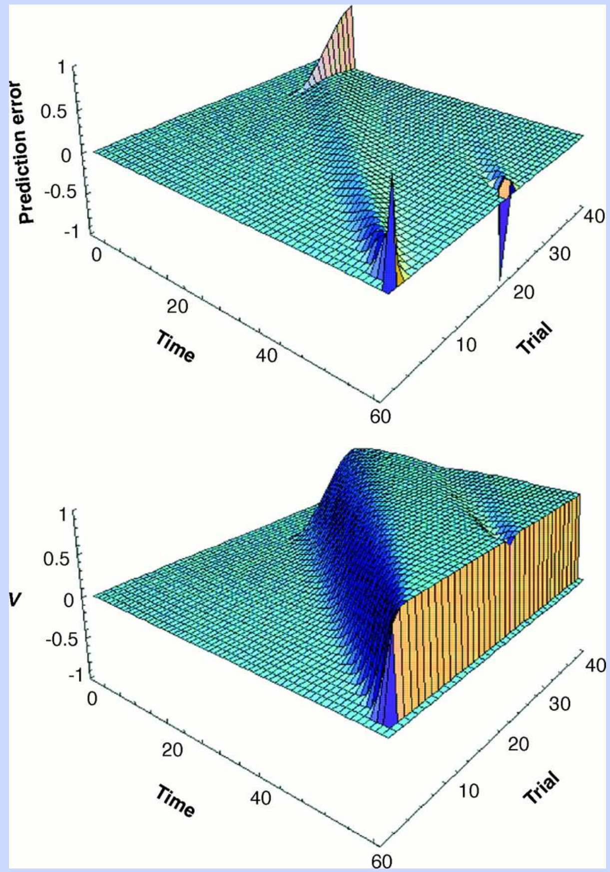
α = learning rate

TD and Dopamine Relationship

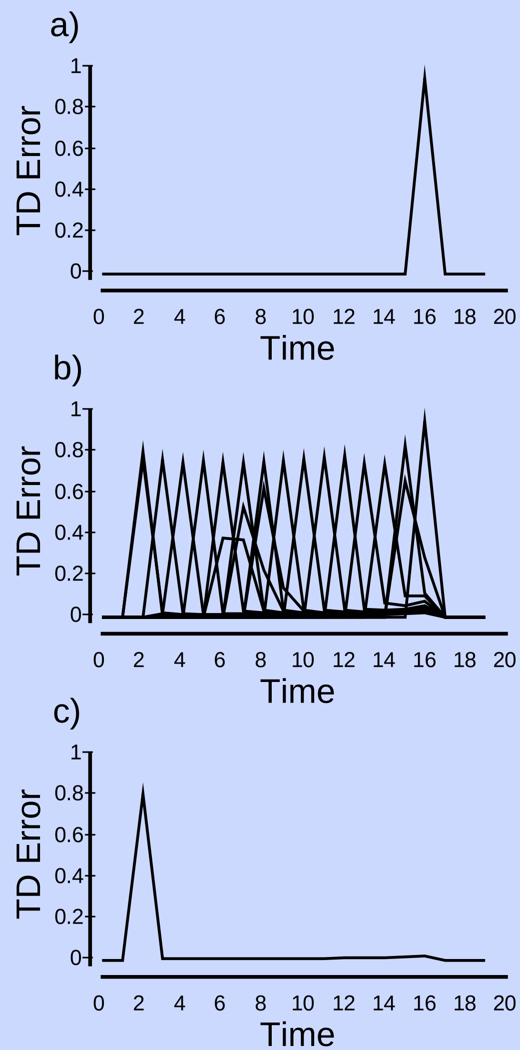
Schultz, Dayan & Montague, 1997, *Science*

δ

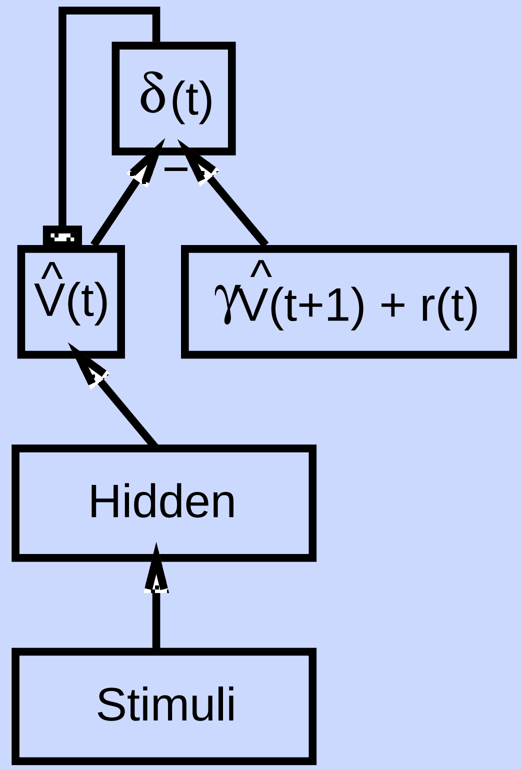
V



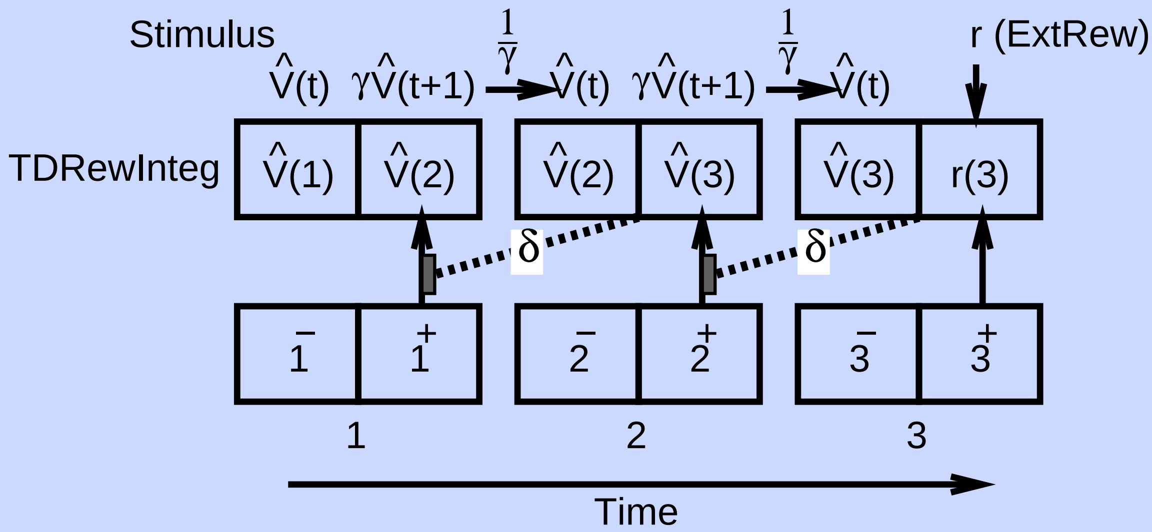
Model: CS at t=2, US at t=16



Network Implementation



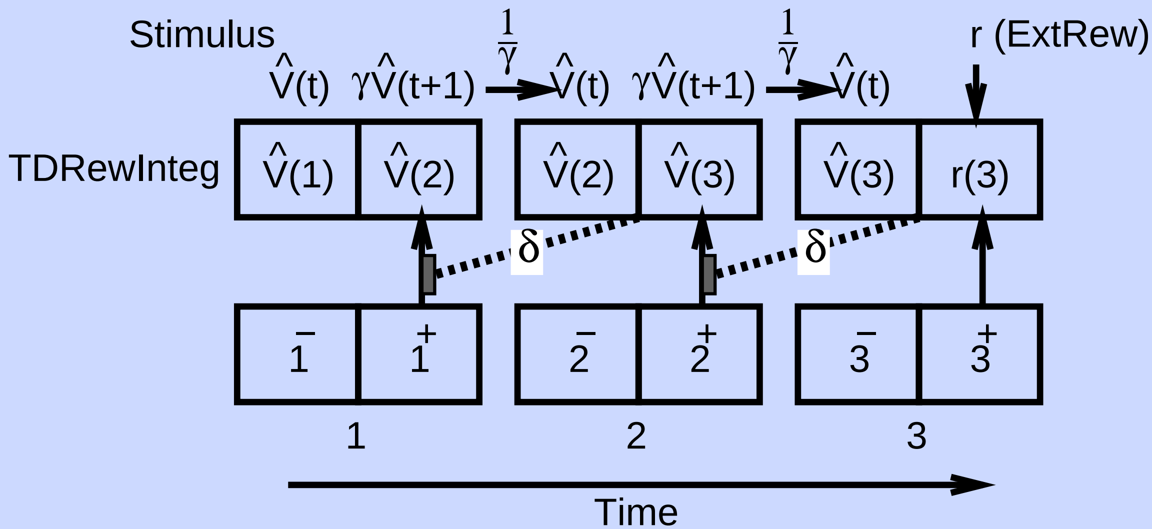
Phase-based Implementation



TDRewInteg = TDRewPred + ExtRew

Minus phase: TDRewInteg clamped to prev plus phase value.

Phase-based Implementation

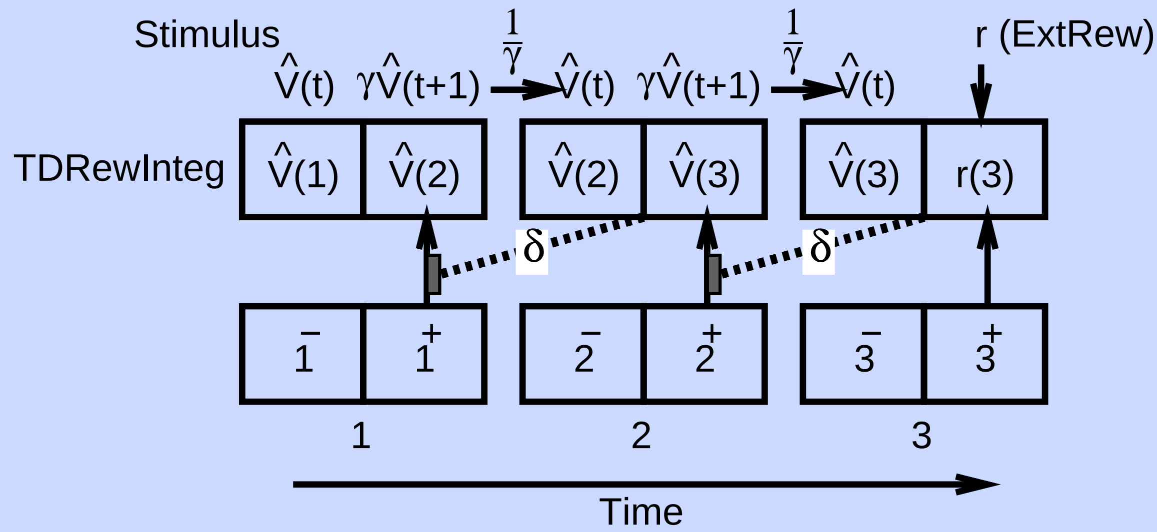


TDRewInteg = TDRewPred + ExtRew

Minus phase: TDRewInteg clamped to prev plus phase value.

Plus phase: TDRewInteg settles via weights
= expected reward at t+1, plus any ExtRew at time t.

Phase-based Implementation



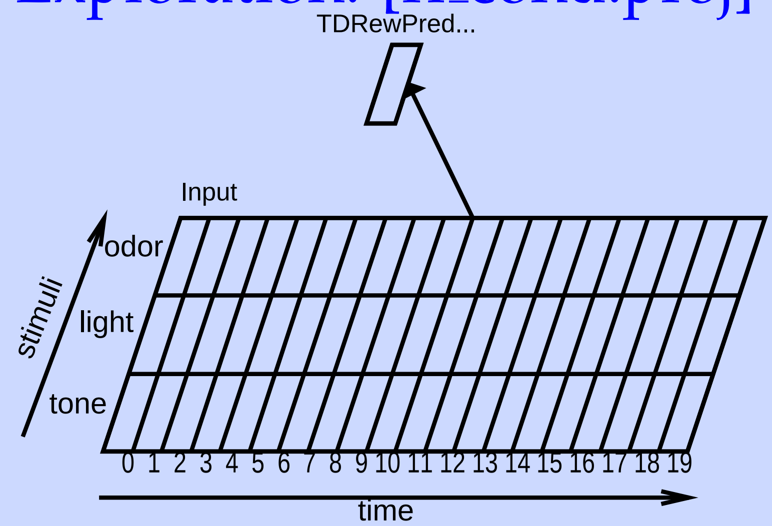
TDRewInteg = TDRewPred + ExtRew

Minus phase: TDRewInteg clamped to prev plus phase value.

Plus phase: TDRewInteg settles via weights
= expected reward at t+1, plus any ExtRew at time t.

Learning signal δ (= "TD") trains prediction for *previous* time step.
(*eligibility traces* needed)

Exploration: [rl_cond.proj]



’Complete Serial Compound’ (CSC) input representation:
unique unit for each stimulus at each time point
(used in Sutton & Barto, Montague et al, etc)

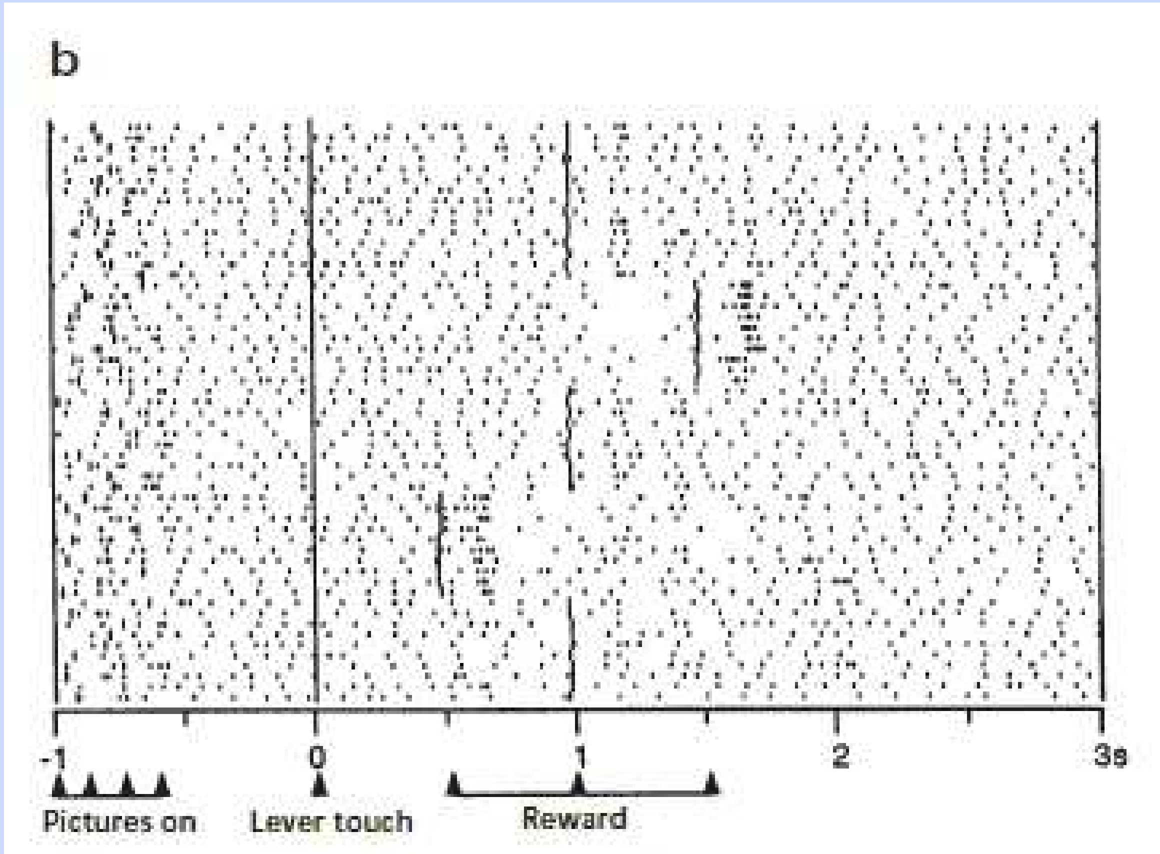
Not realistic, but good for demonstration. This assumption can be relaxed without changing core ideas (e.g. Ludvig et al, 2008).

Exploration: [rl_cond.proj]

Standard TD: $\hat{V}(t) = \sum_i w_i x_i(t)$ [x_i are inputs: tone, light]

Here: passed thru activation function – has to surpass threshold, subject to inhibitory competition from other value reps

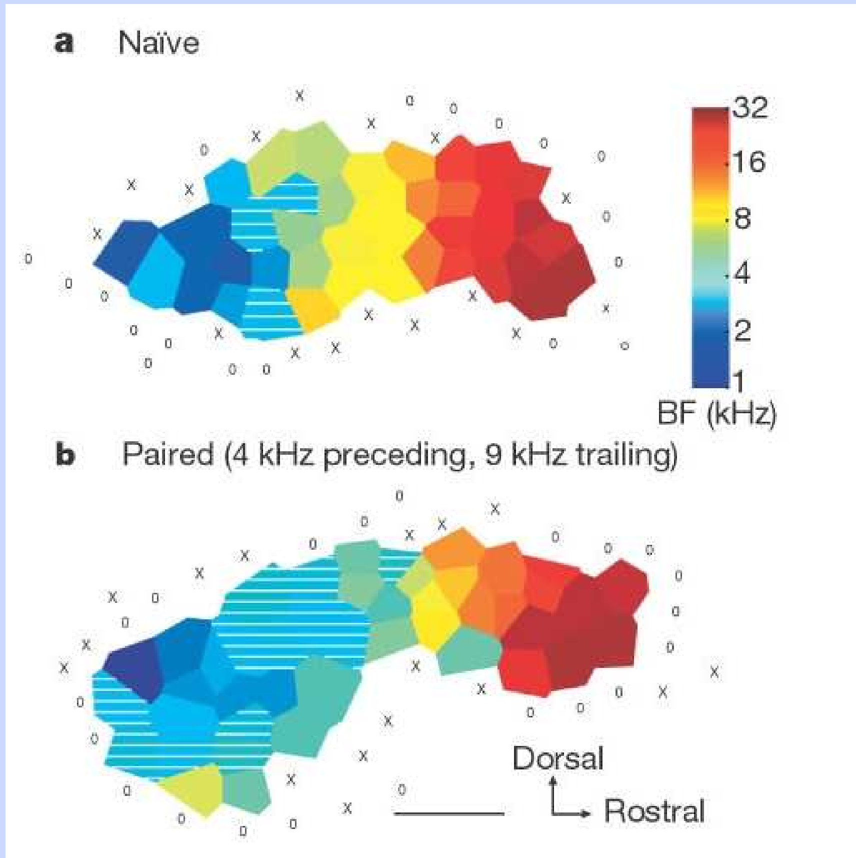
DA and Timing: Late and Early Rewards



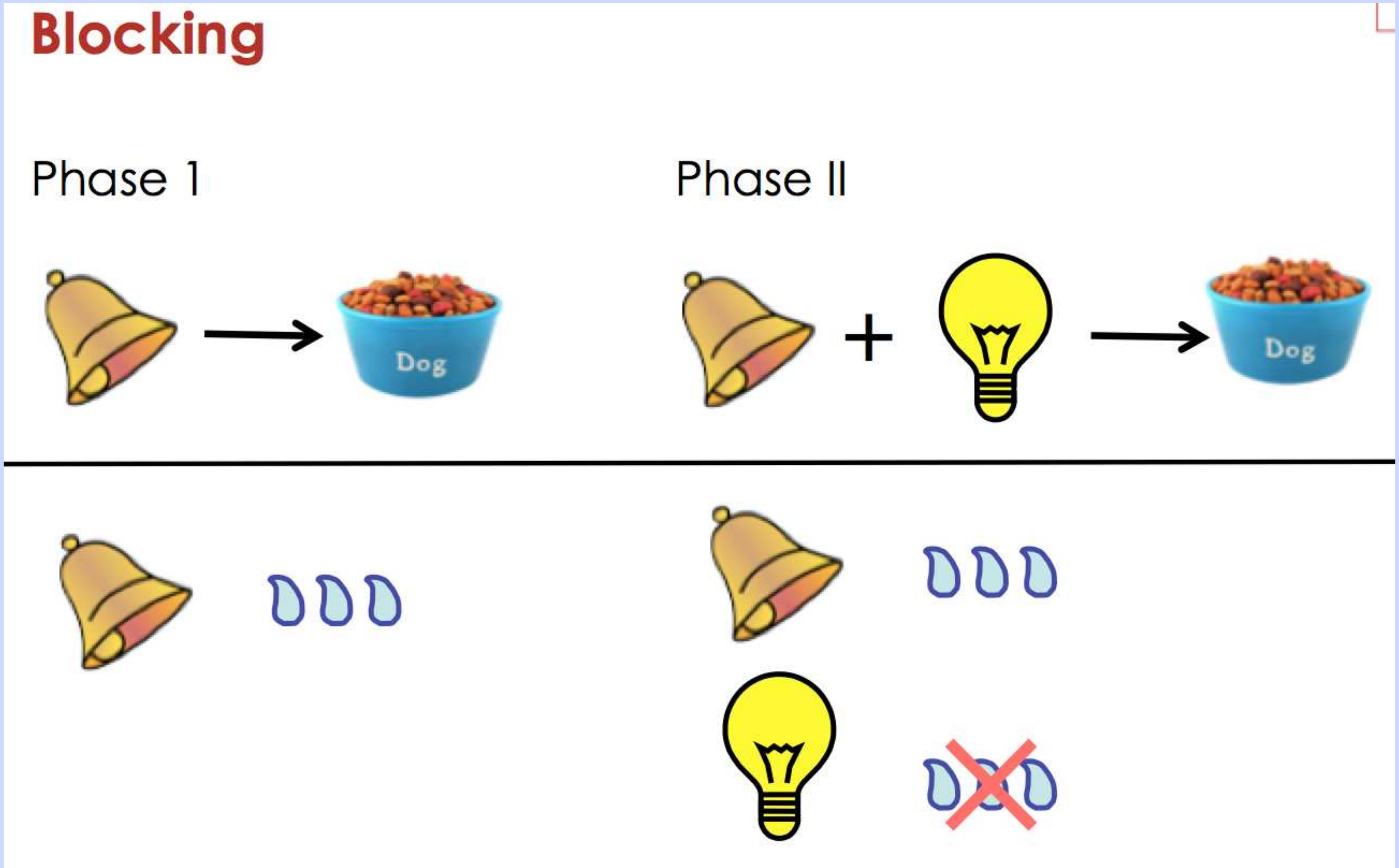
Hollerman & Schultz 1998

DA and Learning: Auditory Cortex

Bao et al, 2001, *Nature*

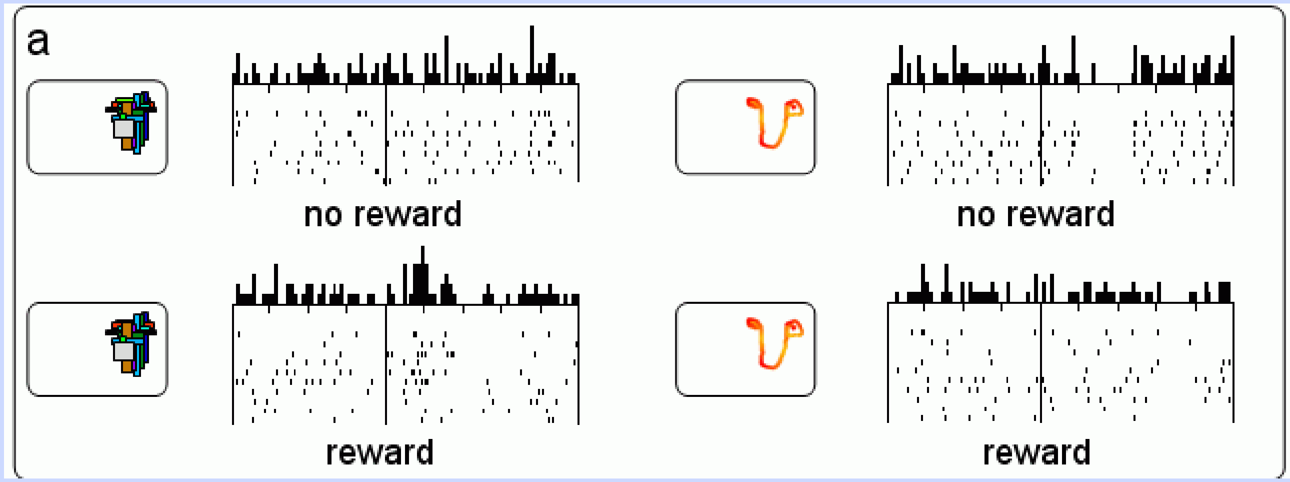


Learning Theory: Blocking (Behavior)



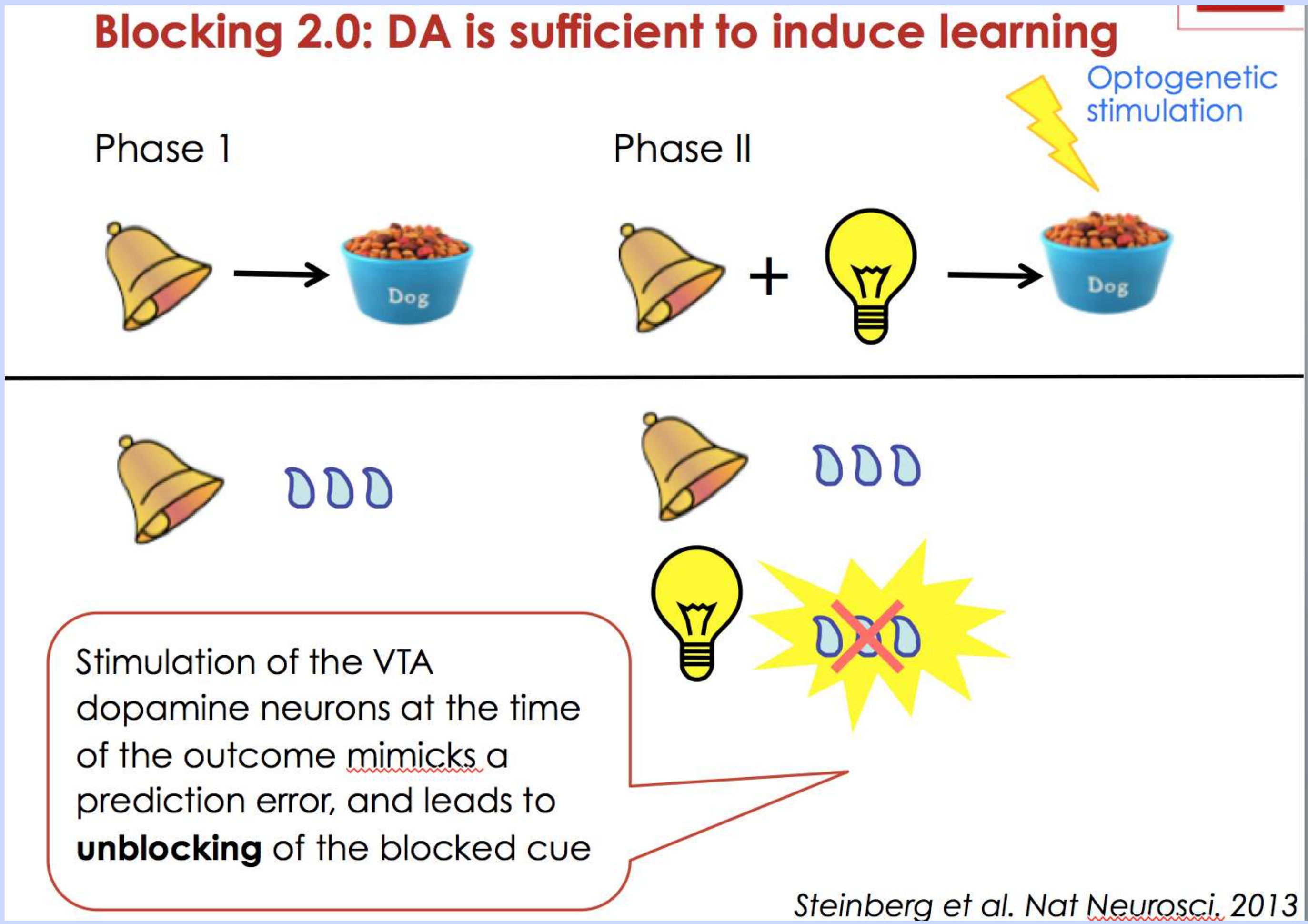
Learning Theory: Blocking (Dopamine)

Waelti et al, 2001, *Nature*



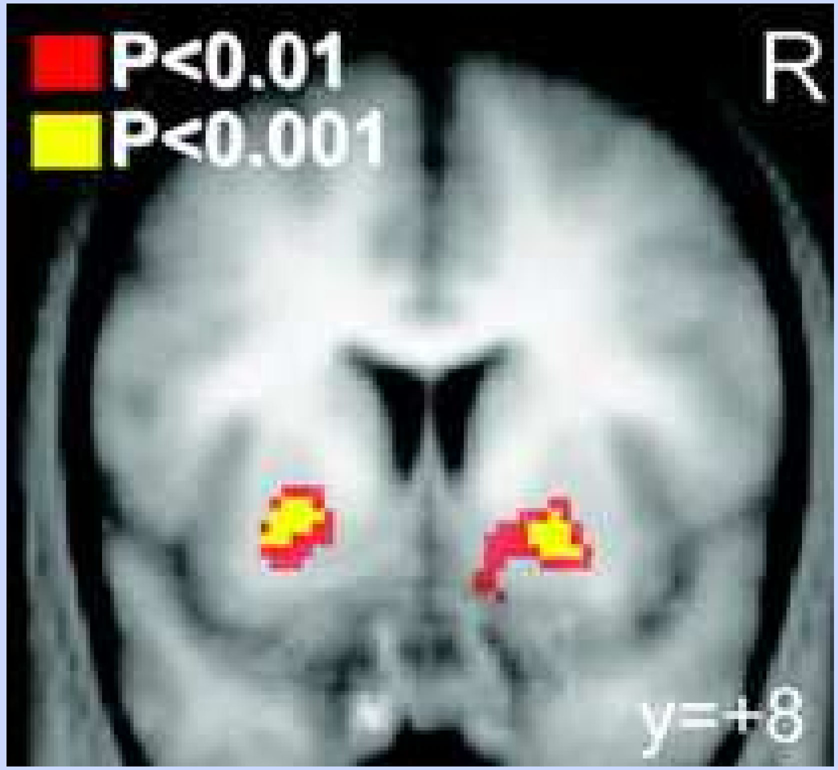
Blocked stimulus Control (not blocked) stim

Learning Theory: (Un)Blocking



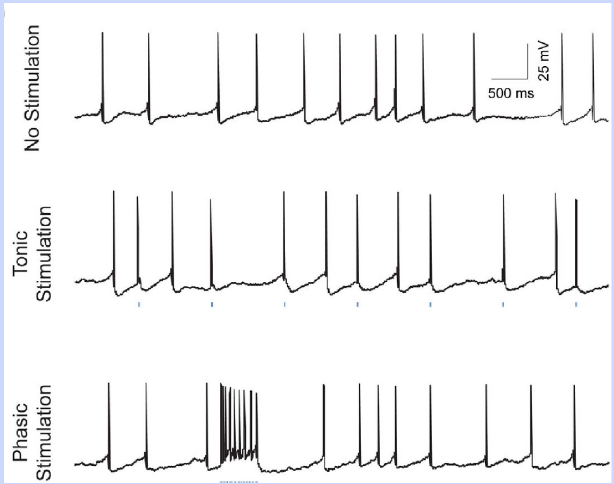
TD prediction error and human functional imaging

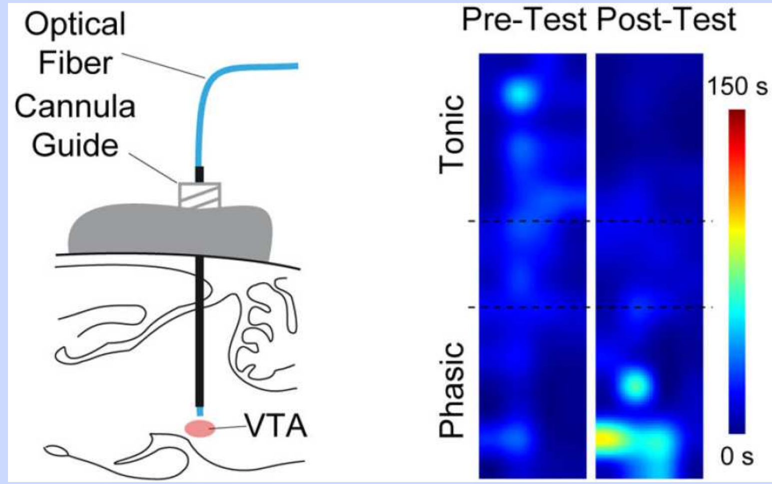
O'Doherty et al, 2004, *Science*



Ventral striatum = DA enriched, correlates with TD PE=Critic!

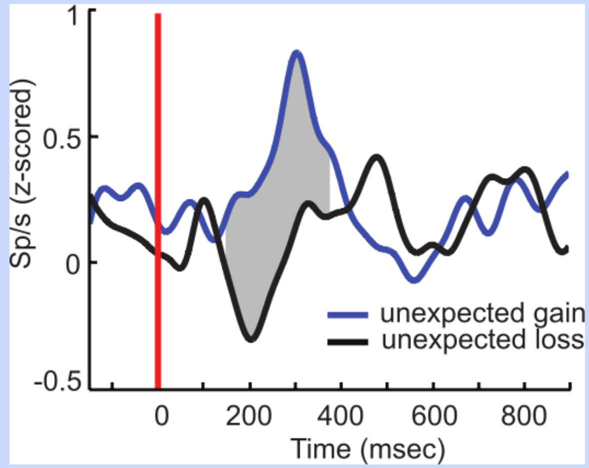
Optical phasic DA stimulation causally induces conditioning





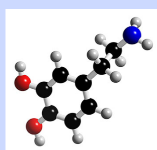
(Tsai et al, 2009, Science)

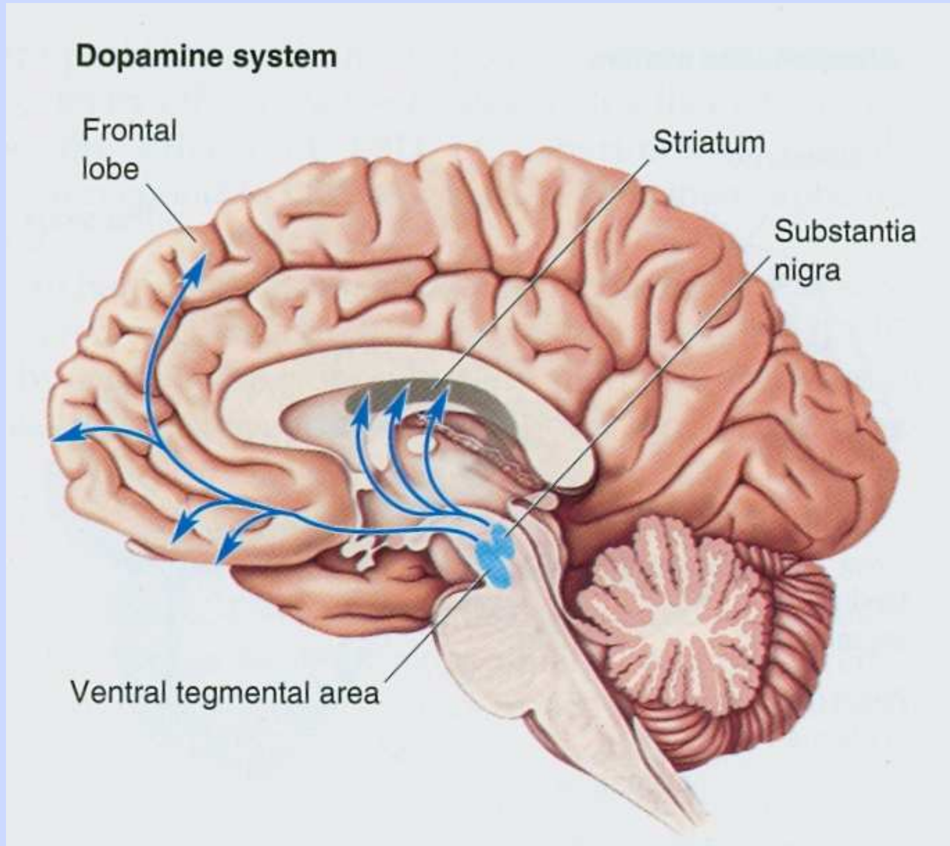
DA neuron spiking during reinforcement task in humans



(Zaghoul et al, 2009, Science)

How are dopamine-based RPE signals used to select *actions*?





Will consider biological implementation in basal ganglia later

Q learning: extending prediction error learning to actions

Error in predicted reward:

$$\delta_t = \left(r_t + \gamma \max_a Q_t(s_{t+1}, a)\right) - Q_t(s, a)$$

Update value estimate:

$$Q_t(s, a) \leftarrow Q_t(s, a) + \alpha\delta(t)$$

Select among Q values:

$$P_t(a) = \frac{e^{\frac{Q_t(s,a)}{\beta}}}{\sum_{i=1}^{n} e^{\frac{Q_t(s,i)}{\beta}}}$$

γ = discount, α = learning rate, β = “temperature” / exploration parameter

Watkins & Dayan 1992

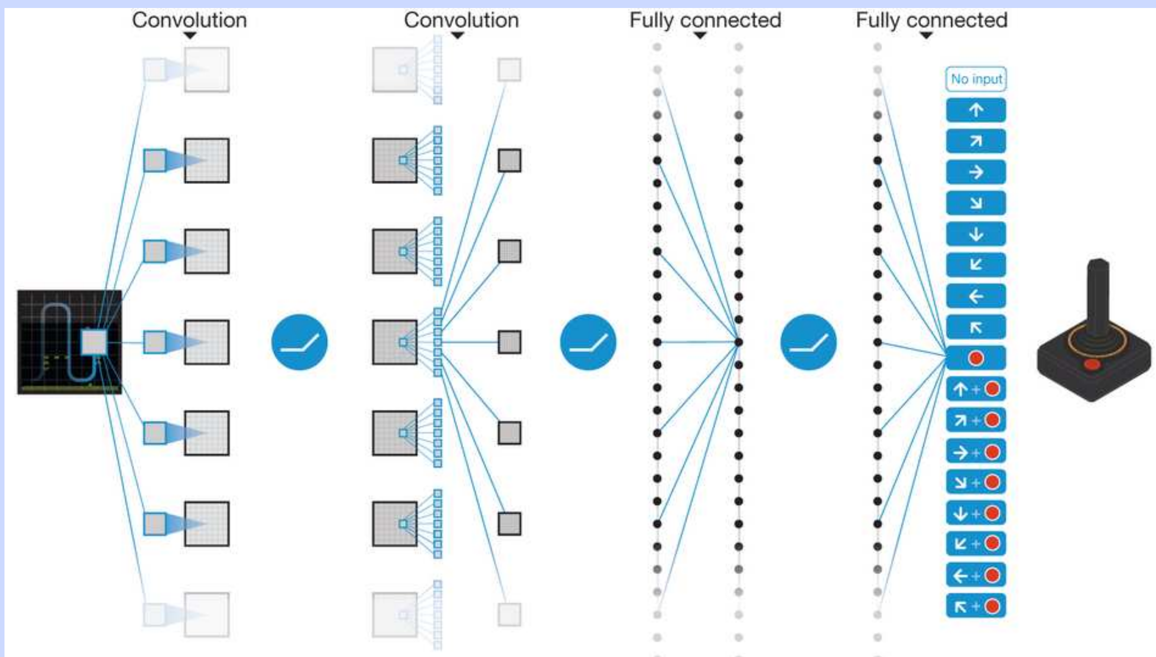
Google Deep Mind RL Network ("DQN") Plays Atari

LETTER

doi:10.1038/nature14236

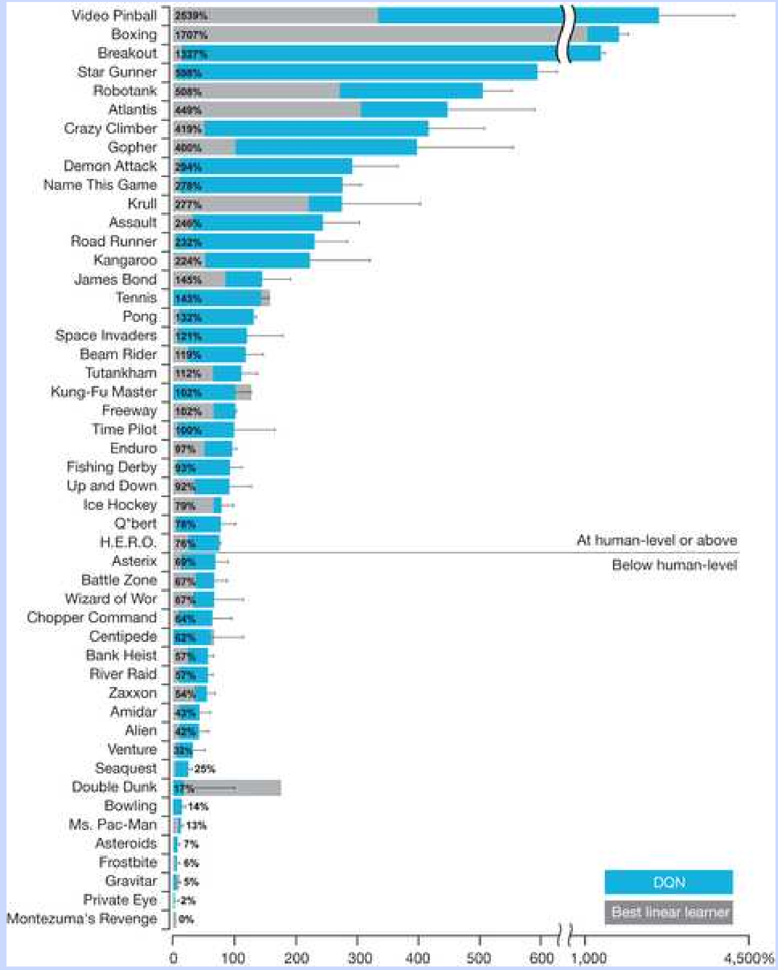
Human-level control through deep reinforcement learning

Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]



for video, DQN_spaceinvaders.mov

Google Deep Mind RL Network ("DQN") Plays Atari



Extra

The following slides describe a recently developed alternative to TD, called PVLV, which we think is more biologically plausible and computationally powerful. This material is optional for the course.

The Problem

Q: How do we learn to attach positive/negative valence to environmental stimuli?

The Problem

Q: How do we learn to attach positive/negative valence to environmental stimuli?

A: The same way we learn lots of other stuff: the Delta Rule!

$$\delta_{pv} = r - \widehat{V}_{pv}$$

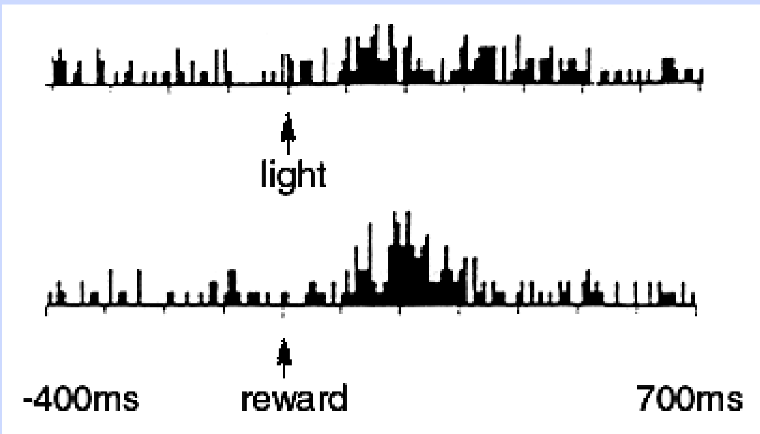
$\widehat{V}_{pv}$: expected reward based on prior associations
r: reward
δ_{pv}: learning signal

The Problem

Q: But what happens when environmental stimulus occurs *before* reward?

Basic Data: VTA DA Neural Firing in Conditioning

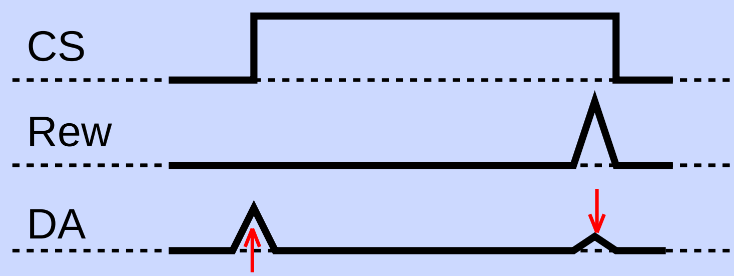
Before:



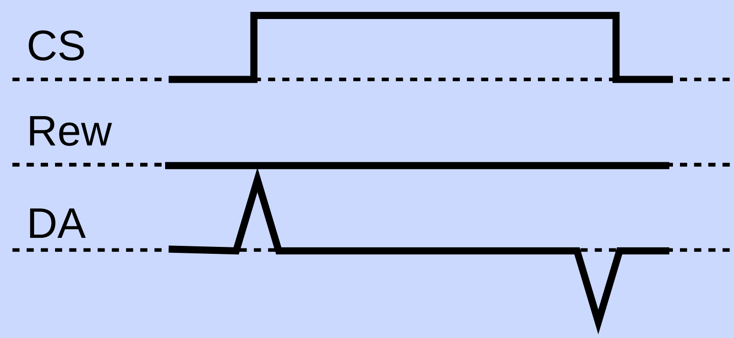
After:

Basic Data: VTA DA Neural Firing in Conditioning

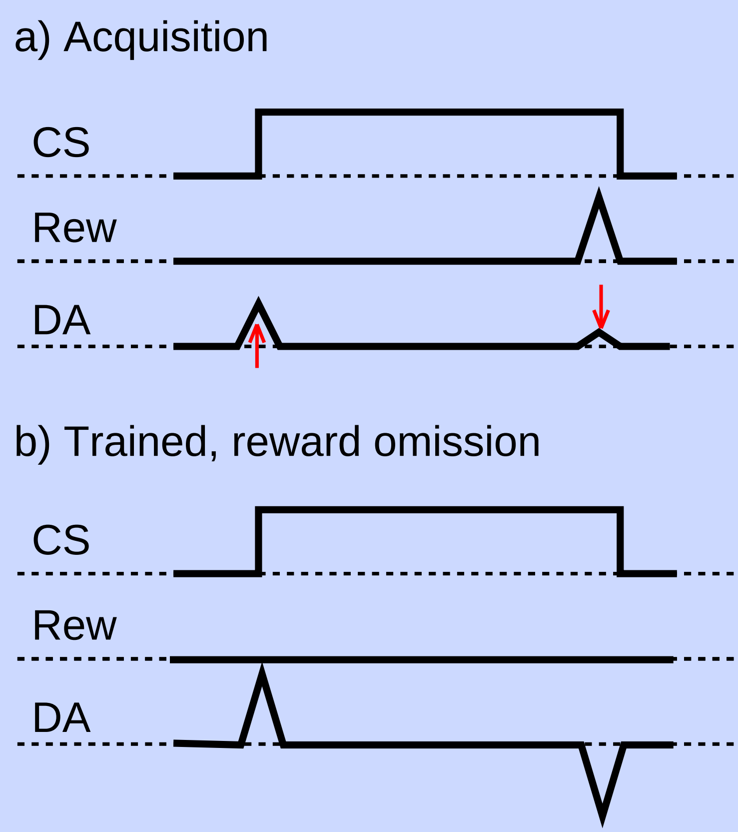
a) Acquisition



b) Trained, reward omission

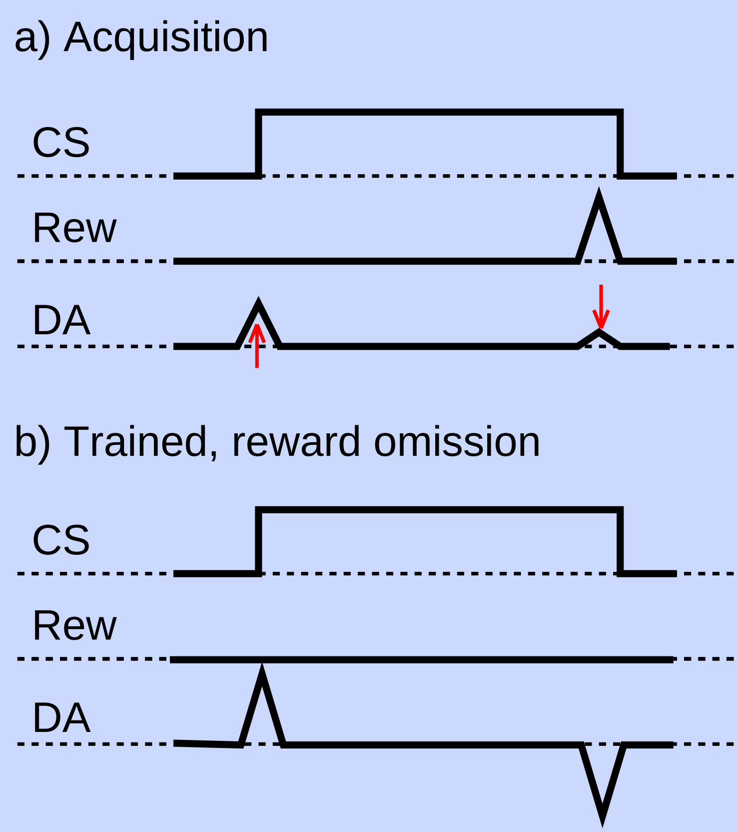


Basic Data: VTA DA Neural Firing in Conditioning



Dopamine spikes/dips are learning signals

Basic Data: VTA DA Neural Firing in Conditioning



Dopamine spikes/dips are learning signals

Delta rule fails to account for predictive DA spike!

Standard Approach: TD

Predict all future rewards (discounted):

$V_t = \sum_{\tau=t+1}^{\tau=\infty} \gamma^{\tau-(t+1)} r_\tau$

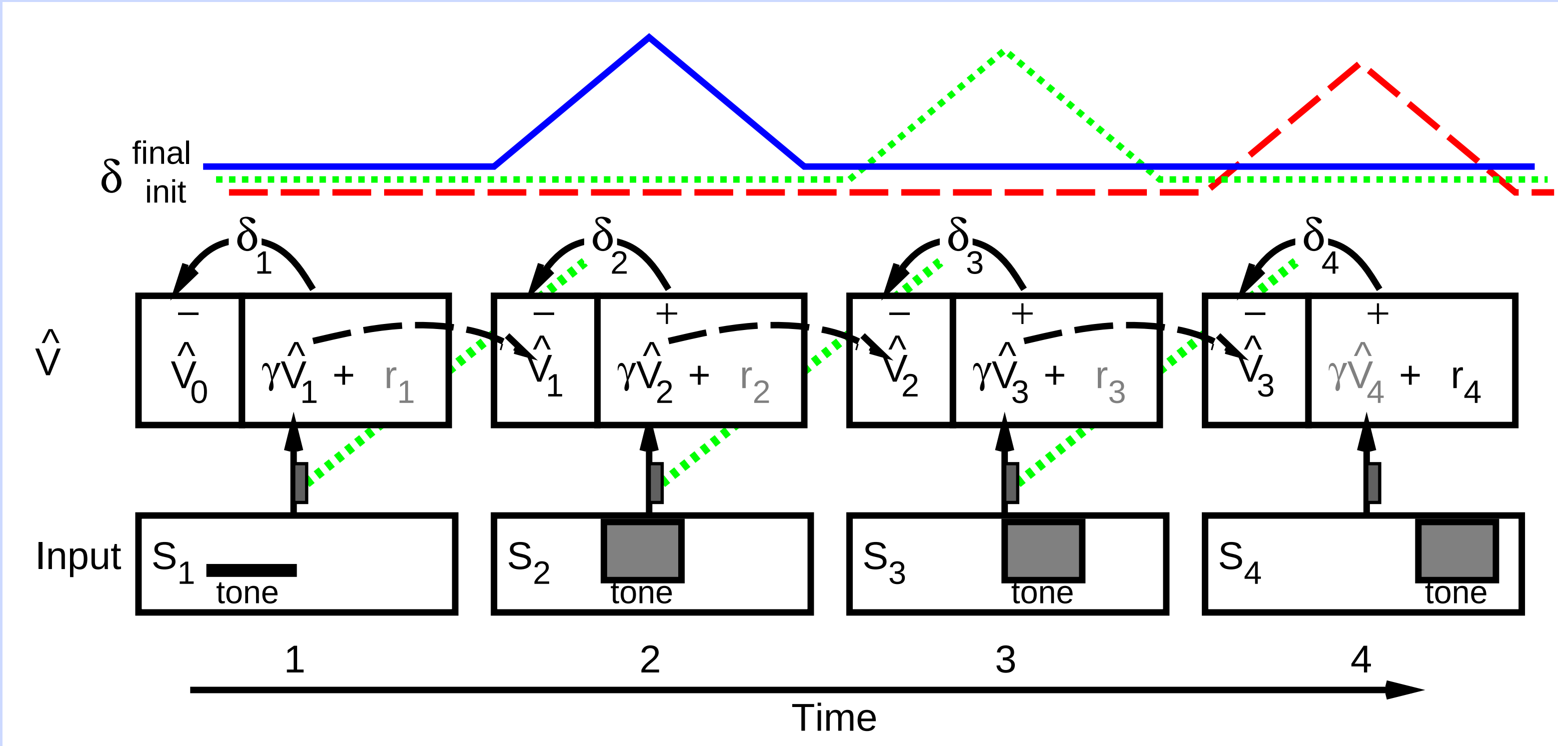
Recursively:

$\widehat{V}_{t-1} = r_t + \gamma \widehat{V}_t$

Error = Temporal Difference = TD:

$DA = \delta_t = [r_t + \gamma \widehat{V}_t] - \widehat{V}_{t-1}$

TD Illustrated

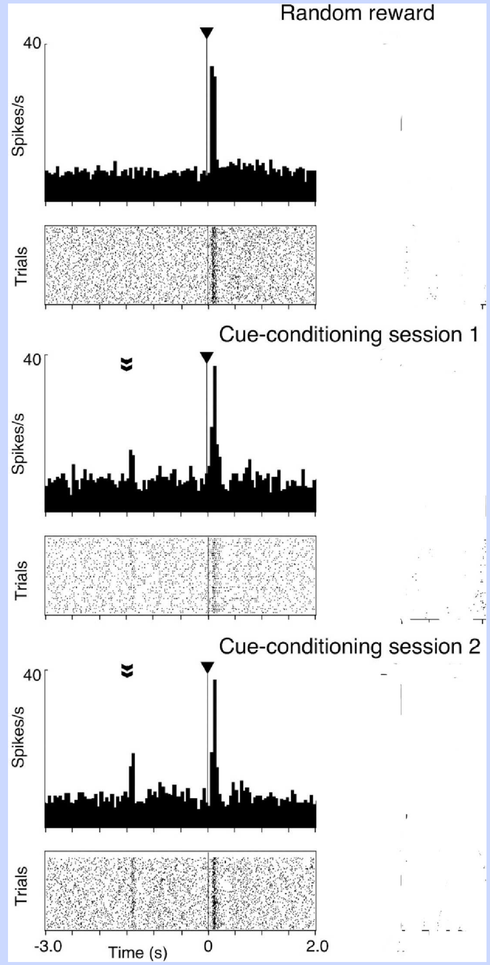


Problems with TD

- Great algorithm, developed in computer science / machine learning, but is this actually what the brain does?
- Even if so, doesn't specify how these signals are computed by systems upstream of DA... just predicts DA and δ but says nothing about V, etc.
- Current reward value is always relative to what happened just before. Too much temporal dependency?
- Chaining not seen in neural recordings.
- What determines "discount factor" γ, biologically?

Rat study: simultaneous CS and US DA spike

Pan et al, 2005, *Journal of Neuroscience*



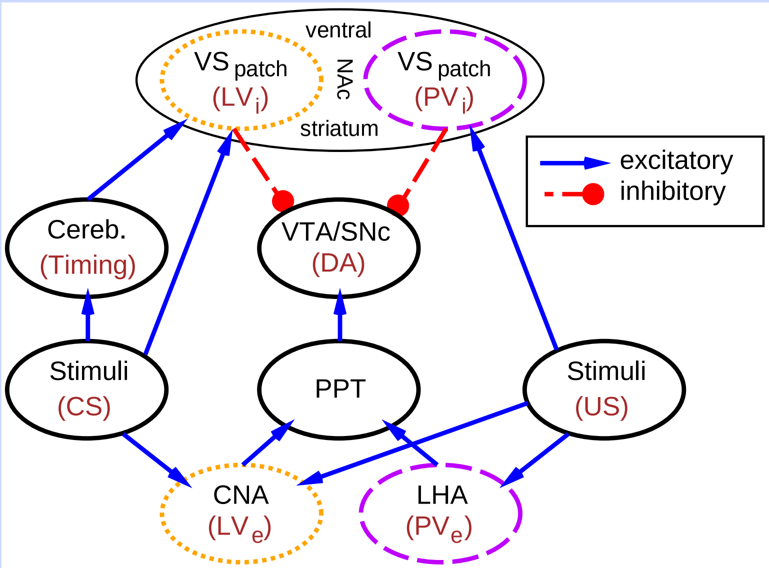
Inconsistent with standard TD!

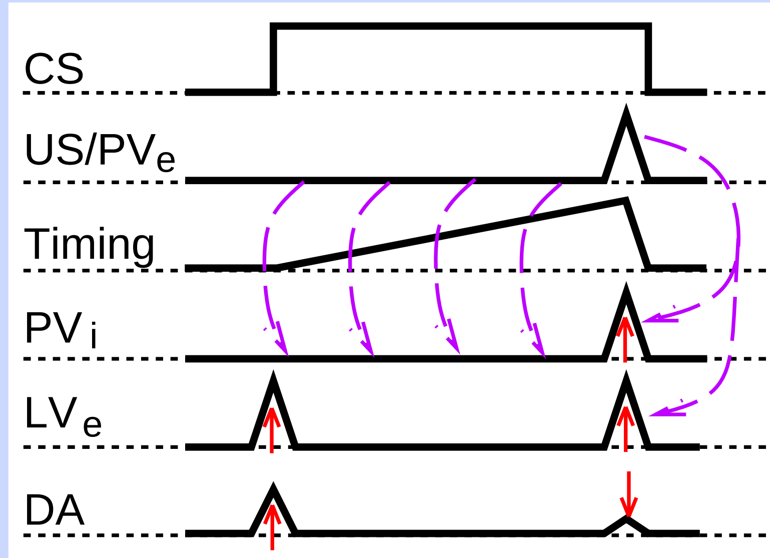
The PVLV Alternative

PVLV = Primary Value, Learned Value
(O'Reilly, Frank, Hazy & Watz, 2007, Behav Neurosci)

- No reward predictions, just *associations*!
- No temporal dependencies: DA depends only on current state.
- Uses same basic delta-rule learning as TD (Rescorla-Wagner).

PVLV: Two Separate Mechanisms (PV, LV)





- PV (Primary Value): Primary rewards (US), canceled.
- LV (Learned Value): Learned associations (CS $\rightarrow$ DA).

PVLV: Two Separate Mechanisms (PV, LV)

PV: Primary Value

- Trained at each point in time on actual reward value present:
  $\delta_t = r_t - \widehat{V}_t$

PVLV: Two Separate Mechanisms (PV, LV)

PV: Primary Value

- Trained at each point in time on actual reward value present: $\delta_t = r_t - \hat{V}_t$
- This uses *immediate* prediction ($\hat{V}_t$) of current rew value (r_t)

PVLV: Two Separate Mechanisms (PV, LV)

PV: Primary Value

- Trained at each point in time on actual reward value present: $\delta_t = r_t - \widehat{V}_t$
- This uses *immediate* prediction ($\widehat{V}_t$) of current rew value (r_t)
- Accounts for canceling of DA spike @ rew, and DA dips when no rew received.

PVLV: Two Separate Mechanisms (PV, LV)

PV: Primary Value

- Trained at each point in time on actual reward value present:
  $\delta_t = r_t - \hat{V}_t$
- This uses *immediate* prediction ($\hat{V}_t$) of current rew value (r_t)
- Accounts for canceling of DA spike @ rew, and DA dips when no rew received.
- But this doesn't account for predictive DA spikes... (actually results in predictive DA dips!)

PVLV: Two Separate Mechanisms (PV, LV)

LV: Learned value

- Represents perceived values of stims **even when there is no current rew expectation**.

PVLV: Two Separate Mechanisms (PV, LV)

LV: Learned value

- Represents perceived values of stims **even when there is no current rew expectation**.

- Only gets training signal @ rew, or when PV expects some rew. (ie learning is *filtered* by primary PV system.)

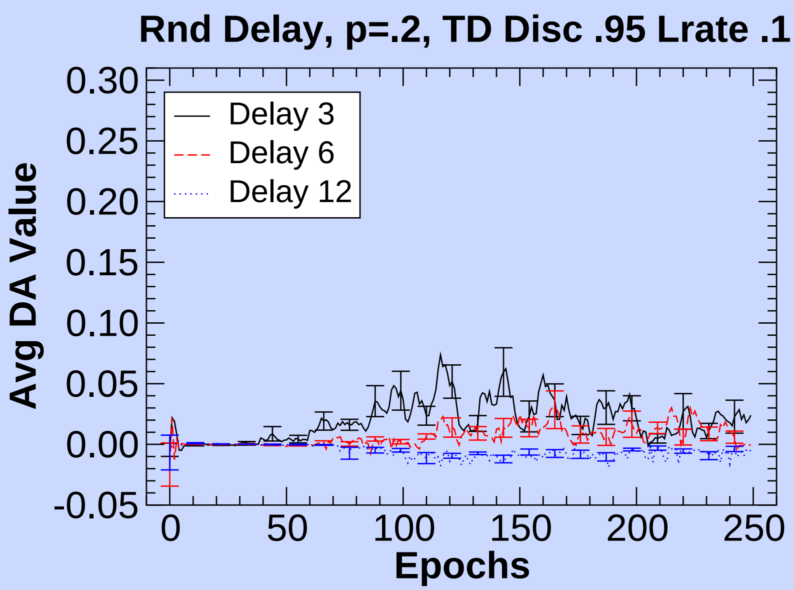
PVLV: Two Separate Mechanisms (PV, LV)

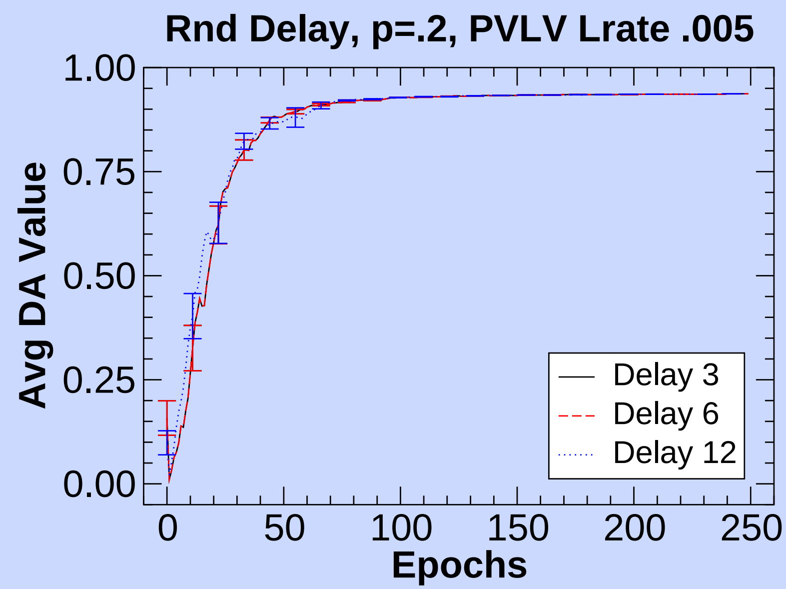
LV: Learned value

- Represents perceived values of stims **even when there is no current rew expectation**.
- Only gets training signal @ rew, or when PV expects some rew. (ie learning is *filtered* by primary PV system.)
- → Learns at time of rew, but not at CS onset.
- → Generalizes rew values to CS...
- → Accounts for DA spikes for stimuli that have previously been associated with reward!

PVLV: Computationally Powerful

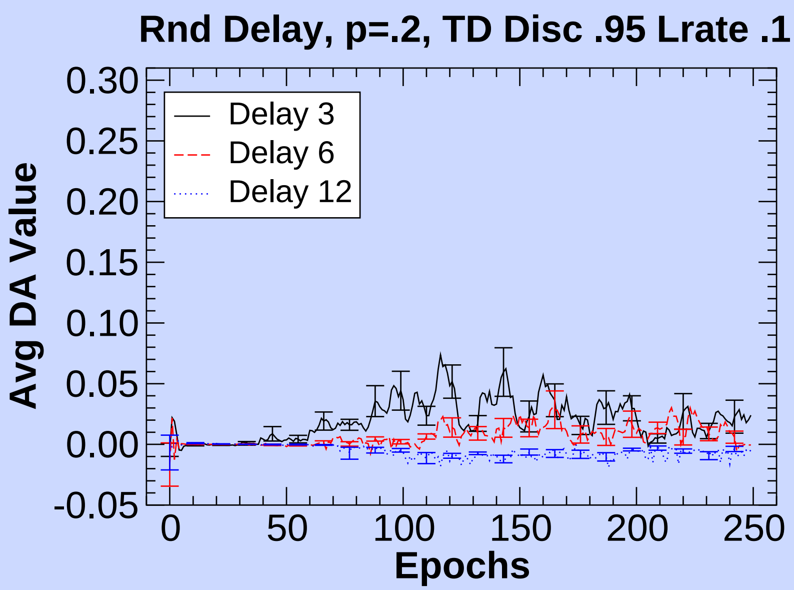
Comparison with TD on Random Delays (breaks TD chaining):

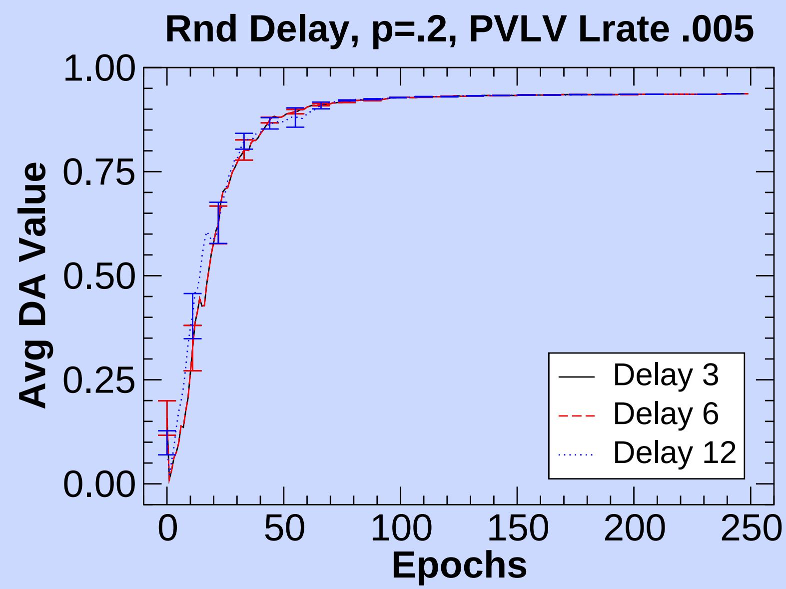




PVLV: Computationally Powerful

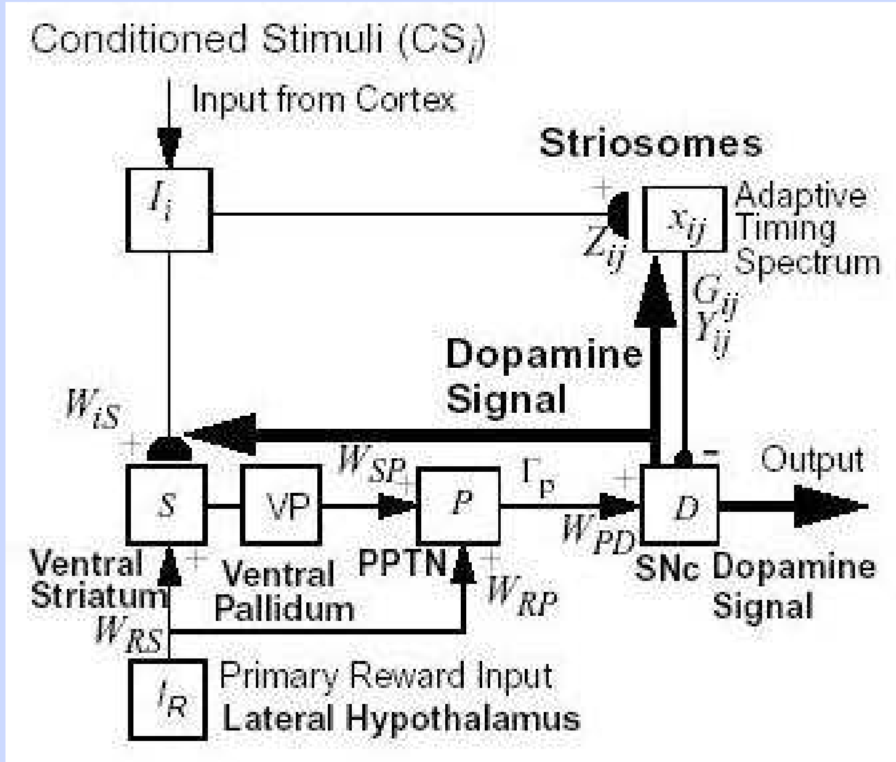
Comparison with TD on Random Delays (breaks TD chaining):





Enables working memory model to learn complex WM tasks.

Similar to Brown, Bullock & Grossberg, ’99



Diffs: Anatomical (CNA vs. VS; Dorsal vs. Ventral Patch)
Functional (intrinsic timing? LV system cannot train itself).

PVLV accounts for timing data better than TD!

- Data: during transient learning period, both rews and CS elicit activation.
- This accounted for by PV, LV systems operating in parallel.
- TD: predicts chaining back in time from rew to CS.

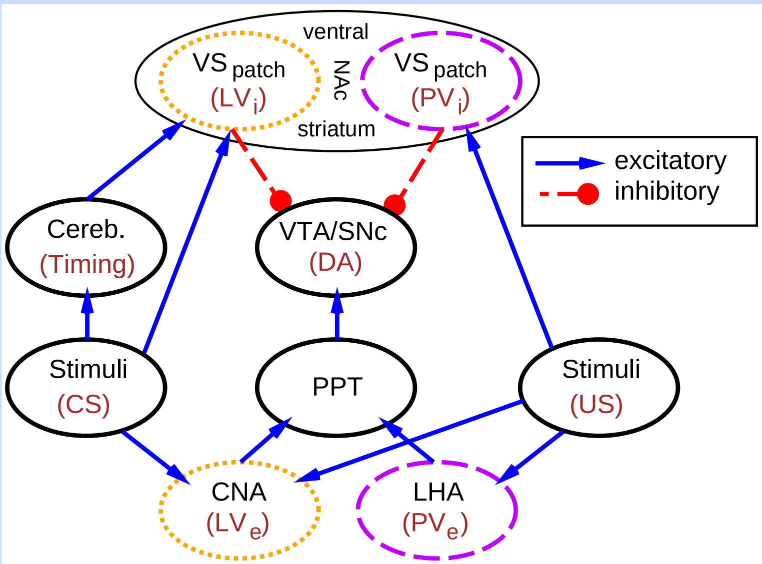
PVLV accounts for timing data better than TD!

- Data: delayed rewards cause dips @ usual time, then spikes
- This accounted for by both TD and PV.

PVLV accounts for timing data better than TD!

- Data: delayed rewards cause dips @ usual time, then spikes
- This accounted for by both TD and PV.
- Data: early rewards cause spikes, then dips @ usual time
- This accounted for by PV (spike), PV (dip), but TD only accounts for spike.

More Key Predictions from PVLV



- CNA = Pavlovian conditioning (e.g., Killcross et al. '97).
- NAc (patch/shell) = Extinction (Ferry et al. '00; Annett et al., 89), Blocking (data?).
- NAc (matrix/core) = Basic actions (OR's, approach, avoid).
- CNA can't train itself: No 2nd order conditioning!
- BLA = 2nd order cond, uses DA-independent mechanisms (CNA/BLA double-dissoc).

Conclusions

PVLV provides computationally motivated architecture that seems to fit with biology & behavioral data.

These learning mechanisms enable *arbitrary* stimuli/goals to be plugged into our fixed set of built-in motivational drives.

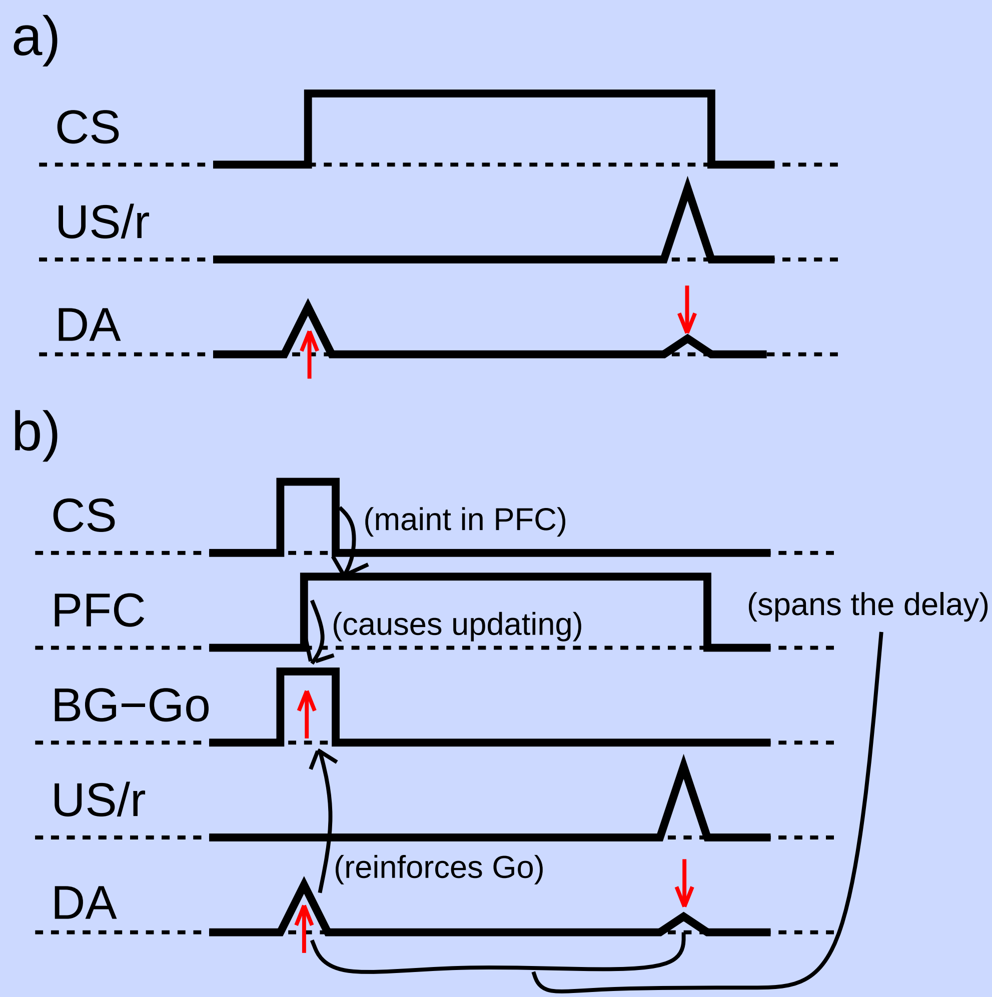
Conclusions

PVLV provides computationally motivated architecture that seems to fit with biology & behavioral data.

These learning mechanisms enable *arbitrary* stimuli/goals to be plugged into our fixed set of built-in motivational drives.

Something motivates every generated mental-state, always!

PVLV, WM, and DA



PVLV: Two Separate Mechanisms (PV, LV)

PV learning:

$\delta_{pv} = r - \widehat{V}_{pv}$ – or – $\delta_{pv} = PV_e - PV_i$

$\Delta w_i = \epsilon x_i \delta_{pv}$

PVLV: Two Separate Mechanisms (PV, LV)

PV learning:

$\delta_{pv} = r - \widehat{V}_{pv}$ – or – $\delta_{pv} = PV_e - PV_i$

$$\Delta w_i = \epsilon x_i \delta_{pv}$$

LV learning (filtered by PV):

$$\Delta w_i = \begin{cases} \epsilon(r_t - \widehat{V}_{lv})x_i & \text{if } \widehat{V}_{pv} > \theta_{pv} \text{ or } r_t > 0 \\ 0 & \text{otherwise} \end{cases}$$

PVLV: Two Separate Mechanisms (PV, LV)

PV learning:

$\delta_{pv} = r - \widehat{V}_{pv}$ – or – $\delta_{pv} = PV_e - PV_i$

$$\Delta w_i = \epsilon x_i \delta_{pv}$$

LV learning (filtered by PV):

$$\Delta w_i = \begin{cases} \epsilon(r_t - \widehat{V}_{lv})x_i & \text{if } \widehat{V}_{pv} > \theta_{pv} \text{ or } r_t > 0 \\ 0 & \text{otherwise} \end{cases}$$

Global DA (PV dominates):

$$\delta_t = \begin{cases} \delta_{pv} & \text{if } \widehat{V}_{pv} > \theta_{pv} \text{ or } r_t > 0 \\ \delta_{lv} & \text{otherwise} \end{cases}$$

$$\delta_{lv} = LV_e - LV_i$$

LV Extras

- DA spikes only observed @ CS onset, don't continue throughout delay until reward. Problem for PV?

LV Extras

- DA spikes only observed @ CS onset, don't continue throughout delay until reward. Problem for PV?

- Solution: PV system has synaptic depression, accommodates to constant sensory inputs; only perceives values of stims that were not present in last time step.

- This is also important for PFC learning.. (stay tuned)