

# Reinforcement Learning Lab

Odelia Schwartz,

2019

# Rescorla-Wagner rule (1972)

- Minimize difference between received reward and predicted reward
- Binary variable  $u$  (1 if stimulus is present; 0 if absent)
- Predicted reward  $v$
- Linear weight  $w$

$$v = wu$$

- If stimulus  $u$  is present:

$$v = w$$

based on Dayan and Abbott book

# Rescorla-Wagner rule (1972)

- Minimize squared error between received reward  $r$  and predicted reward  $v$ :

$$(r - v)^2$$

(average over presentations of stimulus and reward)

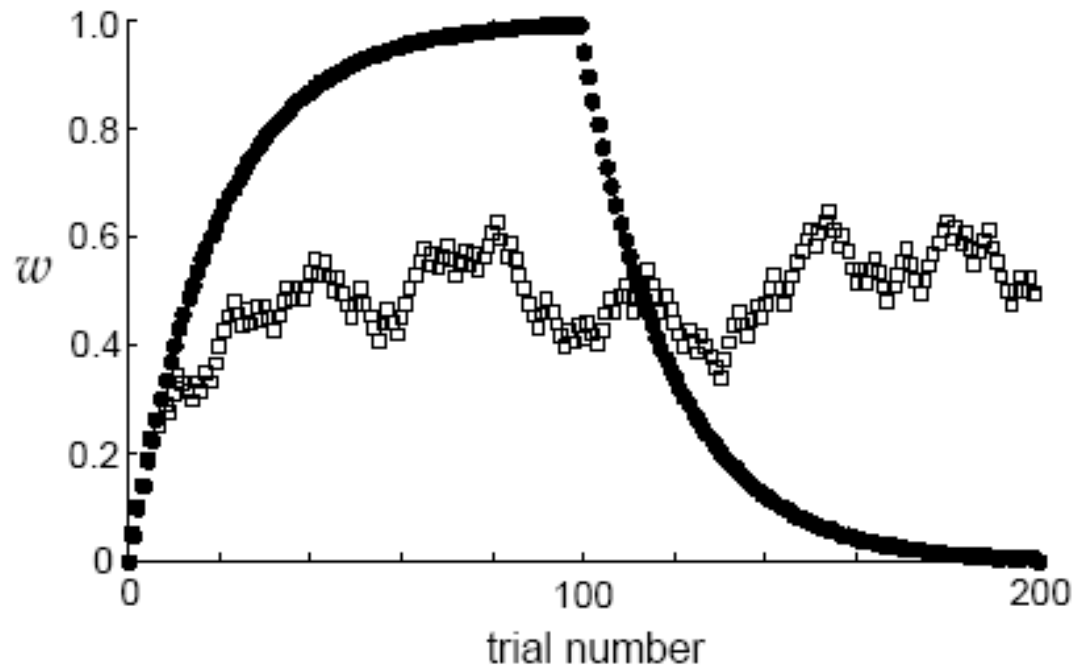
- Update weight:

$$w \rightarrow w + \epsilon(r - v)u$$

$\epsilon$  learning rate

Also known as delta learning rule:  $\delta = r - v$

# Acquisition and extinction



- Solid: First 100 trials: reward ( $r=1$ ) paired with stimulus; next 100 trials no reward ( $r=0$ ) paired with stimulus (learning rate .05)
- Dashed: Reward paired with stimulus randomly 50 percent of time

From Dayan and Abbott book

# Temporal Difference Learning

Want  $V_t = r_t + r_{t+1} + r_{t+2} + r_{t+3} \dots$

(here  $t$  represents time within a trial; reward can come at any time within a trial. Sutton and Barto interpret  $V_t$  as the **prediction of total future reward expected from time  $t$  onward until the end of the trial**)

Based on Dayan slides; Daw slides

# Temporal Difference Learning

$$\text{Want } V_t = r_t + r_{t+1} + r_{t+2} + r_{t+3} \dots$$

(here  $t$  represents time within a trial; reward can come at any time within a trial. Sutton and Barto interpret  $V_t$  as the **prediction of total future reward expected from time  $t$  onward until the end of the trial**)

Prediction error:

$$\delta_t = (r_t + r_{t+1} + r_{t+2} + r_{t+3} \dots) - V_t$$

# Temporal Difference Learning

Want  $V_t = r_t + r_{t+1} + r_{t+2} + r_{t+3} \dots$

(here  $t$  represents time within a trial)

But we don't want to wait forever for all future rewards...

$$r_{t+1}; r_{t+2}; r_{t+3} \dots$$

# Temporal Difference Learning

Want  $V_t = r_t + r_{t+1} + r_{t+2} + r_{t+3} \dots$

(here  $t$  represents time within a trial)

Recursion  
“trick”:

$$V_t = r_t + V_{t+1}$$

Based on Dayan slides; Daw slides



# Temporal Difference Learning

From recursion  
want:

$$v_t = r_t + v_{t+1}$$

Error:

$$\delta_t = r_t + v_{t+1} - v_t$$

# Temporal Difference Learning

From recursion  
want:

$$v_t = r_t + v_{t+1}$$

Error:

$$\delta_t = r_t + v_{t+1} - v_t$$

Update:

$$\begin{aligned} v_t &\rightarrow v_t + \varepsilon(r_t + v_{t+1} - v_t) \\ &= (1 - \varepsilon)v_t + \varepsilon(r_t + v_{t+1}) \end{aligned}$$

# RV versus TD

- Rescorla-Wagner error: (n represents trial)

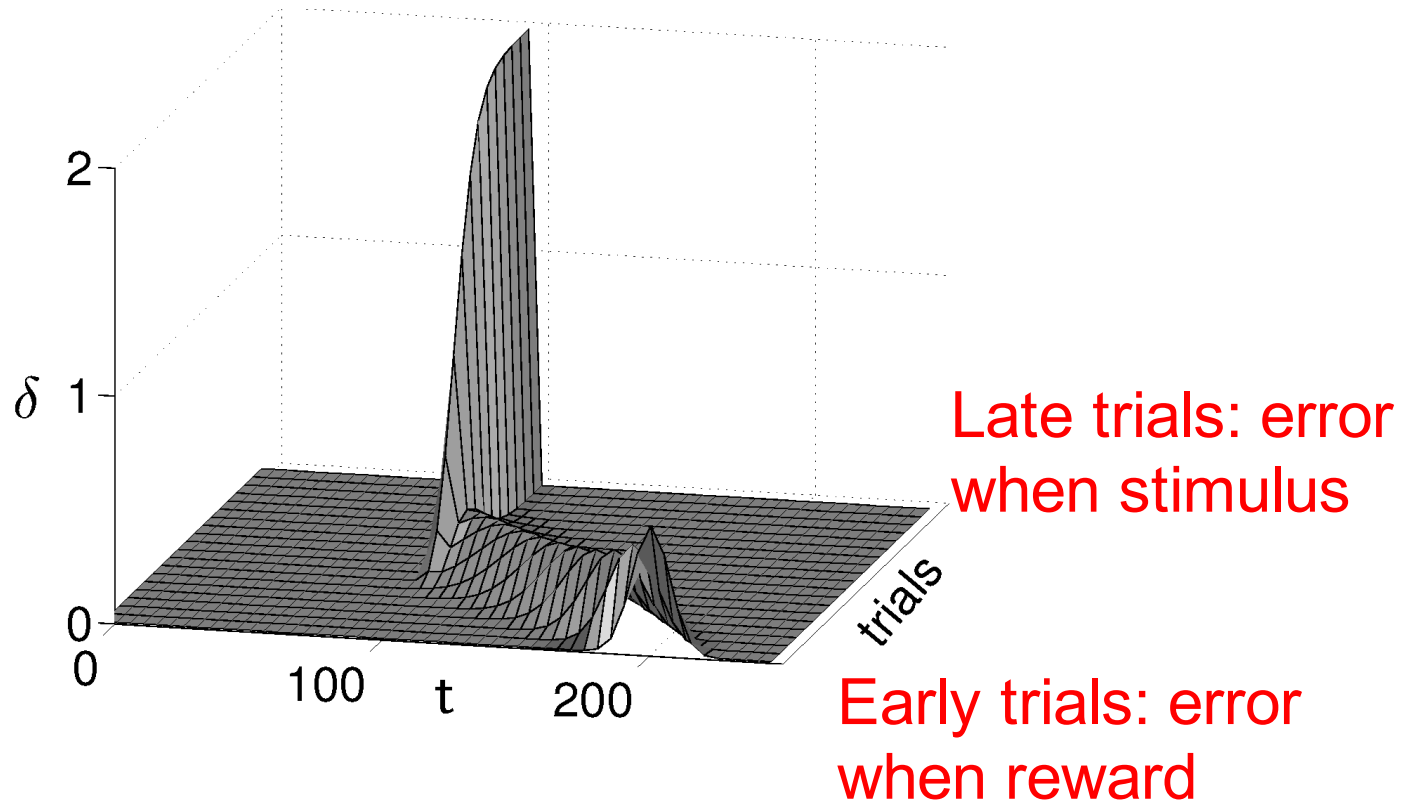
$$\delta_n = r_n - v_n$$

- Temporal Difference Error: (t is time within a trial)

$$\delta_t = r_t + v_{t+1} - v_t$$

Updates are causal

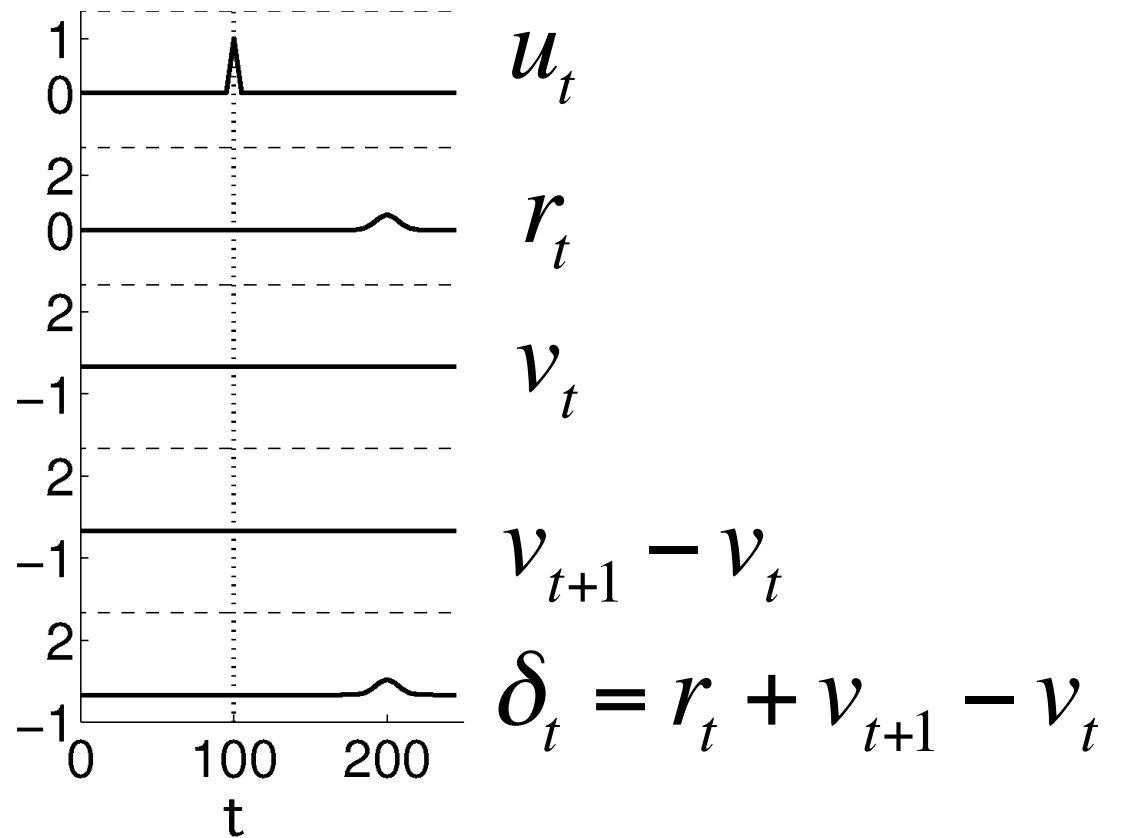
# Temporal Difference Learning



Dayan and Abbott Book: Surface plot of prediction error (stimulus at 100; reward at 200)

# Temporal Difference Learning

Before learning



# Temporal Difference Learning

After learning

