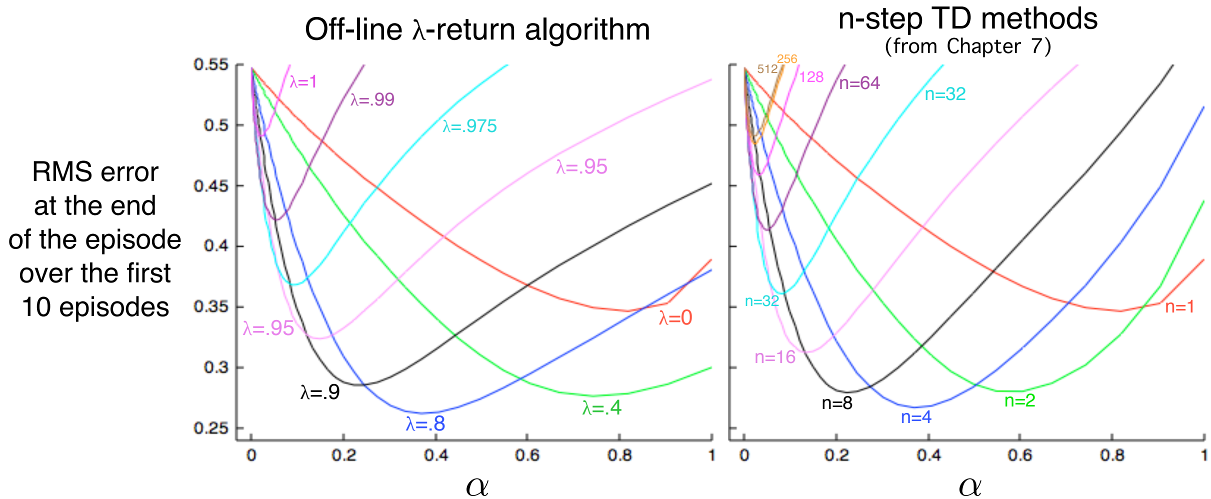
Eligibility Trace + TD(λ)
&
Start of Model-based

The off-line λ-return "algorithm"

- Wait until the end of the episode (offline)

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \Big[G_t^\lambda - \hat{q}(S_t, A_t, \boldsymbol{\theta}_t) \Big] \nabla \hat{q}(S_t, A_t, \boldsymbol{\theta}_t), \quad t = 0, \ldots, T-1$$

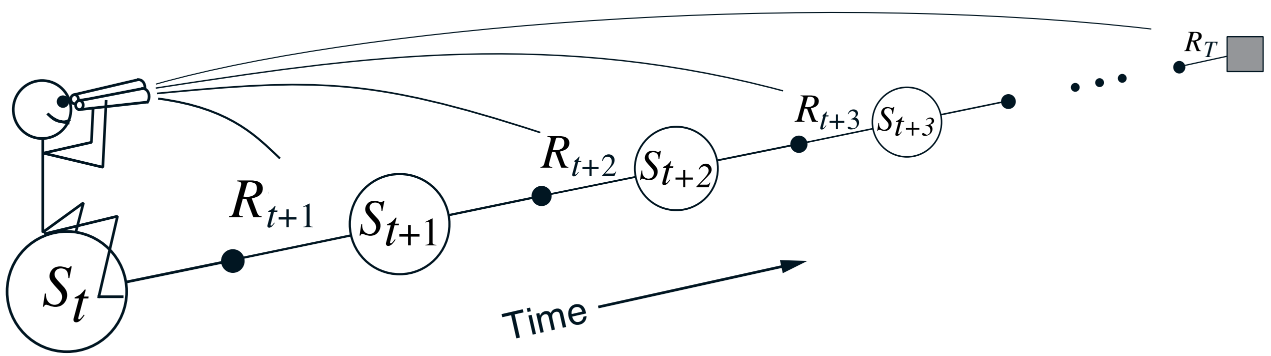
The λ-return alg performs similarly to



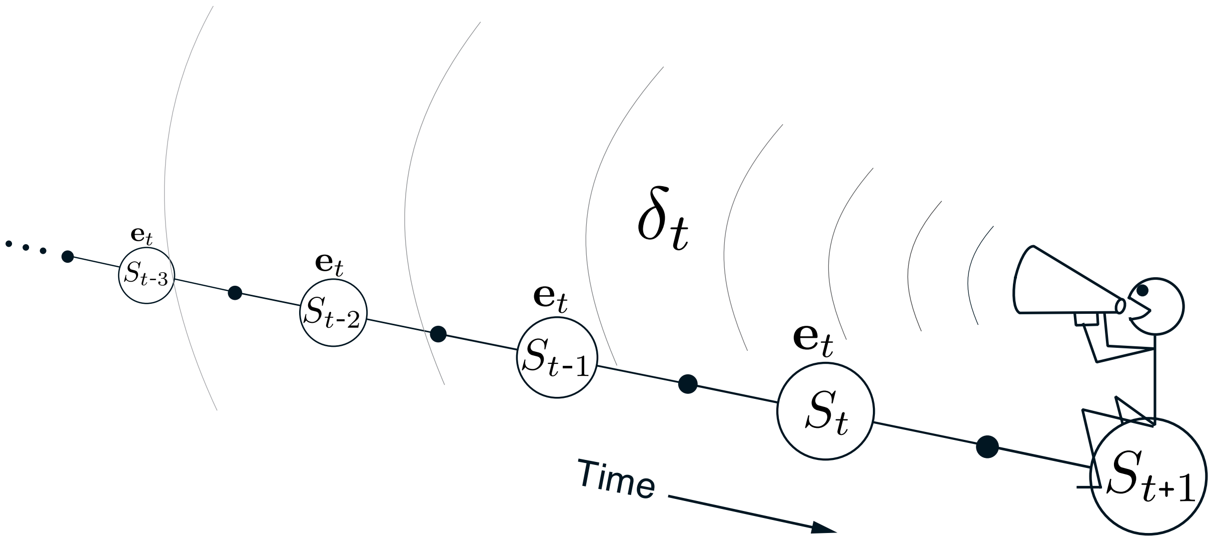
Intermediate λ is best (just like intermediate *n* is best)

λ-return slightly better than *n*-step

The forward view looks forward from the state being updated to future states and rewards



The backward view looks back to the recently visited states (marked by eligibility traces)

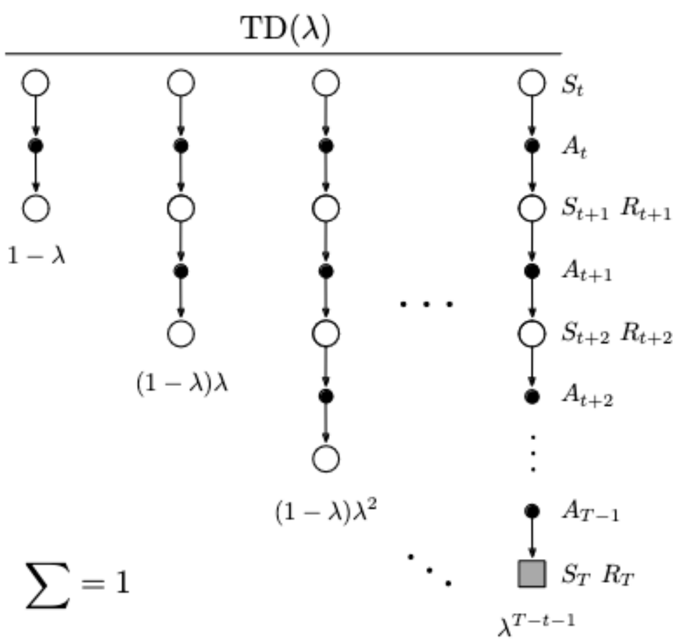


- Shout the TD error backwards
- The traces fade with temporal distance by γλ

The λ-return is a compound update target

- The λ-return a target that averages all *n*-step targets
  - each weighted by λ^{n-1}

$$G_t^\lambda \doteq (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_{t:t+n}.$$



$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \hat{v}(S_{t+n}, \mathbf{w}_{t+n-1}), \quad 0 \le t \le T-n,$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $$\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$$

$$\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t)$$

Updates:

Tabular:

$$Q_{k+1}(A_t, S_t) = Q_k(A_t, S_t) + \alpha \Delta_t^\lambda$$

Function Approx:

$$\theta_{k+1} = \theta_k + \alpha \Delta_t^\lambda \nabla_\theta Q(A_t, S_t, \theta)$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) +$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) + \quad (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1})\right)$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\begin{aligned} \Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) + \quad & (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1})\right) \\ + \quad & (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(A_{t+2}, S_{t+2})\right) \\ + \quad & \end{aligned}$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\begin{aligned}
\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) + & \quad (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1})\right) \\
+ & \quad (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(A_{t+2}, S_{t+2})\right) \\
+ & \quad (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 Q(A_{t+3}, S_{t+3})\right) \\
+ & \quad \dots \\
+ & \quad \lambda^{T-t-1} \left(R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-t-1} R_T\right)
\end{aligned}$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\begin{aligned}
\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) + \quad & (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(A_{t+2}, S_{t+2})\right) \\
+ \quad & (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 Q(A_{t+3}, S_{t+3})\right) \\
+ \quad & \dots \\
+ \quad & \lambda^{T-t-1} \left(R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-t-1} R_T\right)
\end{aligned}$$

$$= -Q(A_t, S_t) +$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\begin{aligned}
\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) + \quad & (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(A_{t+2}, S_{t+2})\right) \\
+ \quad & (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 Q(A_{t+3}, S_{t+3})\right) \\
+ \quad & \dots \\
+ \quad & \lambda^{T-t-1} \left(R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-t-1} R_T\right) \\
= -Q(A_t, S_t) + \quad & (\gamma\lambda)^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1}) - \gamma\lambda Q(A_{t+1}, S_{t+1})\right)
\end{aligned}$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\begin{aligned}
\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) + \quad & (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(A_{t+2}, S_{t+2})\right) \\
+ \quad & (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 Q(A_{t+3}, S_{t+3})\right) \\
+ \quad & \dots \\
+ \quad & \lambda^{T-t-1} \left(R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-t-1} R_T\right) \\
\\
= -Q(A_t, S_t) + \quad & (\gamma\lambda)^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1}) - \gamma\lambda Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (\gamma\lambda)^1 \left(R_{t+2} + \gamma Q(A_{t+2}, S_{t+2}) - \gamma\lambda Q(A_{t+2}, S_{t+2})\right)
\end{aligned}$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\begin{aligned}
\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) + \quad & (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(A_{t+2}, S_{t+2})\right) \\
+ \quad & (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 Q(A_{t+3}, S_{t+3})\right) \\
+ \quad & \dots \\
+ \quad & \lambda^{T-t-1} \left(R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-t-1} R_T\right) \\
\\
= -Q(A_t, S_t) + \quad & (\gamma\lambda)^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1}) - \gamma\lambda Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (\gamma\lambda)^1 \left(R_{t+2} + \gamma Q(A_{t+2}, S_{t+2}) - \gamma\lambda Q(A_{t+2}, S_{t+2})\right) \\
+ \quad & (\gamma\lambda)^2 \left(R_{t+3} + \gamma Q(A_{t+3}, S_{t+3}) - \gamma\lambda Q(A_{t+3}, S_{t+3})\right)
\end{aligned}$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\begin{aligned}
\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) + \quad & (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(A_{t+2}, S_{t+2})\right) \\
+ \quad & (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 Q(A_{t+3}, S_{t+3})\right) \\
+ \quad & \dots \\
+ \quad & \lambda^{T-t-1} \left(R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-t-1} R_T\right)
\end{aligned}$$

$$\begin{aligned}
= -Q(A_t, S_t) + \quad & (\gamma\lambda)^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1}) - \gamma\lambda Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (\gamma\lambda)^1 \left(R_{t+2} + \gamma Q(A_{t+2}, S_{t+2}) - \gamma\lambda Q(A_{t+2}, S_{t+2})\right) \\
+ \quad & (\gamma\lambda)^2 \left(R_{t+3} + \gamma Q(A_{t+3}, S_{t+3}) - \gamma\lambda Q(A_{t+3}, S_{t+3})\right) \\
+ \quad & \dots \\
+ \quad & (\gamma\lambda)^{T-t-1} \left(R_T\right)
\end{aligned}$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\begin{aligned}
\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) + &\quad (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1})\right) \\
+ &\quad (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(A_{t+2}, S_{t+2})\right) \\
+ &\quad (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 Q(A_{t+3}, S_{t+3})\right) \\
+ &\quad \ldots \\
+ &\quad \lambda^{T-t-1} \left(R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-t-1} R_T\right) \\
= -Q(A_t, S_t) + &\quad (\gamma\lambda)^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1}) - \gamma\lambda Q(A_{t+1}, S_{t+1})\right) \\
+ &\quad (\gamma\lambda)^1 \left(R_{t+2} + \gamma Q(A_{t+2}, S_{t+2}) - \gamma\lambda Q(A_{t+2}, S_{t+2})\right) \\
+ &\quad (\gamma\lambda)^2 \left(R_{t+3} + \gamma Q(A_{t+3}, S_{t+3}) - \gamma\lambda Q(A_{t+3}, S_{t+3})\right) \\
+ &\quad \ldots \\
+ &\quad (\gamma\lambda)^{T-t-1} \left(R_T\right) \\
= &\quad (\gamma\lambda)^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1}) - Q(A_t, S_t)\right) \\
+ &\quad (\gamma\lambda)^1 \left(R_{t+2} + \gamma Q(A_{t+2}, S_{t+2}) - Q(A_{t+1}, S_{t+1})\right) \\
+ &\quad (\gamma\lambda)^2 \left(R_{t+3} + \gamma Q(A_{t+3}, S_{t+3}) - Q(A_{t+2}, S_{t+2})\right) \\
+ &\quad \ldots \\
+ &\quad (\gamma\lambda)^{T-t-1} \left(R_T - Q(A_{T-1}, S_{T-1})\right)
\end{aligned}$$

TD(λ)'s Δ

Useful Identities:

$$\sum_{k=0}^{N-1} \lambda^k = \frac{1-\lambda^N}{1-\lambda}$$

$$(1-\lambda)\left(\sum_{k=0}^{N-1} \lambda^k\right) + \lambda^N = (1-\lambda)\frac{1-\lambda^N}{1-\lambda} + \lambda^N = 1$$

Definition: $\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$

$$\begin{aligned}
\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t) = -Q(A_t, S_t) + \quad & (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(A_{t+2}, S_{t+2})\right) \\
+ \quad & (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 Q(A_{t+3}, S_{t+3})\right) \\
+ \quad & \dots \\
+ \quad & \lambda^{T-t-1} \left(R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-t-1} R_T\right) \\
\\
= -Q(A_t, S_t) + \quad & (\gamma\lambda)^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1}) - \gamma\lambda Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (\gamma\lambda)^1 \left(R_{t+2} + \gamma Q(A_{t+2}, S_{t+2}) - \gamma\lambda Q(A_{t+2}, S_{t+2})\right) \\
+ \quad & (\gamma\lambda)^2 \left(R_{t+3} + \gamma Q(A_{t+3}, S_{t+3}) - \gamma\lambda Q(A_{t+3}, S_{t+3})\right) \\
+ \quad & \dots \\
+ \quad & (\gamma\lambda)^{T-t-1} \left(R_T\right) \\
\\
= \quad & (\gamma\lambda)^0 \left(R_{t+1} + \gamma Q(A_{t+1}, S_{t+1}) - Q(A_t, S_t)\right) \\
+ \quad & (\gamma\lambda)^1 \left(R_{t+2} + \gamma Q(A_{t+2}, S_{t+2}) - Q(A_{t+1}, S_{t+1})\right) \\
+ \quad & (\gamma\lambda)^2 \left(R_{t+3} + \gamma Q(A_{t+3}, S_{t+3}) - Q(A_{t+2}, S_{t+2})\right) \\
+ \quad & \dots \\
+ \quad & (\gamma\lambda)^{T-t-1} \left(R_T - Q(A_{T-1}, S_{T-1})\right) \\
\\
= \delta_t + \gamma\lambda\delta_{t+1} + (\gamma\lambda)^2\delta_{t+2} + \dots + (\gamma\lambda)^{T-t-1}\delta_T
\end{aligned}$$

Forwards and Backwards TD(λ)

Definition:

$$\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$$

$$\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t)$$

TD(λ) eligibility trace discounts time since visit,

Tabular:

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$

Funct. Approx:

$$E_t(s) = \gamma\lambda E_{t-1} \quad + \nabla_\theta Q(A_t, S_t, \theta)$$

Forwards and Backwards TD(λ)

Definition:

$$\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$$

$$\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t)$$

TD(λ) eligibility trace discounts time since visit,

Tabular:

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$

If state s is visited at time k then:

$$= \begin{cases} 0 & \text{if } t < k \\ (\gamma\lambda)^{t-k} & \text{if } t \geq k \end{cases}$$

Funct. Approx:

$$E_t(s) = \gamma\lambda E_{t-1} \quad + \nabla_\theta Q(A_t, S_t, \theta)$$

Forwards and Backwards TD(λ)

Definition:

$$\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$$

$$\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t)$$

TD(λ) eligibility trace discounts time since visit,

Tabular:

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$

If state s is visited at time k then:

$$= \begin{cases} 0 & \text{if } t < k \\ (\gamma\lambda)^{t-k} & \text{if } t \geq k \end{cases}$$

Funct. Approx:

$$E_t(s) = \gamma\lambda E_{t-1} + \nabla_\theta Q(A_t, S_t, \theta)$$

Backward TD(λ) updates accumulate error *online*

$$\sum_{t=1}^{T} \alpha\delta_t E_t(s) = \alpha \sum_{t=k}^{T} (\gamma\lambda)^{t-k} \delta_t = \alpha \left(G_k^\lambda - V(S_k) \right)$$

Forwards and Backwards TD(λ)

Definition:

$$\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$$

$$\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t)$$

TD(λ) eligibility trace discounts time since visit,

Tabular:

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$

If state s is visited at time k then:

$$= \begin{cases} 0 & \text{if } t < k \\ (\gamma\lambda)^{t-k} & \text{if } t \geq k \end{cases}$$

Funct. Approx:

$$E_t(s) = \gamma\lambda E_{t-1} \quad + \nabla_\theta Q(A_t, S_t, \theta)$$

Backward TD(λ) updates accumulate error *online*

$$\sum_{t=1}^{T} \alpha\delta_t E_t(s) = \alpha \sum_{t=k}^{T} (\gamma\lambda)^{t-k}\delta_t = \alpha\left(G_k^\lambda - Q(A_k, S_k)\right)$$

- By end of episode it accumulates total error for λ-return
- For multiple visits to s, $E_t(s)$ accumulates many errors

Forwards and Backwards TD(λ)

Definition:

$$\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$$

$$\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t)$$

TD(λ) eligibility trace discounts time since visit,

Tabular:

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$

Funct. Approx:

$$E_t(s) = \gamma\lambda E_{t-1} + \nabla_\theta Q(A_t, S_t, \theta)$$

If state s is visited at time k then:

$$= \begin{cases} 0 & \text{if } t < k \\ (\gamma\lambda)^{t-k} \nabla_\theta Q(A_k, S_k) & \text{if } t \geq k \end{cases}$$

Forwards and Backwards TD(λ)

Definition:

$$\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$$

$$\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t)$$

TD(λ) eligibility trace discounts time since visit,

Tabular:

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$

Funct. Approx:

$$E_t(s) = \gamma\lambda E_{t-1} \quad + \nabla_\theta Q(A_t, S_t, \theta)$$

If state s is visited at time t then:

$$= \begin{cases} 0 & \text{if } t < k \\ (\gamma\lambda)^{t-k} \nabla_\theta Q(A_k, S_k) & \text{if } t \geq k \end{cases}$$

Backward TD(λ) updates accumulate error *online*

$$\sum_{t=1}^{T} \alpha\delta_t E_t(s) = \alpha \sum_{t=k}^{T} (\gamma\lambda)^{t-k} \delta_t \nabla_\theta Q(A_t, S_t, \theta) = \alpha \left(G_k^\lambda - Q(A_k, S_k) \right) \nabla_\theta Q(A_t, S_t, \theta)$$

Forwards and Backwards TD(λ)

Definition:

$$\delta_k := R_{k+1} + \gamma Q(A_{k+1}, S_{k+1}) - Q(A_k, S_k)$$

$$\Delta_t^\lambda := G_t^\lambda - Q(A_t, S_t)$$

TD(λ) eligibility trace discounts time since visit,

Tabular:

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$

Funct. Approx:

$$E_t(s) = \gamma\lambda E_{t-1} \quad + \nabla_\theta Q(A_t, S_t, \theta)$$

If state s is visited at time t then:

$$= \begin{cases} 0 & \text{if } t < k \\ (\gamma\lambda)^{t-k} \nabla_\theta Q(A_k, S_k) & \text{if } t \geq k \end{cases}$$

Backward TD(λ) updates accumulate error *online*

$$\sum_{t=1}^{T} \alpha\delta_t E_t(s) = \alpha \sum_{t=k}^{T} (\gamma\lambda)^{t-k} \delta_t \nabla_\theta Q(A_t, S_t, \theta) = \alpha \left(G_k^\lambda - Q(A_k, S_k) \right) \nabla_\theta Q(A_t, S_t, \theta)$$

- By end of episode it accumulates total error for λ-return
- For multiple visits to s, $E_t(s)$ accumulates many errors

Eligibility traces (mechanism)

- The forward view was for theory
- The backward view is for *mechanism* same shape as $\boldsymbol{\theta}$

$$\mathbf{e}_t \in \mathbb{R}^n \geq \mathbf{0}$$

- New memory vector called *eligibility trace*
  - On each step, decay each component by γλ and increment the trace for the current state by 1
  - *Accumulating trace*

$$\mathbf{e}_0 \doteq \mathbf{0},$$
$$\mathbf{e}_t \doteq \nabla \hat{v}(S_t,\boldsymbol{\theta}_t) + \gamma\lambda\mathbf{e}_{t-1}$$

The Semi-gradient TD(λ) algorithm

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \delta_t \mathbf{e}_t$$

$$\delta_t \doteq R_{t+1} + \gamma \hat{v}(S_{t+1},\boldsymbol{\theta}_t) - \hat{v}(S_t,\boldsymbol{\theta}_t)$$

$$\mathbf{e}_0 \doteq \mathbf{0},$$
$$\mathbf{e}_t \doteq \nabla \hat{v}(S_t,\boldsymbol{\theta}_t) + \gamma\lambda \mathbf{e}_{t-1}$$

Online TD(λ)

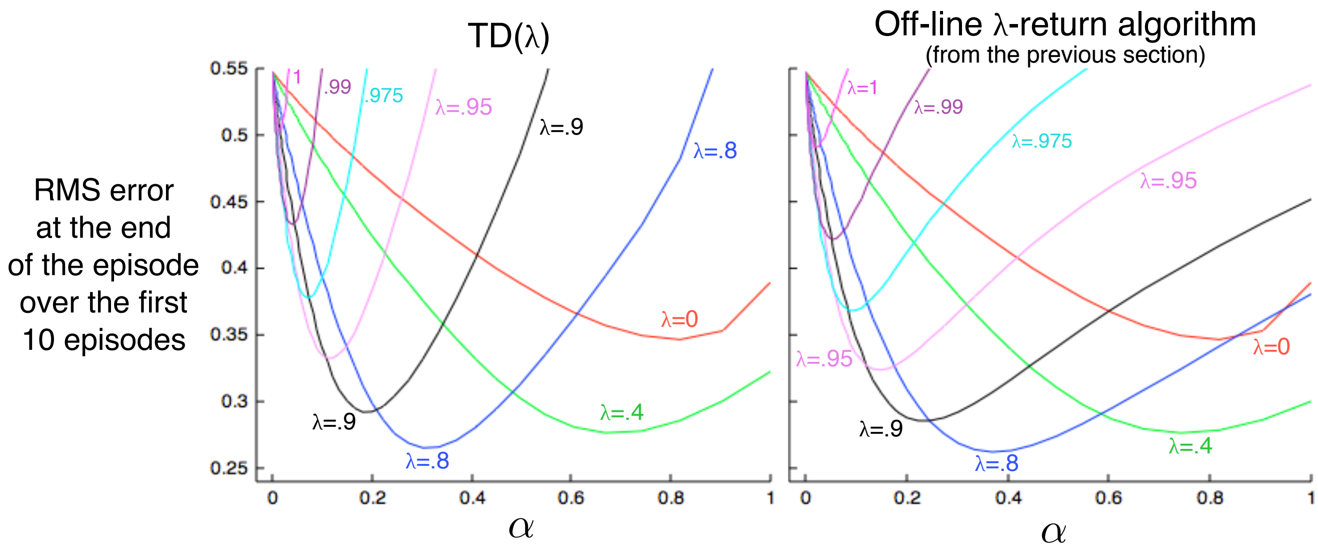
Semi-gradient TD(λ) for estimating $\hat{v} \approx v_\pi$

Input: the policy π to be evaluated
Input: a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^d \rightarrow \mathbb{R}$ such that $\hat{v}(\text{terminal},\cdot) = 0$
Algorithm parameters: step size $\alpha > 0$, trace decay rate $\lambda \in [0, 1]$
Initialize value-function weights $\mathbf{w}$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop for each episode:
    Initialize S
    $\mathbf{z} \leftarrow \mathbf{0}$ (a d-dimensional vector)
    Loop for each step of episode:
    | Choose $A \sim \pi(\cdot|S)$
    | Take action A, observe R, S'
    | $\mathbf{z} \leftarrow \gamma\lambda\mathbf{z} + \nabla\hat{v}(S,\mathbf{w})$
    | $\delta \leftarrow R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$
    | $\mathbf{w} \leftarrow \mathbf{w} + \alpha\delta\mathbf{z}$
    | $S \leftarrow S'$
    until S' is terminal

TD(λ) performs similarly to offline λ-

Tabular 19-state random walk task



Can we do better? Can we update online?

Conclusions

- Value-function approximation by stochastic gradient descent enables RL to be applied to arbitrarily large state spaces
- Most algorithms just carry over the targets from the tabular case
- With bootstrapping (TD), we don't get true gradient descent methods
    - this complicates the analysis
    - but the linear, on-policy case is still guaranteed convergent
    - and learning is still *much faster*

How do we decide what to do?

How do we decide what to do?

	state values	action values
prediction	v_π	q_π
control	v_*	q_*

- Distinct from their estimates: $V_t(s)$ $Q_t(s, a)$

How do we decide what to do?

- Emotions/Intuition  $V_t(s)$ $Q_t(s,a)$

How do we decide what to do?

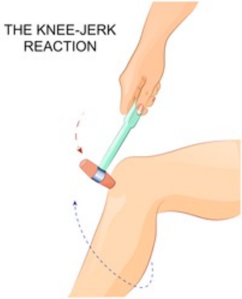
- Emotions/Intuition

$$V_t(s) \qquad Q_t(s,a)$$

- Thinking

$$S_{t+1} = M(S_t, A_t, \theta)$$

- Reflexes/Habits



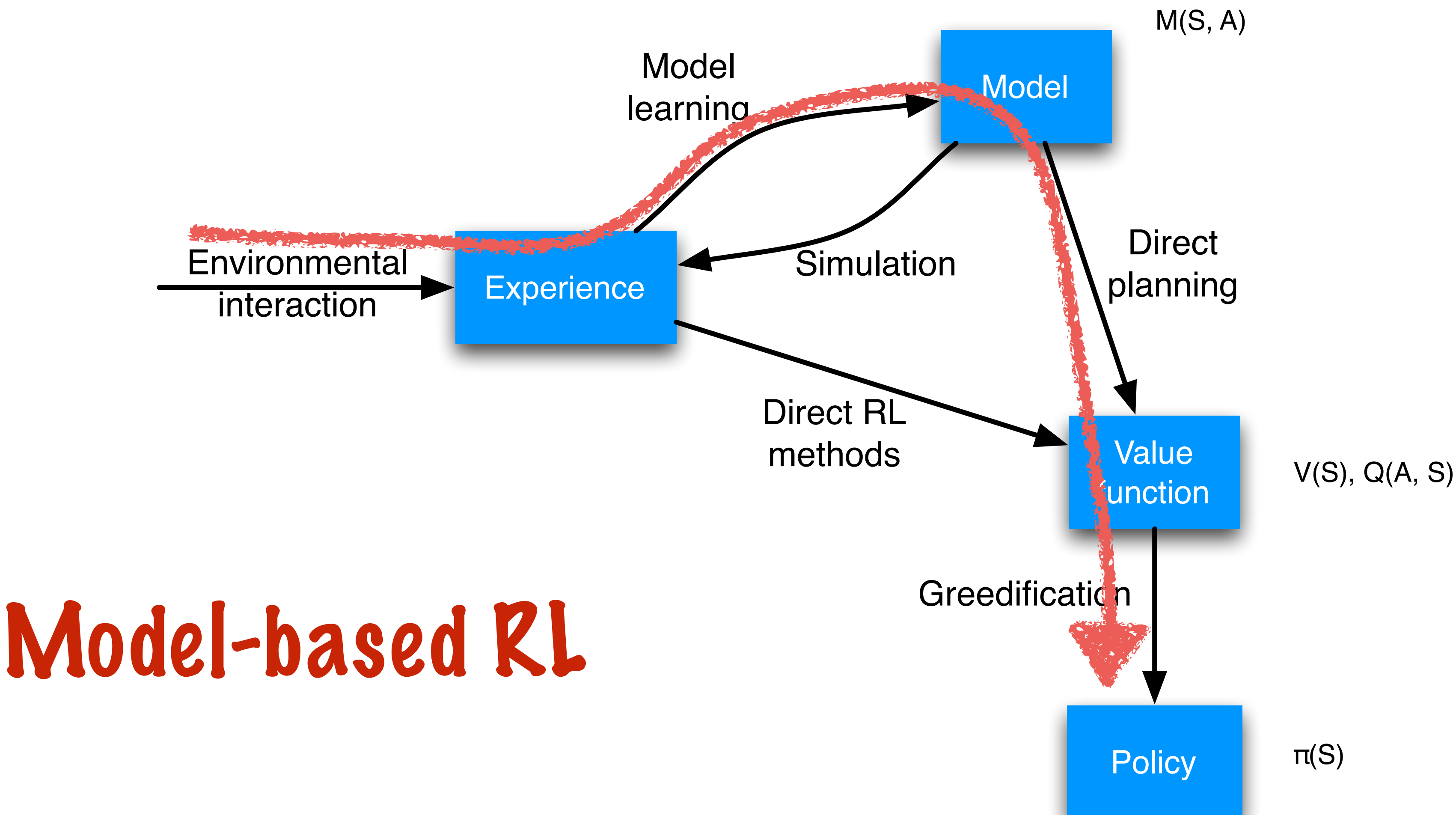
$$A_t = \pi(S_t, \theta)$$

Chapter 8: Planning and Learning

Objectives of this chapter:

- To think more generally about uses of environment models
- Integration of (unifying) planning, learning, and execution
- “Model-based reinforcement learning”

Paths to a policy



Why Going Beyond Model-Free RL?

- Models provide "understanding" of the world (cf physics, causality...)
- Even if some parts of the problem change, others stay the same, which can help with faster learning

  Eg. Reward may change but the layout and dynamics of thee world may be thee same
- Models can be used to "dream" up new experiences, and use them to update the value / policy