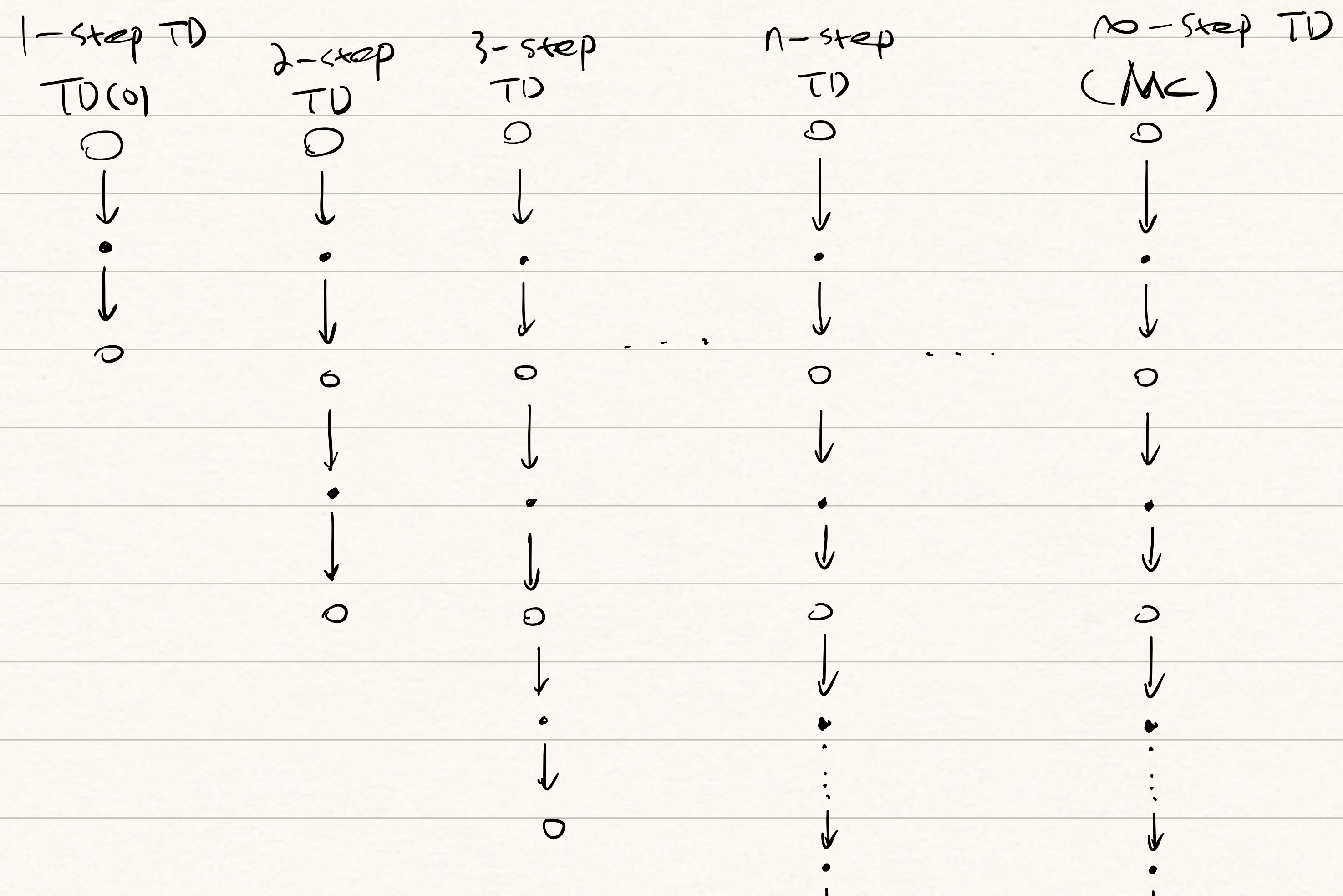
n-step Bootstrapping

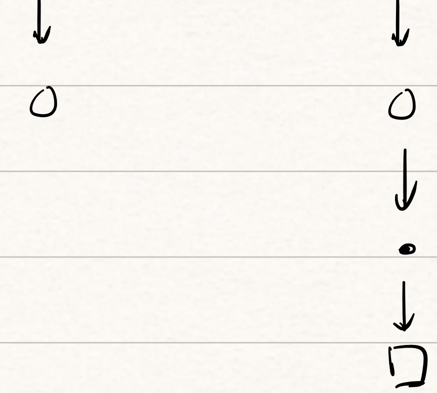
"enable bootstrapping to occur over multiple steps.

"Eligibility traces"

MC : "update" entire sequence of observed rewards from that state until the end of the episode

one-step TD : just one next reward





$S_t, R_{t+1}, S_{t+1}, R_{t+2}, \ldots, R_T, S_T$

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$

T : Last time step of the episode

(quantity the target of the update)

MC : target : Return

one-step return :

$$G_{t:t+1} \doteq R_{t+1} + \gamma V_t(S_{t+1})$$

two-step return :

$$G_{t:t+2} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V_{t+1}(S_{t+2})$$

n-step return :

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$

All n-step returns can be considered approximations to the full return, truncated after n steps and then corrected for the remaining missing term by

$$\boxed{V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]}$$

$$\begin{cases} G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i \\ G \leftarrow G + \gamma^n V(S_{\tau+n}) \text{ if } \tau+n<T \\ V(S_\tau) \leftarrow V(S_\tau) + \alpha[G - V(S_\tau)] \end{cases}$$

Error reduction property:

$$\max_s \left| \mathbb{E}_\pi[G_{t:t+n} \mid S_t = s] - V_\pi(s) \right|$$

$$\leq \gamma^n \max_s \left| V_{t+n-1}(s) - V_\pi(s) \right|$$

n-step Sarsa

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^{n} Q_{t+n-1}(S_{t+n}, A_{t+n})$$

$$Q_{t+n}(S_t, A_t) \doteq Q_{t+n-1}(S_t, A_t) + \alpha \left[G_{t:t+n} - Q_{t+n-1}(S_t, A_t) \right]$$

n-step Off-policy Learning

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha \rho_{t:t+n-1} \left[G_{t:t+n} - V_{t+n-1}(S_t) \right]$$

$$\rho_{t:h} \doteq \prod_{k=t}^{\min(h, T-1)} \frac{\pi(A_k \mid S_k)}{b(A_k \mid S_k)}$$

n-step $Q(\sigma)$

$\sigma_t \in [0, 1]$ degree of sampling on step t

$$G_{t:h} = R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{h-1}(S_{t+1}, a) + \gamma \pi(A_{t+1}|S_{t+1}) G_{t+1:h}$$

$$= R_{t+1} + \gamma \bar{V}_{h-1}(S_{t+1}) - \gamma \pi(A_{t+1}|S_{t+1}) Q(S_{t+1}, A_{t+1}) + \gamma \pi(A_{t+1}|S_{t+1}) G_{t+1:h}$$

$$= R_{t+1} + \gamma \pi(A_{t+1}|S_{t+1}) \left(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1}) \right) + \gamma \bar{V}_{h-1}(S_{t+1})$$

tree-backup n-step return

$$G_{t:h} \doteq R_{t+1} + \gamma \left(\sigma_{t+1} \rho_{t+1} + (1 - \sigma_{t+1}) \pi(A_{t+1}|S_{t+1}) \right) \left(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1}) \right) + \gamma \bar{V}_{h-1}(S_{t+1})$$

Model of the environment:

Model-based : planning

Model-free : learning

SAME: look ahead to future events, computing a backed-up value, and then using it as an update target for an approximate value function

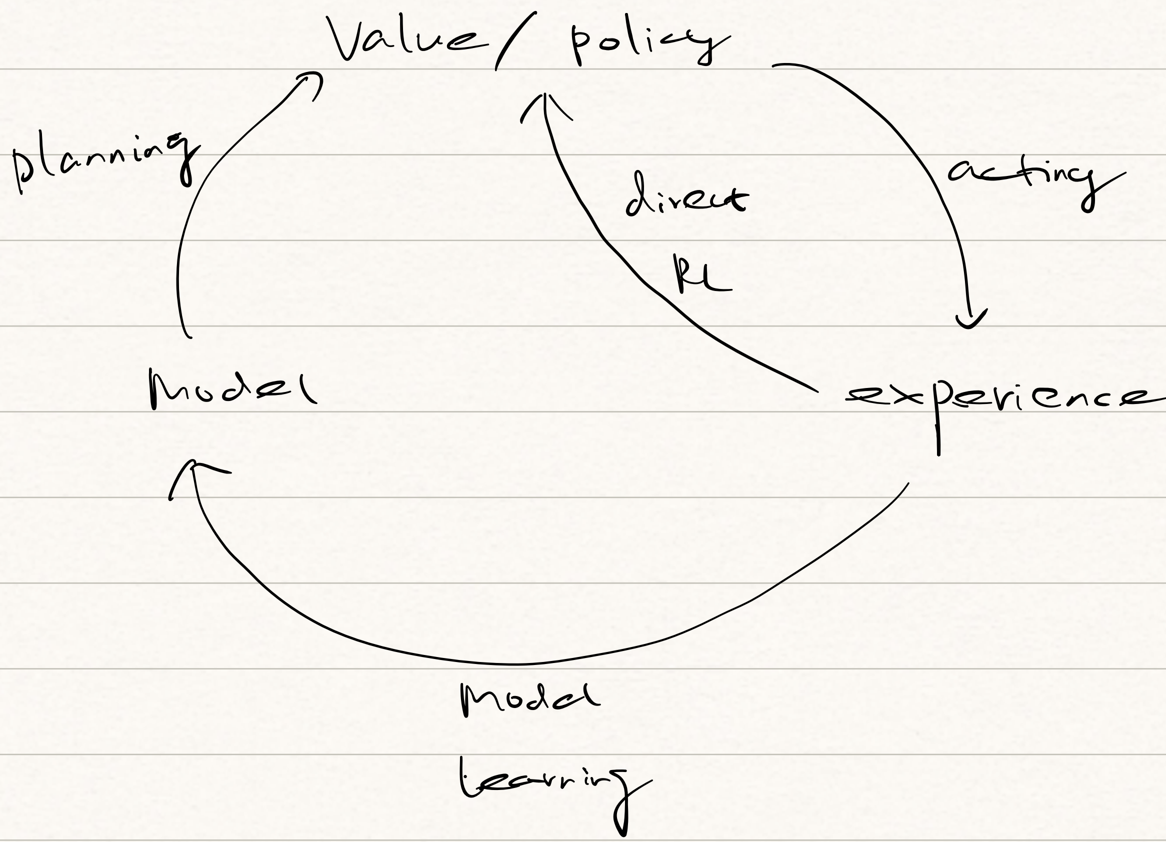
Model-based

"An agent can use to predict how the environment will respond to its actions"

distribution model: consists of the probabilities of next states and rewards for possible actions

sample model: single transitions and rewards generated according to these probabilities.

$$\boxed{\text{model} \xrightarrow{\text{planning}} \text{policy}}$$


Value / policy
planning
direct
RL
acting
Model
experience
Model
learning