FPT Education
FPT UNIVERSITY

The Objective for Temporal Difference

Objectives

- Understand the TD-update for function approximation
- Understand that TD converges to a biased value estimate
- Understand that TD converges much faster than Gradient Monte Carlo

FPT Education
FPT UNIVERSITY

Gradient Monte Carlo

- The gradient Monte Carlo update equation.

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha[G_t - \hat{v}(S_t, \mathbf{w})] \nabla \hat{v}(S_t, \mathbf{w})$$

- It updates our current value estimate to be closer to a sample of the return Gt
- We can replace the return in this update with any estimate of the value.

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha[U_t - \hat{v}(S_t, \mathbf{w})] \nabla \hat{v}(S_t, \mathbf{w})$$

Temporal Difference Update

- The TD update for Function Approximation
  - Ut is the estimation. If Ut is an unbiased estimate of the true value then our function approximator will converge to a local optimum under the appropriate conditions.
  - This was the case for the return, but we can also replace Ut with a bootstrap target, such as the one step TD target.

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha[U_t - \hat{v}(S_t, \mathbf{w})] \nabla \hat{v}(S_t, \mathbf{w})$$

$$U_t \doteq R_{t+1} + \gamma \hat{v}(S_{t+1}, \mathbf{w})$$

Temporal Difference Update

- The TD update is not actually a stochastic gradient descent update.
- Ut is equal to the TD target.
- Using the chain rule we get this expanded expression for the gradient of the squared error for

$$\nabla \frac{1}{2}[U_t - \hat{v}(S_t, \mathbf{w})]^2$$

$$= (U_t - \hat{v}(S_t, \mathbf{w}))(\nabla U_t - \nabla \hat{v}(S_t, \mathbf{w}))$$

$$U_t \doteq R_{t+1} + \gamma \hat{v}(S_{t+1}, \mathbf{w})$$

Temporal Difference Update

- In TD the target contains an estimate of the value, which depends on the weights
- Therefore TD is not performing gradient descent updates on the squared error

$$\begin{aligned}\nabla U_t &= \nabla (R_{t+1} + \gamma \hat{v}(S_{t+1}, \mathbf{w})) \\ &= \gamma \nabla \hat{v}(S_{t+1}, \mathbf{w}) \\ &\neq 0\end{aligned}$$

- TD is a semi-gradient method

FPT Education
FPT UNIVERSITY

Temporal Difference Update

- Semi-gradient TD(0) for estimating Vhat~ Vpi

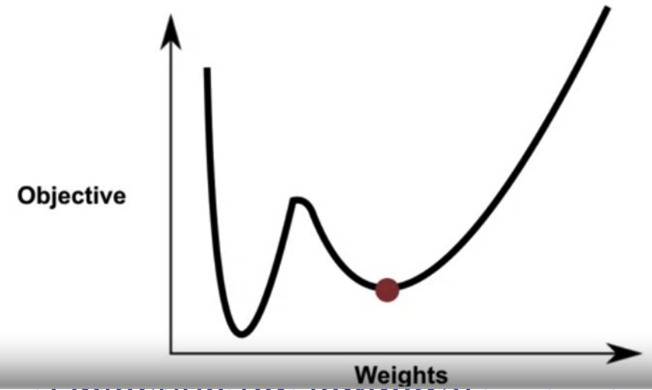
Input: the policy π to be evaluated
Input: a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^d \to \mathbb{R}$ such that $\hat{v}(\text{terminal},\cdot) = 0$
Algorithm parameter: step size $\alpha > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g.,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

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        Choose $A \sim \pi(\cdot|S)$
        Take action A, observe R, S'
        $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})\big]\nabla\hat{v}(S,\mathbf{w})$
        $S \leftarrow S'$
    until S is terminal

Gradient Monte Carlo

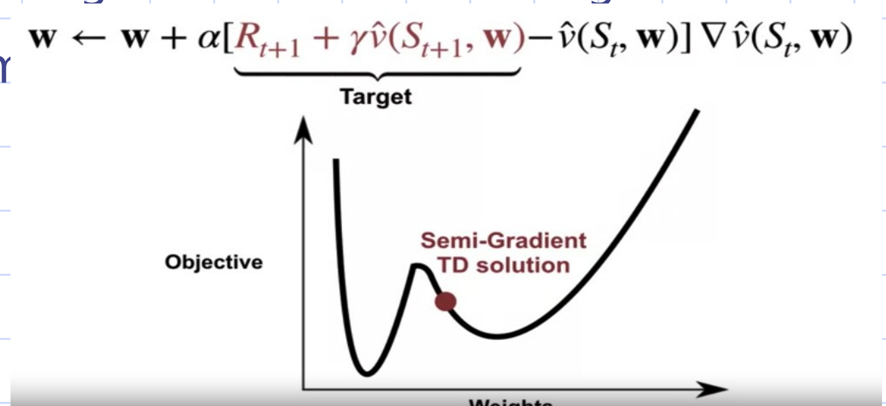
- Gradient Monte Carlo will approach a local minimum of the Mean Squared Value Error
- This is because it uses an unbiased estimate of the gradient of the value error

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha[\underbrace{G_t}_{\text{Target}} - \hat{v}(S_t, \mathbf{w})]\, \nabla \hat{v}(S_t, \mathbf{w})$$



Semi-Gradient TD

- Semi-Gradient TD will not necessarily converge to a local minimum of the Mean Squared Value Error
  - The TD target depends on our estimate of the value in the next state.
  - This means our update could be biased because the estimate in our target may not be accurate.
  - We cannot guarantee that semi-gradient TD will converge to a local m error.



Temporal Difference vs Monte Carlo

- Temporal Difference (TD) learning and Monte Carlo methods are both popular approaches for estimating value functions in reinforcement learning (RL).
- While they share similarities, they also have distinct characteristics.

Temporal Difference vs Monte Carlo

- Update Timing:
  - **Temporal Difference**: TD learning updates value estimates based on each individual time step of experience, bootstrapping from subsequent estimates of state values. It updates value estimates after every time step, using the observed reward and the estimated value of the next state.
  - **Monte Carlo**: Monte Carlo methods update value estimates only after the completion of an episode. They rely on the full return obtained from the episode to update value estimates for each state visited during the episode.

Temporal Difference vs Monte Carlo

- Bootstrapping:
  - **Temporal Difference**: TD learning bootstraps by updating value estimates using estimates of subsequent states. It uses the estimated value of the next state to update the value of the current state.
  - **Monte Carlo**: Monte Carlo methods do not bootstrap and rely entirely on the observed returns from complete episodes to update value estimates. They do not use estimates of subsequent states.

Temporal Difference vs Monte Carlo

- Bias:
  - **Temporal Difference**: TD learning can converge to biased value estimates, especially in non-stationary environments or with suboptimal choices of parameters. It may exhibit bias due to the use of bootstrapping.
  - **Monte Carlo**: Monte Carlo methods provide unbiased estimates of value functions since they rely only on observed returns without bootstrapping. However, they require complete episodes, which may be impractical in some environments.

FPT Education

FPT UNIVERSITY

Temporal Difference vs Monte Carlo

- Variance:
  - **Temporal Difference**: TD learning typically has lower variance in value estimates compared to Monte Carlo methods. This is because TD learning updates value estimates at each time step, leading to faster convergence and reduced variance.
  - **Monte Carlo**: Monte Carlo methods can have higher variance in value estimates, especially when dealing with long episodes or environments with high variability in rewards.

Temporal Difference vs Monte Carlo

- Sample Efficiency:
  - **Temporal Difference**: TD learning is generally more sample-efficient than Monte Carlo methods since it updates value estimates after each time step, allowing for faster learning.
  - **Monte Carlo**: Monte Carlo methods may require more samples to converge since they update value estimates only after complete episodes, which may take longer to generate.

Temporal Difference vs Monte Carlo

- Application:
  - **Temporal Difference**: TD learning is well-suited for online learning and environments where episodes are not well-defined or may be infinitely long. It is commonly used in real-time applications and with function approximation techniques.
  - **Monte Carlo**: Monte Carlo methods are suitable for episodic tasks where complete episodes can be simulated or executed, making them practical for environments with finite episodes and discrete-time steps.

Summary

- Understand the TD-update for function approximation
- Understand that TD converges to a biased value estimate
- Compare TD and Gradient Monte Carlo

Q & A