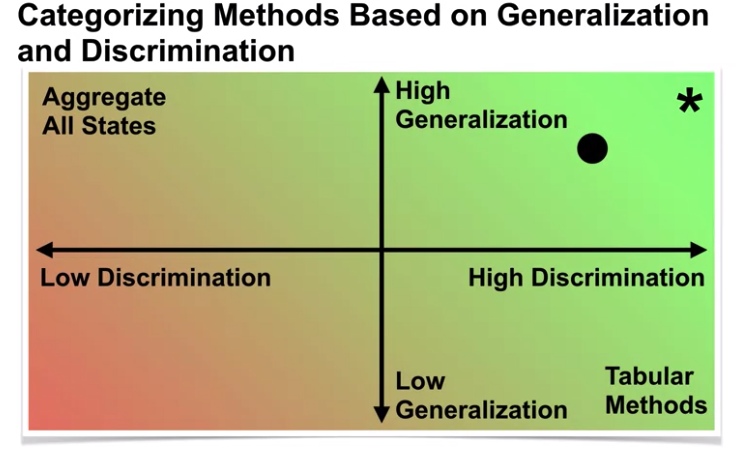
Module 1 Learning Objectives

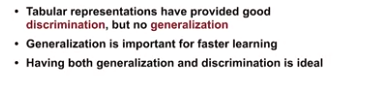
**Lesson 1: Estimating Value Functions as Supervised Learning**

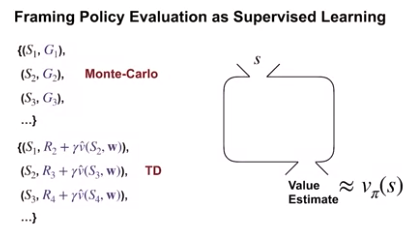
* Understand how we can use parameterized functions to approximate value functions
* Explain the meaning of linear value function approximation
* Recognize that the tabular case is a special case of linear value function approximation.
* Understand that there are many ways to parameterize an approximate value function
* Understand what is meant by generalization and discrimination

Generalization can speed learning by making better use of the experience we have. You may not have to visit every state as much to get this values correct if we can learn its value from similar states. discrimination means the ability to make the values for two states different to distinguish between the values for these two states

* Understand how generalization can be beneficial
* Explain why we want both generalization and discrimination from our function approximation



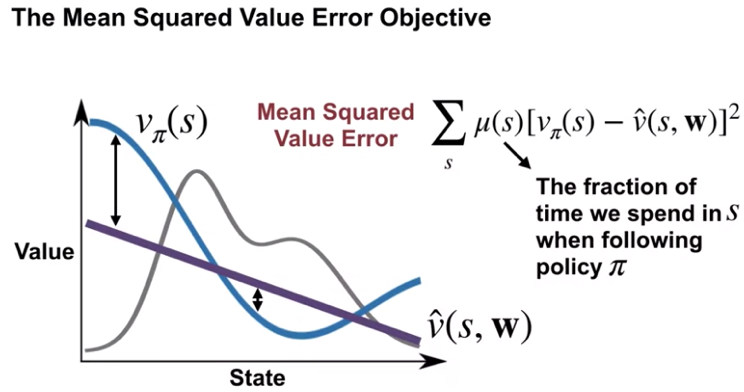


* Understand how value estimation can be framed as a supervised learning problem
* 
* Recognize not all function approximation methods are well suited for reinforcement learning

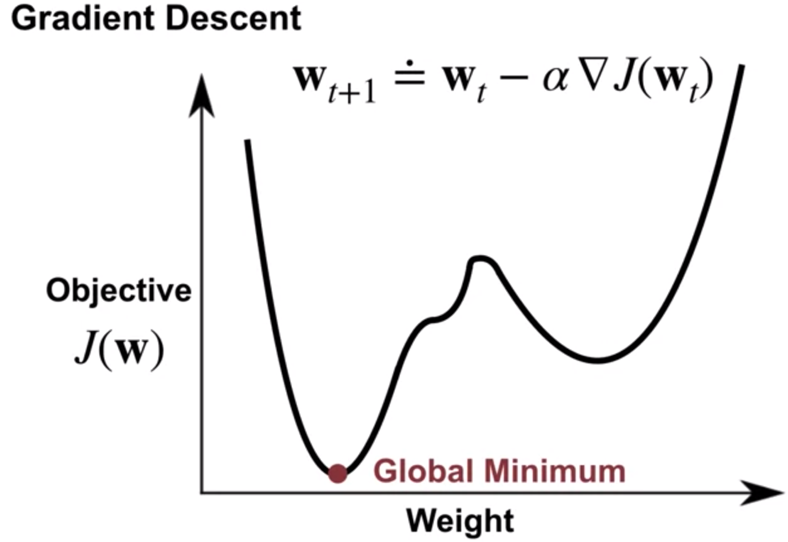
If we want to use a function approximation technique, we should make sure it can work in the online setting. Some methods are not compatible with the online setting because they are either designed for a fixed batch of data or there are not designed for temporally correlated data, and the data in reinforcement learning is always correlated. TD methods introduce an additional complication when applying techniques from supervised learning. TD methods use bootstrapping, meaning that our targets now depend on our own estimates. These estimates change as learning progresses and so our targets continually change.

**Lesson 2: The Objective for On-policy Prediction**

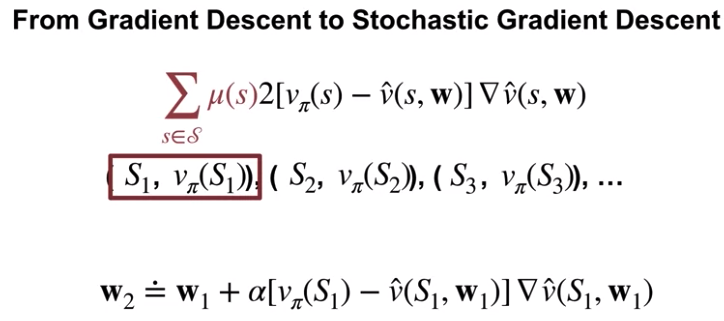
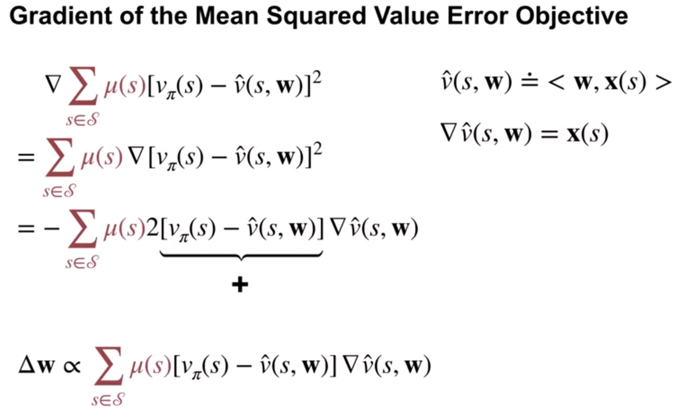
* Understand the mean-squared value error objective for policy evaluation



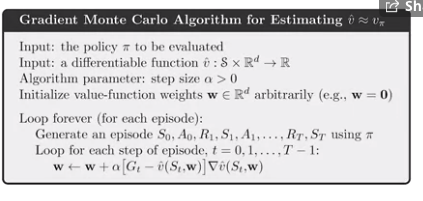
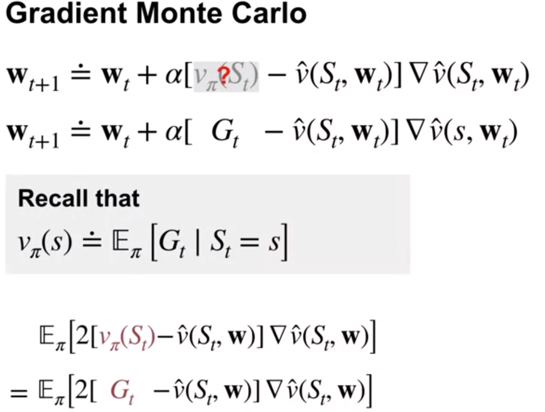
* Explain the role of the state distribution in the objective
* Understand the idea behind gradient descent and stochastic gradient descent



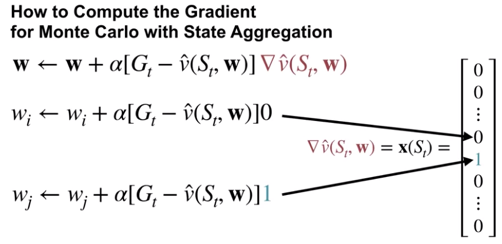
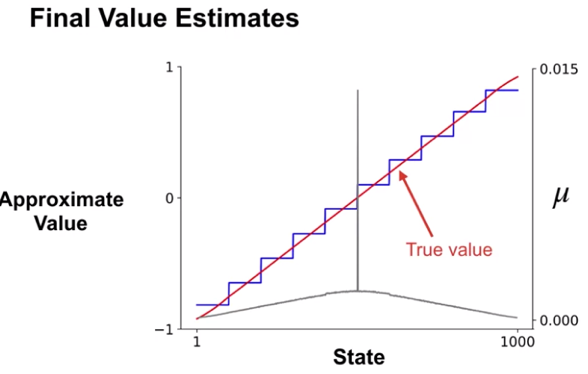
* Outline the gradient Monte Carlo algorithm for value estimation



* Understand how state aggregation can be used to approximate the value function

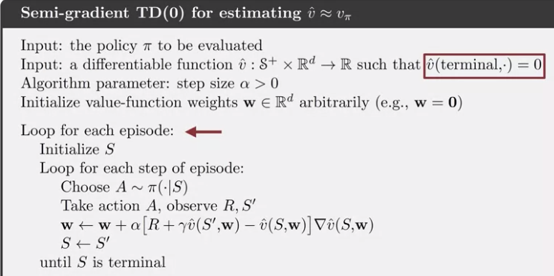
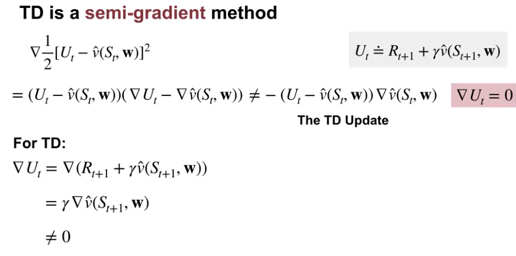


* Apply Gradient Monte-Carlo with state aggregation



**Lesson 3: The Objective for TD**

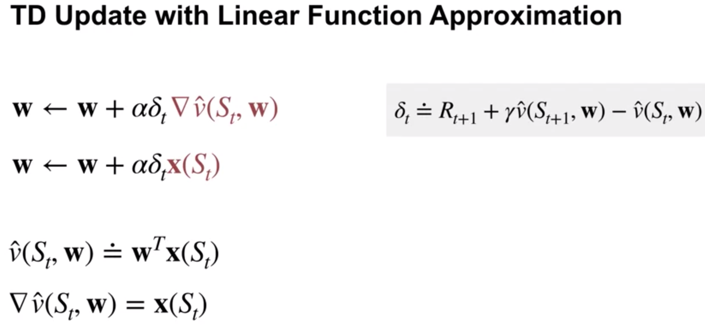
* Understand the TD-update for function approximation



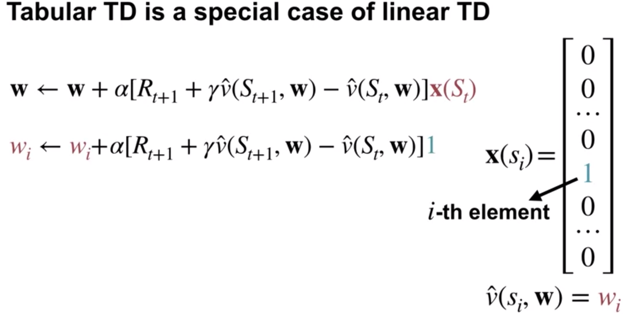
* Highlight the advantages of TD compared to Monte-Carlo
  + MonteCarlo long run performance is better than TD
  + TD converges faster
  + TD learns during the episode and has lower variance update.
* Outline the Semi-gradient TD(0) algorithm for value estimation
* Understand that TD converges to a biased value estimate
* Understand that TD converges much faster than Gradient Monte Carlo

**Lesson 4: Linear TD**

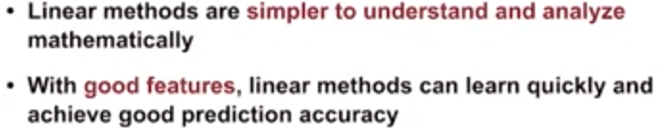
* Derive the TD-update with linear function approximation



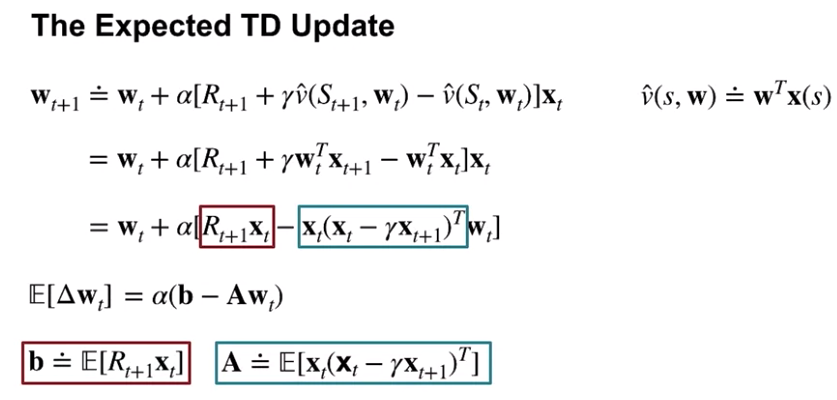
* Understand that tabular TD(0) is a special case of linear semi-gradient TD(0)

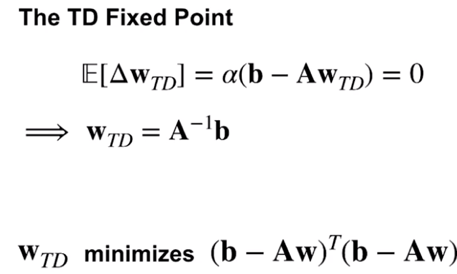


* Highlight the advantages of linear value function approximation over nonlinear

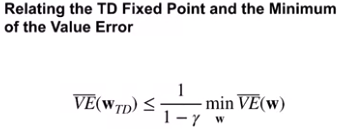


* Understand the fixed point of linear TD learning





* Describe a theoretical guarantee on the mean squared value error at the TD fixed point



The difference between the TD fixed point and the minimum value error solution can be large if Gamma is close to one. If Gamma is very close to zero on the other hand, the TD fixed point is very close to the minimum value error solution. This bound also depends on the quality of the features. If the features are limited, both the minimum mean squared value error and the value era,

the TD fixed point, may be large. If we can perfectly represent the value function, then regardless of Gamma, the TD fixed point is equivalent to the minimum value error solution. This is because both the left and right-hand sides would be zero. So in general, why isn't the TD fixed point equal to the minimum value error solution? This is because of bootstrapping under function approximation. If our estimate of the next state is persistently inaccurate due to function approximation, then TD forever update towards an inaccurate target. On the other hand, if our function approximator is very good, then our estimate of the next state will become very accurate. So bootstrapping off this estimate is not problematic and the error for the TD Solution is close to the minimum value error.

