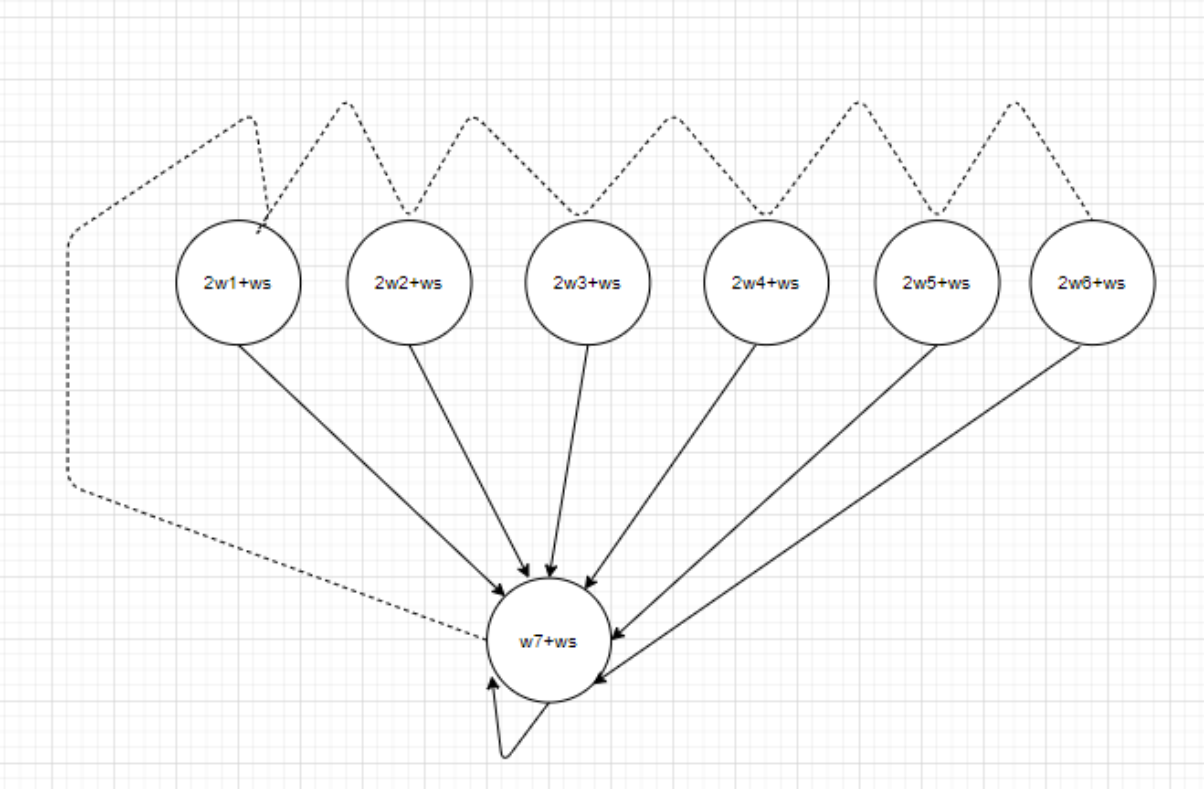
Homework 8

Environment set up:

Here we work on a model which has 7 states and 2 actions , namely ‘solid’ and ‘dash’. We first start randomly at any state from ‘1-7’. A solid action always takes us to the state ‘7’ while dashed actions can take us back to any state randomly except the final state . This all can be better understood by the following figure .



Given:

- policy:
  - Solid :1/7
  - Dashed :6/7
- Target policy for solid :1
- Gamma :0.99 [discount factor]
- Alpha :0.01 [step size]
- Reward=0
- Delta =0

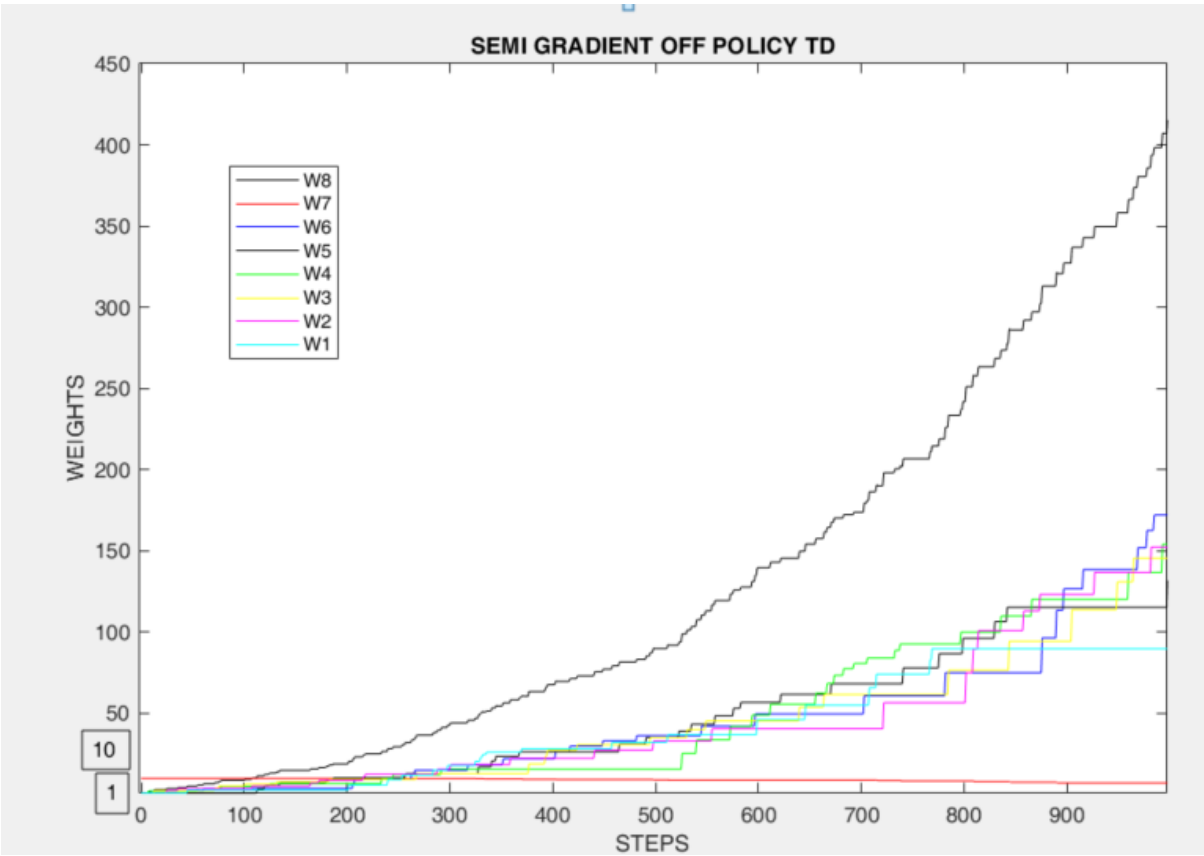
Algorithm explanation:

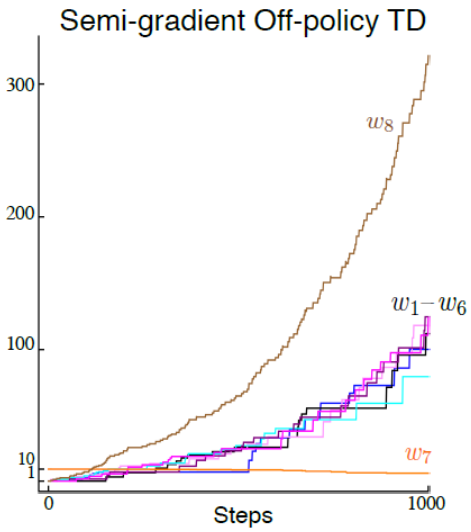
- First, we start by initializing the constants alpha ,gamma . Next we set the number of states possible as "7".
- We then initialize the initial weights as [1,1,1,1,1,1,10,1]
- We also use another matrix of weights "W" to help us plot them finally
- Next we define the probability of solid action as given .
- Next we define the feature vector which helps us define states as a function of the weight . It can be visualized from the diagram . For example : for state 1 is '2'.
- Next we choose a state randomly .
- Then we start the loop for 1000 episodes .
- We use a binomial random function which has one trial and a probability of success of 1/7, which is the probability of choosing a solid action
- If the outcome of the previous step is '1' we understand it is a solid action and hence initialize the next state to be equal to '7' and rho, that is the importance ratio to be 1/(1/7)
- If the outcome is zero, we understand that it is a dotted action and hence assign a probability of rho to be 0 and randomly choose the next action as defined before .
- Next we define v hat as scalar product of the feature vector of the current state and the weight vector ;using the dot function.
- Similarly, we define v hat for the next state as well.
- Next we use the formula: $\delta_t = R_{t+1} + \gamma(S_t, w_t) - \hat{v}(S_t, w_t)$
- Next we update the weights using the formula :$w_{t+1} = w_t + \alpha\rho_t\delta\nabla\hat{v}(S_t, w_t)$
- Lastly, we assign the next state as the current state and loop for the next episode .
- Finally, we plot the weights vs the states .

Time taken to Run: 2.037066 seconds

Plots

Plot from algorithm :





Plot from textbook