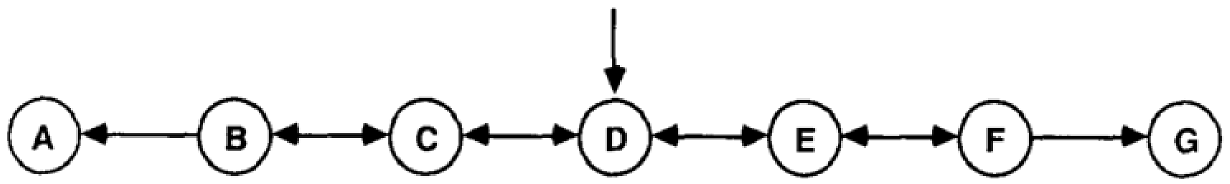
Replicating the Result of Random Walk Experiment

Ruonan Ding

The documentation intend to replicate the Bounded Random Walk experiment in Sutton's 1988 Paper . This experiment requires all episodes start in the center state C. All episodes start in the center State C. It either proceed to the left or right by one state on each step, with equal probability. Episodes terminate either on the extreme left or the extreme right. When an episode terminates on the right, a reward of 1 occurs; all other rewards are zero. Because the task is undiscounted and episodic, the true value of each state is the probability of terminating on the right if starting from that state. The true values of all the states, A through E, are 1/6, 2/6, 3/6, 4/6, 5/6.

Create a random walk Xi vector for each nonterminated state. Therefore, Random Walk function replicates the random walk matrix by moving through different stage to the terminate stages. To achieve statistically reliable results, 10 sequences / episodes were repeated 100 times in the experiment. Weighted were calculated using TD(λ). Lambdas were defined in the a series of [0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0]. When λ equals 1, we will have the Windrow-Hoff supervised learning procedure.

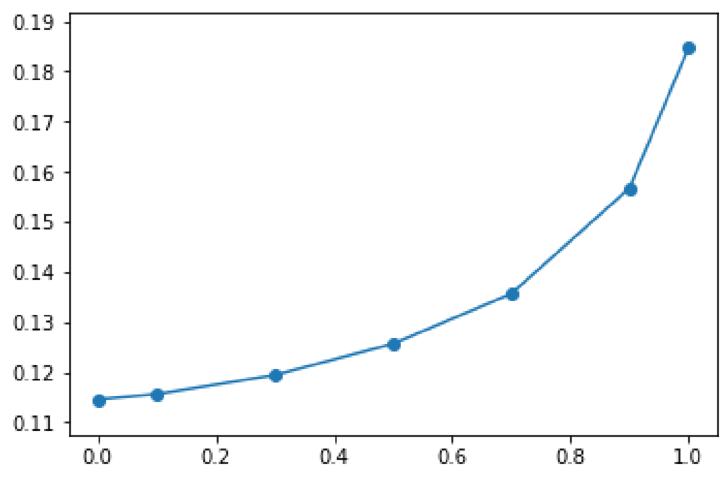


Experiment 1:

Batch Updating (Average error on the random-walk problem under repeated presentations).

The weight vector was not updated after each sequence. Instead, update the delta weight (weight increment) after training completely on a finite amount of data, e.g. train repeatedly on 10 episodes until convergence. In the paper, it says "Each training set was presented repeatedly to each learning procedure until the procedure no longer produced *any significant changes* in the weight vector." The root mean squared (RMS) error of w against the ideal predictions T were calculated and averaged across the 100 training sets.

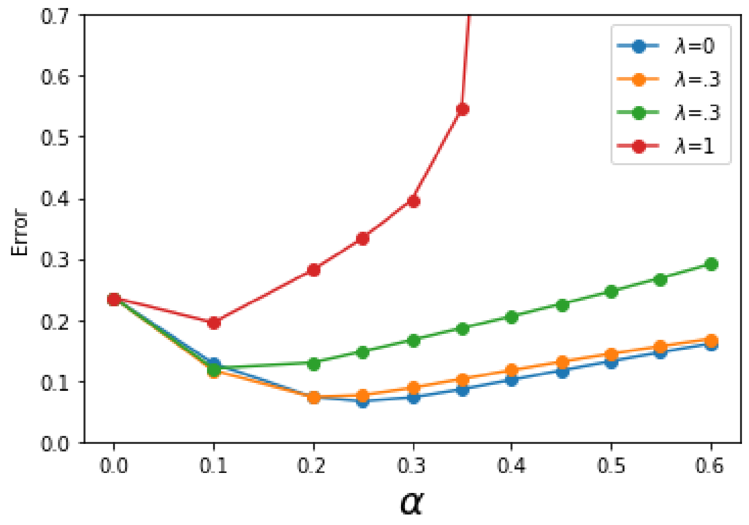
Assumption: Sutton mentioned that TD(λ) will always converge when learning rate is small, but the learning rate or the convergence threshold were not reported. In this case, the assumption here is that the alpha is 0.01. Lambda is between 0.0 and 1.0 with an increment of 0.1. Initialized weights is set to 0.5 for all the nonterminated states. And the convergence happens if the delta weight of this sequence (increment delta) is less than 0.001.



In the experiment, the error increases rapidly as λ approaching to 1 (Figure 2) with TD(1) has the worst performance and TD(0) has the best performance. Two inconsistency are 1) there is no dip in the curve at lambda 0.2. 2) The range of the RMSE was [.11 .19], which is lower than the paper stated. I suspect there are two main reasons for this first is that the convergence threshold is not the same as what Sutton used. This demonstrates that TD(1) only minimizes the error on training dataset but not the future experience.

<u>Experiment 2:</u>

In experiment 2, the algorithm will see the training sets only once update w after each sequence incrementally. It demonstrates the effect of learning rate (α) on the performance of TD(λ). multiple learning rate were applied to the learning algorithm. TD(0) convers to what can be considered the optimal estimates for matching future experience, which is consistent with the MLE of the Markov process.

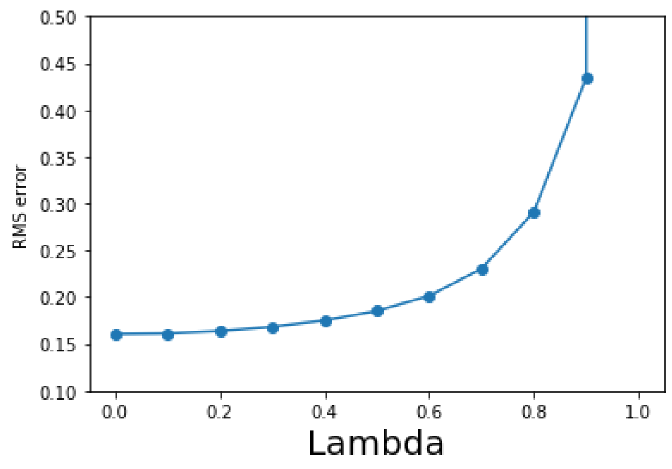


The graph also shows that alpha, the learning rate, plays an essential role with that the midpoint alpha around 0.3 has the best result. TD(1) has the worst estimates with all the alphas in the range. The difference between the replicated graph is that there is no cross-over in between lambda equals 0 and 0.3. The backwards learning of TD(0) might have been slowly captured and not enough episodes.

Experiment 3:

The best error level achieves for each lambda value, that is using alpha value that was best for that lambda value. The learning rate was selected from those that yields the lowest error for that lambda value.

The assumption in the test alpha range and lambda range is the same as experiment 2. Alphas are in the range of [0, 0.6] and lambdas are in the range of [0, 1]



I didn't see the best lambda at 0.3. Lambda in between 0.0 to 0.2 gives a steady increase with the best lambda at the time. It shows a gradual increase in the error as the lambda increases given the best alpha at the time. The speed of error increases takes up at the elbow turning point of 0.6.