Desperately Seeking Sutton

Maksim Kazakov, CS 7642: Project 1

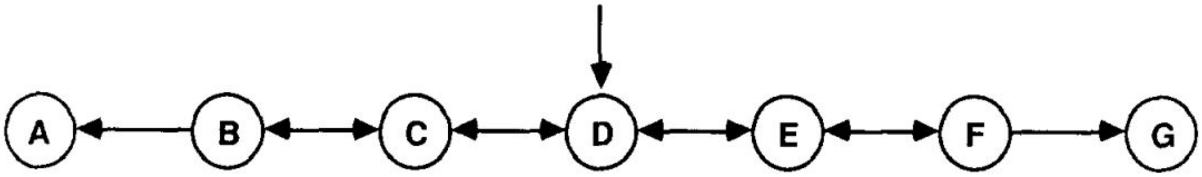
Introduction

Purpose of these project is to reproduce the results presented by Richard Sutton in “Learning to Predict by the Methods of Temporal Differences”. In this project we’ll explore the concept of temporal differences methods, recreate two experiments from Sutton’s original paper and compare the results.

Experiments

Description

Both experiments that will be recreated in this paper use well-known bounded random walk as a data generator. Random walk in these experiments randomly moves to the left or to the right next state. There are five total intermediate states and two terminal states on the sides. Random walk always begins from the middle state and ends when reaches one of the terminal states.



Sequence of the states before the terminal state is considered an observation in training data for supervised learning. And terminal state is the outcome that the learner should predict.

Implementation

For the purpose of these experiments terminal states are assigned numerical values: zero for leftmost state and one the right one. Also all intermediate states are represented by the basis vectors of length five. These vectors have a value 1 assigned to the component with index equal to the order of the corresponding state counting from left, other components in these vectors are equal to zero.

To get a statistically reliable data all experiments were done and results were averaged across 100 training sets.

Both experiments use LMS method to learn weights that can be used to make an accurate prediction of the future outcome using states vector representation. LMS relies on calculating updates for weights and for that Sutton used TD(λ) rules, that allows us to calculate updates while iterating through the sequences of observations.

$$\Delta w_t = \alpha(P_{t+1} - P_t)e_t$$

$$e_{t+1} = \sum_{k=1}^{t+1} \lambda^{t+1-k} x_k = \lambda e_t + x_{t+1}$$

Using these methods significantly reduces amount of needed memory and computational power.

Experiment 1

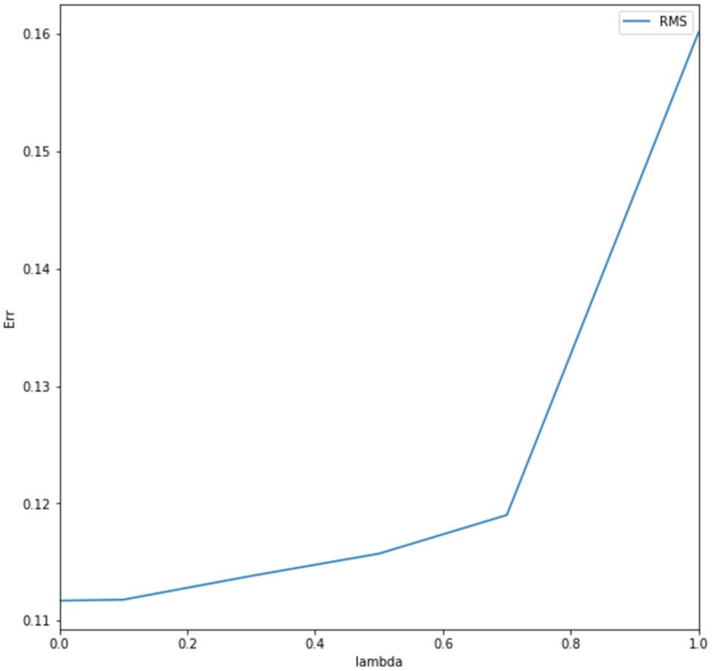
Though TD(λ) rules can be used to update weights immediately and not wait for the outcome results in experiment 1 Sutton updates weights only after observing all sequences in training set.

Also in experiment 1 repeated presentation training paradigm was used when the same training set is used several times to train a learner until it converges. Though Sutton didn’t specify in his work how it is determined if learner has converged we use following commonly used method: if norm of the new weight update vector is smaller than some preset small value then we conclude that weights have converged.

To estimate learners performance Sutton used RMSE between converged weights and probabilities of ending in rightmost state for each intermediate states. The latter can be calculated mathematically and for our setup its [⅙, ⅓, ½, ⅔, ⅚].

Also original paper doesn’t specify how weights are initialized in experiment 1, so in this reproduction weights were randomly generated with uniform distribution in range [0, 1).

Following figure is similar to figure 3 in original paper where we calculate averaged error for different λ values for $TD(\lambda)$ rule.



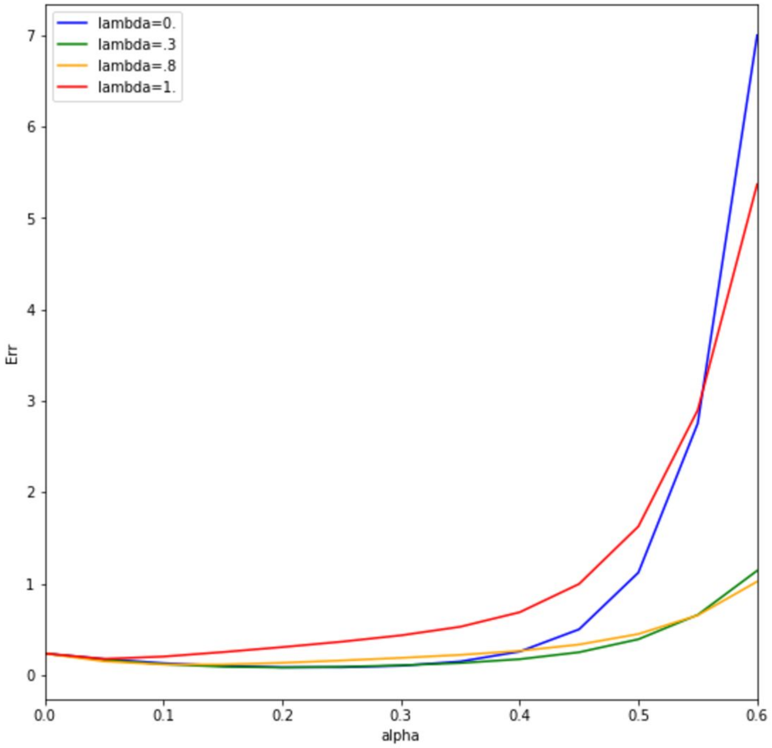
Just like in the original work we see that error significantly reduces as we move from $\lambda = 1$ (standard delta rule method) to $\lambda = 0$ (extreme TD method when we only use previous predictions to calculate weight updates).
Also variance in errors for all lambdas is below 0.08, which is higher than the ones declared in Sutton's work but still not big enough to question the overall results. This discrepancy can be explained by Sutton using more reliable method to determine convergence or having more strict convergence threshold.

Experiment 2

Experiment 2 explores in greater details the influence of learning rate on final result as well as applies some significant implementation changes compared to Experiment 1:

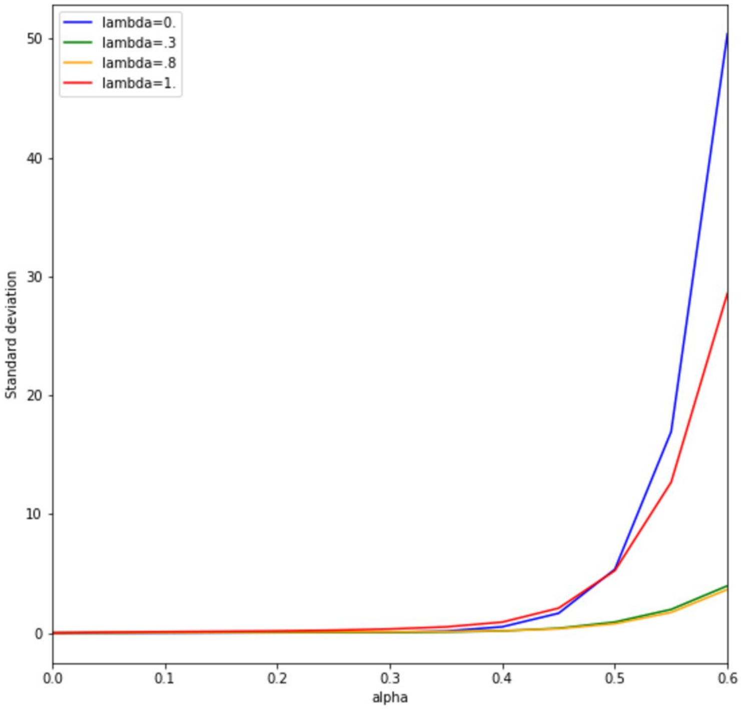
1) Each training sample presented to learner only once unlike repeated presentation in experiment 1;
2) Weight updates are accumulated across sequences and applied immediately after each sequence;
3) All weights are initialized with value 0.5;
4) Each $TD(\lambda)$ procedure was repeated for different values of α.

The results of this experiment are presented on the following graph which is equivalent to figure 4 from original work. Here for each $TD(\lambda)$ procedure shown a dependency of the final error and learning rate α.



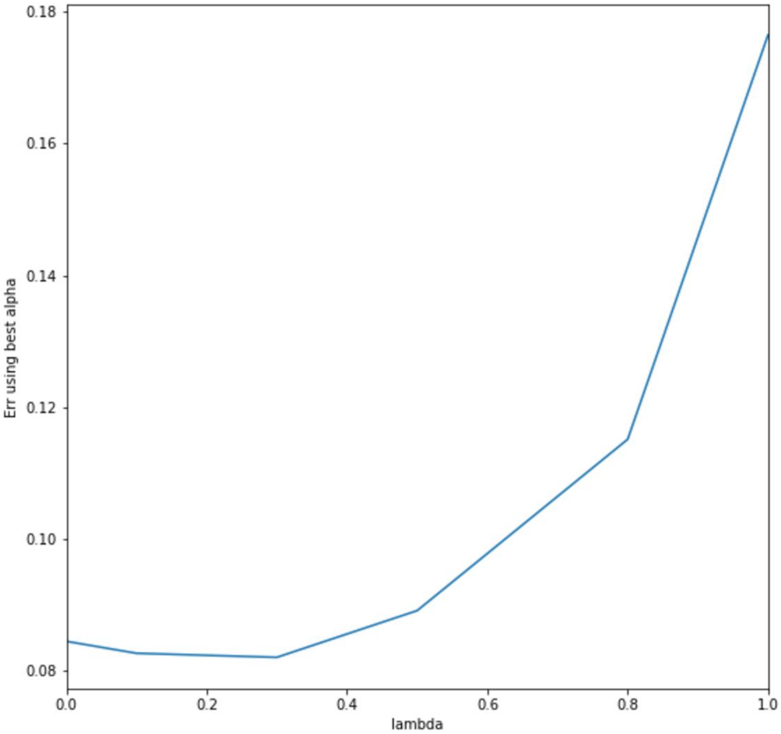
Though it's not as obvious as in the original paper, but we can see that best optimal learning rate for all $TD(\lambda)$ procedures lies somewhere in the range [0, 0.4], but it's different for each λ.

Also unlike original paper $\lambda = 0$ doesn't dominate over $\lambda = 1$ on all α values. But this can be explained by significant increase in variance after $\alpha = 0.45$ for all λ, and especially for $\lambda = 0$.



At $\alpha = 0.6$ variance is so big that we can't actually conclude anything about $\lambda = 0$ compared to other values.

By picking best alpha value for each $TD(\lambda)$ procedure we can create a graph similar to the one from experiment 1 and equivalent to figure 5 in original paper.



Just like in original work we can see that optimal lambda lies somewhere in the area of $\lambda = 0.3$, though difference between it and TD(0) is not as significant in this replication. This phenomenon was explained by Sutton and verified in this work by the fact that TD(0.3) learns faster than TD(0). Also it was noticed that with all these implementation changes in Experiment 2 we were able not only improve on error but also significantly save of time resources needed to train the model.

Pitfalls

Though Sutton included very detailed description of the experiments in his paper some details were still missing to reproduce the original experiments in the exact manner.

One of the biggest issues was the divergence of the weights on big alphas in experiment 1. It can be explained by big differences in predictions when constantly switching back and forth between states in some sequences, which were later repeated during repeated training on the same set. Sutton doesn't mention if such problem occurred in his work and what method was used to counter it. This is probably because he used smaller training rates and that's what helped to tackle this problem in this work.

Also, as it was mentioned earlier, Sutton didn't specify what method was used to decide when weights have converged. That might be the reason why some results are a little bit different from original experiments even though they show the same behavioural patterns.