We consider multi-step prediction problems in which experience comes in observation-outcome sequences of the form x1, x2, x3, …xm, z, where each **xt is a vector of observations** available at **time t** in the sequence, and **z is the outcome** of the sequence.

For each observation-outcome sequence, the learner produces a corresponding sequence of predictions P1, P2, P3…, Pm, each of which is **an estimate of z.**

The predictions are also based on a **vector of modifiable parameters or weights, ω.**

For each non-terminal state *i*, there was a corresponding observation vector Χ*i* ; if the walk was in state *i* at time t then xt = Χ*i* .

All learning procedures will be expressed as rules for updating ω. For the

moment we assume that **ω is updated only once for each complete observation-outcome sequence** and thus does not change during a sequence.

In the first experiment, the weight vector was not updated after each sequence

as indicated by (1). Instead, the Δω’s were accumulated over sequences and only

used to update the weight vector after the complete presentation of a training set.

Each training set was presented repeatedly to each learning procedure until the

procedure no longer produced any significant changes in the weight vector.

The second experiment concerns the question of learning rate when the training set is presented just once rather than repeatedly until convergence. Although it is difficult to prove a theorem concerning learning rate, it is easy to

perform the relevant computational experiment. We presented the same data to

the learning procedures, again for several values of λ, with the following procedural changes:

* First, each training set was presented once to each procedure.
* Second, weight updates were performed after each sequence (i.e., as in (1)) rather than after each complete training set.
* Third, each learning procedure was applied with a range of values for the learning-rate parameter α.
* Fourth, so that there was no bias either toward right side or left side terminations, all components of the weight vector were initially set to 0.5.

Piazza 185

Piazza 291

Piazza 229

# How do I back-propagate TD Error to previous states based on the eligibility trace?

Sutton textbook shows Zt = Lambda(Zt-1) + gradient(St,wt). I am unclear how they backpropagate across several states.. it appears like they are only doing it to the one state. What exactly is the gradient portion of the formula?

The semi-gradient TD(Lambda) pseudocode on P293 does not mention decaying the eligibility trace, only 12.5 does, and it only seems to do it on one step from t to t-1.

[project1](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

[**edit**](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)·[good question](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)2

Updated 7 days ago by

Abhinav Agarwalla

 and [2 others](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

**the students' answer,**

*where students collectively construct a single answer*

Click to start off the wiki answer

**followup discussions**

*for lingering questions and comments*

**1 endorsed followup comment**

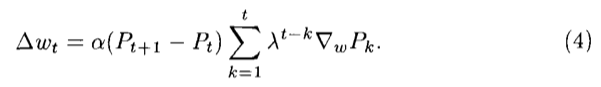
Resolved Unresolved

/var/folders/m2/02s5_6cx2gnfg7__3r24l_ph0000gp/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/1481328003_35.png

**[Deniz Sanli](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

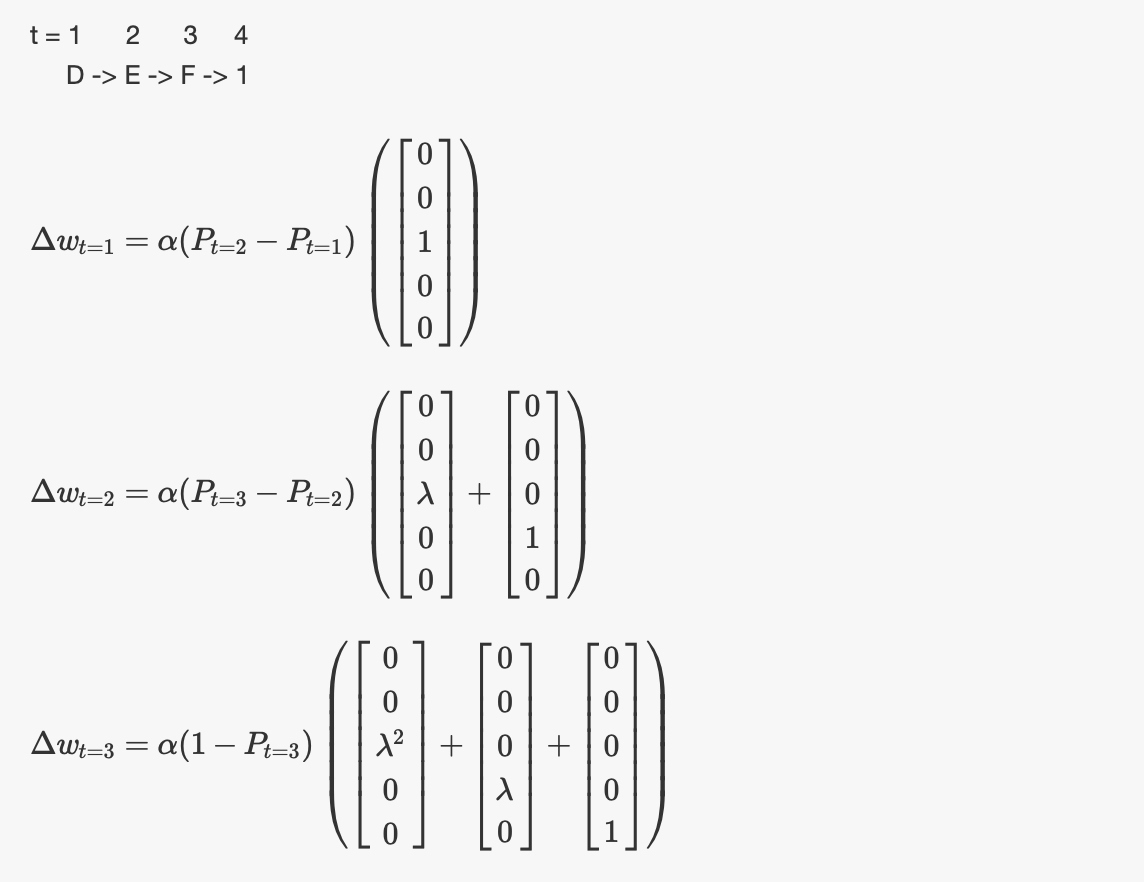
[11 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

The gradient of a prediction at t = k is the partial derivatives of the prediction at t = k with respect to the weights in the weight vector. This is simply the observation vector xi at t = k.



Let's work with the equation in Sutton's paper (p15).

Let's also assume the episode:



If we let λ = 1, then we are modifying all previously seen states to the recent error. If λ = 0, then we are only modifying the state at the previous time step.

By repeating this procedure with many episodes (or the same episode) until convergence, we can propagate the outcome z = 1 to the previously seen states.

**~ An instructor (**

**Vahe Hagopian**

**) thinks this is a good comment ~**

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 10



**[Iraklis Koutrouvelis](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[11 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

Thank you so much Deniz! This was extremely helpful!

I was wondering what happens in the case of repeated states? Does the above explanation still hold?

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0

/var/folders/m2/02s5_6cx2gnfg7__3r24l_ph0000gp/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/1503244998_35.png

**[David Schmidlin](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[10 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

In this scenario is P\_t the Reward or the Value

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0

/var/folders/m2/02s5_6cx2gnfg7__3r24l_ph0000gp/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/1481328003_35.png

**[Deniz Sanli](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[10 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

In this scenario, Pt is the value or prediction of the state, which is desired to be equal to the probability of receiving an outcome = 1 starting at that state.

That's a great question Iraklis. I believe that this still holds in the case of repeated states.

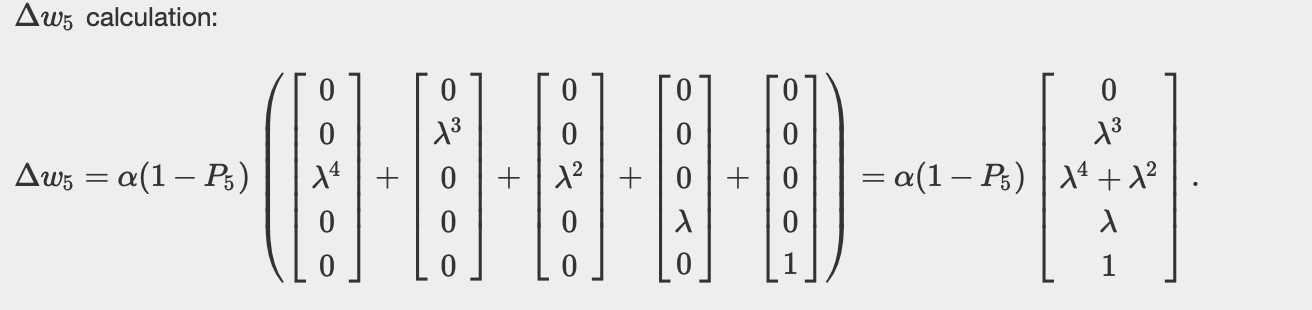
[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 1



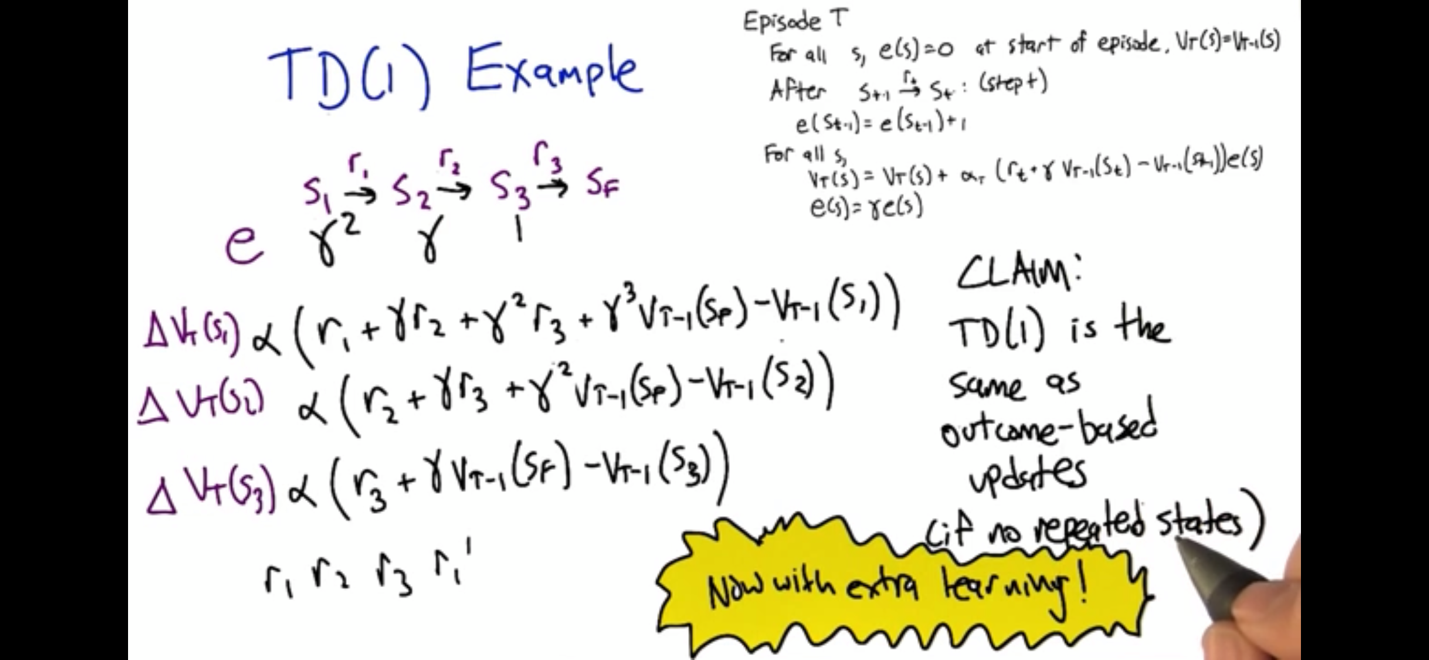
**[Jake Knigge](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[9 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

Say that we had the episode T=(x1=D,x2=C,x3=D,x4=E,x5=F,x6=1). We pass through D twice, so we (should) learn more about D; in RL-dialect, D becomes eligible twice. We can see this in the Δw5 calculation:



I believe that's the idea conveyed in one of the "TD and Friends" lectures: "Now with extra learning!". See the below screenshot.



[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 4

/var/folders/m2/02s5_6cx2gnfg7__3r24l_ph0000gp/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/1481328003_35.png

**[Deniz Sanli](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[9 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

That's a great analysis Jake. I would also like to add that these updates are performed at every time step. This means that state D's prediction had been updated many times already, and so on the revisit, the error update between it and the previous state would be more effective.

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 1



**[Jake Knigge](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[9 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

Thanks Deniz! I was just riffing off of the helpful example that you'd put together.

And yes, great clarification regarding the fact that state D's prediction (that is, its weight or its value estimate) would have already been updated multiple times.

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0



**[Will Beckman](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[9 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

this is immensely helpful - thank you guys!

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0



**[Brian Vincent](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[8 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

So, just to try to make sure I understand this correctly, if we end up with an episode which ends on the left (z=0), the same calculations still apply similarly but the result till ultimately decrease the weight?

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0

/var/folders/m2/02s5_6cx2gnfg7__3r24l_ph0000gp/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/1481328003_35.png

**[Deniz Sanli](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[7 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

Yes Brian, that should be the case.

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0



**[Nicholas Jordan Mitchell](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[7 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

With respect to the values for PT=2, this is basically the 2nd component in the w array?

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0



**[Jake Knigge](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[7 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) Good question. We define Pt=2 as

Pt=2=wTxt=2=∑iwixi,t=2=wj,

where j denotes the nonzero element of xt=2, which may or may not be the second component of xt=2. The nonzero element will correspond to the state at t=2.

Also note that the *weight* in Sutton’s 1988 terminology corresponds to the *value of the state* (in our terminology): wSt=2=V(St=2).

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 2



**[Nicholas Jordan Mitchell](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[7 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

With respect to the last thing you said, will this value be either 1 or 0 depending on weather or not the final state of the sequence is G or A?

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0



**[Jake Knigge](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[7 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) The weights/values depend on how we initialize them. They should converge to the “true values” after we run the TD algorithm. Sutton initialized the weights/values for non-terminal states to be 0.5.

The value of the final prediction PT (or z in Sutton’s notation) will be 0 (if we ended on the left in state A) or 1 (if we ended on the right in state G).

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 2



**[Sankaranarayana Ramamoorthy](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[6 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

Correct me if I am wrong, So at least for first iteration the first sequence only w4 will have a value as initially every weight is 0.5 .

and if the weight vector is updated only after all the 10 sequences are completed then only    wsizeofsequence in each sequence will have a value.   
and if any one of these sequences does not go to G it will not be counted

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0



**[Jake Knigge](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[6 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) Yep, that’s right. We (at least in replicating figures 4 and 5) would initialize the weights to have initial values of 0.5.

The length (that is, number of elements/entries) of w is the number of states, not the length of the sequence.

To your point, when a sequence ends at state G, we would expect (other things equal) that the entries of w to increase. When a sequence ends in state A, then we would expect (some of) the entries of w to decrease.

Does that help?

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 1



**[Brian Vincent](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[5 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

Jake - you lost me on "the length of w is the number of states, not the length of the sequence".

I've gotten through the first iteration of my sequences, but all the sequences (for the most part) vary in length and "w" is indexed by time, correct?  Or, by number of states do you mean basically how many steps you took (which would make sense to me)?

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0



**[Jake Knigge](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[5 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

Good clarifying question, Brian. You’re right that w is indexed by time and we update by accumulating Δwt over episodes.

With "the length of w is the number of states, not the length of the sequence,” I meant to convey that w∈Rd, where d is the number of states. So I should’ve said that the dimension of w is the number of states.

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 2



**[Brian Vincent](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[5 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

Also, to clarify the statement above from Sankaranarayana Ramamoorthy, is it really true that episodes that don't go to 'G' (aka go to 'A') should not be counted?  Shouldn't the math, in theory, decrease the predicted value of states then?

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0



**[Jake Knigge](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[5 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) I guess I’d say it a bit differently: it’s not that they shouldn’t be counted, but they wouldn’t increase the value estimates.

So like you said: if a trajectory/sequence goes to A, then we would expect that the value estimates would decrease.

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 4



**[Travis](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[1 day ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

So it appears, based upon this discussion, my experiments,  and my reading of the 1988 paper that gammas should be 1 for this project. Is that correct? If we included gamma in the one hot examples, it would be lambda \* gamma with each to their respective power instead of just lambda, correct?

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1) 0

/var/folders/m2/02s5_6cx2gnfg7__3r24l_ph0000gp/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/1503244998_35.png

**[David Schmidlin](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)**

[23 hours ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=229_f1)

I am setting gamma to 1.0

Piazza 176

learning rate decay

The lecture talks about learning rate has to be decayed with something like α/T, why there isn't any decay in Sutton's paper?

Also, it looks to me that with a fixed α, the ω will fluctuate forever, get influenced by the most recent sequence it sees. If we want a converged ω, we either have to decay the learning rate, or average over multiple training set. Is my understanding correct?

[project1](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)

[**edit**](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)·[good question](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)2

Updated 16 days ago by

Qinghui Ge

**the students' answer,**

*where students collectively construct a single answer*

Hey! I missed your question when I posted my own, I'm trying to start a conversation on the same issue in [@180](https://piazza.com/class/k4ws6a8i2uu71u?cid=180). I think you're right - I didn't initially decay the learning rate and I wasn't getting stability.

[**edit**](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)·[thanks!](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)0

Updated 15 days ago by

David Leber

**the instructors' answer,**

*where instructors collectively construct a single answer*

yes for a large alpha if you don't decay it will fluctuate at a magnitude of alpha \* td error of the sample, this error can be large even for converged w because of the randomness. so yeah decay it or use a averge td error to kill the randomness or use a very small alpha.

[thanks!](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)2

Updated 14 days ago by

tianhang zhu

**followup discussions**

*for lingering questions and comments*

Resolved Unresolved



**[Iraklis Koutrouvelis](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)**

[11 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)

Hi! I am getting the same issue but not sure I understand the response. If we were to decay 'alpha', wouldn't that defeat the objective of the project (i.e. replicate experiment results)? Also, if the learning rate is really small, will that always converge to the optimum value for all lambda?

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=172) 0



**[tianhang zhu](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)**

[11 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=172) the only theoretically plausible rates are the alpha/T alike, however, theory is simplification right so alpha/T may not work as well in practice seldomly. constant small usually works in practice. why would that defeat the goal of the project. the paper didn't specify the rate right

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=172) 0

/var/folders/m2/02s5_6cx2gnfg7__3r24l_ph0000gp/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/1481328003_35.png

**[Deniz Sanli](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)**

[11 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)

I think what OP means by defeating the objective of the project is that if Sutton never explicitly mentioned decaying alpha in his experiment, then we are technically not repeating the experiment--we are making assumptions.

Unless the true goal of the assignment is to make assumptions and defend them, which seems to be more likely at this point.

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=172) 0



**[tianhang zhu](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)**

[11 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=172) correct, making assumptions are crucial part of this work. simply because it's not stated and you still want the same result

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=172) 1



**[Paul Bockelmann](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)**

[11 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)

I think it was stated in the project assignement description that we will find that we need to make assumptions. If we do, then we should write about what assumptions we made and why we made them the way we did.

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=172) 1

/var/folders/m2/02s5_6cx2gnfg7__3r24l_ph0000gp/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/1546887719_35.png

**[Bruce Nielson](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)**

[9 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)

For figure 4, you're only running across 10 samples and then averaging 100 times. What exactly are you decaying alpha over? Each sample? Each state changes?

[helpful!](https://piazza.com/class/k4ws6a8i2uu71u?cid=172) 0

/var/folders/m2/02s5_6cx2gnfg7__3r24l_ph0000gp/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/1481328003_35.png

**[Deniz Sanli](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)**

[7 days ago](https://piazza.com/class/k4ws6a8i2uu71u?cid=172)

Bruce, any of the above. Decaying alpha in any way may or may not have been what Sutton did so it doesn't really matter. Some of us have gotten closer results by decaying within episodes and between episodes, but from what I understand, the correct assumption is the one that you can best defend. Of course, there may be other ways to replicate his results as well.

Piazza 173:

Figure 3 convergence

I'm trying to interpret the correct way to generate figure 3 from Sutton 1988. One point of confusion is the repeated presentations for training. Does this mean that a particular training set is repeated until convergence, then the next training set separately? This would result in 100 different weight vectors to calculate the RSME. Or is the right way to interpret it to use all training datasets to get to one converged weight vector?

**the students' answer,**

*where students collectively construct a single answer*

There are 100 training set

Each training set has 10 sequences.

For each training set:

  Repeat cyclically (1,2, ..., 9,10, 1,2 ...) its ten sequences, update weight every 10 sequences, until the weight is converged.

  Calculated RMS error based on this converged weights vector

Final RMS (point on the graph) is the average of the 100 above rms measures.

Piazza 176

# learning rate decay

The lecture talks about learning rate has to be decayed with something like α/T, why there isn't any decay in Sutton's paper?

Also, it looks to me that with a fixed α, the ω will fluctuate forever, get influenced by the most recent sequence it sees. If we want a converged ω, we either have to decay the learning rate, or average over multiple training set. Is my understanding correct?

[project1](https://piazza.com/class/k4ws6a8i2uu71u?cid=237)

[**edit**](https://piazza.com/class/k4ws6a8i2uu71u?cid=237)·[good question](https://piazza.com/class/k4ws6a8i2uu71u?cid=237)2

Updated 16 days ago by

Qinghui Ge

**the students' answer,**

*where students collectively construct a single answer*

Hey! I missed your question when I posted my own, I'm trying to start a conversation on the same issue in [@180](https://piazza.com/class/k4ws6a8i2uu71u?cid=180). I think you're right - I didn't initially decay the learning rate and I wasn't getting stability.

[**edit**](https://piazza.com/class/k4ws6a8i2uu71u?cid=237)·[thanks!](https://piazza.com/class/k4ws6a8i2uu71u?cid=237)0

Updated 16 days ago by

David Leber

**the instructors' answer,**

*where instructors collectively construct a single answer*

yes for a large alpha if you don't decay it will fluctuate at a magnitude of alpha \* td error of the sample, this error can be large even for converged w because of the randomness. so yeah decay it or use a averge td error to kill the randomness or use a very small alpha.

Piazza 259:

Error calculation for Fig 4. / initialization of weights or V

I~~am suddenly very confused~~ I know now about which error we use for creating the graphs:

If we take Figure 4, where the error is plotted over different learning rates and lambdas,

then I think that if we have alpha = 0. (leftmost point) then the algorithm doesn't learn anything.

If we initialize V or w with all zeros, then the mean squared error would produce ~ 0.218, the root mse ~ 0.467.

The leftmost point from Fig. 4 shows roughly a value of 0.23 - it doesn't really fit to any of the above values.

The only thing that now would produce this error would be an initialization of w or V to non-zero values.

**... Ah, ok, nice, after reading the text once again... \*Ehhm\* - weights get initialized by 0.5 each.**

That also means that the error calculation does **not include the both terminal states**! -> error = 0.236

