INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

REINFORCEMENT LEARNING

Eligibility Traces

with

Prof.Balaraman Ravindran

Department of Computer Science and Engineering

Indian Initiation of technology, Madras

So so we looked at TD methods in the last class right so we looked at a very, very popular Q-learning and serve sir we looked at the differences between Q-learning and serve sir when it comes to learning from exploratory trajectories and other things right. So I should mention that, though Q-learning answers are kind of a very popular reinforcement learning algorithms you know how when people talk about using neural networks almost everybody talks about using back proper eight even though if you look at actually ask the neural network community itself people will say off only back off.

I mean let us do something more fancy okay that is not what it is but anybody outside of that community home server is trying to use it will end up using back propagation as they to likewise anybody outside the rural community is trying to use reinforcement and I end up using Q-learning assess their first tool of choice right so if you are looking to solve problems and even care if your work is going to get published or are you are not here on care about getting it published in a computer science.

When you write then that is probably what you would do as well right, but then the whole bunch of other things that kind of engages the attention of the reinforcement learning community itself right so a whole variety of other algorithms including, you see trees which she spoke about very briefly so that's also used very popular know.

And other methods where the aim is to reduce the sample complexity you know how many samples do you need to learn effectively right so the aim is to reduce those kinds of sample complexity and so those kinds of algorithms are getting more attention right. So what we will do today is talk about, one way of trying to speedup convergence in TD algorithms there has been around forever it is not it's not something new it's been around forever and then perhaps maybe in the next lecture or to talk about some of the newer variants of this right.

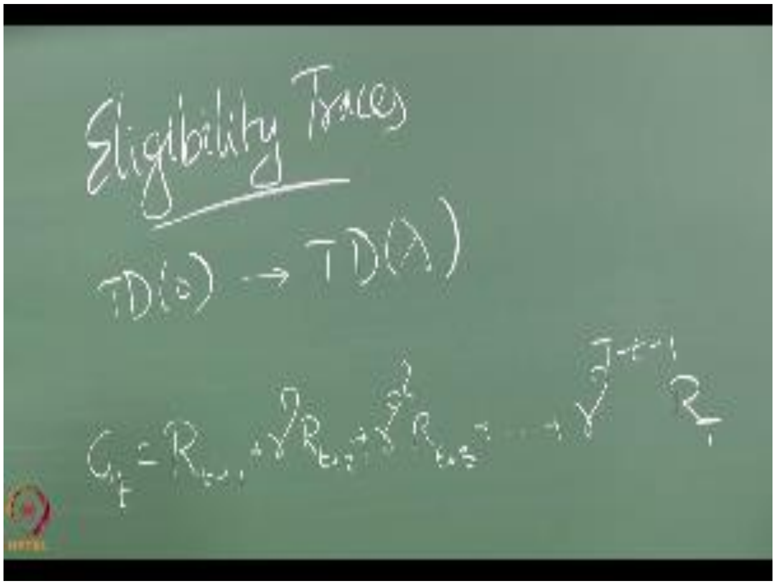
(Refer Slide Time: 2.27)

So I'm going to talk to you about what are known as eligibility traces right. So you remember the, the basic TD algorithm that I wrote to have the policy evaluation algorithm right so I said it is called TD0 okay.

(Refer Slide Time:2.55)

So that is where the thing comes now so we will go from CD 0 now we will talk about something small TD lambda all right so where lambda is a parameter that you have to choose depending on the nature of a problem so before getting into TD lambda right. I’ll talk about a couple of let's go back and look at the return definition that we have had so far right.

(Refer Slide Time: 3.22)

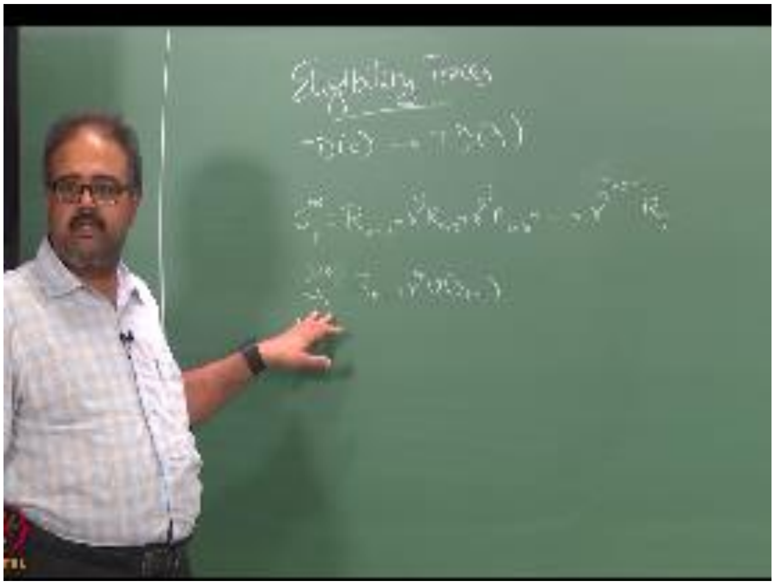

Eligibility Traces
TD(0) → TD(λ)
$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$

So what is the definition of return that we used right T-1 or T +1, T- 1.So this is essentially what we have right so this is the definition of return that we have assuming that it stops at some point and if it does not stop it you just keep going now assuming let's assume that we have episodic has together some issues that will come up with episodic tasks that I want to talk about right .

(Refer Slide Time: 4.20)

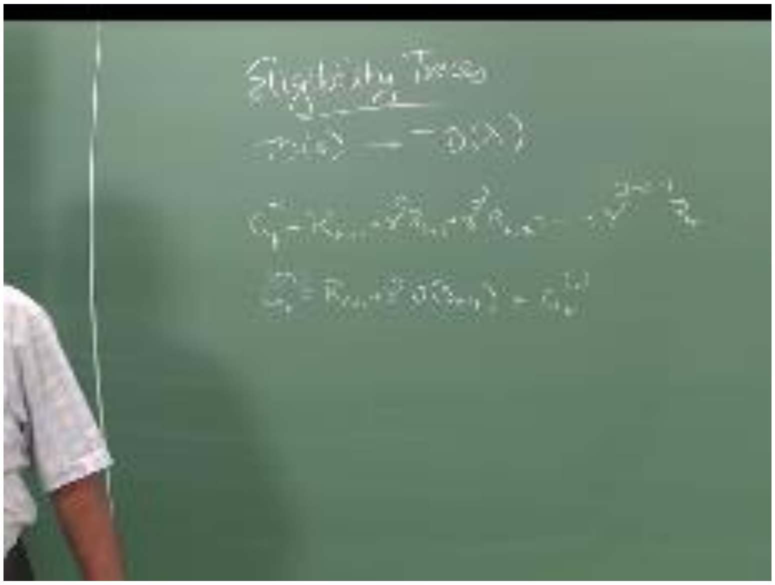
So what was the so I am going to call this the Monte Carlo return I mean this is the full return but this is the return that we use when we did Monte Carlo learning right so you could think of designing a learning algorithm where I look at my current estimate of the value function right plus alpha times GTM C- the current estimate right.

(Refer Slide Time: 5.00)



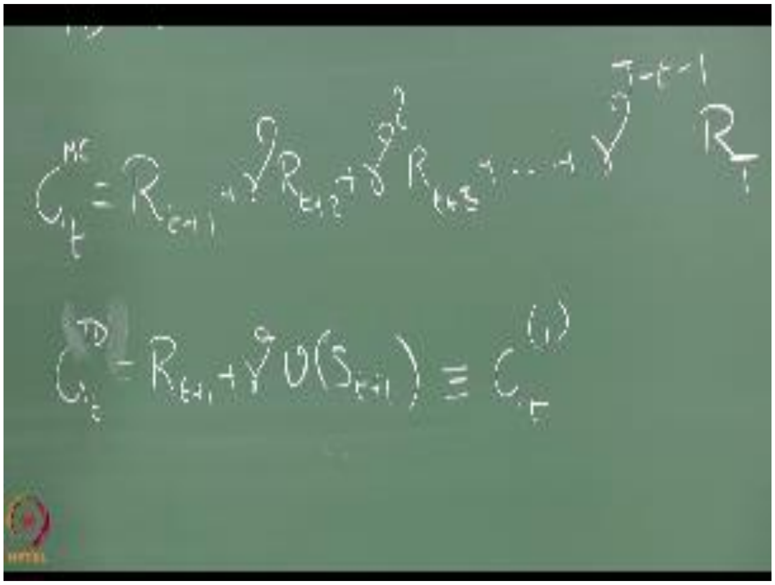
And then what was the return that we use for TD right so we used to call this the TD return right so we remember we use this earlier so I am going to introduce another notation for this I am going to also call this

(Refer Slide Time: 5.24)


Eligibility Traces
TD(0) → TD(λ)
$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots \gamma^{T-t-1} R_T$
$G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1}) = G_t^{(1)}$

The, the one-step return right so, just RT + 1 will be the one step reward this is the one step return or more correctly you should call it the one step truncated corrected return okay. So it is one step truncated return because we take the film return and truncated edited one step right and that then we correct it by adding a VST + 1 this one is called the one-step truncated corrected written.

(Refer Slide Time: 6.03)

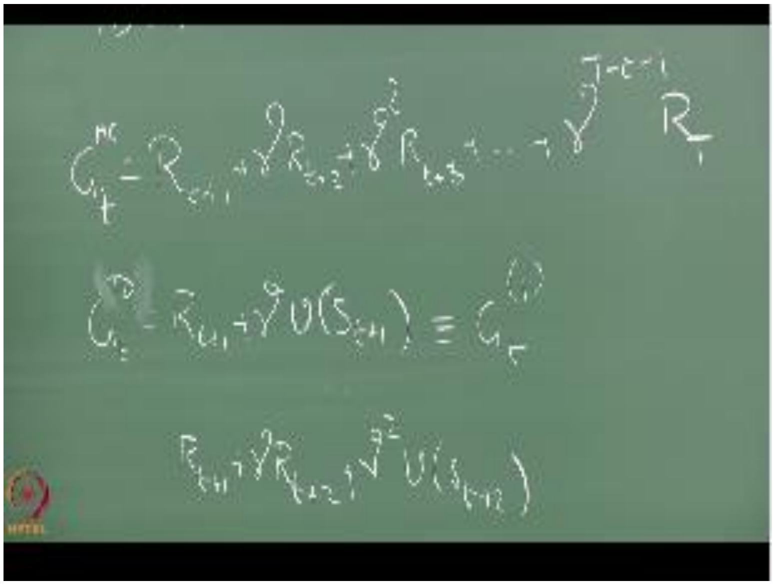


$$G_t^{MC} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots + \gamma^{T-t-1} R_T$$

$$G_t^{TD} = R_{t+1} + \gamma V(S_{t+1}) \equiv G_t^{(1)}$$

But sometimes we call it the one-step return right so last class I was talking about how you could actually think of

(Refer Slide Time: 6.10)



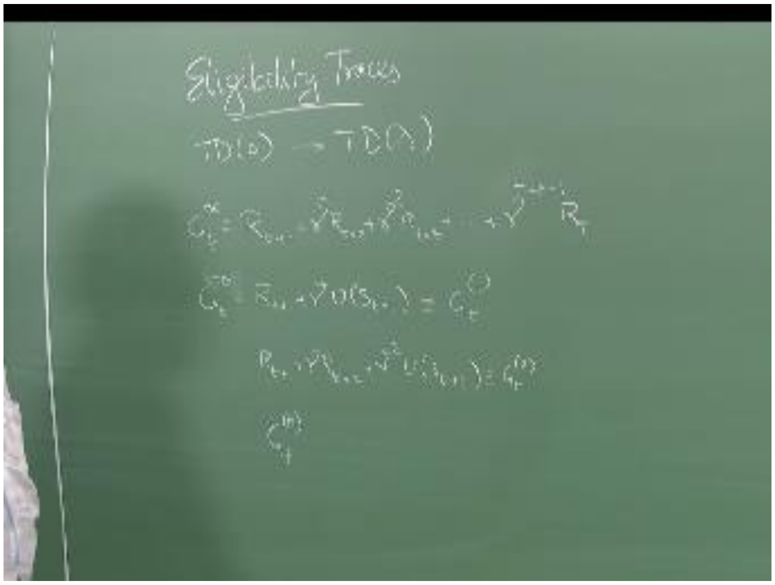
$$G_t^{MC} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots + \gamma^{T-t-1} R_T$$

$$G_t^{TD} = R_{t+1} + \gamma V(S_{t+1}) \equiv G_t^{(1)}$$

$$R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})$$

Other things right like this right so. What do you think that is called to step 38 corrected whatever to step with it right so

(Refer Slide Time: 6.32)



Eligibility Traces

TD(0) → TD(λ)

$G_t^{\infty} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$

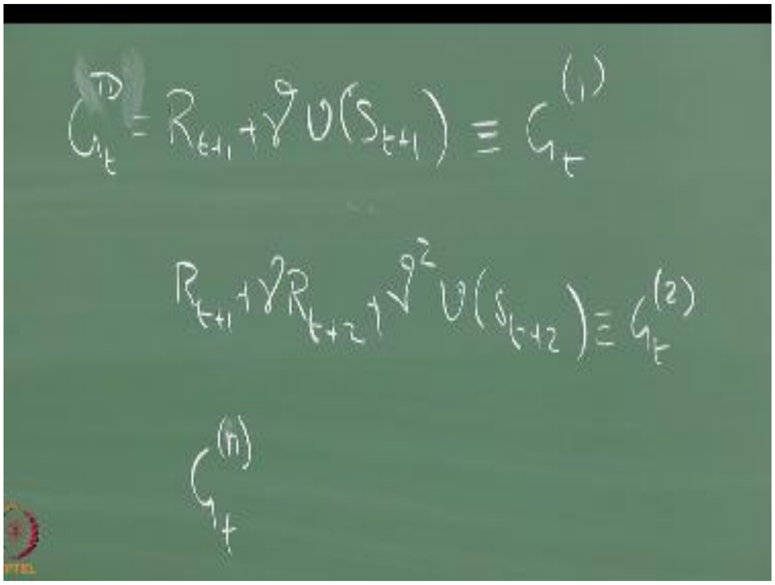
$G_t^{TD} = R_{t+1} + \gamma v(S_{t+1}) \equiv G_t^{(1)}$

$R_{t+1} + \gamma R_{t+2} + \gamma^2 v(S_{t+2}) \equiv G_t^{(2)}$

$G_t^{(n)}$

I am going to note this night so in general I can think of I can think of any instep truncated corrected returns and I told you that there is appear to be certain tasks for which a definition of return other than the TD return other than the one-step truncated returns seem to do well right and of course there might be some tasks for which the full written thus well

(Refer Slide Time: 7.13)



$G_t^{TD} = R_{t+1} + \gamma v(S_{t+1}) \equiv G_t^{(1)}$

$R_{t+1} + \gamma R_{t+2} + \gamma^2 v(S_{t+2}) \equiv G_t^{(2)}$

$G_t^{(n)}$

So what is the N , I should choose for a given task so one of the things I don't know how many of you have been reading ahead in the book how many of you have been reading ahead in the book. My lord, so I do not use slides and unfortunately we are not having our TV screens in operation today so here is a picture so when I change the Alpha what is alpha, alpha times that update parameter right so alpha times GT(n) GT n - V Л.

So as I keep changing the Alpha each of this curve did not set different value of N as I keep changing the Alpha so the prediction error that I finally get after all the learning is done right I finished training everything so the prediction error I get for different values of N write fiction error is the true value functionary so for different values of N the prediction error.

I get is plotted here and it turns out you get the minimum prediction error for N of 3 for this problem for 3 or 5 you could use one of those right. So each of this curves is for a fixed alpha for different values of N okay I, I do the updates right and then I finally get the converged value for my TD updates and what is a prediction error at the convergence point rate so this is each of these curves and you can see that for the one step truncated return which is here so you can see that the prediction error at convergence is rather bad right.

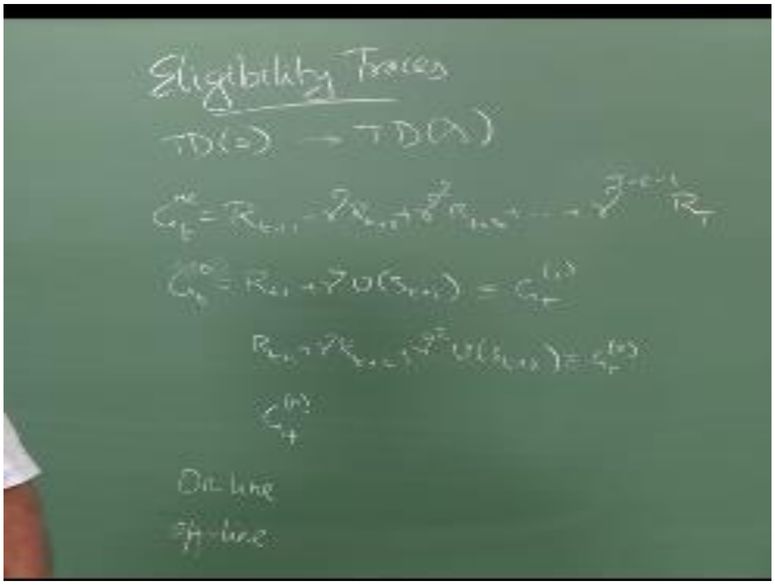
So it is because that's the lowest it reaches why for higher values of N right it keeps getting lower and lower but at some point it starts going back up again right so this is 3 this is 5 and then 8 12 just keep going up again so it turns out to be that that seems to be some intermediate values of n for a given problem that seemed to do well.

The true value function, is simple enough MDP that I can actually compute the true value function right I can see the difference between these kinds of graphs right if you remember when you still talking about constant alpha graphs I generate a certain number of episodes right and then I keep running those episodes again and again until I converge.

So that is essentially what we are doing here right so I generate a set number of emphasis is final episode finite data case right but then I can solve dynamic programming on knowing the film MDP and I can get the true value function so in this case I do not do dynamic programming I only take samples from the true MVP and based on the samples I am trying to make an estimate right but to find the true value function I just do dynamic program because I know the MVP.

MVP is simple enough it's just a simple random walk MDP so it's simple enough to estimate so that is basic so different values of N gives you different things like there are a couple of other things which I want to mention here now that we have

(Refer Slide Time: 10.39)



Eligibility Traces

TD(0) → TD(λ)

$G_t^{mc} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots + \gamma^{T-t-1} R_T$

$G_t^{(0)} = R_{t+1} + \gamma V(S_{t+1}) = G_t^{(1)}$

$R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) = G_t^{(2)}$

$G_t^{(n)}$

On line

Off-line

This end so one thing talk about is online versus offline if we have a conversation about online versus offline earlier not on policy enough policy online and offline yeah. So we spoke about this so in this case what is online so I wait till n steps see at end of n steps I can compute GTN rate I can't compute it immediately I need to wait for n steps to compute GTN so as soon as you finishing steps I make that update right so offline this regardless of when I finish computing GTN.

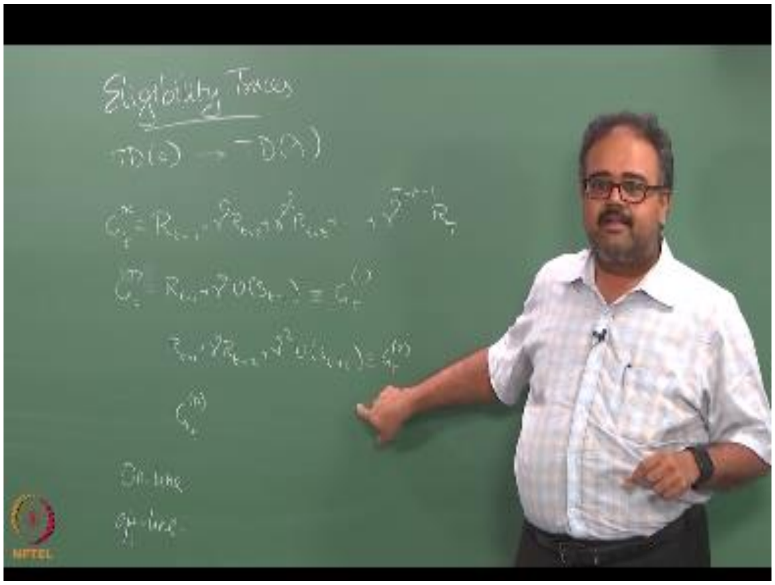
I wait till the end of the episode and I make the updates right so why is one so would that be any changes in the online versus offline cases you would see why would you see a change in the offline updates versus online updates. Yeah anything else for what computation will be less than the online case so it is not as straightforward as you think no convergence might not be faster in online case okay so you need to always be a min guys who did ml should know this mean you should always be aware of bias variance trade-off sphere right.

So if you are doing online updates the variance in the updates you make would be higher if you're doing offline updates essentially what you would do is you will take all the updates so every time you visited that state right you will compute the GTN for this so you will be combining many such updates and will be making one average update for veteran so sometimes you might increase the value sometimes you might want to decrease the value of the written might be higher than the current value the return might be lower than the current value said there may be multiple times you hit the state.

And if you never revisit the state right online offline no difference it only an offline makes a difference only if you are listening the state multiple times right so I am going to take a some trajectory that goes from say that end of the room to this end of the room never coming back to the state again right doesn't matter whether it wait till the end or whether updated along the way so if I update it along the way why it matters.

Same state what will happen yeah I am talking about the valuation here I'll use the new what value function for what for plugging in here

(Refer Slide Time: 13.32)


Eligibility Traces
TD(0) → TD(λ)
$G_t^{*} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots + \gamma^{T-t-1} R_T$
$G_t^{(1)} = R_{t+1} + \gamma v(S_{t+1}) \equiv G_t^{(1)}$
$R_{t+1} + \gamma R_{t+2} + \gamma^2 v(S_{t+2}) = G_t^{(2)}$
$G_t^{(n)}$
On-line
Off-line
NPTEL

That I am not using it for his decision-making or anything right so some of you said you lose it like for juicing action so that is not the right thing so I will only be using it for plugging it in here but if I visit the same take twice in a single trajectory if I use online updates the second time I, I compute this GT 2 the value here will be the one with the changed value after the first time I computed GT2 from using offline updates I will use the same V here the first time and the second time a visit okay.

So that is the difference right and this might help us because I might be averaging out the value function may be a value function has converged to the true value right but the first time visitor I got a sample that was slightly lower than the true value the second time basically I got a sample slightly higher than the true value so combining with two together I will not change it right but if

I have done online updating the first time I would have bumped it down a little at the second time i will actually since i bumped it down a little of the bumper i will bump it up a large amount because the air will be larger now right so, so these things will happen so sometimes offline updating can work better sometimes online dating works better because offline updating you have to wait for the fact trajectories specially if the trajectories are very long so we'll be waiting for a long time to accumulate your changes so that works better sometimes and online updating also seems to work better if you are doing control .

Once per episode or it could even more batch of batch mode so if you remember I didn't drive this point through very forcefully then when we did that MC versus TD calculation we did batch mode updates there is not just offline there is not at the end of the episode it was the end of all the eight episodes was considered so you consider some six or eight episodes ring so the A BBBB and then A B and n so we did I think some seven or eight episodes reconsidered.

For that we did batch mode TD I mentioned that but I didn't tell you what batch mode is that furnace essentially you take all the trajectories compute all the changes once and then update it in one go okay so that is batch mode soft line is at the end of each episode you update all the values ones I mean basically all the states that occurred during the trajectory your plate Thurman see each state that would be only one update in offline and online there could be multiple updates for each state.

And in batch mode there is only one update for each state for every set of policies at your consider in fact batch mode is even better if you are loud if we can wait for so long if your trajectories are very short patch mode is better in terms of variance reduction. And so it is not as soon as like I said right many of these questions that I ask you the answer is not straightforward it is not always A or B it is almost always in their friends right.

And so remember especially as engineers or most of you are right or will be it is important for you to not know what the right answer is its for you to know what it depends on right so more often than not you will be jerry-rigging things in the field right so how many of you have seen that whatsapp video that went around on how to debug production code or the equivalent of debugging production code.

So that is Charlie Chaplin trying to sit on a train and trying to clear the track in front of him yeah so they say this is how you debug production code it so this is like that so it will be you will not be worried about so too much about theoretical finish and other things if when you go out and do things on the field right so you are more interested in getting things to work at that point you need to know what things depend on right not, not necessarily what is the right answer so yeah.

So just to show you the difference between online and offline that this is all through online updates I should at only three and five give you the least error okay and this is through offline updates right and the eight and six and eight step returns give you the least error right for online updates three and five steps like GT3 and GT5 give you the least error these are the two curves here and the two curves here are eight and six and yeah this is a little bit because the,

The walk remedy the problem that we are looking at is a 19 state problem alright so,

(Refer Slide Time: 18.16)

So you start from somewhere in the middle right and then you have how many states you need I have nine states this side right I have nine states that side right totally 19 states right and then I have a reward

(Refer Slide Time: 18.32)

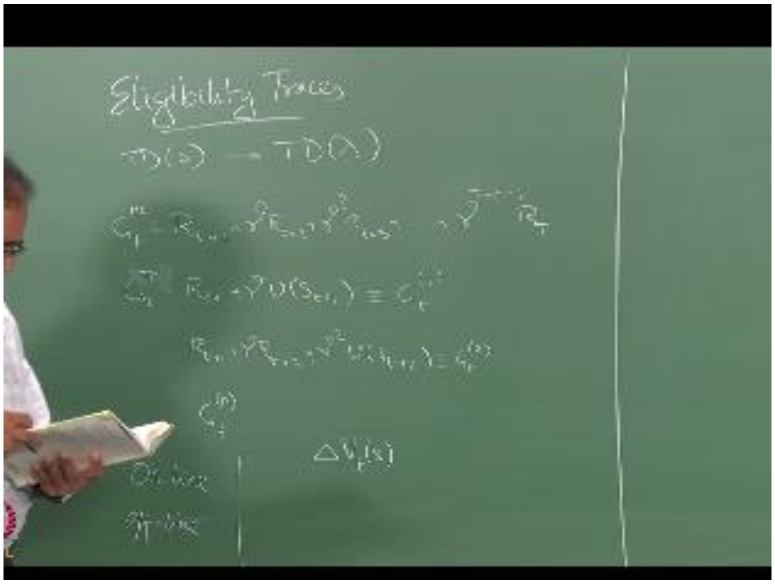
For reaching one end so I get a reward of +1 for reaching in and I get a reward of minus one for reaching their right and then just do a random walk so I am evaluating the uniform random policy all right so it really does not make sense to see why five should be the point where the lowest error is achieved or why should 6 meter the point or eight be the point eight I mean you think say nine or 19 or something right so but so it is five or six and five or six are actually much shorter lengths than the expected length of the random work.

You would need to obviously expected length has to be greater than nine right I am starting at the center so even if I rushed to one end it has to be nine right expected length of the trajectory has to be greater than and it will be much greater than nine because I am doing a uniform random walk I can go left or right with equal probability the link will be much higher but still it turns out that something much shorter in fact they run till 100-200 so the GT 100 GT 200.

And those look really bad so they basically are somewhere up there in the graph they do not even come anywhere if to a comparable point so it's a lot of complicated mechanisms in play here so a pre re trying to figure out which is the best n is not easy okay so where were we also online offline.

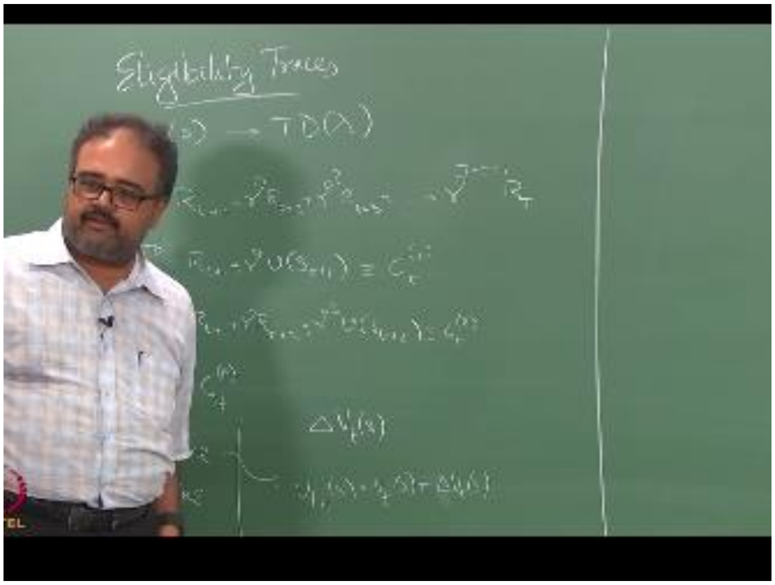
So just a little bit of notation here so , so far we have been talking about update rules as VT plus V in V of s T equal to V of s T plus alpha times an error term right so just to make it easier for me to do some of the results later just an additional notation that will introduce I am going to say

(Refer Slide Time: 20.35)


Eligibility Traces
TD(0) → TD(λ)
$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots + \gamma^{T-t-1} R_T$
$G_t^{(1)}$ $R_{t+1} + \gamma V(S_{t+1}) \equiv G_t^{(1)}$
$R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) = G_t^{(2)}$
$G_t^{(n)}$
$\Delta V_t(s)$
Online
Off-line

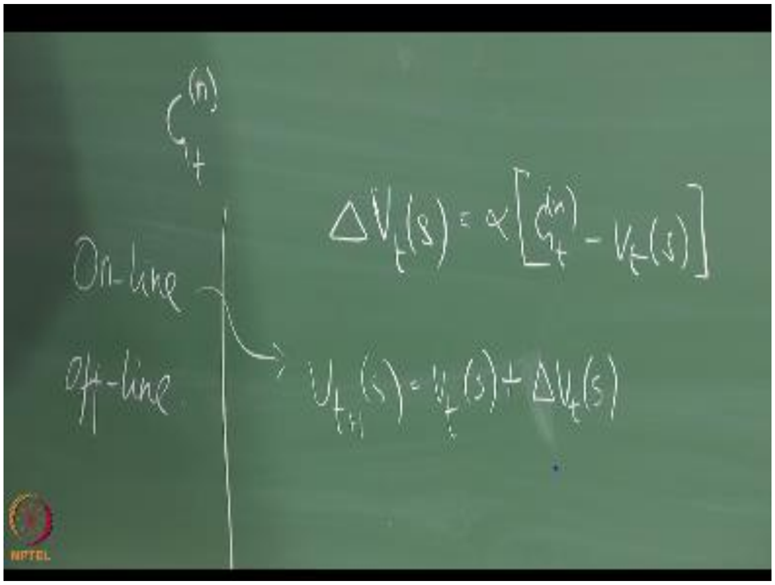
That every, every time we make an update I am going to change it by some quantity ΔV T of s right so the value function i will change it by

(Refer Slide Time: 21.01)



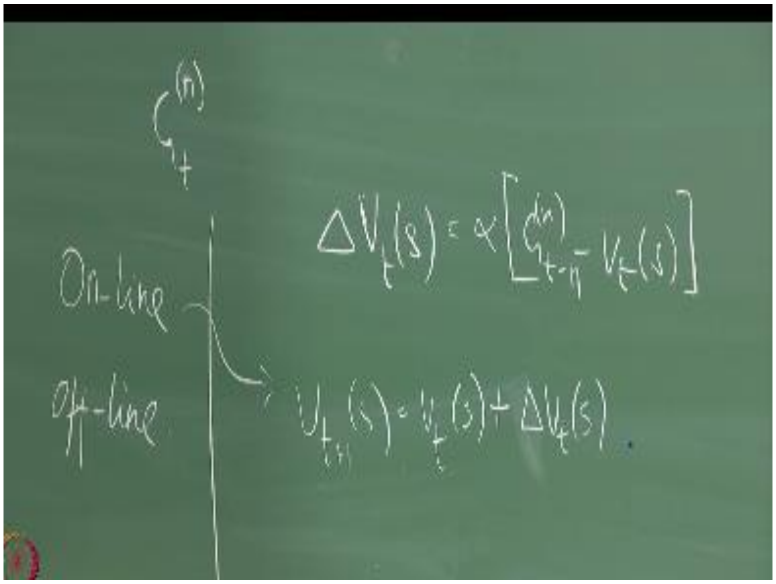
So this is essentially what i will do so it's VT + 1 of s equal to VT of S+ Δ V T of s right and what this Δ V T should be will compute It every time step right so the Δ V Twill be zero until n steps have gone since s occurred till n steps have gone I cannot compute GTN right after n steps this Δ V T will be right

(Refer Slide Time: 21.54)


$G_t^{(n)}$
On-line
Off-line
$\Delta V_t(s) = \alpha\left[G_t^{(n)} - V_t(s)\right]$
$V_{t+1}(s) = V_t(s) + \Delta V_t(s)$

So after n steps this will be GTN - VTS it till then it will be 0 so this update will happen VT + 1 yes will happen after n time steps for status right so till that point I will just be 0 already okay somewhat makes sense internships it till that time I will be making a zero update here that is all S occurs at some time right this is a little this is fine right so this S occurs at some time so this should be

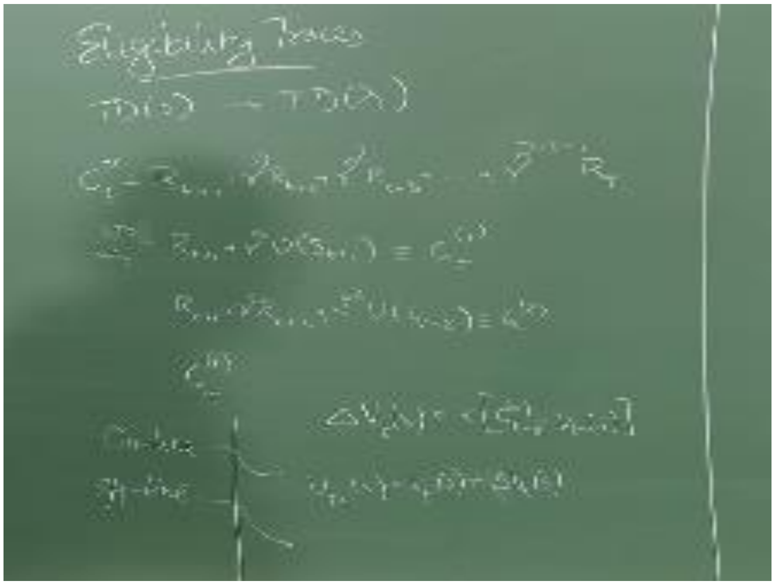
(Refer Slide Time: 22.54)


$G_t^{(n)}$
On-line
Off-line
$\Delta V_t(s) = \alpha\left[G_{t-n}^{(n)} - V_t(s)\right]$
$V_{t+1}(s) = V_t(s) + \Delta V_t(s)$

Okay yes occur t minus n steps ago okay then I start accumulating the return from then so i will get the N step return okay but then I do not want to say that I'll update the states only when it occurs and so I am going to update the states every time and I am going to say that if I have not completed the return so far I'll just keep it updating by zero value and whenever I completely written I’ll update it by that value just a trick okay because later on i am going to come up with a mechanism where even though i am you completing the end step return.

I will give you a mechanism for defining Δ V for every step along the way okay so this is something which will do later right now if you think of this what we are doing is we are waiting for n time steps after s occurs and then we make one update okay and what about the offline case

(Refer Slide Time: 23.51)



Eligibility Traces

TD(0) → TD(λ)

$G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots + \gamma^{T-t-1} R_T$

$G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1}) \equiv G_t^{(1)}$

$R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) \equiv G_t^{(2)}$

$G_t^{(n)}$

On-line

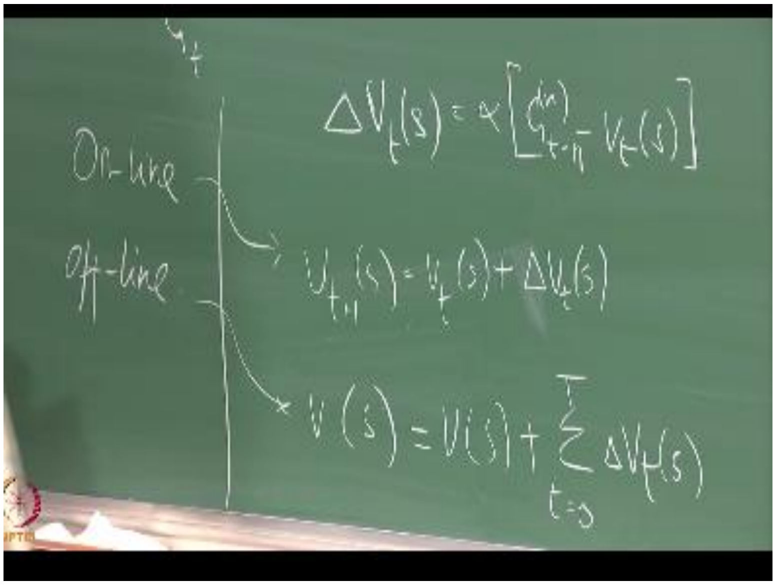
Off-line

$\Delta V_t(s) = \alpha\left[G_t^{(n)} - V_t(s)\right]$

$V_{t+1}(s) = V_t(s) + \Delta V_t(s)$

Offline case we do not make any changes until the episode comes to an end right so,

(Refer Slide Time: 24.07)



G_t

On-line

Off-line

$\Delta V_t(s) = \alpha\left[G_{t}^{(n)} - V_t(s)\right]$

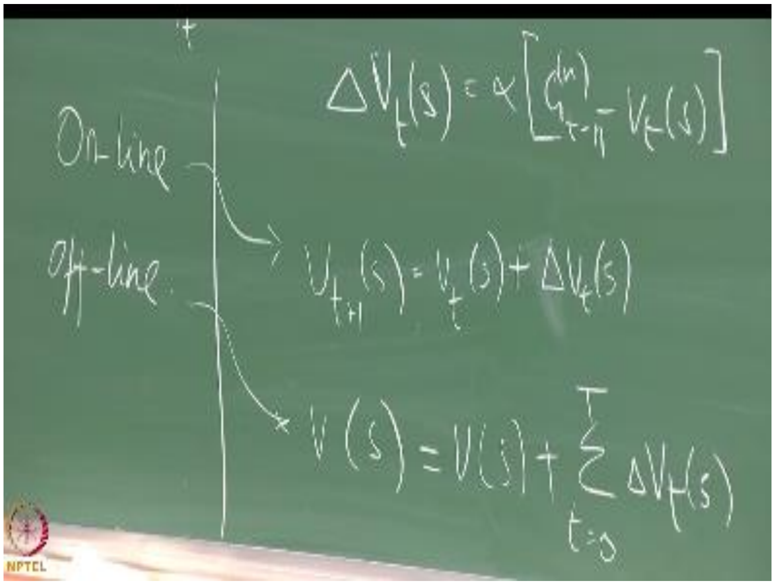
$V_{t+1}(s) = V_t(s) + \Delta V_t(s)$

$V(s) = V(s) + \sum_{t=0}^{T} \Delta V_t(s)$

So this is the new V okay it is equal to the old V plus all the Δ I should have applied to it through the entire episode starting from zero to T right so how will compute each of this Δ V T it will be completed by, every n steps after S occurs I will compute one Δ V T all the other times it will be 0. Tell me, how many of you say I will average, it how many of you say I'll submit, how many of you think the two answers are different.

What is the original update rule call I said these are a class of update rules called even before stochastic averaging, because that we are having this alpha thingy which is a small step right I can actually add it up okay that is essentially doing averaging what do you think I am doing here all right and just adding this right if you just think about it at the end of time I would added all the Δs any way that I will toss will be different since I am doing this alpha times this and moving it a small portion then it says in the averaging rubric what do you have to even separately averaging so that is basically it okay .

(Refer Slide Time: 26.13)



On-line

Off-line.

$$\Delta V_t(s) = \alpha\left[G_{t-n}^{(n)} - V_t(s)\right]$$

$$V_{t+1}(s) = V_t(s) + \Delta V_t(s)$$

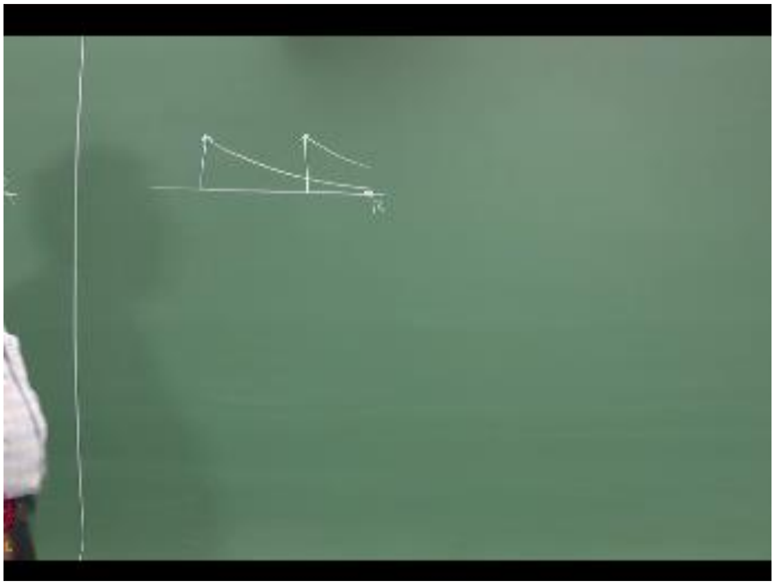
$$V(s) = V(s) + \sum_{t=0}^{T} \Delta V_t(s)$$

So this is what I do in the online case and that is what I do in the offline case this just to get you familiar with the notation when you start writing things later okay nothing big did no big deal about this nice okay great so now. The question way I asked you is how will you pick what is the right value of n right so I want to put a pause to the discussion here okay and give you a little bit of history.

And this whole next notion of eligibility traces predates pre dates RL to some extent it comes from behavioral psychology right so people in psychology started talking about what kind of a stimuli is causing a particular change in a behavior right and then you kind of you try to say that okay this stimuli is eligible for any changes that come out of this behavior because we believe that this is the one that costs the change in the behavior so for example the light comes on this before food is given to you right.

Maybe the light is more important for the foot as opposed to a belt that rank five minutes ago right so things that are more proximal to the outcome okay people started thinking of them as being more eligible for getting the payoff from the outcome frequency also matters will come to that in a minute right so this is so they started thinking about this kind of a trace mechanism right so what happens

(Refer Slide Time: 28.08)



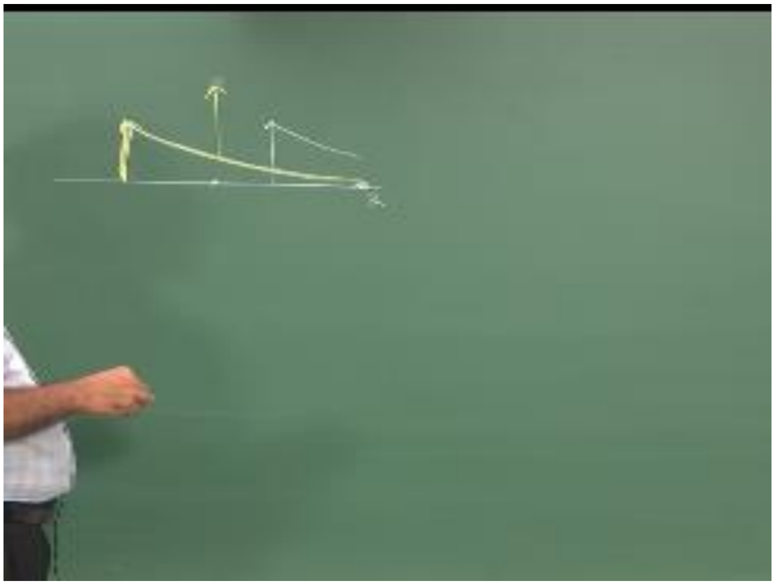
So let us say this is time right this is time see some reward occurs here right so some reward occur see some more occurs here right I have a stimulus that occurred here right I have another stimulus that occurred here right now I want to say that this reward belongs more to this stimulus than to the stimulus so how did they go about doing it one mechanism is to say that okay I will start a trace.

I will start something like this I will start a trace that goes keeps decaying with time right and at the point where the reward occurs whichever state whichever stimulus has the higher eligibility will get the higher update so for this particular trace this happened right so I'd then do this average over multiple traces then we'll be fine so the correlation would come when you do multiple traces.

Exactly so that is the normal bias that you have the things that are temporarily or spatially more closer to the outcome so the spatially more closely with the outcome will already taken care of by our discs own factor right so specially closer to the outcome you are giving it more reward so temporarily closer to the outcome you are giving it more share of the things so that is called the eligibility test.

Right so it is like so it's a little tricky so I am forcing it to you as a credit assignment problem right well it turns out that is something slightly different we'll come to that later right so now to come to your question about multiple occurrences so this stimulus occurred here say it occurred here again the same stimulus.

(Refer Slide Time: 30.08)



Let me come in different colors okay so the yellow stimulus occurs here again right so what do i do I take this instability swamp it up again now I start decaying it from there so it will start oops.

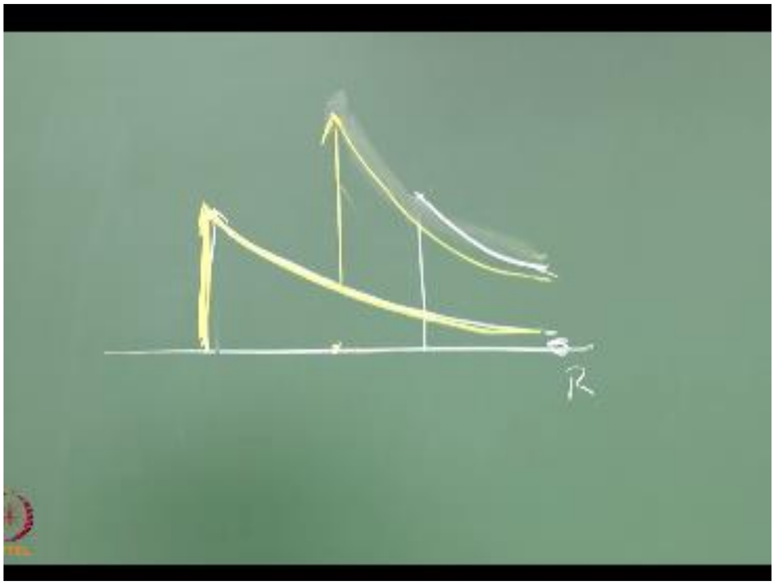
(Refer Slide Time: 30.35)

It should be decaying at the same rate so and so now they are more or less equally important because it occurred multiple times before the outcome at this occurred only once but it is closer temporarily closer to the outcome right but suppose this occurs only time right then it will go somewhere up there and obviously so at this point I am going to say hey no, no the other one is more important than the white one because that occurred so many times before I got the reward.

Right so this is how it is one law of thinking about it said they don't necessarily be exponentially decay that I have shown here but there is one way of thinking about how this eligibility things will operate right so people started thinking about this kind of an eligibility mechanism for modeling this kind of temporal proximity right so.

This has been around for a long time in the psychology literature people talked about these kinds of eligibility mechanisms for a long time right so what happened was so when, when a certain Bartow started working on TD learning right so they came up they introduced to this kind of eligibility mechanism they say they said that whenever a state occurred right I kind of compute the TD error for that state and then I will give it back to states that occurred before me also right. I will just not whenever the reward occurs or whenever something in case i will just not update the immediate previous state i will also update many states that occurred before that also. So why is that essentially an eligibility Mechanism.

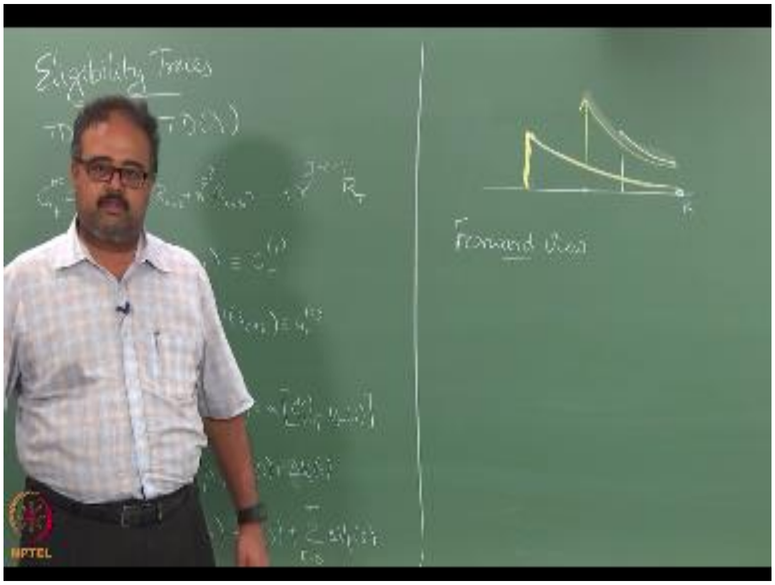
(Refer Slide Time: 32.25)



Right the guy who occurred just before the reward happen edit is probably most responsible for their ward but this guy's work hard even before that right but also have some amount of responsibility for the river so let me give something to them right so just not the immediate update let us go back a little bit and also do those updates so they introduce this kind of an eligibility mechanism inspired largely by what the, the psychology literature was telling them.

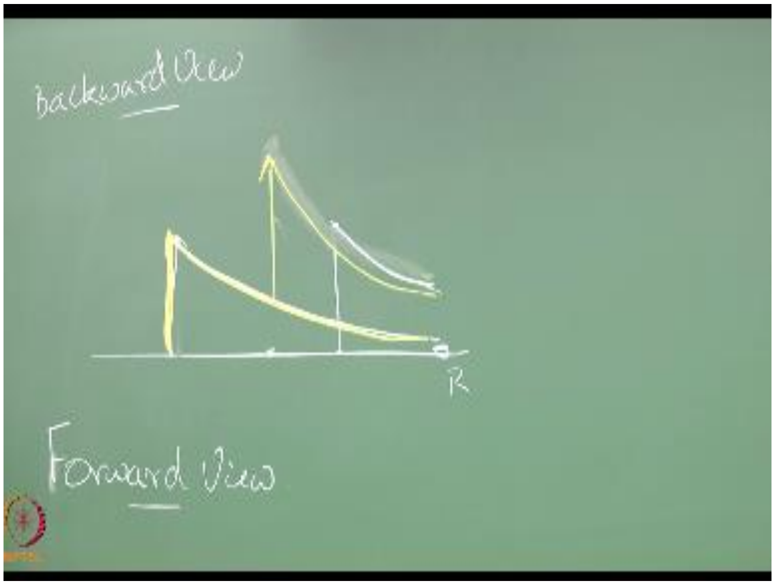
Right and then later on you know several years down the line they came up with the mechanism to explain what this eligibility traces was actually doing in terms of what was the notion of return that they were estimating what is the value function that they are estimating in that case and so on so forth a salary says somehow the value function does it get changed right and what is the whole process of updating that is happening when you are using eligibility traces okay so they came up they try to come up with an explanation for this and then they came up with what they call the

(Refer Slide Time: 33.25)



Forward view of TD lambda ok and this thing they call the no prizes for guessing.

(Refer Slide Time: 33.50)



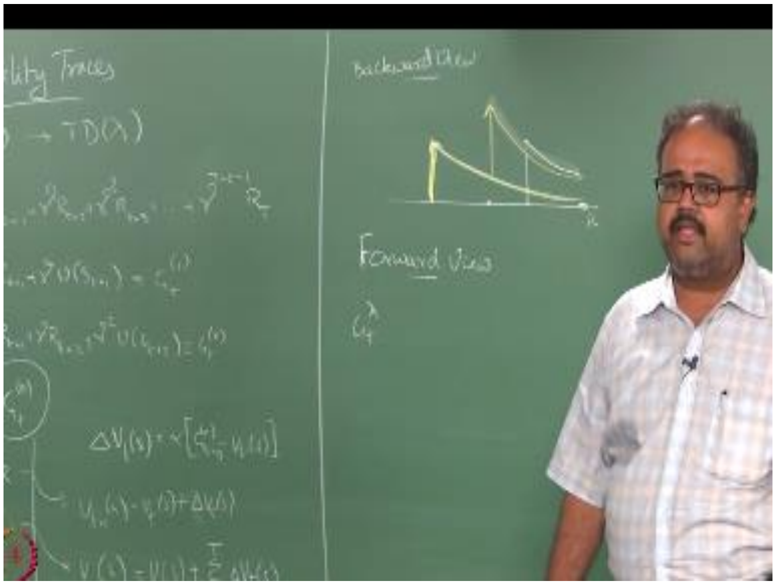
The backward view of KD lambda because you look at window ray what happens okay and then you go back and see which are the states that should get the reward so it is called the backward view of TD lambda okay the forward view of TD line that says that ok now I mean stay guess what is the motion of return that I should use for updating the value of status so I'm going to look forward and I see what is going to be the future expected return from here right.

I am going to use that return to update the value at status so this is called the forward view the backward use ok something has happened okay i have now completed a TD error whatever the states that i have already visited should i update this TD error right so that's called the backward view so i will come back and write down the backward be more formally but now let's go back and look at what the forward view states.

Right so the reason I gave you this explanation is if you look at what we do at taking the end step return and doing TD lambda looks like an arbitrary choice so the way they convert the instep returns into a TD lambda return say we did the TD return we did a Monte Carlo written so we are going to do a lambda return now okay.

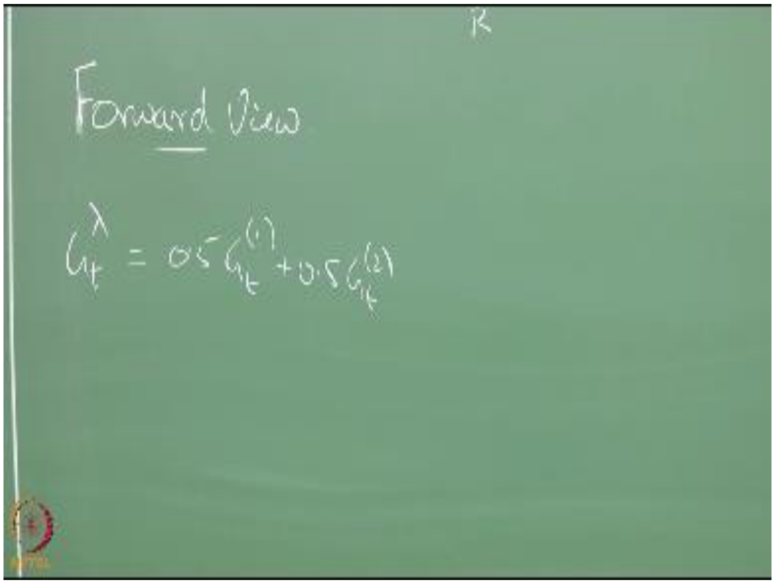
So the way they define lambda written looks very arbitrary but the reason they did the lambda return is so that the forward view matches the backward view so the back ward view came first but in the book when they present things they present the forward view because it has got more nice mathematical explanation as to what is happening okay. So what is the forward view do the forward view essentially allows you to define

(Refer Slide Time: 35.32)



A new notion of return which we call the lambda return so what is the lambda return so lambda return tries to do.

(Refer slide Time: 35.54)



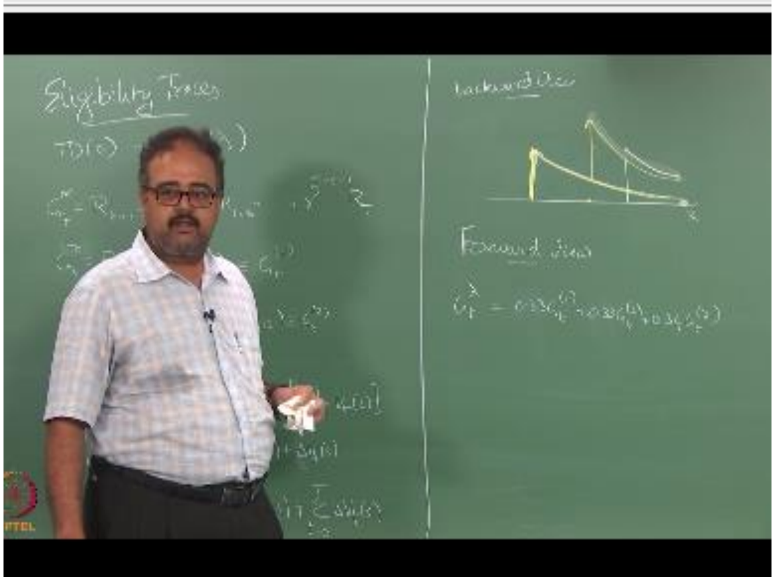
Forward View

$$G_t^{\lambda} = 0.5\, G_t^{(1)} + 0.5\, G_t^{(2)}$$

Find some kind of an average some kind of a weighted average of the different n step returns right so you could think of something like say I am going to take point five so i am going to use this instead of using either the one-step return or the two-step return i am going to use some kind of an average of the two okay so that gives you some advantage of good shopping from here and the some advantage of having actual real samples right so.

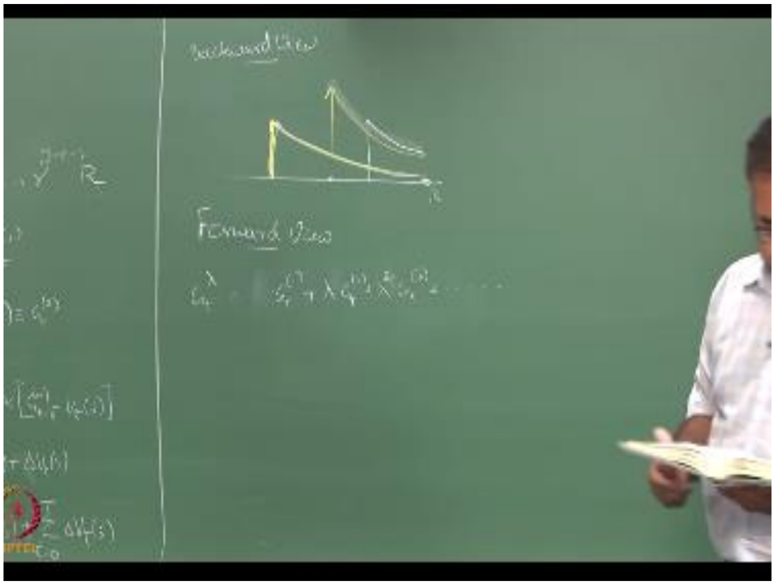
I am using some kind of a mix of both right but why should i stop with something like this right can you do this right I can,

(Refer Slide Time: 36.42)



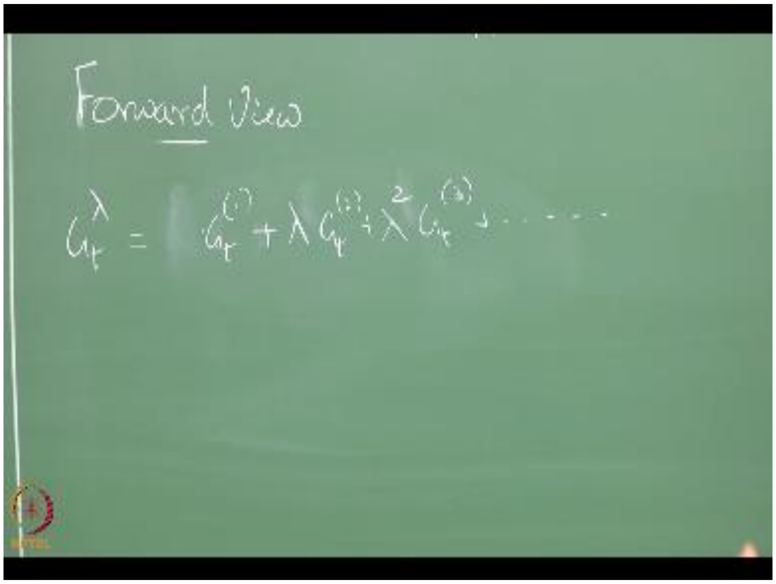
Can think of generalizing this forever right so it would be a good way of doing this summation some, some, some factor right so what will i do so I am going to do something like this.

(Refer Slide Time: 37.15)



I am going to take lambda times yeah so remember okay I’m going to pick some arbitrary summation like this okay but that is a problem with this also problem coefficient have to sum to 1 so what should i do further because i am trying to do a weighted average of these multiple returns rate if it does not sum to 1

(Refer Slide Time: 38.15)

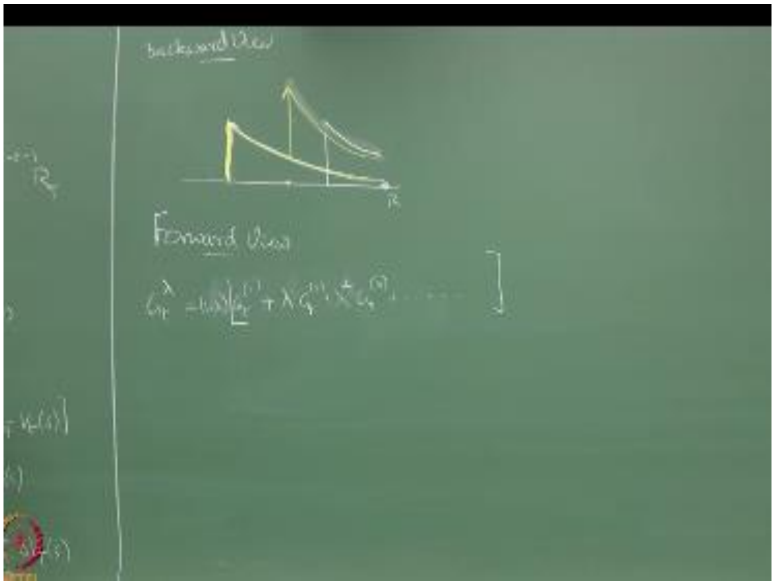


Forward View

$$G_t^{\lambda} = G_t^{(1)} + \lambda G_t^{(2)} + \lambda^2 G_t^{(3)} + \cdots$$

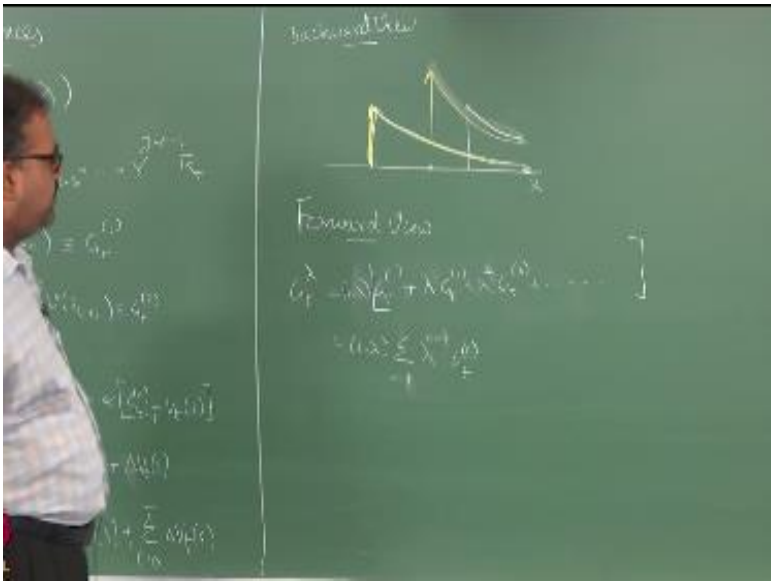
Then i am actually making an overestimate of the return i am taking multiple samples of The so if you think about it each one of this is a different measurement of the return right this is one measurement of the return this is another measurement of the return and so coefficients known sum to1 I am going to make it an over estimate right so I wanted to some too and so what should I do to make it sum to 1 x 1 minus lambda throughout it will sum to 1 so essentially it is right.

(Refer Slide Time: 38.45)



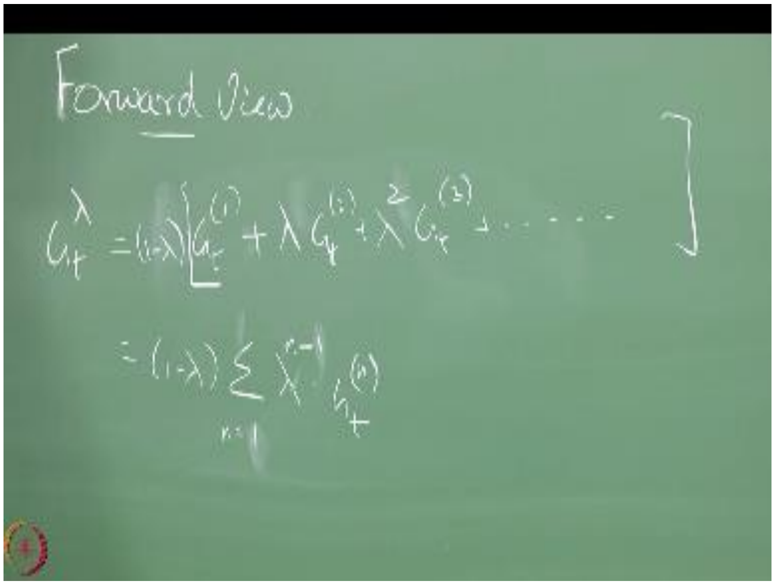
So this is essentially what the lambda return is.

(Refer Slide Time: 39.03)



So tightly related GTE of what rotted week a fine okay so someone's from n equal 1to infinity I am all set why I am sorry what episode has to terminate other ways I mean I do not have this notion of first even we talked about we do discounted returns episodes do not have to terminate right the pieces can go on forever so what happened in this case my lambdas will become smaller and smaller.

(Refer Slide Time: 40.30)



Forward View

$$G_t^{\lambda} = (1-\lambda)\left[G_t^{(1)} + \lambda G_t^{(2)} + \lambda^2 G_t^{(3)} + \ldots\ \right]$$

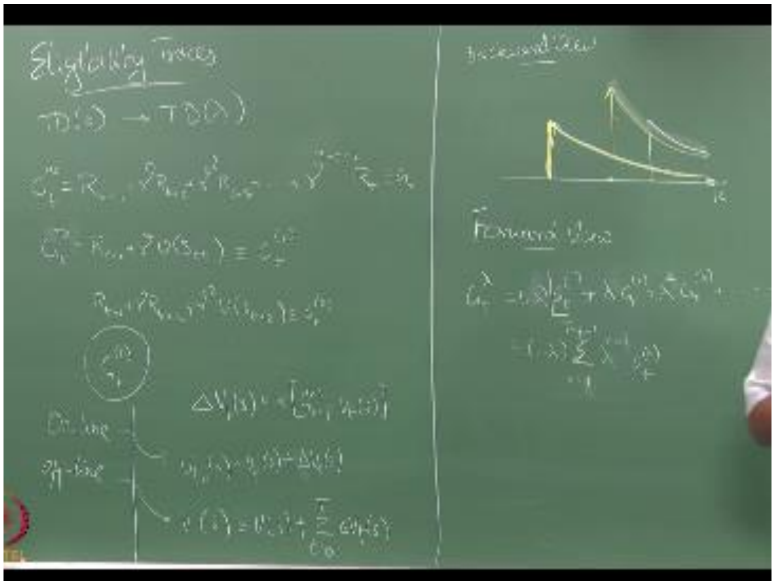
$$= (1-\lambda)\sum_{n=1} \lambda^{n-1} G_t^{(n)}$$

It could go on forever right so there is no problem with this can go on forever but typically what happens is the episodes tend to terminate right so what will I do here in the episode terminates what should be the summation in upper end right we need one more right but there is a silver problem what is the problem.

So I have all of these things let met episode terminates at some point so what about the rest of it nothing if I say nothing then I run into problem when lambda stone sum up to 1again so I need something there right so if you just think about it one sake so if i say my n is greater than t minus t minus t minus 1 so if the n is so large and it is actually goes beyond the episode end what is expected return I am going to get whatever is there for a full episode right.

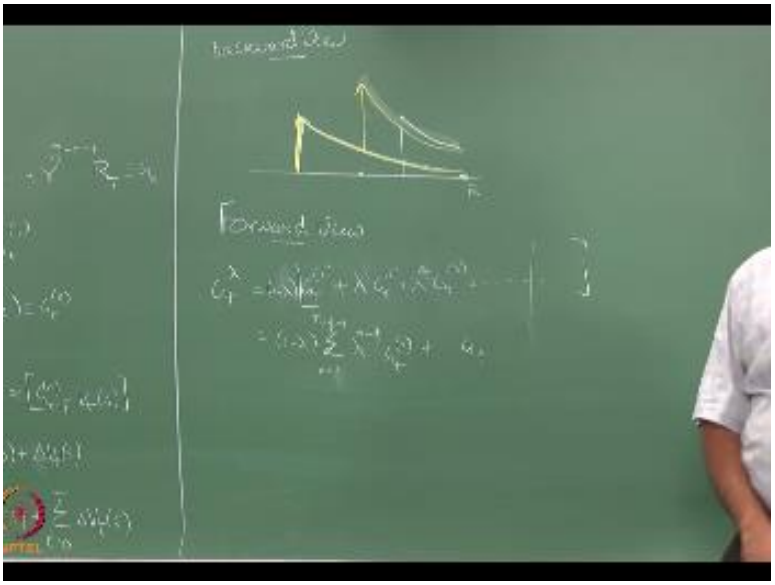
I mean it matter how much however long I wait after the episode ends the return is not going to change right so after a certain point when n becomes greater than capital T minus T right the return will be the same as the Monte Carlo return right so the Monte Carlo return i am going to denote simply to say,

(Refer Slide Time: 42.05)



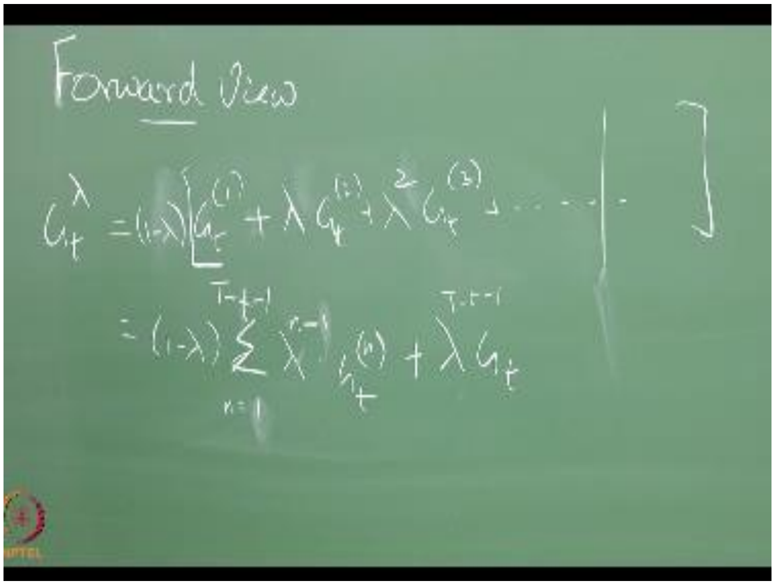
Say SGT right so GT without any super cute is the original return so we all know that so it is going to be the same so what I have to do now

(Refer Slide Time: 42.21)



I have to treat that as a special case I will have a GT there and what should be the coefficient of GT well I have summed up by lambda from 1 to t minus t minus t minus T from that all I were to sum up the rest of the lambda right so i can just take that lambda power out and the rest of it will sum to1 minus lambda so my oh 1 by 1 minus lambda so that part will cancel out I will just have lambda power.

(Refer Slide Time: 42.50)



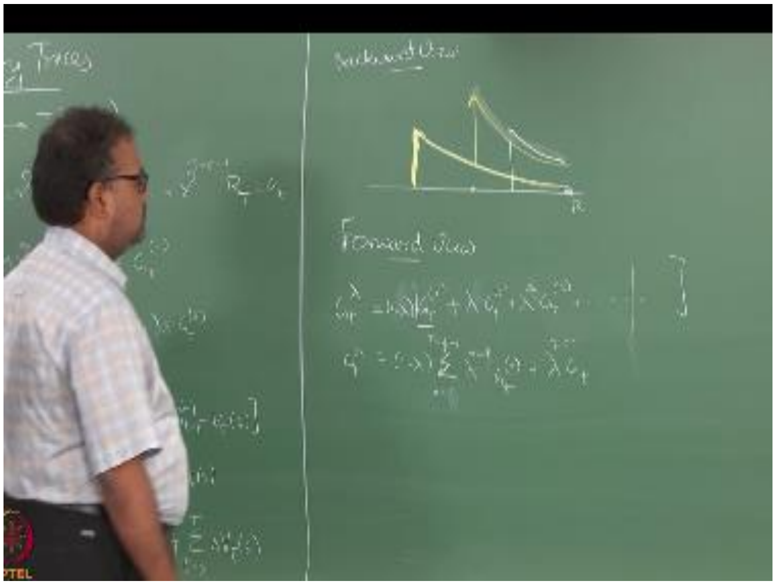
Forward View

$$G_t^{\lambda} = (1-\lambda)\left[G_t^{(1)} + \lambda G_t^{(2)} + \lambda^{2} G_t^{(3)} + \cdots \middle| \cdot \right]$$

$$= (1-\lambda)\sum_{n=1}^{T-t-1} \lambda^{n-1} G_t^{(n)} + \lambda^{T-t-1} G_t$$

Right so that clearance oh how that lambda came with lambda term came there is essentially summing out the rest of the coefficients I get the lambda term so this is essentially by GT lambda.

(Refer Slide Time: 43.11)



Sending any questions on this so is there an online versus offline version of GT lambda you have to wait for the whole episode before you can compute D T lambda right there is no question of an online is all offline there is no question of an online or offline update here it is all offline go ahead and wait till the end of the episode because I need GT I just like the Monte Carlo case we talked about right so you have to wait till the end of the episode because you are using GTS here as soon as you put GT in the target.

You have to wait till end of the episode so the cool thing about the backward view is that it allows you to implement ed lambda in a online fashion right but this, this view now gives you an offline okay so this is essentially now we have converted this whole thing into an offline problem we cannot do this online well let mere phrase it online offline do not make any sense here right we said online here was when you, you wait till your computer return.

And then you make the update right so here also you do the same thing you wait till computer return but turns out that the return is computed only at the end of the trajectory so it is both online offline everything is the same okay so people are clear with so the forward view so here is here again some pictures which unfortunately since they do not use slides is going to have to make do with this so this is for different values of lambda the same 19state random work at four different values of lambda and for different values of alpha.

And you can see that the minimum is achieved when lambda is 0point nine so the minimum here is achieved when lambda is point nine that if you remember in the offline case the minimum was achieved for n of eight or six or eight right n was six or eight and we say that is a minimum but here it turns out lambda point name is the many more similar kind of curves just like the previous case right so of course now here is more incentive for you guys to actually go and look at the book because you can see the curves up close and personal in the book the new book called the minute.

And what is online why what is what is the notion of online here for lambda returns okay so they did the T DTD lambda version of it okay let's find that okay so yeah so I will show you the online version of TD lambda in a bit right so offline at this 0.9 yeah this is offline curve referring yeah so the online version we are not had seen so we seen only the offline version right so far you can't appreciate what was the difference between online and offline and I told you the backward view is what lets you do the online version so we will come to that nervous

IIT Madras Production

Funded by

Department of Higher Education

Ministry of Human Resource Development

Government of India

www.nptel.ac.in