NPTEL

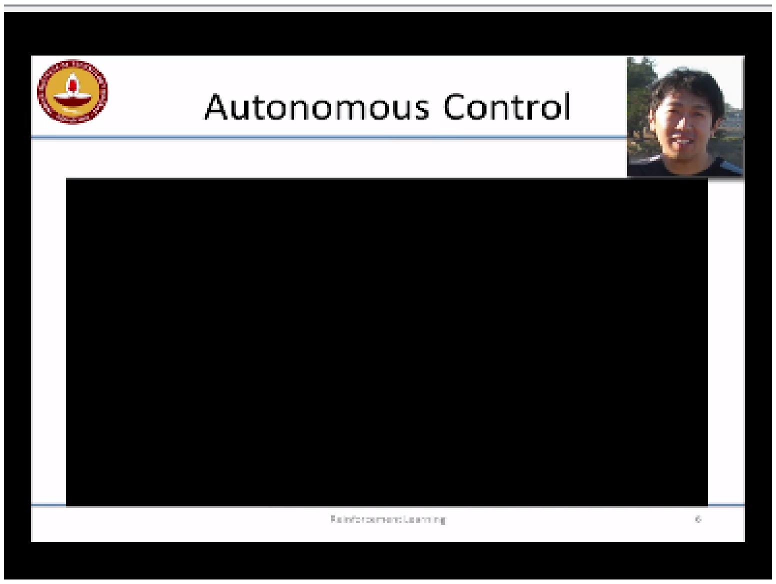
NPTEL ONLINE CERTIFICATION COURSE

REINFORCEMENT LEARNING

RL Framework and Applications

Prof. Balaraman Ravindran
Department of Computer Science and Engineering
Indian Institute of Technology Madras.

(Refer Slide Time: 00:16)

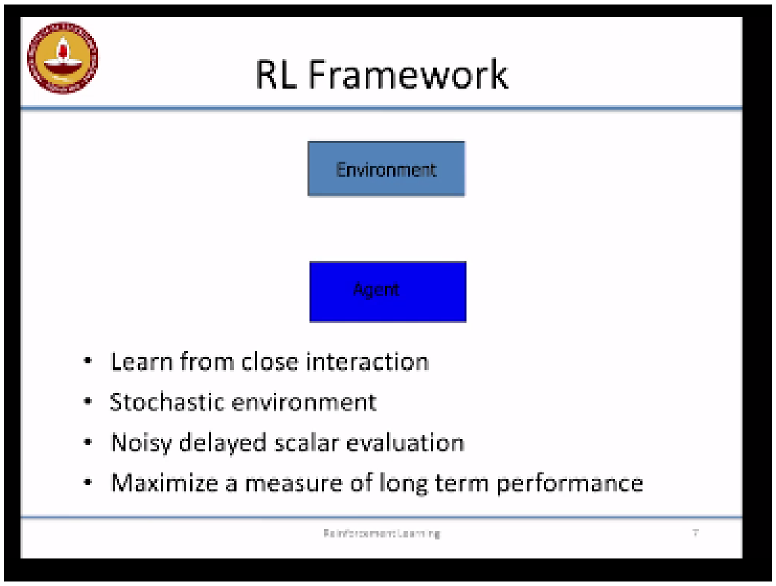


Autonomous Control

Reinforcement Learning 6

You can do all kinds of fancy stuff with reinforcement learning, right.

(Refer Slide Time: 00:18)


RL Framework

Environment
Agent

- Learn from close interaction
- Stochastic environment
- Noisy delayed scalar evaluation
- Maximize a measure of long term performance

So the Kirk's in reinforcement learning is that the agent is going to notice is learning agent right it is going to learn in close interaction with an environment, right. So the environment could be the helicopter it could be the cycle right and or it could be your backgammon board and your opponent all of this could constitute environment variety of different choices, so you sense the state in which the environment is in right.

Since the state of the environment and figure out what is the action that you should take in response to the state, right. So in apply the action back to the environment this causes a change in the state right. So now comes in a tricky part right so you should not just choose actions that are beneficial in the current state but it should choose actions in such a way that they will put you in a state which is beneficial for you in the future, right.

Just capturing the queen of your opponent is not enough in the chase that might give you a higher reward but it might put you in a really bad position, so you do not want that I do you really want to be looking at the entire sequence of decisions that you are going to have to make and then try to behave optimally with respect to that, right. So what we mean by behave optimally in this case right where we had assume that the environment right is giving you some kind of an evaluation.

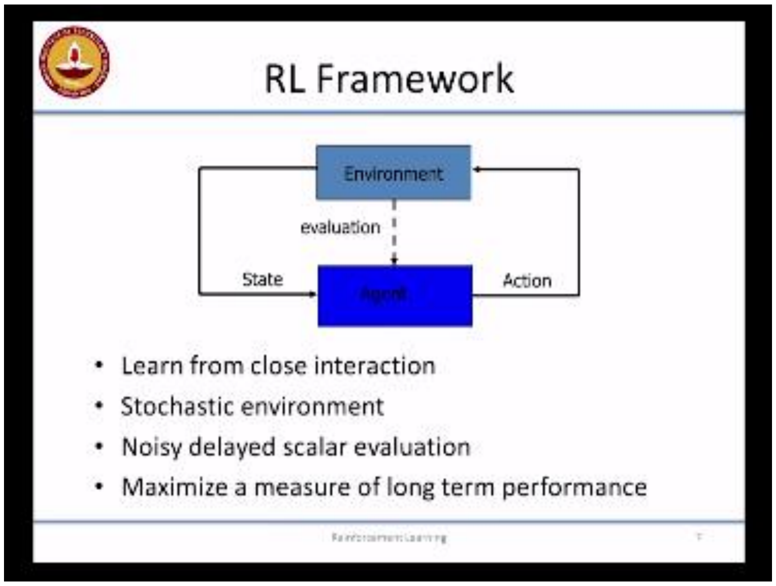
It is like falling down hurts when or capturing a piece maybe gives you a small +0.5 or winning the game gives you like hundred right, so every time you win every time you make a move or every time you execute an action you did not get a reward or you did not get an evaluation from the environment right so it could be just zero it could be this could be nothing, right. So I should point out that this whole idea of having an evaluation come from the environment is just a mathematical convenience that we have here.

But in reality if you think about biological systems that are learning using reinforcement learning right all they are getting is the usual sensory inputs right so there is some fracture in the brain okay that sits there and interprets some of those sensory input as rewards or punishments, right so you fall down you get hurt I mean that is still a sensory input that is coming from your skin right or somebody pats you on aback that still a sensory input that comes from the skin.

And it is just another kind of an input right so it could choose to interpret this as a reward right or this as a collision with an obstacle something is brushing against my shoulder let me move in right or you can just take it as somebody is patting my back so I did something good right so it is a matter of interpretation so this is a this whole thing about having a state signal and having a separate evaluation coming from the environment is a friction right.

There is created to have a clearer and clearer mathematical model but in reality things are a lot messier you know you do not have such a clean separation, right and like I said so you have a stochastic environment you have delayed evaluation.

(Refer Slide Time: 03:22)



RL Framework

- Learn from close interaction
- Stochastic environment
- Noisy delayed scalar evaluation
- Maximize a measure of long term performance

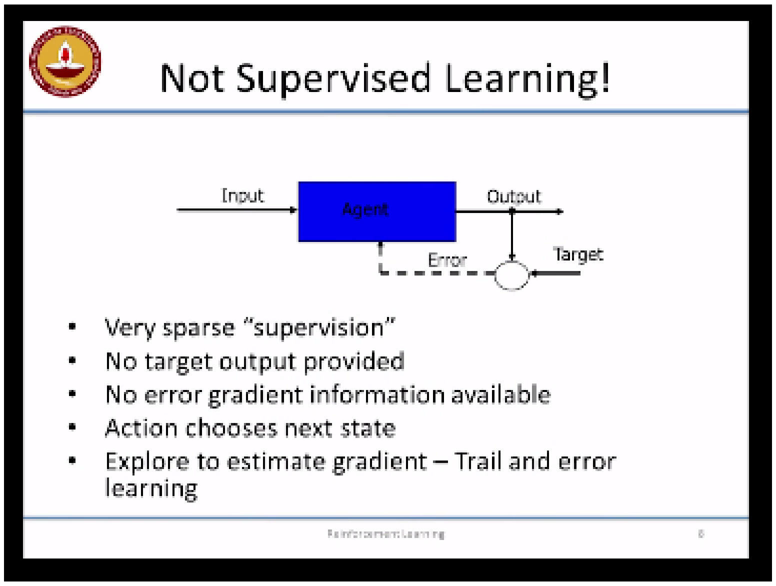
Noisy so the new term that we have added here is scalar the new term we have added is scalar, so that is one of the things with the classical reinforcement learning approaches he said I am going to assume that my reward is a scalar signal right so have we talked about getting hurt and having food and so on so forth what will all of this will happen mathematically is I will convert that into some kind of a number on a scale right.

So getting hurt might be -100 right getting food might be +5 winning the game might be +20 right capturing a piece might be +0.5 or something like that, so I am going to convert them to a scale right and the goal is now that I have a single number that represents the evaluation the goal is now to get as much as possible of that quantity over the long run, okay make sense right. So if you have questions doubts stop me and ask.

So mathematically a scalar which is easy at optimized, not necessarily right I am just talking about which is like a cost function if you want to think about it in terms of in terms of control systems right so this is like a cost and I am trying to optimize the cost all right and so for the cost is going to be vector value then I will have to start trading off one direction of director against the other.

So which component of the vector is more important so then you get into all kinds of separate optimality and of questions so it is not really clear what exactly is optimal in such cases right, so in fact later on I we have a homework exercise for you to argue for and against scalar rewards so you can put on your thoughts then right, so here again let me emphasize its not supervised learning right in supervised learning, right.

(Refer Slide Time: 05:18)



Not Supervised Learning!

Input
Agent
Output
Error
Target

- Very sparse "supervision"
- No target output provided
- No error gradient information available
- Action chooses next state
- Explore to estimate gradient – Trail and error learning

Reinforcement Learning

In supervise learning this is essentially what you are going to see there will be an input and that will be an output that you are producing and somebody will be giving you a target output okay so this is what you are supposed to produce and essentially compare the output you are producing to the target output right and we can form some error signal right and you can use that error in order to train your agent right.

You can try to minimize error you can do gradient descent on their work into variety of things you can try to train the agent, so here I do not have a target I do have to learn a mapping from the input to the output but I do not have a target and hence I cannot form an error right and therefore my trial and error becomes very essential see if I have errors rate I can form gradients of the errors and they can go in the opposite direction of the gradient of the error, right.

And then that gives me some direction in which to change my parameters and that constitute the agent, right neither it is going to be described in some way right the error gives me a direction right but now since I do not know a direction right so I just I do something I get one evaluations I don't know that the evaluation is good or bad right so think of writing an exam right I do not tell you the right answer I just tell you three right.

And so what do you do now do happy with answer should you change it should he change it in one direction or should he change it to the other direction, so what makes it even more tricky is I you do not even know how much the exam is out of, so when I say 3 it could be three out of three it could be three out of 100 right and or it could be like 3 out of 18 which is what the class average in level in the first quiz right.

So it could be any of these things right so you do not even know whether there is a good number or a bad number so you have to explore to figure out a if you can get higher than three right or three is the best the second thing is if I can get higher than three how should I change my parameters to get to become higher than three it is I have to change my parameters a little bit that way okay.

I have to change the parameter is a little bit this way right so if I am cycling wait I have to push down a little harder on the pedal okay all I have to push down a little softer on the pedal to figure out whether I am staying balanced for a longer period of time or not I do not know that otherwise unless I try these things I would not know this is why the trial and error part, so if you pushed on a little harder and I say balance maybe I should try pushing down even more harder next time right.

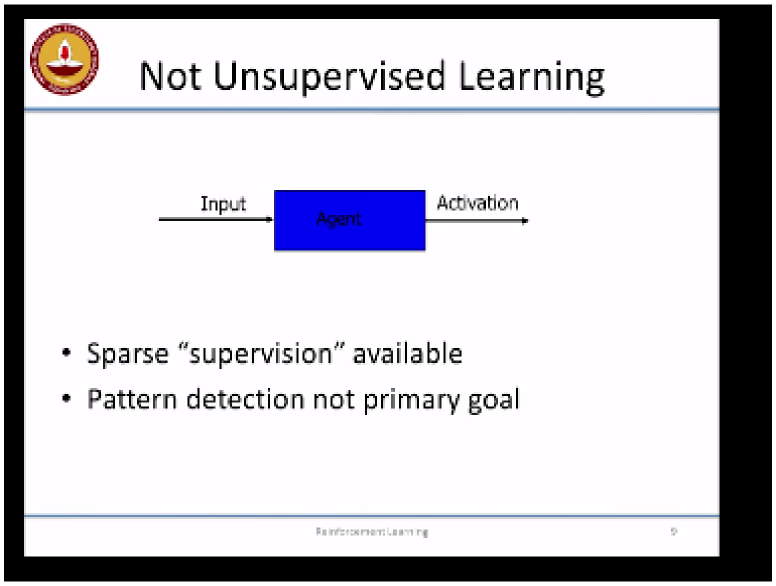
So maybe that will make it better and then there might be some point where into port so I need to come back so this is of things which you have to try unless you try that you do not even know which direction you have to move in right so this is much more than just the psychological aspects of trial and error there is also a mathematical reason if you want to adapt my parameters wait I need to know the gradient okay so that he will need to look, yeah.

The reward is the one that you know that gives you the evaluation for the output right so here in the supervised case the error is the evaluation for the output of the error is 0 then your output is perfect right but then have a way of gauging what the error is because you have a target which you can compare right and from there you get the error so in the reinforcement learning case the evaluation is directly given to you as the evaluation of the output right.

It is not necessarily comparing against target value or anything you do not know how the evaluation was generated now you just get an evaluation directly so you just get some number corresponding to the output right so maybe I should have done put an arrow from the top saying evaluation comes in from there but that is exactly where it is coming it is a substitute for the error signal but it just that you do not know what the evaluation is.

Of course way differs from the error is minor differences you typically tend to minimize error but you are tend to maximize evaluation right it is also not unsupervised learning so I'm super learning you have some kind of an input right.

(Refer Slide Time: 09:29)



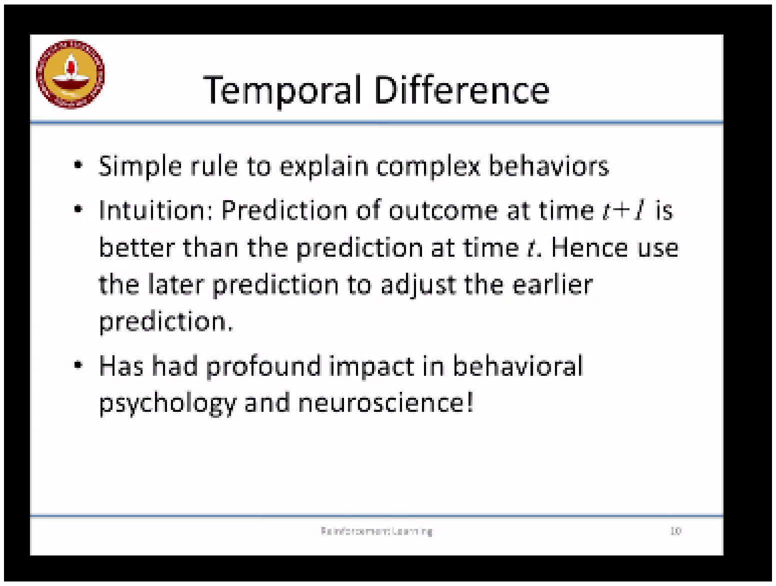
Not Unsupervised Learning

Input
Agent
Activation

- Sparse "supervision" available
- Pattern detection not primary goal

That goes to the agent and then it figures out what are the patterns for the in the input right here you have some kind of an evaluation and you are expected to produce an action in response to the input it is not simply pattern detection right so you might want to detect patterns in the input so that you know what is the right response to give but that is not the primary goal right but in unsupervised learning the pattern reduction itself is the primary curve so that is the difference right.

(Refer Slide Time: 10:03)



Temporal Difference

- Simple rule to explain complex behaviors
- Intuition: Prediction of outcome at time $t+1$ is better than the prediction at time t. Hence use the later prediction to adjust the earlier prediction.
- Has had profound impact in behavioral psychology and neuroscience!

Reinforcement Learning 10

So here is this one slide which I think is kind of the soul of reinforcement learning right it is called temporal difference so I will explain it little more detail and in a couple of slides but the intuition here right so if you remember the Pavlov's dog experiment right what was the dog doing it was predicting the outcome of the bell you know if the bell rings there is an outcome that is going to happen is predicting the outcome which is food is going to happen.

And then it was reacting appropriate to the outcome right so most of reinforcement learning you are going to be predicting some kind of outcome this going to happen since I am I going to get a reward if I do this or if I am I going to not get a reward am I going to win this game if I make this move what am I not going to win this game all right so I am print always trying to predict the outcome the outcome here is the amount of reward or punishment I am going to get right. This is essentially what I am trying to predict at every point right so the intuition behind what is called temporal difference learning is the following right so the prediction that I make at time T+1 okay of what will be the eventual outcome let us say I am playing a game right I am going to say I am going to win now I am very sure I am going to win now right, so I can say that with a greater confidence when I am closer to the end of the game.

Then I can at the beginning of the game right so I have all the pieces set up and if I am going to sit there and say I am going to win the game right it is most probably wishful thinking right but then you have played the game for like 30 minutes or something and there are like five pieces left on the board now I am going to say I am going to win the game now I say I am going to win the game that is a much more confident prediction then what I did at the beginning right.

So taking this to the extreme right so the prediction I make at t+1 is probably more accurate than the prediction I make it t, right the prediction I make a t+1 is more accurate than the prediction I make it t, so if I want to improve the prediction I make a t what can I do I can look go forward in time basically go to the next day let the clock tick over and see what is the prediction I will make a time T+1 with additional knowledge I am getting right.

I would have moved one step closer to the end of the game so I know is little bit better about the game right I know how the game is proceeding I know I can may now make a prediction about whether I will win or lose right and use this go back and modify the prediction I make it time t, right T I think there is a possibility of say probability of 0 point 6 of me winning the game okay and then they make a move then I find out that I am going to lose the game with a very high probability.

Then what will I do is I will go back and reduce the probability of winning that I made at time T so instead of 0 point six I will say okay maybe point five or something alright, so next time I come to the same state as I was at time T, I would not make the prediction of 0.6 I will say 0.55

okay that essentially the idea behind capital difference learning, right. So it is a whole lot of advantages we will talk about it a couple of slides down.

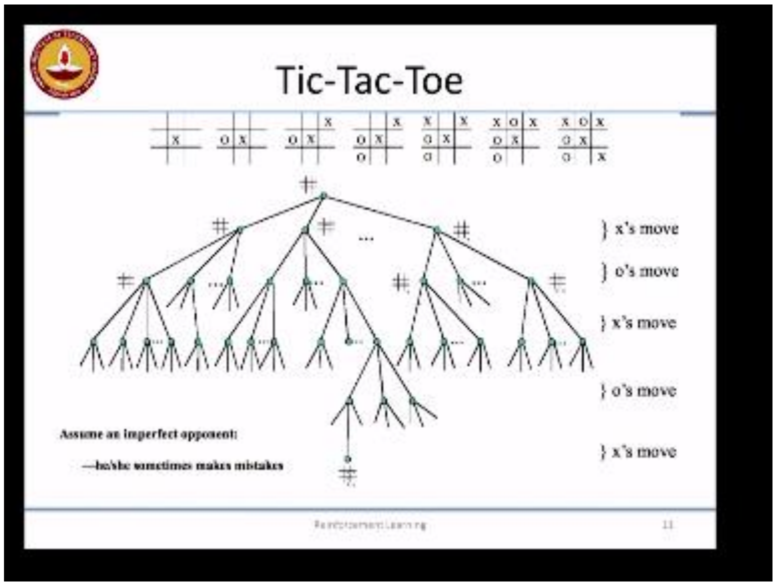
But one thing is he said the significant impact in behavioral psychology and in neuroscience right so it is widely accepted that animals actually use some form of temporal difference learning and in fact there are specific models right that that have been proposed for temporal difference learning which seemed to explain some of the neurotransmitter behaviors in the brain, yeah. No yeah see at this point I will be making a prediction about what is the probability of winning.

And it could be for each of the moves right if I make this move what is the probability of finding if I make this move what is the probability of finding, let us say I make move two okay and then I go I see a new position let me open it responds to it and then I decide oh my god this is a much worse move than I thought earlier so what I do is I will change the prediction I make for move two in the previous state.

You see that the other moves would not be affected because the only move it took was two only about that move I have additional information therefore I can go back and change the prediction I make for move to alone, so you still have the ten moves so they are not changing any of them. Seeing the prediction is not like I mean if an ideal world you should be able to take back a bad move right except if it is a parent playing with the kid I do not think those things are allowed right.

Infact I when I play with my son we have sometimes had to remain back all the way to the beginning it will probably be me asking to do the remaining not him because he will be drubbing me in some one of those games but yeah otherwise you cannot you just make the change the prediction so next time you play the game it will be better at it, it not for that game well basically you are messed up. Or you did well I mean so whatever it is yeah looking at RL right so listen look at tic-tac-toe.

(Refer Slide Time: 15:33)


Tic-Tac-Toe

Assume an imperfect opponent:

—he/she sometimes makes mistakes

Reinforcement Learning 11

How many of you have played tic-tac-toe, good oh you and you put it right up okay good okay good so in tic-tac-toe so you have these both positions right and so you make different moves so in the first this is what I have drawn here is called the game tree right so I start off with the initial board which is empty right and there are how many possible branches there for people making moves nine possible branches right.

For excess move there are nine possible moves I have nine possible branches and then for each of these I will have like eight possible I am not sure this is the right TV for each of this I have eight possible branches and they keep going right so what we are going to be doing is essentially trying to this formulate this as a reinforcement learning problem, so how will you do this as a reinforcement problem right.

So I have all these board positions right let us say X is the reinforcement learning agent and O is the opponent right so initially given a blank board I will have to choose one among nine actions right so there the state that I am going to see is this black the accessory goes on the board right and the moves I will be making are the actions right, so in the initial position I have nine actions I make that do I get any reward not really there is no natural reward signal that you can give.

Essentially the reward that I am going to get in this case is if at the end of the moves if I win I will get a one if I do not win I get a zero and if I when I get a one if I do not print I get a zero right so what is going to happen is I am going to keep playing these games multiple times right and at each point right yeah okay so there is a note here so what is it not say we ought to assume it is an imperfect opponent right.

Away there is no point in trying to learn tic-tac-toe why, they will always draw and the way we are set up the game your indifferent between drawing and losing so you learn nothing I mean basically so you will not know even learn to draw okay you will just learn to nothing basically you learn nothing because you can never win right so you are never going to get a reward of 1 so it will just be playing randomly.

So it is this is a bad idea so let us assume that you have an agent that in that is imperfect right that makes mistakes so that you can actually learn to figure out where the agent makes mistakes where the open it makes mistakes and learn to exploit those things okay, right. So your states are going to be this board positions as you can see we give you see a game that has been played out on the top of the slide right and the actions you take or in response to those board positions.

And finally at the end of the game and if you win you get a 1 if you do not win you get a 0, right. There are people you lose a 0 if you draw 1-1 sure you could even know other things like if you win it is one if you lose this minus 1 yeah you possibly could but you probably have to pay a lot of not of games because the perfect opponent it is almost impossible for you to start getting any feedback in the beginning right you will always be losing.

So it is going to be hard for it learn but you will eventually learn something yeah it will take a lot of Missoula will she learn something if I say that at every point like at a particular stage the probability of winning and like you are having to us date so your storing information for your storing information for each and every state that you have entered right so how will it be different from exploring the proper know like states place every time.

Because after you have done let us say a thousand ten thousand games or a million games you really explored a lot of states and you have to store for each state the probability of you bidding at that point yeah and all that so how will that be different from exploring it again like I know the probability of winning oh this still I'm not even totally hope you are going to solve it okay let me explain that and then you can come back and ask me descriptions okay if you still have it okay great.

So what the way we are going to try and solve this game is as follows right for every board position I am going to try and estimate the reward I will get if I start from there and play the game to the end, right every board position I am going to look at the reward I will get if I start from there and play till the end, now if you think about it what will this reward connect Connor Tate right.

So if I win from there I will get a 1 if I lose from that or if I do not win from that I will get a 0 right, when I say what is the reward I expect to get starting from this board position right it is essentially this average over multiple games it some games I will win some games I will lose I or I will not win like some games I win some games I will not win so what will this expected reward represent.

After having played many games, the probability of winning right the reward is going to represent the probability of winning in this particular case that if the robot had not been one right if you had been something else if it had been +5 that you would have been some function of the probability of any right after it has been +1 for winning -1 for loosing and 0 for draw well it is something more complex.

It is no longer the probability of winning right it is the gain a expect to get right how much what fraction of games I expect to win over the fraction of games I expect to lose or something like the rate it becomes a little bit more complex, so that there could be some interpretation for the value function but in general it is just the expected reward that I am going to get starting from a particular board position okay.

So that is what I am trying to estimate right let us assume that I have such a expectation well defined for me right as I say I have such an expectation well defined right now I come to us a specific position let us say I come to this position here let us say I come to this position how will I decide what is the next move I have to make, sorry whichever next state has the highest probability of ending.

So I just look ahead to see okay where if I put if I put the X here right if I put the X here what is the probability of winning if I put the X here what is the probability of winning if you put takes here what is the probability of winning and I do this for each one of these right and then I figure out which ever has the highest probability of winning and I will put the X there alright so that is how I am going to use this function.

Does it make sense yes it is very important so this is this is something which you should understand this is the crux of all reinforcement learning algorithms right I am going to learn this function that tells me if you are in this state right if you play things out to the end what will be the expected payoff that you will get right whether the rewards or punishment or cost whatever you want to call it what is the expected value you are going to get and I want to behave according to a this learnt function.

So when I come to a state I look ahead figure out which of the next States has the highest expectation and then go to the state okay, great how do I learn this expectation. What is the simplest way to learn the expectation, this is actually keep track of what happens this is to keep track of the trajectory through the game tree that you play a game you go all the way to the end right so you keep track of the trajectory and if you win right you go back along the trajectory and update every state that you saw on the trajectory.

You update the probability of winning right is this increase it a little bit or you come to the end of the game and he found that you are not one if you go back around the subject tree decrease the probability of winning a little bit right alternatively you can keep the history of all the games you have played so far right at every after every game has been completed you can go back and compute the average probability of winning across the entire history of all the games in which you saw that particular position right make sense this easiest way of estimating this probability right but the problem with this is A you have to wait till the game ends right or you have to store the history of all the games you have played item in all of these could be potential drawbacks okay you can get around the history part by coming up with an incremental rule but the main difficulty here is you have to wait all the way to the end of the game right before you can change anything along the way.

So tic-tac-toe is easy like how many moves can you make in tic-tac-toe at best, four right the fifth one is determined for you and so it is basically four choices that you can make right there so and that is easy enough to remember right you can always wait till the end of the game and then you can always make the updates right what if it's a much more complex situation right what if you are playing chess.

Maybe you can wait till the end so what if you are cycling maybe you can wait till end, exactly he we do not know right this it depends on where your cycling if you are cycling learning to cyclonite embedded it is fine when you are learning to cycle somewhere on the southern portal road even do not even think about what end is there right so this is not some tasks for which you really like to learn along the way right.

So this is where TD learning comes into it comes to help wait I do not think I have a slight anyway and I am not using the fancy thing where I can draw on the on the projection so let us see if I can do it here right suppose I have come here right and from here I have played at this point I know the probability of winning is say 0.4 right so I came here by making a move from this position.

So I said we were here net and we made a we know that the probability of winning from here is 0.3 right but I made the move from here to come here right but here I had thought my probability of winning was let us say 0.6 then I thought my probability of winning was 0.6 right but then I looked at my next States and I found that the best one was 0.3 somehow right so I went there right but now since the best I can do from here is 0.3 me saying 0.6 here there is something wrong right so I should probably decrease the probability of winning from here right.

So why could it be why could it have happened that I thought that was 0.6 but the best among the next was 0.3 may be the other 0.4 that was fully explored all of that the I thing is so that whenever I came to came through this part right maybe I one before right that it so happened that when I went through like this right initially I would have gone through like this and played the game and I am the examples I drew I might have actually won some of those games right.

So I would have changes 2.6 right but it is possible for me to get here by playing a different sequence of moves also right so for example to come here right I could have put the X first here and then here or I could have put the X like I did here I put the X first here and then here but either way I would have reached this position right so there are many combinations in which I could have reached the same positions right it just to be nice to these guys right to reach here there are different orders in which I could have put the zeros and the X'S right here we are showing a specific order now 0 first 1st put here then put here that the x was first put here then put in it could very well be I put the X first here and the o first here and then I put the X here in the O here right the multiple ways in this thing goes reach right.

So sometimes when I play those games right I lost right sometimes when I play these games I won therefore it turns out that for due to some random fluctuations right so sometimes I will when I go through this specific point and therefore I have a higher evaluation of winning right but when I went through the other paths I had a lower evaluation of winning right but we know that really does not matter what path you went through in tic-tac-toe right once your each that point right what is going to happen further is determined only by that point right.

So what I can do now is take this point 3 right you should update that 0.6 down so I am very confident here I think will the probability of 0.6 right but the best probability I have from the next stage is 0.3 therefore here is should not be so confident right good point that depends on how sarcastic your game is right so if you are game has a lot of variability then you do not want to make a complete you know commitment 2.3 so you might want to say okay now let me move it little bit towards 0.3 right but if it is a more or less a deterministic game then you can say okay yeah 0.3 yes sure let me go to all the way 0.3 it depends on the nature of the game we will talk about these issues later there but how far down is a tricky thing yeah it is misleading it is called game tree actually but it is a game graph in this case yeah .

So as I said kid when this is this is an instance of temporal difference learning so how while I use the thing to update this is called temporal difference learning ok so there is one other thing which is Should mention here right if I always take the move that which I think is the best move right now right let us talk about it I start tab Laura's I never played tic-tac-toe before right so I play the game I play it once I get to the end I win so now what I do I go back right whether I am using temporal difference learning or waiting till the end up dating whatever it is I change the value of all the moves I have made in this particular game right.

So the next time I come to a board position what am I going to do I look at all possible outcomes everything except the one that I have played will have a 0 right and the one that I have played will have something slightly higher than 0 I am going to take that in fact it will be like how many of you watch the movie Groundhog Day it will be like Groundhog Day I will be playing I will be playing the same game again and again because that is what happened to give you win in the first time around right that but that might not be the best way to play this right.

So I need to explore right so I need to explore multiple options right so I should not be always playing the best rest room right and I should always be paying the best move I need to do some amount of exploration so that I can figure out if there are better moves than what I think is currently the best move right so tic-tac-toe there is inherently some kind of noise if your opponent is random right but if an operand is not random and if operand is also playing a fixed rule and if you are playing greedy then you will be just play a very small fraction of the game tree and you would not have explored the rest of the outcomes right.

So you have to do some amount of things at random so that you learn more about the game right so here is a question for you when I am estimating this probabilities of winning right let us say I have reached here I look downright and the action that gives me the highest probability of winning set gives me a probability of say 0.8 right but I want to explore right so I take an action that gives me a probability say 0.4 okay so I will go from here to another action that has a probability 0.4 another board position that has a probability of 0.4 of winning so should I use this 0.4 to update this probability or not know why that you are questioning the whole TD idea yeah okay any other answer because you are learning that whether good or a bad move will be found out I.

Have to update the value of that smooth I agree to update the value of winning from the previous birth position was the question so that 0.4 I will have to change right but to exchange the 0. 8 that was a question the 0.8 was a probability of winning from here right I look or whatever the probability say I had a probability of winning of 0.6 from here I look at the bottom and the best probability of winning says 0. 8 but then I take because I am exploring I would take an action that has a probability of winning of 0.4.

Right the question is do I go back and change the 0.6 towards point for or do I leave the 0.6 as it sorry I am exploring rate I mean this is we may be necessarily be less than 0.8 this will be 0 .4 for that will be 0.6 so the question here is see one way of arguing about this is to say that hey if I am playing to win right I will play the best action from here right then the best action says 0.8 therefore I should not penalize it for the bad action which is 0.4 which I did to learn more about the system.

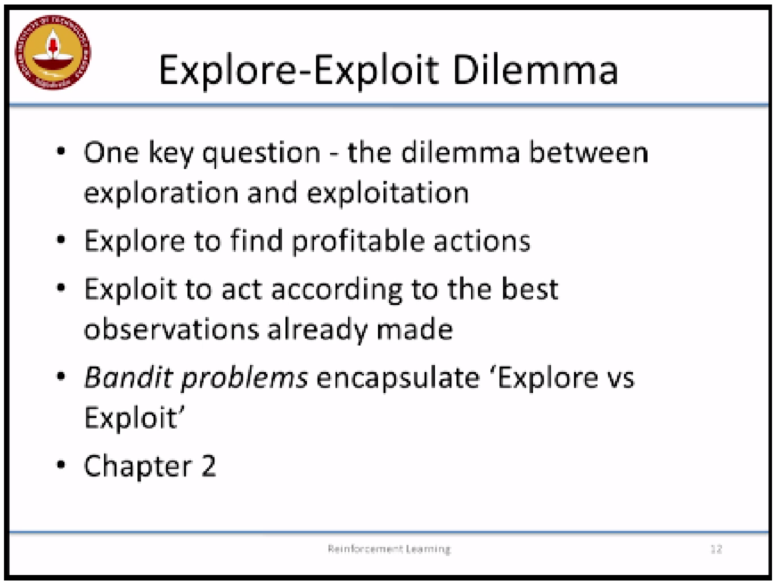
That is one way of thinking about it another way of arguing is to say that hey no this is how I am actually behaving now right so I should give you the probability of winning a book about the current behavior policy right this should not be some other ideal policy should be about to what I am behaving currently and therefore I should update it right so which one is correct first or the second you do not know what the second you would say the second one great so all of you can go right here in your homework answers.

Okay I think this is one of the homework questions as well right are you looking at the question I know you are browsing you are not paying attention to me but can you look at the question Jasper you do not have network here I see is our network does not work okay there should be a Wi-Fi access point called ICS our studio or something no fine do you have the question we have the book offline I do believe this one of the question say so otherwise make up a question and write the answer to that so we will feel will make sure this is one of the questions but this is something these are this is like I said I am going to ask you to think about the whole tic-tac-toe thing and many of these answers have relevance later on in fact there are two different algorithms one does option one and one does option two.

Right so there is no right answer or wrong answer right answer is depends yeah so yeah so this is a different things that you can think about in this but I told you about two different ways of learning with tic-tac-toe one wait till the end and figure out what the probabilities will be the other one this keep adapting this as you go along right and both cases you will have to explore that is it to key part here in both cases you have to explore otherwise you will not learn about the entire game so this is where the Explorer exploit thingy comes in okay yeah great question different algorithms deal with it different way that is one of the crucial questions that you have to answer in NRL.

So it is called the Explorer exploit dilemma aright so you have to explore to find out which is the best action right and you have to exploit.

(Refer Slide Time: 37:42)



Explore-Exploit Dilemma

- One key question - the dilemma between exploration and exploitation
- Explore to find profitable actions
- Exploit to act according to the best observations already made
- *Bandit problems* encapsulate 'Explore vs Exploit'
- Chapter 2

Reinforcement Learning 12

Whatever knowledge you have gathered right and you have to act according to the best observations already made right so this is called exploitation right so the key was the key questions is when do you know you have explored enough right should I explore know or should I exploit now this is called the Explorer exploit dilemma right and a slightly simpler version of reinforcement learning called the Bandit problems okay some colorfully called banded problems they of course he is an expert on bandit problems here you can the Bandit problems encapsulate this Explorer exploit dilemma.

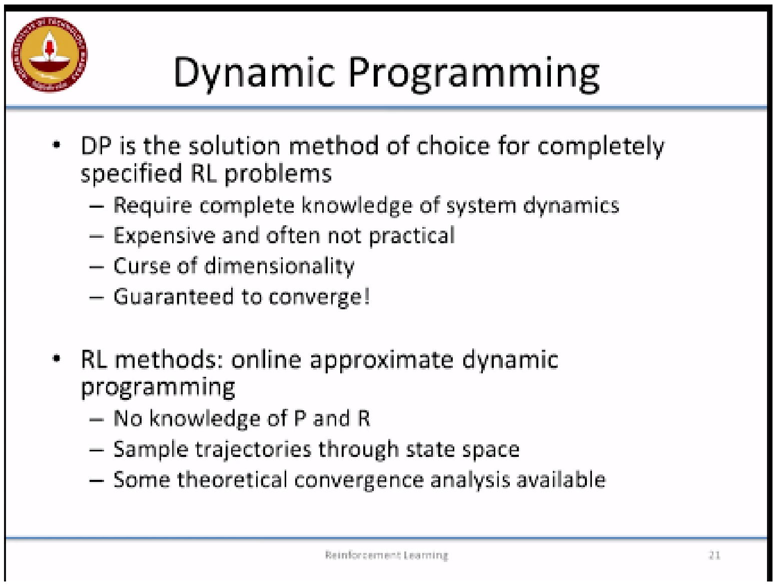
My god lot of people are turning and looking at irritable and so this is the next chapter we are going to look at this so probably in the next class I will start plunging into bandit problems in more detail but so this will ignore a whole bunch of other things like the delayed rewards you know the sequential decisions and other things even in the absence of all of these other complications that even if I say that you are all your problem is you have to take an action and you will get a reward okay your goal is to pick the action that gives the highest reward.

I will give you 10 actions you are to pick the action that gives you the highest reward right but the problem is you do not know what is the probability distribution from which these rewards are coming right so you will have to do some exploration I have to actually do every action at least once okay to know what will be the reward even if they are deterministic right so I cannot say which is the best action before I try every action at least once if it is deterministic it is fine I can just try every action once and I know what is the payoff right but if it is stochastic I have to try every action multiple times right.

How many times do you have to try it depends on the variability of the distribution right so these are the subtle questions which we will talk about as we look at the next chapter in the first chapter in the book the first real chapter in the book okay there is an intro chapter which talks about tic tac toe and a whole bunch of other things and yeah you ever stop soon not sure they do all of this yeah so we come back and talk about all of this later right so I thought I would not spend enough time talking about the general background of RNL applications so I put in a little bit more slides I typically do this on the board right.

So I will do this on the board when I get to chapter 3 right and so one thing which I wanted to tell you was this right there are you know two classes of algorithms that we will be talking about so one of them is based on what is called dynamic programming right.

(Refer Slide Time: 40:37)



Dynamic Programming

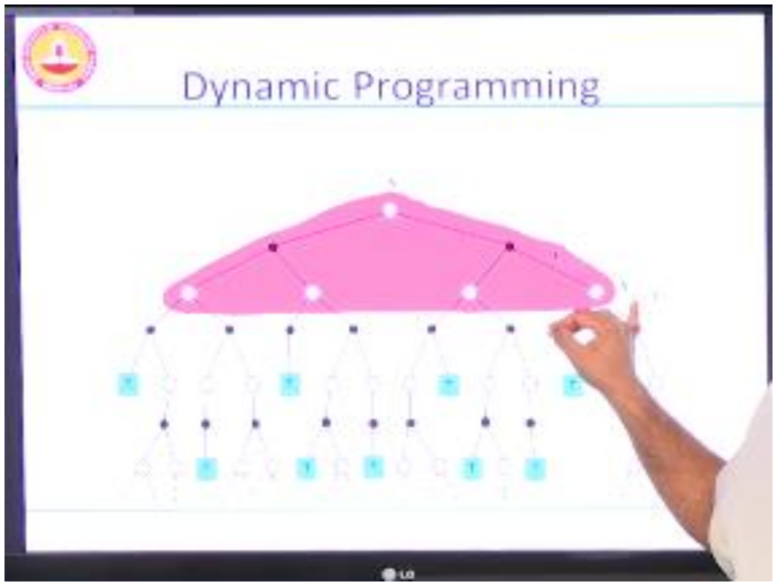
- DP is the solution method of choice for completely specified RL problems
  - Require complete knowledge of system dynamics
  - Expensive and often not practical
  - Curse of dimensionality
  - Guaranteed to converge!
- RL methods: online approximate dynamic programming
  - No knowledge of P and R
  - Sample trajectories through state space
  - Some theoretical convergence analysis available

Reinforcement Learning 21

So dynamic programming how many of you know what dynamic programming is I expect at least 23 hands out and I having mechanical engineers putting their heads have not the CS guys really you know what dynamic programming is something wrong with your hand then okay fine so what this essential idea behind dynamic I will talk again I will talk ass as usual with other things will well win this in larger detail the essential idea behind dynamic programming is you will be using some kind of repeated structure in the problem right.

So and to solve the problem more efficiently right suppose I have a solution that I can give for a problem of say size n right then I will try to see if I can use that for defining the solution for the problem of size n + 1 so some kind of a repeater sub structure in the problems right very rough way of describing more dynamic programming is right so for example one way of thinking about dynamic programming is I have this game tree right so I look at the values of winning or the expectations of winning from all of these steps right.

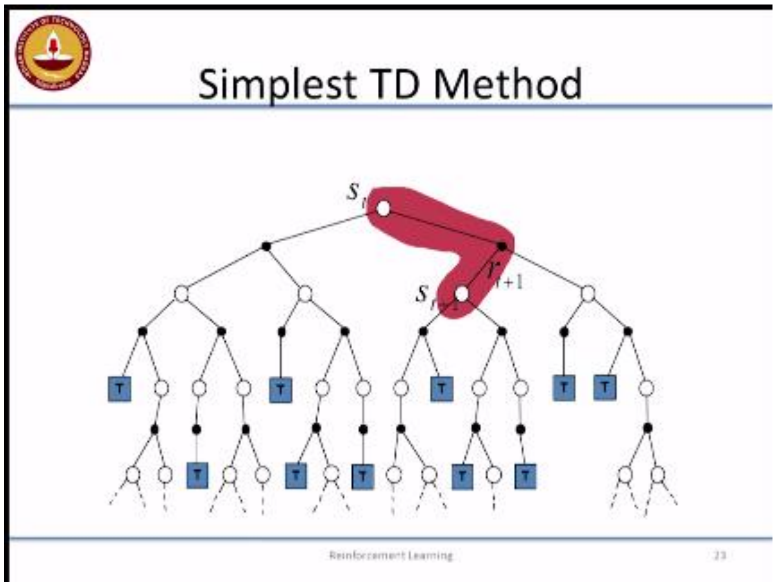
(Refer Slide Time: 41:56)



I will use these in order to compute the value of winning or the probability of winning from the state right so if you think about it if from here if I'm going to take say n steps from here how many steps will be expect me to take n- 1 steps right so I look at the probability of winning when I when I have only n - 1 steps left right I will estimate that first I will use that solution for estimating the probability of winning when I have n steps left right so that is essentially the idea behind dynamic programming and so what we will do is well we will talk about how do you use dynamic programming to solve reinforcement learning problems right.

And then we will talk about various reinforcement learning methods which are essentially online approximate dynamic programming you know so they are they use dynamic programming ideas but they do not have any knowledge of the system dynamics right so the P&R other things become clear later and so all you do now is instead of having the entire out come.

(Refer Slide Time: 42:58)



Simplest TD Method

And using that for estimating the probability of winning here right I am going to just use one step that I take through the tree right I use this what happens in this one step I will use that in order to update the probability of winning here rights instead of using the entire outcome right as in dynamic programming in reinforcement learning methods we will be using samples that you are getting through the state space.

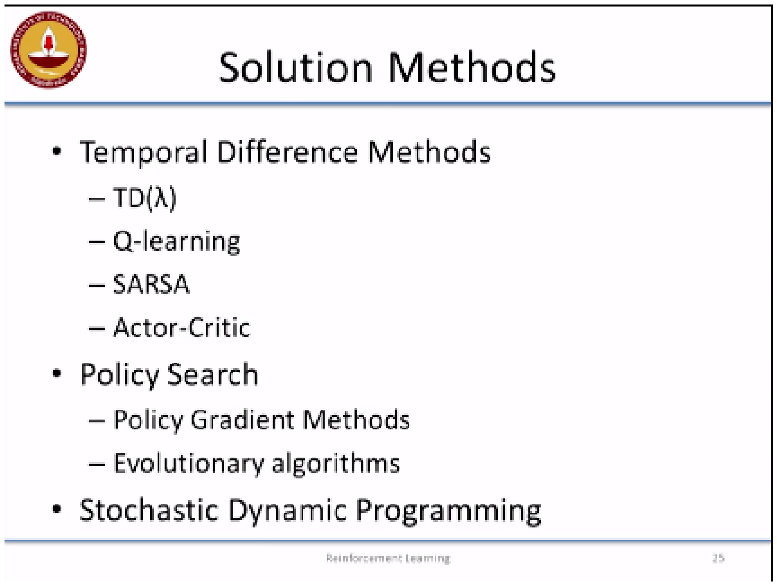
This is the TD method in the other method I explained to win for tic-tac-toe what would you do your sample will run all the way to the end right and you use that to update so this is the two different ways of using samples here so if you are doing it for the first time right first times down the tree there will be no updates actually because ST + 1 will also be 0ST will also be 0 the first time we go down there will be no updates but once you reach an end then from there you will start updating the previous day.

So the next time you go down the tree then it will keep going further up whether in what the game have lost the game I actually I am taking the exact outcome that happened in that particular trial right at that particular game and I used that to change my probabilities right but here I am not just taking the outcome of that particular game right I am looking at the expected value of winning from the next board position right so if I wait till all the way here and I say I won and it take this in update as T then I will be only updating it with the fact whether I won or not okay but if I am updating it from ST + 1 see I could have reached this T+ 1 multiple ways before right

When I am doing the update on from XT + 1 it is essentially the average accumulated over all the previous trails that I will be using right so if I play all the way to the end and update it will be with a one or a zero but ST+ 1 could be anywhere between 1 and 0 depending on what is the probability of winning,.

So I will be using that value for update that is a crucial difference I so there are many different solution methods so there are all this which are called temporal difference methods.

(Refer Slide Time: 45:43)



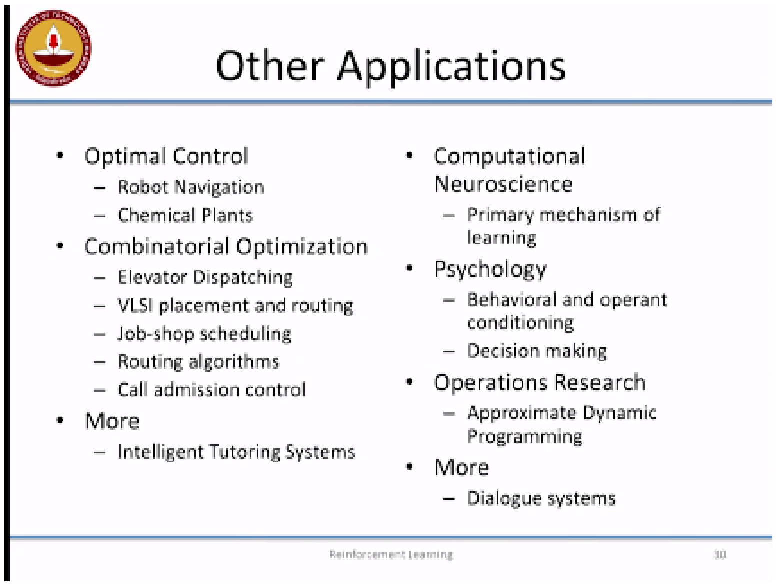
Solution Methods

- Temporal Difference Methods
  - TD(λ)
  - Q-learning
  - SARSA
  - Actor-Critic
- Policy Search
  - Policy Gradient Methods
  - Evolutionary algorithms
- Stochastic Dynamic Programming

Reinforcement Learning 25

So these are all different algorithms TD λ Q learning’s our sector critic so on so forth and then there is a so of algorithms which called policy search algorithms and then there is dynamic programming so will actually look at each of these classes of algorithms in the and the next few lectures right these are all essentially mostly from the textbook except policy search algorithms which I will be covering from other material right and then whole bunch of other applications for Aaron it.

(Refer Slide Time: 46:16)



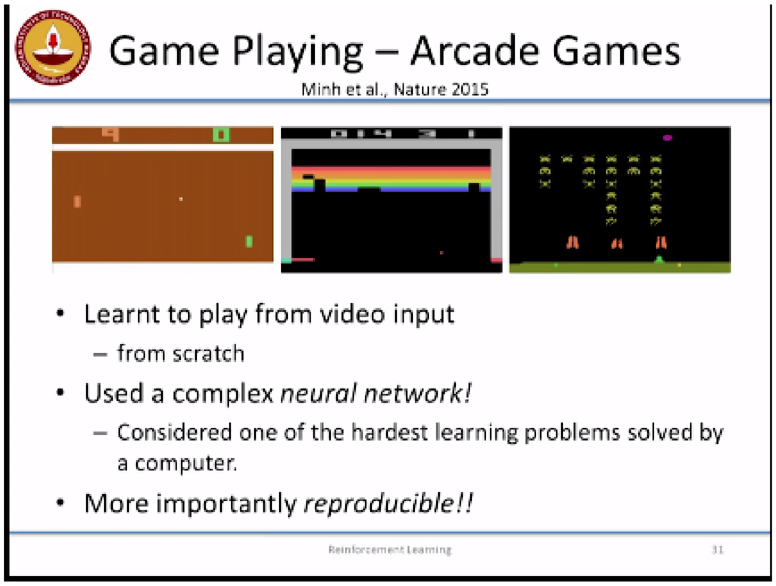
Other Applications

- Optimal Control
  - Robot Navigation
  - Chemical Plants
- Combinatorial Optimization
  - Elevator Dispatching
  - VLSI placement and routing
  - Job-shop scheduling
  - Routing algorithms
  - Call admission control
- More
  - Intelligent Tutoring Systems
- Computational Neuroscience
  - Primary mechanism of learning
- Psychology
  - Behavioral and operant conditioning
  - Decision making
- Operations Research
  - Approximate Dynamic Programming
- More
  - Dialogue systems

Reinforcement Learning 30

So could they are all over the place as you could see optimal control optimization company to combinatorial optimization for psychology neuroscience so that is not a three since it was asking is there anyone from biotech because biotech people do use reinforcement learning a lot and usually there are one or two people in the RL class so this is a surprise or maybe that is that was a bear trick with this trip according to the economic website I mean this is gave up in CS 36 and win back I do not know right.

(Refer Slide Time: 46:54)



Game Playing – Arcade Games

Minh et al., Nature 2015

- Learnt to play from video input
  - from scratch
- Used a complex *neural network!*
  - Considered one of the hardest learning problems solved by a computer.
- More importantly *reproducible!!*

Reinforcement Learning 31

So here is the most recent the hot thing that comes from came from RL right more game playing and for a change is not from IBM is from Google but the company that actually built the first this arcade game playing engine was called deep mind and as soon as deep mind built a successful engine Google bought them right and so now it is Google deep mind but it is a separate entity right it is not part of Google deep mind operate out of London and they are doing all kinds of interesting stuff many of the hot advances a very recent advances in the last year or so in the reinforcement learning seem to be coming out of different.

So what they did was how many if you know about this Atari games right everyone knows about Atari games people are getting tired really no one has played bat -man yeah how would how about pong breakout space invaders come on yeah anyway so what happened was this is team in University of Alberta ok which put out this their what they call the LED arcade learn and I the Atari learning environment or arcade learning environment which essentially they allowed computers to play these games right these Atari games and what the deep mine fellows came up with is a reinforcement un engaging that learn to play this game from scratch I just by looking at the screen.

Okay that is all the input it was getting just the pixels from the screen they are all pixels on the screen are given as inputs to it used to very complex neural network so it us a deep learning deep network right and it is considered one of the hardest learning problem solved by a computer and I believe it is a one the only computational reinforcement learning paper ever appear in nature right so usually is very hard for non natural science people to publish in nature kind of obviously it is usually Hartford computer science folks the population nature but this was totaled out as a next step in trying to understand how humans process inputs blah all kinds of marketing jargon right but more importantly than anything else about this it is reproducible.

So I told you about I think that is a warning sign for me to stop so I told you about the helicopter right so there is basically Stanford and Berkeley or the two people who get the helicopter o fly I told me about the backgammon player is like Jarrett SRO is the one person who gets the backgammon player to work right partly because all the input features he uses in there or proprietary but partly because has a very hard problem to solve right what is amazing thing about this Atari game engine is that this case are very sub code right you can if you have enough powerful GPUs right you can set it up here and get it to play and get a reasonably working engine right.

That plays those places Atari games that's the amazing thing about it that is reproducible as opposed to many of the other things other success stories other success stories we have had in the past so I do believe I had just one more slide after this so let us see if this will work.

(Refer Slide Time: 50:14)

Google Deepmind's Atari Player

Before training

Reinforcement Learning 32

Okay so if you are any doubts the green one is the learning agent is it no this just sped up for you to see you mean it is not like the game is progressing at the same rate but you can see the score slow mind you it was not given a reward okay it is just given the screen never got to reward for winning ideas and understand what the pixels on the top or rewarding and if we give it roars it becomes cheating right how does it which is what they did they did they did at a game over which is misty the Ailey pure is considered as cheating but they did not a game over side so the longer you keep it going the better it is basically nothing this is getting boring right.

So it is learnt here so this is a sea quest you have to swim and then sink down get some things and come up so sequence is a game that it never learned to do greatly on and sequence is not something which it learn to solve well so there are a few games like this so they initially published the nature paper I think they had like 45 or 46 out of the 50 ale games they were able to play well and I think in 43 of them it had better than human performance and I think the current state is they have like one game that doesn't play well right and have better than human performance in like 48 of those games or something.

So is this is amazing you think about it right so these are all lot of references but we will be using the Sutton and bought Oh textbook the second edition we there is a pre-release copy which rich certain this continuously updating on his webpage right and the very short introduction to heading for learning in Tom Mitchell's book and there is a chapter in Russell and nor wig and if you are interested in looking at the neuroscience perspective of reinforcement learning or the history of how behavioral psychology models evolved to temporal difference learning you can look at Diana and Abbott it is one chapter on reinforce learning you do not have to understand the neuroscience to understand the RL chapter.

So do not get put off by the title of the book right and then the more mathematically well grounded introduction to reinforce money would be from vs occurs and sickness so apart from this we will use one book which is Markov decision processes by putter men that will use certain chapters from put Roman and they are just for us to understand MDPs better motivation process better.