[Music]

Welcome to the third part of the lecture on reinforcement learning, this part will actually only be on the special time difference method called Q-learning. Q-learning is a model free of policy temporal difference reinforcement learning algorithm. The goal of Q learning is to learn a policy which tells an agent what action to take under what circumstances. So for any finite Markov decision process FMDP, Q-learning finds a policy that is optimal in the sense that it maximizes expected value of the total reward over any and all successive steps, starting from the current state. Q-learning can identify an optimal action selection policy for any given FMDP, given infinite exploration time and partly random policy. "Q" names the function then Q(s,a) that can be said to stand for the "quality" of an action a taking in a given state s. And assuming that we will calculate an optimal Q function (s,a) the optimal policy is in the state s, is actually the maximum argument, the maximum value in the Q function table corresponding, it's actually the action corresponding to the max value and in the Q function table for the state s. So it's pretty simple to infer.

Here you see Q-learning algorithm in a pretty abstract form, so as you can see in the beginning it says Q(s,a) so that is the this quality function. In practice this quality function is represented by a table where the rows are the states correspond to the states and the columns correspond to the actions. When you start you can have an arbitrary values there, you can have zeros or what you like. So what the algorithm does is that it goes through a number of episodes with the same semantics of episode as we have discussed earlier, it goes through that this out stage by step by step so it takes the first state and then for each step it updates the quality function which means essentially that it updates the different items in the table and the central formula here is actually that you take the old value, then you add an expression which is the sum of the reward for taking that step, and the discounted Max value of the Q value for the next stage is to become to, so the max value overall new actions that one can take from that state, and then you subtract with the old value here. So the gamma is this discount parameter that we have seen earlier and the Alpha here is normally called some learning rate, so essentially if you have a low value of alpha, the effect of the change for each step can be diminished so by having an alpha which is one, you have the full effect of the change if you have a lower value than one, you can damp the effect of the changes from step to step. So these are this kind of normal hyperparameters that all learning algorithms have. As you can see we with alpha as one or Gamma also one the formulas gets more simple.

So we will look now at in detail on a small example to illustrate how all this works. So you can see here we have a board with a 5 x 4 board and all the elements on the rim has a reward value of -8, those in the yellow ones in the middle has a reward value of 0, and only a single one item on the rim as real value 8 and the blue dot shows which is the start state. So one can enumerate the states in sequence for this purpose this learning rate alpha is set to 1 and the gamma is to 0.5 and then we have the various actions which is north south east west. So then you can have a look as I said the Q function is actually a table sorry I think I said something wrong earlier and so you see the actions are the rows and the states are the columns not the opposite. Yeah and this is the realization and obviously here in this case we have said I just set all the values to 0. So then as you understood from the algorithm we look at an episode, so we look here at an episode with 4 steps going from 12 to 13 to 8 to 7 and then 2, and then we go through the algorithm and what we do is that we update the Q values in those table entries which are concerned by this particular episode and you can see here step by step how the calculation is done exactly with the with the parameter values we had set and the actual Q values. So not much happened here until the end, so it's only in the first step that we get a new Q value in one of the table entries. So it still it still very boring table, now we got the -8 in one element. Okay so then we try another episode starting from twelve and as an alternative moving to choosing a path towards the end terminal with a positive reward. Actually then we can make a similar calculation so we also end up with a new Q-value in another position. So this is how it goes on and we can take more episode and the values will change, so gradually the this table will take shape. This is just to show you that after a while you can we will reach a state with a Q value that happens to be one of the optimal policies and that could be depicted graphically like in this slide. The another optimal policies could look like this and this is as another pattern. So this is an illustration of this Q learning, so Q learning we looked at many algorithms, obviously here of different kinds, Q learning is a pretty straightforward method to illustrate how this can work. So this is the end of this lecture. Thanks for your attention the next lecture will be on the topic case based reasoning.