Mini-project 1
Artificial neural networks CS-456

Albert Aillet, Damien Gomez Donoso

June 2022

2 *Q*-leaning

2.1 Learning from experts

Question 1

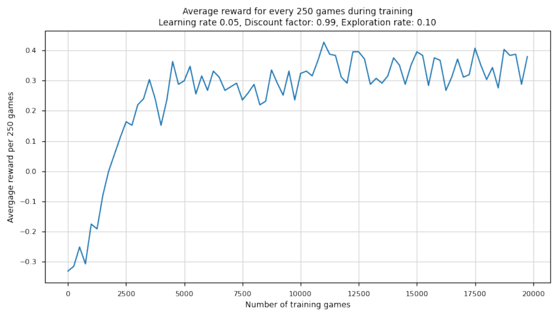


Figure 1: Average reward during training with a fix value of $\epsilon = 0.1$.

After running the Q-leaning agent with $\epsilon = 0.1$ against `Opt(0.5)` for 20'000 games, we notice that the agent learns to play *Tic Tac Toe* quite fast during the first 5'000 games as the average reward quickly increases, after that it does not seem to learn as the reward does not increase further.

2.1.1 Decreasing exploration

Question 2

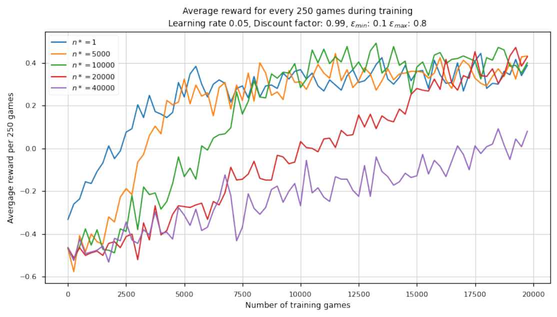


Figure 2: Average reward during training with decreasing exploration for different values of n^*.

In figure 2, it can be seen that when using larger values of n^*, the average reward increases slower. For $n^* = 1$, epsilon is always equal to ϵ_{min} and we therefore get a very similar graph to the one in figure 1. For $n^* = 40'000$ ϵ always has a large value and the agent therefore often takes random actions, in consequence the average reward does not increase very much. For $n^* = 10'000$, decreasing ϵ helps training as around 10'000 games, the agent achieves higher average reward than without using decreasing exploration.

Question 3

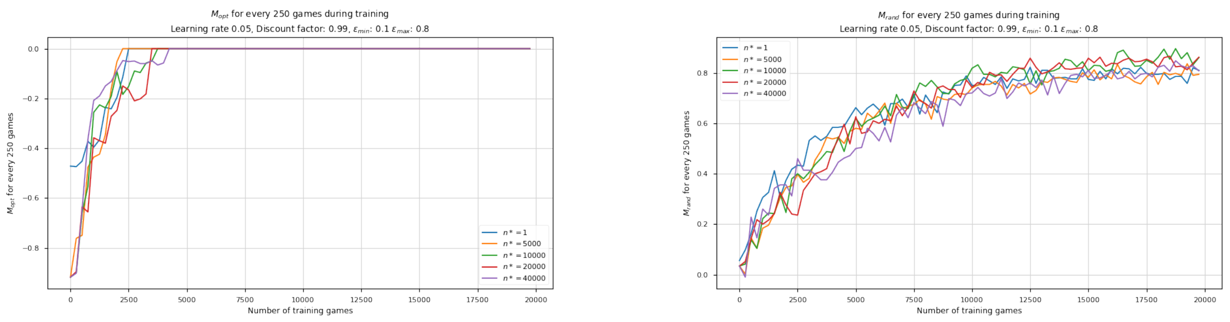


Figure 3: M_{opt} and M_{rand} during training with decreasing exploration for different values of n^*.

Similarly as for the average reward seen in figure 2, we can see that the values of M_{opt} and M_{rand} in figure 3 increase slower with larger values of n^*. It is however interesting to note that even if the average reward for the run with $n^* = 40'000$ is lower during training as it then uses a large ϵ, when testing it and using $\epsilon = 0$, it performs very similarly to the other runs, meaning that it has not learnt less from the increased exploration.

2.1.2 Good experts and bad experts

Question 4

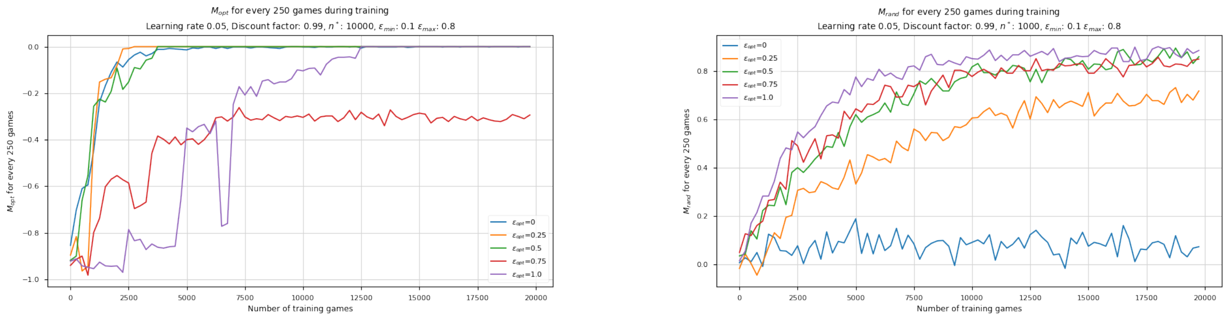


Figure 4: M_{opt} and M_{rand} during training with decreasing exploration for different values of ϵ_{opt}.

For this part, we choose $n^* = 10'000$ because the Q-leaning with decreasing exploration has good behaviour with this value. In figure 4 we see that when ϵ_{opt} is large, M_{opt} increases slower and for $\epsilon_{\text{opt}} = 0.75$, M_{opt} never reaches 0. On the contrary, M_{rand} increases slower when ϵ_{opt} is small. Both of these observations make sense, as when we train our agent against a close to optimal player (small ϵ_{opt}) the agent performs well against an optimal player (large M_{opt}) and when we train our agent against a close to random player (large ϵ_{opt}) the agent performs well against a random player (large M_{rand}). The agent learns how to respond to specific play-style.

Question 5

The highest values of M_{opt} and M_{rand} that we could achieve after playing 20'000 games was $M_{\text{opt}} = 0$ and $M_{\text{rand}} = 0.902$.

Question 6

Assuming that Agent 1 learns by playing against `Opt(0)` and finds the optimal Q-values $Q_1(s, a)$ and assuming that Agent 2 learns by playing against `Opt(1)` and finds the optimal Q-values $Q_2(s, a)$, we argue that $Q_1(s, a)$ and

$Q_2(s, a)$ do not have then same values. This is because Agent 1 that plays against the optimal player will never encounter certain states that would be possible for Agent 2 to encounter. For example, imagine we are in the state illustrated in figure 5. `Opt(0)` would never let such a state happen because the state would let the learner agent win in the next move. The Q-values for Agent 1 for this state will therefore always stay 0. That's not the case for Agent 2 that plays against `Opt(1)` that make sub-optimal actions and lets the learner win. This will change the Q-values of this state from 0 for Agent 2.

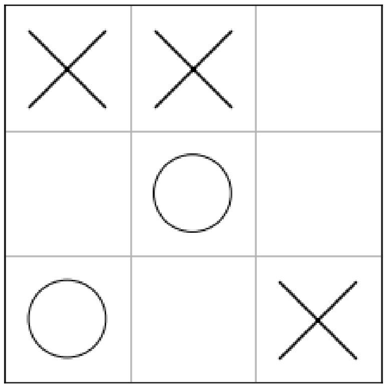


Figure 5: Impossible state with `Opt(0)` playing as X.

2.2 Learning by self-practice

Question 7

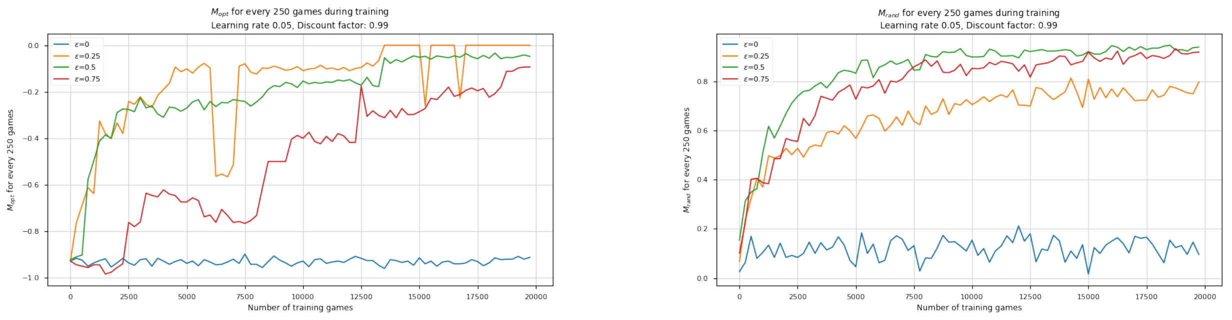


Figure 6: M_{opt} and M_{rand} during self-practice training using epsilon-greedy policy for different values of ϵ.

The agent learns to play *Tic Tac Toe* as both M_{opt} and M_{rand} increase during training.
M_{opt} reaches higher values for smaller values of ϵ different from 0. The highest M_{opt} is obtained with $\epsilon = 0.25$.
M_{rand} reaches higher values for larger values of ϵ. The highest M_{rand} is obtained with $\epsilon = 0.5$.
For $\epsilon = 0$, we obtain low values for both M_{opt} and M_{rand} as the agent will not explore and get stuck in a sub-optimal type of play with itself.

Question 8

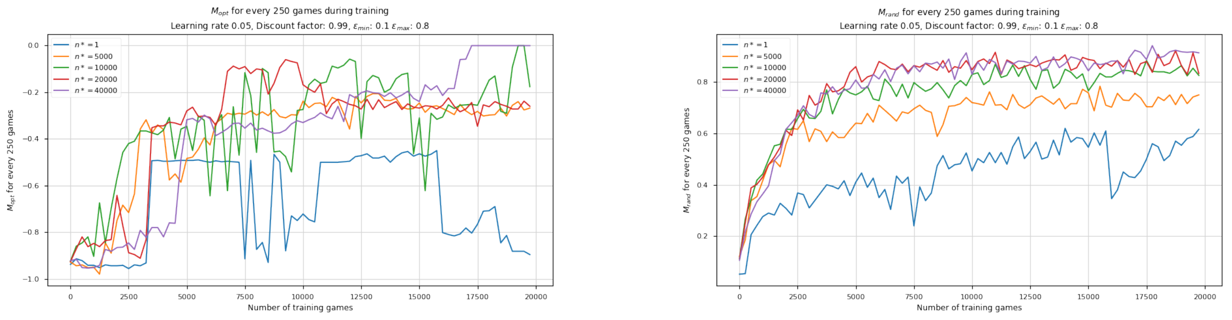


Figure 7: M_{opt} and M_{rand} during self-practice training using epsilon-greedy policy with decreasing exploration for different values of n^*.

Decreasing ϵ during training does not help to improve the results of M_{opt} and M_{rand} compared to having a fixed ϵ. M_{opt} does not reach higher values than when using a decreasing ϵ and is not stable as it sometimes jumps between values. For M_{rand} we obtain very similar curves as when using a fixed ϵ. For larger values of n^*, we obtain larger final values for M_{opt} and M_{rand}.

Question 9

The highest values of M_{opt} and M_{rand} that we could achieve after playing 20'000 games was $M_{\text{opt}} = 0$ and $M_{\text{rand}} = 0.942$.

Question 10

Figure 8 shows the Q-values for 3 board arrangements. For the first board at the left, the disposition of the Q-values does not have a clear explanation as it is the empty board.

The highest Q-value is on the slot $(0, 1)$ and we can suppose that our agent has won a lot of time when he starting with this slot.

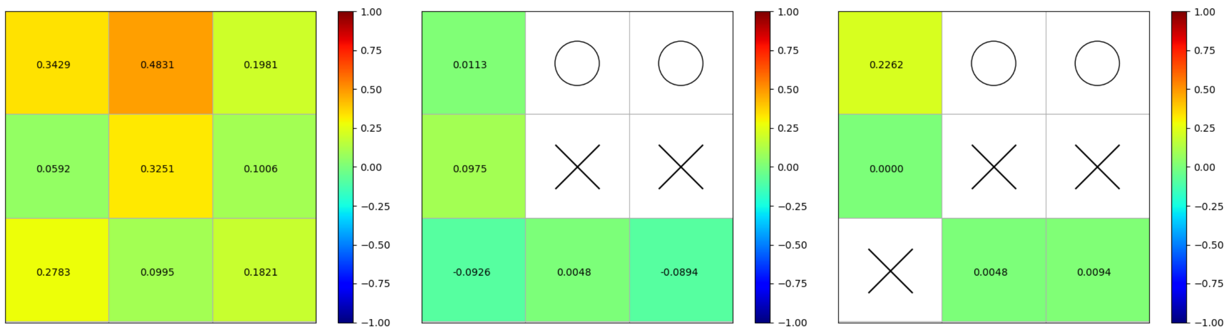


Figure 8: A few tables and their Q-values.

If we look at the second board in the middle, the result makes sense as for the player 'X' to win, it should choose the spot $(1,0)$, that indeed has the highest Q-value.

Similarly, for the third picture on the right, for the player 'O' to win, it should choose the slot $(0,0)$ which has the highest Q-value.

We can conclude that our agent has learnt the game well.

3 Deep Q-leaning

3.1 Learning from experts

Question 11

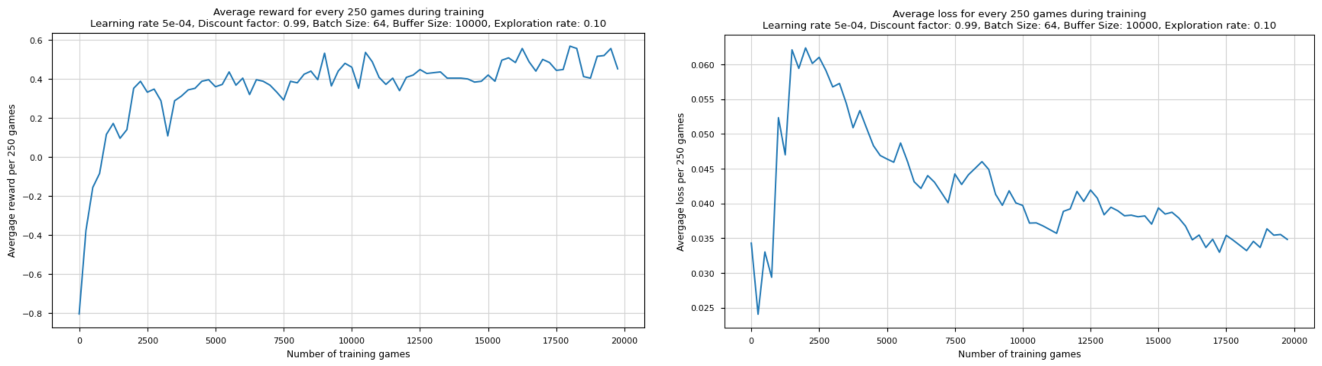


Figure 9: Average reward and loss during training using epsilon-greedy policy with $\epsilon = 0.1$ and Deep Q-leaning with a replay buffer of size 10'000 and a batch size of 64.

In figure 9, it is apparent that the loss first goes up quickly and then decreases to end up around 0.035. The average reward first increases quickly up to 0.4 and then increases very little. We can conclude that the agent learns how to play *Tic Tac Toe*.

Question 12

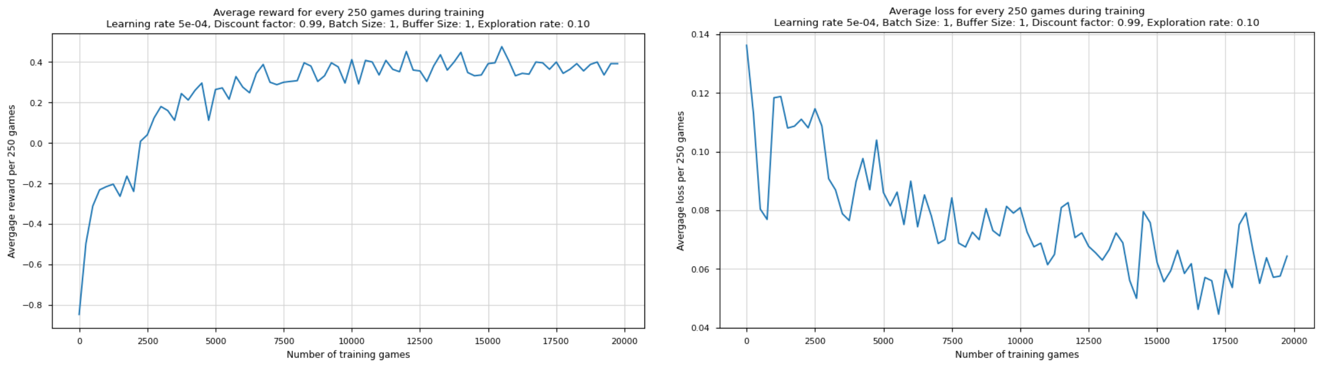


Figure 10: Average reward and loss during training using epsilon-greedy policy with $\epsilon = 0.1$ and Deep Q-leaning with a replay buffer of size 1 and a batch size of 1.

In figure 10 it can be seen that the average reward stays around 0.4 while in the previous question and in figure 9, the average reward reached 0.5. The average loss does not go below 0.4 while in figure 9 it was around 0.035. However as the average reward increases, the agent still learns how to play *Tic Tac Toe*.

Question 13

When using $n^* = 1$, it is equivalent to using a fixed $\epsilon = \epsilon_{min}$ except for the first game. By comparing $n^* = 1$ to the other values of n^* in figure 11, there does not seem to appear any significant differences when using a decreasing epsilon. Varying n^* does not affect M_{rand}, as it always stays around 0.9. For all n^*, M_{opt} reaches 0 at least one time. While not very significant, the stability of M_{opt} seems to increase for smaller values of n^*. We can also notice that the M_{opt} increases (on average) slower for larger values of n^*. For $n^* = 5$'000 the decreasing exploration method attains high values for both M_{opt} and M_{rand} quickly.

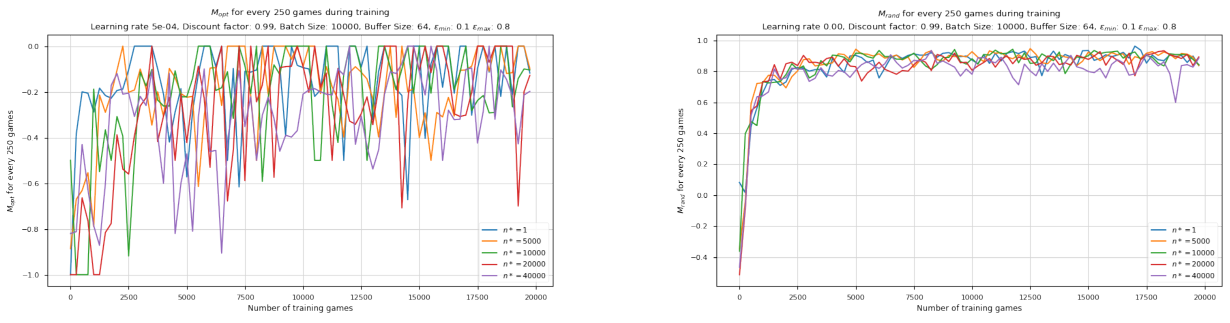


Figure 11: M_{opt} and M_{rand} for Deep Q-leaning using epsilon-greedy policy with decreasing exploration for different values of n^*

Question 14

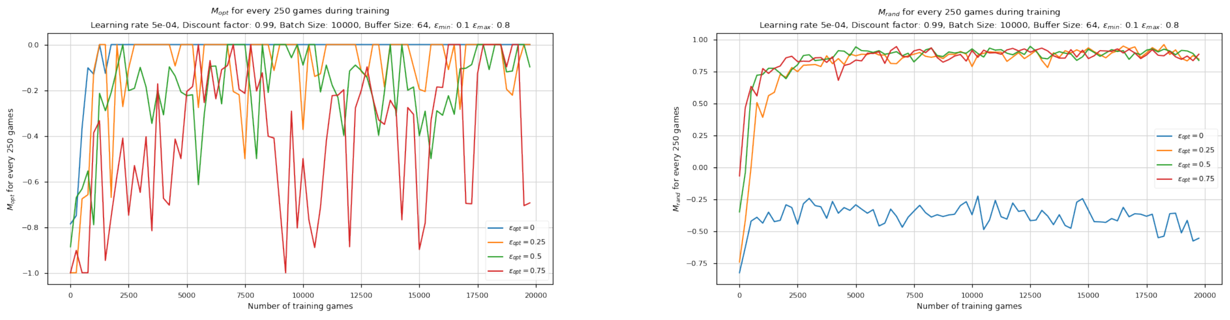


Figure 12: M_{opt} and M_{rand} for Deep Q-leaning using epsilon-greedy policy with decreasing exploration $n^* = 5000$ and for different values of ϵ_{opt}

As seen in figure 12, for M_{rand}, we obtain the same result (around 0.8) for all values of ϵ_{opt} except for $\epsilon_{\text{opt}} = 0.0$ that gives a much lower result M_{rand} around -0.3. This can be explained by the fact that if we train an agent against an optimal player that only makes optimal moves, the agent learns only one way to play, i.e. the optimal way. This agent would be overfit to the optimal playstyle and would not play well against a player taking sub-optimal actions. For ϵ_{opt} different from 0, the agent has high values for M_{rand}. This is due to the optimal player sometimes making sub-optimal moves during the training phase. We also observe that M_{rand} increases very quickly during the first 2'500 games but then stays close to constant.

M_{opt} becomes higher for smaller values of ϵ_{opt}, which is explained by the fact that for small values of ϵ_{opt} the agent learns against a close to optimal player that often makes optimal moves and will therefore learn the optimal way to play *Tic Tac Toe*.

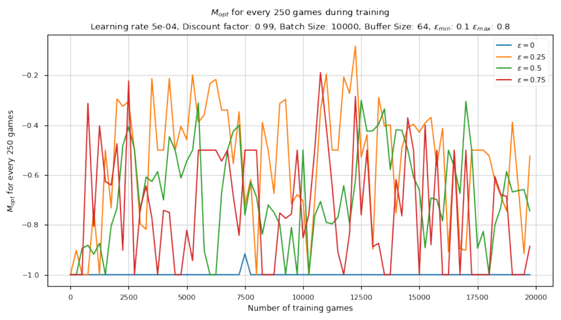
Question 15

The highest values of M_{opt} and M_{rand} that we could achieve after playing 20'000 games was $M_{\text{opt}} = 0.0$ and $M_{\text{rand}} = 0.964$.

3.2 Learning by self-practice

Question 16

As seen in figure 13, M_{opt} presents instability in the results, i.e. the curves make big "jumps", but the tendency for M_{opt} seems to be that it reaches higher values as ϵ decreases. However, that's not the case for $\epsilon = 0$ for which M_{opt} stays around -1. The best value we obtain is $M_{\text{opt}} = -0.082$ for $\epsilon = 0.25$. M_{rand} initially increases to a value around 0.75 for all ϵ different from 0. After an initial increase up to game 1'000, M_{rand} seems to decrease as number of games increases. Here $\epsilon = 0$ is also an exception for which M_{rand} does not have values larger than 0.5.



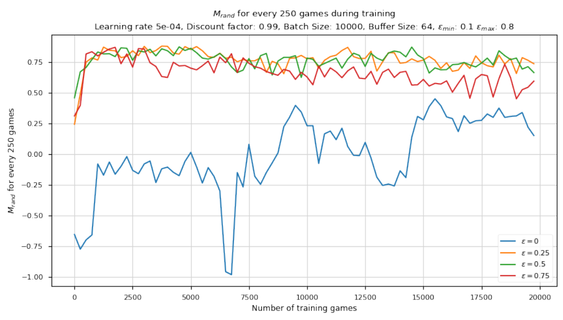
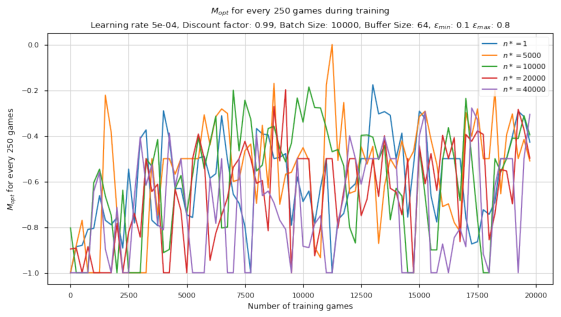


Figure 13: M_{opt} and M_{rand} for Deep Q-leaning by self-practice using epsilon-greedy policy for different values of ϵ decreases. But that's not the case for

Question 17



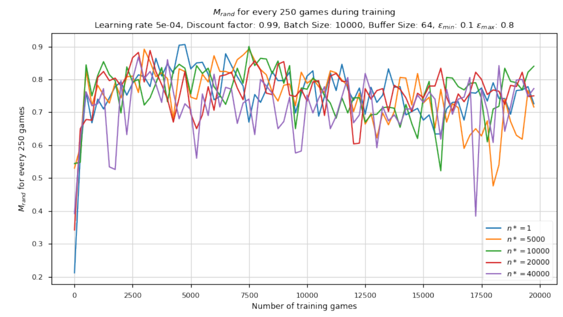


Figure 14: M_{opt} and M_{rand} for Deep Q-leaning using epsilon-greedy policy with decreasing exploration for different values of n^*.

In figure 14, M_{opt} has similar behaviour as in figure 13. However we can observe that for $n^* = 5'000$, M_{opt} reaches 0 at one point, which is a higher value than for M_{opt} in figure 13. We observe higher results for M_{rand} because, in figure 13, it was around 0.75 and now it is around 0.8 (for the 10'000 first training games). However, the M_{rand} values present more instability than for fixed ϵ. After an initial increase up to game 2'500, M_{rand} seems to decrease as number of games increases. As we get higher values for M_{rand} and M_{opt}, we can conclude that decreasing exploration helps compared to having fixed epsilon. From figure 14, we could not observe any significant effects from n^* on M_{rand} and M_{opt}.

Question 18

The highest values of M_{opt} and M_{rand} that we could achieve after playing 20'000 games is $M_{\text{opt}} = 0.0$ and $M_{\text{rand}} = 0.91$.

Question 19

In the first board to the left in figure 15, which is the empty game board, the agent predicts high Q-values for the center and bottom left slot. This is not directly explainable but may be due some random choices of the agent during training. If we look at the second board in the middle, the agent predicts a high Q-value for the slot $(1, 0)$ as this will win it the game. For the third board on the right, the highest Q-values is on the slot $(2, 0)$. That makes sense because the agent wants to avoid that the opponent ('X') wins by blocking it and putting an 'O' in this slot. Note that for this board, there are only negative Q-values because it's sure that the agent will lose with such a state. We can conclude the agent learns to play *Tic Tac Toe* well.

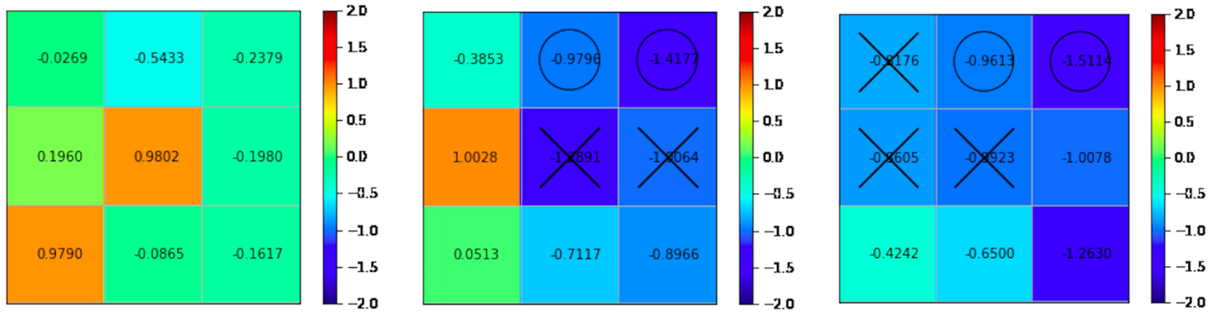


Figure 15: A few tables and their Q-values.

4 Comparing Q-leaning with Deep Q-leaning

Question 20

Algorithm	highest M_{opt}	highest M_{rand}	T_{train}
Q-learning with learning from experts	0.0	0.902	3000
Q-learning with learning by self-practice	0.0	0.942	6750
Deep Q-learning with learning form experts	0.0	0.964	1000
Deep Q-learning with learning by self-practice	0.0	0.906	5250

Table 1: Table showing the best performance (the highest M_{opt} and M_{rand}) of Q-leaning and Deep Q-leaning

Question 21

From table 1, the best algorithm to train the agent seems to be the Deep Q-learning with learning from experts method as we get the lowest T_{train}, the highest M_{rand} and the highest M_{opt}.

Let's focus our attention on M_{opt}. While all the methods as shown in table 1 lead to $M_{\text{opt}} = 0$, they do not have the same behaviour. In general, learning by experts methods more often give higher and more stable values for M_{opt} and M_{rand} and we therefore don't need to tune ϵ_{opt} or n^* as much. That's not the case for learning by self-practice because only some parameter settings lead to $M_{\text{opt}} = 0$. For example, for Deep Q-learning by self-practice, we only get $M_{\text{opt}} = 0$ with decreasing exploration and $n^* = 5000$. We also observe that M_{opt} is more stable for Q-learning than for Deep Q-learning.

If we now focus on M_{rand}, we observe that we obtain higher values for M_{rand} quicker using Deep Q-learning than for Q-learning. We also notice that for Deep Q-learning by self practice, M_{rand} tends to decrease once we reach the highest value, while it does not significantly decrease for the other methods.

We can conclude that with all these methods, the agent learns how to play *Tic Tac Toe*. However we have to keep in mind that there is a trade-off between the learning speed, the stability of the learning and the quality of the agent.