ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

ARTIFICIAL NEURAL NETWORKS MINI-PROJECT REPORT

TIC-TAC-TOE MINI-PROJECT

Castiglione Thomas, Seydou Fadel Mamar

12th September 2022

0.1 LEARNING FROM EXPERT

Question 1.

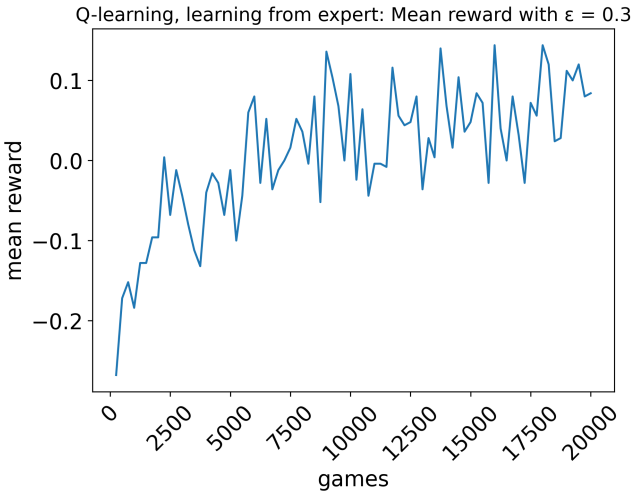

FIGURE 1

Average reward over time for $\epsilon = 0.3$. An overall increase in average reward is noticed over time with a maximum average reward slightly above 0.1. The average is computed for every 250 non-overlapping games.

Question 2.

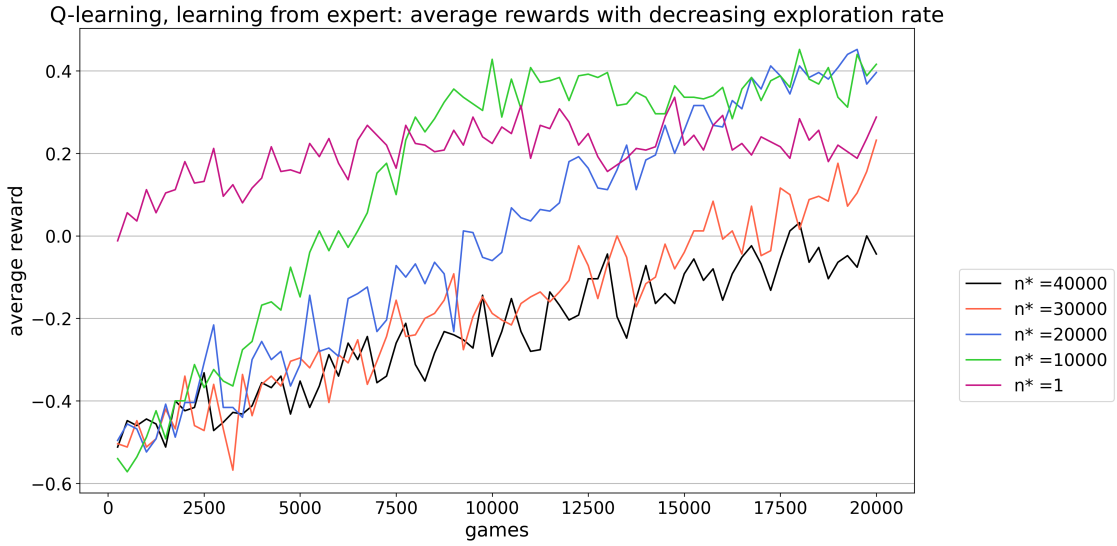

FIGURE 2

Average reward over time for varying n^* values. An overall increase in average reward can be observed for every n^* but with different maximum average rewards. The highest average rewards are reached for $n^* = 10'000$ where as the lowest are reached for $n^* = 40'000$. We notice that gradually decreasing ϵ (i.e. exploration) until it reaches 0.1 helps to learn much better than with a fixed ϵ (e.g. for $n^* = 1.0$). For $n^* = 10'000$ and $n^* = 20'000$, exploration rate decreases to reach 0.1 whereas it doesn't happen for $n^* = 30'000$ and $n^* = 40'000$.

Question 3.

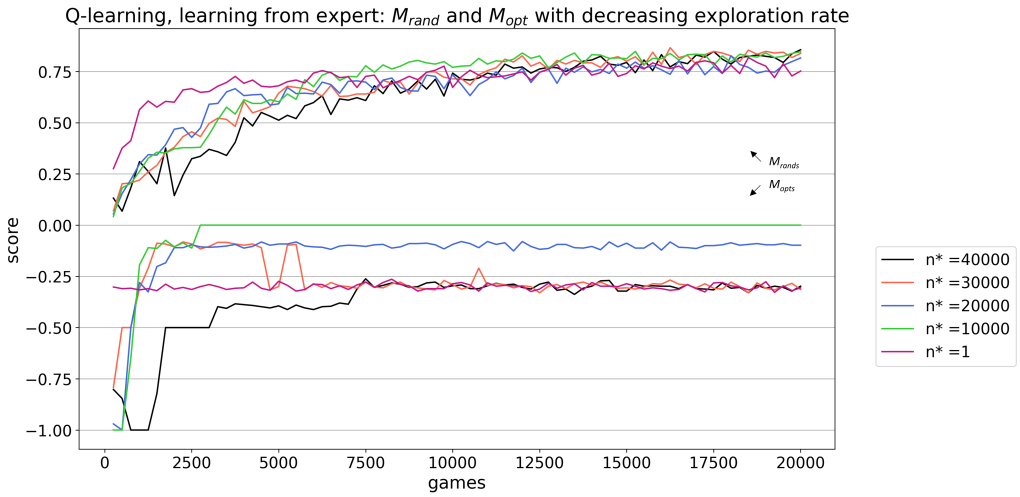

FIGURE 3

M_{opt} and M_{rand} over time for varying n^* values. There is an overall increase for both M_{opt} and M_{rand} for every n^*. $n^* = 10'000$ $(green)$ provides the best performance as we reach 0.85 for M_{rand} and 0 for M_{opt}. For other values of n^* we get good performance on M_{rand} but not on M_{opt}.

Question 4.

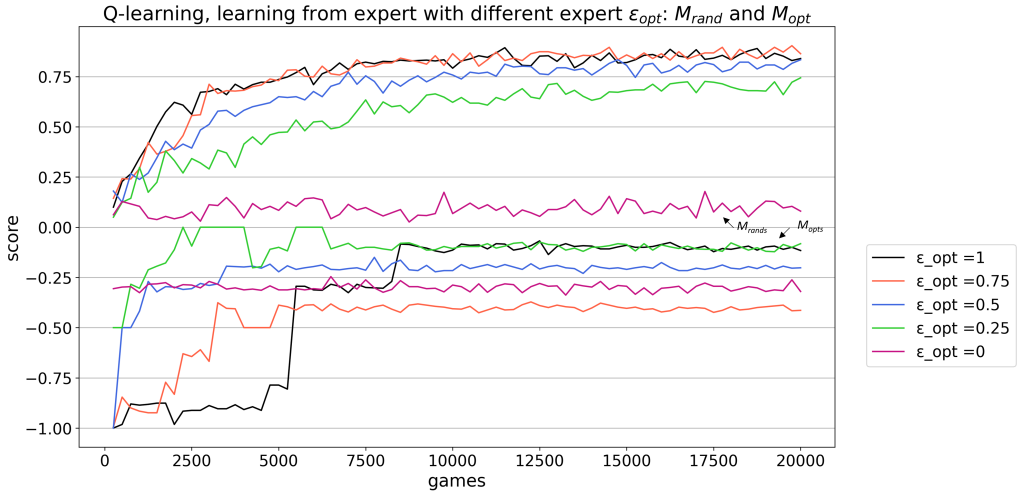

FIGURE 4

M_{opt} and M_{rand} over time for $n^* = 10'000$. Regarding M_{opt}, the best performance is achieved for $\epsilon_{opt} = 0.25$ and the worst for $\epsilon_{opt} = 0.75$. On the other hand, for M_{rand}, the best performance is achieved for $\epsilon_{opt} = 0.75$ closely followed by $\epsilon_{opt} = 0.1$ and the worst for $\epsilon_{opt} = 0.0$. We notice that when learning from the optima player, M_{opt} is almost constant and doesn't exceed -0.246. Additionally, the agent performs poorly against a random player (i.e. amateur). This behavior was unexpected as it seemed sensible to overfit the behavior of the optimal player. We can explain this by the high number of negative rewards that the agent receives when playing against the optimal. With more positive rewards (e.g. when playing against a semi-optimal) the agent might learn better. This hypothesis is sensible as the overall (i.e. on M_{rand} and M_{opt} best performing agent is obtained when playing against a random player.

Question 5. The highest values achieved are 0.904 for M_{rand} and 0.0 for M_{opt}. These values are derived from the results in *question 4*.

Question 6. In the setting where agent 1 would learn by playing against $Opt(0)$ and agent 2 would learn by playing against $Opt(1)$, they would have different Q-values. We can argue for that by looking at the plot of *Question 4* where see big differences in M_{opt} and M_{rand} for such agents. If they had the same Q-values, they would take similar actions which is obviously not happening in our experiments. The results we obtain provide a counter-example.

0.2 SELF-LEARNING

Question 7.

FIGURE 5

M_{opt} and M_{rand} over time for different exploration rates. Without exploration ($\epsilon = 0$), the agent get quickly stuck an cannot improve. The best performances in this setup was obtained with an exploration rate of 0.75, which is quite high. With $\epsilon = 0.98$, the agent learns well to beat a random player but does not spend enough time in exploitation mode to learn to beat an optimal player. Medium exploration rates perform relatively well but not as well as $\epsilon = 0.75$.

Question 8.

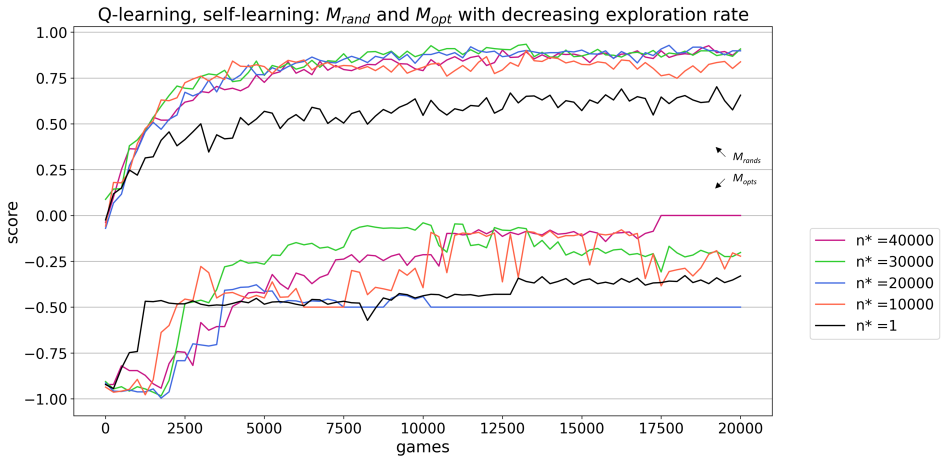

FIGURE 6

M_{opt} and M_{rand} over time for different speed of decreasing exploration rates. The best performances in this setup was obtained with very slow decreasing exploration rate ($n* = 40000$). With a medium exploration rate decreasing speed, the odds of beating a random player was similar to $n* = 40000$, but against optimal layer it performed poorly, especially with $n* = 20000$. With an exploration rate of 0.8 for the first game and 0.1 for all the others $n* = 1$, the agents do not spend enough time exploring and therefore do not perform well.

Question 9. The highest values achieved are 0.898 for M_{rand} and 0.0 for M_{opt}. These values are derived from the results in *question 8*.

Question 10.

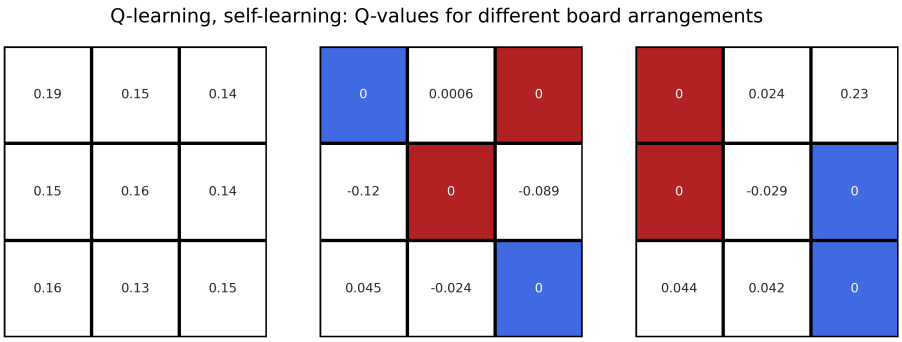

FIGURE 7

The different board arrangements are represented by the colors (blue=agent, red=opponent, white=free) and the annotations are the Q-values corresponding to each move. For the first board, which is the first move of the game, the highest Q-value is in the top left corner which is the best move (like any corner), showing that the rewards obtained at the end of the game are "back-propagated" to the start of the game with a training of 20'000 games. However the other Q-values are rather close. For the second board arrangement, The highest Q-value is by far in the bottom left corner which is the move that will block the opponent and win the game at the same time. This is a rather critical move the agent must learn to perform well. The last board is an arrangement where the agent must choose between blocking and winning. The highest Q-value is for the winning move, showing the agent learnt that if a move can win a game, he should choose it disregarding the opponent positions.

0.3 Deep Q-learning

0.3.1 Learning from expert

Question 11.

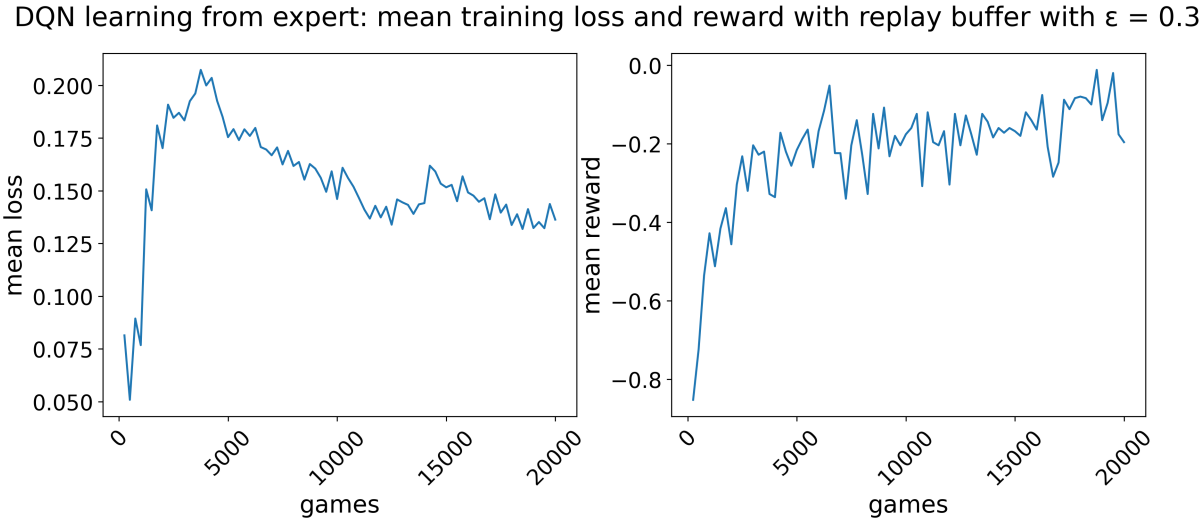

Figure 8
Average reward average training loss over time for $\epsilon = 0.3$ with replay buffer. The average loss increases rapidly at the beginning but then slowly decreases. The average reward increases non-monotonously (i.e. it decreases at times). The averages are computed for every 250 non-overlapping games.

Question 12.

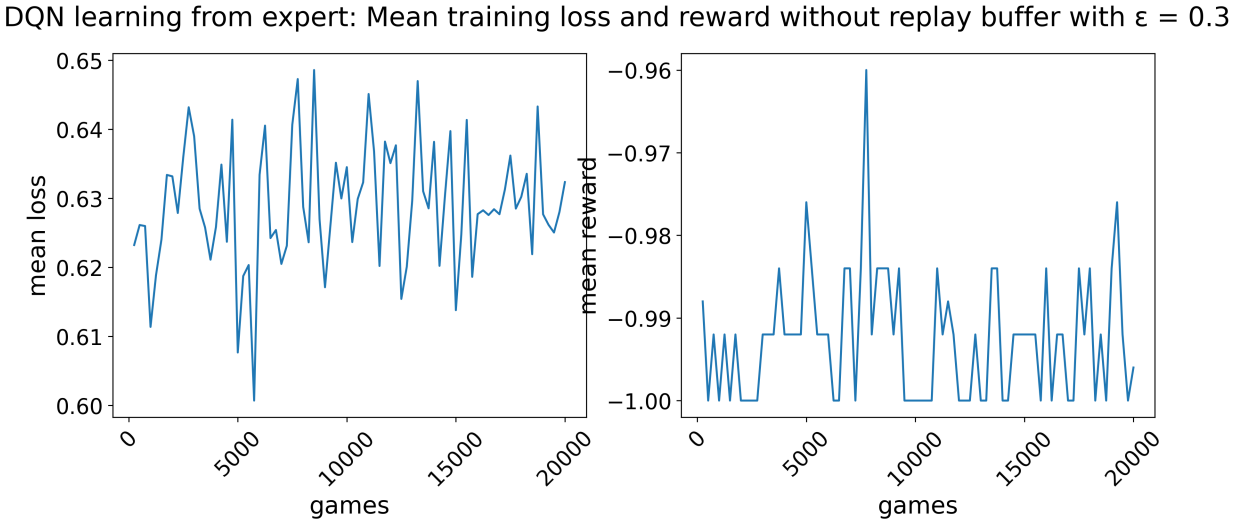

Figure 9
Average reward average training loss over time for $\epsilon = 0.3$ with online updates. The average loss and the average reward do not increase, they more or less "oscillates". We can see that with this optimization schemes, the network doesn't learn anything. In fact, the agent keeps making illegal actions.

Question 13.

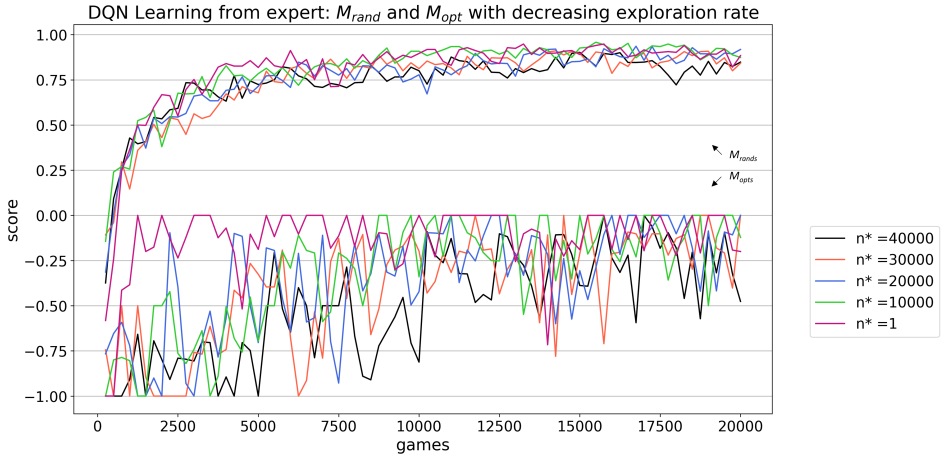

FIGURE 10

M_{opt} and M_{rand} over time for varying n^* while training with replay buffer. There is an overall increase for M_{rand} for every n^* with a maximum reached for $n^* = 10'000$ but closely followed by $n^* = 1$. For M_opt, the values are more difficult to visualize. With an exponential moving average filter, we can see that M_opt increase more rapidly for $n^* = 1$ whereas $n^* = 10'000$ is a little behind. The other two values of n^* achieve good performance on M_{rand} (i.e. above 0.8) but not on M_{opt}. In a nutshell $n^* = 1$ which keeps ϵ constant provides better performance in this setting of Deep Q-learning.

Question 14.

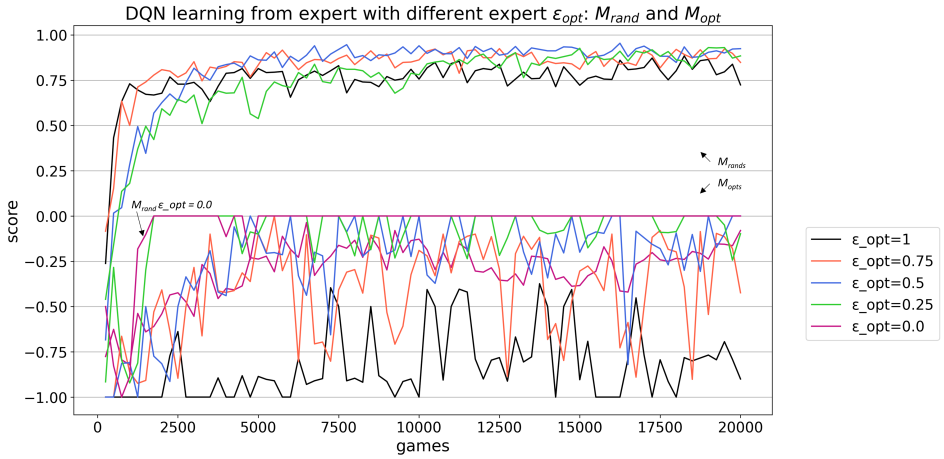

FIGURE 11

M_{opt} and M_{rand} over time for $n^* = 1$ while training with replay buffer. Using exponential moving average for M_opt values we can observe that the best performance is obtained for $\epsilon_{opt} = 0.0$ as we reach 0.0 more easily that with other values of ϵ_{opt}. The worst performance with regard to M_{opt} is obtained for $\epsilon_{opt} = 1$. Regarding M_{rand}, the best performance is observed for $\epsilon_{opt} = 0.5$ and the worst with $\epsilon_{opt} = 0.0$. Overall, the best model is obtained for $\epsilon_{opt} = 0.25$. It appears that when learning from an optimal player, the network overfits the observed behavior and becomes weak against a random player (i.e. amateur). Similarly, learning from a random player, the network overfits and becomes weak against an optimal player (i.e. expert). Introducing an $\in (0, 1)$ acts like a regularization and allows to learn better.

Question 15. The highest values achieved are 0.954 for M_{rand} and 0.0 for M_{opt}. These values are derived from the results in *question 14*.

0.3.2 SELF-LEARNING

Question 16.

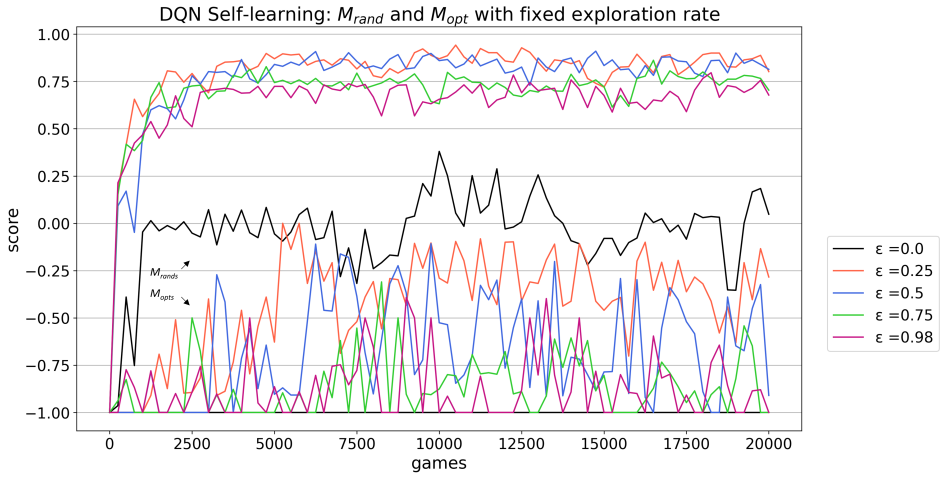

FIGURE 12

M_{opt} and M_{rand} over time for different exploration rates. Without exploration ($\epsilon = 0$), the agent does not learn to play Tic-Tac- Toe, it barely learns the legal moves. M_{opt} for $\epsilon = 0$ is not clearly visible in the figure but is equal to 0 at each training point. The best performance was obtained with $\epsilon = 0.25$, where M_{opt} reaches 0.0 twice. higher exploration rates did not performed well. Interestingly, the use of DQN led to much noisier M_{opt} scores than tabular Q-learning.

Question 17.

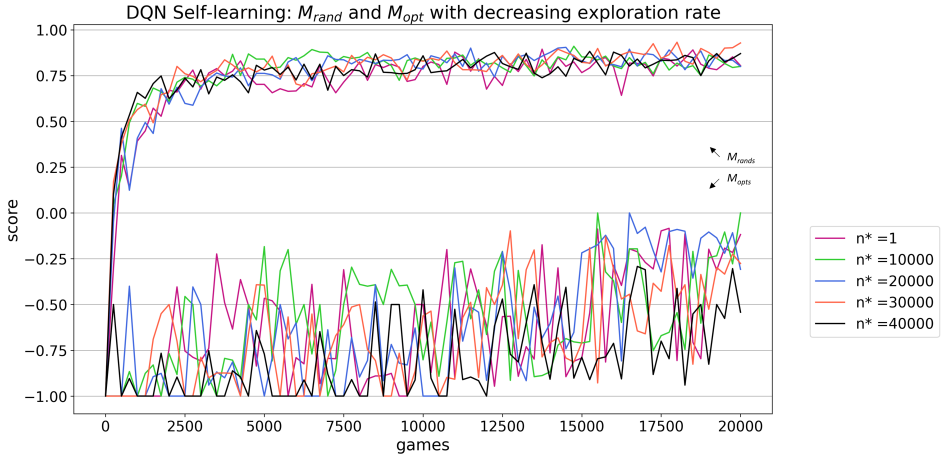

FIGURE 13

M_{opt} and M_{rand} over time for different decreasing exploration rates. The speed of decreasing exploration curiously did not had a significant impact on learning. However at the end of training, the best performances were obtained with $n^* = 20000$. Fixed exploration rates have more noisy M_{rand} scores, but the noise in M_{opt} score was still important for fixed and varying exploration rates.

Question 18. The highest scores achieved are 0.823 for M_{rand} and 0.0 for M_{opt}. These values are derived from the results in *question 17.* and obtained with a decreasing exploration rate of $n^* = 20000$. the low M_{rand} is explained by the fact that is was hard to end after exactly 20'000 games with the best agent in both M_{rand} and M_{opt}. The best overall scores was 0.942 for M_{rand} and 0.0 for M_{opt} and was obtained with a fixed exploration rate of 0.25.

Question 19.

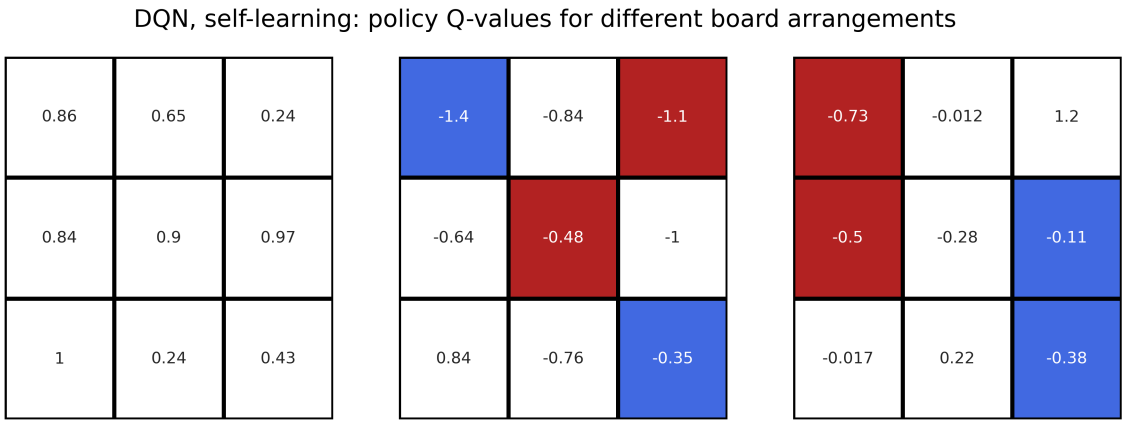

FIGURE 14

The different board arrangements are represented by the colors (blue=agent, red=opponent, white=free) and the annotations are the Q-values corresponding to each move. The boards are the same than *question 10.* For the first board, which is the first move of the game, the highest Q-value is in the bottom-left corner which is the best move (like any other corner). Interestingly, the bottom-right corner has a relatively low Q-value, meaning that the agent may not have learned the symmetries of the game. For the second board arrangement, The highest Q-value is by far in the bottom left corner which is the move that will block the opponent and win the game at the same time. This is a rather critical move the agent must learn to perform well. The last board is an arrangement where the agent must choose between blocking and winning. The highest Q-value is for the winning move, showing the agent learnt that if a move can win a game, he should choose it disregarding the opponent positions. It is worth mentioning that the agent learned well the illegal moves, having negative Q-values in all occupied spaces.

Question 20.

TABLE 1

Best performances for DQN and Q-learning. The training time the number of games it takes to reach 80% of the final performance.

	M_{opt}	M_{rand}	Training time [games]
DQN - From Expert	0.0	0.904	3750
Q-learning - From Expert	0.0	0.954	7500
DQN - Self-play	0.0	0.942	5500
Q-learning - Self-play	0.0	0.898	11000

Question 21.

Comparing the results between tabular Q-learning and DQN, one can straight-away say that both of them allowed agents to learn to play Tic Tac Toe efficiently, meaning beating a random player and not losing to an optimal player. The differences reside in the following:
First, the DQN resulted in sightly better performances for learning by self-practice, but at the cost of higher total training computation cost. When learning for expert, the tabular Q-learning performed better with a lower total training computational cost.
Secondly, the DQN showed a much higher noise in the M_{opt} score but not in the M_{rand} which is interesting. Knowing that M_{opt} is computed by playing against an optimal player, one could make the hypothesis that slight differences in the policy could result in a high M_{opt} score difference because when playing against an optimal player, making one mistake will very often result in a loss. Whereas, M_{rand} being computed by playing against

a random player, recovering from one mistake should be more often easier. But this does not really explain the difference between DQN and tabular Q-learning, only the difference between the score noises.