Miniproject 1: Tic-Tac-Toe

Giulia Mescolini (343142), Anna Peruso (343226)
CS-456 Artificial Neural Networks course project

I. Q-LEARNING

A. Learning from experts

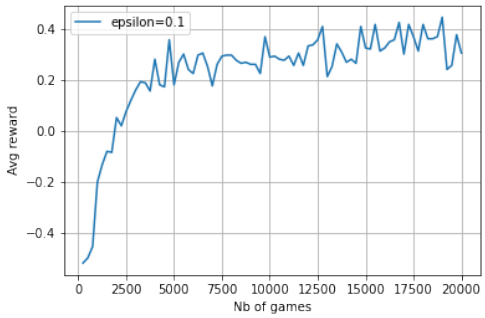


Fig. 1: Question 1.

Question 1. Figure 1 plots the average reward ($\varepsilon = 0.1$) computed every 250 games. As the number of games increases, so does the reward. However, after a steep increase at the beginning, the mean reward remains stable around 0.3. This is consistent with the fact that, even playing optimally, one cannot always win.

B. Decreasing exploration

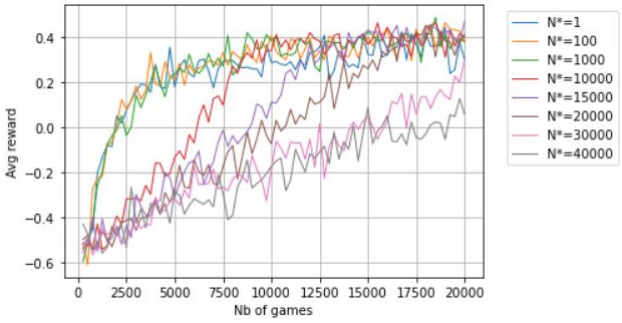
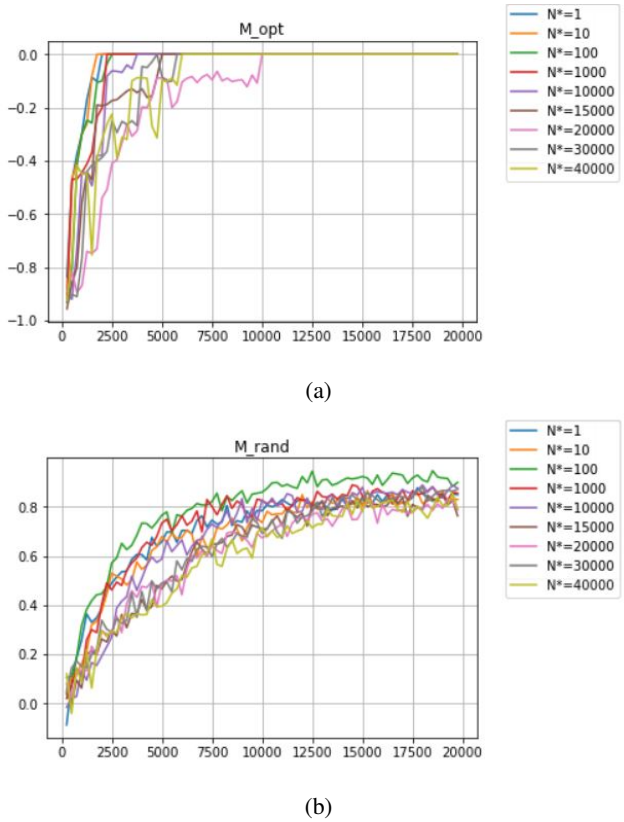


Fig. 2: Question 2.

Question 2. In the case of Figure 2, ε is not fixed but decreases over time. In particular, for $N^* = 40000$, ε decreases from 0.8 to 0.4, whereas in the opposite case of $N^* = 1$ we obtain $\varepsilon \equiv 0.1$ as in Figure 1. In general, a higher N^* has the effect of exploring more at the beginning of the learning, because the ε has bigger values, and also it takes more games to settle to ε_{min}. This property is proved by the fact that, at least initially, the average reward is lower for higher N^*. However, in the long run, for some chosen $N^* \geq 10$ we can have better performances compared to the results obtained with fixed $\varepsilon = 0.1$. For instance, considering the last 10000 games, the average reward for $N^* = 10000$ is 0.388, whereas for $\varepsilon = 0.1$ is 0.327.



(a)

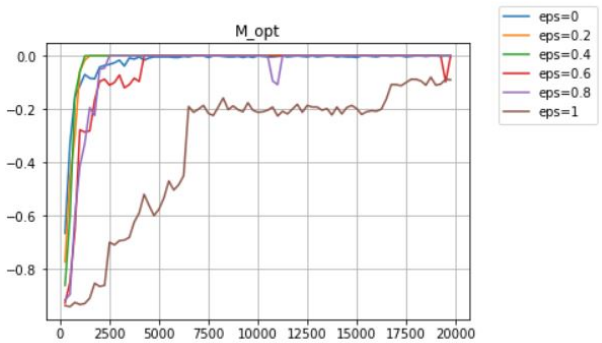
(b)

Fig. 3: Question 3.

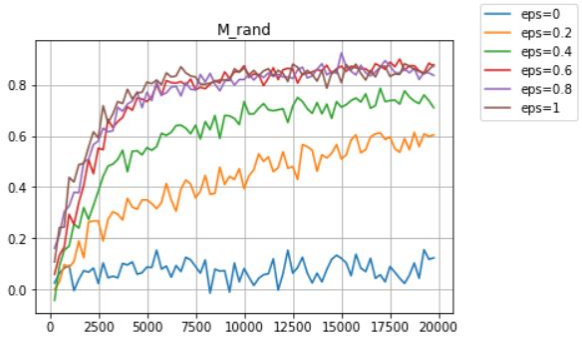
Question 3. As in the previous question, ε is not fixed over time. However, the performance is not tested during training but separately, with $\varepsilon = 0$ and against the optimal player (Figure 3a) and the random one (Figure 3b). In the former case, for all N^*, the agent learns to play. This is evident from the fact that the average reward always reaches 0 (so the tie), meaning the it learns to play optimally. Instead, in the latter case, an average reward of at least 0.8 is reached. Here, the best performance is obtained with $N^* = 100$, as it has the highest M_{rand}.

C. Good experts and bad experts

Question 4. N^* being fixed, during training we now vary ε_{opt}, *e.g.* $\varepsilon_{opt} = 1$ corresponds to the random player, while $\varepsilon_{opt} = 0$ is the optimal one. As before, the performance is

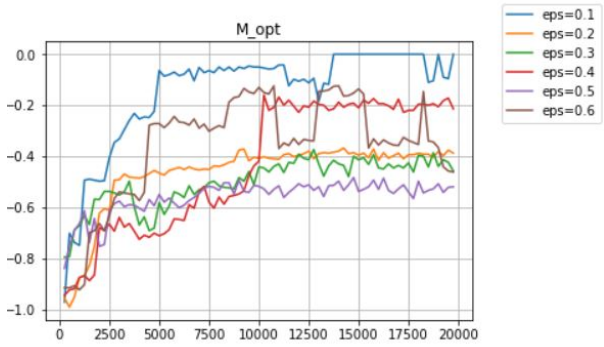


(a)

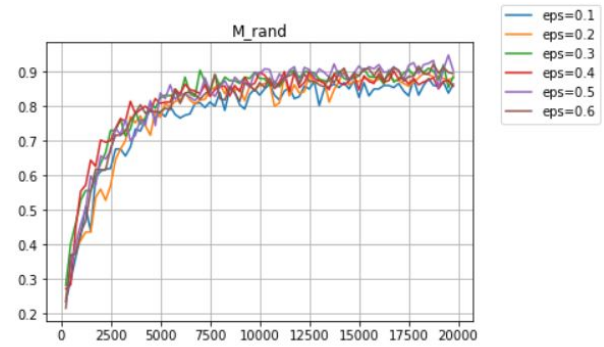


(b)

Fig. 4: Question 4.



(a)



(b)

Fig. 5: Question 7.

evaluated with 500 test games every 250 training games. As one should expect, when testing the agent against the optimal player, the worst performance is obtained when the training is done against the random player (Figure 4a). However, for all the other values of ε_{opt}, it can be stated that the agent learns to tie, which is the best outcome possible when playing against the optimal player. On the contrary, Figure 4b shows that choosing $\varepsilon_{opt} = 0$ does not lead to high rewards when testing against the random player. Indeed, during training, the agent learns the strategy to tie against the optimal player but not how to outperform a random one. As ε_{opt} increases, so does M_{rand}, with the best performances obtained by $\varepsilon_{opt} = 0.6, 0.8, 1$.

Question 5. The highest M_{opt} is 0, the peak of M_{rand} is 0.924, obtained with $\varepsilon = 0.8$.

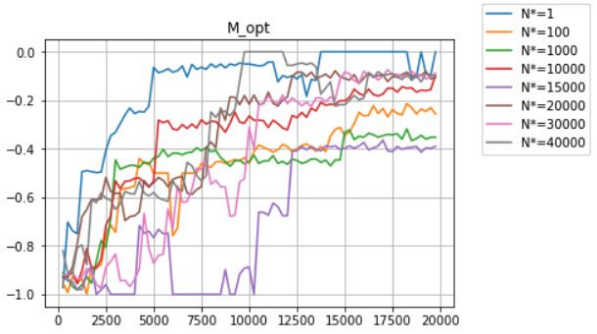
Question 6. When training against an optimal and a random player there are two main differences; on one hand, some states cannot be reached by the Agent 1, because the agent cannot find itself in a state where it has the possibility to win. On the other hand, all the Q-values of Agent 1 are necessarily non-positive, since the reward is always less or equal to 0. This is not the case for Agent 2, as it can get reward equal to 1.

D. Learning by self-practice

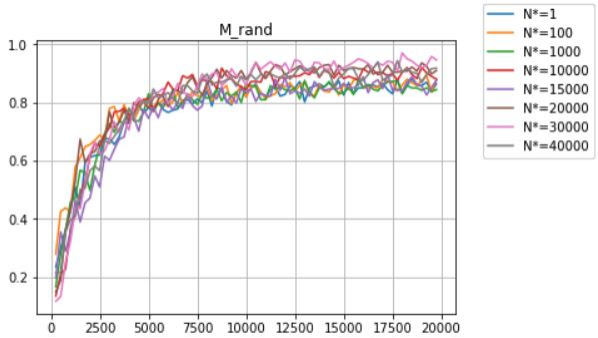
Question 7. Varying ε in the range $[0.1, 0.6]$, we note that the trend of M_{rand} is similar for all values (Figure 5a), while the trend of M_{opt} is strikingly different (Figure 5b). M_{opt} always grows with the number of training games, but it settles on different values for different ε, and the highest performance is achieved with $\varepsilon = 0.1$, which is the only case where the agent learns to tie against the optimal player. For the remaining cases, we note that the agent learns to tie at least in half of the games (being M_{opt} never below -0.5).

Question 8. The trend for M_{rand} does not show any remarkable difference with respect to **Question 7** (Figure 6b). Regarding M_{opt} (Figure 6a), although all M_{opt} are increasing over the number of played games and therefore the agent progressively learns to play, the best performance is obtained with $N^* = 1$, which corresponds to the case $\epsilon = 0.1$ of **Question 7**. Hence, contrary to learning-by-experts, in self-practice there is no evidence that beginning with higher exploration rate improves the learning. This may come from the fact that, with two self-trained agents, exploration is automatically higher.

Question 9. The highest M_{opt} is 0, while the maximum value of M_{rand} is 0.88, obtained with $N^* = 1$.



(a)



(b)

Fig. 6: Question 8.

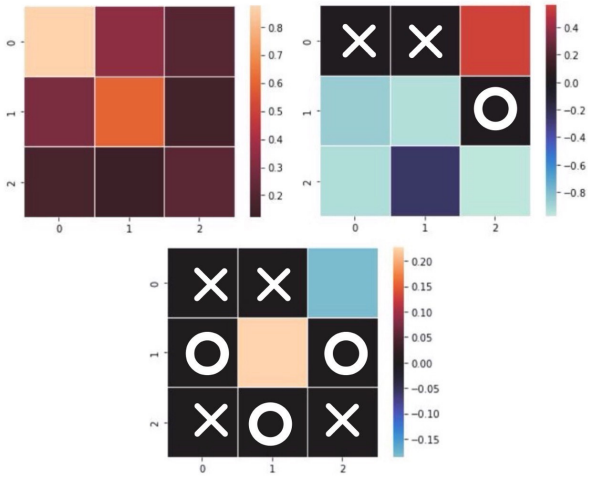


Fig. 7: Question 10.

Question 10. We report the Q-value table $Q(s, a)$ of states corresponding to three different degrees of occupation of the grid. The first heatmap corresponds to the empty board, and the most appealing positions appear to be the top-corner one and the central one; as known from the optimal strategy, the best starting moves are the central one and the corners, so on one hand the result is coherent with intuition, on the other hand we may have expected that all the four corners to be equally attractive, and this highlights a flaw of our agent. In the second heatmap, the highest Q-value corresponds to the right-corner move, which prevents the opponent (X) from winning. In the last example, the agent (here O) is one step away from winning and the Q-values suggest the right move.

II. Deep Q-Learning

A. Implementation details

With respect to the initial set-up proposed in the homework details, we modified the learning rate of the deep agent (both for learning from experts and self-learning); specifically, we started from a slightly higher value (10^{-3}) and halved it after each 2.500 episodes. This contributed to the reduction of strong oscillations, and hence led to better performances with respect to the case of fixed learning rate.

B. Learning from experts

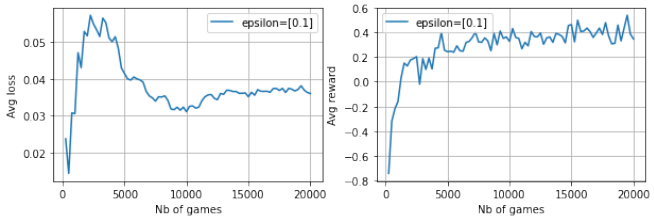


Fig. 8: Question 11.

Question 11. As shown by Figure 8 on the right, the average reward grows from -0.8 to around 0.4, hence the agent learns to play. Instead on the left, the average training loss, after an initial growth due to exploration of many invalid states, starts to decrease, as one should expect.

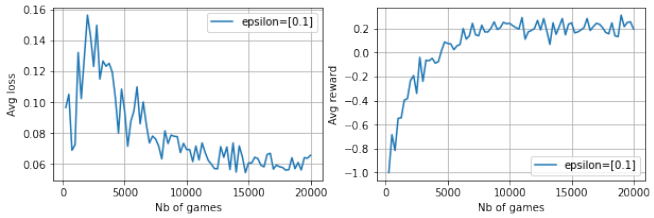
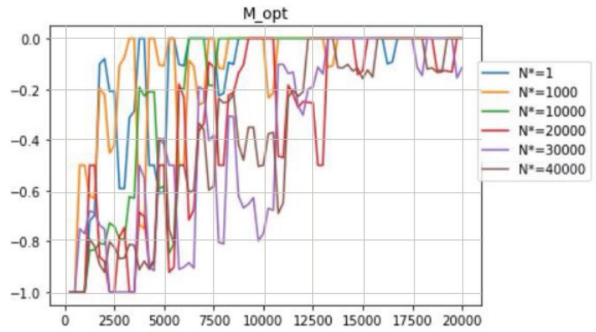


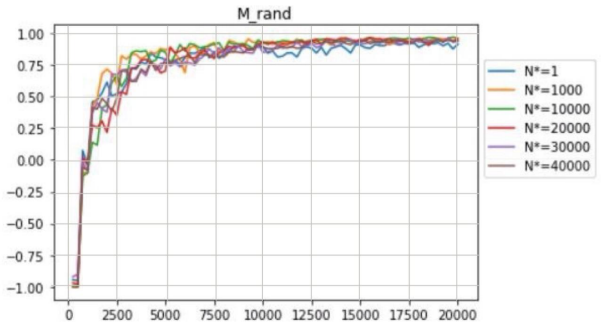
Fig. 9: Question 12.

Question 12. Similarly to the previous case, the average reward increases (Figure 9). However, it reaches a smaller value (around 0.2). This behaviour is also confirmed by an oscillating average training loss, which reaches a value of 0.06. In conclusion, using 1 as batch size leads to inaccurate estimates of the error gradient, in agreement with the theory.

Question 13. This is the counterpart of Figure 3 for the Deep Q-learning case. The aim is to investigate how the performances against the optimal and the random player (respectively Figure 10a and Figure 10b) vary when increasing the exploration rate at the beginning of the training. The average reward is computed on 500 tests every 250 training games. There is no striking difference for the test against the random

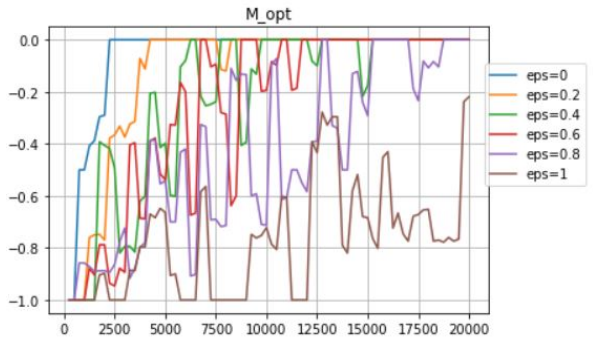


(a)

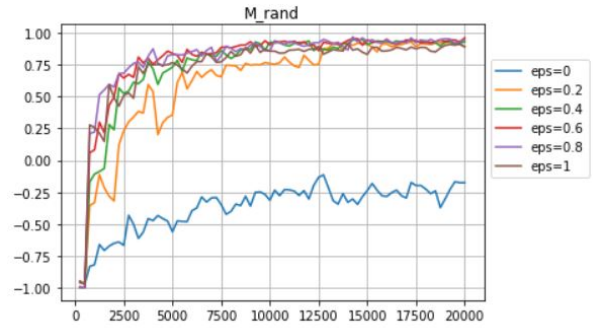


(b)

Fig. 10: Question 13.



(a)



(b)

Fig. 11: Question 14.

player ($\varepsilon_{opt} = 1$), as all the N^* yield an average reward of at least 0.8 after 20.000 games. For the test against the optimal player, we obtained a performance indicator above -0.2 for all values of N^* (less candidates have been tested with respect to **Question 3** due to the high computational cost). In the first half of the training, M_{opt} shows visible oscillations; this is due to a higher learning rate α which decreases over time, as described in the section *Implementation details*. However, after the first 10.000 games, fluctuations become significantly smaller and the average reward reaches 0, which is the upper bound on the reward when playing against `Opt(0)`. In particular, no oscillations are obtained for $N^* = 10.000$. For this reason, we chose $N^* = 10.000$ to address the next question.

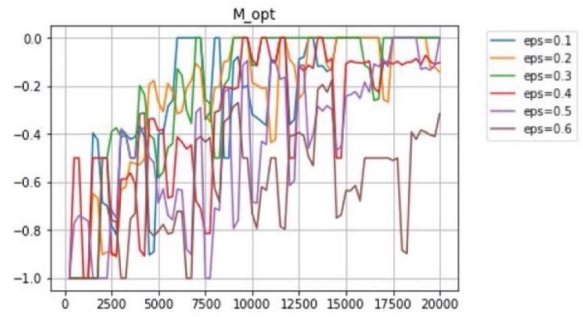
Question 14. In this case we vary ε_{opt} as in **Question 4** (Figure 4). The agent is trained to learn against different players, from the optimal ($\varepsilon_{opt} = 0$) to the completely random one ($\varepsilon_{opt} = 1$). As shown in Figure 11a, the best trend for M_{opt} is achieved when training against $\varepsilon_{opt} = 0$, which is the same player against whom the test is run. Specifically, it quickly settles around the optimal value 0. Also for $\varepsilon_{opt} = 0.2, 0.4, 0.6$ the agent learns to tie against the optimal player, even if it takes around 15.000 games to stabilize. Finally, when training against a random player ($\varepsilon_{opt} = 1$), instead, the agent does not learn to tie against the optimal player, but reaches an averaged test reward around -0.2.
Regarding M_{rand} (Figure 11b), the average reward reaches a value around 0.9 for all values of ε_{opt}, except $\varepsilon_{opt} = 0$. This is different from what happens for the tabular Q-learning, where the performance is highly dependent on ε_{opt}. However, for $\varepsilon_{opt} = 0$, M_{opt} cannot go higher than -0.2, meaning that in this case the agent mainly loses or ties against `Opt(1)`.

Question 15. The highest value of M_{opt} that can be obtained is 0, optimal value when testing against the optimal player, while for M_{rand} the peak is at 0.97, obtained when training with $\varepsilon_{opt} = 0.8$.

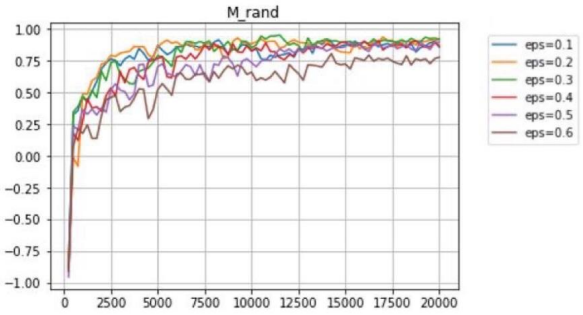
C. Learning by self-practice

Question 16. As for **Question 7**, the agent plays against itself for different fixed values of the exploration rate ε. By looking at Figure 12a and Figure 12b, we can observe that the trends of M_{opt} and M_{rand} are overall increasing, meaning that the agent learns despite some oscillations, reduced with the tuning of the learning rate above-mentioned. The highest value of ε ($\varepsilon = 0.6$) yields the worst performance both when tested against the the optimal and the random player; for the other values, the performances are fairly good, reaching a value of M_{opt} above -0.2 and of M_{rand} above 0.9.

Question 17. Instead of fixing a constant value for the exploration rate, in Figure 13 ε decreases over the number of games. While against the random player all choices of N^* lead to excellent performances (M_{rand} is always above

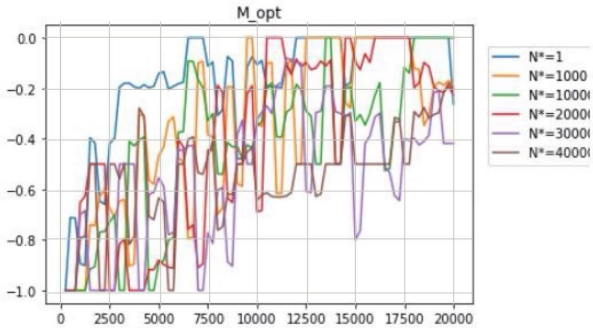


(a)

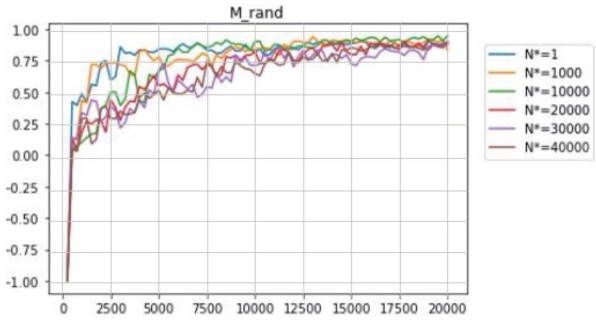


(b)

Fig. 12: Question 16.



(a)



(b)

Fig. 13: Question 17.

0.8), the same cannot be stated for the test against the optimal player. For high values of N^*, indeed, the agent does not learn to tie against the optimal player, and the best performance is achieved with $N^* = 10.000$, which reaches the stable value of $M_{opt} = 0$.

Question 18. The highest value of M_{opt} that can be reached is 0, while the best value of M_{rand} is 0.918. Both values are obtained with $N^* = 10.000$.

Question 19. In Figure 14, we report three different boards in which player X has to perform its next move. First, it is worth noticing that even the Q-values of unavailable moves are non-zero, contrary to what happens in the tabular case. In fact, the agent is not constrained to select only empty positions.
In the first picture, the most appealing move is the bottom-left corner, consistently with the optimal strategy; however symmetry is here lost, since the upper-right corner, which would lead to the same outcome, has not the same Q-value. In the picture in the middle, the highest Q-value corresponds to the action that leads the player X to the victory; additionally, we can observe that the other action with non-negative Q-value is the one that blocks the victory of player O. The same situation is witnessed in the picture on the right.

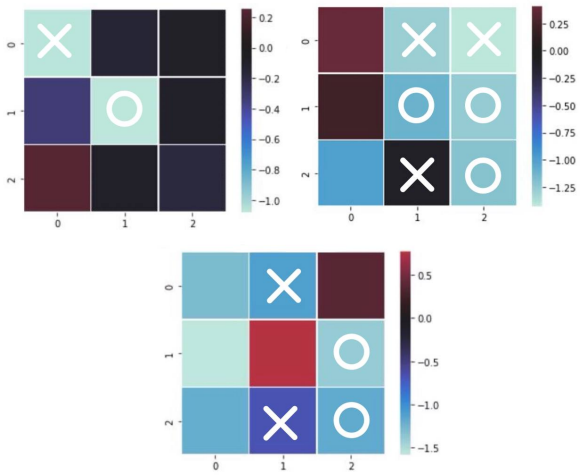


Fig. 14: Question 19.

III. Comparing Q-learning with Deep Q-learning

Question 20. The reported indicators have been computed as a mean of their values on the last 10.000 games.

Question 21. In conclusion, thanks to the carried analysis, it can be stated that both the tabular agent and the deep agent learn to play, with overall similar performances. However, we should stress the fact that Deep Q-learning could still be improved with a better tuning of the parameters, (*i.e.* learning rate,
batch size, *etc...*). Going more in depth, some differences can be highlighted. For instance, with DQ-Learning the choice

	M_{rand}	M_{opt}	T_{train}
Q-Learning from experts	0.856	0.0	4750
Self Q-Learning	0.851	-0.044	7250
DQ-Learning from experts	0.929	-0.014	6750
Self DQ-Learning	0.866	-0.077	17.500

TABLE I: Question 20.

of ε_{opt} has a milder influence on the outcome of the training. Focusing on Figure 4 and Figure 11, we can observe that the DQ agent learns to play against a random player even when trained against small (but positive) ε_{opt}, such as $\varepsilon_{opt} = 0.2, 0.4$, contrary to what happens for the tabular case. For M_{opt}, instead, we have similar trends for all ε_{opt}.
In both tabular and Deep Q-learning, the less satisfying performances are obtained when the self-practice agent plays against the optimal player (see Figure 5 and Figure 12). However, also in this case, the DQ algorithm achieves better results, as M_{opt} is greater than -0.2 for all the exploration rates ϵ. Finally, DQ-Learning often ensures a definitely faster learning.
That being said, the main drawback of the DQ-Learning is the huge computational effort compared to the tabular Q-Learning, that already reaches fairly good performances (see Table I). Considering that DQ-Learning takes 5 times the tabular Q-Learning, it may sometime be worth rely on the latter.