EPFL Artificial Neural Network Miniproject Report

by Hao Zhao, by Yixuan Xu

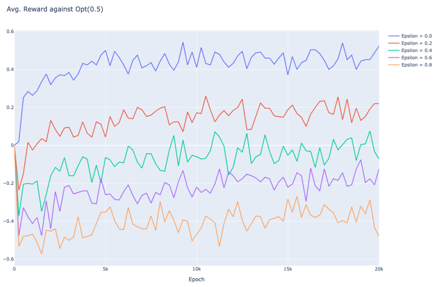


Figure 1: (Question 1) Our agent learns to play tic-toa-toe. Our selection for ϵ is [0.0, 0.2, 0.4, 0.6, 0.8], although there are fluctuations for average reward against OPT(0.5), we can observe obvious upward trend, especially when the agent chooses action fully based on Q-learning algorithm ($\epsilon = 0.0$).

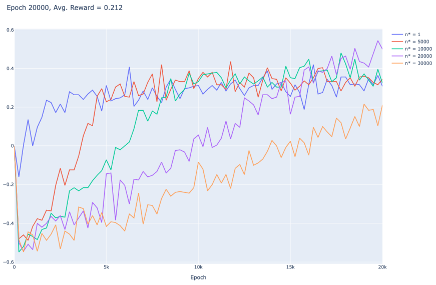


Figure 2: (Question 2) n^* does help training comparing to have a fixed ϵ. As can be seen from Fig.1, the highest average reward we obtained is just below 0.6, and it varies dramatically from 0.6 to -0.4 given different ϵ value. Whereas in Fig.2, even though the ϵ changes in a continuous manner with n^*, the average rewards are all above 0 after 15,000 epochs. To be specific, with n^* in [1, 5000, 10000], average reward ends up with around 0.4, and the best average reward 0.6 is obtained when $n^* = 20000$. The effect of n^* is to decrease ϵ continuously from 0.8 to 0.1 while playing games, the higher the n^*, the slower the decreasing.

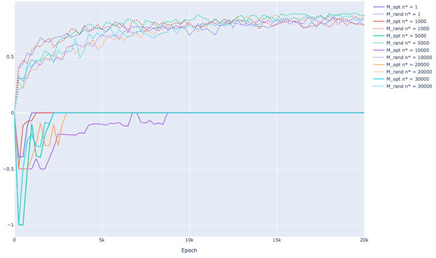


Figure 3: (Question 3) M_opt measures the relative win rate against a 100% optimal player and M_{rand} is the performance against unseen agents. It shows the agent's generalized performance during the learning process against a optimal player with a fixed ϵ. It complements the evaluation of agent's behaviour. All three metrics objectively show the agent's ability to play tic-tac-toe and increase during the training process.

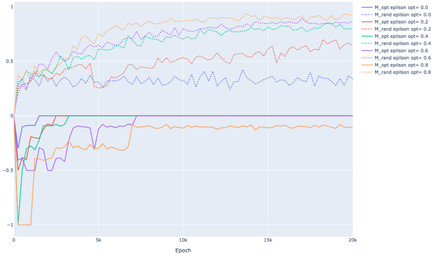


Figure 4: (Question 4) M_{opt} measures the relative win rate against a 100% optimal player and M_{rand} is the performance against unseen agents. It shows the agent's generalized performance during the learning process against a optimal player with a fixed ϵ. It complements the evaluation of agent's behaviour. All three metrics objectively show the agent's ability to play tic-tac-toe and increase during the training process.

Question 5 The highest value of M_{opt} is 0. A Q-learning agent with $n^* = 20000$ can achieve this value when playing against OPT with any ϵ_{opt} in [0, 0.2, 0.4, 0.6]. The highest value of M_{rand} is 0.936, which is obtained in epoch 13,750 when Q-learning agent plays against OPT(0.8).

Question 6 $Q_1(s, a)$ and $Q_2(s, a)$ do not have the same values. In our project, the given optimal player plays according to hard-coded rules from routine experiences. Our agent can almost perfectly learn this strategy by playing against Opt(0). Some states against this rule are never achieved in this process. However, if playing against a fully random player, our agent can explore rich states and games but it cannot learn a rule like before. So we think $Q_1(s, a)$ and $Q_2(s, a)$ do not have the same values.

Figure 5: (Question 7) As depicted in Fig.5, the curves of M_{rand} and M_{opt} both present an upward trend, and end up with M_{rand} of nearly 1 against random player and M_{opt} of 0, thus we can conclude the agent learns to play tic-toa-toe. ϵ controls the degree of exploration, and is crucial since two agents share the same q-table, we should leave room for mistakes of one agent so the other one could learn from it. When $\epsilon = 0$, in which case both M_{rand} and M_{opt} does not learn

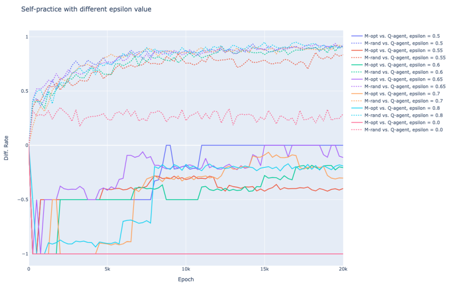
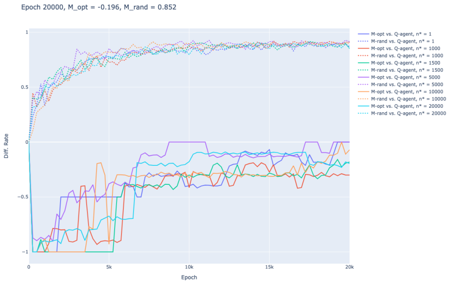


Figure 6: (Question 8) The strategy of decreasing ϵ helps training compared to having a fixed ϵ. In the decreasing ϵ case, the agent learns to play tic-tac-toe with an initial $\epsilon = 0.8$ and ends with $\epsilon = 0.1$. Compared with 5, we can see that decreasing ϵ helps to achieve significantly higher M_{rand} performance. And it also helps to improve its performance when playing against optimal players according to the M_{opt} curve. The fluctuation of M_{opt} is significantly reduced and our agent achieves higher M_{opt} value.



Question 9 After playing 20000 games, the highest value of M_{opt} is 0 and the highest value of M_{rand} is 0.95.

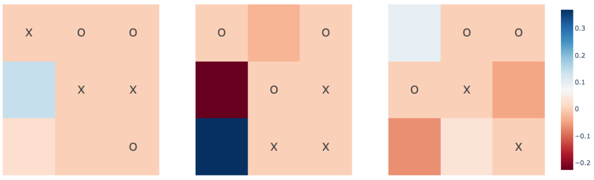


Figure 7: (Question 10) 'X' is the piece of Q-learning agent and 'O' is the piece of opponent, all the boards are the state at which point Q-learning agent is going to play. As can be seen from the first board, the q-value clearly indicates which action our player should take, and the action indicated by blue is the winning position. On the second board, the action is extremely reasonable as the agent would win if it takes the blue action, and it would lose the game if it takes the red one. On the third board, the position indicated by blue also leads the agent to the winning position, on the meanwhile blocks the winning of opponent. Therefore, it is sufficient to say the agent plays well.

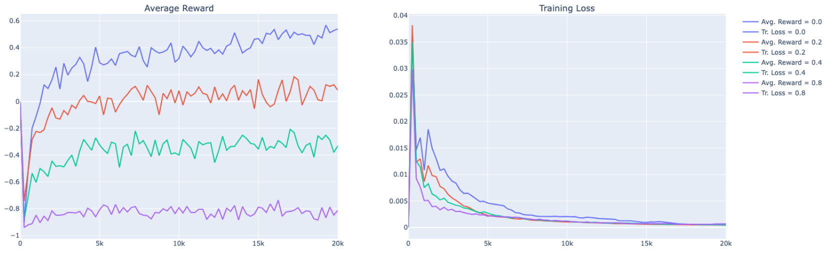


Figure 8: (Question 11) We choose a list of ϵ $[0.0, 0.2, 0.4, 0.8]$ and make our DQN agent compete with a optimal player with a fixed ϵ_{opt}. The training loss keeps decreasing. And we can clearly see that our agent learns to play tic-tac-toe from the average reward curve.

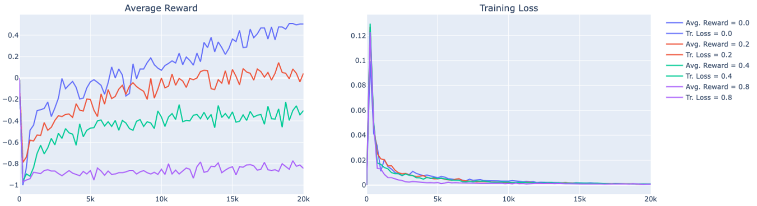


Figure 9: (Question 12) Without the replay buffer and with a batch size of 1, our agent learns slower than an agent with a large replay buffer to provide training samples. Also, the average training loss is higher than the previous case. But the agent can still learn how to play tic-tac-toe.

Figure 10: (Question 13) The strategy of decreasing ϵ helps training compared to have a fixed ϵ. The epsilon-greedy and decaying-epsilon-greedy algorithms both help our agent converge to the optimal policy. However, the epsilon greedy algorithm continues to pay the price of exploration, and therefore never catches up to the performance of the decaying-epsilon-greedy algorithm. We choose several values of n^* from a reasonably wide interval between 1 to 40000. n^* represents the decay velocity of epsilon. A larger n^* leads to a slower epsilon decay. And the epsilon level keeps the same after the agent completes n^* epochs. A slow decay strategy (e.g., $n^* = 10000$) is not preferred which means it may suffer more from exploration. And $n^* = 1000$ slightly outperforms the other cases, which means a smaller n^* is not always a better choice. For $n^* = 1$, it does not explore well in the beginning, so it shows larger fluctuations in the whole training process.

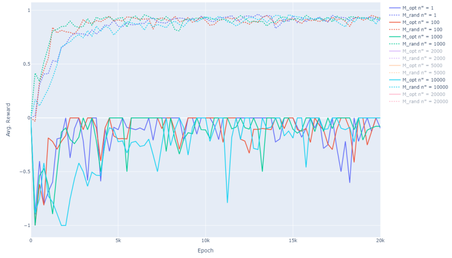
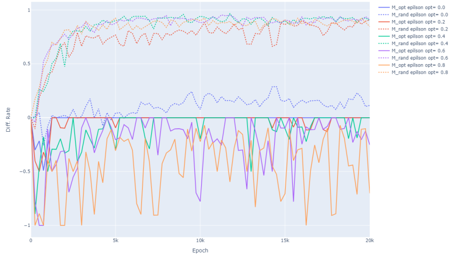


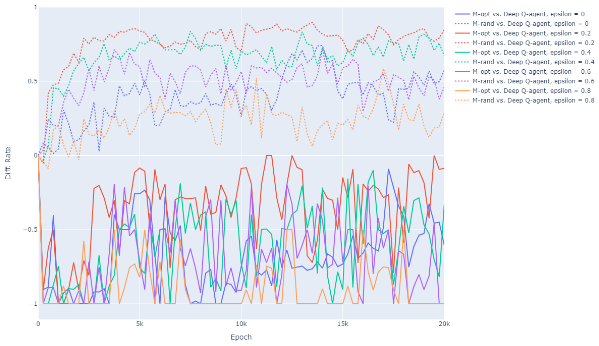
Figure 11: (Question 14) We choose a list of ϵ_{opt} $[0.0, 0.2, 0.4, 0.6, 0.8]$ and make our DQN agent compete with a optimal player with a fixed ϵ_{opt}. When $\epsilon_{opt} = 0$, our agent learns by playing against a 100% optimal player. M_{opt} increases fast and reaches to 0 after around 1000 games and keeps steady since then, which means it can quickly master the optimal player's experience of playing tic-tac-toe. However, it does not show big advantage over a player with random move. I think the reason is that some board states are never shown in the games against the optimal player and our policy model cannot learn to approximate the Q values for these states. Agents learn with $\epsilon \neq 0$ finally achieve similar M_{rand} performance. In terms of M_{opt}, the introduce of non-zero ϵ shows no benefits of learning optimal policy from the optimal player.



Question 15

After playing 20000 games, the highest value of M_{opt} is 0 and the highest value of M_{rand} is 0.966 with $\epsilon_{opt} = 0.4$.

Figure 12: (Question 16) The agent learns to play tic-tac-toe by playing against itself. We choose a list of ϵ $[0.0, 0.2, 0.4, 0.6, 0.8]$ and make our DQN agent with different levels of randomness compete with itself. When $\epsilon = 0.2$, the maximum M_rand reaches to 0.938 and maximum M_{opt} is up to 0. A higher ϵ allows the agent to explore more and learns better policy. But a too large ϵ also fails to learn a good policy. An agent with $\epsilon = 0$ is easy to be stuck in a local optimal point.



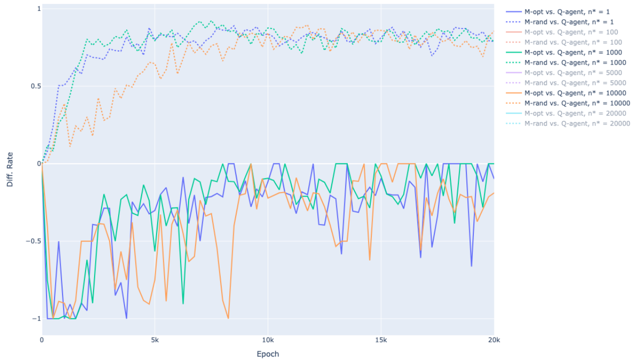


Figure 13: (Question 17) The strategy of decreasing ϵ helps training compared to having a fixed ϵ. In the decreasing ϵ case, the agent learns to play tic-tac-toe with an initial $\epsilon = 0.8$ and ends with $\epsilon = 0.1$. Compared with 12, we can see that decreasing ϵ helps to achieve significantly higher M_{rand} performance. And it also helps to improve its performance when playing against optimal players according to the M_{opt} curve. The fluctuation of M_{opt} is significantly reduced and our agent achieves higher M_{opt} value.

Question 18

After playing 20000 games, the highest value of M_{opt} is 0 and the highest value of M_{rand} is 0.924 with $n^* = 1000$.

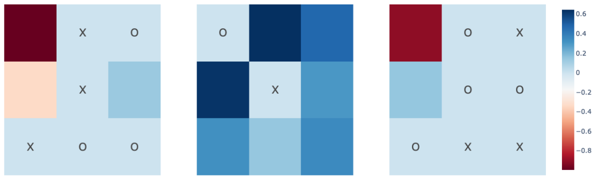


Figure 14: (Question 19) 'X' is the piece of Deep Q-learning agent and 'O' is the piece of opponent. As can be seen from the first board, the q-value clearly indicates which action our player should take, and the action indicated by blue on position (0,2) could block the opponent from winning. On the second board, the game just starts so no position with negative q-value, however, the two dark blue position could also block opponent from future winning. On the third board, the position (1,0) indicated by blue also prevents the opponent from winning. This is different from the board arrangements in 7, where the winning is overwhelming. This is because the model trained via self-practising, thus, the opponent is also fine-tuned, any dark red action would lead to winning from the opponent vision.

Question 20

	Q-Learning: Experts	DQN: Experts	Q-Learning: Self-practice	DQN: Self-practice
M_{opt}	0	0	0	0
M_{rand}	0.906	0.966	0.95	0.924
T_{train}	4500	1500	4000	2250

Table 1: Best performance (the highest M_{opt} and M_{rand}) of Q-Learning and DQN (both for learning from experts and for learning by self-practice) and their corresponding training time.

Question 21 In terms of M_{opt} value, agents in 4 different scenarios all reach 0, which shows they all learn how to play with the optimal player but in different ways. As for the M_{rand} metric, agents in 4 different scenarios all perform well. Among them, DQN for learning by self-practice outperforms the other 3 scenarios. It shows DQN has better ability to learn. Among all scenarios, DQN for learning from experts took the shortest time to reach to 80% of its final performance, which demonstrates its stronger learning ability.