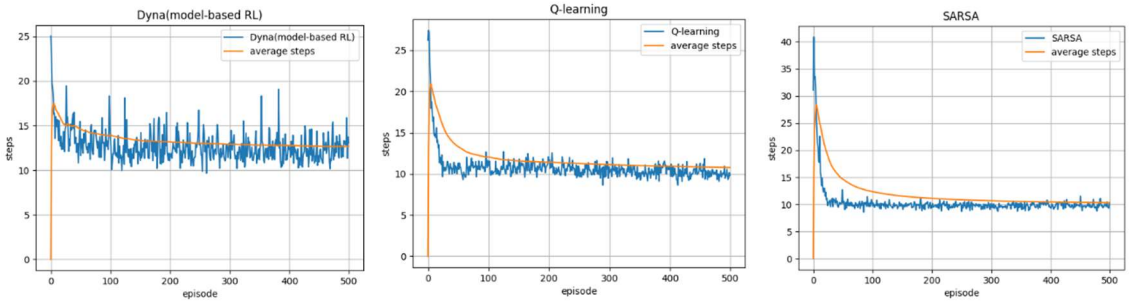
Grid World assignment

Student ID : 2016145015

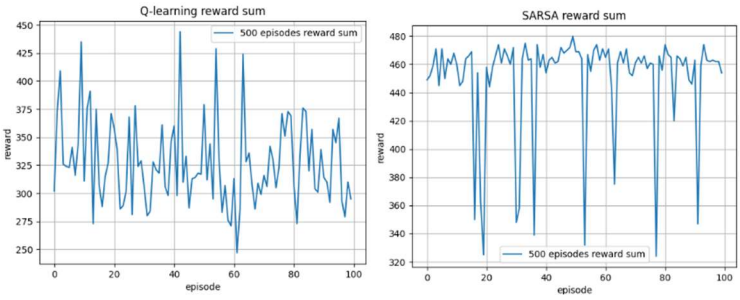
Name : DooHyun Lee

1. Compare the Value-iteration and the following algorithms when applied to the grid world in terms of the data-efficiency

	Average Steps(start → goal)	Average converge time	Average converge iterations
Value iteration	-	0.04284	583
Model-based RL(Dyna)	12.39	0.37783	117.33
Q-learning	9.986	0.07077	159.78
SARSA	10.285	0.1032	215.26



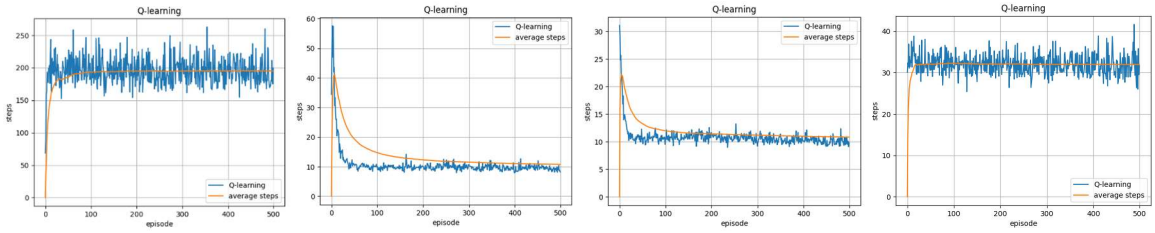
- 먼저, Value-iteration과 다른 algorithms를 비교하기 위해서 tolerance를 0.01로 맞추고 converge를 얼마나 빠르게 하는지를 통해서 data efficiency를 비교해봤습니다. Value-iteration의 경우에는 Bellman Equation에서 optimal value function을 iteration을 통해서 찾는 방식인데 이 grid world는 크기가 작다보니 converge하는 속도가 다른 알고리즘에 비해서 빠른 것을 알 수 있지만, environment가 더 커지거나 복잡해지는 경우 훨씬 느린 속도를 보일 것으로 예측됩니다. 그래서 converge time으로 비교해봤을 때, Value-iteration → Q-learning → Model-based RL(Dyna) → SARSA 순으로 좋은 것을 확인했습니다. 또한, Model-based RL의 경우 experience에서 sampling해서 학습을 하므로 더 적은 iteration으로도 학습이 가능하지만, 각 step의 속도는 매번 iteration을 통해 연산을 해야하므로 느린 것을 알 수 있었습니다. Q-learning과 SARSA의 경우 거의 비슷한 것을 알 수 있고, SARSA의 경우에는 Q-learning보다 살짝 느린 것을 알 수 있는데 이것은 on policy의 영향인 것으로 생각이 됩니다. 하지만 밑에 그래프를 보면 Q-learning의 경우 살짝 더 빠르게 결과를 얻을 수 있지만 policy에 stochastic한 부분이 담겨있으면 – reward를 주는 곳에 갈 가능성이 높아 reward sum이 전반적으로 SARSA에 비해 낮게 나온 것을 확인할 수 있었습니다. SARSA의 경우에는 전체적인 Q값을 보고 update하므로 Q-learning에 비해 안정적일 가능성이 높아 reward sum에서 Q-learning에 비해 전반적으로 높게 나온 것을 확인할 수 있었습니다.



2. Comparing results with and without the Epsilon-greedy policy

→ Q-learning

	Average Steps	Average converge time	Average converge iteration
Epsilon = 0	195.021	0.2688	45.21
Epsilon = 0.1	10.73	0.07992	191.14
Epsilon = 0.3	10.84	0.06919	163.56
Epsilon = 0.9	31.96	0.07711	55.64



(Epsilon = 0) (Epsilon = 0.1) (Epsilon = 0.3) (Epsilon = 0.9)

Epsilon = 0 이게 되면 exploration을 전혀 하지 않기 때문에 초반에 결정되는 Q값에만 영향을 받아 최적 경로를 찾지 못하게 된다. 따라서 average steps이 195로 매우 큰 것을 알 수 있습니다. 또한, Epsilon이 매우 크게 되면 거의 random action과 같아져서 average steps가 커지는 것을 알 수 있고, 따라서 적당한 값으로 Epsilon을 정해야 최적 경로를 잘 탐색할 수 있음을 알 수 있었습니다.