세미나

2019305050 이제희

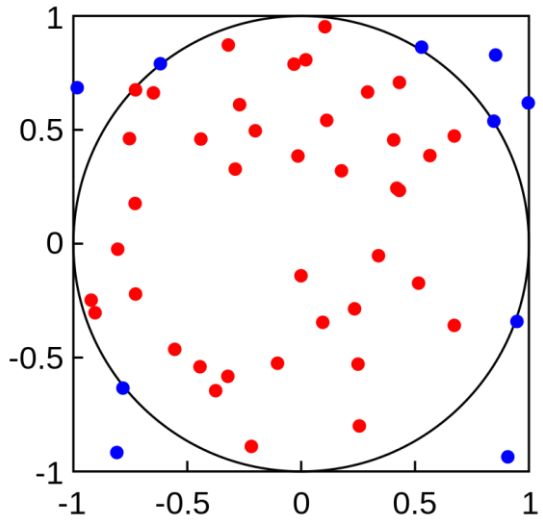
목차

1. 몬테카를로 예측

2. 시간차 예측

3. 성능 비교

몬테카를로 예측



실제 경험을 통해 참 가치함수의 값을 추정

에피소드 단위로 가치함수를 업데이트

-> sampling

$$\boldsymbol{V(s) \leftarrow V(s) + \alpha(G(s) - V(s))}$$

α : step size (0 ~ 1)

시간차 예측(SARSA, q-learning)

Time step 단위로 가치함수를 업데이트

● SARSA

ε – greedy 정책에 따라 행동을 선택

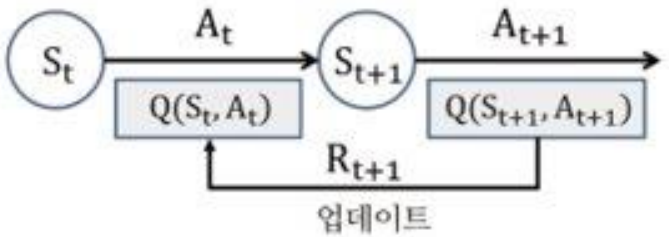
$$[\boldsymbol{S_t}, \boldsymbol{A_t}, \boldsymbol{R_{t+1}}, \boldsymbol{S_{t+1}}, \boldsymbol{A_{t+1}}]$$

-> 상태 S_t에서 ε – greedy 정책에 따라 행동 A_t을 선택
-> 보상 R_{t+1}가 나오고 다음 상태 S_{t+1}으로 바뀜
-> 정책에 따라 행동 A_{t+1}을 선택

- Q(s, a) 함수 업데이트(학습)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))$$

SARSA는 **on-policy** 제어
->탐색하는 정책 = 예측하는 정책 (ε – greedy)



ε – greedy 정책
ε : 0 ~ 1
일정한 확률 ε은 random으로 선택
1- ε 의 확률로는 greedy한 결과 선택

시간차 예측(SARSA, q-learning)

● q-learning

$[\boldsymbol{S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}}]$

-> 상태 S_t에서 ε – greedy 정책에 따라 행동 A_t을 선택
-> 보상 R_{t+1}가 나오고 다음 상태 S_{t+1}으로 바뀜

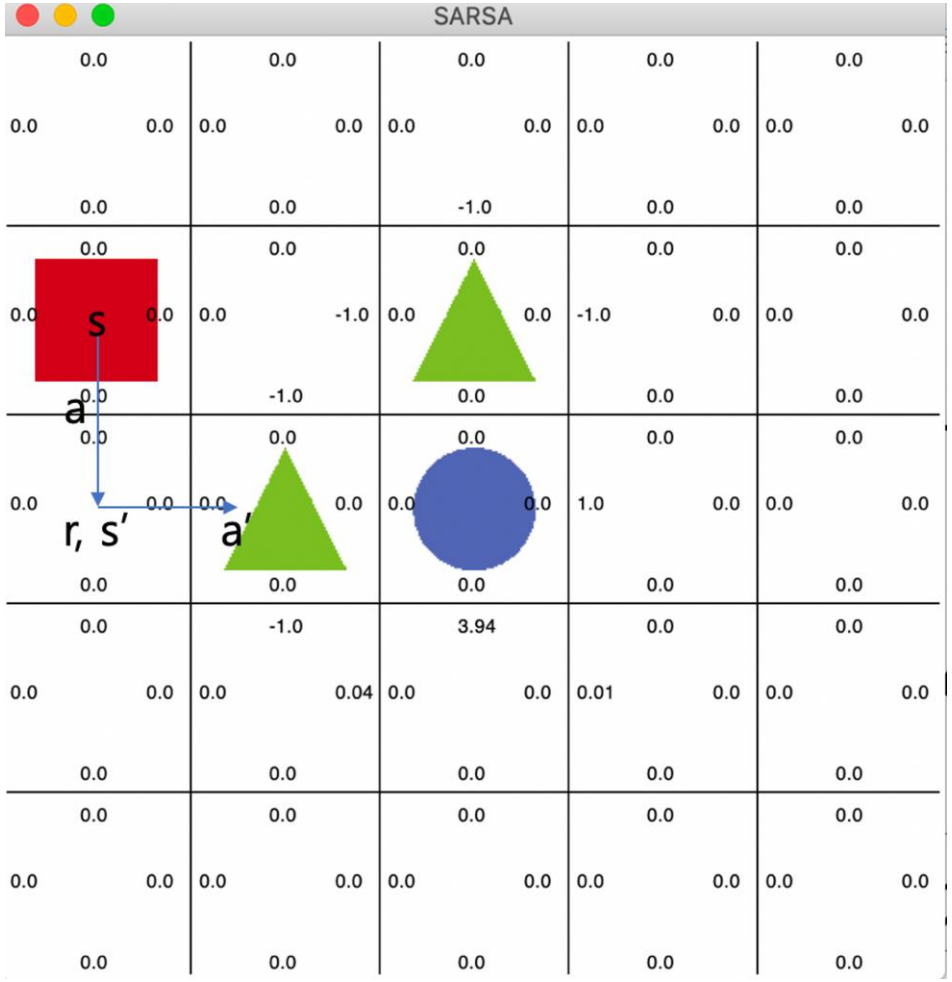
-> 정책에 따라 행동 A_{t+1}을 선택 (SARSA)
-> 가장 큰 Q 함수의 행동을 선택 (q-learning)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max Q(s_{t+1}, a_{t+1}) - Q(S_t, A_t))$$

q-learning은 **off-policy based**한 방법
->탐색하는 정책 (ε – greedy) ≠ 예측하는 정책 (greedy)

(SARSA의 on-policy정책은 잘못된 예측 학습 문제가 발생할 수 있음)

성능비교(monte-carlo, SARSA, q-learning)



좌측 맨 위에서 시작

삼각형, 동그라미에 도착했을 경우 에피소드 종료

epsilon=0.1

step size=0.01

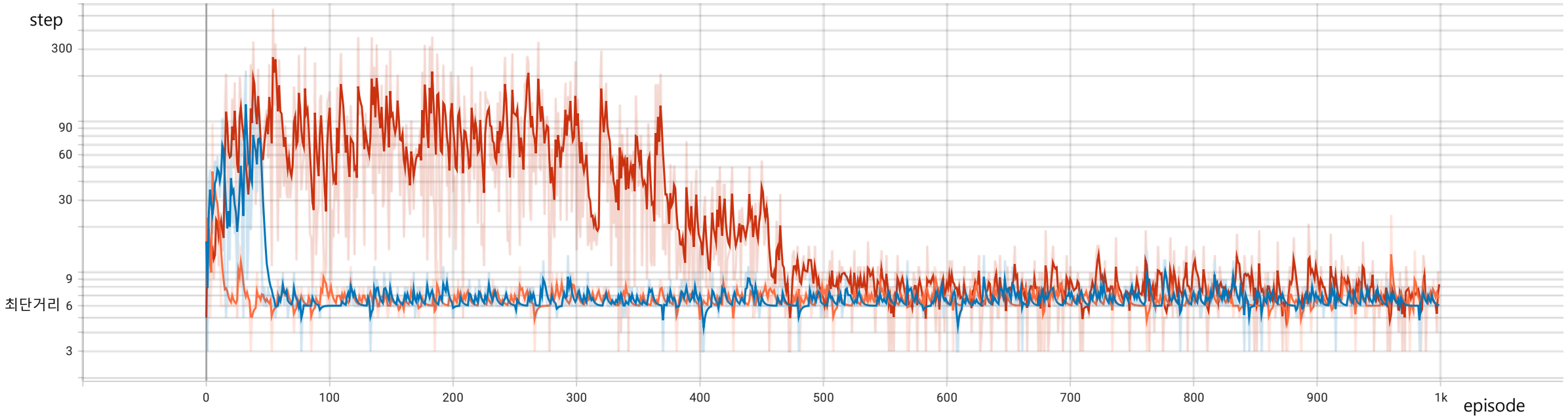
	삼각형	동그라미	그 외
보상	-100	100	0

- 성능비교

몇 에피소드만에 최단거리로 수렴하는지

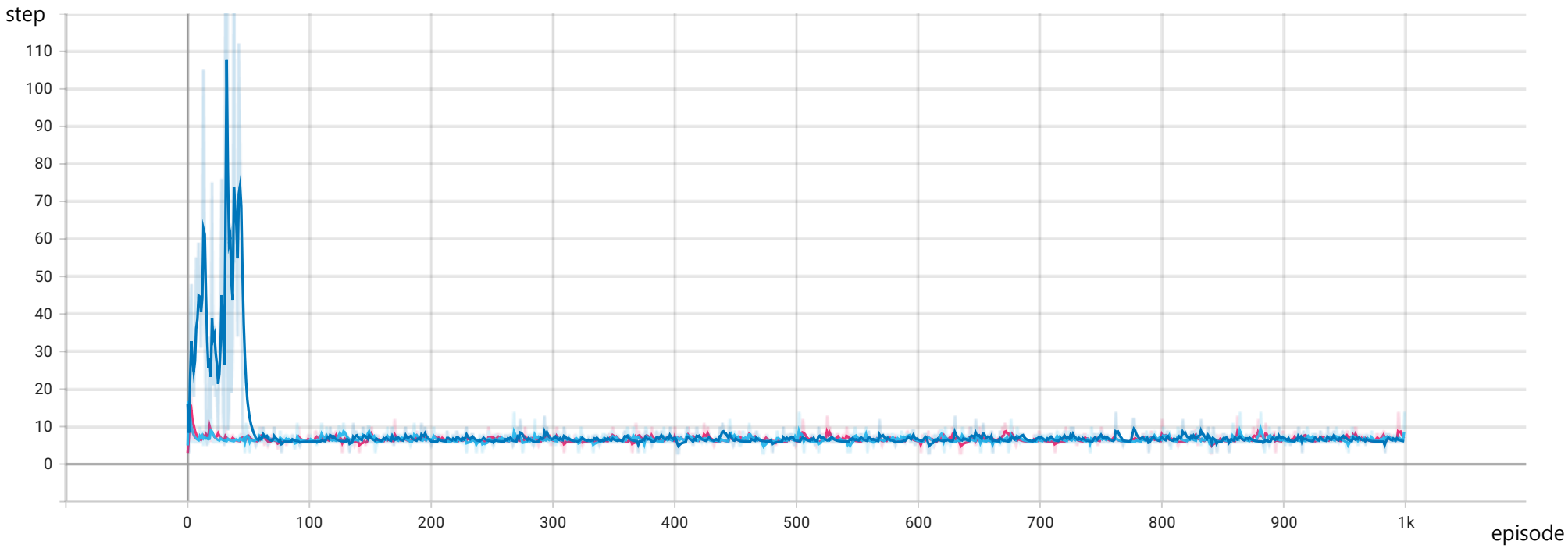
최단거리 : 6

성능비교(monte-carlo, SARSA, q-learning)

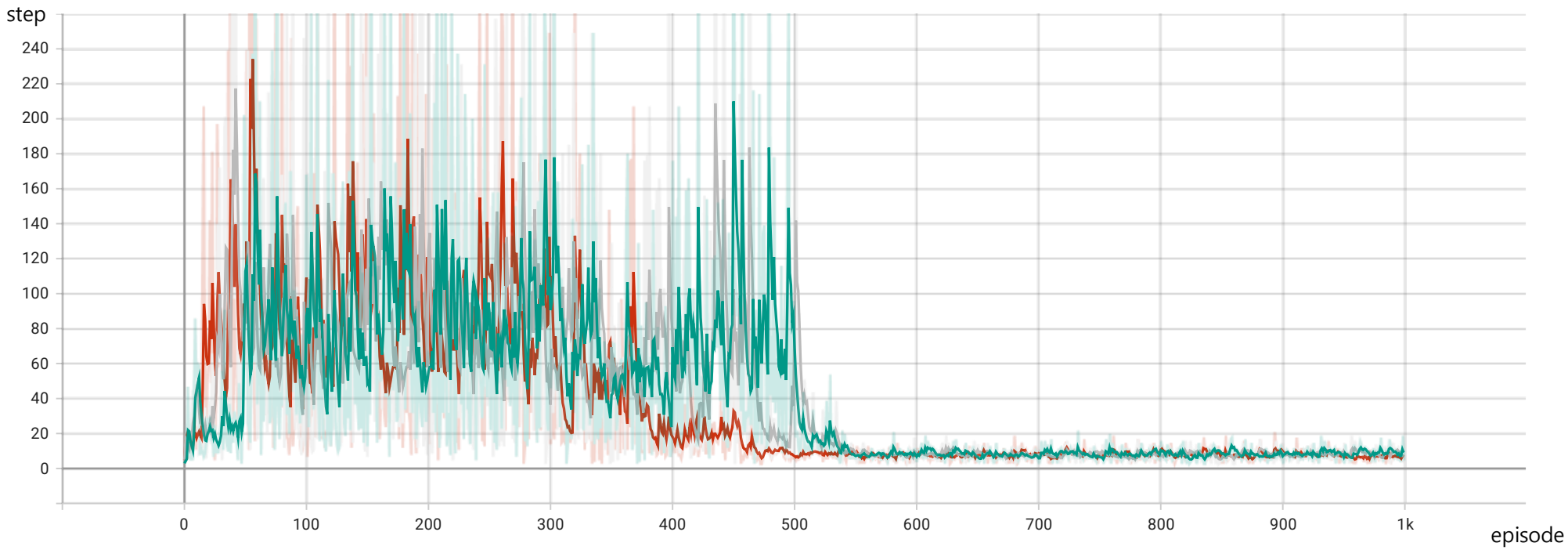


Monte-carlo : 45 Step	초반에 좋은 경로로 탐색을 하게되면 빠르게 수렴
SARSA : X	초반에 탐색은 잘 하지만 고립문제 발생
q-learning : 30 Step	다른 방법들보다 빠르게 수렴

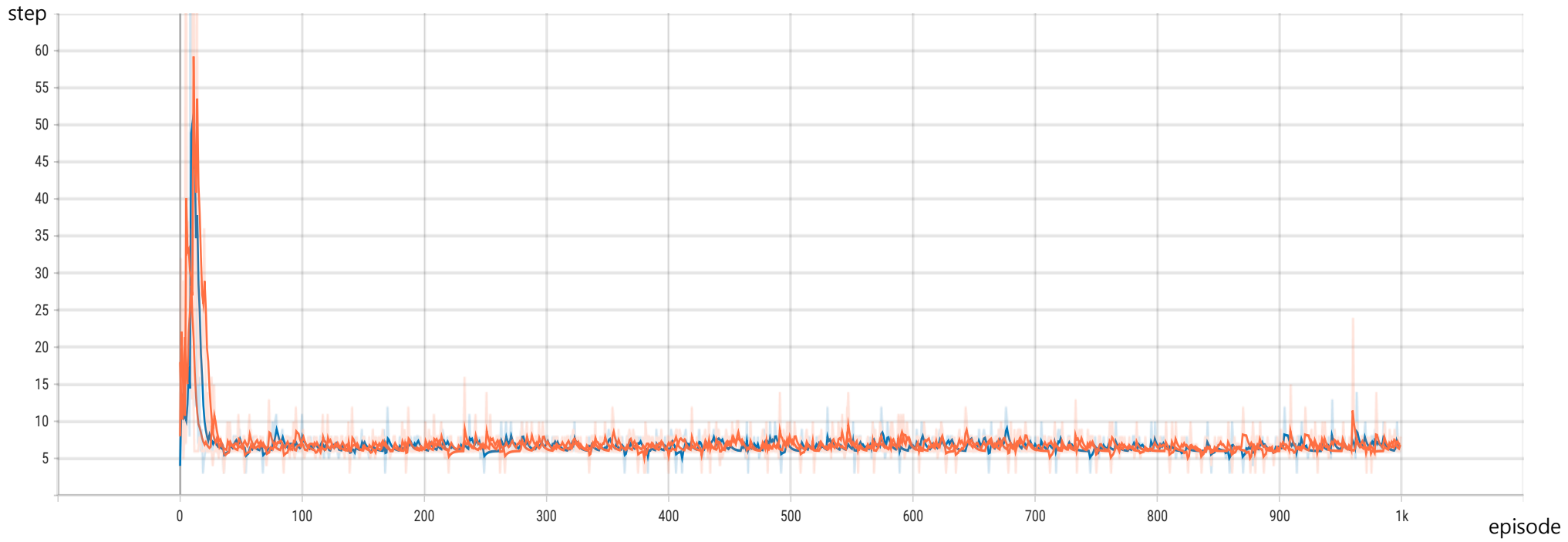
Monte-carlo


step
110
100
90
80
70
60
50
40
30
20
10
0
0
100
200
300
400
500
600
700
800
900
1k
episode

SARSA


step
240
220
200
180
160
140
120
100
80
60
40
20
0
0
100
200
300
400
500
600
700
800
900
1k
episode

q-learning


step
60
55
50
45
40
35
30
25
20
15
10
5
0
100
200
300
400
500
600
700
800
900
1k
episode