AI ASSIGNMENT FINAL

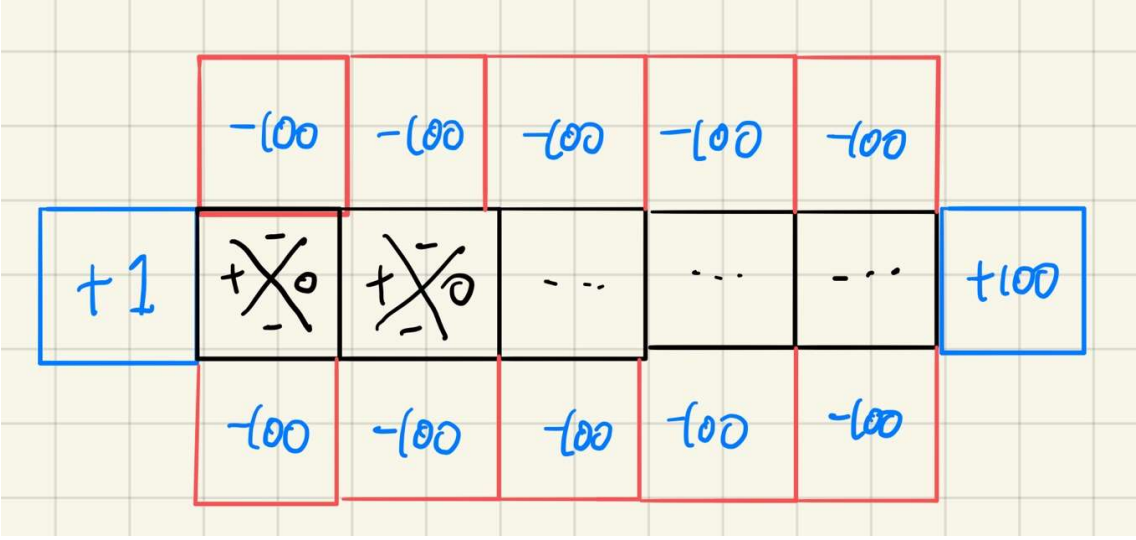
2018320196 컴퓨터학과 유지훈

SCREENSHOT

```
Provisional grades
==================
Question q1: 0/4
Question q2: 0/1
Question q3: 0/5
Question q4: 0/1
Question q5: 0/3
Question q6: 4/4
Question q7: 2/2
Question q8: 1/1
Question q9: 1/1
Question q10: 3/3
------------------
Total: 11/25
```

DISCUSS

Q8. NOT POSSIBLE



해당 문제는 불가능하다. 해당 grid에서 agent가 학습을 진행하면 각 타일들은 다음의 그림과 같이 생겼다. 검은색 상자가 다리에 해당하는 부분으로 왼쪽은 양수, 위 아래는 음수, 오른쪽은 0이 값으로 들어있다. 첫번째 검은색 노드에 도달했다고 했을 때, agent는 다리를 건너기 위해 5번의 epsilon에 의한 탐색 활동으로 오른쪽 행동을 택해야 한다. 그 확률은 ((1/4) * eps)^5이다. 이는 비록 정교하지 못한 계산이지만 policy에 따르면 계속 왼쪽으로 회귀하려 하고, 다리를 건너기 위해선 입실론 그리디에 의한 탐색의 burst가 발생해야 함을 보여주는 내용으로 충분하다고 생각한다. Agent가 다리를 건너기 위해서는 충분히 많은 양의 episode가 수행되어야 할 것이다. 그리 크지 않은 episode 내에서 99%이상의 확률로 항상 최적의 정책을 찾는 것을 eps과 alpha의 조정만으로는 불가능한 일이다.

2. Discuss the different behaviors of the weights for each feature in log_Q_weights.png image

ApproximateQAgent에서 SimpleExtractor를 사용했을 때, bias, # of ghosts 1 step away, eats-food, closest-food 네 가지 항목에 대한 특성 벡터를 이용해 Q 값을 계산한다.

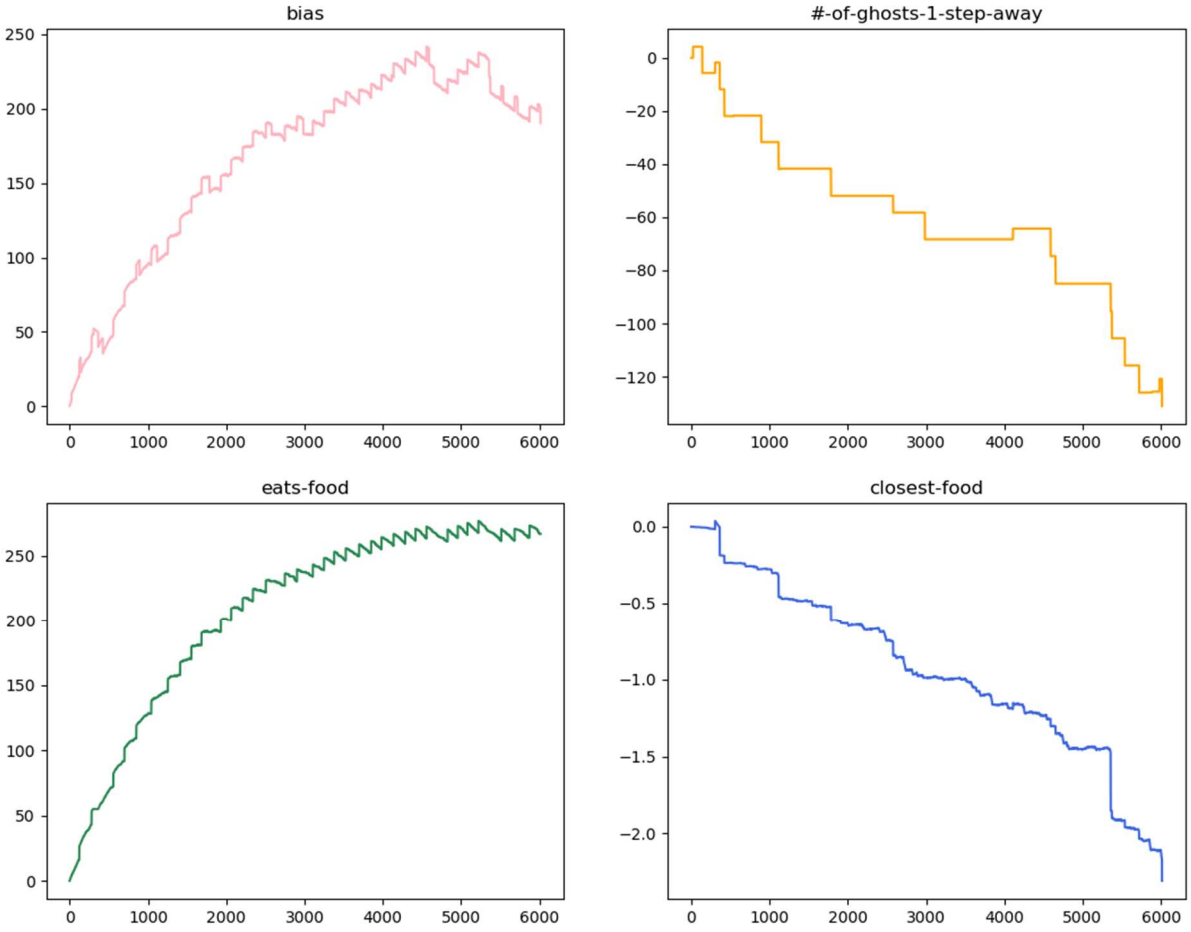
# of ghosts 1 step away는 인접한 곳에 유령이 얼마나 존재하는지를 알려주는 특성이다. 유령에 잡히면 점수의 손실과 더불어 게임이 종료되므로 팩맨은 최대한 유령을 피해가면서 점수를 획득해야 한다. 그래서 해당 특성에 대한 가중치는 음수를 가져야 팩맨이 유령들이 난잡히 돌아다니는 영역에 경계할 것이다. 그래프에 보이듯이 해당 가중치는 갈수록 작아지고 있다.

eats-food는 팩맨이 dot을 먹어 점수를 획득할 수 있도록 해주는 특성이다. dot을 먹는 것은 게임의 주 목적이므로 팩맨은 충분히 강력한 동기로 dot을 먹기위해 행동해야 한다. 따라서 해당 특성에 대한 가중치는 큰 양수에 머무르게 된다. 다만 그래프를 살펴보면 해당 특성의 가중치가 일정수준에 도달한 이후로는 값에 큰 변화가 없다. 그 이유는 팩맨이 dot을 먹는데 유령이라는 위험요소가 존재하기 때문에 가중치의 값이 한없이 커져버리면 위험요소에 대한 다른 특성들이 주는 경계를 묵살하여 결국 학습에 실패할 수 있다. 따라서 eat-food는 일정수준에 도달한 이후로는 해당 값을 유지한다.

closest-food는 팩맨이 dot을 향해 움직이도록 하는 길잡이 역할을 한다. 해당 특성은 팩맨의 위치와 가장 가까운 dot의 위치의 거리를 값으로 가지므로, 팩맨은 음식과의 거리를 좁히도록 행동해야 한다. 따라서 가중치는 음수를 갖게 되어 팩맨이 가까운 음식과의 거리를 좁히도록 안내한다.

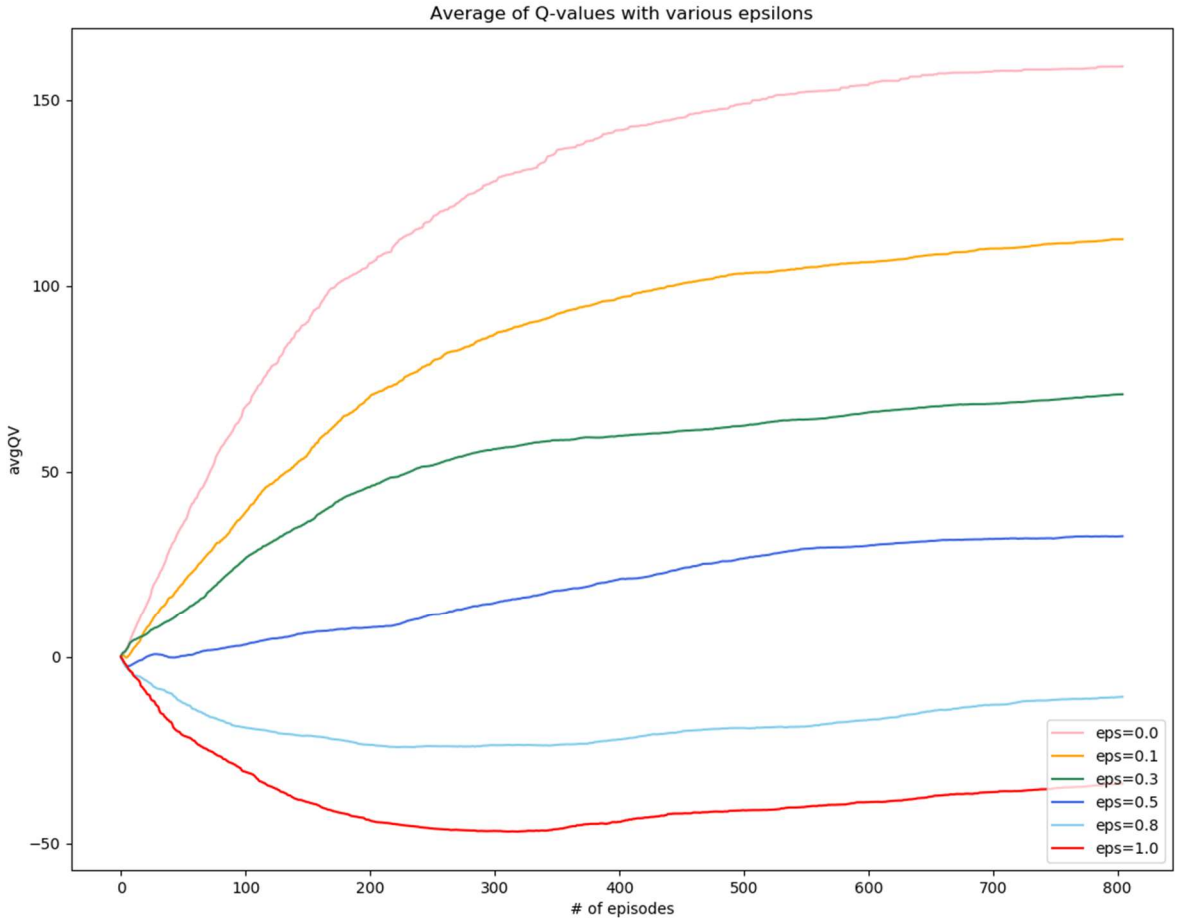
Approximate Q learning은 각 state에서 추출한 feature를 이용하여 일종의 regression을 통해 hyperplane을 찾는 것에 목표를 둔다. agent는 total error를 최소화하는 초평면을 찾을 텐데, bias는 최적의 초평면을 찾기 위해 함수를 shift해주는 역할을 한다. 만약 bias가 없다면 원점을 지나

고 기울기가 다른 초평면 만을 탐색하게 되므로 굉장히 비효율적인 학습이 될 수 있다. 따라서 bias는 최적의 초평면을 찾는 것에 적절한 값을 찾아 변화하고 있다.



3 + Extra credit problem.

In the epsilon-greedy search, discuss the convergence of Q-value according to epsilon.



mediumGrid에서 epsilon-greedy search로 팩맨을 800에피소드 학습시켰다. eps가 0인 경우에 평균 Q 값이 가장 크고, eps가 1인 경우에 평균 Q 값이 가장 작았다. eps가 0인 경우에는 exploration이 가장 떨어지는 환경이지만, 에피소드에서 무사히 optimal policy를 발견하여 해당 policy만을 따라가면서 가장 높은 Q 값이 나온 것으로 보인다. 반면 eps가 점점 커지는 경우에 optimal policy를 발견함에도 불구하고 policy에 벗어나 탐색하고자 하는 경향이 강해지면서 보다 많은 실패에 도달하는 것으로 보인다. 그래서 eps가 커질수록 평균 Q 값이 작은 수로 나왔다. eps가 각기 다른 경우라도 모든 경우에 대체로 학습한 episode가 커질수록 값의 변동이 줄어들고 수렴하는 양상을 보인다.

eps가 0인 경우에는 에피소드에 의해 찾아낸 policy만을 따라 갈 테니 수렴하는 것에 대한 수긍이 갔으나, eps가 1인 경우에는 무엇이 수렴을 보장할 수 있을 까 고민이 되었다. 비록 eps의 값에 따른 Q 값의 수렴성을 살펴본 그래프지만, default로 설정된 alpha에 따라 학습이 진행되었다. EMA와 TD등 optimal policy를 찾기 위한 여러 적용들이 만들어낸 우리의 Q value function에

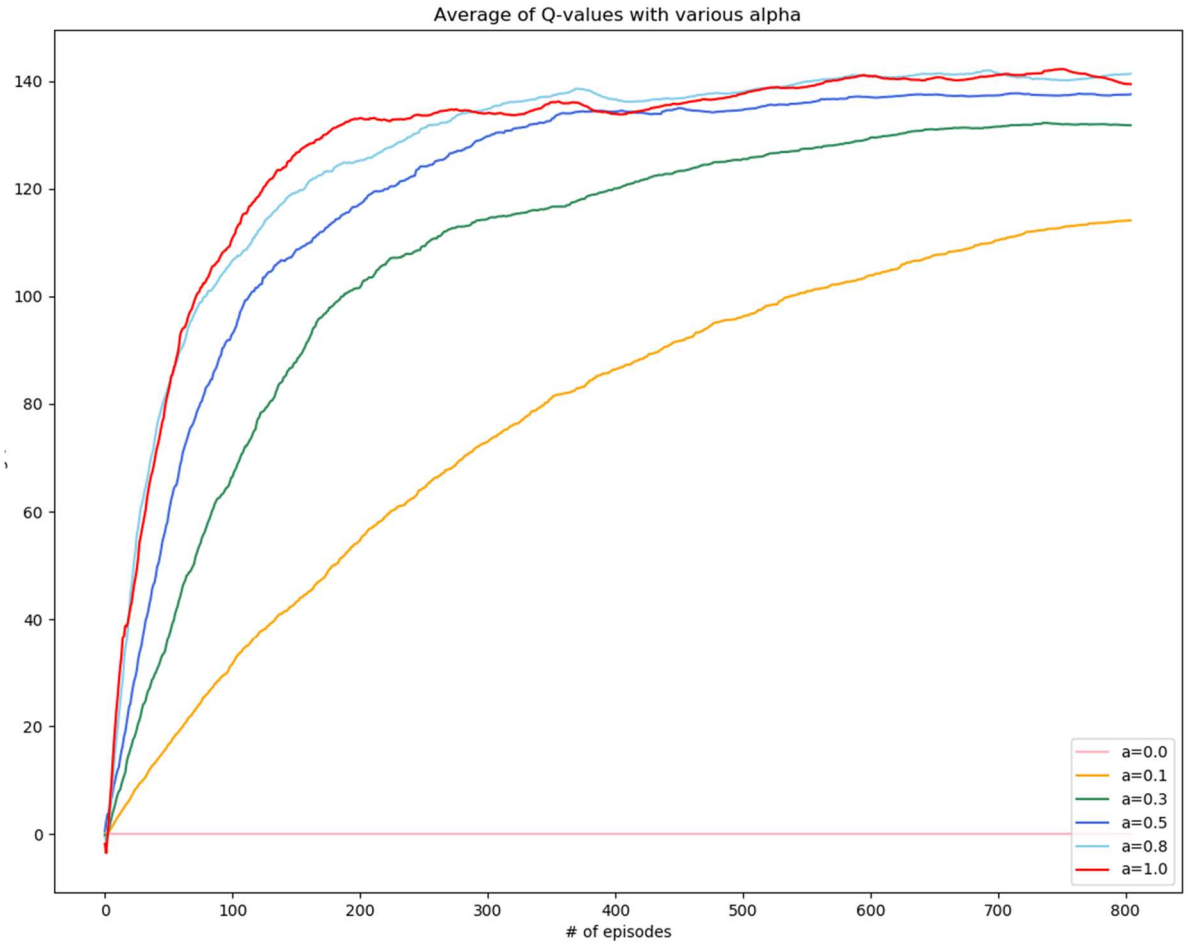
기본 설정된 alpha가 적용되면서 그래도 점차 수렴하는 모습을 보이지 않았을까 하는 생각이 든다.

해당 그래프는 update 메소드가 실행 될 때 마다 계산된 Q(s,a)의 값들을 저장하였고, 한 에피소드가 끝날 때마다 저장한 값들을 이용해서 평균 Q-value를 계산하였다. final 메소드를 이용해서 에피소드가 끝나는 시점에 계산한 평균 Q-value를 pickle 라이브러리로 파일 출력하였고, mathplot을 이용해 그래프로 그렸다.

4 + Extra credit problem.

In the Q-learning, discuss the convergence of Q-value according to alpha.

alpha의 값에 따른 Q 값의 수렴성을 보여주는 그래프를 만들었다.



해당 그래프 또한 epsilon에 따른 평균 Q 값의 수렴성을 찾은 그래프와 같은 방식으로 만들었다.

eps에서와 마찬가지로 mediumGrid에서 800에피소드 학습하였다. alpha의 값이 0인 경우를 제

외하고는 모두 일정한 값으로 수렴하는 모습을 보인다. alpha가 0일 때는 아무것도 학습하지 않겠다는 뜻으로 학습내용이 없고 결국 0에 수렴하게 된다. alpha가 1인 경우에는 기존의 내용을 전부 잊고 새롭게 학습한 내용만을 따른다는 것인데, 본 학습의 경우 에피소드만을 통해서 optimal policy를 찾아낼 수 있어서 큰 무리없이 좋은 성적을 내는 것으로 보인다.

alpha가 0인 경우를 제외하고는 EMA를 적용함으로써 탐색하는 episode가 커짐에 따라 수렴을 보장할 수 있는 학습이 진행되었다.

5 + Extra credit

Discuss any new features that might improve ApproximateQAgent and specific situations that the new features might be helpful

ApproximateQAgent에는 거리1에 존재하는 유령의 수, eat-food, closest-food가 존재한다. 여기에 유령의 scaredTime을 고려하여 # of ghosts 1 step away를 업데이트 하였고, eat-ghost, eat-capsule의 특성을 추가하였다.

팩맨에게 capsule을 먹고 유령을 무서워하지 않게 하는 정보를 주고자 하였다. 기존의 팩맨은 우연히 캡슐을 먹은 상태에서도 유령을 무서워하여 비효율적인 행동을 하였고, 또 캡슐이 있는 좁은 공간에 유령들이 여럿 몰려올 때 캡슐을 먹으면 살 수 있지만, 이를 알지 못하고 죽는 경우를 발견하였다. 따라서 유령에게 scaredTime이 존재함을 알려주고, 캡슐을 먹는 것이 도움이 될 수 있음을 알려주게 하였다. 이러한 정보들은 좁은 통로를 유령이 막고 있을 때, 캡슐을 이용하여 길을 열거나, 유령들에게 포위당했을 때 주변의 캡슐을 이용하는 등의 유리한 장면이 나올 수 있다.

먼저 # of ghosts 1 step away의 특성을 계산할 때에 scaredTime에 있는 유령은 제외하여, scaredTime에 있는 유령을 무서워하지 않도록 하였다. 이를 통해 개선된 # of ghosts 1 step away가 존재하지 않는 경우 주변에 유령이 있다면, 그 유령은 scaredTime에 있는 유령이다. 팩맨에게 무조건 적으로 유령을 먹는 것은 아니고, 필요에 따라서 유령을 먹는 행동도 가능함을 알려주었다. eat capsule의 경우 팩맨의 다음 행동으로 인한 새로운 위치에 캡슐이 있다면 이를 먹도록 동기를 유발하는 특성을 작성하였다. 이 모든 내용은 featureExtractor.py에 작성하였다.