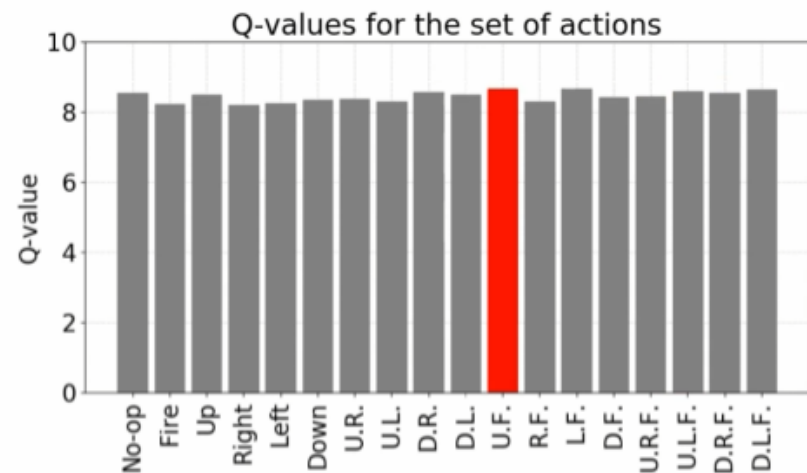


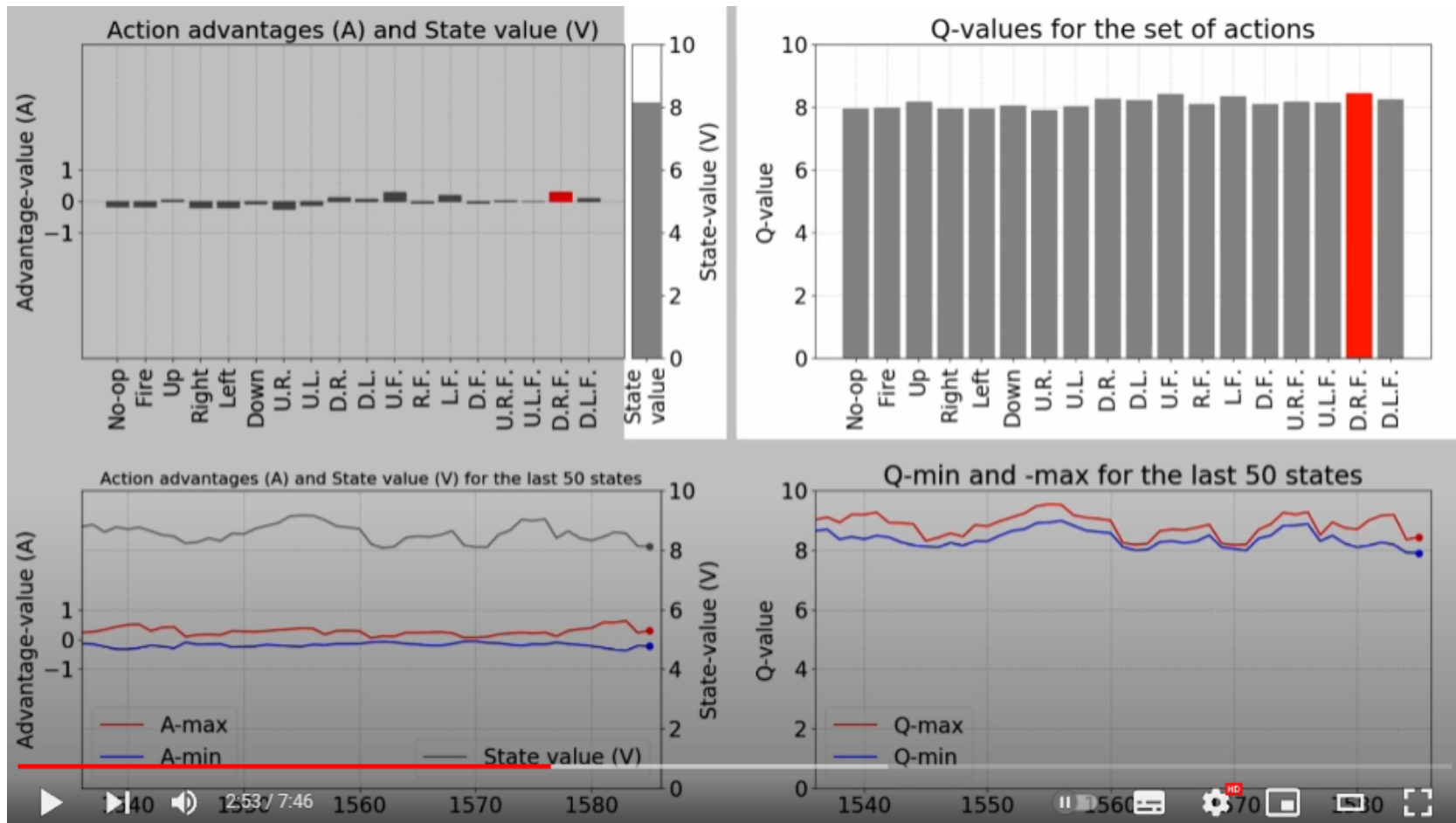
Dueling Network Architectures

for Deep Reinforcement Learning

Introducing game

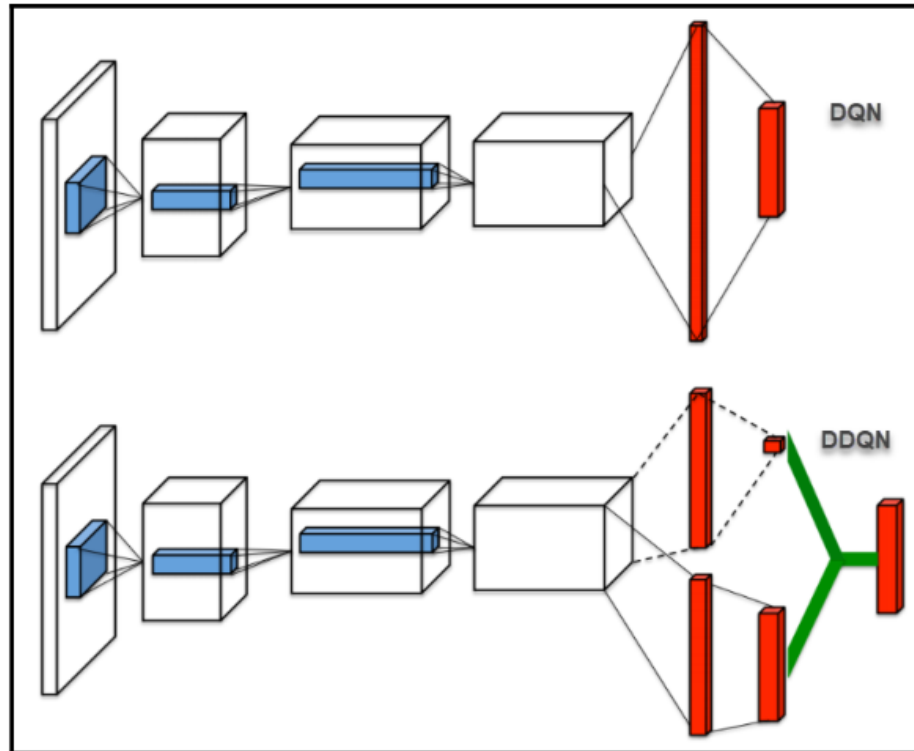


Introducing dueling DQN



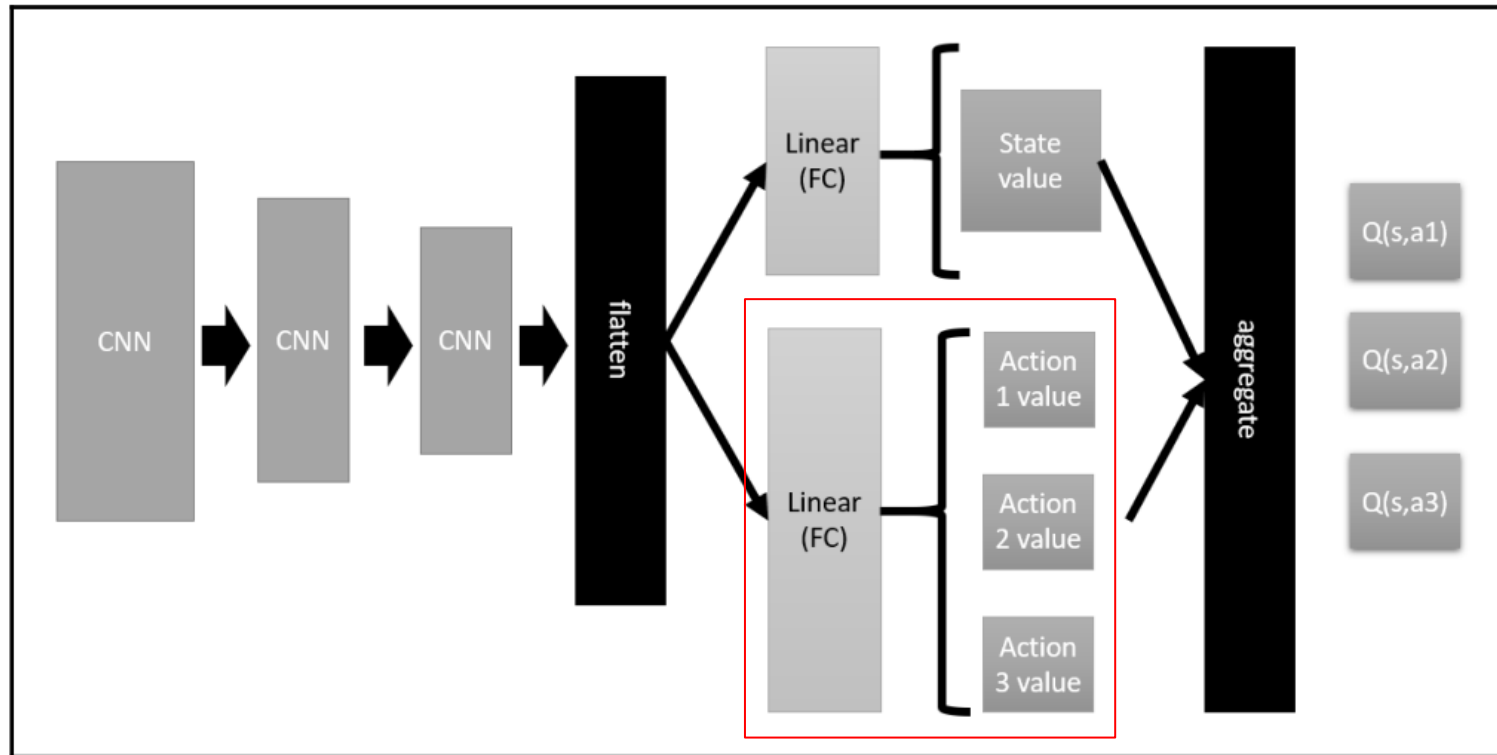
Introducing DDQN

- DDQN은 dueling DQN이며, double DQN이랑 다르다.
- Double DQN은 두 개의 다른 DQN 모델을 사용한다.



Dueling DQN or the real DDQN

- 한 행동을 선택해서 Q값을 구하는 것이 아니라 여러 행동을 통해 얻은 Q값을 모두 사용하고자 하는 모델



Advantage

Dueling DQN or the real DDQN

- Value: state s 에서 받을 보상의 크기
- Advantage: 다른 action에 비해 해당 action이 얼마나 좋은 지에 대한 척도

$$Q(s, a) = V(s) + A(s, a)$$

- Q값에 영향을 주는 값이 무엇인지 알기 힘들다는 단점 발생
- Optimal action a^* 을 선택하면 $Q(s, a) = V(s)$ 로 설정

$$Q(s, a) = V(s) + A(s, a) - \max_a A(s, a)$$

Dueling DQN or the real DDQN

- 게임: 달 착륙

```
class DDQN(nn.Module):
    def __init__(self, num_inputs, num_outputs):
        super(DDQN, self).__init__()

        self.feature = nn.Sequential(
            nn.Linear(num_inputs, 128),
            nn.ReLU()
        )

        self.advantage = nn.Sequential(
            nn.Linear(128, 128),
            nn.ReLU(),
            nn.Linear(128, num_outputs)
        )

        self.value = nn.Sequential(
            nn.Linear(128, 128),
            nn.ReLU(),
            nn.Linear(128, 1)
        )
```

```
def forward(self, x):
    x = self.feature(x)
    advantage = self.advantage(x) # (batch_size, 4)
    value = self.value(x) # (batch_size, 1)
    return value + advantage - advantage.mean()
```

Reference

- 논문발표 유튜브: <https://www.youtube.com/watch?v=XjsY8-P4WHM>
- <https://velog.io/@snoop2head/Dueling-DQN>

Hands-On Reinforcement Learning for Games

Ch7, Going Deeper with DDQN

감사합니다