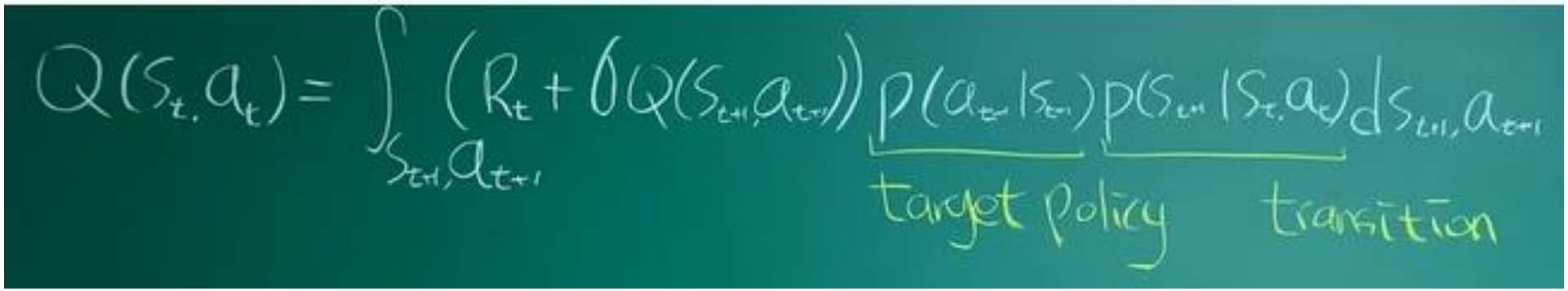
Ch 8

가치 기반 에이전트

Ch6 review

Q-function 요약



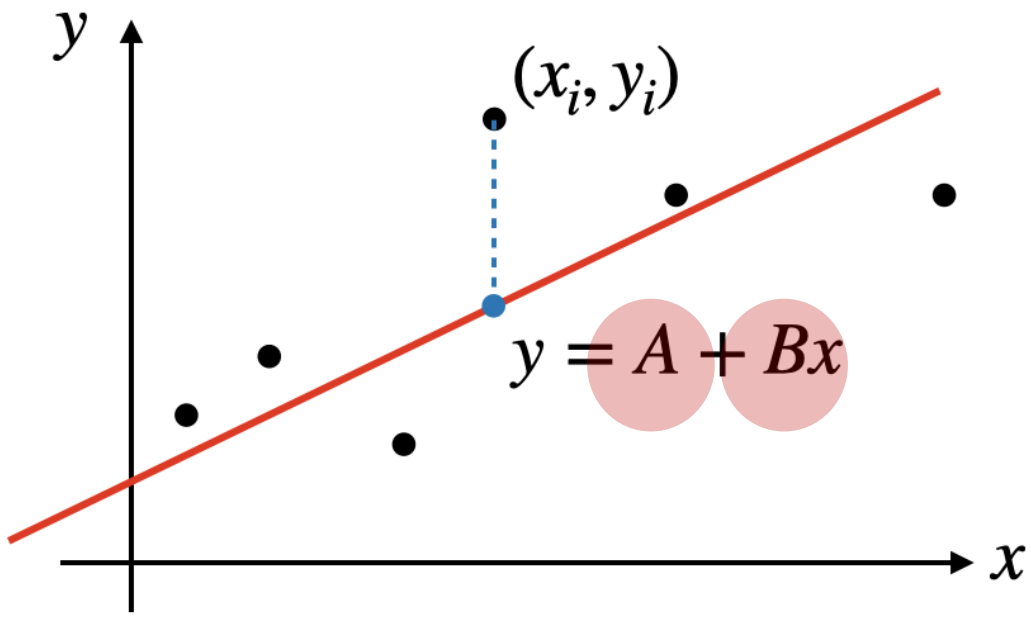
현재 state에서 현재 액션을 취했을 때 다음 state에 대한 평가를 내리고,
그 다음 state에서 target policy 로 액션을 뽑아 평균을 취해주는 것

Introduction

Deep Q Network?

Regression + Q_w

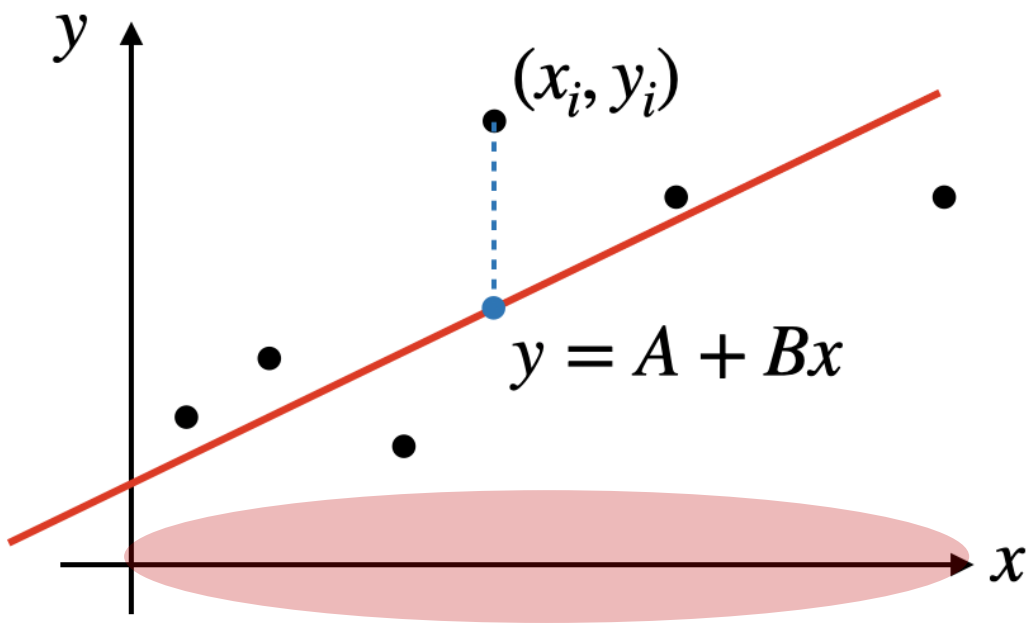
데이터에 잘 피팅되는 Q-function 찾기, w라는 파라미터를 통해 표현하자



Introduction

Why Deep Q Network?

현실에서는 state가 매우 다양,
state를 x축으로 봤을 때 경험해본 state s에 대해 학습을 통해 regression 된 Q-function을 알아내고, 경험해보지 못한 state s'에 대한 예측치를 찾아낸다.



Introduction

Deep Q Network? = Q-learning + DNN

Q-function의 parameter인 w을 gradient descent update하여
최종적으로 q-function의 최적 파라미터 w를 찾는다.

$$w_{t+1} = w_t - \eta \nabla w_t$$

$$L_i(\theta_i) = \mathbb{E}_{s,a\sim\rho(\cdot)}\left[(y_i - Q(s,a;\theta_i))^2\right]$$

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s,a\sim\rho(\cdot);s'\sim\mathcal{E}}\left[\left(r + \gamma \max_{a'} Q(s',a';\theta_{i-1}) - Q(s,a;\theta_i)\right)\nabla_{\theta_i} Q(s,a;\theta_i)\right]$$

W, 즉 여기선 theta에 대해 미분한 식

DQN

DQN 2013

Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih **Koray Kavukcuoglu** **David Silver** **Alex Graves** **Ioannis Antonoglou**

Daan Wierstra **Martin Riedmiller**

DeepMind Technologies

{vlad,koray,david,alex.graves,ioannis,daan,martin.riedmiller} @ deepmind.com

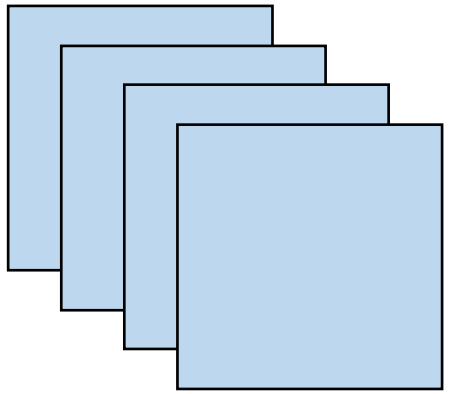
Deep Q Network

https://www.youtube.com/watch?v=V1eYniJ0Rnk

DQN

Deep Q Network 구조

IMAGE → CNN → 액션의 개수에 따른 Q(S,A)



84 * 84 * 4

방향 자체를 골랐다고 가정하고
Q 값을 구함

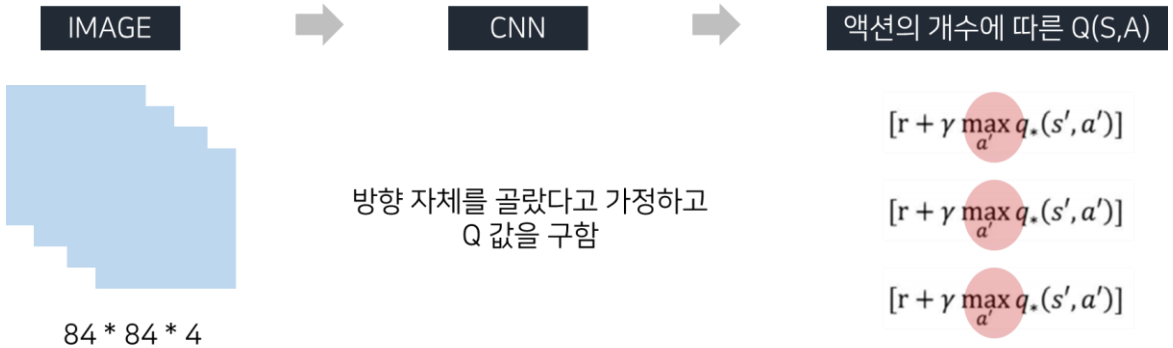
Q_w(S, A1) = $[r + \gamma \max_{a'} q_*(s', a')]$

Q_w(S, A2) = $[r + \gamma \max_{a'} q_*(s', a')]$

Q_w(S, A3) = $[r + \gamma \max_{a'} q_*(s', a')]$

DQN

Deep Q Network 구조



Loss = ($[r + \gamma \max_{a'} q_*(s', a')]$ - Q_w(S, A1)) ^2

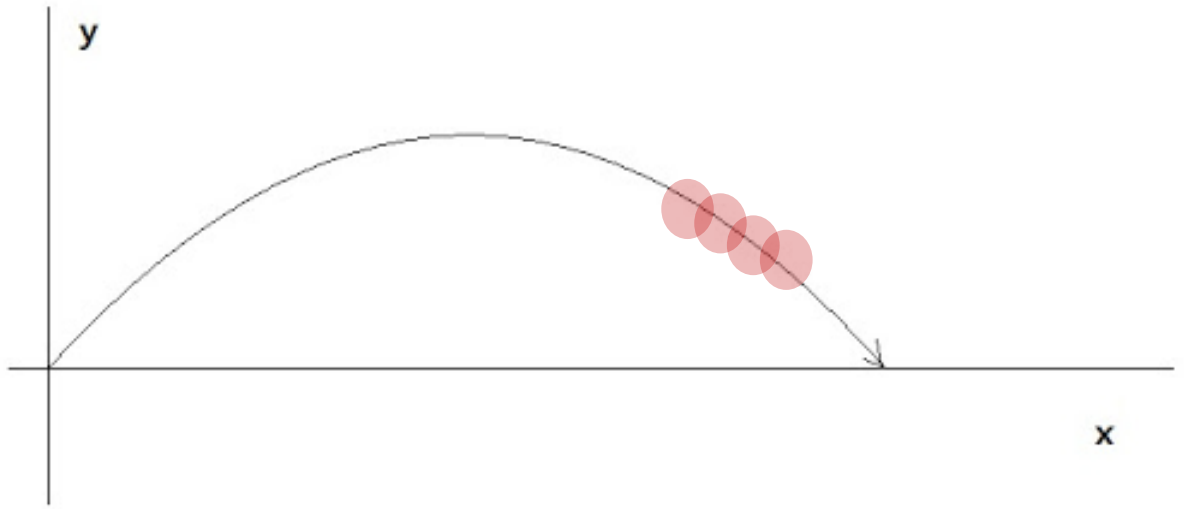
다른 action을 고려하지 않음,
고른 action에 대한 q값만 업데이트

DQN

Contribution

1. Experience replay with mini-batch SGD

➔ state correlation 감소



DQN

Contribution

1. Experience replay with mini-batch SGD

➔ state correlation 감소

IMAGE

DQN

Algorithm

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N Least N개만 저장 넘치면 오래된 것
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$

r의 함수 아님!

Action도 넣고 preprocessing
연속된 프레임동안 같은 action유지

DQN 2015

Algorithm

Initialize replay memory D to capacity N

Initialize action-value function Q with random weights θ

Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$

For episode $= 1, M$ **do**

Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\phi_1 =$

For $t = 1,$T **do**

With probability ε select a random action a_t

otherwise select $a_t = \text{argmax}_a Q(\phi(s_t), a; \theta)$

Execute action a_t in emulator and observe reward r_t and i

Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$

Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in D

Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ f

Set $y_j = \begin{cases} r_j & \text{if episode termina} \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^{-}) & \text{other} \end{cases}$

Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ w

network parameters θ

Every C steps reset $\hat{Q} = Q$

DQN 2015

Algorithm

$$L(\theta_i) = \mathbb{E}\left[\left(R + \gamma \max_{A'} Q_{\theta_i^-}(S', A') - Q_{\theta_i}(S, A)\right)^2\right]$$

상수화

타겟 네트워크
θ_i^-

Q네트워크
θ_i

일정 주기마다
$\theta_i^- \leftarrow \theta_i$