DQN

: Deep Q-Network

24.02.15

정상혁

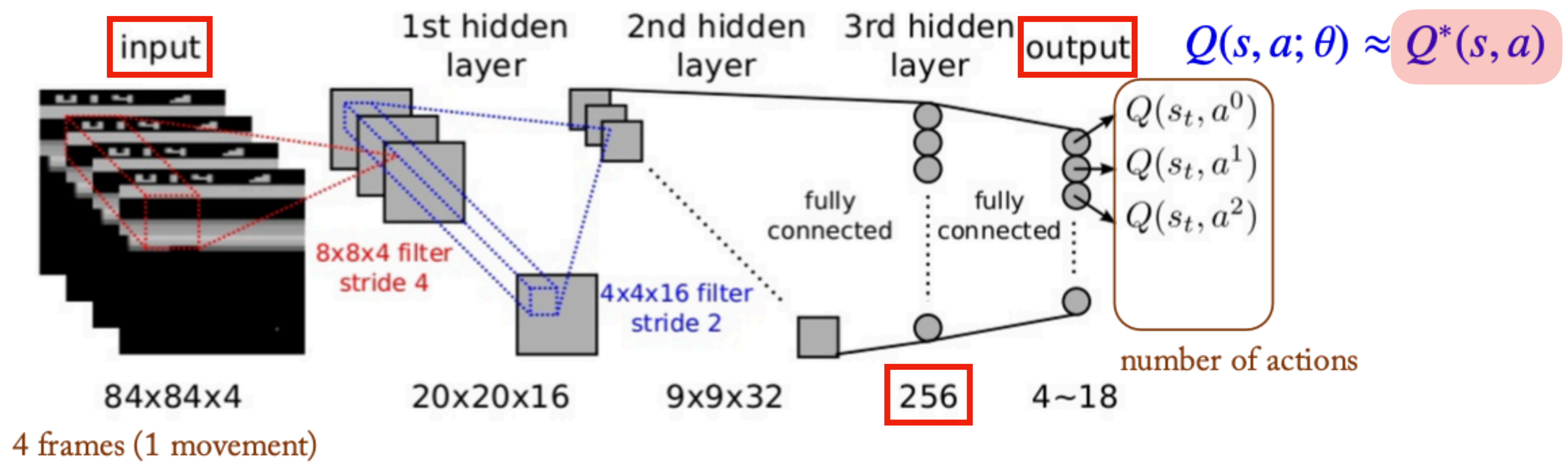
Deep Q-Network (DQN)

- **DQN :** Q-learning model

  Q-learning → ~~Q-table~~ → Q-Network (CNN)

  * CNN in DQN does not use Max pooling

  (∵ max pooling causes translation invariance)



- Naive DQN

$$L(\theta) = [r_{t+1} + \gamma \max_a Q(s_{t+1}, a; \theta) - Q(s_t, a_t; \theta)]^2$$

Problem		DQN
• Temporal correlation	→ solution	• Experience Replay
• Non-stationary target	→	• Target Network

Q-learning (policy evaluation)

$$q_*(s, a) = E[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \mid S_t = s, A_t = a]$$ (Bellman optimality equation)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{n(S_t, A_t)}[R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta) - Q(S_t, A_t)]$$ (Incremental mean)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta) - Q(S_t, A_t)]$$ (Constant-α)

(based on law of large numbers with i.i.d assumption)

Target Network

- Naive DQN

$$L(\theta) = [r_{t+1} + \gamma \max_a Q(s_{t+1}, a; \theta) - Q(s_t, a_t; \theta)]^2$$

$\rightarrow$ **Non-stationary problem**

- Q-Network $\rightarrow$ Behavior Q-network & **Target $\hat{Q}$-network**

Initialize θ, $\hat{\theta}\ (= \theta)$ $\longrightarrow$ Updates θ (N times) $\rightleftarrows$ $\hat{\theta} \leftarrow \theta$

- Weight update

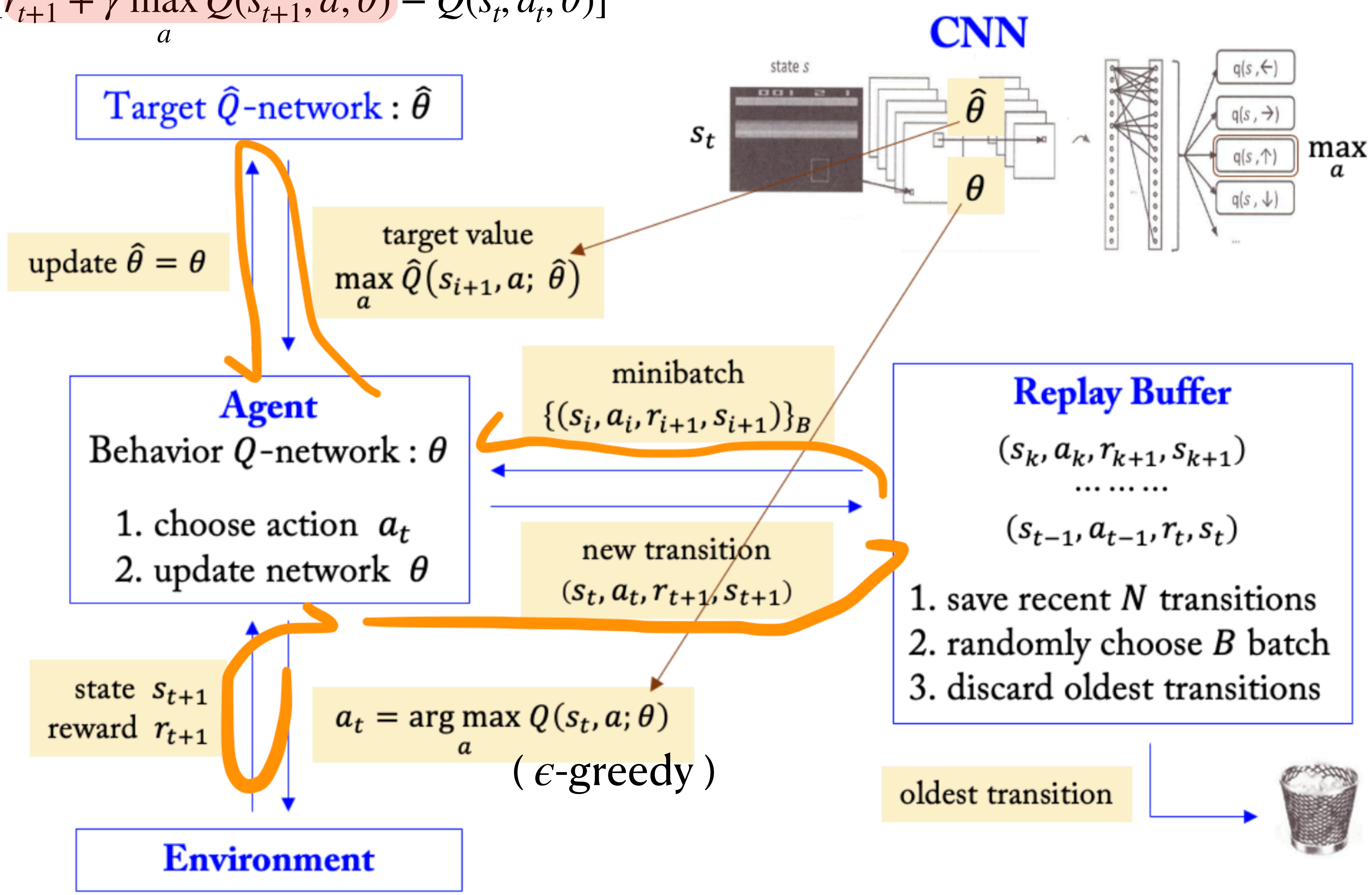
$$L(\theta) = [r_{t+1} + \gamma \max_a \hat{Q}(s_{t+1}, a; \hat{\theta}) - Q(s_t, a_t; \theta)]^2$$

$$\nabla_\theta L(\theta) = -[r_{t+1} + \gamma \max_a \hat{Q}(s_{t+1}, a; \hat{\theta}) - Q(s_t, a_t; \theta)]\, \nabla_\theta Q(s_t, a_t; \theta)$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta)$$

DQN

$$L(\theta) = [r_{t+1} + \gamma \max_a \hat{Q}(s_{t+1}, a; \hat{\theta}) - Q(s_t, a_t; \theta)]^2$$



DQN

Initialize behavior network Q with random weights θ
Initialize target network $\hat{Q}$ with weights $\hat{\theta} = \theta$
Initialize replay buffer $\mathcal{R}$ to capacity N
for episode $= 1, M$ **do**
  Initialize sequence $s_1 = \{x_1\}$ and preprocess $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ, select a random action a_t
    otherwise select $a_t = \arg\max_a Q(\phi_t, a; \theta)$ ϵ-greedy CNN θ
    Execute a_t in emulator and observe reward r_{t+1} 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 $(\phi_t, a_t, r_{t+1}, \phi_{t+1})$ in $\mathcal{R}$ Replay Bufffer
    Sample minibatch of B transitions $(\phi_i, a_i, r_{i+1}, \phi_{i+1})$ from $\mathcal{R}$
    Set $y_i = r_{i+1} + \gamma \max_a \hat{Q}(\phi_{i+1}, a; \hat{\theta})$ CNN $\hat{\theta}$
    Perform a gradient descent on $(y_i - Q(\phi_i, a_i; \theta))^2$ update θ
    Every C steps, reset $\hat{Q} = Q$ (i.e., $\hat{\theta} = \theta$) update $\hat{\theta}$
  end
end

Behavior network

Target network

* Preprocessing ϕ: raw frame x_t (128 colors, 210×160 pixels) is converted to gray-scale 110×84 pixels and again cropped to 84×84 pixels.

Performance

	DQN			Naïve DQN	Linear NN
Replay	O	O	X	X	
Target N	O	X	O	X	
Breakout	316.8	240.7	10.2	3.2	3.0
River Raid	7446.6	4102.8	2867.7	1453.0	2346.9
Seaquest	2894.4	822.6	1003.0	275.8	656.9
Space Invaders	1088.9	826.3	373.2	302.0	301.3