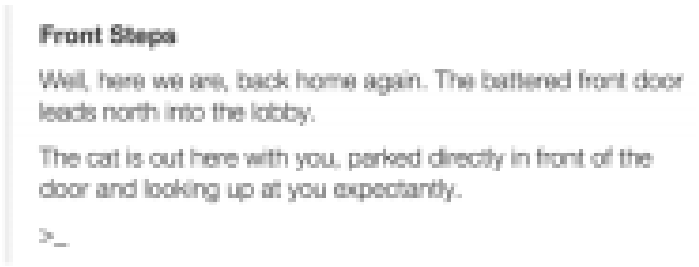
DRRN

deep reinforcement learning with a natural language action space

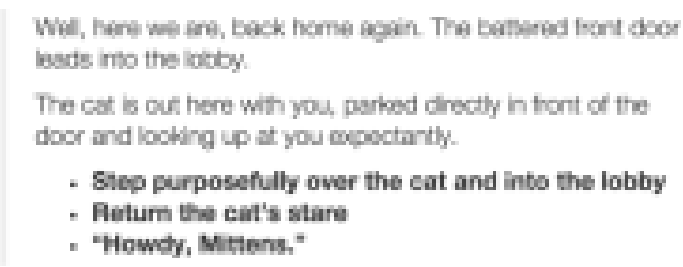
김지훈
20.08.10

- text-based game 에서 사용되는 자연어들
- state 공간과 action 공간들을 다루는 알고리즘
- Deep Reinforcement Relevance Network

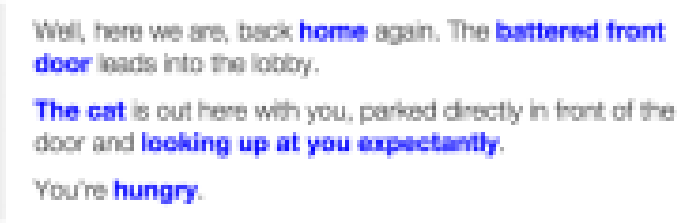
Text based game?



(a) Parser-based



(b) Choiced-based

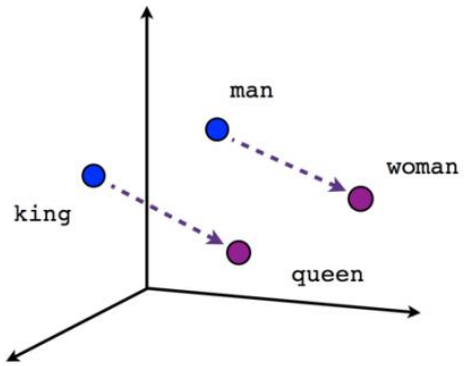


(c) Hypertext-based

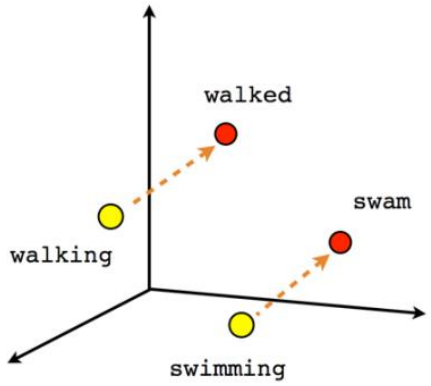

"Ah, FINALLY someone came around.
Boy, was I getting tired of sand in my speakers.
What's up, guy? How did you end up on this deserted island?"
...

분리된 embedding vector로 state space와 action space를 표현한다

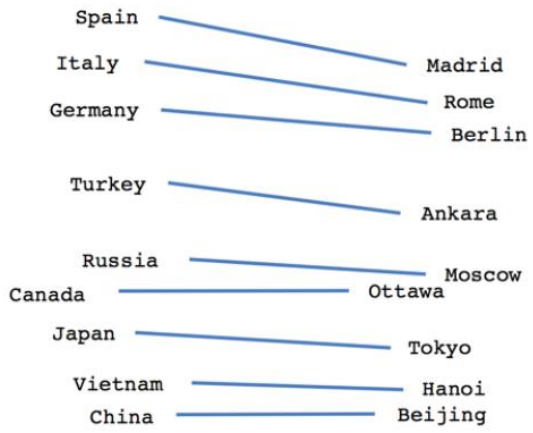
Embedding vector



Male-Female



Verb tense



Country-Capital

출처 https://towardsdatascience.com/deep-learning-4-embedding-layers-f9a02d55ac12

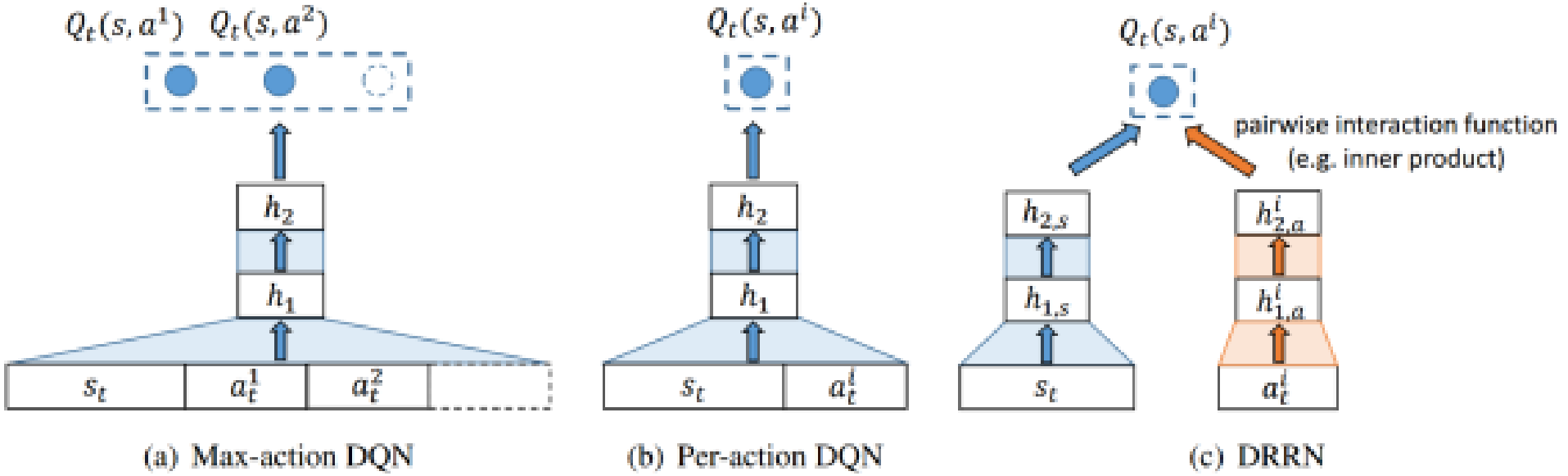
Introduction

DRRN은 분리된 deep neural network를 사용

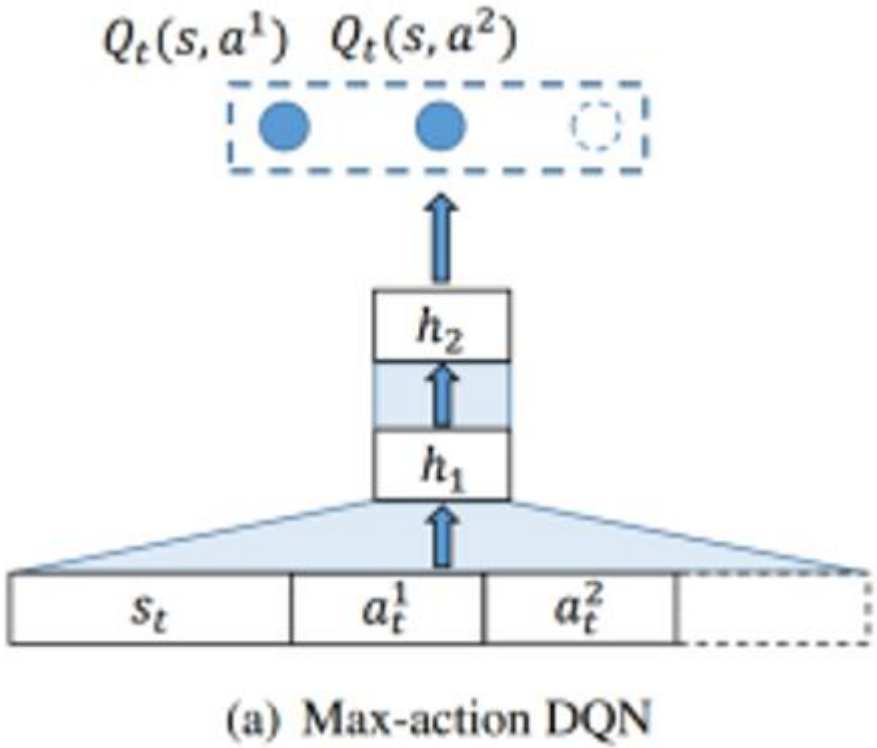
이 함수의 output은 현재 state-action 쌍의 Q함수의 값으로 정의된다.

DRRN의 가장 큰 특징은 다른 의미를 표현하는 두 가지 다른 타입이 학습된다는 것

3가지 DQN


$Q_t(s, a^1)$ $Q_t(s, a^2)$
h_2
h_1
s_t a_t^1 a_t^2
(a) Max-action DQN
$Q_t(s, a^i)$
h_2
h_1
s_t a_t^i
(b) Per-action DQN
$Q_t(s, a^i)$
pairwise interaction function
(e.g. inner product)
$h_{2,s}$
$h_{1,s}$
s_t
$h_{2,a}^i$
$h_{1,a}^i$
a_t^i
(c) DRRN

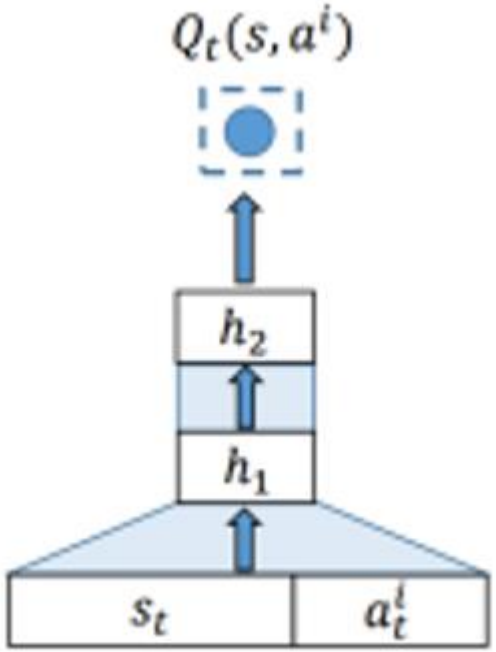
3가지 DQN



(a) Max-action DQN

State 벡터와 Action 벡터가 합쳐져 있다.

3가지 DQN



(b) Per-action DQN

State 벡터와 Action 벡터의 쌍으로 이루어져 있다

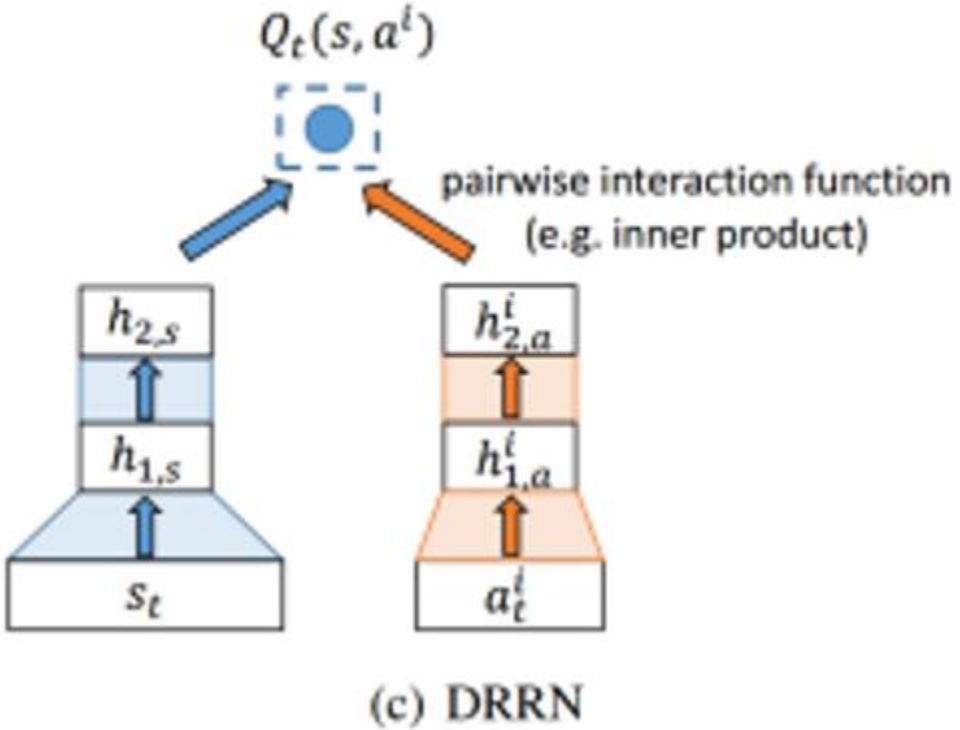
3가지 DQN

State –〉 매우 길다

Action –〉 짧다

같은 Network?

3가지 DQN



(c) DRRN

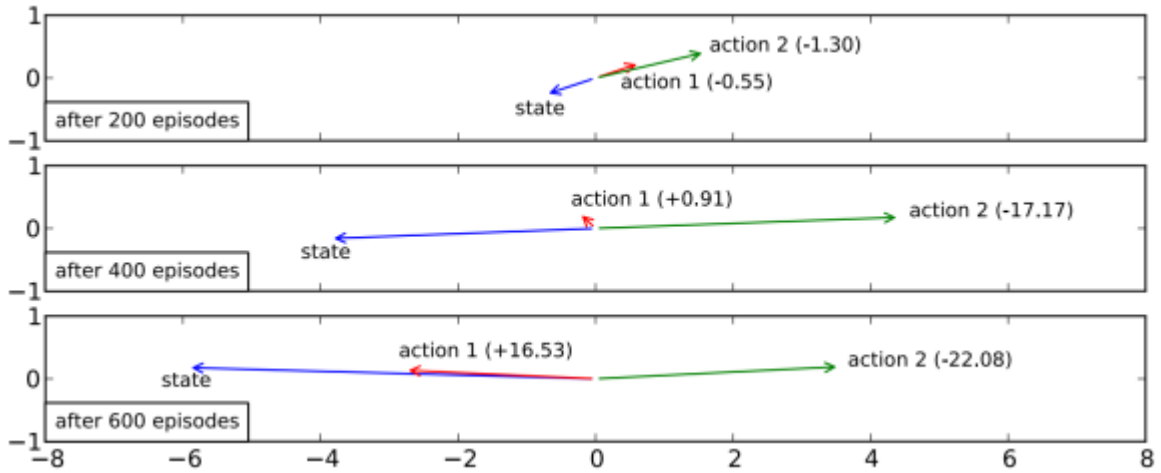
DNN들의 쌍으로 이루어져 있다.

DRRN

DNN 쌍으로 분리
State(긴) vector와 Action(짧은) vector을 따로 학습

Hidden layer는 2개
Hidden layer의 dimension을 20-50-100 차원으로 테스트

PCA



state 구문 : "As you move forward, the people surrounding you suddenly look up with terror in their faces, and flee the street."

action 1 : "look up" **-> good choice**
action 2 : "Ignore the alarm of others and continue moving forward"

알고리즘

Algorithm 1 Learning algorithm for DRRN

Initialize replay memory $\mathcal{D}$ to capacity N.
Initialize DRRN with small random weights.
Initialize game simulator and load dictionary.
for $episode = 1, \ldots, M$ **do**
  Restart game simulator.
  Read raw state text and a list of action text from the simulator, and convert them to representation s_1 and $a_1^1, a_1^2, \ldots, a_1^{|\mathcal{A}_1|}$.
  for $t = 1, \ldots, T$ **do**
    Compute $Q(s_t, a_t^i; \Theta)$ for the list of actions using DRRN forward activation (Section 2.3).
    Select an action a_t based on probability distribution $\pi(a_t = a_t^i | s_t)$ (Equation 2)
    Execute action a_t in simulator
    Observe reward r_t. Read the next state text and the next list of action texts, and convert them to representation s_{t+1} and $a_{t+1}^1, a_{t+1}^2, \ldots, a_{t+1}^{|\mathcal{A}_{t+1}|}$.
    Store transition $(s_t, a_t, r_t, s_{t+1}, A_{t+1})$ in $\mathcal{D}$.
    Sample random mini batch of transitions $(s_k, a_k, r_k, s_{k+1}, A_{k+1})$ from $\mathcal{D}$.
    Set $y_k = \begin{cases} r_k & \text{if } s_{k+1} \text{ is terminal} \\ r_k + \gamma \max_{a' \in A_{k+1}} Q(s_{k+1}, a'; \Theta)) & \text{otherwise} \end{cases}$
    Perform a gradient descent step on $(y_k - Q(s_k, a_k; \Theta))^2$ with respect to the network parameters Θ (Section 2.4). Back-propagation is performed only for a_k even though there are $|\mathcal{A}_k|$ actions at time k.
  end for
end for

Saving John vs Machine of Death

Game	Saving John	Machine of Death
Text game type	Choice	Choice & Hypertext
Vocab size	1762	2258
Action vocab size	171	419
Avg. words/description	76.67	67.80
State transitions	Deterministic	Stochastic
# of states (underlying)	≥ 70	≥ 200

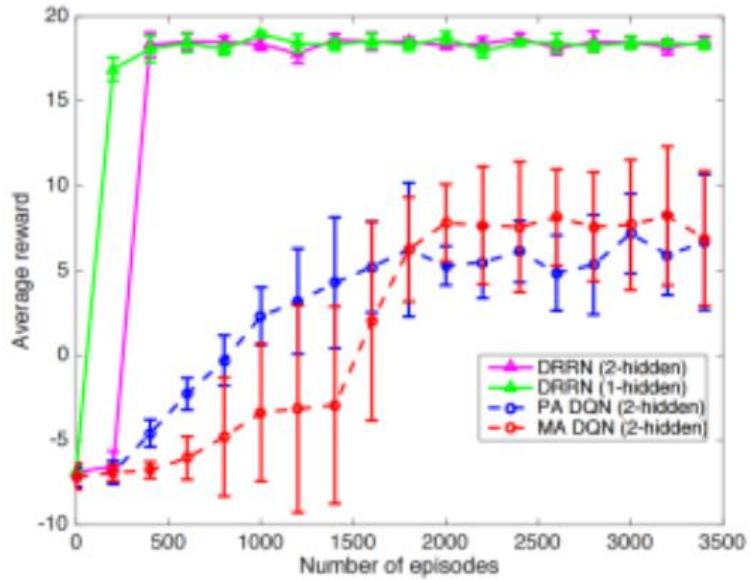
Table 1: Statistics for the games "Saving John" and and "Machine of Death".

Eval metric	Average reward		
hidden dimension	20	50	100
Linear	4.4 (0.4)		
PA DQN ($L = 1$)	2.0 (1.5)	4.0 (1.4)	4.4 (2.0)
PA DQN ($L = 2$)	1.5 (3.0)	4.5 (2.5)	7.9 (3.0)
MA DQN ($L = 1$)	2.9 (3.1)	4.0 (4.2)	5.9 (2.5)
MA DQN ($L = 2$)	4.9 (3.2)	9.0 (3.2)	7.1 (3.1)
DRRN ($L = 1$)	17.1 (0.6)	18.3 (0.2)	18.2 (0.2)
DRRN ($L = 2$)	18.4 (0.1)	18.5 (0.3)	**18.7** (0.4)

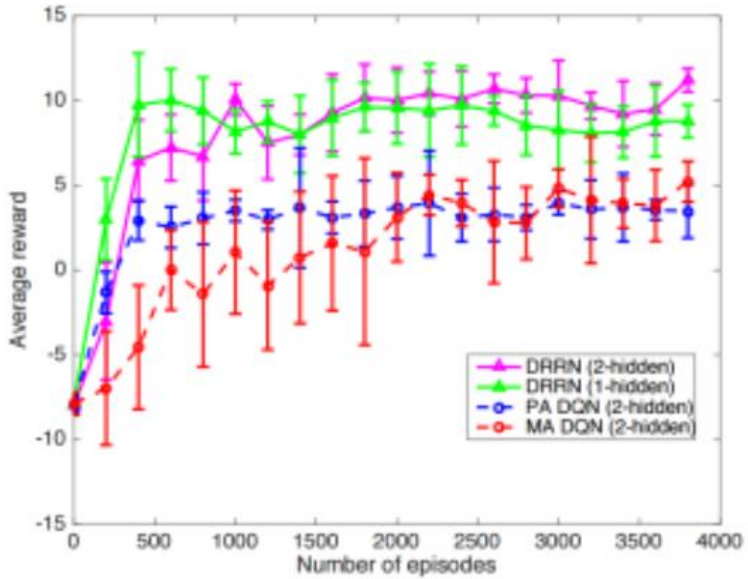
Table 2: The final average rewards and standard deviations on "Saving John".

Eval metric	Average reward		
hidden dimension	20	50	100
Linear	3.3 (1.0)		
PA DQN ($L = 1$)	0.9 (2.4)	2.3 (0.9)	3.1 (1.3)
PA DQN ($L = 2$)	1.3 (1.2)	2.3 (1.6)	3.4 (1.7)
MA DQN ($L = 1$)	2.0 (1.2)	3.7 (1.6)	4.8 (2.9)
MA DQN ($L = 2$)	2.8 (0.9)	4.3 (0.9)	5.2 (1.2)
DRRN ($L = 1$)	7.2 (1.5)	8.4 (1.3)	8.7 (0.9)
DRRN ($L = 2$)	9.2 (2.1)	10.7 (2.7)	**11.2** (0.6)

Table 3: The final average rewards and standard deviations on "Machine of Death".



(a) Game 1: "Saving John"



(b) Game 2: "Machine of Death"