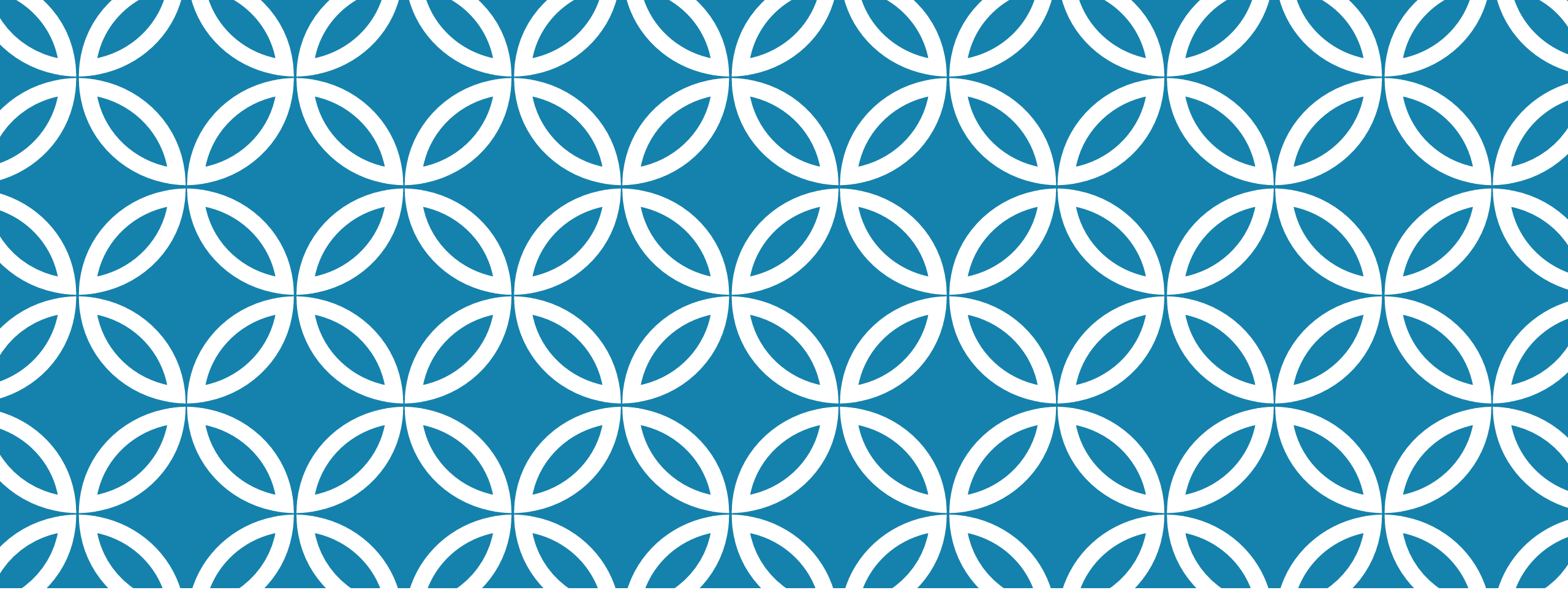


HUMAN-LEVEL CONTROL THROUGH DEEP REINFORCEMENT LEARNING

2020097010 남도현

THIS PAPER IS ABOUT…?

- DQN : Deep Q-Network
- Famous Atari Breakout video used this method

- To understand DQN, we have to know Q-Learning

Q-LEARNING

Q-Table	State 1	State 2	…	State n
Action 1	Q(1, 1)	Q(2, 1)		Q(n, 1)
Action 2	Q(1, 2)	Q(2, 2)		Q(n, 2)
:				
Action m	Q(1, m)	Q(2, m)		Q(n, m)

Q(S, A) -> What is Quality of Action in State?

If this Q-Table is perfect, all you have to do is just do action which has highest value in the state

But then, how to make perfect Q-Table?

Q-LEARNING (CONT.)

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

Image from Wikipedia

Quality of state s_t, action a_t in time t is determined by

Weighted addition of known quality of the state and the action
and immediate reward + estimated quality * certain value (discount factor : immediate is important)

Get reward from environment, and repeat it.

Method note : to travel unknown action per state, use ε-greedy
i.e., select full random action time to time,
high probability in earlier stage, low in late stage.

BUT WHY NOT PURE Q-LEARNING?

- Is applied only in discrete states, discrete actions

- Cannot apply for continuous tasks, like Cartpole

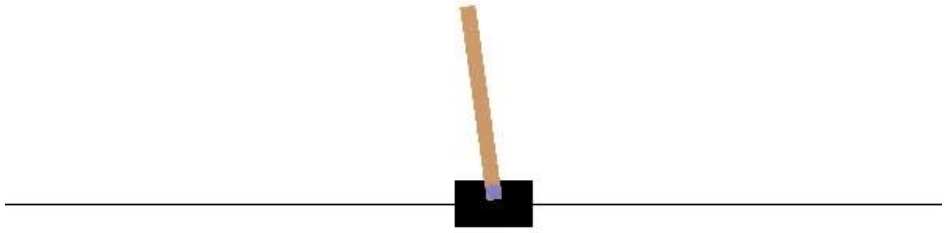


Image from OpenAI Gym

IDEA IS SIMPLE AND OLD…

- Neural Network can be used as universal function estimator.

- Why not just use NN instead Q-Table?
  - You know, table is function – can't we just use neural network?
  - It can replace Q-Table, and can be used in continuous task!

. . . BUT THERE WAS PROBLEM.

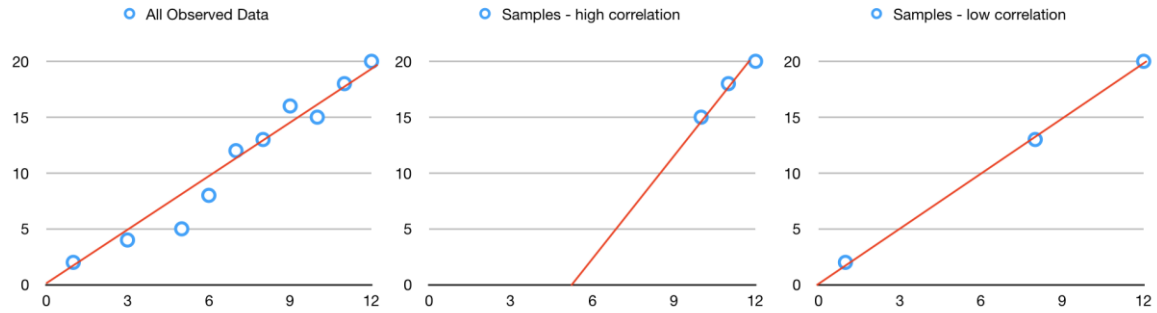
- Of course authors was not first ones who came up with this idea

- But there was certain problem, and NN just refused to learn.

- The paper suggested there is three causes.

WHY DOESN'T NEURAL Q-LEARNING WORK?

- Correlation presents in observation
  - Deep learning don't do well when sample has high correlation



Images from https://curt-park.github.io/2018-05-17/dqn/

- We have to learn Q / but Q is also target

Naïve method

$$L_i(\theta_i) = \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma max_{a'} Q(s', a'; \theta_i) - Q(s, a; \theta_i)\right)^2\right],$$

where θ_i are the parameters of the Q-network at iteration i.

-> Loss is MSE of [(instant reward + certain value * future quality) - quality]

SOLUTION

-Experience replay

- Store experiences, and then replay
- 'removing correlations in the observation sequence'

-Freeze target (Target Q)

- Action value is updated iteratively
- Target value is updated periodically

- -> Action Q / Target Q correlation reduced

DEEP Q LEARNING LOSS FUNCTION

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s') \sim \mathrm{U}(D)} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right)^2 \right]$$

D – Set of experiences (state s, action a, reward r, next action a')

U(D) – Draw random (uniformly) sample from D

-> Experience replay

γ – Discount factor

θ_i – 'Network' to learn in time i. Updated every iteration

θ_{-i} – 'Network' to calculate target in time i. Updated periodically

Loss of the Network is MSE of

instant reward + future quality *calculated in fixed network* * certain value

- quality calculated in current network

using randomly drawn sample from experience set

Method note : Same ε-greedy can be used.

PLAY ATARI GAMES WITH IT!

- Tested with 49 Atari 2600 games
  - Including infamous 'Breakout'

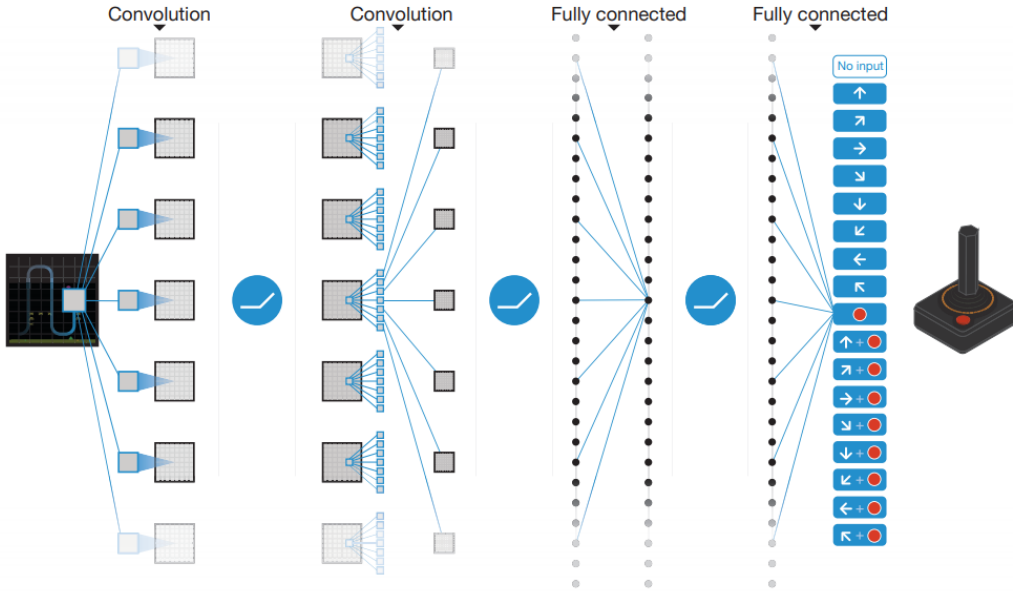
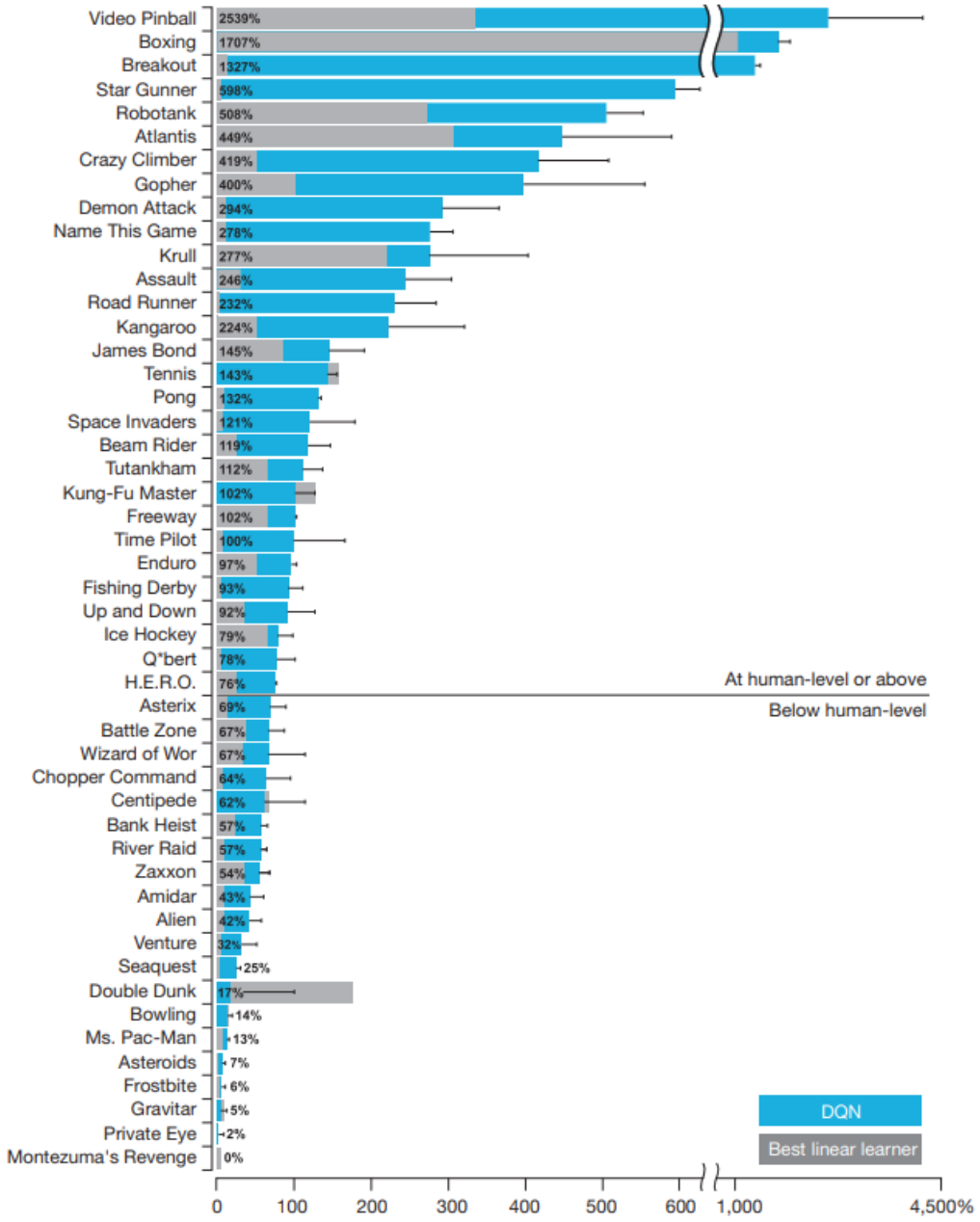


Image taken from paper

RESULTS


Video Pinball 2539%
Boxing 1707%
Breakout 1327%
Star Gunner 598%
Robotank 508%
Atlantis 449%
Crazy Climber 419%
Gopher 400%
Demon Attack 294%
Name This Game 278%
Krull 277%
Assault 246%
Road Runner 232%
Kangaroo 224%
James Bond 145%
Tennis 143%
Pong 132%
Space Invaders 121%
Beam Rider 119%
Tutankham 112%
Kung-Fu Master 102%
Freeway 102%
Time Pilot 100%
Enduro 97%
Fishing Derby 93%
Up and Down 92%
Ice Hockey 79%
Q*bert 78%
H.E.R.O. 76%
Asterix 69%
Battle Zone 67%
Wizard of Wor 67%
Chopper Command 64%
Centipede 62%
Bank Heist 57%
River Raid 57%
Zaxxon 54%
Amidar 43%
Alien 42%
Venture 32%
Seaquest 25%
Double Dunk 17%
Bowling 14%
Ms. Pac-Man 13%
Asteroids 7%
Frostbite 6%
Gravitar 5%
Private Eye 2%
Montezuma's Revenge 0%
At human-level or above
Below human-level
DQN
Best linear learner
0
100
200
300
400
500
600
1,000
4,500%

BONUS : WITH/WITHOUT REPLAY/TARGET Q

Extended Data Table 3 | The effects of replay and separating the target Q-network

Game	With replay, with target Q	With replay, without target Q	Without replay, with target Q	Without replay, without target Q
Breakout	316.8	240.7	10.2	3.2
Enduro	1006.3	831.4	141.9	29.1
River Raid	7446.6	4102.8	2867.7	1453.0
Seaquest	2894.4	822.6	1003.0	275.8
Space Invaders	1088.9	826.3	373.2	302.0

DQN agents were trained for 10 million frames using standard hyperparameters for all possible combinations of turning replay on or off, using or not using a separate target Q-network, and three different learning rates. Each agent was evaluated every 250,000 training frames for 135,000 validation frames and the highest average episode score is reported. Note that these evaluation episodes were not truncated at 5 min leading to higher scores on Enduro than the ones reported in Extended Data Table 2. Note also that the number of training frames was shorter (10 million frames) as compared to the main results presented in Extended Data Table 2 (50 million frames).

THANK YOU FOR LISTENING

HAI paper study