

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_{t+1} + \lambda \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

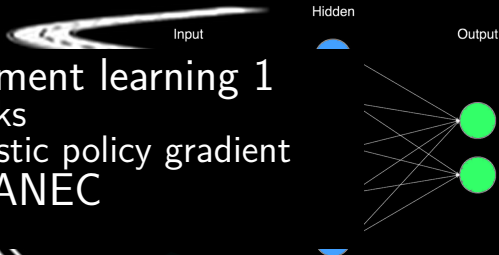
(The New Action Value = The Old Value) + The Learning Rate \times (The New Information - the Old Information)



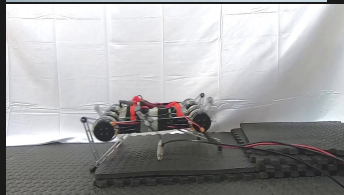
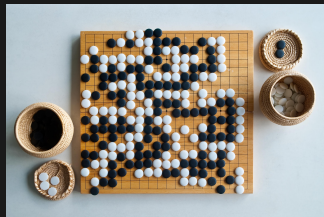
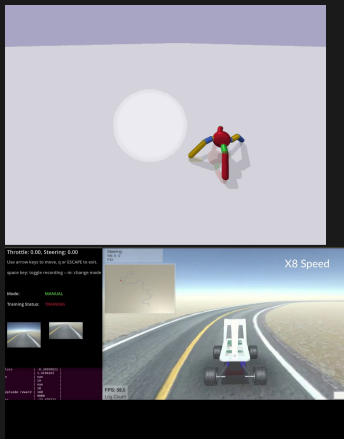
deep reinforcement learning 1

- deep Q networks
- deep deterministic policy gradient

Michal CHOVANEC

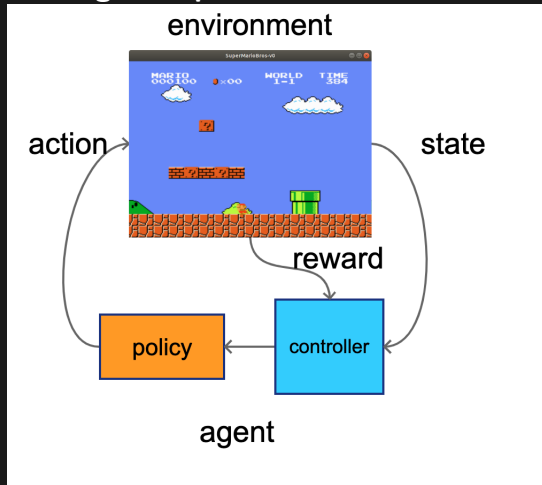


reinforcement learning

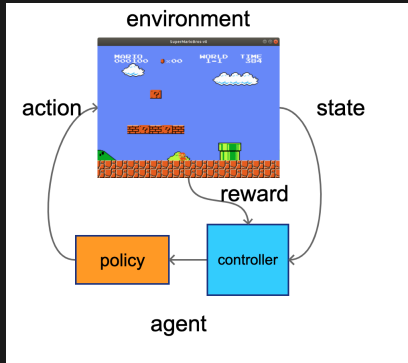


reinforcement learning

learning from punishments and rewards



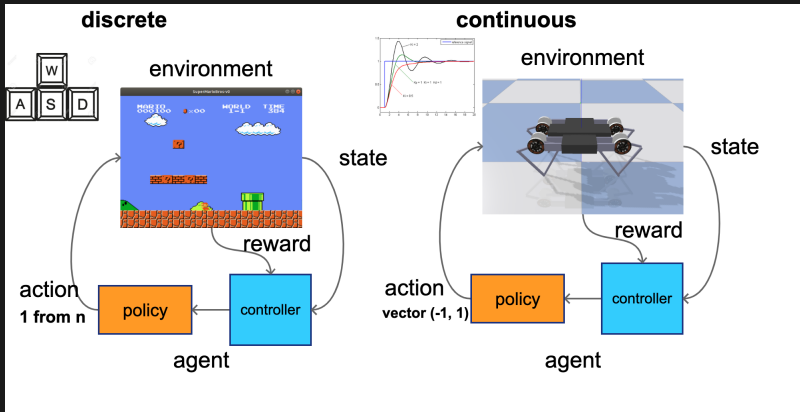
reinforcement learning



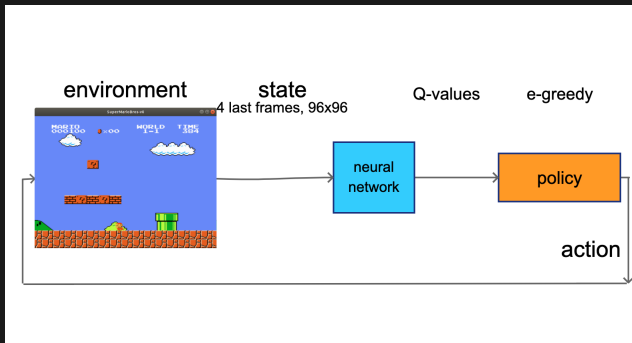
- obtain state
- select action
- execute action
- learn from experiences

action space

- discrete action space
 - keys, keypad
- continuous action space
 - motors, PWMs, steering, force control



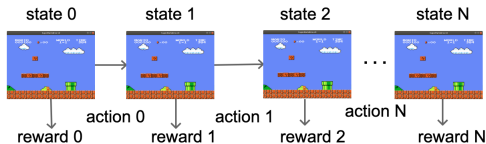
deep Q learning



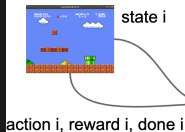
- ① play games
- ② store transitions into buffer
 - state, action, reward, done
- ③ learn from buffer

deep Q learning

1, game play



2, experience replay buffer

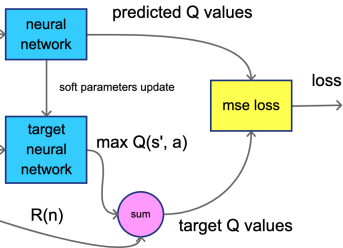


random sample
fifo buffer, 32768 samples

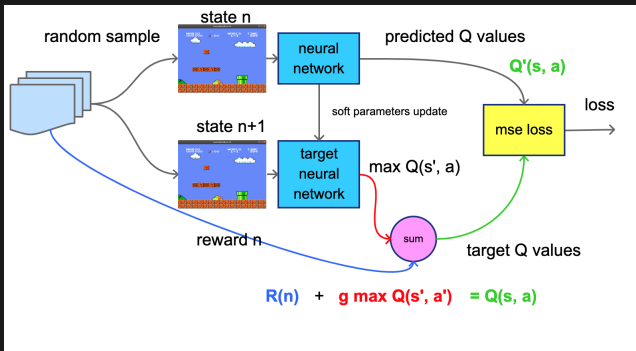


reward n

3, train network



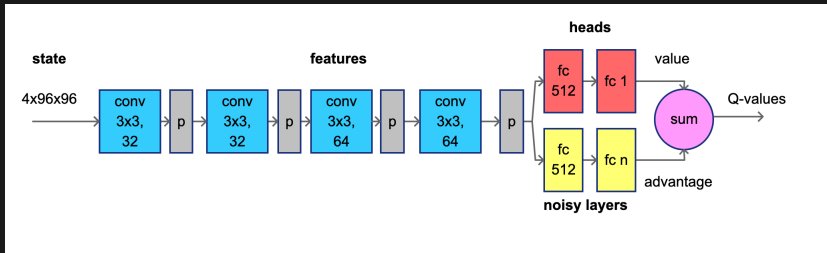
deep Q learning



$$Q(s, a; \theta) = \underset{\text{reward}}{R} + \underset{\text{discounted future reward}}{\gamma \max_{a'} Q(s', a'; \theta^-)}$$

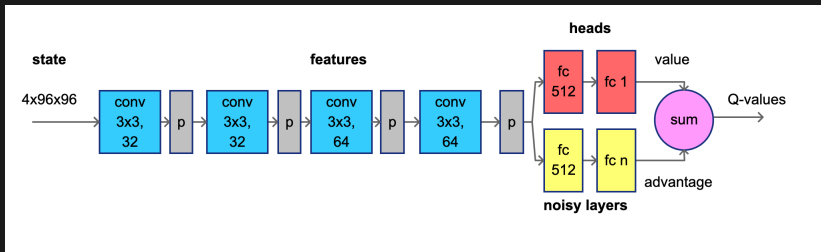
$$\mathcal{L}(\theta) = \left(R + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2$$

model architecture



- input 96x96 grayscale, 4 stacked frames
- 3x3 convs + pooling
- two fully connected layers
- small learning rate $\eta = 0.0001$, batch size = 32
- $\gamma = 0.99$
- exploration ϵ -greedy, 1M samples linear decay from 1 to 0.05
- total training 10M samples

dueling DQN, model architecture



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

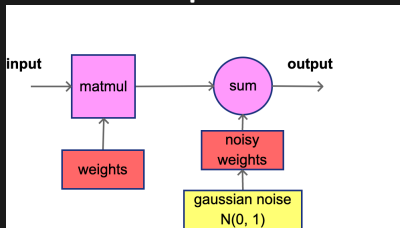
$$Q(s, a) = V(s) + A(s, a) - \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}} A(s, a')$$

WRONG : `q = value + advantage - advantage.mean()`

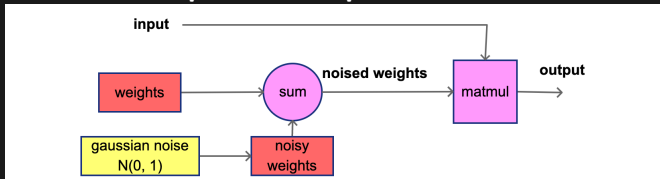
CORRECT : `a = value + advantage - advantage.mean(dim=1, keepdim=True)`

noisy layers for exploration

action space noise



parameter space noise



books to read

- Maxim Lapan, 2020, Deep Reinforcement Learning Hands-On second edition
- Maxim Lapan, 2018, Deep Reinforcement Learning Hands-On
- Praveen Palanisamy, 2018, Hands-On Intelligent Agents with OpenAI Gym
- Andrea Lonza, 2019, Reinforcement Learning Algorithms with Python
- Rajalingappaa Shanmugamani, 2019, Python Reinforcement Learning
- Micheal Lanham, 2019, Hands-On Deep Learning for Games

Q&A



`michal.nand@gmail.com`

`https://github.com/michalnand/imagination_reinforcement_learning`