Deep Reinforcement Learning

Junling Hu

Feb 1, 2017

SV AI and Big Data Association

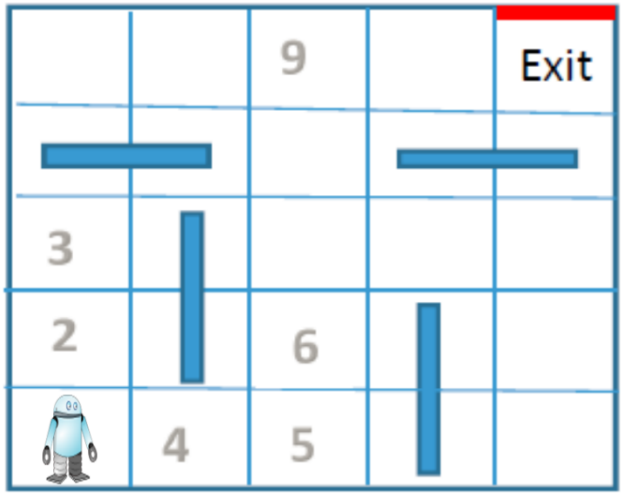
What is Deep Reinforcement Learning

- Deep Learning + Reinforcement Learning
  - Deep Learning

    Capture the complex state
  - Reinforcement Learning

    Discover the best action

Reinforcement Learning Concepts



- State (s): position (indexed by 1, 2, …)
- Action (a): up, down, right, left
- Reward (r): 0, -1, 1
- Long-term Reward Q: Reach Exit

Basic Concepts of Reinforcement Learning

- Q-values: Long-term Rewards

  Q(s,a)

  =r(s,a) + max_a Q(s’,a)

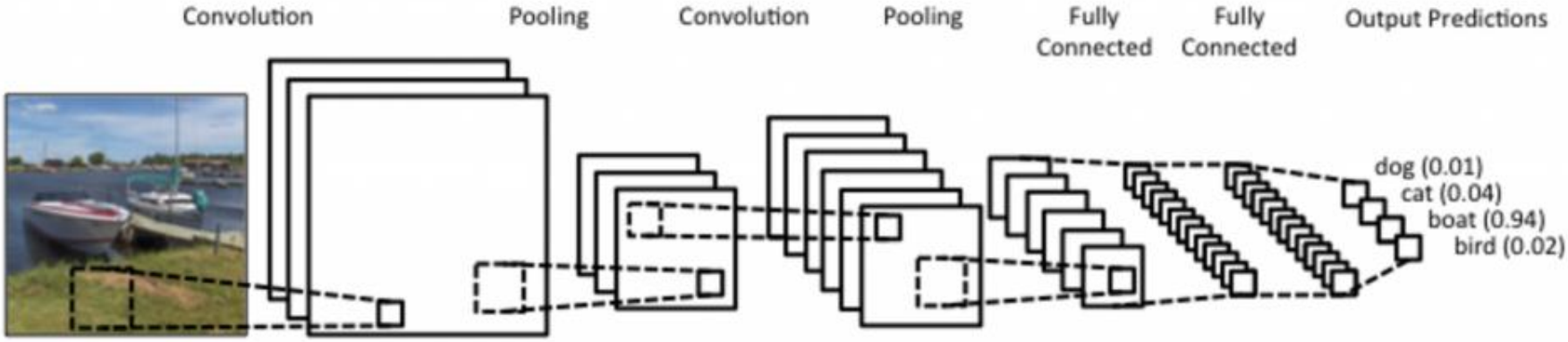
- Policy
  - The right action given a state
  - Typically means a sequence of actions for reaching a goal

Example of Deep RL: AlphaGo

- Complex board situation
  - 10^170 (total atoms in the Universe: 10^68)

- Use 19x19 picture to represent a board
- Use Convolutional Neural Network to map the board to action

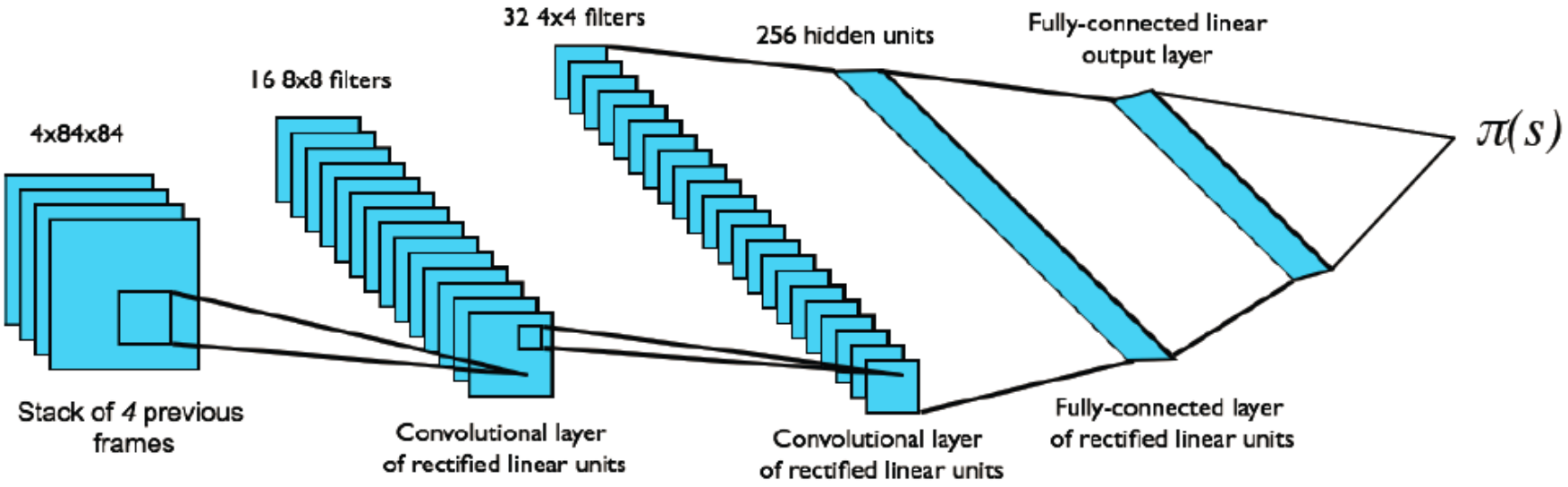
Convolutional Neural Network



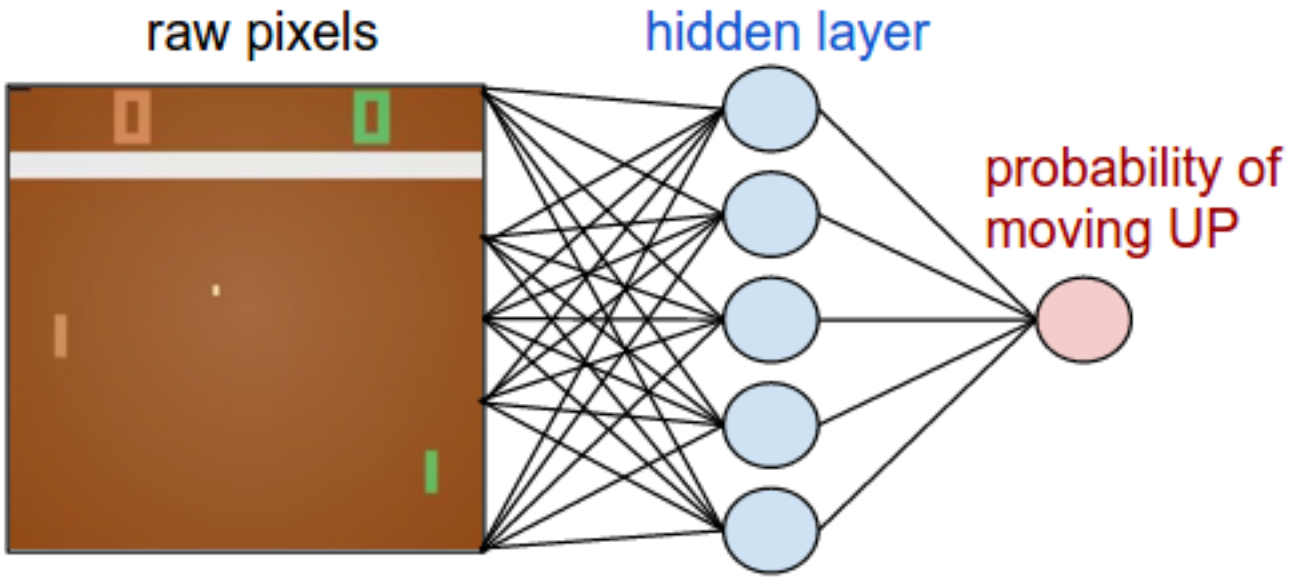
Reinforcement Learning in Go Game

- State: Board situation (tranformed by CNN)
- Action: Where to put the next piece
  - 361 possible actions

Deep Policy Network



Policy network for Game of Pong



Training Deep Policy Network

Use two networks: an actor and a critic

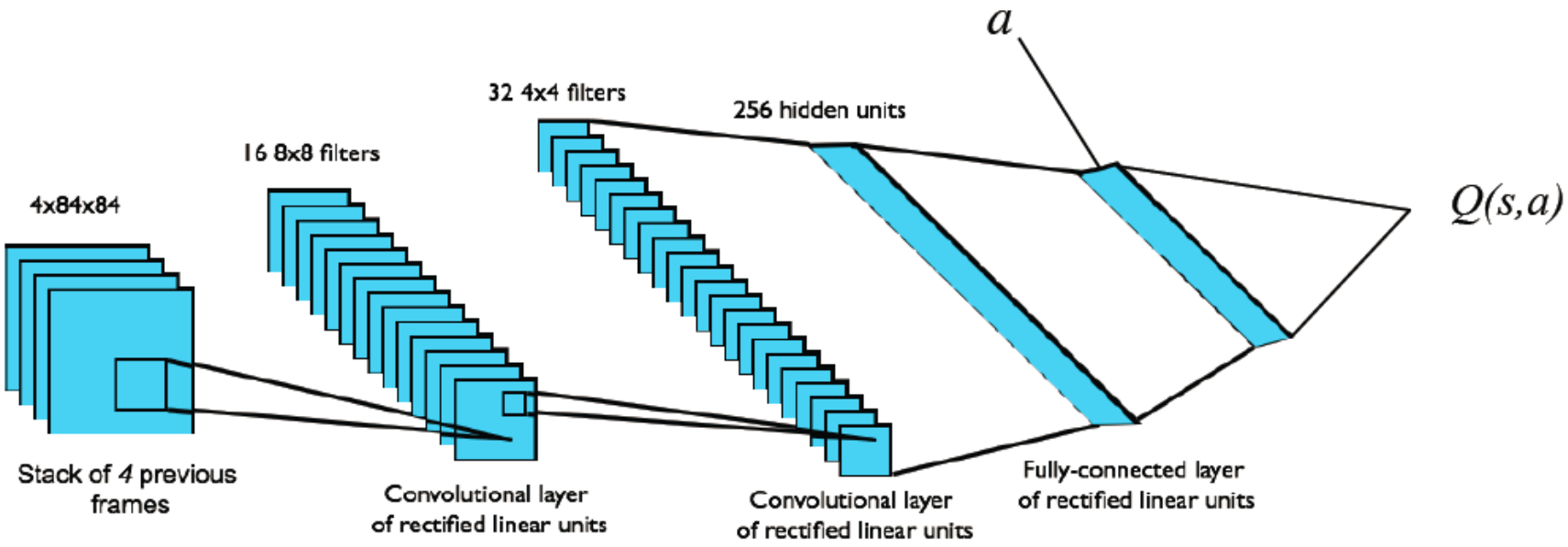
- Critic estimates value of current policy by Q-learning

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \mathbb{E}\left[\left(r + \gamma Q(s', \pi(s'), w) - Q(s, a, w)\right)\frac{\partial Q(s, a, w)}{\partial w}\right]$$

- Actor updates policy in direction that improves Q

$$\frac{\partial J(u)}{\partial u} = \mathbb{E}_s\left[\frac{\partial Q(s, a, w)}{\partial a}\frac{\partial \pi(s, u)}{\partial u}\right]$$

Deep Q-network



Training Deep Q-network

- Represent value function by deep Q-network with weights w

$$Q(s, a, w) \approx Q^{\pi}(s, a)$$

- Define objective function by mean-squared error in Q-values

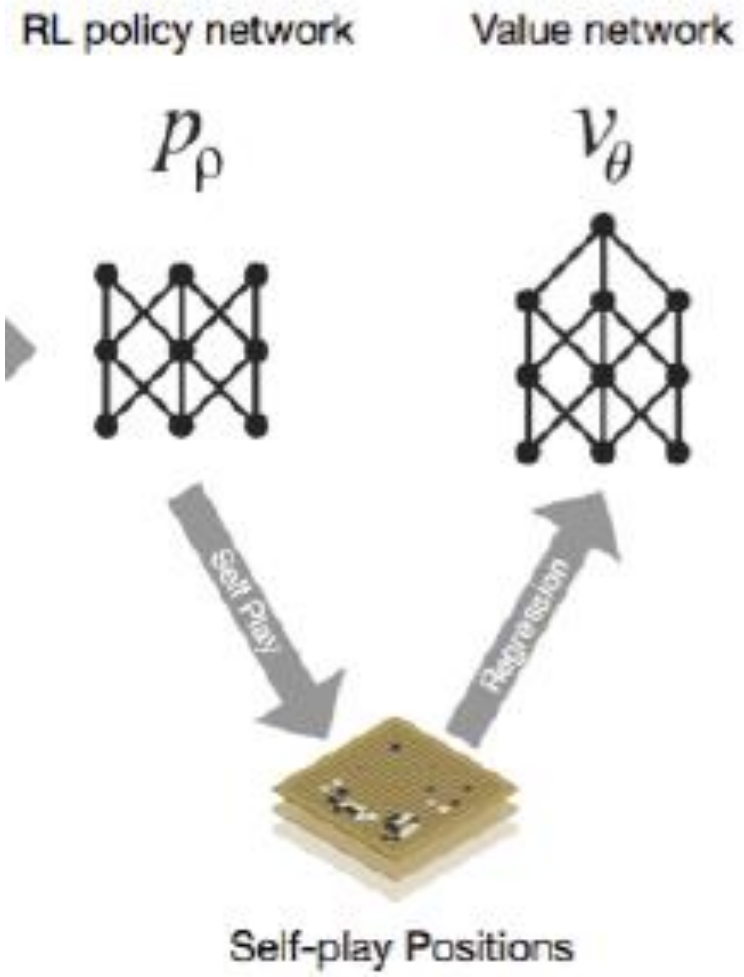
$$\mathcal{L}(w) = \mathbb{E}\left[\left(\underbrace{r + \gamma \max_{a'} Q(s', a', w)}_{\text{target}} - Q(s, a, w)\right)^2\right]$$

- Leading to the following Q-learning gradient

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q(s', a', w) - Q(s, a, w)\right) \frac{\partial Q(s, a, w)}{\partial w}\right]$$

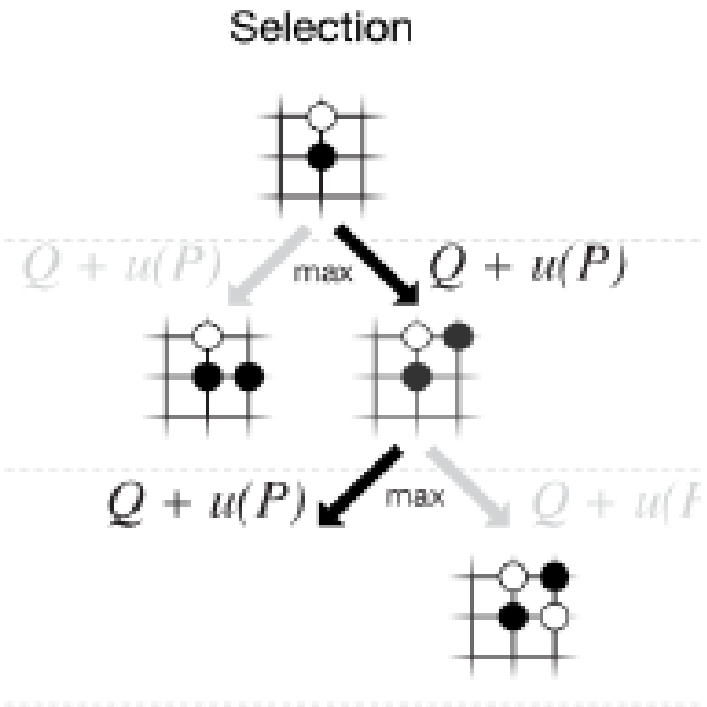
- Optimise objective end-to-end by SGD, using $\frac{\partial L(w)}{\partial w}$

Training for AlphaGo



Monte Carlo Tree Search

- Repeated simulation
- Each simulation traverse a tree, select best Q



Modeling Dialog

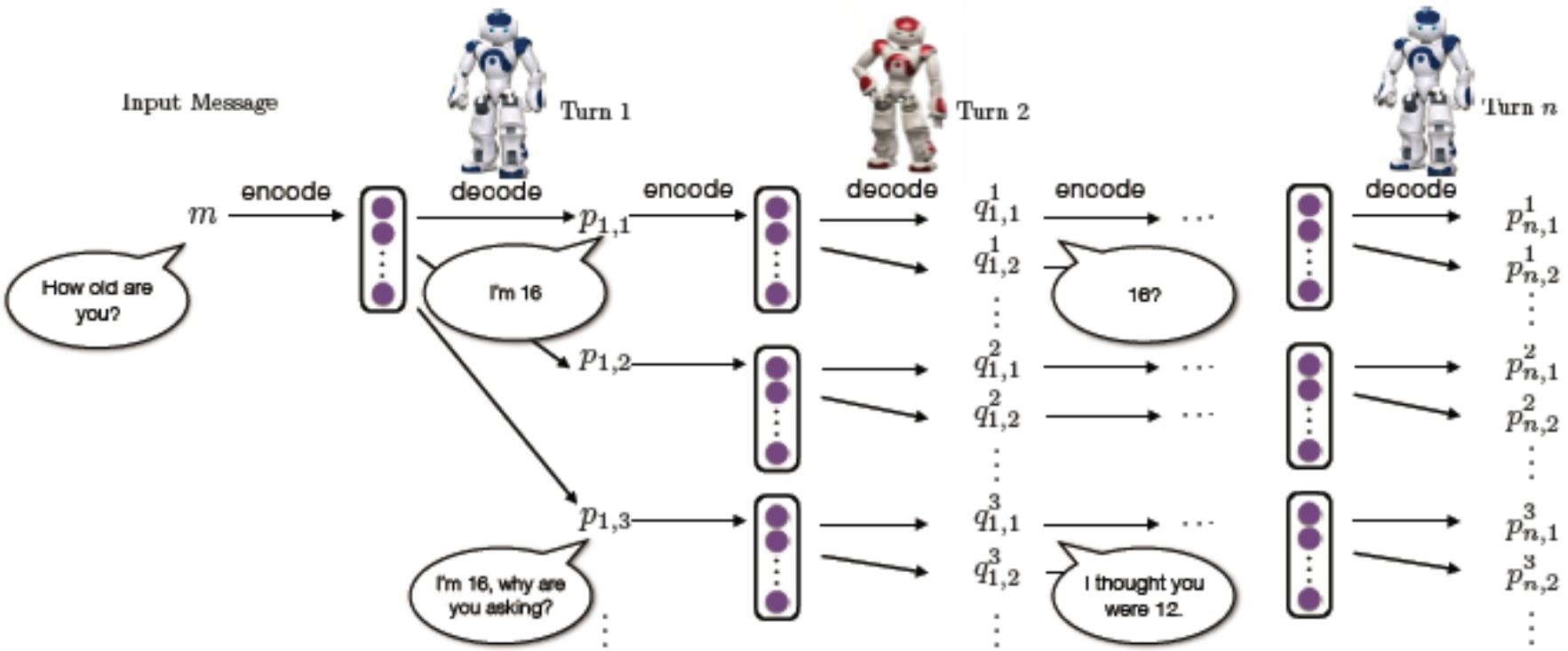


Figure 1: Dialogue simulation between the two agents.

Deep RL for Dialog

- State: previous turns
- Action: What to say
- Reward: Easiness to understand

Result from Deep RL

- Traditional model

A: how old are you? (1)
B: I'm 16. (2)
A: 16? (3)
B: I don't know what you are talking about. (4)
A: You don't know what you are saying. (5)
B: I don't know what you are talking about . (6)
A: You don't know what you are saying. (7)

- Deep RL

A: How old are you? (1)
B: I'm 16. Why are you asking? (2)
A I thought you were 12. (3)
B: What made you think so? (4)
A: I don't know what you are talking about. (5)
B: You don't know what you are saying. (6)
...

Contact

Junling Hu
junlinghu@gmail.com

Wechat: junlingmv

Training Deep Policy Network

- Represent deterministic policy by deep network $a = \pi(s, u)$ with weights u
- Define objective function as total discounted reward

$$J(u) = \mathbb{E}\left[r_1 + \gamma r_2 + \gamma^2 r_3 + ...\right]$$

- Optimise objective end-to-end by SGD

$$\frac{\partial J(u)}{\partial u} = \mathbb{E}_s\left[\frac{\partial Q^\pi(s,a)}{\partial a}\frac{\partial \pi(s,u)}{\partial u}\right]$$

  - Update policy in the direction that most improves Q
  - i.e. Backpropagate critic through actor