Reinforcement Learning:
Not Just for Robots and Games

Jibin Liu 

Joint work with Giles Brown, Priya Venkateshan, Yabei Wu, and Heidi Lyons

***Fig 1**. AlphaGo. Source: deepmind.com*

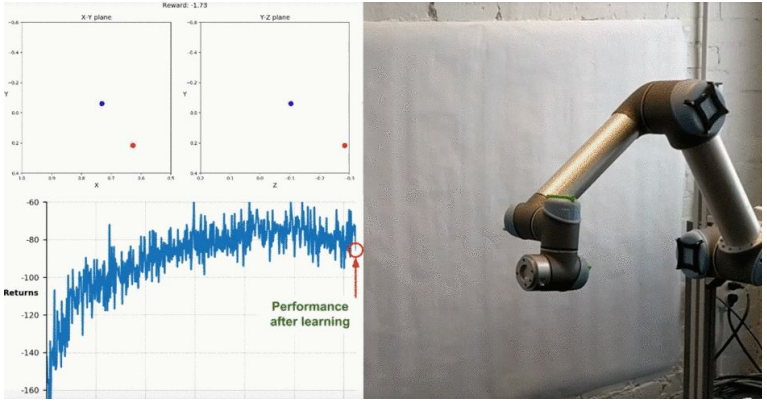


***Fig 3**. Training robotic arm to reach target locations in the real world. Source: Kindred AI*

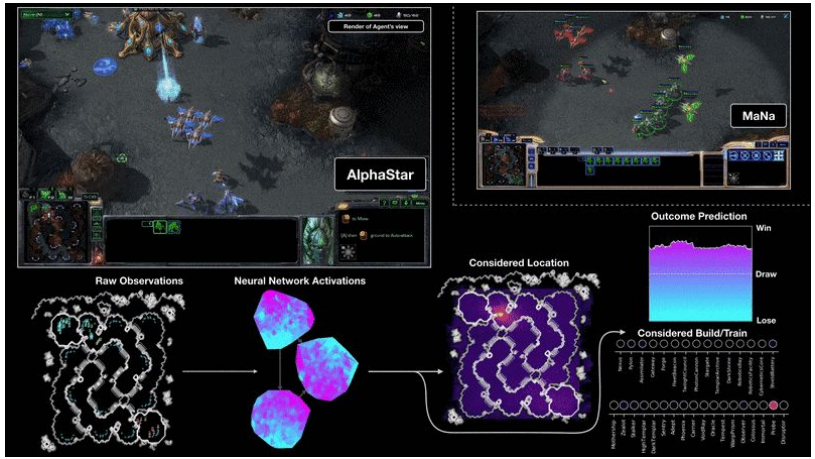
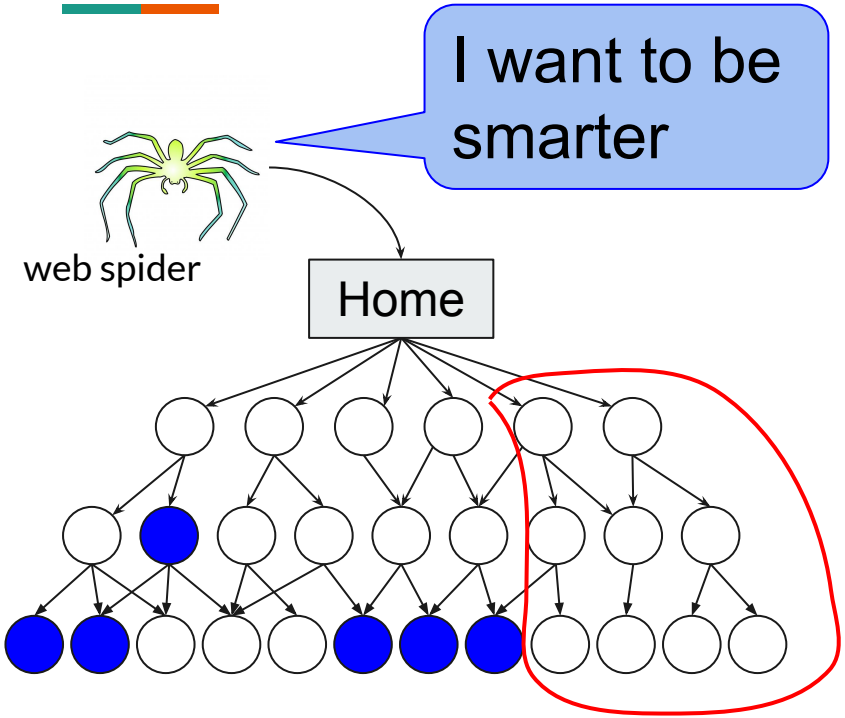


***Fig 2**. A visualisation of the AlphaStar agent during game two of the match against MaNa. Source: deepmind.com*

***Fig 4**. A visualisation of how the OpenAI Five agent is making value prediction. Source: openai.com*

Where traditional web crawling fails



- In traditional web crawling, if we only want the targeted pages, we waste the time and bandwidth when crawling unnecessary pages (e.g., in red box)

In this talk:

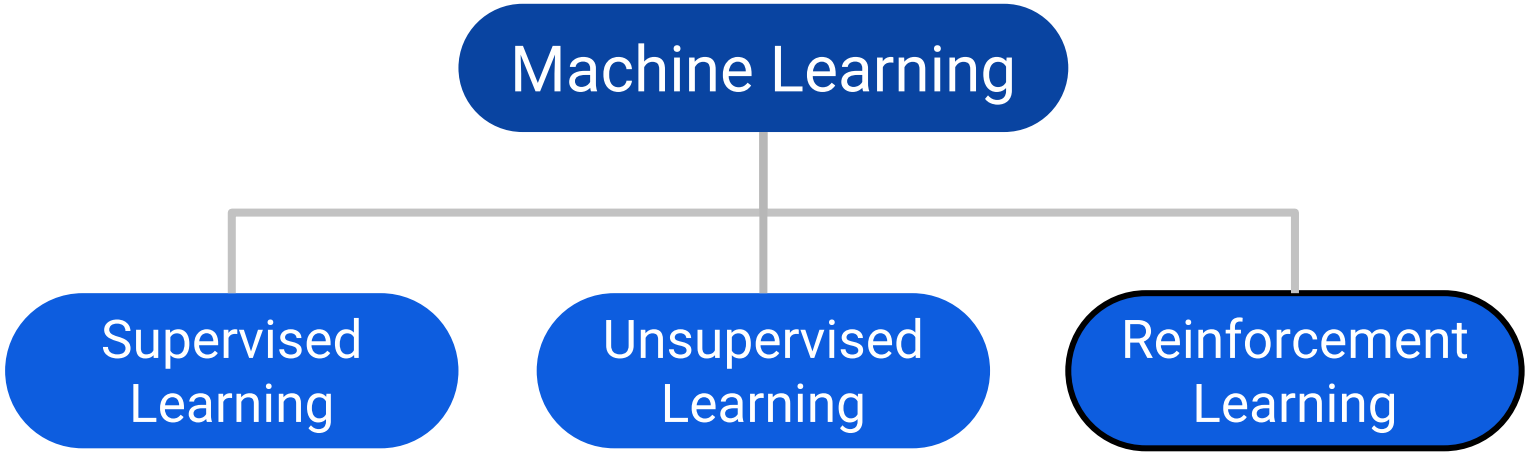
- Reinforcement Learning Demystified

- Reinforcement Learning In Web Crawling

- What Could Reinforcement Learning Work For Me

Reinforcement Learning Demystified

Where RL sits in Machine Learning



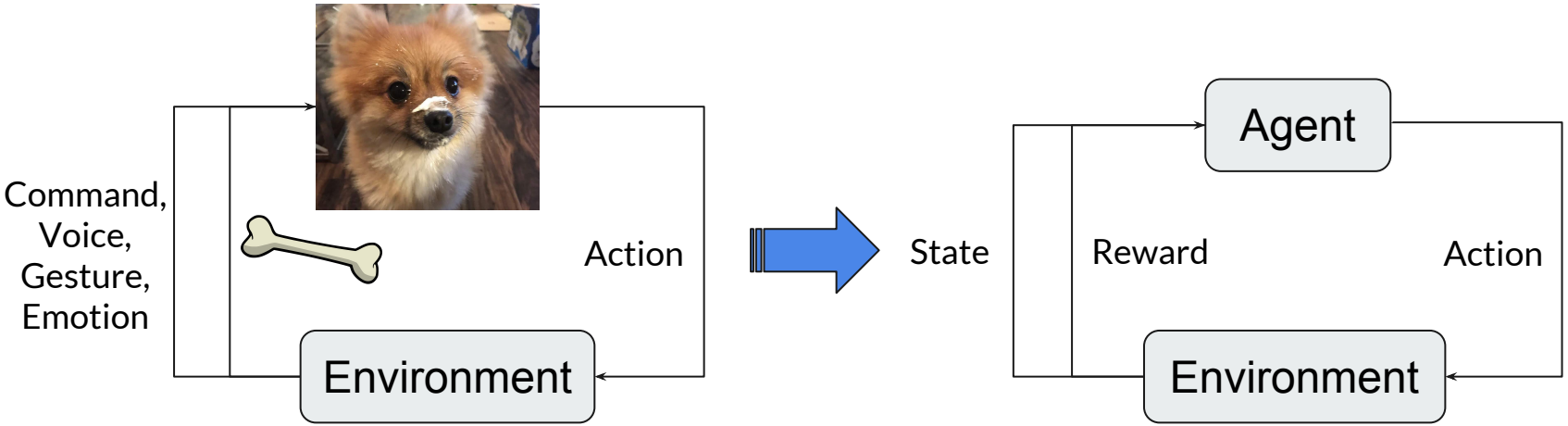
Reinforcement Learning Demystified

Example of dog training

Fig 5. *Dog sit. Source: giphy.com*

Reinforcement Learning Demystified

Dog training vs. Reinforcement Learning



Reinforcement Learning Demystified

Elements of RL

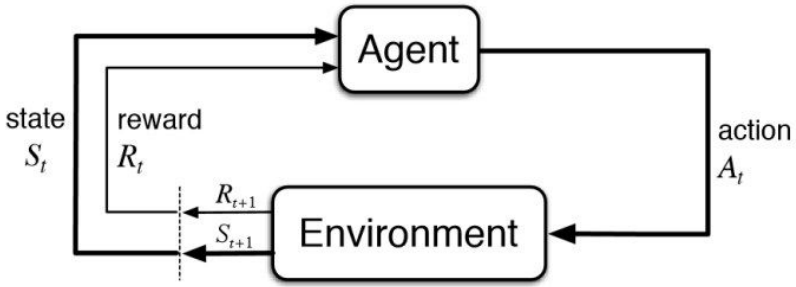


Fig 6. *The agent–environment interaction in a Markov decision process. Source: Reinforcement Learning: An Introduction, second edition*

Experience

$S_t, A_t, R_t, S_{t+1}, A_{t+1} \ldots$

- Policy: mapping from state to action, deterministic or stochastic
- Reward signal: immediate desirability
- Value function: how good the state (state value) or action (action value) is in long term
- Model (optional): model-based vs. model-free solution

Reinforcement Learning Demystified

One way to solve RL problem: Q-learning

- **Action Value**: How good it is, when the agent observes a given state and take an action - $Q(s, a)$

- To update q values (with SARSA sequence), we use:

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

Fig 7. Equation showing how to update q value. Source: wikipedia.com

- To pick an action for a given state, we use **ε-greedy**:
  - trade-off between exploration and exploitation

Reinforcement Learning Demystified

Q-learning code example: update q value

```
def update_Q_using_q_learning(
        Q, state, action, reward,
        new_state, alpha, gamma
    ):
    max_q = max(Q[new_state]) if new_state is not None else 0
    future_return = reward + gamma * max_q
    Q[state][action] += alpha * (future_return - Q[state][action])
    return
```

Reinforcement Learning Demystified

Exploration vs. Exploitation Dilemma

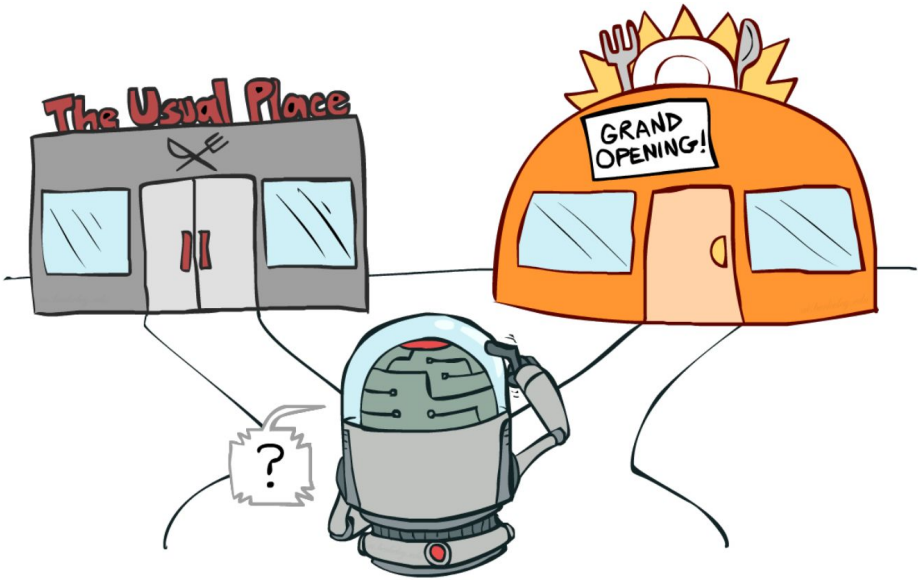


***Fig 8**. A real-life example of the exploration vs exploitation dilemma: where to eat? Source: [UC Berkeley AI course, lecture 11](#).*

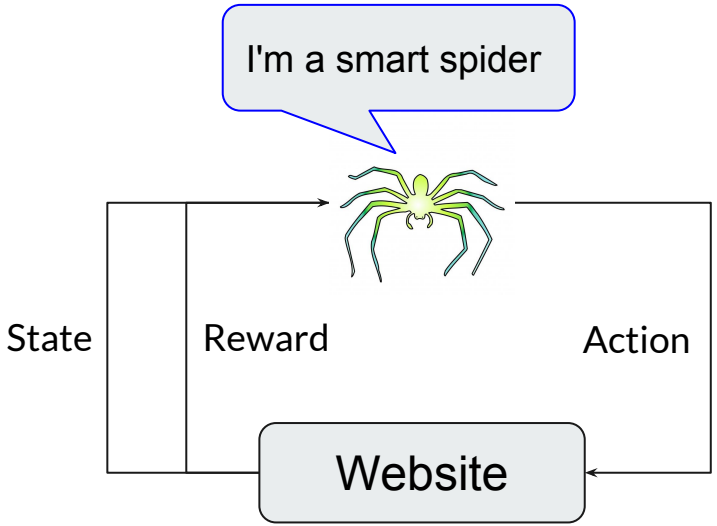
Reinforcement Learning Demystified

Q-learning code example: pick an action using ε-greedy

```
import numpy as np

def pick_action_using_epsilon_greedy(Q, state, epsilon):
    action_values = Q[state]
    if np.random.random_sample() > epsilon:
        return np.argmax(action_values)
    else:
        return np.random.choice(list(range(len(action_values))))
```

RL concepts on web crawling



State: a given web page

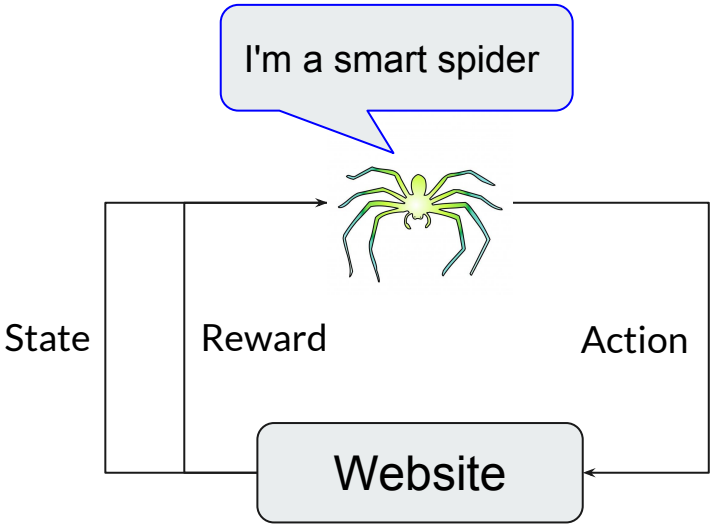
- the url
- links (i.e., actions) in the page
- css/xpath/html attributes of each link

Action: which link to visit next?

Reward: defined by human

- if targeted page: +100
- else: -10

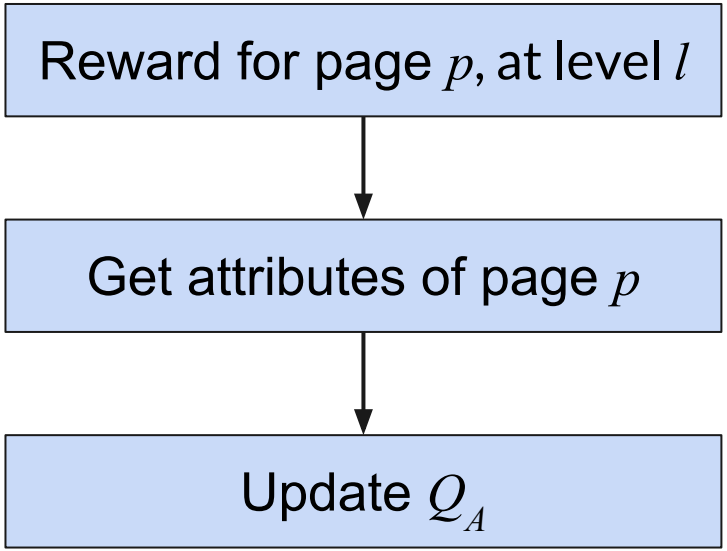
RL concepts on web crawling (cont'd)



Value Functions:

- Q_L(page, link)
  - Value of a given link at a specific web page
- Q_A(level, attribute)
  - Value of a given attribute at a specific level

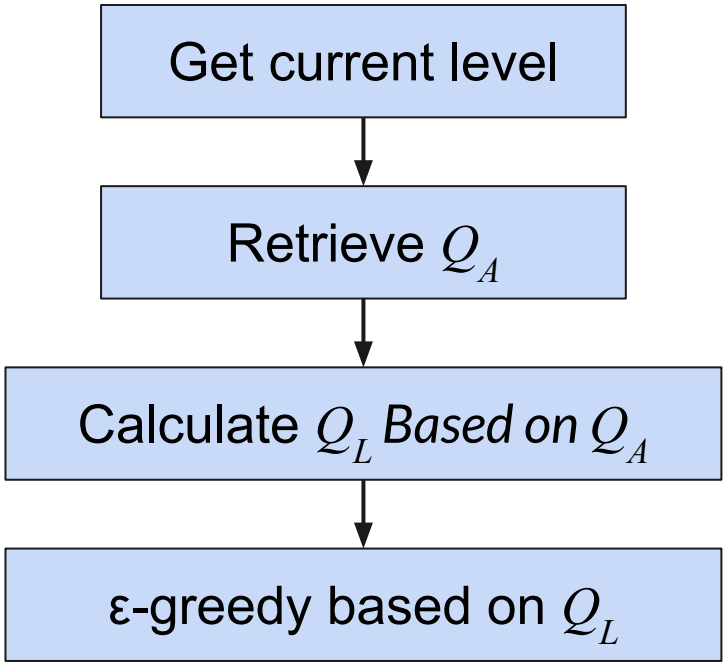
How to update the action values?



Value Functions:

- Q_L(page, link)
  - Value of a given link at a specific web page
- Q_A(level, attribute)
  - Value of a given attribute at a specific level

How to pick the next link?



Value Functions:

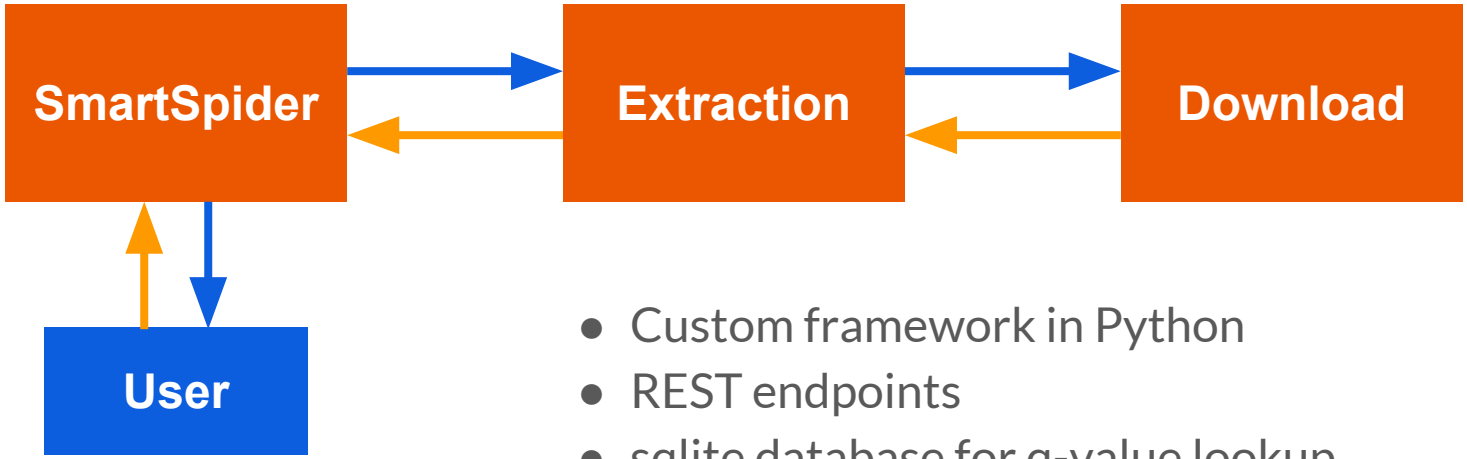
- Q_L(page, link)
  - Value of a given link at a specific web page
- Q_A(level, attribute)
  - Value of a given attribute at a specific level

Results by metrics

	SmartSpider	Traditional Spider
Cumulative rewards	increased and maintained the increase rate after 500 - 1000 episodes	didn't increase but fluctuated around some values (e.g., 0 or 10)
$\frac{\text{\# of target pages}}{\text{\# of total downloads}}$	0.2 - 0.4	< 0.1

Engineering overview

Dockerized App (microservice)



- Custom framework in Python
- REST endpoints
- sqlite database for q-value lookup

What problems RL can help

	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Training	• Labeled data • Direct feedback	• No labels • No feedback	• Trial-and-error search • Reward signal
Usage	• "Best" answer • Prediction	• Find "hidden structure" in data	• Learn actions • Delayed reward • short-term vs long-term

References:

- *Reinforcement Learning: An Introduction, second edition, by by Richard S. Sutton and Andrew G. Barto*
- *Deep Learning Research Review Week 2: Reinforcement Learning, by Adit Deshpande. Original post link*

Learning resources

- Book and Lecture
  - ***Reinforcement Learning: An Introduction, second edition***, by Richard S. Sutton and Andrew G. Barto
  - [UCL] COMPM050/COMPGI13 Reinforcement Learning by David Silver
- Hands-on
  - OpenAI Gym
  - Unity ML-Agents Toolkit

Questions?

Thank you