Deep Reinforcement Learning and Explainable AI (XAI)

IEIE (June 26th, 2019)

Prof. Joongheon Kim
Korea University, Seoul, Korea
https://joongheon.github.io/

Outline

Explainable AI (XAI)

Reinforcement Learning Review

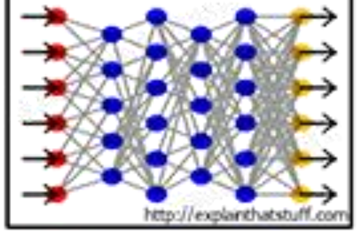
Imitation Learning

Concluding Remarks

- David Gunning (DARPA),
  IJCAI 2016 Workshop Presentation

XAI

AI System



- We are entering a new age of AI applications
- Machine learning is the core technology
- Machine learning models are opaque, non-intuitive, and difficult for people to understand

DoD and non-DoD Applications

Transportation

Security

Medicine

Finance

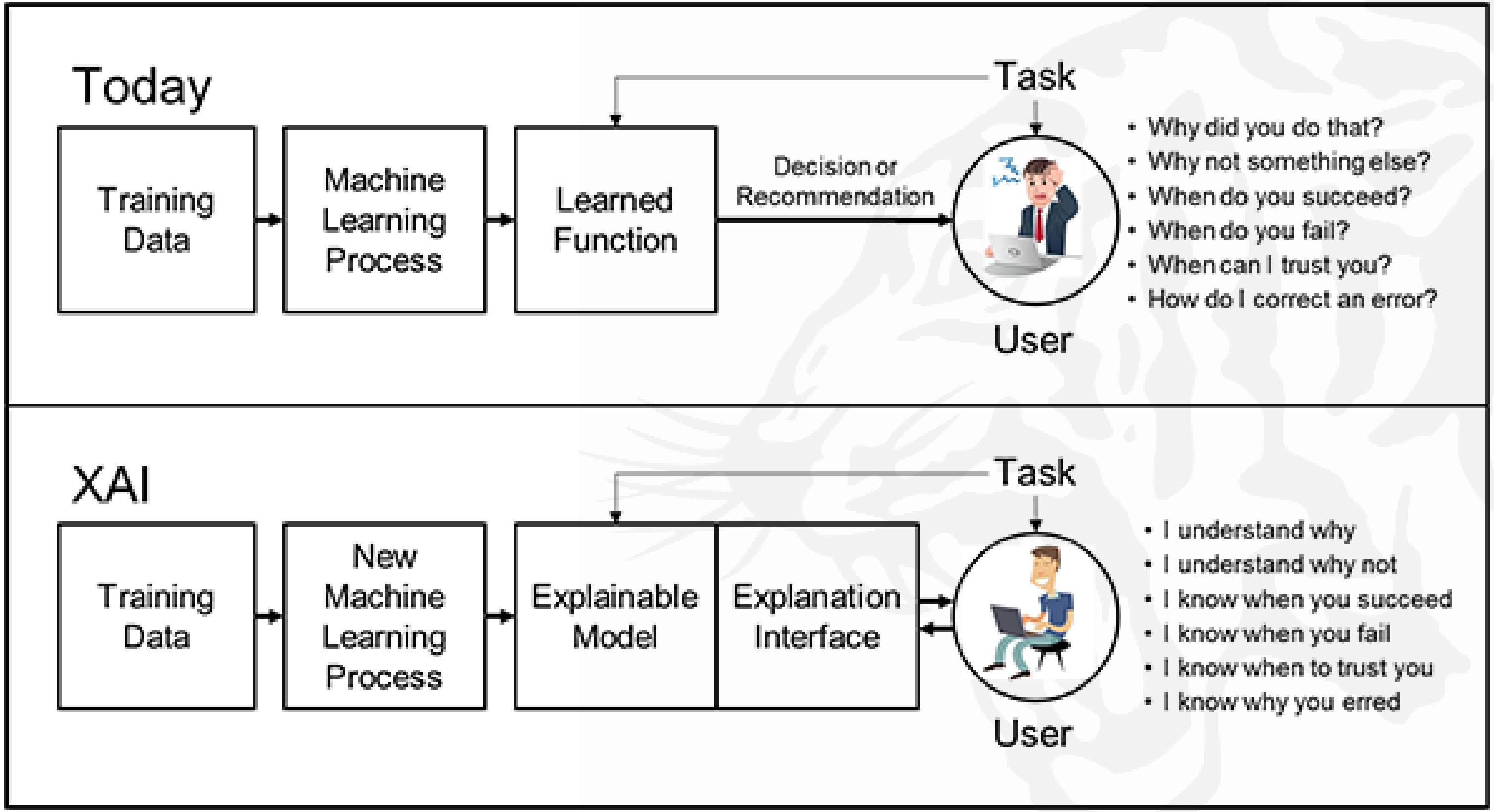
Legal

Military

User

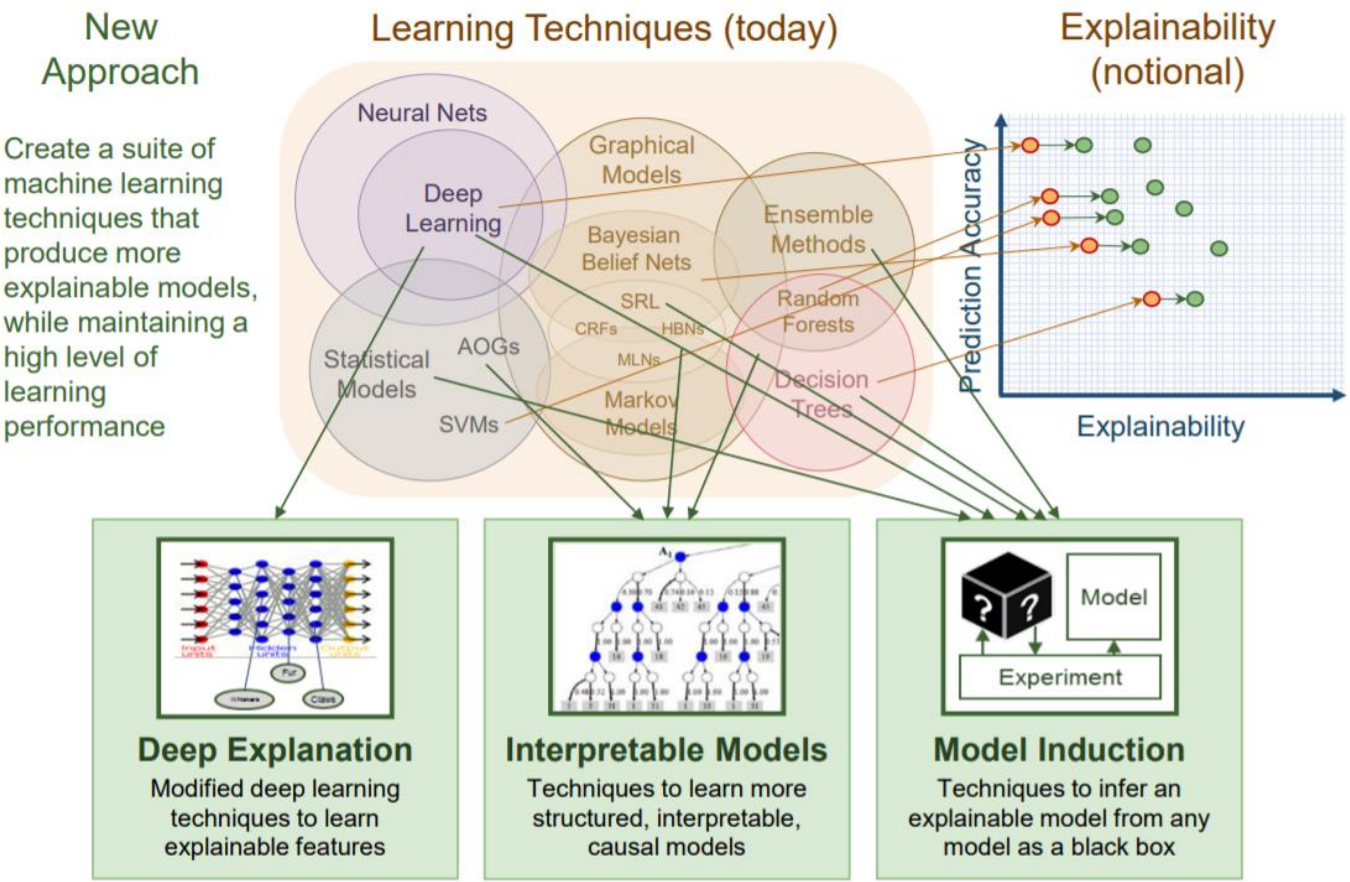
- Why did you do that?
- Why not something else?
- When do you succeed?
- When do you fail?
- When can I trust you?
- How do I correct an error?

XAI Concept

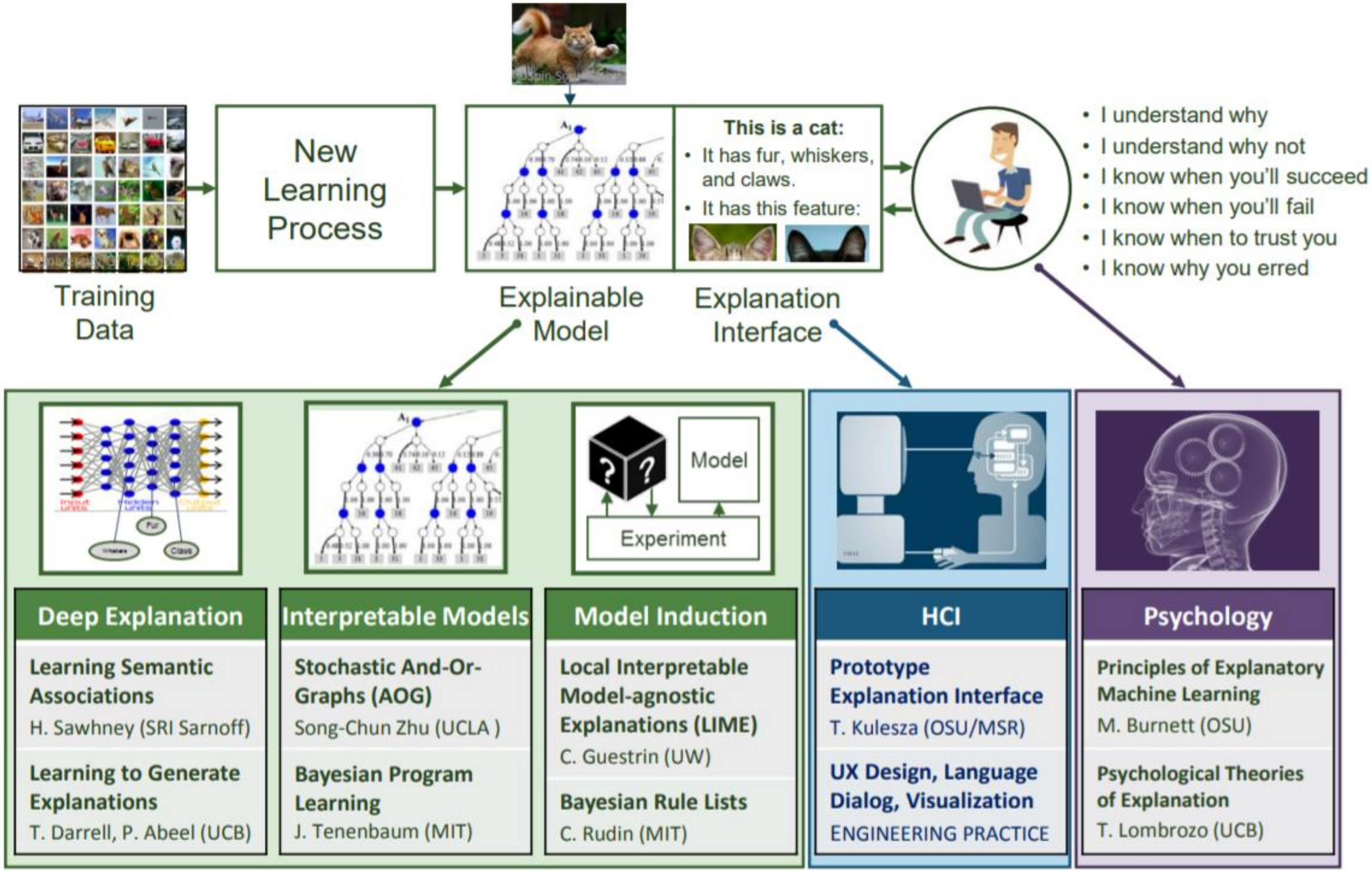


Today
Training Data
Machine Learning Process
Learned Function
Decision or Recommendation
Task
User
• Why did you do that?
• Why not something else?
• When do you succeed?
• When do you fail?
• When can I trust you?
• How do I correct an error?
XAI
Training Data
New Machine Learning Process
Explainable Model
Explanation Interface
Task
User
• I understand why
• I understand why not
• I know when you succeed
• I know when you fail
• I know when to trust you
• I know why you erred

XAI: Performance vs. Explainability



Why Do You Think It Will Be Successful



Outline

Explainable AI (XAI)

Reinforcement Learning Review

- **Introduction**
- Q-Learning
- Deep Q- Network (DQN)

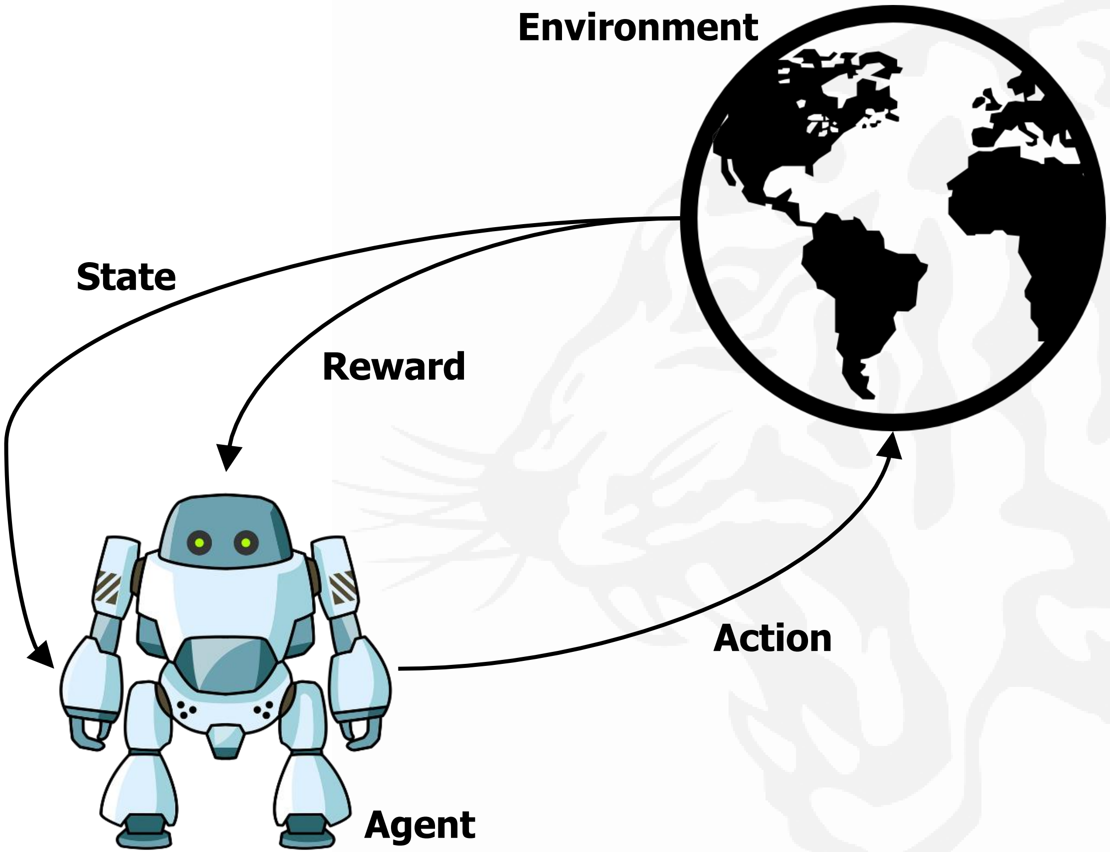
Imitation Learning and Automotive Applications

Concluding Remarks

- Brief History and Successes
  - Minsky's PhD thesis (1954): Stochastic Neural-Analog **Reinforcement** Computer
  - Analogies with animal learning and psychology
  - Job-shop scheduling for NASA space missions (Zhang and Dietterich, 1997)
  - Robotic soccer (Stone and Veloso, 1998) – part of the world-champion approach

- When RL can be used?
  - Find the (approximated) **optimal action sequence** for **expected reward maximization (not for single optimal solution)**
  - Define **actions** and **rewards**. These are all we need to do.

Introduction to RL



Outline

Explainable AI (XAI)

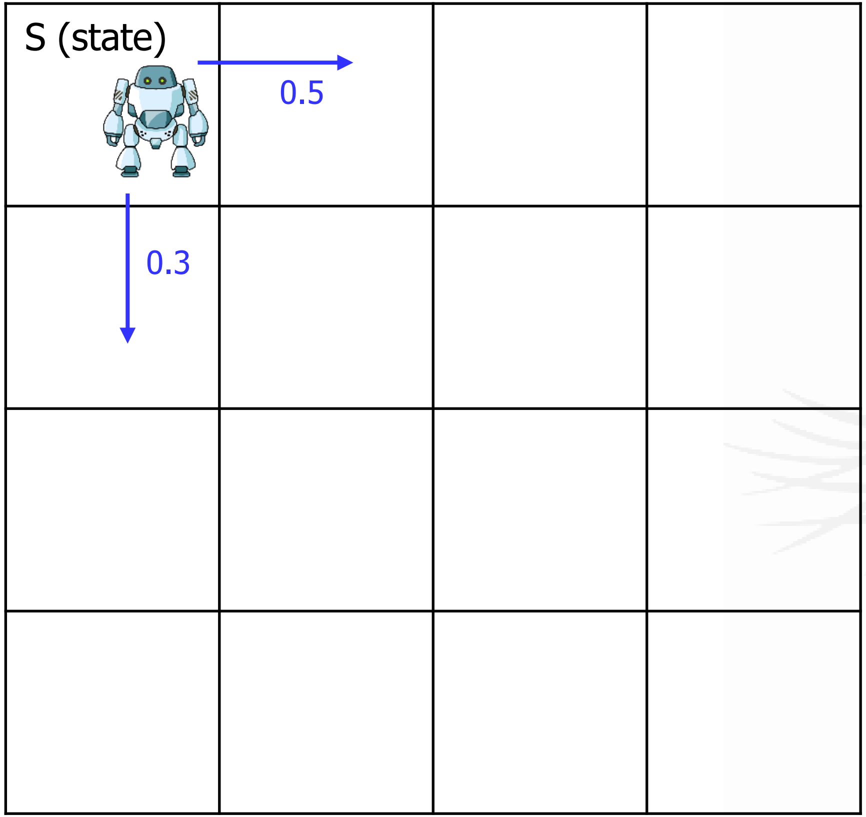
Reinforcement Learning Review

- Introduction
- **<u>Q-Learning</u>**
- Deep Q- Network (DQN)

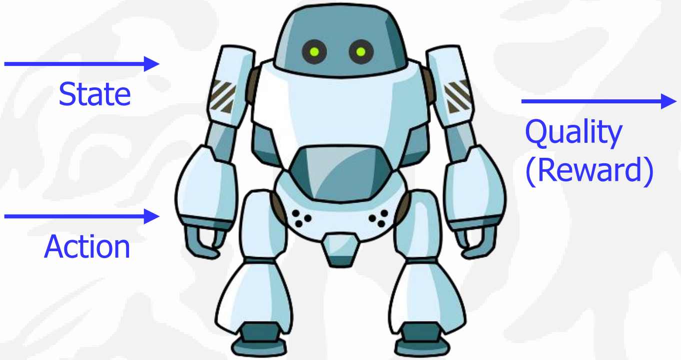
Imitation Learning and Automotive Applications

Concluding Remarks

Quick Example: Q-Learning



- Q-Function
  - State-action value function

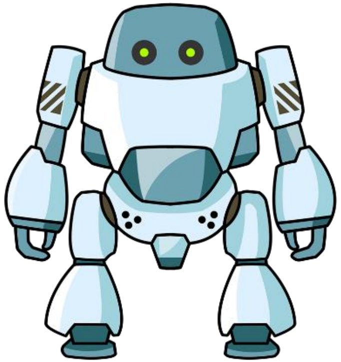


Q (state, action)

Q(s1, LEFT): 0.0
Q(s1, RIGHT): 0.5 → Maximum
Q(s1, UP): 0.0
Q(s1, DOWN): 0.3

$$\text{RIGHT} \leftarrow arg \max_{a \in A} Q(s_1, a)$$

Quick Example: Q-Learning



Q (state, action)

Q(s1, LEFT): 0.0
Q(s1, RIGHT): 0.5 → Maximum
Q(s1, UP): 0.0
Q(s1, DOWN): 0.3

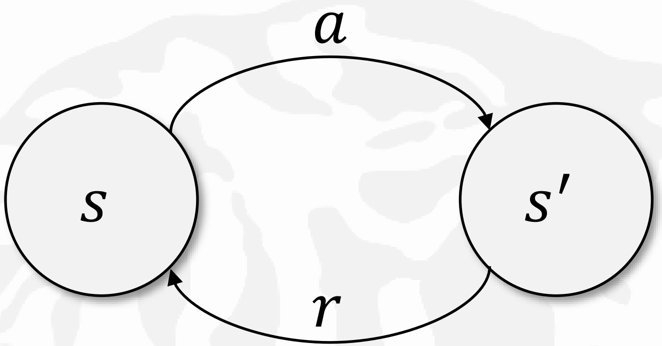
$$\boxed{\text{RIGHT} \leftarrow arg \max_{a \in A} Q(s_1, a)}$$

Optimal Policy π and Max Q

- Max Q = $\max_{a'} Q(s, a')$
- $\pi^*(s) = arg \max_{a} Q(s, a)$

Quick Example: Q-Learning

- My condition
  - I am now in state s
  - When I do action a, I will go to s'.
  - When I do action a, I will get reward r
  - Q in s', it means $Q(s', a')$ exists.



- How can we express $Q(s, a)$ using $Q(s', a')$?

$$Q(s,a) = r + \max_{a'} Q(s', a')$$

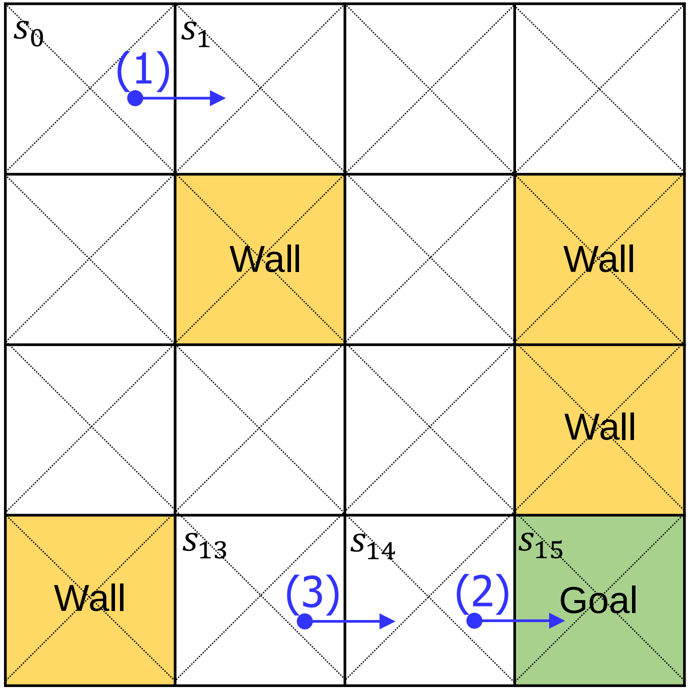
Recurrence (e.g., factorial)

```
F(x){
        if (x != 1){ x * F(x-1) }
        if (x == 1){ F(x) = 1 }
        }
}
```

```
3! = F(3) = 3 * F(2)
          = 3 * 2 * F(1)
          = 3 * 2* 1 = 6
```

Quick Example: Q-Learning

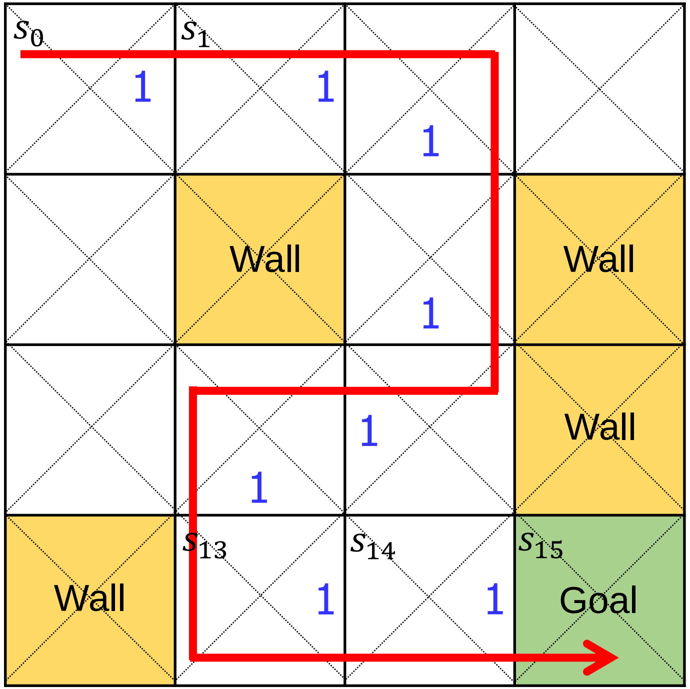
- 16 states and 4 actions (U, D, L, R)



- Initial Status
  - All 64 Q values are 0,
  - Reward are all zero except $r_{s_{15},L} = 1$
- For (1), from s_0 to s_1
  - $Q(s_0, a_R) = r + \max_a Q(s_1, a) = 0 + \max\{0,0,0,0\} = 0$
- For (2), from s_{14} to s_{15} (goal)
  - $Q(s_{14}, a_R) = r + \max_a Q(s_{15}, a) = 1 + \max\{0,0,0,0\} = 1$
- For (3), from s_{13} to s_{14}
  - $Q(s_{13}, a_R) = r + \max_a Q(s_{14}, a) = 0 + \max\{0,0,1,0\} = 1$

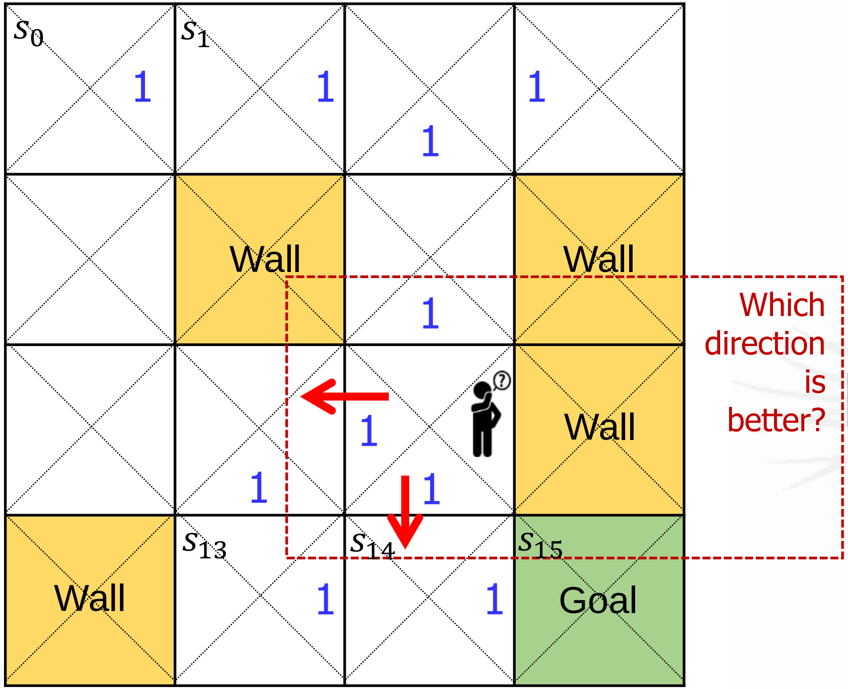
Quick Example: Q-Learning

- 16 states and 4 actions (U, D, L, R)



Quick Example: Q-Learning

- 16 states and 4 actions (U, D, L, R)



Learning $Q(s, a)$ with Discounted Reward

$$Q(s, a) = r + \boxed{\gamma} \cdot \max_{a} Q(s', a')$$

$$0 < \gamma \leq 1$$

Outline

Explainable AI (XAI)

Reinforcement Learning Review

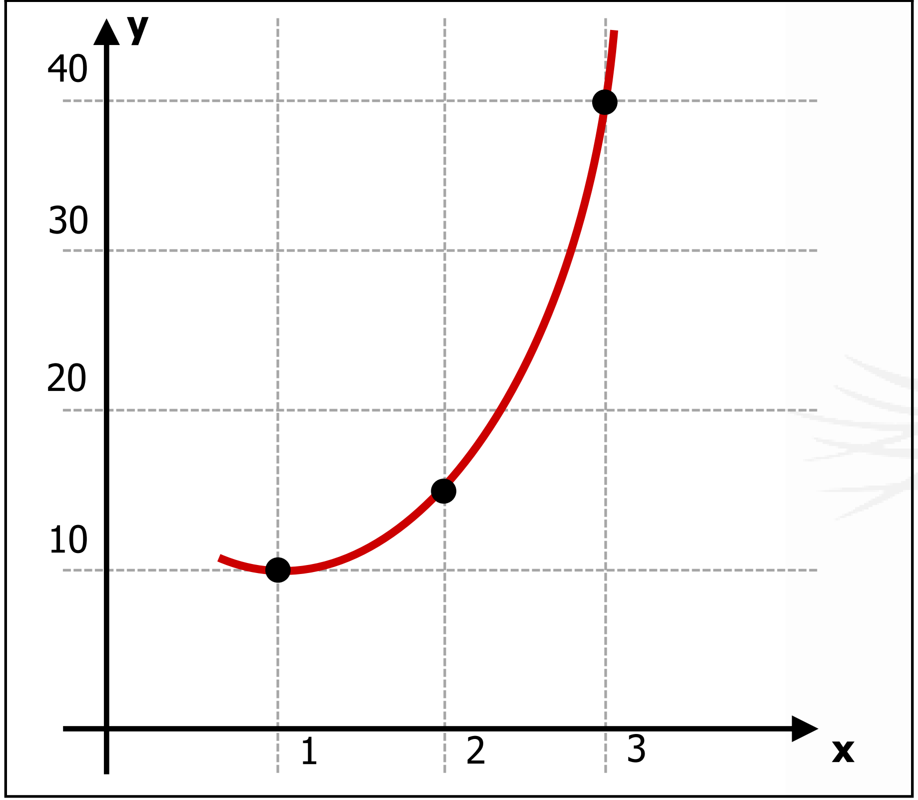
- Introduction
- Q-Learning
- **Deep Q- Network (DQN)**

Imitation Learning

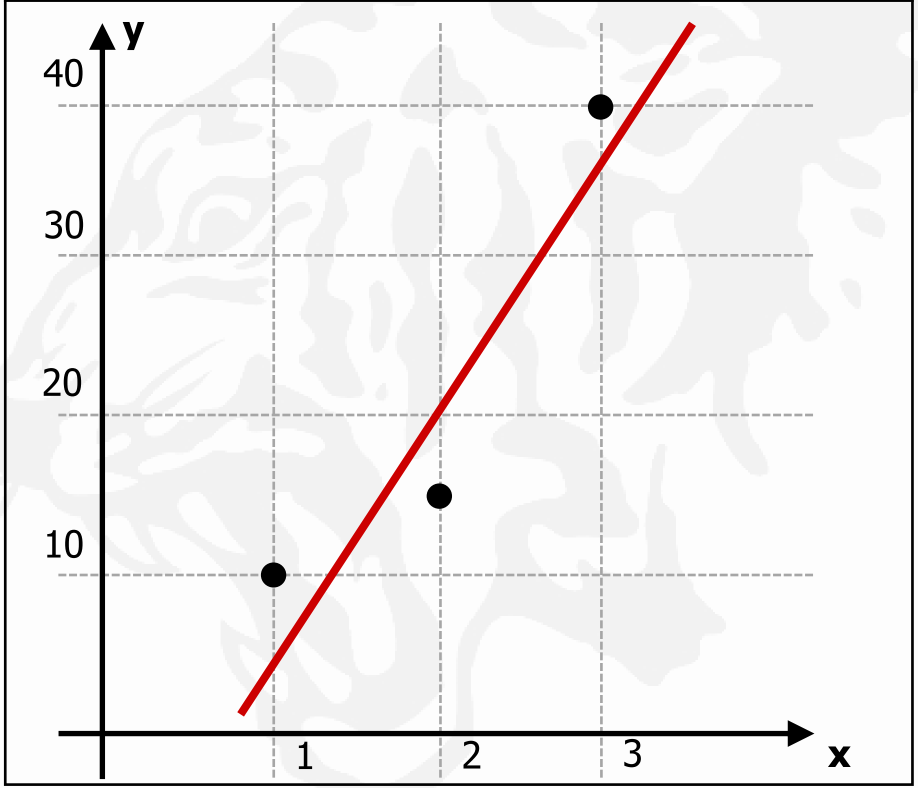
Concluding Remarks

Interpolation vs. Linear Regression

Interpolation

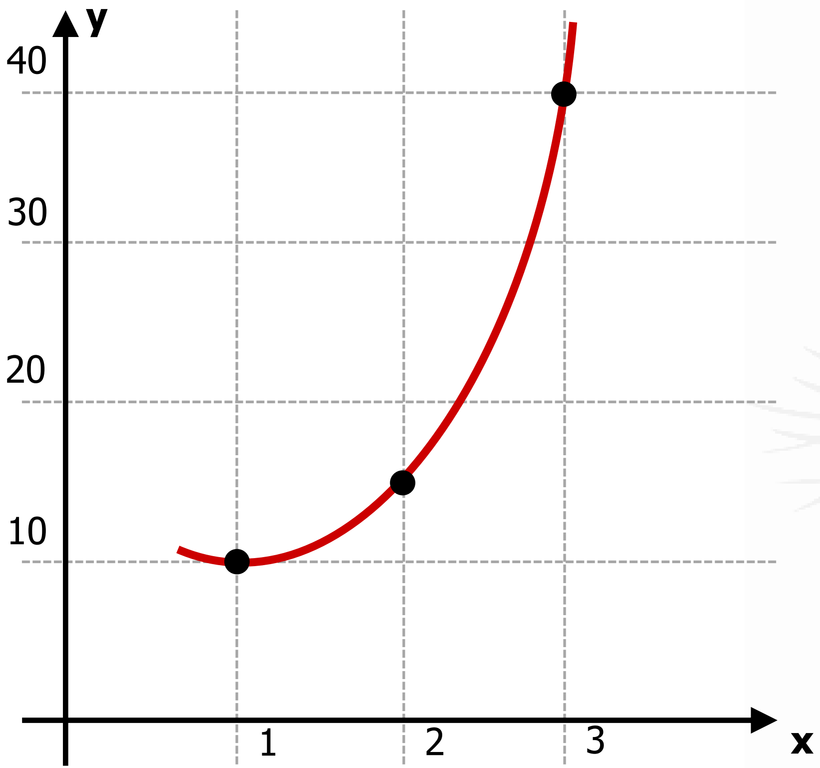


Linear Regression



Interpolation vs. Linear Regression

Interpolation



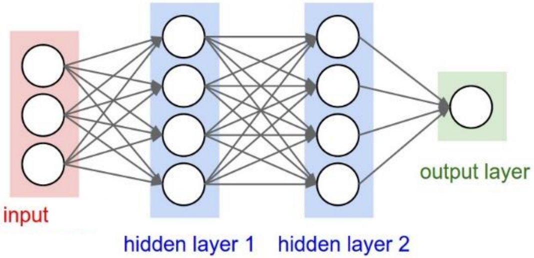
Interpolation with Polynomials

$$y = a_2 x^2 + a_1 x^1 + a_0$$

where three points are given.
→ Unique coefficients (a_0, a_1, a_2) can be calculated.

Is this related to
Neural Network Training?

Interpolation and Neural Network Training



$$Y = a(a(a(X \cdot W_1 + b_1) \cdot W_2 + b_2) \cdot W_o + b_o)$$

where training data/labels (X: data, Y: labels) are given.

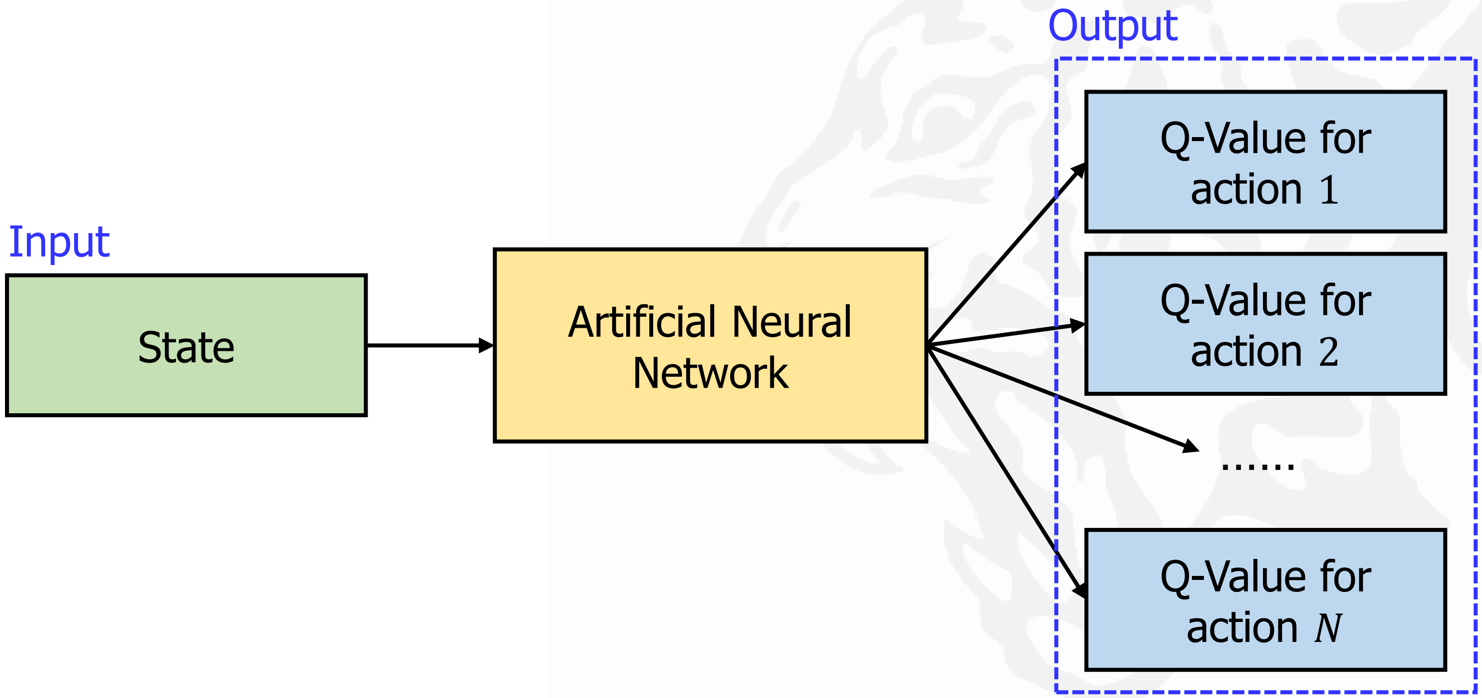
→ Find $W_1, b_1, W_2, b_2, W_o, b_o$

→ This is the mathematical meaning of neural network training.

→ **Function Approximation**

→ The most well-known function approximation with neural network: **Deep Reinforcement Learning**

- Large-Scale Q-Values
  - It is inefficient to make the Q-table for each state-action pair.
    → ANN is used to **approximate the Q-function**.



Outline

Explainable AI (XAI)

Reinforcement Learning Review

Imitation Learning

Concluding Remarks

- ICML 2018 Tutorial
  - https://sites.google.com/view/icml2018-imitation-learning/

Imitation Learning Tutorial ICML 2018

- Gameplay

Pro-Gamer

Trained Agent

The goal of Imitation Learning is to train a policy to mimic **the expert's demonstrations**

- Problems of RL

1. Reward Shaping
2. Safe Learning
3. Exploration process

Imitation Learning **handles with** these problems through the demonstration of the experts.

Introduction to Imitation Learning

- Starcraft2

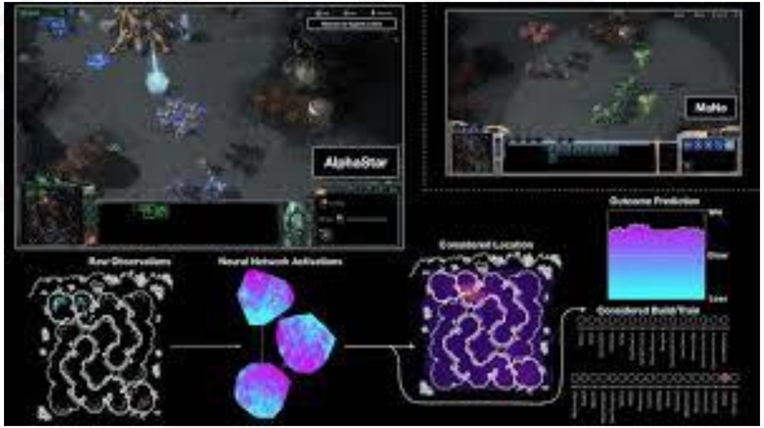
States: s = **minimap**, **screen**
Action: a = **select**, **drag**
Training set: $D = \{\tau \coloneqq (s, a)\}$ from expert
Goal: learn $\pi_\theta(s) \rightarrow$ a

States: s
Action: a
Policy: π_θ
- Policy maps states to actions : $\pi_\theta(\mathrm{s}) \rightarrow \mathrm{a}$
- Distributions over actions : $\pi_\theta(s) \rightarrow P$(a)

State Dynamics: P(s'|s,a)
- Typically not known to policy
- Essentially the simulator/environment

Rollout: sequentially execute $\pi_\theta(s_0)$ on initial state
- Produce trajectories $\boldsymbol{\tau}$

$\mathbf{P}(\boldsymbol{\tau}|\boldsymbol{\pi})$: distribution of trajectories induced by a policy
$\mathbf{P}(\mathbf{s}|\boldsymbol{\pi})$: distribution of states induced by a policy

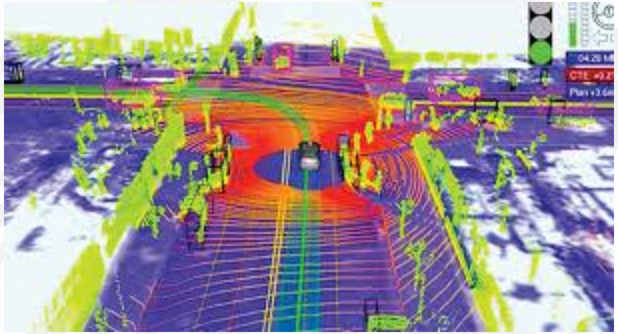


- PPF/RFTN Injection Control in Medicine

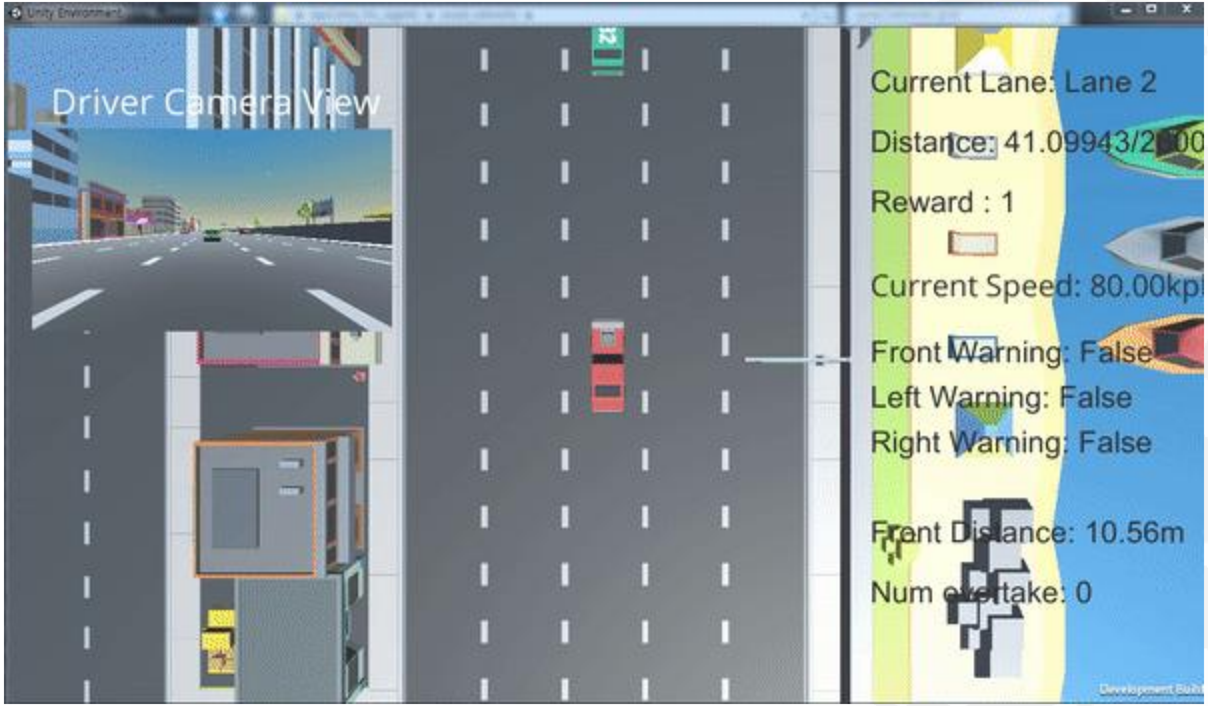
States: s = **BIS**, **BP**, …
Action: a = **PPF**, **RFTN**, …
Training set: $D = \{\tau \coloneqq (s, a)\}$ from expert
Goal: learn $\pi_\theta(s) \rightarrow$ a

- Autonomous Driving Control

States: s = **sensors**
Action: a = **steering wheel**, **brake**, …
Training set: $D = \{\tau \coloneqq (s, a)\}$ from expert
Goal: learn $\pi_\theta(s) \rightarrow$ a



Autonomous Driving with Imitation Learning



Outline

Explainable AI (XAI)

Reinforcement Learning Review

Imitation Learning

Concluding Remarks

Concluding Remarks

- **Explainable AI (XAI)**
- **Reinforcement Learning:** Q-Learning, DQN
- **Imitation Learning:** Reinforcement Learning, Imitation Learning

- Special Thanks to MyungJae Shin (CAU)
- More questions?
  - Joongheon@korea.ac.kr
- More details?
  - https://joongheon.github.io/