Deep Reinforcement Learning: An Overview

Author: Yuxi Li

Presenter: Jiaru Zhang

Report Date: July 19th

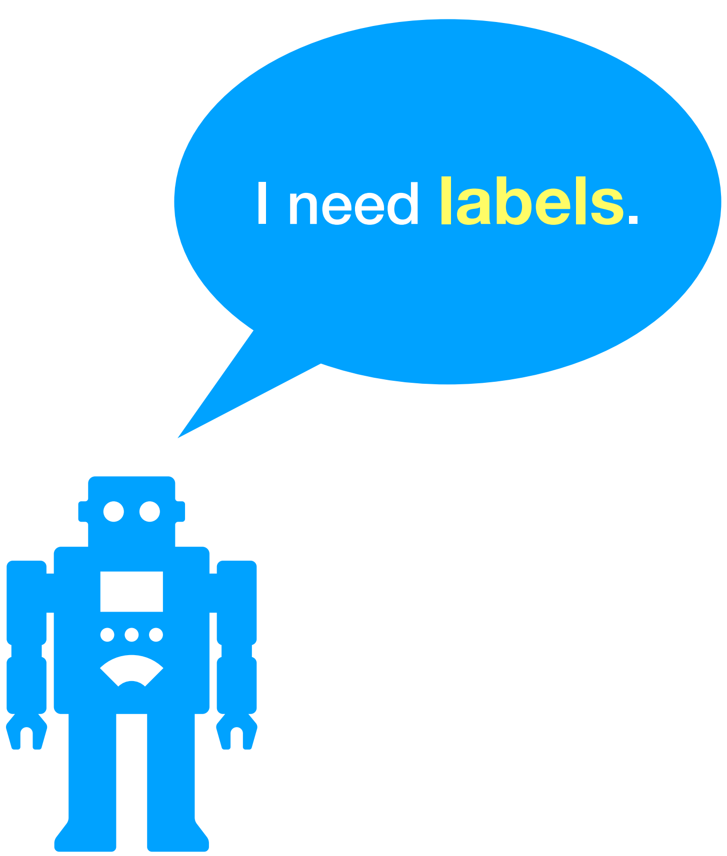
Contents

Introduction

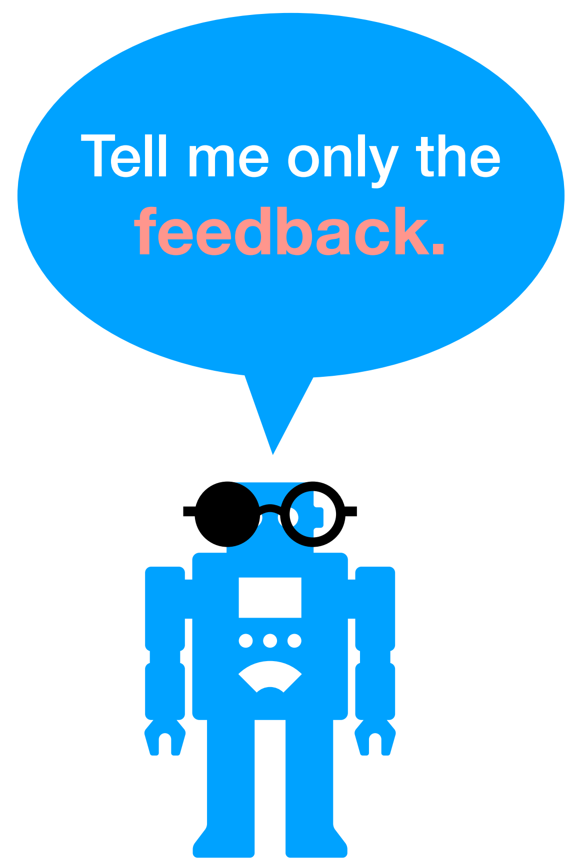
Machine Learning:

Learning from data and making predictions and/or decisions

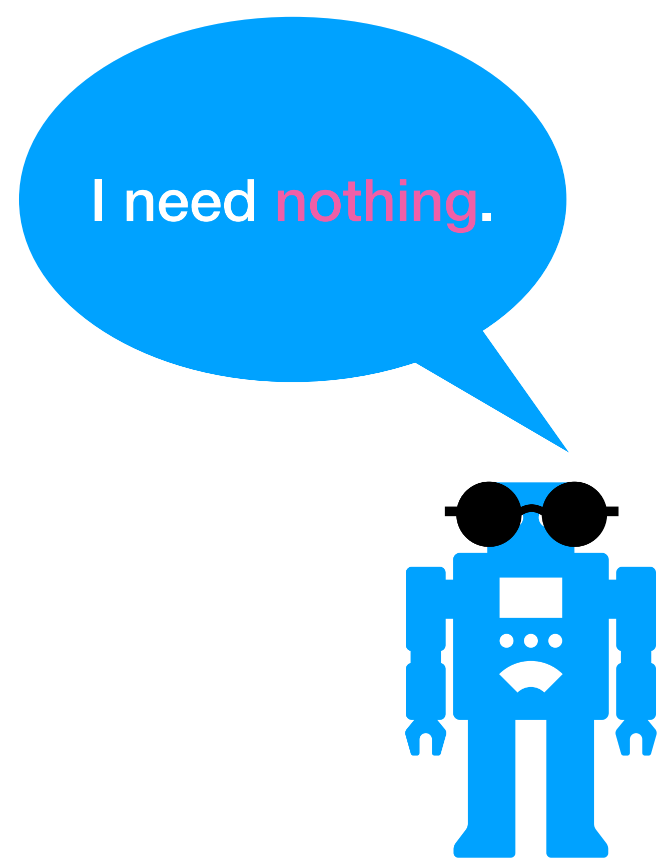
Supervised



Reinforcement

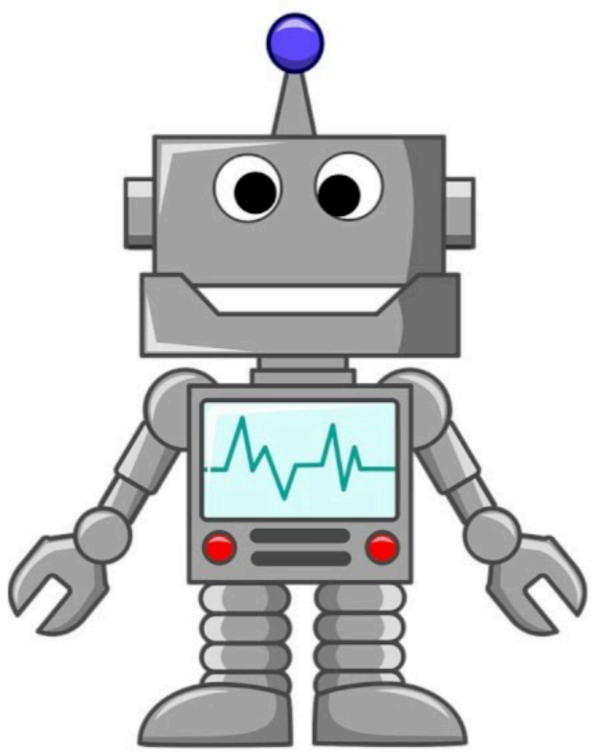


Unsupervised



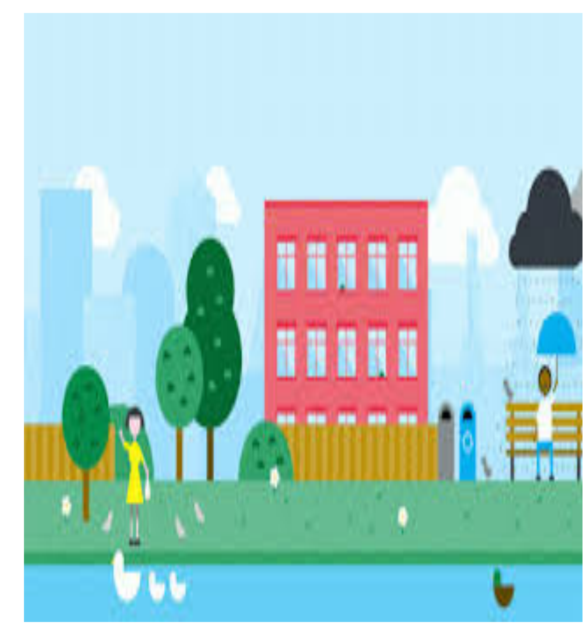
Introduction

What is RL?



Action

State Reward

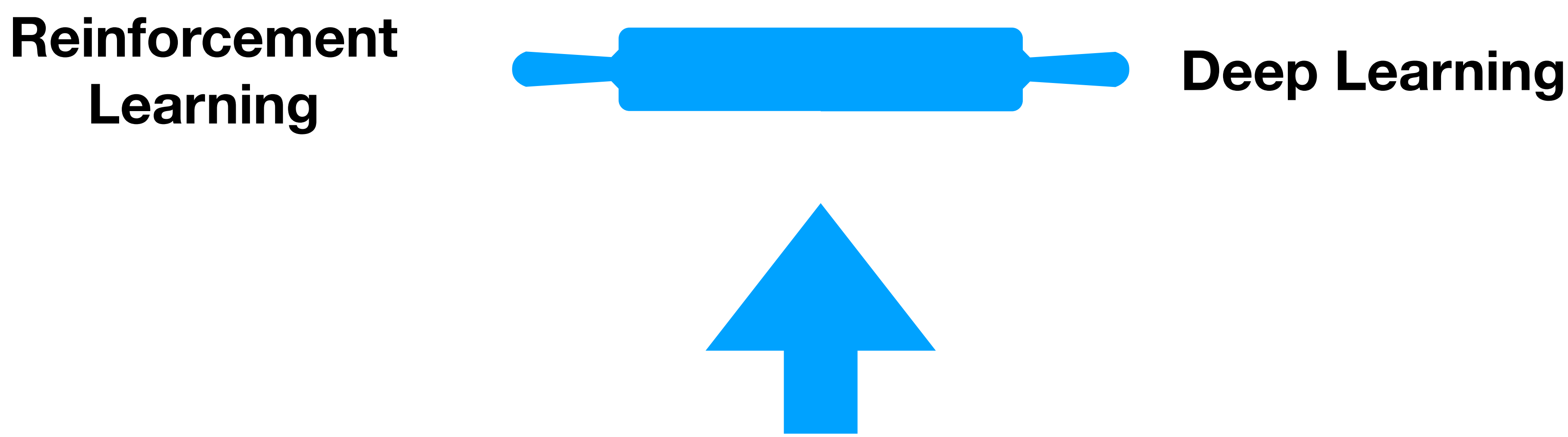


Agent

Environment

Figure Source:
https://www.youtube.com/watch?v=ub6mA2m3rq4
https://www.london.gov.uk/what-we-do/environment/london-environment-strategy

Introduction



- Big data
- Powerful computation
- New algorithmic techniques
- Mature software packages and architectures
- ……

Introduction

Why is deep learning so significant?

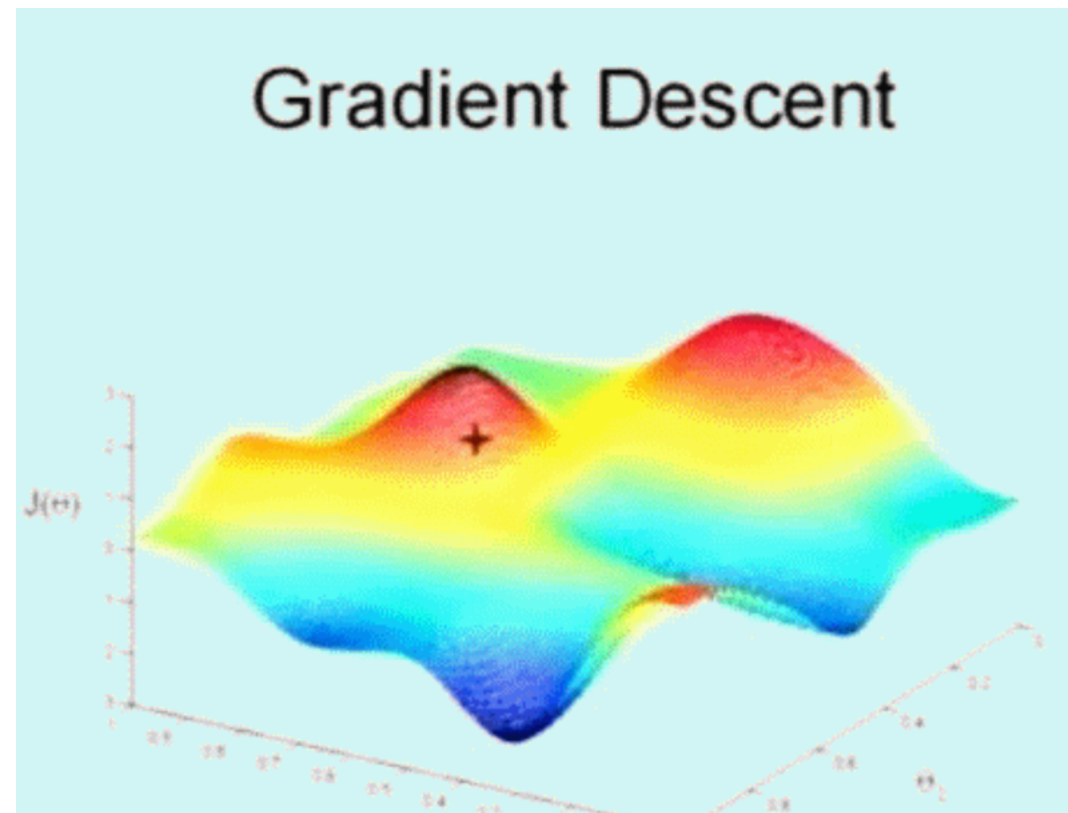
Deep learning

Features:
Radish/Red
Fruit
Shape
etc…

Features:
Sky Blue
Logo
Shape
etc…

Features:
Yellow
Fruit
Shape
etc…

Feature Engineering



End-to-end learning through gradient descent

Figure Source:
https://www.retentionscience.com/blog/feature-engineering-a-closer-look-part-1-rs-labs/

Introduction

Basic Reinforcement Algorithms

Value based

Policy based

Input: state S and action

Output: Value $V(S, a; \theta)$

Introduction

Basic Reinforcement Algorithms

Value based

Policy based

Temporal Difference(TD) learning

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$$

$$= (1 - \alpha)V(s) + \alpha(r + \gamma V(s'))$$

Introduction

Basic Reinforcement Algorithms

Value based

Policy based

SARSA

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$$

$$= (1-\alpha)Q(s,a) + \alpha(r + \gamma Q(s',a'))$$

Introduction

Basic Reinforcement Algorithms

Value based

Policy based

Q learning

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$
$$= (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a'))$$

Introduction

Basic Reinforcement Algorithms

Value based

Policy based

Q learning

Offline!

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

$$= (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a'))$$

Introduction

Basic Reinforcement Algorithms

Value based

Policy based

TD learning with Function approximation

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha[r + \gamma\hat{v}(s', \mathbf{w}) - \hat{v}(s, \mathbf{w})] \nabla \hat{v}(s, \mathbf{w})$$

Introduction

Basic Reinforcement Algorithms

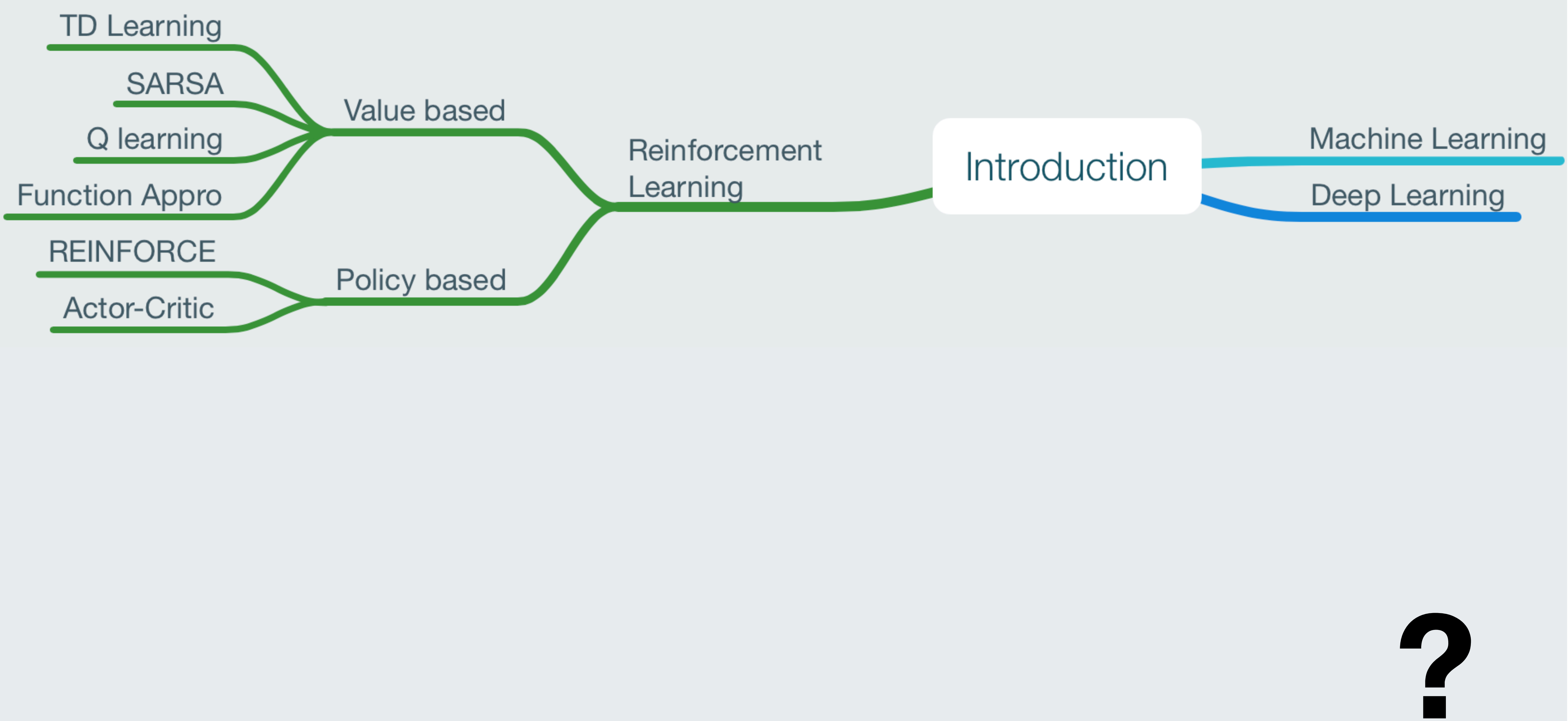
Value based

Policy based

Input: state S

Output: Policy $a(S)$

- REINFORCE
- Actor-Critic


TD Learning
SARSA
Q learning
Function Appro
REINFORCE
Actor-Critic
Value based
Policy based
Reinforcement Learning
Introduction
Machine Learning
Deep Learning
?

Contents

Core Elements

Deep Q-Learning(DQN)

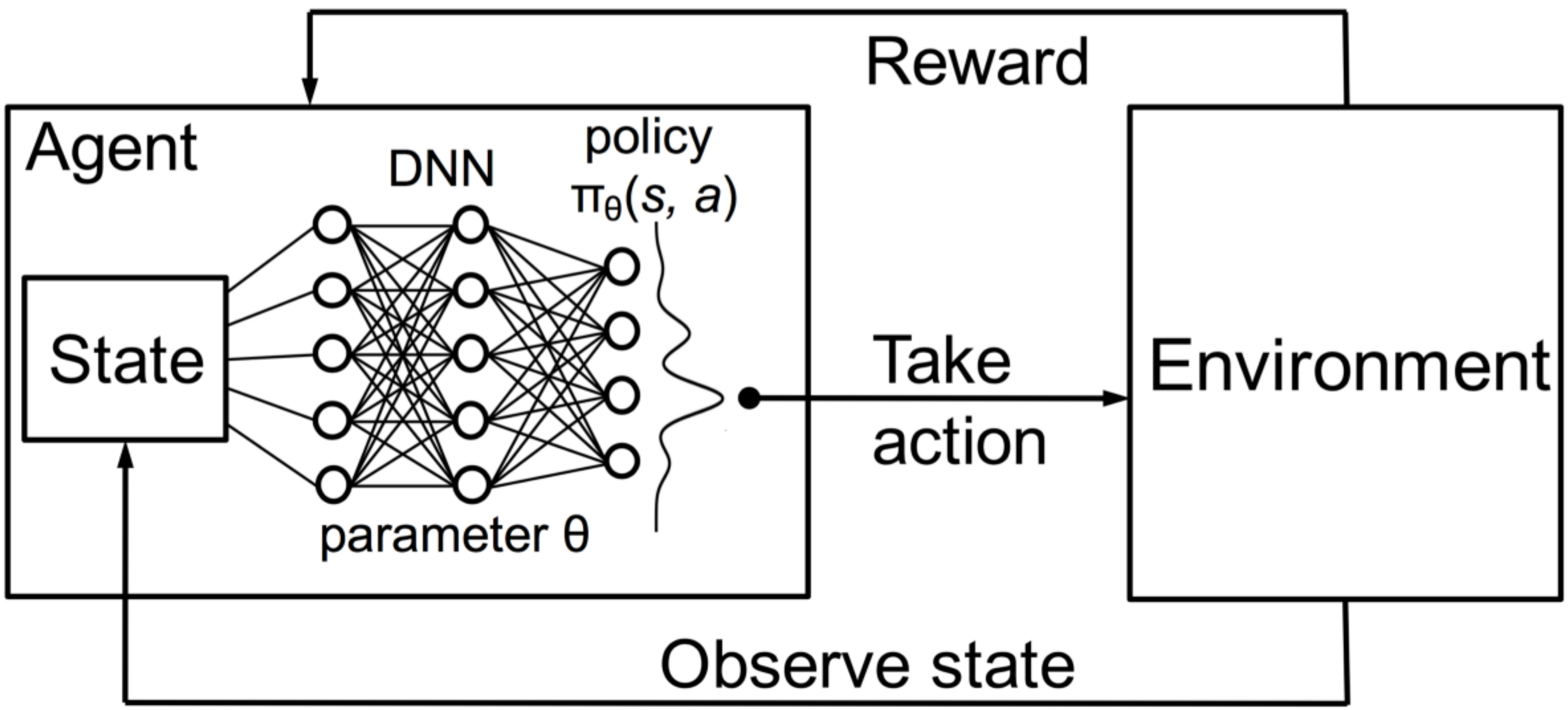


Figure Source:
http://people.csail.mit.edu/hongzi/content/publications/DeepRM-HotNets16.pdf

Core Elements

Deep Q-Learning(DQN)

Advantages:

- Experiment Replay
- End-to-end RL approach
- General network framework

Core Elements

Double DQN(D-DQN)

Original DQN:

$$y_t^Q = r_{t+1} + \gamma Q(s_{t+1}, arg \max_a Q(s_{t+1}, a_t; \theta_t); \theta_t)$$

Over Estimation!

Core Elements

Double DQN(D-DQN)

Solution:

Use one network to select,

use another network to evaluate.

$$y_t^{D-DQN} = r_{t+1} + \gamma Q(s_{t+1}, arg \max_a Q(s_{t+1}, a_t; \theta_t); \theta_t^-)$$

Core Elements

Double DQN(D-DQN)

Two methods:

Four networks: Double the original architecture.

Two networks: Just use the target network to evaluate.

Core Elements

Prioritized Experience Replay

$$TDerror = \Delta Q$$

prioritization p

Choose samples according to p by SumTree

Core Elements

Dueling Architecture

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left(A(s, a; \theta, \alpha) - \frac{a}{|\mathscr{A}|} A(s, a'; \theta, \alpha)\right)$$

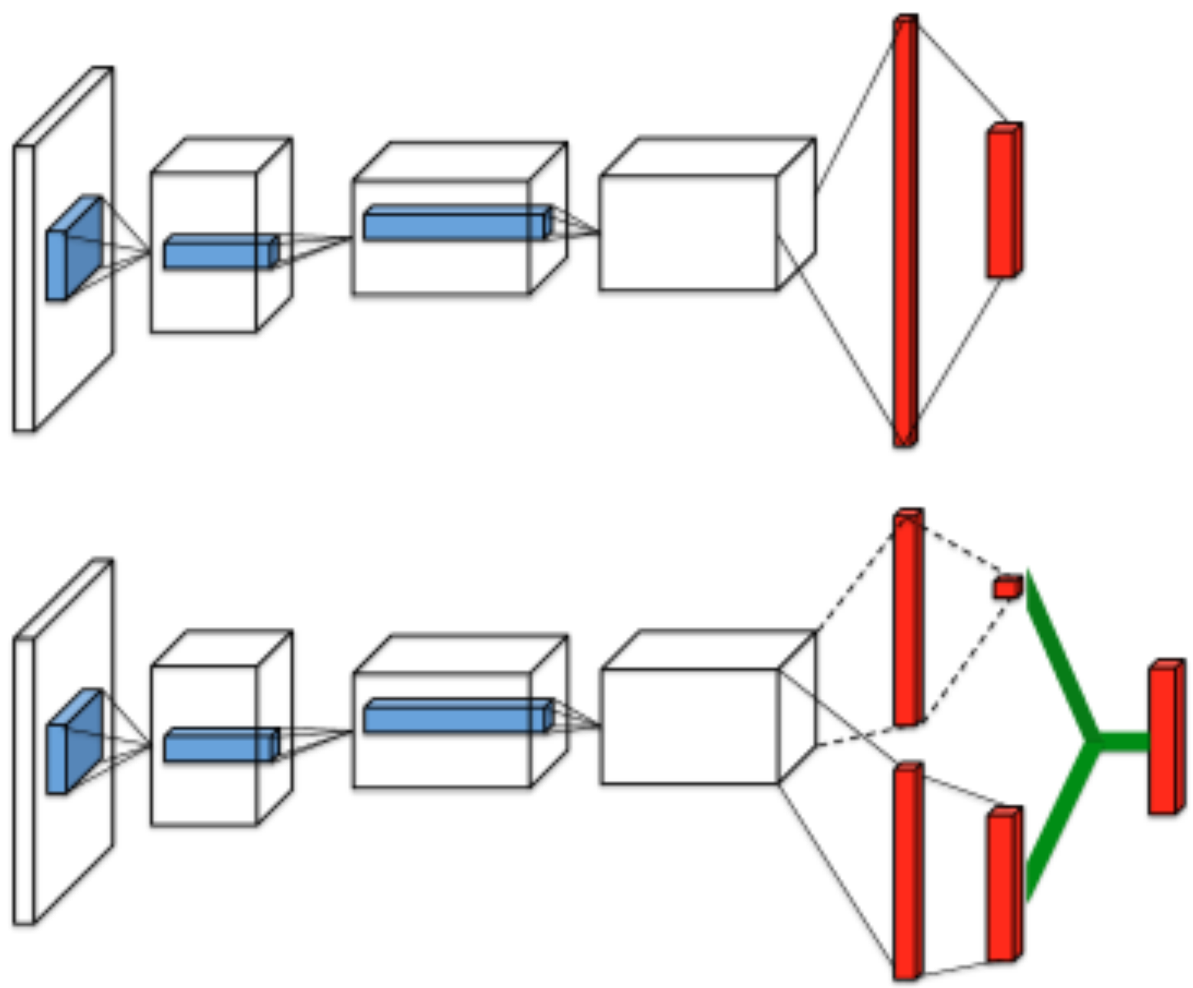


Figure Source:

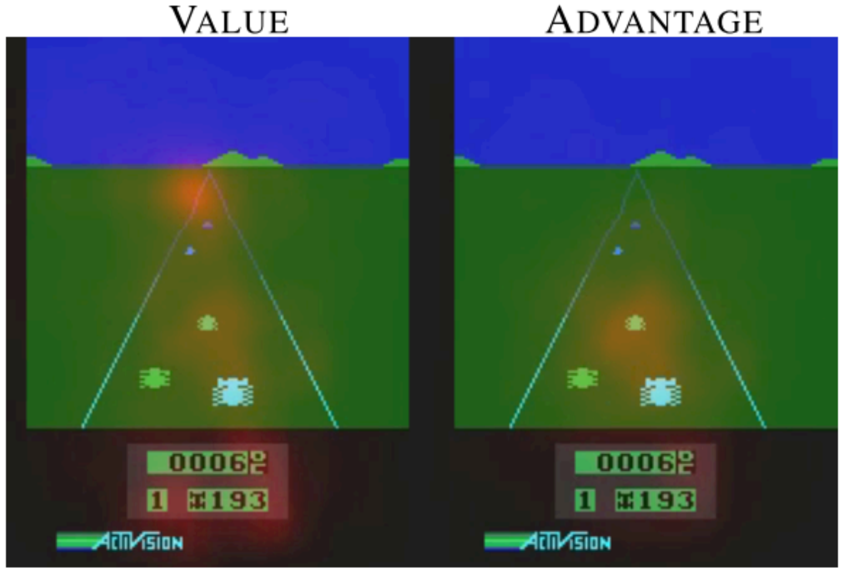
Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., & de Freitas, N. (2015). Dueling Network Architectures for Deep Reinforcement Learning, (9). https://doi.org/10.1109/MCOM.2016.7378425

Core Elements

Dueling Architecture

Value: Only consider state

Advantage: Consider both state and action



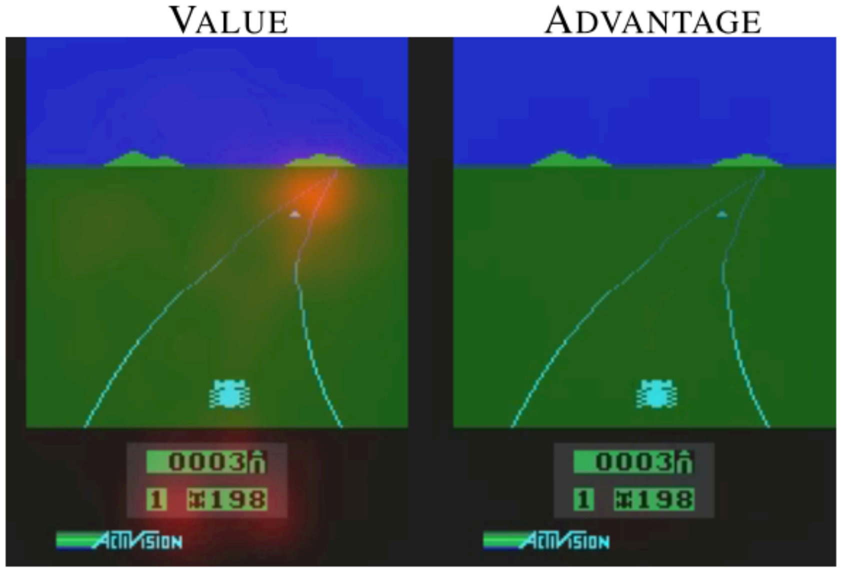


Figure Source:
Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., & de Freitas, N. (2015). Dueling Network Architectures for Deep Reinforcement Learning, (9). https://doi.org/10.1109/MCOM.2016.7378425

Core Elements

Policy based Deep algorithms

- Policy Gradient
- Asynchronous Advantage Actor-cirtic(A3C)

Core Elements

Reward

- Learning from demonstration:

  Deep Q-learning from Demonstrations(DQfD)

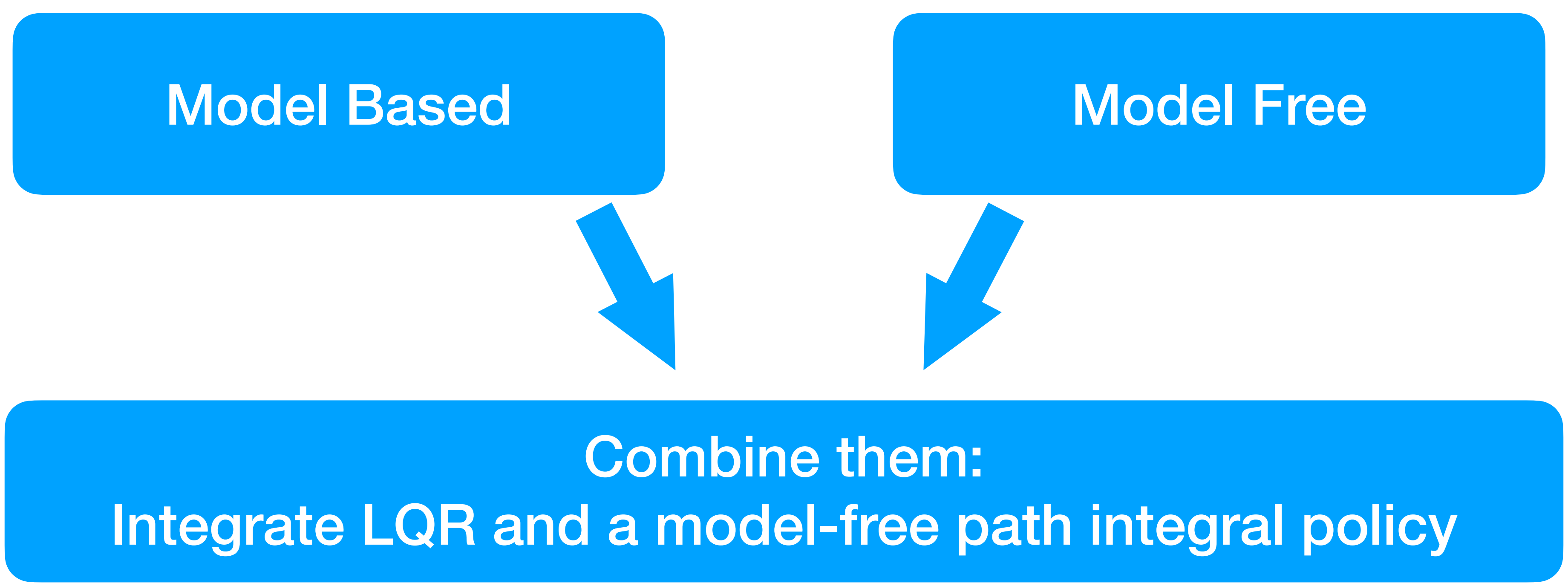
- Generative Adversarial Imitation Learning

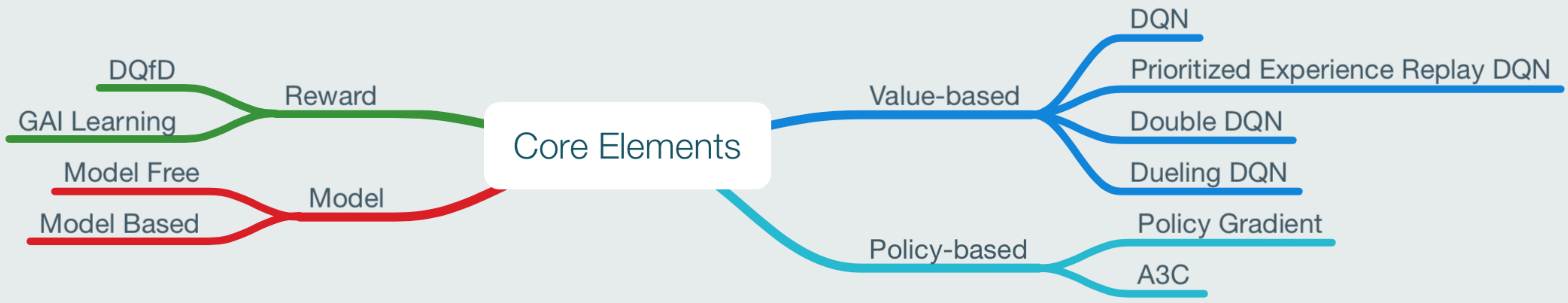
Core Elements

Model

Model: An agent’s representation of the environment.

e.g. transition model and the reward model.



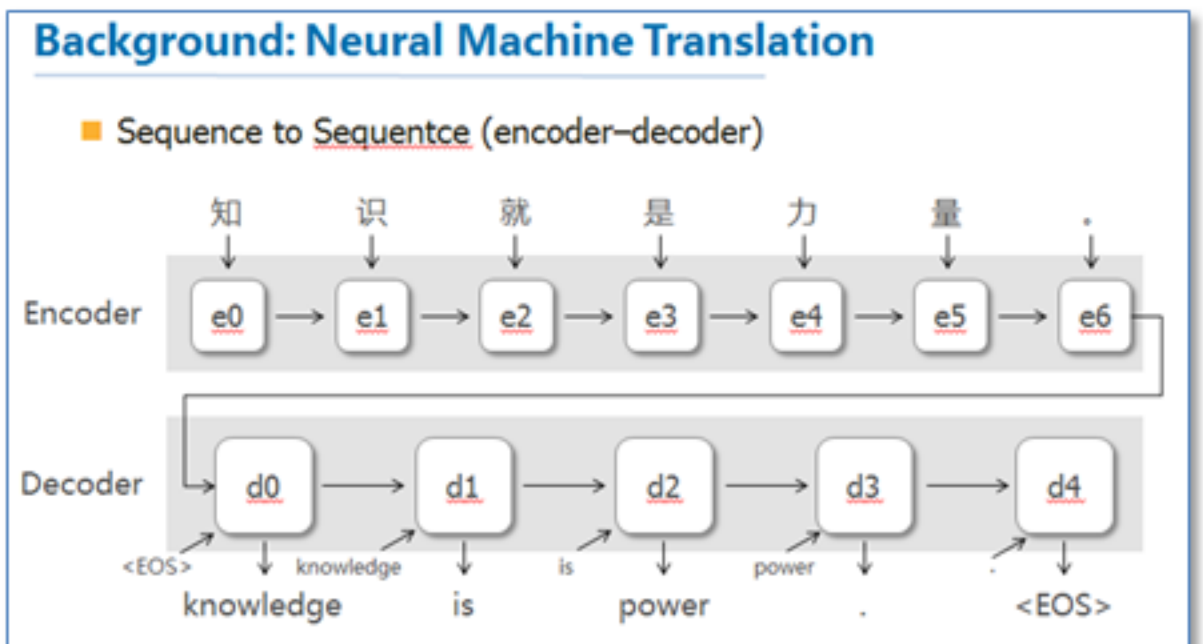

Core Elements
Reward
DQfD
GAI Learning
Model
Model Free
Model Based
Value-based
DQN
Prioritized Experience Replay DQN
Double DQN
Dueling DQN
Policy-based
Policy Gradient
A3C

?

Contents

Important Mechanisms

Attention



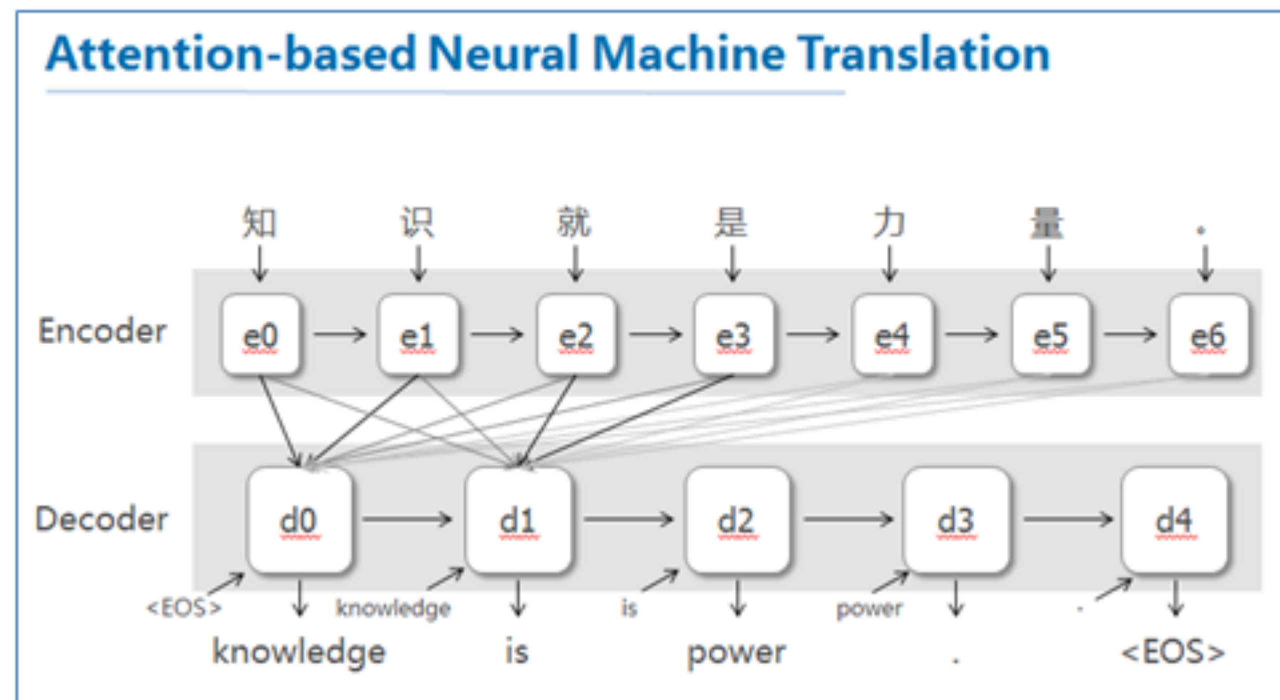


Figure Source:
https://www.cnblogs.com/robert-dlut/p/5952032.html

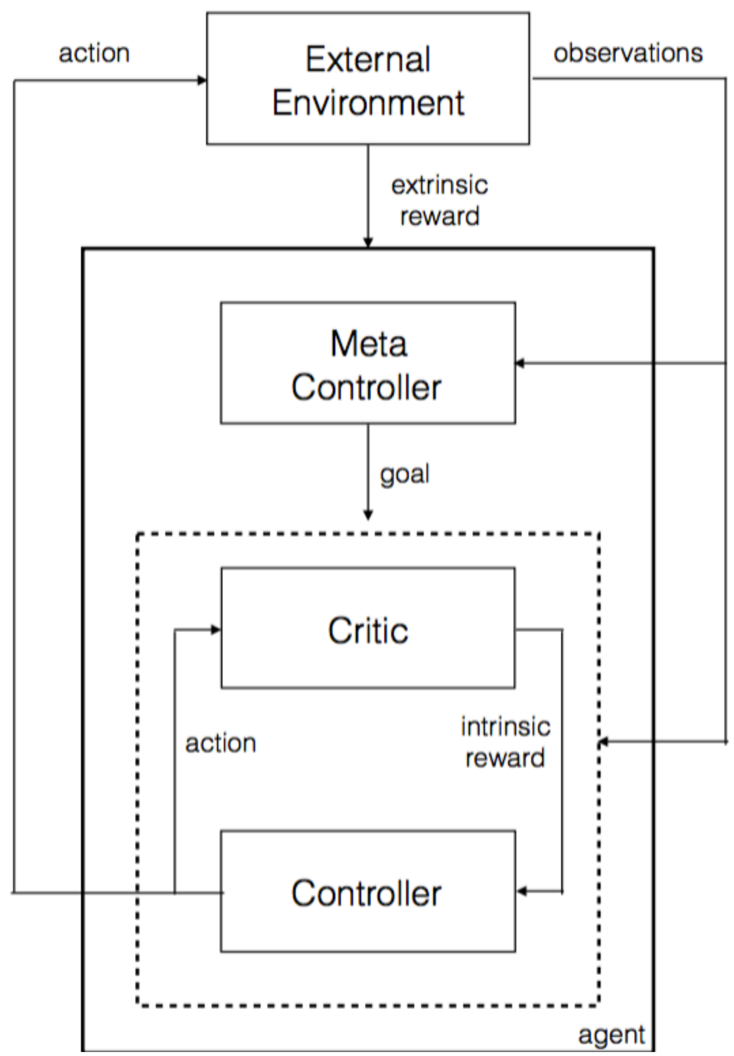
Important Mechanisms

Unsupervised Learning

- UNsupervised REinforcement and Auxiliary Learning(UNREAL)
- Generative Adversarial Networks(GAN)

Important Mechanisms

Hierarchical Reinforcement Learning



Use two networks:

One to determine destination,

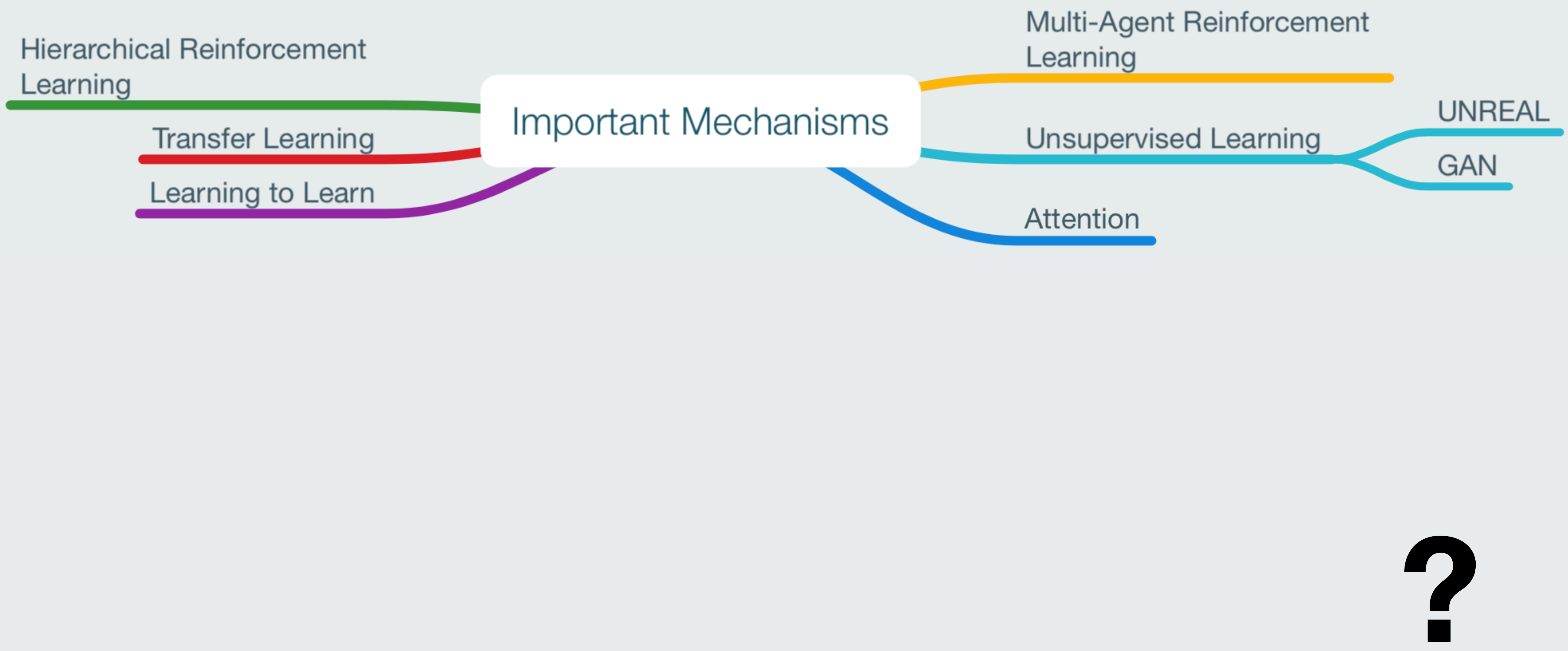
another to determine concrete action.

Figure Source:
Kulkarni, T. D., Narasimhan, K. R., Saeedi, A., & Tenenbaum, J. B. (2016). Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation, 1–14. https://doi.org/10.1023/A:1025696116075

Important Mechanisms

Other Topics

- Transfer Learning
- Multi-Agent Reinforcement Learning
- Learning to Learn


Important Mechanisms
Hierarchical Reinforcement Learning
Transfer Learning
Learning to Learn
Multi-Agent Reinforcement Learning
Unsupervised Learning
UNREAL
GAN
Attention
?

Contents

Applications

Games

Perfect Information Broad Games

Computer Go: AlphaGo

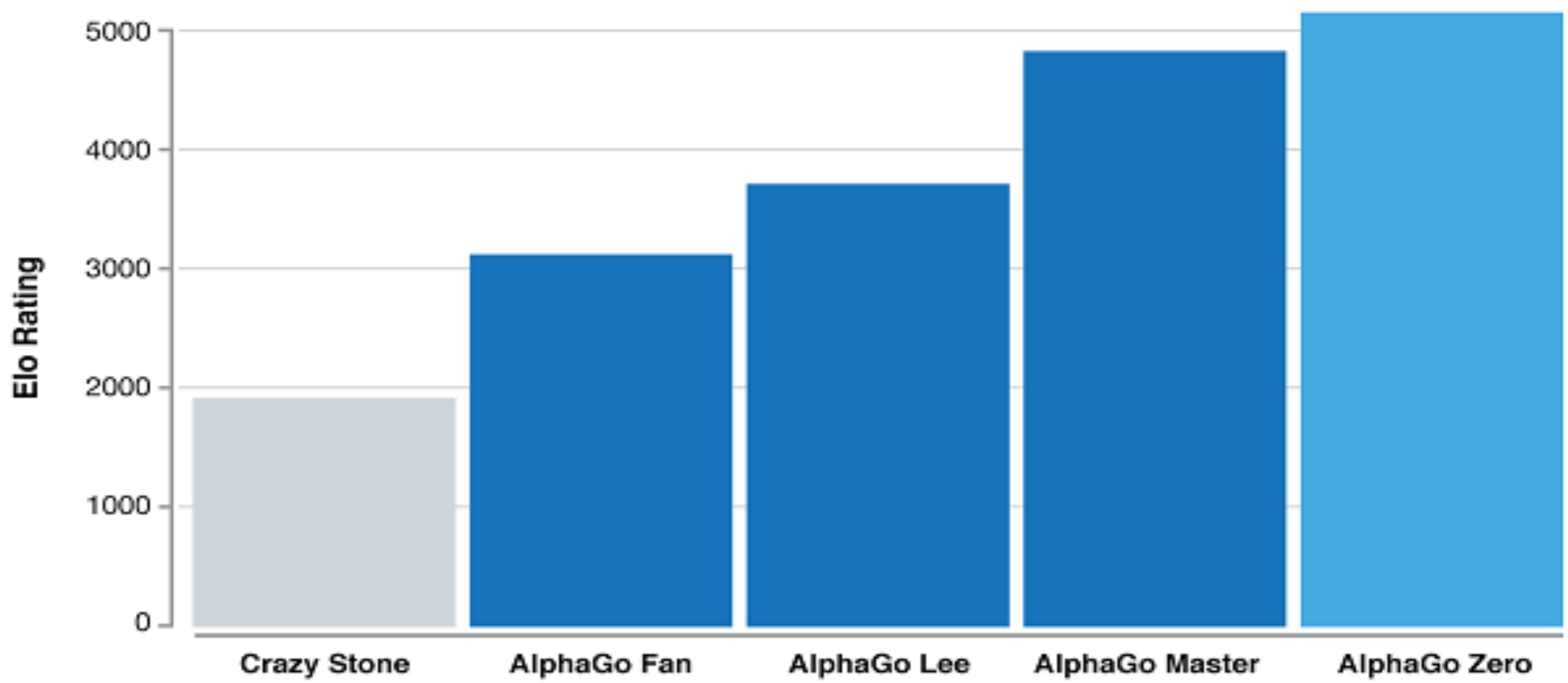


Figure Source:
https://deepmind.com/blog/alphago-zero-learning-scratch/

Applications

Games

Computer Go: AlphaGo

- Supervised Learning(SL) policy network

- Reinforcement Learning(RL) policy network

- Reinforcement Learning(RL) value network

- Monte Carlo tree search(MCTS)

Applications

Games

Computer Go: AlphaGo Zero Starting from scratch

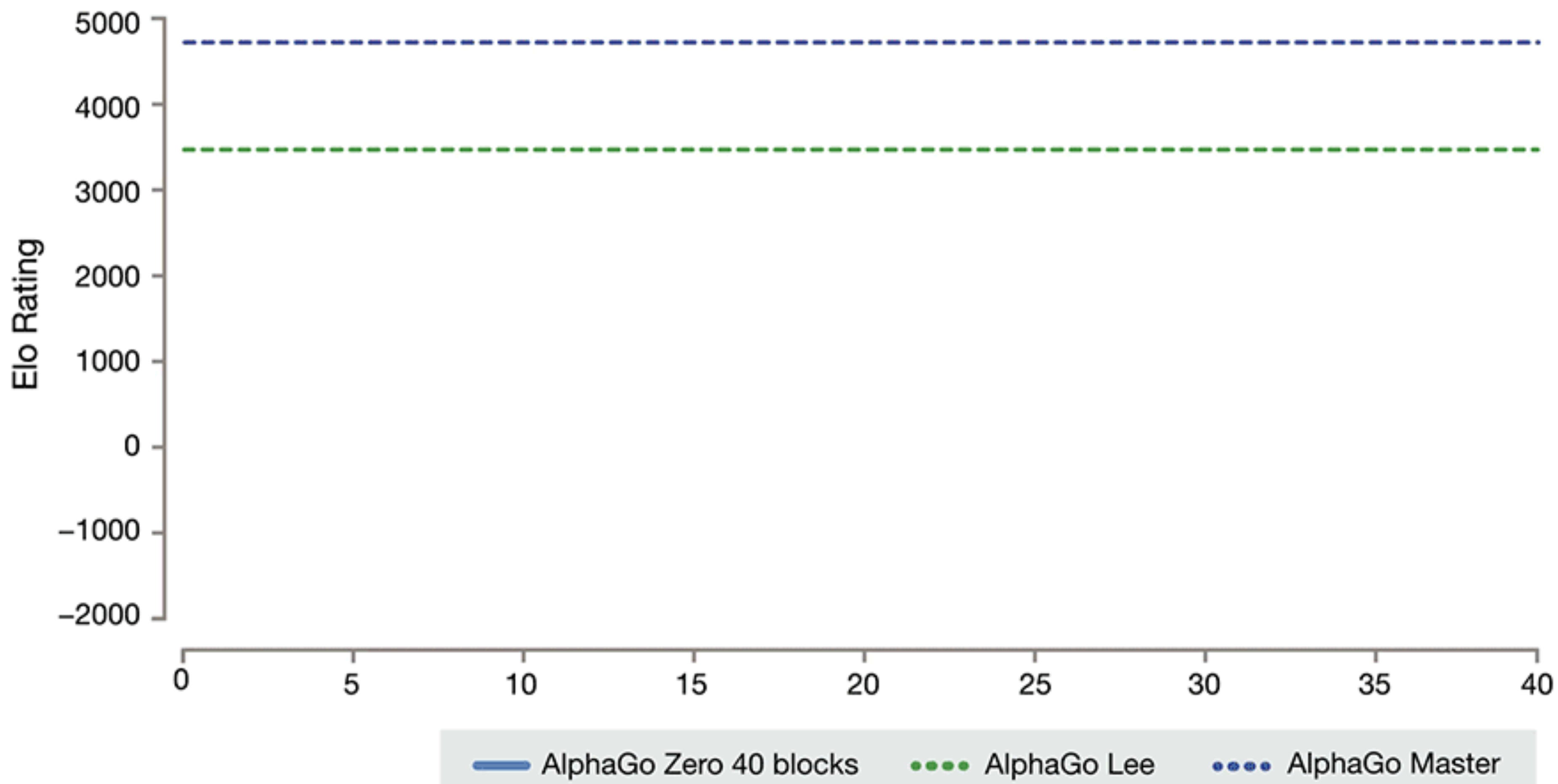


Figure Source:
https://deepmind.com/blog/alphago-zero-learning-scratch/

Applications

Games

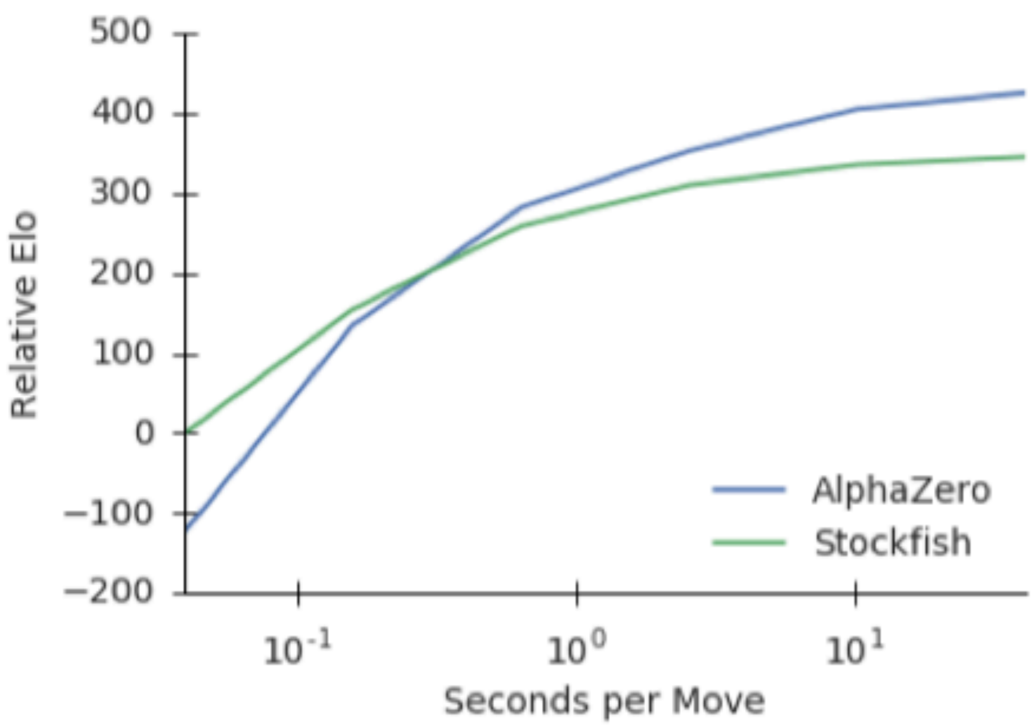
Computer Go: AlphaGo Zero
Starting from scratch

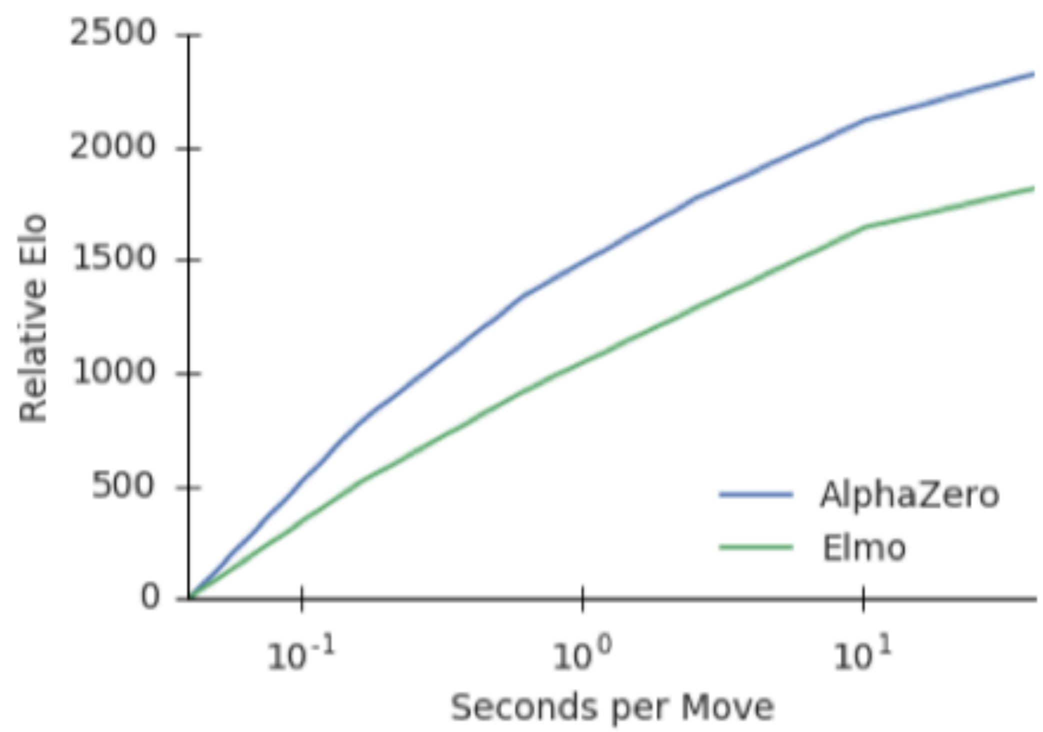
- No human knowledge
- Only one Reinforcement Learning(RL) network
- Monte Carlo tree search(MCTS)

Applications

Games

More Generally: Alpha Zero

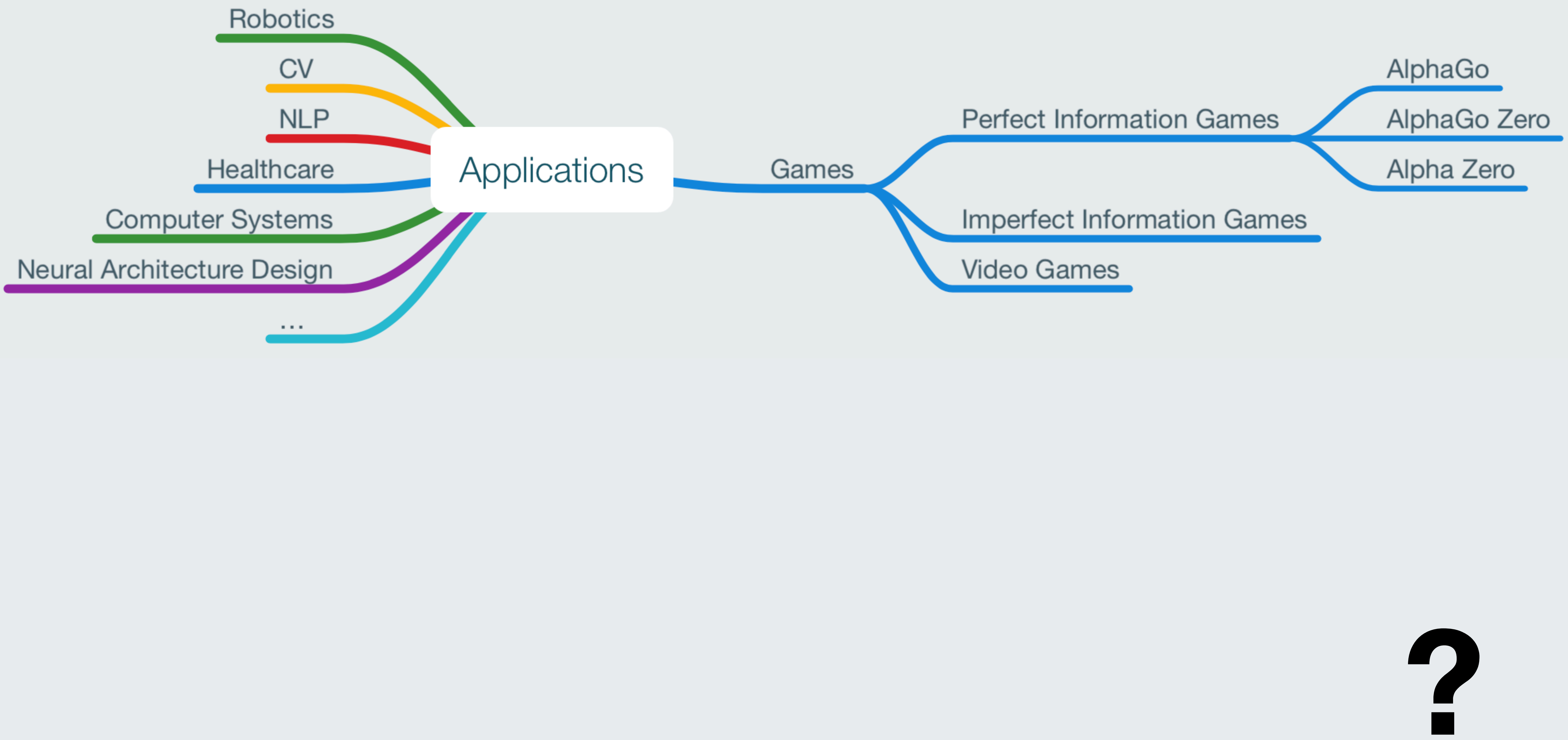


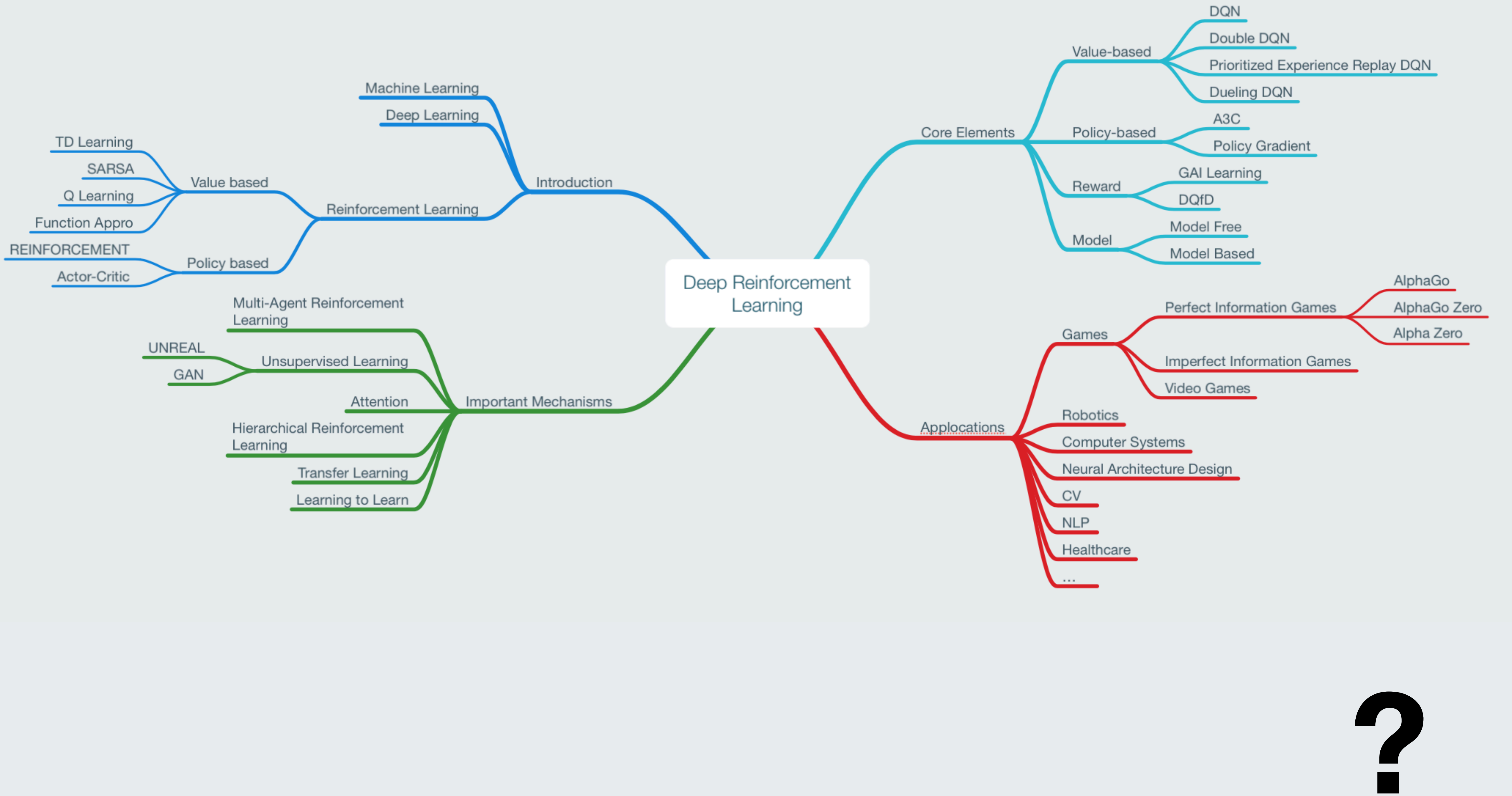


Applications

Other applications

- Imperfect Information Broad Games
- Video Games
- Robotics
- Natural Language Processing(NLP)
- Computer Vision(CV)
- Neural Architecture Design
- Healthcare
- Computer Systems
- …


Robotics
CV
NLP
Healthcare
Computer Systems
Neural Architecture Design
...
Applications
Games
Perfect Information Games
AlphaGo
AlphaGo Zero
Alpha Zero
Imperfect Information Games
Video Games
?


Deep Reinforcement Learning
Introduction
Machine Learning
Deep Learning
Reinforcement Learning
Value based
TD Learning
SARSA
Q Learning
Function Appro
Policy based
REINFORCEMENT
Actor-Critic
Important Mechanisms
Multi-Agent Reinforcement Learning
Unsupervised Learning
UNREAL
GAN
Attention
Hierarchical Reinforcement Learning
Transfer Learning
Learning to Learn
Core Elements
Value-based
DQN
Double DQN
Prioritized Experience Replay DQN
Dueling DQN
Policy-based
A3C
Policy Gradient
Reward
GAI Learning
DQfD
Model
Model Free
Model Based
Applocations
Games
Perfect Information Games
AlphaGo
AlphaGo Zero
Alpha Zero
Imperfect Information Games
Video Games
Robotics
Computer Systems
Neural Architecture Design
CV
NLP
Healthcare
...
?