KAIST Summer Session 2018

Module 3. Deep Learning with PyTorch

Deep Reinforcement Learning

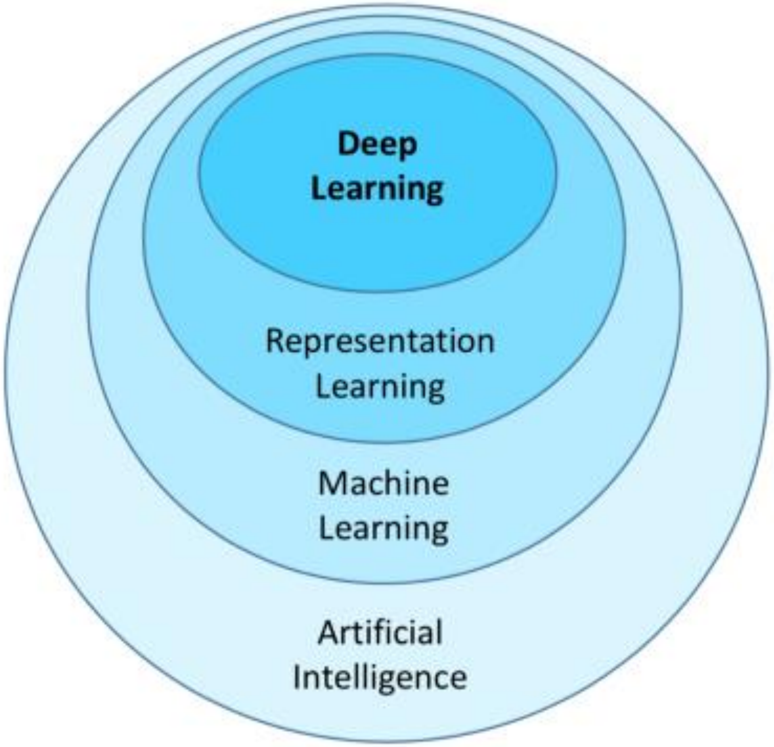
KAIST College of Business

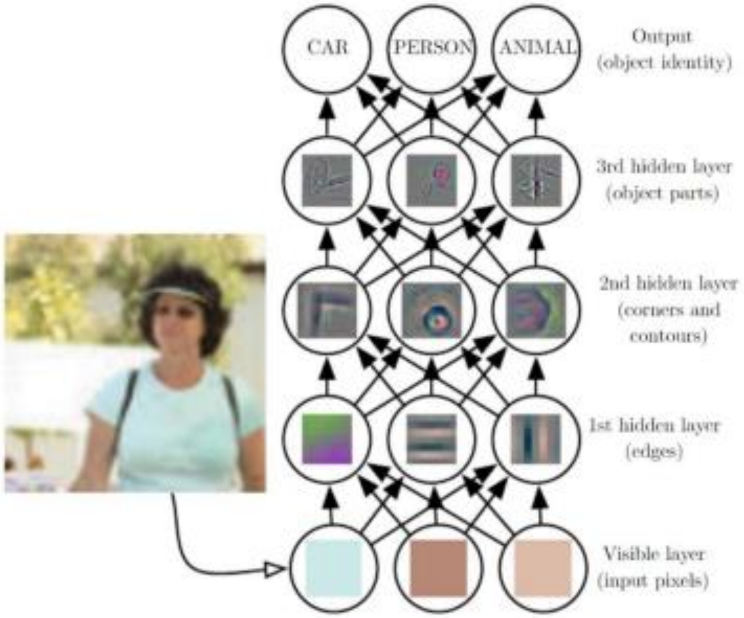
Jiyong Park

27 August, 2018

Reinforcement Learning

Deep Learning is Representation Learning

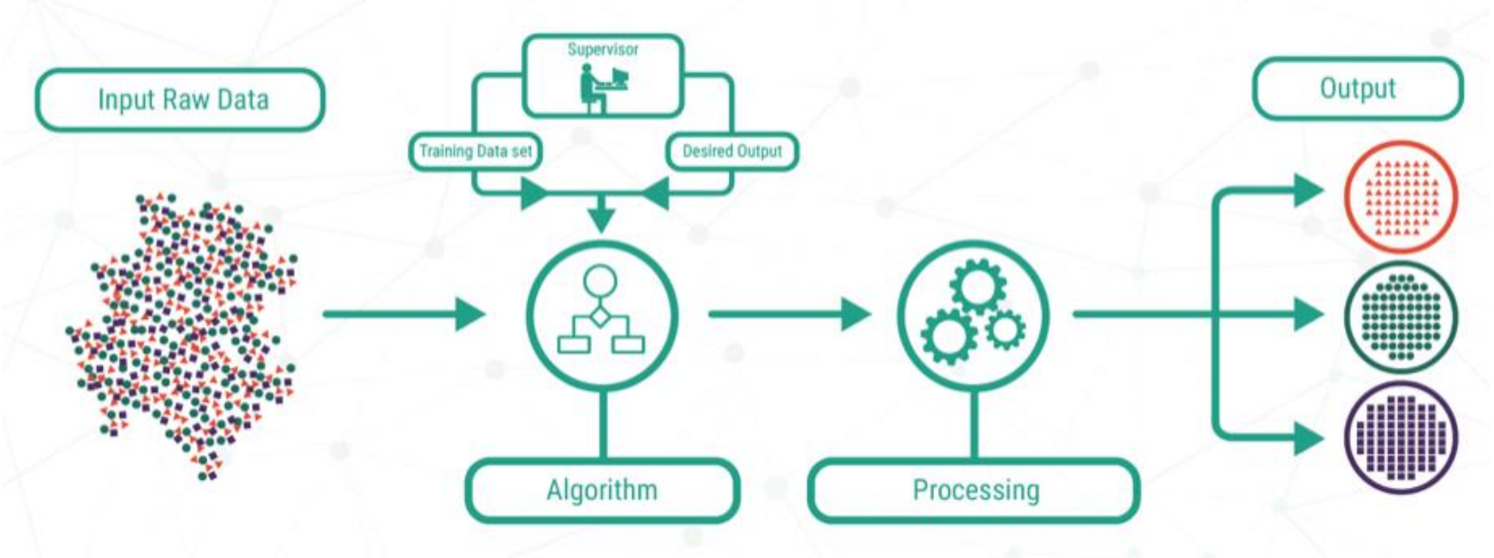




https://hackernoon.com/mit-6-s094-deep-learning-for-self-driving-cars-2018-lecture-3-notes-deep-reinforcement-learning-fe9a8592e14a

Supervised and Unsupervised Learning

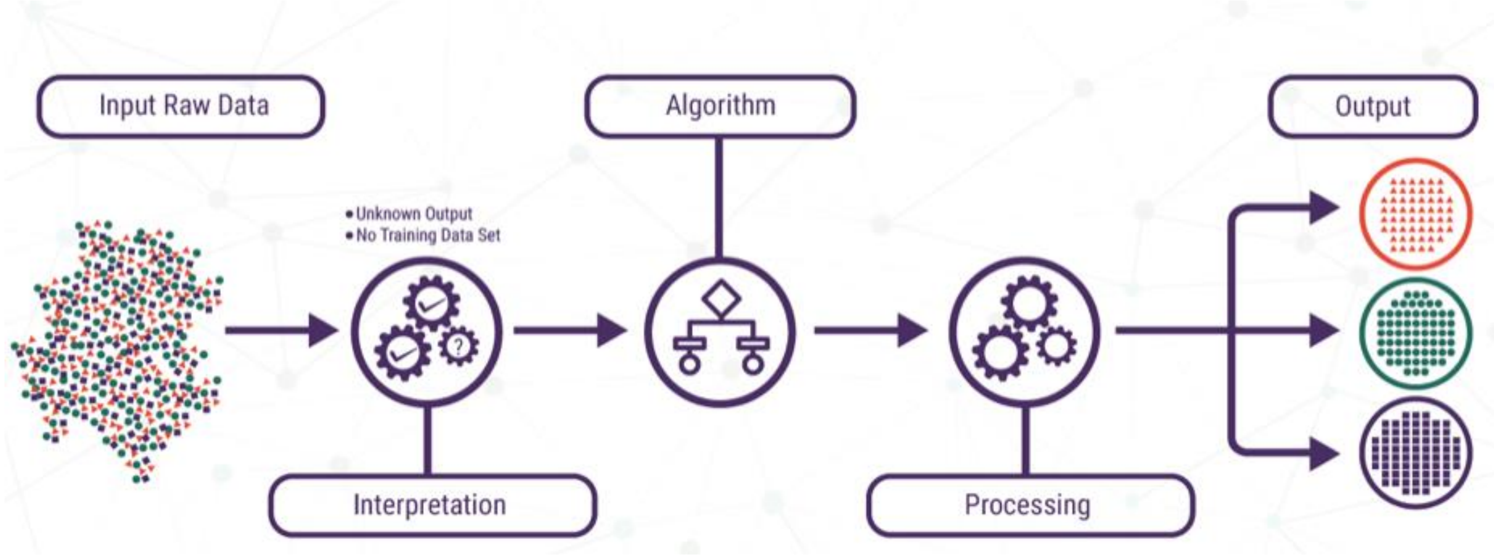
- Supervised learning is to imitate the desired outputs given the inputs.
  - ➢ We should know what we want to train the model concretely (e.g., classifying dog and cat)



http://bigdata-madesimple.com/machine-learning-explained-understanding-supervised-unsupervised-and-reinforcement-learning/

Supervised and Unsupervised Learning

- Unsupervised learning is to describe the patterns behind the data.
  - ➢ We can explore how the data look like (e.g., feature representation using auto-encoder)



http://bigdata-madesimple.com/machine-learning-explained-understanding-supervised-unsupervised-and-reinforcement-learning/

Reinforcement Learning

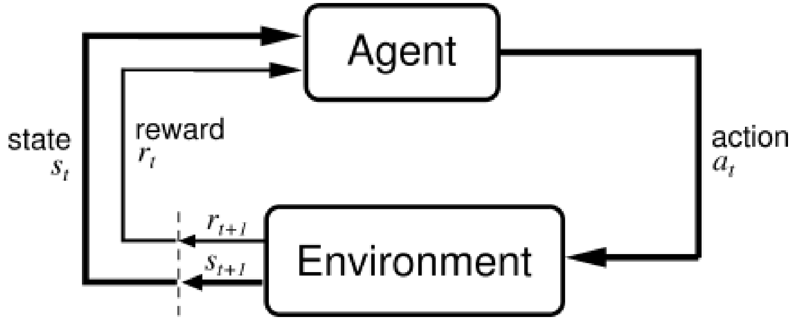
- Reinforcement learning is to reinforce some behaviors for specific purposes.
  - We do not know exactly how to do it, but we should know what we want to achieve through the model (e.g., toilet training for dogs)

http://bigdata-madesimple.com/machine-learning-explained-understanding-supervised-unsupervised-and-reinforcement-learning/

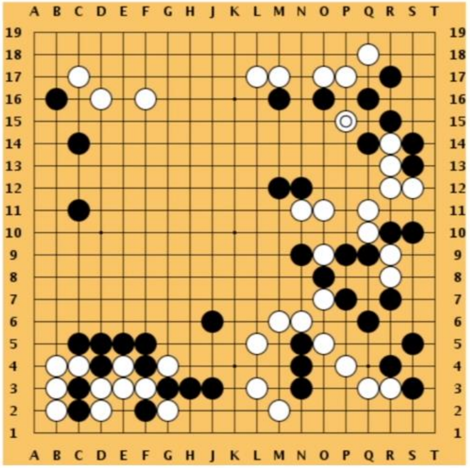
Deep Q-Network

State, Action, and Reward

- Reinforcement setting



  - (Example) Go



Objective: Win the game!

State: Position of all pieces
Action: Where to put the next piece down
Reward: 1 if win at the end of the game, 0 otherwise

Q-Learning

- How to determine what action should be taken?
  - Let's ask the Q!

The Q knows everything. Just follow.

  - Anyway, what is the Q? *(Expected reward for that state given that action)*

$$\underbrace{Q^{\pi}(s_t, a_t)}_{\text{Q value for that state given that action}} = \underline{E}[\underbrace{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots}_{\text{Expected discounted cumulative reward ...}} | \underbrace{s_t, a_t}_{\text{given that state and that action}}]$$

Q-Learning

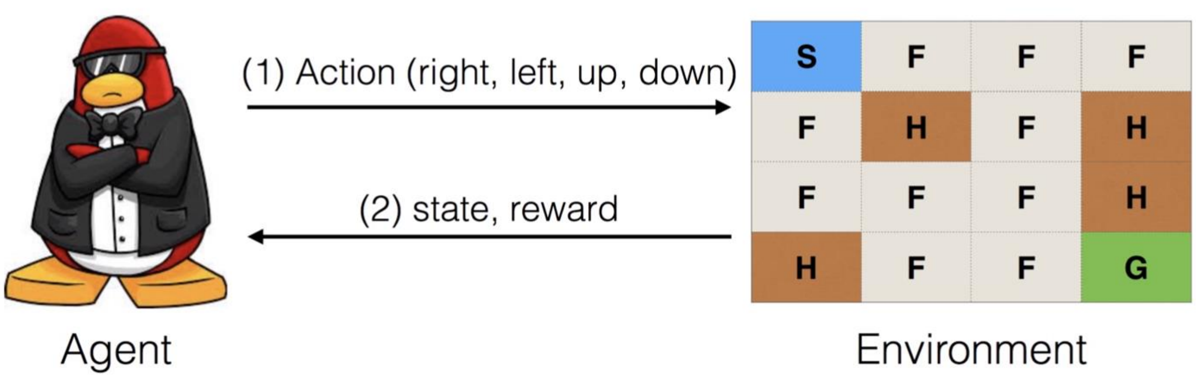
PYTORCH

- Bellman equation

$$NewQ(s,a) = R(s,a) + \gamma \, max \, Q'(s',a')$$

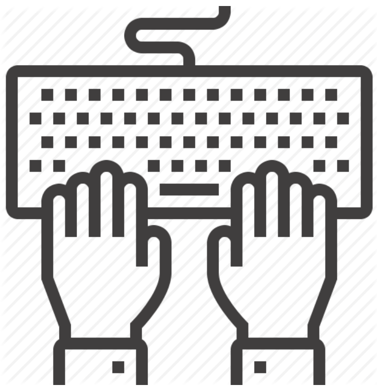
Discount rate

- Let's practice in the Frozen Lake World (see OpenAI [Gym])
  - ➢ An agent should reach the goal while avoding holes.



https://github.com/hunkim/ReinforcementZeroToAll

KAIST COLLEGE OF BUSINESS

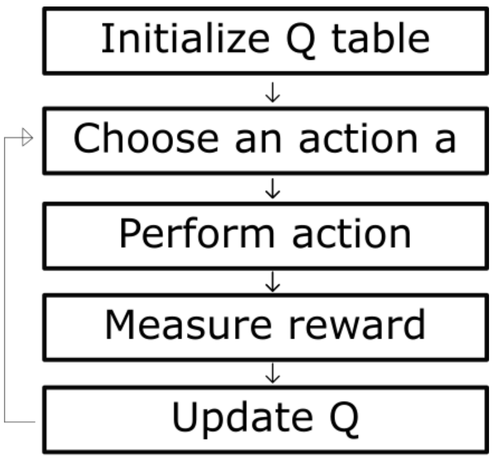


Frozen Lake World

M3.7 Q-Learning_FrozenLake.ipynb

Q-Learning in a Deterministic Environment

- Q-Table (State × Action Table)



Q-table initialised at zero

	UP	DOWN	LEFT	RIGHT
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0

After few episodes

	UP	DOWN	LEFT	RIGHT
0	0	0	0	0
1	0	0	0	0
2	0	2.25	2.25	0
3	0	0	5	0
4	0	0	0	0
5	0	0	0	0
6	0	5	0	0
7	0	0	2.25	0
8	0	0	0	0

Eventually

	UP	DOWN	LEFT	RIGHT
0	0	0	0.45	0
1	0	1.01	0	0
2	0	2.25	2.25	0
3	0	0	5	0
4	0	0	0	0
5	0	0	0	0
6	0	5	0	0
7	0	0	2.25	0
8	0	0	0	0

Q-Learning in a Stochastic Environment

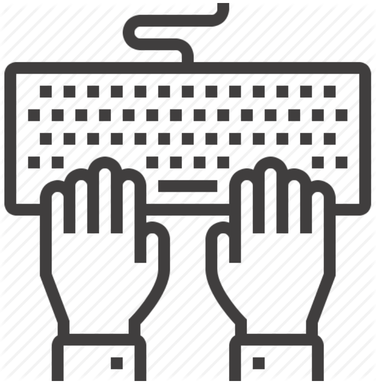
- Adding a stochastic component to the Q function is helpful in a stochastic environment
  - ➢ (1) Learning rate (= let's learn part of the Q)

$$\underbrace{NewQ(s,a)}_{\text{New Q value for that state and that action}} = \underbrace{Q(s,a)}_{\text{Current Q value}} + \underset{\text{Learning Rate}}{\alpha}[\underbrace{R(s,a)}_{\text{Reward for taking that action at that state}} + \underset{\text{Discount rate}}{\gamma} \underbrace{max\, Q'(s',a')}_{\text{Maximum expected future reward given the new s' and all possible actions at that new state}} - Q(s,a)]$$

  - ➢ Exploration and exploitation (ϵ-Greedy Approach)

$$If\ random.value\ < epsilon,\ then\ choose\ a\ random\ action$$

$$If\ random.value\ \geq epsilon,\ then\ choose\ the\ action\ guided\ by\ the\ Q - value$$



Frozen Lake World

M3.7 Q-Learning_FrozenLake.ipynb

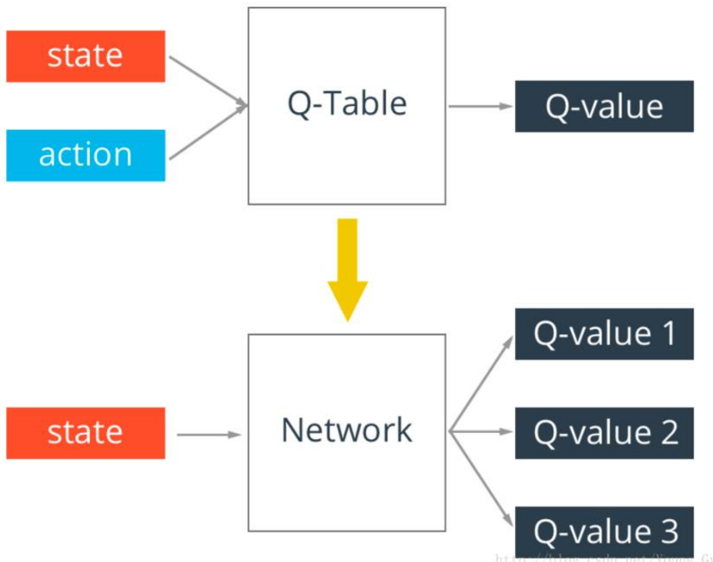
Limitation of Q-Table

- What if complex states? Too large table?

https://blog.openai.com/dota-2/

Q-Network

- Using neural networks instead of Q-table
  - ➢ Input: states / Output: Q-value for each action

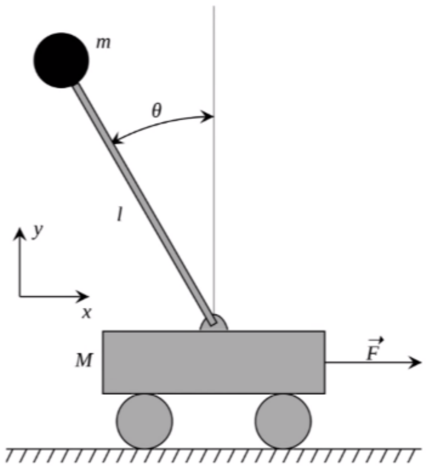


  - ➢ Loss function

$$loss = \left(\underbrace{\overset{\text{Reward}}{r} + \overset{\text{Decay Rate}}{\gamma} \max_{a`} \hat{Q}(s,a`)}_{\text{Target}} - \underbrace{Q(s,a)}_{\text{Prediction}}\right)^2$$

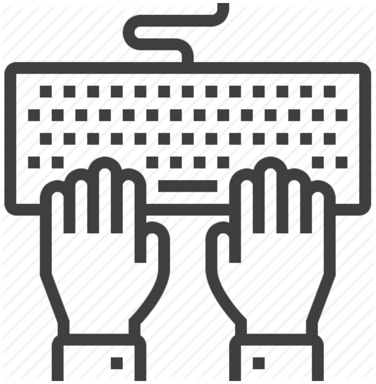
New Task for Q-Network

Cart-Pole Problem



Objective: Balance a pole on top of a movable cart

State: angle, angular speed, position, horizontal velocity
Action: horizontal force applied on the cart
Reward: 1 at each time step if the pole is upright



Cart-Pole Problem

M3.7 Deep Q-Network_CartPole.ipynb

Deep Q-Network (DQN)

- Reinforcement learning using neural networks is not new idea.
  - ➢ But, why does DQN become popular recently?

Reinforcement Learning for Robots Using Neural Networks

Long-Ji Lin

January 6, 1993
CMU-CS-93-103

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

A Dissertation
Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

This research was supported in part by Fujitsu Laboratories Ltd. and in part by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597.

Deep Q-Network (DQN)

- Reinforcement learning using neural networks is not new idea
  - ➢ But, why does DQN become popular recently? (*Thanks to DeepMind*)

LETTER

doi:10.1038/nature14236

Human-level control through deep reinforcement learning

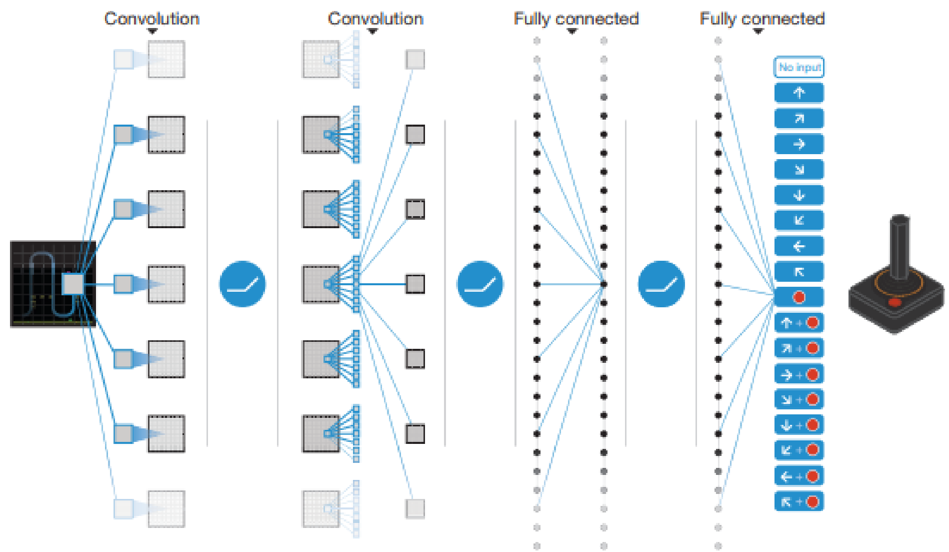
Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Be
Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir S
Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

The theory of reinforcement learning provides a normative account[1], deeply rooted in psychological[2] and neuroscientific[3] perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted with a difficult task: they must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalize past experience to new situations. Remarkably, humans and other animals seem to solve this problem through a harmonious combination of reinforcement learning and hierarchical sensory processing systems[4,5], the former evidenced by a wealth of neural data revealing notable parallels between the phasic signals emitted by dopaminergic neurons and temporal difference reinforcement learning algorithms[3]. While reinforcement learning agents have achieved some successes in a variety of domains[6–8], their applicability has previously

agent is to select actions in a fashion th
reward. More formally, we use a deep c
approximate the optimal action-value

$Q^*(s,a) = \max_\pi \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_t$

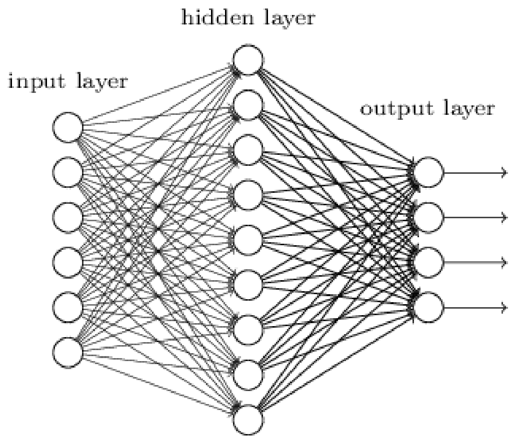
which is the maximum sum of rewards
step t, achievable by a behaviour poli
observation (s) and taking an action (
Reinforcement learning is known t
when a nonlinear function approxima
used to represent the action-value (als
instability has several causes: the corre
of observations, the fact that small upda
the policy and therefore change the data
between the action-values (Q) and the t



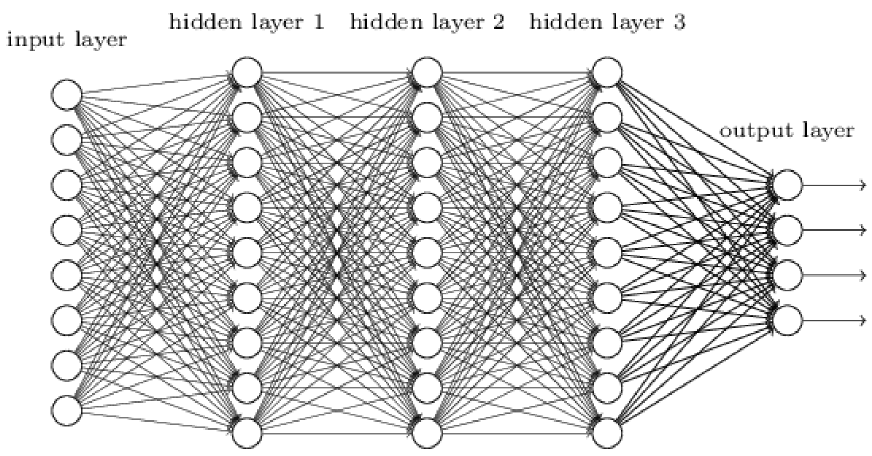
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G. and Petersen, S., 2015. Human-Level Control through Deep Reinforcement Learning. *Nature*, 518(7540), p.529-533.

Deep Q-Network (DQN)

- Problems
  - ➢ (1) Swallow layer

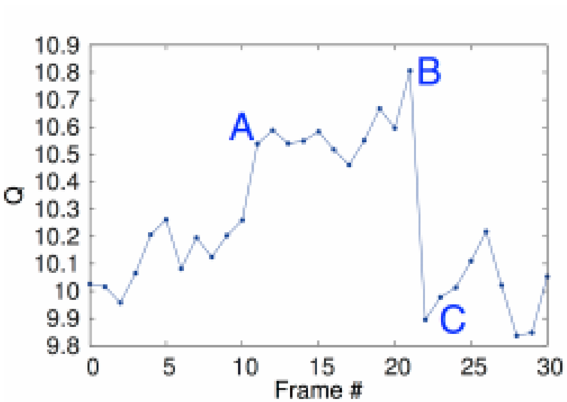


- Solutions
  - ➢ (1) Go deeper

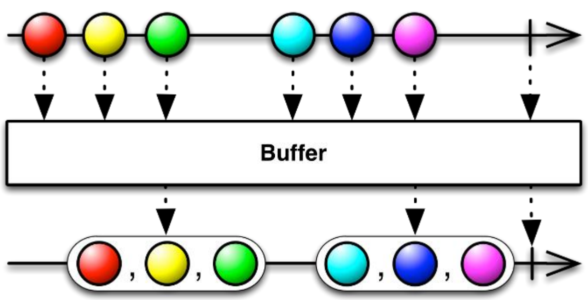


Deep Q-Network (DQN)

- Problems
  - ➢ (2) Highly correlated data



- Solutions
  - ➢ (2) Experience replay



Learning based on random sampling from buffer (experience replay or memory)

Deep Q-Network (DQN)

- Problems
  - ➢ (3) Non-stationary target

Reward

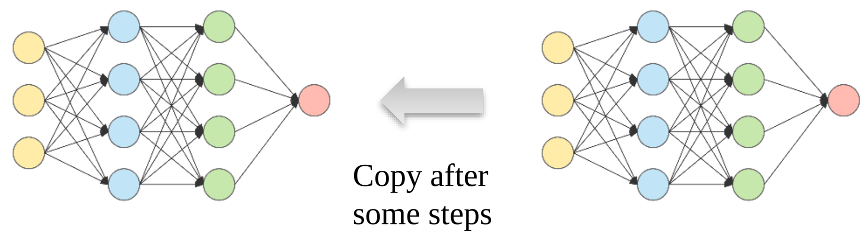
Decay Rate

$$loss = \Big(\underbrace{r + \gamma \max_{a`} \hat{Q}(s, a`)}_{\text{Target}} - \underbrace{Q(s, a)}_{\text{Prediction}}\Big)^2$$

But, the target also changed because they share the same Q-networks

Updating networks to reduce the loss from the target

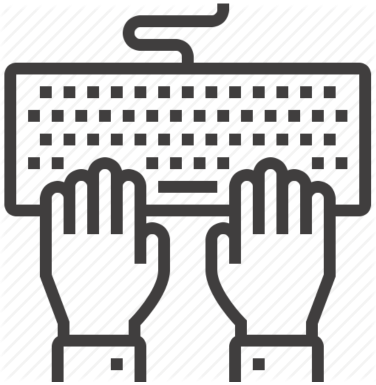
- Solutions
  - ➢ (3) Separating learning and target networks



Target networks

Learning networks

$$loss = \Big(\underbrace{r + \gamma \max_{a`} \hat{Q}(s, a`)}_{\text{Target}} - \underbrace{Q(s, a)}_{\text{Prediction}}\Big)^2$$



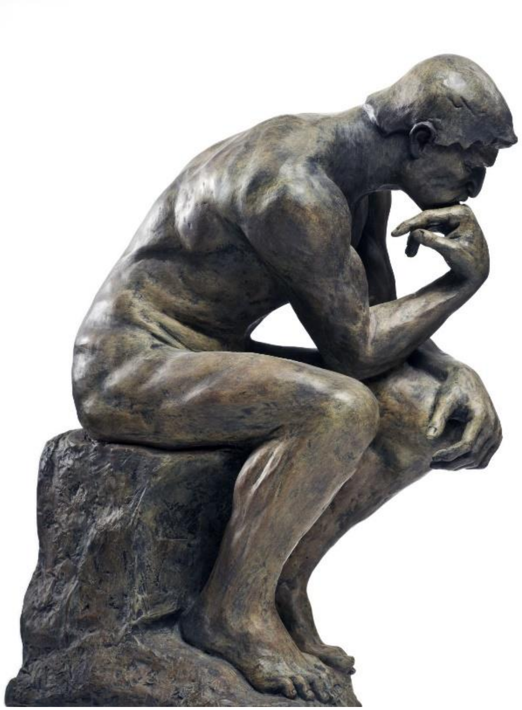
Cart-Pole Problem

M3.7 Deep Q-Network_CartPole.ipynb

Artificial Intelligence

What is Artificial Intelligence?

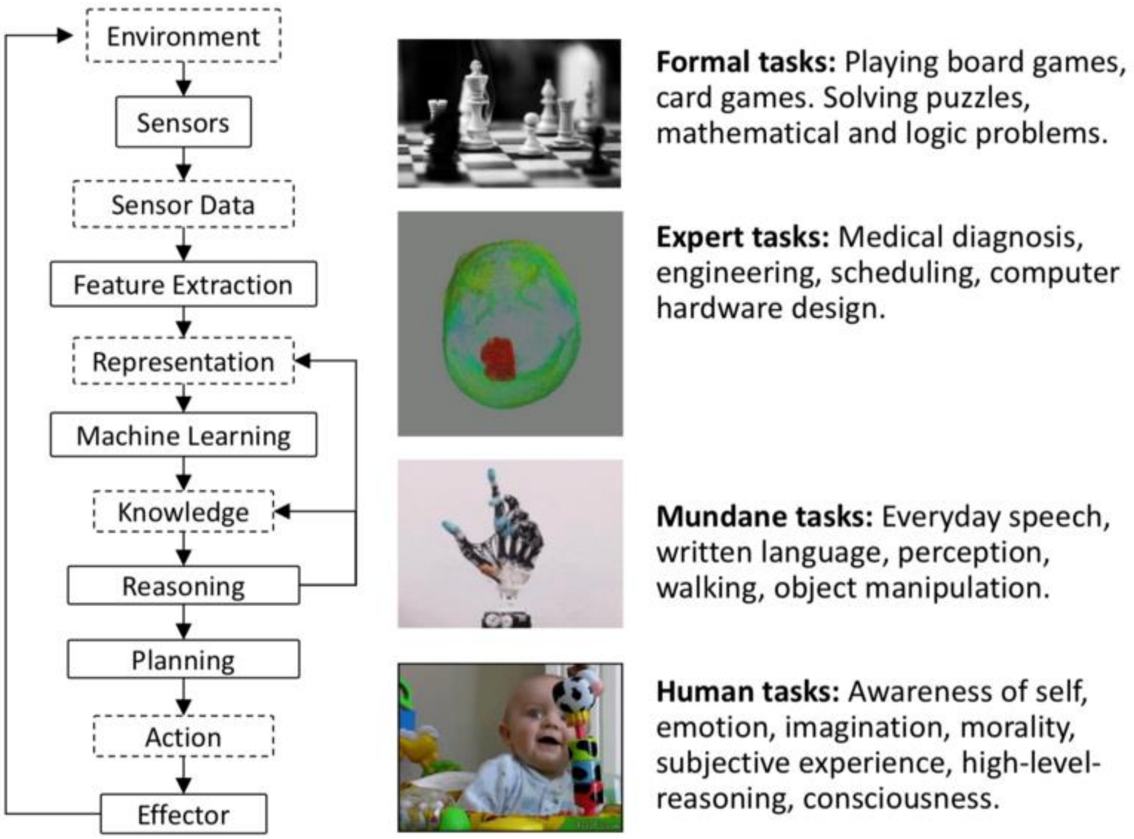
- What is the relationship between deep learning and artificial intelligence?



Artificial Intelligence

- Artificial intelligence is required to take actions given the environments.

  Deep learning algorithms can help each stage of reasoning.

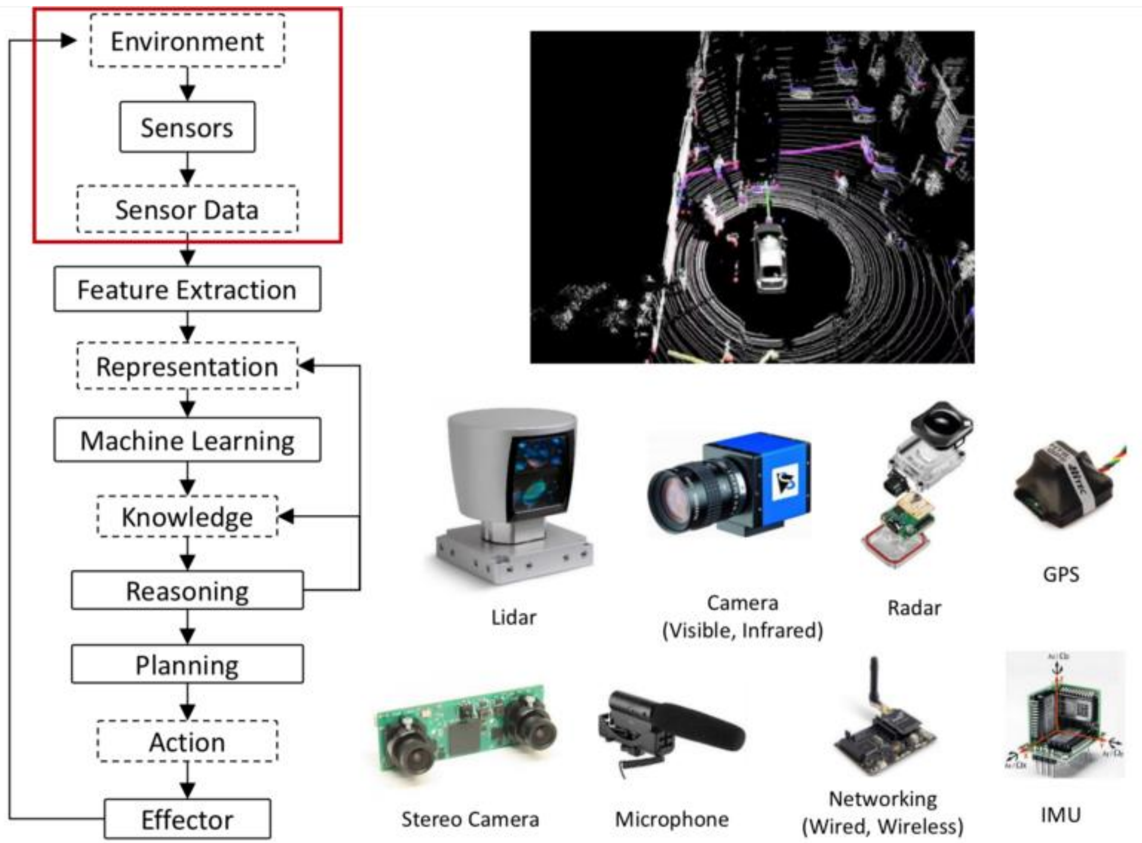


https://hackernoon.com/mit-6-s094-deep-learning-for-self-driving-cars-2018-lecture-3-notes-deep-reinforcement-learning-fe9a8592e14a

Artificial Intelligence

- Artificial intelligence is required to take actions given the environments.

  Various technologies can help each stage of reasoning.

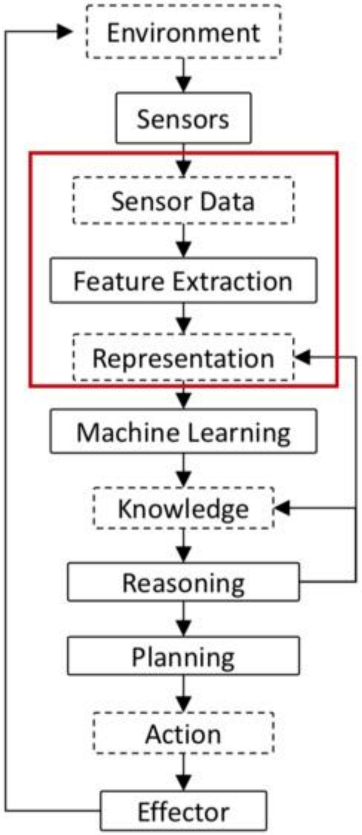


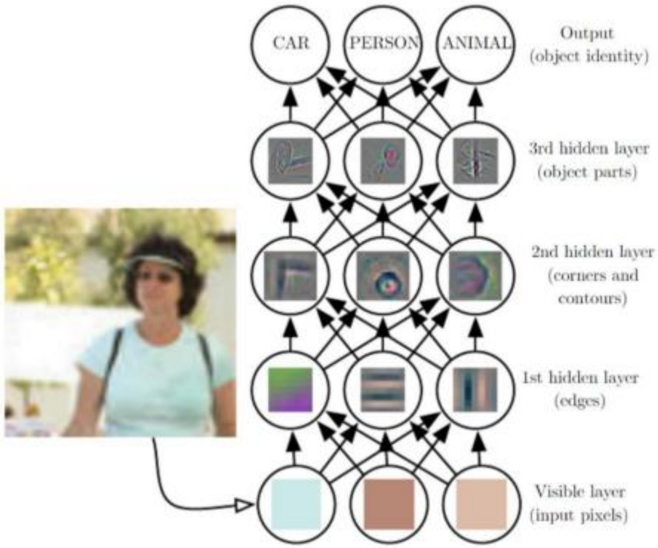
https://hackernoon.com/mit-6-s094-deep-learning-for-self-driving-cars-2018-lecture-3-notes-deep-reinforcement-learning-fe9a8592e14a

Artificial Intelligence

- Artificial intelligence is required to take actions given the environments.

  Various technologies can help each stage of reasoning.



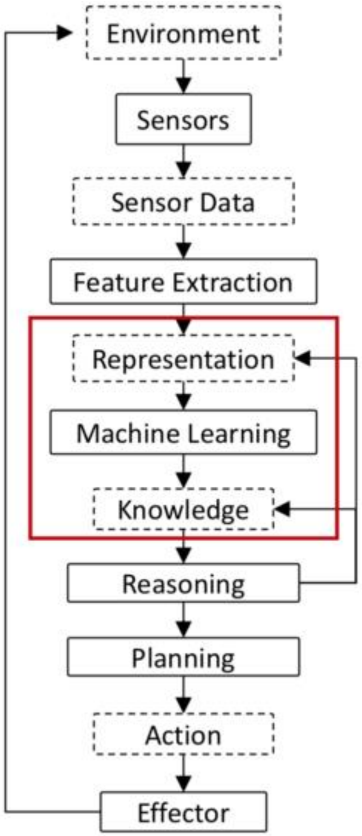


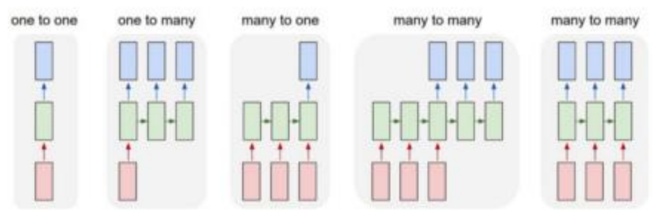
https://hackernoon.com/mit-6-s094-deep-learning-for-self-driving-cars-2018-lecture-3-notes-deep-reinforcement-learning-fe9a8592e14a

Artificial Intelligence

- Artificial intelligence is required to take actions given the environments.

  Various technologies can help each stage of reasoning.



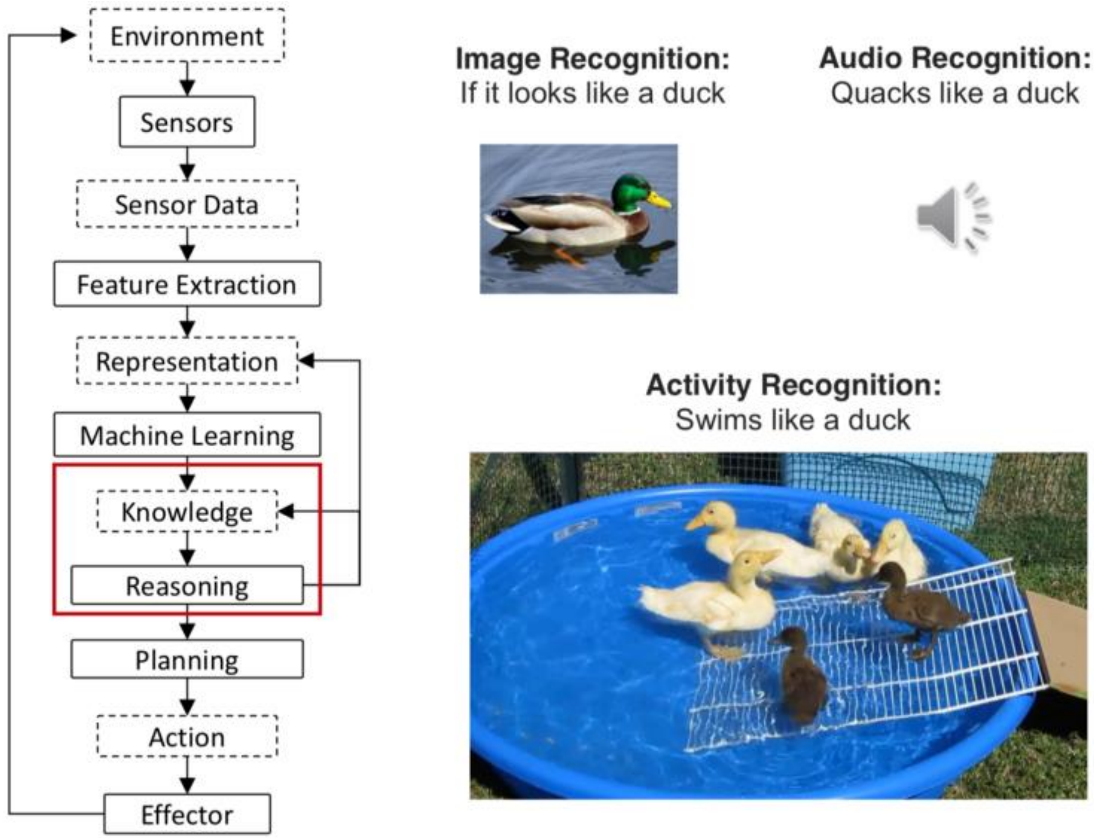


https://hackernoon.com/mit-6-s094-deep-learning-for-self-driving-cars-2018-lecture-3-notes-deep-reinforcement-learning-fe9a8592e14a

Artificial Intelligence

- Artificial intelligence is required to take actions given the environments.

  Various technologies can help each stage of reasoning.

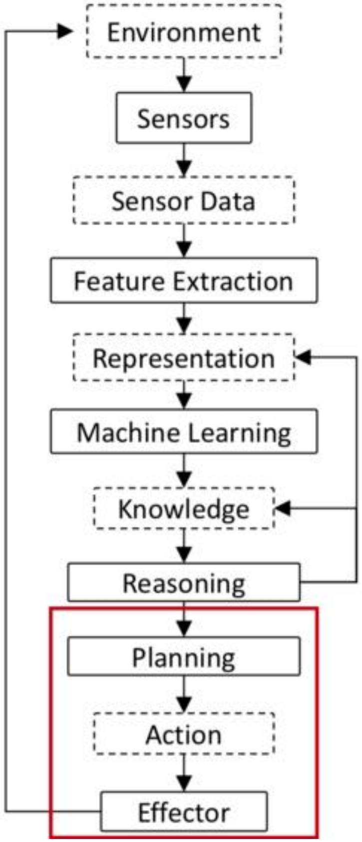


https://hackernoon.com/mit-6-s094-deep-learning-for-self-driving-cars-2018-lecture-3-notes-deep-reinforcement-learning-fe9a8592e14a

Artificial Intelligence

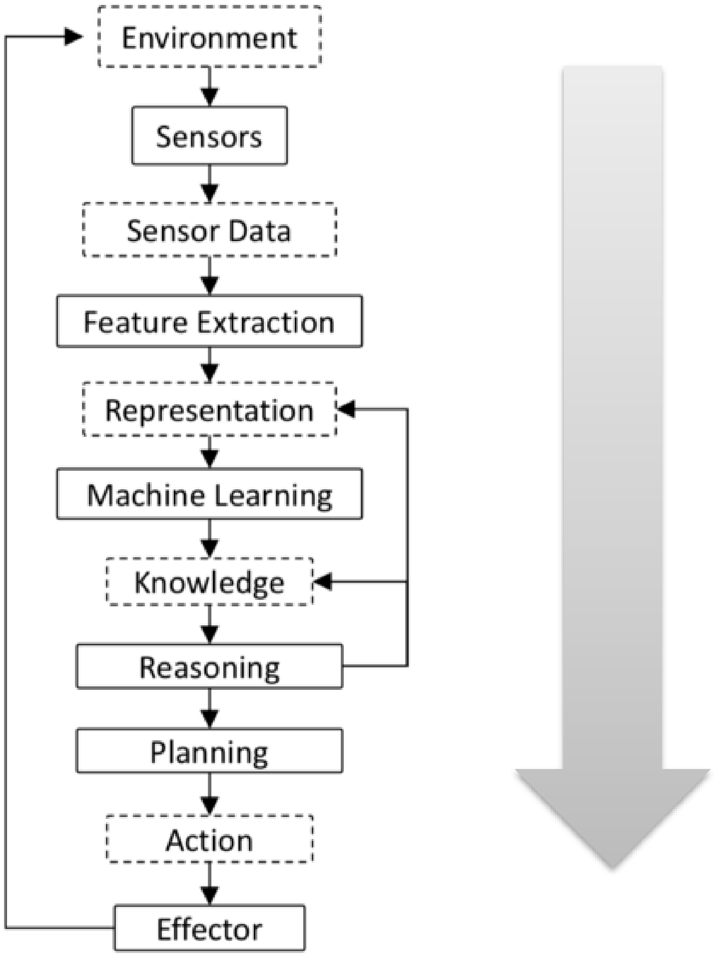
- Artificial intelligence is required to take actions given the environments.

  Various technologies can help each stage of reasoning.



https://hackernoon.com/mit-6-s094-deep-learning-for-self-driving-cars-2018-lecture-3-notes-deep-reinforcement-learning-fe9a8592e14a

Is it Possible a End-to-End Reasoning?



Learning what action should be taken, based on the information directly from the environments

Surprising Power of Reinforcement Learning

- Google DeepMind's Deep Q-learning playing Atari Breakout (2015)

- AlphaGo (2016)
  - ➢ Supervised learning + Reinforcement learning
- AlphaZero (2017)
  - ➢ Only reinforcement learning (learning without prior knowledge)

https://deepmind.com/

Surprising Power of Reinforcement Learning

- Reinforcement learning can be applied widely in business areas.
  - Inventory management (When and how much firms need to order?)

    What are rewards?

    To decrease operational costs

  - Resource allocation (How should organizations allocate the resources?)

    What are rewards?

    To increase profits

  - Marketing strategy (When and how should marketers offer mobile coupons?)

    What are rewards?

    To increase redemption rate

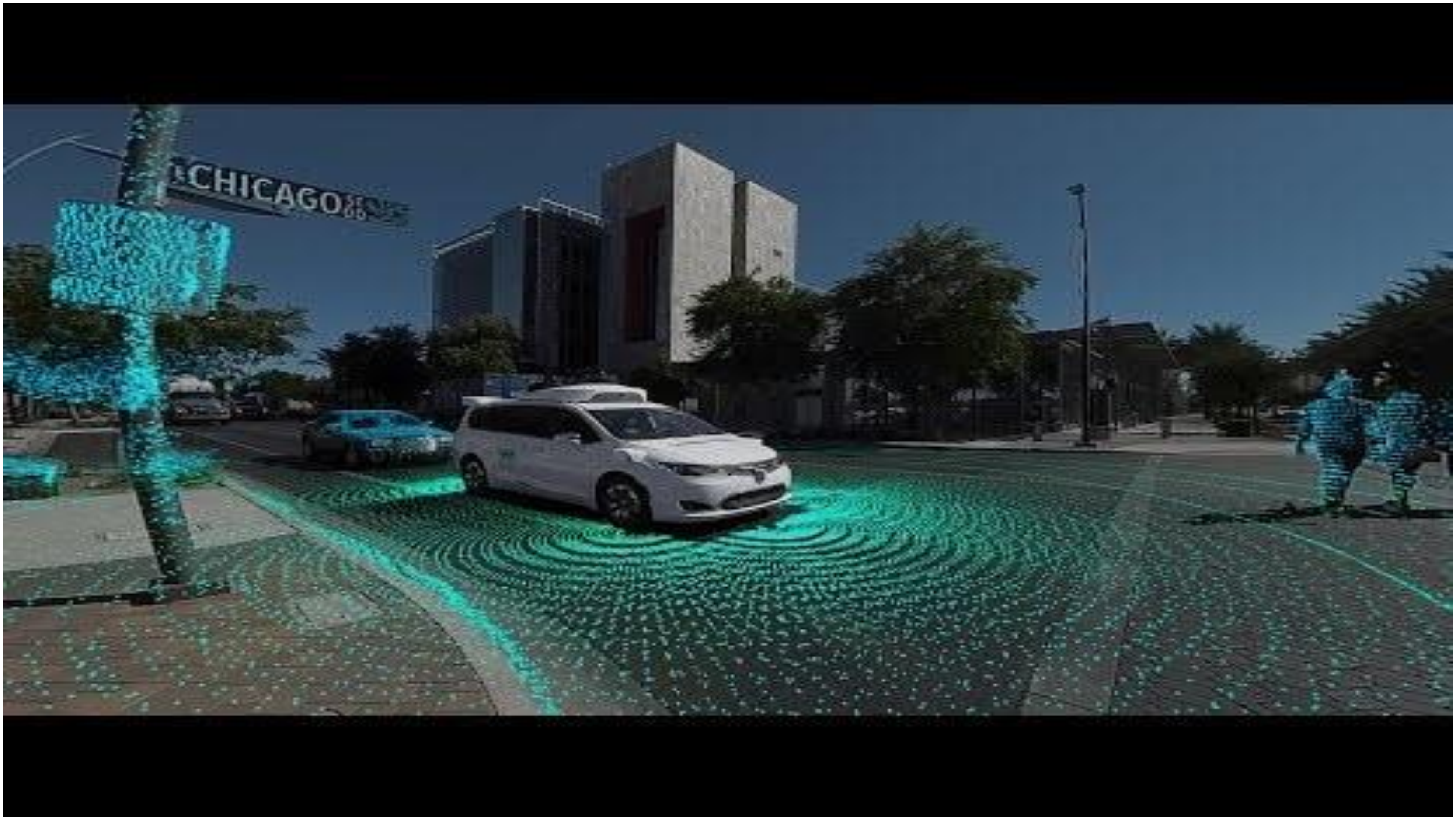
  - Investment strategy (How can investors design their portfolios?)

    What are rewards?

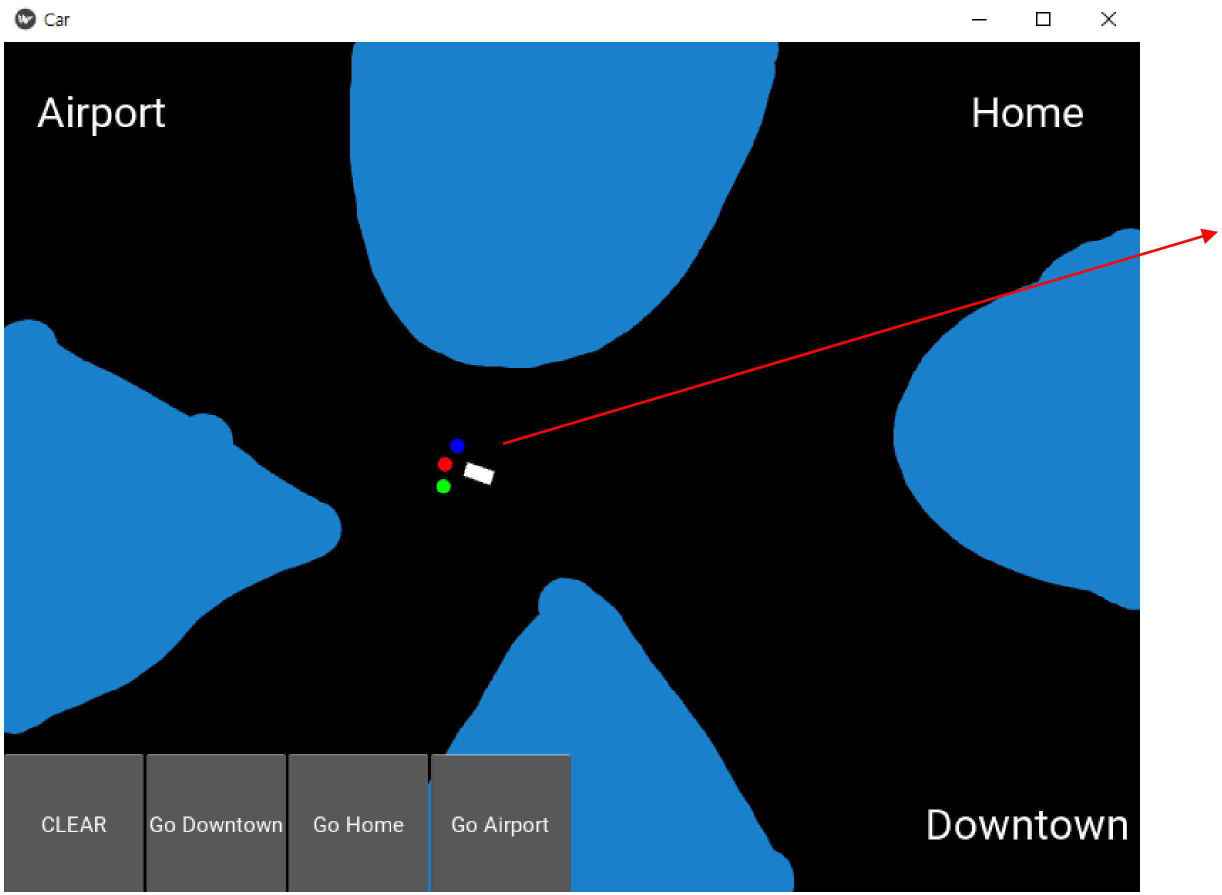
    To increase returns on investment

Project for Artificial Intelligence

Self-Driving Car



Let’s Make a Simple Self-Driving Car



The car has three sensors (balls) for obstacle detection and one angle sensor.

Based on the signals, the car adjusts its rotation and velocity to reach a destination.

Let’s Make a Simple Self-Driving Car

- Instruction for the project
  - (1) Install the following dependencies in an Anaconda Prompt

    pip install kivy

    pip install docutils pygments pypiwin32 kivy.deps.sdl2

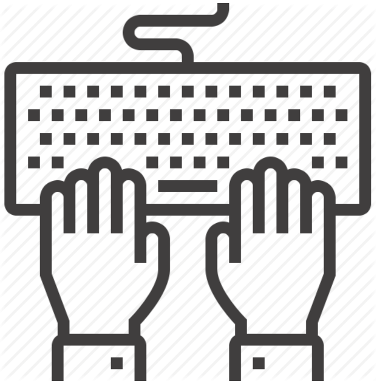
    pip install kivy.deps.glew

    (If any problems, open an Anaconda prompt window as administrator)

  - (2) There are five exercises in two files (main.py and DQN.py)

  - (3) To run the program, type the following in an Anaconda Prompt

    python main.py

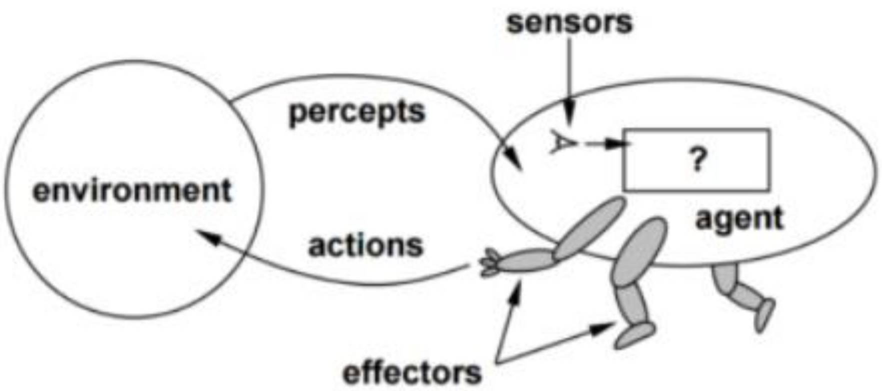


Self-Driving Car

M3.7 Deep Q-Network_Self-Driving Car

Things to Ponder

Open Question:
What can we **not** do with Deep Learning?



End of Document