Deep Reinforcement Learning

school of ai

Beril Sirmacek

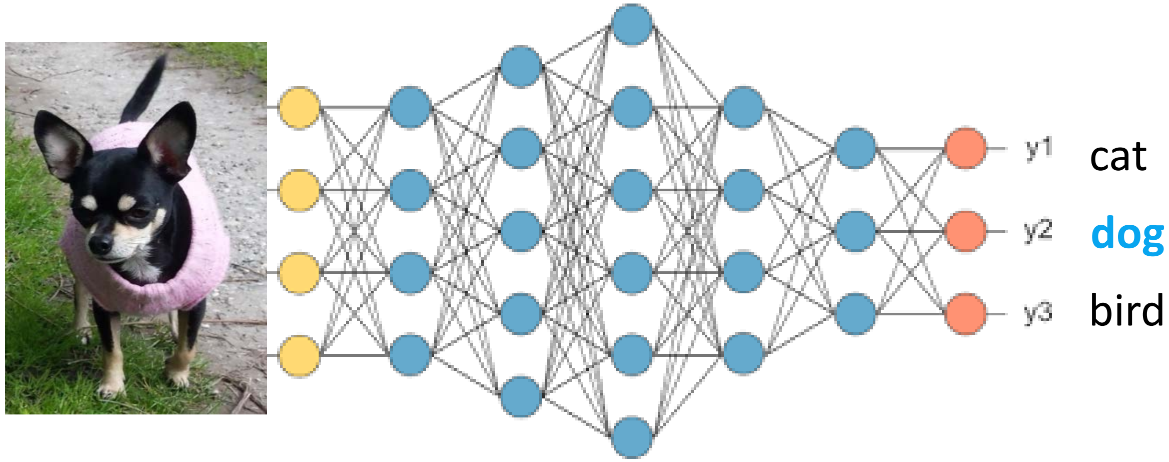
7th March 2019, Enschede

Outline

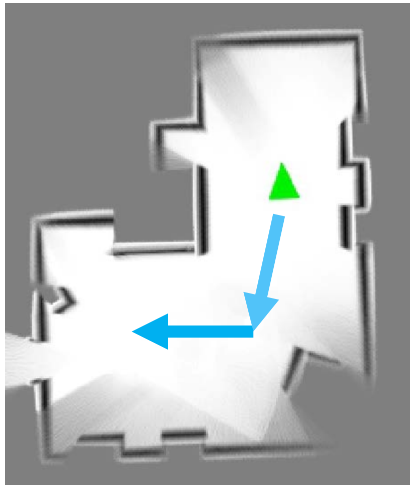
- Reinforcement Learning (RL)
- Deep reinforcement learning (DRL)
- Practical applications of RL

What is reinforcement learning?

- Similarity with other deep learning nets and RL



Supervised learning

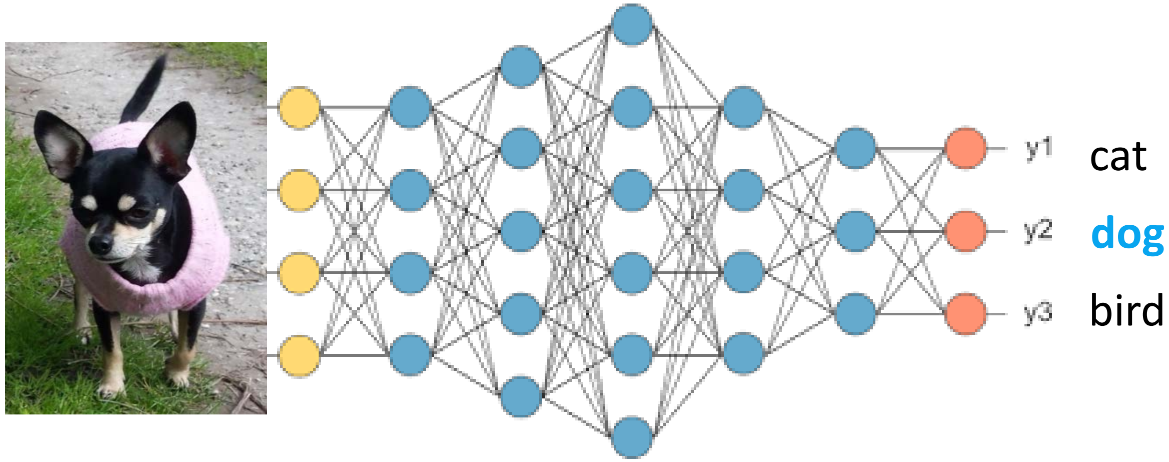


Reinforcement learning

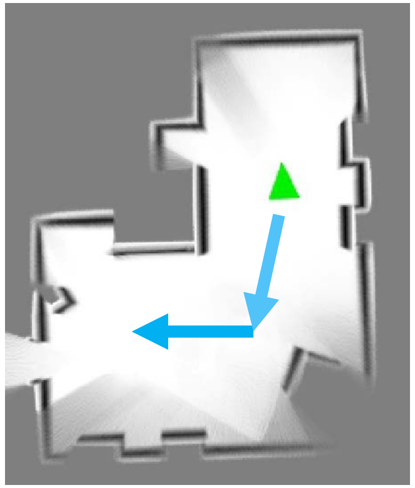
decision making!

What is reinforcement learning?

- Difference from other deep learning nets and RL



Supervised learning



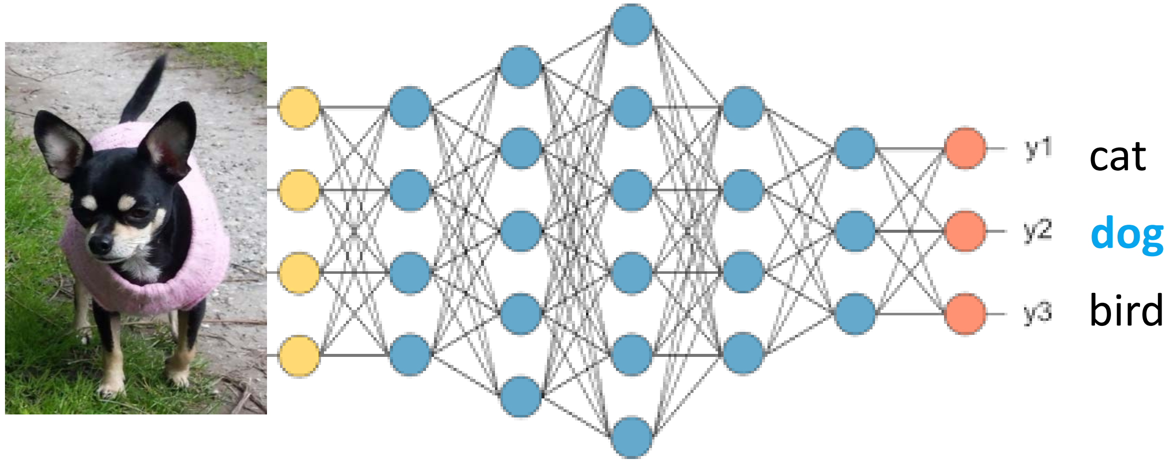
Reinforcement learning

A chain of decisions (actions) & dependency on the previous decisions

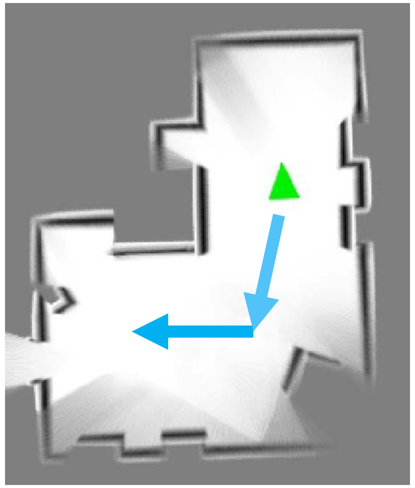
What is reinforcement learning?

- Difference from other deep learning nets and RL

Where is UNSUPERVISED LEARNING?



Supervised learning



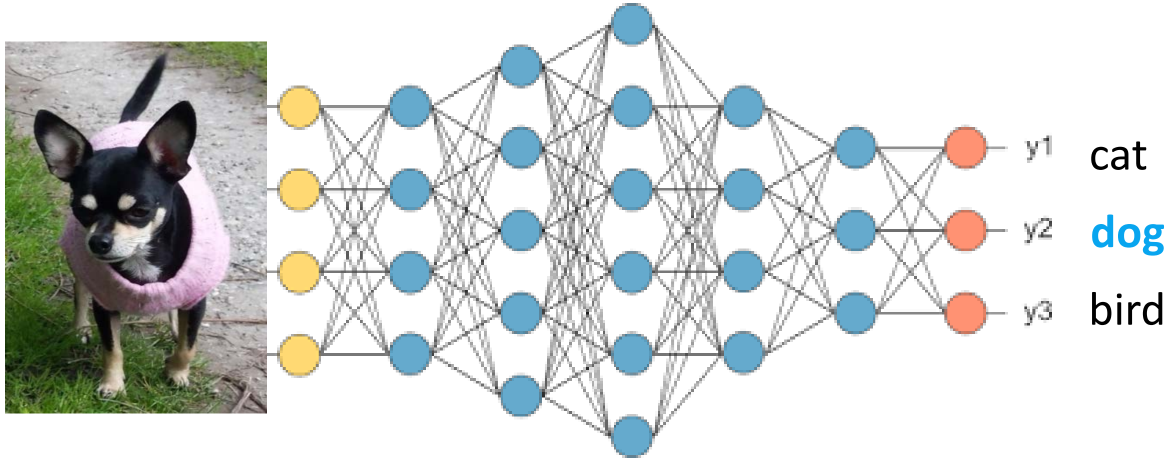
Reinforcement learning

A chain of decisions (actions) & dependency on the previous decisions

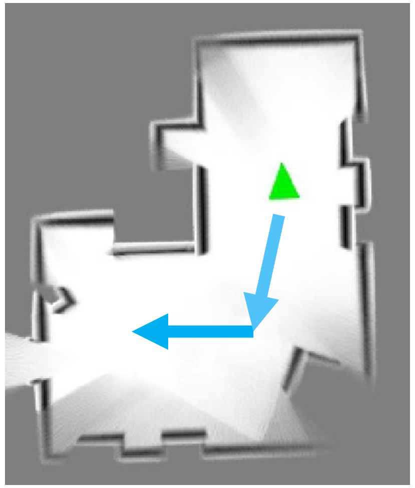
What is reinforcement learning?

- Here are the examples / Loss function
- Learn it (learning by example)

- Here is the environment
- Learn it (learning by experience)



Supervised learning

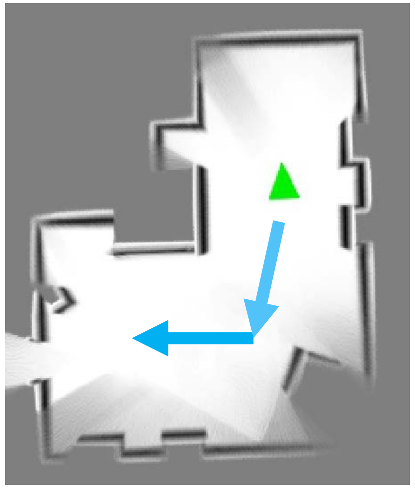


Reinforcement learning

Problem description

- What series of actions will give me the highest reward?

(or helps me to reach to my goal faster/safer/better?)



Reinforcement learning

Problem formulation

Agent

Environment

Actions

Problem formulation

Agent

Environment

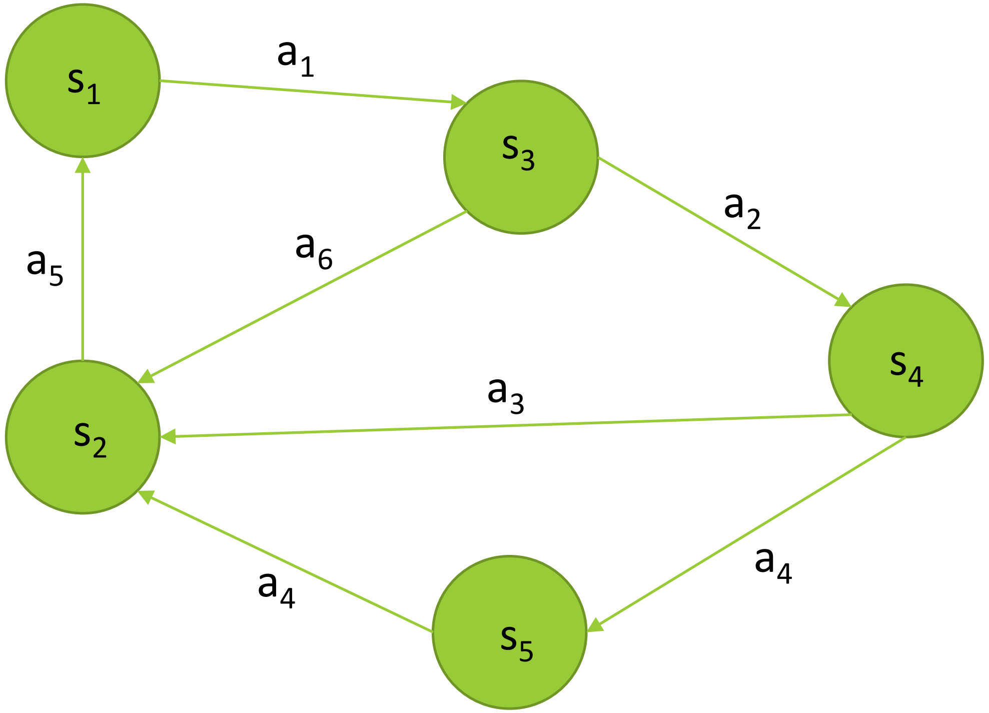
Actions

How to decide which actions to take?

Policy

Mathematical formulation

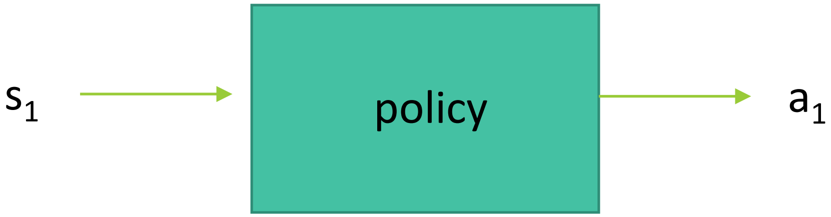
Agent



States: s_1, s_2, s_3, s_4,…
Actions: a_1, a_2, a_3, …
Rewards: $r(s_n,a_m)$

Mathematical formulation

Agent



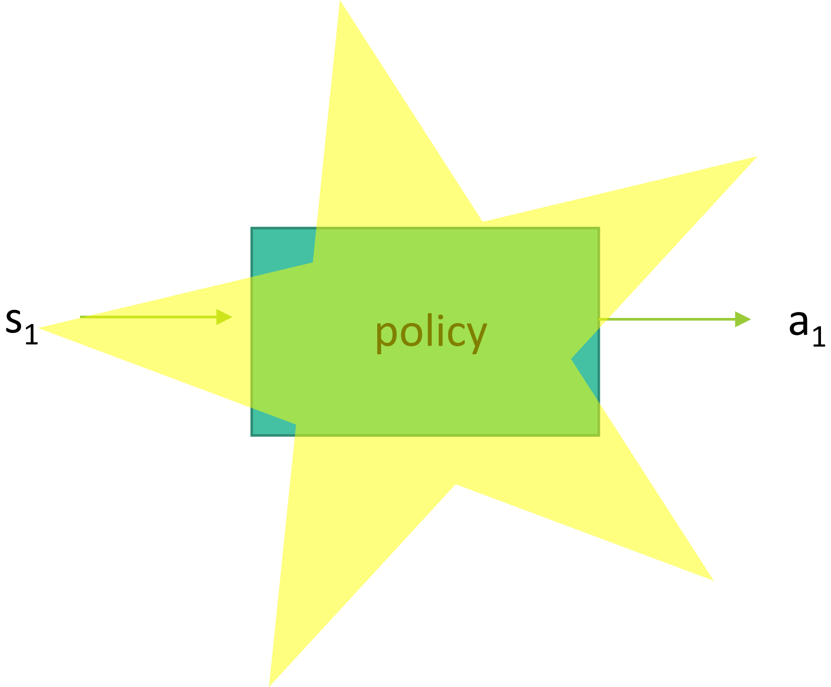
s_1 → policy → a_1

Policy: Action that determines the next state.

Reward: Measure of the outcome of taking an action.

Mathematical formulation

Agent



Choosing the ***optimal policy***?

Choosing the optimal policy

Agent

Greedy strategy:

Choosing the optimal policy

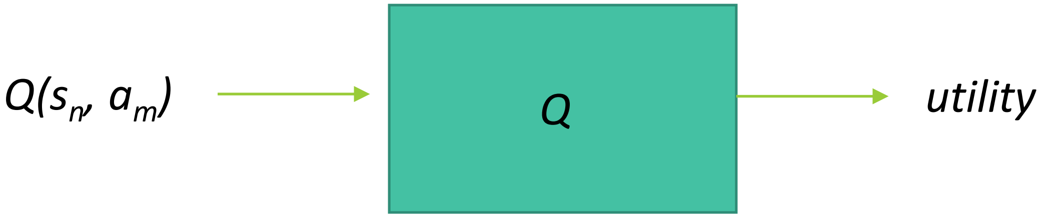
Agent

Random strategy:

Multiple random scenarios should be tested before choosing the next action.

Mathematical formulation

Agent



$Q(s_n, a_m)$

Q

utility

Utility: The long term reward is called a utility.

The utility of performing an action a at *a* state *s* is written as a function *Q(s, a)*, called the utility function.

Mathematical formulation

Agent

If you were given the utility function Q(s,a), how would you use it to derive a policy function?

Mathematical formulation

Agent

If you were given the utility function Q(s,a), how would you use it to derive a policy function?

$Policy(s) = argmax_a\ Q(s_n\ ,\ a_m)$

Mathematical formulation

Agent

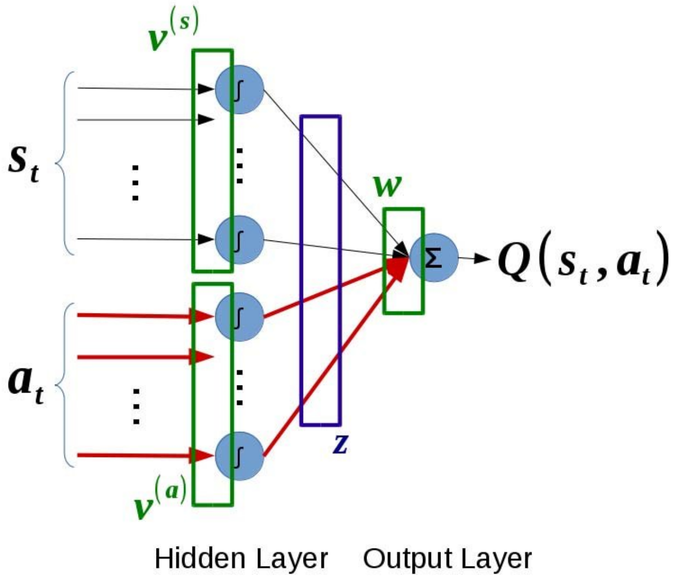
$$Q(s,a) = r(s,a) + \gamma\, Q(s',a')$$

Where is the deep part?

Agent

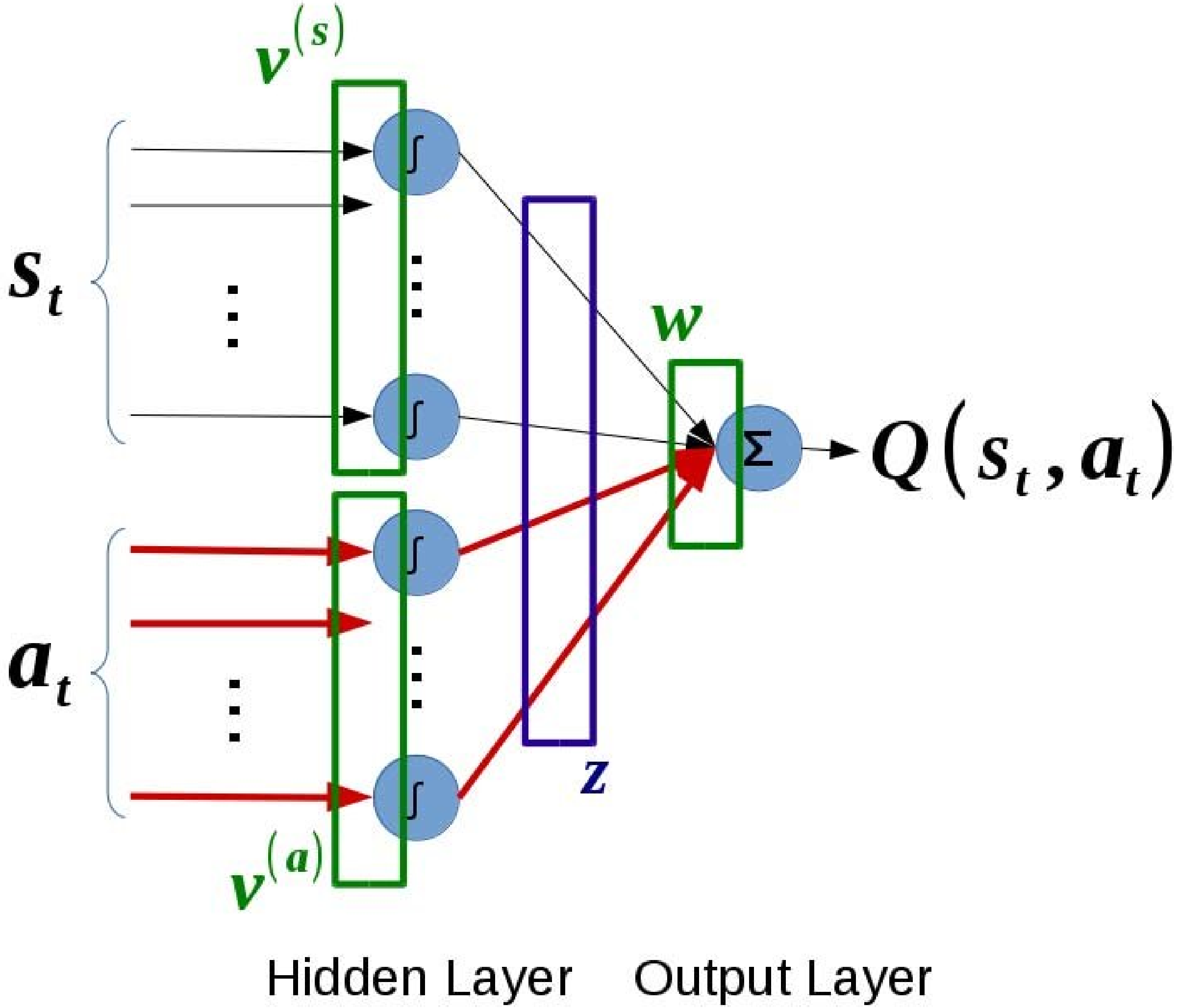
If you were given the utility function Q(s,a), how would you use it to derive a policy function?

Policy(s) = argmax$_a$ Q(s$_n$, a$_m$)

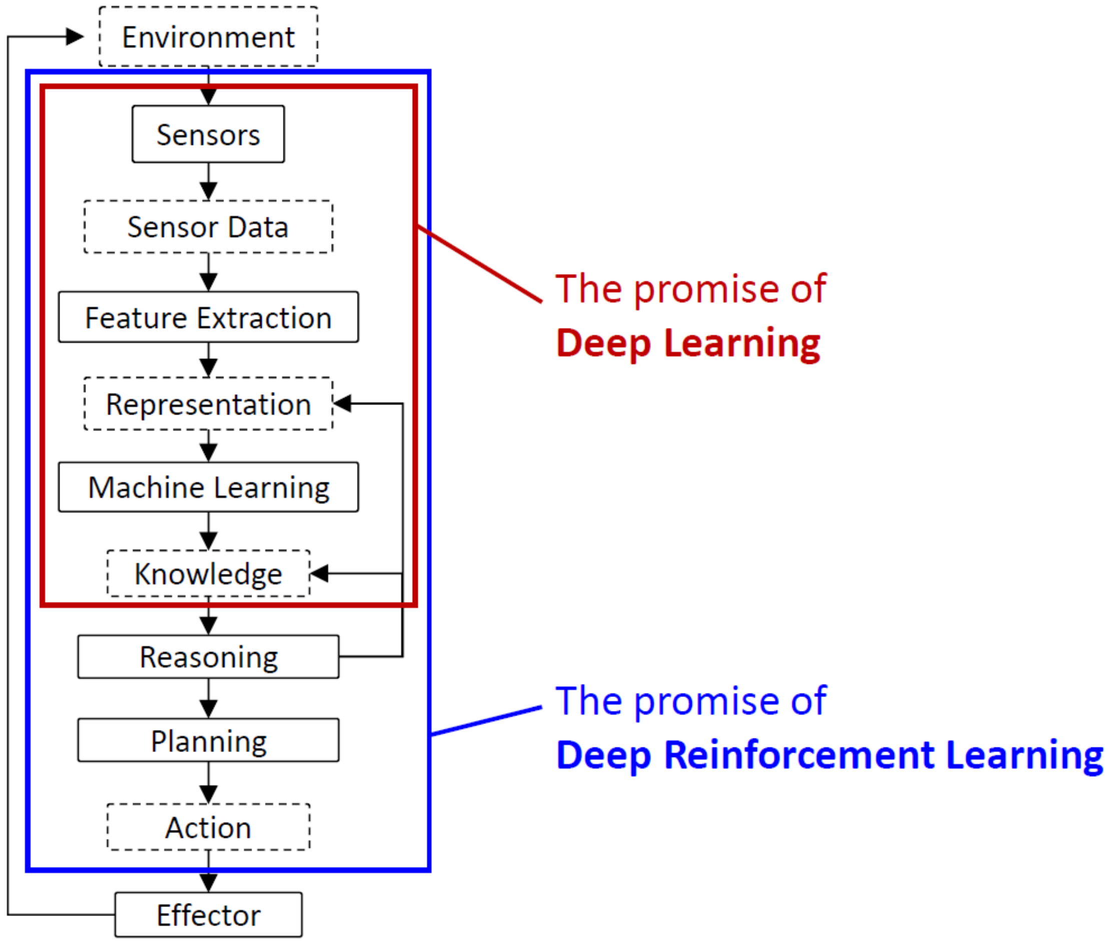


Where is the deep part?

Agent



Where is the deep part?



The promise of
Deep Learning

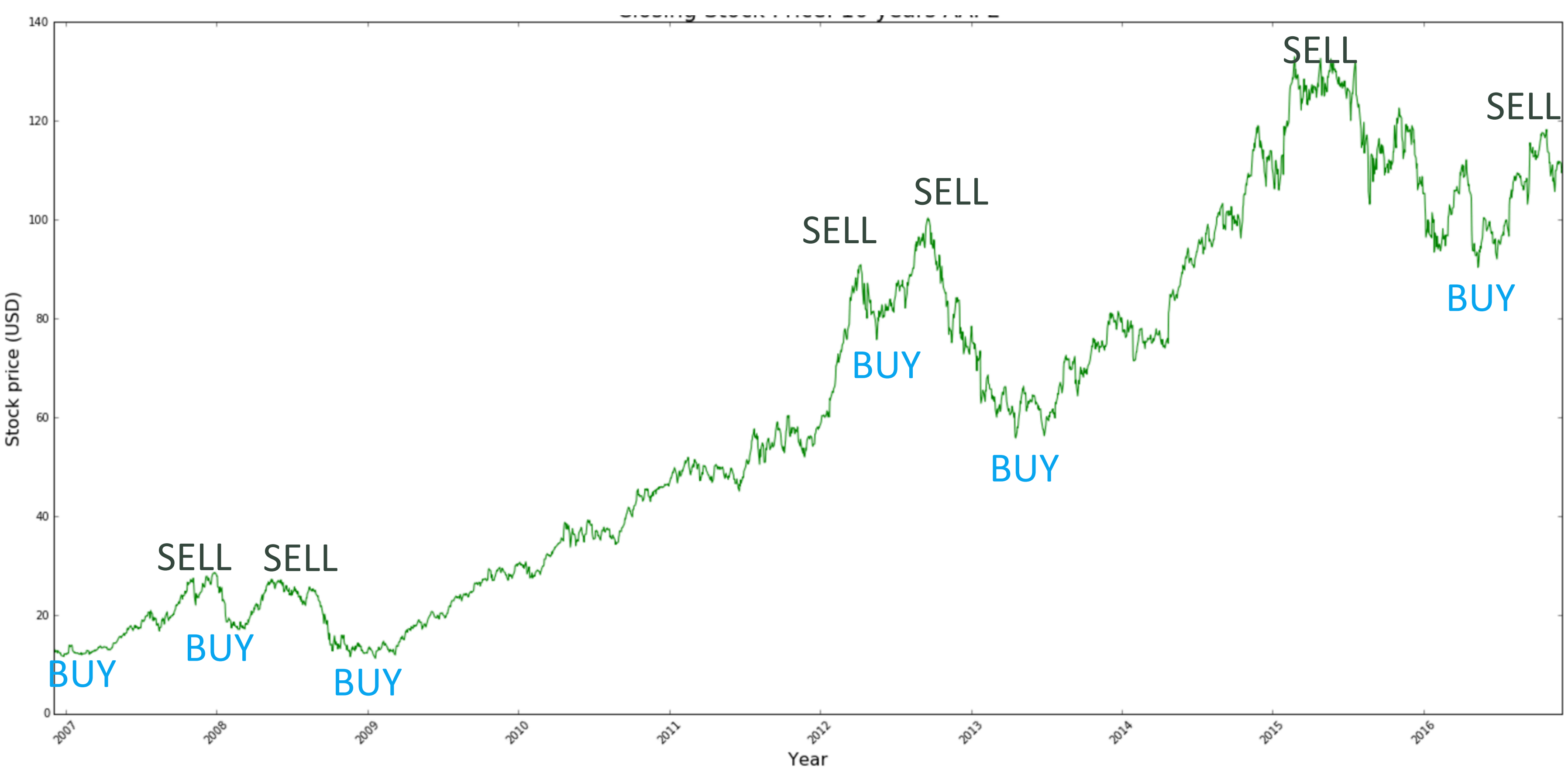
The promise of
Deep Reinforcement Learning

https://deeplearning.mit.edu

Do humans use deep reinforcement learning?

Practice

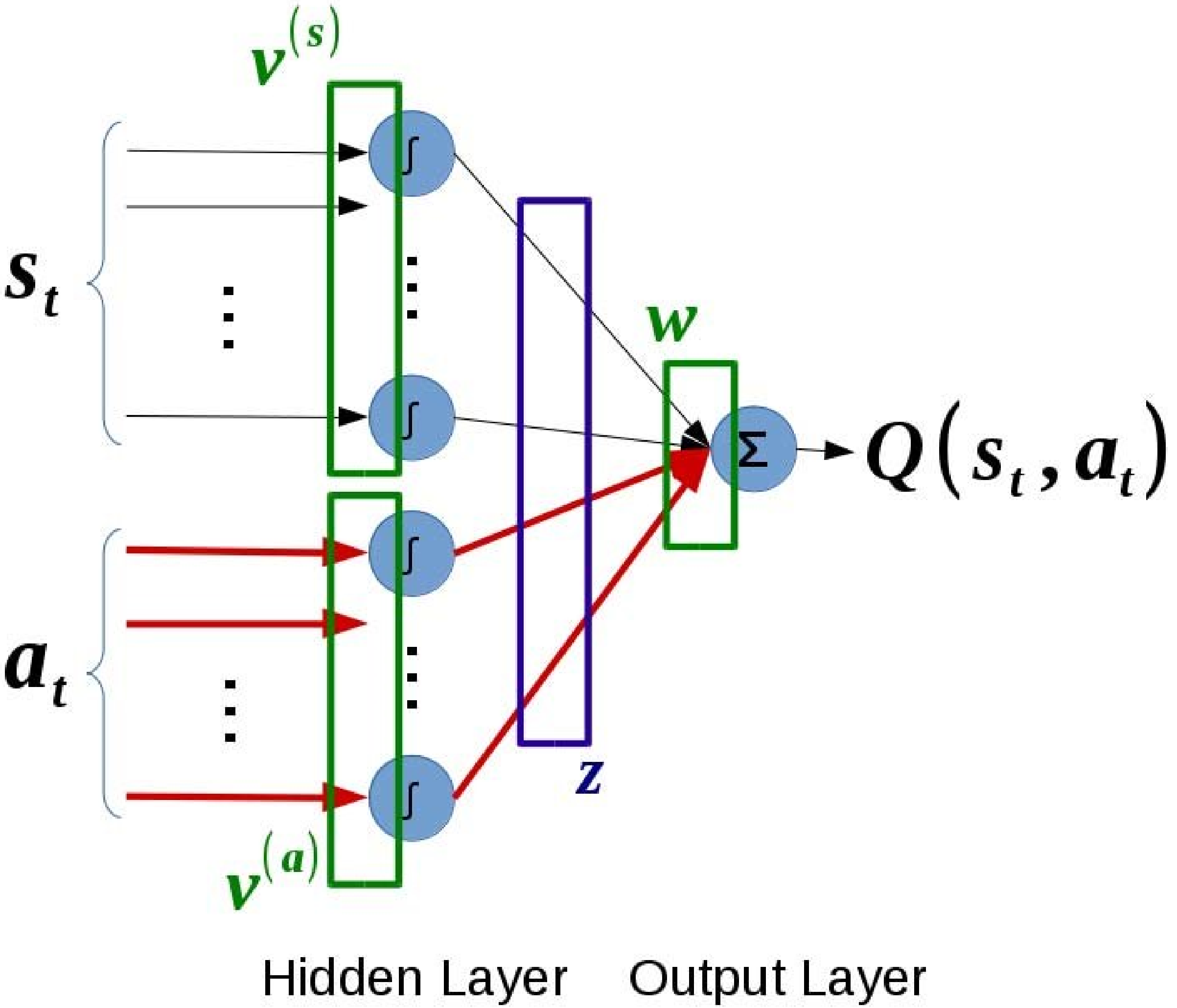
```
$ pip install yahoo-finance
```



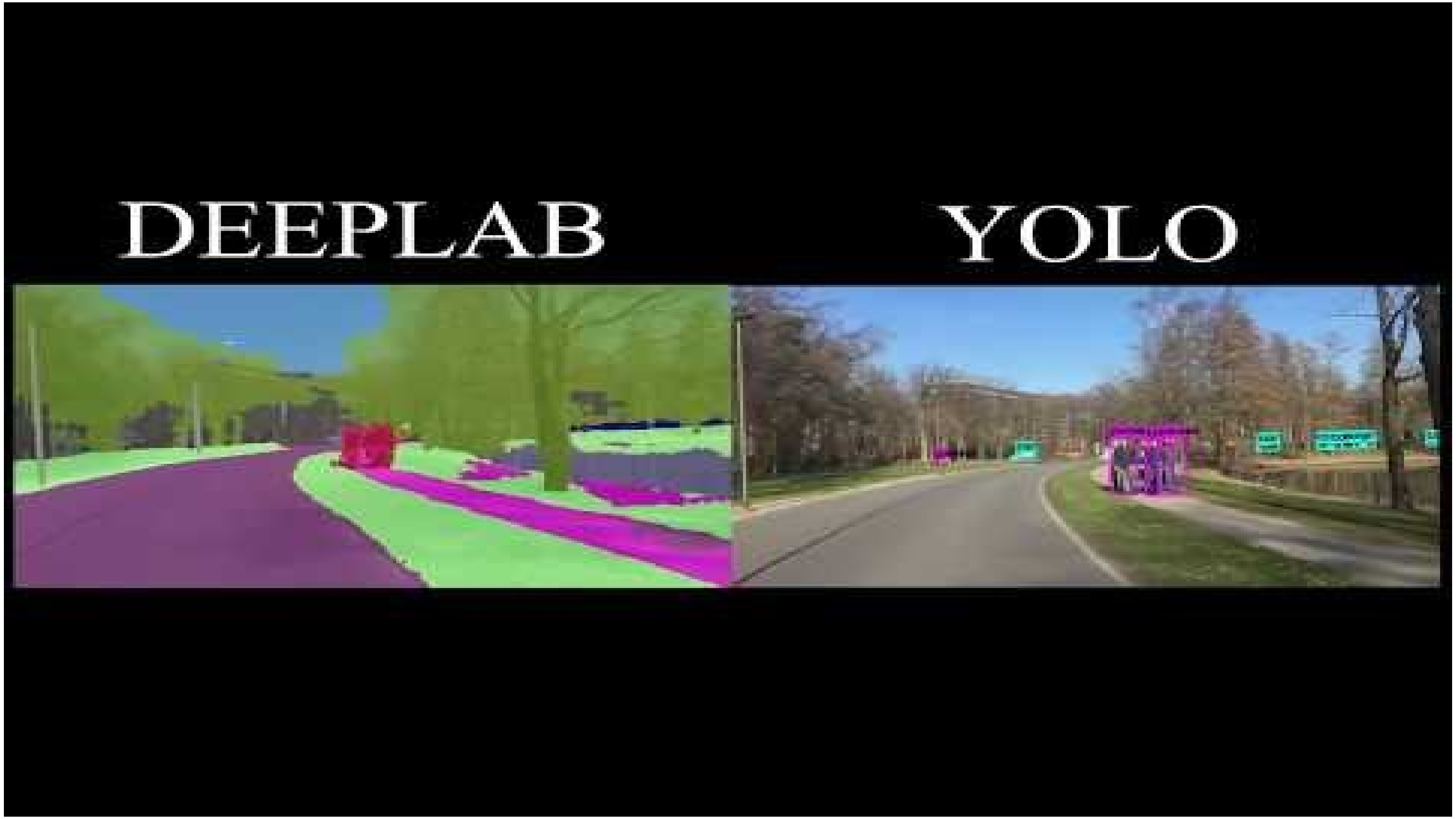
Practice

Last 10 days history of the prices

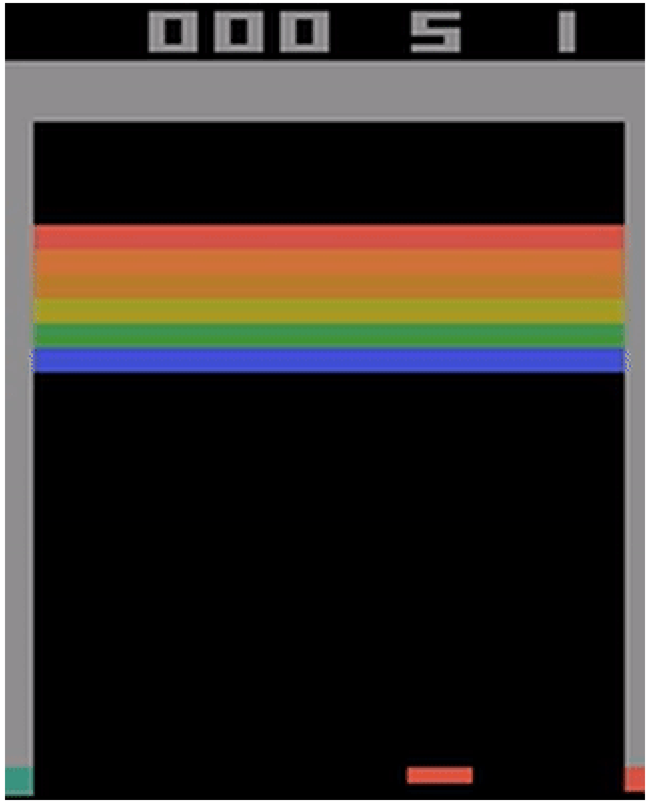
Budget Shares



Other applications of reinforcement learning

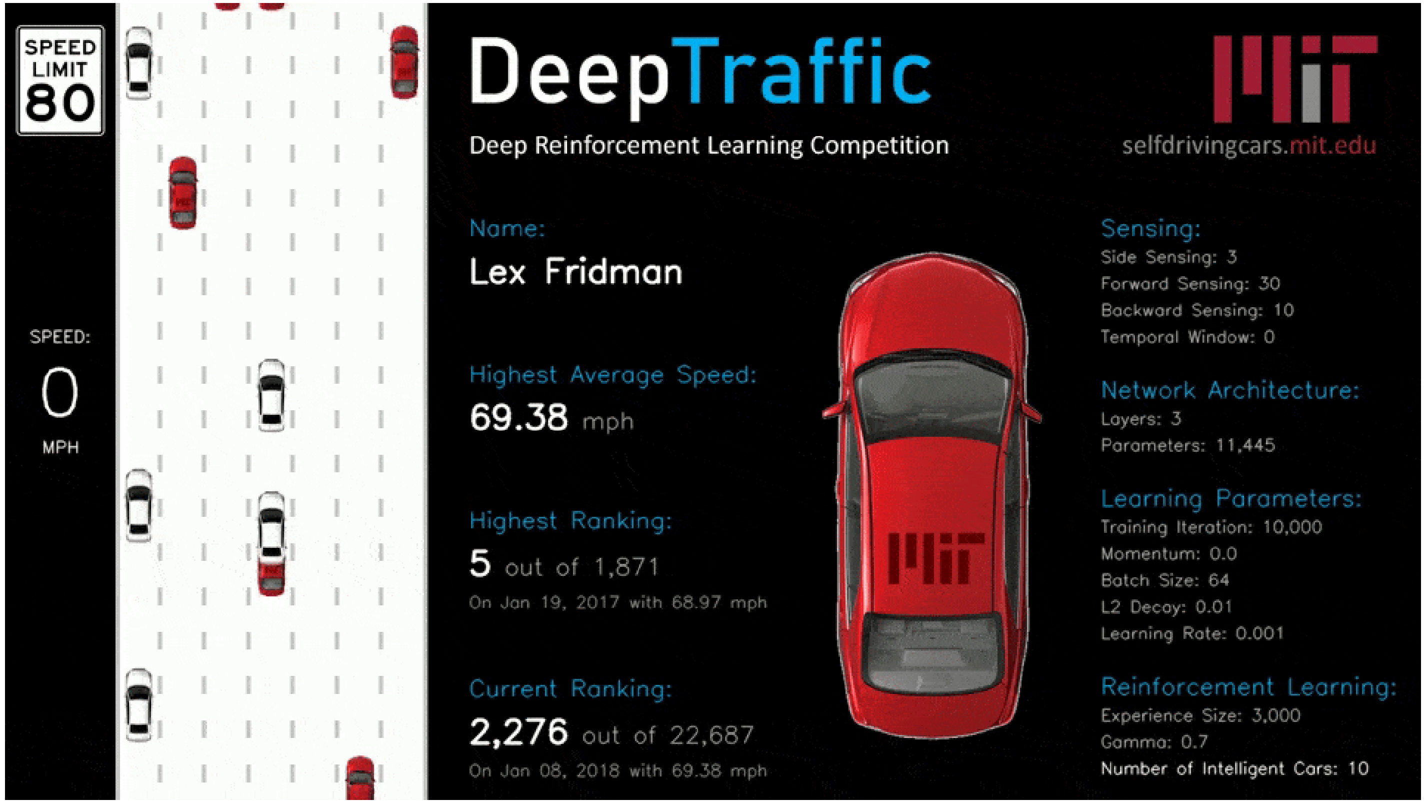


Keras RL



https://github.com/keras-rl/keras-rl

Deep Traffic



DeepTraffic

Deep Reinforcement Learning Competition

selfdrivingcars.mit.edu

Name:
Lex Fridman

Highest Average Speed:
69.38 mph

Highest Ranking:
5 out of 1,871
On Jan 19, 2017 with 68.97 mph

Current Ranking:
2,276 out of 22,687
On Jan 08, 2018 with 69.38 mph

Sensing:
Side Sensing: 3
Forward Sensing: 30
Backward Sensing: 10
Temporal Window: 0

Network Architecture:
Layers: 3
Parameters: 11,445

Learning Parameters:
Training Iteration: 10,000
Momentum: 0.0
Batch Size: 64
L2 Decay: 0.01
Learning Rate: 0.001

Reinforcement Learning:
Experience Size: 3,000
Gamma: 0.7
Number of Intelligent Cars: 10