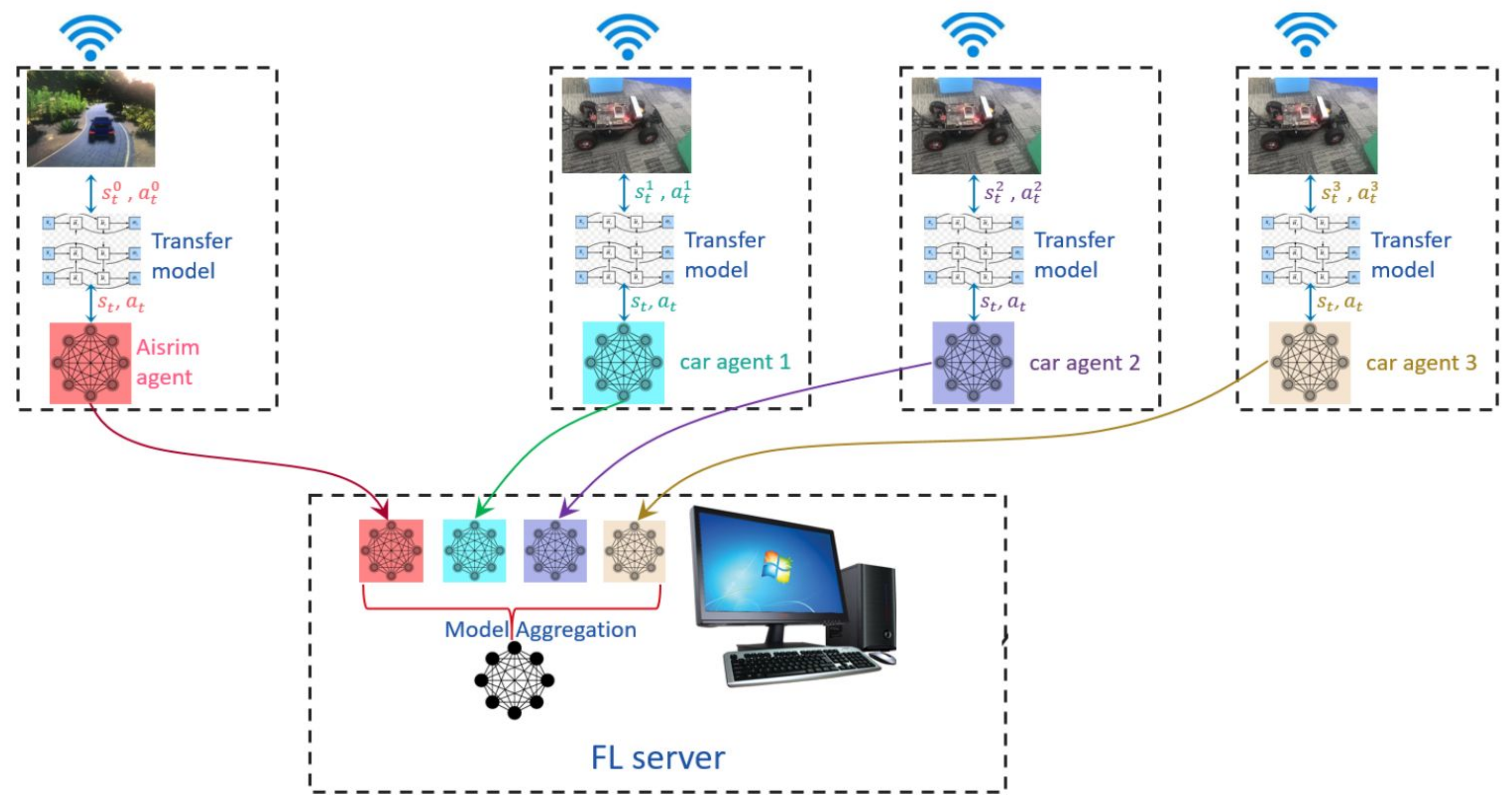
Federated Transfer Reinforcement Learning

Mikel de la Fuente
Maider Amuchastegui
Jesús Rugarcía

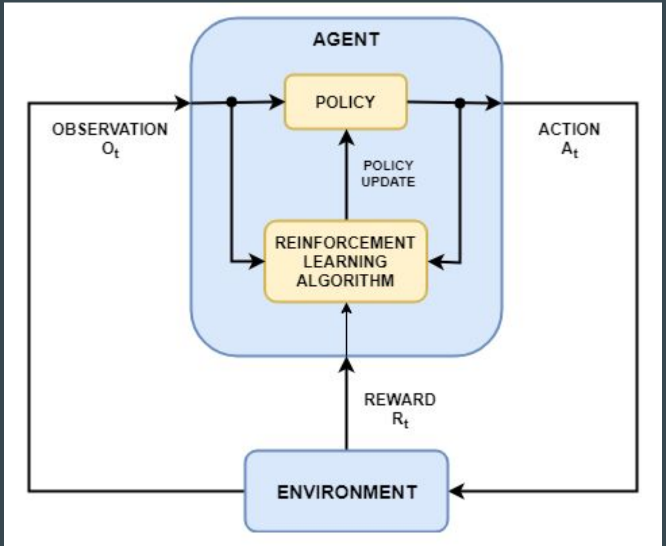
Contents

Introduction



Reinforcement Learning

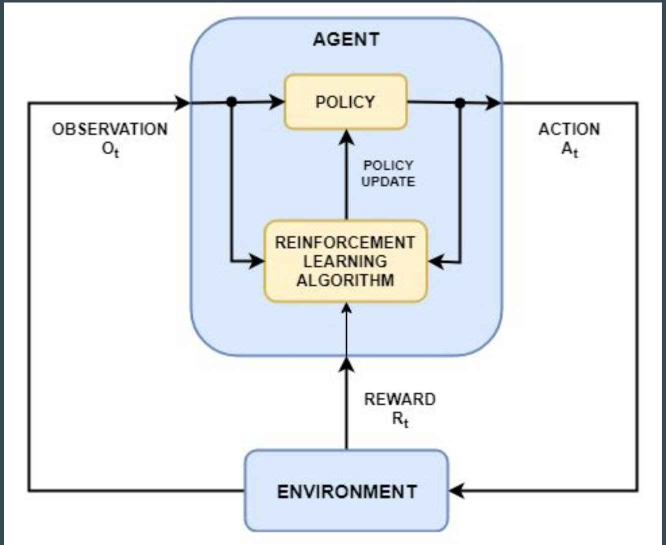
- A paradigm of ML
- Tries to solve problems by trial and error
- Useful for long term problems in complex environments



credit: MathWorks

Reinforcement Learning

- The programmer defines a reward policy
- RL agents perceive their environment
- The action policy determines actions
- The reward policy makes "successful" actions be repeated more often



credit: MathWorks

Reinforcement Learning in autonomous driving

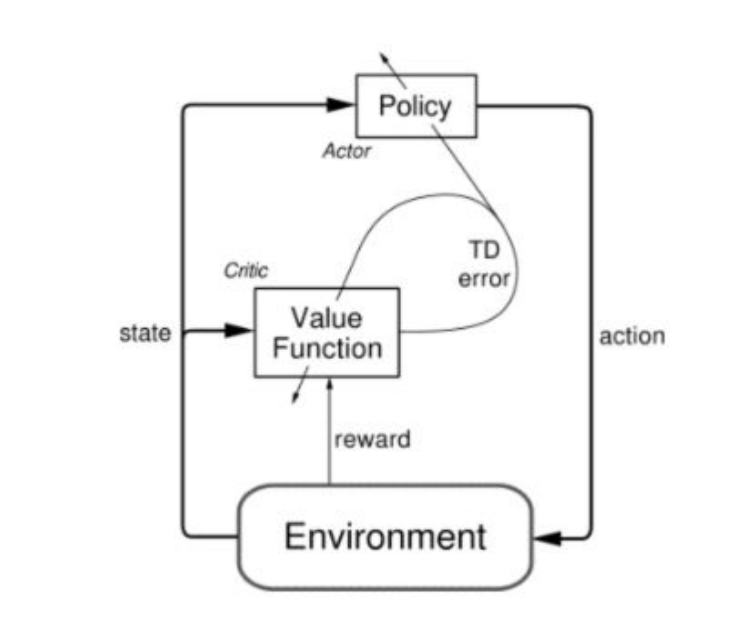
Deep Deterministic Policy Gradient is often used

The environment is measured in a discrete way

An actor chooses actions to take based on its policy, which takes the environment state into account. Actions have three dimensions :

- Acceleration, valued from 0(no acceleration) to 1(full acceleration).
- Brake, value from 0(no brake) to 1(full brake).
- Steering, valued from -1 (max steering to the left).

The critic uses a function to assess its performance on each dimension and give a reward



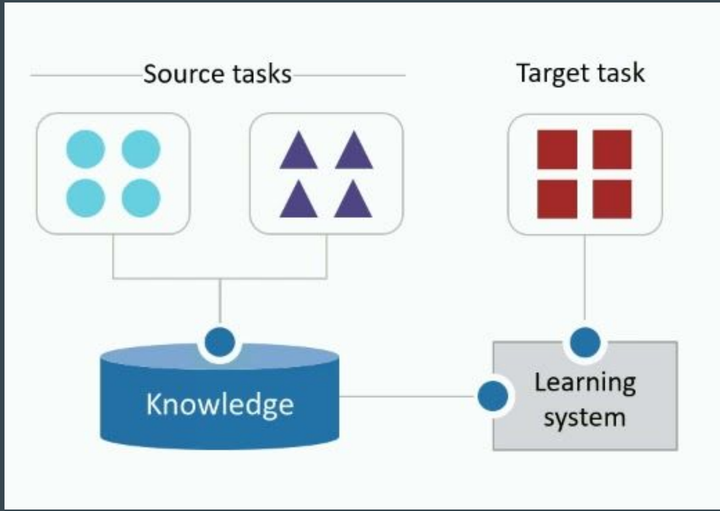
Transfer Learning (TL)

- Research problem in ML.

- It’s store knowledge gained while solving one problem and applying it to a different but related problem.

- For example: In a training to detect food in images, it's use the knowledge gained to recognize drinks.

- The goal: Improve learning in the target task by leveraging knowledge from the source task.



- TL and RL are related. The importance of TL on RL:

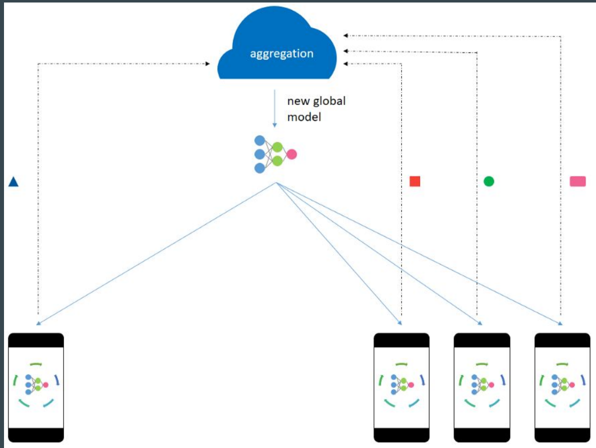
    1. RL techniques have achieved remarkable successes in difficult tasks.
    2. Classic ML techniques are mature enough that they can now be easily leveraged to help with TL.
    3. Results show that these transfer methods are possible and can be very effective in accelerating learning.

Federate Learning

- ML technique.

- Trains an algorithm across multiple decentralized edge devices or servers holding local data samples, without exchanging them.

- For example: mobile phones collectively study a shared prediction model, while keeping the device’s training data local instead of uploading and storing it.



- Federated learning is also used in self-driving cars.
    - The traditional approach can create safety risks.
    - FL can represent a solution to limit the volume of data transfer and accelerate learning processes.

Federated Transfer Reinforcement Learning

- Consists in implementing a model that mixes the three previous concepts
- Similar but different RL problems are used to train models
- Knowledge is transferred from one problem to another
- The training data is stored locally on each device, but the gained knowledge is loaded on a server

FTRL Framework in Autonomous driving

- FTRL framework is not designed for any specific RL method,Deep Deterministic Policy Gradient (DDPG) is chosen to be the RL implementation.

- RL agents interact with an stochastic environment in discrete time. At each time step, each agent makes observations, takes actions and receives rewards, as in standard RL models.

- All agents share the same model structure. They communicate with the FL server through a wireless router,

-A basic training process

1. Online transfer process. Since distributed RL agents are acting in various environments, a knowledge transfer process is needed when each RL agent interacts with its specific environment;

2. Single RL agent training and inference. This process serves as a standard RL agent training and inference process.

3. FedAvg process. All the useful knowledge of distributed RL agents is aggregated by FedAvg process of the RL models.

-An experiment

- In the next slide, you can see the results of different application settings, including DDPG results on single RC cars, FTRL-DDPG results with the federation of three RC cars (FTRL-DDPG) and FTRL-DDPG results with the federation of three RC cars and Airsim platform (FTRL-DDPG-SIM).

	car1		car2		car3	
	avg_dist	coll_no	avg_dist	coll_no	avg_dist	coll_no
DDPG	0.39	18	0.29	31	0.38	24
FTRL-DDPG	0.42 (7.7%)	**9** (50%)	0.37 (27.6%)	27 (12.9%)	**0.51** (34.2%)	17 (29.2%)
FTRL-DDPG-SIM	**0.45** (15.4%)	12 (33.3%)	**0.39** (34.5%)	**16** (48.4%)	0.50 (31.6%)	**13** (45.8%)

In the table above are represented the average LIDAR distances and collision number results of three cars on the test experiments. For each approach on each car, 50 cycles in the race are executed.

Conclusions

- Reinforcement learning, Transfer learning and Federated learning have its own set of benefits.

- The integration of the three of them gets better results.

- More investigation is necessary to further develop the field of this promising integration.

- About the different application settings, FTRL-DDPG-SIM performs better than both single execution of single RL agents and federation model with identical RL agents with better training speed and performance.