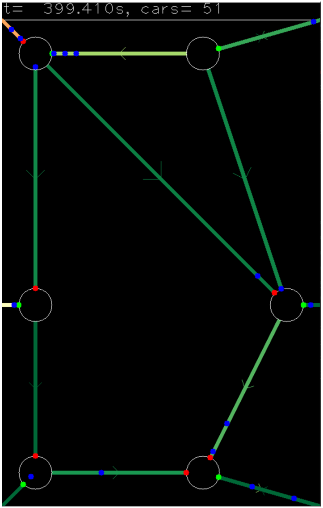
gym-traffic

Controlling traffic lights with Reinforcement Learning

Dominik Woiwode

24.03.2021

The Environment



- Street network as **directed graph**
  - The world consists of multiple **intersections**.
  - Intersections are connected with **streets**.
  - Intersections have **traffic lights** for each incoming street.
  - Traffic lights can either be **red or green**.
  - For each intersection there **cannot be more than one green light**.
  - **Vehicles** can spawn at some intersections with a predefined route.
  - Vehicles drive on streets and **stop at red traffic lights**.
  - Vehicles **do not crash** into each other.

⇝ **An Agent has to control the traffic lights**.

  - "Traffic flow" should be maximized

Observation- and Actionspace

Observationspace

- For each intersection t:
  - For each incoming street s:

$$\begin{cases} -1 & \text{if } |v(s)| = 0 \\ \sum_{v \in v(s)} \frac{v.pos}{|v(s)|} & \text{else} \end{cases}$$

$\rightsquigarrow$ observation $\in \mathcal{R}^{n \times 1}$ with one entry for each (relevant) street

Observation- and Actionspace

Observationspace

- For each intersection t:
  - For each incoming street s:

$$\begin{cases} -1 & \text{if } |v(s)| = 0 \\ \sum_{v \in v(s)} \frac{v.pos}{|v(s)|} & \text{else} \end{cases}$$

$\rightsquigarrow$ observation $\in \mathcal{R}^{n \times 1}$ with one entry for each (relevant) street

Actionspace

- For each intersection t:
  - Index of green street

$\rightsquigarrow$ Multidiscrete Actionspace

Observation- and Actionspace

Observationspace

- For each intersection t:
  - For each incoming street s:

$$\begin{cases} -1 & \text{if } |v(s)| = 0 \\ \sum_{v \in v(s)} \frac{v.pos}{|v(s)|} & \text{else} \end{cases}$$

$\rightsquigarrow$ observation $\in \mathcal{R}^{n \times 1}$ with one entry for each (relevant) street

Actionspace

- For each intersection t:
  - Index of green street

$\rightsquigarrow$ Multidiscrete Actionspace

Reward

r_1 Mean Velocity $\forall$ vehicles

r_2 Mean Acceleration $\forall$ vehicles

$$r' = \frac{1}{|n_v|} \sum_{v \in world} v.velocity$$

$$r_1 = \frac{r' - 5}{5}$$

$$r_2 = \frac{r_{1,2} - r_{1,1}}{\Delta t}$$

Observation- and Actionspace

Observationspace

- For each intersection t:
  - For each incoming street s:

$$\begin{cases} -1 & \text{if } |v(s)| = 0 \\ \sum_{v \in v(s)} \frac{v.pos}{|v(s)|} & \text{else} \end{cases}$$

⇝ observation $\in \mathcal{R}^{n \times 1}$ with one entry for each (relevant) street

But:

⇝ Requires knowledge about the street network

⇝ Agent only works for a specific street network

Actionspace

- For each intersection t:
  - Index of green street

⇝ Multidiscrete Actionspace

Reward

r_1 Mean Velocity $\forall$ vehicles

r_2 Mean Acceleration $\forall$ vehicles

$$r' = \frac{1}{|n_v|} \sum_{v \in world} v.velocity$$

$$r_1 = \frac{r' - 5}{5}$$

$$r_2 = \frac{r_{1,2} - r_{1,1}}{\Delta t}$$

More generalized approach

Idea:

- Add empty fake-streets so that each intersection has k incoming streets. (in practice: append -1's)
- In each time step:
  - Only provide observation for one intersection and ask to control only this traffic light.
- Reward: Stays the same as before.

$\rightsquigarrow$ Generalized observation and action space

Pro:

- Can transfer knowledge to other street networks
- Training is more efficient
  - No need to learn correlation of streets far away from intersection
  - Effect of good/bad action is more predictable

Drawback:

- Intersections cannot "communicate" between each other

Results

- stable-baselines3: PPO with MlpPolicy (network architecture: (64,64))
  - Supports Multi-Discrete Actionspace and Multi-Processing
- Horizon: 1000, World: `3x3circle`
- At least 5 episodes to evaluate an algorithm.

Conventional approach

Algorithm	$\overline{velocity}$	$\sum reward$	**Training duration**
random	4.200	2.626	-
PPO-acceleration	6.120	4.672	1.5M steps $(\sim 10h)$

Results

- stable-baselines3: PPO with MlpPolicy (network architecture: (64,64))
  - Supports Multi-Discrete Actionspace and Multi-Processing
- Horizon: 1000, World: `3x3circle`
- At least 5 episodes to evaluate an algorithm.

Conventional approach

Algorithm	$\overline{velocity}$	$\sum reward$	**Training duration**
random	4.200	2.626	-
PPO-acceleration	6.120	4.672	1.5M steps ($\sim 10h$)

Generalized approach

Algorithm	$\overline{velocity}$	$\sum reward$	**Training duration**
random	5.525	5.645	-
argmax	**8.115**	8.023	-
PPO-velocity	6.383	5.815	1.5M steps ($\sim 9h$)
PPO-velocity-shuffled	7.736	8.548	1.25M steps ($\sim 7.5h$)
PPO-acceleration-shuffled-1	**7.991**	7.522	1.25M steps ($\sim 7.5h$)
PPO-acceleration-shuffled-2	7.987	5.775	1.25M steps ($\sim 7.5h$)

Discussion/Future Extensions

- Does Markov assumption holds true?
  - Yes, especially in generalized approach.

Possible Extensions:

- Communication between intersection
  - For each incoming street: add observation of preceding intersection
  - Use RNN to generate a fixed sized representation of all intersection states
- Automatic world generation for different training models
- More complex traffic-light rules
  - In reality 2 lanes are allowed to drive in parallel if they do not cross each other
  - In reality left turner wait until opposing lanes drive
- Multiple lanes per street