Homework 1:

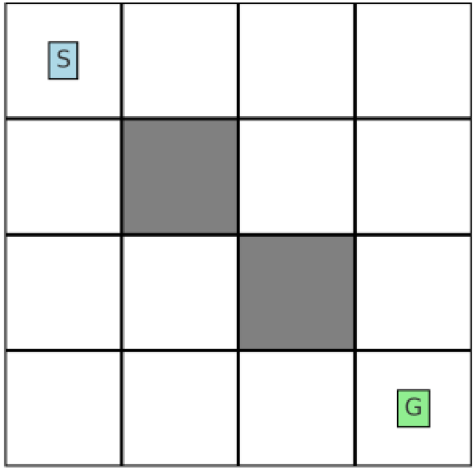
Introduction to RL

Spring 2025

Creating Custom Grid-world Environment

State Space

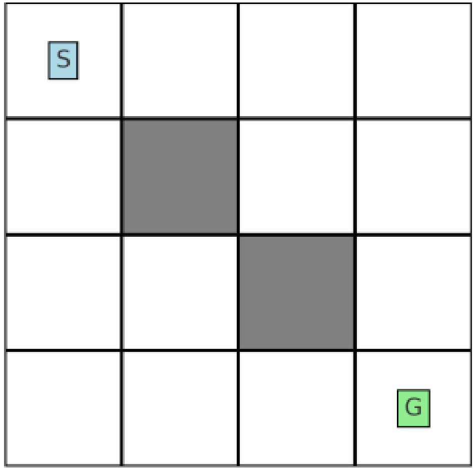
- 4*4 = 16 cells in the grid
- s_t = (row, column)
- Dicrete(16) for better implementation of Q-learning



Creating Custom Grid-world Environment

Action Space

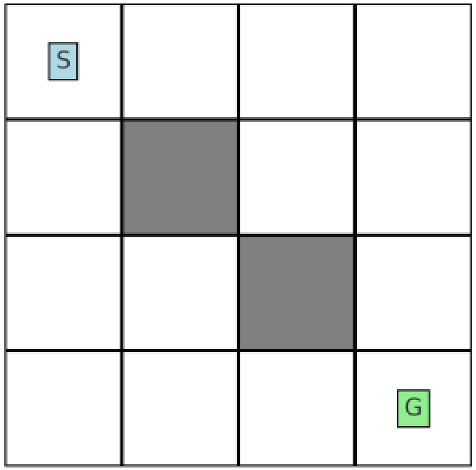
- The agent can move in four directions
- 0: up
- 1: down
- 2: left
- 3: right



Creating Custom Grid-world Environment

Implementation

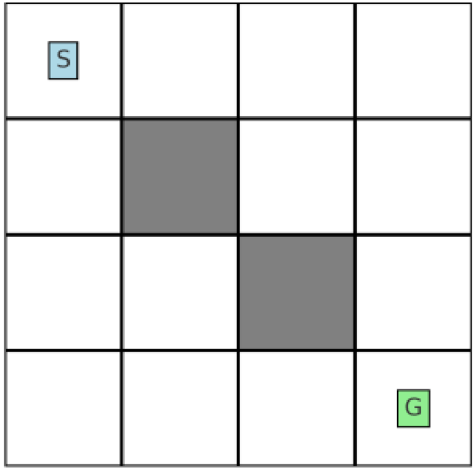
```
class GridWorldEnv(gym.Env):
    def __init__(self):
        # Grid world setup
        self.grid_size = 4
        self.start = (0, 0)  # Starting position 'S'
        self.goal = (3, 3)   # Goal position 'G'
        self.holes = [(1, 1), (2, 2)]  # Hole positions
        self.state = self.start

        # Define action and observation spaces
        self.action_space = spaces.Discrete(4)  # 0: up, 1: down, 2: left, 3: right
        self.observation_space = spaces.Discrete(16)  # from 0 to 15 for the problem states

    def get_observation(self):
        """convert internal state (i, j) to integer observation using state = 4*i + j."""
        i, j = self.state
        return 4 * i + j
```



Creating Custom Grid-world Environment

Reward Space

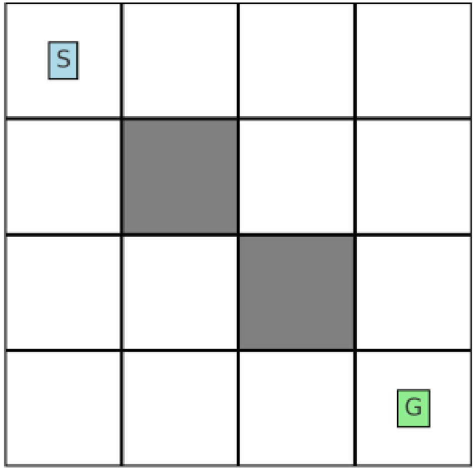
- Reaching the goal (G) at (3,3): +10 reward.
- Falling into a hole at (1,1) or (2,2): -1 reward.
- Moving anywhere else: 0 reward.



Creating Custom Grid-world Environment

Transition Probability

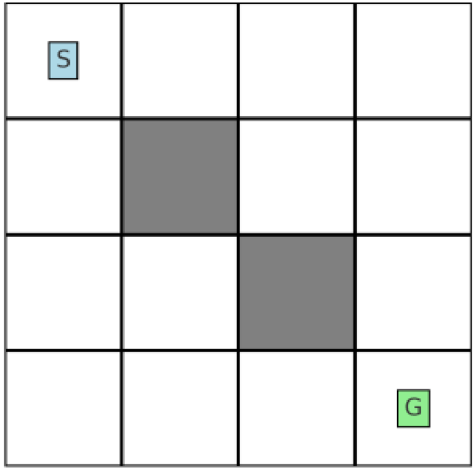
We assume this is deterministic environment.



Creating Custom Grid-world Environment

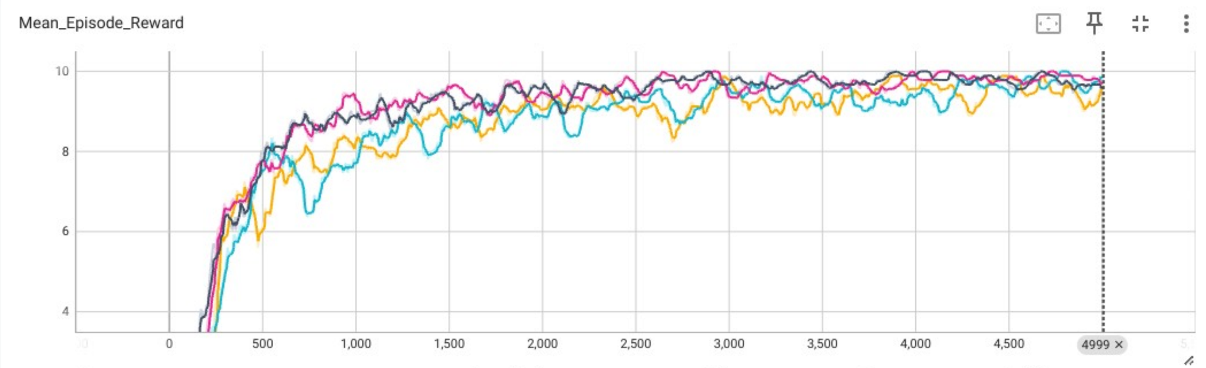
Rendering

```python
def render(self):
    """Render the grid world with the agent's position."""
    # Initialize a 4x4 grid with empty cells
    grid = [['.' for _ in range(4)] for _ in range(4)]

    grid[0][0] = 'S'                # Start
    grid[3][3] = 'G'                # Goal
    for hole in self.holes:         # Holes
        grid[hole[0]][hole[1]] = 'H'

    agent_row, agent_col = self.state
    if grid[agent_row][agent_col] == 'S':
        grid[agent_row][agent_col] = 'A'  # Agent on start
    elif grid[agent_row][agent_col] == 'G':
        grid[agent_row][agent_col] = 'A'  # Agent on goal
    elif grid[agent_row][agent_col] == 'H':
        grid[agent_row][agent_col] = 'A(H)'  # Agent on hole
    else:
        grid[agent_row][agent_col] = 'A'     # Agent on empty cell

    # Print the grid
    for row in grid:
        print('    '.join(row))
    print()  # empty line
```



Creating Custom Grid-world Environment

Solution using Qlearning



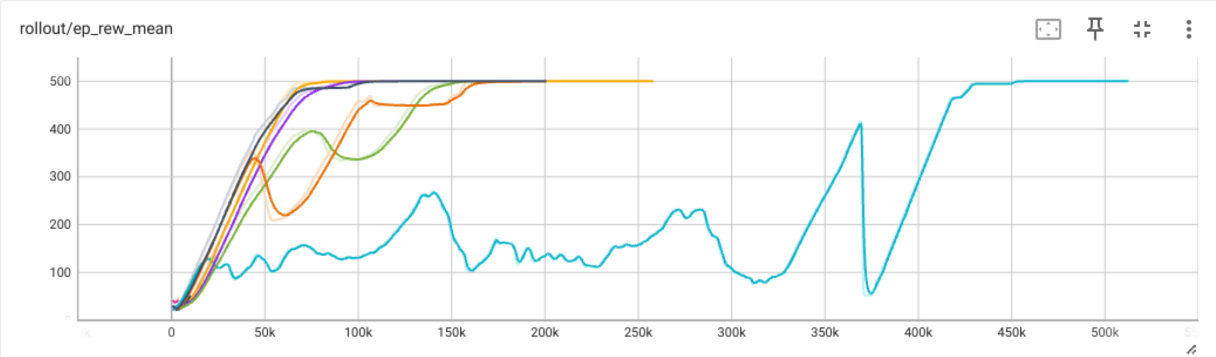
	Run ↑	Smoothed	Value	Step	Relative
●	experiment_lr_0.1_0.95	9.8287	9.78	4,999	353.7 ms
●	experiment_lr_0.1_0.99	9.6863	9.67	4,999	363.4 ms
●	experiment_lr_0.9_0.95	9.5752	9.67	4,999	337.5 ms
●	experiment_lr_0.9_0.99	9.5742	9.56	4,999	364.5 ms

Supervised learning vs RL

- When we want to make decisions
- Environment changes
- We can’t label all of the input and outputs

Solving Predefined Environments

CartPole



Run ↑	Smoothed	Value	Step	Relative
PPO_CartPole_tensorboard/PPO_1	500	500	258,048	2.44 min
PPO_CartPole_tensorboard/ppo_lr0.0001_gamma0.95/PPO_1	500	500	200,704	1.843 min
PPO_CartPole_tensorboard/ppo_lr0.0001_gamma0.99/PPO_1	500	500	200,704	2.009 min
PPO_CartPole_tensorboard/ppo_lr0.01_gamma0.95/PPO_1	499.9745	500	200,704	2.049 min
PPO_CartPole_tensorboard/ppo_lr0.01_gamma0.99/PPO_1	500	500	200,704	2.064 min
a2c_CartPole_tensorboard/A2C_1	51.902	58.81	10,000	5.875 sec
a2c_CartPole_tensorboard/A2C_2	500	500	512,500	8.476 min
a2c_CartPole_tensorboard/A2C_3	42.3393	46.1333	3,500	2.896 sec

Solving Predefined Environments

CartPole with reward *2 wrapper

Solving Predefined Environments

CartPole

Comparison of Reinforcement Learning Algorithms applied to the Cart-Pole Problem

Savinay Nagendra
Dept. Elect. Electro. Engg.,
PES Inst. of Tech.,
Bangalore, India.
nagsavi17@gmail.com

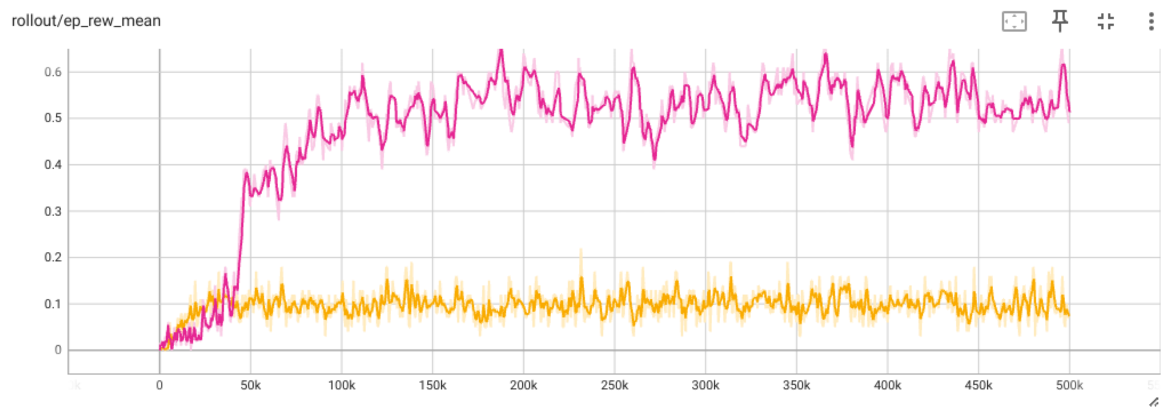
Nikhil Podila
Dept. Elect. Electro. Engg.,
PES Inst. of Tech.,
Bangalore, India.
nikhilpodila.94@gmail.com

Rashmi Ugarakhod
PES Cen. Int. Systems;
Dept. Electro. Commun. Engg.,
PES University,
Bangalore, India.
rashmi.ugarakhod@gmail.com

Koshy George
PES Cen. Int. Systems;
Dept. Electro. Commun. Eng.,
PES University,
Bangalore, India.
kgeorge@pes.edu

Solving Predefined Environments

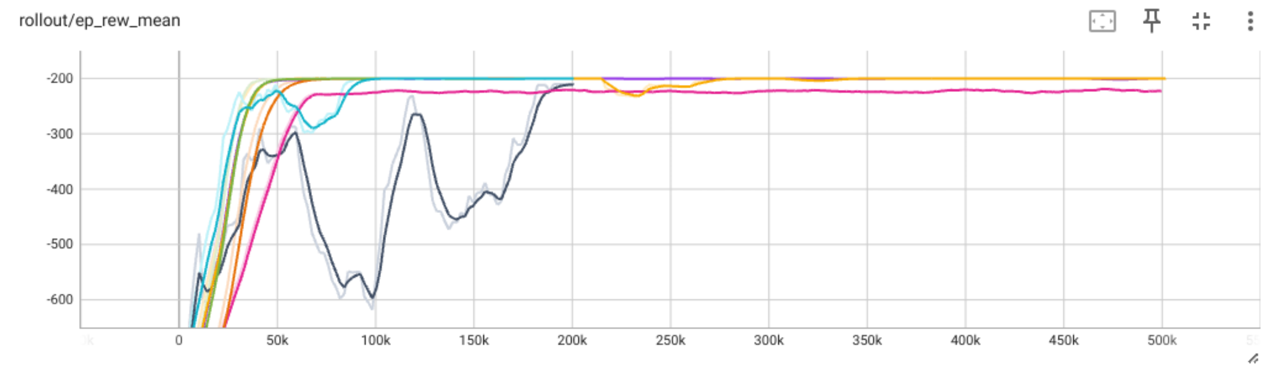
FrozenLake



Run	Smoothed	Value	Step	Relative
A2C_FrozenLake_tensorboard/A2C_1	0.0741	0.07	500,000	10.21 min
DQN_FrozenLake_tensorboard/DQN_1	0.5131	0.49	499,979	5.567 min

Solving Predefined Environments

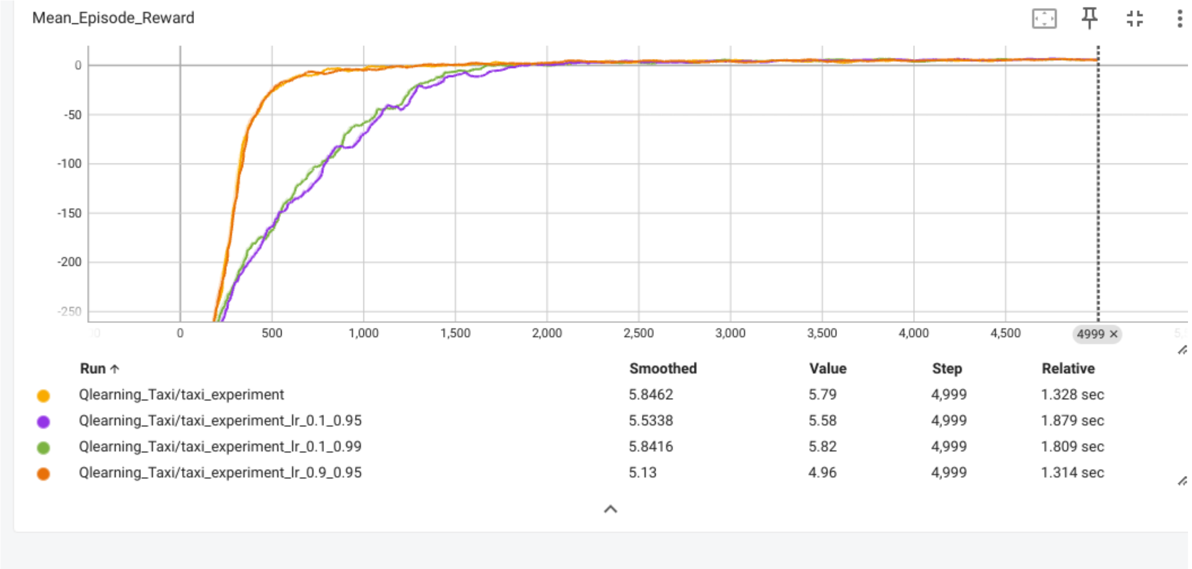
Taxi



Run ↑	Smoothed	Value	Step	Relative
DQN_Taxi_tensorboard/DQN_1	-222.6926	-222.61	499,661	6.797 min
PPO_Taxi_tensorboard/PPO_1	-200.1016	-200.09	501,760	16.23 min
PPO_Taxi_tensorboard/PPO_2	-200.2428	-200.36	501,760	11.18 min
PPO_Taxi_tensorboard/ppo_lr0.0001_gamma0.95/PPO_1	-200.1926	-200.18	200,704	4.314 min
PPO_Taxi_tensorboard/ppo_lr0.0001_gamma0.99/PPO_1	-200.0537	-200	200,704	4.066 min
PPO_Taxi_tensorboard/ppo_lr0.01_gamma0.95/PPO_1	-211.0146	-210.35	200,704	3.517 min
PPO_Taxi_tensorboard/ppo_lr0.01_gamma0.99/PPO_1	-200.0011	-200	200,704	3.931 min

Solving Predefined Environments

Taxi with Q learning



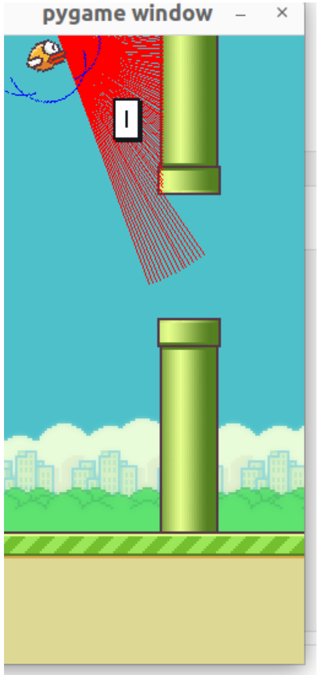
Solving Predefined Environments

Flappy Bird

- We use flappy-bird-gymnasium
- Actions :
  - 0 - do nothing
  - 1 – flap

Rewards:

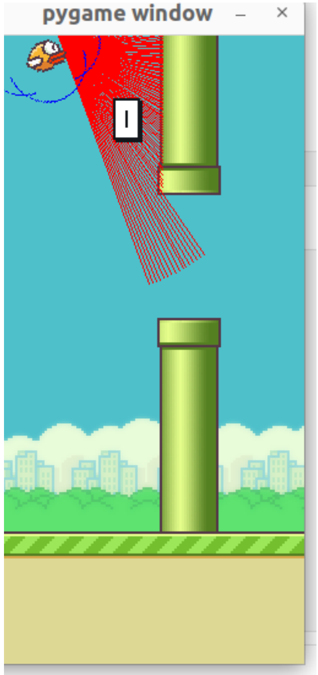
- +0.1 - every frame it stays alive
- +1.0 - successfully passing a pipe
- -1.0 - dying
- −0.5 - touch the top of the screen



Solving Predefined Environments

Flappy Bird

- State space: The LIDAR sensor



Playing Flappy Bird Based on Motion Recognition Using a Transformer Model and LIDAR Sensor

by Iveta Dirgová Luptáková, Martin Kubovčík *, and Jiří Pospíchal *

Institute of Computer Technologies and Informatics, Faculty of Natural Sciences, University of Ss. Cyril and Methodius, J. Herdu 2, 917 01 Trnava, Slovakia

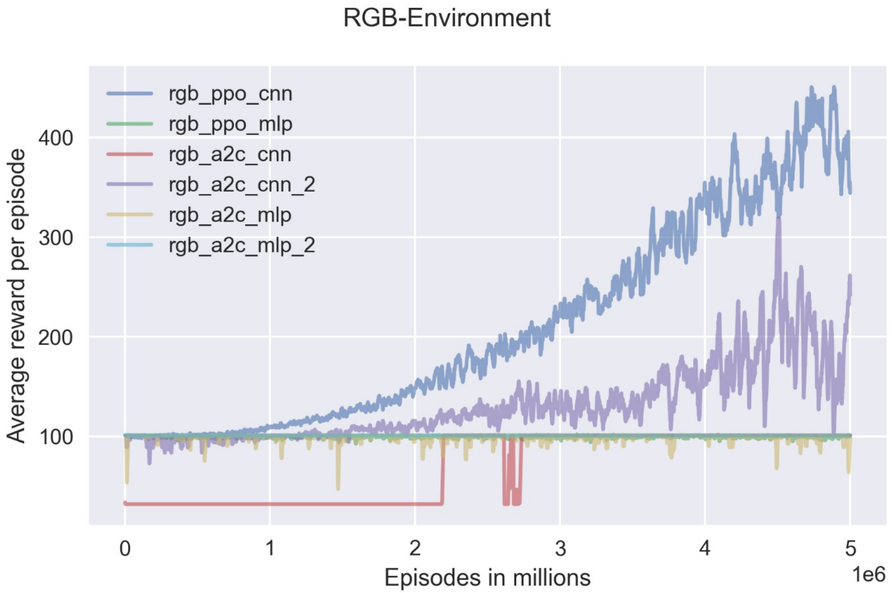
* Authors to whom correspondence should be addressed.

Sensors **2024**, *24*(6), 1905; https://doi.org/10.3390/s24061905

Solving Predefined Environments

Flappy Bird

- See the link for details



Source: https://github.com/LukasDrews97/flappy-bird-reinforcement-learning/

Solving Predefined Environments

Flappy Bird

- Hyperparameters

Environment	Config	Lr	Gamma	Best Result	Training Time
Simple	PPO MLP	1e-5	0.95	101	4h
RGB	PPO MLP	1e-5	0.95	102	17h
RGB	PPO CNN	1e-5	0.95	450	15h
Simple	A2C MLP	7e-4	0.99	1800	1h
RGB	A2C MLP	7e-4	0.99	101	30h
RGB	A2C MLP	7e-5	0.95	101	30h
RGB	A2C CNN	7e-4	0.99	101	30h
RGB	A2C CNN	7e-5	0.95	318	30h

References

1. https://github.com/markub3327/flappy-bird-gymnasium
2. https://github.com/LukasDrews97/flappy-bird-reinforcement-learning/
3. Nagendra, S., Podila, N., Ugarakhod, R., & George, K. (2017). Comparison of reinforcement learning algorithms applied to the cart-pole problem. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).