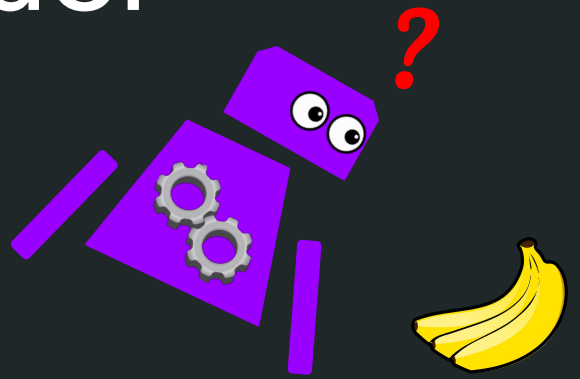
Tuning the Lunar Lander



Al Saqib Majumder, Atharv Suryawanshi, Parsa'eian Mohamad Rasoul, Pavitra Batra, Reza Samavat, Sepehr Farzaneh Raziabad

Samin Nili-Ahmadabadi [TA], Soan Kim [Project TA]

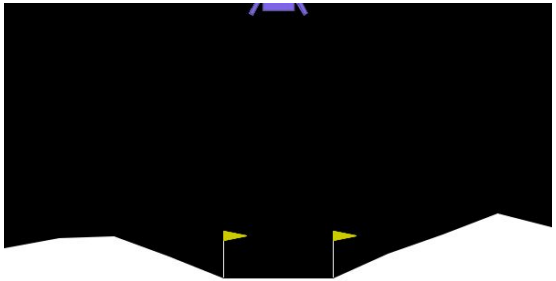
neuromatch academy

POD: Jinfengopteryx Mambo

The Lunar Lander 🚀🚀

Description

- Land agent on a baseline (can be rugged)
- We use the gymnasium library to implement



Features

- Discrete Action Space (4)
- Discrete Observation State (8)
- Infinite Fuel

Improving performance

Learning

- DQN

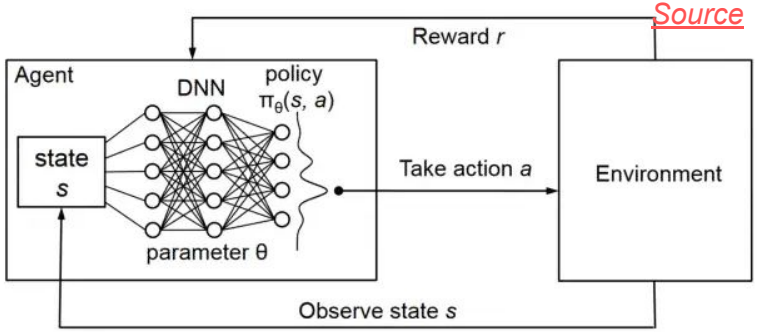
Improve the performance in terms of

- Stability of reward
- Speed of reaching the goal(landing)

Improve How?

- Reward Shaping
- Hyperparameter Tuning

Source



Custom Reward Shaping Function

Default Reward Shaping

```
shaping = (
-100 * np.sqrt(state[0] * state[0] + state[1] * state[1])
- 100 * np.sqrt(state[2] * state[2] + state[3] * state[3])
- 100 * abs(state[4])
+ 10 * state[6]
+ 10 * state[7]
)
```

Custom Reward Shaping

```
shaping = (-100 * np.sqrt(state[0] * state[0] + state[1] * state[1])
- 100 * np.sqrt(state[2] * state[2] + state[3] * state[3])
- 100 * abs(state[4])
+ 10 * state[6]
+ 10 * state[7]
+10 if (np.abs(state[4])<1.5) else 0
-100*np.abs(state[0])
+10 if (np.abs(state[1])<1) else 0
+10 if (np.abs(state[5])<3) else 0)
```

state[0] = pos_x
state[1] = pos_y
state[2] = vel_x
state[3] = vel_y
state[4] = angle
state[5] = angular_vel
state[6] = a
state[7] = b

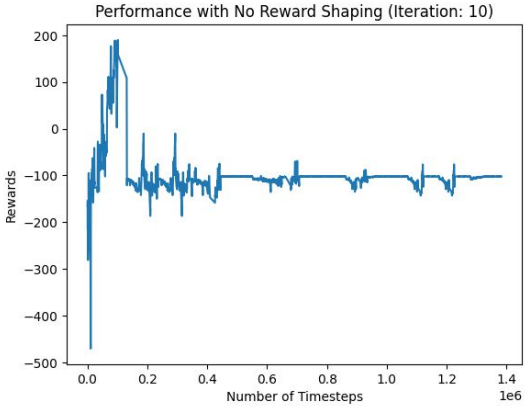
Background of Reward Shaping:

- Reward shaping is a method for engineering a reward function in order to provide more frequent feedback on appropriate behaviors. https://doi.org/10.1007/978-0-387-30164-8_731

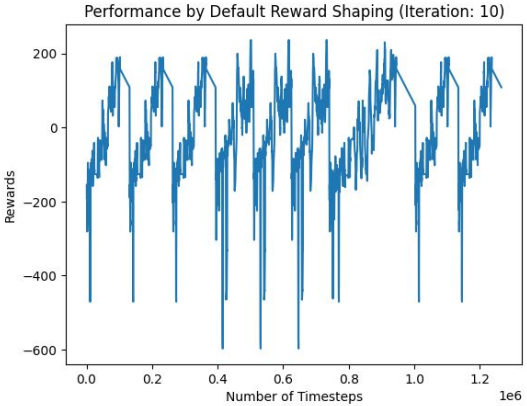
- As you can see in the next page's chart the default reward shaping is not sufficient because the lowest and highest rewards has an extreme vibration.This means agent is learning sometimes good and sometimes bad rather than frequently good. That's because it uses just the position of x and y but instead we used limitation on the unwanted actions(x_pos) and some positive reward on limitation of unwanted action.Then as you can see we have better results and less up and downs in its chart. https://github.com/iamshan794/Custom_Reward_Lunar_Lander-V2.git

Rewards Across Shaping Conditions (10 Iterations)

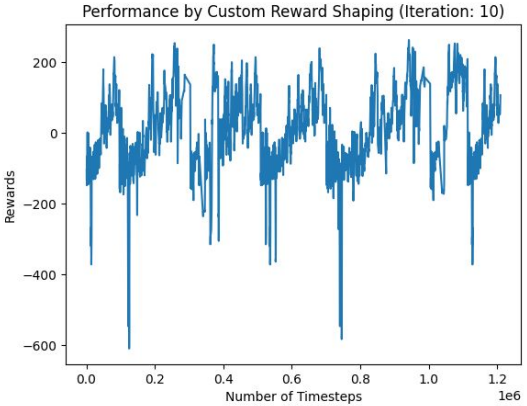
No Shaping



Default Shaping

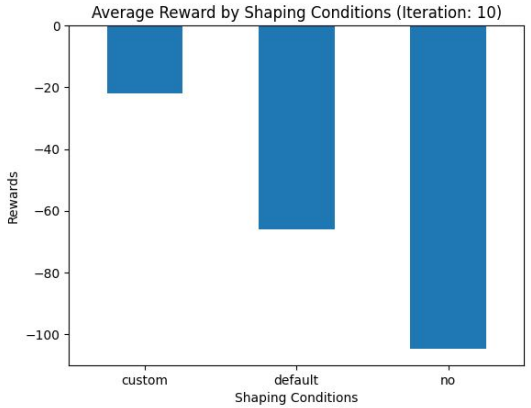


Custom Shaping

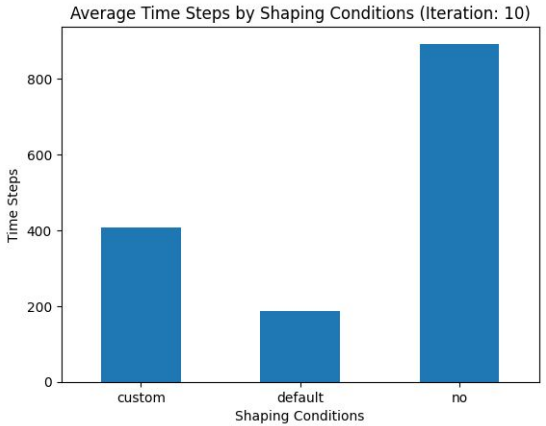


Comparisons Across Shaping Conditions (10 Iterations)

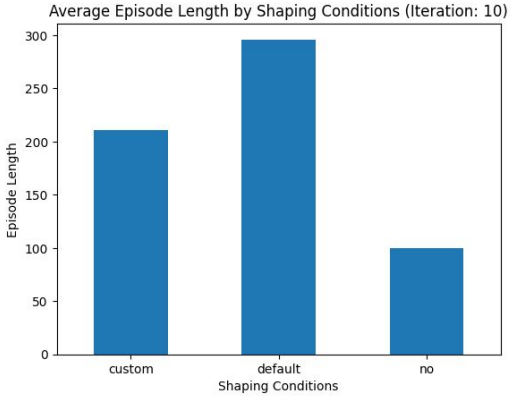
Avg. Reward



Avg. Time Steps



Avg. Episode Length



Conditions	Reward	Time Step	Episode Length
No Shaping	-104.76	892.4	99.67
Default Shaping	-65.88	186.7	296
Custom Shaping	-21.92	408.66	210.61

Avg. Reward per Episode by Shaping			
Timesteps	No shaping	Default shaping	Custom shaping
20,000	-132.43	-465.16	-546.19
40,000	-123.14	-39.11	-231.79
80,000	-120.94	125.81	-9.43

Hyperparameter tuning

- **Learning rate:** Determines how quickly our model adapts to the data during training
- **Architecture of hidden layers:** Size and depth of a network influences its ability to learn complex patterns from the data
- **Exploration rate:** A measure of times an agent chooses a sub-optimal reward in order to learn new actions and states

Hyperparameter Tuning Results

By increasing number of layers the performance gets inefficient and even by changing “lr” the results are not reliable due to simplicity of our problem.

The effect of lr on the performance of agent in terms of speed and max rewards is inevitable.

Exploration rate could enhance the speed but it couldn’t deal with stability.

Num of Layers	Learning Rate	Avg. Total Rewards	Steps to max reward	Total steps
2	0.001	83 (131)	70	120,000
2	0.004	222 (72)	110	120,000
3	0.0008	125 (187)	110	120,000

Discussion - What have we learnt?

Reward shaping

I. Important to guide the agent's actions. The *sparse reward condition* is an open problem in RL. To overcome this, reward shaping is introduced to guide the direction of the agent.

II. Could not find a better reward function than the default. Reward function tends to be specific and sensitive to each environment. Need expert domain knowledge. Reward function design is difficult to generalize across environments.

Hyperparameter Tuning

I. Too small learning rate harms learning (plateau problem)

II. Depth and size of the hidden layers can improve the performance. However, a very deep hidden layer is not necessary in our simple environment (avoiding overfitting).

Thank You!

We want to thank our mentor Muhammad Moustafa for providing us with valuable insight and wisdom in interpreting our model and code.

We also want to thank Soan Kim (Project TA) & Samin Nili-Ahmadabadi (Pod TA) for helping us with our project and guiding us through unchartered territories in deep reinforcement learning.

Finally, we want to thank Neuromatch Academy for arranging the summer school and providing us with this opportunity to engage in this topic.

References

Dawood, M., Dengler, N., de Heuvel, J., & Bennewitz, M. (2023, May). Handling Sparse Rewards in Reinforcement Learning Using Model Predictive Control. In 2023 IEEE International Conference on Robotics and Automation (ICRA) (pp. 879-885). IEEE.

Ainsworth, M., & Shin, Y. (2021). Plateau phenomenon in gradient descent training of ReLU networks: Explanation, quantification, and avoidance. SIAM Journal on Scientific Computing, 43(5), A3438-A3468.