Ball Sort Puzzle

Reinforcement Learning

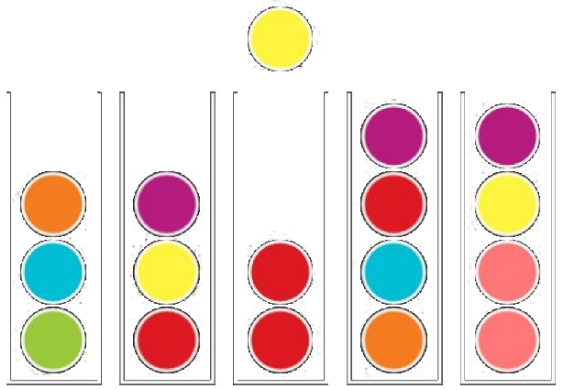
Artificial Intelligence

Master in Informatics and Computing Engineering

Gonçalo Alves – up201806451
Gustavo Mendes – up201806078
Pedro Seixas – up201806227

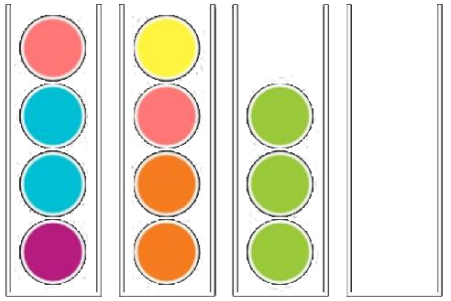
Formulation of the problem as an ML problem

The objective of the Ball Sort Puzzle is to sort the colored balls in the tubes until all balls with the same color stay in the same tube.



The player can only move a ball at the top of a tube to:

- an empty tube;
- another tube that has a ball with the same color on top and has enough space;



The player can also undo his moves if he finds himself without moves.

Formulation of the problem as a RL problem

State Representation:

- Ball: [1..n] (n = the number of colors)
- Tube: [Ball, Ball, Ball, Ball] or []
- Game: [Tube,Tube,…]

Actions:

- Move(X,Y): move Ball from Tube X to Tube Y

Algorithms:

- Q-Learning
- SARSA
- Monte Carlo

Rewards:

Three ways to assign a reward based on verifications:

-Verify if the game is finished (+2) or the tube that received a ball is completed (+1)

-Verify if the game is invalid (-5) or finished (+20)

-Verify if the game is invalid (-1000), finished (+50) or count the number of consecutive balls with the same colour

Related Work

Mainly 2D games:

- Retro
- Slime Volley Gym
- Demon Attack

On OpenAI:

- Spaces
- Environments

On Policies:

- Q-Learning
- Jupyter Notebook about Q-Learning
- SARSA

Description of the tools

Jupyter Notebook – Allows for interactive computing

OpenAI gym - Allows for modeling of a reinforced learning environment

Algorithms - Since the action and the observation space of the puzzle are both discrete, the policies to be implemented are Q-Learning and SARSA

Implementation work already carried out

Programming language: Python, OpenAI, Jupyter Notebook (with Anaconda)

Development environment: VSCode/IntelliJ

Data Structures:

- Lists, for representing the Tubes and the Game

File Structure: The Project's Repository is available at Github

Implemented Work: Modeled Environment and Agent, Q-Learning Algorithm

Models

As previously stated, both the Action and the Observation Space are discrete. Therefore, we can apply Q-Learning and SARSA to our RL problem. These algorithms are Model-Free.

```python
#Function to learn the Q-value, using Q-Learning
def updateQ(state, state2, reward, action):
    predict = Q[env.get_index(state), action]
    target = reward + discount_factor * np.max(Q[env.get_index(state2), :])
    Q[env.get_index(state), action] = predict + alpha * (target - predict)
```

Fig 1. Q-Learning Implementation

```python
#Function to learn the Q-value, using SARSA
def updateSarsa(state, state2, reward, action, action2):
    predict = Q[env.get_index(state), action]
    target = reward + discount_factor * Q[env.get_index(state2), action2]
    Q[env.get_index(state), action] = predict + alpha * (target - predict)
```

Fig 2. SARSA Implementation

Analysis

Started with a simple configuration, a puzzle with 2 colours and 3 tubes, and used the first evaluation function described: +10, if finished; +1, if a tube was completed using an action; -1, for every other move.

The parameters used were:

- Learning rate, 0.5
- Discount factor, 0.6
- Epsilon, 0.2
- Train episodes, 500

Analysis

Fig 3. Q-Learning with
Training score over time: 9.434

Fig 4. SARSA with
Training score over time: 9.316

Analysis

After this successful try, a more complex puzzle was tested: 3 colours and 5 tubes. This test was made with the same evaluation function as before.

As we can see and as expected, SARSA is a more conservative algorithm, evident by the absence of “dips” in later episodes. This is because SARSA is on-policy.

Fig 5. Q-Learning with
Training score over time: 8.424

Fig 6. SARSA with
Training score over time: 8.45

Conclusions

Altough a very simplified version of a RL problem, this project allowed us to learn more about the topic of Machine Learning.

As for future work, we would’ve liked to create na abstraction of the puzzle, so that we could “feed” our agent with various configurations of the puzzle and it would be capable of solving them.