Feature-Based Reinforcement Learning for the Rubik's Cube

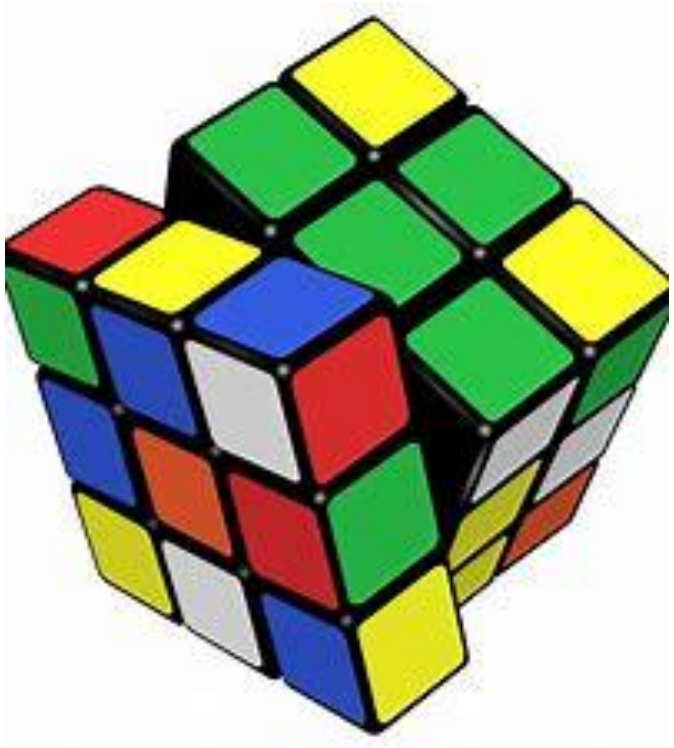
ADRIAN AMBROSE

DSCI-6670-01 AI COURSE

12/06/2020

Project Objectives

1. Build 2x2 Rubik's cube simulator
2. Train reinforcement learning agent to solve Rubik's cube using approximate q-learning
3. Demonstrate reinforcement learning agent solving Rubik cube



Source: Wikipedia

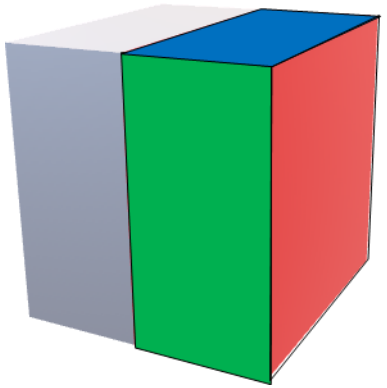
Objective 1. Build Rubik's cube simulator

VIDEO TO DEMONSTRATE CODE

Objective 2. Train reinforcement learning agent to solve Rubik's cube

Approximate Q-learning Features:

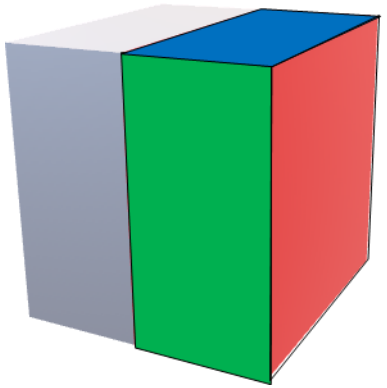
- Number of one, two, three, and four colored faces
- Number of Distinct colors on face pairs (face pair example: front and top)
- Number of full layers
  - Layer = face plus adjacent squares
  - Full = face is one color, each pair of adjacent squares are a single color



Objective 2. Train reinforcement learning agent to solve Rubik's cube

Approximate Q-learning Features:

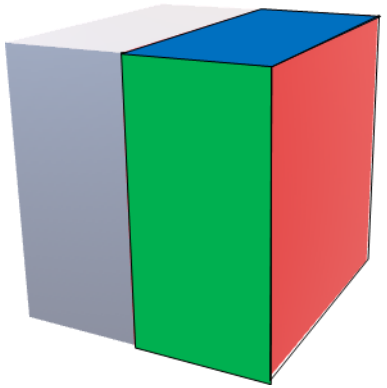
- **Number of one, two, three, and four colored faces**
- Number of Distinct colors on face pairs (face pair example: front and top)
- Number of full layers
  - Layer = face plus adjacent squares
  - Full = face is one color, each pair of adjacent squares are a single color



Objective 2. Train reinforcement learning agent to solve Rubik's cube

Approximate Q-learning Features:

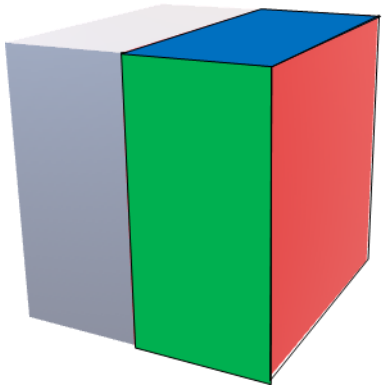
- Number of one, two, three, and four colored faces
- **Number of Distinct colors on face pairs (face pair example: front and top)**
- Number of full layers
  - Layer = face plus adjacent squares
  - Full = face is one color, each pair of adjacent squares are a single color



Objective 2. Train reinforcement learning agent to solve Rubik's cube

Approximate Q-learning Features:

- Number of one, two, three, and four colored faces
- Number of Distinct colors on face pairs (face pair example: front and top)
- **Number of full layers**
  - Layer = face plus adjacent squares
  - Full = face is one color, each pair of adjacent squares are a single color



Objective 2. Train reinforcement learning agent to solve Rubik's cube

Pseudo Code:

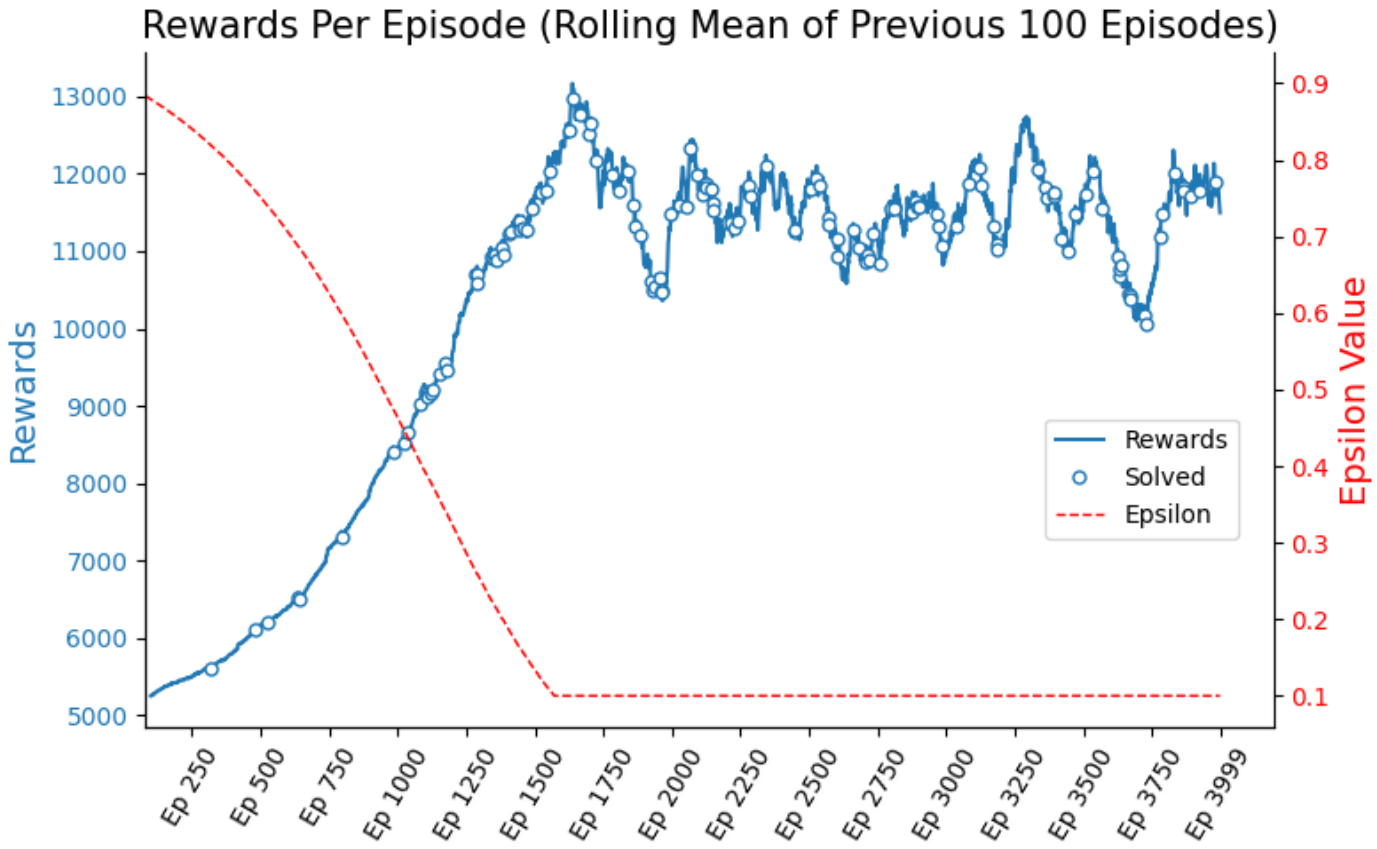
```
Def getReward(cubeState):
    If Goal State is True:
        Reward = 1000
    Else:
        Reward = (# Full Layers)*10 + (# 1  Color
        Faces)*6 + (# 2 Color Faces)*3 +    # 3
        Color Faces – Living Tax
```

Reward Structure Features By Importance:

1. Number of full layers
2. Number of one colored faces
3. Number of two colored faces
4. Number of three colored faces

Objective 2. Train reinforcement learning agent to solve Rubik's cube

- **Rewards**
  - The total reward accumulated for the episode. Shown as the rolling mean of the previous 100 episodes to smooth the curve
- **Solved**
  - Indicates the cube was solved in the episode
- **Epsilon**
  - Shown as a probability of the agent choosing a random action during training. The probability of a random action decays as training progresses, with a hardcoded stop at .1 to allow for continued exploration.



Note: Episode ends when cube is solved or at 1000 actions

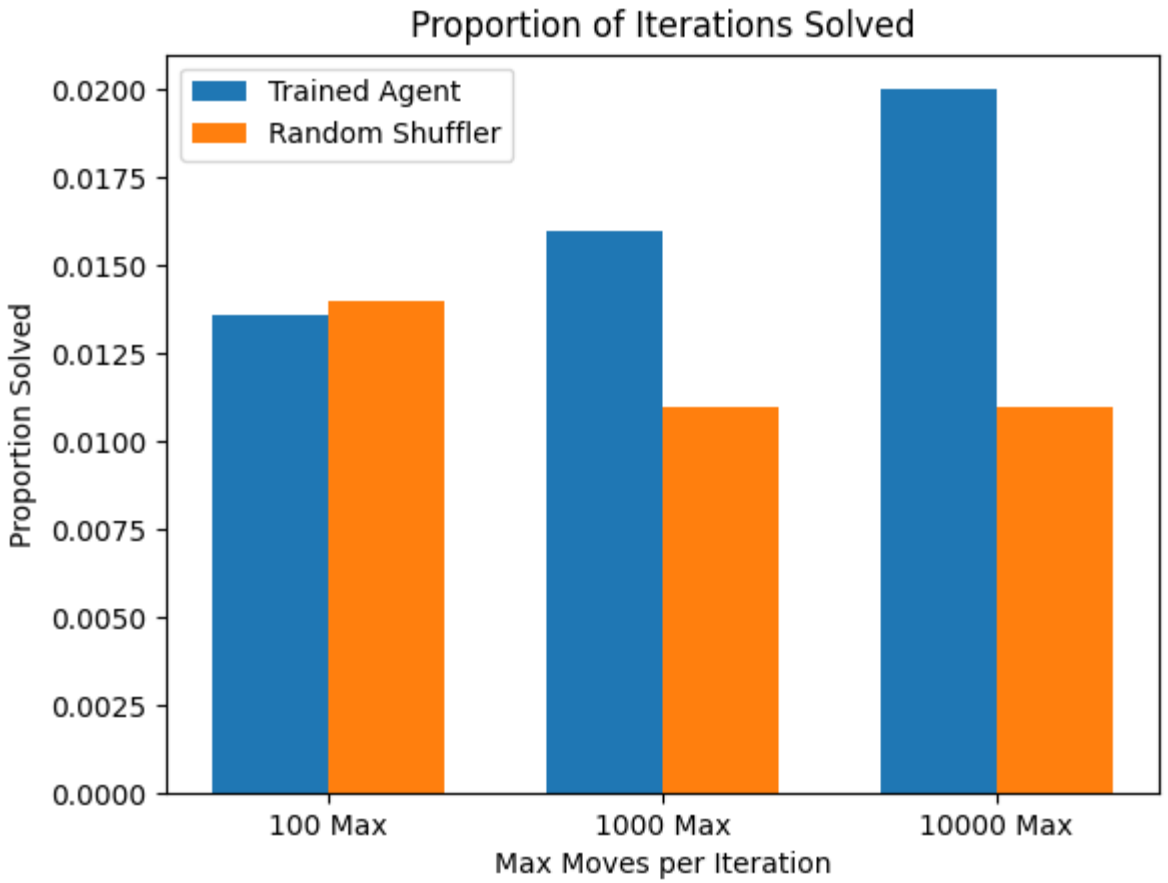
Objective 3. Demonstrate reinforcement learning agent successfully solving Rubik cube

Hypothesis

- H_0 = proportion of cubes solved by random shuffler = proportion solved by the trained agent
- H_a = Proportion of cubes solved by random shuffler < proportion solved by the trained agent
- α = **.05**

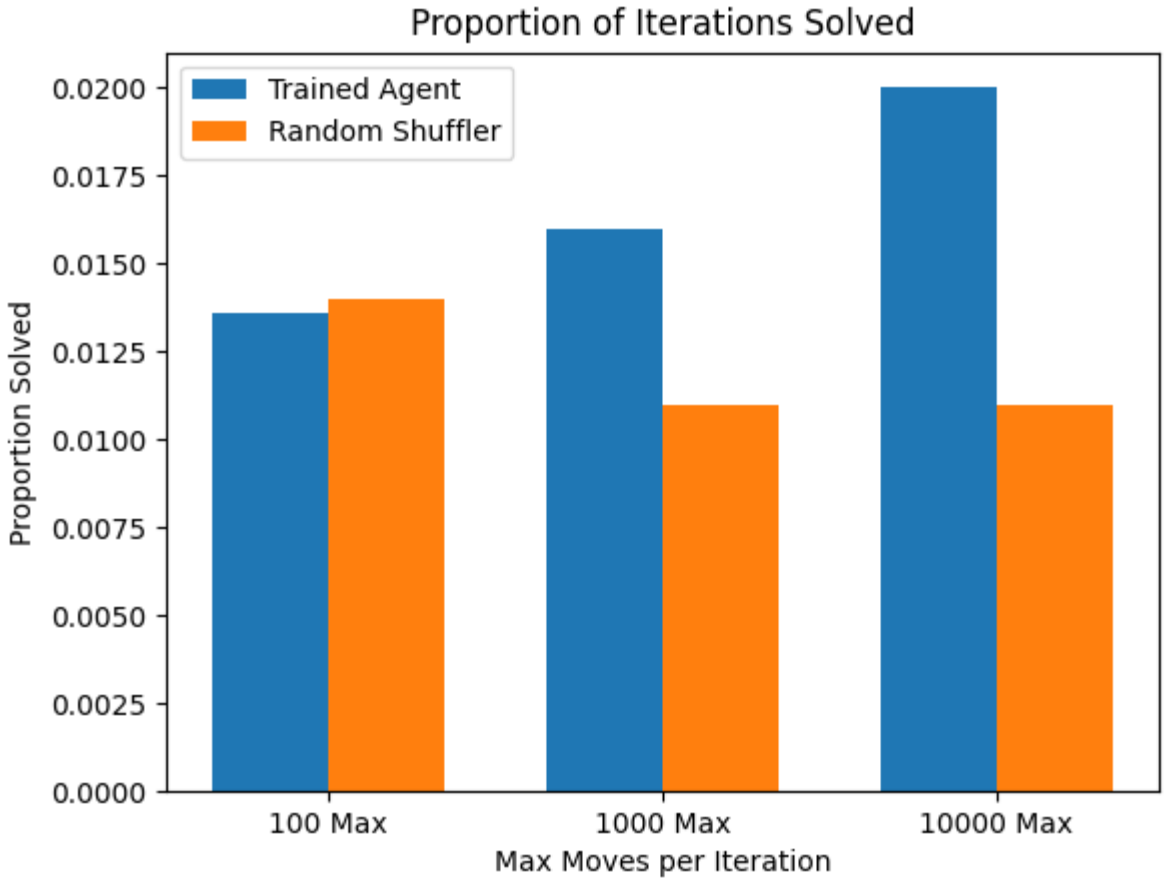
Results

- Chi Squared Test P-Value = **0.761**
- Z Test P-Value = **0.380**
- **Can not reject H_0**



Objective 3. Demonstrate reinforcement learning agent successfully solving Rubik cube

- **Takeaways:**
  - Agent did not perform as hoped
  - Agent better with more moves allowed
  - Number of max moves could be increased
  - Features and rewards need to be re-evaluated.
    - They do not appear to provide enough information to solve the Rubik's Cube.



References and Similar Works

- Resources
  - Artificial Intelligence: A Modern Approach (3rd Edition).
    By Stuart Russell and Peter Norvig.
    Prentice Hall, 2009. ISBN 0-13-604259-7.
  - Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). Cambridge, MA: Bradford Books.
  - Project 3: Reinforcement Learning from Fall 2020 University of New Haven DSCI-6670-01
- More complex implementation
  - McAleer, Stephen & Agostinelli, Forest & Shmakov, Alexander & Baldi, Pierre. (2018). Solving the Rubik's Cube Without Human Knowledge. https://arxiv.org/pdf/1805.07470.pdf
- Similar approaches:
  - https://github.com/arjunsatheesan/Rubik-s-Cube-Solver-using-Reinforcement-Learning/blob/master/Q_learn.py
  - https://github.com/AashrayAnand/rubiks-cube-reinforcement-learning/blob/master/Agent.py