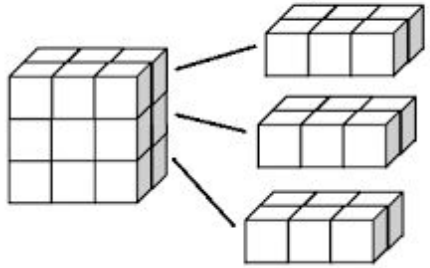
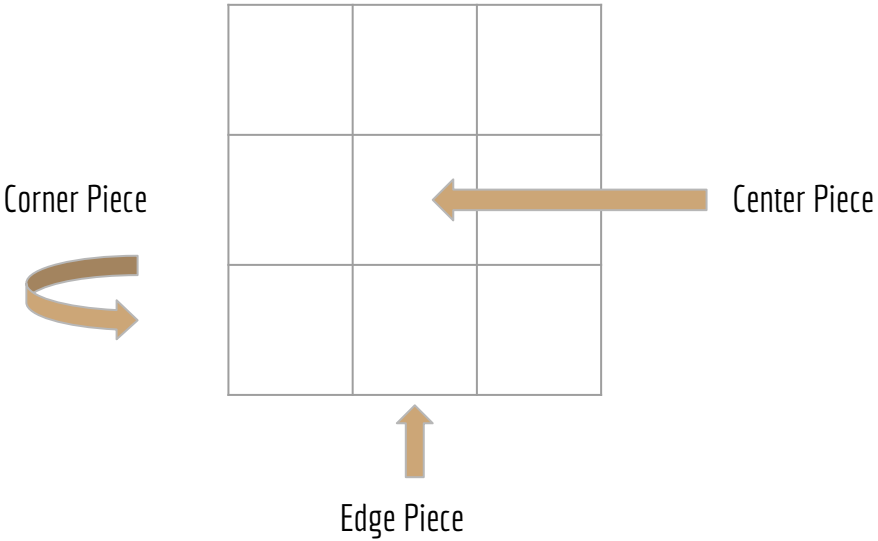
Solving a Rubik's Cube with Reinforced Learning

Sara Zhou

Cube Background

Each Rubik's Cube has 6 faces with 3 difference piece types:

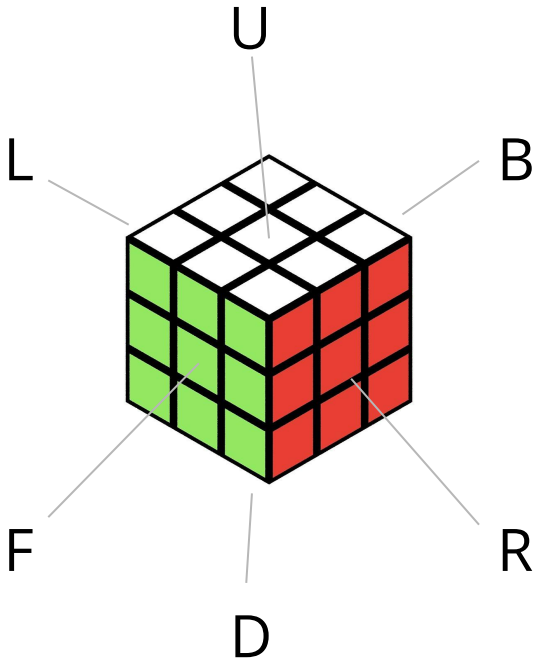




Problem Statement

Can I train a DNN to solve a rubik's cube?

Facial Notation will be represented as such and each movement will be recorded with ‘ to represent a counterclockwise movement.

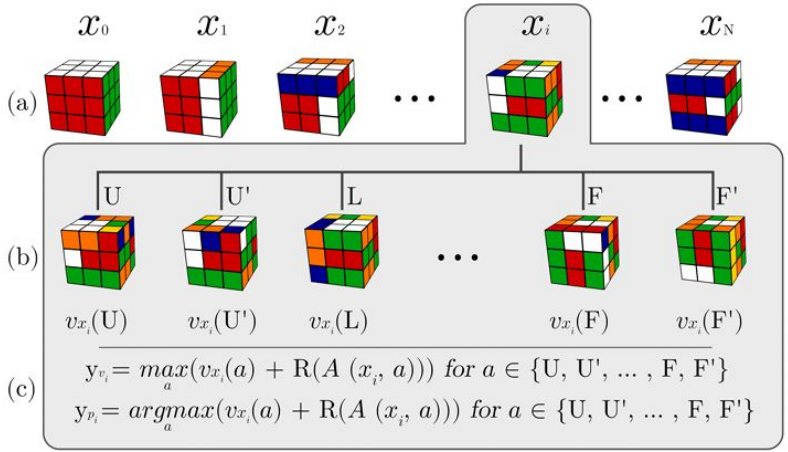


Example Notation:
F U R L R’ D’ U B

Approach

For data collection I decided to generate training samples starting from a solved state and taking random actions.

The method I chose to use is a method called Autodidactic Iteration (ADI) ,developed by UC Irvine Statistics and CS alumni, which is essentially an interactive surprised learning procedure that trains a neural network.



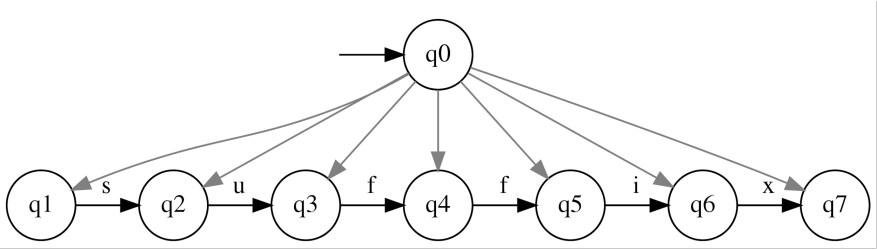
Algorithm 1: Autodidactic Iteration

Initialization: θ initialized using Glorot initialization
repeat
  $X \leftarrow$ N scrambled cubes
  for $x_i \in X$ **do**
    for $a \in \mathcal{A}$ **do**
      $(v_{x_i}(a), \boldsymbol{p}_{x_i}(a)) \leftarrow f_\theta(A(x_i, a))$
    $y_{v_i} \leftarrow \max_a(R(A(x_i, a)) + v_{x_i}(a))$
    $\boldsymbol{y}_{p_i} \leftarrow \operatorname{argmax}_a(R(A(x_i, a)) + v_{x_i}(a))$
    $Y_i \leftarrow (y_{v_i}, \boldsymbol{y}_{p_i})$
  $\theta' \leftarrow train(f_\theta, X, Y)$
  $\theta \leftarrow \theta'$
until $iterations = M$;

Solving the cube:

Along with our trained neural network we build a search tree iteratively by beginning with a tree consisting only of our starting state.

This simulation is performed until we reach the solved state or exceeds maximum computation time.



Results

While optimizing for both computation time and moves made, it would average around 30 moves.

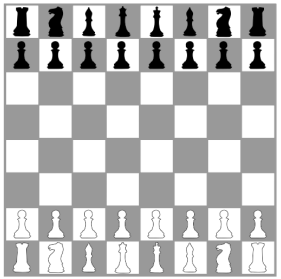
Compared to a human, between 50 - 60 moves, and the human world record sitting at 20 moves.

Computers say that the almost 100% of solves are within 16-19 moves.

Extensions

Some applications of DRL (Deep Reinforced Learning):

- Self Driving Cars
- NLP (Natural Language Processing)
- Other Games: Ex. Go, Chess, etc.



Recommendations

- I would've liked to tried other types of methods while developing this neural net. Methods that optimized for computation time or moves.

- Creating a better interactive visualization

- Reattempt this project using self collected image data

Steamlit launching time!