Editing Journal For My Poker Solver

Eric Wang

Start: December 23, 2023

1 Code Structural Diagram

PreflopHands Class

- contains a 75x75 array of the preflop ranking of every starting hand in poker

Card Class

- each card is given a number and a suit
- each card also has a unique cardValue to make it simpler to distinguish different cards. the card value is the number times 5 + the suit, suits are 1-4

Hand Class

- a hand has 2 cards
- additionally, a hand stores the information about whether or not it is suited

Deck Class

- a deck is initialized as a list of all 52 cards, shuffled randomly

Board Class

- a board has a list of all the cards on it
- it comes with methods based on a deck to deal the next cards in accordance to the game

Player Class

- a player has a hand, a stack, and a position for which order they take their turn
- a player also comes with the actions fold, check, raise quarter-pot, raise half-pot, raise three-quarter-pot, raise pot, and reraise

Round Class

- each round has a deck, from which each player is dealt a hand
- each round also has a board and a pot, and each round implements the normal circle of betting.
- after every circle, the flop, turn and river are dealt and eventually the result of the game is reached, meaning that each player's hand is ranked and a winner is chosen

Game Class

- a game has a number of players, the starting stack for each player, as well as the number of rounds that will be played.

2 Dec. 23, 2023

Brainstorm:

- first, implement a program that calculates the odds of every hand winning without any other information, preflop, postflop, after the turn and river. This way, it will make it easier later on to categorize hands by strength to ease the decision making process for the machine
- then, we can implement a neural net to play the actual hands, passing in hand strength and betting history as parameters
- for this, we will have to learn python as it is likely the easiest language to do this in
- after implementing the first neural net, we can see how it does playing against me/itself and figure out how to best train the program going forward.

Report:

- hello world in python!

3 Dec. 24, 2023

Plan:

- start hardcoding all the preflop hands and setting up the infrastructure of my code
- figure out how classes work and define a hand class, card class, maybe a board class

Notes:

- implemented hand, card, and deck classes
- hardcoded a small portion of the preflop hands. look to finish them tomorrow and then start working on implementing the rules of poker and who has the best hand

4 Dec. 25, 2023

Notes:

- finished hardcoding the preflop hands
- working on how to select the winning hand given a board and each player's hand

Notes:

- Preflop hands have been implemented.
- We need a way to calculate odds postflop, as precalculation would be 10^{12} and take way too long.
- Instead, we will just take the hand we currently have and compare it to every other possible hand as well as every other possible runout, which is around 10^7.
- So now, we need to implement picking a winning hand from a board and a list of hands
- we also need to be able to run the actual game of poker
- wrote up a general outline for all of the classes that are made

5 Dec. 27, 2023

Notes:

- took a break yesterday, as I had some other arrangements
- spent today implementing choosing a winning hand after the board has been dealt
- also finished implementing the round class

6 Dec. 28, 2023

Notes:

- spent some time on researching how to construct the actual machine learning parameters of the code
- the structural aspect and the actual game is approaching completion, but the neural network related stuff seems to be posing a few problems
- how am I going to determine which modification to make with the regret adjusting? And how am I going to determine whether or not the program is improving?

7 Jan. 2, 2024

Notes:

- took a break, back from some celebrations as well as recovering from being sick

- spent the day on the Micrograd$_{F}ollow_{A}longaswellasneuralnetworknotes.tex, watchingvideostostudyneur$

8 Jan. 5, 2024

Notes:

- finished understanding neural networks and implementing micrograd, taking some time to draw out the neural network on paper taking ideas from "Building a neural network from scratch" -Samson Zhang video
- include a softmax function, and activation function which are very important towards the functionality

9 Jan. 7, 2024

- re-implement micrograd to make it my own in Neural$_{N}etwork_{C}lasses_{B}ackpropogation.py, inordertocustom$