DeepStack Poker

CS 5804 MINI PROJECT - GROUP 10

Intro

1. Nathan Arnold (MEng CS)
2. Hyndavi Venkatreddygari (MEng CS)
3. Adam Schneider (MEng CS)
4. Sai Nikhita Nayani(MEng CS)

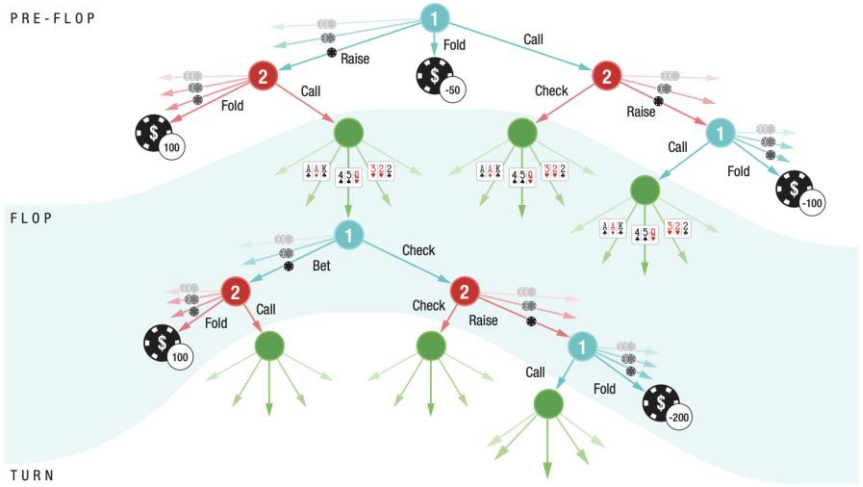
Problem Description

Can AI learn the skills needed to beat Professional Poker Players?

- Players hidden cards
- Cards on the board
- Sequence of betting

Leduc Hold'em:

A variation of poker where each player receives 1 card face-down from a deck of six cards and one card is shown on the board after betting rounds. The player with the highest card shown at the end wins. No limits to betting or raising each round

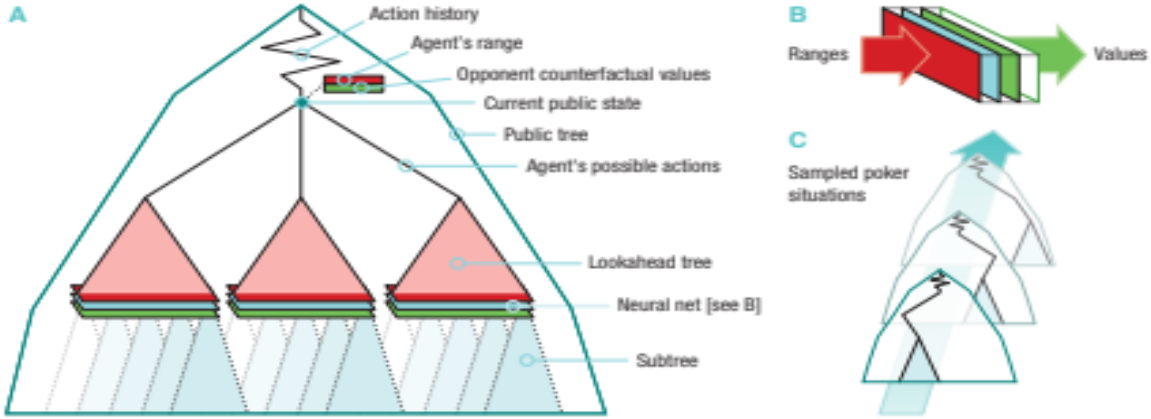


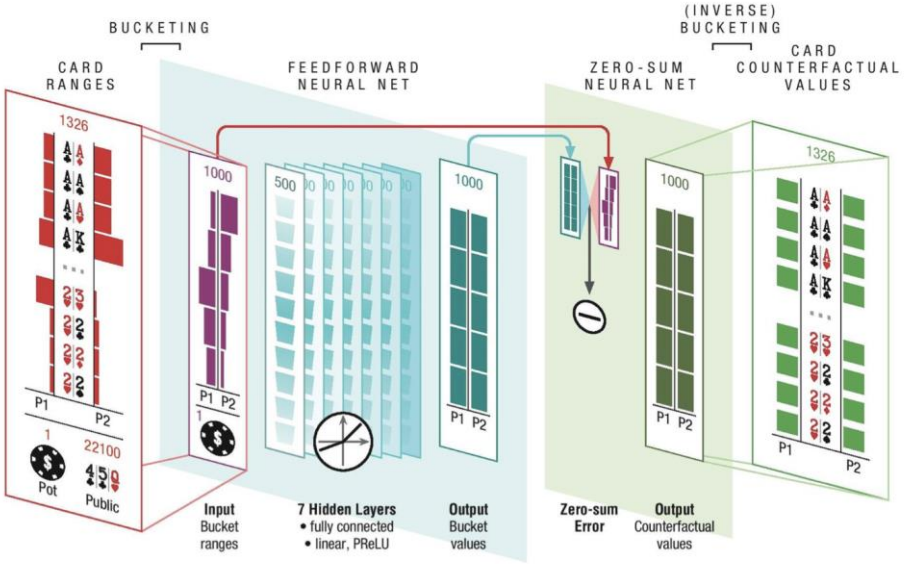
Poker Tree

- Incomplete knowledge of state
- Needs contextual knowledge of previous moves
- Difficult to build an evaluation function

Approaches

- The DeepStack Algorithm - Three ingredients
  1. Continuous re-solving
  2. Depth-limit
  3. Restricted actions





Why DeepStack works?

- No Explicit Abstraction
- Adapting a heuristic search like method
- Suitable evaluation function using deep neural networks

Results

GAME DEFINITION

nolimit

numPlayers = 2

numRounds = 2

stack = 1200 1200

blind = 100 100

numSuits = 2

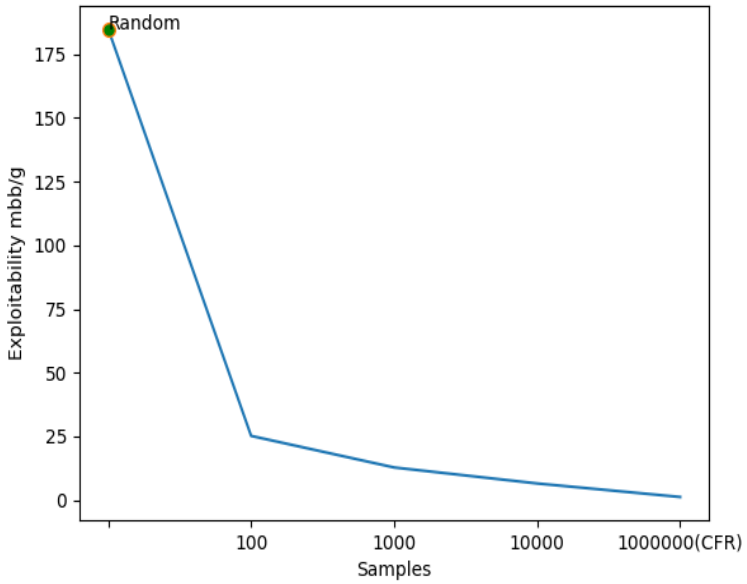
numRanks = 3

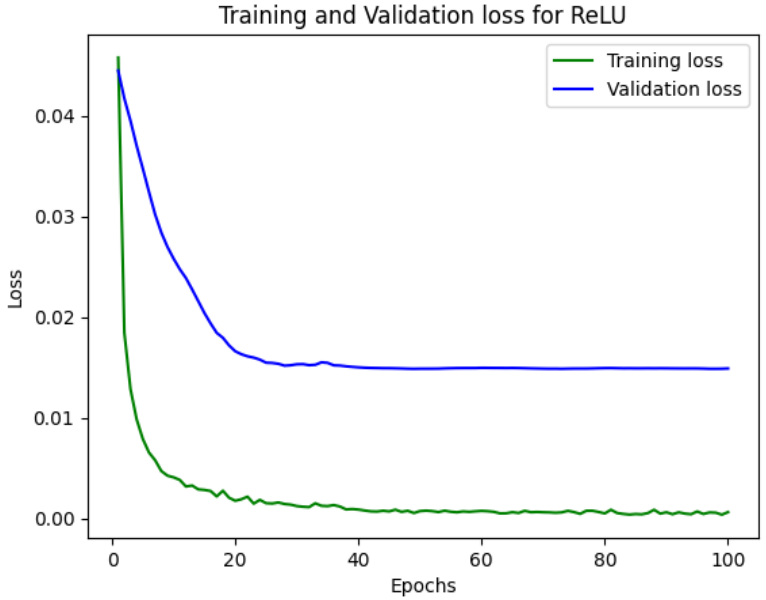
numHoleCards = 1

numBoardCards = 0 1

Run	Score (Chip)	Random Agent Action Probabilities
Random vs Random	SCORE: 3714|-3714: Random|Random	fold = 0.06 call = 0.47 Raise = 0.47
DeepStack vs Random	SCORE: 15602|-15602: DeepStack|Random	fold = 0.06 call = 0.47 Raise = 0.47
DeepStack vs Never Fold	SCORE: 3372|-3372: DeepStack|NoFold	fold = 0.0 call = 0.5 Raise = 0.5
DeepStack vs Raise often	SCORE: 6216|-6216: DeepStack|Raise Often	fold = 0.06; call = 0.094 raise = 0.846
DeepStack vs Always Fold	SCORE: 5285|-5285: DeepStack|Always Fold	fold = 1.0; call = 0.0 raise = 0.0

Exploitability





Lessons learned

- Integration of machine learning and reinforcement learning algorithms
  - When environment is not stationary, deterministic machine learning algorithms have been proven effective in approximating the value or policy function

Future Work

- Scaling it up to larger games with more players and more complex decision-making
- Focus on developing techniques that can adapt to different game variants and rule sets
- Techniques can be applied to other problems in AI and beyond

Q&A + References

Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K.,Johanson, M., & Bowling, M.(2017). DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker. Science, 356(6337), 508–513. https://doi.org/10.1126/science.aam6960

DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker | Papers With Code

Github repo: lifrordi/DeepStack-Leduc: Example implementation of the DeepStack algorithm for no-limit Leduc poker (github.com)