May the Flop be with you - An Intelligent Texas Hold'em PokerBot

Srivatsan Srinivasan, Sebastien Baur, Donghun Lee

CS281-Advanced Machine Learning, SEAS, Harvard University

Objective

To build a self-trained intelligent Texas Hold'em PokerBot that is capable of

- Learning hand strength using neural nets(NN).
- Choosing appropriate training experiences.
- Evolving its game via NN fictitious self-play.
- Beating baseline and advanced AI opponents.

Key Concepts

Convolutional and Fully-Connected Neural Nets, MDP, Double Deep-Q Networks, Policy Networks, Prioritized Experience Replay, Nash Equilibrium, Neural Fictitious Self-Play.

Introduction

No-Limit Texas Hold'em Poker presents an interesting template for probabilistic modeling and deep RL as it involves sequential decisions, imperfect information, adversarial behavior and humongous state action space where simulating possible cases for supervised learning is prohibitive, we use fictitious self-play, which is proven to achieve near-optimal Nash Equilibrium, using Double DQNs and policy networks with prioritized experience replay buffers so that agent plays against its own clones to learn optimal behavior. Also, we employ state space abstraction through a NN hidden feature layer that infers the win probability of given hand and board.

Problem Definitions

- Hand(H) - 13 * 4
- Plays(P) - Actions in four rounds(PF,F,T,R).
- Board(B) - 4(PF,F,T,R) * 13 * 4
- Game Variables(G) - Pot, Dealer, Stacks
- State - (H,B,G,P)
- Action - (Call, Fold, Check, All-In, Raise-Bin).
- Rewards - Δ ($) with each action.
- Setup - Initial : $100 each, SB : $1, BB : $2

Card Featurizer

A model that implictly represents hand strength as function of cards and board with network weights(State-Space Abstraction - Eg: (2H, 3S) and (4H,5S) with board of KH,KD,KC treated the same)

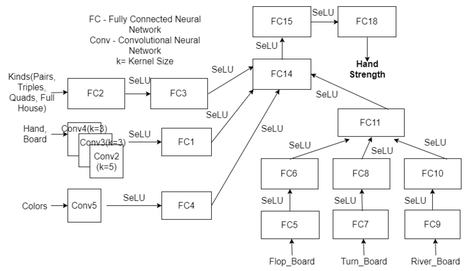

Figure 1: Card Featurizer Network

Model Architecture

Here, we have a shared network architecture that predicts Q-values and π values using card features and game states.

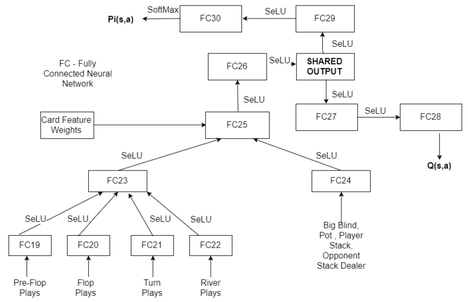

Figure 2: π and Q network architecture

Important Result

Here is where we specify the most important result of the project. It should be the biggest takeaway for the audience from our project and approach.

Neural Fictitious Self-Play(NFSP)

Init : Game transitions memory(using PER) and best-response memory M_{RL}, M_{SL} ; Q-network, π network. Repeat the following until termination.

- Choose action with η probability from β or π.
- Populate M_{RL} using each (s,a,r,s,a') and M_{SL} using (s,a) when action is chosen from β.
- Train Q-network(off-policy RL) to yield ϵ-greedy strategy β and average policy network - π learned from prior best responses.

Why choose from both networks?

Classic Off-Policy Learning would involve playing π and learning Q. This would limit the experiences that π is trained on. Here, We train π on experiences generated from our best-response behavior β while also doing off-policy Q-Learning. We sample our next action from both networks probabilistically in order to create a mixture of exploration and exploitation.

Results

Placeholder

Image

Figure 3: Figure caption

Nunc tempus venenatis facilisis. Curabitur suscipit consequat eros non porttitor. Sed a massa dolor, id ornare enim:

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Table 1: Table caption

Conclusion

Nunc tempus venenatis facilisis. **Curabitur suscipit** consequat eros non porttitor. Sed a massa dolor, id ornare enim. Fusce quis massa dictum tortor **tincidunt mattis**. Donec quam est, lobortis quis pretium at, laoreet scelerisque lacus. Nam quis odio enim, in molestie libero. Vivamus cursus mi at *nulla elementum sollicitudin*.

Additional Information

Maecenas ultricies feugiat velit non mattis. Fusce tempus arcu id ligula varius dictum.

- Curabitur pellentesque dignissim
- Eu facilisis est tempus quis
- Duis porta consequat lorem

Future Work

- Hyperparameter tuning and initialization
- Opponent Modeling
- Incorporation of Game Theory insights
- Simulations against commercial AI

Acknowledgements

Nam mollis tristique neque eu luctus. Suspendisse rutrum congue nisi sed convallis. Aenean id neque dolor. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.