Playing Cards with Machines

Robert Grönsfeld

Baloise

2022

Motivation

- Idea for Code Camp: Extend Open Source Project for Poker with additional functionality
- DeepStack.ai: Leduc Hold'em
- Problem: Project too old and not maintained
- Found another project RLCard
- Implement Mau-Mau as simple card game into the framework

What we played: MauMau

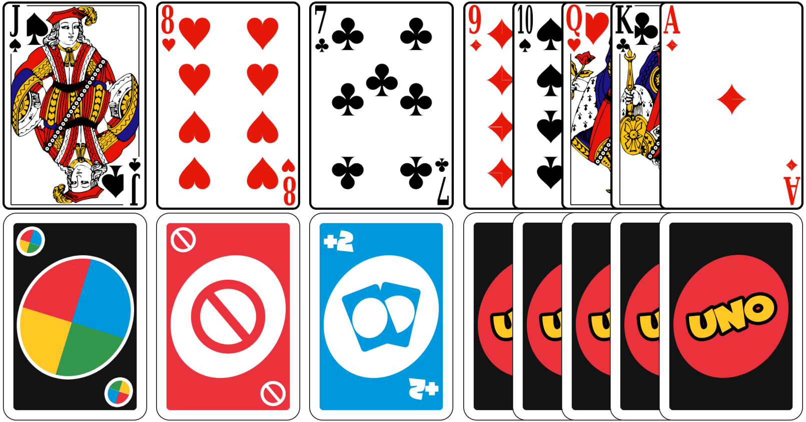

Figure: MauMau = Uno +

7♣	7
8♥	8
9♠	9
10♦	10
D♥	10
K♠	10
A♦	11
V♣	20

Figure: Card values

Mau-Mau Rules

- Card deck: 32 cards, 5 hand cards at start
- Process
  - play a card with the same rank or suit
  - if not possible: draw a card
- Goal: play all hand cards
- Special cards
  - 7♣ 7♥ 7♦ 7♠ : next players has to draw 2 cards
  - 8♣ 8♥ 8♦ 8♠ : next player misses a turn
  - V♣ V♥ V♦ V♠ : can be played on any suit, player can wish a suit for the next card

Part I

RLCard Toolkit

RL Card

- Toolkit for Reinforcement Learning in Card Games
- DATA Lab at Texas A&M University
- Programming language Python
- different games (also Poker)
- different Reinforcement Learning algorithm
- possibility to play against trained model
- possibility for UI with RLCard Showdown (game specific)

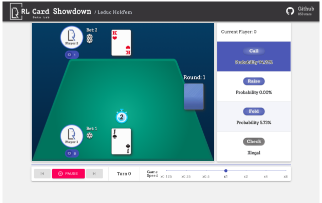

Figure: RL Card Showdown

Implementation

- Structure for implemented games
  - Game: contains players, dealer, round, payoffs
  - Round: process of a round in a game
  - Dealer: shuffling and giving cards
  - Player: plays the cards following a strategy
  - Judger: determines winner
- Train different models to the implemented game

Important Methods

- perform_draw_action: handling when player draws a card
- perform_card_action: handling when player plays a card
- get_legal_actions: allowed cards in the given situation
- Reward function get_payoffs: Decides how many points the winner/penalty points the loser gets
- RL algorithm tries to optimize the reward function

Part II

Theory

Artificial Intelligence

- hard to define, since intelligence itself is precisely defined
- "Artificial intelligence leverages computers and machines to mimic the problem-solving and decision-making capabilities of the human mind."
- distinction: strong and weak AI
  - strong: generic AI that can solve generic issues
  - weak/narrow: for one specific issue

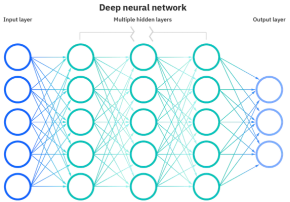

Figure: Deep Neuronal Network

Deep Learning vs. Reinforcement Learning

Deep Learning is essentially an autonomous, self-teaching system in which you use existing data to train algorithms to find patterns and then use that to make predictions about new data.

Reinforcement Learning

- is an autonomous, self-teaching system that essentially learns by trial and error
- performs actions with the aim of maximizing rewards, or in other words: learning by doing
- are both systems that learn autonomously.
- aren't mutually exclusive.

Markow Decision Process

A Markov decision process is a 4-tuple (S, A, P_a, R_a), where:

- S is a set of states called the state space,
- A is a set of actions called the action space (alternatively, A_s is the set of actions available from state s),
- $P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ is the probability that action a in state s at time t will lead to state s' at time $t + 1$,
- $R_a(s, s')$ is the immediate reward (or expected immediate reward) received after transitioning from state s to state s', due to action a
- S is a set of states

Optimizing for highest reward

Algorithms supported by RLCard

- DMC: Deep Monte Carlo
- DQN: Deep Q-Network
- NFSP: Neural Fictitious Self-Play
- CFR: Counterfeit Regret Minimization

Part III

Training

Training

1. Effect of payoff function
2. Effect of time
3. Effect of algorithm
4. Playing with oneself
5. Training with existing models
6. Tournament

Effect of payoff function

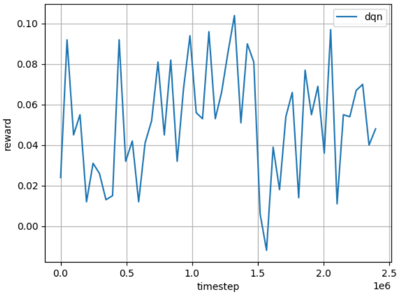

Figure: DQN: win or lose

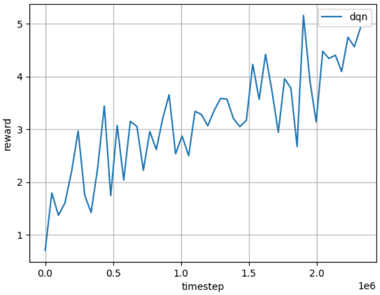

Figure: DQN: count points

- Deep-Q Learning, $t \approx 2\,500\,000$, random agents
- Average payoff stagnates or increases
- Quality of payoff function determines success

Effect of time

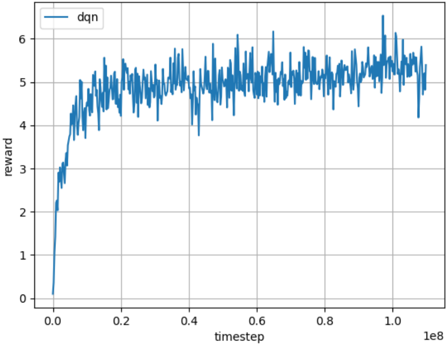

Figure: DQN: count points

- Deep-Q Learning, $t \approx 100\,000\,000$, random agent
- Reward at global maximum after $20\,000\,000$ steps
- Longer training will not lead to better results per se

Effect of algorithm

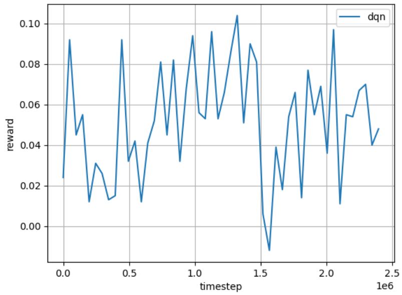

Figure: NFSP: win or lose

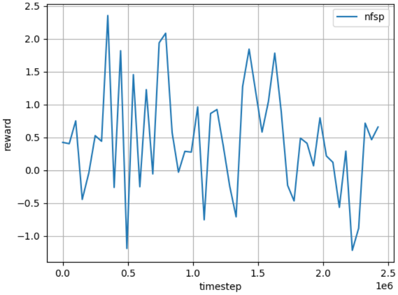

Figure: NFSP: count points

- Neural Fictitious Self-Play, $t \approx 2\,500\,000$, random agents
- Regardless of the payoff function the AI can not beat a random player

Playing against oneself

- Deep-Q Learning, $t \approx 8\,000\,000$, experienced DQN agent
- Beats experienced agent after 2 000 000 time steps
- Only slightly better average payoff, even after 8 000 000 steps
- DQN agents do not seem to improve iteratively

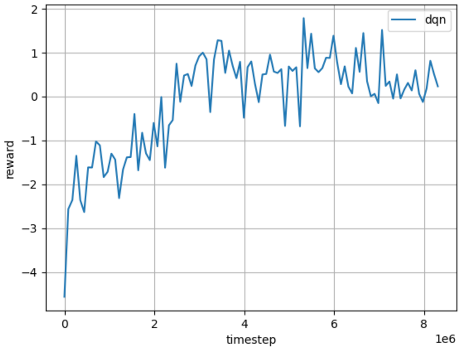

Figure: DQN vs DQN: count points

Training with existing models

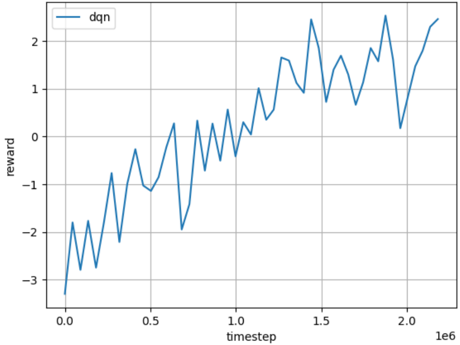

Figure: DQN vs DMC

- Deep-Q Learning, $t \approx 2\,000\,000$, Deep Monte-Carlo agent
- Beats adversary after $1\,000\,000$ time steps
- Eventually achieves a slightly better average payoff

Tournament

	$DMC4$	$NFSP$	DQN_{DMC}	DQN_{DQN}	$DQN100$	$Random$
$DMC4$	1.3	3.4	-0.7	0.2	-0.3	4.5
$NFSP$	-1.8	-0.2	-2.5	-2.3	-2.4	2.0
DQN_{DMC}	1.2	4.0	0.7	1.4	1.6	4.7
DQN_{DQN}	0.9	3.5	0.2	1.2	0.6	5.3
$DQN100$	1.0	3.6	0.7	1.3	1.4	5.1
$Random$	-3.1	-1.5	-4.0	-3.2	-3.3	0.5

- Deep-Q Agent trained with Deep Monte-Carlo agent wins the tournament
- All agents are better than a random player
- Player position gives an advantage of up to 1.4 points

Part IV

Conclusion

Conclusions

- games with much randomness impact algorithm performance
- leads to good results for stable, easy to simulate scenarios with clear rules
- algorithms show significantly different results / learning effects
- reward function is very important
- easy to implement
- learning was slow on CPU; using GPU may lead to better results
  - no strategy knowledge required
  - only define rules
  - no need to know AI in depth

Sources

- https://arxiv.org/pdf/1910.04376.pdf
- https://www.forbes.com/sites/bernardmarr/2018/10/22/artificial-intelligence-whats-the-difference-between-deep-learning-and-reinforcement-learning/?sh=5c1f658a271e
- https://www.ibm.com/cloud/learn/what-is-artificial-intelligence
- https://hci.iwr.uni-heidelberg.de/system/files/private/downloads/541645681/dammann_reinfocement-learning-report.pdf
- https://arxiv.org/pdf/1910.04376.pdf
- https://www.simplilearn.com/tutorials/deep-learning-tutorial/neural-network
- https://rlcard.org/