Learning to Play Blackjack using Deep Reinforcement Learning

Chris Durbin

Research Draft Summary

I have achieved my original research project goal of being able to learn optimal play for blackjack using deep reinforcement learning with a Deep Q-Network (DQN) agent. In the remaining weeks I have decided to attempt a more difficult challenge of counting cards to achieve a positive expected reward and learn optimal bet sizes with another DQN agent. My primary goal with the project is to continue to learn and gain more experience with deep reinforcement learning so that I can apply it to other problems in the future after graduation. I fleshed out the paper some to give an idea of what I am attempting to achieve by the end.

Introduction

The purpose of the project is to attempt to use deep reinforcement learning to learn optimal play in the game of Blackjack. By counting cards and properly sized bets, human players have been able to gain an advantage over casinos and have a positive expected reward. In the project I will attempt to entirely use deep reinforcement learning algorithms to achieve the same goal.

Related Work

There are relevant publications for both optimal play for the game of blackjack as well as for deep reinforcement learning that I relied on to help design my research project.

Blackjack

Vidámi describes how card counting can be utilized to improve the odds such that there are situations based on the remaining cards in the deck that the player has odds better than the dealer (Vidámi, 2020). In this situations the player can increase their bet size to increase their expected return meaning that it should be possible to learn how to play such that the expected return is positive.

Reinforcement Learning

There is a large amount of prior work on using reinforcement learning to learn how to play games without any built-in knowledge of optimal play for the game. In the past ten years there has also been a large focus on using deep reinforcement algorithms. In a ground-breaking paper for deep reinforcement learning, Minh introduced the Deep Q-Network algorithm which used a neural network and experience replay to estimate future rewards for a given state and all possible actions (Minh, 2013). Their success on a large number of

Atari games without any game specific information makes it likely that a similar approach would also work well for a game such as blackjack.

One of the largest concerns with the approach to use deep reinforcement learning is the "Deadly Triad" described by Sutton and Barto. When combining function approximation, off-policy learning, and bootstrapping (such as temporal difference methods), learning can diverge, and the value estimates become unbounded (Sutton, 2018). As such I may need to employ special techniques and perform many tests with different hyperparameters to attempt to avoid instability. One example technique to try was provided by Wang with the concept of dueling DQN networks which utilize two different neural networks when performing learning and identifying targets for the error calculation. They freeze the target network for each minibatch of experience replay learning so that the target does not keep updating as the examples from the batch update the weights during backpropagation leading to more stable weight updates (Wang, 2016).

Research Project Problem

There are two main goals for the research project which are to answer the following questions:

1. Can we use deep reinforcement learning to learn optimal play for a blackjack hand?
2. Can we utilize card counting and deep reinforcement learning to have a positive expected value for standard casino blackjack rules with a minimum bet size and no maximum bet size?

My hypothesis is that the answer to both questions is yes, and that we will be able to train an agent using deep reinforcement learning to play blackjack and have a positive expected return without any programmed knowledge of optimal play nor using the Kelly criterion calculation (Kelly, 1956) to determine bet size. As mentioned previously we cannot guarantee that learning will converge, and it is not a given that it will be possible to implement a deep reinforcement algorithm to achieve our goal making this a worthwhile research project.

Blackjack Rules

The project will utilize standard casino blackjack rules as follows:

- The player (agent) places a bet that is at least the minimum bet size.
- Both the player (the agent) and the dealer will be dealt two cards. The player's cards will both be dealt face up and the dealer will have one card dealt face up and the other face down.
- If the player has a natural 21 they will automatically win 1.5 times their bet unless the dealer also has 21 in which case it is a push. If the dealer receives a natural 21 and the player does not the dealer automatically wins, and the player's bet is lost.

- Otherwise play continues. The player can choose to hit or stay. If the player hits they receive another card. As long as they have less than 21 they can continue to hit.
- If the total value of the player's cards is greater than 21 they lose the hand.
- If the player has 21 or fewer and they choose to stay it is then the dealer's turn.
- The dealer will always stay with 17 or higher and always hit with 16 or less. If the dealer has a total greater than 21 or the dealer stays with fewer points than the the player, the player wins the hand.
- If both the dealer and player have the same value for their hands it is a push
- If the dealer has a greater value than the player, the player loses the hand.

When the player wins the hand they receive a reward equal to their bet, on a push no reward is received, and on a loss the player loses their bet. The one exception is for a winning natural 21 on the first two cards the player receives 1.5 times their bet.

Methods

The project includes several components to implement the reinforcement learning environment:

- Blackjack environment that implements the mechanics and enforces the blackjack rules.
- DQN agent to make the decisions for selecting actions for playing hands.
- DQN agent to make the decision on the bet size to use for each hand (not fully implemented yet).
- Driver to put all the components together and run the experiments.

TODO for final paper – Add an illustration of the component interaction.

In addition to the deep reinforcement learning components two other agents are included as a baseline comparison for selecting actions for playing hands. One agent implements the first-visit Monte Carlo algorithm, and the other agent implements the Q-learning algorithm.

To train my agents I repeat the same experiment for TBD episodes. The constraints of an episode are as follows:

- Agent starts with a balance of $2,500
- Minimum bet for a hand is $10
- No maximum bet for a hand
- Play continues until the agent no longer has enough to place a minimum bet or has completed 1,000 hands.

For my tests after the initial training has completed, I repeat the same experiment twenty times and track the results of the individual experiments as well as the average results across all twenty experiments.

Metrics

The primary metrics I am tracking are the final average balance for each of the experiments, and if the balance is less than the minimum bet, I track the number of hands played for that experiment.

In addition, I track a few other metrics to better understand how quickly the agent learns, both in terms of number of hands played as well as the wall clock time to train.

Results

TODO for final paper – includes charts and results averaged across all experiments with:

- Expected return over time
- Win rate over time
- Bet size percentage over time

Here's an example of the expected return over time (prior to implementation of bet size).

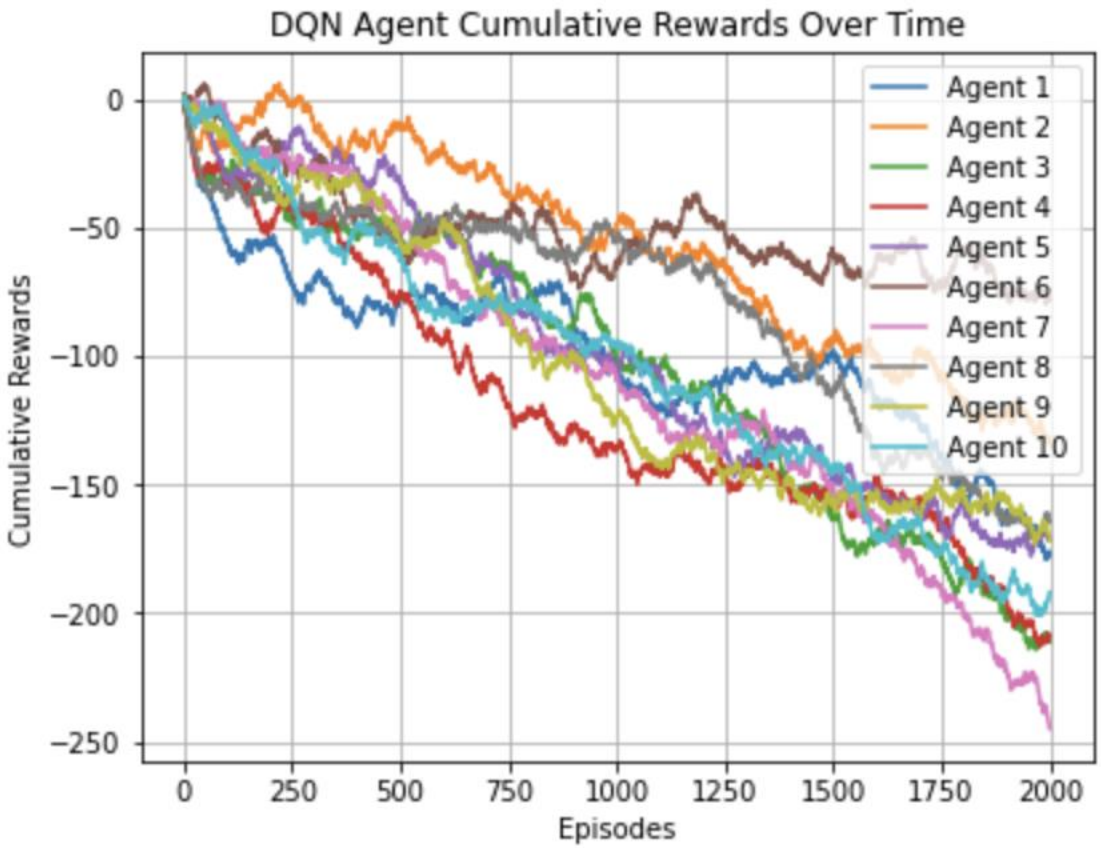

Figure 1 – DQN agent cumulative rewards over time

TODO for final paper - Answer whether the hypothesis was correct or incorrect.

Conclusions

TODO for final paper – based on the results what would be the next steps for further relevant research.

References

1. Kelly, John L. "A new interpretation of information rate." *the bell system technical journal* 35.4 (1956): 917-926.
2. Riedmiller, Martin. "Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method." *Machine Learning: ECML 2005: 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005. Proceedings 16*. Springer Berlin Heidelberg, 2005.
3. Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." *arXiv preprint arXiv:1312.5602* (2013).
4. Schaul, Tom, et al. "Prioritized experience replay." *arXiv preprint arXiv:1511.05952* (2015).

5. Wang, Ziyu, et al. "Dueling network architectures for deep reinforcement learning." *International conference on machine learning*. PMLR, 2016.
6. Sutton, Richard S., and Andrew G. Barto. *Reinforcement Learning, second edition: An Introduction*. MIT Press, 2018.
7. van Hasselt, Hado, et al. "Deep Reinforcement Learning and the Deadly Triad." *arXiv preprint arXiv:1812.02648* (2018).
8. Vidámi, M., Szilágyi, L., and Iclanzan, D. "Real Valued Card Counting Strategies for the Game of Blackjack." *Neural Information Processing*, edited by H. Yang et al., vol. 12533, Springer, Cham, 2020, pp. 1-10. doi:10.1007/978-3-030-63833-7_6.