Machine Learning Game Theory, an Investigative Study

Susan Hamilton, Robert Hamilton

ABSTRACT:

Due to rapid progress, machine learning, a form of artificial intelligence, has quickly become a transformative technology [1]. It has been utilized in many diverse areas such as self-driving cars, pattern recognition, sentiment analysis, and many other applications. In particular, reinforcement learning (RL), a specific branch of machine learning, uses a goal-directed agent to optimize the performance in an uncertain environment. In this study in particular, we investigate an application of RL in game theory, utilizing the RL methodology within a simple goal-directed game and characterize the efficiency and effectiveness of the theory and associated algorithms in a concrete example. That characterization is achieved through measuring the lifetime of the character as a stand-in for the learning progress towards the goal. We found that the algorithm was quite effective towards learning, and we found indications of large discontinuous jumps, suggesting a process of discovery in addition to the gradual learning process.

GOAL:

The project goal was to explore the methodology and quantify the learning capability afforded by the reinforcement learning model within the particular application of a simple game. We hoped to characterize the agent's learning progress over time through the use of variations in the algorithm. [3]

DESIGN:

Our project design consisted of four major components: The game, the agent, the controller, and the reinforcement model. The agent was the component which simulated an independent actor that interacted with the game. The controller managed the environment and interactions between the agent and the game. Finally, the reinforcement model handled the learning algorithm.

The game contained the following rules: Within the game, the character had simple options for what to do, such as go to the wild and pick up resources which they would then have to travel to the market in order to sell and buy food. They were also able to merge items in order to sell the new item for more money. The objective of the game was to survive, and a character died the day after their hunger bar fell all the way to zero. After writing the game, we created a controller and its agent to control the player in the game, and the model component that they used was the RL method to develop a strategy that encouraged it to win, making it assess a unique

“score” for each potential move- a higher score meaning a better chance at surviving. After each game, the score values were updated based on what the machine learned in that game.

TEST RUNS:

There were three training modes utilized which were (1) no learning with completely random moves, (2) for a configurable percentage of time, picking a random move instead of the currently known best, and (3) a test mode in which the optimal move will always be chosen (no learning occurs). The continuous reinforcement and feedback is applied in the second strategy, the test case was used to measure the learning progress, and the first one will be used as a baseline against which the second strategy will be compared. Each run consisted of two phases: the training session and the test phase. The training session entailed running a specified number of games using either strategy 1 or 2 and the test phase consisted of playing a specified number of games using the previously defined learning.

DATA COLLECTION

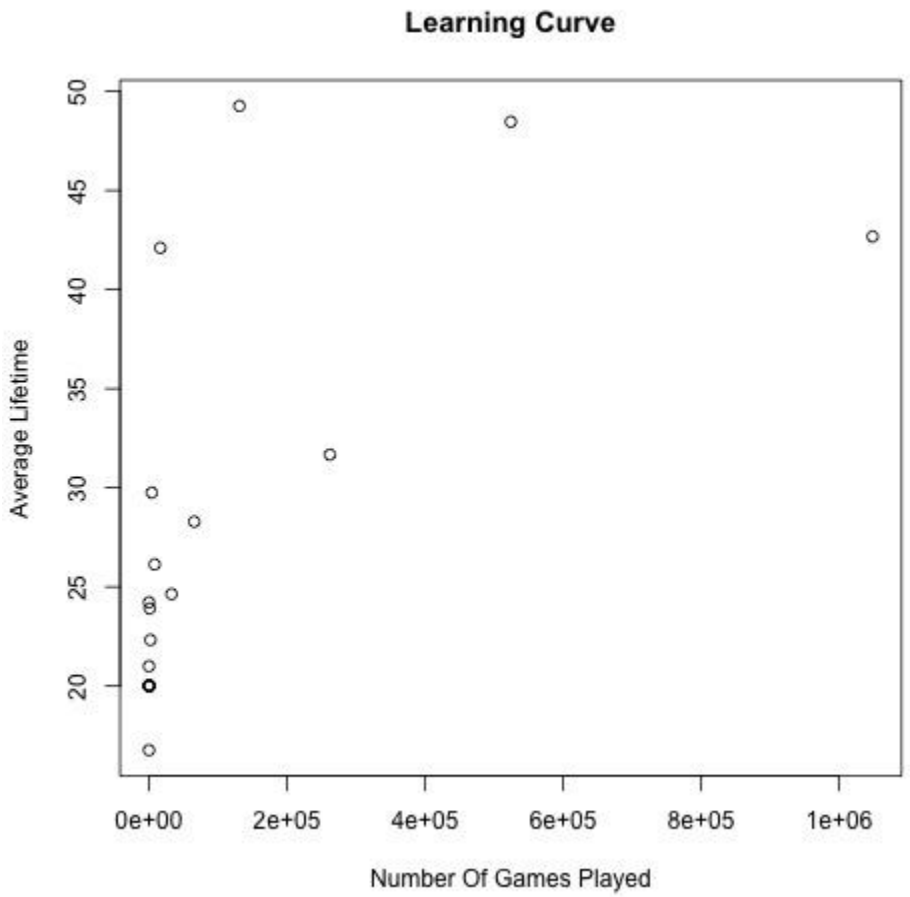
After each training session, in which a machine played a certain number of games using either strategy 1 or 3, we played a fixed number of games using strategy number 2 and captured the average lifetime of the player. This data was captured in graphs that detailed the lifetime of the player versus the number of games played during training. Then,the rate of learning was determined by discerning how rapidly the lifetime had increased as the number of games increased.

We compared three different things within our training session: how long the player lived without any learning at all and only making random moves, how long it lived if the training session consisted of a fixed number of games with the learning applied only at the end of the session, and how long the player lived after using reinforcement learning after each game in the training session.

DATA ANALYSIS:

After each training session, in which the machine played a certain number of games using either strategy 1 or 3, we played a fixed number of games using strategy number 2 and captured the average lifetime of the player. This data was captured in graphs that detailed the lifetime of the player versus the number of games played during training. Then, we determined the rate of learning by discerning how rapidly the lifetime increased as the number of games increased.

DATA:



RESULTS:

For each run, I ran 3 million games in increments and measured the performance after each increment and repeated this process ten times in order to obtain an average of our data for each learning increment. I observed significant learning progress as a larger number of games were played, indicating that the algorithm functioned to improve the performance and encourage learning by the agent. I found that the exploratory algorithm worked better than the random choosing algorithm, as the machine consistently learned faster and lived longer within this strategy. I also noticed an unexpected result where there was large discontinuous jumps in performance, and the machine lived for over 10,000 moves in one game, a pattern that one would not expect to see as a result of the gradual learning algorithm.

In graph 1, each point represents the average lifetime of the game actor over one hundred tests versus the number of games played for each test. The data starts out with a pattern of steep growth, indicating the rapid learning of the machine, as the average lifetime is increasing as the machine plays more games and applies information learning from its previous games to its next ones. At a certain point, the curve undergoes a discontinuity in that it spikes to a much higher average lifetime, suggesting that a “discovery process” has taken place, where the machine has randomly discovered how to live for a very large number of moves, greatly increasing the average lifetime at this point. After this anomalous spile, the graph gradually decreases, which may be due to the data returning to the point of convergence of the learning.

CONCLUSION:

In conclusion, our reinforcement learning model validated the process of machine learning and demonstrated strong convergence, as the machine learned under our model to increase its average lifetime rapidly as the number of games played increased.

These results demonstrated a higher level of learning, which suggests a pattern of discovery, rather than just a convergence in the normal learning process. These findings indicate that further research could be done to better understand the discovery process and further characterize it. In addition, future research could also explore differences in learning with varying degrees of difficulty within the game parameters.

BIBLIOGRAPHY:

[1]: Nichols, James A et al. “Machine learning: applications of artificial intelligence to imaging and diagnosis.” Biophysical reviews vol. 11,1 (2019): 111-118. doi:10.1007/s12551-018-0449-9

[2]: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[3]: Watkins, C.J.C.H. (1989): Learning from Delayed Rewards (ph.D. thesis), University of Cambridge

Kaelbling, Leslie Pack, Michael L. Littman, and Andrew W. Moore. "Reinforcement learning: A survey." Journal of artificial intelligence research 4 (1996): 237-285.

M. Abdoos, "A Cooperative Multiagent System for Traffic Signal Control Using Game Theory and Reinforcement Learning," in IEEE Intelligent Transportation Systems Magazine, vol. 13, no. 4, pp. 6-16, winter 2021, doi: