Title

Investigating Emergent Strategies in an Iterated Prisoner’s Dilemma Environment With the use of Machine Learning.

Abstract -194 words

This paper investigates emergent strategies in the Iterated Prisoner’s Dilemma (IPD) using Machine learning. The study focuses on the comparison between the Machine learned strategies and already existing documented ones with a focus on the Tit for tat strategy. The Study is split into three parts the AI vs Random choice, AI vs AI and AI vs Tit for tat.

In the first experiment, the AI emerges the Always Defect strategy when facing an opponent that chooses at random. The second experiment, The AI trains against itself, the AI also emerge with the Always Defect strategy but faster than expected. The third experiment, aiming to force the emergence of the Tit for tat Strategy results in failure with it emerging as the Always Cooperate strategy.

Analysis reveals that all the experiments align with predictions with a few differences mainly on how fast it reached those predictions. The third experiment did ultimately fail which prompt consideration of flaws in the design of the experiment.

The results are interesting enough to possibly prompt future work; however, they are not substantial enough to have any noticeable effect on the current research, industry, and future in this field.

Introduction – 641 words

Introduction (sub)

My exploration delves into the ethical dilemma that emerge, within the realm of game theory. Focusing on the Iterated Prisoner’s Dilemma (IPD), Which is an extended version of the original Prisoner’s Dilemma (PD). PD is a single round, IPD extends the game to multiple rounds. IPD allows for strategies to develop and evolve over multiple rounds. The classic example for PD is two prisoners, A and B, who independently choose to cooperate (Stay silent) or defect (Snitch), based on their choices they face Harsher or lesser sentences. The conflict lies in the tension between mutual benefit and individual gain.

Significant research has been done on IPD, particularly by a man named Axelrod –[1]. The research done highlighted various strategies. One strategy is that I will be focusing on is Tit for Tat (TFT) – [2]. This strategy won in Axelrod’s tournament overall despite never winning a single game.

This study is using Reinforcement Learning (RL), a Machine Learning (ML) variant where an agent’s decision in an environment lead to rewards or punishments. A variant of RL is Proximal Policy Optimisation (PPO) this is the algorithm I will be using, and it is chosen for its simplicity and reliability.

This Paper explores the strategies emerging from ML in the IPD environment comparing them to already well researched and documented strategies, attempting to identify which strategies emerged and attempting to force the agent to emerge as TFT by attempting to alter the environment. Environmental changes aim to uncover potential impacts and patterns in the agent’s behaviour.

TFT, considered to be one of the best in a simple version of IPD, works best when its opponents are cooperative strategies. However, when introducing randomness to simulate real-world misunderstanding and mistakes, TFT’s gets diminished performance, revealing its limitations.

The experiment will be conducted in three parts, each involving the AI facing off against a different opponent. The first opponent will randomly choose to cooperate or defect with 50/50 odds, serving as base or standard for what should be the best default strategy. The second part will have the AI facing off against itself, offering possibility of emerging unique results. In the third part, the opponent is the TFT strategy, this is where I aim to change the environment to force the AI to emerge the TFT strategy.

This multifaceted approach intends to uncover differences between emerging strategies and identify overarching patters, hopefully to contribute insight to the realm of PD strategy using AI.

Background

Prisoner dilemma (Sub)

The Prisoner’s dilemma believed to have been formulated by Merrill Flood and Melvin Dresher in 1950 is a thought Experiment that has been heavily researched. A key figure in this research is Robert Axelrod who created a tournament in 1980 - [1] In this tournament he placed all the strategies against each other in an iterated prisoner’s dilemma and evaluated the outcomes. A strategy that stood out as the winner during this tournament was the TFT strategy. This is the reason I chose to target the TFT strategy rather than any other in experiment number three. On Axelrod’s works 20th anniversary, a book was made to see if his work withstood the test of time - [3]. This book showed whilst new strategies have emerged that work better than TFT, Axelrod’s work held up very well against time.

Machine learning in prisoner’s dilemma has also been heavily researched. With varied results, A research article – [4] has very promising results with the trained strategies winning in both standard and noisy tournaments. Results like these are desired, but unlikely in this experiment. There are papers which use Q-learning like – [5] We will be using PPO instead of Q learning, but this paper focuses on characterizing the algorithms, so we can still take valuable information and use it to structure our environment.

Proximal Policy Optimisation (Sub)

Proximal Policy Optimisation or PPO is an algorithm developed by OpenAI – [6] this paper is my primary source for information regarding PPO. OpenAI made PPO to be better than Trust Region Policy Optimisation (TRPO) which means PPO still holds the benefits that TRPO offer but PPO is much simpler to use and implement and better for general purposes.

How it works, going off OpenAI’s paper – [6]. PPO is an algorithm that prioritises the maximum reward over time. It does this by updating the policy parameters that improve the policy but stop it from making massive or drastic changes to it. PPO also uses Mini-Batch Updates, this means that the observations it collects from the environment are split into mini batches this is meant to improve the stability of the training process.

Software (sub)

The software that allowed me to use PPO and create an environment are Unity- [7] and Unity ML agents - [8]. Unity was released in 2005 and is a Game engine, I used this software since it uses C# a language, I was the most familiar with which helped speed up the process of setting up the environment and agent. Unity ML Agents is a package for unity that allows me to set up PPO very easily.

Predictions (sub)

Agent vs Random

The prediction is that the agent will I the end emerge as the Always Defect strategy- [9] meaning it consistently defects. This expectation arises from assuming the opponent will choose defect and cooperate equal amounts. Meaning the Agents optimal strategy would be to always defect since there is no pattern and this will score it higher on average per turn.

Agent vs Agent

If cumulative scoring were the criteria, an Always Cooperate - [9] strategy might be expected. However, Since the AI is rewarded based on individual score. The predictions are that it will take a considerable amount of time to reach Nash Equilibrium, where both agents emerge as the AD strategy. If trained for long enough I do believe there is a chance for AC strategy to emerge, but I would consider this very rare.

Agent vs Tit for tat

I predict that the Agent will quickly reach its Nash equilibrium, emerging as AC. However, a potential flaw in the experiment is that the number of rounds is not randomized but rather set at 50 rounds for a game. This means the agent might remember to Defect on the final round since it remembers that it will be the last and realise its actions will have no consequences -[10]

Method

Software Setup (sub)

The experiment is setup in Unity Engine (version 2022.3.2f1)-[7] and Unity ML Agents(Version 21)-[8], using the Proximal Policy Optimisation (PPO) algorithm. Python3.10.12 –[11] is required for Unity ML agents.

Environment Creation (Sub)

The Unity scene consists of two cubes representing both the agent and opponent, smaller cubes are attached to these indicating cooperation (Green) and defection (red). This setup helps visualise the choices made by both parties during the IPD.

Program Structure (Sub)

Scripts include a manager for results and calculations, an agent’s script for Unity ML agents’ integration, and an opponent script (varies with each experiment). The structure can be seen in fig 3 (UML diagram). The first experiment establishes a baseline by having an opponent that chooses at random. The second experiment will have the opponent be a copy of the current Agents, so they both train simultaneously. The third has the opponent has the TFT algorithm.

[insert uml diagram]

Setting the Parameters (sub)

The environment I create will decide what parameters I give the Agent.

Partially observable -due to the Agent not knowing what its opponent will chose in the future.

Multi agent – Since it has opponents that it will face off against.

Competitive- The agent will be trained to score its highest score rather than highest cumulative score.

Deterministic and stochastic – the first experiment will be stochastic due to its opponent’s randomness but second and third experiments are deterministic.

Static- Since the environment only changes every round.

Discrete- it’s a round based game, so it is all discrete rather than continuous.

Observation and decision making (subsub)

The agent’s observations are represented by a 50-elemtent float array, which contains the opponent’s past choices (0 for cooperate, 1 for defect). For invalid moves are yet to be played rounds a value of -1 is set to maintain partial observability. However, the -1 introduces a potential flaw, this might provide the AI with the extra data required to figure out that number of rounds most likely leading to defect on the final round – [10].

Since the environment is discrete, the agent is restricted to choosing cooperation or defect. The behaviour parameter in Unity ML agents is set to discrete with one branch, meaning that we are requesting a single decision, with a branch size of 2 means the two possible outcomes of that decision being defect or cooperate.

Reward system (subsub)

The agent requires rewards to learn if what it is doing is correct. The agent receives rewards (0,1,2,3) based on its and its opponents’ decisions. These values were decided to maintain clarity and simplicity. The conditions that decide which rewards get assigned when are in Figure 2 which was sourced from Ghislain Fourny - [12].

Measurements (sub)

The agent’s strategy is identified and compared to preexisting strategy to see if they match any. Key strategies I will be trying to identify are the Always Cooperate, Always Defect and TFT.

Method Conclusion (sub)

Ensure that software is correct version and work together, creating a suitable environment, and correctly apply agent parameters and environment observations and rewards. This will lead to an exact copy of the experiments done in this paper, Further you can go to the GitHub link of the actual experiments. –[GitHub link to my repo]

Results

First Experiment: Agent Vs Random(sub)

Experiment one compared the agent against a randomly cooperating or defecting opponent. To mitigate the impact of randomness, this experiment was rerun multiple times. The results, showed in figure [experiment one fig], show that both instances of the test reached the optimal mean score of 2. The slight 0.015 discrepancy in the first test is chalked up to randomness. The first test achieved the desired mean score more quickly than the second test. The mean score of 2 indicates the Always Defect strategy.

A graph with lines and numbers

Description automatically generated

Second Experiment: Agent Vs Agent (Sub)

In the second experiment, the agent faced itself, meaning that both the agent and the opponent would train simultaneously while facing off against each other, both compiling their learning together but being rewarded individually. Initially there was a lot of back and forth between the two agents, at approximately 20,00 steps, the agents started to consistently defect. This outcome signifies the adoption of the Always Defect strategy.

A graph with orange lines

Description automatically generated

Third experiment: Agent Vs Tit for tat(Sub)

In the final experiment, Agent against the TFT strategy, the agent very quickly at around 7,000-8000 steps determined that cooperation yields the best results. Scoring 99 in 50 rounds, averaging 1.98 per round, the agent was nearing the optimal score of 100. The highest possible score is 101, this can only be obtained by defecting in the last round, if this emerged then it would most likely be to a flaw in the experiment. This outcome signifies the adoption of the Always Cooperate strategy.

A graph with numbers and lines

Description automatically generated

Analysis (Sub)

First experiment (Subsub)

The initial experiment, conducted twice to minimize randomness skewing results, aimed to create a baseline for the other experiment, although running it more times would further reduce randomness. The outcomes align with the prediction, reaching the anticipated Nach equilibrium and adopting the Always Defect strategy. Despite a slower than expected start, the experiment is deemed successful and as predicted. This experiment also verified the environment was correctly set up.

Second Experiment (sub sub)

The second experiment involved pitting thee two agents against each other, training them concurrently with individual rewards. In line with predictions both agents exclusively defect, even though it means it reaching only half its optimal score of always cooperating. This in makes sense, since the Agents are rewarded individually which prioritises defecting overall. Always Cooperate strategy I see having a chance of appearing but not being sustainable if the agents kept on training, since defection is a self-reinforcing cycle in this experiment due to it being rewarded and punishing the other agent’s choice of cooperating.

Third Experiment (sub sub)

In the third experiment, the object was to force the agent to adopt the TFT strategy, although this was predicted to be unlikely due to the sole reliance on individual score rewards. TFT typically rewards opponents who always cooperate, not specifically TFT strategy. This means this means it’s more likely to adopt the Always Cooperate strategy which is what happened. The agent ending up adopting the Always Cooperate strategy, suggesting that if I am to force the TFT strategy their must be drastic changes to the environment and opponent, this may be possible through setting up a tournament against various strategies, rather than individual games.

Conclusion

The experiments aligned with my predictions, with the second and third trials reaching the outcomes faster than anticipated. However, the overall success is hindered by the fact that the generated strategies were only Always Cooperate and Always Defect. This falls short of the more desired results being the TFT or any other more complex strategy. The third experiment is deemed a failure as it failed to yield these more complex strategies. This is due to the environment not being properly setup to produce these results. The experiment’s impact on the research in the Prisoner’s dilemma and game theory is negligible die to lack of substantial results.

Future Work

For future iterations of the paper, fixing known experiment flaws in particular the finite number of rounds this would be the priority. Rerunning each experiment multiple times and extending the duration (max training steps) to minimize the chance of randomness skewing results. The third experiment may benefit from a tournament environment, next time using software like the Axelrod python library- [13] instead. Exploring different algorithms such as MAPOCO- [14] which rewards teamwork between agents rather than induvial score. Experimenting with the combination of PPO and MAPOCA could produce interesting results.

Legal social ethical professional

My experiment is about an AI that decides in a dilemma which is a difficult decision between morals, If AI were used in situations like the prisoner’s dilemma that appear in real life, then that creates a plethora of problems one of those legal problems being Responsibility “A Circus tiger can learn tricks from the tamer, but nevertheless stays a predator. If such a creature would hurt or kill the tamer, it is not to blame, as it is a part of its nature.”- [15]. Which translate to its not the fault of the AI for the decision it makes and cannot be the one to blame. The decision for who responsibility relies on is still up for debate. Other legal issue like copyright and data protection an interesting case that has been compared to AI is the “monkey Selfie”-[16]. A lot of AIs require large datasets, these datasets need to be fine tuned or even created by humans to ensure that the dataset is correct. The humans that labour on these datasets are often paid as little as $1.46/hour with “precarious” labour conditions according to an Noema article- [17]. This is also an ethical, social and sustainability since the bad labour and underpaid wage is believed to have an effect on Carbon effects –[18], the cost used to train the AI is also extremely high, For a BERT model estimated at anywhere from $2.5k-$1.6m according to the Cost of training NLP models – [19]. Professional issues I have made sure that adheres to what is stated by the government in regards to Safety and ethics-[20].

My Experiments biggest issue would be relating to ethical, the Agent I have trained is meant to make a choose between a difficult moral decision. There are a lot of ethical issues with getting AI in this field since AI cannot have any emotions or morals but can only mimic them –[21]. The AI will eventually make a decision that will have consequences based on where the AI is making those decisions, extreme cases would be in the use of nuclear weapons if AI is used to determine when to launch a nuclear weapon it could lead to nuclear war, this article goes over a case that can be related to the use of AI – [22].

Usage in Future

My finding is unlikely to have any meaningful influence or application in the future. If my results proved to be significant like an emergence of a superior strategy for the prisoner’s dilemma. This could have led to being implemented in a variety of scenarios, ranging from everyday activities like deciding what grocery assistants to more critical situations like in the justice system as an assistant to a judge, this will alter the way trials are conducted if human involvement becomes unnecessary and the “optimal” decisions is reached without them. However, it is improbable to see these impacts in the foreseeable future.