

Behavior Cloning (BC) of Human Policy via Logged Data

Aashna Kumar, Evelyn Jin, Hooriya Faisal, Samuel Sosa, Tyler Paik

Professor: Zhengyuan Zhou

Mentors: Junyao Chen, Dailin Ji, Ni Yan, Ethan Cao

Abstract

Human decision policy can be learned by machine learning (ML) models using logged data. Our research aims to train a convolutional neural network (CNN) that can predict the next action of a user given the current game state in the snake game. Predicting the user's next action is called behavior cloning. We collected the logged data manually and by heuristics replicating high scoring rounds. The collected data serves as our dataset, consisting of input-output pairs representing the game state and the corresponding actions taken by the human players. After training, our CNN reached an accuracy of 93% on the testing dataset.

Background - Literature Review

Human behavior cloning in games is a technique used in machine learning to teach an agent how to imitate human or expert gameplay behavior. The process involves recording data from skilled human players or predefined expert strategies and then training the AI agent to mimic these actions. By learning from human demonstrations, behavior cloning allows these agents to acquire complex and effective strategies in games, enabling them to perform at a level comparable to experienced players. Furthermore, behavior cloning in general can be applied to complex tasks as well, such as shopping and browser logs.

An example is "User Behavior Cloning for Energy Efficiency in Buildings" by Hong Tianzhen. This project explored the use of human behavior policy cloning to improve energy efficiency in buildings. The researchers collect real-world building data, including occupant behavior patterns from sensors and logs, and then use imitation learning to train models that replicate human energy-saving behavior.

Ultimately, the goal of our research is to teach a computer model to use recorded data to learn and mimic the behavior of expert players. The use of a conventional neural network and fully connected neural network are what we chose to start with in order to find the best set of weights and biases. Once we have built and are on track with these models, we can use them to further extend our research.

Methodology

Before training, we learned related tutorials and collected data. Furthermore, in preparation for the research, we underwent training in essential skills such as training with AWS Cloud, Introduction to Machine Learning, Python & Pandas, Intro to Game AI and Reinforcement Learning, Security and Cryptography, and Version Control. During the data collection phase, each team member played the snake game ten times every day for a week.

In the following weeks, as a team, we conducted the roles of data engineering, modeling, and model evaluation. As data engineers, we connected AWS S3 from Python, extracted all JSON files and concentrated them together, analyzed JSON, extracted all images and directions from the JSON and printed them. In addition, heuristics were utilized to increase the volume of the overall dataset. In terms of modeling, given the input-output dimensions, we implemented a convolution neural network consisting of two convolutional layers, two max pooling layers, 2 fully connected layers, and one softmax layer. Our model evaluators then evaluated the model performance by designing a system that can deploy the trained CNN model to play the snake game on a testing dataset.

Results - Comparative Analysis

Like Tianzhen's project, our main goal was to create an imitation model. One of the key tasks required in order to accomplish this goal were heuristics. Furthermore, We've explored three different ways to guide the computer's learning: Manhattan Distance, Euclidean Distance, and Hamiltonian Cycle. The Euclidean method measures distances as if you were drawing a straight line between two points. The Manhattan distance, on the other hand, calculates the distance in a more grid-like manner, it looks at how many squares you need to move horizontally and vertically to get to a target spot.

The Hamiltonian Cycle method is like planning the most efficient route to reach the fruit without any overlaps or missing spots. It might not always be the quickest way, but it ensures we cover all bases. This strategy is especially helpful when aiming for a comprehensive understanding of human behavior patterns rather than just the fastest results.

This part of the project was like teaching a computer to think like a skilled Snake player by studying how humans play the game and applying heuristics. These heuristics guide the computer to understand what's important in making decisions and help it navigate the game board effectively. Each approach has its unique benefits, and by combining them, we can gain a better grasp of how humans approach decisions in various situations. This kind of learning isn't just limited to games – it can be used to tackle complex challenges like optimizing shipping routes or analyzing browsing patterns. It's all about teaching computers to learn from human expertise and use that knowledge to make smart choices in various situations.

Limitations, Discussion, Future Work

Our project was meant to predict human behavior using a machine learning technique called behavior cloning, and we utilized a Snake game to model that. A few limitations to our research include the limited data and sources we used as well as the amount of time we had allocated to the project. Additionally, the data we used was from other datasets rather than our own for the snake game. Our CNN achieved an accuracy of 93%, however, we could improve our model and data to increase our accuracy. Finally, the last limitation to point out is the lack of using different CNNs to compare and contrast. For future work, we should incorporate various CNNs such as EfficientNet, VGG, and AlexNet to compare the performances and evaluate which should be preferably used. We should also mimic the strategies and actions of players in games based on the predicted action that would be taken and play the game itself without human action.

Conclusion

Our model accurately estimated the next actions in successful rounds with 93% accuracy, indicating that our model is highly effective at mimicking experienced players' gameplay. Through the development of a computer-driven player of the Snake game, we have witnessed the potential of machine learning to adapt, learn, and strategize in real-time scenarios. The successful creation of this player not only underscores the advancements achieved in machine learning but also highlights their applicability in enhancing user experiences within the gaming domain. Ultimately, this project serves as a testament to the possibilities that lie ahead in shaping intelligent gaming systems that push the boundaries of both entertainment and technological innovation.

References

[1] Finding the optimal solution for solving the classic game of snake with pathfinding and heuristics. Harrison Hogg. (n.d.). <https://hogg.io/writings/snake-solution>

[2] Rosebrock, A. (2023, June 8). PyTorch: Training your first Convolutional Neural Network (CNN). PyImageSearch. <https://pyimagesearch.com/2021/07/19/pytorch-training-your-first-convolutional-neural-network-cnn/#:~:text=now%20we%20can-,evaluate,-our%20model%20on>

[3] Learn Python, Data Viz, pandas & more: Tutorials. Kaggle. (n.d.). <https://www.kaggle.com/learn>

[4] Your learning center to build in-demand cloud skills. Self-paced digital training on AWS - AWS Skill Builder. (n.d.).

<https://explore.skillbuilder.aws/learn/course/external/view/elearning/134/aws-cloud-practitioner-essentials>

Tianzhen hong. Tianzhen Hong | Energy Technologies Area. (n.d.).

<https://eta.lbl.gov/people/tianzhen-hong>