AI Self-Driving Car Racing Game

Abstract

This report presents an autonomous driving simulation where a virtual car learns to navigate through a track using an evolutionary approach. The project aims to develop a dynamic learning process that improves the car's ability to avoid collisions and optimize driving behavior through simulation and machine learning techniques. The solution involves evolving neural networks using NEAT (NeuroEvolution of Augmenting Topologies) within a real-time simulation environment created with Pygame.

Introduction

Autonomous driving technology is a rapidly growing field with applications ranging from self-driving cars to intelligent robots. This project explores a simplified model of autonomous driving through a virtual simulation where a car learns to drive itself by adapting to the environment. Instead of manually programming the car's behavior, we employ a machine learning technique that evolves a neural network to control the vehicle. The goal is to create a self-sustaining learning agent capable of mastering complex driving scenarios over time.

Project Objective

The primary objective of this project is to build an autonomous driving simulation that:

- Allows a virtual car to learn to navigate a track independently.
- Utilizes a feedback-driven learning process to improve driving efficiency and safety.
- Demonstrates the potential of machine learning in evolving intelligent behavior without human intervention.

Strategy

Approach

The project follows a structured approach to achieving autonomous navigation:

1. **Environment Creation:** Build a virtual track and vehicle within a simulation environment.
2. **Input Mechanism:** Equip the car with virtual sensors to detect obstacles and measure distances to the track boundaries.
3. **Learning System:** Implement a neural network to interpret sensor data and control the car's movements.
4. **Feedback Loop:** Develop a fitness function that rewards safe and effective driving behaviors.
5. **Iteration:** Allow the system to iterate over many trials to refine its driving strategies.

The Proposed Solution

Technological Components

- **Simulation Environment:** The Pygame library is used to create a real-time visual interface for the simulation, offering a dynamic and interactive platform for testing.
- **Autonomous Driving Logic:** Neural networks serve as the "brain" of the car, translating sensor inputs into driving actions such as turning or adjusting speed.
- **Learning Methodology: NEAT (NeuroEvolution of Augmenting Topologies)**
    - **How NEAT Works:**
        - NEAT simulates natural selection by evolving neural networks over successive generations.
        - The process begins with a diverse population of simple networks, each controlling a car in the simulation.
        - Networks are evaluated using a fitness function that measures their driving performance.
        - The best-performing networks are selected to reproduce, introducing new traits through crossover and mutation.
        - This approach evolves both the neural weights and the architecture, allowing the system to adapt to complex scenarios.
    - **Benefits of Using NEAT:**

        - Does not require labeled training data; learns through experimentation and feedback.
        - Provides flexibility by evolving both the network's topology and its connection weights.
        - Helps avoid overfitting and supports generalization to new driving environments.
- **Evaluation and Feedback:** The fitness function plays a crucial role by scoring cars based on how far they travel without crashing, promoting safe and optimized driving strategies.
- **Iteration and Improvement:** The system undergoes hundreds of generations of evolution, resulting in increasingly intelligent and capable driving agents.

Results

While the project is still in development, preliminary results indicate that:

- The evolved neural networks demonstrate a clear improvement in navigation skills over successive generations.
- The cars begin to avoid obstacles more effectively and optimize their path through the track.
- The NEAT algorithm successfully enhances both the decision-making speed and accuracy of the AI agent.

Conclusion

This project highlights the effectiveness of combining simulation with evolutionary learning to develop autonomous driving behaviors. The NEAT algorithm's ability to evolve neural networks without predefined solutions allows for the emergence of intelligent and adaptable driving strategies. Beyond autonomous vehicles, this project serves as a proof of concept for applying neuroevolution to a wide range of AI-driven systems, showcasing the potential of evolutionary algorithms in solving complex real-world challenges.

References

- Stanley, K. O., & Miikkulainen, R. (2002). Evolving Neural Networks through Augmenting Topologies. Evolutionary Computation, 10(2), 99-127.
- Pygame Documentation. (https://www.pygame.org/docs/)
- NEAT-Python Documentation. (https://neat-python.readthedocs.io/)