How it learnt

How effective

Story

Context

Abstract -> summary

Intro: What you wanted to achieve

Background:

* Include formulas (sq2)
* Simulation – action resarch (sq1)
* Fundamental concepts of MLAs
* How can I train+test

Methodology: what did i do, choice of functions

Results: how did it learn

Discussion:

* How effective
* How did this one specifically learm
* Deeper understanding of functions and why it works
* Comparison (potentially to pezzza’s work?)

My research aimed to discover how a Machine Learning Algorithm (MLA) learns, with the broader goal of applying MLAs to general solutions. To achieve this, I built a 2D physics simulation from scratch in Python, using Pygame for rendering and NumPy for the math. I then integrated an MLA using the neat-python library, designed fitness functions, and ran the model for three days straight. Challenges included the disruptive noise from my PC’s fans and limited expert guidance beyond YouTube videos. **(Effectiveness yet to be analyzed, need to buff up with more words)**

My primary goal was to gain a deeper understanding of how MLAs learn and adapt over time, specifically through applying them to solve a relatively simple task like balancing a pendulum. I also aimed to compare and evaluate different training methods to determine which would most effectively teach the MLA to stabilize the system. Balancing a pendulum was chosen because it is straightforward, fits within the project's time limits, and serves as a fundamental problem that could be expanded to more complex scenarios. Understanding how MLAs learn is vital, as it provides insight into broader machine learning applications, and learning how to implement them directly aligns with my interest in artificial intelligence research. **(Need to buff up with more words)**

The core of the physics simulation relied on the formula for a pendulum attached to a moving cart **(formula)** to model the pendulum’s motion accurately. This formula is crucial because it governs the simulation's mechanics, especially the pendulum's response to external forces. However, a persistent bug caused the pendulum to rotate in an unexpected direction. Performance issues also limited the number of MLA models I could run due to Python’s low multithreading capabilities. The reinforcement learning MLA used trial and error to learn, while a fitness function rewarded successful balance attempts, driving its learning process. Training and testing were essential since this task formed the foundation of my research. **(Need to buff up with more words)**

To implement the MLA, I began by defining the core problem: balancing a pendulum on a moving cart within a physics simulation. I used neat-python for its accessible documentation and ease of integration after trying pyneat, which proved too difficult due to a lack of resources. Pygame was selected as the rendering engine because of its well-documented and user-friendly approach to graphical rendering. My MLA used five inputs; pendulum angle, angular velocity, cart x-position, cart speed, and acceleration. The cart's movement was determined by an outcome formula, where the result was clamped between -1 and 1 to set the cart’s acceleration. I designed a fitness function based on the pendulum’s angle: , rewarding the MLA for balancing the pendulum as long as possible and deducting points when it failed. Training ran for over 400 generations, with 50 models per generation, each model running for a maximum of 10 seconds. The best-performing models evolved into the next generation. Due to performance issues and Python’s limited multithreading, only one model could run at a time, making the entire process last almost 70 hours, with the simulation running non-stop for three days. This led to challenges such as loud fan noise and disrupted sleep. **(Need to buff up with more words maybe?)**

Results: How Did It Learn?

* Outline how the MLA began to learn through repeated simulation runs.
* Explain the reinforcement learning process: how the MLA received rewards for successful balancing and adjusted its decisions based on this feedback.
* Include visualizations or data that show the MLA's progress.

Discussion

* How Effective: Assess how well the MLA performed, mentioning specific metrics (e.g., consistency, stability of the pendulum).
* How Did This One Specifically Learn: Dive into the specific learning patterns observed in your MLA, including any unexpected behaviors.
* Deeper Understanding of Functions: Explain how certain algorithms or functions contributed to the learning process.
* Comparison to Pezzza’s Work: Compare the method and outcomes of your MLA with Pezzza’s approach, noting similarities or differences in effectiveness and learning paths.