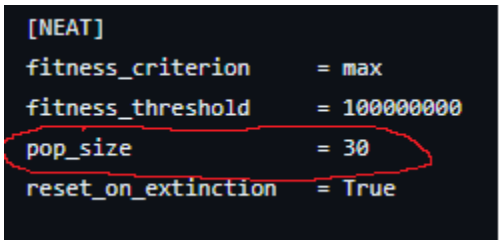
The relationship between iterations in a generation and the success of learning to drive

Introduction

As machine learning becomes more and more valuable to our society, we must research the most efficient ways to train the machine in order for it to learn as fast, but also as accurately as possible. This report aims to test 1 way that we can train a ML algorithm faster - increasing how many chances the algorithm has to get the process correct.

The algorithm used is provided by the youtuber 'cheesy AI'. the map used for this experiment is map2.png. And finally the way that the competition generation was determined was the number next to the generation text in the middle of the track.

To change the number of cars in a generation, change this variable in the config.txt file:



Investigation parameters

Changing the number

Aim

Aim: the aim of this experiment is to see how changing the number of cars in a generation will affect how many generations it will take before the cars complete 1 loop

Hypothesis

It is predicted that increasing the number of cars in a generation will reduce the number of generations needed for a car to complete 1 loop of the track

Risk assessment

Due to this experiment being a digital simulation, there is very little risk associated with it, other than with the hardware dangers.

risk	Management strategy
electrocution	Make sure that computer configuration and hardware is done correctly, including making sure that power supply does not short circuit, damaging you or the hardware components
Fire hazard	Make sure that the hardware has ample cooling capabilities (air/ fluid cooled). Also ensure that the power supply does not overheat due to short circuiting.

Method

1. Open the github file on VS code, or other code running software
2. Set the map variable to the map you desire the experiment to run on (in this case map2.png was used
3. Open config files and set the number of cars in a generation equal to 10
4. Run the simulation until at least 1 car has completed 1 full lap of the track
5. Close the simulation and record the generation that the car was a part of in your results table
6. Run the simulation at least 4 more times
7. Repeat steps 3-6, but increase the number of a cars in a generation by 10 each time

Materials

- VS code or other code running software
- Sufficient hardware to run simulation

Variables

Independent - the number of cars in each generation
Dependent - the number of generations needed for a car to complete the track
Controlled - speed of the cars, map, method of reproduction, hardware simulated with

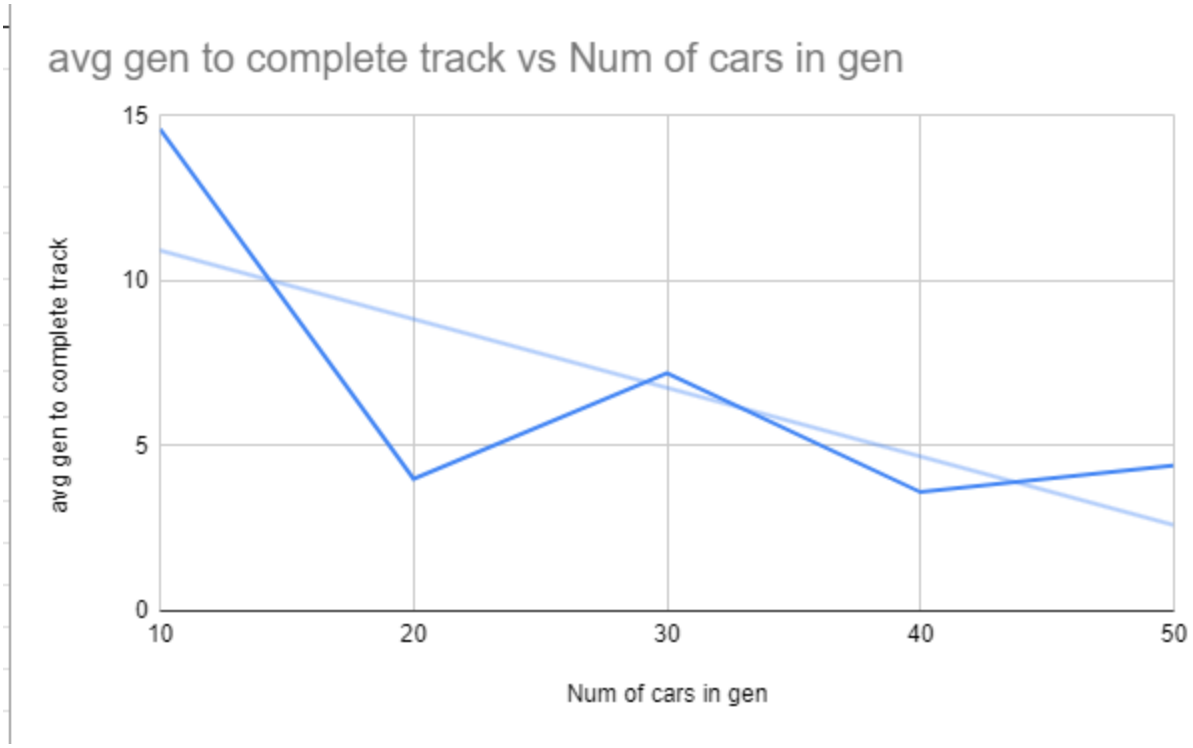
Results - raw

	gen to complete track

Num of cars in gen	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
10	15	3	9	7	39
20	4	2	3	4	7
30	9	14	4	3	6
40	2	5	1	5	5
50	5	10	2	1	4

Results - refined

	gen to complete track					
Num of cars in gen	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	avg
10	15	3	9	7	39	14.6
20	4	2	3	4	7	4
30	9	14	4	3	6	7.2
40	2	5	1	5	5	3.6
50	5	10	2	1	4	4.4



Analysis, extension and improvements

Based on the data above, we can see that there is an overall linear trend between the number of cars in each generation, and the number of generations needed for a ML algorithm to complete 1 loop of the track. Even though the number of generations goes down steadily, there are a number of generations that are outliers from the trend. The most notable example of these outliers is when there are 20 cars in each generation. This change of the independent variable is interesting, as it is the 2nd lowest required generation to complete the track, even beating out when the number of cars is more than double it. There is little explanation for why this outlier exists, but one way that it can be explored is by conducting more than 5 repeats of it, in order to see if the average is brought up, and the data collected was just a fluke.

One extension that could be beneficial to this investigation could be to run this experiment on a larger scale, up to 100+ cars in each generation, in order to see if there is any correlation between factors of the number of cars in a generation, and their success.

One improvement that could be applied to this experiment would be to improve the hardware allocated to running the simulation, which would benefit the cars in the simulation by allowing them to collect more data, and therefore run more efficiently

Bibliography

Shorten, C. (2019). Neuroevolution of Augmenting Topologies (NEAT). YouTube. Available at: https://www.youtube.com/watch?v=b3D8jPmcw-g&t=635s [Accessed 10 Sep. 2023].

Code Bullet (2018). What are Neural Networks || How AIs think. YouTube. Available at: https://www.youtube.com/watch?v=JeVDjExBf7Y&t=3s [Accessed 10 Sep. 2023].

Zero, S. (2021). Visualizing the NEAT Algorithm - 1. Evolution. YouTube. Available at: https://www.youtube.com/watch?v=j8oU0ksQ3Bc&t=2s [Accessed 10 Sep. 2023].

NeuralNine (2021). Self-Driving AI Car Simulation in Python. YouTube. Available at: https://www.youtube.com/watch?v=Cy155O5R1Oo [Accessed 10 Sep. 2023].

Hpe.com. (2023). HPE Machine Learning Development System. [online] Available at: https://www.hpe.com/au/en/hpe-machine-learning-development-system.html?jumpid=ps_55w9fqqn4_aid-520074550&ef_id=CjwKCAjwr_CnBhA0EiwAci5sis9C0D5Uube6LV1Z3KnaC5V0Nqw0cxK-Bv1x2Y-xfumy96zrs6-zjxoCqg4QAvD_BwE:G:s&s_kwcid=AL!13472!3!653472420716!p!!g!!machine%20learning%20system!19918654719!144601127661& [Accessed 10 Sep. 2023].

Matrix Science Team (2020). How to Write a Scientific Report | Step-by-Step Guide. [online] Matrix Education. Available at: https://www.matrix.edu.au/how-to-write-a-scientific-report/ [Accessed 10 Sep. 2023].

Harvard University. (2018). Data Science: Machine Learning | Harvard University. [online] Available at: https://pll.harvard.edu/course/data-science-machine-learning [Accessed 10 Sep. 2023].

Ox.ac.uk. (2023). Department of Computer Science - research theme: Artificial Intelligence and Machine Learning. [online] Available at: https://www.cs.ox.ac.uk/research/ai_ml/ [Accessed 10 Sep. 2023].