Reward Parameter Scientific Investigation

By James Bates

Research Question

How does a change in the reward of the neural network affect the average fitness and best fitness of the population?

Introduction/Parameter

The parameter that will be altered for the purpose of this experiment is the 'reward' present in the function 'get_reward'. When talking about a neural network the reward is one of the most important factors to train a population. By specifying the reward you specify what attributes should be rewarded and therefore what genomes should be passed on to the next generation. By altering the reward you can alter how favourable a specific trait is. In this scenario the reward revolves around the distance that each car has travelled, the higher the distance, the higher the reward. Within the experiment I will be altering the reward to make the distance travelled by each car more highly rewarded. This will incentivise the neural network to reproduce using the genomes of cars which are able to achieve a higher distance within the given generation. By incentivising this, it should lead to a change within the fitness of the population as more and more cars are reproduced to favour distance. To experiment the correlation between fitness and reward, I will change how much the cars are rewarded for distance travelled, using 5 different reward functions.

Variables

Independant

The independent variable (which is what will be changed per iteration of the experiment) is the 'get_reward' function which is present within the 'car' class. More specifically I will be altering the following line of code: return self.distance / (CAR_SIZE_X / a). Where 'CAR_SIZE_X' represents the cars width (being 30 in this situation), 'self.distance' represents the distance that the car travelled (which will be the attribute being rewarded), and 'a' which will change in each iteration of the experiment (representing the independent variable).

Dependant

The dependent variable (which is what will be measured by the experiment) is both the average and best fitness of the population after the simulation has run for 10 generations. This will be measured using the reporter which will print the best fitness and average fitness for each generation. I will then take the given information and place it in a table of values as well as graphs for easy interpretation.

Controlled

The controlled variables are the variables of which will remain constant throughout each iteration of the experiment to ensure fair test conditions in which only a specific variable is measured and changed.

Variable to be Controlled	Why is the Variable Controlled?	How is the Variable Controlled?
Config File	The config file contains many values that alter the performance of the neural network when changed. For this reason it is essential the config file remains identical for each of the experiments, since nothing within the config file is being experimented. A change in the config file could lead to the results being inaccurate and will fail to represent the effect of reward on the neural network's fitness.	The config file will maintain all the same values, being identical to the way it was initially downloaded. It will remain this way for each different trial.
Newcar.py File (Besides Reward Function)	This file defines many factors that could influence the performance of the cars and therefore taint the results of the experiment. Since the experiment is exclusively measuring the fitness of the cars in relation to the reward, all other attributes must remain the same or the experiment will be inaccurate and will fail to represent the effect of reward on the neural network's fitness.	The 'get_reward' function will be the only function within the Newcar.py file that is altered for each experiment. Aside from this, all other functions and lines will remain controlled across each iteration of the experiment.
Record on Generation 10	If the generation that is recorded changes throughout each experiment it will give some models more time to train the population, therefore granting a higher fitness making the test inaccurate in representing the correlation between fitness and reward.	I will use the statistics reporter to gain the average and best fitness for all the models after generation 10 rather than any other generation.

Aim

To determine whether a change in the reward of a neural network affects the ongoing fitness of the neural network.

Hypothesis

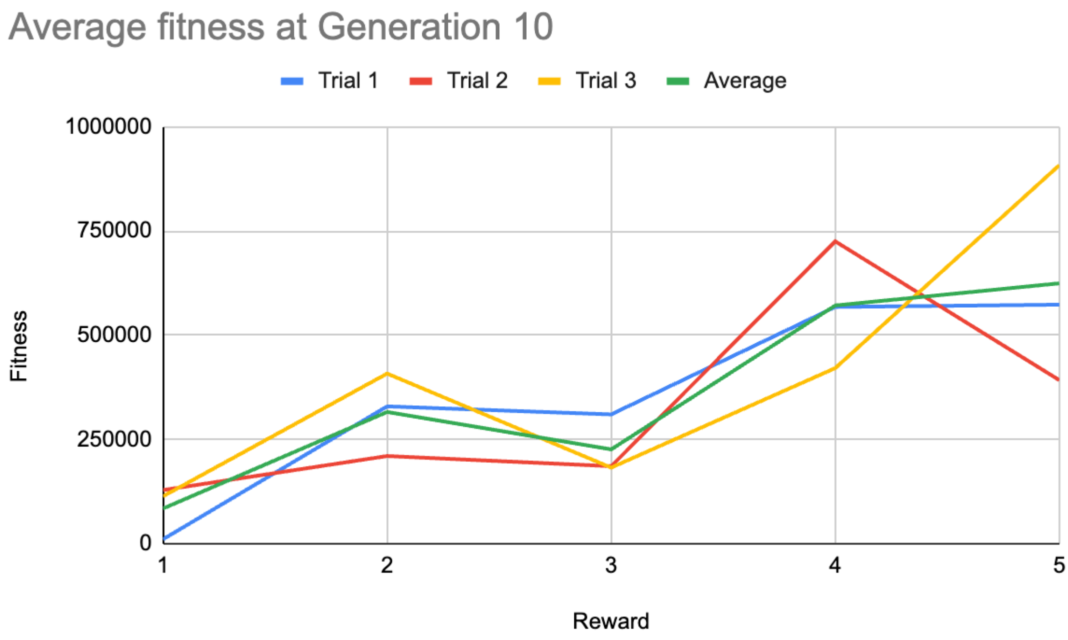
As the reward increases, so will the average and best fitness of the population since distance travelled is more favourable.

Method

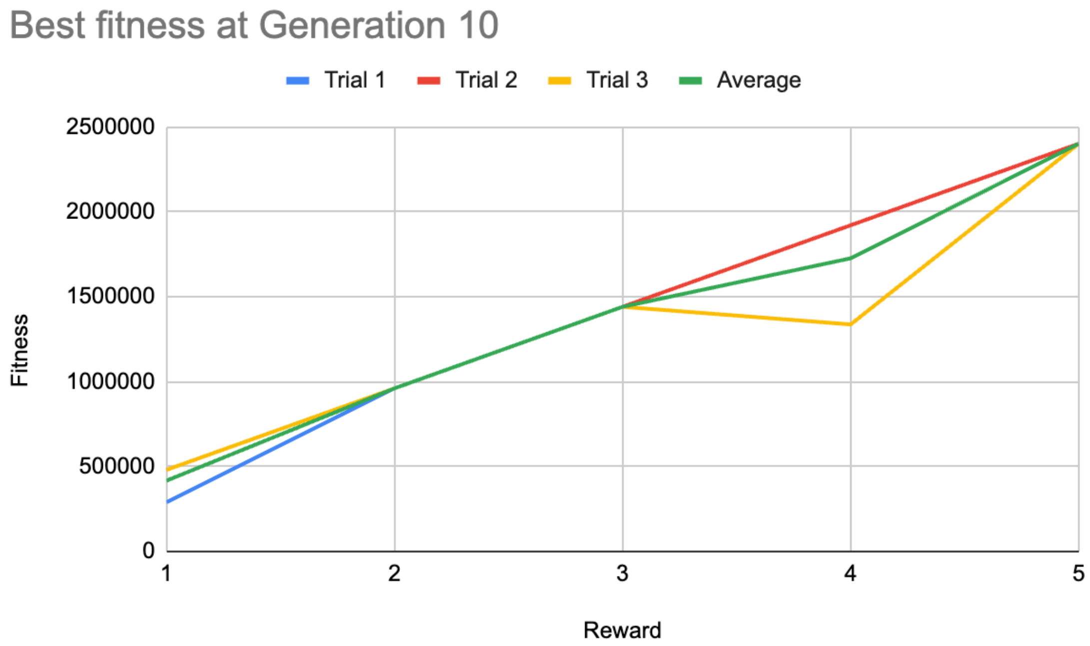
1. Open VScode inside the folder in which the simulation code, config file, and map is kept (ML_Task2).
2. Download the necessary libraries for the code to run using *pip install* followed by each library.
3. Locate the 'get_reward' function present within the 'newcar.py' file and set 'a' to 1 (within the line: *return self.distance / (CAR_SIZE_X / a)).*
4. Enter the 'newcar.py' file and type 'python3 newcar.py' within the terminal to run the simulation.
5. Allow the simulation to run for 10 generations and then hold the 'command' and 'Q' key to exit the simulation.
6. Locate the terminal and find the text reading: 'Population's average fitness:' and take note of the number following the text in a table.
7. Locate the terminal again and find the text reading: 'Best fitness:' and take note of the number following the text in a table
8. Repeat steps 4-7 twice more
9. Add 1 to the value of 'a' in step 3
10. repeat steps 3-9 for a total of 5 times

Results

Reward	Average Fitness at Generation 10			
	Trial 1	**Trial 2**	**Trial 3**	**Average**
1	10612.74444	128669.75556	113715.47556	84332.65852
2	329378.39556	210226.65333	408058.19481	315887.7479
3	310107.30667	185948.61333	182523.55333	226193.1578
4	567947.14667	725237.67111	421189.16444	571457.9941
5	573367.7	392279.17778	908094.33333	624580.4037



Reward	Best Fitness at Generation 10			
	Trial 1	Trial 2	Trial 3	Average
1	289677.73333	480400	480400	416825.9111
2	960800	960800	960800	960800
3	1441200	1441200	1441200	1441200
4	1921600	1921600	1337323.73333	1726841
5	2402000	2402000	2402000	2402000



Analysis

As hypothesised, the higher the reward, the higher the average and best fitness. While this is a clear trend present in the majority of the collected data, there is an anomaly that goes against the hypothesis. In the table and graph for average fitness we can see that the average fitness drops when the reward is 3 compared to when the reward is 2. This is likely a result of the neural network having a certain level of randomness to it, which I attempted to counteract by doing three trials of the same reward. Despite this anomaly the rest of the data still supports my hypothesis, with the average and best fitness of each population rising as the reward is increased.

Another strange trend within the data is observed when looking at the best fitness table and graph. As seen, when the reward rises so does the best fitness, however when the reward is the same the best fitness also stays the same throughout each trial on the majority of the tests. This is seen especially when the reward was equal to 2,3, and 5, with each trial ending with the same best fitness. This is very unusual and while there is a chance that the statistics are incorrect in some way (although unlikely), there could be other reasons for the unusual trends in data. This could be the result of a persistent genome appearing throughout the trials, having found the most efficient path for the given reward value by generation 10. This could also explain why this trend isn't seen when the reward is equal to 1 and 4 with a certain amount of chance being required for the same best fitness to appear.

Conclusion

In conclusion the experiment went quite smoothly and the data gained supported the hypothesis, showing the correlation between a higher reward and a higher fitness level. An example of this can be seen from the results of the average fitness of the three trials when the reward is equal to 1 vs the average fitness of the three trials when the reward is equal to 5, by a factor of approximately 7.4.

References

Connor Shorten. (2019). *Neuroevolution of Augmenting Topologies (NEAT)*. *YouTube*. Retrieved September 7, 2023, from https://www.youtube.com/watch?v=b3D8jPmcw-g&t=635s.

Macaulay, T. (2022, March 2). *Reinforcement learning: How rewards create intelligent machines*. TNW | Deep-Tech. https://thenextweb.com/news/how-rewards-work-in-reinforcement-learning-deepmind#:~:text=In%20AI%20systems%2C%20the%20rewards,agent%20to%20evaluate%20its%20performance.

NeuralNine. (2021). *Self-Driving AI Car Simulation in Python*. *YouTube*. Retrieved September 7, 2023, from https://www.youtube.com/watch?v=Cy155O5R1Oo.

Zampogna, M. (2023). *Jetscholar/ML_TASK2*. GitHub. https://github.com/jetscholar/ML_Task2/tree/main

Adibhatla, B. (2019, February 19). *Understanding the role of reward functions in reinforcement learning*. Analytics India Magazine. https://analyticsindiamag.com/understanding-the-role-of-reward-functions-in-reinforcement-learning/#:~:text=This%20is%20known%20as%20a,determines%20the%20rewards%20for%20actions.