# Relationship between population size and the average final fitness score

**Research Question:**
How will changing the population size of each generation affect the change in rate of fitness between each generation?

**Background**
This scientific report is about running simulations on a neural network that controls cars around a racetrack. It'll be done within Python using NEAT as its algorithm. Each car begins with randomly generated nodes and is assigned a reward (called fitness) based on the distance it has travelled. Each generation of the simulation runs for a maximum of 1200 ticks and will end when either the timer has run out or every car has died by colliding with a wall.
Once a generation has ended a new one will begin with new cars that contain nodes that have achieved high fitness scores from the previous generation as well as some randomly generated mutations.
Population size is the amount of cars that are used in each generation and increasing or decreasing this number will affect the amount of mutations that can occur.
Average fitness score is how well the average car has done in a specific generation

Within this scientific experiment, we will see how changing population size will affect the average fitness score. This is to determine whether or not you can continuously increase the population size to get increasingly better cars and if there is whether or not there'll be diminishing returns the higher you go up. This experiment will also allow us to see any other patterns that might help increase efficiency within simulating virtual cars and help draw conclusions.

TO AVOID CONFUSION:
Since the measured variable is called "average fitness score" and within graphs averages for that variable are calculated instead of calling the average of the collected data "average average fitness score" it will simply be referred to as average fitness and the singular word "average" will be referring to the calculated averages.

## Variables:

**Independent Variable**

Population Size

**Dependent Variable**

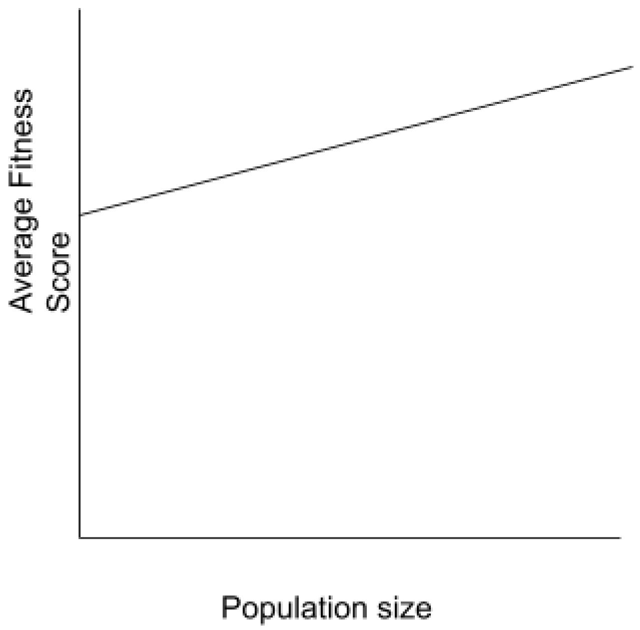
Average Fitness Score

| Controlled Variable | Why does it need to be controlled? |
|---|---|
| Map | Each map has an individual difficulty level and if maps vary then different fitness scores will be assigned for the same car |
| Number of nodes | Differing amounts of nodes could prove to be beneficial for some population sizes than others due to nodes being complicated and differing population sizes being able to handle the complexity at different rates because of the amount of mutations happening |
| Max time per generation | Different max times could allow for some populations to create more careful cars that could travel great distances whilst others would have to create much faster cars that crash often due to less time |
| Number of generations | Additional generations naturally means more time to evolve and thus better fitness scores for those who got additional time |
| Stats of cars | There are specific starting statistics that are optimal for a car to get the best fitness score and this would skew the results in favour of the generations who got better stats |
| Rewards calculation | If we calculate each cars fitness score differently this would invalidate the entire experiment as even with enough experiments to cancel out randomness, we would no longer be measuring the specified value. |

**Hypothesis:**

If the population size of a given simulation increases then there'll be a higher average fitness score due to more mutations occurring and therefore more chances to receive positive changes within a population allowing for higher overall fitness

Prediction Graph: Population Size vs Final fitness score




**Equipment / Materials:**

- 1 Computer
- Files downloaded from [https://github.com/jetscholar/ML_Task2](https://github.com/jetscholar/ML_Task2)
- A program capable of running python
- The following libraries installed onto your program
  - math
  - random
  - sys
  - os
  - neat
  - pygame

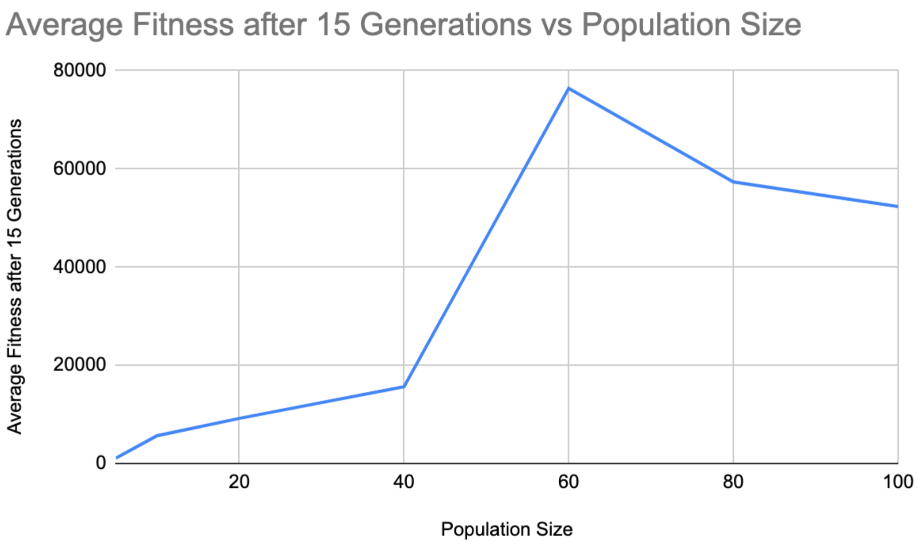
**Method**

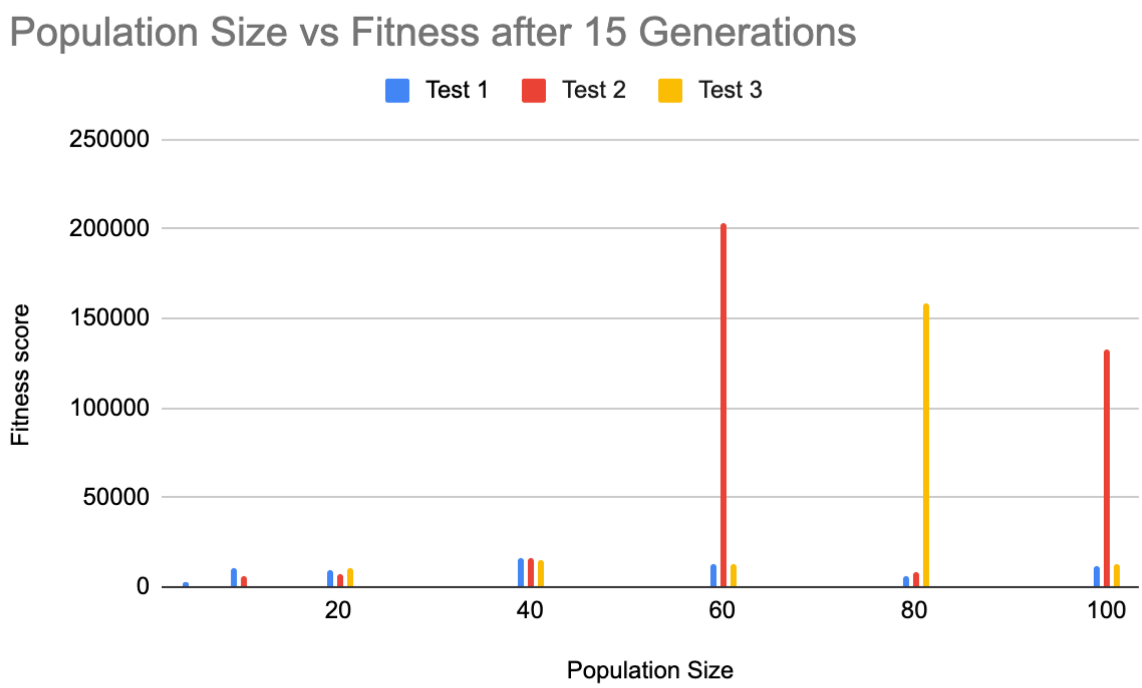
1. Open the downloaded config.txt file
2. Change "pop_size            = 30" to "pop_size            = 5"
3. Save the file
4. Run your Python file through your program
5. Once the simulation has reached 15 generations, record the average fitness at the start of generation 16
6. Repeat steps 4 and 5 3 times
7. Repeat steps 2-6 seven times and each time change pop_size to 10, 20, 40, 60, 80, and 100

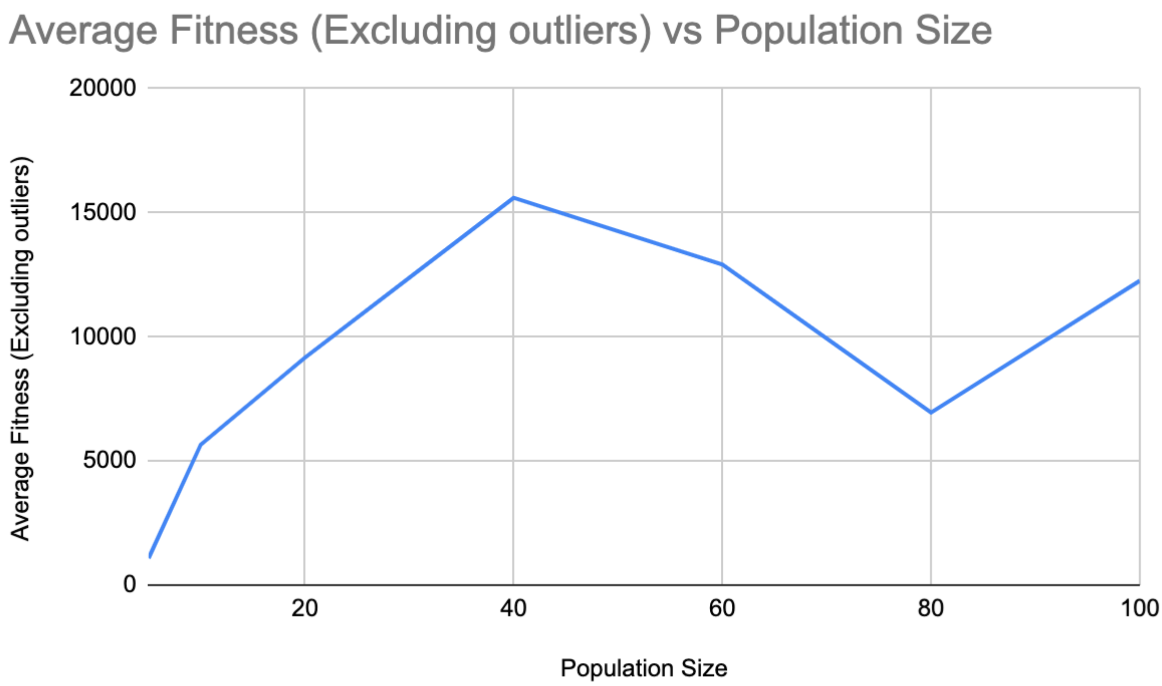
**Results**

| | Fitness after 15 generations | | | |
|---|---|---|---|---|
| Population Size | Test 1 | Test 2 | Test 3 | Average |
| 5 | 2,663.46667 | 277.82667 | 309.18667 | 1,083.493337 |
| 10 | 10,888.4733 | 5,861.61333 | 200.52667 | 5,650.204433 |
| 20 | 9,882.91667 | 7,527.21000 | 10,093.39667 | 9,167.841113 |
| 40 | 15,650.04833 | 16,636.59000 | 14,530.57949 | 15,605.73927 |
| 60 | 12,570.30111 | 203,408.24000 | 13,245.69556 | 76,408.07889 |
| 80 | 5,606.64774 | 8,287.21688 | 158,162.78917 | 57,352.21793 |
| 100 | 11,966.02376 | 132,511.97400 | 12,551.60726 | 52,343.20167 |

**Graphs:**






**Calculations:**

$$\frac{\text{Test 1} + \text{Test 2} + \text{Test 3}}{3} = \text{Average Fitness}$$

**Analysis of data**
The first graph containing the average fitness scores clearly shows that fitness scores up to a population of 40 are low, however, a big jump occurs at population 60 followed by a slow decrease through population sizes of 80 and 100. This does not support my hypothesis due to the fitness score not continuously increasing with population size but rather having an optimal value being around 60.

The second graph highlights a flaw within the first graph. When plotting Fitness scores to Population size without calculating averages 3 outliers are displayed within the dataset. These three outliers all show how different test runs with the same population size can differ greatly depending on luck. This graph does still show value by indirectly supporting my hypothesis due to the test runs containing fitness scores in the hundreds of thousands only existing in the 3 highest population values. This supports my hypothesis due to the reasoning that since there are more cars there is more of a chance for really good genes to be developed and quickly spread amongst the population. This would then in turn rapidly increase the average fitness score. The bell curve still exists and illustrates the fact that 60 has the highest average fitness score and each subsequent highest value afterwards descends.

The third graph contains data on all fitness scores under 100,000. Whilst unscientific to exclude data simply because it doesn't fit, there is value in looking at this graph. It is valuable to look at since it supports the trends of the first graph. It contains a peak which is optimal population size and dips on either side. It does differ from the first graph because it dramatically decreases average fitness as the population size goes past 40 with 80 population size having a lower average fitness than 20. Another value in this graph is the fact that it appears to dip back up again at 100. This could mean that there is another more optimal population size past 100 which could be looked at within further research.

## Conclusion of data
Overall the data displays slightly varying trends depending on how you look at it, however, the common trend is that both lower and higher population sizes have lower average fitness scores than the middle values. This creates a bell curve that displays an optimal population size being in the middle. This bell curve is skewed towards the right as all the higher population sizes contain test runs with average fitness scores above 100,000. This bell curve could be explained as either appearing through chance or a greater population actually inhibiting a generation's ability to quickly spread new genes as there is a greater amount of cars a good gene needs to pass through to increase the average fitness score.
It is important to note that this scientific experiment was done with a small sample size and clear faults within this sample size are displayed. As with all experiments, these results are not conclusive and need to be verified through more test runs as well as other people experimenting.

## Evaluation of Hypothesis
The hypothesis stated, "If the population size of a given simulation increases then there'll be a higher average fitness score due to more mutations occurring and therefore more chances to receive positive changes within a population allowing for higher overall fitness". This was not supported by the data. Overall the data did not display an upward trend in terms of population size to average fitness score instead it showed a bell curve with the highest average fitness scores appearing in the middle. Nonetheless, the reasoning for my hypothesis was somewhat supported by this data because population sizes 5-60 showed an upward trend and the dip within populations 80 and 100 could potentially be explained by higher populations being unable to spread good genes quickly.

**Evaluation of Method:**
The method used here was flawed only regarding numerical values needing to be different. This experiment was done digitally making it easy to control different variables and consistently run test simulations. Overall the methodology of the method was not flawed and was able to adhere to the scientific method.

Regarding the numerical values that could've been changed to allow for a better method. The first is adding more test simulations would've allowed for better control over the inherent randomness of neural networks and allowed for a better view as to how much the outliers affected the results as if they appear more often. The second adjustment would've been to increase the amount of generations the simulation is run for. As discussed there could've been a delay in spreading good genes within a larger population so adding additional time to allow these genes to spread could've greatly affected the results and potentially making them align with the hypothesis. The final adjustment would have been running experiments for up to 200 population sizes or greater. As there was potentially another upward trend seen within graph 3 running further experiments to verify this fact would allow for better conclusions to be made.

**Bibliography**

www.youtube.com. (6/10/17). But what is a Neural Network? | Deep learning, chapter 1. [online] Available at: https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi.

Zampogna, M. (2023). Machine Learning Task 2. [online] GitHub. Available at: https://github.com/jetscholar/ML_Task2 [Accessed 17 Sep. 2023].

www.whiletrue.it. (n.d.). HTML5 Genetic Algorithm 2D 3-Wheelers - WhileTrue.it. [online] Available at: https://www.whiletrue.it/genetic_3-wheelers/ [Accessed 14 Sep. 2023].

www.youtube.com. (15/3/18). *11.2: Neuroevolution: Crossover and Mutation - The Nature of Code*. [online] Available at: https://www.youtube.com/watch?v=kCx2DEIEpP8 [Accessed 14 Sep. 2023].

*Neuroevolution of Augmenting Topologies (NEAT)* n.d., www.youtube.com, viewed 14 September 2023, <https://www.youtube.com/watch?v=b3D8jPmcw-g&t=635s>.

www.youtube.com. (23/9/21). *Visualizing the NEAT Algorithm - 1. Evolution*. [online] Available at: https://www.youtube.com/watch?v=j8oU0ksQ3Bc&t=2s.

Baheti, P. (2022). *12 Types of Neural Networks Activation Functions: How to Choose?* [online] www.v7labs.com. Available at: https://www.v7labs.com/blog/neural-networks-activation-functions.

www.youtube.com. (2023). *Get Started in Pygame in 10 minutes!* [online] Available at: https://www.youtube.com/watch?v=y9VG3Pztok8 [Accessed 14 Sep. 2023].