The relationship between the number of sensory radars on a car and a population's best fitness

Date: 12/9/2023

Research question:

How does the number of radar's affect a population's best fitness, across three different maps in increasing difficulty in the newcar.py file?

Background information

The newcar.py file is a genetic algorithm python file that simulates the genetic evolution of a population of cars with neural networks (the inputs from the car radars) as it aims to travel the furthest distance around a certain map. Each car is a neural network, with each input neuron being the trigonometric distance between the ends of one of the radars to the centre of the car. Each generation of the simulation, each car in the population is mutated to a certain degree, whilst keeping the best model from the previous generation in the new generation. The effectiveness of each model is determined by its fitness, a function in the code that measures whether the car crashed or not, as well as the distance travelled by the car.

In terms of the mechanisms of evolution, the population first determines the surroundings of the car using the distance of radars to calculate spatial information the neural network can work with. This information, a.k.a the input or first layer of neurons, is then connected through weightings of synapses, to a number n (0) layers of hidden nodes, eventually coming up with an output neuron, either turning the car left, right, decelerating, or if it is uncertain, accelerate. After the each generation, the neural network is assigned a fitness variable, determine whether the output was successful and avoided a crash or increased distance. The most successful neural network models synapse weightings are then mimicked throughout the population of the next generation, creating a artificially selective environment. Initially the synapse weightings are random, however as the more successful neural networks survive longer and have their synapse weightings distributed into the population, thus evolving the population.

Map 1 contains 2 turns, and longer straights. Map 2 contains a chicane, 8 turns, and shorter straights. Map 4 contains 17 turns, and the shortest straights.

Variables

Independent Variable: The number of radars (integer) - This will be manipulated increasing the number of radars in the radar array, as well as decreasing the angle between each radar.
Dependent Variable: The time it stays swinging (seconds) - This will be measured as the final best fitness of the population, after 25 generations.
Control Variables:

Name the variable that you need to keep the same	Explain why you need to keep this variable the same	How will you keep this variable the same?
Map	To ensure each population adapts to the same turns and straights, as well as the same level of difficulty.	Simulate and compare tests on the same map
Car dimensions	To ensure that the car's dimensions are the same, i.e, corner lengths, variables, centre position, etc.	Keep the code in Car() the same
Car functionality	Keeps all of the decision making between each population the same.	Keep the code in the functionality the same.
Population size	Ensure that there is a fair number of diverse car populations.	Population remains constant in config.txt file at 35

Hypothesis

As the number of radars increases, the population's best fitness will increase at an increasing rate, as more radars increase the depth of inputs and senses the neural network is given.

Method

1. Decide on a number of radars, must be an odd number where the number of radars is larger than 2, and 180 is divisible by two subtracted from the number of radars.
2. For the number of radars, add a 0 to the `return_values` array
3. Change the `num_inputs` value in config.txt to the number of radars
4. Calculate the interior angle between each radar, according to the equation below
5. Change the value of the angle (third parameter) in the line `for d in range(-90, 120, 90):`
6. Change the value of the angle (the divisor) in the line `return_values[i] = int(radar[1] / 90)`
7. Run the simulation for 25 generations.
8. Repeat steps 1-7 trial for maps the other 2 maps excluding map 3 and 5
9. Repeat steps 1-8 two more times for reliability.

$$Interior\ Angle\ = \frac{180}{Number\ of\ radars - 2}$$

Data Collection and Processing:

Table 1: Effect of number of radars on best fitness on map.png

Number of radars	Best Fitness ± 1x10^-5			Average Best Fitness ± 1x10^-5	Standard Deviation (s) ± 1x10^-5
	Trial 1	Trial 2	Trial 3		
3	205552.92000	195650.56000	306792.44000	235998.64	50221.74677
5	602144.80000	600072.56000	687618.72000	629945.36	40789.99783
7	320772.76000	776543.16000	518339.32000	538551.7467	186615.596
11	506212.88000	587921.00000	647728.64000	580620.84	58003.71732

Table 2: Effect of number of radars on best fitness on map2.png

Number of radars	Best Fitness ± 1x10^-5			Average Best Fitness ± 1x10^-5	Standard Deviation (s) ± 1x10^-5
	Trial 1	Trial 2	Trial 3		
3	13001.12000	13001.12000	945.20000	8982.48	5683.21519
5	264325.16000	187440.98000	178394.68000	210053.6067	38553.08039
7	202426.76000	288240.00000	288240.00000	259635.5867	40452.74928
11	288240.00000	264325.16000	288240.00000	280268.3867	11273.56369

Table 3: Effect of number of radars on best fitness on map4.png

C

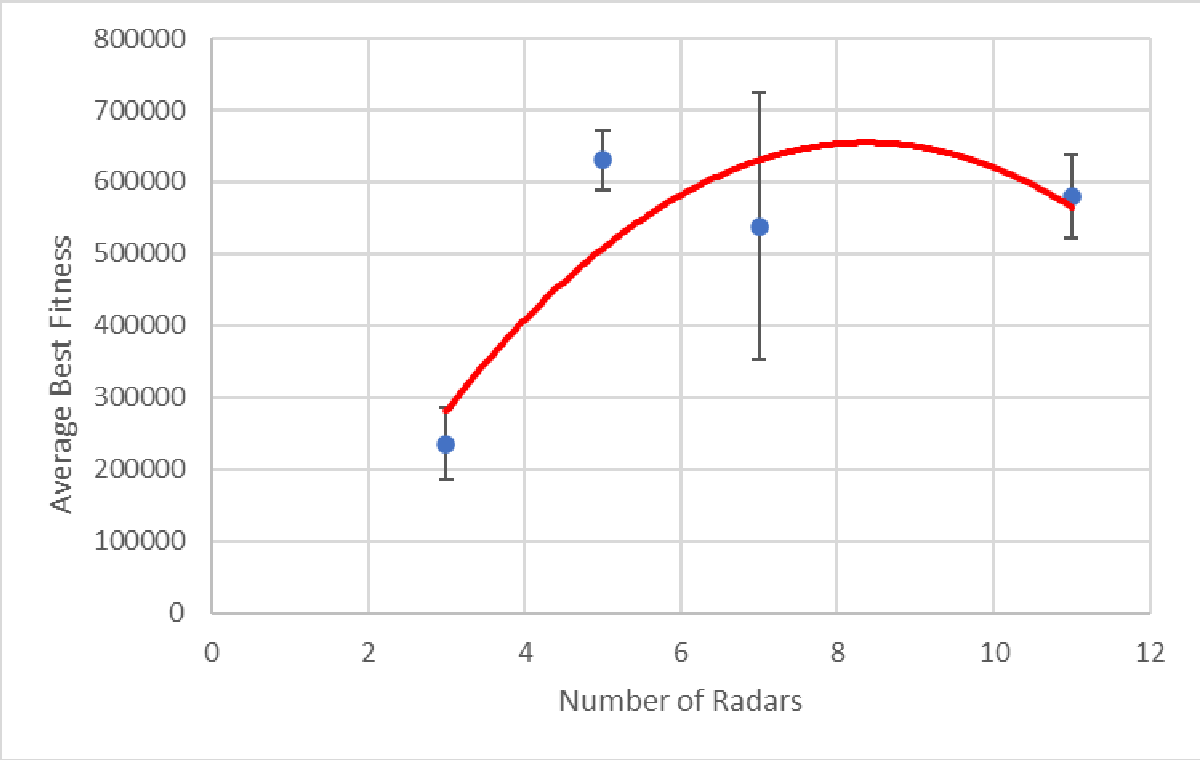
Number of radars	Best Fitness ± 1x10^-5			Average Best Fitness ± 1x10^-5	Standard Deviation (s) ± 1x10^-5
	Trial 1	Trial 2	Trial 3		
3	391.20000	391.20000	421.24000	401.2133333	14.1609918
5	35788.64000	35272.40000	35402.64000	35530.52	258.12
7	174045.04000	177218.48000	176280.80000	175848.1067	1331.189443
11	174577.52000	178305.76000	178305.76000	177063.0133	1757.509191

Sample Calculations:

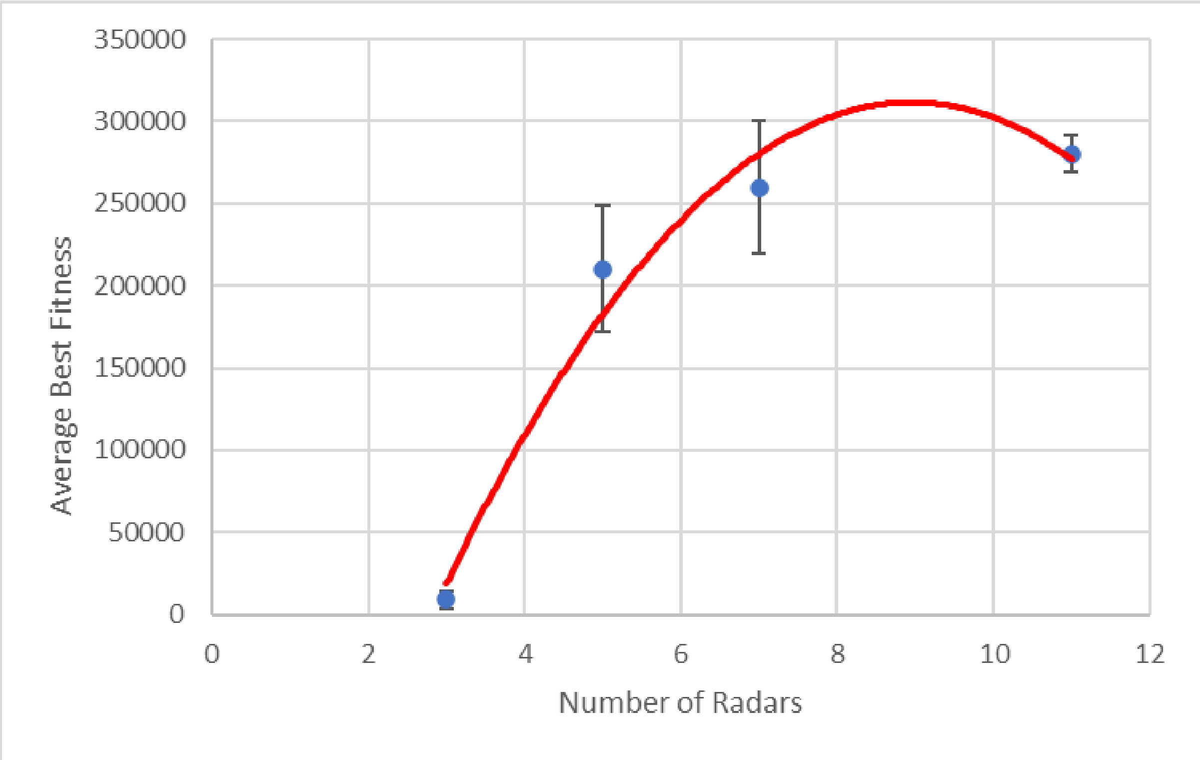
$$Average = \frac{Trial\ 1 + Trial\ 2 + Trial\ 3}{3} = \frac{205552.92000+195650.56000+306792.44000}{3} = 235998.64$$

$$Standard\ Deviation = \sqrt{\frac{\Sigma(x_i+\mu)^2}{N}} = \sqrt{\frac{(205552.92000-235998.64)^2+(195650.56000-235998.64)^2+(306792.44000-235998.64)^2}{3}} = 50221.74677$$

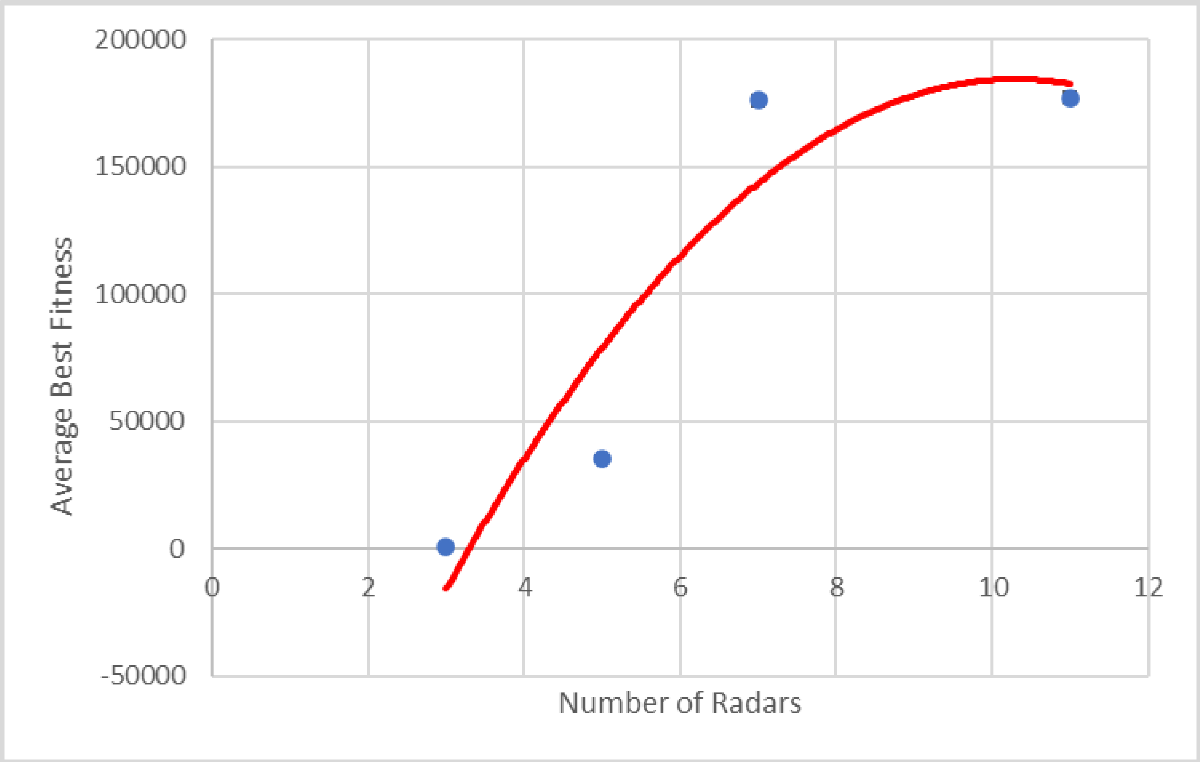
Graph 1: Effect of number of radars on best fitness on map.png



Graph 2: Effect of number of radars on best fitness on map2.png



Graph 3: Effect of number of radars on best fitness on map4.png



Interpretation and Analysis of Data:

Graph 1: On map 1, the easiest map, the best fitness increased as the number of radars increased, then after reaching 7 radars, began to drop. The number of radars producing the highest best population fitness was 5. The error remained relatively small, however there was a particularly significant standard deviation at 7 radars. The highest average best fitness was at 5 radars.

Graph 2: On map 2, the next most difficult map, the relationship between the number of radars and the population's best fitness was similar to graph 1. However there was a slightly smaller error rate.

Graph 3: On map 4, the most difficult map, the data followed a similar trend, where the population fitness grew at a rapid rate from a low number of radars, the rate then decreasing and flattening out between 7 and 11 radars, but still increasing. The error was miniscule.

Overall, as the difficulty increased, the trend approached closer to a relationship increasing the most between 3 and 7 radars, and then increasing at a slower rate between 7 and 11 radars.

Evaluation

Evaluation of Hypothesis

On map 1, the hypothesis was partially correct; the population's best fitness did increase rapidly as the number of radars increase, though began to decrease at the higher numbers of radars, though the rate of which trended was not at a similar magnitude, there is a maximum threshold for the number of radars increasing the population's best fitness. There was quite a large error range, therefore this data is not as reliable.

On map 2, the hypothesis was still partially correct, similar to map 1, and approaching map 3, though with an increased reliability in comparison to map 1 due to its lower error rate.

On map 3, the hypothesis was partially correct; the population's best fitness did increase as the number of radars increased, and continued to increase at the higher numbers of radars. However, the factor of which best population fitness increased decreased rapidly. The error range was extremely small, indicating high data reliability.

Overall, the hypothesis was consistently partially supported. As the depth of sensory inputs increased, the population's best fitness increased, as well, initially increasing at a rapid rate, before still increasing but at a lower rate.

Evaluation of Method

A factor that could be taken into account for the high uncertainty is the acceleration determiner. Because the code accelerates the car only when the neural network is uncertain whether to turn left, turn right, or decelerate, there is a higher randomness in maps where there is more opportunity for error.

Conclusion

In conclusion, regardless of the map, as the number of radars increases, the population's best fitness increases as well, initially at a high rate, then at a decreasing rate. This could potentially be because the car's code accelerates when their neural network is uncertain. As the maps become narrower and with more turns, there are also less opportunities for the car to accelerate, meaning the car's ability to take turns is more important. In map 1, the uncertainty was extremely high, though as the map difficulty increased, the uncertainty decreased as well.

Works Cited

3Blue1Brown. (2017, October 6). But what is a neural network? | Chapter 1, Deep learning [Video]. YouTube. https://youtu.be/aircAruvnKk

3Blue1Brown. (2017, October 17). Gradient descent, how neural networks learn | Chapter 2, Deep learning [Video]. YouTube. https://youtu.be/IHZwWFHWa-w

3Blue1Brown. (2017, October 17). What is backpropagation really doing? | Chapter 3, Deep learning [Video]. YouTube. https://youtu.be/Ilg3gGewQ5U