**Experimental Report**

**Code Analysis - Genetic Algorithms**

**Subject: IT1002**

**Name:** William Chapman **Teacher**: Mr. Zamponga

Contents

[1.0 Research Question 3](#_Toc145327724)

[2.0 Aim 3](#_Toc145327725)

[3.0 Background 3](#_Toc145327726)

[4.0 Variables 3](#_Toc145327727)

[4.1 Independent Variables 3](#_Toc145327728)

[4.2 Dependent Variables 3](#_Toc145327729)

[4.3 Constant Variables 4](#_Toc145327730)

[4.4 Control group 4](#_Toc145327731)

[5.0 Hypothesis 4](#_Toc145327732)

[6.0 Materials and Equipment 4](#_Toc145327733)

[7.0 Method 7](#_Toc145327734)

[7.1 Initial Steps 7](#_Toc145327735)

[7.2 Repeated Steps 7](#_Toc145327736)

[8.0 Results and Findings 7](#_Toc145327737)

[8.1 Results table 7](#_Toc145327738)

[8.2 Graphs 8](#_Toc145327739)

[9.0 Conclusion 8](#_Toc145327740)

[10.0 References 8](#_Toc145327741)

# Research Question

To understand which of the 5 different activation function effects the fitness of the program the best and gives the program the best result.

# 2.0 Aim

To explore whether the input of 5 different activation functions (sin, cube, inv, gauss and hat) affect the fitness of the program. This will be achieved through examining the change in fitness of the program after running the program for 30 generations.

# 3.0 Background

The research question is related to the built-in activation functions and its relationship to the training dynamics specifically with the fitness of the program. The activation functions determine the output of a neuron in a neural network and enabling the network to learn complex patterns from the data. Neural networks resemble those present in living organisms, particularly the human brain. Neuron activity depends on how strong the impulses arriving from nearby brain areas are, like the activation function in artificial neural networks, this process works. Neural networks rely on these components - weight, bias, and activation function to operate correctly. To improve accuracy in a neural network, we adjust the connections between neurons based on errors at the outputs (Geeksforgeeks,2023, February 17). Back-propagation refers to working back from output nodes to input nodes finding and fixing errors. The activation functions make back-propagation possible (Geeksforgeeks,2023, January 5).

# 4.0 Variables

## 4.1 Independent Variables

An Independent variable is the variable that is being changed throughout the experiment (Pearson Australia, 2017, p.4) Below is the independent variables identified for this experiment:

1. Activation functions (sin, cube, inv, gauss and hat)

In this experiment the changes being made will be testing each activation function, changing each function.

## 4.2 Dependent Variables

A dependent variable is a variable whose value depends upon an independent variable. The dependent variable is what is being measured in this experiment. (Pearson Australia, 2017, p.4). Below is the dependant variable identified for this experiment:

1. The Fitness of the program

## 4.3 Constant Variables

Table 2 provides the proposed variables of the experiment that needs to be controlled as constant variable.

|  |  |  |
| --- | --- | --- |
| **Constant Variable** | **Explanation** | **Control Measure**  **(How do you keep the variable constant)** |
| 30 generations for each test | This variable needs to be controlled as having each test with different generations reduces the reliability and validity of the experiment. | Running each test for 30 generations and stopping the program at 30 generations. |
| The map | This variable needs to be controlled as it greatly effects how the program learns | Use only one map |

*Table 1 - Constant Variables for Experiment*

## 4.4 Control group

A control group is an experimental condition that does not receive the actual treatment and may serve as a baseline (The Pennsylvania state university, 2023). Below is the control group identified for this experiment:

1. The original program with the original activation function

# 5.0 Hypothesis

If the 5 activation functions that are run for 30 generations each in the program, then the one or two of the activation functions will demonstrate a statistically higher improvement in fitness, because the functions will be able to update the weights and biases based on the error outputs.

# 6.0 Materials and Equipment

|  |  |  |
| --- | --- | --- |
| **Description** | **Quantity** | **Picture** |
| List of activation functions | 1 |  |
| Program | 1 |  |
| Config.txt | 1 |  |

*Table 2 - Materials and Equipment*

# 7.0 Method

## 7.1 Initial Steps

1. Open the program

2. Open the config.txt

3. Find where the activation functions are, so that the activation functions can be changed

4. Pick a map

5. Run the control group (original version of the program) for 30 generations then stop the program

6. Record the results into a table

## 7.2 Repeated Steps

7. Open Config.txt

8. Find where to change that activation function and change the function to the activation function ‘cube’

9. Once the activation function is changed then save the file

10. Use the side navigation bar to go back to the program

11. Run the program for another 30 generations then stop the program after the 30 generations

12. Once 30 generations are reached press (alt-f4 for windows and command Q) to close the program

13. Record the results into a table

14. Repeat the experiment from steps 6 – 12 for sin, inv, gauss and hat

15. Repeat each test for 3 times and calculate the average fitness of the tests

# 8.0 Results and Findings

## 8.1 Results table

Table 3 records the fitness measurements for when the program is run for 30 generations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activation Functions | Adjusted Fitness | | | Avg. |
| Cube | 0.240 | 0.309 | 0.264 | 0.271 |
| Sin | 0.177 | 0.194 | 0.187 | 0.186 |
| Inv | 0.131 | 0.137 | 0.154 | 0.140 |
| Gauss | 0.123 | 0.057 | 0.049 | 0.076 |
| Hat | 0.075 | 0.081 | 0.084 | 0.080 |
| Control Group (lanh) | 0.610 | 0.405 | 0.210 | 0.408 |

*Table 3 – Activation function fitness measurements*

The data from table 3 shows that the average for the original activation function is the best one to use for the code since the fitness measurements recorded were much higher than any of the other activation functions (Cube, Sin, Inv, Gauss, and Hat).

## 8.2 Graphs

# 9.0 Conclusion

The purpose of this experiment was to test the effect of the activation functions (Cube, Sin, Inv, Gauss, and Hat) on the program’s fitness. The fitness of the activation functions stayed reasonably low other than the original program that had a significantly higher fitness average than the other functions. Additionally, the activation function that was able to update the weights and biases based on the error outputs such as the original function used will improve the fitness of the program. Therefore, the experiment did not support the hypothesis as the activation function that were being tested were not able to demonstrate an improvement in the fitness compared to the original activation function that was being used.

# 10.0 References

The Pennsylvania state university. (2023). Control and Placebo Groups. Pennstate. <https://online.stat.psu.edu/stat200/lesson/1/1.4/1.4.4>

Geeksforgeeks. (2023, January 5). Backpropagation in data mining. Geeksforgeeks. https://www.geeksforgeeks.org/backpropagation-in-data-mining/

Geeksforgeeks. (2023, February 17). Activation functions in Neural Networks. Geeksforgeeks. https://www.geeksforgeeks.org/activation-functions-neural-networks/

Pearson Australia. (2017). Pearson science S.B. 2nd edition, Pearson Australia, 707 Collins Street, Melbourne, Victoria.