AI-Car-Simulator Experiment

By Jake Handel

Table of contents:

Experiment setup:

Question:

How does the activation function affect the learning of the car's learning patterns through generations?

Aim:

This experiment aims to figure out which activation function allows the cars to complete the same track first (in the least amount of generations).

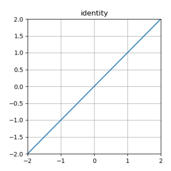
Hypothesis:

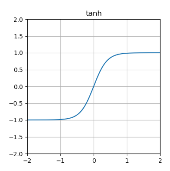
If the exponential growth of a function is related to how fast the AI learns the track then the softplus and exp functions will learn the fastest and the functions such as tanh and sigmoid will learn the slowest due to their plateau in their learning.

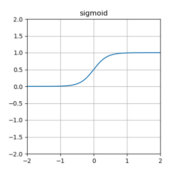
Variables:

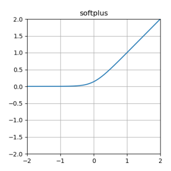
Independent:

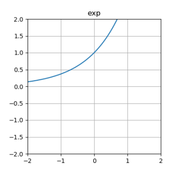
In this experiment, we will be changing the activation functions that allow the cars to learn. The experiment will change between the functions: *identity*, *tanh*, *sigmoid*, *softplus*, and *exp* functions. The control variable is the identity function due to it being a simple linear relation.











identity tanh sigmoid softplus exp

Source: CodeReclaimers, 2019

Dependent:

In the experiment, the dependent is the number of generations it takes for one car to complete the track for each type of function.

Controlled Variables:

What needs to be controlled	Why does it need to be controlled
The map	The map needs to be controlled, if the map is changed every time we change a function the map may get harder or easier which affects how quickly the car will learn the track and complete the track. If the map changes we won't know if the map is affecting the learning or the functions.
Different variables	Variables such as starting speed the angle and the size of the car and many more variables within the Python and config files need to stay the same. Hence, it is a more reliable test. If some of the variables changed in the test then it would make the test invalid due to the machine learning and the car acting differently.
General code	If the code changes or the algorithm changes then the learning of the cars will change and impact the results of the experiment. If the code changes it will change how the code runs and how the car learns which affects the different variables.

Method:

1. Copy and pull code from the GitHub page, click here for the repo, or the original repo here (Jetscholar, 2023)
2. Click the green code button and copy the https link
3. Go to your terminal and paste "git pull [link]", this will load the code in the directory you downloaded it into
4. Open that directory with "code ." to open it with Visual Studio code
5. Open the config.txt file
6. Find the lines 11 and 13 in the code, these will be the lines you change
7. Change the functions to 'identity' for the starting test
8. Run the program by typing "python newcar.py" into the terminal
9. Record how many generations it took for the car to cross the finish line
10. Repeat each function three times total
11. Change the function to tanh, sigmoid, softplus, exp
12. Average all the trials and conclude the experiment

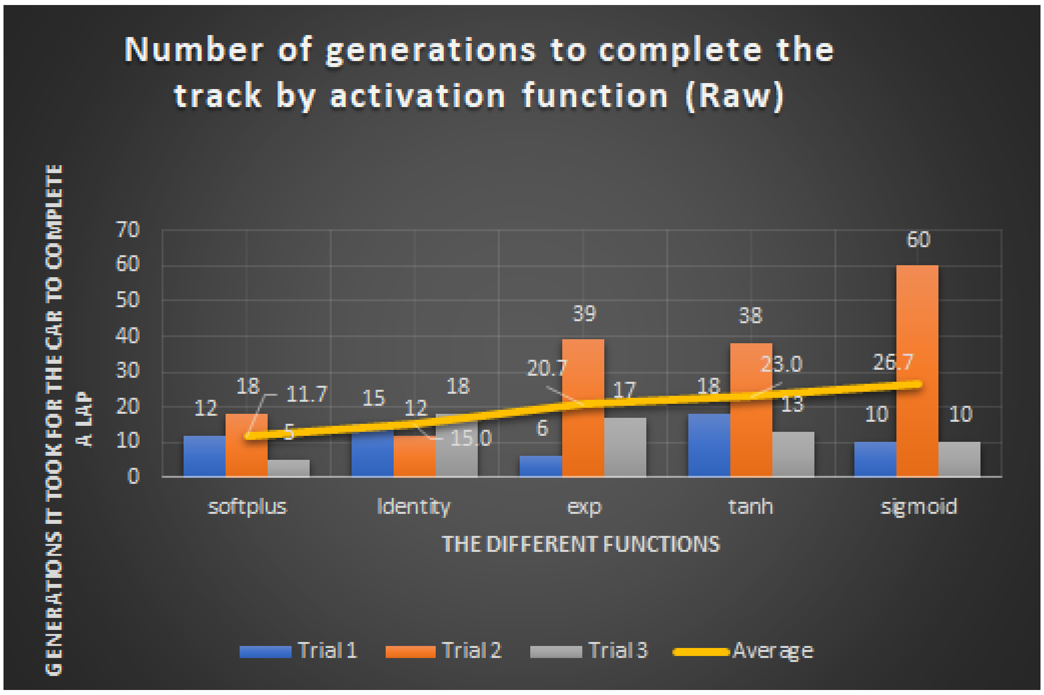
Results:

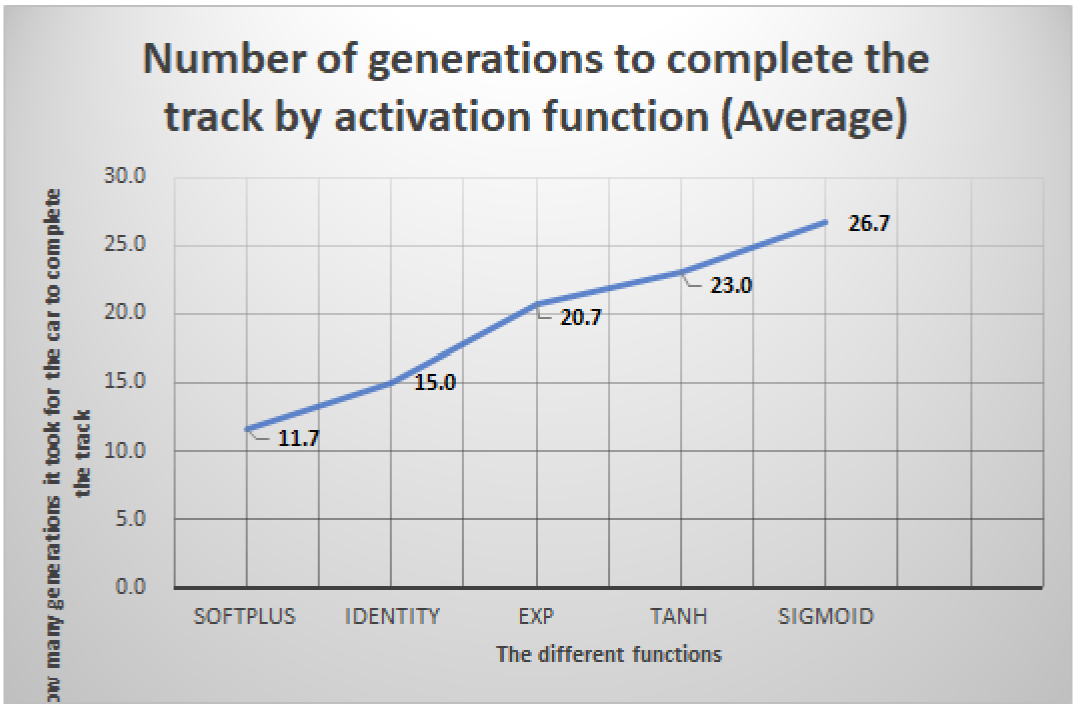
Table:

How many generations does it take to complete one lap of the same track when changing the type of function used?

	How many generations does it take for one car to do a lap?			
Type of functions	Trial 1 (T1)	Trial 2 (T2)	Trial 3 (T3)	Average (1dp)
Identity (Control)	15	12	18	15.0
tanh	18	38	13	23.0
sigmoid	10	60	10	26.7
softplus	12	18	5	11.7
exp	6	39	17	20.7

Graphs:





Summary of results:

Shown by the graphs, the softplus function on average was the best machine learning algorithm as it was able to learn the track the quickest. The control function, identity, learnt the track pretty fast with an average of 15 generations. The other exponential function, exp, performed okay with an average of 20.7 but that did not correlate with what I hypothesized at the start. Sigmoid and Tanh functions performed how I expected them to as they were the slowest to learn the track. The results shown in the average graph are what was predicted to happen besides the identity and exp functions as it was predicted that the exp function would be faster learning than the identity function. There was one anomaly in the experiment which was the second trial of the sigmoid function where it had a difference of 50 generations compared to the other two trials. The rest of the experiments were reliable and useful for the experiment.

Evaluation:

This investigation was very useful in showing the difference and the significance of the activation function used in machine learning and neural networking. The experiment was generally reliable and valid with only one outlier in the trials but aside from that the experiment was reliable. If the experiment were to be redone, we would have more trials so the average could be more reliable and useful for the results. If we were to increase the trials the experiment would be more valid as there would be more information to average out and it should account for distinct anomalies in the data. If done again, the experiment should include more activation functions, and group them together. The groups should be categorised, such as exponential, linear, and s-shaped/ numerically controlled. Each of these groups would consist of two or three different functions that would be tested more times for a better result. With the functions being grouped together, the experiment could define what type of function works best for machine learning and you can break down the results more. This would allow for a better understanding of the activation and what affects the learning the most.

To understand why the results happened it all comes down to the basic graph of the activation functions. TanH and Sigmoid functions are very similar in their shape and the way they process information and input data. The main difference between the two is that the sigmoid ranges from a numerical distance of 0 to 1 in an S-shape and the tanh function ranges from -1 to 1 and is also an S-shape (dishashree26, 2023). These functions didn't perform well in the experiment compared to the other functions due to the limit in numerical range and the maximum and minimum numbers for learning. These two S-shaped functions are in contrast to the infinite rises of the identity, exp and softplus functions. These functions are categorized in two, linear and exponential functions with identity being a linear function and exp and softplus both being exponential functions. The identity function works well for the learning but doesn't learn as much as others due to the gradient being the same for every iteration, this will mean it won't be able to train well and capture complex patterns (dishashree26, 2023). The linear equation dows not have any weight when taking the inputs and only spits out values with no calculations (Bahet, 2021). Overall the identity function is good for simple programs but not good for very complex ideas. The use of softplus and exp functions are an important contrast between the functions. Where tanh and sigmoid have a finite range, both sxp and softplus have an infinite range meaning they have an infinite learning capacity (Serengil, 2020). The difference between exp and softplus is the change in exponential rise between the graphs. The softplus function has a slow but steady increase and then a fast exponential rise in the graph making it able to change easily and efficiently for each generation. This allows for the function to learn faster and adapt to the conditions better then other functions. The exp function is a generic exponential function which moves quick but can't be adjusted easily in the learning of a car but is able to still learn and change unlike the identity function. It is still a good function to use but the rapid increase does not help the learning process of the AI.

The different activation functions were significant to the experiment as they provided different ways of figuring out the weight of each input and how much they effected learning of the AI. Neural networking has neurons that work with the weight and/ or bias of the inputs and how they interactive with the different activation functions (Tiwari, 2023). The activation function changes how the weighting of the inputs are being calculated and biased towards the output layer which is how the AI makes the decisions on how to move. The input layer is the values of the radars in this instance and then the distances from the car to the border are calculated and those go through the activation functions and give an output of, go left, right, speed up or slow down. The activation function changes the biasing of outputs which is how the activation effects the outputs of the AI's decision making.

The experiment has demonstrated that the activation function has a significant effect on the machine learning patterns and the NEAT algorithm. The experiment has demonstrated to us that there is better ways to learn for each function and in this specific case the softplus function had the most beneficial effect on the machine learning algorithm. There were only one error that made this experiment a little invalid and not enough trials but the experiment was very true and reliable for the purpose. The experiment gave us a good demonstration on how the activation function effects the learning of the Ai car and the significance between them.

References:

Bahet, P. (2021) *Activation functions in neural networks [12 types & use cases]*, *V7*. Available at: https://www.v7labs.com/blog/neural-networks-activation-functions (Accessed: 11 September 2023).

CodeReclaimers (2019) *Overview of builtin activation functions*, *NEAT*. Available at: https://neat-python.readthedocs.io/en/latest/activation.html (Accessed: 11 September 2023).

dishashree26 (2023) *Fundamentals of deep learning - activation functions and when to use them?*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/ (Accessed: 11 September 2023).

Jetscholar (2023) *Jetscholar/ML_TASK2*, *GitHub*. Available at: https://github.com/jetscholar/ML_Task2 (Accessed: 10 September 2023).

Serengil, S. (2020) *Softplus as a neural networks activation function*, *Sefik Ilkin Serengil*. Available at: https://sefiks.com/2017/08/11/softplus-as-a-neural-networks-activation-function/ (Accessed: 11 September 2023).

Tiwari, S. (2023) *Activation functions in neural networks*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/activation-functions-neural-networks/ (Accessed: 11 September 2023).