Effects of activation functions on neural networks

Date: 10 September 2023

Research Question:

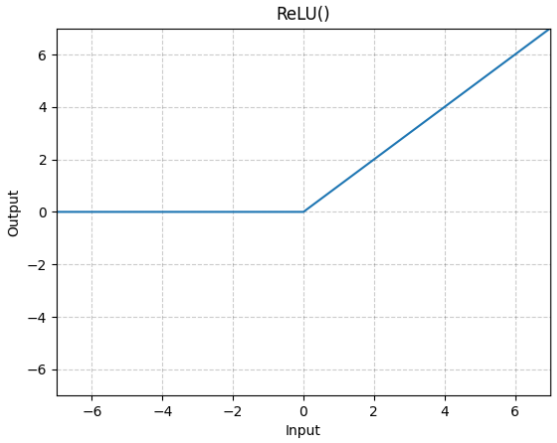
What is the comparative efficiency of various neural network activation functions, such as ReLU, Sigmoid, and Tanh and Cube, in optimizing the performance of autonomous vehicles within a racing track simulation?

Background Information:

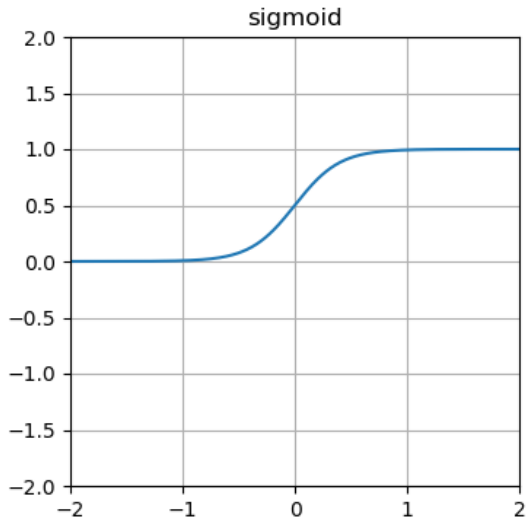
An activation function in a neural network is a mathematical function that determines the output of a neuron or node based on its input. It introduces non-linearity into the model, allowing the neural network to learn complex patterns and relationships in the data. Common activation functions include the Rectified Linear Unit (ReLU), which is known for its simplicity and effectiveness, as well as Sigmoid and Tanh functions, which are used in specific contexts where a bounded output range is desired.

Types of activation functions:

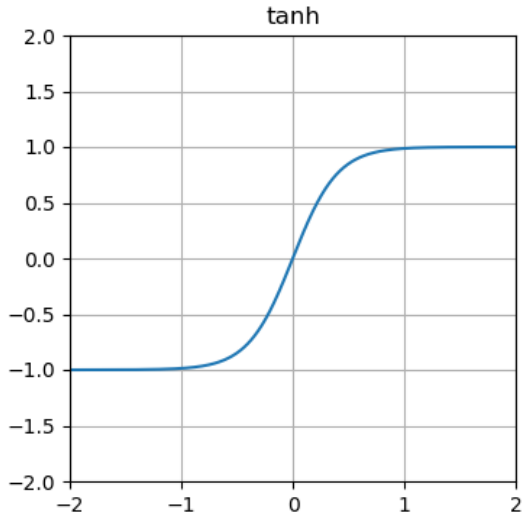
ReLU (Rectified Linear Unit): The ReLU activation function is defined as f(x) = max(0, x). It outputs the input value if it's positive, and zero otherwise. ReLU is widely used due to its simplicity and effectiveness in training deep neural networks.



Sigmoid: The sigmoid activation function, also known as the logistic function, is defined as f(x) = 1 / (1 + e^(-x)). It squashes the input values into a range between 0 and 1, making it suitable for binary classification problems and as an output activation in the final layer of such networks.



Tanh (Hyperbolic Tangent): The hyperbolic tangent activation function, tanh(x), is similar to the sigmoid but maps input values to a range between -1 and 1. It helps mitigate the vanishing gradient problem and is often used in hidden layers of neural networks.



Variables

Independent:

The independent variable will be the activation function that the neural network uses to calculate the output of the node and whether it should be activated or not. Based on this the program will have different learning speeds depending on the function used.

Dependent:

The speed of the learning process of the cars. This will be measured in terms of seconds until one of the cars is able to complete a full lap.

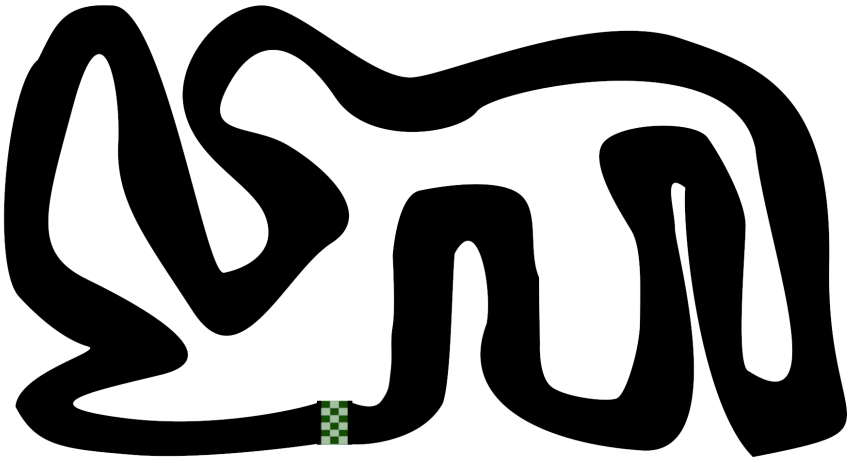
Controlled variables:

Name of the control variable	Why do you need to keep this variable the same
Track	This variable is required to be kept the same so that the cars will have an equal opportunity to learn the track. If a simpler track is used then the cars will learn quicker on one trial than on another.
Radars	The amount of radars will affect the sensitivity of the car to its surroundings. This means that if more or less radars are in use by the car then the car will be able to detect its surrounding environment better making the learning process quicker.
Starting position	The starting point should be the same so that the cars have an equal opportunity to travel around the track at a certain speed and thus time.
Same settings (speed etc)	The car settings like the speed will influence how quickly the car drives thus if this is not controlled then some will have a faster time than others.

Hypothesis:

As the activation function changes the speed of the learning process will also change reflecting the suitability of the function to the task at hand.

Track which will be used:



Method:

1. Clone the GitHub code to a local workspace
2. Change the map to map number 3
3. Select 3 activation functions to assess training efficiency: Tanh, Relu, Sigmoid
4. Next modify the activation function, by navigating to the 'DefaultGenome' object within the config.txt file and edit the 'activation_default' and 'activation_options' variables according to your desired settings.
5. Open a new terminal window and execute the program
6. Start a stopwatch the moment that the program opens and stop it when a car has completed a total lap
7. Repeat step 6, 3 times for each function and place the results into a table
8. After all the 3 activation functions had been tested, with data collected for 3 separate trials for each individual function, all the relevant data had been collected

Table of Results

The effect that the activation function used has on the learning speed of a neural network, measured in seconds.

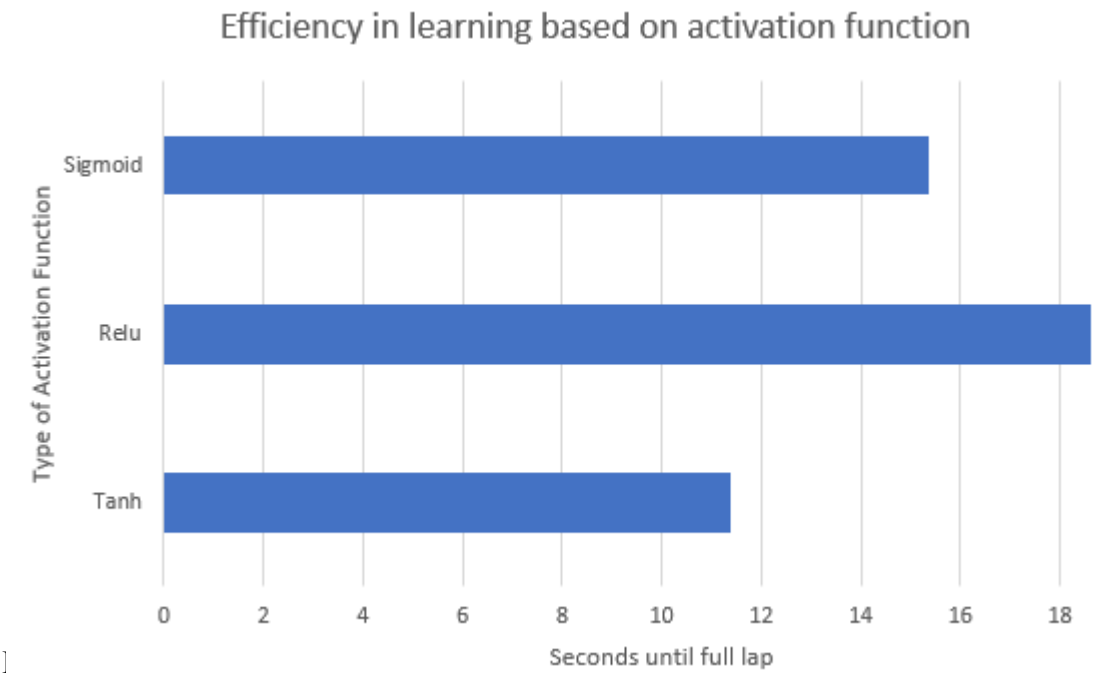
Activation function used	**Time to reach one full lap**			**Average (seconds)**
	Trial 1	**Trial 2**	**Trial 3**	
Tanh	11.36	10.29	13.45	11.37
Relu	19.66	18.31	17.89	18.62
Sigmoid	15.45	15.12	16.30	15.362

Sample Calculations:

Average time = $\frac{time1 + time2 + time3}{3}$

Average time = $\frac{11.36 + 10.29 + 13.45}{3} = 11.37$

Graph:



]

Analysis of results

In this experiment, we investigated the influence of different activation functions on the speed of neural networks (dependent variable), while considering the type of activation function as the independent variable. Three commonly used activation functions, Tanh, Relu, and Sigmoid, were examined. Our results show that the choice of activation function has a significant impact on the speed of neural networks. Tanh activation led to the shortest average processing time (11.36 seconds), followed by Sigmoid (15.29 seconds), and Relu (19.62 seconds). This suggests that Tanh activation may facilitate faster neural network computations, while Relu tends to be the slowest among the tested functions. Further analysis is needed to understand the underlying mechanisms driving these differences in neural network performance with various activation functions.

Conclusion

The data reveals that the choice of activation function significantly impacts the speed of neural networks during training. Specifically, the Tanh activation function demonstrates the fastest training times with average durations of 11.36, 10.29, and 13.45 seconds across three trials. In contrast, the ReLU activation function takes longer, with average times of 19.66, 18.31, and 17.89 seconds. The Sigmoid activation function falls in between, with average times of 15.45, 15.12, and 16.30 seconds. This suggests that selecting the appropriate activation function is crucial for optimizing neural network training efficiency.

Reference List:

- 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.

- SHARMA, S. (2017). Activation Functions in Neural Networks. [online] Medium. Available at: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

- Panneerselvam, L. (2021). Activation Functions | What are Activation Functions. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/04/activation-functions-and-their-derivatives-a-quick-complete-guide/.