# Neural Networks Fail to Learn Periodic Functions and How to Fix It

***NikunjRathod*** 202211014 | ***Dipen Padhiyar*** 202211058 | ***Arjun Vankani*** 202211036 | ***Vivek Soni*** 202211069

*Mentor: Prof. Rachit Chhaya*

*DhirubhaiAmbani Institute of Information and Communication Technology, Gandhinagar, 388007*

## Reproducibility Summary

### Scope of Reproducibility

How to Fix the Neural Networks' Failure to Learn Periodic Functions [1] uses experimental evidence to show that common activations like ReLU, tanh, sigmoid, and their variations are all incapable of learning how to extrapolate basic periodic functions. The research went on to suggest a new activation, which is referred as the snake function.

The paper's two main assertions are as follows: (1) The neural networks inherit the characteristics of the activation functions. While ReLU extrapolates in a linear fashion, a tanh network will be smooth and extrapolates to a constant function. Periodic functions cannot be extrapolated using conventional neural networks with conventional activation functions. (2) The suggested activation function is able to optimize as well as traditional activation functions and learn periodic functions. Although the claims are supported by both experimental evidence and theoretical arguments, we will only be interested in experimentally testing the claims.

### Methodology

All experiments were replicated using only the information provided in the paper. All dataset links were also provided in the original paper. This enabled us to build the majority of our experiments from the scratch.

### Results

The central thesis of the paper, that the proposed snake non-linearity can learn periodic functions, was successfully replicated in our studies.

### What was easy?

Many experiments included descriptions of the neural network architectures and performance graphs, providing us with a precise standard against which to compare our findings.

### What was difficult?

There was no information available for the experiment on human body temperature. Snake was used to initialize the weights in neural networks and with RNNs, but proper implementation details weren't provided.
Project Link : https://github.com/dipenpadhiyar/AML-reproducibility/tree/master/AML-Reproducibility

# 1. Introduction

In a variety of disciplines, including computer vision, speech recognition, and language modeling, deep neural networks are becoming more and more important. But even soalthough ordinary versions of these networks are not suitable for extrapolation outside of the training range, neural networks are effective tools for interpolating between existing data. Thismakes it difficult for them to forecast solutions to issues having periodic components.Using periodic activation functions has been one approach taken in the past to overcome neural networks' incapacity to learn periodic functions. For instance, use sin(x) asComplex natural signals and their derivatives can be successfully represented by implicit neural representations' activation function.However in more general cases, experimental results suggest that using sin as the non-linearity cannot compete against ReLU-based activation functions on standard tasks.

The original paper: (1) investigates the extrapolation properties of a neural network beyond a bounded region; (2) demonstrates that standard activation functions are insufficient for neural networks to learn periodic functions outside the bounded region where data points are present; and (3) suggests a novel activation function as a solution to this issue function and its variations, demonstrating how well it performs on toy and real-world problems. We verified the assertions stated in the original study by conducting experiments that showed that ordinary activation functions were unable to learn periodic functions also the outcomes of the unique activation function on plaything and practical tasks.

# 2. Scope of Reproducibility

The authors make two key claims:

- Standard neural networks with conventional activation functions are insufficient to learn periodic functions outside the restricted region where data points are present.
- The novel activation function that has been presented can learn periodic functions while retaining the ReLU-based activations' advantageous optimisation property. The "snake" activity in the novel is described as follows:

$$\text{snake}_a(x) := x + (1/a)\sin^2(ax)$$

where a is treated as a fixed parameter in initial experiments, and as a learnable parameter in a few experiments. Snake is shown to outperform standard activation functions ReLU, tanh, LeakyReLU, as well as more recently proposed functions such as swish, and sin.

The two statements in the original work are supported by both theoretical reasoning and a lengthy list of experiments that span from evaluating performance on toy datasets to real-world applications because of the two claims' broad and far-reaching implications. Utilizing the suggested activation function in a Deep Learning framework, we have thoroughly recreated the original set of trials and carried out a few extra experiments of our own. To create images of handwritten numbers, a Convolutional Generative Adversarial Network (DCGAN) was used, together with a Long Short Term Memory (LSTM) network for sentiment analysis.

# 3. Methodology

The code used by the authors had not been made public at the time we started working on re-implementing the paper. The provided code was coded using PyTorch Library while we have code the same using Tensorflow library and get somewhat same results.

### 3.1 Model descriptions

Models used in the original paper included fully-connected, feed-forward neural networks with different architectures for the various experiments. Larger standard models such as ResNet18 were also used. The authors of the original paper had initially not made their code available and we had to implement most models ourselves.

### 3.2 Datasets

The data used in the extrapolation experiments are directly sampled from periodic functions such as sin(x). Some experiments dealt with standard dataset **CIFAR-10**.

Data for the real-life dataset **Patient body temperature** was downloaded.

### 3.3 Hyperparameters

Different trials featured various levels of hyperparameters detail. The architecture of the neural network was generally described in many trials (e.g., "4-layer fully connected neural network"), but other hyperparameters and important information, such as batch size, loss function, or learning rate, were not included. When necessary, assumptions had to be made in the absence of information in order to get a near approximation of the original result. Grid searching was used to test the architecture (number of layers and neurons in each layer), batch size (16–512), optimizer (Adam, SGD, RMSProp), learning rate (0.001–0.1), and value of a in network with the snake activation (1–30).

### 3.4 Experimental setup

The entire codebase has been uploaded to GitHub and is publicly available: https://github.com/mayurak47/Reproducibility_Challenge. The experiments were run locally as well as on GPU enabled sessions on Google Colab. All the models and experiments were codedusing the Tensorflow library.

### 3.5Computational requirements

Many of the experiments, particularly those relating to regressing different functions and datasets, could be run locally on a Windows machine with 8 GB of RAM, not requiring more than a few minutes to train. The more demanding experiments required the use of GPUs. Training a ResNet18 on CIFAR-10 with six activation functions for 100 epochs took roughly 12 hours on a Tesla T4 GPU on Google Colab.

## 4.Results

### 4.1 Application

Wherever possible, the claims of the original papers were tested and in each case, we were able to reproduce the original results. Multiple experiments are conducted to illustrate the performance of snake on a range of tasks.

**Body temperature regression:**

The core utility of snake is shown via two real-life problems. The two tasks are predicting the evolution of temperature in Minami-Tori-shimaisland in Japan, and the modeling the body temperature of a patient. The architectures used are $1 \rightarrow 100 \rightarrow 100 \rightarrow 1$ and $1 \rightarrow 64 \rightarrow 64 \rightarrow 1$ respectively, as in the original paper. In the Minami-Tori-shima weather experiment, the parameters a were made learnable; in the body temperature experiment, a = 30. In both cases, snake is the only activation function that makes meaningful extrapolation and predictions.

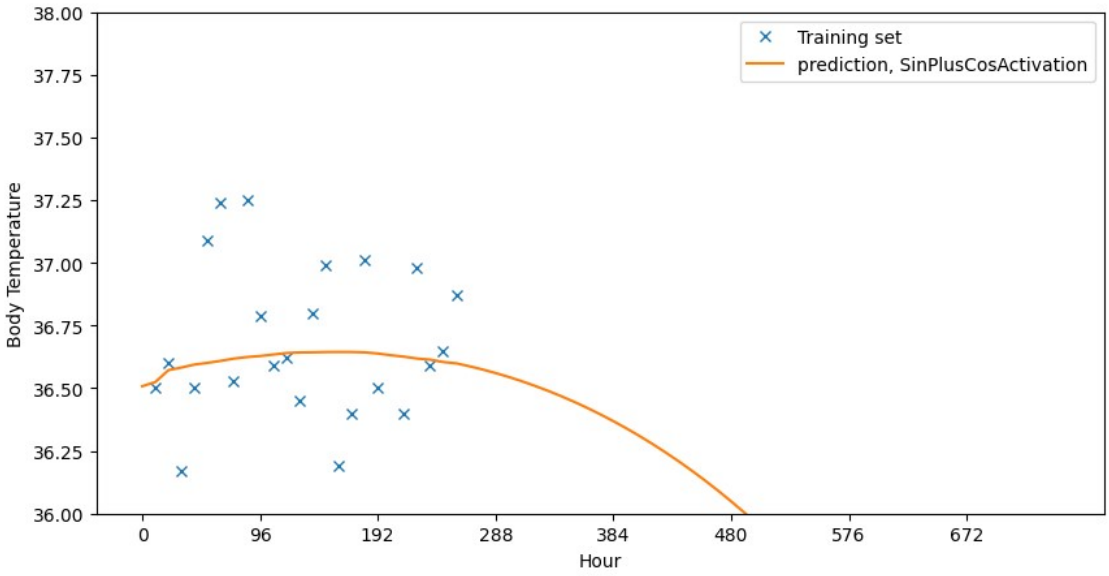


Figure 1

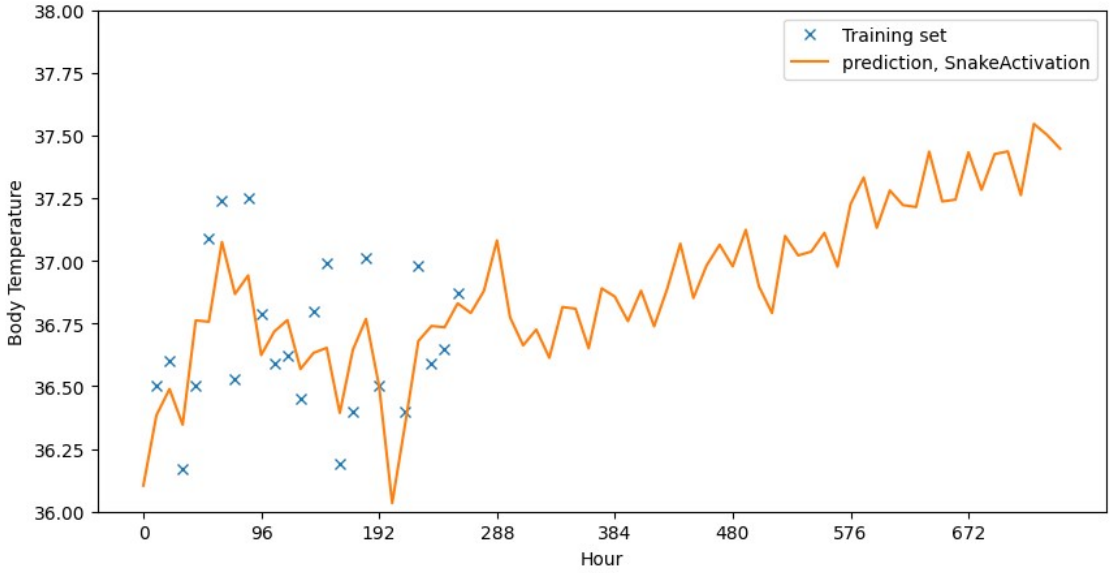


Figure 2

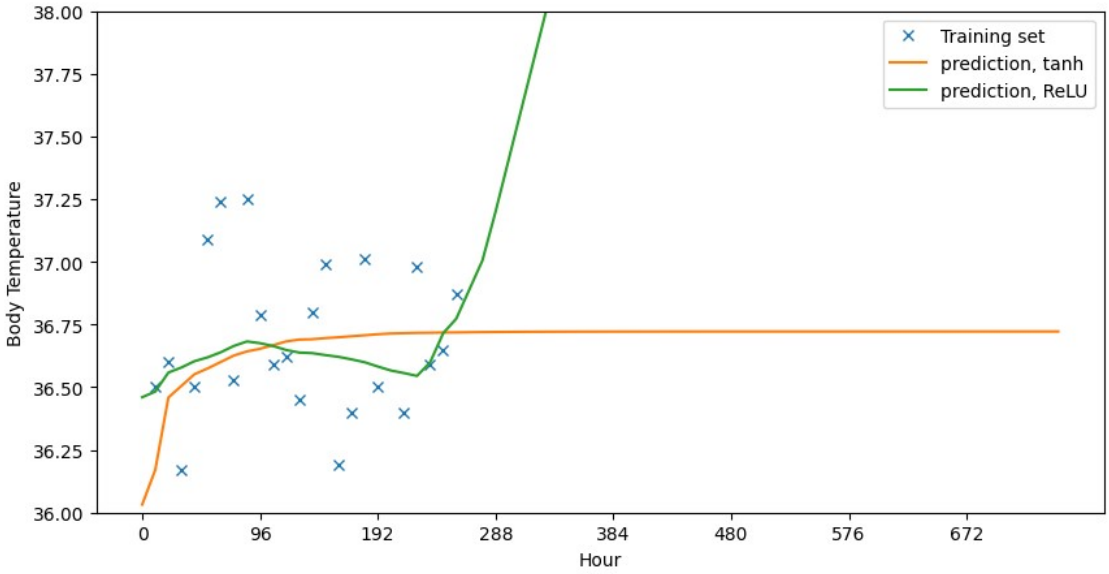


Figure 3

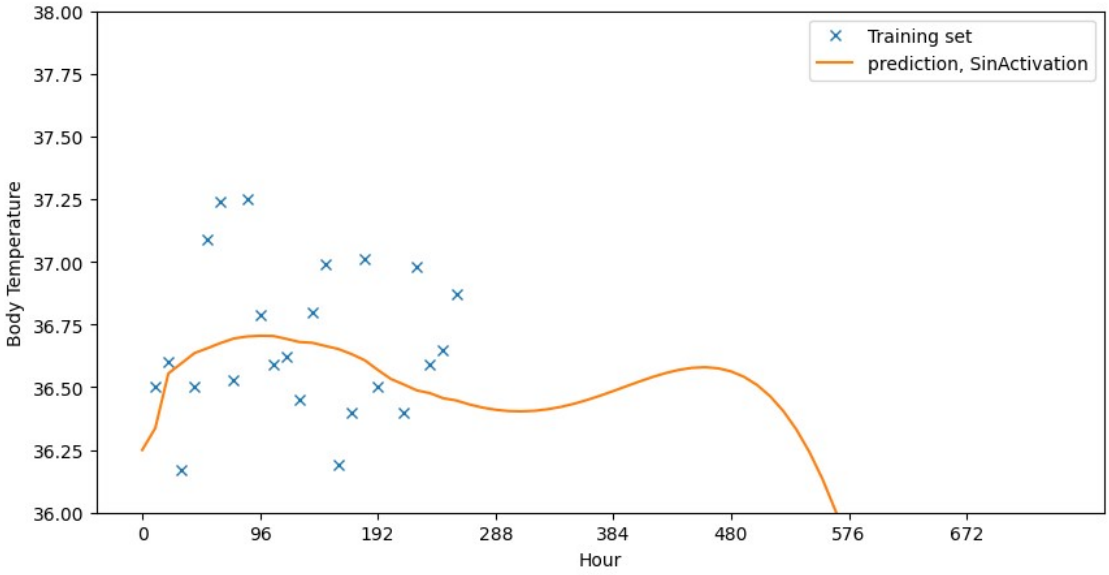


Figure 4

Observation: Here we can see from figure that activation functions except snake function can't fit to data and it will be having more training and generalization error.We have taken hyper parameter 'a' and its value for body temperature regression is 1. Here we have shown different graphs for different activation function so that we can easily understand it.

**Cifar10 with resnet18 (in this project we have used resnet50):**

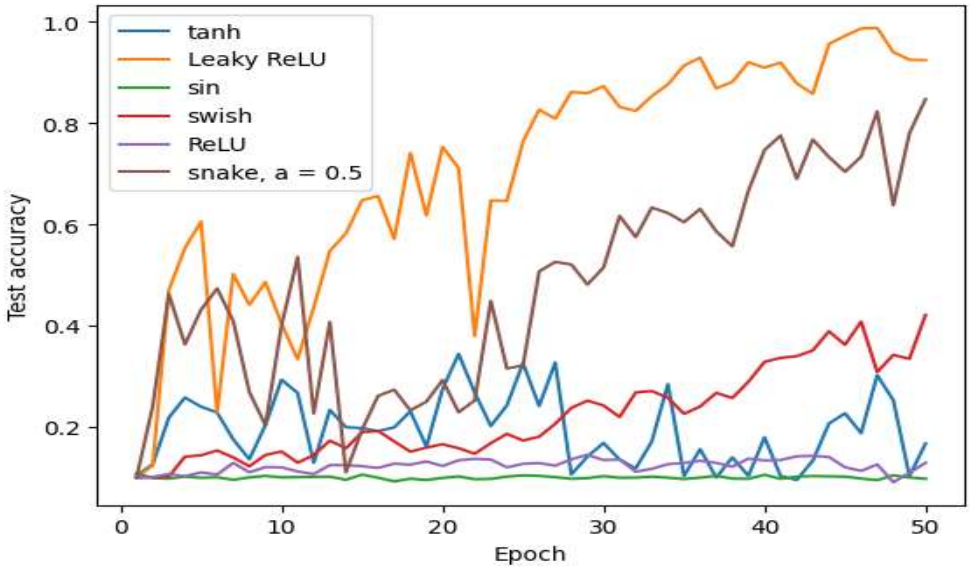


Figure 5

In original paper they use resnet18 architecture but we use resnet50 to reproduce the code.Multiple experiments are conducted to illustrate the performance of snake on a range of tasks.ResNet50 [], with 20M parameters, is trained on the CIFAR-10 dataset. This is a 10-way image classification task. The ReLU layers in the architecture are replaced with the specified activation, and the network is trained for 50 epochs. The Adam optimizer is used in this; the learning rate is 1× 10−4 for the 50 epochs. A test accuracy of 91-92% is achieved by the snake network according to figure, in line with that of the other standard activation functions. This suggests that snake is suitable for large-scale image classification problems, and may be used as a straightforward alternative to other activation functions.

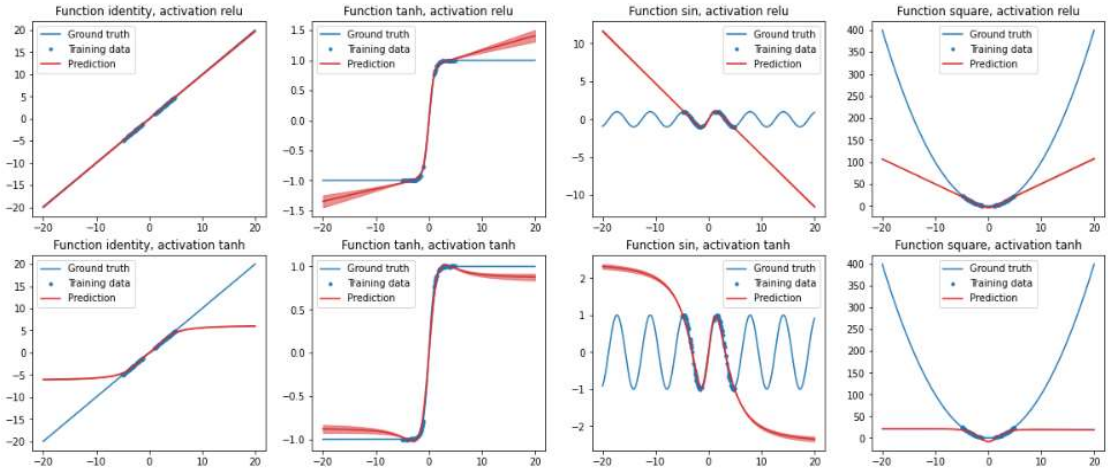


Figure 6: Different functions with activation function as ReLu and tanh

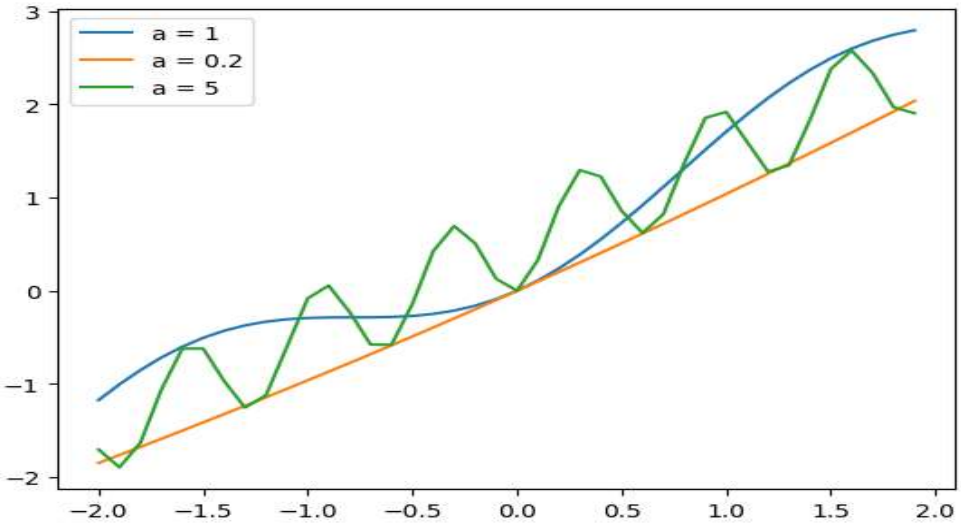


Figure 7: Snake Function with different Parameters (a)

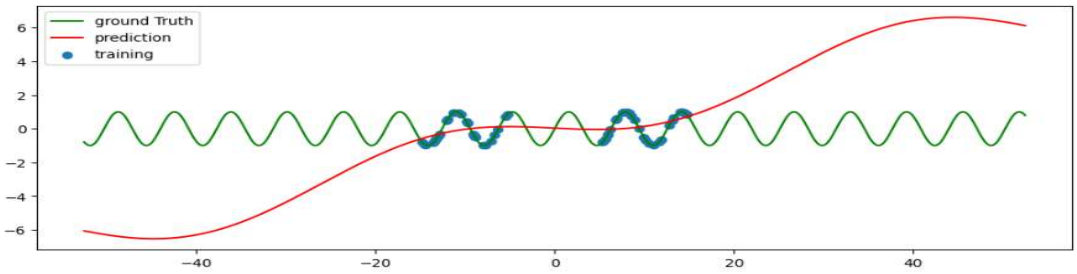


Figure 8: Sin Function with Sin as Activation Function

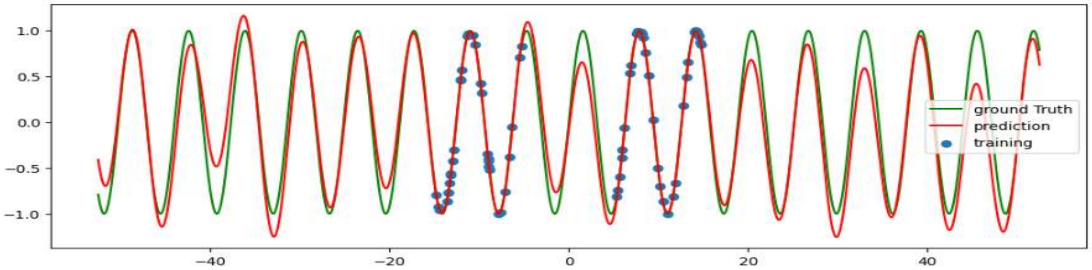


Figure 9: Sin Function with Snake as Activation Function

## 5. Discussion

Because the authors originally did not publish their code and only included a brief descriptionscripts for network architectures used in experiments, exact replicationtheir test results were not possible. But paper by meant that only the relative performance of the stomach compared to the rest of the activationfunctions in defined problems was interesting, as opposed to precise architecturedetails or loss values achieved. For example, in Table 1 and observed lossesFigures 5b are orders of magnitude different from the values in the original paper, which is probably due to various normalization techniques and hyperparameters, although

they are again commonin the brackets seen in figures 5a and 5. 7 are the same as on the original paper.We were able tosupport the claim that neural networks with standard activation work,not enough to learn periodic operations outside the training area. We werealso ensures the normal operation of the proposed activation functionactivation functions, ReLU, tanh, LeakyReLU, in various tasks (e.g.LSTM experiment concept), repeating the experiments in the original paperand some others we introduce ourselves. Further work could focus ontheoretical reasons for snake behavior and develop more appropriate optimization algorithms.

### 5.1 What was easy?

We were able to closely replicate the experiments thanks to a comprehensive description of the neural network architectures used for training on the MNIST dataset and human body temperature. The paper itself also included links to datasets for all but one experiment. Additional experiments comparing snake's performance with other activation functions were listed in extensive appendix sections. These experiments were supported by graphs showing snake's performance with other activation functions, giving us a clear metric against which to compare our reproductions' results.

### 5.2 What was difficult?

The first source code was not given at first and we needed to depend on the portrayals of models and hyperparameters (which were missing as a rule) and taught mystery while endeavoring to imitate the outcomes. There was a lack of data for the human body temperature experiment. Hypothetical support for change rectification and the consequences of this fluctuation remedy utilizing ResNet101 on CIFAR-10 were given, be that as it may, execution subtleties were excluded. The use of snake on a feed forwardnetwork is stated in the section on Comparison with RNN on Regressing a Simple Periodic Function without any additional information regarding the hyperparameters used. Since white noise had been added to the data, it was impossible to precisely replicate the experimental setup, so the dataset for the experiment had to be inferred from the graphs of the results.

## 6. References

[1] Original paper { https://proceedings.neurips.cc/paper/2020/file/1160453108d3e537255e9f7b931f4e90-Paper.pdf }

[2] Github repo { https://github.com/mayurak47/Reproducibility_Challenge }

[3] Snake function blog { https://datasciencemystic.com/snake-activation-function-to-learn-periodic-functions-in-tensorflow-2/ }