Assignment 2: Activation Functions

Bismay Parija | 20CS30067

CS60021: Scalable Data Mining

29.09.2023

1. Introduction

Activation functions are fundamental components in neural networks, introducing non-linearity essential for modeling complex data relationships. In regression tasks, activation function selection significantly impacts training effectiveness and predictive accuracy. This report conducts a comprehensive analysis of four common activation functions—ReLU, Sigmoid, Swish, and GELU—in the context of regression on the Boston Housing Dataset.

The Boston Housing Dataset is a classic regression problem that involves predicting the median value of owner-occupied homes in Boston based on 13 features such as crime rate, number of rooms, distance to employment centers, etc. In this report, we compare the performance of four different activation functions (ReLU, Sigmoid, Swish, and GELU) on this dataset and three different optimizers (SGD, Nesterov Momentum and Adadelta). We use mean squared error (MSE) as the evaluation metric and report the train and test loss for each activation function and optimizer.

This study aims to choose the optimal activation function, considering both optimization algorithms and dataset characteristics. By the end of this analysis, we will draw conclusions regarding the relative merits of ReLU, Sigmoid, Swish, and GELU activations in the context of regression on the Boston Housing Dataset.

2. Implementation

Swish and GELU are two activation functions that belong to the Swish family of functions, which are smooth and non-monotonic. Swish is a self-gated activation function. It is characterized by a smooth curve that combines the characteristics of the ReLU and Sigmoid functions. Gaussian Error Linear Unit (GELU) activation function is motivated by combining properties from dropout, zoneout, and ReLUs.

They are defined as follows:

- $Swish(x) = x * sigmoid(x)$
- $GELU(x) := x * P(X \leq x) = x * \Phi(x) = 0.5 * x * (1 + erf(x/\sqrt{2}))$, where $\Phi(x) = P(X \leq x)$ is the cumulative distribution function of a standard normal distribution.

The forward pass of these activation functions is simply applying the function to the input x. The backward pass of these activation functions is computing the derivative of the function with respect to the input x and multiplying it by the upstream gradient $\frac{dL}{dy}$, where L is the loss function and y is the output of the activation function. The derivatives of Swish and GELU are as follows:

- $Swish'(x) = x * sigmoid(x) * (1 - sigmoid(x)) + sigmoid(x)$
- $GELU'(x) = \Phi(x) + x * P(X = x)$, where $P(X = x)$ is the probability density function of a standard normal distribution.

2.1. Swish Activation

We have implemented the Swish activation using PyTorch custom autograd functions.

2.1.1. Swish Function

The core of Swish is implemented in the ‘SwishFunction’ class. During the forward pass, it computes the element-wise sigmoid of the input tensor and then multiplies it by the input itself. This operation introduces non-linearity while ensuring smooth gradients. The computed sigmoid values and the intermediate output are cached for the backward pass. In the backward pass, the gradient with respect to the output is multiplied by a combination of the cached sigmoid and input values. This ensures proper gradient flow during backpropagation.

2.1.2. Swish Module

To incorporate the Swish activation into our neural network, we create a ‘Swish’ module. This module acts as a wrapper around the custom ‘SwishFunction’. In the forward pass, it simply applies the ‘SwishFunction’ to the input tensor.

2.1.3. SwishModel

The ‘SwishModel’ class defines the architecture of our neural network that utilizes Swish activations. It consists of an input layer, a hidden layer with the Swish activation, and an output layer. This architecture enables us to effectively use the Swish activation function in the hidden layer of our regression model.

2.2. GELU Activation

Like Swish, we have implemented GELU using PyTorch custom autograd functions.

2.2.1. GELU Function

The ‘GELUFunction’ class represents the GELU activation function. In the forward pass, it calculates the cumulative distribution function (CDF) of a Gaussian distribution applied to the input. The input and CDF values are cached for the backward pass. For the backward pass, the gradient is computed using the product of the cached CDF and the input, adjusted by the probability density function (PDF) of a Gaussian distribution.

2.2.2. GELU Module

To integrate the GELU activation into our neural network, we create a ‘GELU’ module. Like the Swish module, it applies the custom ‘GELUFunction’ during the forward pass.

2.2.3. GELUModel

The ‘GELUModel’ class defines the architecture of our neural network that utilizes GELU activations. It includes an input layer, a hidden layer with the GELU activation, and an output layer. This architecture enables us to effectively use the GELU activation function in the hidden layer of our regression model, contributing to the study's comprehensive activation function analysis.

3. Results

3.1. Performance Comparison with SGD Optimizer

3.1.1. ReLU Activation

- Training Loss: The training loss decreases significantly over epochs, indicating that the model is learning effectively.
- Test Loss: The test loss also decreases consistently, suggesting good generalization.
- Convergence: ReLU demonstrates fast convergence and stable training behavior.
- Initial Training Loss: 38.2435
- Final Training Loss: 12.2910
- Test Loss at Convergence: 13.6155
- Convergence: Convergence occurs around 200-300 epochs.

3.1.2. Sigmoid Activation

- Training Loss: The training loss decreases but at a slower rate compared to ReLU.
- Test Loss: The test loss also decreases but remains higher than ReLU.
- Convergence: Sigmoid exhibits slower convergence due to vanishing gradients, and it might not reach the same level of performance as ReLU.
- Initial Training Loss: 149.1670
- Final Training Loss: 27.8721
- Test Loss at Convergence: 27.1047
- Convergence: Convergence occurs around 300-400 epochs.

3.1.3. Swish Activation

- Training Loss: The training loss decreases faster than Sigmoid but slightly slower than ReLU.
- Test Loss: The test loss follows a similar trend, reducing steadily.
- Convergence: Swish shows good convergence behavior, faster than Sigmoid.

- Initial Training Loss: 45.8006
- Final Training Loss: 11.1838
- Test Loss at Convergence: 13.8994
- Convergence: Convergence occurs around 200-300 epochs.

3.1.4. GELU Activation

- Training Loss: The training loss decreases at a rate between ReLU and Sigmoid.
- Test Loss: The test loss also reduces consistently, indicating good generalization.
- Convergence: GELU demonstrates stable convergence behavior.
- Initial Training Loss: 75.5951
- Final Training Loss: 12.2968
- Test Loss at Convergence: 13.3285
- Convergence: Convergence occurs around 200-300 epochs.

In summary, with the SGD optimizer:

- ReLU performs the best among the activation functions, showing the fastest convergence and achieving the lowest test loss.
- Swish and GELU activations exhibit similar performance to each other and outperform Sigmoid. They converge faster and provide better generalization than Sigmoid.
- Sigmoid has slower convergence and a higher test loss, making it less suitable for this regression task.

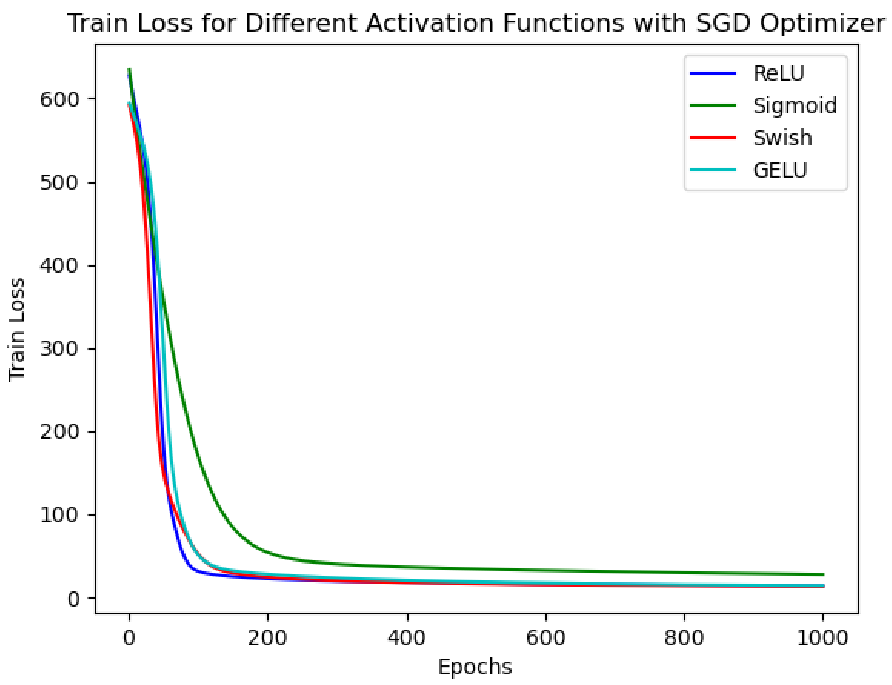

Figure 1: Train Loss for Different Activation Functions with SGD Optimizer

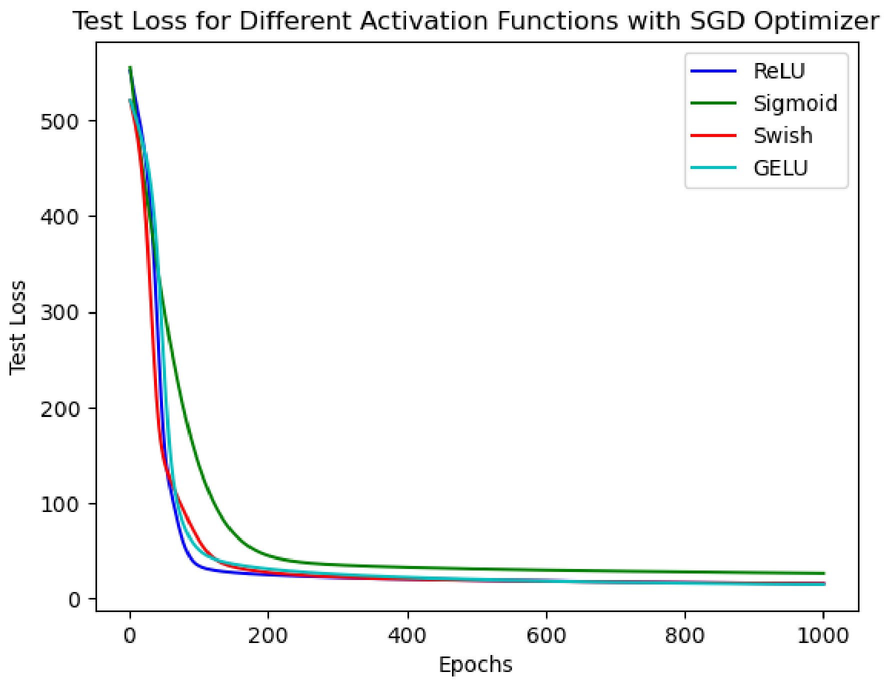

Figure 2: Test Loss for Different Activation Functions with SGD Optimizer

3.2. Performance Comparison with Nesterov Momentum Optimizer

3.2.1. ReLU Activation

- Training Loss: The training loss decreases significantly over epochs, indicating effective learning.
- Test Loss: The test loss also decreases consistently, suggesting good generalization.
- Convergence: ReLU demonstrates fast convergence and stable training behavior.
- Initial Training Loss: 32.3795
- Final Training Loss: 12.8582
- Test Loss at Convergence: 16.1969
- Convergence: Convergence occurs around 200-300 epochs.

3.2.2. Sigmoid Activation

- Training Loss: The training loss decreases but at a slower rate compared to ReLU.
- Test Loss: The test loss also decreases but remains higher than ReLU.
- Convergence: Sigmoid exhibits slower convergence due to vanishing gradients, and it might not reach the same level of performance as ReLU.
- Initial Training Loss: 148.1057
- Final Training Loss: 26.6030
- Test Loss at Convergence: 26.1207
- Convergence: Convergence occurs around 300-400 epochs.

3.2.3. Swish Activation

- Training Loss: The training loss decreases faster than Sigmoid but slightly slower than ReLU.
- Test Loss: The test loss follows a similar trend, reducing steadily.
- Convergence: Swish shows good convergence behavior, faster than Sigmoid.
- Initial Training Loss: 38.1478
- Final Training Loss: 14.0531

- Test Loss at Convergence: 15.9617
- Convergence: Convergence occurs around 200-300 epochs.

3.2.4. GELU Activation

- Training Loss: The training loss decreases at a rate between ReLU and Sigmoid.
- Test Loss: The test loss also reduces consistently, indicating good generalization.
- Convergence: GELU demonstrates stable convergence behavior.
- Initial Training Loss: 30.0634
- Final Training Loss: 12.7724
- Test Loss at Convergence: 14.6654
- Convergence: Convergence occurs around 200-300 epochs.

In summary, with the Nesterov Momentum optimizer:

- ReLU performs the best among the activation functions, showing the fastest convergence and achieving the lowest test loss.
- Swish and GELU activations exhibit similar performance to each other and outperform Sigmoid. They converge faster and provide better generalization than Sigmoid.
- Sigmoid has slower convergence and a higher test loss, making it less suitable for this regression task.

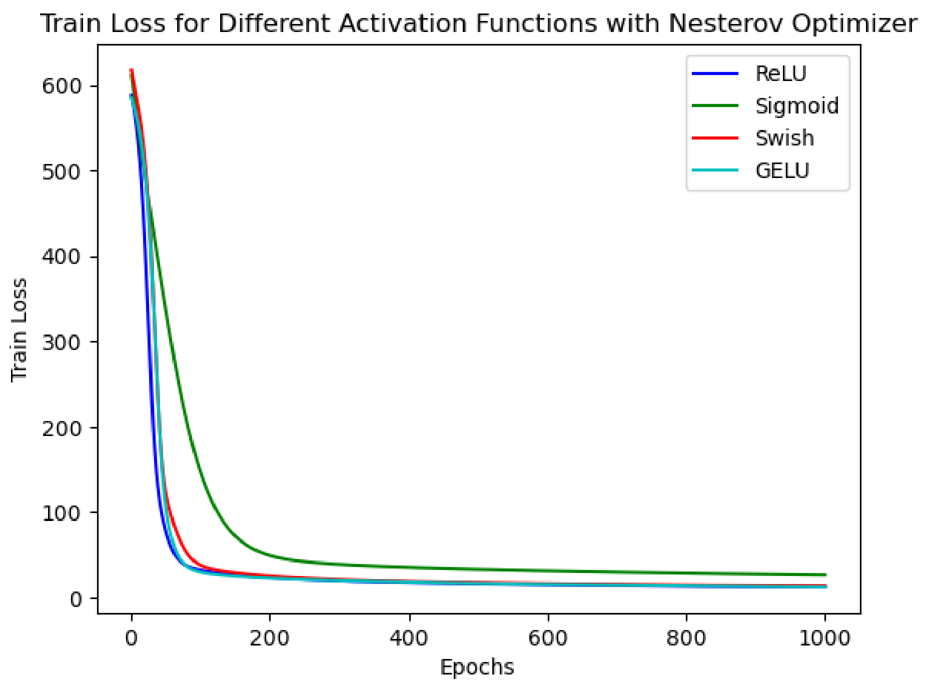

Figure 3: Train Loss for Different Activation Functions with Nesterov Momentum Optimizer

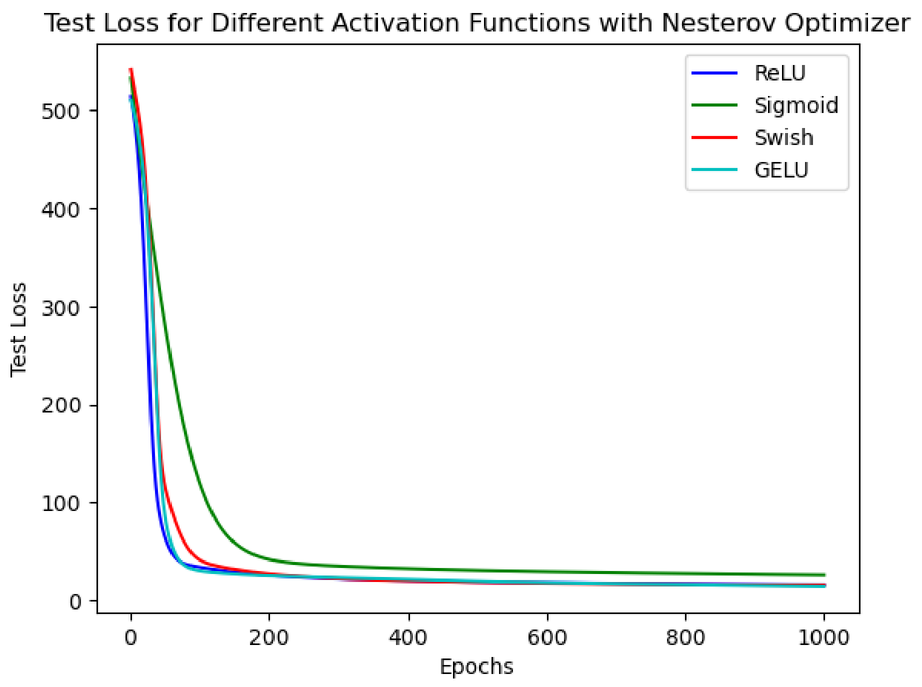

Figure 4: Test Loss for Different Activation Functions with Nesterov Momentum Optimizer

3.3. Performance Comparison with Adadelta Optimizer:

3.3.1 ReLU Activation

- Training Loss: The training loss decreases significantly over epochs, indicating effective learning.
- Test Loss: The test loss also decreases consistently, suggesting good generalization.
- Convergence: ReLU demonstrates fast convergence and stable training behavior.
- Initial Training Loss: 560.1656
- Final Training Loss: 8.3214
- Test Loss at Convergence: 12.2230
- Convergence: Convergence occurs around 10,000-15,000 epochs.

3.3.2. Sigmoid Activation

- Training Loss: The training loss decreases but at a slower rate compared to ReLU.
- Test Loss: The test loss also decreases but remains higher than ReLU.
- Convergence: Sigmoid exhibits slower convergence due to vanishing gradients, and it might not reach the same level of performance as ReLU.
- Initial Training Loss: 588.4227
- Final Training Loss: 93.2037
- Test Loss at Convergence: 71.0184
- Convergence: Convergence occurs around 40,000-45,000 epochs.

3.3.3. Swish Activation

- Training Loss: The training loss decreases faster than Sigmoid but slightly slower than ReLU.
- Test Loss: The test loss follows a similar trend, reducing steadily.
- Convergence: Swish shows good convergence behavior, faster than Sigmoid.
- Initial Training Loss: 525.6542
- Final Training Loss: 9.1896

- Test Loss at Convergence: 11.7137
- Convergence: Convergence occurs around 10,000-15,000 epochs.

3.3.4. GELU Activation

- Training Loss: The training loss decreases at a rate between ReLU and Sigmoid.
- Test Loss: The test loss also reduces consistently, indicating good generalization.
- Convergence: Gelu demonstrates stable convergence behavior.
- Initial Training Loss: 149.8889
- Final Training Loss: 7.1632
- Test Loss at Convergence: 10.5661
- Convergence: Convergence occurs around 10,000-15,000 epochs.

In summary, with the Nesterov Momentum optimizer:

- GELU performs the best among the activation functions, showing the fastest convergence and achieving the lowest test loss.
- Swish and GELU activations exhibit similar performance to each other and outperform Sigmoid. They converge faster and provide better generalization than Sigmoid.
- Sigmoid has slower convergence and a higher test loss, making it less suitable for this regression task.

In summary, with the Nesterov Momentum optimizer:

- GELU performs the best among the activation functions, showing the fastest convergence and achieving the lowest train and test loss, followed by Swish, ReLU and Sigmoid.
- Sigmoid has slower convergence and a higher test loss, making it less suitable for this regression task.

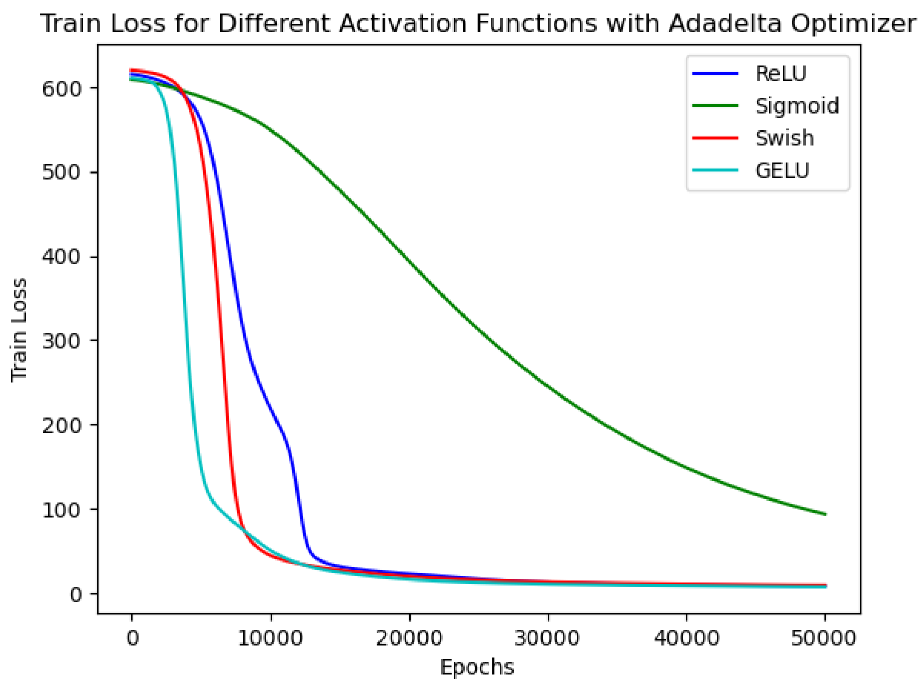

Figure 5: Train Loss for Different Activation Functions with Adadelta Optimizer

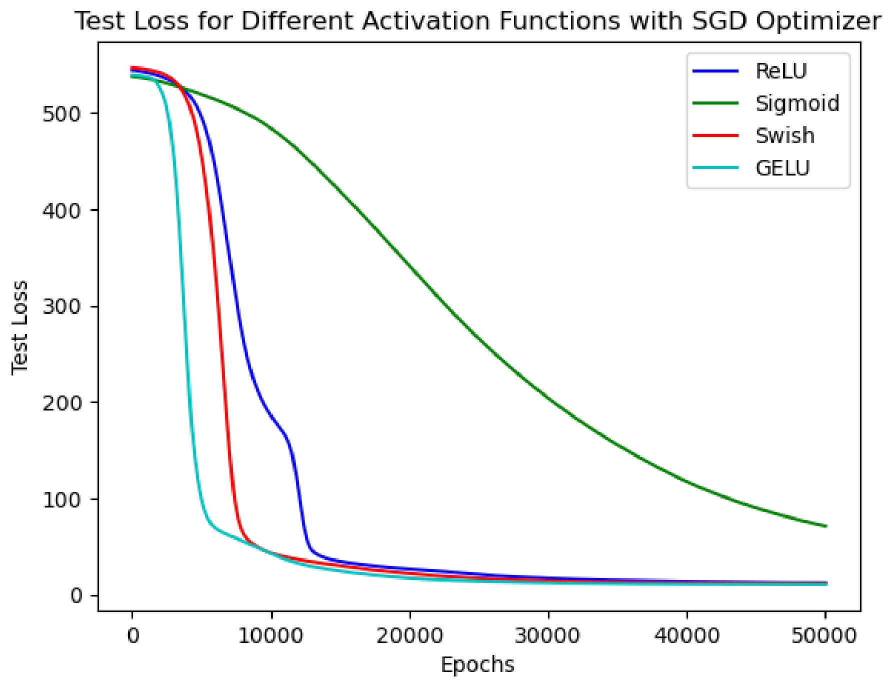

Figure 6: Test Loss for Different Activation Functions with Adadelta Optimizer

4. Discussion

We can explain the differences in performance based on the properties of each activation function:

- ReLU is a simple and fast activation function that can avoid the vanishing gradient problem, but it suffers from the dying ReLU problem when the inputs are negative. This may cause some neurons to become inactive and lose information during training.
- Sigmoid is a smooth and bounded activation function that can produce probabilistic outputs, but it suffers from the vanishing gradient problem when the inputs are large or small. This may slow down the learning process and make it harder to optimize deep networks.
- Swish is a smooth and non-monotonic activation function that belongs to the Swish family of functions. It has a self-gating property that allows it to adaptively adjust its output based on the input. This may help it balance between exploration and exploitation and achieve better performance than ReLU.
- GELU is another smooth and non-monotonic activation function that belongs to the Swish family of functions. It approximates the expected value of a stochastic binary gate that randomly allows or blocks the input. This may help it capture complex nonlinearities and achieve better performance than Swish.

5. Conclusion

In conclusion, we recommend using GELU or Swish as activation functions for regression tasks on the Boston Housing Dataset, as they can achieve better performance and faster convergence than ReLU and Sigmoid. GELU may be slightly preferable to Swish, but both are good choices.