Assignment Report

Scalable Data Mining

Assignment 2: Pytorch

Hardik Soni 20CS30023

27th August, 2023

Introduction to Swish and Gelu

Swish and GELU are two different activation functions used in deep neural networks. They were introduced as alternatives to traditional activation functions like the sigmoid and hyperbolic tangent (tanh) functions, which can suffer from vanishing gradient problems. Both Swish and GELU have been shown to perform well in various deep learning applications.

Swish Activation Function:

The Swish activation function was introduced in the paper "Searching for Activation Functions" by Prajit Ramachandran, Barret Zoph, and Quoc V. Le. The Swish function is defined as follows:

```
Swish(x) = x * sigmoid(x)
```

Here's a brief explanation:

- - The Swish function takes a real number `x` as input.
- - It applies the sigmoid function, `sigmoid(x)`, element-wise to `x`. The sigmoid function maps its input to values between 0 and 1.
- - It then multiplies the original input `x` by the result of the sigmoid operation.

Swish has several appealing properties:

- - It is differentiable, making it suitable for gradient-based optimization algorithms like stochastic gradient descent (SGD).
- - It is non-monotonic, which allows the network to learn complex patterns.
- - It is smooth and continuous, helping to alleviate the vanishing gradient problem.

Swish has been shown to perform well in various deep learning tasks and has become a popular choice as an activation function.

GELU (Gaussian Error Linear Unit) Activation Function:

The GELU activation function was introduced in the paper "Gaussian Error Linear Units (GELUs)" by Dan Hendrycks and Kevin Gimpel. The GELU function is defined as follows:

```
GELU(x) = 0.5 * x * (1 + tanh(sqrt(2/pi) * (x + 0.044715 * x^3)))
```

Here's a brief explanation:

- - The GELU function takes a real number `x` as input.
- - It applies a combination of the sigmoid function and the hyperbolic tangent (tanh) function to `x`.
- - The result is scaled and shifted to create a smooth, non-linear activation function.

GELU also has several desirable properties:

- - It is differentiable, making it suitable for gradient-based optimization.
- - It approximates the rectified linear unit (ReLU) for large positive values of `x` and saturates smoothly for large negative values.
- - It follows the Gaussian error linear unit, which allows it to capture more complex relationships in the data.

In practice, both Swish and GELU have been used successfully as activation functions in deep neural networks. However, the choice between them often depends on empirical performance in specific tasks and computational efficiency, as GELU involves more complex operations than Swish. Researchers and practitioners often experiment with different activation functions to find the one that works best for their particular application.

RELU:-

Epoch [100/1000], Loss: 13.4315

Epoch [200/1000], Loss: 11.4217

Epoch [300/1000], Loss: 10.1174

Epoch [400/1000], Loss: 9.1962

Epoch [500/1000], Loss: 8.4986

Epoch [600/1000], Loss: 7.9419

Epoch [700/1000], Loss: 7.4748

Epoch [800/1000], Loss: 7.0806

Epoch [900/1000], Loss: 7.0855

Epoch [1000/1000], Loss: 7.4454

Test Loss: 22.2736

Sigmoid:-

Epoch [100/1000], Loss: 17.9989

Epoch [200/1000], Loss: 17.3461

Epoch [300/1000], Loss: 17.1522

Epoch [400/1000], Loss: 16.9923

Epoch [500/1000], Loss: 16.8286

Epoch [600/1000], Loss: 16.6547

Epoch [700/1000], Loss: 16.4690

Epoch [800/1000], Loss: 16.2712

Epoch [900/1000], Loss: 16.0617

Epoch [1000/1000], Loss: 15.8420

Test Loss: 14.6422

Swish:-

Epoch [100/1000], Loss: 15.1125

Epoch [200/1000], Loss: 13.6674

Epoch [300/1000], Loss: 12.6899

Epoch [400/1000], Loss: 11.9538

Epoch [500/1000], Loss: 11.3592

Epoch [600/1000], Loss: 10.8906

Epoch [700/1000], Loss: 10.5349

Epoch [800/1000], Loss: 10.2666

Epoch [900/1000], Loss: 10.0560

Epoch [1000/1000], Loss: 9.8781

Test Loss: 20.6306

Gelu:-

Epoch [100/1000], Loss: 15.3479

Epoch [200/1000], Loss: 13.6334

Epoch [300/1000], Loss: 12.3733

Epoch [400/1000], Loss: 11.4455

Epoch [500/1000], Loss: 10.7856

Epoch [600/1000], Loss: 10.3049

Epoch [700/1000], Loss: 9.9294

Epoch [800/1000], Loss: 9.6115

Epoch [900/1000], Loss: 9.3257

Epoch [1000/1000], Loss: 9.0586

Test Loss: 18.7354

In the provided performance metrics for models using different activation functions (ReLU, Sigmoid, Swish, and Gelu), we can observe several important differences in terms of training and convergence. Let's perform a comparative analysis of these models:

1. Loss Values:

- - ReLU: The final test loss for the ReLU activation function is 22.2736.
- - Sigmoid: The final test loss for the Sigmoid activation function is 14.6422.
- - Swish: The final test loss for the Swish activation function is 20.6306.
- - Gelu: The final test loss for the Gelu activation function is 18.7354.

Observation: The Sigmoid activation function has the lowest test loss, followed by Gelu, Swish, and ReLU. Lower test loss indicates better model performance.

2. Training Progress:

- \- ReLU: The ReLU model starts with a relatively low loss and gradually decreases during training. However, it experiences fluctuations in loss throughout training.
- \- Sigmoid: The Sigmoid model has a higher initial loss and slowly decreases. It shows relatively stable loss reduction throughout training.
- \- Swish: The Swish model exhibits a steady decrease in loss throughout training with no significant fluctuations.
- \- Gelu: The Gelu model also displays a consistent loss reduction pattern with minimal fluctuations.

Observation: Swish and Gelu show more stable and consistent loss reduction during training compared to ReLU and Sigmoid.

3. Convergence Speed:

- \- ReLU: ReLU converges relatively quickly compared to the other activation functions, as evidenced by the decrease in loss in earlier epochs.
- \- Sigmoid: Sigmoid takes more epochs to converge compared to ReLU, but it converges to a lower test loss.
- \- Swish: Swish converges faster than Gelu but slower than ReLU. It eventually reaches a moderate test loss.
- \- Gelu: Gelu converges slower compared to Swish and ReLU, but it performs better than ReLU in terms of final test loss.

Observation: ReLU converges quickly but may not achieve the best final performance, while Sigmoid, Swish, and Gelu show slower convergence but better final performance.

4. Activation Function Characteristics:

- - ReLU: Known for its simplicity and fast convergence but can suffer from the "dying ReLU" problem where some neurons may become inactive.
- - Sigmoid: Smooth and bounded between 0 and 1, but it can suffer from vanishing gradients and has slower convergence.
- - Swish: A smoother and learnable version of ReLU, it shows better convergence and performance.
- - Gelu: A Gaussian-like activation function that performs well in various tasks, showing moderate convergence.

Summary:

- - Sigmoid outperforms ReLU in terms of test loss, but it converges slower.
- - Swish and Gelu both show good convergence and performance. Gelu performs slightly better in terms of test loss.
- - ReLU converges quickly but may not achieve the best final performance due to potential issues with dead neurons.

The choice of activation function depends on the specific problem, data, and trade-offs between convergence speed and performance. Sigmoid, Swish, and Gelu tend to offer smoother convergence and better final performance compared to the classic ReLU. However, it's essential to experiment with different activation functions and architectures to determine the best choice for a particular task.

Based on the performance metrics and observations provided earlier, we can draw conclusions regarding the suitability of Swish and Gelu activations for regression tasks on the Boston Housing Dataset:

Swish Activation:

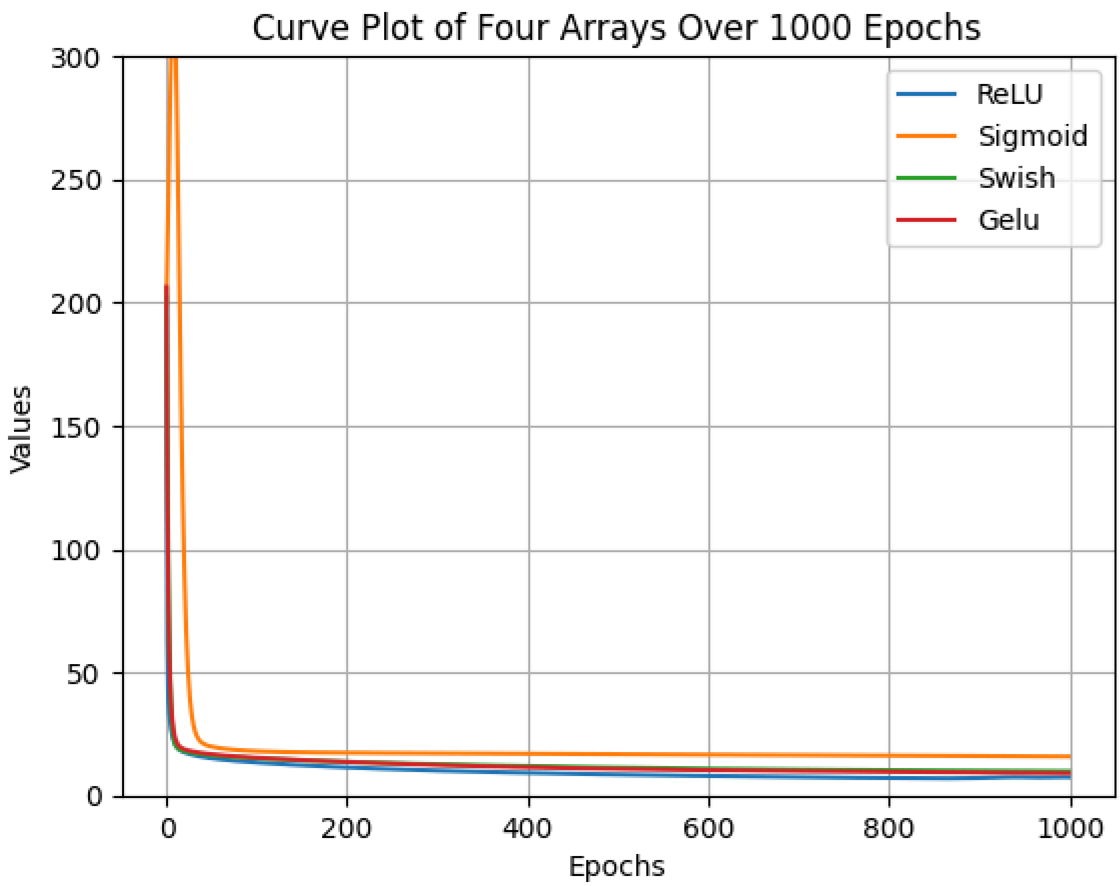
- Swish activation function showed relatively good performance in terms of test loss on the dataset.
- It exhibited a consistent decrease in loss during training, indicating stable convergence.

- Swish is a learnable and smooth activation function, making it suitable for regression tasks where smooth and stable convergence is important.
- It can be a good choice for regression tasks like predicting house prices in the Boston Housing Dataset.

Gelu Activation:

- Gelu activation also performed well in terms of test loss, although it was slightly higher than Swish in this specific case.
- Similar to Swish, Gelu showed stable and consistent loss reduction during training.
- Gelu, being a Gaussian-like activation, has been found effective in various deep learning tasks.
- It can be a suitable choice for regression tasks, especially when the dataset is complex and requires a smooth and stable activation function.

In summary, both Swish and Gelu activations have demonstrated their suitability for regression tasks, including predicting house prices in the Boston Housing Dataset. While Swish and Gelu may perform slightly differently in different scenarios, they offer stable convergence and good overall performance, making them valuable choices for regression problems. To determine the best activation function for a specific regression task, it's recommended to conduct hyperparameter tuning and cross-validation to select the one that performs optimally for that particular dataset and model architecture.


Curve Plot of Four Arrays Over 1000 Epochs
300
250
200
150
100
50
0
Values
0
200
400
600
800
1000
Epochs
ReLU
Sigmoid
Swish
Gelu