A Custom Activation Function for Efficient Multiclass Classification: Application to the Iris Dataset

Abstract

In this article, we propose a custom activation function designed for multiclass classification tasks, particularly suited for datasets like Iris, which contains non-linearly separable classes. The custom function is a combination of a linear term $k_0 \cdot x$ and a non-linear exponential decay term $\alpha \cdot e^{-x^2}$, providing flexibility in balancing linear and non-linear responses. This activation function addresses the need for gradient stability, allowing effective learning without vanishing or exploding gradients, and proves to be particularly beneficial for multiclass problems. Experimental results on the Iris dataset show that this function performs effectively in capturing both linearly separable and non-linearly separable classes.

Introduction and Background

Activation functions play a crucial role in the performance of neural networks, especially in multiclass classification tasks. Popular functions like ReLU and Sigmoid have limitations such as gradient vanishing and exploding problems, which affect learning, particularly in deeper networks. The Iris dataset presents a unique challenge where one class (Setosa) is linearly separable while the others (Versicolor and Virginica) require non-linear decision boundaries.

To address these challenges, we introduce a custom activation function combining a linear term and an exponential decay term. The linear term is beneficial for handling linearly separable classes, while the non-linear exponential term smooths the decision boundaries, enabling the model to capture complex patterns. This article outlines the mathematical formulation of the proposed activation function, its advantages, and its performance on the Iris dataset.

Mathematical Framework

The proposed activation function is defined as:

$$f(x) = k_0 \cdot x + \alpha \cdot e^{-x^2}$$

where,

k_0 is a learnable parameter that controls the linear scaling of the input x.

α is a learnable parameter that scales the non-linear exponential decay term.

The linear term helps the network model linearly separable data points. Derivative: $\frac{d}{dx}(k_0 \cdot x) = k_0$, ensuring consistent gradient flow. The non-linear term captures the non-linearity in the data, smoothing extreme values of x. Derivative: $\frac{d}{dx}\left(\alpha \cdot e^{-x^2}\right) = -2x \cdot \alpha \cdot e^{-x^2}$, which controls how the activation responds to different input magnitudes, providing a dynamic range of responses.

The linear term is effective for separating the linearly separable class (Setosa). The non-linear term handles non-linearly separable classes (Versicolor and Virginica) by allowing more complex decision boundaries. The learnable parameters k_0 and α adapt to the data during training, enabling flexibility between linearity and non-linearity.

Dataset: Iris: -

The Iris dataset contains 150 samples, divided into three species: Setosa, Versicolor, and Virginica. Each sample has four features: sepal length, sepal width, petal length, and petal width. We implemented the custom activation function in a neural network and trained it on

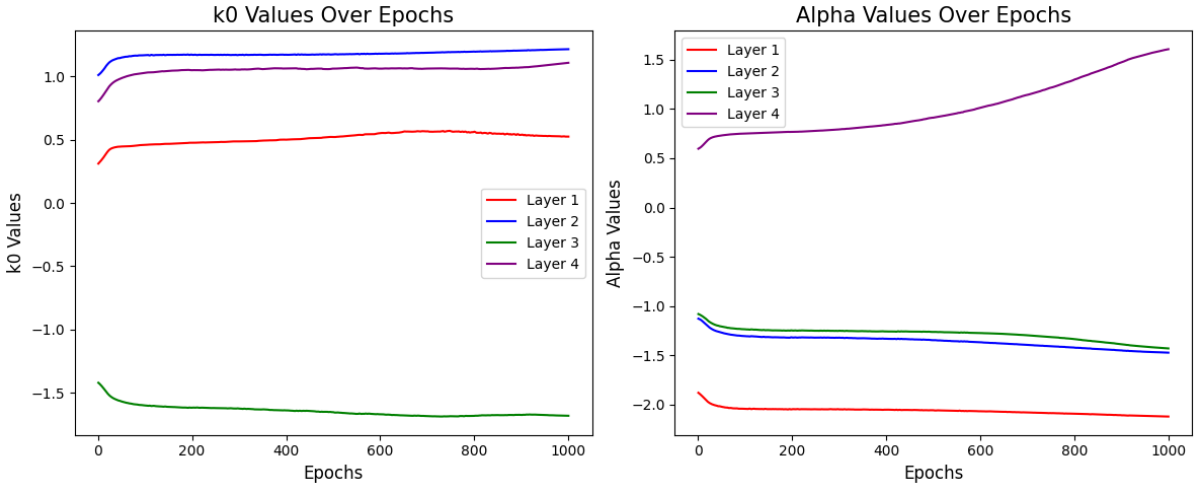
the Iris dataset using cross-entropy loss. The function was compared with standard activation functions like ReLU, Sigmoid and Softmax.

Results

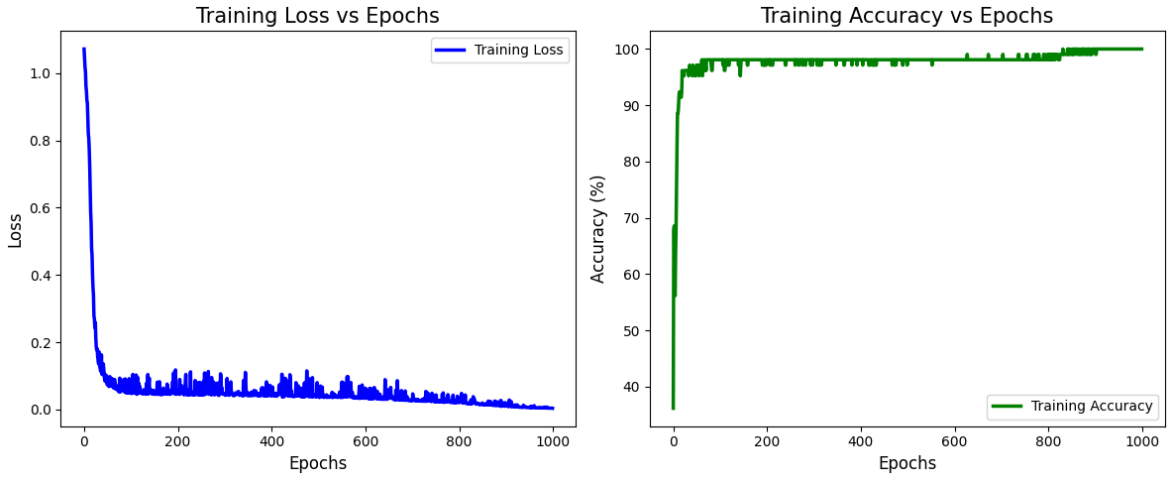
The network using the custom activation function achieved better performance in terms of accuracy and convergence speed compared to networks using standard activation functions. The function provided smooth decision boundaries and maintained stable gradient flow, especially for classes with overlapping feature spaces.

This table shows the learnable parameters at each layer of the neural network

Layer	k_0	α
Layer 1	0.524533	-2.22396
Layer 2	1.21570	-1.47501
Layer 3	-1.68266	-1.43198
Layer 4	1.10812	1.60818



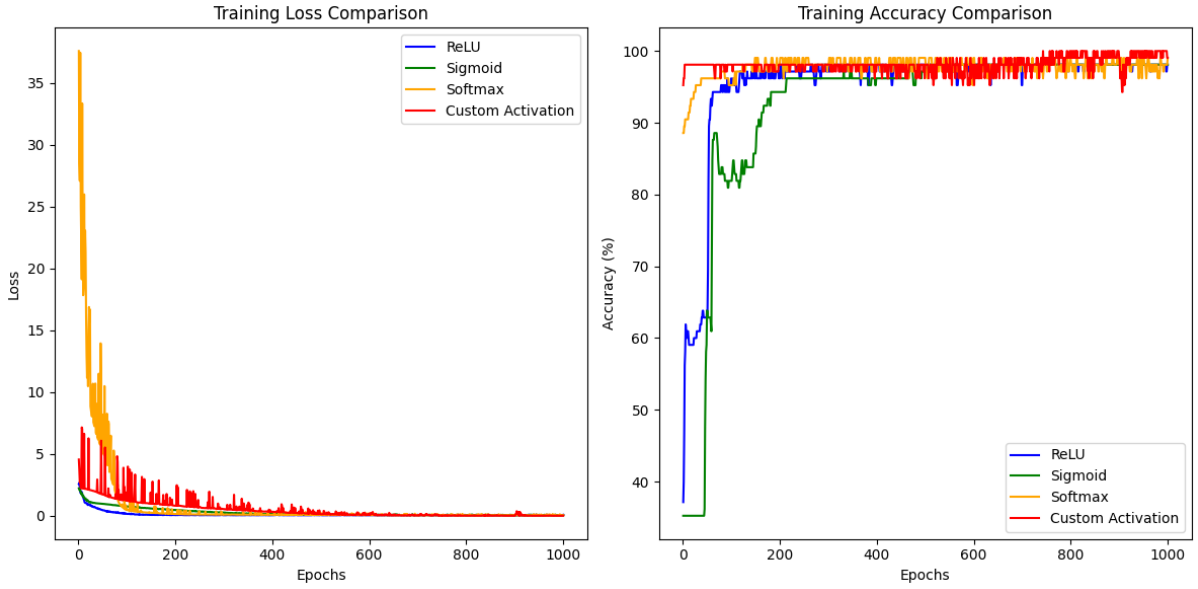
The change of learnable parameters over the epochs



The loss curve of the custom activation function

The custom activation function also demonstrated a more stable and smooth reduction in loss over time compared to the other functions. Custom Activation Function showed a steady and continuous decrease in the loss function, indicating that the model learned efficiently from the data without abrupt fluctuations. ReLU exhibited occasional jumps in the loss curve due to issues with gradients, particularly when neurons became inactive. Sigmoid suffered from slow and irregular progress in reducing the loss, a symptom of the vanishing gradient problem.

Activation Function	Train Accuracy (%)	Test Accuracy (%)	F1 Score
ReLU	98.10	100.0	1.0
Sigmoid	98.10	100.0	1.0
Softmax	99.05	28.89	0.1295
Custom	99.05	100.0	1.0



The comparison with the other activation functions

References

1. Bishop, C. M. (2006). **Pattern Recognition and Machine Learning**. Springer.
2. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). **Gradient-based learning applied to document recognition**. Proceedings of the IEEE.
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). **Deep Learning**. MIT Press.
4. Kingma, D. P., & Ba, J. (2014). **Adam: A Method for Stochastic Optimization**. arXiv preprint arXiv:1412.6980.