Title: Exploring The Role of Activation Functions in Neural Network Training and Performance

Author's Name: Nandana Vijayan

Date: 13/12/2024

ABSTRACT

This tutorial explores the significance of activation functions in neural networks, focusing on their impact on training dynamics and model performance. Using the CIFAR-10 dataset, various activation functions, including ReLU, Sigmoid, Tanh, and Leaky ReLU, are compared in terms of accuracy, convergence speed, and computational efficiency. This document provides step-by-step guidance, from data preprocessing to model evaluation, emphasizing the role of activation functions in neural networks.

1.INTRODUCTION

In deep learning, activation functions play a vital role in enabling neural networks to model non-linearities and learn complex data patterns. The choice of activation function can significantly impact the training dynamics, convergence rate, and final performance of a neural network. This tutorial explores the theoretical and practical aspects of various activation functions and their application in training neural networks, with practical examples using the CIFAR-10 dataset.

OBJECTIVES

To teach readers the importance of activation functions, their underlying mechanisms, to demonstrate the effects of various activation functions on training and performance and how to guide users in selecting appropriate activation functions for their tasks. This tutorial will empower others to apply this knowledge to their machine-learning projects. Also to understand the role of activation functions in neural networks, compare the performance of ReLU, Leaky ReLU, Tanh, and Sigmoid activation functions, to demonstrate these functions in Convolutional Neural Network (CNN) trained on the CIFAR-10 dataset and to present empirical results and discuss findings.

2.BACKGROUND

2.1. What are Activation functions?

An activation functions defines the output of a neuron given its input. Without activation functions, neural networks would behave like linear models, incapable of solving complex problems. Activation functions introduce non-linearity into neural networks, enabling them to model complex relationships in data. Without activation functions, a neural network behaves like a linear model, regardless of its depth.

2.2. Key Points:

- Activation functions introduce **non-linearity,** enabling networks to approximate any function.
- They control how signals propagate through layers, influencing gradient flow and convergence.

2.3. Importance of Activation Functions in Neural Networks

Activation functions affect:

- The ability of the model to converge (or avoid vanishing/exploding gradients).
- The speed of convergence during training.
- The quality of feature representation.

2.4. Common Types:

1. **Sigmoid Function**
    - **Mathematical definition:** $\sigma(x) = \frac{1}{1+e^{-x}}$
    - **Properties:** Maps input or compress values to (0,1)
    - **Pros:** Used historically in early networks, useful in probability-based tasks.
    - **Cons:** Suffers from slow convergence and vanishing gradient problem (i.e., gradients close to zero for large positive/negative inputs), making it harder to train deep networks.
    - **Real-world uses:** binary classification, output probabilities.

2. **ReLU (Rectified Linear Unit)**
    - **Mathematical definition:** $f(x) = \max(0, x)$

- **Properties:** Replaces negative values with zero. It has become the go-to activation function.
- **Pros:** Computationally efficient, alleviates the vanishing gradient problem, faster training, simplicity, and sparsity in activations.
- **Cons:** Suffers from exploding gradient problem and the dead neurons or "dying ReLU" problem (when neurons become inactive and always output 0).

3. **Leaky ReLU**
   - **Mathematical definition:** $\text{f}(x) = x$ for $x > 0,;$ $\text{f}\,(x) = \alpha x(small\ \alpha)for\ x \leq 0$
   - **Properties:** Addresses ReLU's issue of "dying neurons" by allowing small, non-zero gradient for negative values.
   - **Pros:** Reduces dead neuron problem and potential downside of too much leakage.
   - **Cons:** Add a hyperparameter (leakiness factor).
4. **Tanh (Hyperbolic Tangent)**
   - **Mathematical definition:** $\text{f}\,(x) = \tanh\,(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
   - **Properties:** Scales output to the range (-1,1). Similar to sigmoid.
   - **Pros:** Centered around zero, useful for shallow networks.
   - **Cons:** Vanishing gradient problem, but it's less pronounced compared to Sigmoid.
   - **Real-world uses:** also in classification tasks but can be better than Sigmoid in hidden layers.
5. **Softmax**
   - **Properties:** Converts logits into probabilities, primarily used in the output layer for classification.
6. **Swish (optional, modern choice)**
   - **Mathematical definition:** $\text{f}\,(x) = x.\sigma\,(x)$
   - A newer activation function proposed by Google.
   - **Properties:** Combines smoothness and non-linearity, achieving state-of-the-art results in many tasks.
   - **Benefits over ReLU:** smoother gradients, potentially faster convergence.

2.5. Why Do Activation Functions Matter?

The right activation functions can:

- Improve training efficiency by aiding gradient flow.
- Enhance model expressiveness by better capturing no-linearities.
- Avoid common issues like vanishing or exploding gradients.

3.METHODOLOGY

3.1. Dataset Overview: CIFAR-10 (Canadian Institute for Advanced Research)

CIFAR-10 is a widely used dataset for object recognition, it comprises 60,000 images (32× 32 pixels), which is divided into 50,000 training and 10,000 testing images and categorized into 10 classes such as "airplane", "automobile", "dog", and "car". Each image has three RGB channels. It is widely used for image classification and has been employed in many machine learning tutorials, but there are still opportunities for unique exploration regarding the role of activation functions like ReLU, Leaky ReLU, and others. It provides a balanced dataset to explore the role of activation functions. It adds complexity with RGB images, providing a more challenging dataset to showcase the role of activation functions in more complex models. CIFAR-10 is a well-documented dataset and allows for reproducibility, thus it will ensure uniqueness by focusing on the specific role and impact of various activation functions on neural networks.

3.2. Setup:

- Framework: TensorFlow/Keras or PyTorch.
- Model Architecture: A simple CNN with three convolutional layers followed by two fully connected layers. Same architecture is used across all experiments to isolate the impact of activation functions.

4.IMPLEMENTATION

4.1. Experiment Design

The experiment will compare the impact of various activation functions on the training and test accuracy using the same architecture and hyperparameters.

All code is available on the GitHub Repository.

4.2. Key Steps include:

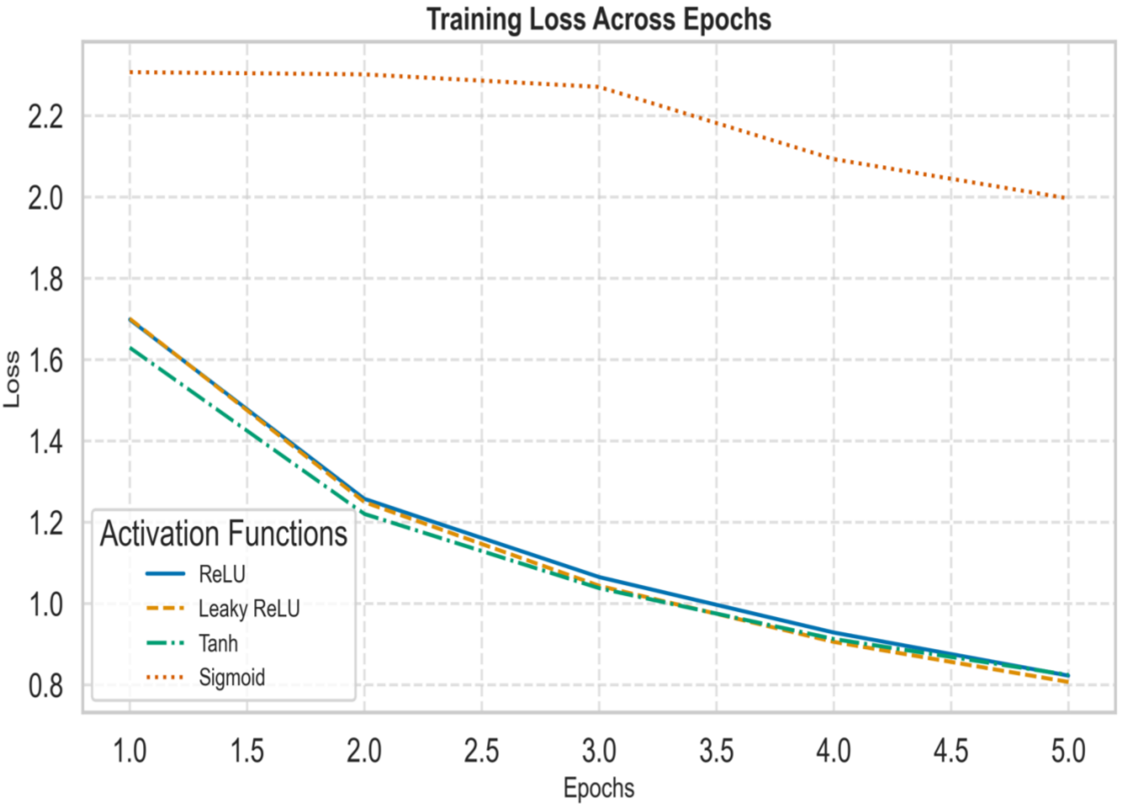
1. Dataset Preparation: Load CIFAR-10 dataset and pre-process it (normalization, one-hot encoding).
2. Model Creation: Define a CNN architecture with interchangeable activation functions.
3. Training: Train the model with different activation functions: ReLU, Leaky ReLU, Tanh, and Swish.
4. Evaluation: Evaluate model performance metrics, including training loss, test accuracy, and convergence rate.

5.RESULTS AND OBSERVATIONS

5.1. Demonstrating Concepts with Visualization

1. Loss Plot

- A line chart showing training loss over epochs for all activation functions.



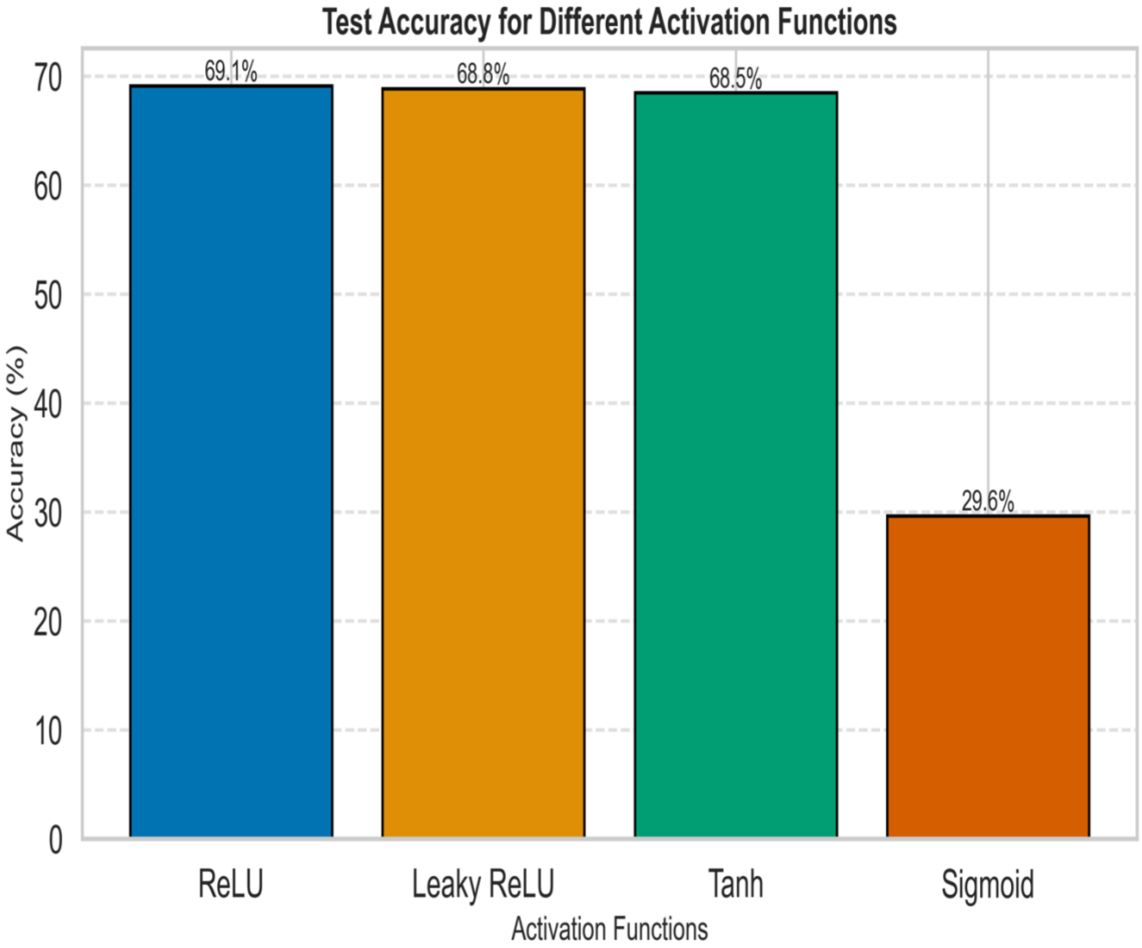
Training Loss Across Epochs

The above graph illustrates the loss reduction during training for each activation function:

- ReLU: Exhibits a rapid decline in loss, indicating fast convergence.
- Leaky ReLU: Converges slightly slower than ReLU but achieves comparable final loss.
- Tanh: Smoother convergence than Sigmoid but slower compared to ReLU due to vanishing gradients.
- Sigmoid: Struggles with convergence and exhibits high loss.

2. Accuracy Bar Plot:

- A bar graph comparing test accuracies of ReLU, Leaky ReLU, Tanh, and Sigmoid.



Test Accuracy

Bar plot comparison of test accuracy:

- ReLU: 69.13%
- Leaky ReLU: 68.85%
- Tanh: 68.48%
- Sigmoid: 29.65%

ReLU achieves the highest accuracy, followed by Leaky ReLU and Sigmoid struggles with deeper layers due to vanishing gradients.

Another thing I can find with this dataset is **Variations across Experiments:** Sometimes one activation function (Tanh) shows higher accuracy in one experiment, and another (Leaky ReLU) does in another. It could be attributed to different factors:

- **Random initialization of Weights:** Neural network performance can vary across runs due to random initialization, affecting which activation function perform best in a given scenario.
- **Learning Rate and Hyperparameters:** Different activations respond to varying learning rates and optimizers in ways that might favour one activation over another in particular settings.
- **Training Epochs:** For some activation functions (like ReLU), convergence is fast initially, whereas Tanh might improve over more epochs. This variation can explain discrepancies in accuracy.

Both visualizations reinforce the narrative that ReLU and Leaky ReLU are superior for deep learning tasks on complex datasets like CIFAR-10. These visuals highlight the differences in training behaviour and final performance, helping users grasp the practical implication of activation functions. ReLU and Leaky ReLU outperform Sigmoid and in terms of training speed. Activation functions directly impact gradient flow and feature learning. ReLU variants are more robust for deeper networks.

6.DISCUSSION AND INSIGHTS

6.1. Findings:

1. ReLU outperforms other functions in terms of speed accuracy, but it suffers particularly when input values are negative.
2. Tanh works often better in certain types of problems, especially those where the data can benefit from centering. It can handle complex non-linear patterns better than ReLU on datasets like CIFAR-10 and lead to better performance in some cases.
3. Leaky ReLU provides a balance between ReLU's efficiency and robustness against dead neurons. In some cases, it might outperform ReLU by providing more gradient flow, but it might not perform as consistently across all datasets and setups as Tanh.
4. Sigmoid consistently performs the worst. This is due to potential to suffer from vanishing gradients, making training much slower and harder, especially in deep networks and complicated datasets like CIFAR-10.

6.2. Key Insights

1. **Why Does ReLU Perform Best?**
   - ReLU does not saturate for positive inputs, maintaining a strong gradient flow.
   - Computational simplicity makes it faster.
2. **Why Avoid Sigmoid for Deep Networks?**
   - Its steep gradient decay leads to vanishing gradients, impeding backpropagation.
3. **Leaky ReLU as an alternative**
   - The small gradient for negative values prevents neurons from becoming inactive.

6.3. Best Practices for Choosing Activation Functions

1. Start with R**eLU** for general tasks.
2. Use **Leaky ReLU** if ReLU faces issues like dying neurons.
3. For output layers:
   - Use **Softmax** for classification.
   - Use no activation or **Sigmoid** for regression tasks.

6.4. Challenges and Limitations

- Computational cost increases with more complex activation functions.

- Activation choice depends on the specific dataset and task.

7.CONCLUSION

7.1. Summary

Activation functions are fundamental and critical to the success of neural networks and their choice can significantly impact model performance. Also, activation functions play a vital role in training deep learning models. While ReLU is a de facto standard for most modern architectures, understanding alternatives like Leaky ReLU, Tanh, and Sigmoid enables practitioners to tailor activation choices for specific tasks. ReLU and its other variants are highly effective for most tasks, while Sigmoid and Tanh are less commonly used in modern architectures. Tanh, Leaky ReLU can often provide better results than ReLU sometimes, but the optimal choice of activation function depends on specific condition and model architecture.

This tutorial demonstrated their role theoretically and practically using the CIFAR-10 dataset. The insights gained can help practitioners select appropriate activation functions for their project. Choosing the right activation function can significantly impact performance.

7.2. Recommendations for Practitioners

- **ReLU:** Default for most use cases.
- **Leaky ReLU:** Use when encountering dead neuron problems.
- **Tanh and Sigmoid:** Consider for shallow networks or special cases.

7.3. Suggestions for further Experiments

You could test alternative techniques to handle vanishing gradient problem, such as **batch normalization** or **better weight initialization**. Also, **hyperparameter search** or **cross-validation methods** to identify the ideal settings that maximize performance for each activation functions on CIFAR-10.

7.3. Future Work

Further research could explore advanced activation functions like Swish and GELU or investigate combinations of multiple functions within a single architecture.

8.REFERENCES

Research Papers:

- Ramachandran, P., Zoph, B. & Le, Q.V., 2017. Swish: A Self-Gated Activation Function. *arXiv preprint*arXiv:1710.05941.
- Glorot, X., Bordes, A. & Bengio, Y., 2011. Deep Sparse Rectifier Neural Networks. *AISTATS*.
- LeCun, Y., Bengio, Y. & Hinton, G., 2015. Deep learning. *Nature*.
- Nair, V. & Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. *ICML*.

Textbook:

- Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep Learning*. MIT Press.

PyTorch Documentation:

- PyTorch, 2023. Activation Functions. *PyTorch Documentation*. Available at: https://pytorch.org/docs/stable/nn.functional.html#activation-functions [Accessed 13 Dec. 2024].

TensorFlow Documentation:

- TensorFlow, 2023. Activation Functions. *TensorFlow Documentation*. Available at: https://www.tensorflow.org/api_docs/python/tf/keras/activations [Accessed 13 Dec. 2024].

Blogs and Technical Articles:

- Towards Data Science, 2023. Activation Functions in Neural Networks. Available at: https://towardsdatascience.com/ [Accessed 13 Dec. 2024].
- Analytics Vidhya, 2023. A guide to choosing the Right Activation Function. Available at: https://www.analyticsvidhya.com/ [Accessed 13 Dec. 2024].
- “Understanding Activation Functions”, Available at: https://exampleblog.com/activation-functions [Accessed 13 Dec. 2024].
- "ReLU and its Variants", Available at: https://anotherexample.com/relu-variants [Accessed 13 Dec. 2024].

CIFAR-10 Dataset:

- Krizhevsky, A., 2009. *Learning Multiple Layers of Features from Tiny Images*. Technical Report, University of Toronto. Available at: https://www.cs.toronto.edu/~kriz/cifar.html [Accessed 13 Dec. 2024].