Neural Networks as Nearest Neighbor: Abs Activation Functions for Learning Gaussian Clusters

Alan Oursland

June 29, 2024

Abstract

1 Introduction

Neural networks have revolutionized machine learning, achieving unprecedented performance across a wide range of tasks. At the heart of these networks lies a crucial component: the activation function. The Rectified Linear Unit (ReLU) has emerged as the de facto standard activation function due to its simplicity and effectiveness in mitigating the vanishing gradient problem. However, the quest for alternative activation functions that can offer improved performance or interpretability remains an active area of research.

Before delving into neural networks, it's important to consider one of the fundamental classification methods in machine learning: k-Nearest Neighbors (kNN). This simple yet effective algorithm classifies data points based on the majority class of their k nearest neighbors. However, kNN faces limitations, particularly with high-dimensional data and when the underlying data structure is complex.

To address these limitations, researchers have developed advanced kNN methods that incorporate Gaussian representations and metric learning. These approaches, such as Locally Adaptive Metric Nearest-Neighbor (LMNN), Adaptive Nearest Neighbor Classification, and Discriminant Adaptive Nearest Neighbor Classification (DANN), aim to improve classification by adapting to local data structures.

This paper introduces a novel perspective that bridges the gap between these advanced kNN methods and neural networks by examining the absolute value function (Abs) as an alternative to ReLU. Our investigation is motivated by a key theoretical insight: there exists a homomorphism between Gaussian clusters and linear separators with absolute value activation functions. This relationship suggests that neural networks utilizing Abs activations can implicitly learn to represent Gaussian clusters, potentially offering enhanced interpretability and performance in certain scenarios.

The primary objectives of this study are:

1. To demonstrate the effectiveness of the Abs activation function compared to ReLU across various datasets and model architectures.

2. To explore the interpretability benefits of Abs activations, particularly in the context of Gaussian cluster representation.

3. To investigate the practical implications of the theoretical homomorphism between Gaussians and linear separators with Abs activations.

4. To examine the connections between advanced kNN methods and neural networks with Abs activations.

This research encompasses a comprehensive empirical study using diverse datasets, including image classification (MNIST, CIFAR-10, CIFAR-100) and tabular data (UCI Adult, Wine Quality, Iris). I employ two main types of neural network architectures: Convolutional Neural Networks (CNNs) for image data and Multilayer Perceptrons (MLPs) for tabular data. This selection ensures a thorough comparison between Abs and ReLU activations across different data types and model complexities, while also exploring scenarios where clustering capabilities may be particularly relevant.

By bridging the gap between probabilistic models (Gaussians) and discriminative models (linear separators), this work offers a new perspective on how neural networks learn and represent data distributions. The findings have the potential to influence future neural network designs, particularly in applications where model interpretability is crucial.

The remainder of this paper is organized as follows: Section 2 presents the theoretical foundation underlying our work, Section 3 describes our methodology and experimental setup, Section 4 presents our results, Section 5 discusses the implications of our findings, and Section 6 concludes with a summary and directions for future research.

2 Theoretical Foundation

The core of our research lies in a novel discovery: a homomorphism between Gaussian clusters and linear separators with absolute value activation functions. This theoretical foundation explains how neural networks utilizing absolute value activations can implicitly learn to represent Gaussian clusters, offering a new perspective on the learning capabilities of these networks.

2.1 k-Nearest Neighbors (kNN)

We begin by examining the k-Nearest Neighbors algorithm, a fundamental classification method in machine learning. In its basic form, kNN classifies a data point based on the majority class of its k nearest neighbors in the feature space. This approach implicitly assumes that the data points form spherical clusters in the feature space, which can be viewed as spherical Gaussian distributions.

2.2 Advanced kNN Methods

To overcome the limitations of basic kNN, several advanced methods have been developed:

- Locally Adaptive Metric Nearest-Neighbor (LMNN): This method learns a Mahalanobis distance metric to improve kNN classification. By learning a global linear transformation of the feature space, LMNN can adapt to the overall structure of the data.

- Adaptive Nearest Neighbor Classification: This approach adapts local metrics based on the covariance structure of nearby points. It allows for more flexible decision boundaries that can better capture the local structure of the data.

- Discriminant Adaptive Nearest Neighbor Classification (DANN): DANN adapts the metric locally to the structure of the data to improve classification. It combines ideas from discriminant analysis with adaptive nearest neighbor methods.

These advanced methods implicitly or explicitly model the data using Gaussian distributions, moving beyond the spherical assumption of basic kNN.

2.3 From kNN to Gaussian Clusters

As we transition from basic kNN to these more sophisticated methods, we move from spherical clusters to full covariance Gaussian clusters. However, using full covariance matrices can be computationally expensive and may lead to issues with singularity, especially in high-dimensional spaces.

To address these challenges, we turn to Principal Component Analysis (PCA). PCA allows us to represent the Gaussian clusters more efficiently by focusing on the principal components that capture the most variance in the data.

2.4 From PCA to Mahalanobis Distance

Our analysis begins with Principal Component Analysis (PCA) applied to a multivariate Gaussian distribution. We focus on individual one-dimensional Gaussians along the principal component axes in the original data space. This approach allows us to decompose the complex multivariate structure into more manageable components.

For points projected onto these one-dimensional Gaussians, we derive the Mahalanobis distance formula:

$$D = \frac{|v^T(x - \mu)|}{\sqrt{\lambda}} \tag{1}$$

Where:

- v is the eigenvector corresponding to a principal component
- x is the data point
- μ is the mean of the Gaussian
- λ is the eigenvalue (variance) along the principal component

2.5 Transformation to Linear Separators

The key insight emerges when we rearrange and simplify the Mahalanobis distance formula:

$$y = |Wx + b| \tag{2}$$

Where:

- $W = \frac{v^T}{\sqrt{\lambda}}$
- $b = -\frac{v^T}{\sqrt{\lambda}}\mu$

This transformed equation reveals a striking similarity to a linear separator with an absolute value activation function:

$$y = Abs(Wx + b) \tag{3}$$

2.6 Comparison with ReLU Activation

The form we've derived is analogous to the widely-used ReLU-based linear separator in neural networks:

$$y = ReLU(Wx + b) \tag{4}$$

This parallel underscores the potential of absolute value activations as an alternative to ReLU, with the added benefit of implicit Gaussian cluster representation.

2.7 Implications and Significance

The homomorphism we've uncovered has several important implications:

1. Gaussian clusters can be effectively represented as linear separators with absolute value activations.
2. Neural networks employing absolute value activations have the capacity to implicitly learn and represent Gaussian clusters in the data.
3. The decision boundaries formed by these separators correspond to surfaces of equal Mahalanobis distance from the Gaussian means, providing a geometrically interpretable structure to the learned representations.
4. This approach unifies concepts from kNN, Gaussian mixture models, and neural networks, providing a coherent framework for understanding classification and representation learning.

This theoretical foundation bridges the gap between probabilistic models (Gaussians) and discriminative models (linear separators). It offers a novel perspective on how neural networks can learn and represent data distributions, potentially leading to more interpretable models and improved performance in certain scenarios. In the subsequent sections, we will explore the practical implications of this theoretical insight through a series of experiments across various datasets and model architectures.

3 Methodology

Our study employs a comprehensive experimental approach to compare the performance of neural networks using Rectified Linear Unit (ReLU) and Absolute Value (Abs) activation functions across various datasets and model architectures.

3.1 Datasets and Model Architectures

We conduct experiments on six diverse datasets, encompassing both image classification and tabular data tasks:

- Image Classification:

  - MNIST: Handwritten digit recognition (28x28 grayscale images, 10 classes)
  - CIFAR-10: Object recognition (32x32 color images, 10 classes)
  - CIFAR-100: Fine-grained object recognition (32x32 color images, 100 classes)

- Tabular Data:

  - UCI Adult: Income prediction based on census data
  - Wine Quality: Wine quality rating prediction
  - Iris: Flower species classification

For each dataset, we employ the following model architectures:

- MNIST: LeNet-5 Convolutional Neural Network (CNN)
- CIFAR-10 and CIFAR-100: ResNet-18 CNN
- UCI Adult, Wine Quality, and Iris: Multilayer Perceptron (MLP)

For the MLP models used on tabular data, we propose a 3-layer architecture with hidden layer sizes adjusted based on the input dimensionality of each dataset. The specific configurations for these MLPs will be determined during the implementation phase.

3.2 Experimental Setup

Each experiment is conducted using PyTorch and runs on an NVIDIA GeForce RTX 3080 Ti GPU. We perform multiple runs (typically 5) for each experiment to ensure statistical significance. The hyperparameters and configuration details for each experiment are stored in JSON files, allowing for easy replication and modification of the experiments. For CIFAR-10 and CIFAR-100 datasets, we employ data augmentation techniques as described in the original ResNet paper, including random cropping and horizontal flipping.

3.3 Training and Evaluation

We use the pre-defined training and test splits provided by each dataset. The models are trained using stochastic gradient descent (SGD) with momentum. Learning rates, batch sizes, and other hyperparameters are specified in the configuration files for each experiment.

The primary evaluation metric is classification accuracy on the test set. We also implement an error overlap analysis to assess the potential for ensemble models constructed from individual ReLU and Abs models.

3.4 Comparative Analysis

Our analysis focuses on several key aspects:

1. Performance Comparison: We compare the test accuracy of models using ReLU and Abs activations across all datasets.
2. Statistical Significance: We use appropriate statistical tests (e.g., t-tests) to determine if the differences in performance between ReLU and Abs models are statistically significant.

3. Error Overlap Analysis: We examine the extent to which errors made by ReLU and Abs models overlap, providing insights into the potential benefits of ensemble methods combining both activation functions.

4. Training Dynamics: We analyze the training curves to compare convergence rates and stability between ReLU and Abs models.

3.5 Interpretability Analysis

[TBD: Specific techniques for interpretability analysis will be determined and implemented in future stages of the research.]

3.6 Reproducibility

To ensure reproducibility, we provide detailed configuration files, model architectures, and random seeds used in our experiments. All code and configuration files are made available in a public repository, allowing other researchers to replicate our results and build upon our work.

4 Results

Our experiments compare the performance of neural networks using Rectified Linear Unit (ReLU) and Absolute Value (Abs) activation functions across various datasets and model architectures. This section presents our findings, organized by dataset type and specific tasks.

4.1 Image Classification Tasks

4.1.1 MNIST

We conducted experiments on the MNIST dataset using the LeNet-5 architecture. The results from 5 runs for each activation function are as follows:

- ReLU: Average Accuracy: 98.57
- Abs: Average Accuracy: 98.76
- Statistical Significance: t-statistic: 4.1216, p-value: 0.0033

The p-value ¡ 0.05 indicates a statistically significant difference, with Abs outperforming ReLU on this dataset.

4.1.2 CIFAR-10

[TBD: Add results for CIFAR-10 using ResNet-18 architecture. Include average accuracy, loss, and statistical significance.]

4.1.3 CIFAR-100

[TBD: Add results for CIFAR-100 using ResNet-18 architecture. Include average accuracy, loss, and statistical significance.]

4.2 Tabular Data Tasks

4.2.1 UCI Adult Dataset

[TBD: Add results for UCI Adult dataset using MLP architecture. Include average accuracy, loss, and statistical significance.]

4.2.2 Wine Quality Dataset

[TBD: Add results for Wine Quality dataset using MLP architecture. Include average accuracy, loss, and statistical significance.]

4.2.3 Iris Dataset

[TBD: Add results for Iris dataset using MLP architecture. Include average accuracy, loss, and statistical significance.]

4.3 Error Overlap Analysis

[TBD: Present results of error overlap analysis between ReLU and Abs models for each dataset. Discuss implications for potential ensemble methods.]

4.4 Training Dynamics

[TBD: Analyze and compare training curves for ReLU and Abs models across datasets. Discuss convergence rates and stability.]

4.5 Interpretability Analysis

[TBD: Present results of interpretability analysis. This may include visualization of learned features, analysis of decision boundaries, or other techniques that demonstrate how Abs activation allows for better interpretation of Gaussian clusters in the data.]

4.6 Summary of Findings

Based on the preliminary results from the MNIST dataset, we observe that the Abs activation function shows promise as an alternative to ReLU. The statistically significant improvement in accuracy suggests that Abs may offer benefits in terms of model performance, at least for certain types of tasks.

[TBD: Summarize overall trends across all datasets once results are available. Discuss whether the performance benefits of Abs generalize across different types of data and model architectures.]

5 Discussion

6 Conclusion

References