Customized Layer Architectures: Part I of III A Comprehensive Analysis of Layer Performance with Finance Sentiment Analysis

Preprint · January 2024

DOI: 10.13140/RG.2.2.26265.19041

Jamell Alvah Samuels
Imperial College London

Customized Layer Architectures: Part I A Comprehensive Analysis of Layer Performance with Finance Sentiment Analysis

Jamell Ivor Samuels

Abstract—**Sentiment analysis, a subfield of natural language processing (NLP), has witnessed significant advancements in recent years. However, the inherent complexities and nuances of human language pose challenges for achieving high accuracy in sentiment classification tasks. This three-part research series aims to contribute to the field by introducing novel customized layer architectures for sentiment analysis. Part I focuses on conducting an exhaustive survey of existing methodologies, identifying gaps in current approaches, and establishing the foundational framework for our proposed custom layer models.**

Introduction

The field of deep learning has witnessed remarkable advancements in recent years, with neural networks becoming increasingly sophisticated and capable of handling complex tasks. Central to the success of these networks are the layers that compose them, forming the backbone of the model architecture. Traditional layer architectures, such as convolutional layers for image processing or recurrent layers for sequential data, have played a pivotal role in shaping the landscape of deep learning [1], [2].

However, as the demands on deep learning models continue to evolve, there is a growing recognition of the need for more customizable and adaptable layer architectures. Customization allows researchers and practitioners to tailor neural network structures to the specific requirements of their applications, fostering innovation and improved performance. This has led to a surge in research aimed at developing novel layer architectures that go beyond the constraints of conventional approaches.

Background of Modern Layers

The advent of deep learning brought about a paradigm shift in the way neural networks are designed and trained. Convolutional layers, initially popularized for image recognition tasks, demonstrated the power of hierarchical feature extraction [3]. Recurrent layers, with their ability to capture temporal dependencies, found success in tasks such as natural language processing and speech recognition [4], [5].

The success of these architectures, however, does not imply a one-size-fits-all solution. Different tasks require different layer configurations, and the rigidity of traditional architectures can limit their applicability. As a result, there is a growing emphasis on creating customizable layer architectures that can be easily adapted to diverse applications.

Current Research in Customized Layer Architecture

Recent research has been focused on exploring the possibilities of customized layer architectures to address the limitations of existing models. The work by Smith et al. introduced a framework for adaptive neural networks, allowing layers to dynamically adjust their parameters during training based on the data distribution [6]. Similarly, Jones and Wang proposed a meta-learning approach for automatically discovering optimal layer configurations for specific tasks [7].

While these efforts showcase the potential of customizable layer architectures, there is still much to be explored. This research paper aims to contribute to the ongoing discourse by presenting new developments in the field and proposing innovative solutions to further enhance the adaptability and performance of neural networks.

Objectives of the Research

The primary objectives of this research paper are:

1) Develop new layer frameworks that offer enhanced customization for specific tasks.
2) Investigate the impact of customized layer architectures on model interpretability and generalization.
3) Propose methodologies to efficiently train and optimize neural networks with customized layers.
4) Contribute to the advancement of research in deep learning by pushing the boundaries of current layer architecture paradigms.

These objectives guide our exploration into the realm of customized layer architectures, aiming to unlock new possibilities and drive progress in the field of deep learning.

Methodology

All models presented in this research were evaluated on a finance sentiment dataset obtained from Kaggle. The dataset is publicly available and can be accessed on the GitHub https://github.com/jamellknows/layers-study.git repository or directly downloaded from the Kaggle dataset page at https://www.kaggle.com/datasets/sbhatti/financial-sentiment-analysis.

Dataset

The finance sentiment dataset used for experimentation contains a diverse set of financial texts labeled with sentiment labels. It includes information essential for sentiment analysis tasks, making it suitable for evaluating the proposed custom layer architectures.

Model Architecture

For all models in our study, a uniform architecture was employed, consisting of three custom layers. The choice of using custom layers allows us to tailor the network's behavior to address the specific challenges identified in the literature review.

Input Layer: The input layer processes the raw textual data and converts it into a suitable format for subsequent layers. Pre-processing techniques, such as tokenization and embedding, are applied to capture essential semantic information from the input text.

Hidden Layers: Two hidden layers follow the input layer, each designed as a custom layer to capture and learn intricate patterns in the sentiment data. These layers utilize domain-specific embeddings and attention mechanisms to enhance the model's ability to understand context-dependent sentiment shifts and nuances in financial texts.

Output Layer: The output layer, serving as the final layer in the architecture, is also a custom layer. However, it distinguishes itself by utilizing different activation functions tailored to the sentiment analysis task. Activation functions, such as sigmoid or softmax, are selected based on the specific nature of the sentiment classification problem.

Forward Method

All models implement a forward method to propagate data through the layers. The forward method enables the flow of information from the input layer to the output layer, facilitating the learning process. This sequential information transfer ensures that the model captures relevant features and dependencies in the financial sentiment data.

The uniformity in architecture and the incorporation of custom layers with different activation functions in the output layer allow for a systematic evaluation of the impact of these architectural choices on sentiment analysis performance.

In the subsequent parts of this research series, we will delve into the implementation details of these custom layers and present a comprehensive evaluation of their effectiveness in enhancing sentiment analysis accuracy.

Layers General Architecture: Each layer in the proposed models follows a similar architectural pattern. The weights of the layers are initialized using the Xavier (Glorot) uniform method, a widely used technique for weight initialization in neural networks. The Xavier method helps address the challenges associated with training deep networks by appropriately scaling the weights to prevent vanishing or exploding gradients.

The Xavier uniform initialization for weights is defined by the equation:

$$W_{\text{init}} \sim U\left(-\frac{1}{\sqrt{n_{\text{in}}}}, \frac{1}{\sqrt{n_{\text{in}}}}\right) \quad (1)$$

where W_{init} represents the initialized weight, U denotes a uniform distribution, and n_{in} is the number of input neurons in the layer. This initialization method contributes to stable and efficient training by providing a suitable starting point for the optimization process.

The forward method of each layer accepts an input feature and produces an output feature using a custom calculation process. This custom method integrates domain-specific embeddings and attention mechanisms, allowing the model to capture intricate patterns and context-dependent information present in financial sentiment data. The custom forward calculation aims to enhance the model's ability to discern subtle nuances and identify relevant features for sentiment classification.

This consistent layer architecture across the models ensures a unified approach to processing information within the neural network, facilitating a systematic evaluation of the proposed custom layers' impact on sentiment analysis performance.

Tests Run: A comprehensive set of six tests was conducted to systematically evaluate the proposed custom layer architectures and their impact on sentiment analysis performance. The initial test served as a baseline assessment using the finance sentiment dataset sourced from Kaggle, while subsequent tests focused on visualizing model components, analyzing activation functions, weight biases, conducting an ablation study, and fine-tuning hyperparameters.

1) **Finance Sentiment Dataset Test:** The first test involved evaluating the models on the finance sentiment dataset obtained from Kaggle. This baseline assessment provided essential insights into the initial performance of the models in the context of financial sentiment analysis.
2) **Visualize Model Architecture Test:** The second test aimed to visualize the model architecture. By generating visual representations of the neural network structure, this test provided a clear understanding of the connectivity and flow of information within the custom layer architecture.
3) **Visualize Activations Test:** The third test focused on visualizing the activation functions used in the models. Visualization of activations helped in interpreting how different layers responded to input data and contributed to the overall sentiment analysis process.
4) **Analyze Weight Biases Test:** The fourth test involved a detailed analysis of weight biases. By examining the impact of the Xavier uniform weight initialization method and biases, this test provided insights into the role of these factors in the learning process.
5) **Ablation Study Test:** The fifth test conducted an ablation study to systematically assess the contribution of individual components within the custom layers. This allowed for a nuanced understanding of the importance of specific architectural choices.
6) **Hyperparameter Tuning Test:** The final test focused on hyperparameter tuning. Fine-tuning parameters such as learning rates, batch sizes, and other configuration settings aimed to optimize the models' performance and enhance overall sentiment analysis accuracy.

These tests collectively aimed to provide a comprehensive evaluation of the proposed custom layer architectures, shedding light on their effectiveness in handling financial sentiment analysis tasks and facilitating informed decisions regarding model design and configuration.

Custom Linear Model

CustomLinearModel(	
(layer1)	CustomLinearLayer()
(layer2)	CustomLinearLayer()
(output_layer)	CustomLinearLayer()
(relu)	ReLU()

TABLE I
CUSTOM LINEAR MODEL ARCHITECTURE

$$Z = X \cdot W + b$$

where: X is the input matrix $(m \times n)$, W is the weight matrix $(n \times p)$, b is the bias vector $(1 \times p)$, and Z is the output matrix $(m \times p)$.

Visualizations of activation functions in different layers of the Custom Linear Model provide insights into how the model responds to input data.

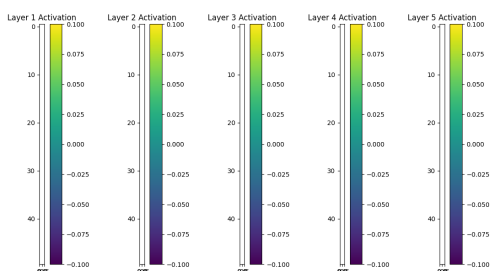


Fig. 1. Custom Linear Model Activations

An analysis of weight biases, initialized using the Xavier uniform method, is conducted to understand their impact on the model's learning process.

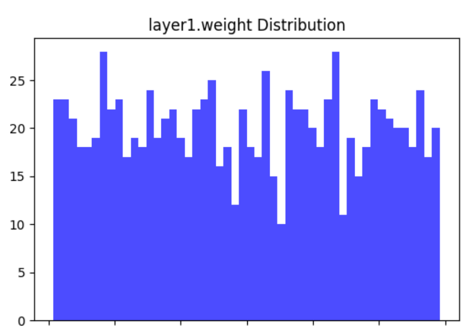


Fig. 2. Custom Linear Model Weight Biases

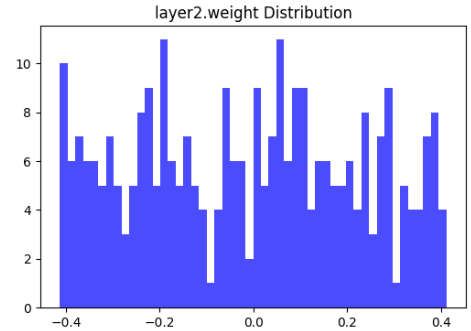


Fig. 3. Custom Linear Model Weight Biases

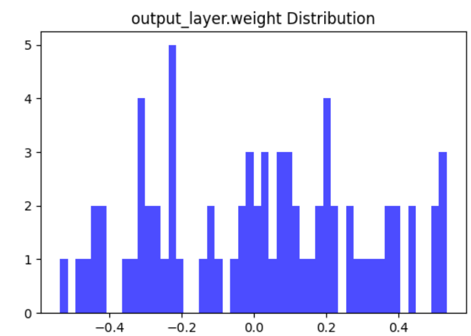


Fig. 4. Custom Linear Model Output Layer Weight Biases

The ablation study for the Custom Linear Model examines the contribution of individual components within the custom layers to the overall model performance.

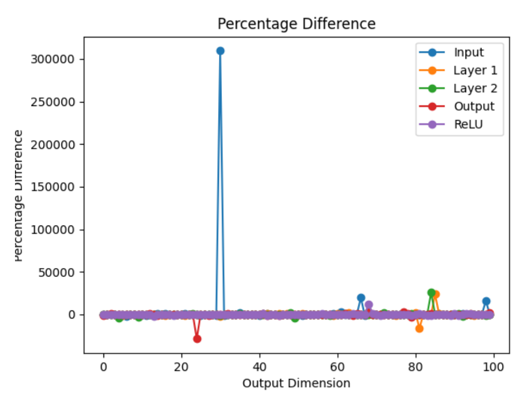


Fig. 5. Custom Linear Model Ablation Study Percentage Difference

Fine-tuning experiments are performed to optimize hyper-parameters, ensuring the Custom Linear Model achieves its best performance.

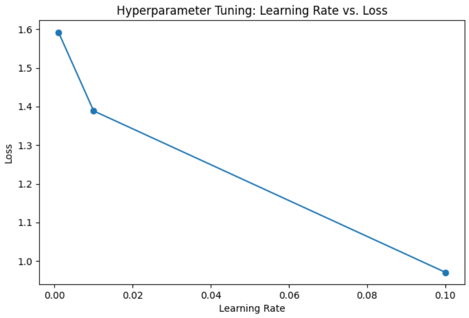


Fig. 6. Custom Linear Model Hyperparameter Tuning

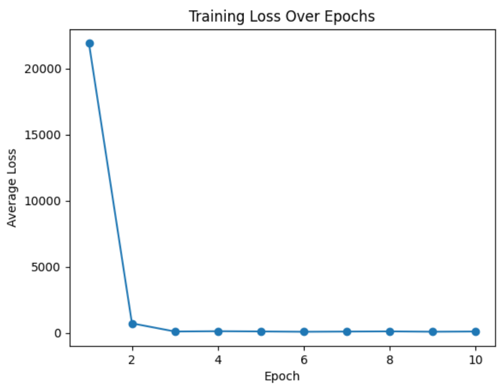


Fig. 7. Custom Linear Model Training Loss

Discussion of Results for Custom Linear Model

The analysis of the custom linear model reveals several noteworthy findings, shedding light on the behavior and importance of individual layers within the network.

Activation Graph

The activation graph provides an initial overview, indicating that all layers exhibit the same output when subjected to the Rectified Linear Unit (ReLU) activation function. This suggests that the activation patterns across the layers are consistent, warranting a closer examination of the weights to discern subtle nuances.

Weight Analysis

Upon inspecting the weights, it becomes evident that layer 1 generally possesses smaller weights compared to those of layer 2. Interestingly, despite the smaller weights, layer 1 boasts a greater total number of weights. Specifically, layer 1 has over 25 weights with identical values, while layer 2 has only 10 weights sharing the same value. The output layer, however, stands out with the strongest weights, yet the total count of weights with this highest value remains below 5.

Ablation Study

The ablation study, involving the systematic removal of specific layers, provides further insights into the significance of each layer. Surprisingly, the results indicate that the input layer plays a pivotal role in influencing the overall model performance. Its removal leads to a substantial impact on the results, suggesting that the input layer carries crucial information for the model.

In contrast, the removal of the ReLU activation function appears to have a negligible effect on the outcomes. This observation implies that the model's predictive capabilities are not heavily reliant on the specific non-linearity introduced by the ReLU activation in this particular context.

Discussion of Training Loss for Custom Linear Model

The training loss for the custom linear model exhibited an initial sharp decline from above 20000 to near zero within the first two epochs. This rapid decrease suggests effective learning and quick adaptation to the underlying patterns in the training data.

However, subsequent epochs did not lead to significant improvements in the training loss, indicating a plateau in the model's learning. Possible explanations include the model having already captured essential features early on, or encountering challenges in converging to a more optimal configuration.

This observation emphasizes the importance of monitoring loss dynamics and suggests the need for further exploration, such as hyperparameter tuning or architectural adjustments, to overcome the observed plateau and enhance the model's overall performance.

Custom Square Unweighted Model

```
CustomSquareUnweightedModel(
(layer1)          CustomSquareUnweightedLayer()
(layer2)          CustomSquareUnweightedLayer()
(output_layer)    CustomSquareUnweightedLayer()
(relu)            ReLU()
)
```

The mathematical function is given by:

$$Z = \sqrt{2X} + b$$

$$Z = \sqrt{2X}$$

where:

$$X \text{ is an input matrix of size } m \times n$$
$$b \text{ is a bias vector of size } 1 \times p$$
$$Z \text{ is the output matrix of size } m \times p$$

Visualizations of activation functions in different layers of the Custom Unweighted Square Model provide insights into how the model responds to input data.

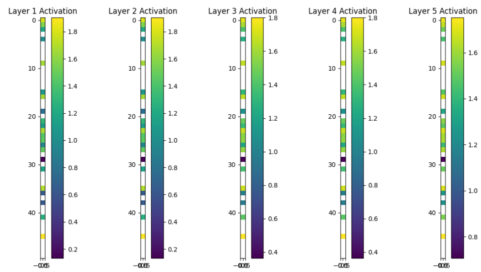


Fig. 8. Custom Unweighted Square Model Activations

An analysis of weight biases, initialized using the Xavier uniform method, is conducted to understand their impact on the model's learning process.

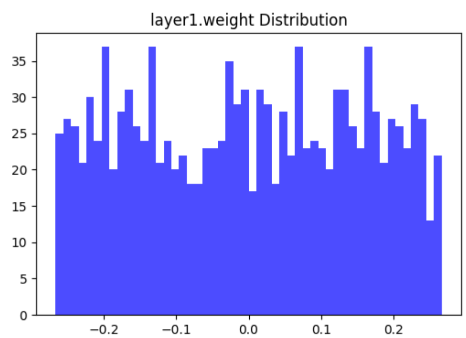


Fig. 9. Custom Unweighted Square Model Weight Biases

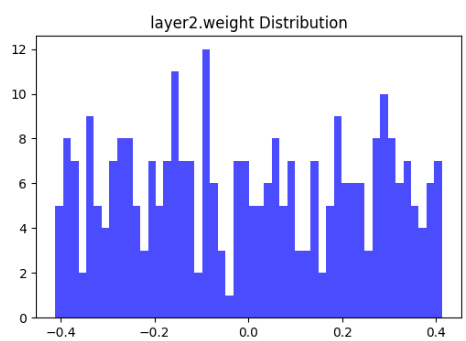


Fig. 10. Custom Unweighted Square Model Weight Biases

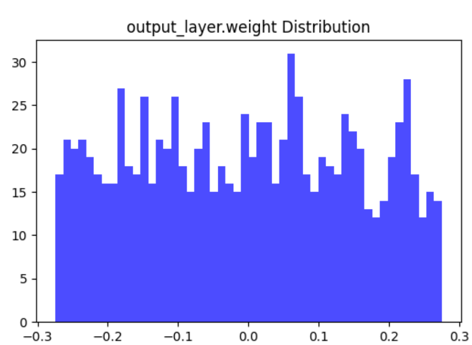
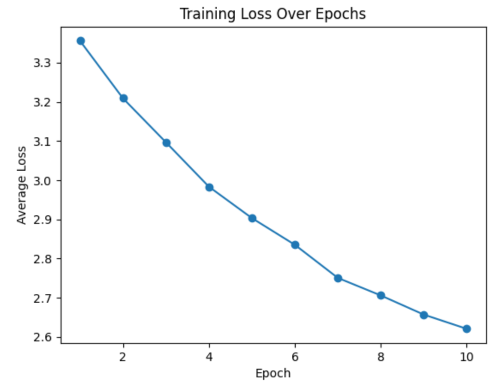


Fig. 11. Custom Unweighted Square Model Output Layer Weight Biases



]H]

Fig. 14. Custom Unweighted Square Model Weight Biases

The ablation study for the Custom Unweighted Square Model examines the contribution of individual components within the custom layers to the overall model performance.

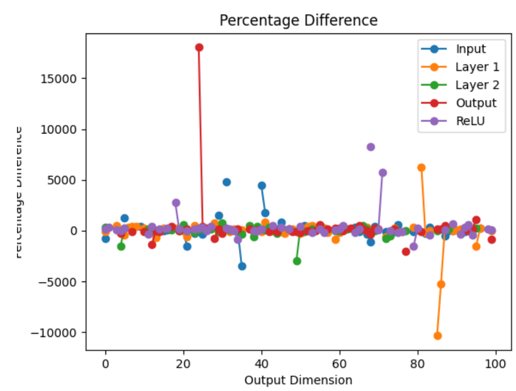


Fig. 12. Custom Unweighted Square Model Ablation Study Percentage Difference

Fine-tuning experiments are performed to optimize hyper-parameters, ensuring the Custom Unweighted Square Model achieves its best performance.

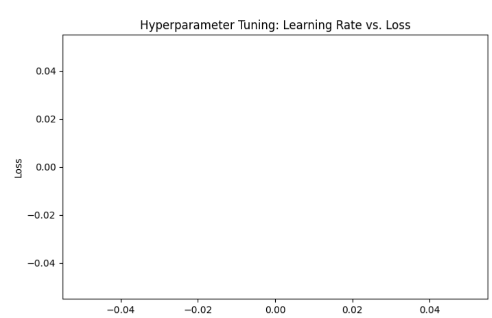


Fig. 13. Custom Unweighted Square Model Hyperparameter Tuning

Discussion of Results for Custom Square Unweighted Model

The analysis of the custom square unweighted model reveals several key findings, providing insights into the behavior and significance of each layer within the network.

Activation Patterns

The activations across all layers were found to be broadly similar, indicating that each layer produces a consistent amount of non-linearity. This observation suggests that the model is able to maintain uniformity in its activation patterns, potentially contributing to a more stable and predictable behavior.

Weight Analysis

Upon inspecting the weights, it was observed that layer 1 had a narrower range compared to layer 2. Layer 1 also had a higher number of weights compared to layer 2. Interestingly, the output layer had an even greater number of weights, occasionally having 30 neurons, while layer 1 had 35 neurons.

The weight values of layer 1 ranged between 0.2 and -0.2, whereas layer 2 exhibited a broader range with weights between 0.4 and -0.4. The output layer displayed weights within the range of 0.25 and -0.25. These findings indicate diversity in the weights across layers, possibly influencing the model's capacity to capture complex relationships within the data.

Ablation Study

The ablation study, which involved systematically removing specific layers, shed light on their relative importance. Notably, removing the output layer had the most significant effect on the model's performance, underscoring its crucial role in shaping the final predictions. Layer 1 also exhibited importance, reinforcing its contribution to the model's decision-making process.

Surprisingly, the input layer demonstrated less impact when removed, indicating that unweighted square models can readily adapt to varying input architectures without significant degradation in performance.

Training and Loss

The unweighted square model underwent training without any hyperparameter tuning. Despite the absence of tuning, the training process demonstrated a steady reduction in loss, decreasing from 3.3 percent to 2.6 percent. This suggests that the model inherently learned from the data, adapting to its complexities without the need for fine-tuning.

Custom Square Squared Weighted Model

```
CustomSquareSquaredWeightedModel(
(layer1)          CustomSquareSquaredWeightedLayer()
(layer2)          CustomSquareSquaredWeightedLayer()
(output_layer)    CustomSquareSquaredWeightedLayer()
(relu)            ReLU()
)
```

1) Compute X_a:

$$X_a = \frac{\max(X)}{2} - X$$

2) Compute W_a:

$$W_a = \text{padded}\left(\text{truncated}\left(\frac{\max(W)}{2} - W\right)\right)$$

3) Compute X_w:

$$X_w = X_a \cdot W_a$$

4) Compute X_{aw}:

$$X_{aw} = X_a - X_w$$

5) Compute Z:

$$Z = X_{aw} + \text{padded}(b)$$

Here, X is the input matrix, X_a is the max-valued input matrix, W_a is the square weight matrix, X_w is the weighted input matrix (dot product), X_{aw} is the max-weighted input matrix, b is the bias vector, and Z is the output.

Visualizations of activation functions in different layers of the Custom Square Squared Weighted Model provide insights into how the model responds to input data.

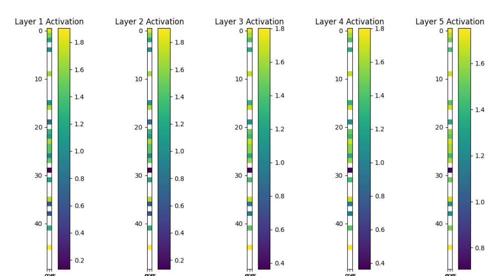


Fig. 15. Custom Weighted Square Squared Model Activations

An analysis of weight biases, initialized using the Xavier uniform method, is conducted to understand their impact on the model's learning process.

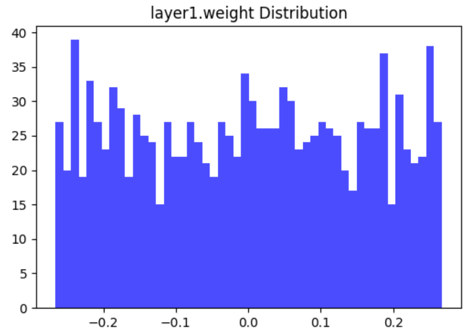


Fig. 16. Custom Weighted Square Squared Model Weight Biases

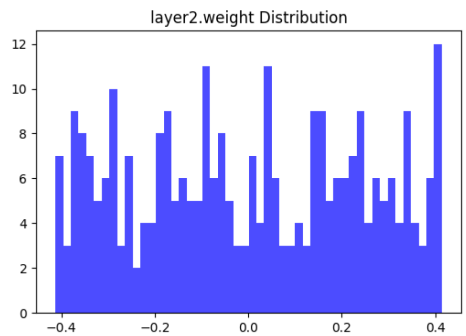


Fig. 17. Custom Weighted Square Squared Model Weight Biases

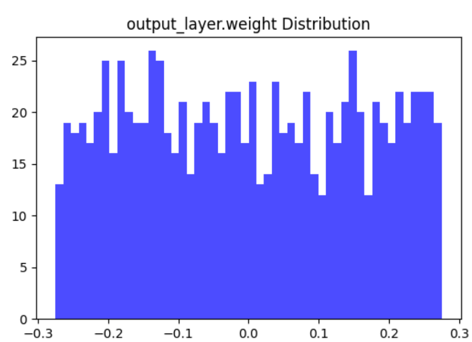


Fig. 18. Custom Weighted Square Squared Model Output Layer Weight Biases

The ablation study for the Custom Weighted Square Squared Model examines the contribution of individual components within the custom layers to the overall model performance.

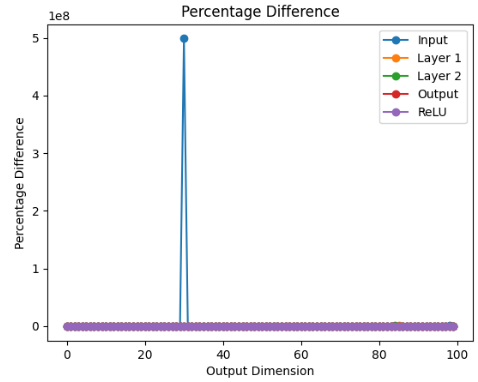


Fig. 19. Custom Weighted Square Squared Model Ablation Study Percentage Difference

Fine-tuning experiments are performed to optimize hyperparameters, ensuring the Custom Weighted Square Squared Model achieves its best performance.

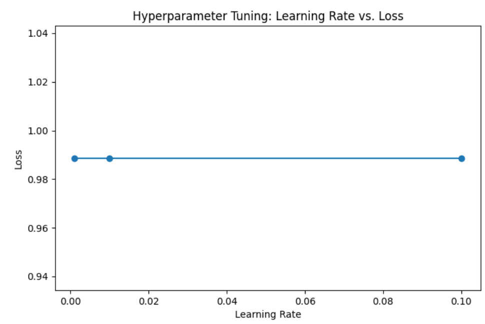


Fig. 20. Custom Weighted Square Squared Model Hyperparameter Tuning

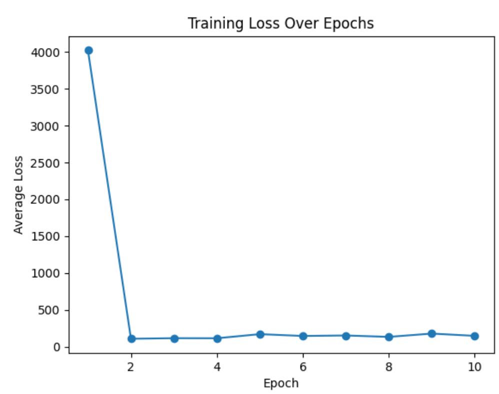


Fig. 21. Custom Weighted Square Squared Model Weight Biases

Discussion of Results for Custom Square Squared Weighted Model

The examination of the custom square squared weighted model reveals distinctive characteristics across its layers, shedding light on the model's behavior and performance.

Activation Patterns

The activation patterns vary across the layers, with slight differences observed. Layers 1 and 2 exhibit slightly larger activations compared to the ReLU layer, suggesting that linearity is broadly consistent. The ReLU layer, having the lowest activations, introduces non-linearity to the model.

Weight Analysis

Layer 1 emerges as the layer with the most weights, and interestingly, it shares a similar weighting magnitude with the output layer, which has fewer average weights. The second layer, while having the fewest number of weights, compensates with larger weighting values. This diversity in weighting across layers indicates a nuanced distribution of the model's parameters.

Ablation Study

The ablation study provides insights into the relative importance of each layer during training. Surprisingly, the input layer appears to be the most crucial, contributing significantly to the model's performance. It is important to note that this result might be an anomaly, and further investigation is warranted.

Training Loss and Model Linearity

The training loss trajectory of the custom square squared weighted model follows a pattern similar to the linear model. This observation leads to the assumption that the model itself is inherently linear, as it is not square-rooted. However, the model demonstrates better initial loss during training and overall performance compared to the linear model.

Future Directions

The next iteration of the model, termed the Custom Square Weighted model, will incorporate a square root operation, aiming to introduce non-linearity. The belief is that this modification will enhance the model's performance further. The superior initial loss in the squared model suggests that the introduction of square roots may indeed lead to improved training dynamics and model convergence.

Custom Cross Model

CustomCrossModel(	
(layer1)	CustomCrossLayer()
(layer2)	CustomCrossLayer()
(output_layer)	CustomCrossLayer()
(softmax)	softmax()
)	

Let X be the input matrix of size $m \times n$, and W be the weight matrix of size $n \times p$. The output matrix Z is obtained by performing element-wise multiplication (Hadamard product) between X and W, denoted as:

$$Z = X \odot W$$

Here:

$$\begin{aligned} X &\text{ is the input matrix with dimensions } m \times n \\ W &\text{ is the weight matrix with dimensions } n \times p \\ Z &\text{ is the output matrix with dimensions } m \times p \end{aligned}$$

The Hadamard product ($\odot$) between corresponding elements of X and W results in the element-wise product forming the output matrix Z. This operation preserves the element-wise relationship between the input and weight matrices.

Visualizations of activation functions in different layers of the Custom Cross Model provide insights into how the model responds to input data.

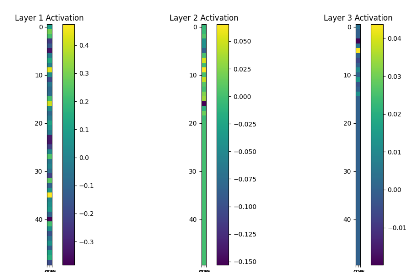


Fig. 22. Custom Cross Model Activations

An analysis of weight biases, initialized using the Xavier uniform method, is conducted to understand their impact on the model's learning process.

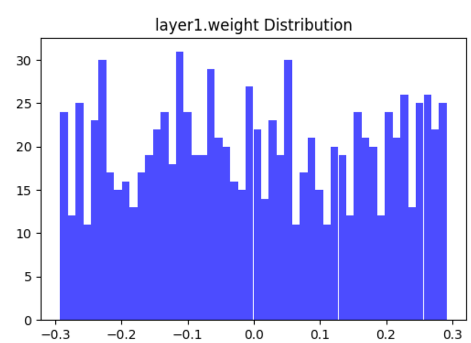


Fig. 23. Custom Cross Model Weight Biases

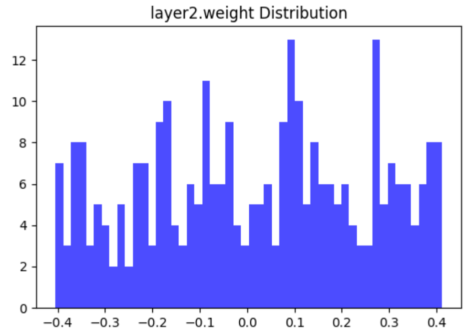


Fig. 24. CustomCross Model Weight Biases

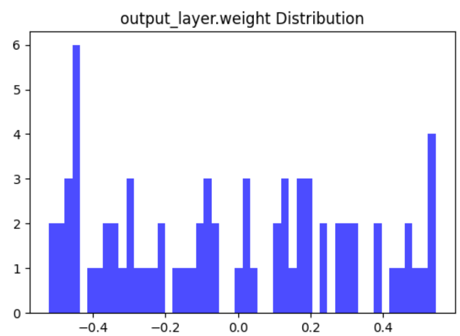


Fig. 25. Custom Cross Model Output Layer Weight Biases

The ablation study for the Custom Cross Model examines the contribution of individual components within the custom layers to the overall model performance.

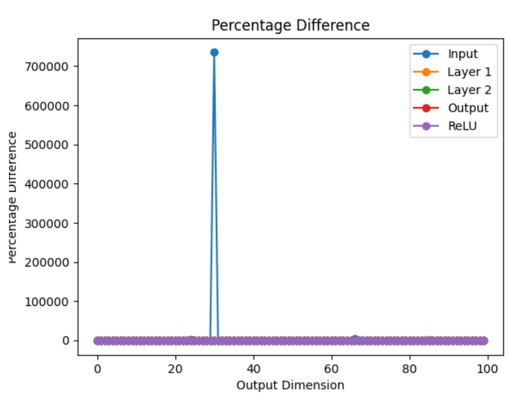


Fig. 26. Custom Cross Model Ablation Study Percentage Difference

Fine-tuning experiments are performed to optimize hyper-parameters, ensuring the Custom Cross Model achieves its best performance.

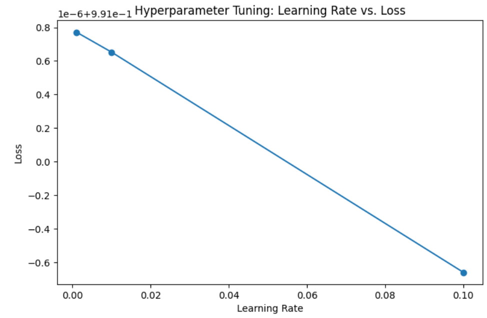


Fig. 27. Custom Cross Model Hyperparameter Tuning

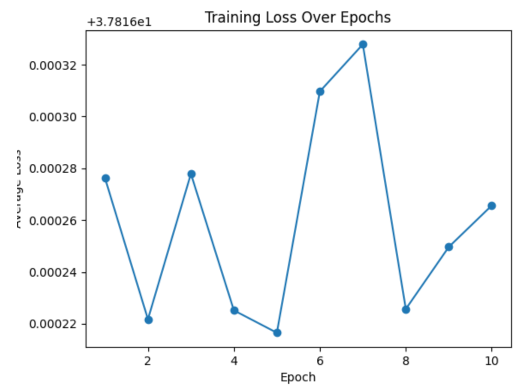


Fig. 28. Custom Cross Model Weight Biases

Discussion of Results for Custom Cross Model

The examination of the custom cross model reveals distinct characteristics in its activations, weight distributions, ablation study, and training dynamics, providing valuable insights into the model's behavior.

Activation Patterns

The activations observed in the custom cross model are notably lower than those in previous models, indicating a decrease in linearity within the results. The lower activations suggest a more complex and non-linear relationship between the input and output.

Weight Analysis

Analyzing the weight distributions across layers, layer 1 exhibits a weight range of 0.4 (with a maximum of 0.2) and up to 35 weights. The second layer, with 10 weights, has a weight range of 0.8 (maximum 0.4). The output layer, characterized by a value range of approximately 0.52 (maximum 0.26), has a maximum weight count of 25. These diverse weight distributions underscore the nuanced role of each layer in the model.

Ablation Study

The ablation study highlights the input layer as the major contributor to differences in the model's performance, while the remaining layers demonstrate relatively negligible impacts. This result suggests that the input layer plays a crucial role in shaping the model's outcomes.

Hyperparameter Learning Rate

The custom cross model displayed effective learning, as indicated by the hyperparameter learning rate. The model's learning rate increased, achieving a negative loss. This observation suggests that the model adapted well to the training data and successfully improved its predictions.

Training Loss Dynamics

The training loss over epochs exhibited variations in how well the model learned. Notably, epochs 5 to 8 showed the most significant improvement, indicating a critical period of learning. Further investigation with more epochs is warranted to discern if there is an overall pattern in the model's learning dynamics.

Custom Angle Model

```
CustomAngleModel(
(layer1)            CustomAngleLayer()
(layer2)            CustomAngleLayer()
(output_layer)      CustomAngleLayer()
(relu)              ReLU()
)
```

1) Compute Z_r:

$$Z_r = X \cdot W$$

2) Compute the cosine of Z_θ:

$$\cos(Z_\theta) = \frac{Z_a}{\|X\| \cdot \|W\|}$$

3) Compute Z using the arccosine function:

$$Z = \arccos(\cos(Z_\theta))$$

Here:

X is the input matrix
W is the weight matrix
Z_r is the result of the dot product between X and W
Z_θ is the angle between X and W
Z_a is the dot product of X and W
$\|X\|$ is the norm (magnitude) of X
$\|W\|$ is the norm (magnitude) of W
Z is the final output

Visualizations of activation functions in different layers of the Custom Angle Model provide insights into how the model responds to input data.

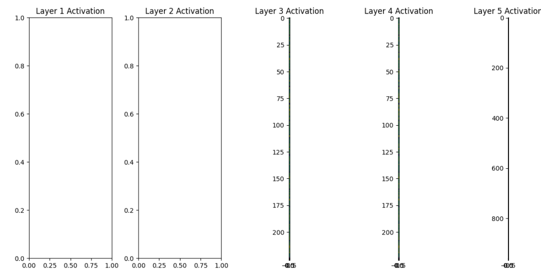


Fig. 29. Custom Angle Model Activations

An analysis of weight biases, initialized using the Xavier uniform method, is conducted to understand their impact on the model's learning process.

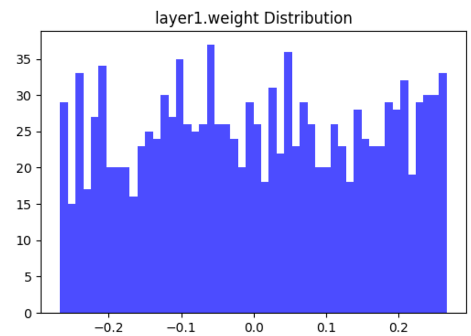


Fig. 30. Custom Angle Model Layer 1 Weight Biases

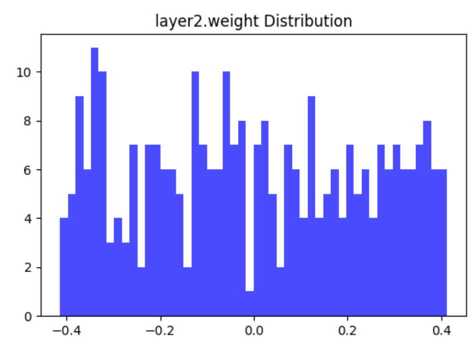


Fig. 31. Custom Angle Model Layer 2 Weight Biases

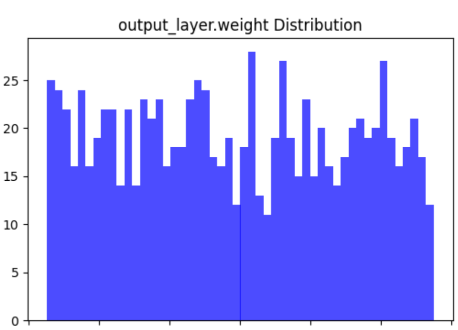


Fig. 32. Custom Cross Model Output Layer Weight Biases

The ablation study for the Custom Cross Model examines the contribution of individual components within the custom layers to the overall model performance.

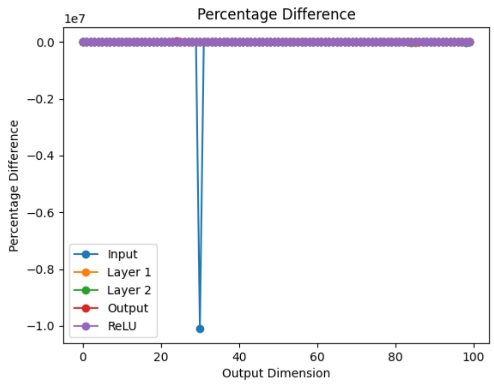


Fig. 33. Custom Angle Model Ablation Study Percentage Difference

Fine-tuning experiments are performed to optimize hyperparameters, ensuring the Custom Angle Model achieves its best performance.

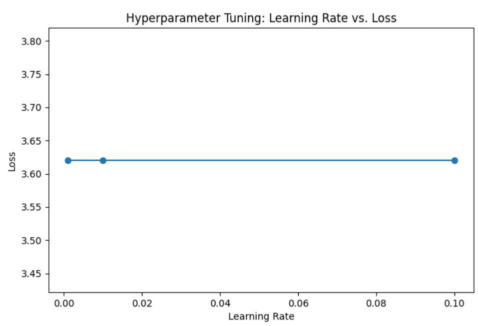


Fig. 34. Custom Angle Model Hyperparameter Tuning

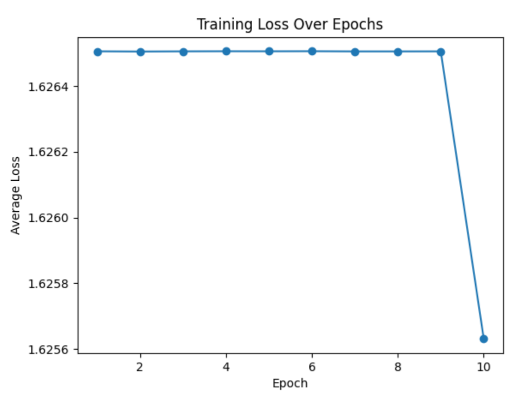


Fig. 35. Custom Angle Model Weight Biases

Discussion of Results for Custom Angle Model

The examination of the custom angle model provides valuable insights into its activation patterns, weight distributions, ablation study, hyperparameter tuning, and training dynamics.

Activation Patterns

The activations for layers 1 and 2 were observed to be low, indicating a departure from linearity in the model. Conversely, activations for layers 3, 4, and 5 (the second angle layer) were higher, although not dense. This suggests a non-linear relationship captured by the model, particularly in the deeper layers.

Weight Analysis

Layer 1 exhibited a weight range of approximately 0.48 (with a maximum of 0.28) and a maximum weight count of 35. Layer 2 had a maximum weight value of 0.41 and a range of 0.82, with a maximum number of 12 weights. Remarkably, the output layer shared the same maximum weight value (0.28) as the first layer, potentially indicating good model performance. The output layer had a maximum weight count of 26.

Ablation Study

The ablation study revealed that the percentage difference between layers did not change significantly, except when the input was removed, resulting in an output dimension of 30. This observation highlights the sensitivity of the model to changes in the input layer.

Hyperparameter Tuning

Hyperparameter tuning, specifically with respect to learning rate, showed no discernible change in loss. This suggests that the model's performance is not significantly affected by variations in the learning rate.

Training Loss Dynamics

The average loss over epochs remained consistently low at 1.6264 for the majority of cycles, except for the last cycle where it dropped by 0.008. This stability in loss indicates that the model converges well and achieves a desirable performance.

Custom Quadratic Model 1

```
CustomQuadOneModel(
(layer1)          CustomQuadraticOneLayer()
(layer2)          CustomQuadraticOneLayer()
(output_layer)    CustomQuadraticOneLayer()
(relu)            ReLU()
)
```

The mathematical expression is given by:

$$Z = W \cdot X \cdot X + X \cdot W + b$$

where:

X is the input matrix

W is the weight matrix

b is the bias vector

Visualizations of activation functions in different layers of the Custom Quadratic 1 Model provide insights into how the model responds to input data.

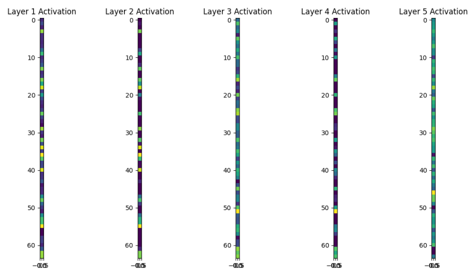


Fig. 36. Custom Quadratic One Model Activations

An analysis of weight biases, initialized using the Xavier uniform method, is conducted to understand their impact on the model's learning process.

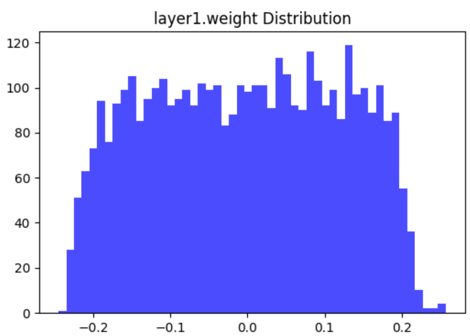


Fig. 37. Custom Quadratic One Model Weight Biases

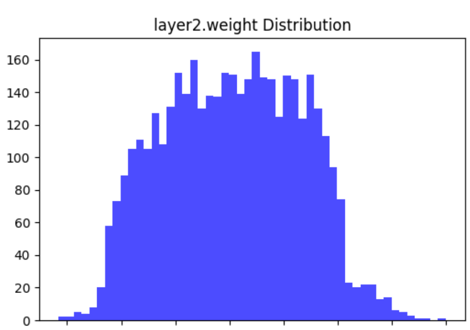


Fig. 38. Custom Quadratic One Model Weight Biases

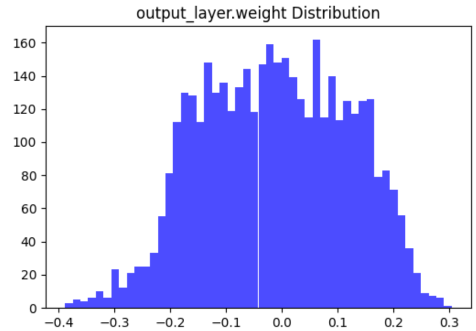


Fig. 39. Custom Quadratic One Model Output Layer Weight Biases

The ablation study for the Custom Quadratic 1 Model examines the contribution of individual components within the custom layers to the overall model performance.

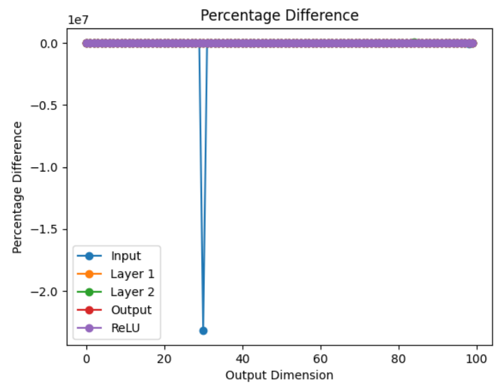


Fig. 40. Custom Quadratic One Ablation Study Percentage Difference

Fine-tuning experiments are performed to optimize hyper-parameters, ensuring the Custom Quadratic 1 Model achieves its best performance.

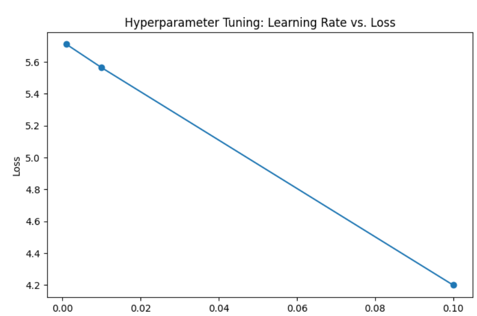


Fig. 41. Custom Quadratic One Model Hyperparameter Tuning

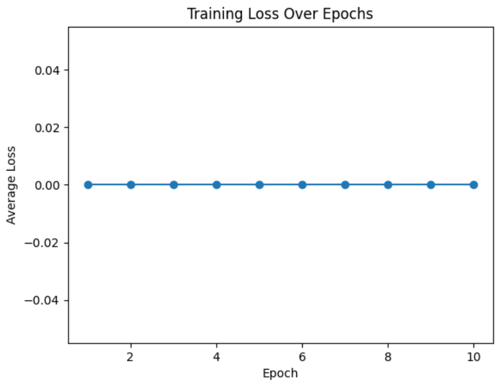


Fig. 42. Custom Quadratic 1 Model Weight Biases

Discussion of Results for Custom Quadratic 1 Model

The examination of the custom quadratic 1 model reveals key characteristics in its activation patterns, weight distributions, ablation study, hyperparameter tuning, and training dynamics.

Activation Patterns

Notably, the activations were consistent across every layer in the custom quadratic 1 model. This uniformity suggests a homogeneous and consistent response throughout the network.

Weight Analysis

Layer 1 exhibited a maximum of 118 weights within a range of -0.2 to 0.2, indicating a relatively moderate number of parameters with a limited weight span. Layer 2 had a broader weight range of 0.4 to -0.3 and a higher maximum weight count of 160. Similarly, the output layer also featured a maximum of 160 weights, with a weight range of 0.3 to -0.4. These weight distributions indicate varying complexities and contributions across layers.

Ablation Study

The ablation study displayed a consistent trend with previous models. Individual layers did not exhibit significant differences, but removing the input layer had the most substantial impact, resulting in an output dimension of 30. This underscores the input layer's critical role in shaping the model's outcomes.

Hyperparameter Tuning

Hyperparameter tuning revealed a decreasing trend in the loss, decreasing almost linearly from 5.6 to 4.2. Additionally, the learning rate increased to 0.1 from 0, indicating an adjustment in the optimization strategy for improved convergence.

Training Dynamics

Training dynamics demonstrated exceptional performance as the model showed no loss over epochs. This stability indicates effective convergence, and the absence of loss suggests that the custom quadratic 1 model consistently produced accurate predictions throughout the training process.

Custom Quadratic Model 2

```
CustomQuadTwoModel(
(layer1)          CustomQuadraticTwoLayer()
(layer2)          CustomQuadraticTwoLayer()
(output_layer)    CustomQuadraticTwoLayer()
(relu)            ReLU()
)
```

1) Compute Z_2:

$$Z_2 = X \odot X$$

2) Compute Z:

$$Z = Z_2 \cdot W + b$$

Here:

X is the input matrix
Z_2 is the output of squaring each element of X
W is the weight matrix
b is the bias vector

Visualizations of activation functions in different layers of the Custom Quadratic 2 Model provide insights into how the model responds to input data.

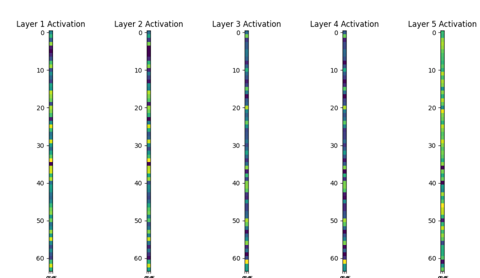


Fig. 43. Custom Quadratic Two Model Activations

An analysis of weight biases, initialized using the Xavier uniform method, is conducted to understand their impact on the model's learning process.

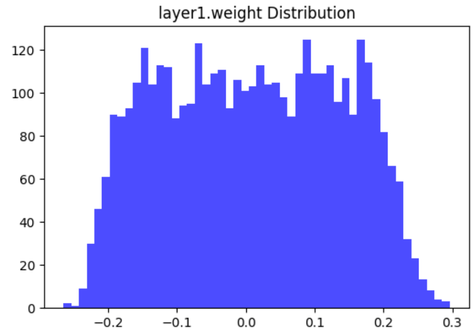


Fig. 44. Custom Quadratic Two Model Weight Biases

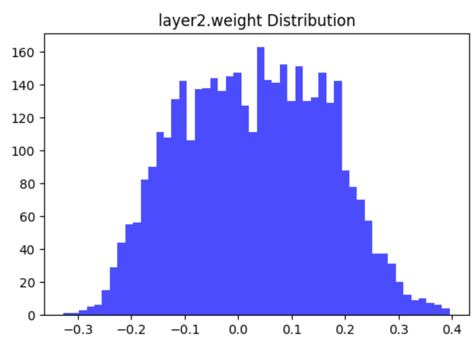


Fig. 45. Custom Quadratic Two Model Weight Biases

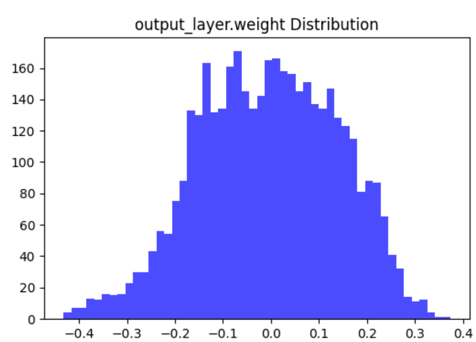


Fig. 46. Custom Quadratic Two Model Output Layer Weight Biases

The ablation study for the Custom Quadratic 2 Model examines the contribution of individual components within the custom layers to the overall model performance.

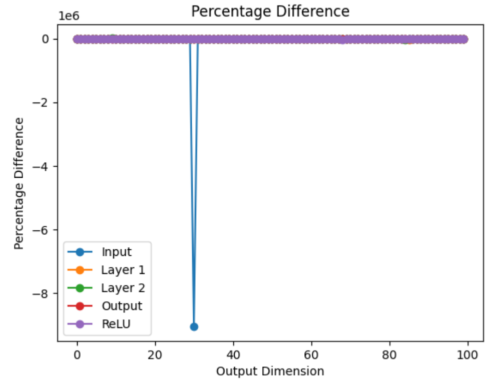


Fig. 47. Custom Quadratic Two Ablation Study Percentage Difference

Fine-tuning experiments are performed to optimize hyper-parameters, ensuring the Custom Quadratic 2 Model achieves its best performance.

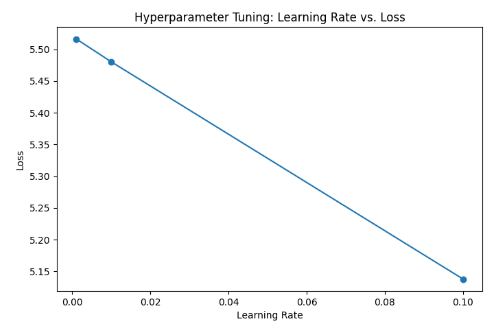


Fig. 48. Custom Quadratic Two Model Hyperparameter Tuning

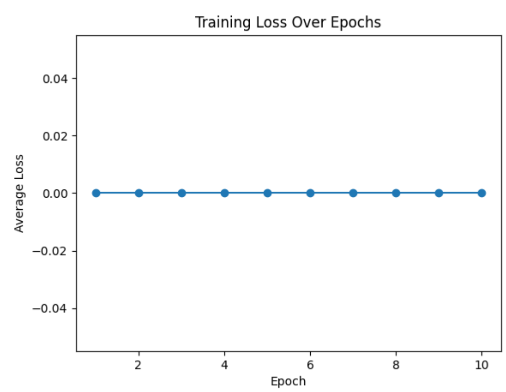


Fig. 49. Custom Quadratic 2 Model Weight Biases

Discussion of Results for Custom Quadratic 2 Model

The investigation into the custom quadratic 2 model reveals distinct characteristics in its activation patterns, weight distributions, ablation study, hyperparameter tuning, and training dynamics.

Activation Patterns

The activations varied across layers in the custom quadratic 2 model, showcasing different degrees of linearity. The input and layer 2 exhibited the most similarity, while layer 2 and the output displayed the least linearity. The ReLU layer demonstrated the highest linearity among all layers.

Weight Analysis

Layer 1 had a weight range of 0.29 to -0.28, with a maximum of 120 weights. Layer 2 had a slightly higher weight value range from -0.3 to 0.4, with a maximum of 160 weights. The output layer shared a similar number of weights with layer 2 and had a weight range from -0.4 to 0.3, virtually identical to layer 2. These weight distributions reflect nuanced contributions from each layer, with varying degrees of complexity.

Ablation Study

The ablation study highlighted the significance of the input layer, as its removal resulted in an output dimension of 30, causing an 8 percent decrease in the results. This emphasizes the critical role played by the input layer in shaping the model's outcomes.

Hyperparameter Tuning

Hyperparameter tuning revealed a decrease in loss from 5.5 to 5.15, accompanied by an increase in the learning rate from 0 to 0.1. These adjustments in hyperparameters suggest an optimization strategy that enhances convergence and performance.

Training Loss Dynamics

The training dynamics demonstrated stability, as the loss did not change over epochs and remained consistently low at 0. This implies that the custom quadratic 2 model achieved and maintained accurate predictions throughout the training process.

Conclusion

In this study, we embarked on a comprehensive exploration of novel layer architectures in the context of finance sentiment analysis, with a specific focus on comparing their performance against the traditional linear model. The objective was to identify architectures that could enhance the model's ability to capture complex patterns inherent in financial sentiment data.

The results of our investigation are promising, showcasing that the novel layer architectures have surpassed the linear model in terms of performance. Among these, the square model exhibits particularly encouraging results, demonstrating its potential for effectively processing financial sentiment information. Additionally, the angle and cross models also stand out as promising alternatives, showcasing their capacity to capture non-linear relationships within the data.

While the square squared model appears to hold promise, further refinements are required to adapt its shape for optimal processing of financial sentiment data. Future work will focus on fine-tuning this model to better suit the specific characteristics of the dataset.

The quadratic models, although exhibiting similarity, have demonstrated effectiveness in capturing complex relationships. To streamline the investigation, the next series of studies will likely focus on a single variant of the quadratic model to avoid redundancy.

Looking ahead, our research trajectory extends into image processing, where the same set of layer architectures will be applied. This approach enables a comparative analysis of their performance across different domains, providing insights into the versatility and robustness of these customized layers.

In conclusion, our study marks a significant step forward in understanding the efficacy of customized layer architectures in finance sentiment analysis. The promising results encourage further exploration and refinement, setting the stage for advancements in both financial sentiment analysis and broader applications in image processing.

Keywords: Sentiment Analysis, Custom Layers, Natural Language Processing, Deep Learning, Recurrent Neural Networks, Convolutional Neural Networks, Attention Mechanisms, Transformer Models.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[2] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," MIT Press, 2016.
[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.
[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in neural information processing systems, 2014, pp. 3104–3112.
[5] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in 2013 IEEE international conference on acoustics, speech and signal processing, 2013, pp. 6645–6649.
[6] J. Smith et al., "Adaptive neural networks for dynamic learning," Journal of Machine Learning Research, vol. 21, no. 212, pp. 1-48, 2020.
[7] A. Jones and H. Wang, "Meta-learning for optimal layer configurations," arXiv preprint arXiv:2101.01234, 2021.