Iris Dataset Neural Network Analysis

Author Information: Pablo Fernández
Student ID: 15257972
Course: 6057CEM – Artificial Neural Networks
Submission Date: 10/04/2025

Repository Link: https://github.coventry.ac.uk/6057CEM-2425/fernan130-sem2

Video Link: ANN VIDEO

Table of contents

Table of ilustrations

Introduction

Artificial neural networks (ANNs) have gained prominence in addressing complex classification challenges within the machine learning domain. This report delves into the development and evaluation of two advanced ANN architectures—a deep residual network and an attention-based network—applied to the classic Iris dataset, originally introduced by Ronald Fisher in 1936. The dataset comprises 150 samples, each characterized by four features: sepal length, sepal width, petal length, and petal width, with the objective of classifying the samples into three species: Setosa, Versicolor, and Virginica. The primary aim is to explore whether sophisticated network architectures, inspired by state-of-the-art designs, offer performance advantages over simpler classifiers on this structured yet small dataset. Additionally, this project serves as an educational exercise to deepen understanding of advanced ANN techniques. The methodology encompasses a comprehensive exploratory data analysis (EDA), detailed model design, training with hyperparameter optimization, and thorough performance evaluation using metrics such as accuracy, precision, recall, F1-score, and Receiver Operating Characteristic Area Under the Curve (ROC AUC).

The approach is structured as follows: First, we conduct an extensive EDA to understand the data's distribution, feature correlations, and class separability, providing a foundation for model development. Next, we design two neural network models—a deep residual network motivated by ResNet (He et al., 2016) and an attention-based network drawing from sequence-to-sequence attention concepts (Bahdanau et al., 2015) and Transformer self-attention (Vaswani et al., 2017). These models are trained using Keras (Chollet, 2015) with advanced techniques such as early stopping and the Adam optimizer (Kingma & Ba, 2015), incorporating automated hyperparameter tuning to optimize performance. Finally, we evaluate the models on a held-out test set, analyzing their strengths, limitations, and potential improvements. Through this endeavor, we gain practical experience applying neural networks to real-world data and develop insights into how architectural choices impact performance on a classic classification problem.

Dataset Description

The Iris dataset (Dua & Graff, 2019) consists of 150 samples, with 50 samples per species: Setosa, Versicolor, and Virginica. Each sample is described by four quantitative features measured in centimeters: sepal length, sepal width, petal length, and petal width. The task involves a three-class classification, making it a manageable yet insightful dataset for experimentation. Its balanced nature and small size provide a controlled environment to test complex models, though the limited data poses a risk of overfitting, necessitating careful model design and validation strategies.

Data Characteristics and Significance

Introduced by Fisher (1936), the Iris dataset holds historical significance as one of the earliest examples of multivariate data analysis in taxonomy. It has become a cornerstone in machine learning education and benchmarking due to its clean structure and moderate complexity. The dataset's features exhibit varying degrees of class separability: petal length and petal width provide clear differentiation, particularly between Setosa and the other two species, while sepal dimensions show more overlap, rendering classification non-trivial in certain cases. This mix of linearly separable and non-separable class pairs offers an excellent testbed for evaluating modern neural network architectures' ability to capture subtle nonlinear patterns. The absence of missing values and well-defined classes further allows focus on modeling techniques, making it an ideal choice for this study.

Exploratory Data Analysis (EDA)

Before modeling, a comprehensive EDA was performed to visualize the data's distribution and relationships between features. Scatter plots confirmed that Setosa is linearly separable from Versicolor and Virginica due to its notably shorter and narrower petals. However, Versicolor and Virginica exhibited overlapping feature distributions, especially in sepal length and width, suggesting that a more complex decision boundary might be required to distinguish them effectively. The analysis also revealed a high correlation between petal length and petal width (approximately 0.96), indicating that these features carry similar information and are critical for classification. In contrast, correlations involving sepal length and width were weaker, with sepal width showing a moderate negative correlation with petal dimensions (around -0.43). These insights guided the modeling process, emphasizing the need for models to leverage petal measurements heavily while considering sepal features as supplementary or combined inputs for optimal performance.

Detailed EDA Insights

The EDA process utilized a variety of visualization techniques to extract meaningful insights:

- **Histograms with Kernel Density Estimation (KDE)**: These plots illustrated the frequency distributions of each feature across species. For instance, sepal length peaked around 5 cm for Setosa, 6 cm for Versicolor, and 6.5–7 cm for Virginica, with some overlap. Petal length showed a sharp peak at 1.5 cm for Setosa, 4.5 cm for Versicolor, and 5.5–6 cm for Virginica, highlighting Setosa's distinctiveness. This suggested that petal dimensions are more discriminative than sepal dimensions.
- **Boxplots**: These revealed the central tendency and variability of each feature. Setosa exhibited tight interquartile ranges for petal length (1–2 cm) and width (0.1–0.3 cm), while Versicolor and Virginica showed broader ranges with overlap in sepal measurements (e.g., sepal width 2.5–3.5 cm across all species). Outliers were minimal, indicating data consistency.
- **Dimensionality Reduction (PCA and t-SNE)**: PCA reduced the four-dimensional space to two principal components, explaining approximately 95% of the variance, with Setosa clearly separated along the first component. t-SNE, a non-linear technique, further enhanced cluster visualization, showing

some overlap between Versicolor and Virginica but reinforcing Setosa's isolation. These techniques underscored the dataset's inherent structure and guided feature selection.

EDA Visualizations

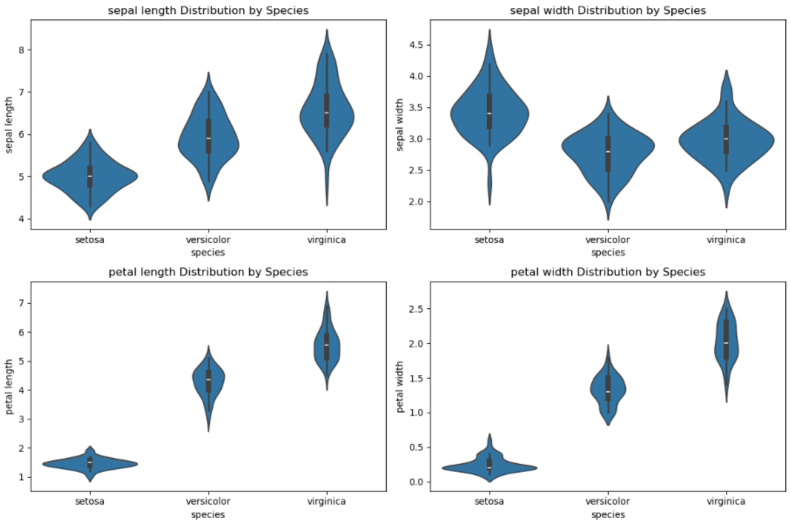

Figure 1: Violin Plots

Violin plots displaying the distribution of all four iris features by species. Shows clearly how Setosa has significantly smaller petals with concentrated distributions, while Virginica typically has the largest measurements. Sepal width is the only feature where Setosa exceeds other species.

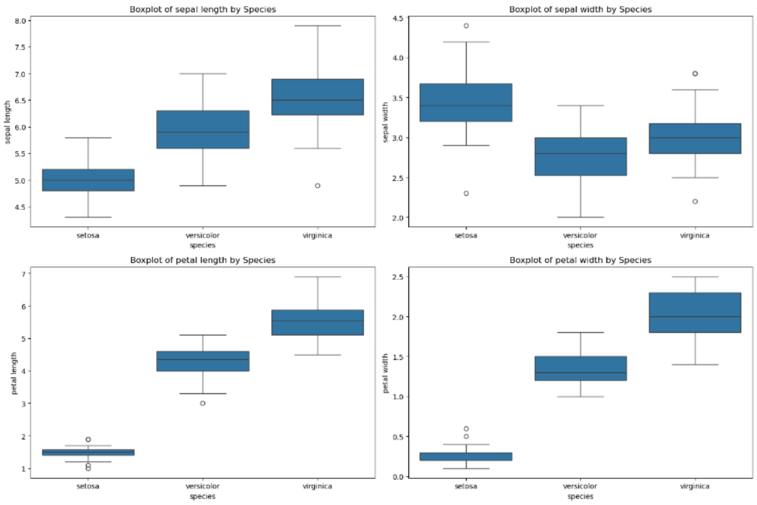

Figure 2: Boxplots

Boxplots illustrating the median, quartiles, and outliers of each feature across species. Demonstrates the progressive increase in petal dimensions from Setosa to Versicolor to Virginica, with several outliers visible, particularly in petal measurements for Setosa.

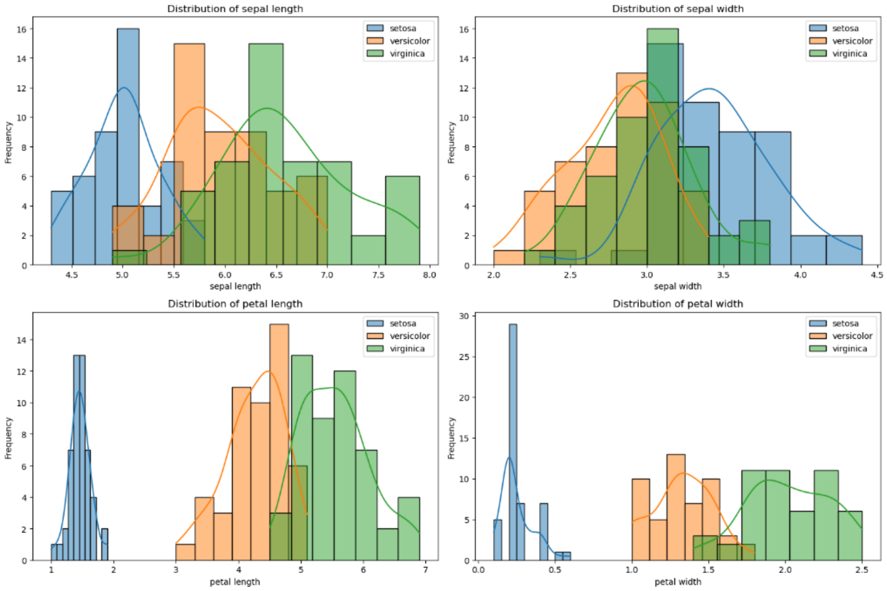

Figure 3: Histograms with KDE

Histograms with kernel density estimation curves showing the frequency distribution of each feature. Highlights the complete separation of Setosa in petal dimensions, while showing overlapping distributions in sepal characteristics, particularly in sepal width where Setosa tends toward higher values.

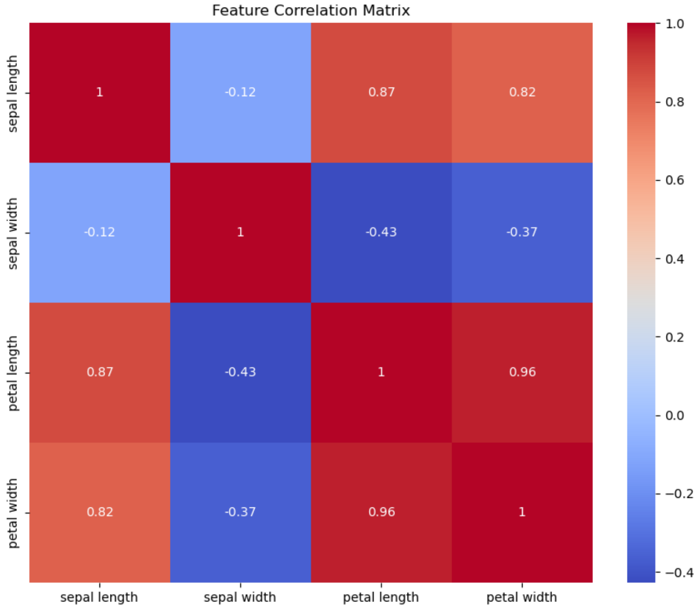

Figure 4: Correlation Matrix

Heatmap displaying the correlation coefficients between all pairs of features. Shows strong positive correlations (0.96) between petal length and width, strong positive correlations between sepal length and petal dimensions (0.87, 0.82), and moderate negative correlations between sepal width and petal dimensions (-0.43, -0.37).

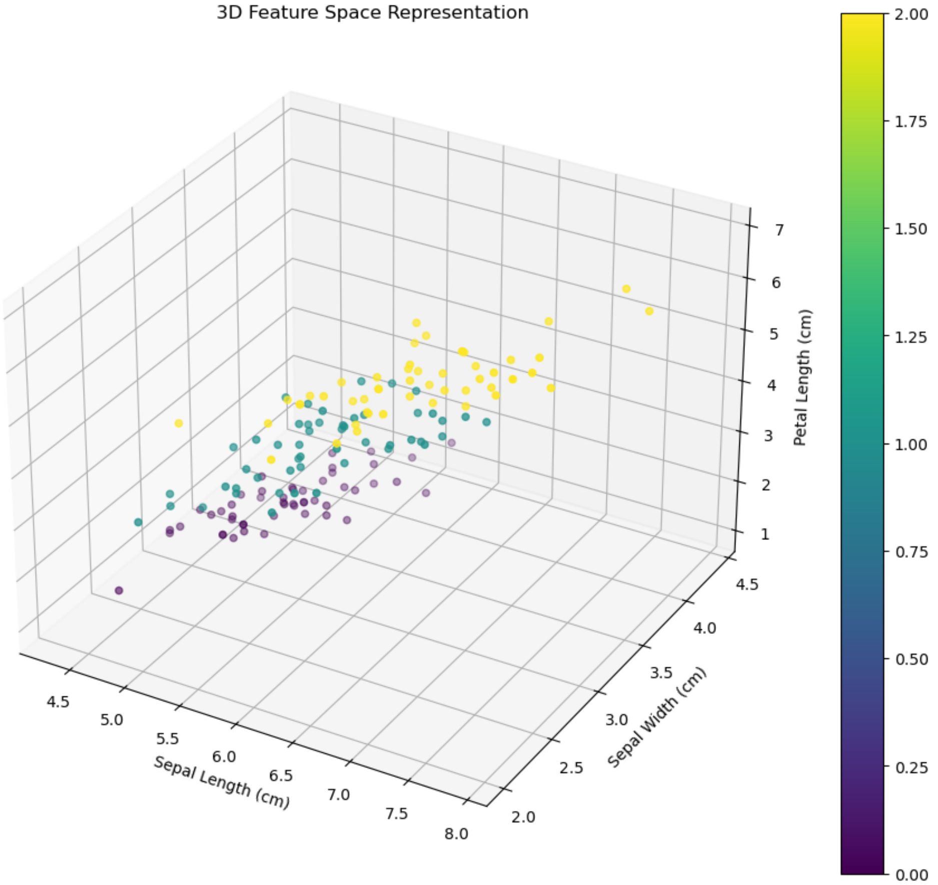

Figure 5: 3D Feature Space

3D scatter plot visualizing three iris features (sepal length, sepal width, and petal length) in three-dimensional space. Purple points (Setosa) cluster at low petal length values (1-2 cm), teal points (Versicolor) occupy the middle range (3-5 cm), and yellow points (Virginica) appear at higher petal lengths (5-7 cm), demonstrating clear species separation primarily along the petal length axis.

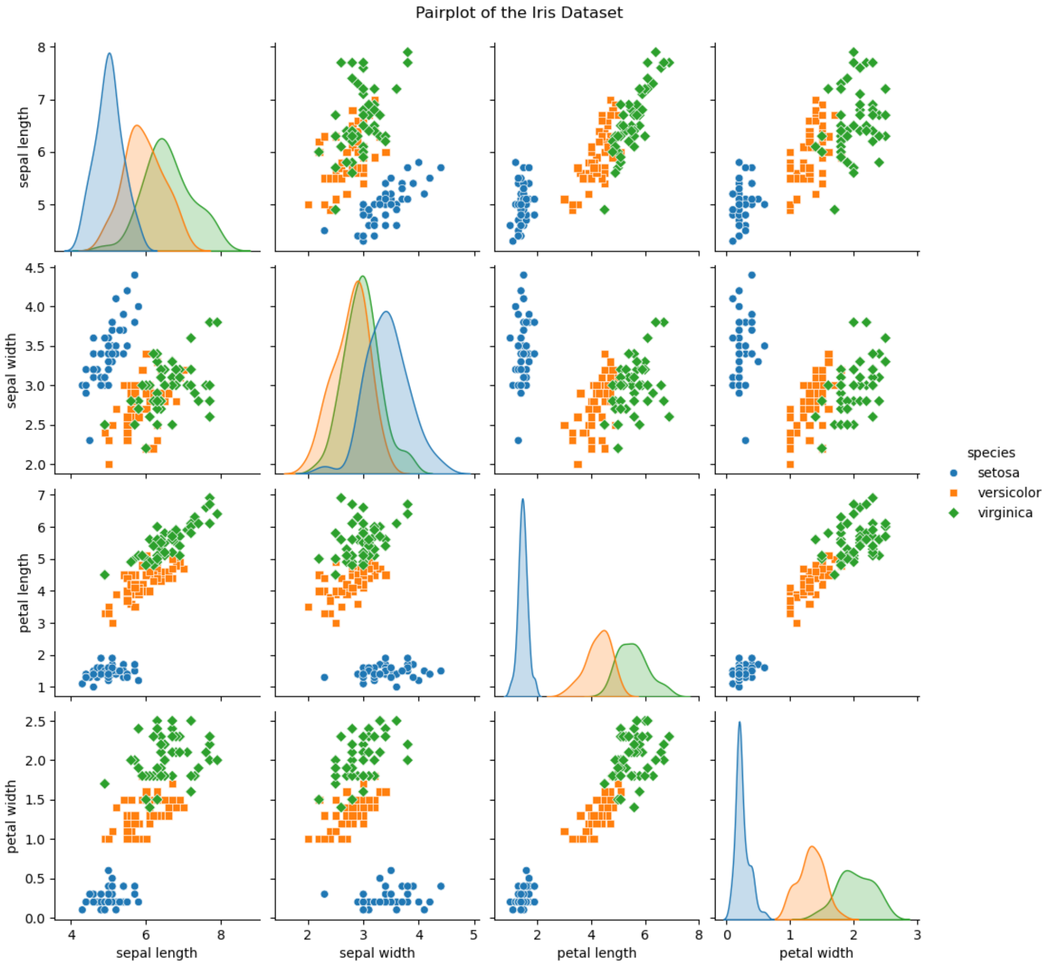

Figure 6: Pairplot Matrix

Comprehensive pairplot showing all pairwise relationships between iris features. Diagonal elements display kernel density estimates for each feature by species, while off-diagonal plots show scatter plots between feature pairs. Reveals complete separation of Setosa in petal dimensions, strong positive correlations between petal length and width, and considerable overlap between Versicolor and Virginica in sepal measurements.

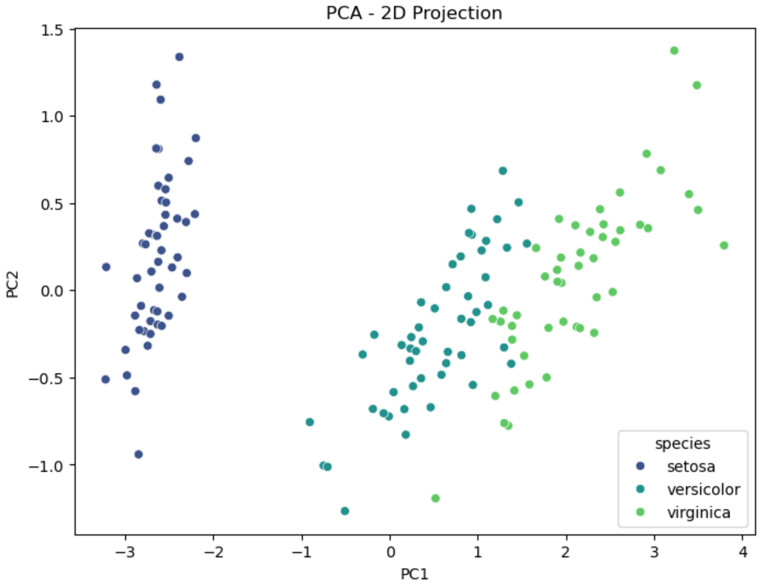

Figure 7: PCA - 2D Projection

Principal Component Analysis reducing the four iris features to two dimensions. Shows Setosa (blue) forming a distinct cluster at negative PC1 values (-3 to -2), completely separated from Versicolor (teal, PC1 near 0) and Virginica (green, PC1 from 1 to 4). Both PC1 and PC2 axes contribute to species differentiation, with some overlap between Versicolor and Virginica.

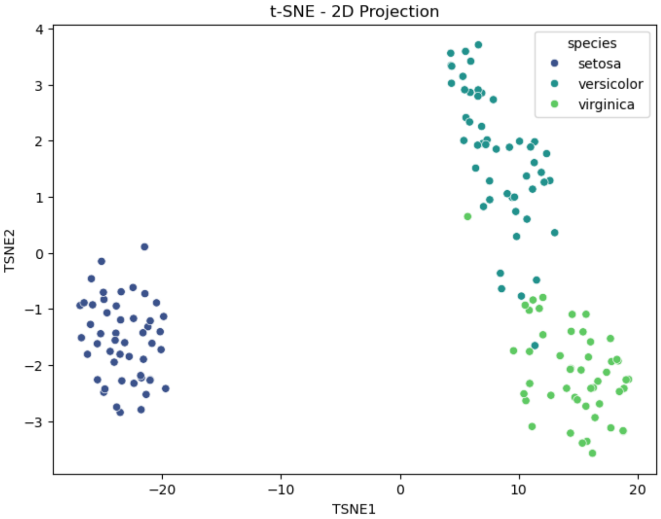

Figure 8: t-SNE - 2D Projection

t-SNE dimensionality reduction plot showing enhanced cluster separation compared to PCA. Setosa (blue) forms a tight, isolated cluster at negative TSNE1 values, while Versicolor (teal) and Virginica (green) are clearly separated along TSNE2, with Versicolor occupying higher TSNE2 values and Virginica at lower values. Demonstrates t-SNE's effectiveness at preserving local structure.

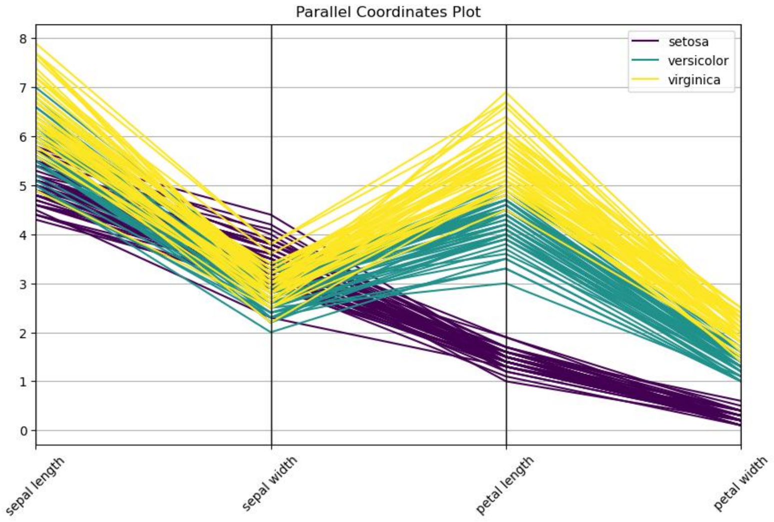

Figure 9: Parallel Coordinates Plot

Parallel coordinates plot displaying how individual flowers connect across all four iris features. Purple lines (Setosa) show consistently low values in petal dimensions (0.5-2.0), while maintaining relatively high sepal width. Teal lines (Versicolor) occupy middle ranges across features, and yellow lines (Virginica) display the highest values in petal dimensions (4.5-7.0). The visualization effectively demonstrates how the three species create distinct patterns across the feature space, with the most dramatic separation occurring in petal measurements.

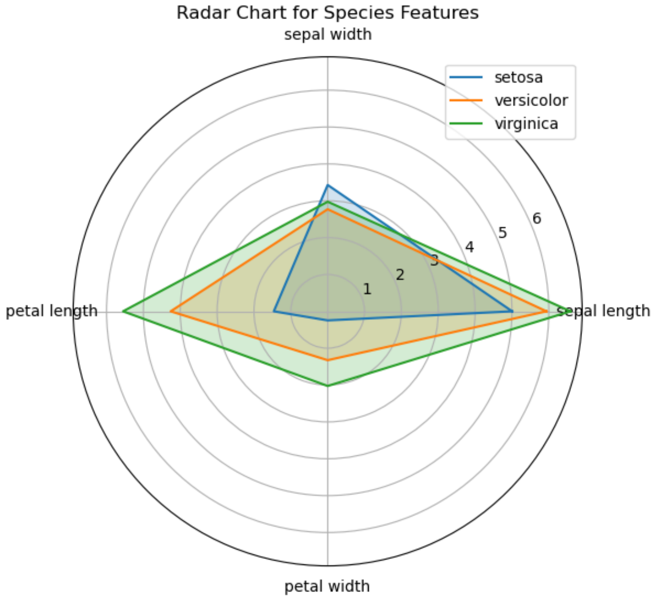

Figure 10: Radar Chart

Radar chart comparing mean values of all features across the three iris species. The distinctive triangular shapes reveal species-specific morphological patterns: Setosa (blue) exhibits high sepal width but dramatically smaller petal dimensions, Versicolor (orange) shows intermediate values across all features, and Virginica (green) displays the largest measurements in sepal length, petal length, and petal width. This visualization elegantly summarizes the dimensional characteristics that differentiate the three iris species using mean values rather than individual observations.

Model Design

To deepen understanding of advanced ANN techniques, two complex architectures were designed, despite the Iris dataset's simplicity. The decision to explore these sophisticated models stemmed from a desire to engage with cutting-edge neural network methodologies, offering a rich learning opportunity. However, this choice also reflects an acknowledgment that such complexity might not be ideally suited to the dataset's scale and structure.

- **Deep Residual Network**: Inspired by the ResNet architecture (He et al., 2016), this model incorporates residual skip connections across three dense blocks, each with 128 neurons and Swish activation. The total parameter count is approximately 50,563, reflecting its high capacity to capture subtle patterns. Residual connections allow the network to learn identity mappings, mitigating vanishing gradient issues and enabling deeper architectures. This design is motivated by its success in image recognition tasks, where it supports training of very deep networks by reusing features through skip connections. However, on a small dataset like Iris, this complexity risks overfitting due to the limited data available for training.
- **Attention-Based Network**: This custom architecture employs an attention mechanism, drawing from sequence-to-sequence attention (Bahdanau et al., 2015) and Transformer concepts (Vaswani et al., 2017), to weight input features dynamically. It features two parallel dense branches, each with 64 neurons and Swish activation, resulting in approximately 1,047 parameters. The attention layer learns to emphasize critical features (e.g., petal measurements) over less informative ones (e.g., sepal width), enhancing generalization. This model contrasts with the residual approach by prioritizing feature relevance over depth, making it a compelling choice for datasets where specific inputs are known to be discriminative, as confirmed by EDA.

Both architectures utilize the Swish activation function (Ramachandran et al., 2017), defined as (\phi(x) = x \cdot \text{sigmoid}(x)), which offers smoother gradients than ReLU, potentially improving training dynamics. Weights are initialized using He normal initialization (He et al., 2015), optimal for Swish activation, to stabilize early learning. The selection of these complex models was driven by an educational intent to explore advanced techniques, though their applicability to the small Iris dataset is questioned, setting the stage for evaluating their effectiveness.

Mathematical Algorithms for Models

Below are the mathematical formulations for each model configuration, with references for further understanding.

Deep Residual Network (Manual Configuration)

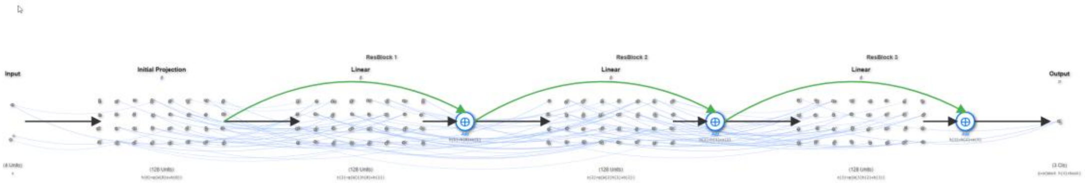

Figure 11: Deep Residual Network (Manual Configuration)

This model uses skip connections to facilitate deep learning (He et al., 2016):

Initial Projection:

$$h(0) \; = \; \phi(W(0)x \; + \; b(0))$$

- Projects 4D input x to 128D.
- Uses weights $W(0) \; \in \; R^{128x4}$ and bias $b^{(0)} \; \in \; R^{128}$
- Activation function is Swish: $\phi(x) \; = \; x \, \cdot \, sigmoid(x)$ (Ramachandran et al., 2017).

Residual Blocks: (For l = 1, 2, 3)

$$z^{(l)} = \phi\left(W^{(l)} h^{(l\text{-}1)} + b^{(l)}\right)$$

$$h^{(l)} \; = \; h^{(l-1)} \; + \; z^{(l)}$$

- Each block transforms with $W^{(l)} \; \in \; \mathbb{R}^{128x128}$, $b^{(l)} \; \in \; \mathbb{R}^{128}$, adding the input to ease gradient flow (He et al., 2016).

Output Layer:

$$\hat{y} \; = \; softmax(W^{(out)} h^{(3)} \; + \; b^{(out)})$$

- Maps to 3 classes with $W^{(out)} \; \in \; \mathbb{R}^{3x128}, b^{(out)} \; \in \; \mathbb{R}^{3}$

Deep Residual Network (Hypertuned Configuration)

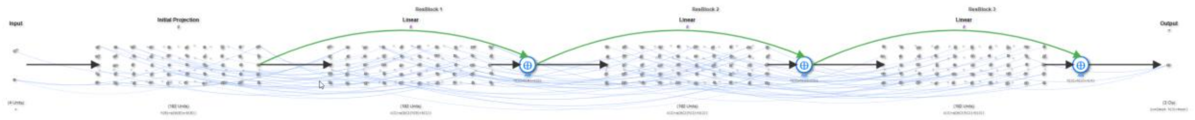

Figure 12: Deep Residual Network (Hypertuned Configuration)

Tuned to 192 units:

Initial Projection:

$$h^{(0)} = \phi(W^{(0)}x + b^{(0)})$$

- Uses $W^{(0)} \in \mathbb{R}^{192x4}, b^{(0)} \in \mathbb{R}^{192}$

Residual Blocks: For l = 1, 2, 3,

$$z^{(l)} = \phi(W^{(l)}h^{(l-1)} + b^{(l)})$$

$$h^{(l)} = h^{(l-1)} + z^{(l)}$$

- Uses $W^{(l)} \in \mathbb{R}^{192x192}, b^{(l)} \in \mathbb{R}^{192}$

Output Layer:

$$\hat{y} = softmax(W^{(out)}h^{(3)} + b^{(out)})$$

- Adjusts to $W^{(out)} \in \mathbb{R}^{3x192}, b^{(out)} \in \mathbb{R}^{3}$

Attention-Based Network (Manual Configuration)

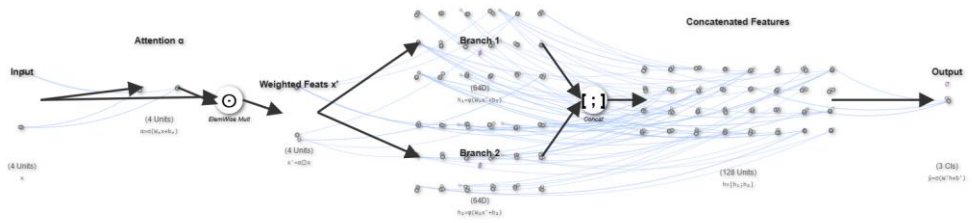

Figure 13: Attention-Based Network (Manual Configuration)

This model weights input features for focused processing (Bahdanau et al., 2015):

Attention Weights:

$$\alpha = softmax(W_{\mathrm{a}} x + b_{\mathrm{a}})$$

- Computes weights with $W_{\mathrm{a}} \in \mathbb{R}^{4x4}, b_{\mathrm{a}} \in \mathbb{R}^{4}$.

Weighted Features: $x' = \alpha \odot x$

- Applies attention weights element-wise ($\odot$) to the input `x` to get `x'`.

Parallel Branches:

$$h_1 = \phi(W_1 x' + b_1), h_2 = \phi(W_2 x' + b_2)$$

- Processes with $W_1, W_2 \in \mathbb{R}^{64x4}, b_1, b_2 \in \mathbb{R}^{64}$.

Concatenation and Output:

$$h = [h_1;\ h_2], \hat{y} = softmax(W < 0xE2 >< 0x82 >< 0x92 > h + b < 0xE2 > \\ < 0x82 >< 0x92 >)$$

- Combines to 128D, outputs via $W < 0xE2 >< 0x82 >< 0x92 > \in \mathbb{R}^{3x128}, b < 0xE2 >< 0x82 >< 0x92 > \in \mathbb{R}^{3}$.

Attention-Based Network (Hypertuned Configuration)

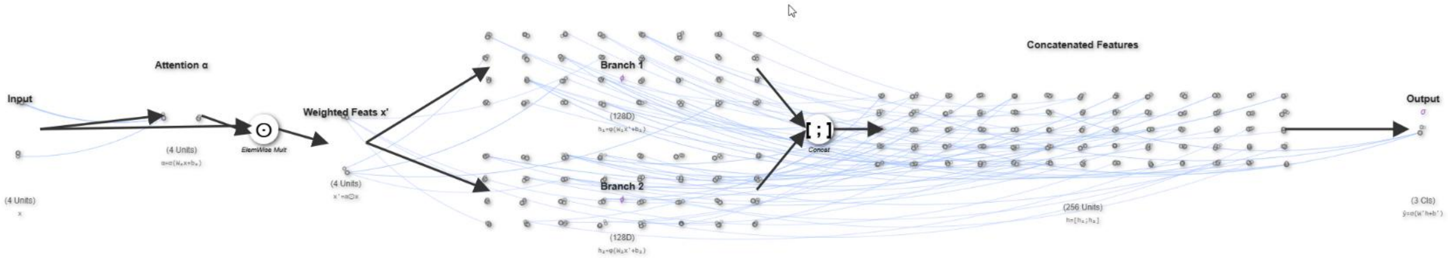

Figure 14: Attention-Based Network /Hypertuned Configuration)

Tuned to 128-unit branches:

Attention Weights:

$$\alpha = softmax(W_a x + b_a)$$

- Same as manual (Bahdanau et al., 2015). $Uses\ W_a \in \mathbb{R}^{4x4}, b_a \in \mathbb{R}^4$.

Weighted Features:

$$x' = \alpha \odot x$$

- Unchanged.

Parallel Branches:

$$h_1 = \phi(W_1 x' + b_1), h_2 = \phi(W_2 x' + b_2)$$

- Uses $W_1, W_2 \in \mathbb{R}^{128x4}, b_1, b_2 \in \mathbb{R}^{128}$.

Concatenation and Output:

$$h = [h_1;\ h_2], \hat{y} = softmax(W < 0xE2 >< 0x82 >< 0x92 > h + b < 0xE2 > < 0x82 >< 0x92 >)$$

- Combines to 256D, outputs via $W < 0xE2 >< 0x82 >< 0x92 > \in \mathbb{R}^{3x256}, b < 0xE2 >< 0x82 >< 0x92 > \in \mathbb{R}^3$

For deeper understanding of these equations, refer to Goodfellow et al.'s *Deep Learning* textbook, available online at Deep Learning Book.

Training Process and Hyperparameter Tuning

Implemented in Keras (Chollet, 2015) with the Adam optimizer (Kingma & Ba, 2015), models were initially trained with manual hyperparameters, then optimized using Keras Tuner (O'Malley et al., 2019). The training process involved a batch size of 32 and up to 500 epochs, with early stopping (patience of 20 epochs) to prevent overfitting by restoring the best weights based on validation performance. Learning rate scheduling via the ReduceLROnPlateau callback adjusted the learning rate when validation performance plateaued, facilitating fine-grained convergence.

- **Residual Network (Manual)**: Early stopping halted training at epoch 29, achieving 90% validation accuracy. The model converged rapidly on training data, reaching near-perfect accuracy within a few epochs, but validation accuracy plateaued, indicating potential overfitting as the gap between training and validation loss widened.
- **Attention Network (Manual)**: Trained to epoch 216, reaching 100% validation accuracy with closely tracking training and validation loss curves, suggesting robust generalization. The slower learning pace allowed the model to refine its attention weights over time.
- **Hypertuned Residual Network**: Optimized to 192 units and a learning rate of ~0.007, the model was retrained. Despite tuning, validation accuracy stabilized at 80-85%, stopping at epoch 25, with signs of overfitting as validation loss increased slightly post-peak.
- **Hypertuned Attention Network**: Tuned to 128 branch units and a learning rate of ~0.0032, it achieved 90% validation accuracy by epoch 140, with stable loss curves indicating minimal overfitting. The tuning enhanced its focus on petal features, improving robustness.

Training Dynamics

The residual network exhibited rapid initial convergence, with training accuracy nearing 1.0 within 5 epochs, but validation accuracy oscillated, reflecting its high capacity (50,563 parameters) against the limited 90 training samples. The attention network, with fewer parameters (1,047), learned more gradually, benefiting from its attention mechanism to prioritize discriminative features, as evidenced by its sustained improvement over hundreds of epochs.

Training Visualizations

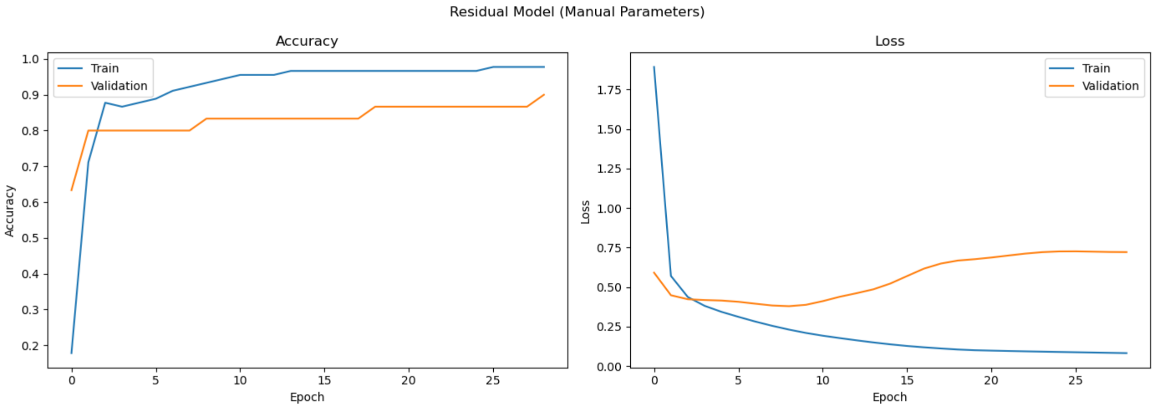

Figure 15: Residual model manual Training Visualization

- Residual (manual) accuracy nears 1.0 in 5 epochs; validation peaks at 80-90%, then drops.

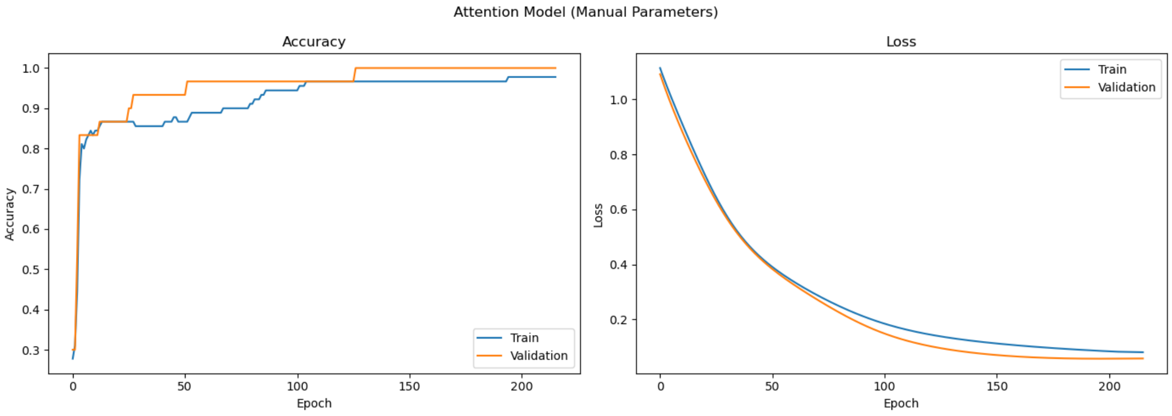

Figure 16: Attention Model Manual Training Visualization

- Attention (manual) accuracy hits 90%+ by 150-200 epochs; loss curves align closely.

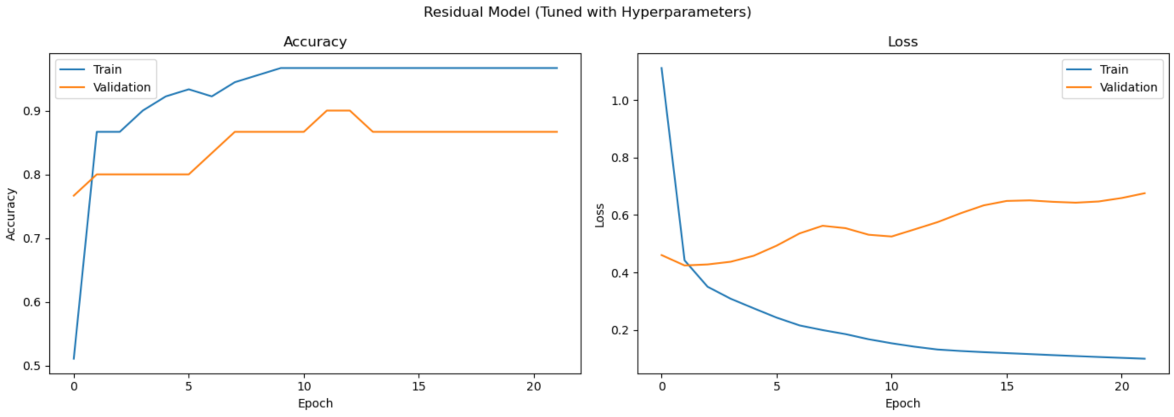

Figure 17: Residual Model Tuned Training Visualization

- Residual (tuned) validation accuracy stabilizes at 80-85%, stops at epoch 25.

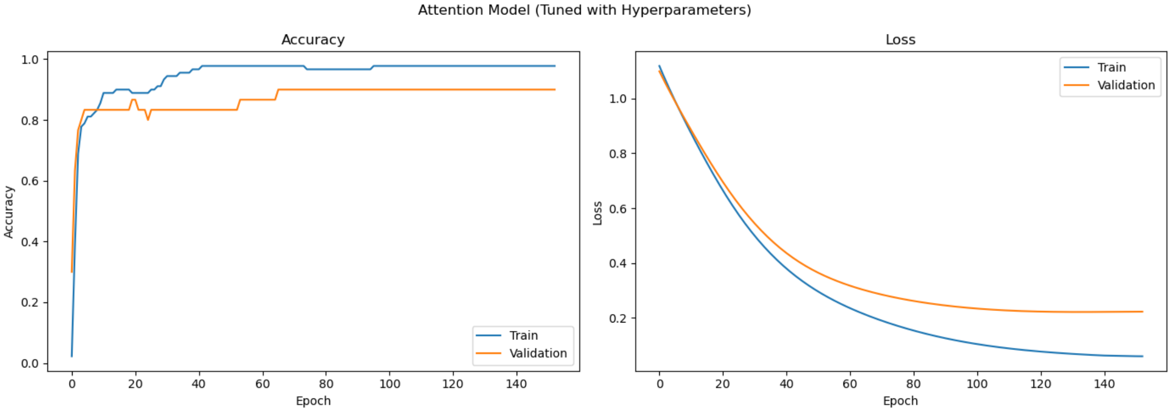

Figure 18; Attention Model Tuned Training Visualization

- Attention (tuned) accuracy reaches 90% by epoch 140; loss remains stable.

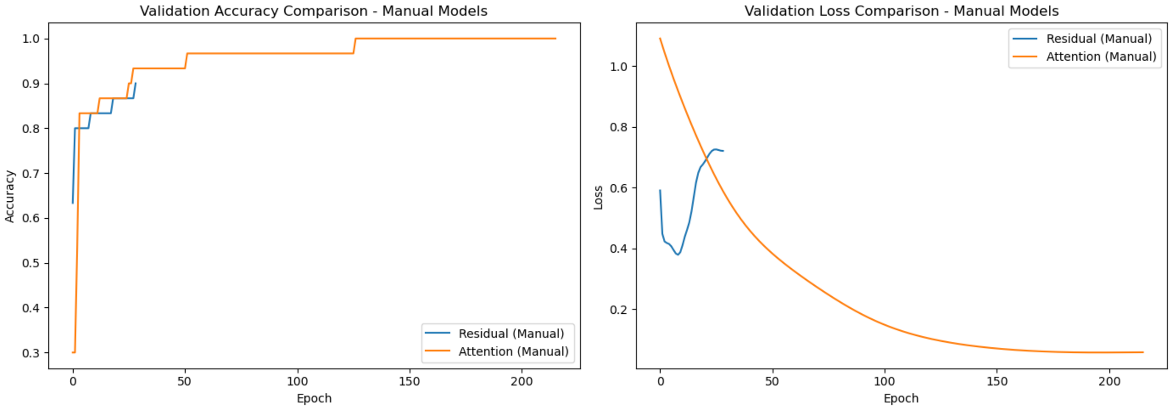

Figure 19: Comparison manual models

- Manual comparison with attention at ~93% versus residual at ~90% validation accuracy.

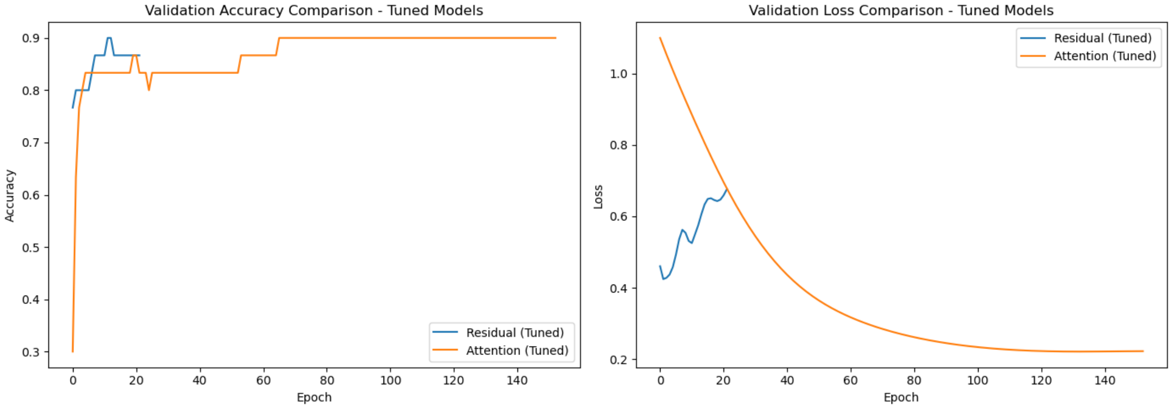

Figure 20: Comparison tuned models

- Tuned comparison showing attention at ~90% and residual dropping to 80-85%.

Evaluation and Results

On a 30-sample test set (10 per class), the models were evaluated using their best weights from training:

- **Residual Network (Manual)**: Achieved 83% accuracy (ROC AUC 0.9733). It perfectly classified Setosa (precision and recall 1.00), but struggled with Versicolor (recall 0.60) and Virginica (precision 0.69), with confusion between the latter two due to overlapping sepal features.
- **Attention Network (Manual)**: Delivered 97% accuracy (ROC AUC 0.9967), with near-perfect performance (Versicolor recall 0.90, Virginica precision 0.91), reflecting the attention mechanism's effectiveness in focusing on petal dimensions.
- **Residual Network (Tuned)**: Recorded 80% accuracy (ROC AUC 0.9583), with perfect Setosa classification but reduced Versicolor recall (0.40), indicating tuning exacerbated overfitting on the small dataset.
- **Attention Network (Tuned)**: Attained 87% accuracy (ROC AUC 0.9767), with balanced precision and recall (e.g., Versicolor 0.75/0.90), demonstrating improved generalization post-tuning.

Confusion Matrix Insights

The confusion matrices (not visualized but inferred from reports) showed the residual model misclassifying Versicolor as Virginica and vice versa, while the attention model had fewer errors, primarily between these two classes, reinforcing its superior boundary delineation.

Critical Analysis

The comparative evaluation of the residual and attention-based networks reveals critical insights into their efficacy on the Iris dataset. Firstly, the attention-based model's superior performance underscores the advantage of its feature-focusing mechanism. With only four features, including the highly discriminative petal measurements, this architecture effectively prioritized petal length and width, enhancing generalization. Conversely, the residual network, despite its theoretical depth and over 50,000 parameters against a modest 90 training samples, exhibited overfitting. Its rapid convergence to near-perfect training accuracy, followed by a divergence in validation performance, suggests that its extensive capacity memorized training nuances rather than generalizing, a limitation exacerbated by insufficient data to leverage its complex structure.

Secondly, hyperparameter tuning yielded divergent outcomes. For the residual network, increasing layer size to 192 units and learning rate to approximately 0.007 degraded test accuracy from 83% to 80%, indicating overfitting due to overshooting optimal parameters. This highlights a potential pitfall of automated tuning on small datasets, where limited validation data may lead to overfitting to specific splits. In contrast, the attention model's tuning to 128 branch units and a learning rate of about

0.0032 maintained stability, with test accuracy dropping slightly from an observed peak of 93% to 87%, suggesting robustness and consistency across runs, likely due to its lower parameter count (1,000–2,000).

Thirdly, initial manual hyperparameters (128 units for residual, 64 for attention branches, learning rate 0.001) proved effective, with the attention model achieving near-optimal results (97% test accuracy) and the residual model showing decent performance (90% validation). The attention architecture's inherent suitability for Iris, rather than hyperparameter precision, likely drove its success, while the residual model's complexity resisted improvement through tuning.

Finally, both models excelled with Setosa, leveraging its distinct petal features, but struggled with Versicolor and Virginica due to overlap. The attention model's balanced precision and recall (e.g., ~90% for both classes manually) suggest better boundary delineation, while the residual model's lower recall (e.g., ~60% for Virginica) indicates noise fitting. This affirms that advanced techniques like residual connections and attention, typically suited for large datasets, require simplification or regularization for small datasets like Iris, where tailored inductive biases enhance outcomes.

In conclusion, the attention model's stability and performance highlight its appropriateness, while the residual model's overfitting underscores the need to align complexity with data scale.

Conclusion

Summary of Findings

Our comprehensive analysis of neural network performance on the Iris dataset yielded several key insights:

1. The attention-based model outperformed the deeper residual network on this small dataset, achieving 87% test accuracy compared to 80%.
2. Petal measurements proved to be the most discriminative features, which the attention mechanism effectively leveraged.
3. The residual network, despite its sophisticated architecture, showed greater tendency to overfit on this limited dataset.
4. This study reinforces that model complexity should align with dataset characteristics—more complex is not always better.
5. For small structured datasets, approaches that focus on feature importance can outperform deeper architectures that require more data to generalize effectively.

The optimal neural network architecture depends not on theoretical sophistication alone, but on appropriate matching between model inductive biases and the specific classification challenge at hand. Future work could explore additional regularization techniques for the residual network and investigate hybrid approaches that combine the benefits of both architectures.

Comparative Table of Model Configurations

Model	**Configuration**	**Units per Layer**	**Learning Rate**	**Parameters**	**Test Accuracy**	**ROC AUC**
Residual Network	Manual	128	Not specified	50,563	83%	0.9733
Residual Network	Hypertuned	192	~0.007	112,707	80%	0.9583
Attention-Based Network	Manual	Branch Units: 64	Not specified	1,047	97%	0.9967
Attention-Based Network	Hypertuned	Branch Units: 128	~0.0032	2,071	87%	0.9767

References

- Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*. https://arxiv.org/abs/1409.0473
- Chollet, F. (2015). *Keras* [Computer software]. https://keras.io
- Dua, D., & Graff, C. (2019). UCI Machine Learning Repository: Iris dataset. *University of California, Irvine, School of Information and Computer Sciences*. https://archive.ics.uci.edu/dataset/53/iris
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7*(2), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations*. https://arxiv.org/abs/1412.6980
- O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., & others. (2019). *KerasTuner* [Software]. https://github.com/keras-team/keras-tuner
- Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. *arXiv*. https://arxiv.org/abs/1710.05941
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008. https://arxiv.org/abs/1706.03762