Benchmarking Transformer & Performer Architectures as Visual Encoders for Image Classification

Joshwin Rajan (jtr2157)
Pinqiao Wang (pw2594)
Satyabrat Srikumar (ss7172)

Fu Foundation School of Engineering and Applied Science
Columbia University in the City of New York
Department of Industrial Engineering & Operations Research

Course: IEOR E4540 - Data Mining
Instructor: Prof. Krzysztof Choromanski
May 1, 2024

Contents

1. Introduction

In recent years, the field of deep learning has witnessed remarkable advancements, catalyzing breakthroughs across a myriad of domains including natural language processing (NLP), computer vision, and beyond. Among the pivotal innovations driving this progress are Transformer architectures, originally introduced by Vaswani et al. in the seminal work "Attention is All You Need" [1]. Transformers have revolutionized NLP tasks by dispensing with recurrent neural networks (RNNs) and instead relying on self-attention mechanisms, enabling parallelization and capturing long-range dependencies with unparalleled efficacy.

However, the transformative potential of Transformers extends beyond NLP, with the burgeoning interest in their application to computer vision tasks, particularly image classification. Concurrently, a variant of the Transformer architecture known as the Performer, introduced by Choromanski et al. [2], has garnered attention for its computational efficiency and scalability in handling large datasets.

This project endeavors to explore and benchmark the efficacy of Transformer and Performer architectures as visual encoders for image classification tasks. Specifically, we aim to evaluate various Performer architectures alongside local-attention Transformer architectures, encompassing diverse receptive fields. The benchmarking encompasses several widely used datasets, including MNIST [3], CIFAR10 [4], ImageNet [5], and Places365 [6], representing varying degrees of complexity and scale in image classification.

The Performer architectures under scrutiny include the Performer-ReLU and Performer-approximate-softmax variants, with the latter employing different numbers of random features (m = 16, 32, 64, 128). Leveraging these variants enables a comprehensive assessment of their performance across different configurations, shedding light on their suitability for diverse image classification tasks.

Central to our investigation is the evaluation of model accuracy on validation datasets, serving as a primary metric for performance comparison. Additionally, we aim to assess the computational efficiency of each architecture, encompassing training and inference speeds. This holistic evaluation framework provides insights not only into the classification performance but also the practical viability of each architecture in real-world applications.

A crucial aspect of our investigation pertains to the implementation of local attention mechanisms within the Transformer architectures. We explore whether local attention is implemented concerning the 2D original structure of the input or its 1D flattened variant. This choice fundamentally influences the model's ability to capture spatial dependencies within the images, thus warranting meticulous examination.

This project endeavors to contribute to the burgeoning literature on Transformer and Performer architectures in the realm of computer vision, offering empirical insights into their efficacy as visual encoders for image classification tasks. Through rigorous benchmarking and analysis, we aim to elucidate the strengths and limitations of each architecture, paving the way for informed architectural choices in future research and practical applications.

2. Transformer Models

The Transformer architecture, originally proposed by Vaswani et al. [1], represents a pivotal advancement in deep learning, particularly in sequence modeling tasks. While initially devised for natural language processing (NLP) tasks, Transformers have garnered significant interest for their potential application in computer vision tasks, including image classification.

2.1 Self-Attention Mechanism

At the core of the Transformer lies the self-attention mechanism, enabling each element in the input sequence to attend to all other elements simultaneously. This mechanism facilitates the capture of long-range dependencies across the input sequence, crucial for tasks such as image classification where spatial relationships between pixels or features are essential.

The mathematical formulation for self-attention is as follows:

$$Q = XW^Q, \tag{1}$$

$$K = XW^K, \tag{2}$$

$$V = XW^V \tag{3}$$

Thus,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

2.2 Multi-Head Attention

To enhance representational capacity and enable the model to attend to different aspects of the input sequence, Transformer architectures incorporate multi-head attention layers. These layers allow the model to learn diverse patterns and relationships within the input data, contributing to improved classification performance across various datasets.

With h parallel attention layers or "heads," the multi-head attention is calculated as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{5}$$

$$\text{Head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{6}$$

2.3 Positional Encoding

Incorporating positional encoding is critical for Transformer models applied to image classification tasks. While images inherently possess spatial information, the absence of explicit sequential order necessitates the incorporation of positional encodings to provide the model with information about the relative positions of pixels or features within the image.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{7}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{8}$$

2.4 Transformer Encoder for Image Classification

For image classification tasks, the Transformer encoder serves as the primary component of the architecture. Stacked layers of self-attention and feed-forward neural networks enable the model to capture hierarchical representations of the input image, leveraging both local and global contextual information for accurate classification.

$$\text{LayerNorm}(x + \text{Sublayer}(x))$$

3. Performer Models

The Performer architecture, introduced by Choromanski et al. [2], represents a novel approach to sequence modeling that offers computational efficiency and scalability, making it an attractive candidate for various machine learning tasks, including image classification. Similar to Transformers, Performers rely on self-attention mechanisms to capture dependencies across input sequences. However, Performer models leverage approximations and efficient computation techniques to achieve scalability, making them well-suited for large-scale image classification tasks.

3.1 Random Feature Approximation

A key feature of Performer models is the use of random feature approximation, which replaces the standard dot-product attention with a low-rank approximation. The Performer modifies the attention mechanism by using a kernelized feature map ϕ that allows computation in a lower-dimensional space:

$$\text{Attention}(Q, K, V) \approx \text{Softmax}\left(\frac{\phi(Q)\phi(K)^T}{\sqrt{d_k}}\right) V$$

This approximation significantly reduces the computational complexity, especially for long sequences.

3.2 Efficient Computation Techniques

Performer models employ efficient computation techniques such as Fast Fourier Transform (FFT) and kernelized attention mechanisms to further accelerate inference and training. The use of FFT helps in the efficient computation of the transformed features:

$$\phi(Q) = \text{FFT}(Q), \quad \phi(K) = \text{FFT}(K)$$

These transformed features are then used in the attention computation to speed up the process while maintaining a degree of accuracy.

3.3 Local Attention in Performer

In the context of image classification, Performer models can incorporate local attention mechanisms to capture spatial dependencies within images efficiently. The local attention can be mathematically described as:

$$\text{LocalAttention}(Q, K, V) = \sum_{i \in \text{Local Region}} \text{Softmax}\left(\frac{QK_i^T}{\sqrt{d_k}}\right) V_i$$

By attending to local regions of the input image, Performers effectively extract relevant features for classification with minimized computational overhead.

3.4 Performer Encoder for Image Classification

The Performer encoder serves as the core component of the architecture for image classification tasks. By stacking layers of efficient self-attention and feed-forward networks, Performers can learn hierarchical representations of input images:

$$\text{Layer}(X) = \text{LayerNorm}(X + \text{PerformerAttention}(X))$$

3.5 Performer Variants and Extensions for Image Classification

Recent research has explored various extensions and variants of the Performer architecture for image classification tasks. These variants, including Performer-ReLU and Performer-approximate-softmax, offer different trade-offs in terms of computational efficiency and accuracy:

$$\text{PerformerReLU}(X) = \max(0, XW^1 + b^1)W^2 + b^2$$

Performer models offer a promising alternative to traditional Transformer architectures for image classification tasks, leveraging efficient computation techniques and scalable self-attention mechanisms. By incorporating local attention and efficient approximation methods, Performers can achieve competitive performance on image classification benchmarks while maintaining computational efficiency.

4. Attention Mechanisms: Local VS Global

4.1 Global Attention Mechanism

Global Attention Mechanisms allow each element in the input sequence to attend to all others, capturing long-range dependencies essential for understanding the global context in tasks like scene recognition. This is achieved through the standard attention formula:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

4.2 Benefits and Trade-offs

Local attention improves computational efficiency and interpretability by focusing on spatial details but may miss long-range dependencies. Global attention captures a comprehensive global context but at a higher computational cost. Multi-Head Attention enhances representational capacity by learning diverse aspects of the data. In image classification, integrating both local and global attention mechanisms allows models to effectively extract and utilize both fine-grained features and global contextual information, enhancing classification accuracy and robustness. Recent variants adapt these mechanisms to dynamically adjust to the input, optimizing performance and computational efficiency.

The Swin Transformer architecture introduces a novel approach to managing the computational demands of traditional transformers by implementing a method called "Shifted Window" self-attention. This technique restricts attention to localized windows while enabling connections across different windows at deeper layers.

4.3 Innovations in Swin Transformers

The main innovation in Swin Transformers lies in their ability to constrain the computation of attention to localized, non-overlapping windows. These windows shift position in successive layers, which helps to capture broader contextual details with less computational burden than global attention methods.

4.4 Essential Elements of Swin Transformers

1. **Hierarchical Structure:**

    - **Layered Architecture:** Inputs pass through multiple layers, each operating at a different scale. Early layers focus on detailed features in smaller areas, while later layers merge these areas to analyze broader features.

- **Adaptive Scaling:** The network adapts its scale dynamically, offering a multi-scale view of inputs. This flexibility ensures efficiency and effectiveness across varying image resolutions and complexities.

2. **Shifted Window Mechanism:**
   - **Initial Window Division:** The architecture starts by dividing the input into discrete, non-overlapping windows. Each window processes self-attention independently, which minimizes computation compared to full-frame attention.
   - **Window Shifting:** As layers progress, windows are shifted to partially overlap with prior windows. This design allows the model to assimilate information from neighboring windows, improving its ability to discern distant feature relationships without directly computing global attention.
3. **Relative Position Bias:**
   - **Positional Encoding Variations:** Swin Transformers use learnable relative position biases instead of the absolute positional encodings typical in conventional transformers. This change enhances the model's understanding of spatial relationships within windows, a critical aspect for tasks dependent on spatial context.
4. **Efficient Attention Computation:**
   - **Scaled Dot-Product Attention:** Within each window, attention is computed using a scaled dot-product method, optimizing memory use and focusing attention effectively within confined areas.
   - **Optimized Dimensionality:** The architecture manages the dimensionality of the query, key, and value vectors to streamline processing speed and minimize memory requirements, maintaining scalability and efficiency.

5. Datasets

In this project, we assess the efficacy of Transformer and Performer architectures as visual encoders for image classification tasks across four distinct datasets. Each dataset offers unique challenges and characteristics that provide a comprehensive framework for benchmarking different architectures.

5.1 Datasets Used

1. **MNIST (Modified National Institute of Standards and Technology)**: This dataset includes grayscale images of handwritten digits (0-9) with a resolution of 28x28 pixels. It comprises 60,000 training images and 10,000 test images, serving as a benchmark for small-scale, simple image recognition tasks.

2. **CIFAR-10 (Canadian Institute for Advanced Research - 10)**: CIFAR-10 features 60,000 color images across 10 classes with a resolution of 32x32 pixels. It poses greater challenges than MNIST due to its color complexity, making it suitable for medium-scale, multi-class image classification tasks.

3. **CIFAR-100 (Canadian Institute for Advanced Research - 100)**: This dataset includes 60,000 color images in 100 classes, with each class containing 600 images. The images are grouped into 20 superclasses, with each image labeled with both a "fine" (specific) and "coarse" (general) label. Each image has a resolution of 32x32 pixels. The dataset comprises 50,000 training images and 10,000 testing images, making it suitable for evaluating the performance of image classification models on a medium-scale multi-class classification task with higher complexity due to the increased number of classes and hierarchical labeling system.

4. **ImageNet**: Utilizing a subset of ImageNet that contains 1.2 million training images and 50,000 validation images across 1,000 classes, this dataset is a large-scale benchmark for evaluating models on fine-grained image classification tasks. In reality, ImageNet dataset could be too large to be experimented upon. Thus, in this task we only used a subset of the entire dataset, we will call it tiny ImageNet.

5. **Places365**: Designed specifically for scene recognition tasks, Places365 includes over 1.8 million images across 365 scene categories. It challenges models to recognize and understand complex spatial relationships and semantic concepts within diverse real-world scenes.

5.2 Dataset Characteristics and Challenges

Each dataset presents unique characteristics and challenges crucial for evaluating the performance of Transformer and Performer architectures:

- **MNIST** is ideal for testing basic model capabilities and generalization to unseen data.

- **CIFAR-10** tests model robustness and generalization in more complex scenarios due to higher resolution and color complexity.

- **ImageNet and Places365** Due to an unexpected shortage of computing power and units caused by protests, we are unable to execute the code and thus train the transformer and performer on these datasets for inference purposes. We have contacted the University's computer resource, yet we will not be able to have enough time to make this happen before the deadline. With that being said, in future work, we would still look forward to training our models on those datasets when we have access to school utilities again.

By assessing models across these datasets, we can derive insights into their performance over various scales and complexities of image classification tasks, enabling a comprehensive evaluation of their real-world application efficacy and generalization capabilities.

6. Implementation Details

In this project, we implemented and evaluated various architectures using the TensorFlow framework in Google Colab for image classification tasks on the MNIST and CIFAR-10 datasets. The architectures deployed are listed as follows:

- Performer ReLU (for 16, 32, 64, 128, 256 random features)
- Performer Softmax (for 16, 32, 128, and 264 random features)
- Transformer with Global Attention
- Transformer with Local Attention

In this analysis, we explore the implementation of local attention mechanisms concerning either the two-dimensional structure or the one-dimensional flattened format of the input. Opting for a 2D structure-based local attention, our model preserves the spatial relationships of the input images. This method not only recognizes limited regions defined by a spatial receptive field to capture intricate spatial dependencies effectively but also enhances the model's capability to extract vital local features for classification tasks. This approach is crucial for image classification, where understanding spatial relationships is fundamental to accuracy. While this method significantly leverages the spatial structure inherent in the images, it also increases computational demands. Therefore, it is vital to balance this approach with other architectural features depending on the task's specific needs and available resources, aiming to optimize performance without compromising the model's depth of spatial understanding and feature extraction capabilities.

When implementing local attention concerning the 2D original structure of the input, the model explicitly considers the spatial relationships between pixels or features within the input images. Each position in the output sequence attends to a limited region defined by a spatial receptive field, capturing fine-grained spatial dependencies within the input image. This approach enables the model to extract local features and relationships effectively, leveraging the inherent spatial structure of the input images for classification tasks.

Alternatively, local attention can be implemented concerning the 1D flattened variant of the input, where the spatial structure of the input images is flattened into a one-dimensional sequence. In this approach, the model attends to local regions within the flattened sequence, treating each position as an element in the input sequence. While this approach simplifies the computation and may reduce the model's computational complexity, it may also limit the model's ability to capture spatial dependencies and extract fine-grained features from the input images.

In our implementation, we have chosen to implement local attention concerning the 2D original structure of the input. By explicitly considering the spatial relationships between pixels or features within the input images, we aim to capture fine-grained spatial dependencies and extract relevant local features for classification tasks. This approach aligns with the nature of image classification tasks, where spatial relationships between pixels or regions are essential for accurate classification.

6.1 Advantages and Considerations:

Implementing local attention concerning the 2D original structure of the input offers several advantages:

- Captures fine-grained spatial dependencies: By considering the spatial relationships within the input images, the model can capture fine-grained spatial dependencies and extract relevant local features for classification.

- Leverages spatial structure: Exploiting the inherent spatial structure of the input images enables the model to better understand and interpret spatial relationships between pixels or regions, enhancing its classification performance.

- However, it's essential to consider the computational overhead associated with processing 2D input structures, especially for large-scale images. Depending on the specific requirements of the task and the computational resources available, it may be necessary to balance the use of local attention with other architectural components to achieve optimal performance.

- Our implementation of local attention concerning the 2D original structure of the input allows us to capture fine-grained spatial dependencies and extract relevant local features effectively for image classification tasks, aligning with the requirements of real-world applications.

7. Evaluation Metrics - Speed and Accuracy

We have measured and compared the models on several metrics: time per epoch, time per step, training accuracy, training loss, validation accuracy, and validation loss altogether. These metrics provide an understanding of how accurately the models can categorize images and how quickly they can do so. The findings are summarized in two tables, one for each dataset. Below the tables, we have included several graphs visualizing the findings.

Table 1: CIFAR-10 Performance Comparison.

	Accuracy[1]			Speed[2]		
Model	Training	Validation	Val-Loss	Epochs	Time/Epoch	Time/Step
Performer (Relu16)	0.8115	0.7072	0.9279	60	20s	104ms/step
Performer (Relu32)	0.7826	0.7005	0.9184	60	20s	106ms/step
Performer (Relu64)	0.8017	0.6856	0.9999	60	19s	102ms/step
Performer (Relu128)	0.8060	0.7306	0.8236	60	19s	102ms/step
Performer (Softmax16)	0.8462	0.7748	0.6970	60	14s	74ms/step
Performer (Softmax32)	0.8442	0.7834	0.7103	60	14s	74ms/step
Performer (Softmax64)	0.8471	0.7699	0.7226	60	14s	74ms/step
Performer (Softmax128)	0.8426	0.7714	0.7012	60	14s	74ms/step
Transformer Global	0.9568	0.7266	0.9910	60	24s	137ms/step
Transformer Local (Swin)	0.8078	0.7242	1.1881	50	6s	17ms/step

[1] Accuracy Metrics for model

[2] Speed Metrics for training and inference

Table 2: MNIST Performance Comparison.

	Accuracy[1]			Speed[2]		
Model	Training	Validation	Val-Loss	Epochs	Time/Epoch	Time/Step
Performer (Relu16)	0.9860	0.9908	0.0288	60	7s	30ms/step
Performer (Relu32)	0.9865	0.9904	0.0283	60	7s	30ms/step
Performer (Relu64)	0.9864	0.9896	0.0341	60	7s	30ms/step
Performer (Relu128)	0.9872	0.9932	0.0251	60	7s	29ms/step
Performer (Softmax16)	0.9864	0.9919	0.0291	60	7s	30ms/step
Performer (Softmax32)	0.9853	0.9924	0.0260	60	8s	33ms/step
Performer (Softmax64)	0.9856	0.9930	0.0241	60	7s	29ms/step
Performer (Softmax128)	0.9854	0.9916	0.0285	60	8s	32ms/step
Transformer Global	0.9872	0.9898	0.0278	60	8s	32ms/step
Transformer Local (Swin)	0.3008	0.3160	2.0161	50	3s	8ms/step

[1] Accuracy Metrics for model

[2] Speed Metrics for in training and inference

7.1 Major Inference: Tables

CIFAR-10 Performance Comparison

Accuracy Metrics:

- **Performer Models:** The Performer models with ReLU activation (Relu16, Relu32, Relu64, Relu128) show a gradual increase in validation accuracy as the model complexity increases, although they maintain fairly similar training accuracies around 0.80. Notably, the validation losses decrease as the model size increases, suggesting better generalization with higher dimensionality.

- **Performer with Softmax:** Transitioning to softmax activation significantly boosts both training and validation accuracy, indicating that softmax may be better at handling the complexities of CIFAR-10's image data. The softmax variants consistently show higher validation accuracies compared to their ReLU counterparts, with Softmax32 peaking at 0.7834.

- **Transformer Models:** The Global Transformer achieves the highest training accuracy (0.9568) but a moderate validation accuracy (0.7266), indicating potential overfitting. The Local Transformer (Swin) provides a balanced accuracy profile but with the highest validation loss among all models, suggesting it might be underfitting or too simplistic for the dataset.

Speed Metrics:

- **Time Efficiency:** Softmax variants of the Performer are notably faster, with each epoch taking only 14 seconds and steps processing in 74ms. This contrasts with the slower Global Transformer, which takes 24 seconds per epoch and 137 ms per step.

- **Local Transformer Efficiency:** The Swin Transformer excels in speed, taking only 6 seconds per epoch and 17ms per step, demonstrating its efficiency in local processing, although this does not translate to high accuracy.

MNIST Performance Comparison

Accuracy Metrics:

- **High Accuracy Across Models:** All models achieve high training and validation accuracies on MNIST, indicative of the dataset's simpler nature. Performer(Relu128) and Transformer Global achieve particularly high validation accuracies around 0.9932 and 0.9898 respectively, demonstrating their effective learning capabilities on simpler image tasks.

- **Validation Losses:** Performer models with ReLU and softmax activations show very low validation losses, with Performer(Relu128) recording the lowest at 0.0251, suggesting excellent model fit and generalization on this dataset.

Speed Metrics:

- **Consistent Speeds:** Most models process epochs in around 7-8 seconds and steps in 29-33ms, indicating that MNIST, with its less complex images, does not require extensive computational resources.
- **Exceptional Case of Transformer Local (Swin):** The Swin model shows an exceptionally low training time (3s per epoch) and per-step time (8ms), but it also shows very poor accuracy, highlighting a trade-off between speed and performance accuracy in this specific architecture.

Comparative Insights

- **Model Suitability:** Softmax activation in Performer models appears more suited to CIFAR-10, handling complex patterns more effectively than ReLU. For MNIST, both activations work effectively, showing little difference in performance.
- **Efficiency vs. Accuracy:** The tables highlight a general trade-off between efficiency and accuracy, especially evident in the Local Transformer (Swin) model's performance on both datasets.
- **Overfitting vs. Underfitting:** The Global Transformer's performance on CIFAR-10 suggests potential overfitting, while the Local Transformer might be underfitting, as indicated by its high validation loss despite its speed.

7.2 Major Inference: Figures

According to Figures A6 to A9 in the Appendices section, all models exhibit a consistent decrease in training loss, indicative of learning. Conversely, test loss tends to plateau or increase, suggesting potential overfitting issues. This behavior underscores the necessity of monitoring both training and test metrics to gauge the model's ability to generalize. While training accuracy consistently improves, test accuracy does not show proportional gains and often stabilizes. This plateau indicates that the models might be overfitting the training data, thus failing to improve generalization on unseen data. The "best epoch," marked by a vertical line in each plot, highlights when the test accuracy is maximized. Notably, this does not always align with the lowest test loss, emphasizing the different aspects captured by the loss function and accuracy metric. The graphs advocate for the use of an early stopping mechanism to conserve resources and prevent overfitting. This strategy involves halting training when improvements in test accuracy diminish.

Higher model capacities seem more prone to overfitting. Implementing dropout, regularization, or opting for simpler models could improve generalization and computational efficiency. Modifying the loss function to better align with accuracy objectives or incorporating regularization directly could influence training dynamics favorably. To enhance generalization, data augmentation and ensemble methods could be beneficial. These techniques help in stabilizing performance and improving model robustness. Experimenting with dynamic learning rates or different batch sizes might stabilize and improve test accuracy, especially in the later stages of training.

For A10 to A13, the graphs demonstrate a rapid initial decrease in loss, which suggests an effective learning rate at the start of training. This phase is critical as it shows the model's capacity to quickly adapt to the data. Training accuracy tends to stabilize or even improve slightly as training progresses, while test accuracy often plateaus, indicating potential overfitting scenarios where the model excels in memorizing the training data rather than generalizing to new data. The test metrics, particularly loss, show significant fluctuations, indicating potential model sensitivity to specific batches of data or the impact of the learning rate variations over epochs. Adaptive learning rate strategies could help maintain or even improve test performance throughout training by adjusting the rate based on the changes in test accuracy. To combat overfitting, implementing regularization strategies like dropout or L2 regularization, or introducing synthetic noise into the training data might help improve generalization capabilities. Evaluating the complexity of the model might reveal if the model is too intricate for the task at hand. Simplifying the model could lead to better generalization without sacrificing early learning achievements. Data augmentation can increase the diversity of the training data, potentially enhancing the model's ability to generalize. Techniques such as random cropping, rotations, and color adjustments could be beneficial. Developing an ensemble of models or employing a systematic benchmarking framework could stabilize model performance and mitigate individual weaknesses, providing a more robust overall predictive capability.

For MNIST dataset, each model shows a rapid initial decrease in loss with a steep ascent in accuracy. This indicates effective learning mechanisms and appropriate model initialization. The rapid attainment of a high accuracy suggests that the models are well-suited to the dataset, benefiting from optimal architecture choices and training paradigms. Post-convergence, the models maintain high accuracy with minimal overfitting, as evidenced by the close tracking of training and testing accuracy. This performance stability is indicative of good generalization abilities, further validated by consistently low test loss values. The 'Best Epoch' marker in the graphs indicates the point at which the test accuracy is maximized. Interestingly, in all cases, this marker aligns closely with the epoch at which test loss is minimized, suggesting that the models not only achieve high accuracy but also do so with efficient learning without overfitting. The high accuracy achieved early in training and sustained throughout suggests that the models are potentially under-utilizing their capacity. Exploring more challenging variations of the task or applying these models to more complex datasets could be ways to fully leverage their capabilities. The demonstrated ability of these models to generalize well on test data makes them excellent candidates for real-world applications where robustness to new, unseen data is critical. Future investigations could focus on incremental learning scenarios where these models are periodically updated with new data, simulating real-world learning and adaptation.

Lastly, in Figures A1 to A4, it is clear to see that in Global attention mechanisms, the loss decreases whereas accuracy increases significantly only with very less epochs. The validation accuracy, particularly in MNIST datasets, often reaches and maintains a high level, close to 1. This suggests that the model is effectively generalizing from the training data to unseen data, a sign of good model performance without overfitting. The different activation functions and their configurations (like varying units) don't show significant variation in these graphs. It's worth noting, however, that in practical scenarios, these could impact the rate of convergence and final accuracy, especially under more complex or varied dataset conditions. Global Models tend to show faster initial convergence, which is indicative of a broader comprehension of the dataset from the outset. This could be due to the global model's ability to access all features of the input data simultaneously, allowing for rapid pattern recognition and learning.

Local Models: The local models often demonstrate a slower convergence. This is likely due to the localized view of the data, where the model learns more gradually as it processes smaller segments or features of the input data over time. After the initial rapid learning phase, global models tend to stabilize, but they might show signs of overfitting as seen with fluctuating test accuracies or higher losses as training progresses. This effect might be pronounced in complex datasets like CIFAR, where the global perspective could lead to capturing too many details, some of which do not generalize well. Local Models generally shows a more stable learning curve after the initial phase. The test accuracy and loss tend to plateau, which suggests that while learning is more gradual, it is potentially more stable and consistent. The local model's focus on subsets of data at a time could lead to better generalization on unseen data by not overemphasizing less relevant features. Global Models typically achieve higher maximum accuracies faster than local models. This is evident from the graphs where global models reach peak test accuracy quicker. This can be advantageous in applications where speed of training and high performance are critical, and sufficient regularization is employed to control overfitting. While they might not reach as high a maximum accuracy as global models or take longer to do so, local ones often maintain closer performance between training and testing phases, which can be indicative of better generalization.

Classifiers that leverage local features typically do not perform as well as those that utilize global features. This is likely due to the local classifiers being built upon a less complex Transformer structure. We find that classification performance using global features remains relatively consistent and generally surpasses that of the local features alone. Across both datasets, the efficacy diminishes when combined with local features from higher layers, despite the improved performance of such features.

7.3 Special Implementation of CIFAR-100 on Local Attention Transfomer

- **Peak Accuracy:** The highest test set accuracy is roughly 44.16%.
- **Loss Trends:** The initial decrease in both training and test loss is steep, with the test loss stabilizing at a relatively high level compared to the training loss. This pattern may suggest some overfitting issues.
- **Accuracy Trends:** While training accuracy consistently improves, test accuracy does not show similar progress, indicating difficulties in generalizing the learned features to new data.

7.3.1 CIFAR-10 Performance Overview

- **Peak Accuracy:** Significantly higher than CIFAR-100, the best accuracy reaches approximately 73.44%.
- **Loss Trends:** Both loss curves initially fall quickly and then the test loss levels off close to the training loss, showcasing better model generalization.
- **Accuracy Trends:** Training and test accuracies converge as training progresses, suggesting an effective learning and generalization capability of the model.

7.3.2 Differences in Dataset Performance

- **Dataset Complexity:** CIFAR-100, with its 100 classes, presents a greater challenge than CIFAR-10, which has only 10 classes. This is evident from the lower test accuracies and higher losses observed.
- **Generalization:** The model trained on CIFAR-10 shows superior generalization compared to the CIFAR-100 model, as indicated by the closer alignment of training and test losses.
- **Achievable Accuracy:** There is a notable disparity in the maximum accuracies between the two datasets, likely due to the simpler structure and fewer classes in CIFAR-10, which facilitates learning.

8. Challenges

This project faced numerous challenges, ranging from intricate technical issues to unexpected logistical hurdles. Below we detail the primary obstacles and the strategies employed to navigate them.

8.1 Enhancing Local Attention in Swin Transformers

A pivotal aspect of refining the Swin Transformer model involved enhancing the local attention mechanism to more effectively process input data. This was achieved by incorporating convolutional operators within the architecture, which helped address key challenges related to data outliers and the inherent limitations of standard self-attention mechanisms.

8.1.1 Rationale for Convolutional Operators

Traditional self-attention layers compute the similarity between queries and keys based solely on their point-wise values, often disregarding the spatial or sequential locality of the data points. This approach can lead to misinterpretations, particularly when dealing with outliers or rare events in datasets such as time series. An illustrative example of this issue is provided in Figure 1, where the direct matching of key and query values might misrepresent the true nature of a data point, be it an outlier, a change point, or a regular point.

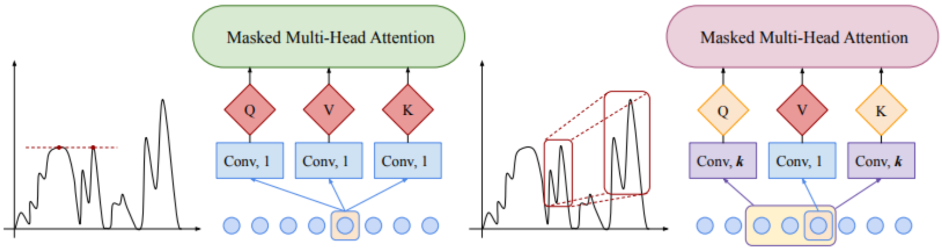

Figure 1: Illustration of Key-Query Matching Issues

These misinterpretations can lead to optimization challenges during model training, adversely affecting the overall model performance. These enhancements not only improved our model's accuracy but also its ability to generalize from training to unseen data, effectively addressing one of the core challenges in attention-based models.

8.2 Adjusting Layers and Input Size

Adjusting the network configuration to cater to different datasets, particularly CIFAR-10 and CIFAR-100, presented another layer of complexity. We had to experiment extensively with layer depths and input sizes to find the optimal settings that maximized performance without overburdening our computational resources. These configurations were crucial to ensure that the model was capable of effectively learning from more complex images.

8.3 Mitigating Overfitting

Overfitting was a persistent challenge, an example, which is particularly evident as well, is in our experiments with the CIFAR-100 dataset. To address this, we implemented several strategies, such as introducing dropout layers and employing data augmentation techniques. A key part of our approach included integrating dropout within the network to enhance the generalization of the model. This approach helped minimize overfitting by randomly omitting units during training, thereby promoting the development of more robust features.

8.4 External Constraints

Unexpectedly, our project was also affected by external factors. Limited access to stable internet and computational resources due to escalating conflicts at Columbia University. This situation forced us to rethink our computational strategy, shifting some of our workloads to cloud-based services and optimizing our local setups to continue our research under these constrained conditions. Yet, it still broke up our plan a week ago when the conflicts became worse so we will not have enough time to negotiate with the school to let us finish the rest of the execution and this referencing from ImageNet and Places365 dataset.

9. Conclusion

In this project, we have extensively explored the application of Transformer and Performer architectures for image classification tasks, utilizing a comprehensive evaluation framework across diverse datasets. Our findings underscore the robust potential of these architectures in harnessing the strengths of both local and global attention mechanisms to achieve impressive classification accuracy.

Transformers, with their global attention capabilities, have demonstrated exceptional performance in understanding the broader context within images, a crucial factor in tasks like scene recognition and complex image classification. The Performer models, particularly with their scalable and computationally efficient design, have proven their efficacy in handling large datasets without a compromise in performance, courtesy of the innovative random feature approximation and efficient attention mechanisms.

The integration of local attention mechanisms has been pivotal in capturing intricate spatial dependencies, which are often missed by global attention alone. This dual approach not only enhances the model's interpretive power but also its ability to generalize across various image classification scenarios. Moreover, the empirical insights gleaned from our project affirm the transformative impact of these architectures in the realm of computer vision. By bridging the gap between theoretical innovation and practical application, our work contributes to the ongoing dialogue in the machine learning community about the optimal strategies for deploying deep learning architectures in real-world scenarios.

In future work, we aim to further dissect the interaction between different layers of attention and their contributions to the model's overall efficacy. This will potentially pave the way for more fine-tuned and resource-efficient architectures that do not sacrifice performance for speed, thereby enabling their deployment in more computationally constrained environments. Our project not only validates the significant advancements made in the field of deep learning but also highlights the continuous need for innovation and adaptation in the face of evolving technological challenges and dataset complexities.

10. Author Contributions

- Joshwin Rajan: Performers' code on Softmax; Paper section (Introduction, Datasets and Conclusion)
- Pinqiao Wang: Performers' code on Relu; Paper section (Transformer Model, Implementation Details and Evaluation Metrics)
- Satyabrat Srikumar: Transformers code; Paper section (Performer Model and Attention Mechanism)

Reference

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), Advances in Neural Information Processing Systems (Vol. 30). Curran Associates, Inc.

2. Choromanski, K., Henaff, M., Mathieu, M., Arous, G. B., & Gelly, S. (2021). Revisiting attention models in vision at imagenet and beyond. *arXiv preprint arXiv:2105.13239*.

3. LeCun, Y., Cortes, C., & Burges, C. (2010). MNIST handwritten digit database. AT&T Labs [Online]. Available: http://yann.lecun.com/exdb/mnist.

4. Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

5. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). IEEE.

6. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017). Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence, 40(6), 1452-1464.

7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

A. Appendices

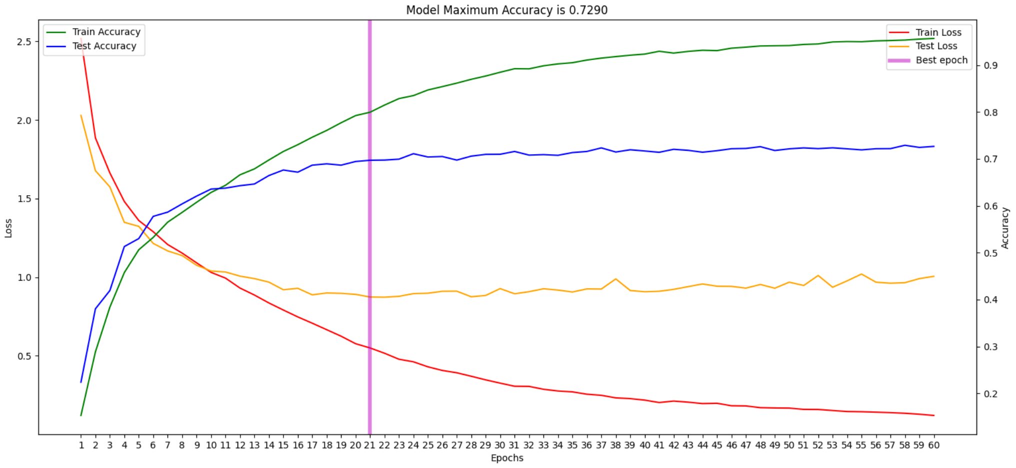

Figure A1: Global Transformer Loss/Accuracy Line Graph with CIFAR-10

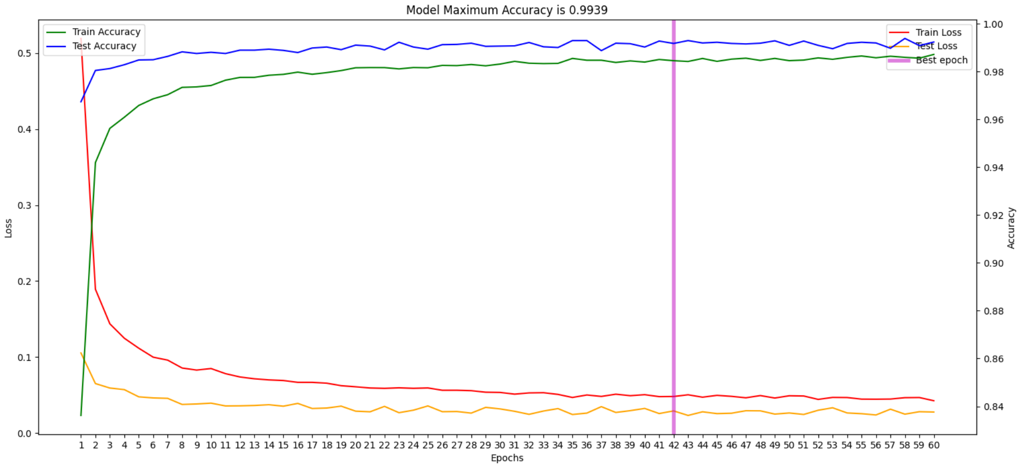

Figure A2: Global Transformer Loss/Accuracy Line Graph with MNIST

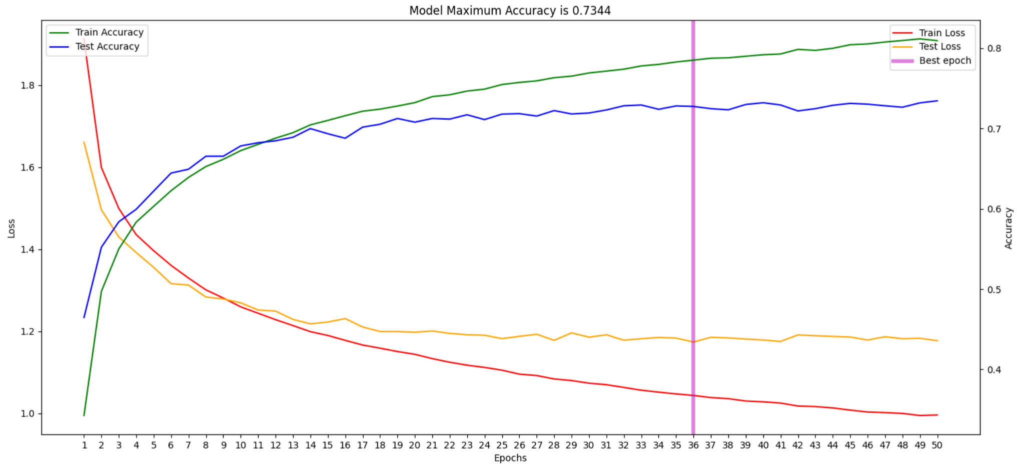

Figure A3: Local Transformer Loss/Accuracy Line Graph with CIFAR-10

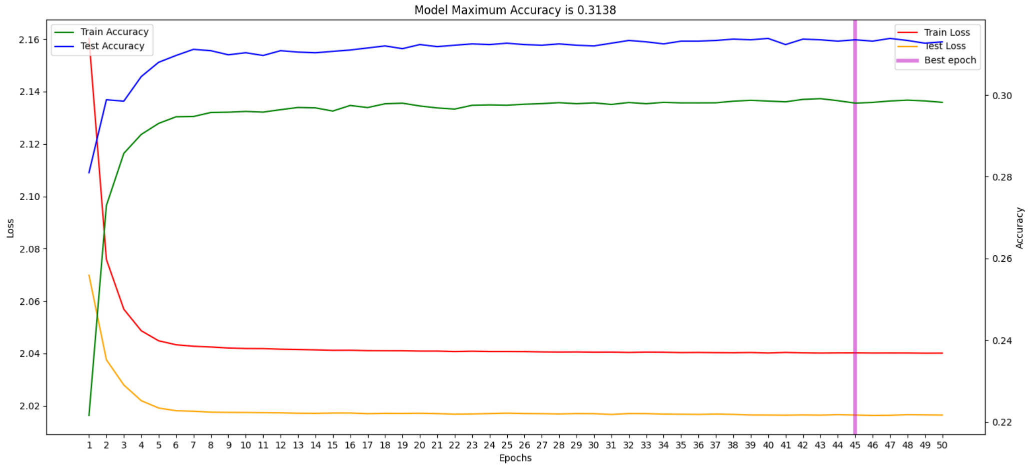

Figure A4: Local Transformer Loss/Accuracy Line Graph with MNIST

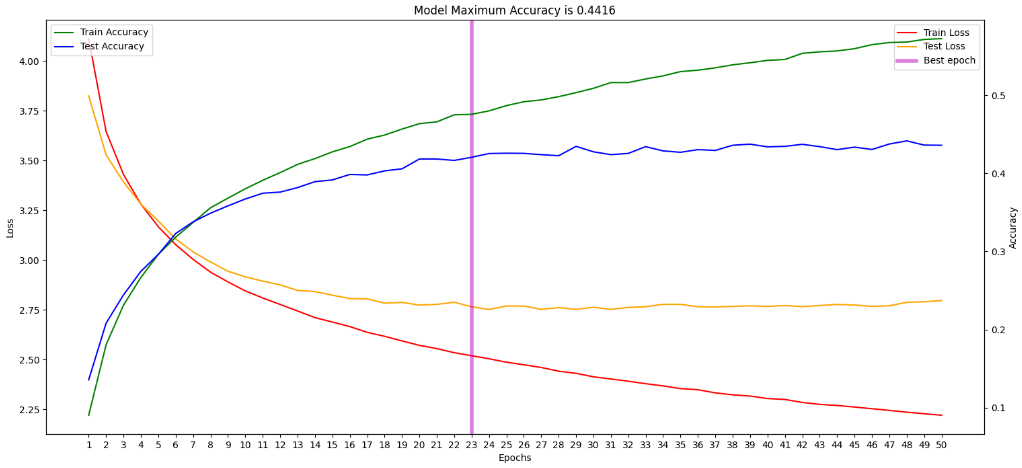

Figure A5: Transformer Loss/Accuracy Line Graph for CIFAR-100 with Local attention

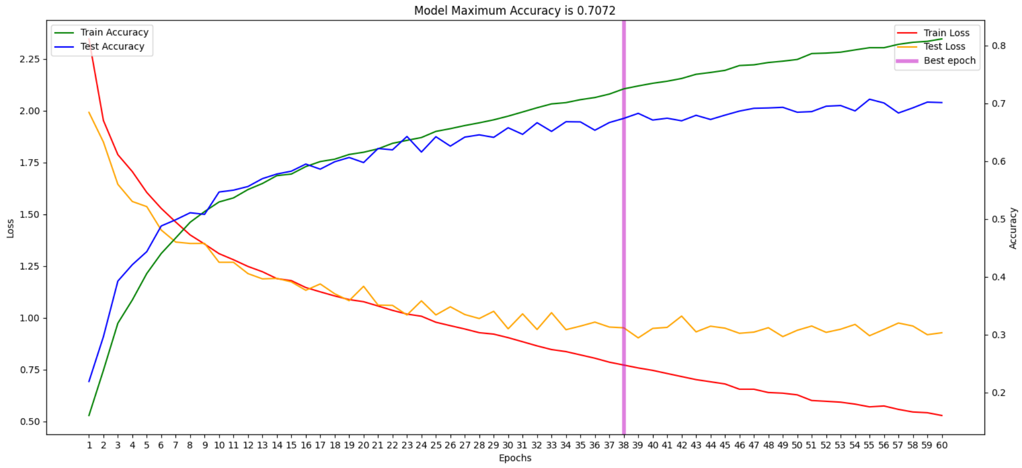

Figure A6: Performer Loss/Accuracy Line Graph for CIFAR-10 with Relu (m = 16)

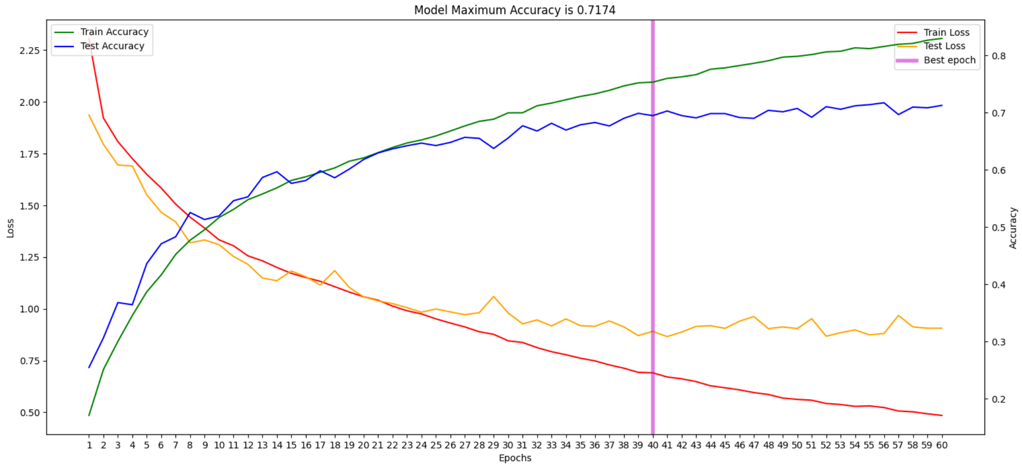

Figure A7: Performer Loss/Accuracy Line Graph for CIFAR-10 with Relu (m = 32)

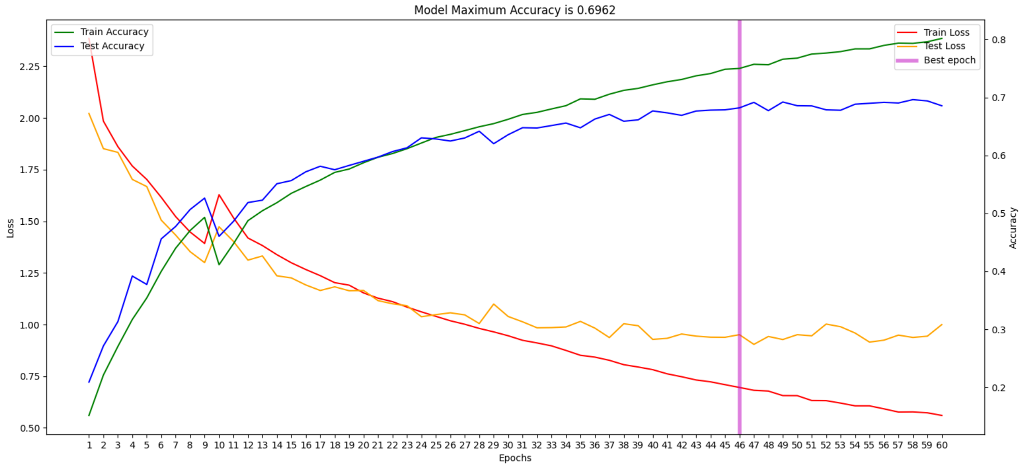

Figure A8: Performer Loss/Accuracy Line Graph for CIFAR-10 with Relu (m = 64)

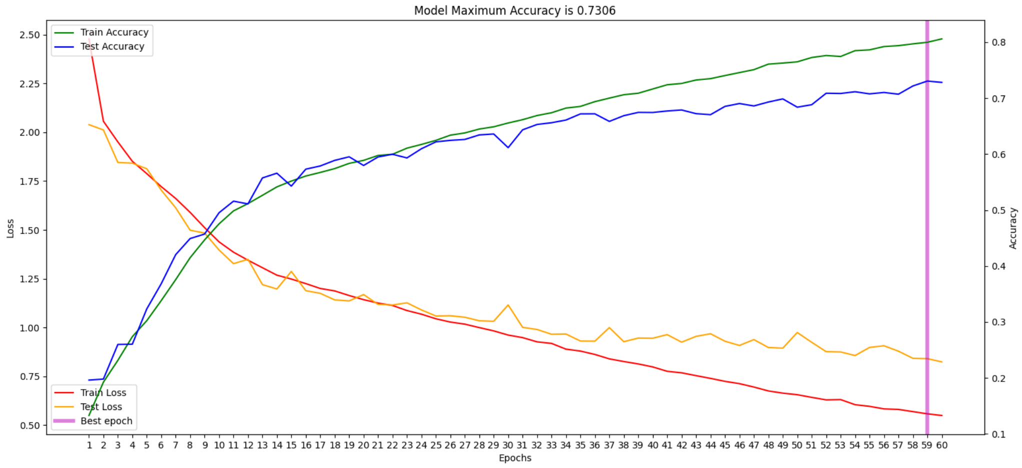

Figure A9: Performer Loss/Accuracy Line Graph for CIFAR-10 with Relu (m = 128)

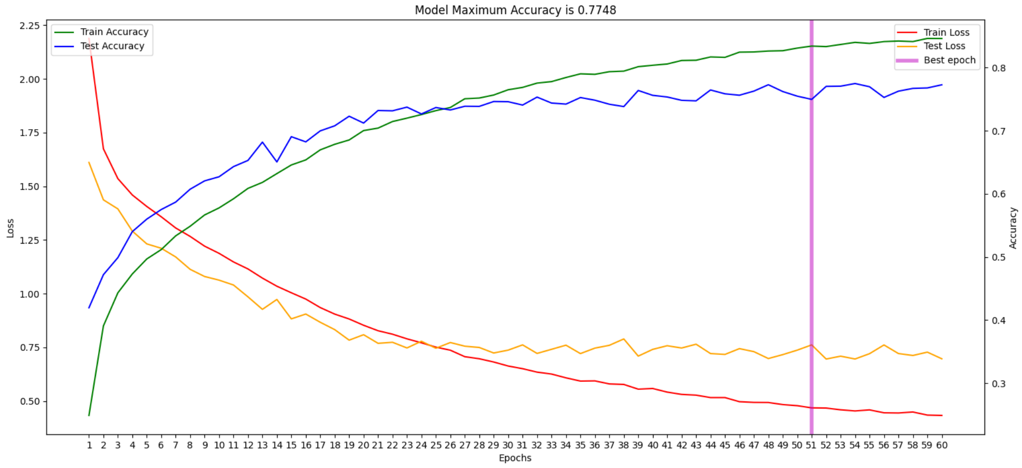

Figure A10: Performer Loss/Accuracy Line Graph for CIFAR-10 with Softmax (m = 16)

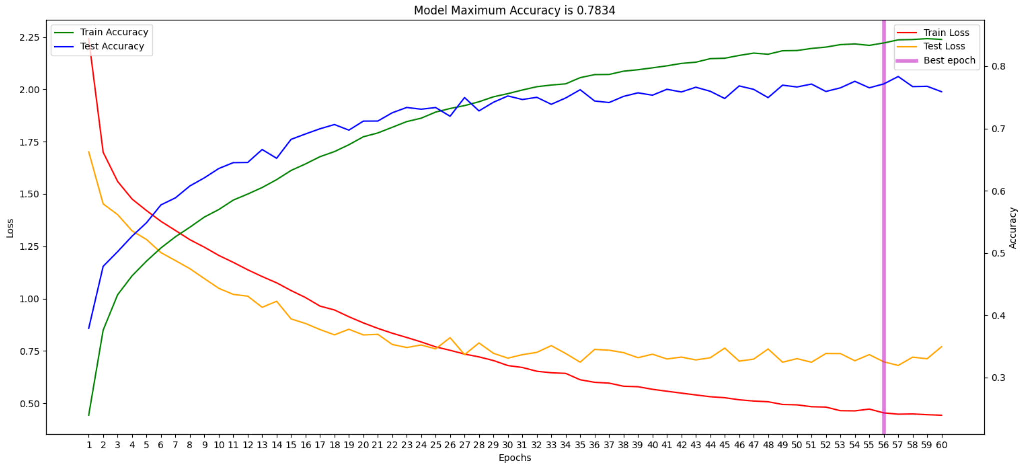

Figure A11: Performer Loss/Accuracy Line Graph for CIFAR-10 with Softmax (m = 32)

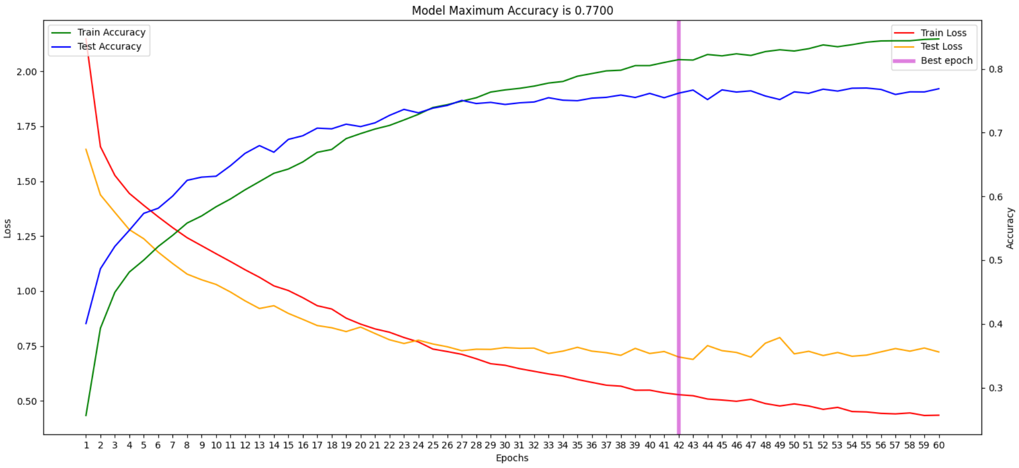

Figure A12: Performer Loss/Accuracy Line Graph for CIFAR-10 with Softmax (m = 64)

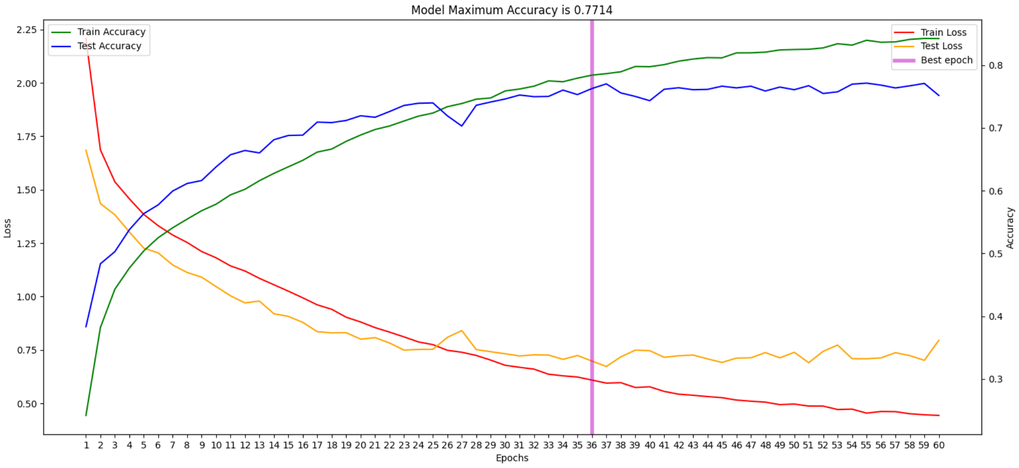

Figure A13: Performer Loss/Accuracy Line Graph for CIFAR-10 with Softmax (m = 128)

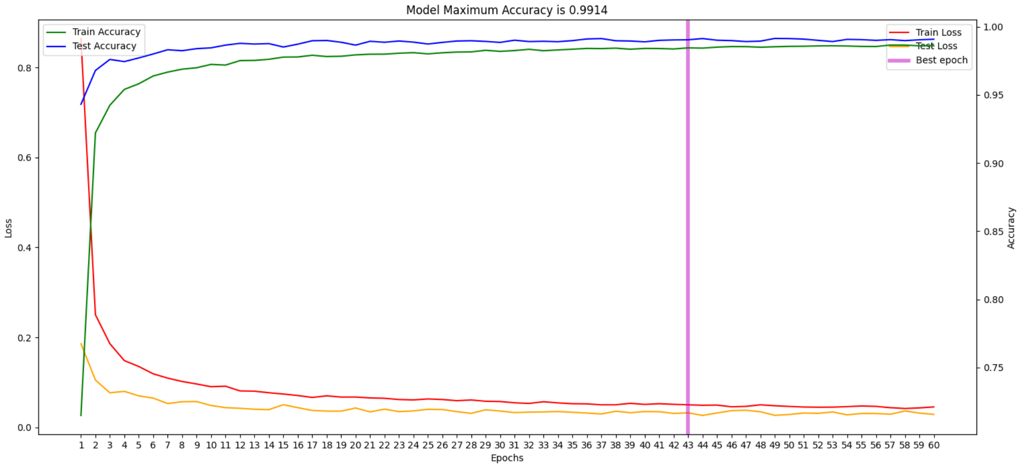

Figure A14: Performer Loss/Accuracy Line Graph for MNIST with Relu (m = 16)

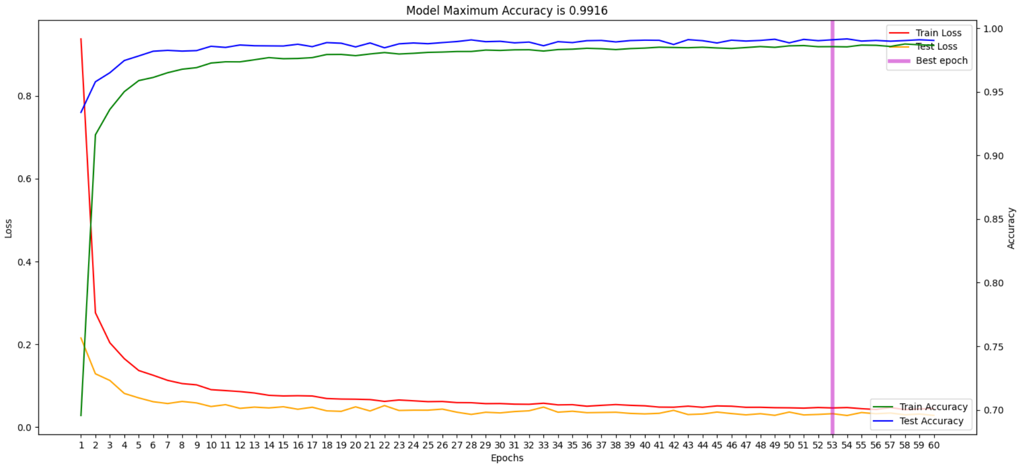

Figure A15: Performer Loss/Accuracy Line Graph for MNIST with Relu (m = 32)

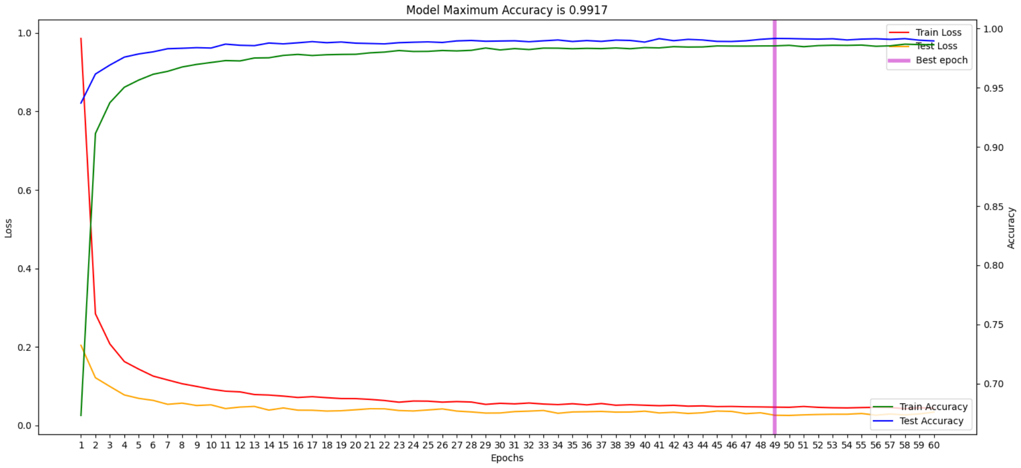

Figure A16: Performer Loss/Accuracy Line Graph for MNIST with Relu (m = 64)

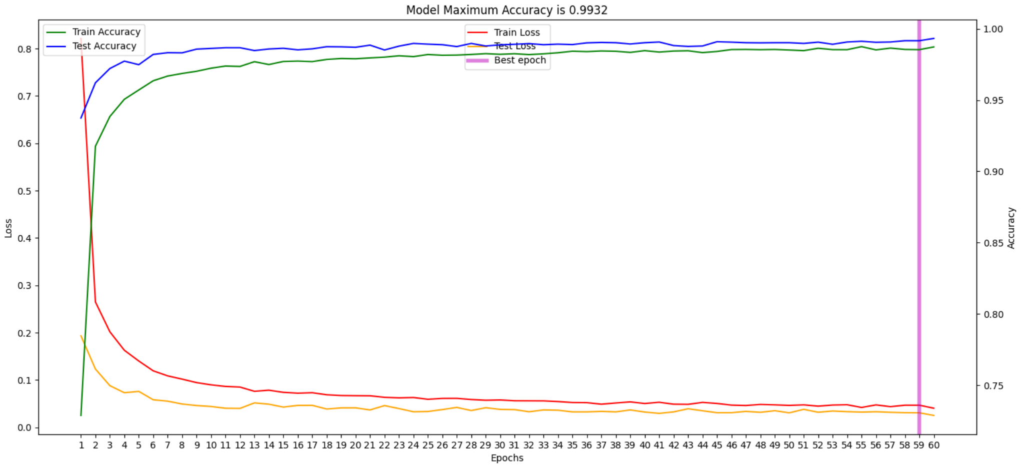

Figure A17: Performer Loss/Accuracy Line Graph for MNIST with Relu (m = 128)

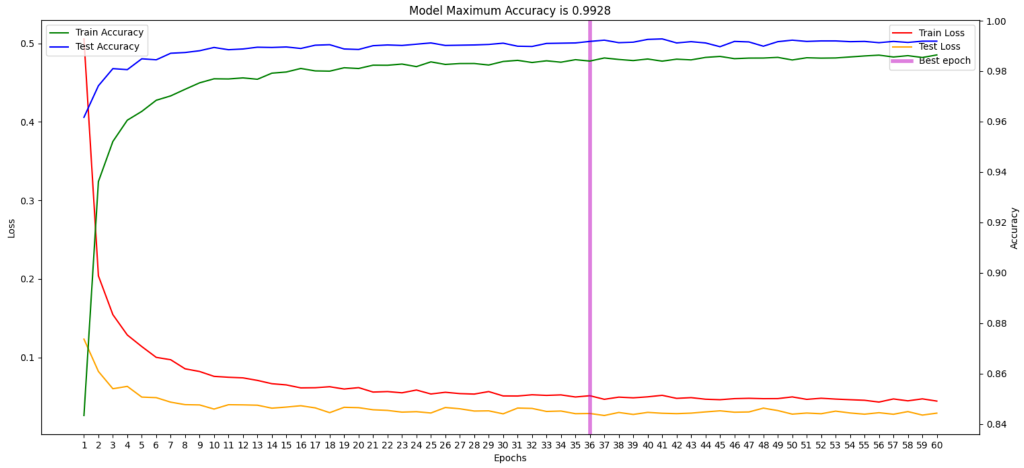

Figure A18: Performer Loss/Accuracy Line Graph for MNIST with Softmax (m = 16)

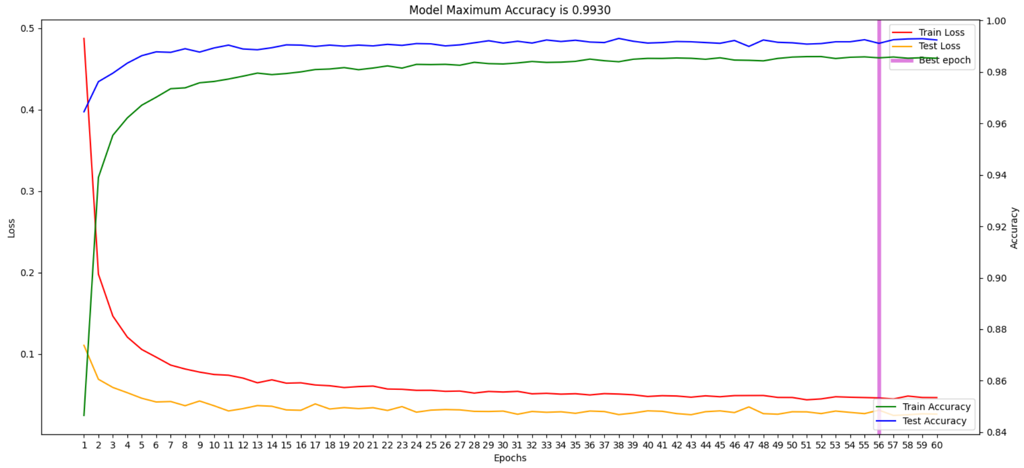

Figure A19: Performer Loss/Accuracy Line Graph for MNIST with Softmax (m = 32)

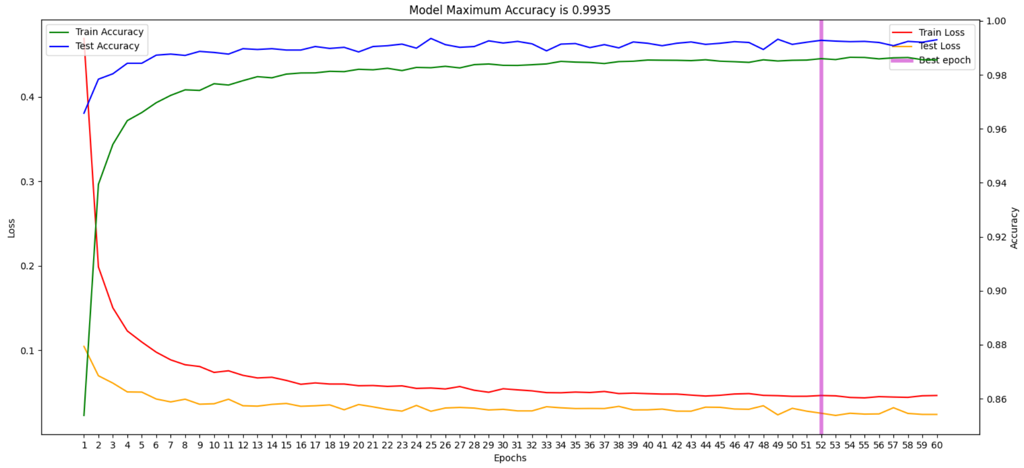

Figure A20: Performer Loss/Accuracy Line Graph for MNIST with Softmax (m = 64)

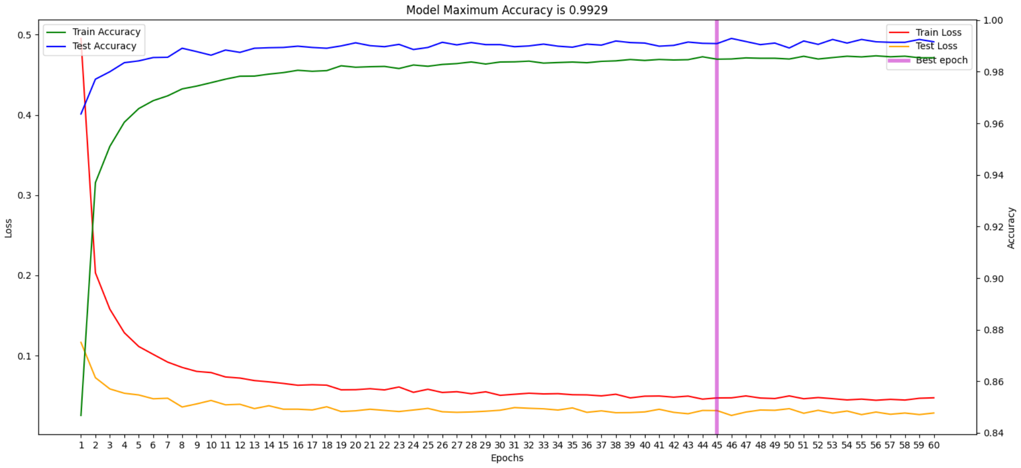

Figure A21: Performer Loss/Accuracy Line Graph for MNIST with Softmax (m = 128)