Modified Residual Network (ResNet) on CIFAR-10 Image Classification Dataset

Phakphum Artkaew*and Vorrapard Kumthongdee*

Department of Electrical and Computer Engineering, New York University
pa2497@nyu.edu, vk2584@nyu.edu

Abstract

Convolutional Neural Networks (CNNs) have revolutionized image recognition, with Residual Networks (ResNets) emerging as a powerful architecture capable of training extremely deep models. However, the quest to enhance model efficiency and accuracy within constrained computational resources remains an active challenge. This study introduces a Modified Residual Network tailored for the CIFAR-10 image classification dataset, exploring innovative adjustments to the traditional ResNet model. By experimenting with layer configurations, data augmentation strategies, and optimization methods, the research aims to uncover configurations that maximize performance while adhering to a stringent five million parameter limit. The proposed architecture undergoes systematic iterations, fine-tuning channel sizes, modifying training protocols, and leveraging advanced techniques like CosineAnnealingLR scheduling. The final model, with channel sizes [100, 150, 215, 256] trained using SGD and CosineAnnealingLR over 200 epochs, achieves an impressive 95% accuracy on CIFAR-10 while satisfying the parameter budget. This study contributes to the broader discourse on neural network optimization, offering insights into balancing model complexity, computational efficiency, and performance for image classification tasks. The findings pave the way for developing more efficient and effective deep learning models applicable to real-world scenarios with limited computational resources.

Introduction

In the realm of deep learning, Convolutional Neural Networks (CNNs) have emerged as a cornerstone for solving complex image recognition tasks, revolutionizing the way machines understand visual information [1]. Among the various architectures proposed to enhance the performance of CNNs, Residual Networks (ResNets) have been particularly notable for their ability to train very deep networks through the introduction of skip connections [3]. These connections allow the network to bypass one or more layers, addressing the vanishing gradient problem and facilitating the training of networks that are substantially deeper than was previously possible [7].

The CIFAR-10 dataset [6], consisting of 60,000 32x32 color images in 10 distinct classes, presents a benchmark challenge for image classification models. Its diverse and balanced dataset provides an ideal platform for evaluating the effectiveness of neural network architectures and modifications. Despite the success of traditional ResNets in achieving high accuracy on CIFAR-10, there remains a continuous quest for optimization, aiming to enhance model efficiency and accuracy within the constraints of computational resources and model complexity [4].

This study introduces a Modified Residual Network architecture, exploring innovative adjustments to the traditional ResNet model to address these challenges. By experimenting with alterations in layer configurations, data augmentation, and optimization method, this research aims to uncover configurations that maximize performance specifically for the CIFAR-10 dataset. The modifications are designed not only to improve classification accuracy but also to reduce the model's computational demands, making high-performance deep learning models more accessible and practical for real-world applications. In conducting this experiment, we confront the dual challenge of maintaining or enhancing model accuracy while adhering to five millions parameter limit. This research contributes to the broader discourse on neural network optimization, offering insights into the balance between model complexity, computational efficiency, and performance in the context of image classification tasks. Through a rigorous evaluation of the Modified Residual Network on CIFAR-10, this study seeks to provide valuable guidance for the development of more efficient and effective deep learning models. The code and report for this study can be accessed at GitHub repository https://github.com/PhakphumAdev/ECE-GY-7123-mini-project.

Related Work

Residule Network

The landscape of deep learning has been dramatically reshaped with the advent of deep convolutional neural networks (CNNs). However, as networks became increasingly deep, the community faced the vanishing gradients problem, which hinders the training of very deep architectures. The introduction of Residual Networks (ResNets) by He et al. in 2015 [3] addressed this issue by employing shortcut connections that allow gradients to flow through the network, thereby enabling the training of networks that are signifi-

*These authors contributed equally.

cantly deeper than previous architectures. The shortcut connections are also called skip connections, which create a sequence of the convolutional layer in between called residual block. Then, a sequence of blocks creates a residual layer as Figure 1. The number of residual block in each residual layer is various depending on the designed architecture.

ResNets have set new records in various benchmark datasets, including the CIFAR-10 dataset, due to their ability to learn representational features at multiple levels of abstraction. This breakthrough was foundational to ResNets' subsequent adaptations and improvements. Szegedy et al. [8] incorporated Inception modules within ResNet to form Inception-ResNet, combining the strengths of Inception networks with the residual connections, leading to improvements in training speed and accuracy.

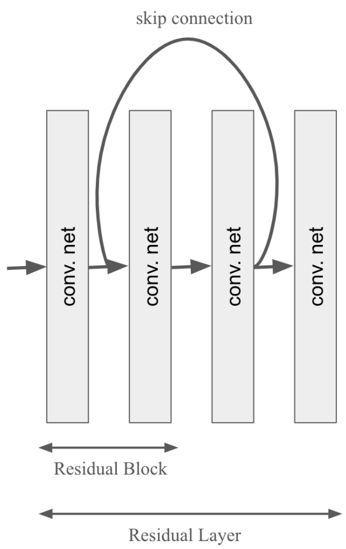

Figure 1: Residual Network

ResNets have also been adapted for tasks beyond image classification. In the domain of object detection, for instance, Feature Pyramid Networks (FPNs) have employed ResNet backbones to create feature pyramids that enhance the network's representational power across different scales, which is critical for detecting objects of various sizes.

Lastly, the advent of neural architecture search (NAS) has led to automated methods for designing network architectures. ResNet-like architectures have been a common starting point for these algorithms, which seek to optimize various aspects of network design, including the configuration of residual blocks.

This body of work establishes the significance of ResNets in the evolution of deep neural networks. The current study builds upon these foundational models, proposing modifications aimed at further improving computational efficiency and classification accuracy within the constraints of parameter efficiency, specifically targeting the CIFAR-10 image classification dataset.

Hyperparameters

Hyperparameter optimization in deep learning is a critical factor in determining the efficacy of a neural network. The field has taken a keen interest in the study of hyperparameters in Residual Networks (ResNets), as these parameters significantly influence the training dynamics and final performance of the model. This section outlines previous work focusing on various hyperparameters within the ResNet architecture and their impact on model training and generalization.

Initial strides in understanding ResNet hyperparameters were taken by He et al. [4], who provided the first insights into the depth and width of residual networks. Their work established a baseline for the number of layers and filters that could be effectively trained, revealing that deeper and wider networks tend to exhibit superior performance, albeit with diminishing returns beyond a certain point.

Building upon these findings, Zagoruyko and Komodakis [9] introduced Wide Residual Networks (WRNs), emphasizing the trade-off between depth and width. They argued for increasing the width—i.e., the number of channels—of residual layers as a more effective way to improve performance than increasing depth, particularly when computational resources are a limiting factor.

Another significant area of study involves the optimization algorithms used for training ResNets. The works by Choi et al. [2] have provided a comparative analysis of Stochastic Gradient Descent (SGD) with momentum versus adaptive methods such as Adam, with SGD with momentum often emerging as the superior choice for generalization in image classification tasks.

The role of batch size in training ResNets is another focus area, as it affects gradient estimation and memory utilization. Keskar et al. [5] examined the implications of large batch sizes on the quality of the model, indicating that while larger batches provide computational efficiency, they can also negatively impact the model's ability to generalize from the training data.

This study contributes to the studies by investigating a combination of these hyperparameters within a Modified Residual Network trained on the CIFAR-10 dataset. Specifically, we examine the channels in the residual layer, data augmentation, and the model accuracy between the SGD and Adam optimization method.

Methodology

Dataset and Preprocessing

In this study, the CIFAR-10 dataset was utilized, comprising 60,000 images of 32x32 color resolution across 10 classes. The dataset was divided into a training set comprising 50,000 images and a test set containing 10,000 images. To enhance model generalization, data augmentation techniques such as random crop, random horizontal flip, random rotation, and color jitter were applied. These transformations introduced variability into the training data, simulating real-world scenarios and helping prevent overfitting.The architecture of the Modified Residual Network was based on a standard Residual Network with key alterations tailored to optimize performance under a stringent parameter budget of five million parameters. The modification of this study mainly focuses on altering the number of channels in residual layer.

Model and Training Procedure

This study is an implementation on the original Github respository [1] of the ResNet18 with an accuracy of 93.02 %. The initial architecture, based on a standard Residual Network configuration consist of 4 Residual layer with 2 Residual block each. The stride is set to be 1,2,2, and 2 respectively in each block. The initial channel configurations of [64, 128, 256, 512] was modified in multiple iterations to optimize parameter efficiency in this study. The first modification reduced the channel configuration to [64, 128, 160, 256], successfully decreasing the total parameters from 11 million to 3,757,194.

The models were trained using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.1, momentum of 0.9, and weight decay of 5e-4. The learning rate was adjusted using a stepLR scheduler with a step size of 10 and gamma of 0.1. Initially, models were trained for 25 epochs, with subsequent iterations extending this to 50 and then 200 epochs for more profound training as modifications increased model complexity. In the fourth iteration, the architecture was adjusted to [100, 150, 215, 256] with a slight increase in parameters to 4,993,583. A new training strategy involving the SGD optimizer paired with CosineAnnealingLR was adopted and extended to 200 epochs, significantly boosting model accuracy to 95%.

Optimization

In subsequent trials, further adjustments to the channel configurations were tested, each aiming to balance the model's performance with computational efficiency. Additional training strategies were introduced, such as removing color jitter and random rotation from the data augmentation routine to focus on more impactful transformations. In addition, all experiments were conducted using a GPU-accelerated environment to facilitate rapid model training and evaluation. This setup was critical for handling the extensive computational demands of training deep neural networks.

Result

This study explored several iterations of model configurations with the objective of enhancing the CIFAR-10 classification accuracy while reducing the overall parameter count of a Modified Residual Network. Each trial aimed to find an optimal balance between model complexity and computational efficiency.

First Iteration: Initially, the channels in residual layer were adjusted to [64, 128, 160, 256] from the original [64, 128, 256, 512], which successfully reduced the total number of parameters from 11 million to 3,757,194. This configuration was trained using a variety of data augmentation techniques, and despite the significant reduction in parameters, the model maintained a robust performance. Training was implemented over 25 epochs, which provided a foundational understanding of the impacts of reduced complexity. The learning graph is plotted in Figure 2. The accuracy of the model rapidly increases in the first 10 epoch before turning into steady afterwards.

Second Iteration: Observing the capacity for further tuning, the channel sizes were modified to [64, 128, 222, 256], increasing the parameter count slightly to 4,635,362. This iteration excluded color jitter from the augmentation strategy to focus on the most impactful transformations. Extending the training to 50 epochs allowed for a deeper maturation of model weights, showing a slight improvement in accuracy.

Third Iteration: Further refinements led to a channel configuration of [64, 128, 200, 256], resulting in a parameter count of 4,299,994. Random rotation was removed from the data augmentation suite, refining the strategy to leverage the most effective transformations. Again, this model was trained over 50 epochs.

Fourth Iteration: The most substantial modifications were made in this trial, with channels set to [100, 150, 215, 256] and parameters adjusted to 4,993,583. An advanced training regimen using SGD coupled with CosineAnnealingLR was employed over 200 epochs, significantly enhancing the model's accuracy to 95

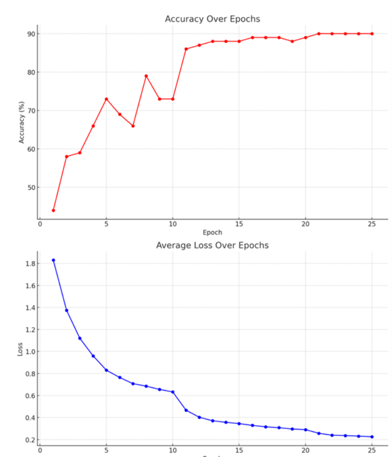

Figure 2: Training Graph of First Iteration

The graphs from both results shows the differences between the slope of the accuracy. The first iteration reaches the steady state after 10 epoch of training, while the fourth iteration reaches the steady state after 20 epoch of training. However, the fourth achieves a higher final accuracy after 25 epoch of training. From four iterations, the model with the highest accuracy is found with the channels of [100, 150, 215, 256] training with SGD optimizer and CosineAnnealingLR.

Conclusion

This study successfully demonstrated the potential of optimizing Residual Network architectures to achieve high ac-

[1]https://github.com/kuangliu/pytorch-cifar. Accessed on April 5, 2024

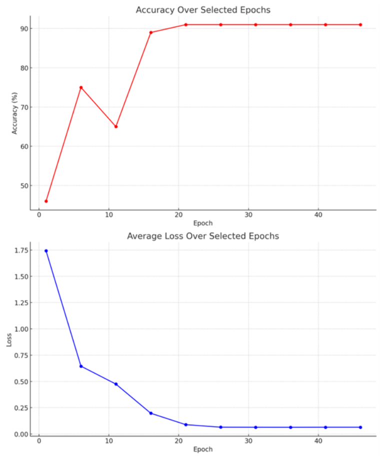

Figure 3: Training Graph of Fourth Iteration

curacy in CIFAR-10 image classification while adhering to strict parameter constraints. Through a systematic series of modifications and enhancements to the network structure and training protocols, we were able to reduce the complexity of the model significantly—from an initial 11 million parameters to an effective configuration under 5 million parameters—without compromising, and in fact, improving, the performance.

The final model configuration, utilizing channel sizes [100, 150, 215, 256] and an advanced training strategy incorporating SGD with CosineAnnealingLR, achieved an impressive 95% accuracy over 200 training epochs. This represents a significant improvement over the original architecture and highlights the efficacy of targeted architectural and procedural refinements. These findings underline the importance of thoughtful design in the development of deep learning models, particularly in contexts where computational resources are limited. By optimizing both the architecture and the training process, it is possible to achieve state-of-the-art results with reduced computational demands.

Future work could explore further enhancements to the data augmentation strategies and regularization techniques to refine the model's performance and efficiency. Additionally, applying the insights gained from this CIFAR-10 study to other datasets and classification tasks could provide broader validation of the model's adaptability and effectiveness.

In conclusion, the modifications explored in this study not only enhance the performance of Residual Networks on a standard image classification benchmark but also contribute to the broader field of neural network optimization, offering a promising avenue for developing more efficient and powerful deep learning models in various real-world applications.

References

[1] Chauhan, R.; Ghanshala, K. K.; and Joshi, R. 2018. Convolutional Neural Network (CNN) for Image Detection and Recognition. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 278–282.

[2] Choi, D.; Shallue, C. J.; Nado, Z.; Lee, J.; Maddison, C. J.; and Dahl, G. E. 2020. On Empirical Comparisons of Optimizers for Deep Learning. arXiv:1910.05446.

[3] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

[4] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Identity Mappings in Deep Residual Networks. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision – ECCV 2016*, 630–645. Cham: Springer International Publishing. ISBN 978-3-319-46493-0.

[5] Keskar, N. S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; and Tang, P. T. P. 2017. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. arXiv:1609.04836.

[6] Krizhevsky, A.; Nair, V.; and Hinton, G. 2024. CIFAR-10 (Canadian Institute for Advanced Research).

[7] Patnaik, S.; Kumari, S.; and Das Mahapatra, S. 2020. Comparison of deep CNN and ResNet for Handwritten Devanagari Character Recognition. In *2020 IEEE 1st International Conference for Convergence in Engineering (ICCE)*, 235–238.

[8] Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. arXiv:1602.07261.

[9] Zagoruyko, S.; and Komodakis, N. 2017. Wide Residual Networks. arXiv:1605.07146.