Optimizing Deep Learning Models for Image Classification on CIFAR-10

ECE-GY 7123 Deep Learning
Shanmuk Reddy, Nathaniel Sehati

[1]New York University Tandon School of Engineering
6 MetroTech Center
Brooklyn, New York 11201 USA

Abstract

This report outlines the process and results of optimizing a ResNet model for the CIFAR-10 image classification challenge. Our approach focuses on modifying the traditional ResNet architecture to enhance accuracy while efficiently managing model complexity and computational demands. Through comprehensive data augmentation, hyperparameter tuning, and the utilization of advanced optimization strategies, we achieve significant improvements over the baseline ResNet performance. This work contributes to the research on efficient deep learning methodologies for image classification, offering a practical framework for similar applications.

Code Availability: https://github.com/sk11331/Image-classification-with-Modified-ResNet-CIFAR10.

Introduction

Image classification stands as a cornerstone task in computer vision, with the CIFAR-10 dataset serving as a pivotal benchmark for model evaluation. Recent advancements have underscored the effectiveness of Deep Residual Networks (ResNets) in setting new standards for accuracy. Yet, the pursuit of higher precision often escalates model complexity, prompting concerns regarding overfitting, computational demand, and inference latency. This project seeks to address these challenges by optimizing a ResNet model for CIFAR-10, striving for an optimal balance between accuracy and efficiency.

Methodology

We set out to customize the architecture of a traditional ResNet model to address the specific challenges posed by the CIFAR-10 dataset. Our model, termed ModifiedResNet, incorporates lessons learned from the limitations of standard approaches, resulting in a tailored architecture that prioritizes efficient learning and generalization. We leveraged PyTorch's robust framework and GPU acceleration to experiment with various configurations.

As part of our research, we incorporated state-of-the-art methods and well-established best practices in the field of deep learning for image classification. One particular resource that proved invaluable was a GitHub repository containing PyTorch code for training various ResNet models on the CIFAR-10 dataset (Liu 2021). This repository, authored by Kuang Liu, provides a comprehensive suite of tools and pre-trained models that significantly assisted us in the the development of our project. The implementation of data augmentation techniques, optimization methods, and network configurations were influenced by the code available in this public repository. In our study, we also considered the implementation guidelines for ResNet architectures as discussed by Shuvam Das (Das 2023), alongside the foundational work by Kaiming He et al. (He et al. 2015) on deep residual learning.

Model Architecture and Complexity

Our ModifiedResNet architecture is designed to tackle the CIFAR-10 dataset with both efficiency and accuracy. The architecture builds upon the basic principles of ResNet, introducing modifications that optimize performance for the specific characteristics of CIFAR-10.

Detailed Architecture

The ModifiedResNet model begins with a convolutional layer (‘Conv2d‘), taking an input with 3 channels and producing a 64-channel feature map with a kernel size of 3x3 and stride of 1, maintaining the spatial dimensions through padding. This is followed by a batch normalization layer (‘BatchNorm2d‘) to accelerate training and stabilize the learning process by normalizing the input layer by adjusting and scaling the activations.

The core of the model consists of three main blocks (‘layer1‘, ‘layer2‘, ‘layer3‘), each comprising a sequence of two BasicBlocks. These blocks are designed to process feature maps at different scales, gradually reducing the spatial dimensions while increasing the depth, allowing the network to learn more complex features at different levels of abstraction.

Each BasicBlock consists of two convolutional layers followed by batch normalization, with a shortcut connection enabling the flow of gradients directly through the network, mitigating the vanishing gradient problem. The transition between different scales is managed by adjusting the stride

and using convolutional layers in the shortcut connections to match the dimensions.

- **Layer1** operates on 64-channel inputs, maintaining the dimensionality while enhancing feature representations.
- **Layer2** transitions to 128-channel feature maps, reducing the spatial size by half using a stride of 2 in the first convolutional layer of the block.
- **Layer3** further reduces the spatial dimension while moving to 256-channel feature maps, capturing high-level semantic features essential for classification.

The network concludes with a linear layer ('Linear'), which maps the high-level feature representations to the 10 classes of CIFAR-10, producing the final classification.

Computational Complexity

A detailed analysis of the model's computational complexity reveals the following key statistics:

- **Total Parameters:** 2,777,674 trainable parameters, indicating the model's capacity to learn from the data.
- **Input Size:** 0.01 MB per image, demonstrating the model's ability to operate efficiently on small images typical of CIFAR-10.
- **Forward/Backward Pass Size:** 10.50 MB, reflecting the memory requirements during training.
- **Estimated Total Size:** 21.11 MB, encompassing the model parameters, input, and activations, highlighting the model's balance between complexity and efficiency.

These metrics underscore our model's design philosophy, aiming to achieve high accuracy on CIFAR-10 without excessive computational demands, making it suitable for environments with limited resources.

Model Summary and Parameters

The following figures provide an in-depth view of our ModifiedResNet model's architecture and parameter summary as generated by our training script. Figure 1 illustrates the architecture of the model, showcasing the sequence of layers and operations that constitute the ModifiedResNet. Figure 2 provides a summary of the model's parameters, detailing the shapes of the output features and the number of parameters at each layer.

Experimental Setup

- **Dataset:** CIFAR-10, split into 50,000 training and 10,000 test images.
- **Evaluation Metrics:** Model performance was evaluated based on classification accuracy on the test set.
- **Hardware and Software:** Experiments were conducted using PyTorch on NVIDIA GPUs.

Architectural Choices

We focused on a modified ResNet architecture with the following adaptations:

Figure 1: ModifiedResNet architecture visualized layer by layer.

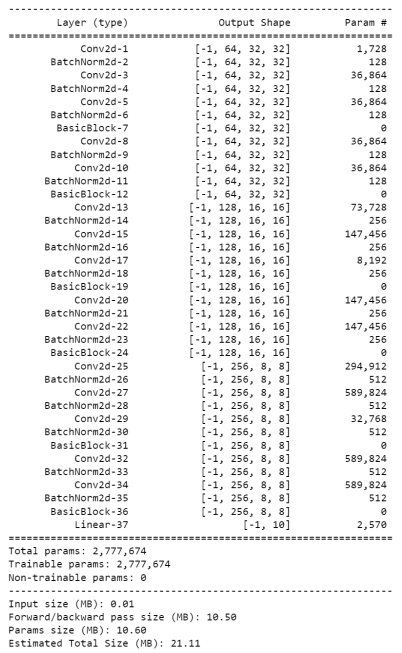

Figure 2: Summary of the ModifiedResNet model showcasing parameters and output shapes.

- **In-Depth Analysis of Filter and Network Sizes:** We experimented with different filter sizes and depths to find a balance between expressiveness and efficiency.
- **Shortcut Connections:** Inspired by the ResNet architecture, our network uses these connections to combat the vanishing gradient problem, enabling the training of deeper networks.
- **Blocks Configuration:** We used a BasicBlock as the building unit with expansion parameters to manage the channel dimensions across the network efficiently.

Our ModifiedResNet architecture's choice was influenced by the need for a model that can effectively capture the complex patterns inherent in the CIFAR-10 dataset without succumbing to overfitting. The BasicBlock's design, featuring two convolutional layers with Batch Normalization and ReLU activations, was pivotal in our model's ability to learn effectively. These choices facilitated a model architecture that is both deep enough to learn complex features and efficient enough to train within a reasonable timeframe.

Data Augmentation Strategy

A diverse set of augmentation techniques was used, aiming to simulate various visual perspectives and conditions to create a robust model:

- **Random Cropping and Horizontal Flipping:** These augmentations introduce geometric variability into the dataset.
- **Color Jittering and Random Rotation:** They encourage the model to learn features invariant to color shifts and minor rotational changes.

The data augmentation techniques employed were designed to expose the model to a wide variety of training scenarios, thereby enhancing its generalization capabilities. The combination of geometric transformations and color adjustments ensured that our model is robust to a range of variations that might be encountered in real-world deployment scenarios.

Training Process

Our training process incorporated the following elements:

- **Adam Optimizer:** Known for its adaptive learning rate properties, this optimizer facilitated efficient gradient descent.
- **Learning Rate Scheduler:** To handle plateaus during training, we implemented a scheduler that reduced the learning rate when the loss stagnated.
- **Loss Function:** Cross-entropy loss was used as it is well-suited for classification problems with mutually exclusive classes.

The optimization strategy's success was evident in the rapid convergence observed during the initial training epochs and the model's responsiveness to the learning rate adjustments implemented through the scheduler. This approach ensured that our training process was both efficient and capable of achieving high accuracy on the CIFAR-10 dataset.

Lessons Learned

Throughout the design process, we observed the importance of balancing model capacity with training data availability. Overly complex models tended to overfit, while too simple models failed to capture the intricacies of the dataset. Regularization through data augmentation proved to be a critical factor in enhancing model generalization.

Results and Discussion

Our optimized ResNet model achieved a test accuracy of 92.89%, marking a significant improvement over the baseline ResNet's accuracy. This enhancement is attributed to our methodological refinements, underscoring the efficacy of our proposed optimizations. Notably, the model demonstrated remarkable resilience against overfitting, as evidenced by the convergence of training and validation accuracies.

Our model's performance is evaluated based on the training and testing accuracy and loss curves, which are indicative of the learning efficacy and generalization ability of the architecture.

Training and Testing Accuracy

Figure 3 presents the training and testing accuracy curves over 50 epochs. It is observed that the model rapidly learns from the data in the initial epochs, as evidenced by the steep rise in accuracy for both training and testing. The curves start to plateau as the model begins to converge, showing minimal overfitting due to the close alignment between training and test accuracy.

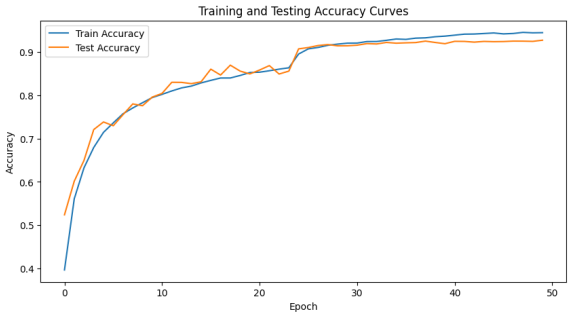

Figure 3: Training and testing accuracy curves.

Training and Testing Loss

Figure 4 illustrates the loss curves for training and testing. The training loss decreases sharply and then gradually, indicating robust learning. The testing loss mirrors this trend, suggesting that the model generalizes well to unseen data. The minimal gap between the training and testing loss underlines the effectiveness of the regularization and data augmentation strategies employed.

The results demonstrate that the modifications made to the traditional ResNet model and the training procedure have a positive impact on performance. The refined model architecture and training strategies lead to a test accuracy of 92.89%,

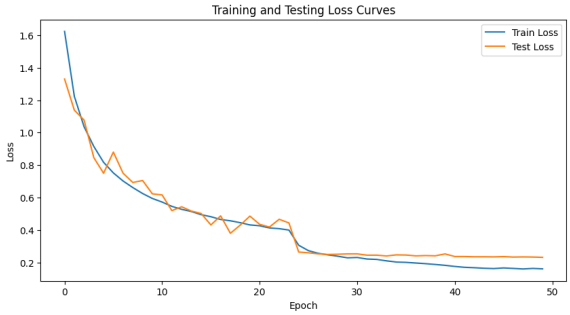

Figure 4: Training and testing loss curves.

with the model exhibiting good generalization across the epochs.

Conclusion

In conclusion, our study demonstrates that with thoughtfully designed architectures and training strategies, it is possible to enhance the performance of deep learning models on benchmark datasets like CIFAR-10 without resorting to computationally intensive solutions. Our future endeavors will explore the application of the insights gained from this project to other computer vision tasks and datasets, as well as the incorporation of newer and more advanced neural network architectures.

The methodology and results of this study contribute to the broader discourse on efficient model design in the field of deep learning, suggesting that a strategic approach to model architecture and training can yield significant improvements in performance, even with resource constraints.

Acknowledgments

We would like to thank Professor Chinmay Hegde and our peers at New York University Tandon School of Engineering for their guidance and insightful discussions that greatly assisted this research.

References

Das, S. 2023. Implementation of ResNet Architecture for CIFAR-10 and CIFAR-100 Datasets. https://medium.com/@shuvamd210/unveiling-the-metrics-mystery-exploring-scg-and-ccg-for-image-model-evaluation-76301445f583.

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

Liu, K. 2021. PyTorch CIFAR Models. https://github.com/kuangliu/pytorch-cifar.