Deep Learning Mini Project – Image Classification (Spring 24 - NYU): Enhancing CIFAR-10 Performance With Modified ResNet

Giorgi Merabishvili[1], Aaftab Mohammad[2], Mohd Faizaan Khan[3]

[1]New York University, Tandon School of Engineering, Department of Electrical and Computer Engineering (ECE)
[2]New York University, Tandon School of Engineering, Department of Electrical and Computer Engineering (ECE)
[3]New York University, Tandon School of Engineering, Department of Computer Science and Engineering (CSE)
Mini-Project GitHub Repository: click here

Abstract

This project studies the use of modified residual network (ResNet) architecture for CIFAR-10 image classification under 5 million parameters. To outperform established accuracy benchmarks, we conducted extensive experiments, leveraging different parameters and techniques. We analyzed the results of our experiments, presenting the findings in this paper.

Introduction

In the field of deep learning, achieving high accuracy in image classification using compact models is critical for resource-constrained applications. This project shows a modified ResNet architecture designed to perform well on the CIFAR-10 dataset while keeping the parameter count around 5 million. We created a model that complies to the severe parameter limitation, and also achieves excellent classification accuracy by balancing architectural changes, optimization strategies, and data preprocessing. Our findings, which show a final test accuracy of 94.16%, demonstrate the efficiency of our approach in pushing the frontiers of what is possible with limited computational resources. This work seeks to contribute to the larger discussion on creating efficient yet powerful deep learning models for picture classification problems.

Related Work

The pursuit of efficient deep learning architectures has been a focus of research, driven by the growing demand for deploying high-performing models in resource-constrained situations. Pioneering efforts, such as the original ResNet architecture, established the foundation for deep residual learning, allowing models to attain amazing accuracy on a variety of datasets, including CIFAR-10. Subsequent inventions, such as MobileNets and EfficientNets, have further this trend by creating lightweight models that do not compromise performance. These models use techniques like depthwise separable and scalable architecture to improve efficiency. Our work is inspired by these improvements, and we strive to achieve a balance between model size and accuracy. By adding changed block structures and optimization methodologies, our project stands out aimed at optimizing deep learning models for performance and efficiency. This research contextualizes our contributions within these advances, emphasizing the originality and impact of our modified ResNet architecture in the larger landscape of efficient deep learning.

Architecture

The project's foundation is a sophisticated adaption of the ResNet architecture customized for the CIFAR-10 dataset, with a primary focus on keeping the model size under 5 million parameters. At its core, the architecture uses a modified ResNet model, beginning with an initial layer (Conv2d) with a kernel size of 3x3, stride of 1, and padding of 1 applied to the 3-channel input image, which expands the feature dimension to 66. This is followed by batch normalization (BatchNorm2d), which stabilizes learning by normalizing the layer's inputs. Our ResNet variation employs a series of BasicBlock modules, each of which has two layers with 3x3 kernels, followed by batch normalization and ReLU activations, so increasing non-linear learning capabilities while avoiding the vanishing gradient problem.

A significant change is the addition of a shortcut connection in each block to assist identity mapping, which bridges the input and output of the block with a layer in cases of dimensional mismatch.

The design develops via three major stages, each of which doubles the number of feature channels from

66 to 132, and then to 264, allowing for increasingly finer feature extraction, which is required for accurate picture categorization. The transition between stages is managed by stride change in each stage's initial layer, essentially halving spatial dimensions while doubling depth while retaining network efficiency. The model converges on an adaptive average pooling layer (AdaptiveAvgPool2d), which reduces each feature map to a single value and flattens the network output in preparation for the final classification layer. A linear transformation (Linear) converts the pooled features into the ten classes of the CIFAR-10 dataset. Optimizations such as stride and channels adjustments are methodically calculated to meet the under-5-million parameter constraint, resulting in a model that balances efficiency and efficacy, as proven by the overall parameter count of 4,916,284. This careful engineering ensures an optimal mix of model depth, computational resource allocation, and classification performance, resulting in commendable accuracy while remaining within the parameter budget.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 66, 32, 32]	1,782
BatchNorm2d-2	[-1, 66, 32, 32]	132
Conv2d-3	[-1, 66, 32, 32]	39,204
BatchNorm2d-4	[-1, 66, 32, 32]	132
Conv2d-5	[-1, 66, 32, 32]	39,204
BatchNorm2d-6	[-1, 66, 32, 32]	132
BasicBlock-7	[-1, 66, 32, 32]	0
Conv2d-8	[-1, 66, 32, 32]	39,204
BatchNorm2d-9	[-1, 66, 32, 32]	132
Conv2d-10	[-1, 66, 32, 32]	39,204
BatchNorm2d-11	[-1, 66, 32, 32]	132
BasicBlock-12	[-1, 66, 32, 32]	0
Conv2d-13	[-1, 66, 32, 32]	39,204
BatchNorm2d-14	[-1, 66, 32, 32]	132
Conv2d-15	[-1, 66, 32, 32]	39,204
BatchNorm2d-16	[-1, 66, 32, 32]	132
BasicBlock-17	[-1, 66, 32, 32]	0
Conv2d-18	[-1, 132, 16, 16]	78,408
BatchNorm2d-19	[-1, 132, 16, 16]	264
Conv2d-20	[-1, 132, 16, 16]	156,816
BatchNorm2d-21	[-1, 132, 16, 16]	264
Conv2d-22	[-1, 132, 16, 16]	8,712
BatchNorm2d-23	[-1, 132, 16, 16]	264
BasicBlock-24	[-1, 132, 16, 16]	0
Conv2d-25	[-1, 132, 16, 16]	156,816
BatchNorm2d-26	[-1, 132, 16, 16]	264
Conv2d-27	[-1, 132, 16, 16]	156,816
BatchNorm2d-28	[-1, 132, 16, 16]	264
BasicBlock-29	[-1, 132, 16, 16]	0
Conv2d-30	[-1, 132, 16, 16]	156,816
BatchNorm2d-31	[-1, 132, 16, 16]	264
Conv2d-32	[-1, 132, 16, 16]	156,816
BatchNorm2d-33	[-1, 132, 16, 16]	264
BasicBlock-34	[-1, 132, 16, 16]	0
Conv2d-35	[-1, 132, 16, 16]	156,816
BatchNorm2d-36	[-1, 132, 16, 16]	264
Conv2d-37	[-1, 132, 16, 16]	156,816
BatchNorm2d-38	[-1, 132, 16, 16]	264
BasicBlock-39	[-1, 132, 16, 16]	0
Conv2d-40	[-1, 264, 8, 8]	313,632
BatchNorm2d-41	[-1, 264, 8, 8]	528
Conv2d-42	[-1, 264, 8, 8]	627,264
BatchNorm2d-43	[-1, 264, 8, 8]	528
Conv2d-44	[-1, 264, 8, 8]	34,848
BatchNorm2d-45	[-1, 264, 8, 8]	528
BasicBlock-46	[-1, 264, 8, 8]	0
Conv2d-47	[-1, 264, 8, 8]	627,264
BatchNorm2d-48	[-1, 264, 8, 8]	528
Conv2d-49	[-1, 264, 8, 8]	627,264
BatchNorm2d-50	[-1, 264, 8, 8]	528
BasicBlock-51	[-1, 264, 8, 8]	0
Conv2d-52	[-1, 264, 8, 8]	627,264
BatchNorm2d-53	[-1, 264, 8, 8]	528
Conv2d-54	[-1, 264, 8, 8]	627,264
BatchNorm2d-55	[-1, 264, 8, 8]	528
BasicBlock-56	[-1, 264, 8, 8]	0
AdaptiveAvgPool2d-57	[-1, 264, 1, 1]	0
Linear-58	[-1, 10]	2,650

Total params: 4,916,284
Trainable params: 4,916,284
Non-trainable params: 0

Input size (MB): 0.01
Forward/backward pass size (MB): 16.63
Params size (MB): 18.75
Estimated Total Size (MB): 35.40

None

Figure 1: Model Summary

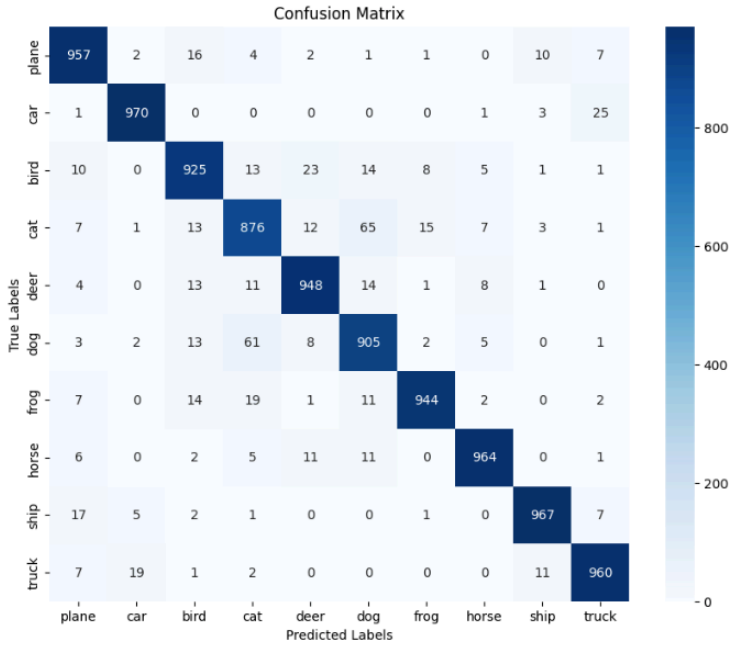

Figure 2: Confusion Matrix

Methodology

- **Environment Setup and Reproducibility:**
  Initialization of a fixed seed for all random operations to ensure consistent and reproducible results.
  Use of PyTorch's deterministic mode to further enhance reproducibility across runs.

- **Data Preprocessing**: Application of data augmentation techniques such as random crops and horizontal flips to the training dataset, enhancing the model's generalization capability.
  Normalization of images using dataset-specific mean and standard deviation values to standardize input data distribution.

- **Architecture Customization**: Design of a modified ResNet architecture, initiating with a layer that increases the input channel size to 66, optimizing feature extraction capabilities.
  Implementation of **BasicBlock** modules as the architectural cornerstone, incorporating shortcut connections to facilitate effective deep learning without performance degradation.
  Strategic adjustments in channel sizes across the network's stages to maximize depth and learning capacity while maintaining the model within the parameter budget.

- **Training Strategy**: Utilization of the CrossEntropyLoss function to handle multi-class classification, combined with the Adam optimizer for efficient backpropagation.
  Adoption of a learning rate scheduler to adjust the learning rate over epochs, optimizing the training process and improving model convergence.

- **Evaluation and Analysis**: Performance evaluation on both validation and test datasets to measure the model's accuracy and generalization to unseen data.
  Detailed analysis of training and validation loss and accuracy trends over epochs, providing insights into model learning behavior and stability.
  Use of a confusion matrix to identify and analyze the model's classification performance across different classes, highlighting strengths and potential areas for improvement.

Results

Modified ResNet developed for the CIFAR-10 image classification task demonstrated substantial progress over the course of 60 epochs. Starting from an initial state where the model's understanding of the data was relatively basic, with training and validation accuracies at 10.76% and 11.08% respectively, the model's performance improved significantly through the training process.

The training process was characterized by a steady increase in accuracy and a decrease in loss, both on the training and validation datasets. By the first epoch, the model had shown encouraging results, with a training accuracy of 49.15% and a validation accuracy of 56.30%. This upward trend persisted, with major achievements during the 10th, 20th, 30th, and final epochs. By the tenth epoch, the model had achieved 84.23% training accuracy and 84.37% validation accuracy, demonstrating the effectiveness of the learning process.

By the 59th epoch, the model's accuracy had risen to 99.36% for training and 94.25% for validation. The last epoch demonstrated the model's high degree of acquired generalizability and capacity to accurately categorize unseen data, marking a critical milestone in the study.

These findings demonstrate the ability to learn complicated patterns from picture data, and also highlight the value of a well-structured training regimen and model design. The model's steady improvement in accuracy and loss reduction over time demonstrates the efficiency of the strategies used, which include data augmentation, regularization, and a carefully selected optimization strategy.

The project concludes with a highly accurate model, adept at navigating the intricacies of the CIFAR-10 dataset, thus setting a robust foundation for future explorations in image classification.

In Figure 3, we plot the training and validation loss curves against each epoch. Our model converged at a training loss of 0.5652 and a validation loss of 0.6888.

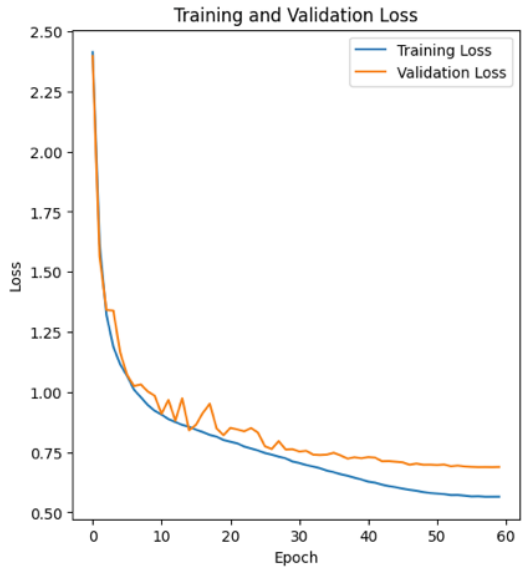

Figure 3: Confusion Matrix

In Figure 4, we plot the training and validation accuracy curves against each epoch. Our model converged at a training accuracy of 99.36% and a validation accuracy of 94.25%.

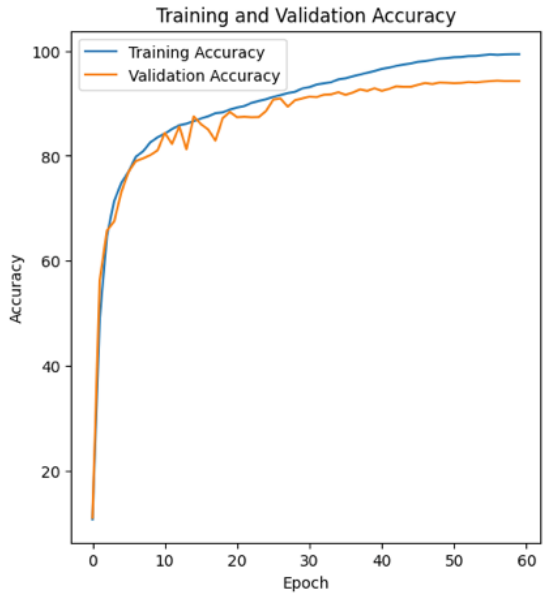

Figure 4: Confusion Matrix

System Specification

CPU: Intel(R) Xeon(R) CPU @ 2.20GHz
GPU: Tesla V100-SXM2-16GB
System Memory: 50.9937 GB
Python Version: 3.10.12
CUDA version: 12.1
Torch Version: 2.2.1+cu121

References

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770– 778.

Shuvam Das (2023). Implementation of ResNet Architecture for CIFAR-10 and CIFAR-100 Datasets.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report. Liu, Y.; Gao, Y.; and Yin, W. 2020. An improved analysis of stochastic gradient descent with momentum. Advances in Neural Information Processing Systems, 33: 18261–18271.