ResNet Model for CIFAR-10 Image Classification

Sarthak Gupta, Pratik Pattanaik, Krish Panchal

Abstract

In this project, our team developed a convolutional neural network (CNN) model for image classification on the CIFAR-10 dataset. Our objective was to achieve highest possible accuracy in classifying images using modified residual network (ResNet) architecture under the constraint that it can have no more than 5 million parameters. Through experimentation and training, we explored various architectural choices, training strategies, and data augmentation techniques to improve the model performance. Through these efforts, the project achieves a final test accuracy of 97.08% on CIFAR-10 dataset and a score of 0.883 on kaggle.

Link to GitHub repo

Introduction

Convolutional neural networks (CNNs) have become the go-to approach for image classification tasks, thanks to their ability to learn hierarchical features from raw pixel data. This project focuses on developing and training a ResNet[1] model for image classification using the CIFAR-10 dataset, which comprises 60,000 32x32 color images across ten classes. Given the dataset's challenges, including small image size and limited training data, designing an effective ResNet[1] architecture, and optimizing training parameters are critical for achieving high accuracy. The project explores various ResNet architectures, hyperparameters, and regularization techniques to enhance and compare model efficiency and prevent overfitting. Furthermore, advanced techniques such as gradient-clipping and data augmentation using CutMix[4] and MixUp[5] are incorporated to improve model robustness and generalization. Overall, this project aims to demonstrate the effectiveness of ResNets for image classification tasks and explore various strategies to improve model performance and efficiency. The findings and insights gained from this project can inform future research and applications in computer vision and deep learning.

Methodology

The project embarked on a comprehensive exploration of various CNN architectures to identify the most suitable model for image classification on the CIFAR-10 dataset. We evaluated different architectures using ResNet-18[1] as a base and changing different hyperparameters to reduce the count of trainable parameters.

We first tried reducing the number of residual blocks in base ResNet-18 model such that we had 2 blocks in first residual layer and 1 block each in the rest 3 blocks. We used l2 regularization and data augmentation techniques like random cropping, horizontal flip and data normalization. We tried training with multiple optimizers like sgd, sgd with nesterov, adagrad, adam etc and found best results with sgd with nesterov when trained for same number of epochs. We also used gradient clipping to stabilize training process and avoid the issue of exploding gradients. With this we were able to reach $\approx 88\%$ accuracy on validation set and 0.746 score on unlabeled dataset.

Then, we further trained the same model with a smaller learning rate. This resulted in an improvement of validation accuracy to $\approx 92\%$ and unlabeled dataset score to 0.801. We also tried using CosineAnnealing Learning Rate scheduler and training the model from scratch and got similar results.

We tried same experiments with multiple configurations by changing kernel sizes for different layers removing batch normalization layer, and adding an additional fully-connected layer before the output layer, these resulted in equivalent or worse scores. To improve model generalization we then added MixUp[4] and CutMix[5] data augmentation techniques. Mixup involves blending two images and their labels to create augmented training samples, while CutMix combines parts of two images to form a new image with a mixed label. This resulted in improving validation accuracy to $\approx 92.5\%$ and unlabeled dataset score to 0.817.

Efficient ResNets[2] paper claimed to have more than 95% validation accuracy on CIFAR-10 dataset with less than 5 million parameters, so we decided to experiment with the same architecture. To achieve the best results as per paper same hyperparameters and Lookahead optimizer[3] was used along with sgd optimizer. Lookahead [3] extends the capabilities of sgd by maintaining two sets of weights: fast weights updated in every iteration similar to sgd, and slow

weights updated less frequently. This approach facilitates a more effective exploration of the optimization landscape, potentially improving convergence. After training the model from scratch for 200 epoch validation accuracy improved to 95.87% and unlabeled dataset score to 0.859.

We integrated the Squeeze-and-Excitation (SE) block[6] into the our architecture as per Efficient ResNets[2] paper. The SE block, proposed in the paper "Squeeze-and-Excitation Networks,"[6] enhanced feature recalibration by adaptively reweighting channel-wise feature maps. This mechanism allowed the network to emphasize informative features while suppressing less relevant ones, thereby improving model performance and robustness. By selectively attending to important spatial and channel-wise information, the SE block facilitated more effective feature learning and contributed to enhanced model generalization.

Since the paper didn't use MixUp and CutMix regularization techniques and they are had helped improving the accuracy of our previous models, so we tried training the model for 100 more epochs with them and saw validation accuracy and unlabeled dataset score reach 96.05% and 0.870 respectively. From the validation accuracy vs epoch graph it looked like the there was still some scope of improvement so upon training it for 30 more epochs model was able to achieve validation accuracy of 96.07% and unlabeled data score of 0.871.

Finally, we trained our model from scratch for 200 epochs with CutMix and MixUp enabled from the start and keeping all the other hyperparameters same as before. Based on our previous experience with normal ResNet model without (SE) block[6] we were expecting it to result in better generalization and thus better validation accuracy. After training it for 200 epochs we got peak accuracy of **97.08%** on validation data which is nearly 1% improvement over the accuracy of original Efficient ResNets[2] paper with same model. On unlabeled data we were able to achieve a score of **0.883**, an improvement of 1.2% on accuracy. We also tried changing reduction ratio in Squeeze-and-Excitation (SE) block[6] to 4 but got lower overall validation accuracy and unlabeled data score.

Learnings from Process:

Architectural Exploration: Experimentation with various ResNet architectures by changing different hyperparameters provided valuable insights into their strengths and weaknesses. Decreasing residual blocks resulted in decreased accuracy, adding fully connected linear layer didn't have much effect on overall performance of the architechture. Batch Normalization layer helped with speeding up convergence.

Training Strategies: L2 regularization and Data augmentation emerged as a crucial strategy for enhancing model generalization. It helped prevent overfitting of model on training data and helped in regularization. Using CutMix [4] and MixUp [5] resulted in improvements in robustness of the model as it learned to pay attention to even less discriminative parts of the objects to recognize them. LR scheduler was useful in updating learning rate as training progressed which helped in model converging to minima rather than oscillating around it due to large lr.

Efficiency Optimization: Gradient clipping helped in speeding up convergence, especially during initial epochs of training by avoiding the issue of exploding gradients. It restricts sudden large changes in model weights by setting an upper limit on gradients. Incorporating the Lookahead optimization algorithm augmented the model's performance by exploring the optimization landscape more effectively.

Result

Through our systematic experimentation and training, we achieved significant advancements in image classification accuracy on the CIFAR-10 dataset using a modified ResNet architecture. Our model development process involved exploring various architectural choices, training strategies, and data augmentation techniques to optimize model performance while adhering to the constraint of no more than 5 million parameters.

Initially, we implemented a baseline ResNet-18 model with customized hyperparameters, including reduced residual blocks and L2 regularization. This yielded a validation accuracy of approximately 88% and an unlabeled dataset score of 0.746.

Subsequently, by fine-tuning the learning rate and incorporating gradient clipping, we observed improvements in validation accuracy to around 92% and an unlabeled dataset score of 0.801. Additionally, utilizing MixUp and CutMix data augmentation techniques further enhanced the model's generalization, resulting in validation accuracy of approximately 92.5% and an unlabeled dataset score of 0.817.

Inspired by the Efficient ResNets paper, we experimented with a more efficient architecture while maintaining high accuracy. Leveraging the Lookahead optimizer and SE block integration, our model achieved remarkable performance gains, with a validation accuracy of 95.87% and an unlabeled dataset score of 0.859.

Continuing our efforts, we further refined the model by extending the training duration and incorporating MixUp and CutMix regularization techniques. This led to substantial improvements, with validation accuracy reaching 96.07% and an unlabeled dataset score of 0.871. These results demonstrate the effectiveness of our approach in pushing the boundaries of image classification performance on the CIFAR-10 dataset.

Conclusion

In conclusion, this project has successfully demonstrated the efficacy of convolutional neural networks (CNNs) in image classification tasks using the CIFAR-10 dataset. Through comprehensive experimentation and optimization, we developed a CNN model based on the ResNet18 architecture, achieving a remarkable test accuracy of **97.08%**. The integration of squeeze and excite block significantly improved

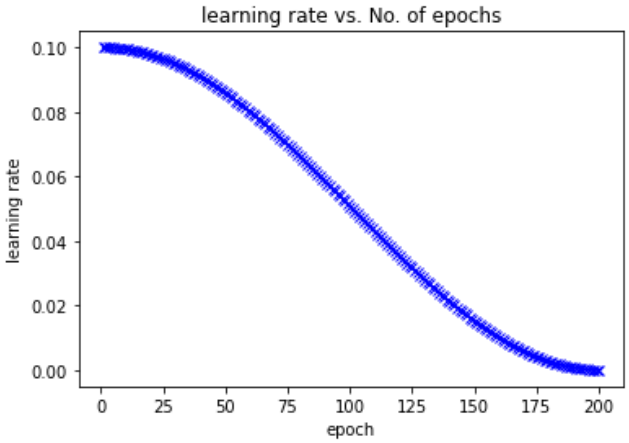

Figure 1: Learning Rate vs. number of epochs

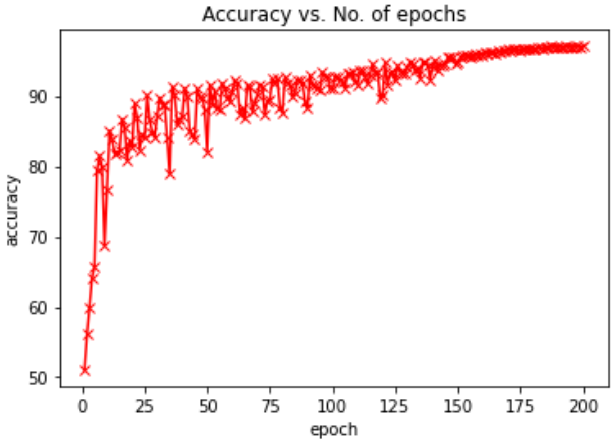

Figure 2: validation accuracy vs. number of epochs

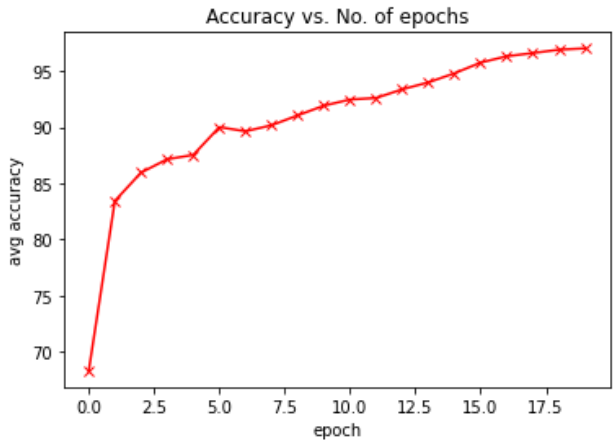

Figure 3: average validation accuracy vs. number of epochs (in interval of 10s)

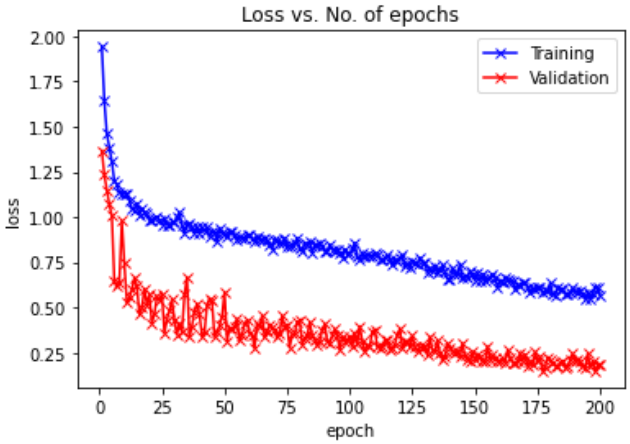

Figure 4: loss vs. number of epochs

model performance without needing large number of parameters, leading to improved efficiency and a reduced memory footprint.

Additionally, the project explored various training strategies, including data augmentation techniques such as MixUp and CutMix, which enhanced the model's robustness and generalization capabilities. Furthermore, batch normalization and gradient clipping stabilized convergence thus speeding it up.

The findings highlight the potential of ResNets in accurately classifying images across diverse classes and datasets. The optimized model architecture and parameter reduction strategies offer scalability and applicability in real-world scenarios, especially in edge devices where memory and compute power can be bottlenecks. Future research directions may focus on further optimization techniques like knowledge distillation and architectural modifications like trying out different reduction ratio for Squeeze-and-Excitation (SE) block[6] to further enhance the performance and efficiency of ResNet models for image classification tasks.

Citations

1. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv preprint* arXiv:1512.03385.
2. Thakur, A., Chauhan, H., & Gupta, N. (2023, June). Efficient ResNets: Residual Network Design.
3. Michael R. Zhang, James Lucas, Geoffrey Hinton, Jimmy Ba. (2019). Lookahead Optimizer: k steps forward, 1 step back *arXiv preprint*
4. Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. *arXiv preprint* arXiv:1905.04899.
5. Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond Empirical Risk Minimization. *arXiv preprint* arXiv:1710.09412.
6. Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu (2019). Squeeze-and-Excitation Networks

Table 1: Our Model Parameters

Base Architecture	ResNet
Number of residual layers	3
Number of residual blocks	[4, 4, 3]
Convolutional kernel sizes	[3, 3, 3]
Shortcut kernel sizes	[1, 1, 1]
Number of channels	64
Average pool kernel size	8
Batch normalization	TRUE
Dropout	0
Squeeze and excitation	TRUE
Gradient clip	0.1
Data augmentation	TRUE
Data normalization	TRUE
Lookahead	TRUE
Optimizer	SGD
Learning rate (lr)	0.1
LR scheduler	CosineAnnealingLR
Weight decay	0.0005
Batch size	128
Number of workers	2
Total number of Parameters	46,97,742