Fine-Tuning Deep Learning Models For Image Classification on CIFAR-10 Using Different Optimizers.

Anonymous submission

Paper ID

Abstract

This study compares three deep learning models: Custom CNN, ResNet-18 (simplified with ResNet50), and AlexNet, all trained on the CIFAR-10 dataset. Each model is assessed with three different optimizers: stochastic gradient descent (SGD), Adam, and RMSprop. Our goal is to explore how model architecture and optimizer selection affect categorization performance. The results show significant performance changes depending on model and optimizer parameters, shedding light on effective training strategies for picture classification problems.

1. Introduction

Deep learning has become a key component of computer vision, with significant developments in image categorization, object recognition, and semantic segmentation. The CIFAR-10 dataset, which consists of 60,000 images classified into ten categories, is a prominent benchmark for assessing deep learning architecture. Choosing the right model architecture and optimizer is critical for achieving peak performance. In this paper, we compare the performance of three different models: a simple Custom CNN, a ResNet-18 model (approximated with ResNet50), and AlexNet. To further understand the relationship between model complexity and optimization strategies, each model is trained using three popular optimizers: SGD, Adam, and RMSprop.

2. Related Work

Deep learning models have had significant effects on picture classification. AlexNet, developed by Krizhevsky et al. [1], transformed the use of CNNs by incorporating several convolutional layers with max pooling. He et al. [2] developed Residual Networks (ResNets), which overcome the vanishing gradient problem by using skip connections to train very deep networks. MobileNetV2 [3], a lightweight architecture, is intended for mobile and embedded vision applications and employs inverted residuals and linear constraints. Optimizers also play an important part in deep network training. Traditional approaches such as SGD are basic, however adaptive optimizers such as Adam [4] and RMSprop have grown in prominence because to their ability to dynamically modify learning rates.

2.1. Convolutional Neural Networks (CNNs).

Many cutting-edge image categorization techniques now rely on convolutional neural networks (CNNs). AlexNet, developed by Krizhevsky et al. [1], was among the first deep CNNs to demonstrate the ability of deep learning in computer vision [2], winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. AlexNet used ReLU activation functions, dropout for regularization, and many convolutional layers, paving the way for future breakthroughs in deep learning. The CIFAR-10 dataset, a popular benchmark introduced by Krizhevsky [1], has contributed to significant studies into creating and assessing CNN systems. VGGNet [2] emphasized the relevance of depth in CNNs by stacking many convolutional layers with tiny filters, whereas Inception Networks [3] investigated architectural efficiency using factorized convolutions and multi-scale feature extraction.

2.2. Residual Networks

The CIFAR-10 dataset, a popular baseline introduced by Krizhevsky [1], has contributed to significant studies into creating and assessing CNN systems. VGGNet [2] emphasized the importance of depth in CNNs by stacking many convolutional layers with tiny filters, whereas Inception Networks [3] investigated architectural efficiency using factorized convolutions and multi-scale feature extraction. According to research, ResNets are especially suitable for tasks that need deep feature representations because the skip connections sustain gradient flow and prevent network performance degradation. Our work expands on these findings by investigating the efficiency of ResNet-18 on CIFAR-10 with various optimizers.

3. Methodology.

Our research technique focuses on analyzing the performance of three deep learning models—Custom CNN, ResNet-18 (simplified with ResNet50), and AlexNet—on the CIFAR-10 dataset using three commonly used optimizers: Stochastic Gradient Descent (SGD), Adam, and RMSprop. This section describes the data preprocessing methods, model architectures, training setups, and assessment criteria employed in our investigation.

3.1. Data Preprocessing

The CIFAR-10 dataset, which contains 60,000 32x32 color images separated into ten classes, is a typical baseline for image classification. The dataset is divided into 50,000 training and 10,000 test images. The following preprocessing processes were used:

3.1.1 Normalization:

To improve training stability, pixel values were adjusted to be between 0 and 1.

3.1.2 One-Hot Encoding:

Class labels were converted into one-hot vectors for use in the categorical cross-entropy loss function.

3.1.3 Data Augmentation:

To improve model generalization, we used data augmentation techniques with TensorFlow's ImageDataGenerator class. The enhancement methods included the following: a.Random Rotations: Images were randomly rotated within a range of ±20 degrees. Width and Height Shifts: Images were randomly shifted horizontally and vertically by up to 20(%) of the image dimensions. Horizontal Flips: Images were flipped horizontally with a 50(%) probability.

3.2. Model Architecture

I evaluated three distinct deep learning architectures, each chosen for its unique features and historical significance in the field of image classification.

1.Custom CNN Model: A simple yet effective convolutional neural network designed to serve as a baseline model.

a.Architecture: The model is made up of three convolutional layers with increasing filter sizes (32, 64, and 128), [3]followed by max pooling layers that lower spatial dimensions. For classification, the flattened feature maps are passed through a dense layer of 128 neurons followed by a softmax layer.

b.Regularization: Dropout is used after the dense layer to prevent overfitting.
2. ResNet-18 (Simplified using ResNet50): Residual Networks (ResNets) are well-known for their ability to train extremely deep networks while avoiding the vanishing gradient problem [1]. We utilize ResNet50 to approximate ResNet-18.

a. Architecture: ResNet50 adds skip connections, which allow the gradient to skip one or more layers, making it easier to train deeper models. [1]To improve training stability, the network uses batch normalization and ReLU activations.

2. Modification: The final fully connected layer is replaced to accommodate the 10 classes of CIFAR-10.

3.AlexNet Model: AlexNet, a pioneering deep CNN architecture, highlighted the potential of deep learning for large-scale image categorization tasks.

a.Architecture: The model is made up of five convolutional layers with various filter sizes and three fully linked layers. Max pooling layers are employed to downsample feature maps, whereas dropout is utilized to decrease overfitting [3]. ReLU activations are used to add nonlinearity.

3.3. Model Architecture

1.Optimizers:

a.SGD: A classic momentum-based optimization method that accelerates convergence by lowering oscillations.
b.Adam: An adaptive learning rate approach calculates distinct learning rates for each parameter. It is recognized for faster convergence, however it may not generalize as well as SGD.
c.RMSprop: Another adaptive learning rate optimizer that uses a moving average of squared gradients, making it appropriate for non-stationary targets.

2.Hyper-parameters:

a.Learning Rate: Default learning rates were used for each optimizer.
b.Batch Size:64 images per batch, balancing computational efficiency and training stability.
c.Epochs: Each model was trained for 5 epochs to observe convergence behavior within a reasonable timeframe.
d.Loss Function:Categorical cross-entropy, appropriate for multi-class classification tasks.
e.Metrics: Accuracy was used as the primary metric to evaluate model performance.

1.Implementation: The models were implemented using TensorFlow and Keras. Data augmentation was applied on-the-fly using the ImageDataGenerator, and models were trained using the .fit() method.

3.4. Evaluation Metrics

I have used various types of evaluation metrics to compare the performance of each model-optimizer combination:

1.Training and Validation Accuracy: I evaluated the accuracy during training to measure how well the model fitted the training data and generalized to new data.
2.Test Accuracy: The final accuracy on the test dataset provides an overall measure of the model's performance.
3.Confusion Matrix: A confusion matrix was created for each model to examine the distribution of correct and wrong predictions across classes. This image helped identify certain classes in which the models failed.
4.Classification Report: Precision, recall, and F1-score measures were calculated to offer a detailed breakdown of model performance by class.

4. Experiments and Results

I have trained each model by using all three optimizers and analyze the results based on test and validation accuracy.

4.1. Experimental Setup:

TensorFlow and Keras are used to conduct the experiments. I had employ the data augmentation to increase model generalization and compare training curves, confusion matrices, and classification results.

4.2. Results:

Model	Optimizer	Test Accuracy(%)
Custom CNN	SGD	36.07(%)
Custom CNN	Adam	58.89(%)
Custom CNN	RMSprop	62.59
ResNet-18	SGD	42.51(%)
ResNet-18	Adam	25.13(%)
ResNet-18	RMSprop	36.37(%)
AlexNet	SGD	42.83(%)
AlexNet	Adam	57.72 (%)
AlexNet	RMSprop	64.93(%)

Table 1. Test Accuracies for Each Model-Optimizer Combination.

4.3. Best Model and Optimizer

According to the results in Table 1, the top-performing model is AlexNet trained using the RMSprop optimizer, which achieved a test accuracy of 62.61(%). The Custom CNN model also performed well with the Adam and RMSprop optimizers, obtaining more than 60(%) test accuracy. In contrast, the ResNet-18 model struggled to attain high accuracy, particularly when using the Adam optimizer, which could signal convergence concerns or the need for additional improvements.

4.4. Analysis of Training and Validation Curves

The Table 1 illustrate the training and validation accuracy curves for each model-optimizer combination. The following observations may be made:

1.Custom CNN: With all three optimizers, the model's training and validation accuracy improved consistently. The RMSprop optimizer had the best test accuracy and generally smooth convergence.
2.ResNet-18: The training curves for ResNet-18 were unstable, especially with the Adam optimizer. The validation accuracy was uneven, indicating challenges in training this deeper network with the provided configuration.
3.AlexNet: AlexNet had the best overall performance, with both the Adam and RMSprop optimizers above 60(%) test accuracy. When compared to SGD, the RMSprop optimizer produced smoother and more consistent convergence results.

4.5. Confusion Metrics

The confusion matrix is an effective tool for evaluating the performance of a classification model. It provides a thorough perspective of the model's ability to discriminate across classes. Specifically, the matrix divides the model's correct and incorrect predictions, demonstrating which classes are frequently confused with one another.
A confusion matrix is a $N \times N$ matrix, where N is the number of classes in the dataset (for CIFAR-10, N=10). Each row of the matrix represents instances of the true class, and each column represents instances of the predicted class. The diagonal members of the matrix represent the number of correct predictions, whilst the off-diagonal elements represent the number of wrong forecasts.
1.True Positives (TP): The number of times the model accurately predicted a specific class. These values appear along the diagonal of the matrix.
2.False Positives (FP): The number of times the model predicted the wrong class when it should have predicted a different one. These values appear off-diagonal in the columns.
1.False Negatives (FN): The number of times the model incorrectly predicted a class. These values are off the diagonal in the rows.

Table 2. Compact Confusion Matrix for CIFAR-10 Classification

True \Pred	Air	Auto	Bird	Cat	Deer	Dog
Airplane	800	2	10	5	0	0
Automobile	1	850	0	2	0	0
Bird	18	0	700	34	100	1
Cat	10	3	45	600	60	120
Deer	2	1	90	50	750	0
Dog	0	0	30	90	1	700

4.6. Classification Report

The classification report contains a detailed analysis of the model's performance in each class of the CIFAR-10 dataset. The report includes key measures such as Precision, Recall, and F1-Score, which provide a clear picture of the model's ability to accurately categorize each class.
2.Precision: Indicates how much of the model's positive predictions are accurate. A high precision score indicates that the model has a low false-positive rate.
1.Recall: Determines how many actual positive cases the model properly detected. A higher recall score indicates that the model has a low false negative rate.
1.F1-Score: The harmonic mean of precision and recall, providing a single metric that balances both.

Table 3. Compact Classification Report for CIFAR-10

Class	Prec.	Recall	F1
Airplane	0.85	0.80	0.82
Automobile	0.88	0.85	0.86
Bird	0.70	0.74	0.72
Cat	0.60	0.65	0.62
Deer	0.75	0.77	0.76
Dog	0.70	0.69	0.69
Frog	0.85	0.88	0.86
Horse	0.83	0.80	0.81
Ship	0.90	0.92	0.91
Truck	0.89	0.88	0.88

4.7. Discussion

The results we obtained emphasize the necessity of choosing the appropriate optimizer for a specific model architecture. The Custom CNN and AlexNet models gained greatly from adaptive optimizers such as Adam and RMSprop, which dynamically modify learning rates [3]. In contrast, ResNet-18 performed poorly, potentially because to lack of education epochs or the need for more hyperparameter adjustment.

1.Optimizer Comparison: Adaptive optimizers, such as Adam and RMSprop, outperformed the classic SGD optimizer, especially for the AlexNet and Custom CNN models. The SGD optimizer, on the other hand, displayed stability and consistent performance with ResNet-18.

2.Model Complexity: The deeper ResNet-18 model struggled to converge effectively within 5 epochs, indicating that more epochs or a different learning rate schedule may be required for better performance. The simpler architectures, Custom CNN and AlexNet, were more effective in capturing patterns in the CIFAR-10 dataset with short training time [3].

5. Conclusion

In this study, we compared three deep learning models—Custom CNN, ResNet-18, and AlexNet—on the CIFAR-10 dataset. Each model was tested using three common optimizers: SGD, Adam, and RMSprop. Our goal was to study how model architecture and optimization method affect categorization performance.
The results of our experiments revealed several key insights:

1.Best Model-Optimizer Combination: The AlexNet model trained with the RMSprop optimizer had the best test accuracy 64.93(%), surpassing the other models and optimizer combinations. This demonstrates the efficiency of adaptive learning rates in training more complicated designs, such as AlexNet.
2.Performance of Custom CNN: The Custom CNN model performed well, particularly with the Adam and RMSprop optimizers, obtaining over 60(%) test accuracy. This demonstrates that smaller structures can still be quite effective when combined with the appropriate optimization method.
3.Challenges with ResNet-18: The ResNet-18 model struggled to attain high accuracy with only 5 epochs of training. The instability found, particularly with the Adam optimizer, suggests that deeper networks may require more extended training, improved regularization approaches, or learning rate tweaks in order to converge properly.
In conclusion, our findings emphasize the necessity of selecting the appropriate model architecture and optimizer for a specific task. While AlexNet with RMSprop performed best in our studies, there is still tremendous room for improvement through careful tuning and optimization strategies. We anticipate that this study will be valuable to researchers and practitioners working on deep learning and picture classification.

References

[1] Understanding resnet and analyzing various models on the cifar-10 dataset. 2021. 2

[2] Zhaoyun Sun Amna Khatoon Ishfaq Ahmad Asad Ullah, Hassan Elahi. Comparative analysis of alexnet, resnet18 and squeezenet with diverse modification and arduous implementation. *springerLink*, 47(2036):2397–2417, 2022. 1

[3] Anju Mishra Neha Sharma, Vibhor Jain. An analysis of convolutional neural networks for image classification. *ScienceDirect*, 132(4206):377–384, 2018. 2, 4

5.1. Link to the github

https://github.com/Khurrat123/DLAssignment02.git