CIFAR-10 Image Classification Using Residual Network (ResNet)Architecture

Andrea Zhou, Elisha Edme, Abdul Samad Zaheer Khan
New York University
az2482@nyu.edu, eae8374@nyu.edu, ak9943@nyu.edu

Links to Supporting Material
GitHub Repository: GitHub (For the best results use learn_rate_consistent_model.pth in the model folder)

Abstract

The CIFAR-10 dataset serves as a benchmark for image classification tasks in the domains of machine learning and computer vision. The ResNet9 model is a compact version of ResNet-18, designed with a reduced number of trainable parameters to meet specific constraints. ResNet9 is defined as ResNet(BasicBlock, [1, 1, 1, 1]) and incorporates a single BasicBlock in each layer. Multiple iterations of hyperparameter tuning were meticulously conducted to optimize the model's performance. These tuning processes involved experimenting with various combinations of optimizers, learning rates, momentum, and weight decay. After thorough evaluation, the optimal set of hyperparameters was identified, resulting in an impressive training accuracy of 97% and a testing accuracy of 92%. The model was trained over 100 epochs to achieve the current levels of accuracy.

Overview

Deep neural networks, characterized by their multiple hidden layers, have proven remarkably effective in various tasks such as image recognition and natural language processing. However, constructing deeper networks often encounters a degradation issue, wherein the training error unexpectedly increases despite an increase in network capacity. Additionally, the vanishing gradient descent problem arises, where gradients diminish significantly during backpropagation, impeding learning in deeper layers. These challenges collectively limit the performance and practical applications of deep networks.

ResNet, a groundbreaking architecture that incorporates residual connections, has emerged as a promising solution to these challenges. By addressing the degradation problem and enabling more efficient training of deep networks, ResNet offers significant advantages in various domains. The effectiveness of ResNet lies in its ability to alleviate the vanishing gradient descent problem and facilitate the flow of information across layers, thereby improving the overall performance of deep neural networks.

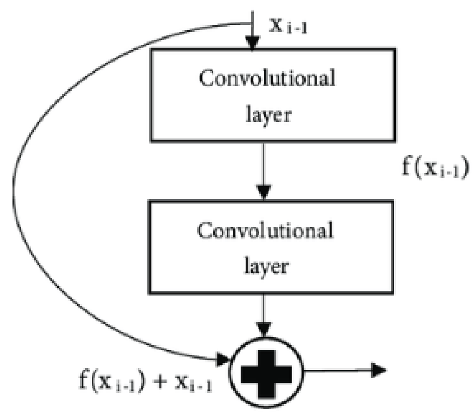

Fig. 1

ResNet's contributions to deep learning have been transformative, solidifying its status as a foundational architecture for computer vision tasks. Its innovative approach to addressing training challenges in deep networks has inspired a plethora of variants and extensions, collectively driving the field forward. The insights gleaned from ResNet have had a profound impact on the design and training methodologies of contemporary deep neural networks, facilitating their successful application across a diverse range of tasks and domains.

Data Augmentation

Data augmentation is a technique in deep learning that expands the training set by applying transformations to existing data. This includes rotations, scaling, and flipping. Augmentation enhances performance and generalization by exposing the network to a diverse set of examples. It reduces overfitting and improves the handling of data variations. It's commonly used in computer vision to boost deep neural network performance. Data augmentation techniques implemented for the current ResNet9 model are as follows:

Random Crop with Padding:

Images were augmented by adding pixels with a given width, and then the padded images were cropped to the appropriate size. The current implementation, 4 pixels were added to each 32x32 image to get a 40x40 padded image. After random cropping, the final 32x32 image is ready for training.

Normalize:

First, convert the image to a tensor using ToTensor() and then normalize the tensor image with mean and standard deviation.

Random Horizontal Flip:

This transformation randomly flips the images horizontally, creating new images with mirrored versions of the original images.

Random Rotation:

This transformation randomly rotates the images by a random angle. The current implementation randomly rotates the image between -10 and 10 degrees, augmenting the dataset with images captured from different perspectives.

About the CIFAR-10 Dataset

CIFAR-10 is a benchmark dataset used for image classification tasks in the field of machine learning and computer vision. It consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The 10 classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

The dataset is split into 50,000 training images and 10,000 test images. It is often used as a standard benchmark to evaluate the performance of image classification algorithms, and many state-of-the-art models have been trained on this dataset.

The name "CIFAR" stands for Canadian Institute for Advanced Research, which is the organization that created the dataset.

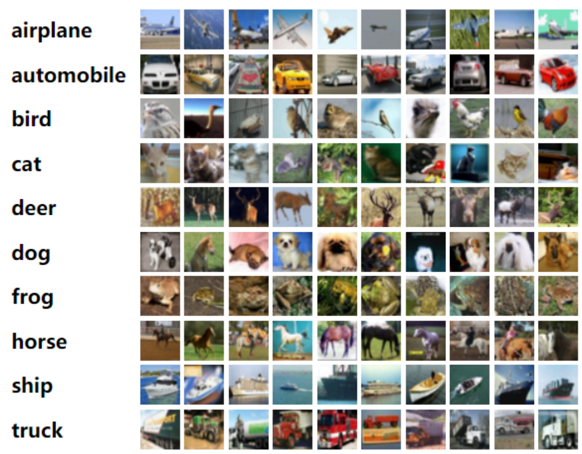

Fig. 2

Architecture

The number of basic blocks per convolutional layer was from two to just one BasicBlock in all the layers of the model, resulting in the architecture shown in Fig. 3. After this we decided to remove one residual layer from the above architecture. Also, one residual layer was removed from the architecture.

The model will consist of a preprocessing convolution layer with 64 output channels, one residual network, and one fully connected layer.

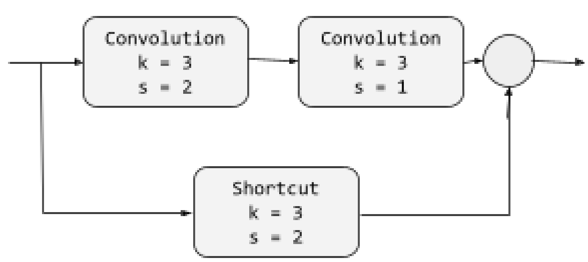

Fig. 3

Methodology

Starting with the ResNet18 architecture, our main objective was to reduce the number of trainable parameters to fulfill the 5 million parameter limit. This is done by implementing the architecture explained in the previous section. This section outlines the methodology employed for implementing the ResNet9 model for image classification on the CIFAR-10 dataset.

Data Preprocessing:

We started by downloading the CIFAR-10 dataset and subjecting it to preprocessing to ensure uniformity and compatibility with the model. To enhance the model's robustness and augment the training set, we applied data augmentation techniques, including random flipping, random rotation, and random cropping with padding. Then we normalized the images and converted them into tensors.

Model Architecture:

The ResNet9 model was constructed by leveraging the fundamental principles of the ResNet architecture, specifically utilizing the BasicBlock building block. The model comprises a preprocessing convolutional layer with 64 output channels, followed by a residual network consisting of a single residual layer, culminating in a fully connected layer

Training:

The model underwent rigorous training using the Stochastic Gradient Descent (SGD) optimizer, with a carefully chosen learning rate of 0.01 and a momentum of 0.9. To measure the discrepancy between the predicted and actual labels, we employed the cross-entropy loss function.

The model was diligently trained for 100 epochs, ensuring its optimal performance, with the batch size judiciously set to 64.

A custom function was implemented to calculate the accuracy of the model in the no_labels_dataset. Every epoch if the accuracy exceeded the previous best accuracy then the model was saved and reloaded before the next iteration to avoid a significant reduction in accuracy in the following iteration.

Evaluation:

To ascertain the efficacy of the trained model, we conducted a thorough evaluation of the CIFAR-10 test set, assessing its accuracy.

The model demonstrated acceptable performance, achieving a training accuracy of 97% and a testing accuracy of 92%, showcasing its proficiency in classifying images.

The train loss and test loss over 60 epochs are plotted in Fig. 4.

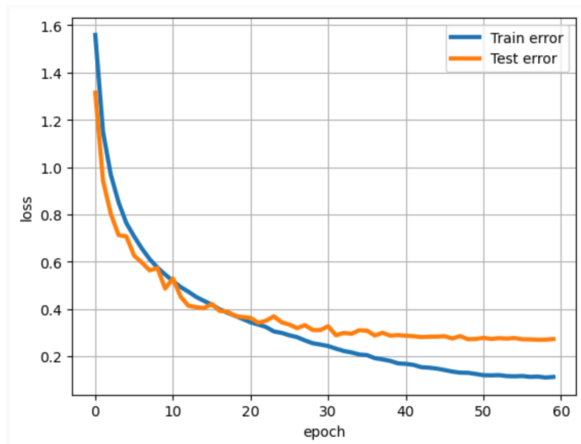

Fig. 4

The train accuracy and test accuracy of the model on the first 60 iterations of the more than 100 epochs of the model is plotted in Fig. 5.

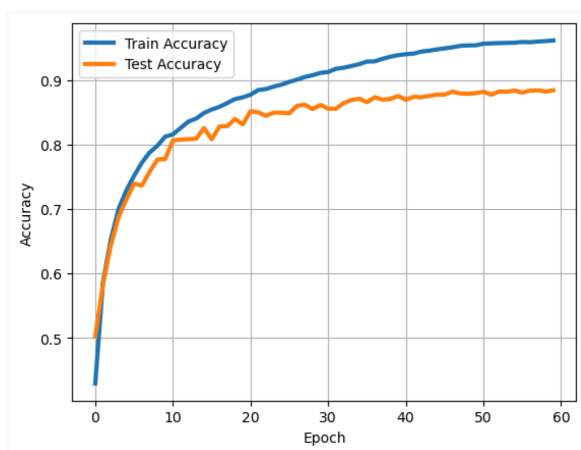

Fig. 5

Implementation Details:

The basic structure of the model was modified to meet the parameter constraints. The model was implemented with assistance from the PyTorch deep learning library, harnessing its capabilities for efficient and effective deep learning development.

Results

Performance:

- Test accuracy: 92%

Number of Parameters: 4903242

Improvements

The attempted pruning and fine-tuning of the model failed as the accuracy significantly plummeted after the pruning of parameters. Pruning could work in case of more training or proper implementation using deeper models.

Possible improvements using different architectures like Teacher-Student Architecture for Knowledge Distillation which might not meet the trainable parameters constraint. The model can also be improved using better Data Augmentation Techniques, trying Deeper or Wider Neural Networks, experimenting with Different Activation Functions like ELU, etc.

References

1. Deep Residual Learning for Image Recognition, 2016. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.
2. "Deep Learning with Python" by Francois Chollet
3. "The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman.
4. "Data Augmentation for Deep Learning" by Jason Brownlee
5. Teacher-Student Architecture for Knowledge Distillation: A Survey, 2023. Chengming Hu, Xuan Li, Dan Liu, Haolun Wu, Xi Chen, Ju Wang, Xue Liu
6. https://github.com/kuangliu/pytorch-cifar
7. https://openreview.net/forum?id=SeFiP8YAJy
8. https://www.researchgate.net/publication/352772872_Improvement_of_CIFAR-10_Image_Classification_Based_on_Modified_ResNet-34