CIFAR-10 Image Classification Based on Modified ResNet Model

Junyu Sui[1], Renhao Li[2], Minxi Bi[3]

[1,2,3] New York University

js12883@nyu.edu, rl5235@nyu.edu, mb7648@nyu.edu

Abstract

Image classification is one of the most fundamental tasks of computer vision. In this study, we defined a ResNet model that incorporates residual blocks into a normal convolutional neural network and includes 32 parameter layers to improve the accuracy of classification for CIFAR-10 images. The model was trained for 50 epochs on the Google Colab platform, leveraging data preprocessing techniques such as random flipping and cropping, and advanced training strategies including cosine annealing for learning rate adjustment and label smoothing to refine loss functions. The architecture integrates two 3x3 convolution layers followed by batch normalization and ReLU activation to effectively capture complex patterns within the dataset while managing computational efficiency. Residual blocks with skip connections were employed to mitigate degradation issues, enhancing training performance without compromising the depth of the network.

The number of total parameters is 4,693,962. Results demonstrated a consistent reduction in training and test loss. The accuracy for the validation batch in the CIFAR-10 dataset reached 92%. In the test dataset, the accuracy of our modified ResNet was 83.4%.

The source code was uploaded to Github

https://github.com/junyuSui/Team-123/blob/main/FInal_Mini_Project.ipynb

Introduction

CIFAR-10 is a multi-class dataset consisting of 60,000 32 × 32 color images in 10 classes, with 6,000 images per class. This dataset is uniquely structured into five training batches and one test batch, each batch containing 10,000 images. The test batch is composed of 1,000 randomly selected images from each class, ensuring a balanced evaluation platform. In this study, the purpose is to define a Residual Network(ResNet) model, a type of Convolutional Neural Network(CNN) that is suitable for tackling the problems regarding vanishing gradients and uses the CIFAR-10 dataset to train the designed model.

The primary goal of this project is to design a modified ResNet architecture that has no more than 5 million trainable parameters while striving to exceed a baseline accuracy of 80%. The model training was on Google Colab. To improve the performance and generalization ability, we preprocessed and augmented the data by randomly flipping, cropping, rotating and other techniques. To optimize the training process, we have run 50 epochs. During the training, a cosine annealing learning rate adjustment strategy was introduced, which modulates the learning rate in a cyclical pattern to aid in escaping local minima and refining model parameters. Additionally, we fine-tuned the loss function, specifically incorporating techniques such as label smoothing, to further enhance the model's accuracy by preventing the model from becoming overly confident. These adjustments were instrumental in stabilizing the training process during the early phases and achieving better generalization on unseen data.

Methodology

For this study, the customized ResNet 32 model had two important components in its architecture: a 3x3 Convolution layer and a Residual Block. The 3x3 convolutional layer captures 3x3 kernel input size while maintaining spatial dimensions and excluding bias terms to produce output value through the Rectified Linear Unit (ReLU) activation function. By setting two 3x3 convolutional layers, it can capture a similar receptive field compared to other kernel sizes(e.g. 5x5) but with fewer parameters and more non-linear transformations

The Residual Block was fundamental to the ResNet model as it tackled the degradation to saturate the accuracy during training. Each block contained two 3x3 convolutional layers followed by batch normalization to control the internal covariate shift, thus impeding the model's training. The model used the ReLU activation function, a non-linear transformation function to learn complex patterns. Skip connection inside the residual block was vital to the learning process. First, it identified shortcuts by passing the input directly to the end of the block, which ensured that deeper layers could at least perform the same as shallower ones by learning the identity function as a baseline. Meanwhile, it can promote the network to learn the residual function more easily than learning the unreferenced function directly. Specifically for CIFAR-10, it can learn more abstract and complex features, which allows additional data augmentation during the training dataset setup.

The ResNet network performs as follows, starting with two 3x3 convolution layers, they expand the input's channel depth to 64, followed by batch normalization and ReLU activation function. Then construct four residual layers, each with 3, 4, 6, and 2 residual blocks, respectively. For the remaining three residual layers, the input's channel expands to 64, 128, and 256 respectively. Ideally, the input of each channel should be expanded accordingly, but due to the limitation of no more than 5 million parameters, the input of channels is set as described above. An adaptive average pooling layer is processed after residual layers to reduce each feature map to a single value with a dropout layer set as 0.5 to prevent overfitting. The final output layer is a fully connected layer, which helps to reduce the feature to 10 because of the class number in the CIFAR-10 dataset. The model also contains a downsampling sequential module, in case the stride is not 1 or the number of input channels does not match with the number of output channels.

Prior to training on the CIFAR-10 dataset, images were converted to tensors and normalized to achieve uniform data scale and distribution among inputs. Normalization parameters were set with mean values of [0.4914, 0.4822, 0.4465] for mean values across RGB channels respectively, and standard deviation values between 0.243 - 0.261 respectively. This step improves the training stability and convergence speed of the neural network by ensuring that the feature distributions are similar across different input variables. To increase the model's ability to generalize from the training set to unseen data, several augmentation techniques were employed during training:

- Random Horizontal Flips: This augmentation mirrors images horizontally with a probability of 0.5 to simulate image reversal and change orientations
- Random Cropping: Images were padded by 4 pixels on either side, before extracting a random crop of 32x32 pixels. This helps the model learn to recognize objects regardless of relative position within an image frame.
- Random Rotation: Images were rotated up to 15°, which introduces rotational variability.
- Random Cropping: Images were padded by 4 pixels on either side, before extracting a random crop of 32x32 pixels. This helps the model learn to recognize objects regardless of relative position within an image frame.
- Color Jitter: This randomly changes the brightness, contrast, saturation, and hue of images.

These steps help to simulate real-world variations in images, thus making the model robust to variations in object orientation, location, and color conditions within the image. This robustness is crucial for the model to be applied to unseen images with various settings and orientations in the real world.

The model uses label smoothing cross-entropy, a variant of standard entropy cross-entropy loss function to prevent overconfidence in image classification(Muller et al, 2019). In this model, instead of setting a one-hot code 0 for the non-target class and 1 for the target class, label smoothing assigns a fraction of the smoothing value to the non-target(0.05) and calculates the log probability of the prediction. The optimizer was using AdamW, Adam optimizer including weight decay to prevent overfitting by adding a penalty on the size of the weights to the loss function. The learning rate and weight decay are set as 0.001 and 0.01 respectively. For the scheduler, the model uses Cosine Annealing Warm Restart, which adjusts the learning rate based on the cosine annealing graph(Liu et al, 2022). The process is periodically reset every 15 epochs to help the model escape local minima and increase the testing accuracy. The training process runs for 50 epochs, and calculates the loss, and updates the model parameters. After calculating the loss, the model used backpropagation to compute the gradients of the loss with respect to the model parameters with the AdamW optimizer to update the weights. The training loss and accuracy are calculated and printed for each epoch.

The pros for this ResNet model is that it can train a deep network efficiently with residual blocks and skip connections for hundreds of layers. The skip connection allows gradients to flow directly through the networks during the backpropagation, which can effectively solve problems such as vanishing gradients. However, such models require captive computational resources, such as NYU HPC for the training process. Finding optimal configurations for layers, blocks, learning rates, and schedulers is very time-consuming and requires extensive experiments. Also, even though the model used several methods to prevent overfitting, there is still a chance, especially for such a deep network.

The process of designing the ResNet model, reveals the importance of skip connections, allowing the gradients to bypass layers during backpropagation. Additionally, during the hyperparameter tuning, it introduces more specific methods such as label smoothing cross-entropy, AdamW, and Cosine Annealing Warm Restart. These methods significantly improve the accuracy and decrease computation time for this designed model.

Result

Loss

The results of the CIFAR-10 classification task exhibit a promising trend. Over the 50 epochs, the training loss decreased steadily from 1.8417 to 0.5475, while the test loss showed a decline similarly, suggesting that the model generalized well to unseen data without significant overfitting. The convergence of test loss with the training loss underscored the ResNet32 model's consistent performance across both train and test datasets.

The test loss also showed a significant decreasing trend from 1.756 at the first epoch to 0.5314 at the 50th epoch.

The parallel descent of both training and test losses indicated the model's ability to learn generalizable features without overfitting. This convergence between the two loss metrics is particularly noteworthy as it signifies the model's consistent performance, which does not waver significantly between the training and validation phases.

Accuracy

For the test accuracy of the images, the decreasing trend from 43.33% at the first epoch to 91.09% in the 50th epoch indicated the model's improving proficiency in correctly predicting the test images as it learned along 50 epochs. The stability of the accuracy curve in the later 30 epochs also maintained a level close to the highest value of 92.68%. This finding is consistent with the efficacy of the model after sufficient training data iterations.

Learning Rate

The learning rate illustrated the application of the cosine annealing schedule, with the scheduler restarted every 10 epochs. The adaptive learning rate strategy is a factor in the model's robustness, preventing it from stalling in local minima and facilitating following learning throughout the training process.

Confusion Matrix

The confusion matrix presents a comprehensive view of the model's performance across the different classes in the CIFAR-10 dataset. The high values along the diagonal indicate a significant number of true positives for each class, which indicates the accurate predictions by the model. For instance, class 0 (plane) showed 869 correct predictions, while class 8 (ships) had 960 correct predictions, showing the model's recognition capabilities for these categories of images.

On the other hand, classes like 3 (cat) and 6 (frog) have 845 and 941 correct predictions respectively, showing a slightly lower but still commendable level of accuracy. These predictions implied that while the model performs well across all classes, there might be certain features in classes 3 and 6 that it occasionally confuses with other classes.

The off-diagonal numbers, while comparatively low, indicate instances of misclassification and provide insight into specific domains where the model may be mistaking one class for another, such as mistaking a cat for class 5 (dog). Addressing these domains could further refine the model's accuracy.

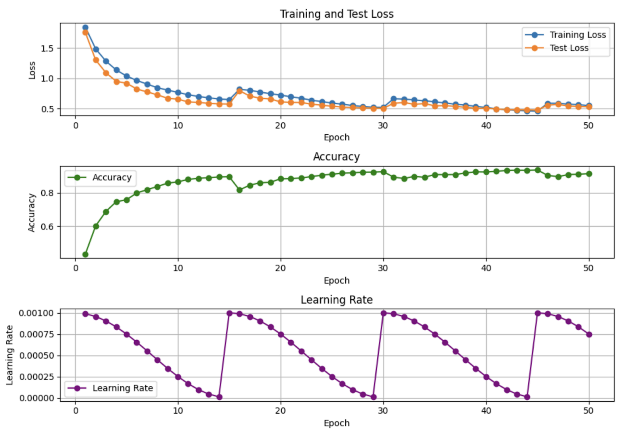

Fig 1. Model Performance Metric for customed design ResNet model, including training and test loss, accuracy, and learning rate.

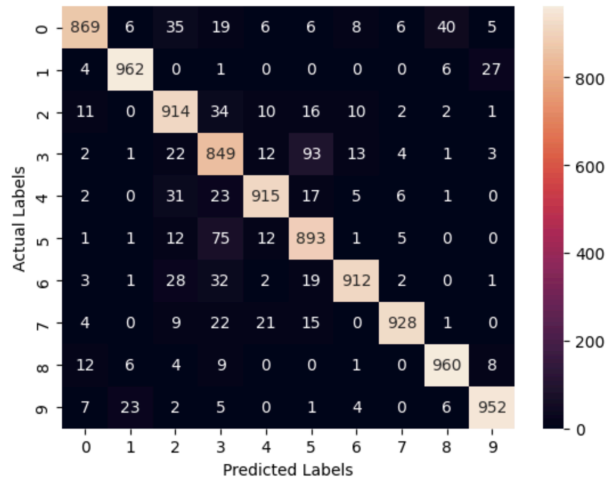

Fig 2. Confusion matrix of prediction for CIFAR-10 dataset prediction from custom-designed ResNet model.

References

Müller, R., Kornblith, S., & Hinton, G. E. (2019). When does label smoothing help?. Advances in neural information processing systems, 32.

Liu, Zhao. "Super Convergence Cosine Annealing with Warm-Up Learning Rate." CAIBDA 2022; 2nd International Conference on Artificial Intelligence, Big Data and Algorithms