Intelligent Visual Signal Understanding (ELEC5304)
Project 3
Deep Convolutional Neural Network for Classification: Training and Performance Analysis

Junkai (Kevin) Mei

May 28, 2024

1 Introduction

Since the invention of computers, humans have been striving to make computers "see" and "understand" visual data like humans. Computer vision can be applied in a very wide range of fields.

In the past, people tried using traditional algorithms like SIFT and SURF. They relied on feature extraction, representation, and classification for categorizing data. However, these methods had many limitations, such as a high dependence on prior knowledge and difficulty in handling global information.

In recent years, with the development of deep learning, the field of computer vision has made remarkable progress, especially in image classification tasks. Convolutional Neural Networks (CNNs) have features like automated feature extraction and end-to-end learning. Among them, the Inception architecture and ResNet are known for their fast training speed and efficient feature extraction. However, some neural networks are very large in size. With the diversification of application scenarios and the continuous increase in data volume, deep neural network (DNN) models that can train quickly and achieve high accuracy have become an important research focus.

In this project, we developed a DNN that addresses the long training times of some neural networks. By optimizing the training process and network structure, we achieved high accuracy and fast classification within a limited time. This project demonstrates the potential of efficient neural network design and optimization techniques. The results provide a deeper understanding of the effectiveness of different network architectures and training strategies for image classification tasks.

2 Method

In this project, our network mainly adopts the Inception-ResNet-V2 architecture. It performs excellently in classifying the CIFAR-10 dataset. Among the classification networks we tested, it achieved a high accuracy of 95.3

The main part of this model is a class called BasicConv2d. Here, we defined the Conv2d convolution layer, the BatchNorm2d layer, and the ReLU layer. The convolution layer is used to extract features from the image. It captures low-level features like edges and textures through a series of convolution operations. BatchNorm reduces internal covariate shift, making the data distribution more stable. Additionally, it allows the use of a larger learning rate, improving model stability and convergence speed, which is crucial for models requiring quick training. Finally, ReLU maintains gradient flow and non-linearity, helping the network learn complex feature patterns.

In our neural network, there are a total of 7 convolutional layers, and they are separated by max-pooling layers after the third and fifth layers. Max-pooling layers downsample the feature maps, reducing their size and making the model more robust. The model is concluded with an adaptive average pooling layer, a dropout layer, and a fully connected layer. The adaptive average pooling layer adjusts the size of the input feature map to a specified size for further data processing. The dropout layer is a regularization technique to prevent overfitting. The fully connected layer typically follows the adaptive average pooling layer and ultimately performs the classification function. This is our DNN model.

Compared to the reference Inception-ResNet-V2 model, we made two modifications. First, we simplified the network structure. We reduced the number of layers in the Inception-ResNet-V2 model. This reduced the computational burden and sped up its convergence while retaining the necessary feature extraction capability. Second, to use the CIFAR-10 dataset, we adjusted the network input size and the number of output classes. This made the classification more efficient.

3 Experiment

To meet our core needs for fast training and precise classification, we designed the network with the following approach:

3.1 Reduced the model complexity

To ensure efficient training, we used the cross-entropy loss function. This is a suitable method for classification tasks, considering that its output is a probability value between 0 and 1. We also used the Adam optimizer. It combines the adaptive gradient algorithm (AdaGrad) and root mean square propagation (RMSProp), which performs excellently in classification tasks.

3.2 Dynamic adjustment of hyperparameters

We set the initial learning rate to 0.003. Additionally, the maximum learning rate of the OneCycleLR scheduler is set to 0.01. The small initial learning rate ensures stability, and with dynamic learning rate adjustments, it allows for rapid model convergence. Under time constraints, this often results in better outcomes.

4 Results and Analysis

4.1 Results

In the final model, the network's accuracy increased from 0.1 to 0.89, and the loss decreased from 2 to 0.1. We also experimented with different network models. Please refer to Figure 2 for the comparison of the results.

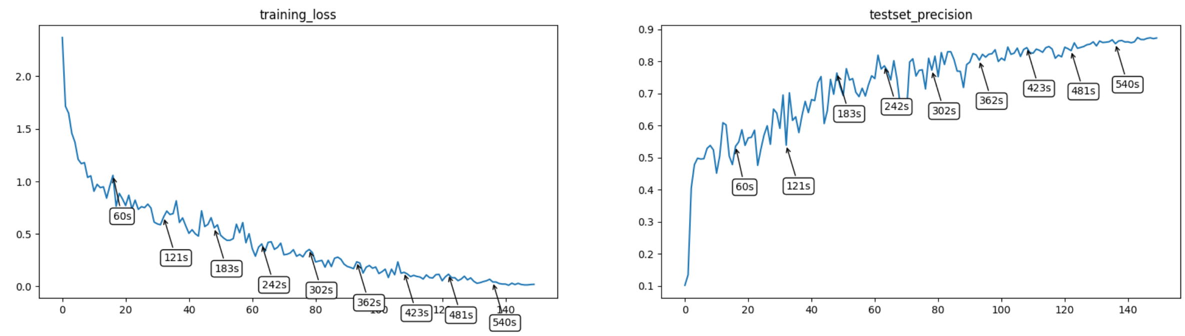

Figure 1: The Loss and Precision of the ImprovedNet

Table 1: Comparison of Different Networks on `CIFAR-10` Dataset

Network	**Initial Loss**	**Final Loss**	**Initial Accuracy**	**Final Accuracy**
ImprovedNet	2.0	0.1	0.1	0.89
Pretrained Model	1.0	1.0	0.1	0.1
Project2 Model	2.5	0.3	0.1	0.74
Simple ResNet	3.0	0.5	0.1	0.69
Traditional CNN	3.6	0.4	0.1	0.71

4.2 Analysis

It can be seen that our final model, ImprovedNet, performed the best. The final loss was 0.1, and the final accuracy was 0.89. Both the loss and accuracy exceeded those of other methods and the original model. In comparison, considering the loss and accuracy of the pre-trained model, it is not suitable for this type of classification task. Our previous image project model, although achieving a final accuracy of 0.74, was not stable enough. Secondly, SimpleResNet's final accuracy was only 0.69, which did not meet the project's requirements. Lastly, for the traditional DNN model, it still did not perform as well as ImprovedNet.

Additionally, the accuracy of ImprevedNet begin to stabilize above 0.8 after 6 minutes. In terms of training composition, its accuracy stability was also excellent. This means that the improved model has advantages in multi-scale feature extraction and residual design. These designs not only enhanced the network's representation ability but also accelerated the training process, ensuring that the network could converge in a shorter time and achieve higher accuracy.

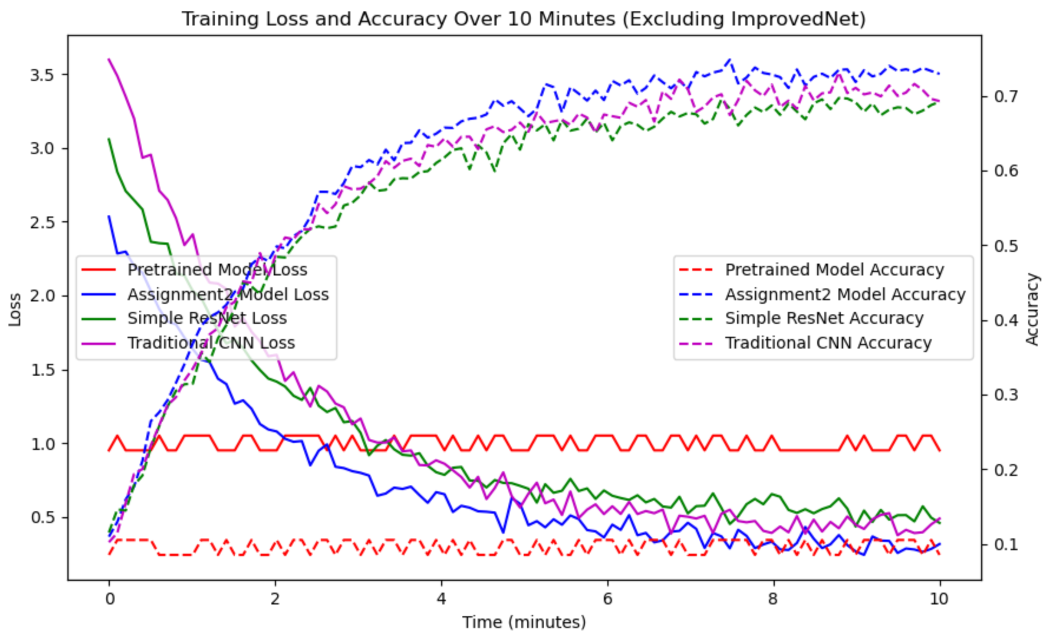

Figure 2: Loss and Precision of Networks

5 Conclusion

Overall, the network designed in this project achieved an accuracy of 0.89 on the CIFAR-10 dataset in a short time, surpassing other models. This balanced both performance and efficiency. In future developments, introducing more advanced data augmentation techniques, such as Generative Adversarial Networks (GANs) or attention mechanisms, can further improve the model's generalization ability and robustness. With these improvements, even better performance is expected in the future.