Realm of Deep Learning

Hu Xiaolong
The University of Adelaide
Adelaide SA 5005
A1912235@adelaide.edu.au

Abstract

This paper analyzes the performance of three different convolutional neural network (CNN) architectures in image classification tasks, including ResNet-18, AlexNet and MobileNet. By training and evaluating each model on the CIFAR-10 dataset, comparing the advantages and disadvantages of lightweight models, classic architectures, and deep networks.

1. Introduction

Convolutional neural networks (CNN) have been widely used in computer vision tasks such as image classification and target detection. In recent years, with the rapid evolution of model architectures, various CNN architectures with significantly different complexity and resource requirements have appeared. In practical applications, a trade-off between accuracy and computational efficiency is required depending on the task requirements and computing resources. Therefore, understanding the performance of different architectures in these aspects is important for model selection.

1.1. Purpose

This paper aims to use three representative CNN architectures for image classification tasks, ResNet-18, AlexNet, and MobileNet. They represent different design concepts of high-performance deep networks, classic standard networks, and lightweight models. ResNet-18 introduces residual connections, allowing the model to learn complex feature representations at a deeper level, thereby achieving higher accuracy in image classification tasks.[1]However, such deep architectures are usually accompanied by high computational costs. AlexNet, as the first deep convolutional network to achieve a breakthrough in the ImageNet competition, laid the foundation for the development of subsequent CNN architectures. Despite its simple structure, AlexNet has a robust performance in classification tasks and moderate computing resource requirements, and is still a classic choice. In contrast, MobileNet reduces the number of model parameters and computational complexity through deep separable convolutions, and performs well in resource-limited environments such as mobile devices and embedded systems.

1.2. Process

This paper will apply ResNet-18, AlexNet, and MobileNet to the CIFAR-10 dataset and evaluate the model performance using metrics such as classification accuracy, loss, and training efficiency. The goal of this experiment is to deeply analyze the performance of each architecture in image classification tasks. Explore the advantages and disadvantages of these architectures when applied to image classification tasks, and summarize their applicable scenarios and potential application value. To provide insights into the trade-offs of different architectures in deep learning applications and provide a reference for further model selection and optimization strategies.

2. Method Description

The following will explain the algorithm and the model used in detail, and provide examples to illustrate their strengths and weaknesses.

2.1. ResNet-18

ResNet-18 is a shallower version of the Residual Neural Network (ResNet) series. It consists of 18 convolutional layers to solve the "vanishing gradient" problem.[1] This problem is alleviated by introducing "residual connections". ResNet-18 consists of multiple residual blocks stacked together, including the following main components:

1. **Input layer:** a 7×7 convolution kernel and a maximum pooling layer to reduce the input size and extract preliminary features.
2. **Residual block:** consists of basic convolution

modules. Each residual block contains two 3×3 convolutional layers, a ReLU activation function, and a batch normalization layer. Multiple residual blocks are stacked in the network to allow information to be passed between multiple layers.

3. **Global average pooling layer:** averages all features after feature extraction, thereby reducing the number of parameters.
4. **Fully connected layer:** a 10-dimensional output for the 10-category classification task of CIFAR-10.

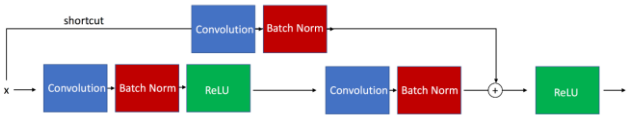

Fig.1. ResNet-18 network architecture

ResNet-18 usually performs well in image classification tasks because it can effectively train deeper layers and reduce the risk of gradient vanishing due to the introduction of residual blocks. However, the ResNet architecture is relatively "heavyweight" and has high requirements for computing resources, which limits its application in mobile devices or resource-constrained environments.

2.2. MobileNet

The core innovation of MobileNet is depthwise separable convolution, which significantly reduces the number of parameters and the amount of calculation.[2] MobileNet divides the convolution into two steps: Depthwise Convolution and Pointwise Convolution, which are used for feature extraction and channel mixing respectively.

The calculation is divided into two steps tasks:

1. **Depthwise Convolution:** Apply a separate convolution kernel to each input channel, the computational complexity is:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F$$

2. **Pointwise Convolution:** Use 1×1 convolution to map M input channels to N output channels, the computational complexity is:

$$M \cdot N \cdot D_F \cdot D_F$$

Depthwise separable convolutions reduce the amount of computation by about:

$$\frac{1}{N} + \frac{1}{D_K^2}$$

This is especially beneficial for small networks. MobileNet is favored for its low computational complexity and number of parameters, and is particularly suitable for use on mobile devices and other scenarios with limited computing resources. However, compared with deeper networks such as ResNet, MobileNet's classification accuracy is generally lower.

2.3. AlexNet

AlexNet extensively uses the ReLU activation function in the convolutional layer, avoiding the vanishing gradient problem that may be caused by traditional activation functions (such as Sigmoid or Tanh).[3]

It consists of 8 learnable layers, including 5 convolutional layers and 3 fully connected layers. Its overall architecture is as follows:

1. Input layer: The input is a fixed-size RGB image, usually 227 × 227 pixels.

2. Convolutional layer: Each convolutional layer uses a different number of convolution kernels to extract features. The activation function uses ReLU (Rectified Linear Unit), which effectively solves the gradient vanishing problem in deep networks.

3. Pooling layer: Use the Max Pooling layer to reduce the dimension of the feature map and reduce the computational complexity.

4. Fully connected layer: The last fully connected layer maps high-level features to specific class labels.

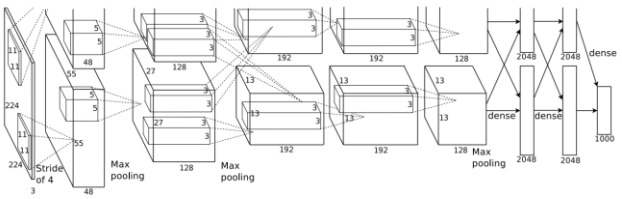

Fig.2. Illustration of the architecture of AlexNet CNN

AlexNet performed well in the image classification tasks set at the time, especially significantly improving classification accuracy in the ImageNet challenge. However, its architecture is still relatively large and has a large number of parameters (about 60 million), which leads to high computing resource consumption and limits its application on resource-limited devices.

3. Experimental Test

The following shows the data preprocessing steps, the experimental setup for testing the models, and the results obtained from different configurations.

3.1. Data Preprocessing

CIFAR-10 is used as dataset for this experimental test, which contains 60,000 32×32 pixel color images divided into 10 categories. To improve the generalization of the model and prevent overfitting, I apply a variety of data

augmentation techniques to the images, including random horizontal flipping and random cropping. This increases the diversity of the training set and enables the model to better learn different features. Since different models have different requirements for the size and distribution of input images, the images of MobieNet and AlexNet are also adjusted to 224×224 pixels to meet the training requirements.

Both ResNet-18,MobieNet,AlexNet were extensively tested with different hyperparameters such as learning rate and optimizer. The following settings were tested:

Learning rate: 0.01 and 0.001
Optimizer: Adam and SGD
Number of epochs: 20 and 30

The models were evaluated based on the loss reduction during training and the accuracy achieved on the test set.

3.2. ResNet-18 Results

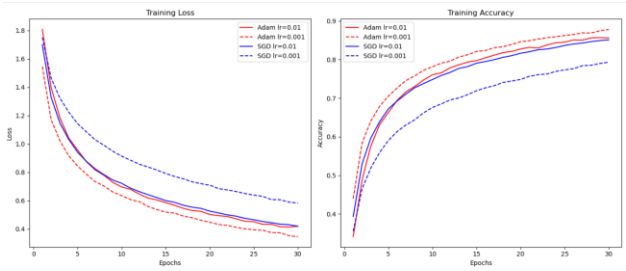

Fig.3. Observe the loss,accuracy curve change of the ResNet-18 model in optimizer Adam, SGD and lr 0.01, 0.001 (epoch 30)

With the Adam optimizer and in learning rate of 0.01, the loss decreased frm 1.8 in the first epoch to 0.4, and the accuracy increased from 34.13% to 85.6%. While in learning rate of 0.001, the final loss is 0.34 and accuracy achieves 87.82%. Although the convergence speed is slow, the model shows a more stable convergence trend and higher accuracy in the later stages of training due to the lower learning rate.

In the meantime, When using the SGD optimizer, the model with a learning rate of 0.01 performs relatively well, with the loss decreasing from 1.7 to 0.4 and the accuracy increasing from 39.36% to 85.12%. And in learning rate of 0.001, the final loss and accuracy are 0.58,79.3%.

Compared with the Adam optimizer, SGD converges faster in the initial stage, but performs relatively smoothly in the later stage. Although the final accuracy is lower than that of Adam, the loss of SGD decreases more smoothly during the entire training process, indicating that it has strong adaptability and can effectively avoid large fluctuations.

3.2.1 Test loss, accuracy Results

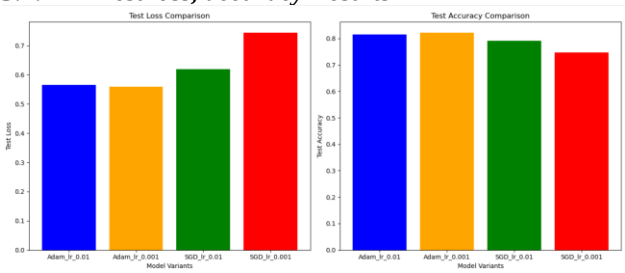

Fig.4. Observe the loss,accuracy curve in validation set of the ResNet-18 model in optimizer Adam, SGD and lr 0.01, 0.001 (epoch 30)

Adam optimizer is significantly better than SGD in test loss and accuracy. Especially when the learning rate is 0.001, Adam achieves the best accuracy and the lowest loss. The adaptive learning rate feature of the Adam optimizer allows it to adapt to different loss terrains faster and reduce instability during the training process.[4] While SGD can achieve good results in some cases, in this experiment it failed to show the same flexibility as expected, especially at lower learning rates.

3.3. Comparative Analysis of MobileNet and AlexNet

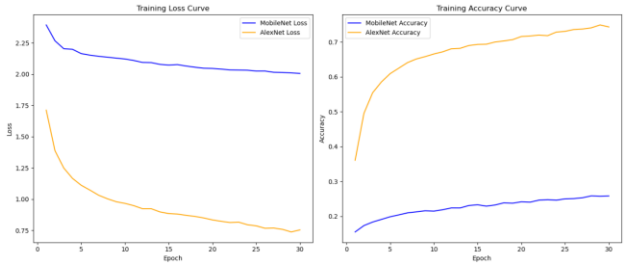

Fig.5. Observe the loss,accuracy curve change of the MobieNet,AlexNet model from 0.01and 0.001 in epoch 30.

It can be seen that MobileNet is not suitable for this dataset. The initial loss is 2.3944. As the training progresses, the loss gradually decreases and eventually reaches 2.0062. While, the loss of AlexNet finally drops to 0.75, showing stronger learning ability and faster convergence speed. Although the loss of MobieNet does not change much, showing the stability of the model, but the large loss may also indicate that there may be problems of overfitting and underfitting.

In terms of accuracy, AlexNet's final 74.34% can match the performance of ResNet-18, but MobileNet's accuracy of 25.81 cannot be compared at all.This also shows that compared with the two models, AlexNet is more effective in feature learning and can better capture the key information in the image.

4. Results Analysis

First, the impact of optimizer on model and parameter adjustment is also important to analysis. Adam combines momentum and adaptive learning rate, which helps accelerate convergence and is suitable for complex loss surfaces.[4] It performs well on each model, especially on ResNet-18, where both training loss and accuracy are the best. This also shows that Adam usually achieves good results for most deep learning tasks, especially when training in small batches.

SGD can help the model jump out of the local optimum, especially on the loss surface with multiple saddle points and local minima. Through randomization, SGD can introduce some noise, which helps to explore a wider parameter space.[5] However, it does not perform as well as Adam in this experiment, especially on ResNet-18, where the test accuracy is significantly lower than the result using Adam. Although SGD is simpler and easier to implement, it may lead to slower convergence in complex models.

Second, the performance of the model. ResNet-18 performed the best in this experiment. Its training loss and test accuracy are significantly higher than other models. MobileNet performs very poorly. Although it is designed to be lightweight and efficient, it may be limited in feature extraction and cannot effectively capture complex information in images. Moreover, lightweight models may be easily overfitted on smaller data sets, resulting in performance degradation

The performance of AlexNet is better than MobileNet, but inferior to ResNet-18. It has a mature architecture and good feature extraction capabilities. Although the training loss decreases rapidly, compared with ResNet-18, the feature extraction capability is still insufficient.

5. Conclusion

This paper provides a comprehensive analysis of the performance of ResNet-18, MobileNet, and AlexNet architectures in image classification tasks, focusing on the impact of different optimization techniques and learning rates. Among the three models, ResNet-18 consistently demonstrated superior accuracy and lower loss compared to both MobileNet and AlexNet, highlighting its effectiveness in handling complex datasets.

The choice of optimizer also plays a crucial role in model training. While SGD provided stability, it was sensitive to learning rates and often resulted in slower convergence. In contrast, the Adam optimizer facilitated faster learning and improved accuracy, particularly at lower learning rates, underscoring the importance of adaptive optimization techniques.

The insights gained from comparing these CNN architectures deepen my understanding of the subtle differences in their performance, strengths, and weaknesses in image classification tasks. These findings are valuable for future research and development in deep learning, and can guide the selection and optimization of models based on specific application needs.

GitHub links: https://github.com/SpiderJockey7/Realm-of-Deep-Learning

References

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770778).https://www.cvfoundation.org/openaccess/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

[2] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*.https://doi.org/10.48550/arXiv.1704.04861.

[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems* (Vol. 25, pp. 1097-1105). https://doi.org/10.1145/3065386

[4] Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, San Diego, CA, USA. arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

[5] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*. https://doi.org/10.48550/arXiv.1609.04747