ECE371 Neural Networks and Deep Learning Assignment 1

Peiwen Luo
School of Electronics and Communication Engineering
Sun Yat-sen University, Shenzhen Campus
luopw5@mail2.sysu.edu.cn

Abstract: In this study, fine - tuning of the ResNet101 pretrained model significantly improved classification accuracy on the dataset_flower dataset. By implementing a comprehensive training strategy in the main function, including data augmentation and optimized learning rate scheduling, the model achieved an accuracy of over 80%.

Keywords: Resnet,Neural Network,Image Recognition,Model,Fine-tuning

1 Introduction

With the increasing complexity of image recognition tasks, the necessity for efficient data organization and model optimization has become increasingly vital. This study investigates the application of state-of-the-art deep learning models and training strategies on a specific dataset to achieve high recognition accuracy. The research is divided into two tasks.

In this study, we focused on fine-tuning a ResNet101 pre-trained network to improve classification accuracy on the dataset_flower dataset. By implementing a comprehensive training strategy in a single main function, we achieved an accuracy of over 80%. This involved meticulous data augmentation techniques, including random cropping, horizontal flipping, and random erasing, to enhance model robustness. We also employed a sophisticated learning rate scheduler that combined linear warm-up and cosine annealing phases to optimize training efficiency. These strategies collectively contributed to the model's improved performance and stability during training.

2 Related Work

In the process of delving into this field, I have read numerous milestone papers that not only demonstrate the evolution of technology but also reveal the development trends in image classification tasks

With the advent of deep learning, Convolutional Neural Networks (CNNs) have gradually supplanted traditional methods and emerged as the predominant technology for image classification tasks. In 2012, the breakthrough achieved by AlexNet, proposed by Krizhevsky et al. [1], in the ImageNet competition marked the beginning of the widespread application of deep learning in image classification. Subsequently, VGGNet, introduced by Simonyan and Zisserman [2], further deepened the network architecture and enhanced classification performance.

Following the advent of AlexNet and VGGNet, CNN architectures have continued to evolve and optimize. In 2015, ResNet, proposed by He et al. [3], introduced residual learning to address the vanishing gradient problem in deep network training, thereby enabling the training of even deeper networks. ResNet not only achieved remarkable performance on the ImageNet dataset but has also been extensively applied to various image classification tasks, including medical image classification, natural scene recognition, and face recognition. Moreover, ResNet has been employed as the backbone network for object detection and image segmentation tasks, where it has further enhanced performance through fine-tuning or combination with other models.

In 2016, researchers from Google introduced InceptionV4 and Inception-ResNet, which further optimized the model architecture. InceptionV4 achieved reduced computational burden while maintaining high accuracy through module optimization and lightweight design [4]. Inception-ResNet, on the other hand, combined the Inception structure with residual learning, thereby further improving model performance [4]. These models demonstrated superior performance compared to their predecessors in benchmark tests such as ImageNet, becoming classic architectures in image classification tasks.

In recent years, novel architectures such as EfficientNet have optimized model depth, width, and resolution through compound scaling, achieving significant reductions in computational cost while maintaining high accuracy [5]. The development of these technologies has not only enhanced the accuracy of image classification but also improved the efficiency and practicality of models.

3 Method

In this study, we selected ResNet101 as the pre-trained model and fine-tuned it to improve the classification accuracy on the dataset_flower dataset. The detailed implementation steps are as follows:

3.1 Data Preprocessing

Firstly, all images in the base dataset were shuffled and integrated. Then, the dataset was split into training and validation sets in a ratio of 8:2, providing an independent dataset to evaluate the model's performance during training and preventing overfitting. This split offers sufficient training data for the model to learn while also retaining enough data for performance assessment. Class documents and validation and test set documents were generated (the code for this part is packaged in the `imagenet` folder and uploaded to GitHub). Data augmentation was performed, with attempts at both random and fixed augmentation, including random cropping, horizontal flipping, and random erasing, making the training results more stable and enhancing the model's robustness. These techniques train the model by simulating various image transformations, making it more stable and accurate when faced with new images. Random and fixed augmentations were attempted to compare the impact of different augmentation strategies on model performance.

3.2 Model Selection and Fine-tuning

Three pre-trained models were selected for comparison: ResNet101, ResNet152, and VGG11, each experimented three times. The number of frozen layers was modified, with each model freezing the first two and first three layers for comparative experiments, allowing us to observe performance changes when the model learns features at different levels.

3.3 Fine-tuning Model Training Process

The Adam optimizer was used due to its adaptive learning rate characteristics. It combines the advantages of momentum and adaptive learning rates, suitable for optimizing non-stationary objectives and sparse gradients. The learning rate change strategy employed a custom combination of warm-up and cosine annealing strategies. This strategy quickly raises the learning rate at the beginning of training and then gradually lowers it later to fine-tune model parameters. This approach helps the model converge quickly in the initial stages of training and avoids divergence caused by excessive learning rates later on. When a smaller learning rate was found to be insufficient to update the model, a gradient accumulation strategy was adopted to simulate larger learning rate updates. This method accumulates gradients from multiple steps and then updates the model, effectively utilizing smaller learning rates without increasing computational cost.

Model	Frozen Layers	Data Augmentation
Resnet101	2	Fixed
	2	Mix
	3	Fixed
	3	Mix
Resnet152	2	Fixed
	2	Mix
	3	Fixed
	3	Mix
VGG11	2	Fixed
	2	Mix
	3	Fixed
	3	Mix

Table 1: Comparison of different pre-trained models with varying frozen layers and their Data Augmentation.

The learning rate schedule is defined as follows:

$$\mathrm{lr(epoch)} = \begin{cases} \frac{lr_{\text{init}}}{lr_{\text{final}}} + \left(1 - \frac{lr_{\text{init}}}{lr_{\text{final}}}\right) \cdot \frac{\text{epoch}}{10}, & \text{if epoch} < 10 \\ \left(\frac{lr_{\text{final}}}{lr_{\text{init}}}\right) \cdot \gamma^{(\text{epoch}-10)}, & \text{if } 10 \leq \text{epoch} \leq 20 \\ \left(\frac{lr_{\text{final}}}{lr_{\text{init}}}\right) \cdot \cos\left(\frac{\pi \cdot (\text{epoch}-20)}{epoch_{\text{tot}}-20}\right) + lr_{\text{final}}, & \text{if epoch} > 20 \end{cases} \tag{1}$$

Through these detailed steps and strategies, we aim to construct a model that can learn quickly and converge stably, while preventing overfitting and improving the model's generalization ability on unseen data.

4 Experiments

The fine-tuning outcomes of the various combinations of pre-trained models, frozen layers, and data augmentation strategies after ten training epochs are depicted in the table below.

Model	Frozen Layers	Data Augmentation	Training Accuracy (%)	Validation Accuracy (%)
Resnet101	2	Fixed	85.00	80.43
	2	Mix	84.50	80.10
	3	Fixed	84.00	80.10
	3	Mix	83.00	79.44
Resnet152	2	Fixed	86.00	80.10
	2	Mix	85.50	79.70
	3	Fixed	83.50	79.47
	3	Mix	82.00	78.50
VGG11	2	Fixed	70.00	65.77
	2	Mix	68.00	64.00
	3	Fixed	65.00	64.34
	3	Mix	63.00	62.00

Table 2: Comparison of different pre-trained models with varying frozen layers, data augmentation strategies, training accuracy, and validation accuracy.

The experimental results indicate that for the ResNet101 and ResNet152 models, the variation in the number of frozen layers has a minimal impact on the recognition accuracy, and the performance under both data augmentation strategies is comparable. However, for the VGG11 model, an increase in the number of frozen layers significantly reduces the recognition accuracy, and the mixed data augmentation strategy does not yield performance improvements compared to the fixed strategy.

These results suggest that for deeper models (such as ResNet101 and ResNet152), the number of frozen layers and data augmentation strategies have a lesser impact on model performance. In contrast, for shallower models (such as VGG11), these factors may significantly affect the model's generalization capabilities. Additionally, the fixed data augmentation strategy may be more conducive to performance enhancement in certain scenarios, especially when the model architecture is shallower.

Epoch	Training Accuracy	Validation Accuracy	train Loss
1	0.7825	0.8049	1.0287
2	0.7988	0.8049	0.5090
⋮	⋮	⋮	⋮
10	0.7988	0.8049	0.5037

Table 3: Training and Validation Accuracy

The training of the ResNet101 model with a mixed learning rate strategy and random augmentation over forty epochs yielded promising results. The model demonstrated a consistent decrease in training loss from 1.0287 to 0.5037, while maintaining a stable validation accuracy of 0.8049 across epochs. This indicates that the model was able to generalize well without exhibiting overfitting.

In conclusion, the ResNet101 model proved to be effective under the given experimental conditions. Future research could explore alternative learning rate schedules and augmentation techniques to potentially enhance model performance further.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[2] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[5] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019.