ECE371 Neural Networks and Deep Learning Assignment 1

Zhiyuan Yuan 22368053
School of Electronics and Communication Engineering
Sun Yat-sen University, Shenzhen Campus
yuanzhy29@mail2.sysu.edu.cn

Abstract: Image classification is crucial in computer vision, enabling object recognition and scene understanding. However, deep neural networks for this task face challenges like vanishing/exploding gradients and degradation as depth increases. ResNet addresses these via residual blocks with shortcut connections, allowing easier training of deep networks. Our work uses ResNet-152, fine-tuned for a 5-class classification task, achieving high accuracy and showcasing ResNet's effectiveness in specialized image classification scenarios.

Keywords: ResNet, Image Classification

1 Introduction

Image classification is a fundamental and critical task in the field of computer vision, with numerous applications such as object recognition, scene understanding, medical image analysis, and more. Over the years, researchers have been dedicated to developing more accurate and efficient image classification models. In recent times, deep learning, particularly Convolutional Neural Networks (CNNs), has achieved remarkable success in image classification tasks and has become the mainstream approach. But traditional deep neural networks face challenges such as vanishing/exploding gradients and degradation problems when increasing network depth, which limit further improvements in model performance.

In this work, we use ResNet [1]. Introduced by Kaiming He [1], ResNet revolutionized deep network training via residual blocks with shortcut connections. These connections enable the network to learn residual mappings, alleviating the vanishing gradient and degradation issues in deep networks. ResNet-152, a deep variant, has achieved remarkable classification accuracy on large - scale datasets like ImageNet, enhancing model generalization and becoming a cornerstone in computer vision.

2 Related Work

In the field of image classification, early research mainly focused on hand - crafted features and shallow models. For instance, traditional machine learning algorithms like SVM or decision trees were used in combination with hand - crafted features such as SIFT and HOG. These methods could achieve decent classification results but were constrained by feature extraction and model expressiveness when dealing with complex scenarios and large - scale datasets. With the rise of deep learning, CNNs have gradually become the mainstream approach for image classification. In 2012, AlexNet [2] made a breakthrough in the ImageNet image classification competition, sparking the widespread application of deep learning in this field. By using multiple layers of convolution and pooling, as well as nonlinear activation functions, AlexNet could automatically learn hierarchical feature representations of images, significantly improving classification accuracy.

Since then, a series of deeper and more optimized CNN architectures have emerged. For example, VGGNet [3] proposed a simple and fixed network structure, enhancing model performance

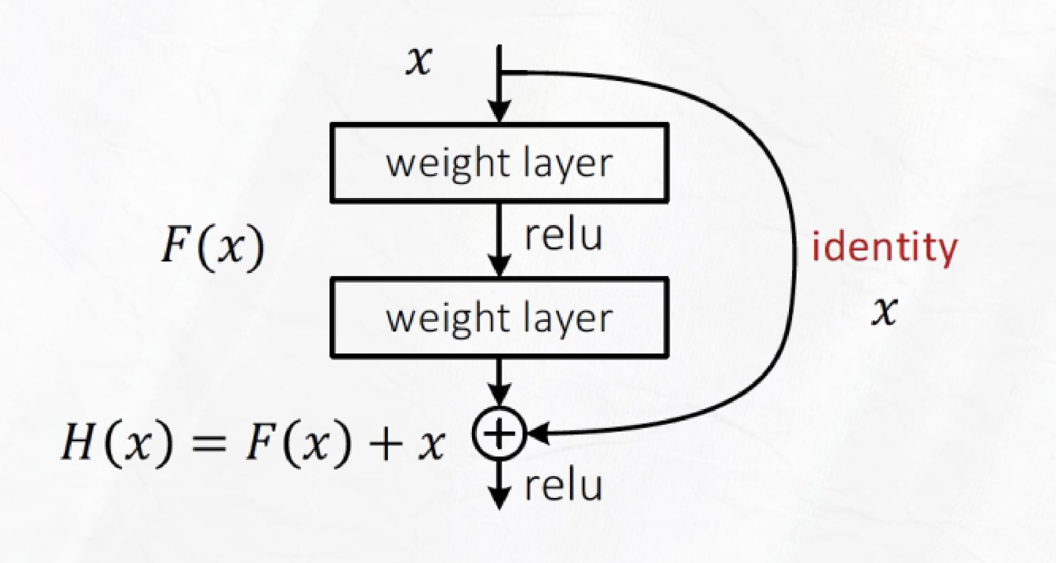

Figure 1: residual block

by increasing network depth; GoogLeNet [4] introduced the Inception module to expand network width without increasing computational cost, boosting feature diversity; and ResNet [1] innovatively brought in residual blocks and shortcut connections, effectively addressing the vanishing gradient problem in deep network training and enabling deeper and more efficient training to further improve classification performance.

In recent years, researchers have been exploring new methods to boost image classification performance. These include using attention mechanisms to guide models towards key image regions for higher accuracy; applying model compression techniques like pruning and quantization to reduce computational and storage costs while maintaining performance for deployment on resource - limited devices; and leveraging pre - trained models for fine - tuning, applying transfer learning to utilize learned generic features for specific tasks and reducing the need for training data and computational resources.

3 Method

3.1 ResNet

ResNet-152 is a deep convolutional neural network that employs residual learning to enable the effective training of extremely deep networks. With 152 layers, it utilizes residual blocks 1 to address the vanishing gradient problem often encountered during the training of deep neural networks. Earlier CNNs faced difficulties in increasing depth due to vanishing or exploding gradient issues. To tackle this, ResNet introduced skip connections that allow the network to learn residual functions, making it easier to optimize deep and structured models.

The network starts with a convolutional layer of 7×7 kernel size and stride 2, extracting initial features from input images. Following this is a batch normalization layer and a ReLU activation function, which accelerate network convergence and introduce non - linearity. Then comes a max - pooling layer with a 3×3 kernel and stride 2, reducing the spatial dimensions of feature maps to cut computation costs while retaining key features and boosting efficiency.

The subsequent layers are organized into four stages, each with multiple residual blocks. Within each stage, convolutional layers with stride 2 halve the feature maps' spatial dimensions and double the number of filters. This design reduces computational costs and gradually extracts higher - level semantic features. Each residual block has two 3×3 convolutional layers and a skip connection.

The connection adds the block's input to its output for residual mapping, helping the network learn residuals between inputs and outputs more easily, thus improving training efficiency and model performance.

At the network's final stage, a global average pooling layer averages the feature maps into single values each. This layer cuts model parameters, lowering overfitting risks and automatically weightedly averaging feature maps for more representative features. The output then goes through a fully connected layer with softmax activation for final class predictions. Softmax converts outputs to probability distributions, suiting classification tasks.

3.2 Fine-tuning strategy

Model Modification We replace the original fully connected layer of ResNet-152, which was designed for the ImageNet dataset with 1,000 classes, with a new one containing 5 output units. This new layer is randomly initialized and will be trained to predict our target classes. Except for this new layer, all the other layers of the network are kept frozen initially. These layers contain pre-trained weights that have proven effective in extracting meaningful features from images and will provide a strong starting point for our task.

Loss Function and Evaluation Metric We employ the standard cross-entropy loss function, which is widely used for classification tasks and provides a clear measure of the difference between the predicted and true class distributions. To evaluate the model's performance, we use accuracy as the primary metric, which is calculated as the percentage of correctly classified samples.

This fine-tuning approach enables us to leverage the powerful feature extraction capabilities of ResNet-152 while adapting it to our specific 5-class classification problem.

4 Experiments

4.1 Training Strategy

Data Preprocessing and Augmentation The input images are resized to 224×224 pixels to match the input size expected by ResNet-152. We also apply data augmentation techniques such as random horizontal flipping, random cropping, and color jittering to increase the diversity of our training data and reduce overfitting. During validation, we use only resizing and center cropping to ensure consistent evaluation.

Training detail We use Adam optimizer with a learning rate of 0.0001. The learning rate is chosen to be relatively small to perform a gentle fine-tuning of the model. We train the model for 50 epochs with a batch size of 64 on a RTX 4090 GPU in 6m 12s. The small learning rate helps prevent the large pre-trained weights from being abruptly changed, which could lead to forgetting the useful features they have learned.

4.2 Datasets

We trained our model on Flower dataset consists of 5 classes, and uniformly sampled training set and validation set with a ratio of 8:2.

4.3 Results

Our fine-tuned model has achieved high accuracy 0.947368 on validation set.

References

[1] 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.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, volume 25, 2012.

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

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.