Dean&Francis

ISSN 2959-6157

Image Classification Based on ResNet Models

Yirong Xu[*]

University of Michigan - Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai, China

*Corresponding author: sheron15@alumni.sjtu.edu.cn

Abstract:

Convolutional neural networks (CNNs) hold significant importance in the image categorization field. The issue of vanishing gradients with increasing dataset size is effectively addressed by ResNet through its residual blocks. This paper focuses on comparing the outcomes of 2 ResNet models using the CIFAR10 dataset. The 60,000 photos in the CIFAR10 dataset are split equally between 10 classes. Preprocessing steps for the data, including normalization and augmentation, are part of the experiment. The cross-entropy loss function is employed for optimization. Results indicate that ResNet18 outperformed ResNet50. For ResNet18, the training loss decreased from 1.8464 to 0.2006, and accuracy increased from 0.3311 to 0.9286 with a test accuracy of 0.8294. In contrast, for ResNet50, training loss went from 5.3736 to 0.4618, and accuracy rose from 0.0989 to 0.8377 with a test accuracy of 0.7604. One possible reason for this outcome is that ResNet50 might be more prone to overfitting due to many more parameters and the CIFAR10 dataset's small size. Additionally, different hyperparameter settings and data augmentation fine-tuning might also contribute.

Keywords: ResNet18; ResNet50; CIFAR10; Convolutional neural network.

1. Introduction

CNNs have become the industry standard for image classification [1]. Nevertheless, the issue of vanishing gradients arises as dataset sizes increase [2]. ResNet addresses the issue of training very deep networks without experiencing the usual deterioration by introducing the notion of residual blocks [3]. The network can learn intricate features and patterns in images more successfully thanks to its special design [4].

Previous research has shown that ResNet variations perform well in a range of classification tasks. Devvi Sarwinda et al. applied ResNet18 and ResNet50 to colorectal cancer classification with different train-test ratios and found that ResNet50 provided better performance in many aspects [5]. Li Ma et al. combined deep convolutional generative adversarial network and ResNet to classify blood cell images with high accuracy [6]. Swalpa Kumar Roy et al. used an attention-based adaptive spectral-spatial kernel ResNet to classify hyperspectral images, outperforming other cutting-edge models [7]. Sadia Showkat and Shaima Qureshi applied five ResNet variants based on transfer learning to recognize COVID-19 pneumonia from chest X-ray images, all with high accuracy [8]. Junlong Cheng applied ResNet structure to group attention blocks to create a new ResNet variant for medical image classification tasks [9].

This paper focuses on ResNet18 and ResNet50's performance on the CIFAR10 dataset. The following sections describe the ResNet18 architecture and principle, discuss its differences from ResNet50, detail the experiment setup, present the results, and offer a discussion on the findings.

2. Method

A potent CNN architecture, ResNet18 has demonstrated outstanding performance in image classification applications. It has a few essential elements that set it apart from conventional CNNs and provide special advantages.

2.1 ResNet18 Architecture and Principle

Mathematically, in a traditional CNN, the goal is to discover a mapping H(x) that connects the input x to the intended result. However, in ResNet18, the network aims to learn a residual function $F(x)=H(x)-x$. The formula for a layer's outcome is then $H(x)=F(x)+x$ [10]. This residual learning formulation addresses the problem of vanishing gradients in deep networks and facilitates the network's learning of the identity mapping when the optimal mapping is close to identity.

The architecture of ResNet18 can be described as follows: The first layer has a 7x7 convolutional kernel and 64 filters. It moves at a step of 2. Following is batch normalization and a ReLU activation function. This initial layer extracts low-level features from the input images. Mathematically, let I be an input image. The output O_1 of this layer can be represented as

$$O_1 = ReLU\left(BatchNorm\left(Conv\left(I\right)\right)\right). \quad (1)$$

Where Conv is the convolution operation, BatchNorm is batch normalization, and ReLU is the rectified linear unit activation function.

Next, there are four stages of residual blocks. Each stage consists of multiple residual blocks with varying filter numbers. For example, the first stage has two residual blocks with 64 filters each. A residual block is composed of two 3x3 convolutional layers with a certain number of filters, batch normalization layers, and ReLU activation functions. The shortcut connection in each residual block adds the input to the output of the block. Let x_n be the input to the nth residual block. The output y_n can be expressed as

$$y_n = F\left(x_n\right)+x_n. \quad (2)$$

Where $F\left(x_n\right)$ is the function computed by the two convolutional layers within the block.

At the end of the network, there is an average pooling layer and a fully connected layer with N output units corresponding to the N classes in the dataset.

Compared to traditional CNNs, ResNet18 offers several benefits. The vanishing gradient issue is resolved with the use of residual connections, enabling the development of extremely deep networks. The use of batch normalization and ReLU activation functions contributes to improved generalization and quicker convergence.

2.2 Differences between ResNet18 and ResNet50

ResNet18 and ResNet50 are both variants of the ResNet architecture but with different depths. ResNet18 has 18 layers and ResNet50 has 50 layers. As the depth increases, the number of residual blocks and filters in each stage also changes. When the number of layers grows large enough, taking ResNet50 as an example, more complex residual blocks with bottleneck structures are used to reduce computational complexity. A bottleneck residual block consists of three convolutional layers with varying filter numbers. The main idea is to use a smaller number of filters in the middle layer to reduce the computational complexity while still maintaining the ability to capture useful features [10]. Mathematically, assume the input to a bottleneck block is x. The block first applies a 1×1 convolution with n filters to reduce the dimensionality of the input. This can be represented as

$$y_1 = Conv_{\{1\times1,n\}}\left(x\right). \quad (3)$$

Where $Conv_{\{1\times1,n\}}$ denotes a 1×1 convolution with n filters.

Then, a 3×3 convolution is applied to the output y_1. Let

$$y_2 = Conv_{\{3\times3,m\}}\left(y_1\right). \quad (4)$$

Where m is the number of filters for this intermediate convolution.

Finally, another 1×1 convolution is used to restore the dimensionality back to the original or a desired output size.

$$y_3 = Conv_{\{1\times1,p\}}\left(y_2\right). \quad (5)$$

Where p is the number of output filters.

The shortcut connection in the bottleneck block adds the input x to the output y_3. So the final block outcome is $H\left(x\right)=y_3+x$. Deeper networks may also need more processing power and longer training durations because to their higher parameter count. However, they also have the potential to capture more complex features and achieve better performance on complex tasks.

3. Experiment

The experiment aims to assess the efficiency of ResNet18 and ResNet50 using the CIFAR10 sample.

3.1 Dataset

In this article the CIFAR10 dataset is used. It has 60000 images uniformly distributed in 10 classes without overlapping. Each image has an equal size of 32*32 pixels and is colorful using RGB representation, thus being stored in a 32*32*3 array. The dataset is separated into a training set with a size of 50000 and a testing set with a size of 10000. In both sets the numbers of images from different classes remain the same [11].

3.2 Data Preprocessing

Normalization: The photos' pixel values are adjusted to have a 0.5 mean and 0.5 standard deviation. This facilitates learning for the network and lessens the effect of various pixel value ranges and scales. Mathematically, let p be a pixel value. The normalized pixel value p_{norm} is given by

$$p_{norm} = \frac{p - mean}{std}. \tag{6}$$

Where mean is 0.5 and std is 0.5.

Data augmentation: The training data is made larger and more varied by using random cropping and horizontal flipping. By selecting a random area of the image, random cropping reduces its size to 32×32 pixels. The image is randomly and with a specific probability flipped horizontally.

3.3 Loss Function

The loss function for optimization in this experiment is cross-entropy loss function. Mathematically, given a set of predicted probabilities $P = \{p_1, p_2, \ldots, p_n\}$ for n classes and the true labels $Y = \{y_1, y_2, \ldots, y_n\}$, the cross-entropy loss L is calculated as

$$L = -\sum_{i=1}^{n} y_i logp_i. \tag{7}$$

4. Results

The outcome data throughout the course of 100 epochs are listed in tables below.

Table 1. Outcome data of ResNet18 on CIFAR10

Epoch	Loss	Accuracy
1	1.8464	0.3311
11	0.7212	0.7497
21	0.5342	0.8123
31	0.4382	0.8449
41	0.3681	0.8695
51	0.3245	0.8841
61	0.2831	0.8986
71	0.2503	0.9108
81	0.2249	0.9194
91	0.2006	0.9286

Table 2. Outcome data of ResNet50 on CIFAR10

Epoch	Loss	Accuracy
1	5.3736	0.0989
11	1.8222	0.2828
21	1.3513	0.5106
31	1.0735	0.6174
41	0.8822	0.6905

51	0.7540	0.7358
61	0.6581	0.7689
71	0.5926	0.7924
81	0.5149	0.8192
91	0.4618	0.8377

As shown in Table 1, for ResNet18 model, the training loss dropped to 0.2006 from a starting value of 1.8464, and the training accuracy improved to 0.9286 from 0.3311. The test accuracy achieved was 0.8294. As shown in Table 2, for ResNet50 model, the training loss dropped to 0.4618 from 5.3736, and the training accuracy grew to 0.8377 from 0.0989. The test accuracy achieved was 0.7604. For clearer observation, the data is plotted as the following figures.

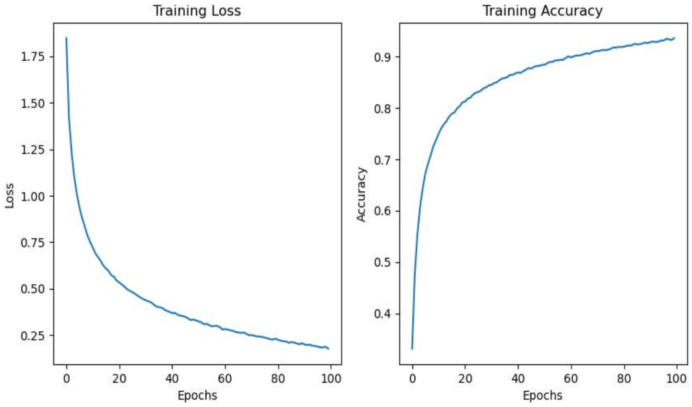

Fig. 1 Outcome plot of ResNer18 on CIFAR10

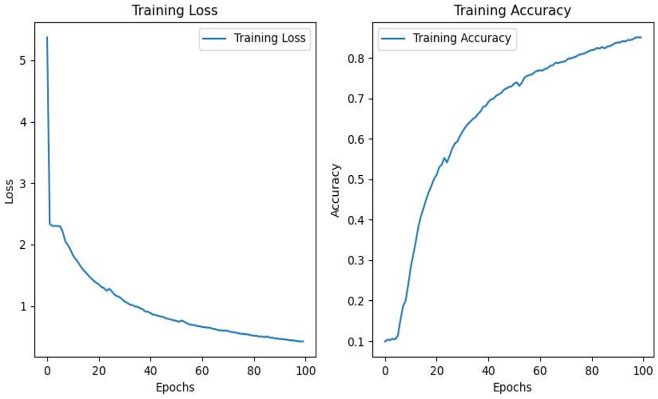

Fig. 2 Outcome plot of ResNer50 on CIFAR10

As shown in Fig. 1 and Fig. 2, both two models demonstrated good performance in classifying the CIFAR10 images. The models were clearly learning as indicated by the steady reducing training loss and rising training accuracy over the epochs. The test accuracy, which is relatively high, indicates that the models have good generalization to new data.

However, ResNet18 outperformed ResNet50 on the CIFAR10 dataset. This is somewhat counterintuitive as ResNet50 is generally expected to have better performance due to its deeper architecture and larger capacity.

One possible reason for this outcome could be overfitting. ResNet50 has a significantly larger number of parameters compared to ResNet18. With the limited size of the CIFAR10 dataset, it is possible that ResNet50 is more prone to overfitting. The additional layers and complexity of ResNet50 may lead to memorizing the training data rather than learning generalizable features.

Another factor could be the optimization process. It's possible that some hyperparameters like learning rate,

ISSN 2959-6157

batch size, and momentum weren't tailored especially for ResNet50. Different models often require different hyperparameter settings to achieve their best performance. It is possible that the current settings are more suitable for ResNet18 and not fully exploiting the potential of ResNet50.

5. Conclusion

In conclusion, this study tested the performance of ResNet models for image classification using the CIFAR10 dataset. The models achieved a high test accuracy, indicating their potential for practical applications in image recognition tasks. Future work could involve further improving the performance of the models by exploring different architectures and techniques. Additionally, investigating the application of the models to other image datasets and real-world scenarios would be valuable to assess its generalization ability and practicality.

However, in this particular experiment, ResNet18 outperformed ResNet50, which is contrary to the general expectation as ResNet50 is typically considered a more powerful model. Future work in this regard could involve further optimization of hyperparameters, exploring different data augmentation techniques, and expanding training dataset to determine if ResNet50 can reach its expected superior performance.

References

[1] Chen L, Li S, Bai Q, et al. Review of image classification algorithms based on convolutional neural networks. Remote Sensing, 2021, 13(22): 4712.

[2] Borawar L, Kaur R. ResNet: Solving vanishing gradient in deep networks//Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022. Singapore: Springer Nature Singapore, 2023: 235-247.

[3] He F, Liu T, Tao D. Why resnet works? residuals generalize. IEEE transactions on neural networks and learning systems, 2020, 31(12): 5349-5362.

[4] Shafiq M, Gu Z. Deep residual learning for image recognition: A survey. Applied Sciences, 2022, 12(18): 8972.

[5] Sarwinda D, Paradisa R H, Bustamam A, et al. Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer. Procedia Computer Science, 2021, 179: 423-431.

[6] Ma L, Shuai R, Ran X, et al. Combining DC-GAN with ResNet for blood cell image classification. Medical & biological engineering & computing, 2020, 58: 1251-1264.

[7] Roy S K, Manna S, Song T, et al. Attention-based adaptive spectral–spatial kernel ResNet for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 2020, 59(9): 7831-7843.

[8] Showkat S, Qureshi S. Efficacy of Transfer Learning-based ResNet models in Chest X-ray image classification for detecting COVID-19 Pneumonia. Chemometrics and Intelligent Laboratory Systems, 2022, 224: 104534.

[9] Cheng J, Tian S, Yu L, et al. ResGANet: Residual group attention network for medical image classification and segmentation. Medical Image Analysis, 2022, 76: 102313.

[10] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[11] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. 2009.