Investigating ResNet on the Perspective of Experimental Results and Analysis

Fenglin Liu
fenglinliu98@pku.edu.cn

Abstract

Deep residual network, e.g., ResNet [3] and PreAct-ResNet [2], has attracted extensive interests from both academic and industrial worlds. Recently, many theoretical analysis have been proposed to verify the effectiveness of deep residual network. In this work, we aim to further demonstrate the effectiveness of deep residual network from the perspective of experimental results and analysis. In particular, we implement the ResNet-18 on CIFAR-10 image classification dataset. The experimental results show that ResNet-18 achieves a performance of **82.03%** accuracy and **79.60%** accuracy of train and test sets, respectively, at the stage of approximately 10,000 training iterations. After about 50,000 iterations, the performance of test set is promoted from **79.60%** to **93.56%** accuracy. Especially, the performance of the training set is **100%** accuracy. These experimental results prove the effectiveness of ResNet.

1 Introduction

There is a surge of research interest in solving the optimization problem of deep neural networks, which presents a considerable number of novel difficulties that are in need of practical solutions. Batch normalization [4], layer normalization [1] and residual learning [3] are widely-used techniques to facilitate the optimization of deep neural networks, which prove to be effective in multiple contexts [7, 8, 2].

In particular, residual learning bypasses the gradient exploding or vanishing problem and tries to solve the model optimization problem from the perspective of information transfer. It enables the delivery and integration of information by adding an identity mapping from the input of the neural network to the output, which may ease the optimization and allow the error signal to pass through the non-linearities.

- To validate the argument and prove the effectiveness of residual learning, we conduct extensive experiments and analysis of ResNet-18 on CIFAR-10 image classification dataset.

2 Experiments

The ResNet-18 is evaluated on a representative tasks, i.e., image classification.

2.1 Baseline and Dataset.

We conduct experiments using the ResNet-18 as proposed by He et al. [3]. CIFAR-10 [5] is a small dataset used for 10 class image classification, which contains 50k images for training and 10k images for testing. The implementation details and hyper-parameters from [3] are mostly adopted. Specifically, the learning rate starts from 0.1, and is divided by 10 at 32k iterations. The batch size is set to 32 and train on single GPU, the weight decay rate is 0.0002, since we do not conduct distributed

Table 1: Results on the image classification task. The lower the better for error rate, the higher the better for accuracy rate.

Dataset	Method	Iterations	Training set		Test set	
			Error (%)	Accuracy (%)	Error (%)	Accuracy (%)
CIFAR-10	ResNet-18	10000	17.97	82.03	20.40	79.60
		20000	12.50	87.50	16.60	83.40
		30000	7.03	92.97	12.17	87.83
		40000	**0.00**	**100.0**	8.94	91.06
		50000	**0.00**	**100.0**	**6.44**	**93.56**

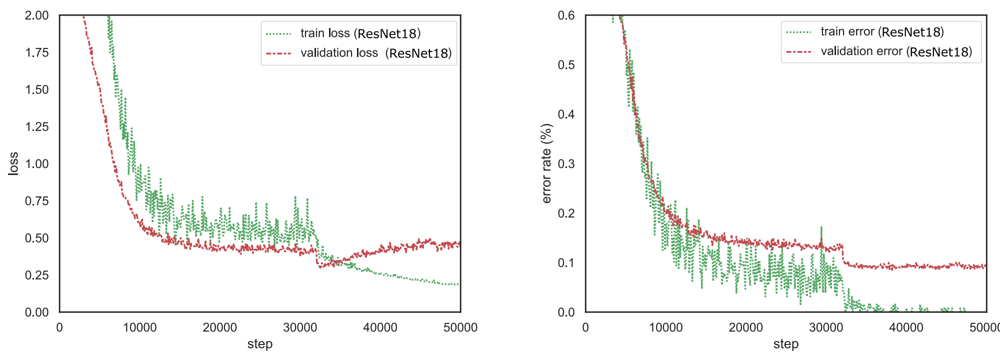

Figure 1: Learning curve for ResNet-18 on CIFAR-10.

training and it results in more stable training. The momentum is 0.9. We use SGD [6] for parameter optimization. Error rate of classification is reported as the evaluation metric.

2.2 Results

The results are reported in Table 1. As we can see, after training 10,000 steps, the ResNet-18 achieves 82.03% and 79.60% accuracy on the training set and test set, respectively. The introduction of longer training (50,000 iterations) will further improve the performance to 93.56% accuracy on the test set, which indicates the number of training steps play an important role in improving the performance of ResNet-18.

3 Analysis

In this section, we give the learning curve and the visualization of filters and feature maps to demonstrate the strength of ResNet-18 intuitively.

3.1 Learning Curve

Figure 1 shows the learning curve of ResNet-18. As we can see, ResNet-18 achieves low training loss after 40,000 training steps. It is worth noticing that the accuracy of validation set achieves significant improvements at the 32k iterations, where the learning rate is divided by 10. However, the validation loss begins to become larger, which indicates the ResNet-18 suffers a slight over-fitting error.

3.2 Visualizing Filters

In this section, we devoting on visualizing the filters to understand the patterns that first convolution layer (conv1) extracts from the input image. As shown in Figure 2, we can learn that the convolution layer learns a collection of filters such that their inputs can be expressed as a combination of these

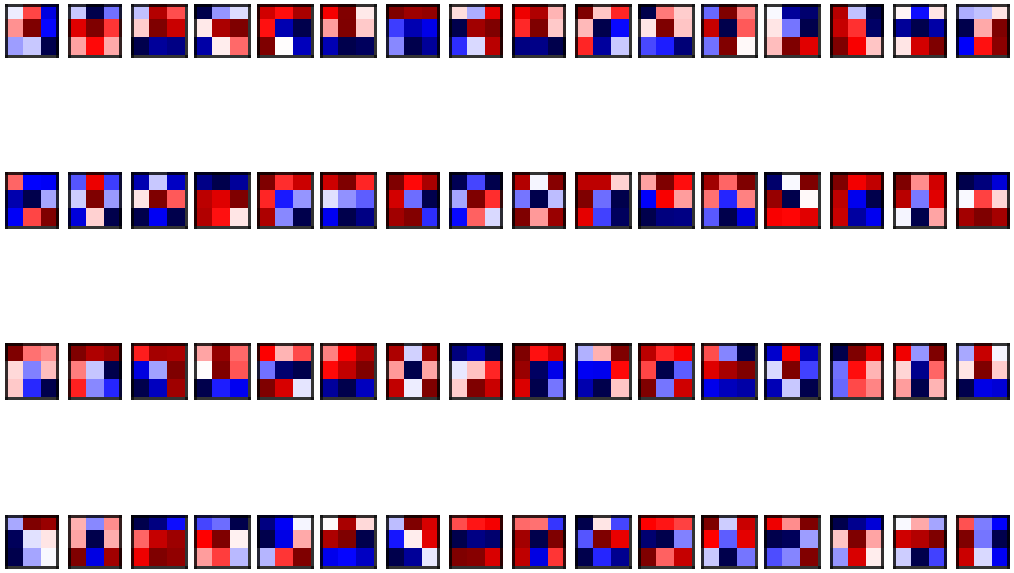

Figure 2: The visualization of filter (Conv1).

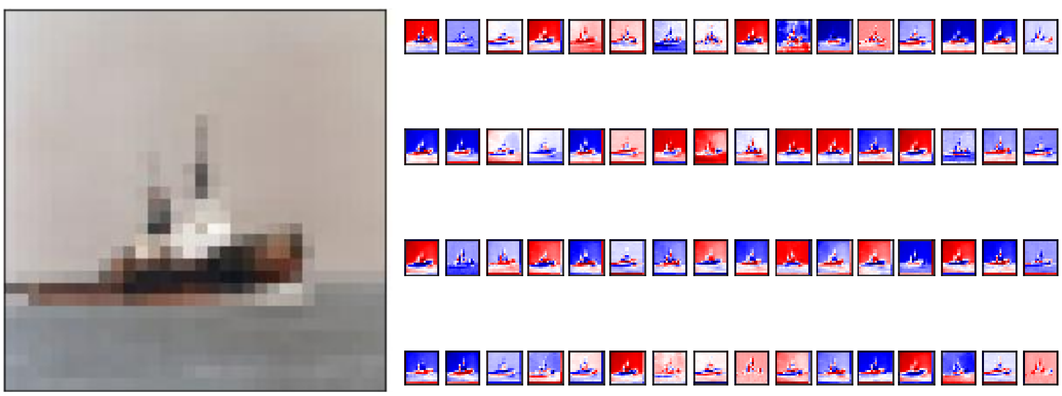

Figure 3: Left: The original input image; Right: The visualization of feature maps (Conv1).

filters. In particular, the filters of initial layer (conv1) look like a collection of simple directional edges and colors (or colored edges in some cases), which shows the same phenomenon as in Sec. 3.3.

3.3 Visualizing Feature Maps

In this section, given an input image, we will visualize the output of the convolution operation (conv1 and conv5_x). For example, in our ResNet-18, the input layer dimension is 32x32x3 and the output dimension after the first convolution operation is 32x32x64. Here, 64 is the number of filters which are used to extract input features after 1st convolution operation, so we will just plot these sixty-four 32x32 outputs (see Figure 3).

Interpretations

- The initial layers (e.g., conv1) retain most of the input image features, which shows the convolution filters are activated at every part of the image. This indicates that these filters in initial layers might be primitive edge detectors.

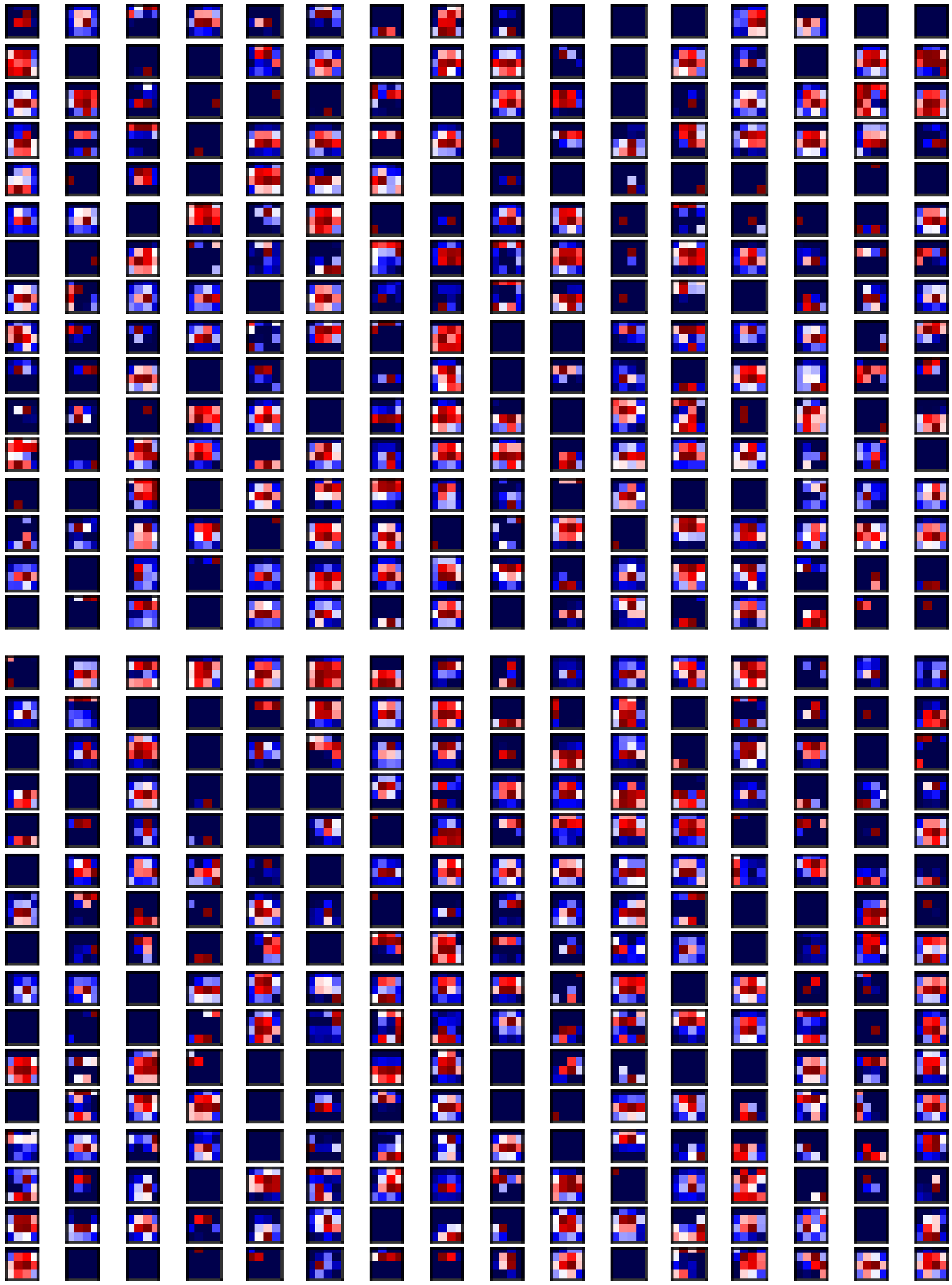

Figure 4: The visualization of feature maps (Conv5).

- As we go deeper in ResNet, the feature maps become visually less interpretable. Especially in conv5, as shown in Figure 4, we see a lot of blank convolution outputs, which means that the pattern encoded by the filters were not found in the input image.

In short, we find that the neural network can act as an information distillation pipeline, in which the input image is converted into a visually inexplicable domain by removing irrelevant information, but it can be helpful for making a selection from the output classes in the last softmax layer.

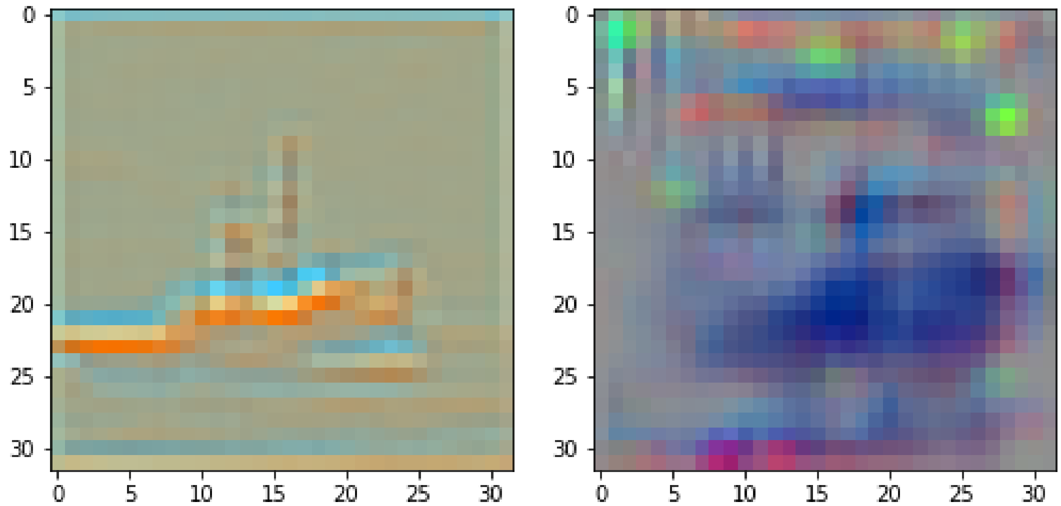

Figure 5: The left plot and right plot illustrate the reconstructed image from conv1 and conv5, respectively.

3.4 Visualizing Using Deconvnet

In this section, we adopt the Deconvolutional Network (deconvnet) [9] to project the feature maps back to the input image space. The deconvnet can be thought of as a convolutional network that uses the same filtering and pooling, but in reverse, so instead of mapping image to features, deconvnet projects the feature activations (i.e., the outputs of convolution layer) back to the input image space,

The left plot of Figure 5 shows conv1 responds to corners and other edges/color conjunctions. The right plot of Figure 5 shows the entire localized ship.

4 Feedback

In this section, we will provide some feedback to improve the quality of AI course in the future.

Time. To accomplish the assignment, I have spent countless nights, maybe 100+ hours in total.

Comment on AI Course. I am very happy to attend this course, which make me enjoy a detailed understanding of artificial intelligence, on the application level. However, I think the course can try to explain the AI itself, from the mathematical level and the theoretical level. In addition, the course could also explain the traditional AI and logic.

Comment on the Assignment. This assignment gives me a more intuitive understanding of CNN, so that the understanding of CNN is no longer at the formula level. I think it will be helpful for my development in the future.

Suggestions for Following Lectures. I think the following lectures can try to introduce the meta / federated / unsupervised / transfer learning, etc. I hope that after completed this course, I have a global and deep understanding of AI in my head, not limited to some specific sub-areas.

5 Conclusions

In this work, we aim to verify the effectiveness of ResNet-18 from the perspective of experimental results. Experiments on CIFAR-10 image classification dataset proves the viability and effectiveness of ResNet-18. The experimental results show that giving more training iterations, the ResNet-18 is able to achieve better performance in terms of error rate, which demonstrates the potential of giving larger training iterations in residual learning. The extensive visualizations illustrate what the ResNet has learned to recognize in the image, which gives insights into the interpretability of ResNet, thus we can confidently deploy ResNet in the future academic researches and real-world applications.

Acknowledgement

Thanks to Torchvision for providing the implementation of ResNet, PyTorch team for providing PyTorch 1.3.0, CIFAR team for providing dataset, Conviz for providing the example of visualizing weights and output of convolutional layers.

References

[1] L. J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[4] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[5] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Computer Science Department, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/cifar.html`.

[6] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3): 400–407, 09 1951.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.

[9] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, 2011.