Proceedings of CONF-SEML 2025 Symposium: Machine Learning Theory and Applications
DOI: 10.54254/2755-2721/154/2025.TJ23152

A Comparative Study of ResNet Depth and Training Strategies

Yinuo Sheng

School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
122090461@link.cuhk.edu.cn

Abstract: Deep Convolutional Neural Networks (DCNNs) have become the cornerstone of image classification tasks. Among them, Residual Network (ResNet), as a landmark discovery, occupies an irreplaceable position in both academic research and practical applications. Although many excellent architectures based on ResNet have emerged, the techniques accumulated through modern practices can also significantly improve model performance without changing the ResNet model structure itself. This study investigates the impact of network depth and training strategies on ResNet performance, comparing the performance of ResNet-18 and ResNet-34 on ImageNet subsets and CIFAR-10. It also evaluates the impact of image augmentation and different learning rate scheduling strategies on training outcomes. The experimental results show that deeper ResNet models achieve superior performance, although careful adjustments are needed to mitigate overfitting, especially on smaller datasets. In addition, combining image augmentation can significantly improve the generalization ability of the model. Among the learning rate schedulers tested, Cosine Annealing provided the most stable training process and yielded the best final performance. This work concludes that while increasing model depth can improve representation capacity, employing effective training techniques such as data augmentation and appropriate learning rate scheduling is crucial for optimizing model performance without altering the network architecture itself, highlighting their practical importance in computer vision tasks.

Keywords: ResNet, Image augmentation, Learning rate scheduler

1. Introduction

Image classification is one of the fundamental tasks in computer vision, encompassing image preprocessing, feature extraction, and classification using classifiers. Deep Convolutional Neural Networks (CNNs) have become dominant in computer vision fields related to image classification due to their use of convolutional neural networks to automatically extract features from input images. They can effectively learn and express features from a large number of samples and possess stronger generalization capabilities. Technical breakthroughs in the field of deep convolutional neural networks have been continuously occurring in recent years. As early as the birth of AlexNet in 2012, the academic community recognized that the depth of the model is crucial for model performance [1]. However, before Residual Neural Network (ResNet), training a deep model was not only computationally challenging but also faced problems during training, such as gradient vanishing/exploding and decreased training accuracy due to training errors.

DOI: 10.54254/2755-2721/154/2025.TJ23152

ResNet was initially proposed by Kaiming He and others in 2015, achieving a 3.57% error rate on the ImageNet test set and winning 1st place in the ILSVRC 2015 classification task. It also achieved first place in the COCO object detection dataset with a 28% relative improvement [2]. Compared to traditional deep convolutional neural network computer vision algorithms, ResNet significantly increased depth while reducing model complexity and made model performance positively correlated with model depth. Simultaneously, various techniques used and summarized in CV task instances have gradually been applied, including minor modifications to the model architecture, data preprocessing, loss function, and learning rate schedule [3]. Methods like these can still bring significant improvements to overall performance without changing the model structure. According to Tong He et al., ResNet-50 with appropriate tricks even outperforms DenseNet with the same FLOPs [3, 4].

This paper will use ResNet models of different depths for image classification and introduce tricks that do not change the model structure to improve overall performance. First, this paper will compare the classification performance of ResNet-18 and ResNet-34 models on an ImageNet sub-dataset. Then, image augmentation techniques and a more suitable learning rate scheduling strategy will be introduced to improve the training accuracy of the model.

2. Experimental methods

2.1. Dataset

The dataset selected for evaluating the performance of ResNet-18 and ResNet-34 is a subset of ImageNet, including 120 classes of different breeds of dogs [5]. The image sizes vary, with approximately 10k images used for training and testing. Images in the ImageNet dataset are organized according to the hierarchical structure of WordNet. Each node (synset) represents a concept and is represented by multiple images.

The dataset selected for testing image augmentation and other technical tricks is from CIFAR-10 [6]. CIFAR-10 contains 60,000 32x32 pixel color images. These images are divided into 10 categories, with 6,000 images per category. Compared to the selected ImageNet subset, this dataset has smaller images, making it more suitable for higher computational complexity. Due to the larger data scale, it is more suitable for training the model from scratch instead of using pre-trained models.

2.2. Basic principles

2.2.1. Model introduction

The ResNet model was used in both sets of experiments in this study: testing the performance of models with different depths and applying tricks to improve model performance. The structure of ResNet is roughly the same as traditional deep convolutional neural networks, following a block-based design, with differences from previous generation models in the model blocks.

Assume the original input is x, and the desired ideal mapping that the model should learn is $f(x)$. In the residual block part, as shown in Figure 1, based on the VGG $3 * 3$ convolutional layer design, ResNet transforms the original ideal mapping of input data into a residual mapping, i.e., from $f(x)\ to\ f(x) - x$. Therefore, when obtaining the residual mapping, it is necessary to perform an addition operation with the input data through a shortcut connection. Since the residual mapping is added to the input data after it is obtained, when it is necessary to change the number of channels of the data during the learning process, a $1 * 1$ convolutional layer needs to be added to the cross-layer data path to make the shape of the input data consistent with the output data. At the same time, a batch normalization layer is added after each convolutional layer [2, 7]. The final residual block structure is shown in Figure 1:

DOI: 10.54254/2755-2721/154/2025.TJ23152

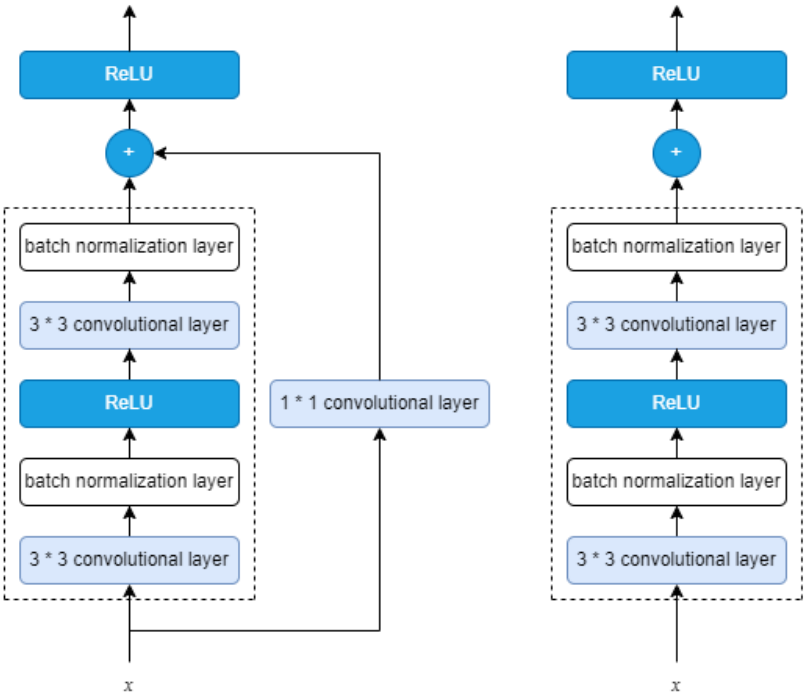

Figure 1: ResNet block with and without 1*1 convolutional layer [2, 7]

This construction form enables the model to achieve good gradient preservation during the training process. A simple mathematical idea is provided here: Assume $g(x)$ is the original network, and $f(x)$ is the newly added layer. Then in the process of model training, the solution to the gradient is essentially a multiplicative relationship (chain rule):

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial g(x)} * \frac{\partial g(x)}{\partial x} \tag{1}$$

Therefore, the gradient becomes extremely small when the number of layers increases, i.e., the gradient vanishing problem. The construction of the ResNet block makes the gradient calculation of the model become the following form:

$$\frac{\partial f(g(x))+g(x)}{\partial x} = \frac{\partial f(g(x))}{\partial g(x)} * \frac{\partial g(x)}{\partial x} + \frac{\partial g(x)}{\partial x} \tag{2}$$

Therefore, the gradient can be well preserved [8].

The ResNet-18 and ResNet-34 model architectures used in this study are similar. First, there is a $7 * 7$ convolutional layer with an input channel number of 64 and a stride of 2, followed by a $3 * 3$ max pool layer with a stride of 2, and then different numbers and channel numbers of residual blocks are added, and finally a fully connected layer is added. The main difference between ResNet-18 and ResNet-34 is that the number of residual blocks leads to different total layers. The former has 18 layers, and the latter has 34 layers [2].

2.2.2. Introduction to tricks

Image augmentation is a technique that expands the size of the training set by generating similar but different training samples after performing a series of random transformations on the training images. At the same time, randomly changing the training samples can reduce the model's over-reliance on certain features, thereby effectively improving the model's generalization ability [4, 7]. The image

DOI: 10.54254/2755-2721/154/2025.TJ23152

augmentation techniques used in this study include randomly cropping at different positions in the training image, randomly flipping the image in the horizontal and vertical directions, and adjusting the image's attributes, such as adjusting brightness, hue, and saturation. Image augmentation techniques are now widely used in the field of computer vision.

In the model training process, a single learning rate sometimes cannot achieve good final performance, so a learning rate scheduler is needed to implement different scheduling strategies. The aspects that need to be focused on include the size of the learning rate, the decay rate, and initialization. This study adopted different strategies for the decay rate of the learning rate to seek better results applicable to the ResNet model [7, 9].

2.3. Experimental design

2.3.1. ResNet depth experiment

First, batch process the test data, setting the batch size to 128 and using 10% of the data as valid cases. Then, perform image augmentation processing on the test data: randomly crop images to a size where the height and width are 0.64 to 1 times of the original image, with an aspect ratio between 3/4 and 4/3. Crop a 224x224 size image from the center of the image, randomly change brightness, contrast, and saturation, add random noise, and normalize each channel of the image.

After preprocessing the test data, fine-tune the ResNet18 and ResNet34 models pre-trained on ImageNet: use a small custom output network to replace the output layer of the original models. Define the training function to use the SGD (Stochastic Gradient Descent) method and select the stepLR (step learning rate scheduler) strategy in the learning rate scheduler. Finally, fine-tune the hyperparameters of the custom model and train and validate the model.

2.3.2. Image augmentation and learning rate scheduler

First, batch process the test data, setting the batch size to 128 and using 10% of the data as valid cases. Then, choose whether to perform image augmentation processing on the test data: enlarge the image to a 40-pixel square, and then randomly crop a small square with a height and width of 0.64 to 1 times of the original image, and scale it to a square with a height and width of 32 pixels. Also, randomly flip horizontally, add random noise, and normalize each channel of the image.

Use an untrained ResNet18 model to train the processed data, define the training function to use the SGD method, and select three strategies in the learning rate scheduler: constant learning rate, stepLR, and cosine annealing LR. Finally, adjust the hyperparameters and perform model training and validation.

3. Experimental results

3.1. ResNet depth experiment

This group of experiments adopted a smaller learning rate because it was based on fine-tuning pre-trained models. To achieve ideal results, a relatively smaller learning rate and a larger number of epochs were used for training the ResNet-34 custom model. When the ResNet-34 group used the same number of epochs and learning rate as the ResNet-18 group, the training image showed a clear overfitting trend. After only fine-tuning the parameters to eliminate overfitting, as Table 1 shows, ResNet-34 showed a clear advantage.

DOI: 10.54254/2755-2721/154/2025.TJ23152

Table 1: Results using Kaggle competition scores[9]

Model	Score
ResNet-18	0.8018
ResNet-34	**1.3327**

3.2. Image augmentation and learning rate scheduler

When the group that did not implement image augmentation used the hyperparameters when the group that implemented image augmentation achieved ideal results, its training image showed a clear overfitting trend: the training accuracy quickly reached 1, the training error quickly dropped to 0, and the valid accuracy quickly converged but the performance was not ideal. The group that implemented image augmentation significantly outperformed the group that did not implement image augmentation in terms of final performance (Table 2).

Table 2: Results using Kaggle competition scores with different learning rate schedular [10, 11]

	Constant	Step	Cosine Annealing
Original	0.8294	0.8372	0.8387
Image augmentation	**0.8814**	**0.8827**	**0.8848**

The groups using three learning rate scheduling strategies showed different performances during training. Among them, the group applying a constant learning rate was the most unstable in fitting, and the group applying the Cosine Annealing strategy was the smoothest in fitting. From the final performance, the group with a constant learning rate performed the worst, and the Cosine Annealing group performed the best (Figure 2).

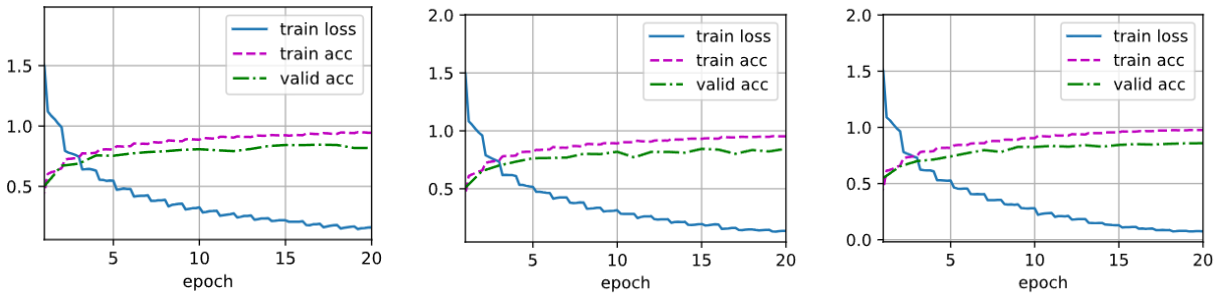

Figure 2: Training process of using different learning rate schedular. (a) Constant; (b) StepLR; (c) Cosine Annealing LR (photo/picture credit: original)

4. Discussion

This study believes that because ResNet-34 has a larger capacity and deeper layers, the gradient experiences more complex accumulation during backpropagation, so it is more prone to overfitting when the training data is small. At the same time, the results show that ResNet-34 does have stronger expressive power than ResNet-18.

In terms of image augmentation, although the group applying image augmentation did not substantially increase the number of training sets, related random images were generated each time the trainer accessed the training data. Therefore, conceptually, the training data was indeed increased, so the limited training data could be fully utilized, and the trend of overfitting could be reduced. Notably, due to the nature of ResNet's residual mapping, although the group without image

DOI: 10.54254/2755-2721/154/2025.TJ23152

augmentation exhibited overfitting, it did not significantly negatively affect the results. This is consistent with the results obtained in the original experiments using very deep ResNets [2]. At the same time, a suitable learning rate scheduling strategy is also very important for the model training process.

5. Conclusion

This study aims to verify the correlation between ResNet depth and performance and to improve model performance through training strategies that do not change the model structure. The experimental results show that when using ResNet models of different depths to classify the same dataset, deeper models have better performance, which is consistent with the original research results. At the same time, image augmentation and appropriate learning rate scheduling strategies can indeed bring significant progress to the model training process and final performance. Compared with optimizing the model structure, the difficulty of choosing appropriate training techniques in practical applications is much smaller. Therefore, although the experimental method of this study is not very novel, it still emphasizes the importance of model structure optimization and technical means for the final performance.

The limitations of this study are also not to be ignored. Due to computing power limitations, this study did not use models with greater depths, such as ResNet50, ResNet101, and ResNet152. Also, because the scale of the dataset used was relatively small, using deeper models may be more affected by problems such as overfitting. In terms of technical tricks, this study only focused on data augmentation and learning rate scheduling strategies for image classification. Based on these limitations, future research on this topic can test deeper and more advanced models, conduct practices in other sub-fields of computer vision such as semantic segmentation, and try to adjust and optimize training performance from more aspects, including learning rate scheduling.

References

[1] *Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.*

[2] *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. 770-778).*

[3] *He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., & Li, M. (2019). Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 558-567).*

[4] *Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A comprehensive survey of image augmentation techniques for deep learning. Pattern Recognition, 137, 109347.*

[5] *Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.*

[6] *Abouelnaga, Y., Ali, O. S., Rady, H., & Moustafa, M. (2016, December). Cifar-10: Knn-based ensemble of classifiers. In 2016 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 1192-1195). IEEE*

[7] *Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). Dive into deep learning. Cambridge University Press.*

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

[9] *Smith, S. L., Kindermans, P. J., Ying, C., & Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489.*

[10] *Kaggle. (2015). Playground Prediction Competition. CIFAR-10 - Object Recognition in Images. Kaggle.com, www.kaggle.com/c/cifar-10.*

[11] *Kaggle. (2018). Dog Breed Identification. www.kaggle.com/c/dog-breed-identification.*