Image Classification Based on ResNet and SeNet

Authors: Translate to English

Date: 2024-01-07T00:00:00+00:00

Abstract

Image classification and recognition are of significant importance in modern society. Numerous excellent convolutional neural network works have been developed to optimize the accuracy of image classification, among which a prominent representative is ResNet 1, which substantially increases the depth of neural networks, thereby greatly enhancing their performance. Meanwhile, there are also pluggable performance optimization submodules that can help improve all networks, with one outstanding representative being SeNet 3. However, they do not always perform well when faced with complex real-world scenarios. The main work of this paper is to study how to effectively improve the recognition performance of convolutional neural networks (ResNet) in some special scenarios (small images, high-noise images), and to attempt to analyze some underlying mechanisms of neural networks.

Full Text

Image Classification Research Based on ResNet and SeNet

Dai Ying[1]
[1]Beijing University of Chemical Technology, Beijing 100029, China

Abstract

Image classification and recognition are of great significance in modern society. Numerous excellent convolutional neural network architectures have been developed to optimize classification accuracy, among which ResNet[1] stands out as a prominent representative that substantially increases network depth and thereby greatly improves performance. Meanwhile, various pluggable performance optimization sub-modules can enhance any network architecture, with SeNet[3] being another outstanding representative. However, these models do not always perform well when faced with complex real-world scenarios. The main contribution of this paper is to investigate how to effectively improve the recognition performance of convolutional neural networks (specifically ResNet)

in special scenarios such as small images and high-noise images, and to analyze the underlying mechanisms of these networks.

Keywords: Image classification, convolutional neural networks

1. Introduction

Convolutional Neural Networks (CNNs) are among the most commonly used models in deep learning and have achieved tremendous success in computer vision, speech recognition, and other domains. This paper briefly reviews both classic and state-of-the-art CNN models, from AlexNet to ResNet and SeNet.

AlexNet[5] represents a milestone in CNN development, winning first place in the 2012 ImageNet Large Scale Visual Recognition Challenge. Subsequent improvements based on AlexNet include VGG[6], GoogLeNet[7], and particularly ResNet[1]. ResNet is a deeper CNN architecture that addresses the vanishing gradient problem through residual connections, achieving exceptional results in ImageNet competitions. ResNet has become a fundamental building block for many state-of-the-art CNN models, demonstrating the power of deep learning in computer vision. SENet[3], or Squeeze-and-Excitation Network, is a more recent convolutional neural network architecture that has achieved outstanding results on ImageNet classification tasks. SENet introduces a novel module called the Squeeze-and-Excitation block, which explicitly models interdependencies between channels to adaptively recalibrate channel-wise feature responses. This technique allows the network to assign greater weight to informative channels while suppressing less useful ones, thereby improving both accuracy and efficiency. SENet has become another important building block for modern CNNs, showcasing the continuous evolution of deep learning techniques.

Beyond these models, numerous other CNN architectures such as DenseNet[4], MobileNet[2], and EfficientNet[8] have demonstrated excellent performance in various application scenarios. However, for complex and variable real-world situations, these outstanding algorithms still cannot excel in all scenarios. The primary objective of this paper is to investigate network adjustments for specific scenarios (small images and high noise), attempting to analyze general methods for these scenarios and the underlying mechanisms of the relevant networks.

2. Related Work

2.1 ResNet

ResNet is a deep neural network architecture proposed in 2015. Its full name is "Residual Network," designed to solve the vanishing gradient problem in deep neural networks by introducing residual blocks. In traditional neural networks, each layer transforms the input and outputs a new feature representation. When networks become very deep, these transformations cause the input signal to gradually disappear, resulting in the vanishing gradient problem. To address this, ResNet introduces residual blocks, each containing two branches: a main branch

and a skip connection branch. The main branch performs a series of transformations on the input and adds the result to the output of the skip connection branch to obtain the final output of the residual block. This design makes it easier for the network to learn identity mappings where input equals output. If no transformation occurs in a residual block, the input can be directly passed to the output through the skip connection, thereby avoiding information loss and vanishing gradients. This design enables ResNet to train very deep neural networks and achieve excellent performance on several computer vision tasks, including image classification, object detection, and semantic segmentation.

2.2 SeNet

A common issue in most CNN-based image processing research is the neglect of inter-channel relationships. To address this problem, SeNet proposes the Squeeze-and-Excitation (SE) module, which strengthens channel interdependencies by learning weights for each channel, thereby improving model expressiveness. The SE module comprises two operations: squeeze and excitation. The squeeze operation compresses each channel's feature map into a single value through global average pooling to obtain channel weights. The excitation operation uses a Multi-Layer Perceptron (MLP) to combine each channel's feature map and weight them according to their importance. Specifically, the squeeze operation first applies global average pooling to each channel of the input feature map, then processes each channel's resulting value through a fully connected layer and ReLU activation function to obtain channel weights. The excitation operation uses an MLP to weight and combine each channel's feature map, producing an output feature map with enhanced channel interdependencies. The MLP's input consists of the channel weights obtained from the squeeze step, and its output is a weight vector of the same size as the input feature map. The SE module can be embedded into other network structures by inserting it into each module of the convolutional neural network, enabling the model to adaptively learn channel relationships while learning feature representations.

Figure 1 [Figure 1: see original paper] illustrates the typical SeNet structure with embedded SE modules. The SE module has demonstrated strong performance across various computer vision tasks, including image classification, object detection, and semantic segmentation. In the ImageNet image classification challenge, the SE module improved top-1 and top-5 accuracy by approximately 2.5% and 1.0%, respectively. Furthermore, the SE module can be applied to various deep learning models such as MobileNet and DenseNet.

3. Dataset

The CIFAR-10 dataset is a collection of images commonly used for training machine learning and computer vision algorithms, and represents one of the most widely utilized datasets in machine learning research[1][2]. CIFAR-10 contains 60,000 color images of 32$\times$32$pixelsdividedinto10distinctcategories$[3].$Thesecategoriesrepresentairplanes, a$u
$10servesasanimagecollectionforteachingcomputersobjectrecognition.Duetoitslowresolution$(32$\times$$32),

CIFAR-10 enables researchers to quickly experiment with different algorithms to determine which works best. CIFAR-10 is a labeled subset of the 80 Million Tiny Images dataset, annotated by students during its creation[5]. Various types of convolutional neural networks generally perform best at recognizing images in CIFAR-10. The dataset is divided into five training batches and one test batch, each containing 10,000 images. The test batch contains 1,000 randomly selected images per class. The training batches contain the remaining images in random order, though some batches may contain more images from certain classes than others. Across all training batches, each class contains exactly 5,000 images. Figure 2 [Figure 2: see original paper] shows the dataset classes with 10 random images from each class.

4. ResNet Performance Analysis

4.1 ResNet Architecture

ResNet architectures consist of two block types: Basic Block based on two convolutional layers, and Bottleneck Block based on three convolutional layers. Basic Blocks are suitable for shallow networks such as ResNet18 and ResNet34, while Bottleneck Blocks are appropriate for deeper networks like ResNet50 and ResNet101.

The structures of Basic Block and Bottleneck Block are shown in Figure 3 [Figure 3: see original paper], with Basic Block on the left and Bottleneck Block on the right. Basic Block comprises residual modules of two 3$×3convolutionallayers, withidenticalinputandoutputchannelnumbers.Wheninputandoutputdimensionsdi convolutional layer should be added to the input to match dimensions.

Unlike Basic Block, Bottleneck Block adds a 1$×1convolutionallayertoeachresidualconnectionfordimensionm 3×3+1×1structuretoreplacethetwo3×3convolutionallayers, therebyreducingcomputationalcomplexityandpa convolutional layers in shallow networks may lose some feature information and cause performance degradation.

Our ResNet implementation has the following structure: The input layer accepts images of size 224$×224×3(3RGBchannels).Thefirstconvolutionallayerisa3×3layerwithstride2, using64filte tensor. Finally, a fully connected layer maps the 512-dimensional features to the number of classes.

4.2 Experimental Setup and Results

Experiments were conducted using the PyTorch framework with the CIFAR-10 dataset described above, where images are 32$×$32 pixels. Two models were trained and evaluated: (1) a baseline ResNet-18 model, and (2) our modified ResNet-18 implementation. We used Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.1, momentum of 0.9, and weight decay of 5e-4. Two different learning rates were used for training to compare their effects. Models were trained from scratch for 200 epochs, with training images normalized to ensure proper data standardization.

Training Results: Figures 4 and 5 show the training set results. Training accuracy is not particularly meaningful as both models can achieve 100% accuracy.

Test Results: Figures 6 [Figure 6: see original paper] and 7 [Figure 7: see original paper] present test results for the baseline ResNet and our modified version, respectively. Figure 8 [Figure 8: see original paper] compares the accuracy of both trained models. Comparing classification results on the test set reveals that our ResNet achieves a 6% accuracy improvement over the baseline while exhibiting lower loss.

4.3 Discussion

Our ResNet implementation uses smaller convolutional kernels compared to the baseline. Specifically, we employ 3$\times3kernelswhilethebaselineuses7\times7kernels.The7\times7kernelsenabledownsan$ kernels enable finer-grained feature extraction in the input layer, preserving edge information and reducing blur and ambiguity. Our smaller kernel size may enhance detection capability for small objects, though it may underperform the baseline for larger object detection.

Our modified ResNet adds a global average pooling layer before the output layer. Using pooling and fully connected layers as output enables global average pooling of feature maps, reducing model parameters, lowering overfitting risk, and capturing more feature information through overall feature map averaging. Additionally, average pooling makes the model's feature representations more generalizable and transferable. Using a fully connected layer as output enables direct linear transformation of feature maps for predictions but requires more parameters and computation, potentially necessitating adjustments to output dimensions, class numbers, or regression ranges. This may also cause inconsistent gradient updates between the output and other layers, requiring residual connections to address the issue.

5. Impact of SeNet

5.1 Integrating SeNet into ResNet

As shown in Figure 9 [Figure 9: see original paper], SENet provides four methods for embedding SE modules into ResNet architecture: SE, SE-Pre, SE-Post, and SE-Identity.

In SEResNet design, the SE module is inserted into each residual block of the ResNet architecture after the final convolutional layer and before the final addition operation. The SE module's output is then added to the residual connection to obtain the block's final output.

In SE-Pre ResNet design, the SE module is inserted into each pre-activation residual block of the pre-activation ResNet architecture after the first batch normalization layer and before the final convolutional layer. The SE module's

output is then added to the residual block's input before the first batch normalization layer.

In SE-Post ResNet design, the SE module is added after the final convolutional layer of each residual block in the network. The residual block's output is then added to the SE module's output to obtain the block's final output. This approach is similar to SEResNet but places the SE module after the final convolutional layer rather than before the final addition operation.

In SE-Identity ResNet design, the SE module is added to the identity shortcut connection within each residual block. The SE module's output is then added to the identity shortcut to obtain the block's final output. This method resembles SEResNet except that the SE module is added to the identity shortcut rather than the main branch of the residual block.

For our evaluation, we adopt the SEResNet design to integrate SE into ResNet-18, which is also the standard design in the original SENet paper.

5.2 Evaluation

Original images in the CIFAR-10 dataset are 32$\times$32$pixels, whichweresizedto$224$\times$$224 pixels for training. Image upscaling introduces some noise into the data that may affect model performance, but we believe this helps evaluate the effectiveness of the SENet architecture in improving ResNet-18 classification performance.

Two models were trained and evaluated: (1) a baseline ResNet-18 model, and (2) a ResNet-18 model with integrated SE modules. We used Stochastic Gradient Descent (SGD) optimizer with learning rates of 0.1 and 0.001, momentum of 0.9, and weight decay of 5e-4. Two different learning rates were used to compare their effects. Models were trained from scratch for 100 epochs, with training images normalized to ensure proper data standardization.

Figure 10 [Figure 10: see original paper] shows training accuracy and loss on the test set across epochs for both models trained with learning rate lr=0.1. Due to the large learning rate and noise introduced by image upscaling, the ResNet model exhibits significant accuracy fluctuations during training. After integrating the SENet module, these fluctuations are notably reduced and average accuracy improves. Figure 11 [Figure 11: see original paper] presents results using learning rate lr=0.001, where severe accuracy fluctuations disappear and accuracy stabilizes after 50 epochs. In this case, ResSENet still achieves higher accuracy than ResNet, showing a 1% improvement. Figure 12 [Figure 12: see original paper] visualizes the accuracy comparison between the two models under different learning rates. Experimental results demonstrate that adding SENet is effective across various scenarios, providing significant noise suppression and accuracy improvement.

5.3 Discussion

This study evaluated the impact of adding SE blocks to ResNet on image classification performance and examined effects under different learning rates. Our experimental results demonstrate that adding SENet significantly improves model performance, particularly by substantially reducing accuracy fluctuations.

The primary reason for SENet's performance improvement is the attention mechanism provided by SE blocks, which can selectively amplify informative features while suppressing less useful ones. This mechanism helps the network focus on the most important features, thereby improving feature discriminability and making the network more robust to noise.

Furthermore, our results indicate that SENet integration is particularly effective in high-noise scenarios where accuracy fluctuations are more pronounced. SE blocks enhance ResNet's overall stability by reducing the amplitude of accuracy fluctuations during training. This suggests that SENet can help mitigate the impact of noise on model performance, which is an important consideration for practical deep learning applications.

In summary, our research demonstrates that adding SENet to ResNet can significantly improve performance on image classification tasks. The attention mechanism provided by SE blocks plays a crucial role in enhancing feature discriminability and reducing noise impact on the model. These findings have important implications for deep learning model design across various applications.

As discussed in Section 4.3, smaller images may require smaller convolutional kernels and minimal stride to prevent missing important image features. To reduce computational complexity while maintaining a broad receptive field and preserving original image information, $7\$\times 7 kernels can be used for downsampling. Conversely, 3\times\$3$ kernels enable more precise feature extraction, preserving edge details and reducing blur and ambiguity. While our implementation with smaller kernels may enhance detection of smaller objects, it may not perform as well as the baseline for detecting larger objects.

As mentioned in Section 5.3, the primary reason for SENet's performance improvement is the attention mechanism provided by SE blocks, which selectively amplifies informative features while suppressing less useful ones. This mechanism helps the network focus on the most important features, improving feature discriminability and robustness to noise. Our results also demonstrate that SENet integration is particularly effective in high-noise scenarios where accuracy fluctuations are more pronounced.

Moreover, SE blocks enhance ResNet's overall stability by reducing accuracy fluctuations during training. This indicates that SENet can help mitigate noise impact on model performance, which is an important consideration for practical deep learning applications.

In conclusion, our research shows that adding SENet to ResNet significantly improves image classification performance. The attention mechanism provided by SE blocks plays a key role in enhancing feature discriminability and mitigating noise effects on the model. These findings have important implications for designing deep learning models for various applications.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[2] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[3] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018.

[4] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In Proceeding of the IEEE conference on computer vision and pattern recognition, pages 4700–4708, 2017.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[7] ChristianSzegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, DragomirAnguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.

[8] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pages 6105–6114. PMLR, 2019.

Note: Figure translations are in progress. See original paper for figures.