Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

Development of convolutional neural network and its application in image classification: a survey

Wei Wang
Yujing Yang
Xin Wang
Weizheng Wang
Ji Li

SPIE.

Wei Wang, Yujing Yang, Xin Wang, Weizheng Wang, Ji Li, “Development of convolutional neural network and its application in image classification: a survey,” *Opt. Eng.* **58**(4), 040901 (2019), doi: 10.1117/1.OE.58.4.040901.

Optical Engineering 58(4), 040901 (April 2019)

REVIEW

Development of convolutional neural network and its application in image classification: a survey

Wei Wang, Yujing Yang, Xin Wang,* Weizheng Wang, and Ji Li
Changsha University of Science and Technology, School of Computer and Communication Engineering, Changsha, China

Abstract. In recent years, convolutional neural networks (CNNs) have been widely used in various computer visual recognition tasks and have achieved good results compared with traditional methods. Image classification is one of the basic and important tasks of visual recognition, and the CNN architecture applied to other visual recognition tasks (such as object detection, object localization, and semantic segmentation) is generally derived from the network architecture in image classification. We first summarize the development history of CNNs and then analyze the architecture of various deep CNNs in image classification. Furthermore, not only the innovation of the network architecture is beneficial to the results of image classification, but also the improvement of the network optimization method or training method has improved the result of image classification. We also analyze each of these methods' effect. The experimental results of various methods are compared. Finally, the development of future CNNs is prospected. [DOI: 10.1117/1.OE.58.4.040901]

Keywords: visual recognition; image classification; convolutional neural network; deep convolutional neural network; network optimization.

Paper 181562V received Nov. 3, 2018; accepted for publication Mar. 12, 2019; published online Apr. 11, 2019.

1 Introduction

The convolutional neural network (CNN) was first proposed in 1960s. Hubel and Wiesel[1] observed for the first time that neurons in the visual cortex were sensitive to moving edge in their experiments on visual cortex cells of cats and proposed the concept of "receptive field." They further discovered the hierarchical processing mechanism of information in visual cortical pathways, pointing out that simple cells detect location information, and complex cells integrate information stimulated by simple cells. The concept of "receptive field" proposed in Ref. 1 was later introduced into the research work of CNNs. In the 1980s, Fukushima and Miyake[2] proposed "neocognitron" based on the "receptive field," which can be regarded as the first implementation of CNNs. Neocognitron decomposes a visual model into several submodels and then processes them on the hierarchical and progressive connected feature planes so that the recognition can be completed even if the object has displacement or slight deformation. Neocognitron is the first artificial neural network based on local connectivity and hierarchical structure among neurons. But at that time, due to the lack of suitable learning algorithm, the network adopted other unsupervised algorithms and was mainly applied to handwritten digit recognition.

After that, researchers have tried to use the multilayer perceptron to learn features instead of manual design features and trained the model with the backpropagation (BP) algorithm, which was first proposed by Paul.[3] Through the work of Rumelhart et al.,[4] BP gained recognition. LeCun et al.[5] presented an application of BP networks to handwritten digit recognition, which show that large BP networks can be applied to real image-recognition problems without a large, complex preprocessing stage requiring detailed engineering. LeCun et al.[6] summarized the end-to-end training principle of modular system and proposed a CNN architecture called "LeNet-5," which showed better performance than all other techniques on a standard handwritten digit recognition task at that time. However, since some shallow machine learning models[7,8] were proposed one after another at that time, and the traditional BP neural network would encounter problems such as local optimum, overfitting, and vanishing-gradient[9] as the number of network layers increased, the research on deep neural network model was shelved.

Hinton et al.[10,11] found that the artificial neural network with multiple hidden layers has excellent feature learning ability. The learned features are more fundamentally to characterize the data, which is beneficial to visualize or classify the data, and the vanishing-gradient problem in neural network training can be alleviated through normalized initialization.[12] Since then, deep learning has attracted more and more attention. The CNN model AlexNet presented by Krizhevsky et al.[13] at the ILSVRC-2012 image classification competition[14,15] achieved a top-5 test error rate of 15.3%, almost halved the error rate of image classification compared to 26.2% achieved by the second-best entry.

CNNs have been proved to be effective in various fields of visual recognition[13,16–18] and have attracted more and more attention from researchers in the field of deep learning. Lecun et al.[19]published a review article in *Nature* titled "Deep learning," which sheds light on the basic principles and core strengths of deep learning.

First, this paper introduces the history of CNN and then analyzes the development of CNN architecture in image classification. Then the advantages and disadvantages of various convolution network architectures are compared and analyzed, and the future development of CNN is prospected.

2 Deep Convolutional Neural Network

Since AlexNet[13] achieved amazing results in ILSVRC-2012 image classification competition, more and more researches have focused on the improvement of the architecture of

*Address all correspondence to Xin Wang, E-mail: wangxin@csust.edu.cn

CNN. Visual geometry group (VGG)[20] and the inception module of GoogLeNet[21,22] demonstrated the benefits of increasing network depth and width. ResNets[23,24] constructed the residual learning block through the shortcut connection of identity mapping, making the neural network model break through the barrier of hundreds or even thousands of layers. DenseNet[25] and others[26] confirmed that refomulations of the connections between network layers can further improve the learning and representational properties of deep networks. In this section, we first introduce the basic composition and characteristics of CNNs through the network model of LeNet-5 proposed by LeCun et al.[6] Then the classical deep CNN model structure in recent years is analyzed accordingly.

2.1 *LeNet*

Lecun et al.[6] proposed a CNN named LeNet-5. The network model of LeNet-5 is shown in Fig. 1. According to Fig. 1, CNN architecture is generally composed of convolution layers, subsampling (pooling) layers, and fully connected layers. The following three sections are explained in turn.

2.1.1 *Convolution layer*

The convolution layer consists of multiple feature maps, which are obtained by convolution of the convolution kernel with the input signal. Each convolution kernel is a weight matrix, which can be a 3×3 or 5×5 matrix for a two-dimensional (2-D) image of a single channel. Figure 2 illustrates an example of the 2-D convolution.

The convolution operation provides a way to process variable-size inputs using convolution kernels, and different input features are extracted through convolution operation in convolution layer. The first layer extracts lower-level features such as edges, end points, and corners. Then the higher layer extracts more complex and higher-level features by processing the lower-level features. Convolution layer mainly has the characteristics of sparse interactions and weight sharing.

Sparse interactions. Traditional neural networks use matrix multiplication to build connections between inputs and outputs. Each output unit interacts with each input unit. When an input image contains thousands of pixels, this connection will increase the storage requirements of the model and increase the amount of calculation. Different from the traditional connection, the convolution network has the characteristic of sparse interactions (also known as sparse connectivity), which is achieved by controlling the size of the convolution kernel far less than the size of the input. The graphical interpretation of the sparse connections is shown in Fig. 3. In this figure, the input unit X_3 and the output unit affected by X_3 are highlighted. If there are m inputs and n outputs, the fully connected form of the model requires $m \times n$ parameters and the complexity of the corresponding algorithm is $O(m \times n)$. In sparse connection, the connections number of per output is $k(k \ll m)$, so this connection only needs $k \times n$ parameters and the complexity of the corresponding algorithm is $O(k \times n)$. The sparse interaction of convolution layer not only reduces the storage requirements of the model but also requires less computation to obtain the output, thus improving the efficiency of the model.

Weight sharing. The convolution layer also has the characteristic of weight sharing, which is realized by the convolution kernel. Convolution kernels are used to control the number of parameters and to impose a spatially restricted weighting to handle variable-size inputs. Weight sharing means that units in a layer use the same weights and deviations. For example, the C1 layer of LeNet-5 is a convolution layer, which is obtained through the calculation of six convolution kernels, and each convolution kernel has a fixed weight when convolving with the previous layer. When the input is a single-channel signal, the C1 layer contains six convolution kernels with the size of $1 \times 5 \times 5$. If the bias is taken into account, the C1 layer contains a total of $(63 \times 5 \times 5 + 6) = 156$ parameters. Compared with the fully connected network architecture, the weight sharing reduces the network training parameters to a greater extent, which can effectively prevent the network overfitting caused by a large number of parameters and improve the efficiency of network operation.

2.1.2 *Subsampling layer*

Usually, a subsampling (pooling) layer is inserted periodically between the convolution layers, whose function is to gradually reduce the spatial size of the data, so as to reduce the number of parameters in the network and reduce the consumption of computing resources. The pooling layer can also learn some invariant features of the input. Commonly used pooling layer methods are global average pooling[27] and max pooling. The input data processed by the pooling layer is generally a feature map obtained after convolution operation. The most commonly used max pooling layer is shown in Fig. 4. It can be seen that the max pooling unit is only

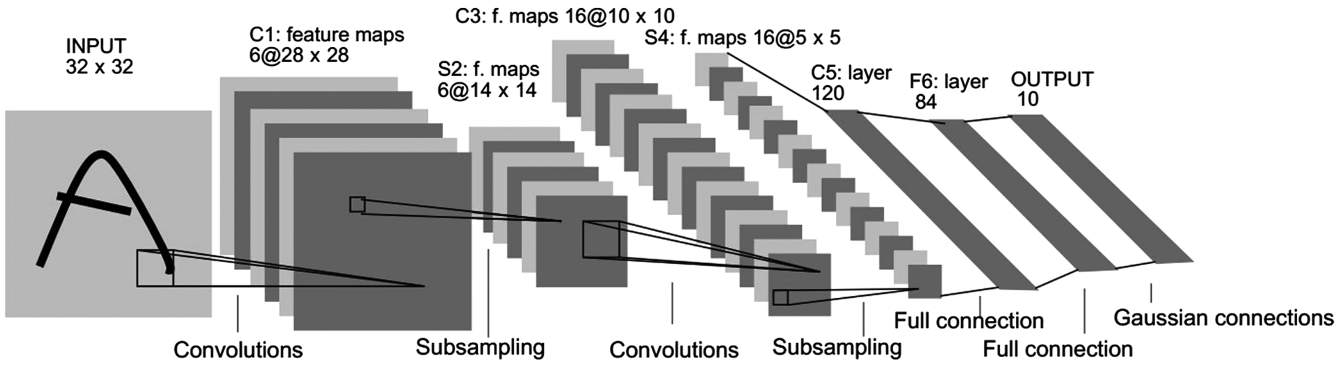

Fig. 1 LeNet-5 model diagram.[6]

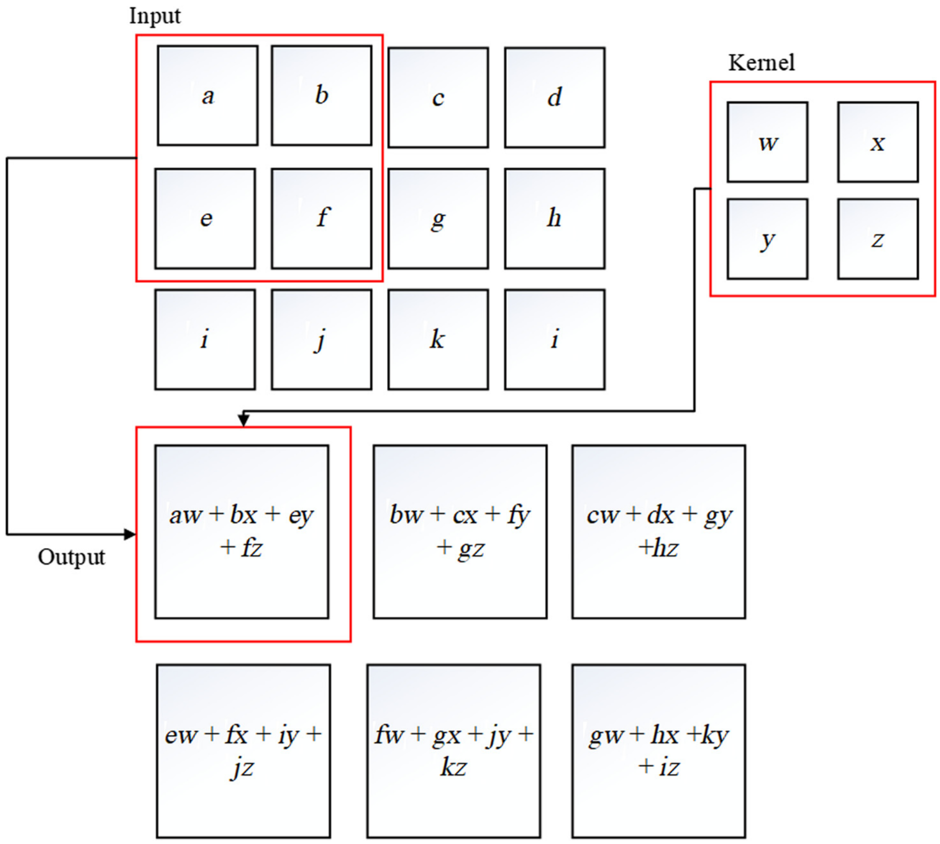

Fig. 2 An example of a 2-D convolution.

sensitive to the surrounding maximum, not to the exact location. Therefore, by pooling the obtained features, we can learn some invariant features of the input. In LeNet-5, the max pooling layer mainly uses a spatial window with a size of 2×2 and a step size of 2 to convolute. The maximum value in this window is taken as the output result.

2.1.3 *Fully connected layer*

After a series of convolution and pooling layers, the feature map of the image is extracted, and all the neurons in the feature map are transformed into a fully connected layer. Finally, the output can be classified by softmax layer. The function of the fully connected layer is to integrate the local information with class distinction both in convolution layer and pooling layer[28] so as to improve the performance of the whole CNN.

LeNet-5 is a classical CNN architecture. The combination of convolution layer, pooling layer, and fully connected layer is still the basic components of modern deep CNN. LeNet-5 has a groundbreaking significance for the development of deep CNNs.

2.2 *AlexNet*

Due to insufficient hardware computing and data, LeNet-5 did not attract enough attention after it was proposed. With the development of computer hardware and the increase in the amount of data available for neural network training, in 2012, AlexNet network[13] won the ILSVRC-2012 image classification competition[15] with a far lower error rate than the second place. Since then, deep neural networks have begun to attract widespread attention. The structure of AlexNet is shown in Fig. 5. Compared with LeNet-5, the improvements of AlexNet network architecture are as follows:

(1) *ReLU activation function*.[29] ReLU can introduce both nonlinearity and sparsity into the network. Sparsity can activate neurons selectively or in a distributed manner. It can learn relatively sparse features and achieve automatic dissociation.

(2) *Data augmentation*. AlexNet uses label-preserving transformations to artificially enlarge the dataset. The form of data augmentation consists of generating image translations, horizontal reflections, and altering the intensities of the RGB channels in training images.

(3) *Dropout*.[30] Neurons can be discarded from the network according to a certain probability to reduce network model parameters and prevent overfitting.

(4) Training on two NVIDIA GTX 580 3GB GPUs. With the development of GPU parallel computing ability, this method speeds up network training.

(5) *Local response normalization (LRN)*. The nearest data are used for normalization, and the classification results are improved slightly in Ref. 14.

(6) *Overlapping pooling*. The pooling step size is smaller than the corresponding edge of pooling kernel.

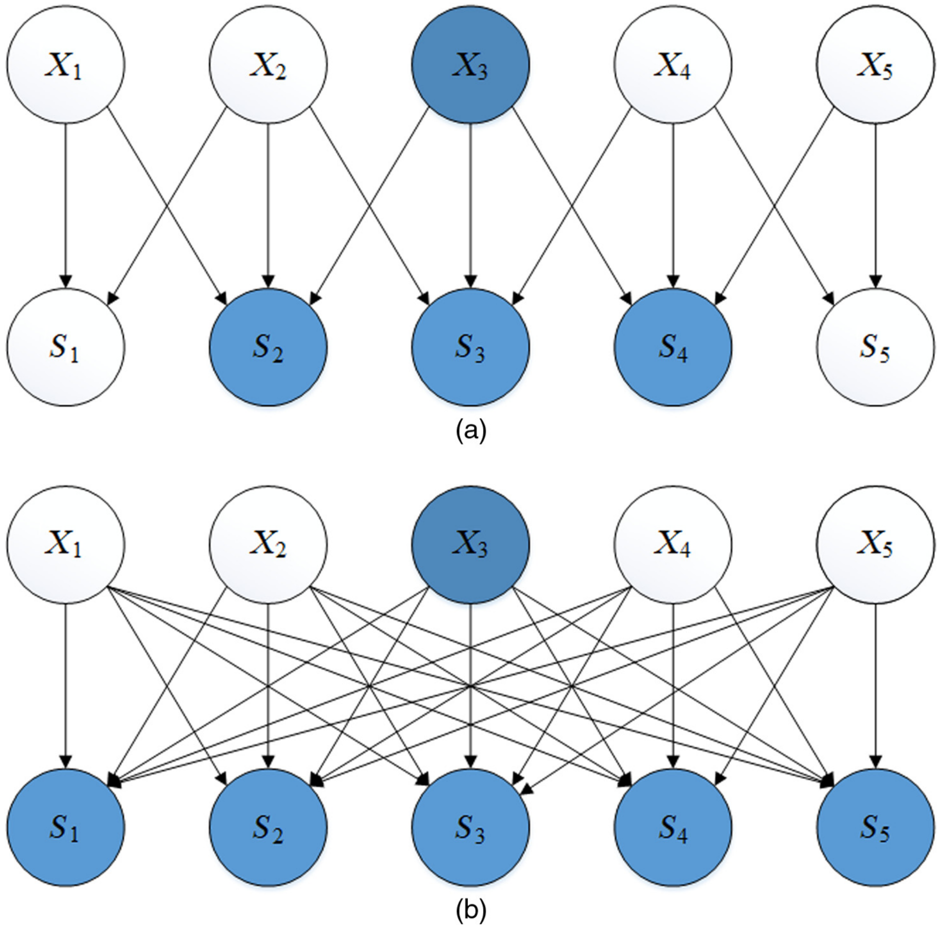

Fig. 3 Sparse connections: (a) when $\boldsymbol{S}$; is generated by convolution with a kernel width of 3, only three outputs are affected by X_3 and (b) when using the fully connected form, the connection is no longer sparse, and all outputs are affected by X_3.

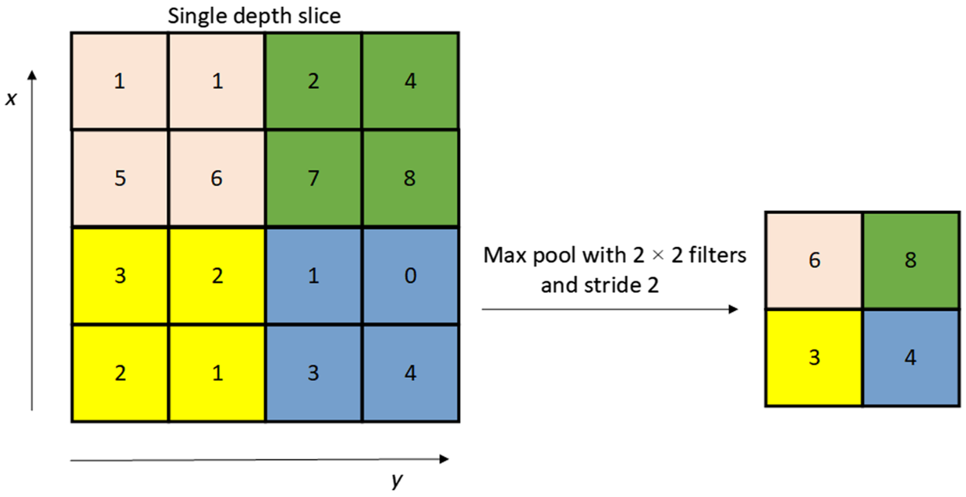

Fig. 4 Schematic diagram of the max pooling layer.

AlexNet is a milestone in the development of deep CNN, which has caused a new wave of neural network research. The success of AlexNet mainly depends on the development of computer hardware and the enhancement of data sets.

2.3 *ZFNet*

After AlexNet achieved excellent results in the ImageNet image classification competition, researchers began to study the CNN more deeply. However, there is no clear theoretical

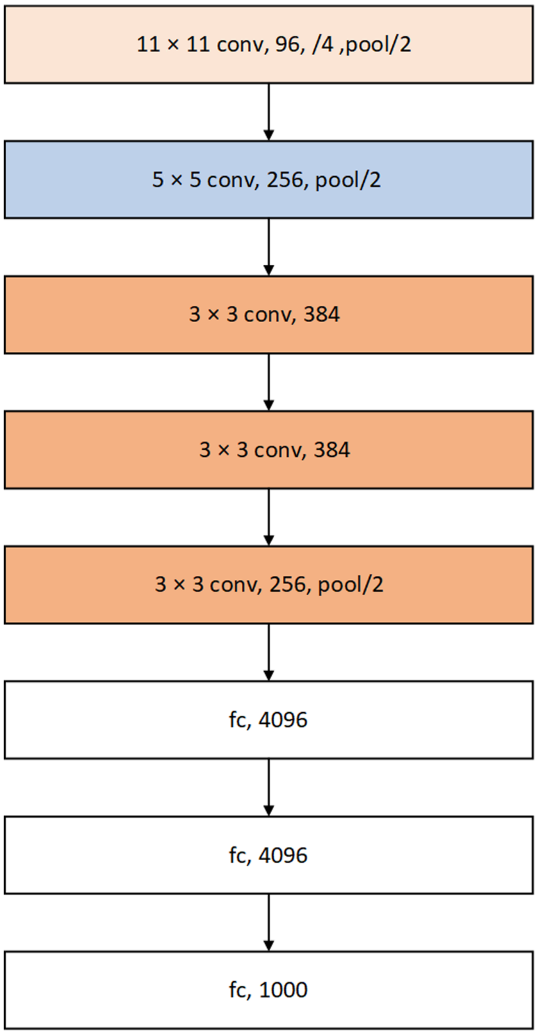

Fig. 5 AlexNet structure diagram.

explanation for why a CNN model can perform well. Zeiler and Fergus[31] proposed a visualization technique to understand CNNs and proposed ZFNet. The network has made minor improvements on AlexNet; the main contribution of Ref. 31 is to explain to a certain extent why CNNs are effective and how to improve network performance. The main contributions are detailed as follows:

(1) The deconvolution network is used, and the feature map is visualized. The feature maps prove that the shallow network learns the edge, color, and texture features of the image, and the high-level network learns the abstract features of the image. There are hierarchies among features. When the level is deeper, the invariance of the feature's invariance is stronger, and its discriminative ability is stronger too.

(2) By visualizing the feature map of the middle layer of the convolution network model, it is found in AlexNet, the feature extracted is blurred due to the large convolution kernel of the first convolution layer.

(3) Several occlusion experiments show that model is highly correlated with local features in classification.

(4) It is demonstrated that deeper network models have better performance.

The ZFNet is shown in Fig. 6. It changed the size of the convolution kernel in AlexNet's first layer from 11 × 11 to 7 × 7 and changed the step size of the convolution kernel from 4 to 2. Comparing ZFNet model with AlexNet single model, the error rate of top-5 is reduced by 1.7%,[31] which confirms the correctness of this improvement.

2.4 *VGG-16/19*

The shallow neural network model has certain limitations in large-scale image recognition tasks. In order to further explore the performance of the deeper network model, Simonyan and Zisserman[20] proposed the VGG. The main contribution of VGG is a thorough evaluation of networks of increasing depth using an architecture with very small (3 × 3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16 to 19 weight layers. Simonyan and Zisserman[20] mentioned six different network configurations and compared them on the ImageNet dataset. The configuration information of convolution network is shown in Table 1, and the performance of the corresponding network model is shown in Table 2.

Unlike AlexNet and ZFNet, VGG uses a small convolution kernel of 3 × 3 throughout the construction of the network and superimposes deep networks by superposing 3 × 3 small convolution kernels. In the experiment, in order to keep the computational complexity of the constituent structures at each feature layer roughly consistent, the number of convolution kernels at the next layer is doubled when the size of the feature map is reduced by half through the max pooling layer. The various configurations in Table 1 almost have

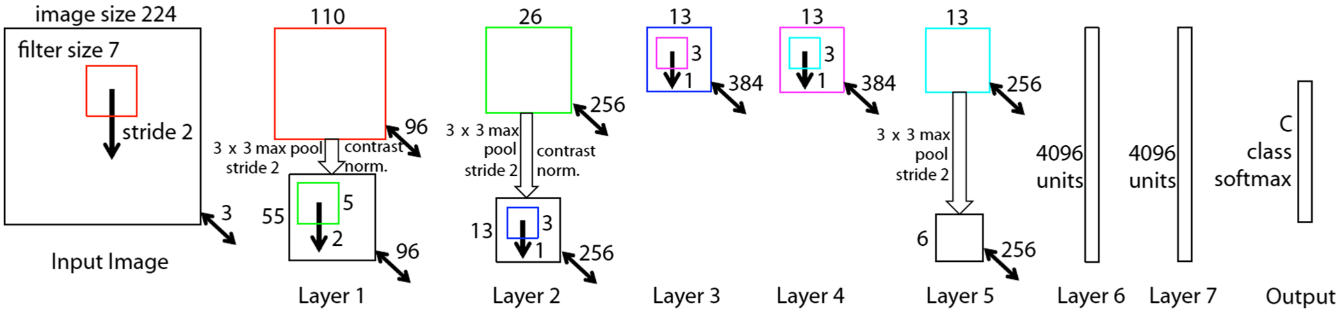

Fig. 6 ZFNet structure diagram.[31]

Table 1 ConvNet configuration.[20]

ConvNet configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
Input (224×224 RGB image)					
conv3-64	conv3-64	conv3-64	conv3-64	conv3-64	conv3-64
	LRN	**conv3-64**	conv3-64	conv3-64	conv3-64
Maxpool					
conv3-128	conv3-128	conv3-128	conv3-128	conv3-128	conv3-128
		conv3-128	conv3-128	conv3-128	conv3-128
Maxpool					
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
			conv1-256	**conv3-256**	conv3-256
					conv3-256
Maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
			conv1-512	**conv3-512**	conv3-512
					conv3-512
Maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
			conv3-512	**conv3-512**	conv3-512
					conv3-512
Maxpool					
FC-4096					
FC-4096					
FC-1000					
Softmax					

the same number of parameters, and Table 2 shows the results of various VGGNets in a single-scale test. The results show that the VGG-19 model achieved the best results, with an error rate of 8.0%. This also confirms that increasing network depth is beneficial to improve the accuracy of image classification. At the same time, it can be found that the result of A-LRN in Table 2 is worse than that of A. This also shows that the effect of LRN on classification results is not beneficial. With the introduction of batch normalization (BN),[32] LRN is replaced already.

The innovation of VGG is mainly the application of 3×3 small convolution kernels. The receptive field of two 3×3 convolutions is equivalent to that of a 5×5 convolution (as shown in Fig. 7), and the receptive field of three 3×3 convolutions is equivalent to that of a 7×7 convolution. The network used three 3×3 convolutions instead of a 7×7 convolution for two main reasons: First, it contains three ReLU layers instead of one, making the decision function more discriminatory; second, it can reduce the number of parameters. For example, if the input and output both have

Table 2 ConvNet performance at a single test scale.[20]

ConvNet config. (Table 1)	Smallest image side Train (*S*)	Test (*Q*)	Top-5 val. error (%)
A	256	256	10.4
A-LRN	256	256	10.5
B	256	256	9.9
C	256	256	9.4
	(256, 512)	384	8.8
D	256	256	8.8
	(256, 512)	384	8.1
E	256	256	9.0
	(256, 512)	384	**8.0**

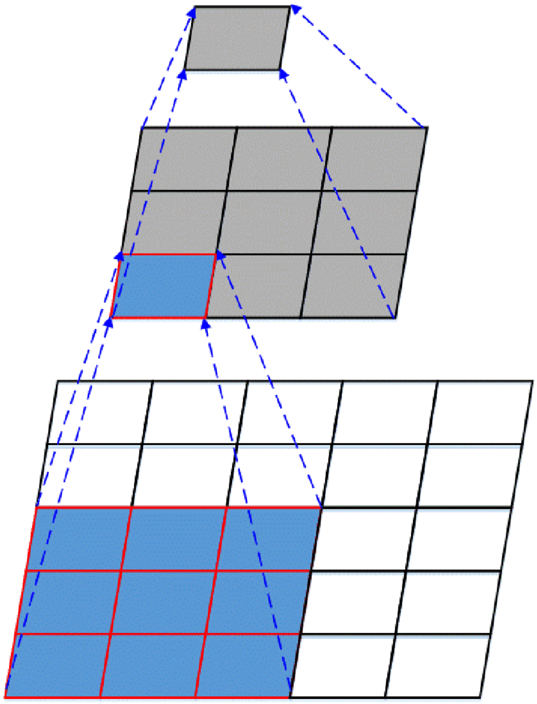

Fig. 7 A stack of two 3 × 3 convolutions replacing the 5 × 5 convolutions.

C channels, $3 \times (3 \times 3 \times C \times C) = 27 \times C \times C$ parameters are required for three convolution layers of 3×3, and $7 \times 7 \times C \times C = 49 \times C \times C$ parameters are required for one convolution layer of 7×7.

Before VGG, An et al.[33] also used small convolution kernels for experiments, but the network was not as deep as VGG and was not tested on large-scale ImageNet datasets. Using small convolution kernels, VGG can make the CNN reach a depth of 19 layers. In the ILSVRC-2014 image classification competition, VGG took the second place with a 7.3% (Ref. 20) top-5 error rate, this also confirms the benefits of neural network depth for neural network classification results.

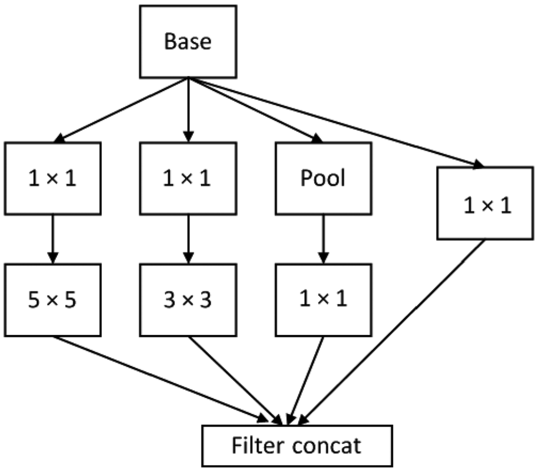

Fig. 8 Inception v1 module.[21]

2.5 *GoogLeNet/Inception v1 to v3*

GoogLeNet and VGG were the winner and runner-up of the ILSVRC-2014 image classification competition. VGG built a deeper network model through the construction of small convolution kernels, and GoogLeNet was inspired by network in network[27] to broaden the network structure and skillfully proposed the inception module.[21] The network with the inception module allowed the model to better describe the input data content while further increasing the depth and width of the network model. The inception module has been constantly updated and improved since it was proposed. The different versions of the inception modules are described as follows.

2.5.1 *Inception v1*

The biggest highlight of inception v1 is the introduction of 1×1 convolution kernel inspired by network in network.[27] The structure of inception v1 is shown in Fig. 8.

As can be seen from Sec. 2.1, one function of the convolution layer is to reduce and increase the dimension via using the number of channels (filters) in the convolution layer. In inception v1, the dimension is reduced mainly by 1×1 convolution kernel, which can reduce the number of network parameters and feature maps. The input feature maps are convoluted by 1×1 convolution kernel. This operation is equivalent to the original image scale transformation under the condition of unchanged size, which can greatly improves the accuracy of image classification. Inception v1 also uses convolution kernels of 1×1, 3×3, and 5×5, which also increases the adaptability of the network to the scale transformation of the input image.

The GoogLeNet constructed by inception v1 is shown in Fig. 9. Compared with VGG, GoogleNet has 22 layers, and the network is deeper and wider. GoogLeNet took the first place in the ILSVRC-2014 image classification competition with a 6.7% (Ref. 21) top-5 error rate.

2.5.2 *Inception v2*

The architecture of inception v2, as shown in Fig. 10, is mainly updated on the basis of inception v1 from the following aspects:

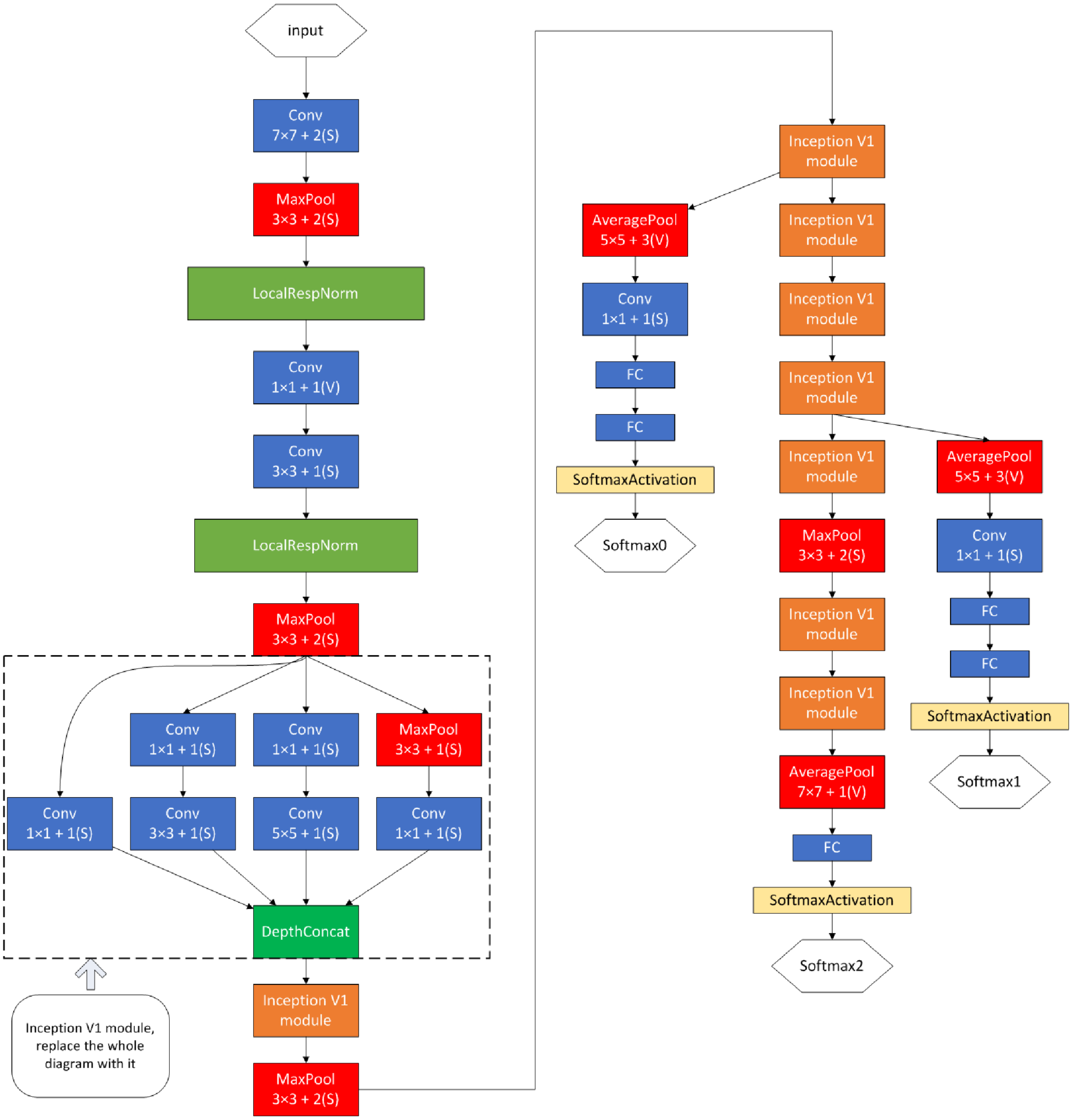

Fig. 9 GoogLeNet structure diagram.[21]

(1) BN[32] layer is added to normalize the output of each layer to a N (0, 1) Gaussian distribution so that the network can be converges faster and can be initialized more freely. BN layer not only increases the robustness of the model but also reduces the use of dropout as a regularization technique.

(2) In the model, a stack of two 3×3 convolution kernels are used to replace 5×5 convolution kernels in the inception v1 module, thus increasing the network depth. The overall depth of the 22-layer GoogLeNet built with inception v2 module has been increased by 9 layers.

Inception v2 architecture on the ImageNet test data set yielded a top-5 error rate of 4.9%,[22] which was lower than the 4.94% top-5 error rate of PReLU proposed by He et al.[34] in the same time. PReLU's top-5 error rate of 4.94% was the first to surpass human-level performance (5.1%)[15] on the visual recognition challenge.

2.5.3 *Inception v3*

The architecture of inception v3[22] is shown in Fig. 11. It is mainly updated on the basis of inception v2 as follows:

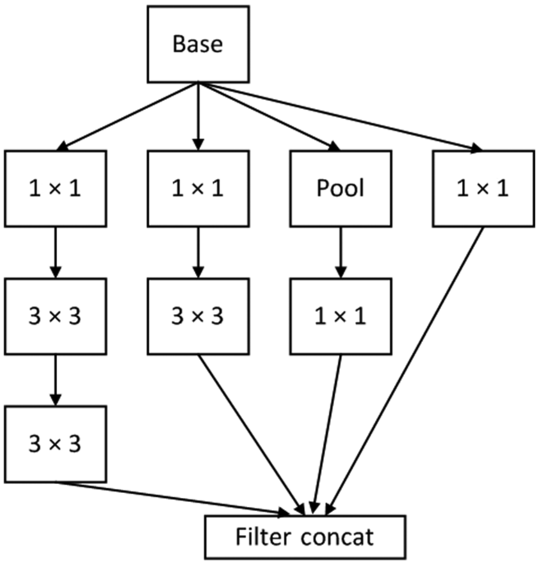

Fig. 10 Inception v2 module.[22]

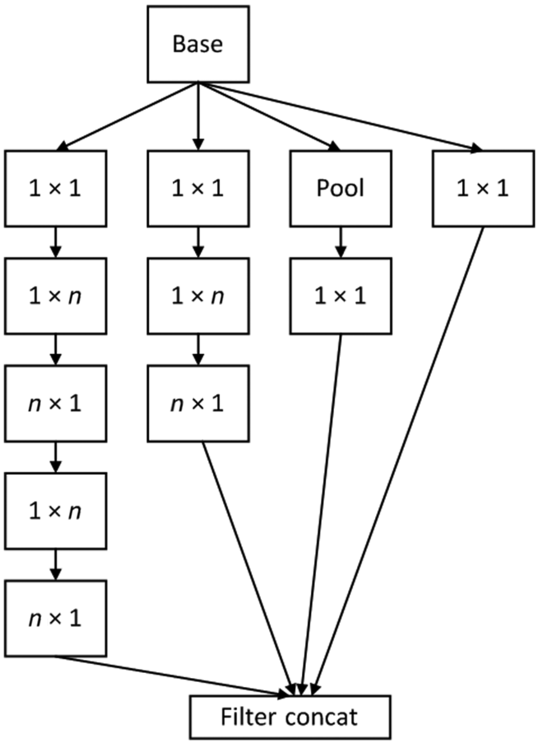

Fig. 11 Inception v3 module.[22]

(1) *Spatial factorization into asymmetric convolutions*: Using a 3 × 1 convolution followed by a 1 × 3 convolution is equivalent to sliding a two-layer network with the same receptive field as in a 3 × 3 convolution (see Fig. 12). If the number of input and output filters is equal, the two-layer solution is $\frac{[9-(3+3)]}{9} = 33\%$ cheaper than a single convolution layer. In theory, if one can replace any $n \times n$ convolution by a $1 \times n$ convolution followed by a $n \times 1$ convolution and the computational cost saving increases dramatically as n grows.

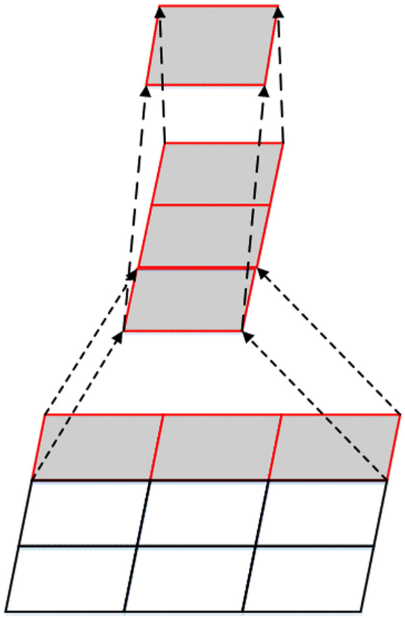

Fig. 12 Spatial factorization into asymmetric convolutions diagram.

(2) The network width has increased, and the network input has changed from 224 × 224 to 299 × 299.

Inception v3 module obtained 3.58% (Ref. 21) top-5 error rate of on the ImageNet test set.

2.6 *ResNets*

2.6.1 *ResNet*

It can be found from the above development of various CNN models that increasing the depth and width of neural network can improve the network performance. For example, VGG greatly improves network performance by adding network depth to AlexNet. For the original network such as VGG, simply increasing the depth will lead to vanishing/exploding gradients. He et al.[23] pointed out that the problem of vanishing gradients has been largely addressed by normalized initialization[12] and intermediate normalization layers. Although it is possible to train dozens of layers of networks by the above method, another problem arises, i.e., degradation problems. As shown in Fig. 13, when the number of network layers increased, the accuracy of training set was saturated or even decreased. This cannot be interpreted as overfitting, as overfit should be better in the training set. The degradation problem shows that deep networks cannot be optimized easily and well.

He et al.[23] proposed the ResNet in order to solve the above problems. The main contribution of ResNet is to solve the side effects (degradation) caused by increasing network depth so that network performance can be improved by simply increasing network depth. ResNet constructed by residual learning blocks can break through a 100-layers barrier and even reach 1000 layers.

The ResNet is mainly composed of the residual learning block, as shown in Fig. 14. In the residual learning block in Fig. 14, assuming the original function to be learned is $H(x)$, the residual learning block is then converted to $F(x) = H(x) - x$. These two expressions have the same effect, but

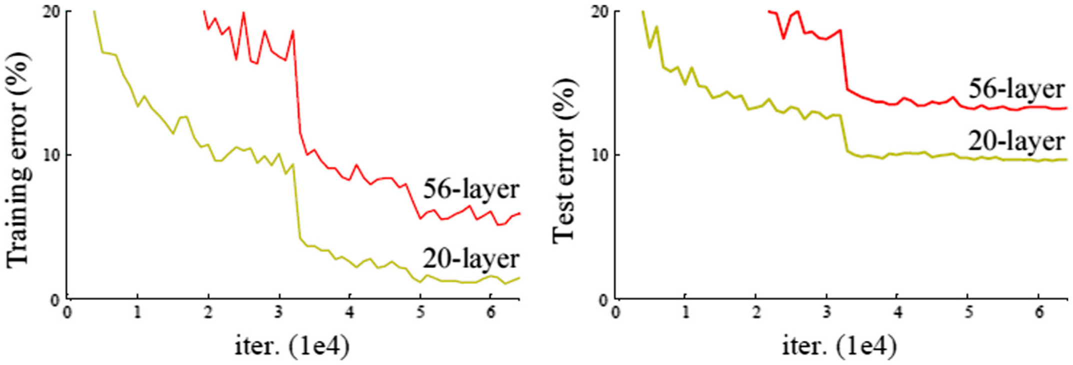

Fig. 13 Training error and test error on CIFAR-10 with 20-layers and 56-layers “plain” networks.[23]

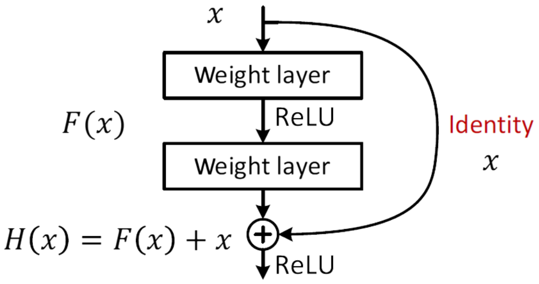

Fig. 14 Residual learning block.[23]

the difficulty of optimization is different. To the extreme, if an identity mapping was optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers.[23] The addition connection of the identity mapping does not add additional parameters and computation to the network but can greatly increase the training speed of the model and improve the training effect.

Two residual blocks are used in the ResNet network structure for ImageNet. One is to concatenate two convolution kernels of size 3×3 as one residual block shown in Fig. 15(left), and the other is to connect the 1×1, 3×3, and 1×1 kernels together as a “bottleneck” building block shown in Fig. 15(right). In the “bottleneck” building block, the first 1×1 convolution kernel mainly reduces the dimension of the feature map from 256-dimensional to 64-dimensional. Next, the convolution kernel of 3×3 is used for calculation, and finally the data dimension is changed to 256 using the convolution kernel of 1×1.

ResNet, constructed by residual learning block, won the first place in the ILSVRC-2015 image classification competition with a top-5 error rate of 3.57%.[23] As the number of layers increases, ResNet solves the “degradation” problem well, as shown in Figs. 16 and 17.

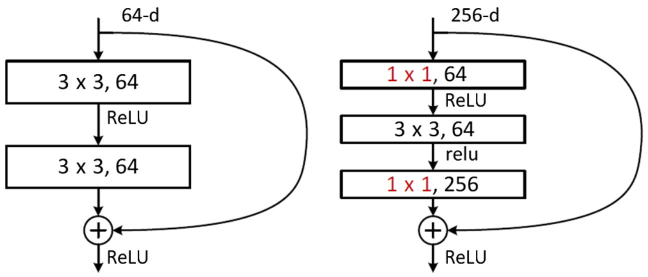

Fig. 15 A deeper residual function F for ImageNet.

2.6.2 *Improvement of ResNet*

Soon after ResNet put forward, He et al.[24] further studied the identity mapping in the residual learning block and improved it. The method is compared with the original ResNet and Highway networks.[23,35] The results further confirm the importance of identity mapping. The main contributions of Ref. 24 are as follows:

(1) By comparing the original shortcut connection with other shortcut connections as shown in Fig. 18, the importance of identity mapping is demonstrated further.

(2) Change the location of the ReLU activation function in the residual learning block and preactivate it as shown in Fig. 19 (e) so that the output of the residual learning block is still an identity mapping.

Among the various types of shortcut connections shown in Fig. 18, the network consisting of the original shortcut connections achieved a 6.61% (Ref. 24) error rate on the CIFAR-10 data set, which is better than the other connections. This confirms the importance of the identity mapping. In the experiment,[24] among the various usages of activation, the best classification results were obtained using a full preactivated connection [Fig. 19(e)].

2.6.3 *Other residual networks*

With the increasing depth of residual networks, the diminishing feature reuse will make the networks training very slow.[36] In order to reduce the impact of “feature disappearance,” Zagoruyko and Komodakis[37] proposed a wide-dropout block, as shown in Fig. 20(d). This block makes it possible to increase the depth of the original residual network by increasing the network width. The experiment also proves its feasibility.

ResNeXt, proposed by Xie et al.,[38] puts forward the concept of cardinality beyond depth and width, and points out that increasing cardinality is more effective than increasing the depth and width. The residual learning block of ResNext is shown in Fig. 21. ResNeXt secured second place in ILSVRC-2016 image classification competition with 3.03% top-5 error rate.[38] In addition, Szegedy et al.[39] proposed inception v4 by combining inception module with residual

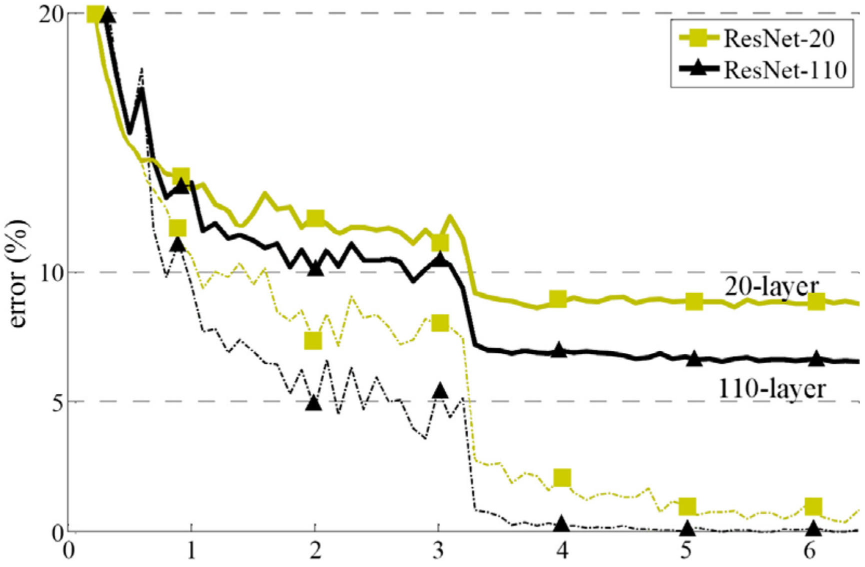

Fig. 16 Results of ResNet classification on CIFAR-10.[23] Dashed lines denote training error and bold lines denote testing error.

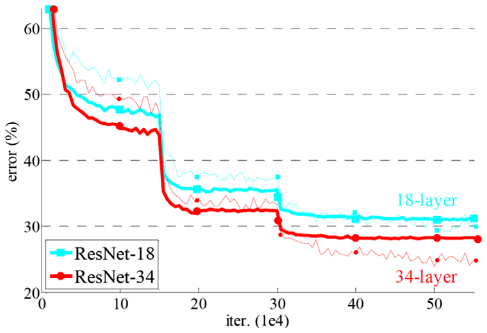

Fig. 17 Results of ResNet classification on ImageNet.[23] Thin curves denote training error and bold curves denote validation error of the center crops.

learning block and constructed inception-ResNet-v2 network. The network achieves good results in the ILSVRC-2016 image classification competition with a top-5 error rate of 3.08%.[39]

2.7 *DenseNet*

Since ResNet was put forward, many networks have been developed using ResNet. Each network has its own characteristics and its performance has been improved. As the depth of CNNs increases, the input or gradient must passes through many layers, which will vanish and "wash out" when it reaches the end (or beginning) of the network.[25] This has aroused people's rethinking of the network structure. Before the dense block was put forward, Huang et al.[40] trained deep network with stochastic depth to achieve good results. This shows that some network layers in the residual network carry unnecessary information for classification results, which can be discarded in training.

Based on Ref. 40, considering create short paths from early layers to later layers, Huang et al.[25] proposed DenseNet, which is mainly composed of dense blocks as shown in Fig. 22.

There are L connections with the traditional L layer neural network, and the dense block of L layer has $\frac{L\times(L+1)}{2}$ connections. The network setup growth rate k indicates the added number of input channels when pass through a layer. For example, assuming that K_0 as the number of input feature maps, and the output of each nonlinear transformation H is k feature maps, then the input of the i'th layer is $K_0 + (i-1)\times k$. One major difference between DenseNet and the previous mentioned networks is that DenseNet can accept fewer feature maps as the output of the network layer. DenseNet is constructed mainly by dense blocks, as shown in Fig. 23. In the same dense block, the feature size is required to be the same size. The transition layers are set between different dense blocks to achieve down sampling.

The main advantage of DenseNet is that the features extracted by some earlier layers can still be directly used by deeper layers through dense connections. Through the setting of the growth rate k, DenseNet can adjust the number of feature maps, thus effectively reducing the number of parameters.

DenseNet outperformed ResNet on the CIFAR-10 dataset, and on the ImgeNet dataset, DenseNet was able to converge faster by increasing the number of layers.[25] DenseNet is also widely used as a commonly used neural network model today.

3 Auxiliary Methods and Strategies

This section mainly introduces some auxiliary methods and strategies in the development of CNNs, including the improvement of activation functions, normalization, and some other strategies.

3.1 *Activation Function*

Before the ReLU activation function, the traditional neural network mostly uses sigmoid as the activation function. In general, sigmoid functions can be divided into logistic sigmoid and tanh sigmoid. The sigmoid function in this paper generally refers to the former, as shown in Fig. 24.

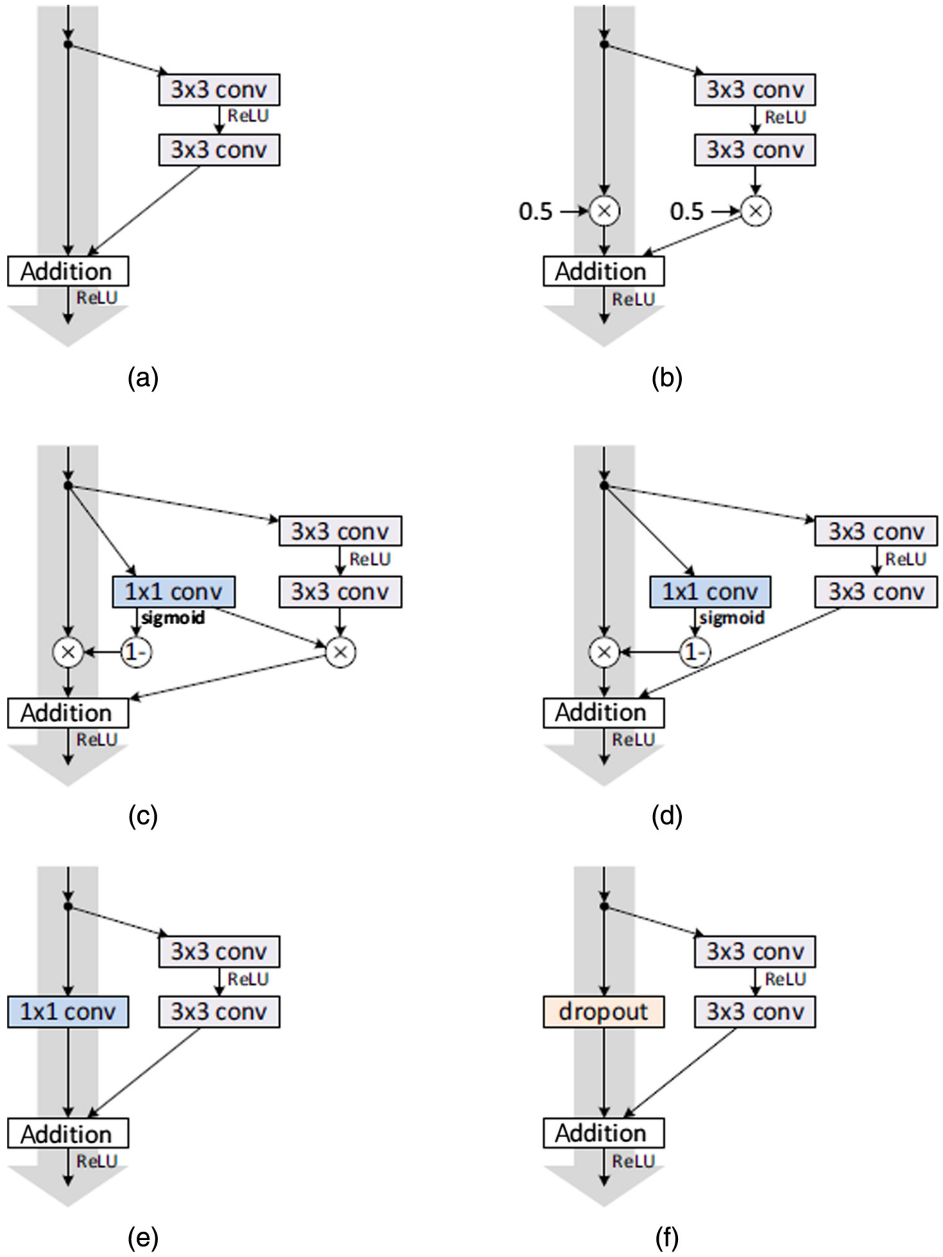

Fig. 18 Various types of shortcut connections: (a) Original, (b) constant scaling, (c) exclusive gating, (d) short-cut only gating, (e) conv shortcut, and (f) dropout shortcut.[24]

The output value of the sigmoid function is between 0 and 1, which is consistent with the definition of probability output. The nonlinear sigmoid function is widely used in the activation function because of its large signal gain in the central region and small signal gain on both sides, similar to the excitation and suppression states of neurons. However, when the number of neural network layers increases, the sigmoid gradient value will gradually become smaller, network learning becomes very slow, and even the gradient will vanish. Therefore, the network cannot be deepened indefinitely until the ReLU function is presented.

ReLU is the activation function used by many current network models. ReLU has the following advantages over sigmoid: unilateral inhibition, relatively wide excitation boundaries, sparse activation, and alleviate the vanishing-gradient problems.

Leaky ReLU[41] improved the negative half axis of ReLU function to avoid zero gradient, but the experimental results were not greatly improved. He et al.[34] put forward the PReLU function on this basis, as shown in Fig. 25. The learnable parameter a is added to PReLU. When $a = 0$, PReLU becomes the ReLU function, and when $a = 0.01$, PReLU becomes the leaky ReLU. Experiments show that this adaptive activation function can improve the classification results of the network.

3.2 *Normalization*

In the training process, when the input distribution in the hidden layer of the deep neural network is offset, the global distribution will gradually approach the upper and lower bounds of the value range of the nonlinear function, resulting

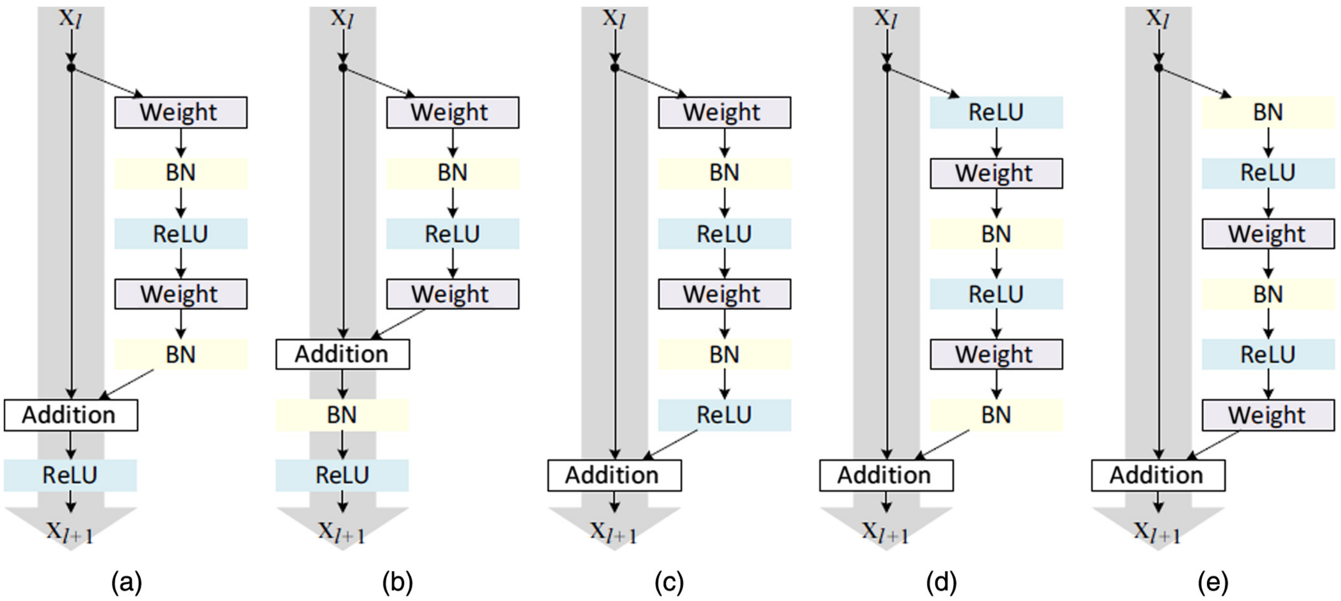

Fig. 19 Various usages of activation: (a) original, (b) BN after addition, (c) ReLU before addition, (d) ReLU-only preactivation, and (e) full preactivation.[24]

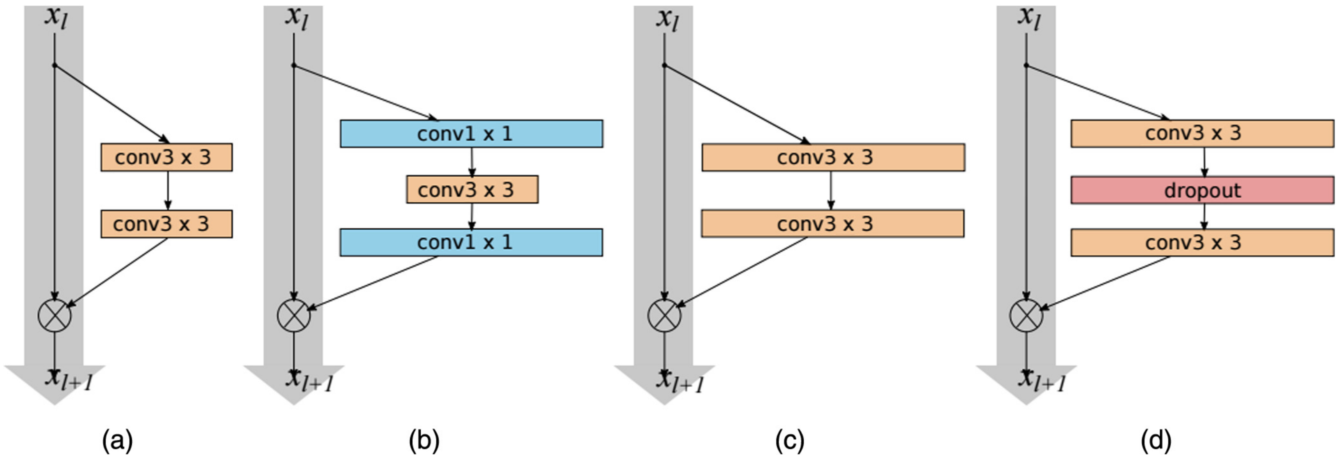

Fig 20 Wide-dropout block: (a) basic, (b) bottleneck, (c) basic-wide, and (d) wide-dropout.[37]

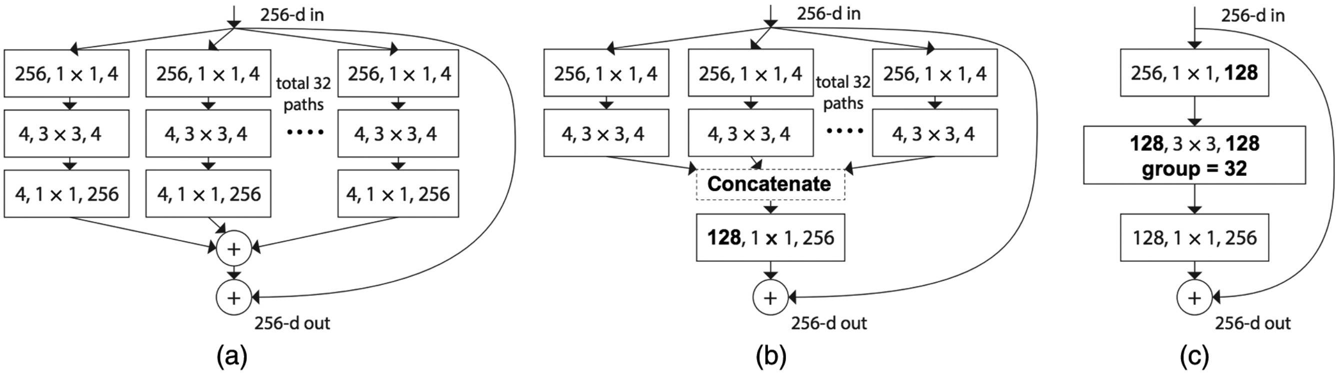

Fig. 21 Equivalent building blocks of ResNeXt.[38] (a) Aggregated residual transformations. (b) a block equivalent to (a), implemented as early concatenation. (c) a block equivalent to (a, b), implemented as grouped convolutions.

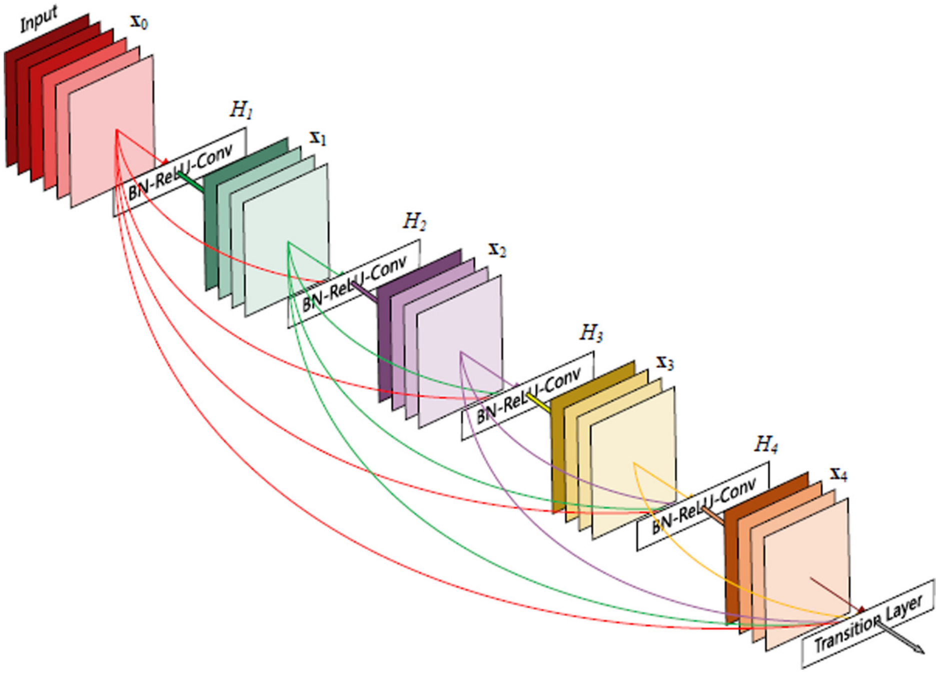

Fig. 22 Dense block.[25]

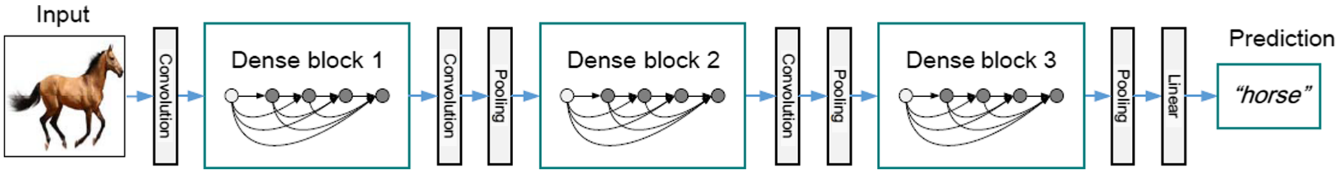

Fig. 23 DenseNet structure diagram.[25]

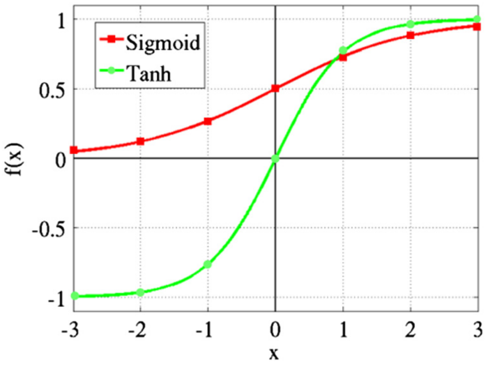

Fig. 24 Sigmoid and tanh function diagram.

in slow training convergence. Therefore, the network needs to be normalized.

The proposition of BN[32] has a milestone significance in the field of deep learning. BN takes minibatch as the unit to unify the input distribution of the nonlinear function into a standard normal distribution with a mean of 0 variance of 1, which makes the input value of activation function to fall in the region where the nonlinear function is sensitive. BN improves the speed of training, accelerates the convergence process, and improves the classification results. Moreover, BN can be seen as a regularization technique that prevents overfitting, similar to dropout.[30] The addition of BN in the network also makes the network initialization less demanding.

The disadvantage of BN is that the network is dependent on minibatch dimension, and the change of batch dimension will affect the classification effect. When the batch size is small, the network effect of using BN layer is obviously worse as shown in Fig. 26. This does not satisfy some networks that require batch size 1 or 2 for other visual recognition tasks.[16,42,43]

To alleviate this problem, layer normalization,[44] instance normalization,[45] group normalization,[46] and other normalization methods have been proposed. Figure 26 confirms that group normalization computation accuracy is more stable than BN when batch size changes. Figure 27 is a schematic diagram of various normalization methods.

Normalization is an indispensable part of the modern convolution network architecture. It has made a vital contribution to the development of CNNs.

3.3 *Other Strategies*

In the development of a CNN in image classification field, the improvement of some network initialization methods has also played a positive role. Network initialization is to ensure that the activation value of each layer does not appear saturation when the network is initialized, and the activation value of each layer is not 0. Sutskever et al.[47] proposed a

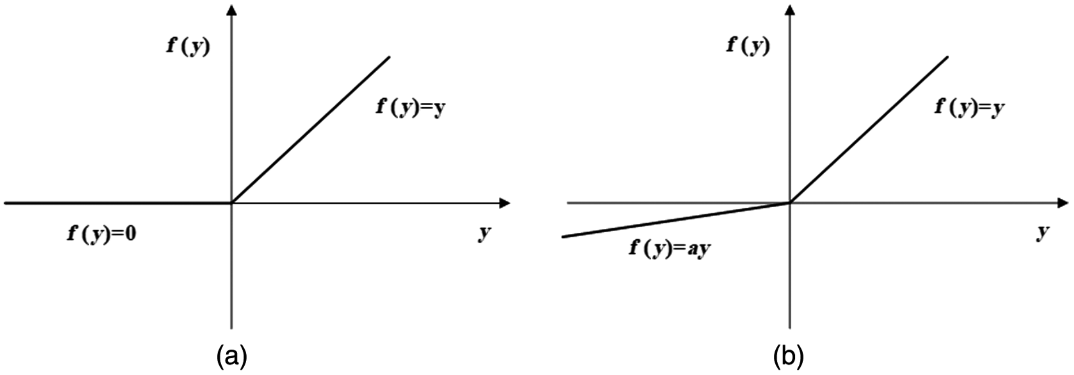

Fig. 25 ReLU vs. PReLU.[34] (a) ReLU function and (b) PReLU function.

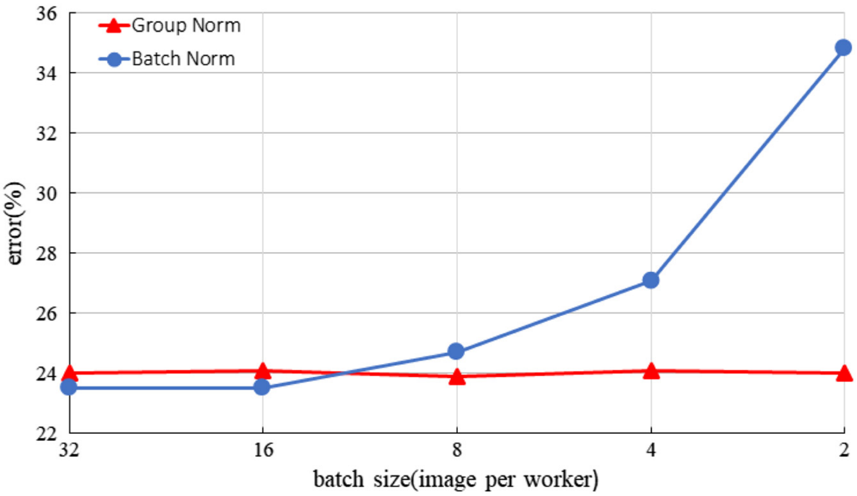

Fig. 26 Experimental results based on ImageNet dataset.

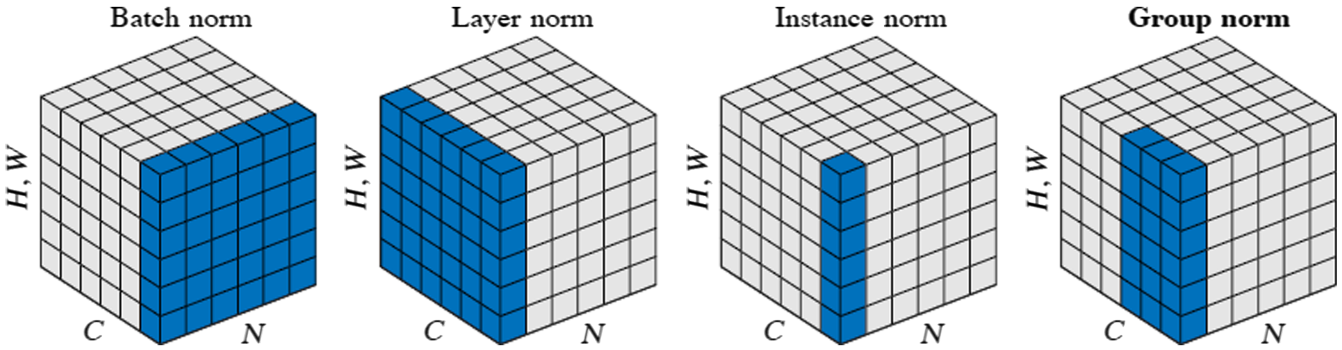

Fig. 27 All kinds of standardized schematic diagram.

Xavier initialization method to solve the network initialization problem. AlexNet[13] used random initialization for network training, and VGGNet[20] initialized the deep network by initializing the shallow model first and then applying its parameters to the deeper model. Ioffe and Szegedy[32] proposed BN and He et al.[34] proposed Microsoft Research Asia (MSRA) initialization method, they all considered the nonlinear ReLU function situation, and the deep neural network initialization problem was solved more effectively.

In addition to network initialization, the innovation of optimization method has also promoted the development of CNN. The optimization algorithm develops from stochastic gradient descent (SGD) to gradient descent with momentum[47], and then to Adam with adaptive learning rate,[48] which is widely used nowadays. In the latest work, Reddi et al.[49] explained how the exponential moving average used in Adam leads to nonconvergence through a simple convex optimization problem and proposed a beyond Adam algorithm.

Table 3 Experimental results of various image classification methods.

Model name	Year	Data set	Evaluating indicator	Parameters (million)	Experimental result (%)	Characteristics	Notes
Sparse coding[50]	2010	ImageNet	Top-5 (test) error	—	28.2	Averaging the prediction produced by six sparse coding models	ILSVRC-2010: first
SIFT + FVs	2012	ImageNet	Top-5 (test) error	—	26.2	Classifiers use different types of SIFT features to calculate and use the FV training	ILSVRC-2010: second
AlexNet[13]	2012	ImageNet	Top-5 (test) error	60	16.4	ReLU activation function is used. Dropout is used to reduce network parameters and prevent overfitting	ILSVRC-2012: champion. Experimental results are obtained through five similar CNN predictions. Training is carried out on double GPU
ZFNet[30]	2013	ImageNet	Top-5 (test) error	—	14.8	The deconvolution network is used to visualize the feature map. The size of AlexNet's first convolution kernel was changed from 11 × 11 to 7 × 7, and the step size was changed from 4 to 2	The experimental results are predicted through an average of six ZFNet network models
VGG[20]	2014	ImageNet	Top-5 (test) error	144 (VGG-19)	7.3	VGG uses small convolution kernels and modularized the design of the whole network	ILSVRC-2014: second. The result is the average of the results predicted by the seven VGG models
GoogLeNet[21]	2014	ImageNet	Top-5 (test) error	6.8	6.8	The inception module is designed, and the dimension reduction operation is carried out using 1 × 1 kernels	ILSVRC-2014: second. The result is the average of the results predicted by the seven VGG models
Inception-v2[22] (BN-inception)	2015	ImageNet	Top-5 (test) error	—	4.82	The BN layer is added, and two stacked 3 × 3 convolution kernels are used to replace 5 × 5 in the inception module	The results are the average of the six sets and multiscale training is adopted
PReLU[34]	2015	ImageNet	Top-5 (test) error	—	4.94	The adaptive PReLU function is proposed	The test model is changed to PreLU on the basis of MSRA[51] model
Inception-v3[22]	2015	ImageNet	Top-5 (test) error	—	3.58	Convolution operation of $n \times n$ is decomposed into convolution operation of $1 \times n$ and $n \times 1$	The experimental results obtained from the average four sets of models
ResNet[23]	2015	ImageNet CIFAR-10	Top-5 (test) error Error	25.5 (ResNet-50) 1.7 (ResNet-110)	3.57 6.61	The design of residual learning block is added	ILSVRC-2015: champion. The parameters of CIFAR-10 network are different from those of ImageNet, and the experimental results are processed by data enhancement operation

Table 3 (*Continued*).

Model name	Year	Data set	Evaluating indicator	Parameters (million)	Experimental result (%)	Characteristics	Notes
ResNeXt[38]	2016	ImageNet	Top-5 (test) error	25.0 (ResNeXt-50)	3.03	The concept of cardinality is proposed as an essential factor in addition to the dimensions of depth and width	ILSVRC-2016: second
DenseNet[25]	2017	CIFAR-10	Error	1.0 (DenseNet-40, $K = 12$)	5.24	A densely connected network structure is constructed	The network architecture is improved from the perspective of feature maps
				7.0 DenseNet-100 ($k = 12$)	4.10		
				27.2 DenseNet-100 ($k = 24$)	3.74		
SENet[52]	2017	ImageNet	Top-5 (test) error	35.7 ($r = 4$)	2.251	The channel relationship of feature map is reformed	ILSVRC-2017: champion

4 Comparison of Various Image Classification Methods

The analysis and comparison results of various image classification methods are shown in Table 3. The main comparative factors include model name, publication year, algorithm test data set, algorithm evaluation index, network model parameters, algorithm experimental results, algorithm characteristics, and notes (such as algorithm achievements, whether multiscale training is needed, and so on).

Table 3 compares and analyzes the performance of various image classification algorithms on the ImageNet data set or CIFAR-10 data set and summarizes the characteristics of the algorithm.

5 Summary

From the initial appearance of AlexNet to the gradual increase of network layers of VGGNet, all of them show the potential of neural network depth. The ingenious design of the inception module also shows the charm of the neural network architecture. ResNets further explores the effect of the neural networks depth, which plays a crucial role in the development of today's networks. On the other hand, DenseNet makes CNNs better for learning representation from the point of feature reuse, which provides a new perspective for the development of network architecture. In the following, the development trend of CNNs in image classification is prospected through several aspects.

5.1 *Application of Transfer Learning*

In the application of deep neural network, when we are faced with a large amount of data, it takes a lot of calculation and time to train the model and optimize the parameters after building the deep neural network model. If a model that has been trained for a large amount of time can solve the same kind of problems, then the cost performance of the model will be greatly improved, which promotes the use of transferable model to solve the same kind of problems.

Zeiler and Fergus[31] used a CNN for pretraining on ImageNet data sets, and then migrated the network to caltech-101 and caltech-256 for image classification data sets, respectively, for training and testing. The accuracy of image classification was improved by about 40%. Through transfer learning, we can apply a well-trained model to solve the similar problems by making small adjustments and achieve good results. At the same time, we can effectively solve the problem with less original data by adopting transferable model. Using transfer learning, the network model in image classification can be further applied to semantic segmentation, object detection, and other fields. In recent years, many researchers have devoted themselves to the field of transfer learning.

5.2 *Introduction of Visual Attention Mechanism*

In recent years, attention mechanism has been adopted in the field of deep learning. Visual attention mechanism is a special brain signal processing mechanism of human vision. By rapidly scanning the whole image, human vision can obtain the target area that needs to be paid attention to, then devote more attention resources to this area to obtain more detailed information of the target, and inhibit other useless information. Hu et al.[52] introduced attention mechanism to construct squeeze-and-excitation module, which reconstructs the relationship between feature channels by embedding multiscale processing. SENet won the ILSVRC 2017 image classification championship with a top-5 test set error rate of 2.251%.

In the future, the design of CNN framework can seek for introducing and strengthening attention mechanism in different layers to make computer vision closer to human visual ability.

5.3 *Study on the Stability of CNN*

A CNN has a large number of parameters, so the experiment of CNN often fails to achieve the effect of network in corresponding papers. At present, the parameter setting in training CNN is mostly based on experience and practice. The optimization analysis of parameters and the study of system stability are the problems to be solved.

5.4 *Hardware Development and Data Set Building*

The development of deep learning is inseparable from the innovation of hardware devices and the expansion of data sets. With the support of hardware devices and data sets, CNN will further help and solve the cognitive defects existing in the current network structure.

References

1. D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.* **160**(1), 106–154 (1962).
2. K. Fukushima and S. Miyake, "Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognit.* **15**(6), 455–469 (1982).
3. W. Paul, "Beyond regression: new tools for prediction and analysis in the behavioral sciences," PhD Thesis, Harvard University (1974).
4. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature* **323**(6088), 533–536 (1986).
5. Y. LeCun et al., "Handwritten digit recognition with a back-propagation network," *Adv. Neural Inf. Process. Syst.* **2**(2), 396–404 (1990).
6. Y. LeCun et al., "Gradient-based learning applied to document recognition," *Proc. IEEE* **86**(11), 2278–2324 (1998).
7. C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.* **20**(3), 273–297 (1995).
8. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Eur. Conf. Comput. Learn. Theory*, Berlin, Heidelberg, Vol. **55**, pp. 23–37 (1995).
9. M. Jones and T. Poggio, "Regularization theory and neural networks architectures," *Neural Comp.* **7**(2), 219–269 (1995).
10. G. E. Hinton, S. Osindero, and Y. W. The, "A fast learning algorithm for deep belief nets," *Neural Comput.* **18**(7), 1527–1554 (2006).
11. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science* **313**(5786), 504–507 (2006).
12. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.* **9**, 249–256 (2010).
13. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Int. Conf. Neural Inf. Process. Syst.*, Vol. **60**, 1097–1105 (2012).
14. J. Deng et al., "ImageNet: a large-scale hierarchical image database," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 248–255 (2009).
15. O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vision* **115**(3), 211–252 (2015).
16. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vision and Pattern Recognit.* (2015).
17. S. Ren et al., "Faster R-CNN: towards real-time object detection with region proposal networks," in *Int. Conf. Neural Inf. Process. Syst.*, Vol. **39**, pp. 91–99 (2015).
18. A. Toshev and C. Szegedy, "DeepPose: human pose estimation via deep neural networks," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1653–1660 (2014).
19. Y. Lecun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature* **521**(7553), 436–444 (2015).
20. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.* (2015).
21. C. Szegedy et al., "Going deeper with convolutions," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)* (2015).
22. C. Szegedy et al., "Rethinking the inception architecture for computer vision," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2818–2826 (2016).
23. K. He et al., "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vision and Pattern Recognit.* (2016).
24. K. He et al., "Identity mappings in deep residual networks," in *Eur. Conf. Comput. Vision*, pp. 630–645 (2016).
25. G. Huang et al., "Densely connected convolutional networks," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2261–2269 (2017).
26. Y. Chen et al., "Dual path networks," in *Int. Conf. Neural Inf. Process. Syst.* (2017).
27. M. Lin, Q. Chen, and S. Yan, "Network in network," in *Int. Conf. Learn. Represent.* (2014).
28. T. N. Sainath et al., "Deep convolutional neural networks for LVCSR," in *IEEE Int. Conf. Acoust. Speech and Signal Process.*, pp. 8614–8618 (2013).
29. V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Int. Conf. Mach. Learn.*, pp. 807–814 (2010).
30. N. Srivastava et al., "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014).
31. M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Eur. Conf. Comput. Vision*, pp. 818–833 (2014).
32. S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Int. Conf. Mach. Learn.*, pp. 448–456 (2015).
33. D. C. An et al., "Flexible, high performance convolutional neural networks for image classification" in *Int. Joint Conf. Artif. Intell.*, Vol. **30**, pp. 1237–1242 (2011).
34. K. He et al., "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in *IEEE Int. Conf. Comput. Vision*, pp. 1026–1034 (2015).
35. R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," in *Int. Conf. Mach. Learn. Workshop* (2015).
36. R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Conf. and Workshop Neural Inf. Process. Syst.* (2015).
37. S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Br. Mach. Vision Conf.* (2016).
38. S. Xie et al., "Aggregated residual transformations for deep neural networks," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 5987–5995 (2017).
39. C. Szegedy et al., "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Workshop Track Int. Conf. Learn. Represent.* (2016).
40. G. Huang et al., "Deep networks with stochastic depth," in *Eur. Conf. Comput. Vision*, pp. 646–661 (2016).
41. A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Int. Conf. Mach. Learn.* (2013).
42. R. Girshick, "Fast R-CNN," in *IEEE Int. Conf. Comput. Vision*, pp. 1440–1448 (2015).
43. K. He et al., "Mask R-CNN," in *IEEE Int. Conf. Comput. Vision*, pp. 2980–2988 (2017).
44. J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv: 1607.06450 (2016).
45. D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: the missing ingredient for fast stylization," arXiv: 1607.08022v3 (2016).
46. Y. Wu and K. He, "Group normalization," in *IEEE Conf. Comput. Vision and Pattern Recognit.* (2018).
47. I. Sutskever et al., "On the importance of initialization and momentum in deep learning," in *Int. Conf. Mach. Learn.*, Vol. **28**, pp. 1139–1147 (2013).
48. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv:1412.6980 (2014).
49. S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *Int. Conf. Learn. Represent.* (2018).
50. A. Berg, J. Deng, and L. Fei-Fei, "Large scale visual recognition challenge 2010," 2012, http://www.imagenet.org/challenges
51. K. He et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(9), 1904–1916 (2015).
52. J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conf. Comput. Vision and Pattern Recognit.* (2018).

Wei Wang received his BS, MS, and PhD degrees in information and communication engineering from the National University of Defense Technology, China, in 1997, 2003, and 2010, respectively. He is currently a professor at the College of Computer and Communication, Changsha University of Science and Technology,

China. His research interests include signal processing, computer vision, and pattern recognition.

Yujing Yang received his BS degree in communication engineering from Changsha University of Science and Technology, China, in 2017. He is currently a postgraduate at Changsha University of Science and Technology, China. His research interests include computer vision and pattern recognition.

Xin Wang received her BS and MS degrees in information and communication engineering from Wuhan University of Technology, China, in 1998 and 2006, respectively. She is currently a lecturer at the College of Computer and Communication, Changsha University of Science and Technology, China. Her research interests include signal processing, computer vision, and pattern recognition.

Weizheng Wang received his BS degree in applied mathematics from Hunan University in 2005 and his PhD in technology of computer application from Hunan University in 2011, respectively. Presently, he is a lecturer at the College of Computer and Communication Engineering of Changsha University of Science and Technology. His research interests include built-in self-test, design for testability, low-power testing, and test generation.

Ji Li received his BS degree from Beijing Information Science and Technology University, China, in 2002, and his PhD from Wuhan University, China, in 2010, respectively. He is currently a lecturer at the College of Computer and Communication, Changsha University of Science and Technology, China. His research interests include signal processing, computer vision, and pattern recognition.