Artificial Neural Network Models for Image Recognition

Junyuan Cao

College of Information Engineering, Shanghai Maritime University, Shanghai, China

202110320144@stu.shmtu.edu.cn

Abstract. Image recognition is an important application of artificial intelligence. Due to the continuous development of chips and algorithms, this field has made great progress in the past decade. At the same time, AI based image recognition has been widely used in the emerging fields like robotics, autonomous vehicles, and surveillance cameras, and their demand for AI has promoted the development of image recognition technology. Resent research has found that the convolutional neural network model is particularly effective for image classification and detection and smaller convolution kernels with deeper network structures are conducive to improving the accuracy. However, problems such as overfitting and activation function gradient descent need to be solved during the operation process. The latest convolutional neural network model ResNet applies residual units to reduce the redundant calculations and improve the efficiency of the model. In general, different variants of convolutional neural network structures have different effects on image recognition, but regional convolutional neural network structures are preferred in engineering applications for its balance between processing speed and recognition accuracy.

Keywords: Convolutional neural network, deep learning, computer vision.

1. Introduction

Image recognition is an important application of artificial intelligence. Due to the continuous development of chips and algorithms, this field has made great progress in the past decade. For example, the AlexNet model that appeared in 2012 has reached an unprecedented level of recognition rate in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), and has made a breakthrough in the field of AI image recognition. Table 1 shows the fact that with the development of AI chips like ASICs and AI accelerators such as GPU, FPGAs, and CGRAs has enabled some artificial neural network models that once relied on a large amount of computing power to run successfully in recent years. According to a paper published in 2008 [1], kernel methods were more popular than neural networks in the domain of machine learning at that decade. As demonstrated in the paper [2], the increase in chip computing power has shifted the technical route of image recognition from nuclear methods to neural networks, especially with the help of FPGA chips.

Table 1. Comparison of training data sets and chip cache and computing power in different eras

	1970s	1980s	1990s	2000s	2010s	2020s
Data Size (Samples)	10^2	10^3	10^4	10^7	10^{10}	10^{12}
RAM Size (B)	10^3	10^5	10^7	10^8	10^9	10^{11}
CPU Frequency (Hz)	10^5	10^6	10^7	10^9	10^{11}	10^{15}
Method	\	\	ANN	Kernal Method	Kernal Method	ANN

2. Artificial Neural Network

Artificial neural networks emulate the information processing mechanism of the human brain. When it comes to image processing, the eyes capture electromagnetic waves within the visible light frequency range, which are then transmitted to the visual cortex of the brain through nerves and processed by different cortical layers to obtain relevant information. Similar to the brain, artificial neural networks consist of an input layer, multiple hidden layers, and an output layer. Although these networks can generally fit arbitrary continuous functions with just two to three hidden layers, those with more hidden layers and parameters have greater potential. However, training such large models

requires more time, computational resources, and may result in overfitting issues. The activation function, which operates on the neurons of the artificial neural network and maps the input to the output, is crucial for nonlinearly fitting the desired function. Common activation functions include sigmoid, tanh, and rectified linear unit (ReLU). As shown in Fig. 1, the gradient of ReLU function remains constant in the non-negative interval, which can solve the problem of gradient disappearance and enable the model to maintain a stable convergence speed.

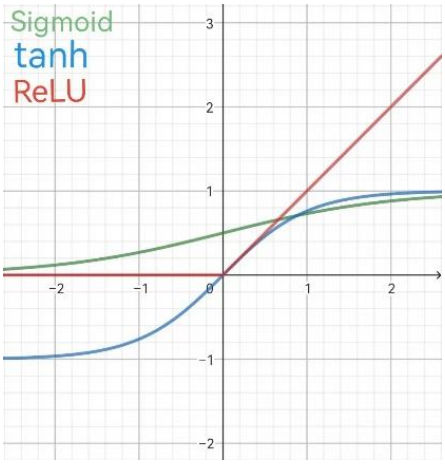

Figure 1. Images of different activation functions

3. Convolutional Neural Network Model

The convolutional neural network is a type of artificial neural network that is specifically designed for image recognition. To address the spatial invariance issue in regular artificial neural networks when processing images, convolutional neural networks employ convolution kernels to extract image feature information, reduce computation through pooling, and ultimately convert two-dimensional images into a one-dimensional array that can be understood by computers. The introduction of AlexNet in 2012 marked a significant shift in image recognition technology, from SIFT feature engineering and kernel methods to autonomous learning of image features through convolution kernels, which demonstrated unprecedented potential.

3.1. Basic Structure

The convolutional layer extract features from input data through various convolution kernels. Each neuron in the convolutional layer is connected to multiple neurons in the adjacent region of the previous layer, and each element in the convolution kernel corresponds to a weight coefficient and bias. During operation, the convolution kernel regularly scans the input features and performs matrix multiplication on the input features in the receptive field to obtain the stacked bias. The output feature map size of the convolutional layer is determined jointly by the convolution kernel size, stride, and padding. The main function of the pooling layer is to compress the feature map and extract critical features, using the pooling function to replace the output of a single point in the feature with the statistical information of its surrounding area. The pooling effect is determined by the pooling size, step size, and padding. These layers are typically inserted between convolutional layers to reduce the image size and simplify network calculations. The two most common pooling methods are average pooling and max pooling, which are suitable for different convolutional neural network structures. Max pooling is beneficial for reducing gradient descent issues, making it easier to train the model. The fully connected layer converts the result of convolutional pooling from a two-dimensional image into a one-dimensional array recognizable by a computer. It is located at the end of the convolutional neural network and produces the final classification result. In the fully connected layer, the feature map is expanded into a feature vector, and the extracted features from the previous layer are nonlinearly combined to obtain the output result.

3.2. Model History

This section will present the history of convolutional neural networks, including the LeNet, AlexNet, VGGNet, GoogLeNet (Inception V1-V4), and ResNet models. By examining the development of these models, we can understand the relationship between various subdivision technologies and the future direction of convolutional neural network models. This analysis will help to appreciate the significance of different techniques and gain insights into the trajectory of convolutional neural network research.

3.2.1 LeNet

In 1989, a convolutional neural network was developed by LeCun et al. for recognizing handwritten postal codes. The network was trained using the backpropagation method and was applied to the US Postal Service [3]. LeNet, an improved version of this network, was proposed in 1998 and achieved a remarkable 1% error rate on the zip code digit dataset. However, due to hardware limitations at the time, running large-scale convolutional neural network models was not feasible, which restricted its performance on more complex tasks and larger datasets. As a result, the success of LeNet was limited to a specific field and did not gain wider attention from academia and industry. Nevertheless, the basic structure used in LeNet is still widely used today, despite the fact that the current mainstream convolutional neural network structures differ significantly from it.

3.2.2 AlexNet

The neural network AlexNet was designed by Hinton and his student Alex Krizhevsky in the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [4]. It surpassed the results of traditional machine learning methods and emerged as the winner of the classification competition. The success of AlexNet is attributed to its deeper network structure compared to LeNet and its ability to suppress overfitting issues. This marked a significant turning point in the development of convolutional neural networks and deep learning. The basic parameters of AlexNet can be found in Table 2.

AlexNet's architecture is comparable to LeNet but has a larger scale with 5 convolutional layers and 3 fully connected layers, totaling around 60 million parameters. The network employs the ReLU function as an activation function, which achieves a lower error rate of 25% in the training cycle compared to sigmoid and tanh functions. Since the ReLU function's gradient in the non-negative interval is constant, it avoids the problem of gradient disappearance and maintains the model's convergence speed. AlexNet also uses several techniques such as local response normalization, overlapping pooling, data augmentation, and random deactivation, which have proven to be effective in enhancing recognition rates and have become standard practices in convolutional neural networks.

Table 2. Basic structure and parameters in the AlexNet model

AlexNet	Kernel	Input	Output		Kernel	Input	Output
C1	11*4*96	227*227*3	55*55*96	C5	3*3*1	13*13*384	13*13*256
P1	3*2	55*55*96	27*27*96	P3	3*2	13*13*256	6*6*256
C2	5*2*1	27*27*96	27*27*256	F1	6*1	6*6*256	1*1*4096
P2	3*2	27*27*256	13*13*256	F2	1*1	1*1*4096	1*1*4096
C3	3*1*1	13*13*256	13*13*384	F3	1*1	1*1*4096	1*1*N
C4	3*1*1	13*13*384	13*13*384				

** C for convolutional layer, P for pooling layer, and F for fully connected layer, followed by a number indicating the order of the layer.

***The kernel size is denoted by "kernel", with different meanings depending on whether the layer is a convolutional layer or a pooling layer. In the case of a convolutional layer, it refers to the size of

the convolution kernel, while in the case of a pooling layer, it refers to the size of the pooling kernel. The output categories are denoted by N.

3.2.3 VGGNet

VGGNet is a convolutional neural network developed collaboratively by the Oxford Visual Geometry Group and DeepMind researchers [5]. The goal was to investigate the correlation between the depth and performance of convolutional neural networks. To achieve this, they created a 16-19 layer deep convolutional neural network, which involved stacking small 3x3 convolution kernels and 2x2 pooling layers. In the 2014 ILSVRC, VGGNet achieved the first place in the positioning task and the second place in the classification task. VGGNet has 5 sets of convolutional layers, and each set of convolutional layers is accompanied by a maximum pooling layer. Subsequently, there are 3 consecutive fully connected layers, and finally, a Softmax layer for classification. Although the fundamental structure of VGGNet is similar to AlexNet, the model's hidden layers are increased to a greater extent. Table 3 provides an overview of the VGGNet architecture.

VGGNet differs from AlexNet by using multiple layers of 3x3 convolution kernels instead of larger kernels like 5x5 or 11x11. This change reduces computation while improving non-linear expression of the kernels, which is better for extracting small visual features from images. VGGNet uses a stackable block network structure and reuses convolution kernels of the same size to extract more complex and expressive image features. This idea of building a deep neural network model by reusing simple basic modules has become widespread since VGGNet. Although VGGNet is highly scalable and flexible, its large size results in slow computation speed.

Table 3. Basic structure and parameters in the VGGNet model

VGGNet	Kernel	Input	Output		Kernel	Input	Output
C1	3*1*1	224*224*3	224*224*64	C9	3*1*1	28*28*512	28*28*512
C2	3*1*1	224*224*64	224*224*64	C10	3*1*1	28*28*512	28*28*512
P1	2*2	224*224*64	112*112*64	P4	2*2	28*28*512	14*14*512
C3	3*1*1	112*112*64	112*112*128	C11	3*1*1	14*14*512	14*14*512
C4	3*1*1	112*112*128	112*112*128	C12	3*1*1	14*14*512	14*14*512
P2	2*2	112*112*128	56*56*128	C13	3*1*1	14*14*512	14*14*512
C5	3*1*1	56*56*128	56*56*256	P5	2*2	14*14*512	7*7*512
C6	3*1*1	56*56*256	56*56*256	F1	7*1	7*7*512	1*1*4096
C7	3*1*1	56*56*256	56*56*256	F2	1*1	1*1*4096	1*1*4096
P3	2*2	56*56*256	28*28*256	F3	1*1	1*1*4096	1*1*N
C8	3*1*1	28*28*256	28*28*512	Softmax	\	1*1*N	1*1*N

** C for convolutional layer, P for pooling layer, and F for fully connected layer, followed by a number indicating the order of the layer.

***The kernel size is denoted by "kernel", with different meanings depending on whether the layer is a convolutional layer or a pooling layer. In the case of a convolutional layer, it refers to the size of the convolution kernel, while in the case of a pooling layer, it refers to the size of the pooling kernel. The output categories are denoted by N.

3.2.4 GoogLeNet

GoogLeNet, which was released in 2014, surpassed VGGNet and won the championship of the classification task in the ILSVRC [6]. GoogLeNet uses a deeper network structure and reduces the number of parameters by compressing network connections through a more detailed network structure, which leads to a decrease in computing resources. GoogLeNet also identified that deep neural network responses often contain many redundant values that do not bring valuable information. Therefore, the connection between the input and output of an efficient network should be sparse. To achieve this, GoogLeNet introduced the inception unit, which uses a dense structure to approximate a sparse convolutional neural network. The inception unit is inspired by the Network in Network (NiN) model that was released in 2013 [7]. The main feature of the inception unit is using convolution

kernels of different sizes to process input images from multiple receptive fields in the same layer, followed by using a 1x1 convolution kernel to compress the number of channels of the input image. This compression reduces network redundancy and helps to control the number of network parameters.

The inception unit has convolutions of multiple sizes (1x1, 3x3, 5x5) and pooling operations (3x3) in one layer, which enhances the network's ability to extract feature colors of different scales by increasing the width of the network, but this significantly increases the calculation. quantity. The inception structure replaces the fully connected layer with average pooling to reduce the amount of calculation, and increases the number of channels of the 1x1 convolutional layer to compress the input image, which is called the Bottleneck Layer. Compared with VGGNet, its multi-size convolutional layer increases the network width, enhances the ability to extract different features of the image, and achieves better network performance. Inception v2 to Inception v4 have carried out a series of optimizations on GoogLeNet. Inception v2 introduced the batch normalization layer and used two cascaded 3x3 convolutions to replace the 5x5 convolutions in Inception v1, which improved the convergence of the network performance [8]. Inception v3 has been optimized in terms of reducing the size of feature maps and convolution decomposition [9]. The parallel structure of pooling and convolution has further improved the operating efficiency of GoogLeNet. The factorizing convolutions operation decomposes the n*n convolution kernel into n*1 or 1*n convolution kernels, further reducing the number of network parameters. Inception v4 uses the residual network and the Inception module to optimize the model [10].

3.2.5 ResNet

In 2016, Kaiming He presented a novel ResNet model [11] which incorporated features from VGGNet and GoogLeNet Inception V4. Similar to GoogLeNet, ResNet uses numerous 3x3 small convolution kernels. However, in the network stacking process, a residual unit is added through a short-circuit mechanism. The whole ResNet is fully convolutional except for the last fully connected layer utilized for classification, which considerably enhances computation speed. The residual unit first duplicates the input data and then performs learning on the duplicate before adding the learning result to the original input data and outputting it to the next learning unit. This structure allows the deep network to always acquire the complete information from the previous network and learn new knowledge based on this foundation. Even if new knowledge cannot be effectively learned, the performance of the deep network will not be worse than that of the shallow network by simply passing the learning results of the previous layer back. These networks are referred to as residual networks because the information mapped by the identity of the previous layer can be regarded as the learned model, and each layer only needs to learn the missing part or residual part between the existing knowledge and the ideal model. ResNet V2, an improved version of the model, was introduced soon after with more residual units and a new activation function called full pre-activation [12]. Furthermore, the ResNext model, which combined ResNet with the Inception unit from GoogLeNet, was developed in 2017 [13].

ResNet is based on the design principle of doubling the number of feature maps while reducing the size of the output feature maps by half in order to maintain the complexity of the network. Although it may seem passive, the residual learning unit is actually an effective solution to the degradation problem of deep networks, as evidenced by the improved performance of deeper networks. Thus, ResNet's design principles have been proven successful in practice.

3.3. Summary

LeNet was the original architecture for convolutional neural networks in the field of text recognition. AlexNet further developed this structure and applied it to image classification, achieving remarkable success and paving the way for subsequent models. VGGNet expanded on AlexNet by deepening the network structure and introducing a trend of using multiple small convolution kernels instead of large ones. GoogLeNet simplified the network structure through the Inception module and achieved modularization, continuously optimizing the model through version iterations. By

eliminating redundant calculations, it opened up the possibility for further improvement in network depth. ResNet, based on VGGNet and GoogLeNet Inception V4, adopted a residual unit structure that simplified unnecessary calculations and improved image information recognition. Although AI chips and computing power have improved, it is still important to consider reducing redundant calculations in models like ResNet as the convolutional neural network structure cannot be continuously deepened.

4. Convolutional Neural Networks in Different Variants

4.1. R-CNN

The convolutional neural network model has demonstrated impressive classification accuracy on the ImageNet dataset in research studies and is continuously enhanced and refined with this dataset as a benchmark. However, the performance of the model on other datasets does not match the high level of accuracy on ImageNet. To address this limitation and enhance the model's robustness while minimizing overfitting, a new model called Regional Convolutional Neural Network (R-CNN) has been proposed [14].

The R-CNN model is used for detecting objects in an image, requiring the identification of multiple possible objects and their locations. It uses a sliding window approach for positioning and applies region recognition technology to achieve accurate target detection and semantic segmentation. The model is composed of three crucial components: region proposal, feature extraction, and region classification. Region proposal generates almost a thousand regions of interest from the input image using selective search. Feature extraction extracts a 4096-dimensional feature vector from each proposal using a convolutional network that transforms the proposal into the input size of the convolutional network and performs mean subtraction. The final module, region classification, uses the strategy of non-maximum suppression to score and filter each proposed region. On the 200-class ILSVRC 2013 detection dataset, R-CNN can achieve 31.4% average accuracy. R-CNN's success is due to the effective use of convolutional networks and three efficient training steps. It has served as the basis for many subsequent object localization and detection methods.

4.2. Fast R-CNN

In 2015, R-CNN was introduced as a high-accuracy object detection model that combines convolutional neural network and region recognition technology, but its long model training time and slow detection speed were drawbacks [15]. To overcome these issues, Fast R-CNN was proposed in the same year, which introduced a region of interest (RoI) structure. RoI means that instead of performing feature extraction on the whole image, a single convolutional neural network generates a global feature map, from which each candidate area extracts its own feature code based on its coordinates. This greatly reduces computation and increases running speed. The RoI pooling layer generates a fixed-size feature map, which is then mapped into a feature vector by the fully connected layer. Each RoI produces two output vectors: softmax probability and bounding box regression offset.

4.3. Faster R-CNN

Kaiming He proposed Faster R-CNN in 2017, which improved upon R-CNN and Fast R-CNN by using Region Proposal Networks (RPN) instead of selective search [16]. The RPN predicts initial candidate regions of different sizes and shapes, known as anchors, centered on each feature point of the feature map. The RPN network uses a classification loss function to determine whether the anchor box is foreground or background and a regression loss function to determine the spatial geometric difference between the anchor box and the real target bounding box. The RPN prediction results are used to select anchor boxes with higher scores, and non-maximum suppression is applied to merge overlapping anchor boxes. The remaining anchor boxes are scored, and the highest scoring ones are selected as the final candidate regions. These regions are then passed to the R-CNN part of the

network for further classification and bounding box optimization. Faster R-CNN improves detection speed and accuracy compared to previous models.

4.4. YOLO

YOLO is a novel object detection approach that utilizes GoogLeNet as its base architecture [17]. This method treats object detection as a regression problem, in which multiple bounding boxes and associated class probabilities are predicted. YOLO performs only one forward pass through the neural network and directly predicts bounding boxes and class probabilities for the entire input image. As the entire detection pipeline is a network, it can be optimized end-to-end and runs incredibly fast. A variant of YOLO, called Fast YOLO, can process images at 155 frames per second while still maintaining high accuracy in target detection. In YOLO, the input image is divided into an N by N grid, where each grid cell is responsible for detecting an object whose center falls within that cell. YOLO then predicts the bounding box and its corresponding confidence score for each object detected.

5. Conclusion

As the level of networks has evolved from LeNet to AlexNet, VGGNet, and GoogLeNet, the performance has been improving with the deeper network. From an empirical perspective, a deeper network can extract more complex image features, which can better describe the target. However, it has been found that the network performance does not improve, but declines when the network depth reaches a certain level. The deeper the network, the greater the risk of gradient explosion or gradient disappearance. Although methods such as batch normalization have been used to control these risks, performance degradation still exists due to information distortion during the transmission process of the deep network. The use of smaller convolution kernels and maximum pooling can help reduce the problem of gradient descent. Also, different activation functions can be used between hidden layers to avoid overfitting and improve the recognition effect. Faster R-CNN and YOLO models are efficient in processing images and can be used in real-time applications such as autonomous driving. ResNet is a breakthrough model that incorporates the inception unit of GoogLeNet and relies on the residual unit structure to reduce repeated calculations, making it more in line with the human brain's image recognition process, making it currently the most potential convolutional neural network structure.

References

[1] Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola. Kernel methods in machine learning, 2008: 1171-1220.

[2] Zhichen Wang, Hengyi Li, Xuebin Yue, et al. Briefly Analysis about CNN Accelerator based on FPGA. Procedia Computer Science, 2022, 202: 277-282.

[3] Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola. Kernel methods in machine learning, 2008: 1171-1220.

[4] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

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

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

[7] Christian Szegedy, Wei Liu, Yangqing Jia, et al. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015: 1-9.

[8] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. arXiv preprint arXiv, 2013, 1312.4400.

[9] Sergey Ioffe, Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning, 2015: 448-456.

[10] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, et al. Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016: 2818-2826.

[11] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, et al. Inception-v4, inception-resnet and the impact of residual connections on learning. Proceedings of the AAAI conference on artificial intelligence, 2017, 31(1).

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016: 770-778.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. Identity mappings in deep residual networks. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14, pp. 630-645. Springer International Publishing, 2016.

[14] Saining Xie, Ross Girshick, Piotr Dollár, et al. Aggregated residual transformations for deep neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017: 1492-1500.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 2014: 580-587.

[16] Ross Girshick. Fast R-CNN. Proceedings of the IEEE international conference on computer vision, 2015: 1440-1448.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in neural information processing, 2015, 28.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, et al. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016: 779-788.