Research on Multi-Model Classification Algorithms Based on Weight Distribution (Postprint)

Authors: Jiang Mengying, Lin Xiaozhu, Ke Yan, Wei Zhanhong

Date: 2018-11-29T00:00:00+00:00

Abstract

To improve the accuracy of convolutional neural networks for image classification, this study investigates network architectures and proposes a multi-model fusion convolutional neural network. By extracting output feature vectors from individual models and fusing them to obtain new output feature vectors, a single-layer classifier is subsequently constructed for image classification, thereby enhancing classification accuracy. Comparative experiments between single models and multi-model fusion demonstrate that the multi-model fusion convolutional neural network achieves improved classification accuracy. Additionally, we analyze the weight distribution of the final fully connected layer in convolutional neural networks, revealing that the weight distribution curves of the same model on different datasets exhibit similarity, and that network models with superior classification performance possess flatter weight distribution curves.

Full Text

Preamble

Vol. 37 No. 1
Application Research of Computers (ChinaXiv Cooperative Journal)

Research on Multi-Model Classification Algorithm Based on Weight Distribution

Jiang Mengying[1,2], Lin Xiaozhu[1], Ke Yan[1,2], Wei Zhanhong[1]
[1] College of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing 102617, China
[2] College of Information Science & Technology, Beijing University of Chemical Technology, Beijing 100029, China

Abstract: To improve the accuracy of image classification by convolutional neural networks, this paper investigates network structures and proposes a multi-

model fusion convolutional neural network. By extracting output feature vectors from individual models and fusing them to obtain new output feature vectors, a single-layer classifier is then constructed for image classification, thereby improving classification accuracy. A comparison between single-model and multi-model fusion classification accuracy demonstrates that the multi-model fusion convolutional neural network achieves higher classification accuracy. Furthermore, analysis of the weight distribution in the final fully connected layer of convolutional neural networks reveals that the weight distribution curves of the same model on different datasets are similar, and that network models with better classification performance exhibit smoother weight distribution curves.

Keywords: convolutional neural network; multi-model fusion; feature vector; feature extraction; weight distribution

0 Introduction

Image feature extraction and classification have always been fundamental and important research directions in computer vision. Deep learning represents a significant breakthrough in artificial intelligence over the past decade, with convolutional neural networks (CNNs) demonstrating remarkable effectiveness in image classification. CNNs can mine abstract feature information from raw input images, yielding feature representations with strong generalization capabilities. In 1998, LeCun proposed the LeNet-5 network architecture [1], marking the true emergence of CNNs, though this early CNN was only suitable for small image recognition and performed poorly on large-scale data inputs.

In 2012, Krizhevsky et al. [2] designed the AlexNet convolutional neural network structure, which won the ImageNet Large Scale Visual Recognition Challenge (LSVRC), drawing increasing attention from researchers to CNNs. In 2014, Szegedy et al. [3] increased CNN depth by proposing the GoogLeNet model with over 20 layers. Simonyan et al. [4] investigated CNN depth and proposed the VGGNet model, increasing network depth by continuously adding 3×3 convolution layers. In 2015, He et al. [5] proposed a residual network with 152 layers, which won first place in the LSVRC-15 image classification competition. As convolutional neural network structures deepen, models require massive amounts of training data. Due to limited dataset sizes, training networks suffers from problems such as gradient dispersion and overfitting [6], leading to decreased image classification accuracy. Researchers have proposed numerous solutions to address these issues. In terms of network structure, methods such as changing activation functions [7], introducing dropout layers [8], transfer learning, parallel cross CNN models [9], and our previous work on cross-connections [10] have been employed to improve models' feature expression capabilities and thereby enhance image classification accuracy.

This paper first proposes a multi-model fusion convolutional neural network for image classification. The dataset is trained using transfer learning [11] on several

existing convolutional neural network models, and the output feature vectors from the final fully connected layer of each trained model are extracted. Feature vectors from multiple models are then fused, and the fused feature vectors undergo Softmax classification to improve classification accuracy. Experimental results validate the feasibility of this approach. Subsequently, the weight distribution curves of the final fully connected layer for both single and multi-model approaches are analyzed.

1 Multi-Model Fusion Convolutional Neural Network

Convolutional neural networks can directly input raw images, avoiding complex preprocessing and eliminating the need for manual feature extraction, thus demonstrating strong feature expression capabilities [12]. CNNs primarily consist of convolutional layers, pooling layers, local response normalization (LRN) layers [13], fully connected layers, and Softmax classification layers, with convolutional and pooling layers being essential components. The training process of CNNs 主要包括前向传播和反向传播两个过程 [14], primarily learning network parameters such as convolution kernel parameters in convolutional layers and inter-layer connection weights.

1.1.1 AlexNet Model

The AlexNet model addressed the limitation of previous CNNs that could only handle small-sized image inputs and demonstrated the effectiveness of CNNs under complex models. Compared to traditional CNNs, AlexNet introduced improvements in network architecture, activation functions, and dropout layers. AlexNet increased network depth, with deeper architecture providing better feature extraction and expression capabilities. Local response normalization performs normalization operations on adjacent regions or feature maps of feature maps, creating mutual inhibition between neighboring responses. For activation functions, AlexNet abandoned traditional sigmoid and tanh functions in favor of the ReLU piecewise linear activation function, enabling rapid network convergence, providing sparse expression capability, and effectively mitigating gradient vanishing problems. Dropout was applied in fully connected layers, where during training, some neural network units are randomly and temporarily dropped from the network with a certain probability, enhancing model generalization and improving overfitting.

1.1.2 VGGNet Model

The VGGNet model evolved from AlexNet, primarily demonstrating that network depth is a key component of algorithm performance. VGGNet introduced two main improvements: using 3×3 convolution kernels exclusively, and training and testing images across the entire picture and multi-scale. VGGNet has multiple variants based on different network depths, with VGG-16 being commonly used. Its structure is shown in [Figure 1: see original paper].

1.2 Multi-Model Fusion Convolutional Neural Network

Training CNN models often encounters overfitting problems, reducing image classification accuracy. Due to limited input images and numerous network parameters, feature vectors are prone to redundancy during classification in fully connected layers. Analysis of network model fully connected layers reveals that methods to improve overfitting and increase classification accuracy mainly involve reducing fully connected layer weight parameters and increasing output feature vectors. Based on this foundation, we propose a multi-model fusion convolutional neural network model. For the same training dataset, output feature vectors from different network models after transfer learning are extracted and fused. The fused output feature vectors then undergo classification using a newly constructed single-layer classifier, as shown in [Figure 2: see original paper].

After fusing output feature vectors from different network models, the increased output feature vectors participate in final classifier classification, providing certain improvements to classification accuracy. Meanwhile, when obtaining output feature vectors from different models, only the pre-trained models are used to fine-tune network parameters to obtain the final fully connected layer output feature vectors. After fusing different models' output feature vectors, only single-layer classifier training is required, reducing computational requirements to some extent and lowering hardware demands during training.

2 Weight Distribution in the Final CNN Layer

Through analysis of convolutional neural network structures, we observe that convolutional and pooling layers aim to extract image features, while the final fully connected layers transform extracted features into one-dimensional output feature vectors. Each unit of the final fully connected layer output feature vector can be considered as features contained in training samples, yielding a one-dimensional vector (1×n) after the final fully connected layer, where n represents the number of categories in the sample dataset. The final fully connected layer essentially performs classification on feature vectors. Research on the final fully connected layer involves obtaining the network model' s final layer weight parameters and analyzing their distribution. Assuming the output feature vector contains 4096 units, the final fully connected layer process is shown in [Figure 3: see original paper].

As shown in the figure, the fully connected layer input is {X , X , X , ···, X }, output is {a , a , ···, a }, where n represents the number of categories for the classification task, and w represents the fully connected layer weight parameters. The fully connected layer can be represented by a system of multivariate equations.

Analysis reveals that each classification category is composed of 4096 features weighted by certain proportions. During classification, the fully connected layer input multiplied by weights yields net activation, which then passes through the

ReLU activation function. Net activation values less than or equal to zero are set to zero by ReLU, retaining only values greater than zero. Therefore, only partial output features actually determine classification categories. Consequently, the weight distribution in the final fully connected layer largely determines model classification effectiveness.

3 Experiments

3.1 Experimental Datasets

This study employs the Caltech-101 dataset [15] and the 2017 Baidu Image Competition dataset for experiments. The Caltech-101 dataset is a digital image collection created by Fei-Fei Li et al. at Caltech in 2003, containing 9,146 images across 101 foreground object categories and one background category, with 30-800 images per category (most categories contain 50 images). [Figure 4: see original paper] shows sample images from Caltech-101.

The 2017 Baidu Image Competition dataset comprises 100 categories of different dog breeds, as shown in [Figure 5: see original paper]. The training set contains 8,153 images, and the test set contains 10,624 images, with 30–200 images per category.

3.2.1 Multi-Model Fusion

This study randomly selects 30 images per category from Caltech-101 as training samples, using the remainder as test samples. For the 2017 Baidu Image Competition dataset, 8,153 training images are used for training and 10,624 test images for testing. Images are first preprocessed by scaling all dataset images to 256×256 pixel patches. Since lower-level filters learned from the ImageNet dataset typically describe various local edges and texture information, these filters demonstrate good generalization for general images. When extracting final fully connected layer output vectors from different models, parameters trained on the ImageNet dataset are used as initial weights, followed by fine-tuning using the input datasets. For CaffeNet model fine-tuning and feature vector extraction, input images are processed to 227×227 pixels, and all images undergo mean subtraction. The CaffeNet model structure is similar to AlexNet, differing only in LRN layer placement: in AlexNet, the LRN layer precedes pooling layers, while in CaffeNet it follows pooling layers. For VGG-16 model fine-tuning and feature vector extraction, 224×224 pixel images are used as input. The initial learning rate is 0.0001, decreasing by a factor of 10 every 1,000 iterations, with total training iterations of 8,000. After fusing the two network models, the output vector contains 8,192 units, which then undergo single-layer Softmax classifier classification.

Using Caltech-101 dataset, classification results on the test set are shown in for: CaffeNet feature vector extraction with single-layer classifier training, VGG-16 feature vector extraction with single-layer classifier training, and fused feature vectors from both models with single-layer classifier training.

Classification accuracy of different models on Caltech-101 dataset

Using the 2017 Baidu Image Competition dataset, classification results on the test set are shown in for: CaffeNet feature vector extraction with single-layer classifier training, VGG-16 feature vector extraction with single-layer classifier training, and fused feature vectors from both models with single-layer classifier training.

Classification accuracy of different models on 2017 Baidu Image Competition dataset

Experimental results on both datasets demonstrate that for the same training sample dataset, VGG-16 model training yields higher classification accuracy than CaffeNet, and multi-model fusion achieves higher classification accuracy than single-model training.

3.2.2 Analysis of Final Fully Connected Layer Weights

From the previous experiments, trained convolutional neural network models of different structures are obtained. The final fully connected layer weights are extracted from each model and plotted in order of magnitude. The weight distributions of CaffeNet' s final fully connected layer after training on Caltech-101 and the 2017 Baidu Image Competition dataset are shown in [Figure 6: see original paper]. In [Figure 6: see original paper], the horizontal axis represents the dimension of the final fully connected layer feature vector (typically 4,096 units, ranging from 0 to 4,500), and the vertical axis represents weight parameter values (ranging from -0.06 to 0.08). Comparing weight distribution curves of the same model on different datasets reveals similar distribution patterns.

Weight distribution curves of CaffeNet and VGG-16 models on Caltech-101 dataset are plotted in [Figure 7: see original paper]. Analysis of different models on the same dataset shows that VGG-16 exhibits smoother weight distribution in its final fully connected layer compared to CaffeNet.

Weight distribution curves of the multi-model fusion CNN after training on Caltech-101 and the 2017 Baidu dataset are extracted and shown in [Figure 8: see original paper]. In [Figure 8: see original paper], the horizontal axis represents the dimension of the final fully connected layer feature vector (8,192 for the fused model, ranging from 0 to 10,000), and the vertical axis represents weight parameter values (ranging from -0.2 to 0.1). Since the 2017 Baidu Image Competition dataset is not cleaned and contains lower-quality images compared to Caltech-101, classification accuracy is lower. Comparing weight distribution curves of the multi-model fusion CNN on both datasets reveals that fused model weight distribution depends on dataset quality—the better the dataset quality, the smoother the classification curve, with better overall consistency.

4 Conclusion

Convolutional neural networks automatically and implicitly learn features without manual definition. With sufficient training samples, networks can learn excellent features for classification. However, excessive network parameters with limited training samples often lead to overfitting, reducing classification accuracy. Complex network structures also entail substantial computation and high hardware requirements. This paper first proposes a multi-model fusion convolutional neural network model, which achieves higher classification accuracy than single models. Since only single-layer network training is required after multi-model fusion, hardware requirements are reduced and computational speed is improved. Analysis of the final fully connected layer reveals that the same model exhibits similar weight distribution curves across different datasets, while different models show distinct weight distribution curves on the same dataset. Multi-model fusion weight distribution correlates with dataset quality—better quality yields smoother classification curves with greater consistency.

References

[1] Lécun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[2] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks [C]// Proc of International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012: 1097-1105.

[3] Szegedy C, Liu Wei, Jia Yangqing, et al. Going deeper with convolutions [C]// Computer Vision and Pattern Recognition. 2015: 1-9.

[4] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. Computer Science, 2014.

[5] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep Residual Learning for Image Recognition [C]// Computer Vision and Pattern Recognition. 2016.

[6] Sermanet P, Eigen D, Zhang Xiang, et al. OverFeat: Integrated recognition, localization and detection using convolutional networks [J]. Eprint Arxiv.

[7] Goodfellow I J, Wardefarley D, Mirza M, et al. Maxout networks [J]. Computer Science, 2013: 1319-1327.

[8] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[9] Tang Pengji, Wang Hanli, Zuo Lingxuan. Parallel cross deep convolution neural networks model [J]. Journal of Image and Graphics, 2016, 21(3): 339-347.

[10] Li Yong, Lin Xiaozhu, Jiang Mengying. Facial expression recognition with cross-connect LeNet-5 network [J]. Acta Automatica Sinica, 2018, 44(1): 176-

182.

[11] Pan Sinno Jalin, Yang Qiang. A Survey on Transfer Learning [J]. IEEE Trans on Knowledge & Data Engineering, 2010, 22(10): 1345-1359.

[12] Chang Liang, Deng Xiaoming, Zhou Mingquan, et al. Convolutional neural networks in image understanding [J]. Acta Automatica Sinica, 2016, 42(9): 1300-1312.

[13] Fleming A D, Philip S, Goatman K A, et al. Automated microaneurysm detection using local contrast normalization and local vessel detection [J]. IEEE Trans Med Imaging, 2006, 25(9): 1223-1232.

[14] Bouvrie J. Notes on Convolutional Neural Networks [J]. Neural Nets, 2006.

[15] Borji A, Sihite D N, Itti L. Salient Object Detection: A Benchmark [M]// Computer Vision. Berlin: Springer, 2012: 414-429.

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