Design of Lightweight Convolutional Neural Network Based on Dimensionality Reduction Module

Yue Zhou1,a, Yanyan Feng1, Shangyou Zeng1 and Bing Pan1

1College of Electronic Engineering, Guangxi Normal University, 541000 Guilin, China

Abstract. The traditional CNN extracted image features is insufficient, so the classification accuracy is not ideal. It also brought overmuch model parameters and calculation. Based on these problems, this paper proposes a dimension reduction residual module. First, reducing the dimension of the output feature map. Then, feature extraction used two different sets of convolution kernels. It can get more sufficient and different characteristic information. Two sets constitute the cascaded layer. Finally, the cascaded layer act as the input of the next layer. This module can reduce parameters. Meanwhile, it increases the depth of the network and enriches the diversity of feature acquisition. A new convolution neural network is build through this module. The performance of new network and other recognition algorithms is compared on GTSRB and 101_food datasets. The new network model is reduced to about 6.2MB, and the classification accuracy can reach 98.2% on GTSRB, 72.3% on 101_food. The experimental results show that this module can effectively improve network performance and control model size.

1. Introduction

Since Geoffrey Hinton proposed the concept of deep learning[1] in 2006, deep learning has become the hottest research field of artificial intelligence. It has made breakthroughs and solved the long-term stagnation in the field of artificial intelligence. The situation has set off an innovative revolution. In terms of recognition or natural language processing, deep learning algorithms outperform other algorithms. So far, deep learning algorithms have become the top algorithms in the field of artificial intelligence. The artificial intelligence technology represented by deep learning has begun to slowly change people's daily life. Researchers have made major breakthroughs in many aspects, such as automatic driving and real-time voice processing. It has entered a new stage.

CNN[2] is an important part of the deep learning algorithm. Up to now, various excellent models have been proposed, such as VGG[3], GoogLeNet[4], ResNet[5]. The CNN network model has been pursuing the recognition accuracy. With the development of the network, the recognition accuracy is continuously improved, and the parameter quantity is also increasing. However, the development of computer hardware cannot keep up with artificial intelligence and big data, so in this era of massive data, we need to streamline the parameters of the network, thereby reducing the requirements on computer hardware. To ease the pressure on the computer and pursue the better recognition accuracy, this paper proposes a lightweight convolutional neural network named ReduceNet, which can run on a limited-capacity hardware possibly.

In order to design a lightweight CNN with high recognition rate, we mainly made the following contributions: (1) lightweight convolution sampling module Reduce module is proposed. In the case of the same input and output, the module has fewer parameters than the standard convolution. (2) Based on the basic Reduce module,
we propose an improved module ReduceV2 module. The module has stronger expressive ability than the Reduce module. (3) Based on the proposed two modules, two lightweight and efficient convolutional neural network architectures are designed and compared, ReduceNet and ReduceV2Net.

2. Convolution Neural Network
The CNN training process is divided into forward and backward propagation. Forward propagation is used for extracting feature and obtaining image features through a series of transformations such as convolution, down sampling, and so on. Backpropagation uses the traditional BP mechanism to propagate the error forward by layer and use the chain derivation to update the convolution kernel (ie, weight and bias). In the application of machine learning, the convolutional input is usually a multi-dimensional array of data. The parameters of the convolution kernel are the multi-dimensional optimized arrays.

The excitation layer of CNN is mainly used to perform nonlinear features. The excitation function has been changing. They include the early saturated linear functions such as sigmoid, tanh and Relu[6]. The Relu solves the phenomenon of gradient dispersion to some extent. It is now widely used. In the forward calculation and back propagation calculation, it makes the network avoid the sigmoid or tanh complex exponential calculation and function derivation. Due to simple calculation, the CNN saves the computational memory and reduces the training time.

However, the work done by the incentive layer is far from enough to move the network to the hardware. We still need to do more. The related work we have done is shown below.

3. Convolution Neural Network Model Based on Dual Channels and Cascading Modules

3.1. Model Principle
The traditional convolutional neural network adopts a alternate channel between convolutional layer and downsampling layer. In the process of feature extraction, model parameters and computational complexity are large. Due to the unity of the channel, the features acquisition are not sufficient. Figure 1 shows the standard convolution structure.

![Figure 1. Tradictional 3*3 convolution kernels convolution structure](image)

The Reduce module is the core building block of the ReduceNet model. Divide a convolutional layer into a dimension reduction layer and a sampling layer, each with a ReLU activation layer. The dimension reduction layer uses the 1x1 convolution kernel to reduce the number of the input feature map. The sampling layer uses two c (G1 and G2) to extract features. G1 uses a layer of 3*3 convolution kernels. G2 uses two layers of 3*3 convolutions. The Reduce module is shown in Figure 2.

![Figure 2. Reduce model](image)
The Reduce module adopts two important strategies: (1) Using a 1×1 compressed convolution layer, reduces the number of input, which can greatly reduce the convolution computational complexity and parameters. (2) Feature extraction use two different scales of convolutions. The traditional network is basically a stack of convolution layers. VGG used large number of 3×3 convolution layers. Although its accuracy is improved, there are more parameters. In fact, the feature maps can use multiple and different sizes convolution kernels to obtain different scale features. Then combine these features to get better features[7]. By applying these two strategies, ReduceNet reduces the model size while ensuring network accuracy.

The Reduce module is explained by 128 feature map inputs. Firstly, 128 input are reduced to 64 by 64 1*1 convolutional. Then, the layer uses dual channel extraction feature. One channel uses 64 3*3 convolution kernels. The other channel of two 16 3*3 convolutional layers. Lastly, the output is the layer after the cascade of two channels. The output is the input of the next layer. When the input and output are 128 feature maps, the traditional 3*3 convolution parameter is 128*3*3*128=147456. The Reduce module parameters are 128*1*1*64+64*3*3*64+64*3*3*64+64*3*3*64=118784. Compared with the parameters, the Reduce module is less.

In the traditional CNN, the areas of the Pooling are not overlapping. Non-overlapping ignores the neighboring pixels influence on the features, which will cause the network accuracy to decrease. Overlapping Pooling can alleviate the over-fitting to some extent. However, overlapping structures may introduced noise. If Max Pooling is too much overlap, there is a greater possibility of amplifying the noise. If adding the Avg Pooling, it can reduce the noise introduced. Therefore, ReduceNet uses the overlapping Max+Max+Max+Avg in the Pooling layer.

The structure of ReduceNet is the alternate stack of the Reduce module and the downsampling layer, inserting BatchNorm layer[8] and ReLU active layer between each layer, as shown in Figure 3. Most of the parameters come from the fully connected layer. It results in over-fitting. In order to prevent over-fitting and achieve lightweight, ReduceNet uses the global average pooling instead of the fully-connected layer [9].

![Figure 3. ReduceNet](image)

3.2. Deformation Model

As shown in Figure 4, Suppose the input of the layer is x, the output is F(x), and the activation values of the upper and lower layer are added as H(x), ie H(x) = F(x) + x. The output H(x) is then input to the next layer. Add an identity mission, convert the original function H(x) to F(x)+x. This connection structure is called shortcut connection. It decompose a problem into multiple scale direct residual problems. It can optimize the training effect. Furthermore, this simple addition does not add extra parameters and calculations, but it can greatly increase the training speed of the model and improve the training effect.

The traditional CNN increases the depth of the network by stacking convolutional layers, thereby improving the classification accuracy. However, excessive convolution layers can cause gradient dispersion problems. It is impossible to effectively update the gradient to the previous layer, so that parameters cannot be updated.
When the number of layers is deepened, the shortcut connection can solve the degradation problem well. It can better spread the gradient to a shallower level in the backpropagation process. Thus, it solves the gradient dispersion problem of traditional CNN. The shortcut connection is added between the input and output. When the Reduce module gets shortcut connection, it becomes the deformed structure named ReduceV2 module, as shown in Figure 5. Then, the ReduceV2 network is designed with the ReduceV2 module.

4. Experiment

4.1. Experiment Setup
This experiment was conducted using the caffe[10] of deep learning framework. The computer was configured as an i7-6700K quad-core CPU, Ubuntu 14.04 operating system, 32 GB memory, and NVIDIA-GTX1070 GPU. Experimental images come from the currently widely used 101_food dataset and GTSRB dataset.

The 101_food dataset has a total of 101,000 sheets, 101 food images, and 1000 images for each type of food. According to 3:1, it is divided into training set and test set. As shown in Figure 6.

The GTSRB dataset is a traffic sign dataset with a total of 51,831, 43 traffic signs. The training set contains 39,209 images and 12,432 test sets. All images were taken in a natural environment. As shown in Figure 7.

When networks train on the GTSRB dataset, the solver file is configured with an initial learning rate of 0.005, a learning rate change of multistep, a gamma of 0.1, a stepvalue of 24000 and 48000, a maximum number of iterations of 60000, and a test of 500 times per training. When networks train on the 101_food dataset, the solver file was configured with an initial learning rate of 0.005, a learning rate change of multistep, a gamma of 0.1, a stepvalue of 40000, 80000, and 120000, a maximum number of iterations of 150000, and a test of 500 times per training.

Data enhancement, normalization, and de-averaging of all images prior to training[11]. All images are converted to 256*256 specifications, and the average value of the training images is subtracted for each pixel of the image. All images are randomly cropped to a size of 227*227 in five orientations (upper left, upper right, lower left and lower right and middle) and horizontally flipped to increase the amount of data.

4.2. Experimental Results and Analysis
This experimental evaluation uses the ReduceNet, ReduceV2Net, the traditional CNN (TraNet) and others published identification methods to identify the performance on the GTSRB and 101_food datasets. Through the experimental data, Table 1 and Table 2 are obtained, which are the accuracy and model size obtained by each network.
Table 1 Performance of each model on 101_food

| Network model    | Accuracy(%) | Model size(MB) |
|------------------|-------------|----------------|
| TraNet           | 53.1        | 78.9           |
| AlexNet          | 56.7        | 234.8          |
| VGG[12]          | 59.3        | 553.6          |
| ResNet[12]       | 67.4        | 95.1           |
| DCNN[13]         | 68.4        | __             |
| FOOD-DCNN[13]    | 70.4        | __             |
| Squeezenet[13]   | 52.3        | 3.2            |
| ReduceNet        | 71.0        | 6.4            |
| ReduceV2Net      | 72.3        | 6.4            |

Table 2 Performance of each model on GTSRB

| Network model    | Accuracy(%) | Model size(MB) |
|------------------|-------------|----------------|
| LeNet-5          | 87          | __             |
| SqueeNet         | 94.2        | 3.0            |
| AlexNet          | 95.9        | 233.8          |
| TraNet           | 96.6        | 77.9           |
| GoogLeNet        | 96.5        | 41.8           |
| ReduceNet        | 98.2        | 6.3            |
| ReduceV2Net      | 98.2        | 6.3            |

Figure 8 shows the variation of the test accuracy of the two networks during 101_food training. When the maximum number of iterations set to 15000 is reached, the classification accuracy of ReduceNet and ReduceV2Net is still significantly higher than other methods. Compared with the SqueezeNet model, the model size has increased, but the accuracy has increased by about 19%.

Figure 9 shows the variation of the test accuracy of the two networks during GTSRB training. It can be seen that when the maximum number of iterations is reached, the classification accuracy of two network are better than other method. The classification accuracy of two network can reach 98.2%.

In terms of accuracy, the ReduceNet and ReduceV2Net has better performance than the others methods. In terms of model size, the ReduceNet and ReduceV2Net is more lightweight. Compared with the SqueezeNet model, although the model size has increased, the accuracy of model classification has been greatly improved. It can be seen from the experimental results that compared with other network models, the classification performance of ReduceNet and ReduceV2Net is better and lightweight. So they are more convenient for network hardware application and bring better performance than others published identification methods.

Combining two comparative experiments can lead to the below conclusions:

1. This module can effectively improve network performance and control model size.
2. Dual-channel feature extraction contributes to the overall recognition rate.
3. The dimensional reduction of the 1*1 convolutional can effectively control the network parameters.
4. Using global pooling instead of the traditional fully connected layer can effectively reduce network parameters.

5. Conclusion

Based on the Reduce module, we proposed an more optimized ReduceV2 module. The module is used to design the lightweight ReduceV2Net. The experiment found that the ReduceV2 module has stronger
performance in classification. To verify the effectiveness of two networks, we conducted experiments. Experimental data proves that the ReduceNet and ReduceV2Net have a good effect in reducing the model size and improving the classification accuracy. They are convolutional neural network model with less memory requirements and convenient portability. The latter work is to continue to improve on this module, further test the performance of the module in other fields, such as target detection, image segmentation, etc. Test and optimize on larger dataset to compare the performance of the network.

References
[1] Hinton G E, Salakhutdinov R R. Reducing the Dimensionality of Data with Neural Networks[J]. Science, 2006, 313(5786):504.
[2] LECUNY, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceeding of the IEEE, 1998, 86(11):2278-2324.
[3] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
[4] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2015:1-9.
[5] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2016:770-778.
[6] V. Nair, G. E. Hinton. Rectified linear units improve restricted boltzmann machines[C]. In Proc. 27th International Conference on Machine Learning, 2010.
[7] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C] //Proc of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2818-2826.
[8] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. Computer Science, 2015.
[9] Lin M, Chen Q, Yan S. Network In Network[J]. Computer Science, 2013.
[10] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional Architecture for Fast Feature Embedding[C]// ACM International Conference on Multimedia. ACM, 2014:675-678.
[11] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]// Advances in Neural Information Processing Systems. 2012: 1097-1105.
[12] Heravi E J, Aghdam H H, Puig D. An optimized convolutional neural network with bottleneck and spatial pyramid pooling layers for classification of foods[J]. Pattern Recognition Letters, 2017.
[13] Yanai K, Kawano Y. Food image recognition using deep convolutional network with pre-training and fine-tuning[C]// IEEE International Conference on Multimedia & Expo Workshops. IEEE, 2015:1-6.
[14] Iandola F N, Moskewicz M W, Ashraf K, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size[J]. 2016.