AN IMPROVED DEEP CONVOLUTIONAL NEURAL NETWORK MODEL WITH KERNEL LOSS FUNCTION IN IMAGE CLASSIFICATION

YUANTIAN XIA, JUXIANG ZHOU AND TIANWEI XU*

Key Laboratory of Education Informatization for Nationalities
Ministry of Education, Yunnan Normal University
Kunming 650500, China

WEI GAO
School of Information Science and Technology, Yunnan Normal University
Kunming 650500, China

(Communicated by Ding-Xuan Zhou)

Abstract. To further enhance the performance of the current convolutional neural network, an improved deep convolutional neural network model is shown in this paper. Different from the traditional network structure, in our proposed method the pooling layer is replaced by two continuous convolutional layers with $3 \times 3$ convolution kernel between which a dropout layer is added to reduce overfitting, and cross entropy kernel is used as loss function. Experimental results on Mnist and Cifar-10 data sets for image classification show that, compared to several classical neural networks such as Alexnet, VGGNet and GoogleNet, the improved network achieve better performance in learning efficiency and recognition accuracy at relatively shallow network depths.

1. Introduction. Image recognition and classification is an extremely important research direction in the field of computer vision, and it is also an essential foundation for image semantic annotation and segmentation. Since the gradient-based learning algorithms were proposed [12] and successfully applied to the fields of convolutional neural networks and document recognition, the rapid development of deep convolutional neural network has made great progress in image, video, voice and audio processing especially with the application of GPU computing [13]. In recent years, some classical network models have been proposed. The Alexnet neural network model was proposed by Krizhevsky et al. [11] to train a large deep convolutional neural network, and finally won the title of champion in ImageNet ILSVRC-2010 competition. Then, VGG Neural Network (shortly called VGGNet) model was proposed and won the runner-up of ImageNet challenge in 2014. Szegedy et al. [22] suggested the GoogLeNet network model improving the utilization of the computing resources inside the network, allowing the depth and width of the network to be increased, while keeping the computing efficiency constant. This network...
model won the champion in ImageNet in 2014. A ResNet model based on the residual learning framework theory was presented by He et al. [7], which can achieve higher recognition accuracy while greatly increasing the network depth. This network model won the champion in the 2015 ImageNet. Dropout theory [21] is used to solve the problem of over-fitting in network training and improve the performance of neural network in visual, speech recognition, document classification and other learning tasks. Shortly after, a series of improved networks such as DCGANs [18], SegNet [3] and so on [6, 9, 17] were improved.

With the developments and the widespread of the application along with the deep convolutional neural network, higher requirements are put forward for the computing power of hardware with the increasing number of neural network layers. More and more neural networks use GPU for network training to improve the efficiency of training. Literature [1] describes the TensorFlow interface and the implementation of the interface built on Google. The calculation represented by TensorFlow can be carried out on a variety of heterogeneous systems, allowing a large number of computing devices such as GPU to form a large-scale distributed computing system, which greatly improves the computing speed and efficiency. Courbariaux et al. [5] introduces a method for training binomized neural networks (BNNs), which can achieve 7 times the speed without any loss of accuracy compared with unoptimized GPU cores.

Kernel methods have been developed as powerful tools for machine learning, statistical analysis and probability numeral calculations [16], [8]. In the recent years, scientists have paid large attention to kernel methods to study various learning frameworks, such as extreme learning [19], deep learning [2], Bayesian learning [10] and others [14], [4], [15]. We attempt to extend the kernel method to find a suitable loss function for CNN problem. The basic idea is that cross entropy characterizes the distance between the actual output probability and the expected output probability, that is, the smaller the value of the cross entropy is, the closer the two probability distributions are. In neural networks, we hope that the distribution of the predicted data learned on the training data is as close to the distribution of the real data as possible, so choosing the cross entropy function as the loss function can better achieve the classification of the neural network. We observe that a kernel with cross entropy characters is actually a collocation trick for CNN applications.

Aiming at the problems of low learning efficiency and low recognition accuracy in the traditional convolutional neural network, this paper proposes an improved convolutional neural network. Compared with other network structures, it has the following characteristics:

(1) The pooling layer used in the traditional network is completely removed to better preserve the integrity of image features.

(2) The pooling layer is replaced by two continuous convolutional layers with convolution kernel size of 3 × 3 and a stride of 2, while keeping the image features intact, as few training parameters as possible.

(3) Add a dropout layer between two consecutive convolutional layers to better prevent overfitting to the training data.

(4) The role of the kernel function is to change two linearly indivisible points in a low-dimensional space into linearly separable. In this work, we use cross entropy kernel as loss function in CNN to calculate the error between the predicted vector and the real vector.
The validity and accuracy of proposed model was tested by using Mnist and cifar-10 data sets for image classification. The experiments show that the neural network constructed by the proposed structure achieves higher recognition accuracy and higher convergence speed than the traditional network model.

2. The improved network structure. The traditional convolutional neural network (CNN) is generally divided into three parts: a convolutional layer, a pooling layer, and a fully connected layer. The core layer of CNN is the convolution layer and the convolution is part of the essence. In a convolutional neural network, both the input and the convolution kernel are tensors. The convolution kernels are also called weight filters, which can detect the horizontal edges, vertical edges of the image, and enhance the weight of the central area of the image. Indeed, CNN aroused widespread attention from both theoretical and applied aspects. Zhou [23] pointed that a CNN is universal which means it can be used to approximate any continuous function to an arbitrary accuracy if the depth of the neural network is large enough. Zhou [24] presented that the downsampled deep CNN can be applied to approximate ridge functions nicely, and the output of any multi-layer fully-connected neural network can be realized by that of a downsampled deep CNN with free parameters of the same order.

For a convolutional neural network, the choice of the size of the convolution kernel has a crucial impact on the recognition rate of the entire network., Simonyan et al. [20] studied the effect of the depth of the convolutional network on the accuracy of large-scale image recognition, and used the convolutional layer with convolution kernel size of $3 \times 3$ instead of the $5 \times 5$ and $7 \times 7$ large convolution kernels in neural network. From Figure 1, it is not difficult to find that the field of view observed by two $3 \times 3$ convolution kernels is equivalent to the field of view observed by a $5 \times 5$ convolution kernel, and the field of view observed by three $3 \times 3$ convolution kernels is equivalent to one field of view observed by a $7 \times 7$ convolution kernel. Therefore, the use of multiple small convolution kernels instead of a single large convolution kernel can increase the nonlinear capability of the network and effectively reduce the number of parameters without affecting the perception field. Hence, the convolution kernel in this work will also adopt the small convolution kernel with the size of $3 \times 3$.

![Figure 1. Mini-network replacing the 3 x 3 convolutions](image)

Another important concept in the convolutional layer is the stride. The stride is the position interval of the applied filter. Its size determines the size of the output image, as shown in formula (1).

$$OH = \frac{H + 2P - FH}{S} + 1, OW = \frac{W + 2P - FW}{S} + 1.$$  \hspace{1cm} (1)
where \((H,W)\), \((FH,FW)\) and \((OH,OW)\) denote the input size, filter size and output size respectively, and \(P\) and \(S\) express the padding and step size respectively. Figure 2 describes a convolution operation with a convolution kernel size of \(3 \times 3\) and a stride of 2.

![Figure 2. Kernel size: 3x3, stride: 2](image)

Pooling layer is another important part of the traditional CNN structure. Unlike the convolutional layer, no parameters need to learn in pooling layer, and its function is simply to take the maximum (or average) value from the target region. Figure 3 illustrates the operation of the pooling layer by taking the maximum pool as an example.

![Figure 3. Max pooling operation, kernel size: 4x4, stride: 2](image)

Since there is no parameter in the pooling layer, the channels of input data and output data will not change after the pooling operation. It has the characteristics of robustness to small deviations of the input data, and is widely used in convolutional neural networks. But it is why many details are lost after the pooling operation, although the largest features are worth retaining. Maybe these discarded details are also crucial to the recognition rate of the network. Figure 2 indicates that the convolutional layer with a stride of 2 can also reduce the spatial operations in the high and long directions, and retain the detailed features of the image while retaining the main features. Therefore, we completely remove the pooling layer, and replace the pooling layer in the traditional neural network with two continuous convolutional layers with convolution kernel size of \(3 \times 3\) and a stride of 2.

Although the learning parameters of the convolutional layer increase the computation of the whole network, the input data can be reduced to 1/2 of a single pooling layer by superimposing two continuous convolutional layers with a stride of 2, while retaining the image details. So the training time of the entire network is reduced, and good results have been achieved in terms of time efficiency.
The theoretical basis of using convolution layer with a stride of 2 to replace pooling layer can be described as follows:

\[
s_{i,j,u}(f) = \left( \sum_{h=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{w=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} |f_{g(h,w,i,j,u)}|^p \right)^{1/p}. \tag{2}
\]

\[
c_{i,o}(f) = \sigma \left( \sum_{h=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{w=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{u=1}^{N} \theta_{h,w,u,o} \cdot f_{g(h,w,i,j,u)} \right).
\tag{3}
\]

\[
\sigma(x) = \max(x, 0). \tag{4}
\]

where (2) and (3) express the operation of the pooling layer and convolution layer respectively, and the output characteristic of the convolution layer with linear activation function ReLU is presented by (4). Here function \( g(h, w, i, j, u) \) mapping to \( f \) and \( p \) is the magnitude of the \( p \)-norm, where \( f \) is the feature image of a layer of the convolutional neural network, \( \theta \) is the weight matrix of the convolutional layer, and \( \sigma \) is the activation function. Both the pooling and convolution operations depend on the same elements of the previous layer feature map, and (2) is the most commonly used maximum pooling operation when \( p \to \infty \). The pooling layer can be viewed as a convolutional layer using the \( p \)-norm as the activation function which provides a theoretical basis for the convolutional layer to replace the pooling layer from the perspective of mathematical expressions.

In addition, dropout is one of the strong and effective approach to effectively suppress the overfitting of the neural network to the training data. Its main idea is to randomly selected and deleted neurons in the hidden layer during neural network learning. The deleted neurons no longer perform signal transmission. When the entire neural network is tested by using test data, all neuron signals are transmitted, and the number of neurons is multiplied by the deletion ratio during training and then output. Dropout’s workflow is shown in Figure 4.

The training data is always limited in actual training, which results in the phenomenon that the training data is easily over-fitted in the neural network, and the generalization ability of the entire neural network is greatly weakened. Therefore, in the network structure designed in this paper, between every two consecutive convolutional layers a dropout layer is added. Experiments on the Mnist and Cifar-10
Datasets show that our designed network can well suppress the phenomenon of overfitting and achieve good generalization ability, while improving the performance of the entire network ability and further improves the accuracy of recognition.

Combining the advantages of the traditional convolutional neural network and improving its existing problems, we design a convolutional neural network with an improved structure. Figure 5 shows the two-layer improved network structure size of the Cifar-10 dataset as an example.

As shown in Figure 5, the biggest improvement of the traditional neural network in proposed model is to completely remove the pooling layer in the traditional neural network, and replace it with two continuous convolutional layers with convolution kernel size of $3 \times 3$ and a stride of 2, and add a dropout layer between every two convolutional layers. The use of small size convolution kernel can further reduce the number of parameters in the whole network while keeping the perception field unchanged. Two continuous convolutional layers with a stride of 2 were selected to replace the pooling layer, so that while ensuring that the main and detailed features of the image are not lost, the computation of the whole network is further reduced, and the training time is reduced while improving the recognition accuracy. Through experimental comparison with classic convolutional neural network models, the network structure designed in this paper can achieve higher recognition accuracy with higher convergence speed, which fully proves the advantages of the network model designed in this paper. Because the network structure designed in this paper has a clear hierarchical structure, when facing different data sets, you can flexibly increase or decrease the number of layers to obtain the best training model.
No matter what the structure of the network model is, its purpose is to find the best parameters to fit the training data and test data. In the neural network, this process takes the loss function as the performance index, continuously optimizes the parameter value through the gradient descent algorithm, and looks for the parameter with the minimum value of the loss function as the optimal parameter. In this paper, cross entropy kernel in formula (5) is selected as the loss function which is stated as

\[ k(t, y) = - \sum_i t_i \log y_i. \] (5)

where \( t_i \) is the \( i \)-th label value, and \( y_i \) is the output value of the \( i \)-th neuron that is the predicted value. In light of \( k(t, y) \), the loss of the label value and the predicted value will be obtained, the gradient descent procedure gradually reduce the loss and finally obtain the optimal parameters. With the cross entropy kernel, the neural network need not calculate the complicated nonlinear updates, which greatly reduces the computational complexity, the computation and the storage space.

Now, we explain why we use cross entropy kernel in our developed learning framework. Since the parameter update during the training of the convolutional neural network is achieved by the error back propagation trick, when using other loss functions to transfer the gradient to the weight of the last layer, it will inevitably occurring an association with the activation function in the last layer. For instance, if we use square function as the loss function, then derivative formula of the back propagation can be stated as

\[
\frac{\partial L(y, a)}{\partial w} = -|y - \sigma(z)| \sigma'(z)x, \quad \frac{\partial L(y, a)}{\partial b} = -|y - \sigma(z)| \sigma'(z).
\] (7)

Here \( a \) is the predicted value, \( y \) is the real value, \( \sigma'(z) \) is the derivative of the activation function of the last layer, \( x \) is the initial input, \( w \) is the weight parameter, and \( b \) is the offset. It implies that when the square loss function is used as the loss function, the gradient is proportional to the derivative of the activation function of the last layer. Therefore, the stability of the activation function can directly affect the value of the loss function and thus bring the bad effect on stabilize the entire network model.

Conversely, if we use cross entropy kernel as the loss function, the final gradient derivation can be formulated as follows:

\[
\frac{\partial k(y, a)}{\partial w} = x(\sigma(z) - y), \quad \frac{\partial k(y, a)}{\partial b} = \sigma(z) - y.
\] (8)

It reveals that the gradient of the weight of the last layer is no longer related to the derivative of the activation function, but only proportional to the difference between the output value and the real value which leads to faster convergence. Since the back-propagation algorithm is continuously multiplied, the update of the entire weight matrix is accelerated. Therefore, cross entropy kernel as loss function in the neural network can make the derivation simpler, and the loss value is only related to the probability of the correct category.

3. Experimental results and analysis. The software and hardware environment of this experiment is shown in Table 1.
In this paper, the improved network model is compared with Alexnet, VGGNet and GoogLeNet network models on Mnist handwritten data set and Cifar-10 data set. Mnist data set is a handwritten digital database, divided into training data set and test data set. There are 60,000 handwritten image samples in the training data set, and 10,000 handwritten image samples in the test data set, all of which are single-channel images of $28 \times 28$ size, divided into 10 categories. The Cifar-10 data set is composed of 60,000 $32 \times 32$ three-channel RGB color images, in which the training data set contains 50000 image samples and the test data set contains 10000 image samples, which are divided into 10 categories.

3.1. **Experimental network implementation.** Step 1: Load the Mnist and Cifar-10 datasets separately, because the focus of this experiment is to demonstrate the superiority of the improved network structure in this paper, rather than simply pursuing the recognition accuracy on the data set, the data sets are not enhanced, and the original data in the data set is directly loaded.

Step 2: Build the network and set the parameters. Build Alexnet, VGGNet, GoogLeNet and the improved network model of this paper. The learning rate is set to 0.0001, the number of batch samples is set to 200, the number of training times is set to 20,000, and the sample test period is set to 5000. Since the pictures contained in the Mnist dataset are all single-channel pictures of size $28 \times 28$. In other words, the training difficulty is greatly reduced, so when the two networks train the Mnist data set, the training times are set to 800 and the sample test period is set to 100. For the Mnist dataset, the input size is 784. For the Cifar-10 dataset, the input size is 3072. The number of categories is 10 and the dropout rate is set to 0.5. The rest of the parameters in the network are automatically and randomly generated by the system.

The improved network is compared with other three classical networks in Minist and cifar-10 data sets. Figure 6 shows the curve of recognition accuracy changing with training times in the comparison between Alexnet network and improved network in the data set cifar-10. Figure 7 shows the curve of recognition accuracy changing with training times in the comparison between VGGNet and improved network in the cifar-10 dataset. Figure 8 shows the curve of recognition accuracy changing with training times in the comparison between the GoogLeNet network and the improved network in the cifar-10 dataset. Figure 9 shows the curve of recognition accuracy changing with training times in the comparison between Alexnet network and improved network on Mnist dataset. Figure 10 shows the curve of recognition accuracy changing with training times in the comparison between VGGNet and improved network on Mnist dataset. Figure 11 shows the curve of recognition accuracy changing with training times in the comparison between GoogLeNet network and improved network on Mnist dataset.
Depicted from Figure 6-8, the improved network model obtained nearly 100% recognition accuracy on the training set and over 90% recognition accuracy on the test set after 20,000 trainings on the cifar-10 data set. Due to the extensive use of the dropout layer, the recognition accuracy of the network in the training set gradually increased to 100%. At the same time, the rising trend of recognition accuracy was basically synchronized with that of the test set. This shows that the improved network structure will not produce over-fitting phenomenon for the training set, and has good generalization ability. As shown in Figure 6, the recognition accuracy of the AlexNet network model in the training set is only 90%, and that of the test set is only about 75%. Compared with the improved network model, the recognition accuracy is quite different. As shown in Figure 7, in the VGGNet, although the recognition accuracy in the training set finally reached nearly 100%, but it only reached about 80% in the test set. Such a big difference in the recognition accuracy between the training set and the test set indicates that the VGGNet model produces a serious over-fitting phenomenon to the training data, which reduces the generalization ability of the network. As shown in Figure 8, both the GoogLeNet network and the improved model have 100% recognition accuracy on the training set, but the recognition accuracy on the test set is still slightly lower than the improved network model, and the improved network model is significantly higher in convergence speed than the GoogLeNet network model.

It can be seen from Figure 9-11 that although all the network models on the Mnist dataset, whether in the training set or the test set, the recognition accuracy rate can quickly reach 100%. However, through comparison of each figure, it will be found that the improved network model is better than the other three network models in terms of convergence speed.
Figure 7. The curve of recognition accuracy of VGGNet and improved network with the training times on cifar-10

Figure 8. The curve of recognition accuracy of Google network and improved network with the training times on cifar-10
Figure 9. The curve of recognition accuracy of Alexnet network and improved network with the training times on Minist.

Figure 10. The curve of recognition accuracy of VGGNet and improved network with the training times on Minist.
Figure 11. The curve of recognition accuracy of GoogleNet network and improved network with the training times on Mnist.

For easy observation, the final accuracy on Cifar-10 and Mnist of this experiment is shown in Table 2.

| Network         | Cifar          | Mnist          |
|-----------------|----------------|----------------|
| Alexnet:        | train acc:0.95 | train acc:0.98 |
|                 | test acc:0.78  | test acc:0.97  |
| VGGNet:         | train acc:0.98 | train acc:0.99 |
|                 | test acc:0.83  | test acc:0.98  |
| Google network: | train acc:1.0  | train acc:1.0  |
|                 | test acc:0.90  | test acc:1.0   |
| Improve network:| train acc:1.0  | train acc:1.0  |
|                 | test acc:0.94  | test acc:1.0   |

4. **Conclusion.** This paper describes and studies the structure of the traditional convolutional neural network model and the technique adopted, and proposes an improved network structure. Through experimental comparison with AlexNet, VGGNet and GoogLenet network models on two different data sets, it is concluded that the improved network structure in this paper has obvious advantages in terms of recognition accuracy, convergence speed and recognition stability. Two continuous convolutional layers with convolution kernel size of $3 \times 3$ and a stride of 2, are used to replace the pooling layer, which enables the whole network to retain both the main features and the details of the image, greatly improving the recognition accuracy. The extensive application of dropout layer greatly improves the generalization ability of the network, inhibits the over-fitting of training data, increases the performance ability of the network, and has a positive impact on the recognition accuracy of the network.
Acknowledgments. We thank the reviewers for their constructive comments in improving the quality of this paper.

Competing interests. We declare that none of the authors have any competing interests in the manuscript.

REFERENCES

[1] M. Abadi, A. Agarwal and P. Barham, et al., TensorFlow: Large-scale machine learning on heterogeneous distributed systems[J], arXiv preprint, arXiv:1603.04467, 2016.
[2] Z. An, S. Li and J. Wang, et al. Generalization of deep neural network for bearing fault diagnosis under different working conditions using multiple kernel method, Neurocomputing, 352 (2019), 42–53.
[3] V. Badrinarayanan, A. Kendall and R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell., 39 (2017), 2481–2495.
[4] C. K. Chui, S. B. Lin and D. X. Zhou, Deep neural networks for rotation-invariance approximation and learning, Anal. Appl., 17 (2019), 737–772.
[5] M. Courbariaux, I. Hubara and D. Soudry, et al., Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1[J], arXiv preprint, arXiv:1602.02830, 2016.
[6] K. He, X. Zhang and S. Ren, et al., Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, Proceeding of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, Publisher: IEEE, 2015, 1026–1034.
[7] K. He, X. Zhang, S. Ren, et al. Deep residual learning for image recognition, Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA, Publisher: IEEE, 2016, 770–778.
[8] M. Heydari, E. Shivanian and B. Azarnavid, et al. An iterative multistep kernel based method for nonlinear Volterra integral and integro-differential equations of fractional order, J. Comput. Appl. Math., 361 (2019), 97–112.
[9] G. Huang, Z. Liu and L. V. D. Maaten, et al., Densely connected convolutional networks, IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, Publisher: IEEE, 2017, 2261–2269.
[10] Y. Jiang, Z. Liang and H. Gao, et al., An improved constraint-based Bayesian network learning method using Gaussian kernel probability density estimator, Expert Syst. Appl., 113 (2018), 544–554.
[11] A. Krizhevsky, I. Sutskever and G. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM, 60 (2017), 84–90.
[12] Y. Lecun, L. Bottou and Y. Bengio, et al., Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86 (1998), 2278–2324.
[13] Y. Lecun, Y. Bengio and G. Hinton, Deep learning, Nature, 521 (2015), 436–444.
[14] Y. Lei and D. X. Zhou, Convergence of online mirror descent, Appl. Comput. Harmon. Anal., 48 (2020), 343–373.
[15] S. B. Lin and D. X. Zhou, Optimal learning rates for kernel partial least squares, J. Fourier Anal. Appl., 24 (2018), 908–933.
[16] J. Niu, L. X. Sun and M. Q. Xu, et al., A reproducing kernel method for solving heat conduction equations with delay, Appl. Math. Lett., 100 (2020), 106036, 7 pp.
[17] Y. Qu, L. Lin and P. Shen, et al., Joint hierarchical category structure learning and large-scale image classification, IEEE Trans. Image Process., 26 (2017), 4331–4346.
[18] A. Radford, L. Metz and S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks[J], arXiv preprint, arXiv:1511.06434, 2015.
[19] K. Roushangar and S. Shahmazi, Bed load prediction in gravel-bed rivers using wavelet kernel extreme learning machine and meta-heuristics methods, Int. J. Environ. Sci. Te., 16 (2019), 8197–8208.
[20] K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition[J], arXiv preprint, arXiv:1409.1556, 2015.
[21] N. Srivastava, G. Hinton and A. Krizhevsky, et al., Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res., 15 (2014), 1929–1958.
[22] C. Szegedy, W. Liu and Y. Jia, et al., Going deeper with convolutions, Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, Publisher: IEEE, 2015.

[23] D. X. Zhou, Universality of deep convolutional neural networks, Appl. Comput. Harmon. Anal., 48 (2020), 787–794.

[24] D. X. Zhou, Theory of deep convolutional neural networks: Downsampling, Neural Networks, 124 (2020), 319–327.

Received December 2019; revised December 2019.

E-mail address: xia.ming.ze.ok@163.com
E-mail address: zjuxiang@126.com
E-mail address: xutianwei@ynnu.edu.cn
E-mail address: gaowei@ynnu.edu.cn