International Journal of Artificial Intelligence and Neural Networks
Volume 6 : Issue 2 *[ISSN 2250-3749]*
Publication Date : 30 December 2016

Comparison of Deep Convolutional Neural Network Structures

The effect of layer counts and kernel sizes

[B. Melis ÖZYILDIRIM, Serkan KARTAL]

Abstract— Deep learning algorithms have become popular methods for pattern recognition due to their advantages over traditional methods such as providing deep representations of data, high-level semantic features. Deep convolutional neural network is one of the deep learning technique used in computer vision. Deep convolutional neural network consists of alternating convolution and pooling layers, and feedforward layers after them. It has not a fixed structure hence determining the optimal structure such as number of convolution and pooling layers, kernel size of these layers is crucial for faster and high performance implementations. Hence, in this work different convolutional neural network structures were established and tested on recognition of 28x28 MINST handwritten digits. According to the test results, kernels should cover at least 2 neighbor pixels of the current pixel from each side. Moreover, increasing the number of layers provide better results at the same time leads to decreases in the kernel size which may lead to worse performance. Hence, while the number of layers are increased, kernel size must be considered.

***Keywords*— convolutional neural network, deep learning, kernel size, convolutional layer count**

I. Introduction

Deep learning algorithms have become efficient approaches for pattern recognition and outperformed traditional methods due to their high-level semantic feature supports [2,5,6,7,8,9]. In addition, they provide deep representations of data [6]. One of these algorithms is called as Convolutional Neural Network (CNN). It is especially used in computer vision problems and performs great success on pattern recognition problems [1,6,7,9]. It consists of alternating convolution and pooling layers, and fully connected layers. While convolution layers extract features that are common in local regions (kernel) of all training images, pooling layers cumulate the features in a small kernel and provides pooled feature map. Pooling layer provides independence from exact locations of features [9]. It can be observed that the more number of convolution layers, the better features that are invariant to transformation will be provided [7]. Although there is not a fixed and optimal structure for deep CNN, there are some models that are well-known and used for various applications [9].

Buse Melis ÖZYILDIRIM Serkan KARTAL
Computer Engineering Department /Cukurova University
Turkey

In [10], a CNN model was proposed for a fixed image resolution; 224x 224. It consists of 5 convolutional layer alternating with pooling layers and 3 fully connected layers. Although it performed best in ImageNet Large Scale Visual Recognition Challenge- ILSVRC2012 competition, it has image resolution limitation and the reason behind its performance is unknown [9]. Another model, winner of the ILSVRC2013, consists of five convolutional layers and three fully connected layers [9,13]. A very deep model with small convolution filters was introduced in [11]. It consists of thirteen or fifteen convolution layers and three fully connected layers and it had come second in ILSVRC2014 [9]. In [12], GoogLeNet including twenty-one convolution layers and one fully connected layer was proposed and it won ILSVRC2014 competition [9]. As seen from previous works, different structures perform well in different competitions, there is not a standard structure for CNN. Moreover, since CNN has large computations, searching for optimal structure will also be challenging work. If some rules can be extracted from relation between structure and performances, efficient model can be easily established.

In this work, various CNN structures were tested and compared to find some rules for CNN models. Firstly, different sized filters were used. Then, number of CNN layers was increased and size of the filters were reduced. Lastly, different number of filters at each convolution layer was utilized. According to the test results, kernels should include more than 2 neighbor pixels of the current pixel. Moreover, while increasing the number of layers provide better results, decreases in the kernel size may lead to worse performance. Hence, kernel size must be considered while determining the number of layers.

II. Deep Convolutional Neural Network

CNN was introduced by Lecun in 1998 [4]. It consists of three layers named as convolution, pooling and fully connected layers. While convolution and pooling layers are alternating, fully connected layers are carried out after them [9]. An example structure of CNN is given in Figure 1.

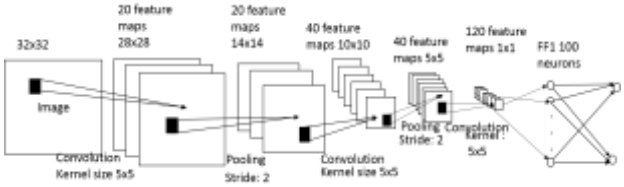

Figure 1 An example structure of CNN

In each convolutional layer, determined sized diverse filters are convolved over whole images or feature maps by windowing. It calculates inner product of kernel and every

location of the image. (1) shows the convolution process of layer L where K_L number of kernels in L, K_{L-1} number of kernels in $L-1$, x and y are indices of the height and width of each kernel in layer L, respectively. w_{size} denotes the kernel size and w is applied kernel (filter). An activation function is applied to calculated inner product such as tangent hyperbolic sigmoid, logarithmic sigmoid, linear-rectification [3].

$$\begin{aligned} &for\ (n = 0;\ n < K_L;\ n++)\\ &for\ (m = 0;\ m < K_{l-1};\ m++)\\ &\ for(y = 0;\ y < width; y++)\\ &\ for(x = 0;\ x < height; x++)\\ &for(p = 0; p < wsize; p++)\\ &\ \ for\ (q = 0;\ q < wsize\ ;\ q++)\\ &y_L(n;\ x,y) += y_{L-1}\ (m, x+p, y+q)\\ &\qquad * w\ (m\ , n;\ p, q); \end{aligned} \tag{1}$$

Convolution layers provide features invariance to the location of the object and local connectivity expressing the correlations among neighboring pixels. Moreover, weight sharing within the same feature map reduces the number of parameters to be stored and tuned [9]. After a convolutional layer, pooling layer is carried out to decrease size of the feature maps. Pooling layer also provides translation invariant features due to its neighborhood based mechanism. There are two types of pooling operators frequently used: max-pooling and average-pooling [3,9].

(2) shows max-pooling function at Lth layer with kernel size 2x2 and stride 2 where x and y denote a location on 2D pooled map [3].

$$\begin{aligned} &for(p = 0; p < stride; p++)\\ &\ \ for\ (q = 0;\ q < stride\ ;\ q++)\\ &y_L(x,y) = \max\left(y_L\ (x,y), y_{L-1}\ (x * s + p, y * s + q)\right); \end{aligned} \tag{2}$$

At last, fully connected layer produce 1D feature vector from 2D representation. Generally, the same structure of fully connected layer is utilized in studies [9].

Training of convolutional neural network has two steps as in backpropagation networks. Firstly, forward steps are calculated. Then, the prediction error is calculated. Lastly, this error is propagated back and parameter values are updated in accordance with the gradient of the error [9].

III. Test and Results

As mentioned in studies, CNN provides many advantages for computer vision problems. However, determining optimal structure is important and difficult process. Although CNN can be implemented on GPU due to its parallel structure, searching for an optimal structure will take quite a long time. Hence, in this work some structures are established and tested on MINST dataset to extract some knowledge for optimal structure.

MINST dataset includes 28x28 sized 1000 training and 1000 test handwritten digits. In this work, 1000 test values were divided into two groups including 500 validation data and 500 test data, randomly. Images are resized to 32x32 at the beginning.

Three different tests were implemented on MATLAB and CPU was utilized. In the first group of tests, only kernel sizes of convolutional layers were varied. In the second group, number of convolutional layers were increased. In the last group, for the structure providing the best results among group1 and group2 were examined with different numbers of kernels at each convolutional layers. For all tests, pooling rate and the size of pooling kernels were chosen as 2 and 2x2, respectively. For the convolutional layers, stride chosen as 1. For the all structures, at the end instead of using pooling layer, output of the convolutional layer was directly used as input of the fully connected layers. Filters were randomly initialized in the range of -1 and 1. The last layer of all structures includes 10 neurons representing digits. TABLE 1, TABLE **2**, andTABLE **3** show test results obtained for test groups 1-3, respectively.

TABLE 1 Accuracy of Test Group 1

Test 1	***Accuracy (# of correct samples / #of tests)*100***
20 different 9x9 convolutional kernels 1st convolutional layer 22 different 9x9 convolutional kernels 2nd convolutional layer 120 different 2x2 convolutional kernels 3rd convolutional layer 40 fully connected neurons	54.40%
20 different 5x5 convolutional kernels 1st convolutional layer **22 different 5x5 convolutional kernels 2nd convolutional layer** **120 different 5x5 convolutional kernels 3rd convolutional layer** **40 fully connected hidden neurons**	**78.20%**
20 different 13x13 convolutional kernels 1st convolutional layer 22 different 3x3 convolutional kernels 2nd convolutional layer 120 different 4x4 convolutional kernels 3rd convolutional layer 40 fully connected hidden neurons	71.60%
20 different 17x17 convolutional kernels 1st convolutional layer 22 different 5x5 convolutional kernels 2nd convolutional layer 120 different 2x2 convolutional kernels 3rd convolutional layer 40 fully connected hidden neurons	56.40%
20 different 7x7 convolutional kernels 1st convolutional layer 22 different 8x8 convolutional kernels 2nd convolutional layer 120 different 3x3 convolutional kernels 3rd convolutional layer 40 fully connected hidden neurons	69.20%

According to the test results given in TABLE 1 it can be seen that if the kernel size of any convolutional layer is less than 3x3, accuracy decreases. Neighboring determines the efficiency of the convolution layer. When 2x2 kernel is chosen, current cell and three neighbors of it will be considered and it is inadequate to obtain useful features. Hence, kernel should cover at least two neighbors of the current pixel from each sides. The best performance obtained from second structure given in TABLE 1 is represented as bold.

TABLE 2 Accuracy of Test Group 1

Test 2	***Accuracy (# of correct samples / #of tests)*100***
20 different 5x5 convolutional kernels 1st convolutional layer 22 different 3x3 convolutional kernels 2nd convolutional layer 40 different 3x3 convolutional kernels 3rd convolutional layer 120 different 2x2 convolutional kernels 4th convolutional layer 40 fully connected neurons	46%
20 different 3x3 convolutional kernels 1st convolutional layer 22 different 4x4 convolutional kernels 2nd convolutional layer 40 different 3x3 convolutional kernels 3rd convolutional layer 120 different 2x2 convolutional kernels 4th convolutional layer 40 fully connected neurons	52.20%
20 different 3x3 convolutional kernels 1st convolutional layer 22 different 2x2 convolutional kernels 2nd convolutional layer 40 different 2x2 convolutional kernels 3rd convolutional layer 120 different 3x3 convolutional kernels 4th convolutional layer 40 fully connected neurons	37.40%
20 different 5x5 convolutional kernels 1st convolutional layer 22 different 5x5 convolutional kernels 2nd convolutional layer 40 different 2x2 convolutional kernels 3rd convolutional layer 120 different 4x4 convolutional kernels 4th convolutional layer 40 fully connected neurons	34%
20 different 7x7 convolutional kernels 1st convolutional layer 22 different 2x2 convolutional kernels 2nd convolutional layer 40 different 3x3 convolutional kernels 3rd convolutional layer 120 different 2x2 convolutional kernels 4th convolutional layer 40 fully connected neurons	44.88%

According to the test results given in TABLE 2, accuracies of group 2 are lower than that of group 1. The main reason behind this decrease is the size of the image. Since the size of the image is small, increase in number of layers leads to decrease in kernel sizes and performance also decreased.

In test 3, the best structure among group1 and group2 is chosen and tested with different number of convolutional kernels. According to the test results, the best structure obtained from group1 and group2 is given below:

20 different 5x5 convolutional kernels 1st convolutional layer
22 different 5x5 convolutional kernels 2nd convolutional layer
120 different 5x5 convolutional kernels 3rd convolutional layer
40 fully connected hidden neurons

TABLE 3 Accuracy of Test Group 3

Test 3	***Accuracy (# of correct samples / #of tests)*100***
6 different 5x5 convolutional kernels 1st convolutional layer 16 different 5x5 convolutional kernels 2nd convolutional layer 120 different 5x5 convolutional kernels 3rd convolutional layer 84 fully connected hidden neurons	77.60%
20 different 5x5 convolutional kernels 1st convolutional layer 22 different 5x5 convolutional kernels 2nd convolutional layer 120 different 5x5 convolutional kernels 3rd convolutional layer 40 fully connected hidden neurons	78.20%
10 different 5x5 convolutional kernels 1st convolutional layer 30 different 5x5 convolutional kernels 2nd convolutional layer 90 different 5x5 convolutional kernels 3rd convolutional layer 40 fully connected hidden neurons	77.20%
30 different 5x5 convolutional kernels 1st convolutional layer 35 different 5x5 convolutional kernels 2nd convolutional layer 100 different 5x5 convolutional kernels 3rd convolutional layer 40 fully connected hidden neurons	72.20%
6 different 5x5 convolutional kernels 1st convolutional layer 16 different 5x5 convolutional kernels 2nd convolutional layer 120 different 5x5 convolutional kernels 3rd convolutional layer 40 fully connected hidden neurons	79%
120 different 5x5 convolutional kernels 1st convolutional layer 40 different 5x5 convolutional kernels 2nd convolutional layer 10 different 5x5 convolutional kernels 3rd convolutional layer 40 fully connected hidden neurons	42%

Test 3 gives information about features obtained from convolutional layers. While lower layers detect basic features, higher layers provide complex features [14]. Hence, usage of small number of kernels at lower convolutional layers will be adequate for detecting simple features. However, when feature become complex, the number of kernels should be incremented to obtain complex structures.

While training deep architectures, one of the most important issue that may arise is overfitting due to the large amount of the parameters [9]. According to the test results given in TABLE 3, fourth structure performing 72.20% includes 1050 parameters between first pooling layer and second convolutional layer and 3500 parameters between second pooling and third and it may lead to overfitting.

As shown in TABLE 3, the number of fully connected also affects the performance of the structure. Since most of the parameters are contained at fully connected layers, increase in the number of neurons located at fully connected layers may also lead to overfitting. Number of neurons at

fully connected layers must be adequate for the problem however; overfitting should also be considered.

IV. Conclusion

CNN is an efficient and frequently used structure for pattern recognition problems. However, there is not a standard structure for CNN and determining the optimal one takes quite long time due to its computational complexity. Hence, in this work different structures were established, tested for MINST dataset and compared to find some knowledge for optimal model.

According to the test results, some information can be obtained. Firstly, since CNN is based on neighboring the size of the kernel should be greater than 2x2. Although translation invariant features may be extracted when the number of convolutional layers are increased, kernel size should be considered. When the number of convolutional layer is increased, kernel size will be decreased and may lead to worse performance. Secondly, lower layers of the structure provide general features hence starting with small number of kernels may avoid overfitting. Large number of kernels at lower convolutional layers may lead to decrease in performance. On the other hand, to obtain deep features number of kernels should be increased in higher layers. Lastly, neurons at the fully connected layers also affect the efficiency. The number

of hidden neurons must be adequate in accordance with the input size however, large amount of neurons will lead to worse performance due to the overfitting.

Consequently, some information for optimal structure has been provided with this work. Repeating this kind of work for larger dataset in the future will give more general information. This work will provide at least decrease in search space of optimal structure.

References

[1] B. Leng, Y. Liu, K. Yu, X. Zhang, and Z. Xiong, "3D object understanding with 3D Convolutional Neural Networks," Information Sciences,in press.

[2] Y. Dong, Y. Liu, and S. Lian, "Automatic Age Estimation Based On Deep Learning Algorithm", Neurocomputing,187,4-10, 2016.

[3] B. Ginzburg, "Deep Learning Summer Workshop Ver. 06." http://courses.cs.tau.ac.il/Caffe_workshop/Bootcamp/pdf_lectures/Lecture%202%20Caffe%20-%20getting%20started.pdf, 2014.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, " Gradient-based learning applied to document recognition," Proceedings of the IEEE, 86(11), 2278–2324,1998.

[5] M. Liu, S. Li, S. Shan, and X. Chen, "AU-inspired Deep Networks for Facial Expression Feature Learning," Neurocomputing, 159,126-136, 2015.

[6] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A.X. Falcão, and A. Rocha, "Deep Representations for Iris, Face, and Fingerprint Spoofing Attack Detection," IEEE Transactions on Information Forensics and Security,10(4),864-879, 2015.

[7] Nogueira, R. F. (2016), Fingerprint Liveness Detection Using Convolutional Neural Networks, *IEEE Transactions on Information Forensics and Security*, 11(6),1206-1213.

[8] T. Niwa, K. Naruse, R. Ooe, M. Kinoshita, T. Mitamura, and T. Kawakami, "An Associative Memorization Architecture of Extracted Musical Features From Audio Signals by Deep Learning Architecture," Procedia Computer Science, 36, 515-522, 2014.

[9] Y. Guo, Y. Liu, Ard, O., S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," Neurocomputing, 187, 27-48, 2016.

[10] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in: Proceedings of the NIPS, 2012.

[11] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition", in: Proceedings of the ICLR, 2015.

[12] C. Szegedy, W. Liu, Y. Jia, etal., "Going deeper with convolutions," in:Proceedings of the CVPR, 2015.

[13] M.D. Zeiler and R. Fergus, "Visualizing and understanding convolutional neural networks", in: Proceedings of the ECCV, 2014.

[14] H. Lee, R. Grosse, R. Ranaganath, and A. Y. Ng, "Convolutional Deep Belief Neural Networks for Scalable Unsupervised Learning of Hierarchical Representations", in:Proceddings of the 26th International Conference on Machine Learning, Canada, 2009.

About Author (s):

She received B.E and MSc degree from Department of Computer Engineering, Cukurova University in 2010 and 2012, respectively. She received Phd degree from Department of Electrical & Electronics Engineering, Cukurova University in 2015. She works as an Assistant Professor in Department of Computer Engineering, Cukurova

He received B.E and MSc degree from Department of Computer Engineering, Cukurova University in 2010 and 2013, respectively. He continues his Phd degree at department of Computer Engineering, Cukurova University. He has been working in Department of Computer Engineering, Cukurova University since 2011.