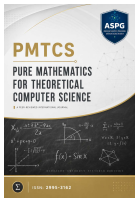
Pure Mathematics for Theoretical Computer Science

♦ ISSN: 2995-3162 Vol: 04, No. 01, 2024 ♦ PP. 07–11

Article DOI: https://doi.org/10.54216/PMTCS.040102



FULL LENGTH ARTICLE

Incorporating Kernels into Convolutional Neural Networks for Enhanced Feature Extraction

Sackineh Shamil Jasim[1,*]

[1] Department of Statistics, College of Administration and Economics, University of Karbala, Iraq

Email: sackineh.sh@uokerbala.edu.iq

Received: December 13, 2023 Revised: March 17, 2024 Accepted: May 24, 2024 ⋆ Corresponding author

ABSTRACT

The simulation was used to evaluate the method of kernel K, Neural network NN, Convolution Neural network CNN bys using (MINIST) data set. The accuracy of the method was tested and compared with the convolutional neural network as well as with the kernel function for the same input data (training and testing). The results of simulation showed that there is a high accuracy of the method, and at the same time there is a decreasing loss over the epochs, which indicates the. We note high smooth by method for recognize among features.

Keywords: Convolution ▪ MINIST ▪ kernel ▪ neural network ▪ convolution ▪ simulation

1. INTRODUCTION

Convolutional neural networks are considered a type of deep machine learning that mimics the human brain, while tradi-tional neural networks rely only on input matrices and do not use convolution on data. Convolutional neural networks have been trained on millions of images, but in the case of very large data and complex data, we need a faster time to complete the training. This network takes its idea from the communi-cation between artificial neurons and natural cells of humans or animals, which achieves greater speed and lower error rates than traditional neural networks, as it is very effective in classification. This type of uncertainty has been applied in drug discovery and in identifying potential treatments by predicting interactions between biological molecules and pro-teins. This type of network was developed and published for the first time through the discovery of handwritten numbers. Since then, this type of network has been used to read postal codes and security numbers [1].

2. CONVOLUTIONAL NEURAL NETWORK CNNS

It is termed a filtration neural network, and it is a feed forward neural network, and its function is analogous to the action of biological cells in the visual lobe of the human brain. It is a particular kind of deep learning technique that is mostly used for tasks involving object identification, such as segmentation, detection, and classification of images. CNNs provide solutions for numerous computer vision issues in artificial intelligence, including image and video processing, and are used in a wide range of real-world applications, including security camera systems and self-driving cars. The French computer scientist Yann André LeCun (1998) introduced one of the most well-known models with a real-world use [2].

2.1 CNNs Structure

The fold or convolution mathematical procedure, which is applied to two functions, f and g, yields the function o, which is a modified copy of the two functions and indicates the overlapping area between them, is the source of the network's name. Image processing is significantly impacted by this process. These layers are the outcome of applying various

© 2024 The Author(s). Published by ASPG. This article is licensed under CC BY 4.0.

filters or kernels together with the mathematical convolution or convolution process on the input elements (pixels of the input picture or neurons of the preceding layer). To generate single values, the above convolution procedure is repeated for every set of input elements with a filter size of one. We then create a new feature map for each filter by repeating all of the preceding procedures from the feature maps, and so on. These filters' values are regarded as the network's weights. The following are seen in these layers [2].

2.2 Shared Weights

Applying the filter convolution process again to all of the input parts will result in a lot more connections, but these connections will have shared weights, which will significantly boost efficiency [3]. This is one of the most crucial characteristics of this kind of network as it improves learning efficacy and qualifies the network to provide superior generalization outcomes. As an example, every resultant neuron in the feature map will be linked to the number of filter elements; however, the weights of these connections will match the weights of the neurons in the same feature map that are next to it. Since it joins neighboring points and enables the representation of regions, this approach will also guarantee the identification of image characteristics based on where they are located within the fundamental picture [4].

2.3 Duplicate Filters

Different sets of features in the image can be identified by using multiple convolutional filters with the same input because each will produce a unique feature map. However, the number of filters should be chosen so as to take into account the complexity of the calculations involved, as this depends on the quantity of training examples that are available. While a single filter's size, or its dimensions, is mostly determined by the data in the collection of training examples, the task, contrast, and dimensions of the input picture all play a role [5].

2.4 Subsampling Layers

- In the network architecture, these levels are optional, which means that their presence is not necessary. In the event that they exist, they will follow each convolutional layer and work to decrease the number of samples or neurons by reducing each set of input neurons of a certain size to a single neuron. The network architecture determines its size, and its value is the best size is 2×2, since increasing it might result in information loss. Reduction can be accomplished in a number of methods, such as [6]:
- Maximum pooling value: takes the highest value among them.
- Average pooling: takes the average of all values.
- Fully connected layers.
- After numerous layers of the previous two kinds, these layers come to link all the neurons of the preceding layer (regardless of their type) and make them input to every neuron in it, like in ordinary neural networks. They do not have to be of a certain number, but being the last layers in the network, they often have two consecutive layers since they cannot appear before a convolutional layer [7].

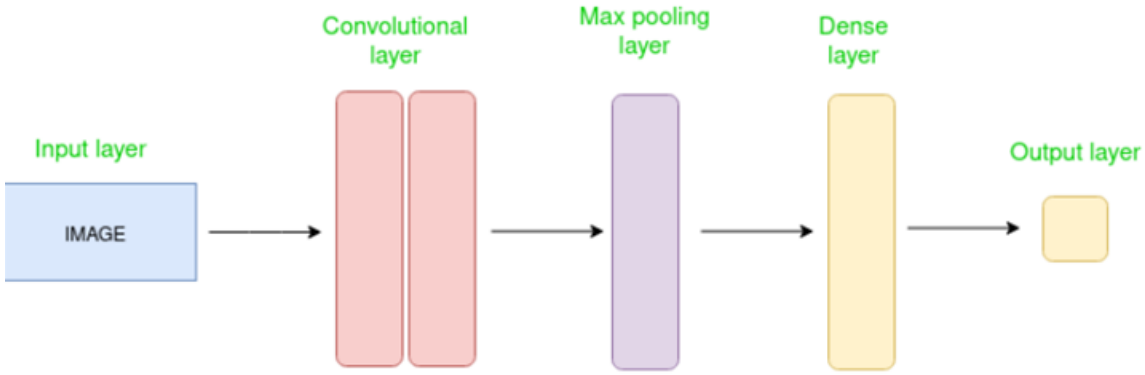


Figure 1. Structural of convolution neural net work.

Now we take a little portion of this picture and represent it vertically using a small neural network called a filter or kernel, for example, with K outputs. Upon running the neural network over the whole picture, we will get an alternate image that varies in width, height, and depth. Currently, we possess additional channels that are more slender and taller in comparison to the original R, G, and B channels. This process is often known as convolution. A neural network is considered normal when the patch size and image size are the same [8].

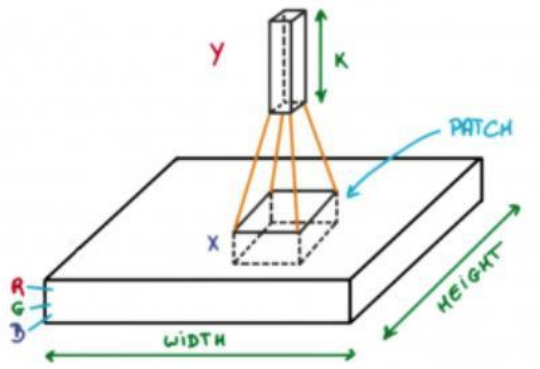


Figure 2. A perpendicular function on an CNN.

3. CONVOLUTIONAL NEURAL NETWORKS (CNNS)

This type of network was used for the purpose of distinguishing patterns and features in images and finding the characteristics of the image with high accuracy, which depends on convolutional hidden layers through which accurate discrimination is carried out according to the following steps [9, 10].

3.1 Method of Convolution

It is the method that is used to show the basic features in the data by convolution and activating the role of filters or kernel functions on the input data, according to the following mathematical formula [11]:

$$(K*Z)(i,j)=\sum_{\alpha}\sum_{\beta}K(\alpha,\beta)Z(i+\alpha,j+\beta). \tag{1}$$

Input matrix (image), K is kernel function, (i,j) the point in the input matrix, and α,β are kernel inputs.

3.2 Activation Layer

Then we apply the activation function to the model to add nonlinearity. The Gaussian function is the most often used [12].

$$f(x)=\max(0,x). \tag{2}$$

This function keeps positive values as they are and zeros out negative values.

3.3 Pooling Layer

The pooling layer reduces the dimensions of the matrix generated by the convolution layer, which reduces computational complexity and helps avoid overfitting. There are two basic types of aggregation [12].

3.3.1 Max Pooling

$$P(i,j)=\max_{(\alpha,\beta)\in R}Z(i+\alpha,j+\beta). \tag{3}$$

3.3.2 Average Pooling

$$P(i,j)=\frac{1}{|R|}\sum_{(\alpha,\beta)\in R}Z(i+\alpha,j+\beta). \tag{4}$$

3.4 Fully Connected Layers

Finally used to produce the final forecast. Each node in this layer is connected to every node in the previous layer. The mathematical equation is [13]:

$$y=W\cdot x+b. \tag{5}$$

Y output, W is a weight matrix, X is the input matrix, and B is the biased vector.

4. GENERAL APPLICATION OF THE MATHEMATICAL MODEL

A convolutional neural network consists of several successive layers of convolution, activation, and pooling, followed by fully connected layers. The final mathematical model can be summarized by the following formula [14]:

$$F(x)=f_L\big(W_L(\cdots f_2(W_2(f_1(W_1x+b_1))+b_2)\cdots)+b_L\big). \tag{6}$$

x is the initial input, W and b are the weights and biases of each layer, f is the activation function, and L is the number of layers in the network.

5. KERNELS IN CNN

Filters, also known as small matrices, are utilized in the convolution process. As the kernels move through the input data, they perform multiplication with the corresponding pixels, which produces a feature map that highlights specific patterns or traits of the input. Features in Convolutional Neural Networks (CNNs) refer to the important patterns or unique characteristics obtained from the input data through the convolutional layers. The mathematical modeling for the argument the Kernel in CNN is as follows:

$$(Z*K)(x,y)=\sum_{i=1}^{\beta-1}\sum_{j=1}^{\alpha-1}Z(x+i,y+j)K(i,j). \tag{7}$$

Z is the input image, K is the kernel, (x,y) is the position of the kernel on the image, and m and n are the kernel dimensions.

6. SIMULATION STUDY

The famous data (MINIST), which means (Modified National Institute of Standards and Technology database), was used to train and test the K (Kernel), NN (Neural Network), KCNN ((Convolution Neural Network)) method. It is one of the most widely used data in research and practical applications, which consists of an input image that includes (60,000) training images, (10,000) test images, and all image consisting of a number written by hand from (0) to (9). The image size is 28×28. It also consists of image labels, which express the number written in the image from (0) to (9). A kernel function with a size of (3×3) and (8) feature maps were used for the purpose of teaching the network on it. The accuracy of the method was tested and compared with the convolutional neural network as well as with the kernel function for the same input data (training and testing), as shown in Table 1. The algorithm iteration was (600).

Table 1. Accuracy of the methods

Method	Kernel	NN	CNN
Accuracy	0.89	0.93	0.98

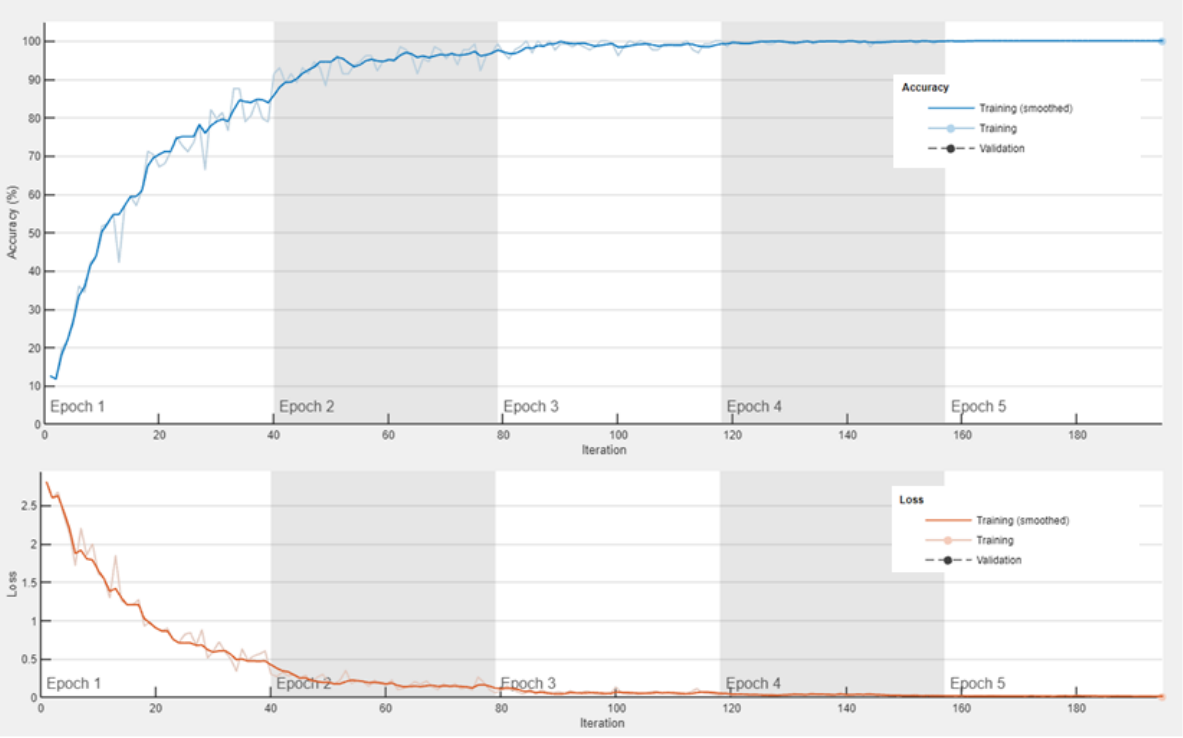


Figure 3. Simulated training method and accuracy, Loss in method.

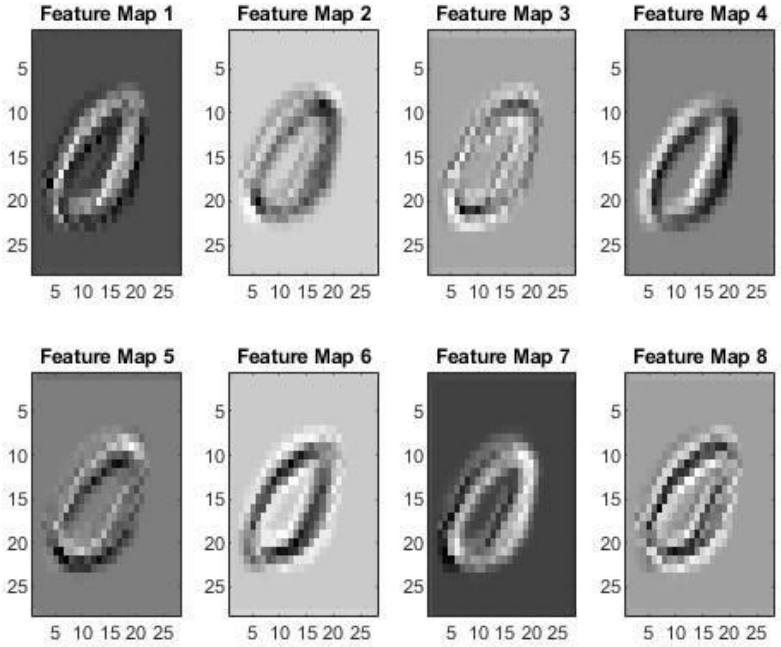


Figure 4. Output of the convolutional operation on the input images.

7. RESULT DISCUSSION

We can see from Table 1 there are high accuracy for CNN is (0.98) compared with K & NN. Figure 3 shows that there is a high accuracy of the method after the training data set is completely passed through the neural network at once over the epochs (Epoch), and at the same time there is a decreasing loss over the epochs, which indicates the accuracy of the proposed method. From Figure 4 we note high smooth by method for recognize among features.

REFERENCES

[1] J. Wu, "Introduction to convolutional neural networks," *National Key Laboratory for Novel Software Technology, Nanjing University, China*, vol. 5, p. 495, 2017.

[2] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[3] L. Anuj and M. T. Gopalakrishna, "ResNet50-YOLOv2-convolutional neural network based hybrid deep structural learning for moving vehicle tracking under occlusion," *Solid State Technology*, vol. 63, pp. 3237–3258, 2020.

[4] Q. A. Al-Haija, M. Smadi, and O. M. Al-Bataineh, "Identifying phasic dopamine releases using DarkNet-19 convolutional neural network," *Proceedings of the 2021 IEEE International IoT, Electronics and Mechatronics Conference*, pp. 1–5, 2021.

[5] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya *et al.*, "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, p. 2470, 2021.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[7] H. Zhang, "A review of convolutional neural network development in computer vision," *EAI Endorsed Transactions on Internet of Things*, vol. 7, no. 28, p. e2, 2022.

[8] X. Ji, Q. Yan, D. Huang, B. Wu, X. Xu, A. Zhang *et al.*, "Filtered selective search and evenly distributed convolutional neural networks for casting defects recognition," *Journal of Materials Processing Technology*, vol. 292, p. 117064, 2021.

[9] C.-C. J. Kuo, "Understanding convolutional neural networks with a mathematical model," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 406–413, 2016.

[10] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 6999–7019, 2021.

[11] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, G. E. Dahl, G. Saon, H. Soltau *et al.*, "Improvements to deep convolutional neural networks for LVCSR," *Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 315–320, 2013.

[12] S. Shetty, "Application of convolutional neural network for image classification on Pascal VOC Challenge 2012 dataset," *arXiv preprint arXiv:1607.03785*, 2016.

[13] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems*, vol. 28, pp. 802–810, 2015.

[14] T. Wiatowski and H. Bölcskei, "A mathematical theory of deep convolutional neural networks for feature extraction," *IEEE Transactions on Information Theory*, vol. 64, pp. 1845–1866, 2018.