EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2025, Vol. 18, Issue 1, Article Number 5707
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

Proposal for a Generalized Convolution to Mitigate Heat Generation in Convolutional Neural Networks

Hwajoon Kim[1], Eunyoung Lim[1], Byeongjae Kang[1], Sunyoung Yeun[1,*]
[1] *Kyungdong University, Kyungdong Univ. Rd. 27, Yangju, Gyeonggi, S. Korea*

Abstract. In convolutional neural networks (CNN), the problem of heat generation is becoming a significant issue. This challenge can be mitigated through both hardware and software methods. This study focuses on drastically reducing the amount of computations and, consequently, the heat generation. Specifically, the proposed approach reduces computations by approximately $m \times 2^n$, where m is the number of layers and n is the size of the node.

2020 Mathematics Subject Classifications: 68T01, 68T27, 44A35

Key Words and Phrases: Heat generation, convolutional neural networks, geralized convolution

1. Introduction

CNN can be understood as a specialized artificial intelligence framework particularly suited for processing image data. In CNN, convolution serves to transform the original image into a feature map using a matrix, while pooling acts as a form of data compression. However, thses tasks often involve unnecessary computational processes, inspiring this research to explore the possibility of eliminating such inefficiencies. Consequently, this study proposes an algorithm that integrates convolution and pooling, traditionally performed as separate tasks, into a single, simplified operation. This integration aims to mitigate the heating problem caused by extensive computations in deep learning systems. Issues related to heating problems are discussed in [4].

When CNN receives input, it generates multiple feature maps using the kernel matrix and then proceeds with classification. The kernel functions as the equivalent of weights in a neural network. Convolution, represented by a kernel matrix, operates as follows: by sweeping an $n \times n$ kernel matrix across the original image, the data is transforms into a new form. This process involves multiplying each $n \times n$ segment of the input matrix by the kernel matrix and summing all the resulting components, yielding the convolution.

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v18i1.5707

Email addresses: cellmath@gmail.com (Hj. Kim), minkimama0229@gmail.com (Ey. Lim), kbj1216@kduniv.ac.kr (Bj. Kang), ysy3324@kduniv.ac.kr (Sy. Yeun)

Deep learning concepts related to this study are discussed in [3], [2], [7], [8], [1], [9], and [10], while topic concerning integral transforms are presented in [5] and [6]. In this study, we propose a generalized convolution through straightforward examples, with the mathematical details to be elaborated on in future work.

2. Proposal for a generalized convolution to mitigate heat generation in convolutional neural networks

Existing neural networks update weights using the process

$$w_{ij}^a = w_{ij}^b - \alpha e_w,$$

where α is the learning rate, w^b represents the weight before the update, w^a denotes the weight after the update, and e_w represents the partial derivative of error e with respect to w. In CNN, convolution can be interpreted as a specific form of weight.

Definition 1. *The convolution of matrix A and matrix B is denoted by array multiplication $A \circ B$.*

The following lemma shows the relationship between convolution in mathematics and convolution in AI.

Lemma 1. *$A \circ B = tr(AB^T)$, where T is the transpose and tr is the trace.*

Proof. The detail is can be found in [2].

The second step of CNN is the pooling. This is the work of reducing size, which is to transform the image of the convolution into a small image by extracting a representative from each part. For example, in short, if we have max-pooling a matrix

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix},$$

we obtain a matrix

$$(4)$$

of 1 row and 1 column(just a simple task of drawing the largest number). This pooling is simple, but it makes a contribution to enhance performance. Because, it is the principle that the resolution is increased when the screen is reduced. [8] has shown that features are invariant in fine translation and reduce noise. The currently widely used pooling method is max-pooling, which is quite simple as the above but has a reasonable effect. Now, let us define a generalized convolution as follows. Then, this definition can provide the simplicity of integrating the dualized convolution and max-pooling into one. Without further mention, s_p, s_c, s_s and s_i refer to pooling size, convolution size, stride size, and image size, respectively.

A matrix B is called a submatrix of A if $B \subset A$ and satisfies the conditions of a matrix.

Definition 2. *Let A be an image data and Let M be a submatrix of A that contains the maximum element of A in the center. A generalized convolution incorporating convolution and max-pooling can be denoted by the maximum element of $C \circ M$, where C is a given convolution.*

Let A. C, m_b, and m_a be the given image, convolution, the maximum element before convolution, and the maximum element after convolution, respectively. Consider the submatrix $\{M|M \subset A\}$ containing the maximum element m_b. Note that $M \subseteq A$ and $s_M = s_c$. A generalized convolution incorporating convolution and max-pooling can be denoted by $m_a \in (C \circ M)$. This means the maximal element of the matrix created as a result of array multiplication of the image containing m_b and the convolution.

The significance of this definition is that there is no need to stack the convolution layer and the pooling layer as a pair, which is the conventional method. Instead, convolutional deep neural networks can be constructed using only a subset of the convolutional layers. This subset corresponds to the multiplication of maximum elements. To find the maximum element, simply select the matrix containing the largest number within $n \times n$ matrix (where n is fixed). In summary, CNN can be configured using a single task, generalized convolution, departing from the traditional approach that treats convolution and pooling as separate tasks.

Comparison of generalized CNN algorithm and existing CNN algorithm

A. Existing CNN algorithm

(1) Input
(2) Convolution
(3) Pooling
(4) Repeat (2) and (3) until the desired output is obtained.
(5) Output

B. generalized CNN algorithm

(1) Input
(2) Generalized convolution
(3) Repeat (2) until the desired output is obtained.
(4) Output

In this deep learning method, generalized convolution applies convolution only to specific nodes and does not require pooling work. This is similar to CNN applying only to specific nodes of the given image.

Example. (Generalized convolution that greatly reduces computations) When the original image is

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix},$$

the convolved feature map is

$$\begin{pmatrix} 6 & 3 & 4 \\ 2 & 4 & 3 \\ 4 & 3 & 4 \end{pmatrix}$$

for stride 1, where the kernel is

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Then if we do the max-pooling operation here, we get 6. Note that the convolution operation is used 9 times here. If $s_s = 2$, the convolved feature map changes to

$$\begin{pmatrix} 6 & 4 \\ 4 & 4 \end{pmatrix}.$$

Of course, we get 6 after max-pooling, where the calculation of convolution is used 4 times.

Next, let us consider the generalized convolution. The matrix from which the largest component can be obtained is the first 3 by 3 matrix. As a result of convolution, we get 6. This is the same as the above result. Here, note that the convolution operation was performed only once and the pooling operation was not performed. As seen above, the generalized convolution becomes more effective as the size of the kernel matrix increases.

It is only an estimate from which matrix can be obtained the largest component. This estimate is not necessarily accurate. Even a slight error will only bring about a slight difference in resolution. After choosing a matrix with large components, we just need to find the convolution. When the number of channels is multiple, the final feature map can be obtained by array addition of feature maps obtained from each channel.

Example. Consider the following image

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 4 & 1 & 0 & 0 \\ 2 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 3 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix},$$

and let us $s_s = 1$. Applying the given kernel matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix},$$

we get the feature map

$$\begin{pmatrix} 17 & 9 & 8 & 9 & 12 & 12 \\ 9 & 9 & 10 & 12 & 11 & 11 \\ 10 & 9 & 11 & 15 & 14 & 9 \\ 10 & 9 & 11 & 12 & 14 & 9 \\ 10 & 13 & 17 & 18 & 14 & 11 \\ 15 & 12 & 14 & 16 & 17 & 14 \end{pmatrix}.$$

Applying max-pooling here, we get 18 for $s_p = 6$. If the $s_p = 5$, 4, and 3, we get the matrices of

$$\begin{pmatrix} 18 & 18 \\ 18 & 18 \end{pmatrix},$$

$$\begin{pmatrix} 17 & 15 & 15 \\ 18 & 18 & 18 \\ 18 & 18 & 18 \end{pmatrix},$$

and

$$\begin{pmatrix} 17 & 15 & 15 & 15 \\ 11 & 15 & 15 & 15 \\ 17 & 18 & 18 & 18 \\ 17 & 18 & 18 & 18 \end{pmatrix},$$

respectively. The convolution operation was used $6 \times 6 = 36$ times.

In order to apply the generalized convolution here, let us select the following matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 4 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 3 & 1 & 1 \end{pmatrix}.$$

Array multiplication of the kernel matrix with this matrix gives 17. This is slightly different from the result of the traditional method, which is 18. However, as mentioned earlier, this small difference in resolution can be safely ignored. If $s_s = 2$, the convolved feature map is

$$\begin{pmatrix} 17 & 8 & 12 \\ 10 & 11 & 14 \\ 10 & 17 & 14 \end{pmatrix}.$$

The result of max pooling is 17. Here convolution is used $3 \times 3 = 9$ times. Similarly, if the $s_s = 3$, 4, 5 or 6, we also get a result of 17.

This approach reduces the computational load by $m \times 2^n$ in deep learning, where m is the number of layers and n is the size of the node. This reduction helps mitigate heat generation.

Acknowledgements

The first author (Hj. Kim) acknowledges the support of Kyungdong University Research Fund, 2024.

References

[1] Silver et. al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[2] Young Hee Geum, Arjun Kumar Rathie, and Hwajoon Kim. Matrix expression of convolution and its generalized continuous form. *Symmetry*, 12(11):1791, 2020.

[3] Daniel Graupe. *Deep learning neural networks: design and case studies*. World Scientific Publishing Company, 2016.

[4] Hwajoon Kim. The intrinsic structure and properties of laplace-typed integral transforms. *Mathematical Problems in Engineering*, 2017(1):1762729, 2017.

[5] Hwajoon Kim. Editorial for special issue "various approaches for generalized integral transforms". *Symmetry*, 16(5):576, 2024.

[6] Erwin Kreyszig, K Stroud, and G Stephenson. Advanced engineering mathematics. *Integration*, 9(4):1014, 2008.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[8] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[9] Jason Weston, Frédéric Ratle, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Proceedings of the 25th international conference on Machine learning*, pages 1168–1175, 2008.

[10] Joshua Fredrick Wiley. *R deep learning essentials*. Packt, 2016.