Minimizing Multiplication of Kernel Computation in Convolutional Neural Networks Using Strassen Algorithm

Dary Mochamad Rifqie[1], Dewi Fatmarani Surianto[2], Sudarmanto Jayanegara[3], Muhammad Fajar B[4], M Miftach Fakhri[5]
Universitas Negeri Makassar[1,2,3,4,5]
[1]dary.mochamad.rifqie@unm.ac.id
[2]dewifatmaranis@unm.ac.id
[3]sudarmanto@unm.ac.id
[4]fajarb@unm.ac.id
[5]fakhri@unm.ac.id

Abstract - Convolution neural network (CNN) have been widely applied for the computer vision task. However, the success of CNN is limited by the computational complexity of the network, so it is difficult for the model to run inference process in real-time. In this paper, we apply strassen matrix multiplication to reduce multiplications in convolution operation in CNN, in order to get faster execution for CNN. First, we transform the convolution operation into matrix multiplication operation using toeplitz mapping method, then after that we apply strassen method into these matrices. In the end, we compare the number of arithmetic operation (multiplication and addition) in convolutional layer using strassen and standard algorithm. We apply this algorithm implementation in convolution layer 1 and 3 in LeNet-5 Architecture.
Keywords: Convolutional Neural Network , Strassen Algorithm, Matrix Multiplication , Kernal Computation

I. PENDAHULUAN

Convolutional neural networks (CNN) have been applied across various applications in computer vision such as face detection and self-driving cars [1]. Nevertheless, as the accuracy of the prediction process in CNN model improves, the computational complexity of the network also increases significantly. This make it difficult for the model to run inference process in real-time because it requires low latency. The achievement of CNN is limited by the problem of their heavy computational burden.

CNN architecture have several networks layer dominated by the convolution operation [2]. In this operation, every element in the output feature is computed individually using multiply-and-accumulate (MAC) operations. These MAC operation in convolution consume much computational cost in CNN. In this study, in order to reduce the high-cost operations in the network, we apply strassen algorithm to reduce the number of convolution, especially multiplication operation.

Strassen algorithm is a matrix multiplication algorithm developed by Volker Strassen in 1969 [3]. This algorithm is based on concept divide and conquer and it is an optimal method to multiply large matrices. The naive algorithm for applying matrix multiplication of two n x n matrices needs O(n3) operations, which is computationally high-cost, especially for large matrices. Strassen algorithm enhance on this by recursively split the matrix into smaller submatrices, then performing operations on these submatrices[4].

This paper introduces a transformations of convolution operation into strassen matrix multipilication. This algorithms can minimize the multiplication operations of a convolutional layer compared to conventional convolution. By reducing arithmetic complexity, it can speed up computation in convolutional layer in CNN.

II. CONVOLUTIONAL NEURAL NETWORKS

CNN enhance the capabilities of neural networks by adding convolutional operation to the network. Figure 1 shows the architecture of the first convolutional neural network, LeNet-5 [5]. This architecture comprises of two convolutional networks, two averagepooling layers and three fully connected layers. From each the layers in CNN, convolution operation dominates in CNN computation. Figure 2 displays an implementation of convolution operation using standard algorithm. In this model, MAC operation is utilized with the data from input and kernel to compute every element in output feature.

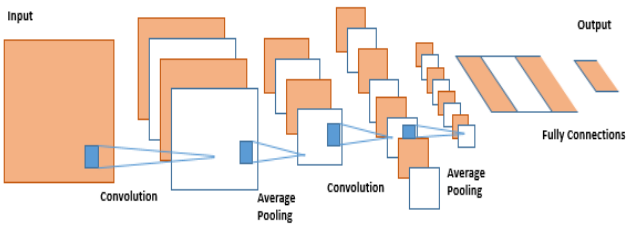

Figure 1. Lenet-5 Architecture

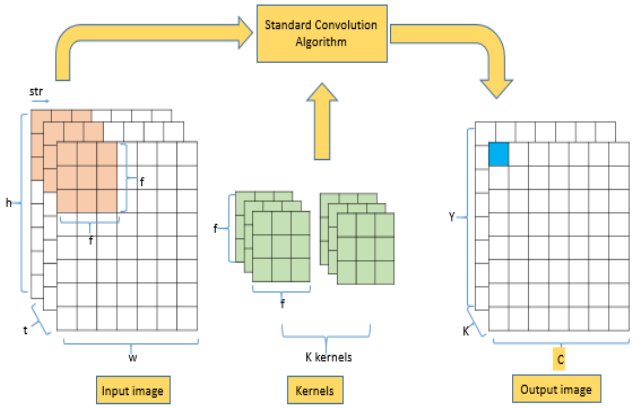

Figure 2. Standard Convolution Algorithm

III. METHOD IN STRASSEN ALGORITHM

Before hardware platform like CPU and GPU execute the kernel computation, the input images and kernel are usually mapped into matrix multiplication form. This transformation is applied using toeplitz transformation as shown in Figure 3 [6]. By transforming this convolution layer, we can implement this convolution operation using matrix multiplication. Therefore, we can utilized strassen algorithm to speed up this process. By implementing this algorithm, we can reduce the number of multiplication in CNN.

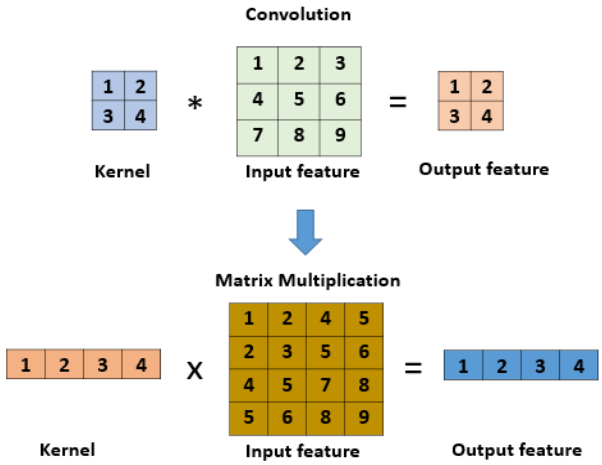

Figure 3. Toeplitz transformation in convolution operation

An illustration of the application of the Strassen algorithm to a 2 x 2 matrix multiplication is shown in the following way:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

$$AB = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

In the naive matrix multiplication above, 8 multiplications and 4 additions are produced by this algorithm. By using strassen transformation, we can converted into 7 multiplications and 18 additions along with the creation of 7 secondary values (Ki) as follows:

$$K1 = a(f - h)$$

$$K2 = (a + b)h$$

$$K3 = (c + d)e$$

$$K4 = d(g - e)$$

$$K5 = (a + d)(e + h)$$

$$K6 = (b - d)(g + h)$$

$$K7 = (a - c)(e + f)$$

The final output AB is constructed from the secondary (Ki) values as follows:

$$AB = \begin{bmatrix} K5 + K4 - K2 + K6 & K1 + K2 \\ K3 + K4 & K1 + K5 - K3 - K7 \end{bmatrix}$$

This strassen method can operate with the multiplication of non-square matrices. For non-square matrices with even dimensions, we can divide the matrices evenly into 2 x 2 matrices. For matrix with odd dimension, we can pad a row or a column that is filled with zeros so that the matrix dimension sizes become even.

In CNN operation, one of the matrices will have kernel that have a constant value across different inputs feature. By this condition, we can precalculate this using a constant matrix (kernel) to decrease the number of additions in strassen matrix multiplication [7].

IV. EXPERIMENT AND RESULT

We do an experiment to evaluate and compare the arithmetic complexity of convolutional layer in LeNet-5 architecture using strassen matrix multiplication and standard convolution. There are three convolutional networks in Lenet-5 model, each of layer having kernels of size 5x5 with 6, 16, and 120 feature maps, respectively, and all the convolutional layer apply stride 1. Moreover, The kernel and also input feature map will be converted to matrix multiplication form using Toeplitz transformation. After implementing this transformation, we can compute the convolutional layer using naive matrix multiplication and also strassen matrix multiplication, and compare the arithmetic complexity such as the quantitiy of multiplication and addition operation in both algorithm.

In this work, we evaluate the convolutional network in layer 1 and 3 to do this implementation. All of the algorithm implementation is programmed using Python programming language using Google Colaboratory IDE.

Table 1. Number of Arithmetic Operation in Convolution Layer Using Naïve algorithm.

Type of Layer	Number of Multiplication using naive algorithm	Number of Addition using naive algorithm
Convolution in layer 1	122304	117600
Convolution in layer 3	240000	238400

Table 2. Number of Arithmetic Operation in Convolution Layer Using Strassen algorithm.

Type of Layer	Number of Multiplication using strassen algorithm	Number of Addition using strassen algorithm
Convolution in layer 1	107016	198792
Convolution in layer 3	210000	390296

Table 1 and 2 show the result of the experimentation using naive and strassen algorithm. From the table, we can be clearly seen that implementation of convolution operation using strassen algorithm can reduce the number of multiplication. This result occurs in layer 1 and 3 of convolutional network in LeNet model with 5x5 kernel size. The benefits of applying Strassen algorithm increase if the size of the matrix also increase. This can be seen from the table results on layer 1 and 3 convolution operations. The size of the matrices at layer 3 is larger that layer 1. Therefore, the reduction of multiplication operations in layer 3 is also much greater.

However, in the tables also display that strassen algorithm requires more addition operations than the naïve algorithm. These additions involve operation in secondary value and also output in the final result. Optimizing number of multiplication is important because executing multiplication in a hardware platform is slower than addition operation. Therefore, this result can increase the speed of execution in CNN. Although Strassen can decrease the multiplications, there is a penalty that comes from the cost of increased memory usage requirements for secondary results and often decrease numerical stability [8].

V. CONCLUSION

In this paper, we use strassen matrix multiplication to reduce multiplications in convolution operation in CNN, in order to get faster execution for CNN. We do this algorithm implementation in convolution layer 1 and 3 in LeNet-5. First of all, we transform the convolution operation into matrix multiplication operation using toeplitz mapping method, then after that we apply strassen method into these matrices. In the next step, we compare the number of arithmetic operation using strassen and standard algorithm. The result point out that strassen method can decrease the multiplications significantly by 15288 point in layer 1 and 30000 point in layer 3. However, this algorithm have more additions operator than the standard one.

REFERENCE

[1] Barman, U., Choudhury, R.D., Sahu, D. and Barman, G.G., 2020. Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease. Computers and Electronics in Agriculture, 177, p.105661.

[2] Chishti, Syed OwaisAli, Sana Riaz, Muhammad BilalZaib, and Mohammad Nauman. "Self-driving cars using CNN and Q-learning." In 2018 IEEE 21st International Multi-Topic Conference (INMIC), pp. 1-7. IEEE, 2018.

[3] Hedtke, Ivo. "Strassen's Matrix Multiplication Algorithm for Matrices of Arbitrary Order." arXiv preprint arXiv:1007.2117 (2010).

[4] Zhao, Yulin, et al. "A faster algorithm for reducing the computational complexity of convolutional neural networks." Algorithms 11.10 (2018): 159.

[5] LeCun, Yann, et al. "Comparison of learning algorithms for handwritten digit recognition." International conference on artificial neural networks. Vol. 60. No. 1. 1995.

[6] Townsend, Alex, Marcus Webb, and Sheehan Olver. "Fast polynomial transforms based on Toeplitz and Hankel matrices." Mathematics of Computation 87.312 (2018): 1913-1934.

[7] Strassen, Volker. "Gaussian elimination is not optimal." Numerische mathematik 13.4 (1969): 354-356.

[8] WINOGRAD, Shmuel. Arithmetic complexity of computations. Siam, 1980.

[9] Cong, J., and B. Xiao. "Artificial Neural Networks and Machine Learning." Proceedings of the 24th International Conference on Artificial Neural Networks,(ICANN'14). Vol. 8681.

[10] Rifqie, Dary Mochamad, et al. "POST TRAINING QUANTIZATION IN LENET-5 ALGORITHM FOR EFFICIENT INFERENCE." (2022).

[11] Sze, Vivienne, et al. "Efficient processing of deep neural networks: A tutorial and survey." Proceedings of the IEEE 105.12 (2017): 2295-2329.