Research on a convolution operation method based on domain transformation in deep learning

Zhichen Wang$^{1,2*}$, Hongliang Li$^1$

$^1$Jiangnan Institute of Computing Technology, Wuxi, Jiangsu, 214083, CHINA
$^2$Officers College of PAP, Chengdu, Sichuan, 610213, CHINA

*Corresponding author’s e-mail: wangzhch@mail.ustc.edu.cn

Abstract. In deep learning, convolution operation is an important part of convolutional neural network. Various transformations have been used to deal with convolution problems, but complex number operation and the inability to carry out convolution in the DCT domain have been introduced. In this paper, the relationship between DCT transform of output matrix and DCT transform of input matrix is found. At the same time, in view of the reality of partitioning operation in the case of large matrix, the method of overlapping of blocks is introduced. Finally, experiments on LeNet show that the proposed method can achieve the accuracy of the traditional method and can be used in practical applications.

1. Introduction

Neural networks are widely used in image recognition, automatic translation and advertising recommendation. Because of the growing deep structure, the multilayered structure with a large number of neurons and connections, or synapses. Although these neural networks are very powerful, the large weights consume considerable storage and memory bandwidth. For example, AlexNet is 240MB and vgg-16 is 520MB. In recent years, a series of advances have been made in neural network acceleration. Jong Hwan Ko et al. proposed the design of a frequency-domain accelerator for CNN energy saving training [1]. By applying the Fourier transform, the complex convolution operation is transformed into a simple dot product operation. In order to eliminate the Fourier transform and inverse transform of each layer, the entire network is trained completely in the frequency domain by using the approximate frequency domain nonlinear operation. Simulation results show that the delay of lenet-5 training is reduced by 2 times and the power consumption is reduced by 2.5 times. In the case of AlexNet with a large input image and kernel, the training delay was reduced by 4.7 times and the power consumption was reduced by 6.3 times. Jong Hwan Ko et al. also proposed an adaptive weight compression neural network inference engine based on JPEG image coding algorithm [2]. The quality factor of the JPEG encoder is adaptive controlled according to the accuracy of each block to maximize the compression ratio with the minimum loss of accuracy. At a 1% accuracy loss, the MLP was 63.4 times compressed, the LeNet-5 (MNIST dataset) 31.3 times compressed, and the AlexNet and ResNet-50 (ImageNet) 15.3 times and 10.2 times compressed, respectively.

We can see that the introduction of transformation technology makes the training of neural network significantly accelerated. However, there are some obvious limitations. Jong Hwan Ko et al. proposed the introduction of Fourier transform to accelerate convolution during CNN training. This is taking advantage of the property of $\hat{f} \ast \hat{g} = \hat{f} \cdot \hat{g}$ [3]. While speeding up the convolution operation, complex multiplication and addition are introduced, resulting in significantly more overhead on...
hardware. Another neural network inference engine based on JPEG image coding algorithm uses the DCT (discrete cosine transform) as the basis for compression operation, which can greatly compress the data and reduce the storage pressure. At the same time, DCT transformation is carried out in real number domain, which will not increase the operation complexity due to complex number operation. Since DCT transform does not satisfy the property of convolution operation in Fourier transform, convolution cannot be carried out in DCT transform. This causes that every convolution operation needs to extract the original compressed matrix and restore it to the original matrix for operation, which increases the cost of calculation accordingly.

There is also some progress in the field of DCT transformation. Martucci[4] first derived the convolution product property of all DCT and DST (discrete sine transform), and the algorithm required symmetric expansion of signals, which could not be simply extended to linear convolution. Reju et al. [5] deduced the circular convolution in DCT and DST fields, and it can be extended to linear convolution by adding zero. Suresh et al. [6] improved the algorithm of Reju et al., and proposed a linear convolution algorithm with lower algorithm complexity. In the convolution operation of 2-d DCT domain, Kresch et al. [7][8][9] proposed a series of algorithms to directly partition JPEG images for convolution filtering, but only limited to image blocks of size 8×8 pixels. In this paper, the relationship between DCT transform of output matrix and DCT transform of input matrix is found. At the same time, experimental verification has been carried out in CNN network.

2. Related technologies
In 1807, French mathematician and physicist Jean Baptiste Joseph Fourier proposed Fourier Transform (Fourier Transform, FT). There are many forms of Fourier transform. The normalized two-dimensional Discrete Fourier transform (Discrete Fourier transform, DFT) can be written as follows:

\[
F(u,v) = \frac{1}{\sqrt{NM}} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) e^{-\frac{2\pi i u x}{N}} e^{-\frac{2\pi i v y}{M}}
\]  

(1)

\[
f(x,y) = \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{\frac{2\pi i u x}{M}} e^{\frac{2\pi i v y}{N}}
\]

(2)

The Fourier transform contains complex operations, which are more complex and store longer than real operations. In order to simplify the above process and achieve a better transformation effect, cosine transformation came into being. In 1974, three professors k. r. Rao, n. Ahmed and t. Natarajan established the Discrete Cosine Transform (Discrete Cosine Transform, DCT). In the field of digital signal and digital image processing, the effect of DCT is close to the theoretical optimal transform -- kahunen-loeve transform (K-L transform).

In the one-dimensional case, the specific expression is as follows:

\[
F(0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x)
\]  

(3)

\[
F(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos \frac{(2x+1)u\pi}{2N}
\]  

(4)

In equation (4), F(u) is the value of the u-th cosine transformation, and u is the generalized frequency variable, u=1,2,.., N-1; F (x) is a sequence of N points in the time domain; X = 1, 2, ..., N-1.

The two-dimensional discrete cosine transform can be expressed by the following expression.

\[
F(u,v) = \sqrt{\frac{2}{M}} \sqrt{\frac{2}{N}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos \frac{(2x+1)u\pi}{2M} \cos \frac{(2y+1)v\pi}{2N}
\]  

(5)

Equation (5) is the positive transformation formula of the two-dimensional discrete cosine transform, where f(x,y) is a two-dimensional vector element of an M×N in the space domain, namely an M×N matrix, x= 0,1,2,.., M - 1; Y = 0,1,2,.., N-1; F(u,v) is the transformation domain matrix obtained by calculation, u = 0,1,2,.., M-1, v = 0,1,2,.., N-1.
3. Domain transformation convolution

F(x,y) complement 0 in equation (1) is extended to a 2M×2N matrix, and then the real part is taken after OTDFT operation, equation (6) can be obtained:

\[ \text{Re}\{F_{\text{2Mx2N}}^{\text{OTDFT}}(u,v)\} = \sum_{x=0}^{2M-1} \sum_{y=0}^{2N-1} f(x, y) \cos\left(\frac{2(x+1)\pi}{2M}\right) \cos\left(\frac{2(y+1)\pi}{2N}\right) \]  

(6)

Combining equation (6) and equation (5), it can be seen that the relationship between DCT transformation and FFT transformation is as follows:

\[ F_{\text{DCT}}^{\text{MxN}}(u,v) = \frac{1}{\sqrt{MN}} \left( \text{Re}\{F_{\text{2Mx2N}}^{\text{OTDFT}}(u,v)\} + \text{Re}\{F_{\text{2Mx2N}}^{\text{OTDFT}}(u,-v)\} \right) \]  

(7)

As we know \( W_k = e^{-\frac{2\pi i}{K}} \), then

\[ F_{\text{DCT}}^{\text{MxN}}(u,v) = \frac{1}{\sqrt{MN}} \left( W_u W_v \text{Re}\{F_{\text{2Mx2N}}^{\text{OTDFT}}(u,v)\} + W_u W_v \text{Re}\{F_{\text{2Mx2N}}^{\text{OTDFT}}(u,-v)\} \right) \]  

(8)

Equation (8) is the relationship between the two-dimensional DCT transform and FFT transform.

In the case of matrix convolution operation in neural network, let the size of the matrix be M×N, and the size of the convolution kernel be h×w, where h and w are the length and width of the convolution respectively. Before the beginning of convolution operation, zero is added around the original matrix, and then the convolution kernel is applied to each element of the matrix one by one to obtain the convolution result. Through DFT transformation, the above process is converted to the product operation in the frequency domain, and then the convolution result is obtained through the inverse DFT transformation. Similarly, if the coefficients of the matrix after convolution are known in the DCT frequency domain, then the convolution result is obtained through the inverse DFT transformation.

In equation (11), it can be seen that, \( G_{\text{DCT}}^{\text{MxN}}(u,v) \), obtained by \( \text{Re}\{G_{\text{DCT}}^{\text{MxN}}(u,v)\} \) calculation, since both calculation methods are the same, the following calculation of \( \text{Re}\{G_{\text{DCT}}^{\text{MxN}}(u,v)\} \) is derived.

\[ \text{Re}\{G_{\text{DCT}}^{\text{MxN}}(u,v)\} = \text{Re}\{F_{\text{2Mx2N}}^{\text{OTDFT}}(u,v) \cdot H_{\text{2Mx2N}}^{\text{DFT}}(u,v)\} \]  

(9)

When the matrix is extended to 2M×2N, it still satisfies the relation of equation (9), then

\[ G_{\text{2Mx2N}}^{\text{DFT}}(u,v) = F_{\text{2Mx2N}}^{\text{DFT}}(u,v) \cdot H_{\text{2Mx2N}}^{\text{DFT}}(u,v) \]  

(10)

And because of equation (10)

\[ \text{Re}\{W_u W_v \cdot G_{\text{2Mx2N}}^{\text{DFT}}(u,v)\} = \text{Re}\{F_{\text{2Mx2N}}^{\text{OTDFT}}(u,v) \cdot H_{\text{2Mx2N}}^{\text{DFT}}(u,v)\} \]  

(11)

In combination with equation (7), it can be seen that, \( G_{\text{DCT}}^{\text{MxN}}(u,v) \) obtained by \( \text{Re}\{G_{\text{DCT}}^{\text{MxN}}(u,v)\} \) and \( \text{Re}\{G_{\text{DCT}}^{\text{MxN}}(u,v)\} \) calculation, since both calculation methods are the same, the following calculation of \( \text{Re}\{G_{\text{DCT}}^{\text{MxN}}(u,v)\} \) is derived.

\[ \text{Re}\{G_{\text{DCT}}^{\text{MxN}}(u,v)\} = \text{Re}\{F_{\text{2Mx2N}}^{\text{OTDFT}}(u,v) \cdot H_{\text{2Mx2N}}^{\text{DFT}}(u,v)\} \]  

(12)

(13)
\[
\text{Re}\{G_{2M,2N}^{\text{OT DFT}}(u,v)\} = |H_{2M,2N}^{\text{DFT}}(u,v)| \times F_1(u,v) \quad (14)
\]

In the same way
\[
\text{Re}\{G_{2M,2N}^{\text{OT DFT}}(u,-v)\} = |H_{2M,2N}^{\text{DFT}}(u,-v)| \times F_1(u,-v) \quad (15)
\]

Substitute equation 14 and 15 into equation 7 to obtain the final result as follows
\[
G_{M \times N}^{\text{DCT}}(u,v) = \frac{1}{2} \left[ \sum_{k=0}^{2M-1} \sum_{v=0}^{2N-1} \left( |H_{2M,2N}^{\text{DFT}}(u,v)| \times \left( F_T^{\text{DCT}}(u,v) + F_T^{\text{DST}}(u,v) \right) + |H_{2M,2N}^{\text{DFT}}(u,-v)| \times \left( F_T^{\text{DCT}}(u,-v) + F_T^{\text{DST}}(u,-v) \right) \right) \right] \quad (16)
\]

4. Overlap-add method

After the birth of the fast Fourier transform (FFT), the convolution calculation in the time domain is transformed into the multiplication operation in the frequency domain by FFT, which greatly reduces the operation time and improves the efficiency of digital signal processing. However, when the matrix is large, the convolution process still takes a long time. At the same time, processing large matrices requires more storage space. In order to realize the real-time processing of data, this paper will introduce a long sequence convolution calculation method: overlap and addition method.

The overlapping addition method actually describes a method of data recovery, which is a method of recombining long sequence segments into a whole segment after convolution operation. There are two main benefits of convolution. First, it can realize the real-time processing of matrix convolution. Second, it can greatly reduce the demand for hardware storage space.

The overlapping addition method first divides the long sequence \( x(n) \) into \( N \) segments
\[
x_1(n) = \{x(0), x(1), x(2), x(3), x(4)\}
\]
\[
x_2(n) = \{x(5), x(6), x(7), x(8), x(9)\}
\]
\[
\vdots
\]
\[
x_N(n) = \{x(5N-5), x(5N-4), x(5N-3), x(5N-2), x(5N-1)\}
\]

Then, \( x(n) \) convolve with \( h(n) \) (can be converted to the frequency domain by FFT) respectively, and the result is \( N \) segments. In the process of convolution, the shift times and sum is carried out after the inversion of \( h(n) \). In the beginning of the shift, the pre-transition zone will be generated, and similarly, the post-transition zone will be generated in the end phase. All data in the transition zone need to be processed to return to the correct data, and the processing here is overlapping and adding.

Sum the posterior transition zone of the previous section of data and the pretransition zone of this section of data, and arrange them in order to obtain a complete result.
\[
y(kN) = y_k(N) + y_{k+1}(0)
\]
\[
y(kN+1) = y_k(N+1) + y_{k+1}(1)
\]
\[
\vdots
\]
\[
y(kN+M+1) = y_k(N+M-2) + y_{k+1}(M-2)
\]

5. Experiment

In this paper, the Tesla series P100 computer card of NVIDIA is selected for experiment, and MNIST data set and cifar-10 data set are tested on LeNet network. The test results are shown in the following table.
Table 1. The traditional convolution method and our method in this paper test results

|      | Traditional method | Our method |
|------|-------------------|------------|
| MNIST| 99.2%             | 99.1%      |
| CIFAR-10 | 98.5%     | 98.6%      |

6. Summary
In this paper, the relationship between the traditional Fourier transform and the orthogonal cosine transform is analysed. And then we derive the Fourier transform using the orthogonal cosine transform. At the same time, in view of the fact that the convolution matrix is large in size, the method of overlapping of blocks is introduced. Experiments show that the method can achieve the same precision as the traditional convolution method and can be used in practice.

References
[1] Jong Hwan Ko, Burhan Mudassar, Taesik Na, Saibal Mukhopadhyay. Design of an energy-efficient accelerator for training of convolutional neural networks using frequency-domain computation. DAC '17:Proceedings of the 54th Annual Design Automation Conference 2017 Austin, TX, USA 2017.
[2] Jong Hwan Ko, Duckhwan Kim, Taesik Na, Saibal Mukhopadhyay. Design and analysis of a neural network inference engine based on adaptive weight compression. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (Volume: 38 , Issue: 1 , Jan. 2019 109 – 121 10.1109/TCAD.2018.2801228).Another reference
[3] Michael Mathieu, Mikael Henaff, Yann LeCun. Fast Training of Convolutional Networks through FFTs. DOI=http://hgpu.org/p=11136.
[4] Martucci S A. Symmetric convolution and the discrete sine and cosine transforms. IEEE Transactions on Signal Processing, 1994, 42( 5) : 1038-1051.
[5] Reju V G, Koh S N, Soon I Y. Convolution using discrete sine and cosine transforms. IEEE Signal Processing Letters, 2007, 14( 7) : 445-448.More references
[6] Suresh K, Sreenivas T V. Block convolution using discrete trigonometric transforms and discrete Fourier transform. IEEE Signal Processing Letters, 2008, 15: 469-472.
[7] Kresch R, Merhav N. Fast DCT domain filtering using the DCT and the DST. IEEE Transactions on Image Processing, 1999, 8( 6) : 821-833.
[8] Merhav N, Kresch R. Approximate convolution using DCT coefficient multipliers. IEEE Transactions on Circuits and Systems for Video Technology, 1998, 8(4) : 378-385.
[9] Yim C H. An efficient method for DCT-domain separable symmetric 2-D linear filtering. IEEE Transactions on Circuits and Systems for Video Technology, 2004, 14( 4) : 517-521.