Research on accelerating CEE-DGCNN event extraction algorithm based on multi-CPU platform

Yan Li¹, Jinfei Jiang¹*, Jinwei Xu¹, Zikai Gao¹, Yong Dou¹ and Haifang Zhou²

¹Science and Technology on Parallel and Distributed Laboratory, National University of Defense Technology, Changsha, 410005, China
²Department of Computational Science, National University of Defense Technology, Changsha, 410005, China

*Corresponding author’s e-mail: jingfeijiang@nudt.edu.cn

Abstract. With the development of the Internet, the volume of information is expanding rapidly, and the complex information makes it particularly important to extract information quickly and intelligently. Event extraction algorithm is such a kind of fast and accurate information acquisition algorithm. With the development of deep learning in recent years, deep learning also shines in the field of time extraction. CEE-DGCNN network is an efficient and accurate event extraction algorithm for Chinese text. At present, the acceleration research on deep learning in the academic community mainly focuses on GPU, FPGA and other platforms. In fact, compared with these platforms, CPU is more versatile. Considering that the algorithm should have universality and universality in industrial application, this paper makes an in-depth analysis of CEE-DGCNN algorithm to study its acceleration on CPU. Experiments were carried out on x86 and ARM architectures respectively, and good acceleration effects were achieved.

1. Introduction

With the rapid development of Internet information technology, the volume of information expands rapidly, and more and more chaotic information makes information extraction intelligent and fast become particularly important. The task of event extraction is proposed under such a background. In general, the event extraction task is divided into two subtasks: events trigger word extraction and elements of the recognition and classification [8], since deep learning fast development, deep learning achieved good effect in the event extraction task, Chen [5], etc in 2015, was proposed based on dynamic pooling more convolution event extraction algorithm of neural network, Liu, [6] and so on was proposed in 2016 by using the two-way LSTM network event detection, Nguyen [7] in 2016 put forward such as event extraction algorithm based on two-way LSTM network.

The EE-DGCNN network was proposed by Zhang [1] et al. for event extraction task, mainly based on expansive convolution and gate convolution technology. Compared with other models, the F1 value of the ACE English characters of this model was significantly improved. The F1 value of the ELEMENT recognition and classification tasks proposed by Nguyen[10] and Yang[11] increased by 5.8% and 3.2%, respectively.

CEE-DGCNN network [2] is a variant of EE-DGCNN network, which is targeted at Chinese corpus. In terms of network structure, it expands the number of network layers, and adjusts the expansion coefficient and input parameters to make it more suitable for Chinese characters. Compared with DGCNN, CEE-DGCNN improves the F1 value of triggering word classification task by 3.07%.
This paper studies CPU acceleration based on CEE-DGCNN network forward reasoning. Firstly, most of the researches on deep learning acceleration are based on GPU, FPGA and other platforms. However, in fact, compared with these platforms, general-purpose CPU has a broader market scene with better universality and universality. Second, CPU is evolving to adapt to the changing times. Both AVX extended instruction set proposed by Intel and ARM-based NEON extended instruction set have achieved good results in deep learning network. Thirdly, compared with the training operation that takes several days or even months, deep learning network reasoning does not require much computation in small-scale tasks. Moreover, CEE-DGCNN is a one-dimensional convolutional network, and its number of parameters and computation are smaller than the two-dimensional convolutional neural network of images, so CPU can meet its requirements. Based on X86 and ARM architectures, this paper explores the application potential of the algorithm on THE CPU through in-depth analysis of the algorithm and combining the parallel capability of the processor at the thread level and instruction level.

The main research work of this paper includes: First, proposing a fitting method of Sigmod function for CEE-DGCNN network to reduce the running time of Sigmod function; Then, a new convolution computation method is designed for one-dimensional expansion convolution to solve the problem of data discontinuity in expansion convolution, and SIMD operation is adopted to accelerate the convolution computation; At last, OpenMP framework is adopted to parallel CEE-DGCNN network at thread level.

2. Related Work

CEE-DGCNN is an event extraction network for Chinese, which is similar to DGCNN. First, Bert network is adopted to process words, and word vectors are obtained as the input of the network. In addition, expansion convolution and gate convolution are also adopted to adjust the network structure, and new expansion coefficient, new size of input and output are accepted to make it more suitable for Chinese. Like other event extraction algorithm, CEE-DGCNN is also divided into two parts, Trigger network to solve the sub-task of event keyword extraction, and argument network to solve the sub-task of event element identification and classification. The two networks are similar in structure except for differences in input and output. Table 1 shows the differences between the two.

![Figure 1. CEE-DGCNN network structure](image)

The whole network is composed of 25 layers, including 24 convolutional layers and a full connection layer. The convolution kernel is computed by means of expansive convolution, and a gate structure is adopted. Its specific structure is shown in Figure 1.

| Sub-task | Trigger | Argument |
|----------|---------|----------|
| Input    | 1024    | 2048     |
| Output   | 34      | 36       |
From the perspective of network structure, it mainly includes convolutional layer and full connection layer. Convolutional layer has fewer parameters and requires a lot of computation, while full connection layer has more parameters and requires a little computation. Table 2 shows the comparison of parameters and computation between the two in a single calculation. Because of the dense feature of convolutional layer, parallel optimization for convolutional layer can effectively improve the computing speed of network.

| Sub-task     | Parameter | Computation  | Parameter | Computation  |
|--------------|-----------|--------------|-----------|--------------|
| convolution  | 96        | 98304        | 96        | 196608       |
| connection   | 34816     | 34816        | 73728     | 73728        |

2.1. Dilated Convolutional Neural Network.
EE - DGCNN network is a kind of one-dimensional convolution depth, with the method of expansion convolution as convolution. The authors of paper [3] put forward expansion convolution network, and get high marks in the semantic segmentation task, convolution compared with ordinary convolution, add a few empty in the kernel, in order to increase the feeling of the convolution, shown in figure 2 for the expansion coefficient of 1 and 2 respectively two layer neural network.

![Dilated Convolutional](image)

As shown of figure 2, for expansion convolution, when the expansion coefficient is “1”, dilated convolution is ordinary convolution, and the higher the coefficient of expansion, the greater the acceptable information more widely, but in fact, inflation convolution also defects in itself, which is also proposed in the paper [3], expansion convolution increased empty of data, which makes the continuity of data missing, and will affect the effect. In general, the expansion convolution and ordinary convolution mixing in deep learning network. CEE-DGCNN use the hybrid approach, and different expansion coefficient is adopted to reduce the impact of continuity.

For the network, expansive convolution causes discontinuity of data reading and affects operation performance. In this paper, the convolution computation is reconstructed by means of circular expansion to transform discontinuous data access into continuous data access, thus eliminating discontinuity of data and improving performance.

2.2. Gated convolution
Gated convolutional neural network is proposed by the paper [4], and its basic structure is shown in Figure 3.

The mathematical expression is shown in Formula 1.

\[ Y = X \times (1 - \sigma) + Conv1D_1(X) \times \sigma \]  

(1)

The calculation formula of parameters \( \sigma \) is shown in Formula 2.

\[ \sigma = \text{sig mod}(Conv1D_2(X)) \]  

(2)
In formula 1 and Formula 2, X and Y represent input and output vectors respectively. In CEE-DGCNN, X and Y are one-dimensional vectors with 1024 parameters. Conv1D1 and Conv1D2 represent two independent one-dimensional convolution processes, and the Sigmod function is expressed as formula 3:

$$Y = \frac{1}{1 + e^{-x}}$$  \hspace{1cm} (3)

Gated convolution introduced a number of exponent calculation, and exponent calculation is complex, which takes up a lot of computing resources. Actually, Sigmod function spend 68% of time, in order to solve this problem, the paper analysing training data, find a piecewise function insteading sigmod function calculation, and the calculation time ratio drops to 20.41%. So getting a huge increase in speed.

3. methodology

Compared with the network structure of Python language, C language has the advantages of faster running speed and stronger platform compatibility, which can be applied to the chips of various architectures. In the paper, C language is used to re-realize the network and optimize on this basis. In the paper, the multi-layer optimization network is used to realize the acceleration of the network, and the verification tests are carried out on x86 and ARM architectures, and significant results are achieved.

In the paper, network optimization is accelerated from three levels, as shown in Figure 4. First, a piecework approximation method is designed for Sigmod function, which reduces the calculation time of Sigmod function although some accuracy is lost. Second, we noticed that there were many single-instruction multi-data flow operations in the network. In view of this feature of the network, we adopted the method of circular expansion to reconstruct the convolution computation, and used the processor SIMD to extend the instruction set to realize the optimization acceleration of the single core. Finally, we give full play to the multithreading ability of the processor, realize the multi-threading running on the network on the processor, and explore the parallel efficiency under different parallel modes.
3.1. Sigmod function approximation

Gated convolution makes the network carrying out a lot of computation of Sigmod function. In order to reduce the computation time, the paper analysing the computation part of the function of Sigmod, and adopts linear approximation to reduce the computation time.

Figure 5. The data distribution of Sigmod function

Figure 5 shows the distribution of data in the network for function calculation. It can be seen that most of the trigger network data is concentrated in range [-2.5,2.5], in fact, the proportion of this part of data reaches 99.79%. The argument network data is mainly concentrated in range [-4,4], and the proportion of this part of data reaches 99.40%. According to this feature, the approximate calculation function is designed in this paper, such as formula (4)(5).

\[
f(x) = \begin{cases} 
0 & x < -2.5; \\
0.2x + 0.5 & -2.5 < x < 2.5; \\
1 & x > 2.5. 
\end{cases} 
\]  

(4)

\[
f(x) = \begin{cases} 
0 & x < -4; \\
0.02027x^2 + 0.17x + 0.375 & -4 < x < -2; \\
0.2109x + 0.501 & -2 < x < 2; \\
-0.02189x^2 + 0.1797x + 0.6196 & 2 < x < 4; \\
1 & x > 4 
\end{cases} 
\]  

(5)
Equation (4) is the trigger network approximation function, and equation (5) is the argument network approximation function. The accuracy loss can be effectively reduced by designing different approximate functions for different data distribution. In this paper, the computation time of Sigmod was reduced from 68% to 20%, and the speed was improved.

3.2. SIMD optimization
In order to adapt to the development of The Times, both X86 architecture and ARM architecture have introduced the extended instruction set to support SIMD operation. The latest X86 extended instruction set AVX-512 extends the instruction width to 512bit and can realize 16 32-bit floating point calculation at a time, which can greatly improve the computing efficiency. However, the AVX 2.0 extended instruction set supported by the mainstream x86 CPU has an instruction width of 256 bits, which can realize a single calculation of 8 32-bit floating point Numbers. By contrast, the ARM architecture extension instruction set NEON has a slightly lower instruction width, just 128 bits wide, and can perform four floating-point calculations at a time. As can be seen from Table 2, in CEE-DGCNN network, convolution layer parameters are less calculated and there are many single-instruction multi-data flow operations, which are suitable for acceleration of SIMD extended instruction set.

3.3. Thread level parallelism
In the forward operation of EE-DGCNN, each word vector is operated relatively independently, so it has good parallelism. At present, processors in various architectures adopt multithreading design, and making good use of multithreading capability of processors can effectively improve computing efficiency. As the mainstream parallel technology, OpenMP[9] can make full use of the multithreading
capability of the processor based on Shared memory and can be supported in various architectures. It is very suitable for parallelization of CEE-DGCNN.

OpenMP parallel technology as a framework, itself also has its own cost, to have a certain influence on the algorithm of parallel also, thread, the more cost will lead to greater, for the CEE-DGCNN, calculated between each thread can be completely independent, but there is a conflict in terms of reading data, this paper tries to a single thread to read data and multi-threaded read data in two ways, analysing their advantages and disadvantages.

4. experiment
In this paper, Intel Core I7-8700K processor with x86 architecture and FT-2000+ processor with ARM architecture are selected as experimental platforms.

The Sigmod function was approximated and some effects were achieved. The test results were carried out on the i7-8700k processor. Table 3 shows the test results. The basis is compared with the test results of the network before the approximation. Trigger and Argument respectively represent two parts of the network.

|          | Speed++ | Precision loss |
|----------|----------|----------------|
| Trigger  | 1.72     | <1%            |
| Argument | 1.6      | <3%            |

The approximate result proves that the approximation method proposed in this paper can achieve a large acceleration effect. Although it brings some precision loss, it is still an acceptable result compared with the previous work.

Figures 6 and 7 show the acceleration effects of AVX and NEON, respectively.

![AVX acceleration renderings](image1)

![NEON acceleration renderings](image2)

The results show that AVX can accelerate CEE-DGCNN by more than 1.4 times on the x86 architecture, and the acceleration ratio increases with the number of threads. On the ARM architecture, NEON delivers 1.8 times more acceleration on the network, which doesn't change as the number of threads increases.

In this paper, we make full use of the CPU multi-threading capability to conduct multi-threading experiments on processors of two architectures respectively. There are data conflicts when multi-threading reads data, which will affect the efficiency. Therefore, the paper adopt two parallel strategy, one is that each thread reads the data separately for calculation, the other is for a single thread to read the data to a common storage area and then each thread executes separately.

The experimental results are shown in Figure 8 and Figure 9, where the acceleration comparison results are the running time of CEE-DGCNN under the Keras framework of Python language.
As shown from the results of Figure 8 and 9, the acceleration increases significantly with the increase of the number of threads in both x86 architecture and ARM architecture, but the marginal effect of the increase of threads in x86 architecture decreases with the increase of the number of threads, while the acceleration ratio change in ARM architecture is basically linear.

For different parallel methods, there is a time delay, although there are conflicts when multiple threads read the data. However, multi-threading still has a considerable advantage over single-threaded data reading, and the more threads, the more obvious the advantage.

5. conclusion
In this paper, CPU parallel acceleration of CEE-DGCNN algorithm is studied. By analysing the algorithm network, the potential of CEE-DGCNN algorithm is explored at multiple levels. Firstly, a Sigmoid approximation method is proposed to achieve better acceleration effect under the condition of small precision loss. Secondly, the processor SIMD is fully utilized to extend the instruction set and realize the algorithm instruction level parallelism. Finally, the CPU multithreading capability is brought into play to realize the parallel of the algorithm at the thread level.

References
[1] Zhigang Kan, Linbo Qiao, Sen Yang, Feng Liu, Feng Huang. Event Arguments Extraction via Dilate Gated Convolutional Neural Network with Enhanced Local Features.//Access 2020.
[2] Han Zhe, Jiang Jing-fei, Qiao Lin-bo, Dou Yong, Xu Jin-wei, Kan Zhi-gang. Design and Implementation of Event Extraction Model and FPGA-based Accelerator [J].//Computer Engineering & Science, 2020.
[3] Fisher Yu. Vladlen Koltun. Multi-Scale context aggregation by dilated convolution.//ICLR 2016.
[4] Yann N.Dauphin, Angela Fan, Michael Auli, David Gragier. Language Modeling with Gated Convolutional Networks//arXiv:1612.08083v3.
[5] Chen Y, Xu L, Liu K, et al. Event extraction via dynamic multi-pooling convolutional neural networks[C]. // Proceedings of the 53rd ACL and the 7th IJCNLP. Beijing,China: ACL,2015:167-176.
[6] Feng X, Qin B, Liu T. A language-independent neural network for event detection[J]. //Science China: Information Sciences, 2018, 61(9):092106.
[7] Nguyen T H, Cho K, Grishman R. Joint event extraction via recurrent neural networks[C]. //Proceedings of the 2016 NAACL: human language technologies. San Diego, California:ACL,2016:300-309.
[8] Doddington G R, Mitchell A, Przybocki M A, et al. The automatic content extraction (ACE) program-tasks, data, and evaluation[C] // Proceedings of the 2004 international conference on
language resources and evaluation. Lisbon, Portugal: European Language Resources Association, 2004:837-840.

[9] OpenMP Application Programming Interface, Version 5.0. https://www.openmp.org.
[10] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 300-309, 2016.
[11] S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li, "Exploring pre-trained language models for event extraction and generation," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, (Florence, Italy), pp. 5284-5294, Association for Computational Linguistics, July 2019.