Algorithm of Heterogeneous High Performance System Based on Deep Learning Model

Yafang Li

Suzhou Chien-Shiung Institute of Technology, Taicang, Jiangsu, 215411, China

Keywords: Deep Learning Model; Heterogeneous High Performance System; Performance Optimization

Abstract: This paper focuses on the research of heterogeneous high-performance system algorithm based on deep learning (DL) model. With the development of DL, heterogeneous high-performance system becomes the key to improve computing efficiency. The purpose of this study is to design a DL algorithm model adapted to heterogeneous environment. By analyzing the related theories and technologies of DL and heterogeneous systems, an algorithm model including task analysis, resource allocation, data flow optimization and execution coordination modules is constructed. The convolution neural network is tested on heterogeneous platform, and compared with the CPU-only running model, the training time of heterogeneous system using this algorithm model is significantly shortened, from an average of 45.6 minutes to 12.3 minutes, and the classification accuracy is improved from 78.5% to 85.2%. The results show that the proposed algorithm model can effectively utilize the advantages of heterogeneous systems, improve the classification performance while accelerating the training speed of the model, and provide an effective way for DL algorithm optimization in heterogeneous high-performance systems.

1. Introduction

At present, DL has made remarkable achievements in many fields, such as image recognition and natural language processing. DL model usually has a large number of parameters and complex calculation process, and the demand for calculation resources is extremely huge [1]. The traditional homogeneous computing system gradually exposed the performance bottleneck when dealing with such high-intensity computing tasks [2]. In order to meet the increasing computing demand of DL, heterogeneous high-performance systems came into being [3]. Heterogeneous high-performance system integrates many different types of computing units, such as CPU, GPU, FPGA, etc., and provides strong computing support for DL model by virtue of the unique advantages of each computing unit.

How to give full play to the potential of heterogeneous high-performance system and make DL algorithm run efficiently on this system has become a key problem to be solved urgently [4]. At present, a lot of research has been devoted to this field, but there are still many challenges [5]. On the one hand, the architecture of different computing units in heterogeneous systems is quite different, and the data transmission and task scheduling are complex, so it is difficult to achieve collaborative optimization among the units. On the other hand, existing algorithms often fail to fully exploit the performance advantages of each computing unit when adapting to heterogeneous environments, resulting in limited overall performance improvement [6].

This paper focuses on the research on the algorithm of heterogeneous high performance system based on DL model, aiming at designing a DL algorithm model that can make full use of the characteristics of heterogeneous high performance system. By deeply analyzing the architecture of heterogeneous system and the principle of DL algorithm, the algorithm is innovatively designed from the aspects of reasonable allocation of computing resources and optimization of data transmission. At the same time, an experimental platform is built to strictly verify the proposed algorithm model in order to evaluate the effectiveness and feasibility of the algorithm.

  DOI: 10.25236/iiicec.2025.024

2. Heterogeneous high performance system architecture

Based on artificial neural network, DL automatically learns feature representation from a large number of data by constructing a multi-level network structure. Its core components include neurons and layers. Neurons simulate the working mode of biological neurons, receive input and generate output through activation function [7]. The training process of DL depends on the back propagation algorithm, which calculates the gradient of the loss function about the network parameters and updates the parameters in the opposite direction of the gradient, so that the loss function is continuously reduced, thus optimizing the model performance.

Heterogeneous high-performance system integrates many different types of processors, and each processor has unique characteristics in performance and power consumption. CPU is versatile and good at logic control and complex instruction processing, but its performance in large-scale data parallel computing is limited [8]. GPU is specially designed for large-scale parallel computing, and has a large number of computing cores, which is excellent in DL matrix operations, but its control ability is relatively weak. FPGA is reconfigurable, and users can customize the hardware circuit according to their needs, which can achieve efficient acceleration in specific application scenarios. These processors are connected with each other through a high-speed bus, and cooperate to complete the DL task. Data parallelism and model parallelism are the keys to improve DL efficiency of heterogeneous high performance systems [9]. Data parallelism divides data into multiple subsets and processes them simultaneously on different processors; Model parallelism assigns different parts of DL model to different processors for execution. These two technologies can reduce the idle computing resources and improve the overall performance of the system by reasonably allocating tasks.

3. Algorithm model of heterogeneous high performance system based on DL model

In heterogeneous high-performance systems, different types of processors have their own unique advantages and limitations. In order to give full play to the potential of these processors and improve the efficiency of DL model, the algorithm model proposed in this paper is based on the refined management of heterogeneous system resources and task allocation. The core idea is to map the DL model to the most suitable processor according to the characteristics of each part of the computing task, and optimize the data transmission between different processors to reduce the extra overhead caused by data communication. For highly parallel and computation-intensive tasks, such as convolution layer calculation in convolutional neural networks, GPU is given priority to process, and its large number of computing cores are used to realize fast operation. For tasks with complex logic control and relatively small data volume, such as some control logic and parameter updating parts in neural network, CPU will perform them, giving full play to its powerful logic processing ability. FPGA technology can effectively accelerate hardware customization processing for specific tasks.

The algorithm model architecture designed in this paper mainly includes task division module, resource allocation module, data transmission optimization module and model coordination module, which cooperate with each other to accomplish the efficient execution of DL tasks on heterogeneous high-performance systems.

The task division module is responsible for analyzing the input DL model and dividing it into different types of subtasks according to the computational characteristics of tasks. The resource allocation module allocates the most suitable processor resources for each subtask according to the results of task division and the current load and performance characteristics of each processor in the heterogeneous system. The module will monitor the resource usage status of each processor in real time to ensure the rationality and efficiency of resource allocation. The data transmission optimization module aims to reduce the delay when data is transmitted between different processors. Before the task is executed, the module will plan the transmission path and timing of data in advance, and try to preload the data into the cache of the target processor to avoid the processor being idle due to data waiting. At the same time, efficient data compression and coding technology

is adopted to reduce the data transmission volume and further improve the data transmission speed. The model coordination module is responsible for coordinating the execution order and progress of each subtask on different processors. The model training and inference process requires the correct handling of data dependencies between subtasks to avoid data inconsistencies or task waiting issues.

Taking the application of Convolutional Neural Network (CNN) in heterogeneous systems as an example, the core of the algorithm is expounded. In CNN model, convolution layer, pooling layer and fully connected layer have different computational characteristics.

Convolution layer task processing: Convolution layer has a large amount of calculation and high parallelism. Firstly, the algorithm identifies this feature by the task analysis module, and the resource allocation module assigns it to GPU. The data flow optimization module asynchronously transmits the input data from the memory to the GPU memory in advance, and divides the data blocks to fit the GPU multithreading architecture. During GPU execution, each thread block processes different data blocks and convolution kernel operations in parallel. The convolution operation formula can be expressed as:

$$Y_{i,j,k} = \sum_{m=0}^{\omega-1} \sum_{n=0}^{h-1} \sum_{l=0}^{C-1} X_{i+m,j+n,l} K_{m,n,l,k} + b_k \quad (1)$$

Where Y is the output characteristic diagram of convolution operation; (i, j) is the coordinate of the output feature map; k represents the k convolution kernel; b is an offset. For the 3×3 convolution kernel, the corresponding data block is divided into different threads, and the threads multiply and accumulate the elements in the data block and the weight of the convolution kernel according to the convolution operation rules to complete the convolution operation. After processing a data block, the threads are synchronized, and then the next round of operation is carried out until the whole convolution layer is calculated.

Task processing of pooling layer: the calculation of pooling layer is relatively simple, mainly for data dimension reduction. After the task analysis module identifies, if the GPU load is high, the resource allocation module assigns the pooled task to the CPU. The CPU traverses the input data to pool the maximum value or average value according to the set pool window and step size. Assuming that the input feature map is Z and the pool window size is $k \times k$, the maximum pool operation can be expressed as:

$$P_{ij} = \max_{m=0}^{k-1} \max_{n=0}^{k-1} Z_{i \cdot k+m, j \cdot k+n} \quad (2)$$

Where P_{ij} is the value of the pooled result at position (i, j). For the 2×2 maximum pool window, the CPU scans the data in the window in turn, selects the maximum value as the pool result output, and gradually completes the pool layer calculation.

Fully connected layer task processing: fully connected layer includes matrix multiplication and logical operation. After the analysis of the task analysis module, the resource allocation module depends on the system resources. If the GPU resources are sufficient, the matrix multiplication part is handed over to the GPU to speed up the operation by using its parallel computing ability. The remaining logical operation part is allocated to CPU. Fully connected layer formula:

$$F^{[l]} = f\left(W^{[l]T} P^{[l-1]} + b^{[l]}\right) \quad (3)$$

Where $W^{[l]T}$ is the weight of the fully connected layer and $b^{[l]}$ is the offset. When the GPU performs matrix multiplication, the block matrix multiplication strategy is adopted to divide the large matrix into several small matrix blocks, and each thread block processes the multiplication of different matrix blocks in parallel to improve the operation efficiency. The CPU completes the subsequent logical operation and updates the model parameters. Through the precise allocation and optimal execution of CNN tasks in heterogeneous systems, this algorithm model can effectively improve the performance of DL model in heterogeneous environments and give full play to the potential of heterogeneous high-performance systems.

4. Experimental verification

The experiment was built on a heterogeneous platform composed of CPU (Intel Corei7-12700K) and GPU (NVIDIA GeForce RTX 3080), with Ubuntu20.04 as the operating system and PyTorch as the DL framework. The classic AlexNet model is selected for image classification, and the data set is a self-built small image data set, which contains 5,000 images in 5 categories. The experiment set up two groups of comparisons, namely, using only CPU running model and using the heterogeneous high-performance system algorithm model based on DL model in this paper. The comparison indexes were the training time and classification accuracy of the model, and each group of experiments was repeated five times to take the average value.

Energy consumption is an important index to measure system performance. Figure 1 shows the energy consumption of different systems in the process of completing a training. The energy consumption of homogeneous CPU system is 200 watt-hours, that of heterogeneous system with only GPU is 350 watt-hours, and that of heterogeneous high-performance system in this algorithm model is 280 watt-hours. Although GPU has strong computing power, its energy consumption is relatively high. By reasonably allocating tasks, the algorithm model in this paper can effectively reduce the overall energy consumption while ensuring high performance, and achieve a good balance between performance and energy consumption.

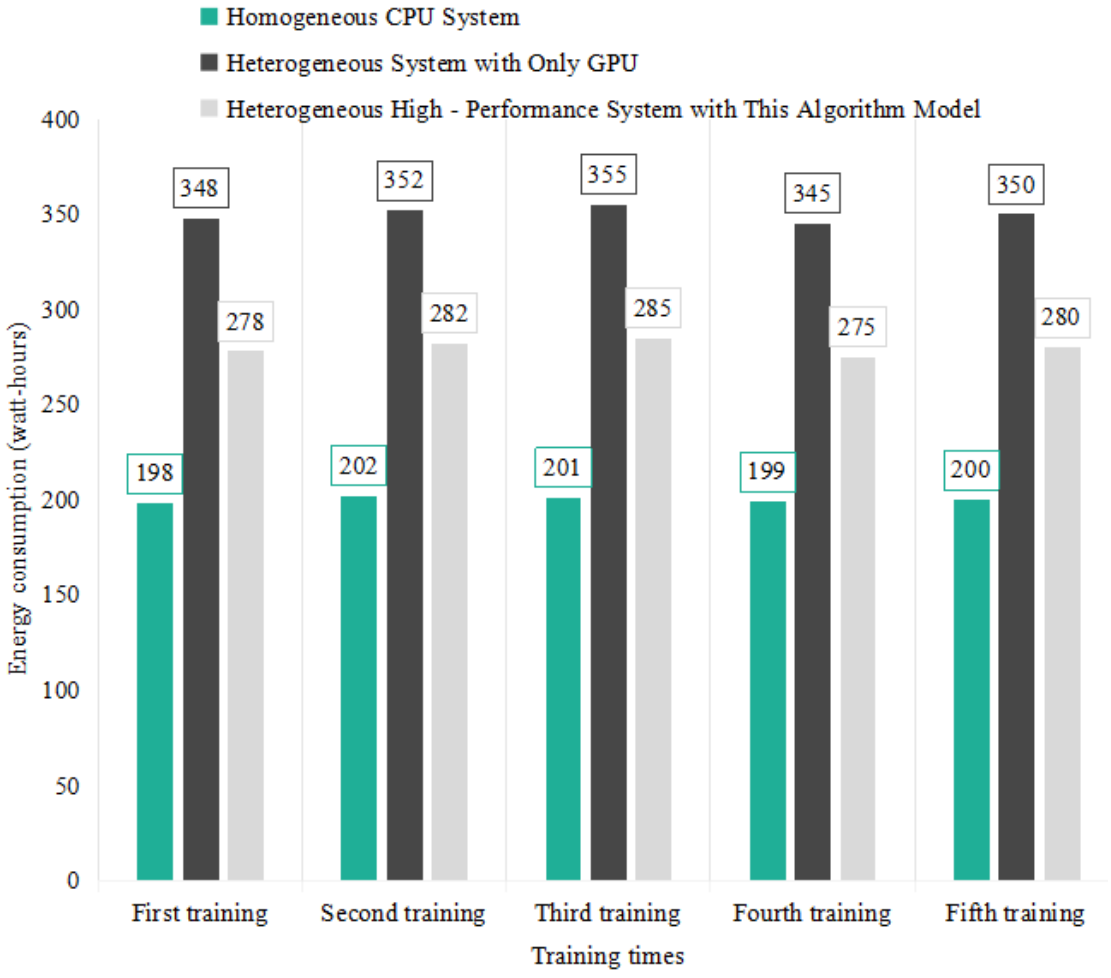

Figure 1 Comparison of model energy consumption under different systems

Table 1 shows the training time of the two running modes. From the data in the table, it can be seen that when only CPU is used, the model training takes an average of 45.6 minutes. However, the average training time of the heterogeneous system with the proposed algorithm model is reduced to 12.3 minutes. This is because GPU in heterogeneous system undertakes a lot of parallel computing tasks, which greatly improves the training efficiency.

Table 1: Comparison of Model Training Time under Different Operating Modes

Operating Mode	Average Time for One Training Round (Minutes)	Average Time for Two Training Rounds (Minutes)	Average Time for Three Training Rounds (Minutes)	Average Time for Four Training Rounds (Minutes)	Average Time for Five Training Rounds (Minutes)	Overall Average Training Time (Minutes)
Using CPU Only	46.2	45.1	45.9	45.5	45.3	45.6
Heterogeneous System of the Algorithm Model in This Paper	12.5	12.1	12.4	12.2	12.3	12.3

The classification accuracy reflects the performance of the model, and the specific data are shown in Table 2. When only CPU is used, the model classification accuracy is 78.5%. After the heterogeneous system is combined with this algorithm model, the accuracy is improved to 85.2%. This is due to the cooperative work of components in heterogeneous systems, which makes the model learn features better in the training process, thus improving the classification accuracy.

Table 2: Comparison of Model Classification Accuracy under Different Operating Modes

Operating Mode	First-Time Accuracy (%)	Second-Time Accuracy (%)	Third-Time Accuracy (%)	Fourth-Time Accuracy (%)	Fifth-Time Accuracy (%)	Overall Average Accuracy (%)
Using CPU Only	78.2	78.8	78.0	78.6	79.0	78.5
Heterogeneous System of the Algorithm Model in This Paper	85.0	85.5	85.3	84.8	85.4	85.2

Through the comparison of the above experimental data, it can be seen that the algorithm model of heterogeneous high performance system based on DL model proposed in this paper is superior to the case of using only CPU in training time and classification accuracy. This shows that the algorithm model can effectively utilize the advantages of heterogeneous systems, which not only accelerates the training speed of the model, but also improves the classification performance of the model, and verifies the effectiveness and practicability of the algorithm model.

5. Conclusions

This paper focuses on the algorithm of heterogeneous high performance system based on DL model. Firstly, the related theories and technologies of DL and heterogeneous high performance systems are elaborated in detail, which lays a solid foundation for the subsequent algorithm design. On this basis, a targeted algorithm model is constructed. Through the cooperation of task analysis, resource allocation, data flow optimization and execution coordination, the model maps all tasks of DL model to different processors of heterogeneous systems reasonably, and optimizes data transmission. The experimental results show that the heterogeneous system based on this algorithm model shows excellent performance when using convolutional neural network for image classification tasks, compared with the running model only using CPU. The training time is greatly shortened, from an average of 45.6 minutes to 12.3 minutes, which significantly improves the training efficiency; The classification accuracy is improved from 78.5% to 85.2%, which effectively enhances the classification ability of the model.

To sum up, the algorithm model proposed in this paper can give full play to the advantages of heterogeneous high-performance systems and achieve double improvement in training speed and model performance. In the future, it is expected to further expand the experimental scope and explore more optimization strategies to promote the wider application and development of heterogeneous high-performance systems in DL field.

References

[1] Mao Runze, Wu Ziheng, Xu Jiayang, et al. DeepFlame: An Open-Source Platform for Reaction Flow Simulation Based on Deep Learning and High-Performance Computing[J]. Computer

Engineering & Science, 2024, 46(11): 1901-1907.

[2] Liu Zhongbao, Wang Jie. Research on Improving Spectral Classification Performance Using High-Performance Hybrid Deep Learning Network[J]. Spectroscopy and Spectral Analysis, 2022, 42(03): 699-703.

[3] Liu Puguang, Wei Ziling, Huang Chenglong, et al. A High-Performance Data Decompression Method Based on FPGA Acceleration[J]. Chinese Journal of Computers, 2023, 46(12): 2687-2704.

[4] Wang Linbo, Bai Linting, Wen Pengcheng. Research on the Embedded Applicability of Domestic Deep Learning Inference Frameworks[J]. Aeronautical Computing Technique, 2023, 53(3): 121-125.

[5] Zhu Yue'an, Jian Huaibing, Long Yongchao, et al. Constructing a New High-Performance and Highly Available Key-Value Database System[J]. Journal of Software, 2021, 32(10): 3203-3218. DOI: 10.13328/j.cnki.jos.006023.

[6] Wang Cheng, Ye Baoliu, Mei Feng, et al. A High-Performance Key-Value Storage System Based on Remote Direct Memory Access[J]. Journal of Computer Applications, 2020, 40(02): 316-320.

[7] Lin Yongzhen, Xu Chuanfu, Qiu Haozhong, et al. Research on Heterogeneous Parallelism and Performance Optimization of DSMC/PIC Coupled Simulation Based on MPI+CUDA[J]. Computer Science, 2024, 51(9): 31-39.

[8] Liu Sheng, Lu Kai, Guo Yang, et al. An Independently Designed Heterogeneous Fusion Accelerator for E-level High-Performance Computing[J]. Journal of Computer Research and Development, 2021, 58(06): 1234-1237.

[9] Xu Shun, Wang Wu, Zhang Jian, et al. High-Performance Computing Algorithms and Software for Heterogeneous Computing[J]. Journal of Software, 2021, 32(08): 2365-2376.