2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT) | 978-1-7281-9948-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICCASIT50869.2020.9368829

A New Algorithm of Handwritten Numeral Recognition Based on GPU Multi-stream Concurrent and Parallel Model

Gaili Du[a], Liwei Jia[b], Li Wei[c]
Henan Medical College
Zhengzhou, China
[a]gailidu@yeah.net, [b]jialiwei2013@126.com, [c]weilihappy@126.com

***Abstract*—Recognition of handwritten numeral using convolutional neural networks (CNN) is a hot research topic at present. However, CNN algorithm requires high hardware equipment and has slow prediction speed. Traditional machine learning classification algorithms usually rely on manual experience to extract features. This paper proposes a new algorithm of handwritten numeral recognition based on GPU multi-stream concurrent and parallel model. The new algorithm constructs a programming environment based on the CUDA architecture, uses CUDA/C++ programming to implement the convolutional neural network algorithm. Applying the convolutional neural network algorithm to the handwritten numeral recognition problem, and selecting the appropriate network model and related parameters, GPU has high concurrency performance, which can improve the speed of convolutional neural network training data. By comparing the GPU and CPU implementation process, it is verified that it is feasible and effective to perform CUDA parallelization training and recognition on the convolutional neural network algorithm. The experiment shows that the GPU-implemented convolutional neural network algorithm has the characteristics of fast convergence, high recognition rate and fast recognition speed. The features extracted by CNN have various characteristics, which could improve the efficiency of handwritten numeral feature recognition, and solve the process of traditional manual experience extracting features and dependence on prior knowledge.**

Keywords—GPU Concurrent Model, Feature Extraction, Convolutional Neural Network, Handwritten Numeral, Parallel Acceleration

I. INTRODUCTION

With the rapid development of information technology, it is becoming more and more convenient for people to obtain information in daily life, and at the same time, a large amount of data information is also produced. In this context, people use computers to process certain data from our existing data to obtain certain rules [1]. After processing, we can use the learned rules to get something meaningful behind the data, not just the data we see. Finally, the true meaning of the data is obtained [2]. This process is called machine learning [3]. Its main purpose is to process the provided data, then find the classification of this data, or perform regression on the data to make numerical predictions [4].

The use of machine learning can be seen everywhere in life, such as the search engine ranking of search results. When people use a search engine, the search engine will record people's search habits and what people click on the search results [5]. After a long period of accumulation, the search engine will guess your work based on your daily habits of using the search engine, Reorder the search results based on these contents, and push the search results that suit you to you, which uses machine learning; There are many such applications in life, such as face recognition, handwritten recognition, spam filtering, product recommendation, purchase behavior prediction. Machine learning is important for science and is essential for people. The study of machine learning has profound significance.

At present, the hot spot in the field of machine learning is the convolutional neural network. Because its network structure is similar to the receptive field in the visual nervous system, it is particularly suitable for processing image tasks [6]. Although the convolutional neural network has a good performance in terms of accuracy, in practice, there are still many problems, such as the amount of input data is too large, the calculation speed is slow, even if the most advanced CPU is used, the time consumption is also unacceptable [7]. In recent years, GPU general parallel technology has developed particularly rapidly, because the number of computing cores is much more than that of CPU. In terms of highly parallel computing, the computing efficiency of GPU is much higher than that of CPU. CUDA is a GPU general-purpose parallel computing platform launched by NVIDIA [8]. It not only provides a series of APIs to enable developers to directly access GPU hardware, but also provides a C compiler to enable developers to use. CUDA-based code can run on the GPU, which makes the development of convolutional neural network algorithms on the GPU possible [9].

II. DEEP LEARNING

Many large companies have gradually begun to invest in deep learning algorithms and set up their own deep learning teams [10]. Google is the one with the most investment. In June 2008, the Google Brain project was disclosed. In January 2014, Google acquired Deep Mind. And then in March 2016, the Alphago algorithm developed by Google defeated Lee Sedol, a South Korean 9th-grade chess player, and proved that the algorithm designed by deep learning can defeat the strongest player in the world. In terms of hardware, NVIDIA started as a display chips producer, but from 2006 and 2007, it mainly promoted the use of GPU chips for general computing [11]. It is particularly suitable for a large number of simple and repeated calculations in

978-1-7281-9948-1/20/$31.00 ©2020 IEEE

deep learning. At present, many people choose NVIDIA's CUDA toolkit for deep learning software development. Since 2012, Microsoft has used deep learning for machine translation and Chinese speech synthesis. Behind its AI Cortana is a set of data algorithms for natural language processing and speech recognition. Baidu announced the establishment of Baidu Research Institute in 2013, the most important section of which is Baidu Deep Learning Research Institute. Facebook and Twitter have also conducted deep learning research. Facebook and Yann Lecun, Professor of New York University, set up their own deep learning algorithm lab; in October 2015, Facebook announced the open source of its deep learning algorithm framework, Torch framework. Twitter acquired Madbits in July 2014 to provide users with high-precision image retrieval services.

Although machine learning has a history of decades, two relatively recent trends have promoted the widespread use of machine learning: the emergence of massive training data and the powerful and efficient parallel computing provided by GPU computing [12]. People use GPU to train these deep neural networks. The training set used is much larger, the time consumed is much shorter, and the data center infrastructure is much less occupied. GPUs are also used to run these machine learning training models for classification and prediction in the cloud, which can support much larger amounts of data and throughput than before when they consume less power and occupy less infrastructure [13].

Early users of GPU accelerators for machine learning include network and social media companies of many sizes, as well as leading research institutions in the fields of data science and machine learning [14]. Compared with the practice of using the CPU alone, the GPU has thousands of computing cores and can achieve 10-100 times the application throughput. Therefore, the GPU has become a data scientist's processor for processing big data.

III. Convolutional Neural Network

In the field of image recognition and speech analysis, since the multi-level processing of convolutional neural networks is similar to biological neural networks, convolutional neural network algorithms have become a hot spot. And in the convolutional neural network, weighting sharing and other characteristics reduce the parameters in network training. The classic model of deep learning is a convolutional neural network. CNN actually appeared in the 1980s and 1990s, and the LeNet-5 model has achieved great success in simple handwriting recognition problems.

But the computing power at that time could not support a larger network, and LeNet-5 did not perform well in the recognition of complex objects. This is mainly because deep networks are highly non-linear and have too many local extremisms, making it difficult to guarantee an acceptable good solution; the core of the optimization strategy is the gradient descent method. But for a deep network, the gradient of its function is also very complicated, so the error back propagation algorithm often produces gradient disappearance or gradient explosion when the initial value of the parameter is not properly selected, resulting in the optimization cannot be carried out normally; For training a deep network, too small data sets often cause over fitting problems; for a large number of parameter training, the limitation of computing power often takes several months. Until 2012, Hinton and his student Alex increased the accuracy rate to 84.7% in the Imagenet competition. Of course, their success has taken advantage of ImageNet — a sufficiently large data set, the powerful computing power of the GPU, the deeper CNN network, the optimization techniques such as stochastic gradient descent (SGD) and Dropout, and the Data Augmentation. But in any case, they shocked the field of machine learning by relying on deep learning. Since then, a large number of researchers have entered this field, and it has been out of control. In 2013, it was 89%, and in 2014 it was 93.4%. As of May 2015, the accuracy of the ImageNet data set has reached more than 95%, which is comparable to human resolution.

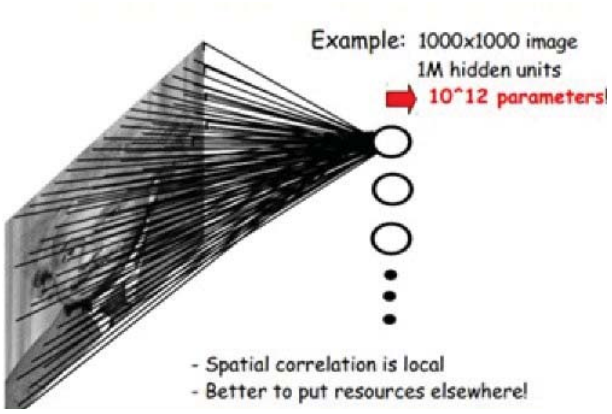

Figure 1 Fully connected network

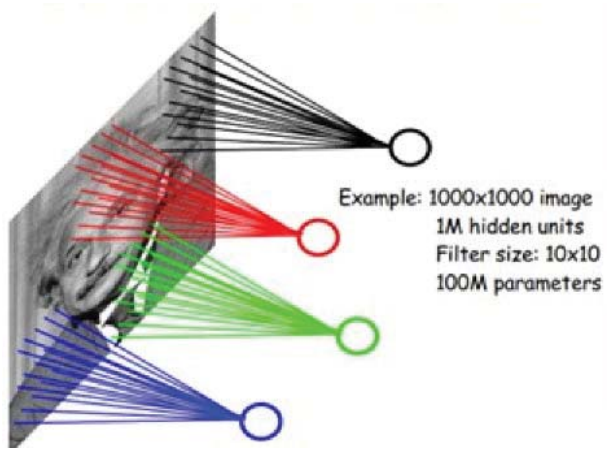

Figure 2 Partly connected network

CNN is a feed-forward neural network, which is proposed by the mechanism of biological receptive field. In the visual nervous system, the receptive field of a neuron refers to a specific area on the retina. Only the stimulation in this area can activate the neuron. Convolutional neural networks have achieved great success in the field of image processing. On the international standard ImageNet dataset, many successful models are based on CNN. One of the advantages of CNN compared to traditional image processing algorithms is that it avoids the complicated pre-processing of the image and can directly input the original image.

Traditional neural networks are all connected, that is, the neurons in the input layer to the hidden layer are all connected. This will lead to a huge number of parameters, making network training time-consuming and even difficult to train, while CNN avoids this difficulty through partly connection, weighting sharing and other methods.

Figure 1 is a fully connected network and Figure 2 is a partly connected network. For a 1000 × 1000 input image, if the number of neurons in the next hidden layer is 10^6, and a fully connected network is used, there are 1000 × 1000 × 10^6 = 10^{12} weighting parameters. It is almost difficult to train for such a huge number of parameters; on the contrary, if a local connection network is used, each neuron of the hidden layer is only connected to a 10 × 10 local image in the image, then the number of weighting parameters at this time is 10 × 10 × 10^6 = 10^8, which will directly reduce by 4 orders of magnitude.

Weighting sharing can further reduce parameters. The method of further reducing the parameters is weighting sharing. The specific approach is as follows: First, in the local connection, each neuron of the hidden layer is connected to a 10 × 10 local image, so there are 10 × 10 weighting parameters, and these 10 × 10 weighting parameters are shared with The remaining neurons, that is to say, the weighting parameters of the 10^6 neurons in the hidden layer are the same, then regardless of the number of hidden layer neurons, the parameters to be trained are these 10 × 10 weighting parameters (Convolution kernel, also called the size of the filter). Despite having so few parameters, CNN still has excellent performance. If you want to extract more features, you can add multiple convolution kernels. Different convolution kernels can get the features under different maps of the image, which is called a feature map.

IV. Model Building Based on GPU Multi-stream Concurrency and Parallelization

A. Network Model Design

The CNN framework in this paper is deployed on the GPU, and the training period is in seconds, which is faster than traditional training, so that you can explore a larger parameter space, study the impact of various structures, and select the appropriate network model. This article takes handwritten numeral recognition as an example to select a suitable network structure model and related parameters.

In this paper, all 60,000 samples of the MNIST data set are selected as training samples, which are converted into a format suitable for processing on the GPU before training and testing. For the training phase, when the input layer is initialized, the sample data of the entire input layer is scrambled in units of each sample to improve the generalization ability of the network. The scrambling method adopted by this algorithm is to generate an array with a size of 60,000, the elements in the array are random integer sequences between 0-59999, Next, the sample data set and the sample label set are rearranged in the same position as before, so as to ensure that the sample image data and the sample label still correspond correctly. For the test phase, in order to prevent random errors, the processing of disturbing the input layer is not used, but 1000 image data are sequentially read from the pre-processed files in the inherent order as the test samples to weight and bias the training. The network model is set to verify the loss and evaluate. In the process of neural network training and testing, you can configure optimization algorithms, learning rate, training times, network structure, and individual parameters of each layer through json files. The program will read this configuration file to initialize the training model and network model.

B. Convolutional Layer

The convolution layer is calculated by sliding the convolution kernel on the upper input layer one by one, and each parameter in the convolution kernel is equivalent to the weighting parameter in the traditional neural network, which is connected to the corresponding local pixel. For the convolution kernel, each parameter is multiplied by the corresponding local pixel value, and the sum of the product terms is the result on the convolution layer, the calculation diagram of the convolution layer is shown in Figure 3.

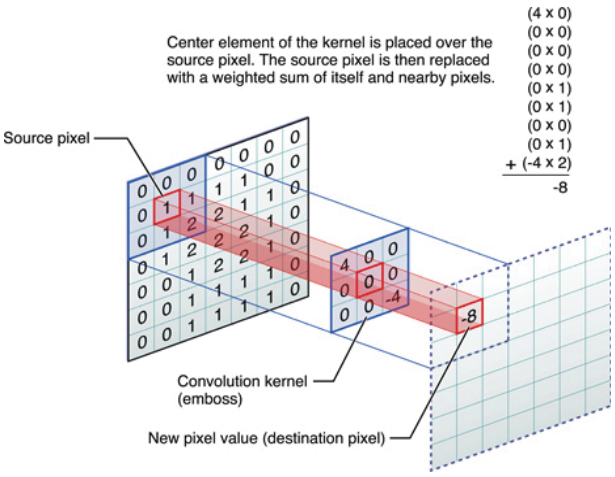

Figure 3 The calculation diagram of the convolution layer

C. Multi-stream Concurrent and Parallel Design

The parallel scheme based on the neural network structure is as follows: on the GPU, n neuron nodes of each layer are set to n threads, and each thread is responsible for calculating one neuron. This article defines a macro that facilitates the positioning of a parallel task with n tasks to the thread index on the GPU, so that each thread can execute an independent task and complete the task of parallel computing.

Many calculations in convolutional neural networks are operations between matrices and vectors, and between matrices. Matrix operations have parallelization characteristics. CUDA is very suitable for matrix operations. Therefore, if the calculation process is performed on the GPU, the calculation speed can be improved. This article uses the function cublasSgemm(). Fully connected layers have weights and offsets. They can act on the input data, and finally produce outputs through weighted operations. At this level, there are a lot of calculations, and these calculations can concur through matrix operations, thereby speeding up the calculation speed. This algorithm uses CUDA-based cublas basic linear calculation subroutine, which provides the function cublasSgemm that performs matrix operations in parallel.

V. Simulation Results and Analysis

According to the network structure, four comparative experiments are designed. The comparison between CNN-1 and CNN-2 verifies whether the addition of convolutional layer has an impact on the classification results and recognition rate of network models; similarly, a comparative experiment is conducted on CNN-1 and CNN-3 to verify whether adding full connection layer will affect the classification result and recognition rate of network model.; The comparison between CNN-3 and CNN-4 verifies

whether the number of convolution kernels in the convolutional layer has an impact on the classification results and recognition rate of network models.

TABLE I RECOGNITION RATE OF DIFFERENT NETWORK NODES

Network name	CNN-1	CNN-2	CNN-3	CNN-4
Recognition rate	97.53%	96.36%	97.66%	97.24%

In the experiment, 60,000 training data sets were used, and these data sets were iterated once for each batch of 50 data, for a total of 2400 iterations, that is, two iteration cycles were performed on the total training data set. Table I shows the average value of 10,000 test data predicted by the trained neural network.

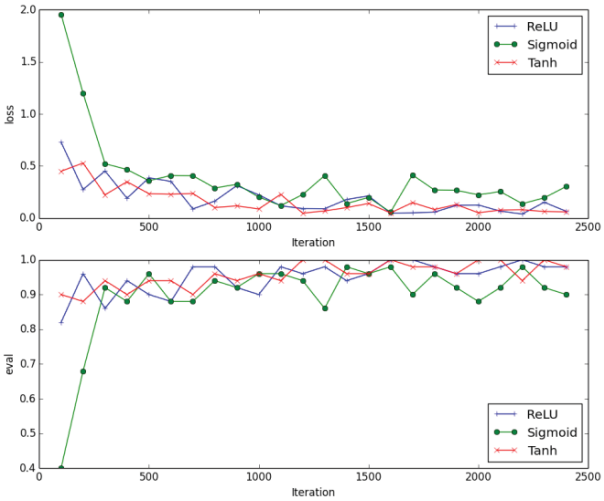

Figure 4 The loss and eval of different activation methods

Based on the selection of the CNN-3 network structure, the Xavier method is used to initialize the weights and offset parameters of each layer, and then the following three methods are used for the activation function after the fully connected layer: Sigmoid, ReLU and Tanh. Next, through experiments analyze and observe the influence of each activation function on the classification results and recognition rate of the network.

The training and testing methods of the experiment are the same as before. From Figure 4, it can be seen that when the activation function is ReLU, the average recognition rate of the test data set is the highest, and the loss value also reaches the lower value faster during the training iteration process, and the learning is faster.

In the experiment, after reading 60,000 MNIST training data sets as a training cycle, the neural networks implemented by CPU and GPU were compared, and under different training cycles, the time required to train the same amount of data was observed. The other experimental conditions are the same as the previous experiment. Finally, 10,000 pieces of MNIST test data were verified to obtain the recognition rate. The experimental results are shown in Table II.

TABLE II COMPARISON RESULTS OF GPU AND CPU

Iteration cycle	GPU training time	GPU test accuracy	CPU training time	CPU test accuracy
2	75.122s	97.59%	89.28s	97.46%
4	147.039s	98.32%	175.94s	97.89%
8	300.984s	98.73%	356.08s	98.42%

As shown in Table II, the training time and recognition rate of the implementation of the convolutional neural network CPU and the GPU are compared. It can be seen from the table that the GPU implementation has a certain acceleration effect compared to the CPU implementation training time. When the iteration period is 8, it is accelerated by 15%, and the accuracy rate is also improved, which is improved by 0.31%. The calculation has an accelerating effect on the convolutional neural network. The experimental results show that the new algorithm designed in this paper can accelerate the training time of the neural network under the condition of ensuring a high recognition rate.

VI. CONCLUSION

Traditional classification algorithms require manual experience selection, and then high-quality obvious features are extracted. For complex classification problems, feature extraction is more complicated, which makes the experimental process cumbersome. The classification based on neural network can get better results, but as the number of layers of the neural network increases, the amount of matrix operations increases, and the prediction time is also longer. The new algorithm uses the feature extraction of CNN and combines with GPU multi-stream concurrent model, first solves the problem of feature design, feature extraction, feature selection, for individual handwritten numeral, secondly, combines GPU multi-stream concurrent model with the fully connected neural network softmax classification, training takes less time and recognition The accuracy rate is higher and the prediction speed is faster.

The experiment shows that the GPU-implemented convolutional neural network algorithm is better than the CPU-based convolutional algorithm on the ordinary PC. Although the neural network algorithm has only improved by 0.31% in accuracy, it has been accelerated by 15% in speed. The experimental results indicate that the new algorithm has a better recognition effect than the simple application of CNN or machine learning algorithm.

REFERENCES

[1] Girshick, Ross, et al. Region-based convolutional networks for accurate object detection and segmentation[J], Pattern Analysis and Machine Intelligence, 2016: 142-158.

[2] Schmidhuber, Jürgen. Deep learning in neural networks: An overview [J], Neural Networks, 2015:85-117.

[3] Zeiler, Matthew D. ADADELTA: an adaptive learning rate method [J], arXiv preprint arXiv, 2012.

[4] Long, Jonathan, Evan Shelhamer. Fully convolutional networks for semantic segmentation [J], Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[5] Zeiler, Matthew D., and Rob Fergus. Visualizing and understanding convolutional networks [J], Computer vision–ECCV 2014, Springer International Publishing, 2014.

[6] He, Kaiming, et al. Deep Residual Learning for Image Recognition [J], arXiv preprint arXiv, 2015.

[7] Szegedy, Christian, et al. Going deeper with convolutions [J], Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[8] Simonyan, Karen.Very deep convolutional networks for large-scale image recognition [J], arXiv preprint arXiv, 2014.

[9] Bergstra, James, and Yoshua Bengio. Random search for hyper-parameter optimization [J], The Journal of Machine Learning Research, 2012: 281-305.

[10] Bottou, Léon. Stochastic gradient descent tricks [J], Neural Networks, Springer Berlin Heidelberg, 2012.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks [J], Advances in Neural Information Processing Systems, 2012.

[12] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization [J], The Journal of Machine Learning Research, 2011.

[13] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the Importance of Initialization and Momentum in Deep Learning [J], Proceedings of the 30th International Conference on Machine Learning, 2013.

[14] M. Saravanan and A. Priya (2019). An Algorithm for Security Enhancement in Image Transmission Using Steganography. Journal of the Institute of Electronics and Computer, 1, 1-8.

[15] Sharma Kartik; Aggarwal Ashutosh; Singhania Tanay; Gupta Deepak; Khanna Ashish (2019). Hiding Data in Images Using Cryptography and Deep Neural Network. Journal of Artificial Intelligence and Systems, 1, 143–162.