Sparse-Aware Deep Learning Accelerator

Yihang Li *

School from Heilongjiang University, Harbin, Heilongjiang, China.150000, China

* Corresponding Author Email: 20194266@s.hlju.edu.cn

Abstract. In view of the difficulty of hardware implementation of convolutional neural network computing, most of the previous convolutional neural network accelerator designs focused on solving the bottleneck of computational performance and bandwidth, ignoring the importance of convolutional neural network scarcity for accelerator design. In recent years, there are a few convolutional neural network accelerators that can take advantage of the scarcity, but they are usually difficult to consider in terms of computational flexibility, parallel efficiency and resource overhead. In view of the problem that the application of convolutional neural network (CNN) on the embedded side is limited by real-time, and there is a large degree of sparsity in CNN convolution calculation. This paper summarizes the methods of sparsification from the algorithm level and based on FPGA level. The different methods of sparsification and the research and analysis of different application layers are introduced. The advantages and development trend of sparsification are analyzed and summarized.

Keywords: CNN; Sparsification; Sparse-Aware.

1. Introduction

In recent years, Convolutional neural Network (CNN) models have been widely used in computer vision, automatic driving, natural language processing and other task scenarios, and the growth of neural Network models has greatly increased the accuracy of the results of specific applications. However, the order of magnitude increases in the number of parameters and the rapid increase in the demand for computing power bring difficulties to hardware deployment. How to deploy the huge neural network model to the equipment with limited resources and give full play to its due performance has become the focus of people. CNN topology structure is complex, computation-intensive and parameter scale is large. Increasing network depth can effectively improve the identification accuracy. However, more weight parameters lead to prolonged CNN inference time, high power consumption and large storage requirements, which are not conducive to edge calculation. How to reduce the complexity of the network and design an efficient system in accordance with CNN operation mode is the key to break through the bottleneck of artificial intelligence deployment.

By analyzing the characteristics of CNN algorithm, the researchers found that both the weight parameters and the feature graph parameters of the algorithm model have the characteristics of sparsity. The sparse feature indicates that a large number of invalid operations involving "0" will occur during the forward operation of the algorithm. If only valid operations involving non-" 0" elements are run in a computing system, the computation of the algorithmic model can be greatly reduced. However, traditional parallel accelerators or data flow optimization for CNN are usually only applicable to dense matrix operations and cannot reduce operands in a practical sense. Therefore, it is necessary to optimize sparsity to explore sparse neural network hardware accelerator.

The research on neural network convolution layer hardware accelerator has been carried out several years ago, such as DianNao, DaDianNao and Eyeriss, etc [1][2]. The research focus also involves hardware parallel development, operation array improvement, data flow optimization, data reuse, etc. In addition, a number of schemes began to consider improved algorithms to optimize design. For example, EIE (Efficient Inference Engine) introduced Sparse network to speed up computation at the full connection layer, and SCNN (Sparse CNN) used coding and hardware structure to speed up convolution computation in Sparse network. In the sparse method, the network is pruned and retrained to greatly reduce the number of network parameters without affecting the accuracy. The pruned sparse network requires a special sparse neural network accelerator. At the

algorithm level, the model size can be reduced 35-49 times by pruning, quantization, Huffman coding and other techniques to sparse the network and reduce the bit width. However, at the hardware level, how to give full play to the advantages of sparse CNN has become a major difficulty. By exploring the relationship between data reuse and power consumption, Eyeriss proposes row invariant data flow to achieve low power acceleration. However, Eyeriss failed to make full use of CNN sparsity, and the operation with a multiplier of 0 could not be omitted, and the operation delay was not reduced. By exploring the sparsity of CNN, Cambricon-X adds addressing modules on the basis of traditional hardware accelerators to avoid invalid operations and improve the computing speed and efficiency. However, its addressing modules are more complex. At the same time, AS a convenient and fast development platform, FPGA is also adopted by many developers, and test new ideas. The neural network layer fusion algorithm can improve the storage efficiency. Fast Fourier transform is used to speed up the operation. Winograd algorithm is used to speed up the convolution operation.

The CNN model will continuously increase its parameter scale, which will increase the number of convolutional operations of the neural network. In the case of embedded platform deployment, this requires the platform to have more storage and computing resources. In addition, when the embedded platform runs the CNN model for inference, it is necessary to continuously transmit the network layer parameter information used in the current propagation process through the I/O bus. This information is input to the IP core of the corresponding structure to complete the inference process of convolution calculation. However, the transmission efficiency of I/O bus is often limited, and the low transmission efficiency of high bit parameters limits the inference rate of CNN model. Because the traditional model parameter pruning method can greatly reduce the storage volume, but it needs to be restored into dense matrix in the actual reasoning process of the model, which ***cannot*** achieve substantial acceleration. At the same time, the traditional quantization method still uses floating-point number expression form, which ***cannot*** accelerate the effect, and does not consider the embedded platform transmission efficiency limit. To solve these problems, the model can also be sparsified based on the contribution rate, so as to reduce the demand for computing and storage resources of the embedded platform, ensure the speed of model inference, and optimize the computational complexity of model parameters.

2. Research and Progress of Deep Learning Accelerator for Sparse Perception

2.1 Research based on Algorithm Level

Winograd convolution is a new fast ALGORITHM of CNNs, derived from Winograd's minimum filtering algorithm [3]. Winograd algorithm reduces the number of multiplication operations through a series of transformations on the input graph and convolution kernel. Winograd algorithm has been proved to effectively reduce the operational complexity of convolution. At the same time, the algorithm also adopts a sparse network design to reduce the workload and storage space requirements. In addition, the hardware accelerator of convolutional neural network is designed to support this algorithm in a customized way, so as to obtain the benefits brought by this algorithm to convolution operation (currently, mainstream computing platforms do not support the direct implementation of this algorithm, so there is no room for improvement from this algorithm). For this reason, Liu et al. [4] proposed a new algorithm -- Winograd-Sparse algorithm. This algorithm mainly makes two improvements on the original Winograd [5] algorithm. Firstly, the ReLU layer is placed after the Winograd transform, so that the inputs involved in element multiplication remain sparse. In addition, the weight that has completed the transformation is reduced so that it remains sparse when participating in the multiplication operation, thus reducing the number of multiplication operations. Different from conventional hardware accelerators, a new accelerator structure is designed to adapt to the algorithm. According to the module division, the first design of transformation module, through the simplest addition operation, the module can quickly transform the data to be processed into Winograd domain. Secondly, in order to take advantage of the sparsity of the network, a zero-skip module is designed in front of the Processing Engine PE(PE), so as to reduce the pressure of using

multipliers in the calculation process. The design also designs a compression coding method for the sparse weight data that has been processed offline, which can not only reduce the storage requirements, but also reduce the workload of the accelerator. Finally, a linear BUFFER caching method is designed to reuse data as much as possible and reduce The Times of DRAM reading.

At present, many experts have proposed some solutions to solve the problems caused by the large scale of CNN model parameters, such as sparse model weight parameters, framework sparse network structure, low bit representation method of model parameters [6], etc. The research on reducing the scale of CNN model has become a research hotspot in recent years. For example, the design method of finer model, researchers try to achieve the adaptive deployment of model on embedded hardware by directly involving lightweight model structure. For example, the classic manual lightweight network models include Squeeze Net [7], Mobile Net [8], etc., and the current CNN network model FBnet, which is obtained based on automatic means. At present, many scholars have proposed many sparse methods of network models. Schmidhuber and Hinton G.E. adopted the method of randomly removing edges to realize the effective thinning of neural network based on this theory, and dealt with the problem of over-fitting caused by too many training times better. Molchanov et al. pointed out that the deeper the network layer, the higher the degree of pruning. This means that the last convolutional layer is pruned the most, which also leads to a significant reduction in the number of neurons in the following fully connected layer. In addition, in terms of the computational complexity of the network model, Li et al. used the contribution ranking method of the convolutional window to perform iterative sparring on the convolutional layer to achieve complete pruning. However, the complexity of calculating sensitivity for large networks is too high, so the size of the pruning method based on the value use heavy to approximate its sensitivity, specifically, namely remove weight value smaller weight, Han and others apply this idea to the recent network, and realize the model of greatly reduced size, they iteratively trim and globally fine-tune the network, always with zero weight. Guo et al. proposed a sparse algorithm based on weight values and allowed the restoration of pruning weights in previous iterations, which were closely combined with pruning and global fine-tuning stages to achieve larger model compression multiples.

At present, there are many realization methods for the SPARSE method of CNN model, among which the two common perspectives are the scale of weight parameter and the scale of convolution kernel. However, when the CNN model is deployed on the embedded platform, it will still be restored to a dense matrix form in the IP core even if the CNN model is sparsely stored in the process of reasoning and prediction. In the execution process of single-layer network on the embedded platform, model parameters in the convolution layer are first cached, and IP cores 1 to N are responsible for realizing the calculation rules, size, step size and other parameter information of the convolution kernel. To realize convolution kernels with the same structure, different values can be set through the same IP core to build multiple convolution kernels to realize the multiplexing function. Then the feature graph and convolution kernel are delivered to DSP for calculation, and the results are stored by DDR. It can be seen that parameters in the IP core store values in the form of dense matrices. For the model with sparse parameters, most of its convolution kernels are sparse matrices. However, when constructing convolution kernels, they will still be restored to the form of dense matrices and then sent to DSP for processing. At this time, although most parameters in the convolution kernel are already zero, the calculation will still be performed. In this case, the model cannot be improved. Therefore, this direction has no obvious improvement effect on the deployment of CNN model on embedded platform, which requires special software and hardware acceleration support scheme. Therefore, this direction has no obvious effect on the improvement of CNN models deployed on embedded platforms, and it needs to rely on special software and hardware acceleration support schemes. For the direction of sparse convolution kernel, the information of each channel feature map output by the network layer is analyzed, and then the convolution kernel that needs to be sparse is screened according to the proportion of the number of sparse channels. In this way, model deployment and accelerated inference can be effectively implemented on embedded platforms without relying on special hardware and software acceleration support schemes.

The structure of the neural network model can be sparsified by adopting the sparsification method for the convolution kernel and the design scheme based on the contribution rate. The convolution kernel number that contributes less to the prediction accuracy of the CNN model can be selected and the output channel parameters in the CNN model can be modified. On the one hand, it can reduce the parameter scale and model volume of CNN model. On the other hand, it can reduce the number of convolution operations performed by the model in the inference process, reduce the demand for the computing power of the embedded platform, and improve the inference speed of CNN model. The contribution rate measurement method is based on the output results of the feature map of each channel included in the network layer of the neural network model, and the forward propagation method characteristics of the process are inferred based on the neural network. If the output result of the feature map of the current channel is close to zero value, it will be used as the input data of the next network layer. At the same time, the neural network model needs the activation function to introduce nonlinear factors. However, the activation function usually eliminates the influence of the input value directly when dealing with the input value with small value. Therefore, if the feature map output of the channel conforms to the above characteristics, it cannot effectively contribute to the subsequent inference and prediction of the network layer. The method of calculating the contribution rate is to measure the input data of a batch, obtain the output result of the intermediate feature map of the neural network layer, and calculate the contribution rate of the channel in the form of L2 regularization.

2.2 Sparse Convolutional Neural Network Accelerator based on FPGA

In order to eliminate the invalid operations caused by the sparsity of model parameters in the forward calculation process of convolutional neural network, a data stream and parallel accelerator for the sparse neural network model were designed based on field programmable gate array (FPGA). The non-zeros in the feature map matrix and the convolutional filter matrix are screened out in the direction of the input channel by a dedicated logic module, and the effective data are transferred to the array composed of digital signal processors for multiplication and accumulation operation. On this basis, the final output feature map points are obtained by addition tree for all relevant intermediate results, and the best design parameters are found by parallel coarse granularity of feature map width, height and output channel direction. Experimental results on Xilinx devices show that the comprehensive performance of the VGG16 convolutional layer implemented in this design is excellent. Compared with the FPGA-based dense network gas pedal and sparse network gas pedal, its performance and power consumption are significantly improved. Compared with FPGA-based dense network accelerators and sparse network accelerators, their performance and power consumption are significantly improved.

The "best brain injury" technique needs to calculate the significance of each weight, which is difficult to be applied in modern large-scale convolutional neural networks with tens or even hundreds of megabytes of parameters. In this regard, Han et al. chose the absolute value of the weight itself as the measurement standard. This pruning method is divided into three steps: the first step is to train the convolutional neural network; In the second step, the connections whose absolute weight value is less than a certain threshold are found and these connections are cut out. The third step is to retrain the pruned network and fine-tune the remaining weight values. The second and third steps need to be iterated several times to improve the sparsity of the network as much as possible without affecting the accuracy of network prediction.

The sparsity of CNN network is defined as the proportion of zero values in the convolutional filter matrix and feature map matrix. On the one hand, the sparsity characteristics of feature map and weight parameters can be used to reduce the size of parameters and thus reduce the storage pressure of the system. On the other hand, the hardware circuit is designed to avoid the operation of "0" in sparse matrix, which can reduce the operation cost and improve the real-time performance. The sparsity of weights is mainly derived from pruning optimization technique, which is proved to be able to effectively compress the model while maintaining the accuracy of neural networks. It is found THAT

the recognition accuracy of the model will not be damaged by cutting off the unimportant value of the weight parameter (i.e., setting it to 0) according to specific rules during training. In addition, the sparsity of the feature map mainly comes from the activation function ReLU, because ReLU sets all the non-positive values of the intermediate results of the convolution operation to 0. Taking VGG16 network as an example, the experimental results show that the proportion of feature graph 0 value points in the convolutional layer can reach 80% at most, and the average proportion is more than 50%. Different from weight sparsity, the 0 value points of feature map are acquired at runtime and depend on the input image data, while the former becomes static data that can be analyzed offline after training.

3. Existing Advantages and Development Trend

For the algorithm, by moving the ReLU and pruning operations in the network to the Winograd domain, the Winograd algorithm can be used for acceleration and the sparsity of the network can be used to reduce the number of multiplication operations and further improve the efficiency. This modified CNN network, also known as the Winograd-ReLU network. It should be noted that this kind of network will abandon the convolution kernel in the original spatial domain and adopt the new weight in the Winograd domain. The two are not completely equivalent and need to be retrained. In addition, the Winograd algorithm can compress the weight and the output activation value to 40% of the original density at the same time, while maintaining the correct rate.

For the embedded side, FPGA dominates the market due to its cost and flexibility advantages. For example, there are various optimization methods for CNN accelerator: the fixed-point optimization method is used to quantify each layer one by one to reduce the number of network parameters and thus reduce the computational complexity. In this paper, some measures are adopted, such as block, memory access optimization and reuse. The concept of CLP (Convolutional layer processor) is proposed to optimize the allocation of available hardware resources on FPGA to improve the overall computational efficiency. Acceleration is achieved by placing the CNN accelerator in the charge-coupled element attachment and designing specific data access patterns within the CNN. The iteration order of hardware memory, first - and second-level cache and convolution calculation are optimized. At the same time, the parallel execution of multiplication and addition computation and the piecewise approximate representation of sigmoid function are modified to speed up the computation.

Zhang et al. [9] proposed the method of expanding the loop in the dimension of input channel and output channel, and used design space exploration to find the optimal expansion parameters. However, the parallelism of this design is limited to 7×64=448 multipliers, and the resource utilization of this design scheme will decrease significantly for higher parallelism Settings. Motamedi et al. [10] used the convolutional kernel cycle expansion to accelerate AlexNet. They chose 3×3 as the expansion parameter to expand the two-layer cycle of the two-dimensional convolution kernel. Although AlexNet contains 11×11 and 5×5 two-dimensional convolution kernels, the overall computational resource utilization of the design still reaches 97.4%. Another benefit of convolutional kernel loop unfolding is that it can be used to speed up fast convolutional algorithms such as Winograd. At present, many researches [9-12] have focused on how to select appropriate expansion parameters. There's been a huge breakthrough.

4. Concluion

This paper summarizes some design methods for sparse-aware deep learning gas pedals such as sparse network design can be used for Winograd algorithm, sparse model weight parameters, sparsification method with convolutional kernel and contribution rate-based design scheme, which makes the structure of neural network model sparse. The sparse method of model convolution kernel based on contribution rate is aimed at the redundant convolution kernel in convolutional neural network. It not only reduces the scale of parameters in convolutional neural network, but also reduces

the number of convolution computation required by the model in the process of inference and prediction. This reduces the need for computing resources for the embedded platform deployment model. Convolutional neural network models with high prediction accuracy usually have a large number of parameters and computation, which is difficult to be deployed on mobile application platforms. The pruning technique can effectively reduce the redundant parameters and computation amount of convolutional neural network model, so as to reduce the demand of hardware storage capacity and storage bandwidth. However, traditional convolutional neural network accelerators can not effectively process sparse convolutional neural networks, and can not use the reduction of network parameters and computation to improve the acceleration effect. FPGA is a common platform for the design of convolutional neural network accelerator, which has high design flexibility, short development cycle and low development threshold. This paper also summarizes some sparse perception algorithms matched to FPGAs. In conclusion, the algorithms using sparse perception have a great improvement for deep learning gas pedals.

References

[1] Chen T, Du Z, Sun N, et al. DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning[J].ACM SIGPLAN Notices,2014,49(4):269-284.

[2] CHEN Y H, KRISHNA T, EMER J S, et al. Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks[J]. IEEE Journal of Solid-State Circuits,2017,52(1):127-138.

[3] Lavin A, Gray S.Fast algorithms for convolutional neural networks[C]//Proc of Computer Vision and Pattern Recognition,2016:4013-4021.

[4] Liu Xing-yu, Pool J, Han S, et al. Efficient sparse-Winograd convolutional neural networks[C]//Proc of the 6th International Conference on Learning Representations,2018:1.

[5] Dou Jiamin, WANG Yan, Wang Yijun. Journal of Changchun University of Science and Technology (Natural Science Edition),202,45(01):72-78.

[6] Rui Xu, Sheng Ma, Yang Guo, You Huang, Yihuang Li. Design and research of Convolutional Neural Network Accelerator based on Winograd Sparse Algorithm [J]. Computer engineering and science, 2019, 41 (09):1557-1566.

[7] Young S I, Zhe W, Taubman D, et al. Transform Quantization for CNN Compression[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[8] Qiang B, Zhai Y, Zhou M, et al. SqueezeNet and Fusion Network-Based Accurate Fast Fully Convolutional Network for Hand Detection and Gesture Recognition[J]. IEEE Access, 2021, PP (99):1-1.

[9] Abdo H, Amin K M, Hamad A M, et al. Fall Detection Based on RetinaNet and MobileNet Convolutional Neural Networks[C]// ICCES2020.2021.

[10] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, Optimizing fpga-based accelerator design for deep convolutional neural networks[C], in Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. ACM, 2015, pp. 161–170.

[11] Mohammad Motamedi, Philipp Gysel, Venkatesh Akella, and Soheil Ghiasi. 2016. Design space exploration of fpga-based deep convolutional neural networks. In Design Automation Conference (ASP-DAC), 2016 21st Asia and South Paci.c. IEEE, 575–580.

[12] Wenyan Lu, Guihai Yan, Jiajun Li, Shijun Gong, et al., FlexFlow: A Flexible Dataflow Accelerator Architecture for Convolutional Neural Networks, in IEEE International Symposium on High Performance Computer Architecture, 2017: 553-564.

[13] Yufei Ma, Yu Cao, Sarma Vrudhula, and Jae-sun Seo. 2017. Optimizing Loop Operation and Data.ow in FPGA Acceleration of Deep Convolutional Neural Networks. In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. ACM, 45–54.