The implementation of an object detection algorithm on the FT processor

Bin Li[1], Shuming Chen[1] and ChaoYang[2]

[1]School of Computer,National University of Defense Technology,Changsha,China

Abstract: With the continuous development of automatic drive and neural networks, it is possible to use neural network algorithm to carry out object detection in unmanned driving. Usually, the computation of neural network algorithm is huge. How to efficiently compute the algorithm and meet the real-time requirement is a challenge. In this paper, a sparse neural network algorithm is proposed, which can improve the utilization rate of processors. The object detection algorithm YOLO is implemented on the processor. Its performance is equivalent to the current best processor performance.

1 INTRODUCTION

The deep neural network is now the foundation of many artificial intelligence applications. Because of the breakthrough application of DNN in speech recognition and image recognition, the use of DNN has increased explosively. These DNN are deployed from a variety of applications, such as autopilot, cancer detection to complex games, etc. The object detection recognition technology based on convolutional neural network has made breakthroughs. It has been widely used in image recognition, speech recognition, natural language processing and other fields. The vision based automatic driving scheme is becoming a possibility. The specific realization needs the object detection algorithm for multi target recognition of the scene. The object detection algorithm based on deep learning is fast, accurate and feasible.

2 OBJECT DETECTION

Object detection in the image need to pinpoint the locations of objects,and marked object categories. The position of an object is usually marked with a border. There may be several borders in an image. The target detection needs to give the category and probability of the object in the frame.

The YOLO algorithm solves the object detection as a regression problem. Based on a single end-to-end network, the output from the input of the original image to the position and category of the object is completed. Training and testing are carried out in a single network.The input image can get the location of all objects and their categories and corresponding confidence probability after inference.

YOLOV2 proposes a joint training method that can train target detectors at the same time by using detection data sets and classified data sets. The specific idea is to use the target detection data set to learn the accurate location of the target,and increase the number of detected targets and the robustness of the detector with the classification data set.

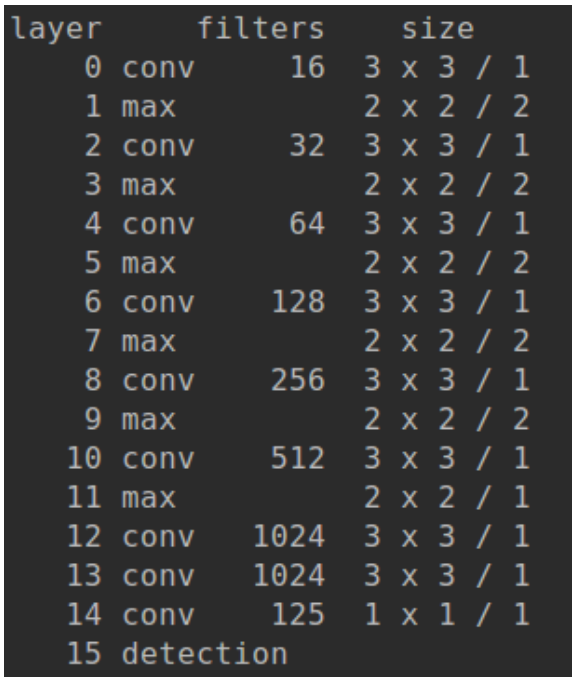
On the basis of YOLOV2, YOLO9000 is trained by COCO target detection dataset and ImageNet image classification dataset, which can detect more than 9000 kinds of targets in real time.

This article implements tiny-YOLOV2 on the processor, a simplified version of YOLOV2, with a total of 15 layers. There are 9 convolutional layers and 6 pooling layers. The amount of calculation is very large.The tiny-YOLOV2 structure diagram is shown in Figure 1.

3 ARCHITECTURE

The normalization operation of convolution and pooling in neural network will greatly improve the efficiency if it is implemented by a vectorization method. With the real-time applications emerging, computer architecture have changed greatly, some new architecture emerged, such as many-core architecture of GPU, heterogeneous multi-core architecture and vector processor architecture.

* Corresponding author: 18782103028@163.com



layer		filters	size
0	conv	16	3 x 3 / 1
1	max		2 x 2 / 2
2	conv	32	3 x 3 / 1
3	max		2 x 2 / 2
4	conv	64	3 x 3 / 1
5	max		2 x 2 / 2
6	conv	128	3 x 3 / 1
7	max		2 x 2 / 2
8	conv	256	3 x 3 / 1
9	max		2 x 2 / 2
10	conv	512	3 x 3 / 1
11	max		2 x 2 / 1
12	conv	1024	3 x 3 / 1
13	conv	1024	3 x 3 / 1
14	conv	125	1 x 1 / 1
15	detection		

Figure 1: tiny-YOLOV2 structure diagram

The new architecture integrates multiple processor cores on a single chip, each core contains rich processing components, and thus greatly improve the computational performance of the chip. A new architecture of vector processor is one of the common, including vector processor unit (VPU) and scalar processing unit (SPU), vector processing components usually contain multiple parallel vector processing unit (VPE). Data interaction between VPE can be done by protocol and shuffle.All VPE performs the same operation based on SIMD. The processor used in this paper is the FT 7002 processor, which is a vector processor, which can perform vectorization operations. There are 2 cores, each core has 16 PE, and each PE has three MAC. FT-7002 uses 2DSP core low power mode, which greatly reduces power consumption. At the same time, it has many compatible external fast communication interfaces (such as RIO, PCIE, I2C, etc.), and has strong real-time communication capability. The general processing capability of the above processor is very suitable for embedded computing in automatic driving platform.Its architecture has been very good support for typical algorithms in deep learning, such as convolution operation and matrix operation.

The processor has the following two features:

(1) Multiple cores of the processor can be operated in parallel. In the process of calculation, we need to share all the computation to 2 cores. Each core computation is almost the same, to improve efficiency. There are two key ways, one is that all cores calculate a graph together, and the amount of computation of a graph is allocated to each core, which is suitable for many situations with large number of weights and large amount of computation. The other is to calculate one graph for each core, and all cores are calculated independently. It can be adjusted flexibly according to the scale of the program and the actual situation. A multi core division method for large-scale matrix convolution calculation can be constructed based on the architecture characteristics of multi-core processors and the computation of mass matrix convolution.The input feature graph and convolution kernel are divided at the same time, so that the calculation of the matrix convolution calculation which has the calculated correlation is transformed into a completely independent matrix convolution calculation. The calculation efficiency of the mass matrix convolution increases with the increase of the computing core.The method is simple and convenient to operate, and it can fully excavate the parallelism of each layer of multi core processor. In this paper, the second method is used, and each kernel performs a single graph calculation alone.

(2) For a single kernel, all the PE perform the same operation. The calculation of the convolution neural network can be efficiently processed. When the parallelism of the processor is fully excavated, there will be a lot of convolutions in the calculation, and many different weights can be calculated at the same time. The number of channels can be grouped to make it a multiple of PE, which is calculated according to the group, so that it is convenient for the unified and parallel operation.

4 SPARSE MODE

The convolution matrix is a computationally intensive and memory intensive computation. The convolution operation in convolution neural network usually takes up more than 85% of a convolution neural network model, so how to accelerate the matrix convolution operation is an important and difficult point of current research.

In order to improve the performance, it is necessary to excavate the parallelism of calculation, improve the operation efficiency, reduce the repeated operation, reduce the energy consumption, and take a reasonable calculation method. The feature of CNN is that there are many convolutional layers, and when the input of each layer is convolution, there are multiple sets of weights convolution with the input feature.

In this paper, tiny-YOLO object detection algorithm is implemented on FT 7002 processor. It is applied to automatic driving applications. It has high requirement for real-time. Sparse mode can be used to reduce multiplication times and speed up data processing. The activation function used in the convolutional layer in tiny-YOLO is leaky, as follows:

$f(x)=\alpha x$，$(x<0)$

$f(x)=x$，$(x>=0)$

When the eigenvalue is activated, the multiplication operation is required when the eigenvalue is less than zero. Based on the fault tolerance of the neural network, in the original network, we improved the activation function of the first 3 convolutional layers to Relu. That is, after data entry, the number of output greater than 0 is the number itself, and the output less than 0 is 0. This can save the multiplication operation resulting from the calculation of the leaky function.

Using pytorch to test tiny-YOLO on PC, changing the activation function of 3 convolutional layers to Relu, the time consumed by program operation is significantly reduced, and the consumption time on the FT processor will also decrease. As a result, see Table 1.

Table 1: test on PC

leaky	Relu
55(ms)	49(ms)

The multiplication operation is reduced without reducing the accuracy of the network, and the output value has a large number of zero, which can make use of the sparsity of the network.

When the output results are broadcast for the next layer of calculation, it is judged that if the input data is 0, the operation unit will be closed and output automatically 0. If the data is nonzero, then continue to calculate. This method reduces the multiplication times greatly, not only reduces the invalid 0 multiplier operation to reduce the power consumption, but also improves the computing efficiency.

5 CONCRETE REALIZATION

5.1 ADDRESS ALLOCATION

YOLO is a multi-layer convolutional neural network. After calculating the results of the convolutional layer, the calculation of the pooling can be carried out directly. There is no need to wait for all the results to be calculated. The pooling layer calculates all the results to calculate the convolution, so it needs to wait for all the data to be calculated and then carry out the next operation.

The weight and input feature are imported into DDR in advance, and the weights are first introduced into the core in the calculation, and the different cores can calculate the different channels in parallel. The input feature is broadcast to the core, and then the result is introduced into the DDR after the result is calculated.

Because of the huge volume of data and the complex structure of neural network, the address assignment is rather complicated. We need to address the partition, use the parameterized representation method to facilitate the call, and improve efficiency when programming. The processor is divided into DDR addresses and internal addresses. For DDR addresses, a block address is used to store weights, and a corresponding address for each core is used to input and output results. which facilitates parallel operation. In multi core debugging, All the subfunctions that are called are the same. It is very convenient to change the address parameters corresponding to the multi core.

Ping-pong structure is used for the division of the address in the core.The address space inside the kernel is limited, and the intra core space is divided into two parts. Input feature map stored in an area, the calculation results are stored in another piece, After all the calculations are completed, the calculation results are used as new inputs to participate in the next calculation. So alternately, in the process of convolution and pooling, the efficiency can be greatly improved.

5.2 SINGLE BROADCAST MULTI COMPUTING

For a space sensitive architecture such as FT, we discusses how data flow can be invoked from memory at low cost and efficiently reuse data to reduce energy consumption.

Due to the limited address space in the kernel, the input feature graph in DDR is sent to each core by broadcasting. However, the broadcast consumes a lot of time.

In order to improve the efficiency, the input feature is broadcast, at the same time, it is calculated with multiple groups of weights, one broadcast, and multiple calculations. The number of broadcasting times of the input feature graph is reduced, and the efficiency is greatly improved.

After testing, the single core calculated results required 110ms, and the processor could reach a speed of 36 frames per second.

6 CONCLUSION

In this paper, a general and efficient method to deal with convolutional neural networks is proposed, which is applied to the general convolutional neural network. And it can get very high performance. The object detection algorithm YOLO is implemented on the FT processor. Finally, the speed of 36 frames per second can be processed to meet the needs of auto driving car.

REFERENCES

1. Vivienne Sze,Yu-Hsin Chen.Efficient Processing of Deep Neural Networks:A Tutorial and Survry in *Proceedings of the IEEE*,2017,105(12):2295-2329
2. Joseph Redmon,Santosh Divvala,Ross Girsick.You Only Look Once:Unified,Real-Time Object Detection in *IEEE Conference on Computer Vision & Pattern Recongnition* ,2016:779-788
3. Joseph Redmon,Ali Farhadi. YOLO9000: Better, Faster,Stronger in *IEEE Computer Society*
4. Wen W,Wu C,Wang Y,et al.Learning Structured Sparsity in Deep Neural Networks in *Advances in Neural Information Processing Systems*,2016
5. Faraone J,Fraser N,Gambardella G,et al.Compressing Low Precision Deep Neural Networks Using Sparsity-Induced Regularization in Ternary Networks in *International Conference on Neural Information Processing*,2017:393-404
6. Liu J,Chi L H,Xie L C,et al.A peak performance model for Matrix multiplication on general-purpose DSP in *Hunan Daxue Xuebao/journal of Hunan University Natural Sciences*,2013,40(11):148-152
7. Shen J,Huang Y,Wang Z,et al.Towards a Uniform Template-based Architecture for Accelerating 2D and 3D CNNs on FPGA[C]//*Hunan Acm/sigda International Symposium*. ACM,2018:97-106
8. Huang D,Luo L,Chen Z,et al.Applying Dectection Proposals to Visual Tracking for Scale and Aspect

Ratio Adaptability[J]. *International Journal of Computer Vision*,2017,122(3):524-541

9. Wang Y,Chen S,Chen H,et al.Dual-Core Framework:Eliminating the Bottleneck Effect of Scalar Kernels on SIMD Architectures[J].*Ieice Transactions on Information & Systems*,2013,E96.D(2):365-369
10. Wang Y,Chen S,Wan J,et al.A multiple SIMD,multiple data(MSMD)architecture:Parallel execution of dynamic and static SIMD fragments[J].2013:603-614