Convolutional Neural Network Acceleration Design Based on FPGA

Xixiong Zhou[1] Sheng Zhong[1]
1.College of Automation, Huazhong University of Science and Technology, Wuhan, China;

Introduction

Compared with traditional methods, the performance of neural network-based methods is usually much better than that of traditional algorithms in the domain field. And it has been applied in many fields, such as speech recognition, target detection, target segmentation and so on. However, the computational complexity of network-based methods is usually very large, which limits the application of neural network methods in embedded scenarios, such as VR/AR, mobile phones, smart security, automatic driving and so on. In order to solve this problem, this thesis aims to explore an engineering feasible and embedded platform-oriented convolution network deployment scheme, and verify it with FPGA. .

Considering that the computational process of FPGA needs to use fixed-point calculation, which conflicts with the floating-point operation process of existing network, an integer quantization method for convolutional networks is improved. This method uses statistical extremum to dynamically quantify the input, output and weight of the network, ensuring that the network forward derivation uses integer calculation only and solves the numerical calculation conflicts on the FPGA. Then, based on HLS, the hardware acceleration architecture of YOLO algorithm is designed.

Keywords: YOLO, HLS, FPGA, Network Quantization, Hardware Acceleration

Design and Implementation

Block design

This paper proposes an improvement of the IAO algorithm proposed by Google in 2018 CVPR, which reduces the computational redundancy of the quantization method, and makes the forward propagation process of the network only involve integer calculation. This method not only satisfies the computing characteristics of hardware platforms such as FPGA and ASIC, but also ensures the accuracy of network quantization. In addition, a convolution network accelerator based on FPGA is designed by using various parallel means according to the simplified network forward propagation calculation flow. This paper adopts a "PL+PS"-based convolutional network deployment framework. PL refers to the programmable logic part of ZYNQ, which is mainly responsible for the forward propagation process of the convolutional network. According to the specific neural network structure, three basic acceleration units are designed to realize the forward propagation process of the neural network in parallel. PS refers to the ARM processor part of ZYNQ, which is mainly responsible for system scheduling, including the configuration of network structure parameters, as well as the weight, input, output data movement and visualization of results. Specifically, it includes the following:

first, drawing on NVIDIA's quantization scheme, the corresponding absolute values are calculated according to the distribution of values in the network, and the weight, input and output of the network are dynamically quantized, and the quantization process is not set with saturation cropping. For the sake of simplifying the calculation, the quantization process is performed using a simplified quantization formula: A = scale_A × Q_A.A represents the floating point number before quantization, scale_A represents the scale of quantization, and Q_A represents the fixed number of points after quantization. In order to make the network performance not seriously degraded, according to Google's quantitative framework, the training process is used to compensate for the loss caused by the quantitative operation. The statistical method of calculating maxium absolute value is EMA (exponential moving mean). By the way it's guaranteed that the quantized network forward propagation process only involves integer calculation.

second, after the network is quantized, according to the size of the input/output feature map of each layer, the number of input/output channels, the activation type, the pooling type, and the attributes of different network layers, designing FPGA-based network accelerators in a block-wise manner by combining different parallel computing methods. Specifically, the fp_conv module is processed in the "input parallel" manner for the first layer of the convolutional layer.; the conv_3 module is designed for the 3x3 convolutional layer of the middle layer in form of "output parallel + input parallel + pixel parallel", and conv_1 is designed for the last layer of convolutional layer in the form of "pixel parallel + input parallel".

third, in order to realize pipeline processing in the processing unit, the sliding window is repeatedly used for convolution operation and pooling operation. In the handwritten RTL code, the way to achieve this is to use the front and rear cascaded fifo and multiple sets of cascaded registers to get the corresponding window data, and get the data of the corresponding window. In order to implement this hardware processing method in HLS, it is necessary to use optimization instructions and some coding techniques

Results and Conclusions

In this paper, we improves IAO quantization method by combining NVIDIA quantization method. Compared with the original Google method, this method can reduce the network computation by 75% compared with the original quantization method, and ensure that the accuracy loss of 8 bit quantization is less than 2%, which is similar to the quantization error of Google..

In this paper, we take YOLOv2-tiny as an example, a network acceleration architecture is designed based on HLS, and three convolution acceleration modules are designed by combining various parallel modes. The speed of the convolution accelerator is 4.6 frames/s at 123 MHz clock, 19 times faster than that of the non-optimized CPU version, and 4.87 times faster than that of the CPU version accelerated by GEMM.

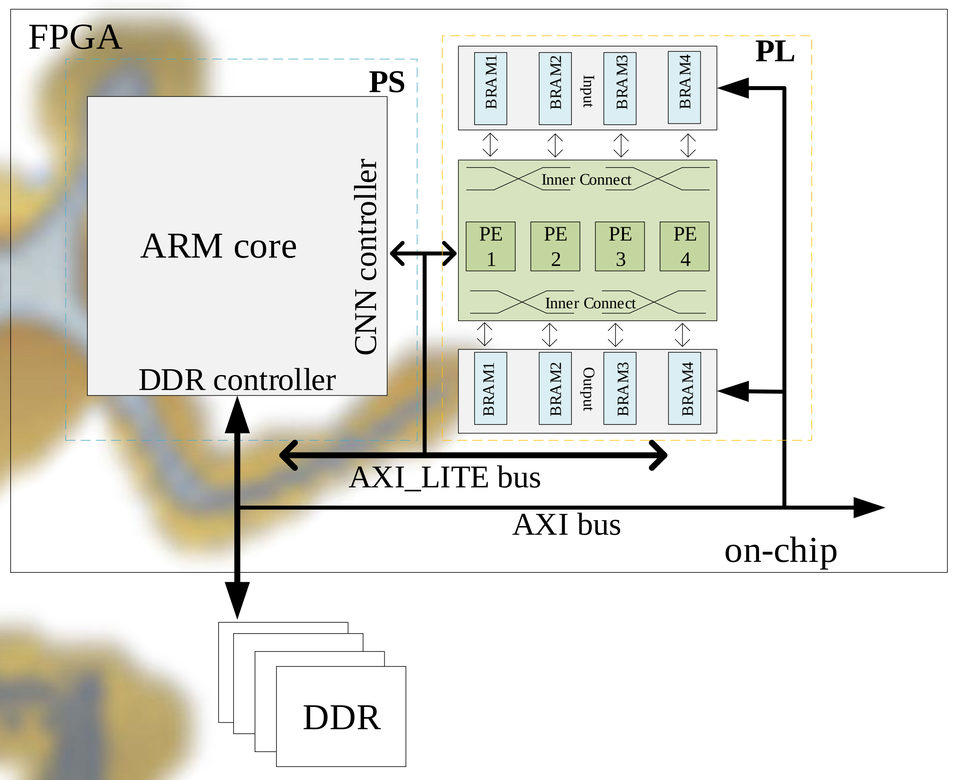

Fig. 1. acceleration architecture

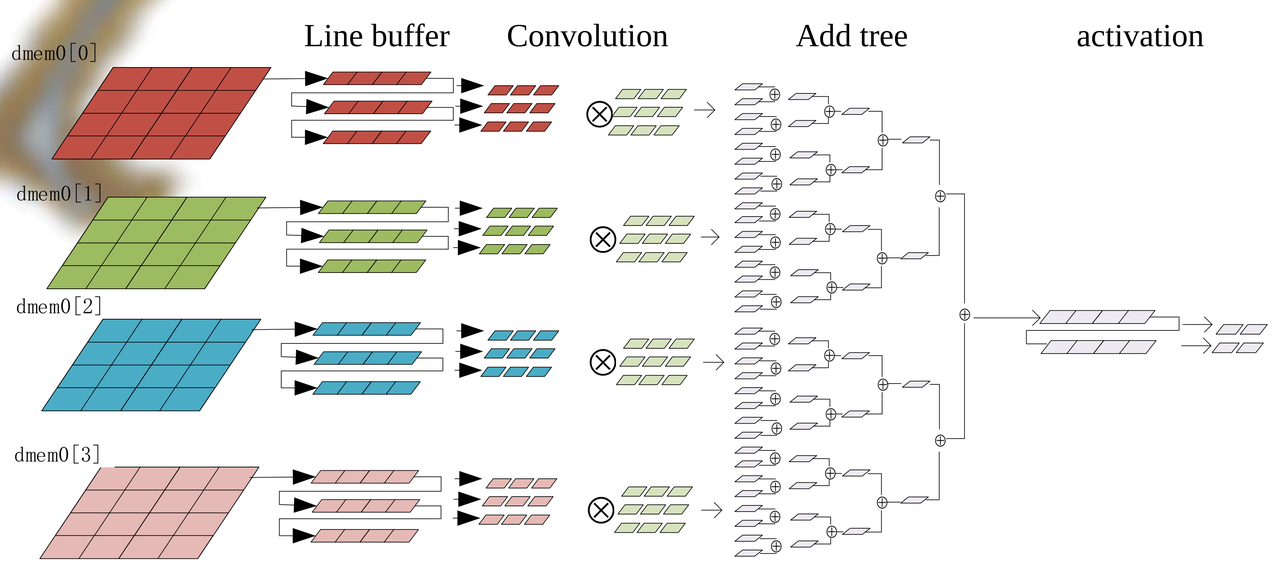

Fig. 2. pipeline of the PE

Acknowledgment

The authors acknowledge the school of Artificial Intelligence and Automation of Huazhong University of Science and Technology for providing funding.