Report for project

Wan Haochuan, Li antong

I. Project Introduction

The software code is **proj/sw/src/demo.c**.
Use **proj/tools/check.py** to check the correctness. If all correct, it will print "True", otherwise it print "False". **check.py** compares **groundtruth.bin** with specified address in **data1.bin**. **groundtruth.bin** is generated by **model.py** with original quantified model.

II. Implementation Details

A. Neural Networks

The output of conv1 need to multiply the conv1.output_scale. However, conv1.output_scale is a floating-point, which needs to convert the operation to shift. So, the conv1.output_scale is quantified to 2^{-n}. In this project, $n = 10$.

B. Hardware Platform

Besides the instructions of scalar processor from lab3, several new instructions are added.
inst_vmul_vc: convolution vs1 and vs2, save the result to vd.
inst_vadd_vr: vd[i] = (vs1[i] + vs2[i]) >>> 10 if vs1[i] + vs2[i] is positive, otherwise vd[i] = 0.
inst_vcomp: vd[i] = max(vs1[i], vs1[i+1], vs1[i+2], vs1[i+3]), where i mod 4 = 0.

C. Application Software

In **model.py**, image data are converted to im2col format and save to **data.bin**.

In **demo.c**, firstly, use inst_vmul_vc to convolution 3 channels. Then use inst_vadd_vv and inst_vadd_vr to add results of 3 channels together and keep positive value. Then, reorder the data by the order of pooling windows in memory. Finial, use inst_vcomp to pick the max value from pooling window. In **check.py**, the correctness of the calculation is checked.

III. Conclusions

In this project, conv1.output_scale is quantified to 2^{-n} for RISC-V processor design. A RISC-V processor with special instructions for convolution operation is implemented. Based on this, software is also designed to completed to calculation the first 3 layers of the provided neural network. These layers are calculated correctly and the total execution cycles is 118589.

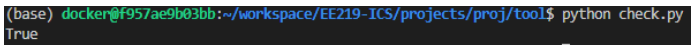

Fig. 1. The result is correct.