Homework#4

Due in Two Weeks (12/30)

Problem:

Please complete a CNN model - Lenet-5, which is shown in Fig.1. The model has eight layer. Each layer is explained below.

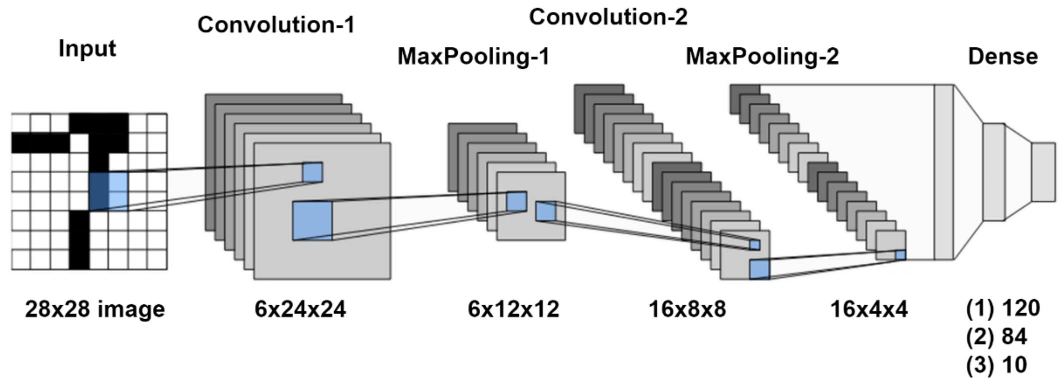

Fig.1 Lenet-5 model architecture

1. Input layer:

   The input data are 28x28 images which are from MNIST. We provide 1 input image (number 7) that you can test. Input data has been loaded and stored in the ROM as shown in Fig.2.

ROM

Index	Data
0-155	Conv_1_weight
156-2571	Conv_2_weight
2572-33411	Dense_1_weight
33412-43575	Dense_2_weight
43576-44425	Dense_3_weight
44426-45209	Input_data

Fig.2 ROM data

2. Convolution layer - 1:

   Kernel size is 5x5, output data channel is 6, stride is 1 and activation function is ReLU in this layer. Hence, there are six 24x24 output feature maps.

3. MaxPooling layer - 1:

The pooling operation is used to compress the feature maps. The MaxPooling layer adopt max pooling which is to find the maximum value of the input data within the scope of the kernel. The kernel size is 2x2 and the stride is 2. The pooling operation is shown as Fig.3.

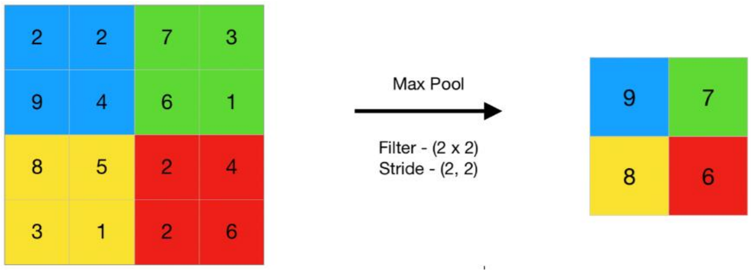

Fig.3 MaxPooling operation

4. Convolution layer - 2:

Kernel size is 5x5, output data channel is 16, stride is 1 and activation function is ReLU in this layer. Hence, there are sixteen 8x8 output feature maps. In addition, **the input data channel of this layer is 6**.

5. MaxPooling layer - 2:

The operation of this layer is as same as the MaxPooling layer - 1.

6. Dense layer - 1:

Before doing the Dense (Fully connected) layer operation, **it is necessary to flatten the output feature maps from previous layer**. The order of input data is arranged from the first feature, as shown in Fig. 4. Input neuron number is 256, output neuron number is 120, and activation function is ReLU in this layer.

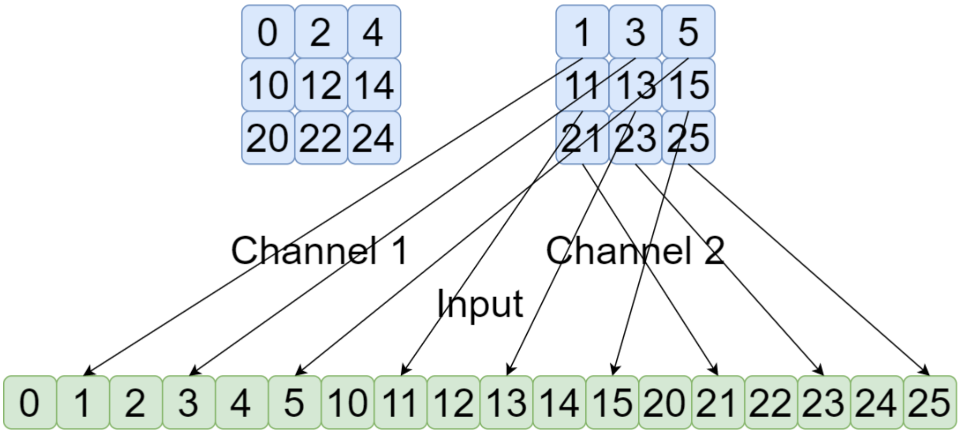

Fig.4 Flatten

7. Dense layer - 2:

Input neuron number is 120, output neuron number is 84, and activation function is ReLU in this layer.

8. Dense layer - 3 (Final Layer):

Input neuron number is 84, output neuron number is 10, and there is no activation function in this layer. The value of these neurons is the output data.

The system block is shown as Fig. 5. The ROM is used to store the weights and input data of CNN model. The convolution weight is arranged by kernel. The order of each kernel is first horizontal, then vertical and finally depth, as shown in Fig.6. The order of fully connected layer weight is arranged by neuron, as shown in Fig.7. We provide a RAM that you can store the temporary data. The RAM has 16KB entry. The RAM can read/write data. When the ram_wr becomes 0 (write valid), you can write data to RAM. When the ram_wr becomes to 1 (read valid), you can read data from RAM. The Monitor is used to print the output value. When the output_valid signal change to 1, the result is valid. When all the results are collected (10 outputs in total), the simulation will end.

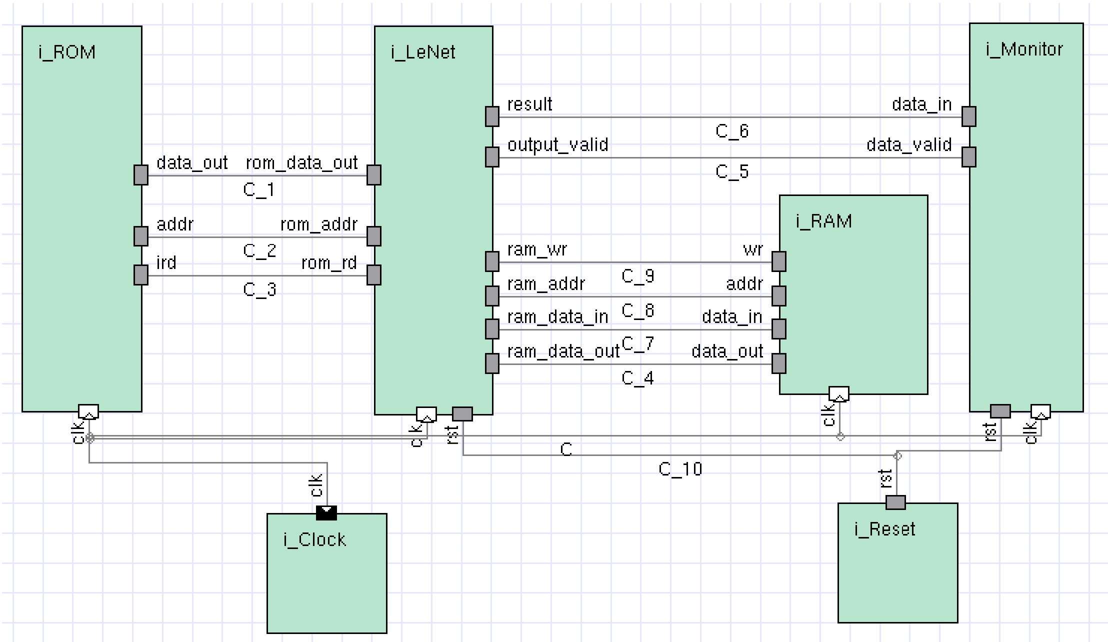

Fig.5 System block (example)

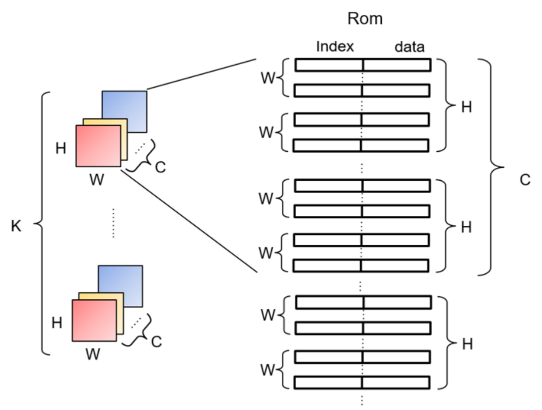

Fig.6 Kernel weight in ROM

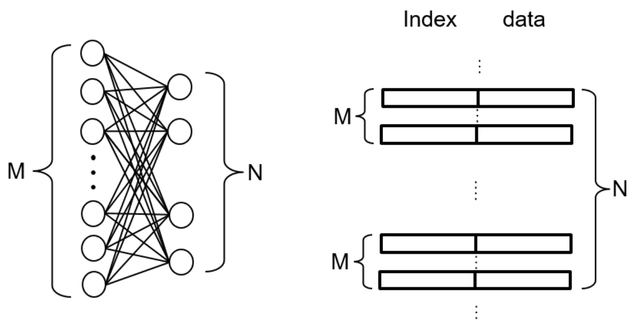

Fig.7 Dense layer weight in ROM

Finally, you must complete two version of the HW#4 by use floating-point (float) and fixed-point (16-bit) and compare the difference between the results in two version. The define.h file is used to determine the data type of the system, you can undefine/define fixed_DATA_TYPE to change the current mode. In addition, in fixed-point version system, floating-point number type **such as Double or Float cannot be used**.

Requirements:

Please upload the compressed file (.zip or .rar) which including following files to Cyber University (Compressed File Name: Team ID_Student ID_HW4)

1. Complete two version of the HW#4 (floating-point and fixed-point)
2. Source code (Include PA project file and all your source code)
3. Report (Word file Include design concept, block diagram, and simulation result of PA)