Practice 7 - Convolution Lowering SW

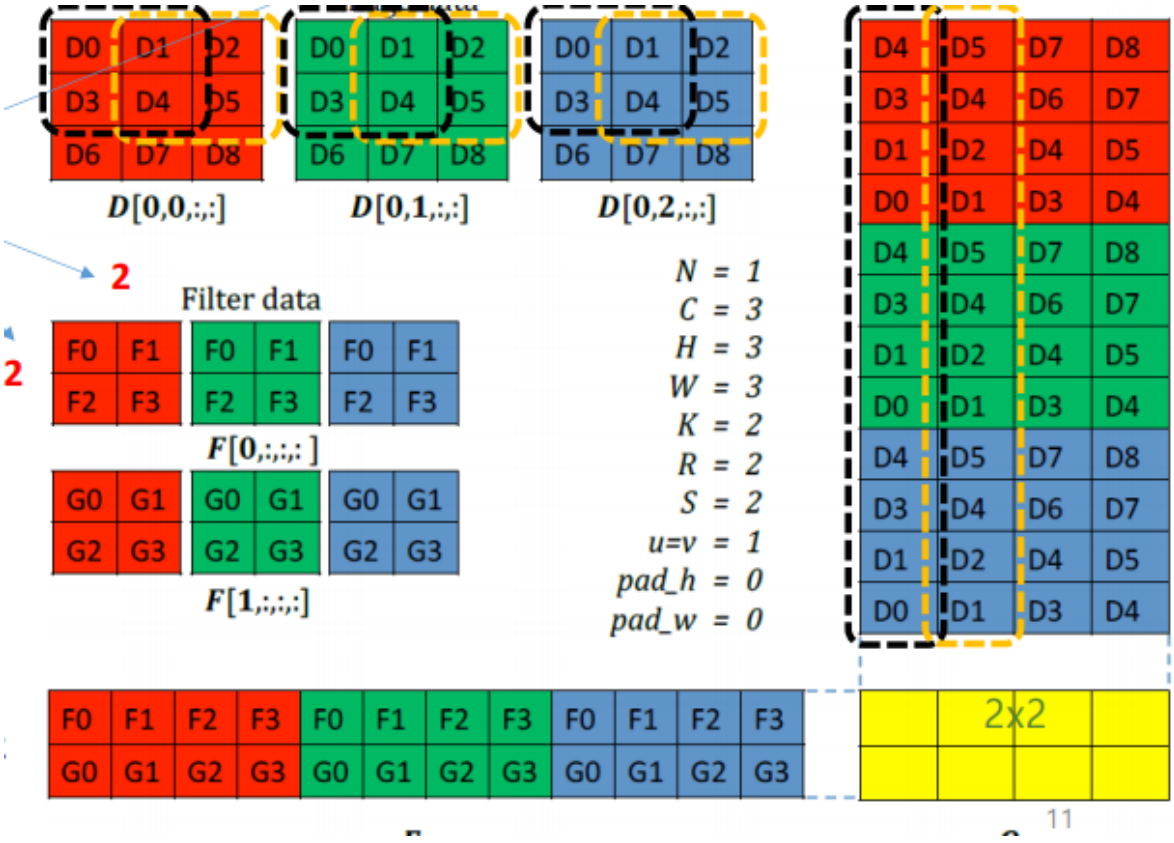
2017-19428

컴퓨터공학부 서준원

1. Introduction

Lab 2에서는 미리 학습된 weight를 가지고 matrix와 input vector를 적당한 크기로 나누어 matrix multiplication을 수행하는 c++ 코드를 작성하였다.

하지만 Verilog로 구현한 Processing element IP는 1차원 벡터들 간의 내적 곱을 수행한다. 따라서 2차원 벡터의 연산을 성공적으로 수행할 수 없다. 특히 컴퓨터 비전에서 이미지 학습을 위해 널리 사용되는 CNN을 이 IP에 사용하기 위해서는 데이터의 조작이 필요하다. Convolution을 하는 과정에서 2차원 혹은 3차원 데이터인 이미지를 matrix multiplication을 하기 위해서는 데이터의 차원을 낮추어 주어야 한다. 이를 Convolution Lowering이라고 한다. 아래 그림은 데이터를 convolution lowering하는 과정을 나타냈다.



이를 정확하게 하기 위해서 3차원 (channel, height, width) 인 이미지와 4차원인 convolution weight (channel, input channel, height, width)를 적당한 2차원 벡터들로 바꾸어 주어야 한다.

2. Implementation

벡터를 인풋으로 받아 적당한 위치에 재배치해주었다.

New weight의 경우 4차원 데이터를 2차원으로 lowering 해준다. 아래와 같은 간단한 코드로 가능하다.

```
for(int c=0; c<conv_channel; c++)
{
  // vector row new_weight[c]
  int cnt = 0;
  for(int ic = 0; ic<input_channel; ic++)
  {
      for(int h=0; h<conv_height; h++)
      {
          for(int w=0; w<conv_width; w++)
          {
              new_weights[c][cnt++] = (cnn_weights[c][ic][h][w]);
          }
      }
  }
}
```

각 Convolution Channel 마다 input, height, width를 펴주면 된다.

인풋 데이터의 경우 조금 예민하게 처리해주어야 한다. Padding을 고려해주어야 하며, 적당한 행에 weight가 곱해지는 순서대로 데이터를 넣어주어야 한다. 각 채널마다 w, h의 값을 고려하여 행을 결정해주여아 하며, 열은 순서대로 데이터를 넣어준다. 아래와 같이 그래도 간단한 코드로 구현을 했다.

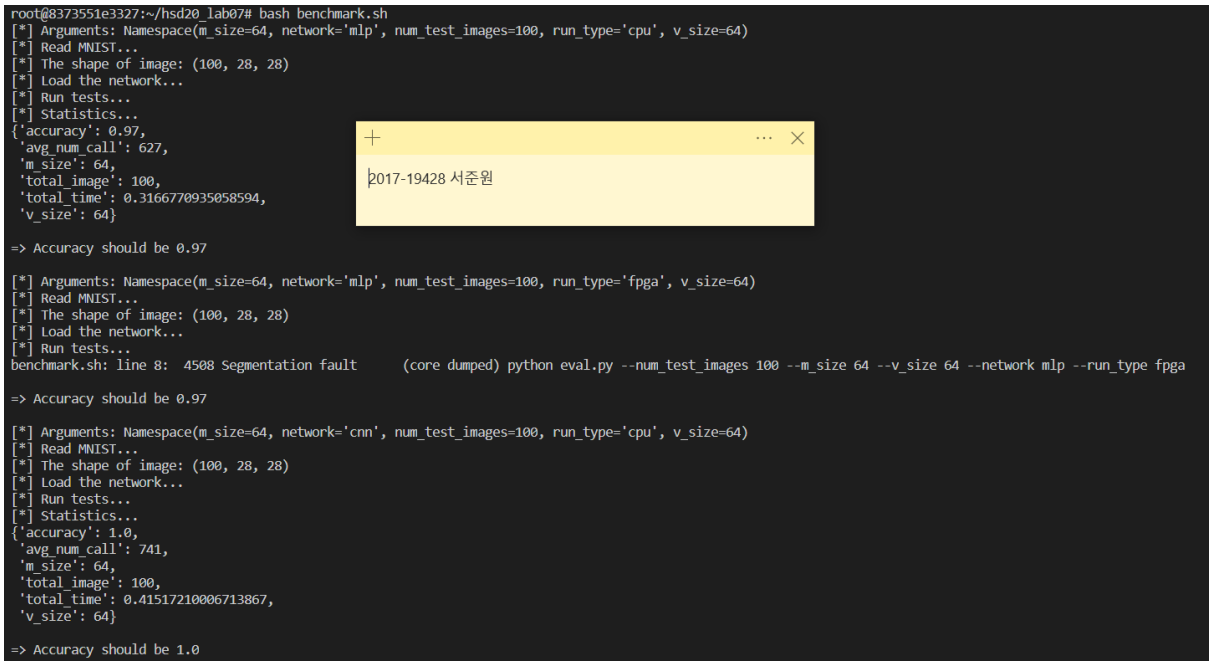
```
// first move row-wise
int cnt = 0;
for(int y=0; y<input_height-conv_height+1; y++)
{
    for(int x=0; x<input_width-conv_width+1; x++)
    {
        // input row new_input[cnt];
        for(int ic=0; ic<input_channel; ic++)
        {
              for(int h=0; h<conv_height; h++)
              {
```

```
                    for(int w=0; w<conv_width; w++)
                    {
                        int idx = w + conv_width * h + (conv_width*conv_height
) * ic;
                        new_inputs[idx][cnt] = inputs[ic][h+y][w+x];
                    }
                }
            }
          cnt++;
        }

    }
```

3. Result

구현 내용이 정확한지 테스트 하는 것은 간단하다. Matrix multiplication이 정확하다면 Test set에 대한 Classification 테스트 결과가 변하지 않을 것이다. 사진과 같이 테스트를 통과한 것을 확인할 수 있었다. Cnn의 경우와 mlp의 경우 모두 원래의 테스트 결과와 같은 정확도를 얻었다.



Mlp의 경우 97%의 정확도를, CNN의 경우 100%의 정확도를 100개의 테스트 이미지로 테스트했을 때 얻었다.