Homework #4

- Computation: max pooling

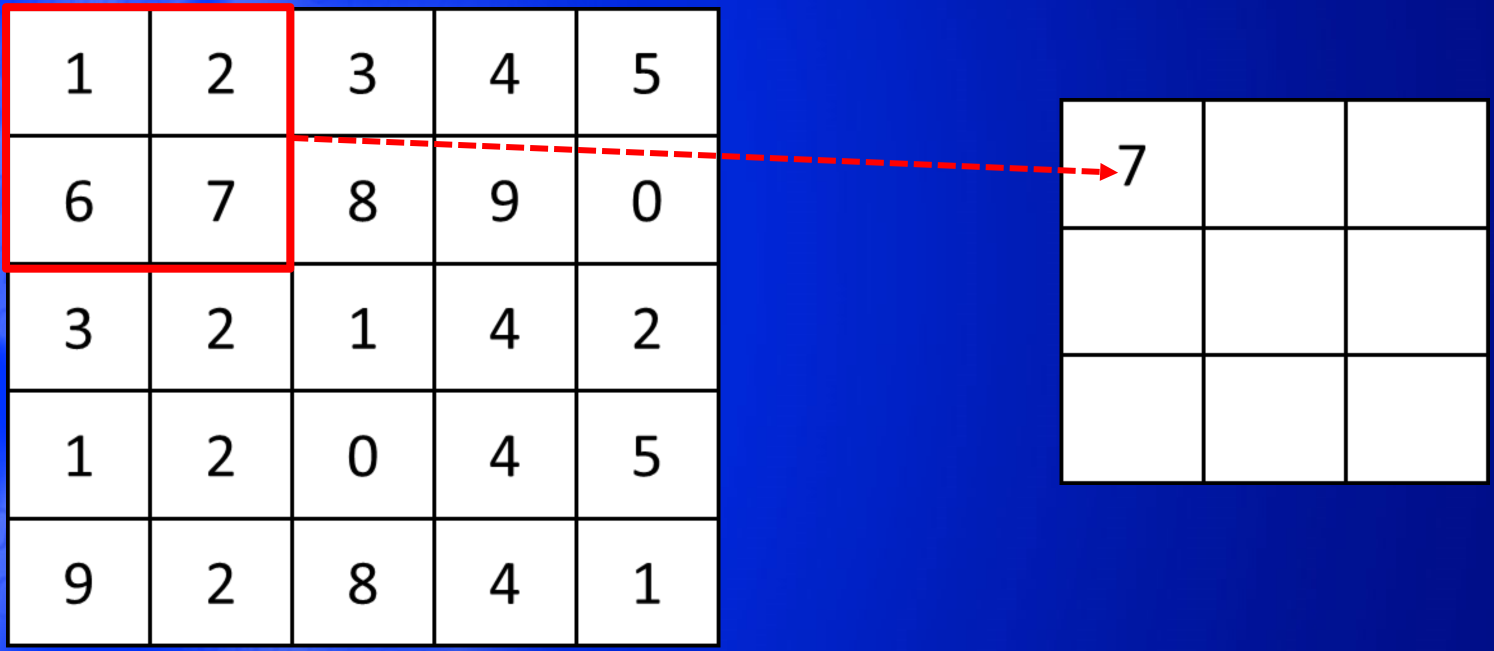
1	2	3	4	5
6	7	8	9	0
3	2	1	4	2
1	2	0	4	5
9	2	8	4	1

Input: 5x5

Kernel: 2x2
Stride: 2

Computation

Kernel: 2x2
Stride: 2

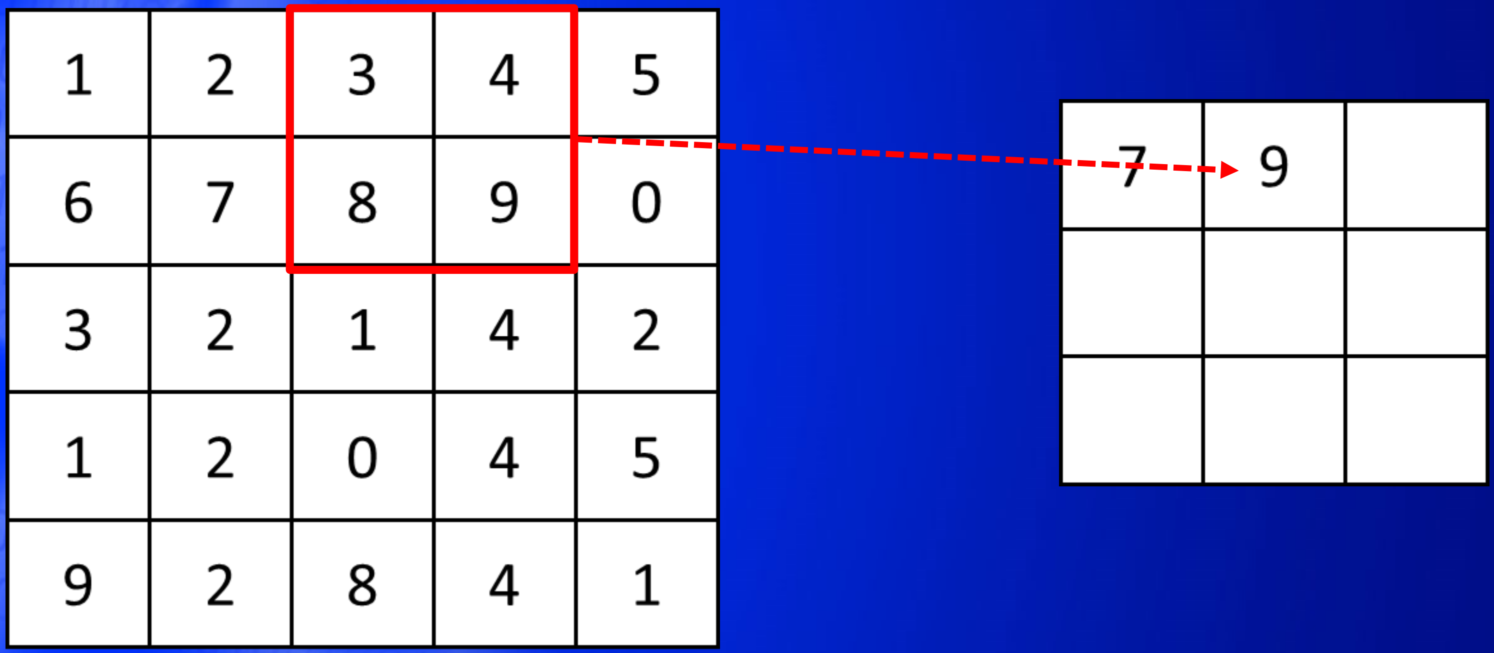


1	2	3	4	5
6	7	8	9	0
3	2	1	4	2
1	2	0	4	5
9	2	8	4	1

7		

Computation

Kernel: 2x2
Stride: 2

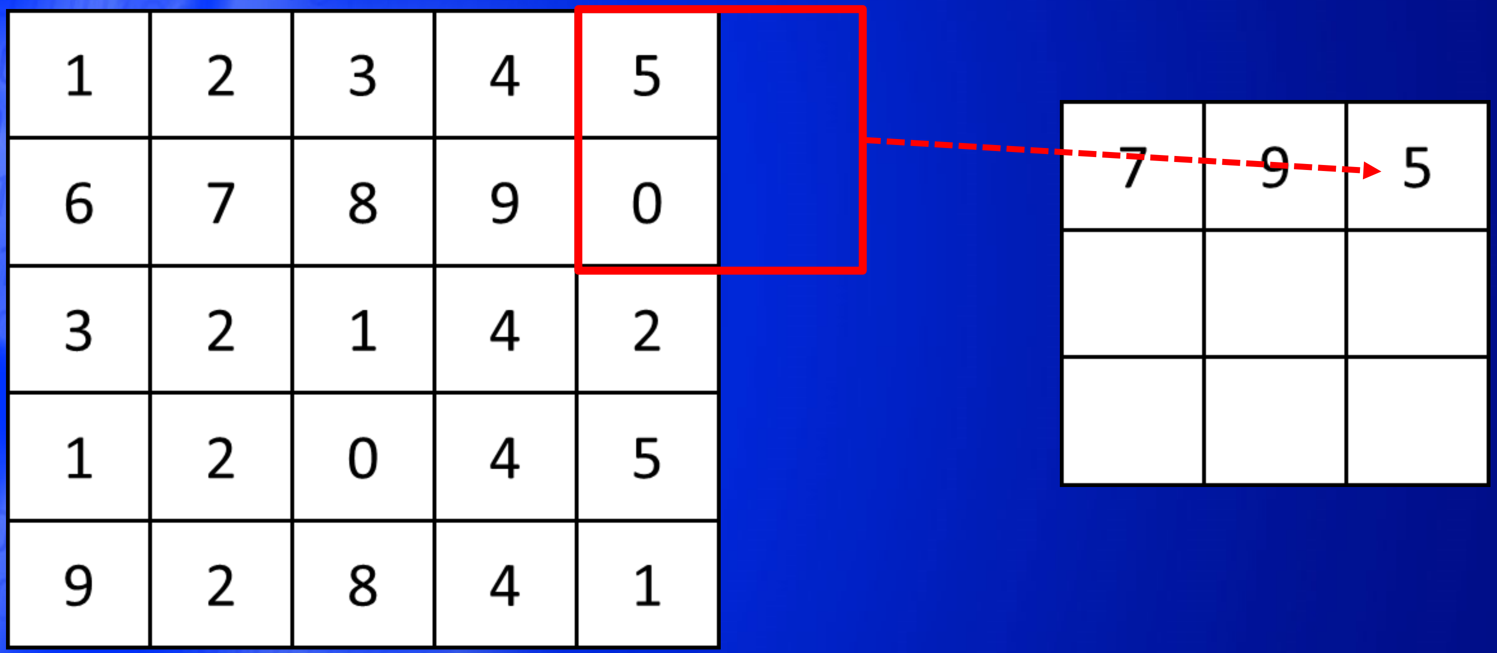


1	2	3	4	5
6	7	8	9	0
3	2	1	4	2
1	2	0	4	5
9	2	8	4	1

7	9	

Computation

Kernel: 2x2
Stride: 2

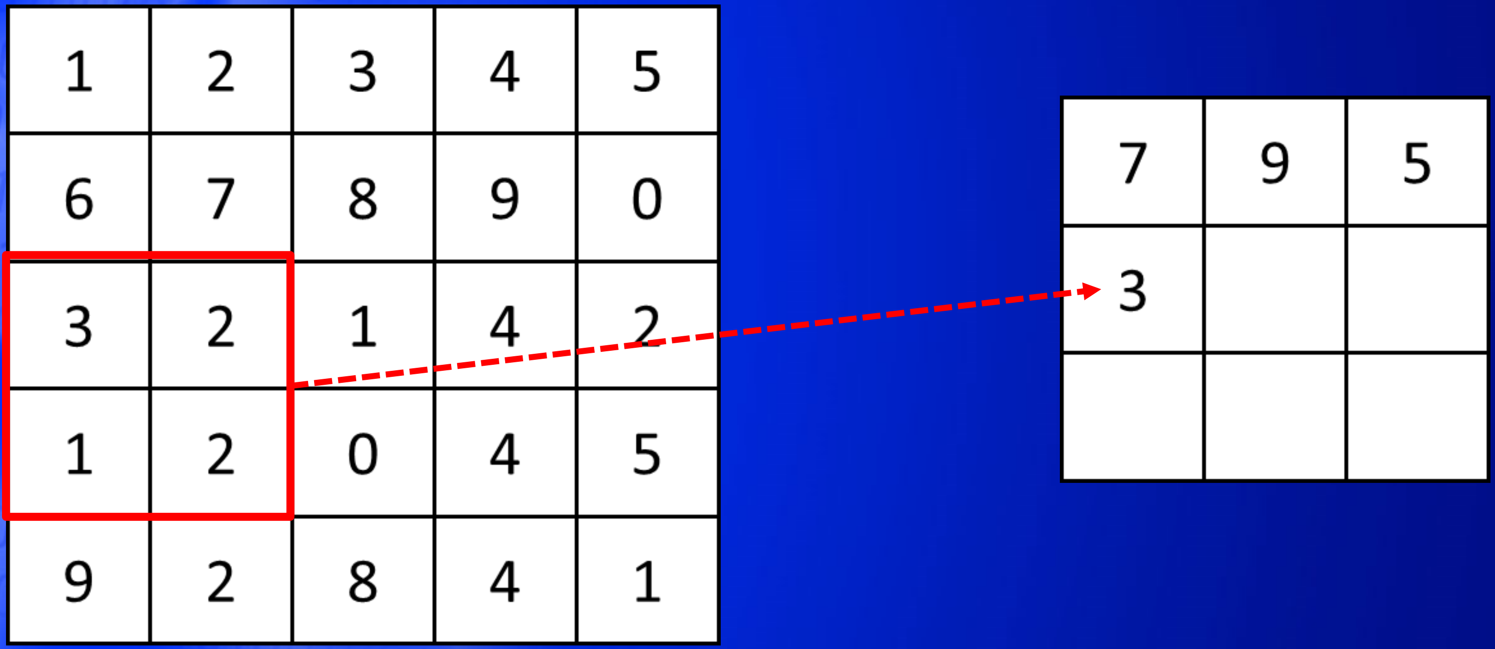


1	2	3	4	5
6	7	8	9	0
3	2	1	4	2
1	2	0	4	5
9	2	8	4	1

7	9	5

Computation

Kernel: 2x2
Stride: 2

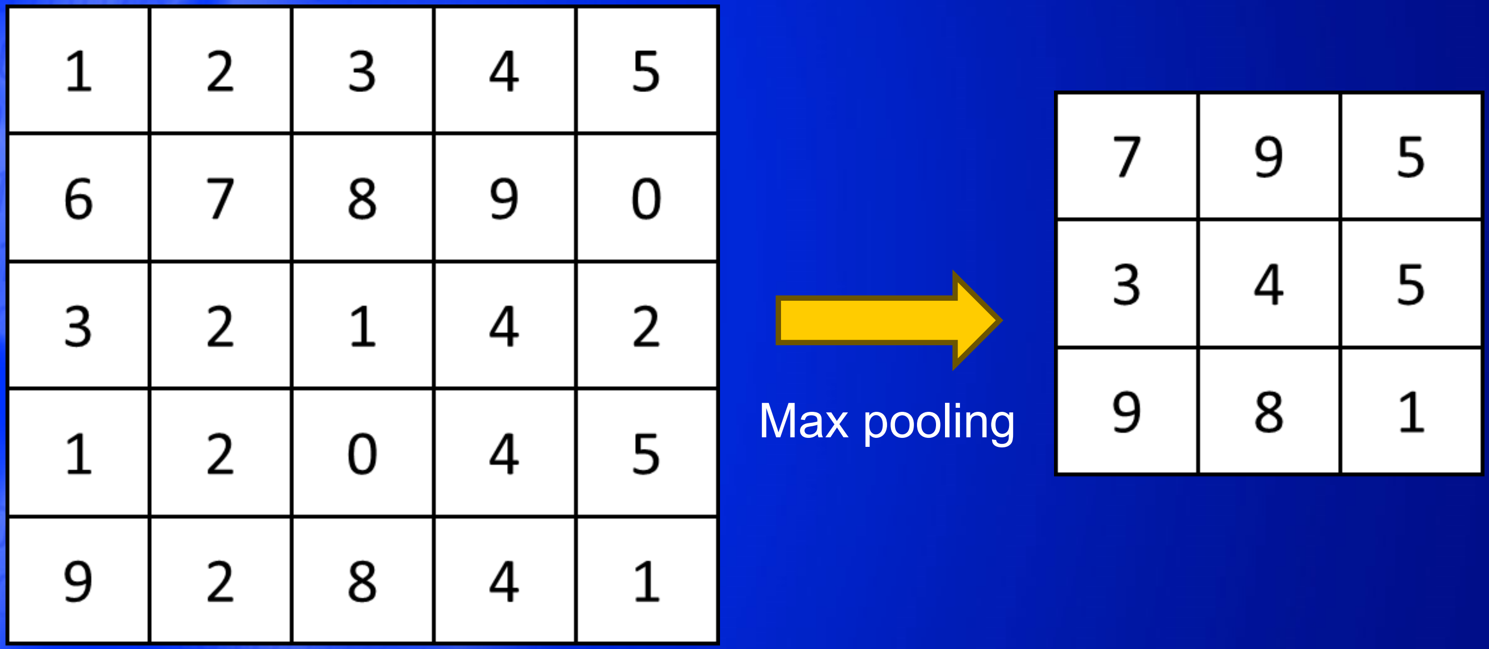


1	2	3	4	5
6	7	8	9	0
3	2	1	4	2
1	2	0	4	5
9	2	8	4	1

7	9	5
3		

Computation

Kernel: 2x2
Stride: 2



1	2	3	4	5
6	7	8	9	0
3	2	1	4	2
1	2	0	4	5
9	2	8	4	1

Max pooling

7	9	5
3	4	5
9	8	1

Pseudo Code

input: input[5][5],　　　　output: result[3][3]

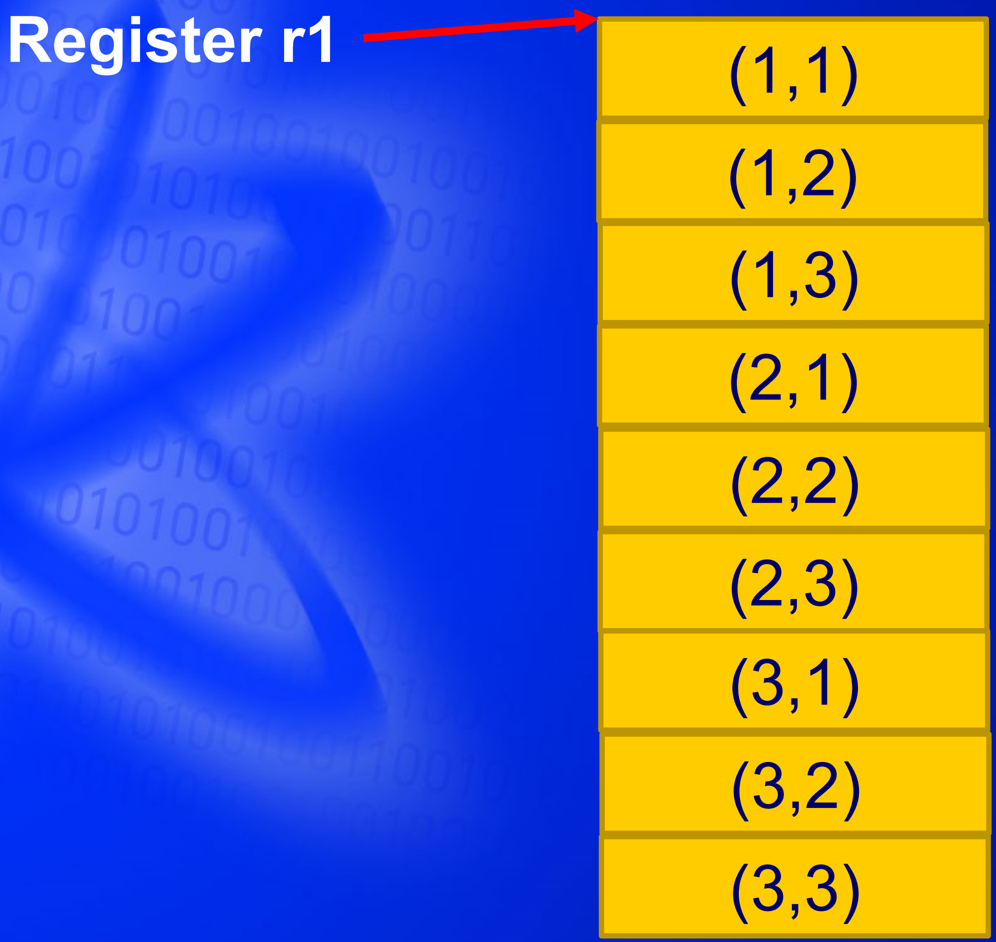
```
for (m=0; m<5; m=m+2) {  // stride = 2.
   for (n=0; n<5; n=n+2) {

      //------------------------
      // find a max value.
      tmp = 0;
      for (i=n; i<min(n+2,5); i++) {  // kernel 2x2.
        for (j=m; j<min(m+2,5); j++) {
           if (input[i][j] > tmp)
              tmp = input[i][j];
        }
      }
      // -----------------------
      result[n/2][m/2] = tmp;

   }
}
```

Homework #4 (1)

Write an ARM assembly program to do the computation, max pooling.

- input is a 5x5 matrix, kernel is a 2x2, stride is 2.
- output is a 3x3 matrix.
- Figure 1 shows the layout of the output matrix.
- Each element in input and output is a word-sized signed integer. The element values are zero or positive integers.
- The integer values of input are assigned by yourself.

Layout of Output Matrix



Register r1

(1,1)

(1,2)

(1,3)

(2,1)

(2,2)

(2,3)

(3,1)

(3,2)

(3,3)

Homework #4 (2)

- The overflow/underflow problems are not considered during the computation
- After computation, register r1 will point to the address of output’s first element
- 請勿繳交【利用編譯器所自動產生的組合語言程式】
- 請勿抄襲

Homework #4 (3)

- Program should be assembled and linked by GNU cross toolchain.
- Program can be executed under GDB ARM simulator
- 程式中應有適當的說明（註解）
- You should turn into ECOURSE2
  - “README.txt” file: 文字檔，描述你程式的內容、如何編譯程式、程式的執行環境、如何執行你的程式

    (特別註明你的執行環境是否為Mac系統)
  - “hw4.s”: Your ARM assembly program
  - “hw4.exe”: 編譯好的執行檔
  - Makefile
  - 請將欲繳交的檔案壓縮成 <hw4_學號.tar.xz>，上傳壓縮檔

Deadline: November 10 (Sunday), 24:00, 2024.