

Homework #7

(Due date: 10 November 2022)

Objective :

學習使用檔案輸出與輸入、字串與函式

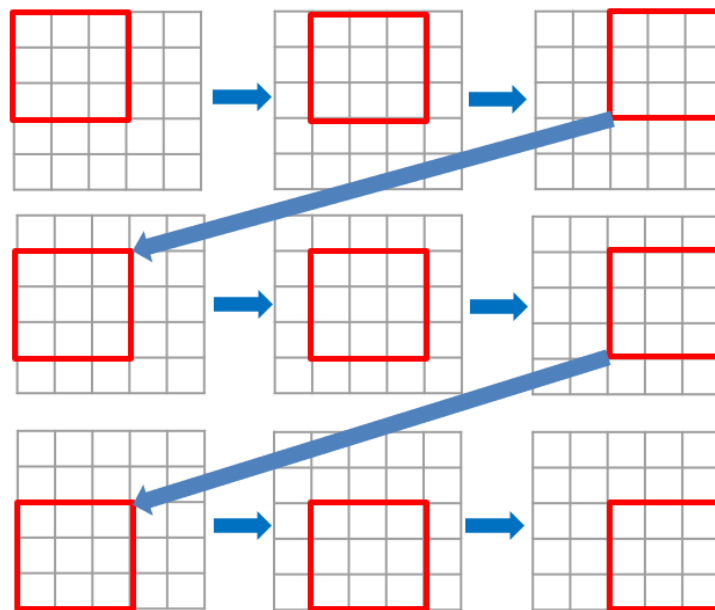
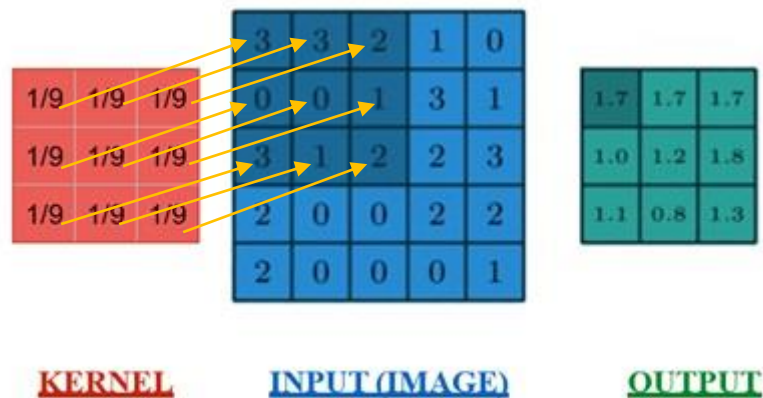
7-1

卷積神經網路 CNN 是一種前饋神經網路，對於大型圖像的機器學習有出色表現，其中 2-D CONVOLUTION 是其中重要的一層，利用 Kernel 中的 Weight 與上一層輸入的 Data 進行卷積運算，再通過啟動函數可得到本層的輸出。

請依照以下要求設計一個程式來實作 2-D CONVOLUTION：

1.請先利用 `srand(5)`，作為時間種子產生-10~+10 的亂數矩陣(16*16)並將之輸出成 `Matrix.txt`。並且輸出一個元素皆為 1 的 bias 矩陣(14*14)，`Bias.txt`，最後將兩者印出。

2.利用前面亂數產生的矩陣與 bias 與附檔 `kernel.txt` 來進行卷積運算，矩陣卷積運算流程如下：



運算結束後再加上 bias 會得到一 14*14 之矩陣，將之印出
並輸出成 Conv.txt

3.利用第 2 點的輸出經過啟動函數 $\text{ReLU}(y = \max(0, x))$ ，
後得到結果，印出結果並將之輸出成 7-1_Result.txt。

注意：

1.你需要交的檔案有 Matrix.txt、Bias.txt、Conv.txt、
7-1_Result.txt 與第一題的.c 檔

2.卷積運算的輸出需捨入到最接近的整數

如 1.66 進位至 2 / 1.47 捨去小數點後數值為 1

3.ReLU 為負值歸零函數，即正數保留原值、負數將其歸 0

輸出範例：

```
Input :
 4  3  5 -1 -10  0 -3 -1  2 -10  4 -9  3  6 -4 -3
-7 -3  0  1  7  2  3  1  6 -1  2 -7 -8  1  1 -1
 9  0  3  5 -8 -3 -7  4 -4 -4  4 -5  5  9 -10  5
-4  2 -6 -5  1  9  3  9  0 -6  7  6 -2 -4  8 -9
 6  0 -1  0  3  1 -4  5  0 -2 -4 -8  5  5 -8 -10
 3 -10 -1  3 -5  0  6  9  3 -7 -5 -4 -6  7  8  8
-10  9 -3 -5 -10 -4  1 -10  4  8 -5 -7  3  3 -1  9
 6 -9  5  2 -7  5 -4  9 -4  0 -4 -9  5 -4  8  2
-8  1 -3 -8  6  7  1  6 -2 -9 -5 -1 -4  1  3 -3
-1 -9 -4  3 -8 -6  1 -1 -7  4 -3  2  2  6  6  6
 7  7 -5  3 -3  8 -4  2 -2 -3 -2  9 -9  8 -8 -5
-4  5  6  3 -1  2 -8 -2  4 -9  1 -10 -3  4  8  0
 0 -10  2 -6  5 -4 -2 -3  6  7  1  4 -4 -4 -2  6
-8  0  5  9 -6 -4  6  4  1  3  1 -10 -5  8 -5  3
 1 -4  4  2  2  3 -10 -8  0 -10  0  8  8 -3  8 -5
-4 -6 -6 -2 -6  6  7 -10 -6 -10  5 -1 -10  5 -1 -8

Bias :
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1

Kernel :
0.330 0.330 0.330
0.540 0.890 0.540
0.330 0.330 0.330
```

```

Convolution :
  2   5   4   3  -2   2   4   2   2  -8  -8  -6   1   3
  2   3   2   0  -2   4   6   0   0   0   1   3   5  -2
  3  -4  -5   3   5  12   9   2  -4   1   8   4   3  -2
 -2  -6  -2   5   6  13  13   5  -6 -11  -7   3   7  -3
 -6  -4  -5  -7  -3   8  13   6  -6 -15 -14  -2  11  16
  0  -3 -11 -13  -9   3   1   7   2 -12 -14  -2  11  16
 -6  -1  -6  -6   0   5   5   1  -6 -14 -13  -3   6  11
 -8  -9  -7   2   5  10   4  -2 -15 -12  -7  -2   9  11
-10  -7  -6  -3  -3   5  -3  -6  -8  -2  -2   8   8  12
  6   3  -1   2  -2  -2  -5  -5  -7  -2  -1   2   7   5
  7   8   5   4  -2  -7  -4   1  -2  -3  -9  -4   1   9
 -5   3   5   1  -5  -5   2   9  11   0  -6  -8  -3   6
 -3   6  11  -1  -5  -2   3   3   6   1  -4  -2   4   2
 -6   3   4   4   0  -8 -11 -15 -10  -4   6   7   4   4

Result :
  2   5   4   3   0   2   4   2   2   0   0   0   1   3
  2   3   2   0   0   4   6   0   0   0   1   3   5   0
  3   0   0   3   5  12   9   2   0   1   8   4   3   0
  0   0   0   5   6  13  13   5   0   0   0   3   7   0
  0   0   0   0   0   8  13   6   0   0   0   0  11  16
  0   0   0   0   0   3   1   7   2   0   0   0  11  16
  0   0   0   0   0   5   5   1   0   0   0   0   6  11
  0   0   0   2   5  10   4   0   0   0   0   0   9  11
  0   0   0   0   0   5   0   0   0   0   0   8   8  12
  6   3   0   2   0   0   0   0   0   0   0   2   7   5
  7   8   5   4   0   0   0   1   0   0   0   0   1   9
  0   3   5   1   0   0   2   9  11   0   0   0   0   6
  0   6  11   0   0   0   3   3   6   1   0   0   4   2
  0   3   4   4   0   0   0   0   0   0   6   7   4   4

```

7-2

有一個機器人位於 (0,0) 的位置，給予一個字串"UDLR"代表這機器人的一連串行動，"U"意思是 UP 表示機器人往上走，"D"意思是 DOWN 表示機器人往下走，"L"意思是 LEFT 表示機器人往左走，"R"意思是 RIGHT 表示機器人往右走。

上述字串表示機器人往上後往下，又往左後往右，所以機器人回到了原點。

附檔 **Test.txt** 有 21 個字串，請設計一個程式來判斷通過每一個字串的行動後，機器人是否能夠會到原點，可以則回報 **True**，不行則回報 **False**，且須將結果印出，並輸出成 **7-2_Result.txt**。

注意： 你需要交 **7-2_Result.txt** 與第二題的.c 檔

輸出範例：

```
第1個字串 : False
第2個字串 : False
第3個字串 : True
第4個字串 : True
第5個字串 : True
第6個字串 : True
第7個字串 : False
第8個字串 : False
第9個字串 : True
第10個字串 : False
第11個字串 : True
第12個字串 : True
第13個字串 : False
第14個字串 : True
第15個字串 : False
第16個字串 : True
第17個字串 : False
第18個字串 : False
第19個字串 : False
第20個字串 : False
第21個字串 : True
```

Please note: : users of visual studio add the following code to your program's beginning

```
2 #pragma warning(disable:4996)
```

繳交格式及規定：

程式重點地方請加註解,給分也會酌量參考註解。

請繳交 .c 檔即可。

c 檔的檔名一律統一,以學號為檔名壓縮成一個以學號為名的壓縮檔上傳,

上傳請一律繳交壓縮檔。

Example:

若學號為 B113040000, 則.c/.cpp 檔名為 B113040000.c,

而壓縮檔名為 B113040000.rar。

繳交期限 2022.11.10 (四) 09:10 分之前,逾期一律不收,

無輸入輸出及逾期者一律以 0 分計算。

作業請上傳中山網路大學 網大上傳方式:

1.點選要繳交的作業,選擇「進行作業」。



2.依照流程上傳檔案。

