

Readme

1. 每個 Sparse matrix 裡都只能存 100 個數值
2. **Sparse matrix** 的位置從 **(0, 0)** 開始算
3. 輸出矩陣的第一行都是此 sparse matrix 的總行數、總列數、共有幾個數
4. 需要先輸入各個矩陣的數值

```
Please input total row & total col of matrix W: 6 6
Please input how many number is in the matrix W: 8
Please input W matrix(row, col, value):
0 0 15
0 3 22
0 5 -15
1 1 11
1 2 3
2 3 -6
4 0 91
5 1 28
Please input total row & total col of matrix X: 6 1
Please input how many number is in the matrix X: 4
Please input X matrix(row, col, value):
0 0 2
2 0 3
3 0 1
5 0 4
Please input total row & total col of matrix B: 6 1
Please input how many number is in the matrix B: 5
Please input B matrix(row, col, value):
0 0 3
1 0 6
2 0 9
4 0 1
5 0 5
```

5. 執行程式且輸完 sparse matrix 的資料後，會出現這個指示列，直接選擇需要做的動作輸入

```
0: exit!
1: Can output the values of the specified rows and columns of the specified matrix.
2: Computes the product  $W X$  of the sparse matrix  $W$  and the matrix  $X$ .
3: Calculation of the fully connected layer  $Z = W X + B$ .
4: Perform ReLU activation function on  $Z$ 
5: Perform Sigmoid activation function on  $Z$ 
6: Multiple fully connected layer ( $W$  and  $B$ ) inputs can be specified, and their dimensions can be different (e.g.  $m * n$ , then  $n * k$ ), and multiple.
```

0: 結束程式

1: 選擇特定的 sparse matrix 的某一行列位置輸出

```
1
Please enter matrix name(w, b, x, z), row, col:w 0 0
The value at (0, 0) in the matrix w is 15.
```

如果在該矩陣的指定位置是 0 也會有輸出，不會報錯

```
1
Please enter matrix name(w, b, x, z), row, col:w 5 0
The value at (5, 0) in the matrix w is 0.
```

2: 計算 WX 並且輸出結果 (row, col, value)

第一行為例外，請看 3.

```
2
6 1 4
0 0 -8
1 0 9
2 0 -6
4 0 182
```

3: 計算 $Z = WX + Y$ 並且輸出結果 (row, col, value)

第一行為例外，請看 3.

```
3
6 1 5
0 0 -5
1 0 15
2 0 3
4 0 183
5 0 5
```

4: (需先執行過 2 或 3 才能跑) 執行 ReLU activate function，並輸出結果 (row, col, value)

運算後 value 為 0 也會輸出

第一行為例外，請看 3.

```
4
6 1 5
0 0 0
1 0 15
2 0 3
4 0 183
5 0 5
```

5: (需先執行過 2 或 3 才能跑) 執行 Sigmoid activate function，並輸出結果
(row, col, value)

運算後 value 為 0 也會輸出

第一行為例外，請看 3.

```
5
6 1 5
0 0 0
1 0 0
2 0 0
4 0 1
5 0 0
```

6: 選擇要跑幾層的神經網路、使用哪種 activate function (ReLU or Sigmoid)，
第一層的 W、X、B 一定是一開始輸入的 sparse matrix，從第 2 層開始才能
重新輸入 W, X，結束後會輸出運算後的 Z matrix 的非 0 項

6. 可使用 g++ 編譯程式

請在終端機輸入 `g++ hw1_41147046S.c -o hw1`

要執行請輸入 `./hw1`