This is a sample of the report, but applicable for all homework.

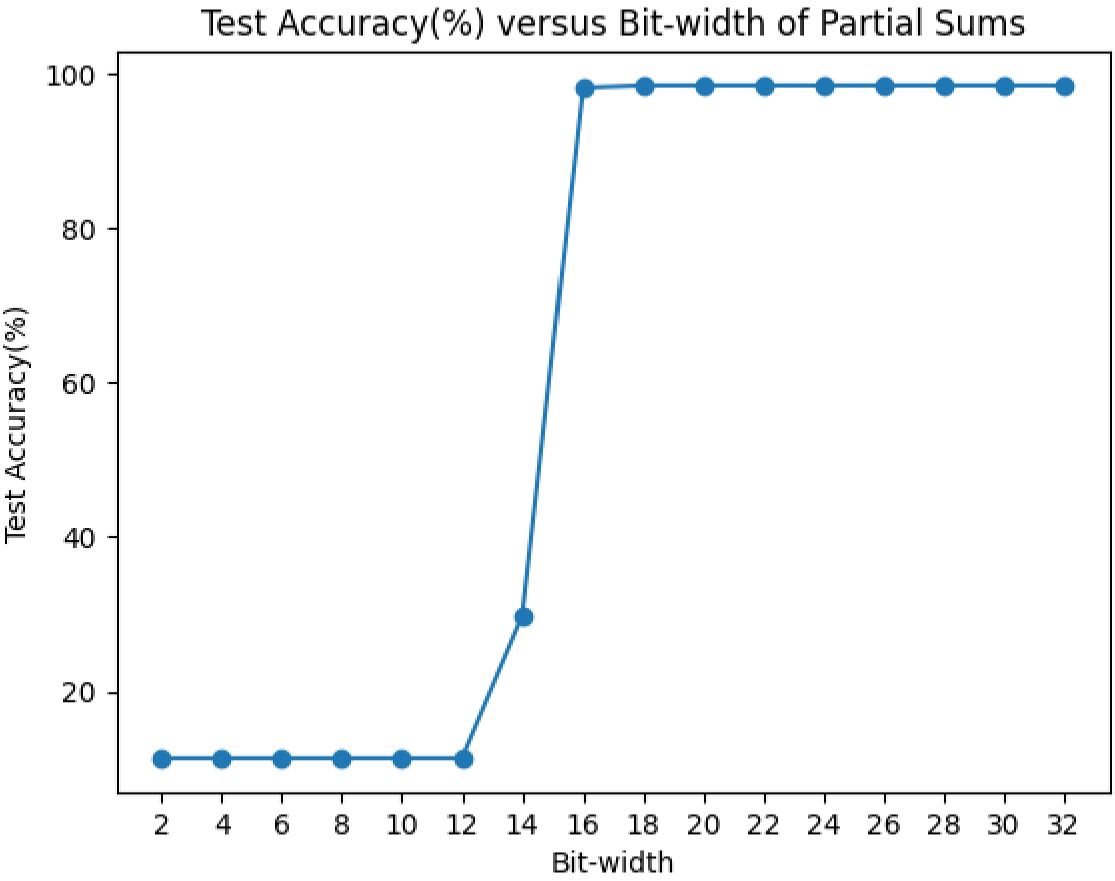
112062542 賴琮翰 This is for double verification.

Don't copy the problem statement, just write the answer.

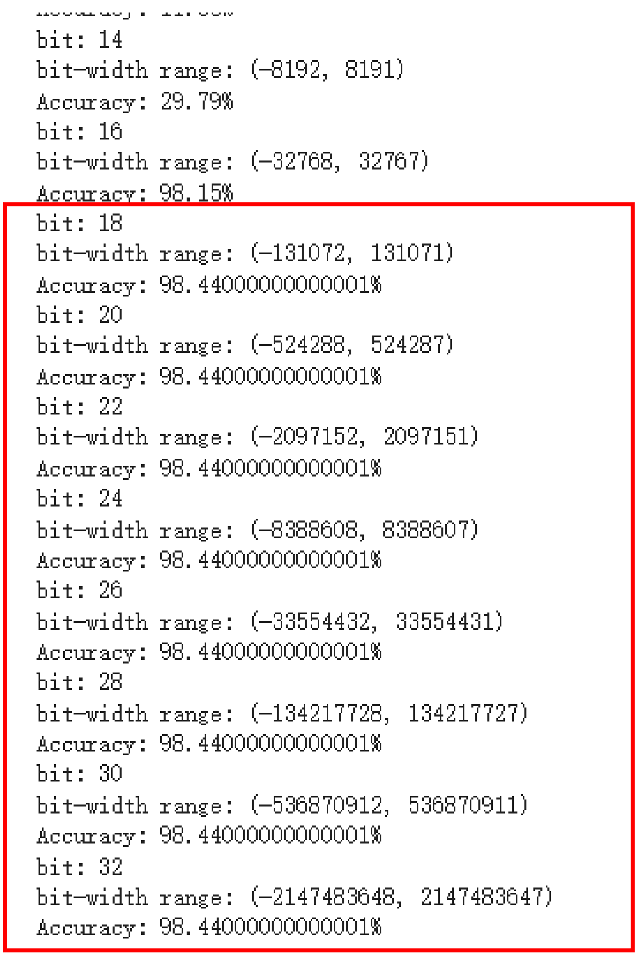
Please write down the question number in unit of sub-question.

Please write down the sub-question number even if you don't know how to solve it.

(2.1.1)

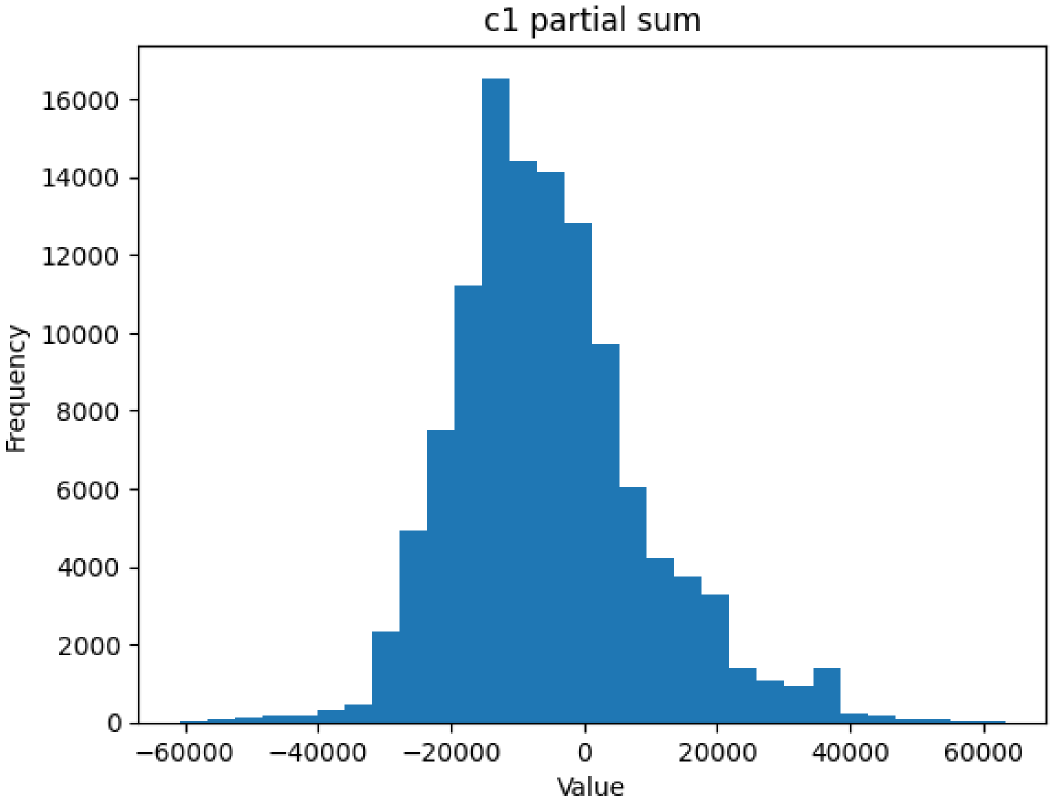


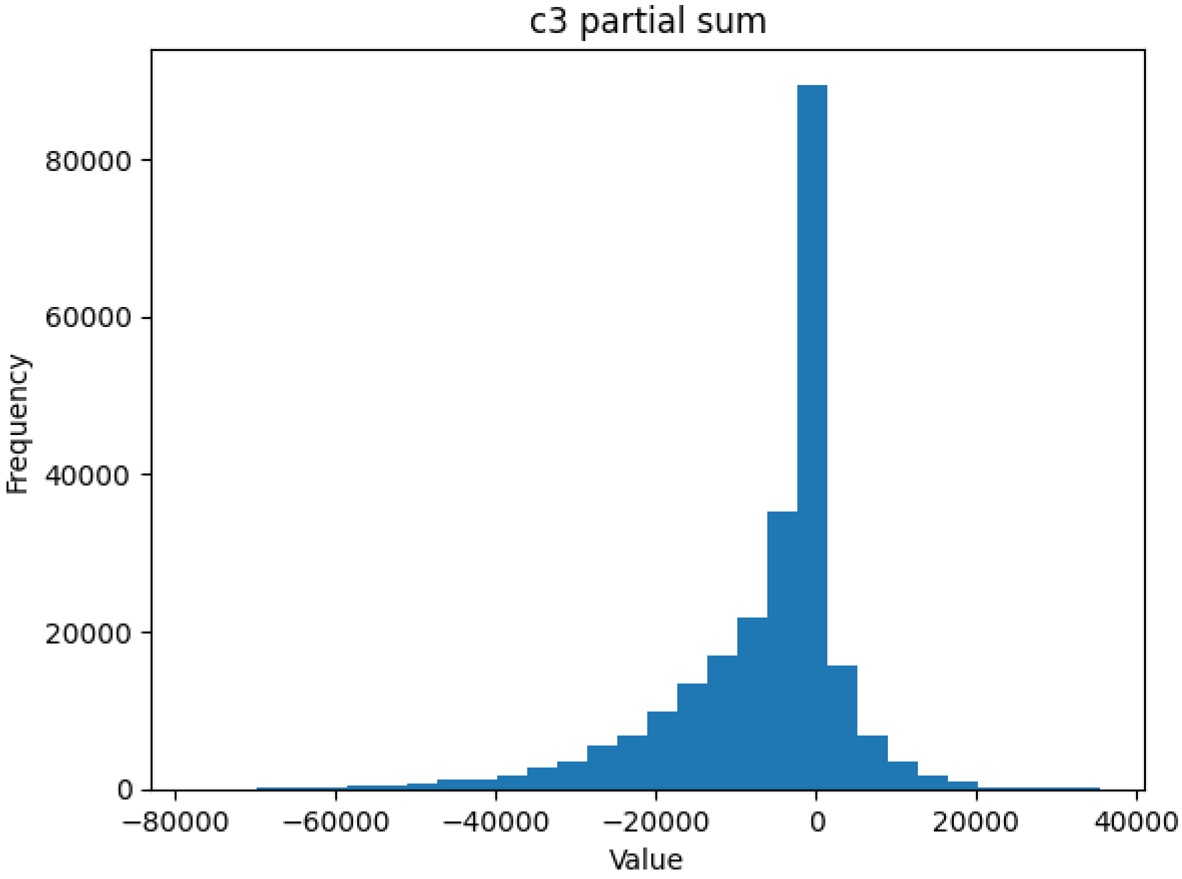
(2.1.2) 能保持原模型準確率的最小bit-width為18

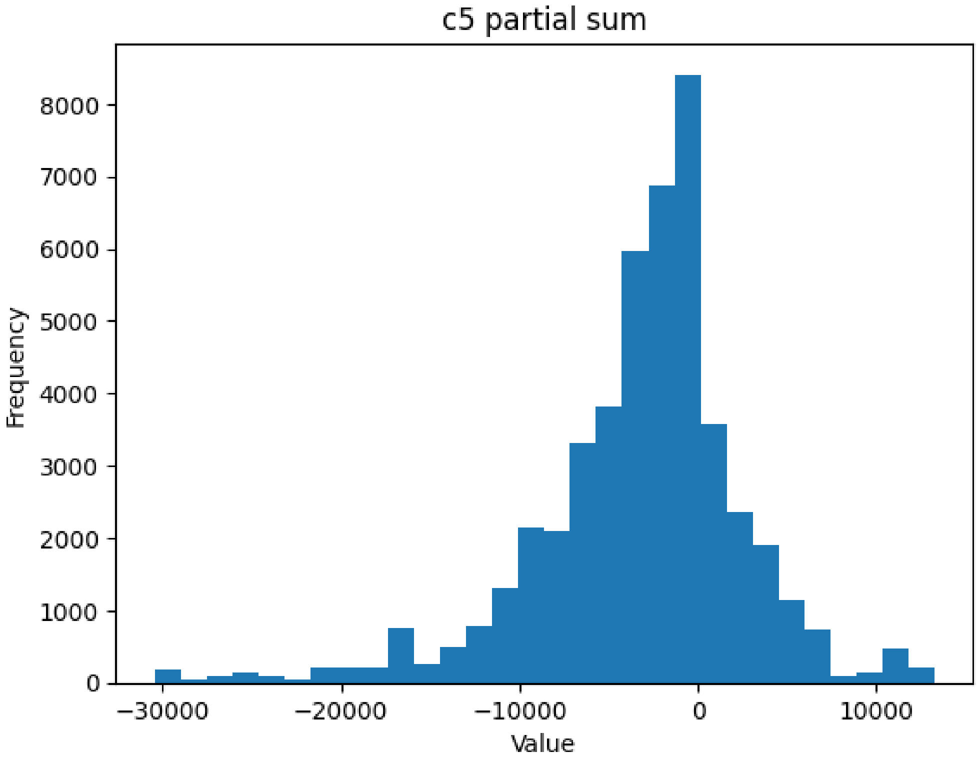


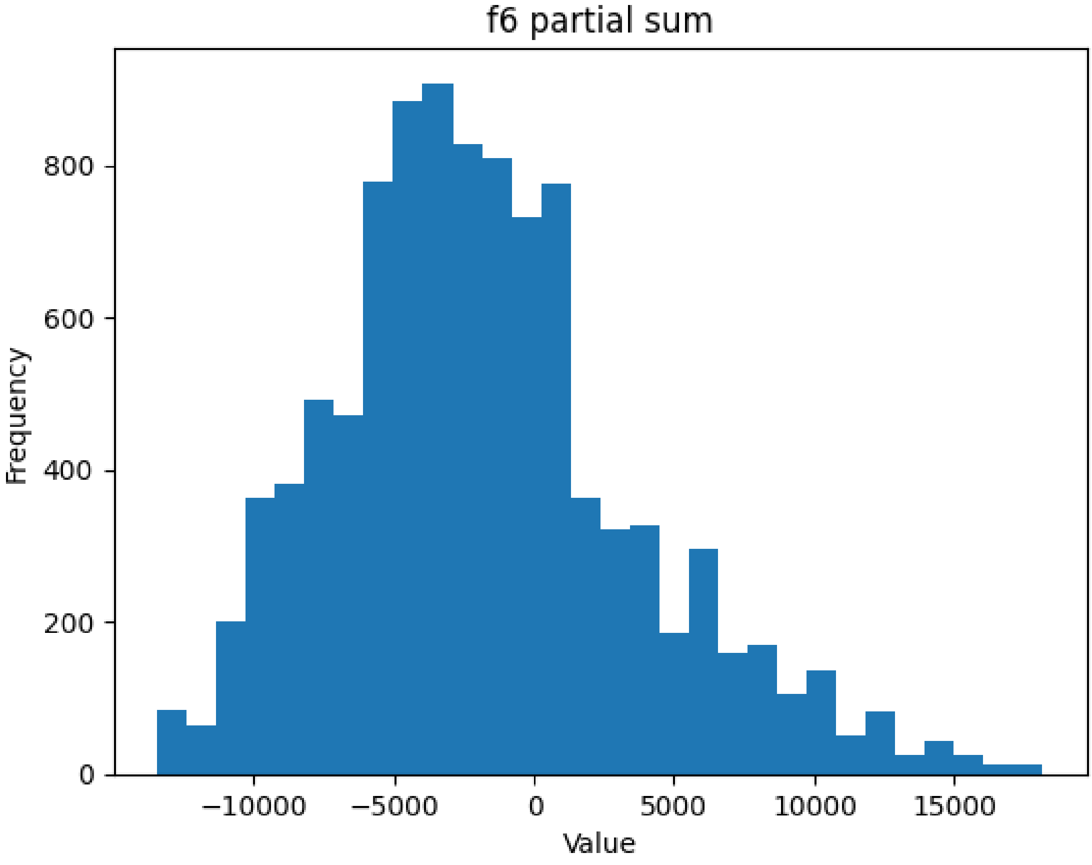
```
Accuracy: 11.35%
bit: 14
bit-width range: (-8192, 8191)
Accuracy: 29.79%
bit: 16
bit-width range: (-32768, 32767)
Accuracy: 98.15%
bit: 18
bit-width range: (-131072, 131071)
Accuracy: 98.44000000000001%
bit: 20
bit-width range: (-524288, 524287)
Accuracy: 98.44000000000001%
bit: 22
bit-width range: (-2097152, 2097151)
Accuracy: 98.44000000000001%
bit: 24
bit-width range: (-8388608, 8388607)
Accuracy: 98.44000000000001%
bit: 26
bit-width range: (-33554432, 33554431)
Accuracy: 98.44000000000001%
bit: 28
bit-width range: (-134217728, 134217727)
Accuracy: 98.44000000000001%
bit: 30
bit-width range: (-536870912, 536870911)
Accuracy: 98.44000000000001%
bit: 32
bit-width range: (-2147483648, 2147483647)
Accuracy: 98.44000000000001%
```

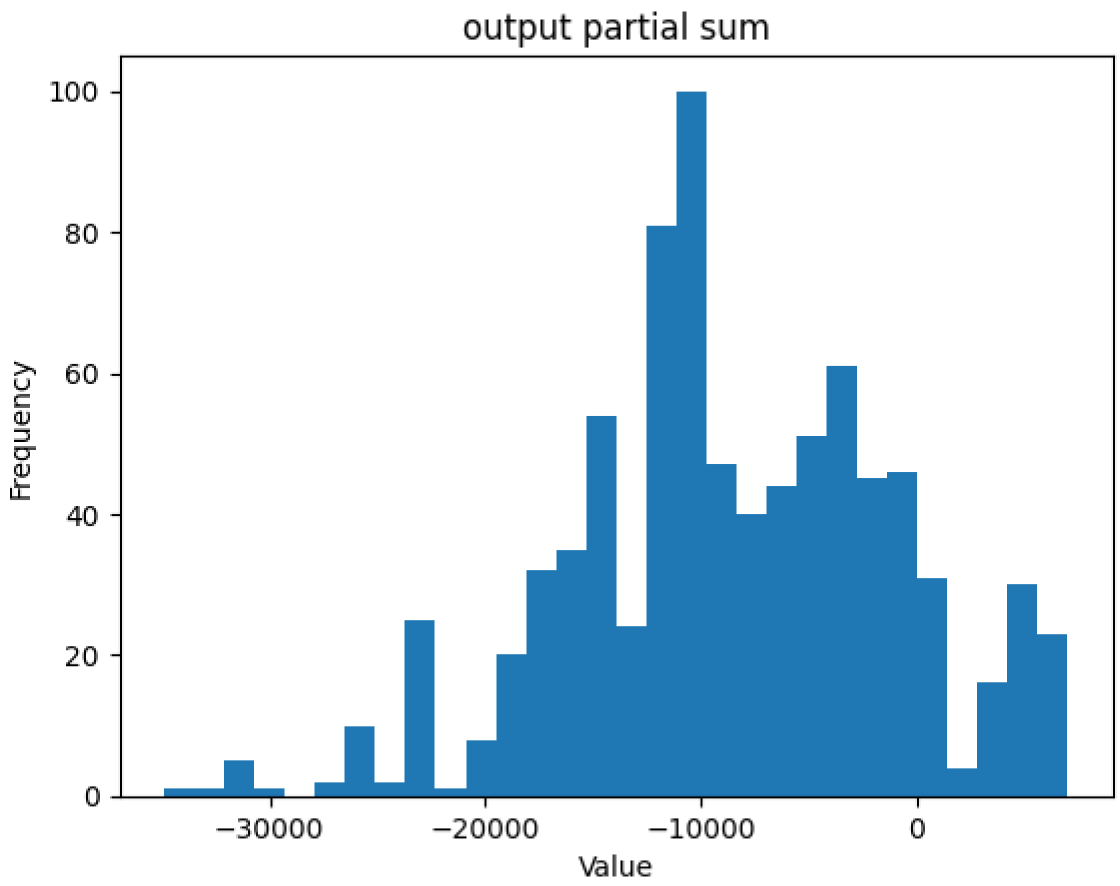
(2.2.1)


c1 partial sum
Frequency
16000
14000
12000
10000
8000
6000
4000
2000
0
−60000
−40000
−20000
0
20000
40000
60000
Value


c3 partial sum
Frequency
80000
60000
40000
20000
0
−80000
−60000
−40000
−20000
0
20000
40000
Value


c5 partial sum
Frequency
0
1000
2000
3000
4000
5000
6000
7000
8000
−30000
−20000
−10000
0
10000
Value

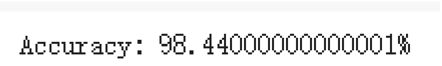

f6 partial sum
Frequency
0
200
400
600
800
−10000
−5000
0
5000
10000
15000
Value



	min	max	standard deviation
C1	-60776	63250	14443.035626
C3	-77393	35430	11192.060494
C5	-30439	13239	5837.751351
F6	-13418	18125	5422.936770
output	-35112	7029	7704.879242

(2.2.2)

	Bit-width
C1	18
C3	18
C5	16
F6	16
output	16



```
Accuracy: 98.44000000000001%
```

(3.1.1)

因為模型 quantize 成 8bit int，故乘法器固定為 8bits。Bmul = 8

Badd 則跟各層的 minimum bit-width 有關，此題各層皆為 18

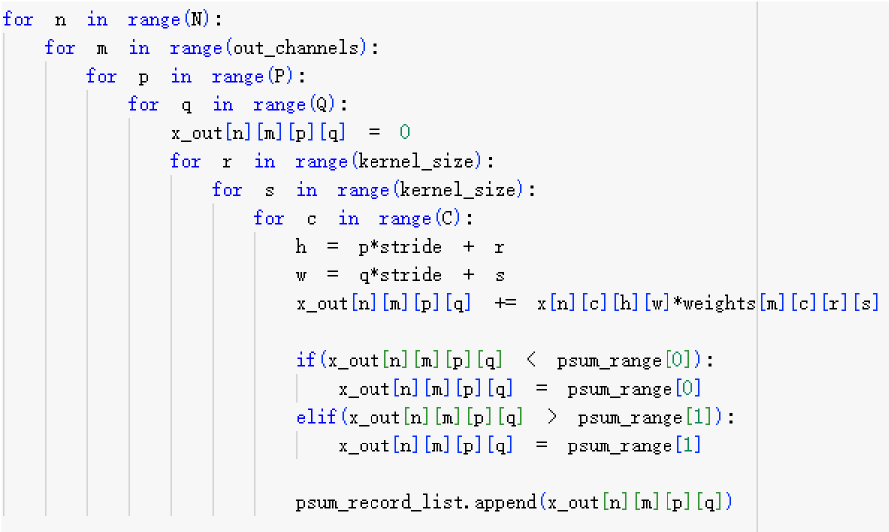
	迴圈次數
C1	470400
C3	960000
C5	192000
F6	40320
output	3360

	Nadd	Nmul	Bmul	Smul	Badd	Sadd	Ew
C1	1411200	1411200	8	64	18	18	115718400
C3	2880000	2880000	8	64	18	18	236160000
C5	576000	576000	8	64	18	18	47232000
F6	40320	40320	8	64	18	18	3306240
output	3360	3400	8	64	18	18	276240

Etotal = 402692880

Con2d 運算需 3 次 mul + 3 次 add

迴圈次數 N*M*P*Q*R*S*C

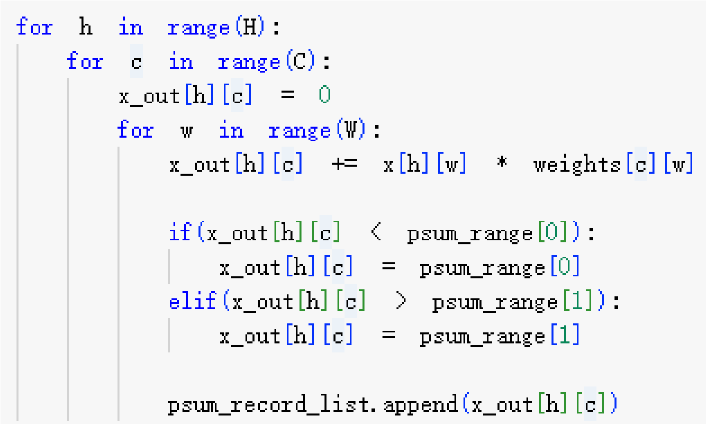


```
for n in range(N):
    for m in range(out_channels):
        for p in range(P):
            for q in range(Q):
                x_out[n][m][p][q] = 0
                for r in range(kernel_size):
                    for s in range(kernel_size):
                        for c in range(C):
                            h = p*stride + r
                            w = q*stride + s
                            x_out[n][m][p][q] += x[n][c][h][w]*weights[m][c][r][s]

                            if(x_out[n][m][p][q] < psum_range[0]):
                                x_out[n][m][p][q] = psum_range[0]
                            elif(x_out[n][m][p][q] > psum_range[1]):
                                x_out[n][m][p][q] = psum_range[1]

                            psum_record_list.append(x_out[n][m][p][q])
```

Linear 運算最內圈迴圈需 1 次 mul + 1 次 add

Output 層有 bias，每跑一次 HC 迴圈加一次 add

迴圈次數 H*C*W



```
for h in range(H):
    for c in range(C):
        x_out[h][c] = 0
        for w in range(W):
            x_out[h][c] += x[h][w] * weights[c][w]

            if(x_out[h][c] < psum_range[0]):
                x_out[h][c] = psum_range[0]
            elif(x_out[h][c] > psum_range[1]):
                x_out[h][c] = psum_range[1]

            psum_record_list.append(x_out[h][c])
```

Batch Size = 4

(3.1.2)

	迴圈次數
C1	470400
C3	960000
C5	192000
F6	40320
output	3360

	Nadd	Nmul	Bmul	Smul	Badd	Sadd	Ew
C1	1411200	1411200	8	64	18	18	115718400
C3	2880000	2880000	8	64	18	18	236160000
C5	576000	576000	8	64	16	16	46080000
F6	40320	40320	8	64	16	16	3225600
output	3360	3400	8	64	16	18	276240

Etotal = 401460240