实验报告

实验名称: 神经网络卷积层的计算 与优化

院系: 计算机学院 19 级计算机科学 与技术

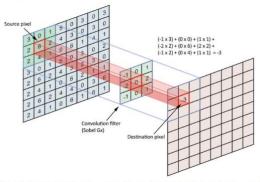
班级: 计科2班

姓名: 施天予

指导老师:陈刚

一、实验内容

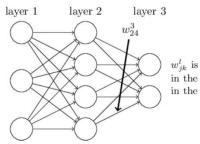
1. 使用 MIPS 汇编和 MIPS 仿真器,设计并实现一个普通整数卷积计算算子,要求有完整的输入输出,输入为 7*7*1 格式的张量,对应的卷积核一个,尺寸为 3*3*1,步长为 1,输出为经过卷积计算后的对应的 5*5*1 张量,要求计算结果正确。参考卷积操作图:



- 2. 设计并实现一个二值卷积计算算子,要求有完整的输入输出,输入为 7*7*16bit 格式的 张量,对应的卷积核一个,尺寸为 3*3*16bit,输出为经过卷积计算后的对应的 5*5*1 的 张量,要求计算结果正确。参考文献:论文 1、论文 2
- 3. 在 2 的基础上,将卷积层的偏置项、BN 层整合到同一层内。参考文献:论文 2 (3,4 二 选一)
- 4. 在 3 的基础上,对二值卷积计算算子进行寄存器复用优化,参考文献: 论文 1,论文 1 已经开源基于 ARM 的 BNN 代码,可以参照学习 (3,4 二选一)

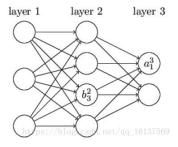
二、设计及分析

神经网络前向推理引擎



 w_{jk}^l is the weight from the $k^{\rm th}$ neuron in the $(l-1)^{\rm th}$ layer to the $j^{\rm th}$ neuron in the $l^{\rm th}$ layer

https://blog.esdn.not/no.16137569



记 w^l_{jk} 为第l-1层第k个神经元到第l层第j个神经元的权重, b^l_j 为第l层第j个神经元的偏置, a^l_j 为第l层第j个神经元的激活值(激活函数的输出)。不难看出, a^l_j 的值取决于上一层神经元的激活:

$$a_j^l = \sigma(\sum_k w_{jk}^l a_k^{l-1} + b_j^l) \tag{1}$$

将上式重写为矩阵形式:

$$a^l = \sigma(w^l a^{l-1} + b^l) \tag{2}$$

为了方便表示,记 $z^l=w^la^{l-1}+b^l$ 为每一层的权重输入, (2)式则变为 $a^l=\sigma(z^l)$ 。 利用 (2)式一层层计算网络的激活值,最终能够根据输入 X得到相应的输出 \hat{Y} 。

优化原理

二进制神经网络(BNN)是一种有前途的移动计算解决方案,因为它们的模型尺寸大大减小,算术运算复杂。通过使用按位运算来代替浮点运算,BNN可以实现显着的模型压缩,并以较小的精度损失为代价来加快推理速度。PhoneBit,这是一种 GPU 加速的 BNN 推理引擎,用于基于 Android 的移动设备,它探索了移动 GPU 上 BNN 的软件和硬件级优化机会。PhoneBit 为 BNN 的执行提供了高度优化的框架,该框架探索了高效的 BNN 运算符以在移动设备上应用高级优化。使用这些运算符优化,可以在运行时以最小的内存占用量有效地执行 BNN 的推断。在 PhoneBit 框架中,几乎所有常见的 BNN 层类型都得到了很好的支持,例如卷积,池化,批处理规范化和密集(即完全连接)层。

PhoneBit 框架利用有效的二进制神经网络运营商在手机上进行高级优化。它实现了许多操作员级别的优化,包括:本地友好的数据布局,可实现有效的内存访问;带矢量化的位打包,可在通道方向上打包位;以及层集成,它将不同层中的多个操作集成在一起,重点是二进制卷积。

二值神经网络与普通卷积神经网络在卷积层计算上的差别与好处

神经网络二值化能够最大程度地降低模型的存储占用和模型的 计算量,将神经网络中原本 32 位浮点数参数量化至 1 位定点数, 降低了模型部署的存储资源消耗,同时极大加速了神经网络的推断过 程。但二值化会不可避免地导致严重的信息损失,其量化函数不连续 性也给深度网络的优化带来了困难。

在 BNN 中,权值被二进制化,这大大减少了内存大小和访问量。 为了实现高效的卷积算子,二进制卷积运算中的点产生可以用 xor(乘 法)和 popcount (累加)代替。

$$ec{A} \cdot ec{B} = Len - 2 imes (popcount\left(xor\left(a_i, b_i
ight)
ight))$$

卷积层的输入通常是图像,这与二进制卷积层对二进制输入的要求相冲突。在位平面 Ii 和二进制权值 W 上进行二进制卷积运算,将 所有位平面的卷积相加得到输出 s。

$$s = \sum_{n=1}^8 2^{n-1} < I_i \cdot W >$$

三、实现及调试分析

1、 普通整数卷积 CNN

程序代码

```
.data
 2 arr1: .word
                1:101
 3 arr2: .word 1:101
 4 arr3: .word 1:101
   space: .asciiz " "
   line: .asciiz "\n"
 7
          .text
 8
                                    # 输入的尺寸为7*7
          addi
                 $t0,$t0,7
 9
          addi
                 $t1, $t1, 3
                                     # 卷积核的尺寸为3*3
10
          move
               $s0,$0
                                     # 初始化
11
               $s1,$0
12
          move
               $s2,$0
          move
13
                                    # 行号$s0 * 7
               $s0,$t0
   readl: mult
14
               $s2
                                     # $s2 = $s0 * 7
          mflo
15
          add
                $s2, $s2, $s1
                                    # $s2 = $s2 + $s1, 代表数组偏移量
16
                                    # 输入一个元素
          1i
                 $v0,5
17
          syscal1
18
          move $t4,$v0
                                    # 将输入的元素放入数组$t4
19
                                    # 因为一个输入4个字节, $s2*4是正确的偏移量
                 $s2, $s2, 2
20
          s11
                                    # 将$t4的值存到arr1相应位置中
                $t4, arr1($s2)
          sw
21
22
          addi
              $s1, $s1, 1
                                    # 如果列号$s1小于7,继续循环
23
          bne
                $s1, $t0, read1
                                    # 如果列号$s1等于7, 清零
               $s1,$0
24
          move
                                    # 行号$s0加1
25
          addi $s0, $s0, 1
                                    # 如果行号$s0小于7,继续循环
26
                $s0, $t0, read1
27
                                    # 初始化
               $s0.$0
28
          move
               $s1,$0
29
          move
               $s2,$0
30
          move
                $s0,$t1
                                    # 行号$s0 * 3
   read2:
          mult
31
               $s2
                                    # $s2 = $s0 * 3
          mflo
32
                                   # $s2 = $s2 + $s1, 代表数组偏移量
          add
               $s2, $s2, $s1
33
                                    # 输入一个元素
          1i
                $v0,5
34
35
          syscal1
          move $t4,$v0
                                    # 将输入的元素放入数组$t4
36
                                    # 因为一个输入4个字节, $s2*4是正确的偏移量
          s11
                $s2, $s2, 2
37
                                   # 将$t4的值存到arr2相应位置中
          sw
                $t4, arr2($s2)
38
          addi $s1, $s1, 1
39
                                   # 如果列号$s1小干3, 继续循环
               $s1, $t1, read2
          bne
40
                                    # 如果列号$s1等于3, 清零
          move
               $s1,$0
41
          addi $s0, $s0, 1
                                    # 行号$s0加1
42
          bne $s0, $t1, read2
                                   # 如果行号$s0小于3,继续循环
43
44
```

```
$t2, $t0, $t1
            sub
45
                                            # 卷积计算结果$t2 = 7 - 3 + 1 = 5
            addi
                    $t2, $t2, 1
46
47
            move
                    $s0,$0
                                            # $s0 = i (循环变量)
48
                                            # $s1 = j (循环变量)
     for1:
            move
                    $s1,$0
49
                    $s2,$0
                                            # $s2 = k (循环变量)
    for2:
50
            move
                    $t6,$0
                                            # $t6清零,用于每次arr3的赋值
51
            move
                    $s3,$0
                                            # $s3 = 1 (循环变量)
    for3:
            move
52
                    $t3, $s0, $t2
53
    for4:
            mu1
54
            add
                    $t3, $t3, $s1
            s11
                    $t3,$t3,2
                                            # $t3 = i*5+j arr3的偏移量
55
            add
                    $t4, $s0, $s2
56
            mu1
                    $t4, $t4, $t0
57
            add
                    $t4, $t4, $s1
58
            add
                    $t4, $t4, $s3
59
                                            # $t4 = (i+k)*7+j+1 arr1的偏移量
            s11
                    $t4, $t4, 2
60
                    $t5, $s2, $t1
61
            mu1
            add
                    $t5, $t5, $s3
62
                                           # $t5 = k*3+1 arr2的偏移量
            s11
                    $t5, $t5, 2
63
                    $t7, arr1($t4)
                                           # $t7为arr1对应值
64
            1w
            1w
                    $t8, arr2($t5)
                                           # $t8为arr2对应值
65
                    $t9, $t7, $t8
                                           # $t9 = $t7 * $t8 临时变量
            mu1
66
                    $t6.$t6.$t9
                                           # $t6 = $t6 + $t9 arr3对应值
            add
67
                                           # 将$t6的值存入arr3对应位置
68
            sw
                    $t6, arr3($t3)
            addi
                    $s3, $s3, 1
                                           # 1 = 1 + 1
69
                    $s3, $t1, for4
                                           # 如果1 == 3, 跳出循环
            bne
70
            addi
                    $s2, $s2, 1
                                           \# k = k + 1
71
            bne
                    $s2, $t1, for3
                                           # 如果k == 3, 跳出循环
72
73
            addi
                    $s1, $s1, 1
                                           \# j = j + 1
            bne
                    $s1, $t2, for2
                                           # 如果j == 5, 跳出循环
74
                                           \# i = i + 1
            addi
                    $s0, $s0, 1
75
                    $s0, $t2, for 1
                                           # 如果i == 5, 跳出循环
            bne
76
77
            move
                    $s0,$0
                                           # $s0 = i (循环变量)
78
                    $s1,$0
                                           # $s1 = j (循环变量)
    printl: move
79
    print2: mul
                    $t3, $s0, $t2
80
            add
                    $t3, $t3, $s1
81
            s11
                    $t3, $t3, 2
                                           # $t3 = i*5+j arr3的偏移量
82
            1w
                    $a0, arr3($t3)
83
            1i
                    $v0.1
84
                                            # 输出arr3对应值
            syscal1
85
            la
                    $a0, space
86
            1i
                    $v0,4
87
                                           # 输出空格
            syscal1
88
            addi
                    $s1, $s1, 1
                                           \# j = j + 1
89
                                           # 如果j == 5, 跳出循环
90
            bne
                    $s1, $t2, print2
            la
                    $a0, line
91
            1i
                    $v0, 4
92
            syscal1
                                           # 输出回车
93
             addi
                    $s0, $s0, 1
                                            \# i = i + 1
94
                    $s0, $t2, print1
                                            # 如果i == 5, 跳出循环
             bne
95
     end:
96
                    $v0, 10
             1i
97
                                            # 结束程序
             svscal1
98
```

调试及结果

在程序中一个一个地输入以上 7*7 和 3*3 的张量,得到如下结果

与写好的 C++程序比较发现结果正确

```
3 0 1 5 0 3 0

2 6 2 4 3 0 3

2 4 1 0 6 1 4

3 0 1 5 0 3 0

2 6 2 4 3 2 3

2 4 1 0 6 2 1

2 6 2 4 4 0 3

-1 0 1

-2 0 2

-1 0 1

-3 -3 6 -9 -2

-4 -5 10 -4 -4

-5 4 4 -5 -2

-3 -3 6 -4 -5

-2 -12 13 -2 -11
```

2、 二值卷积 BNN

方法分析

将 3*3*16bit 中的 3*3 位打包,通过异或操作和公式得到结果

$$ec{A} \cdot ec{B} = Len - 2 imes (popcount \left(xor\left(a_i, b_i
ight)
ight))$$

程序代码

```
1
            .data
  2 arr1:
           .word
                    1:3136
    arr2:
                    1:576
           .word
    arr3:
           .word 1:100
     space: .asciiz " "
            .asciiz "\n"
     line:
  6
            .text
  7
  8
            addi
                    $t0, $0, 7
                                           # 输入的尺寸为7*7
  9
                    $t1,$0,3
                                           # 卷积核的尺寸为3*3
            addi
10
                                           # 通道数位为16
                    $s7, $0, 16
            addi
11
12
                                           # 初始化 (循环变量)
                    $s0,$0
            move
13
                    $s2, $t0, $t0
            mu1
14
            mu1
                    $s2, $s2, $s7
                                           # 输入大小为7*7*16
15
16
    read1:
           s11
                    $t3, $s0, 2
                    $v0,5
                                           # 输入一个元素
17
            syscall
18
                   $v0, $v0, 1
            addi
19
                   $v0, $v0, 1
                                          \# A = (a+1)/2
20
                                          # 存入数组对应位置中
                   $v0, arr1($t3)
            sw
21
            addi
                   $s0, $s0, 1
22
            bne
                   $s0, $s2, read1
                                          # 循环次数未到,继续循环
23
24
            move
                   $s0,$0
                                          # 初始化(循环变量)
25
                   $s2, $t1, $t1
26
            mu1
                   $s2, $s2, $s7
                                          # 卷积核大小为3*3*16
27
    read2:
                   $t3, $s0, 2
           s11
28
                   $v0,5
                                          # 输入一个元素
            1i
29
            syscal1
30
            addi
                   $v0, $v0, 1
31
                                          \# B=(b+1)/2
            sr1
                   $v0, $v0, 1
32
                                          # 存入数组对应位置中
                   $v0, arr2($t3)
33
            addi
                   $s0, $s0, 1
34
```

```
bne
                $s0, $s2, read2
                                    # 循环次数未到, 继续循环
35
36
                 $t2, $t0, $t1
           sub
37
                                    # 卷积结果的尺寸$t2 = 7 - 3 + 1 = 5
           addi $t2, $t2, 1
38
39
           move
                $s0,$0
                                     # $s0 = i (循环变量)
40
    for1:
                $s1,$0
                                     # $s1 = j (循环变量)
          move
41
    for2:
                $s5,$0
                                     # $s5用于保存count函数后得到的1的数量
42
          move
                                     # $s2 = k (循环变量)
                $s2,$0
43
           move
                $t6,$0
                                     # $t6用于得出输入张量1与0连起来后的结果
    for3:
           move
44
                 $t7,$0
                                     # $t7用于得出卷积核1与0连起来后的结果
           move
45
                                     # $s3 = 1 (循环变量)
           move
                 $s3,$0
46
                 $s4,$0
    for4:
           move
                                     \# \$s4 = r
47
    for5:
           mul
                  $s6,$t0,$t0
48
           mu1
                  $s6, $s6, $s2
49
           add
                  $t4, $s0, $s3
50
           mul
                 $t4, $t4, $t0
51
           add
                 $t4, $t4, $s1
52
           add
53
                 $t4, $t4, $s4
           add
                 $t4, $t4, $s6
54
                 $t4, $t4, 2
                                      # $t4 = (i+1)*7+j+r+k*7*7 arr1的偏移量
           s11
55
56
                 $s6, $t1, $t1
57
           mu1
           mul
                  $s6, $s6, $s2
58
           mu1
                 $t5, $s3, $t1
59
           add
                 $t5, $t5, $s4
60
           add $t5, $t5, $s6
61
           s11
                 $t5, $t5, 2
                                      # $t5 = 1*3+r+k*3*3 arr2的偏移量
62
63
           1 w
                 $t8, arr1($t4)
                                      # $t7为arr1对应值
64
                  $t9, arr2($t5)
                                      # $t8为arr2对应值
           1w
65
66
           s11
                 $t6, $t6, 1
67
           add
                 $t6, $t6, $t8
                                      # 合并输入张量3*3*1中的1和0
68
           s11
                 $t7, $t7, 1
69
                                      # 合并卷积核3*3*1中的1和0
                $t7, $t7, $t9
70
           add
71
           addi
                $s4, $s4, 1
                                      \# r = r + 1
72
                                      # 如果r == 3, 跳出循环
           bne
                  $s4, $t1, for5
73
           addi
                $s3, $s3, 1
                                      # 1 = 1 + 1
74
           bne
                  $s3, $t1, for4
                                      # 如果1 == 3, 跳出循环
75
                 $a0,$t6,$t7
                                       # $t6与$t7异或
           xor
76
                                       # 调用count函数, 计算1的个数
           jal
                  count
77
           add
                  $s5, $s5, $v0
                                      # 将结果加到$s5中
78
           addi $s2, $s2, 1
                                      \# k = k + 1
79
                  $s2, $s7, for3
                                      # 如果k == 16, 跳出循环
           bne
80
81
                  $t3,$s0,$t2
           mu1
82
           add
                  $t3, $t3, $s1
83
           s11
                  $t3,$t3,2
                                       # $t3 = i*5+i arr3的偏移量
84
85
```

```
$s6, $t1, $t1
 86
                      $s6, $s6, $s7
 87
              mu1
              s11
                      $s5, $s5, 1
 88
                     $s5, $s5, $s6
              sub
 89
                                             # 公式result=N-2*BCNT(XNOR(A*B))
                     $s5, $0, $s5
 90
                                             # 将$s5的值存入arr3对应位置
                      $s5, arr3($t3)
 91
 92
              addi
                     $s1, $s1, 1
                                            \# j = j + 1
 93
                     $s1, $t2, for2
                                            # 如果j == 5, 跳出循环
              bne
 94
                                             \# i = i + 1
              addi
                     $s0, $s0, 1
 95
                                             # 如果i == 5, 跳出循环
              bne
                     $s0, $t2, for 1
 96
 97
                     $s0,$0
                                             # $s0 = i (循环变量)
 98
              move
                                             # $s1 = j (循环变量)
                     $s1,$0
      printl: move
      print2: mul
                     $t3, $s0, $t2
100
                     $t3, $t3, $s1
              add
101
102
              s11
                     $t3, $t3, 2
                                            # $t3 = i*5+j arr3的偏移量
                    $a0, arr3($t3)
103
             1 w
                    $v0,1
104
                                          # 输出arr3对应值
             syscal1
105
             1a
                    $a0, space
106
             1i
                    $v0,4
107
                                          # 输出空格
             syscal1
108
             addi
                    $s1, $s1, 1
                                          \# j = j + 1
109
                    $s1, $t2, print2
                                          # 如果j == 5, 跳出循环
110
             la
                    $a0,1ine
111
             1i
                    $v0, 4
112
                                          # 输出回车
113
             syscal1
                                          \# i = i + 1
                    $s0, $s0, 1
114
             addi
                                          # 如果i == 5, 跳出循环
115
                    $s0, $t2, print1
116
             1i
                    $v0, 10
117
             syscal1
                                          # 结束程序
118
119
                                             # count函数,用于计算一个二进制数中1的个数
      count:
120
121
              move
                      $t4,$0
              addi
                      $t5, $0, 0x00000001
                                             # $t5用于判断二进制数末尾是否是1
122
              move
                      $v0,$0
123
                                             # 循环次数
              addi
                      $t8, $0, 9
124
                      $t9,$0
                                             # 循环变量
125
              move
                      $t4, $a0, $t5
                                             # $t4用于保存按位与结果,得到1说明末尾位是1
             and
      100p:
126
                                             # 计数
              add
                      $v0, $v0, $t4
127
                      $a0, $a0, 1
                                             # 将二进制数右移一位
              sr1
128
              addi
                      $t9, $t9, 1
129
                      $t9, $t8, loop
                                             # 未到循环次数,继续循环
              bne
130
                                             # 函数返回
              jr
                      $ra
131
132
```

调试及结果

为了输入方便, 我设置每一通道 7*7 的输入和 3*3 的卷积核都相同

(真实测试将通道数设为1)即假设16个通道都是以下输入

```
-111111-1

1111111

1111-111

1111111

1111111

111-1111

111

-1-1-1
```

调用 mips 程序得到如下结果

```
16 48 48 48 16

48 16 16 16 48

48 80 80 80 48

48 16 16 16 48

48 16 16 16 48

— program is finished running —
```

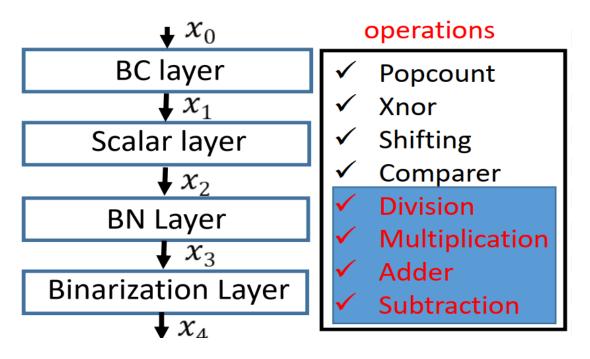
发现与 C++测试结果相同,程序正确!

```
sty@sty-virtual-machine:~$ g++ bnn.cpp
sty@sty-virtual-machine:~$ ./a.out
-1 1 1 1 1 1 -1
1111111
1111111
1 1 1 -1 1 1 1
1 1 1 1 1 1 1
1111111
111-1111
1 1 1
-1 -1 -1
1 1 1
16 48 48 48 16
48 16 16 16 48
48 80 80 80 48
48 16 16 16 48
48 16 16 16 48
```

3、 层整合

方法分析

根据实验 2,我们得到了 BC 层,通过一个方法可以不用这么多层的计算,直接得出最后我们要的二值化层。



$$x_2 = x_1 + b$$

$$x_3 = \gamma \cdot \frac{x_2 - \mu}{\sigma} + \beta$$

$$x_4 = \begin{cases} 1 & \text{if } x_3 \ge 0 \\ 0 & \text{otherwise} \end{cases}$$

层整合公式

$$\xi = \mu - \frac{\beta \cdot \sigma}{\gamma} - b$$

$$x_4 = egin{cases} 1 & ext{if } x_1 \geq \xi ext{ and } \gamma > 0 \ 0 & ext{if } x_1 < \xi ext{ and } \gamma > 0 \ 1 & ext{if } x_1 \leq \xi ext{ and } \gamma < 0 \ 0 & ext{if } x_1 > \xi ext{ and } \gamma < 0 \end{cases}$$

程序代码

```
.data
  2 arrl: .word 1:3136
  3 arr2:
           .word 1:576
  4 arr3: .word 1:100
  5 space: .asciiz " "
  6 line: .asciiz "\n"
           .text
  7
  8
           addi $t0, $0, 7
                                     # 输入的尺寸为7*7
  9
                                      # 卷积核的尺寸为3*3
                $t1,$0,3
 10
           addi
           addi
                  $s7, $0, 16
                                      # 通道数位为16
 11
 12
                  $s0,$0
                                      # 初始化(循环变量)
 13
           move
                  $s2,$t0,$t0
           mu1
14
           mu1
                  $s2, $s2, $s7
                                      # 输入大小为7*7*16
 15
    readl: s11
                  $t3, $s0, 2
 16
           1i
                  $v0.5
                                      # 输入一个元素
 17
           syscall
18
           addi
                  $v0, $v0, 1
19
                  $v0, $v0, 1
                                       \# A = (a+1)/2
20
           srl
                  $v0, arr1($t3)
                                      # 存入数组对应位置中
21
           addi $s0,$s0,1
22
                                       # 循环次数未到,继续循环
                  $s0, $s2, read1
23
           bne
24
                $s0,$0
                                       # 初始化(循环变量)
           move
25
           mu1
                 $s2, $t1, $t1
26
                 $s2, $s2, $s7
                                      # 卷积核大小为3*3*16
27
           mu1
    read2: s11
                 $t3, $s0, 2
28
           1i
                  $v0,5
                                       # 输入一个元素
29
           syscal1
30
          addi $v0, $v0, 1
31
          srl
                 $v0, $v0, 1
                                       \# B=(b+1)/2
32
          sw
                 $v0, arr2($t3)
                                      # 存入数组对应位置中
33
           addi $s0, $s0, 1
34
                                    # 循环次数未到, 继续循环
          bne
                $s0, $s2, read2
36
          sub
                $t2, $t0, $t1
37
                $t2, $t2, 1
                                    # 卷积结果的尺寸$t2 = 7 - 3 + 1 = 5
          addi
38
39
                $s0,$0
                                    # $s0 = i (循环变量)
40
          move
                                    # $s1 = j (循环变量)
                 $s1,$0
   for1:
          move
41
                 $s5,$0
                                    #$s5用于保存count函数后得到的1的数量
   for2:
          move
42
          move
                 $s2,$0
                                    # $s2 = k (循环变量)
43
   for3:
                 $t6,$0
                                    # $t6用于得出输入张量1与0连起来后的结果
          move
44
                                    # $t7用于得出卷积核1与0连起来后的结果
          move
                $t7,$0
45
                $s3,$0
                                    # $s3 = 1 (循环变量)
          move
46
                                    # \$s4 = r
47 for4:
                $s4, $0
         move
   for5:
          mu1
                $s6, $t0, $t0
48
          mu1
                $s6, $s6, $s2
49
          add
                $t4, $s0, $s3
50
          mul
                $t4, $t4, $t0
51
```

```
$t4.$t4.$s1
              add
 52
              add
                      $t4, $t4, $s4
 53
                      $t4, $t4, $s6
              add
 54
              s11
                      $t4, $t4, 2
                                               # $t4 = (i+1)*7+j+r+k*7*7  arr 1 的偏移量
 55
 56
                      $s6, $t1, $t1
              mu1
 57
                      $s6, $s6, $s2
              mu1
 58
              mu1
                      $t5, $s3, $t1
 59
              add
                      $t5, $t5, $s4
 60
              add
                      $t5, $t5, $s6
 61
              s11
                      $t5, $t5, 2
                                               # $t5 = 1*3+r+k*3*3 arr2的偏移量
 62
 63
                      $t8, arr1($t4)
                                               # $t7为arr1对应值
              1w
 64
                      $t9, arr2($t5)
                                               # $t8为arr2对应值
              1w
 65
 66
              s11
                      $t6, $t6, 1
 67
                                               # 合并输入张量3*3*1中的1和0
                      $t6, $t6, $t8
 68
              add
                       $t7, $t7, 1
              s11
 69
              add
                      $t7, $t7, $t9
                                               # 合并卷积核3*3*1中的1和0
 70
 71
                                               \# r = r + 1
              addi
                      $s4, $s4, 1
 72
              bne
                       $s4, $t1, for 5
                                               # 如果r == 3, 跳出循环
 73
                                               #1 = 1 + 1
              addi
                      $s3, $s3, 1
 74
              bne
                       $s3, $t1, for4
                                               # 如果1 == 3, 跳出循环
 75
                      $a0, $t6, $t7
                                               # $t6与$t7异或
              xor
 76
                                               # 调用popcount函数, 计算1的个数
              jal
                      popcount
 77
                                               # 将结果加到$s5中
              add
                      $s5, $s5, $v0
 78
                                               \# k = k + 1
              addi
                       $s2, $s2, 1
 79
                       $s2, $s7, for3
                                               # 如果k == 16, 跳出循环
              bne
 80
 81
                       $t3, $s0, $t2
              mu1
 82
              add
                      $t3, $t3, $s1
 83
              s11
                       $t3, $t3, 2
                                               # $t3 = i*5+j arr3的偏移量
 84
 85
                       $s6, $t1, $t1
               mu1
 86
                       $s6, $s6, $s7
               mu1
 87
               s11
                       $s5, $s5, 1
 88
               sub
                       $s5, $s5, $s6
 89
                       $s5, $0, $s5
                                                 # 公式result=N-2*BCNT(XNOR(A*B))
 90
               sub
                        $s5, arr3($t3)
                                                 # 将$s5的值存入arr3对应位置
               sw
 91
 92
                       $s1, $s1, 1
                                                 \# j = j + 1
               addi
 93
                                                 # 如果j == 5, 跳出循环
               bne
                       $s1, $t2, for 2
 94
                                                 \# i = i + 1
                       $s0, $s0, 1
 95
               addi
               bne
                       $s0, $t2, for 1
                                                 # 如果i == 5, 跳出循环
 96
 97
 98
                       $s2, $0, -48
                                                 # 参数b = -48
               addi
 99
                       $s3, $0, 1
                                                 # 参数\gamma = 1
               addi
100
               addi
                       $s4, $0, 1
                                                 # 参数 μ = 1
101
                       $s5, $0, 1
                                                 # 参数 σ = 1
               addi
102
```

```
103
             addi
                    $s6, $0, 1
                                           # 参数β = 1
104
             mu1
                    $s6, $s6, $s5
105
             div
                     $s6, $s6, $s3
106
                     $s7, $s4, $s6
              sub
107
              sub
                    $s7, $s7, $s2
                                           #\xi = \mu - (\beta * \sigma) / \gamma - b
108
109
                    $s0,$0
                                           # $s0 = i (循环变量)
110
             move
                                           # $s1 = j (循环变量)
                     $s1.$0
111
      print1: move
                     $t3, $s0, $t2
      print2: mul
112
             add
                     $t3, $t3, $s1
113
             s11
                     $t3, $t3, 2
                                           # $t3 = i*5+j arr3的偏移量
114
                     $a0, arr3($t3)
115
116
                     $s2, $0, 1
                                           # 以下注释是层整合的公式
             addi
117
                     $a0, $s7, assign
                                           # x4=1 if x1 \ge \xi and \gamma > 0
              beq
118
                                           # x4=0 if x1<\xi and \gamma>0
                     $s4, $a0, $s7
119
              sgt
                     $s5, $s3, $0
                                             # x4=1 if x1 \le \xi and \gamma < 0
              sgt
120
                     $s2, $s4, $s5
                                             # x4=0 if x1>\xi and \gamma<0
121
              seq
122
      assign: move
                     $a0,$s2
              1i
                      $v0,1
123
              syscal1
                                              # 输出arr3对应值
124
                      $a0, space
125
              la
                      $v0, 4
126
              syscal1
                                              # 输出空格
127
                                              \# j = j + 1
              addi $s1, $s1, 1
128
                                              # 如果j == 5, 跳出循环
                     $s1, $t2, print2
129
130
              la
                      $a0, line
              1i
                      $v0, 4
131
                                              # 输出回车
              syscal1
132
              addi
                      $s0, $s0, 1
                                              \# i = i + 1
133
                                              # 如果i == 5, 跳出循环
                      $s0, $t2, print1
134
              bne
135
                    $v0, 10
136
                                              # 结束程序
              svscal1
137
138
                                              # popcount函数,用于计算一个二进制数中1的个数
139
      popcount:
                      $t4.$0
              move
140
                                              # $t5用于判断二进制数末尾是否是1
              addi
                      $t5, $0, 0x00000001
141
142
                      $v0,$0
              addi
                     $t8, $0, 9
                                              # 循环次数
143
                    $t9,$0
                                              # 循环变量
              move
144
              and
                    $t4,$a0,$t5
                                              # $t4用于保存按位与结果,得到1说明末尾位是1
145
      100p:
                   $v0,$v0,$t4
                                              # 计数
              add
146
                                              # 将二进制数右移一位
              srl
                     $a0, $a0, 1
147
              addi
                      $t9, $t9, 1
148
                     $t9, $t8, loop
                                              # 未到循环次数,继续循环
              bne
149
                                              # 函数返回
150
              jr
                      $ra
151
```

调试及结果

设置 b = -48, γ = 1, μ = 1, σ = 1, β = 1

为了输入方便,我设置每一通道 7*7 的输入和 3*3 的卷积核都相同 (真实测试将通道数设为 1)即假设 16 个通道都是以下输入

程序结果如下

```
0 1 1 1 0
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
- program is finished running —
```

经检验发现结果正确!

四、实验总结

刚开始神经网络、卷积这些陌生的知识让我有些恐惧。但通过反复仔细阅读论文,还有广泛搜集相关资料,我逐步找到了解决的办法。第一个实验 CNN 比较简单,其实就是一个4重循环的计算。第二个实验 BNN 稍微复杂了一些,但通过位打包、异或、popcount等操作最终我也解决了问题。第三个实验层整合看起来很复杂,但读懂了之后发现其实也很简单,只要用参数代入公式计算一下,就能得到正确的结果。通过这次大作业我学到了很多知识,但最重要的是让我提升了自己解决问题的能力。过程虽然痛苦,但成功的喜悦也让人振奋!