

AI on Chip 2024

LAB V AI Compiler REPORT

Student name: __胡家豪__

Student ID: __N26122246__

目錄

| | |
|------------------------------|---|
| 一.Implementation Result..... | 3 |
| 二.Tiling Strategy | 3 |
| 三.DRAM assignment | 3 |
| 四.Lesson Learned | 4 |

一.Implementation Result

```
[[('n02123159', 'tiger_cat', 0.7677315), ('n02123045', 'tabby', 0.16499023), ('n02124075', 'Egyptian_cat', 0.062334243), ('n02123394', 'Persian_cat', 0.0010981582), ('n04589890', 'window_screen', 0.0007930122)]]
aoc2024101@user:~/Lab5/AOC1ab5$
```

二.Tiling Strategy

```
int j = 0; j < ifmap_H + 1; j += tiled_stride_h;

int tiled_w = std::min(tiled_size, ifmap_W - i + 2);
```

對於 tile 的移動，我將位置指標每次移動 tiled_stride 個，直到遇到 feature map 的邊界，但是由於在邊界通常還需要多做一次，但是這個多做的一次通常與原本 tile 的大小不同，如果是要多做一次的，這個多的 feature map 需要 tile 的大小會比原本的 size 小，因此使用 min 進行判斷。

```
int padding_t = (j == 0) ? 1 : 0;
int padding_r = (i + tiled_w >= ifmap_W + 2) ? 1 : 0;
```

對於 padding 是否需要則看目前是不是在最邊邊，這個可以用 i 與 j 進行判斷，而上述的行為高跟寬是一樣的。

三.DRAM assignment

在記憶體地址的 assign 部分，需要指派的有 ifmap 的地址與 ofmap 的地址。

在 ifmap 需要注意的是初始的地方還有第一行以及第一列。首先先定義一開始放置的位置，接下來則注意第一行還有第一列的擺放，後面的放置就很簡單了。

```
//ifmap dram_addr you define!!!
if ((j == 0) && (i == 0)){
    dram_addr = DRAM_INPUT_START_ADDR + ifmap_W * ifmap_H * k;
} else if (j == 0) {
    dram_addr = DRAM_INPUT_START_ADDR + ifmap_W * ifmap_H * k + (i - 1);
} else if (i == 0) {
    dram_addr = DRAM_INPUT_START_ADDR + ifmap_W * ifmap_H * k + (j - 1) * ifmap_W;
} else {
    dram_addr = DRAM_INPUT_START_ADDR + ifmap_W * ifmap_H * k + (j - 1) * ifmap_W + (i - 1);
}
```

最後是 ofmap 的擺放，由於是輸出，就隨著地址增加擺放即可

```
// ofmap dram_addr you define!!!
dram_addr = DRAM_OUTPUT_START_ADDR + ifmap_W * ifmap_H * out_c + j * ifmap_W + i;
```

四. Lesson Learned

本次作業我學到了如何切 tile 以及如何分配記憶體的地址。我覺得最難的部分在於這個沒有辦法 debug，只能事先想好然後一次一次的不斷嘗試。

如果除去這個難點，另一個困難的部分是了解搬運資料如何進行，單看這次 project 的資料很大，很難了解。我們只能片面的去理解助教的投影片，所以費了不少時間。不過經過這次作業，我大概理解如何進行 DRAM 與 SRAM 的溝通，以及需要注意的部分，或許在做 final 的時候可以進行參考。