基於二值化卷積神經網路權重共享軟硬體整合設計

指導教授: 林英超
專題組員: 薛文威、吳仲軒、孫宇亭、陳品翰

目錄

簡介

基於現今社會對於影像辨識等運算需求大幅提升，且現今可攜式裝置的使用量增加，而傳統卷積神經網路的計算會造成大量運算且功耗較高，較不適合嵌入其中，所以我們參考優化深度學習電路的相關論文後，採用卷積核分解(Kernel Decomposition) 的方法，來實現 MNIST 手寫數字辨識，並額外提出兩種方法。

1. Image Compute Fusion (ICF) : 在減少面積的情況下，同時降低功耗。

2. Weight Lookahead with Zero Skipping (WL/ZS) : 更進一步降低面積、功耗，完成硬體架構的優化。

實驗方法

軟體端 (使用工具：Keras、Python)

以 Keras 自製 BNN 模型 → 訓練並取得測試資料 → 將資料轉為十六進制 → 設計軟體模擬運算次數

硬體端 (使用工具：Ncverilog、Simvision)

實作 conv 電路 → 實作 ICF 及 WL/ZS 改進效能 → 設計 testbench 並進行 RTL 模擬 → 合成電路

卷積神經網路介紹

Convolution Neural Network (簡稱 CNN)，又稱為卷積神經網路，在影像辨識方面的功能很強大，現今有許多的模型是依照 CNN 的來去做延伸的。

其計算過程為:
1. 擷取和 Kernel 相同大小的原始圖片後，將其和 Kernel 之間相對應的位置做乘法運算。
2. 移動所擷取的區域，並重複運算，最後將所有結果加總，即為該位置卷積運算後的結果。

8	5	13		
4	7	18		
8	0	7		

Image

⊗

6	4	2
5	0	-6
1	9	10

Kernel

=

84		

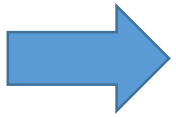
計算方式: 8*6 + 5*4 + 13*2 + 4*5 + 7*0 + 18*-6 + 8*1 + 0*9 + 7*10 = 84

Binary Neural Network 介紹

傳統卷積運算中，kernel 的權重會有任意的數值，而二值化顧名思義，是將權重調整為 1 或是 -1。方法是將權重 (8 bits) 除以 255 後，根據其權重之正負，來判定二值化後權重為 1 或 -1 ，以省略繁雜的運算，為 Binary Neural Network (簡稱BNN)。

-0.7	0.9	0.92
0.87	-0.97	-0.34
1.0	0.67	-0.46

調整後之卷積運算 Kernel

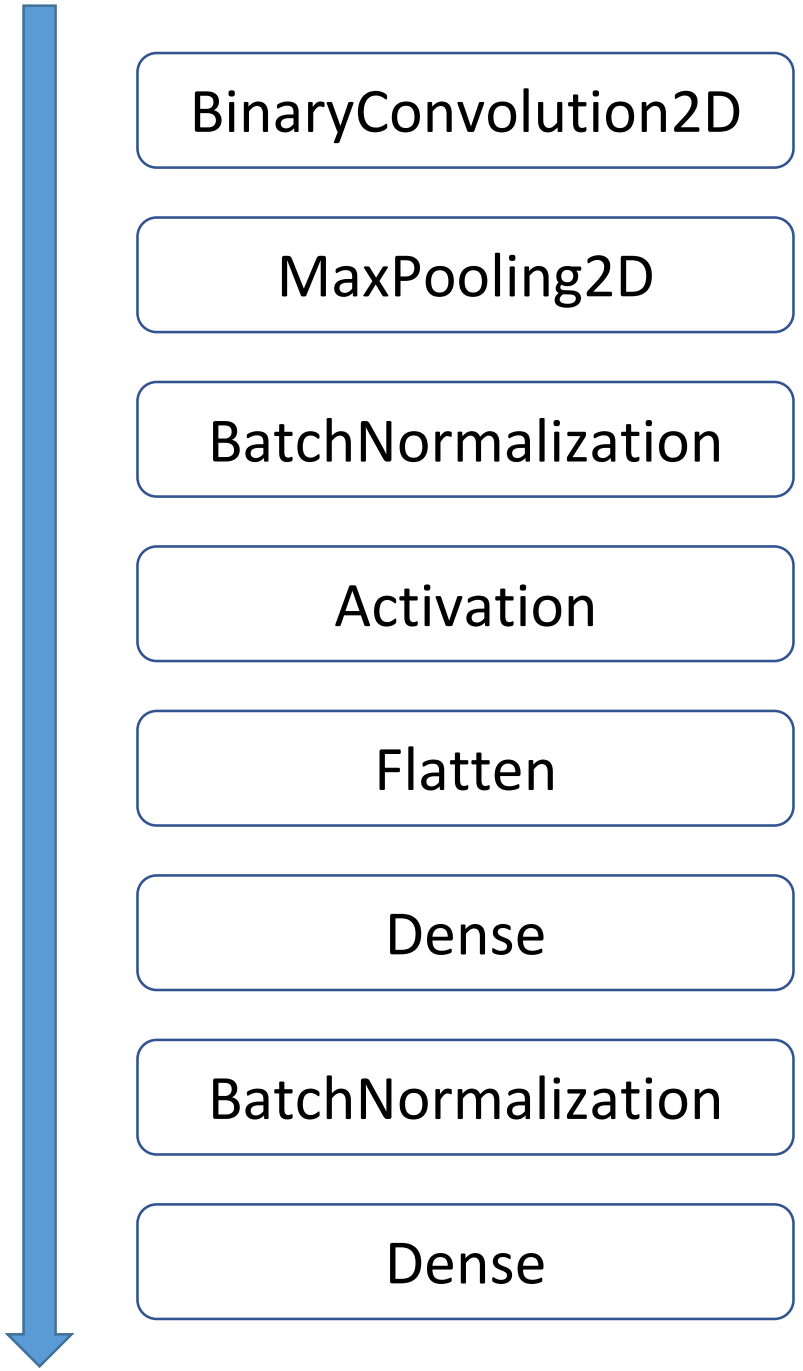


-1	1	1
1	-1	-1
1	1	-1

二值化後之 Kernel

BNN模型架構

Layer1 : BinaryConvolution2D
Layer2 : MaxPooling2D
Layer3 : BatchNormalization
Layer4 : Activation
Layer5 : Flatten
Layer6 : Dense
Layer7 : BatchNormalization
Layer8 : Dense



Kernel Decomposition

Kernel Decomposition 是將二值化的權重中 -1 的部分改為 0，形成 Filter Kernel，並且創建一所有的權重都是 -1 之 Base Kernel，在運算時，將原始圖片之部分擷取後，分別將其和 Base Kernel、Filter Kernel 做卷積運算，並將其結果相加，即可得到該位置卷積運算的結果。

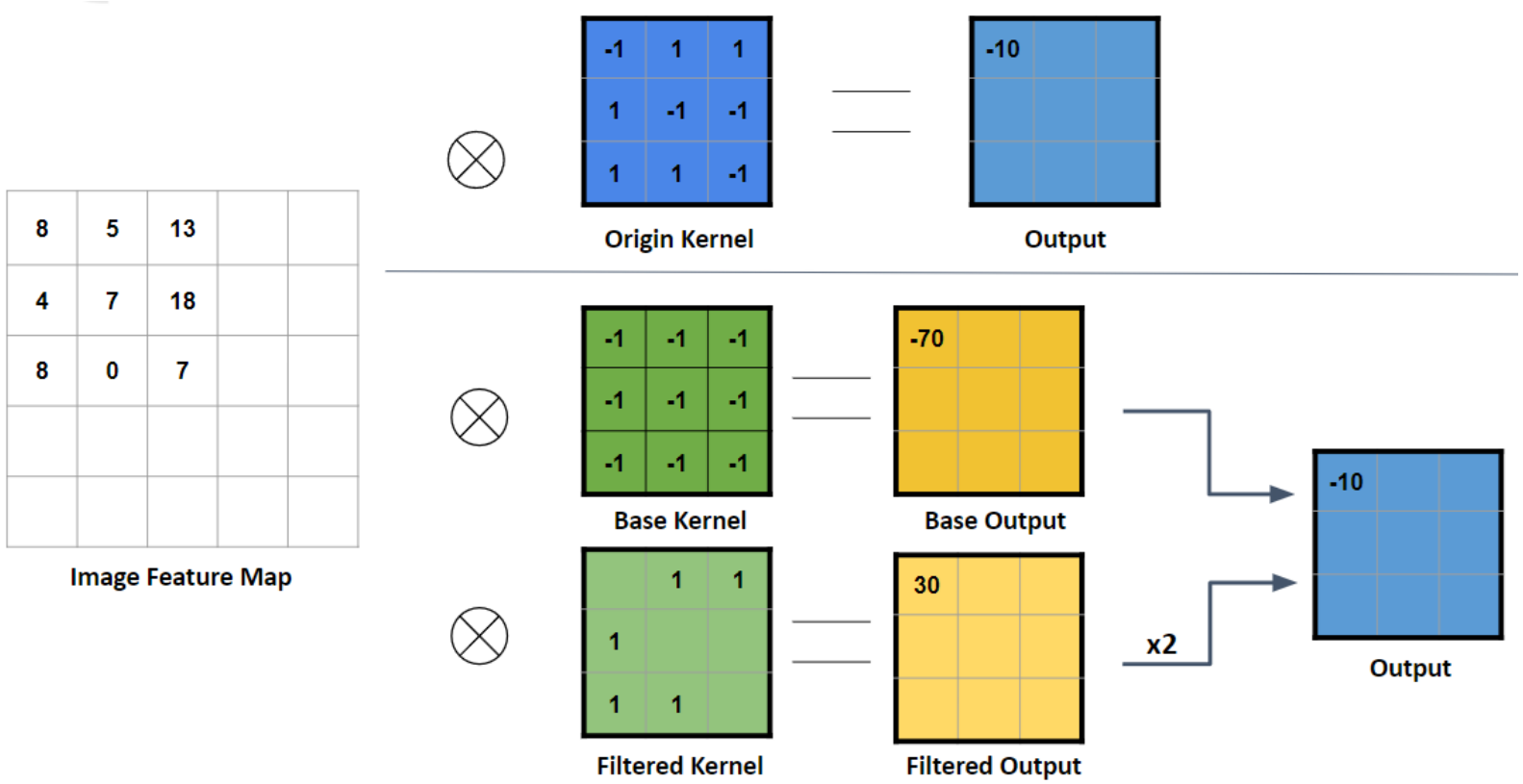
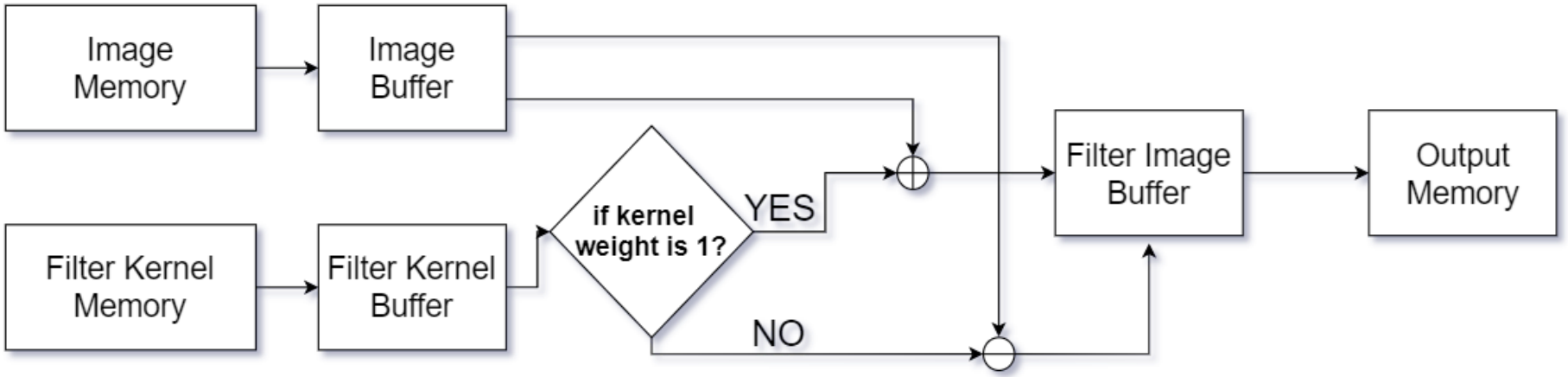


Image Compute Fusion

將 Output 先初始化為 0 ，之後判斷 Filter Kernel 中該位子的值是 1 或 -1 。

- 如果為 1 ，那麼便將 Output 和原始圖片做相加
- 如果為 -1 ，就將 Output 和原始圖片做相減

當 Filter Kernel 中的每個位子都判斷完畢且做完運算時，就將 Output 輸出。

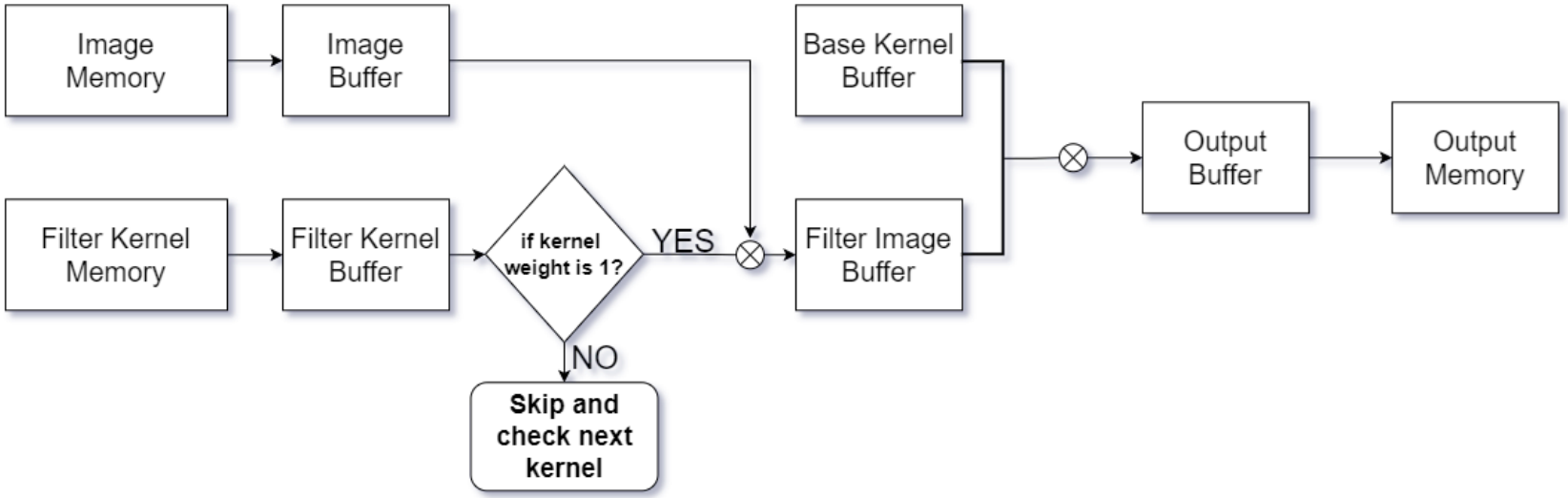


Weight Lookahead with Zero Skipping

先建出 Base Kernel，並且把 Filter Kernel 中值為 -1 的位置的值改為 0，之後先將原始圖片和Base Kernel做卷積運算，接著當原始圖片和 Filter Kernel 做卷積運算時，先判斷Filter Kernel中該位置的值。

- 如果為 1，則直接進行運算
- 如果為 0，則跳過該次運算過程，並判斷下一個權重的值

以此降低總執行時間並進一步減少總能耗。

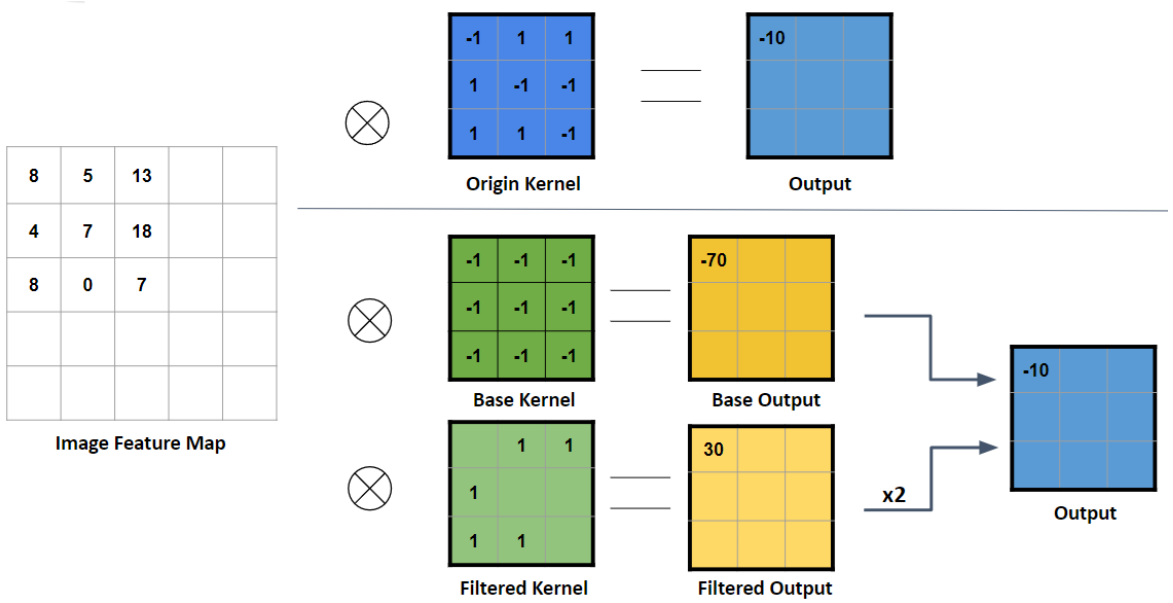


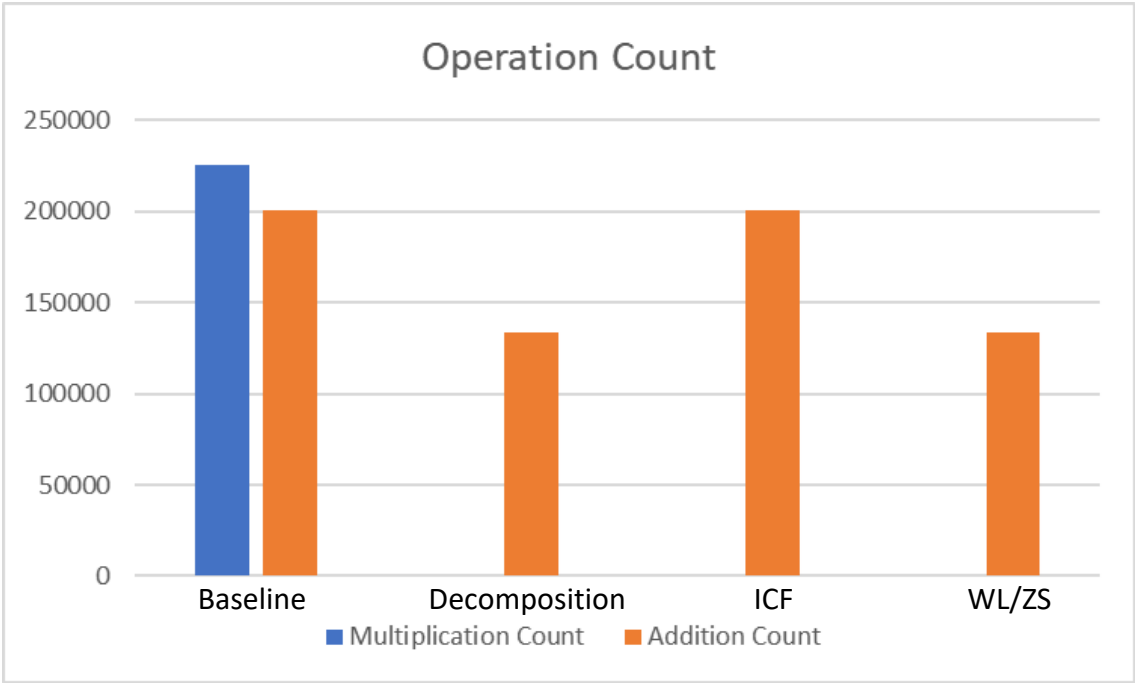
軟體模擬運算次數

用軟體實做傳統的 CNN 以及優化後的模型

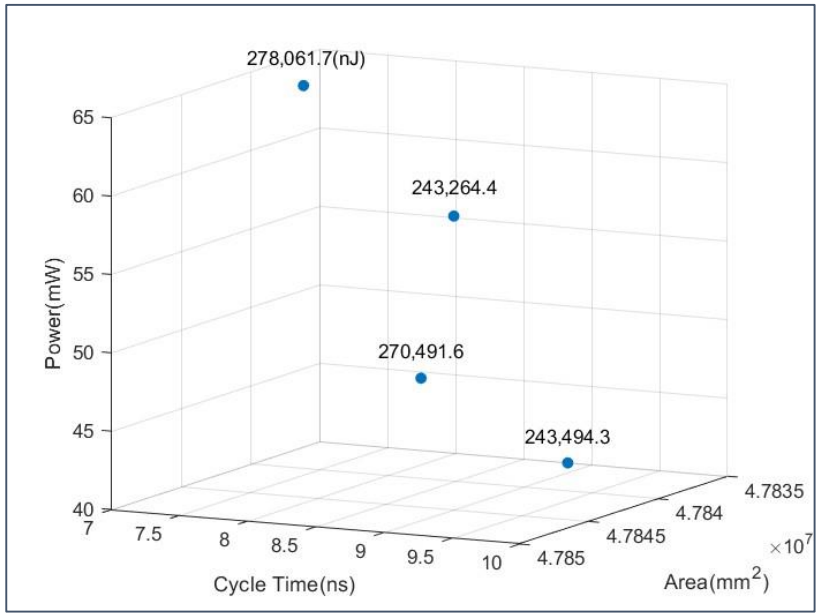
目的：
1. 配合 Convolution 硬體計算的結果，得到預測的結果

2. 使用不同的設計邏輯，並計算其運算次數

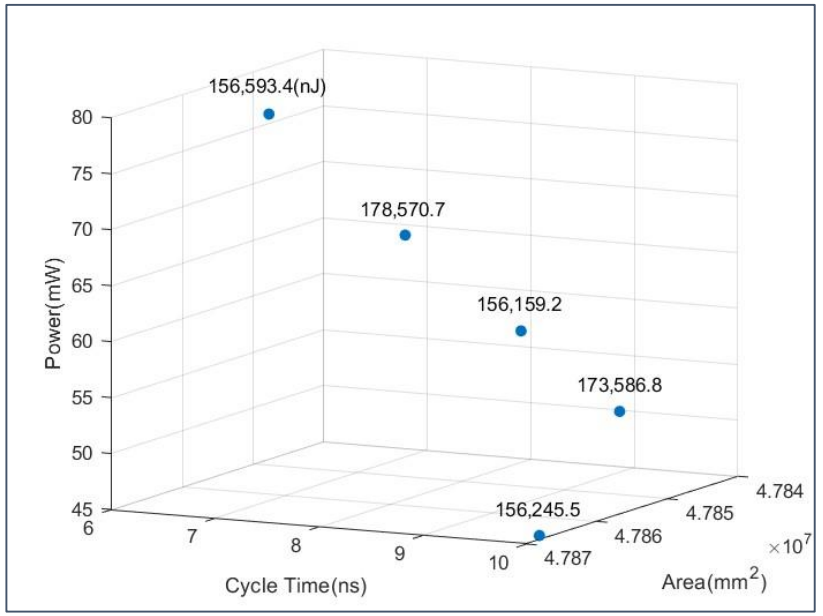




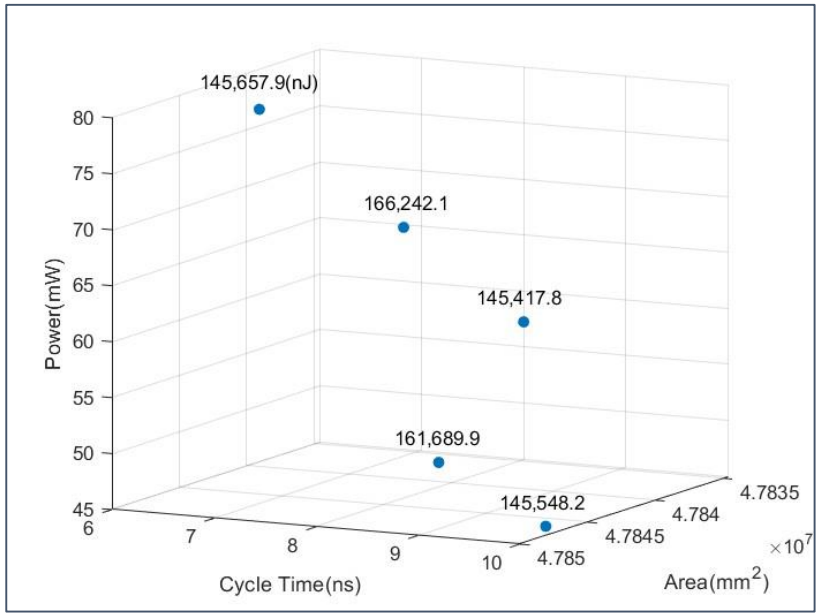
硬體成本比較



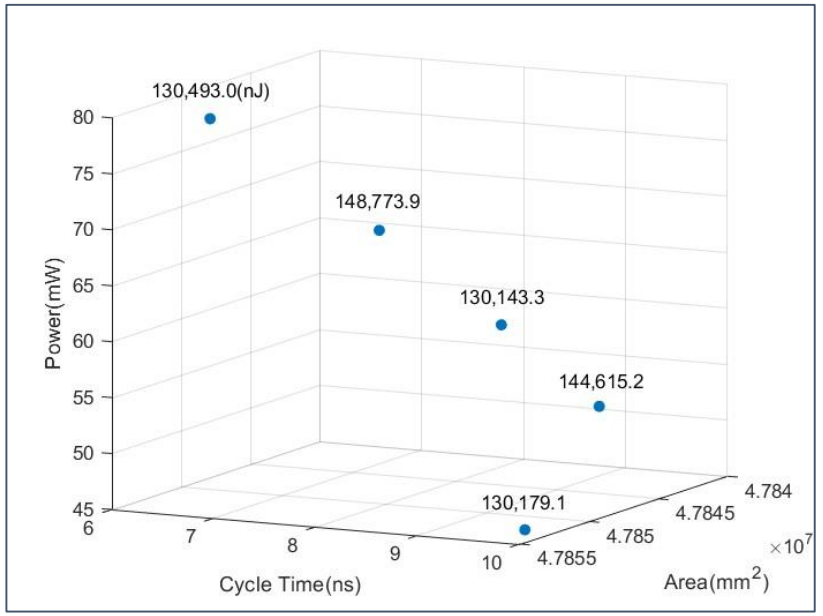
Baseline



Kernel Decomposition



ICF



WL/ZS

結論

我們提出了一個軟硬體協同設計，做了一個神經網路的二值化，硬體設計上，我們實作了 **Kernel Decomposition**，並提出了兩個硬體設計的方法，**ICF** 及 **WL/ZS**，和 **Kernel Decomposition** 的比較如下。

	Energy	Area
ICF	- 6.85%	- 0.04%
WL/ZS	- 16.68%	- 0.02%

10ns 下 Energy 與 Area 比較

謝謝大家