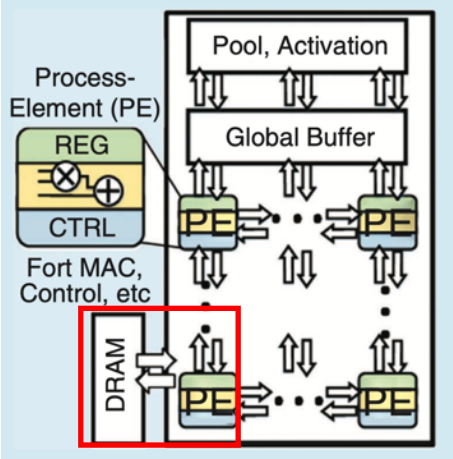
(一) 摘要

近年來，深度學習被廣泛運用在各個領域，包含影像辨識、自然語言處理等，並獲得卓越的表現。但是由於深度學習模型在一般的硬體設計下需要不斷將資料在運算單元和儲存單元間搬移(如圖(一))，導致效率受到限制，因此 CIM (compute-in-memory)硬體加速器這項技術被提出，也是目前最能有效提高運算效率的硬體架構。不過 CIM 硬體加速器會受到一些硬體非理想效應影響，導致運算過程產生誤差，因此雖然在理想狀態下較深的模型普遍擁有較好的表現，但是在存在誤差的情況下，較深的模型容易在運算過程中累積更多誤差，表現反倒不如較淺的模型。因此本研究嘗試使用深度及架構具有彈性的模型，使其能根據不同誤差程度進行動態推理(dynamic inference)，獲得優於一般模型的表現。目前能夠彈性調整模型深度的方法有兩個，第一個方法會在模型不同深度的位置新增輸出點供模型選擇，第二個方法則是利用 ResNet 模型不同區域間存在捷徑的特性，訓練另一個模型用來決定要使用哪些模型區域。本研究將在考慮硬體非理想效應的情況下實作以上兩種方法，嘗試設計出適合 CIM 硬體加速器的模型架構。

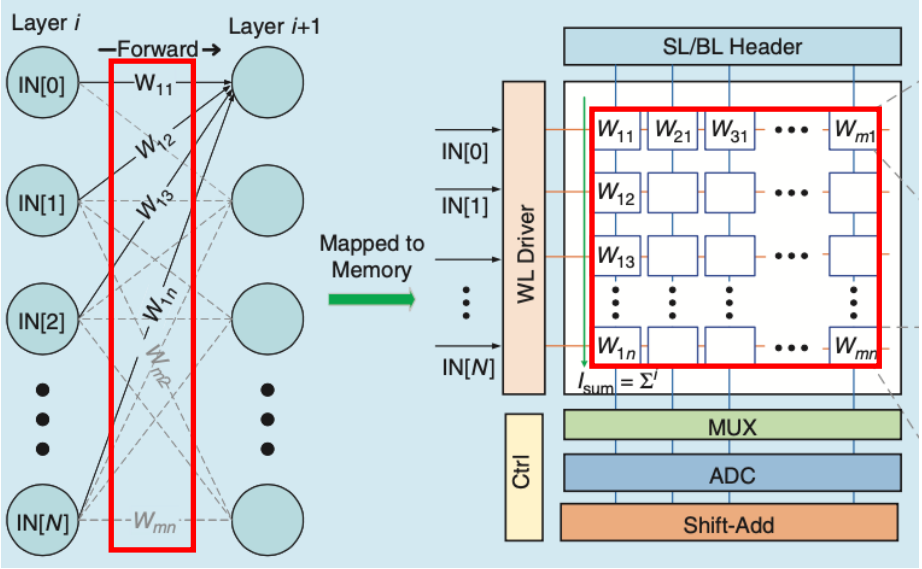
(二) 研究動機與研究問題

深度學習模型近期被嘗試實作在各種硬體裝置上，這樣的趨勢也使得如何在運算資源較少且儲存能量有限的邊緣裝置為其設計一個有效率的硬體加速器成為一個重要的課題。深度學習模型在運算時往往需要在層與層之間不斷對輸入值和權重值進行矩陣乘法運算，因此在一般的硬體設計下，需要頻繁地將權重值從儲存單元間搬移到運算單元(如圖(一))。但由於近年來儲存單元效能的成長速度遠小於運算單元(memory-wall-problem)，因此這樣的設計使得運算效率受到嚴重限制。為了解決這個問題，CIM (compute-in-memory) [1] 的設計利用儲存單元的電導值記錄權重值，並在儲存單元完成輸入值(類比電壓訊號)和權重值(電導值)的矩陣乘法運算(如圖(二))，有效改善運算效率。

但是 CIM 的設計並不只有正向的幫助，有些硬體的非理想效應會造成運算誤差。舉例來說，記錄權重值的電導值就會存在誤差，而層層誤差的累加可能導致模型最後得到不同的結果。因此普遍表現較好且深度較深的模型反而容易累積更多的誤差，導致最終表現可能不如較淺的模型，這樣的特性使得一般模型在各個誤差程度不一的硬體裝置上表現容易起伏不定。因此如何在設計模型的階段增加架構彈性，使得在使用 CIM 硬體加速器的裝置上能找到最適合的模型架構成為本研究最大的目標。



圖(一)　一般硬體架構需要不斷將資料在儲存單元和運算單元間搬移 [1]



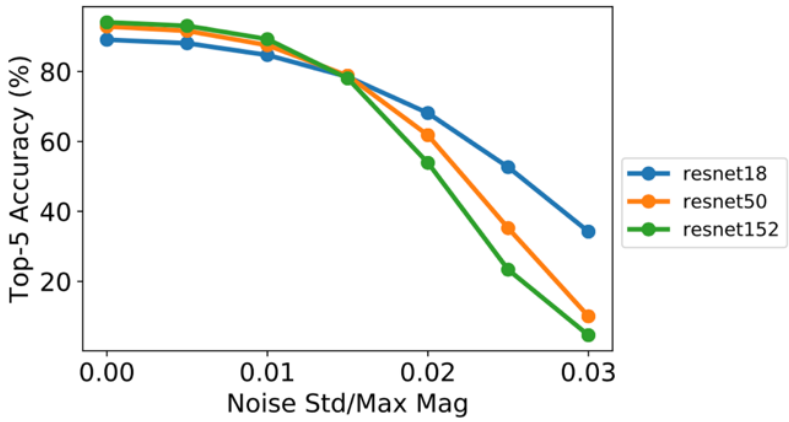
圖(二)　CIM 架構在儲存單元完成輸入值和權重值的矩陣乘法運算 [1]

(三)　文獻回顧與探討

在設計模型架構之前，本研究首先回顧硬體非理想效應對於不同深度模型的影響。

(1) 硬體非理想效應之影響 [2]

在論文中，作者在模型每一層輸出經過激活函數(activation function)的時候加上高斯分佈的雜訊，模擬硬體非理想效應的影響。接著作者調整不同大小標準差的高斯分佈雜訊，並使用不同深度的 ResNet 模型進行影像辨識，發現較深的 ResNet 模型儘管在理想狀態下擁有更高的正確率，但是隨著雜訊增大，正確率下滑地也更快，甚至當雜訊分佈的標準差夠大時，正確率反倒會低於較淺的模型(如圖(三))。這樣的結論顯示出在誤差程度不同的裝置上，表現最佳的模型深度可能會不盡相同，因此一般架構固定的模型難以在各個 CIM 架構硬體裝置上維持穩定的表現，也是本研究欲改善的問題。

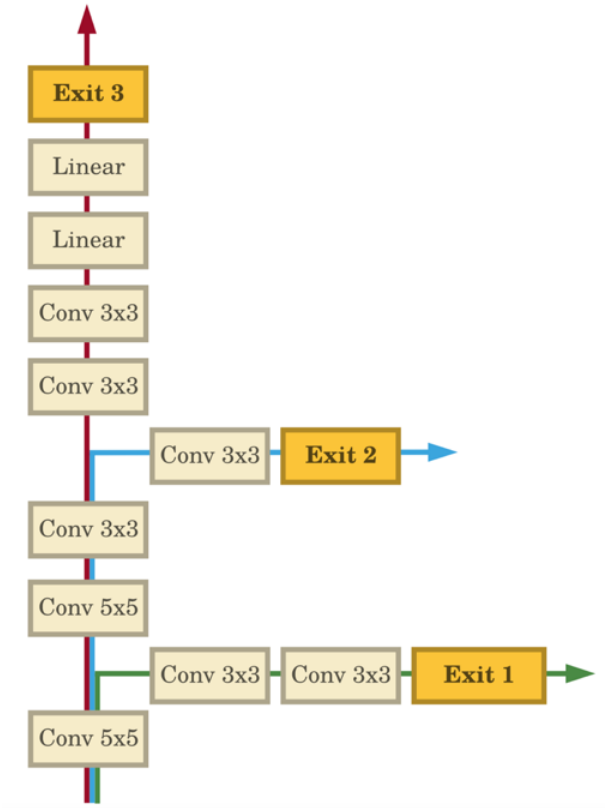


圖(三)　不同深度的 ResNet 模型在不同誤差程度下的表現 [2]

彈性調整模型架構及深度的方法目前大多用來提升模型運算效率，所以本研究從相關文獻中擷取兩種方法進行探討。第一種方法為多輸出模型架構，透過在模型不同深度的位置新增輸出點提供模型深度上的彈性。第二種方法則是利用 ResNet 模型不同區域間存在捷徑的特性另外訓練一個模型，用來決定要省略哪些部分的模型，具有彈性調整模型深度及架構的能力。以下將詳細回顧兩篇文獻。

(2) 多輸出模型架構 (multi-exit architecture) [3]

在論文中，作者設計一個具有三個不同深度輸出點的模型(如圖(四))，執行相同的影像辨識任務(dataset: MNIST / CIFAR10)。模型在執行任務時，作者會為各個輸出點設定門檻值，當輸入值運算到其中一個輸出點，模型會計算輸出值 entropy(如圖(五))。當 entropy 高於門檻值，代表在假定輸出結果正確的情況下，輸出值擁有足夠低的 loss，因此作者認定模型對於此結果具有足夠信心，即無須完成模型更深層的運算，藉此提高任務執行的效率。因此為了讓更多的資料能在淺層的輸出點結束運算，並且降低淺層輸出點誤判結果的機率，淺層輸出點的表現尤其重要。所以在模型訓練階段，作者分別對最淺、中等、最深的輸出點給予最大、中等、最小的權重 ，乘上各自輸出點的 cross-entropy-loss 之後加總，成為模型最佳化的目標函數。此研究最終也在正確率幾乎不受影響的表現下有效提升運算效率。



圖(四)　多輸出模型包含多個不同深度的輸出點 [3]

$$\text{entropy}(\boldsymbol{y}) = \sum_{c \in \mathcal{C}} y_c \log y_c,$$

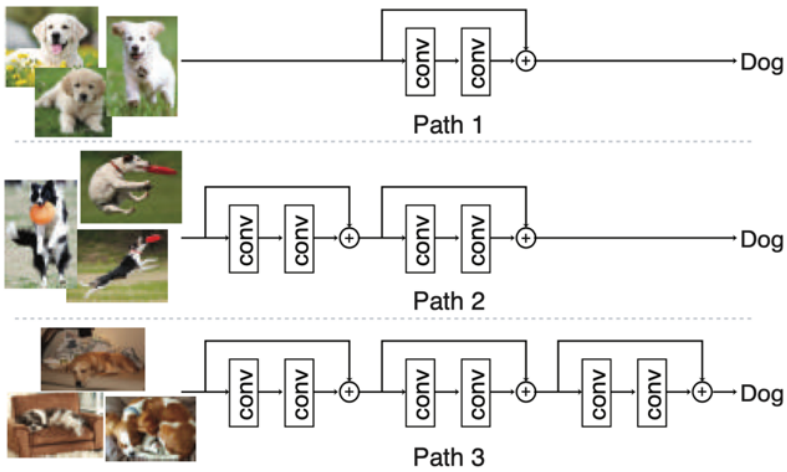
where $\boldsymbol{y}$ is a vector containing computed probabilities for all possible class labels and $\mathcal{C}$ is a set of all possible labels.

圖(五)　entropy 公式 [3]

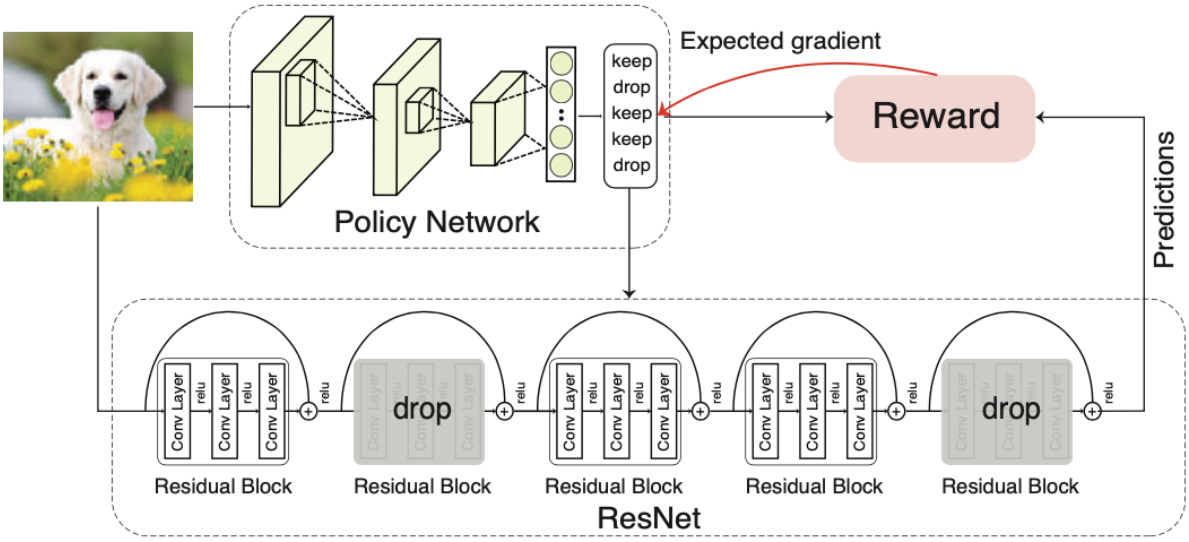
(3) ResNet 模型動態架構 (BlockDrop) [4]

在論文中，作者利用 ResNet 模型不同區域間存在捷徑的特性，以強化學習(reinforcement learning) 的方式訓練另一個模型(policy network)，使其能夠根據不同的輸入決定要使用模型的哪些區域(如圖(六))，藉此提高任務執行的效率。所謂強化學習是指為模型的若干行為定義獎懲使模型在最佳解未知的情況下學習特定任務。在訓練過程中，作者首先選擇一個已經訓練好用來執行影像辨識任務(dataset: CIFAR10 / IMAGENET)的 ResNet 模型，而 policy network 會決定要使用哪些部分的 ResNet 模型，接著 ResNet 模型會根據 policy network 的結果使用部分區域進行影像辨識(如圖(七))。作者為了讓模型盡量使用較少的區域，當模型辨識正確，會獲得 1-(模型使用率)^2 的獎勵，不過辨識的正確性畢竟也非常重要，所以當 ResNet 模型辨識錯誤，反倒會獲得常數值 -c 的處罰，c 值會根據使用者比較重視運算效率還是正確率進行調整。值得注意的是由於模型各個區

域是否使用的所有組合跟模型深度為指數關係，所以一般的訓練方式容易因為搜索空間(search space)維度過大導致效率不佳。因此作者在模型訓練的第 t 個 epoch 時只會決定模型最後 t 個區域是否使用，前面的區域都會直接使用，這樣的方法會讓訓練過程更快速且穩定，又稱為 curriculum learning。完成訓練之後，作者最後會再根據 policy network 的結果對 ResNet 模型進行少量的訓練，以獲得更好的整體表現。此研究最終不止有效提升運算效率甚至獲得更高的正確率。



圖(六)　ResNet 模型動態架構示意圖 [4]



圖(七)　policy network 的訓練過程 [4]

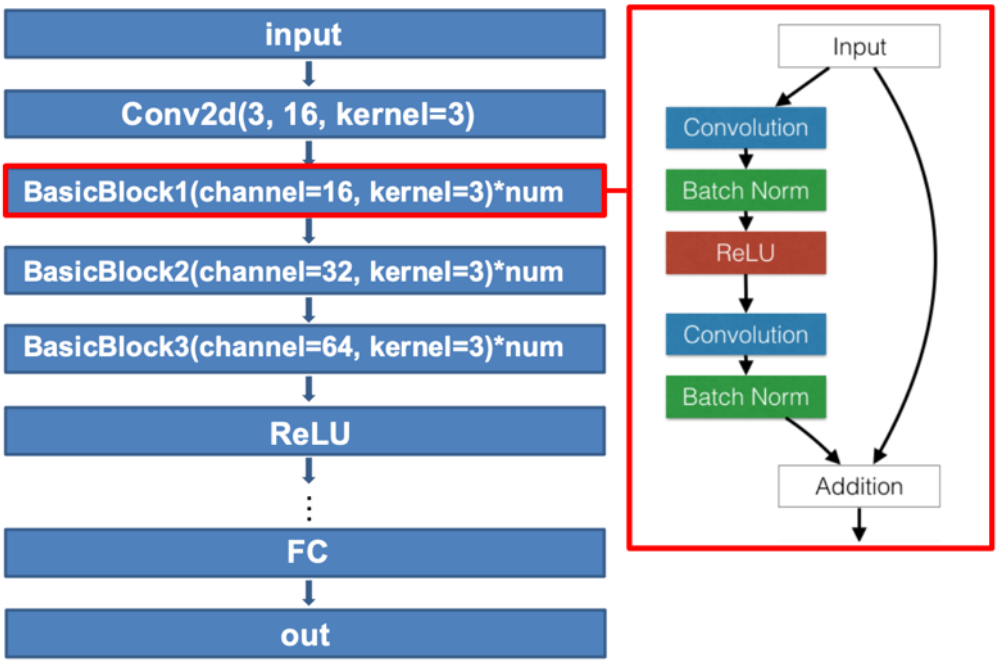
雖然本研究的重點目標不在於提高模型效率，但是基於非理想效應造成的誤差容易隨模型加深而增大，我們同樣需要能夠調整多種深度的模型並且希望大部分的輸出值能在淺層的輸出點完成運算，因此與上述兩種模型的訓練目標不謀而合。

(四) 研究方法及步驟

本研究欲嘗試以上兩種不同模型架構是否能應用在考慮硬體非理想效應的情況並獲得優於一般模型的表現，所以首先必須要有一套模擬硬體非理想效應的方法。因此本研究引入一套工具 DNN+NeuroSim [5]，這套工具可以模擬量化訓練(quantization-aware-training)的模型被實作在 CIM 架構硬體裝置上的表現。所謂量化訓練是指在模型的訓練過程中將權重限制到有限精度的訓練方法。舉例來說，如果有一層神經網路的權重範圍為[-100, 100]，且量化訓練要把權重限制到 b 個二進制位元，則針對權重 w 會將其調整為 $[w/100 * 2^{(b-1)}]/2^{(b-1)} * 100$ 。

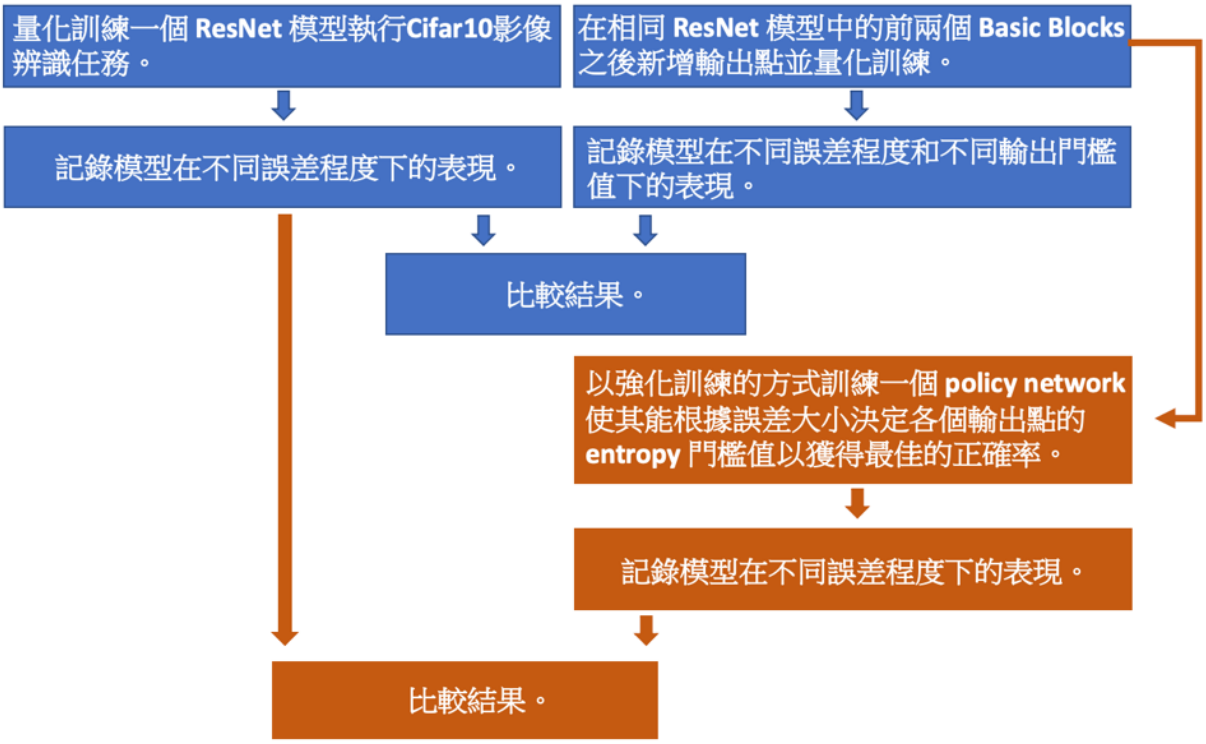
將模型套入 DNN+NeuroSim 之後，它會實際模擬將權重記錄到各個儲存單元的過程，並加入硬體非理想效應的模擬，包含儲存單元電導值記錄權重時的誤差、儲存單元電導值隨時間發生飄移的誤差以及將類比輸出訊號轉換成數位訊號時轉換器精度不足造成的誤差。由於後兩者在不同類型的記憶體或是不同的周邊電路設計下會有不同的影響，難以量化，因此本研究將其設定為控制變因，然後將儲存單元電導值記錄權重時的誤差設定為操縱變因。此套工具透過在電導值加入常態分佈的誤差模擬非理想效應造成的影響，本研究將以不同大小標準差的常態分佈代表不同大小的誤差。

擁有可以量化並模擬誤差程度的工具之後，本研究以 ResNet 模型為原型(如圖(八))，針對兩種不同的模型架構進行研究。



圖(八) ResNet 模型架構

(1) 多輸出模型架構

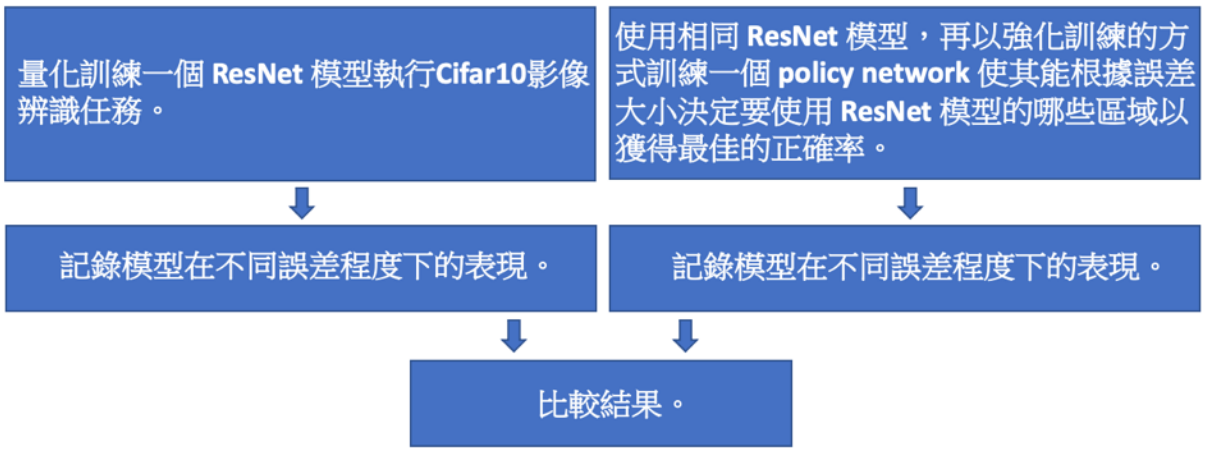


圖(九)　多輸出模型架構之研究方法及步驟

模型架構	輸入	各輸出點 entropy 門檻值
本研究	圖片、**權重誤差大小**	**policy-network 決定 (優化正確率)**
文獻 [3]	圖片	**自訂 (在運算效率與正確率間權衡)**

圖(十)　模型架構比較 (本研究 vs 文獻[3])

(2) ResNet 模型動態架構



圖(十一) ResNet 模型動態架構之研究方法及步驟

模型架構	輸入	任務成功時 policy-network 的獎勵
本研究	圖片、權重誤差大小	定值 (優化正確率)
文獻 [4]	圖片	與 ResNet 模型使用率呈負相關 (在運算效率與正確率間權衡)

圖(十二) 模型架構比較 (本研究 vs 文獻[4])

(五) 預期結果

加入權重誤差後的模型表現：

文獻多輸出模型 (entropy 門檻較高)	文獻 ResNet 動態模型 (policy-network 獎勵和模型深度的相關係數較低)	一般模型
佳	佳	不佳

文獻多輸出模型 (entropy 門檻較低)	文獻 ResNet 動態模型 (policy-network 獎勵和模型深度的相關係數較高)	一般模型
更佳	更佳	不佳

本研究之多輸出模型	本研究之 ResNet 動態模型	一般模型
最佳	最佳	不佳

圖(十三) 階段性預期結果

(六) 參考文獻

[1] S. Yu, H. Jiang, S. Huang, X. Peng and A. Lu, "Compute-in-Memory Chips for Deep Learning: Recent Trends and Prospects," in IEEE Circuits and Systems Magazine, vol. 21, no. 3, pp. 31-56, thirdquarter 2021, doi: 10.1109/MCAS.2021.3092533.

[2] T. Yang and V. Sze, "Design Considerations for Efficient Deep Neural Networks on Processing-in-Memory Accelerators," 2019 IEEE International Electron Devices Meeting (IEDM), 2019, pp. 22.1.1-22.1.4, doi: 10.1109/IEDM19573.2019.8993662.

[3] S. Teerapittayanon, B. McDanel and H. T. Kung, "BranchyNet: Fast inference via early exiting from deep neural networks," 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 2464-2469, doi: 10.1109/ICPR.2016.7900006.

[4] Z. Wu et al., "BlockDrop: Dynamic Inference Paths in Residual Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8817-8826, doi: 10.1109/CVPR.2018.00919.

[5] DNN+NeuroSim Framework V1.3, (https://github.com/neurosim/DNN_NeuroSim_V1.3)

(七) 需要指導教授指導內容

(1) CIM 硬體加速器之原理。

(2) DNN+NeuroSim 硬體非理想效應模擬之原理。

(3) 機器學習模型的原理及設計方式。

(4) 數據處理及分析。

(5) 研究中遇到的問題。