國立清華大學 電機工程學系

實作專題研究成果摘要

Spiking Neural Network Processor with Integer Leaky Integrate-and-Fire Neuron

搭載整數化漏電式累積放電神經元模型之突波神經網路處理器

專題領域： 系統領域

組　　別： B322

指導教授： 鄭桂忠 教授

組員姓名： 黃于菲 、 劉奇泓

研究期間： 112年 1月29 日 至 112 年 11月15日 止，共 10.5個月

摘要

近年來深度學習的快速發展，主導了人工智慧的主要技術，機器學習之應用也逐漸充斥在我們的生活之中，而隨著其應用越來越廣，神經網路的模型也越來越複雜，使神經網路的學習演算法實現於終端裝置成為挑戰。現今主流之深度學習神經網路模型相較於如人腦等生物體神經系統，其功耗與效率都差距甚遠，而隨著近年的系統神經科學發展，可觀察至以細胞為尺度的神經網路連結圖譜，因其擁有高度的訊號迴授傳遞，神經模型可以神經群為單位建立與模擬各功能區塊的神經系統架構。

突波式神經網路模仿了生物大腦的特徵，利用突波傳遞訊息，並使用更接近生物的表現，其低延遲、低運算需求、低功耗的特性也使得突波式神經網路被視為具有極高的潛力。本研究以仿生角度設計出以神經群模型的突波神經網路處理器，神經元的設計上，以 Leaky Integrate-and-Fire(LIF)神經元模型為雛形，設計硬體友善之神經元。而在突觸的連接與運算累積上，透過調控式切割神經群系統，優化應用多神經群時的運算靈活性，進而增加運算速度。

在本研究的設計中，突波神經網路處理器可利用參數調整、切換在神經元數量以及神經群配置，應用各式各樣的突波神經網路。不管是以雙層全連接層架構辨識手寫數字圖片，可達到92%的正確率，或以 RNN 網路架構處理4x4的數獨填寫，可98%正確率分析結果，證明本研究設計出從仿生角度出發，在應用端具高效率、低運算需求的突波神經網路處理器。

1. Introduction

1-1. Research Background

人工智慧(Artificial Intelligence, AI)的發展幾乎與近代電腦的興起同步，許多年來經歷幾番起落更迭，到最近再度受到全面的重視。這是因為 AI 的機器學習領域近來取得重要突破，讓許多過去無法有效解決的問題如語音辨識、影像辨識、電腦圍棋等，能夠實質解決並帶出具商業價值的實際應用。因此幾乎所有主要網路及電腦公司都設立人工智慧及大數據部門，各重要大學及研究機構也都紛紛建立機器學習研究團隊。這場如海嘯般顛覆性的科技變革正在引領我們步入人工智慧的新時代，就像蒸汽機之於第一次工業革命，電氣之於第二次工業革命，數字化之於第三次工業革命，人工智慧正在引發「第四次工業革命」。

現今蓬勃發展之卷積神經網路加速器雖能以多層結構處理較複雜的演算法，但其深度結構和大量的參數導致了計算複雜性的增加，需要大量的計算資源，尤其是在訓練大型網路時可能導致訓練時間較長和對高性能硬體的需求，而相較於卷積神經網路，突波神經網路仍具有以下四點優勢：

1. 突波訊號具二值化(Binarization)：僅記載突波發生與否，因此在後端突觸累積(Post-synaptic accumulation)只需要加法運算，在沒有乘法運算下，於邊緣硬體上應用可節省大量運算能耗(Fig.1-2)。
2. 突波資訊至時間維度相關性(Spike time-correlation)：使得突波神經網路能夠充分利用基於時域的訊號，適合處理和感知連續訊號與影像。
3. 突波訊號具稀疏性(Sparsity)：透過Address Event Representation(AER)的編碼處理下取代大量的突波序列傳遞，使得訊號傳輸頻寬(Transmission band-width)需求大幅降低。訊號雖少，但在有時間維度下亦紀錄了有價值的信息。
4. 累積放電（Integrate-and-fire）型神經元：實現非連續的信息傳遞，並發出不可微分的離散突波，且只有在接收突波訊號或者神經元膜電位(membrane potential)累積超過閾值進而發出突波訊號。因此突波神經網路屬於事件驅動(event-driven)系統，儘管需要計算神經元每個時刻的膜電位累積的變化值，但相較之下依舊能節省整體能耗。

憑藉著突波訊號稀疏性(Sparsity)、突波資訊至時間維度相關性(Spike time-cor-relation)、事件驅動系統(Event-based system)特性和對於訊號與權重傳輸頻寬(Band-width)以及總運算需求低，使得突波神經網路擁有低運算需求、低功耗、低延遲的優勢，儘管發展時間較卷積神經網路長，依舊成為了未來最具潛力的第三世代神經網路。

2. Research Methodology

2-1. System Design

2.1.1. Spiking Neuron Network Processor

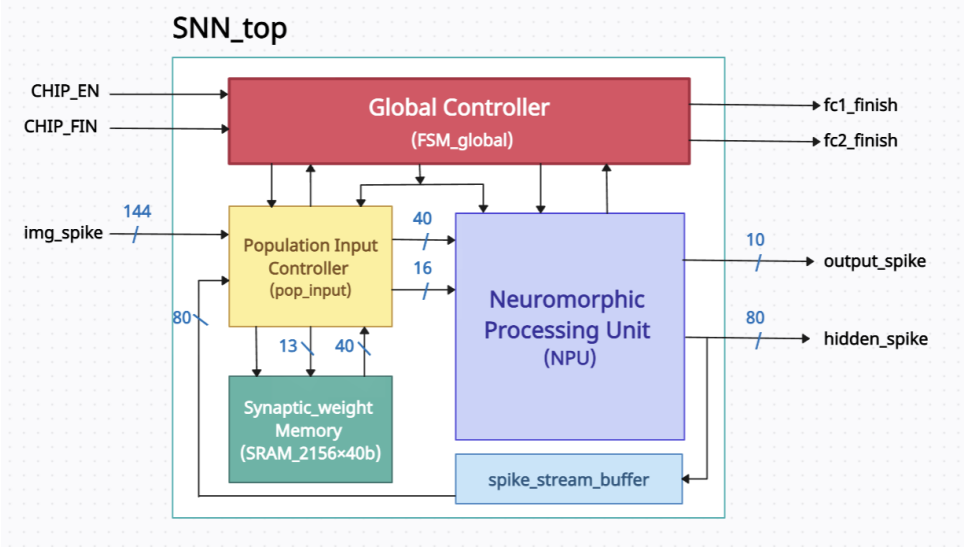


Fig.1 突波處理器整體架構圖

本研究之突波神經網路處理器整體架構如 Fig.1所示，在處理器左方代表輸入訊號包含起始、結束訊號以及圖像突波輸入，在處理器右方代表輸出訊號包含神經網路各層狀態、輸出突波等，而處理器內主要可以分為五大區塊:

1. 主控制器(Global Controller): 整體由有限狀態機(Finite State Machine，簡稱FSM)控制，可控制接收突波輸入至內部暫存器，及更新神經處理單元(NPU)內神經元膜電位、輸出突波，以及各區塊間的起始、結束之相互關係。
2. 突波輸入控制器(Population Input Controller): 接收突波並逐一檢查各輸入，若讀取到突波訊息後，控制器會將突波發生的位址換算成記憶體中存取該神經元前突觸權重之位址，並將相對應的突觸權重自權重記憶體取出，分配到神經處理單元(NPU)內神經元中。
3. 神經處理單元(Neuromorphic Processing Unit, NPU): 將神經元根據配置進行分組、分群，並根據突波輸入控制器輸入的權重分配到神經群中，並控制神經群進行累加運算、更新膜電位以及輸出突波。
4. 突波陣列暫存器(Spike Stream Buffer): 將輸出存放至突波暫存器，如有設定下一層神經網路則會在該層運算結束時輸入到突波輸入控制器，成為下一層運算時的輸入突波。
5. 突觸權重記憶體(Synaptic Weight Memory): 使用2156x40位元的 SRAM 配置，儲存各突觸之權重量值。

2.1.2. Neuromorphic Processing Unit (NPU)

神經群仿生處理單元(NPU)內主要為神經群，接收來自神經群輸入控制器(Populational Input Controller)所分送的突觸權重並累加於神經元後突觸上。累加完後，所有神經元會同步輸出突波，藉由暫存器存取此時間步的突波陣列訊息。神經組內有數顆神經元，以多層或單層結構分群，再將神經群分成組以降低系統描述複雜度，群組數量的配置會依照應用方向而有所變動。

採用 Leaky Integrate-&-Fire(簡稱 LIF)神經元，以數位訊號表達膜電位偏微分方程(式1.)與 激活方程式(式2.):

$$V[t] = V[t-1] + \frac{1}{\tau}\big(X[t] - (V[t-1] - V_{reset})\big)$$
$$V_{reset} = 0 \Rightarrow V[t] = \Big(1 - \tfrac{1}{\tau}\Big)V[t-1] + \tfrac{1}{\tau}X[t] = \tfrac{1}{\tau}(V[t-1] + X[t]) \quad \text{(式1.)}$$

$$V[t] = \begin{cases} V_{threshold} & if\ V[t] > V_{threshold} \\ V[t] & if\ 0 \le V[t] \le V_{threshold} \\ 0 & if\ V[t] < 0 \end{cases} \quad \text{(式2.)}$$

式1.中，推導出在 t 時刻的膜電位為: (t-1)時刻的膜電位 與 t 時刻的後突觸累加 X[t]之和並除以漏電係數τ，在硬體端使用位移實現漏電。式2.中，$V_{threshold}$為閾值膜電位，當神經元膜電位超過$V_{threshold}$時，該神經元會於該時間步發出突波並重置膜電位；反之則讓膜電位維持不變，或在膜電位為負值時歸零。

神經群輸入控制器會將一次時間步內的該神經元所接收到的所有權重累加至後突觸累加器(Post-synaptic accumulator)中，累加完成後會送至膜電位並經過激活方程式(Activation Function)決定此時間步的神經元突波輸出情形與下個時間步的膜電位。

2-2. Application – Handwritten Digit Recognition

我們使用 MNIST dataset 用於訓練 SNN 模型來識別手寫數字的正確數字。每個 MNIST 圖像都有一個對應的標籤表示該圖像中的數字，因此這是一個監督學習(Supervised Learning)的數據集，並且可以用於訓練和評估分類模型的性能。此應用之設計流程如下: 首先利用 Pytorch 建立 SNN 模型以及設計訓練方法，將 MNIST dataset 匯入進行模擬，並且我們利用 OpenCV 自行繪製手寫圖片放入模型進行測試。在完成軟體模擬後，我們將訓練出的突觸權重取出，放入硬體加速器中的 SRAM，並輸入手寫圖片進行測試，最後在輸出端得到結果。

在軟體的部分，我們使用到 Pytorch 進行模型建構以及模擬，在神經元連接路徑的部份我們定義出前向路徑(forward path)之描述式，也代表典型的突波神經元輸出，若膜電位大於閥值則輸出突波為1、小於則輸出為0；後向路徑(backward path)之描述式，我們使用到 arctangent function 的微分式，目的是為了在 backpropagation 演算法中所需之梯度函數，於每個 timestep 中計算 loss function 並進行優化。

我們建立 SNN 模型的方式參考了多數 CNN 模型後端使用的全連接層(fully connected layer)，建構出了輸出層為144個神經元、隱含層為80個神經元、輸出層為10個神經元的 SNN 模型(Fig.2)。進行訓練後，我們提取出各突觸權重並進行量化(quantization)，使原本為浮點數的權重值變為-介於128~127間的整數，使我們可以用硬體友善的方式將模型參數用8位元表示。在訓練模型後，我們利用 OpenCV 中的畫布功能自行手寫數字(Fig.2)，將圖片由原本的360x360像素轉為12X12像素且二值化的圖片，確認模型能成功辨識手寫數字圖片。

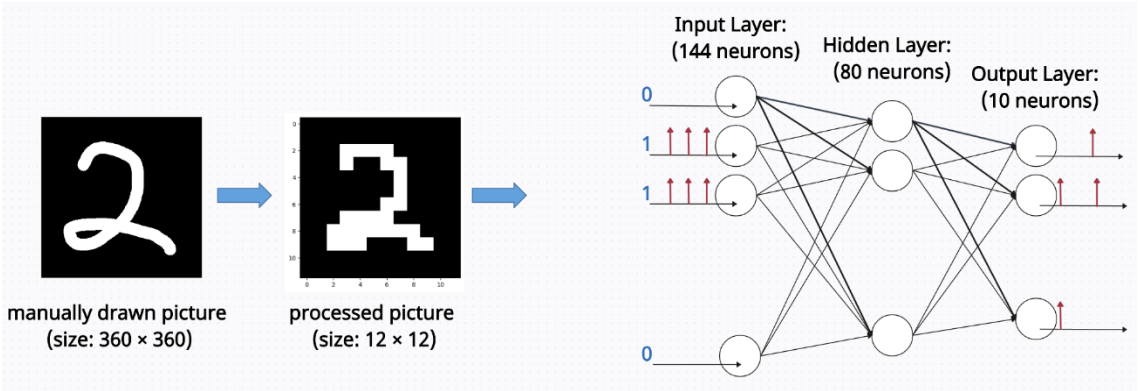


Fig.2 SNN 模型架構圖以及 OpenCV 手寫數字圖像預處理流程

在硬體的部分，我們將軟體模擬設計出的 SNN 全連接層架構同樣在處理器的 NPU 中硬體實現，並在後級使用一個突波累加器(Spike Accumulator)收集每個時間步的突波輸出，並採取贏者全拿(Winner-Take-All)的方式決定手寫圖片辨識結果。

我們在 NPU 中建構出神經網路架構以及 LIF 神經元，並且將神經元以每五顆做為一神經組(Neuron group)，用兩個神經群(Neuron cluster)代表著 SNN 全連接架構中的隱含層與輸出層，因此隱含層中含有16個神經組，輸出層則有兩組神經組，我們利用從 pop input 輸入的 load group 以及 weight 訊號分配權重至各神經組，以利其神經元進行突觸累加以及更新膜電位和突波輸出。

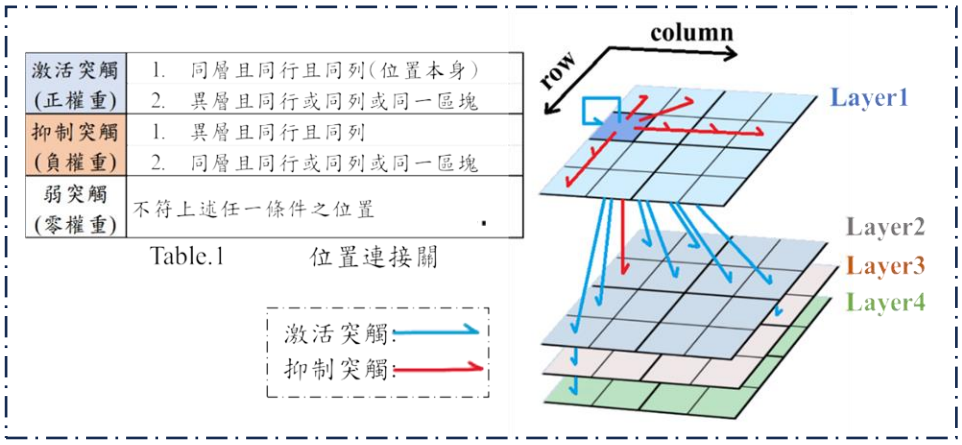
整體系統在一個時間步(Timestep)內的流程如下: 首先我們在 pop input 接收手寫圖片轉換成的突波輸入，並且根據輸入的突波將隱含層中相對應的突觸權重值經由 SRAM 取出、分配到 NPU 中的隱含層神經群中，結束分配後將各神經元膜電位更新、突波更新，最後將突波經由暫存器回傳至 pop input，成為下一層的輸入突波，並且用同樣的方式將輸出層神經元更新、得到輸出突波，如此一來便完成了一個時間步。

2-3. Application – Sudoku Answer Prediction

數獨是由 N*N 的方格組成，又可分為 N 個$\sqrt{N} \times \sqrt{N}$的小區塊。數獨的規則在每一行、每一列以及每個小區塊都會有1~N 個不重複的數字，亦即每一行、列與小區塊皆不會出現相同的數字。通常對於一個數獨問題會存在多組解，在此專題實作我們專注於只有唯一解的數獨問題。我們將未完成的數獨根據數字分層成三維陣列(層,列

,行)，再將此陣列攤平成一維突波陣列輸入至神經網路之中，可以觀察到突波神經的輸出訊息會回傳至輸入層，以達成循環神經網路的架構，最後會再藉由輸出突波的累積多寡預測空格中應填寫的數字為何。

4*4數獨新增了數字層概念，共有4*4*4=64格，每格皆為相互連接的神經元，而突觸連接又分為激活突觸(Excitatory)與抑制突觸(Inhibitory)。激活突觸會將正權重對外送入其他神經元的膜電位中，效果為增加該神經元的突波輸出機會；抑制突觸則對外送出負權重，減少該神經元輸出突波的機會。依據數獨規則，若位置(層,列,行)=(1,2,1)之神經元為已填入數字1狀態(Fig.3)，而未有箭頭連接的突觸則歸類為弱突觸，權重為零。



激活突觸 (正權重)	1. 同層且同行且同列(位置本身) 2. 異層且同行或同列或同一區塊
抑制突觸 (負權重)	1. 異層且同行且同列 2. 同層且同行或同列或同一區塊
弱突觸 (零權重)	不符上述任一條件之位置

Table.1 位置連接關

Fig.3 抑制突觸與激活突觸配置一覽圖表

2. Experimental Results

3.1. Handwritten Digit Recognition

經由建立 SNN 模型以及訓練模型，我們引入 MNIST 數據集進行訓練，訓練過程中我們調整一些訓練參數例如訓練時間步 T、隱含層神經元數目等。訓練完成後，我們將 SNN 處理器內的 NPU 架設隱含層80顆神經元、輸出層10顆神經元的全連接層後進行模擬， 確認處理器的可合成性後，我們使用 PYNQ-Z2開發板進行電路在 FPGA 上的驗證，將處理器放進可編程邏輯中，並撰寫 Python 語言使處理器透過開發板內建的 AXI GPIO 接口向處理器輸入突波並接收輸出。

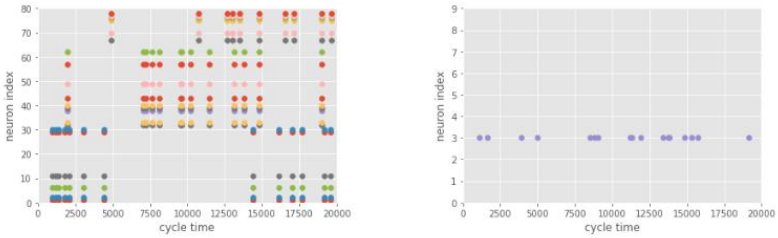
使用處理器後，我們使用 OpenCV 自行手寫出數字3，重複輸入圖片到處理器中，並將由 AXI GPIO 回傳的突波陣列整理為 Fig.4 的圖表，可以發現輸出突波能準確判讀出數字3。

3.2. Sudoku Result Prediction

將未完成數獨所產生的一維突波陣列送入網路，且利用後端突波累計器記錄每一顆神經元的突波輸出次數。4*4數獨的每一格會對應到4個數字層，分別為數字1、2、

3、4。以 Winner-take-all 決定每一格會填入的數字為何。

使用開源數據集[3] 測試準確率，數據集共有288筆數據集，每筆資料包含問題與解答，利用 Jupyter Notebook 將問題轉換成突波輸入陣列，以及依循數獨規則產生具有強弱連接的突觸權重，並導入 testbench 端的 SRAM 中。將以上參數及資料匯入至模擬得以求出正確率，見 Fig.5，Original Answer 為數據集中的解答，Predicted Answer 為突波神經網路的預測結果，藉由調整權重連接的強弱及閾值電位可以提升正確率，最終達到288筆中的285筆會是正確的預測結果，正確率為98.96%，平均約9個時間步預測出結果。



Hidden Layer　　Output Layer

Fig.4 隱含層與輸出層之輸出突波編號對時間點狀圖

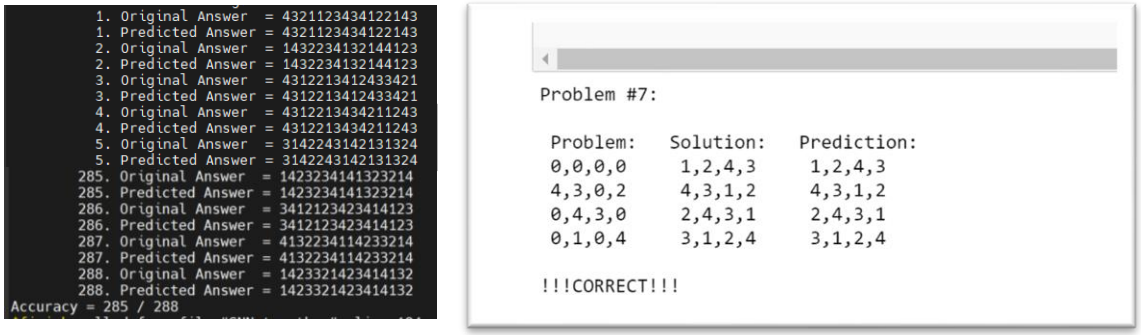


Fig.5 數獨數據集測試結果以及 Jupyter Notebook 中所顯示預測結果

利用與硬體模擬時相同的數獨資料，包括數獨題目、題目所轉成的突波陣列以及解答，導入至 Jupyter Notebook。完成電路 I/O 配置後，開始送入未完成數獨的突波陣列，經歷過5000個週期後，取得運算結果，並擷取每個周期輸出的突波陣列。

4. Conclusion

專題以仿生神經網路為主題設計出數位電路架構，還原出生物體的神經群的訊號傳遞行為，整體的應用結論包含以下兩點:

- 多層單向傳遞的神經網路模型: 在MNIST數字辨識的應用上，利用軟體架構出突波神經網路的模型以及模型的訓練與測試，模型採用多層單向傳遞結構。硬體端利用Verilog實現、合成與模擬，也進一步驗證硬軟體模擬結果的一致性。

- 單層循環式突波網路模型: 在數獨結果預測之中，手動定義連接突觸的抑制與激活及連接突觸的強弱，並調整閾值膜電位提高預測準確率。模型採循環式圖

波神經網路的結構，並於硬體端利用Verilog實現、合成與模擬。

不僅完成了突波神經網路的硬體架構設計，也擴及至突波神經網路的訓練與測試，並如預期地呈現出突波神經網路的特性與行為模式。現今對於資料的運算量需求呈現指數成長，在追求低功耗及硬體友善上顯得格外重要，而突波神經網路旨在以低功耗與高準確率完成這項工作，目前此領域仍在開發研究階段。此實作專題證明了突波神經網路處理器具有其應用價值，期望在未來能根據此神經網路的特性延伸應用在高運算量的圖片處理或事件觸發的即時偵測上。

5. 心得

雖然我們分工明確，但我們對於每項工作都相當熟悉且能相互支援，在研究初期，我們遇到的問題是找不到有效並且方便訓練 SNN 的模型，也不懂怎麼建立高效率的硬體系統架構，而在利用網路資源多進行參考以及研究後，最後我們也能自行設計出適當的軟、硬體設計，不僅使我們學會尋找適合的論文進行研究，和撰寫 RTL code 來實現參考架構，更學會使用 PyTorch 設計演算法後利用硬體語言描述進行系統設計，也磨練出我們對 Verilog、Python 等語言的熟悉度。

專題研究期間的開會我們積極向實驗室學長請教並汲取建議，因而快速的進入狀況，儘管在這長時間的獨立研究過程中遇到不少難題與挫折，但克服這些問題後學到的也更多，著實感謝這段時光的自我淬鍊。

6. Reference

[1] Zuo-Wei Yeh, “Configurable population-based spiking neural network processor with Integer Quadratic-Integrate-and-Fire(I-QIF) Neuron”, Ch1.1, P.3.

[2] Yann LeCun, Corinna Cortes, Christopher J.C. Burges, “The MNIST Database of handwritten digits” [Online],Available: http://yann.lecun.com/exdb/mnist/

[3] Vaibhav Arcot, “Dataset of 4x4 Sudoku puzzles and solutions” [Online], Available: https://github.com/Black-Phoenix/4x4-Sudoku-Dataset.