

利用脈衝神經網路辨識 MNIST 手寫數字資料庫

Recognize MNIST Handwritten Digit Database by Spiking Neural Network

朱霽 王仲源 吳泓洵
指導老師：鄭桂忠 教授

ABSTRACT

作為機器學習的一部份，類神經網路一直扮演著重要的角色，從 Deep neural network (DNN)到模擬真實神經元運作的 Spiking neural network (SNN)都是其中的一部份，而我們專題實作的就是 SNN，SNN 作為類神經網路的一部份，與 DNN 或是 CNN 不同之處是，它引入了時間軸，也就是學習以及預測結果時，都是在一段時間中進行學習。SNN 在學習方法上一直存在著瓶頸，因此我們主要針對它的學習方法做出一些研究及結論，以期望找出適合 SNN 的訓練方法。

我們的研究包括 STDP learning rule 的實作、利用 back propagation 訓練方式的 SNN，以及部分 SNN 數位電路的搭建，在 STDP learning rule 實作中，我們利用 python 完成了 STDP 學習的模擬，在 back propagation 學習中，我們利用 PyTorch，以及經過一些修改完成這個實作，在數位電路搭建中，我們試圖利用 verilog 來描述 neuron 以及 synapse 的行為。

INTRODUCTION

我們所使用的測試資料是 MNIST 手寫 0 到 9 的數字圖片，使用的網路架構採用 fully-connected layer 的概念，總共設計成 3 層，分成 input、hidden 以及 output layer，如 FIG. 1 所示。

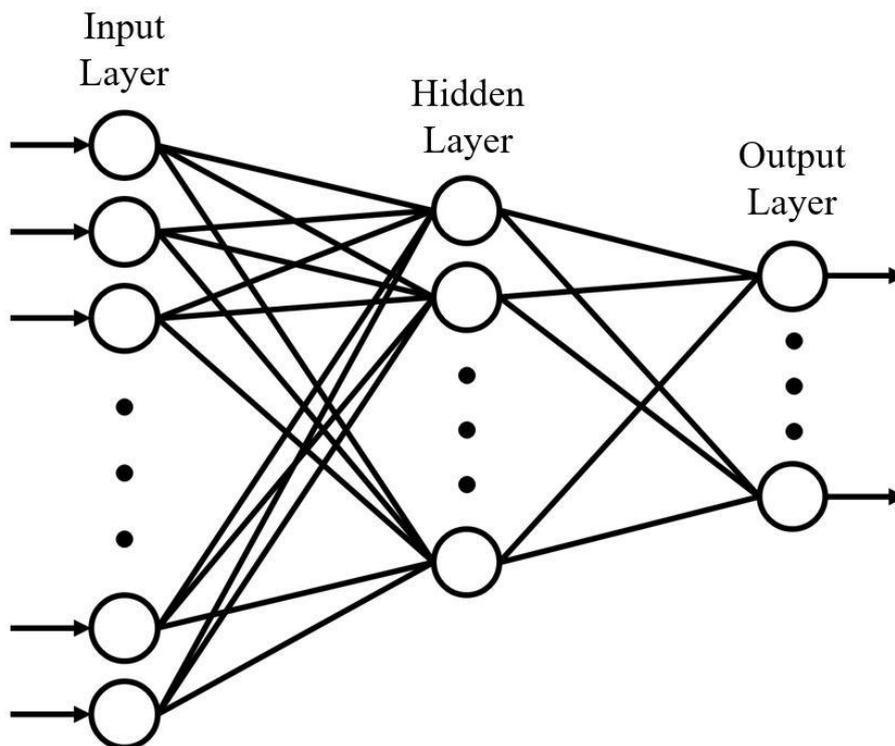


FIG. 1

1. 使用 Python 模擬 STDP 訓練 SNN 的情況

各層的 neuron 個數分別是 784、50 以及 10。我們利用 Leaky integrate-and-fire (LIF) neurons model 來模擬 neuron 的行為模式：

$$\begin{bmatrix} MemPot_1(t) \\ MemPot_2(t) \\ \vdots \\ MemPot_{50}(t) \end{bmatrix} = \begin{bmatrix} W_{1,1} & W_{1,2} & \dots & W_{1,784} \\ W_{2,1} & W_{2,2} & \dots & W_{2,784} \\ \vdots & \vdots & \ddots & \vdots \\ W_{50,1} & W_{50,2} & \dots & W_{50,784} \end{bmatrix} \begin{bmatrix} spike_1 \\ spike_2 \\ \vdots \\ spike_{784} \end{bmatrix} \\ + \begin{bmatrix} MemPot_1(t - \Delta t) \\ MemPot_2(t - \Delta t) \\ \vdots \\ MemPot_{50}(t - \Delta t) \end{bmatrix} - \begin{bmatrix} leakage \\ leakage \\ \vdots \\ leakage \end{bmatrix}$$

透過 STDP 進行 unsupervised learning，其中， $\Delta t = t_{pre} - t_{post}$ ：

$$\Delta W = \begin{cases} -A_- \cdot e^{\frac{\Delta t}{\tau_-}}, & \Delta t > 0 \\ A_+ \cdot e^{\frac{\Delta t}{\tau_+}}, & \Delta t < 0 \end{cases}$$

2. 使用 PyTorch 模擬 gradient descent 訓練 SNN 的情況

與一般設計 $spike_{out} = 1.0$ 不同，我們在 PyTorch 上實作的 LIF neuron 中 $spike_{out}$ 運算如下所示：

$$spike_{out} = membrane\ potential - threshold$$

由 **FIG. 2** 可以得到 neuron 隨時間的 spike out，橫軸為時間，縱軸為 spike out 的數值，可以觀察到 neuron spike out 的值並不是單純的 1.0，而是有不同的數值所組成。

使用 gradient descent 訓練出的 model 在 MNIST 手寫數字集正確率上取得不錯的結果。如 **Table. 1** 所示，在疊代 8 次後正確率已經可以達到 95% 的正確率。

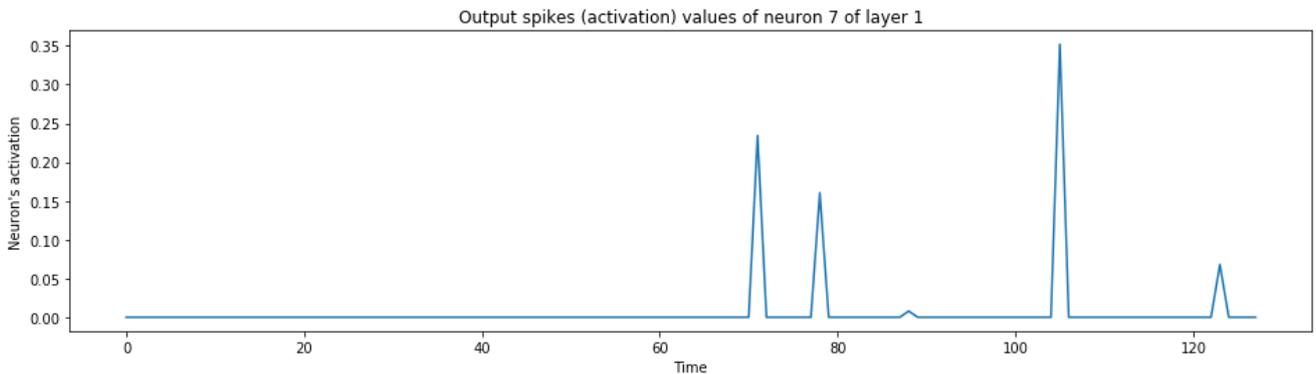


FIG. 2

epoch	accuracy(%)
1	87.84
2	90.79
3	92.23
4	93.01
5	93.72
6	94.39
7	94.55
8	95.01

Table. 1

3. 將各層神經元表現以數位電路實現

在數位電路實現的部分，我們選用 LIF neuron 在 PyTorch 訓練較成功的模型(沒有膜電位恢復期)，並將其各層的 neuron 架構以 verilog 的 simulation 呈現。這個 model 是由三層 neuron 所組成，第一層 network 的運作機制是將 MNIST 數字資料庫的圖檔灰階值轉換成 spike output 傳遞到下一層如 FIG. 3 所示。第二層 network 則是實現突觸以及 Leaky integrate-and-fire 機制，將上一層的訊號統合轉換成 spike output 傳遞到下一層如 FIG. 4 所示。第三層 network 也是實現突觸以及 Leaky integrate-and-fire 機制，並在最後將分類訊號統整在 neuron 中的 membrane potential 裡面如 FIG. 5 所示。



FIG. 3

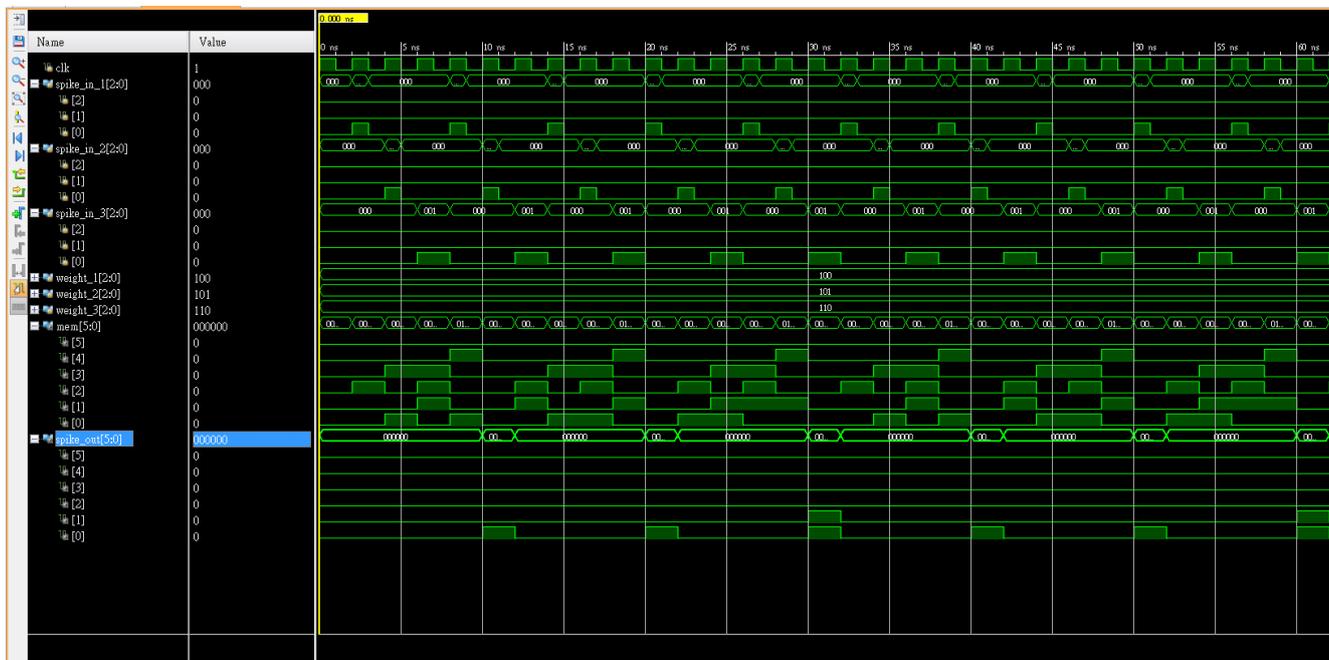


FIG. 4

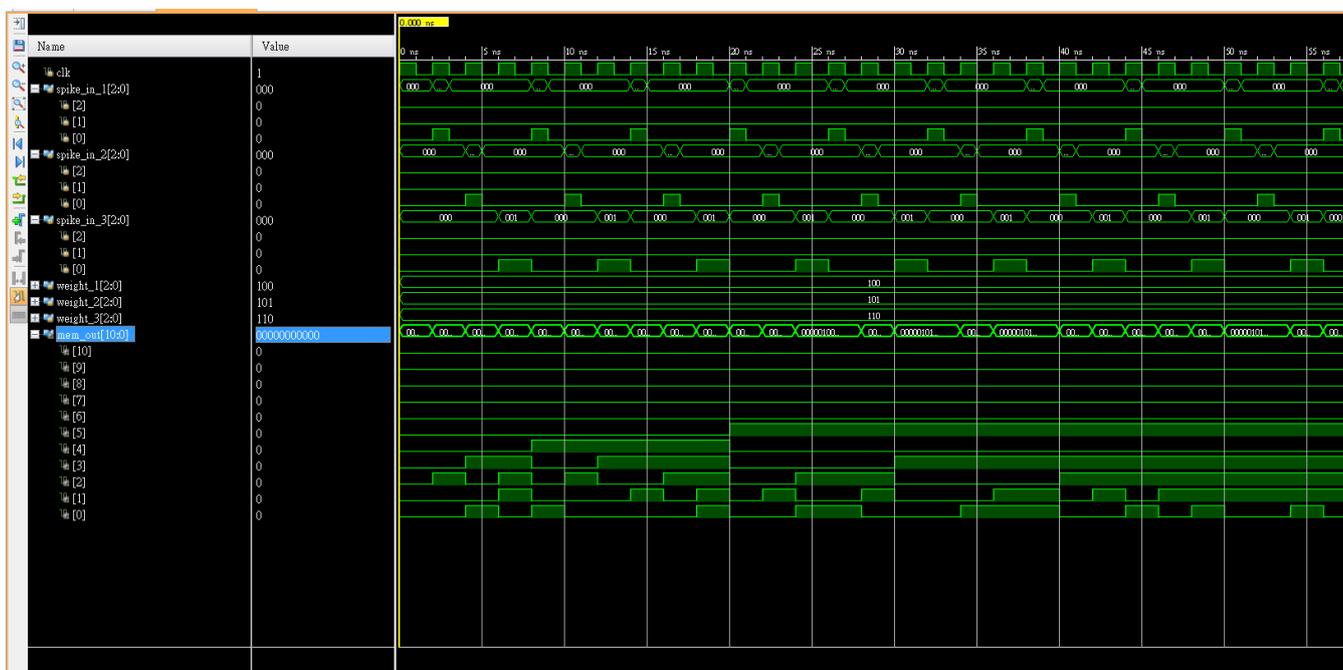


FIG. 5

心得感想

經歷這次實作專題的訓練後，不但加深了我們對於神經網路的認識，也讓我們多多少少體會到研究的本質。從補充知識上的不足，到把概念實作在軟硬體上的過程中，常常需要上網蒐集大量的資料，從中擇取有用的資料，並時常與組員、學長驗證想法是否正確。雖然我們一開始是希望使用 STDP 能夠訓練出一組正確率高的 weight，並利用它時坐在數位電路上，但最後沒有成功，轉而使用 gradient descent 這種偏向數學計算而非模擬 neuron 運作模式的演算法。學會獨立解決問題的能力，並瞭解如何團隊合作，是我們在實作專題中所學習到最重要的事情。