

國立彰化師範大學電子工程學系

碩士論文

具自我預訓練能力之深度學習錯誤函數的就地
常態化器

**Self-Pretrainable In-situ Normalizer for Deep
Learning Error Function**

研究 生：柯 約 鑑 撰

指導教授：黃 宗 柱 教授

中 華 民 國 一 一 二 年 六 月

國立彰化師範大學電子工程學系

碩士論文

具自我預訓練能力之深度學習錯誤函數的就地常態
化器

研究生：柯竣鑫

本論文業經審查及口試合格特此證明

論文考試委員主席：鄭經華博士 鄭經華

口試委員：吳宗益博士 吳宗益

黃宗柱博士 黃宗柱

指導教授：黃宗柱博士 黃宗柱

系主任：陳棟洲博士 陳棟洲

中華民國一一年六月

Self-Pretrainable In-situ Normalizer for Deep Learning Error Function

A Thesis

by

Jyun-Xin Ke

Approved as to style and content by:

Ching-Hua Cheng

Ching-Hua Cheng
(Chair of Committee)

Tsung-Chu Huang

Tsung-Chu Huang
(Advisor)

Tsung-Yi Wu

Tsung-Yi Wu
(Member of Committee)

Tsung-Chu Huang

Tsung-Chu Huang
(Member of Committee)

Tung-Chou Chen

Tung-Chou Chen
(Head of Department)

June 2023

國立彰化師範大學學位論文授權書

(本聯請隨論文繳回學校圖書館，做為授權管理用) ID:111NCUE0428004



* 1 1 1 N C U E 0 4 2 8 0 0 4 *

● 立書人(即論文作者)：柯竣鑫 (下稱本人)

● 授權標的：本人於國立彰化師範大學 (下稱學校) 電子工程學系 (研究所、學位學程) 111 學年度第二學期之碩士學位論文。

論文題目：具自我預訓練能力之深度學習錯誤函數的就地常態化器

指導教授：黃宗柱,HUANG,TSUNG-CHU

(下稱本著作，本著作並包含論文全部、摘要、目錄、圖檔、影音以及相關書面報告、技術報告或專業實務報告等，以下同) 緣依據學位授予法等相關法令，對於本著作及其電子檔，學校圖書館得依法進行保存等利用，而國家圖書館則得依法進行保存、以紙本或讀取設備於館內提供公眾閱覽等利用。此外，為促進學術研究及傳播，本人在此並進一步同意授權學校、國家圖書館等對本著作進行以下各點所定之利用：

對於學校、國家圖書館之授權部分：

本人同意授權學校、國家圖書館，無償、不限期間與次數重製本著作並得為教育、科學及研究等非營利用途之利用，其包括得將本著作之電子檔收錄於數位資料庫，並透過自有或委託代管之伺服器、網路系統或網際網路向校內、外位於全球之使用者公開傳輸，以供該使用者為非營利目的之檢索、閱覽、下載及/或列印。

校內外立即開放

校內立即開放，校外於 年 月 日後開放

校內於 年 月 日；校外於 年 月 日後開放

其他或不同意

註：

(一) 本授權書所定授權，均為非專屬且非獨家授權之約定，本人仍得自行或授權任何第三人利用本著作。

(二) 本人擔保本著作為本人創作而無侵害他人著作權或其他權利。如有違反，本人願意自行承擔一切法律責任。

(三) 本授權書授權對象，應遵守其授權範圍及相關約定。如有違反，由該違反之行為人自行承擔一切法律責任。

立授權書人：柯竣鑫 (正楷親簽) 日期：112 年 6 月 7 日

中文摘要

隨著科技的發展，神經網路已廣泛應用在生活之中。為了提升模型的辨識能力，神經網路的層數也逐漸增加。然而，這樣的深層網路也增加了成本的負擔。近年來，遷移式學習在機器學習領域中開始受到廣泛關注，特別是在骨幹網路的應用上，能夠大幅減少訓練所需的成本。而在神經網路中，激勵函數的選用與學習目標的資料分布特性有著強大關聯性，往往會影響到訓練後模型的準確度，一旦輸入資料為非常態化特徵分布，將不具有相對應的激勵函數，對於模型的準確度會有一定的限制。

激勵函數的種類很多，常見的有 ReLU、Sigmoid 等。以骨幹網路為例，多檢測頭的 Transformer 系列模型則會選用 GeLU 或 softmax 作為激勵函數，而單一檢測頭的 YOLO 系列模型通常選用 Leaky ReLU、Mish 等激勵函數。根據我們的研究，為了加速神經網路的計算，會選用查表法(LUT)或是結合分段線性函式(PWL)所提出的範圍可定址查表法(RALUT)來取代激勵函數。

基於這樣的想法，我們對分段線性搜尋法進行改善，解決現有分段線性搜尋法上，無法有效區分目標斜率轉折處的問題，並增加能夠調整形狀的功能，稱作即線預訓練常態化器。另一方面，為了減少及線調整容易遇到的收斂問題，另外發展離線預取樣常態化器，仍保持就地概念，可在現有已開發骨幹網路上做結合。在神經網路實驗中，以非常態分布資料集作為輸入，我們設計的常態化器皆能提高 AlexNet、ResNet、VGG-16 骨幹模型的準確度，並在 VGG-16 上提高至 71.92%。在辨識自定義複雜化的 PolyMNIST 手寫資料集上，具有 98.85% 準確度的辨識能力，高於所有現有的激勵函數。本論文提出的兩種常態化器，皆能有效調整激勵函數的形狀，稱為就地常態化器(SPINDLE)。

關鍵字—激勵函數、骨幹神經網路、常態化器

Abstract

With the advancement of technology, neural networks have been widely applied in various aspects of life. However, deep networks come with increased cost burdens. In recent years, people have been using backbone neural networks to reduce costs. The choice of activation function in neural networks is strongly correlated with the distribution characteristics of the learning data and can affect the accuracy of the trained model. Non-normalized feature distributions may limit accuracy if there is no corresponding activation function.

Common activation functions include ReLU, Sigmoid, and others. For example, in backbone networks, multi-head Transformer models use GeLU or softmax as the activation function, while single-head YOLO models often choose Leaky ReLU, Mish, and others. To accelerate neural network computations, lookup tables (LUT) or range addressable lookup tables (RALUT) combined with piecewise linear function(PWL) have been proposed to replace activation functions.

Based on this idea, we have improved the PWL search method, addressing the problem of ineffective differentiation at slope inflection points and adding shape-adjustment functionality, called Online Pre-training Normalizer. Additionally, an Offline Pre-sampling Normalizer has been developed to reduce convergence issues in online adjustment. In neural network experiments using non-normal distribution datasets, ours improves the accuracy of backbone models like AlexNet, ResNet, and VGG-16, achieving an accuracy of 71.92% on VGG-16 model. It also outperforms existing activation functions with 98.85% accuracy on a custom complex PolyMNIST. Both normalizers effectively adjust the shape of activation functions and are collectively known as Self-Pretrainable In-situ Normalizer for Deep Learning Error Function(SPINDLE).

Keywords : Activation function, Backbone neural network, Normalizer

誌 謝

首先，在此感謝我的指導教授與口試委員的指導，深感榮幸能夠在寶貴的建議下完成本人的學術研究。

在我就讀期間，由衷感謝我的指導教授，黃宗柱教授，願意耐心聆聽我的想法，幫助我確立研究方向，並協助我前往台灣半導體中心進修相關晶片設計課程。同時，我也要感謝教授您將管理實驗室工作站的責任交付予我，並願意相信我能勝任此項任務。在建立與維護工作站的過程中，我獲得了許多寶貴的經驗和知識，尤其是在解決問題的能力上。我學會了如何從多個角度來看待問題，分析問題的本質，並尋找有效的解決方式。這些寶貴的經驗對我未來的學術和職業生涯具有重要價值，我現在能更加自信地面對並解決各種困難和挑戰，在未來的工作和研究中應對不同的問題及情境。

此外，感謝實驗室的亭羽學姐，願意分享自己的研究並帶領我探討研究主題，也感謝濯陞學長的指教，讓我能更熟練管理工作站。也謝謝冠于與婉如同學協助解決實驗室的事務。謝謝學弟妹們，相信在你們辛勤的努力下，很快便能承擔起實驗室的所有事務。最後感謝家人、朋友以及所有曾經幫助過我的人，謝謝你們。

柯竣鑫謹誌於
國立彰化師範大學電子工程學系(所)

中華民國 112 年 6 月

目 錄

	頁次
中文摘要.....	i
Abstract.....	ii
誌 謝.....	iii
目 錄.....	iv
圖目錄.....	vi
表目錄.....	viii
第一章 緒論.....	1
1.1 研究背景與動機	1
1.2 神經網路的種類	2
1.2-1 類神經網路	2
1.2-2 卷積神經網路	3
1.3 骨幹網路的介紹	6
1.4 基本的激勵函數	7
1.4-1 線性整流函數	7
1.4-2 雙曲正切函數	8
1.4-3 邏輯函數	9
1.5 論文架構編排	9
第二章 文獻探討.....	10
2.1 探討骨幹網路為基底的激勵函數	11
2.1-1 多頭神經網路	12
2.1-2 單頭神經網路	14
2.2 探討非骨幹網路為基底的激勵函數	16
2.2-1 以 LUT 作為激勵函數	16
2.2-2 以 RALUT 作為激勵函數.....	17
2.2-3 以 PWL 作為激勵函數.....	17
第三章 即線預訓練常態化器.....	19
3.1 近似激勵函數	19
3.2 四種建構單元	20

3.3 輕斜率設計以及 SPINDLE 架構.....	21
3.4 二分輕數斜率分段線性搜尋法	25
第四章 離線預取樣常態化器.....	29
4.1 預取樣排序法	29
4.2 逆轉換取樣法	31
4.3 特徵提取器	33
4.4 搜尋非常態特徵分布	35
4.5 核密度估計 SPINDLE 設計	36
第五章 實驗.....	38
5.1 線性搜尋法改進與結果比較	39
5.2 即線預訓練常態化器實現	45
5.3 兩種取樣法量化效果	50
5.4 比較各類骨幹神經網路	54
5.5 比較各類激勵函數	59
第六章 結論與未來展望.....	65
6.1 結論	65
6.2 未來展望	65
參考文獻.....	66
作者簡歷.....	69

圖 目 錄

頁 次

圖 1、(a)常態化特徵分布 (b)雙峰特徵分布	1
圖 2、人工神經網路	2
圖 3、神經網路基本單元	3
圖 4、卷積神經網路	3
圖 5、遞迴神經網路	5
圖 6、骨幹神經網路	6
圖 7、線性整流函數	8
圖 8、雙曲正切函數	8
圖 9、邏輯函數	9
圖 10、文獻探討架構圖	10
圖 11、Vision Transformer[17].....	11
圖 12、GELU 函數	13
圖 13、LeakyReLU 函數.....	14
圖 14、Mish 函數	15
圖 15、Look-up Table 架構圖	16
圖 16、Range Address Look-up Table 架構圖	17
圖 17、雙曲正切函數分段示意圖	17
圖 18、分段線性函數電路示意圖	18
圖 19、(a)錯誤函數 (b)非常態分布錯誤函數	19
圖 20、四種 ReLU 函數圖形.....	21
圖 21、分段線性函數	22
圖 22、以 2 的 m 次方作為斜率	23
圖 23、SPINDLE 結構圖	24
圖 24、CDF 與 PDF 關係圖	25
圖 25、一次微分端點	26
圖 26、二次微分反曲點	27
圖 27、BLS-PWL 搜尋法	28
圖 28、負斜率調整 PWL	28
圖 29、預取樣排序法流程圖	29
圖 30、使用取樣排序法量化學生成績.....	30
圖 31、逆轉換取樣法流程圖	31
圖 32、逆轉換取樣法原理圖	32
圖 33、骨幹網路的特徵提取器	33
圖 34、神經網路連結就地常態化器	34

圖 35、提取特徵分布流程圖	34
圖 36、核密度估計	36
圖 37、KDE-SPINDLE 架構圖	37
圖 38、BLS-PWL 演算法	39
圖 39、(a)一次微分圖形 (b)二次微分圖形	40
圖 40、(a)設立邊界值 (b)設立初始點	40
圖 41、logistic 函數斜率搜尋	41
圖 42、二分輕數斜率分段線性之數據	42
圖 43、二分輕數斜率分段線性之圖形	42
圖 44、(a)LS-PWL[25]雙峰曲線 (b)BLS-PWL 雙峰曲線	43
圖 45、系統方塊圖	45
圖 46、時序圖	46
圖 47、四種不同的激勵函數圖形	46
圖 48、四種激勵函數波形圖	47
圖 49、SPINDLE 電路圖	48
圖 50、SPINDLE 晶片布局圖	48
圖 51、(a) DRC 驗證 (b) LVS 驗證	49
圖 52、(a)雙峰分布資料集 (b)雙峰分布資料集 PDF 和 CDF 函數	50
圖 53、雙峰分布資料集軸需圖	51
圖 54、預取樣排序法軸需圖	51
圖 55、預取樣排序法後產生的量化 CDF	52
圖 56、逆轉換取樣生成隨機子資料集之 PDF 圖形	52
圖 57、取樣與原始 CDF 的誤差值疊圖	53
圖 58、逆轉換取樣法軸需圖	53
圖 59、nirscene1[2]資料集	54
圖 60、VGG-16 特徵提取圖	55
圖 61、各種骨幹網路比較實驗流程圖	56
圖 62、VGG-16 模型準確度	58
圖 63、自定義 PolyMNIST 資料集	59
圖 64、M0-MNIST 特徵提取圖	60
圖 65、各種激勵函數比較實驗流程圖	62
圖 66、各種激勵函數不準確度長條圖	64

表目錄

	頁次
表 1、圖像分類器應用於 ImageNet 資料集[5]	4
表 2、BLS-PWL 搜尋法搜尋結果	43
表 3、LS-PWL[25] 搜尋法搜尋結果	43
表 4、兩種演算法比較表	44
表 5、整體電路之布局數據	49
表 6、各種骨幹網路模型的特徵提取參數.....	55
表 7、各種骨幹網路模型準確度比較表.....	57
表 8、各種 PolyMNIST 資料集的特徵提取參數.....	60
表 9、各種 PolyMNIST 資料集參數.....	61
表 10、各種激勵函數準確度比較表.....	63
表 11、各種激勵函數名次表.....	64

第一章 緒論

1.1 研究背景與動機

隨著科技的發展，人工智慧已然成為人們生活中不可或缺的一部分。然而想要達到高理解力的神經網路模型，如現今流行的聊天機器人 ChatGPT，必須先進行大量資料的收集和大型神經網路模型的開發，並經過長時間的訓練才能完善人工智慧的建立。

隨著蒐集資料量的提升，若是資料集的收集來源不同，將導致分布不再是常態化。一般而言，我們可以根據大數法則(Law of large numbers)，推斷世界萬物在資料特徵分布上會是趨近常態化分布，如圖 1 (a)所示，但這是建立在相同的感測器收集而成，以影像辨識領域為例，若資料集是由兩種不同來源所建構，如圖 1 (b)所示，分別由一般相機與近紅外線相機(near-infrared camera, NIR Camera)[1]所建構成，其輸入資料的特徵分布會呈現出雙峰分布(bimodal distribution)[2]的特性，也就是輸入資料特徵疏密不均勻，此時如果仍未根據輸入特徵對模型進行相對應的調整，將導致模型難以準確解析資料特徵，對於神經網路的準確度會有所限制。

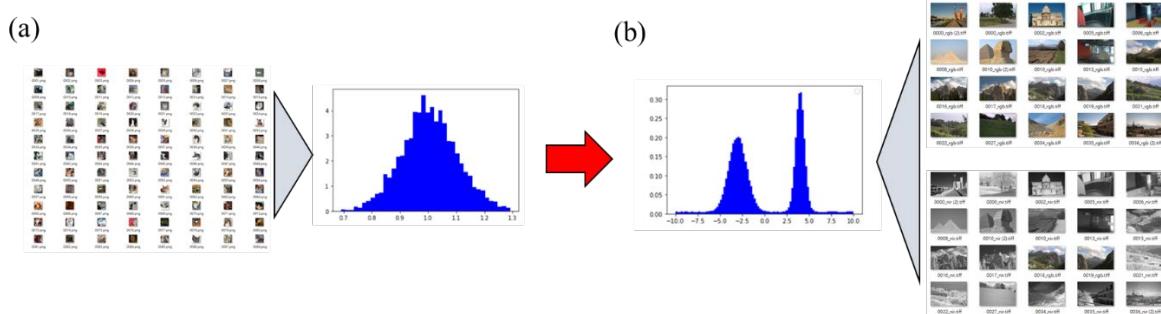


圖 1、(a)常態化特徵分布 (b)雙峰特徵分布

根據這個議題，我們提出了一種改善方式，利用預處理分析輸入的特徵分布，從而設計出能夠任意調整的激勵函數常態化器，並根據其特殊形狀，稱為 SPINDLE。

1.2 神經網路的種類

人工智能的原理是機器在經過深度學習，將學習資料經由多層次階層(layer)中的線性或非線性轉換(linear or non-linear transform)，提取各種類的特徵(feature)，進而達到能夠自我認知，並分類學習資料。其中，多層次階層統稱為神經網路模型(Neural Networks Model)，常見的神經網路有以下幾種，必須根據學習目標(define learning target)來挑選合適的神經網路模型。

1.2-1 類神經網路

類神經網路(Artificial Neural Networks, ANN)[3]的起源是由 Frank Rosenblatt 於 1957 年模仿人類大腦的生物結構，創造出人造神經元概念的感知器模型(perceptron)，又稱作類神經網路。在類神經網路中，通常是由多個階層(layer)組成，主要區分為輸入層(input layer)、隱藏層(hidden layer)、輸出層(output layer)，如圖 2 所示。而在單一階層中，是由多個神經元(neuron)組成。

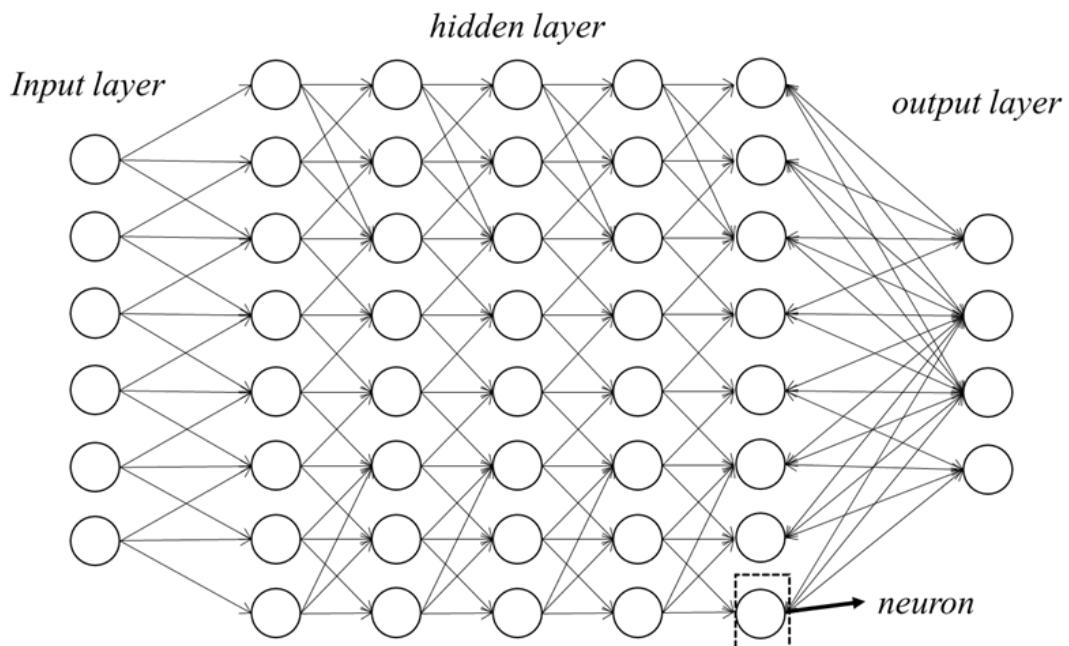


圖 2、人工神經網路

而單一神經元會對上層神經元的輸入進行加總，並進行激勵函數(Activation function)的轉換，激勵函數可以想像成人類對學習資料的判斷，用以區分資料的重要性。換言之，激勵函數就是用來決定神經元的活躍性。而轉換後會透過權重計算(weight)再交給下一層神經元當作輸入，如圖 3 所示。當神經網路層數和神經元數量龐大時，常稱其為深度神經網路(Deep Neural Networks, DNN)。

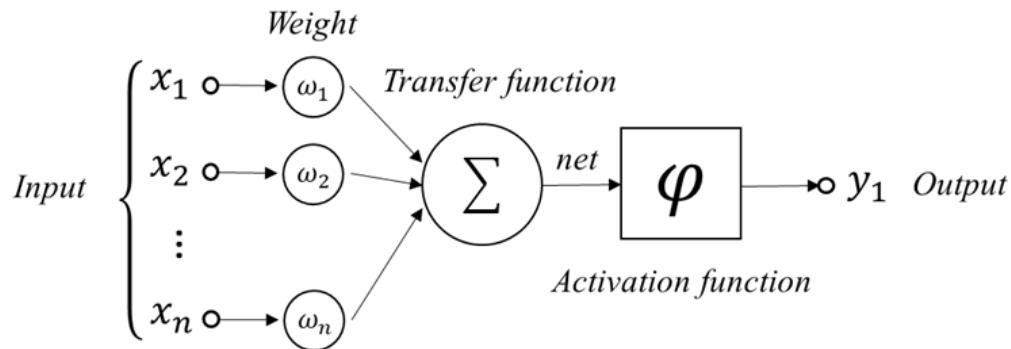


圖 3、神經網路基本單元

1.2-2 卷積神經網路

卷積神經網路(Convolutional Neural Networks, CNN)[4]通常應用於視覺影像處理(computer vision, CV)，其原理是將圖像的像素(pixel)轉換為二維矩陣，作為神經網路的輸入。卷積神經網路是在深度神經網路的基礎上，添加卷積層(Convolution Layer)和池化層(Pooling layer)，簡易原理圖如圖 4 所示。

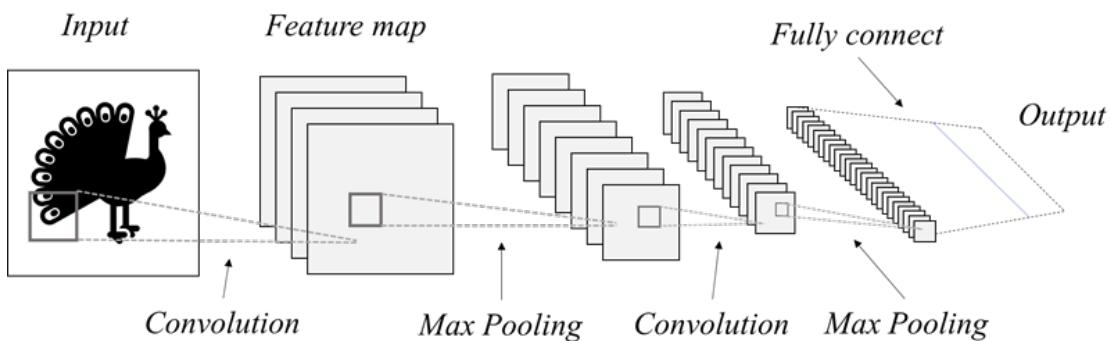


圖 4、卷積神經網路

卷積層是由 $n \times n$ 的過濾器(filter)進行強化不同強度的特徵圖(feature map)，進而增強機器學習的能力。池化層目的是進行圖片壓縮，好的設計更能降低圖像雜訊(image noise)，保留圖像重要的特徵。由卷積層和池化層建構的多層網路模型，常稱作特徵提取器(feature extraction)。提取特徵後，再由全連接層(Fully connect layer)建構的深度神經網路(DNN)進行特徵分類，因此全連接層部分也被稱作分類器(Classifier)。

表 1、圖像分類器應用於 ImageNet 資料集[5]

Name	Year	Number of params	Top 1 Accuracy
AlexNet	2012	60M	63.3%
VGG-16	2014	138M	74.4%
ResNet-101	2016	40M	78.25%
PNASNet-5	2018	88.9M	83.9%
FixResNeXt-101 32x48d	2019	829M	86.4%
EfficientNet-L2	2020	480M	88.5%
ViT-G/14	2022	1843M	90.54%
Coca	2022	2100M	91.0%

由於資訊膨脹，輸入資料集需要分類的種類也增加，若是設計的卷積神經網路層數過少，會導致準確率無法提升。表 1 是根據圖像分類的應用，蒐集 ImageNet 最先進技術(State-of-the-Art, SOTA)的歷年資料[5]，隨著年份的增加，網路的架構也逐漸走向複雜且龐大化，在訓練成本上勢必也會上升，訓練時間動輒以數年起跳。因此，如何減少訓練成本也成為研究者討論的議題。

其中，骨幹網路(Backbone network)的發展成為了新的議題，也是本論文的主要探討方向，詳細內容會在後續進行說明。

1.2-3 遲迴神經網路

一般而言，神經網路的輸入只與當前輸入有關，並不會因先前的輸入影響當前的決策，若是想開發語音辨識的人工智慧，神經網路就必須設計能夠理解前後文的關聯性，使用一般的神經網路就無法達成此學習目標。

為了改善這個問題，人們開發了一項新型態神經網路。遲迴神經網路(Recurrent Neural Network, RNN)[6]是一種與時間序列有關連性的神經網路模型，能夠記憶先前輸入產生的狀態，將資料儲存在暫存的記憶空間(internal memory)內，進而預測未來發展方向，廣泛應用於語音辨識、路線軌跡預測、自然語言處理等領域。

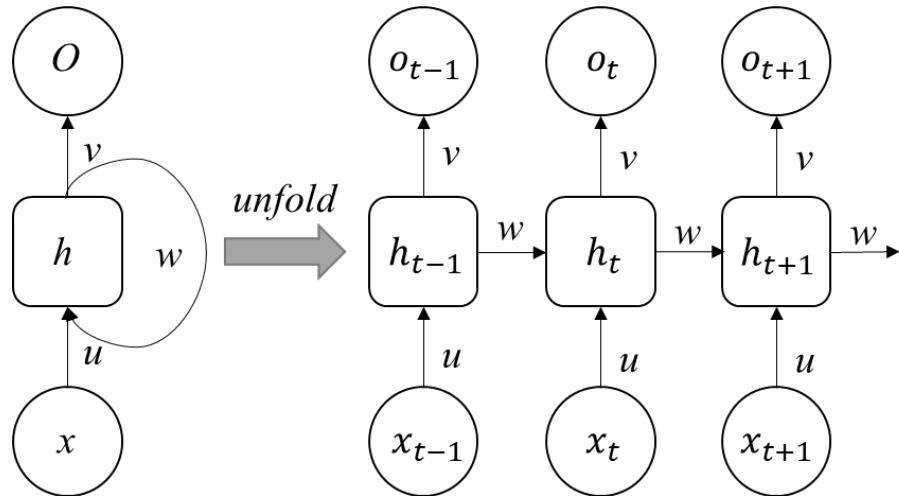


圖 5、遲迴神經網路

如圖 5 所示，遲迴神經網路在設計上與類神經網路相近，最大差異點是在於中間隱藏層具有遲迴的特性，遲迴神經元在依時間 t 中，同時具有當前時序接收的輸入 x_t ，以及上一個時序產生的 O_{t-1} ，因此神經元會具有兩個權重，稱作權重共享(Shared Weights)。透過這樣的方式讓原先毫無關聯的輸入資料產生遲迴關聯性，因此被稱作遲迴神經網路。

1.3 骨幹網路的介紹

近年來，隨著人工智慧的技術逐漸成熟，在影像辨識(image recognition)的領域上，物件偵測(object detection)的技術也格外受到關注，其廣泛應用在如：自動駕駛、監控系統、醫學影像等方面。

但隨著物件辨識種類增加，人們開始注意到，若想有效提升辨識的準確率，神經網路的層數必須提高，但相對的，其中所消耗的學習時間成本會嚴重影響開發的效率，因此提出了遷移學習(Transfer Learning)的想法，其應用稱作骨幹網路(Backbone Network)。

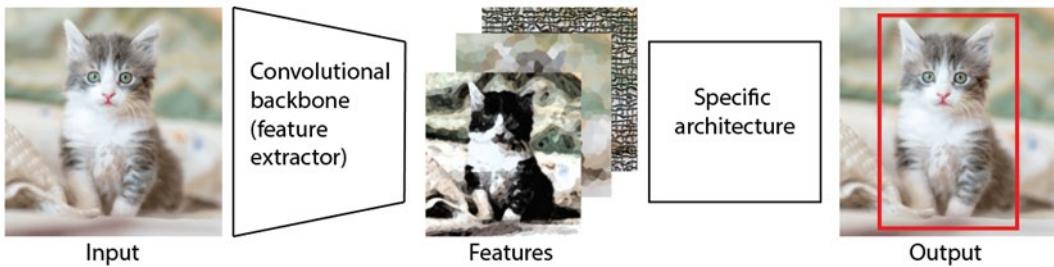


圖 6、骨幹神經網路

在卷積神經網路中，骨幹神經網路是指構成整個模型的核心部分，它通常由多個卷積層和池化層構成，用於提取圖像特徵，此部分會預先在大型數據集，如 ImageNet[7]、COCO[8]等，完成預先訓練，減少神經網路重複訓練提取圖像特徵的部分，研究者主要的工作是改善後續物件檢測層的識別準確度及效率，從而達到減少神經網路開發成本的功效。

骨幹神經網路的設計有很多種，其中比較著名的包括：

- (a) LeNet[9]：最早的卷積神經網路之一，主要用於手寫數字(MNIST)識別。
- (b) AlexNet[10]：提出於 2012 年，由 Alex Krizhevsky 團隊使用多個卷積層和池化層設計的卷積神經網路模型，在大規模圖像辨識上具有很好的效果。
- (c) ResNet[11]：是由 Microsoft Research Asia 團隊設計，目的是解決卷積神經網路訓練時的梯度消失問題，被廣泛應用於圖像識別和物件檢測任務中。
- (d) VGG[12]：是由 Visual Geometry Group 設計的卷積神經網路，其具有深度、小卷積核等特點，被廣泛應用於圖像識別、物件檢測等任務中。

在骨幹網路的應用上，我們注意到了研究者通常只調整分類器的權重，而激勵函數的選用往往都是使用常見的種類，並不會根據輸入資料的不同，進而調整激勵函數的形狀。若輸入資料特徵差異過大，若未做適當的常態化，這將嚴重限制神經網路訓練準確性的提高。

1.4 基本的激勵函數

而在神經網路領域中，激勵函數是使神經元具有非線性的特徵，一般常見的激勵函數有以下幾種，會根據資料分布的疏密性(Density)作激勵函數的選擇。

1.4-1 線性整流函數

在輸入資料分布較為均勻且稀疏，通常使用線性整流函數(Rectified Linear Unit, ReLU)[13]作為激勵函數。其定義為，當神經元輸入大於 0 時，輸入等於輸出，反之則等於 0，因此在計算上簡單且方便，是目前最受歡迎的神經網路之一，如圖 7 所示。

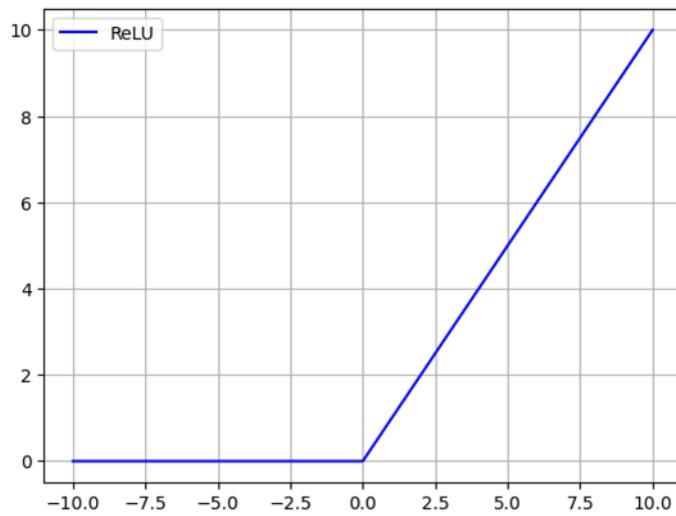


圖 7、線性整流函數

1.4-2 雙曲正切函數

雙曲正切函數(hyperbolic tangent function)[14]是雙曲函數的一種，與 ReLU 相比，輸入為負值並不會完全捨棄，因此在負值部分仍可訓練，而與其他激勵函數相比，由於輸出均值是 0，使得其收斂速度較快，可有效減少迭代次數。

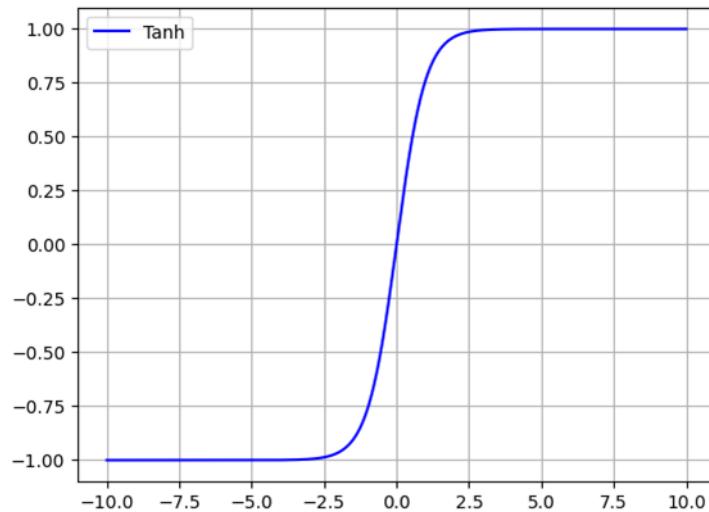


圖 8、雙曲正切函數

1.4-3 邏輯函數

邏輯函數(logistic function)又稱作 Sigmoid 函數[15]，輸出範圍介於 0 到 1 之間，對神經元的輸出進行常態化(Normalization)處理。在學習目標為二分類時，具有良好的表現。

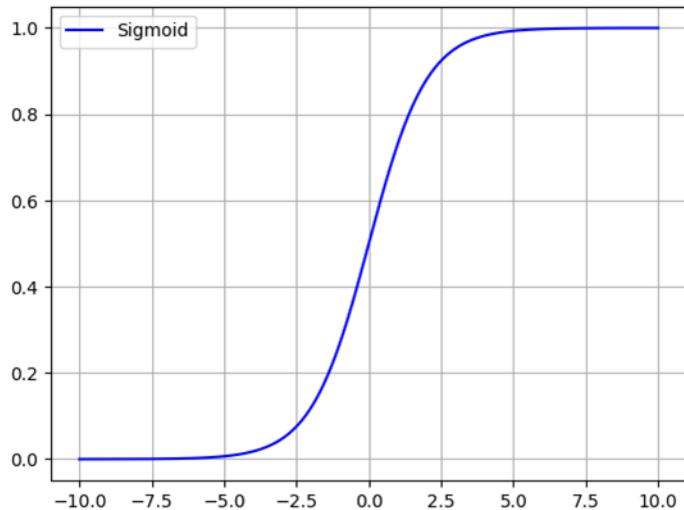


圖 9、邏輯函數

1.5 論文架構編排

本論文主要研究對非常態特徵分布進行改善，設計就地預處理常態化器來調整神經網路的激勵函數，並實踐在視覺辨識領域的骨幹神經網路上。因此，在第二章主要會以骨幹網路為分類依據，分別介紹各模型選用的激勵函數或近似激勵函數的方式。

第三章提出即線預訓練常態化器，在即線訓練上可以有效調整激勵函數形狀，改善先前分段線性演算法在分段技術上的缺陷。除了即線調整外，也可以藉由預取樣資料集特徵分布，達到解析資料特徵分布的效果，因此於第四章提出離線預取樣常態化器。

最後，在第五章呈現相關實驗數據與結果，呈現在分段技術上的改進與應用於骨幹網路上，準確度的提升，並在第六章說明本論文提出的創新點以及貢獻。

第二章 文獻探討

隨著神經網路的蓬勃發展，人工智慧的應用充斥在我們的生活之中，一個好的神經網路是由多方面的研究所組成，其中激勵函數的重要性更是決定訓練成效的關鍵因素所在。在視覺影像處理(computer vision, CV)上，人們通常使用卷積神經網路作為應用，而卷積神經網路可以根據網路層數的大小分成兩類，若是想使用高精度、多類別的神經網路，通常是使用骨幹神經網路作為卷積神經網路的架構。相反的，若是要求以辨識速度為優先目的，則會使用硬體電路進行設計。

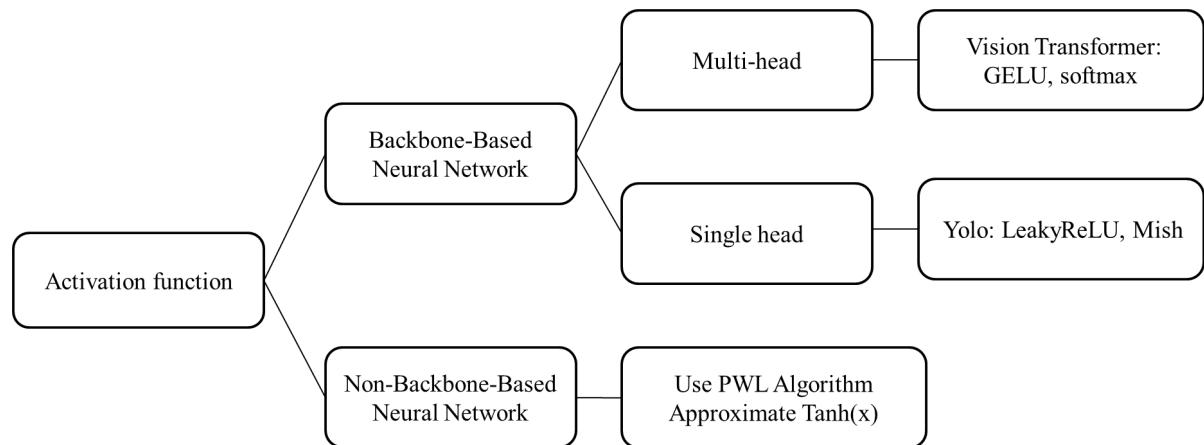


圖 10、文獻探討架構圖

圖 10 是本章節架構圖，以激勵函數做為探討核心，可以區分為以骨幹網路為主軸的激勵函數類型，以及非骨幹網路架構的最佳激勵函數選擇。

本論文的定位是基於非骨幹網路架構的最佳激勵函數做研究，目的是將作加激勵函數應用到大型神經網路上，達成隨著輸入特徵的變化，進而調整激勵函數的形狀，藉此提升神經網路訓練的準確度。

2.1 探討骨幹網路為基底的激勵函數

在視覺影像處理中，使用大型骨幹網路的卷積神經網路可以區分成兩種類型。

受到 Transformer[16]在自然語言處理(Natural Language Processing, NLP)方面的成功啟發，使用了多頭注意力機制(Multi-Head Attention, MHA)方式應用在影像方面，著名的神經網路類型為：Vision Transformer (ViT)[17]、Swin-ViT[18]等。

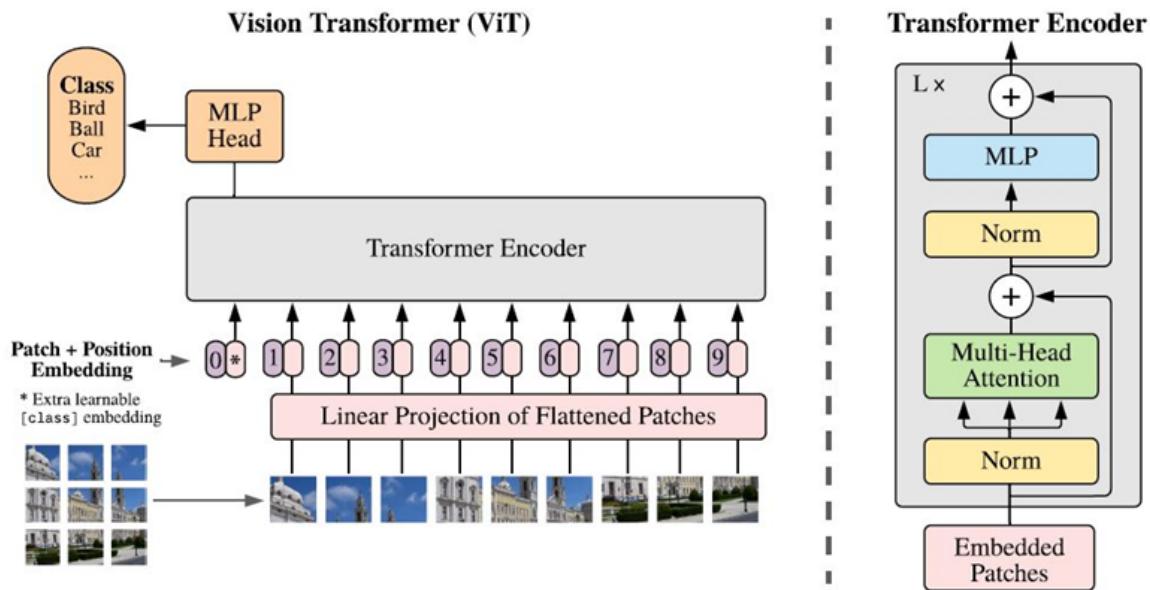


圖 11、Vision Transformer[17]

另一種類型則是由傳統卷積網路所演化，由於特徵提取層走向複雜化，因而提出遷移學習(Transfer Learning)的方式，常見的架構是由骨幹(Backbone)、頸部(Neck)、監測頭(Head)組成，此類型神經網路常應用在物件辨識領域上，具有單頭(Single-Head)的特性，因此在物件辨識領域又稱作一階段物件偵測(one-stage detector)，著名的神經網路為:YOLO[19]、SSD[20]等。

2.1-1 多頭神經網路

在骨幹網路的領域中，多頭神經網路多用於 Transformer 上，其中基於此模型中的參數數量相對於傳統卷積神經網路較為龐大。以 ViT-Base[17]為例，其模型使用參數為 86M (Million)單位，若是使用 ViT-Huge[17]，則需要 632M 的參數。再加上每個參數的計算都需經過激勵函數，可見激勵函數在神經網路的重要性。而常見選用的激勵函數類型，除了基於矩陣乘法的全連接層外，還包括幾種類型的激勵函數。

(a) Softmax 函數:

為 ViT 模型中最常見的激勵函數，其原理是將一個向量轉換為一個機率分布，在所有類別中，機率總和為 1，其中當某類別機率愈大，代表該物件是該類別的機率也愈大，如方程式(2.1)所示。

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad for j = 1, \dots, K. \quad (2.1)$$

Softmax 函數將輸出向量中的每個元素作為指數，對所有指數求和，最後將每個指數除以總和，得到每個元素對應類別的機率，從而分辨該學習目標機率最大為某個對應類別。

而在 ViT 模型中，Softmax 函數通常被用於對解碼器(Encoder)的輸出進行分類，在最後一層解碼器上，會先連接至全連接層。在進行預測之前，全連接層的最終輸出會通過 Softmax 函數進行常態化處理，最終根據機率大小進行分類。

(b) GELU 函數[21]:

由 Hendrycks 和 Gimpel 在 2016 年提出，其特點是它結合了線性和非線性的要素，是基於高斯分布的累積分布函數的近似，如所示，也就是假設隱藏層神經元的輸入屬於高斯分布的狀態，如公式(2.2)所示。

$$GELU(x) = xP(X \leq x) = x\phi(x) = 0.5x \left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right) \right] \quad (2.2)$$

GELU 函數的主要優點在於可以加速模型的收斂速度和提高模型的精度。與其他常用的激勵函數相比，GELU 函數具有更高的非線性能力，從而使得模型可以更好地捕捉複雜的特徵。除此之外，GELU 函數具有解決梯度消失(Gradient Vanishing)的能力，從而使得模型可以更好地訓練。但是 GELU 也具有一定的缺點，在計算上，計算複雜度較高，需要計算誤差函數，會大幅增加計算量，造成模型參數量增加，且 GELU 函數對溢位輸入(overflow input)的敏感度較高，容易造成模型的不穩定性。

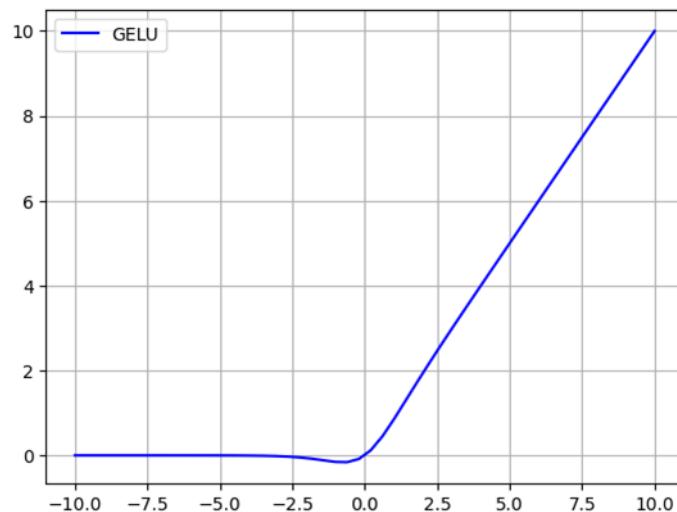


圖 12、GELU 函數

2.1-2 單頭神經網路

在骨幹網路的領域中，單頭(Single-Head)神經網路較為多元，其起源大多來自原先著名的卷積神經網路，近年來，隨著物件辨識和偵測的發展，卷積神經網路朝向大型化發展，此類型最為著名的即為 Yolo[19]系列的相關論文，除了基本的激勵函數外，以下挑選幾種單頭神經網路所用的激勵函數類型。

(a) LeakyReLU 函數:

LeakyReLU 屬於 ReLU 激勵函數的變種，為了修正 ReLU 在負值區域為 0，而造成神經元死亡的問題，通常在設計上會在負值區域給於一定斜率，從而提高模型的訓練效果。與其他激勵函數相比，具有計算數度快，訓練效果好等優點。

但由於並沒有固定的傾斜度值，會導致無法為正負輸入值提供一致的關係預測，也就是產生不一致性，如果設置的傾斜度值不當，可能會導致模型性能下降。

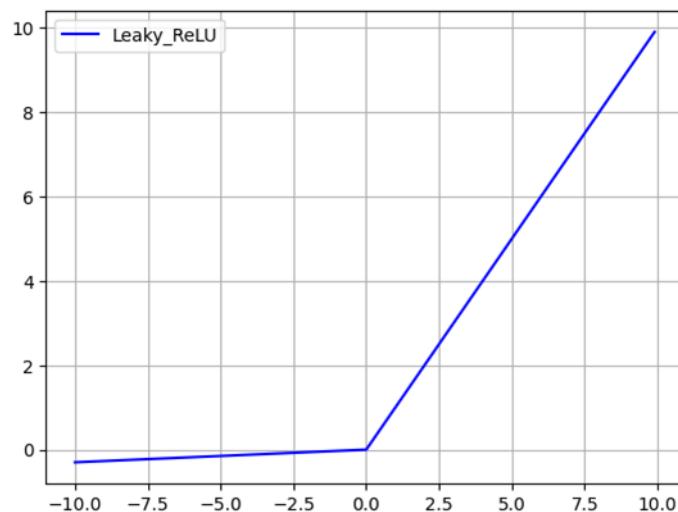


圖 13、LeakyReLU 函數

(b) Mish 函數：

Mish 函數是一種近期提出的激勵函數，它是由 S. Misra 在 2019 年提出的，具備無上界有下界、平滑、非單調的特性。Mish 函數的定義如式子(2.3)所示：

$$Mish(x) = x * \tanh(\ln(1 + e^x)) \quad (2.3)$$

Mish 函數具有平穩的梯度，在訓練上可以少梯度消失和爆炸的問題。其主要缺點在於計算成本較高，Mish 函數中包含複雜的計算，因此在計算上較其他激勵函數慢。

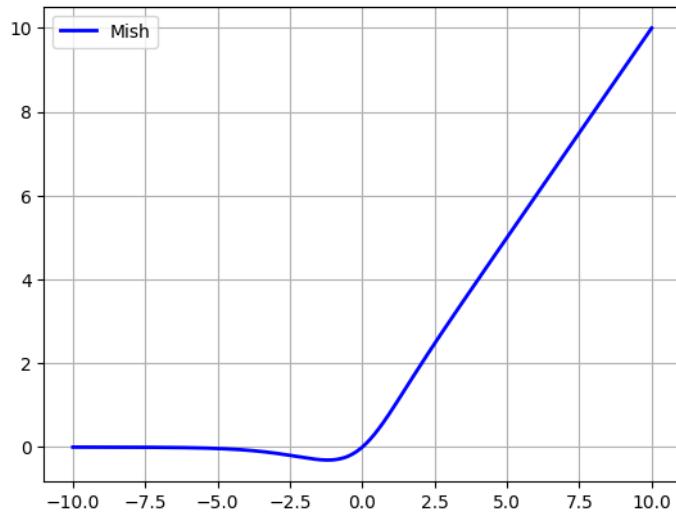


圖 14、Mish 函數

綜上所述，LeakyReLU 函數在面對不同的資料輸入，調教能力較差，僅能以單一斜率調整作為修正激勵函數形狀。而 Mish 函數雖然在神經網路訓練上有很好的性能表現，但訓練成本過高，不適合大量使用在骨幹神經網路中。

因此本論文提出新的激勵函數，使用即線微調及離線取樣的方式，設計出可任意調整激勵函數，能夠有效解決神經網路在激勵函數形狀無法改變的問題，且因建構單元為 ReLU，在反向傳播上不需經過複雜計算，能有效降低計算成本。

2.2 探討非骨幹網路為基底的激勵函數

若神經網路的應用是以消費性電子產品為前提，而非大型骨幹卷積神經網路，則必須保有高速計算且具有低功耗，通常不會選擇使用查表法(Look-up Table, LUT)[22]或是範圍可定址查表法(Range Addressable Look-Up Table, RALUT)[22]來做激勵函數計算，此部分可以利用硬體電路來達到加速計算的目的。

2.2-1 以 LUT 作為激勵函數

在傳統查表法(Look-up table, LUT) 的使用上，每個輸入都有一個預先計算好的輸出值，如所圖 15 所示。當一個輸入被傳遞給激勵函數時，會將 LUT 中與該輸入最接近的值，對應到單一地址(Address)，並使用該值作為激勵函數的輸出。由於 LUT 的值是預先計算的，因此可以減少大量計算時間。但是使用 LUT 也具有一定的缺點，由於 LUT 中的值是離散的，會導致輸出的不連續性和精度損失。且為了提高可使用範圍，可能會造成面積與功耗上升，反而得不償失。

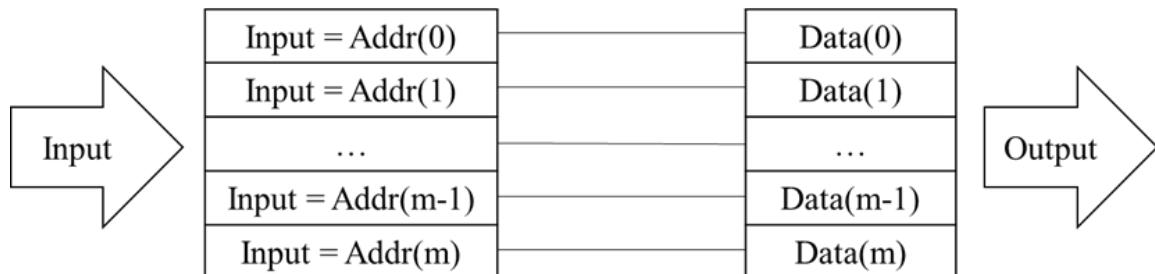


圖 15、Look-up Table 架構圖

2.2-2 以 RALUT 作為激勵函數

範圍可定址查表法(Range Address Look-up Table, RALUT)是為了解決傳統 LUT 具有離散的問題。其原理是將所需數值切割成許多區間，並控制在一定誤差內，以此區間內的地址對應到 LUT 內資料，因此能實現近似連續的激勵函數，如圖 16 所示。但仍會有在大範圍和高精度的條件下，LUT 膨脹的問題。

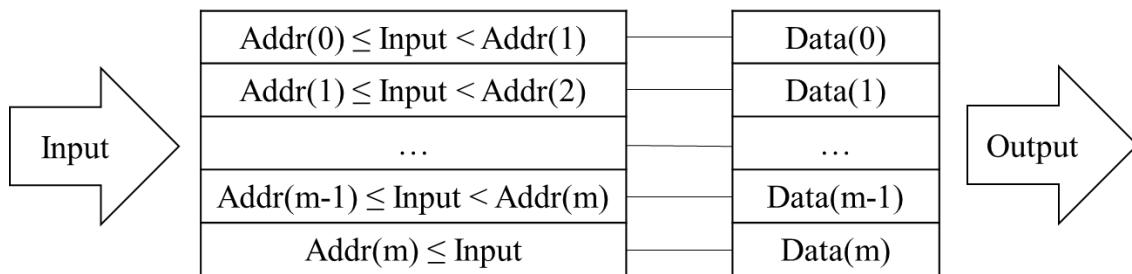


圖 16、Range Address Look-up Table 架構圖

2.2-3 以 PWL 作為激勵函數

分段線性函數(Piecewise Linear Function, PWL)[24]是由分段(Picewise, PW)建構而成，將不同的特徵區分成多種模式計算，如雙曲正切函數為例，共可以分成線性區(I)、變化區(II)以及飽和區(III)，如圖 17 所示。

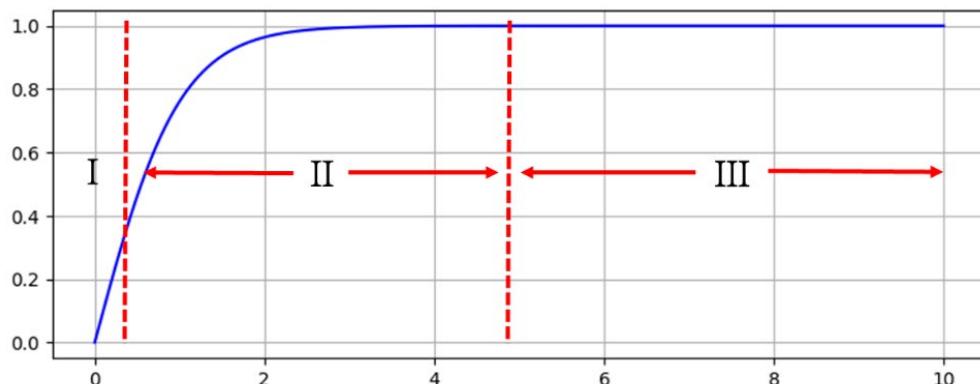


圖 17、雙曲正切函數分段示意圖

如圖 17 所示，I 區為線性區域，利用雙曲正切函數在函數 x 等於 0 時，此區域斜率會接近 1，因此能畫分為線性區域。在 II 區為變化區域，此部分是整個函數變化幅度較大區域，會以範圍式分段儲存數值，以利加速計算。而在最後 III 區域，則為平坦區域，因此稱為飽和區，由於此區域輸出值通常為 1，因此可以直接定值，實際電路設計如圖 18 所示。

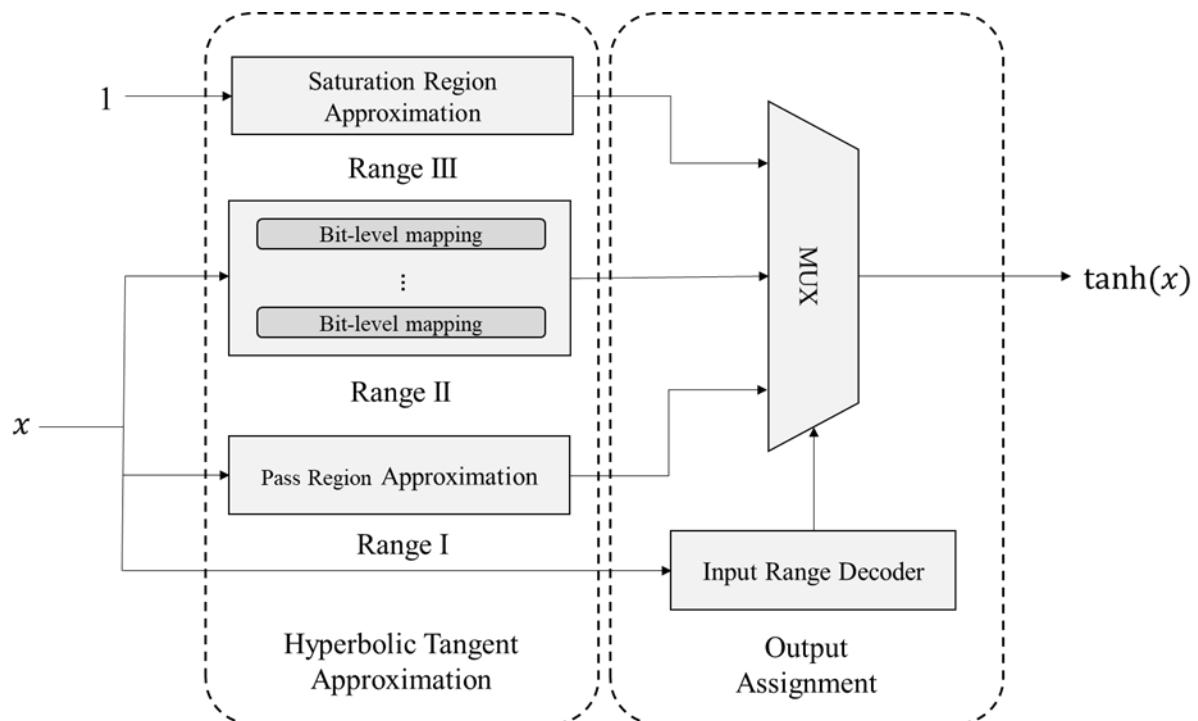


圖 18、分段線性函數電路示意圖

這種以分段線性函數所設計的激勵函數，可以在加速計算的同時降低功耗及電路成本，但仍具有不易調整的缺點，單一的分段線性函數僅能為特定函數做改善，卻無法有效調整激勵函數的形狀，對於每一種不同設定或狀態的函數都須重新規劃線段分配，進而無法降低研發成本。因此，若是有一種設計能夠任意調整為各種激勵函數，將能大大改善這個問題。

第三章 即線預訓練常態化器

3.1 近似激勵函數

根據大數法則(Law of large numbers)，當樣本數愈多資料集特徵分布會趨近常態分布，如圖 19(a)所示。因此激勵函數的累積機率分布函數(cumulative distribution function, CDF)將會是一個錯誤函數。但因錯誤函數不好計算且無法微分，所以使用雙曲正切函數作為近似錯誤函數，如公式(3.1)所示。

$$\text{erf}(x) = \frac{2}{\sigma\sqrt{2\pi}} \int_0^x e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt \quad (3.1)$$

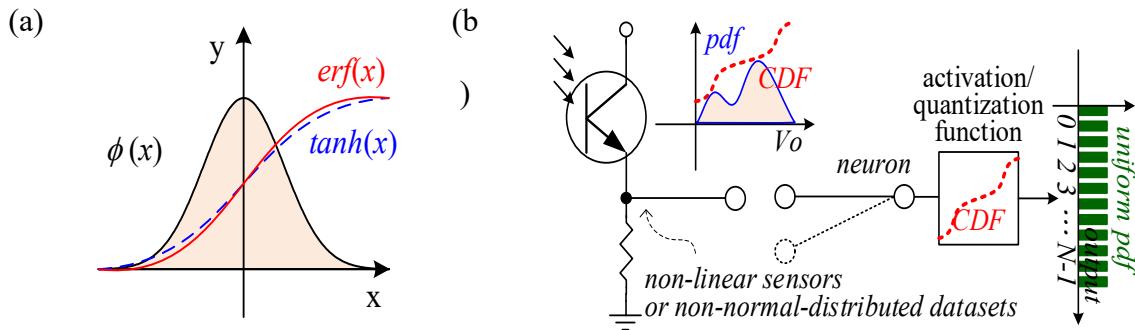


圖 19、(a)錯誤函數 (b)非常態分布錯誤函數

但若是收集兩種以上的感測器資料當作訓練資料集，其 CDF 為非常態分布函數，如圖 19 (b)所示，並不能使用 $\tanh(x)$ 作為近似錯誤函數，因此需藉由新資料集產生的機率密度函數(Probability density function, PDF)來產生新資料集的 CDF，並做為錯誤函數使用，其中 PDF 與 CDF 為積分關係，如公式(3.2)所示。

$$CDF(x) = \int_{-\infty}^x PDF(t)dt \quad (3.2)$$

3.2 四種建構單元

為了能夠近似錯誤函數，並能夠隨時調整激勵函數的形狀，我們選用 ReLU 作為建構 PWL 函數的基本單元。選用 ReLU 的因素為其具有的三種優勢，分別為計算效率高、收斂速度快以及適用於量化神經網路。

由於 ReLU 在數值的處理上，只需比較每個輸入的數值，當數值大於 0，則直接輸出該值，否則輸出 0。因此在神經網路正向傳播具有較高的計算效率。而 ReLU 的導數在正區間恆為 1，在負區間恆為 0，這樣的特性會使 ReLU 在訓練過程中，梯度更新較其他種激勵函數快，從而加快了整個神經網路的收斂速度。由於量化神經網路是將神經網路中的權重和輸入量化為定點或浮點數，從而降低神經網路的儲存和計算需求。在進行量化時，僅需要將 ReLU 的導數量化為 0 或 1，就能進行神經網路傳播，因此 ReLU 非常適合運用在量化神經網路之中。

建構 PWL 函數的四種建構單元是由 ReLU 進行擴展，如以下公式(3.3)~(3.6)所示：

$$ReLU(x) = \max(0, x - b) \quad (3.3)$$

$$ReLU(x) = \max(0, b - x) \quad (3.4)$$

$$ReLU(x) = -\max(0, x - b) \quad (3.5)$$

$$ReLU(x) = -\max(0, b - x) \quad (3.6)$$

基本建構函數圖形如圖 20 所示，與公式(3.3)到(3.6)分別對應為右上、左上、右下、左下等四種不同斜率的 ReLU 函數，其中變量 b 代表偏差值(bias)，能夠調整 ReLU 的起始點位置，藉此建構出完整的 PWL 函數。

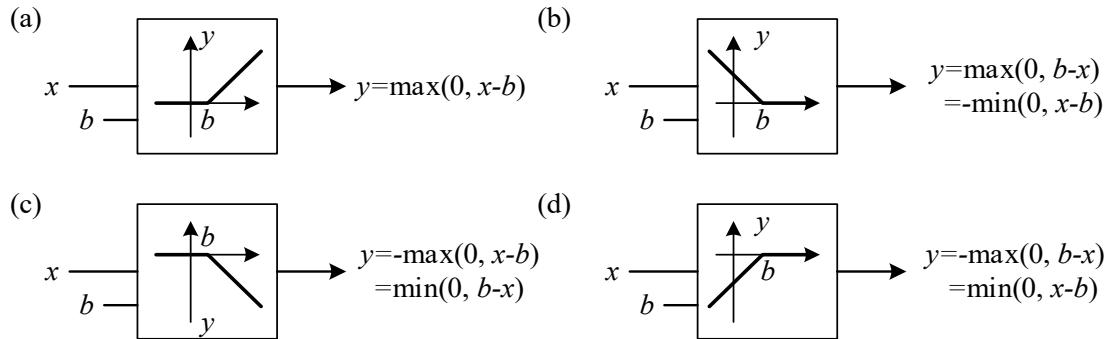


圖 20、四種 ReLU 函數圖形

為了簡化四種建構式，可以藉由定義兩個新符號 S_x 和 S_y ，數值等於 +1 或 -1 兩種形式，代表 x 軸和 y 軸的正方向及負方向，統整為公式(3.7):

$$ReLU(s_x, s_y, x) = s_y \cdot \max(0, s_x \cdot (x - bias)) \quad (3.7)$$

考慮到 CDF 函數的特性，是由 PDF 函數積分所構成，屬於單調遞增函數的特性。因此在實際應用中，不會存在 $S_x = -1$ 的情況，換句話說，函數圖形必定向右遞增，進一步將公式(3.7)簡化成公式(3.8)，並將 ReLU 函數定義為符號 \mathcal{L} 。

$$ReLU(1, s_y, x) = \mathcal{L}(s_y, x - bias) = s_y \cdot \max(0, (x - bias)) \quad (3.8)$$

3.3 輕斜率設計以及 SPINDLE 架構

為了建構可調整激勵函數，我們使用了 PWL 函數做為錯誤函數。PWL 函數主要是利用斜截式(slope-intercept form)所建構而成的，可由簡單公式構成：

$$y = a_i x + b \quad (3.9)$$

公式(3.9)中， a_i 代表斜率， b 代表偏移值。利用不同的斜率產生的線段，組合成錯誤函數，如所示，以雙曲正切函數作為目標函數，可以利用七組線段組合成 PWL。若想使誤差函數更加貼近目標函數，必須降低 PWL 與目標函數之間的誤差值，需使用更多線段來建構，所需的成本也會隨誤差降低而增加。

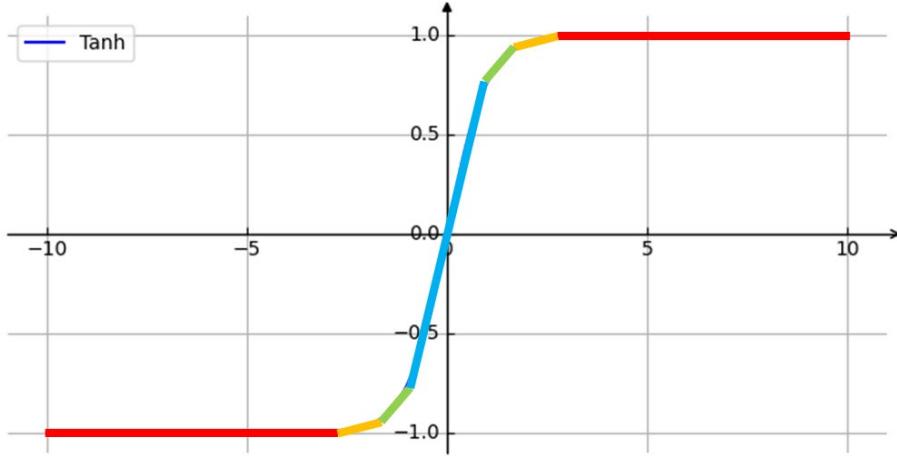


圖 21、分段線性函數

除了降低誤差外，大量乘法的使用也會造成成本上升。為了解決成本問題，在設計上會使用輕數(light Number)來解決乘法面積膨脹的問題，我們使用先前團隊研究的三元二進制(Ternary-Coded Binary Number, TCB)[25]的方式，並利用移位來取代原先乘法器，稱做輕斜率分段線性函數(Light-Slope Piece-Wise Line, LS-PWL)。

將原先公式(3.9)乘法處 a_i 藉由 TCB 定點數單元 2^{-m} 替換成公式(3.10):

$$a_i = (t_j)_{j=0}^{n-1} = 2^{-m} \cdot (t_{n-1}, \dots, t_1, t_0)_{TCB} \quad (3.10)$$

我們定義 t_j 作為 TCB 單元，其中包含0到 $n - 1$ 筆由 TCB 所轉換的輕數，而公式(3.10)可以使用 sigma 整合為公式(3.11):

$$a_i = \sum_{j=0}^{n-1} 2^j \cdot t_j \quad (3.11)$$

此處已完成乘法的輕數轉換，將 a_i 帶回原先斜截式公式(3.9)中，合併為公式(3.12):

$$y = \sum_{j=0}^{n-1} 2^j \cdot t_j + b \quad (3.12)$$

而對於所有的定點二進制補數，乘法可由移位取代，如公式(3.13):

$$2^j \times n = n \lll j \quad (3.13)$$

將單一組輕數斜截式(3.13)與我們前章節統整後的 ReLU 建構式(3.8)，可得到輕斜率分段線性函數，且由 ReLU 單元所構成，如公式(3.14):

$$y = \left(\sum_{i=1}^k \sum_{j=0}^{n-1} \sum_{S_{y \in \{\pm 1\}}} \mathcal{L}(s_y, x - b_{ij}) \lll j \right) \ggg m \quad (3.14)$$

此處符號 \lll 與 \ggg 代表符號擴展左移及右移，其中符號左移 \lll 即為當前輕斜率設計，取代原先乘法器，符號擴展右移 \ggg 具有小於最低有效位的截斷誤差，右移是為了使輸出整體以等比例縮小，以符合目標激勵函數的原先輸出值。

用移位代替乘法器的優點在於，只需利用線路連結就能進行加法運算，可以大幅降低硬體面積，從而減少晶片成本的開銷。圖 22 為 SPINDLE 的組成單元。

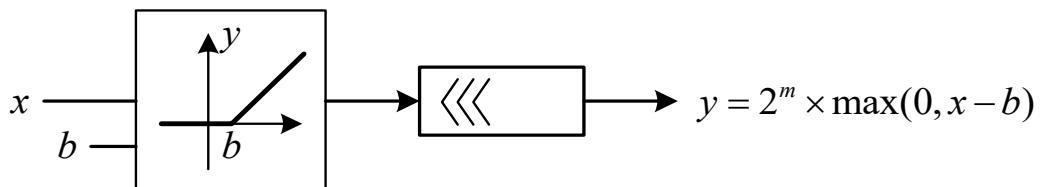


圖 22、以 2 的 m 次方作為斜率

整體的架構圖如圖 23 所示，我們將其命名為 SPINDLE 常態化器，取決於其特殊的外觀會呈現紡錘形狀，SPINDL 常態化器是由 LS-PWL 函數所建構，其中 LS-PWL 又是由四種方向 ReLU 中的其中兩種 SPINDLE 單元所建構。

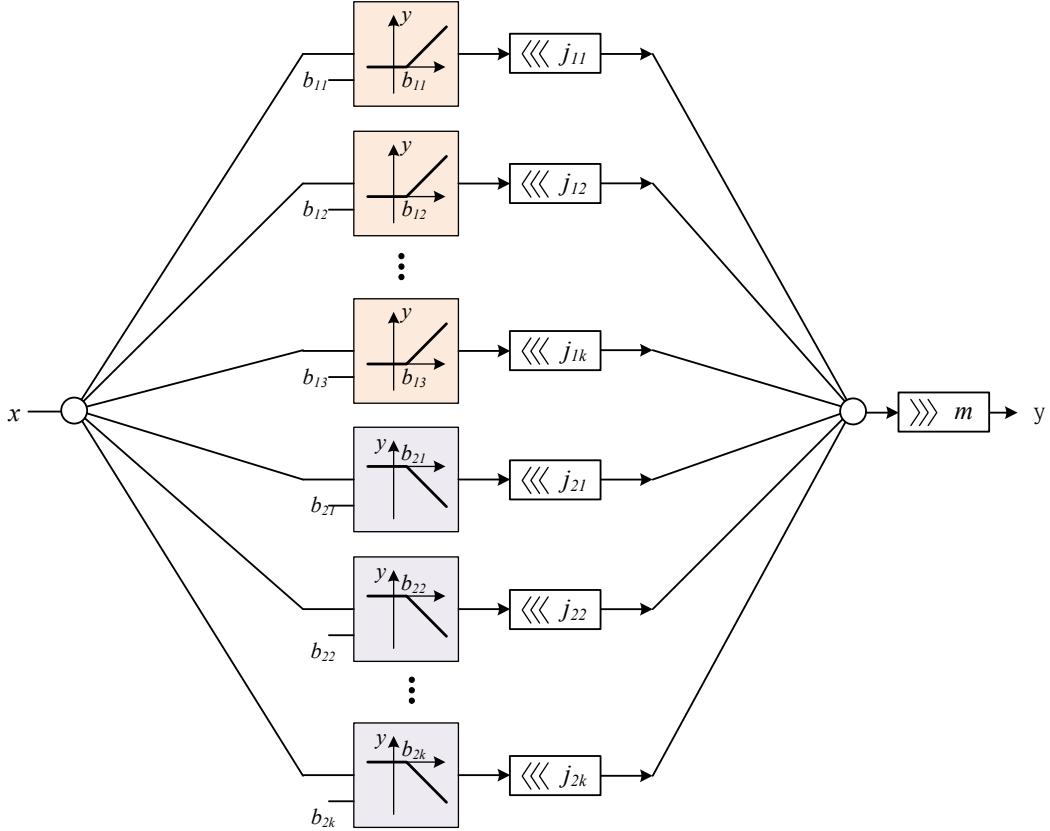


圖 23、SPINDLE 結構圖

如圖 23 所示，當輸入呈現非常態分布時，其資料分布並非均勻化，若還是使用原先的激勵函數，對於特徵分布的解析能力就會降低，而通過我們設計的 SPINDLE 常態化器後，會提取非常態分布的 CDF 作為 LS-PWL 演算法的輸入，得到相對應的分段個數、斜率、偏移量後，就能調整激勵函數的形狀，達到常態化的效果，能提升後續神經網路訓練的準確度。

3.4 二分輕數斜率分段線性搜尋法

在使用 PWL 的技術中，我們使用輕數斜率減少計算成本，並使 ReLU 單元來完成 LS-PWL 的建構。但 PWL 具有一個嚴重缺點，在選取線段的過程中，若是未能根據曲線的特性進行有效選取切割區域，將會影響分段線性函數的準確性。

為了改善這個問題，我們使用了新的演算法來解決，稱做二分輕數斜率分段線性搜尋法(Binary Search Light-Slope Piece-Wise Line, BLS-PWL)，圖 24 為雙峰分布的特徵曲線，可由 PDF 函數提供增強 PWL 演算法分段的能力，從而達到搜尋近似任意 CDF 的分段線性函數。

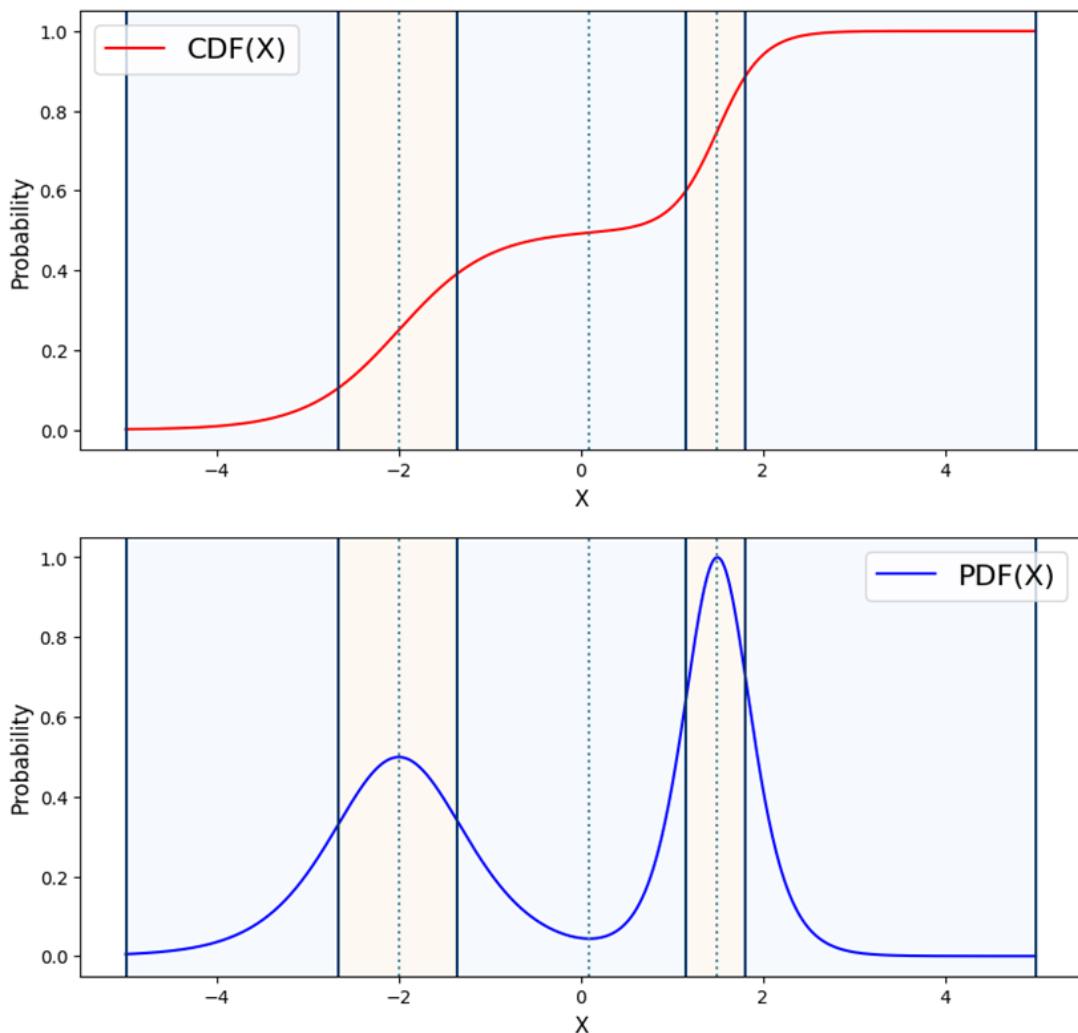


圖 24、CDF 與 PDF 關係圖

首先，我們需要介紹函數微分的兩種特性，分別是一次微分的端點，以及二次微分的反曲點。對於一次微分的端點，其背後代表的涵義為原函數上斜率的變化點。假設現存在端點 P_1 ，若在此端點左右兩點為 P_0 、 P_2 ，其特性分別為公式(3.15):

$$\begin{aligned} f'(P_0) &> 0, \text{slope increasing} \\ f'(P_2) &< 0, \text{slope decreasing} \\ f'(P_1) &= 0, \text{extremum point} \end{aligned} \quad (3.15)$$

由此可知， P_1 會是函數的局部極值，由遞增的斜率轉向遞減斜率，或是由遞減斜率轉向遞增，會是 CDF 函數斜率的轉折點。此點的函數的變化率會逐漸降為 0，代表 CDF 函數在該點附近的線段較為緩和。

因此若以此點做為線性分段的展開點，對於斜率搜尋會更有效率。而在 CDF 的微分部分，我們由定義式(3.2)可知，其背後代表的是 PDF 函數，因此我們會使用 PDF 函數的端點作為輔助斜率搜尋的出發點，如圖 24 所示，虛線部分為 PDF 的端點，實際斜率搜尋的過程如圖 25 所示。

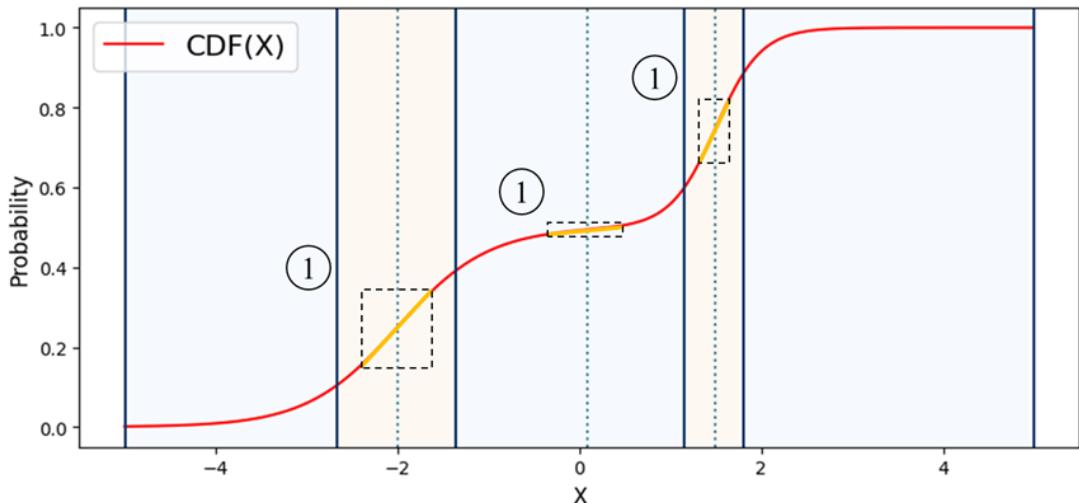


圖 25、一次微分端點

為了解決切割區域不佳導致函數不連續的問題，我們使用特定分隔線，作為輔助斜率搜尋的邊界值。此分隔線是利用二次微分反曲點的特性，找出原函數的凹凸性質。二次微分的反曲點會出現在原函數斜率變化率最大處，又常被稱作拐點，如圖 26 所示。顧名思義，在這些點上會具有曲率的最大或最小值，以此點做為斜率搜尋的分界點，能使每個區域進行各自的線段搜尋，以保持拐點的特性。

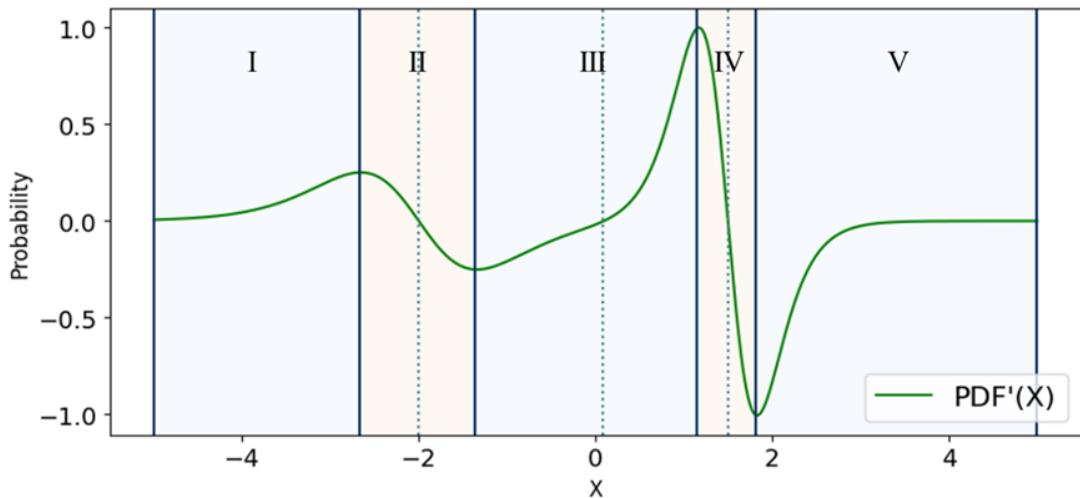


圖 26、二次微分反曲點

因此，在做二分輕數斜率分段搜尋法時，會先利用一次微分的端點以及二次微分的反曲點來做預先的輔助點及分隔線，減少不佳的線段產生。而在決定完所有的區域後，會由區間的中心點或輔助點做線段斜率的搜尋，找出公式(3.10)最佳的 m 值。

所謂的二分法，指的是在我們的搜尋演算法中，會以邊界的中心做斜率搜尋，範圍是原邊界大小，一旦此次搜尋無匹配到小於特定誤差值的斜率時，會以原範圍大小的一半作為新的邊界，如圖 26 所示，直到所有線段被搜尋完畢，因此稱作二分輕數斜率分段線性搜尋法。

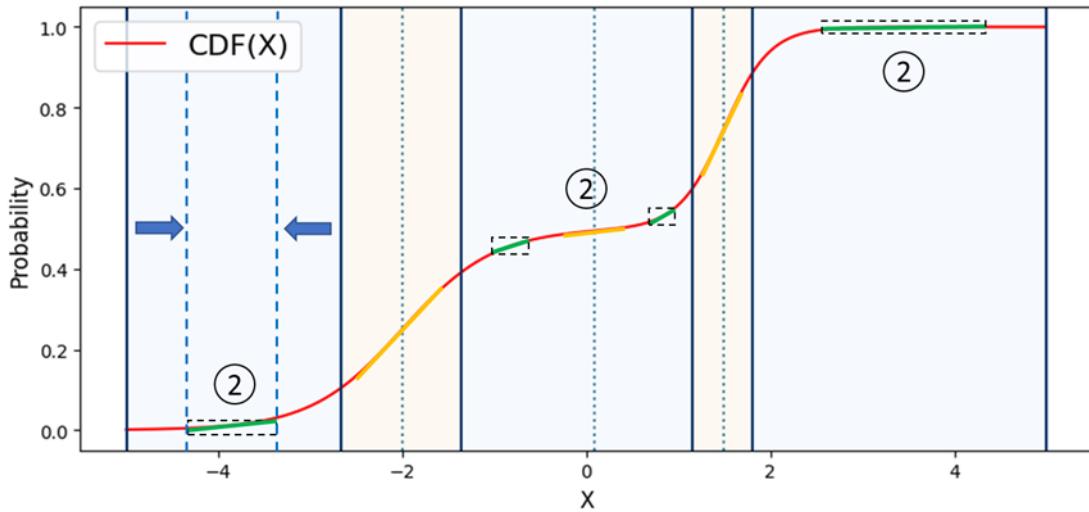


圖 27、BLS-PWL 搜尋法

當使用 BLS-PWL 演算法搜尋完所有線段後，為了增加能夠即線調整能力，會使用與正方向斜率大小相同的負方向 ReLU，做為調整 SPINDLE 形狀的單元。這樣做的目的是為了使正方向增加的斜率能調整回水平軸上，代表兩者斜率相互抵消，如式 3.16 所示，防止斜率無限增加，導致偏離原目標函數。實際操作如圖 28 所示。

$$\text{Slope} = 2^m + (-2^m) = 0 \quad (3.16)$$

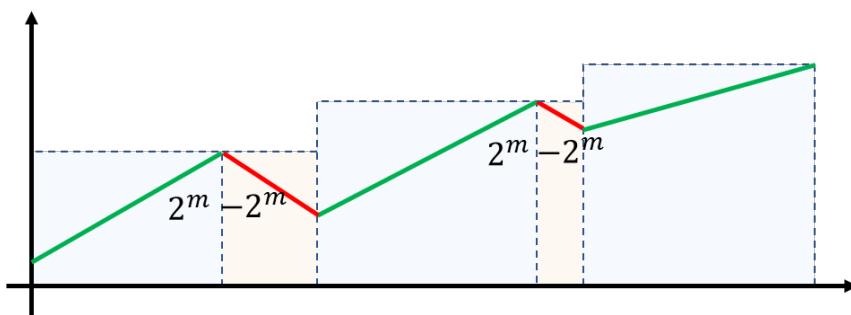


圖 28、負斜率調整 PWL

第四章 離線預取樣常態化器

除了即線預訓練常態化器(SPINDLE)外，在我們也可以在獲取訓練資料集後，利用離線預取樣的方式，直接取樣整個資料集的特徵分布，不再是利用部分資料作激勵函數圖形微調，如此便能大幅降低誤差值，使訓練出來的圖形更加貼近原資料集的特徵分布，減少在使用即線預訓練常態化器時，少部分資料會造成激勵函數圖形產生發散情形。

4.1 預取樣排序法

在神經網路領域，由於資料集的龐大，訓練全資料集的計算成本過高，通常會透過批次學習或預取樣的方式獲取部分資料集的特徵後，立即調整模型的權重值，可以更快訓練神經網路，並使用較少的計算資源執行探索性數據分析。基於這樣的想法，我們提出預取樣排序法(Sample Ranking Method)。

首先，會進行取樣部分，以隨機取樣的方式將原資料集抽出一小部分資料，稱做子資料集，並以 2^n 次方做為子資料集的數量單位，對於硬體處理較為便利。

接著，根據取樣的子資料集做排序(Sorting)的動作，由小到大依序排序，此時能得到與原資料集相似的資料分布。為了能做到量化，我們使用了階級排序(Ranking)的概念，將原資料集做常態化處理，過程如圖 29 所示。

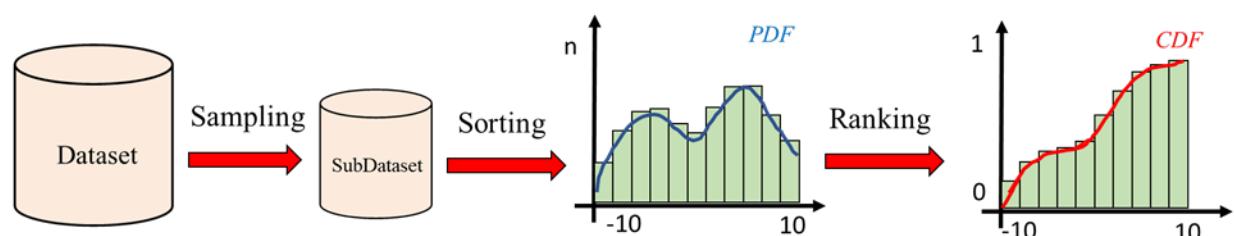


圖 29、預取樣排序法流程圖

階級排序的概念可以以一種簡單的方式進行說明。假設存在某個班級上，男生與女生的成績分別由兩組常態分布所建構，呈現雙峰化的分布情形，其中部分成績為 $\{23,24,25,26,27,73,74,75,76,77\}$ ，兩峰值分別位於 25 與 75 處。由於成績是由兩常態分布組成，在峰值處分數接近或相同者的同學數量較多，不容易進行學生鑑別度(Identification)分析。為了保持原本分布圖形，又能使同學們每筆成績能均勻分散，此時就會需要利用排序法，做量化的動作。

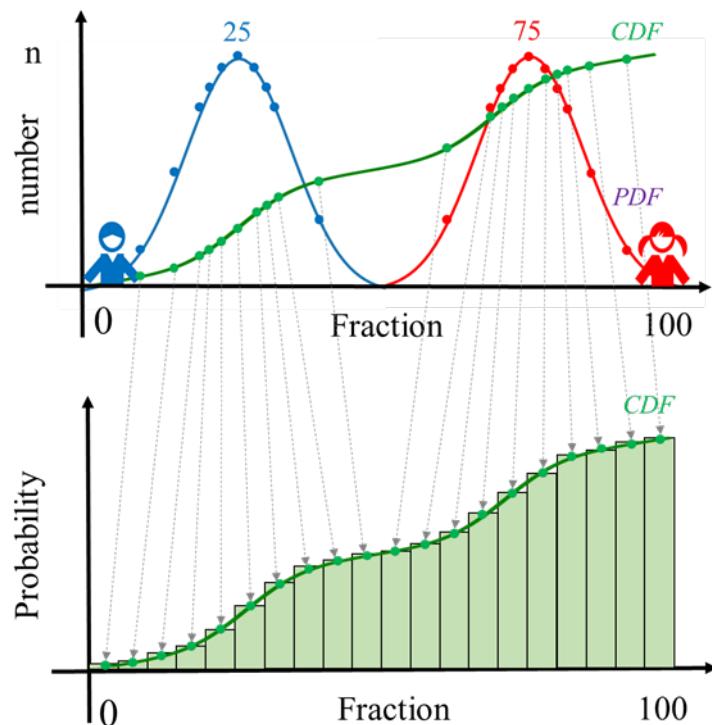


圖 30、使用取樣排序法量化學生成績

首先，會先將原本雙峰分布的 PDF 轉成 CDF，根據學生的總數 n 對 CDF 圖形做切割，將數據按照一定的區間進行劃分，此區間稱為 bin。在每個 bin 上皆會產生新的數值，此數值可藉由排名(Ranking)的方式，將同學的成績映射到產生的新數值上，既能保持 CDF 函數與原函數相近，又能量化 CDF 函數。結合先前的取樣(Sampling)動作，將此作法稱做取樣排序法。

4.2 逆轉換取樣法

除了透過預取樣排序法可以進行量化 CDF 函數外，逆轉換抽樣法(inverse transform sampling)也能達到相同的效果。使用該方法可以隨機生成與原分布相似的樣本。逆轉換取樣的原理是將一個隨機變量的分布轉換成另一個隨機數產生的均勻分布，並近似原隨機變量的分布情形。

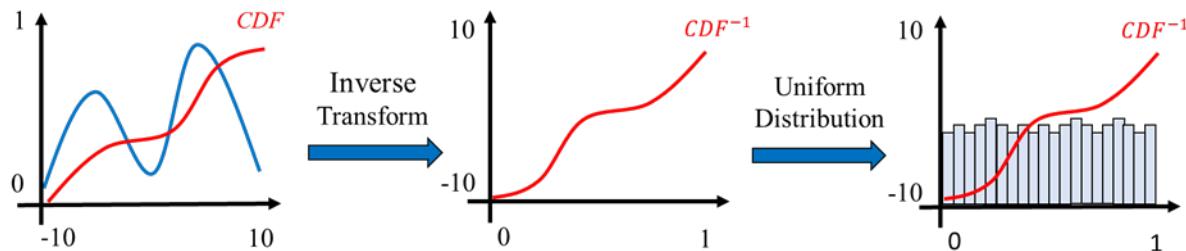


圖 31、逆轉換取樣法流程圖

如圖 31，首先對於一個隨機變量 X 和其機率密度函數 $PDF(x)$ ，其累積分布函數為公式(4.1)：

$$CDF(x) = P(X \leq x) = \int_{-\infty}^{\infty} PDF(t)dt \quad (4.1)$$

將 $CDF(x)$ 等式的兩側進行逆變換，令 $Y = CDF(X)$ ，則可得式公式(4.2)

$$U = CDF(X) \quad (4.2)$$

其中， U 是一個均勻的隨機變量，其取值範圍在 $[0,1]$ 之間。接著，將公式(4.2)進行轉換，可以得到公式(4.3):

$$X = CDF^{-1}(U) \quad (4.3)$$

因此可知，公式(4.3)會是原 CDF 的反函數，而由於 U 是一個均勻分布的隨機變量，因此生成的隨機變量 X 也會是均勻分布，也就是量化型態。

此方式是基於 CDF 函數是一個單調遞增(monotonically increasing)的函數，且為連續函數，因此能將分布的支撐域(support)映射到區間[0,1]。CDF 的反函數將[0,1]區間映射回原分布的支撐域，如圖 32 所示。

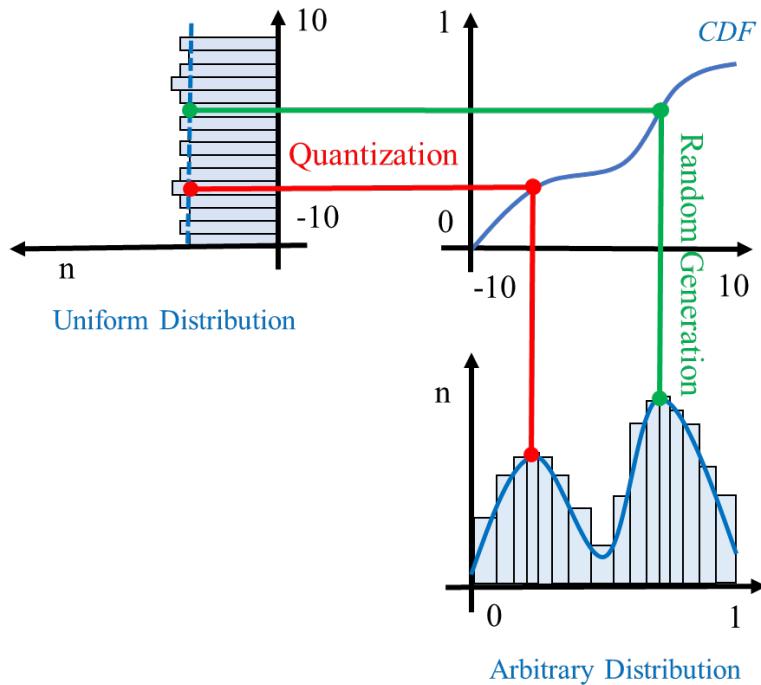


圖 32、逆轉換取樣法原理圖

因此，通過生成均勻分布的隨機變量 U ，並使用 CDF 的反函數轉換它，能得到具有原分布的隨機變量，因此稱做逆轉換取樣法。

逆轉換取樣法的優點有以下幾點：

1. 在擴展性方面，能夠取樣任意隨機變量 X 之 CDF 函數，並具有量化特性。
2. 在簡易性方面，僅需計算 CDF 的反函數，相較於其他種取樣方式較為簡易。
3. 在精確度方面，由於能夠生成與目標相同的機率密度，準確度較高。
4. 在獨立性方面，由於本身均勻分布是隨機生成，相較於其他取樣方式生成資料更加獨立，不會有干擾情形。

4.3 特徵提取器

在神經網路中，為了解析原資料集的特徵分布，必須先進行特徵提取。以圖 33 為例，在骨幹神經網路中，如果輸入資料的 PDF 為非常態化分布，若是以一般的激勵函數進行使用，無法近似特殊的錯誤函數，我們必須去調整分類器的激勵函數，使錯誤函數去近似特徵的 CDF，才能達到量化效果，以提升模型準確度。

因此，必須先提取在分類層前的特徵值，在設計上會先進行資料集部分取樣，使用預取樣排序法將原資料集取樣為子資料集。取樣子資料集後，藉由骨幹網路中卷積網路的多層卷積層和池化層後，轉換成多個特徵圖(Feature Map)，在透過展平輸入(Flatten)的方式，將多維的特徵圖轉成一維向量，並記錄在特徵圖資料集中。

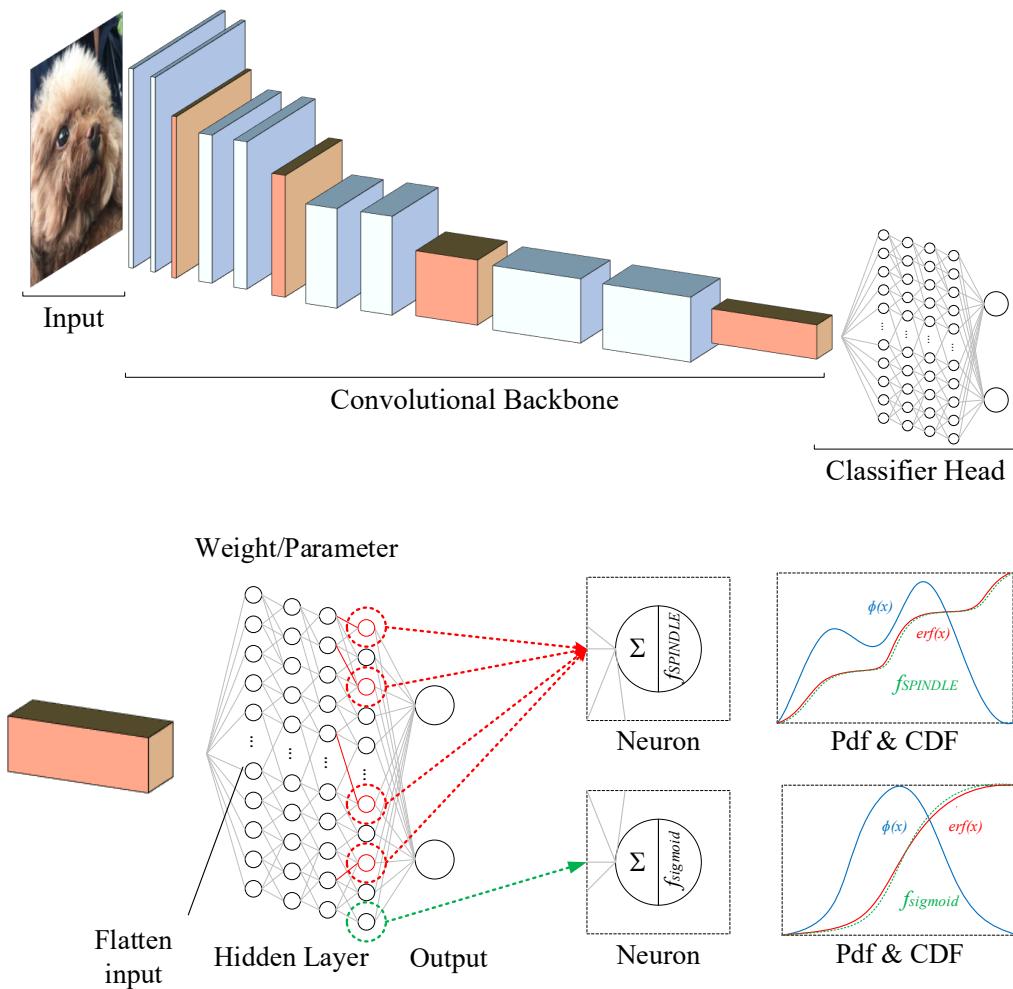


圖 33、骨幹網路的特徵提取器

在此處設計上，特徵的收集會根據骨幹神經網路的深度神經網路做改變，特徵提取器是連接於全連接層的倒數第二層，作為就地(In-situ)處理、分析前一層神經元產生的特徵圖，進行激勵函數形狀的調整。再將 SPINDLE 轉化的輸出傳遞至分類器中進行目標分類。

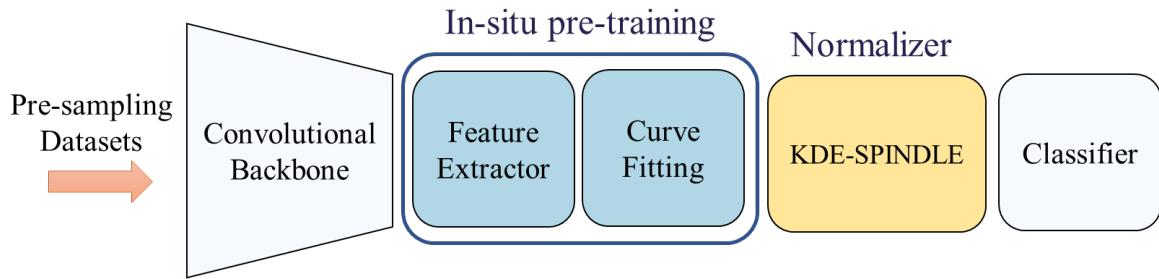


圖 34、神經網路連結就地常態化器

分類器通常是指全連接層的最終層，一般而言會是使 softmax 激勵函數做分類。此外，特徵分布圖是提取向量中的索引(index)，也就是來自每張圖片的某個單一點特徵，藉由分析每個特徵點的分布圖，來改善每個神經元所選用的激勵函數，如圖 35 所示，在得到每個特徵點的向量後，會轉存成表格，再將各點的特徵分布圖輸出，當作 SPINDLE 的學習目標。



圖 35、提取特徵分布流程圖

4.4 搜尋非常態特徵分布

在收集所有特徵分布圖後，部分的特徵具有常態分布的性質，而我們主要改善的是屬於非常態分布的特徵分布曲線，因此在收集特徵分布圖後，必須經由檢測法去挑出非常態分布的圖形。

在統計學中，常使用的無母數統計(nonparametric statistics)的方式，來檢測母群體分布情況未明情形下，所應用的統計方式。換言之，無母數統計代表不需要事先假設待測的資料分布。而其中，科摩哥洛夫-史密諾夫檢定法(Kolmogorov-Smirnov test, K-S test)[26]作為我們選定的統計方法。

K-S 檢測法是一種常用的非參數假設檢驗方法，其定義是通過比對觀測值和理論分布的差異，來評估觀測值是否來自於該理論分布。

在進行統計檢驗時，首先會進行虛無假說(Null hypothesis)， H_0 為觀測值符合該分布，此處應用為常態分布。 H_a 為觀測值不屬於該分布。K-S test 的原假說屬於虛無假說，也就是認定觀測值分布屬於常態分布。

給定一個獨立同分布的樣本 x_1, x_2, \dots, x_n ，隨機變量具有分布函數 $F(x)$ 。

$$F(x) = P(X \leq x) \quad (4.4)$$

進行 K-S 檢測法，檢測法定義為式(4.5)，其中 F_n 經驗分布函數(Empirical Cumulative Distribution Function, ECDF)，檢測值 D 為兩種樣本分布的最大差距，如式(4.5)所示：

$$D = \max_{1 \leq i \leq n} (|F_n(x_i) - F(x_i)|) \quad (4.5)$$

若 D 值愈大則可認為兩個樣本的差異愈大，代表兩者來自不同的分布，否決虛無假說。經過 K-S 檢測法可以有效區分出該輸入特徵分布是否符合常態分布，一旦檢測結果為非常態分布後，則可進一步使用 SPINDLE 來調整原骨幹網路的激勵函數。

4.5 核密度估計 SPINDLE 設計

在經過 K-S 檢測後，若檢測結果非虛無假說，代表測試的特徵分布不屬於常態分布，若此時仍使用原先的激勵函數，對於後續模型準確度提升會有所限制。而為了方便模型在調整上更加便利，因此使用核密度估計(Kernel Density Estimation, KDE)，作為建構 SPINDLE 的方式。

核密度估計原理是來自組合 PDF 函數的想法，由於資料集來自多種常態分布的組合，因此我們可以使用多個 logistic 或 tanh 進行組合，此方法稱作核密度估計。藉由多個核函數模擬合成 PDF 函數，如圖 36 所示，該 PDF 呈現雙峰分布，在核密度估計中可以使用五個核函數，其基本單元為常態分布函數，作為雙峰分布 PDF 函數的建構。

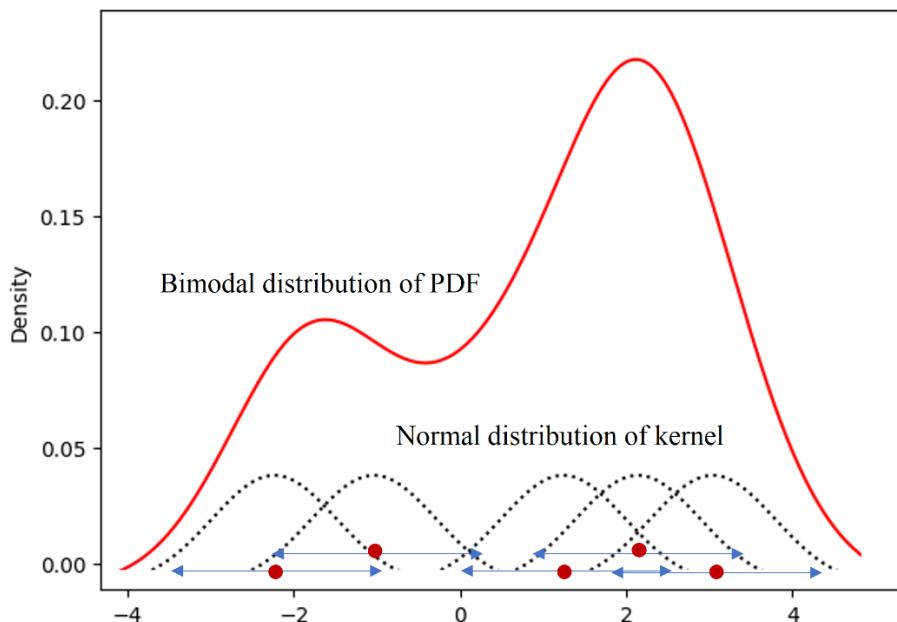


圖 36、核密度估計

原先方式是以 BLS-PWL 搜尋斜率後，使用多段 ReLU 作為建立 SPINDLE 常態化器，但在實驗過程中，由於線段的組合在後續訓練微調上容易產生造成發散，無法穩定輸出相同的 CDF，且有機率造成 CDF 不連續的問題，因此在此處改用多個 logistic 作為建構 SPINDLE 的基本單位，將此方式稱作 KDE-SPINDLE，整體架構圖如圖 37 所示。

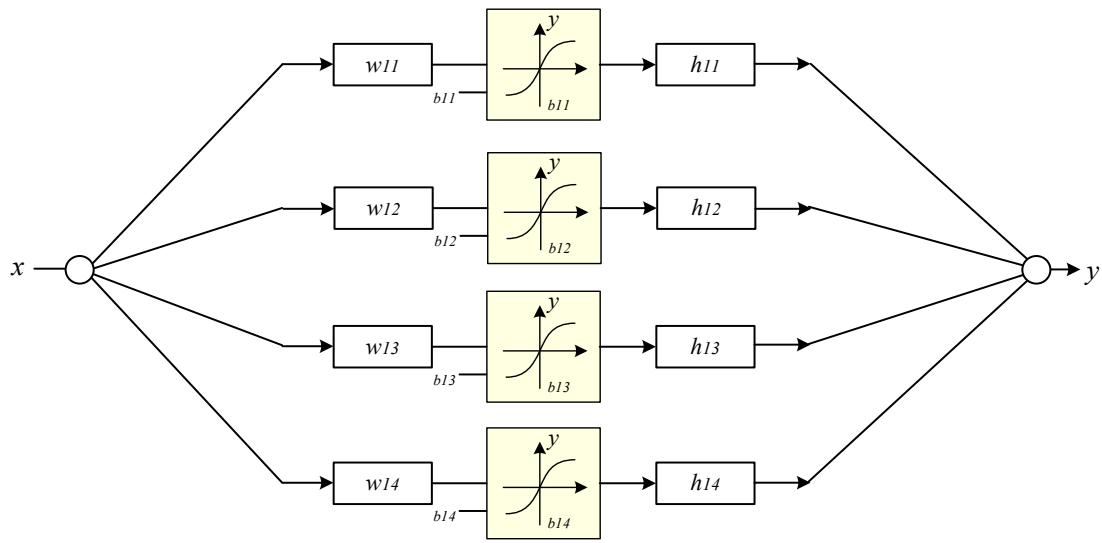


圖 37、KDE-SPINDLE 架構圖

$$y = h * \text{torch.sigmoid}(w * x - b) \quad (4.6)$$

KDE-SPINDLE 中，變數 b 代表每個核函數的偏移量，變數 h 和 w 則是代表能夠每個核函數的高度與寬度的比例，藉此能任意調整激勵函數的形狀。

在調整上，會先利用特徵提取器提取出特徵分布圖，再由 K-S 檢測法檢測非常態分布的特徵分布，最後使用多個 logistic 擬合特殊分布曲線，由於 logistic 兩端處較為平滑，在處理線段連續性的問題上，擁有較優的擬合能力，改善 BLS-PWL 搜尋法產生的 SPINDLE 圖形微調後容易發散的缺點。

第五章 實驗

在本章 5.1 小節會先說明二分輕數斜率分段線性搜尋法(BLS-PWL)演算法，並以多種特殊分布作為輸入，並在多種誤差條件下，皆能生成相對應斜率、偏差以及分段範圍，由於 BLS-PWL 演算法考慮了拐點因素，將輕易實現原先 LS-PWL[25]無法建構的特殊分布輸入。

在 5.2 小節會實現即線預訓練常態化器，由 ReLU 建構而成的 SPINDLE 常態化器。會先以 Python 實現，證明具有微調的功能後，產生對應的 Golden data。接著，設計硬體電路的 SPINDLE 常態化器，並經歷完整的 IC 設計流程，通過 Pre-sim、Post-sim、DRC、LVS 驗證後皆與 Golden data 匹配且符合晶片廠規範。在此實驗中，能夠進行即線預訓練調整形狀的部分，但由於每次輸入單一值容易造成 SPINDLE 常態化器發散，無法完全匹配目標函數，針對此問題我們提出離線預取樣方式進行改善。

在 5.3 小節會呈現通過預取樣排序法及逆轉換取樣法達到量化取樣效果，並驗證原始與取樣資料集間的誤差，證明在取樣後誤差範圍值不會過大。

而為了證明在特殊的特徵分布下 SPINDLE 常態化器能有效改善準確度，因此在 5.4 小節，會使用 NIR 資料集[2]做為訓練集，並與 AlexNet[10]、ResNet-18[11]、VGG-16[12]骨幹神經網路進行比較，所提出的 SPINDLE 常態化器皆能有效提升準確度。

然而，受限於 NIR 資料集[2]資料量不足，因此使用 Python 腳本產生自定義的 PolyMNIST 資料集[27]，並在 5.5 小節比較各常見的激勵函數，包含多頭骨幹網路 Vision Transformer (ViT)[17]所使用的 GeLU 以及單頭骨幹網路 Yolo[19]使用的 LeakyReLU、Mish，做為比較對象，在三種自定義資料集下，使用 LeNet-5[9]網路架構，皆能有高達 97~99% 的準確度，並大幅縮小不準確度的範圍，證明 SPINDLE 能有效解決資料不均勻產生的非常態化特徵分布問題。

5.1 線性搜尋法改進與結果比較

線性搜尋法雖然在近似特殊目標函數的使用上，能大大降低運算成本，但若在分段上未依照目標函數的特性進行分段，在函數變化較大處，將不容易找到適合的線段。

為了改善這個缺點，我們提出了二分輕數斜率分段線性搜尋法(BLS-PWL)，與先前輕數斜率分段線性搜尋法的差異來自於預先切割區段。實際演算法如圖 38 所示

```
Algorithm Dichotomy LS-PWL

def PWL(m,bias,X):
    return (2**m) * X + bias
def intercept(X):
    return CDF(X) - PWL(m,bias,X)

d1 <- gradient(CDF(X))
d2 <- gradient(gradient(CDF(X)))
(In order to find ALL blocks and start points)

for All blocks in PWL(slope from -m to m):
    if PWL (L side + R side) < error rate
        output result(err,m,c,bias,nL,nR)
    else
        do another PWL
        if all PWL > error rate
            do range / 2
        else
            output all PWL(err,m,c,bias,nL,nR)
```

圖 38、BLS-PWL 演算法

首先，定義 PWL 的斜截式的構成，由斜率(m)和偏差值($bias$)所控制，並定義計算 PWL 與原函式的截距(intercept)，此截距會是決定可容許錯誤值(error rate)的依據。在做 PWL 建構前，會先利用一次微分的端點找出 PWL 的中心點，並使用二次微分的反曲點切割出所有區塊。

在輸入部分，以式(5.1)做為非常態特徵分布的累積分布函數 CDF:

$$f(x) = 0.5 * \text{logistic}(x + 2) + 0.5 * \text{logistic}(2 * x - 3) \quad (5.1)$$

目標函數屬於雙峰分布函數(Bimodal Distribution Function)，其一次微分的圖形即為該函數的機率密度函數(PDF)，呈現雙峰分布，如圖 39 (a)所示。

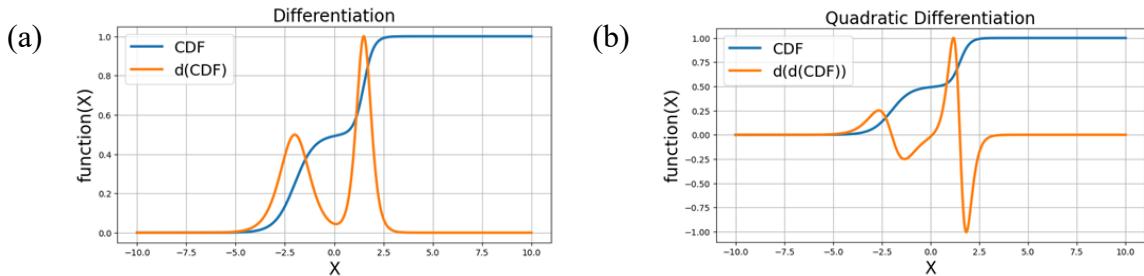


圖 39、(a)一次微分圖形 (b)二次微分圖形

如圖 39(b)及圖 40(a)，藉由分析二次微分圖形的端點後，以此作為演算法的邊界值，切割成五大區段，並根據一次微分的端點，做為初始訓練的起始點，作為後續進行 BLS-PWL 的依據，最終結果如圖 40(b)所示。

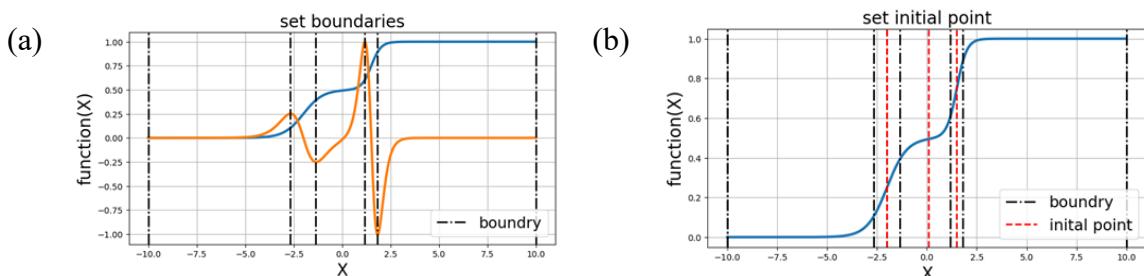


圖 40、(a)設立邊界值 (b)設立初始點

接著開始對每個區塊做二分輕數斜率分段線性搜尋法，此時開始測試此區間的斜率，一旦某一個斜率產生的 PWL 與目標的 CDF 線段截距誤差小於某一範圍後，將搜尋結果記錄於陣列中。

如圖 41 所示，目標函數為 logistic 函數，根據不同區段進行斜率搜尋的過程。

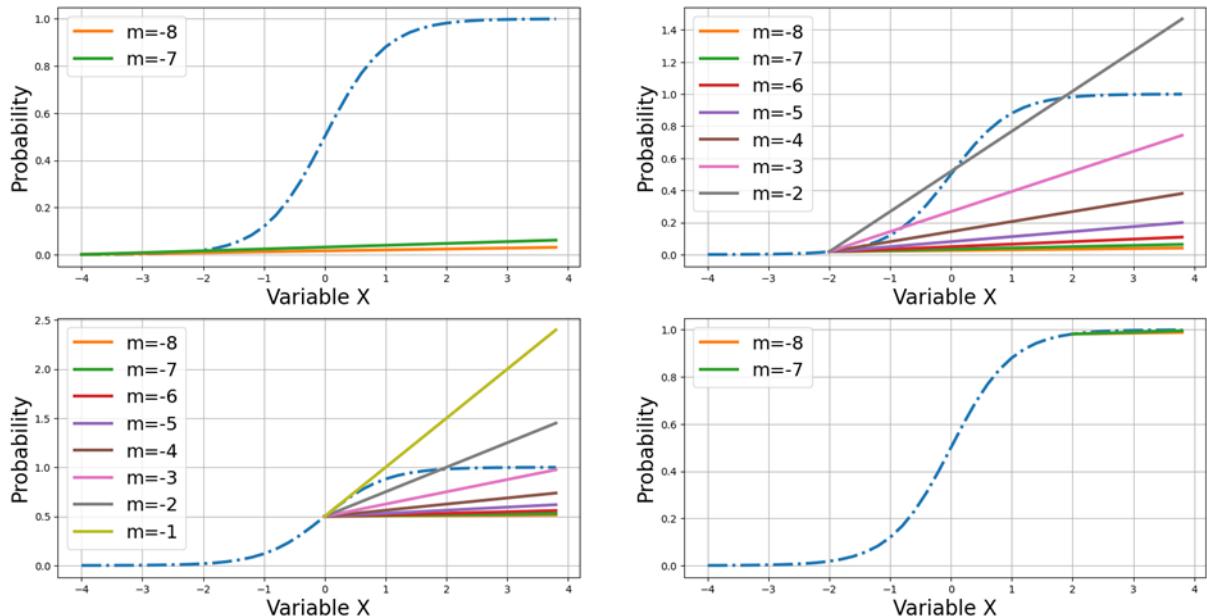


圖 41、logistic 函數斜率搜尋

若區段內皆無符合的斜率時，將會將此區塊的邊界左右縮小一半，並將前後切割部分重新記錄在未做 PWL 的區段中，並重複進行二分輕數斜率分段線性搜尋法，直到找到該中心點處 PWL 的最佳解為止。

當找到該處最佳解後，會跳至下一區段還未進行斜率搜尋的區域，重複執行 BLS-PWL 搜尋法，直到所有目標函數的區段皆完成分段線型函數的建構為止。

在進行二分輕數斜率分段線性搜尋法時，輸出的結果依序是斜率、該線段中心點、偏移量、左邊界以及右邊界。

```

[ -2.      -2.015   0.75    -2.1788  -1.8512]
[ -2.     -1.7438   0.7486  -1.8512  -1.6363]
[ -2.     -1.6056   0.7452  -1.6363  -1.5749]
[ -2.     -1.5346   0.7423  -1.5749  -1.4943]
[ -2.     -1.4675   0.7387  -1.4943  -1.4406]
[ -3.     -1.4305   0.5576  -1.4406  -1.4204]
[ -3.     -1.4003   0.5593  -1.4204  -1.3801]
[ -3.     -1.3701   0.5608  -1.3801  -1.36   ]]

| Area: 3 Range: -1.36 ~ 1.16 |

#####
ans:
[[ -8.      -9.8497  0.0385 -10.      -9.6994]
[ -8.     -9.3987  0.0367 -9.6994 -9.0981]
[ -8.     -8.9477  0.035  -9.0981 -8.7974]
[ -8.     -8.3966  0.0328 -8.7974 -7.9957]
[ -8.     -7.8454  0.0307 -7.9957 -7.6951]]

```

圖 42、二分輕數斜率分段線性之數據

在產生數據後，即可藉由數據轉化成分段線性函數圖形，如圖 43 所示。

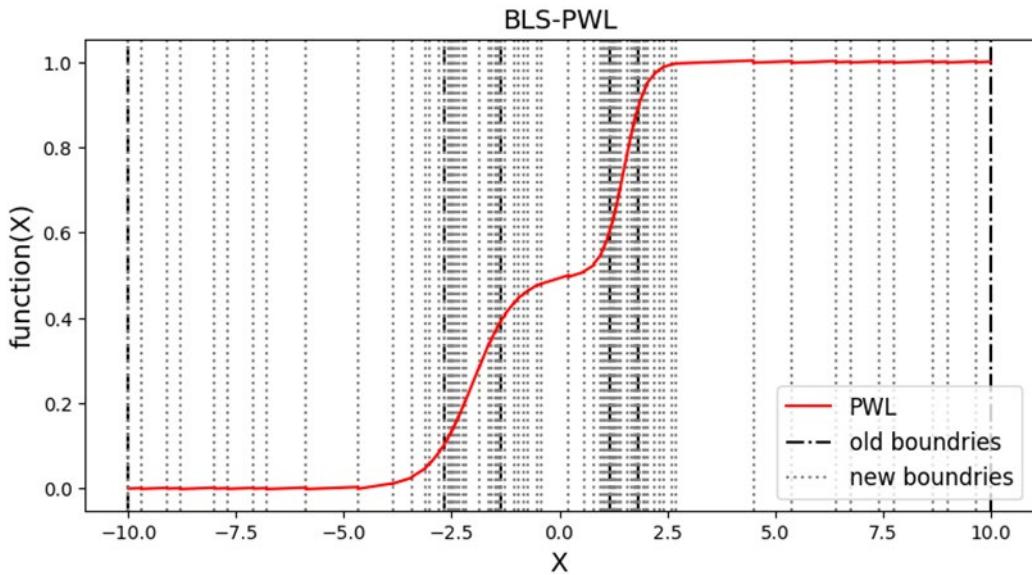


圖 43、二分輕數斜率分段線性之圖形

除了式(5.1)外，為了與先前演算法[25]做比對，選擇多種特殊分布進行實驗，設計目標函數的機率密度函數為單峰分布、雙峰分布以及多峰分布函數。將式(5.1)以及式(5.2)和式(5.3)進行實驗，表 2 為 BLS-PWL 使用成本。

$$y = \text{logistic}(X) \quad (5.2)$$

$$y = 0.1 * \text{logistic}(X + 2) + 0.2 * \text{logistic}(2 * X - 3) + 0.3 * \text{logistic}(X - 5) \quad (5.3)$$

表 2、BLS-PWL 搜尋法搜尋結果

BLS-PWL		式(5.2)			式(5.1)			式(5.3)		
PDF Type		Unimodal			Bimodal			Multimodal		
Target error		0.01	0.005	0.001	0.01	0.005	0.001	0.01	0.005	0.001
Cost	PWL	23	33	163	17	35	137	19	31	129
Cost	ReLU	46	66	326	34	70	274	38	62	258

表 3、LS-PWL[25] 搜尋法搜尋結果

LS-PWL[25]		式(5.2)			式(5.1)			式(5.3)		
PDF Type		Unimodal			Bimodal			Multimodal		
Target error		0.01	0.005	0.001	0.01	0.005	0.001	0.01	0.005	0.001
Cost	PWL	7	11	32	NA	NA	NA	NA	NA	NA

圖 44 為雙峰函數實驗結果，可以發現使用 LS-PWL[25]演算法的結果與預期結果不匹配，推斷該演算法僅適用於一次函數，如: ReLU、Sigmoid、Tanh 等。

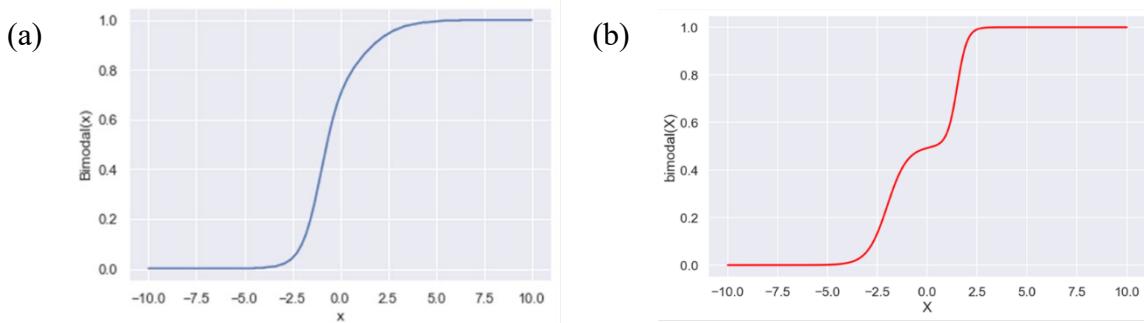


圖 44、(a)LS-PWL[25]雙峰曲線 (b)BLS-PWL 雙峰曲線

根據表 2 以及表 3 實驗結果，可以發現 LS-PWL[25]在改善線段方面比起我們使用的 BLS-PWL 更為優秀，但是在實際給於雙峰函數時，卻因為過度優化造成雙峰特性消失，從而無法正確的合成雙峰分布的激勵函數，由於我們的方式是利用二次微分的拐點作為分界的依據，因此在演算法搜尋線段時，不論進行何種調整，都不會導致特性消失。這是 LS-PWL[25]未考慮到的問題。

除此之外，我們新推出的演算法是為了能夠任意調整激勵函數，因此在使用架構上與原先設計不同，需要另一組負方向的 ReLU 提供微調的功能，這是原先演算法[25]無法做到的部分，原先演算法僅能以固定大小的 LUT 進行合成近似激勵函數的電路，若是想改變激勵函數的形狀，就必須重新進行演算法產生新的 LUT，此過程相當耗費時間成本及人力。

根據上述優缺點，我們得出以下比較表，從表 4 中可以看出，儘管我們的方法需要犧牲 ReLU 的成本，但卻能夠隨時調整激勵函數的形狀，這對於提高方法的可適應性和靈活性具有極大的貢獻。此外，由於我們的方法基於二次微分的拐點作為分界的標準，因此也增加了演算法的穩定性和可靠性，能更準確的合成目標函數。

表 4、兩種演算法比較表

	LS-PWL[25]	BLS-PWL
Fine-Tune	No	Yes
Hardware circuit	Yes	Yes
PWL Cost	Low	high
Function	Only Unimodal distribution	Any distribution

5.2 即線預訓練常態化器實現

即線預訓練常態化器是由四種 ReLU 所構成，稱做 SPINDLE，經由公式推導後，使用 ReLU 即可建構完整 SPINDLE 架構。將 BLS-PWL 演算法產生的斜率，輸入進 SPINDLE 中，為了證實此部分可以使用電路實現，除了使用 Python 語言展示四種不同的 CDF 曲線來證明 SPINDLE 的微調性外，還使用 Verilog 語言合成該 SPINDLE 電路。

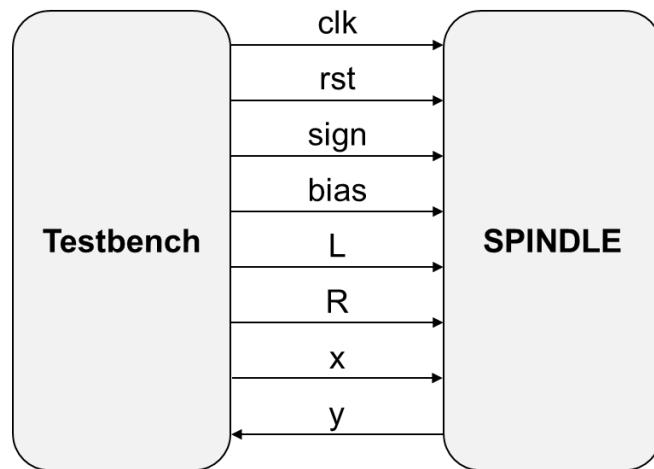


圖 45、系統方塊圖

如圖 45 所示，首先，分別輸入正負符號、偏差、左移以及右移數值後，需先在 Python 上使用十進制轉二進制的功能，生成 BLS-PWL 的參數值，寫入至 tb.txt 作為 testbench 參數檔案使用。

接著執行 SPINDLE 函式，電路時序圖如圖 46 所示，輸入 x 值會立即通過 SPINDLE 轉換輸出 y 值。輸入 x 範圍為 -32 到 32，並輸出經過四種不同激勵函數的 y 值，如圖 47 所示。最後再將四種輸出結果寫入 golden.txt，作為驗證硬體電路的正確性。

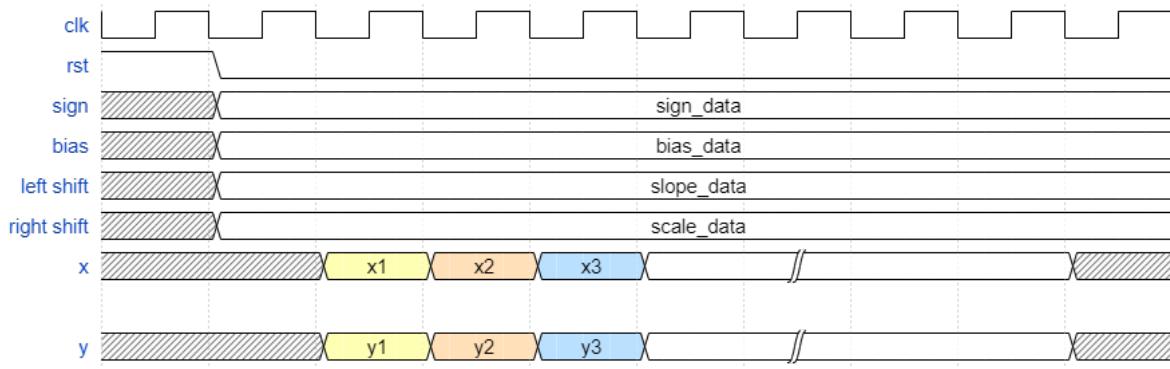


圖 46、時序圖

四種激勵函數皆是使用 ReLU 搭建，由於考量到硬體電路的實現，若要求高精度激勵函數所需的 ReLU 需求龐大，此處實驗目的在於 SPINDLE 的可調性及硬體電路的實現，因此使用四種相異斜率的 ReLU 函數搭建 SPINDLE，共有正負方向不同的八種 ReLU 單元。

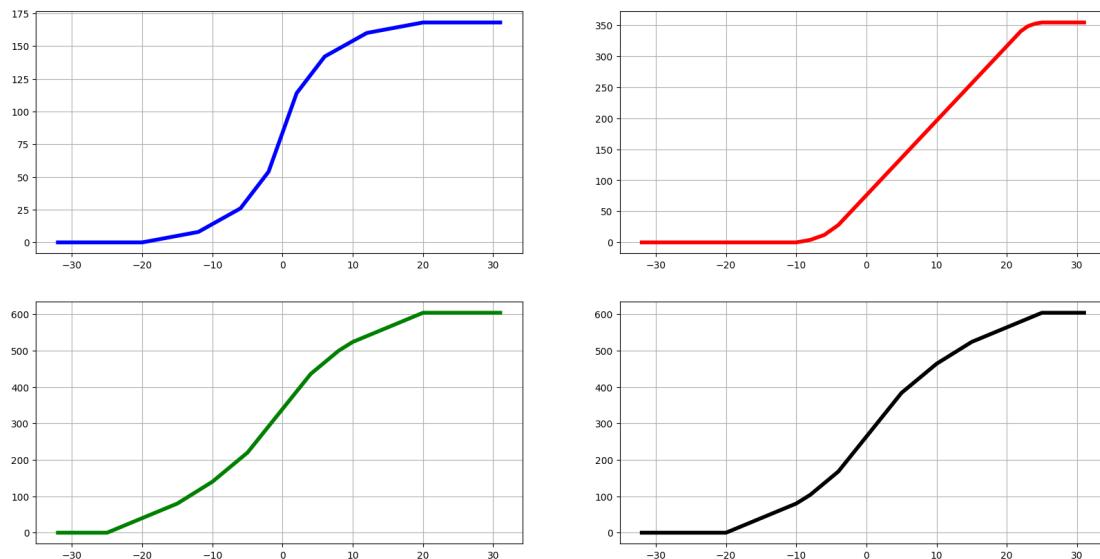


圖 47、四種不同的激勵函數圖形

接著，進行 Python 轉換為 verilog code。ReLU 在硬體電路中可以用大小比較器來實現。此外須考量 $x - b$ 的正負數關係，輸出正確的結果。

合成 RTL code 後，需進行 testbench 的測試，將 tb.txt 的值當作輸入，開啟 Synopsys 提供的 VCS 軟體進行 RTL code 的模擬，圖 48 為產生的波形圖。除了利用波型圖觀察電路時序是否正確外，會輸出 pattern.txt 作為後續驗證電路輸出結果，可將該文件與 golden.txt 作黃金測試，測試結果皆符合 Python 輸出結果，得以驗證電路的功能性。

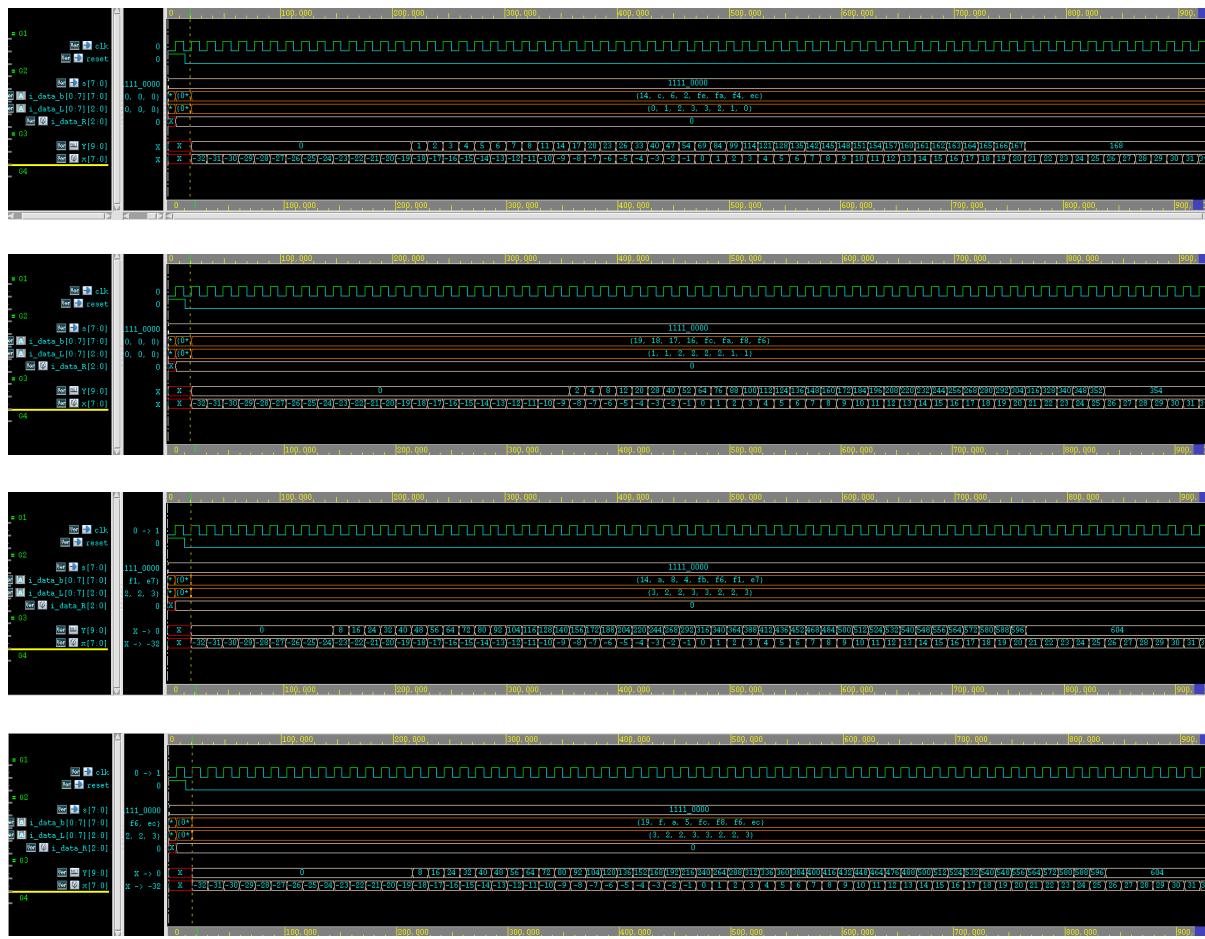


圖 48、四種激勵函數波形圖

完成 RTL code 編譯後，接著進行 Design complier 軟體合成標準元件，先使用 Python 腳本自動產生對應輸出入的 PAD，產生 CHIP.v 的 verilog 檔。在 Design complier 合成的過程中，使用 EDA 腳本自動合成電路，時間週期設為 15ns，約 66.67 MHz。

為了提高電路的效能和節省硬體資源，使用移位代替乘法，由於移位需要靠連接線來傳遞資料，此種設計方法會產生較多的連接線，如圖 49 所示。

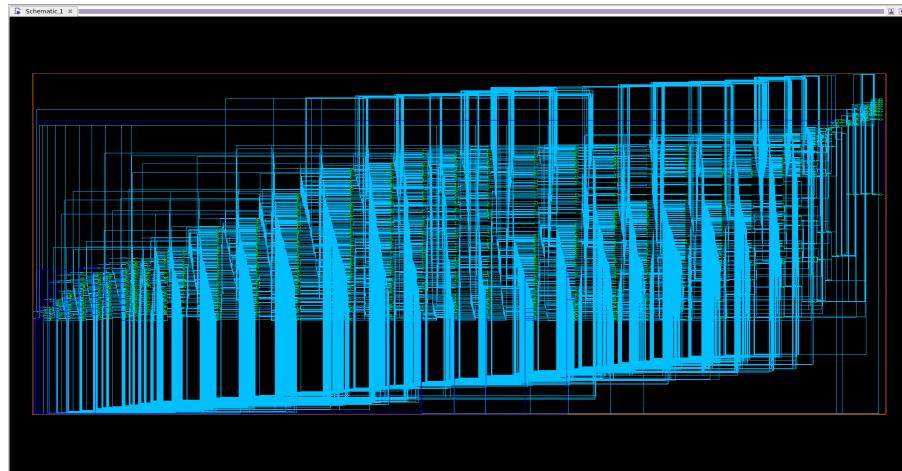


圖 49、SPINDLE 電路圖

完成 Design compiler 的電路合成後，將產生的 CHIP_syn.v 與原先 testbench 進行 VCS 的模擬，確認合成的電路功能是否正確，進行 Per-sim 的流程。

確認功能正確後，使用 Python 腳本自動生成晶片布局的腳位檔案，格式為 io.tdf，將作為 IC compiler 軟體的腳位設定。接著，使用 EDA 腳本完成晶片自動化布局及繞線，如圖 50 為完成的晶片布局圖。

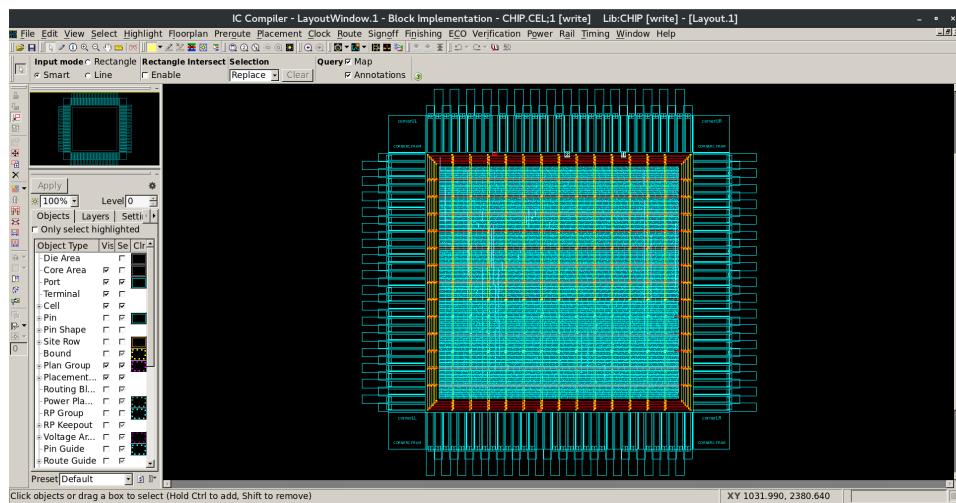


圖 50、SPINDLE 晶片布局圖

完成 IC compiler 的電路合成後，將產生的 CHIP_pr.v 與原先 testbench 進行 VCS 的模擬，確認最終晶片功能是否正確，進行 Post-sim 的流程。

完成晶片製作後，可以得到對應的面積、功耗與時間等數據，如表 5。需再進行驗證 DRC 以及 LVS，驗證正確即實現即線預訓練常態化器的設計，可以任意通過調整斜率和偏差值，進而完成近似任意激勵函數。

在此實驗中，原先預計希望可以透過演算法訓練微調圖形，但若後續給於的輸入與原始圖形差異過大，會因 ReLU 數量不足的原因，導致無法完全匹配原先預計的目標函數，雖然能進行微調，但受限於 ReLU 而無法提高訓練的準確度。

若要解決此問題就不能以單一筆資料進行訓練，而必須取樣部分資料集，才能解決發散的問題，因此我們提出離線預取樣方式來進行改善。

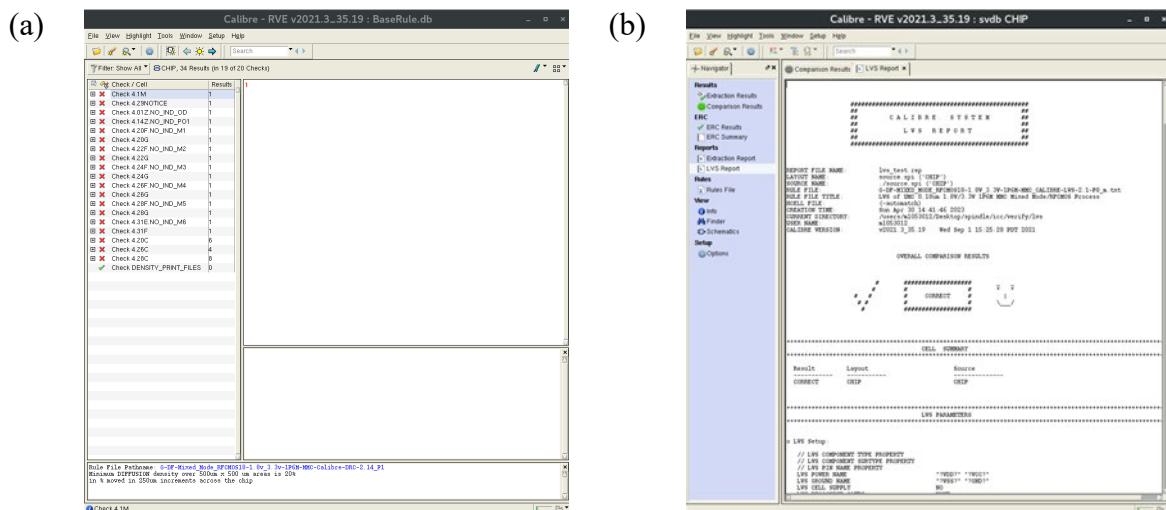


圖 51、(a) DRC 驗證 (b) LVS 驗證

表 5、整體電路之布局數據

	SPINDLE	
函數	Any kind of activation (Fine-Tune)	
階段	Per-sim	Post-sim
Power (mW)	4.4151	5.4151
Timing (ns)	15.24	16.13
Area (μm^2)	991848.318711	997419.837070

5.3 兩種取樣法量化效果

為了驗證使用離線取樣法能具有量化的優點，此處使用雙峰分布資料集做為測試，此資料及由兩種常態分布組成以及均勻分布的資料組成，如公式(5.2)所示，資料量總數為十萬筆資料。

$$f(x) = \begin{cases} N_1(-3, 1) \\ N_2(4, 0.5) \\ U(-10, 10) \end{cases} \quad (5.2)$$

其中，N 代表常態分布(Normal distribution)，-3 和 4 代表平均數，4 和 0.5 代表標準差。而 U 代表均勻分布(Uniform distribution)，-10 和 10 代表邊界，實際分布如圖 52(a)。將雙峰分布資料集做機率密度函數(PDF)和累積分布函數(CDF)，使用公式(3.1)和公式(3.2)進行計算，可繪製出圖 52(b)。

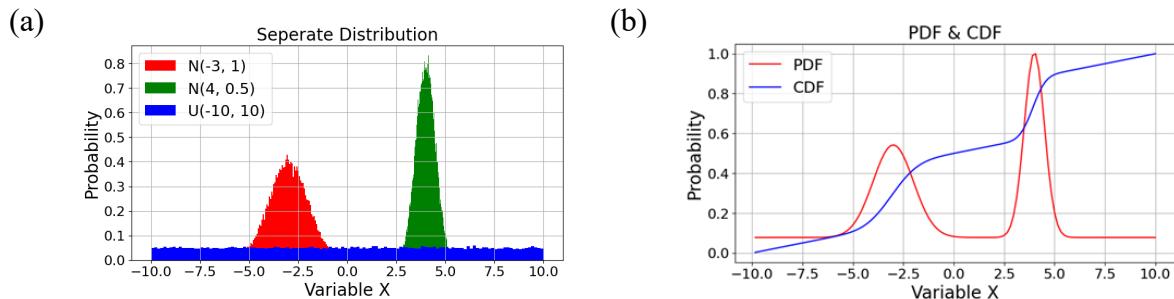


圖 52、(a)雙峰分布資料集 (b)雙峰分布資料集 PDF 和 CDF 函數

為了顯示預取樣能對資料集做量化，因此藉由 Python 中的 seaborn 套件，使用軸須圖(rug plot)來了解數據的分布，由於資料本身一共有十萬筆資料，無法全部顯示全部資料分布，因此在操作上使用等距抽樣(Systematic Sampling)方式來顯示軸需圖。

以原資料集 100000 筆資料做取樣，每隔 250 筆資料做取樣，可得 400 筆與原資料分布密度相同的子資料集，再將其轉化為軸需圖。

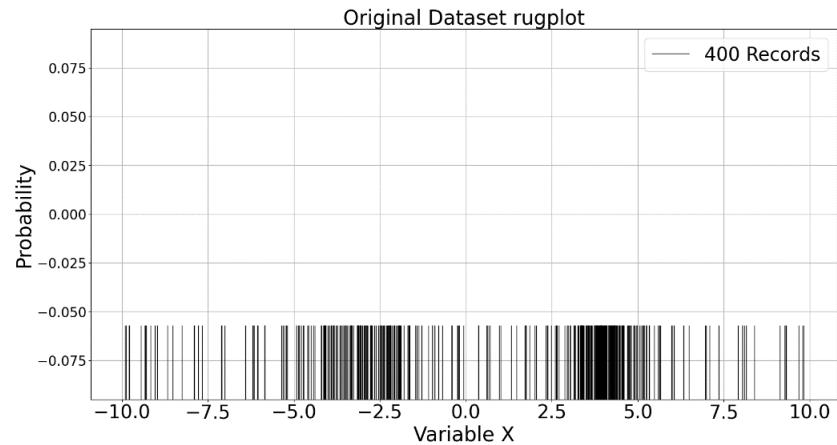


圖 53、雙峰分布資料集軸需圖

如圖 53 所示，可以發現資料會集中在兩峰值附近，由於本身為不均勻資料集，因此在軸需圖也能明顯觀測出疏密的差異。若是直接將資料當做神經網路的輸入，會因疏密關係的不一致，需要更多的位元數(bit)來存放資料，對神經網路解析資料的能力也會下降。

為了加速訓練和提升準確度，我們此處使用預取樣排序法來量化資料分布，此處使用排序法中的分箱數為 128 筆，依序填入後可得取樣排序後的軸需圖，如圖 54 所示。

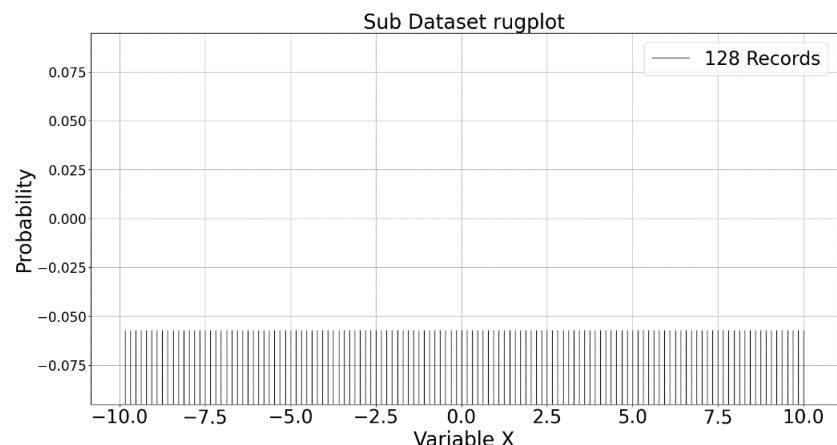


圖 54、預取樣排序法軸需圖

而經過預取樣排序法後，可以觀察取樣後產生的 CDF 圖形，具有量化的特性，且由於子資料集來自原資料集的預取樣，其 CDF 會與原資料集一致，由圖 55 可以觀察出 CDF 量化的特性。

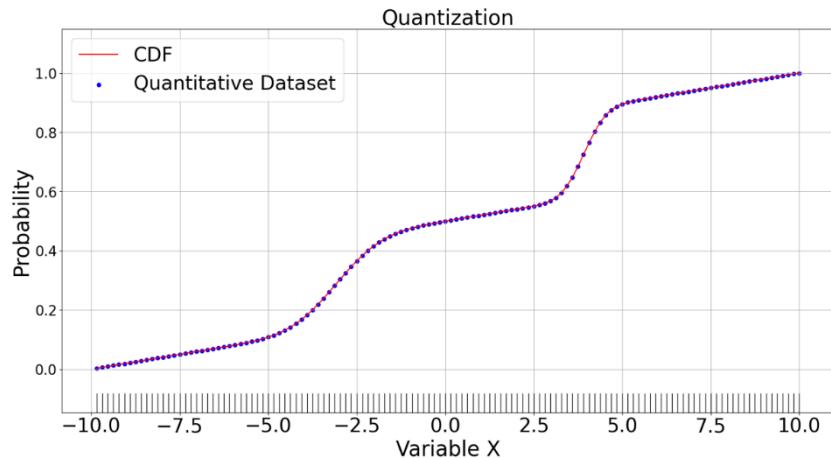


圖 55、預取樣排序法後產生的量化 CDF

除了使用預取樣排序法外，也可以使用逆轉換取樣法來隨機生成量化的 CDF。我們使用 Python 的 numpy 函式庫，生成了一隨機均勻分布的資料集，其數量為 2048 筆資料，範圍界於 0 到 1 之間。

為了驗證生成的隨機均勻資料集與原資料即的匹配程度，我們將其轉化為 PDF，做資料集的直方圖疊圖分析，如圖 56 所示，具有相似卻不相同的特性。

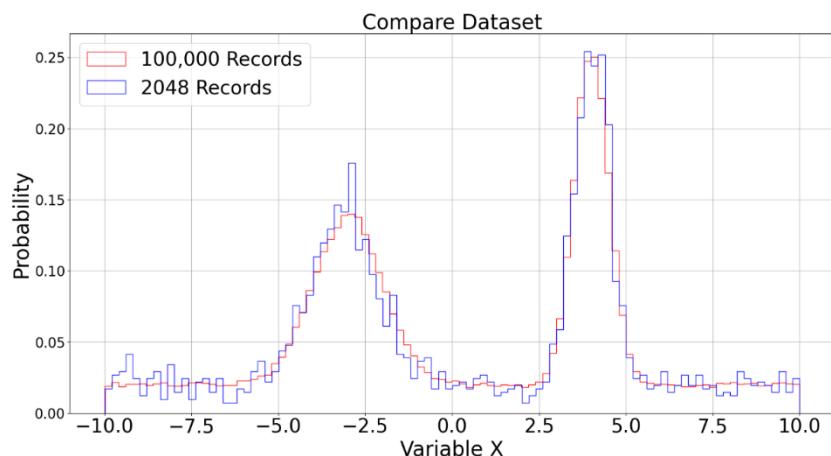


圖 56、逆轉換取樣生成隨機子資料集之 PDF 圖形

而對於隨機生成的均勻分布子資料集，我們比對其生成的 CDF 與原資料集的 CDF，誤差值能控制在 0.05~ -0.05 之間，其誤差值本身來自於隨機生成的資料集與原 CDF 的近似誤差，在可接受範圍內。

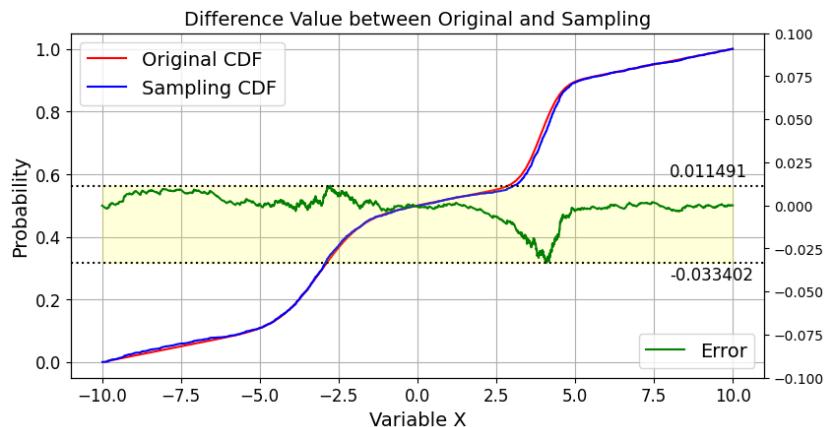


圖 57、取樣與原始 CDF 的誤差值疊圖

而在量化方面，由於子資料集本身是均勻分布，因此會呈現量化狀態，我們可由生成子資料集的軸需圖觀察出。由於資料量是 2048 筆，此處依舊使用等距抽樣，將 2048 筆資料以 32 為基底，轉化顯示為 64 筆資料的軸需圖，如圖 58。

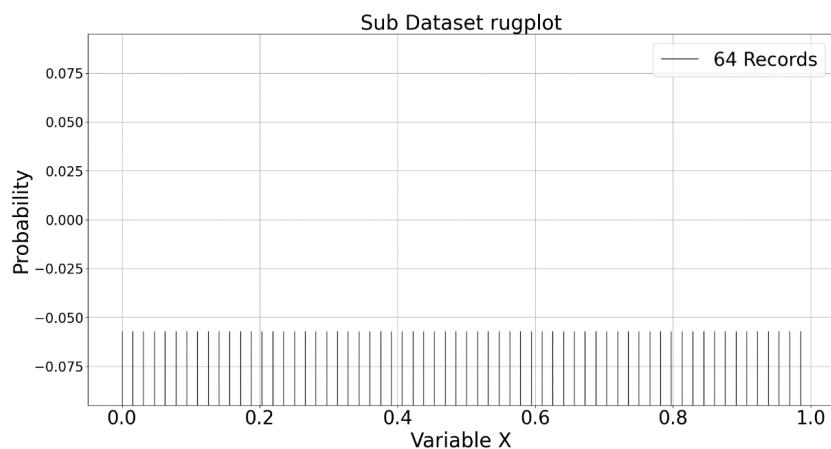


圖 58、逆轉換取樣法軸需圖

5.4 比較各類骨幹神經網路

為了證明在真實世界中，會存在具有多峰分布的特徵資料集，我們參考了文獻[2]，該文獻主要研究在兩種感測器所產生的 NIR 與 RGB 圖像，並說明兩者間具有雙峰分布的特性，因此本實驗將採用該文獻提供的 nirscene1 資料集作為驗證 KDE-SPINDLE 的貢獻。

nirscene1[2]資料集共有 954 張照片，由 477 個場景組成，共分為 9 類，分別為鄉村、土地、森林、室內、高山、舊建築、街道、城市、水景，如圖 59 所示。

由於原先資料集目的在於比較 NIR 與 RGB 圖像的關聯性，並無訓練測試集概念，為了測試訓練的準確度，將從原資料集拆分成訓練集 612 張和測試集 342 張照片。



圖 59、nirscene1[2]資料集

為了證明 KDE-SPINDLE 在骨幹神經網路上會有良好的貢獻，我們使用了三種知名的骨幹網路做為模型，分別為 AlexNet[10]、ResNet-18[11]、VGG-16[12]，比較原先骨幹模型訓練後的準確度，以及預取樣後調整激勵函數後，使用本論文提出的 KDE-SPINDLE 做改善。

表 6 是藉由預取樣後，進行三類模型的特徵提取。由於骨幹網路的差異，提取出的特徵分布並不會相同。進行擬合後，我們能得到特定模型的 KDE-SPINDLE，能做為新的激勵函數使用，如圖 60 所示，此為使用 VGG-16 模型產生的多峰態特徵曲線。

表 6、各種骨幹網路模型的特徵提取參數

Model	AlexNet[10]	ResNet-18[11]	VGG-16[12]
Dataset	nirscene1[2]	nirscene1[2]	nirscene1[2]
Kernel Unit	$y = h * \text{torch.sigmoid}(w * x - b)$		
Kernel Number	2	2	2
Variable Value	[0.2, 522.6, -26.6] [0.8, 788.3, -41.5]	[0.3, 79, -33.1] [0.7, 84.8, -32.7]	[0.8, 36.8, 2.3] [0.2, 51.7, 5.9]

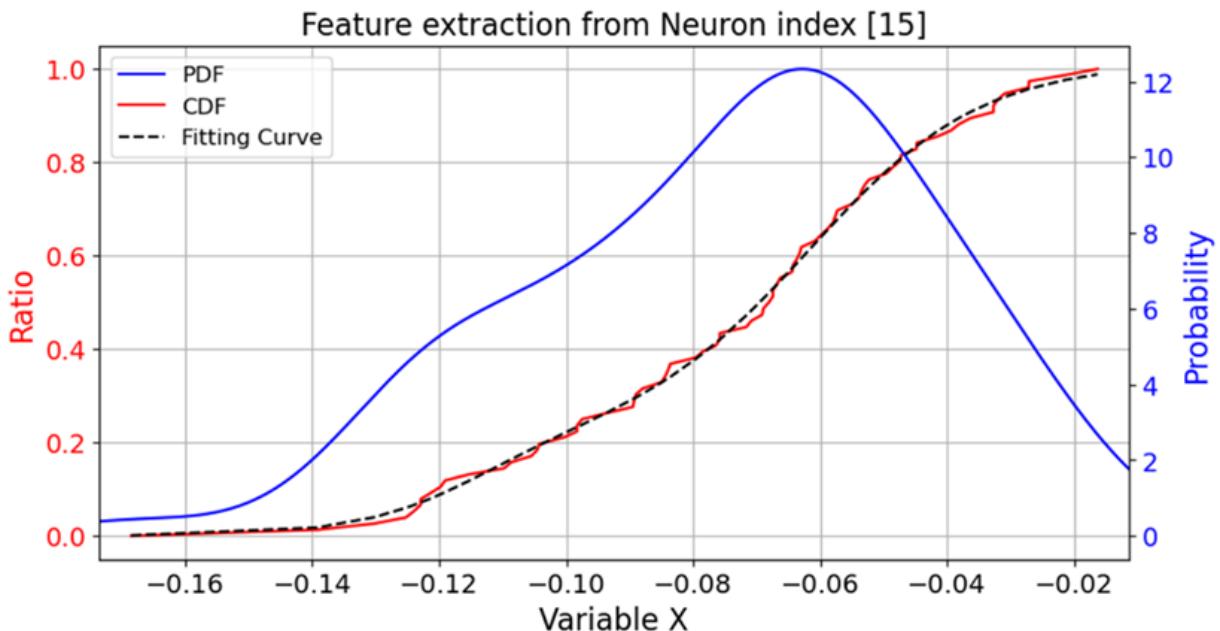


圖 60、VGG-16 特徵提取圖

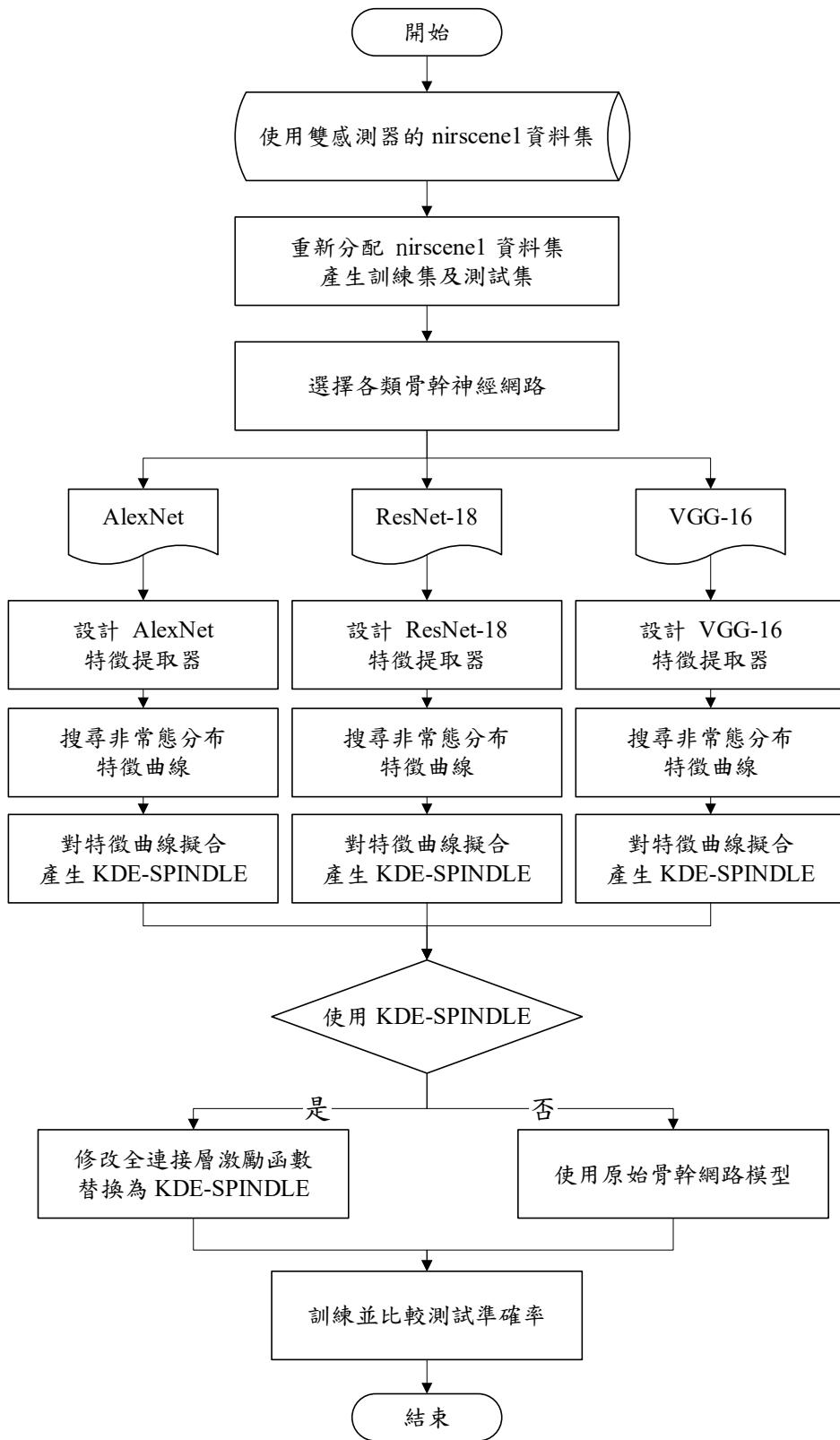


圖 61、各種骨幹網路比較實驗流程圖

在我們的實驗中，使用的 Python 版本為 3.8，GPU 為 NVIDIA Geforce RTX 3060 Ti，CPU 為 intel i9-10900X。神經網路皆是由 Pytorch 建構，版本為 1.7.1，並使用 Adam (Adaptive Moment Estimation)作為優化器，交叉熵(CrossEntropy)作為損失函數。骨幹網路會根據訓練集的類別數做全連接層調整，其餘部分皆與原始論文骨幹網路模型相同。

由於各神經網路設計的輸入大小不一致，本處使用 transforms.Resize 將圖片調整成該神經網路輸入的大小，表 7 是各類骨幹網路相關參數以及目標準確度。

表 7、各種骨幹網路模型準確度比較表

Dataset		nirscene1[2]					
Model		AlexNet[10]		ResNet-18[11]		VGG-16[12]	
		Orig.	Ours	Orig.	Ours	Orig.	Ours
Input	(C,H,W)	(3,256,256)		(3,224,224)		(3,224,224)	
Batch size		8		8		8	
Train / Test num.		612 / 342		612 / 342		612 / 342	
Layer		27		73		40	
Learning Rate		1e-5	1e-5	1e-4	5e-5	1e-5	1e-5
Epoch		500		100		100	
Total Params		11.24 M		44.69M		119.61 M	
Params size	(MB)	182.25		106.27		675.7	
Accuracy	(%)	38.89	42.4	41.8	42.4	66.67	71.92

由表 7 實驗結果可知，在各類骨幹網路中，使用 KDE-SPINDLE 皆有提升模型的測試準確度，在 AlexNet 骨幹模型上，準確度 38.89%，約有 3.51%的提升。在 ResNet-18 上表現較不理想，準確度 42.4%，約有 0.6%的提升，提升幅度較小。而在 VGG-16 骨幹網路模型上，準確度可以突破至 70%，準確度落在 72%，提升大約 5.56%的功效，驗證 SPINDLE 在各種骨幹神經網路中，皆能夠提升準確度。

會有準確度的差距是由於各種骨幹網路的層數不同，VGG-16 模型使用的參數量是 AlexNet 模型的幾十倍，因此 VGG-16 模型解析圖片的能力相較於其他兩種有著更高的測試準確度。

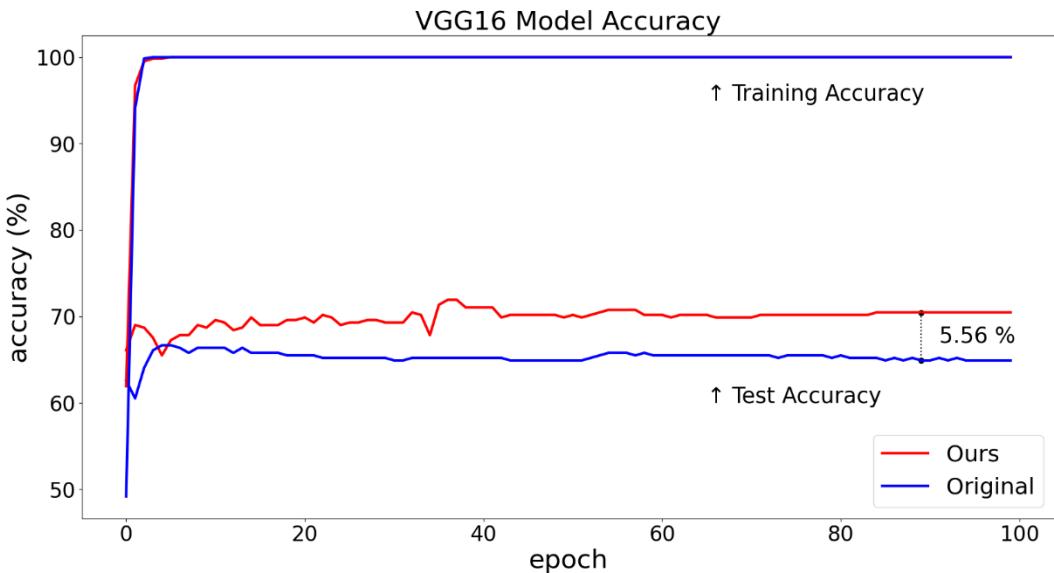


圖 62、VGG-16 模型準確度

在實驗的過程中，我們使用 Pytorch 內部提供的模型及預訓練參數，提升訓練速度。

根據圖 62 所示，VGG-16 模型在訓練達到十個 epochs 之前已經達到了 100% 的訓練準確度，因此可以認為模型已經趨近於收斂。然而，我們發現使用 SPINDLE 常態化器後，模型的測試準確度相較於原始模型能提升約 5.56%。

值得注意的是，我們僅僅通過調整分類器中的全連接層中的激勵函數，就能夠實現 5.56% 的準確度改善，而無需更動骨幹網路的骨幹模型。這表明對於深度學習而言，調整激勵函數能夠對模型性能產生顯著的影響。

然而，由於 nirscenel 資料集的貧乏，使得模型準確度受到一定的限制。拍攝 NIR 的感測器較為昂貴與冷門，目前開源的 NIR 與 RGB 所結合的資料集並不常見。為了有效解決資料集樣本數問題，我們使用另一種方式來產生非常態分布特徵資料集。

5.5 比較各類激勵函數

在前一章節提到受限於真實照片的樣本數不足，影響模型的準確度，因此找尋新的方式來產生非常態分布特徵資料集。PolyMNIST[27]是藉由 MNIST 為基礎，替換背景黑色底圖，藉此達到多模態(Multimodal)的資料集生成。

使用此資料集的優點在於可以改善樣本數不足的情形，並以此資料集的多模態特性，驗證使用 KDE-SPINDLE 作為激勵函數，模型準確率能夠優於各種激勵函數。

生成 PolyMNIST 方式是藉由三種不同的底圖，分別為糖果、火焰、海洋，做為此實驗的資料集，在數字部分進行圖像對比處理，產生後資料集與原 MNIST 進行合併，模擬出類似 NIR+RGB 效果的三種新資料集，如圖 63 所示。

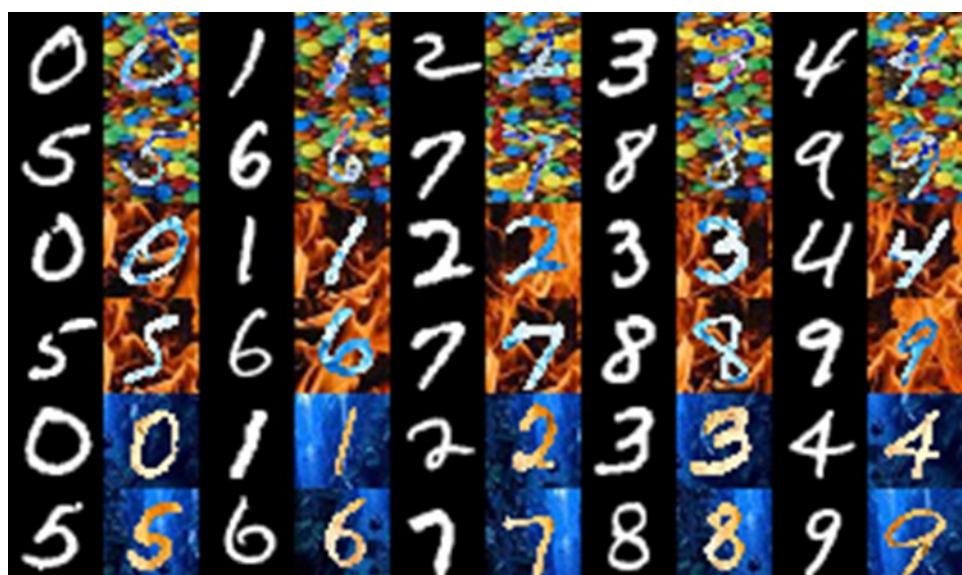


圖 63、自定義 PolyMNIST 資料集

產生 PolyMNIST 資料集後，會進行非常態分布的特徵提取，由於最佳的 MNIST 模型是 LeNet-5[9]，因此選擇此模型做為此實驗的骨幹網路模型。

將產生的三種 PolyMNIST 進行取樣後進行特徵提取，與先前實驗不同處在於，此實驗目的是比較各種激勵函數對模型準確度的差異，因此使用同一種模型。

相關提取參數如表 8 所示，提取環境與先前實驗相同。圖 64 則為糖果底圖的 M0-MNIST，其特徵分布為顯著的雙峰分布，能夠驗證 SPINDLE 常態化器在非常態特徵分布的資料集上具有良好表現。

表 8、各種 PolyMNIST 資料集的特徵提取參數

Model	LeNet-5[9]		
Dataset	M0-MNIST	M1-MNIST	M2-MNIST
Background	Candy	Fire	Sea
Kernel Unit	$y = h * \text{torch.sigmoid}(w * x - b)$		
Kernel Number	2	3	2
Variable Value	[0.5, 522.6, -26.6] [0.8, 788.3, -41.5]	[0.2, 173.1, -17.9] [0.5, 138, -12.9] [0.3 188.5 -15.9]	[0.5, 185.3, -23.3] [0.5, 121, -16.2]

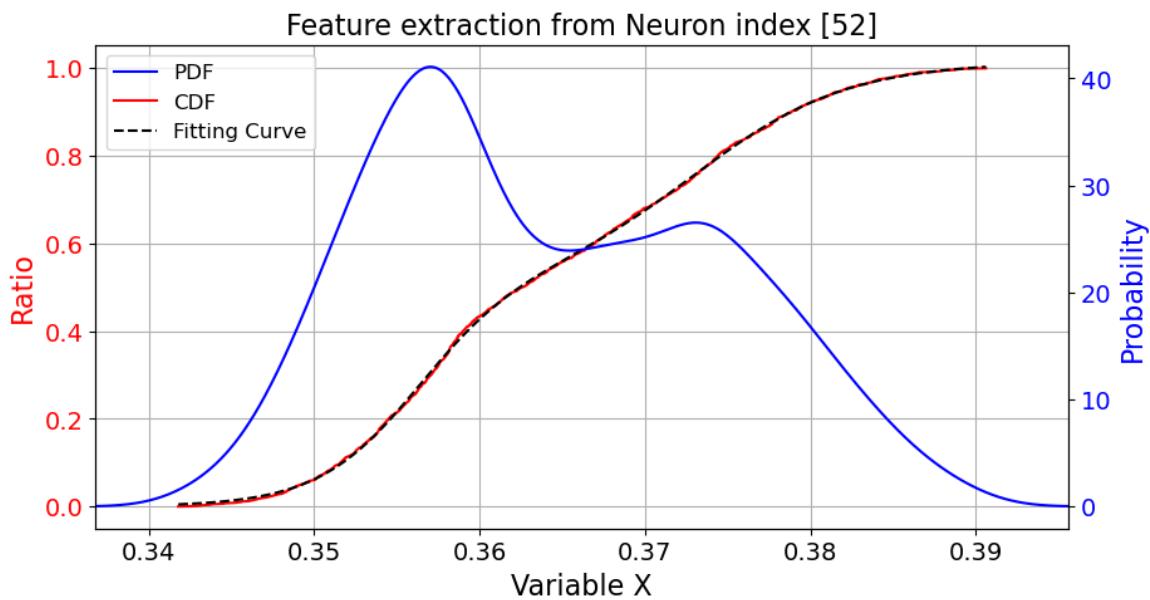


圖 64、M0-MNIST 特徵提取圖

表 9 為模型初始設定值。為了加速神經網路的訓練，將模型批次大小設為 256，代表抓取 256 個樣本後再進行反向傳播，以提升訓練的效率。

此外，須注意模型的通道(channel)應做調整，在原先模型中，MNIST 屬於單色(monochrome)圖片，因此原始模型通道為 1，但 PolyMNIST 屬於彩色圖像，神經網路模型的通道數必須調整為 3，因此會對 LeNet-5[9]的網路模型的骨幹部分做些微修正，但整體層數和架構與原論文並無過多更動，確保實驗比較的公平性。

表 9、各種 PolyMNIST 資料集參數

Dataset		M0-MNIST	M1-MNIST	M2-MNIST
Model		LeNet-5[9]	LeNet-5[9]	LeNet-5[9]
Input	(C,H,W)	(3,28,28)	(3,28,28)	(3,28,28)
Batch size		256	256	256
Train /Test Number		120K/10K	120K/10K	120K/10K
Layer		11	11	11
Learning Rate		1e-3	1e-3	1e-3
Epoch		50	50	50
Total Params		62,006	62,006	62,006
Params size	(MB)	0.24	0.24	0.24

為了驗證 KDE-SPINDLE 為最佳激勵函數，因此會選用不同的激勵函數做為比較對象，可分為兩類，共五種激勵函數。第一類為常見的激勵函數，選用 ReLU 和 Sigmoid 做為比較對象。第二類為骨幹神經網路常用的激勵函數，分別為 Yolo[19]常用的 Mish、Leaky ReLU 激勵函數和 Transformer[17]常用的 GeLU[21]激勵函數。最後，使用我們設計的 SPINDLE 常態化器，先進行預取樣及特徵提取後，進行 CDF 函數的擬合，產生的 KDE-SPINDLE 非常態化器，即為可調整形狀之激勵函數，詳細實驗流程圖如圖 65。

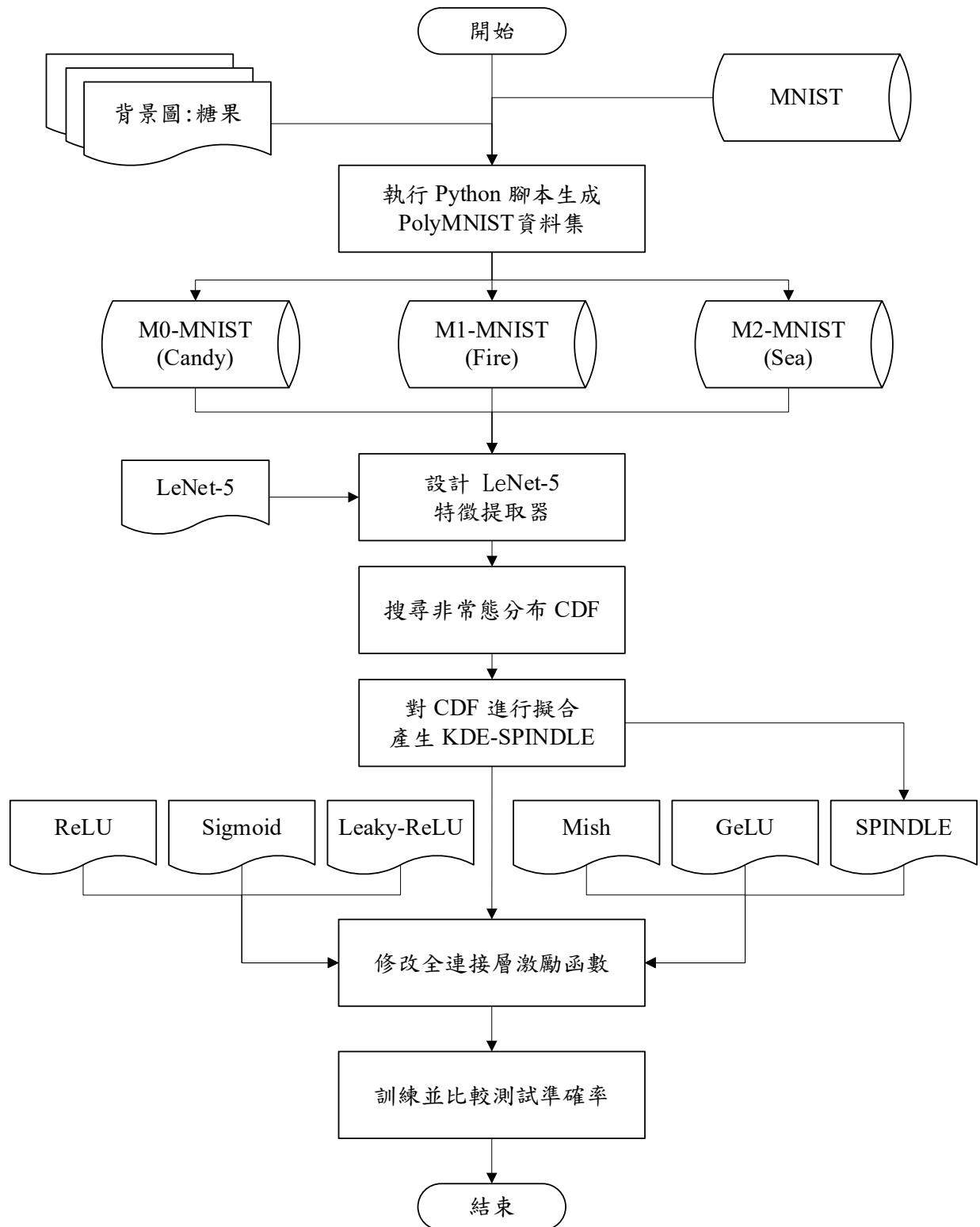


圖 65、各種激勵函數比較實驗流程圖

表 10、各種激勵函數準確度比較表

Kind		Base		Backbone			Ours
Activation		ReLU [13]	Sigmoid [15]	L-ReLU [19]	Mish [19]	GeLU [21]	SPINDLE
M0	Accuracy	97.24	97.09	97.22	97.14	96.99	97.25
	Inaccuracy	2.76	2.91	2.78	2.86	3.01	2.75
M1	Accuracy	98.65	98.71	98.7	98.75	98.61	98.79
	Inaccuracy	1.35	1.29	1.3	1.25	1.39	1.21
M2	Accuracy	98.89	98.73	98.78	98.82	98.83	98.85
	Inaccuracy	1.11	1.27	1.22	1.18	1.17	1.15

表 10 為實驗結果，在各類型激勵函數的測試下，準確度皆受惠於 LeNet-5 模型的良好設計，皆由高達約 97% 到 99% 的準確度。在三種資料集中，M0-MNIST 與原始 MNIST 差異最大，因此測試準確度其他兩者更低，三種資料集的資料複雜度由 M0-MNIST 最高，其次是 M1-MNIST，差異最小的則是 M2-MNIST。

而經過實驗可知，我們僅以預取樣的方式，提取並修改單一層神經網路的激勵函數，將其就地替換為 SPINDLE 常態化器，在測試中皆有良好的表現，相較於其他種類激勵函數，使用 SPINDLE 常態化器的準確度可保持在前兩名，在已經有極高準確度的情況下，我們仍然能夠進一步提高性能。SPINDLE 常態化器可以讓模型在更高的精度下運作，從而使得整個系統更加可靠。

SPINDLE 常態化器在減少模型不準確度方面發揮了重要作用。不準確度是指模型在預測時所犯的錯誤。這些錯誤可能會導致錯誤的結果，並且對實際應用造成嚴重的後果。因此，減少模型的不準確度對於提高模型性能非常重要。

在這種情況下，使用 SPINDLE 常態化器可以降低神經網路模型的測試不準確度，如圖 66 所示。

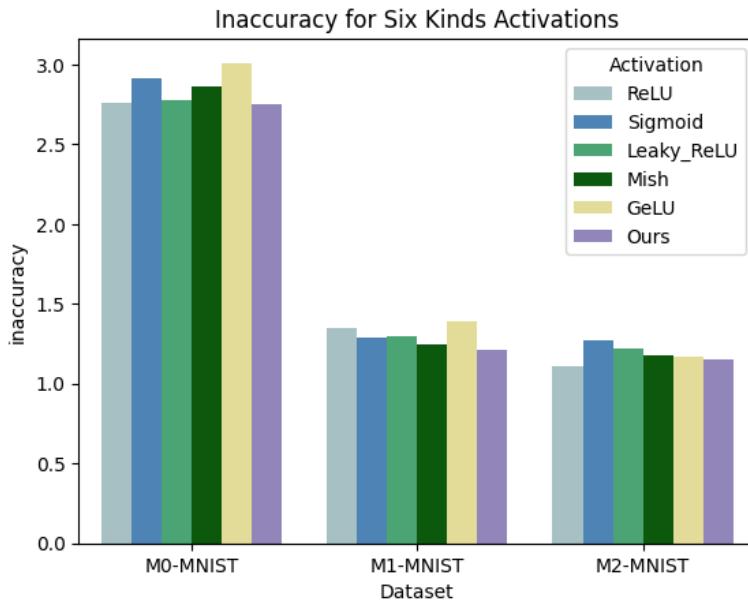


圖 66、各種激勵函數不準確度長條圖

在圖 66 中，可以觀察出 SPINDLE 有較低的不準確度，統計三種特殊分布的資料集中，各激勵函數的表現，以我們所提出的 SPINDLE 常態化器最為優秀，排名第二的 ReLU 雖然在 M0-MNIST 及 M2-MNIST 的資料集上也有不錯的表現，但在 M1-MNIST 的資料集上表現卻不太理想，相較於使用 SPINDLE 常態化器，ReLU 的穩定度較差。

基於這些結果，可以得出以下結論：SPINDLE 常態化器產生的激勵函數，是深度學習中最佳的激勵函數，並且適用於各種學習目標，尤其在非常態分布的資料輸入中，有著最佳表現。

表 11、各種激勵函數名次表

Kind		Base		Backbone			Ours
Activation		ReLU [13]	Sigmoid [15]	L-ReLU [19]	Mish [19]	GeLU [21]	SPINDLE
M0	Rank	II	V	III	IV	VI	I
M1	Rank	V	III	IV	II	VI	I
M2	Rank	I	VI	V	IV	III	II
Total Rank		II	V	IV	III	VI	I

第六章 結論與未來展望

6.1 結論

我們在提出了即線預訓練常態化器，使用二分輕數斜率分段線性搜尋法(BLS-PWL)來解決原先 LS-PWL 演算法[25]在斜率變化處較大處無法有效產生合適線段的問題，並提出 SPINDLE 作為激勵函數，能夠微調為任意激勵函數。為了改善即線微調容易發散的問題，我們提出使用離線預取樣的方式，使用預取樣排序法和逆轉換取樣法，皆能達到量化累積分布函數(CDF)。而在 KDE-SPINDLE 驗證上，在 AlexNet[10]、ResNet[11]、VGG-16[12]皆在 nirscene1[2]資料集上能提升準確度至 72%。受限於真實資料集的不足，為追求高準確度，我們使用 MNIST 為基礎產生的新 PolyMNIST[27]資料集，進行驗證，在不同複雜度的資料集下，SPINDLE 能上升到 99% 的準確度，若是須應用在低錯誤的目標下，我們設計的 SPINDLE 將會是最佳的選擇。

6.2 未來展望

在骨幹網路應用層面，已經證實只需使用我們提出的離線預取樣就地常態化器，就能有效提升模型的準確度。但在研究過程中，即線微調就地常態化器在經過訓練後容易發散，期待後續的研究能有效解決發散問題。

参考文献

- [1] M. Brown and S. Süsstrunk, "Multi-spectral SIFT for scene category recognition," *CVPR 2011*, Colorado Springs, CO, USA, 2011, pp. 177-184, doi: 10.1109/CVPR.2011.5995637.
- [2] Martin Kiechle, Tim Habigt, Simon Hawe, Martin Kleinsteuber, "A Bimodal Co-Sparse Analysis Model for Image Processing," in *Computer Vision and Pattern Recognition*, Jun 2014, arXiv:1406.6538.
- [3] R. M. Hristev, "The ANN Book, " *Released under the GNU general public license*, 1999, pp.129-170.
- [4] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks, " *arXiv preprint*, 2015, arXiv:1511.08458.
- [5] Image Classification on ImageNet, URL: <https://paperswithcode.com/sota/image-classification-on-imagenet>
- [6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors, " *In Nature*, 1986, pp.323, 533–536.
- [7] ImageNet Common Objects in Context, URL: <https://www.image-net.org/>
- [8] COCO: Common Objects in Context, URL: <https://cocodataset.org/>
- [9] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.
- [10] Krizhevsky A, Sutskever I and Hinton GE, "ImageNet Classification with Deep Convolutional Neural Networks, " in *Neural Information Processing Systems 25 (NIPS 2012)*, Communications of the ACM, 60(6), 84–90 (2017). <https://doi.org/10.1145/3065386>.
- [11] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas,

NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

- [12] Simonyan, K. and Zisserman, A. "Very Deep Convolutional Networks for Large-Scale Image Recognition," *in 3rd International Conference on Learning Representations (ICLR2015)*, arXiv:1409.1556.
- [13] V. Nair, and G. Hinton, "Rectified linear units improve restricted boltzmann machines, " *in 27th International Conference on Machine Learning(ICML 2010)*, June 2010, page 807--814.
- [14] W. G. Teich, A. Engelhart, W. Schlecker, R. Gessler and H.-J. Pfleiderer, "Towards an efficient hardware implementation of recurrent neural network based multiuser detection," *in 2000 IEEE Sixth International Symposium on Spread Spectrum Techniques and Applications(ISSTA 2000)*. Proceedings (Cat. No.00TH8536), Parsippany, NJ, USA, 2000, pp. 662-665 vol.2, doi: 10.1109/ISSSTA.2000.876516.
- [15] B. L. Kalman and S. C. Kwasny, "Why tanh: choosing a sigmoidal function, " *in International Joint Conference on Neural Networks(IJCNN)*, Baltimore, MD, USA, 1992, pp. 578-581 vol.4, doi: 10.1109/IJCNN.1992.227257.
- [16] Vaswani, A., et al., "Attention is all you need, " *Proceedings of the Neural Information Processing Systems(NIPS)*, Long Beach, CA, USA, 2017, arXiv:1706.03762.
- [17] Dosovitskiy, A., et al., "An image is worth 16x16 words: transformers for image recognition at scale," *arXiv preprint*, 2020, arXiv:2010.11929.
- [18] Liu, Z., et al., "Swin transformer: hierarchical vision transformer using shifted windows, " *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, Canada, 2021, pp. 9992-1002, doi: 10.1109/ICCV48922.2021.00986.
- [19] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and Pattern Recognition (CVPR)*, Jun 2015, arXiv:1506.02640.

- [20] Liu, W., et al., "SSD: Single Shot MultiBox Detector," *in Proceedings of the 14th European Conference on Computer Vision (ECCV)*, 2016, arXiv:1512.02325.
- [21] Dan Hendrycks and Kevin Gimpel, "Gaussian error linear units," *arXiv preprint*, 2016, arXiv:1606.08415.
- [22] K. Leboeuf, A. H. Namin, R. Muscedere, H. Wu, and M. Ahmadi, "High speed VLSI implementation of the hyperbolic tangent sigmoid function," *in 2008 Third International Conference on Convergence and Hybrid Information Technology*, Busan, 2008, pp. 1070-1073.
- [23] R. Muscedere and K. Leboeuf, "A dynamic address decode circuit for implementing range addressable look-up tables," *in 2008 IEEE International Symposium on Circuits and Systems*, Seattle, WA, USA, 2008, pp. 3326-3329, doi: 10.1109/ISCAS.2008.4542170.
- [24] B. Zamanlooy and M. Mirhassani, "Efficient VLSI Implementation of Neural Networks With Hyperbolic Tangent Activation Function," *in IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 39-48, Jan. 2014, doi: 10.1109/TVLSI.2012.2232321.
- [25] T. -Y. Chen, C. -D. Tsai, H. -W. Fu, Y. -C. Yang and T. -C. Huang, "Error Correctable Range-Addressable Lookup for Activation and Quantization in AI Automotive Electronics," *2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, Penghu, Taiwan, 2021, pp. 1-2, doi: 10.1109/ICCE-TW52618.2021.9603030.
- [26] Justel, A., Peña, D. and Zamar, R. "A multivariate Kolmogorov-Smirnov test of goodness of fit," *Statistics & Probability Letters*, Published in 1997, 35(3), 251-259.
- [27] Sutter, T.M., Daunhawer, I., and Vogt, J.E, "Generalized Multimodal ELBO," *arXiv preprint*, 2021, arXiv:2105.02470.

作者簡歷

1. 基本資訊

姓名	中文	柯竣鑫	性別	男		
	英文	Jyun-Xin Ke	籍貫	彰化縣		
e-mail	s431190@gmail.com					
興趣	系統研究、腳本設計、閱讀					

2. 學歷

畢業學校	系所/組別	學位	起訖年月
國立彰化師範大學	電子所/SoC 組	碩士	110.07~112.06
國立彰化師範大學	電子工程學系	學士	106.09~110.06

3. 碩士期間發表論文

- [1]. J. Ke, S. Ciou, T. Chen and T. Huang, "SPINDLE: Self-Pretrainable In-situ Normalizer for Deep Learning Error Function," *2022 IEEE International Conference on Consumer Electronics - Taiwan*, Taipei, Taiwan, 2022, pp. 59-60, doi: 10.1109/ICCE-Taiwan55306.2022.9869099.