

南華大學科技學院資訊管理學系

碩士論文


Department of Information Management

College of Science and Technology

Nanhua University

Master Thesis

運用直交表因素配置進行循環神經網路之最佳化研究
Using the Factor Configuration of an Orthogonal Array to
Optimize a Recurrent Neural Network



郭育坤

Yu-Kuen Guo

指導教授：陸海文 博士

Advisor: Hai-Wen Lu, Ph.D.

中華民國 111 年 6 月

June 2022

南華大學
科技學院資訊管理學系
碩士學位論文

運用直交表因素配置進行循環神經網路之最佳化研究
Using the Factor Configuration of an Orthogonal Array to
Optimize a Recurrent Neural Network

研究生：郭育坤

經考試合格特此證明

口試委員：劉振益
黃武隆
陸海文

指導教授：陸海文

系主任(所長)：陳信良

口試日期：中華民國 111 年 6 月 11 日

南華大學資訊管理學系碩士論文著作財產權同意書

立書人：郭育坤 之碩士畢業論文

中文題目：運用直交表因素配置進行循環神經網路之最佳化研究

英文題目：Using the Factor Configuration of an Orthogonal Array to Optimize a Recurrent Neural Network

指導教授：陸海文 博士

學生與指導老師就本篇論文內容及資料其著作財產權歸屬如下：

- 共同享有著作權
- 共同享有著作權，學生願「拋棄」著作財產權
- 學生獨自享有著作財產權

學生：郭育坤 (請親自簽名)

指導老師：陸海文 (請親自簽名)

中華民國 111 年 6 月 15 日

南華大學碩士班研究生
論文指導教授推薦函

資訊管理學系碩士班 郭育坤 君所提之

論文

運用直交表因素配置進行循環神經網路之最佳
化研究

係由本人指導撰述，同意提付審查。

指導教授

陸梅文

111年6月15日

謝 誌

轉眼間碩專班的研究生涯也到了尾聲，雖然曾經一度想放棄，但在瓊誼學長與金旺學長的鼓勵下，總算順利完成學業。

首先特別感謝指導老師 陸海文教授在百忙之中抽空悉心指導，並適當的給予建議，使此篇論文得以完成。另外，非常感謝口試委員翁振益老師與黃武隆老師的鼓勵與疏漏處之指正，使得本論文更臻於完善。

感謝在研究所求學路上，陸海文老師、王昌斌老師、陳宗義老師、王佳文老師及尤國任老師用心的授課，課堂上所學讓我受益匪淺。另外，也謝謝嘉文哥、靜姊、秋麗姊、誠哥、繡方姊、宜婷姊、宜蔚姊、凡綜學長、小蘇學長、麟麟與小菁等同學的陪伴，使我的求學路上並不孤單。

最後，特別感謝我的老婆奕昀及家人，在我工作之餘幫忙照顧小孩，讓我可以無後顧之憂地完成研究所學業。

郭育坤 謹誌

中華民國 111 年 6 月

運用直交表因素配置進行循環神經網路之最佳化研究

學生：郭育坤

指導教授：陸海文 博士

南 華 大 學 資 訊 管 理 學 系 碩 士 班

摘 要

辨識系統在現今社會已成為不可或缺的一部分，從早期 AI 人工智慧的發展，到如今深度學習成為主流。科技的發展一日千里，而深度學習的運算卻日漸複雜，但相對來說，辨識能力也隨著深度學習技術更加純熟而提升。

本研究利用卷積神經網路 (Convolutional Neural Network, CNN)，作為特徵提取並結合長短期記憶模型 (Long Short-term memory, LSTM) 來預測，這樣的模型為循環神經網路 (Recurrent Neural Network, RNN) 中的一種。將此模型放在 Google 雲端硬碟裡的 Colaboratory 環境下執行，並搭配 GPU 加速器加速程式碼的運行速度，以田口品質工程實驗設計探討影響準確度的控制因子和水準，其中因子包含：卷積層激勵函數、池化層大小、輸出層激勵函數、Loss 損失函數、Optimizer 優化函數及 Metrics 評估準確率方法等6項。本實驗選擇 L_{12} 直交表進行實驗，並將實驗結果計算 S/N 比、平均值反應圖及反應表，以期得到最佳化參數配置，本研究結果經確認實驗得知卷積層激勵函數 elu、池化層大小3x3、輸出層激勵函數 tanh、Loss 損失函數 binary_crossentropy、Optimizer 優化函數 adam、Metrics 評估準確率方法 accuracy 為最佳化參數組合。

關鍵詞：卷積神經網路、循環神經網路、田口品質工程、最佳化參數



Using the Factor Configuration of an Orthogonal Array to Optimize a Recurrent Neural Network

Student: Yu-Kuen Guo

Advisor: Hai-Wen Lu, Ph.D.

Department of Information Management, College of Science and Technology,
Nanhua University

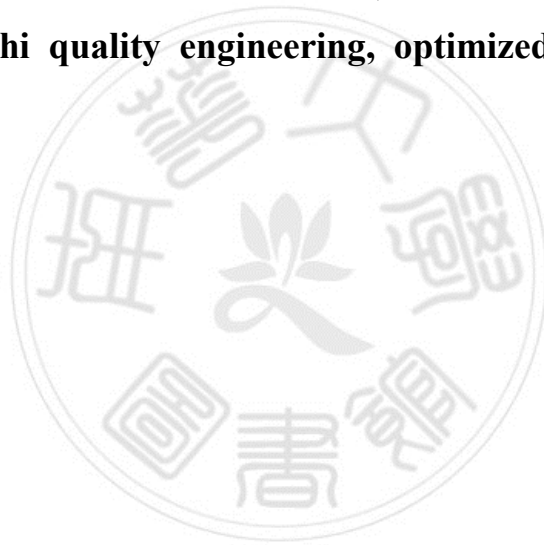
ABSTRACT

Recognition systems have become an indispensable part of Contemporary society. As the technology advances rapidly from the initial development of artificial intelligence to deep learning becoming the mainstream technology today, the complexity of deep learning computation has increased, and with the increasing complexity, its recognition capacity has also improved.

This study employed convolutional neural network (CNN) for feature extraction and, in conjunction with long short-term memory (LSTM), for predictions. The proposed model, which is a recurrent neural network, was executed in Google Colaboratory, and a graphic processing unit accelerator was used to accelerate the code execution. The Taguchi experimental design for quality engineering was used to identify factors and levels that affected accuracy. These factors were the activation function at the convolutional layer, the size of the pooling layer, the activation function at the output layer, the loss function, the optimizer function, and the accuracy evaluation metrics. This study used the L_{12} orthogonal array for its experiment, calculated the signal–noise ratio of the

experimental result, and produced a response graph and table for the mean signal–noise ratio. The aim was to optimize the parameter configuration. According to the study results, the optimal parameter combination was as follows: activation function at the convolutional layer is elu; pooling layer size is 3×3 ; activation function at the output layer is tanh; loss function is binary_crossentropy; optimizer function is Adam; and accuracy evaluation metrics is accuracy.

Keywords: convolutional neural network, recurrent neural network, Taguchi quality engineering, optimized parameters



目 錄

碩士論文著作財產權同意書.....	I
指導教授推薦函.....	II
謝誌.....	III
摘要.....	IV
ABSTRACT.....	VI
目錄.....	VIII
表目錄.....	IX
圖目錄.....	X
第一章 緒論.....	1
第一節 研究背景與動機.....	1
第二節 研究目的.....	1
第三節 研究架構與流程.....	2
第二章 文獻探討.....	4
第一節 田口式品質工程.....	4
第二節 機器學習與深度學習.....	8
第三章 研究方法.....	11
第四章 實驗結果分析.....	28
第五章 結論.....	33
參考文獻.....	34
一、中文文獻.....	34
二、西文文獻.....	36

表 目 錄

表2-1 $L_{12} (2^{11})$ 直交表.....	5
表3-1 控制因子與水準.....	12
表3-2 循環神經網路最佳化實驗 $L_{12} (2^{11})$ 直交表配置.....	18
表3-3 行的合併.....	19
表3-4 $L_{12} (2^{11})$ 新直交表.....	20
表3-5 循環神經網路最佳化實驗步驟.....	26
表3-6 循環神經網路實驗結果紀錄表.....	27
表4-1 循環神經網路實驗結果之平均數.....	28
表4-2 循環神經網路最佳化實驗 S/N 比.....	29
表4-3 循環神經網路最佳化因子平均值反應表.....	29
表4-4 循環神經網路最佳化因子 S/N 比反應表.....	30
表4-5 確認實驗結果.....	32

圖目錄

圖1-1 循環神經網路最佳化實驗研究流程.....	3
圖2-1 深度學習為機器學習中的子集合.....	9
圖3-1 循環神經網路最佳化實驗步驟.....	11
圖3-2 Relu 函數.....	13
圖3-3 Tanh 函數.....	14
圖3-4 Elu 函數.....	14
圖3-5 Selu 函數.....	15
圖3-6 生成幀序列的示例.....	21
圖3-7 生成幀序列的示例.....	22
圖3-8 生成隨機影像的新函式.....	23
圖3-9 定義 CNN 及 LSTM 模型.....	24
圖3-10 生成100個新的隨機序列並估計準確度.....	25
圖3-11 程式碼執行結果.....	25
圖4-1 循環神經網路最佳化因子 S/N 比反應圖.....	30

第一章 緒論

第一節 研究背景與動機

隨著科技高度發展，「機器學習」(Machine Learning, ML)與「深度學習」(Deep Learning, DL)等關鍵技術促成了人工智慧的爆炸性成長與應用，使得影像辨識技術更加成熟。目前影像辨識除了人臉辨識外，還有指紋辨識、車牌辨識等，甚至於醫療圖像辨識更幫助醫生更加準確地判斷病因，讓病人能夠得到最好的治療。然而，由於深度學習卻存在著需要大量的訓練資料，以及花費相當長的訓練時間等缺點。

故本研究以一個 CNN+LSTM 的循環神經網路模型為例，透過 Google Colaboratory 在圖形處理器 (Graphics Processing Unit, GPU) 環境下，配合田口品質工程實驗，以最低的成本與時間找出每一層的最佳化因素配置。

第二節 研究目的

本研究將利用田口品質工程中直交表因素配置去探討循環神經網路之最佳化條件，以 L_{12} 直交表設計實驗，希望以最少實驗次數，求得準確率最高的參數，最後由確認實驗驗證是否為最佳化組合。

由於循環神經網路的訓練資料量較大，因此較為費時，故本研究選擇運用 Google Colaboratory 配合 GPU 加速來縮短訓練時間，並探討在影像辨識的循環神經網路模型中影響品質特徵值的因子與水準，以適當的直交表進行設計實驗，將實驗結果計算出平均數及 S/N 比，選出最佳化的參數組合，最後再以確認試驗驗證。

第三節 研究架構與流程

本論文是以研究 CNN+LSTM 模型的循環神經網路之最佳參數設定，研究流程如圖 1-1 所示，論文架構如下：

第一章 緒論

描述本論文的研究背景及動機、研究目的，並說明論文的架構與流程。

第二章 文獻探討

淺談機器學習與深度學習、介紹田口品質工程及整理近年運用田口方法進行實驗設計的相關文獻。

第三章 研究方法

列出循環神經網路實驗的因子與水準，依照其自由度選定適當直交表進行實驗設計，並紀錄實驗結果。

第四章 實驗結果分析

根據第三章之實驗結果計算其平均數並以望大特性計算 S/N 比，求得最佳化參數設定，最後再以確認實驗作為驗證。

第五章 結論

分析探討第四章之成果並做出總結。

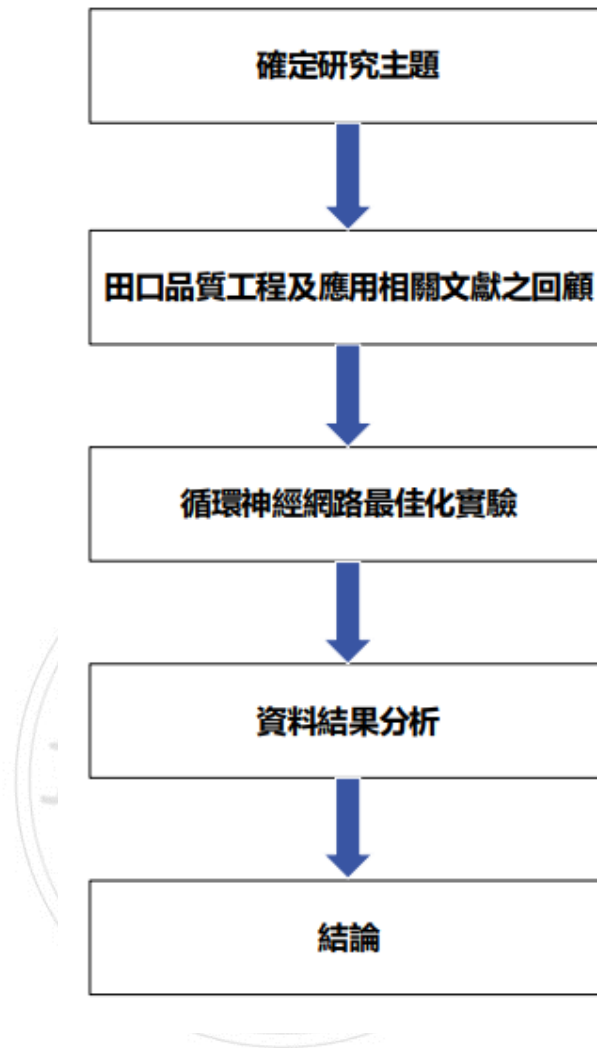


圖1-1 循環神經網路最佳化實驗研究流程

第二章 文獻探討

本章研究目的進行文獻蒐集，經過彙整作為理論分析基礎，分別對田口式品質工程、機器學習及深度學習進文獻行探討如下。

第一節 田口式品質工程

2.1.1 簡介

田口式品質工程 (Taguchi Quality Engineering)，也被稱作是田口式穩健設計方法 (Taguchi Methods of Robust Design) (資料來源：維基百科)。是由日本田口玄一博士所提出的一套試驗設計法，主要依設計參數 (控制因子與水準值) 選用適當的直交表 (Orthogonal Array, OA)，將直交表所設計的實驗完成後，再利用其反應值與平均值去分析每個因子所造成的影響，此方法的優點在於能以較少的實驗次數來對於系統參數做最佳化設計。本研究主要應用田口方法尋找影像辨識之最佳化參數組合。

2.1.2 直交表

利用直交表進行實驗是穩健設計重要的一環，所謂直交 (Orthogonal) 是指每兩行之間，所有組合出現的頻率都是一樣。田口博士一共列出 18 種標準直交表，而本實驗以 L_{12} 直交表為例說明 (如表 2-1 所示)

表 2-1 L₁₂ (2¹¹) 直交表

實驗 編號	控制因子										
	A	B	C	D	E	F	G	H	I	J	K
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	2	2	2	2	2	2
3	1	1	2	2	2	1	1	1	2	2	2
4	1	2	1	2	2	1	2	2	1	1	2
5	1	2	2	1	2	2	1	2	1	2	1
6	1	2	2	2	1	2	2	1	2	1	1
7	2	1	2	2	1	1	2	2	1	2	1
8	2	1	2	1	2	2	2	1	1	1	2
9	2	1	1	2	2	2	1	2	2	1	1
10	2	2	2	1	1	1	1	2	2	1	2
11	2	2	1	2	1	2	1	1	1	2	2
12	2	2	1	1	2	1	2	1	2	2	1

2.1.3 品質損失函數與訊號雜訊比

田口博士為了將品質損失量化而提出品質損失函數，依品質特性可區分為望大、望目與望小。

(1) 望大特性 (larger-the-better, LTB) :

品質特性越大越好，其理想值為無窮大 (假設品質特性是正數)。

$$\text{品質損失函數：} L(y) = ky^2 \quad (2-1)$$

$$\text{S/N 比 } \eta = -10 \log \left(\frac{1}{n} \sum_{i=1}^n \frac{1}{y_i^2} \right) \quad (2-2)$$

其中 n 為各實驗組數， y_i 為各組實驗樣本特徵值數據。

(2) 望目特性 (nominal-the-better, NTB) :

品質特性越接近目標值越好。

$$\text{品質損失函數：} L(y) = (y - m)^2 \quad (2-3)$$

當 $y = m$ 時，則 $L(y) = 0$ ，亦即品質損失為零。

$$\text{S/N 比 } \eta = 10 \log \left(\frac{\bar{y}^2}{s^2} \right) \quad (2-4)$$

其中 \bar{y} 為所觀測的平均值、 s 為標準差、 m 為目標值。

(3) 望小特性 (small-the-better, STB) :

品質特性越小越好，變異數越小越好，其理想值可視為零。

$$\text{品質損失函數：} L(y) = k \frac{1}{y_i^2} \quad (2-5)$$

$$\text{S/N 比 } \eta = -10 \log \left[\frac{1}{n} \sum_{i=1}^n (y^2) \right] \quad (2-6)$$

其中 n 為各實驗組數， y_i 為各組實驗樣本特徵值數據 (林春妙, 2012)。

2.1.4 田口品質工程應用

目前為止，田口方法已被廣泛地運用到各行各業，對於已知的缺點幾乎都能得到明顯的改善。在科學方面，張英彬、駱景堯 (2006) 利用田口方法於被動式諧波濾波器，使其能有效降低系統之諧波失真，以提升供電系統的電力品質；陳宇祥 (2015) 將田口方法應用於最大功率追蹤控制策略之最佳

參數設計，使發電機輸出功率追蹤風能輸入功率變化，減少馬達輸出功率，達成風能轉換電能的目標；周東毅（2015）使用田口方法找出驅動器變壓器及輸出電感器之最佳化參數，來改善轉換器效率，達到高功率 LED 路燈驅動源效率改善。

在醫學方面，江福財（2017）運用田口方法評量 ^{99m}Tc 核醫攝影術中伽瑪影像造影之最佳化調整參數組合，以改善臨床影像品質；葉大銘（2017）利用田口方法分別找出 ACR 假體內腫瘤、纖維化與鈣化的最佳化，發現不同的組織特性，可以藉由改變不同的操作因子，而獲得解析度的改善。

在傳統產業方面，金香君（2019）運用田口方法找出可控制因子來改善加工精度製程，提高零件加工精度，以達到降低生產成本、符合客戶交期、品質之目的；賴銘俊（2019）針對汽車雷達天線效能以田口方法做最佳化設計；施文淵（2018）透過田口實驗，針對 EVA 發泡射出成型製程之工序，取得最佳參數進行製程優化。由此可見，田口品質工程在各個領域皆有巨大的貢獻。

第二節 機器學習與深度學習

機器學習 (Machine learning) 即是讓電腦像人類一樣具有學習的能力，其學習方式與人類極為相近，通常會先進行分類，才能分析、判斷，最後採取行動。主要分成監督式學習、非監督式學習、半監督式學習及強化式學習四種。

監督式學習：

所有資料都被標註，並告訴機器相對應的值，以提供機器判斷輸出時使用，此方法以人工分類為主。

非監督式學習：

所有資料都未進行標註，機器自行尋找資料的特徵並分類，故誤差較大。

半監督式學習：

標註少部分資料，電腦透過有標註的資料尋找特徵，並對其他資料進行分類，此方法可以讓預測時比較精準。

強化式學習：

對資料不進行標註，但讓機器透過一次次與環境互動來學習，根據反饋

的好壞，自行逐步修正，最後取得正確結果。有了強化式學習，能使非監督式學習達到一定程度的準確性（伊雲谷，2019）。

而深度學習是機器學習的一個子集合（如圖2-1），能把資料透過多個處理層中的線性或非線性轉換，自動抽取代表資料特性的特徵，以層的概念建構演算法，打造出能夠自行學習並做出正確選擇的人工神經網路。目前深度學習較常使用的有卷積神經網路（CNN）、循環神經網路（RNN）、深度神經網路（DNN）、深度置信網路（DBN）等。

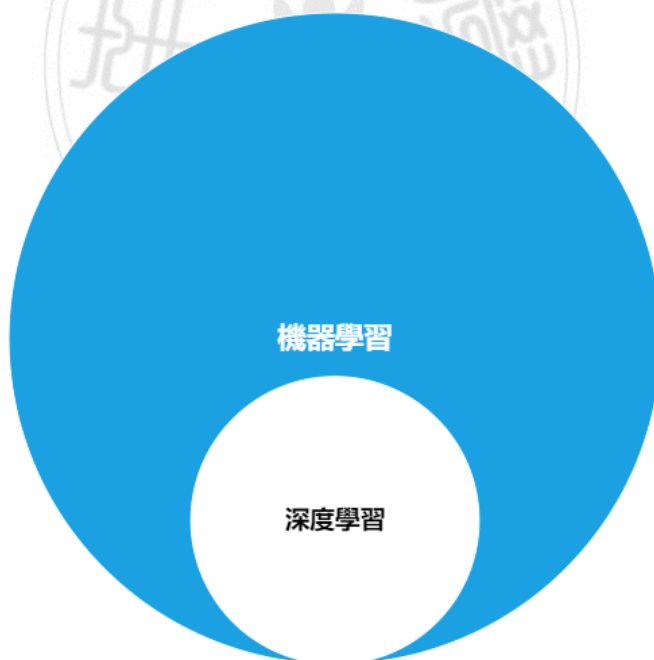


圖2-1 深度學習為機器學習中的子集合

在學術上，林逸青（2019）以LSTM模型對股票市場的漲跌趨勢與股價指數能有效地做預測；張乃恩（2017）以Faster R-CNN+ASM模型應用於心臟左心室超音波影像追蹤達到一定的準確率；許哲昇（2017）利用LSTM模型的特性，準確預估機器剩餘可用壽命；劉冠宏（2018）成功運用深度學習從平坦混和式訊號電路分離出子電路；楊雅筑（2018）以CAE神經網路開發嵌入式車道偵測系統。



第三章 研究方法

本研究將 CNN 作為特徵提取並結合 LSTM 來預測，這樣的模型為長期卷積神經網路（LRCN 模型）。實驗中我們先定義 CNN 模型，並將其包含在 TimeDistributed 層中，再新增到 LSTM 模型，在 Google 雲端硬碟裡的 Colaboratory 環境下執行，搭配 GPU 加速器縮短實驗時間。最後運用田口品質工程找出其最佳化配置。實驗步驟如下：

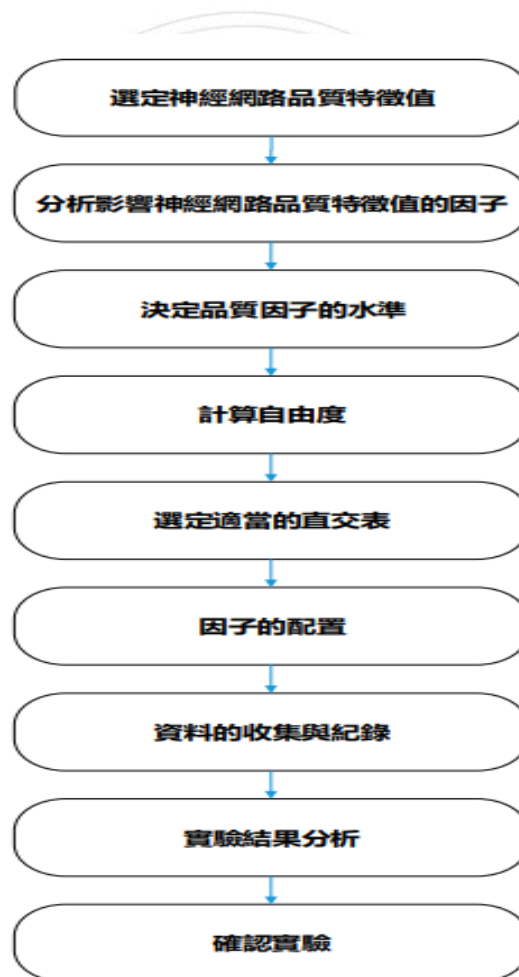


圖 3-1 循環神經網路最佳化實驗步驟

選定神經網路品質特徵值

本實驗以深度學習的準確度為神經網路實驗的品質特徵值，準確度越高表示深度學習的成效越好，故以望大特性作為分析依據。

分析影響神經網路品質特徵值的因子

影響神經網路品質特徵值的變數共有 6 個因子，包含卷積層激勵函數、池化層大小、輸出層激勵函數、Loss 損失函式、Optimizer 最佳化參數以及 Metrics 評估準確率方法。

決定神經網路品質因子的水準

本實驗共分六個因子，包含卷積層激勵函數（代號 A）、池化層大小（代號 B）、輸出層激勵函數（代號 C）、Loss 損失函式（代號 D）、Optimizer 最佳化參數（代號 E）、Metrics 評估準確率方法（代號 F）。其中，每個因子的水準如下：

表 3-1 控制因子與水準

控制因子	編碼	水準 1	水準 2	水準 3	水準 4
卷積層激勵函數	A	relu	tanh	elu	selu
池化層大小	B	2X2	3X3		
輸出層激勵函數	C	sigmoid	tanh		
Loss 損失函數	D	categorical_crossentropy	binary_crossentropy		
Optimizer 優化函數	E	adam	adagrad		
Metrics 評估準確率方法	F	accuracy	binary_accuracy		

以下針對各因子與水準詳細說明：

A. 卷積層激勵函數：

由於現實中的問題皆屬於非線性問題，故激勵函數需具有非線性、可微性、單調性等特點。因此，我們選擇 relu、tanh、elu、selu 作為 4 個水準。

Relu 函數：線性整流函數 (Rectified linear unit, Relu)，又稱為修正線性單元，是類神經網路中最常使用的激勵函數。如圖 3-2 所示，若值為正數，則輸出該值大小，若為負數，則輸出為零，其具有計算速度快、收斂快及解決梯度爆炸等特點。

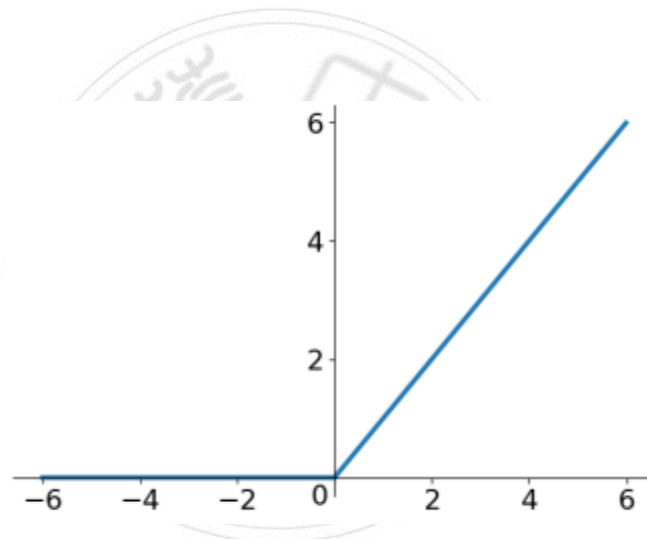


圖 3-2 Relu 函數

Tanh 函數：雙曲正切函數 (Hyperbolic tangent function)，是雙曲函數的一種。如圖 3-3 所示，其優點是完全可微分，且對稱中心在原點的反對稱。

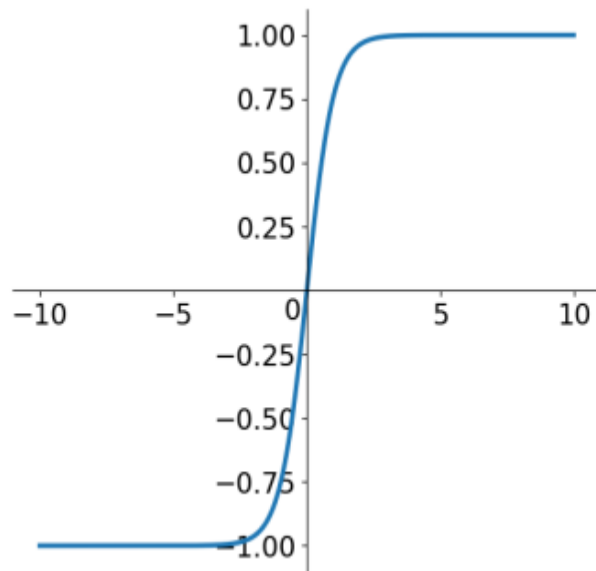


圖 3-3 Tanh 函數

Elu 函數：指數線性單元 (Exponential linear unit, Elu)，屬於 Relu 修正類激勵函數的一種。如圖 3-4 所示，其正值區間與 Relu 相同，而負值區間為負值輸入添加了一個負值輸出。

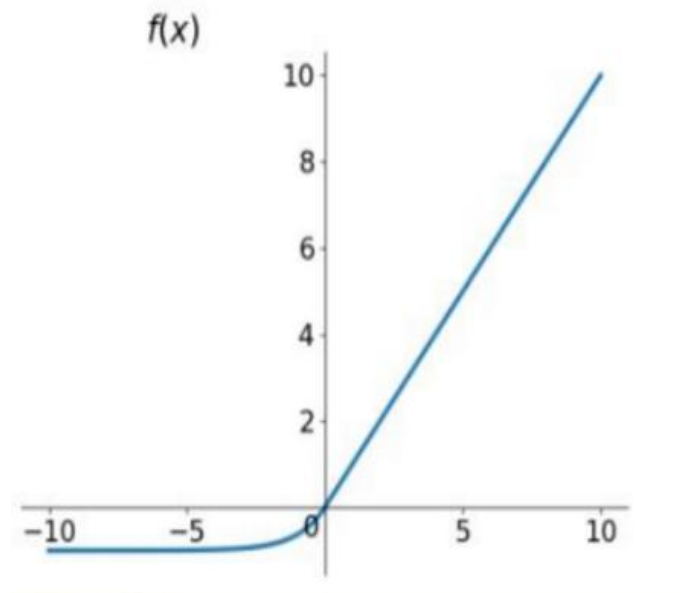


圖 3-4 Elu 函數

Selu 函數：擴展指數線性單元 (Scaled exponential linear unit, Selu) ，為 Elu 乘上一個 λ 值，如圖 3-5 所示，其優點能使樣本分布自動歸一化到零均值和單位方差，在訓練過程中不會產生梯度爆炸或消失。

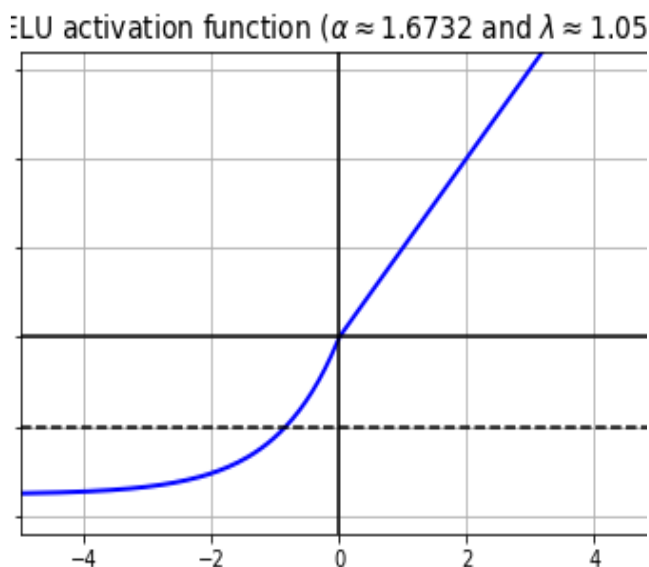


圖 3-5 Selu 函數

B. 池化層大小：

池化層的作用在於壓縮圖片大小來做特徵萃取，本研究以 2x2 與 3x3 兩種濾鏡大小作為水準進行實驗。

C. 輸出層激勵函數：

本研究選擇 Sigmoid 與 Tanh 兩種水準進行實驗，其中 Sigmoid 的輸出介於 0 和 1 之間，因此不會出現梯度爆炸的問題。

D. Loss 損失函數：

當預測值與實際值越相近，損失函數就越小。本研究選擇水準 1 為多元分類的 categorical_crossentropy，水準 2 為二元分類的 binary_crossentropy。

E. Optimizer 優化函數：

優化函數的作用在於幫助模型在訓練過程中，如何更快更好地將參數調整到位，使 Loss 越小越好。本研究選擇水準 1 為 adam，水準 2 為適合處理稀鬆數據的 adagrad。

F. Metrics 評估準確率方法：

本研究選擇水準 1 為 accuracy，水準 2 為 binary_accuracy。

計算自由度

選擇直交表之前需先計算總自由度，而因子自由度=水準-1，故本實驗之各因子的自由度如下：

A 因子的自由度為 $4-1=3$

B 因子的自由度為 $2-1=1$

C 因子的自由度為 $2-1=1$

D 因子的自由度為 $2-1=1$

E 因子的自由度為 $2-1=1$

F 因子的自由度為 $2-1=1$

合計總自由度為 8

自由度是為了要確定所需實驗次數的最小值，所選用的直交表行數須大於總自由度。

選定適當直交表

直交表提供有效率、有系統控制因子的方法，除了可以大量降低實驗成本外，還有簡化資料分析工作極具可靠性等優點。

本實驗選擇 $L_{12}(2^{11})$ 直交表來做分析，因其自由度為 11 大於本實驗自由度 8。 $L_{12}(2^{11})$ 直交表主要是評估因子效應，而非用來評估因子間的交互作用。

$L_{12}(2^{11})$ 直交表有 11 個直行，可以經由建構新的直行來表示其中某兩個直行間的交互作用，而因子效應的評估主要則是透過混和水準直交表來進行，而假設交互作用並不存在；假設有交互作用存在，這些直交表讓交互作用分散到各行中，讓因子保留了相對大小。

本實驗各因子間無交互作用，所以本實驗使用 $L_{12}(2^{11})$ 直交表，而各因子配置情形如表 3-2。

表 3-2 循環神經網路最佳化實驗 $L_{12} (2^{11})$ 直交表配置

實驗 編號	1	2	3	4	5	6	7	8	9	10	11
	B		C		A		D	E		F	
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	2	2	2	2	2	2
3	1	1	2	2	2	1	1	1	2	2	2
4	1	2	1	2	2	1	2	2	1	1	2
5	1	2	2	1	2	2	1	2	1	2	1
6	1	2	2	2	1	2	2	1	2	1	1
7	2	1	2	2	1	1	2	2	1	2	1
8	2	1	2	1	2	2	2	1	1	1	2
9	2	1	1	2	2	2	1	2	2	1	1
10	2	2	2	1	1	1	1	2	2	1	2
11	2	2	1	2	1	2	1	1	1	2	2
12	2	2	1	1	2	1	2	1	2	2	1

因子的配置

本實驗控制因子除卷積層激勵函數 relu 、 tanh 、 elu 、 selu 共 4 個水準外，其餘因子皆為 2 個水準。而 $L_{12} (2^{11})$ 為 2 水準的直交表，卷積層激勵函數卻有 4 水準，故我們可以經由合併直交表中的兩行如表 3-3 所示，來創造出新的直交表詳見表 3-4。第 5 行與第 6 行合併後，成為新的第 5 行。

表 3-3 行的合併

行	5	6	5''
水準	1	1	1
水準	1	2	2
水準	2	1	3
水準	2	1	3
水準	2	2	4
水準	1	2	2
水準	1	1	1
水準	2	2	4
水準	2	2	4
水準	1	1	1
水準	1	2	2
水準	2	1	3

合併規則：

第 5 行水準 1 與第 6 行水準 1 合併為新水準：1

第 5 行水準 1 與第 6 行水準 2 合併為新水準：2

第 5 行水準 2 與第 6 行水準 1 合併為新水準：3

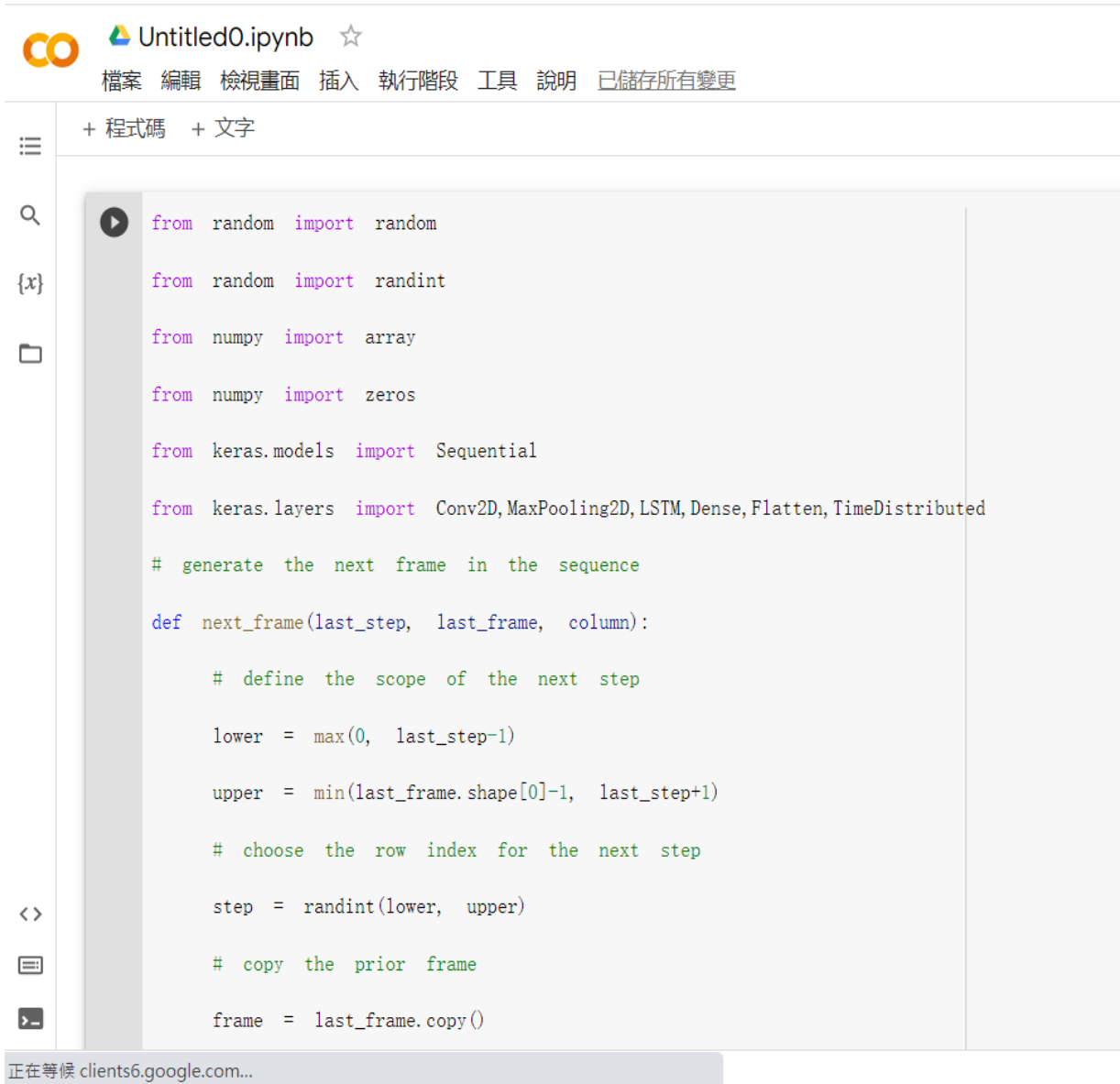
第 5 行水準 2 與第 6 行水準 2 合併為新水準：4

表 3-4 $L_{12} (2^{11})$ 新直交表

實驗 編號	1	2	3	4	5 ^a	7	8	9	10	11
	B		C		A	D	E		F	
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2
3	1	1	2	2	3	1	1	2	2	2
4	1	2	1	2	3	2	2	1	1	2
5	1	2	2	1	4	1	2	1	2	1
6	1	2	2	2	2	2	1	2	1	1
7	2	1	2	2	1	2	2	1	2	1
8	2	1	2	1	4	2	1	1	1	2
9	2	1	1	2	4	1	2	2	1	1
10	2	2	2	1	1	1	2	2	1	2
11	2	2	1	2	2	1	1	1	2	2
12	2	2	1	1	3	2	1	2	2	1

資料的收集與紀錄

本研究以 <https://www.gushiciku.cn/pl/2dUq/zh-tw> 網站的程式碼為例，藉由程式產生影像序列，並以 CNN+LSTM 模型進行預測，在每次實驗完成後均將系統重啟，以維持資料之再現性。程式碼如下：



The image shows a Jupyter Notebook interface with a sidebar on the left containing icons for home, search, {x}, folder, expand, and terminal. The main area displays Python code for generating a frame sequence. The code includes imports for random, numpy, and keras, and a function named next_frame that takes last_step, last_frame, and column as arguments. The function defines a scope for the next step, chooses a row index, and copies the prior frame.

```
from random import random
from random import randint
from numpy import array
from numpy import zeros
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, LSTM, Dense, Flatten, TimeDistributed

# generate the next frame in the sequence

def next_frame(last_step, last_frame, column):
    # define the scope of the next step
    lower = max(0, last_step-1)
    upper = min(last_frame.shape[0]-1, last_step+1)
    # choose the row index for the next step
    step = randint(lower, upper)
    # copy the prior frame
    frame = last_frame.copy()
```

正在等候 clients6.google.com...

圖 3-6 生成幀序列的示例



Untitled0.ipynb ☆

檔案 編輯 檢視畫面 插入 執行階段 工具 說明 已儲存所有變更

```
+ 程式碼 + 文字

▶ # add the new step

    frame[step, column] = 1

    return frame, step

# generate a sequence of frames of a dot moving across an image

def build_frames(size):

    frames = list()

    # create the first frame

    frame = zeros((size, size))

    step = randint(0, size-1)

    # decide if we are heading left or right

    right = 1 if random() < 0.5 else 0

    col = 0 if right else size-1

    frame[step, col] = 1

    frames.append(frame)

    # create all remaining frames

    for i in range(1, size):
```

正在等候 clients6.google.com...

圖 3-7 生成幀序列的示例

```
col = i if right else size-1-i

frame, step = next_frame(step, frame, col)

frames.append(frame)

return frames, right

# generate multiple sequences of frames and reshape for network input
def generate_examples(size, n_patterns):

    X, y = list(), list()

    for _ in range(n_patterns):

        frames, right = build_frames(size)

        X.append(frames)

        y.append(right)

        # resize as [samples, timesteps, width, height, channels]

    X = array(X).reshape(n_patterns, size, size, size, 1)

    y = array(y).reshape(n_patterns, 1)

    return X, y

# configure problem
```

正在等候 clients6.google.com...

圖 3-8 生成隨機影像的新函式



```
+ 程式碼 + 文字

size = 50

# define the model
model = Sequential()

model.add(TimeDistributed(Conv2D(2, (2,2), activation='relu', input_shape=(None, size, size, 1))))

model.add(TimeDistributed(MaxPooling2D(pool_size=(2, 2))))

model.add(TimeDistributed(Flatten()))

model.add(LSTM(50))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

print(model.summary())

# fit model

X, y = generate_examples(size, 5000)

model.fit(X, y, batch_size=32, epochs=1)

# evaluate model

X, y = generate_examples(size, 100)
```

A 因子

C 因子

B 因子

D 因子

E 因子

F 因子

圖 3-9 定義 CNN 及 LSTM 模型

```

+ 程式碼 + 文字

X, y = generate_examples(size, 100)

loss, acc = model.evaluate(X, y, verbose=0)

print('loss: %f, acc: %f' % (loss, acc*100))

# prediction on new data

X, y = generate_examples(size, 1)

yhat = model.predict_classes(X, verbose=0)

expected = "Right" if y[0]==1 else "Left"

predicted = "Right" if yhat[0]==1 else "Left"

print('Expected: %s, Predicted: %s' % (expected, predicted))

```

圖 3-10 生成 100 個新的隨機序列並估計準確度

執行結果如下：

```

+ 程式碼 + 文字

Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
time_distributed (TimeDistr  (None, None, 49, 49, 2)  10
ibuted)

time_distributed_1 (TimeDis  (None, None, 24, 24, 2)  0
tributed)

time_distributed_2 (TimeDis  (None, None, 1152)       0
tributed)

lstm (LSTM)                  (None, 50)                240600

dense (Dense)                (None, 1)                 51
-----
Total params: 240,661
Trainable params: 240,661
Non-trainable params: 0
-----
None
157/157 [=====] - 15s 19ms/step - loss: 0.0000e+00 - accuracy: 0.5042
loss: 0.000000, acc: 47.000000

```

圖 3-11 程式碼執行結果

根據直交表因子配置(如表 3-4)，本實驗執行 12 次實驗步驟如表 3-5 所示：

表 3-5 循環神經網路最佳化實驗步驟

實驗順序	池化層 大小	輸出層 激勵函數	卷積層 激勵函數	Loss 損失函數	Optimizer 優化函數	Metrics 評估準確率方法
1	2X2	sigmoid	relu	categorical_crossentropy	adam	accuracy
2	2X2	sigmoid	tanh	binary_crossentropy	adagrad	binary_accuracy
3	2X2	tanh	elu	categorical_crossentropy	adam	binary_accuracy
4	2X2	sigmoid	elu	binary_crossentropy	adagrad	binary_accuracy
5	2X2	tanh	selu	categorical_crossentropy	adagrad	accuracy
6	2X2	tanh	tanh	binary_crossentropy	adam	accuracy
7	3X3	tanh	relu	binary_crossentropy	adagrad	accuracy
8	3X3	tanh	selu	binary_crossentropy	adam	binary_accuracy
9	3X3	sigmoid	selu	categorical_crossentropy	adagrad	accuracy
10	3X3	tanh	relu	categorical_crossentropy	adagrad	binary_accuracy
11	3X3	sigmoid	tanh	categorical_crossentropy	adam	binary_accuracy
12	3X3	sigmoid	elu	binary_crossentropy	adam	accuracy

依據表 3-5 實驗結果詳見表 3-6 所示：

表 3-6 循環神經網路實驗結果紀錄表

實驗 編號	1	2	3	4	5	7	8	9	10	11	循環神經網路實驗結果									
	B	C	A			D	E	F												
1	1	1	1	1	1	1	1	1	1	1	0.5042	0.5000	0.4994	0.5032	0.4978	0.4956	0.4880	0.5040	0.5064	0.4940
2	1	1	1	1	2	2	2	2	2	2	0.5104	0.5182	0.5170	0.5184	0.5108	0.4968	0.5076	0.4908	0.4982	0.5150
3	1	1	2	2	3	1	1	2	2	2	0.4986	0.4890	0.4876	0.5028	0.5054	0.5158	0.4956	0.4972	0.5038	0.5026
4	1	2	1	2	3	2	2	1	1	2	0.5222	0.5092	0.5230	0.4962	0.5220	0.5154	0.5016	0.4956	0.5042	0.4938
5	1	2	2	1	4	1	2	1	2	1	0.5038	0.5002	0.4996	0.4918	0.5036	0.4986	0.4970	0.4882	0.4996	0.5116
6	1	2	2	2	2	2	1	2	1	1	0.8890	0.8874	0.8520	0.8688	0.8622	0.8558	0.9192	0.8656	0.7750	0.8604
7	2	1	2	2	1	2	2	1	2	1	0.5234	0.4982	0.5070	0.4968	0.5368	0.5116	0.5234	0.5200	0.5252	0.5340
8	2	1	2	1	4	2	1	1	1	2	0.8372	0.8812	0.8712	0.8318	0.8616	0.8860	0.8574	0.8884	0.8250	0.8050
9	2	1	1	2	4	1	2	2	1	1	0.5082	0.5042	0.5068	0.5180	0.4954	0.5004	0.5114	0.4990	0.5062	0.5080
10	2	2	2	1	1	1	2	2	1	2	0.5030	0.4918	0.5000	0.5096	0.4935	0.4948	0.4994	0.5024	0.4958	0.5000
11	2	2	1	2	2	1	1	1	2	2	0.4852	0.4982	0.4966	0.5018	0.4954	0.4860	0.5008	0.4976	0.5064	0.5032
12	2	2	1	1	3	2	1	2	2	1	0.9460	0.9230	0.9290	0.9342	0.9184	0.9360	0.8566	0.8826	0.9366	0.9322

第四章 實驗結果分析

將實驗結果所得到的120筆數據收集且登錄後，運用EXCEL計算每次的平均數（如表4-1）。

表4-1 循環神經網路實驗結果之平均數

實驗 編號	1	2	3	4	5'	7	8	9	10	11	循環神經網路實驗結果										平均數
	B	C	A	D	E	F															
1	1	1	1	1	1	1	1	1	1	1	0.5042	0.5000	0.4994	0.5032	0.4978	0.4956	0.4880	0.5040	0.5064	0.4940	0.4993
2	1	1	1	1	2	2	2	2	2	2	0.5104	0.5182	0.5170	0.5184	0.5108	0.4968	0.5076	0.4908	0.4982	0.5150	0.5083
3	1	1	2	2	3	1	1	2	2	2	0.4986	0.4890	0.4876	0.5028	0.5054	0.5158	0.4956	0.4972	0.5038	0.5026	0.4998
4	1	2	1	2	3	2	2	1	1	2	0.5222	0.5092	0.5230	0.4962	0.5220	0.5154	0.5016	0.4956	0.5042	0.4938	0.5083
5	1	2	2	1	4	1	2	1	2	1	0.5038	0.5002	0.4996	0.4918	0.5036	0.4986	0.4970	0.4882	0.4996	0.5116	0.4994
6	1	2	2	2	2	2	1	2	1	1	0.8890	0.8874	0.8520	0.8688	0.8622	0.8558	0.9192	0.8656	0.7750	0.8604	0.8635
7	2	1	2	2	1	2	2	1	2	1	0.5234	0.4982	0.5070	0.4968	0.5368	0.5116	0.5234	0.5200	0.5252	0.5340	0.5176
8	2	1	2	1	4	2	1	1	1	2	0.8372	0.8812	0.8712	0.8318	0.8616	0.8860	0.8574	0.8884	0.8250	0.8050	0.8545
9	2	1	1	2	4	1	2	2	1	1	0.5082	0.5042	0.5068	0.5180	0.4954	0.5004	0.5114	0.4990	0.5062	0.5080	0.5058
10	2	2	2	1	1	1	2	2	1	2	0.5030	0.4918	0.5000	0.5096	0.4935	0.4948	0.4994	0.5024	0.4958	0.5000	0.4990
11	2	2	1	2	2	1	1	1	2	2	0.4852	0.4982	0.4966	0.5018	0.4954	0.4860	0.5008	0.4976	0.5064	0.5032	0.4971
12	2	2	1	1	3	2	1	2	2	1	0.9460	0.9230	0.9290	0.9342	0.9184	0.9360	0.8566	0.8826	0.9366	0.9322	0.9195

本次實驗目的為循環神經網路最佳化配置實驗，因此準確率要越高越好，符合田口理論中的望大特性，故將實驗結果之數據帶入望大特性的公式（2-2）並計算其S/N比，結果如表4-2所示。

表4-2 循環神經網路最佳化實驗S/N比

實驗 編號	1	2	3	4	5"	7	8	9	10	11	循環神經網路實驗結果										平均數	S/N 比
	B	C	A	D	E	F																
1	1	1	1	1	1	1	1	1	1	1	0.5042	0.5000	0.4994	0.5032	0.4978	0.4956	0.4880	0.5040	0.5064	0.4940	0.4993	-6.0350
2	1	1	1	1	2	2	2	2	2	2	0.5104	0.5182	0.5170	0.5184	0.5108	0.4968	0.5076	0.4908	0.4982	0.5150	0.5083	-5.8817
3	1	1	2	2	3	1	1	2	2	2	0.4986	0.4890	0.4876	0.5028	0.5054	0.5158	0.4956	0.4972	0.5038	0.5026	0.4998	-6.0266
4	1	2	1	2	3	2	2	1	1	2	0.5222	0.5092	0.5230	0.4962	0.5220	0.5154	0.5016	0.4956	0.5042	0.4938	0.5083	-5.8834
5	1	2	2	1	4	1	2	1	2	1	0.5038	0.5002	0.4996	0.4918	0.5036	0.4986	0.4970	0.4882	0.4996	0.5116	0.4994	-6.0330
6	1	2	2	2	2	2	1	2	1	1	0.8890	0.8874	0.8520	0.8688	0.8622	0.8558	0.9192	0.8656	0.7750	0.8604	0.8635	-1.2976
7	2	1	2	2	1	2	2	1	2	1	0.5234	0.4982	0.5070	0.4968	0.5368	0.5116	0.5234	0.5200	0.5252	0.5340	0.5176	-5.7279
8	2	1	2	1	4	2	1	1	1	2	0.8372	0.8812	0.8712	0.8318	0.8616	0.8860	0.8574	0.8884	0.8250	0.8050	0.8545	-1.3793
9	2	1	1	2	4	1	2	2	1	1	0.5082	0.5042	0.5068	0.5180	0.4954	0.5004	0.5114	0.4990	0.5062	0.5080	0.5058	-5.9230
10	2	2	2	1	1	1	2	2	1	2	0.5030	0.4918	0.5000	0.5096	0.4935	0.4948	0.4994	0.5024	0.4958	0.5000	0.4990	-6.0387
11	2	2	1	2	2	1	1	1	2	2	0.4852	0.4982	0.4966	0.5018	0.4954	0.4860	0.5008	0.4976	0.5064	0.5032	0.4971	-6.0731
12	2	2	1	1	3	2	1	2	2	1	0.9460	0.9230	0.9290	0.9342	0.9184	0.9360	0.8566	0.8826	0.9366	0.9322	0.9195	-0.7409

再將S/N比依照各因子與水準計算出反應值，並整理成循環神經網路最佳化因子平均值反應表以及循環神經網路最佳化因子反應表，詳見表4-3及4-4，再以反應表繪製反應圖（如圖4-1所示）。

表4-3 循環神經網路最佳化因子平均值反應表

水準	A	B	C	D	E	F
1	0.5053	0.5631	0.5730	0.5001	0.6890	0.6217
2	0.6230	0.6322	0.6223	0.6953	0.5064	0.5736
3	0.6425					
4	0.6199					

表4-4 循環神經網路最佳化因子S/N比反應表

水準	A	B	C	D	E	F
1	-5.9339	-5.1929	-5.0895	-6.0216	-3.5921	-4.4262
2	-4.4175	-4.3138	-4.4172	-3.4851	-5.9146	-5.0805
3	-4.2170					
4	-4.4451					

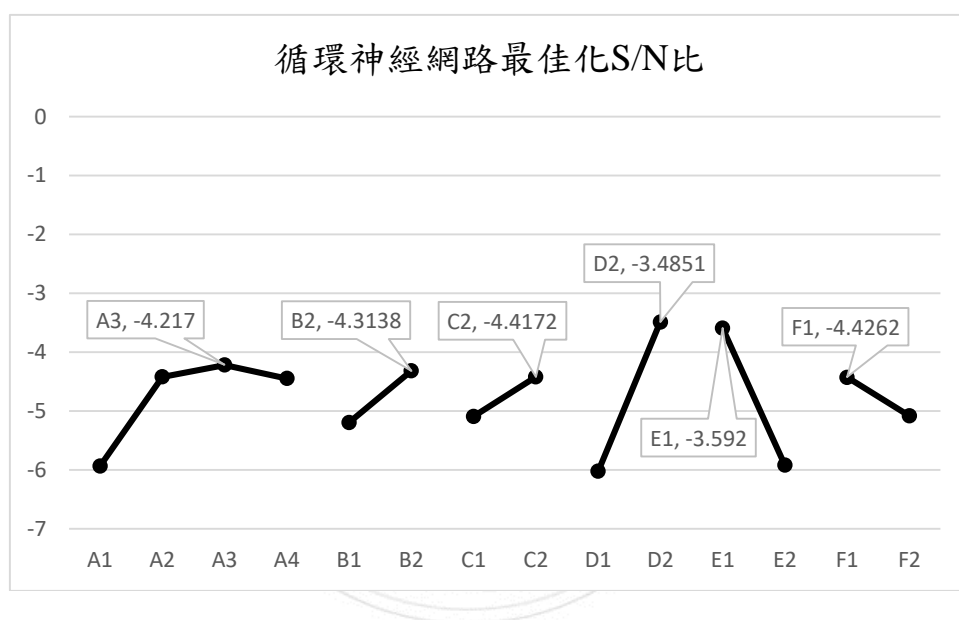


圖4-1 循環神經網路最佳化因子S/N比反應圖

根據表4-3因子平均反應表及圖4-1因子S/N比反應圖得知，A、B、C、D、E、F影響因子有顯著性差異，循環神經網路最佳化因素配置：卷積層激勵函數為elu、池化層大小為3x3、輸出層激勵函數為tanh、Loss損失函數為binary_crossentropy、Optimizer優化函數為adam、Metrics評估準確率方法為

accuracy。

現行的循環網路神經實驗的條件為： $A_1B_1C_1D_1E_1F_1$ ，卷積層激勵函數relu (A_1)、池化層大小為 2×2 (B_1)、輸出層激勵函數為sigmoid (C_1)、Loss損失函數為categorical_crossentropy (D_1)、Optimizer優化函數為adam (E_1)、Metrics評估準確率方法為accuracy (F_1)。

$$\begin{aligned} S/N_{initial} &= \bar{\eta} + (\bar{\eta}_{A_1} - \bar{\eta}) + (\bar{\eta}_{B_1} - \bar{\eta}) + (\bar{\eta}_{C_1} - \bar{\eta}) + (\bar{\eta}_{D_1} - \bar{\eta}) + (\bar{\eta}_{E_1} - \bar{\eta}) + (\bar{\eta}_{F_1} - \bar{\eta}) \\ &= \bar{\eta}_{A_1} + \bar{\eta}_{B_1} + \bar{\eta}_{C_1} + \bar{\eta}_{D_1} + \bar{\eta}_{E_1} + \bar{\eta}_{F_1} - 5\bar{\eta} \\ &= -5.9339 - 5.1929 - 5.0895 - 6.0216 - 3.5920 - 4.4262 + 5 \times 4.7533 \\ &= -6.4896 \text{ (db)} \end{aligned}$$

$$\begin{aligned} \mu_{initial} &= \bar{\mu} + (\bar{\mu}_{A_1} - \bar{\mu}) + (\bar{\mu}_{B_1} - \bar{\mu}) + (\bar{\mu}_{C_1} - \bar{\mu}) + (\bar{\mu}_{D_1} - \bar{\mu}) + (\bar{\mu}_{E_1} - \bar{\mu}) + (\bar{\mu}_{F_1} - \bar{\mu}) \\ &= \bar{\mu}_{A_1} + \bar{\mu}_{B_1} + \bar{\mu}_{C_1} + \bar{\mu}_{D_1} + \bar{\mu}_{E_1} + \bar{\mu}_{F_1} - 5\bar{\mu} \\ &= 0.5053 + 0.5631 + 0.5730 + 0.5001 + 0.6890 + 0.6217 - 5 \times 0.5977 \\ &= 0.4637 \end{aligned}$$

而最佳化的配置為： $A_3B_2C_2D_2E_1F_1$ ，卷積層激勵函數為elu (A_3)、池化層大小為 3×3 (B_2)、輸出層激勵函數為tanh (C_2)、Loss損失函數為binary_crossentropy (D_2)、Optimizer優化函數為adam (E_1)、Metrics評估準確率方法為accuracy (F_1)。

$$\begin{aligned} S/N_{optimum} &= \bar{\eta} + (\bar{\eta}_{A_3} - \bar{\eta}) + (\bar{\eta}_{B_2} - \bar{\eta}) + (\bar{\eta}_{C_2} - \bar{\eta}) + (\bar{\eta}_{D_2} - \bar{\eta}) + (\bar{\eta}_{E_1} - \bar{\eta}) \\ &\quad + (\bar{\eta}_{F_1} - \bar{\eta}) \end{aligned}$$

$$\begin{aligned}
&= \bar{\eta}_{A_3} + \bar{\eta}_{B_2} + \bar{\eta}_{D_2} + \bar{\eta}_{E_1} + \bar{\eta}_{F_1} - 5\bar{\eta} \\
&= -4.2170 - 4.3138 - 4.4172 - 3.4851 - 3.5920 - 4.4262 + 5 \times 4.7533 \\
&= -0.6848 \text{ (db)}
\end{aligned}$$

$$\begin{aligned}
\mu_{optimum} &= \bar{\mu} + (\bar{\mu}_{A_3} - \bar{\mu}) + (\bar{\mu}_{B_2} - \bar{\mu}) + (\bar{\mu}_{C_2} - \bar{\mu}) + (\bar{\mu}_{D_2} - \bar{\mu}) + (\bar{\mu}_{E_1} - \bar{\mu}) + (\bar{\mu}_{F_1} - \bar{\mu}) \\
&= \bar{\mu}_{A_3} + \bar{\mu}_{B_2} + \bar{\mu}_{C_2} + \bar{\mu}_{D_2} + \bar{\mu}_{E_1} + \bar{\mu}_{F_1} - 5\bar{\mu} \\
&= 0.6425 + 0.6322 + 0.6223 + 0.6953 + 0.6890 + 0.6217 - 5 \times 0.5977 \\
&= 0.9145
\end{aligned}$$

將最佳化因子配置與現行因子配置進行比較，其增益為 $-0.6848 - (-6.4896) = 5.8048$ (db)，故我們推定 $A_3B_2C_2D_2E_1F_1$ 為最佳化配置，為了確認此條件是否為最佳組合，需再作確認實驗加以驗證。

進行10次確認實驗後並收集數據結果如下：

表4-5 確認實驗結果

確認 實驗	1	2	3	4	5*	7	8	9	10	11	確認實驗結果										平均數	S/N 比
	B	C	A	D	E	F	1	2	3	4	5	6	7	8	9	10						
	2	2	3	2	1	1	0.8970	0.8778	0.8688	0.8780	0.8536	0.8548	0.9026	0.8766	0.8618	0.8766	0.8748	-1.1662				

根據確認實驗結果可知，平均數為0.8748，S/N比為-1.1662，僅次於第12組實驗，平均數為0.9195，S/N比為-0.7409，並非最佳組合，是排第二，為前92%好的結果，具有其參考價值。

第五章 結論

人工智慧最早可追溯至 1950 年代，到目前為止已有很長一段時間，經由不斷地突破而發展出深度學習技術。藉由機器學習與深度學習的搭配，漸漸地使人工智慧擁有預測能力，並能運用在更廣泛的領域，生活中許多辨識系統都仰賴深度學習而得以實現。

本研究目的是運用田口品質工程法，搭配適當直交表，並透過最佳化實驗，找出 CNN+LSTM 模型的最佳因素配置，實驗結果顯示最佳化組合為卷積層激勵函數為 elu、池化層大小為 3x3、輸出層激勵函數為 tanh、Loss 損失函數為 binary_crossentropy、Optimizer 優化函數為 adam、Metrics 評估準確率方法為 accuracy。然而在確認實驗中，卻發現此組合並非是準確率最高的組合，因此，我們可以思考是否有影響結果的隱藏控制因子還未被尋找出來，導致實驗結果不如預期，不過，以準確率來說，利用田口實驗找出來的最佳化配置還是有第二高的參考價值。

參 考 文 獻

一、中文文獻

1. David Sheehan (2021)。一文看盡 26 種神經網路激活函數 (從 Relu 到 Sinc)。2022 年 5 月 1 日，取自 <https://ppfocus.com/0/ed41c67ef.html>
2. 古詩詞庫 (2019)。手把手教你開發 CNN LSTM 模型，並應用在 Keras 中 (附程式碼)。2022 年 3 月 15 日，取自 <https://www.itread01.com/hkhKyqxe.html>
3. 江福財 (2017)。運用田口最佳化 Tc-99m 偽影像品質評估骨鬆性脊椎骨折。中臺科技大學醫學影像暨放射科學系暨研究所博士論文。
4. 伊雲谷 (2021)。你知道機器學習 (Machine Learning)，有幾種學習方式嗎？2022 年 4 月 20 日，取自 <https://www.ecloudvalley.com/zh-hant/machine-learning/>
5. 周東毅 (2015)。應用田口工程計畫法於高功率 LED 路燈驅動源效率改善之研究。國立高雄應用科技大學電機工程系碩士班碩士論文。
6. 林春妙 (2012)。應用田口方法於咖啡沖製最佳化。南華大學資訊管理系碩士論文。
7. 林逸青、謝孟芬、徐旺興(2019)。以深度學習建構股價預測模型：以台灣股票市場為例。當代商管論叢，4 (1)，35-59。
8. 金香君 (2019)。以田口方法改善航太零件鑽孔位置度的加工精度之研究。國立高雄科技大學工業工程與管理系碩士論文。
9. 許哲昇、江振瑞 (2018, January)。基於長短期記憶遞迴神經網路深度

- 學習之剩餘可用壽命預測。In *NCS 2017 全國計算機會議* (761-766)。國立東華大學。
10. 張英彬、駱景堯 (2006)。田口方法應用在鋼鐵廠諧波濾波器設計之最佳化。 *技術學刊*, 21(3), 217-225。
 11. 張乃恩 (2017)。結合深度學習與主動形狀模型於心臟左心室超音波影像之追蹤。國立中正大學資訊管理系研究所碩士論文。
 12. 陳宇祥 (2015)。田口方法應用於激磁同步風力發電機最大功率追蹤策略之最佳參數設計。國立高雄第一科技大學電機工程研究所碩士班碩士論文。
 13. 葉大銘 (2017)。利用田口分析法對數位乳房假體影像品質的最佳化與臨床驗證。中臺科技大學醫學影像暨放射科學系暨研究所博士論文。
 14. 楊雅筑 (2018)。開發具深度學習之嵌入式車道線偵測系統。國立中山大學機械與機電工程學系研究所碩士論文。
 15. 維基百科 (2010)。品質工程。2022 年 4 月 5 日，取自 <https://zh.m.wikipedia.org/zh-tw/%E5%93%81%E8%B3%AA%E5%B7%A5%E7%A8%8B>
 16. 劉冠宏 (2018)。利用深度學習技術對類比與數位電路辨識以實現混和式訊號設計自動化。國立中正大學電機工程研究所碩士論文。
 17. 鄧文淵 (2021)。Python 機器學習與深度學習特訓班 (第二版): 看得懂也會做的 AI 人工智慧實戰。基峯資訊出版。
 18. 賴銘俊 (2019)。應用田口方法提升汽車雷達天線效能最佳化之研究。國立中央大學工業管理研究所在職專班碩士論文。

二、西文文獻

1. Brownlee, J. (2017). *Long short-term memory networks with python: develop sequence prediction models with deep learning. Machine Learning Mastery.*
2. Carlyle, W. M., Montgomery, D. C., & Runger, G. C. (2000). Optimization problems and methods in quality control and improvement. *Journal of Quality Technology*, 32(1), 1-17.
3. Evan Touger (2018). *What's the Difference Between Artificial Intelligence (AI), Machine Learning, and Deep Learning?* [https:// www.prowesscorp.com /whats-the-difference-between-artificial-intelligence-ai-machine-learning-and-deep-learning/](https://www.prowesscorp.com/whats-the-difference-between-artificial-intelligence-ai-machine-learning-and-deep-learning/)
4. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
5. Kackar, R. N. (1985). Off-line quality control, parameter design, and the Taguchi method. *journal of Quality Technology*, 17(4), 176-188.
6. Maghsoodloo, S. (1990). The exact relation of Taguchi's signal-to-noise ratio to his quality loss function. *Journal of Quality Technology*, 22(1), 57-67.