Journal of Cheng-Shiu Institute of Technology, Vol. 34, pp. 45-54 (2021)

感應爐溫度分類

余元利
正修科技大學機械工程系

摘 要

本文使用具有兩層隱藏層的多層感知器做為深度學習的模型。標籤集有 8 個類別，共有 230 個檔案，總計 1745 筆，而測試集有 36 個檔案，總計 276 筆。資料預處理使每筆的特徵欄位數量為 50、標籤欄位數量為 8，標籤欄位採用一位有效編碼。標籤集的 80%作為訓練、20%作為驗證，評估模型的指標是準確率，分類結果顯示：訓練集與驗證集的準確率分別為 100%與 99.7%。

關鍵詞：多層感知器、一位有效編碼、準確率。

Classification of Induction Furnace Temperature

Yen-Li Yih
Department of Mechanical Engineering
Cheng-Shiu University

ABSTRACT

This paper uses Multilayer Perceptron with two hidden layers as a deep learning model. The labeled dataset has 8 categories, a total of 230 files, a total of 1745 records. The test dataset has 36 files, a total of 276 records. Data preprocessing makes the number of feature columns for each record 50 and the number of label columns 8. The label columns use One-hot encoding. 80% of the labeled dataset is used for training and 20% is used for validation. The metric of evaluating the model is accuracy. The classification results show that the accuracy of training dataset and validation dataset are 100% and 99.7% respectively.

Key Words: Multilayer Perceptron, One-hot encoding, accuracy.

一、緒論

深度學習是人工智慧中，成長最快的領域，常見的架構有多層感知器(MLP)、卷積神經網路(CNN)、遞迴神經網路(RNN)。神經網路[1]-[2]技術早已存在，由於神經網路層數不多(1~3 層)，導致模型不佳；在 1980~2000 年左右轉而流行支援向量機[3]-[4]。

隨著電腦各方面技術的提升，1998 年 Lecun, Bottou, Bengio, and Haffner [5]提出 LeNet 神經網路架構，應用於手寫數字辨識，是卷積神經網路的始祖，隱藏層的激活函數是使用 sigmoid 而

非 relu，包含兩層卷積層和三層全連接層；2012 年 Krizhevsky, Sutskever, and Hinton [6]使用了 NVIDIA 的 GPU 訓練了 8 層的網路「AlexNet」，在當時影像辨識比賽以將近 10%的差距拿下冠軍，是帶動深度學習的重要關鍵；2014 年 Simonyan, and Zisserman [7]提出 VGG 神經網路架構，特點是使用多個 3×3 的卷積層堆疊，來達到 5×5、7×7、9×9 相同的視野，但使用的參數數量卻可以大幅減少；2014 年 Szegedy et al. [8]提出 GoogLeNet 神經網路架構，最大貢獻在於提出全新的架構「Inception Block」；2015 年 He, Zhang, Ren, and Sun [9]提出 ResNet 神經網路架構，可以解決深層網路模型所遇到的退化問題(Degradation problem)；2011 年 Glorot, Bordes, and Bengio [10]使用 relu 函數，可以有效避免梯度消失的問題；神經網路目的是找合適的權重與偏置值使損失函數的損失值越來越小，常用的優化演算法有 GD、SGD、Momentum [11]、AdaGrad [12]和 Adam [13]等。Momentum 加入運動量的概念；AdaGrad 會根據梯度來調整學習率；Adam 可以視為 Momentum 和 AdaGrad 的結合，面對大多數的問題 Adam 優化器都可以達到不錯的效果；2018 年 Recht, Roelofs, Schmidt, and Shankar [14]將訓練資料切分出驗證資料，來做為架構修改的指標，以避免針對測試指標直接進行設計所造成的「元過擬合」(meta-overfitting)，如此測試資料所給出的準確度，才會更加符合網路於真實資料下的表現；1992 年 Krogh, and Hertz [15]提出權重正規化法(weight decay)，來解決過擬合問題；2014 年 Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov [16]提出捨棄模型參數，每次訓練時會隨機捨棄模型中部份參數，如此將可以有效避免神經網路太過依賴局部特徵。1986 年 Rumelhart, Hinton, and Williams [17]提出「反向傳遞」(Backpropagation)，它是一種結合梯度下降法更新神經網路權重的方法；2010 年 Glorot, and Bengio [18]提出權重初始化方法，該方法被廣泛使用在許多深度學習框架；2015 年 He, Zhang, Ren, and Sun [19]提出初始化，該方法為 Glorot 初始化的變形，可以解決 Glorot 初始化在 relu 激活函數時梯度消失的問題；2015 年 Ioffe, and Szegedy [20]提出批次正規化(Batch Normalization)，它不僅可以讓網路的收斂速度加快，還可以在一定程度緩解梯度消失和梯度爆炸的問題，進而讓網路訓練更加容易和穩定。

二、研究動機

以「深度學習」為核心的人工智慧早已進入你我的生活中，例如：手機的語音助理、人臉識別、自動篩選有趣的新聞、影音平台的每日推薦、智慧居家照護系統、智慧視訊控制系統等等。其實人工智慧才剛起步而已，根據研究機構 Tractica 預估，全球 AI 的市場規模，將從 2018 年的 81 億美元成長至 2025 年的 1058 億美元，並且能夠應用到更多的產業，例如：汽車、零售、醫療、商業、電信、消費、廣告、法律、保險等。

深度學習之所以這麼受到矚目，據說是因為 2012 年舉辦的大型視覺辨識競賽 ILSVRC (ImageNet Large Scale Visual Recognition Challenge)的關係，那年的競賽中，深度學習的手法獲得壓倒性的優異成績。ILSVR 自 2010 年開辦以來，全球各知名 AI 企業莫不以取得此項比賽最高名次為殊榮，以宣告其圖像辨識技術已達登峰之境。剛開始是由機器學習及支持向量機等技術逐鹿，然而就在 2012 年，深度學習之父 Hinto 的高徒 Alex Krizhevsky 首次採用深度學習架構參與此競賽，並以極大的差距擊敗了使用支持向量機技術 Xerox Research Centre Europe 隊伍，自此以後，揭開了深度學習吸引全球關注嶄露頭角的布幔。ILSVRC 競賽所使用的資料集來自於

ImageNet，每年會從超過 1400 萬張 full-sized 且標籤的相片中，取出部份樣本進行比賽。競賽中評比的 Top-5 error rate 分數，其計算方式是每位參賽者針對某張圖片進行預測，所給出的五個最有可能的預測中，若有一個為正確就算答對，若沒有一個正確則算錯誤。

下面介紹自 2012 年開始展露頭角，並技冠群雄的歷屆深度學習模型。2012 年於競賽中初試啼聲的深度學習網路 Alexnet，以 Top-5 錯誤率 15.4% 超過 10%的懸殊差距，輕取 Xerox 團隊所使用的支持向量機技術，而奪下桂冠；Alexnet 層數不多僅有八層，架構相當類似 Yann LeCun 用於識別手寫數字的 LeNet，但是 Alexnet 首度使用了下列幾項影響深度學習深遠的技術，且沿用至今：1. 使用 ReLu 取代了 Sigmoid 及 Tanh；2. 使用 Dropout 技術；3. 使用 Image augmentation 技術；4. Pooling 採用 max pooling；5. 使用兩片 GTX 580 GPU，針對 1,500 萬張相片、22,000 種類別，運行一個星期訓練完成。2013 的冠軍是由來自 New York University 的 Matthew Zeiler 及 Rob Fergus 取得，他們建構的模型稱為 ZFnet，成績為 11.2%；ZFnet 架構是由 Alexnet 修改而來，因此兩者架構相當近似，其差異在於：1. 不同於 Alexnet 第一層卷積使用 11×11 的 filter，ZFnet 改用較小的 7×7 filter，以保留更多的原始圖片資訊輸入；2. 隨著層數加深，應用更多數量的 filters，例如，從第 3 至第 5 層的卷積，Alexnet 用了 384、384、256 個 filters，ZFnet 則增加到 512、1024、512；3. ZFnet 亦使用 relu 作為 activate，但 Loss function 則使用 cross-entropy loss。2014 年的冠軍是 Google 提出的 GoogLeNet，創新之處在於大量使用 Inception，它是一種 network in network 的架構，針對輸入資料同時併行不同 filter 尺寸的卷積處理和 max pooling，最後再進行級聯，這讓系統在同一層能取得不同 level 的特徵。2015 年的冠軍是微軟研究院開發的 ResNet，中文多半翻譯為殘差網路，創新之處在於解決當神經網絡的深度持續增加時，所出現的 Degradation 問題，亦即準確率隨著深度增加後，到了某個深度後會達到飽和無法提昇，若再持續增加深度，反而會導致準確率下降，其原因不在 over-fitting，而是增加 training layers 反而帶來的 training errors。2016 年是由中國的 CUImage 提出的 GBD-Net 拿下，但由於僅僅較前一年的 ResNet 提昇了 2.2%，且也無值得稱頌的創新概念，因此 GBD-Net 無法如同歷屆其它 model 一樣成為經典之作。2017 年是由新加坡國立大學與奇虎 360 合作的 SeNet 以 2.3% top-5 error rate 取得冠軍，錯誤率較前兩年的 ResNet 減少了 36%，創新亮點在於依據 loss function 的學習，來調整不同屬性的特徵權重，讓有效的 feature map 權重加大，無效或效果小的 feature map 權重變小，使得模型訓練達到更好的結果。

由於深度學習技術的日益發展，在 ILSVRC 的比賽成績屢創佳績，其錯誤率已經遠低於人類視覺，是故大家對電腦視覺技術的期待，由相當成熟的 image identification 轉向尚待開發的 image understanding。本研究之動機即希望能藉由「感應爐溫度分類」計畫之執行，來學習深度學習。

三、研究目的與方法

1. 研究目的

「2019 全國智慧製造大數據分析競賽」為行政院科技會報辦公室及教育部共同指導舉辦，由漢翔航空、東台精機、公準精密、上銀科技及儀科中心贊助競賽獎金，並提供實證場域數據，透過產業出題促進國內大學校院學生及企業、學研機構接軌產業的實際應用，以創造產官學研互

動交流與學習成長的新模式。競賽總獎金高達 220 萬，參賽團隊分為「企業與學研機構組」及「大專與研究生組」二大組別，每組首獎獎金新台幣 50 萬元，並頒給教育部獎狀。

「企業與學研機構組」獲獎團隊：IVC(中央研究院)、油戲戰隊 Beta(台灣中油股份有限公司)；「大專與研究生組」獲獎團隊：佬酥貢愛蛀蟻哦(國立高雄科技大學)、50 萬的三用電錶 (國立中央大學)、神奇小貓(國立中山大學)、noname (國立中興大學)。

本研究計畫之目的，是希望透過計畫的執行，來評估參加「2020 全國智慧製造大數據分析競賽」的可行性。

2. 研究方法

深度學習架構如圖 1 所示。

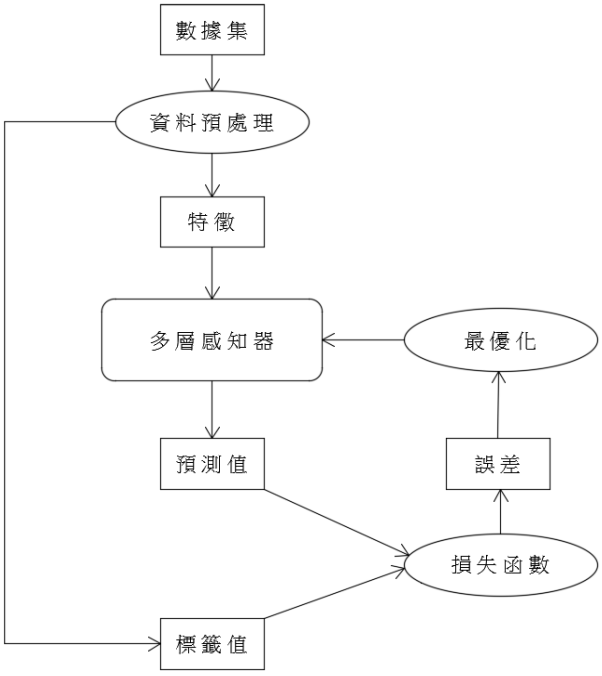


圖 1 深度學習架構

(1)數據集

數據集是記事本檔案，包括標籤集與測試集。

(2)資料預處理

a. 同一個檔案內，若每行內相鄰兩列溫差絕對值大於25，下列溫度用同列其它溫度的平均值取代。

b. 找出標籤集的最高溫度370.1與最低溫度63.1。

c. 每筆資料特徵欄位數量都是50，即C0~C49。

C0：大於(70+48*6)的溫度個數。

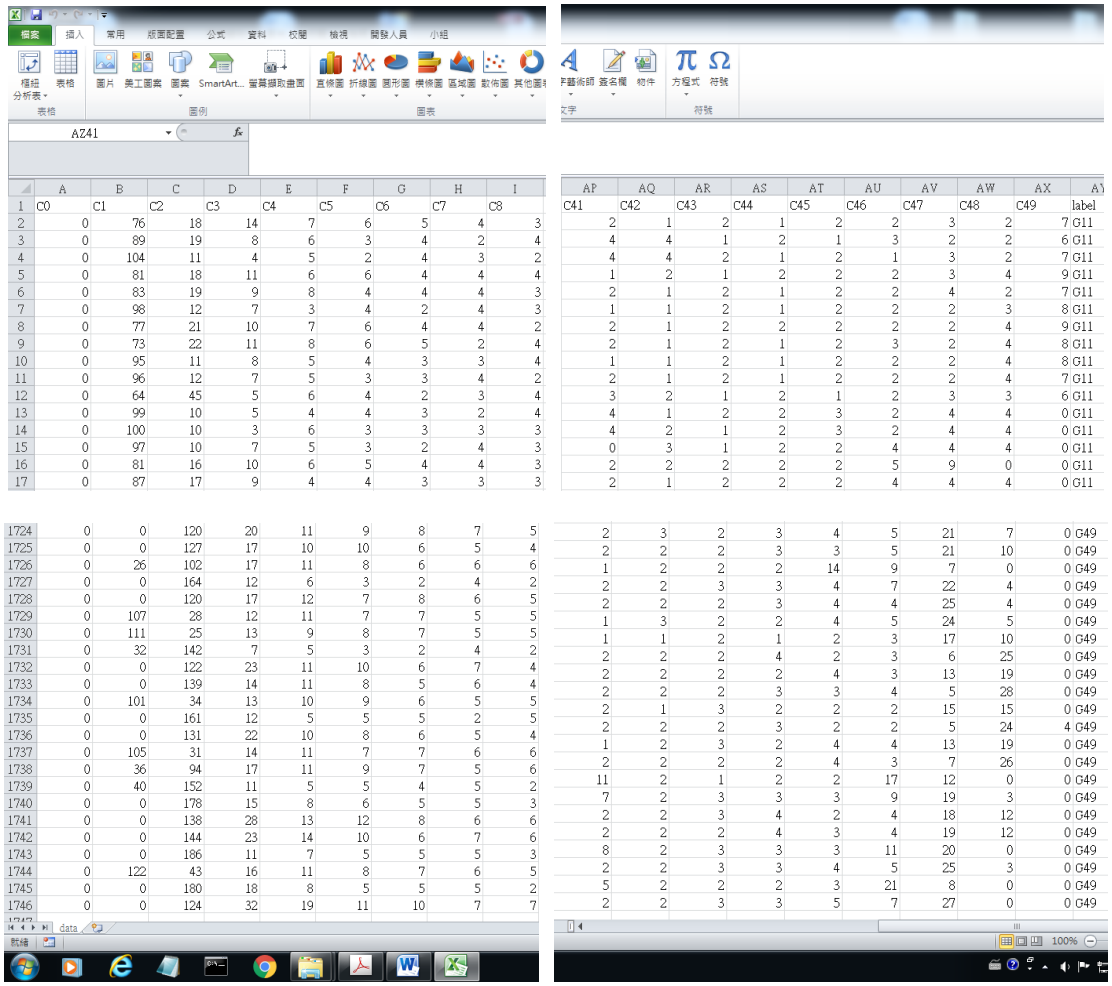
Cj：介於(70+(48-j)*6)和(70+(49-j)*6)之間的溫度個數，其中j=1, 2, …, 48。

C49：小於70的溫度個數。

d. 標上標籤。

e. 資料預處理結果如表一所示。

表一 資料預處理結果



(3)多層感知器模型如圖 2 所示。

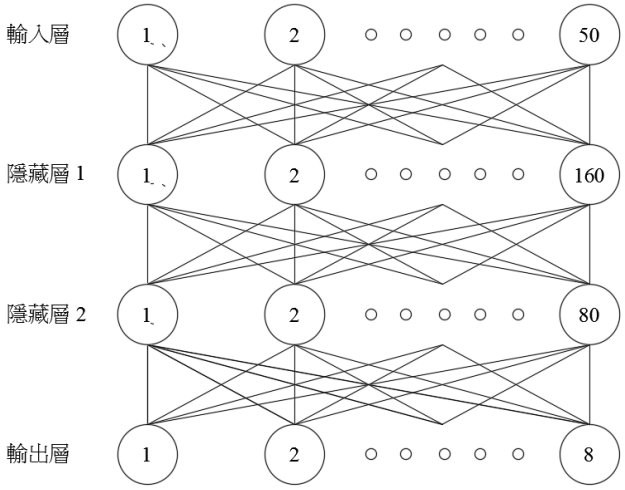


圖 2 多層感知器模型

電腦作深度學習時所需的計算，包括前向傳播與反向傳播。前向傳播計算是從輸入層開始，逐漸往輸出層進行傳播，輸入層神經元對應的權重進行加權和運算後，加上一個偏置項，再通過一個非線性激活函數，得到第一隱藏層神經元的輸出，同理得到第二隱藏層、輸出層的結果，最後再根據輸出層的預測值與標籤值之間的差異來計算損失函數；反向傳播是計算損失函數對權重和偏置的梯度，進而更新權重和偏置，計算是從更新輸出層與第二隱藏層之間的權重和偏置開始，其次更新第二隱藏層與第一隱藏層之間的權重和偏置，最後再更新第一隱藏層與輸入層之間的權重和偏置。迭代至損失函數收斂為止。

a. 前向傳播

輸入層 **X** 為 50 個神經元的 1×50 矩陣；第一隱藏層 $\mathbf{H_1}$ 為 160 個神經元的 1×160 矩陣，權重 **U** 是 50×160 矩陣，偏置 $\mathbf{B_1}$ 是 1×160 矩陣，激活函數 relu；第二隱藏層 $\mathbf{H_2}$ 為 80 個神經元的 1×80 矩陣，權重 **V** 是 160×80 矩陣，偏置 $\mathbf{B_2}$ 是 1×80 矩陣，激活函數 relu；輸出層 **Y** 為 8 個神經元的 1×8 矩陣，權重 **W** 是 80×8 矩陣，偏置 $\mathbf{B_3}$ 是 1×8 矩陣，激活函數 softmax；標籤 **T** 以 One-hot encoding 表示；L 是損失函數，使用 Categorical Cross-Entropy 作為損失函數；C 是類別數量；N 是一個批次的資料量。

$$\mathbf{H_1} = \text{relu}\,(\mathbf{X}\,\mathbf{U} + \mathbf{B_1}) = \text{relu}(\,\mathbf{NET_{H1}}\,) \tag{1}$$

$$\mathbf{H_2} = \text{relu}\,(\mathbf{H_1}\,\mathbf{V} + \mathbf{B_2}) = \text{relu}(\,\mathbf{NET_{H2}}\,) \tag{2}$$

$$\mathbf{Y} = \text{softmax}\,(\mathbf{H_2}\,\mathbf{W} + \mathbf{B_3}) = \text{softmax}(\,\mathbf{NET_Y}\,) \tag{3}$$

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} t_{i,j}\log y_{i,j} \tag{4}$$

b. 反向傳播

$$\Delta\mathbf{W} = -\eta\frac{\partial L}{\partial \mathbf{W}} = -\eta\left[\frac{\partial \mathbf{NET_Y}}{\partial \mathbf{W}}\left(\frac{\partial \mathbf{Y}}{\partial \mathbf{NET_Y}} \circ \frac{\partial L}{\partial \mathbf{Y}}\right)\right] \tag{5}$$

$$\Delta\mathbf{V} = -\eta\frac{\partial L}{\partial \mathbf{V}} = -\eta\left(\frac{\partial \mathbf{NET_{H2}}}{\partial \mathbf{V}}\left\{\frac{\partial \mathbf{H_2}}{\partial \mathbf{NET_{H2}}} \circ \left[\left(\frac{\partial \mathbf{Y}}{\partial \mathbf{NET_Y}} \circ \frac{\partial L}{\partial \mathbf{Y}}\right)\frac{\partial \mathbf{NET_Y}}{\partial \mathbf{H_2}}\right]\right\}\right) \tag{6}$$

$$\Delta\mathbf{U} = -\eta\frac{\partial L}{\partial \mathbf{U}} = -\eta\left(\frac{\partial \mathbf{NET_{H1}}}{\partial \mathbf{U}}\left(\frac{\partial \mathbf{H_1}}{\partial \mathbf{NET_{H1}}} \circ \left(\left\{\frac{\partial \mathbf{H_2}}{\partial \mathbf{NET_{H2}}} \circ \left[\left(\frac{\partial \mathbf{Y}}{\partial \mathbf{NET_Y}} \circ \frac{\partial L}{\partial \mathbf{Y}}\right)\frac{\partial \mathbf{NET_Y}}{\partial \mathbf{H_2}}\right]\right\}\frac{\partial \mathbf{NET_{H2}}}{\partial \mathbf{H_1}}\right)\right)\right) \tag{7}$$

η：學習率；$\circ$: Hadamard Product ，即矩陣中對應的元素相乘。

四、結果與討論

標籤欄位原本是 G11、G15、G17、G19、G32、G34、G48、G49 等 8 個類別。為了將類別的資料轉成數字，而讓程式能夠更好的去理解及運算，使用 One-hot-encoding 轉換為 8 個 0 或 1 的組合。G11 轉換後是 10000000(類別 0)；G15 轉換後是 01000000(類別 1)；G17 轉換後是 00100000(類別 2)；G19 轉換後是 00010000(類別 3)；G32 轉換後是 00001000(類別 4)；G34 轉換後是 00000100(類別 5)；G48 轉換後是 00000010(類別 6)；G49 轉換後是 00000001(類別 7)。分類模型的評估採用準確率，其定義為分類正確的樣本數占總樣本數的比例。

1. 訓練集

訓練集包含類別「0」116 筆、類別「1」169 筆、類別「2」91 筆、類別「3」200 筆、類別「4」203 筆、類別「5」209 筆、類別「6」192 筆、類別「7」216 筆，計 1396 筆，準確率 100%，混淆矩陣如表二所示。

表二 訓練集混淆矩陣

prediction train_label	0	1	2	3	4	5	6	7
0	116	0	0	0	0	0	0	0
1	0	169	0	0	0	0	0	0
2	0	0	91	0	0	0	0	0
3	0	0	0	200	0	0	0	0
4	0	0	0	0	203	0	0	0
5	0	0	0	0	0	209	0	0
6	0	0	0	0	0	0	192	0
7	0	0	0	0	0	0	0	216

2. 驗證集

驗證集包含類別「0」29 筆、類別「1」38 筆、類別「2」28 筆、類別「3」38 筆、類別「4」53 筆、類別「5」55 筆、類別「6」52 筆、類別「7」56 筆，計 349 筆，只有 1 筆預測錯誤，準確率 99.7%，混淆矩陣如表三所示。

表三 驗證集混淆矩陣

prediction val_label	0	1	2	3	4	5	6	7
0	29	0	0	0	0	0	0	0
1	0	38	0	0	0	0	0	0
2	0	0	27	1	0	0	0	0
3	0	0	0	38	0	0	0	0
4	0	0	0	0	53	0	0	0
5	0	0	0	0	0	55	0	0
6	0	0	0	0	0	0	52	0
7	0	0	0	0	0	0	0	56

3. 測試集

測試集每個檔案有 5~8 筆數據，同一檔案的數筆數據屬於同一類別，預測方法如下：

(1)把每個檔案的數筆數據放入已訓練好的模型去預測。

(2)把每個檔案數筆數據預測結果的最多數當成該檔案的類別。

(3)36 個檔案預測結果如下：

[0 0 0 0 0]、[0 0 0 0 0]、[1 1 1 1 1 1 1 1]、[1 1 1 1 1 1]、[1 1 1 1 1 1]、[1 1 1 1 1 1 1 1]、[1 1 1 1 1 1]、[1 1 1 1 1 1 1 1]、[2 2 2 2 2 2 2 2]、[2 2 2 2 2 2 2 2]、[3 3 3 3 3 3 3 3]、[3 3 3 3 3 3 3 3]、[4 4 4 4 4 4 4 4]、[4 4 4 4 4 4 4 4]、[4 4 4 4 4 4 4 4]、[4 4 4 4 4 4 4 4]、[4 4 4 4 4 4 4 4]、[4 4 4 4 4 4 4 4]、[5 5 5 5 5 5 5 5]、[5 5 5 5 5 5 5 5]、[5 5 5 5 5 5 5 5]、[5 5 5 5 5 5 5 5]、[5 5 5 5 5 5 5 5]、[5 5 5 5 5 5 5 5]、[6 6 6 6 6 6 6 6]、[6 6 6 6 6 6 6 6]、[6 6 6 6 6 6 6 6]、[6 6 6 6 6 6 6 6]、[6 6 6 6 6 6 6 6]、[6 6 6 6 6 6 6 6]、[7 7 7 7 7 7 7 7]、[7 7 7 7 7 7 7 7]、[7 7 7 7 7 7 7 7]、[7 7 7 7 7 7 7 7]、[7 7 7 7 7 7 7 7]、[7 7 7 7 7 7 7 7]。

五、結論

本文數據來自「2019 全國智慧製造大數據分析競賽-初賽測驗數據。」電腦作深度學習時所需的時間 7.691000 秒，所需的資源如下：中央處理器 Intel Core i7-7700 3.60 GHz、記憶體 8.00 GB、作業系統 Windows 7 企業版、keras 模組、pandas 模組、numpy 模組、glob 模組、csv 模組、time 模組及 python 軟體。數據集劃分為訓練集、驗證集與測試集，使用 keras Sequential 建立模型、設定 adam 為模型訓練最優化方法，分類結果顯示：測試集 276 筆，準確率 100%。

參考文獻

1. W. S. McCulloch and W. Pitts, ”A logical calculus of the ideas immanent in nervous activity, ” *The bulletin of mathematical biophysics*, vol.5, no.4, pp. 115-133 (1943).
2. F. Rosenblat, ”The perceptron: A probabilistic model for information storage and organization in the brain, ” *Psychological Review*, vol.65, no.6, pp. 386-408 (1958).
3. C. Cortes and V. Vapnik, ”Support-Vector Network, ” *Machine Learning*, pp. 273-297 (1995).
4. T. Joachims, ”Text categorization with support vector machines: Learning with many relevant features, ” European Conference on Machine Learning, pp. 137-142 (1998).
5. 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 (1998).
6. A. Krizhevsky, I. Sutskever, and G. E. Hinton, ”Imagenet classification with deep convolutional neural networks, ” *Advances in Neural Information Processing System*, pp. 1097-1105 (2012).
7. K. Simonyan and A. Zisserman, ”Very deep convolutional networks for large-scale image recognition, ” International Conference on Learning Representations, pp. 1-14 (2015).
8. C. Szegedy et al., ”Going deeper with convolutions, ” IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9 (2015).

9. K. He, X. Zang, S. Ren, and J. Sun, ”Deep residual learning for image recognition, ” Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778 (2016).
10. X. Glorot, A. Bordes, and Y. Bengio, ”Deep sparse rectifier neural networks, ” International Conference on Artificial Intelligence and Statistics, pp. 315-323 (2011).
11. I. Sutskever, J. Martens, G. Dahl, and G. Hinton, ”On the importance of initialization and momentum in deep learning, ” International Conference on Machine Learning, pp. 1139-1147 (2013).
12. J. Duchi, E. Hazan, and Y. Singer, ”Adaptive subgradient methods for online learning and stochastic optimization, ” *Journal of Machine Learning Research*, pp. 2121-2159 (2011).
13. B. P. Kingma and J. Ba, ”Adam: A method for stochastic optimization, ” International Conference for Learning Representation (2015).
14. B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, ”Do cifar-10 classifiers generalize to cifar-10?, “ arXiv:1806.00451 (2018).
15. A. Krogh and J. A. Hertz, ”A simple weight decay can improve generalization, ” *Advances in neural Information processing systems*, pp. 950-957 (1992).
16. N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting, “ *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958 (2014).
17. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagation error, “ *Nature*, vol. 323, pp. 533-536 (1986).
18. X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks, “ Proc. Conf. Artificial Intelligence and Statistics, pp. 249-256 (2010).
19. K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, “ IEEE International Conference on Computer Vision, pp. 1026-1034 (2015).
20. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift, “ International Conference on Machine Learning, pp. 448-456 (2015).