

監督式學習之類神經網路的原理及應用

林哲兆

January 5, 2024

近年神經網路的學習方法崛起，產生如 CNN、RNN、ANN 等多種神經網路的分支，而本文將介紹類神經網路 (ANN)，一種在 90 年代相當熱門的學習器，過去礙於硬體設備限制了 ANN 的發展，直至最近才因為設備的進步得以發揮其真正的效能。本文將藉由簡易的原理敘述，再透過資料來實際操作 ANN，以實現 ANN 的應用。

1 類神經網路之原理

作為最初始發展的神經網路，ANN 的原理並不太複雜，透過模擬人類腦神經的運作，將 p 維的資料輸入後產生 r 個輸出值，圖 1 展示了其運作原理，其中中間的神經網路又分為輸入層、隱藏層與輸出層，各層中的節點代表的是神經元，而隱藏層如何運作是不得而知的，分析人員僅知道其運作原理，至於怎麼產生最後的輸出值無法得知，但可以透過設定隱藏層的層數、神經元數與節點的激發函數，改變神經網路的輸出過程。

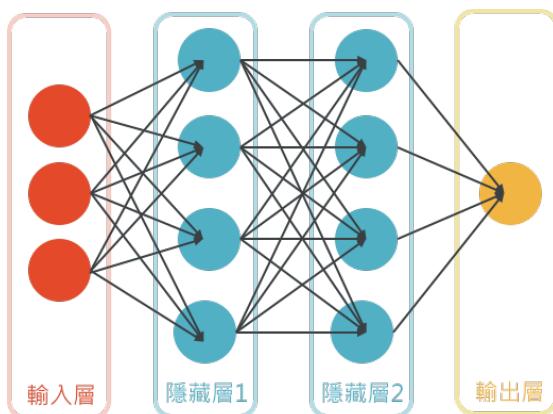


圖 1: 類神經網路的原理圖示

當我們對神經元輸入 p 個變數 ($x_1 \sim x_p$) 後，經過激發函數 (σ) 與內部迴歸模

型對輸入的權重 (w_i) 加乘，再加入偏誤 (b)，便完成了該節點的輸出。接著輸出會再傳給下一個神經元，作為該神經元的輸入值，如此一層層傳遞下去，直到最後一層的輸出層，產生預測結果，式(1)即每個節點的輸出公式：

$$f_{w,b}(x) = \sigma \left(\sum_i w_i x_i + b \right) \quad (1)$$

因此可以知道類神經網路是一種非線性的模型，且隨著資料維度越大、隱藏層越多，所需要估計的參數就更多，這正是為何 ANN 為何會遇到前述硬體設備不夠先進的問題，然而一旦設備進步，那麼其在 x, y 的適配程度非常很高的。

2 類神經網路之應用

類神經網路之應用非常廣泛，從基礎分類到複雜的影像辨識都能使用，本文將利用兩個範例：機械手臂運動、字母數字圖形辨識，來演示使用類神經網路分析的過程與結果。

2.1 機械手臂運動

圖 2 是一張平面的機械手臂運動的示意圖，其手臂有固定長度的兩截 (l1,l2)，透過改變 θ_1 與 θ_2 能控制手臂的移動方位，使得機械手臂的活動範圍會如圖 6 散佈在兩個弧形之間，其中範圍內任何一點為機械手臂鉗子的位置，而鉗子在各位置的機率皆相同。

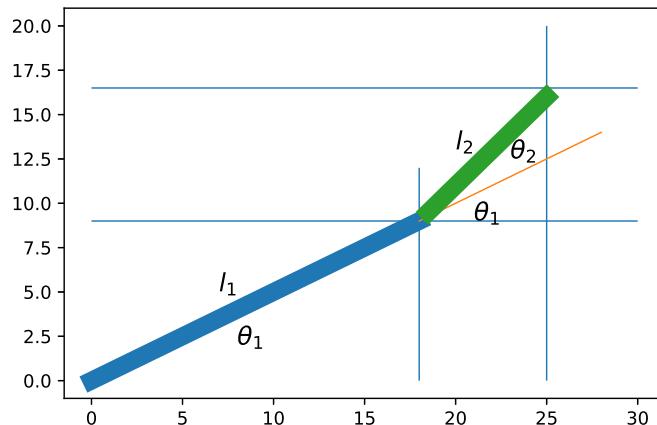


圖 2: 機械手臂示意圖

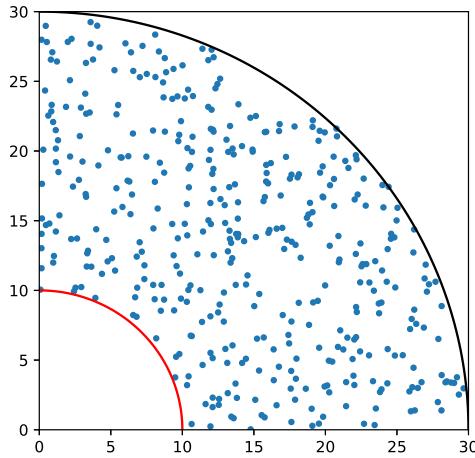


圖 3: 機械手臂活動範圍

為了訓練機械手臂能到正確位置夾取物品，因此需要讓機械手臂有座標的概念，但機械手臂能控制的只有角度，所以需要將角度透過極座標轉換成一般座標，言下之意即輸入給定座標，訓練機械手臂能產出正確的角度，最後將手臂移動到該處。為此，輸入的 x 、 y 座標將透過以下算式表達：

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

並計算反函數，得出 θ_1 、 θ_2 的方程式：

$$\begin{aligned}\theta_2 &= \cos^{-1} \left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \right) \\ \theta_1 &= \tan^{-1} \left(\frac{y}{x} \right) - \tan^{-1} \left(\frac{l_2 \sin(\theta_2)}{l_1 + l_2 \cos(\theta_2)} \right)\end{aligned}$$

如此便能將 x 、 y 座標作為輸入， θ_1 、 θ_2 作為輸出，讓機械手臂使用類神經網路進行學習。以下第一個範例使用較單純的 `linspace` 語法，生成了一個簡單的訓練資料，希望利用類神經網路，使得給定座標能夠產出正確的角度：

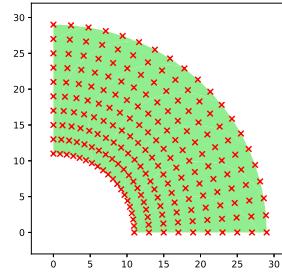


圖 4: 機械手臂分佈點的訓練資料

在將訓練資料放入類神經網路前，需要考慮到不同的參數設定，尤其在隱藏層中神經元的數量。前一小節有提到層數越高、神經元越多，需要估計的參數就越多，也就代表著運算時間會更長，因此該如何選擇隱藏層數、神經元個數是一件重要的事，最理想的情況當然是希望能在最少的層數下有最少的誤差，為此，本文在這筆資料中設定了每層 10、20、40、80 個神經元四種情況，來觀察神經元個數對模型的影響，結果如下圖所示：

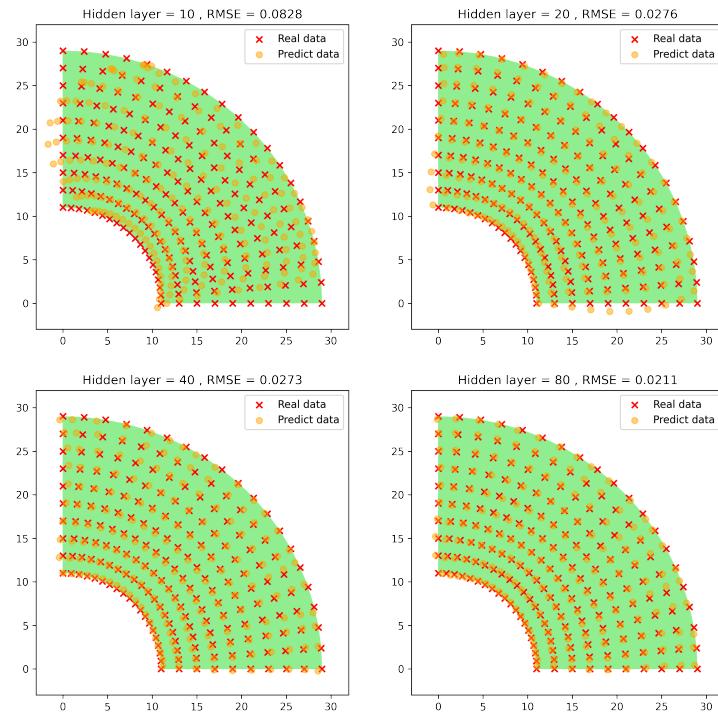


圖 5: 機械手臂在不同神經元數的訓練結果

RMSE (Root Mean Squared Error) 是用來衡量誤差的一種標準，若預測點與實際點相差大，則 RMSE 也會較大，反之則較小。由圖 5 可以看出，隨著神經元數越來越多，RMSE 有越來越小的趨勢，但變小的速度是逐漸遞減的，從 10 個到 20 個的 RMSE 跳動最大，20 個到 80 個則相差不大，因此若考慮效率的話，20 個神經元的模型會是一個好選擇，在不需要估計太多參數下，有一個相對低的 RMSE；但若講求精確性的話，RMSE 最小的 80 神經元模型依然是比較好的選擇。

這筆訓練資料的好處是較簡單，但可以看出並不符合勾爪位置隨機分佈的設定，尤其在越靠近左下方的資料點會較密集，右上方則較鬆散。為了解決該問題，於是這次使用了均勻分配的亂數，先產生一個給定圓內的均勻亂數，再透過適當裁減，將不屬於機械手臂活動範圍的樣本點刪除，最後形成如下圖的樣本：

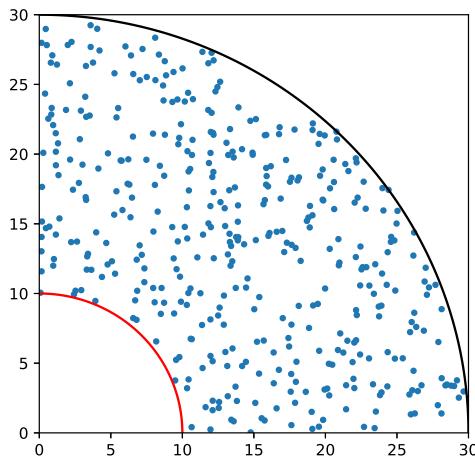


圖 6: 機械手臂分佈點的訓練資料

接著，同樣使用 10、20、40、80 四種神經元數的模型來觀察哪一種模型的 RMSE 較低，與前者不同的在於，此資料會分為訓練資料 (70%) 與測試資料 (30%)，因此 RMSE 所衡量的會是測試資料的部分，以下為訓練結果：

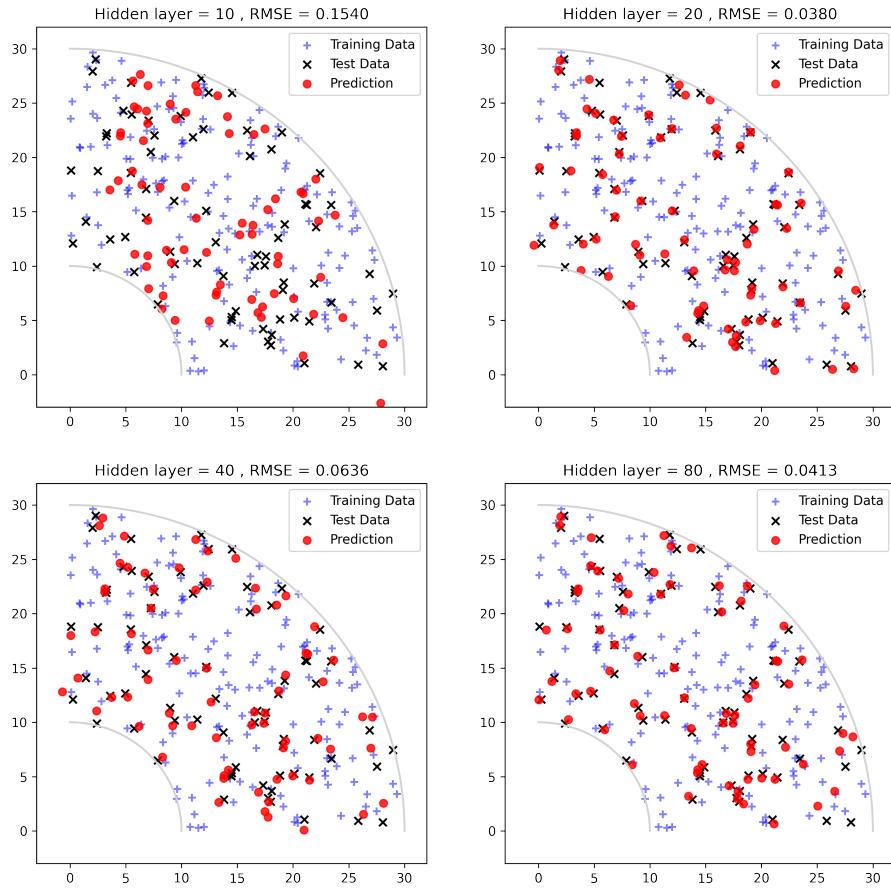


圖 7: 機械手臂分佈點的訓練資料與測試資料

在這個例子中，在 20 個神經元數模型的 RMSE 達到了最低，在 40 個與 80 個反而增加，因此可知並不是神經元個數越多，對預測結果會更適配，訓練資料的品質、分布也是一個重要的因素，所以當實務上應用時，應該多嘗試幾次，因為 ANN 的隱藏層如何運作無法得知，但訓練資料的品質是可以控制的。

因此為了增加訓練資料的參考性，本文進一步探討不同樣本數下的模型差異，也就是檢視樣本數的多寡是否會影響模型的表現，以下展示的是總樣本數為 239 與 662 下的差異：

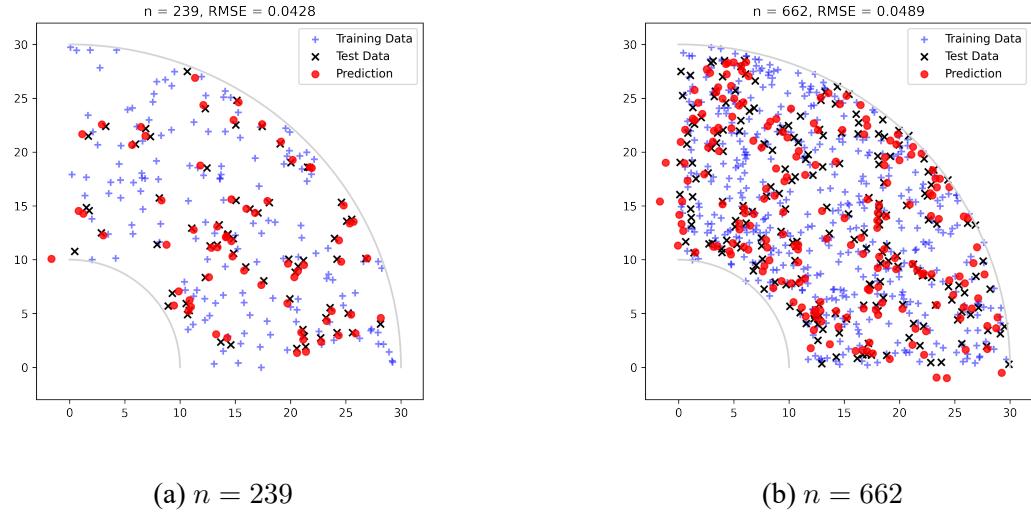


圖 8: 樣本數多寡對模型的影響

從圖 8 可以看出，樣本數增加會使得 RMSE 下降，證明了訓練資料的品質確實會影響模型的預測能力，因此神經元個數固然重要，但樣本品質也是不可忽略的一環。

上述訓練資料的程式，所使用的是 `sklearn` 中的 `MLPRegressor` 套件，除了這個套件外，`NeuroLab` 也是進行類神經網路學習的套件，其參數設定與 `MLPRegressor` 大同小異，但算法有一些不同，至於不同之處套件的開發者並沒有多著墨。`NeuroLab` 除了能進行學習外，還能把計算所需的迭代次數、誤差輸出，讓分析人員能了解一部分運算的過程。

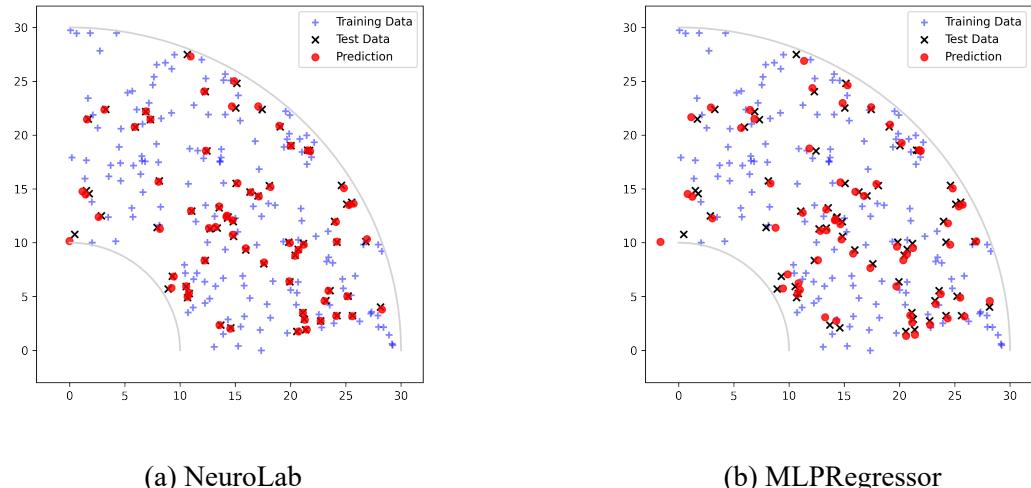


圖 9: 不同套件的展示

圖 9 (a) 是使用 NeuroLab 套件對資料做訓練，資料採用的是樣本數為 239 的樣本，神經元個數設置為 20，圖 9 (b) 則是同筆資料使用 MLP 的結果，可以看出在這筆資料中，使用 NeuroLab 的預測結果似乎比較好，大部分的預測點都比 MLPRegressor 來得更靠近資料點，所以可知不同套件也有可能影響模型的優劣。

2.2 手寫圖形辨識

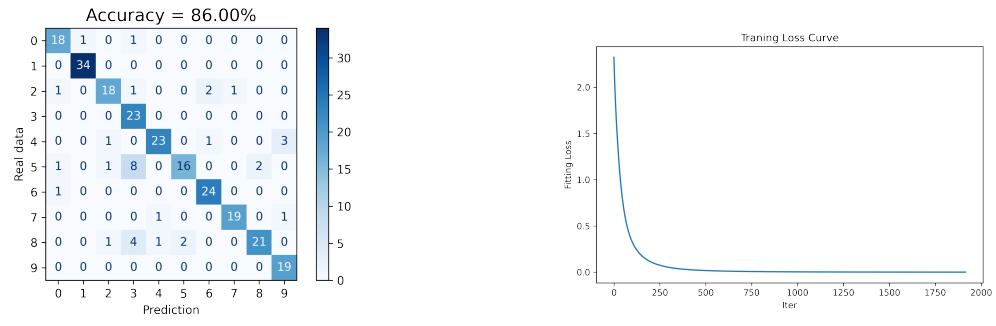
第二個例子是手寫圖形辨識，圖形辨識是神經網路最常見的用途。圖形資料的維度較大，複雜度高，但透過神經網路層次的運作結構，讓每一層都能從前一層中學習更加抽象和高級的特徵，這種階層式學習使得神經網路能夠捕捉複雜的模式和關係，並將結果輸出。



圖 10: 手寫數字資料

本例使用的資料是一筆手寫數字資料，樣本數共 1000，資料來自從美國郵局用戶蒐集的手寫數字，如圖 10 所呈現。由於每個人寫字的方式、美醜不一，所以需要一個系統來辨識肉眼無法辨認的手寫數字，因此本小節將透過手寫資料來訓練，利用類神經網路盡可能讓模型能夠正確辨認數字。

在神經元數設定為 30 下，模型的結果如下圖的矩陣所呈現，此矩陣是一個混淆矩陣 (confusion metrix)，對角線上的數字代表正確辨識的資料數，非對角線則為辨識錯誤的資料數，可以看到標題的 Accuracy 為 89.6%，表示正確辨識之資料 (對角線上的資料) 共佔了 86.0%，最容易誤判的則是 4 和 9，共四次被錯誤判斷。

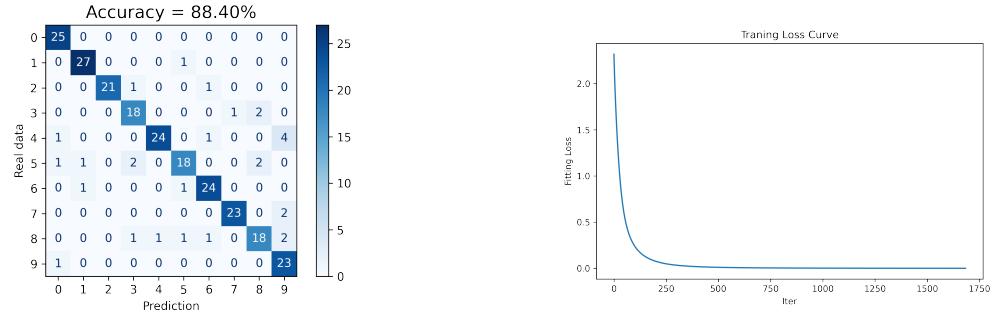


(a) 神經元數 = 30 的混淆矩陣

(b) 損失函數與迭代次數

圖 11: 手寫數字資料辨識結果 (神經元數 = 30)

接著嘗試 40 個神經元的模型：



(a) 神經元數 = 40 的混淆矩陣

(b) 損失函數與迭代次數

圖 12: 手寫數字資料辨識結果 (神經元數 = 40)

可以看到準確率從 86% 上升至 88.4%，且迭代次數甚至比神經元為 30 的模型還低，雖然不見得神經元數越多模型越好，但可知可以嘗試不同的神經元，找到一個最好的結果。

下一個例子使用的是手寫英文字母資料，依舊是利用類神經網路訓練，讓無法肉眼辨識的字母透過電腦正確辨識，但這筆資料的樣本數增加到了 10000，類別種類增加至 26，相對數字資料而言複雜許多。以下為分別採用 30 個神經元與 40 個神經元進行訓練產出的混淆矩陣：

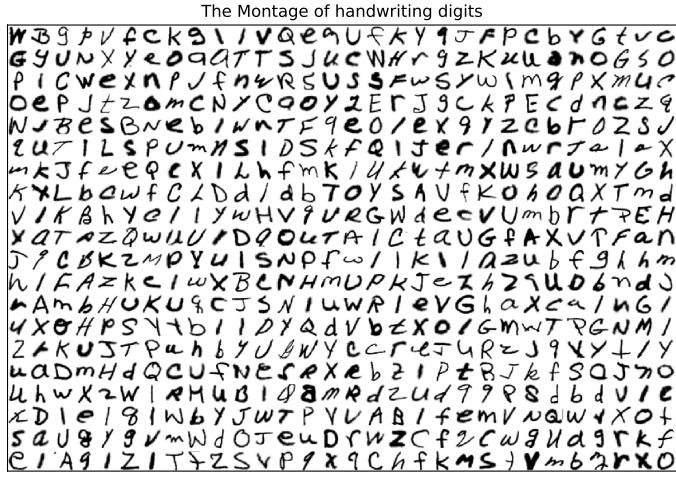


圖 13: 手寫字母資料

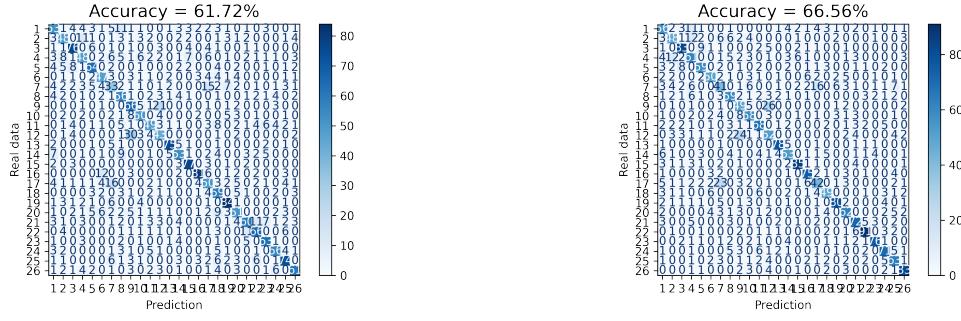


圖 14: 不同神經元的混淆矩陣比較

從圖 14 可以看出其正確率相對數字資料而言降低很多，因此可以發現在資料變得更加複雜時，正確判斷的難度會上升，並且程式執行時間由原本的 20 秒變成將近 2 分鐘，增加了將近 6 倍的時間，雖然迭代次數都是 1000 多，但每次迭代的計算時間會因為資料變得複雜而增加。另外，在本例中增加較多神經元數的模型也依舊比較低神經元的表現好，這也驗證了一開始所說，如果硬體設備足夠，那類神經網路的表現會越來越好。

3 結論

本文介紹了類神經網路的運作原理，也展示了實際使用類神經網路的分析過程，更驗證影響類神經網路表現的三個因素：神經元個數、訓練資料品質與不同套

件的差異，當然，影響模型表現的絕對不只這三個因素，在程式碼中還有許多參數可以進行微調，可能一點微調都會對模型有莫大的改善。類神經網路與過去比較簡單的模型不同，在使用較簡單模型時，分析人往往會花更多時間在解釋模型與思考模型合理性，但類神經網路則不然，由於過程不得而知，便轉而要求結果表現好、硬體設備可以承受即可，所以反而會在模型微調上下更多的功夫，這是與過去簡單模型最大的不同。希望透過本文能讓讀者了解類神經網路的背景與應用，並進一步往深度學習領域探究。