

類神經網路：解開機械手臂與圖形辨識之謎

林貫原

January 10, 2024

人工智慧的興起帶動了類神經網路 (ANN) 技術的迅速發展。這種技術模擬了人類大腦神經元的運作方式，使得機器能夠學習、理解和做出智能決策。本文將探索 ANN 在機械手臂和圖形辨識中的引人注目的應用。機械手臂的靈活性和智能化得益於 ANN 的發展，而圖形辨識技術則賦予機器理解視覺信息的能力同時，圖形辨識技術的崛起使得機器能夠理解並解釋視覺信息。讓我們一同深入瞭解，類神經網路如何推動機械手臂和圖形辨識技術的創新，並展望這些技術共同演進的未來。這是一場探索科技奇蹟的冒險，讓我們一同啟程。

1 背景介紹

神經網路是人工智慧中的一種方法，指導電腦以受人腦啟發的方式來處理資料。這是一種機器學習，使用類似於人腦分層結構中的互連節點或神經元。它會建立一種可讓電腦用來從錯誤中學習並持續改善的適應型系統。因此，人工神經網路嘗試以更高準確度來解決複雜的問題，例如總結文件或識別面部。

圖 1 描繪了一個具有單一隱藏層的典型前饋式類神經網路。其中，一些重要的數字需要特別注意。在左側的輸入端 (Input)，標示為 p 個變數 (在圖中 $p = 2$)，代表著輸入的維度。中間的隱藏層 (Hidden Layer) 包含 q 個神經元 (在圖中 $q = 10$)，而右側的輸出層 (Output Layer) 則有 r 個輸出變數 (在圖中 $r = 2$)。輸入和輸出變數的數量 p 和 r 取決於問題的結構。例如，在本文的機器手臂範例中， $p = 2$ 可能表示平面座標位置，而 $r = 2$ 可能代表機器手臂的兩個角度。值得注意的是，隱藏層和輸出層的結構，特別是隱藏層中包含的神經元數量 q 。當 q 較大時，表示輸入和輸出之間的關係較為複雜。從數學的角度來看，這代表著輸入和輸出之間的關係呈現較高的非線性程度。

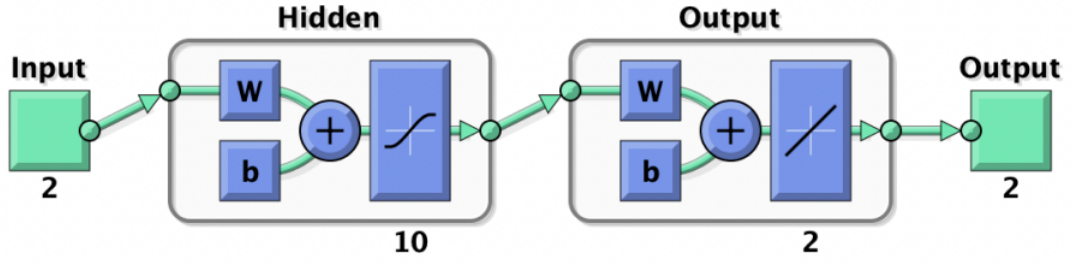


圖 1: 前饋式類神經網路示意圖

假設輸入端的 p 個變數表示為 x_1, x_2, \dots, x_p ，輸出端的 r 個變數表示為 $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_r$ ，則前饋式類神經網路的輸出與輸入間的數學關係寫成

$$\hat{y}_k = \sum_{i=1}^q \omega_{ki}^2 g \left(\sum_{j=1}^p \omega_{ij}^1 x_j \right) + b_k^2, 1 \leq k \leq r \quad (1)$$

其中函數 $g(\cdot)$ 可以選擇為 $(-1 \leq g(z) \leq 1)$

$$g(z) = c_1 \frac{1 - e^{c_2 z}}{1 + e^{c_2 z}} \quad (2)$$

函數 $g(z)$ 的形狀如同圖 1 隱藏層中所示的函數圖。這個函數 $g(z)$ 也可應用在輸出層，以增加模型的複雜度，但通常使用如圖 1 中所示的線性函數 $y = x$ ，即輸出層呈現一條直線方程式。在式 (1) 中， w_{ij}^1 與 b_i^1 代表第一個隱藏層的第 i 個神經元對應第 j 個輸入的權重係數 (Weightings) 與偏差 (Biases)。同樣地， w_{ki}^2 與 b_k^2 則是第二層 (即輸出層) 的第 k 個神經元對應前一隱藏層的第 i 個輸出的權重係數與偏差。

類神經網路根據已知的輸入與輸出資料 $x_j(n)$ 與 $y_k(n)$ ，調整權重係數 w_{ij}^1 、 w_{ki}^2 、 b_i^1 與 b_k^2 ，使得實際資料 $y_k(n)$ 與類神經網路輸出資料 $\hat{y}_k(n)$ 的誤差最小化。假設有 N 筆實際資料，類神經網路的訓練階段可以被表述為一個多變量函數最小值的問題，即

$$\min_{\Omega} e(\Omega) \quad (3)$$

其中函數函數為

$$\begin{aligned} e(\Omega) &= \sum_{n=1}^N \sum_{k=1}^r (y_k(n) - \hat{y}_k(n))^2 \\ &= \sum_{n=1}^N \sum_{k=1}^r \left(y_k(n) - \sum_{i=1}^q \omega_{ki}^2 g \left(\sum_{j=1}^p \omega_{ij}^1 x_j + b_i^1 \right) + b_k^2 \right)^2 \end{aligned} \quad (4)$$

其中參數 $\Omega = \{\omega_{ij}^1, \omega_{ki}^2, b_i^1, b_k^2\}_{i=1,2,\dots,q;j=1,2,\dots,p;k=1,2,\dots,r}$ ，共 $pq + qr + q + r$ 個參數。類神經網路被認為是高度複雜且具有高維度的模型，其包含的參數數量取決於網路的結構。以我們在圖 1 中呈現的類神經網路為例 (其中 $p = 2$, $q = 10$, $r = 2$)，誤差函數 $e(\Omega)$ 共涉及超過 100 個變數。然而，隨著類神經網路的興起，其面臨的是計算高維度最小值的挑戰。當輸入變數 (p) 和所選擇的隱藏層神經元數量 (q) 增加時，追求更完美的訓練 (即 $e(\Omega)$ 越小) 將不可避免地需要更大的計算量，這導致了計算時間的過長，讓研發人員感到望而卻步。

儘管如此，隨著電腦軟硬體的提升和演算法的成熟，類神經網路的學習能力逐漸被推向「量產」的程度。這帶動了新一輪的人工智慧潮流，特別是深度學習在機器學習中的應用，充分發揮了類神經網路的擬合能力。然而，另一個阻礙機器學習進程的因素是實際應用的可行性。儘管類神經網路在訓練階段可以將輸入與輸出之間的關係完美地擬合，但在面對實際的測試資料時，其表現可能不如預期。這一現象與隱藏層數、神經元數量和訓練程度等因素密切相關。因此，測試的多樣性和長時間的測試變得非常重要，這也導致了類神經網路的實際應用價值受到一些質疑。接下來的範例中，我們將透過機器手臂的案例展示不同訓練過程可能導致不同的測試結果。

2 機械手臂

圖 2 展現了一個擁有較低自由度 (DOF, Degree of Freedom) 的機械手臂，包含兩節手臂，分別長度為 $\theta_1 = 20$ 和 $\theta_2 = 10$ 。這兩節手臂隨著角度 θ_1 和 θ_2 的變動而運動，使得手臂最前端的位置 (x, y) 能夠覆蓋圖 3 第一象限的陰影區域。簡而言之，這表示機械手臂的前端指頭或夾具能夠觸及到該區域。

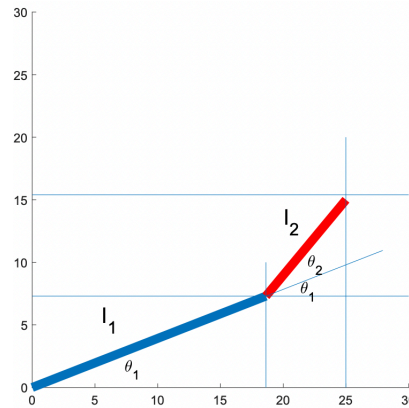


圖 2: 機械手臂角度示意圖

式 (5) 與式 (6) 描述了機械手臂如何算出 θ_1 與 θ_2 如何計算出來的解:

FKE :

$$\begin{aligned} x &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{aligned} \quad (5)$$

IKE :

$$\begin{aligned} \theta_2 &= \cos^{-1} \left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \right) \\ \theta_1 &= \tan^{-1} \left(\frac{y}{x} \right) - \tan^{-1} \left(\frac{l_2 \sin(\theta_2)}{l_1 + l_2 \cos(\theta_2)} \right) \end{aligned} \quad (6)$$

圖 3 表示著若給定一個座標 (x, y) ，第二段機械手臂所能觸及的範圍並生成均勻的點。

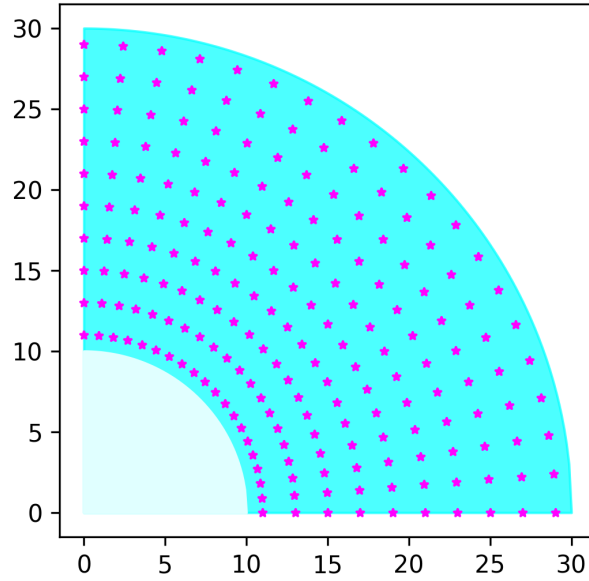


圖 3: 兩截式機械手臂所及範圍與訓練資料分布點

圖 ?? 主要是在探討如何使用類神經網路 (ANN) 來配適具有複雜形狀的資料分布，且不同神經元個數 (10,20,50,100) 的設置下，其表現的如何。先簡單介紹本文會使用到的一些指標：

- R square (R^2): 衡量模型相對於平均值的解釋力，值越接近 1 表示模型解釋方差的能力越好。

- Root Mean Square Error (RMSE): 衡量模型預測值和實際值之間誤差的平方根，值越小表示模型預測的準確性越高。
- Loss Function: 衡量模型在訓練過程中的性能，值越小表示配適效果越好。
- Number of Iterations: 表示模型訓練的迭代次數。

表 1: 神經元個數不同時配適表現指標

神經元個數 \ 指標	10 個	20 個	50 個	100 個
R square	0.9934	0.9973	0.9977	0.9976
Root Mean square error	0.0012	0.0005	0.0005	0.0005
The Loss function	0.0012	0.0008	0.0005	0.0013
Number of iterations	3339	2821	2361	1740

根據表 1 的指標，可以發現到隨著隱藏層的數量增加，表現能力也會有些許的增加，均方根誤差下降，損失函數下降，迭代次數大幅下降。

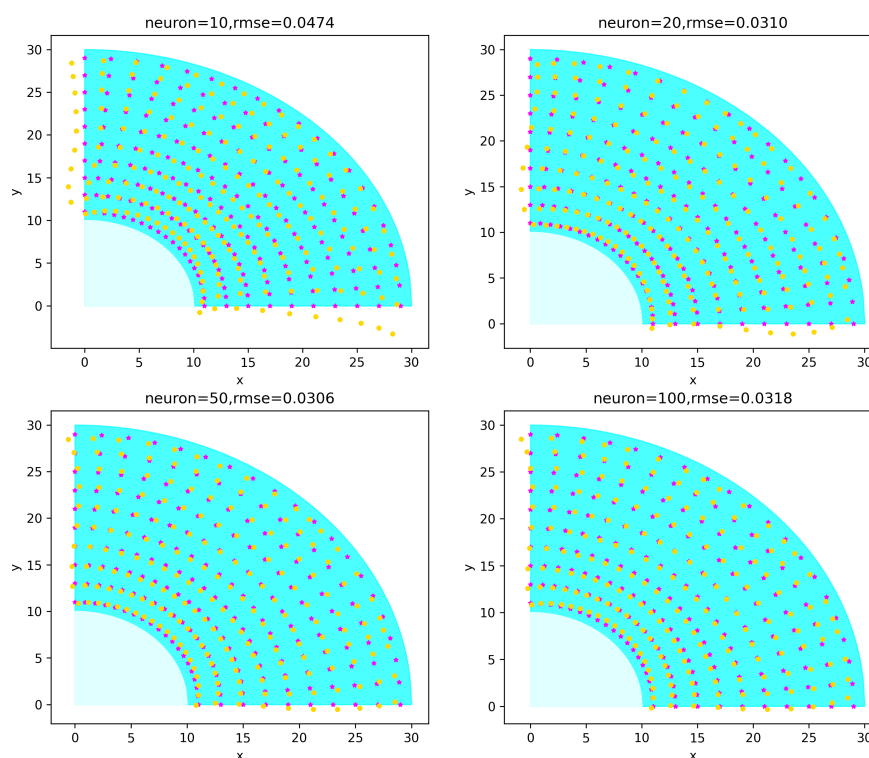


圖 4: 隱藏層不同時配適表現

接下來討論一下類神經網路的演算法，其種類有非常多種，比如說 Adagrad、RMSprop、Adadelta 等等，但此處因為我們使用的 Python 套件是 MLPRegressor，其只支援 LBFGS、SGD、ADAM 演算法，所以僅討論這三種，可以從圖 5 及表 2 發現在此處，面對隱藏層為 30 的情況下 LBFGS 的表現相對來得比較好。

表 2: 不同演算法的配適表現指標

演算法 指標	LBFGS	SGD	ADAM
R square	0.9992	0.9599	0.9951
Root Mean square error	0.0165	0.1168	0.0582
The Loss function	0.0002	0.0068	0.0017
Number of iterations	4948	4364	4411

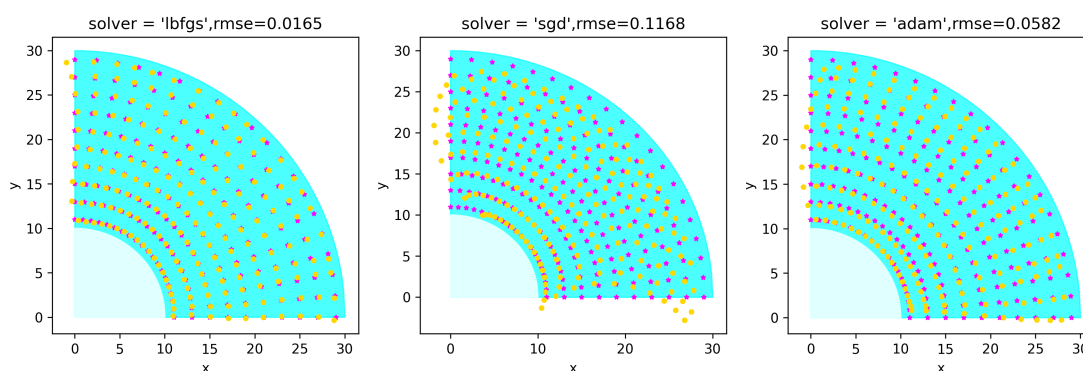


圖 5: 不同演算法的配適表現

圖 6 及圖 7 為自行運用定義函數生成出的亂數，皆有 1000 個點，至於在訓練途中，有輸出如表 3 的結果 (圖 6)，顯示了在每個訓練週期 (Epoch) 結束時的模型誤差。每一行代表一個訓練週期，而”Error”則是在該週期結束時計算得到模型的誤差值。誤差值是用來評估模型在訓練數據上的預測性能的一個指標，它表示模型預測值和實際值之間的差異程度。在這個例子中，隨著訓練週期的增加，模型的誤差逐漸降低。最後一行顯示”The goal of learning is reached”，這表示模型已經達到了預先設定的學習目標 (goal)。在這種情況下，目標可能是訓練誤差降到一個足夠小的值，以滿足模型訓練的要求，以達到更好的預測性能。

表 3: 生成扇形區亂數的訓練週期

Epoch	Error
100	0.11370709105274876
200	0.04612164994856592
300	0.033815021788076154
400	0.028178104130321305
500	0.022168348870310092
600	0.01780327294196251
700	0.016440507537187874
800	0.015011311589278726
900	0.013126301705622047
1000	0.011427963865364231
1100	0.010375537987527397

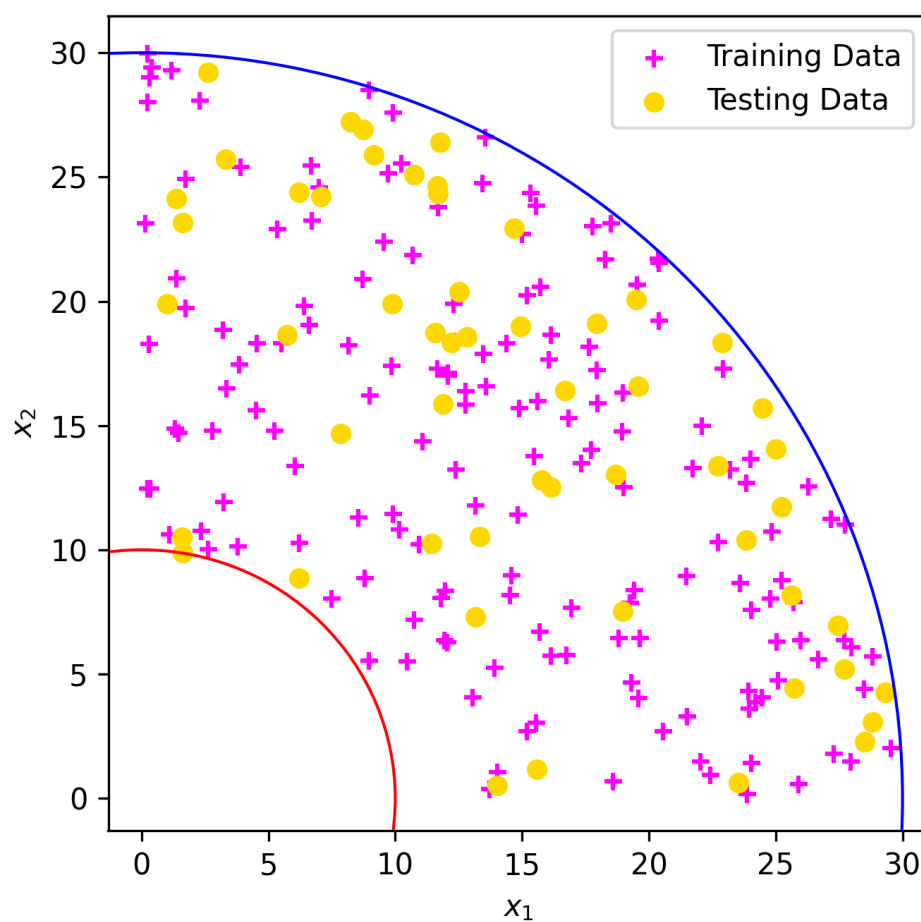


圖 6: 均勻落在扇形區的亂數

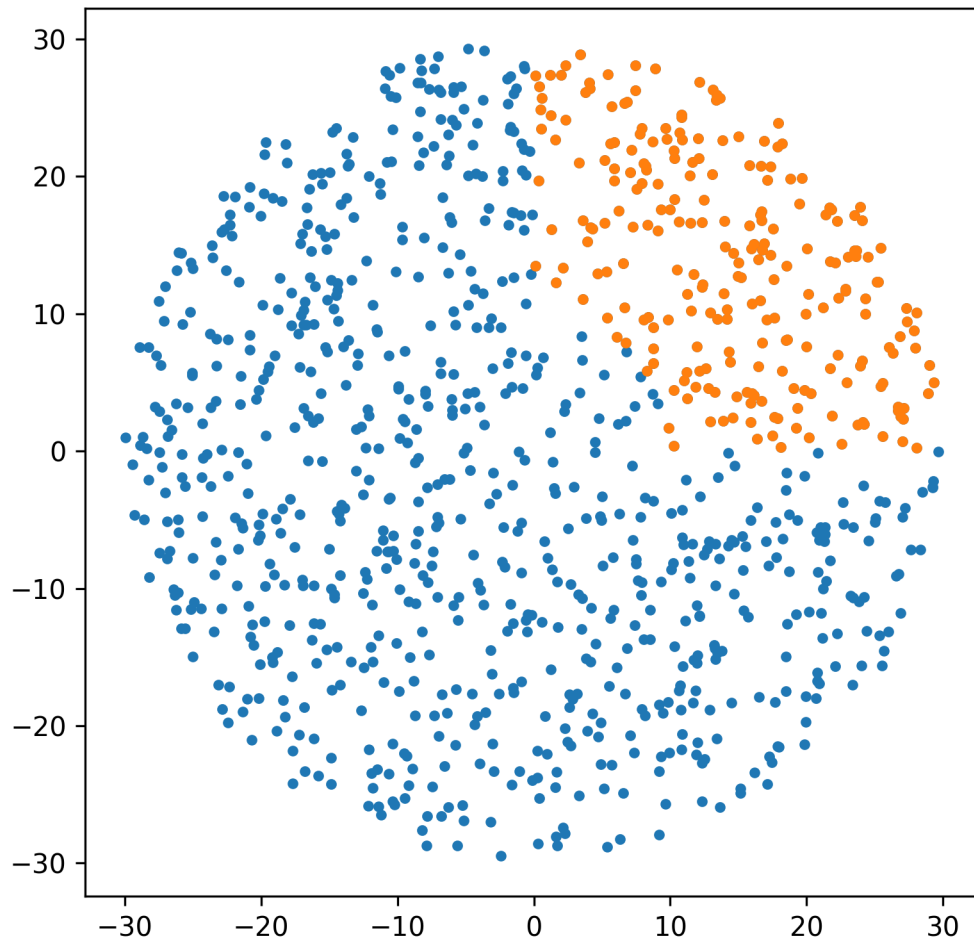


圖 7: 均勻落在圓形區的亂數

圖 8 探討在 LBFGS 演算法下，不同樣本與不同隱藏層的比較，先說明其設定配置，左上圖樣本數為 1000，神經元個數為 20，右上圖樣本數為 1000，神經元個數為 40，左下圖樣本數為 500，神經元個數為 20，右下圖樣本數為 500，神經元個數為 40。在每個子圖中，展示訓練數據訓練集為紫色十字和測試集為金色圓點，以及在測試數據上的預測結果為綠色星號。圖 9 呈現其訓練過程誤差遞減的情形。

1. 第一次訓練 (左上圖)：Epoch 900 時，訓練誤差達到了目標，即學習的目標值。訓練誤差在不同 epoch 逐漸減小，最終為 0.010379。在測試數據上的平方和誤差為 0.069132，表示模型預測值與實際值之間差異非常小。

2. 第二次訓練 (右上圖)：Epoch 100 時，訓練誤差達到了目標。在測試數據上的平方和誤差為 0.033680，表示模型預測值與實際值之間差異非常小。
3. 第三次訓練 (左下圖)：Epoch 200 時，訓練誤差達到了目標。在測試數據上的平方和誤差為 0.007505，表示模型預測值與實際值之間差異非常小。
4. 第四次訓練 (右下圖)：Epoch 100 時，訓練誤差達到了目標。在測試數據上的平方和誤差為 0.040974，表示模型預測值與實際值之間差異非常小。

這表示使用 LBFGS 訓練方法時，神經網絡在訓練數據上取得良好配適，同時在測試數據上也取得相對較低的預測誤差。這可以解釋模型在未見過的數據上具有相對較好的配適能力，從而對機器手臂的姿勢進行有效的預測。

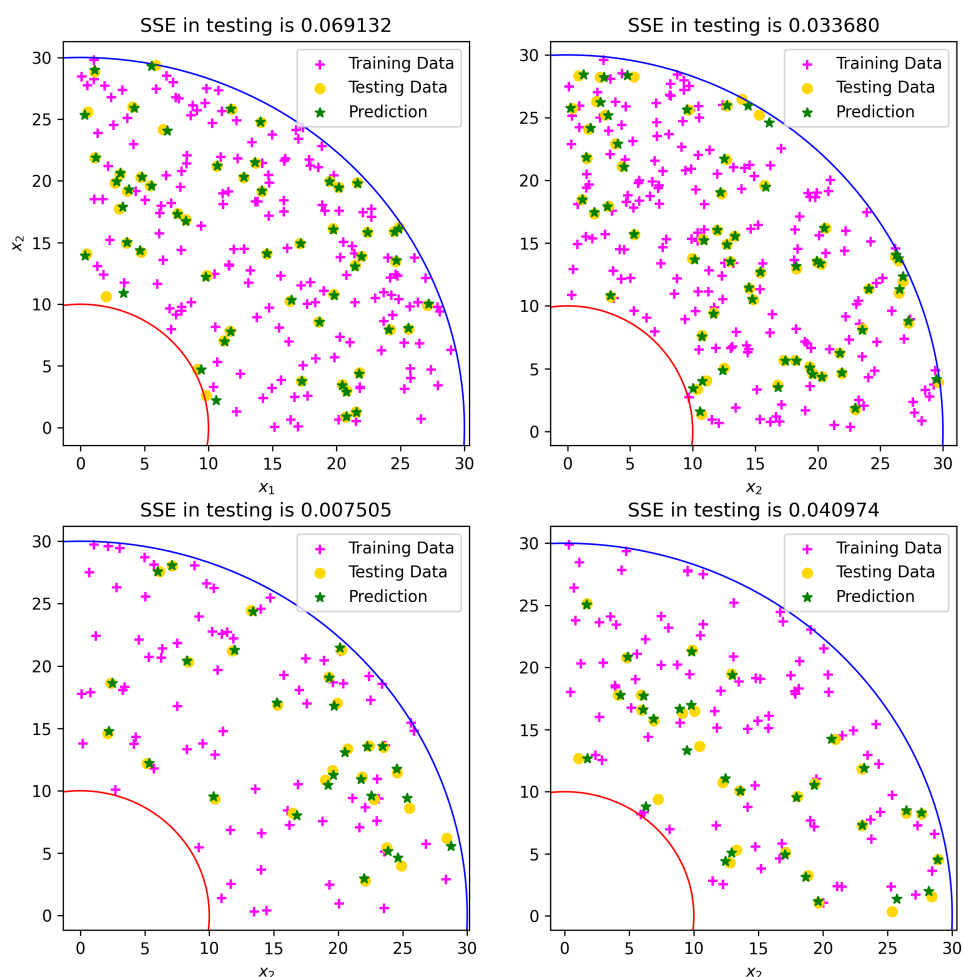


圖 8: LBFGS 演算法不同樣本神經元個數的比較

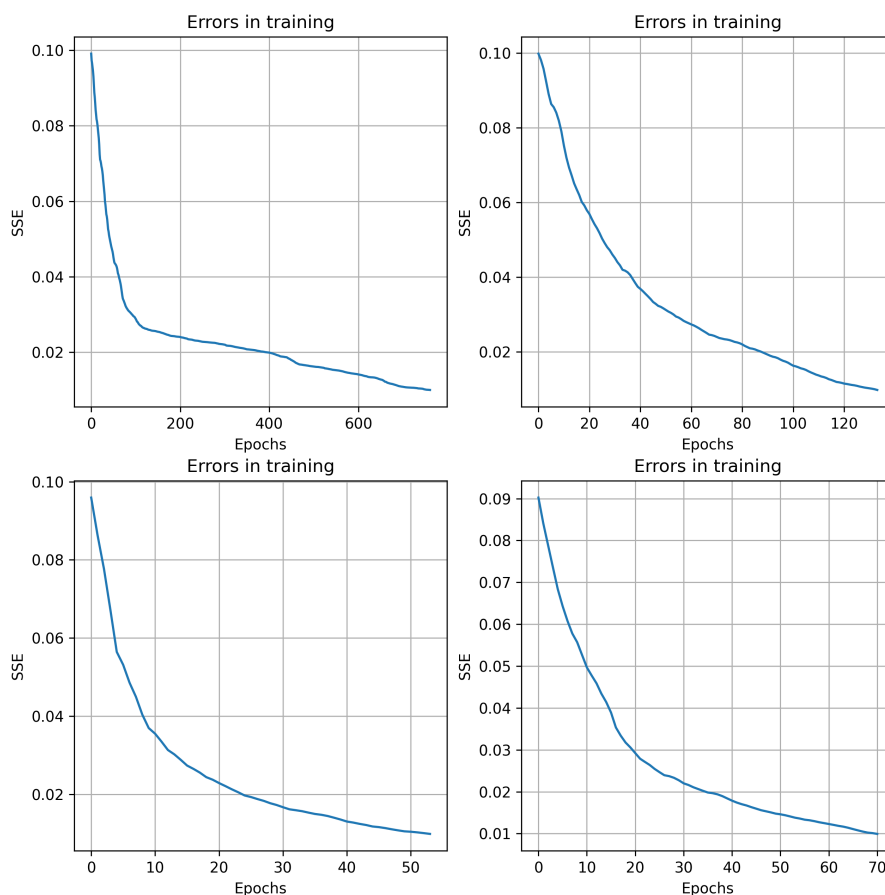


圖 9: LBFGS 演算法不同樣本神經元個數訓練誤差的比較

圖 10 探討在 CG 演算法下，不同樣本與不同神經元個數的比較，使用與圖 8 相同資料，CG 是共軛梯度 (Conjugate Gradient) 的縮寫。共軛梯度是一種迭代法，通常用於解決線性方程組或最小化函數的問題。它是一種常見的優化算法，特別適用於大型矩陣系統。先說明其設定配置，左上圖樣本數為 1000，神經元個數為 20，右上圖樣本數為 1000，神經元個數為 40，左下圖樣本數為 500，神經元個數為 20，右下圖樣本數為 500，神經元個數為 40。在每個子圖中，展示了訓練數據訓練集為紫色十字和測試集為金色圓點，以及在測試數據上的預測結果為綠色星號。圖 11 呈現其訓練過程誤差遞減的情形。總體來說，這表示使用 CG 訓練方法時，神經網絡在訓練數據上取得了良好的配適效果，同時在測試數據上也取得了相對較低的預測誤差。這可以解釋為模型在未見過的數據上具有相對較好的配適能力，從而對機器手臂的姿勢進行了有效的預測。

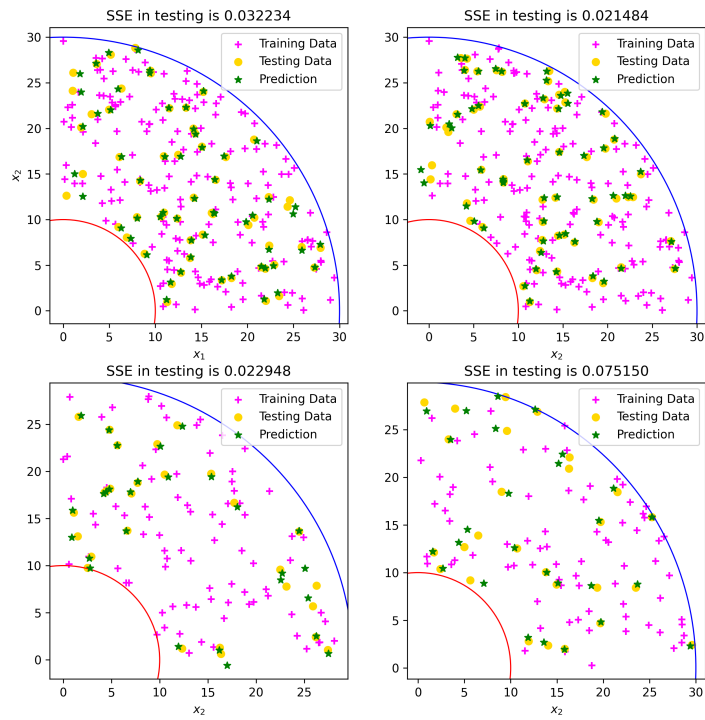


圖 10: CG 演算法不同樣本神經元個數的比較

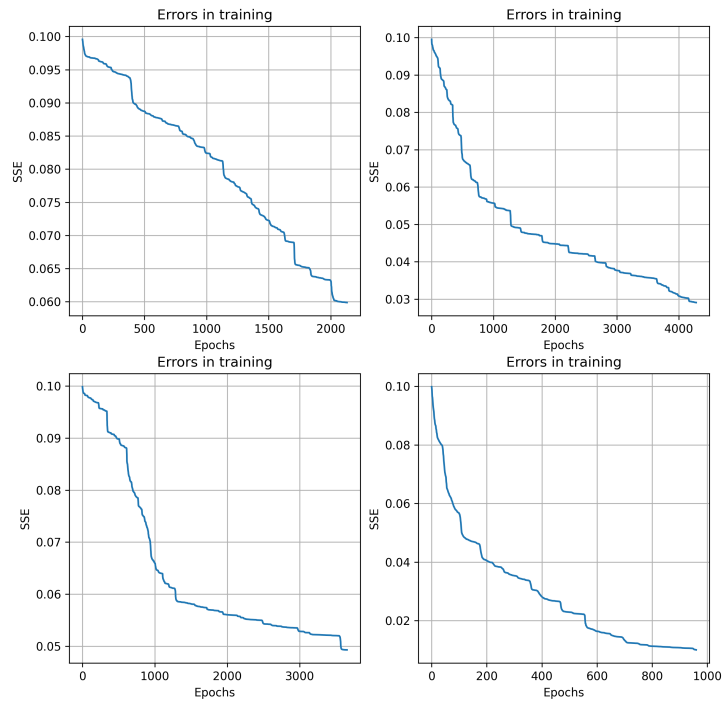


圖 11: CG 演算法不同樣本神經元個數訓練誤差的比較

3 圖形辨識

類神經網路也普遍被應用在圖形識別上 (Pattern Recognition)，一般被歸類為群組判別，接著我們使用 MNIST 資料集，訓練資料有 60,000 筆，每一筆都代表一個 28x28 像素的數字影像，首先查看訓練數據集中的第一張圖片及其對應的結果，第一個圖形是數字 5，圖 12 顯示模型預測的結果，表示模型在訓練數據上取得了良好的表現。

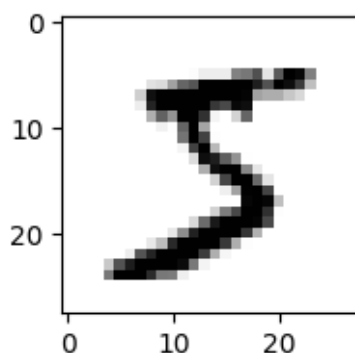


圖 12: 單一手寫數字的圖形辨識

圖 13 輸出判別結果皆在其圖形正上方，可以發現到辨識前十個圖形每個也都有正確辨識。

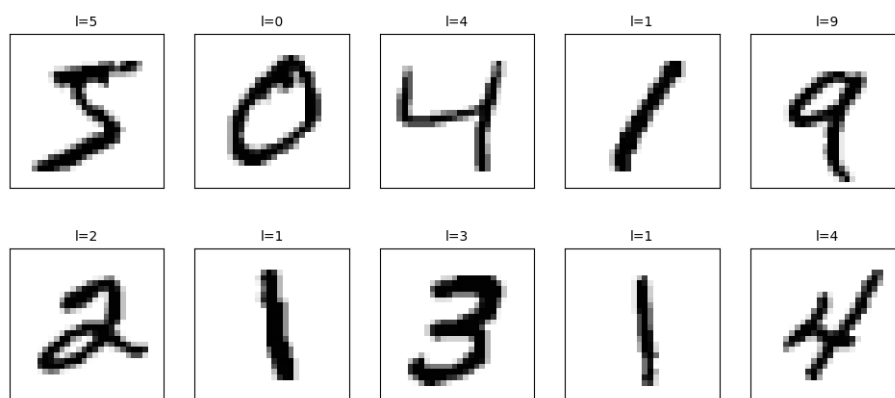


圖 13: 十個手寫數字的圖形辨識

圖 14 為混淆矩陣，展示預測的個數與真實的個數配對組合，對角線上的每個元素，表示模型正確預測的數量。而圖 15 就是透過正規化 60,000 筆資料後顯示其正確預測的準確率，在手寫數字上準確率都相當的高，皆超過 95%。而此混淆矩陣是一個 10x10 的矩陣，因為有 10 個數字類別（0 到 9）。每一行代表實際類別，每一列代表模型預測的類別。

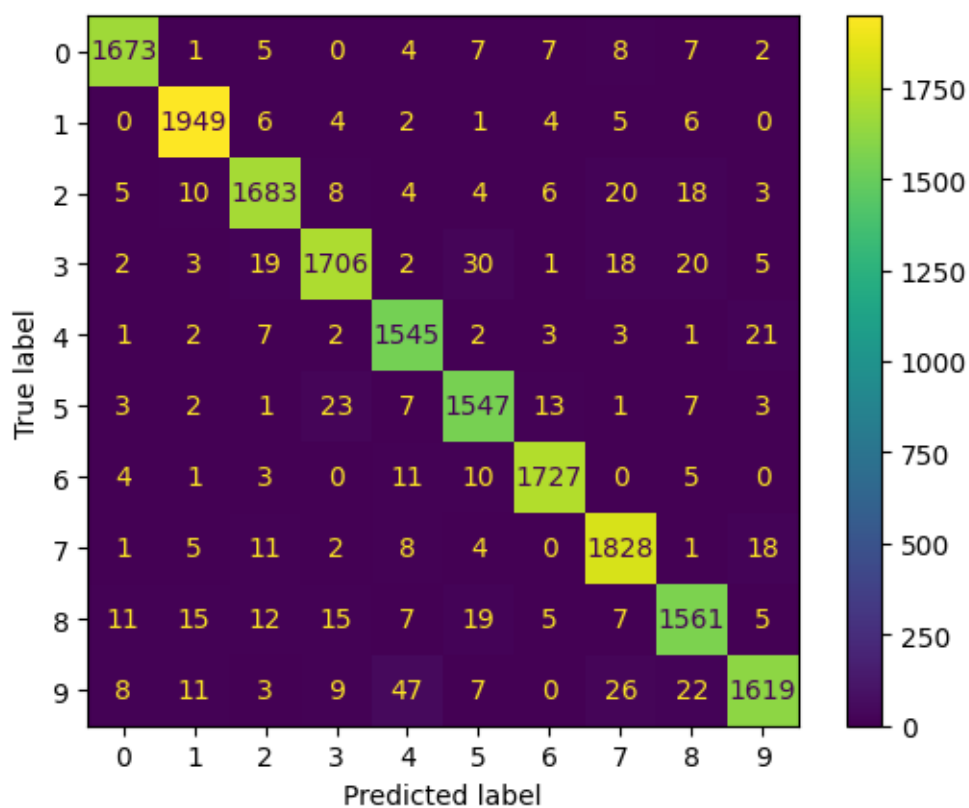


圖 14: 手寫數字辨識總數

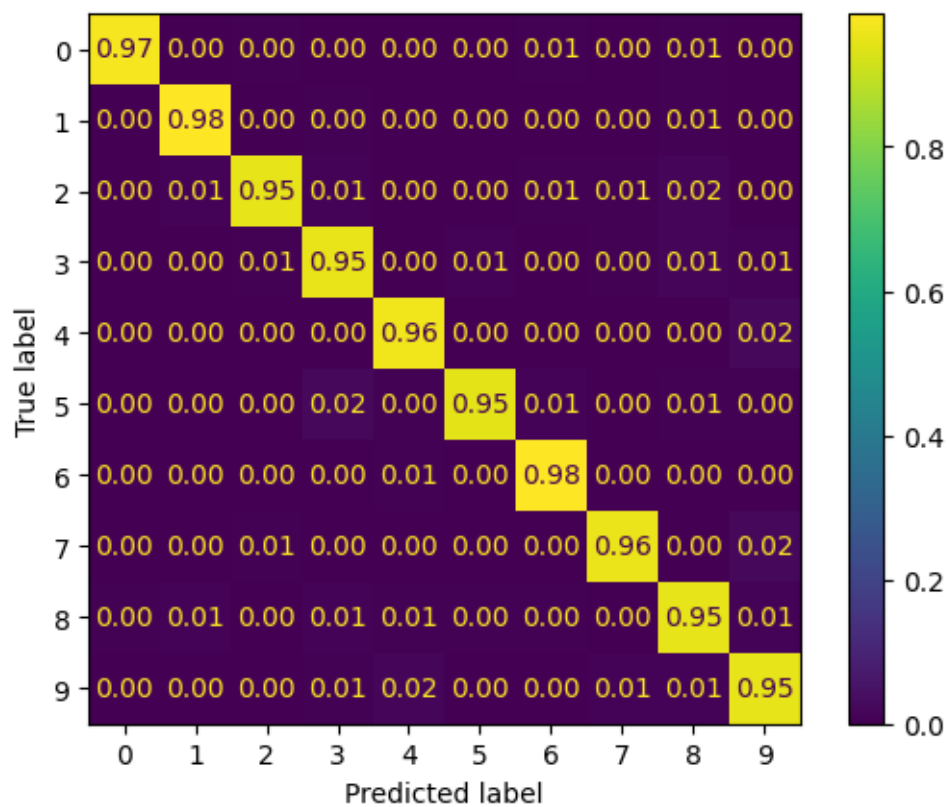


圖 15: 手寫數字辨識準確率

接著看較複雜的手寫字母，可預期的是辨別成果相較於手寫數字會較差，圖 16 先展示待會欲實驗的手寫字母資料，此資料是來自 MNIST 資料集。

The Montage of Handwritten Digits

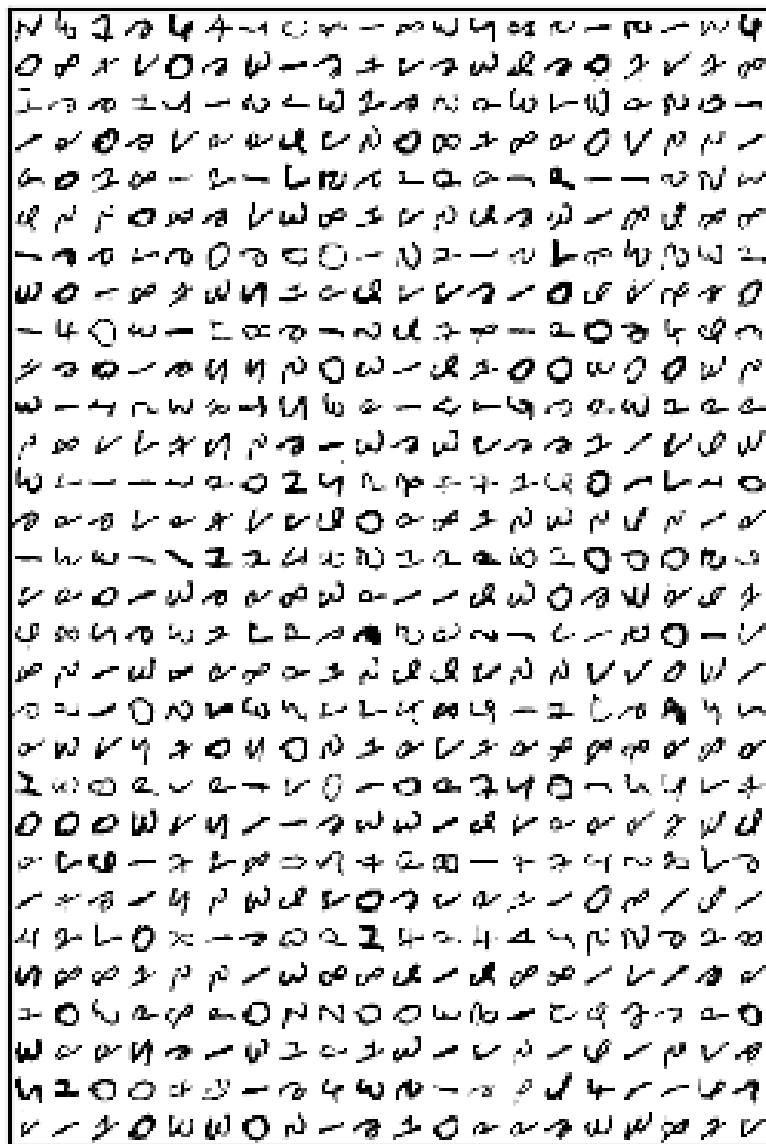


圖 16: MNIST 中 600 筆手寫字母

圖 17 為混淆矩陣，展示預測的個數與真實的個數配對組合，而圖 18 就是透過正規化後顯示其正確預測的準確率，在手寫字母上準確率的確如預想的一樣，成效沒有比手寫數字來的好，在某些字母上的辨識程度也相當差，但當然還是有表現好的時候。

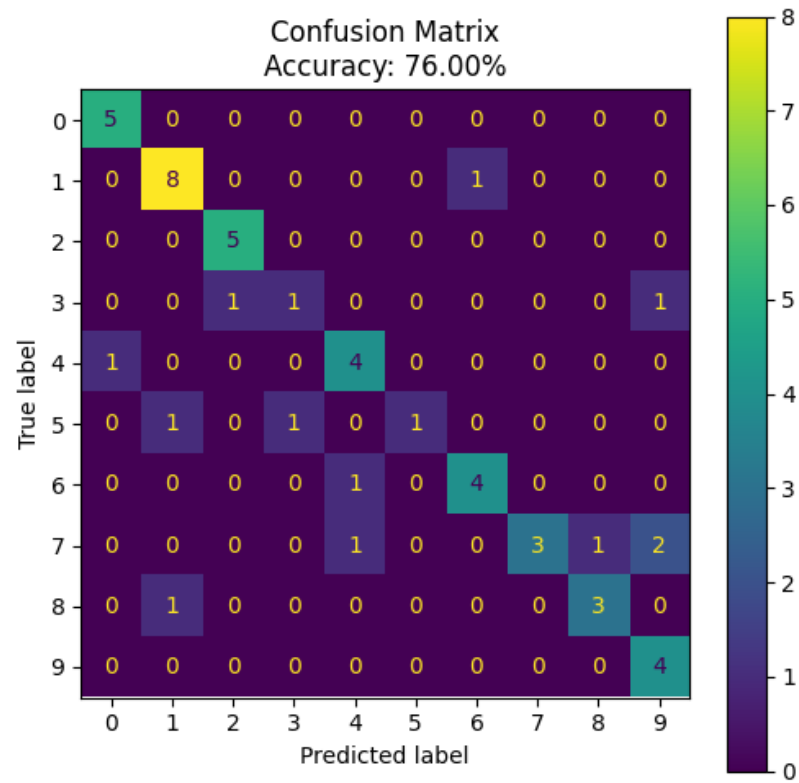


圖 17: 手寫字母辨識總數

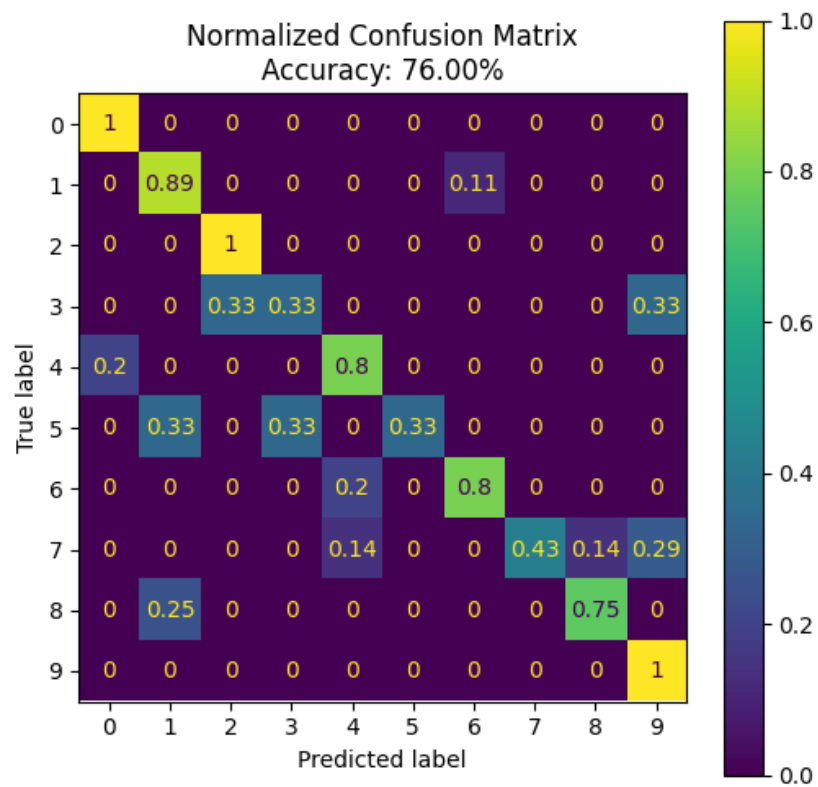


圖 18: 手寫字母辨識準確率

圖 19 是其損失函數，可以藉由函數值遞減的速度，觀察學習的過程順利與否。

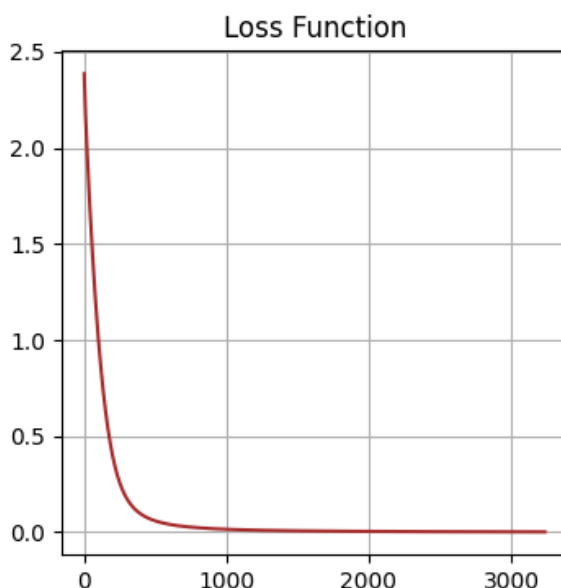


圖 19: 手寫字母辨識損失函數

4 結語

在本文中，我們深入探討了類神經網路 (ANN) 在機械手臂和圖形辨識領域的引人注目應用。通過模擬人腦神經元的運作方式，ANN 技術使得機器能夠學習、理解和做出智能決策，為機械手臂的靈活性和智能化帶來了巨大的提升。同時，圖形辨識技術賦予機器理解視覺信息的能力，為自動化系統的發展帶來了新的契機。

然而，我們也意識到類神經網路在實際應用中面臨著一些挑戰。例如，在訓練過程中，隱藏層數、神經元數量和訓練程度等因素對模型的表現產生著重要影響。此外，雖然類神經網路能夠完美地擬合輸入與輸出之間的關係，但在面對實際的測試資料時，其表現可能不如預期，這使得類神經網路的實際應用價值受到一些質疑。

然而，隨著技術的不斷進步和發展，我們對類神經網路的應用充滿信心。未來，我們期待通過更多的研究和實踐，進一步克服類神經網路在應用中遇到的挑戰，實現更廣泛、更深入的應用。類神經網路的發展將為機器智能和自動化技術帶來更多可能性，並為人類社會的進步做出更大的貢獻。讓我們期待著類神經網路技術在未來的發展中展現出更加璀璨的光芒。