

運用動態模型探討卷積神經網路辨識 CT 影像內肺部結節之超參數設定研究

陳祥輝、黃怡婷*

國立臺北大學統計學系

滕涵菁

怡發科技股份有限公司

April 20, 2020

Abstract

肺癌在台灣與工業化國家都是癌症死亡率排名第一位，胸部 X 光檢查無法滿足肺癌篩檢要求，除了肺部結節可能會被胸骨或其他器官遮蓋之外，X 光檢查時僅能提供二維 (two-dimension) 資訊，對於肺部異常結節 (lung nodule) 篩檢的敏感度不足。近年來發展 3D 立體影像技術，運用斷層掃描將切片 (slice) 影像組合成為一個三維 (3D) 立體影像圖形，再經旋轉進一步檢視不同角度，可得到較完整的影像資訊，藉以提升篩檢的敏感度。

本研究使用肺圖像數據庫聯盟圖像採集 (The Lung Image Database Consortium image collection; 簡稱 LIDC-IDRI) 置放在『The Cancer Image Archive (TCIA) Public Access』網站所提供電腦斷層掃描辨識肺部結節的國際標準影像格式資料 (Digital Imaging and Communications in Medicine; 簡稱: DICOM)，該資料包括肺癌診斷和胸部斷層攝影掃描 (Computed Tomography; 簡稱 CT) 篩檢，及標示病灶的註釋。本研究透過前置處理 DICOM 影像，建構卷積神經網路 (convolutional neural network, 簡稱 CNN)，瞭解 CNN 參數設定，進而對肺部結節的進行辨識。由於參數非常多，本論文提出利用動態建模方式來設定這些參數值，最後本研究有找出使用 CNN 來進行肺部結節辨識的最佳參數值組合。

關鍵字：人工智慧、卷積神經網路、深度學習、動態建模、肺部結核

JEL 分類：C6

1 前言

肺癌是全球癌症死亡的主要原因之一，在美國與臺灣地區，分別約有 6 成與 8 成的肺癌患者在臨床確診時是晚期或末期，治療難度高，五年存活率非常低 [?, ?, ?]。肺癌篩檢一般會使用胸部 X 光 (Chest Radiograph)，由於 X 光檢查是二維投影式影像 (2D projection images)，小於一公分的結節 (nodule) 或者結節的位置與其他的組織重疊時不容易被發現，導致辨識困難 [?]

*Corresponding author. Email: hwangyt@gm.ntpu.edu.tw

為瞭解低劑量電腦斷層 (Low-Dose Computed Tomography; 簡稱 LDCT) 篩檢肺癌表現, 美國國家癌症中心 (National Cancer Institute; 簡稱 NCI) 委託執行一項大型國家型肺癌隨機篩檢試驗 (The National Lung Screening Trial; 簡稱 NLST) [?], 該試驗收錄超過 5 萬個肺癌高風險個案, 將高風險個案隨機分組, 讓一半使用低劑量電腦斷層 (Low-Dose Computed Tomography; 簡稱 LDCT) 掃描, 而另一半採用胸部 X 光檢查, 研究發現使用低劑量電腦斷層掃描的敏感度與特異度分別為 93.8% 與 73.4%, 而 X 光組呈現敏感度 73.5% 與特異度 91.3%。該研究雖發現 LDCT 掃描的敏感度超過 9 成, 但 [?] 指出 [?] 邀請經驗豐富的臨床專科醫生來判讀所有影像, 而一般區域醫院的專科醫生接受的 LDCT 判讀訓練非常有限, 可能無法正確判讀影像。再者, 採用非小細胞肺癌 (non-small cell lung cancer), 共病 (comorbidity), CT 與存活等關鍵字, 搜尋 PubMed 的文獻, [?] 歸納出年齡介於 55 到 80 歲, 抽菸史大於 30 包菸-年 (pack-year), 或過去持續抽菸超過 15 年者, 建議應定期以 LDCT 進行肺癌篩檢, 不過, [?] 也提到 LDCT 有極高的偽陽率 (false positive rate) 和潛在過度診斷 (overdiagnosis) 的風險。[?] 比較三位專科醫生判讀 52 位肺部至少一處的病灶 (lesion) 的 LDCT 影像的受試者, 研究發現三位判讀者的平均 kappa 值為 0.48, 一致性尚可, 但三位專科醫生確有意見不一致的情形, 尤其針對介於中間的等級 (mixed solid and ground-glass attenuation lesion)。

隨著科技進步, 電腦斷層掃描設備與影像處理工具性能提升很多, 電腦斷層掃描主要透過斷層掃描得到許多切片 (slice), 再將這些切片組合成 3D 立體影像, 這類 3D 影像還可透過旋轉來檢視不同角度的影像, 包括正面、側面及上側三個維度。[?] 指出操作 LDCT 設備與判讀結果仍舊需要高度倚賴專科醫師, 但若肺部結節的大小介於 8-10 公釐, 則專科醫師正確判讀的比例會下降。若是可以運用深度學習演算法, 對影像分析與進行初步篩選, 提供專科醫生系統性與完整的影像分析資訊參考, 提昇臨床診斷結果的準確度與效率, 增加早期發現與早期治療的機會, 延長存活率。

影像辨識一般分為兩個階段來處理, 第一階段稱為前置處理或是預處理, 第二階段稱之為建模與測試, 其目的在於建置訓練模型、並會進行模型評估及預測。前置處理的目的是希望有效降低後續建模的錯誤判斷率, 包含清除雜訊、資料切割、標準化...等等, 進行, [?] 透過影像切割技術, 將整個肺部分割出來, 有效排除掉肺部以外的雜訊, 並且從分割出來的圖像計算圓形度、密度、橢圓度和偏心率等特定特徵, [?] 提出了『Black circular neighborhood』演算法進行肺部影線的邊緣偵測有效淨化資料, [?] 採用『分水嶺』(watershed) 演算法將肺部分割, 其概念源自於山與水不相融, 所以山與水之間就會形成一個區隔, 此方法可以讓肺部更精準從影像區隔, 而 [?] 透過門檻技術 (thresholding technique) 以及形態學操作來消彌掉背景以及周圍器官組織, 再計算出可疑且感興趣區域 (Region of Interest; 簡稱 ROI)。第二階段的模型建置方法包含多層感知器 (Multilayer Perceptron; 簡稱 MLP)、K-最近鄰演算法 (K Nearest Neighbor; 簡稱 KNN) 與支持向量機 (Support Vector Machine; 簡稱 SVM) 等人工智慧演算法來進行分類決策。[?] 採用支持向量機演算法進行分類辨識, 該論文集成裝袋算法 (bagging), 自我調整增強算法 (Adaptive Boosting; 簡稱 adaboost), 隨機子空間劃分算法 (random subspace) 等三種分類器, 運用 ROC 曲線下面積發現三種方法的準確性差異不大。[?] 提出利用基因演算法 (Genetic Algorithm; 簡稱 GA) 來進行分類。

腦斷層掃描所得到的 3D 影像的每張切片都是利用國際標準影像格式 DICOM (Digital Imaging and Communications in Medicine; 簡稱 DICOM) 儲存。本研究會從『The Cancer

Image Archive (TCIA)Public Access』網站 [?] 下載最原始 DICOM 格式影像資料檔案，從最原始肺部結節 CT 影像檔案來了解卷積神經網路演算法資料預處理設定程序，包括開啓 DICOM 檔案、資料萃取 (extracting)、探索 (exploration)、篩選 (filtering)、抽樣 (sampling)、分類標籤 (labelling)、檔案轉換 (transforming)、亨斯菲爾德單位轉換、標準化以及調整影像大小。第二階段採用深度學習 (Deep Learning, 簡稱: DL) 的卷積神經網路 (Convolutional Neural Network; 簡稱 CNN) 演算法建構模型、調整參數、訓練模型、評估模型。本研究會交代預處理步驟相關參數的設定方式與流程，但會著重於第二階段的 CNN 模型建構與參數的選擇，研究章節規劃如下，第二章探討 CNN 演算法的基本架構、處理流程、數學原理與評估指標。第三章將介紹本研究整個前置作業過程，也就是第一階段。第四章探究本研究建構的『卷積神經網路』動態模型，第一節會描述本研究的模型結構、參數的起始設定，以及利用不同的情境來探討單一個參數的選擇。第二節的重點將會綜合參數來進行隨機抽樣與最佳化後的固定參數的個別探討。第三節將會針對優化後的結果進行比較與說明，最後選擇出最佳的組合。第四章主要說明本研究運用動態卷積神經網路架構的方法來大到模型穩定性，第五章為本研究的結論以及在研究過程當中所遇到的問題、未解決的問題，或是未來可再深入探討的研究問題提出說明。

2 研究方法

CNN 為深度學習主要使用的演算法，該方法是由人工神經網路 (artificial neural network) 延伸發展出來。早期的神經網路主要有前饋式神經網路 (feedforward neural network) 與遞歸神經網路 (Recurrent Neural Networks; 簡稱: RNN) 兩種，前者會有特定的輸入結構，而後者會在自身網絡中循環傳遞，並廣泛運用於有時間結構資料。由於本研究探討圖像辨識，並沒有涵蓋時間結構，且 CNN 是由前饋式神經網路延伸而來，以下僅討論前饋式神經網路與 CNN 的基本架構，處理流程，數學原理，多元分類函數，最後，介紹常見人工智慧或深度學習的框架，並說明本研究所採用的框架與評估指標。

2.1 前饋式神經網路

前饋式神經網路是最早發明也是最簡單的人工神經網路，以階層方式建構，屬於階層式網路，其階層 (layer) 有三種型態，分為輸入層 (input layer)、隱藏層 (hidden layer) 以及輸出層 (output layer)，其中隱藏層可設定為多層。所謂前饋式意思參數從輸入層透過隱藏層向輸出層單向傳播，並運用非線性函數將不同層間的神經元相連，但不同層的神經元彼此不相連，亦即每一層皆由一些神經元 (neuron) 構成，同一層中的神經元彼此不相連，但不同層間的神經元則彼此相連，訊息傳輸方向是單向，此種連接方式稱為多層感知器 (Multilayer Perceptrons, 簡稱: MLP)。最後，多層感知器採用反向傳播 (back-propagation) 的監督式學習來訓練資料，得到預測分類結果。

2.2 CNN

CNN 是一種特徵萃取的演算法 [?, ?]，從前饋式神經網路延伸而來，常被用在圖像辨識與語音辨識。與前饋式神經網路一樣，CNN 採用多層感知器來串連神經元，但加入有更多不同型態的層，除了輸入層，隱藏層與輸出層，CNN 在輸入層與隱藏層中再加入卷積層 (convolution layer)，池

化層 (pooling layer) 與全連接層，總共有六種型態。再者，CNN 也是採用反向傳播的監督式學習來訓練資料，下一章會說明訓練方式。以下先說明各層資料的操作方式與參數介紹：

2.2.1 輸入層

輸入層通常是影像資料，而 2D 影像分灰階與彩色兩種。若影像大小為 $n \times m$ ，通常資料三維度儲存，以 $[n, m, d]$ 表示，前面兩個值代表影像大小，而第三個值則代表灰階或彩色，當 $d = 1$ 代表灰階影像，而 $d = 3$ 則彩色影像，其中 3 表示三個原色 R 、 G 與 B 。彩色影像也可轉換成灰階影像，以下為常用的三種方式：

$$\text{亮度 (brightness)} = \frac{1}{2} \times (\max(R, G, B) + \min(R, G, B))$$

$$\text{光度 (lightness)} = 0.21 \times R + 0.72 \times G + 0.07 \times B$$

$$\text{平均亮度 (average brightness)} = \frac{1}{3} \times (R + G + B)$$

圖 ?? 提供一個 7×7 的影像與其應的數據資料，後續 CNN 其他 4 層的處理流程都會採用該資料來說明。。一般慣例黑色會用 1 代表，而白色會以 0 代表，但為突顯該影像，以下說明會刻意將黑白反轉，後續圖像皆會採用反轉方式呈現。

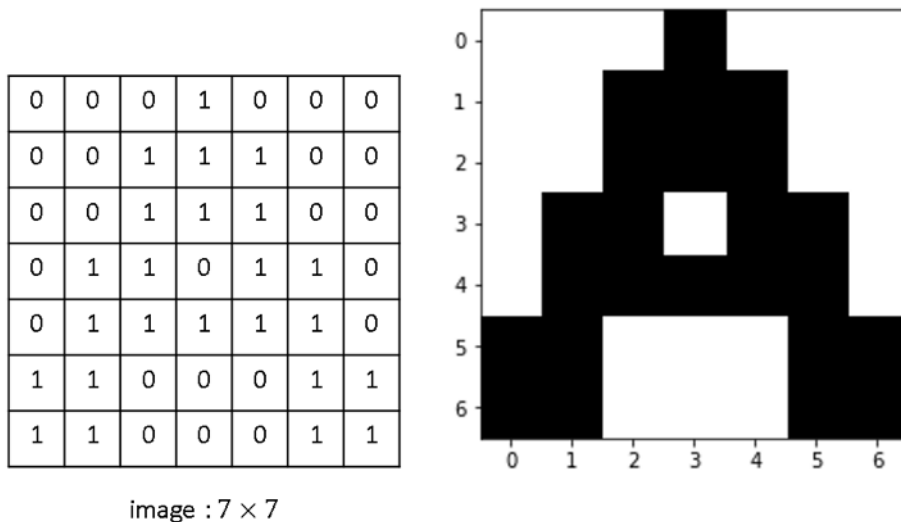


圖 1: 7×7 黑白反轉呈現影像陣列數據與原圖

2.2.2 卷積層

輸入層經由過濾器 (filter) 可以產生影像的特徵值，卷積層記錄這些影像的特徵值 (feature)。產生影像特徵值需設定過濾器與移動步伐 (stride)，而過濾器需設定包含核心大小 (kernel size) 與權重 (weight)，核心大小的決定方式可以隨機產生或是自行設定，而權重一開始是隨機產生，再取出與過濾器大小一樣的影像，運用 Frobenious 內積 (inner produce) 將相同位置的兩個數字相乘，最後計算總和，即為一個資料，再經由設定的移動步伐 (stride)，由列與再行逐一向右向下滑動產生卷積層特徵值的資料，而移動的大小可以自行設定或隨機產生，該輸出就是下一個『池化層』的輸入。

考慮一個 3×3 的過濾器定義於表 ??，並設定移動步伐為 1×1 ，即每次會往右或往下移動 1 步，依照設定的步伐，由列與再行逐一向右上移動，圖 ?? 呈現圖 ?? 資料的轉換方式，轉換後第一列與第一行的資料為 $0 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 1 \times 0 + 0 \times 1 + 0 \times 1 + 1 \times 1 = 1$ 。陸續往右移動 1 個單位，分別計算出 1、2、3、2、1，然後，在往下一列移動，從左至右重新再運算一次，如圖 ??。

表 1: 3×3 濾鏡範例說明

0	0	0
0	0	0
1	1	1

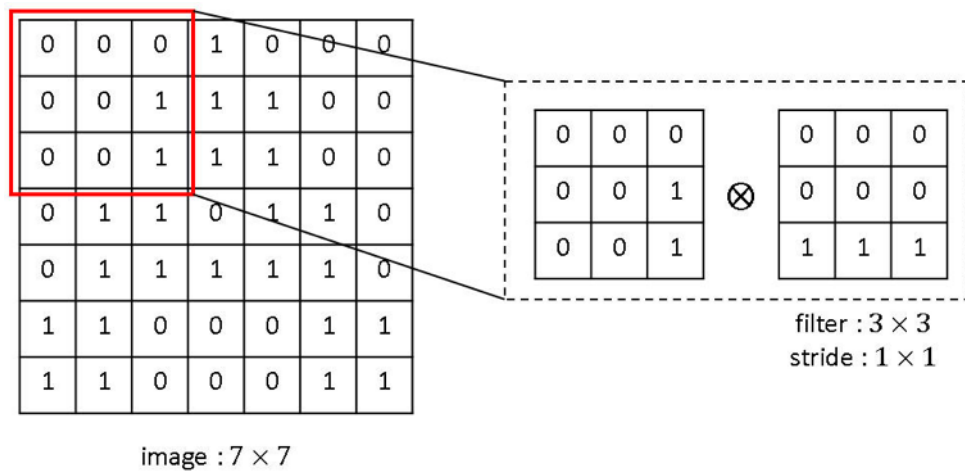


圖 2: 7×7 影像運用 3×3 過濾器與步伐 1×1 滑動示意圖

原始影像經過濾器處理之後，輸出的影像大小最後會與原始影像的大小不同，假設影像大小為 $N \times N$ ，而過濾器的大小為 $F \times F$ ，且步伐設為 $s \times s$ 情況，則輸出的影像大小將會是 $\lceil (N - F + 1)/s \rceil \times \lceil (N - F + 1)/s \rceil$ ，其中 $\lceil x \rceil$ 表示不小於 x 的最小整數。經由表 ?? 的過濾器，圖 ?? 為圖 ?? 最後運算的結果，影像資料會從一個 7×7 的陣列變成一個 5×5 的矩陣，比原本的影像小，從呈現的尺規來看， X 軸與 Y 軸的最大值都從 6 變成 4。

由於影像經過濾器處理會比原來輸入的影像小，為維持輸出與輸入影像大小一致，一般調整會將資料外圍補 0 (zero-padding)，讓原圖變大，如圖 ?? 使得經過濾器運輸出的大小仍會與原圖一樣，而外圍要補幾行與幾列的 0 與過濾器有關，若過濾器的大小為 $F \times F$ ，則要補 0 的行列數為 $(F - 1) \times (F - 1)$ 。例如原本影像大小為 7×7 ，輸出影像會的大小會是 $(7 - 3 + 1) \times (7 - 3 + 1) = 5 \times 5$ ，若是想要讓輸出的影像大小維持不變，則所需補 0 的行列數為 $(3 - 1) \times (3 - 1) = 2 \times 2$ ，也就是左右各補兩行 0，上下都要各補兩列 0，圖 ?? 呈現 7×7 範例補 0 後的卷積層與對應的圖像，而最右邊則是沒有補 0 的圖像。

由於卷積層是經由過濾器擷取特徵，不同的過濾器會提供不同的特徵，因此卷積層會設定多個過濾器，再運用上述處理方式產生新影像，例如隨機產生 16 個不同的過濾器，則一個影像就會產生 16 個輸出資料，如圖 ??，而這 16 個輸出一般就稱之為特徵圖 (feature maps)。

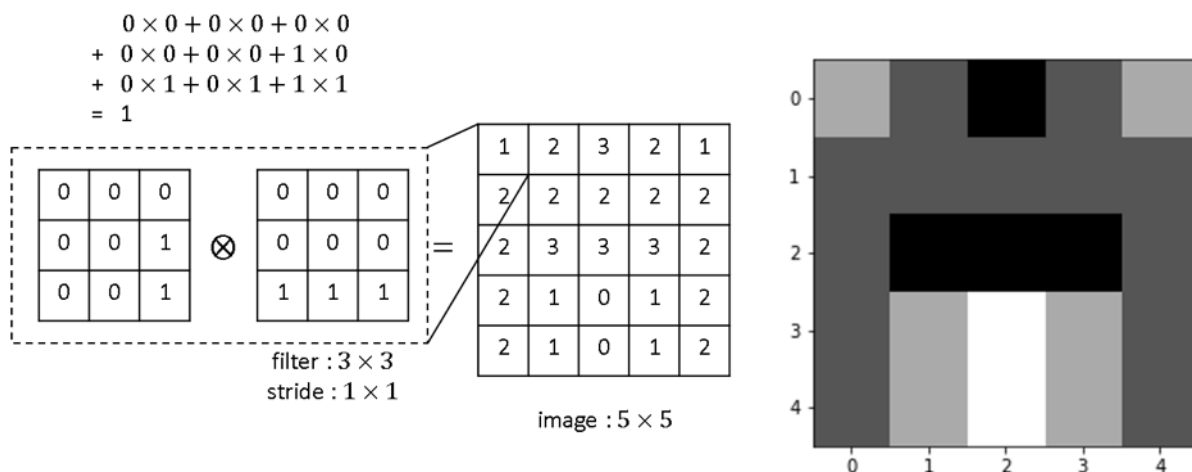


圖 3: 7×7 影像經 3×3 過濾器之權重運算與輸出示意圖

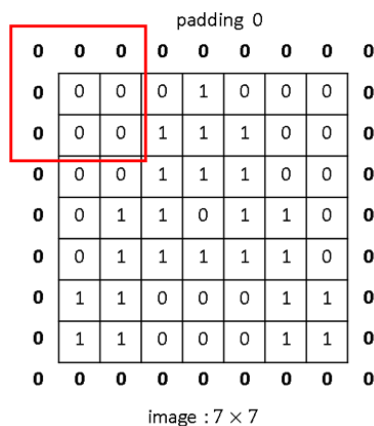


圖 4: 影像補 0 的方式

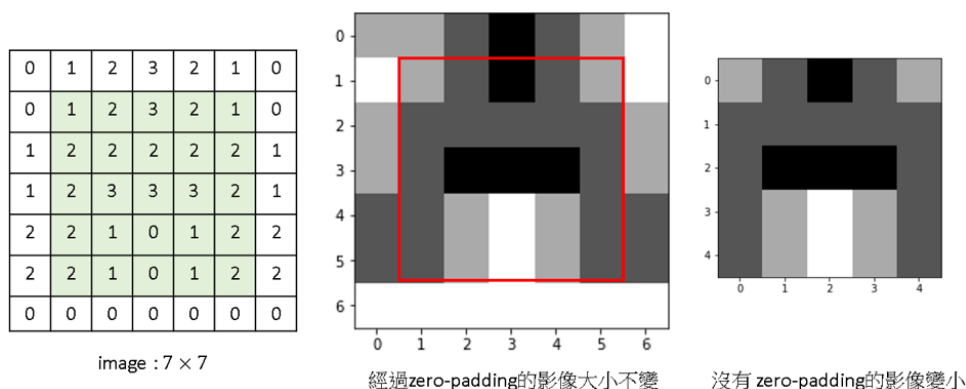


圖 5: 影像補 0 與沒有補 0 經『卷積層』作用後的影像示意圖

2.2.3 池化層

卷積層提供資料的特徵圖，但資料量過多，為提升電腦運算效率，池化層透過過濾器進行降維，萃取主要特徵值。該層僅需要設定過濾器大小，後續稱 Pool_size，而過濾器的權重會依照實際需求

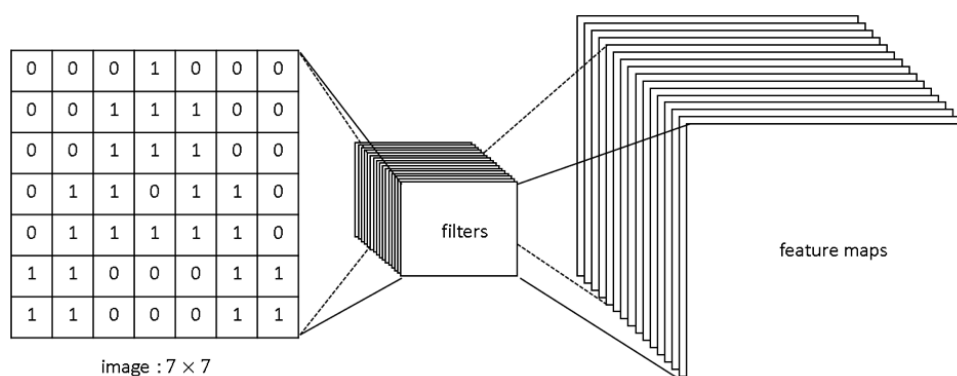


圖 6: 多個過濾器與對應輸出特徵圖

來設定，例如取平均值來平滑該區的資訊，或是取最大值來突顯出該區的資訊，一般常用使用取最大值的方式。而移動處理方式與卷積層不同，池化層擷取資料不重複，移動步伐與過濾器大小一樣，設定所需過濾器後，運用 Frobenious 內積將相通位置的兩個數字相乘，取出最大值，由列再行滑動產生池化層的資料。與卷積層一樣，輸出的影像大小與過濾器有關，設影像大小為 $N \times N$ ，過濾器的尺寸為 $F \times F$ ，則輸出的影像大小將會是 $\lceil (N - F + 1)/F \rceil \times \lceil (N - F + 1)/F \rceil$ 。

以圖 ?? 為例，設定過濾器設定為 2×2 ，且篩選方式為取最大值，操作設定稱為 Max Pooling，步伐為 2×2 ，第一步先取左上角 2×2 紅色方框資料，找出最大值為 2，即為第一個輸出層的元素，第二步往右移 2 個單位，找出最大值為 3，依序設定，圖 ?? 呈現最後池化層的輸出資料與對應圖像。

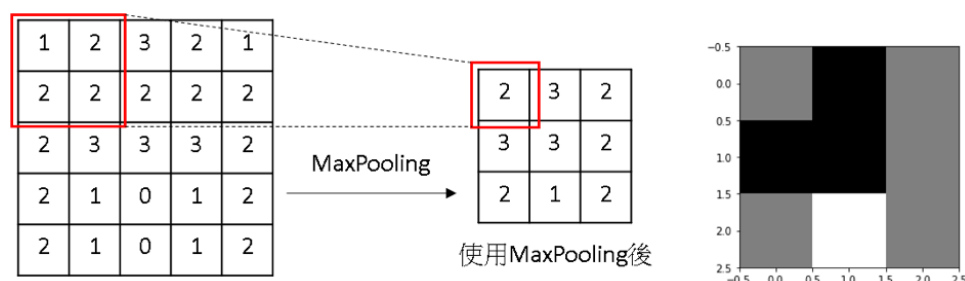


圖 7: 圖 ?? 的資料池化過程與結果影像示意圖

2.2.4 全連接層

為方便後續電腦運算，需將矩陣形式的資料轉化成向量形式，此為全連接層的輸出，轉化的方式是從第一列開始將資料平坦化 (flattening)，即轉為行向量，在轉換第二列，依序將所有列向量，轉成一個行向量。以圖 ?? 的資料來說明，圖 ?? 列出展開後的結果。

2.2.5 隱藏層

卷積層，池化層與全連接層都是 CNN 額外加入的資料處理步驟，從這些層會建構出許多神經元，即為一般神經網絡輸入層，接下來與一般神經網絡演算法一樣，CNN 會執行隱藏層，此部份也

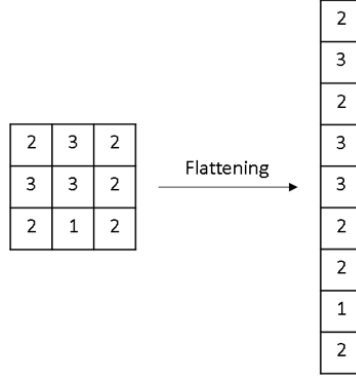


圖 8: 輸入圖 ?? 的資料展開成全連接層的輸出結果示意圖

是採前饋式神經網路設定，再運用反向傳播監督是學習來訓練資料，最後再經由 softmax 激活函數輸出預測類別。以下說明神經網絡神經元連接的理論設定。

令 x_1, x_2, \dots, x_n 表示由全連接層產生的資料， w_{ij}^k 代表第 k 層，第 i 個輸入，第 j 個神經元的權重，而 b_j^k 代表第 k 層，第 j 個神經元的偏差值，並以 L_j^i 代表第 i 個輸入，第 j 個神經元的輸出值， $i = 1, \dots, k, j = 1, \dots, p$ ，最後，令 $\sigma(\cdot)$ 表示激活函數，則從輸入的神經元至第一個隱藏層神經元的輸出值的數學算式定義如下：

$$L_1^1 = \sigma(x_1 w_{11}^1 + x_2 w_{21}^1 + \dots + x_n w_{n1}^1 + b_1^1), \quad (1)$$

$$L_2^1 = \sigma(x_1 w_{12}^1 + x_2 w_{22}^1 + \dots + x_n w_{n2}^1 + b_2^1), \quad (2)$$

\vdots

$$L_p^1 = \sigma(x_1 w_{1p}^1 + x_2 w_{2p}^1 + \dots + x_n w_{np}^1 + b_p^1), \quad (3)$$

而第一個隱藏層神經元輸出值會變成第二個隱藏層神經元輸入值，將以上步驟以矩陣方式呈現

$$\mathbf{L}^1 = \sigma(\mathbf{X}\mathbf{W}^1 + \mathbf{b}^1)$$

其中 $\mathbf{X} = (\mathbf{x}', \dots, \mathbf{x}')^T$, $\mathbf{L}^1 = (L_1^1, L_2^1, \dots, L_p^1)^T$, $\mathbf{b}^1 = (b_1^1, b_2^1, \dots, b_p^1)^T$, $\mathbf{W}^1 = (\mathbf{w}_1^1, \dots, \mathbf{w}_p^1)^T$, $\mathbf{w}_1^1 = (w_{11}^1, \dots, w_{np}^1)^T$ 。

將 (??) 式-(??) 式中 x_1, x_2, \dots, x_n 替換為 $L_1^1, L_2^1, \dots, L_p^1$ ，而 $L_1^1, L_2^1, \dots, L_p^1$ 替換為 $L_1^2, L_2^2, \dots, L_p^2$ 即可求得第二個隱藏層神經元輸出值，以此類推 k 次，可得第 k 個隱藏層神經元輸出值，將以上步驟以矩陣方式呈現如下：

$$\mathbf{L}^2 = \sigma(\mathbf{L}^1\mathbf{W}^2 + \mathbf{b}^2)$$

\vdots

$$\mathbf{L}^k = \sigma(\mathbf{L}^{k-1}\mathbf{W}^k + \mathbf{b}^k),$$

其中 \mathbf{b}^t 與 \mathbf{W}^t 的大小會依照輸入與輸出的神經元調整。圖 ?? 為前饋式神經網路操作示意圖。

激活函數 $\sigma(\cdot)$ 是一個轉換函數，該函數設定的目的是讓強者越強，弱者越弱，也就是希望突顯重要特徵，消彌不重要的資料。以下介紹兩個常用的激活函數，Sigmoid 與 Relu (rectified

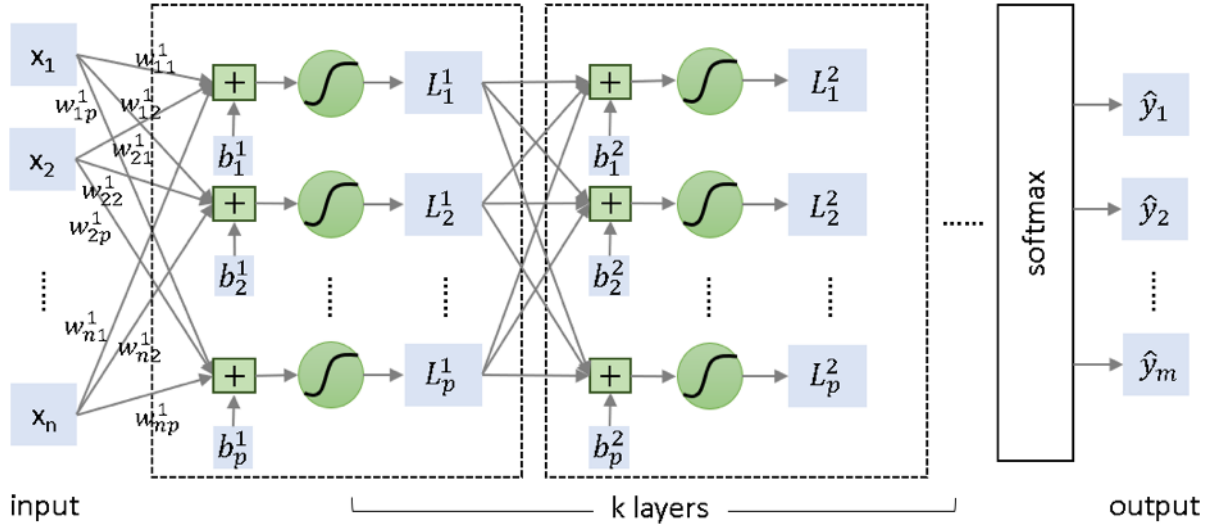


圖 9: 前饋式神經網路操作示意圖

linear unit) 激活函數。Sigmoid 激活函數定義為

$$f(x) = \frac{1}{1 + c \exp(-(a + bx))},$$

其中 a , b 與 c 為給定常數，而 $a = 0$, $b = 1$ 與 $c = 1$ 為最常用的設定。依給定常數不同，該函數會有不同的臨界定義域，但大約落在 $[-5, 5]$ ，而值域則與常數無關，一定落在 $[0, 1]$ 。若輸入資料低於這個臨界定義域則輸出就會趨向於 0，也就是消彌掉很不重要的資訊；而高於定義域，則輸出資料就會趨向 1，也就是突顯出很重要的資訊。依照 b 的符號，Sigmoid 激活函數會的值域會緩慢遞增或遞減。Relu 激活函數定義為

$$f(x) = \begin{cases} 0 & \text{若 } x < 0 \\ x & \text{若 } x \geq 0 \end{cases}$$

該函數定義域與值域均為 $[0, \infty)$ ，輸入資料低於 0，則輸出設為 0，也就是消彌掉很不重要的資訊，而輸出資料與輸入資料設為一樣，當輸入資料大於 0。

令 $\mathbf{L} = (L_1^k, L_2^k, \dots, L_p^k)^T$ 代表由隱藏層 k 輸出影像資料，而輸出的分類假設總共有 m 種分類，並以 y 表示觀察的分類資料，網絡最後會利用一個多元分類函數將全連接層的神經元輸出結果轉換成機率值，而預測分類依據機率值最大的準則來判別最終歸屬哪一類。常用的轉換函數稱為 softmax，該函數之定義與多元邏輯斯回歸模型 (multinomial regression) 的設定方式一樣，定義為：

$$P[y = j | \mathbf{L}] = \frac{\exp(\mathbf{L}^T \mathbf{w}_j^{k+1})}{\sum_{l=1}^m \exp(\mathbf{L}^T \mathbf{w}_l^{k+1})}$$

其中 \mathbf{L}^T 代表 \mathbf{L} 的轉置矩陣， \mathbf{w}_j^{k+1} 代表第 j 類的權重，而 $\mathbf{w}_j^{k+1} = (w_{j1}^{k+1}, \dots, w_{jp}^{k+1})'$ 。

最後經由預設的損失函數 (loss function) 來推估每次迭代的權重，令神經網路在經過前饋式傳遞運算之後的預測值 \hat{y} ，而原本觀察值為 y ，在機器學習的領域最常使用的損失函數為交叉熵 (Cross-Entropy)，其數學定義如下：

$$L(\hat{y}, y) = - \sum_{i=1}^n y_i \log \hat{y}_i$$

運用設定的損失函數，找出讓損失函數有最小值的解，即為該次運算權重 (或稱為過濾器) 及偏差值的解，因損失函數為非線性線函數，權重與偏差值僅能使用數值方式求得，針對每個維度偏微分，找出調整的方向以及學習率，求出第 $t + 1$ 次遞迴的權重，即

$$w_{ij}^{t+1} = w_{ij}^t - \eta \times \frac{\partial L}{\partial w_{ij}} \Big|_{w_{ij}=w_{ij}^t}.$$

2.2.6 輸出層

CNN 採用 OneHot 編碼方式設定輸出層資料，例如要進行手寫數字的辨識，也就是分類 0、1、2、...、8、9，共 10 個數字，輸出層必須為長度 10 的位元陣列。例如分類為 0，則其標籤值 (label) 應該就為 1000000000，也就是第一個位元 (bit) 為 1，其他皆為 0；分類為 3，則其標籤值 (label) 應該就為 0001000000，也就是第四個位元 (bit) 為 1，其他皆為 0，如圖 ?? 所示。若是分類的標籤值是文字型態，也必須要先轉成從 0 開始的數字 0、1、...，在實作上才能轉換成 OneHot 編碼。

0	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

圖 10: 輸出層數字 0-9 的 OneHot 編碼

基於上述說明，從 CNN 的資料處理流程為輸入層輸入影像、經過卷積層萃取出特徵圖 (feature maps)、再經過池化層縮減資料找出影像主要的特徵圖，與隱藏層一樣，卷積層與池化層可以重複執行多次，之後透過全連接層將所有資料展開成一個行向量，最後再由 softmax 激活函數進行非線性多元分類函數轉換來判斷資料所屬類別，其基本架構見圖 ??。

2.3 機器學習框架

近幾年來人工智慧的崛起，很多公司都推出自家的機器學習框架 (machine learning framework) [22]，包括 Apache Singa、Amazon Machine Learning、Azure ML Studio、Caffe、CNTK、H2O、Massive Online Analysis (MOA)、MLlib (Spark)、Scikit-Learn、TensorFlow、Theano、Torch、... 等等。

本研究第一階段前置處理與第二階段模型建構皆採用 Python 的程式語言，第二階段模型建構的底層框架採用 Google TensorFlow，並使用 Keras [?] 來測試資料，搭配後端 TensorFlow 的開發環境，Keras 為一種較高階 neural networks API，如圖 ?? 所示，其最底層皆可支援 CPU 或是 GPU 的運算。詳細說明見 [?]

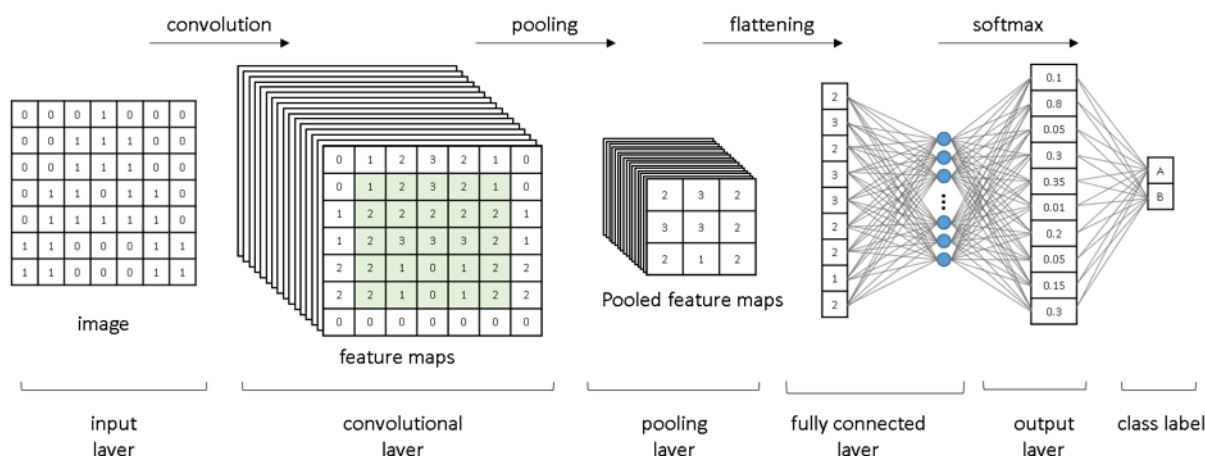


圖 11: 卷積神經網路的基本結構

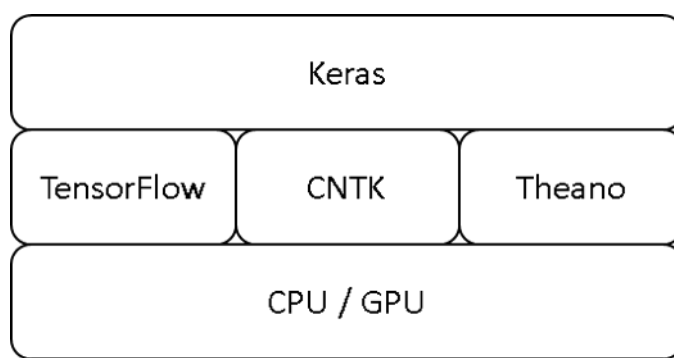


圖 12: Keras 與深度學習的框架

2.4 評估指標

本研究感興趣的反應變數為判定是否有肺部結節，運用檢驗工具會有兩種結果，判定陰性 (Negative) 反應代表正常，陽性 (Positive) 反應代表有肺部結節。將判定結果與真實觀測值比較，可得到以下四種情況：1. 真陽性 (True Positive; 簡稱 TP)，實際觀察為陽性，且正確 (True) 預測為陽性；2. 真陰性 (True Negative; 簡稱 TN)，實際觀察為陰性，且正確預測為陰性；3. 偽陽性 (False Positive; 簡稱 FP)，實際觀察為陰性，卻錯誤 (False) 預測為陽性；4. 偽陰性 (False Negative; 簡稱 FN)，實際觀察為陽性，卻錯誤預測為陰性。一般人工智慧用來評估模型預測正確率會將資料整理成表 ??，此表格稱為混淆矩陣 (confusion matrix)，橫列代表實際情況 (actual outcome)，縱列代表預測結果 (predicted outcome)。表格的左上與右下兩個區域代表正確的預測，也就是真陽性與真陰性；右上與左下兩個區域代表錯誤的預測，也就是偽陰性與偽陽性，其中 A, B, C, D 分別表示該種情況收集到的人，並配合準確率 (accuracy)，定義為

$$\text{準確率} = \frac{A + C}{A + B + C + D}, \quad (4)$$

來評估建立模型的適切性。

表 2: 混淆矩陣

真實狀態	預測狀態	
	陽性	陰性
陽性	<i>A</i>	<i>B</i>
陰性	<i>C</i>	<i>D</i>

2.5 動態建模

由於 CNN 執行過程需要設定許多參數，包含輸入影像的高度與寬度、卷積層核心大小與移動步伐 (strides) 與過濾器個數，池化層核心 (kernel size) 移動步伐 (strides)，另外，架構中額外可以設定卷積層與池化層資料丟棄的比率，與不同的激活函數等。當硬體與時間有限，逐一搜尋最佳參數組合變得不實際，本研究建議先給定每個參數一組參數值，例如卷積層需的過濾器的個數有 16, 32, 64, 128 等四種設定，核心大小與移動步伐都有 2, 4, 8, 16, 32, 64 等六種設定，而池化層設定 max pooling 的大小，有 2, 4, 8, 16 等四種設定，之後採用動態建模方式來選擇參數的值，也就是透過隨機重複抽樣方式來動態選定參數形成模型，再進入模型的訓練，藉由多次重複抽樣與訓練來判斷哪一組的參數值模型會是比较適合欲分析資料的最佳組合。

本研究主要說明動態參數設定的表現，輸入影像的高度與寬度會影響建模的時間，因此本研究設定輸入影像的高度與寬度均為 128，動態參數選擇將只針對以下項目：卷積層與池化層的層數、過濾器的個數、過濾器的大小、滑動視窗的移動步伐大小、池化層的大小、學習丟棄的比率來進行動態參數測試。

3 資料探索與前置處理

本研究使用的資料由肺部圖像數據庫聯盟 (The Lung Image Database Consortium image collection; 簡稱 LIDC-IDRI) 所建置的網站『The Cancer Image Archive (TCIA) Public Access』取得資料，該網站提供原始 DICOM 格式影像資料與臨床診斷和肺癌篩檢胸部斷層攝影掃描及標示病灶的註釋，這是由『國家癌症研究所』(National Cancer Institute; 簡稱 NCI) 發起，透過網絡公開存取方式的國際資源，再由『國家衛生研究院基金會』(Foundation for the National Institutes of Health; 簡稱 FNIH) 進一步推動，並在『食品與藥物管理局』(Food and Drug Administration; 簡稱 FDA) 積極參與完成資料收集，資料可以提供全世界有興趣做相關研究的學者使用，可用於肺癌檢測和電腦輔助診斷 (Computer Assisted Diagnostic; 簡稱 CAD) 開發、培訓與評估。

DICOM 是一種用於醫學影像處理、儲存、列印及傳輸的一種標準協定之檔案格式；也是大部分醫學硬體廠商所採用的影像標準格式。檔案內包含很多不同的識別標籤 (ID's Tag) 來記錄該影像的相關資訊，包括病人資訊、製造商、影像規格、像素資料 (pixel data)、... 等等。本研究資料使用網站所提供最原始 DICOM 影像的資料；也就是未經整理與萃取過的原始資料，主要要解釋資料處理前置處理過程，前置作業細項分類包含 (1) 資料萃取、探索與篩選；(2) 分類標籤、抽樣與檔案轉換；(3) 亨斯菲爾德單位轉換；(4) 標準化影像與調整影像大小。以下針對每一步驟詳細討論。

3.1 資料萃取、探索與篩選

本研究從網站總共下載 124 GB (Giga Bytes) 位元資料，涵蓋 1,010 位受試者，總共有 244,617 個切片影像檔案。除影像檔，資料還包含七個學術中心和八家醫學影像公司合作收集的診斷結果和肺癌篩檢胸部電腦斷層掃描標註說明，而標註說明是由四位有經驗胸部放射科醫師進行兩階段圖像注釋所得的結果。第一階段採盲目閱讀階段 (blinded-read phase)，每位放射科醫師獨立查驗每次 CT 掃描的結果。第二階段為非盲目階段，也就是每位放射科醫師獨立審查自己的標記，再與其他三名放射科醫師的匿名標記比較討論，再提出最終意見，該過程會盡可能識別每次 CT 掃描的所有肺部結節，但不需要強制達成共識。

由於每位受試者有許多檔案，資料以目錄結構提供，LIDC-IDRI 為主要根目錄，其下會有 1,010 個子目錄，每個子目錄會包含一位受試者的資料，子目錄皆以 LIDC-IDRI-xxxx 方式來命名，xxxx 代表 0001-1012。因中間缺漏 2 個數據 (0238 與 0585)，子目錄雖然編號到 1,012，但只有 1,010 個病例資料。DICOM 檔案的每張圖檔規格的大小為 512×512 ，但每位受試者的圖檔數並不一致，且相鄰的圖檔並非是連續掃描出來的，但若依據 DICOM 內 Tag 名稱為 InstanceNumber 排序，則相鄰的圖檔即為掃描的連續影像。因此，後續抽樣時，並不需要特別經過排序，可以直接抽樣即可，亦可避免資料太過於整齊，對應影像圖可參考圖 3-1 與圖 3-2 [?]。

從 DICOM 檔案萃取 Tag 的相關資訊，從每一個檔案 Tag『Samples Per Pixel』(0028, 0002) 皆等於 1，亦即，影像資料的第三維度數據都是 1，可知網站提供的所有影像檔皆為灰階。另外，Tag 還有記錄掃描影像的醫學儀器的製造商，與影像型態 (modality)，包括電腦 X 光攝影 (Computed Radiography; 簡稱: CR)、電腦斷層掃描、數位 X 光攝影 (Digital Radiography; 簡稱: DX) 以及分割 (Segmentation, 簡稱: SEG) 四種。表 ?? 整理 244,617 個切片影像檔案的製造商與影像型態，影像總共來自 10 家製造商所製造的醫學儀器。為了避免規格影響最終分類，本研究採用數量最多 GE Medical Systems 製造商電腦斷層掃描資料，共計 160,048 個切片，資料涵蓋 65.4%。

表 3: 所有切片資訊彙總列表

製造商	Modality				總計
	CR	CT	DX	SEG	
GE Healthcare			16		16
3D Slicer Community				90	90
DeJarnette Research Systems	7				7
FUJI Photo Film Co., ltd.	2				2
GE Medical Systems		160,048	494		160,542
Kodak	5				5
Philips		22,314			22,314
Philips Medical Systems	16				16
Siemens	26	53,588	3		53,617
Toshiba		8,008			8,008
總計	56	243,958	513	90	244,617

3.2 分類標籤、抽樣與檔案轉換

除了 DICOM 檔案外，每個目錄還有一個 XML 格式的檔案資料，該檔案提供每位醫生對每張 CT 影像判讀結果，結果分類成三類，(1) 結節 ≥ 3 毫米，(2) 結節 < 3 毫米及 (3) 非結節 ≥ 3 毫米，對應的切片數量分別為 8,049 片，2,582 片與 4,964 片，總共有 15,595 片，其餘 144,453 片沒有標注分類結果，以下僅針對 15,595 片有 CT 診斷結果的切片進行分析。

CT 診斷結果分成三類，以下分析將資料整併成兩類，0 代表有結節與 1 代表非結節兩類，並以 Label2 表示。但 Label2 類別的 slices 數量有很大的差距，為了平衡分類的正確性，本研究採用重新抽取樣本數，使得每個類別的 slices 數量一樣，抽樣數以 slices 數量較少者為主，並取整數，經過抽樣後，整理如表 ??。依據表 ??，Label2 的總樣本數為 9,800 ($= 4,900+4,900$)，將其分成訓練集 7,800 筆與測試集 2,000。

表 4: 所有切片資訊彙總列表

	標籤 (label)	說明	切片 數量	抽出的 樣本數
Label2	0	結節 ≥ 3 毫米與結節 < 3 毫米	10,631	4,900
	1	非結節 ≥ 3 毫米	4,964	4,900

3.3 亨斯菲爾德單位轉換

電腦斷層掃描資料普遍使用亨斯菲爾德單位 (Hounsfield Unit; 簡稱 HU) 來表示 CT 數值，簡稱 HU 值，該單位運用空氣和水來衡量放射密度的量化尺度，是由 Hounsfield 所創建且命名，水與空氣的 HU 值為 0 與 -1000，其他常見體內物質相對應 HU 值可參考 [?, ?]。轉換公式定義如下：

$$\text{Image}' = \text{RescaleIntercept} + \text{RescaleSlope} \times \text{Image} \quad (5)$$

其中， Image' 為轉換後的數值， Image 為影像檔案內的數值， RescaleIntercept 為 DICOM 檔案內提供的值，代表機器照射時的截距，而 RescaleSlope 為 DICOM 檔案內提供的值，代表機器照射時的斜率。圖 ?? 繪製 LIDC-IDRI-0003 病例 HU 轉換前與轉換後的 CT 數值的直方圖，其中直方圖的座標軸均調整成相同尺度。由圖可看出轉換後的直方圖，幾乎是向左平移，從 HU 轉換後的圖形可知該影像幾乎都是空氣與水所組成。

3.4 標準化影像與調整影像大小

由於每張影像圖檔資料有非常大的全距，而變異不同，在訓練前，需先依據每張影像依據全距進行影像數值標準化，使其數值介於 0 與 1 之間。以下採用灰階數位影像來說明標準化方式：

$$x_{\text{new}} = \frac{\text{UB}_{\text{new}} - \text{LB}_{\text{new}}}{\text{UB}_{\text{old}} - \text{LB}_{\text{old}}} \times (x_{\text{old}} - \text{LB}_{\text{old}}) + \text{LB}_{\text{new}} \quad (6)$$

其中， x_{old} 為原本的影像值， LB_{old} 為原始影像切片的最小值， UB_{old} 為原始影像切片的最大值， x_{new} 為轉換後新的影像值， LB_{new} 為原始影像切片的最小值， UB_{new} 為原始影像切片的最大值。

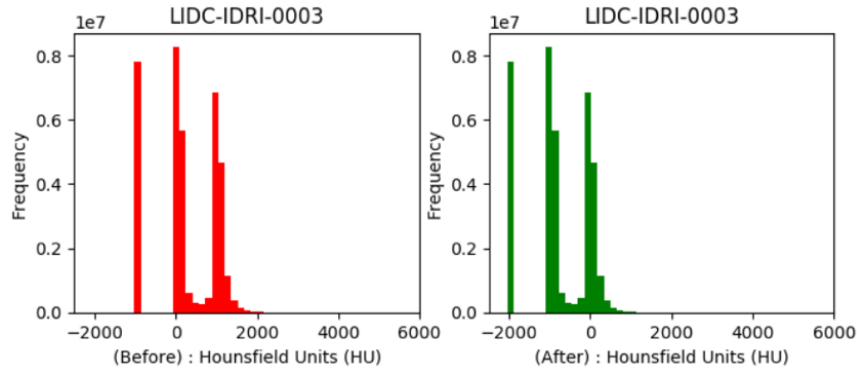


圖 13: 轉換前後 LIDC-IDRI-0003 病例 CT 數據的 HU 直方圖比較

由於轉換後的影像資料之最小值為 0，最大值為 1，將 (??) 式中的 LB_{new} 代入 0， UB_{new} 代入 1，經簡化後可得：

$$x_{new} = \frac{x_{old} - LB_{old}}{UB_{old} - LB_{old}} \quad (7)$$

再者，因原始影像資料數值應該介於 $[-1000, 400]$ ，因此，資料小於 -1,000 將會設定為 0 (黑色)，而資料大於 400 將會設定為 1 (白色)，所以 (??) 式中 LB_{old} 會固定設為 -1,000， UB_{old} 固定設為 400。

圖 ?? 從訓練集與測試集各選出一張圖來說明轉換前後的變化，除了肺部影像與背景，兩張影像在肺部影像與背景都包含一個不感興趣的影像區域，會增加訓練階段的雜訊。本研究的標準化過程也將涵蓋去除極端 HU 值，也就是 HU 值小於 -1,000 設為 0，而大於 400 設定為 1，將兩張圖的影響資料標準化後，如圖 ??，可以很明顯看出邊緣的部分皆呈現黑色 (RGB=#000000)。

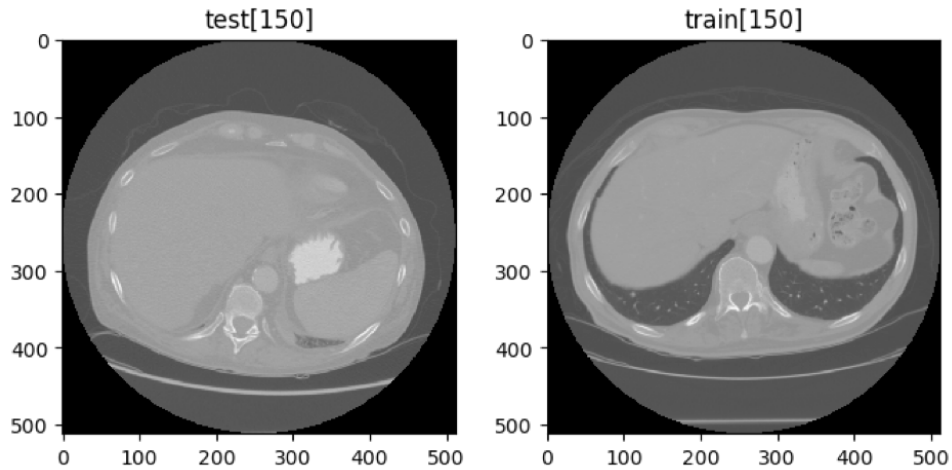


圖 14: 標準化前的原始影像

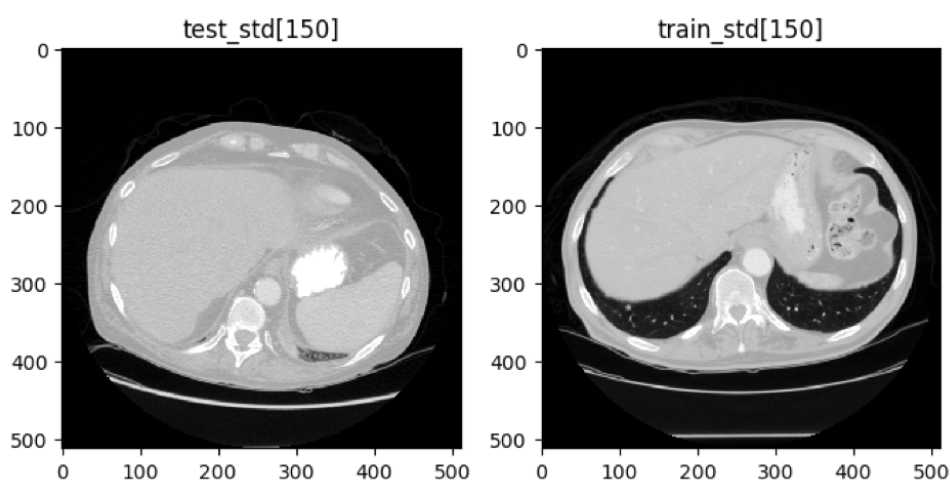


圖 15: 標準化後且有處理邊界值的影像

4 動態建模、訓練、評估與參數值的選擇

影像資料經過第一階段前置處理後，接下來進行影像建模、訓練、評估及參數調整的方式。本研究所使用的軟、硬體規格如表 ?? 所示，Keras 是一種架構在 TensorFlow、CNTK 以及 Theano 三種框架之上的高階 API (application interface)，使用上較為方便。本研究採用 TensorFlow-GPU 為底層，而 Keras 為上層來進行建模。

表 5: 軟、硬體的規格

項目	規格
中央處理器 (CPU)	Intel® Core i7-7700HQ @ 2.80GHz
記憶體 (RAM)	DDR4L 32.0GB
作業系統	Windows 10 (Home Edition)
顯示卡	NVIDIA GeForce GTX 1070 8GB 記憶體
Keras	Keras v 2.1.1
Python	Python v 3.5.3
TensorFlow-GPU	TensorFlow GPU v 1.3.0
Tensorflow-tensorboard	TensorFlow TensorBoard v 0.1.8

本研究採用卷積神經網路，基本的架構如圖 ?? 所示，前後層分別為輸入層與輸出層，中間是有 K 個卷積層與池化層所組成，每個卷積層與池化層均需選擇過濾器與移動步伐，另外，運算過程會額外利用 dropout_1 參數來減少過度配適的問題，最後訓練完就連接全連接層以及另一個 dropout_2 的設定，而全連接層則需選擇隱藏層的個數與激活函數。由於有許多參數需要設定，本研究採用動態建模方式來討論參數設定，以下先探討單一參數的表現，再進行整體參數的評估。

原本 DICOM 圖像的像素大小是 $512 \times 512=262,144$ ，若直接使用原始的圖像會非常耗費時間，且需要效能很好的硬體。由於本研究主要探討選擇參數方式。因此在標準化資料後，本研究會將影響縮小為 $128 \times 128=16,384$ ，其像素只有原本的 1/16 大小。從圖 ?? 的 512×512 像素，經過實作的轉換大小之後，成為 128×128 像素的圖 ??，除 X 與 Y 軸刻度從 512×512 換成

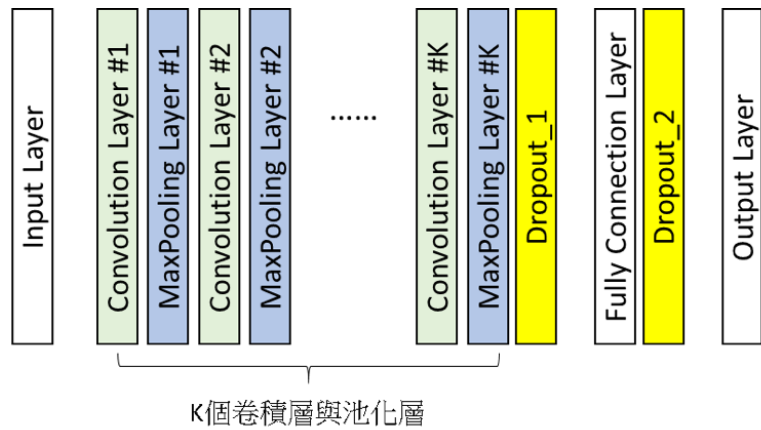


圖 16: 基本的卷積神經網路架構

128 × 128 外，影像顯示都正常。再者，因硬體效能限制與避免過度配適，本研究將 batch size 設為 32。

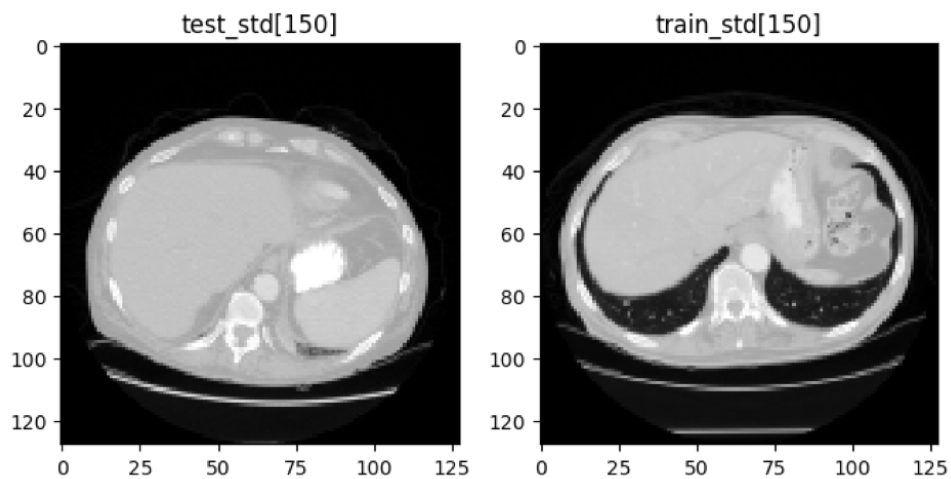


圖 17: 調整圖 ?? 為 128 × 128 的影像

4.1 探討個別參數對正確率的影響

本研究池化層採用 Max Pooling 過濾器，其餘參數會先設定一組數值，如隱藏層的激活函數，dropout_1 及 dropout_2，卷積層的過濾器核心大小與移動步伐，卷積層、池化層與隱藏層所需層數，最後是卷積層過濾器個數。

4.1.1 情境一：激活函數

隱藏層有兩種常用的激活函數， $\text{Relu}(x)$ 的門檻設定在 0，當 x 小於 0 函數值設定為 0，屬於比較極端的一種，而 $\text{Sigmoid}(\cdot)$ 提供比較緩和的數值變化。為選擇一個比較適合的激活函數，表 ?? 提供各類參數的設定，透過隨機抽樣參數，來選擇激活函數 $\text{Relu}(x)$ 與 $\text{Sigmoid}(\cdot)$ ，也就是利用相同參數值的情況下，比較這兩種激活函數所得分類的正確率。

表 6: 情境一與二之各種參數的可能設定

序號	參數	情境一	情境二	說明
1	Filters	(16, 32, 64, 128)	32	過濾器的個數
2	Kernel_size	(2, 4, 8, 16, 32, 64)	(2, 4, 8, 16, 32, 64)	過濾器核心大小
3	Strides	(2, 4, 8, 16, 32, 64)	(2, 4, 8, 16, 32, 64)	滑動視窗每次移動的大小規格
4	Pool_size	(2, 4, 8, 16)	2	池化層的大小
5	Dropout_1	(0.1, 0.25, 0.4, 0.5)	0.25	丟棄的比率
6	Dropout_2	(0.1, 0.25, 0.4, 0.5)	0.25	丟棄的比率
7	Layers	2	2	卷積層與池化層的 K 層
8	nLabel	2	2	固定值, 標籤個數 (0=node 與 1=non-node)
9	wSize	128	128	固定值, 輸入影像圖的寬度
10	hSize	128	128	固定值, 輸入影像圖的高度

在相同參數值的情況下, 在相同參數值的情況下, 經過 30 次的隨機抽樣, 每一次的訓練都是重複 30 個時期 (epoch), 分別使用 Relu(·) 與 Sigmoid(·) 進行訓練與預測結果, 利用盒形圖繪製出正確率的結果。圖 ?? 中的小三角形是平均值所在處, 而且 Relu(·) 有 50% 以上的結果其正確率都超過 0.6, 而 Sigmoid(·) 只有 25% 的結果其正確率超過 0.6, 而且有 50% 的結果其正確率都是 0.5, 與投擲一個公正銅板一樣。針對不同 Relu(·) 與 Sigmoid(·) 參數設定與正確率表現, 可參考 [?].

以下比較 Relu(·) 與 Sigmoid(·) 兩個原型激活函數的差異:

1. Relu(·) 對稱於 0, 小於 0 皆歸為 0; Sigmoid(·) 對稱於 0, 小於 0 仍有其值。
2. Relu(·) 只會將小於 0 的值歸為 0, 大於 0 的值皆保留原有值。Sigmoid(·) 則是會改變所有的原值。
3. Relu(·) 沒有最大的極限值, Sigmoid(·) 有最大的極限值 1。

基於以上三點的分析, 可推得 Relu(·) 正確率會優於 Sigmoid(·) 的原因 [?, ?], 在於 Relu(·) 僅會去除不要的值 (小於 0), 並保留原值, 而 Sigmoid(·) 會改變其值, 最嚴重的是大於 1 的值會全被改成 1, 而造成特質辨識能力變差。因此本研究的資料使用 Relu(·) 的效果比較好, 後續分析將採用 Relu(·) 激活函數。

4.1.2 情境二: 探討卷積層核心大小與步伐的關係

本情境探討卷積層的核心大小與步伐之間的關係, 考慮 6 種過濾器核心大小與滑動視窗每次移動設定, 分別為設為 2、4、8、16、32、64, 其他參數固定, 如表 ??。透過隨機抽取核心大小與步伐的大小搭配, 進行訓練, 並且找出最佳的組合。

每種組合運用 30 次的隨機抽樣, 利用 3D 立體散佈圖來呈現各別的模型訓練與測試結果, 從圖 ?? 發現, 當核心大小或與步伐變大, 準確率並沒有提升, 較高的正確率, 反而集中在較小的核心大小與步伐, 有標示座標的點都是正確率大於 0.7, 詳細資料見表 4-5 [?]。

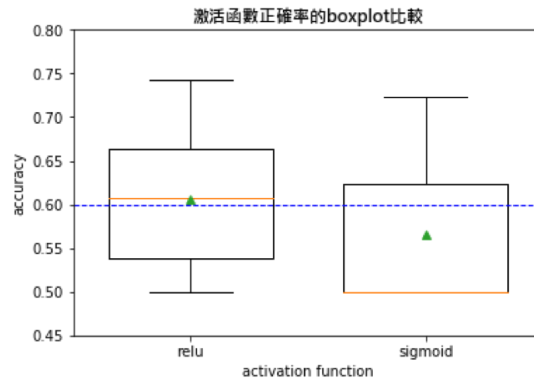


圖 18: Relu(·) 與 Sigmoid(·) 之正確率的盒型圖

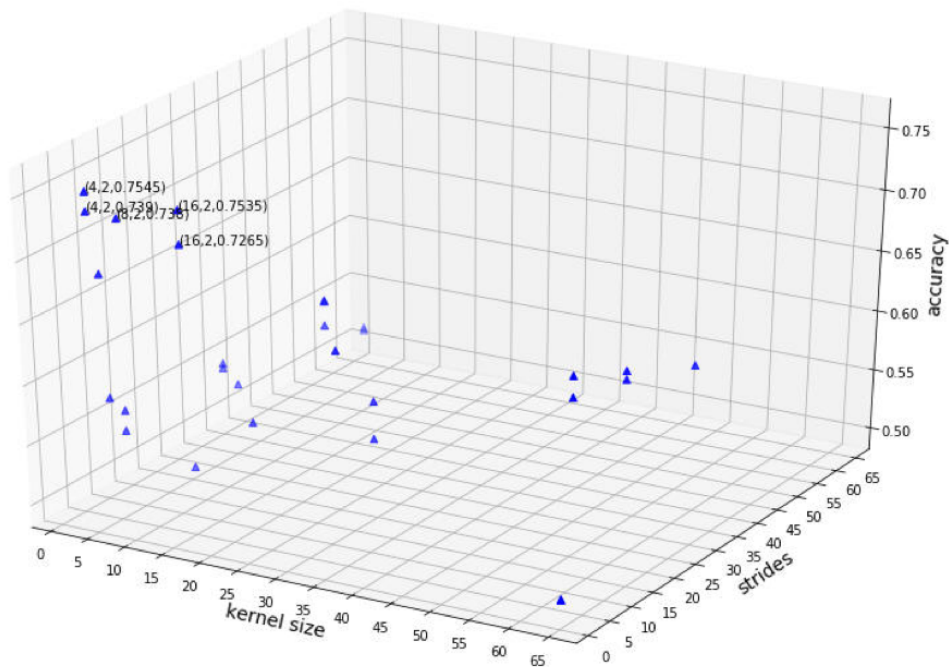


圖 19: 核心大小與步伐與正確率關聯

核心大小代表過濾器大小，也就是一種滑動視窗 (sliding window) 的概念，步伐是每次滑動的步伐大小，兩者之間會有很密切的關係。如圖 ?? 所示，如果核心大小設定為 (5, 5)，步伐設定為 (6, 6)，也就是故意將步伐的大小設定核心還要大，就會造成滑動視窗每次移動後，會有些影像圖得像素被遺漏掉，就像圖中灰色地帶就會被跳掉而遺漏。若這些灰色地帶剛好有重要的辨識資訊，就會被遺漏；尤其本研究探討的結核顆粒都很小，如果設定不當，可能就會造成分類錯誤率提高。綜合以上情境產生的數據說明，本研究將採用過濾器大小設為 `kernel_size=4` 與移動步伐大小設為 `strides=2`，本研究所考慮的資料在此設定可以有較佳的表現。

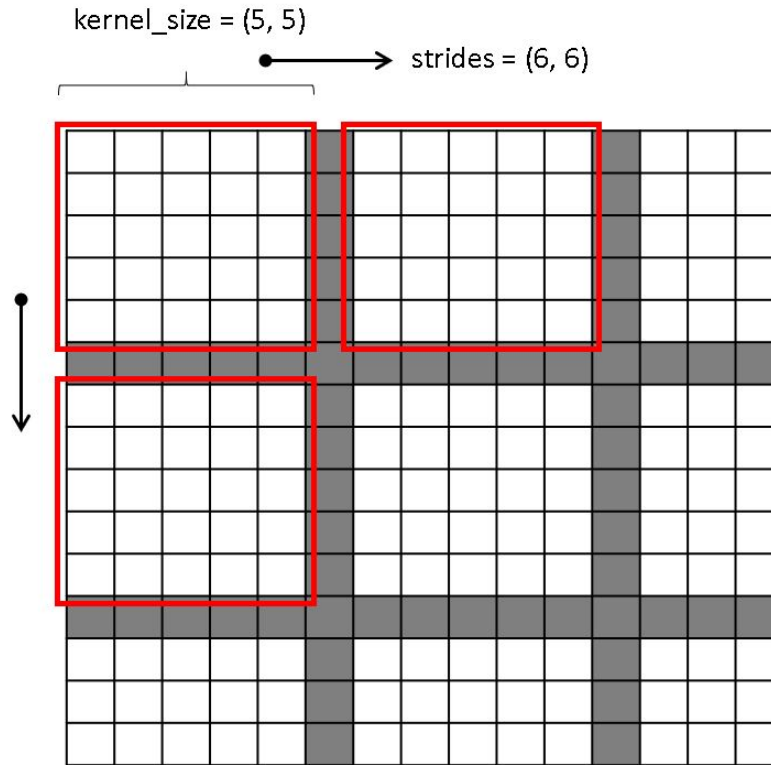


圖 20: 核心大小與步伐的互動關係

4.1.3 情境三：探討卷積層與池化層層數以及隱藏層數目對正確率的影響

本情境討論卷積層與池化層所需設定層的數量，利用表 ?? 情境一序號 1-6 的參數設定，設定可能的層數為 1 至 10 層，隨機取樣三種參數組合，三次抽樣的參數組合列於表 ??。圖 ?? 繪製出三種參數組合，準確性與層級的折線圖，由圖可知，在 $K = 6$ 之後，正確率皆呈現出 0.5，唯有在 $K = 1, \dots, 5$ 有些起伏變化，但是起伏變化太大，所以採用 $K = 2$ 較為穩定，並以此作為後續研究的依據。

表 7: 三次隨機抽樣的參數值

次數	參數名稱					
	filters	kernel_size	strides	pool_size	dropout_1	dropout_2
1	16	2	4	8	0.25	0.25
2	32	32	32	16	0.1	0
3	32	32	8	4	0.1	1

在平坦層之後的全連接層也稱之為隱藏層，以下針對隱藏層 1、2 以及 3 層來進行測試。利用表 ?? 情境一序號 1-6 的參數設定，隨機抽樣 30 種參數組合，每次均執行 30 個時期得到的結果，圖 ?? 中的正確率好像與隱藏層的層數關聯不大，但隱藏層數目越多，卻會明顯耗費計算時間，讓整個執行效率下降，本研究將以 1 層來進行後續的研究。

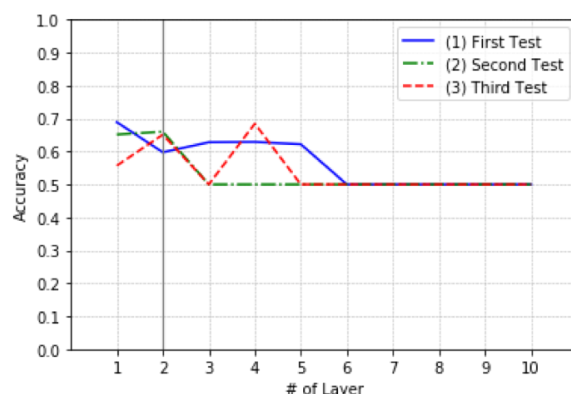


圖 21: 層數與正確率的關聯

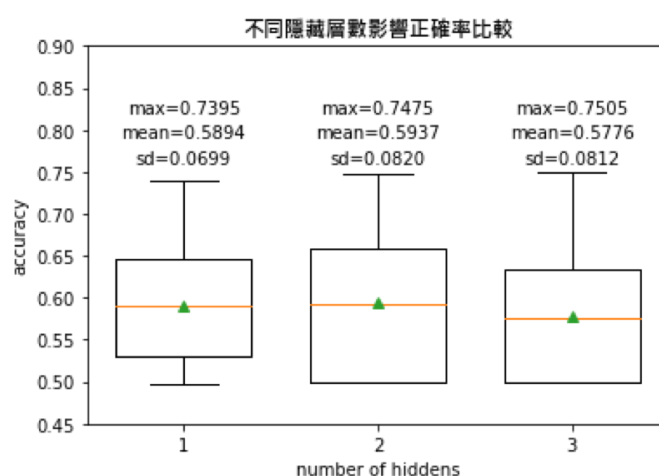


圖 22: 不同隱藏層數目之正確率的盒型圖

4.1.4 情境四：探討卷積層過濾器個數的影響

本情境探討卷積層過濾器的個數設定與正確率之關聯，過濾器個數分別設定為 16, 32, 64, 128 來進行測試，其餘的參數利用表 ?? 情境一序號 1-6 來設定，隨機抽樣 15 種組合，每一次抽樣後依據相同的參數各別進行不同過濾器的模型訓練與預測。圖 ?? 繪製測試結果的折線圖，由圖可知，過濾器的多寡與準確率沒有明顯關聯，但從盒型圖來看，可以發現過濾器個數設定為 32 時，有 50 % 左右的結果其正確率超過 0.65，而且沒有特別低至 0.50 的情況發生。四種設定的正確率的平均值，如圖 ?? 中小三角形所示， $K = 32$ 也是四個設定中最高的，所以後續將設定過濾器個數為 32。

綜合以上的情境設計與討論，表 ?? 整理出最合適的參數組合，下一節會利用這些參數值組合來進行訓練。

4.2 綜合整體參數探討與優化

在上一節是針對個別的單一參數來進行探討，本節將針對整體參數來進行比較，以下將以兩個情境進行，第一個是針對所有參數進行隨機抽樣的組合方式，第二個是針對上一節找出的較佳設定

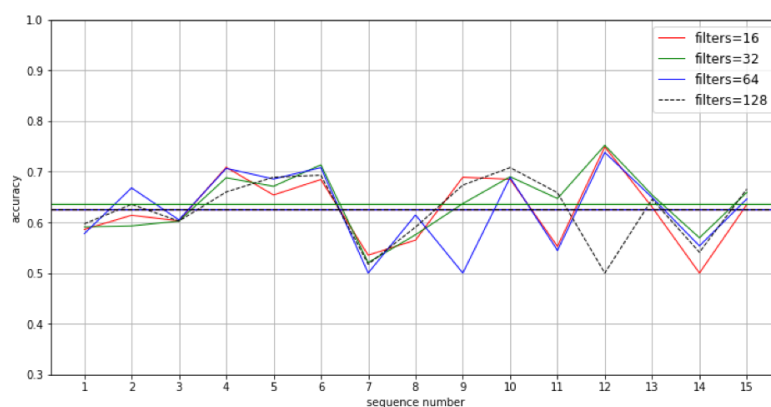


圖 23: 過濾器個數對正確率的影響

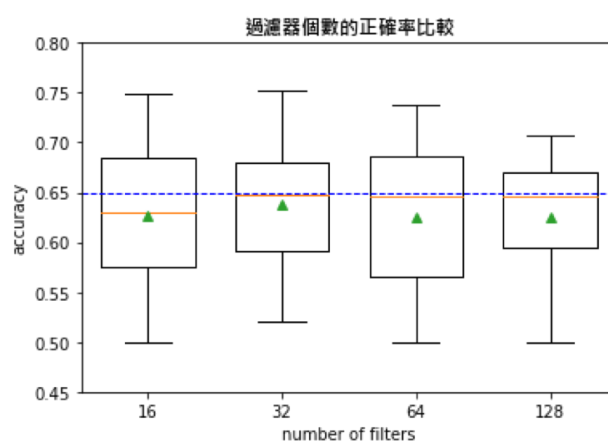


圖 24: 過濾器個數對正確率的影響

表 8: 各種參數的最佳設定

序號	參數	最佳值	說明
1	Filters	32	過濾器的個數
2	Kernel_size	4	過濾器的大小規格
3	Strides	2	滑動視窗每次移動的大小規格
4	Pool_size	2	池化層的大小
5	Dropout_1	0.25	丟棄的比率
6	Dropout_2	0.25	丟棄的比率
7	Layers	2	卷積層與池化層的 K 層
8	nLabel	2	固定值, 標籤個數 (0 : nodule 與 1 : non-nodule)
9	wSize	128	固定值, 輸入影像圖的寬度
10	hSize	128	固定值, 輸入影像圖的高度

值的固定組合，兩組來進行比較。

4.2.1 情境五：所有參數採取隨機抽樣的組合

本情境採用隨機抽樣參數方式來進行資料訓練，也就是所有的參數，包括 filters、kernel_size、strides、pool_size、dropout_1 以及 dropout_2，皆進行隨機抽樣的組合，並連續進行 30 次的模型動態建模、訓練與測試正確率的評估。參考表 ??，除了 7-10 (layers, nLabel, wSize, hSize) 的參數固定外，其他都進行隨機抽樣。執行結果如表 ?? 所示，其中只有兩次正確率超過 0.7，而 30 次的平均值與標準差分別為 0.5974 與 0.0652。

4.2.2 情境六：所有參數採取固定組合

本情境設計的參數皆是來自於情境一至情境四所產生出來參數值的組合，包括 filters=32、kernel_size=4、strides=2、pool_size=2、dropout_1=0.25 以及 dropout_2=0.25，相同的參數值組合，重複進行 30 次的動態建模、訓練與測試正確率的評估。執行出來的結果如表 4-9 所示，其中正確率幾乎都超過 0.7，而 30 次的平均值與標準差分別為 0.7422 與 0.0071。詳細數據見表 4-9 [?]

5 結論

深度學習主要有兩個步驟：(一) 前置作業，(二) 建模、訓練與評估分析。第一階段是前置作業，主要的工作項目為探索所下載資料，去除不符合或不需要的雜質資訊或是影像資料，並經過影像檔案的轉換與調整圖檔大小。第二階段，是將第一階段處理過後的影像資料檔案，當成此階段的輸入資料。建模、訓練與評估分析主要是透過建模，並將第一階段整理後的資料進入訓練、參數調整以及模型評估。將上述的兩個階段繪製成圖 ??，左邊是屬於第一階段的前置處理，利用 Python 程式語言的撰寫，將下載的 DICOM 檔案經過不同的處理方式，最後產生大小的影像檔案，右邊則是第二階段訓練階段，先將資料分割成訓練集與測試集，利用 Keras 內建的模組，選擇合適的超參數，本論文建議採用動態方式選擇，再輸出最終訓練與測試結果。

本研究資料來源是自『The Cancer Image Archive (TCIA) Public Access』網站下載 124 GB 最原始的 DICOM 影像檔案，資料格式包括 CT、DX、CR 以及 SEG 等四種型態，涵蓋 1,010 位病例的數據資料，共計 244,617 個切片影像檔案。透過進行前置的資料萃取、探索、篩選、標籤、抽樣、轉換、影像處理以及後續的動態建模、訓練與評估完成整個流程。由於肺部結核通常都是以毫米為單位，顆粒較小，辨識困難度高，所以前置處理非常重要，若可以去除雜訊，鎖定重要的結節與非結節影像部位，將可以有效提升辨識率。

5.1 前置作業的探討

本研究使用最原始未經處理過的 DICOM 影像檔，資料的預處理會明顯影響後續準確性評估。為減少變異來源，全部 244,617 個切片影像檔案，本研究僅使用有相同製造商的醫學儀器 160,048 個切片電腦斷層掃描出來的影像檔案，並從中選出有標記的 15,595 筆資料進行影像轉換與處理，包括電腦斷層掃描中，將 CT 影像改成放射密度量化尺度亨斯菲爾德單位，並且運用標準化去除邊界值，來提升辨識效果。為了不讓資料失去平衡，所以從表 ?? 標籤 0 與 1 中各抽出 4,900 筆的資料，也就是 1:1 的平衡資料，並自各別的 4,900 中，再抽出 1,000 筆當成測試資料。最

表 9: 隨機抽樣 30 組參數的執行結果

重複次數	正確率	filters	kernel_size	strides	pool_size	dropout_1	dropput_2
1	0.5855	16	2	8	8	0.4	0.25
2	0.656	128	32	8	8	0.4	0
3	0.7365	64	4	2	2	0.1	0.5
4	0.6785	32	4	4	2	0.4	1
5	0.66	32	32	64	2	0.5	1
6	0.6125	128	32	32	8	0.5	0.25
7	0.6345	128	32	16	4	0.1	1
8	0.512	128	2	64	8	0.1	0.4
9	0.589	128	2	8	4	0.1	0.25
10	0.592	128	16	64	4	0.4	0
11	0.555	16	8	64	2	0.1	0.5
12	0.5375	32	2	16	8	0.25	1
13	0.5	32	8	2	4	0.5	0
14	0.5585	32	4	32	16	0.25	1
15	0.6805	128	32	8	8	0.4	0.5
16	0.6275	128	2	2	16	0.4	0
17	0.6145	64	16	32	8	0.4	0.25
18	0.6145	16	16	64	2	0.1	0.4
19	0.7265	32	4	2	2	0.4	0.4
20	0.561	32	4	32	4	0.25	0.4
21	0.5	32	32	4	16	0.5	1
22	0.671	64	16	8	16	0.5	0.5
23	0.6385	16	64	32	2	0.1	0.4
24	0.5	64	2	32	8	0.25	0.5
25	0.5455	128	2	32	16	0.5	1
26	0.548	32	2	32	4	0.1	0
27	0.5335	32	4	32	8	0.4	1
28	0.5465	64	4	32	16	0.4	0
29	0.607	16	16	32	16	0.5	0.5
30	0.601	64	16	32	16	0.5	0.5

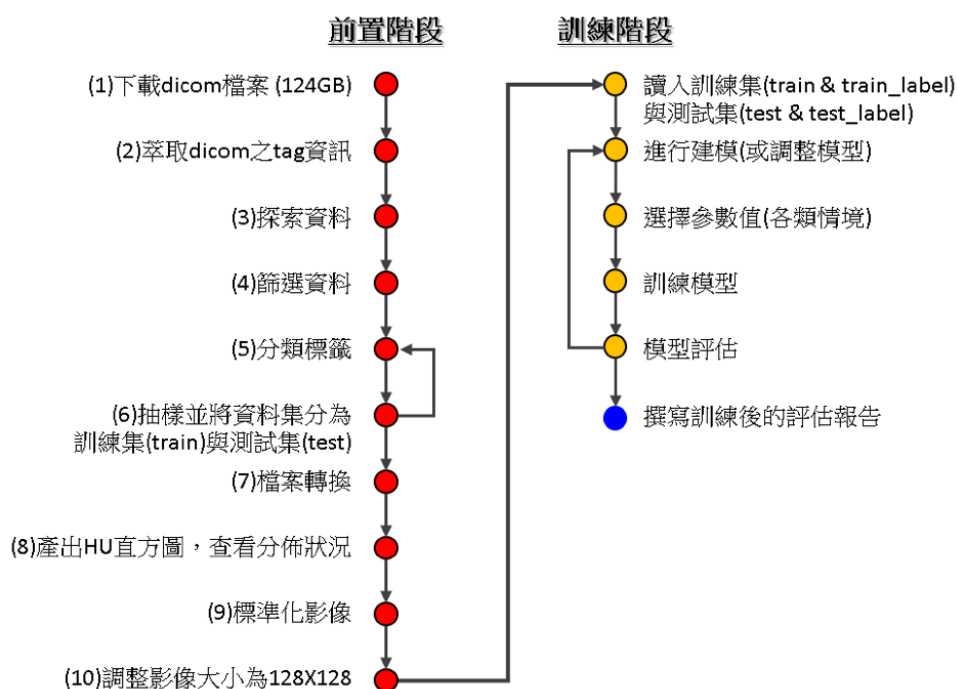


圖 25: 資料前處理步驟與建模流程

後，為避免訓練結果受到資料變異的影響，所以資料需要進行標準化。再者，原本影像的像素為 512×512 ，圖型檔資料太大，無法順利載入 GPU 為 8GB 記憶體處理，電腦常因訓練過程資源耗盡而當機，所以像素重新調整為 128×128 ，才順利使用 GPU 的運算資源，節省許多的訓練時間。

在研究初期沒有比較轉換資料前後的情況，造成後續資料辨識的正確率一直維持在 0.5，敏感度也維持在 0.5，等同於沒有辨識能力。經過逐一驗證後，才發現轉換資料後，不小心將影像轉換成為全黑屏，經過標準化修正後，才讓模型有辨識能力。所以，研究將所有影像資料的轉換前、後影像都呈現出來驗證無誤，才不會影響後續模型的辨識能力。

如果預處理能透過資料切割的影像處理程序，像是門檻演算法、分水嶺演算法等等的方式，都可以將影像資料的主體勾勒出來，並遮罩掉 (mask) 其他不重要的雜訊資料，或是將結節與非結節目標鎖定，縮小辨識範圍，並將其其他不重要的影像資料全數遮罩掉，但由於電腦效能與研究著重的方向，本研究並沒有採用資料切割與目標鎖定的處理方式，這將會是未來研究可以再深入探討之處。

5.2 動態建模與參數選擇的結論與討論

表 ?? 摘要出所有參數的變化情況，先從情境一的激活選擇後，情境二探討核心大小與步伐的彼此關係、情境三對神經網路內層層數的探討、情境四隊過濾器數目的選擇，最後以情境六、七來做最後比較這些參數對模型最佳化與穩定性。

綜合情境五與情境六的產出結果，利用視覺化的盒型圖 (見圖 ??)，可以更明確的比較其間的差異性。明顯地，隨機抽樣參數的正確率全距 $[0.5, 0.7365]$ 分佈非常分散，而且最高的正確率僅僅只有 0.7365，而正確率大於 0.65 的比率也不及 25%；固定參數的正確率全距 $[0.726, 0.759]$

表 10: 情境參數設定摘要

參數	情境二	情境三	情境四	情境五	情境六
Filters	32	抽樣	16,32,64,128	抽樣	32
Kernel_size	抽樣	抽樣	抽樣	抽樣	4
Strides	抽樣	抽樣	抽樣	抽樣	2
Pool_size	2	抽樣	抽樣	抽樣	2
Dropout_1	0.25	抽樣	抽樣	抽樣	0.25
Dropout_2	0.25	抽樣	抽樣	抽樣	0.25
Layers	2	1-10	2	2	2
nLabel	2	2	2	2	2
wSize	128	128	128	128	128
hSize	128	128	128	128	128

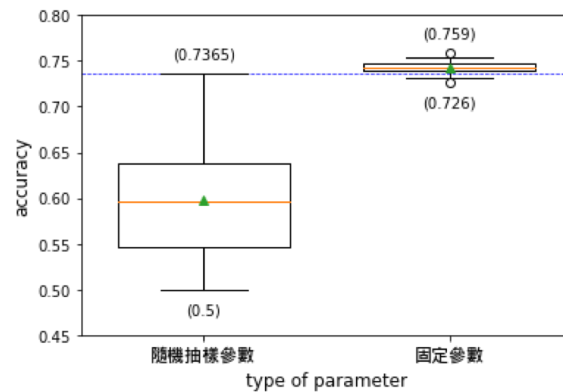


圖 26: 隨機抽樣參數與固定參數之盒型圖比較

非常集中，而且正確率超過 0.7365 就大於 75% 以上，表示是一個預測能力非常穩定的一個模型。表 4-11 是兩種建置模型的敘述統計量比較。

本研究利用六個情境實作測試該資料最佳參數組合 (見表 ??)。隨機抽樣與透過參數最佳化的組合 (就是所謂的固定參數)，兩者的相關統計量 (見表 ??)，可見最佳組合的正確率標準差非常小，也表示此組參數的組合模型較為穩定，正確率平均值部份也相對較優。

本研究採用動態建模方式來選擇參數設定，主要透過參數組設定，利用程式傳入不同參數值，建立出不同的模型，然後再進行編譯、訓練以及測試。利用動態建模可以快速來進行不同情境的探討，免除掉人工重複不斷改寫程式來建立模型的麻煩。而動態建模、訓練以及預測評估所採用的參數設定都是離散數值或是 2 的次方，但實際設定可以採用連續整數區間。

經過以上研究所選擇的最佳組合參數值，辨識的正確率最高只有約 0.76，無法再突破的原因，有可能是前置處理將最原始影像圖檔從 512×512 縮減成 128×128 ，每個維度縮減比率為 1/4，兩個維度的縮減比率就會是 1/16，損失 15/16 的資料量，更很高機會將有興趣的結節所在之處的資料刪除。若有性能比較好的硬體，未來的研究使用原始圖檔，並考慮在前置作業，圈選出結核之處直接裁切成需要的影像大小，避免損失重點影像，造成辨識的正確率無法提高。另外，每個切片檔所對應的原始 XML 檔案會包含最多 4 位放射科醫師判讀的結果，本研究僅採用第一位

表 11: 本研究資料的最佳參數設定

參數名稱	最佳參數值
Activation Function	Relu
Filters	32
Kernel_size	4
Strides	2
Pool_size	2
Dropout_1	0.25
Dropout_2	0.25
Layers	2

表 12: 30 次隨機抽樣參數組合與最佳組合正確率的敘述性統計量

參數設定	最小值	第一四分位	中位數	第三四分位	最大值	平均值	標準差
隨機抽樣	0.5	0.5472	0.5965	0.6365	0.7365	0.5974	0.0652
最佳組合	0.726	0.7388	0.7425	0.7465	0.759	0.7422	0.0071

醫師判讀的結果當成該切片的 label，但後續分析才發現有部份切片 4 位放射科醫師會有不一致的判讀結果，不一致的比例約為 13.86%，這部分會干擾辨識正確率。

5.3 總結

本研究的重心著重於探討卷積神經網路最佳化參數的選擇，利用自建的動態建模方式，進行不同情境的情境設計模擬，反覆進行抽樣來驗證，找出最穩定且最佳化參數的模型。倘若，本研究若有更佳的研究設備，同時利用多部電腦設備以及多張顯示卡的 GPU 運算能力，就可以讓所有參數採用一個區段的連續型設定值來進行 GPU 的分散式運算估計 [?], 或許可以找出更具有辨識能力的模型參數。

運用動態模型探討卷積神經網路辨識 CT 影像內肺部結節之超參數設定研究

Using dynamic method to determine the hyper-parameters in convolutional neural network for
the classification of lung nodules in CT Images

Hsiang-Hui Chen* and Yi-Ting Hwang*¹

* Department of Statistics, National Taipei University

Harn-Jing Terng^{\$}

^{\$} Advpharma, Inc., New Taipei city, Taiwan

ABSTRACT

2D X-rays is often used to screen the lung nodule and a professional physician has to interpret the result. However, the lung nodule can be hidden by thoracic bone and the diagnostic result can be affected. Since there are new inventions in 3D imaging techniques, the low dose computed tomography (LDCT) is used to obtain image slices which are then combined into a 3D image. Using the rotation techniques, the image can be viewed from various angles. It expects to increase the accuracy of diagnostic results.

This research uses data provided by the Lung Image Database Consortium image collection. The data include lung cancer diagnoses, labelling of specific nodules and computed tomography (CT) scans data. The CT scan image data is stored in a DICOM format. The objective of this study is to clarify the procedure of data pre-processing of DICOM image data for implementing the convolutional neural network (CNN) and then to understand the feasible parameter settings for CNN to classify the lung nodule. An open source software library, TensorFlow is used to perform CNN. Since many parameters have to be specified, a dynamic modelling is proposed to choose the parameter for training the data. Finally, a best parameter setting for classifying the lung nodule for this data is recommended.

Key words and phrases: Artificial Intelligence, Convolutional Neural network (CNN), Deep Learning, Dynamic modelling, Lung Nodule.

JEL classification: C6.

¹Corresponding author. Email: hwangyt@gm.ntpu.edu.tw