以倒傳遞神經網路辨識貓咪圖片

鄭逸軒

呂仁園 老師

背景與動機

科學家們為了在電腦上讓語音以及影像辨識……等獲得像人腦辨識相似的結果，開始了類神經網路的研究。

而隨著科技日新月異的進步，類神經網路這塊領域誕生出許多富有價值的理論來應用於日常生活之中。

神經網路對於輸入對應到輸出有著**記憶**與**學習**的功能，並對未知的輸入有推廣性的作用，因此類神經網路可運用於各種領域中。(EX: 工業產品分析、股市投資、影像辨識……等)

現有的神經網路演算法當中，以倒傳遞演算法(*Backpropagation*)所建立的神經網路之辨識效果最好。因此本專題欲使用此演算法來建立一個簡單的辨識網路，且設計一套方法和以辨識寵物貓圖片為範例來呈現成果。

目標

使用 python 程式語言辨且運用類神經網路之理論，實作出一個辨識程式。能夠辨識一張未知的貓咪圖片然後告訴使用者這隻貓咪的品種為何。

設計兩種方法來比較辨識度的高低，分別是以貓頭辨識和以分成三部位(左耳、右耳、臉)來辨識再加以整合。預期使用三部位辨識會比使用貓頭辨識的辨識率還要高。

架構

神經網路採用多層式前向神經網路，分別有輸入、隱藏、輸出三層。(如圖.一)

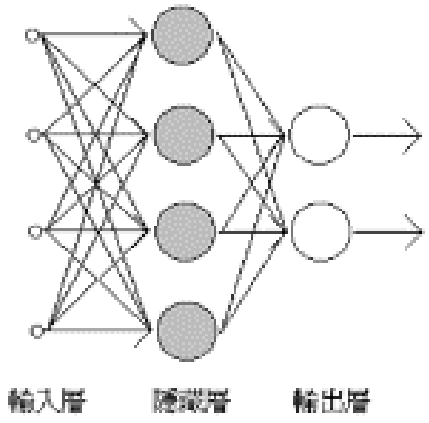

圖.一

使用的辨識演算法為倒傳遞演算法。提供訓練的貓咪圖片為 70 張以及測試網路用的圖片 40 張，總共 110 張。

另外將設計一個簡單的圖形使用者介面來做 DEMO。使用者介面可以提供使用者訓練網路以及輸入任意圖片檔案來做辨識。

以下定義程式之輸入以及輸出:

* 程式內部:
    * Input: 貓咪圖片 (10 x 10 矩陣)
    * Output: 品種的分數值

* 使用者:
    * Input: 貓咪圖片 (10 x 10 矩陣)
    * Output: 辨識結果 (String 訊息)

實作方法

訓練部分，將不同品種的寵物貓圖片手動擷取臉部，將之另存成正方形的圖片來作訓練用樣本。接著辨識程式讀入訓練樣本，將彩色圖片轉換成灰階圖片。然後將轉換過的圖片轉化成矩陣資料以計算特徵值。最後將特徵值作為辨識網路的輸入，和設定一個對應的輸出(正確答案)來訓練網路。

訓練網路時先將連接網路的權重隨機從-10 到 10 初始化，接著透過一組組訓練資料來計算和正確答案的誤差，接著回傳至隱藏層和輸入兩層以更改權重值使網路連接逐漸趨向正確。

測驗部分，同上做成的驗證用樣本，並同樣執行上述處理圖片的動作。將計算完的特徵值做為辨識網路的輸入，然後根據訓練完的網路計算輸出，並且顯示辨識結果。

作為輸入用的寵物貓圖片為十乘十像素大小的圖片，在計算特徵值的時候會得到 10 個特徵值。欲將這些特徵值做為神經網路的輸入，所以神經網路的輸入層有 10 個輸入神經元。辨識的寵物貓種類為 5 種，因此將網路之輸出神經元設定成 5 個。而一般來說隱藏層神經元會取輸入層和輸出層神經元總和平均，因此在此設定為 8 個隱藏神經元。

激發神經元用的活化函數使用 S 型函數(Sigmoid function)（圖.二）

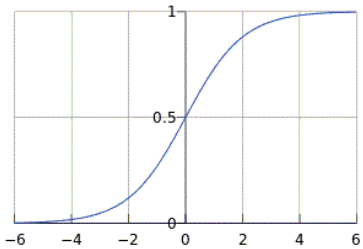

圖.二

倒傳遞演算法為監督式演算法，訓練過程中首先隨機初始化神經元間的權重，

接著提供一組包含輸入資料以及期望結果的學習樣本並利用目前的權重計算輸出結果。然後計算輸出結果和期望結果的誤差來回傳給隱藏及輸入層。根據回傳之誤差來調整與隱藏層和輸入層間的權重。重複上述動作直到誤差收斂。

所使用到的公式如下:

$$E = \frac{1}{2}\sum_{k=1}^{K}(D_k - net_k)^2$$

D_k　　輸出層第k個神經元的目標輸出

E 為誤差，net 為計算後的輸出。

權重調整公式如下:

$$w_{ij} = w_{ij} + \eta \times \delta_j \times net_i$$

$$w_{jk} = w_{jk} + \eta \times \delta_k \times net_j$$

$$\underbrace{\sum_{k=1}^{K}\left[\delta_k \times w_{jk}\right] \times \left(net_j \times \left(1 - net_j\right)\right)}_{\delta_j} \times net_i$$

$$\underbrace{\left(D_k - net_k\right) \times \left(net_k \times \left(1 - net_k\right)\right)}_{\delta_k} \times net_j$$

W 為權重，η 為學習率(預設為 0.5) 。

方法一 ： 以貓頭辨識

圖片使用 10 乘 10 之像素，如以下的貓頭圖片(圖.三)

圖.三

主要程式碼:

```
貓樣本=[]
自動取樣本("D:/類神經網路/貓/訓練/頭/",貓樣本)
#-----------------------------------------------------
if(模式=="訓練"):
    n = 神經網路(10, 8, 5)
    n.訓練(貓樣本,"頭")
elif(模式=="測驗"):
    n = 神經網路(10, 8, 5,模式,"頭")

正確答案 = dict()
貓測試 =[]
檔名陣列 =[]
自動取測試("D:/類神經網路/貓/測試/頭/",貓測試,正確答案,檔名陣
列)
```

n. 測驗(貓測試, 1, "除蟲", 正確答案, 檔名陣列)

以上會先自動到我建置的訓練資料夾內讀取訓練圖片資料(如圖. 4),接著以讀入的訓練資料來訓練網路。

圖. 四

如果是訓練階段的時候，建立網路的權重隨機初始化來訓練，訓練完會另外存成檔案。如果是在測試階段則是直接讀入訓練完的權重來當作初始權重。(如圖. 五)

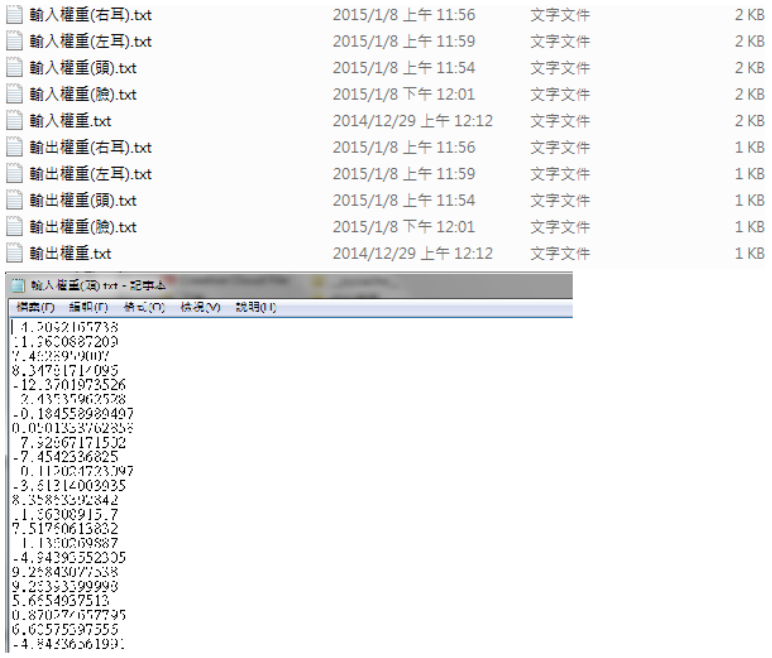

圖. 五

之後自動取測試資料夾的資料和設定字確答案來做測試,回報正確率即完成方法一。

方法二 : 分成三個部分辨識

圖片使用 10 乘 10 之像素，如以下的三個部位(左耳、右耳、臉)之圖片。(圖.六)

圖.六

主要程式碼:

```
右耳樣本=[]
自動取樣本("D:/類神經網路/貓/訓練/右耳/",右耳樣本)

左耳樣本=[]
自動取樣本("D:/類神經網路/貓/訓練/左耳/",左耳樣本)

臉樣本=[]
自動取樣本("D:/類神經網路/貓/訓練/臉/",臉樣本)

if(模式=="訓練"):
    右耳 = 神經網路(10, 8, 5)
    左耳 = 神經網路(10, 8, 5)
```

```
    臉 = 神經網路(10, 8, 5)

    右耳.訓練(右耳樣本,"右耳")
    左耳.訓練(左耳樣本,"左耳")
    臉.訓練(臉樣本,"臉")

elif(模式=="測驗"):
    右耳 = 神經網路(10, 8, 5,模式,"右耳")
    左耳 = 神經網路(10, 8, 5,模式,"左耳")
    臉 = 神經網路(10, 8, 5,模式,"臉")
```

基本上與方法一之訓練方式相同，只是因為要辨識三個部分所以要建置三個辨識網路。

```
    右耳結果,右耳值 = 右耳.測驗(右耳測試,方法=2)
    種類個數[右耳結果]+=1
    三部分結果["右耳"]= 右耳結果

    左耳結果,左耳值 = 左耳.測驗(左耳測試,方法=2)
    種類個數[左耳結果]+=1
    三部分結果["左耳"]= 左耳結果

    臉結果,臉值 = 臉.測驗(臉測試,方法=2)
    種類個數[臉結果]+=1
    三部分結果["臉"]= 臉結果

    最可能的種類= 0
```

```
for z in range(5):
    if(種類個數[z+1] > 最可能的種類):
        最可能的種類 = z+1

if(種類個數[最可能的種類]==1):
    三部分最大值= 右耳值
    最可能 = "右耳"
    if(左耳值>三部分最大值):
        三部分最大值 = 左耳值
        最可能 = "左耳"
    if(臉值>三部分最大值):
        三部分最大值 = 臉值
        最可能 = "臉"
    最可能的種類 = 三部分結果[最可能]
```

三個部分的辨識結果分別回傳回來之後，會去看三個部分當中哪種品種結果占有比較多的部位(即占有兩個部位以上)。例如有兩個部位辨識為波斯貓，另一個部位辨識為折耳貓，則最終辨識結果為波斯貓。

若三個部位都辨識為三種不同的貓時，會根據他們分別的輸出分數值大小來做依據，最終結果會以擁有最大的分數值為結果。

圖片特徵值擷取

將 10 乘 10 像素之彩色圖片轉換成灰階圖片，接著轉換成 10 乘 10 之矩陣以方便計算 10 個特徵值來做為網路之輸入。

主要程式碼:

```
def 取灰階圖特徵值(來源):
    彩圖 = Image.open(來源)
    灰階圖 = 彩圖.convert("L")
    圖矩陣 = asarray(灰階圖)
    特徵值,特徵矩陣 = linalg.eigh(圖矩陣)  #取特徵值和特徵向量
    特徵值 = 特徵值/100000
    return(特徵值)
```

因為 S 型函數的特性，在輸入值為-1 到 1 之間時有最好的輸出結果(彼此輸出有差別)否則太大的輸入值會使輸出值恆為 1，造成無法訓練。因此我將特徵值縮小 100000 倍。經過實驗，若縮小倍率變小會讓訓練誤差無法收斂到最佳值。增加倍率雖能有機會將誤差收斂至最小但要花上大量的時間。

使用者介面

提供一個簡單的使用者介面來做展示之用。提供之功能有訓練網路和讓使用者自行輸入想辨識的貓咪圖片檔案。(圖. 七)(圖. 八)並且回覆辨識結果於介面上提供給使用者參考。(圖. 9)

圖. 七


DEMO
訓練
測驗
請選擇下列想辨識的貓，然後按下測驗
或是訓練網路
開啟舊檔
搜尋位置(I):
頭
最近的位置
桌面
媒體櫃
電腦
網路
波斯1.jpg
波斯2.jpg
波斯14.jpg
波斯15.jpg
檔案名稱(N):
波斯14.jpg
檔案類型(T):
All Files (*.*)
開啟舊檔(
取消

圖. 八


DEMO
訓練
測驗
我覺得這是波斯貓

圖. 九

專題成果

品種設定為五種貓，70 張訓練樣本與 40 張測試樣本中，使用貓頭辨識的正確率為 40%；而使用三部分辨識的正確率為 52.5%。如果是五種品種隨便猜，正確率為 20%，可見使用此理論建置出來的辨識程式使得電腦有一定程度的智慧來辨識五種貓咪圖片的品種。