**Structure：**

在basic的binary classifiers我使用了三層hidden layer，activation function都是使用relu，而最後的output layer因為要做二元分類所以使用的是sigmoid，總共五層layer(包含input layer)的neural network，每一層的nodes數分別為30(input layer)，20, 10, 5, 1(output layer)。

在advance的multi-class classifiers裡因為一開始的units數比較多，為了避免units太快減少，我使用了四層relu的hidden layer，但output layer因為要分類十個數字所以要使用softmax，總共也是六層layer，但是因為input layer是以像素格作為參數，nodes數即為總共的像素格，高達784個，所以在後面hidden layer的nodes數也會相應變大，我選擇的hidden layer的nodes數為512，256，128，64，最後要output 10個數字的個別機率所以output layer的node數要為10。

**Improving model：**

一開始對於neural network不熟悉開了很多layer而且iteration很多次，train到一半的時候發現cost不再改變但也不敢停止，等到train了好幾個小時後發現accuracy不盡人意，後來查了google才知道layer越多層越可能發生vanish gradient problem，隨後我先將iteration的次數減少以觀察cost的變化，然後再將layer減少與對batch size做調整，到如今便形成這個五層和六層的neural network。

**Difficulty：**

* 對於Neural network的觀念和符號不熟悉
* Back propagation的數學微分難以計算
* Dimension的對應

**Summarize：**

此次的作業在實作的coding量不會比前兩次多，但在觀念上卻比前兩次都還要深，其實感覺neural network就是將前兩次的觀念都結合在一起，變成一個更大的model去做learning。而對很多東西的不熟悉當然也是花了不少時間在講義和網路資料中來回才逐漸找到實作方向和鞏固觀念。但是在back propagation的微分計算上面，首先遇到的困難是在code裡面的符號分別對應哪一項偏微分要弄清楚就花了不少時間，而後微分出來的東西打進code裡測出來發現不是因為dimension沒有對應好發生error，就是和expect answer不一樣。在很多輪運算和思考dimension後，總算逐漸實作好了部分的code，後來往下滑的時候才發現——其實公式已經寫在下面了。知道的瞬間突然覺得先前的自己很好笑，但是相信經過自己算過想過之後會對這整個運作更加熟悉。

進入了neural network，也代表正式踏入了deep learning的範疇裡，隨著final project確定題目，相信後面關於deep learning的知識將會更加有趣。