Introduction to Neural Networks II

在神經網路筆記的第二部分，我想深入討論神經網路的工作機制，嘗試得到一個明確的解釋。

首先，我們提到要對 cost function $J(\theta)$ 做梯度下降，而神經網路的 cost function 其實比 GLM 的複雜很多。

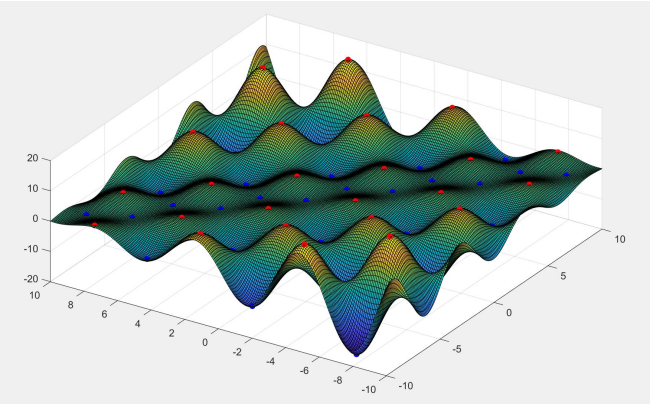
在這裡給出一個直觀想法：假設已經做完梯度下降，即參數已最佳化，這時神經網路的第一層裡，有：

$$a_1^{[1]} = \sigma(W_1^{[1]}x + b_1^{[1]})$$

$$a_2^{[1]} = \sigma(W_2^{[1]}x + b_2^{[1]})$$

$$\vdots$$

注意 $W_1^{[1]}$, $b_1^{[1]}$, $W_2^{[1]}$, $b_2^{[1]}$ 等，是參數向量 θ 裡的小組的向量，即 $\theta = (W_1^{[1]}, b_1^{[1]}, W_2^{[1]}, b_2^{[1]}, \cdots)$。如果我們交換 $a_1^{[1]}$、$a_2^{[1]}$ 的參數，那麼明顯地神經網路的輸出和 $J(\theta)$ 本身不會有任何變化，但使 $J(\theta)$ 有 local optima 的 θ 卻產生了變化，即 $\theta = (W_2^{[1]}, b_2^{[1]}, W_1^{[1]}, b_1^{[1]}, \cdots)$。對其它層的其它神經元也是一樣的。這代表參數空間中存在大量的對稱點，神經網路的成本函數其實是有多個同個值的 local optima 的函數，如果非要畫在三維空間的話，會是這個感覺：

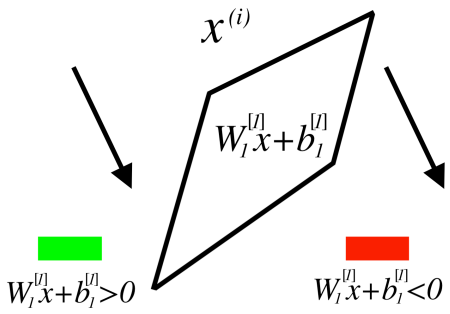


圖一：multiple local optima with the same value

這是一個神經網路的固有特性。

接下來要講的，是神經網路的表徵學習 (representative learning) 的本質。

考慮第一層的第一個神經元，輸入 input layer 的 x，觀察發生了什麼事情。



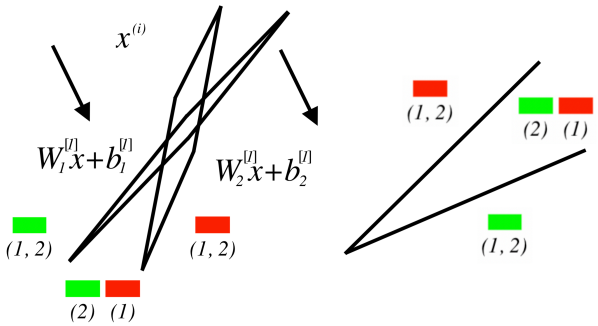
圖二：activation

$$\text{ReLU}(W_1^{[1]}x + b_1^{[1]}) = \max(0,\ W_1^{[1]}x + b_1^{[1]})$$

第一層第一個神經元 $a_1^{[1]}$ 內建一個超平面 $W_1^{[1]}x + b_1^{[1]}$，並代入 activation，輸出 $\sigma(W_1^{[1]}x + b_1^{[1]})$。如果它是 ReLU，那這個神經元對單筆數據的輸出是：

$$a_1^{[1]} = \text{ReLU}(W_1^{[1]}x^{(i)} + b_1^{[1]}) = \begin{cases} 0,\ W_1^{[1]}x^{(i)} + b_1^{[1]} \leq 0 \\ W_1^{[1]}x^{(i)} + b_1^{[1]},\ W_1^{[1]}x^{(i)} + b_1^{[1]} > 0 \end{cases}$$

在這裡我有一個想法。就算我們一開始並不知道神經網路的原理，但根據定義，在 cost function 最小時它能擬合出最符合數據分佈的模型，這時幾何上會發生什麼？

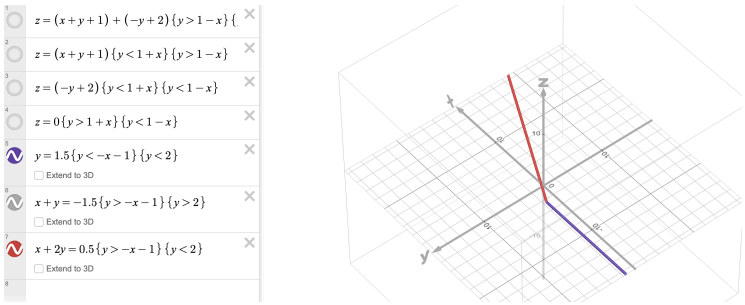


圖三：刺激神經元

如果神經網路直接在這裡結束（輸入層 $\rightarrow$ 隱藏層 $=$ 輸出層，輸出 $a_1^{[1]},\ a_2^{[1]}$），那我們會發現每個神經元是利用了 activation function 的特性，將空間切割成兩個區域——activated / not activated，而且前者還能描述激活的程度，進而直接地導致模型的輸出，例如圖三，在 $a_1^{[1]},\ a_2^{[1]}$ activated 時模型輸出 $(W_1^{[1]}x + b_1^{[1]},\ W_2^{[1]}x + b_2^{[1]})$，在 $a_2^{[1]}$ activated 時模型輸出 $(0,\ W_2^{[1]}x + b_2^{[1]})$，在 $a_1^{[1]},\ a_2^{[1]}$ inactivated 時模型輸出 $(0,\ 0)$。

同理，多個神經元就會將空間分割成多個區域，每個區域會導致不同的輸出。這也是我覺得神經元就像 perceptron 一樣的原因。

在第一層中，每個神經元學將輸入空間劃分為兩個區域：激活區域（超平面一側）和非激活區域（超平面另一側），接收刺激（輸入）來激活。當我們堆疊更多的層時，每一層都對前一層的非線性表示進行新的線性組合和激活。這使得網路能夠逐步學習到更複雜的非線性函數，並在原始輸入空間中形成分段線性逼近的任意形狀決策邊界。在 $\sigma = \text{ReLU}$ 的情況下，決策邊界的形成依賴於不同神經元在不同輸入區域的激活與否，從而將輸入空間分割成多個子區域，每個子區域對應著不同的模型輸出。
為了驗證是否真的能學到任意的決策邊界，我假設了一個簡單的神經網路（兩層，各兩個神經元），並在 desmos 網站上畫函數圖：

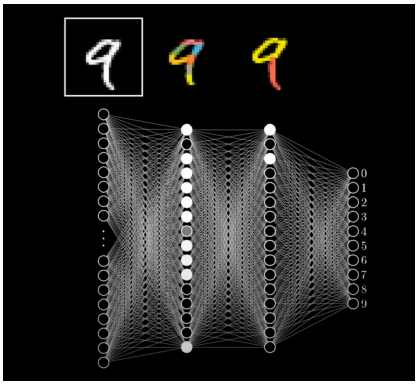


圖四：驗證理論正確性

在計算的過程中，我們會發現由 ReLU 構成的神經網路所形成的決策邊界是 piecewise linear 的，而分段會隨著神經元數與層數變多而變得精細，設定足夠多的話，理論上能逼近任意複雜的非線性邊界。計算後我發現，神經網路的決策邊界就是由通過每一層神經元的激活決定的超平面交集而決定（至少 ReLU 是這樣的）。

Representative Learning

想像這時神經網路已經學習完畢，那麼第一層的那些負責激活神經元的超平面，具體的意義是什麼呢？事實上，它們只是梯度下降後產生的隨機線性組合，人類無法賦予任何意義，卻是機器確實學習到的東西。一直到第 n 層神經元的那些超平面，也只是機器基於前面的層學習到的東西。在一些科普影片中，我們可能會看到這樣的圖：



圖五：圖像辨識任務

在這個訓練神經網路辨識數字 0 到 9 的任務中，大部分的科普影片、甚至是教科書都會說，第一層會學習到原始輸入的低級別特徵（例如某神經元會學到圖像中某些邊緣、角點），在更深的層次，神經元將前一層學習到的低級別特徵作為輸入，會學到更高級別、更抽象的特徵（例如某神經元會學到圖像中某些形狀、筆劃）。這句話我認為只對了一半，不對的地方是如同前面所說，人類其實無法賦予這些線性組合任何意義，對的地方是神經網路學習的趨勢的確是這個樣子，決策邊界的確是從原本雜亂、零碎的線性組合的排列，隨著層的推移而變得明確。所以我們能認為每個神經元分別會學到不同的東西（因為神經元的參數必定不同，所以能讓它們對應到原始輸入的「隱藏特徵」，可能是人類注意到的，也可能是沒有注意到的。那麼我們能賦予神經元的意義就是是否滿足這些隱藏特徵，及滿足的程度），但沒有人能斷定到底是學了什麼。這就是為什麼神經網路被稱為“black box”，這就是為什麼很多研究人工智慧的工程師也會自稱他們不知道機器是怎麼學習的，因為它的學習並非是一個簡單的、人類可以輕易賦予語義標籤的過程，而是一個機器自定義、逐步抽象的過程。雖然學習的趨勢是從低級別到高級別，但中間層的表示往往是機器獨有的語言，人類難以理解。所謂的表徵學習，可以被理解成是這個抽象化的學習過程。