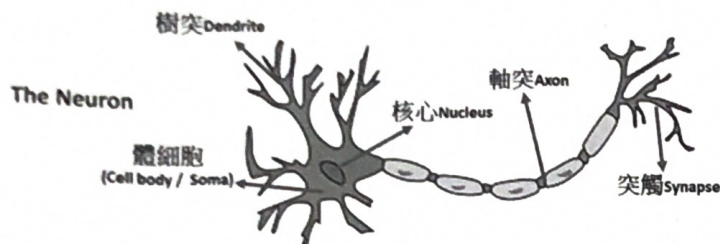




Ch1後：實作類神經網路計算神秘函數

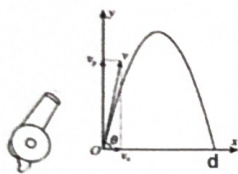
Shun-Shii Lin

Department of Computer Science & Information Engineering
National Taiwan Normal University

Acknowledgements: This presentation is created based on the lecture slides from *The Artificial Intelligence: A Modern Approach* by Russell & Norvig, a PowerPoint version by Professor Berlin Chen, and various materials from the web.

Training a Network = Minimize the Cost Function

例：拋體運動



	速度v	仰角θ	射程d
Training	80	45	640
	80	85	111
Testing	96	70	?

神秘的函數？ Training
公式為何？
實驗值？
猜測值？

	Input	Output
Example 1	0 2 0	0
Example 2	0 0 1	0
Example 3	0 1 1	0
Example 4	1 0 1	1
Example 5	1 1 1	1
Unknown	1 0 0	?

```

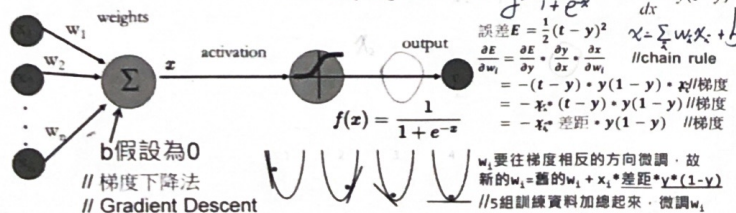
2 from numpy import * #載入函式庫
3 train_in=array([[0,2,0],[0,0,1],[0,1,1],[1,0,1],[1,1,1]])
4 train_sol=array([[0,0,0,1,1]]).T #轉置
5 #train_in= [[0,2,0], [0,0,1], [0,1,1], [1,0,1], [1,1,1]]
#train_sol= [[0], [0], [0], [1], [1]]
6 # initialize nn_weights
7 random.seed(1) #亂數種子固定，每次結果一樣，便於偵錯。
8 nn_weights = 2 * random.random((3,1)) - 1 #產生3*1亂數矩陣。
9 # nn_weight= [[-0.16595599], [0.44064899], [-0.99977125]]

```

```

20 # Run the NN adjustments
21 nn_weights+=dot(train_in.T,(train_sol-train_out)*train_out*(1-train_out))
22 # nn_weight= [[-0.16595599], [0.44064899], [-0.99977125]]
train_in= [[0,2,0], [0,0,1], [0,1,1], [1,0,1], [1,1,1]]
train_sol= [[0], [0], [0], [1], [1]]
更新的nn_weight= [[-0.12025406], [0.21166606], [-0.85063774]]

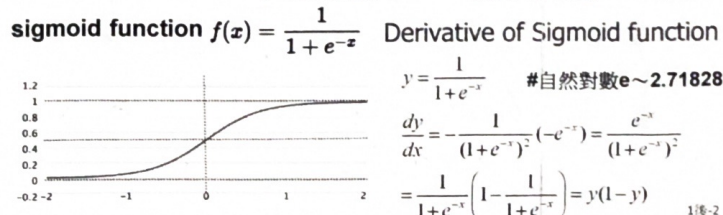
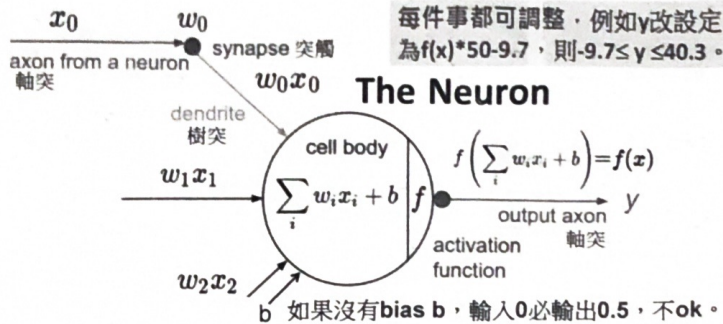
```



```

23 #unknown test input
24 test_in = array([1, 0, 0])
25 # Print the result for our unknown test input
26 print('\nThe final prediction is ', 1 / (1 + exp(-(dot(test_in, nn_weights)))))

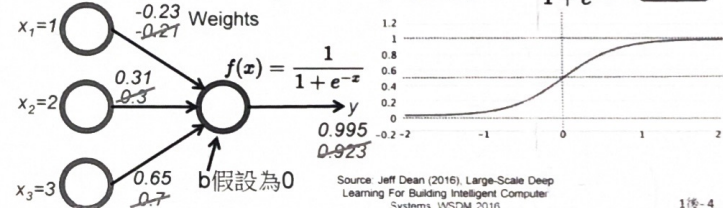
```



Learning Algorithm

While not done:
1. Pick a training example "(input, label)"
2. Forward : Run neural network on "input"
3. Backward : Adjust weights to let output closer to "label"

1. Pick a (input, label)=((x_1, x_2, x_3), 1)=((1,2,3), 1)
2. $x = -0.21x_1 + 0.3x_2 + 0.7x_3 = 2.49$; $y = f(x) = \frac{1}{1+e^{-x}} = 0.923$
3. Adjust weights to -0.23, 0.31 and 0.65. Next time:
 $x = -0.23x_1 + 0.31x_2 + 0.65x_3 = 5.34$; $y = \frac{1}{1+e^{-x}} = 0.995$



```

10 # train the network
11 for i in range(200):
12     print("\n i = ", i, "nn_weight=")
13     print(nn_weights)
14
15 # Calculate the outputs for each training examples
16 train_out = 1 / (1 + exp(-(dot(train_in, nn_weights))))
17 print("train_out = ")
18 print(train_out)
19
# train_in= [[0,2,0], [0,0,1], [0,1,1], [1,0,1], [1,1,1]]
nn_weight= [[-0.16595599], [0.44064899], [-0.99977125]]
= [[0.88129797], [-0.99977125], [-0.55912226], [-1.16572724], [-0.72507825]]
train_out= [[0.70709112], [0.2689864], [0.36375058], [0.23762817], [0.3262757]]

```

補充說明

這個程式只用一個神經元，且簡化了一些功能，因此需修改成適用的數字範圍或用不同的activation function才能稍微適用在稍稍複雜的應用。

```

# train_in= [[6, 20, 65], ...]
# train_sol= [[91], ...]
# nn_weight= [[-0.16595599], [0.44064899], [-0.99977125]]
train_out= [[1.], ...]

```

不管多少回合均會固定輸出1

原因是train_in的元素值如[6,20,65]太大，乘上nn_weight值為 $6 * (-0.16595599) + 20 * (0.44064899) + 65 * (-0.99977125) = -57.16788739$ ，值很容易變太小(或太大)。而加上bias 0.5，使得再經sigmoid $f(-57.16788739 + 0.5) = 2.4515977009438106e-25$ 是一個極小的實數。而即使將這麼小的數乘上3.3再加上1，也不對，最後值仍然一定是1。

因此用這個簡化的一個神經元是不一定能訓練有成的，而且也要特別注意數字的允許範圍，有時會僵在一個固定的數字，動彈不得。

當然有一些解決的方法，如將輸入或輸出或中間的一些位置的數字，或亂數做數字範圍的調整，如 $[6, 20, 65] * 0.01 = [0.06, 0.2, 0.65]$ 。或可疏解一些問題，但這就是要同學們try的地方了。

作業一：實作類神經網路計算神秘函數

首先，你要設法啟動python，怎麼啟動？接著找一個真實應用的data set，擺幾個神經元，架構一個小小的類神經網路，初始化權重，開始訓練一部份的training data，經一段時間，訓練有成，可以用來預測未知的新資料。這時可告訴你自己，到底真的如大咖們說的那麼好嗎？你如何下結論？接著你想自己提昇一下，能做到不斷進展、讓人驚嘆的結果嗎？

相關資訊可參考報章雜誌，或到網路上探索。本作業將期望你好好了解及體驗這個NN的威力，並表達你對一些相關課題的心得。

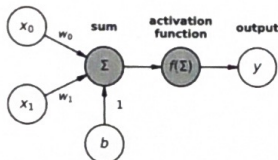
作業一：實作類神經網路計算神秘函數

作業報告 due：2023-03-14

(1) 報告應使用你所學習了解後的詮釋，而不是照抄網站或老師講義上的中、英文句子。請將作業報告上傳到師大 Moodle 平台 (<http://moodle3.ntnu.edu.tw/>)。請務必確認系統已收到你的資料。

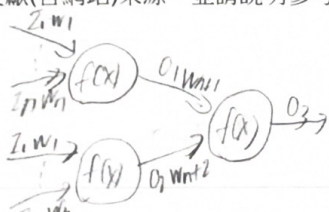
(2) 作業報告請依序詳細說明下列資訊：

- 首先，你要說明你針對這個成績計算的Data set，輸入是甚麼？有幾個？輸出是甚麼？有幾個？你想用多少組當訓練資料？用多少組當測試資料？(你的老闆也沒用過，所以要說明一下。) 注意：請詳細說明你所使用之機器軟體規格、所用的作業系統等相關資訊以及你為何選擇這樣的規格。另外請提供你的連絡電話，以便不時之需。
- 請先用一個神經元(activation function)，寫python程式，用來訓練你這個資料集，看花了不同的時間：訓練資料的mean square error結果為何？測試資料的mean square error結果為何？也請說明如何執行你的程式。
- 請使用不同的activation functions來做上一項任務，同樣地做說明，並比較有沒有比較好的activation functions？



作業一：實作類神經網路計算神秘函數

- 自由申論及發揮：ChatGPT這個聊天機器人程式，請你試著用看看。請記錄問答記錄。A. 你問它十個問題，它回答讓你滿意的有多少題？B. 你有辦法讓它都答不出正確的答案嗎？C. 你問它最近的新聞事件，它會亂答嗎？D. 有人說可用它來寫作業，請你將最近老師們出的作業給它做做看，有讓你驚嘆嗎？E. 有人說可用它來算數學題目，它能解決多困難的數學題目？
- 請說明你做此作業所碰到的一些狀況及困難。
- 請列出你的參考文獻(含網站)來源，並請說明參考了那些部份用於作業中。



- 額外加分：如果你能找一些其它的資料集，做一個NN，訓練有成，也可獲得額外的加分。範例如下：

物理：萬有引力、水平拋射、斜向拋射、

阿特午機、克卜勒行星三大運動定律、...

面積：三角形、長方形、正方形、平行四邊形、

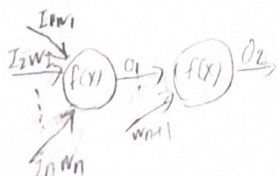
梯形、菱形、圓、扇形、橢圓形、...

體積：圓柱體、...

排列組合：排列數、組合數、...

日期及時間轉換：國曆／西曆

財務：利息計算、...



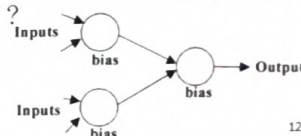
作業一：實作類神經網路計算神秘函數

應用範例：成績計算

抽問分數	作業一	作業二	作業三	作業四	作業平均	測驗一	測驗二	測驗三	測驗平均	總成績
11	86	90	93.5	93	90.63	90	99	106	98.33	94
11	85.5	70	86.5	88.5	82.63	91	88	90	89.67	89
10	77	63.5	58.5	78	69.25	82	73	65	73.33	79
8	82.5	75.5	87.5	82.5	82.00	75	99	91	88.33	85
11	91	89	88.5	89.75	89.56	105	97	114	105.33	99
11	94.5	89.5	95.5	90.5	92.50	93	116	112	107.00	96
6	78.25	63.5	62.5	67.25	67.88	94	66	77	79.00	75
5	86.5	90.5	82.5	0	64.88	85	94	90	89.67	78
11	85	62	63	84.5	73.63	96	95	74	88.33	85
10	89	77.5	90	90.5	86.75	95	105	97	99.00	91
8	91.25	90.5	88.5	0	67.56	83	83	60	75.33	76
7	70.5	61.5	59	52	60.75	98	95	38	77.00	73
7	84	0	0	0	21.00	65	0	0	21.67	44
11	64	9	0	0	18.25	85	48	71	68.00	57

作業一：實作類神經網路計算神秘函數

- 請設法用兩個神經元(如講義上的，有bias，有activation function)，來建構類神經網路。請推導backpropagation公式出來。
- 將上一項結論，寫出python程式，來訓練你這個資料集，看花了不同的時間：訓練資料的mean square error結果為何？測試資料的mean square error結果為何？
- 將上一項結論，使用不同的activation functions，寫出python程式來做上一項任務，同樣地做說明，並比較有沒有比較好的activation functions？
- 請設法用三個神經元(如講義上的，有bias，有activation function)，來建構類神經網路。請推導backpropagation公式出來。
- 將上一項結論，寫出python程式，來訓練你這個資料集，看花了不同的時間：訓練資料的mean square error結果為何？測試資料的mean square error結果為何？



作業一：實作類神經網路計算神秘函數

本課程允許遲交作業(三天內)，遲交的話，一天內約扣2分，二天內約扣4分，三天內約扣6分。

✓ 提醒：報告應親自為之，請述明你所參考之資料的來源。報告如有抄襲同學之狀況(不管被抄襲或抄襲別人)，一律依校規送校方處理。請你也要保護好你的智財權，別讓別人不勞而獲。因此作業請勿抄襲同學的作業，而說是自己所撰寫的。根據以往的經驗，非常容易被發現。(因為修課人數不多，找出抄襲的time complexity ↓)

✓ 作業報告著重於解決這個問題的構想、理由，還有實驗設計以及結果說明、分析的完整性，即使實驗數據不好看、測試效能不佳，只要有完整的結果分析、或是說明有獨到的見解都能拿高分。換句話說，報告的評分重點在你對於作業的用心程度，自己構思所佔的部份是否足夠，而不僅僅是測試效能、實驗數據的美觀。

Recipe of Deep Learning

