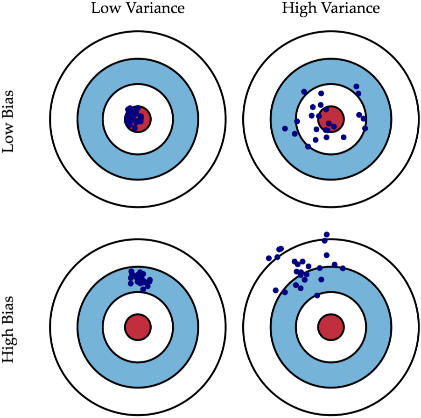
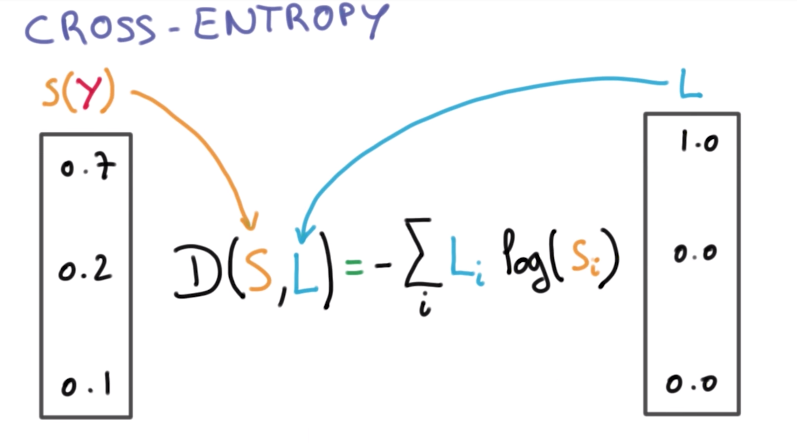
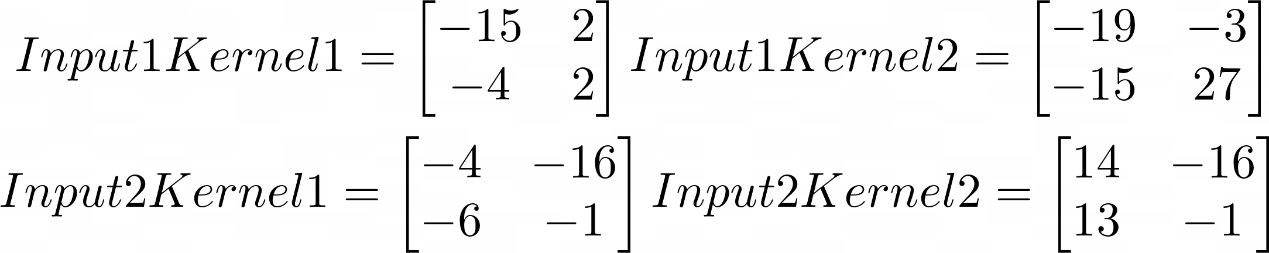
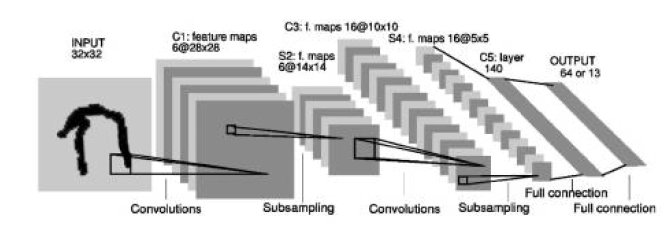
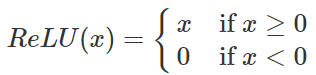
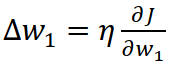
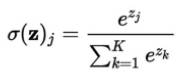
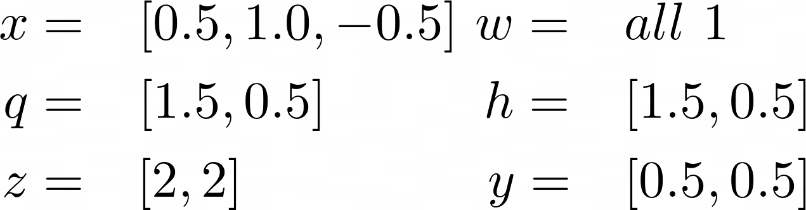
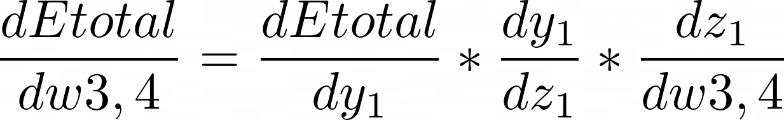
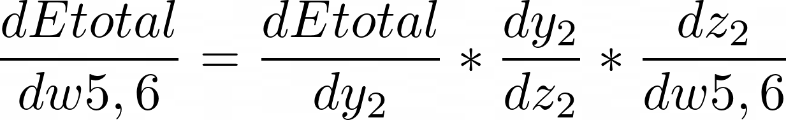
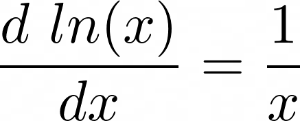
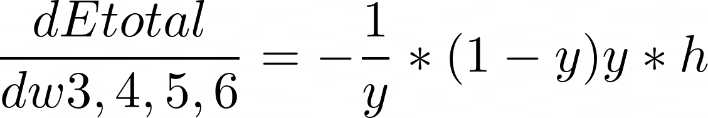
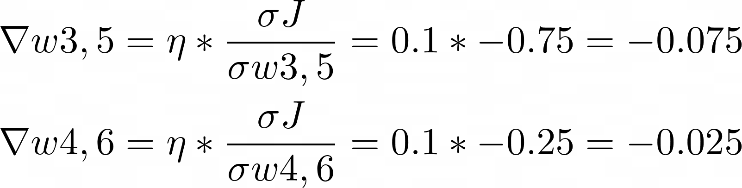
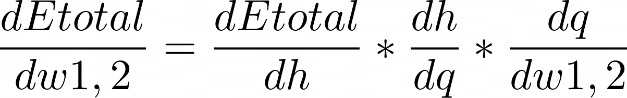
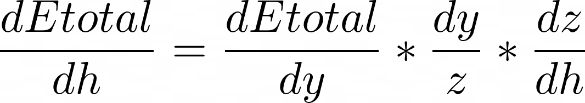
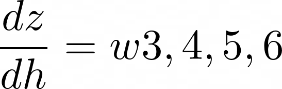
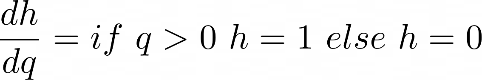
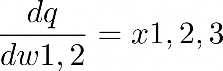
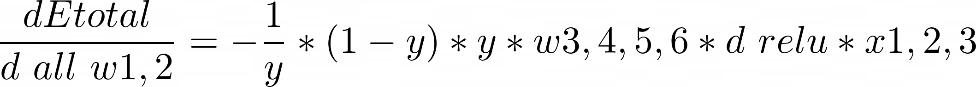
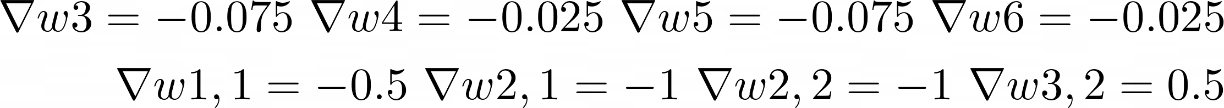
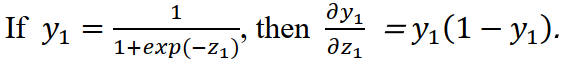
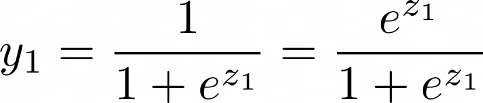
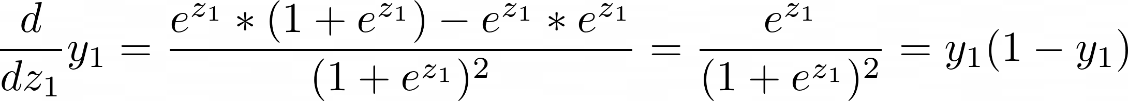
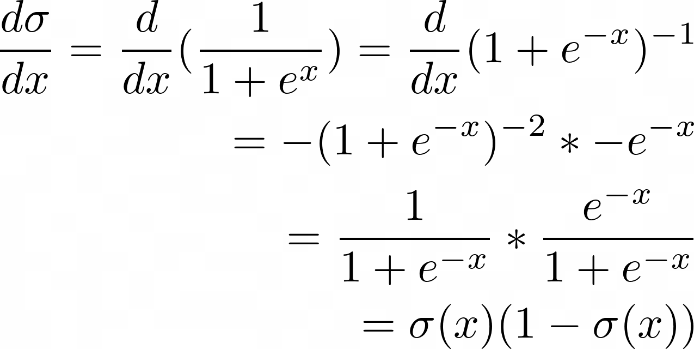
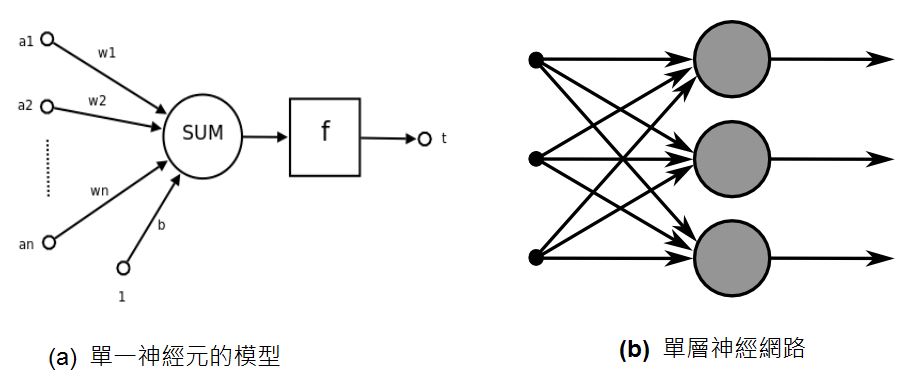
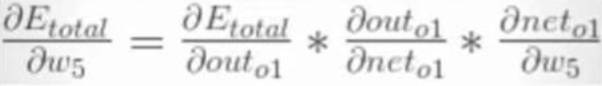
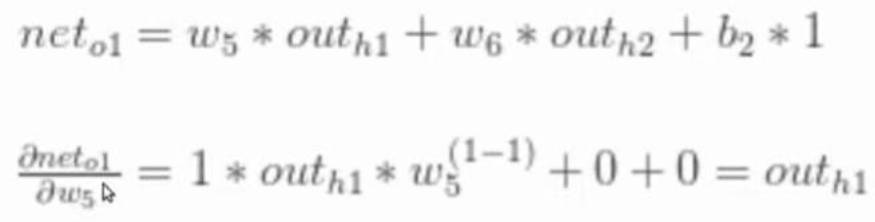
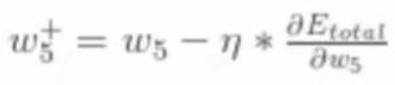
HW 6&7  
Student ID:109598001  
Student Name: 鄭濠營

1. 針對神經網路的高bias偏差，來選擇微調整wx+b的b參數，因為wx參數是調整斜率，又或者說角度；而b參數可以調整線性分類器的移動方向及距離，故可以推測high bias的原因是否為線性分類器的所屬位置不夠精確。  
   神經元多與少表示線性分類器的數量與精確度，故透過先減少神經元數目，針對train與test/validation的accuracy來調整bias，之後再漸漸增加神經元數目來提高精確度，理論上模型越複雜其精確度越準，但要注意過擬合(overfitting)的問題。  
   針對神經網路的高variance偏差，來選擇微調整wx+b的wx參數，如上面所提bias敘述一致，故可判斷high variance原因是否是線性分類器過度複雜導致過擬合(overfitting)的狀況。  
   神經元數量及層數若多代表模型變得更複雜，過度複雜的模型由於過於精確度的影響下，容易導致過擬合(overfitting)的狀況。所以建議先減少神經元的數目，來看train與test/validation的accuracy的狀況進而調整參數。模型複雜雖精確能夠上升，但也須考量到適當平衡。  
   根據下圖可明顯high bias與high variance對照的訓練狀況:  
     
   圖片參考: <https://jason-chen-1992.weebly.com/home/-bias-variance-tradeoff>
2. 若使用multilayer neural network for regression, 可以使用cross entropy 作為 lost function. Entropy 的概念可以以Decision Tree 的實例, 其為取得某特徵的資訊量是否夠大，又換句話說此特徵是否夠明顯能辨識某種類別，很適合用於處理分類問題。下圖是 cross entropy 的公式:  
     
   Cross entropy 的用意是為了觀察預測label機率與實際label機率的分布，其兩者誤差範圍。得到最終誤差範圍，其算出的entropy，用來 back propagation 來調整整個神經網路的權重參數。最終誤差範圍會漸漸收斂，得到相較好的模型。  
   可參考: <https://hengxiuxu.blogspot.com/2019/11/cross-entropy.html>
3. Pre-condition: stride of 2, no zero-padding, using the ReLU activation function.  
   先做卷機 (convolution)，以矩陣來表示:  
     
   本題未討論池化層 (pool layer) 的設定，暫不納入考量。  
   接著到全連接層(Fully Connected Layer)的計算，一維矩陣表示:  
   [-15, 2, -4, 2, -19, -3, -15, 27, -4, -16, -6, -1, 14, -16, 13, -1]   
   接著以MLP的方式進行計算 (權重隨機初設…都設成1吧):  
   [-15, 2, -4, 2, -19, -3, -15, 27, -4, -16, -6, -1, 14, -16, 13, -1] 全加起來等於 -42  
   經過Relu計算，非0保持原樣，<0則設定為0:  
   Relu(-42) = 0。故最終結果為機率0 (因Relu基本是只有0與其餘正數)。
4. 下圖為題目所提供的LeNet-5 模型架構。訓練的權重位在Full Connection的部分，故需計算Full Connection間相對應的數量關係:  
     
   可見圖表上S4的subsampling階段為16個5\*5的Feature maps，則將其做Flatten產生400個神經元，C5對應140個神經元，Output借題目定義為10個神經元。故在這些Full Connection間的權重(weight)數量做計算，總計為400\*140\*10 =960000個weight變數；三層的Full Connections，S4用Flatten成1維資料作為神經元的input layer🡪C5作為hidden layer🡪Ouput就是輸出層。
5. The activation function from 𝑞1 to ℎ1 and 𝑞2 to ℎ2 is ReLU, the outputs 𝑦1 and 𝑦2 are softmax ouput, and the cost function is 𝐽 = − log 𝑦1. Let 𝑤1 to 𝑤6 be 1.0, 𝑥1 = 0.5, 𝑥2 = 1.0, 𝑥3 = -0.5, and 𝜂 = 0.1.  
   Relu 公式: 權重更新公式:  
      
   Softmax:  
     
   Forward Propagation:  
     
   Backward Propagation:  
     
     
   假設log底數為2及cost function微分: Softmax微分:  
      
   Back Propagation更新權重函數的其一參數:  
     
   -(1/0.5)\*(1-0.5)\*0.5\*1.5 = -0.75 & -(1/0.5)\*(1-0.5)\*0.5\*0.5 = -0.25, 兩者皆乘上learning rate.  
     
   計算w1,1; w2,1;w2,2;w3,2:  
      
    dh/dq為relu微分:   
      
   w(1,1)的代表意義w(x\_num, q\_num)，意旨從x1🡪q1:  
   w1,1🡪 (-(1/0.5)\*(1-0.5)\*0.5\*1-(1/0.5)\*(1-0.5)\*0.5\*1)\*1\*0.5 = -0.5  
   w2,1 & w2,2🡪 (-(1/0.5)\*(1-0.5)\*0.5\*1-(1/0.5)\*(1-0.5)\*0.5\*1)\*1\*1 = -1  
   w3,2 🡪 (-(1/0.5)\*(1-0.5)\*0.5\*1-(1/0.5)\*(1-0.5)\*0.5\*1)\*1\*-0.5 = 0.5  
   以下為整理的計算結果:  
     
   ---
6. , 以下是sigmod微分計算過程:  
     
     
   可參考的sigmod微分過程: <http://scwnote.blogspot.com/2017/07/sigmoid-function.html>  
   
7. 神經網路的計算公式:  
     
   圖片參考: <http://programmermagazine.github.io/201404/htm/focus1.html>  
   若不考慮所謂的bias參數，則weight=0會造成back propagation執行毫無效果。因通過forward propagation計算，最終output即為0。  
   觀察back propagation的其一微分過程及梯度運算:  
     
   若weight皆為0的情境下，更新結果也是為0，則毫無效果。使用Relu作為activative function，其負數歸為0，正數保持原樣的做法，但因為back propagation是主要職責是做weight更新，而過程微分的結果是0，所以無法更新。可觀察下圖的weight更新function:  
     
   參考url: <https://www.youtube.com/watch?v=GJXKOrqZauk>
8. Solver參數設定為’liblinear’,’ newton-cg’，皆不會產生無法convergence problem。請參考檔案”Question8.pdf and Question8.ipynb”.
9. 請參考檔案”Question9.pdf and Question9.ipynb”.
10. 請參考檔案”Question10.pdf and Question10.ipynb”.