

Computer Vision #HW3 Report

108061613 黃雅琳

(Describes the difficulties you have encountered.)

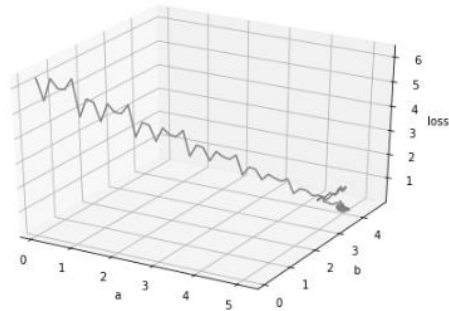
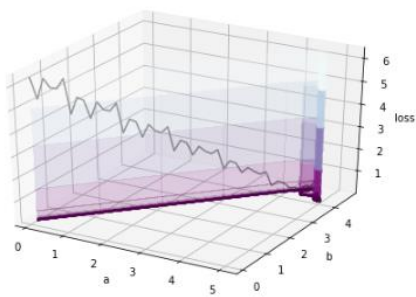
Problem A:

Part A1:

我的 Gradient Descent Plot 在畫 surface 時遇到角度轉了 90 度的問題(左圖)，始終無法解決，但單看 loss 3D plot(右圖)確實與助教的範例相近。

試過 meshgrid 也還是一樣的結果(左圖)

```
# plot Gradient Descent
def plotGD(best_history):
    # line
    A = best_history['a']
    B = best_history['b']
    LOSS = best_history['loss']
    # Y_pred = best_history['pred']
    # Y_true = best_history['gnd']
    # xs = best_history['x']
    # contour
    X,Y,Z = [],[],[]
    for i in range(40):
        X.append(A[10*i])
        Y.append(B[10*i])
        Z.append(LOSS[10*i])
    X = np.array(X)
    Y = np.array(Y)
    Z = np.array(Z).reshape(40,1)
    fig = plt.figure()
    ax = Axes3D(fig)
    ax.plot_surface(X, Y, Z, linewidth=0, antialiased=False, alpha=0.2, cmap=plt.get_cmap('BuPu_r'))
    ax.plot3D(A, B, LOSS, 'gray')
    ax.set_xlabel('a')
    ax.set_ylabel('b')
    ax.set_zlabel('loss')
```

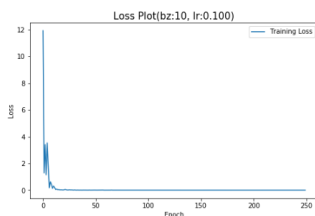


Part A2:

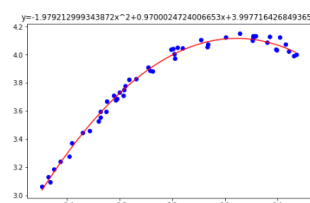
於 code 中，我使用了(1) nn.linear() 和(2) nn.paramter 作為 model 的方式，兩種方式都有很好的收斂跟 fitting 結果

(1)nn.linear(): $y = -1.97921 X^2 + 0.9770 X + 3.9977$

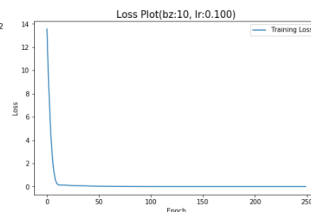
(2)nn.parameter(): $y = -1.9953 X^2 + 0.9770 X + 4.0013$



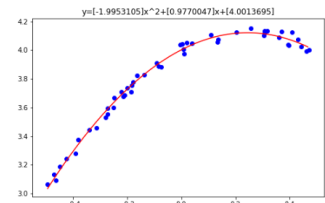
(1)Loss Plot



(1)Fitting Result



(2)Loss Plot



(2)Fitting Result

Problem B:

1. 在此題，我一開始遇到的困難是將使用 `cv2.imread(image)` 的 `image size: (H,W,3)`，轉為 LTI 的 `(3,H,W)` 這件事情上面，找了很久有沒有相關的轉換函數後，才發現可以用很簡單的方式做到這件事。

```
def changeDim(old):  
    '[H,W,3] -> [3,H,W]'  
    a,b,c = old[:, :, 0], old[:, :, 1], old[:, :, 2]  
    return np.array([a,b,c])
```

2. 接下來，我所遇到的難題主要是在 colab 上運行時，會不停斷掉需要重跑，故我的 epoch 不能太大，故在 parameter 和 loss function 的選取上就試了很多種不同組合。

Batch_size = 32; learning rate = 1e-3, 7e-4, 1e-4, 1e-5,

後來選擇 batch size = 32; learning rate = 1e-3

3. 在我的 class Test() 中，有一個 function 是 `def resize()`，在運行時，一直在此出現 error，經過 debug，才發現是 class 裡面的 function 一定要有 "self"。

```
def resize(self, ps_batch, HH, WW):  
    PS = []  
    ps_batch = ps_batch.detach().cpu().numpy()  
    for ps, H, W in zip(ps_batch, HH, WW):  
        H, W = H.detach().cpu().numpy(), W.detach().cpu().numpy()  
        H, W = 1160, 720  
        BR_x, BR_y, BL_x, BL_y, TL_x, TL_y, TR_x, TR_y = ps  
        PS.append([BR_x*W, BR_y*H, BL_x*W, BL_y*H, TL_x*W, TL_y*H, TR_x*W, TR_y*H])  
    return PS
```

一開始我沒加 self，故會一直出現輸入數量對不上的問題。

4. 在我的程式中，有設計希望能夠在 grid search 後，自動 test 最好的 model。在這部分，下了一點功夫。利用 loss 找到最小的

```
log_dir = Path('./runs/') / f'{datetime.now():%b.%d-%H_%M_%S}'  
log_dir.mkdir(parents=True)  
MAXEPOCH = 10  
BZ = [64]  
LR = [1e-3]  
NUMWORKER = 0  
minloss = 0  
for bz in BZ:  
    for lr1 in LR:  
        log_dir = Path('./runs/') / f'{datetime.now():%b.%d-%H_%M_%S}' / f'{-}{-}'.format(bz, lr1)  
        log_dir.mkdir(parents=True)  
        train = Trainer(log_dir, bz, lr1, MAXEPOCH, NUMWORKER)  
        train.run()  
        if train.minloss < minloss:  
            minloss = train.minloss  
            bestCOM = train.bestCOM  
            bestDir = train.bestdir  
print('best:', bestCOM)
```