CS655000 Computer Vision Homework 3

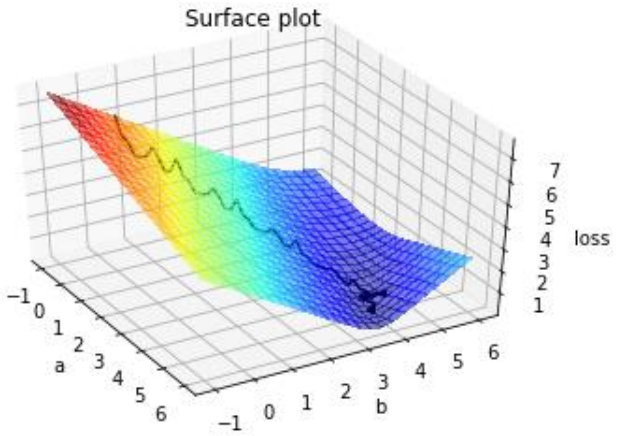
108062586 楊子儀

Part 1. Line Fitting

從這題的練習，大致上能理解 perceptron，這個二元的線性分類器的運作原理。從 y = ax + b 到第二小題 y = ax^2+ bx + c，利用 pytorch 的 nn.module 設計了簡單的 neural network 模型做訓練，也對 pytorch 的使用比較熟悉。

在畫 3D 圖時，花了較多的時間。

(Ref: https://scipython.com/book/chapter-7-matplotlib/examples/simple-surface-plots/)



Part 2. License Plate Localization

a) 資料處理:

一開始在讀取 train data 的影像，是使用 skimage 的函式庫做讀取，將影像使用 numpy 的 array 處理，再將影像的 shape 轉換成 [3, H, W]。發現在 training 時，花費了大量的時間。

之後改用 PIL 讀取影像，array 的 shape 就直接為[3, H, W]的格式，減去了轉換的時間。另外，使用 torchvision 函式庫 transform 的方法，將 pixel 直接轉至 0 到 1 之間，同時也轉換成 pytorch tensor，加快了不少資料處理的時間。

```
31        img_path = str(self.img_dir) + "/" + str(ann['name'])
32        # print(img_path)
33        # img = skimage.io.imread(img_path)
34        img = Image.open(img_path)
35
36        # trans_img = np.array(img, dtype=np.uint8)
37        # trans_img = img.astype('float32') / 255.0
38
39        resize_img = img.resize((192, 320), Image.BILINEAR)
40
41        transform1 = transforms.Compose([
42            transforms.ToTensor(), # range [0, 255] -> [0.0,1.0]
43            ]
44        )
45
46        tensor_img = transform1(resize_img)
```

b) 將預測的點寫入 CSV，傳入 server 驗證

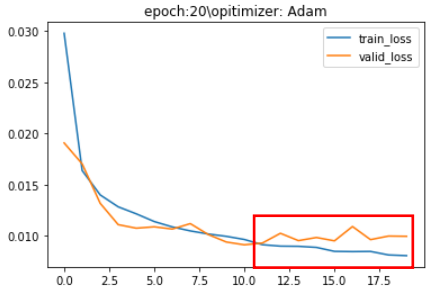
在寫入成 CSV 檔時，會同時產生序列的 index number，但因為與範例的格式不相符，會出現錯誤的訊息。

(Ref: https://blog.csdn.net/qq_38268886/article/details/80744721)

另外，在將訓練好的模型做 testing 的時候，沒開啟 self.model.eval()，導致每次將檔案上傳 server 驗證，error 都很大。

c) 貯存最佳 model 的 checkpoint

當 training 到一定程度時，validation 的 loss 反而會不降反升(可能是 overfitting 了)，所以需要保存 validation 最佳的 model，再做 testing，期望能得到最好的 testing 的結果。




```
    self.chkpt_dir = Path('./runs/model_'+ str(self.epoch) +'.pkl')
    torch.save(self.model.state_dict(), self.chkpt_dir)

    if self.epoch == 1:
        best_valid = valid_loss
    else:
        if valid_loss < best_valid:
            best_valid = valid_loss
            # print("------------")
            # print("best parameter on epoch:"+str(self.epoch))

            # print(best_valid)

            # self.chkpt_dir = Path('./runs/model_'+ self.epoch +'.pkl')
            # self.chkpt_dir = Path('./runs/model_bst.pkl')
            # torch.save(self.model.state_dict(), self.chkpt_dir)

    histor_valid.append(valid_loss)

print("************** history validation loss ****************")
print(histor_valid)

# m = max(histor_valid)
max_epoch = [i for i, v in enumerate(histor_valid) if v == best_valid]
print("------------")
print("best:"+str(best_valid))
print("------------")
print("best_training_epoch:"+str(max_epoch[0]))

best_chkpt_name = 'model_'+str(max_epoch[0])+'.pkl'
```