

姓名:黃宗德

系級:資工四

學號:406410091

Execution description: Please describe the detailed steps how to execute your codes.

輸入python train.py訓練資料, 訓練完後會產生2張圖, loss curve及wrmsecurve。

loss curve:包含train_loss及validation_loss兩條線。

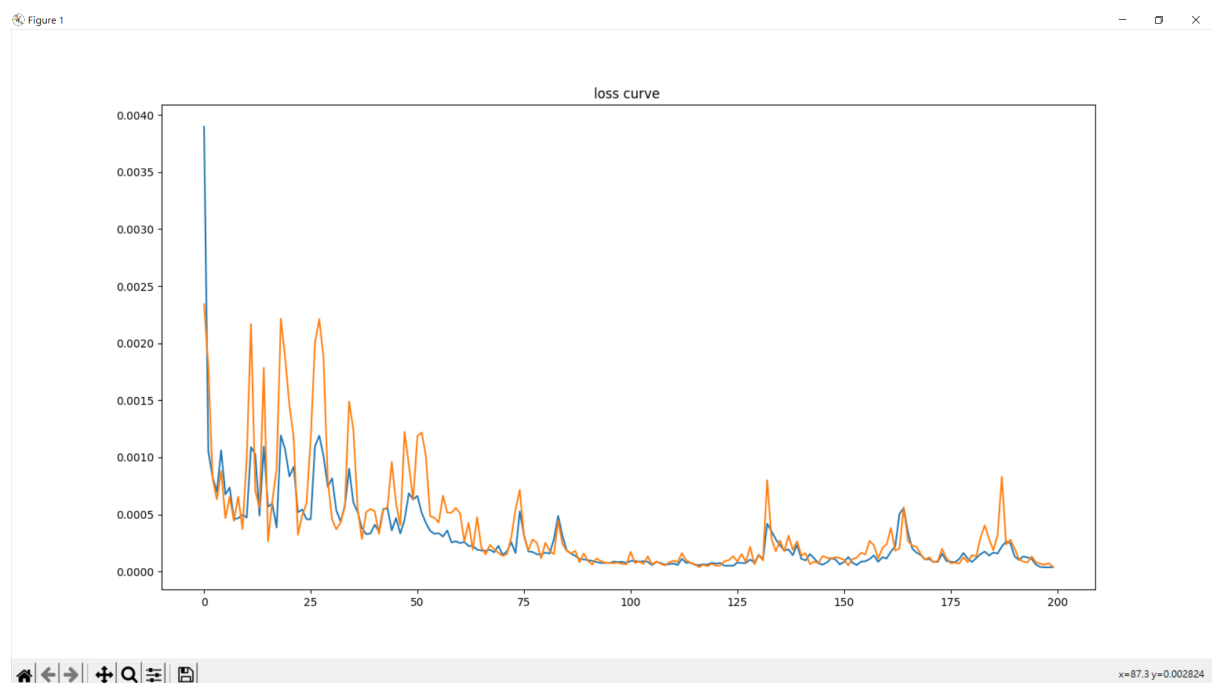
wrmse curve:包含Training WRMSE及validation WRMSE兩條線。

於訓練完資料後,

輸入python test.py產生result.csv。

Experimental results: As specified in the assignment.

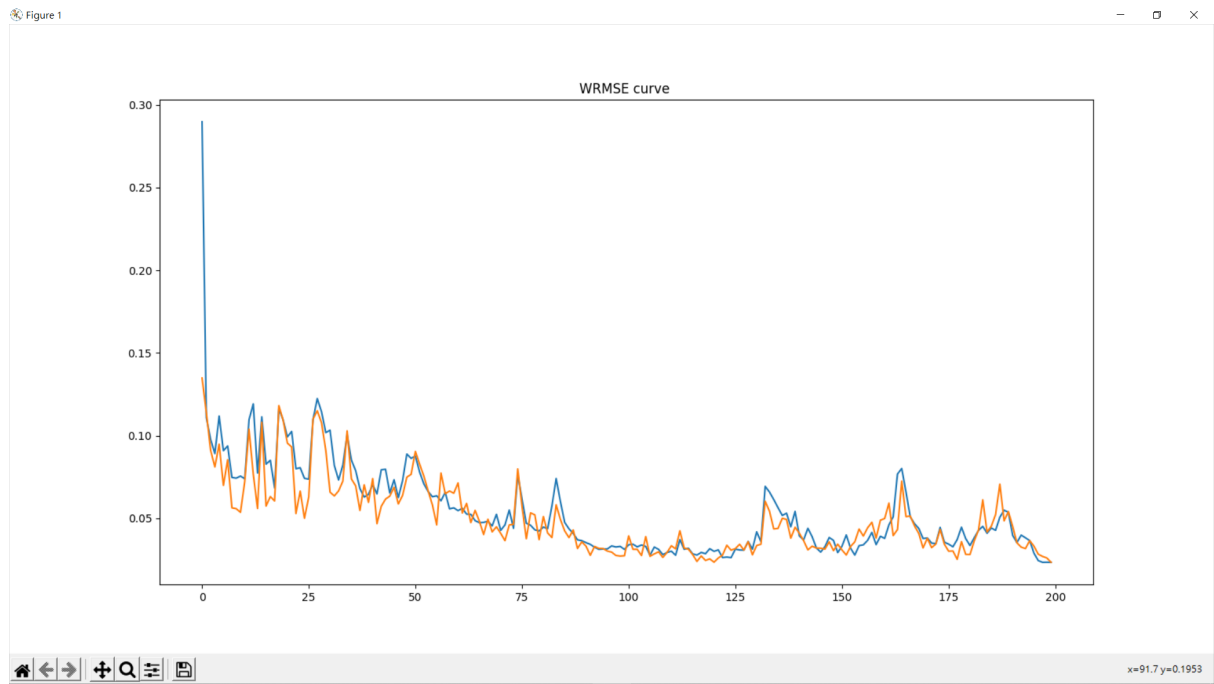
learning rate=0.002,epoch=200,batch_size=32



藍色是train_loss

橘色是validation_loss

(兩條線幾乎fit, train_loss大致低於validation_loss)



藍色是train_WRMSE

橘色是validation_WRMSE

(兩條線幾乎也幾乎fit, train_WRMSE也大致低於validation_WRMSE)

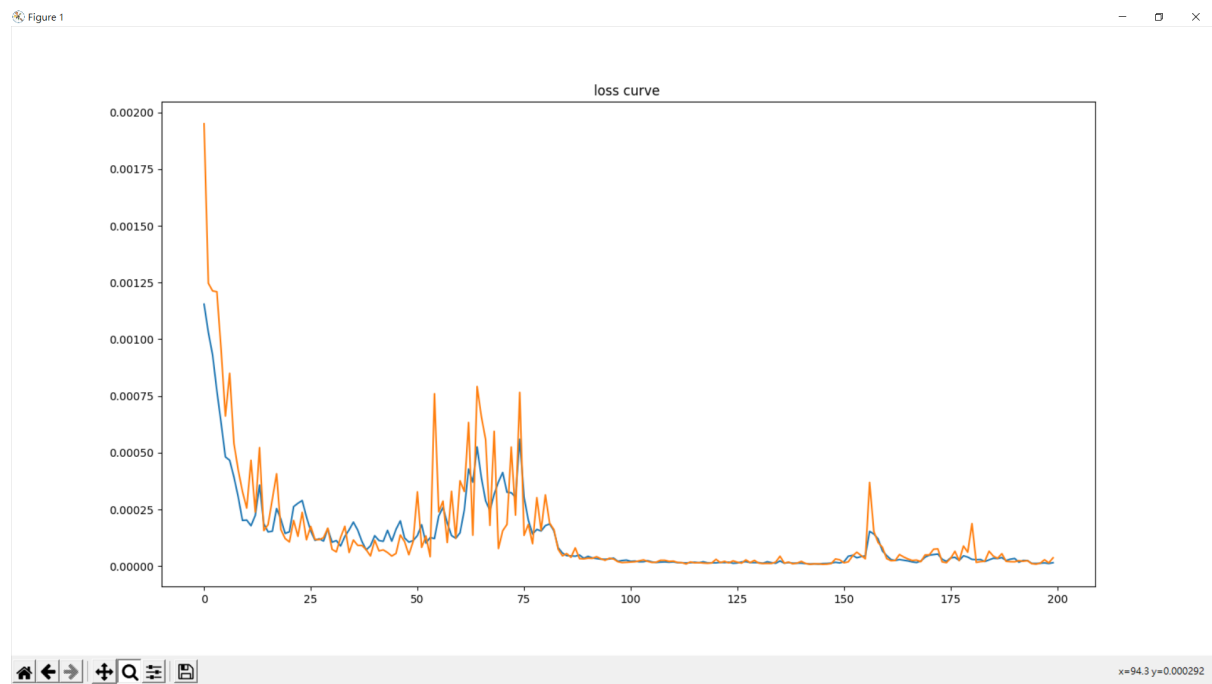
The screenshot shows a Visual Studio Code window with a terminal output for a training script. The terminal displays progress bars and metrics for epochs 199 and 200. For each epoch, it shows the completion status of training (10/10) and validation (2/2) data loaders, followed by the training loss, validation loss, training WRMSE, and validation WRMSE. The training WRMSE values are consistently slightly lower than the validation WRMSE values.

```
35 # learning rate, epoch and batch size. Can change the parameters here.
36
100%|████████████████████████████████████████████████████████████████████████████████| 10/10 [00:00<00:00, 60.25it/s]
100%|████████████████████████████████████████████████████████████████████████████████| 2/2 [00:00<00:00, 76.91it/s]
Training loss: 0.000037
validation loss: 0.000060
Training WRMSE: 0.023329
validation WRMSE: 0.026902

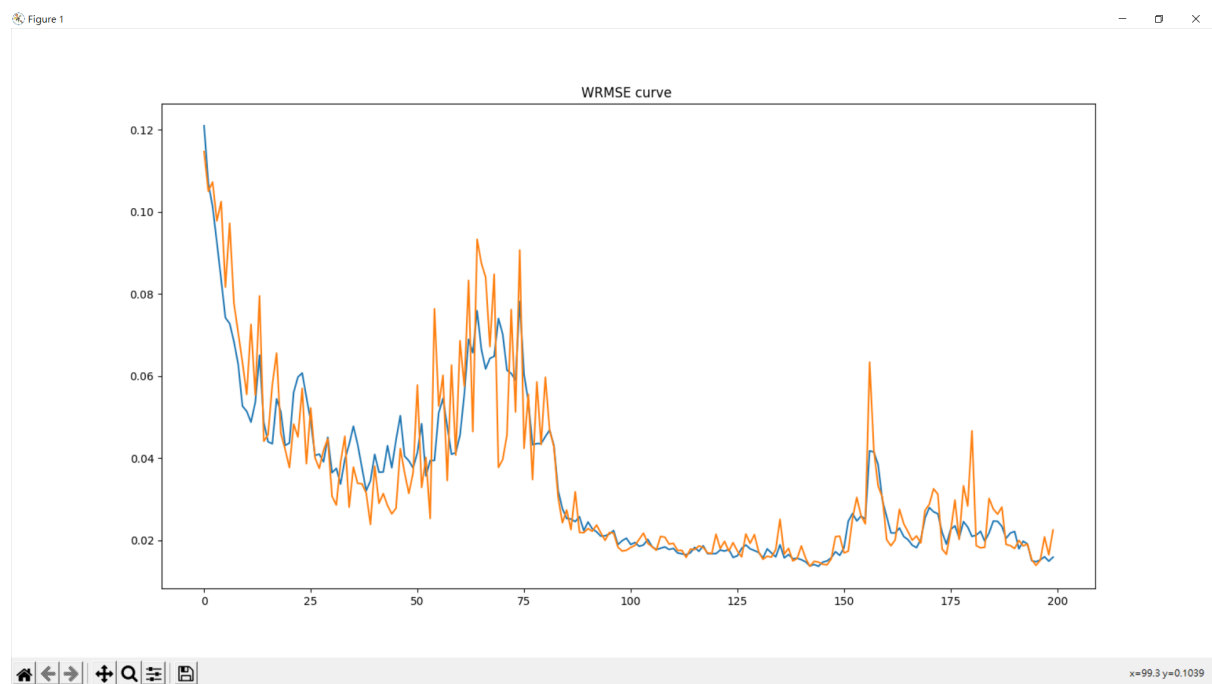
Epoch: 199/200
-----
100%|████████████████████████████████████████████████████████████████████████████████| 10/10 [00:00<00:00, 59.47it/s]
100%|████████████████████████████████████████████████████████████████████████████████| 2/2 [00:00<00:00, 64.69it/s]
Training loss: 0.000037
validation loss: 0.000073
Training WRMSE: 0.023370
validation WRMSE: 0.025964

Epoch: 200/200
-----
100%|████████████████████████████████████████████████████████████████████████████████| 10/10 [00:00<00:00, 55.87it/s]
100%|████████████████████████████████████████████████████████████████████████████████| 2/2 [00:00<00:00, 74.12it/s]
Training loss: 0.000038
validation loss: 0.000040
Training WRMSE: 0.023322
validation WRMSE: 0.023354
```

learning rate=0.002,epoch=200,batch_size=32



藍色是train_loss
橘色是validation_loss
(兩條線幾乎fit, train_loss大致低於validation_loss)



藍色是train_WRMSE
橘色是validation_WRMSE
(兩條線幾乎也幾乎fit, train_WRMSE也大致低於validation_WRMSE)

```

Validation WRMSE: 0.016491
Epoch: 200/200
-----
100%|████████████████████████████████████████████████████████████████████████████████| 10/10 [00:00<00:00, 41.65it/s]
100%|████████████████████████████████████████████████████████████████████████████████| 2/2 [00:00<00:00, 51.36it/s]
Training loss: 0.000016
validation loss: 0.000037
Training WRMSE: 0.015928
validation WRMSE: 0.022548

```

Conclusion: The observation from your results.

透過對資料的標準化後，各組統計數據區間會集中而變得平均，有助於對資料模型的訓練，讓訓練效果更加準確。

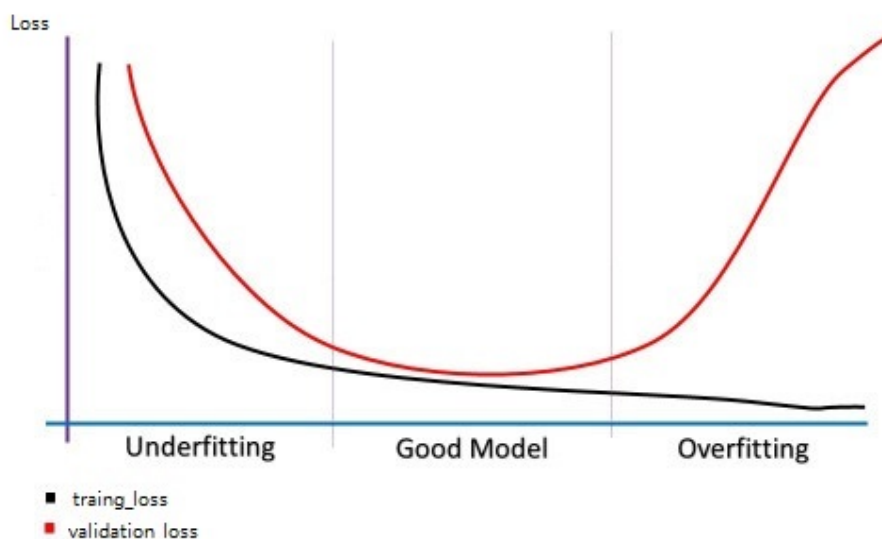
在有效的訓練模型下training loss理論上不會超過validation_loss。

又可分為underfitting、good fit及overfitting如下圖。

在good fit之前，若持續訓練traing data，可使得其曲線更加接近validation data的結果，此時的狀態稱為underfitting。

在good fit之後，若持續訓練traing data，會使得曲線遠離validation data的結果，這段狀態則稱為overfitting。

兩者曲線最接近的那段區間，稱為good fit。



Discussion: The questions or the difficulties you met during the implementation.

在計算train_loss, 及validation_loss時, 因為除的值有誤, 導致圖中的train_loss高於validation_loss, 原先除到整個資料的長度(trainload、validloader)而導致數值不對, 將除的底改成除資料集後(trainload.dataset及validloader.dataset), 各自的loss值即變成合理的樣貌。

在資料標準化時, 原先作法未設定區先範圍, 使得即便標準化後, 區間仍呈現較為分散的結果, 所以透過min_max_scaler = preprocessing.MinMaxScaler(), 來設定大小範圍, 再透過min_max_scaler.fit_transform()來做標準化, 將設定的範圍可以套用到資料集裡。