

Machine Learning HW2

tags: **Machine Learning**

tags: **Machine Learning**

[Files Description](#)

[Execution description](#)

[執行訓練](#)

[執行測試](#)

[Experimental results](#)

[模型架構](#)

[正規化](#)

[補值方法](#)

[訓練參數](#)

[訓練結果](#)

[Training Loss & WRMSE Curve](#)

[Conclusion](#)

[Discussion](#)

Files Description

- `train.py`、`train.csv`、`test.py`、`test.csv`、`models.py`、`dataset.py`
- `mymodel.pth`：模型訓練結果參數檔
- `loss.png`、`wrmse.png`：模型訓練曲線圖
- `result.csv`：測試結果
- `requirement.txt`：執行程式所需額外安裝之函式庫
- `hw2-report.pdf`：作業報告書

Execution description

執行訓練

```
$ python train.py
```

執行測試

```
$ python test.py
```

Experimental results

模型架構

Layer (type)	Output Shape	Param #
Linear-1	[-1, 278, 512]	114,688
LeakyReLU-2	[-1, 278, 512]	0
Dropout-3	[-1, 278, 512]	0
Linear-4	[-1, 278, 512]	262,656
LeakyReLU-5	[-1, 278, 512]	0
Dropout-6	[-1, 278, 512]	0
Linear-7	[-1, 278, 1024]	525,312
LeakyReLU-8	[-1, 278, 1024]	0
Dropout-9	[-1, 278, 1024]	0
Linear-10	[-1, 278, 20]	20,500
Total params: 923,156		
Trainable params: 923,156		
Non-trainable params: 0		
Input size (MB): 0.24		
Forward/backward pass size (MB): 13.07		
Params size (MB): 3.52		
Estimated Total Size (MB): 16.83		

正規化

使用 `sklearn.preprocessing.MinMaxScaler` 正規化到 **0-1** 之間

補值方法

用每個參數的中位數(median)當 NAN 的補值

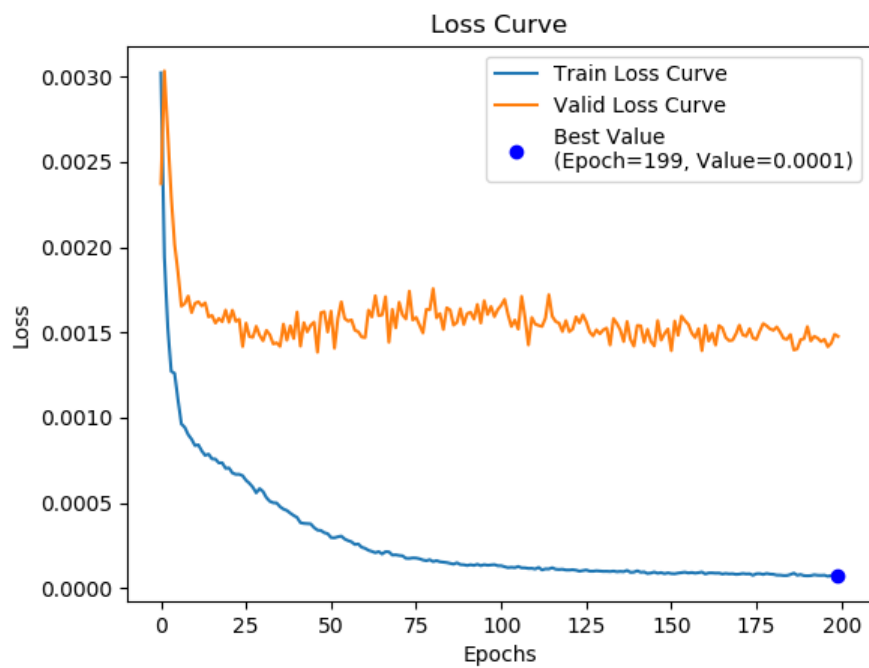
訓練參數

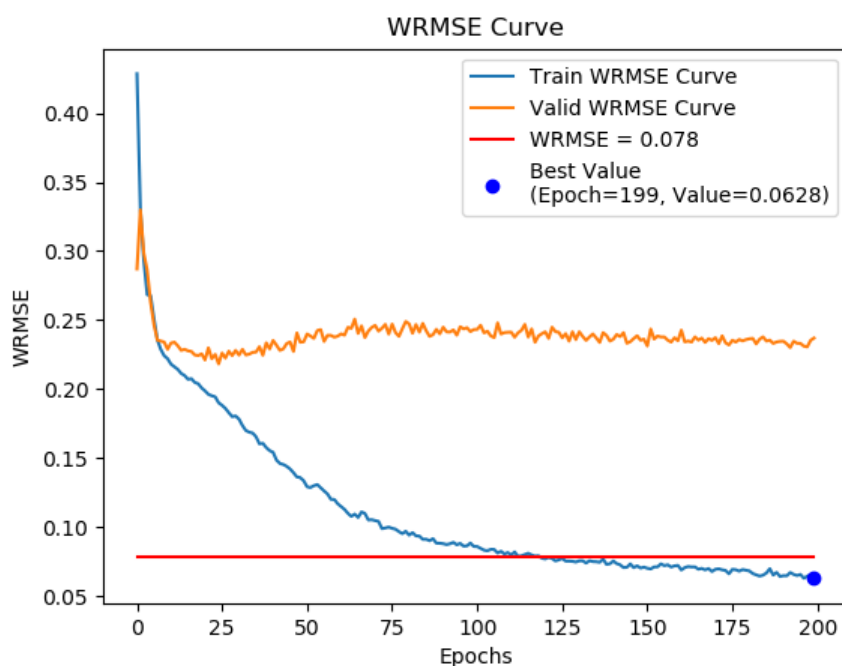
- learning rate: 0.001
- epoch: 200
- batch size: 64
- training dataset: valid dataset = 8: 2

訓練結果

- Best Epoch = 199
- Best Loss = 0.0001
- Best WRMSE = 0.0628

Training Loss & WRMSE Curve





Conclusion

- 在跑到約120回合時，就可以成功地把 WRMSE 降到 0.078 以下，且在200回合左右，保持持平。
- 但是 valid loss 在一開始大幅度下降後，就維持在一定區間，所以有 overfitting 的可能。

Discussion

- WRMSE 的大小跟 training dataset 的值域大小有關，如果把正規化範圍定為 0-0.5 之間，則 WRMSE 會大幅降低。所以題目要求的 $WRMSE < 0.078$ 應該是指測試結果，不是訓練的結果。但因沒有提供最終的正解，所以仍不能確定是否有在標準之內。
- 所有模型的 input 都會經過正規化，但是正規化範圍是自訂，所以照理說模型的 output 應該也會隨正規化範圍不同而值域不同。故在測試時，應讓測試的 output 反正規化，回復到應有的值域，但是因無法得知 target 的值域，所以無法反正規化。