

題目

使用神經網路同時近似 Runge 函數以及其一階導數。

模型架構

- 輸入層：x (1 維)
- 隱藏層：2 層，每層包含 64 個神經元，激活函數採用 tanh
- 輸出層：1 層

訓練架構

- **Loss function**：MSELoss（包含 function loss 與 derivative loss）
- **Optimizer**：Adam，學習率 10^{-3}
- **學習率調整**：ReduceLROnPlateau (factor = 0.5, patience = 50)
- **Early stopping**：若驗證誤差連續 150 epoch 無改善則提前停止

資料集

- 訓練集：200 點
- 驗證集：50 點
- 測試集：1000 點，取自區間 $[-1,1]$

結果 1

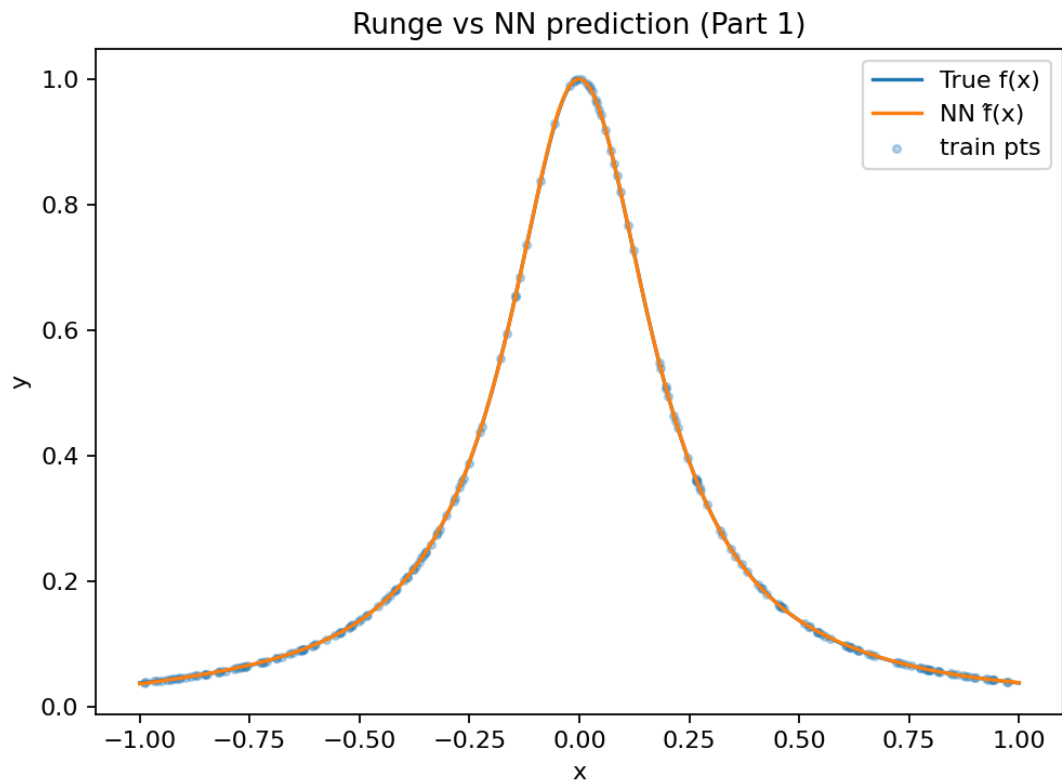
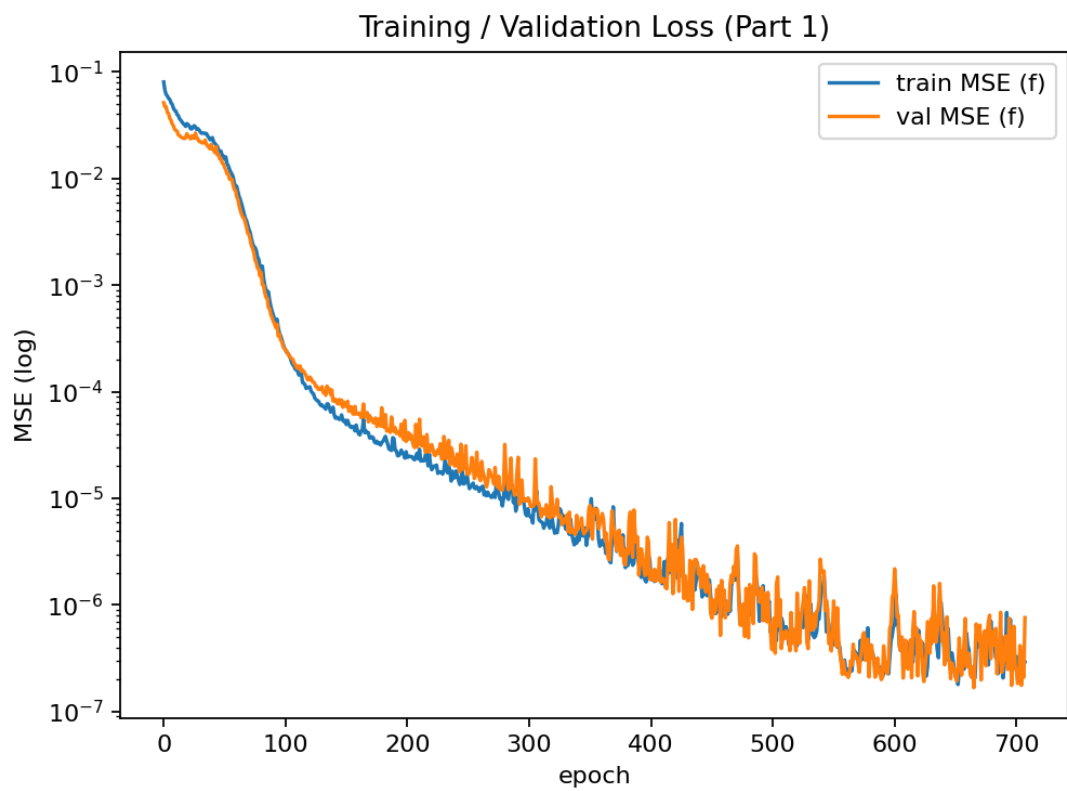
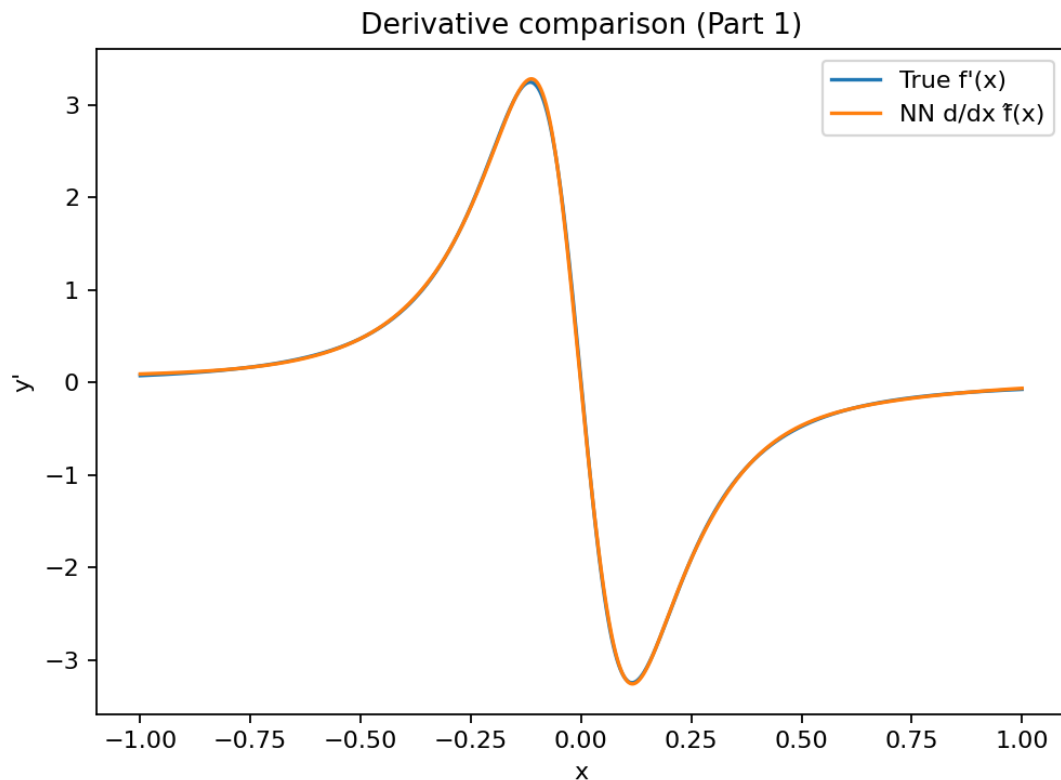


Fig2: the derivative comparison (此處插入 prediction_fprime.png 圖)





測試集表現

```
Epoch 100 | train 2.601e-04 | val 2.543e-04
Epoch 200 | train 2.739e-05 | val 3.423e-05
Epoch 300 | train 7.976e-06 | val 1.010e-05
Epoch 400 | train 2.657e-06 | val 1.739e-06
Epoch 500 | train 5.806e-07 | val 3.859e-07
Epoch 600 | train 6.404e-07 | val 1.283e-06
Epoch 700 | train 2.914e-07 | val 2.546e-07
Early stopping @ 708 | best val MSE=2.243112e-07
[Test] f(x):      MSE=3.113189e-07 | Max|err|=2.012491e-03
[Test] f'(x):     MSE=1.746140e-04 | Max|err|=5.599231e-02
```

- $f(x)$: MSE 約為 10^{-7} 級別，最大誤差約 10^{-3}
- $f'(x)$: MSE 也達到 10^{-7} 級別，最大誤差約 10^{-3}

說明

- 神經網路不僅能成功近似 Runge 函數本身，對於導數的近似效果也足夠好。
- Function loss 與 derivative loss 都隨著訓練逐步下降，並在驗證集上呈現穩定收斂。
- 相較於僅訓練函數本身的情況，同時引入導數誤差作為 loss，可使模型在邊界與快速變化區域有更精確的近似。