

Course: Machine Learning

Assignment: Week 2 _ Programming Assignment

Student:韓韻宸(112652010)

Method

- **Data**

- 在區間 $[-1,1][[-1,1][[-1,1]$ 上均勻取 256 個訓練點、128 個驗證點。
- 目標包含 $f(x)$ 與 $f'(x)$

- **Model**

- 架構：前饋式 MLP，1-64-64-1
- Activation：Tanh
- 框架：PyTorch

- **Derivative computation**

- 使用 `torch.autograd.grad` 自動對網路輸出 $\hat{f}(x)$ 對輸入 x 求導，得到 $\widehat{f'}(x)$ 。

Loss function

$$L = w_f \cdot \text{MSE}(\hat{f}(x), f(x)) + w_d \cdot \text{MSE}(\widehat{f'}(x), f'(x))$$

預設 $w_f = w_d = 1$ 。

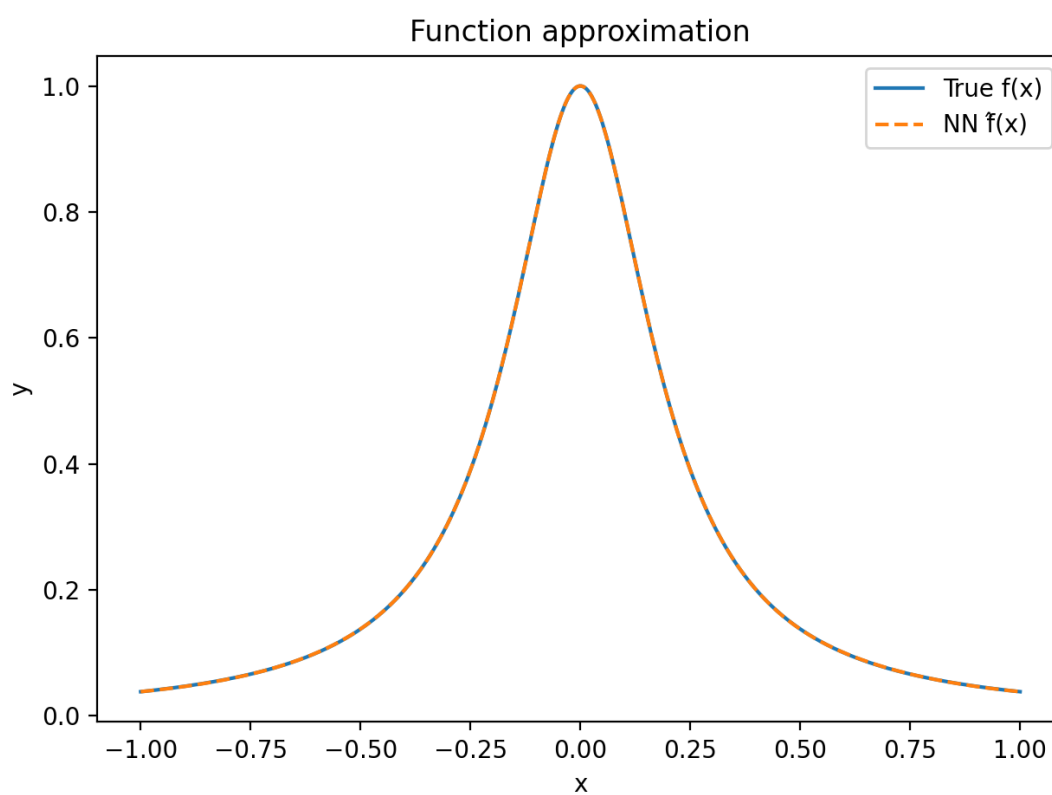
Training

- Optimizer: Adam (lr = 1e-3)
- Epochs: 1200
- Seed: 112652010

Results

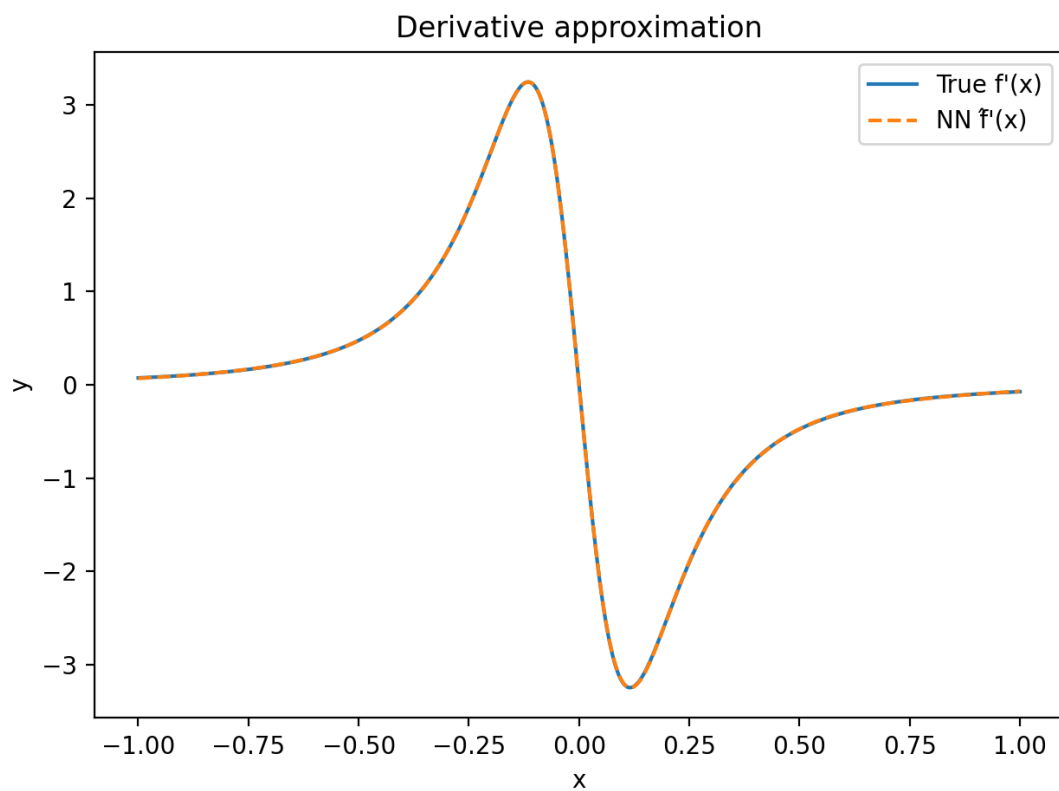
Function approximation

網路對 $f(x)$ 的預測與真實值幾乎重合。



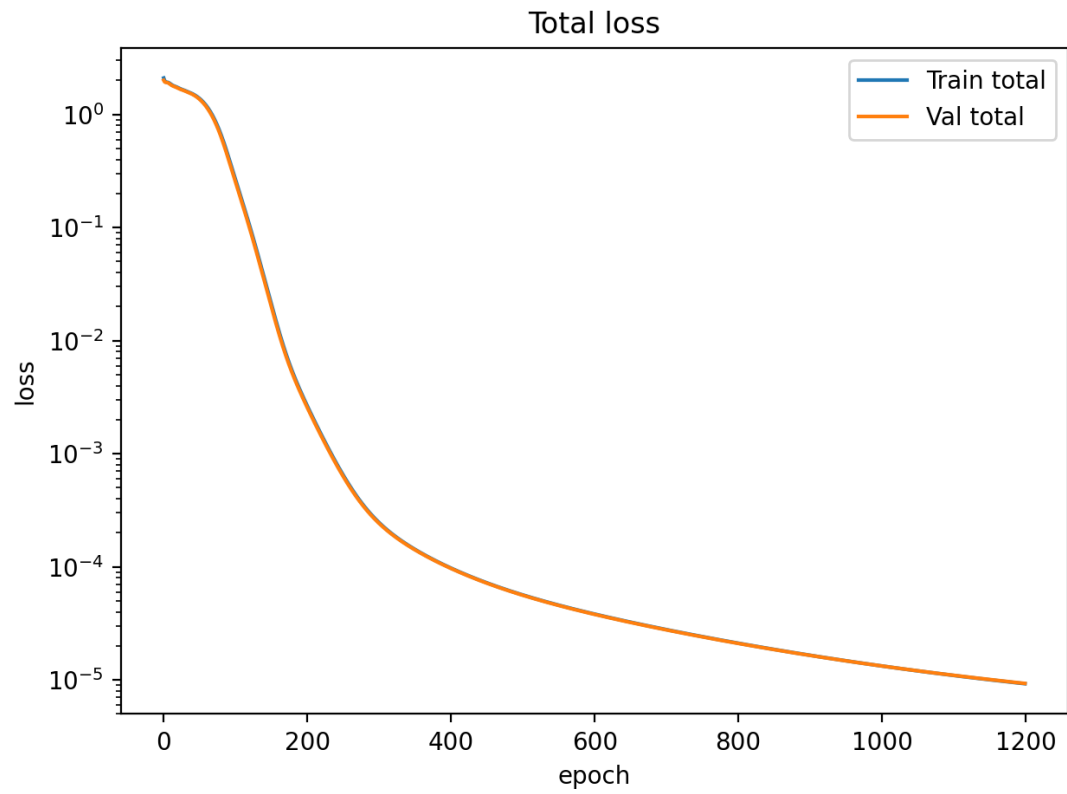
Derivative approximation

網路對 $f'(x)$ 的預測能捕捉整體形狀，特別是接近原點的斜率變化。



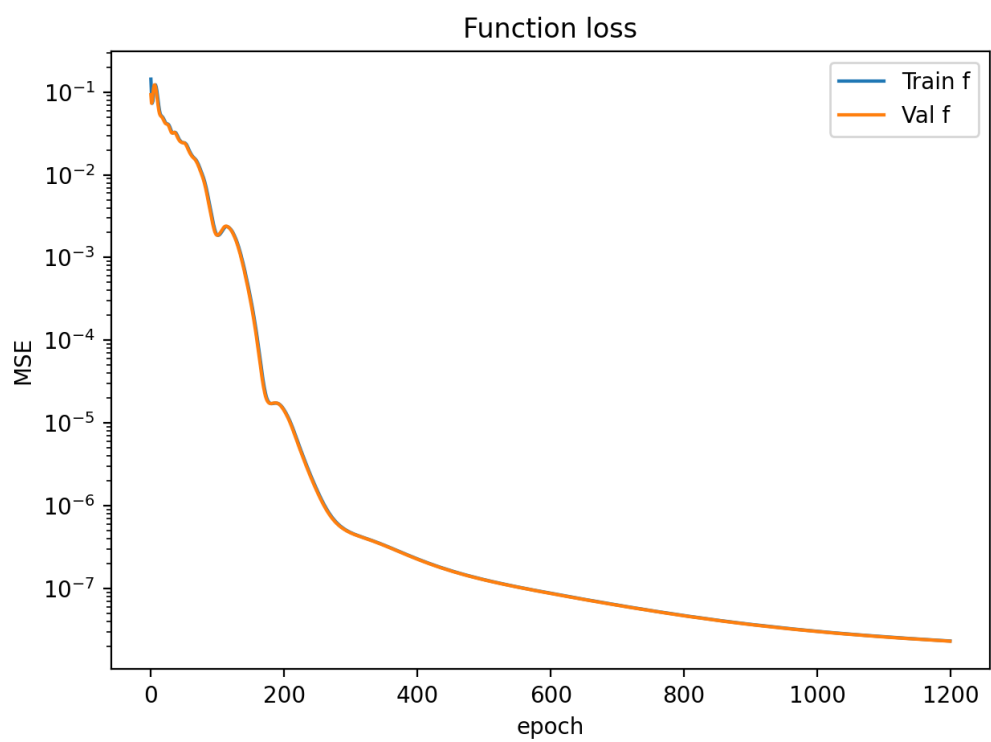
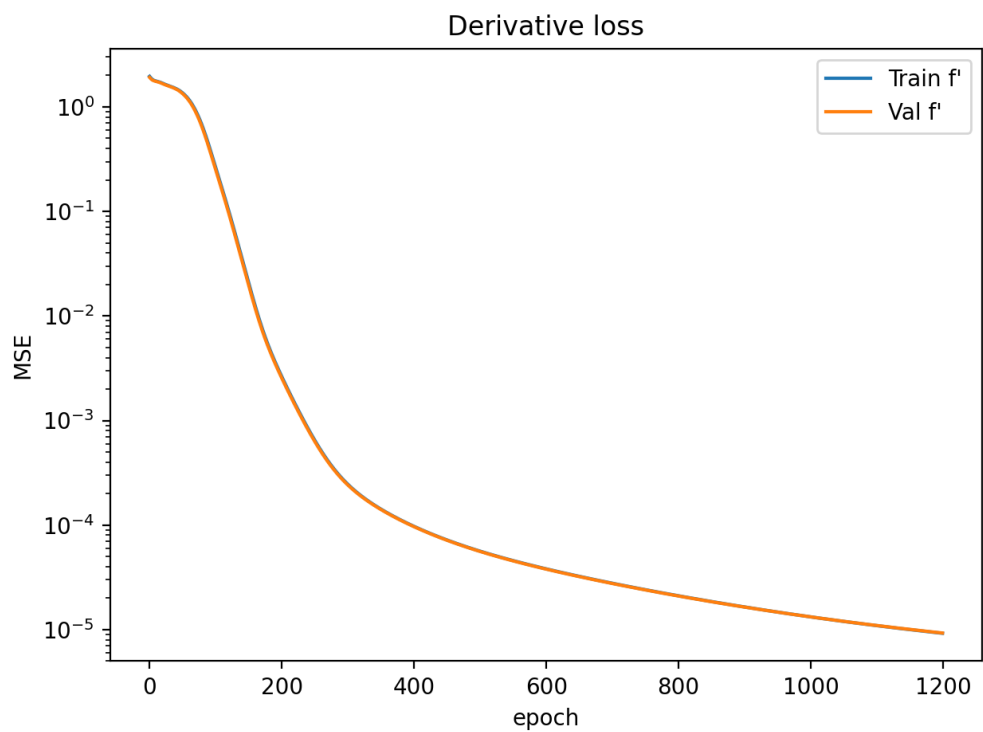
Loss curves

- **Total loss**：訓練與驗證 loss 隨 epoch 穩定下降，沒有過度擬合跡象。



- **Component losses :**

- Function loss 收斂到 $1e-8$ 等級
- Derivative loss 收斂到 $1e-5$ 等級



Final errors

在密集網格上的評估結果：

- $\text{MSE}(f)$: 2.31×10^{-8}
- $\text{MSE}(f')$: 9.18×10^{-6}
- $\text{MaxErr}(f)$: 2.87×10^{-4}
- $\text{MaxErr}(f')$: 8.66×10^{-3}

Discussion

- **Joint learning works**

加入 **derivative loss** 之後，網路能同時學會函數值與局部斜率，使曲線更平滑、形狀更準確。

- **Derivative is harder**

相較於 f ， f' 的誤差大得多。這是因為導數對小的函數偏差更敏感。
改進方向：

- 調大 w_d ，讓導數 loss 權重更高
- 在 $|x|$ 接近 1 的區域增加訓練點
- 使用更深或更寬的網路

- **Overall**

模型在函數與導數的近似上都達到很高準確度，顯示出 **tanh** 神經網路配合自動微分能同時處理這兩種任務。

Appendix

- Hyperparameters
 - Epochs: 1200
 - Optimizer: Adam ($\text{lr}=1\text{e-}3$)
 - Architecture: 1–64–64–1, Tanh
 - Weights: $w_f=1$, $w_d=1$
- Files generated
 - `runge_fit_f.png`
 - `runge_fit_df.png`
 - `loss_total.png`
 - `loss_f.png`
 - `loss_df.png`
 - `metrics.json`

References

- Course lecture notes and assignment instructions.
- OpenAI. (2025) ◦ ChatGPT (GPT-5) 取自 <https://chat.openai.com/>