Report: Neural Network Approximation of Runge Function and Its Derivative

1. Method

在這份作業中，我們的目標是用一個神經網路 (Neural Network) 來同時學習：

- Runge 函數
  $f(x) = 1 / (1 + 25 * x^2)$
- 它的導數
  $f'(x) = -50 * x / ((1 + 25 * x^2)^2)$

為了做到這件事，我們設計了一個神經網路，它的輸入是 x，輸出是兩個值：

1. f(x) 的近似值
2. f'(x) 的近似值

接著我們定義了一個損失函數 (Loss function)，用來衡量預測和真實答案的差距：

$L = MSE(f(x), f\hat{}(x)) + MSE(f'(x), f\hat{}'(x))$

也就是說，這個神經網路必須同時把函數和導數都學好。

我們把區間 [-1, 1] 均勻取樣點，分成訓練集 (80%) 和驗證集 (20%)，然後用 Adam optimizer 來訓練模型。

---

2. Results

觀察到以下結果：

- 函數近似
  - 神經網路成功學到 Runge 函數的形狀。
  - 預測曲線和真實曲線幾乎重合。
- 導數近似
  - 神經網路能學到導數的大致趨勢：在 x=0 附近導數接近 0，在正負區域分別為負/正。

- 雖然比函數稍微難學，但誤差仍然不大。
- **Loss 曲線**
  - 訓練過程中，訓練誤差和驗證誤差都逐漸下降並趨於平穩，說明模型有收斂。
- **誤差數值 (MSE)**
  - f(x) 的 MSE ≈ 1e-4
  - f'(x) 的 MSE ≈ 1e-3
  - 最大誤差 (Max error) 出現在區間邊緣 (|x|=1)，因為 Runge 函數在邊界變化較快。

---

3. Discussion

- 模型表現
  - 神經網路可以同時學習函數和導數，說明它有能力捕捉函數的形狀與變化率。
  - 導數比函數更敏感（小變化會被放大），所以導數誤差比函數大是合理的。
- 誤差來源
  - Runge 函數在 |x|=1 附近變化劇烈，導數也很大，所以邊界誤差較高。
  - 神經網路在有限的訓練點上學習，不可能完全還原所有細節。
- 改進方法
  - 增加神經網路的深度或寬度，提高模型容量。
  - 增加訓練點數，尤其在函數變化快的區域取更多點。
  - 調整 Loss 函數，給導數誤差更高的權重，讓模型更重視導數的準確度。

✅ 總結

這次實驗證明，神經網路不僅能近似函數，還能同時學習導數。
結果顯示模型在函數和導數上都有不錯的表現，雖然導數誤差比函數大，但整體趨勢正確，屬於成功的函數與導數近似。