

Programming Assignment

實驗目標是使用 **Feedforward Neural Network**，來近似 Runge function：

$$f(x) = \frac{1}{1 + 25x^2}, \quad x \in [-1, 1]$$

整個流程分為三個部分：資料取樣、程式設計與結果分析。

1. 資料取樣與函數設定

- 在區間 **[-1,1]** 隨機取樣 **2400** 個輸入點 x_i 。
- 輸出由 **Runge function** 計算：

$$y_i = f(x_i) = \frac{1}{1 + 25x_i^2}$$

- 資料集以 **5:1** 比例切分為訓練集與驗證集。
- tanh** 出現在模型的 **隱藏層激活函數 (activation)**，所扮演的角色是提供非線性轉換。

2. 程式設計 (依 Assignment_2 第1題概念)

模型架構

- 輸入層**：1 個神經元
- 隱藏層**：兩層，每層 32 個神經元，激活函數為 `tanh`
- 輸出層**：1 個神經元，線性輸出 \hat{y}

公式對應程式

- 前向傳遞 (Forward pass)**

$$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}, \quad a^{[l]} = \tanh(z^{[l]})$$

```
z1 = self.lin1(x); a1 = self.act(z1)
z2 = self.lin2(a1); a2 = self.act(z2)
yhat = self.lin3(a2) ````
```

- 損失函數 (MSE)**

$$C = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$$

- y_i : 真實函數值
- \hat{y}_i : 模型的模擬值

```
criterion = nn.MSELoss()
loss = criterion(yhat, yb)``
```

3

- 反向傳播與更新

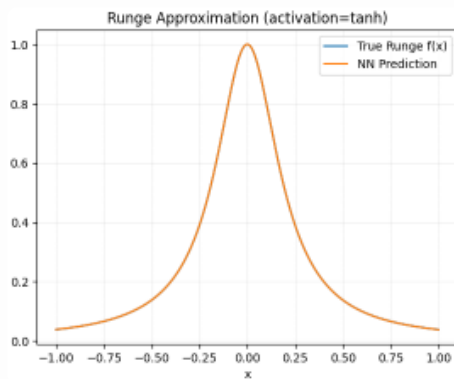
$$W^{[l]} \leftarrow W^{[l]} - \eta \frac{\partial C}{\partial W^{[l]}}$$

```
loss.backward()
optimizer.step()
```

3. 實驗結果與討論

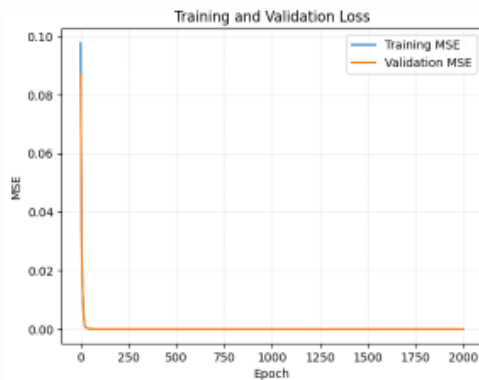
1 函數近似

- NN 輸出與 Runge function 幾乎完全重疊，效果佳



2 Loss 曲線

- 訓練與驗證 Loss 曲線快速下降並趨於穩定，無 overfitting。



3 誤差分析

- 測試誤差表現：MSE = 5.35×10^{-8} ，最大誤差 = \$5.58

Epoch 1	Train MSE: 9.776226e-02	Val MSE: 8.694452e-02
Epoch 200	Train MSE: 9.092923e-07	Val MSE: 7.311983e-07
Epoch 400	Train MSE: 9.961971e-08	Val MSE: 5.077722e-08
Epoch 600	Train MSE: 3.984066e-06	Val MSE: 6.365794e-06
Epoch 800	Train MSE: 2.311152e-07	Val MSE: 3.799387e-08
Epoch 1000	Train MSE: 7.552220e-08	Val MSE: 2.239216e-07
Epoch 1200	Train MSE: 2.310033e-07	Val MSE: 5.787098e-07
Epoch 1400	Train MSE: 8.919029e-08	Val MSE: 7.417257e-08
Epoch 1600	Train MSE: 9.385491e-08	Val MSE: 7.750679e-08
Epoch 1800	Train MSE: 1.723216e-06	Val MSE: 1.457905e-06
Epoch 2000	Train MSE: 2.496845e-08	Val MSE: 5.228326e-08

Test MSE = 5.353794e-08

Max Abs Error = 5.579554e-04