1. Objective

The goal of this study is to approximate the Runge function

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

and its derivative

$$f'(x) = -\frac{50x}{(1 + 25x^2)^2}$$

using a neural network.

Approximating derivatives directly is challenging because small errors in the function can be amplified in its derivative. Therefore, we construct a neural network and train it using a **combined loss** that accounts for both the function values and the derivative values.

2. Methodology

Dataset

- 200 evenly spaced points in $[-1, 1]$ are used as the training set.
- 50 additional points are used as the validation set.
- Both function values $f(x)$ and derivatives $f'(x)$ are computed analytically.

Neural Network Architecture

- Input layer: 1 neuron (x)
- Hidden layers: 2 layers with 64 neurons each, **tanh** activation for smooth approximation
- Output layer: 1 neuron (predict $f(x)$)

Training Setup

The total loss consists of **two components**:

1. Function loss (MSE):

$$\text{loss}_f = \frac{1}{N}\sum_{i=1}^{N}\left(f_{\text{pred}}(x_i) - f_{\text{true}}(x_i)\right)^2$$

2. Derivative loss (MSE):

$$\text{loss}_{f'} = \frac{1}{N}\sum_{i=1}^{N}\left(f'_{\text{pred}}(x_i) - f'_{\text{true}}(x_i)\right)^2$$

Total loss:

$$\text{loss} = \text{loss}_f + \text{loss}_{f'}$$

- The derivative is computed using **PyTorch autograd** during training.
- Optimizer: Adam with learning rate 0.001
- Training epochs: 3000

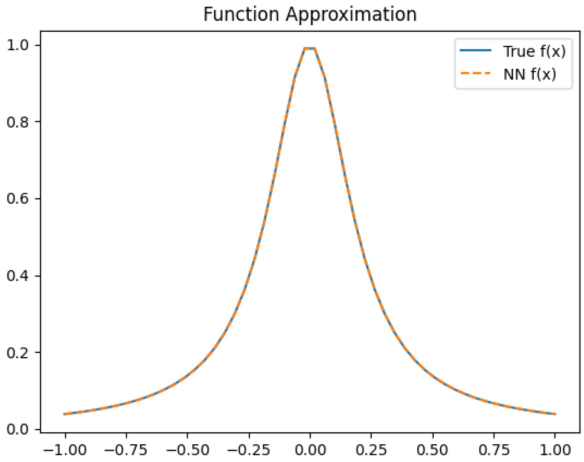
Evaluation

- Compare the true function with the neural network prediction.
- Plot training and validation loss curves.
- Compute validation errors: MSE and maximum absolute error.

3. Results

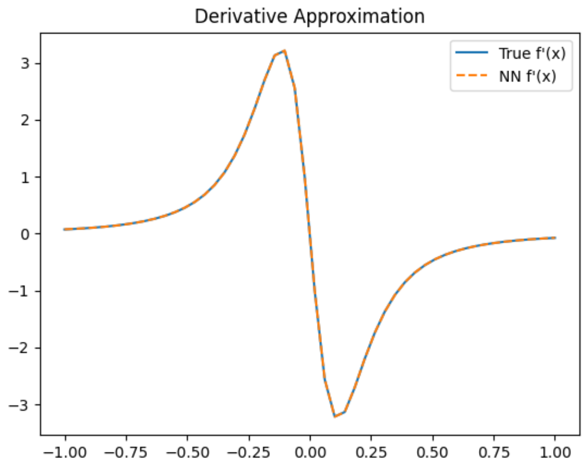
Function Approximation

- Blue solid line: true Runge function
- Orange dashed line: neural network prediction
- The network successfully captures the overall shape, especially in the central region.



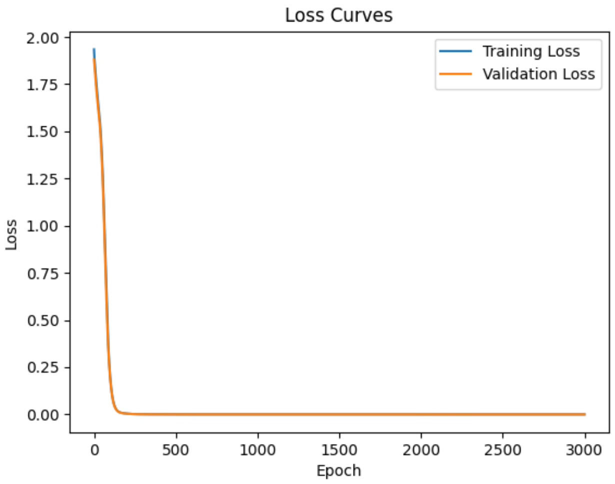
Derivative Approximation

- Blue solid line: true Derivative of Runge function
- Orange dashed line: neural network prediction
- The network successfully captures the overall shape, especially in the central region.



Loss Curves

- Training and validation losses decrease steadily and converge, showing stable learning without overfitting.



Result

- Training with total Loss (Function loss & Derivative loss)

```
Epoch 200/3000, Train Loss: 0.004282, Val Loss: 0.004169
Epoch 400/3000, Train Loss: 0.000354, Val Loss: 0.000348
Epoch 600/3000, Train Loss: 0.000109, Val Loss: 0.000108
Epoch 800/3000, Train Loss: 0.000050, Val Loss: 0.000049
Epoch 1000/3000, Train Loss: 0.000028, Val Loss: 0.000028
Epoch 1200/3000, Train Loss: 0.000017, Val Loss: 0.000017
Epoch 1400/3000, Train Loss: 0.000012, Val Loss: 0.000012
Epoch 1600/3000, Train Loss: 0.000021, Val Loss: 0.000036
Epoch 1800/3000, Train Loss: 0.000008, Val Loss: 0.000008
Epoch 2000/3000, Train Loss: 0.000007, Val Loss: 0.000007
Epoch 2200/3000, Train Loss: 0.000026, Val Loss: 0.000036
Epoch 2400/3000, Train Loss: 0.000005, Val Loss: 0.000005
Epoch 2600/3000, Train Loss: 0.000077, Val Loss: 0.000089
Epoch 2800/3000, Train Loss: 0.000007, Val Loss: 0.000007
Epoch 3000/3000, Train Loss: 0.000003, Val Loss: 0.000004
```

- Error Metrics

```
Validation f(x) - MSE: 9.7083394e-08 Max error: 0.0004952401
Validation f'(x) - MSE: 3.497311e-06 Max error: 0.004785776
```