Neural Network Approximation of the Runge Function and Its Derivative

Your Name: Chiu Bo Hsiang   Student ID: 112652037

September 23, 2025

1 Method

We approximate the Runge function

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

and its derivative

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

using a multilayer perceptron (MLP). The model is trained with a joint loss:

$$\mathcal{L} = \mathrm{MSE}\Big(\hat{f}(x), f(x)\Big) + \lambda\,\mathrm{MSE}\Big(\widehat{f'}(x), f'(x)\Big), \tag{3}$$

where λ balances the derivative term.

2 Experiment Settings

Table 1 lists the training configuration used. Replace any fields as needed.

Table 1: Training configuration

Item	Value
Model	MLP $(64, 64)$, activation: `tanh`
Optimizer	Adam, learning rate 1e−3, weight decay 1e−6
Batch size	128
Epochs (max)	5000 with early stopping (patience 150)
Derivative weight λ	0.1
Data	Train: 256 Val: 128 Sampling: uniform on $[-1, 1]$

3 Results

Figures 1-3 present the function/derivative fits and the training/validation loss curves. Place the three image files produced by your code in the same folder as this `.tex` file: `fig_f_fit.png`, `fig_df_fit.png`, `fig_losses.png`.

4 Quantitative Metrics

Table 2 and Table 3 summarize the errors on a dense grid of 1000 points. Replace the placeholders with your program's outputs (shown in Colab console).

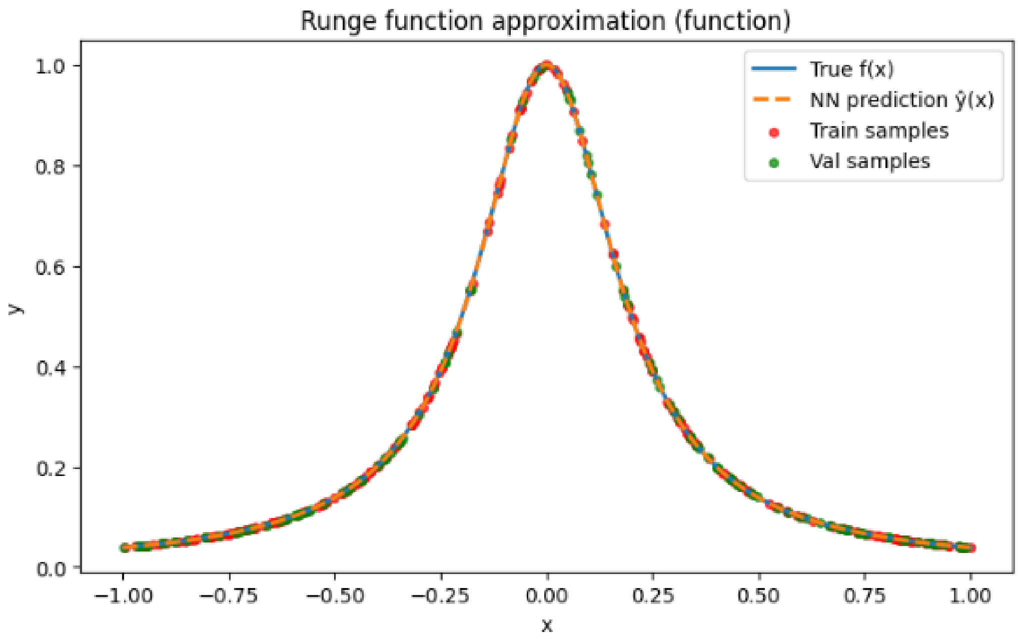

Figure 1: True function $f(x)$ vs. NN prediction $\hat{f}(x)$ together with training/validation samples.

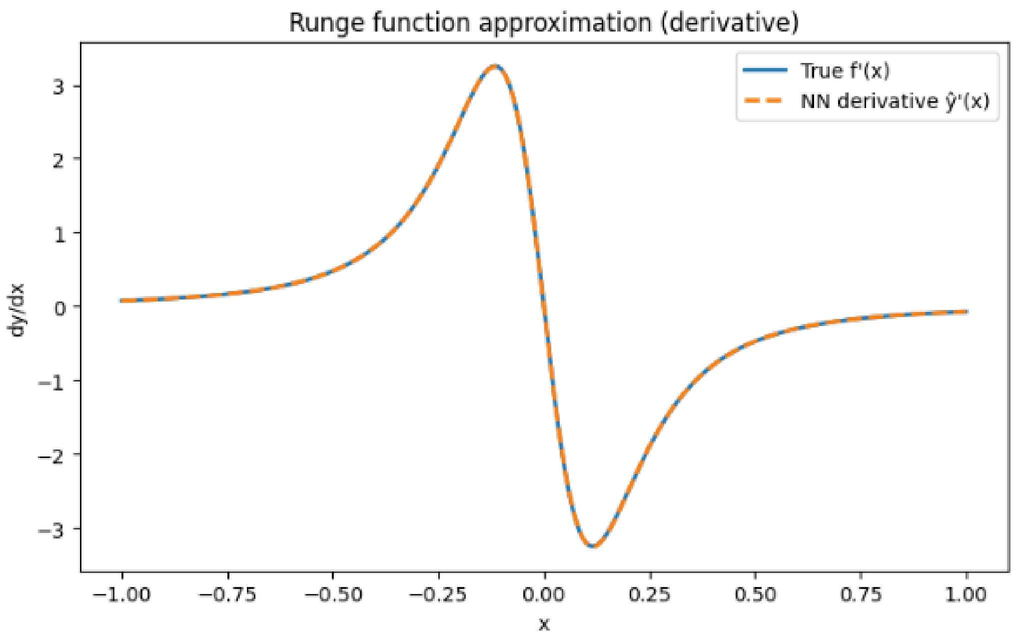

Figure 2: True derivative $f'(x)$ vs. predicted derivative $\widehat{f'}(x)$.

Table 2: Function $f(x)$ metrics

Metric	Value
MSE	1.04×10^{-8}
Max Error	4.62×10^{-4}

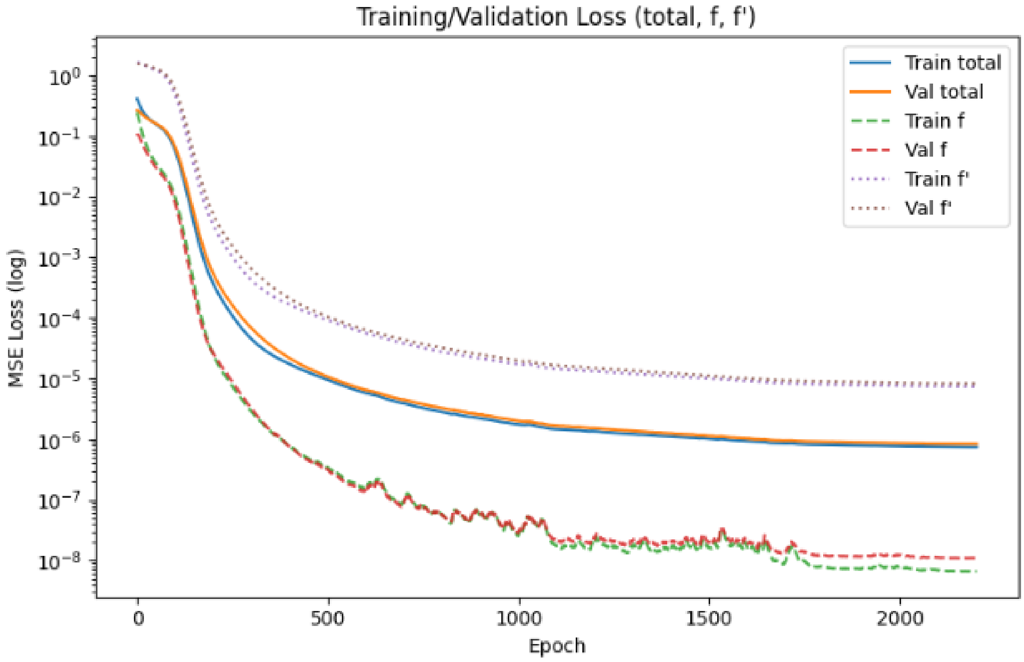

Figure 3: Training/Validation loss curves in log scale: total, function-only, and derivative-only.

Table 3: Derivative $f'(x)$ metrics

Metric	Value
MSE	9.53×10^{-6}
Max Error	1.17×10^{-2}

5 Discussion

The network fits $f(x)$ extremely well (MSE $\sim 10^{-8}$). Learning $f'(x)$ is harder; errors are larger (MSE $\sim 10^{-5}$), but still small. Training and validation curves are close, indicating good generalization without severe overfitting. Increasing λ could further reduce derivative error at the cost of slightly worse function error; using Chebyshev sampling near the interval boundaries can also help.

References

[1] Carl Runge, "Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten," 1901.