1141ML-Week 3 Programming Assignment 02

Runge Function and Derivative Approximation

313652018 王宣瑋

September 2025

Basic Settings

We basically use Neural Network to complete the program. At first, we use ”*tanh*” nonlinear activation function to do the question.

0.1 Settings

- dataset: sampling method (**Chebyshev points** over [-1, 1])
- dataset size:
  - training set: 256
  - validation set: 256
  - test set: 1000
- Neural network architecture:
  - hidden layers: 2
  - nonlinear function: tanh
  - tanh units: 64
- optimizer: Adam
- learning rate: $2e-3$
- weight decay: $1e-5$
- epoches: 800
- Loss metric: MSE

Result

- Test MSE : 1.197438e-07
- Test Max |err|: 6.388045e-04
- Test Derivative MSE : 8.212621e-06
- Test Derivative Max |err|: 8.734465e-03

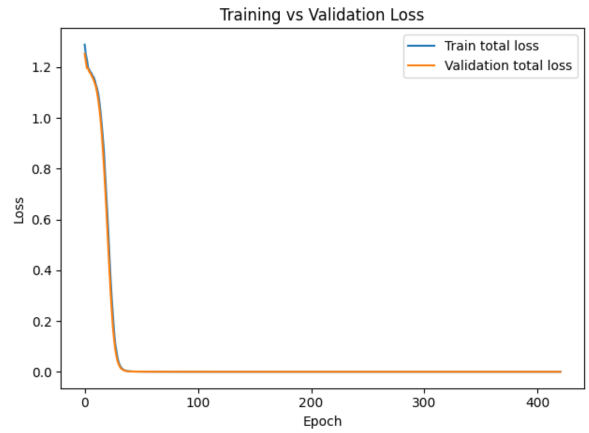
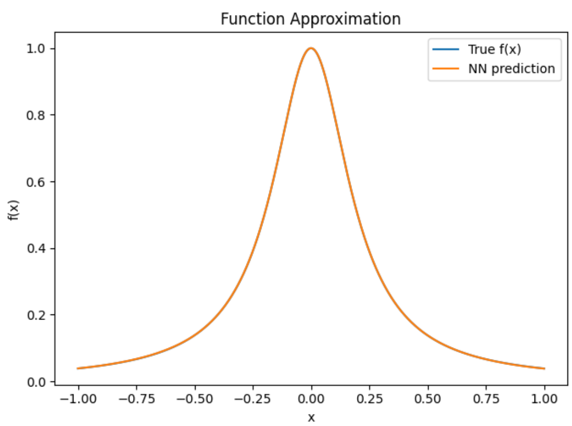
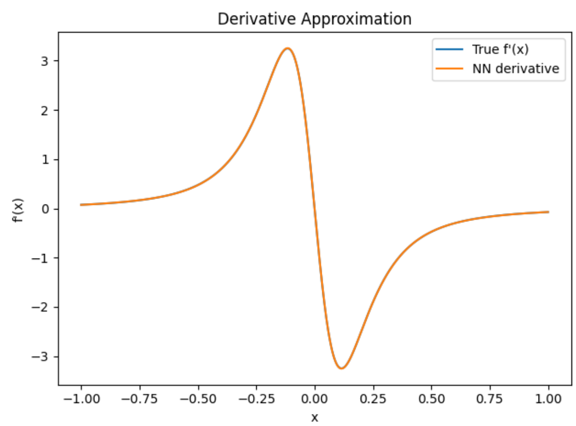


Figure 1: The training and validation loss



(a) Function approximation



(b) Derivative approximation

Figure 2: Neural network results: (a) Function approximation (b) Derivative approximation

Discussion

Function Approximation Analysis

Figure 2a illustrates the comparison between the true Runge function and the neural network prediction. The graph indicates that the neural network successfully captures both the steep behavior near the origin and the flatter regions near the boundaries.

The test mean squared error (MSE) reaches 1.20×10^{-7}, and the maximum absolute error is as low as 6.39×10^{-4}. These error levels confirm that the network provides a highly accurate approximation of the target function.

The Training and Validation Loss

In Figure 1, the rapid convergence of both training and validation losses to nearly zero further suggests that the chosen network architecture, together with Chebyshev nodes as training points, is well-suited for approximating smooth but steep functions such as the Runge function. Importantly, no significant gap between training and validation loss was observed, implying that the model generalizes well without evidence of overfitting.

Derivative Approximation Analysis

Figure 2b shows the comparison between the true derivative $f'(x)$ of the Runge function and the derivative predicted by the neural network. The two curves align closely over the entire domain $[-1, 1]$, *with only minor deviations observed near the origin where the curvature is steepest.*

The test derivative mean squared error (MSE) is 8.21×10^{-6}, while the maximum absolute error is 8.73×10^{-3}. Although these errors are slightly higher than those of the function approximation, they remain small and demonstrate that the network effectively captures not only the function values but also their variations.

This result highlights the advantage of **incorporating a derivative-based loss term during training.** By penalizing discrepancies in both function values and derivatives, the network *learns a more faithful representation of the underlying function.* The smoothness of the predicted derivative curve also indicates that the chosen architecture and activation functions are capable of modeling higher-order information without introducing spurious oscillations.