

Instructions & Screenshots

1. App Interface and Initial Configuration

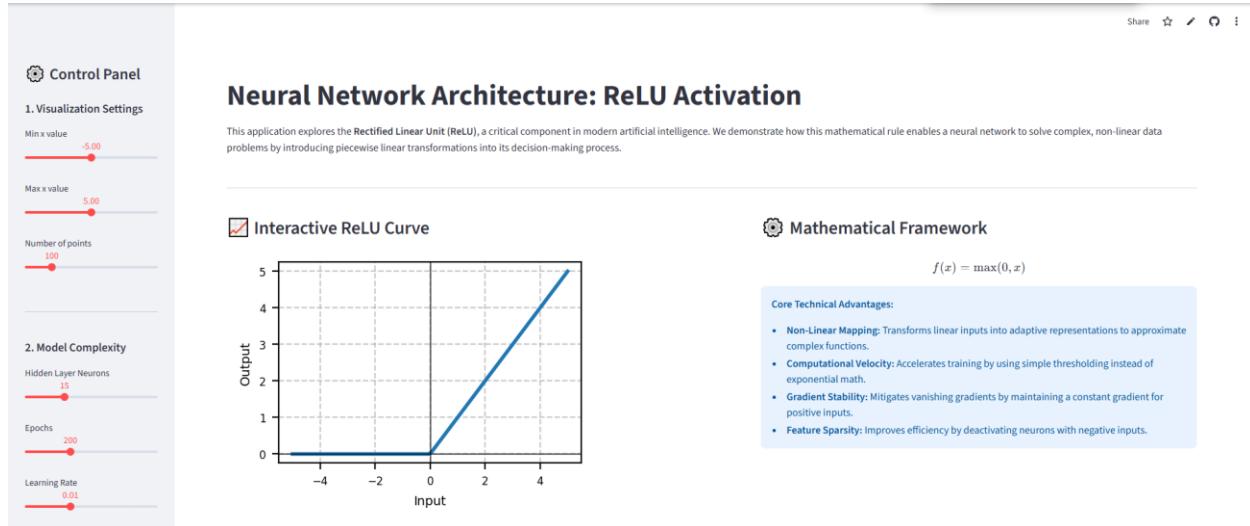


Figure 1

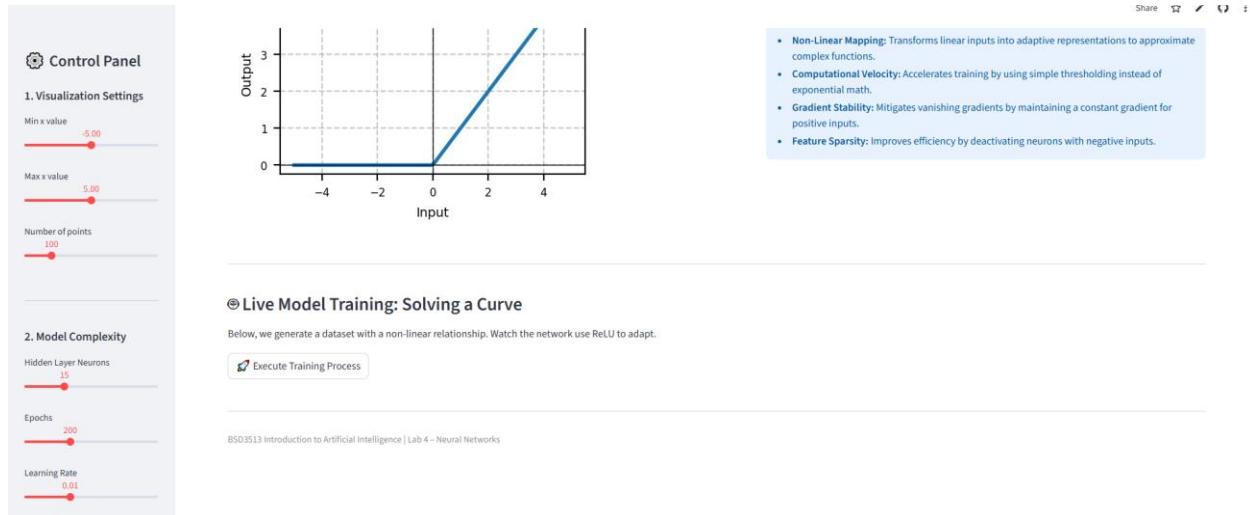


Figure 2

Figure 1 and Figure 2 show screenshots of the application immediately after it loads. The interface is designed as a comprehensive dashboard for exploring neural network components, including:

Sidebar Control Panel	A dedicated section for configuring visualization parameters (input range and number of data points) and neural network hyperparameters (hidden layer neurons, epochs, and the learning rate).
Main Header	A primary title accompanied by a concise technical description that introduces the role of activation functions in introducing non-linearity to neural network models.
Interactive ReLU Curve	Located in the first section, this visualization is dynamically updated in real-time according to the user's selections on the control panel, allowing for an immediate visual understanding of how this specific neural network component handles thresholding behavior.
Mathematical Framework	Positioned beside the curve, this component provides a mathematical and conceptual explanation of the ReLU algorithm, highlighting its advantages within a neural network architecture, such as gradient stability and computational efficiency.
Live Training Module	Figure 2 shows the second section of the app focuses on utilizing a Multi-Layer Perceptron (MLP) neural network to solve a non-linear regression problem.
Execution Training Process Button	A “Execute Training Process” button is provided, which the user can click after finalizing their chosen parameters to initiate the PyTorch-based backpropagation and neural network optimization loop.

2. App Interface Post-Training Execution

Figure 3 shows the screenshot of the app right after the user clicks “Execute Training Process” and the optimization loop completes.

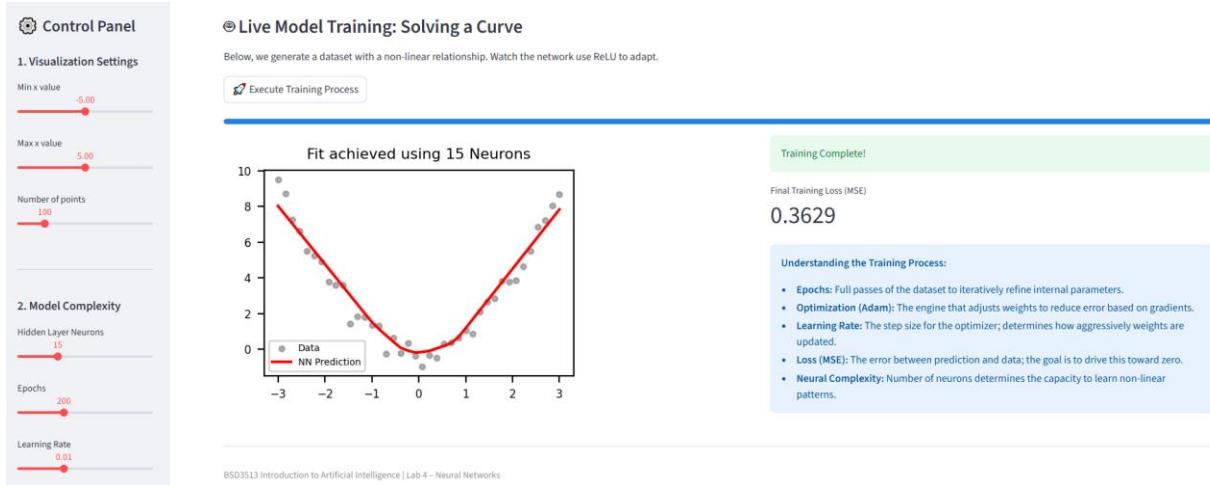


Figure 3

The interface presents the following results:

Neural Network Regression Plot	The main visualization area now displays a red “NN Prediction” line overlaid on the original “U-shape” data points. This line demonstrates how the neural network has successfully utilized the ReLU activation functions to “bend” and approximate the non-linear distribution of the dataset. The plot title reflects the specific neural network complexity used (e.g., “Fit achieved using 15 Neurons”), confirming that the model adapted its internal architecture based on the user’s sidebar selections.
Training Status and Metrics	A green “Training Complete!” status message appears, accompanied by a metric displaying the Final Training Loss (MSE). This numerical value (e.g., 0.3629) provides objective proof of the neural network’s accuracy in minimizing error during the training rounds.
Understanding the Training Process	Positioned to the right of the results, the info box allows users to refer the definitions of Epochs, Optimization (Adam), Learning Rate, Loss (MSE), and Neural Complexity for better understanding of the concepts.