

**Department of Electrical and Electronics Engineering**

**EEE 443: Neural Networks**

*Class Project IV Report*

**Name and Surname:** Ali Aral Takak

**Student ID:** 22001758

**Department:** EEE

**Instructor:** Erdem Koyuncu

**Problem**

The objective of the fourth course project is to implement a simple neural network to accomplish the task of curve fitting using a dataset generated from a given mathematical function. The dataset consists of points, where each input value is drawn uniformly from and each corresponding target value is computed as:

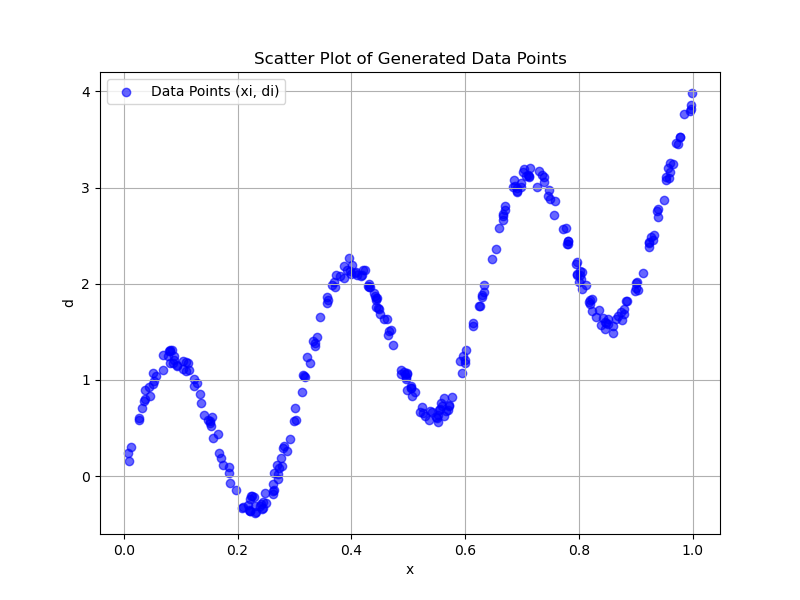
Where can be considered as additive noise sampled from the interval . A feedforward neural network with a architecture is trained using backpropagation to minimize the Mean-squared Error (MSE) between the network’s predictions and the target values. The hidden layer uses the tanh activation function, while the output neuron applies a linear activation. Gradient descent is employed for optimization, with an adaptive learning rate adjustment method to also employ stable convergence.

**Implementation and Results**

The pseudocode given below explains the complete implementation for the task mention in the previous chapter.

* **Step 1: Initialize Parameters**
  1. Set the number of hidden neurons, .
  2. Define the learning rate , such as
  3. Set the number of training epochs, in our case.
  4. Initialize the weight matrices:
     1. as a random matrix.
     2. as a random matrix.
  5. Initialize the bias vectors.
     1. as a zero vector.
     2. as a zero vector.
* **Step 2: Generate Dataset**
  1. Draw random numbers uniformly from .
  2. Draw random numbers uniformly from .
  3. Compute the values using
* **Step 3: Train the Neural Network using Backpropagation**
  1. For each epoch from to the total number of epochs:
     1. Initialize Mean Squared Error (MSE) as .
     2. For each data point from to :
        + **Forward Pass:**
          - Compute hidden layer activation and .
          - Compute output layer activation and .
        + **Compute Error:** 
          - Compute error as .
          - Adjust Mean-squared Error as .
        + **Backpropagation:**
          - Compute gradient for as .
          - Compute gradient for as
          - Compute error propagated back to the hidden layer as
          - Compute gradient for as .
          - Compute gradient for as .
        + **Update Weights and Biases:**
     3. Compute average Mean-squared Error over all data points.
     4. Execute adaptive learning adjustment if Mean-squared error increases compared to the previous epoch as .
* **Step 4: Visualize the Training Process and Results**
  1. Plot the Mean-squared Error vs. Epochs to observe the learning process.
  2. Compute the final fitted function using the trained weights.
  3. Plot the fitted function over the original curve.

We initially set our hyperparameters as epochs and a learning rate of . The figure given below displays the generated data points and the curve:



**Figure 1:** Generated data points and the curve.

The figure given below displays the Mean-squared Error versus epoch plot of the Neural Network:

A graph with a line

AI-generated content may be incorrect.

**Figure 2:** MSE versus epoch chart for epoch=5000.

The figure given below displays the curve fitting executed using the trained Neural Network, over the original data points:

A graph with a red line

AI-generated content may be incorrect.

**Figure 3:** Curve fitting using the trained Neural Network.

**Conclusion**

This project successfully implemented a feedforward neural network to approximate a nonlinear function using a dataset generated from a combination of a sinusoidal term, a linear component, and additive noise. The network architecture consisted of one input neuron, 24 hidden neurons with tanh activation, and one output neuron with a linear activation function. Training was performed using the backpropagation algorithm, whereas an adaptive learning rate ensured stable convergence. Results yielded that the error reduces significantly over time, which indicates the successful adjustment of network parameters. The final fitted curve demonstrated that the designed model has the capabilities and effectiveness for nonlinear curve fitting.