

Lab Assignment 1: Handwritten Character Recognition (MNIST) with ANN

Objective

The goal of this assignment is to build and train a simple Artificial Neural Network (ANN) to (from scratch without any library or framework.) recognize handwritten digits. You will use the classic MNIST dataset, a foundational benchmark in the field of machine learning. By completing this assignment,

The Dataset: MNIST

The **MNIST** (Modified National Institute of Standards and Technology) dataset is a large collection of handwritten digits. It contains 60,000 training images and 10,000 testing images. Each image is a grayscale picture, 28 pixels wide by 28 pixels high. The digits are from 0 to 9.

Neural Network Architecture

Your task is to implement an ANN with the following three key layers:

1. Input Layer

The input to our network will be a single image from the MNIST dataset. Since each image is 28×28 pixels, we need to flatten it into a one-dimensional array of numbers.

- **Number of Neurons:** $28 \times 28 = 784$ neurons.
- **Purpose:** This layer simply holds the pixel values of the input image. Each neuron corresponds to a single pixel.

2. Hidden Layer

The hidden layer is where the network learns to identify patterns and features from the input data, such as lines, curves, and edges.

- **Number of Neurons:** You will use **128 neurons** for this layer.
- **Activation Function:** The **ReLU (Rectified Linear Unit)** activation function should be used. It helps the network learn complex relationships in the data.

3. Output Layer

The output layer provides the final classification result. Since we are trying to predict one of ten possible digits (0-9), this layer will have ten neurons.

- **Number of Neurons: 10 neurons**, one for each digit class (0 through 9).
- **Activation Function:** The **Softmax** activation function should be used. Softmax will convert the outputs into a probability distribution, where the sum of all probabilities is 1. The neuron with the highest probability corresponds to the predicted digit.
- **Loss Function:** The **Categorical Cross-Entropy** loss function is appropriate for this multi-class classification task.