Handwritten Digit Recognition using a forward Neural Network

Samrakshyan Chapagain

Abstract

This project shows a simple feedforward Neural Network which predicts handwritten digits from the images shown to it. MNIST dataset is used for both training and testing the neural network. The neural network consists of one input layer, one hidden layer based on sigmoid activation, and a output layer based on softmax activation. The input layer receives a 28X28 image of a handwritten digit, and the output layer uses softmax activation for classification into 10 digit classes(0-9). The model learns from pixel intensities are flattened into a 784-dimensional vector (e.g. [0,0,0,7,8,....255] → (0-255)). The model is evaluated using accuracy metrics.

Introduction

Handwritten digit recognition is a classic problem in computer vision and machine learning. The goal of this project is to correctly identify digits ranging from 0 to 9 from grayscale images. This project uses MNIST dataset that contains 70,000 labeled images of handwritten digits, each of size 28x28 pixels. It implements a simple neural network model to classify these digits and evaluate its performance.

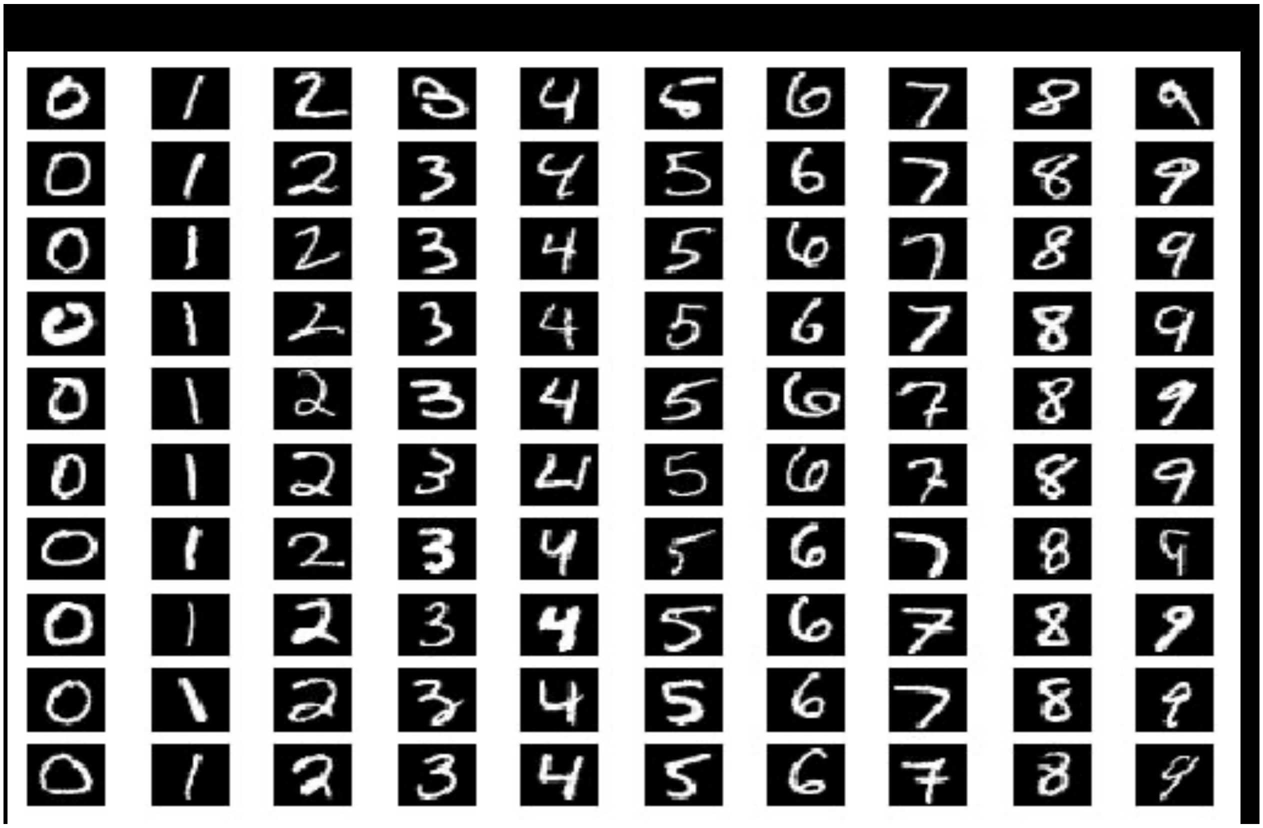
Dataset Description

The MNIST dataset contains:

- 60,000 training images
- 10,000 testing images
- Each image is 28x28 grayscale
- Each image is flattened into a 784-dimensional vector
- Labels range from 0 to 9

Each pixel value represents a number from 0 (black) to 255 (white), which is normalized before training.

Sample MNIST digits:

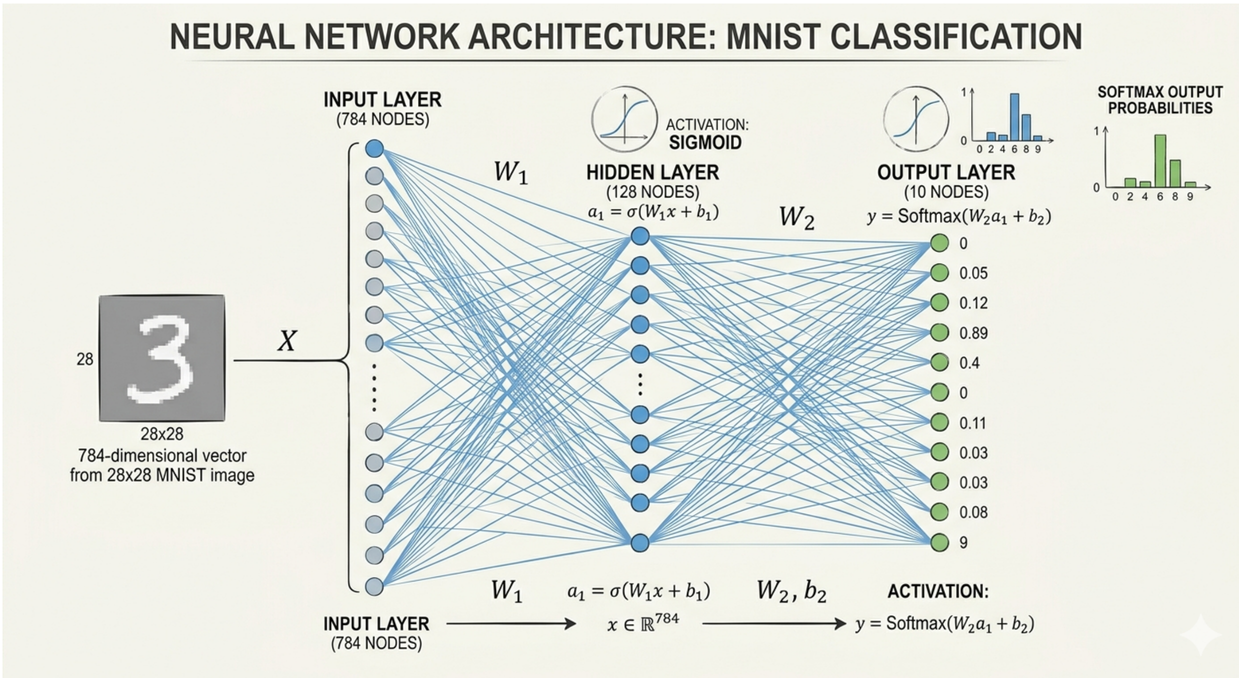


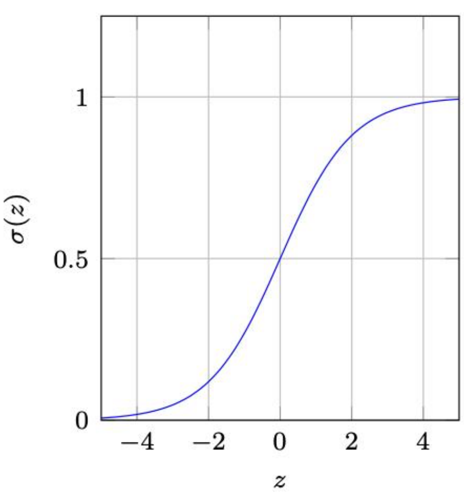
Methodology

1. **Data Preprocessing**
   Each image is flattened from 28x28 scale into a 784-dimensional vector. Pixel values are normalized by dividing every index of the vector by 255 so that index values exist from 0 to 1 only.

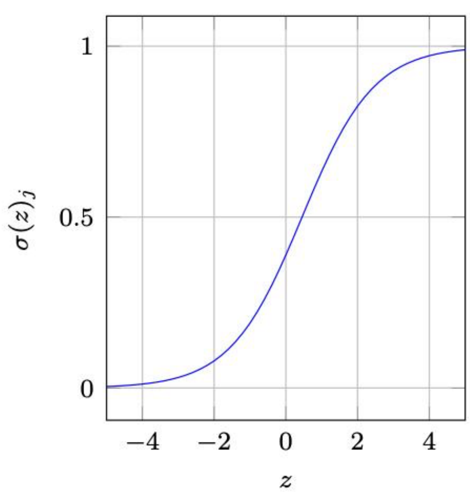
2. **Model Architecture**
   The neural network consists of:
   - Input layer
   - Hidden layer (sigmoid activation)
   - Output layer (softmax activation)

Neural Network Architecture





(a) Sigmoid activation function.



(b) Softmax activation function.

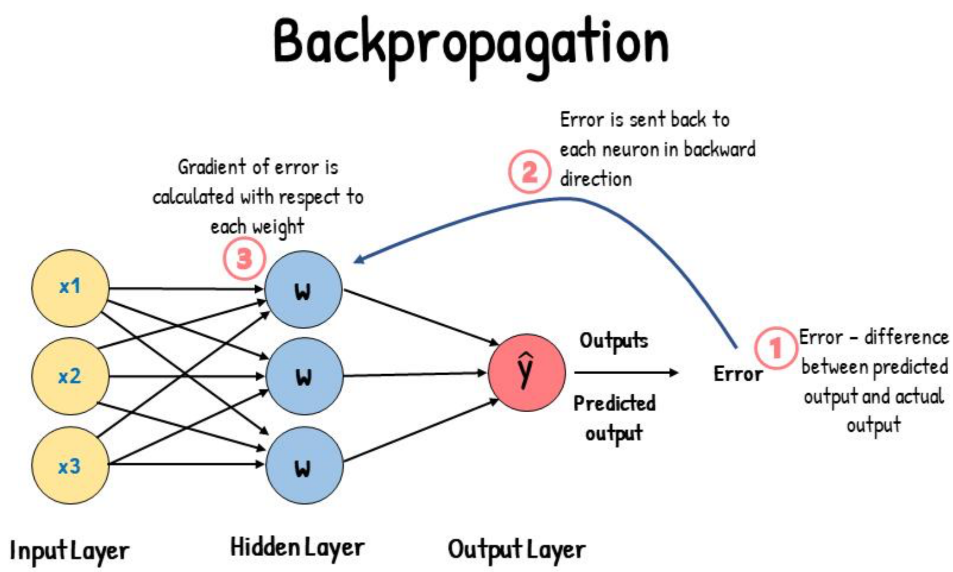
Figure 1: Sigmoid and Softmax activation functions

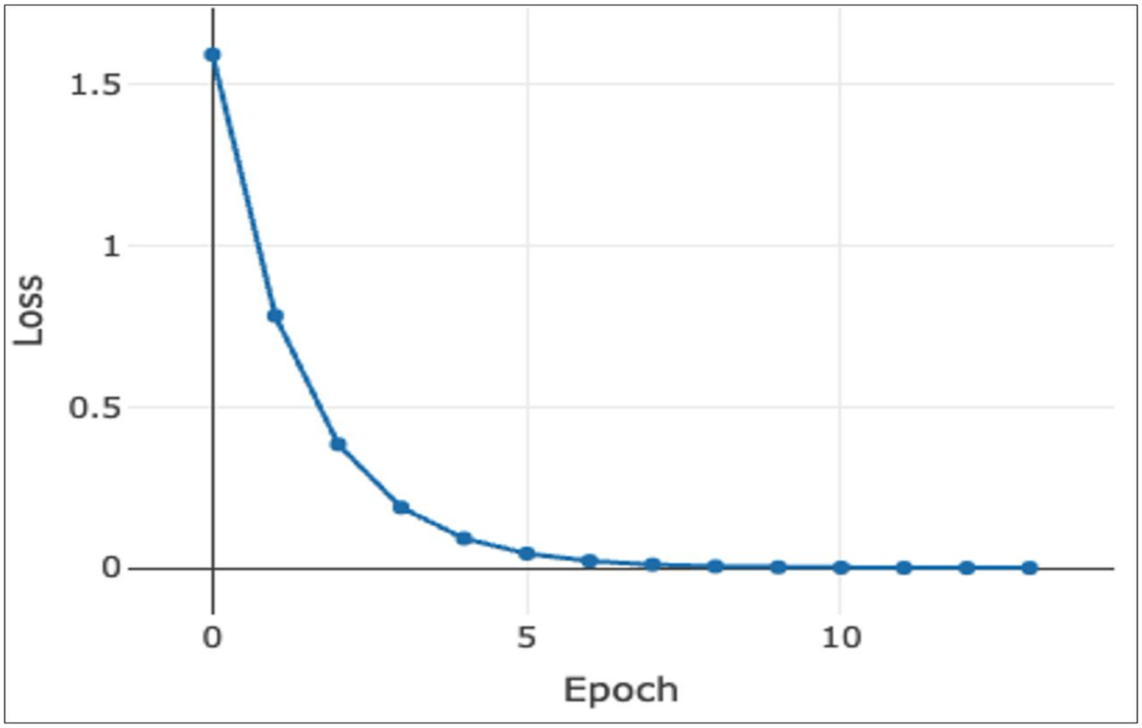
3. **Activation Functions**
   The sigmoid function introduces non-linearity in hidden layer, while softmax converts output values into probabilities across 10 classes (0-9).

Training Process

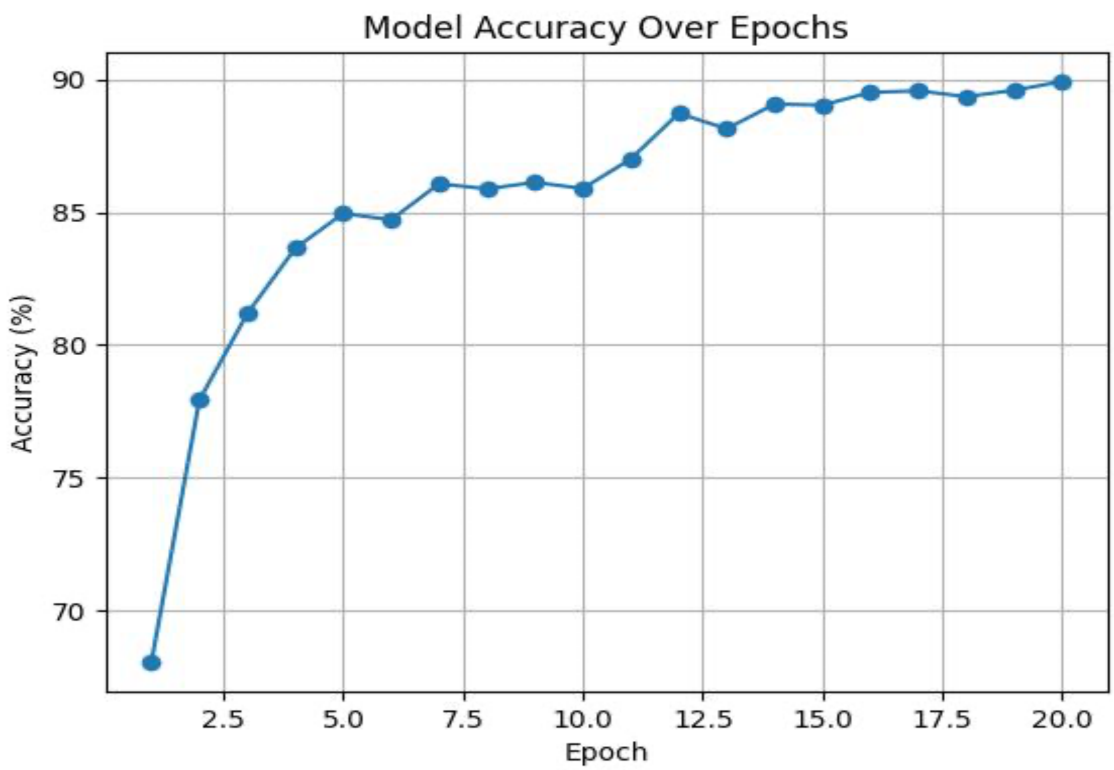
The model is trained using forward propagation and back propagation:

- Forward propagation computes prediction
- Loss is calculated using cross-entropy
- Gradients are computed using backpropagation
- Weights are updated using gradient descent





Loss
Epoch
1.5
1
0.5
0
0
5
10



Model Accuracy Over Epochs
Accuracy (%)
Epoch
90
85
80
75
70
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

Results

The model was evaluated on test data and achieved an accuracy of approximately 97.5%.

Observations:

- The model correctly classifies most of the digits.
- Errors commonly occurs between similar digits (e.g., 1 and 7, 3 and 8, 5 and 6)

```
mistakes: Predicted: 7, Actual: 4
mistakes: Predicted: 0, Actual: 8
mistakes: Predicted: 5, Actual: 9
mistakes: Predicted: 5, Actual: 3
mistakes: Predicted: 5, Actual: 6
```

Discussion

The model shows the capability of neural networks in learning patterns from image. However, performance is limited due to:

- Lack of advanced optimization techniques
- Simple architecture without convolutional networks.
- Use of sigmoid activation, which can suffer from vanishing gradients.

Conclusion

This project successfully implements a feedforward neural network for handwritten digit classification. While the model is functional and demonstrates learning capability, it has limitations in accuracy and generalization. Future improvements include using ReLU activation, deeper architectures, and convolutional neural networks.