A Neural Network-Based Digit Recognizer: Mathematical Foundations and Implementation

Asray Gopa

June 2024

Abstract

This paper presents the development and mathematical foundation of a neural network designed to recognize handwritten digits. The network is trained on the MNIST dataset, leveraging supervised learning techniques. We provide an in-depth analysis of the mathematical concepts, including data preprocessing, network architecture, forward and backward propagation, and optimization methods. Experimental results demonstrate the efficacy of the proposed model.

1 Introduction

Handwritten digit recognition is a classic problem in machine learning and pattern recognition, providing a benchmark for evaluating algorithms. This paper focuses on the implementation of a neural network model, from data preprocessing to training and evaluation, with a rigorous mathematical treatment of each step.

2 Dataset and Preprocessing

The MNIST dataset, a well-known benchmark, consists of 60,000 training and 10,000 test images of handwritten digits (0-9). Each image is 28x28 pixels, flattened into a 784-dimensional vector.

2.1 Data Representation

Given an image $x \in R^{784}$ and its label $y \in \{0, 1, \ldots, 9\}$, the dataset is divided into training and development sets:

$$X_{train} \in R^{784 \times n_{train}}, \quad Y_{train} \in \{0, 1, \ldots, 9\}^{n_{train}}$$

$$X_{dev} \in R^{784 \times n_{dev}}, \quad Y_{dev} \in \{0, 1, \ldots, 9\}^{n_{dev}}$$

Data normalization is performed by scaling pixel values to the range [0, 1]:

$$X = \frac{X}{255.0}$$

3 Neural Network Architecture

The network architecture consists of an input layer, one hidden layer, and an output layer. The input layer has 784 neurons, the hidden layer has 10 neurons, and the output layer has 10 neurons.

3.1 Weight and Bias Initialization

Weights and biases are initialized using a random distribution:

$$W^{[1]} \sim \mathcal{U}(-0.5, 0.5)^{10\times 784}, \quad b^{[1]} \sim \mathcal{U}(-0.5, 0.5)^{10\times 1}$$

$$W^{[2]} \sim \mathcal{U}(-0.5, 0.5)^{10\times 10}, \quad b^{[2]} \sim \mathcal{U}(-0.5, 0.5)^{10\times 1}$$

4 Forward Propagation

Forward propagation calculates the activations of each layer.

4.1 Hidden Layer

Linear combination:

$$Z^{[1]} = W^{[1]}X + b^{[1]}$$

Activation (ReLU):

$$A^{[1]} = ReLU(Z^{[1]})$$

$$ReLU(z) = \max(0, z)$$

4.2 Output Layer

Linear combination:

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$

Activation (Softmax):

$$A^{[2]} = softmax(Z^{[2]})$$

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{10} e^{z_j}}$$

5 Loss Function

The cross-entropy loss function is used to quantify the difference between predicted probabilities and true labels:

$$\mathcal{L}(A^{[2]}, Y) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{10} y_{ij}\log(A^{[2]}_{ij})$$

6 Backward Propagation

Backward propagation computes gradients to update weights and biases, minimizing the loss function.

6.1 Output Layer Gradients

Error at the output layer:

$$dZ^{[2]} = A^{[2]} - Y$$

Gradients:

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} (A^{[1]})^T$$

$$db^{[2]} = \frac{1}{m} \sum dZ^{[2]}$$

6.2 Hidden Layer Gradients

Error at the hidden layer:

$$dZ^{[1]} = (W^{[2]})^T dZ^{[2]} \odot ReLU'(Z^{[1]})$$

$$ReLU'(z) = \{ 1\ ifz > 00ifz \leq 0$$

Gradients:

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$

$$db^{[1]} = \frac{1}{m} \sum dZ^{[1]}$$

7 Gradient Descent Optimization

Parameters are updated using gradient descent:

$$W^{[l]} = W^{[l]} - \alpha dW^{[l]}$$

$$b^{[l]} = b^{[l]} - \alpha db^{[l]}$$

Where α is the learning rate.

8 Experimental Results

The model is evaluated on the development set using accuracy as the performance metric:

$$Accuracy = \frac{Numberofcorrectpredictions}{Totalnumberofpredictions}$$

Empirical results show that the neural network achieves high accuracy on the MNIST dataset, demonstrating its effectiveness in recognizing handwritten digits.

9 Conclusion

This paper presented a neural network-based approach for digit recognition, detailing the mathematical foundations and implementation. The experimental results confirm the model's efficacy, making it a robust solution for handwritten digit classification.

References

- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE.
- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.