



# SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

A PROJECT REPORT ON

“2 LAYER NEURAL NETWORK WITH THE  
FOLLOWING STRUCTURE:  
LINEAR->RELU->LINEAR->SIGMOID”

NEURAL NETWORKS AND DEEP LEARNING LABORATORY

**TEAM MEMBER:-**

Faizal Hussain

**SRN:-**

R21EH124

**CLASS :-**

AI/DS Section – ‘C’

**FACULTY INCHARGE:-**

Prof. Vivek Sharma S

**DATE OF SUBMISSION:-**

19/04/2024

**Title: Develop the two layer neural network: Use the helper functions you have implemented in the previous assignment to build a 2-layer neural network with the following structure:**

**LINEAR -> RELU -> LINEAR -> SIGMOID**

## **1. Brief Problem Statement:**

This project addresses the challenge of handwritten digit recognition using a two-layer neural network architecture. The objective is to create a model capable of accurately classifying grayscale images of handwritten digits from the MNIST dataset. Handwritten digit recognition holds significant importance across multiple domains, including finance, postal services, education, and data entry systems. By automating the process of digit recognition, this project aims to enhance efficiency, reduce manual labor, and improve the overall accuracy of data processing tasks.

The MNIST dataset consists of 28x28 pixel images of handwritten digits, along with their corresponding numerical labels. Each image is represented as a matrix of pixel intensities, with values ranging from 0 to 255. The task involves training a machine learning model to learn patterns and features from these images, enabling it to accurately predict the digit represented in unseen images.

The complexity of the handwritten digit recognition problem lies in variations in writing styles, stroke thickness, and image quality. Additionally, distinguishing between similar-looking digits presents a significant challenge for classification algorithms. This project aims to overcome these challenges by implementing a two-layer neural network architecture and leveraging techniques such as gradient-based optimization and activation functions like ReLU and sigmoid.

The successful development of an accurate handwritten digit recognition model has the potential to streamline various processes, including check processing, mail sorting, and automated grading systems. It represents a step forward in the advancement of machine learning and computer vision technologies, with practical implications for improving productivity and accuracy in real-world applications.

## **2. Background Information:**

### **a). Domain Information:**

Handwritten digit recognition falls within the domain of machine learning and computer vision. It encompasses techniques for interpreting and categorizing images of handwritten digits into their corresponding numerical representations. This domain is crucial for various industries, including finance, postal services, education, and data entry systems, where accurate digit recognition is essential for automating processes and improving efficiency. Machine learning algorithms, particularly neural networks, play a vital role in solving handwritten digit recognition problems by learning patterns and features from labeled data. Computer vision techniques are employed to preprocess and analyze images, extract relevant features, and make predictions. Overall, the domain of handwritten digit recognition intersects with broader areas of artificial intelligence, pattern recognition, and image processing, driving innovation and advancements in technology across multiple sectors.

### **b). Problem Description and Analysis:**

Handwritten digit recognition involves analyzing grayscale images of handwritten digits and accurately assigning them numerical labels. The task is challenging due to variations in writing styles, stroke thickness, and image quality. Distinguishing between similar-looking digits, such as 3 and 8 or 5 and 6, poses significant challenges for classification algorithms. Additionally, handwriting styles vary across individuals, further complicating the task. The primary goal is to develop machine learning models capable of learning discriminative features from labeled data to accurately classify unseen handwritten digits. This involves preprocessing images, extracting relevant features, and training models using optimization algorithms. The analysis of the problem includes assessing the performance of different machine learning algorithms, exploring strategies to handle class imbalance, and optimizing model hyperparameters to achieve high accuracy and generalization on unseen data. Overall, solving the handwritten digit recognition problem requires robust algorithms capable of learning complex patterns and adapting to variations in handwriting styles.

### **c). Possible Applications:**

Handwritten digit recognition has numerous applications across various industries. In finance, it facilitates check processing, where handwritten amounts need to be accurately interpreted and recorded. Postal services utilize digit recognition systems for sorting mail based on handwritten addresses, optimizing delivery routes, and enhancing efficiency. In education, automated grading systems leverage digit recognition to assess handwritten exams, saving time and reducing human error. Additionally, digit recognition enables the automation of data entry processes in administrative tasks, such as form processing and document management systems. Moreover, it finds applications in authentication systems, where handwritten signatures and alphanumeric codes are verified for security purposes. Overall, the versatility of handwritten digit recognition makes it a valuable tool for improving productivity, accuracy, and automation across a wide range of domains and applications.

## **3. Motivation for the Selection of the Project:**

The motivation for selecting the handwritten digit recognition project stems from its practical significance and potential societal impact. Automating the process of digit recognition has the potential to streamline various tasks across different sectors, improving efficiency and reducing manual labor. By leveraging machine learning techniques to accurately interpret handwritten digits, this project aims to contribute to advancements in automation technology. Additionally, the widespread applications of digit recognition in finance, postal services, education, and data entry systems highlight its practical significance and relevance in addressing real-world challenges. Furthermore, the project offers an opportunity to explore and apply cutting-edge machine learning algorithms and computer vision techniques to solve a fundamental problem in pattern recognition. Overall, the motivation lies in harnessing technology to create innovative solutions that enhance productivity, accuracy, and convenience in everyday tasks.

### **a). Social Impact:**

The social impact of handwritten digit recognition extends to various aspects of daily life, offering benefits such as improved accessibility, enhanced efficiency, and increased accuracy. By automating tasks that rely on digit recognition, such as check processing, mail sorting, and data entry, the technology contributes to greater efficiency in administrative processes. Additionally, in educational settings, automated grading systems powered by digit recognition alleviate the burden on educators, allowing them to focus more on personalized instruction and student support. Moreover, digit recogni-

tion enhances accessibility for individuals with disabilities by enabling the development of assistive technologies, such as devices that convert handwritten text to digital format. Overall, the social impact of handwritten digit recognition lies in its ability to simplify tasks, improve accuracy, and empower individuals by making information more accessible and tasks more manageable.

#### **b). Technological Innovation:**

Handwritten digit recognition represents a significant technological innovation in the fields of machine learning and computer vision. By leveraging advanced algorithms and techniques, such as neural networks and image processing, the technology enables computers to interpret and understand handwritten digits with high accuracy. This innovation has paved the way for automation in various industries, including finance, postal services, and education, where digit recognition plays a crucial role in streamlining processes and improving efficiency. Additionally, the development of robust digit recognition models has spurred research and innovation in related areas, such as optical character recognition (OCR) and document analysis. Furthermore, ongoing advancements in digit recognition technology continue to push the boundaries of what is possible, leading to improvements in accuracy, speed, and adaptability. Overall, the technological innovation in handwritten digit recognition holds promise for revolutionizing numerous applications and driving progress in machine learning and computer vision.

#### **c). Practical Significance:**

The practical significance of handwritten digit recognition lies in its ability to automate tasks that rely on interpreting handwritten information. By accurately identifying and classifying handwritten digits, this technology streamlines various processes across industries, including finance, postal services, education, and data entry systems. For example, in finance, digit recognition automates check processing, reducing errors and speeding up transaction processing times. In postal services, it enables efficient mail sorting based on handwritten addresses, optimizing delivery routes and enhancing delivery speed. Moreover, in education, digit recognition supports automated grading systems, freeing up educators' time for more personalized instruction and feedback. Additionally, in data entry systems, it facilitates the extraction of handwritten information from forms and documents, improving data processing efficiency. Overall, the practical significance of handwritten digit recognition lies in its ability to improve accuracy, efficiency, and productivity in various real-world applications, ultimately benefiting businesses and individuals alike.

### **4. Detailed Dataset Description and Dataset Source:**

The dataset used in this project is the Modified National Institute of Standards and Technology (MNIST) dataset, a widely used benchmark dataset in the field of machine learning and computer vision. The MNIST dataset consists of 70,000 grayscale images of handwritten digits, each measuring 28x28 pixels. These images are accompanied by corresponding numerical labels indicating the digit represented in each image, ranging from 0 to 9. The dataset is divided into two main subsets: a training set and a test set.

The training set comprises 60,000 images, while the test set contains 10,000 images. The images in the training set are used to train machine learning models, while the images in the test set are reserved for evaluating the performance of the trained models on unseen data.

The MNIST dataset is sourced from the National Institute of Standards and Technology (NIST) Special Database 3 and Special Database 1, which contain images of handwritten digits collected from various sources, including high school students and employees of the United States Census Bureau. The dataset was modified and standardized to ensure consistency in image format and labeling.

The MNIST dataset is widely used in research and education for benchmarking machine learning algorithms, particularly for tasks related to image classification and pattern recognition. Its popularity can be attributed to its simplicity, ease of use, and availability, making it an ideal starting point for experimenting with machine learning techniques. Additionally, the MNIST dataset serves as a standard benchmark for comparing the performance of different machine learning models, algorithms, and architectures, providing a common reference point for researchers and practitioners in the field.

## **5. Current Benchmark:**

The current benchmark for handwritten digit recognition tasks is set by achieving high accuracy on the MNIST dataset. While the MNIST dataset is considered a relatively simple dataset compared to more complex image datasets available today, achieving near-perfect accuracy on MNIST remains a significant milestone in the field of machine learning and computer vision.

State-of-the-art machine learning models, including deep neural networks, convolutional neural networks (CNNs), and more advanced architectures, have achieved accuracy rates exceeding 99% on the MNIST dataset. These models leverage sophisticated techniques such as data augmentation, dropout regularization, batch normalization, and advanced optimization algorithms to improve performance.

Achieving high accuracy on the MNIST dataset not only demonstrates the effectiveness of a model's architecture and training methodology but also serves as a baseline for evaluating performance on more challenging datasets. Researchers often use the MNIST benchmark to compare the performance of new models and algorithms or to demonstrate advancements in machine learning techniques.

While MNIST remains a popular benchmark, it is worth noting that some researchers have moved towards more challenging datasets, such as Fashion-MNIST, SVHN, or CIFAR-10, to assess the robustness and generalization capabilities of machine learning models in real-world scenarios.

## **6. Proposed Plan:**

The proposed plan involves implementing a two-layer neural network architecture for handwritten digit recognition using the MNIST dataset.

### **a). Approaches:**

The proposed approach involves implementing a two-layer neural network architecture for handwritten digit recognition. This approach leverages the power of deep learning to learn complex patterns and features from the input data. The neural network consists of two main layers: a linear layer followed by a rectified linear unit (ReLU) activation function, and a second linear layer followed by a sigmoid activation function. This architecture allows the network to capture non-linear relationships between input pixels and output digit labels.

Additionally, the approach involves training the neural network using gradient descent optimization to minimize the cost function, which measures the difference between predicted and actual digit labels. Regularization techniques, such as dropout, may also be employed to prevent overfitting and improve

generalization performance. Overall, the approach combines the strengths of neural networks and optimization algorithms to develop an accurate and efficient handwritten digit recognition system.

### **b). Stages with Defined Deliverables:**

The project will be divided into several stages, each with defined deliverables to ensure systematic progress:

- 1. Data Preparation:** Normalize pixel values and reshape images to prepare the MNIST dataset for model training.
- 2. Model Development:** Implement a two-layer neural network architecture for handwritten digit recognition.
- 3. Training:** Train the neural network using the prepared dataset, optimizing parameters to minimize the cost function.
- 4. Evaluation:** Assess the trained model's performance on both training and test datasets, measuring accuracy and loss metrics.
- 5. Optimization:** Experiment with different hyperparameters, such as learning rate and regularization techniques, to improve model performance.
- 6. Validation:** Validate the final model's accuracy and generalization through cross-validation and comparison with existing benchmarks.

Each stage will have specific deliverables, including pre-processed data, trained model weights, evaluation metrics, and a validated model. These milestones will ensure the project progresses effectively towards developing a robust handwritten digit recognition system.

### **c). Methodology:**

Gradient-based optimization techniques, such as gradient descent, are employed to iteratively update model parameters during the training process. Forward and backward propagation algorithms are utilized for efficient computation of gradients and model updates.

The methodology for developing the handwritten digit recognition system involves several key steps:

#### **1. Data Preprocessing:**

- **Normalize pixel values:** Scale pixel values to a range between 0 and 1 to ensure uniformity and improve convergence during training.
- **Reshape images:** Flatten 28x28 pixel images into 784-dimensional vectors to serve as input features for the neural network.

#### **2. Model Architecture:**

- **Design a two-layer neural network:** Construct a neural network with an input layer, a hidden layer with ReLU activation, and an output layer with sigmoid activation.
- **Initialize parameters:** Randomly initialize weights and biases for each layer to break symmetry and facilitate learning.

### 3. Training:

- **Forward Propagation:** Compute activations for each layer using linear transformations followed by activation functions.
- **Calculate Cost:** Measure the discrepancy between predicted and actual labels using a suitable cost function, such as cross-entropy loss.
- **Backward Propagation:** Compute gradients of the cost function with respect to parameters using backpropagation.
- **Update Parameters:** Update weights and biases iteratively using gradient descent to minimize the cost function.

### 4. Evaluation:

- **Train Set Evaluation:** Assess model performance on the training set to monitor learning progress and detect overfitting.
- **Test Set Evaluation:** Evaluate the trained model's performance on the separate test set to assess generalization capability and measure accuracy.

### 5. Optimization:

- **Hyperparameter Tuning:** Experiment with learning rates, batch sizes, and regularization techniques (e.g., dropout) to optimize model performance.
- **Model Complexity Adjustment:** Adjust the number of hidden units in the neural network to balance between underfitting and overfitting.

### 6. Validation:

- **Cross-Validation:** Validate model robustness and stability through k-fold cross-validation to ensure consistency across different data splits.
- **Benchmark Comparison:** Compare the performance of the developed model with existing benchmarks to validate its effectiveness and competitiveness.

By following this methodology, we aim to systematically develop a robust and accurate handwritten digit recognition system capable of achieving high performance on the MNIST dataset.

## 7. Preliminary Exploratory Data Analysis (EDA):

Preliminary EDA involves conducting in-depth analysis and visualization of the MNIST dataset to gain insights into the distribution of handwritten digits, explore variations in writing styles, and examine pixel intensity distributions. This exploratory phase aids in identifying potential challenges and patterns inherent in the dataset, which can inform subsequent model design and training strategies.

**1. Dataset Overview:** Begin by loading and inspecting the dataset to understand its dimensions, data types, and basic statistics. This step ensures data integrity and provides a foundation for subsequent analysis.

**2. Sample Image Visualization:** Visualize a subset of images from the dataset to gain an intuition for the handwritten digits' appearance and variability. Plotting random samples can reveal common patterns, writing styles, and potential challenges.

**3. Label Distribution Analysis:** Analyze the distribution of digit labels across the dataset to ensure class balance and identify any potential biases or imbalances. Understanding label frequencies is essential for interpreting model performance and identifying potential challenges during training.

**4. Pixel Intensity Distribution:** Explore the distribution of pixel intensities across the dataset to identify any trends or anomalies. Understanding pixel intensity distributions can inform preprocessing decisions, such as normalization or contrast adjustments.

**5. Correlation Analysis:** Investigate correlations between pixel values within and across images to identify patterns or relationships that may impact model performance. Correlation analysis helps identify features relevant for classification tasks and informs feature engineering decisions.

**6. Outlier Detection:** Identify and analyze outliers or anomalies in the dataset that may affect model training and evaluation. Outlier detection techniques, such as clustering or statistical tests, can help identify data points that deviate significantly from the norm.

Preliminary EDA provides valuable insights into the MNIST dataset's characteristics, informs preprocessing decisions, and guides model development strategies. By thoroughly exploring the dataset, researchers can better understand its intricacies and develop more effective machine learning models for handwritten digit recognition.

## **8. Expected Outcomes:**

The expected outcome of this project is the development of an accurate and efficient handwritten digit recognition system using a two-layer neural network architecture. The trained model is anticipated to achieve high performance in classifying handwritten digits from the MNIST dataset.

Specifically, the expected outcomes include:

**1. High Accuracy:** The trained model is expected to achieve a high accuracy rate on both the training and test datasets, accurately classifying handwritten digits with minimal errors.

**2. Generalization:** The developed model is expected to demonstrate good generalization performance, effectively recognizing handwritten digits from unseen data samples beyond the training set.

**3. Robustness:** The model is expected to exhibit robustness to variations in handwriting styles, noise, and other potential sources of variability present in real-world data.

**4. Efficiency:** The trained model is anticipated to be computationally efficient, allowing for fast inference times and scalability to larger datasets or real-time applications.

**5. Interpretability:** The model's decision-making process and learned representations are expected to be interpretable, providing insights into how the neural network classifies handwritten digits.

Overall, the expected outcome is a reliable and effective handwritten digit recognition system that can serve as a foundational tool for various applications, including optical character recognition, document processing, and automated data entry.



## **9. Project Demonstration Strategy:**

The project demonstration strategy involves showcasing the developed handwritten digit recognition system to stakeholders and end-users. The demonstration aims to highlight the system's functionality, performance, and potential applications. Key components of the demonstration strategy include:

- 1. System Overview:** Provide an overview of the handwritten digit recognition system, including its architecture, underlying algorithms, and overall capabilities.
- 2. Live Demonstration:** Conduct a live demonstration of the system in action, allowing stakeholders to interact with the interface and observe real-time digit recognition results. This demonstration can involve uploading handwritten digit images or using a drawing tool to input digits directly.
- 3. Performance Evaluation:** Present performance metrics, such as accuracy rates, error analysis, and inference times, to showcase the system's effectiveness and efficiency.
- 4. Use Case Scenarios:** Demonstrate the system's applicability in various use case scenarios, such as digit recognition in handwritten documents, forms, or online forms. Highlight how the system can streamline processes and improve productivity in these contexts.
- 5. User Experience:** Solicit feedback from stakeholders and end-users regarding their experience with the system, including usability, intuitiveness, and usefulness. Use this feedback to iteratively improve the system's design and functionality.
- 6. Future Enhancements:** Discuss potential future enhancements and extensions to the system, such as integration with other applications, support for additional languages or character sets, and improvements in accuracy and speed.

Overall, the project demonstration aims to showcase the handwritten digit recognition system's value proposition and its potential to address real-world challenges and opportunities in various domains.

## **10. Proposed Timeline for Project Stage Executions:**

- Data Preprocessing and EDA: 1 week
- Model Implementation and Training: 2 weeks
- Evaluation and Fine-tuning: 1 week
- Project Documentation and Report Writing: 1 week

## **11. Team Members Names:**

- **Name:** Faizal Hussain
- **SRN:** R21EH124

## **12. Designated Project Guide:**

- **Name:** Prof. Vivek Sharma S
- **Email:** [viveksharma\\_s@reva.edu.in](mailto:viveksharma_s@reva.edu.in)
- **Contact:** +91 9449473728