SRINIVAS UNIVERSITY
INSTITUTE OF ENGINEERING & TECHNOLOGY

(SUBJECT: ARTIFICIAL NEURAL NETWORKS)

(SUBJECT CODE: 24SBT113)

AN GROUP TASK REPORT ON

"Handwritten Digit Recognition using ANN"

Submitted in the partial fulfillment of the requirements for the First semester

BACHELOR OF TECHNOLOGY
IN
ARTIFICIAL INTELLIGENCE & MACHINE LEARNING
Submitted By,

SANUROOP CK-01SU24AI094
SANJAY SAJAN K-01SU24AI093
THARUN PRADEEPAN-01SU24AI108
ROHAN M-01SU24AI087

UNDER THE GUIDANCE OF

Prof. Mahesh Kumar V B

DEPARTMENT OF ARTIFICIAL INTELLIGENCE &MACHINE LEARNING

SRINIVAS UNIVERSITY INSTITUTE OF ENGINEERING & TECHNOLOGY
MUKKA, MANGALURU-574146

2025-26

SRINIVAS UNIVERSITY
INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CERTIFICATE

This is to certify that, **SANUROOP CK (01SU24AI094), SANJAY SAJAN K (01SU24AI093), THARUN PRADEEPAN (01SU24AI108), ROHAN M (01SU24AI087)** has satisfactorily completed the assessment (Group Task) in **ARTIFICIAL NEURAL NETWORKS (24SBT113)** prescribed by the Srinivas University for the 4th semester B. Tech course during the year **2025-26**.

MARKS AWARDED

Staff In charge

Name: Prof. Mahesh Kumar V B

Date: 23.02.2026

TABLE OF CONTENTS

INTRODUCTION

Artificial Neural Networks (ANNs) are computational models inspired by the structure and functioning of the human brain. Just like biological neurons transmit signals to process information, artificial neurons receive inputs, apply mathematical operations, and produce outputs. ANN is a major branch of Artificial Intelligence (AI) and Machine Learning (ML), widely used in solving complex pattern recognition and classification problems. One of the most significant and practical applications of ANN is Handwritten Digit Recognition.

Handwritten digit recognition refers to the process of identifying and classifying handwritten numerical digits (0–9) from images. Humans can easily recognize handwritten numbers even if the writing style varies from person to person. However, for computers, this task is challenging because handwriting differs in shape, size, thickness, orientation, and stroke pattern. A digit such as “2” written by one person may look completely different when written by another. Therefore, designing a rule-based system to detect every possible variation is almost impossible.

Artificial Neural Networks solve this issue by learning patterns directly from data instead of relying on predefined rules. During the training phase, the ANN is provided with thousands of labeled handwritten digit images. The network learns important features such as curves, edges, intersections, and loops that help distinguish one digit from another. For example, the digit “8” contains two closed loops, whereas “1” generally consists of a straight vertical line. By adjusting its internal weights through a learning process called backpropagation, the ANN gradually improves its prediction accuracy.

One of the most commonly used datasets for handwritten digit recognition is the MNIST dataset, which contains 70,000 grayscale images of handwritten digits. This dataset has become a benchmark for evaluating classification algorithms. Using ANN models on such datasets has achieved very high accuracy, often exceeding 98%, demonstrating the effectiveness of neural networks in pattern recognition tasks.

Handwritten digit recognition has numerous real-life applications. In banking systems, it is used to automatically read cheque amounts and account numbers. In postal services, it helps in recognizing PIN codes written on envelopes. Educational institutions use it for automated grading systems. It is also used in form processing, data entry automation, and mobile digit recognition applications. These applications reduce manual effort, save time, and minimize human errors.

The success of ANN in handwritten digit recognition highlights its ability to model complex, non-linear relationships in data. Unlike traditional programming approaches, where explicit instructions are written for every condition, ANN learns from examples and adapts over time. This adaptability makes neural networks powerful tools in modern intelligent systems.

In conclusion, handwritten digit recognition is a classical yet highly important application of Artificial Neural Networks. It demonstrates how machine learning techniques can replicate human-like pattern recognition capabilities. By using supervised learning and efficient training algorithms, ANN-based systems provide accurate, reliable, and scalable solutions for real-world problems.

PROBLEM STATEMENT

The problem of handwritten digit recognition involves designing a computational model that can accurately identify and classify handwritten numerical digits from 0 to 9 based on image input. Although this task appears simple for humans, it is a complex challenge for machines due to the high variability in handwriting styles. Each individual writes digits differently in terms of size, stroke thickness, alignment, curvature, and orientation. Even the same person may write the same digit differently at different times. Therefore, the primary problem is to develop a system that can generalize well across these variations and correctly recognize unseen handwritten digits.

The input to the system is typically a grayscale image containing a single handwritten digit. For example, in standard datasets like MNIST, each image is of size 28 × 28 pixels. Each pixel represents intensity information ranging from black to white. These pixel values form the numerical representation of the image and serve as input features to the Artificial Neural Network. The challenge lies in converting this raw pixel data into meaningful patterns that help distinguish one digit from another.

The desired output of the system is a classification label representing one of the ten possible digits (0–9). Thus, the problem is formulated as a multi-class classification task. The model must analyze the input image and assign it to the correct digit class with high probability. During training, the correct labels are provided along with the images. This makes it a supervised learning problem, where the network learns by comparing its predictions with the actual target values.

One of the main difficulties in this problem is feature extraction. The system must automatically identify distinguishing characteristics such as straight lines, curves, loops, intersections, and stroke patterns. For example, the digit “0” contains one closed loop, while “8” contains two. The digit “1” is generally a vertical stroke, whereas “7” may have a horizontal line and diagonal stroke. The neural network must learn these subtle differences through training without manual feature engineering.

Another significant challenge is achieving high accuracy while avoiding overfitting. The model should perform well not only on training data but also on new, unseen handwritten samples. This requires proper model design, selection of suitable activation functions, learning algorithms, and optimization techniques. Additionally, preprocessing steps such as normalization and noise removal are essential to improve performance and ensure stable learning.

Computational efficiency is also an important consideration. The system should process images quickly, especially in real-time applications such as bank cheque scanning or automated form reading. Therefore, the network architecture must balance accuracy and speed.

In summary, the problem statement focuses on building an efficient Artificial Neural Network model capable of recognizing and classifying handwritten digits accurately despite variations in writing styles. The system must learn from labeled data, extract meaningful features automatically, minimize classification error, and generalize well to new data. Successfully solving this problem demonstrates the power of ANN in handling complex pattern recognition tasks and forms the foundation for many advanced image recognition systems.

DATASET DESCRIPTION

The performance of any Artificial Neural Network (ANN) model largely depends on the quality and quantity of the dataset used for training and testing. In the case of handwritten digit recognition, the most commonly used and widely accepted dataset is the MNIST dataset. MNIST stands for "Modified National Institute of Standards and Technology" database. It is a benchmark dataset in the field of machine learning and image classification, specifically designed for training and evaluating digit recognition models.

The MNIST dataset contains a total of 70,000 grayscale images of handwritten digits ranging from 0 to 9. Out of these, 60,000 images are used for training the model, while 10,000 images are reserved for testing purposes. Each image in the dataset is of size 28 × 28 pixels, making it relatively small and computationally efficient to process. The small size ensures faster training while still preserving sufficient detail for accurate digit recognition.

Each image is stored as a matrix of 28 rows and 28 columns, where each element represents the intensity value of a pixel. The pixel intensity values range from 0 to 255, where 0 represents black and 255 represents white. Since the digits are written in white on a black background (or vice versa), the contrast allows the neural network to easily distinguish the digit from the background. Before feeding the images into the ANN, these pixel values are typically normalized to a range between 0 and 1. Normalization improves numerical stability and speeds up the learning process.

Another important aspect of the MNIST dataset is its diversity. The handwritten digits were collected from different individuals, including students and employees. This ensures variation in writing styles, stroke thickness, alignment, and orientation. Such diversity makes the dataset realistic and challenging, helping the ANN model learn to generalize well to new handwriting samples.

Each image in the dataset is associated with a label that indicates the correct digit class (0–9). During supervised training, the input image is provided to the neural network along with its corresponding label. The network then learns to map the pixel patterns to the correct output class. For classification purposes, the labels are often converted into one-hot encoded vectors. For example, if the digit is "3", the label is represented as [0, 0, 0, 1, 0, 0, 0, 0, 0, 0].

The dataset is well-balanced, meaning that each digit class has approximately the same number of examples. This prevents bias toward any particular digit during training. A balanced dataset ensures that the model performs uniformly across all classes.

In addition to MNIST, other extended datasets such as EMNIST (Extended MNIST) are also available, which include handwritten letters along with digits. However, MNIST remains the most popular dataset for introductory and experimental digit recognition models.

In conclusion, the MNIST dataset provides a standardized, clean, and well-structured dataset for training and evaluating handwritten digit recognition systems. Its simplicity, diversity, and balanced class distribution make it ideal for developing and testing Artificial Neural Network models effectively.

PREPROCESSING OF DATA

Data preprocessing is a crucial step in building an effective Artificial Neural Network (ANN) model for handwritten digit recognition. Raw image data cannot be directly fed into the neural network without proper preparation. Preprocessing ensures that the input data is clean, standardized, and suitable for efficient training. It improves model performance, speeds up convergence, and reduces errors.

The first step in preprocessing is image acquisition and format standardization. Handwritten digits are typically captured as grayscale images. In grayscale format, each pixel represents an intensity value rather than color information. Since color is not important for digit recognition, converting images to grayscale reduces computational complexity while preserving essential shape details.

The second step is resizing the image. For consistency, all images must have the same dimensions. In the MNIST dataset, each image is standardized to 28×28 pixels. Standardization ensures that the neural network receives uniform input size for every training example. If images vary in size, the network cannot process them efficiently.

The third step is noise removal. Handwritten images may contain unwanted marks, smudges, or background noise. Simple filtering techniques, such as smoothing or thresholding, can be applied to enhance the clarity of the digit. Removing noise improves feature detection and reduces misclassification.

Normalization is another important preprocessing step. Pixel intensity values originally range from 0 to 255. Feeding large numerical values directly into the network may slow down training and cause unstable weight updates. Therefore, pixel values are scaled to a smaller range, typically between 0 and 1, by dividing each value by 255. This process ensures numerical stability and allows faster convergence during training.

Next, the image matrix is flattened into a one-dimensional vector. A 28×28 image contains 784 pixels. Since a simple feedforward neural network accepts a vector input, the two-dimensional matrix is converted into a 784-length vector. Each element in this vector corresponds to one pixel value. This flattened vector becomes the input layer of the ANN.

Another important preprocessing step is label encoding. In supervised learning, each input image must be associated with a correct output label. For multi-class classification, labels are converted into one-hot encoded vectors. For example, if the digit is “5”, it is represented as [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]. One-hot encoding helps the network calculate classification error properly using cross-entropy loss.

Data shuffling is also performed before training. Shuffling ensures that the model does not learn patterns based on the order of data. It improves generalization and prevents bias.

Sometimes, additional preprocessing techniques such as data augmentation are applied. Data augmentation artificially increases the dataset size by slightly rotating, shifting, or scaling images. This helps the model become more robust to variations in handwriting.

ANN ARCHITECTURE

The Artificial Neural Network (ANN) architecture designed for handwritten digit recognition consists of multiple layers of interconnected neurons that work together to classify input images into digit classes (0–9). The architecture typically includes an input layer, one or more hidden layers, and an output layer. Each layer plays a specific role in transforming the input data into meaningful predictions.

Input Layer

The input layer is the first layer of the network and is responsible for receiving the processed image data. In the case of the MNIST dataset, each image is of size 28×28 pixels, which results in 784 total pixel values. These pixel values are flattened into a one-dimensional vector of length 784. Therefore, the input layer consists of 784 neurons, where each neuron represents one pixel intensity value. The input layer does not perform any computation; it simply passes the pixel values to the next layer.

Hidden Layers

Hidden layers are the core components of the ANN. They perform computations and extract meaningful features from the input data. A typical architecture may include two hidden layers. For example:

- Hidden Layer 1: 128 neurons
- Hidden Layer 2: 64 neurons

Each neuron in the hidden layer computes a weighted sum of its inputs, adds a bias term, and applies an activation function. The mathematical representation of a neuron is:

$$y = f\left(\sum w_i x_i + b\right)$$

Where:

- x_i are input values
- w_i are weights
- b is bias
- f is the activation function

The commonly used activation function in hidden layers is ReLU (Rectified Linear Unit), defined as:

$$f(x) = \max(0, x)$$

ReLU introduces non-linearity into the network, allowing it to learn complex patterns such as curves, loops, and intersections in handwritten digits. Without non-linear activation functions, the network would behave like a simple linear model and would not capture complex relationships.

The first hidden layer may detect basic features like edges and lines. The second hidden layer combines these features to recognize more complex patterns such as shapes specific to digits like “3,” “6,” or “8.”

Output Layer

The output layer contains 10 neurons, each corresponding to one digit class (0–9). The activation function used in this layer is Softmax, which converts the raw output values into probabilities. The Softmax function ensures that the sum of all output probabilities equals 1.

For example, the output may look like:

[0.01, 0.02, 0.85, 0.03, 0.02, 0.01, 0.02, 0.02, 0.01, 0.01]

The highest probability (0.85) corresponds to digit “2,” so the model predicts the digit as 2.

Overall Structure

The ANN follows a feedforward structure, meaning information moves in one direction—from input layer to hidden layers to output layer. During training, backpropagation is used to update weights and minimize error.

This multi-layer architecture allows the ANN to automatically extract relevant features and accurately classify handwritten digits. The combination of weighted connections, activation functions, and supervised learning makes the ANN architecture powerful and effective for digit recognition tasks.

HIDDEN LAYERS

Hidden layers are the most important components of an Artificial Neural Network (ANN). They are called "hidden" because their outputs are not directly visible to the user; instead, they operate internally between the input layer and the output layer. In handwritten digit recognition, hidden layers are responsible for learning and extracting meaningful features from raw pixel data.

When a 28×28 pixel image is given as input, the input layer simply passes the 784 pixel values to the first hidden layer. The hidden layer then begins processing this numerical information. Each neuron in the hidden layer performs three main operations: it calculates a weighted sum of the inputs, adds a bias term, and applies an activation function. Mathematically, this can be represented as:

$$y = f\left(\sum w_i x_i + b\right)$$

Here, x_i represents the input values, w_i represents the weights, b is the bias, and f is the activation function.

The weights determine how important each input feature is, while the bias allows flexibility in shifting the activation. Initially, weights are assigned random values. During training, these weights are adjusted using backpropagation to minimize the error between predicted and actual outputs.

In handwritten digit recognition, hidden layers help in detecting important features. The first hidden layer typically learns simple patterns such as edges, straight lines, and basic curves. For example, it may detect vertical strokes (important for digit "1") or curved shapes (important for digits "3" and "5"). As the signal moves to deeper hidden layers, the network starts combining simple features into more complex patterns. The second hidden layer may recognize loops, intersections, or combinations of strokes that uniquely identify digits such as "8" or "9."

Activation functions play a crucial role in hidden layers. The most commonly used activation function is ReLU (Rectified Linear Unit), defined as:

$$f(x) = \max(0, x)$$

ReLU introduces non-linearity into the network, allowing it to learn complex relationships. Without non-linear activation functions, the network would behave like a simple linear equation and would not be able to classify digits accurately.

The number of hidden layers and the number of neurons in each layer directly affect model performance. Too few neurons may lead to underfitting, where the model cannot learn complex patterns. Too many neurons may lead to overfitting, where the model memorizes training data instead of generalizing to new data. Therefore, selecting an optimal architecture is important.

Hidden layers also enable hierarchical learning. Lower layers detect basic features, while higher layers detect abstract patterns. This layered learning process makes ANN powerful for image recognition tasks.

ADVANTAGES OF ANN IN DIGIT RECOGNITION

Artificial Neural Networks (ANNs) provide several significant advantages in handwritten digit recognition compared to traditional rule-based or statistical methods. Their ability to learn patterns directly from data makes them highly effective for solving complex classification problems where variations are large and unpredictable.

One of the main advantages of ANN is its ability to handle variations in handwriting styles. Handwritten digits differ widely among individuals in terms of size, thickness, orientation, spacing, and stroke order. It is extremely difficult to write explicit rules that account for all possible variations. ANN overcomes this limitation by learning features automatically from thousands of training examples. Through repeated exposure to diverse samples, the network adapts and recognizes patterns regardless of personal writing style.

Another major advantage is automatic feature extraction. In traditional image processing systems, engineers manually design features such as edge detectors or shape descriptors. However, ANN eliminates the need for manual feature engineering. Hidden layers in the network automatically identify important characteristics such as lines, curves, loops, and intersections. This reduces development effort and improves accuracy.

High accuracy is another key benefit. When trained properly on large datasets like MNIST, ANN models can achieve accuracy levels of 98–99%. This level of performance makes them reliable for real-world applications such as bank cheque processing and postal code recognition. The use of backpropagation and optimization algorithms ensures that the model continuously improves by minimizing classification error.

ANNs are also highly adaptable and scalable. If more training data becomes available, the model can be retrained to improve performance further. Additionally, the same architecture can be extended to recognize not only digits but also letters, symbols, or even complete handwritten words. This scalability makes ANN a flexible solution for expanding applications.

Another important advantage is generalization capability. A well-trained ANN does not simply memorize training data; it learns underlying patterns that allow it to classify new, unseen handwritten samples accurately. This ability to generalize is essential for practical applications where new handwriting styles are constantly encountered.

ANN systems also support real-time processing. Once trained, the model can quickly classify input images within milliseconds. This is especially useful in automated systems such as ATM cheque readers, form scanners, and mobile recognition applications, where fast response time is critical.

Robustness to noise is another advantage. Handwritten images may contain smudges, distortions, or background noise. ANN models can still perform well because they focus on important patterns rather than small irregularities. With proper preprocessing and sufficient training data, the network becomes tolerant to minor distortions.

CONCLUSION

Handwritten Digit Recognition using Artificial Neural Networks (ANN) is one of the most important and practical applications of machine learning in the field of pattern recognition. This system demonstrates how computational models inspired by the human brain can successfully perform tasks that traditionally required human intelligence. Recognizing handwritten digits is challenging due to the large variations in writing styles, shapes, sizes, and stroke patterns. However, ANN provides an efficient and accurate solution to this problem through supervised learning and feature extraction.

Throughout this project, we analyzed how a neural network processes handwritten digit images step by step. The system begins with image acquisition and preprocessing, where raw images are standardized, normalized, and converted into numerical form. The preprocessed pixel values are then provided to the input layer of the ANN. Hidden layers play a critical role in extracting important features such as edges, curves, loops, and intersections. By applying activation functions like ReLU and using weighted connections, the network learns complex patterns from the training data.

The output layer, using the Softmax activation function, converts computed values into probability distributions and identifies the digit with the highest probability. The learning process is guided by backpropagation and gradient descent algorithms, which adjust weights to minimize classification error. With sufficient training data such as the MNIST dataset, the ANN model can achieve high accuracy, often above 98%.

One of the major strengths of ANN in digit recognition is its ability to automatically learn features without manual programming. Unlike traditional rule-based systems, which require explicit instructions for every variation, ANN adapts and improves by learning from examples. This makes it highly flexible and scalable for real-world applications. The same architecture can be extended to recognize letters, symbols, and even complex handwritten text.

Handwritten digit recognition has numerous practical applications in banking, postal services, automated form processing, and educational systems. It reduces manual workload, increases processing speed, and minimizes human errors. In banking systems, for example, cheques can be processed automatically by recognizing account numbers and amounts. In postal services, automatic sorting systems rely on digit recognition to identify PIN codes efficiently.

Despite its advantages, ANN-based systems require sufficient training data and computational resources. Proper preprocessing and model tuning are essential to avoid overfitting and ensure good generalization. However, with modern hardware and optimization techniques, these challenges can be effectively managed.

In conclusion, Handwritten Digit Recognition using Artificial Neural Networks highlights the power and effectiveness of deep learning in solving complex classification problems. By combining mathematical models, data-driven learning, and efficient optimization algorithms, ANN provides a reliable and scalable solution for real-world digit recognition tasks. This application not only demonstrates the fundamentals of neural networks but also lays the foundation for advanced image recognition and artificial intelligence systems