A Novel Method for Handwritten Digit Recognition System

PROBLEM SOLUTION FIT

Problem Statement: Handwritten Digit Recognition

MNIST ("Modified National Institute of Standards and Technology") is considered an unofficial computer vision "hello-world" dataset. This is a collection of thousands of handwritten pictures used to train classification models using Machine Learning techniques.

As a part of this problem statement, we will train a multi-layer perceptron using TensorFlow to recognize the handwritten digits.

SOLUTION:

MNIST Dataset Description

Because everyone in the world has a unique writing style, handwriting identification is one of the fascinating research projects now being conducted. It is the ability of a computer to automatically recognise and comprehend handwritten numbers or letters. Every aspect of life is being digitalized to lessen the need for human labour as a result of advancements in science and technology. Thus, handwritten digit recognition is required in many real-time applications. The MNIST data collection, which contains 70000 handwritten digits, is frequently utilised for this recognition method. In order to train these photos and create a deep learning model, we use artificial neural networks. Web application is created where the user can uploadan image of a handwritten digit. this image is analysed by the model and the detected result is returned on to UI

The MNIST Handwritten Digit Recognition Dataset contains 60,000 training and 10,000 testing labelled handwritten digit pictures.

Each picture is 28 pixels in height and 28 pixels wide, for a total of 784 (28×28) pixels. Each pixel has a single pixel value associated with it. It indicates how

bright or dark that pixel is (larger numbers indicates darker pixel). This pixel value is an integer ranging from 0 to 255.
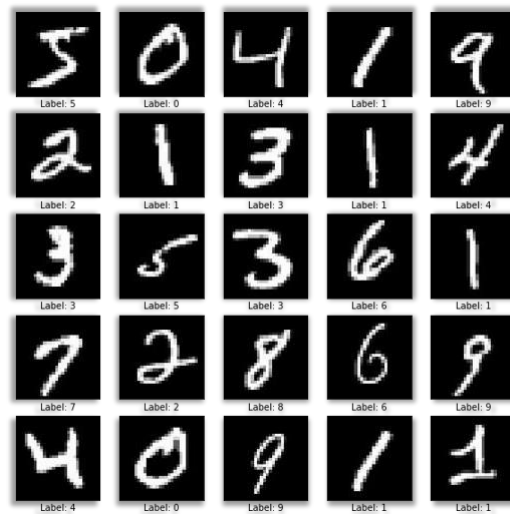

PROCEDURE

- Install the latest TensorFlow library.
- Prepare the dataset for the model.
- Develop Single Layer Perceptron model for classifying the handwritten digits.
- Plot the change in accuracy per epochs.
- Evaluate the model on the testing data.
- Analyse the model summary.
- Add hidden layer to the model to make it Multi-Layer Perceptron.
- Add Dropout to prevent overfitting and check its effect on accuracy.
- Increasing the number of Hidden Layer neuron and check its effect on accuracy.
- Use different optimizers and check its effect on accuracy.
- Increase the hidden layers and check its effect on accuracy.
- Manipulate the batch size and epochs and check its effect on accuracy.

A dataset that is frequently used for handwritten digit recognition is MNIST. 10,000 test photos and 60,000 training images make up the dataset. Artificial neural networks, which are a crucial component in the field of image processing, can most closely resemble the human brain.

Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of handwritten digits.

We have taken this a step further where our handwritten digit recognition system not only detects scanned images of handwritten digits but also allows writing digits on the screen with the help of an integrated GUI for recognition.
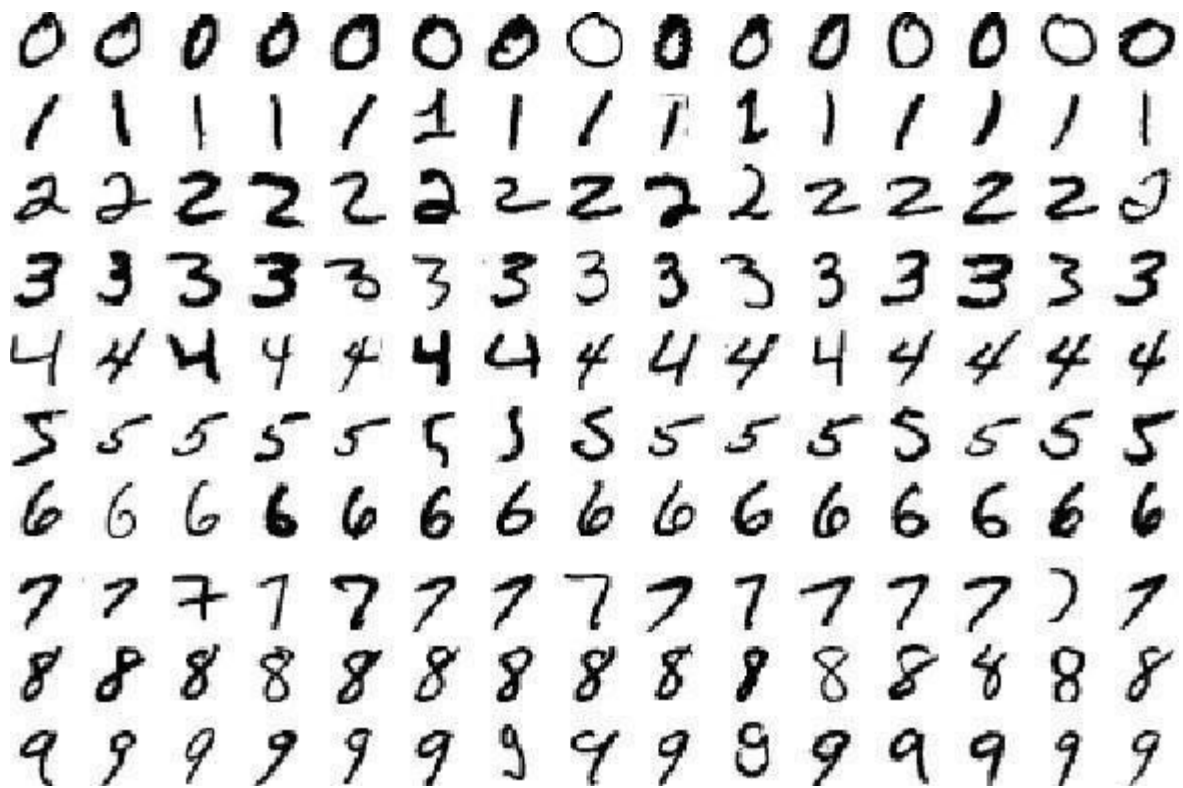

Approach:

We will approach this project by using a three-layered Neural Network.

- **The input layer:** It distributes the features of our examples to the next layer for calculation of activations of the next layer.
- **The hidden layer:** They are made of hidden units called activations providing nonlinear ties for the network. A number of hidden layers can vary according to our requirements.
- **The output layer:** The nodes here are called output units. It provides us with the final prediction of the Neural Network on the basis of which final predictions can be made.

A neural network is a model of the brain's operations. It is made up of numerous layers with a variety of activations; these activations mimic the neurons in our brain. An attempt is made by a neural network to learn a set of parameters from a set of data that might aid in understanding the underlying relationships. Since neural networks are capable of adapting to changing input, the network can produce the best outcome without having to change the output criterion.

METHODOLOGY:

We have implemented a Neural Network with 1 hidden layer having 100 activation units (excluding bias units). The data is loaded from a .mat file, features(X) and labels(y) were extracted. Then features are divided by 255 to rescale them into a range of [0,1] to avoid overflow during computation. 10,000 testing cases and 60,000 training examples make up the data. With the training data, feedforward is used to calculate the hypothesis, and backpropagation is then used to lower the error between the layers. To combat overfitting, the regularisation parameter lambda is set to 0.1. To identify the model that fits the situation the optimizer runs for 70 times.
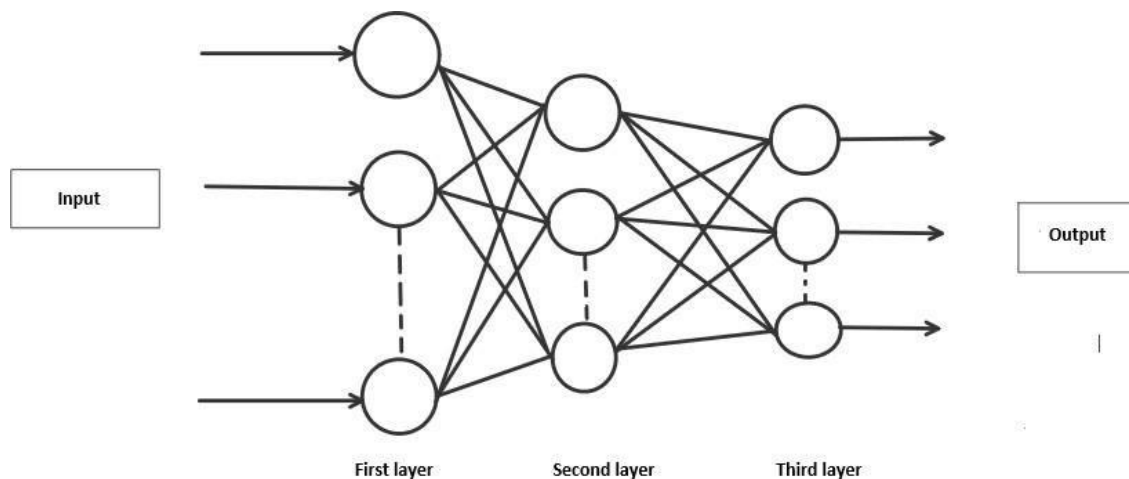

ALGORITHM:

Forward Propagation Architecture:

It is a small workflow of how CNN module will extract the features and classify the image based on it. The architecture shows the input layer, hidden layers and output layer of the network. There are many layers involved in the feature extraction phase of the network which involves convolution and subsampling .

EXPLANATION OF THE PROPOSED SYSTEM

• The first layer of the architecture is the User layer. User layer will comprise of the people who interacts with the app and for the required results.

• The next three layers is the frontend architecture of the application.

The application will be developed using which is the open-source platform for HTML, CSS and JavaScript.

The application is deployed in the localhost which is shown on the browser. Through the app, the user will be able to upload pictures of the handwritten digits and convert it into the digitalized form. • The one in between the database and view layer is the business layer which is the logical calculations on the basis of the request from the client side. It also has the service interface. • The backend layer consists of two datasets: Training Data and Test Data. The MNIST database has been used for that which is already divided into training set of 60,000 examples and test of 10,000 examples. • The training algorithm used is Convolution Neural Network. This will prepare the trained model which will be used to classify the digits present in the test data. Thus, we can classify the digits present in the images as: Class 0,1,2,3,4,5,6,7,8,9.
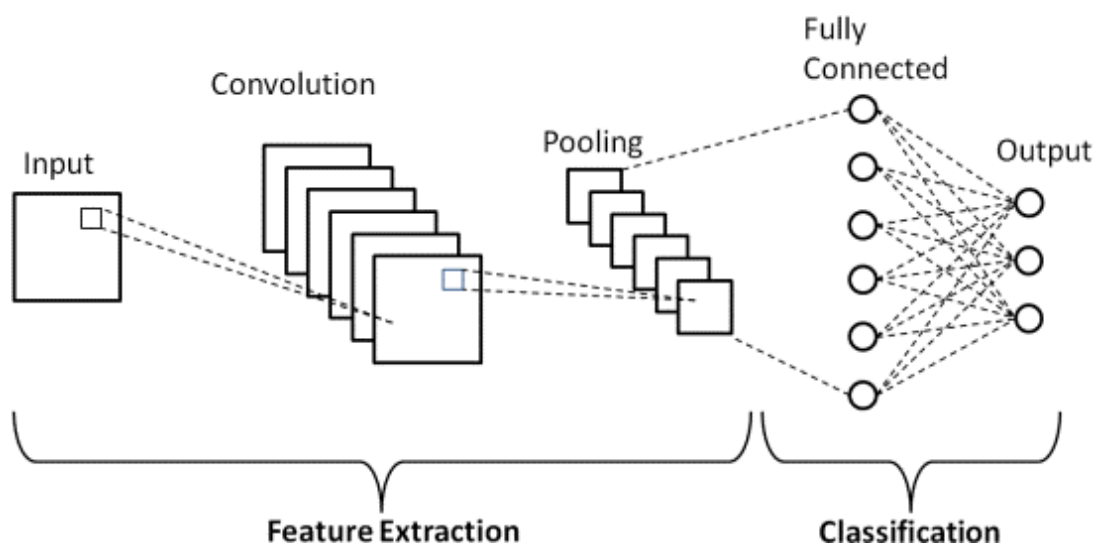
WORKING

• Neural Networks receive an input and transform it through a series of hidden layers.

• Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer.

• Neurons in a single layer function completely independently. • The last fully connected layer is called the "output layer".

Convolution Layer: The foundational component of a CNN is the convolutional layer. The parameters of the layer are a set of learnable filters (or kernels) that cover the entire depth of the input volume but have a narrow receptive field.

During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2- dimensional activation map of that filter.

As a result, the network learns filters that activate when they see some specific type of feature at some spatial position in the input..

Feature Extraction:

All neurons in a feature share the same weights .In this way all neurons detect the same feature at different positions in the input image. Reduce the number of free parameters.

Subsampling Layer: Subsampling, or down sampling, refers to reducing the overall size of a signal .The subsampling layers reduce the spatial resolution of each feature map. Reduce the effect of noises and shift or distortion invariance is achieved.

Pooling layer: It is common to periodically insert a Pooling layer in-between successive Conv layer in a Convent architecture. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation.

TensorFlow: TensorFlow is an open-source machine learning library for research and production. TensorFlow offers APIs for beginners and experts to develop for desktop, mobile, web, and cloud. See the sections below to get started. By scanning the numerical digit and convert into png format using python3 command in terminal we can get text output and sound output.
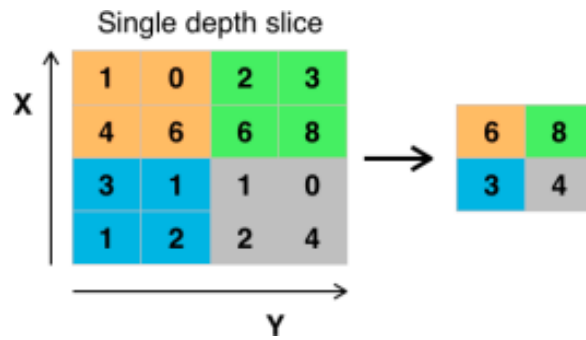
Pooling layer

RESULT:

As with any work or project taken up in the field of machine learning and image processing, we are not considering our results to be perfect.

There is always opportunity for improvement in your methods because machine learning is a topic that is continually developing; there will always be a fresh new idea that solves a given problem more effectively. Three models were used to test the application: Multi-Layer Perceptron (MLP), Convolution Neural Network, and (CNN). We obtain a different classifier accuracy with each model, indicating which is superior.