SOLUTION ARCHITECTURE

Project : A Novel Method for Handwritten Digit Recognition System

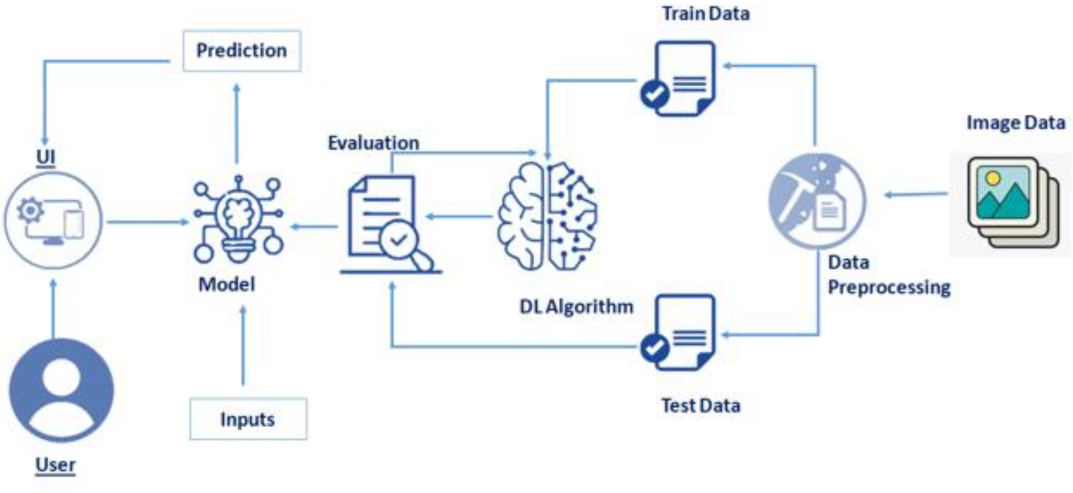
Team Members: Praveenkumar R, Rajapandiyan P, Sanjay Kannan K, Pushpanathan M

PROJECT DESCRIPTION:

Since everyone has a different writing style, handwriting identification is one of the most fascinating research projects now being conducted. It is the ability of a computer to recognise and comprehend handwritten numbers or letters automatically. Science and technology advancements have led to the digitalization of everything, which helps to minimise the need for human labour.

As a result, many real-time applications demand handwritten digit identification. In this recognition method, the MNIST data collection, which contains 70000 handwritten digits, is frequently used. We employ artificial neural networks to train these images and produce a deep learning model. A web application is created that enables users to upload pictures of handwritten numbers.

TECHNICAL ARCHITECTURE:



SOLUTION:

MNIST Dataset Description :

10,000 test handwritten digit images and 60,000 training handwritten digit images make up the MNIST Handwritten Digit Recognition Dataset. The total number of pixels in each image is 784 (2828), with a height of 28 pixels and a width of 28 pixels. A single pixel value connects every pixel. It displays the brightness or darkness of that pixel (larger numbers indicates darker pixel). The integer for this pixel value ranges from 0 to 255.

PROCEDURE:

1. Install the latest TensorFlow library.
2. Prepare the dataset for the model.
3. Develop Single Layer Perceptron model for classifying the handwritten digits.
4. Plot the change in accuracy per epochs.
5. Evaluate the model on the testing data.
6. Analyse the model summary.
7. Add hidden layer to the model to make it Multi-Layer Perceptron.
8. Add Dropout to prevent overfitting and check its effect on accuracy.
9. Increasing the number of Hidden Layer neuron and check its effect on accuracy.
10. Use different optimizers and check its effect on accuracy.
11. Increase the hidden layers and check its effect on accuracy.

12. 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.

Using the MNIST dataset, handwritten digit recognition is a significant effort that was created with the use of neural networks. In essence, it recognises the scanned copies of handwritten numbers.

Our handwritten digit recognition technology goes a step further by allowing you to write numbers on the screen and have them recognised using an integrated GUI in addition to recognising scanned photos of handwritten numbers.

Approach:

This project will be approached utilising a three-layered neural network.

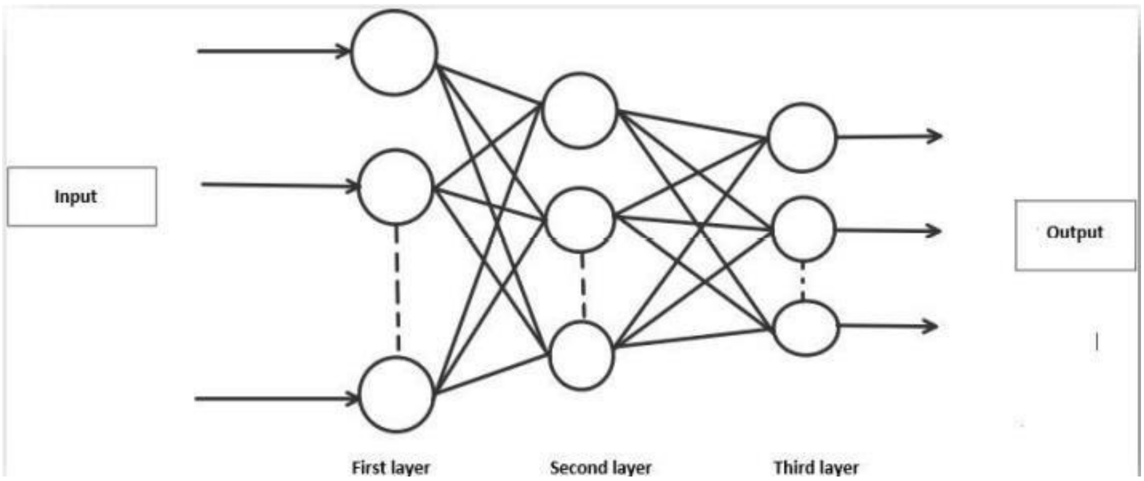
- **The input layer**: It distributes the features from our sample layers to the following layer so that the subsequent layer's activations can be calculated.
- **The hidden layer**: They consist of concealed components called activations that offer the network nonlinear linkages. Depending on our needs, there can be a variety of concealed layers.
- **The output layer**: Here, the nodes are referred to as output units. It gives us access to the neural network's final prediction, which may be used to make final predictions.

A neural network is a representation of the way the brain works. It has many layers and various activations, which resemble the neurons in our brain. An attempt by a neural network to learn a set of parameters from a batch of data may help identify underlying connections. Without needing to reconsider the output criteria, neural networks can offer the greatest outcomes since they can adjust to changing input.

METHODOLOGY:

We built a neural network with 100 activation units and one hidden layer (excluding bias units). A.mat file is used to load the data, after which features (X) and labels (Y) are extracted. The characteristics are then scaled down to a range of [0,1] and split by 255 to prevent calculation overflow. 10,000 testing cases and 60,000 training examples make up the data.

The training set is used to derive the hypothesis, and backpropagation is then utilised to lessen the error between the layers. The regularisation parameter lambda is changed to 0.1 to combat overfitting. To choose the model with the best fit, the optimizer is run 70 times.



ALGORITHM:
Forward Propagation Architecture:

This is a succinct explanation of how the CNN module will extract features from the image and categorise it using those features. The design shows the input layer, hidden layers, and output layer of the network. Convolution and resampling are two of the many layers that are used in the network's feature extraction stage.

Explanation of given system:

- The User layer is the top layer of the architecture. People will make up the user layer.who interacts with the app and for the required results.
- The frontend architecture of the application is comprised of the following three levels. The application will be created on the open-source JavaScript, CSS, and HTML platform. The localhost, which is displayed in the browser, is where the programme is deployed. The user will be able to upload images of the handwritten numbers to the app to have them digitalized.
- The business layer, which consists of logical calculations based on the client's request, sits between the database and view layers. The service interface is also included.
- Training Data and Test Data make up the backend layer's two datasets. The training set, which consists of 60,000 cases, and the test set, which consists of 10,000 examples, have already been separated into the MNIST database.
- A convolution neural network is utilised as the training algorithm. By doing this, the trained model will be ready to be used to categorise the digits found in the test data. As a result, the digits in the photos can be categorised as Class 0,1,2,3,4,5,6,7,8,9.

WORKING :

- Neural networks process input through a number of secret layers after receiving it.
- Each group of neurons in a hidden layer is completely connected to every other neuron in the layer below it.
- Neurons in a single layer function completely independently.
- The "output layer" is the final layer to be fully connected.

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. Each filter is convolved across the width and height of the input volume during the forward pass, computing the dot product between each filter entry and the input to create a two-dimensional activation map of the filter. As a result, the network picks up filters that turn on when they spot a certain kind of feature at a particular location in the input.

Feature Extraction:

Each neuron in a feature has the same weight. All neurons in the input image recognise the same feature in this way, regardless of where it is located. Don't have too many unconstrained parameters.

Subsampling Layer:

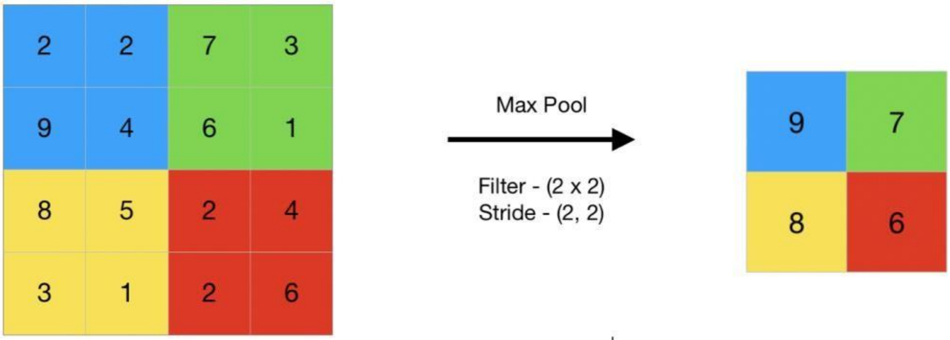
Reducing the overall size of a signal is referred to as subsampling, sometimes known as down sampling. Each feature map's spatial resolution is decreased by the subsampling layers. Shift or distortion invariance is attained, and the impact of sounds is lessened.

Pooling layer:

In a Convent architecture, it is typical to sporadically introduce a Pooling layer between succeeding Conv layers. In order to decrease the number of parameters and computation in the network and, as a result, control overfitting, it gradually shrinks the spatial size of the representation. Every depth slice of the input is independently processed by the Pooling Layer, which then applies the MAX operation to resize each slice spatially.

TensorFlow:

In a Convent architecture, it is typical to sporadically introduce a Pooling layer between succeeding Conv layers. In order to decrease the number of parameters and computation in the network and, as a result, control overfitting, it gradually shrinks the spatial size of the representation. Every depth slice of the input is independently processed by the Pooling Layer, which then applies the MAX operation to resize each slice spatially.



RESULT:

We do not consider our results to be flawless after processing, as with every study or effort in the field of machine learning and image recognition. There is always space for improvement in your method because machine learning is a topic that is constantly developing. Additionally, there will always be new approaches that yield better results for the same problems. The application was sent in.

Multi-Layer Perceptron (MLP), Convolution Neural Network (CNN), and Network models were employed (CNN). Depending on the model that shows which is best, the classifier's accuracy varies.