SOLUTION ARCHITECTURE

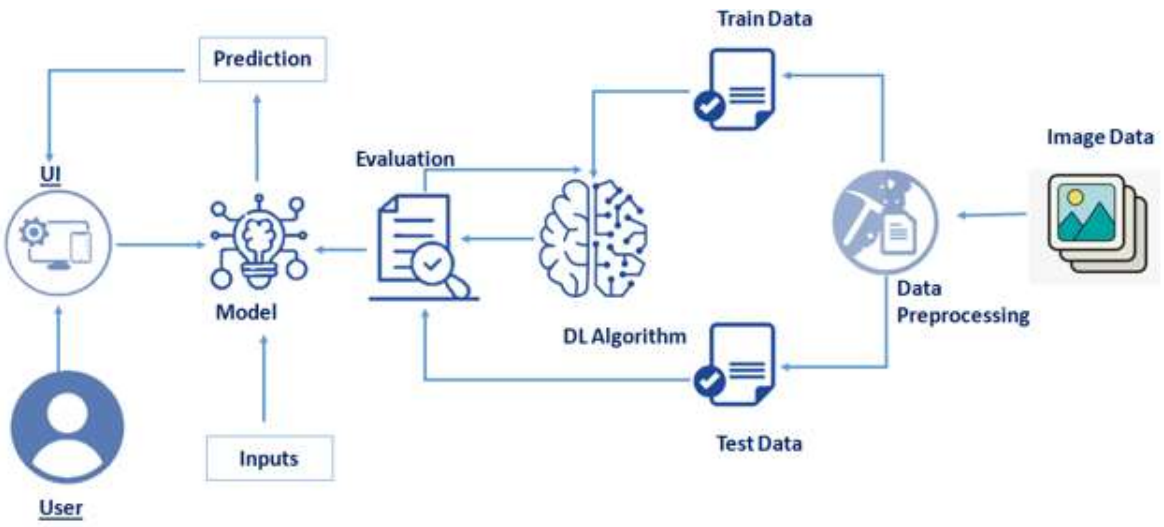
PROJECT : A Novel Method for Handwritten Digit Recognition System

Team Members: Aditi Chakraborty,Akilesh C,Chidhambarasamy,Deepika

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.

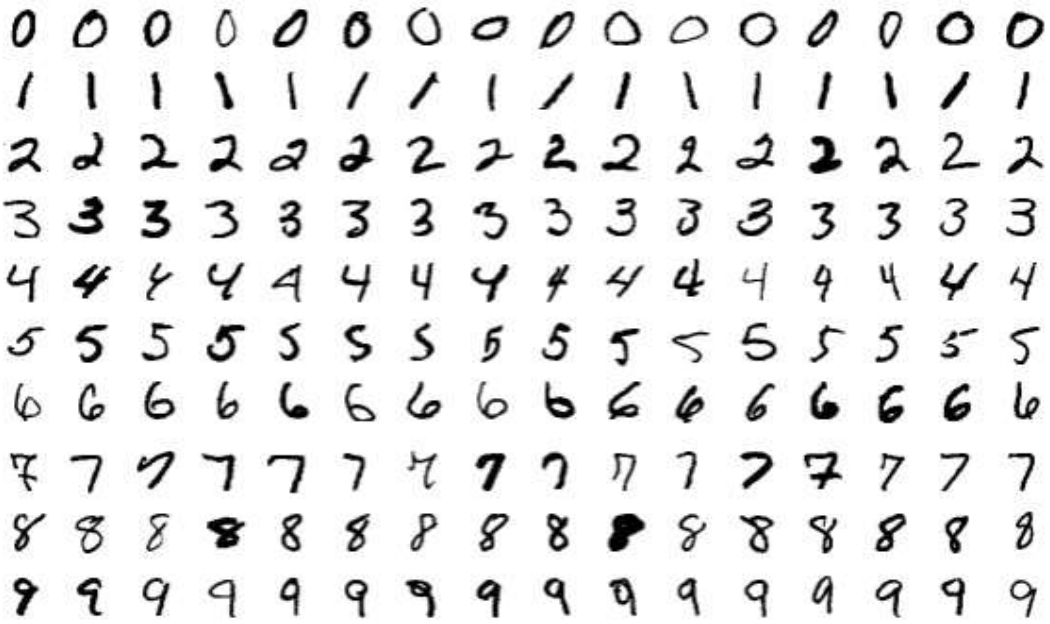
TECHNICAL ARCHITECTURE :



SOLUTION:

MNIST Dataset Description :

There are 60,000 training and 10,000 testing handwritten digit images in the MNIST Handwritten Digit Recognition Dataset. Per image is 784 (2828) pixels in size and has a height and width of 28 pixels each. A single pixel value links every pixel together. It tells whether a pixel is bright or dark (larger numbers indicates darker pixel). This pixel value is a handwritten digit image with an integer value between 0 and 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.

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. We've gone a step further, and our handwritten digit recognition technology not only reads scanned photos of handwritten numbers, but also enables you to write numbers on the screen and have them read by an integrated GUI.

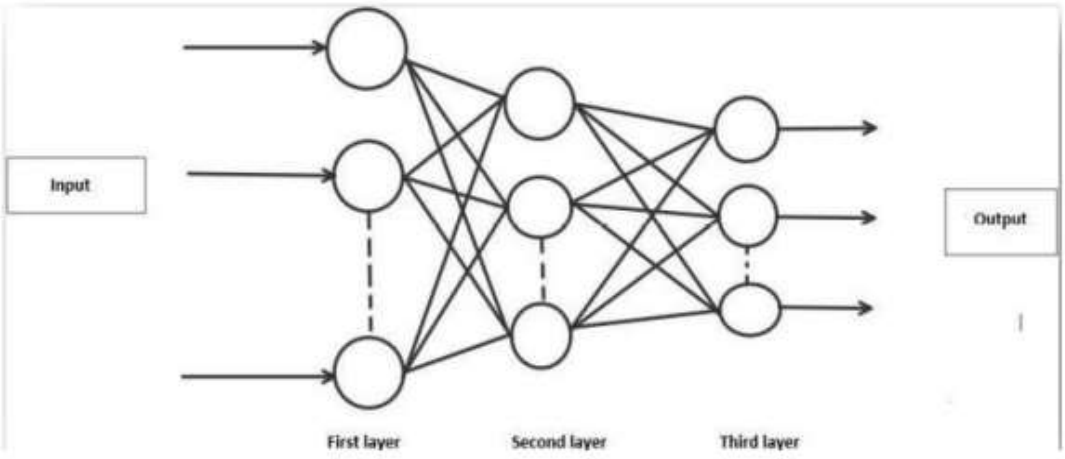
Approach:

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

- **The input layer:** It transfers the characteristics from our example layers to the following layer so that the subsequent layer's activations can be calculated.
- **The hidden layer:** These ties for the network are built up of hidden units known as activations. Depending on our needs, there can be a variety of concealed layers.
- **The output layer :** The nodes in this layer 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.

METHODOLOGY:

We built a neural network with 100 activation units (but not bias units) and one hidden layer. 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. Overfitting is prevented by lowering the regularisation parameter lambda to 0.1. 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.

- The user layer is the initial layer of the architecture, as explained by the example system. The users who engage with the programme and get the desired outcomes make up the user layer.
- The frontend architecture of the application is made up 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. It also includes 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 :

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 the filter entries and the input to create a two-dimensional activation map of that 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 weights during feature extraction. In this manner, the same feature is recognised by all neurons at various locations in the input image. Limit the number of unrestricted parameters.

Layer for Subsampling:

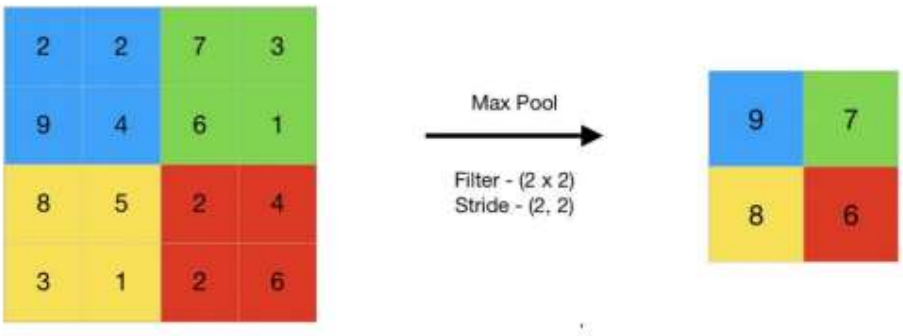
Reducing the overall size of a signal is referred to as subsampling, sometimes known as down sampling.Each feature map's spatial resolution is decreased by the subsampling layers. Shift or distortion invariance is attained, and the impact of sounds is lessened.

Pooling layer:

In a Convent architecture, it is typical to sporadically introduce a Pooling layer between subsequent 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. The Pooling Layer functions separately on

TensorFlow :

TensorFlow is a free machine learning package that may be used for both study and production. TensorFlow provides developers of all skill levels with APIs for desktop, mobile, web, and cloud applications. To get started, refer to the sections below. We can achieve text output and sound output by scanning the number digit and converting it to png format using the python3 command in terminal.



RESULTS:

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.