SOLUTION ARCHITECTURE

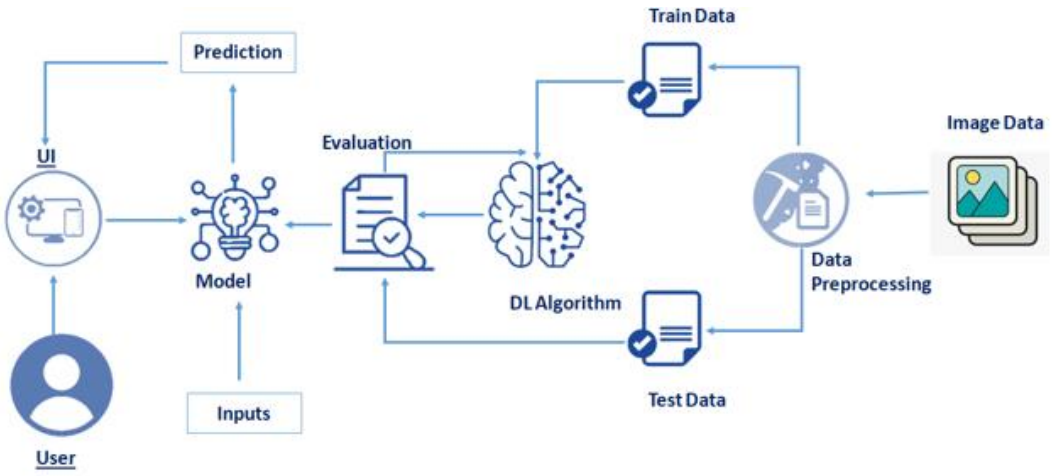
A Novel Method for Handwritten Digit Recognition System

Team Members: NaveenKumar P,Pradesh K,Dev Vinoth S,Prawin Kumar T

PROJECT DESCRIPTION:

Given that everyone in the world has their writing style, handwriting detection is one of the most intriguing research projects now underway. The computer can automatically recognize and understand handwritten figures or letters. Because of advances in science and technology, everything is being digitalized to reduce human effort. As a result, handwritten digit identification is required in many real-time applications. The MNIST data collection, which contains 70000 handwritten digits, is commonly employed in this recognition process. To train these photos and create a deep-learning model, we use artificial neural networks. A web application is developed that allows the user to upload an image of a handwritten digit.

TECHNICAL ARCHITECTURE:



SOLUTION:

MNIST Dataset Description:

The MNIST Handwritten Digit Recognition Dataset includes 60,000 training and 10,000 testing handwritten digit images. Each image has a height of 28 pixels and a width of 28 pixels, for a total of 784 (2828) pixels. Each pixel is connected with a single pixel value. It indicates how bright or dark that pixel is (larger numbers indicate a darker pixel). This pixel value is an integer 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 epoch.

5. Evaluate the model on the testing data.

6. Analyse the model summary.

7. Add a 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 neurons and checking its effect on accuracy.

10. Use different optimizers and check their effect on accuracy.

11. Increase the hidden layers and check their effect on accuracy.

12. Manipulate the batch size and epochs and check its effect on accuracy.

MNIST is a dataset that is widely used for handwritten digit recognition. The dataset consists of 60,000 training images and 10,000 test images. Artificial neural networks can all most mimic the human brain and are a key ingredient in the image-processing field. Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Networks. It detects scanned images of handwritten digits. We've taken it a step further, and our handwritten digit recognition technology not only recognizes scanned images of handwritten numbers but also allows you to write digits on the screen and have them recognized using an integrated GUI.

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 the calculation of activations of the next layer.

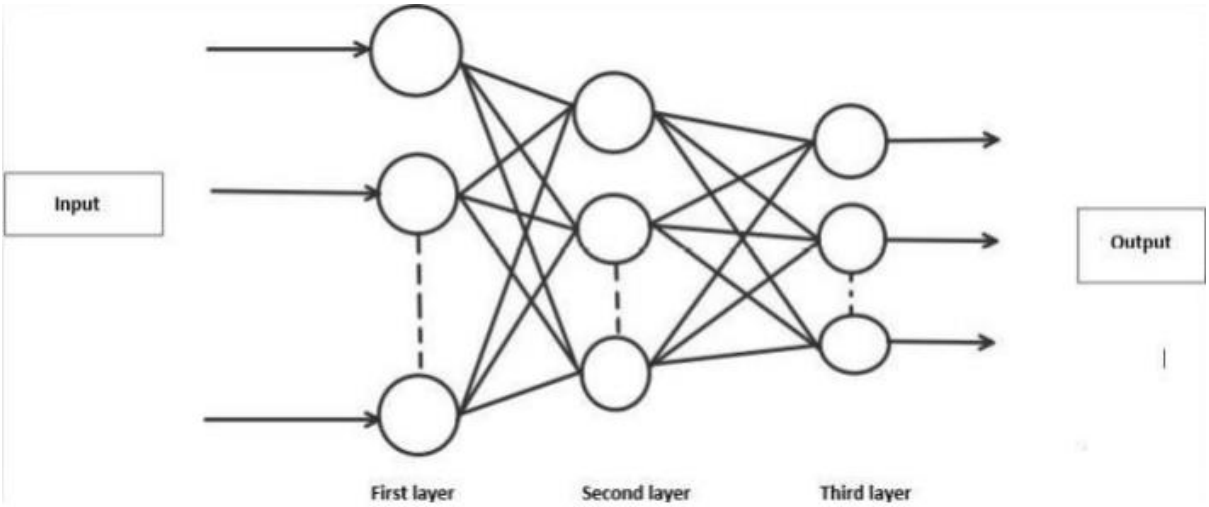
• The hidden layer: They are made of hidden units called activations providing nonlinear ties for the network. The number of hidden layers can vary according to our requirements.

• The output layer: The nodes here are called output units. It provides us with the final prediction of the Neural Network based on which final predictions can be made.

A neural network is a model based on how the brain functions. It is made up of several layers with numerous activations, which mirror neurons in our brain. A neural network attempts to learn a set of parameters from a set of data, which may aid in recognizing underlying links. Because neural networks can adapt to changing input, they can produce the best possible results without having to rethink the output criteria.

METHODOLOGY:

We created a Neural Network with one hidden layer and 100 activation units (excluding bias units). Data is loaded from the a.mat file, then features (X) and labels (Y) are extracted. Then, to avoid overflow during computation, features are divided by 255 and rescaled into a range of [0,1]. The data is divided into 60,000 training instances and 10,000 testing examples. Feedforward is used with the training set to calculate the hypothesis, followed by backpropagation to reduce the error between the layers. To solve the issue of overfitting, the regularisation parameter lambda is adjusted to 0.1. The optimizer is run 70 times to get the best fit model.



ALGORITHM:

Forward Propagation Architecture:

It is a brief description of how the CNN module will extract features and categorize the image based on them. The network's input layer, hidden layers, and output layer are depicted in the design. The feature extraction phase of the network involves multiple layers, including convolution and resampling.

Explanation of the given system:

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

• The next three layers are the frontend architecture of the application. The application will be developed using which is an 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 handwritten digits and convert them into a digitalized form.

• The one in between the database and view layer is the business layer which is the logical calculations based on the request from the client side. It also has a 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 a training set of 60,000 examples and a 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 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 Convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field but extend through the full depth of the input volume. 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:

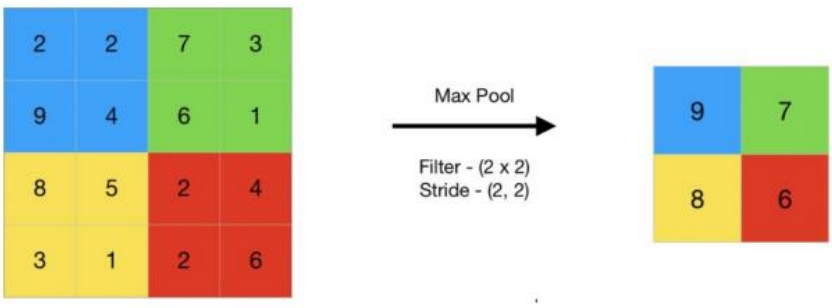
All neurons in a feature share the same weights. In this way, all neurons detect the same feature at different positions in the input image. Reduce the number of free parameters.

Subsampling Layer:

Subsampling, or downsampling, 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 layers 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 the python3 command in the terminal we can get text output and sound output.

RESULT:

As with any study or project was undertaken in the field of machine learning and image recognition, We do not consider our results to be perfect after processing. Machine learning is a field that is always evolving, and there is always room for advancement and improvement in your process; there will always be something new strategy that produces superior outcomes for the same challenge The application has been submitted. Three models were used: Multi-Layer Perceptron (MLP), Convolution Neural Network (CNN), and Network (CNN). The accuracy of the classifier varies depending on the model which demonstrates which is superior.