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

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep-learning model.

The web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned to UI The [MNIST Handwritten Digit Recognition Dataset](#) contains 60,000 training and 10,000 testings labeled 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 indicate darker pixel). This pixel value is an integer ranging from 0 to 255.
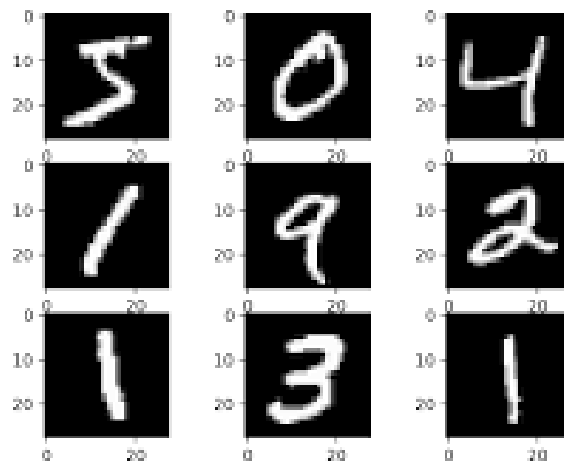
PROCEDURE

- Install 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.

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 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 inspired by how the brain works. It consists of multiple layers having many activations, this activation resembles neurons of our brain. A neural network tries to learn a set of parameters in a set of data which could help to recognize the underlying relationships. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria.

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, and 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. Data is split up into 60,000 training and 10,000 testing examples. Feedforward is performed with the training set for calculating the hypothesis and then backpropagation is done to reduce the error between the layers. The regularization parameter lambda is set to 0.1 to address the problem of overfitting. The optimizer is run for 70 iterations to find the best fit model.
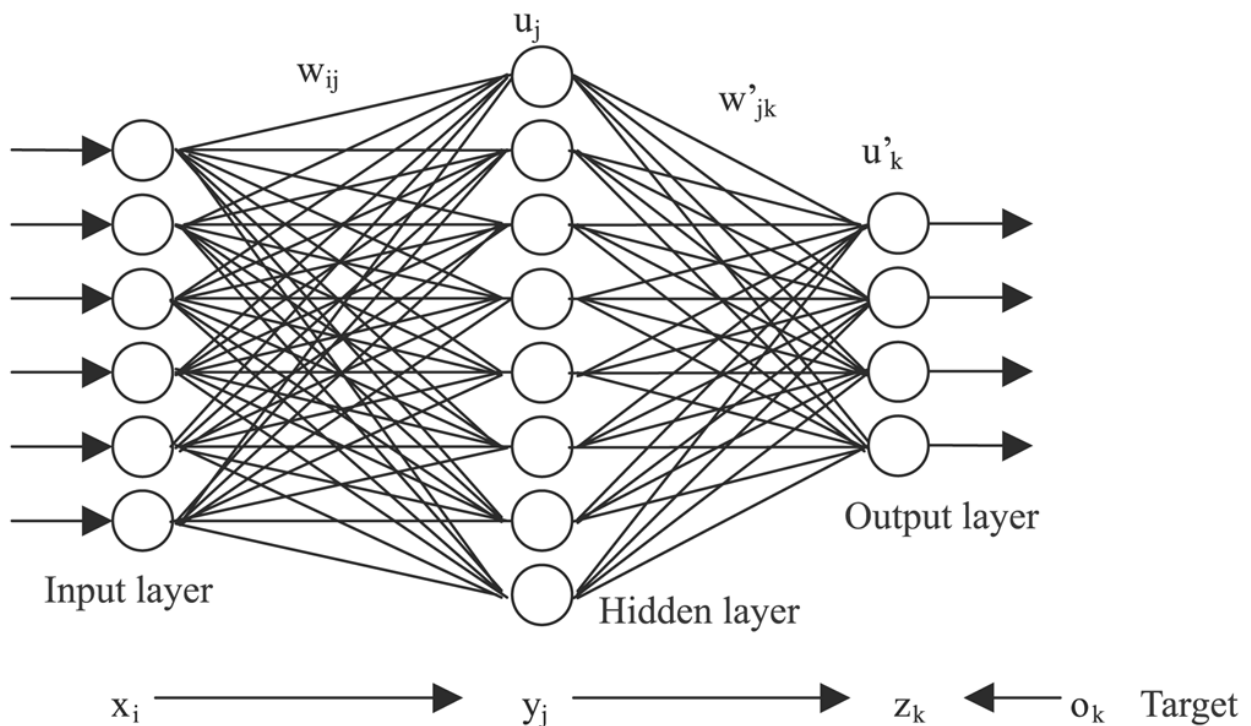
ALGORITHM:

Forward Propagation Architecture:

It is a small workflow of how the CNN module will extract the features and classify the image based on them. 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 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.
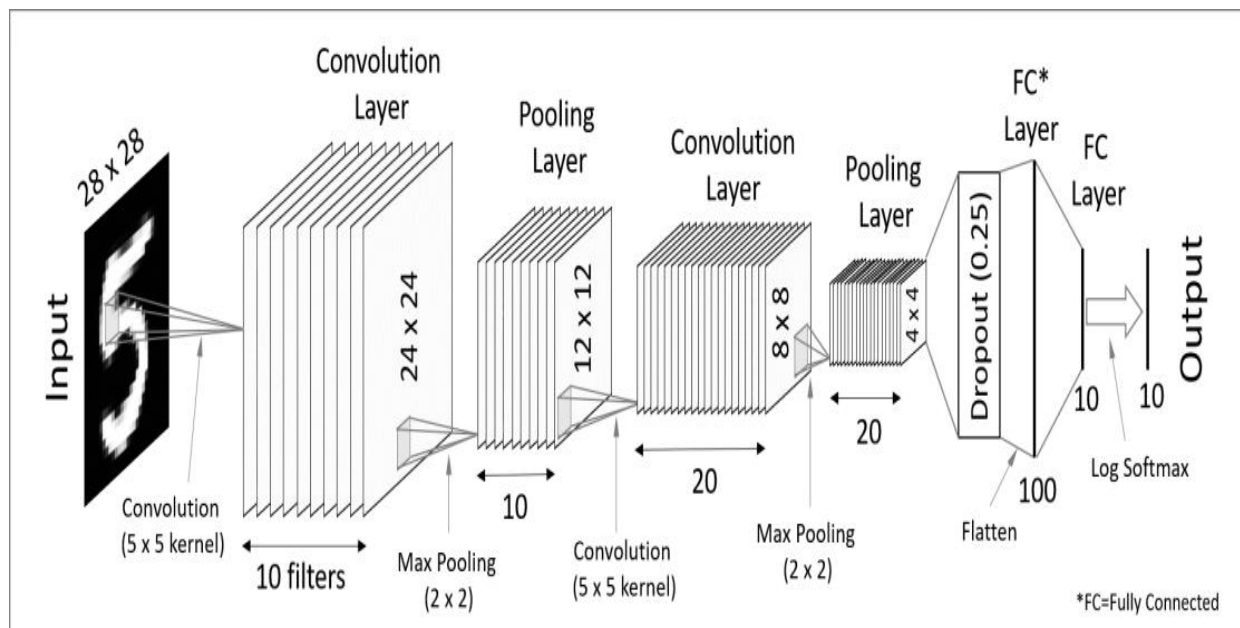
WORKING

• Neural Networks receive 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 fullyconnected 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.
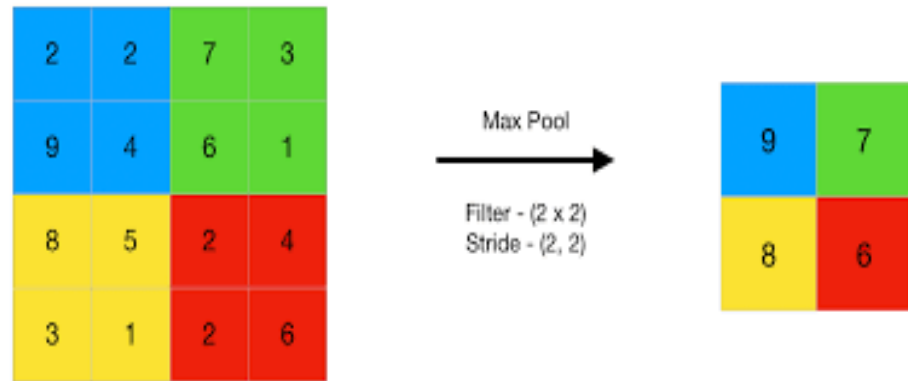
RESULT:

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

Machine learning is a constantly evolving field and there is always room for improvement in your methodology; there is always going to be another new approach that gives better results for the same problem. The application has been tested using three models: Multi-Layer Perceptron (MLP), and Convolution Neural Network (CNN). With each model, we get a different accuracy of the classifier which shows which one is better.

SOLUTION ARCHITECTURE: