

Project Design Phase-I

Solution Architecture

Date	19 September 2022
Team ID	PNT2022TMID18280
Project Name	A Novel Method for Handwritten Digit Recognition
Maximum Marks	4 Marks

Solution Architecture:

Introduction

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.

Neural Networks

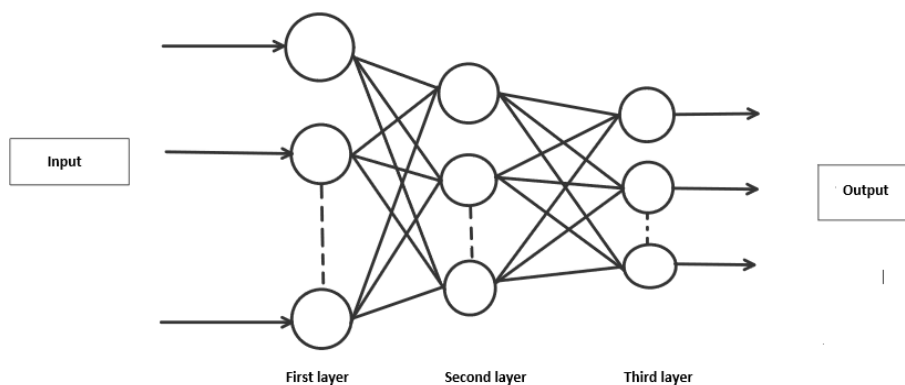
Artificial Neural networks, usually called neural networks (NNs), are interconnected systems composed of many simple processing elements (neurons). It consists of multiple layers having many activations, this activation resembles neurons of our brain. A neural network tries to learn a set of parameters 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

Multi-layered Perceptron

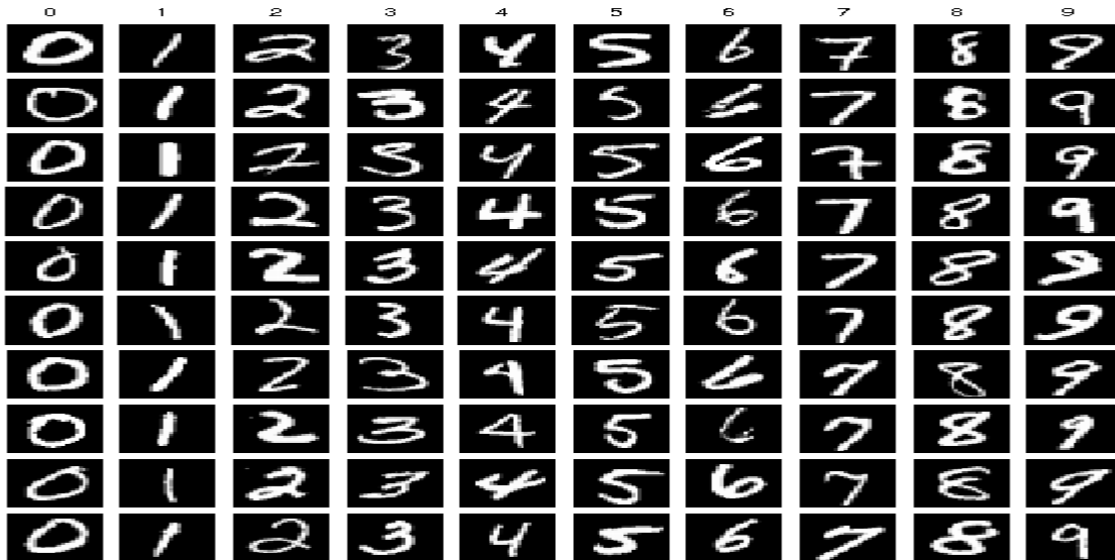
The three-layered neural network is given by,

- **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.



Dataset

MNIST dataset (Modified National Institute of Norms and Technology database) is the subset of the NIST dataset which is a combination of two of NIST's databases Special. It is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9. The sample image in MNIST dataset,



Convolutional Neural Network

CNN is a deep learning algorithm that is widely used for image recognition and classification. It is a class of deep neural networks that require minimum pre-processing. It inputs the image in the form of small chunks rather than inputting a single pixel at a time, so the network can detect uncertain patterns (edges) in the image more efficiently. CNN contains 3 layers namely, an input layer, an output layer, and multiple hidden layers which include Convolutional layers, Pooling layers(Max and Average pooling), Fully connected layers (FC), and normalization layers. CNN uses a filter (kernel) which is an array of weights to extract features from the input image. CNN employs different activation functions at each layer to add some non-linearity. As we move into the CNN, we observe the height and width decrease while the number of channels increases. Finally, the generated column matrix is used to predict the output.

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function.

Convolutional Layer

In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size $M \times M$. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter ($M \times M$).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

The convolution layer in CNN passes the result to the next layer once applying the convolution operation in the input. Convolutional layers in CNN benefit a lot as they ensure the spatial relationship between the pixels is intact.

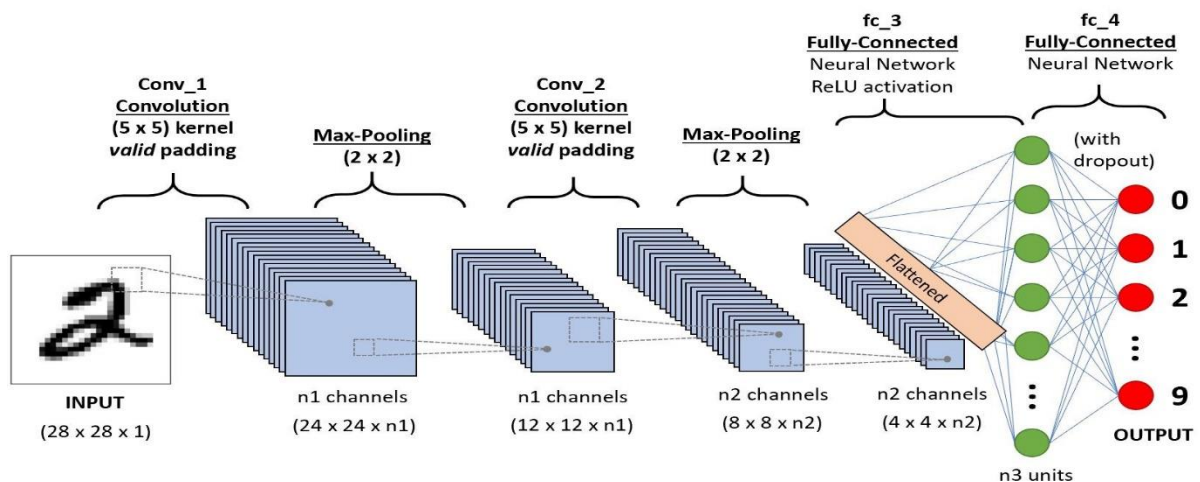
Pooling Layer

Pooling layers reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, tiling sizes such as 2×2 are commonly used. Global pooling acts on all the neurons of the feature map. There are two common types of pooling in popular use: max and average. Max pooling uses the maximum value of each local cluster of neurons in the feature map, while average pooling takes the average value.

Fully Connected Layer

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is the same as a traditional multilayer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

CNN Architecture



The proposed model contains the four stages in order to classify and detect the digits:

- A. Pre-processing
- B. Segmentation
- C. Feature Extraction
- D. Classification and Recognition

Preprocessing

Noise filtering, smoothing and standardization are to be done in this stage. Binarization changes over a gray scale image into a binary image.

Segmentation

Once the pre-processing of the input images is completed, sub-images of individual digits are formed from the sequence of images. Pre-processed digit images are segmented into a sub-image of individual digits, which are assigned a number to each digit. Each individual digit is resized into pixels. In this step an edge detection technique is being used for segmentation of dataset images.

Feature Extraction

Feature Extraction: After the completion of pre-processing stage and segmentation stage, the pre-processed images are represented in the form of a matrix which contains pixels of the images that are of very large size. In the feature extraction stage redundancy from the data is removed.

Classification and Recognition

In the classification and recognition step the extracted feature vectors are taken as an individual input to the classifiers.

Implementation Steps

1. Import the libraries and load the dataset
2. Preprocess the data
3. Create the model
4. Train the model
5. Evaluate the model
6. Create GUI to predict digits

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.

Application

The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, in mobile phones etc.

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. 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), Convolution Neural Network (CNN). With each model we get a different accuracy of the classifier which shows which one is better.