Handwritten Digit Recognition

Abstract

Handwritten digit recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition includes in postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize hand written digits and which is submitted by users by the way of a scanner, tablet, and other digital devices. This project presents an approach to off-line handwritten digit recognition based on different machine learning technique. The main objective of this projectis to ensure effective and reliable approaches for recognition of handwritten digits. In this work, we propose a technique to recognize handwritten digit using deep learning approaches like Convolutional Neural Network (CNN).This is a Digit Recognition System.

Introduction

Automatic digit recognition is a process that converts scanned document images into electronically understandable format. Thus, enabling computers to recognize digit present in images. The latest advancements in technology have highlighted the need for robust methods of automatic digit recognition.There are techniques which have been implemented for digit recognition as discussed in next section but there was a need of more complete and modern architecture for recognition. Thus,this article uses deep learning concepts for digit recognition.

Many artificial intelligence tasks can be solved by identifying the right set of features, and then providing these features to classifier. For example, estimating the size of speaker's vocal tract is a useful feature for speaker identification, estimating pressure points and pen up and down movements are useful feature for online handwriting recognition. However, for many tasks, it is difficult to identify the right set of features. The solution to this problem is deep learning, also called end-to end learning. It is called end-to-end learning because feature extraction and classification phase is automatically done,unlike traditional machine learning, where features are to be explicitly specified. Deep architectures have provided to solutions to some well known problems of pattern recognition which are mental load classification, speech recognition, document recognition, object detection, scene classification, pedestrian detection etc.

Methos

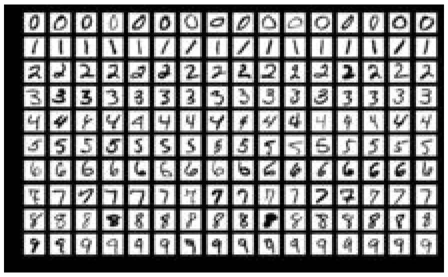
Goals

1. Get a proper insight into the target product/element.
2. Proper classify for a better understanding of the situation about the general public.
3. Getting real-time classification of images

Data Source and Datasets:

The MNIST dataset is an acronym that stands for the Modified National Institute of Standards and Technology dataset.

It is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9.

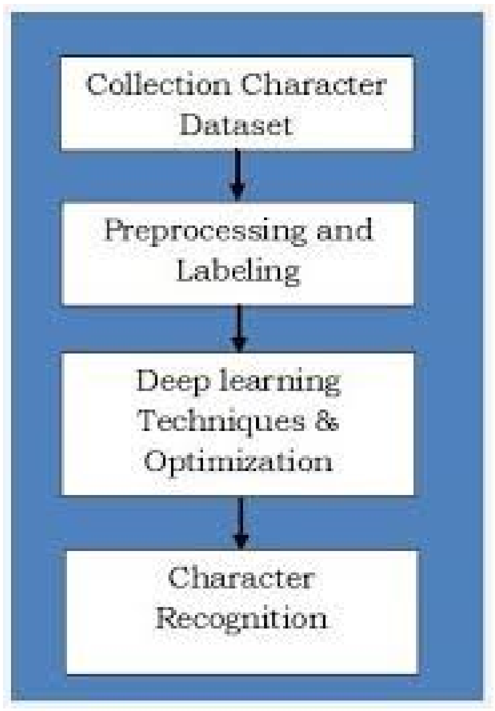

sample image of mnist dataset

- Data Source :
    - http://yann.lecun.com/exdb/mnist/
        - Number Of images: 60000
        - Pixel of images: 28x28

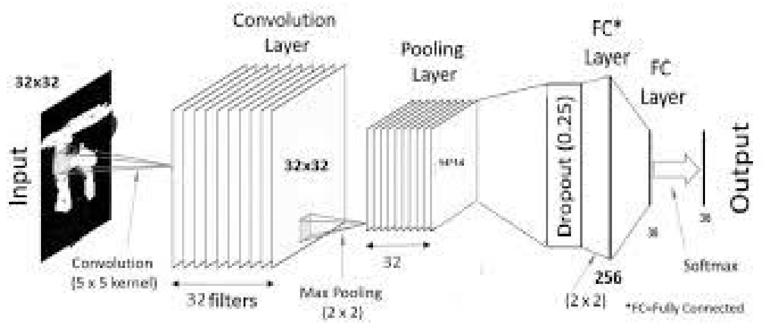
- Dataset Link :
    - http://yann.lecun.com/exdb/mnist/

Convolutional neural network

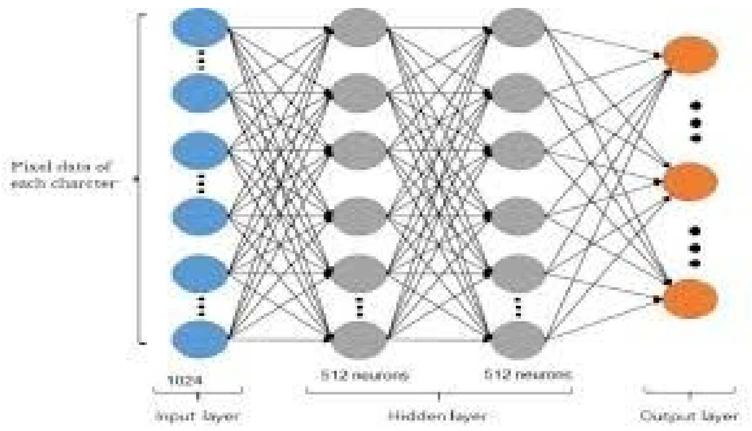
Computer Vision and pattern recognition is a major growing field in area of image processing. In that Convolutional Neural Network (CNNs) plays major role in computer vision. CNN is working on many applications in Image Classification and it is the core of most Computer Vision and pattern recognition systems today, from automatic tagging of photo in Face books to self-driving cars, recognizes digits, alpha-numerals, traffic signal boards, and the other object class[7]. We used five layered Convolutional Neural Networks (CNN) model. On them one layers for convolutional, one layers for max pooling or sub sampling, one Flatten layer which converts 2D array into 1D array and finally two fully connected layers for classification. The initial layer is convolutional (Conv2D) layer has 32 output mapping and the next max pooling layer has 14 output mapping.


Collection Character Dataset
Preprocessing and Labeling
Deep learning Techniques & Optimization
Character Recognition

Block diagram proposed handwritten digit recognition system



The overall structural design of the CNN Model of our proposed system with different layer.



The overall structural design of the DFFNN Model of our proposed system with different layer.

Code Requirements

1. you can install Conda for python which resolves all the dependencies for machine learning.

2. install tensorflow , in conda -> $ conda install tensorflow

3. install keras , in conda -> $ conda install keras

4. install opencv , in conda -> $ conda install opencv

5. digit3.py require data set 'mnist.pkl.gz' for training and test

6. digit3l.h5 model for classify the digits.

Technique Used

I have used convolutional neural networks. I am using Tensorflow as the framework and Keras API for providing a high level of abstraction.

Architecture:

CONV2D --> MAXPOOL --> CONV2D --> MAXPOOL -->FC -->Softmax--> Classification

Method :

Prepare Pixel Data:

We know that the pixel values for each image in the dataset are unsigned integers in the range between black and white, or 0 and 255.

The *prep_pixels()* function implements these behaviors and is provided with the pixel values for both the train and test datasets that will need to be scaled

Model:

The model has two main aspects: the feature extraction front end comprised of convolutional and pooling layers, and the classifier backend that will make a prediction

For the convolutional front-end, I start with a single convolutional layer with a small filter size (3,3) and a modest number of filters (32) followed by a max pooling layer. The filter maps can then be flattened to provide features to the classifier.

Require an output layer with 10 nodes in order to predict the probability distribution of an image belonging to each of the 10 classes.

Evaluate Model:

The model will be evaluated using five-fold cross-validation. The value of *k=5* was chosen to provide a baseline for both repeated evaluation and to not be so large as to require a long running time. Each test set will be 20% of the training dataset, or about 12,000 examples, close to the size of the actual test set for this problem

The *evaluate_model()* function below implements these behaviors, taking the defined model and training dataset as arguments and returning a list of accuracy scores and training histories that can be later summarized.

Save Final Model:

By save final model , to make prediction we did not train the model further in future. By using our saved model we can do predictions.

saving and loading a Keras model requires that the h5py library is installed on workstation.

Make Prediction:

Now I can use my saved model to make a prediction on new images.

The model assumes that new images are grayscale, that they have been aligned so that one image contains one centered handwritten digit, and that the size of the image is square with the size 28×28 pixels.

For prediction save image in current working directory with the filename '*samp le_image.png*'.

Python Implementation

1.Dataset- MNIST dataset

2.Images of size 28 X 28

3.Convolutional Network Support added.

Experimental Outcomes :

- **Experimental Values :**
    - Training : training data of mnist with 10 Epochs
    - Test Set Size : test data of mnist
    - Accuracy: 98.730%

Conclusion : I implement a large multilayer neural network for human handwritten digits. I train the CNN with cross entropy using error back propagation to obtain optimal weights values. I also find useful applications for the CNN I generate. By doing this project, I have practiced using what I have learned in the Machine Learning course. However, I have for the first time observed the practical problems of using the powerful Neural Networks, for example, designing the architecture of a CNN, choosing appropriate activation functions for each layer, error back propagation, convergence issues, stopping criteria, generalization ability. . . This experience will definitely be helpful for my future.