Using deep neural networking concepts to recognize handwritten digits

Nisha Gadhe
March 2018

Handwritten Digits Recognization

[Proposal & The domain background — the field of research where the project is derived;]

Handwriting recognition (or HWR) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. The image of the written text may be sensed "off line" from a piece of paper by optical scanning (optical character recognition) or intelligent word recognition. Alternatively, the movements of the pen tip may be sensed "on line", for example by a pen-based computer screen surface, a generally easier task as there are more clues available.

In computer vision the most difficult task is to recognize the handwritten digit. Since the last decade the handwritten digit recognition is gaining more and more fame because of its potential range of applications like bank cheque analysis, recognizing postal addresses on postal cards, etc. Handwritten digit recognition plays a very vital role in day to day life, like in a form of recording of information and style of communication even with the addition of new emerging techniques.

Handwritten digit recognition has become one of the most challenging tasks in the emerging field of image processing and pattern recognition. Digits which are written in various styles, shapes are easy for human to understand but it is a very complex task for the machine as the handwritten style is changing from person to person. Although handwritten digit recognition is gaining popularity because there are so many potential application areas which it's being used such as to recognize zip code or postal code, mail sorting, bank cheque processing, vehicle license plate recognition.

Digit recognition is basically a process of detecting and recognition digits from input image and convert it into appropriate machine editable forms, but constructing a system for this type of recognition faces a challenging task for the researchers due to the various types of shapes of digits that includes a large set with curves, loops, thickness, orientation size, and may depend upon writer, width, color, etc. There are many digit pairs which are similar in shape but some digits are similarity between them such as 1 and 7, 8 and 9, 5 and 6 and sometimes the digit would be written in different ways.

Since 2009, the recurrent neural networks and deep feedforward neural networks developed in the research group of Jurgen Schmidhuber at the Swiss AI Lab IDSIA have won several international handwriting competitions. In particular, the bi-directional and multi-dimensional Long short-term memory (LSTM) of Alex Graves et al. won three competitions in connected handwriting recognition at the 2009 International Conference on Document Analysis and Recognition (ICDAR), without any prior knowledge about the three different languages (French, Arabic, Persian) to be learned. Recent GPU-based deep learning methods for feedforward networks by Dan Ciresan and colleagues at IDSIA won the ICDAR 2011 offline Chinese handwriting recognition contest; their neural networks also were the first artificial pattern recognizers to achieve human-competitive performance on the famous MNIST handwritten digits problem of Yann LeCun and colleagues at NYU.

The performance of Handwritten digit recognition system is highly depend upon two things: First it depends on feature extraction techniques which is used to increase the performance of the system and improve the recognition rate and the second is the neural network approach which takes lots of training data and automatically infer the rule for matching it with the correct pattern.

Problem Statement

[a problem being investigated for which a solution will be defined;]

The goal is to predict the digit handwritten in an image. The dataset contains images of handwritten digits and their respective numbers which can be used to train and test the model.

This is supervised learning problem more specifically a classification problem. The model should be able to classify which number (0-9) is handwritten in the image. The model can be scored for its ability to predict the number correctly over large different test data and real data.

The datasets and inputs

[data or inputs being used for the problem;]

The MNIST dataset, a subset of a larger set NIST, is a database of 70,000 handwritten digits, divided into 60,000 training examples and 10,000 testing samples. The images in the MNIST dataset are present in form of an array consisting of 28x28 values representing an image along with their labels. The digits have been size-normalized and centered in a fixed-size image.

The dataset can be downloaded and loaded using Keras, a high-level neural networks python library using the following code

```
from keras.datasets import mnist
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

With these lines of code we load all the MNIST data into 4 variables:

- X_train is a tensor of shape (60000, 28, 28) containing, for each training grayscale image, the value of each pixel which is a int in [0;255]
- y_train is a tensor of shape (60000, 1) containing, for training test image, the label of each image which is an int in [0;9]
- X_test is a tensor of shape (10000, 28, 28) containing, for each test grayscale image, the value of each pixel which is a int in [0;255]
- y_test is a tensor of shape (10000, 1) containing, for each test image, the label of each image which is an int in [0;9]

Thus, we can use the above variables to train the model with pixel-values of the handwritten image as features(X_train) and its respective label(y_train) as target. We can also test the model by evaluating it against the test variables X_test and y_test.

Solution Statement

[a the solution proposed for the problem given]

Giventhat the problem is a supervised learning problem and more specifically a classificatio problem, thereare lot of algorithms available for training aclassifier to learn from the data. The algorithm chosen for this project is Deep Neural Network(DNN) using Convolutional Neural Network(ConvNet).

- **(A). A Convolutional Neural Network (CNN)**

CNN is a type of feed-forward Artificial Neural Network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex.

Convolutional Neural Networks consist of neurons that have learnable weights and biases. Each neuron receives some input, performs a dot product and optionally follows it with a non-linearity.

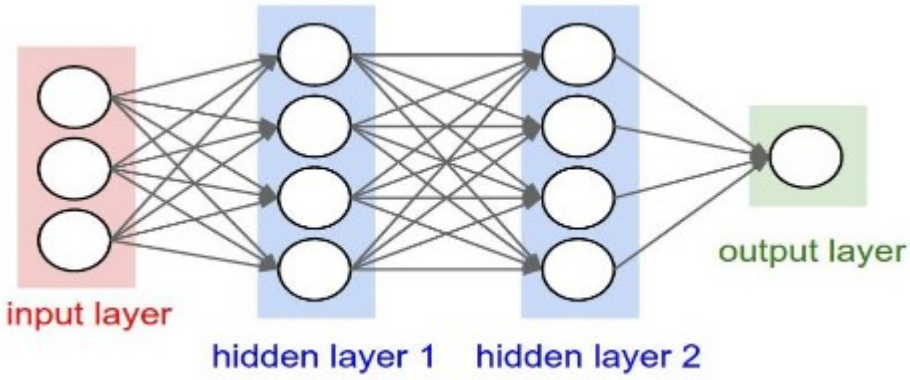
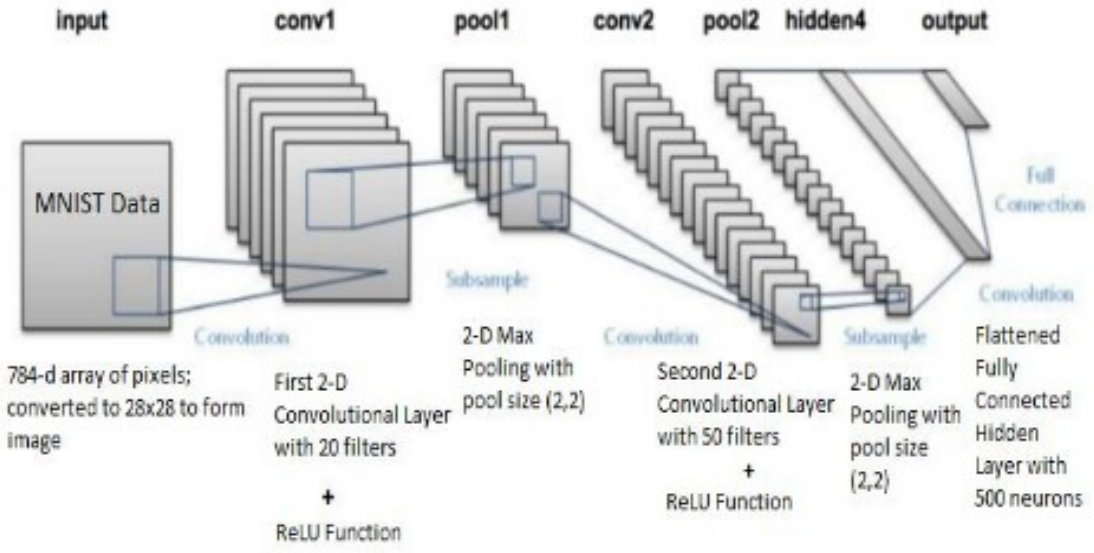


Fig. Convolutional Neural Network Basic Layout

A CNN consists of a lot of layers. These layers when used repeatedly, lead to a formation of a Deep Neural Network. Three main types of layers used to build a CNN are:

1.



Phase1 - Input MNIST Data1 :

The first phase is to input the MNIST data. So firstly we convert it to grayscale images using 28x28 matrix of pixels.

2. Phase2 – Building Network Architecture:

In the second phase, we define the models to be used to build a convolutional neural network. Here, we use the Sequential class from Keras to build the network. In this network, we have three layer sets of layers.

a) First Convolution Layer:

The input to a convolutional layer is a m x m x r image where m is the height and width of the image and r is the number of channels, e.g. an RGB image has r=3. The convolutional layer will have k filters (or kernels) of size nxnxq where n is smaller than the dimension of the image and q can either be the same as the number of channels r or smaller and may vary for each kernel.

The size of the filters gives rise to the locally connected structure which are each convolved with the image to produce k feature maps of size m−n+1.

b) ReLU Function:

The Rectified Linear Unit apply an elementwise activation function, such as the max (0, x) thresholding at zero.

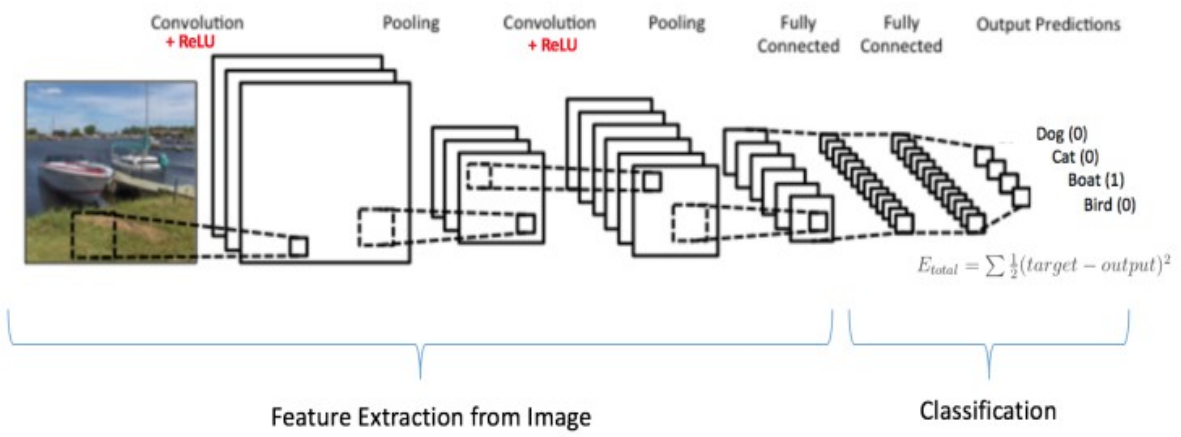
c) Pooling Layer:

perform a down sampling operation along the spatial dimensions (width, height)

3. Phase 3 – Fully Connected Layer:

compute the class scores (between 0-9 digits). As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume.

An example of a Convolutional Neural Network is given below



The model is trained with DNN on the training data (X_train, y_train) and then evaluated against test data (X_test, y_test).

Benchmark Model

[some simple or historical model or result to compare the defined solution to;]

The benchmark model is chosen from one of the kernels of the Kaggle Digit Recognizer competition. For this problem, I will try to beat its performance with other algorithms.

I do the model fitting on the whole training dataset with hyper-parameter ‘number of epochs’ = 10.We can tune this parameter.

Evaluation Metrics

[functional representations for how the solution can be measured;]

After fitting the model, the model can be evaluated on the unseen test data. Using model.evaluate function in Keras, we can calculate loss value & accuracy value.

Project Design

[how the solution will be developed and results obtained]

The dataset is loaded into python using keras library as discussed in the Dataset and Inputs section. The loaded data will be first explored and visualized using numpy and matplotlib library to understand the nature of the data.

Exploring the data will help us in deciding how to approach and whetherany preprocessing of the data is needed. Preprocessing of the data is done as required. Once the data is ready, the Deep neural network(DNN) will be built based on the architecture(ConvNet) as discussed in the Solution Statement. Once the model is built, it will be compiled to check if the architecture has any error.

The dataset is loaded into python using keras library as discussed in the Dataset and Inputs section. The loaded data will be first explored and visualized using numpy and matplotlib library to understand the nature of the data.

Exploring the data willhelp us in deciding how to approach and whether any preprocessing of the data is needed. Preprocessingof the data is done as required. Once the data is ready, the Deep neural network(DNN) will be built based on the architecture(ConvNet) as discussed in the Solution Statement. Once the model is built, it will be compiled to check if the architecture has any error.

Then the compiled model will be trained on the training data (X_train, y_train) and evaluated using accuracy score against the testing data (X_test, y_test).

Then the results can be analyzed and compared with respect to the benchmark model to know the overall performance of the model. Now we have a trained model, a test is conducted against the model by loading images of digits which are not from MNIST dataset to evaluate the performance of the model. The images are loaded and then preprocessed to match the MNIST dataset format so that we can test it. The model is then made to predict the digit in the preprocessed image. Thus, we can evaluate our model's performance over real time data