JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Handwritten Digit Recognition Using Convolutional Neural Network

[1]Mr.P. Ravi Prakash,[2]A. Lavanya,[3]K. Jayanth Krishna,[4]G. Prem Prathyush,[5]P. Divya Naga Sri

[1]Assistant Professor,[2,3,4,5]B.Tech. Student,
[1,2,3,4,5]Department of Information Technology,
[1]Prasad V. Potluri Siddhartha Institute Of Technology, Vijayawada,India.

Abstract : Handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It's a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system may be a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model created using PyTorch library over the MNIST(Modified National Institute of Standards and Technology) dataset to acknowledge handwritten digits.

KEYWORD: Handwritten digit recognition, convolutional neural network ministry data set, deep learning, back propagation.

Introduction

In the last few years, the area of fields where deep learning can be applied has increased vastly. In deep learning, convolutional neural networking is being used currently for visual image analyzing object detection face recognition robotics segmentation pattern recognition natural language processing spam detection regression analysis speech recognition image classification are some of the examples that can be done using CNN. The accuracy of perfection in these fields including handwritten digit recognition using deep convolutional neural networks has achieved human level perfection. In the year 1998, the framework of convolutional neural networks was designed by LeCun et al. The framework had seven layers of convolutional neural networks. It was adapted in handwritten digit classification right from the pixel values of images. To train the model gradient descent and back propagation is used. Characters are given as input in handwritten recognition digits. A simple artificial neural network has an input layer, an output layer and some hidden layers between the input and output layer. CNN has a remarkably similar structure to ANN. The layers in CNN have three dimensions. The layers are not fully connected. Instead, every neuron in the layer is connected to the locally respective field. Cost function generates to train the network. It compares the output of the network with the desired output from the data set. This signal propagates back to the system repeatedly to update the shared weights and biases in all respective fields to minimize the value of the cost function which increases the network's performance. The goal of this article is to observe the influence of hidden layers of a convolutional neural network for handwritten digits. We have applied a unique type of convolutional neural network algorithm to the modified National Institute of standards and technology data set using tensor flow, which is a neural network library written in Python. The main purpose of this paper is to analyze the changes of outcome results for using a different combination of hidden layers of convolutional neural networks. Stochastic gradient and backpropagation algorithms are used for training the network. For testing purposes, a forward algorithm is used.

I. PROPOSED SYSTEM

Handwritten Digit Recognition is that the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defenses, etc, and classify them into 10 predefined classes (0-9).This has been content of bottomless-exploration within the field of deep literacy.
The Modified National Institute of Standards and Technology (MNIST) dataset it's a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9.The task is to classify a given image of a handwritten digit into 10 classes representing integer values from 0 to 9.

II.PYTHON LIBRARIES

3.1 PyTorch

PyTorch is an open source scripting language based on the Lua programming language. It is a machine learning library. In handwritten digit recognition we use deep learning research for identifying the digits that are drawn by the human so pytorch is an MLlibrary used for creating deep neural networks.

3.2 Numpy

Numpy is known as Numerical python Extension are the libraries which are used for work with arrays. To perform a wide variety of mathematical operations on arrays.

3.3 Flask8

Flake8 is an application used for developing web applications using python. There are different built in servers and a fast debugger provider in flake8 library.

3.4 Matplotlib

Matplotllib is a cross platform for data visualization and graphical plot representation of the data.

3.5 Pillow

Pillow is a basic image processing functionality. It is an open source library that will support manipulating and used for support for different image file formats.

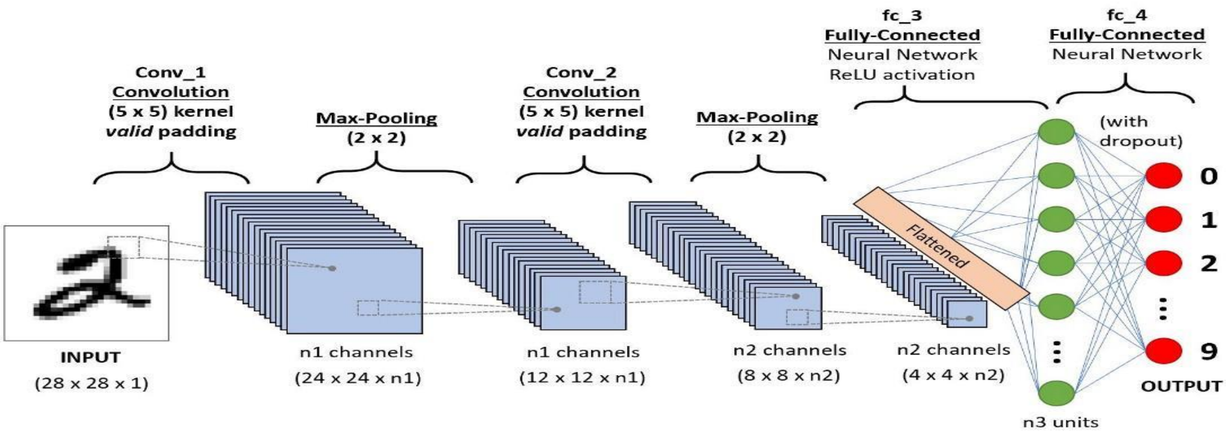
3.6 PIP

PIP is a package management system. This pip helps for installation of python software in the system through command.

These are some python libraries that are used in the handwritten digit recognition to perform different actions while execution of the process.

IV. Convolutional Neural Network

Convolutional neural network is an artificial intelligence that enables machines to do things like humans. This convolutional neural network has 7 layers with one input layer of 28 by 28 pixel image which has 784 neurons as input data. This input pixel is a grayscale with a value zero photo for white pixel and one photo for the black pixel. Whenever input is provided that input image has to travel through two phases that are feature learning and classification. In feature learning convolutional and RELU are used as activation functions to enhance the performance of the model. Second layer is the pooling layer; its function is to reduce the output information from the convolutional layer. Third layer is the flattened layer; its job is to convert the 2D feature map into a 1D feature vector and allow the data to be managed by the fully connected layer. Fourth layer is the fully connected layer, also known as the dense layer. The output layer of the network consists of ten neurons and determines the digits numbered from 0-9 to enhance the performance of the model. The output layer uses an activation function such as SoftMax classifieds the output digit from 0-9 which has the highest activation function[2].



Fig; A seven-layered convolutional neural network for digit recognition

V. Minst Dataset

Modified National Institute of Standards and Technology is a large set of computer vision datasets which is extensively used for training and testing different systems. It was created from the two datasets of National Institute of Standards and Technology which holds binary images of handwritten digits. The training set consists of handwritten digits from 250 people among them 30% training data set versus employees from the Census Bureau and the rest was from high school students. However, it is often attributed as the first datasets among other datasets to prove the effectiveness of neural networks. The database contains 60,000 images used for training as well as a few of them that can be used for cross validation purposes and 10,000 images for testing purposes. All the digits are Gray scaled and positioned in a fixed size where the intensity lies at the center of the image. The images from and array which can be flattened into dimensional vectors where each component of the vector is a binary value which describes the intensity of the pixel[1].

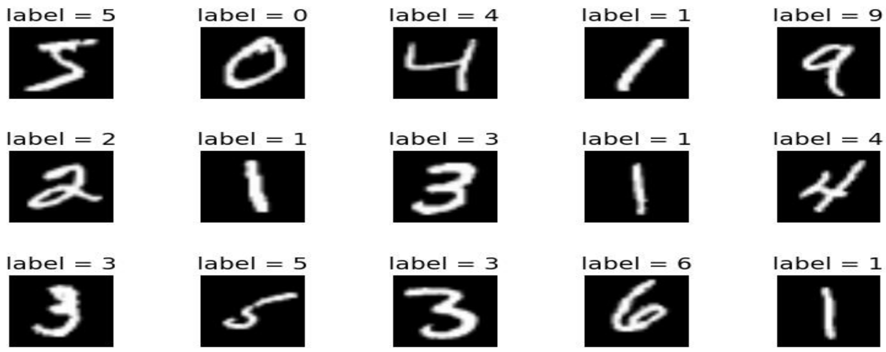


Fig: Sample images of MNIST handwritten digit dataset.

VI. IMPLEMENTATION

Types of Layers

Let's understand the working of layers by taking an example of an image of dimension 28*28*1.

1. Input Layer**:** This layer stores the raw input of an image with dimension 28*28*1.
2. Convolution Layer**:** This layer performs the computations to obtain the output volume. Output volume can be obtained by performing dot products between all filters and image patches[3].

Fig: The Mathematical Operations Occurring at the First and Second Hidden Layer

$$C(w,b) = \frac{1}{2n}\sum_{x}\left[y(x) - a^2\right]^2$$

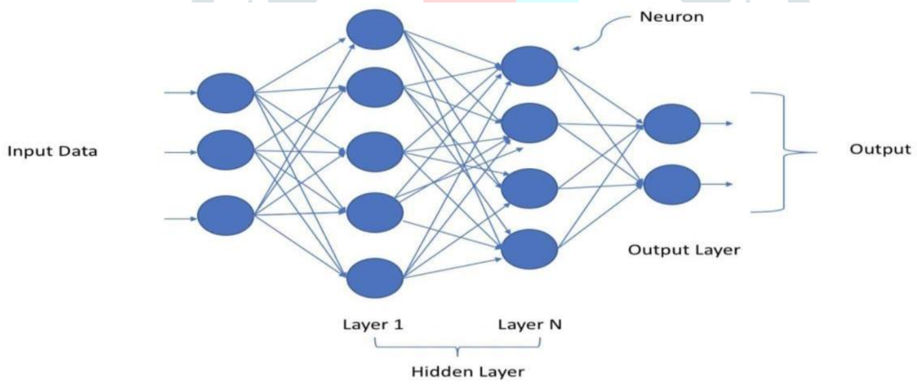


Fig: Hidden layers.

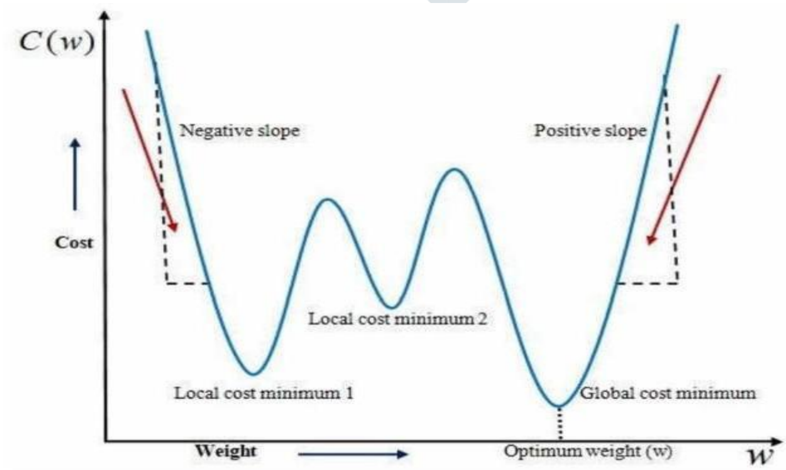


Fig: Graphical Representation of Cost vs. Weight

1. Activation Function Layer: This layer applies activation function to every and every element of the output of the second layer. Some most often used activation functions are Sigmoid, Tanh, RELU, Leaky RELU, etc. the quantity does not change in this layer.

2. Pool Layer: Main function of this layer is to decrease the dimensions of the volume. Decreasing the quantity makes the computation fast and reduces the memory. Pool layer also prevents overfitting. Two common sorts of pooling layers are max pooling and average pooling.
3. Fully-Connected Layer: This layer may be a regular neural network layer. It takes input from the previous layer and outputs the category using activation functions and classifies the input image[3].

VII. RESULT

In this, CNN is applied on MNIST dataset to handwritten digits recognition to observe the accuracies.Here we are using different cases to find the accuracy. In case 1 shown in figure 1, Here the first hidden case is convolutional layer 1. It is used for feature extraction and it consists of 32 filters and ReLU (Rectified Linear Units) is used as activation function to increase the performance. Next hidden layer consists of 64 filters and ReLU. Next pooling layer 1 to define the max pooling with a pooling of 2x2 pixels to minimize the output of the convolutional layer. A dropout layer is used next to the pooling to reduce the output overfitting. A flatten layer is used next to the dropout to convert the 2D filter matrix to 1D filter matrix and then fully connected layer 1 consists of 128 neutrons and ReLU. And finally fully connected layer 2 contains the 10 neutrons for 10 classes to determine the digits from 0 to 9.

A softmax activation function is integrated with the output layer. The CNN has 15 epochs with a batch size of 100. The overall validation accuracy in the performance is found at 99.23%. At epoch 1 the minimum training accuracy of 92.02% is found and 97.83% of validation accuracy is found. At epoch 13, the maximum training accuracy is found at 98.99% and at epoch 14, the maximum validation accuracy is found at 99.07%. The total test loss for this case is found to be approximately 0.037845.

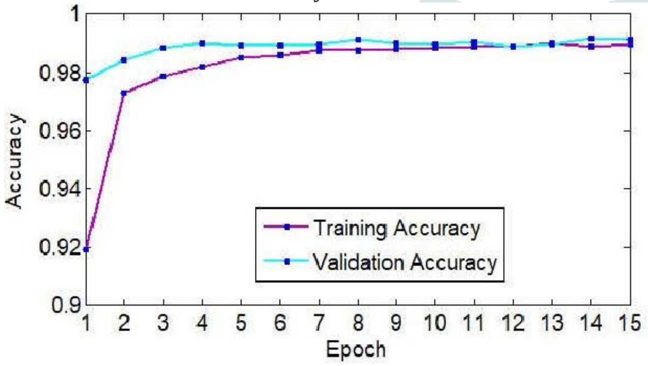


Fig1: Observed accuracy for case 1.

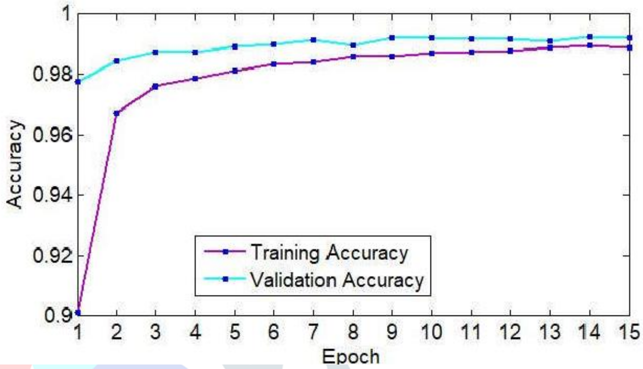


Fig2; Observed accuracy for case 2.

In case 2, where convolution one, pooling one and convolution two, pooling two is employed one after another. A dropout is employed followed by the flatten layer and fully connected layer 1. Before the fully connected layer 2 another dropout is employed . The size and parameters used here and for the next cases are the same which are used earlier for case 1.The validation accuracy within the performance is 99.02%. The minimum training accuracy is 90.11% and therefore the minimum validation accuracy is 97.74%. The utmost training and validation accuracy are found at epoch 14. The utmost training and validation accuracies are 98.94% and 99.24% from figure 2. The entire test loss is found at approximately 0.026303. In case 3, shown in figure 3, where two convolutions are taken 1 after another followed by a pooling layer. After the pooling layer, a flatten layer is used followed by two fully connected layers without any dropout.

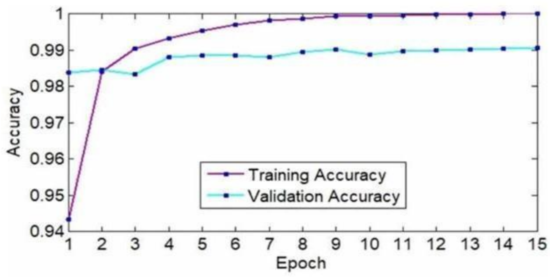


Fig3: Observed accuracy for case 3.

The validation accuracy in the performance is found 99.25%. The minimum training accuracy is found 94.45% at epoch 1 and epoch 3, the minimum validation accuracy is found 98.33%. The maximum training and validation accuracies are 1% and 99.06% found at epoch 15. The total test loss is found to be approximately 0.049449.

Similarly, in case 4 shown in figure 4, convolution 1, pooling 1 and convolution 2, pooling 2 are used alternately followed by a flatten layer and two fully connected layers without any dropout. The overall validation accuracy in the performance is found 99.40%. The minimum training accuracy is 92.54% and the minimum validation accuracy is 97.77%. The maximum training accuracy is found 99.92% at epoch 15 and epoch 13, the maximum validation accuracy also found 99.82%. The total test loss is found at approximately 0.033387.

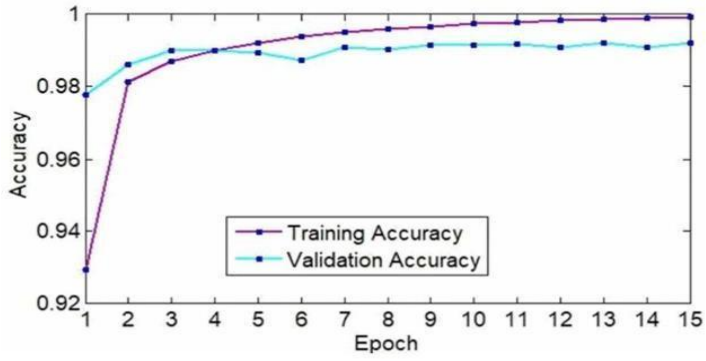


Fig 4: Observed accuracy for case 4.

Again, for case 5 shown in figure 5, two convolutions are used one after another followed by a pooling layer, flatten layer and fully connected layer 1. But this time we are using dropout before fully connected layer 2. The validation accuracy in the performance is found 99.09%. The minimum training accuracy is 91.70% and the minimum validation accuracy is 98.46%. At epoch 13, the maximum training accuracy is found at 99.01% and the maximum validation accuracy is found at 99.14% at epoch 12. The total test loss is found at approximately 0.037336.

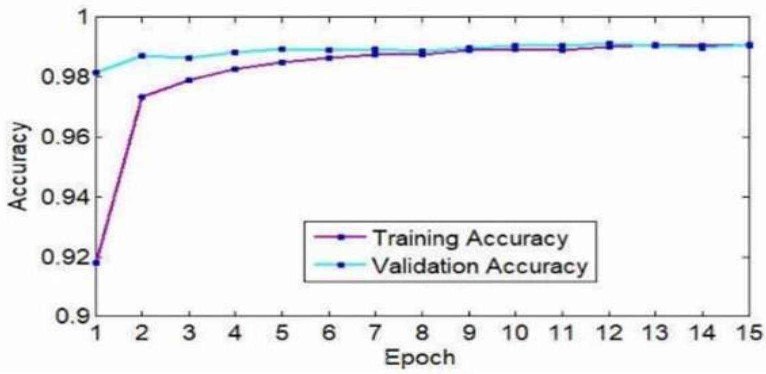


Fig 5: Observed accuracy for case 5.

Finally, for case 6 shown in figure 6, In this case the process will be the same as case 2 but only the difference is a dropout will be used before the fully connected layer 2. The overall validation accuracy in the performance is found 99.17%. The minimum training accuracy is 91.01% and the minimum validation accuracy is 97.13%. The maximum training accuracy is found 99.34% at epoch 15 and the maximum validation accuracy is found at 99.36% at epoch 13. Approximately 0.028696 will be found in total test loss.

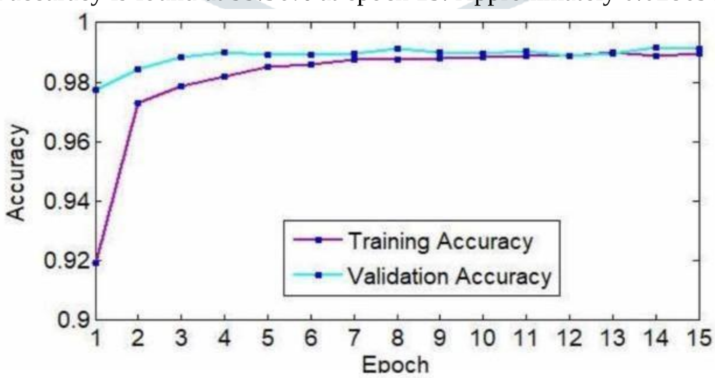


Fig 6: Observed accuracy for case 6.

VIII. Scope Of Use In Future

This system has various applications and can be used for solving real-time problems such as the problem of faultily written checks in banking documents and forensics etcetera. By employing the same strategy with alphabets, we may use it full-fledgedly to rely on a similar index checker of handwritten documents along with their digital conversion. The digital document can be further translated into an audio file. This technique is helpful for visually impaired people. The knowledge attained through this also helps in automating other projects such as big data and IoT AI and neural network will be specifically helped by this compatibility of this project with knowledge-based technologies is high due to its nature raw data based project.

IX. CONCLUSION

In this paper the variations of accuracies for handwritten digits were observed for various digits from datasets. The accuracy of prediction will increase and result in the decrease of error rate by having more training examples. We found out that this CNN approach is more effective than random forest classifier and K nearest neighbor add supervised to vector machine with the utmost 1.28 error rate. The CNN approach gave better results because of the use of multiple layers and filters which helped in reduction of error in classification. This paperwork can be extended to a higher level in future so the software can have some better changes to make it more dependable, secure, and accurate. During our building and working process we came to know that our techniques still have some weak areas, for example the strokes of digits six and nine can be mistaken and three is also like eight. Although we received exceptionally satisfactory results, we still have an approximately 1.28 error rate.

X. REFERENCE

[1].https://www.kaggle.com/code/ngbolin/mnist-dataset-digit-recognizer/notebook

[2].https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[3]. https://www.geeksforgeeks.org/introduction-convolution-neural-network/