IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

MNIST Handwritten Input Database Classification Using Deep Learning

B.Sujani Reddy[1] ,K.RaviTeja[2] , Manish Kumar Yadav[3], Uday Raddi[4]

[1]Associate Professor, Department of CSE (AI & ML), CMR College of Engineering & Technology, Hyderabad, Telangana State, India.

[2,3,4] B.Tech Students, Department of CSE (AI & ML), CMR College of Engineering & Technology, Hyderabad, Telangana State, India.

ABSTRACT: The implementation of a back-propagation neural network to classify the MNIST handwritten digit database is shown in this research. Here, we assess the neural network's performance using the plot of loss during training and classification accuracy. The outcomes of the experiment indicate that back-propagation neural networks can be used to real-world categorization problems to a certain extent. Additionally, we attempt to use Autoencoder to produce picture compression and analysed its results. Reducing the size of the network helps to speed up the neural network and enhance performance. Though several causes are listed, the accuracy rate cannot be guaranteed. Next, we attempt to convert the original neural network into a Convolutional Neural Network (CNN) by changing its structure. According to the findings, CNN can be utilized to enhance performance while tackling image identification tasks. Additionally, we build a Conv Autoencoder structure by combining CNN and Autoencoder, and we run various tests.

I . INTRODUCTION:

In the field of optical character recognition, recognizing handwritten digits is a significant challenge that can serve as a test case for machine learning algorithms and pattern recognition theories. Many common datasets have been developed to support machine training and pattern recognition research. In order to minimize burden and enable researchers to evaluate identification results of different algorithms on an equal footing, the handwritten digits were processed beforehand including segmentation or normalization (Deng, 2012). In this paper, an artificial neural network (ANN) using the MNIST handwritten digit database is used. The MNIST handwritten digit database can be accessed via Yann LeCun's website (Yann.lecun.com, n.d.). It is now accepted practice for quickly testing machine learning and pattern recognition theories. Modified NIST, often known as MNIST, was created using the original NIST database as a foundation. It is made up of 10,000 handwritten digits images for the classifier testing and 60,000 handwritten digits images for the classifier training, all taken from the same allocation. The size normalization of every single black and white digits is cantered in a fixed-size image, with the intensity centre located at the image's 28 x 28-pixel centre. According to Deng (2012), the dimensions of every picture sample vector is 28 * 28 = 784, with binary elements for every element. We have used this MNIST handwritten digits database for a number of purposes. First off, it is a standard, a somewhat straightforward database for quickly testing ideas and methods, as was already mentioned. Additionally, the handwritten digits in the

MNIST database have already undergone segmentation and normalization as part of the preprocessing and formatting that we want to test neural networks when applied to real-world practical problems. This should save us time and effort when it comes to formatting and preprocessing the data. Furthermore, a large body of research has been conducted using MNIST to evaluate theories and algorithms, so we can compare our findings with those of a very extensive body of literature. The goal of this research is to describe the use of neural networks to the MNIST handwritten digit classification issue. The task at hand involves creating a model of neural networks and putting it into practice to address the categorization issue. Additionally, a few more tests have been conducted to evaluate other approaches that could potentially affect our model's functionality. According to Basheer and Hammer (2000), artificial neural networks (ANNs) are based on biological central systems found in brains and are used for modelling, classification, pattern recognition, and multivariate data processing, among other things. It is quite common to use back-propagation neural networks in actual ANN applications. D. Rumelhart and J. McCelland introduced the back-propagation algorithm in 1986, and the neural networks they demonstrated utilizing BP techniques are known as backpropagation neural networks (BPNN) (Rumelhart, Hinton, and McClelland, 1986). In this study, we solve the classification problem using a back-propagation neural network. An input layer, an output layer, and one or more hidden layers are the typical components of back-propagation neural networks, which are controlled multi-layer feed forward neural networks.

This is the three-layer neural network's basic model. The network consists of three layers: the input layer, the hidden layer, and the output layer. Although there are no connections inside the components of the same layer, layers are completely interlinked with one another. Nodes are computational units that require inputs to be processed by neurons in order to produce outputs. Summing the numbers is one of the functions used to handle inputs. The input, weights, activation function, threshold, & output are the components that make up a simple neuron. After multiplying weights by inputs, adding them in the summing function, processing the total in the activation function, and finally producing an output, the model is finished.

II. RELATED WORK

Artificial neural networks: fundamentals, computing, design, and application

Artificial neural networks (ANNs) represent a relatively new class of computational tools that have been widely used to the resolution of numerous challenging real-world issues. Because of its exceptional information processing qualities, which are mostly related to nonlinearity, high parallelism, fault and noise tolerance, and learning and generalization abilities, artificial neural networks (ANNs) are becoming more and more popular. In addition to introducing the reader to ANN-based computing, or neurocomputing, this work intends to be a helpful toolset and companion practical guide for ANNs modelers during the ANN project development process. A brief discussion is given of the development of neurocomputing's history and its connection to the study of neurobiology. The benefits and drawbacks of ANNs are discussed along with comparisons to expert systems and statistical regression. With a focus on the theory and construction of backpropagation (BP) ANNs, an overview of the many kinds of ANNs and the associated learning principles is provided. A comprehensive approach to creating effective ANNs projects, covering the entire spectrum from ideation to design and execution, is explained. During training, the most frequent issues that BPANNs developers run into are enumerated along with potential causes and solutions. Lastly, the microbial growth curves of S. flexneri were modelled using BPANNs as a practical application. The generated model was able to simulate training and test time-dependent growth curves that were influenced by pH and temperature with a reasonable degree of accuracy.

Convolutional neural network committees for handwritten character classification

Following several years of inactivity, the MNIST handwriting recognition benchmark record saw a decline in error rate from 0.40% to 0.35% in 2010. Here, we report a performance margin close to human level of 0.27% for a committee of seven deep CNNs trained on graphics cards. The same architecture is also used to NIST SD 19, a more difficult dataset that contains both capital and lowercase letters. For both NIST digits and NIST letters, a committee consisting of seven CNNs achieves the best performance to date. Through analysis of 78125 distinct 7-net committees, the robustness of our approach is confirmed.

The MNIST database of handwritten digit images for machine learning research

The National Institute of Standards and Technology (NIST) resources, which have been enhanced and are a set of handwritten digit images used extensively in optical character recognition and machine learning research, are included in this edition of "Best of the Web."

Progressive image compression

The quality of the decompressed image is a major factor in many neural network applications for image compression. A feedforward network with three tiers of processing units is typically assumed by the writers. There are no lateral, backward, or multilayer connections; all connections are made between units in one level and those in the next. Simple weighted connections exist between each unit and the units in the layer above. By having fewer units in the hidden layer than in the input layer, the picture is compressed. To recover the compressed image, the output layer—which has the same dimensions as the input layer—is employed. Based on how unique each unit in the compression layer is, they may ensure a constant degree of functionality and gradually shrink the compression layer to achieve the required level of image quality.

III. METHODOLOGY

To Implanting this project In propose two algorithms called Autoencoder and Modified CNN and then comparing accuracy and loss between two algorithms. While running this project first time system must be connected to internet so application can download MNIST dataset from internet and while running at firsime due to dataset download it may take little longer time.

IV. SYSTEM REQUIREMENT

User Interface

The user interface of this system is a user friendly python Graphical User Interface.

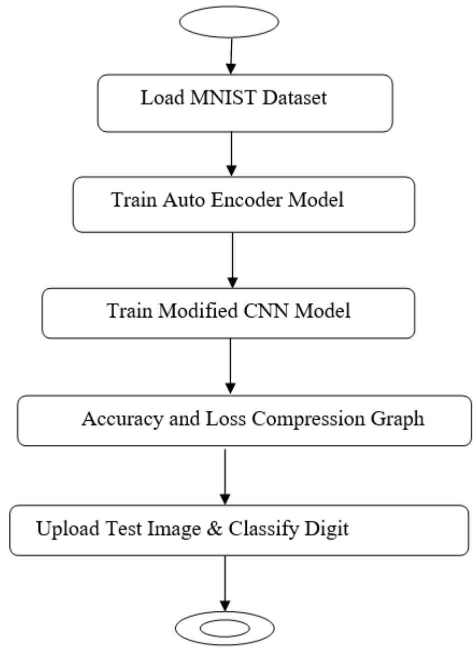


Fig 1: Bock Diagram

Hardware Interfaces

The interaction between the user and the console is achieved through python capabilities.

Software Interfaces

The required software is python.

Operating Environment

Windows XP.

HARDWARE REQUIREMENTS:

- Processor - Pentium –IV
- Speed - 1.1 Ghz
- **RAM - 256 MB(min)**
- **Hard Disk - 20 GB**
- Key Board - Standard
- Mouse - Two or Three

- Monitor - SVGA

SOFTWARE REQUIREMENTS:

- Operating System - Windows7/8
- Programming Language- Python

VI. Modules:

1. Load MNIST Dataset

Load MNIST dataset is the first module in our project, then the data set loaded and we can see total dataset size is 60000 which means dataset contains 60000 images.

2. Train Autoencoder Model

on 'Train Autoencoder Model' module could generated and loaded Autoencoder model and then accuracy is calculated.

3. Train Modified CNN Model

Train Modified CNN Model is the third module in our project. In this module we have to train CNN model on MNIST dataset.

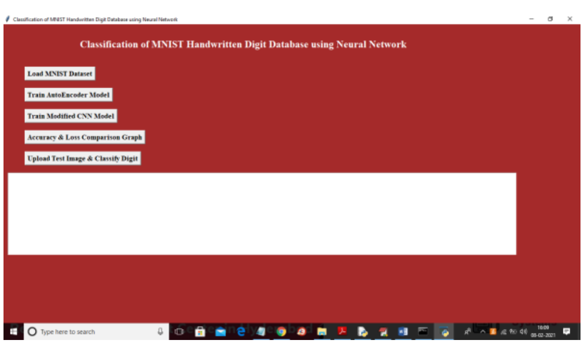
4.Accuracy and Loss Compression Graph

In above screen x-axis represents number of epoch and y-axis represents accuracy/loss value.

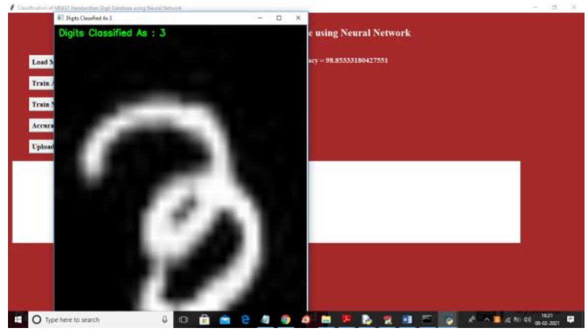
5.Upload Test Image & Classify Digit

Here, we have to upload a test image then the image can loaded and classify the digit.

IX. RESULT AND DISCUSSION



Click on the "Load MNIST Dataset" button on the screen above to import the dataset and access the subsequent interface shown below.



In above screen digit classified as 3 and now test with another image

X. CONCLUSION

In order to classify the MNIST handwritten digits database, we conduct an experiment in this study using a back-propagation neural network. In the case of the experiment, 10 distinct classes of digits, ranging from 0 to 9, are considered the output, and 28 * 28 = 784 pixels are considered the input. In addition, we assess the neural network's performance using the classification precision and loss plot. Following model parameter testing, we established the following system parameters: Batch size = 100, learning rate = 0.005, and number of epochs = 6. The outcomes of the experiment indicate that back-propagation neural networks can be used to real-world categorization problems to a certain extent. Additionally, we attempt to use Autoencoder to produce picture compression and analysed its results. It does aid in decreasing the size of the network and boosting the neural network's speed. It is not possible to guarantee the accuracy rate, though. Next, we attempt to convert the initial neural network into an CNN (Convolutional Neural Network) by changing its structure. According to the findings, CNN can be utilized to enhance performance while tackling image identification tasks. Additionally, we merge CNN and Autoencoder to create a Conv Autoencoder structure. Tests indicate that this Conv Autoencoder outperforms the original Autoencoder when it comes to picture compression.

XI. REFERENCES

1. Basheer, I.A., and Hammer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods, 43(1), pp.3-31.

2. Ceresin, D.C., Meier, U., Gambardella, L.M. and Schmid Huber, J., 2011, September. Convolutional neural network committees for handwritten character classification. In Document Analysis and Recognition (ICDAR), 2011 International Conference on (pp. 1135-1139). IEEE.

3. Deng, L., 2012. The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Processing Magazine, 29(6), pp.141-142.

4. Gedeon, T.D. and Harris, D., 1992, June. Progressive image compression. In Neural Networks, 1992. IJCNN., International Joint Conference on (Vol. 4, pp. 403-407). IEEE.

5. Gedeon, T.D., Catalan, J.A. and Jin, J., Image Compression using Shared Weights and Bidirectional Networks. In Proceedings 2nd International ICSC Symposium on Soft Computing (SOCO'97) (pp. 374-381).

6. LeCun, Y., Bottom, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.

7. O'Reilly | Safari., 2018. TensorFlow for Deep Learning. [online] Available at: https://www.safaribooksonline.com/library/view/tensorflow-for-deep/9781491980446/ch01.html [Accessed 30 May 2018].

8. Rumelhart, D.E., Hinton, G.E. and McClelland, J.L., 1986. A general framework for parallel distributed processing. Parallel distributed processing: Explorations in the microstructure of cognition, 1, pp.45-76.

9. Yann.lecun.com., n.d. MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. [online] Available at: http://yann.lecun.com/exdb/mnist/ [Accessed 30 May 2018].

10. Yann.lecun.com., n.d. MNIST Demos on Yann LeCun's website. [online] Available at: http://yann.lecun.com/exdb/lenet/ [Accessed 30 May 2018].