CHAPTER 41
Implementation of Ultrasonic Specialization for Blind People Using Aurdino

Mr. Biju Balakrishnan
Hindusthan Institute of Technology, Coimbatore, India
Ms. Agarshana S
Hindusthan Institute of Technology, Coimbatore, India
Ms. Jenesha J
Hindusthan Institute of Technology, Coimbatore, India
Ms. Kaviya S
Hindusthan Institute of Technology, Coimbatore, India
Ms. Lam Amulya
Hindusthan Institute of Technology, Coimbatore, India

ABSTRACT

The reliance of humans over machines has never been so high such that from object classification in photographs to adding sound to silent movies everything can be performed with the help of deep learning and machine learning algorithms. Likewise, Handwritten text recognition is one of the significant areas of research and development with a streaming number of possibilities that could be attained. Handwriting recognition (HWR), also known as Handwritten Text Recognition (HTR), 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. Apparently, in this paper, we have performed handwritten digit recognition with the help of MNIST datasets using Support Vector Machines (SVM), Multi-Layer Perceptron (MLP) and Convolution Neural Network (CNN) models. Our main objective is to compare the accuracy of the models stated above along with their execution time to get the best possible model for digit recognition.

Keywords: *Handwritten text recognition , Arduino Bases, ultrasonic sensor, PCB materials, etc.*

INTRODUCTION

The comparison of the algorithms (Support vector machines, Multi-layered perceptron and Convolutional neural network) is based on the characteristic chart of each algorithm on common grounds like dataset, the number of epochs, complexity of the algorithm, accuracy of each algorithm, specification of the

ISBN: 978-93-5782-920-5, www.aetsjournal.com ―Handbook of research in Big data Analytics, Artificial intelligence and Machine learning

device (Ubuntu 20.04 LTS, i5 7th gen processor) used to execute the program and runtime of the algorithm, under ideal

DATASET

Handwritten character recognition is an expansive research area that already contains detailed ways of implementation which include major learning datasets, popular algorithms, features scaling and feature extraction methods. MNIST dataset (Modified National Institute of Standards and Technology database) is the subset of the NIST dataset which is a combination of two of NIST's databases: Special Database 1 and Special Database 3. Special Database 1 and Special Database 3 consist of digits written by high school students and employees of the United States Census Bureau, respectively. MNIST contains a total of 70,000 handwritten digit images (60,000 - training set and 10,000 - test set) in 28x28 pixel bounding box and anti-aliased. All these images have corresponding Y values which apprises what the digit is

SUPPORT VECTOR MACHINE

It is a supervised machine learning algorithm. In this, we generally plot data items in n-dimensional space where n is the number of features, a particular coordinate represents the value of a feature, we perform the classification by finding the hyperplane that distinguishes the two classes. It will choose the hyperplane that separates the classes correctly. SVM chooses the extreme vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. There are mainly two types of SVMs, linear and non-linear SVM. In this paper, we have used Linear SVM for handwritten digit recognition [10].

MULTI LAYERED PERCEPTION

A multilayer perceptron (MLP) is a class of feedforward artificial neural networks (ANN). It consists of three layers: input layer, hidden layer and output layer. Each layer consists of several nodes that are also formally referred to as neurons and each node is interconnected to every other node of the next layer. In basic MLP there are 3 layers but the number of hidden layers can increase to any number as per the problem with no restriction on the number of nodes. The number of nodes in the input and output layer depends on the number of attributes and apparent classes in the dataset respectively. The particular number of hidden layers or numbers of nodes in the hidden layer is difficult to determine due to the model erratic nature and therefore selected experimentally. Every hidden layer of the model can have different activation functions for processing. For learning purposes, it uses a supervised learning technique called backpropagation. In the MLP, the connection of the nodes consists of a weight that gets adjusted to synchronize with each connection in the training process of the model[11].

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 [12]. 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 [13]. As we move into the CNN, we observe the height and width

ISBN: 978-93-5782-920-5, www.aetsjournal.com ―Handbook of research in Big data Analytics, Artificial intelligence and Machine learning

decrease while the number of channels increases. Finally, the generated column matrix is used to predict the output [14].

VISUALIZATION

In this research, we have used the MNIST dataset (i.e. handwritten digit dataset) to compare different level algorithm of deep and machine learning (i.e. SVM, ANN-MLP, CNN) on the basis of execution time, complexity, accuracy rate, number of epochs and number of hidden layers (in the case of deep learning algorithms). To visualize the information obtained by the detailed analysis of algorithms we have used bar graphs and tabular format charts using module matplotlib, which gives us the most precise visuals of the step by step advances of the algorithms in recognizing the digit. The graphs are given at each vital part of the programs to give visuals of each part to bolster the outcome.

IMPLEMENTATION

To compare the algorithms based on working accuracy, execution time, complexity, and the number of epochs (in deep learning algorithms) we have used three different classifiers: • Support Vector Machine Classifier • ANN - Multilayer Perceptron Classifier • Convolutional Neural Network Classifier We have discussed in detail about the implementation of each algorithm explicitly below to create a flow of this analysis to create a fluent and accurate comparison.

PREPROCESSING

Pre-processing is an initial step in the machine and deep learning which focuses on improving the input data by reducing unwanted impurities and redundancy. To simplify and break down the input data we reshaped all the images present in the dataset in 2-dimensional images i.e (28,28,1). Each pixel value of the images lies between 0 to 255 so, we Normalized these pixel values by converting the dataset into 'float32' and then dividing by 255.0 so that the input features will range between 0.0 to 1.0. Next, we performed one-hot encoding to convert the y values into zeros and ones, making each number categorical, for example, an output value 4 will be converted into an array of zero and one i.e [0,0,0,0,1,0,0,0,0,0].

SUPPORT VECTOR MACHINE

The SVM in scikit-learn [16] supports both dense (numpy.ndarray and convertible to that by numpy.asarray) and sparse (any scipy.sparse) sample vectors as input. In scikit-learn, SVC, NuSVC and LinearSVC are classes capable of performing multi-class classification on a dataset. In this paper we have used LinearSVC for classification of MNIST datasets that make use of a Linear kernel implemented with the help of LIBLINEAR [17]. Various scikit-learn libraries like NumPy, matplotlib, pandas, Sklearn and seaborn have been used for the implementation purpose. Firstly, we will download the MNIST datasets, followed by loading it and reading those CSV files using pandas. After this, plotting of some samples as well as converting into matrix followed by normalization and scaling of features have been done. Finally, we have created a linear SVM model and confusion matrix that is used to measure the accuracy of the model [9].

MULTI LAYERED PERCEPTION

The that by numpy.asarray) and sparse (any scipy.sparse) sample vectors as input. In scikit-learn, SVC, NuSVC and LinearSVC are classes capable of performing multi-class classification on a dataset. In this

ISBN: 978-93-5782-920-5, www.aetsjournal.com ―Handbook of research in Big data Analytics, Artificial intelligence and Machine learning

paper we have used LinearSVC for classification of MNIST datasets that make use of a Linear kernel implemented with the help of LIBLINEAR [17]. Various scikit-learn libraries like NumPy, matplotlib, pandas, Sklearn and seaborn have been used for the implementation purpose. Firstly, we will download the MNIST datasets, followed by loading it and reading those CSV files using pandas. After this, plotting of some samples as well as converting into matrix followed by normalization and scaling of features have been done. Finally, we have created a linear SVM model and confusion matrix that is used to measure the accuracy of the model [9].

We used a neural network with 4 hidden layers and an output layer with 10 units (i.e. total number of labels). The number of units in the hidden layers is kept to be 512. The input to the network is the 784-dimensional array converted from the 28×28 image. We used the Sequential model for building the network. In the Sequential model, we can just stack up layers by adding the desired layer one by one. We used the Dense layer, also called a fully connected layer since we are building a feedforward network in which all the neurons from one layer are connected to the neurons in the previous layer. Apart from the Dense layer, we added the ReLU activation function which is required to introduce non-linearity to the model. This will help the network learn non-linear decision boundaries. The last layer is a softmax layer as it is a multiclass classification problem [19].

CONVOLUTIONAL NEURAL NETWORKS

The implementation of handwritten digit recognition by Convolutional Neural Network [15] is done using Keras. It is an open-source neural network library that is used to design and implement deep learning models. From Keras, we have used a Sequential class which allowed us to create model layer-by-layer. The dimension of the input image is set to 28(Height) 28(Width), 1(Number of channels). Next, we created the model whose first layer is a Conv layer [20]. This layer uses a matrix to convolve around the input data across its height and width and extract features from it. This matrix is called a Filter or Kernel. The values in the filter matrix are weights. We have used 32 filters each of the dimensions (3,3) with a stride of 1. Stride determines the number of pixels shifts. Convolution of filter over the input data gives us activation maps whose dimension is given by the formula: ((N + 2P - F)/S) + 1 where N= dimension of input image, P= padding, F= filter dimension and S=stride. In this layer, Depth (number of channels) of the output image is equal to the number of filters used. To increase the non-linearity, we have used an activation function that is Relu [21]. Next, another convolutional layer is used in which we have applied 64 filters of the same dimensions (3,3) with a stride of 1 and the Relu function. Next, to these layers, the pooling layer [22] is used which reduces the dimensionality of the image and computation in the network. We have employed MAX-pooling which keeps only the maximum value from a pool. The depth of the network remains unchanged in this layer. We have kept the pool-size (2,2) with a stride of 2, so every 4 pixels will become a single pixel. To avoid overfitting in the model, Dropout layer [23] is used which drops some neurons which are chosen randomly so that the model can be simplified. We have set the probability of a node getting dropped out to 0.25 or 25%. Following it, Flatten Layer [23] is used which involves flattening i.e. generating a column matrix (vector) from the 2-dimensional matrix. This column vector will be fed into the fully connected layer [24]. This layer consists of 128 neurons with a dropout probability of 0.5 or 50%. After applying the Relu activation function, the output is fed into the last layer of the model that is the output layer. This layer has 10 neurons that represent classes (numbers from 0 to 9) and the SoftMax function [25] is employed to perform the classification. This function returns probability distribution over all the 10 classes. The class with the maximum probability is the output.

RESULT

After implementing all the three algorithms that are SVM, MLP and CNN we have compared their accuracies and execution time with the help of experimental graphs for perspicuous understanding. We

ISBN: 978-93-5782-920-5, www.aetsjournal.com ―Handbook of research in Big data Analytics, Artificial intelligence and Machine learning

have taken into account the Training and Testing Accuracy of all the models stated above. After executing all the models, we found that SVM has the highest accuracy on training data while on testing dataset CNN accomplishes the utmost accuracy. Additionally, we have compared the execution time to gain more insight into the working of the algorithms. Generally, the running time of an algorithm depends on the number of operations it has performed. So, we have trained our deep learning model up to 30 epochs and SVM models according to norms to get the apt outcome. SVM took the minimum time for execution while CNN accounts for the maximum running time

Furthermore, we visualized the performance measure of deep learning models and how they ameliorated their accuracy and reduced the error rate concerning the number of epochs. The significance of sketching the graph is to know where we should apply early stop so that we can avoid the problem of overfitting as after some epochs, change in accuracy becomes constant.

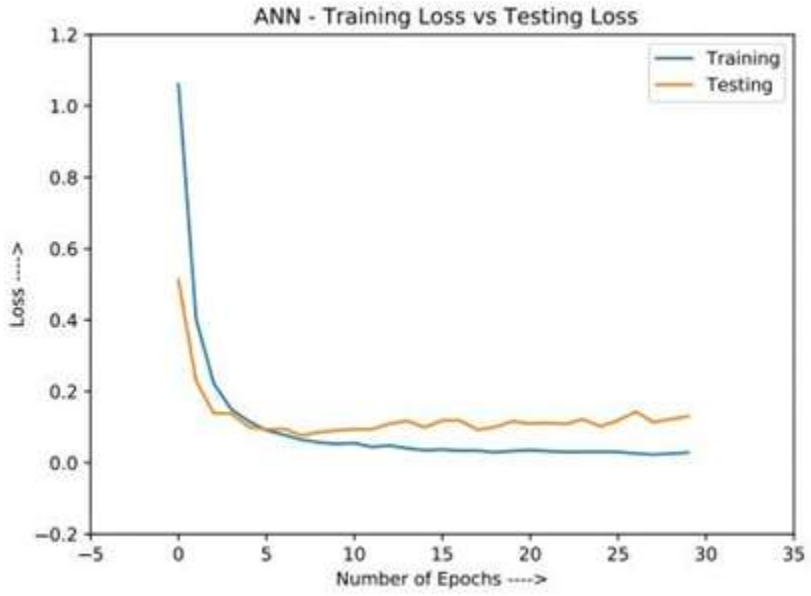


FIGURE 2: Graph illustrating the transition of training loss with increasing number of epochs in Multilayer Perceptron (Loss rate v/s Number of epochs).

ISBN: 978-93-5782-920-5, www.aetsjournal.com ―Handbook of research in Big data Analytics, Artificial intelligence and Machine learning

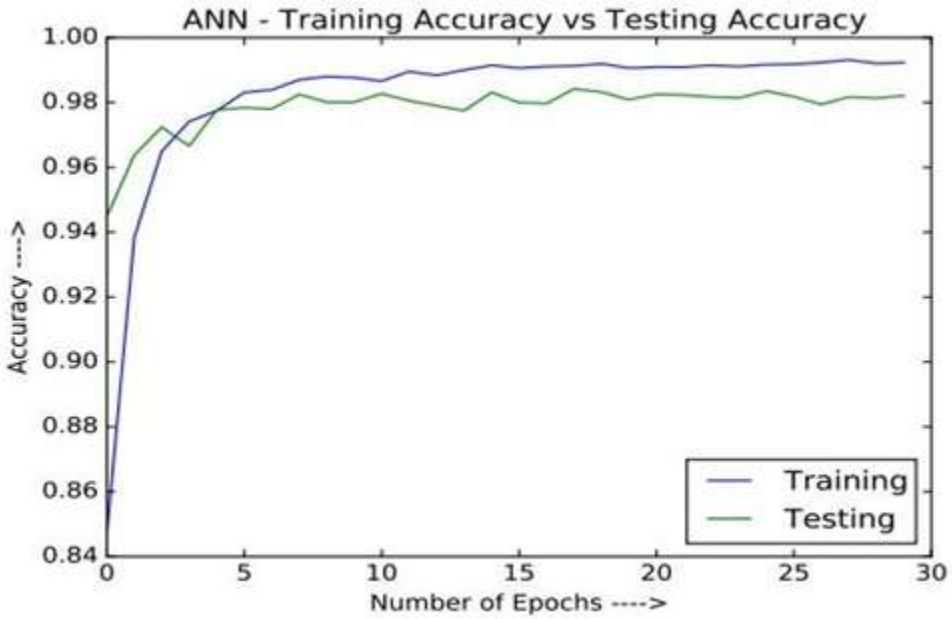
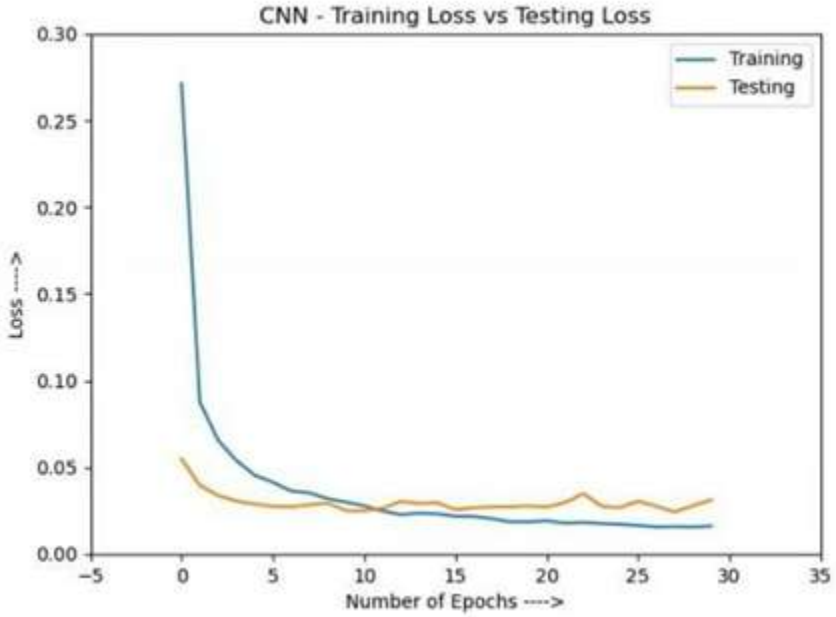


FIGURE 3 Graph illustrating the transition of training accuracy with increasing number of epochs in Multilayer Perceptron (Accuracy v/s Number of epochs).



Copyright © 2023, AETS Global.. Pages: 373- 380, September 2023
ISBN: 978-93-5782-920-5, www.aetsjournal.com ―Handbook of research in Big data Analytics, Artificial intelligence and Machine learning

FIGURE 4 Graph illustrating the transition of training loss of CNN with increasing number of epochs (Loss rate v/s Number of epochs)

CONCLUSION AND FUTURE SCOPE

In this research, we have implemented three models for handwritten digit recognition using MNIST datasets, based on deep and machine learning algorithms. We compared them based on their characteristics to appraise the most accurate model among them. Support vector machines are one of the basic classifiers that's why it's faster than most algorithms and in this case, gives the maximum training accuracy rate but due to its simplicity, it's not possible to classify complex and ambiguous images as accurately as achieved with MLP and CNN algorithms. We have found that CNN gave the most accurate results for handwritten digit recognition. So, this makes us conclude that CNN is best suitable for any type of prediction problem including image data as an input. Next, by comparing execution time of the algorithms we have concluded that increasing the number of epochs without changing the configuration of the algorithm is useless because of the limitation of a certain model and we have noticed that after a certain number of epochs the model starts overfitting the dataset and give us the biased prediction

REFERENCES

[1] "Handwriting recognition": https : //en.wikipedia.org/wiki/Handwritingrecognition

[2] "What can a digit recognizer be used for?": https : //www.quora.com/W hat−can−a−digit−recognizer− be − used − f or

[3] "Handwritten Digit Recognition using Machine Learning Algorithms", S M Shamim, Mohammad Badrul Alam Miah, AngonaSarker, Masud Rana & Abdullah Al Jobair.

[4] "Handwritten Digit Recognition Using Deep Learning", Anuj Dutt and Aashi Dutt.

[5] "Handwritten recognition using SVM, KNN, and Neural networks", Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sharif.

[6] "Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers", Fathma Siddique, ShadmanSakib, Md. Abu Bakr Siddique.

[7] "Advancements in Image Classification using Convolutional Neural Network" by Farhana Sultana, Abu Sufian&Paramartha Dutta.

[8] "Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18 FFDG PET/CT images" by Hongkai Wang, Zongwei Zhou, Yingci Li, Zhonghua Chen, Peiou Lu, Wenzhi Wang, Wanyu Liu, and Lijuan Yu.

[9] https : //github.com/rishikakushwah16/SV M digit recognition using MNIST dataset

ISBN: 978-93-5782-920-5, www.aetsjournal.com ―Handbook of research in Big data Analytics, Artificial intelligence and Machine learning

[10] “Support Vector Machine Algorithm”: https : //www.javatpoint.com/machine−learning−support− vector − machine − algorithm

[11] https://machinelearningmastery.com/neural-networks-crash-course/

[12] “An Introduction to Convolutional Neural Networks”: researchgate.net/publication/285164623 An Introduction to Convolutional Neural Networks

[13] “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning”, ChigozieEnyinnaNwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall: https : //arxiv.org/pdf /1811.03378.pdf

[14] “Basic Overview of Convolutional Neural Network”: https://medium.com/dataseries/basic-overview-of-convolutional-neuralnetwork-cnn-4fcc7dbb4f17

[15] https://github.com/dixitritik17/Handwritten-Digit-Recognition .

[16] https://scikit-learn.org/stable/modules/svm.html

[17] https://en.wikipedia.org/wiki/LIBSVM .

[18] https : //github.com/F lame − Atlas/Handwritten Digit Recognition using MLP/blob/master/ANN.py

[19] https : //www.learnopencv.com/image − classification −using − feedforward − neural − network − in − keras/

[20] “How Do Convolutional Layers Work in Deep Learning Neural Networks”: https : //machinelearningmastery.com/convolutional −layers − f or − deep − learning − neural − networks/

[21] https://machinelearningmastery.com/rectified-linear-activation-functionfor-deep-learning-neural-networks/ .

[22] https://machinelearningmastery.com/pooling-layers-for-convolutionalneural-networks/ .

[23] https://medium.com/@amarbudhiraja/https-medium-comamarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machinelearning-74334da4bfc5 .

[24] https://medium.com/dataseries/basic-overview-of-convolutional-neuralnetwork-cnn-4fcc7dbb4f17 .

[25] https://medium.com/data-science-bootcamp/understand-the-softmaxfunction-in-minutes-f3a59641e86d .

ISBN: 978-93-5782-920-5, www.aetsjournal.com ―Handbook of research in Big data Analytics, Artificial intelligence and Machine learning