English Character Classification using Stochastic Gradient Descent

JOY PAL
18MCA0095
VIT University
joy.pal2018@vitstudent. ac.in

ABHIJEET GIRI
18MCA0088
VIT University
abhijeet.giri2018@vitstudent. ac.in

ARPIT MISHRA
18MCA0039
VIT University
arpit.mishra2018@vitstudent.ac.in

Abstract- Optical Character Recognition has been highlighted problem from the dawn of Image Processing and Machine Learning. Recognising a character from a picture in digital format have been research topics that seems to develop the way of recognition of characters of a particular language more efficient and faster detection. One of the application involves conversion of age old manuscripts into digital format in order to store it for its long durability and other involves application in Computer vision to make smart machine. From time, various machine learning problems such as KNN, SVM and even unsupervised learning such as Clustering have been involved for such recognition, but with rise in data size and image features detection we needed a more accurate and faster way of developing such recognizer. In this paper, we try to make a neural network model for recognition of such features from a pre developed extracted feature file for English Character and used Stochastic Gradient Descent approach for better results and performance.

Index Terms- Activation Function, Character Recognition, Dimension Reduction, Features Extraction, PCA, Neural Networks,

I. Introduction

Machine learning consists of many statistical as well as deep learning methods. In our approach we try to implement classification using artificial neural network. The data set we consider have been borrowed from UCI repositories under authentication of Data World Co. The enlightenment of this paper is based on the survey research of [1]. Both Optical and Hand Character recognition has been one of the hot and challenging topics of this era and the research on this still continues. Optical characters are easier to recognize than handwritten ones as there is scope of variance of shapes and structures depending on the type of handwriting of a person. The aim is to develop a model that recognizes the character with much higher accuracy without compromising both space and time complexity of the system. In this paper we have used features of 16000 English alphabets for classification and trained Neural Networks instead of using database to compare the classes as mentioned in algorithm [2].

II. Literature Survey

The author of this paper [3] have presented a skeptical approach for the recognition of English alphabets by first applying image processing like binarization and skeletalizational and then trained for in the neural network for the classification purposes. The paper concentrates on the importance of Segmentation of an image for faster classification rather than using all the pixel values for training in the neural network. For experimentation purpose, the authors have used first four english alphabets and carried out the whole process with an accuracy of 87.62%. Other than classification, they have improvised an anomaly known as “B-D” as stated in the paper that occurs frequently and is one of the main reason for misinterpretation of data.

Authors of [4] have displayed a wide range of application of handwritten character recognition. Processing applications include forms, digitizing ancient articles, postal address preprocessing, bamk cheque preprocessingand many others. In their survey a varieties of paper have been mentioned that classifies the data on various classification methods like SVM, K Nearest Neighbours etc. An effective recognition have been proposed by them that makes the recognition more neat.

In this paper [5], the authors have implemented Maximum tropy modeling and standardised it for the formulation of classification models for those areas where there is a presence of complex distribution of data which occurs due to mixing of simple underlying (latent) distributions. A theoretical framework have been devise for characterizing data as a mixture of maximum entropy models and thereby formulatio of a maximum-likelihood interpretation of mixture model learning and derive a generalized EM algorithm to solve the corresponding optimization problem. The overall method provides a significant improvement over the standard, single componet and maximum entropy approach.

In this paper [6], an Optical Character recognition system based on ariticial neural network have been showcasted. The proposed methodoogy states, each engish letter gets represented in binary numbers which is used as input to a simple extraction system whose output, in addition to the input, are fed to an ANN. Afterwards, the Feed Forward Algorithm gives insight into the enter workings of a neural network followed by the Back Propagation Algorithm which compromises Training, Calculating Error, and Modifying Weights. A simplistic approach for recognition of Optical characters using artificial neural networks has been described.

In [7] paper Pattern recognition was introduced including concept, method, application and integration. At the same time, ten definitions and more than ten methods of pattern recognition were summarized. Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network Function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this supervised learning, to train a network. Automatic Postal sorting, automatic bank cheque processing is application of Character recognition.

Handwritten character recognition is shown as an important discovery in pattern recognition applications is shown in [8]. Various applications of digit recognition including postal mail sorting, bank check processing, form data entry are formerly described in this paper. This paper presents an approach to off-line handwritten digit recognition based on different machine learning technique. The main objective of this paper is to ensure effective and reliable approaches for recognition of handwritten digits. Several machines learning algorithm namely, Multilayer Perceptron, Support Vector Machine, Naïve Bayes, Bayes Net, Random Forest, J48 and Random Tree has been used for the recognition of digits using WEKA.

Multilayer perceptron has been used for recognizing Handwritten English characters [9]. By tracing the boundary and their Fourier Descriptors the features are made. For differentiating each character analysis of its shapes is done and the features are compared .To get the result of high performance of back propagation network of hidden layers used is found and its analysis is done. A .In less time on training the data an accuracy of 94% on recognition was achieved in English handwritten letters.

By using the discrete and hybrid modelling techniques an off-line cursive recognition of handwriting is introduced which uses hidden Markov Models(HMM). Handwriting recognition experiments using a discrete and two different hybrid approaches, which consist of a discrete and semi- continuous structures, are compared. A segmentation free approach is considered to develop the system. It is found that the recognition rate performance can be improved of a hybrid modelling technique for HMMs, which depends on a neural vector quantizer (hybrid MMI), compared to discrete and hybrid HMMs, based on tired mixture structure (hybrid - TP), which may be caused by a relative small data set.

The author [11], used four feature extraction techniques namely, intersection, shadow feature, chain code histogram and straight line fitting features. Shadow features are computed globally for character image while intersection features, chain code histogram features and line fitting features are computed by dividing the character image into different segments. On experimentation with a dataset of 4900 samples the overall recognition rate observed was 92.80% for Devanagari characters.

Sushree Sangita Patnaik and Anup Kumar Panda May 2011 [12] et al, this paper proposes the implementation of particle swarm optimization (PSO) and bacterial foraging optimization (BFO) algorithms which are intended for optimal harmonic compensation by minimizing the undesirable losses occurring inside the APF itself. The efficiency and effectiveness of the implementation of two approaches are compared for two different conditions of supply. The total harmonic distortion (THD) in the source current which is a measure of APF performance is reduced drastically to nearly 1% by employing BFO. The results demonstrate that BFO outperforms the conventional and PSO based approaches by ensuring excellent functionality of APF and quick prevail over harmonics in the source current even under unbalanced supply.

In [13], Renata F. P. Neves have proposed SVM based offline handwritten digit recognition. Authors claim that SVM outperforms the Multilayer perceptron classifier. Experiment is carried out on NIST SD19 standard dataset. Advantage of MLP is that it is able to segment non-linearly separable classes. However, MLP can easily fall into a region of local minimum, where the training will stop assuming it has achieved an optimal point in the error surface. Another hindrance is defining the best network architecture to solve the problem, considering the number of layers and the number of perceptron in each hidden layer. Because of these disadvantages, a digit recognizer using the MLP structure may not produce the desired low error rate.

G. Pirlo and D. Impedovo in his work on [14], presented a new class of membership functions, which are called Fuzzy membership functions (FMFs), for zoning-based classification. These FMFs can be easily adapted to the specific characteristics of a classification problem in order to maximize classification performance. In this research, a realcoded genetic algorithm is presented to find, in a single optimization procedure, the optimal FMF, together with the optimal zoning described by Voronoi tessellation. The experimental results, which are carried out in the field of handwritten digit and character recognition, indicate that optimal FMF performs better than other membership functions based on abstract level, ranked-level, and measurement-level weighting models, which can be found in the literature. In this paper [15], the methods like “auto encoder support vector machine” has been implemented for the extraction of the feature in a handwritten alphabets. A detailed discussion has been carried out about the artificial neural networks and it even potrays a survey report that highlights the selection of authentic methods plays an vital role in performance of character recognition rate.

III. Methodology

For the implementation of the Artificial Neural Netwrok we followed certain procedures. It includes procedures such as Data Extraction, Data Preprocessing, Normalization, FeedForward Training and Back Propagation.

A. Data Extraction

We have taken our data from UCI repositories.

The data consists of 26 class labels [A-Z] and 16000 tuples. We are going to perform training with 80% of data and rest data is kept for testing purposes. The dataset consist of 16 attributes or independent variables based on which the class labels gets classified. As, the number of training data is more we will mostly reliable on the accuracy of training data.

Table 1.
Attributes Information

Attribute Name	**Information**
1.letter	Captial Letter
2. x-box	Horizontal Position of box
3. y-box	Vertical Position of box
4. width	Width of the box
5. High	Height of the box
6. onpix	Total no of pixel
7. x-bar	Mean x of on pixel in box
8. y-bar	Mean y of on pixel in box
9. x2bar	Mean x variance
10. y2bar	Mean y variance
11. xybar	Mean x*y correlation
12. x2ybar	Mean of x*x*y
13. xy2bar	Mean of x*y*y
14. x-edge	Mean edge count left to right
15. eegvy	Correlation of x-edge with y
16, y-edge	Mean edge count bottom to top
17. yegvy	Correlation of y-edge with x

B. Data Pre-processing

This step involves preparation of the dataset to its most accurate format in order to increase the precision of classification. The major steps involved in data pre processing are:

- Noise Removal
- Encoding categorical data
- Data Normalization
- Splitting of Data

Noise in a tabular data can be of three types. First one is Anomalies that generally occurs due to presence of outliers. Second to that comes Irrelevant Data, that are generally weak data. Third are those records which don"t follow the form or relation which rest of records do. Encoding Categorical data involves the conversion of Categorical data to numerical data.

Data Normalization is a scaling technique or a mapping technique or a pre processing stage. Where we can find new range from an existing one range. It can be helpful for the prediction or forecasting purposes. We are going to utilise Z score normalization. Here unstructured data can be normalized using z- score parameter as per given formulae:

$$v_i' = \frac{v_i - E}{std(E)}$$

Where,

v_i'' is normalized one values
v_i is value of the row E of ith column

$$std(E) = \sqrt{\frac{1}{(n-1)} \sum_{i=1}^{n} (v_i - \quad)^2}$$

$$E = \frac{1}{n} \sum_{i=1}^{n} v_i \quad or\ mean\ value$$

C. Artificial Neural Network

Artificial Neural Network consists of two steps of classification, Feed-Forward Neural Network and Backward Propagation. Generally, neural network consists of Input Layer, n number of Hidden Layer and one Output Layer. As stated in the diagram, we give our input in the input layer via nodes in the input layer. It then goes into transition by activation function.

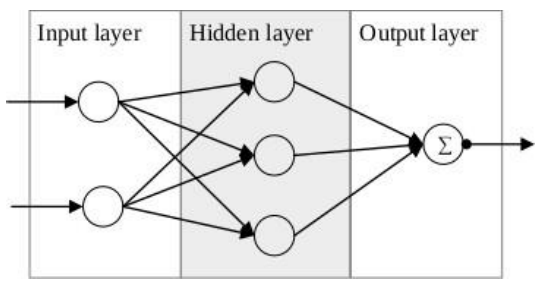

Fig 1. General diagram of ANN

In our case, we have used two types of activation function. From last hidden layer to the output layer we have utilized Softmax function as our activation function as it ranges between 0 and 1, and as the number of classes in our classification is 26, we use Softmax function which is used for multiclass classification.

1. Softmax Function:

$$F(x_i) = \frac{Exp(x_i)}{\sum_{j=0}^{k} Exp(x_j)}, \quad \text{where}$$

i=0,1,2,. ...k

Softmax function calculates the probabilities distribution of the event over „n" different events. In general way of saying, this function will calculate the probabilities of each target class over all possibilities of target classes.

2. Relu Function:

Relu function is the most used function in the world as it is used in all most all convolutional neural network and deep learning.

D. Batch Gradient Descent vs Stochastic Gradient Descent Algorithm

Both of the process is used for the back propagation of the neural networks, for the updatation of the weights in each layers. The former updates the weights after all the training data has been feed forwarded in the network and then count for the error by

$c = \sum_{i=0}^{n} \frac{1}{2}(y - \hat{y})^2$, here „n" is the number of each data records.

But the Stochastic Gradient descent update each weights after ever training data been forwarded in the neural network. The loss function used for this gradient descent algorithm when sigmoid activation function is used is

$$logarithimic\ loss = -\frac{1}{n}\sum_{i=1}^{n}\left(y^i \log \phi\left(\sum_{j=1}^{m} w_j x_j^i\right) + (1-y^i)\log\left(1 - \phi\left(\sum_{j=1}^{m} w_j x_j^i\right)\right)\right)$$

IV. Experimental Results And Discussion

A. Feature Space Reduction

We have often heard about the term "Curse of Dimensionality" in fields like Machine Learning and Data Science. It is used to refer the condition that arises when number of attributed taken for regression and classification is too much the model fails to perform in its optimal structure perfectly. Too many attributes makes the data abrupt and it needs more time and memory at the time of classification, thus compromising the resources. In order to avoid it we take features from the data sets that are highly correlated with the output. Principal Component Analysis is one of the commonly used statistical tools that are used to find the most viable features from the given set of features that makes its most impact on the result. Other tool like Linear Discriminant Analysis too helps in finding the best features from the data sets.

B. Results and Comparison

For the evaluation of performance and model selection we are going to look at the Confusion Matrix and compare the F_{score}. A Confusion matrix consists of four cells when the number of classification to be made is two in our case, and nine when number of classification is three and so on. It generally consist of TP,TN, FP, and FN represent True Positive, True Negative, False Positive, and False Negative, respectively. True positive and True negative are the positive and negative tuples respectively that are being correctly labelled by classifier. False positive and false negative are positive and negative tuples respectively that are either mislabelled or incorrectly labelled by the classifier. From the confusion matrix we are going to find the accuracy rate, error rate and consequently f_{score} for each model.

$$\textbf{accuracy recognition rate} = \frac{(t_p + t_n)}{p+n}$$

$$\textbf{error rate} = \frac{(F_p + F_n)}{(p+n)}$$

From the experimentation we devised following result:

Table 2.

Results before performing PCA

Batch Size	**Number of Epochs**	**Accuracy**

10	60	79.22
15	60	61.83
25	90	85.41

After Applying PCA: n=8,

Table 3
Results After applying PCA

Batch Size	Number of Epochs	Accuracy
10	60	81.56
15	60	60.65
25	90	86.68

CODE

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# Data Preprocessing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('data1.csv')
X = dataset.iloc[:, 2:18].values
y = dataset['letter-name']

# Encoding categorical data
from sklearn.preprocessing import LabelEncoder,
OneHotEncoder
labelencoder_X_1 = LabelEncoder()
'''
X[:, 1] = labelencoder_X_1.fit_transform(X[:, 1])
labelencoder_X_2 = LabelEncoder()
X[:, 2] = labelencoder_X_2.fit_transform(X[:, 2])
onehotencoder = OneHotEncoder(categorical_features =
[1])
X = onehotencoder.fit_transform(X).toarray()
X = X[:, 1:]
'''
labelencoder_y = LabelEncoder()
y = labelencoder_y.fit_transform(y)
yy=labelencoder_y.fit_transform(y)

from keras.utils import to_categorical
y = to_categorical(y)

#Convert back to numerical data
from numpy import argmax
ytret=[argmax(a) for a in y]
ytret=np.asarray(ytret)

#y=y/25

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.2, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Part 2 - Now let's make the ANN!

# Importing the Keras libraries and packages
import keras
from keras.models import Sequential
from keras.layers import Dense

# Initialising the ANN
classifier = Sequential()

# Adding the input layer and the first hidden layer
classifier.add(Dense(output_dim = 16, init = 'uniform',
activation = 'relu', input_dim = 16))

# Adding the second hidden layer
classifier.add(Dense(output_dim = 16, init = 'uniform',
activation = 'relu'))

# Adding the third hidden layer
classifier.add(Dense(output_dim = 16, init = 'uniform',
activation = 'relu'))

# Adding the output layer
classifier.add(Dense(output_dim = 26, init = 'uniform',
activation = 'softmax'))

# Compiling the ANN
classifier.compile(optimizer = 'sgd', loss =
'categorical_crossentropy', metrics = ['accuracy'])
# optimizer= rmsprop/adam/
# Fitting the ANN to the Training set

classifier.fit(X_train, y_train, batch_size = 10, nb_epoch =
60)
classifier.fit(X_train, y_train, batch_size = 15, nb_epoch =
60)
classifier.fit(X_train, y_train, batch_size = 25, nb_epoch =
90)

#batch_size=10, nb_epoch=100
#batch_size=128, nb_epoch=20
# Part 3 - Making the predictions and evaluating the model

# Predicting the Test set results
y_pred = classifier.predict(X_test)
y_pred = (y_pred > 0.5)


y_test=[argmax(a) for a in y_test]
y_test1=np.asarray(y_test)

y_pred1=[argmax(a) for a in y_pred]
y_pred1=np.asarray(y_pred1)



# Making the Confusion Matrix
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test1, y_pred1)

from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_test1,y_pred1)
print(accuracy*100)
```

V. Conclusion

From the above conducted experiments, we can conclude that Artificial Neural Network is the future of what is known as Machine Learning. With days passed, number and features of data classification increase in an exponential manner, causing normal statistical classifier to take more space and time than expected. Our experiments, made an improvement of 86.72% accuracy over the original conducted experiment in 1999 whose accuracy was 76.8%. The experiment can further be improved by inclusion of recurrent neural network and even conducting an image processing techniques parallel, so that real time experiments can be performed on the data.

VI. References

[1] Noman Islam, Zeeshan Islam, Nazia Noor, "A Survey on Optical Character Recognition System", *Journal of Information & Communication TechnologyJICT*, Vol. 10 Issue. 2, December 2016

[2]Sukhpreet Singh, "Optical Character Recognition Techniques: A survey", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, ISSN: 2278 – 1323 Volume 2, Issue 6, June 2013

[3]Suraj Saha, Mythili G Y, "Optical and Handwritten Character Recognition System Using Segmentation and Neural Network", *Advances in Computational Sciences and Technology*, ISSN 0973-6107, Volume 11, Number 7 (2018) pp. 547-554

[4] Shivangkumar R Patel, Jasmine Jha, "Handwritten Charater Recognitiom using Machine Learning Approach- A Survey", *International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, DOI:10.11 09/eesco.2015.7253978, January 2015.

[5] Dimitry Pavlov, Alexandrin Popescul, David M. Penncok, Lyle H. Ungar, "Mixtures of Conditional Maximum Entropy Models", *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, Page: 584-591, August 21, 2003

[6] P. B. Kumbhar, A. N. Holambe, "Survey On Optical Character Recognition Using Neural Network", *International Journal for Research in Engineering Application & Management (IJREAM)*, ISSN: 2454-9150, Vol-03, Issue-03, June 2017

[7] Jie Liu, Jigui Sun, Shengsheng Wang, "Pattern Recognition: An Overview", *International Journal of Computer Science and Network Security*, VOL.6 No.6, June 2006.

[8] S. M. Shamim, Md Badrul Alam Miah, Angona Sarkar, "Handwritten Digit Recognition using Machine Learning Algoithms", *Indonesian Journal of Science and Technology*, DOI:10.17509/ijost, vol3il.10975, March 2018

[9] Anita Pal & Dayashankar Singh, "Handwritten English Character Recognition Using Neural," *Network International Journal of Computer Science & Communication*, Vol. 1, No. 2, July-December 2010, pp. 141-144.

[10] A. Brakensiek, J. Rottland, A. Kosmala and J. Rigoll, "Offline Handwriting Recognition using various Hybrid Modeling Techniques & Character N-Grams", Available at http://irs.ub.rug.nl/dbi/4357a84695495.

[11] Sandhya Arora, “Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition”, *IEEE Region 10 Colloquium and the Third ICIIS*, Kharagpur, INDIA, December 2008.

[12] Sushree Sangita Patnaik and Anup Kumar Panda, “Particle Swarm Optimization and Bacterial Foraging Optimization Techniques for Optimal Current Harmonic Mitigation by Employing Active Power Filter”, *Applied Computational Intelligence and Soft Computing*, Volume 2012, Article ID 897127.

[13] Renata F. P. Neves, Alberto N. G. Lopes Filho, Carlos A.B.Mello, CleberZanchettin, “A SVM Based Off-Line Handwritten Digit Recognizer”, *International conference on Systems, Man and Cybernetics, IEEE Xplore*, pp. 510-515, 9-12 Oct, 2011, Brazil.

[14] G. Pirlo and D. Impedovo, “Fuzzy Zoning Based Classification for Handwritten Characters”, *IEEE Transaction on pattern Recognition and Machine Intelligence*, vol.19, no. 04, pp.780-785, August 2011.

[15]Cinu George, S. Podhumani, “Survey on Handwritten Character Recognition using Artificial Neural Network”, *International Journal of Science Technology &Engineering*, Vol-2, Issue10, April 2016