Project report for letter recognition using Holland-Style Adaptive Classifier

Ashish Pandey

***Abstract*—This paper uses various machine learning algorithms, for pattern classification, to identify each of the large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the english alphabet. The character images were based on 20 different fonts and each letter within 20 fonts was randomly distorted to produce a file of 20,000 unique instances. The features of the dataset and the errors committed by Holland-style adaptive classifiers were analyzed in an attempt to use ML algorithms in-order to reduce the error rate.**

I. Introduction

The letter image recognition data was donated by David J. Slate and P.W. Frey in 1991 to UCI data repository for the researchers and scientists to analyze the patterns in efficient way. In this paper SVC,KNN,logistic regression and neural network is used to recognize and classify the pattern of English capital alphabet among 26-alphabet classes. Improvement is gained by reducing the error rate down to 2%.

II. Dataset

In this data, a set of 20,000 unique letter images was generated by randomly distorting pixel images of the 26 uppercase letters from 20 different commercial fonts. The parent fonts represented a full range of character types including script, italic, serif, and Gothic. Each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. The features of each of the 20,000 characters or stimulus were summarized in terms of 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15. The objective is to identify each black-and-white rectangular pixel display as one of the 26 capital letters in the English alphabet[1].

III. Image Feature Description

For each black-and-white image of the English alphabet, 16-dimensional feature vector was extracted by the author to demonstrate the summary of the alphabet image [1]. This feature vector contains the characteristic features of the image such as vertical and horizontal position of the rectangular box containing the alphabet, total number of ON pixels, edge count etc. The full description of the feature vector can be found in [1].

Ashish Pandey (e-mail: ap696159@ieee.org)

IV. Accuracy Review

The lowest classification error rate of 17% was observed by the Frey and Slate by applying Holland-style adaptive classification [1]. About 16,000 stimuli were observed under training and remaining 4000 were tested for classification. Improvement of 15% accuracy was achieved by applying Support vector machine classification.

V. Building logistic regression model

Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Model was implemented with the help of scikit learn library, the model gave an accuracy of 71.41%

VI. Building SVC model

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

Model used a "rbf" kernel and gave an accuracy of 97.25%.

VII. Building KNN model

The k nearest neighbor algorithm rely on majority. Example for k=3, it will see the nearest 3 points to the test data and count the category group the class having maximum number of the points, that class will be assigned to the data.

Model used k = 5 and got accuracy of 98.25%.

VIII. Building neural network model

The neural network was build using pytorch. The architecture of model was as follows:

Linear 16X32 , 1D batch norm(32) , relu()L,inear(32,64),relu()

The batch norm was added so that model could train faster without overfitting.

Adam optimizer was used as it converges faster for global minima.

relu activation function was used as it was computational efficient as it does not activate all neuron at once.

The model was trained for 100 epochs and the accuracy achieved was 93.55 %.

IX. Conclusion

This paper compares various machine learning algorithms for efficiently classifying letters.

The knn gave the best performance, it can be seen by the algorithm of knn as it will look for the nearest data points for classifying purpose based on the majority.

X. Reference

[1] P. W. Frey and D. J. Slate, Letter Recognition using Holland-style adaptive classifiers, Machine Learning, vol. 6, pp. 161-182 ,1991.