PROJECT TITLE:

Digit Recognizer: A comparison of different classifier techniques and Hypothesis testing

TEAM:

Ankur Garg (agarg12), Abhimanyu Jataria (ajatari), Shriyansh Yadav (scyadav)

INTRODUCTION:

In this project, our goal is to train models to identify digits from a large dataset of handwritten images. Our objective is to compare these models and find out which model gives a better accuracy. We will use various statistical methodologies to verify the accuracies of the cross validation on this models.

MNIST ("Modified National Institute of Standards and Technology") is a dataset of computer vision. Since its release, this dataset of handwritten images has served as the basis for benchmarking classification algorithms. For the newly emerging machine learning techniques, MNIST remains a reliable resource for researchers and learners alike. The original data was 42000 grayscale images of resolution 28x28. Each image was represented as a vector of size 784, with each value representing a pixel between 0 and 255. Each image has a class label associated with it which takes values from {0,1,2,3,4,5,6,7,8,9}, representing one of the numerical digits.

BACKGROUND:

Following research papers were used as reference. The paragraphs below are the short summaries of the research papers read to understand and approach the digit recognizer problem.

Paper [1]:
This is the main paper for our project implementation. The paper focuses on comparison of classifiers based on error rate, time and storage requirements for handwritten digit recognizer problem. It compares the classifiers like basic linear classifier, k-Nearest Neighbor classifier, as well as variants of neural networks. Since the paper represents the in-process experiment, the authors have very well described the comparison using plots. We have used the same idea and implemented the classifiers using Logistic Regression, Random Forest, Maximum Likelihood Classifier (MLC) and Neural Networks. We have also plotted the results for the comparison based on the accuracy and time taken for 10-fold cross validation.

Paper [2]:
Handwritten digit recognition: benchmarking of state-of-the-art techniques.
This paper explains the feature extraction and classification techniques paired together to recognize handwritten digits. The authors have very well explained their experiment that involved test databases (CENPARMI, CEDAR, and MNIST). They have also explained the reasons for pre-processing of data, required to achieve uniformity. On the test datasets, 80 recognition accuracies are given by combining eight classifiers with ten feature vectors. Due to the accuracy of direction features in characterizing the within-class shape invariance and between class differences, for feature extraction the experiments used variants of direction features. Among them, the gradient feature extraction technique with Kirsh operator gives the best result. It gives eight directional elements' strengths from each sample image. Thus, it explained that this feature extraction phase is the baseline for the classification phase, because if good features can be extracted in an earlier stage, the image

classification in the latter stage will also be of higher accuracy. Classification is done using 5 types of classifiers - k-Nearest Neighbor classifier, 3 variants of neural classifiers, and 2 support vector classifiers. The outcome of the experiment is, that good accuracies are achieved through the neural classifiers. From the results of this experiment, we understood that the features extracted in the experiments are best classified or detected by the neural classifiers and hence, for anyone doing pattern recognition can get competitive results if neural classifiers are used. A point to note is that, the results are dependent on extracted features as well. Hence, feature extraction algorithms need to be modified as per available data to achieve the optimal results. Thus, even we have used dimensionality reduction to get the important features in order to recognize the digit in the image.

Paper [3]:
This was one of the recent papers on handwritten digit classification, which explained that even a shallow neural network can be used to solve this problem with the use of ELM- Extreme Learning Machine Algorithm. It explained the various variants of ELM like, Computed Input Weights (CIW-ELM), Constrained ELM (C-ELM), and Receptive Fields ELM (RF-ELM). The combination of these variants were used to obtain reduced error rate in the classification of information. This paper explained the topics that seemed out of scope for the current implementation, but was useful in explaining to us that deep neural networks is not the only best solution for digit recognition problem, but even shallow neural network can be used to solve the problem with the use of ELM and its variants. This paper gives us an overview of some of the complex algorithms that can be implemented to obtain better outcomes for the problems similar to the digit recognizer problem.

Paper [4]:
Having come across various classifiers, among which, were the ones we implemented, this paper gives us an insight about how the classifier accuracy is not the only aspect to consider, but also the reliability of the classification obtained is important. Therefore, in order have a high reliability of the outcome, along with the high accuracy, it explains of a hybrid model which uses, the two most used classifiers, Convolution Neural Network (CNN) and Support Vector machine (SVM). The CNN has the advantage that it automatically extracts the salient features of the image, which are invariant to the shift and shape distortions. The last connected layer of the CNN is replaced by the SVM classifier to predict the input patterns [4]. This hybrid approach uses, both recognition accuracy and reliability performance as the evaluation metric. Because of this paper we understood the fact, that, however accurate results the classifiers we have learned in the course produced, but a combination of classifiers can still help increase the reliability of a model.

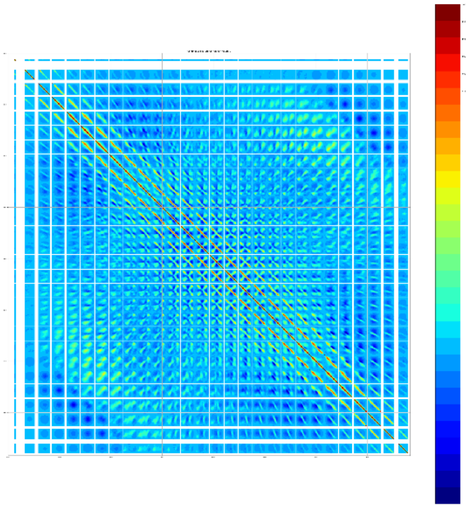
BUSINESS APPLICATIONS:

1. CCTV Footage:
We can use this approach in recognizing the number plate of vehicles by CCTV recordings, to tighten the law enforcement and reduction in road crime.
2. Object Recognizer:
In training an AI agent for self-driven car or industrial robot, this approach is helpful in recognizing the objects so that its knowledge base can be made strong and it does not confuse between objects.
3. Postal Services:
In Post Offices this model can be used in recognizing the hand written zip codes to process the mails to desired destination.
4. Bank Checks:
In recognizing the signature of a customer, so that bank fraudulent can be controlled.

In our model we have approached the Postal Services problem, where the post offices faces problem in recognizing the hand written zip codes, due to the different writing methods that may involve missing strokes, broken numerals, intruder noises and stroke touching [4]. Using our model, given the image of the data we can classify the digits.

METHODOLOGY:

Preprocessing:
Before moving forward with the task of classification, it is important to check if the features being used from the data are correlated or not. Presence of correlated features can cause a classifier to fit incorrect models. Upon analysis, it was found that the features weren't highly correlated. The figure below shows the correlation matrix for the various features. It can be seen that, most pair of features were uncorrelated.



Based on these results, it was found that none of the features were required to be removed.

Dimensionality:
Because of the high dimensionality of the data, it was required that some form of preprocessing is done to reduce the dimensions in the data. Since the images were grayscale, apart from the part of the image where the digit is actually written, remaining part would be pretty similar in different images. So, it doesn't make sense to include such features which do not contain any information which can differentiate between the different classes.
Higher number of dimensions cause many classifiers to become complex and cause overfitting on the training data. This was one of the major reasons for reducing dimensions.

To reduce dimensionality, process of Principal Component Analysis was used on the original set of features. Before applying the principal component analysis, the data was normalized. Otherwise, features with high variance would end up contributing most to the principal components. For this dataset, it wasn't a problem as all features were between the same range [0,255].

After rescaling, principal component analysis was applied to the dataset. It was observed that 150 principal components represented approximately 95% of the variance, 200 components had 96.5% and 300 features had 98.3% of the variance of the original data. To ensure that reduction in the number of features wasn't causing any loss of information which could affect the results, it was decided that different number of principal components would be used to train and test different models to find the appropriate number of principal components. It was observed that the models trained on 300 features resulted in the best accuracies.

EXPERIMENTS:

For the task of classification multiple classifiers were used. For each method, 10-fold cross validation was used to obtain accuracy distribution of each model. Further the results obtained were compared using paired t-test and ANOVA tests. The various classifiers used were:

(1) Maximum Likelihood Classifier:

To begin with, Maximum Likelihood Classifier was implemented in python using *numpy* and *pandas* libraries. For implementing MLC, a multivariate Gaussian distribution was fitted on the data belonging to each class. Using these distributions, likelihood probabilities were calculated for test samples.
Optimization: For improving the runtime of the classifier while predicting on test samples, the classifier was implemented using *numpy* vector operations. It ensured that predictions for multiple test samples could be done at the same time.
During the implementation, it was observed that likelihood probabilities were getting quite small and ended up being outside the precision range of the python floating point data type. To avoid that, natural logarithm was used. It didn't affect the predictions as comparison among the probabilities was used to decide the class of the test sample.

(2) Logistic Regression:

Using python library *sklearn*, Logistic Regression was fitted using parameters: max number of iterations 100 and L2 norm penalty.

(3) Random Forest Classifier:

Random Forest Classifier was used to train a classifier on the data obtained after PCA. The parameters used for the Random Forest were: 100 trees, Gini Index Criterion along with sampling with replacement. For this, python package *sklearn* was used.

(4) Neural Network:

The neural network with 2 hidden layer was trained using the various parameter settings. The final settings used are described as follows: The first hidden layer contained 100 neurons with uniform kernel initialization and *relu* activation function. The second layer contained 50 neurons again with *relu* activation. The output layer had 10 nodes, each for one of the 10 classes, with 'normal' initialization and *sigmoid* activation.
Loss function used for training the model was categorical cross entropy. The model was trained on 100 epochs with a batch size of 800.

For training a neural network, python library *keras* was used which is built on top of the tensor-flow backend.

Comparison of various algorithms:
The 4 methods used to train a classifier were compared using various statistical tests (t-test and ANOVA). The accuracy measures for each method were collected using 10-fold cross validation.

The results obtained for each of the algorithms along with the comparison among with various algorithms is explained in the next section.

RESULTS:

(1) The cross-validation accuracies obtained for the 4 methods are listed in the table below:

Folds	Random Forest	Logistic Regression	MLC	Neural Network
0	0.975737393	0.901760228	0.926666667	0.982142857
1	0.973352367	0.911729717	0.922619048	0.98047619
2	0.975017844	0.909350464	0.932142857	0.984285714
3	0.969545563	0.909826315	0.934285714	0.982380952
4	0.973339681	0.907402999	0.92452381	0.980952381
5	0.975946654	0.921886163	0.934285714	0.982142857
6	0.973797046	0.90686041	0.929047619	0.97952381
7	0.968794664	0.908766079	0.923809524	0.98
8	0.969256435	0.911344137	0.929761905	0.979047619
9	0.972586412	0.910846246	0.933333333	0.978333333

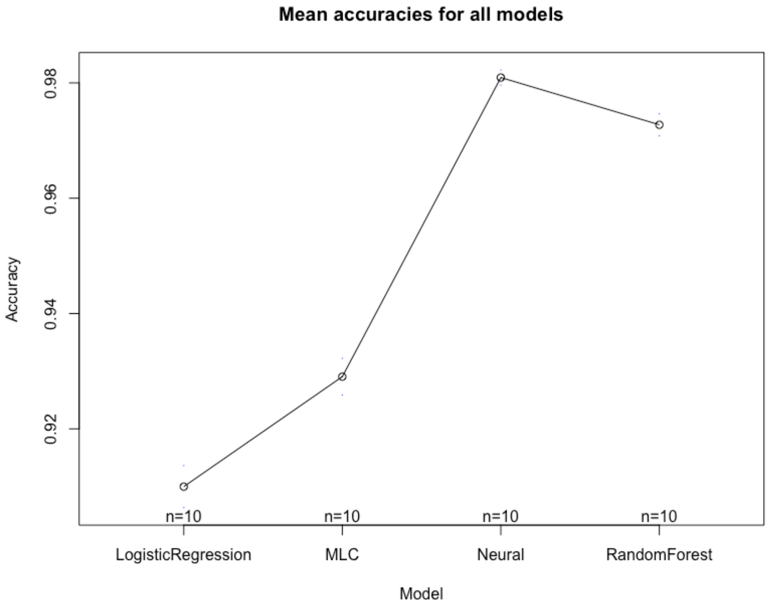
(2) Comparison:

Each pair of the four methods was compared using paired t-test. It was observed the pval for the t tests were all very small. Indicating that, we can reject the null hypothesis that the means of the cross-validation accuracies are same. P-values obtained are listed below:

Algorithm 1	Algorithm 2	T-test p-value
Random Forest	Logistic Regression	5.414e-11
Random Forest	MLC	4.75e-10
Random Forest	Neural Network	2.434e-06
Logistic Regression	MLC	8.129e-07
Logistic Regression	Neural Network	1.272e-11
MLC	Neural Network	3.587e-11

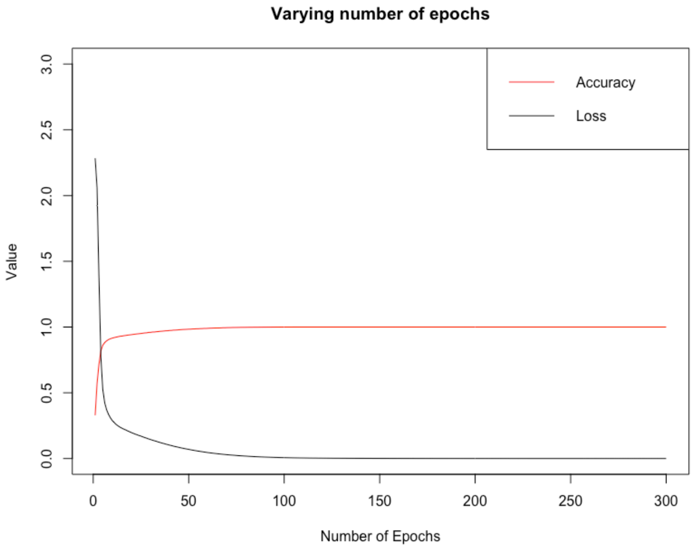
(3) Based on these p-values, it can be concluded that, the mean accuracies obtained from the 4 methods used are statistically significantly different from each other.

(4) The mean of the cross-validation accuracies of the 4 algorithms was compared using the plot below:

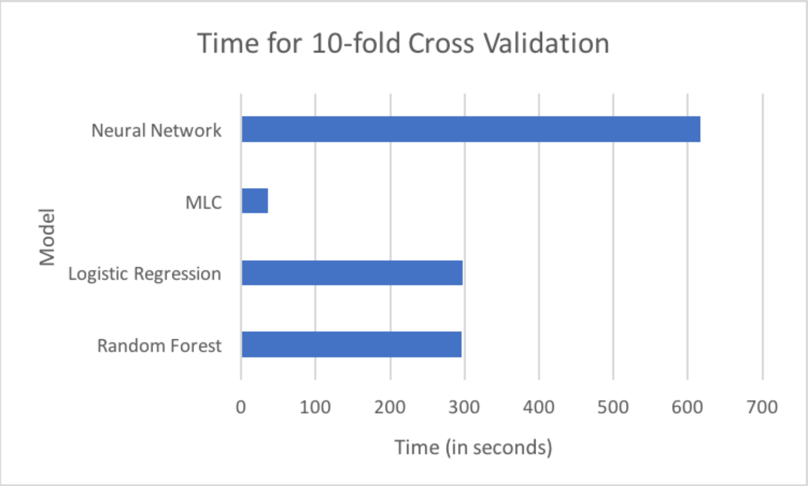


(5) ANOVA test was also used to compare the cross-validation scores for the 4 algorithms. The p-value obtained was 2e-16. This is a very small value and shows that we can reject the null hypothesis of the ANOVA test. This shows that the mean of the cross-validation accuracies is significantly different from each other.

(6) The number of epochs used for neural network was varied to check how the accuracy improved. The plot obtained is shown below:



(7) The various algorithms were also compared based on time taken for running 10-fold cross validation on the dataset. The time taken by each algorithm is shown in the plot below:



CONCLUSION:

(1) Hypothesis Testing:
Based on the t-test, ANOVA test and the mean-plot, it can be concluded that, neural network with the given set of parameters performed the best of all the four algorithms with an average cross-validation accuracy of 98.09%, followed by Random Forest algorithm with an average accuracy of 97.27%. Logistic Regression and MLC performed significantly worse as compared to these two algorithms.

(2) Avoiding Overfitting:
Also, it was observed that the performance of the neural network is quite dependent on the various parameters. For example, increasing the number of epochs certainly increases the training accuracy but it leads to overfitting. So, deciding the number of epochs is crucial to the performance. To find optimal number of epochs, the figure [] was used. Based on this plot, appropriate number of epochs was selected to be around 90-100.

(3) Analysis of runtime:
It was also observed that neural network (with the specifications mentioned in previous section) was computationally most expensive. Logistic Regression and Random Forest were approximately similar but significantly less than the Neural Network. Maximum Likelihood Classifier was the fastest out of all four. The reason behind this is that as part of the training process, MLC just needs to compute the mean vector and covariance matrix for each class, which takes considerably less time when compared to other algorithms.

REFERENCES:

[1] ‘Comparison of Learning Algorithms for Handwritten Digit Recognition’, Y. Lecun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J.Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, and V. Vapnik Bell Laboratories, Holmdel, NJ 07733, USA

[2] ‘Handwritten digit recognition: benchmarking of state-of-the-art techniques’, Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, Hiromichi Fujisawa, Central Research Laboratory, Hitachi, Ltd. 1-280 Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan.

[3] ‘Fast, Simple and Accurate Handwritten Digit Classification by Training Shallow Neural Network Classifiers with the ‘Extreme Learning Machine’ Algorithm’, Mark D. McDonnell, Migel D. Tissera, Tony Vladusich, André van Schaik, Jonathan Tapson.

[4] ‘A novel hybrid CNN–SVM classifier for recognizing handwritten digits’, Xiao-Xiao Niun, Ching Y. Suen Centre for Pattern Recognition and Machine Intelligence, Concordia University, SuiteEV003.403, 1455 de Maisonneuve Blvd. West, Montreal, Quebec, Canada H3G 1M8.