Table Of Contents

Team Details

Member 1

Name - Sharoon Saxena 0902IT151045

Address - 13/2 HIG Geetanjali Complex T.T. Nagar Bhopal, M.P.

Mobile Number - 9425096938

E-mail - charismaticsharoon@gmail.com

Member 2

Name - Yogender Singh

Address - House 52/10 Krishna Nagar Jamuhan Dibiyapur Auraiya, U.P.

Mobile Number - 9039430719

E-mail - syogender799@gmail.com

Introduction

Image classification is an important problem which appears in many application domains like quality control, biometry (face recognition), medicine, office automation (character recognition), geology(soil type recognition)…

This problem is traditionally difficult for the Machine Learning algorithms mainly because of the high number of input variables, learning methods often suffer from a very high variance (models are very unstable) which deteriorates their accuracy. Furthermore, computing times can be detrimental in such extreme conditions.

To handle this high dimensionality, image classification systems usually rely on preprocessing step, specific to particular problem and application domain , which aims at extracting a reduced set of interesting features from initial huge number of pixels.

Here , MNIST database consists of 70000 handwritten digits that have been size normalised and entered in images of 28 x 28 pixels with 256 grey levels per pixel, which we believe is one of the finest examples of the problem mentioned above.

Objective

To implement generic machine learning algorithms over an image classifier.

Optimise them to yield the maximum efficiency possible.

Evaluate them using various parameters and metrics.

Compare them against each other using metrics such as Accuracy and Time taken to Train and time taken to Predict.

Also comparing by using self-defined metrics such as Train-Efficiency and Predict-Efficiency.

Research Methodology

This project is about implementing the generic machine learning algorithms over an image classifier.

MNIST dataset is used in undertaking of this research project.

- The dataset consists of images of handwritten digits.
- It contains 70000 images of numbers from 0 to 9.
- The images are present in grayscale.
- Every image is a square shaped 28x28 pixel with image centred using the centre of mass of the pixels.

This project is based on supervised learning where each image is LABELLED according to the image data from which the Machine learning classification algorithms learn.

Following algorithms will be used:

K-NN (K- Nearest Neighbours)

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:Support Vector Machines.

In *k-NN classification*, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

Naive Bayes

In machine learning, naive Bayes classifiers (sometimes called the idiot Bayes model are a family of simple "probabilistic classifiers "based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

It is a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis.

It is called naive bayes because it assumes strong independence among the features which is often not true.

SVM (Support Vector Machines)

In machine learning, support vector machines (SVMs, also support vector networks are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier .

The SVM algorithm has been widely applied in the biological and other sciences. They have been used to classify proteins with up to 90% of the compounds classified correctly. Permutation tests based on SVM weights have been suggested as a mechanism for interpretation of SVM models. Support vector machine weights have also been used to interpret SVM models in the past. Posthoc interpretation of support vector machine models in order to identify features used by the model to make predictions is a relatively new area of research with special significance in the biological sciences.

Decision Tree

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making).

Following is the procedure followed in this project.:

Data Preprocessing :

It is the act of cleaning and arranging data such that it is ready to be trained.

The MINIST dataset set used was the preprocessed by the carters of the dataset itself.

But we have added the Principle Component Analysis , so as to reduce the training time by reducing the number of features while retaining the 95% variance.

Training Model:

The data will now be trained using the above mentioned Supervised Classification algorithms**.**

Supervised Learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.[1] It infers a function from labeled training data consisting of a set of training examples.[2] In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way

In order to solve a given problem of supervised learning, one has to perform the following steps:

1. Determine the type of training examples. Before doing anything else, the user should decide what kind of data is to be used as a training set. In case of handwriting analysis, for example, this might be a single handwritten character, an entire handwritten word, or an entire line of handwriting.

2. Gather a training set. The training set needs to be representative of the real-world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered, either from human experts or from measurements.
3. Determine the input feature representation of the learned function. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should contain enough information to accurately predict the output.
4. Determine the structure of the learned function and corresponding learning algorithm. For example, the engineer may choose anu suitable algorithm
5. Complete the design. Run the learning algorithm on the gathered training set. Some supervised learning algorithms require the user to determine certain control parameters. These parameters may be adjusted by optimizing performance on a subset (called a validation set) of the training set, or via cross-validation.
6. Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

Testing model:

In this the model will be tested upon a dataset which is unknown to the trained model i.e. the training set and the test set are mutually exclusive**.**

Evaluation:

Once all the models are optimized they are evaluated using the metrics like Accuracy, Training time and overall efficiency.

Visualisation

All the models are evaluated agains each other using the detailed visualisations and graphs.

Expected Outcomes

- The Decision tree algorithm is expected to come over all the other algorithms with respect to time.

  It Has been observed that Decision Tree Algorithm performs exceptionally well over the training set, but it generally tends to overfit the training set, i.e. the accuracy difference between the training set and the test set is significantly large.

  which is not desirable.

- Naive Baye's Algorithm is expected to under Perform as it performs poorly over the image processing and therefore overall accuracy is expected to be low.

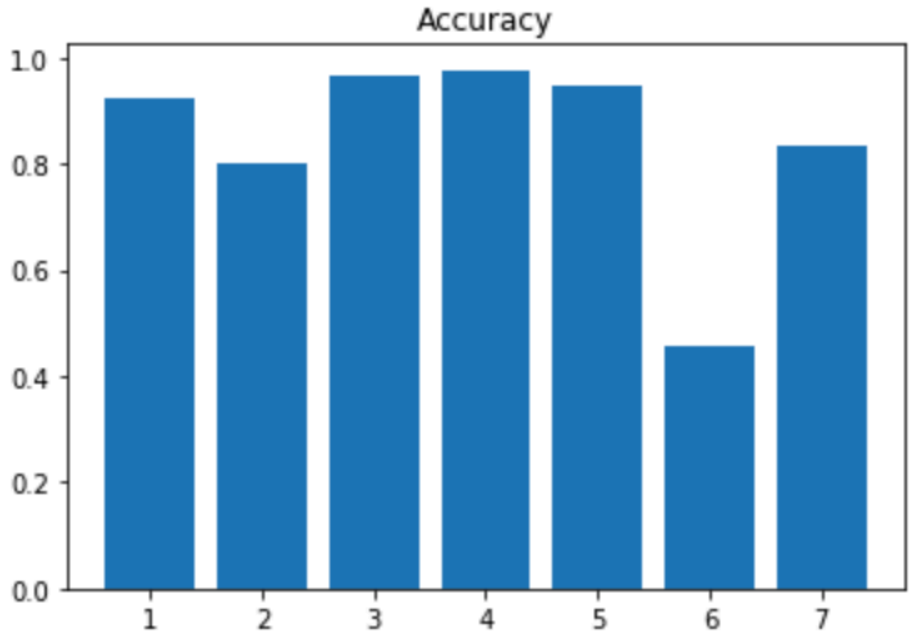
- KNN is expected to take a long Prediction times as it regarded as LAZY algorithm, it is fastest in terms of training but is quite slow when it comes to Predicting.

Conclusion

Accuracy

SVM with rbf kernel, KNNeighbors and SVM with polynomial Kernels had the Highest Accuracy.

For the tasks that require High Accuracy irrespective of time taken to train and prediction time, these 3 algorithms are approximately on par with each other with accuracy 96.5% , 94.8% , 97.6% .

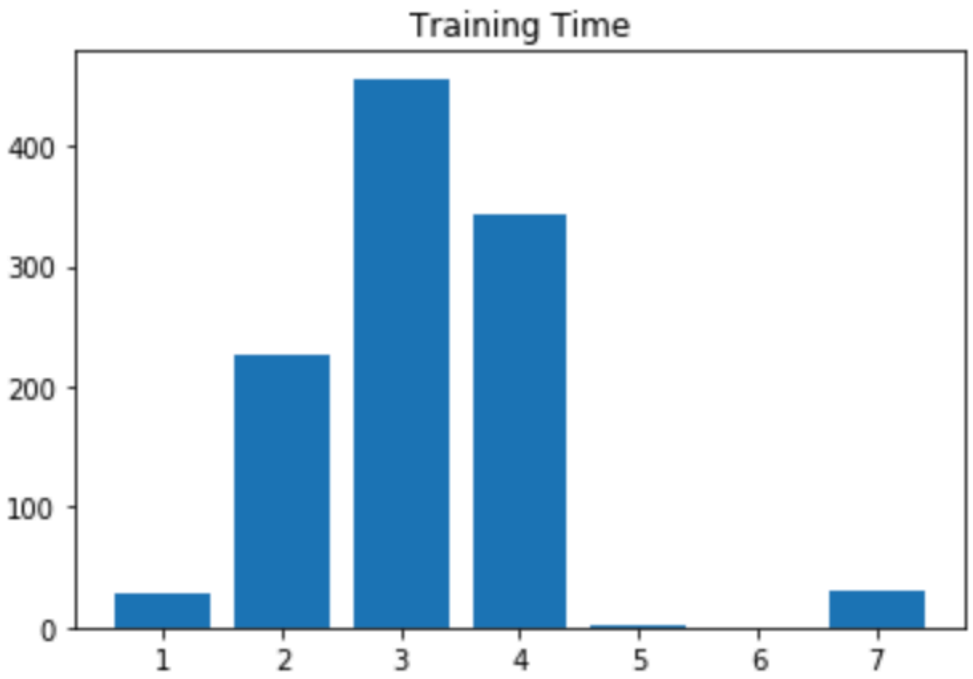


Logistic Regression	0.921
SVM(Sigmoid)	0.798
SVM(rbf)	0.965
SVM(Polynomial)	0.976
KNN	0.948
Naive Bayes	0.455
Decision Tree	0.83

Training Time

In the section of training time KNN , Decision Tree and Logistic regression were the 3 with least training time in increasing order.

These algorithms are highly efficient for the learning purpose where learning is to be done in the minimum amount of the time.



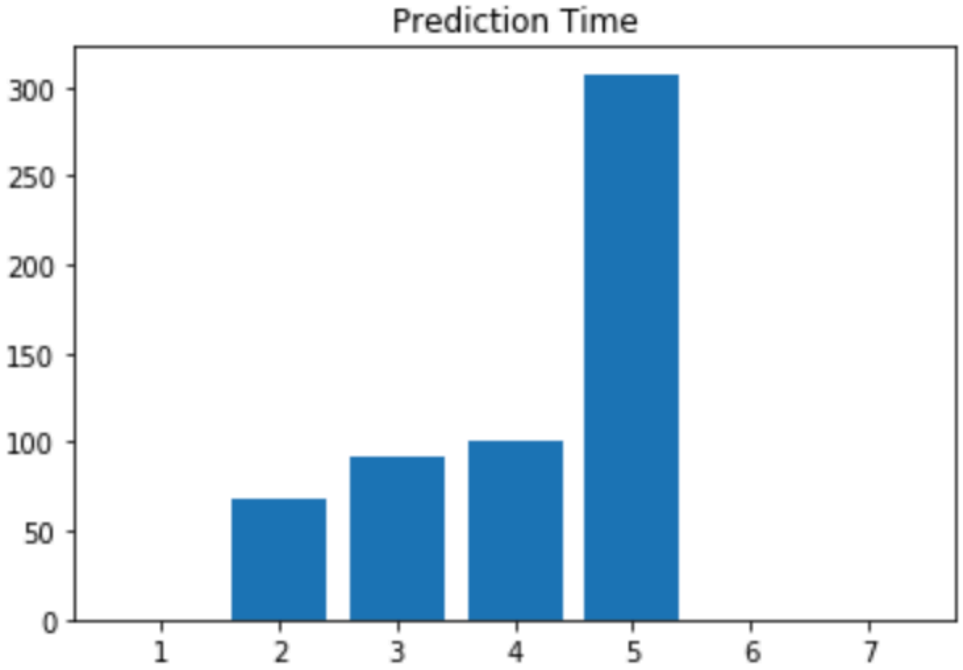
Logistic Regression	27.411
SVM(Sigmoid)	225.38
SVM(rbf)	455.966
SVM(Polynomial)	342.624
KNN	1.284
Naive Bayes	0.530
Decision Tree	29.820

Prediction Time

Logistic Regression and Decision Tree are the two algorithms which were the fastest predictors of the Test set containing 10000 images.

they took time 0.024 seconds, 0.015 seconds respectively., which is indeed very efficient.

these algorithm are best suited for the test where the predictions are to be made on the run or the prediction time must be minimum.



Logistic Regression	0.024
SVM(Sigmoid)	68.005
SVM(rbf)	91.341
SVM(Polynomial)	100.871
KNN	307.307
Naive Bayes	0.226
Decision Tree	0.015

Overall Efficiency

to find the overall efficiency, all the 3 above mentioned parameters must be taken into consideration.

According to which The Decision Tree Algorithm and Logistic Regression looks promising as they are algorithms take take the least amount of time with very good accuracy.

Though Decision Tree tends to overfit the data and gives high variance on the test set, Logistics Regression appears to be the better in this scenario id the accuracy is taken in amount.

These algorithms are very well suited for Reinforced or Online Learning as the decision to be taken on the t+1 instance depends upon the learning upto t instance, therefore requiring low train and prediction time for efficiency.

References

- Maree Raphael, Geurts Pierre, Visimberga Giorgio, Justus Piater, Louis Wehenkel.
  - "*A Comparison Of Generic Machine Learning Algorithms for Image Classification.*"

  Link - http://www.montefiore.ulg.ac.be/services/stochastic/pubs/2003/MGVPW03/maree-ai2003-comparison.pdf

- Aurelion Geron

  OReilly Hands-On Machine Learning with Scikit-Learn and TensorFlow

  http://shop.oreilly.com/product/0636920052289.do

- www.wikipedia.com