CSE301 Bio-computation Assignment 1

Comparative Study of MLP Classifiers on UCI Datasets

Name: Shen Zhun
Supervisor: Bailing Zhang
Due Date: 2011/11/14

Abstract

In this study, there are two parts of work to do. First, comparison of different kinds of BP algorithms, after training and testing, CPU time of running the programs is needed. The other part is to implement Auto encoder classifier and minimum distance classifier, and then compare these two classifiers with KNN and RBF. Also Bagging or boosting classifier ensemble should be implementing and compare with other four classifiers. Through obtained result, Bagging MLP classifier is prove to have Accuracy Rate, however CPU time of Bagging is huge.
Key Word: BP, Batch, CGD, Stochastic, Quickprop, Auto Encoder, Minimum Distance, KNN, RBF, ensemble, Bagging, Boosting

Introduction

Background

Recently, the ANN has received unprecedented attention in many research disciplines. The ANN's parallel, distributive computational structure is reminiscent of the human neural system. In an ANN structure, many simple, nonlinear processing elements, called neurons, are interconnected via weighted synapses to form a network. If neurons are grouped inlayer with weighted synapses interconnecting only neurons in successive layer, the ANN structure is called a multiplayer perceptron model. An MLP model is the most parallel, distributed, nonlinear computing network that mimics the information processing structure of a biologic neural system. From among the numerous paradigms of the ANN, we focus on a model called the feed forward multiplayer perceptron, which has found numerous applications in pattern classification, time series modeling, nonlinear control, and other areas .MLP's features can be conclude into four parts:

1. Universal Approximates-general-purpose models, with a huge number of applications;
2. Nonlinearity - capable of modeling complex functions;
3. Robustness - good at ignoring irrelevant inputs and noise;
4. Adaptability - can adapt its weights and/or topology in response to environment changes;

Development of MLP

Due to its general-purpose feature, computer Scientists use it to study properties of non-symbolic information processing; Statisticians use MLP to perform flexible data analysis; Engineers exploit MLP capabilities in several areas; Cognitive Scientists use MLP to describe models of thinking; Biologists use MLP to interpret DNA sequences and so on.

Applications of MLP

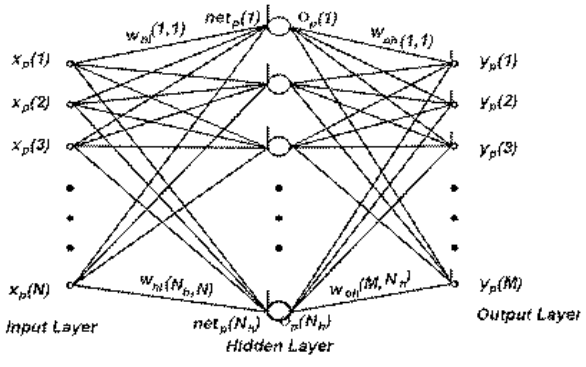
Many applications were developed by MLP, Such as, in combination with standard short-term spectral-based features, there two kind of common models in automatic speech recognition systems. One is Multi-Layer Perceptron (MLP) derived acoustic features, MLP's features are typical used, and have been found to yield consistent performance improvements. The other one is Hybrid of MLP and HMM (Hidden Markov Modeling) techniques, Speech data for each speaker was automatically segmented using a supervised HMM-Viterbi decoding scheme and an MLP was trained with this segmented data. The output scores of the MLP, after appropriate scaling was used as observation probabilities in a Viterbi realignment and scoring step.

In image recognition, MLP is the most popular model, such as Bayesian multi-layer perceptron (MLP) neural networks for image analysis. The Bayesian approach provides consistent way to do inference by combining the evidence from the data to prior knowledge from the problem. Also the MLP network was used to design and develop a job shop scheduling system scheduling software) that can generate effective job shop schedules. The application Uses the back-propagation training process to control local minimal solutions; and develops a heuristics to improve and revise the initial production schedule. It was tested in a real production environment and illustrated in the paper with a sample case.
MLP and RBF, ANFIS (adaptive neuron-fuzzy inference system) can be used for Prediction of swell potential of clayey soils, and compared with the traditional statistical model of multiple regressions. Some Researchers use MLP networks for time series forecasting, and also MLP network was used for Machine Translation Software Cyber Security in Internet.

This study uses MLP and its learning algorithms and Bagging ensemble algorithm to train some UCI Datasets and test the CPU time and accuracy of the algorithms (zoo.mat, vehicle.mat etc).In order to guarantee the validate ,Cross-validate has be carried out during the training and testing by 80% training and 20% testing. In this study Matlab 7.13 software was used in neural network analyses. In the analyses, network parameters of learning rate and momentum were set to 0.015 and 0.5, respectively. Variable learning rate with momentum as networks training function was used. Through our research, we found that CPU time of Algorithm and accuracy is not only depending on algorithm, but also associate with instance and classes' number of data sets. In part1 of the research, CGD is proving the slowest one to training the five data. Among MLP learning algorithms and ensemble Bagging algorithm, Bagging MLP performs best, but cost is also enormous.

MLP model and Ensemble MLP

The MLP network generally consists of an input layer, one or more hidden layers and an output layer. The processing units are arranged in layers. Generally an iterative nonlinear optimization approach using a gradient descent search method is applied to MLP. It provides a feed forward neural network, giving the capacity to capture and represent relationships between patterns in a given data sample. The hidden layer neurons have sigmoid activation functions and output neurons have linear activation function.



Back propagation was the first successful training algorithm. The most common learning algorithm for MLP is error back propagation, in which synaptic strengths are systematically modified so that the response of the network increasingly approximates the desired response can be interpreted as an optimization problem. The generic criterion function optimization algorithm is simply negative gradient descent with a fixed step size. This can be very slow if leaning rate is small. Practically, a must be chosen as large as possible without leading to oscillation. Every supervised training algorithm, including the error back propagation training algorithm, involves the reduction of an error value. For the purpose of weight

adjustment in a single training step, the error to be reduced is usually that computed only for a pattern currently applied at the input of the network. For the purpose of assessing the quality and success of the training, however the joint error must be computed for the entire batch of training patterns. Its learning algorithms are batch, Stochastic, conjugate gradient descent, quick prop and stochastic learning algorithm with momentum.

Batch Backpropagation Algorithm: all patterns are presented to the network before learning takes place. all the training patterns are presented first and their corresponding weight updates summed; only then are the actual weights in the network updated. This process is iterated until some stopping criterion is met.

Stochastic Backpropagation Algorithm: In stochastic training patterns are chosen randomly from the training set, and the network weights are updated for each pattern presentation. This method is called stochastic because the training data can be considered a random variable. In stochastic training, a weight update may reduce the error on the single pattern being presented; yet increase the error on the full training set.

Conjugate gradient Descent: a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions.

Quickprop: It is loosely based on Newton's method and quicker than standard backpropagation, because it uses an approximation to the error curve, and second order derivative information which allow a quicker evaluation.

Stochastic learning algorithm with momentum: The approach is to alter the learning rule in stochastic backpropagation to include some fraction a of the previous weight update.

MLP Ensemble Model-Bagging MLP

Bagging (Bootstrap aggregating) is a “bootstrap” ensemble method that creates individuals for its ensemble by training each classifier on a random redistribution of the training set. Each individual classifier in the ensemble is generated with a different random sampling of the training set. Ensemble is composed of a number of independent MLPs, the input is then propagated to each of the MLP’s input layers, to be processed by each MLP independently, and the connection weights of the neurons in each MLP are initialized randomly. The objective of the learning process is to train each MLP to obtain a unique type of expertise; it is preferable that all MLPs should also have different structures. The structures are diversified by setting different number of neurons in the middle layer of each MLP. In this way, these combined classifiers can produce test-set error lower than that of single classifier. Its algorithm is as follow:

Input:

L: a learning algorithm

N: an integer

For $i = 1$ to N

$\bar{T}$ = bootstrap sample from training set T

$h_i = L(\bar{T})$

End For

Output:

$$h_f(\mathbf{x}) = \underset{y \in \mathbf{Y}}{\operatorname{argmax}} \sum_{i}^{N} h_i(\mathbf{x}) = y$$

Experiment and Results

In order to evaluate each algorithm, a series of experiments were carried out in order to perform a comparative study. The objective of each experiment was to obtain performance criteria for each algorithm under varying circumstances.

Algorithm

To evaluate Backpropagation learning algorithms, Batch, Stochastic, Stochastic Learning algorithm with momentum(**SM**), Quickprop and Conjugate gradient Descent (**CGD**) were used to evaluate the CPU Time of these algorithm with different datasets.

For Ensemble MLP Classifier, I have also implement Auto encoder classifier (**AE**), Minimum Distance classifier (**MD**), KNN and RBF Classifier to compare with Bagging MLP Classifier by their CPU time and Accuracy.

Datasets

Five classification data sets were selected from UCI Machine Learning Repository, and its main features are listed in **Table 1**, namely the *Associate Tasks*, *Number of Instances*, *number of Attributes*.

Table1.A summary of the data sets use.

Dataset	Associate Tasks	Number of Instances	Number of Attributes	Number of Class	Missing
zoo	Classification	101	16	7	No
yeast	Classification	1484	8	10	No
vehicle	Classification	846	18	4	No
satimage	Classification	6435	36	7	No
Glass	Classification	214	9	6	No

Source: UCI Machine Learning Repository.

Evaluating Performance

For Part 1, Batch, Stochastic, SM, CGD and Quickprop algorithms where tested against the above datasets in order to obtain classification performance results, in particular, the classification CPU time on each dataset.

For part 2, As a benchmark, AE,MD，KNN，RBF has been used in order to compare the results from Bagging MLP Classifier.

Various evaluation procedures where considered prior to testing, including full dataset validation, leave-one-out validation, cross-validation and k-folding. Since I am evaluating five algorithms, I prefer to use cross validation and obtaina result which reflects generalization accuracy over training examples that have not been exposed to the algorithms during the training procedure.

The evaluation procedure used to measure generalization performance for both algorithms was the cross validation method, where each training set is divided into two data sets(80% and 20%), the 20% is used for evaluation, and the remaining are all used for training. This way, it was possible to obtain a result that reflected the performance of each algorithm over every separate test set by obtaining the mean error and classification rates.

The order of instances in each training set was randomized prior to training, and normalized (where appropriate). Each algorithm was evaluated over identical training and testing sets folded 100 times. In order to retain a fair estimate over each dataset, both algorithms where initialized with the same learning parameters thought the experiments for part 1:

Number of hidden Units=100
Convergence criterion=0.01
Convergence rate=0.015

Experimental Results

Upon completion of the evaluation procedures, the results were gathered and are shown here. Interpretation of the following results will be discussed in section (Evaluation of Results) of this study.

Table 2. CPU Time for Five Algorithms in Part 1

Data / Algorithm	zoo	yeast	vehicle	satimage	Glass
Batch	160.048	8.2759	1.9344	298.5391	88.4682
CGD	81.3214	327.3525	2169.654	7892.56	186.094
Quickprop	2.8381	8.2805	2.262	393.7138	2.7066
Stochastic	15.418	213.7526	9.7111	3198.1375	17.0665
SM	33.5221	142.8813	5.2416	3115.9718	50.3883

Table 3.CPU Time of Five Models in Part2

Datasets / Models	zoo	yeast	vehicle	satimage	Glass
Auto Encoder	0.0808	1.6411	0.5226	6.0204	0.1891
Min Distance	0.0041	0.1014	0.0148	1.5491	0.0078
KNN	0.0037	0.0714	0.0146	0.5522	0.0072
RBF	0.0094	0.1404	0.0858	1.0826	0.0376
Bagging MLP	9.5707	125.41	58.953	1924.67	25.842

Table 3.Accuracy of Five Models in Part2

Dataset / Models	zoo	yeast	vehicle	satimage	Glass
Auto Encoder	80.50%	48.14%	42.89%	75.13%	50.00%
Min Distance	25.47%	20.44%	40.05%	18.55%	23.10%
KNN	62.66%	40.93%	50.11%	70.66%	47.87%
RBF	53.59%	32.61%	39.64%	42.33%	12.41%
Bagging MLP	73.03%	55.74%	72.19%	89.39%	66.67%

Evaluation of Results

It is worth pointing out that the initial objective of this study was not to discover which algorithm is superior in classification tasks, but to examine the advantages and downfalls of each algorithm under varying conditions. It is clear from brief inspection of the above results that each algorithm has varying performance over different datasets. It is also worth mentioning that dataset pre-processing prior to training (where conversion is needed, i.e. from nominal form to numeric) could have an impact on the accuracy of the algorithm.

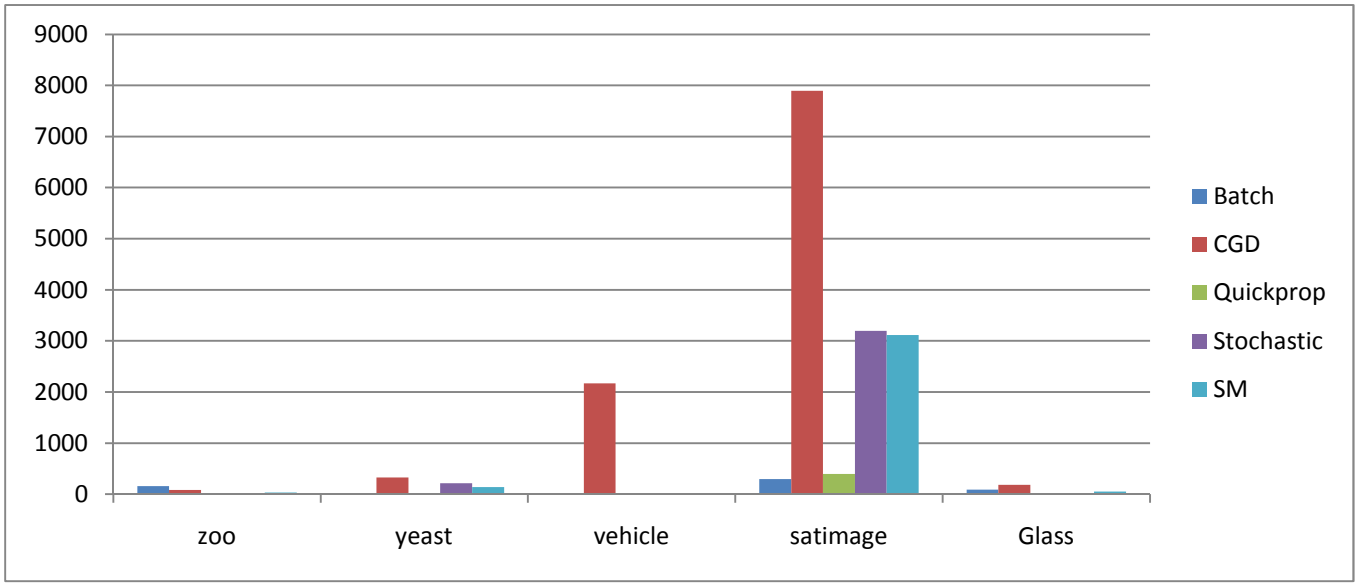


Figure 1 Histogram for CPU Time of Part 1

Obviously, we can find that Datasets is a factor to the CPU time of train and testing.zoo is the fast sample for the five algorithms to test, because it is the simple matrix of the five datasets. satimage is the slowest one due to it has the largest matrix of all. We can find that CGD is the slowest one to train and test the datasets from UCI. When CGD trains satimage, the CPU time is nearly 8000s.Also we can see that SM is faster than Stochastic, it prove that momentum makes a difference.

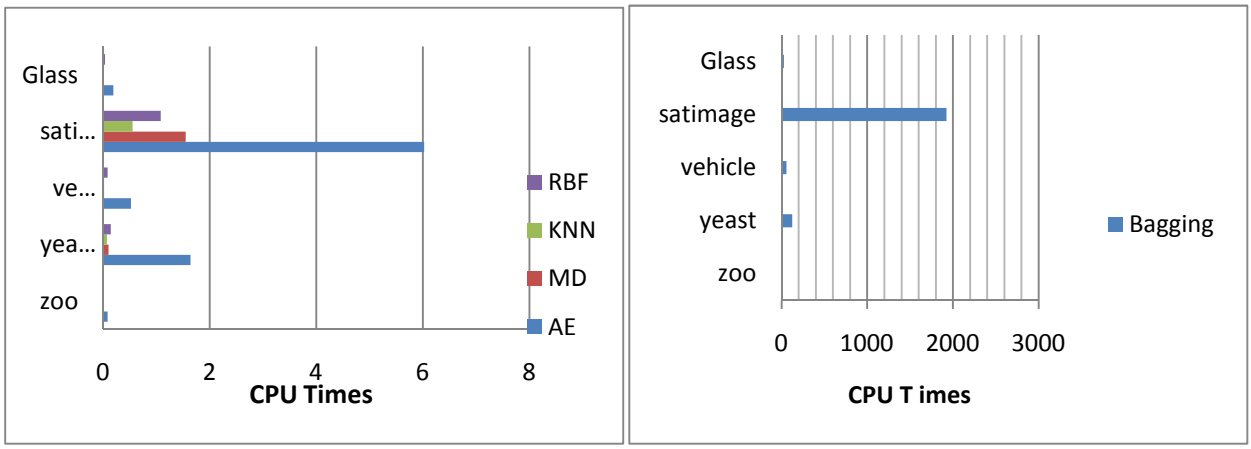
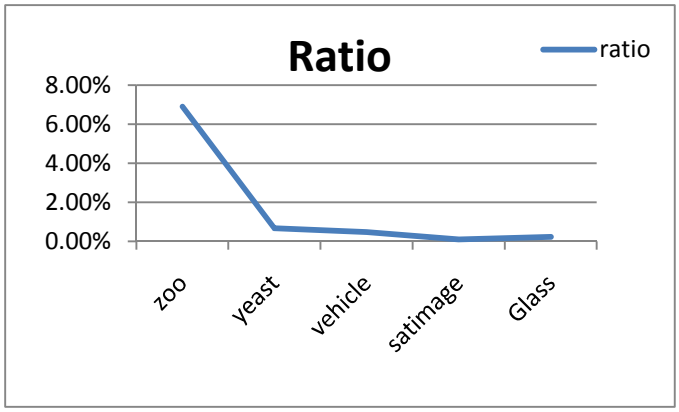


Figure 2 This figure show that obtained data from the training and testing process for part 2. Right and left figure are the CPU Time. Right figure is for *Bagging MLP Classifier*, and Left is for *Auto encoder, KNN, Minimum Distance* and *RBF*. The reason why Bagging is not in Left on is that running time of bagging is better than other. (***AE***=Auto Encoder, MD=Minimum Distance)

The test CPU time of *Auto encoder*, *Minimum Distance*, *RBF*, *kNN* and *Bagging MLP* are summarized in Table 3 and Figure 1 and 2. In order to compare with the target samples from UCI datasets, we run each method for 100 time and get the average CPU time and accuracy. From Table 3 and Figure 1 and 2, we can see that *Bagging MLP* cost the most of time to get result: 9.57s for *zoo*, 125.41s for *yeast*, 58.953s for *vehicle* and 1924.67s for *satimage* and 25.842s for *Glass*, and second cost time model is *Auto Encoder*, and then is *Minimum Distance*, *KNN* and *RBF*. From Figure 1 and 2, we can conclude that CPU running time is not constant; it depends on the model and methods.

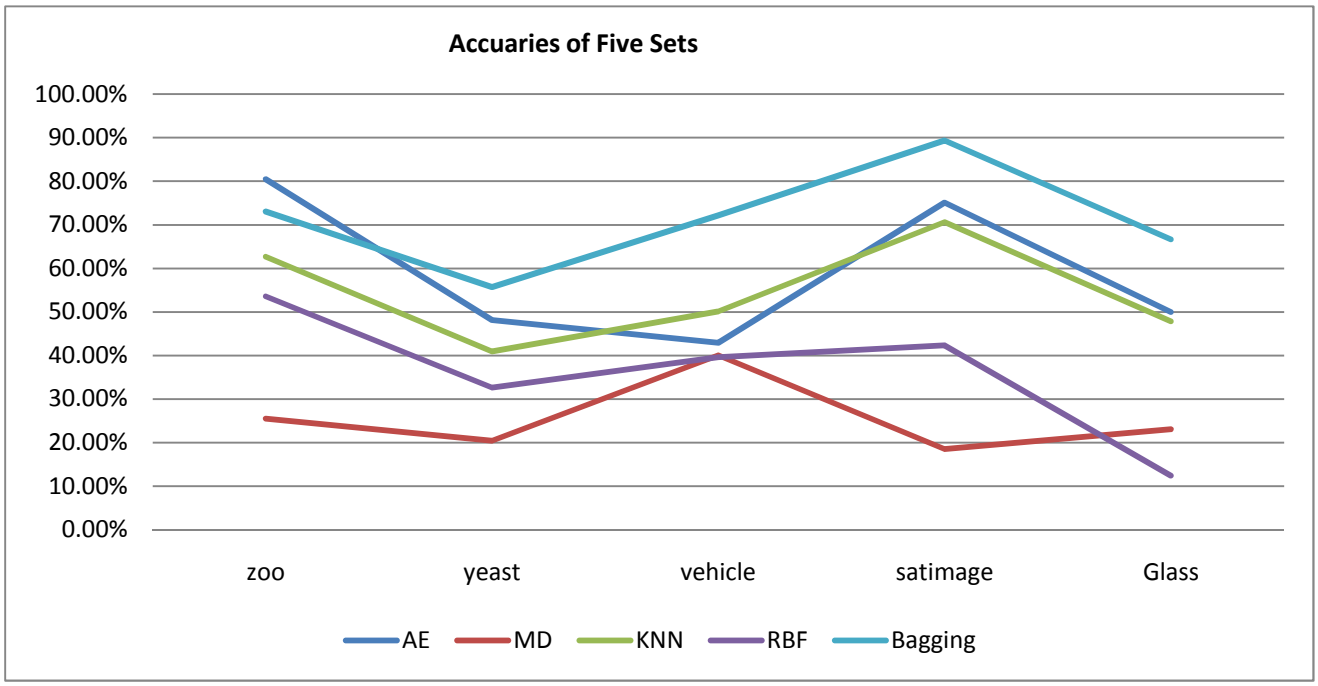
Figure 3 Class Ratio of the five datasets



From the data of table 3, we can get CPU time is decided by the ratio of class number and instances. Ratio of *zoo* is 6.9%, ratio of *yeast* is 0.67%, ratio of *vehicle* is 0.47%, ratio of *satimage* is 0.47% and ratio of *Glass* is 0.23%.It shows in figure 3.

We can see that the ratio of *zoo* is the biggest, and compare with figure 1, the CPU Time of *zoo* is the least. Ratio of *satimage* is the least and its CPU time of Figure 1 is the longest. So CPU Time is also affect by the samples.

Figure 4



We can see from the **Figure 4**, there are three algorithms always above 40%, they are *Bagging MLP*, *Auto Encoder* and *KNN*.*RBF* is the sensitive algorithm and changed sharp in *Glass* Data set. Compare the shape of the wave patterns, we can found that *KNN* and *RBF* are very similar and they should use the similar method to implement the classifier. The reason is that when they training the samples, their way to measure the distance metric both are Euclidean Distance.

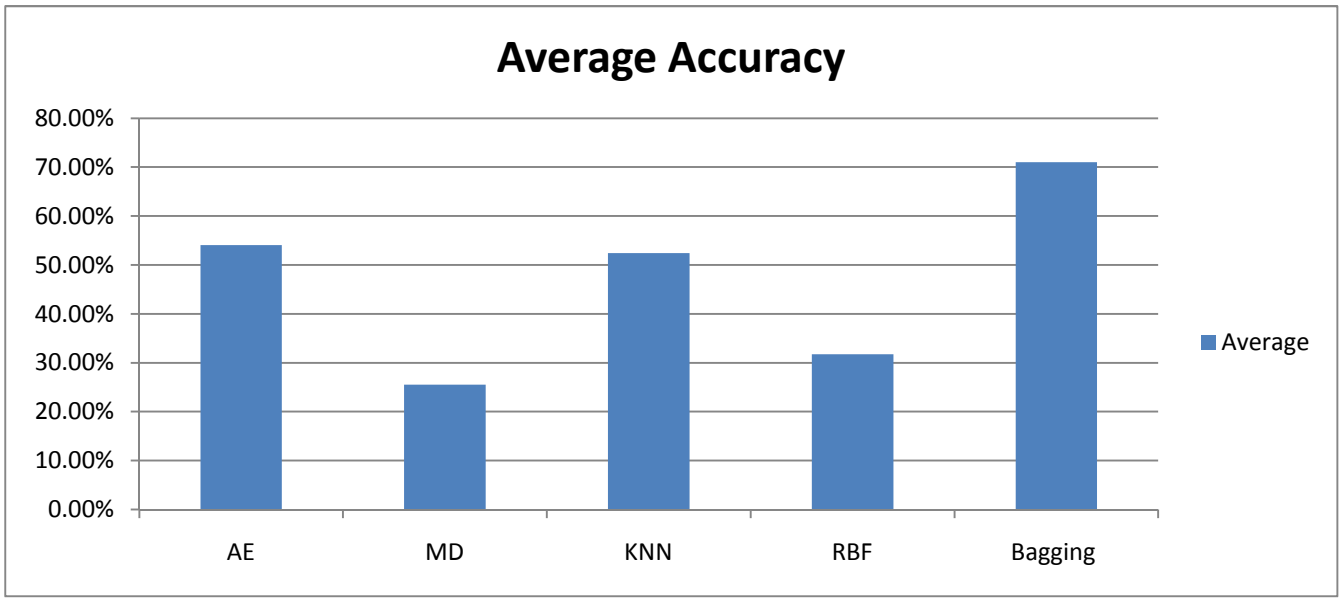


Figure 5 Average CPU Time of Part 2

From Figure 5, we can find that *Bagging MLP Classifier* is most effective way to implement the test for the UCI date sets. However, it also have some changes between different sets, the guess of this phenomena is that numbers of instances for the five data sets affect the accuracy. The reason why Bagging is the most accuracy algorithm is that bootstrap ensemble method generate training set by random drawing ,then train each Bagging's component classifier individual then combined them together .The diversity among these classifier generally components for the increase in error rate of any individual classifier.

Minimum Distance Classifier is the least effective one of these five algorithms. It classifies unknown data to class which minimize the distance between the data and the class in multi-feature space. It does not have the process of training, so the accuracy is low.

In this research, *Auto encoder* also has a perfect performance due to its adjusting the learning rate, Training with too small a learning rate will make agonizingly slow progress. Too large a learning rate will processed much faster, but may simply produce oscillations between relatively poor solutions. Classification based on Auto encoder also use reconstruction errors as indicator for predicting class label, which should be associated with the small error.

Discussion and Conclusion

One aspect I learn from this study was to use back propagation algorithm and understand its structure. During compare Batch, CGD, Stochastic, QuickPort and SM, I understood that the reason why back propagation was the popular and success algorithm. Second aspect is that I learned to compare models based MLP and Ensemble learnig.MLP plays important role in research and industry fields. Speech Recognition, Face Recognition, Handwriting Recognition, Data Mining, Medical diagnosis,Computre Game and so on. It is more and more popular and affects our life and thoughts. Also Classification and Ensemble Learning are useful. Bagging and Boosting are two popular methods for producing ensembles. It prove that bagging is almost always more accurate that a single classifier. One example is that when it trains *satimage*, its accuracy is 89.39%.Another aspect is that I learn from this assignment is that choosing suitable parameter is so important and hard, sometimes we chose a unbefitting value, for example learning rate equals 0.5, the program will run very fast and get bad result, if we choose learning rate 0.001, the program will run for nearly one day time. The training and testing process for ANN is empirical and it is hard to prove theoretical.

Through the result of this study, we can conclude that CPU Times was affect by the samples, especially percentage of class in the instances of the sample. Bagging MLP Classifier is most effective way to implement the test for the UCI date sets. Auto encoder with dynamic learning rate also performs a good work.RBF and KNN perform the similar due to the use similar function to calculate the distance.

There are two strange points in this study, one is that RBF should fast than KNN in theoretical, but the result shows that RBF is a little bit slower than KNN.I try to change the number of hidden units, it make a difference. However, a suitable number is still not found. The other one is that when RBF classify the Glass dataset, the accuracy sharply falls from 42.33% to 12% around. I analysis the train and test process, and the sample features. Finally I found that scatter of Glass is a little bit balanced, and we do not consider the noisy of the datasets, I guess the noisy of Glass causes the fall. In the further research, more datasets should select and according to their features, divided into different groups (Miss Value, with noisy, normal).Then compare the results of them, in this way, we will get better conclusion.

Reference

1. David Opitz, Richard Maclin(1999),” Popular Ensemble Methods: An Empirical Study”, University of Minnesota Duluth, USA.
2. Zhuo Zheng(2006),” Boosting and Bagging of Neural Networks with Applications to Financial Time Series”,University of Pennsylvaniz,USA.
3. Aki Vehtari, Jouko Lampinen(2003),”Bayesian MLP neural networks for image analysis”, Laboratory of Computational Engineering, Helsinki University of Technology, Finland.
4. M. Kashaninejad, A.A. Dehghani, M. Kashiri(2008),” Modeling of wheat soaking using two artificial neural networks (MLP and RBF) ”, Department of Food Science and Technology, Gorgan University of Agricultural Sciences and Natural Resources, Beheshti Ave., Iran
5. Isik Yilmaz, Oguz Kaynar(2010),” Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils”, Cumhuriyet University, Turkey.
6. J. Park, , F. Diehl(2010),” The efficient incorporation of MLP features into automatic speech recognition systems”, Department of Engineering, University of Cambridge,UK.
7. Shan Feng,Ling Li,Ling Cen(2002),” Using MLP networks to design a production scheduling system”, Huazhong University of Science & Technology, Wuhan 430074, China.
8. Włodzisław Duch(2005),” Search-based Algorithms for Multilayer Perceptrons”,p56-61, The Silesian University of Technology , Gliwice.
9. Jayant M. Naik and David M. Lubensky(2002),” A Hybrid HMM-MLP Speaker Verification Algorithm fof Telephone Speech”, Advanced Speech Services NYNEX Science and Technology Inc.New York.
10. A.R. Kazemi, F. Sobhanmanesh(2011),” MLP refined posterior features for noise robust phoneme recognition”, Shiraz University, Shiraz, Iran.
11. Miguel Rocha(2005),” Evolutionary Design of Neural Networks for Classification and Regression”, University of Minho, Portugal.
12. Michael Green a,*, Jonas Bjo¨rk b, Jakob Forberg(2006),” Comparison between neural networks and multiple logistic regression to predict acute coronary syndrome in the emergency room”, Lund University, Sweden.
13. Richard Forsyth(1996 UCI),”Zoo Data Set”, http://archive.ics.uci.edu/ml/datasets/Zoo
14. Japan Association of Remote Sensing, Minimum Distance Classifier. http://stlab.iis.u-tokyo.ac.jp/~wataru/lecture/rsgis/rsnote/cp11/cp11-6.htm
15. Christopher M Bishop, Ian T Nabney(1996,1997), NETLAB Online Reference Documentation.

http://brain.fuw.edu.pl/~jarek/SIECI/netlab_help/
16. Louis Mendelsohn(1993), Training Neural Networks. http://www.stator-afm.com/training-neural-networks.html.
17. Wikipedia(2011), Radial basis functionhttp://en.wikipedia.org/wiki/Radial_basis_function
18. Wikipedia(2011),Bootstrap aggregating, http://en.wikipedia.org/wiki/Bootstrap_aggregating
19. Wikipedia(2011),k-nearest neighbor algorithm, http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm
20. Wikipedia(2011), Autoencoder. http://en.wikipedia.org/wiki/Autoencoder
21. Iffat A. Gheyas, Leslie S. Smith(2009),” A Neural Network Approach to Time Series Forecasting”, London, U.K.
22. UCI Machine Learning Repository.http://www.ics.uci.edu/~mlearn/MLRepository.html