Empirical Determination of Sample Sizes for Multi-layer Perceptrons by Simple RBF Networks

HYONTAI SUG
Division of Computer and Information Engineering
Dongseo University
Busan, 617-716
REPUBLIC OF KOREA
hyontai@yahoo.com http://kowon.dongseo.ac.kr/~sht

Abstract: - It's well known that the computing time to train multilayer perceptrons is very long because of weight space of the neural networks and small amount of adjustment of the wiights for convergence. The matter becomes worse when the size of training data set is large, which is common in data mining tasks. Moreover, depending on samples, the performance of neural networks change. So, in order to determine appropriate sample sizes for multilayer perceptrons this paper suggests an effective approach with the help of simple radial basis function networks that work as a guide. Experiments with the two different data sets that may represent business and scientific domain well showed the effectiveness of the suggested method.

Key-Words: - multilayer perceptron, sample size, radial basis function network, data mining

1 Introduction

As data are gathered in wide application areas, data mining draw many researchers' attention. So neural networks that is one of the successful data mining methods have been applied to the wide areas and reported successful [1]. But even though neural networks are one of the most successful data mining or machine learning methodologies, they have some weak points with respect to performance due to the fact that they are built based on greedy algorithms and the knowledge of experts as well as data set itself used to train the neural networks. So, there are some aspects of improvements; the improvement of the structure of the neural networks, and the improvement of connection weights, and the training data set.

Multilayer perceptrons (MLPs) and radial basis function (RBF) networks are two major neural networks that have been applied successfully for classification tasks in data mining. At a glance the structure of the two neural networks are similar, but their training mechanisms are different. While both networks have three layers including the input layer, hidden layer, and output layer, RBF networks differ from MLPs, because in RBF networks the hidden layer performs some unsupervised learning [2, 3, 4], but the others don't.

Even though many methods have been suggested to find optimal network structures, basically the structure of the networks is usually determined by the knowledge of human experts with some experiments to refine the neural networks. As a result, built neural networks may not represent the best knowledge models that are best for some collection of training examples in the target data set.

For the improvement of connection weights MLPs use backpropagation algorithms. The backpropagation algorithms rely on some greedy search algorithms like gradient decent search algorithm [5]. In order to avoid local optima the weights are adjusted slowly so that the computing time can be very large. The matter becomes worse, if the traing data set is large.

So, because most target databases for data mining are very large, we need sampling process to the target databases. But the found knowledge models based on random samples are prone to sampling errors. An alternative strategy may be to use the original database. But, it may not be a good idea since it might be computationally very expensive, and because the target databases reflect only a portion of the target domain, overfitting problem may happen.

Because RBF networks can be trained relatively in short time unless the network structure is complex, we may hope that if a sample size that has a good result in RBF networks, the same thing may happen in MLPs under the condition that we have appropriate neural network structures of RBF networks and MLPs. So, based on this idea we want to investigate the relationship empirically by experimenting the idea for some representative real world data sets.

In section 2, we provide the related work to our research, and in sections 3 we present our method as well as background technologies. Experiments were run to see the effect of the method in section 4. Finally section 5 provides some conclusions.

2 Related work

Neural networks are widely used for machine learning or data mining tasks since the first neural network algorithm, the perceptron [6]. Because of the limited predictability of the perceptron, multilayer perceptrons have been invented [7, 8, 9, 10]. Multilayer perceptrons have been applied widely including mathematical problems [11, 12, 13] as well as application fields [14, 15]. There are two kinds of networks based on how the networks are interconnected – feed-forward neural networks and recurrent neural networks [16]. MLPs are feed-forward neural networks. The weak point of MLPs is computational intensiveness. So most data mining applications that use MLPs prefer samll-sized samples or data sets.

RBF networks are one of the most popular feed-forward networks [17, 18, 19] that are used as a replacement for MLPs. A good point of RBF networks is that they can be trained in relatively short time. But, due to the feed-forward nature and the hidden layer functions to approximate the target data set, local optima problem also may occur.

In order to overcome local optima problem many evolutionary search algorithms were suggested [20, 21, 22, 23] for RBF networks. Evolutionary search algorithms try to find global optimal solutions so that it is possible to find better RBF networks. But the algorithms require more extensive computing time as well as more elaborate techniques related to the evolutionray computation like the representation technique of network structures and weights.

Because some induction method is used to train the data mining models like neural networks, the behavior of trained data mining models also dependent on the taining data set. So, there is research on sample size as well as the property of samples and sampling scheme. Fukunaga and Hayes [24] discussed the effect of sample size for parameter estimates in a family of functions for classifiers. Raudys and Jain [25] prefer small sized samples for feature selection and error estimation for several classifiers of pattern recognition. In paper [26] the authors showed that class imbalance in training data has effects in neural network development especially for medical domain. Jensen and Oates [27] investigated three sampling schemes, arithmetic, geometric, and dynamic sampling for decision tree algorithms. In arithmetic sampling and geometric sampling, the sample size grows in arithmetic and geometric manner respectively. Dynamic sampling method determines the sample size based on dynamic programming. They found that the accuracy of predictors increases as the sample size increases and the curve of accuracy is logarithmic, so they used the rate of increase in accuracy as stopping criteria for sampling. They experimented with C4.5 decision tree algorithm which is freely available. In paper [28] several resampling techniques like cross-validation, the leave-one-out, etc. are tested to see the effect of the sampling techniques in the performance of neural networks, and discovered that the resampling techniques has very different accuracy depending on feature space and sample size.

3 The method

We apply three existing techniques in our method; radial basis function networks, multilayer perceptrons, and sampling techniques. Let's see the principles of each technique briefly.

3.1 Multilayer perceptrons

Multilayer perceptrons were introduced in middle of 80's to enhance the limited capability of perceptrons. MLPs have an input layer, an output layer, and one or more hidden layers. MLPs became popular by the efforts of 'parallel distributed group' [29]. An important property of MLPs is backpropagation learning algorithm, and by the learning algorithm a variety problem could be solved including linear separability problem that were impossible to solve with perceptrons. Unlike other statistical method MLPs do not need assumptions about data distribution so that they are good when we don't have much statistical knowledge about data. There are many cases that report successful application of MLPs [30, 31] as well as hardware implementation [32]. A MLP is a combination of perceptrons that have simple structure like Fig. 1. The output y of a perceptron is weighted sum of its inputs.

$$y = f(\sum_{i=0\sim m} w_i x_i - \theta) \text{ where is } \theta \text{ threshold for output.} \quad (1)$$

where f is an activation function. There are three representative activation functions; step function, sign function, and sigmoid function. In general, sigmoid functions are widely used. An example of sigmoid function is $1/(1 + e^{-x})$. Fig. 2 shows the graph of the sigmoid function.

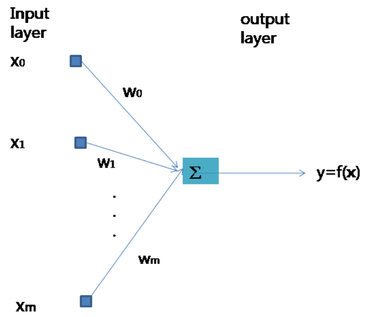

Fig. 1 Schematic view of perceptron

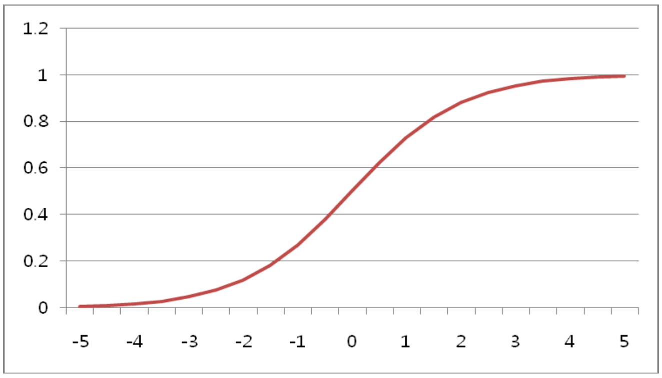

Fig. 2 Sigmoid function

A multilayer perceptron has multiple hidden layers so that it has more power in predictability than perceptrons. Fig. 3 shows a schmetic view of a simple multilayer perceptron where the number of hidden layer is one.

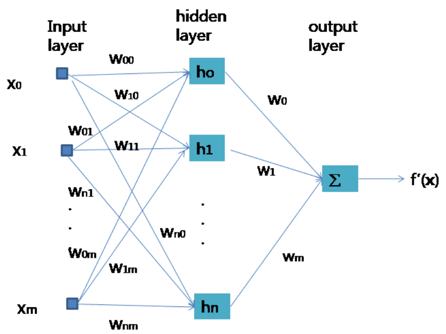

Fig.3 Schematic view of MLP

The task of forecasting with MLP can be stated as a function approximation problem.

Given a set **x** of samples ($\mathbf{x}_i$, y_i) such that f($\mathbf{x}_i$) = y_i for i = 1, ..., n, where n is the sample size and $\mathbf{x}_i$ is the input vector. We want to find an unknown function f' that minimize the error, E(f, f') where f is a prior function that predicts outcome exactly. So, f can be written as follows:

$$f: I \rightarrow O \qquad (2)$$

where I is the domain of input and O is the domain of output. A MLP has similar activation mechanism with that of perceptrons, but the two are different in the sense that MLPs can have multiple hidden layers and weights are adjusted by backpropagation learning algorithms [33].

3.2 Radial basis function networks

Radial basis function networks or RBF networks were also introduced in late 80's. There are many cases that report successful application of RBF networks [34, 35, 36, 37]. The function of RBF networks is based on the function of actual neurons like visual cortices that have the property of being sensitive to some particular visual characteristics [38].

The task of forecasting with RBF network is a classification or regression problem, so the problem can be stated as a function approximation problem like equation (2).

Because in real world situation it is very common to have incomplete traing input data set, error estimation is necessary and usually done by the sum of square of errors E':

$$E' = \sum_{i=1\sim n} (y_i - f'(\mathbf{x}_i))^2 \qquad (3)$$

So, RBF network is a function f'(**x**) having a linear combination of hidden radial function $h_j(\mathbf{x})$. So the RBF network can be written as follows:

$$f'(\mathbf{x}) = \sum_{j=1\sim m} w_j h_j(\mathbf{x}) \qquad (4)$$

where $h_j(\mathbf{x})$ is the radial function in hidden node j and w_j is the weight between function $h_j(\mathbf{x})$ and output node.

While multilayer perceptrons use sigmoid functions for activation functions, RBF networks use radial basis functions at hidden layer. Fig. 4 shows a schematic view of a RBF network.

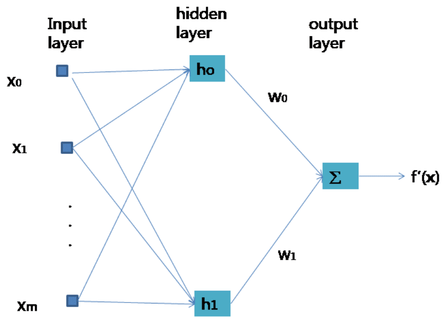

Fig. 4 Schematic view of RBF network

Because radial basis function makes an approximation based on the training data, one should choose a basis function that can represent the target domain well. There can be a variety of radial basis functions, for example, Gaussian, multiquadric, cauchy, etc.

Center point and radius are two parameters for a radial function. The center of the radial function indicates the central position, and the radius determines how the function spreads around its center. If we use Gaussian as a basis function, mean is the center and variance is the radius.

In order to train RBF networks first we should find appropriate centre and radius of radial basis function. For this task, we may use some unsupervised learning algorthms like K-means clustering. After deciding the centers and radiuses the weigts can be trained.

3.3 Sampling method

3.3.1 Arithmetic sampling

In arithmetic sampling sample size is increased arithmetically, so the sequence of sample sizes is in arithmetical progression. We can define the sample size S_i in arithmetic sampling with the following equation:

$$S_i = S_0 + i \times K \tag{5}$$

Here, S_0 is the initial sample size, i is an iteration number, and K is a constant for increment.

So, we can have an arithmetical progression of sample sizes like, S_0, $S_1 = S_0 + 1K$, $S_2 = S_0 + 2K$, $S_3 = S_0 + 3K$, and so on. For example, if $S_0 = 200$ and K = 100, then $S_1 = 300$, $S_2 = 400$, $S_3 = 500$, and so on.

Therefore, if we use arithmetic sampling with some proper K value, we can trace the accuracy of neural networks throughly. On the other hand this property may become a drawback of the arithmetic sampling scheme, because we may need a lot of repeated sampling, if K is small. For example, let's assume we have 1,000,000 records in a data set, and we start from 100,000 records as an initial sample size and the constant K value is 1,000. We have to do sampling 500 times to reach to the half of the target data set. Because most target data sets for data mining contain lots of data, it is highly possible that arithmetic sampling alone cannot be used efficiently.

3.3.2 Geometric sampling

In geometric sampling method sample size is increased geometrically so that the sequence of sample sizes are in geometrical progression. We can define sample size S_i for sample i in geometric sampling with the following equation:

$$Si = S_0 \times K^i \tag{6}$$

Here, S_0 is the initial sample size and K is a constant for increment.

So, we can have a geometrical progression of samples in size, S_0, $S_1 = S_0 \cdot K$, $S_2 = S_0 \cdot K^2$, $S_3 = S_0 \cdot K^3$, and so on. For example, if $S_0 = 2{,}000$ and K = 2, then $S_1 = 4{,}000$, $S_2 = 8{,}000$, $S_3 = 16{,}000$, and so on. As we can see from the example, if we use geometric sampling, sooner or later we can see very big sample sizes. So, the target data set may be exhausted within a few rounds.

As an example, let's assume that we have 1,000,000 records in a data set as before, and we start from 2,000 records as an initial sample size and the constant K value is 2. So, the sequence of sample size becomes like 2,000, 4,000, 8,000, 16,000, 32,000, 64,000, 128,000, 256,000, 512,000. It takes only 9 rounds to reach to the half of the target data set.

Another noticeable fact in geometric sampling is that the sample size values are very sparse at the later stage of the sampling. So, geometric sampling cannot be a good sampling strategy, if used data mining algorithms do not have the tendency of monotonic increase in accuracy. Let's assume that we have a learning curve that have some sudden peaks in accuracy as the training size grows. Because geometric sampling method has very sparse sampling interval with respect to sample size at the later stage of the sampling schedule, we might miss the points. Please look at Fig. 5 that depicts learning curve for some induction algorithm that have ocillating accuracy as the sample size gorws. Because there some sudden peaks in accuracy, sparseness in sample sizes like 1,000, 2,000, 4,000, 8,000, 16,000 may not detect the good points like sample sizes, 12,000, 14,000. In the figure X axis represents sample size and Y axis represents prediction accuracy.

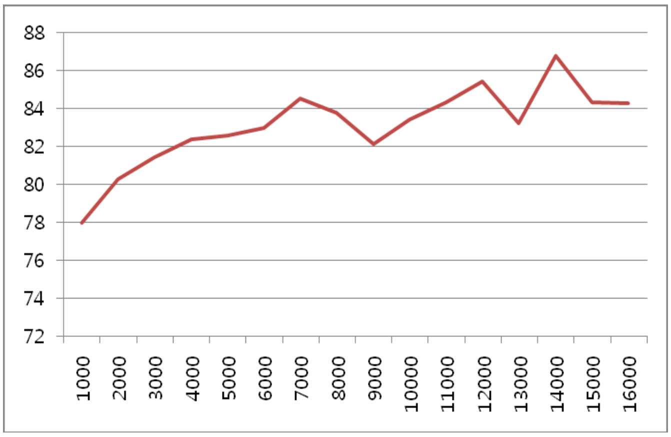

Fig. 5 Learning curve in accuracy for some possible data mining algorithm

3.4 The method

It is not easy to determine an appropriate sample size that is the best for MLPs with target data set. So, in order to overcome this problem we resort to repeated sampling scheme for RBF networks that considers various sizes of samples.

We do the sampling until the sample size is less than or almost the half of the target data set, because we assume that we have some large target data set that is common in data mining domain and we want to have enough test data also. Because we use RBF networks in our method, as a first step we should determine what radial basis function we will use. For this task we should be careful about selecting the radial function that can accomodate the target data well, because it may affect final result much. The following is a brief description of the procedure of the sampling scheme. It has two steps.

[Step 1]
INPUT: a data set for data mining,
  k: the number of random sampling for each sample size,
  s: initial sample size.
OUTPUT: S, A, V, I, D.
/* S: set of sample size,
  A: set of accuracy,
  V: set of average accuracy,
  I: set of average improvement
  D: set of difference in max and min accuracy */
j := 1;
Do while s < | target data set | / 2
  Do for i = 1 **to** k /* generate k RBF networks for each loop*/
    Do random sampling of size s;
    Train and test a RBF network;
    a_{ij} := Accuracy of the RBF network;
    $A_j := A_j \cup \{a_{ij}\}$;
  End for;
  S := $S \cup s$;
  $A := A \cup A_j$;
  v := the average accuracy in A_j;
  $V := V \cup \{v\}$; /* V: average accuracy values */
  i := (the average accuracy of the RBF networks of previous step) – (the average accuracy of the RBF networks); /* average improvement rate */
  $I := I \cup \{i\}$; /* I: set of i values */
  d := (maximum of accuracy among the trained RBF networks) - (minimum of accuracy among the trained RBF networks);
  /* d stands for the fluctuation of accuracy values in the trained RBF networks */
  $D := D \cup \{d\}$; /* D: set of d values */
  If s >= mid_limit **Then**
    s := s + sample_size_increment; j++;
    Else
    s := s $\times$ 2; j++;continue; /* while loop */
  End if
End while;

[Step 2]
Choose a sample size as a starting sample size from Step 1 for MLPs which satisfies the following conditions:

1. A sample size that belongs to a group of some best accuracies,
2. A sample size that have smaller value in difference of minimum and maximum accuracy,
3. A sample size that is bigger.

Repeat
  Train MLPs with the chosen sample sizes like RBF networks in step 1;.
  Increment sample size like RBF networks in step 1;
Until improvement < predefined_limit;

We double the sample size until the size reaches to some point, mid_limit, then we increment the sample size by some fixed value, because doubling the sample size can exhaust the data very soon.

Even though we do random sampling, because we may have some sampling bias and sampling errors, the trained RBF networks may be in variety in accuracy. So, in order to get rid of the effect of variety in accuracy we sample multiple times whthin a sample size, then we average the accuracy values of the trained neural networks for each sample size, and this average accuracy with improvement value as well as fluctuation value in accuracy is used to determine a proper sample size for this purpose. We should set an appropriate value of k for resampling.

Because the accuracy of MLPs have the tendency of somewhat monotonic increase as the sample size grows, we prefer bigger sample size. By selecting a bigger sample size that generates good RBF networks in average case with satisfactory accuracy, we can have better MLPs for the sample size.

4 Experimentation

Experiments were run with two data sets in UCI machine learning repository [39] called ‘adult’ and ‘forest cover types’ to see the effect of the method. The adult data set [40] is a refined version of ‘census income’ data set. The census income data set is census data of 1994. The census income data set is originated from the census bureau database. The number of instances in the adult data set is 48,842. The total number of attributes in the adult data set is forteen, and among them six attributes are continuous attributes and one attribute is a class attribute where it has two classes, yearly income being greater than or equal to 50,000 and less than 50,000 .

The ‘forst cover types’ data set [41] includes forest information in four wilderness areas found in the Roosevelt National Forest of northern Colorado. The number of instances in the adult data set is 581,012. It has twelve continuous attributes as independent variables, while seven major forest cover types were used as a dependent variable or a class variable. ‘Adult’ and ‘forest cover types’ data sets were selected as representatives of business and scientific domains respectively, because the origin of ‘adult’ data set is census and it contains a lot of nominal values, and ‘forest cover types’ data contains continus values only which is common in scientific domain. In addition both are relatively very large so that they are appropriate for the experiment.

4.1 Experiment with adult data set

We used RBF network using K-means clustering to train from various sample sizes of ‘adult’ data set. The used basis function is Gaussian, because the ‘adult’ data set is originated from census database. The given number of clusters for K-means clustering is two which is based on the number of classes.

We also trained MLPs with the same sample sets for each different sample sizes. In order to train MLPs the given number of hidden layers is also two. Bacause we have many nominal values in the data set, we have many nodes in input layer, so the given number of hidden layers is relatively small. To compansate the small number of hidden layer the training time of 10,000 is given. Table 1 and 2 show the summary of the results. For each sample size seven random samples have been selected and seven neural networks have been generated for the experiment.

The initial sample size for training is 200, and the size of samples is doubled as the while loop runs. The given mid_limit value for sample size is 6,400, and the sample size increment from the mid_limit is 3,200. The rest of the data set after sampling is used for testing.

In the table 1 and 2, the third column, improvement(%), means the percentage of improvement in accuracy compared to the neural networks of previous sample size, and the fourth column represents the difference of maximum and minimum values of accuracy among the neural networks in the given sample size, and the last column is for the average computation time in second. The used computer is a pentium 4 personal computer with 2MB main memory.

Table 1. RBF networks for ‘adult’ data set with various sample sizes

Samp. size	Average Accuracy (%)	Improve -ment(%)	Diff. of max & min accuracy (%)	Average compu. time(sec)
200	82.15153	NA	2.4239	0.04
400	83.3527	1.20117	1.6907	0.07
800	82.86174	-0.49096	0.9783	0.14
1,600	83.13183	0.27009	1.5071	0.76
3,200	83.64977	0.51794	1.1419	1.50
6,400	83.38611	-0.26366	2.0288	1.52
9,600	83.57734	0.19123	0.6345	2.21
12,800	83.45717	-0.12017	0.6165	3.01
16,000	83.42126	-0.03591	0.6970	3.82
192,00	83.52089	0.09963	0.6385	5.53

If we look at table 1, sample size 3,200 has the best accuracy, and the second best is sample size 9,600. Because accuracy of MLP increase as the sample size grow, we may choose sample size 9600 as the training point of the MLP. In other words, because the difference of accuracy between sample size 3,200 and 9,600 is only 0.07243%, and the difference of max and min accuracy between the two is almost half in sample size 9,600, and the sample size is bigger, we choose the sample size of 9,600. Note that as the sample size increases, accuracy does not increase monotonically. Note also that bigger sample sizes have less fluctuation in difference of maxmum and minimum accuracy values.

Table 2. MLP networks for 'adult' data set with various sample sizes

Samp. size	Average accuracy(%	Improve -ment(%)	Diff. of max & min accuracy (%)	Average compu. time(sec)
200	77.96967	NA	2.6294	91.5
400	80.27067	2.301	5.5923	186.7
800	81.41629	1.14562	3.0182	351.9
1,600	82.36150	0.94521	0.362	673.4
3,200	82.58996	0.22846	3.6041	1337.3
6,400	82.99027	0.40031	4.5545	2780.1
9,600	84.51573	1.52546	0.3899	3977
12,800	84.55946	0.04373	0.6921	5340.4
16,000	83.86250	-0.69696	1.1414	6367
19,200	84.41695	0.55445	1.0521	7666.3

In table 2 the results of sample sizes other than 9,600 and 12,800 are also presented for reference. If we look at table 2, sample size 12,800 has the best average accuracy, and the second best is sample size 9,600. Note that even with 1.33 times bigger sample size, the accuracy improvement is only 0.04373% which is 1.000517 times better so that we may stop further iteration.

Note also that the training of MLPs takes thousands of times longer than that of RBF networks so that without the help of RBF networks it will take very long time. Fig. 6 displays the change of prediction accuracies of RBF networks (dotted line) and MLPs (solid line) for the data set more clearly. In the figure X axis represents the sample size and Y axis represents average prediction accuracy.

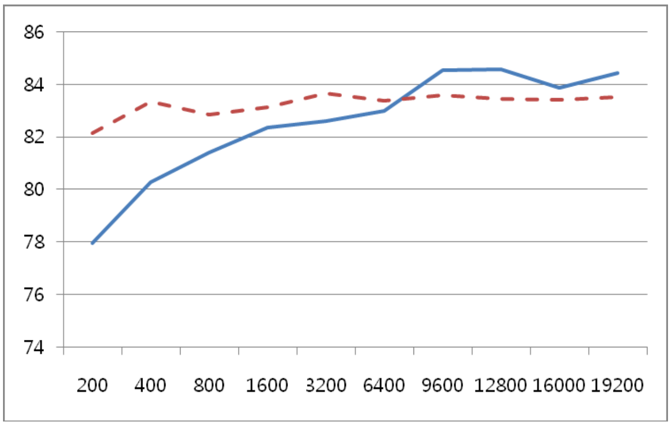

Fig. 5 Average accuracy of RBF networks and MLPs with different sample sizes

4.2 Experiment with forest cover types data set

We also used RBF network using K-means clustering to train from various sample sizes of 'forest cover types' data set. The used basis function is also Gaussian because the data set has several attributes that are in Gausian-like distribution. The given number of clusters for K-means clustering is two, because average class value distribution of the forest cover types data set in each sample size is (38%, 48%, 16%) for classes (1, 2, 3 to 7) respectively. We also trained MLPs with the same sample sets for each different sample sizes. In order to train MLPs the given number of hidden layers is the half of the number of attributes plus the number of classes. Because we have relatively large number of hidden layer, the traing time of 500 is given. Table 3 and 4 show the summary of the results. For each sample size four random samples have been selected and four neural networks have been generated for the experiment.

The initial sample size for training is 200, and the size of samples is doubled as the while loop runs. The given mid_limit value for sample size is 102,400, and the sample size increment from the mid_limit is 51,200. The rest of the data set after sampling is used for testing.

In the table 3 and 4, the third column, improvement(%), means the percentage of improvement in accuracy compared to the neural networks of previous sample size, and the fourth column represents the difference of maximum and minimum values of accuracy among the neural networks in the given sample size, and the last column is for the average computation time in second. The used computer is a pentium 4 personal computer with 2MB main memory.

Table 3. RBF networks for 'forest cover types' data set with various sample sizes

Samp. size	Average Accuracy (%)	Improve -ment(%)	Diff. of max & min accuracy (%)	Average compu. time(sec)
200	62.4881	NA	3.2482	6
400	64.1559	1.6678	3.2483	10
800	65.8715	1.7156	1.4655	15
1,600	67.4969	1.6254	2.1950	23
3,200	68.0128	0.5159	1.2103	36
6,400	68.6423	0.6295	0.7520	80
12,800	69.0365	0.3942	0.5216	173
25,600	68.9293	-0.1072	0.4598	263
51,200	69.0065	0.00772	0.6869	504
102,400	69.2892	0.2827	0.5409	838
153,600	69.2987	0.0095	0.5868	1021
204,800	69.2851	-0.0136	0.3491	1882
256,000	62.7933	-6.4918	0.2313	2796

If we look at table 1, sample size 153,600 has the best accuracy, and the second best is sample size 102,400. Because accuracy of MLP increase as the sample size grow, we may choose sample size 153,600 as the training point of the MLP. In other words, because the difference of accuracy between sample size 153,600 and 103,200 is only 0.0095%, and the difference of max and min accuracy between the two is similar, we choose the sample size of 153,600. Note that as the sample size increases, accuracy does not increase monotonically in RBF networks. Note also that bigger sample sizes have less fluctuation in difference of maxmum and minimum accuracy values.

Table 4. MLP networks for 'forest cover types' data set with various sample sizes

Samp. size	Average Accuracy (%)	Improve -ment(%)	Diff. of max & min accuracy (%)	Average compu. time(sec)
200	60.9312	NA	4.1225	25
400	62.2087	1.2775	5.0622	50
800	66.1581	3.9494	1.2054	101
1,600	68.1597	2.0016	2.5243	201
3,200	70.2124	2.0527	2.6439	403
6,400	72.9120	2.6996	1.5601	807
12,800	75.4644	2.5524	1.2752	1,652
25,600	76.9944	1.5300	0.3487	3,250
51,200	77.9508	0.9564	1.3631	6,616
102,400	78.7463	0.7955	0.9217	13,521
153,600	79.3981	0.6518	0.292	20,984
204,800	79.3237	-0.0744	0.7156	28,807
256,000	68.0523	-11.2724	0.4502	36,794

In table 4 the results of sample sizes other than 15,600 and 204,800 are also presented for reference. If we look at table 4, sample size 153,600 has the best average accuracy, and the second best is sample size 204,800. Note that even with 1.25 times bigger sample size, the accuracy improvement is -0.074% so that we may stop further iteration.

Note also that the training of MLPs takes tens of times longer than that of RBF networks so that without the help of RBF networks it will take very long time. Fig. 2 displays the change of prediction accuracies of RBF networks (dotted line) and MLPs (solid line) for the data set more clearly. In the figure X axis represents the sample size and Y axis represents average prediction accuracy.

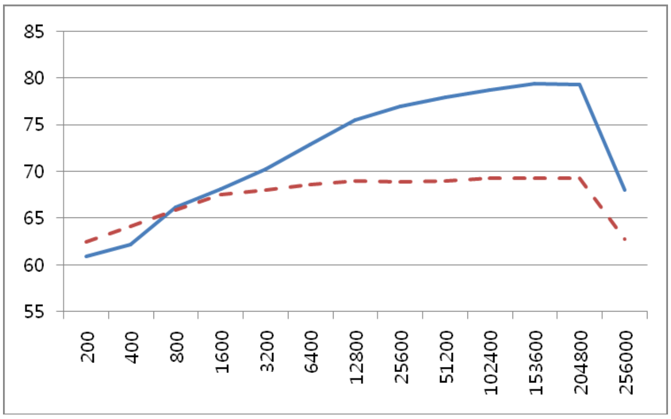

Fig. 2 Average accuracy of RBF networks and MLPs with different sample sizes

5 Conclusion

It is known that neural networks are one of the most successful data mining or machine learning tools for prediction, so that neural networks are widely accepted for the tasks. There are two kinds of neural networks that are widely used for classification – multi-layer perceptrons (MLPs) and radial basis function (RBF) networks. While good points of MLPs their general applicability to almost all domain, good points of RBF networks is relatively fast training time with good predictability. Some drawbacks are high computational complexity in MLPs and domain dependency of basis function in RBF networks. But, whatever neural networks are used, the neural networks may not always be the best predictors due to the fact that they are trained based on some greedy algorithms with limited data sets and the knowledge of human experts. So, some improvements may be possible.

Because the target data sets in data mining tasks contain a lot of data, random sampling has been considered a standard method to cope with large data sets that are very common in data mining task. But, simple random sampling might not generate perfect samples that are good for the used data mining algorithms, and the task of determining a proper sample size is arbitrary so that the reliability of the trained data mining models may not be good enough to be trusted. Moreover, it takes very long computing time to train MLPs so that we have only limited chance to do repeated sampling.

In order to overcome the problem, we propose a method that first applies a repeated progressive sampling method with various sample sizes for RBF networks to decide the best random samples. Then, good sample sizes from RBF networks are used to train MLPs. Experiments with a real world data set showed very promising results.

References:

[1] D.T. Larose, *Data Mining Methods and Models*, Wiley-Interscience, 2006.

[2] C.M. Bishop, *Neural networks for pattern recognition*, Oxford University press, 1995.

[3] J. Stastny, V. Skorpil, Analysis of Algorithms for Radial Basis Function Neural Network, *IFIP International Federation for Information Processing*, Vol. 245, *Personal Wireless Communications*, eds. B. Simak, R. Bestak, E. Kozowska, Springer, 2007, pp. 54-62.

[4] R.J. Howlett, L.C. Jain, *Radial Basis Function Networks I: recent developments in theory and applications*, Physics-Verlag, 2001.

[5] S. Russel, P. Novig, *Artificial Intelligence: a Modern Approach*, 2nd ed., Prentice Hall, 2002.

[6] M.L. Minsky, S.A. Papert, *Perceptrons – extended edition: an introduction to computational geometry*, MIT press,1987.

[7] M.W. Gardner, S.R. Dorling, Artificial Neural networks (The Multilayer Perceptron) – A Review of Applications in the Atmospheric Sciences, *Atmospheric Environment*, vol. 32, no. 14/15, 1998, pp. 2627-2636.

[8] N.E. Mastorakis, The Optimal Multi-layer Structure of backpropagation Networks, *WSEAS Transactions on Information Science and Applications*, vol. 3, issue 9, 2006, pp. 1632-1637.

[9] R P. Lippmann, An Introduction to Computing with Neural Nets, *IEEE Acoustic, Speech, and Signal Processing Magazine*, vol . 4, 1987, pp. 4 -12.

[10] K. Hornik, M. Stinchcombe, H. White, Multilayer Feedforward Networks are Universal Approximator, *Neural Networks*, vol. 2, 1989, pp. 359-366.

[11] C. Lin, Implementation Feasibility of Convex Recursive Deletion Regions using Multi-Layer Perceptrons, *WSEAS Transactions on Computers*, vol. 7, issue 1, 2008, pp. 24-31.

[12] C. Lin, Neural Network Structures with Constant Weights to Implement Dis-jointly Removed Non-convex (DJRNC) Decision Regions: Part A – Properties, Model, and Simple case, *Proceedings of the 8th WSEAS International Copnference on Neural Networks*, vol. 8, 2007, pp. 13-18.

[13] C. Cabrelli, U. Molter, R. Shonkwiler, A Constructive Algorithm to Solve Convex Recursive Deletion (CoRD) Classification Problems via Two-layer Perceptron Networks, *IEEE Transactions on Neural Networks*, vol. 11, no. 3, 2000, pp. 811-816.

[14] Y. Lin, C. Huang, C. Lin, Determination of Insurance Policy Using Neural Networks and Simplied Models with factor Analysis Technique, *WSEAS Transactions on Information Science and Applications*, vol. 5, issue 10, 2008, pp. 1405-1415.

[15] A.C. Comrie, Comparing Neural Networks and Regression Models for Ozone Forecasting, *Journal of Air and Waste Management*, vol. 47, 1997, pp. 653-663.

[16] P. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Addison Wesley, 2006.

[17] M.J.L. Orr, *Introduction to Radial Basis Function Networks*, http://www.anc.ed.ac.uk/~mjo/intro.ps, 1996.

[18] Z. Zainuddin, O. Pauline, Function Approximation Using Artificial Neural Networks, *WSEAS Transcations on Mathematics*, vol. 7, issue 6, 2008, pp. 333-338.

[19] G. Baylor, E.I. Konukseven, A.B. Koku, Control of a Differentially Driven Mobile Robot Using Radial Basis Function Based Neural Networks, *WSEAS Transcations on Systems and Control*, vol. 3, issue 12, 2008, pp. 1002-1013.

[20] A. Esposito, M. Marinaro, D. Oricchio, S. Scarpetta, Approximation of Continuous and Discontinuous Mappings by a Growing Neural RBF-based Algorithm, *Neural Networks*, Vol. 13, No. 6, 2000, pp. 651-656.

[21] O. Buchtala, M. Klimek, B. Sick, Evolutionary Optimazation of Radial Basis Function Classifiers for Data Mining Applications, *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, Vol. 35, No. 5, 2005, pp. 928-947.

[22] A. Hofmann, B. Sick, Evolutionary Optimazation of Radial Basis Function Networks for Intrusion Detection, *Proceedings of the International Joint Conference on Neural Networks*, Vol. 1, 2003, pp. 415-420.

[23] L. Nikolaos, Radial basis Function Networks to Hybrid Neuro-Genetic RBFNs in Financial Evaluation of Corporations, *International Journal of Computers*, vol. 2, issue 2, 2008, pp. 176-183.

[24] K. Fukunaga, R.R. Hayes, Effects of Sample Size in Classifier Design, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, issue 8, 1989, pp. 873-885.

[25] S.J. Raudys, A.K. Jain, Small Sample Size Effects in Statistical Pattern recognition: Recommendations for Practitioners, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 3, 1991, pp. 252-264.

[26] M.A. Mazuro, P.A. Habas, J.M. Zurada, J.Y. Lo, J.A. Baker, G.D. Tourassi, Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance, *Neural Networks*, Vol. 21, Issues 2-3, 2008, pp. 427-436.

[27] T. Oatesm, D. Jensen, Efficient progressive sampling, *Proceedings of the Fifth International Conference on Knowledge Discovery and data Mining*, 1999, pp. 23-32.

[28] S. Berkman, H. Chan, L. Hadjiiski, Classifier performance estimation under the constraint of a finite sample size: Resampling scheme applied to neural network classifiers, *Neural Networks*, Vol. 21, Issues 2-3, 2008, pp. 476 -483.

[29] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Internal Representation by Error Propagation, *Parallel Distributed Processing*, D.E. Rumelhart, J.L. McClelland eds., The MIT Press, vol. 1, 1986.

[30] L. Tarassenko, *Guide to Neural Computing Applications*, Hodder Arnold Publication, 1998.

[31] D. Balageas, C. Fritzen, A. Guemes eds., *Structural Health Monitoring*, Independent Pub Group, 2006.

[32] A.R. Omondi, J.C. Rajapakse, *FPGA Implementations of Neural Networks*, Springer, 2006.

[33] H. White, Learning in Artificial Neural Networks: A Statiscal Perspective, *Neural Computation*, vol. 1, 1989, pp. 425-465.

[34] G. Bayar, E.I. Konukseven, A.B. Koku, Control of a Differentially Driven mobile Robot Using Radial Basis Function Based Neural Networks, *WSEAS Transactions on Systems and Control*, Vol. 3, Issue 12, 2008, pp. 1002-1013.

[35] V.R. Mankar, A.A. Ghatol, Use of RBF Neural Network in EMG Signal Noise Removal, *WSEAS Transactions on Circuits and Systems,* Vol. 7, Issue 4, 2008, pp. 259-265.

[36] M. Qu, F.Y. Shih, J. Jing, H. Wang, Automatic solar flare detection using MLP, RBF, and SVM, *Solar Physics*, vol. 27, no. 1, 2003, pp. 157-172.

[37] S. Marinai, M. Gori, G. Soda, Artificial neural networks for document analysis and recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, 2005, pp. 23-35.

[38] T. Piggio, F. Girosi, Regularization Algorithms for Learning That are Equivalent to Multilayer Networks, *Science*, Vol. 2247, 1990, pp. 987-982.

[39] D. Newman, *UCI KDD Archive* [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science, 2005.

[40] R. Kohavi, Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid, *Proceedings of the scond international conference on knowledge discovery and data mining*, 1996, pp. 202-207.

[41] J.A. Blackard, J.D. Denis J, Comparative Accuracies of Artificial Neural Networks and Discriminant Analysis in Predicting Forest Cover Types from Cartographic Variables, *Computers and Electronics in Agriculture*, vol. 24, no. 3, 2000, pp. 131-151.