The Effect of Training Set Size for the Performance of Neural Networks of Classification

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: - Even though multilayer perceptrons and radial basis function networks belong to the class of artificial neural networks and they are used for similar tasks, they have very different structures and training mechanisms. So, some researchers showed better performance with radial basis function networks, while others showed some different results with multilayer perceptrons. This paper compares the classification accuracy of the two neural networks with respect to training data set size, and shows the performance of the two neural networks can be differently dependent on training data set size. Experiments show the tendency that multilayer perceptrons have better performance in relatively larger training data sets for some data sets, even though radial basis function networks have better performance in relatively smaller training set size for the same data sets. The experiment was done with four real world data sets.

Key-Words: - neural networks**,** multilayer perceptrons, radial basis function networks, training data set, prediction

1 Introduction

Many algorithms have been developed and applied for the task of data mining and machine learning [1, 2]. Among the algorithms, neural networks have played an important role for the task. There are many neural network algorithms suggested [3, 4, 5, 6]. Among them multilayer perceptrons (MLPs) [57] are important tools in various classification tasks so that there are many success stories using MLPs [8, 9]. On the other hand, radial basis function networks (RBFNs) are another neural networks of which functionality is comparable to that of MLPs [10]. Recently many researchers have reported success stories using RBFNs, and have shown better performance of the RBFNs than MLPs [11, 12, 13, 14, 15].

Even though MLPs and RBFNs are used for similar purposes, the two neural networks have very different network structures and training mechanisms. A MLP have several layers including an input layer, several hidden layers, and an output layer, while a RBFN has, in general, only three layers including an input layer, a hidden layer, and an output layer. The hidden layer of RBFN is trained by clustering in general. If the number of hidden layers of MLPs is small, the two neural networks look similar in shape.

Due to the structural difference between the two neural networks, we have very different training mechanisms for the two neural networks. MLPs use backpropagation algorithm to train connection weights between layers, and because the backpropagation algorithm relys on gradient descent, computing time can be long [16]. But a good point of MLPs is their applicability to any field of pattern recognition tasks of supervised learning. On the other hand, RBFNs perform clustering in the hidden layer. Depending on where a data point belongs to a cluster, the data point will have different effect on the output [17, 18]. So the performance of RBFNs can be good, if we have chosen appropriate radial basis functions for target data sets [19]. Some weak point of RBFNs is ineffectiveness for irrevant features, because all features are treated equally in distance calculation. On the other hand, MLPs can have good performance, even if the data set contains irrevant features.

In section 2, we provide the related work to our research, and in section 3 we provide the principles of neural networks like MLPs, RBF networks, and sampling. In sections 4 we present our method of experiment, and several experiments were run to see the property of the two neural networks in section 5. Finally section 6 provides some conclusions.

2 Related work

Neural networks can be divided into two classes based on how the nodes in the networks are interconnected – recurrent neural networks and feed-forward neural networks. In recurrent networks the connection can be interconnected recurrently, while feed-forward networks cannot [20]. MLPs and RBFNs belong to feed-forward networks. Because there can be many parameters for the optimization of the neural networks, many evolutionary search algorithms were suggested [21, 22, 23, 24]. Evolutionary search algorithms try to find global optimal solutions so that it is possible to find better neural 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.

The performance of inducted knowledge is also dependent on the available data sets, because in most cases we do not have complete data sets that reflect the application domain well. Training data set size and data set property itself play important role for the performance of the trained knowledge model. Chan et. Al. [25] discussed the effect of sample size in the design of quadratic and neural network classifiers, and found that good sample size depends on the classifier, dimensionality of feature size, and distribution of features. Raudys and Jain [26] considered sample size in practical sense. They recommended small-sized samples for feature selection and error estimation in pattern recognition fields of several classifiers. Mazuro et. al. [27] showed that imbalanced class value distribution in data set plays an important role in the development of neural networks especially for medical domain. Three sampling schemes, arithmetic, geometric, and dynamic sampling scheme are discussed to find a best decision tree in [28]. The authors found that the accuracy of decision tree classifier 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 their sampling scheme.

3 The method of experiment

We apply three existing techniques in our method; multilayer perceptrons, radial basis function networks, and a sampling technique.

3.1 Multilayer perceptrons

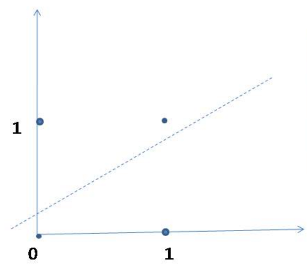
The task of classification with neural networks can be stated as a function approximation problem.

When we are given a set **S** 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 an 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. That is, the prior function f can be written as follows:

$$f : I \rightarrow O \qquad (1)$$

where I is the domain of input and O is the domain of output.

MLPs were introduced in middle of 80's to enhance the limited capability of perceptrons, because perceptrons have difficulty in solving linear separability problem like XOR problem [29]. Fig. 1 shows the linear separability problem of XOR. We cannot draw a line that can separate 0 from 1 in XOR result, so is true with the perceptron.



INPUT		XOR
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 1 XOR problem

A perceptron has only two layers, input layer and output layer. The output of a perceptron is weighted sum of its inputs.

$$f'(\Sigma_{i=0\sim k}\, w_i x_i - \theta) \qquad (2)$$

where f' is an activation function, θ is threshold for output, w_i is a weight, and x_i is an input. There are k+1 input nodes. The activation function determines the final output. Three activation functions like step function, sign function, and sigmoid function are mostly used. Among them sigmoid functions are widely used. An example of sigmoid function is $1/(1 + e^{-x})$. Fig. 2 shows a simple perceptron.

On the other hand, MLPs have three layers, an input layer, an output layer, and a hidden layer. The hidden layer can contain several layers. MLPs became very well known by the efforts of the parallel distributed group [30]. An important property of MLPs is backpropagation learning algorithm [31], and by the learning algorithm a variety problem could be solved including linear separability problem that were impossible to solve with perceptrons.

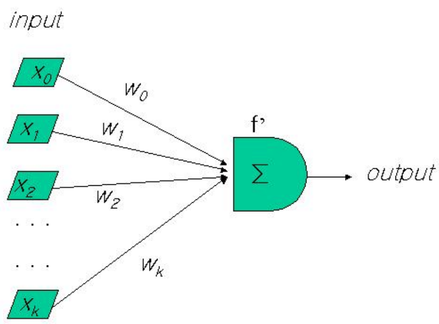


Fig. 2 A perceptron

Unlike other statistical method MLPs do not need assumptions about data distribution so that they are good at prediction tasks where we don't have much statistical information about data. There are many cases that report successful application of MLPs [32, 33]. Because of structural similarity, we can consider that a MLP is a combination of perceptrons with different training mechanism. Because MLPs have multiple hidden layers, they have more power in predictability than perceptrons. Fig. 3 shows a schematic view of a simple multilayer perceptron where the number of hidden layer is two.

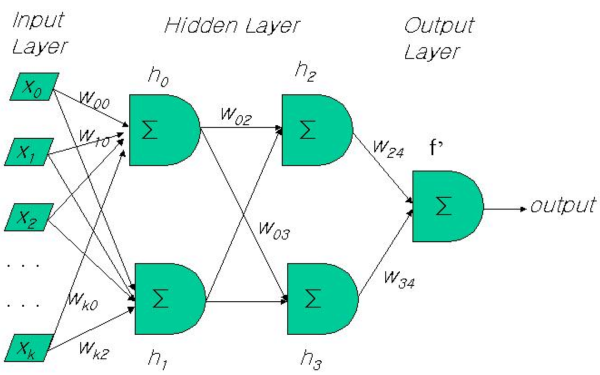


Fig.3 A MLP

3.2 Radial basis function networks

RBFNs were also introduced in late 80's [34] slightly later than MPLs. The function of RBFNs is based on the function of actual neurons like visual cortices that have the property of being sensitive to some particular visual characteristics [35].

The task of classification with RBFN is a function approximation problem, so we want to find f' of f in equation (1).

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

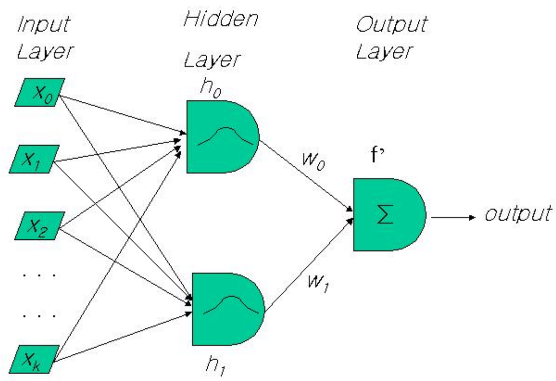


Fig. 4 A RBFN

There can be a variety of radial basis functions, for example, Gaussian, multiquadric, cauchy, etc., and among them Guassian is mostly used. Center point and radius are two parameters for the radial function. If we use Gaussian as a basis function, mean is the center and variance is the radius. In order to find appropriate center and radius, we may use some unsupervised learning algorthms like K-means clustering.

3.3 Geometric sampling

When we sample data, we can increase the sample size progressively. In geometric sampling the sample size is increased geometrically as we sample more and more. We can define sample size G_i for a sample set i in geometric sampling with the following equation:

$$G_i = G_0 \times C^i \tag{3}$$

Here, G_0 is the initial sample size and C is a constant for increment.

So, we can have a geometrical progression of sample sets in size, G_0, $G_1 = G_0 \cdot C$, $G_2 = G_0 \cdot C^2$, $G_3 = G_0 \cdot G^3$, and so on. For example, if $G_0 = 200$ and C = 2, then G_1 = 400, G_2 = 800, G_3 = 1,600, and so on. As we can see from the example, if we use geometric sampling, sooner or later we can have very big sample sizes.

3.4 The method of experiment
We want to see the effects of training data set size in the performance of MLPs and RBFNs. For this purpose some large data sets will be chosen, then geometric and progressive sampling will be done to simulate the situation of various training data set sizes, from small to large. The progressive sampling will be stopped when the sampling size becomes about the half of original data set for us to have enough test data also.

The following is a brief description of the procedure of the experiment for sampling.

INPUT: a data set
σ: initial sample size
OUTPUT: Ar, Am
/* A_{rbf}: the set of average accuracy of RBFNs,
A_{mlp}: the set of average accuracy of MLPs */
j := 1;
Do While σ is about the half of the target data set
For i =1 to 4 **do** /* repeat 4 times */
Do random sampling of size σ;
Train and test RBFN and MLP;
End For;
r_j := the average accuracy of the RBFN;
m_j := the average accuracy of the MLP;
$A_{rbf} := A_{rbf} \cup \{r_j\}$;
$A_{mlp} := A_{mlp} \cup \{m_j\}$;
$\sigma := \sigma \times 2$; j++;
End while;

In the above procedure we repeat 4 times for each sample size to remove accidental effect in sampling. The used radial basis function in RBFN is Gaussian and k-means clustering is used. The number of hidden layers for MLPs is given appropriately.

4 Experimentation
Experiments were run using four data sets in UCI machine learning repository [36] called 'ozone', 'census income', 'statlog', and 'forest cover types' to see the effect of the training data set size for the accuracy of the neural networks. The four data sets have relatively large data set size, so they are good for the experiment

4.1 Experiment with ozone data set
The ozone data set [37] contains two koids of data sets- ozone one hour data set and ozone two hour data set. Between the two data sets, ozone eight hour data set is selected. For 'ozone' data set 4 was given as the number of clusters for RBF networks, and the number of hidden layers for MLPs is half of the number of attributes plus the number of classes, and traing time is 500. The number of instances in 'ozone' data set is 2,536. The initial sample size for training is 200, and the rest of the data set after sampling is used for testing, so we have bigger test set data when sample size is small. Because the data set is not large enough for geometric sampling, we did additional sampling in the size of 1,200 that is almost half of the original data set size. In the experiment four random samples for each sample size are used. The tables contain average accuracy values.

Table 1. RBF networks and MLPs for 'ozone' data set with different sizes of training data sets

Sample size	Accuracy of RBFN (%)	Accuracy of MLP(%)
200	93.0831	92.0269
400	93.7559	92.5843
800	93.6563	92.6759
1,200	93.497	93.1222
1,600	93.6064	93.7399

If we look at table 1, we can notice that RBFNs are usually better for 'ozone' data set. So, for the data set RBFN is better choice.

Fig. 5 displays the trend of prediction accuracy of RBFNs (dotted line) and MLPs (solid line) for the ozone data set more clearly as the training data set size grows. In the figure X axis represents the sample size and Y axis represents prediction accuracy .

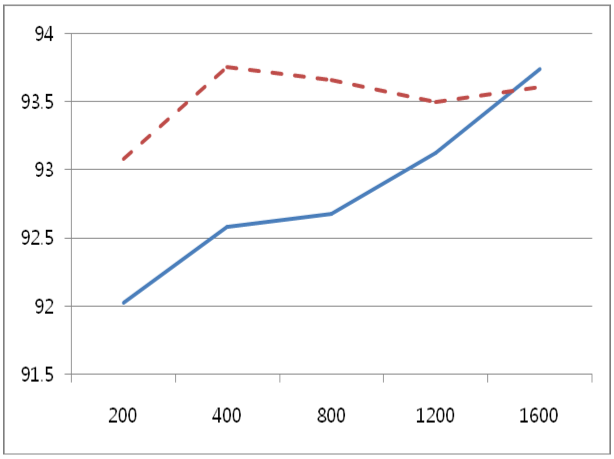


Fig. 5 The accuracy values of RBFNs (dotted line) and MLPs (solid line) for 'ozone' data set with different sizes of training data sets

Table 2 shows individual results of the experiment when sample size is 200, 400, 800, 1,200 and 1,600 for further reference.

Table 2. RBFNs and MLPs for 'ozone' data set when sample size is 200, 400, 800, 1,200 and 1,600

Sample size	Accuracy of RBFN(%)	Accuracy of MLP(%)
200	91.4347	92.1627
	93.7446	92.3736
	93.9227	91.8402
	93.2305	91.7309
average	93.0831	92.0269
400	94.0019	92.8304
	93.4864	92.9709
	93.8144	92.0337
	93.7207	92.5023
average	93.7559	92.5843
800	93.887	92.2722
	93.714	92.3299
	93.4833	93.0796
	93.5409	93.0219
average	93.6563	92.6759
1,200	93.5532	93.1784
	94.2279	93.4783
	93.3283	93.3283
	92.8786	92.5037
average	93.497	93.1222
1,600	93.8972	93.469
	93.0481	93.7968
	93.5829	94.1176
	93.8972	93.576
average	93.6064	93.7399

4.2 Experiment with census income data set

Experiments were also run using a very large data set in the UCI machine learning repository called 'census-income' [38]. The total number of instances for training and testing is 299,285. There are two classes, yearly income being greater than or equal to 50,000 and less than 50,000. There are total of 67,652 duplicate or conflicting instances. The total number of attributes including class attribute is 42. Among them eight attributes are continuous attributes. The census income data set has very big data records, and the size of the data set is very large, so, 84 clusters are used in K-means clustering. In order to train MLPs the given number of hidden layers is ten, and the traing time is 500, because the size of data set is very large.

Table 3 shows the result of training for the two neural network algorithms. The initial sample size for training is 2,500, and the size of samples is doubled as the while loop runs, and we stop sampling when the sample size reaches to 40,000, and sample size 60,000 is tried also. The rest of the data set after sampling is used for testing, so we have bigger test data sets when sample size is small.

Table 3. RBFNs and MLPs for 'census income' data set with different sizes of training data set

Sample Size	Accuracy of RBFN (%)	Accuracy of MLP (%)
2,500	93.91588	94.02115
5,000	94.3767	94.19763
10,000	94.33915	94.0922
20,000	94.35875	94.62895
40,000	94.48245	94.6336
60,000	94.52223	94.70715

If we look at table 3, we can notice the fact that when sample sizes are small, the accuracy of RBFNs is also mostly better, but when sample sizes are large, the accuracy of MLPs is better. Fig. 6 displays the trend of prediction accuracy of RBFNs (dotted line) and MLPs (solid line) for census income data set more clearly as the training data set size grows. In the figure X axis represents the sample size and Y axis represents prediction accuracy .

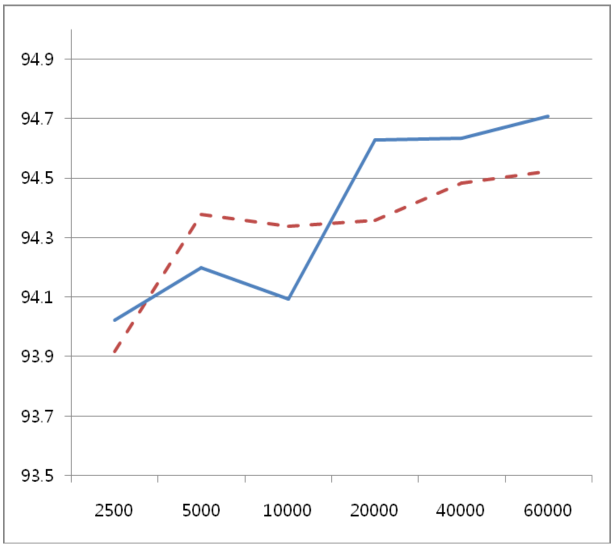


Fig. 6 The accuracy values of RBFNs (dotted line) and MLPs (solid line) for 'census income' data set with different sizes of training data sets

Table 4 shows individual result of the experiment when sample size is 5,000, 10,000, 20,000, and 40,000.

Table 4. RBFNs and MLPs for 'census income' data set when sample size is form 5,000 to 40,000

Sample size	Accuracy of RBFN(%)	Accuracy of MLP(%)
5,000	94.3990	93.8274
	94.5165	94.5655
	94.3565	93.7961
	94.2348	94.6065
average	94.3767	94.19763
10,000	94.3053	93.7912
	94.2313	94.1283
	94.4726	93.9001
	94.3474	94.5492
average	94.33915	94.0922
20,000	94.3076	94.4279
	94.4508	94.8157
	94.5101	94.6363
	94.1665	94.6359
average	94.35875	94.62895
40,000	94.5431	94.8223
	94.5331	94.5481
	94.3988	94.6329
	94.4548	94.5311
average	94.48245	94.6336

4.3 Experiment with statlog data set

Experiments were also run using a medium-sized data set in the UCI machine learning repository called 'statlog' [39]. The data set consists of the multi-spectral values of pixels in 3 by 3 neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood. In the data set, the class of a pixel is coded as a number, and there are seven classes, but there is no data for class 6. The total number of attributes is 36 which comes from 4 spectral bands multiplied by 9 pixels in neighbourhood, and all of them have numerical values in the range 0 to 255. The total number of instances is 6,435.

The statlog data set has relatively small number of instances compared to the other data sets and all attributes are numeric. So, 36 was chosen as the number of clusters for clustering. In order to train MLPs the given number of hidden layers is eighteen, and the traing time is 500.

Table 5 shows the result of training for the two neural network algorithms. The initial sample size for training is 400, and the size of samples is doubled as the while loop runs, and we stop sampling when the sample size reaches to about half of the data set size. The rest of the data set after sampling is used for testing, so we have bigger test data sets when sample size is small.

Table 5. RBFNs and MLPs for 'statlog' data set with different sizes of training data set

Sample size	Accuracy of RBFN (%)	Accuracy of MLP (%)
400	84.62718	84.58163
800	85.11978	86.4463
1,600	87.26538	87.17753
3,200	87.49035	88.49035

If we look at table 5, we can notice the fact that there is almost no relationship between sample size and accuracy between the two nueral networks. So, experiments were done more for some middle sample sizes. Table 6 shows the result of training at some middle sample sizes that were not considered at the experiment in table 5. It also shows that when sample sizes are small, the accuracy of RBFNs is mostly better, but when sample sizes are large, the accuracy of MLPs is better. Figure 7 shows the combined result of table 5 and table 6. In the figure X axis represents the sample size and Y axis represents prediction accuracy, and dotted line is for RBFNs and solid line is for MLPs.

Table 6. RBFNs and MLPs for 'statlog' data set with another different sizes of training data set

Sample size	Accuracy of RBFN (%)	Accuracy of MLP (%)
600	84.9621	84.84213
2,400	87.33123	88.11173
2,800	87.46305	88.23328

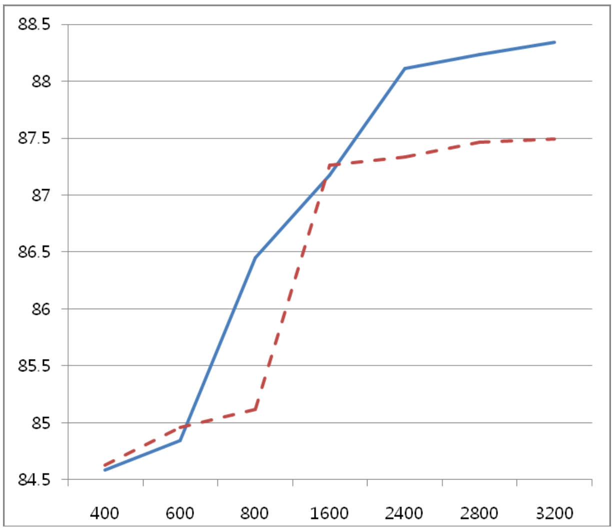


Fig. 7 The accuracy values of RBFNs (dotted line) and MLPs (solid line) for 'statlog' data set with different sizes of training data sets

Table 7 shows individual result of the experiment when sample size is 400, 600, 800, and 1,600.

Table 7. RBFNs and MLPs for 'statlog' data set when sample size is 400, 600, 800, and 1,600

Sample size	Accuracy RBFN(%)	Accuracy MLP(%)
400	85.1698	84.4905
	85.2693	85.5178
	84.0762	84.3248
	83.9934	83.9934
average	84.62718	84.58163
600	85.9640	84.353
	85.1757	86.0154
	83.9246	84.3188
	84.7841	84.6813
average	84.9621	84.84213
800	84.8980	87.2227
	85.9982	86.3354
	84.5075	85.5723
	85.0754	86.6548
average	85.11978	86.4463
1,600	87.1768	87.0527
	87.2622	86.0835
	88.2316	87.7973
	86.3909	87.7766
average	87.26538	87.17753

4.4 Experiment with forest cover types data set

The forest cover types data set [40] includes forest information in four wilderness areas found in the Roosevelt National Forest of northern Colorado. It has twelve continuous attributes as independent variables, while seven major forest cover types were used as a dependent variable. The total number of instances is 581,012. We chose 14 as the number of clusters for clustering. In order to train MLPs the given number of hidden layers is the half of the number of attributes plus the number of classes, and the traing time is 500 for the forest cover types data set, because the forest cover types data set contains continuous values only for dependent variables.

Table 8 shows the result of training for the two neural network algorithms. The initial sample size for training is 400, and the size of samples is doubled as the while loop runs, and we stop sampling when the sample size reaches to about half of the data set size. The rest of the data set after sampling is used for testing, so we have bigger test set data when sample size is small.

Table 8. RBFNs and MLPs for 'forest cover types' data set with different sizes of training data set

Sample size	Accuracy of RBFN (%)	Accuracy of MLP (%)
200	62.4881	60.9312
400	64.1559	62.2087
800	65.8715	66.1581
1,600	67.4969	68.1597
3,200	68.0128	70.2124
6,400	68.6423	72.9120
12,800	69.0365	75.4644
25,600	68.9293	76.9944
51,200	69.0065	77.9508
102,400	69.2892	78.7463
204,800	69.2851	79.3237

If we look at table 8, we can notice also the fact that when sample sizes are small, the accuracy of RBFNs is also better, but when sample sizes are large, the accuracy of MLPs is better. Fig. 8 displays the trend of prediction accuracy of RBFNs (dotted line) and MLPs (solid line) for forest cover types data set more clearly as the training data set size grows. In the figure X axis represents the sample size and Y axis represents prediction accuracy .

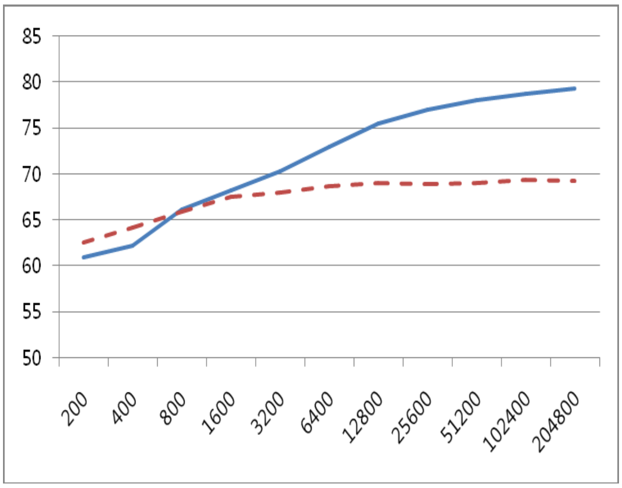


Fig. 8 The accuracy values of RBFNs (dotted line) and MLPs (solid line) for 'forest cover types' data set with different sizes of training data sets

Table 9 shows individual result of the experiment when sample size is 200, 400, 800 and 1,600.

Table 9. RBFNs and MLPs for 'forest cover types' data set when sample size is 200, 400, 800, and 1,600

Sample size	Accuracy of RBFN(%)	Accuracy of MLP(%)
200	61.9033	58.5325
	61.1299	61.6938
	62.5409	60.8436
	64.3781	62.6550
average	62.4881	60.9312
400	65.3745	64.7122
	61.9159	62.6313
	65.1642	61.8411
	64.1688	59.6500
average	64.1559	62.2087
800	65.9504	65.8595
	65.2043	66.9400
	66.6698	66.0983
	65.6615	65.7346
average	65.8715	66.1581
1,600	67.7451	67.5559
	66.6755	69.8988
	66.6964	67.3745
	68.8705	67.8096
average	67.4969	68.1569

5 Conclusion

There are many methods for the task of data mining and machine learning. Among the methods, neural networks are widely accepted, and neural networks are considered very successful tools for the task. There are many neural network algorithms suggested. Among them multilayer perceptrons(MLPs) and radial basis function networks(RBFNs) are two representative neural network algorithms that are widely used for classification task. Interestingly, some researchers have reported that the performance of radial basis function networks are better than that of multilayer perceptrons for their applications, but some other researchers have reported the opposite results. This conflicting reports might be because of the fact that whichever neural network is used, there are many parameters that affect the performance of the used neural network. That is, the structure and training methods of neural network give us many possibilities for further optimization. For example, the structure of neural network is usually determined by the knowledge of human experts, and the training is based on some greedy search algorithms. Another factor is that the performance of a neural network is also dependent on the available data sets.

Because the target data sets in machine learning or data mining tasks may not contain large enough data that represent the target domain well, the trained neural networks might not represent the best neural network for the target application. So we want to find out any relationship between training data set size and the performance of the two neural network algorithms, RBFNs and MLPs.

We experimented the two representive neural network algorithms, RBFNs and MLPs, for classification tasks of some data sets. A repeated progressive sampling method with various sample sizes was applied to find out if there is any relationship between data set size and the performance. The experiment was done with four real world data sets. Among them, one data set showed that RBFN is mostly better than MLPs. On the other hand, it was found out that even though the performance of RBFNs is good when the size of training data set is relatively small, the performance is reversed when the size of training data set size is relatively large in the other three data sets. From this fact, we can notice that the accuracy of MLPs can be improved reltively more than that of RBFNs, if we can have more data.

References:

[1] D.T. Larose, *Data Mining Methods and Models*, Wiley-Interscience, 2006.
[2] S. Marsland, *Machine Learning: An Algorithmic Perspective*, Chapman & Hall/CRC, 2009.

[3] C.M. Bishop, *Neural networks for pattern recognition*, Oxford University press, 1995.
[4] J. Heaton, *Introduction to Neural Networks for C#,* 2nd ed., Heaton Research Inc., 2008.
[5] Z. Zainuddin, O. Pauline, Function Approximation Using Artificial Neural Networks, *WSEAS Transcations on Mathematics*, vol. 7, issue 6, 2008, pp. 333-338.
[6] S. Pervez, I. Ahmad, A. Akram, S.U. swati, A Comperative Analysis of Artificial Neural Network Technologies in Intrusion Detection Systems, *Proceedings of the 6th WSEAS International Conference on Multimedia, Internet & Video technologies*, 2006, pp. 84-89.
[7] R.P. Lippmann, An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, vol. 3, no. 4, 1987, pp. 4-22.
[8] 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.
[9] 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.
[10] M.J.L. Orr, *Introduction to Radial Basis Function Networks*, http://www.anc.ed.ac.uk/~mjo/intro.ps, 1996.
[11] 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.
[12] F.Y. Shih, S.S. Chen, Adaptive document block segmentation and classification, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 27, no. 5, 1996, pp. 797-802.
[13] 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.
[14] C. Baroglio, A. Giordana, M. Kaiser, M. Nuttin, R. Piola, Learning controllers for industrial robots, *Machine learning*, vol. 23, no. 2-3, 1996, pp. 221-249.
[15] K.Z. Mao, K.C. Tan, W. Ser, Probabilistic Neural Network Structure Determination for Pattern Classification, *IEEE Transactions on Neural Networks*, Vol. 11, issue 4, 2000, pp. 1009-1016.
[16] S. Russel, P. Novig, *Artificial Intelligence: a Modern Approach*, 2nd ed., Prentice Hall, 2002.
[17] 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.
[18] R.J. Howlett, L.C. Jain, *Radial Basis Function Networks I: recent developments in theory and applications*, Physics-Verlag, 2001.
[19] Z. Zainuddin, O. Pauline, Function Approximation Using Artificial Neural Networks, *WSEAS Transcations on Mathematics*, vol. 7, issue 6, 2008, pp. 333-338.
[20] P. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Addison Wesley, 2006.
[21] X. Yao, Evolving Artificial Neural Networks, *Proceedings of the IEEE*, vol. 87, no. 9, 1999, pp. 1423-1447.
[22] 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.
[23] 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.
[24] 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.
[25] H.P. Chan, B. Sahiner, R.F. Wagner, N. Petrick, J. Mossoba, Effects of Sample Size on Classifier Design: Quadratic and Neural Network Classifiers, *SPIE Proceeding Series*, vol. 3034(2), 1997, pp. 1102-1113.
[26] 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.
[27] 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.
[28] T. Oates, D. Jensen, Efficient progressive sampling, *Proceedings of the Fifth International Conference on Knowledge Discovery and data Mining*, 1999, pp. 23-32.
[29] M.L. Minsky, S.A. Papert, *Perceptrons – extended edition: an introduction to computational geometry*, MIT press,1987.
[30] 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.
[31] H. White, Learning in Artificial Neural Networks: A Statiscal Perspective, *Neural Computation*, vol. 1, 1989, pp. 425-465.
[32] L. Tarassenko, *Guide to Neural Computing Applications*, Hodder Arnold Publication, 1998.
[33] D. Balageas, C. Fritzen, A. Guemes eds., *Structural Health Monitoring*, Independent Pub Group, 2006.
[34] J. Park, I.W. Sandberg, Universal Approximation Using Radial-Basis-Function Networks, *Neural Computation*, vol. 3, 1991, pp. 246-257.
[35] T. Piggio, F. Girosi, Regularization Algorithms for Learning That are Equivalent to Multilayer Networks, *Science*, Vol. 2247, 1990, pp. 987-982.
[36] A. Suncion, D.J. Newman, *UCI Machine Learning Repository* [http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html]. Irvine, CA: University of California, School of Information and Computer Sciences, 2007.
[37] K. Zhang, W. Fan, Forecasting skewed biased stochastic ozone days: analyses, solutions and beyond, *Knowledge and Information Systems*, vol.14, no. 3, pp. 299-326, 2008.
[38]http://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29
[39]http://archive.ics.uci.edu/ml/datasets/Statlog+%28Landsat+Satellite%29
[40] 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.