A Pilot Sampling Method for Multi-layer Perceptrons

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

Abstract: As the size of samples grows, the accuracy of trained multi-layer perceptrons grows with some improvement in error rates. But we cannot use larger and larger samples, because computational complexity to train the multi-layer perceptrons becomes enormous and data overfitting problem can happen. This paper suggests an effective approach in determining a proper sample size for multi-layer perceptrons with the help of radial basis function networks that work as a pilot for sampling. Experiments with the two neural network algorithms, multi-layer perceptrons and radial basis function networks show very promising results.

Key-Words: multi-layer perceptrons, radial basis function networks, sampling

1 Introduction
A general purpose neural networks like multi-layer perceptrons (MLPs) have been very widely used for machine learning and data mining task, so MLPs with the smallest error rates for a given data set has been a major concern for their success [1]. But even though MLPs are one of the most successful data mining and 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. So, there are two aspects in improvements; the improvement of the structure of the neural networks and the improvement of connection weights.

Even though there are many algorithms to determine the structure of the neural networks, basically the structure of the networks is 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 backpropagation algorithms are used. The backpropagation algorithms rely on some greedy search algorithms like gradient decent search algorithm. In order to avoid local optima the weights are adjusted slowly so that the computing time can be very large, if the training data set is large.

Moreover, 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, the generated neural networks may not be good enough so that they have almost no improvement in error rates. Even though we go back to sampling, it takes very long computing time to train MLPs so that we have only limited chance to train the neural networks. If we could use some fast way that determines the best sample size for the given data set to train the MLPs, we can save a lot of computing time. For this purpose, this paper suggests to using a pilot sampling method for better neural networks.

In section 2, we provide the related work to our research, and in sections 3 we present our method. 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 [2]. Because of the limited predictability of the perceptron, multi-layer perceptrons have been invented. In multi-layer perceptrons there are two kinds of networks based on how the networks are interconnected – feed-forward neural networks and recurrent neural networks [3]. The weak point of MLPs is computational intensiveness. So most data mining applications that use MLPs prefer small-sized samples or data sets.

Radial basis function (RBF) networks are one of the most popular feed-forward networks [4] that are used as a replacement for MLPs. Even though RBF networks have three layers including the input layer, hidden layer, and
output layer, they differ from a multi-layer perceptron, because in RBF networks the hidden layer performs some computation. A good point of RBF networks is that they can be trained in relatively rapid speed. But, due to the feed-forward nature and functions in the hidden layer, local optima problem may occur.

In order to overcome local optima problem many evolutionary search algorithms were suggested [5, 6, 7, 8] for MLPs as well as RBF networks. 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 evolution ray computation like the representation technique of network structures and weights.

Because some induction methods including neural networks are used to train the data mining models, the behavior of trained data mining models also dependent on the training data set. So, there is research on sample size as well as the property of samples and sampling scheme. Fukunaga and Hayes [9] discussed the effect of sample size for parameter estimates in a family of solutions for classifiers. Raudys and Jain [10] prefer small sized samples for feature selection and error estimation for several classifiers of pattern recognition. In [11] the authors showed that class imbalance in training data has effects in neural network development especially for medical domain. Jensen and Oates [12] 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. In [13] 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 have very different accuracy depending on feature space and sample size.

3 The method
It is not easy to determine an appropriate sample size that is the best for the target data set. In order to overcome this problem we resort to our repeated sampling scheme for RBF networks that considers various sizes of samples to find the best one for the target data set.

We do the sampling until the sample size is less than the half of the target data set, because we assume that we have some large target data set and we want to have enough test data also. Because we use RBF networks to in our method, we should first determine what radial basis function we will use for the step 1 of the procedure. The following is a brief description of the procedure of the sampling scheme. It has two steps.

[Step 1]
/* You should choose a proper radial basis function for target data set first */
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_{ij} := A_{ij} ∪ \{a_{ij}\};
   End for;
   S := S ∪ s;
   A := A ∪ A_{ij};
   v := the average accuracy in A_{ij};
   V := V ∪ \{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 ∪ \{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 ∪ \{d\}; /* D: set of d values */
   If s >= mid_limit Then
      s := s + sample_size_increment; j++;
   Else
      s := s × 2; j++;continue; /* while loop */
End if
End while;

[Step 2]
Choose a sample size as a starting sample size 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 to train MLPs with chosen sample sizes like RBF networks until improvement < predefined_limit.

In the above procedure because the error rates of trained RBF networks are also dependent on used basis functions, we should choose appropriate basis functions.

Even though we do random sampling, because we may have some sampling bias and sampling errors as well as the property of RBF networks, the trained RBF networks may be in variety in accuracy. So, in order to get rid of the effect of variety in accuracy we average the accuracies of the trained neural networks for each sample size. 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. Because the accuracy of MLPs have the tendency of some monotonic increase as the sample size grows, we prefer bigger sample sizes. By selecting bigger sample sizes that generates good RBF networks in average case with satisfactory accuracies, we can have better MLPs for future unseen cases.

4 Experimentation
Experiments were run using a data set in UCI machine learning repository [14] called ‘adult’ to see the effect of the method. The number of instances in the data set is 48,842. The data set was selected, because it is relatively very large and contain lots of values so that it represents the characteristic of data mining domain well. The total number of attributes is 14, and among them six attributes are continuous attributes.

We used RBF network using K-means clustering to train for various sample sizes. The used basis function is Gaussian because the ‘adult’ data set is originated from 1994 census database [15]. The following The given number of clusters for K-means clustering is two. We also trained MLPs with the same sample data sets for each different sample size. In order to train MLPs the given number of hidden layers is two and the traing time is 10,000. 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>
<thead>
<tr>
<th>Samp. size</th>
<th>Average accuracy(%)</th>
<th>Improve -ment(%)</th>
<th>Diff. of max &amp; min accuracy (%)</th>
<th>Average compu. time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>82.15153</td>
<td>NA</td>
<td>2.4239</td>
<td>0.04</td>
</tr>
<tr>
<td>400</td>
<td>83.3527</td>
<td>1.20117</td>
<td>1.6907</td>
<td>0.07</td>
</tr>
<tr>
<td>800</td>
<td>82.86174</td>
<td>-0.49096</td>
<td>0.9783</td>
<td>0.14</td>
</tr>
<tr>
<td>1,600</td>
<td>83.13183</td>
<td>0.27009</td>
<td>1.5071</td>
<td>0.76</td>
</tr>
<tr>
<td>3,200</td>
<td>83.64977</td>
<td>0.51794</td>
<td>1.1419</td>
<td>1.50</td>
</tr>
<tr>
<td>6,400</td>
<td>83.38611</td>
<td>-0.26366</td>
<td>2.0288</td>
<td>1.52</td>
</tr>
<tr>
<td>9,600</td>
<td>83.57734</td>
<td>0.19123</td>
<td>0.6345</td>
<td>2.21</td>
</tr>
<tr>
<td>12,800</td>
<td>83.45717</td>
<td>-0.12017</td>
<td>0.6165</td>
<td>3.01</td>
</tr>
</tbody>
</table>

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 MLPs increase as the sample size grow, we may choose sample size 9600 as the starting point to train the MLPs. 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 as an initial sample to train MLPs. Note that in table 1 as the sample size increases, accuracy does not increase monotonically. Note also that bigger sample sizes have less fluctuation in difference of maximum and minimum accuracy values.

<table>
<thead>
<tr>
<th>Samp. size</th>
<th>Average accuracy(%)</th>
<th>Improve -ment(%)</th>
<th>Diff. of max &amp; min accuracy (%)</th>
<th>Average compu. time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>77.96967</td>
<td>NA</td>
<td>2.6294</td>
<td>91.5</td>
</tr>
<tr>
<td>400</td>
<td>80.27067</td>
<td>2.301</td>
<td>5.5923</td>
<td>186.7</td>
</tr>
<tr>
<td>800</td>
<td>81.41629</td>
<td>1.14562</td>
<td>3.0182</td>
<td>351.9</td>
</tr>
<tr>
<td>1,600</td>
<td>82.36150</td>
<td>0.94521</td>
<td>3.0182</td>
<td>673.4</td>
</tr>
<tr>
<td>3,200</td>
<td>82.58996</td>
<td>0.22846</td>
<td>3.6041</td>
<td>1337.3</td>
</tr>
</tbody>
</table>
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 may take very long time to find the best one. Fig. 1 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.

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>Average Accuracy</th>
<th>Standard Deviation</th>
<th>Overfitting Rate</th>
<th>Training Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>6,400</td>
<td>82.99027</td>
<td>0.40031</td>
<td>4.5545</td>
<td>2780.1</td>
</tr>
<tr>
<td>9,600</td>
<td>84.51573</td>
<td>1.52546</td>
<td>0.3899</td>
<td>3977</td>
</tr>
<tr>
<td>12,800</td>
<td>84.55946</td>
<td>0.04373</td>
<td>0.6921</td>
<td>5340.4</td>
</tr>
</tbody>
</table>

5 Conclusion

There are two kinds of neural networks that are widely used – multi-layer perceptrons (MLPs) and radial basis function (RBF) networks. A good point of MLPs is their general applicability to almost all domain and a good point of RBF networks is relatively fast training time. Some drawbacks are high computational complexity in MLPs and domain dependency of basis function in RBF networks. Even with the some drawbacks, neural networks are widely accepted for data mining or machine learning tasks, and it is known that neural networks are one of the most successful data mining tools for prediction. 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 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. So repeated sampling is an alternative. But, because 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 for MLPs. Experiments with a real world data set showed very promising results.

References:


