A Method for KDD Based on Neural Network Ensemble

Xi Lin

Basic Course Department

Jimei University, Xiamen 361021, P.R. China

Abstract: Neural networks are important tools for KDD (Knowledge Discovery from Databases), but the knowledge derived from neural networks (especially the regression neural networks) is difficult to comprehend, and the duration of training neural networks is rather long for mass data sets. In this paper, a method based on RBF neural network ensemble is proposed for KDD. It is not only fast but also effective in using this method to extract rules from data.

Keywords: KDD, neural network ensemble, radial basis function (briefly, RBF) networks, extracting rules.

1. Introduction

Neural networks have been applied extensively in many fields for their powerful capability in non-linear processing and noise tolerance. Two problems exist when neural networks are applied in KDD. Firstly, the knowledge derived from neural networks (especially the regression neural networks) is difficult to comprehend, since neural networks are a kind of "black boxes". Secondly, their training duration is rather long for large data sets. In this paper, a method for KDD based on RBF network ensemble is proposed trying to solve the above two problems partly.

D.S.Broomhead and D.Lowe [1] first applied RBF to the design of neural networks in 1988. W.A.Light [2] and M.J.D.Powell [3] proved that an RBF network can arbitrarily approximate to any multi-variable continuous function on a compact domain if a sufficient number of RBF units are given. C.H.Li [4] proposed a new kind of radial basis functions which are not only infinitely differentiable but also locally supported. The RBF networks based on this kind of functions can infinitely approach the functions in $L^2(R^n)$. In [5], we have improved the RBF networks which have been used for rapid extraction of fuzzy rules from data.

The RBF networks have some good performances:

- Their learning time is shorter then that of general feed-forward neural networks (such as BP networks).

- Error may not fall in local minimum in their training process.
- There are only few neurons activated when a value is inputted, so they can be used for speedy forecast.
- They have incremental learning ability, so we can increase the learning data without training the whole networks again.

Therefore we try to apply the RBF networks to KDD, especially, in extracting the rules from data with fuzzy attributes.

In order to improve the generalization ability of the RBF networks, we adopt the neural network ensemble technology that was proposed by Hansen and Salamon [6] in 1990. Some research work has been published in [7], [8]. Neural network ensemble is to train a finite number of neural networks and then combine their results, so it can improve the generalization ability of learning systems. But neural network ensemble makes it more difficult to extract the rules. We propose a method on RBF network ensemble and apply it to extraction of the rules form data.

2. A new kind of RBF networks

We define a kind of radial basis functions as follows:

$$R(x;c,\sigma)=\begin{cases}\exp\{1+[(\frac{x-c}{\sigma})^2-1]^{-1}\}, & |x-c|<\sigma,\\ 0, & |x-c|\geq\sigma,\end{cases}$$

where $\sigma>0$ is half the width of the finite support, briefly width, and c is the center. Their figures look like a bell.

Let (X_i,Y_i), $i=1,2,...,n$, be the set of given data, where $X_i=(x_1(i),x_2(i),...,x_N(i))$ and $Y_i=(y_1(i),y_2(i),...,y_P(i))$. So, there are N inputs and P outputs in the system, where N and P represent the number of input and output variables respectively.

Suppose that $[-l_j,l_j]$, $j=1,2,...,N$ is the ranges of input attribute spaces. Let $c_j(1),c_j(2),...,c_j(s_j)$, $j=1,2,...,N$, be the centers of the input attribute subspaces, and $\tau_k(1),\tau_k(2),...,\tau_k(t_k)$, $k=1,2,...,P$, be the centers of the output attribute subspaces. We equally divide $[-l_j,l_j]$ into s_j-1 subintervals, $j=1,2,...,N$, respectively, and then take the ends of the subintervals as the initial values of the subspaces' centers.

According to the known data, the unsupervised competitive learning method is used to determine each center of the attribute subspaces. Take the j-th component $x_j(1)$ of first input data as an example, the steps of competitive clustering are as follows:

(1) input the $x_j(1)$;

(2) compute the matching degrees: $d_{jr} = |x_j(1) - c_j(r)|$, $r = 1,2,...,s_j$;

(3) choose the best matching neuron, and adjust its linking weight as follows, such that it is more possible for the neuron to win, so we can cluster the data:

Suppose $|x_j(1) - c_j(r_1)| = \min\limits_r \{d_{jr}\}$, then $c_j(r)$ can be adjusted by

$$c_j(r) = c_j(r) + \Delta c_j(r), \quad r = 1,2,...,s_j,$$

where $\Delta c_j(r) = \begin{cases} \eta[x_j(1) - c_j(r)], & r = r_1, \\ 0, & r \neq r_1, \end{cases}$ and $\eta \in (0,1)$ is a learning coefficient. In the above process, if $c_j(r) < -l_j$, then $c_j(r) = -l_j$; and if $c_j(r) > l_j$, then $c_j(r) = l_j$.

And then input the other components, and repeat above three steps, until n data are all inputted. After the competitive learning steps, we have got the centers of input attribute subspaces. The centers of output attribute subspaces can be obtained by the same way.
In order to avoid oversize fluctuation in the competitive clustering process, we choose the alterable learning coefficient $\eta / \sqrt{i}$. It is bigger at the beginning and gradually decreases with the gain of training times i.

The value of widths are defined by

$$D_{jr} = \begin{cases} c_j(2) - c_j(1), & r = 1, \\ \max\big(c_j(r+1) - c_j(r), c_j(r) - c_j(r-1)\big), & 2 \leq r \leq s_j - 1, \quad j = 1,2,...,N, \\ c_j(s_j) - c_j(s_j - 1), & r = s_j, \end{cases}$$

$$\sigma_j(r) = D_{jr} \Big/ \sqrt{2s_j}, \; j = 1,2,...,N.$$

Since the competitive clustering is made singly for every input and output variable, and the programs are very simple, so the process can be completed at a short time even if there are many data and attribute variables.
Figure 1 shows the structure of the RBF network established. The first layer is input layer with N neurons. The second layer is matching layer with $\sum\limits_{j=1}^{N} s_j$ neurons. The third layer is reasoning layer with $\prod\limits_{j=1}^{N} s_j$ neurons each of which will corresponds to a rule. The fourth layer is output layer with P output neurons.

The weights linking input layer and matching layer are real numbers, which are the centers of attribute subspaces. The weights between matching layer and reasoning layer are 1. The weights between reasoning layer and output layer are denoted by ω_{lk}, $l = 1,2,...,s$, $k = 1,2,...,P$, $s = \prod_{j=1}^{N} s_j$, whose initial value can be obtained as follows:

Let $C_l = (c_1(r_{1l}), c_2(r_{2l}),..., c_N(r_{Nl}))$ be the center vector corresponding to the l-th neuron of reasoning layer. Let X_{i_k} be the proximate one to the center vector, namely $\|X_{i_k} - C_l\| = \min_i \|X_i - C_l\|$. Then correspondent Y_{i_k} with X_{i_k} can be used for the initial weight vector between the l-th neuron of reasoning layer and output layer, i.e. $(\omega_{l1}, \omega_{l2},..., \omega_{lP}) = (y_1(i_l), y_2(i_l),..., y_P(i_l))$.

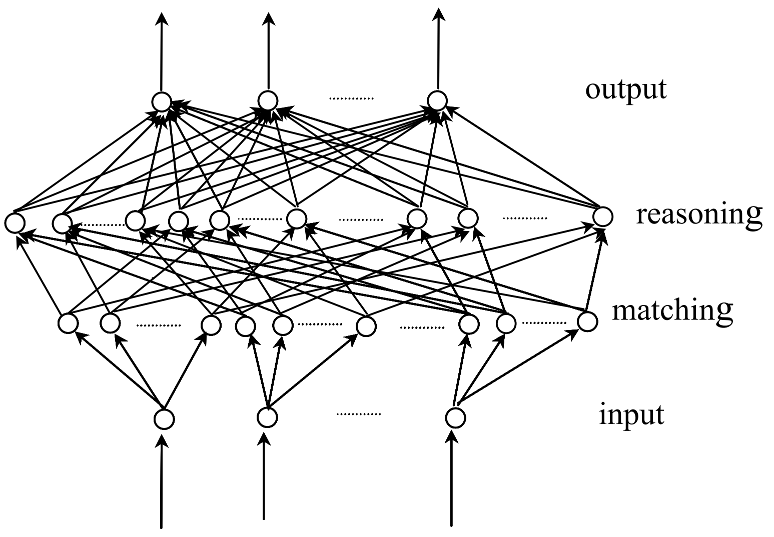

Figure 1. The construction of the RBF network

Input a value $X = (x_1, x_2,..., x_N)$. The outputs of the matching layer can be represented as

$$\mu_{jh_jl}(x_j) = R(x_j; c_j(r_{h_j}), \sigma_j(r_{h_j})), \quad j \in (1,2,...,N), h_j \in (1,2,...,s_j).$$

Since the RBF network is locally supported, so there are only a few of the neurons activated. The outputs of reasoning layer are

$$\mu_l(x_1, x_2,..., x_N) = \prod_{j=1}^{N_i} \mu_{jh_jl}(x_j), \quad l = 1,2,...,s,$$

where N_i is the number of activated neurons which are connected with the i-th neuron in reasoning layer.

If the RBF network is used for regression estimate, the outputs of the network are defined as

$$y_k^*(x_1,x_2,...,x_N)=\sum_{l=1}^{s}\omega_{lk}.\mu_l(x_1,x_2,...,x_N) \;\diagup\; \sum_{l=1}^{s}\mu_l(x_1,x_2,...,x_N),\quad k=1,2,...,P.$$

The error is defined by e = $\sum_{k=1}^{P} e_k$, where

$$e_k=\frac{1}{2}\sum_{i=1}^{n}[y_k(x_1(i),x_2(i),...,x_N(i))-y_k^*(x_1(i),x_2(i),...,x_N(i))]^2 .$$

If the RBF network is used for cluster, the outputs of the network are defined as $v_{ki}=R(x;\tau_{k_i}(r_{k_i}),\sigma_{k_i}(r_{k_i}))$, $k=1,2,...,P$, where

$$\left|y_k^*(x_1,x_2,...,x_N)-\tau_{k_i}(r_{k_i})\right| = \min_{1\le r\le t_k}|y_k^*(x_1,x_2,...x_N)-\tau_k(r)|.$$

When the training process ends, the extracted rules with the RBF network are:

"if x_1 is μ_{1j_hl}, x_2 is μ_{2j_hl} … and x_N is μ_{Nj_hl}, then y_1 is v_{1l}, y_2 is v_{2l}, …, y_P is v_{Pl}", $l=1,2,...,s$,

where μ_{1j_hl}, $\mu_{2j_hl},...,\mu_{Nj_hl}$, v_{1l}, $v_{2l},...,v_{Pl}$ are the RBF and they represent some input and output attribute subspaces respectively. In fact, we indicate the input and output attribute subspaces by fuzzy sets. So we get an important conclusion that the RBF network is equivalent with a system based on the fuzzy rules. If the attribute subspace is not fuzzy, the center of the RBF can be taken as the attribute value.

Adjusting the network parameters can reduce the error of the RBF networks. For example, the weight $\{\omega_{lk}\}$ can be tuned by the least square method.

3. The RBF neural network ensemble

Neural network ensemble can be used to improve the generalization ability of learning systems. We train a finite number of neural networks and then combine their results. But in general way, the neural network obtained by ensemble is more difficult to comprehend. According to the characteristics of the above-mentioned RBF networks, we employ the special ensemble way. Firstly, we train m RBF networks by the method in section 2 using different initial values of the network parameters and selecting input data from data sets at random. Secondly, we combine m results by simple average, i.e.

$$y_k^*(x_1, x_2, ..., x_N) = \frac{1}{m}\sum_{i=1}^{m} y_k^{*(i)}(x_1, x_2, ..., x_N) .$$

Since each neuron of reasoning layer corresponds to a rule, the extracted rules after neural network ensemble can be obtained by the relative majority principle for each premise and conclusion respectively.

4. Simplifying the rules

In the RBF networks established above, although the number of rules is large, the running speed is fast. But decreasing rules appropriately can make the law of the system clear. We combine and delete some rules by the following methods:

(1) For each center of the attribute subspaces, count up the number of data which are closest the center. If the proportion in total number is smaller then a defined small positive number, the center can be deleted. Now, the widths of the radial basis functions that are close to the center need to be adjusted. And the proportion can represent the degree of the importance of the rule.

(2) If the change of some attribute variable does not influence the conclusion of the rules, these rules can be combined.

5. An example

We use "BUPA liver disorders" as an example, which takes from UCI test database (http://kdd.ics.uci.edu/summary.data.type.html). There are 345 instances and 7 attributes for every instance. The first 5 variables are all blood tests, which are thought to be sensitive to liver disorders. The sixth variable is the number of half-pint equivalents of alcoholic beverages drunk per day. The seventh variable is the selector, which split data into two sets. We select 300 instances randomly as the learning sets, and the remaining data as the testing sets. We train 5 RBF networks using the input data selected from the learning sets at random, and then combine 5 output results by relative majority principle. Here, each of the 6 input variables has four attribute subspaces. They can be represented by the language variables such as low, medium, higher, and high. Using the method of the RBF network ensemble to extract if-then rules from data is rapid. And the result of the test shows that the method is effective. The proportion of correct classification is 97.8%.

We can simplify the rules by the method in section 4. Firstly, for each center of the attribute subspaces, count up the number of data which are closest to the center. Then make a list of the numbers as follows:

$$T=\begin{pmatrix} 1 & 16 & 259 & 24 \\ 10 & 218 & 67 & 5 \\ 181 & 110 & 7 & 2 \\ 41 & 241 & 15 & 3 \\ 217 & 69 & 8 & 6 \\ 167 & 130 & 2 & 1 \end{pmatrix},$$

where each row corresponds with an attribute variable. If we delete the corresponding centers while the number is smaller then 6, the extracted rules can be reduced to 15.8% of the total rules. If we delete the centers while the number is smaller then 9, the extracted rules can be reduced to 5.3% of the total rules. Under the latter situation, we adjust the widths of the radial basis functions which are close to the centers. For new RBF network the proportion of correct classification is 95.6%.

Our conclusion is that the RBF network may be equivalent to a system based on the fuzzy rules, and using above-mentioned method, the RBF neural network ensemble can be applied to extracting the rules from data rapidly.

References

[1] D.S.Broomhead and D.Lowe, Multivariable functional interpolation and adaptive networks, Complex System*s*, 2(1988), 321-355.

[2] W.A. Light, Some aspects of radial basis function approximation, Approx. Theory, Spline Functions Applic., 356(1992), 163-190.

[3] M.J.D. Powell, Radial Basis Functions in 1990, Adv. Num. Anal., 2(1992), 105-210.

[4] C.H. Li, N.N. Zheng, and Y.P. Zhang, Radial basis function network for elimination of noise from image, Journal of Xi’an Jiaotong University, 3(1999), 10-14.

[5] Xi Lin and Yee Leung, Fast extracting fuzzy rules and system simulation based on radial basis function neural networks, The Journal of Fuzzy Mathematics, 3(2001), 745-751.

[6] L.K. Hansen and P.Salamon, Neural network ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(1990), 993-1001.

[7] Y.Freund and R.E. Schapire, A decision theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, 1(1997), 119-139.

[8] H.Schwenk and Y.bengio, Boosting neural networks, Neural Computation, 8(2000), 1869-1887.