一种新型的神经网络构造方法 RCBNN*)

谢振华 李 宁 商 琳 陈兆乾 陈世福
（南京大学计算机软件新技术国家重点实验室 南京 210093）

摘 要 一个好的神经网络结构可以大大提高它的处理能力和收敛速度，所以神经网络的构造方法一直是人们研究的热点问题。本文利用粗集理论的数据分析能力和决策树对数值属性的分割能力，提出一种基于粗集与决策树的新型神经网络构造方法 RCBNN。经试验表明，使用该方法构造的神经网络，具有易于构造、可理解性好、收敛速度快且构造的网络规模较小的特点。

关键词 神经网络，粗集，决策树

RCBNN: A New Constructing Method for Neural Network

XIE Zhen-Hua LI Ning SHANG Ling CHEN Zhao-Qian CHEN Shi-Fu
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract A good structure can greatly improve the power and convergent speed of neural network. According this, the constructing method is always a hotspot in neural network research. This paper proposes a constructing method for neural network based on Rough Set and Decision Tree. Rough Set has good capability of data analysis; Decision Tree is good at segmentation of continues-valued attributes. This method makes good use of these advantages. Experimental results show that the neural network designed by this method is easy constructible, good understandable, fast convergent, and small dimensional.

Keywords Neural network, Rough set, Decision tree

1 引言

波兰科学家 Pawlak 提出的粗集(Rough set)[1]是描述不完整、不精确和含噪声数据的有力工具，其主要特点是不需要任何与数据有关的先验和附加知识。从本质上看，它反映了认知过程在非确定性、非模型化信息处理方面的机制和特点，从而成为一种有效的非单调推理工具。粗糙集模型是一种结构化的、非数值化的信息处理方法，处理的是数据的符号描述，以属性、语义决策规则等形式构造知识表达，其主要特点是不需要任何与数据有关的先验和附加知识，这一点明显区别于模糊集理论中的隶属函数等方法。目前，粗糙集主要应用于属性约简、规则生成及预测等几个方面。但粗集理论容错能力与推广能力相对较弱，适于处理离散的非确定数据，对于连续数据的处理能力有限[2]，且因对数值型属性作为离散值看待，规则生成后的规模大；决策树学习是应用最广的归纳推理算法之一，对噪声数据有很好的健壮性，其学习速度快，规则学习能力强，本文所采用的 C4.5 决策树算法[3]能通过对连续数据的分割，有效地缩小规则的规模，但决策树容易产生过拟合现象，虽然通过裁减可一定程度避免过拟合，但精度的损失较大；人工神经网络最具吸引力的是其数值逼近能力，并能够处理定量的、数值化的信息。人工神经网络关注的是建模和学习过程中获得的数据的细节信息，具有较强的自组织能力、容错能力和推广能力，但不能优选条件属性，而粗集的属性约简能力恰好能弥补这一缺陷。已有研究者成功地将粗集理论与神经网络结合，构造出新型的神经网络结构[4,5]。本文利用粗集与决策树理论的优点，并结合人工神经网络的特点提出了一种基于粗集与决策树的神经网络构造方法——RCBNN，该方法易于构造、可理解性好、计算简单、收敛速度快且构造的神经网络规模较小，通过实验表明，该方法能够得到较满意的结果。

2 粗集理论与决策树算法

给定一个有穷对象集 U、有穷属性集 A、各属性值的集合 V 以及信息函数 f 构成一个信息系统，表示为(U,A,V,f)，

*)基金项目：国家自然科学基金(60273033)；江苏省自然科学基金(BK2003067)。**谢振华** 硕士研究生，主要研究方向：神经计算。**李 宁** 硕士，讲师，主要研究方向：机器学习。**商 琳** 博士研究生，主要研究方向：人工智能、数据挖掘。**陈兆乾** 教授，博士生导师，主要研究领域：人工智能、机器学习。**陈世福** 教授，博士生导师，主要研究领域：人工智能、机器学习。

及 Vague 集属性信息系统的属性约简，其结果将另文发表。

参 考 文 献

1 Pawlak Z. Rough set approach to multi-attribute decision analysis. European Journal of Operational Research, 1994, 72: 443~459
2 Pawlak Z. Rough set theory and its applications to data analysis. Cybernetics and System, 1998, 29(27): 661~688
3 Chen M S, Han J, Yu P S. Data Mining: An overview from a database perspective. IEEE Transaction on Knowledge and Data Engineering, 1996, 8(6): 866~883
4 Hu XiaoHua, Cercone N. Learning in relational database: a rough set approach. Computational Intelligence, 1995, 11(2): 323~337
5 Grzymala-Busse J W, Hu M. A comparison of several approaches to missing attribute values in data mining. In: Proc. of the 2nd Int'l. Conf. on Rough Sets and Current Trends in Computering. Berlin: Springer Verlag, 2000. 378~385
6 王国胤. Rough 集理论与知识获取. 西安：西安交通大学出版社，2001
7 曾黄麟. 粗集理论及其应用. 重庆：重庆大学出版社，1996

用 V_a 表示属性 a 的值构成的集合，称为 a 的域，$V=\bigcup V_a$，信息函数 $f:U\times A\rightarrow V$ 为对象的属性到其值的映射。

定义 1 对属性集 A 的任意子集 B，$I(B)=\{(x,y)\mid f(x,a)=f(y,a),\forall a\in B\}$ 定义了 U 上的二值关系，称为不可分辨关系。用此关系可将 U 分割为等价类族，任意元素 x 所在的等价类记为 $B(x)$。

定义 2 三元组 $AS=(U,I,v)$ 为广义逼近空间，其中 $I:U\rightarrow P(U)$ 为非确定函数，一般定义 I 就是元素 x 到等价类 $B(x)$ 的映射，$v:P(U)\times P(U)\rightarrow[0,1]$ 为包含函数，一般的定义如下：

$$v(X,Y)=\begin{cases}\dfrac{|X\cap Y|}{|X|},X\neq\varnothing\\ v(X,Y)=1,X=\varnothing\end{cases}$$

显然，$v(x,y)$ 表示 x 包含于 y 的程度。AS 空间中集合 U 的任意子集 X 的下逼近和上逼近可一般地表示为如下的式子：

$$\underline{AS}(X)=\{x\in U:v(I(x),X)=1\}$$

$$\overline{AS}(X)=\{x\in U:v(I(x),X)>0\}$$

定义 3 令 $A=C\cup D$，$C\cap D=\varnothing$，C 为条件属性集，D 为决策属性集，如果 $\exists C'\subset C$ 使得 $\gamma(C',D)=\gamma(C,D)$，并且 $\forall C''\subset C'$，有 $\gamma(C'',D)\neq\gamma(C',D)$，则称 C' 为 C 的一个 D 约简（D-reduct），其中

$$\gamma(C,D)=\frac{|POS_C(D)|}{|U|}$$

$$POS_C(D)=\bigcup_{X\in U/D}\underline{C}(X)$$

Rosetta 工具[6]提供了两种好的约简算法：Johson 约简算法和遗传约简算法，本文使用 Johson 算法来求约简，试验也证明使用 Johson 算法得到的属性集作为神经网络的输入属性，效果要好于其他约简算法。

决策树是一种逼近离散值目标函数的方法，在这种方法中学习到的函数被表示为一颗决策树，或多个 IF-THEN 的规则，以提高可读性。决策树通过把实例从根节点排列到某个叶子节点来分类实例，叶子节点即为实例所属的分类。树上的每一个节点说明了对实例的某个属性的测试，并且该节点的每一个后继分支对应于该属性的一个可能值[7]。目前流行的决策树算法是 C4.5 算法，通过该算法可实现对数值类型的属性的分割，即用不等式对属性进行测试，正因为这一特点，使决策树生成的规则规模大大缩小。

3 RCBNN 构造方法

3.1 RCBNN 结构的设计方法

人工神经网络通常关注的是神经元间连接的固定拓扑结构，如前向型、Hopfield 网络等。一般地，神经元转移和激活函数使用 S 型函数，常用的如式(1)所示。

$$f(x)=\frac{1}{1+e^{-\beta x}} \tag{1}$$

x 表示输入连接的权值和输入值的某种映射结果 $g(w,X)$，β 参数称为增益（$gain$）。

粗集约简后生成的规则的一般形式为 IF A is(isn't)a AND B is(isn't)b AND…AND Z is(isn't)z THEN Class is C，因此基于粗集的 KBNN 的构造基本分为四层：输入层、前件层、规则层和输出层。前件层表示 A is a 型的谓词，它根据输入层的输入值确定输出 0 还是 1，前件层到规则层是“与”连接，规则层到输出层是“或”连接。Geoffrey 等人介绍了 KBNN 的初始连接权和阈值的设置方法[8]。除输入层与前件层间不用补全连接外，将其他的层间连接补全后就形成分层全连接前馈网络，这些补足的连接的初始权值为 0 或为 -1 到 1 之间的随机数。

然而对数值类型的属性，通过决策树生成的规则的一般形式是 IF $A>a$ $\wedge\cdots\wedge$ $B\leqslant b$ THEN Class is C，这种类型的规则就不适合按上述方式构造神经网络。Nguyen Hong Son 等人[9]提出了一种针对具有连续值属性数据的神经网络设计方法，其步骤是先构造超平面，再通过超平面的合取形成的规则构造神经网络来进行分类，其本质就是将前件层改为超平面层。该方法的缺点是超平面的确定较难，本文的 RCBNN 方法使用 C4.5 确定初始超平面，方法可靠而简单，在构造神经网络方面也对其进行了一些修改。决策树生成的规则的前件可改写为 $(A-a)>0$ $\wedge\cdots\wedge$ $(B-b)\leqslant 0$，显然，其中的 $(A-a)$ 为一简单超平面表达式，此即为初始超平面。然后以生成的规则为基础，构造四层前向分层全连接神经网络，其结构如图 1 所示，其中实线表示初始连接权非 0，虚线表示初始连接权为 0。

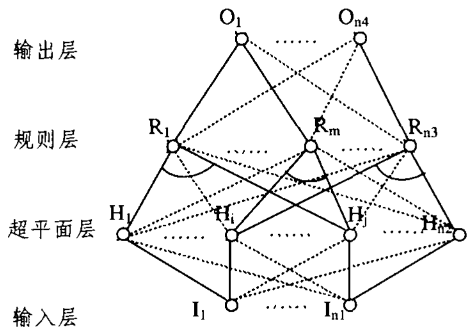

图 1 RCBNN 神经网络结构图

设所有前件合取式子项的超平面集合为 H，第一隐层各神经元表示各简单超平面，第 i 个神经元采用式(2)所示的激活函数。

$$\frac{2}{1+e^{-a_i x}}-1 \tag{2}$$

输入层与第一隐层的神经元间的初始连接权值就是对应超平面在该属性上的系数，因此非 1 即 0。该隐层神经元的初始阈值为对应简单超平面的常数项。第二隐层神经元表示规则，该层各神经元的激活函数使用式(1)提供的函数，该隐层所有神经元的阈值初始化为 0。如果第 m 条规则表示为：

$$((H_i(u)\leqslant 0)\wedge(H_j(u)>0))\Rightarrow D(u)=v_m$$

则第一隐层与第二隐层的第 m 个神经元间的初始连接权以如下方式设置：

$$w_{mv}=\begin{cases}-1/2 \text{ for } n=i\\ 1/2 \text{ for } n=j\\ 0 \text{ otherwise}\end{cases}$$

输出层神经元则对应于决策类别，该层各神经元的激活函数也使用式(1)提供的函数，输出层的所有神经元阈值为 0，阈值在学习过程中不进行调整。第二隐层与输出层神经元间的初始连接权值为 1，表示对应的规则输出对应的类别，否则取为 0。

3.2 RCBNN 的处理算法

基于以上给出的神经网络结构，本文提出的 RCBNN 处理算法的步骤如下：

```
Input: Dataset ds, Max cycle number max_cycle, Max unchanged number max_unchg;
Output: RCBNN nn;
Step 1. SplitDataSet(ds, trains, tests);
        //将数据集随机分割成训练集和测试集
Step 2. attrs = GetReductAttrs(trains);
        //使用粗集约简算法求约简属性集
Step 3. n_trains = GetDataSetFromAttrs(trains, attrs);
        n_tests = GetDataSetFromAttrs(tests, attrs);
        //根据 Step 2 中求出的约简属性集，删除训练集与测试集中不在其中的属性，并去除多余数据项，从而构造出新的训练集与新的测试集
Step 4. des_prec = RCBNNTree(n_trains, rules);
        //使用有裁减的 C4.5 算法生成规则
Step 5. nn = ConstructRCBNN(rules);
        //利用生成的规则构造如图 1 所示的网络
Step 6. cycle = 0;
        do
        {
          prec = TrainRCBNN(nn, trains);
        } while(prec >= des_prec ||
            ++cycle < max_cycle);
        //训练网络到原有规则精度
Step 7. old_prec = prec;
        tmp_nn = nn;
        int unchang = 0;
        while(cycle < max_cycle ||
            unchang < max_unchg)
        {
            prec = TrainRCBNN(tmp_nn, trains);
            if(prec >= old_prec)
            {
              old_prec = prec;
              unchang = 0;
              nn = tmp_nn;
            }
            else unchang++;
        }
        //训练网络到精度不再显著下降为止
```

RCBNN 采用传统的 BP 算法进行训练，输出类别采用“胜者为王”的方式选择，输出值最大的神经元对应的类为输出类。

4 试验及分析

4.1 实验环境

用本文提出的 RCBNN 方法在三个 UCI 数据集——BCW(Breast-Cancer-Wisconsin)、Australian、Iris——上进行了试验，实验中首先对数据集采用随机抽样将其分成训练集和测试集，然后对需要离散化的数据集执行离散化过程：BCW 的数据已经是离散化过的数据；Australian 除了第 1、4、5、6、8、9、11、12 个属性已经是离散化过的属性，不再进行离散化处理外，训练集上的其他属性使用 Rosetta 工具提供的 Orthogonal Scale 方法进行离散化，根据所得的分割点再将测试集离散化；Iris 数据属性和数据项数很少，就没有使用离散化处理。数据集及其项数如表 1 所示。

表 1 数据集及其项数

数据集	BCW	Australian	Iris
训练集项数	478	483	105
测试集项数	205	207	45
属性数	9	14	4
约简后属性数	4	6	3
约简后训练集项数	225	425	99

4.2 RCBNN 与 KBNN 规则数的比较

在这三个数据集的训练集上执行属性约简后生成的规则规模很大，规则数就是约简后训练集项数，如果再使用值约简[10]，则规则数会大大缩小，但相比 RCBNN 算法所获得的规则数而言，仍然十分庞大，表 2 给出了采用 RCBNN 算法获得的规则数与 KBNN 的规则数的比较，显然采用 RCBNN 方法构造的神经网络规模比采用粗集约简后生成的规则构造的 KBNN 小很多。

表 2 RCBNN 与 KBNN 规则数比较

数据集	RCBNN 规则数	KBNN 规则数	RCBNN 裁减因子
BCW	5	68	0.001
Australian	9	269	0.1
Iris	3	56	0.01

4.3 RCBNN 与 C4.5 的比较

表 3 是 RCBNN 与 C4.5 算法的训练精度与测试精度实验结果的比较，左边是 RCBNN 的实验结果，试验中 RCBNN 的激活函数的增益使用相同值，设为 β，并且连接权学习率和阈值学习率都设为 0.25；右边是 C4.5 有裁减和无裁减算法训练精度与测试精度的实验结果，从中看出无裁减规则集泛化能力较差，经裁减的规则集泛化能力提高，但训练精度下降。对比 RCBNN 与 C4.5 的实验结果可看出，RCBNN 训练与测试精度基本都比裁减的 C4.5 决策树的精度高，其原因一方面是调整了已获得的超平面，另一方面是调整了与规则相关的连接权。

表 3 RCBNN 与 C4.5 算法试验结果比较（训练精度的计算使用约简后无重复记录的训练集）

数据集	RCBNN 实验结果		C4.5 实验结果			
			无裁减		有裁减	
	训练精度	测试精度	训练精度	测试精度	训练精度	测试精度
BCW	96.45%	97.56%	97.33%	96.59%	94.22%	96.59%
Australian	84.47%	81.64%	87.71%	75.36%	77.65%	79.23%
Iris	95.96%	95.56%	95.96%	93.33%	93.94%	97.78%

4.4 RCBNN 与三层前馈 BP 的比较

表 4 是使用三层前馈 BP 神经网络在约简后的训练集上训练的实验结果，为进行相互比较，其隐层神经元数是相应 RCBNN 的隐层神经元数的和，并且连接权学习率和阈值学习率均设为 0.25，激活函数的增益也均取相同值，且与相应的 RCBNN 的相同，也采用“胜者为王”方式选择输出类。由于三层前馈 BP 神经网络的初始连接权的取值是随机的，因此要经过多轮训练才能得到一个比较好的训练精度与测试精度的神经网络，即使这样，从表中可以看出其训练次数基本远远多于 RCBNN 的训练次数，且精度也基本无法达到 RCBNN 方法的精度，因此 RCBNN 具有较快的收敛速度和较高的精度。

表 4 RCBNN 与三层前馈 BP 神经网络试验结果比较

数据集	β	RCBNN 实验结果						BP 网络实验结果			
		H1*	H2*	训练次数	训练精度	测试精度	H*	训练次数	训练精度	测试精度
BCW	1.0	4	5	332	96.45%	97.56%	9	1348	95.56%	97.07%
Australian	0.9	6	9	327	84.47%	81.64%	15	345	78.82%	80.19%
Iris	1.0	2	3	278	95.96%	95.56%	5	1136	93.94%	95.56%

*注:H1 表示第一隐层神经元数,H2 表示第二隐层神经元数,H 表示隐层神经元数

结论 本文提出的基于粗集与决策树规则生成的神经网络构造方法 RCBNN,充分利用了粗集的数值处理能力、决策树的规则生成能力和神经网络准确的逼近收敛能力,互相扬长避短。试验表明,该方法取得了较好的综合效果。该方法利用粗集的约简方法减少了数据集的数据量,使训练量减少;利用决策树的规则生成能力产生规模很小的规则集,弥补了粗集生成规则规模很大的缺陷;又利用神经网络的准确的逼近能力,进一步提高了测试精度,并一定程度上弥补了决策树算法因裁减而丢失的训练精度。用本文所述方法构造神经网络具有规模小、易于构造、学习速度快、泛化能力较好等优点。

参考文献

1 Pawlak Z. Rough sets. International Journal of Information and Computer Science, 1982, 11:341~256
2 Duntsch I, Gediga G. Simple Data Filtering in Rough Set Systems. International Journal of Approximate Reasoning, 1998, 18 (1-2):93~106
3 Quinlan J R. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993
4 Wu Zhaocong, Li Deren. Neural Network Based on Rough Sets and Its Application to Remote Sensing Image Classification. Geo-Spatial Information Science, 2002, 5(2):17~21
5 Lingras R. Fuzzy-rough and Rough-fuzzy Serial Combinations in Neurocomputing. Neurocomputing, 2001, 36:29~44
6 Ohrn A, Komorowski J. Rosetta-A Rough Set Toolkit for Analysis of Data. In: Proc. Third Intl. Joint Conf. on Information Sciences, Durham, NC, USA, 1997, 3:403~407
7 Mitchell T M. 曾华军,张银奎,等译. 机器学习. 北京:机械工业出版社, 2003
8 Towell G G, Shavlik J W. Knowledge-based Artificial Neural Networks. Artificial Intelligence, 1994, 70:119~165
9 Hong Son N, Szczuka M S, S'lezak. Neural Networks Design: Rough Set Approach to Continuous Data. In: Komorowski J, Zytkow J, eds. The First European Symposium on Principle of Data Mining and Knowledge Discovery (PKDD'97), Trondheim, Norway, Lecture Notes in Artificial Intelligence 1263. Berlin: Springer-Verlag, June 1997. 359~266
10 王国胤. Rough 集理论与知识获取. 西安:西安交通大学出版社, 2001

(上接第 146 页)

竭集,这时有,

$$Bel(S)\geqslant 0, Bel(\Theta)\leqslant Bel(\Theta\cup S), Bel(\Theta\cup S)=1$$

由此得到一个封闭世界,必然事件为 $\Theta\cup S$,因而指派给 $A(A\subseteq\Theta)$ 的信念度实际上是指派给 $A\cup S$。记识别框架 Θ 的不可能事件为 $\phi\Theta$,那么指派给 $\phi\Theta$ 的信念度就是指派给 $\phi\Theta\cup S$,这样在开放世界预设下应用 Dempster 规则无需正则化。依这一观点,在这个悖论中实际情况是:嫌疑人名单可能是不完备的。因此基本概率指派应该为:

$$m_1(\{a\}\cup S)=0.99, m_1(\{b\}\cup S)=0.01$$

$$m_2(\{b\}\cup S)=0.01, m_2(\{c\}\cup S)=0.99$$

那么由 Dempster 组合规则得:

$$m_{12}(\phi_\Theta\cup S)\approx 0.99, m_{12}(\{b\}\cup S)\approx 0.01.$$

这里的 S 为已有嫌疑人的延长清单。组合基本概率指派表明,凶手不在嫌疑人名单之中的可能性大约有 99%,从而前文中的悖论得以消除。

不可否认,如果抛弃识别框架的封闭性这一预设,上述解决方法具有一定的启发性及合理性。但是,倘若嫌疑人的名单是完备的,这一悖论仍然没有得到解决;而本文在这个预设下分析了悖论所产生的原因。

基于 D-S 理论的随机译码器解释,我们可以利用概率表达该理论的基本概念和组合规则。撇开随机译码器解释,从定义的形式上看,基本概率指派(1)和(2)是合法的,并且可以应用 Dempster 组合规则来组合它们。但是在随机译码器解释下,由于这些基本概率指派所对应的证据 E_1 和 E_2 是不相容的,这使得组合规则表达式(8)失去意义。此时,倘若坚持应用组合规则来组合它们将导致上述悖论。这就是产生这一悖论的原因。

参考文献

1 Shafer G. A mathematical theory of evidence. Princeton University Press, 1976
2 Dempster A. A generalization of Bayesian inference. J. R. Statist. Soc. 1967, 30:204~247
3 Walley P. Measures of Uncertainty in expert systems. Artificial Intelligence, 1996. 1~58
4 Hsia Y, Shenoy P. An evidential language for expert systems. In: Ras Z. eds., Mathodologies for Intelligent systems 4, North-Holland, New York, 1989. 9~16
5 Yager R, Kacprzyk J, Fedrizzi M. Advances in the Dempster-Shafer Theory of Evidence. John Wiley & Sons, Inc., 1994
6 Gordon J, Shortliff E. A method for managing evidential reasoning in a hierarchial hypothesis space. Artificial intelligence, 1985, 26:323~358
7 Bordley R. Reformulating decision theory using fuzzy set theory and Shafer's theory of evidence. Fuzzy Sets and Systems, 2003 (139):243~266
8 Smets P. Belief Functions. In: Non-Standard Logic for Automated Reasoning. Academic Press Limited, 1988. 253~289
9 Shafer G, Tversky A. Languages and Designs for Probability Judgment. Cognitive Science, 1985, 9:309~319
10 Shafer G, Srivastava R. The Bayesian and Belief-Function Formalisms: A General Perspective for Auditing. In: Reading in Uncertain Reasoning Morgan Kanfmann, San Mateo California, 1990. 480~520
11 Shafer G. Constructive Probability. 1Synthese, 1981(48):1~60