

基于知识的神经网络产生式规则的获取方法研究

刘道华^{1,2}, 原思聪¹, 江祥奎¹, 陆地¹, 王 瑞³

(1. 西安建筑科技大学机电学院, 陕西 西安 710055; 2. 信阳师范学院计算机科学系, 河南 信阳 464000;

3. 西安建筑科技大学信息与控制工程学院, 陕西 西安 710055)

摘 要:将连续数据离散化并将已有知识规则的依赖度作为神经网络的初始权值, 构建新的网络结构并对其动态训练, 给出其具体网络训练算法。从训练后的网络权值中利用正确的分类及该网络结构具有的性质, 从而给出其具体的规则抽取算法, 并将抽取的具有冗余性的产生式规则利用粗集理论进一步对其精化处理, 最后得出最简化的产生式知识规则。充分结合神经网络及粗集理论的优点, 探索两者的有机结合无疑对智能信息处理系统的研究具有重要的现实意义。

关键词:神经网络; 产生式规则; 知识获取; 粗集约简

中图分类号: TP182

文献标识码: A

文章编号: 1006-7930(2007)03-0423-06

知识获取问题一直是人工智能研究的难点, 当前在知识获取方面也有许多人提出不同的知识获取方法, 如采用广度优先抽取规则的 Subset 算法、Towell 等人提出的 MofN 方法、RF(Rule from facts)算法、基于 VI 分析(Validity-Interval Analysis)的规则抽取算法和 KT 算法等。这些知识规则的获取算法对获取不同知识起着重要作用。其后, 由于神经网络在各个领域广泛和深入的应用, 人们提出了基于 KBANN(knowledge-based artificial neural network)的规则提取算法及其各种改进的网络模型算法, 但这些算法基本上都是从训练样本中通过网络训练后, 从连接权中提取规则, 却很少利用专家已有的知识经验获取新的知识^[1-3]。因此本文充分利用已有的知识规则, 并将规则的前提各依赖度作为网络初始训练的连接权, 即构建了神经规则作为基本的神经元。然而利用神经网络获取知识, 其原理基本上都是利用权值分类来获取最后规则, 因此其获得的规则具有很大的冗余性, 很有必要利用其它技术进一步对知识进行精化, 而粗糙集理论可以描绘知识表达中不同属性的重要性, 通过约简去掉冗余知识, 以达到知识表达空间的约简, 最后从约简后的元组中提取精简规则。本文充分利用粗集理论将从神经网络中获取的规则进一步精化, 即充分结合神经网络及粗集理论两者的优点, 探索它们之间的有机结合。该方法无疑对智能信息系统的研究具有十分重要的现实意义。

1 数据准备

现实生活中许多数据形式多样, 有离散的、有连续的, 而作为神经网络的输入必须将连续的数据离散化, 才能统一作为网络的输入。这种离散化过程就是对数值型数据划分区间, 用相应的二进制数据对应相应的数值区间, 而相应的数据区间对应相应的语言描述事实。这样数据经离散化处理并经网络训练后, 从网络结构中抽取知识规则的数据才能对应上相应的语言描述事实。而常采用的二值化处理数据的方法^[4], 首先要设计好将连续数据划分成多少个不同的区间集。假设第 i 个连续性数据 x_i 的值为 V , 其上限为 V_U , 下限为 V_L , 若将其分成 j 个区间集, 划分方法为:

若 $V_L \leq V \leq V_L + (V_U - V_L)/j$, 则该区间以 A_{i1} 表示;

若 $V_L + (V_U - V_L)/j \leq V \leq V_L + 2(V_U - V_L)/j$, 则该区间以 A_{i2} 表示;

.....

收稿日期: 2006-07-10

基金项目: 陕西省教育厅产业化项目(04JC21)

作者简介: 刘道华(1974-), 男, 河南信阳人, 讲师, 博士研究生, 主要从事智能系统及软件开发。

若 $V \geq V_L + (j-1)(V_U - V_L)/j$, 则该区间以 A_{ij} 表示.

其中 $A_{i1}, A_{i2}, \dots, A_{ij}$ 代表该属性类的所有符号代表值, 在作为神经网络输入时, 如果有该分类的符号代表值, 则该值以二进制值 1 表示, 否则以二进制值 0 表示.

对于本来就是离散属性值和标称属性值的输入 x_j , 假如有 t 个不同值, 则用 t 个不同的符号来代表, 即用 $A_{j1}, A_{j2}, \dots, A_{jt}$ 表示. 且在参入神经网络运算时有该分类的某一符号代表值, 则该值以二进制值 1 表示, 否则以二进制值 0 表示.

经过上述处理后的数据才能作为神经网络的规范化输入. 且便于规则提取后的值对应上相应的语言分类值.

2 网络构成及网络训练

2.1 网络构成

为了充分利用现有的各种产生式规则, 其规则形式如图 1 所示. 将各个已有的规则前件作为神经规则的前件, 各规则前件的依赖度 (cf_{ji}) 作为网络初始权值, 而初始规则的后件作为网络的初始输出, 这样的神经元叫神经规则^[5], 如图 2 所示. 对于问题域中没有对应的规则前件的权值则随机确定, 同时对没有对应的规则的输出后件将根据输入值事先确定.

在充分利用神经规则的基础上, 系统结构采用 4 层结构的神经网络. 第一层为输入层、第二层为语言层、第三层为神经规则

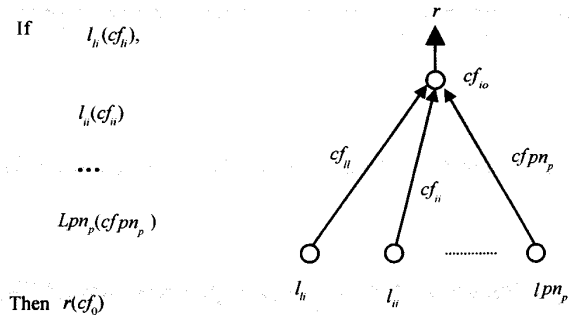


图 1 产生式规则形式

图 2 神经规划结构

Fig. 1 The form of IF/THEN rules Fig. 2 The structure of neural rule

层、第四层为输出层. 其中第一层激活函数为分段线性函数, 其作用是对输入的各类数据(连续的、离散的或标称的)作区域判断, 即辨别 A_{ni} 的输出 l_{ni} 是近 0 值还是近 1 值, 其中 i 表示第 i 个输入所对应的分类, n_i 表示该分类个数(各分类的 n_i 可能不等, 即分类数不等); 第二层神经元的激活函数为普通线性函数, 即当 l_{ni} 的输入值为近 1 值时, 其激活函数 $\sigma(x) = 1$, l_{ni} 的输入值为近 0 值时, 其激活函数 $\sigma(x) = 0$; 第三层神经规则的神经元, 其激活函数常采用高斯核函数那样的辐射状作用函数构成, 这样在网络学习时起动第 j 个规则强度时可按前件最大的权值获得 w_j , 而 $w_j = \max\{cf_{j11}, \dots, cf_{jin_i}, \dots, cf_{jpn_p}, w'_{j11}, \dots, w'_{jin_i}, \dots, w'_{jpn_p}\}$, 其中 w'_{jin_i} 为初始数据分类没有对应的规则前件项的随机权值, 并限定其初始随机值为近 0 值, 以使网络训练时有前件依赖度 cf_{jin_i} 的项优先被选择上, cf_{jin_i} 表示第 j 个神经规则与第 i 个语言属性的第 n_i 个分类间的连接权, 而且在该神经规则神经元中又增加反馈比较阶段的权值 w_{mj} , 其目的是用于在学习时对初始样本的比较, 以便判断是否增加新的神经规则神经元节点. 整个系统最后

输出可用每个规则的加权和 $f(x) = \frac{\sum_{i=1}^r w_i u_i}{\sum_{i=1}^r w_i}$ 表示, 其中 u_i 为神经规则层的高斯函数输出, r 是神经规则

层神经元个数. 而且每个结论对应一个神经规则层神经元节点, 其节点激活函数都选择具有高斯函数相同的形式, 即:

$$\theta(l_{ni}) = \exp\left[-\frac{(l_{ni} - C_{r_j})^2}{\sigma_j^2}\right] \quad (1)$$

其中: C_{r_j} 、 σ_j 分别为相应接收域的中心和宽度. 第四层也采用普通的线性函数作为激活函数, 其作用仅将神经规则层的输出传给输出层. 其具体的网络构成如图 3 所示.

2.2 网络的训练过程

采用动态构造和训练神经网络神经规则层节点的方法, 即对任一个类 $C_j (j \in 1, \dots, m)$, 先增加一个规

则层节点 r_1 , 然后用全部输入样本训练网络, 若网络无法收敛, 再增加一个规则层节点 r_2 , 重复这一过程, 直至收敛为止. 收敛后, 在输出层增加一个输出节点, 把属于类 C_j 的所有规则层节点联结到该输出节点上, 其具体算法如下:

- ① 初始化网络;
- ② 输入新样本, 并对样本进行初始分区间处理;
- ③ 自下而上的运算, 竞争选择出优胜者, 即在前馈连接权 cf_{jin_i} 及 w'_{jin_i} 中寻找最大输出值的节点号作为获胜者;

- ④ 警戒性检验(即自上而下的比较阶段);
计算 $R = N_H(CF, W) = 1 - M_H(CF, W)$ 的 Hamming 贴近度, 若 $R > \rho$ (ρ 为事先设定的贴近度警戒值) 成立, 即通过警戒检验, 转 ⑤;

若获胜节点没有通过警戒测试, 则屏蔽该获胜节点(即此节点不参加下一轮识别) 返回 ③ 寻找新的获胜节点 i , 假如在所有的输出节点中找不到通过警戒测试的获胜节点时, 则累计这种要求(开辟新类的样本数加一) 转 ⑥;

- ⑤ 修改权值;

取, $cf_{jin_i}(k+1) = \frac{w_{in_j}(k) \cdot l'_{in_i}}{c + \sum w_{in_j}(k) \cdot l'_{in_i}}$, $W_{in_j}(k+1) = w_{in_j}(k) * L'_k$ ($i = 1, 2, 3, \dots, p$), 其中 c 为

常数, 一般取 0.5; $L'_k = [l'_{1n_1}, l'_{2n_2}, l'_{3n_3}, \dots, l'_{pn_p}]^T$ 为输入模式, $W_{in_j} = [W_{1n_1j}, W_{2n_2j}, \dots, W_{pn_pj}]^T$ ($j = 1, 2, 3, \dots, m$) 为学习过的输入模式存储的反馈连接权矢量. 也就是前馈连接权矢量 $cf_{jin_i} = [cf_{j1n_1}, cf_{j2n_2}, \dots, cf_{jp_n_p}]^T$ 和反馈连接权矢量 $W_{in_j} = [W_{1n_1j}, W_{2n_2j}, \dots, W_{pn_pj}]^T$ 向着与 L'_k 更接近的方向调整;

- ⑥ 判断本次训练所提供的样本是否学习一遍;

如果未完成样本学习转第 ② 步, 输入下一个样本. 如果完成, 检查是否开辟新节点, 若要求开辟新节点的样本数 $k > r = 5$ (为了避免落入新节点的样本太少, 可事先规定落入新节点的样本数不小于 r), 此时 $m = m + 1$, 并增加一个输出节点, 并初始化新节点的权值, 从而完成创建一个新节点的工作; 否则不开辟新节点, 转入下一轮样本学习.

其中在增加新节点时, 首先把未通过识别的样本权值 cf_{jin_i} 或 w'_{jin_i} 用数组分别存储, 在判断是否增加新节点时, 先分类统计 cf_{jin_i} 及 w'_{jin_i} , 把相近值的 cf_{jin_i} 及 w'_{jin_i} 作为一类, 并统计属此类的数组元素个数 u , 并判别 $u > r$ 成立否. 对成立的增加此节点, 然后清除本类的所有数组元素值. 对不满足 $u > r$ 的仍保存数组元素值, 为下一次样本训练时作为累加基数. 这样能累计以往的训练样本, 同时本网络模型就可以在不同时期内对其进行训练, 而且下次训练时能统计以前训练时与已有样本不匹配的样本, 从而决定是否增加新节点, 以至不浪费以前的样本信息, 这样使网络更具有健壮性.

3 规则提取

从训练好的神经网络中提取规则, 其具体的步骤是: 分类, 即将连接权分成若干等价类; 平均, 即将每个等价类中的权值平均化; 删除, 即删除对神经元的作用较小的等价类; 优化, 即在删除了部分连接权后对神经元的阈值进行优化; 提取, 即从经优化得到的神经网络中提取规则; 简化, 即将上述规则简化, 使其更易于理解.

根据上面所建立的网络结构, 该网络存在下列性质^[6].

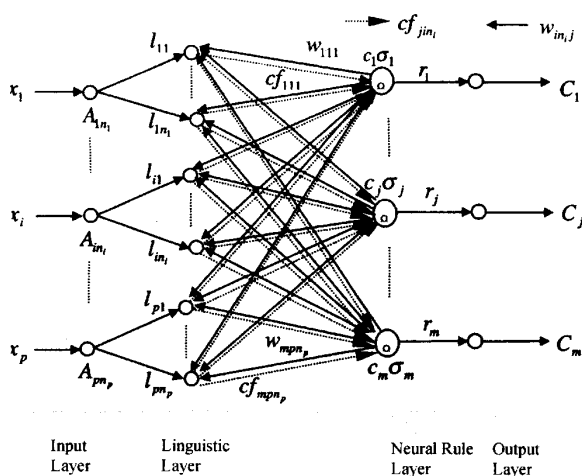


图3 获取知识的神经网络模型

Fig. 3 Neural network model of knowledge acquisition

性质:若 $l_{ik_i} = 1, (i \in 1, \dots, p)$, 则 $\forall k'_i \in \{1, \dots, n_i\}$, 且 $k'_i \neq k_i, l_{ik'_i} = 0$, 则 $(\exists k'_i) \in \{1, \dots, n_i\}$, 且 $k'_i \neq k_i, l_{ik'_i} = 1$.

上述性质所描述的就是: X 的任一区间分量 x_i 的 n_i 个区域均不重叠这一性质. 对一个训练好的网络, 只要输入矢量在输入样本范围内, 规则层每一个神经元的输出值约等于 1 或 0, 此时, 规则层节点的激活函数可近似视为一个布尔函数, 显然有:

(1) 若 $w_{ik_j} + \sum_{1 \leq i' \leq p, i' \neq i, w_{i'k_j} < 0} w_{i'k_j} + \theta_j \geq 0, i \in \{1, \dots, p\}, k_i \in \{1, \dots, n_i\}, \theta_j$ 为该节点的阈值, 则可在规则层节点 r_j 产生规则: if $x_i \in A_{ik_i}$ then C_j .

(2) 若 $\sum_{i=i_a}^{i_b} w_{ik_j} + \sum_{1 \leq i' \leq p, i' \neq i_a, \dots, i_b, w_{i'k_j} < 0} w_{i'k_j} + \theta_j \geq 0, i_a, i_b \in \{1, \dots, p\}, k_i \in \{1, \dots, n_i\}$, 则可在规则层节点 r_j 产生规则: if $(x_{i_a} \in A_{i_a k_{i_a}} \wedge \dots \wedge x_{i_b} \in A_{i_b k_{i_b}})$ then C_j .

根据该网络结构所具有的性质及从网络中提取规则的步骤可知, 从训练好的上述网络中提出规则的算法描述如下:

① 分别对每个输入分量 $x_i (i = 1, \dots, p)$ 与任一个规则层节点 $j (j = 1, \dots, m)$ 的反向正联接权按降序排序, 得到一有序集 $W_{ij}^+ = \{w_{ik_j} \mid w_{ik_j} \geq 0, w_{ik_j} \geq w_{i_{k+1}j}, k \in (1, \dots, n_i)\}$, 反向负联接权按升序排序, 得到一有序集 $W_{ij}^- = \{w_{ik_j} \mid w_{ik_j} < 0, w_{ik_j} \leq w_{i_{k+1}j}, k \in (1, \dots, n_i)\}$.

② 从 1 阶至最大为 q 阶项, 搜索出满足: $(w_{ik_j} - \sum_{1 \leq i' \leq p, i' \neq i, w_{i'k_j} < 0} \max(|w_{i'k_j}|) + \theta_j \geq 0$ 及 $\sum_{i=i_a}^{i_b} w_{ik_j} - \sum_{1 \leq i' \leq p, i' \neq i_a, \dots, i_b, w_{i'k_j} < 0} \max(|w_{i'k_j}|) + \theta_j \geq 0$ 的各种可能的组合项, 若某项 $P = \prod_{i_a}^{i_b} l_{ik_i}$ 所对应的规则 $P \rightarrow C_j$ 存在, 把 P 加到项集 I 中, 并把与该项相对应的权值从序集中删除.

③ 对项集 I 中的任一项 P , 用 $x_i \in A_{ik_i}$ 的语言值替代其中的项 l_{ik_i} , 并生成一条规则 $\prod_{i_a}^{i_b} x_i \in A_{ik_i} \rightarrow C_j$.

4 规则的精化过程

利用神经网络的规则抽取算法得到的产生式规则, 因冗余规则多, 因此有必要进一步利用粗集理论对其精化处理. 其具体的方法是, 先将获取的产生式规则的前件及后件构成粗集里的决策表, 对于决策属性 $d \in A$ 及其特定值 V_d , 关注的是满足 $d(e) = V_d$ 的对象 e 的集合 $\{V_d\}$. 用矩阵形式可以将区分所有属于集合 $\{V_d\}$ 的对象和属于 $U - \{V_d\}$ 集合的对象的属性值对表示出来.

用 e_i 代表任何一个属于集合 $\{V_d\}$ 的对象, $i = 1, 2, \dots, \text{card}(\{V_d\}) = m$; e_j 代表任何一个属于集合 $U - \{V_d\}$ 的对象, $j = 1, 2, \dots, \text{card}(U - \{V_d\}) = n$. 决策矩阵定义为:

$$DM_{ij} = \{(a, a(e_i)); a(e_i) \neq a(e_j)\} \quad (2)$$

集合 DM_{ij} 包含所有 e_j 和 e_i 相异的属性值对. 其形式上表示为矩阵 $DM = [DM_{ij}]_{m \times n}$.

而从决策矩阵提取最大泛化规则(最小决策规则集)的基本方法为^[7]:

$$RUL = \bigcup_i RUL_i \quad (3)$$

(3) 式中, $RUL_i = \{r \in RUL \mid A_r(e_i) = (V_{i1}, V_{i2}, \dots, V_{im})\}; e_i \in \{V_d\}$, 符号 RUL_i 表示所有条件与对象 e_i 相匹配的最大泛化规则集.

对于给定的决策矩阵 DM 和决策值 V_d , 笛卡尔积 $F_i = DM_{i1} \times DM_{i2} \times \dots \times DM_{i_r}$ 包含决策矩阵 DM 中第 i 行的所有序列对, 因 F_i 中的某些分量可能相同, 对于集合 $\tilde{F}_i = \{\{t\} \mid t \in F_i\}$ 中, $\{t\}$ 是向量中所有不同分量的集合, \tilde{F}_i 中的元素是对应于 $\{V_d\}$ 的所有规则, 因这些规则至少匹配一个对象 $e_i \in \{V_d\}$, 并且不与任何属于 $U - \{V_d\}$ 的对象相匹配. \tilde{F}_i 中的规则是偏序的, 其中最小的元素集合表示为 MIN_i .

而 MIN_i 集合的最大泛化规则可以通过一个布尔函数(决策函数)来计算. 决策函数 B_i 由决策矩阵的第 i 行构造, 即对于 $(DM_{i1}, DM_{i2}, \dots, DM_{in})$ 的分量, 把出现在 DM_{ij} 中的每个属性值对作为一个布尔变量, 并且对每个 $DM_{ij} (j = 1, 2, \dots, n)$ 的分量进行先析取后合取的布尔运算, 即

$$B_i = \bigwedge_j (\bigvee DM_{ij}) \quad (4)$$

通过(4)式的计算可知, $\bigcup MIN_i$ 即为当前决策类的所有的最大泛化规则(最小决策规则集). 最后从最大泛化规则的每一个元组中, 以属性条件的并作为产生式规则条件, 以决策属性的值作为结论构成产生式规则.

但完整的产生式规则还要计算其规则强度(可信度), 对得到的每一条产生式规则作如下计算得到规则强度.

令 R 表示产生式规则, $[x]_R$ 表示满足该规则条件属性的元组集合, D 表示满足该规则决策属性的元组集合, $\text{card}([x]_R)$ 表示元组集合中元组总数, 可以利用决策表的元组属性求得该值. 令规则 R 的可信度为 Cf_k , 定义如下:

$$Cf_k = \frac{\text{card}([x]_R \cap D)}{\text{card}([x]_R)} \quad (5)$$

由于规则前件中每一个条件属性对得出该决策结论的贡献大小不同, 也就是该前提属性对决策属性的依赖度不同, 因此通过值约简后的决策表, 重新计算没有删除前的条件属性对决策属性的依赖度 cf . 其计算方式为^[8]:

$$cf = \gamma_R(P) - \gamma_{R-\{a\}}(P) \quad (6)$$

而属性集 P 对 R 的依赖程度用 $\gamma_R(P)$ 表示. 定义为:

$$\gamma_R(P) = \frac{\text{card}(\text{POS}_R(P))}{\text{card}(U)} \quad (7)$$

$$\text{POS}_R(P) = \bigcup_{X \in U/\text{IND}(P)} \text{apr}_R(X) \quad (8)$$

(7) 式中 $\text{card}(\cdot)$ 表示集合的基数; (8) 式 $\text{POS}_R(P)$ 中是属性集 R 在 $U/\text{IND}(P)$ 中的正区域. 而对于任何一个属性集合 $P \subseteq A$, 不可分辨关系 IND 为:

$$\text{IND}(P) = \{(x_i, x_j) \in U \times U \mid f(x_i, a) = f(x_j, a), \forall a \in P\} \quad (9)$$

而属性集 $C \subseteq A$ 对论域 U 进行划分得到的等价关系族记为 U/C ;

对任何一个对象子集 $X \subseteq U$ 和属性子集 $R \subseteq A$, R 的下近似定义为:

$$\text{apr}_R(X) = \bigcup \{Y \in U/\text{IND}(R) \mid Y \subseteq X\} \quad (10)$$

通过(6)式计算出 cf_{kj} ; 然后再将规则中简化的数值通过反模糊化处理, 从而得到语言描述式的产生式知识规则.

至此得到的产生式规则形式为:

$$R_k : \bigwedge_{j=1}^m P_{kj} (cf_{kj}) \rightarrow C_k (cf_k) \quad (11)$$

式中: P_{kj} 表示产生式规则的前提条件; C_k 表示产生式规则的结论; k 表示新产生的规则序号; j 表示第 k 条产生式规则的第 j 个前提条件; m 表示该条规则的前提总数.

5 结论

本文在总结以往的知识获取方法时, 给出了基于知识的神经网络产生式规则获取方法.

(1) 将数据分类后已存的专家规则的依赖度作为神经规则的初始权值, 这样能充分利用已有的专家经验;

(2) 在构建新的网络结构时, 为了降低网络结构的复杂性, 提高网络训练的效率, 系统采用了动态训练网络的方法, 并且能利用已训练过的样本信息来决定是否增加新节点;

(3) 从训练后的网络反馈权值中, 凭借网络所具有的性质来提取规则, 并给出了提取规则的算法;

(4) 为了剔除所提取的冗余规则, 系统又进一步将所提取的规则通过粗集理论精化处理, 从而得出

符合实际地最简化的产生式规则。

参考文献 References

- [1] IOANNIS H G, JIM P. Integrating (rules, neural networks) and cases for knowledge representation and reasoning in expert systems[J]. Expert Systems with Applications, 2004, 27(1): 63-75.
- [2] WEI Y, LI P K, CHUN-H C. A QFD-enabled product conceptualisation approach via design knowledge hierarchy and RCE neural network[J]. Knowledge-Based Systems, 2005, 18(6): 279-293.
- [3] SARIDAKIS K M, DENTSORAS A J. Case-DeSC: A system for case-based design with soft computing techniques [J]. Expert Systems with Applications, 2007, 32(2): 641-657.
- [4] 胡月明, 薛月菊, 李 波, 等. 从神经网络中抽取土地评价模糊规则[J]. 农业工程学报, 2005, 21(12): 93-97.
HU Yue-ming, XUE Yue-ju, LI Bo, et al. Extracting fuzzy rules from neural networks for land evaluation[J]. Transactions of the CSAE, 2005, 21(12): 93-97.
- [5] 罗 梅, 温绍洁, 王万森. WBITS 中基于神经规则的知识表示与管理[J]. 微计算机信息, 2006, 22(314): 304-306.
LUO Mei, WEN Shao-jie, WANG Wan-sen. Neurule-based Knowledge Representation and Knowledge Management in WBITS[J]. Micro-Computer Information, 2006, 22(314): 304-306.
- [6] 韦卫星, 莫 赞. 一种基于神经网络的知识获取方法研究与应用[J]. 计算机工程与应用, 2004(5): 95-98.
WEI Wei-xing, MO Zan. Study of Knowledge Extracting from Neural Network and Its Application[J]. Computer Engineering and Application, 2004(5): 95-98.
- [7] 刘同明. 数据挖掘技术及其应用[M]. 北京: 国防工业出版社, 2001.
LIU Tong-min, et al. Data Mining Technology and Its Application[M]. Beijing: National Defense Industry Press, 2001.
- [8] 张文修, 仇国芳. 基于粗糙集的不确定决策[M]. 北京: 清华大学出版社, 2005.
ZHANG Wen-xiu, QIU Guo-fang. Uncertain Decision Based on Rough Sets[M]. Beijing: Tsinghua University Press, 2005.

Research on the acquisition method of neural network production rule based on knowledge

LIU Dao-hua^{1,2}, YUAN Si-cong¹, JIANG Xiang-kui¹, LU Di¹, WANG Rui³

(1. School of Mech. & Elec. Eng., Xi'an Univ. of Arch. & Tech., Xi'an 710055, China;

2. Dept. of Computer Science, Xinyang Normal University, Xinyang 464000, China;

3. School of Information & Control Enec., Xi'an Univ. of Arch. & Tech., Xi'an 710055, China)

Abstract: In this paper, the continual data are divided and the credible degrees of former knowledge rules are taken as the initial weights of neural network so as to build a new network structure which will be executed as dynamic training, whose network training algorithm is illuminated in the paper. The correct classification of the weights of trained network and its properties which the structure of network has used illuminate the extracting algorithm of the rules. The production rule with redundancy will be purified by using the rough set theory, so that the simplest production rules will be acquired. Making full union between neural network, and the rough set theory, and exploring their merits have a vital practical significance on the research of the intelligent information systems.

Key words: neural network; production rule; knowledge acquisition; reduction of rough set theory