文章编号：1007-4619 (2004) 04-0331-08

遥感图像识别中粗糙集理论与神经网络的结合

余春艳[1,2], 吴明晖[3], 吴 明[2]
(1. 浙江大学 计算机学院, 浙江 杭州 310027;
2 福州大学 计算机系, 福建 福州 350002;
3 浙江大学 城市学院计算机系, 浙江 杭州 310027)

摘 要： 由于传统神经网络与遥感图像信息量不相匹配, 为此, 提出将粗糙集理论集成至遥感图像神经网络识别中。首先分析了神经网络与粗糙集理论结合的可能性以及优势, 在此基础上提出了基于粗糙集的遥感图像神经网络识别模型, 并就其中的粗糙集方法处理样本特征集模块和遥感图像识别神经网络模块展开详细的分析。通过对比实验数据说明集成粗糙集理论的遥感图像神经网络识别能够有效提高遥感图像的识别效率, 具有较强的现实意义。

关键词： 粗糙集理论; 神经网络; 遥感图像; 模式识别
中图分类号： TP391.4 文献标识码： A

1 引 言

神经网络理论是以人脑为基础的一门智能科学, 由多个非常简单的神经元按照某种方式相互连接而形成神经网络。每个神经元接受大量其它神经元的输入, 通过非线性输入/输出关系, 产生输出, 影响其它神经元, 从而实现从输入状态空间到输出状态空间非线性映射, 具有高度并行性、容错性、自适应性和自学习能力等特点。

模式识别就是利用计算机对某些物理对象进行分类, 在错误概率最小的条件下, 使得识别的结果尽量与客观事物相符[1]。由于神经网络所具有的自适应性和自学习能力, 神经网络在模式识别领域中的应用被认为是神经网络应用最成功的一个领域之一[2]。而遥感图像识别是模式识别技术在遥感技术领域的具体应用, 其主要识别对象是遥感图像, 也就是对地球表面及其环境在遥感图像上的信息进行属性的识别和分类, 从而达到识别图像信息对应的实际地物, 提取所需地物信息的目的。基于人工神经网络方法的遥感图像识别是当前遥感信息处理的一个热点。

神经网络利用复杂的非线性系统对遥感图像中大量不准确而且不完整的数据进行处理, 并且克服网络本身的不精确性, 有较强的鲁棒性和容错能力。但是冗余信息甚至错误信息的干扰, 容易导致神经网络结构复杂, 训练时间过长, 收敛速度过慢, 计算量急剧增加等问题, 从而极大地影响了神经网络在遥感图像识别领域进一步的应用。因此如何将其它人工智能理论与神经网络有机结合, 提高其效率已经成为当前理论研究的热点问题。

产生于 20 世纪 70 年代初的粗糙集理论提供了一个知识发现的框架, 从事实中抽取规则, 用决策表的形式来获取更简单的知识表示, 从而实现从样本数据中进行特征选择, 加快概念学习和分类, 减少分类的代价, 提供分类的质量。由此, 本文考虑提出一种结合粗糙集理论对神经网络在遥感图像识别中的应用进行改进的方法。

2 神经网络结合粗糙集理论的可行性和重要意义

遥感图像处理的重要任务是从大量数据中获取知识, 表达知识并推理决策规则, 其中涉及如何对大量的冗余数据和不完全数据进行处理。对于不准确以及不完整的知识, 粗糙集理论和人工神经网络方

收稿日期：2003-05-30; 修订日期：2003-07-14

作者简介：余春艳(1976—), 女, 博士, 福州大学计算机系, 曾就读于浙江大学计算机学院, 研究方向为协同虚拟环境, 人工智能等, 已在国内外发表论文十余篇。

法都显示了它们各自的优势。粗糙集理论模拟人类的抽象逻辑思维[3]，基于不可分辨性的思想和知识简化的方法，从数据中推理逻辑规则作为知识系统的模型；其要点是将知识与分类联系在一起，而其基本框架可以归纳为：以不可辨关系划分所研究领域的知识，形成知识表达系统，利用上、下近似集逼近描述对象，通过知识约简，从而获得最简知识[4, 5]。而神经网络模型是形象知觉思维，利用非线性映射的思想和并行处理的方法，用神经网络本身结构表达输入和输出关联知识的隐函性编码[6]。因此，粗糙集理论可以输入定性、定量或者混合性信息，而神经网络一般不能处理具有语义形式的输入。神经网络可以实现无导师聚类学习，但不能确定哪些知识是冗余的，哪些知识是有用的，而粗糙集理论可以描述知识表达中不同属性的重要性，简化知识表达空间，但它是从训练数据中推理规则的。此外，粗糙集理论的知识简化可以用并行算法实现，而神经网络实现了信息并行处理[7]。

将粗糙集与神经网络结合，已有一些成果，但这些成果主要利用粗糙集来进行特征处理，并没在后续的识别过程中利用得到的规则。例如曾黄麟在文献[6]中提出用粗糙集提取汉字的特征，得到特征之后再用传统的 BP 神经网络进行识别，但值得指出的是粗糙集只应用于特征处理部分。与此相对应的是本文首先使用粗糙集对特征进行处理，形成输入数据，然后利用处理特征时产生的规则，进一步指导识别网络的构造，从而提高网络的效率。实验表明，针对基于神经网络的模式识别在冗余数据处理上的不足，结合粗糙集理论，对神经网络的输入样本特征集进行一定的约简，利用粗糙集推导知识的能力来构造神经网络，是可行的，并且能够提高整个系统的效率[8, 9]。

同时，将粗糙集方法作为神经网络遥感图像识别的前置系统还具有以下的意义：

(1) 通过粗糙集方法减少了信息表达的属性数量，从而减少了神经网络构成系统的复杂性，也相应减小了后续使用过程中信息作为网络输入时的特征值计算时间；

(2) 通过粗糙集方法去掉冗余信息，使得训练集简化，也减少了网络的训练时间；

(3) 使用神经网络作为后置的信息识别系统，有容错以及抗干扰的能力；

(4) 由于粗糙集理论在简化知识的同时，很容易推导出决策规则，因而也可以作为后续使用中的信息识别规则，可以将粗糙集方法得到的结果与神经网络方法得到的结果相比较，做进一步的修正。

3 构建集成粗糙集理论的遥感图像识别神经网络

3.1 基本框架

以遥感图像识别为应用目标的集成粗糙集理论的神经网络，是将人工智能的粗糙集理论同神经网络模式识别相结合而建立的系统，其目标是利用粗糙集理论的知识推导优势，弥补神经网络一些自身不可克服的缺陷，同时利用神经网络的强大分析优势，更好地发挥粗糙集理论的作用。具体说来，是利用粗糙集处理样本特征集，根据得到的结果，建立结构更简单，训练事件相对较短的神经网络遥感图像识别系统[10]。因此，集成粗糙集理论的遥感图像识别神经网络包括 3 个部分：特征提取模块、粗糙集方法处理样本特征集模块和遥感图像识别神经网络构建模块，如图 1。

原始数据源以遥感图片等多种形式存在并作为系统的初始输入，经过特征提取模块的一系列处理之后，针对不同问题的需要，形成了不同的数字特征，如形状特征、纹理特征、空间特征等，其最终输出为原始样本特征集。不同的特征将影响特征向量的维数和表示，但具体识别方法是一致的，本文以基于像素的分类数据为例。

得到样本特征集之后，就进入了真正的识别处理阶段。这一阶段包括 3 个步骤。

首先要降低样本特征的维数，这一步是根据粗糙集中条件属性的约简原理来设计算法的，其中又包括了属性的离散归一算法以及条件属性的约简算法。

其次，在降低特征维数的同时，需要产生相应的规则。这些规则就是根据样本的分类特性得到的知识，规则的前部是约简后剩余属性的组合，规则的后部是决策属性。在大多数情况下，就是分类的先验知识。

这两步构成了整个系统的粗糙集处理模块，后面识别神经网络构造的前提和基础。

最后进入识别神经网络构建这一核心部分。识别神经网络的隐层以及隐层节点数的选择、初始权值的设置、激活函数的选择等，都是在前一部分得到的属性集和规则集的指导下进行的。由于具有明确的目标性，以及接近问题的本身特征，网络的训练时间相对缩短，识别精度也有了一定的提高。

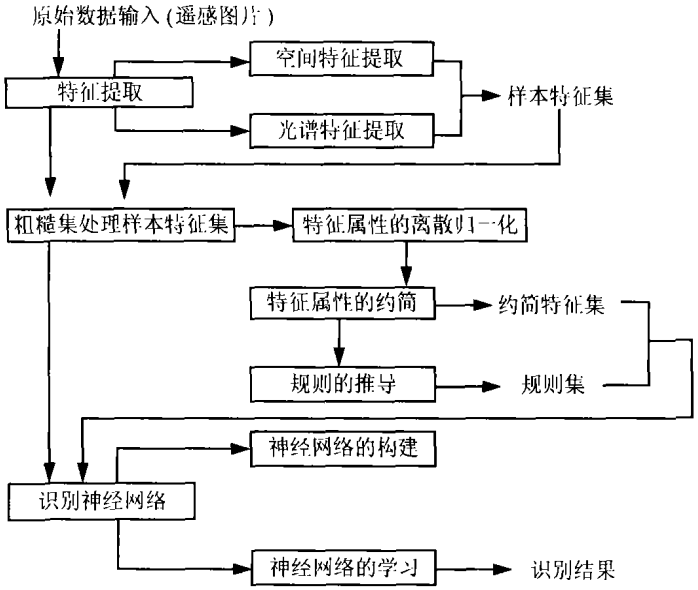


图 1　基本框架图

Fig. 1　Basic Framework of The System

3.2　粗糙集方法对特征的处理

我们利用粗糙集方法作为遥感图像神经识别网络的前置步骤, 其主要目的是做特征筛选, 即利用粗糙集方法来选择分类的属性, 作为建立识别神经网络学习的输入数据, 目的是减少神经网络的输入层的节点个数, 从而缩短神经网络的训练时间, 简化神经网络的结构, 同时在约简特征维数时, 推导出相应的规则, 即分类知识, 作为构造识别神经网络中间层的依据。

根据上述目标, 我们将粗糙集方法对特征的处理分为 3 个部分: 特征属性的离散归一化、特征属性的约简和规则的推导。

3.2.1　特征属性的离散归一化

在粗糙集的特征约简中, 必须把所得的信息数据归一化, 使得用于粗糙集方法分析的决策表是有限维数的离散化数据表, 即:

对于一个连续值属性 a, 设属性值的域为 $[a_{\min}, a_{\max}]$, 离散归一化就是产生一个对属性值域的划分 π_a:

$$\pi_a = \{[d_0, d_1)\ [d_1, d_2), \cdots, [d_{k-1}, d_k]\} \quad (1)$$

其中, $d_0 = a_{\min}$, $d_k = a_{\max}$, $d_{i-1} < d_i$, $i = 1, 2, \cdots, k$, i 就是离散归一化代表值, k 是离散归一化的级。

离散归一化方法应该满足如下条件: (1) 属性离散归一化后的空间维数应尽量小, 即每一离散归一化后属性值的种类应尽量少; (2) 属性值被离散归一化中信息丢失应尽量少。

离散归一化的方法可分为等间隔划分法、等频率间隔划分法、全局聚类分析方法等。我们提出一种适用于遥感图像亮度区间特点的特征属性离散归一算法——区间搜索算法。该算法基于一种简单的划分区间的思想, 在不同类别同种条件属性间数值分段比较明显情况下, 是一种简单有效的算法。

对每一个特征属性 a, 将其属性值集 V_a 分类, 得到如下排序: $v_a^1 < v_a^2 < \cdots < v_a^i < \cdots < v_a^{|v_a|}$。用 C_a 表示属性 a 产生的所有分离区间的集合, 除了具有相同决策属性并且属性值明显不需要区分的区间外, 包括两个观测属性值之间的所有区间。

$$X_a^i = \{x \in U \mid a(x) = v_a^i\} \quad (2)$$

$$\Delta_a^i = \{v \in V_a \mid \exists x \in X_a^i \text{ such that } d(x) = v\} \quad (3)$$

$$C_a = \left\{ \frac{v_a^i + v_a^{i+1}}{2} \,\middle|\, |\Delta_a^i| > 1 \text{ 或 } |\Delta_a^{i+1}| > 1 \text{ 或 } \Delta_a^i \neq \Delta_a^{i+1} \right\} \quad (4)$$

事实上, 除了用条件属性 a 来划分决策属性相同的属性值外, 在所有 v_a^i 和 v_a^{i+1} 中间。都有一个区间。如果对于一个条件属性, 找不到合适的区间, 该属性不作处理。

3.2.2　特征属性的约简

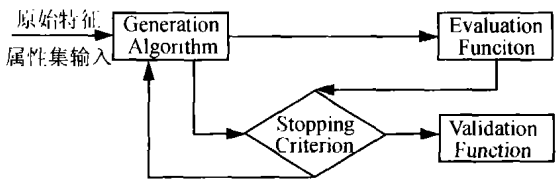
先作如下定义:

令 R 为一等价关系族, 且 $r \in R$, 如果 $ind(R) = ind(R - \{r\})$, 则称 r 为 R 中可省略的, 否则 r 为 R

中不可省略的。

对于属性子集 $P \subseteq R$, 若存在 $Q = P - R$, $Q \subseteq P$, 使得 $ind(Q) = ind(P)$, 且 Q 为最小子集, 则 Q 称为 P 的简化。

特征属性的约简算法应满足如下两个标准: 分类精度不能下降; 并且按照简化特征集的结果类分布, 必须尽可能地接近按所有特征得到的结果集分布。特征属性的约简算法分为 4 步: (1) 用于产生下一个被评价的子集的过程(Generation Algorithm); (2) 用于评价子集的代价函数(Evaluation Function); (3) 为避免搜索过程过长而中止搜索过程的准则(Stopping Criterion); (4) 确定子集是否合法的合法化检查过程(Validation Function)。图 2 中(a) 给出了 4 个步骤之间的流程关系, 而(b) 描述了具体的算法。



(a)

1. 设 $F=\{a_1,a_2,\cdots,a_n\}$, 初始化 *MinQuality*
2. 当 $|F|>1$ 时, 重复 2-6
3. 对 F 中的每一个属性 a, 重复 4-6
4. 计算从 F 中除去 a 后的集合 F_1 的 $f(F_1)$, $F_1=F-\{a\}$
5. *CurrentQuality*=$f(F_1)$, 比较 *CurrentQuality* 与 *MinQuality*
6. 若 *CurrentQuality*>*MinQuality*, 将属性 a 加入 *KeepAttribute* 集合
7. 转到 3
8. *F*=*KeepAttribute*, 返回 2
9. 结束

(b)

图 2 特征属性的约简算法

Fig. 2 Reduction algorithm for features

算法中 Evaluation Function f 定义如下:

$$f(B) = (1-a) \times \frac{\text{cost}(A) - \text{cost}(B)}{\text{cost}(A)} + a \times \min\left\{\varepsilon, \frac{|[\sin S \mid S \cap B \neq \Phi]|}{|S|}\right\} \quad (5)$$

其中, S 是根据 Generation Algorithm 产生的集合, a 定义为子集的值和 Hitting Fraction 间的权值, B 是 A 通过演化搜索算法得到的 A 的子集, 函数 cost 定义了属性子集的值。而且, 产生子集的 Generation Algorithm 从所有特征属性的几何开始, 逐次减去一个特征属性, 达到约简特征属性的目的。

3.2.3 规则的推导

规则的推导是以决策表为单位进行的。3.2.1 和 3.2.2 已经简化了条件属性, 将条件属性作为规则的前部, 决策属性作为规则的后部, 形成了决策规则 $\alpha \rightarrow \beta$, 同时, 我们给出 $accuracy(\alpha \rightarrow \beta)$ 以表示规则的可信度。

$$accuracy(\alpha \rightarrow \beta) = \frac{support(\alpha \cdot \beta)}{support(\alpha)} \quad (6)$$

其中, $support(\alpha)$ 为含属性 α 的对象集合, $support(\alpha \cdot \beta)$ 为规则 $\alpha \rightarrow \beta$ 支持的集合。在其后的神经网络构造过程中, 如果出现如下情况: 有的对象包含规则的前部 α, 却有不同的 β 值, 可以用这种规则的可信度作为权值。

3.3 构建遥感图像识别神经网络

在粗糙集方法处理样本特征集之后, 进入构建遥感图像识别神经网络这一核心阶段。由于 BP 网络在有监督的训练过程中, 可以实现知识自动获取, 从而表现出有效的学习机制。因此, 我们依然选用 BP 网络作为基本的网络结构, 但在选择各层节点上采用了与粗糙集理论相结合的方法, 充分利用 3.2 节中粗糙集处理部分得到的结果辅助神经网络的构造。

3.3.1 网络结构

与传统的用于遥感图像识别的 BP 神经网络不同, 本文采用的方式是根据粗糙集方法处理样本特征集模块推出的样本集规则来构建遥感图像的识别神经网络的结构。这个识别神经网络分为两个部分:

• 规则的条件部分由前两层网络连接实现。第 1 层神经元$\{X_1, X_2, \cdots, X_m\}$为离散和约简后的输入特征向量, 即规则的条件属性; 第 3 层神经元$\{R_1, R_2, \cdots, R_s\}$表示的是粗糙集推出的规则, 其中每个神经元代表一条规则, 中间第 2 层$\{H_1, H_2, \cdots, H_q\}$为隐层神经元, 连接条件属性和规则, 作用是便于网络收敛和学习, 提高条件属性与规则对应的精度。

• 结论部分为规则与结论的连接层, 第 4 层神经元$\{Y_1, Y_2, \cdots, Y_n\}$是规则的结论, 即问题的输出, 在本文中为地物的种类输出。第 3 层规则节点$\{R_1, R_2, \cdots, R_s\}$与第 4 层结论节点的连接实现了结论部分的学习。

如图 3, 这个识别神经网络模型描述的是一个多输入多输出系统, 网络共由 4 层神经元组成:

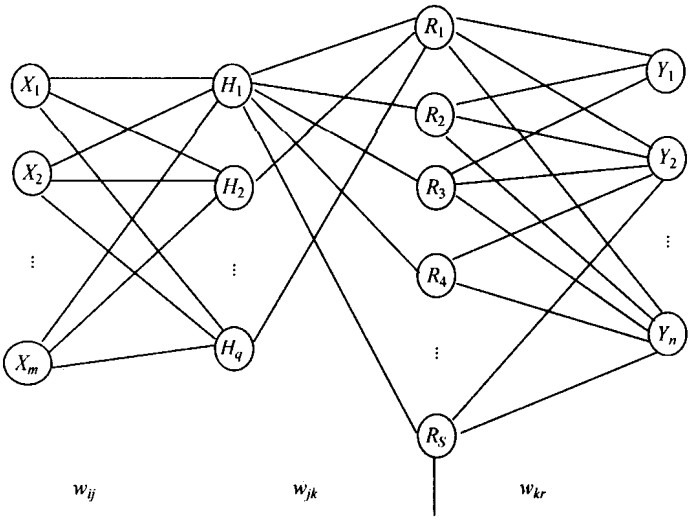


图 3　集成粗糙集理论的遥感图像识别神经网络结构图

Fig. 3　Structure of the remote sensing image recognition neural network combined with rough set theory

第 1 层是输入层, 表示输入 $X=\{X_1, X_2, \cdots, X_m\}^{\mathrm{T}}$。

第 2 层是隐含层, 表示为 $H=\{H_1, H_2, \cdots, H_q\}$, 作用是将 m 个输入量按某种非线性关系映射到下一层上。输入层与隐含层之间的连接权矩阵为 w_{ij}。H 层的输出计算公式为:

$$H_j = f_1\left(\sum_{i=1}^{m} w_{ij} \times X_i + \alpha_i\right), \quad j = 1, 2, \cdots, q \quad (7)$$

其中, f_1 是隐含层的激活函数。在较简单的情况下, 隐含层可以不要, 前提是输入层可以快而准确地收敛到规则层。增加这一层的目的是增加精度, 加快收敛速度。

第 3 层为规则层, 表示为 $R=\{R_1, R_2, \cdots, R_s\}$, 每个神经元代表一条规则, 这些规则的获取和描述的理论依据是粗糙集理论。该层节点与相邻两层节点的连接取决于节点所表示的规则的条件属性和决策属性分别与第 2 层和第 4 层的哪些节点有关。第 3 层的输出计算公式为:

$$R_k = f_2\left(\sum_{j=1}^{q} w_{jk} \times H_j + \beta_j\right) \quad k = 1, 2, \cdots, s \quad (8)$$

其中 f_2 是规则层的激活函数, 它的选取必须根据具体问题选择合适的映射函数。

第 4 层为输出层, 表示为 $Y=\{Y_1, Y_2, \cdots, Y_n\}$, 每个神经元代表了问题的一个决策属性, 计算公式为:

$$Y_r = f_3\left(\sum_{k=1}^{s} w_{kr} \times R_k + \theta_k\right), \quad r = 1, 2, \cdots, n \quad (9)$$

3.3.2　学习算法

(1) 将经过粗糙集方法处理的样本特征集 S 作为识别神经网络的训练输入(其中 m 为特征约简集中属性的个数, n 为样本的个数):

$$S=\{s_1, s_2, \cdots, s_n\} \text{ 且 } s_i=\{a_{i1}, a_{i2}, \cdots, a_{im}\}, \quad i=1, 2, \cdots, n;$$

(2) 规则层的输出为 $R=\{R_1, R_2, \cdots, R_n\}$, 选择中间隐层的数目;

(3) 开始学习过程: 设置网络的最大训练次数 *epochs*, 期望误差 *errors* 以及学习率 *lr*, 初始化识别神经网络权值矩阵;

(4) 开始迭代, 按照公式(7)和(8)计算第 3 层的输出值, 并计算输出值与期望值之间的误差 ε;

(5) 如果 $\varepsilon >$ *errors* 或者训练次数没有达到 *epochs*, 则返回(4);

(6) 将第 3 层规则的期望输出值作为第 2 部分网络的输入值, 决策属性集作为第 4 层的输出向量集 $Y=\{y_1, y_2, \cdots, y_n\}$;

(7) 按第 1 层的训练方法迭代网络, 直到满足停止条件, 计算输出值公式为(9)。

其中(1)到(5)为学习算法的第一阶段, 而(6)到(7)为学习算法的第二阶段。

4　与传统神经网络的实验分析比较

以某地域的遥感图像为原始数据输入, 对该遥感图像的地物进行识别为例来加以实验性的分析比较。在原始数据中, 随机选取 90% 的数据作为训练

集,剩余 10%作为测试集。

4.1 测试集描述

表 1 给出了 7 种地物在 4 个波段中亮度的具体数值;其中地物分别为水域(water)、沙滩(sand)、阴影(shade)、旱地(dryland)、森林(forest)、城区(city)和水稻(paddy)。我们将表 1 中的数据作为测试数据,而表 2 给出的是将测试数据离散化后的结果,表 3 是从粗糙集处理模块中推导得到的规则集。

表 1 测试数据

Table 1 Training samples

	ch4	ch5	ch6	ch7	class
1	16	14	10	2	water
2	14	12	10	2	water
3	14	13	8	2	water
4	15	14	8	2	water
5	16	14	9	2	water
6	15	14	11	4	sand
7	16	14	12	5	sand
8	17	15	14	4	sand
9	16	16	12	7	sand
10	16	16	13	5	sand
11	12	15	15	6	shade
12	13	15	17	7	shade
13	13	12	15	7	shade
14	13	14	18	8	shade
15	13	12	18	8	shade
16	18	20	25	11	dryland
17	19	20	26	10	dryland
18	18	21	27	12	dryland
19	18	19	27	13	dryland
20	20	21	22	12	dryland
21	13	13	29	13	forest
22	13	14	28	14	forest
23	16	15	30	16	forest
24	14	16	29	15	forest
25	15	13	25	12	forest
26	16	18	18	7	city
27	17	21	22	7	city
28	17	20	20	7	city
29	18	20	19	7	city
30	17	20	21	8	city
31	19	21	29	13	paddy
32	18	21	28	13	paddy
33	18	20	30	14	paddy
34	17	21	31	12	paddy
35	19	22	28	11	paddy

表 2 离散处理后的测试数据

Table 2 Discretization of attribute values

	ch4	ch5	ch6	ch7	class
1	[16,17]	[14,15]	[*,11]	[*,3]	water
2	[14,15]	[*,13]	[*,11]	[*,3]	water
3	[14,15]	[13,14]	[*,11]	[*,3]	water
4	[15,16]	[14,15]	[*,11]	[*,3]	water
5	[16,17)	[14,15]	[*,11]	[*,3]	water
6	[15,16)	[14,15)	[11,15)	[3,6]	sand
7	[16,17)	[14,15)	[11,15)	[3,6]	sand
8	[17,18)	[15,16)	[11,15)	[3,6]	sand
9	[16,17)	[16,17)	[11,15)	[7,8]	sand
10	[16,17)	[16,17)	[11,15)	[3,6]	sand
11	[*,13)	[15,16)	[15,18)	[6,7]	shade
12	[13,14)	[15,16)	[15,18)	[7,8]	shade
13	[13,14)	[*,13)	[15,18)	[7,8]	shade
14	[13,14)	[14,15)	[18,19)	[8,9]	shade
15	[13,14)	[*,13)	[18,19)	[8,9]	shade
16	[18,19)	[20,21)	[24,26)	[11,12]	dryland
17	[19,20)	[20,21)	[26,28)	[9,11]	dryland
18	[18,19)	[21,22)	[26,28)	[12,13]	dryland
19	[18,19]	[19,20)	[26,28)	[13,14)	dryland
20	[20,*]	[21,22)	[22,24)	[12,13)	dryland
21	[13,14]	[13,14)	[29,30)	[13,14)	forest
22	[13,14]	[14,15)	[28,29)	[14,15)	forest
23	[16,17]	[15,16)	[30,31)	[15,*)	forest
24	[14,15]	[16,17)	[29,30)	[15,*)	forest
25	[15,16]	[13,14)	[24,26)	[12,13)	forest
26	[16,17)	[17,19)	[18,19)	[7,8)	city
27	[17,18)	[21,22)	[22,24)	[7,8)	city
28	[17,18)	[20,21)	[19,22)	[7,8)	city
29	[18,19)	[20,21)	[19,22)	[7,8)	city
30	[17,18)	[20,21)	[19,22)	[8,9)	city
31	[19,20)	[21,22)	[29,30)	[13,14)	paddy
32	[18,19)	[21,22)	[28,29)	[13,14)	paddy
33	[18,19)	[20,21)	[30,31)	[14,15)	paddy
34	[17,18)	[21,22)	[31,*)	[12,13)	paddy
35	[19,20)	[22,*)	[28,29)	[11,12)	paddy

我们将规则层和结论层的输出矩阵 T 和 T_1 设置如下:

$$T=\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0\\
0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&1\\
1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0
\end{bmatrix}$$

$$T_1=\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&1&1&1&1&1\\
0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0
\end{bmatrix}$$

表 3　规则集

Table 3　Deduced rules

1	ch4([16, 17)) AND ch6([*, 11)) → class(water)
2	ch4([14, 15)) AND ch6([*, 11)) → class(water)
3	ch4([15, 16)) AND ch6([*, 11)) → class(water)
4	ch4([15, 16)) AND ch6([11, 15)) → class(sand)
5	ch4([16, 17)) AND ch6([11, 15)) → class(sand)
6	ch4([17, 18)) AND ch6([11, 15)) → class(sand)
7	ch4([*, 13)) AND ch6([15, 18)) → class(shade)
8	ch4([13, 14)) AND ch6([15, 18)) → class(shade)
9	ch4([13, 14)) AND ch6([18, 19)) → class(shade)
10	ch4([18, 19)) AND ch6([24, 26)) → class(dryland)
11	ch4([19, 20)) AND ch6([26, 28)) → class(dryland)
12	ch4([18, 19)) AND ch6([26, 28)) → class(dryland)
13	ch4([20, *)) AND ch6([22, 24)) → class(dryland)
14	ch4([13, 14)) AND ch6([29, 30)) → class(forest)
15	ch4([13, 14)) AND ch6([28, 29)) → class(forest)
16	ch4([16, 17)) AND ch6([30, 31)) → class(forest)
17	ch4([14, 15)) AND ch6([29, 30)) → class(forest)
18	ch4([15, 16)) AND ch6([24, 26)) → class(forest)
19	ch4([16, 17)) AND ch6([18, 19)) → class(city)
20	ch4([17, 18)) AND ch6([22, 24)) → class(city)
21	ch4([17, 18)) AND ch6([19, 22)) → class(city)
22	ch4([18, 19)) AND ch6([19, 22)) → class(city)
23	ch4([19, 20)) AND ch6([29, 30)) → class(paddy)
24	ch4([18, 19)) AND ch6([28, 29)) → class(paddy)
25	ch4([18, 19)) AND ch6([30, 31)) → class(paddy)
26	ch4([17, 18)) AND ch6([31, *)) → class(paddy)
27	ch4([19, 20)) AND ch6([28, 29)) → class(paddy)

4.2　网络学习结果

将原始输入数据送入网络学习, 当设置学习率为 0.1, 训练次数上限为 1500 次, 隐含层节点为 4 时, 误差为 0.1447, 而当训练次数上限为 3000 次, 误差为 0.1239。

将由粗糙集处理模块得到特征约简数据送入网络学习, 作为第一部分的输入特征集, 当我们设置学习率为 0.1, 训练次数上限为 3000 时, 误差为 0.03259。而将其规则层的期望输入作为第二部分的输入集, 同样设置网络第二部分学习率为 0.1, 训练 300 多次的时候就达到 0.06 误差。

同时网络的前后两部分可以同时进行并行网络学习, 从而节约整体学习时间, 提高整体学习效率。

4.3　测试结果

将上面的测试数据送入系统以检测其识别率, 结果以 3×35 的矩阵的形式给出, 其中 M_1 为文中识别网络的输出结果, M_2 为传统 BP 网络的输出结果。

$$M_1=\begin{bmatrix}
0.0000&0.0000&0.0000\\
0.0000&0.6501&0.0000\\
0.0000&0.0000&0.0000\\
0.0000&0.0000&0.0000\\
0.0000&0.0000&0.0000\\
0.0000&0.0000&1.0000\\
0.0000&0.0000&1.0000\\
0.0000&0.0000&1.0000\\
0.0000&0.0000&1.0000\\
0.0095&0.0000&0.9869\\
0.0000&1.0000&0.0000\\
0.0030&1.0000&0.0005\\
0.0030&1.0000&0.0005\\
0.0000&1.0000&0.0000\\
0.0823&1.0000&0.0000\\
0.0058&1.0000&0.7436\\
1.0000&0.0000&0.0000\\
0.4578&0.7749&0.8278\\
0.2032&1.0000&0.9032\\
0.9999&1.0000&1.0000\\
1.0000&0.0000&0.9000\\
0.9999&0.0000&0.7043\\
0.9999&0.0000&0.3382\\
0.9999&0.0000&0.0000\\
1.0000&0.2050&0.3000\\
0.8373&0.0000&0.9911\\
1.0000&0.4712&0.7041\\
1.0000&0.0000&1.0000\\
1.0000&0.2712&1.0000\\
1.0000&0.2712&1.0000\\
1.0000&1.0000&0.0000\\
1.0000&1.0000&0.0000\\
1.0000&0.9843&0.0000\\
1.0000&0.9485&0.0000\\
1.0000&0.9986&0.0000
\end{bmatrix}^{\mathrm{T}}$$

$$M_2=\begin{bmatrix}
0.0038&0.0576&0.4260\\
0.0051&0.0696&0.4330\\
0.0055&0.7333&0.4349\\
0.0045&0.0649&0.4303\\
0.0036&0.0552&0.4244\\
0.0077&0.0918&0.4434\\
0.0049&0.0684&0.4323\\
0.0051&0.0699&0.4331\\
0.0155&0.1434&0.4611\\
0.0180&0.1570&0.4648\\
0.4936&0.7600&0.5644\\
0.4562&0.7399&0.5606\\
0.0787&0.3583&0.5035\\
0.4194&0.7187&0.5568\\
0.2026&0.5443&0.5280\\
0.4899&0.7580&0.5640\\
0.4061&0.7109&0.5556\\
0.5432&0.7847&0.5691\\
0.4978&0.7530&0.5563\\
0.1894&0.5336&0.5288\\
0.9933&0.1392&0.0216\\
0.9933&0.1412&0.0216\\
0.9920&0.1263&0.0212\\
0.9934&0.1428&0.0216\\
0.9812&0.0767&0.0204\\
0.3396&0.6675&0.5485\\
0.5255&0.7762&0.5675\\
0.4467&0.7346&0.5597\\
0.2492&0.5950&0.5376\\
0.4763&0.7508&0.5626\\
0.5323&0.7768&0.5655\\
0.5519&0.7869&0.5680\\
0.8791&0.5319&0.2213\\
0.8943&0.5303&0.2102\\
0.5381&0.7824&0.5687
\end{bmatrix}^{\mathrm{T}}$$

4.4　对比结果分析

我们取 *threshold* = 0.3000, 所有大于 *threshold* 的值认为是 1, 小于 *threshold* 则为 0。重新处理 M_1 和 M_2 后, 将结果与 T_1 相比(T_1 是从测试数据中得

到的正确的分类结果), 从而计算精确率。

由上述矩阵看到, M_1 中, 27 例被分对, 正确率为 27/35=77.14%; M_2 中, 只有 12 例分对, 正确率为 12/35=34.29%。如给定更长的训练时间, 和更多的训练样本, 两个网络的精确度都会提高, 但可以看出, 结合粗糙集的神经网络比未结合粗糙集的网络来说, 收敛得更加快, 训练时间更短, 识别准确率也更高。

5 结 论

本文分析了神经网络与粗糙集理论结合的可能性以及优势, 在此基础上提出了基于粗糙集的遥感图像神经网络识别模型, 并就其中的粗糙集方法处理样本特征集模块和模式识别神经网络模块展开详细的分析。最后的对比实验数据说明集成粗糙集理论的遥感图像神经网络识别能够充分发挥粗糙集和神经网络各自的优势, 一定程度上避免各自的缺陷, 有效提高遥感图像的识别效率, 具有较强的现实意义。

参 考 文 献 (References)

[1] Yao M, Luo J H. Research on Generalized Computing System [J]. *Journal of Systems Engineering and Electronics*, 1998, **9**(3): 39—43.

[2] Huang D S. Neural Network for Pattern Recognition [M]. Beijing: Electronic Industry Press, 1996. [黄德双. 神经网络模式识别 [M]. 北京: 电子工业出版社, 1996.]

[3] Tsumoto S, Tanaka H AQ. Rough Sets, and Matroid Theory [A]. *Proc. of RSKD'98* [C]. Japan, 1993, 290—297.

[4] Skowron A. The Rough Sets Theory and Evidence Theory [J]. *Fundamental Informatics*, 1990, **13**: 245—262

[5] Mrozek A. Rough Sets and Some Aspects of Expert Systems Realization [A]. Proc. 7th Int. Conf. on Expert Systems & Their Application[C]. France, 1987.

[6] Zeng H L. Rough Set Theory and Applications [M]. Chongqing: Chongqing University Press, 1996. [曾黄麟. 粗集理论及其应用 [M]. 重庆: 重庆大学出版, 1996.]

[7] Pao Y H. Adaptive Pattern Recognition and Neural Networks [M]. Addison-Wesley Publishing Company Inc., 1989.

[8] Li Y M, Zhu S J. A Rough Set Approach to BP Neural Network Designing [J]. *Theory and Application of System Engineering*, 1999, 19(3): 62—69.

[9] Jolonek J. Rough Set Reduction of Attributes and Their Domains for Neural Networks [J]. *Computational Intelligence*, 1995, **11**(2): 35—39.

[10] Arai M. Mapping Abilities of Three-Layered Neural Networks [A]. *IJCNN*[C], 1989, Vol, **1**, 16—22.

Combining Rough Set Theory with Neural Network for Remote Sensing Image

YU Chun-yan[1,2], WU Ming-hui[3], WU Ming[2]

(1. *Computer College, Zhejiang University, Hangzhou* 310027, *China*;
2. *Department of Computer, Fuzhou University, Fuzhou* 350002, *China*;
3. *Department of Computer Science and Technology, City College, Zhejiang University, Hangzhou* 310027, *China*)

Abstract: There is a gap between traditional neural network's information process ability and amount information of remote sensing image recognition. Focusing on this problem, this paper proposes a method to combine rough set theory with neural network theory and uses it in remote sensing image recognition. First, this paper analyzes the feasibility and advantages of combination of neural network with rough set theory. Based on this analysis, a rough set theory based remote sensing image recognition model is presented. Furthermore, analysis on rough set module and remote sensing image recognition module are given in details. Finally, contrastive experiment data are given to prove that combining rough set theory with neural network theory for remote sensing image recognition has a high converging rate, shorter training time and more accuracy. The potential of this method, is also shown.

Key words: rough set theory; neural network; remote sensing image; image recognition