

## 粗糙集约简的WNN隐层节点优化方法

孟月波<sup>1,2</sup>, 邹建华<sup>1</sup>, 刘光辉<sup>2</sup>, 甘旭升<sup>3</sup>

(1. 西安交通大学 系统工程研究所, 西安 710049; 2. 西安建筑科技大学 信息与  
控制工程学院, 西安 710055; 3. 空军工程大学 工程学院, 西安 710038)

**摘要:** 在确保网络性能的前提下, 如何确定最佳隐层节点, 获得最简网络结构是小波神经网络(WNN)应用推广的关键. 对此, 引入粗糙集理论, 提出了基于信息熵的卡方离散化算法和启发式的属性约简递归算法, 利用粗糙集约简过程对WNN隐层节点进行精简, 并将其应用于飞行器气动力建模. 仿真结果表明, 采用改进的粗糙集方法设计WNN, 不仅能够简化网络结构, 而且与未经结构优化的WNN相比, 其模型精度和训练速度都得到了实质性改善.

**关键词:** 小波神经网络; 粗糙集; 气动力建模

**中图分类号:** TP183

**文献标志码:** A

## Optimization method for hidden layer nodes of WNN based on rough set reduction

MENG Yue-bo<sup>1,2</sup>, ZOU Jian-hua<sup>1</sup>, LIU Guang-hui<sup>2</sup>, GAN Xu-sheng<sup>3</sup>

(1. Systems Engineering Institute, Xi'an Jiaotong University, Xi'an 710049, China; 2. School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China; 3. Engineering College, Air Force Engineering University, Xi'an 710038, China. Correspondent: MENG Yue-bo, E-mail: mengyuebo@163.com)

**Abstract:** Under the premise of ensuring network performance, the key of wavelet neural network(WNN) application and promotion is how to get the most simple network structure by determining the optimal hidden layer nodes. Therefore, a Chi-Square discretization algorithm based on the information entropy and heuristic attribute reduction recursive algorithm is proposed, reduction process of the rough set theory is used to optimize wavelet neural network hidden layer nodes without changing network performance, and an aircraft aerodynamic model is built by modifying wavelet neural network. Simulation results show that WNN optimized by the proposed improved rough set method the can not only simplify the network structure, but also improve model accuracy and training speed.

**Key words:** wavelet neural network; rough set; aerodynamic modeling

## 0 引言

WNN可用于神经网络的所有应用领域, 但其相关理论尚不够成熟, 如何确定最佳隐层节点一直是WNN结构设计的难点和关键, 迄今尚缺少完善的理论依据. 隐层节点个数与问题的要求、输入和输出节点的数目有关, 选取的隐层节点数过多, 会使训练时间过长并导致泛化能力下降, 甚至出现过拟合; 而隐层节点数选取过少, 则可能无法充分学习训练样本, 导致欠拟合, 甚至无法训练. 目前, 一般大都采用反复试凑、增长、删剪或增长删剪相结合的方法实现隐节点数量的优化<sup>[1]</sup>. 李倩等<sup>[2]</sup>提出混合删剪算法, 应用多

重删剪策略简化并确定网络节点结构; Philippe等<sup>[3]</sup>利用网络输出值的敏感度分析获取隐层与输出层之间的权值贡献, 删除贡献较小的权值以及与其相连接的隐含层节点, 达到优化网络结构的目的; Monirul等<sup>[4]</sup>提出了一种自适应合并增长算法, 通过判断隐节点的学习能力来自适应删减或者分裂隐节点, 确定隐节点的数量; 韩红桂等<sup>[5]</sup>在文献[3]的基础上, 在大贡献节点邻近处增加新节点并删剪小贡献节点, 以实现隐节点的动态调整; 张昭昭等<sup>[6]</sup>通过隐节点连接权矩阵的协方差矩阵计算网络信息熵, 度量网络的复杂度, 删除对网络复杂度影响最小的隐节点. 但

收稿日期: 2013-07-26; 修回日期: 2013-11-21.

基金项目: 陕西省自然科学基金项目(2013JM8030, 2012JM8026); 陕西省教育厅专项基金项目(2013JK1091).

作者简介: 孟月波(1979—), 女, 博士生, 从事智能算法的研究; 邹建华(1964—), 男, 教授, 博士生导师, 从事智能算法与图像处理等研究.

是,上述隐含层节点设计方法大都预先根据经验设定网络评价准则,并采用贪婪搜索策略,使得算法适用性较差,而且极易陷入局部最优.另外,近些年还出现了利用网络自身特性选取隐层节点的方法,例如 Skeletonization 算法<sup>[7]</sup>和 Weight-Elimination 算法<sup>[8]</sup>等,但效果都不理想,尤其在网络结构较复杂,隐层节点数过多时,难度更大,适用性也有待研究.因此,在确保网络性能的前提下,如何确定最佳隐层节点,获得最简网络结构是 WNN 应用推广的关键.

针对上述问题,本文引入粗糙集 (RS) 理论,提出了基于信息熵的自动连续属性卡方离散化算法和启发式的属性约简递归算法,实现了在不改变网络性能的前提下,利用粗糙集理论中的约简过程对 WNN 隐层节点进行精简,并将其应用于飞行器气动力建模.仿真结果表明,采用本文提出的改进的粗糙集 (IRS) 方法设计 WNN,不仅能够简化网络结构,模型精度和训练速度也比未经过结构优化的 WNN 得到了实质性改善.

## 1 自动连续属性的卡方离散化算法

目前,出现了许多新的粗糙集的连续属性离散化方法,但这些离散化方法大多依赖于人的主观经验.为更好地解决 WNN 结构优化问题,需选择一种不依赖于经验、自动对连续属性进行离散化的算法.

下面先简述卡方离散化算法,然后从决策表信息熵角度引入一种自动连续属性的卡方离散化算法.

### 1.1 卡方离散化算法

卡方离散化算法 (Chi Square, Chi 2) 是对 Kerber 提出的 ChiMerge 方法的改进<sup>[9]</sup>. Chi 2 算法的基本概念包括区间、断点、统计量  $\chi^2$  和  $\chi^2_\alpha$  以及不一致率 Incon.rate,具体表述如下.

1) 区间与断点. 数据集的区间是属性值的集合,初始区间是将每一个值均作为一个区间.断点可对相邻区间进行区分.连续属性离散化是指在一定准则约束下消除断点,合并相邻区间的过程.

2)  $\chi^2$  和  $\chi^2_\alpha$ . 由皮尔逊定理可知,统计量  $\chi^2$  的渐进分布是自由度为  $k-1$  的  $\chi^2$  分布,即  $\chi^2_{k-1}$  分布.在离散化算法中,需要计算相邻区间的  $\chi^2$  值,其计算方法为

$$\chi^2 = \sum_{i=1}^2 \sum_{j=1}^k \frac{(A_{ij} - E_{ij})^2}{E_{ij}}. \quad (1)$$

其中:  $k$  为决策类别数,将断点前后两间隔分别记为第 1、第 2 间隔;  $A_{ij}$  为  $i$  区中  $j$  类样本的个数;  $E_{ij}$  为  $A_{ij}$  的期望频率,可由下式计算得出:

$$E_{ij} = \begin{cases} R_i \cdot C_j / N, & R_i \cdot C_j \neq 0; \\ b, & R_i \cdot C_j = 0. \end{cases} \quad (2)$$

其中:  $R_i = \sum_{j=1}^k A_{ij}$  为  $i$  区间中样本数;  $C_j = \sum_{i=1}^2 A_{ij}$

为该两间隔中  $j$  类样本个数之和;  $N = \sum_{i=1}^2 R_i$  为断点两边间隔中样本个数之和;  $b$  为常数,可取为 0.1. 当显著性水平为  $\alpha$  时,可确定出统计量  $\chi^2$  对应的临界值  $\chi^2_\alpha$ , 有

$$\int_0^{\chi^2_\alpha} f(x) dx = \alpha,$$

其中  $f(x)$  是自由度为  $k-1$  的  $\chi^2$  分布的概率密度函数.

3) Incon.rate. 在信息系统的条件属性相同而决策属性不相同的情况下,可采用下式表征决策表中分类信息不一致的程度:

$$\text{Incon.rate} = 1 - \gamma_P, \quad (3)$$

其中  $\gamma_P$  是近似精度. 通常利用 Incon.rate 来控制离散化过程中的合并程度和信息丢失.

### 1.2 基于信息熵的卡方离散化算法

Chi 2 算法通过两个阶段进行离散化,并使用 Incon.rate 检验离散化程度,离散效果优于 ChiMerge 算法,但也存在一些不足:

1) Incon.rate 的下限值需预先设定,但目前尚缺乏统一标准,需通过多次试选,增加了工作量;

2) 分类特性反映对样本数据离散化程度的要求,而 Incon.rate 无法全面反映所需的分类特性;

3) 若存在多个属性需要离散化的情况,离散化顺序的不同会使得最终结果不同,Chi 2 算法没有考虑这一点.

针对上述问题,本文基于粗糙集原理,引入信息熵替代 Incon.rate,以更好地反映样本数据的固有特性,避免参数设定导致的大量计算;同时根据样本数据自身的固有特性自动进行离散化,从而得到改进的 Chi 2 算法,这里将其称为基于信息熵的卡方离散化算法.

设信息系统有  $m$  个连续条件属性,分别为  $c_1, c_2, c_3, \dots, c_m$ , 有 1 个表示类别的决策属性,类别数为  $k$ . 采用本文提出的基于信息熵的卡方离散化算法对连续条件属性进行离散化的基本过程如下.

#### Step 1

Step 1.1: 参数初始化. 将  $m$  个属性样本取值点存入断点集  $S$ ; 给定显著性水平初值  $\alpha$  及下降系数初值  $\eta$ ; 确定相应的临界值  $A_\alpha$ ; 根据式 (4) 求取决策系统的信息熵初值  $H_0$ . 计数值  $i$  记为 1.

$$H = \sum_{X_i \in V} \frac{|X_i|}{|U|} \ln \left( \frac{|U|}{|X_i|} \right) + |U - V| \ln(|U|). \quad (4)$$

其中: 条件属性集合  $X_i$  为对象集合  $U$  的一个子集,  $V$  为属性取值集合.

**Step 1.2:** 对于属性  $c_i$ , 根据式 (1) 求取每一断点的  $\chi^2$  值并找到其最小值  $\chi_{\min}^2$ , 判断  $\chi_{\min}^2 < A_\alpha$  是否成立. 若成立, 则删去  $\chi_{\min}^2$  所对应的断点并再次执行 Step 1.2, 继续对该属性进行离散化; 若不成立, 则继续执行 Step 1.3.

**Step 1.3:** 判断  $i < m$  是否成立. 若成立, 则令  $i = i + 1$ , 再次执行 Step 1.2, 对下一个属性进行离散化; 若不成立, 则继续执行 Step 1.4.

**Step 1.4:** 根据式 (4) 求取决策系统信息熵  $H$ , 判断  $H_0 \geq H$  是否成立. 若成立, 则令  $H_0 = H, \alpha = \alpha \cdot \eta$ , 再次计算  $A_\alpha$ , 更新断点集  $S$ , 将现有断点取作新的断点集, 并令  $i = 1$ , 执行 Step 1.2; 若不成立, 则执行 Step 2.

### Step 2

**Step 2.1:** 依据断点集  $S$  恢复各属性断点, 按照式 (5) 求取每个属性  $(c_1, c_2, \dots, c_i, \dots, c_m)$  的重要性和断点平均重要性, 并将其进行升序排列. 计数  $i$  置 1.

$$\sigma_{C \cdot D}(c_i) = \gamma_C(D) - \gamma_{C - \{c_i\}}(D). \quad (5)$$

其中:  $\gamma_C(D)$  表示决策属性支持度;  $\gamma_{C - \{c_i\}}(D)$  表示去掉条件属性  $c_i$  的条件属性集对决策属性集  $D$  的支持度;  $\sigma_{C \cdot D}(c_i)$  表示从条件属性集  $C$  中去掉属性  $c_i$  后对于决策属性集  $D$  的影响程度,  $\sigma_{C \cdot D}(c_i)$  越大, 说明  $c_i$  在整个属性集中的影响程度越大, 重要性越高.

**Step 2.2:** 对于属性  $c_i$ , 进行如下操作:

**Step 2.2.1:** 令  $\alpha = \alpha_0, A_\alpha = A_{\alpha_0}$ , 将该属性现有断点存入  $S_P$ .

**Step 2.2.2:** 根据式 (1) 求取各断点  $\chi^2$  值并找出其最小值  $\chi_{\min}^2$ , 判断  $\chi_{\min}^2 < A_\alpha$  是否成立. 若成立, 则删去  $\chi_{\min}^2$  所对应的断点并再次执行 Step 2.2.2; 若不成立, 则执行 Step 2.3.

**Step 2.3:** 根据式 (4) 计算信息熵  $H$ , 判断  $H_0 \geq H$  是否成立. 若成立, 则令  $H_0 = H, \alpha = \alpha \cdot \eta$ , 再次计算  $A_\alpha$  并更新断点集  $S$ , 执行 Step 2.2; 若不成立, 则根据断点集  $S$  恢复属性  $c_i$  的断点, 判断  $i < m$  是否成立. 若成立, 则令  $i = i + 1$ , 再次执行 Step 2.2, 处理下一个属性; 若不成立, 则算法结束.

## 2 启发式属性约简算法

最小属性约简是在保持知识结构和容量不发生变化的前提下, 获取信息系统所有约简中含属性数目最少的约简. 已经证明, 寻找所有约简和最小约简是一个 NP-hard 问题<sup>[10]</sup>, 因此一些研究者提出了属性约简的启发式算法, 其中大多通过构建知识与信息熵的

关系实现约简<sup>[11-12]</sup>. 在此类算法中, 按相容关系划分得到的相容类一般不构成对论域的划分, 而构成了对论域的覆盖, 即将知识视为关于论域的覆盖, 这显然欠缺合理性.

基于此, 本文引入了一种启发式的属性约简递归算法, 通过加入不同于信息熵的启发性信息缩小问题的搜索空间, 以期获得最优解和近似最优解. 启发式属性约简递归算法实现过程如下.

**Step 1:** 判别决策系统的相容关系. 对于信息系统的条件属性, 若不可分辨集中对象的决策值相同, 则认为该决策系统是相容的; 否则, 认为其是不相容的.

**Step 2:** 计算相容决策系统中全部条件属性的等价关系. 对于任意一个等价关系, 首先计算其相对于决策值的上近似  $|B^*(X)|$  和下近似  $|B_*(X)|$ ; 然后计算下式所示的启发函数值  $\lambda$ :

$$\lambda = (|B^*(X)| - |B_*(X)|) / |U|, \quad (6)$$

比较所求取的所有  $\lambda$ , 保留最小  $\lambda$  所对应的属性作为必选属性.

**Step 3:** 利用下式:

$$U = U - \text{POS}_{\text{red}}\{d\}, \quad (7)$$

并通过样本集合  $U$  的下近似值对  $U$  进行剪枝. 其中  $\text{POS}_{\text{red}}\{d\}$  表示对于约简集合 red 的条件属性, 必可归类到样本集合  $U$  中决策属性  $d$  的对象集合.

**Step 4:** 重复过程 Step 2 和 Step 3, 直至样本集合  $U$  为  $\phi$ , 最终挑选出所需的所有必选属性. 在每次循环过程中, 均将剩余属性与约简集合 red 中的属性整合, 生成新的等价关系.

## 3 基于粗糙集约简的WNN结构设计

典型的3层WNN结构设计常采用自顶向下的方法, 即预先构造一个拥有足够多隐层节点的网络, 然后凭经验进行删减. 通过这种方式得到的网络, 冗余多、耗时长、收敛慢, 泛化能力也存在提升空间. 而粗糙集可在保持知识结构和容量不发生变化的前提下, 利用约简过程删除不必要的知识, 且无需先验知识. 因此, 可考虑应用粗糙集约简理论来优化WNN结构, 以期去除网络结构中的冗余部分, 最终获得简洁高效的网络.

在给定的WNN模型中, 有些隐层节点对误差贡献较小, 对网络变化的影响较小, 而有些隐层节点对误差贡献较大, 可引起网络的大幅波动. 为获得最优的网络结构, 需保留对网络误差贡献大的隐层节点, 删除贡献小的隐层节点, 从而提升网络学习速度, 改善泛化能力. 具体做法如下: 将网络输出误差作为决

策属性,网络各隐层节点输出作为决策表的条件属性,构建一个二维决策表,并对决策表进行约简,消除存在的冗余属性,进而去除网络中对应的隐层节点,获得最优的WNN拓扑结构.

根据以上阐述,利用粗糙集优化设计WNN结构的流程如图1所示.其具体实现步骤如下.

**Step 1:** 确定WNN的初始结构.输入层节点个数由训练和测试样本确定,输出层节点个数由要解决的实际问题确定,隐层节点个数可根据经验公式初步确定.

**Step 2:** 设定相关参数,采用文献[13]中的改进粒子群优化算法对网络进行训练,并记录各样本输入时各隐层节点的输出值以及此时输出层的误差,并生成决策表.

**Step 3:** 利用本文基于信息熵的卡方离散化算法,对决策表中的决策属性和条件属性进行离散化.

**Step 4:** 利用本文启发式属性约简算法,对离散化后的决策表进行约简,去除冗余属性.

**Step 5:** 根据约简后的决策表对WNN网络进行修剪,删减冗余属性所对应的隐层节点及其连接权.

**Step 6:** 整理得到一个连接权与节点均没有冗余的WNN结构.

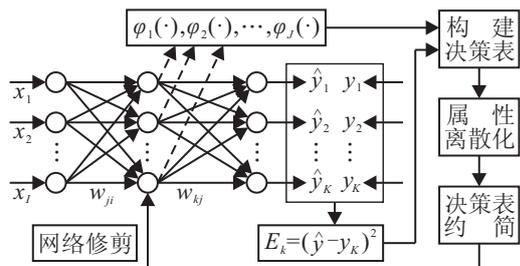


图 1 基于粗糙集的WNN结构设计流程

### 4 Hermit 函数逼近测试

Hermit 函数逼近问题如下:

$$F(x) = 1.1(1 - x + 2x^2) \exp\left(-\frac{x^2}{2}\right), \quad (8)$$

其中  $x \in R$ . 训练样本产生方式如下: 样本输入  $x$  服从  $[-4, 4]$  内的均匀分布, 相应的样本输出为  $F(x)$ , 样本总数取 300, 其中前 200 为训练样本, 余下的为测试样本. 隐层节点数预先取为 10, 连接权值、平移因子和伸缩因子的初始值取  $[-1, 1]$  内随机数.

根据图 1 的设计流程, 利用粗糙集优化 WNN 结构. 将训练样本输入 WNN 进行训练, 构建初始决策

信息如表 1 所示. 对此决策表进行离散化处理, 离散化的决策表如表 2 所示.

表 1 初始决策表

样本编号	$\varphi_1$	$\varphi_2$	$\varphi_3$	$\dots$	$\varphi_{10}$	$E$
1	0.000 0	0.717 6	0.001 7	$\dots$	0.000 1	0.000 1
2	0.000 0	0.961 4	-0.048 8	$\dots$	0.001 1	0.000 6
3	0.000 0	0.573 7	0.000 1	$\dots$	0.000 0	0.000 2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
200	0.000 0	0.644 4	0.000 5	$\dots$	0.000 0	0.000 1

表 2 离散化后的决策表

样本编号	$\varphi_1$	$\varphi_2$	$\varphi_3$	$\dots$	$\varphi_{10}$	$E$
1	9	18	20	$\dots$	17	13
2	9	25	12	$\dots$	22	29
3	9	11	17	$\dots$	15	23
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
200	9	14	18	$\dots$	13	18

对离散化后的决策表进行属性约简, 约简结果为

$$\text{red} = \{\varphi_3, \varphi_4, \varphi_5, \varphi_7, \varphi_9\}.$$

从约简结果可以看出, 节点  $\varphi_1, \varphi_2, \varphi_6, \varphi_8, \varphi_{10}$  为冗余结点, 应予以修剪. 修剪后的WNN拓扑结构如图 2 所示.

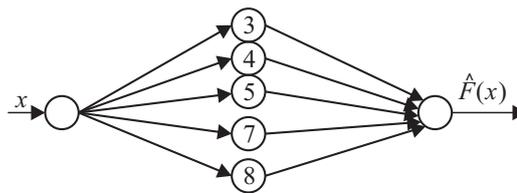


图 2 修剪后的WNN拓扑结构

用图 2 所示的网络结构对原样本重新训练, 得到的结果如表 3、图 3 和图 4 所示. 其中: 表 3 给出了分别由 Skeletonization 法、Weight-Elimination 法和本文改进的粗糙集 (IRS) 法设计的 WNN 训练和测试性能对比; 图 3 为使用不同方法设计的 WNN 训练收敛曲线; 图 4 为本文 IRS 法设计的 WNN 模型逼近效果.

由上述结果可得到如下结论: 采用本文 IRS 法优化后的 WNN 结构大大简化, 收敛性能有所提高, 但相对其他方法并不显著; 网络结构精简给 WNN 模型带来了两个好处, 一是求解规模缩小, 迭代一次所需时间相对减少, 总训练时间也明显减少, 二是消除了网

表 3 基于不同设计方法的WNN性能对比

结构设计方法	网络隐节点	训练MSE	测试MSE	训练时间/s
未优化结构	1~10	0.000 5	0.010 7	112.0
Skeletonization 法	3, 4, 5, 6, 8, 9	0.004 5	0.012 3	72.4
Weight-Elimination 法	3, 4, 6, 7, 8, 9, 10	0.006 0	0.009 2	86.5
IRS 法	3, 4, 5, 7, 9	0.000 7	0.008 7	54.1

络中存在的冗余, 测试精度也得到了一定程度的改善; 另外, 相对于基于WNN自身特性的 Skeletonization法和 Weight-Elimination法而言, IRS法更易于实现, 优化设计效果更好。

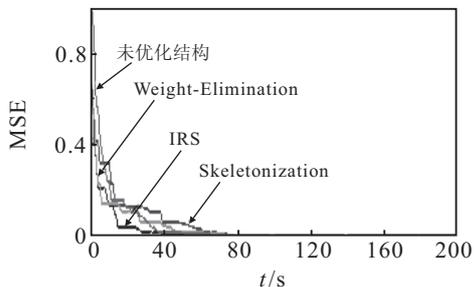


图3 基于不同设计方法的WNN收敛曲线对比

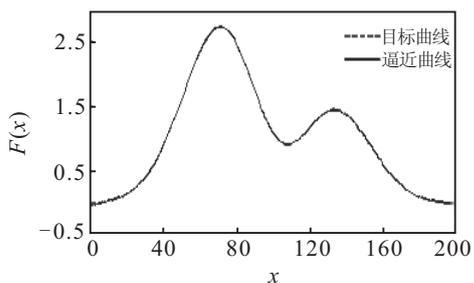


图4 IRS法设计的WNN模型逼近曲线

### 5 飞行器气动力建模实验分析

气动力建模的准确性是飞控系统设计的基础, 也是飞行性能分析和半实物仿真的关键. 本文WNN气动力建模实验分为横侧向建模和纵侧向建模2个实验, 前者为空中验证机 ATTAS 在 12 000 英尺的副翼输入产生的机动动作 (626 个训练样本); 后者为 HFB-320 空中飞行模拟机在 13 800 英尺的升降舵输入产生的机动动作 (200 个训练样本). 横侧向、纵侧向 WNN 气动力模型可分别表示为

$$[C_Y, C_l, C_n] = f_{lat}(\beta, p, r, \delta_a, \delta_r), \quad (9)$$

$$[C_D, C_L, C_m] = f_{lon}(\alpha, q, V, \delta_e). \quad (10)$$

其中:  $C_Y, C_l, C_n$  代表横侧向气动力系数和力矩系数 (侧力系数、滚转力矩系数、偏航力矩系数), 为横侧向 WNN 输出;  $C_D, C_L, C_m$  代表纵侧向气动力系数或力

矩系数 (阻力系数、升力系数、俯仰力矩系数), 为纵侧向 WNN 输出;  $\beta, p, r, \delta_a, \delta_r$  为横侧向 WNN 输入,  $\alpha, q, V, \delta_e$  为纵侧向 WNN 输入,  $p, q, r$  表示倾斜、俯仰和偏航角速率,  $\alpha, \beta$  表示攻角和侧滑角,  $\delta_e, \delta_a, \delta_r$  表示升降舵、副翼和方向舵偏角,  $V$  表示质心真速.

具体气动力建模实验流程和数据流向如图 5 所示, 虚线部分为本文重点.

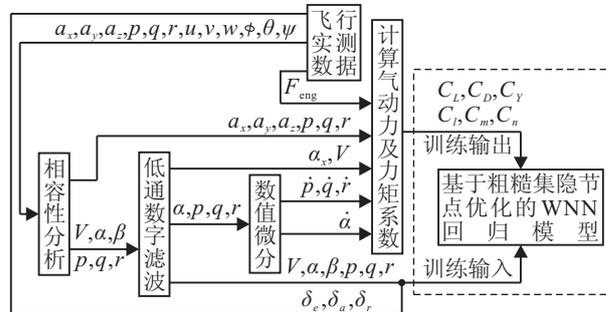


图5 气动力建模实验流程和数据流向示意图

从建模流程中可看出, 实验需进行相容性分析、低通数字滤波、数值微分等预处理. 另外, 与风洞数据的气动力建模过程不同, 使用飞行数据建立 WNN 气动力模型时, 训练输出  $C_Y, C_l, C_n, C_D, C_L, C_m$  无法直接测量得到, 必须通过计算得出. 以上处理过程因篇幅所限不再详述, 具体可参见文献 [13].

在实验中, 隐层节点取为 20, 其他参数设置同 Hermit 函数逼近测试算例. 为消除量纲的影响, 建模前需对数据进行标准化处理. 因为 Skeletonization 和 Weight-Elimination 算法在处理这种多输入多输出的网络结构时过程过于复杂, 结果也很不稳定, 本实验中并未与其进行对比.

从表 4 可得出与第 4 节相同的结论, 利用 IRS 法设计的 WNN 在训练时间和测试精度方面明显优于未优化结构的 WNN. 这也说明, 先利用 IRS 法去除网络的冗余隐层节点, 再进行 WNN 气动力建模预测是可行的, 不仅优化了 WNN 结构, 而且改善了 WNN 建模性能, 同时也为解决气动力建模中的 WNN 结构优化问题提供了有益经验.

表4 气动力建模的实验结果对比

实验	气动系数	未优化结构 WNN		由 ISR 优化结构 WNN		
		预测 MSE	训练时间/s	red	预测 MSE	训练时间/s
横侧向	$C_Y$	0.019 7		1, 3, 4	0.011 5	
	$C_l$	0.016 5	1 064.7	7, 8, 15	0.009 5	426.2
	$C_n$	0.005 5		16, 17	0.008 2	
纵侧向	$C_D$	0.002 4		4, 5, 6, 7	0.001 3	
	$C_L$	0.026 6	352.6	10, 14, 15	0.011 0	132.5
	$C_m$	0.006 3		17, 18	0.005 4	

## 6 结 论

粗糙集通过分析集合间的不可分辨关系对知识进行分析和约简,进而删除与决策属性无关的冗余条件属性. 本文将粗糙集理论引入 WNN 结构设计,在不改变网络性能的前提下,利用粗糙集理论中的约简过程对 WNN 隐层节点进行精简. 基于处理过程的实际问题,分别引入基于信息熵的卡方离散化算法和启发式约简算法,以提升粗糙集优化设计 WNN 结构的便捷性和有效性. Hermit 函数逼近测试以及飞行器气动力建模实验结果表明,使用本文改进的粗糙集方法设计 WNN,不仅能够简化网络结构,模型精度和训练速度也比未经结构优化的 WNN 有了实质性改善.

### 参考文献(References)

- [1] 张昭昭, 乔俊飞, 杨刚. 自适应前馈神经网络结构优化设计[J]. 智能系统学报, 2011, 6(4): 312-316.  
(Zhang Z Z, Qiao J F, Yang G. An adaptive algorithm for designing optimal feed-forward neural network architecture[J]. CAAI Trans on Intelligent Systems, 2011, 6(4): 312-316.)
- [2] 李倩, 王永县, 朱友芹. 人工神经网络混合剪枝算法[J]. 清华大学学报: 自然科学版, 2005, 45(6): 831-834.  
(Li Q, Wang Y X, Zhu Y Q. Hybrid pruning algorithm for artificial neural network training[J]. J of Tsinghua University: Science and Technology, 2005, 45(6): 831-834.)
- [3] Philippe Lauret, Eric Fock, Thierry Alex Mara. A node pruning algorithm based on a fourier amplitude sensitivity test method[J]. IEEE Trans on Neural Networks, 2006, 17(2): 273-293.
- [4] Monirul Islam, Abdus Sattar, Faijul Amin, et al. A new adaptive merging and growing algorithm for designing artificial neural networks[J]. IEEE Trans on Systems, Man and Cybernetics — Part B: Cybernetics, 2009, 39(3): 705-722.
- [5] 韩红桂, 李淼, 乔俊飞. 基于模型输出敏感度分析的动态神经网络结构设计[J]. 电子学报, 2010, 38(3): 731-736.  
(Han H G, Li M, Qiao J F. Design of dynamic neural network based on the sensitivity analysis of model output[J]. Acta Electronica Sinica, 2010, 38(3): 731-736.)
- [6] 张昭昭, 乔俊飞, 韩红桂. 一种基于神经网络复杂度的修剪算法[J]. 控制与决策, 2010, 25(6): 821-830.  
(Zhang Z Z, Qiao J F, Han H G. A pruning algorithm based on neural complexity[J]. Control and Decision, 2010, 25(6): 821-830.)
- [7] 魏海坤. 神经网络结构设计的理论与方法[M]. 北京: 国防工业出版社, 2005: 228-233.  
(Wei H K. The theory and method of neural network structure design[M]. Beijing: National Defense Industry Press, 2005: 228-233.)
- [8] Weigend A S, Rumelhart D E, Huberman B A. Generalization by weight-elimination applied to currency exchange rate prediction[C]. Proc of the Int Joint Conf on Neural Networks. Seattle, 1991: 837-841.
- [9] Liu H, Setiono R. Feature selection via discretization[J]. IEEE Trans on Knowledge and Data Engineering, 1997, 9(4): 642-645.
- [10] Wong S K M, Ziarko W. On optimal decision rules in decision tables[J]. Bulletin of the Polish Academy of Sciences, 1985, 33(11/12): 693-696.
- [11] 吴静, 邹海. 基于属性重要性的属性约简算法[J]. 计算机应用与软件, 2010, 27(2): 255-257.  
(Wu J, Zou H. Attribute reduction algorithm based on importance of attribute value[J]. Computer Applications and Software, 2010, 27(2): 255-257.)
- [12] 任小康, 吴尚智, 马如云. 基于可辨识矩阵的属性频率约简算法[J]. 兰州大学学报, 2007, 43(1): 138-140.  
(Ren X K, Wu S Z, Ma R Y. Attribute reduction algorithm based on discernibility matrix[J]. J of Lanzhou University, 2007, 43(1): 138-140.)
- [13] Meng Yue-bo, Zou Jian-hua, Gan Xu-sheng, et al. Research on WNN aerodynamic modeling from flight data based on improved PSO algorithm[J]. Neurocomputing, 2012, 83(4): 212-221.

(责任编辑: 曹洪武)