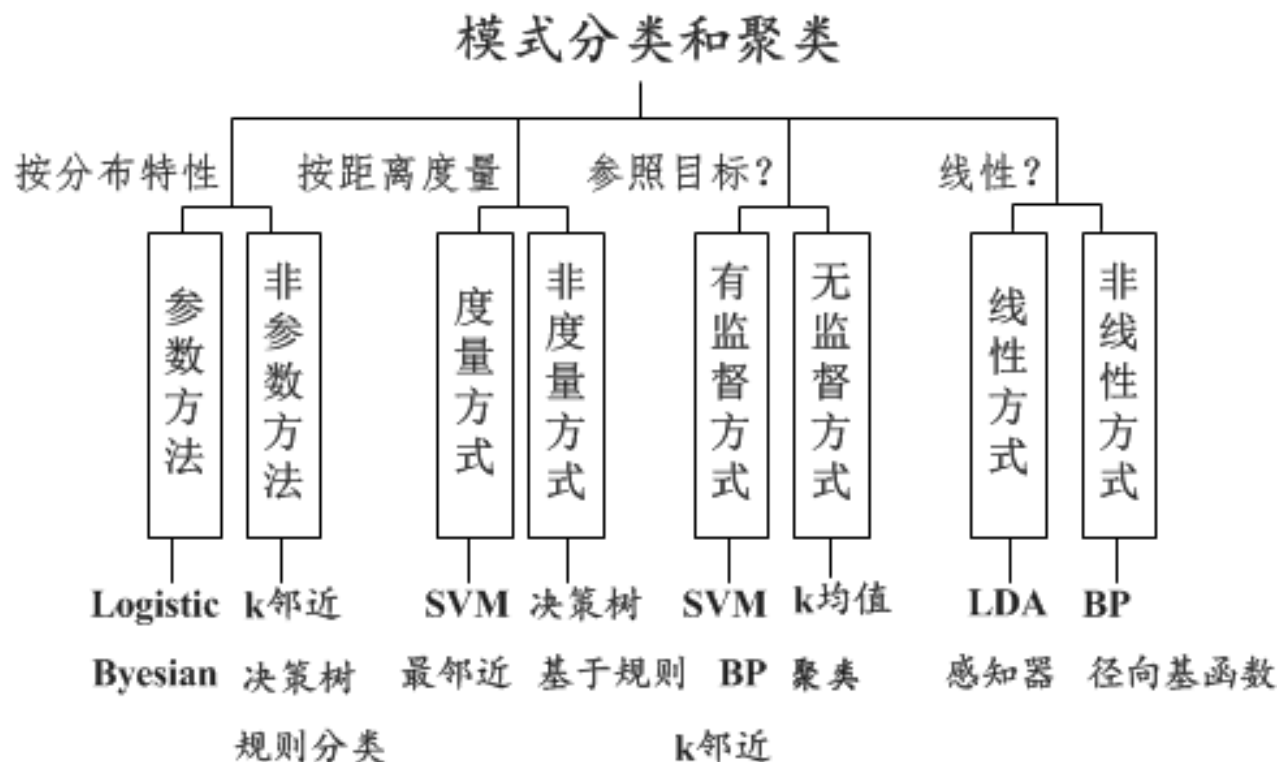


模糊分类与聚类

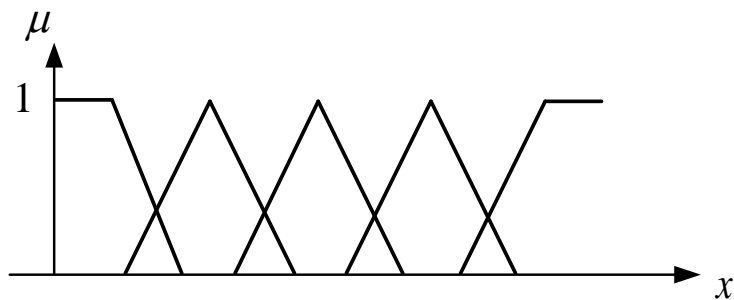
模糊分类与聚类

模式识别方法的选择取决于对象的特征和问题的性质，模糊分类采用模糊数学语言对事物按照**规则**或**隶属度**进行描述和分类。



模糊分类

非参数分类方法——以类别隶属度表达特征属于某一类别的程度



若以颜色的深浅程度表达的分类任务，可分为五类——暗、中暗、中、中淡和淡，不同类别以闭区间上的不同隶属度值衡量.

需要注意的是 表示在 $[0,1]$ 上的特征隶属度，是特征性状的程度，即属于某一特征性状的“资格”，并不包含“发生”的概率或可能性.

模糊分类

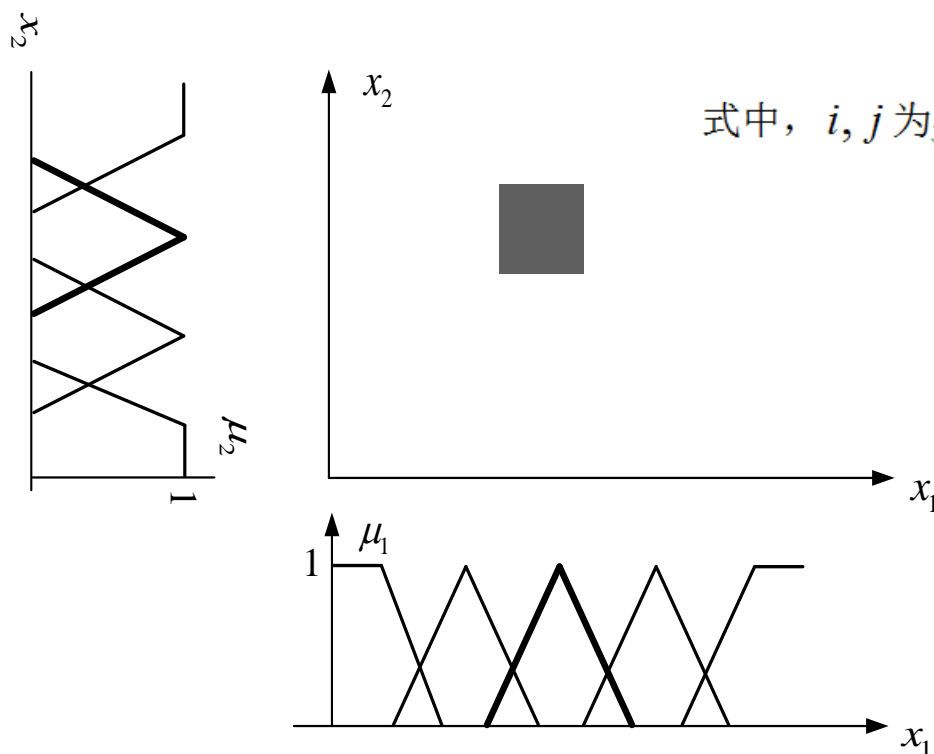
推理过程

若对两特征任务 x_1, x_2 进行模糊分类，特征的类别隶属度

分别为 $\mu_i(x_1)$ ， $\mu_j(x_2)$ ，可采用乘积推理规则

$$\mu_i(x_1) \cdot \mu_j(x_2)$$

式中， i, j 为各特征的 x_1, x_2 类别数， $i = 1, \dots, n, j = 1, \dots, m$



基于经验规则而不具有自学习的特征

模糊分类简单实用，方法的局限性限制了其应用，主要在于：

- 可用于设计分类规则的信息相当有限，例如类别隶属度的数量、宽度及位置等等；
- 当特征数较多或高维情况时，模糊分类方法不再适用；
- 模糊分类不适用于学习率变化的情况；
- 模糊分类不具有自适应性能。

基于规则的分类

*If-Then*规则

一般地，模式分类中*If-Then*规则形如

If 色泽(x1) 青绿 And 根蒂(x2) 蜷缩 And 敲声 (x3)浊响 Then (x)好瓜

其中，色泽x1、根蒂x2和敲声x3为对象x的若干属性，若x同时满足色泽青绿、根蒂蜷缩和敲声浊响的前提特征条件，则可得出结论x为好瓜（周志华，2016）。

模糊*If-Then*规则

若给出对象x的特征属于某一属性的隶属程度，则分类依照

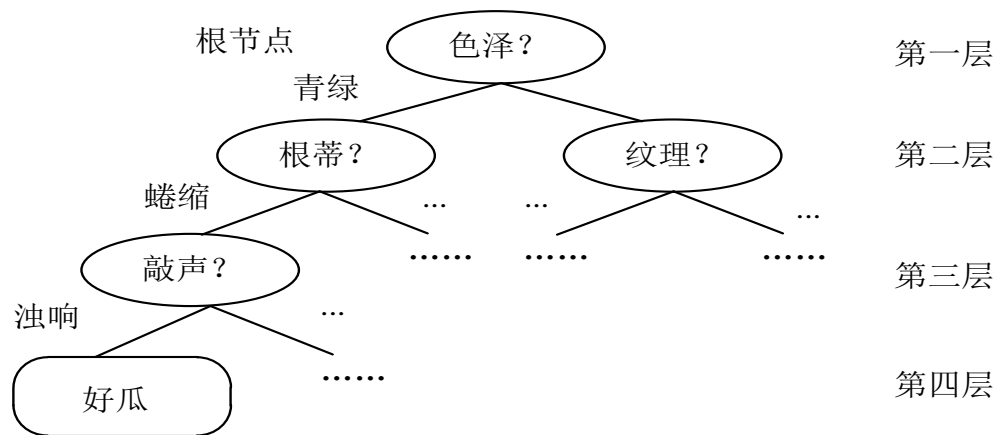
If 色泽(x1=0.80) 青绿 And 根蒂 (x2=0.75)蜷缩 And 敲声 (x3=0.90) 浊响

Then (x=0.95) 好瓜

其中，属性x1,x2,x3分别赋予隶属度值为0.80, 0.75, 0.90，则由一定的规则，可推导出对象x属于结论论域——好瓜的隶属程度. 这里隶属度包含的数值表达了一定的度量，因而是一种度量式分类方式。

基于规则的分类

决策树



- 决策树的生成是一个以规则信息来划分属性的递归过程；
- 与基于规则的分类不同在于，前者对规则的提取是一次完成的，判断与分类同时操作并获得结果，后者则通过顺序比对逐个进行而得出结论；
- 由于规则的可解释性较明确，能够帮助人们直观地了解判别过程；
- 同时，由于规则引入了人类经验和领域知识，因而在逻辑规则的抽象描述能力上更能够体现人工智能处理问题的优越性；
- 由于缺乏更多分布或概率信息，当样本噪声较大时，则不再适用。

分类规则的提取与设计需满足：

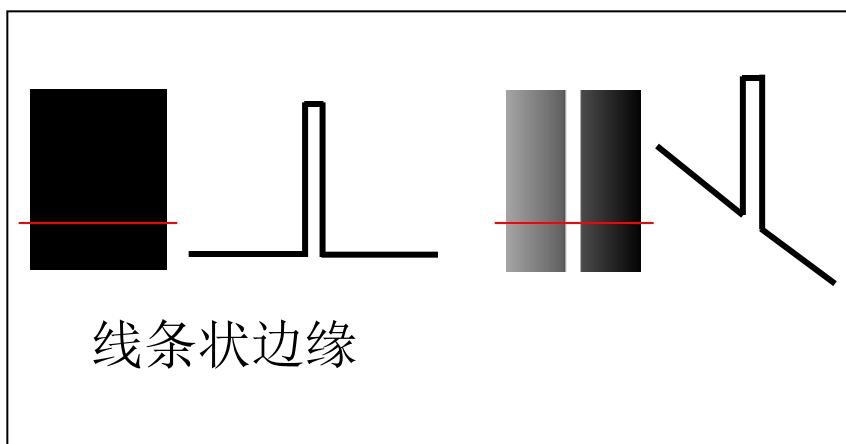
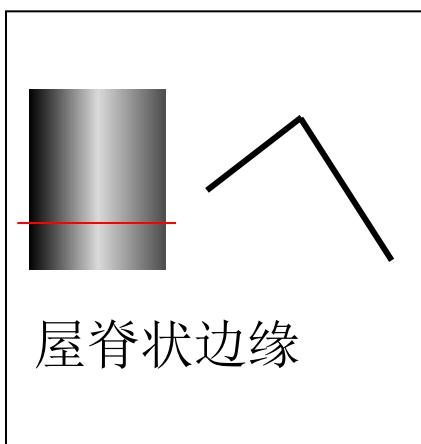
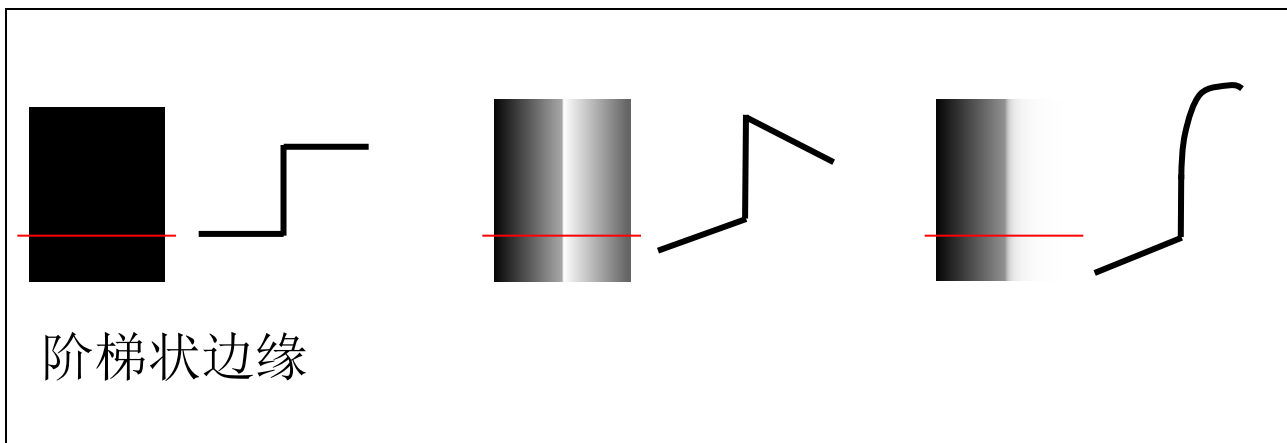
- 互斥规则（Mutually exclusive rules）：每一个样本记录最多只能触发一条规则，规则集中不存在两条规则被同一条记录覆盖的情况，则称规则集是互斥规则；
- 穷举规则（Exhaustive rules）：每一个样本记录至少触发一条规则，如果对属性值的任一组合，都存在一条规则可以覆盖该情况，则称规则集是穷举规则。

【例】基于模糊逻辑的图像边缘提取

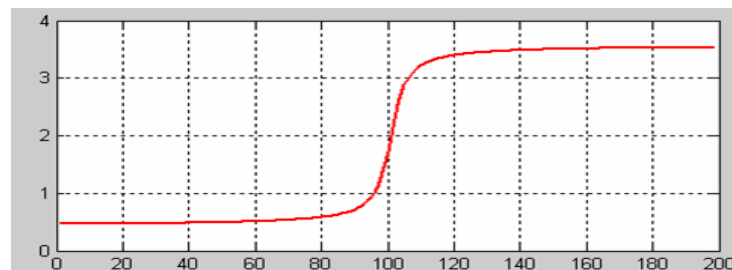
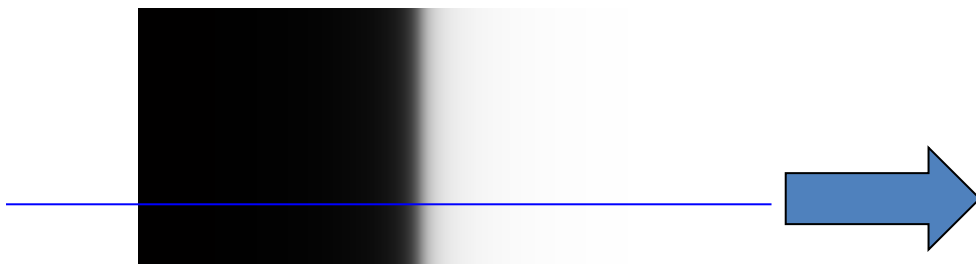
1. 图像边缘简介
2. 经典边缘提取方法
3. 基于模糊逻辑的边缘提取方法

1. 边缘的定义

图像边缘是像素灰度存在阶跃变化或屋脊状变化的像素的集合。

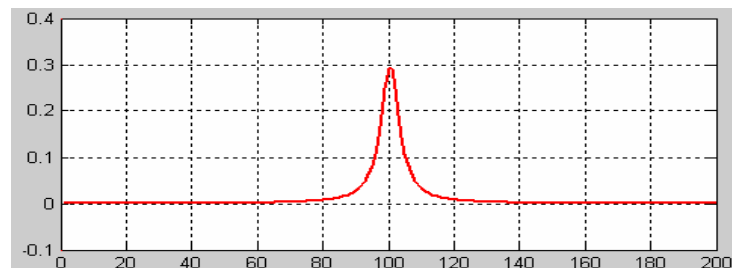


2. 经典边缘提取的核心思想



一阶导数的极大值点：

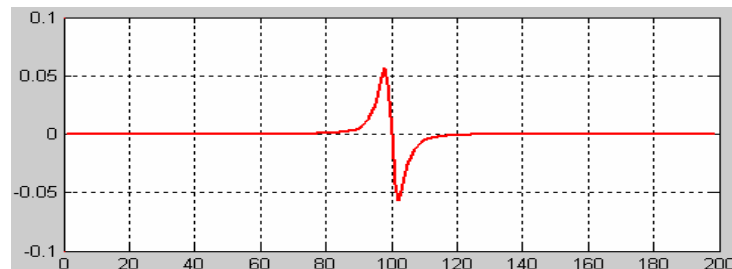
$$Edge = \{x \mid x = \arg \max(f'(x))\}$$



二阶导数的过零点：

$$Edge = \{x \mid f''(x) = 0, \text{ zero crossings}\}$$

注意：仅仅等于0不够，常数函数也为0，必须存在符号改变



更简化的规则：

像素梯度为0的区域，为图像的平坦区域

像素梯度不为0的区域，为图像的边缘区域

——模糊逻辑中的知识推理规则

3. 基于模糊逻辑的图像边缘提取

3.1 模糊系统分析

1. 系统输入：灰度图像两个方向梯度
2. 系统输出：是否为图像边缘
3. 模糊化

输入模糊子集：

{zero}

输入论域为 $[-1 \ 1]$

隶属度函数：高斯

输出模糊子集：

{black (图像边缘), white (不是图像边缘)}

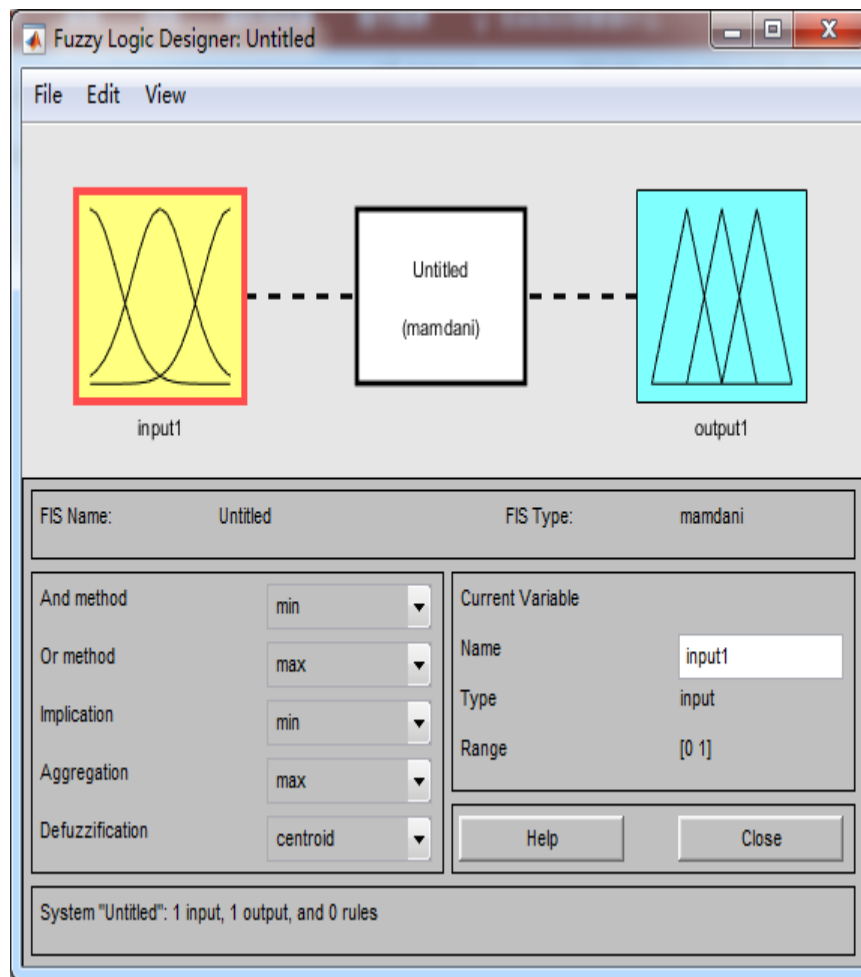
输出论域 $[0 \ 1]$

隶属度函数：三角形

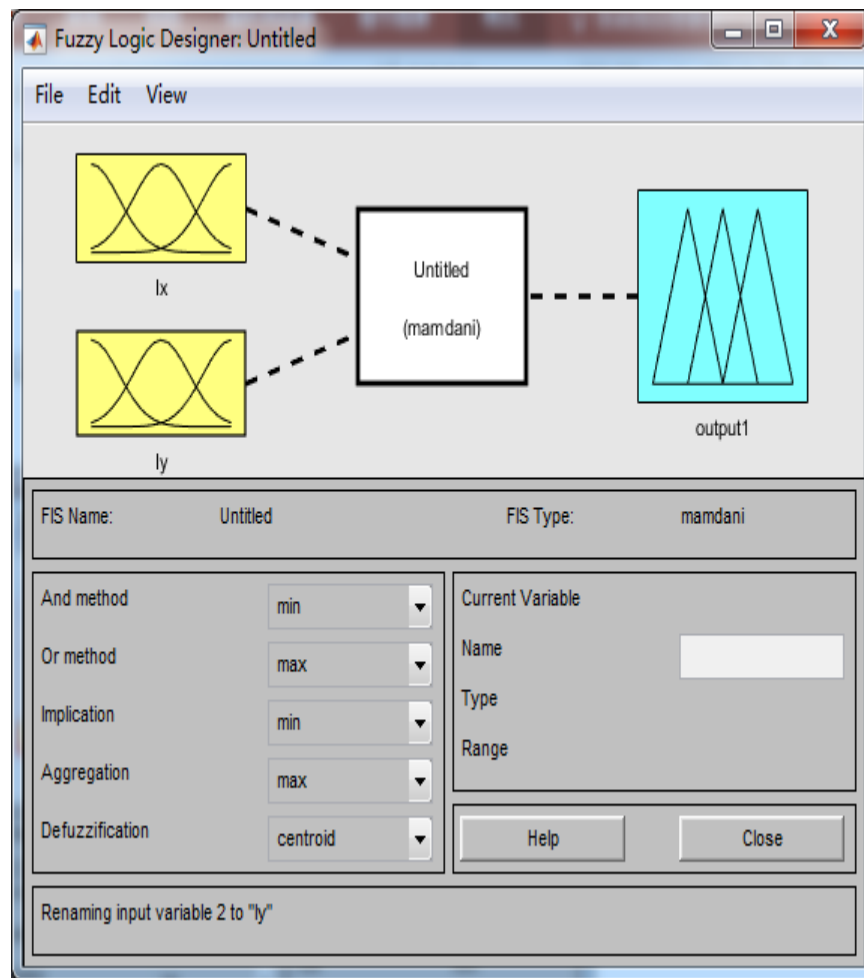
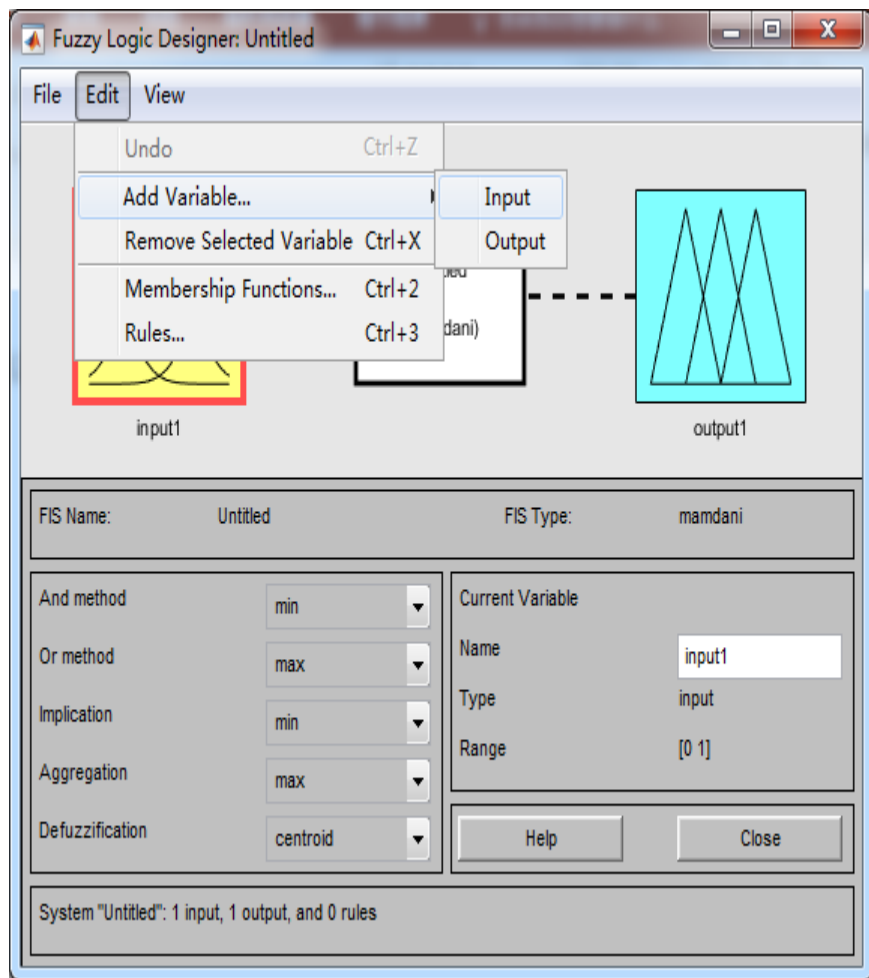
4. 模糊推理：规则库如上述规则
5. 清晰化：加权平均法（重心法）

3.2 利用simulink搭建

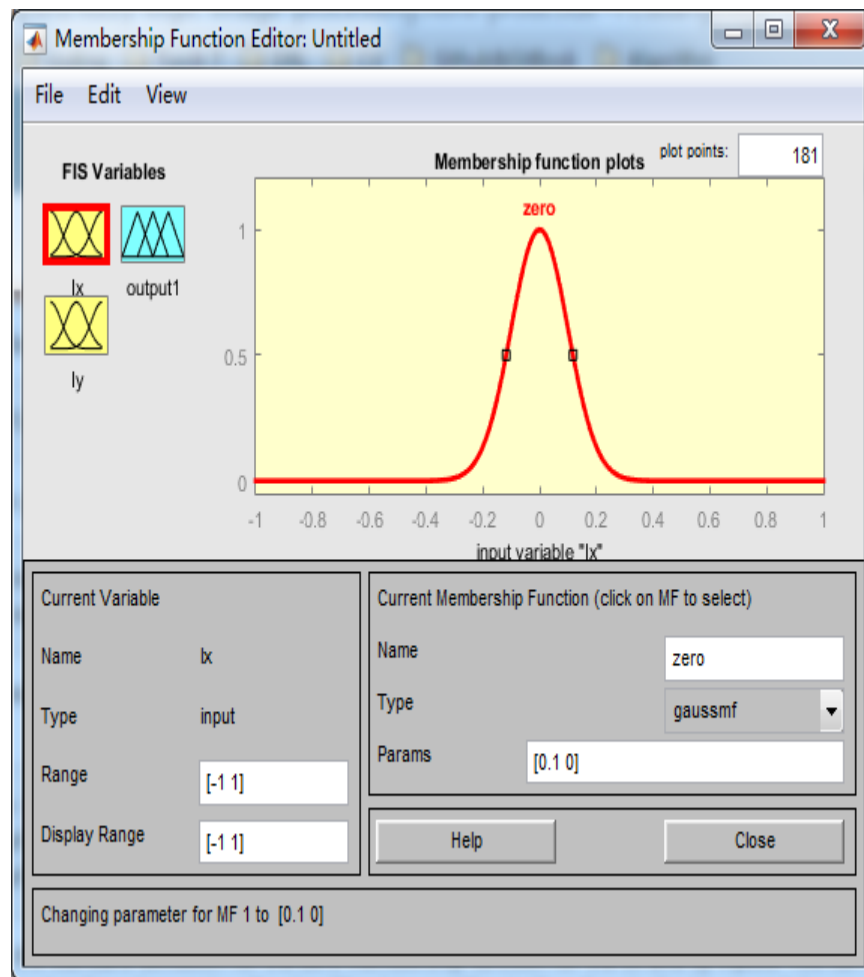
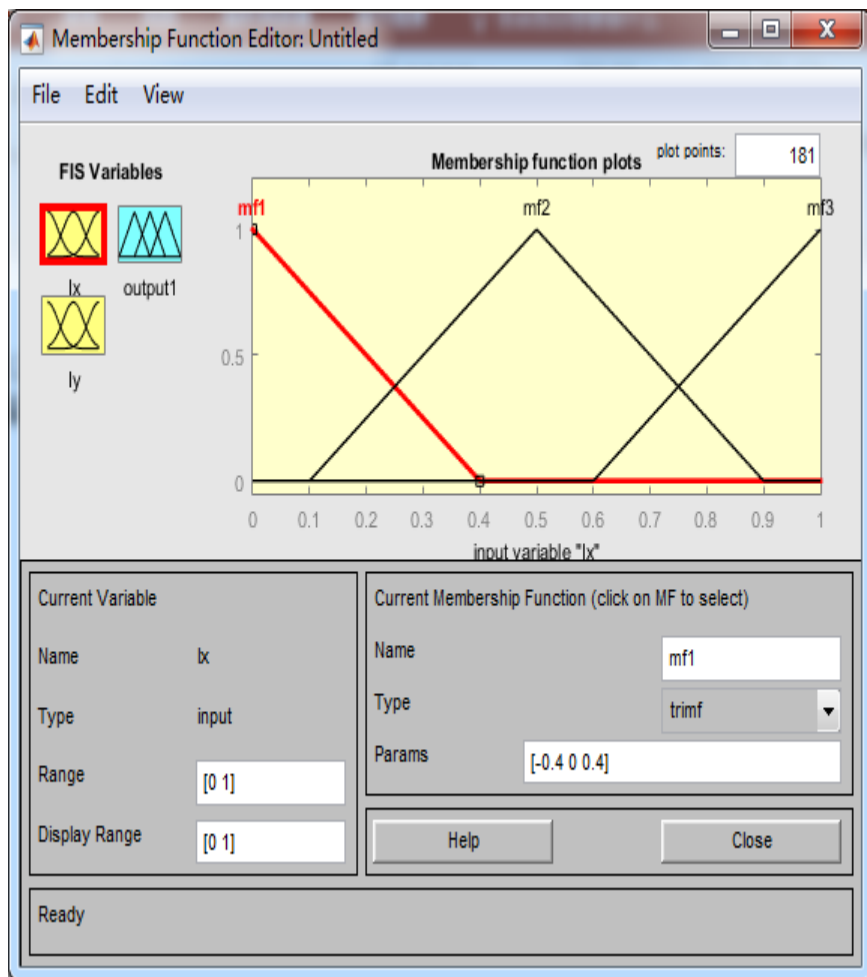
>> fuzzy



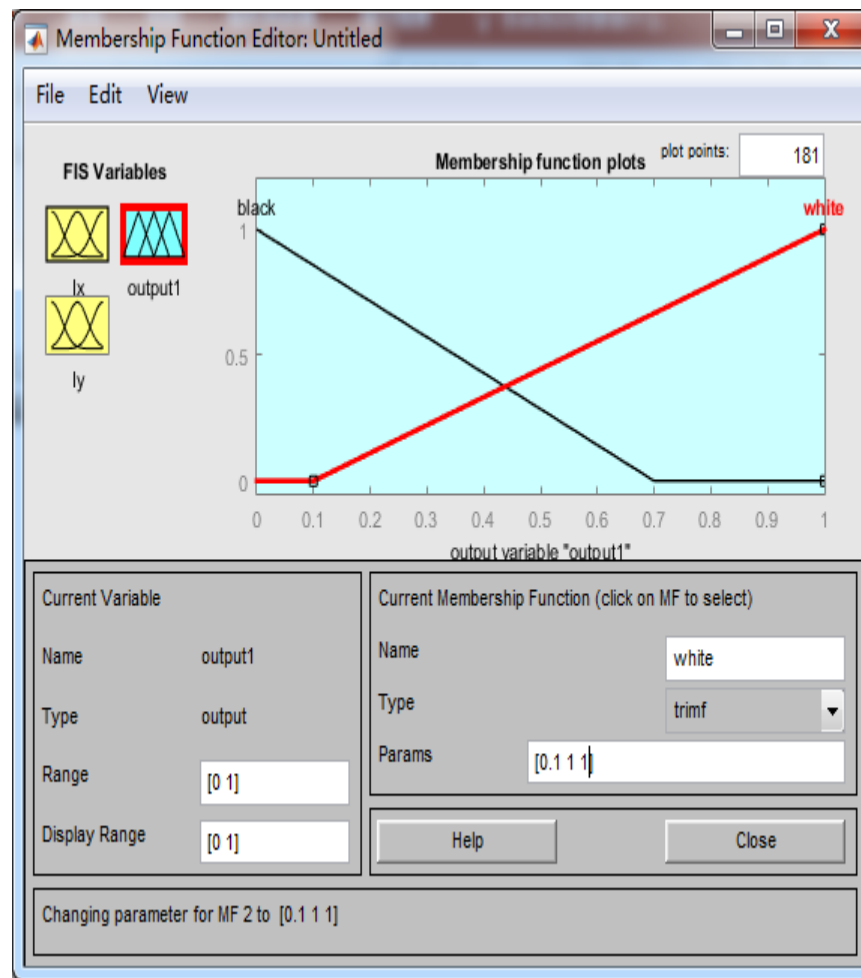
3.2 利用simulink搭建 添加输入



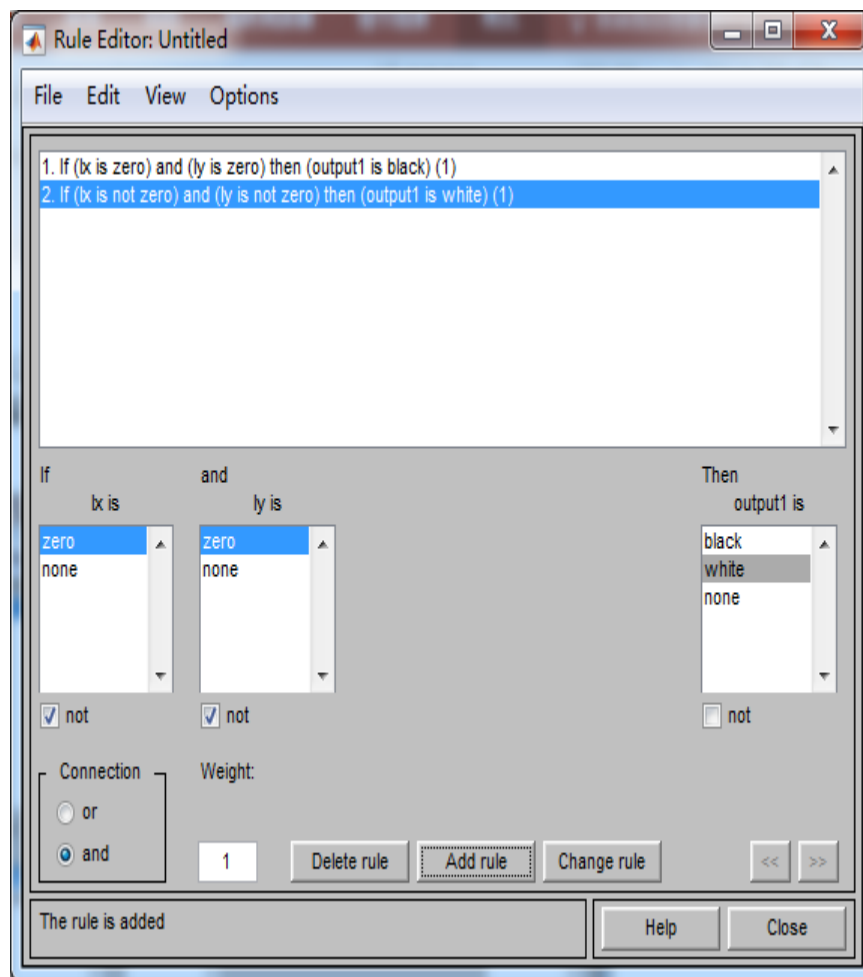
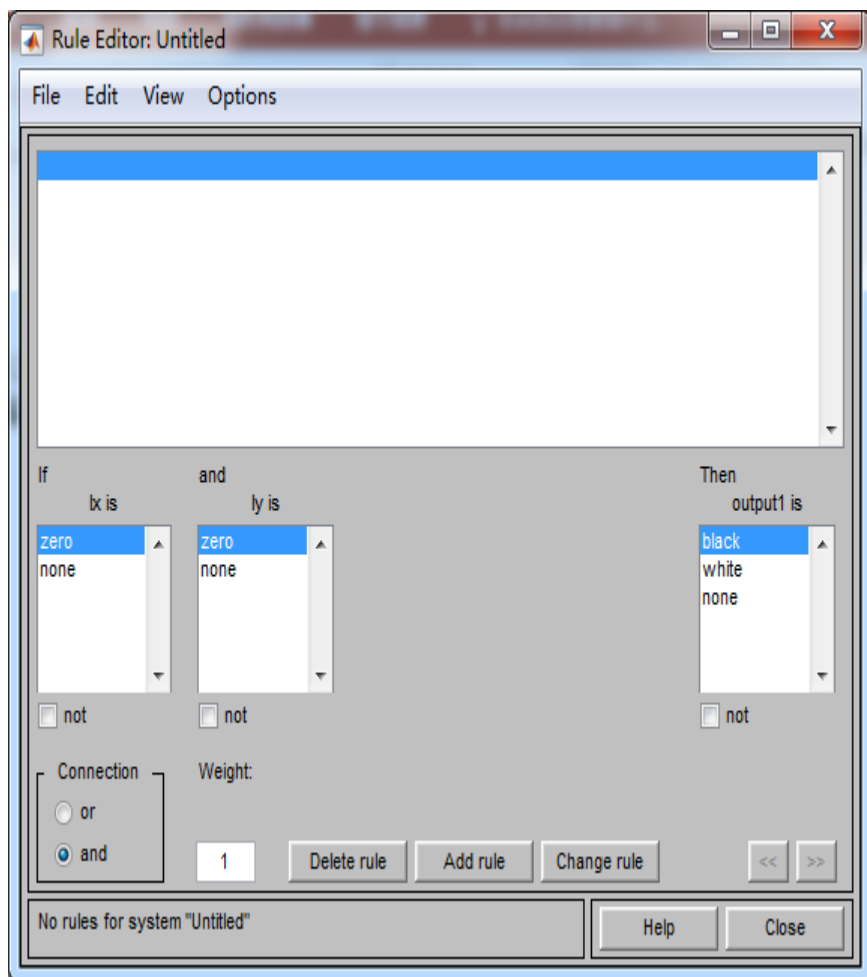
3.2 利用simulink搭建 编辑输入



3.2 利用simulink搭建 编辑输出



3.2 利用simulink搭建 编辑模糊规则



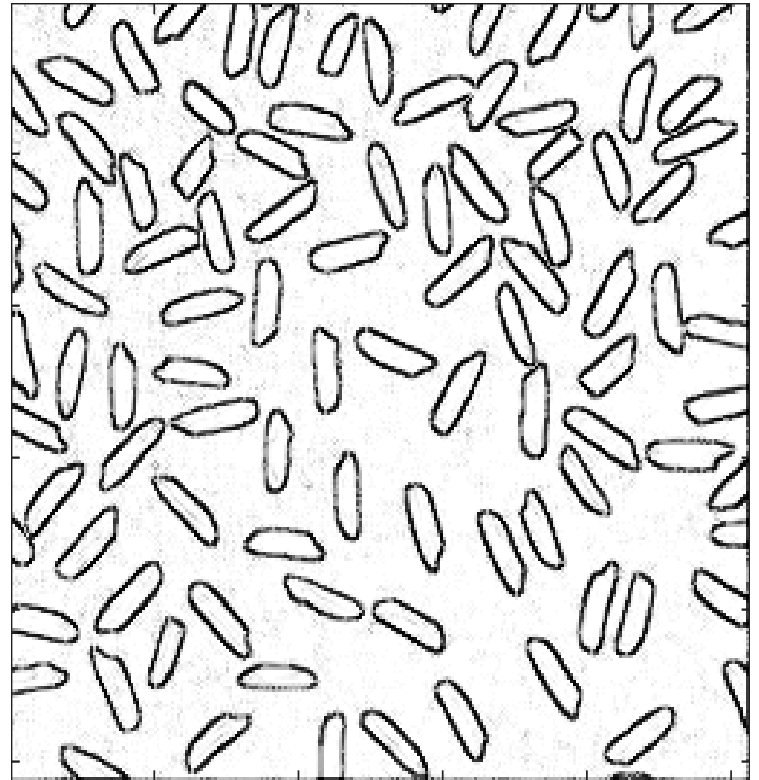
3.3 利用.m搭建

```
edgeFIS = newfis('edgeDetection'); % 建立一个名为edgeDetection的模糊控制器
edgeFIS = addvar(edgeFIS,'input','Ix',[-1 1]); % 添加输入变量
edgeFIS = addvar(edgeFIS,'input','Iy',[-1 1]);
sx = 0.1; sy = 0.1; % 定义标准差
edgeFIS = addmf(edgeFIS,'input',1,'zero','gaussmf',[sx 0]); % 编辑输入变量的隶属度函数
edgeFIS = addmf(edgeFIS,'input',2,'zero','gaussmf',[sy 0]);
edgeFIS = addvar(edgeFIS,'output','Iout',[0 1]); % 添加输出变量
wa = 0.1; wb = 1; wc = 1; % 定义输出变量隶属度函数参数
ba = 0; bb = 0; bc = .7;
edgeFIS = addmf(edgeFIS,'output',1,'white','trimf',[wa wb wc]); % 编辑输出变量的隶属度函数
edgeFIS = addmf(edgeFIS,'output',1,'black','trimf',[ba bb bc]);
r1 = 'If Ix is zero and Iy is zero then Iout is white';
r2 = 'If Ix is not zero or Iy is not zero then Iout is black';
r = char(r1,r2); % 创建规则字符矩阵
edgeFIS = parsrule(edgeFIS,r); % 添加规则
```

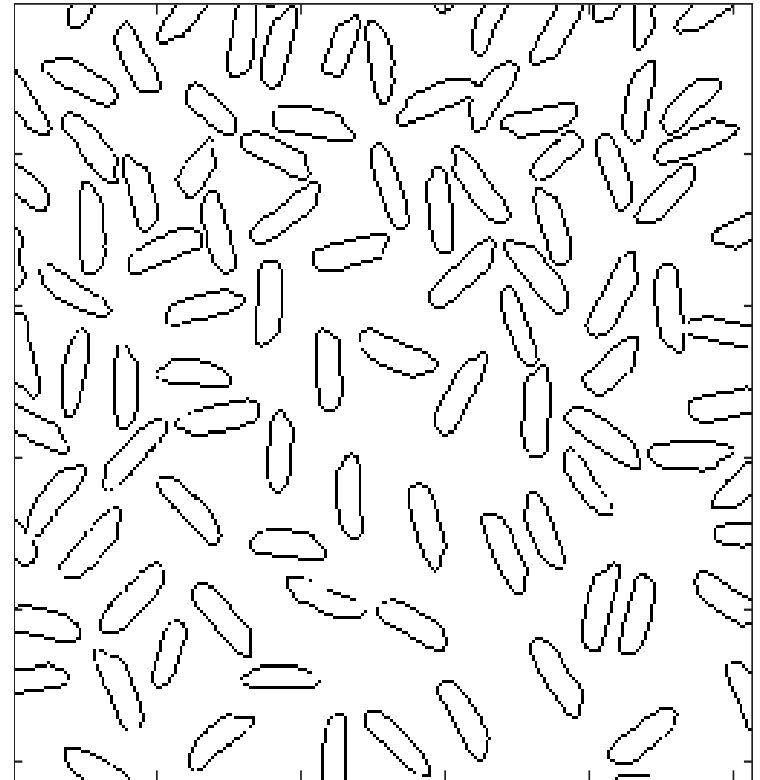
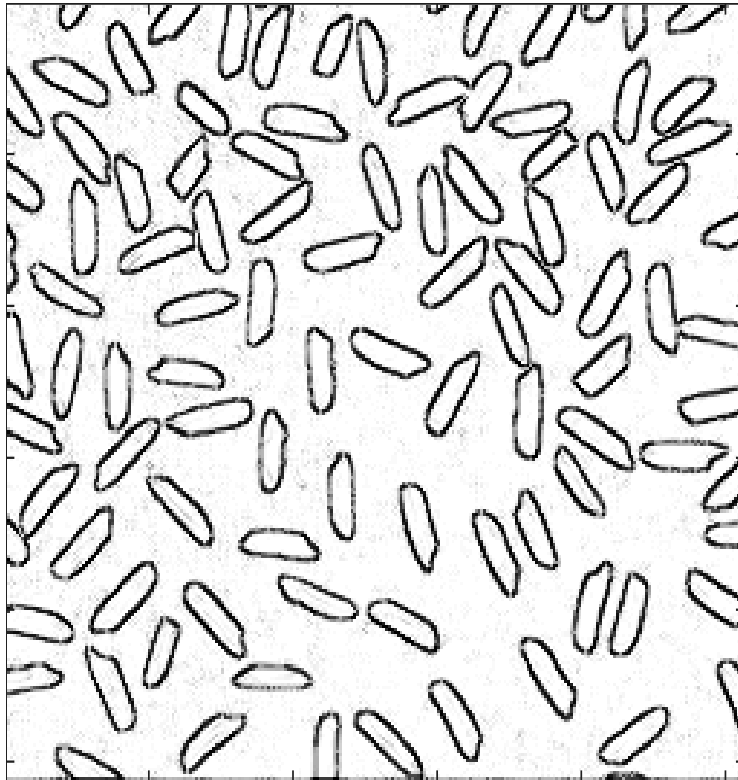
3.3 边缘提取的实现

```
Igray = imread('rice.jpg'); % 图像读入并显示
figure; image(Igray,'CDataMapping','scaled');
colormap('gray');
title('Input Image in Grayscale')
I = im2double(Igray);
Gx = [-1 0 1]; % 梯度算子
Gy = Gx';
Ix = conv2(I,Gx,'same'); % 卷积进行梯度计算
Iy = conv2(I,Gy,'same');
figure; image(Ix,'CDataMapping','scaled'); colormap('gray');
title('Ix');
figure; image(Iy,'CDataMapping','scaled'); colormap('gray');
title('Iy');
edgeFIS = readfis('Image_detect.fis'); % 将已建立好的模糊控制器
读入
Ieval = zeros(size(I));% 输出矩阵
for ii = 1:size(I,1)
    Ieval(ii,:) = evalfis([(Ix(ii,:));(Iy(ii,:))],'edgeFIS');
% 进行模糊推理
end
figure; image(I,'CDataMapping','scaled'); colormap('gray');
title('Original Grayscale Image')
figure; image(Ieval,'CDataMapping','scaled');
colormap('gray');
title('Edge Detection Using Fuzzy Logic')
```

Result



Compare



簇划分由样本之间的相似性进行度量和划分 —— 无监督方法

对无标记样本集 X ， $X = \{x_i \mid i=1,2,\dots,m\}$ ，由聚类过程将 m 个样本划分为 k 个不相交的簇 C_k ， $C_k = \{C_j \mid j=1,2,\dots,k\}$ ，同时满足 $C_j \cap C_{k \neq j} = \emptyset$ ，且 $X = \bigcup_{j=1}^k C_j$ 。

度量方式

- 外部度量，将聚类结果与某个“参考指标”作比较，如邻近聚类相关方法；
- 内部度量，直接考察聚类结果而不利用任何参考指标，如k-均值聚类。

度量方法

对给定 n 维向量样本 x_1, x_2 ，其欧氏距离为

$$d = \left(\sum_{p=1}^n |x_{1p} - x_{2p}|^2 \right)^{1/2}$$

式中， $x_1 = \{x_{1p} \mid p=1,2,\dots,n\}$ ， $x_2 = \{x_{2p} \mid p=1,2,\dots,n\}$ 。

模糊聚类——模糊 k -均值聚类

首先，什么是 k -均值聚类

若给定 m 个样本，将其划分为 k 个簇， $1 < k \leq m$ ，并给出簇心 k_j ，这一聚类过程可按

如下步骤：

(1) 初始随机选取 k 个样本点作为聚类中心；

(2) 计算每一个样本 x_i 与各聚类中心 k_j 之间的距离， $i = 1, 2, \dots, m$ ， $j = 1, 2, \dots, k$ ，采用 Euclidean 距离平方和为目标函数

$$J = \sum_{i=1}^m \sum_{j=1}^k |x_i - k_j|^2$$

式中， J 为目标函数， x_i 为第 i 个样本向量， $|x| = \sqrt{x^T x}$ ， k_j 为第 j 个聚类中心。

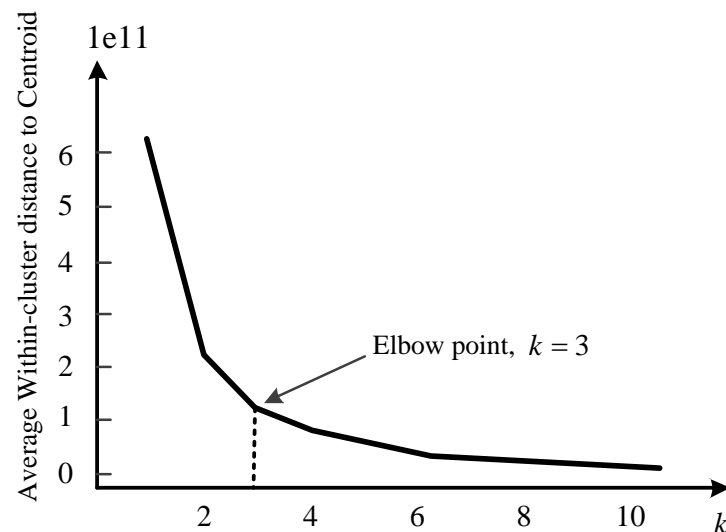
(3) 将各样本按设定阈值归入距离最近的簇，并求得每一个簇中的样本均值，作为新的聚类中心；

(4) 返回第二步，计算各个样本与新的聚类中心的距离，更新样本归类并得到新聚类中心，如此迭代计算直到簇心变化小于给定值或达到迭代次数，计算完成。

模糊聚类——模糊 k -均值聚类

k -均值聚类

聚类收敛与簇心数的肘形曲线
(Andrew Ng, 2010)



初始簇心选择影响着聚类结果:

- 簇心数量，也就是聚类类别数目的影响

若 k 的值选取过大，对于特定的数据集，将可能有太多的聚类，例如，数据集中有3个明显的聚类，如果选择 $k=5$ ，将破坏分类，导致错误的结论。

- 簇心初值，即簇心初始位置的影响

选择不同的初始点可能导致不同的簇划分规则和迭代过程，导致不同的聚类结果。

模糊聚类——模糊 k -均值聚类

模糊 k -均值聚类——由 k -均值算法派生而来

模糊 k -均值聚类通过迭代求取隶属度 μ_{ij} 与聚类中心 k_j 使目标函数最小。

$$J = \sum_{i=1}^N \sum_{j=1}^K (\mu_{ij})^m |x_i - k_j|^2$$

式中， $m > 1$ ，为模糊因子，表示类别间重叠的程度，若 m 越大，迭代中新旧类别之间的影响越小， x_i 为第 i 个样本向量， k_j 为第 j 个聚类中心，

$$i = 1, 2, \dots, N, \quad j = 1, 2, \dots, K, \quad |x| = \sqrt{x^T x}.$$

模糊聚类——模糊k-均值聚类

模糊 k -均值聚类的具体步骤：

- (1) 选取模式类数 K ， $1 < K \leq N$ ， N 为样本个数；
- (2) 根据先验知识确定样本对于各个类别的隶属度 $\mu_{ij}(0)$ ，
 $i = 1, 2, \dots, N$ ， $j = 1, 2, \dots, K$ ， μ_{ij} 表示第 i 个样本对第 j 个类别的隶属度；

- (3) 求新的聚类中心 k_j^{new}

$$k_j^{new} = \frac{\sum_{i=1}^N x^i (\mu_{ij}^{new})^m}{\sum_{i=1}^N (\mu_{ij}^{new})^m}$$

其中，隶属度更新按照

$$\mu_{ij}^{new} = \left[\sum_{r=1}^K \left(\frac{1/d_{ij}}{1/d_{ir}} \right)^{1/(m-1)} \right]^{-1}$$

式中， d_{ij} 为迭代过程中第 i 个样本到第 j 类聚类中心的距离， d_{ir} 为

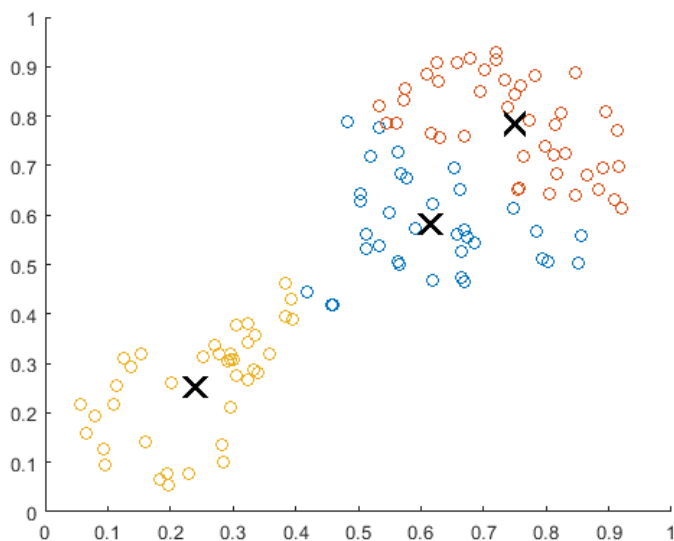
迭代过程中第 i 个样本分别到 K 个聚类中心的距离， $r = 1, 2, \dots, K$ 。

为避免可能当 $d_{ij} = 0$ 时， μ_{ij} 分母为零的情况，特别指定 $\mu_{ij} = 1$ 当 $d_{ij} = 0$ ；

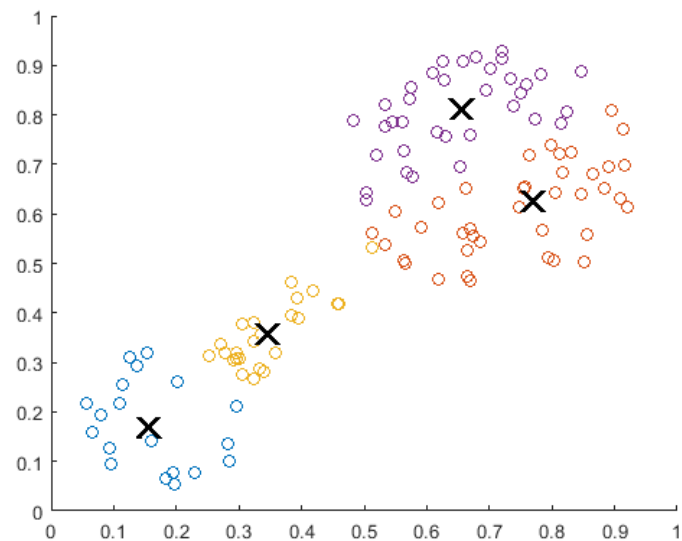
模糊聚类——模糊 k -均值聚类

(4) 重复(3)，直到隶属度的变化小于给定值 ε ；

(5) 根据隶属度 μ_{ij} 进行划分，完成聚类。



a) 聚类簇数 $k=3$



b) 聚类簇数 $k=4$

选取不同簇数时的聚类情况

模糊聚类——模糊 k -均值聚类

区别和联系：

- k -均值算法在聚类过程中，每次得到的结果，类别之间的边界是明确的，聚类中心根据当前属于该类样本迭代更新；
- 模糊 k -均值算法在聚类过程中，每一次计算聚类中心都需要用到全部样本，每次得到的类别边界是模糊的，聚类准则也体现了模糊性。

模糊 k -均值聚类特点：

- 首先设定一些类及每个样本对各类的隶属度，然后通过迭代不断更新隶属度，直至隶属度的变化量小于规定的阈值，达到收敛；
- 预先指定的模糊参数 $m>1$ ，确定了聚类之间的交叉程度；
- 假若参数 $m>1$ 很多，意味着第 j 个聚类中心点对新聚类中心的影响较小。

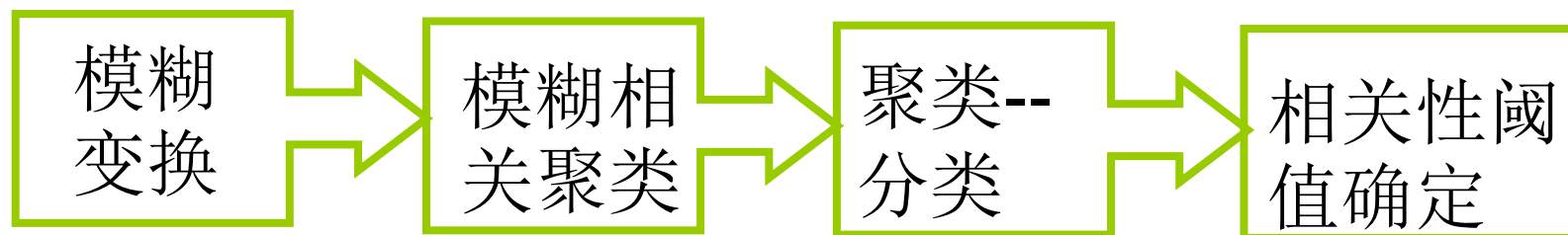


举例

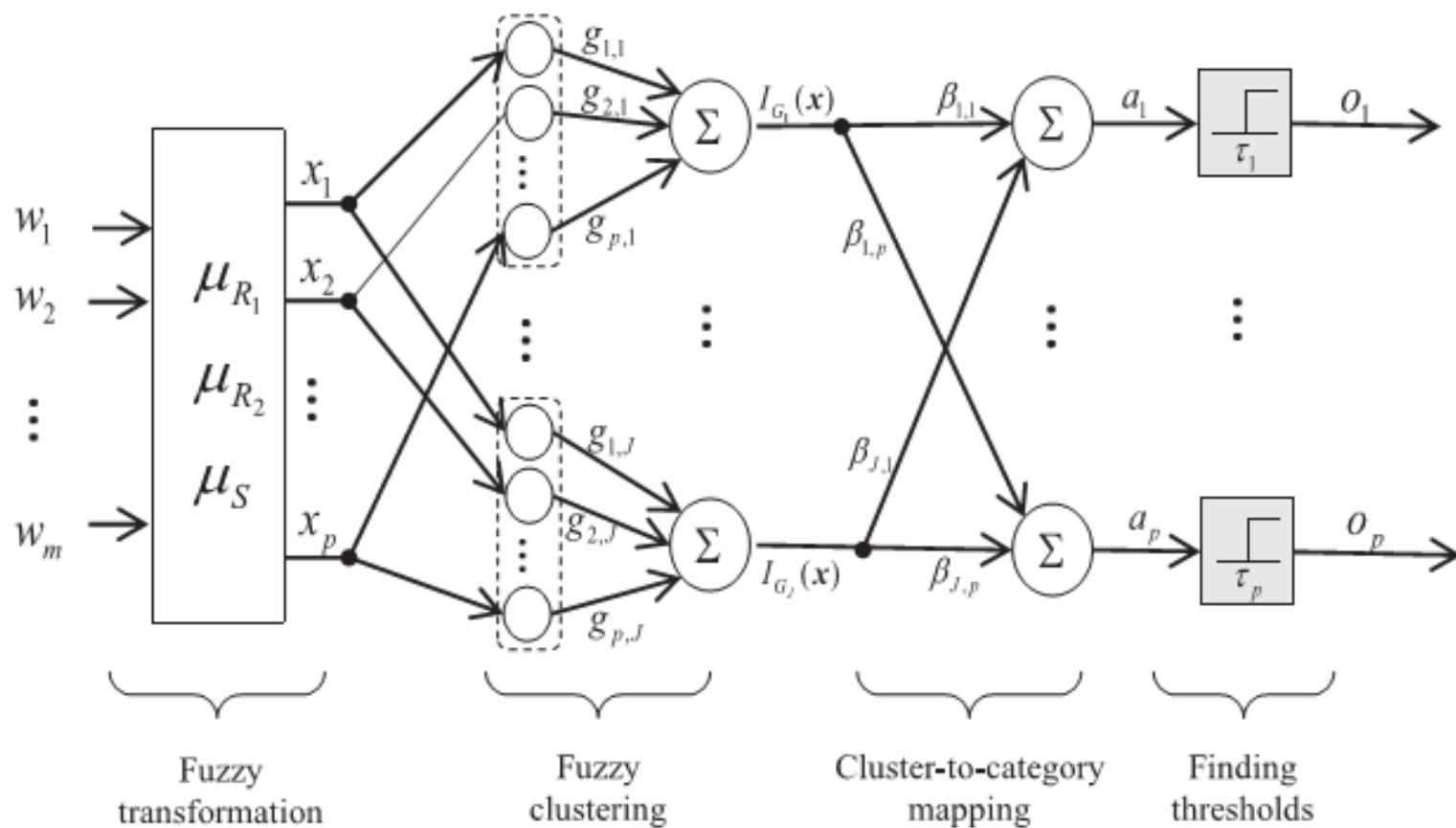
模糊控制在多元文本分类中的应用：

【背景】在信息检索，文本处理，网络搜索中，多元文本分类（一个文档可以属于不只一个分类）是一个重要的研究方向。

模型的建立：



例



例

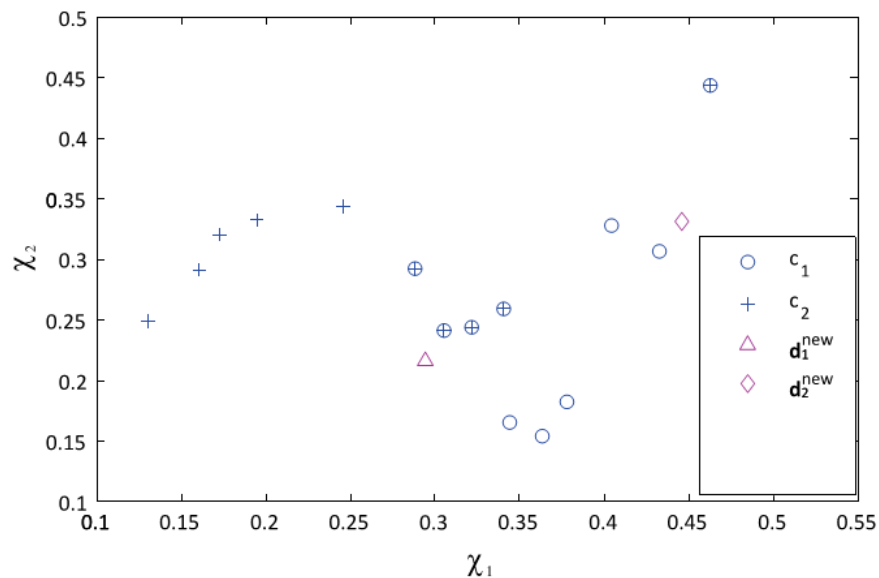
小例子:

$$D: \left\{ \begin{array}{l} (\mathbf{d}^{(1)} = \langle 8, 9, 8, 1, 2, 1, 2, 1, 2, 1 \rangle, \mathbf{y}^{(1)} = \langle 1, 0 \rangle) \\ (\mathbf{d}^{(2)} = \langle 6, 4, 7, 1, 2, 1, 2, 2, 1, 0 \rangle, \mathbf{y}^{(2)} = \langle 1, 0 \rangle) \\ (\mathbf{d}^{(3)} = \langle 5, 7, 6, 2, 1, 2, 1, 1, 2, 2 \rangle, \mathbf{y}^{(3)} = \langle 1, 0 \rangle) \\ (\mathbf{d}^{(4)} = \langle 2, 2, 3, 1, 1, 2, 1, 2, 1, 2 \rangle, \mathbf{y}^{(4)} = \langle 1, 0 \rangle) \\ (\mathbf{d}^{(5)} = \langle 3, 2, 1, 1, 2, 1, 2, 1, 2, 2 \rangle, \mathbf{y}^{(5)} = \langle 1, 0 \rangle) \\ (\mathbf{d}^{(6)} = \langle 1, 0, 1, 3, 1, 5, 2, 1, 1, 1 \rangle, \mathbf{y}^{(6)} = \langle 1, 1 \rangle) \\ (\mathbf{d}^{(7)} = \langle 0, 2, 1, 1, 2, 5, 3, 0, 1, 2 \rangle, \mathbf{y}^{(7)} = \langle 1, 1 \rangle) \\ (\mathbf{d}^{(8)} = \langle 1, 2, 0, 1, 2, 2, 1, 2, 2, 1 \rangle, \mathbf{y}^{(8)} = \langle 1, 1 \rangle) \\ (\mathbf{d}^{(9)} = \langle 2, 0, 3, 0, 4, 1, 0, 2, 1, 2 \rangle, \mathbf{y}^{(9)} = \langle 1, 1 \rangle) \\ (\mathbf{d}^{(10)} = \langle 0, 2, 1, 1, 0, 1, 0, 1, 2, 0 \rangle, \mathbf{y}^{(10)} = \langle 1, 1 \rangle) \\ (\mathbf{d}^{(11)} = \langle 3, 0, 1, 0, 2, 0, 1, 10, 3, 2 \rangle, \mathbf{y}^{(11)} = \langle 0, 1 \rangle) \\ (\mathbf{d}^{(12)} = \langle 1, 0, 0, 2, 1, 0, 0, 4, 5, 1 \rangle, \mathbf{y}^{(12)} = \langle 0, 1 \rangle) \\ (\mathbf{d}^{(13)} = \langle 1, 0, 1, 0, 1, 2, 1, 2, 4, 2 \rangle, \mathbf{y}^{(13)} = \langle 0, 1 \rangle) \\ (\mathbf{d}^{(14)} = \langle 2, 1, 0, 1, 2, 0, 1, 5, 2, 8 \rangle, \mathbf{y}^{(14)} = \langle 0, 1 \rangle) \\ (\mathbf{d}^{(15)} = \langle 0, 1, 1, 0, 2, 1, 2, 6, 3, 3 \rangle, \mathbf{y}^{(15)} = \langle 0, 1 \rangle) \end{array} \right\}$$

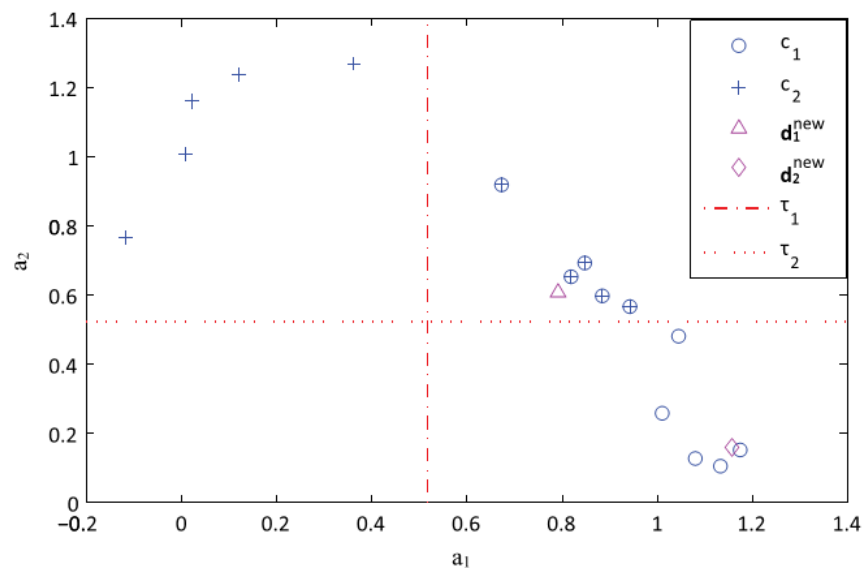
$$T: \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$$

$$C: \{c_1, c_2\}$$

例



a) 类别相关模糊隶属度分布



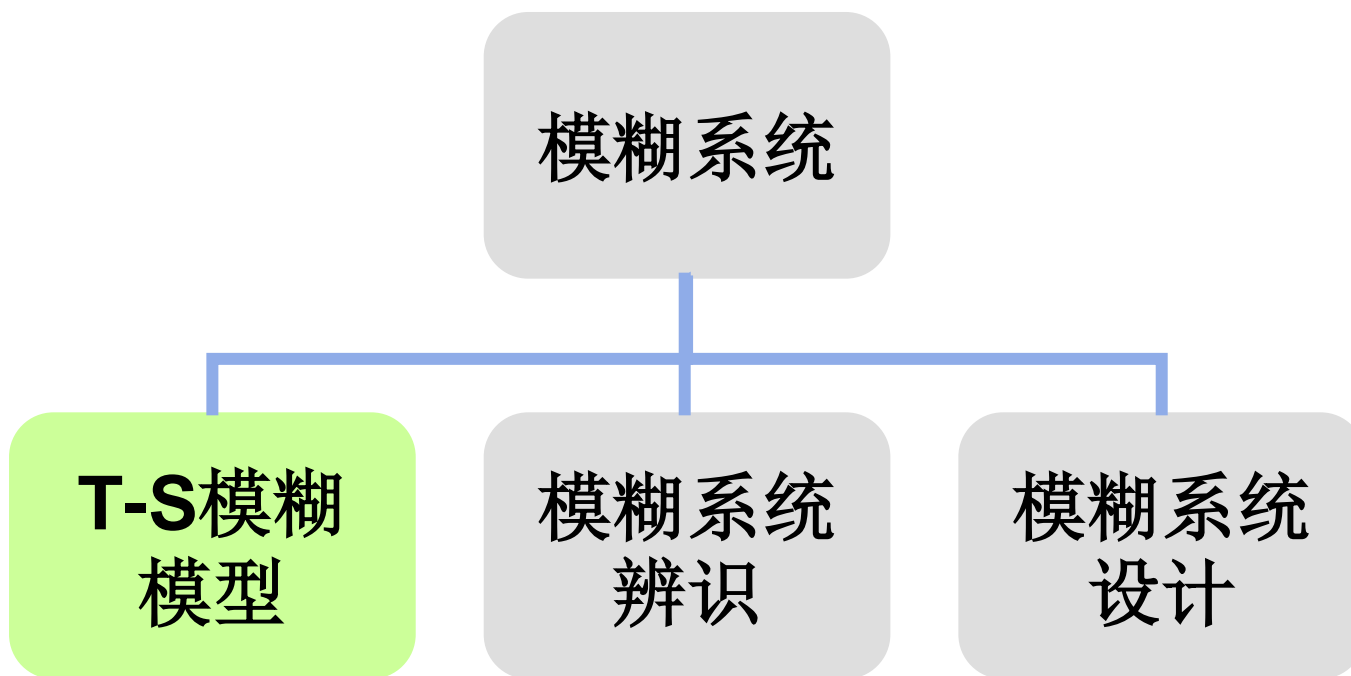
b) 类别相似度分布

样本数据分布示意图(Lee, TFS, 2014)

样本与类别之间较分散

c_1, c_2 聚类类别、聚集中中心明显，
类内差异小，类间差异显著

T-S型（函数型）模糊系统



1 T-S模糊模型

1985年, Tokyo Institute of Technology 学者Takagi T. 与Sugeno M.在试图解决多变量之间的蕴涵关系和推理问题时, 提出了一种蕴涵规则(Takagi, 1985), 即 Takagi-Sugeno函数型模型.



Michio Sugeno

116

IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. SMC-15, NO. 1, JANUARY/FEBRUARY 1985

Fuzzy Identification of Systems and Its Applications to Modeling and Control

TOMOHIRO TAKAGI AND MICHIO SUGENO

Abstract—A mathematical tool to build a fuzzy model of a system where fuzzy implications and reasoning are used is presented in this paper. The premise of an implication is the description of fuzzy subspace of inputs and its consequence is a linear input-output relation. The method of identification of a system using its input-output data is then shown. Two applications of the method to industrial processes are also discussed: a water cleaning process and a converter in a steel-making process.

I. INTRODUCTION

reasoning is given by the aggregation of the values inferred by some implications that were applied to an input.

This paper also shows the method of identification of a system using its input-output data. As is well-known, identification is divided into two parts: structure identification and parameter identification.

In its nature structure identification is almost independent of a format of system description. We omit this part

1 T-S模糊模型

T-S模糊模型——适用于局部线性、能够分段进行控制的系统

- 输出可以是清晰值，或输入量的函数
- 可逼近任意非线性系统
- 数学分析，定量研究

一般地，表示为：

$R^i : \text{if } x_1 \text{ is } A_1^i, \text{ and } x_2 \text{ is } A_2^i, \text{ and } \dots, \text{ and } x_k \text{ is } A_k^i$

$\text{then } y^i = p_0^i + p_1^i x_1 + \dots + p_k^i x_k$

式中， i 为第 i^{th} 条规则， x_j 是第 j 个输入量， $j=1, \dots, k$ ， k 为输入量的数目， A_j^i 是第 i 条规则下第 j 个输入所属的模糊子集， y^i 为第 i 条规则时的输出量， p_j^i 为结论参数。

1 T-S模糊模型

如果给定输入模糊向量 $(x_1^i, x_2^i, \dots, x_n^i)$, 那么由各条规则的输出 y^i ($i = 1, 2, \dots, n$) 的加权平均可求得输出

$$\hat{y} = \frac{\sum_{i=1}^n G^i y^i}{\sum_{i=1}^n G^i}$$

式中, n 为模糊规则的数量, y^i 由第 i 条规则的结论方程式计算, G^i 为第 i 条规则的真值

$$G^i = \prod_{j=1}^k A_j^i(x_j^i)$$

其中, Π 为模糊化算子, 通常由取小或代数积计算, 也就是每条规则的激活度.

1 T-S模糊模型

[例1] 假如有如下3条规则：

R^1 : if x_1 is **big1** then $y = 5x_1$

R^2 : if x_1 is **small1** and x_2 is **small2** , then $y = 2x_1 + 0.5x_2$

R^3 : if x_2 is **big2** , then $y = 3x_2$

试求当 $x_1 = 10, x_2 = 6$ 的输出.

规则	前 题	结 论	真 值
R^1		$y = 5 \times 10 = 50$	0.2
R^2		$y = 2 \times 10 + 0.5 \times 6 = 23$	$0.25 \wedge 0.333 = 0.25$
R^3		$y = 3 \times 6 = 18$	0.5

1 T-S模糊模型

□ 采取取小运算的推理过程

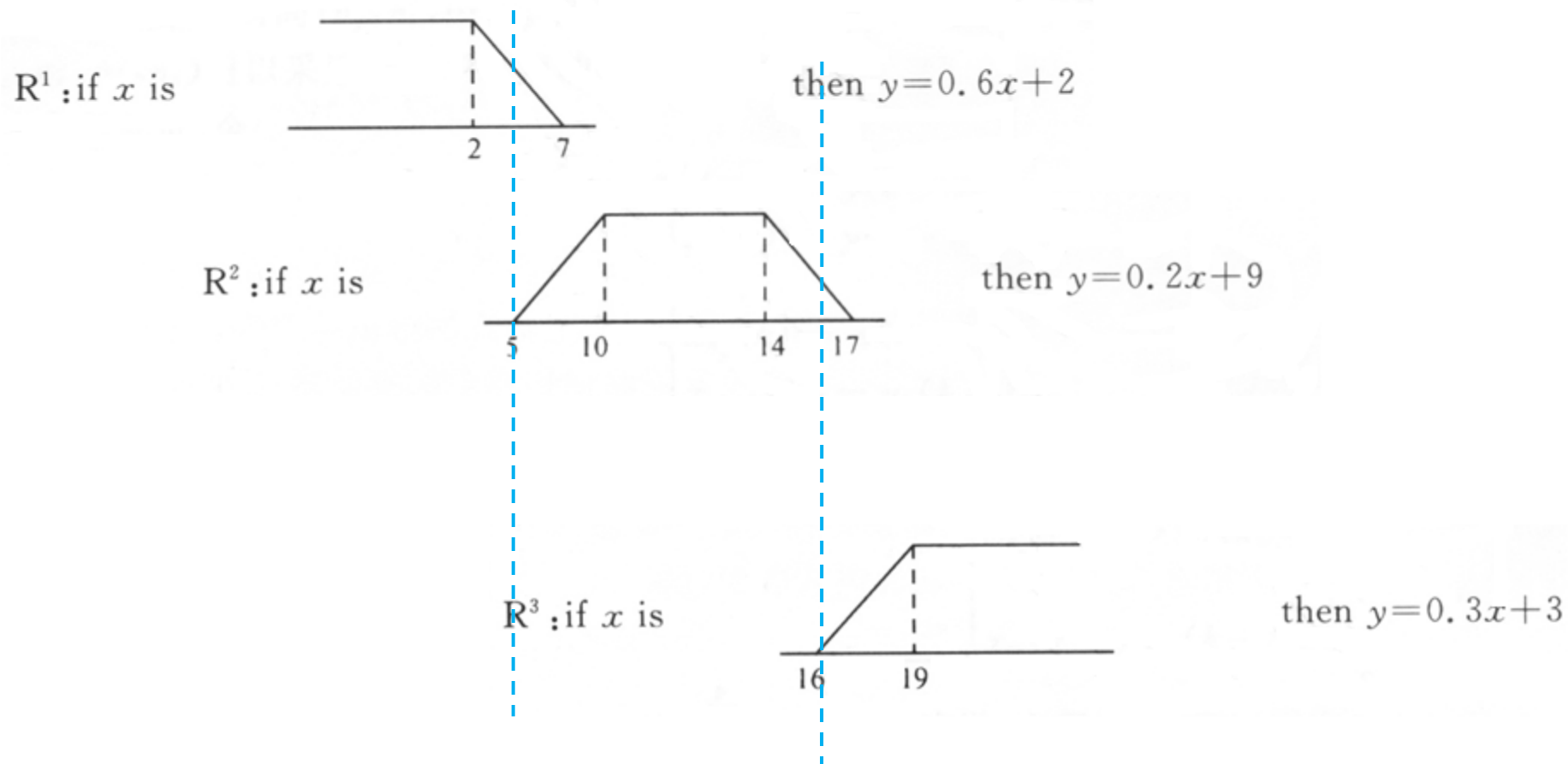
- “前提”：规则中前提模糊子集的隶属函数
- “结论”：由各规则结论方程式计算 y^i
- “真值”：显示了结论 “ $y = y^i$ ” 的真实程度

由该模糊规则推导的最终输出值为

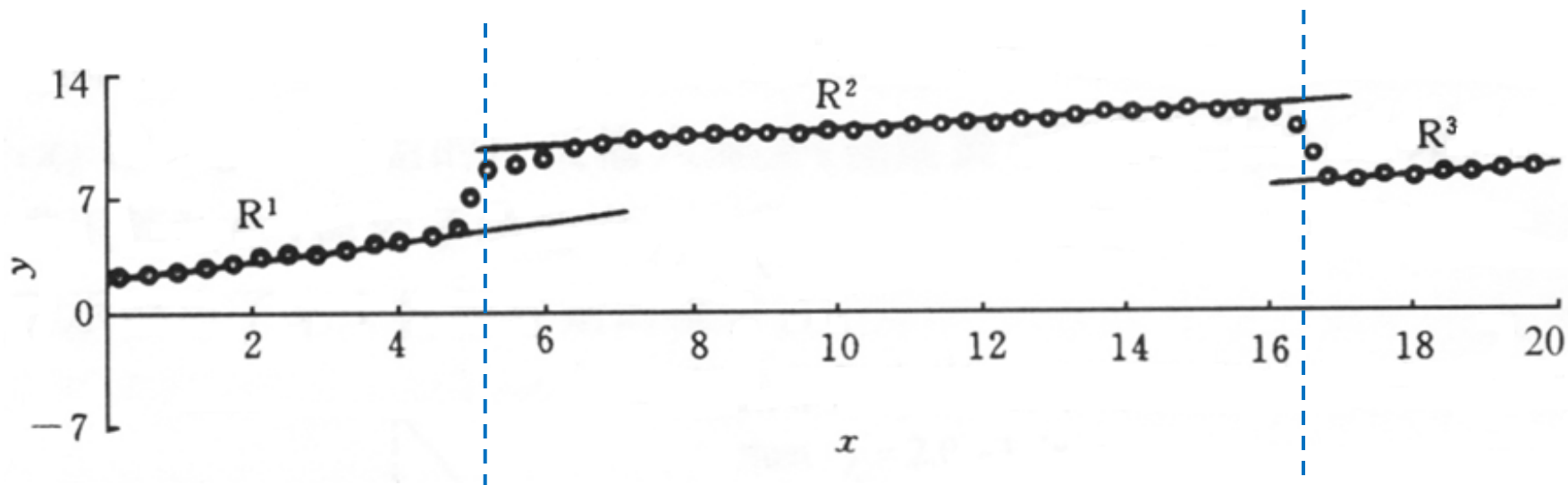
$$y = \frac{0.2 \times 50 + 0.25 \times 23 + 0.5 \times 18}{0.2 + 0.25 + 0.5} \approx 26.05$$

1 T-S模糊模型

[例2] 假如有3条规则如下：



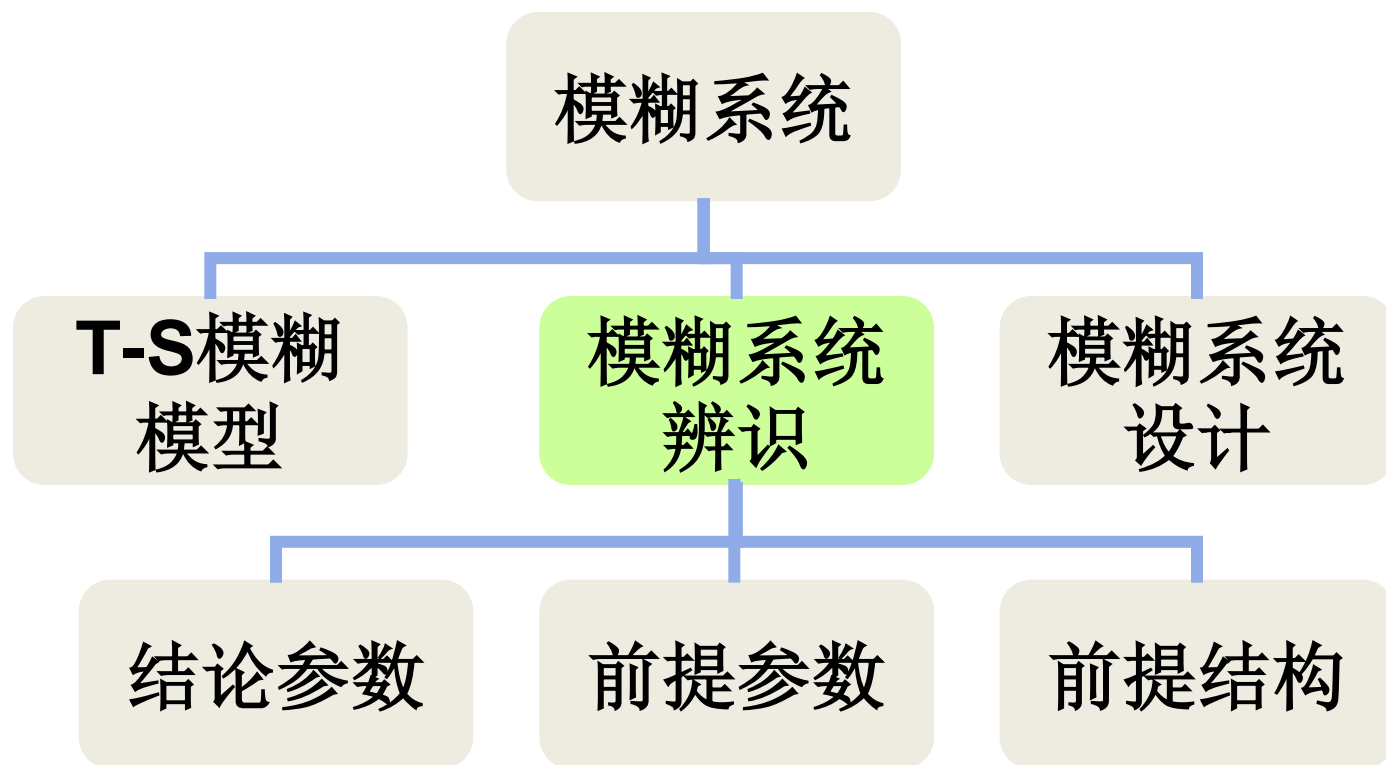
1 T-S模糊模型



模糊模型表示的关系

- 直线 R^i 表示由规则 R^i 结论方程式描述的 x 和 y 之间的线性关系
- “。”表示 x 和 y 之间的关系曲线，受到若干规则中前提条件的约束

2 模糊系统辨识



2 模糊系统辨识

*For the things we have to learn before we can do them,
We learn by doing them.
-Aristotle*

考虑具有如下逻辑蕴涵形式的模糊模型：

$R^i : \text{if } x_1 \text{ is } A_1^i, \text{ and } x_2 \text{ is } A_2^i, \text{ and } \dots \text{ and } x_k \text{ is } A_k^i$

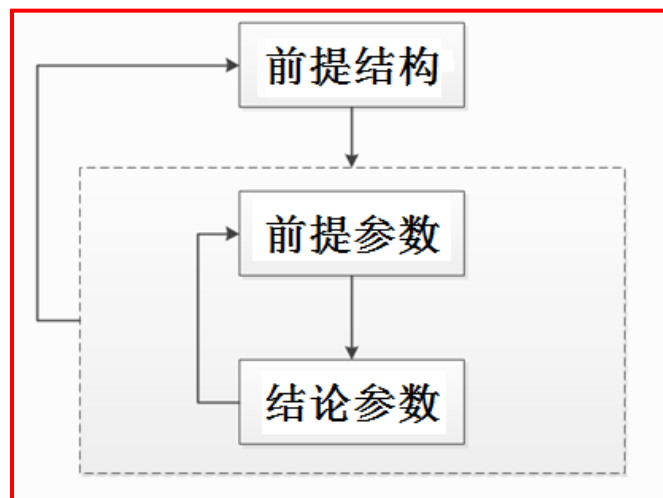
$\text{then } y^i = p_0^i + p_1^i x_1 + \dots + p_k^i x_k$

为了辨识系统，需要根据目标系统的输入-输出数据确定以下三类参数：

1. x_1^i, \dots, x_k^i 组成蕴涵关系的前提结构
2. A_1^i, \dots, A_k^i 模糊集的隶属度函数（前提参数）
3. p_0^i, \dots, p_k^i 结论参数

2 模糊系统辨识

- ◆ 前提结构、前提参数与模糊子空间在**输入变量空间**上的划分相关；
- ◆ 结论参数与每个模糊集上**输入-输出**的模糊关系相关。



2.1 模糊系统辨识--结论参数

◆ 系统辨识——结论参数

要点：给定前提结构、前提参数的情况下，确定结论参数

若给定系统有如下蕴涵逻辑：

$$R^1: \text{if } x_1 \text{ is } A_1^1, \dots \text{ and } x_k \text{ is } A_k^1 \text{ then } y^1 = p_0^1 + p_1^1 \cdot x_1 + \dots + p_k^1 \cdot x_k$$
$$\vdots$$

$$R^n : \text{if } x_1 \text{ is } A_1^n, \dots \text{ and } x_k \text{ is } A_k^n \text{ then } y^n = p_0^n + p_1^n \cdot x_1 + \dots + p_k^n \cdot x_k$$

对于输入 (x_1, \dots, x_k) 可以得到输出：

$$y = \frac{\sum_{i=1}^n (A_1^i(x_1) \wedge \dots \wedge A_k^i(x_k)) \cdot (p_0^i + p_1^i \cdot x_1 + \dots + p_k^i \cdot x_k)}{\sum_{i=1}^n (A_1^i(x_1) \wedge \dots \wedge A_k^i(x_k))}$$

2.1 模糊系统辨识--结论参数

令 β 为

$$\beta_i = \frac{A_1^i(x_1) \wedge \cdots \wedge A_k^i(x_k)}{\sum_{i=1}^n (A_1^i(x_1) \wedge \cdots \wedge A_k^i(x_k))}$$

y 表示为

$$\begin{aligned} y &= \sum_i^n \beta_i \cdot (p_0^i + p_1^i \cdot x_1 + \cdots + p_k^i \cdot x_k) \\ &= \sum_{i=1}^n (p_0^i \cdot \beta_i + p_1^i \cdot x_1 \cdot \beta_i + \cdots + p_k^i \cdot x_k \cdot \beta_i) \end{aligned}$$

2.1 模糊系统辨识--结论参数

当已知输入-输出数据

$$x_j = (x_{1j}, x_{2j}, \dots, x_{kj}) \quad Y = y_j$$

$$\beta_i = (\beta_{i1}, \beta_{i2}, \dots, \beta_{im}), \quad (i = 1, 2, \dots, n), \quad (j = 1, 2, \dots, m)$$

可通过最小二乘法得到结论参数：

$$P = (X^T X)^{-1} X^T Y$$

$$X = \underbrace{\left[\begin{array}{cccc} \beta_{11} \cdots \beta_{n1} & x_{11} \cdot \beta_{11} \cdots x_{11} \cdot \beta_{n1} & \cdots & \cdots & x_{k1} \cdot \beta_{11} \cdots x_{k1} \cdot \beta_{n1} \\ \beta_{12} \cdots \beta_{n2} & x_{12} \cdot \beta_{12} \cdots x_{12} \cdot \beta_{n2} & \cdots & \cdots & x_{k2} \cdot \beta_{12} \cdots x_{k2} \cdot \beta_{n2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{1m} \cdots \beta_{nm} & x_{1m} \cdot \beta_{1m} \cdots x_{1m} \cdot \beta_{nm} & \cdots & \cdots & x_{km} \cdot \beta_{1m} \cdots x_{km} \cdot \beta_{nm} \end{array} \right]}_{n(k+1)} \right] \Bigg\} m$$

2.1 模糊系统辨识--结论参数

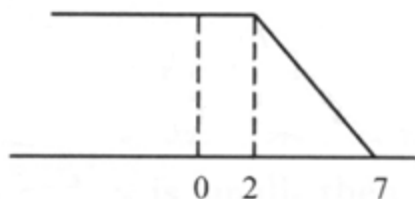
$$Y = \underbrace{\begin{bmatrix} \beta_{11} \cdots \beta_{n1} & x_{11} \cdot \beta_{11} \cdots x_{11} \cdot \beta_{n1} & \cdots & \cdots & x_{k1} \cdot \beta_{11} \cdots x_{k1} \cdot \beta_{n1} \\ \beta_{12} \cdots \beta_{n2} & x_{12} \cdot \beta_{12} \cdots x_{12} \cdot \beta_{n2} & \cdots & \cdots & x_{k2} \cdot \beta_{12} \cdots x_{k2} \cdot \beta_{n2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{1m} \cdots \beta_{nm} & x_{1m} \cdot \beta_{1m} \cdots x_{1m} \cdot \beta_{nm} & \cdots & \cdots & x_{km} \cdot \beta_{1m} \cdots x_{km} \cdot \beta_{nm} \end{bmatrix}}_{n(k+1)} \begin{bmatrix} p_0^1 \\ p_0^2 \\ \vdots \\ p_1^1 \\ p_1^2 \\ \vdots \\ p_1^n \\ \vdots \\ p_k^1 \\ p_k^2 \\ \vdots \\ p_k^n \end{bmatrix} \Bigg\} n(k+1)$$

2.1 模糊系统辨识--结论参数

[例] 对于上例中的模糊系统，加入标准方差为0.5的高斯白噪声，并假定模型的**前提结构和前提参数与原系统相同**。

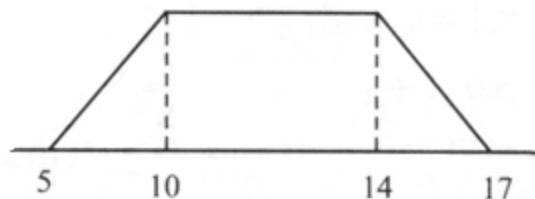
采用200组输入输出数据进行结论参数的辨识，得到如下结果：

R^1 : if x is



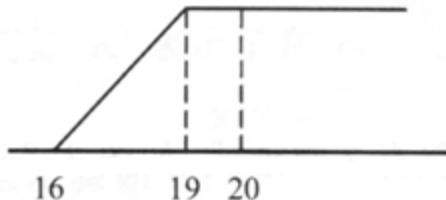
$$\text{then } y = 0.5542x + 2.3663$$

R^2 : if x is



$$\text{then } y = 0.2345x + 8.6705$$

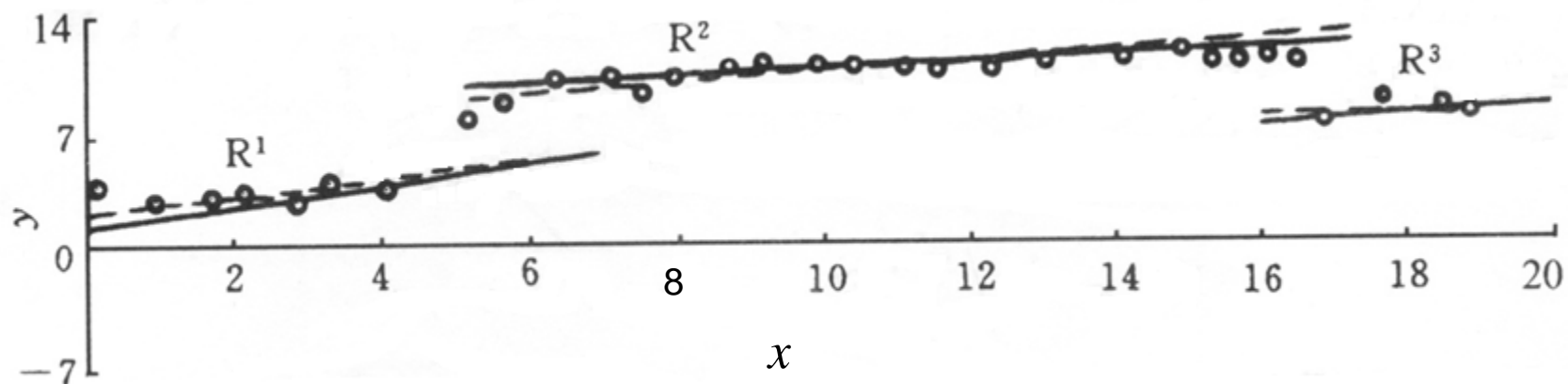
R^3 : if x is



$$\text{then } y = 0.2997x + 3.1702$$

2.1 模糊系统辨识--结论参数

- 下图绘出了含有噪声的输入输出数据、原始的结论和辨识结论
- 如果这组数据中不含有噪声，那么辨识出来的模型和原模型完全相同



———— 辨识的结论方程式 - - - - - 原始的结论方程式

原始数据和辨识的结果

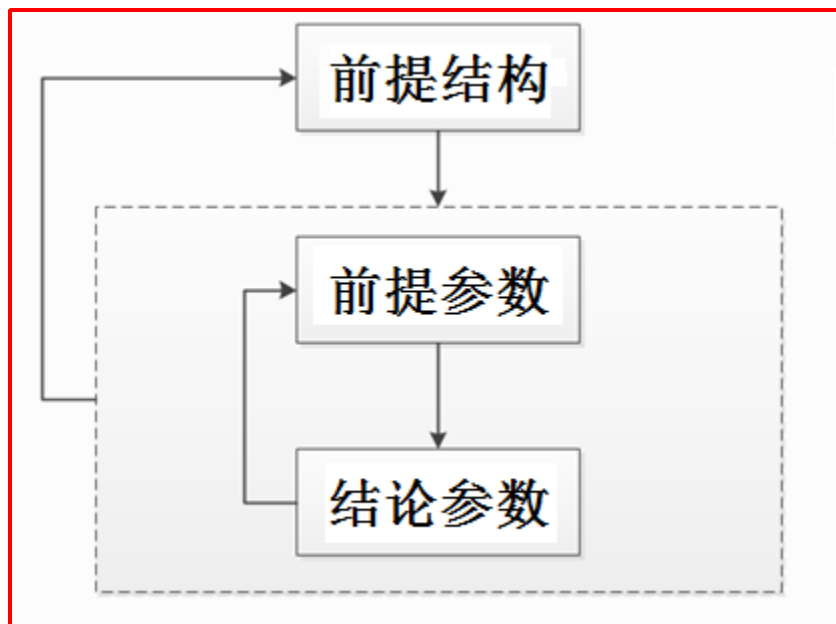
2.2 模糊系统辨识--前提参数

□ 模糊辨识的方法及步骤

对Takagi-Sugeno(T-S)模型：

- (1) 确定前提结构
- (2) 前提参数辨识(隶属度函数的中心, 宽度…)
- (3) 结论参数辨识

算法的框架



搜索法

非线性规划

最小二乘法

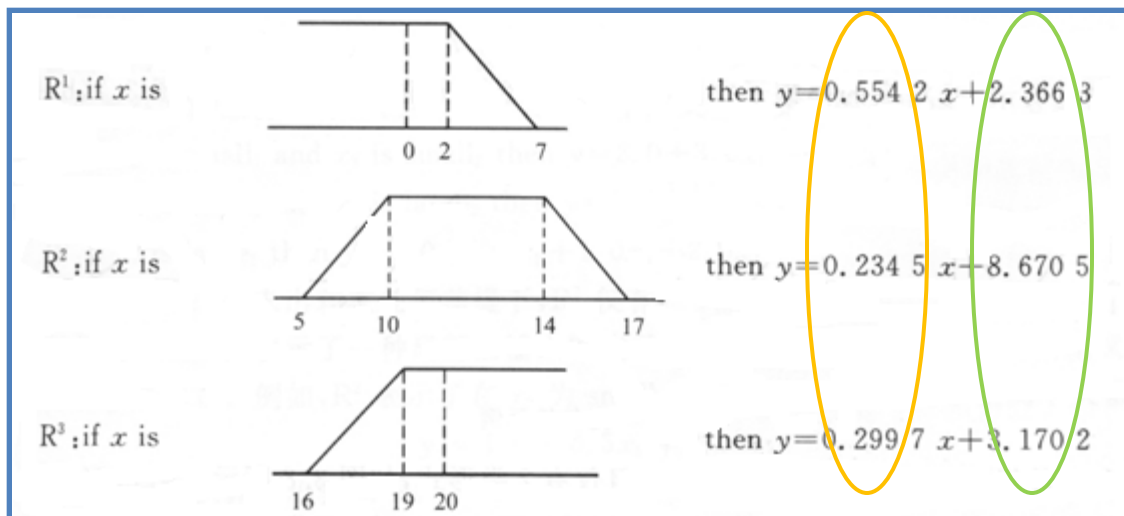
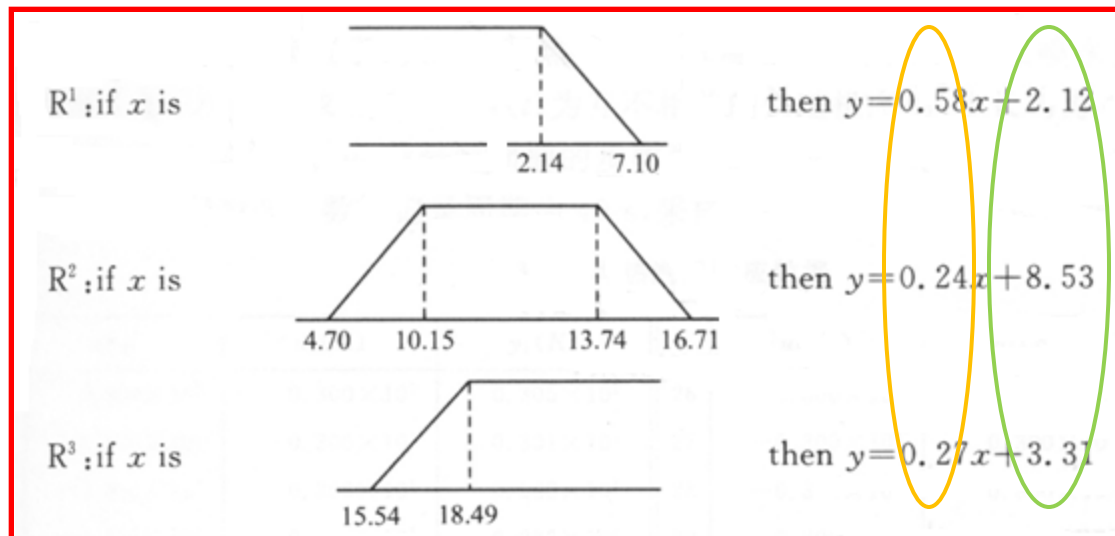
2.2 模糊系统辨识--前提参数

□ 前提参数的识别：

- 假定模糊集的隶属度函数为线性函数，通过每次给定的前提参数计算相应的结论参数
- 使得性能参数最小
- 选择性能参数取得最小值时的前提参数与结论参数组合

2.2 模糊系统辨识--前提参数

对于前例模糊系统，
当高斯噪声的标准方差
为0.5时，采用200组数
据对前提参数和结论参
数同时辨识，得到的模
糊模型如右图



- 原型参数与辨识模型参数的差别不大，引起误差的原因为**噪声**
- 若不加噪声，则可辨识出完全相同的前提参数和结论参数
- 必须加噪声，否则就不能表明本辨识算法的有效性

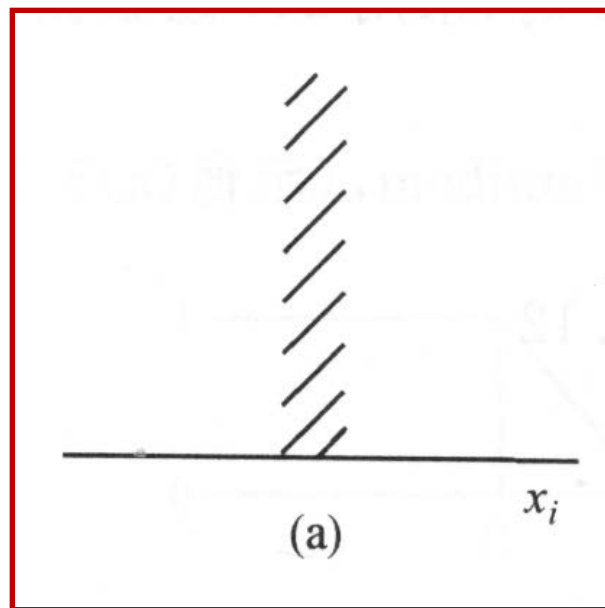
2.3 模糊系统辨识--前提结构

□ 前提结构

- 将 x_1 的范围划分为2个模糊子空间small和large，其他变量不分解。前提结构由2条规则表示：

R^1 : if x_1 is small then ...

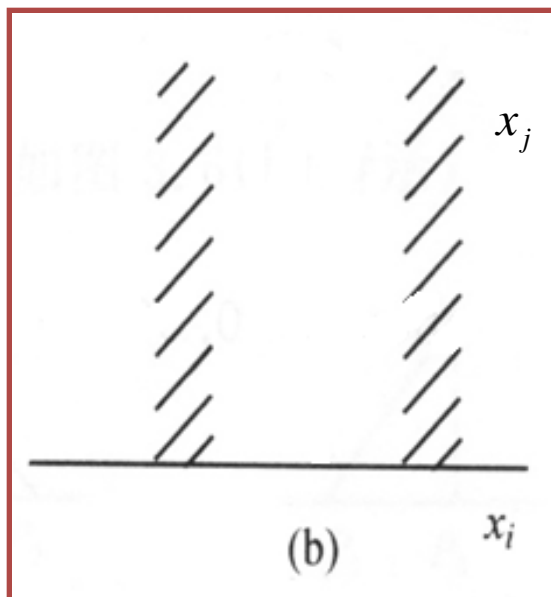
R^2 : if x_1 is large then ...



2.3 模糊系统辨识--前提结构

- 将前提中的模糊子空间由2个增加至3个
- 两种方法:

一是将 x_i 本身划分为3个模糊子空间，如图b所示：



前提结构由3条规则表示：

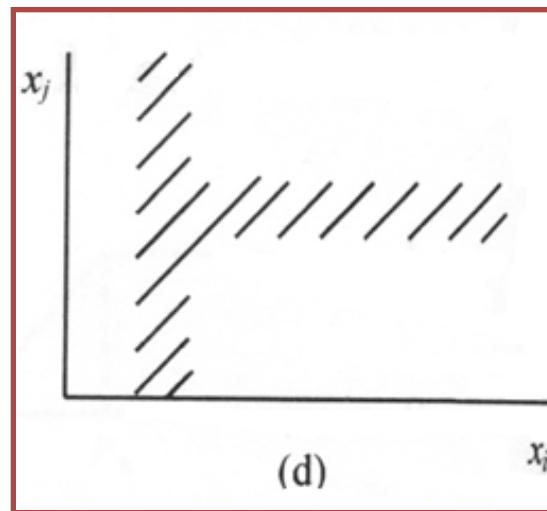
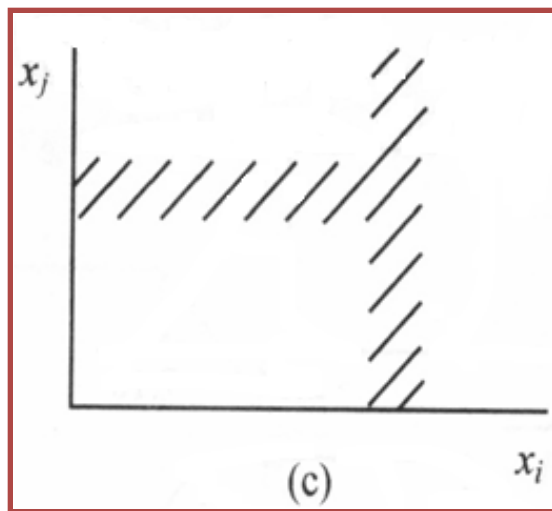
R^1 : if x_1 is small then ...

R^2 : if x_1 is medium then ...

R^3 : if x_1 is large then ...

2.3 模糊系统辨识--前提结构

- 还可另加入输入变量，如 x_j ，并将 x_j 范围划分为2个模糊子空间，如图c和d所示：



图c表示的前提结构由如下3条规则表示：

R^1 : if x_i is small₁ x_j is small₂ then ...

R^2 : if x_i is small₁ x_j is large₂ then ...

R^3 : if x_i is large₁ then ...

2.3 模糊系统辨识--前提结构

- 前件空间的划分

1. 变量空间的划分

2. 模糊子空间的选择：

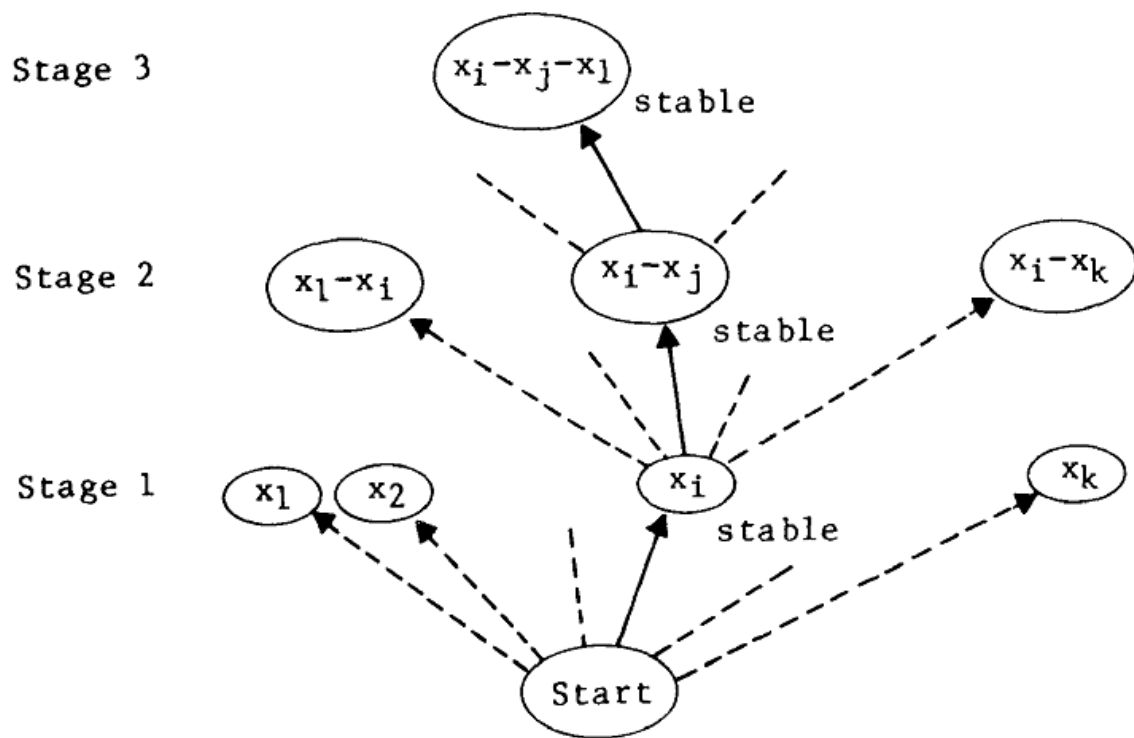
2个 (大、小)

或3个 (小、中、大)

或多个

- 搜索法

步骤 1. 把 $x_i (i=1, 2 \dots k)$ 划分为两个模糊子集，从中选择使特性指标为最小的 x_i ，称 $1-i$ 模型的稳态；



2.3 模糊系统辨识--前提结构

步骤 2. 从稳态开始, 将 x_i 与其他变量组合, 如 $x_i - x_j$, 称为2-j 模型, 各类模型可有四个组合, 选其中最小的1组, 形成2-j的稳态;

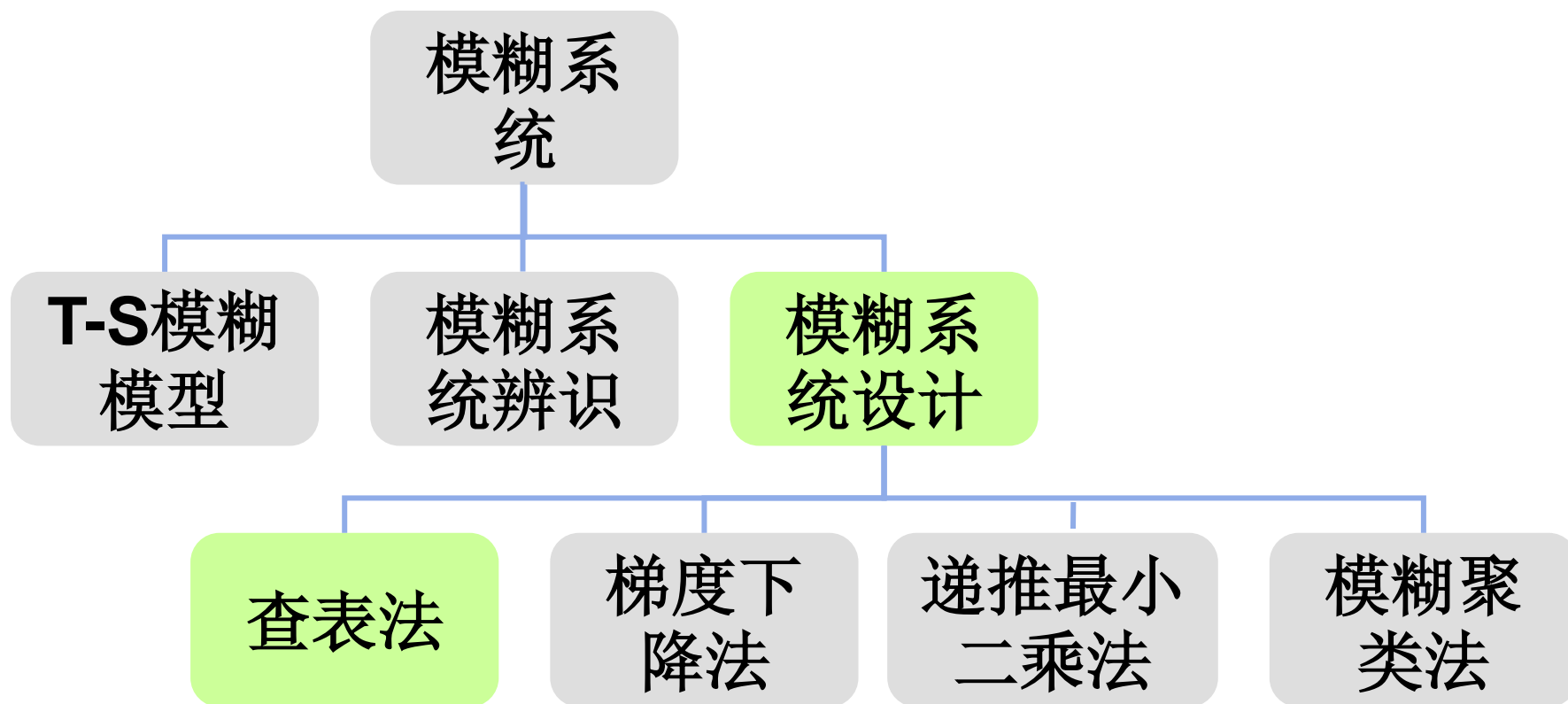
步骤 3. 将 $x_i - x_j$ 与其他变量进行第3变量组合, 形成如 $x_i - x_j - x_k$, 选择最优值直到:

- a. 稳态的性能参数小于某一阈值
- b. 稳态的蕴涵数目超过某一阈值

在输入空间已划分的情况下, 模糊辨识的实质

——就是在所定义的空间下, 给出规则的集合

3.模糊系统设计



3.1 模糊系统设计--查表法

查表法设计模糊系统的步骤

- ① 把输入输出空间划分为模糊空间
- ② 由一个输入输出数据产生一条模糊规则
- ③ 对步骤2中的每条规则都赋予一个强度
- ④ 创建模糊规则库
- ⑤ 构造模糊系统

3.1 模糊系统设计--查表法

步骤1: 将输入输出空间划分为模糊空间

- 在每个区间 $[\alpha_i, \beta_i]$ ($i = 1, 2, \dots, n$) 上定义 N_i 个模糊集 A_i^j ($j = 1, 2, \dots, N_i$)
- 在 A_i^j 处 $[\alpha_i, \beta_i]$ 是完备模糊集, 即对任意的 $x_i \in [\alpha_i, \beta_i]$ 都存在 $\mu_{A_i^j}(x_i) \neq 0$
- 可选 $\mu_{A_i^j}(x_i)$ 为四边形隶属度函数:

$$\mu_{A_i^j}(x_i) = \mu_{A_i^j}(x_i, a_i^j, b_i^j, c_i^j, d_i^j)$$

- 类似地, 定义 N_y 个模糊集 B^j ($j = 1, 2, \dots, N_y$), 在 $[\alpha_y, \beta_y]$ 上亦为完备模糊集合

下图中, $n = 2$, $N_1 = 5$, $N_2 = 7$, $N_y = 5$, 选用三角形隶属函数。

3.1 模糊系统设计--查表法

步骤2 由一个输入输出数据对产生一条模糊规则

- 根据每一个输入输出数据对 $(x_{01}^p, \dots, x_{0n}^p; y_0^p)$

确定 $x_{0i}^p (i = 1, 2, \dots, n)$ 的模糊集 $A_i^j (i = 1, 2, \dots, N_i)$ 的隶属度

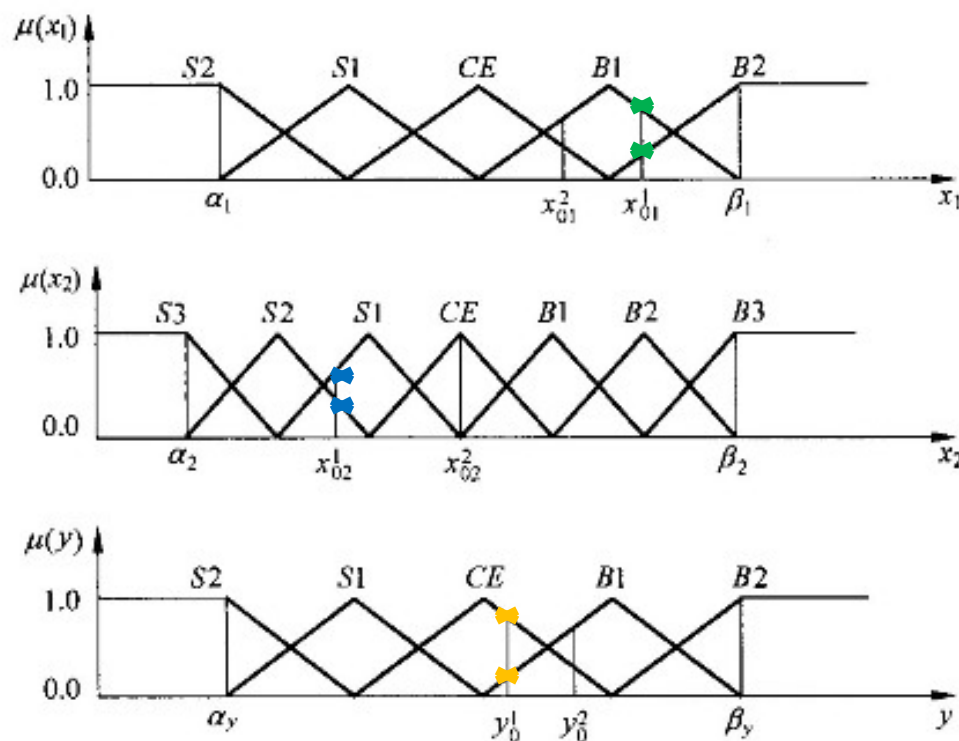
$$\mu_{A_i^j}(x_{0i}^p) (j = 1, 2, \dots, N_i, i = 1, 2, \dots, n)$$

- 确定 y 隶属于模糊集 $B^l (l = 1, 2, \dots, N_y)$ 的隶属度值

$$\mu_{B^l}(y_0^p) (l = 1, 2, \dots, N_y)$$

3.1 模糊系统设计--查表法

两个输入一个输出情况下的空间划分
及相应的隶属度函数定义



- 近似地, x_1 隶属于B1的隶属度值为0.8, 隶属于B2的值为0.2, 隶属于其他模糊集的隶属度值为0;
- x_2 隶属于S1的隶属度值为0.6, 隶属于S2的隶属度值为0.4, 隶属于其他模糊集的隶属度值为0;
- y 隶属于CE的隶属度值为0.8, 隶属于B1的隶属度值为0.2, 隶属于其他模糊集的隶属度值为0.

3.1 模糊系统设计--查表法

- 对每个输入变量 $x_i (i = 1, 2, \dots, n)$ 确定使其有最大隶属度值的模糊集，即确定 $A_i^{j^*}$ 使得
$$\mu_{A_i^{j^*}}(x_{0i}^p) \geq \mu_{A_i^j}(x_{0i}^p) (j = 1, 2, \dots, N_i)$$

- 确定 B^{l^*} 使得
$$\mu_{B^{l^*}}(y_0^p) \geq \mu_{B^l}(y_0^p) (l = 1, 2, \dots, N_y)$$

- 可得 IF - THEN 规则：

If x_1 is $A_1^{j_1^*}$ and \dots and x_n is $A_n^{j_n^*}$, then y is B^{l^*}

- 对上述例子，由输入输出数据对 $(x_{01}^1, x_{02}^1, y_0^1)$

可以得到 $A_1^{j^*} = B_1, A_2^{k^*} = S_1, B^{l^*} = CE$

由输入输出数据对 $(x_{01}^2, x_{02}^2, y_0^2)$ 可得 $A_1^{j^*} = B_1, A_2^{k^*} = CE, B^{l^*} = B_1$

- 可建立规则如下：

If x_1 is B_1 and x_2 is S_1 , then y is CE

If x_1 is B_1 and x_2 is CE , then y is B_1

3.1 模糊系统设计--查表法

步骤3 对步骤2的每条规则赋予一个强度

设输入输出数据对为 $(x_{01}^p, \dots, x_{0n}^p; y_0^p)$, 则其强度可定义为:

$$D(\text{规则}p) = [\mu_{A_1^{j_1}}(x_{01}^p) \mu_{A_2^{j_2}}(x_{02}^p) \cdots \mu_{A_n^{j_n}}(x_{0n}^p)] \mu_{B^{j^*}}(y_0^p)$$

- 因输入输出数据对的数量通常较大且每对数据均将产生一条规则, 所以会出现互相矛盾的规则, 即IF部分相同, THEN部分不同
- 为解决这一冲突, 赋予步骤2中的每条规则一个强度, 使得一个冲突群中仅有一条规则具有最大的强度
- 不仅解决了冲突问题, 且大大减少规则数量

3.1 模糊系统设计--查表法

在上例中，

由数据对 $(x_{01}^1 x_{02}^1; y_0^1)$ 产生的规则强度为

$$\begin{aligned} D(\text{规则1}) &= \mu_{B_1}(x_{01}^1) \mu_{S_1}(x_{02}^1) \mu_{CE}(y_0^1) \\ &= 0.8 \times 0.6 \times 0.8 = 0.384 \end{aligned}$$

由数据对 $(x_{01}^2 x_{02}^2; y_0^2)$ 产生的规则强度为

$$\begin{aligned} D(\text{规则2}) &= \mu_{B_1}(x_{01}^2) \mu_{CE}(x_{02}^2) \mu_{B_1}(y_0^2) \\ &= 0.6 \times 1 \times 0.7 = 0.42 \end{aligned}$$

3.1 模糊系统设计--查表法

步骤4 创建模糊规则库

模糊规则库由以下三个规则集合组成：

- 步骤2中产生的与其他规则不发生冲突的规则——隐性知识
- 一个冲突规则群体（具有相同 IF 部分的规则）中具有最大强度的规则 ——隐性知识
- 来自于专家的语言规则 ——显性知识

步骤5 基于模糊规则库构造模糊系统

- 根据步骤4产生的模糊规则库可以构造任意方案的模糊系统

3.1 模糊系统设计--查表法

直观地，两维输入情况下可以把一个模糊规则库描述为一个查询表格

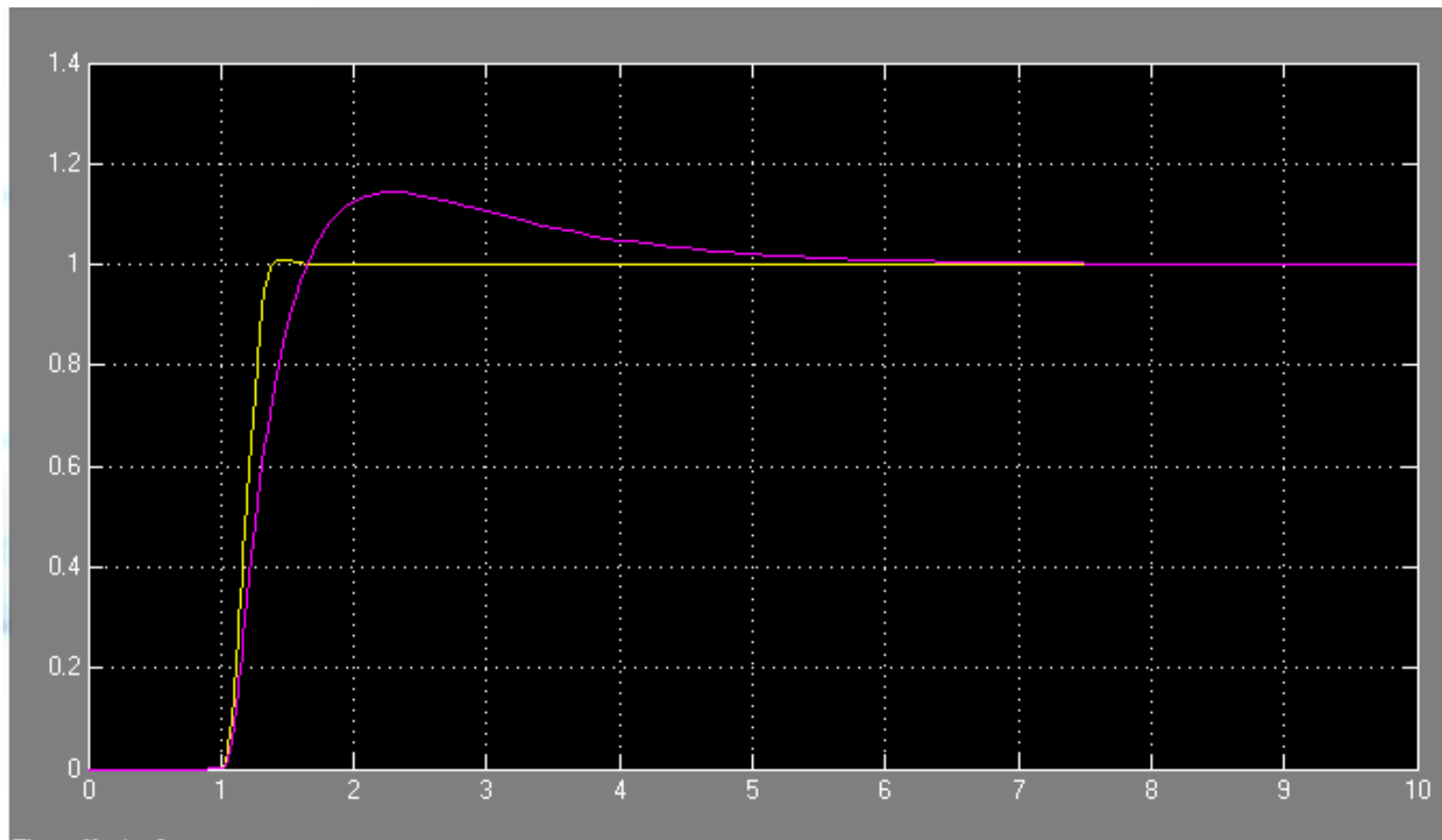
S_3					
S_2					
S_1					
CE					
B_1					
B_2					
B_3					
	S_2	S_1	CE	B_1	B_2

- 每个格子代表 $[\alpha_1, \beta_1]$ 中的模糊集合和 $[\alpha_2, \beta_2]$ 中的模糊集合的一个组合
- 由此能得到一个可能的规则
- 一个冲突的规则群体是由同一个格子中的规则组成的——查表法。

举例

模糊控制器设计——查表法

运行，Scope输出曲线
相同参数时模糊控制与普通PID控制



黄—模糊控制器输出，红—PID输出

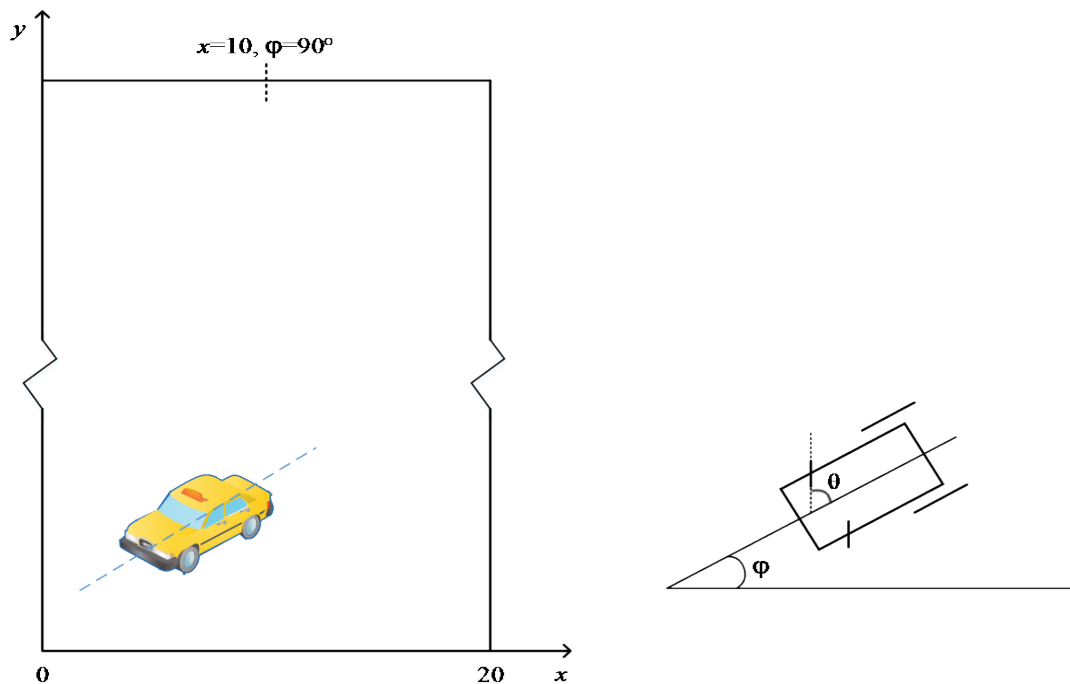
查表法在倒车控制中的应用

将卡车倒至指定装卸车位是一个非线性控制问题。采用传统的控制方法，首先需要建立该系统的数学模型，然后根据非线性理论设计控制器。

设计一个模糊控制器来模仿司机。

假设有一个有经验的司机且已知卡车的状态以及司机将卡车后倒至车位的控制行为，即获得输入输出数据对集合。那么就可以采用查表法，根据这些输入输出数据对来设计一个模糊系统，并用设计的模糊系统来取代司机。

查表法在倒车控制中的应用



上图是仿真的卡车和装卸区域。卡车的位置由 Φ , x , y 确定, 其中 Φ 是卡车与水平线的夹角, θ 是卡车的控制量即方向盘转动角。

查表法在倒车控制中的应用

为了简化该问题，假设卡车只允许后倒，卡车每隔一段时间都会向后移动，移动距离是固定的。而且在卡车和装卸车位之间有足够的间隙，不用考虑 y 的控制。

设 $x \in [0, 20]$ ， $\Phi \in [-90^\circ, 270^\circ]$ ， $\theta \in [-40^\circ, 40^\circ]$ ；

即 $U = [0, 20] \times [-90^\circ, 270^\circ]$ ， $V = [-40^\circ, 40^\circ]$ 。

要求设计一个输入为 (x, Φ) 、输出为 θ 的控制器，使卡车的最终状态为

$$(x_f, \Phi_f) = (10, 90^\circ)$$

查表法在倒车控制中的应用

首先，产生一组初始的输入输出数据对 (x^p, Φ^p, θ^p) ，再利用试错法产生一系列输入输出数据对。即，当卡车从一个初始状态开始后倒车时，在每一时刻根据常识确定控制量 θ （实际上这种常识就是在某种状态下如何控制方向盘角度的经验），经过几次实践后，就可以找到与最平滑最成功的轨迹相对应的输入输出数据对。以下14个初始状态将分别用来产生所需要的输入输出数据对。

$$\begin{aligned}(x_0, \phi_0^\circ) = & (1, 0), (1, 90), (1, 270), (7, 0), (7, 90), (7, 180), \\ & (7, 270), (13, 0), (13, 90), (13, 180), (13, 270), \\ & (19, 90), (19, 180), (19, 270)\end{aligned}$$

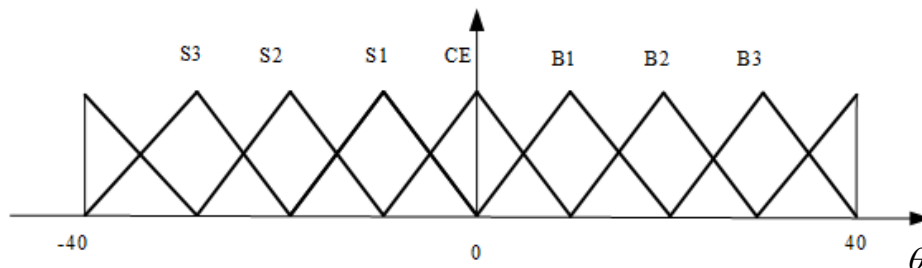
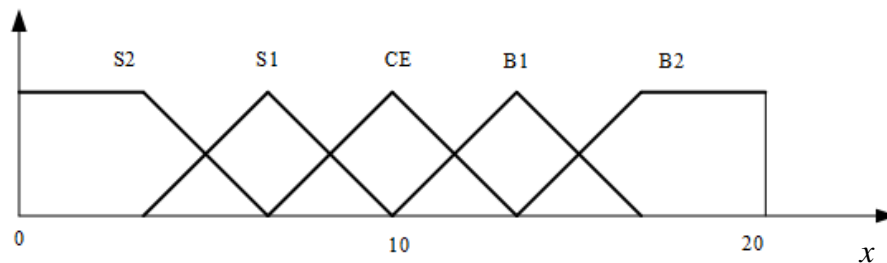
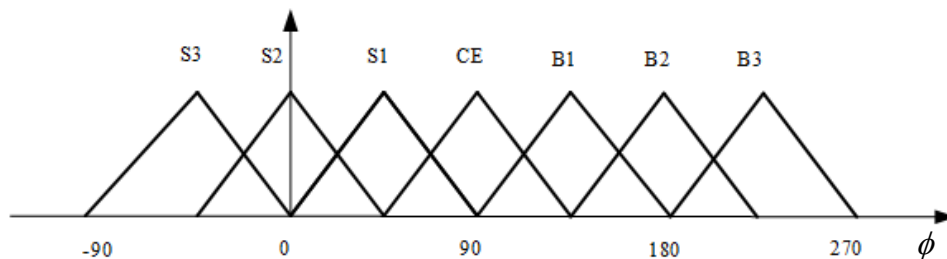
查表法在倒车控制中的应用

- ◆ 右表给出了 $(x, \phi) = (1, 0^\circ)$ 时的输入输出数据对
- ◆ 其他13种初始状态下的输入输出数据对也可以利用类似的方法获得
- ◆ 因此可以根据这些输入输出数据对，利用查表法设计模糊系统.

t	x	ϕ	θ
0	1.00	0.00	-19.00
1	1.95	9.37	-17.95
2	2.88	18.23	-16.90
3	3.79	26.57	-15.85
4	4.65	34.44	-14.80
5	5.45	41.78	-13.75
6	6.18	48.60	-12.70
7	7.48	54.91	-11.65
8	7.90	60.71	-10.60
9	8.72	65.99	-9.55
10	9.01	70.75	-8.50
11	9.28	74.98	-7.45
12	9.46	78.70	-6.40
13	9.59	81.90	-5.34
14	9.72	84.57	-4.30
15	9.81	86.72	-3.25
16	9.88	88.34	-2.20
17	9.91	89.44	0.00

查表法在倒车控制中的应用

步骤1 在 $[-90^\circ, 270^\circ]$ 上定义7个模糊集，在 $[0, 20]$ 上定义5个模糊集，在 $[-40^\circ, 40^\circ]$ 上定义7个模糊集。



查表法在倒车控制中的应用

步骤2和步骤3 由每对输入输出数据获得一条规则及强度

x	ϕ	θ	强度
S2	S2	S2	1.00
S2	S2	S2	0.92
S2	S2	S2	0.35
S2	S2	S2	0.12
S2	S2	S2	0.07
S1	S2	S1	0.08
S1	S1	S1	0.18
S1	S1	S1	0.53
S1	S1	S1	0.56
S1	S1	S1	0.60
CE	S1	S1	0.35

x	ϕ	θ	强度
CE	S1	S1	0.21
CE	S1	CE	0.16
CE	CE	CE	0.32
CE	CE	CE	0.45
CE	CE	CE	0.54
CE	CE	CE	0.88
CE	CE	CE	0.92

查表法在倒车控制中的应用

步骤4 产生最终的模糊规则库，如下表所示

ϕ

	x				
S3	S2	S3			
S2	S2	S3	S3	S3	
S1	B1	S1	S2	S3	S2
CE	B2	B2	CE	S2	S2
B1	B2	B3	B2	B1	S1
B2		B3	B3	B3	B2
B3				B3	B2
	S2	S1	CE	B1	B2

查表法在倒车控制中的应用

步骤5 采用带有乘积推理机、单值模糊器、中心平均解模糊器的模糊系统。

□ 将上述模糊系统作为一个倒车控制器应用在卡车上，为了仿真该控制系统，需要一个倒车的数学模型。

选用如下的逼近模型：

$$x(t+1) = x(t) + \cos[\phi(t) + \theta(t)] + \sin[\theta(t)]\cos[\phi(t)]$$

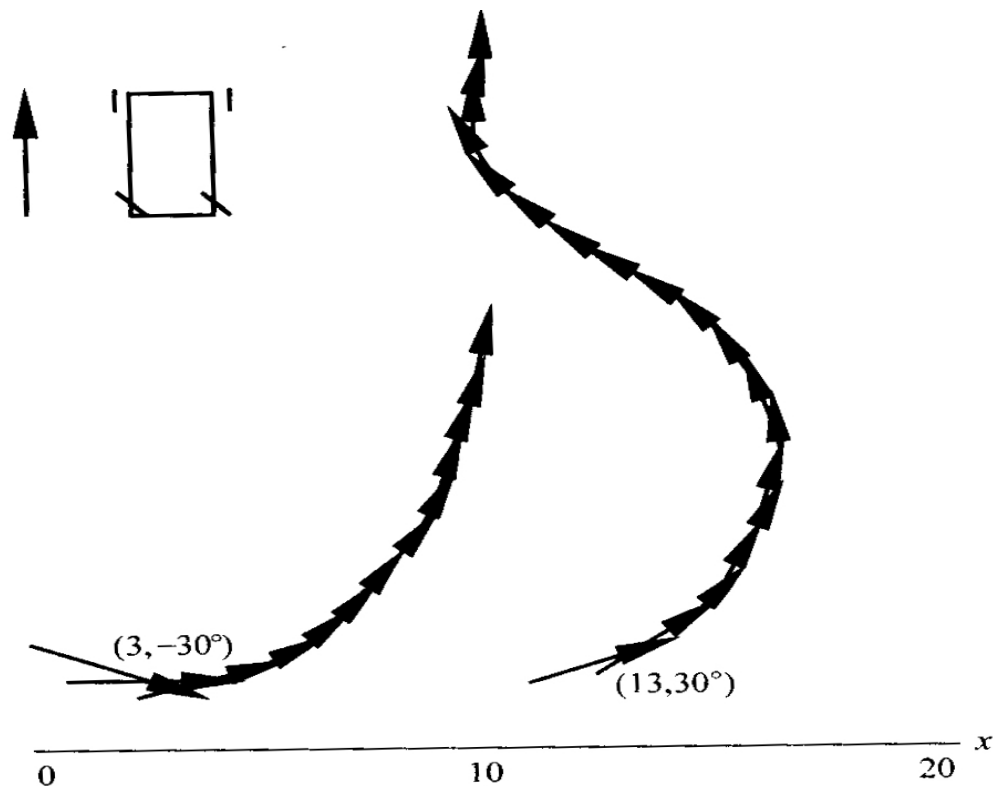
$$y(t+1) = y(t) + \sin[\phi(t) + \theta(t)] - \sin[\theta(t)]\sin[\phi(t)]$$

$$\phi(t+1) = \phi(t) - \sin^{-1}\left[\frac{2\sin(\theta(t))}{b}\right]$$

b为卡车的长度，仿真中假设b=4。

查表法在倒车控制中的应用

采用上述步骤1~5设计的模糊系统，从两个初始条件 $(x_0, \varphi_0)=(3, -30^\circ)$ 和 $(x_0, \varphi_0)=(13, 30^\circ)$ 出发的倒车轨迹如：

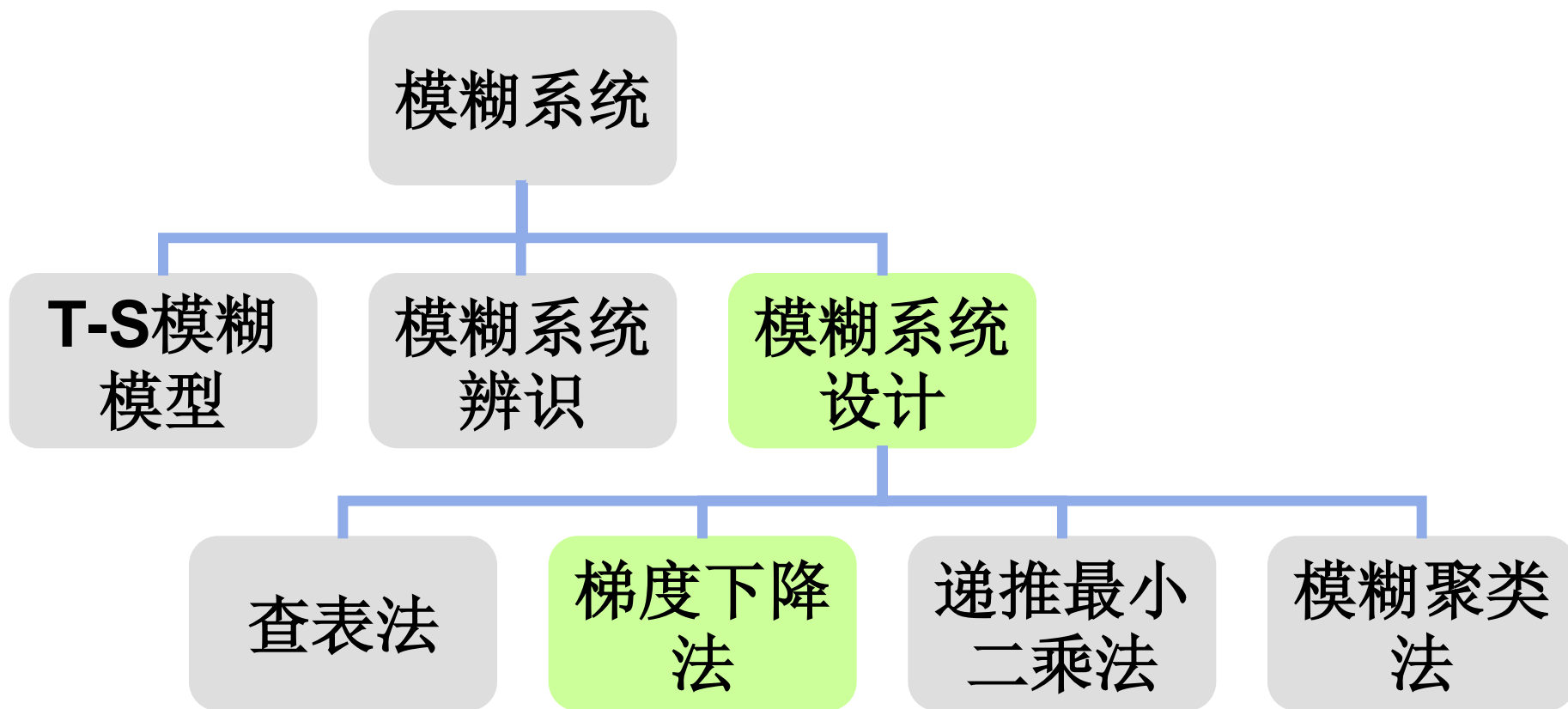


查表法在倒车控制中的应用



- ◆ 其他实例——侧方
- ◆ 全方位旋转底盘
- ◆ 实景示例

3.2 模糊系统设计--梯度下降法



3.2 模糊系统设计--梯度下降法

模糊系统结构的选择

□ 隶属度函数的设置与选择

- 查表法中，隶属度函数的选取**固定不变**的且不依赖于输入输出数据对，即隶属度函数并不根据输入输出数据对进行优化
- 梯度下降法设计模糊系统时，隶属度函数进行某种程度**优化选择**获得

3.2 模糊系统设计--梯度下降法

◆ 模糊系统结构设计 将模糊系统选定为带有乘积推理机、单值模糊器、中心平均解模糊器和高斯隶属度函数的模糊系统

设模糊系统形式为

$$f(x) = \frac{\sum_{l=1}^M \bar{y}_l \left[\prod_{i=1}^n \exp \left(- \left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \right]}{\sum_{l=1}^M \left[\prod_{i=1}^n \exp \left(- \left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \right]}$$

式中 M 是固定不变的, \bar{y}_l , \bar{x}_i^l , σ_i^l 是自由变化的参数。

该模糊系统结构已知, 但是参数未知。

3.2 模糊系统设计--梯度下降法

◆ 模糊系统参数设计

同用查表法设计模糊系统一样，数据由输入输出数据对给定，任务是设计一个乘积推理机、单值模糊器、中心平均解模糊器和高斯隶属函数的模糊系统 $f(x)$ ，使得 $e^p = [f(x_0^p) - y_0^p]^2 / 2$ 的拟合误差最小，即确定参数 $\bar{y}^l, \bar{x}_i^l, \sigma_i^l$ 使得 e^p 最小。

参数 $\bar{y}^l, \bar{x}_i^l, \sigma_i^l$ 可采用梯度下降法来确定。

$$\bar{y}^l(q+1) = \bar{y}^l(q) - \alpha \frac{\partial e}{\partial \bar{y}^l} \Big|_q$$

$$\bar{x}_i^l(q+1) = \bar{x}_i^l(q) - \alpha \frac{\partial e}{\partial \bar{x}_i^l} \Big|_q$$

$$\sigma_i^l(q+1) = \sigma_i^l(q) - \alpha \frac{\partial e}{\partial \sigma_i^l} \Big|_q$$

3.2 模糊系统设计--梯度下降法

◆ 采用误差反向传播算法：

$\frac{\partial e}{\partial \bar{y}^l}, \frac{\partial e}{\partial \bar{x}_i^l}, \frac{\partial e}{\partial \sigma_i^l}$ 可根据复合函数求导规则得到

$$\frac{\partial e}{\partial \bar{y}^l} = (f - y)z^l / b$$

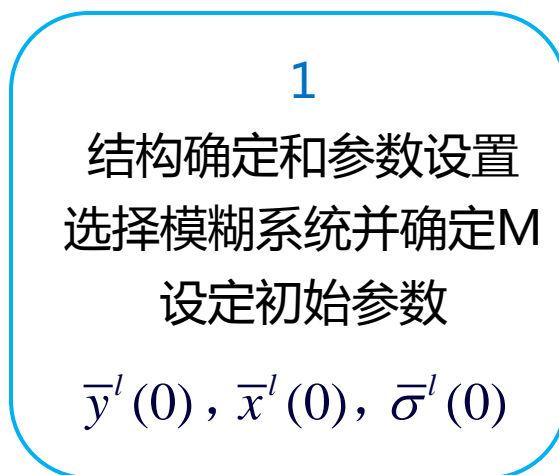
$$\frac{\partial e}{\partial \bar{x}_i^l} = (f - y)(\bar{y}^l - f)z^l 2(x_{0i}^p - \bar{x}_i^l) / (b\sigma_i^{l2})$$

$$\frac{\partial e}{\partial \sigma_i^l} = (f - y)(\bar{y}^l(q) - f)z^l 2(x_{0i}^p - \bar{x}_i^l)^2 / (b\sigma_i^{l3})$$

$$\text{其中, } f = a / b, \quad a = \sum_{l=1}^M \bar{y}^l \bar{z}^l, \quad b = \sum_{l=1}^M \bar{z}^l, \quad \bar{z}^l = \prod_{i=1}^n \exp\left(-\left(\frac{x_i^l - \bar{x}_i^l}{\sigma_i^l}\right)^2\right)。$$

3.2 模糊系统设计--梯度下降法

模糊系统的梯度下降法设计流程



- M越大，产生的参数越多，运算就越复杂，但是给出的逼近精度越高。
- 初始参数可能是根据专家的语言规则确定的，也可能是由均匀地覆盖输入输出空间的相应的隶属度函数确定的。
- 梯度下降法初始参数的选择对计算方法成败很关键。如果初始参数的选择接近最优参数，则算法会很快收敛于最优解，反之，算法可能收敛至非最优解甚至不收敛。

3.2 模糊系统设计--梯度下降法

2

给定输入数据

计算模糊系统的输出

$$z^l = \prod_{i=1}^n \exp \left(- \left(\frac{x_{0i}^p - \bar{x}_i^l(q)}{\sigma_i^l(q)} \right)^2 \right)$$

$$b = \sum_{l=1}^M z^l$$

$$a = \sum_{l=1}^M \bar{y}^l(q) z^l$$



- 对一组给定数据 $(x_0^p \ y_0^p)$ $p=1, 2, \dots$, 在学习的第 q ($q=1, 2, \dots$) 阶段, 将输入 x_0^p 作为模糊系统的输入层, 计算第1层~第3层的输出

3.2 模糊系统设计--梯度下降法

3 调整参数

$$\bar{y}^l(q+1), \bar{x}_i^l(q+1), \bar{\sigma}_i^l(q+1)$$

其中, $y = y_0^p$, z^l , b , a , f 为步骤2计算的 z^l , b , a , f

4 令 $q = q + 1$, 返回步骤2重新计算
直到误差 $|f - y_0^p| < \varepsilon$ 或 q 预先给定值

5 令 $p = p + 1$, 返回步骤2重新计算
即用下一组输入输出数据对 $(x_0^{p+1} \ y_0^{p+1})$
来调整参数

6 如有必要, 令 $p = 1$, 重新计算
步骤2 ~ 5直到得到满意的结果

举例

梯度下降法设计模糊系统 in 非线性系统辨识中的应用

- 系统辨识根据传感器测量结果确定系统模型
- 模糊系统是万能逼近器，作为非线性系统的辨识模型是合适的

□ 采用梯度下降法逼近动态系统中未知非线性部分的动态特性

【例】 考虑如下的离散时间非线性动态系统

$$y(k+1) = f(y(k), \dots, y(k-n+1), u(k), u(k-m+1))$$

式中 f 是一个要辨识的未知函数， u 和 y 分别是系统的输入和输出， n 和 m 是正整数。

设计任务是使用模糊系统来辨识未知函数 f 。

模糊控制器设计——梯度下降法例

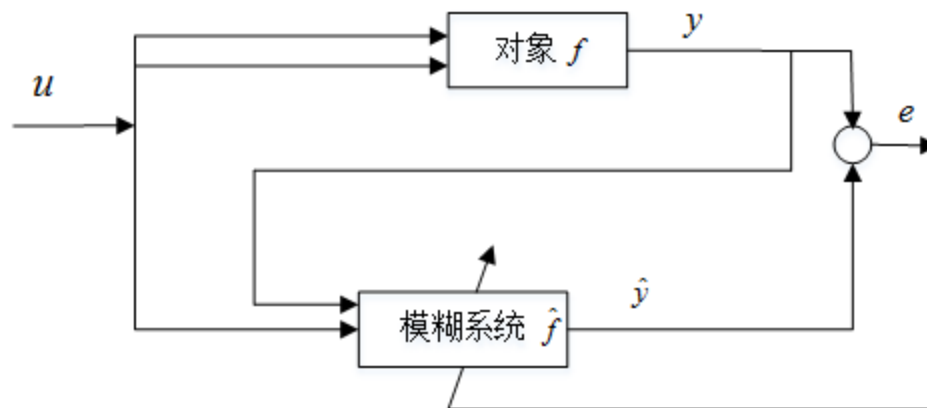
➤ 模糊系统框图

令 $\hat{f}(x)$ 是设计的模糊系统，用 $\hat{f}(x)$ 取代 $f(x)$ ，可得到如下的辨识模型

$$\hat{y}(k+1) = \hat{f}(y(k), \dots, y(k-n+1), u(k), u(k-m+1))$$

需调整 $\hat{f}(x)$ 中的参数，使得 k 趋于无穷时，辨识模型

$\hat{y}(k+1)$ 收敛于真实系统的 $y(k+1)$ 输出，辨识框图如图所示。



◆ 辨识对象满足如下差分方程

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + g[u(k)]$$

式中，未知函数为 $g(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u)$

设辨识模型满足如下的差分方程

$$\hat{y}(k+1) = 0.3y(k) + 0.6y(k-1) + \hat{f}[u(k)]$$

式中， $\hat{f}[*]$ 为所设计的模糊系统， $M = 10$ 。

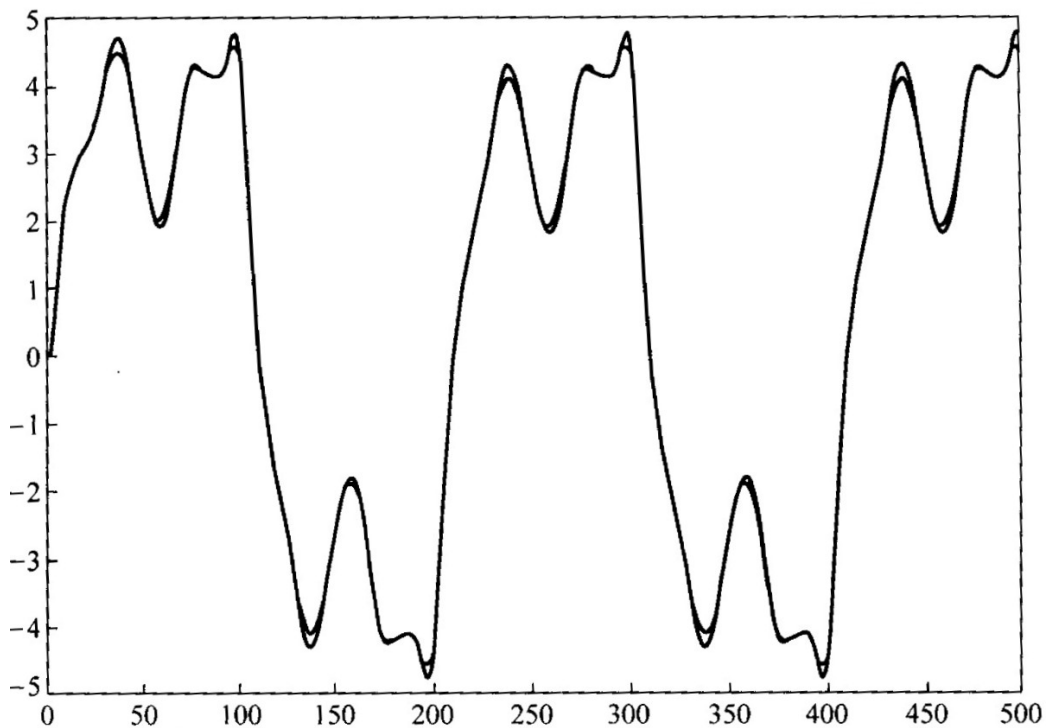
学习算法如前面所示，取 $\alpha = 0.5$ 选用在线参数选择法。辨识器从时刻 $k = 10$ 开始学习，然后调整一个周期内每个时刻点的参数 $\bar{y}^l, \bar{x}_i^l, \sigma_i^l$ 。

下图给出了 $k = 200$ 时系统的输出和辨识模型的输出。

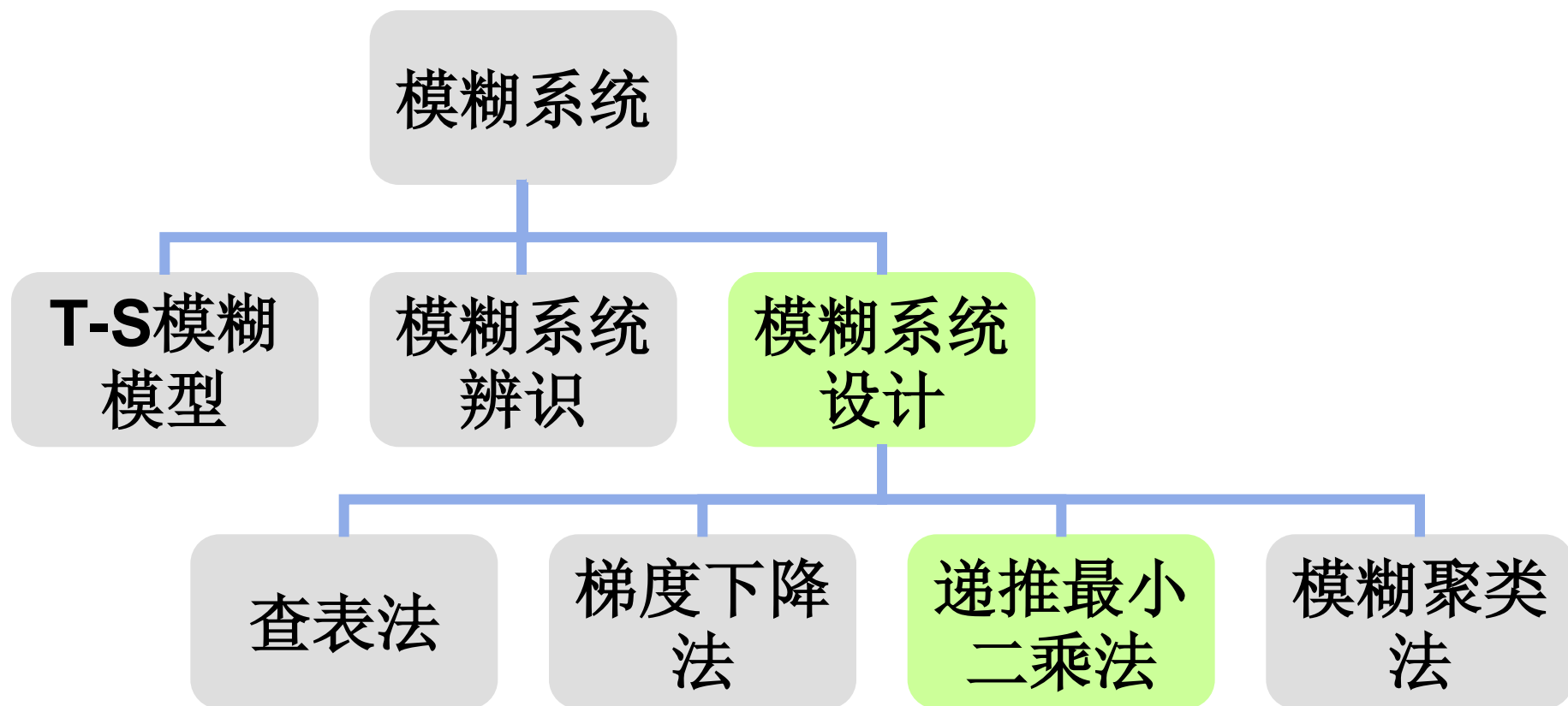
此时的输入为 $u(k) = \sin(2\pi k / 200)$ 。

模糊控制器设计——梯度下降法例

- 辨识模型的输出立刻跟踪了系统输出



3.3 模糊系统辨识--递推最小二乘法



3.3 模糊系统辨识--递推最小二乘法

□ 递推最小二乘法设计模糊系统

- 最小二乘法运算量、存储量大，新观测量致大规模矩阵求逆
- 梯度下降法设计模糊系统力图使得 e^p 达到最小，仅考虑某一输入输出数据对 $(x_0^p; y_0^p)$ 的拟合误差
- 递推最小二乘法模糊系统可使得由 1 到 p 输入输出数据对的拟合误差之和达到最小
- 即模糊系统 $f(x)$ 使下式最小

$$J_p = \sum_{j=1}^p [f(x_0^j) - y_0^j]^2$$

3.3 模糊系统辨识--递推最小二乘法

递推最小二乘模糊系统设计步骤如下所示

步骤1 假设 $U = [\alpha_1, \beta_1] \times \cdots \times [\alpha_n, \beta_n] \subset R^n$ 。在每个区间 $[\alpha_i, \beta_i] (i=1, 2, \dots, n)$ 上定义 N_i 个模糊集 $A_i^{l_i} (l_i=1, 2, \dots, N_i)$ ，它们在 $[\alpha_i, \beta_i]$ 是完备的模糊集。可选 $A_i^{l_i}$ 为四边形模糊集 $\mu_{A_i^{l_i}}(x_i) = \mu_{\mu_{A_i^{l_i}}}(x_i; a_i^{l_i}, b_i^{l_i}, c_i^{l_i}, d_i^{l_i})$ ，其中 $a_i^1 = b_i^1 = \alpha_i$ ， $c_i^j \leq a_i^{j+1} \leq b_i^{j+1} \leq d_i^{j+1} (j=1, 2, \dots, N_i-1)$ $c_i^{N_i} = d_i^{N_i} = \beta_i$ 。

步骤2 根据如下形式的 $\prod_{i=1}^n N_i$ 条模糊IF-THEN规则来构造模糊系统

If x_1 is $A_1^{l_1}$ and \cdots and x_n is $A_n^{l_n}$ then y is $B^{l_1 \cdots l_n}$

其中 $l_i = 1, 2, \dots, N_i (i=1, 2, \dots, n)$ ， $B^{l_1 \cdots l_n}$ 是中心为 $\bar{y}^{l_1 \cdots l_n}$ 的任意模糊集。就是选择带有乘积推理机、单值模糊器、中心平均解模糊器的模糊系统。即所设计的模糊系统为

$$f(x) = \frac{\sum_{l_1=1}^{N_1} \cdots \sum_{l_n=1}^{N_n} \bar{y}^{l_1 \cdots l_n} \left[\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right]}{\sum_{l_1=1}^{N_1} \cdots \sum_{l_n=1}^{N_n} \left[\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right]}$$

3.3 模糊系统辨识--递推最小二乘法

其中 $\bar{y}^{l_1 \dots l_n}$ 是要设计的自由参数, $A_i^{l_i}$ 在步骤1中给定。然后将自由参

数 $\bar{y}^{l_1 \dots l_n}$ 放到 $\prod_{i=1}^n N_i$ 维向量中

$$\underline{\theta} = (\bar{y}^{1 \dots 1}, \dots, \bar{y}^{N_1 1 \dots 1}, \bar{y}^{1 2 1 \dots 1}, \dots, \bar{y}^{N_1 2 1 \dots 1}, \bar{y}^{1 N_2 \dots N_n} \dots, \bar{y}^{N_1 N_2 \dots N_n})^T$$

可得

$$f(x) = b^T(x)\theta$$

其中 $b(x) = (b^{1 \dots 1}, \dots, b^{N_1 1 \dots 1}, b^{1 2 1 \dots 1}, \dots, b^{N_1 2 1 \dots 1}, b^{1 N_2 \dots N_n} \dots, b^{N_1 N_2 \dots N_n})^T$

$$b^{l_1 \dots l_n}(x) = \frac{\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i)}{\sum_{l_1=1}^{N_1} \dots \sum_{l_n=1}^{N_n} \left[\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right]}$$

3.3 模糊系统辨识--递推最小二乘法



步骤3 选择初始参数 $\theta(0)$ ：如果专家（显性知识）能提供 IF 部分的语言规则，则选择 $\bar{y}^{l_1 \cdots l_n}(0)$ 为这些语言规则中的 THEN 部分的模糊集中心；否则在输出空间 $V \subset R$ 上任意选择 $\theta(0)$ 。可见最初的模糊系统是由显性知识组建而成的。



步骤4 当 $p=1, 2, \dots$ 时，用以下的递推最小二乘法计算参数 θ

$$\theta(p) = \theta(p-1) + K(p)[y_0^p - b^T(x_0^p)\theta(p-1)]$$

$$K(p) = P(p-1)b(x_0^p)[b^T(x_0^p)P(p-1)b(x_0^p) + 1]^{-1}$$

$$P(p) = P(p-1) - P(p-1)b(x_0^p)$$

$$[b^T(x_0^p)P(p-1)b(x_0^p) + 1]^{-1}b^T(x_0^p)P(p-1)$$

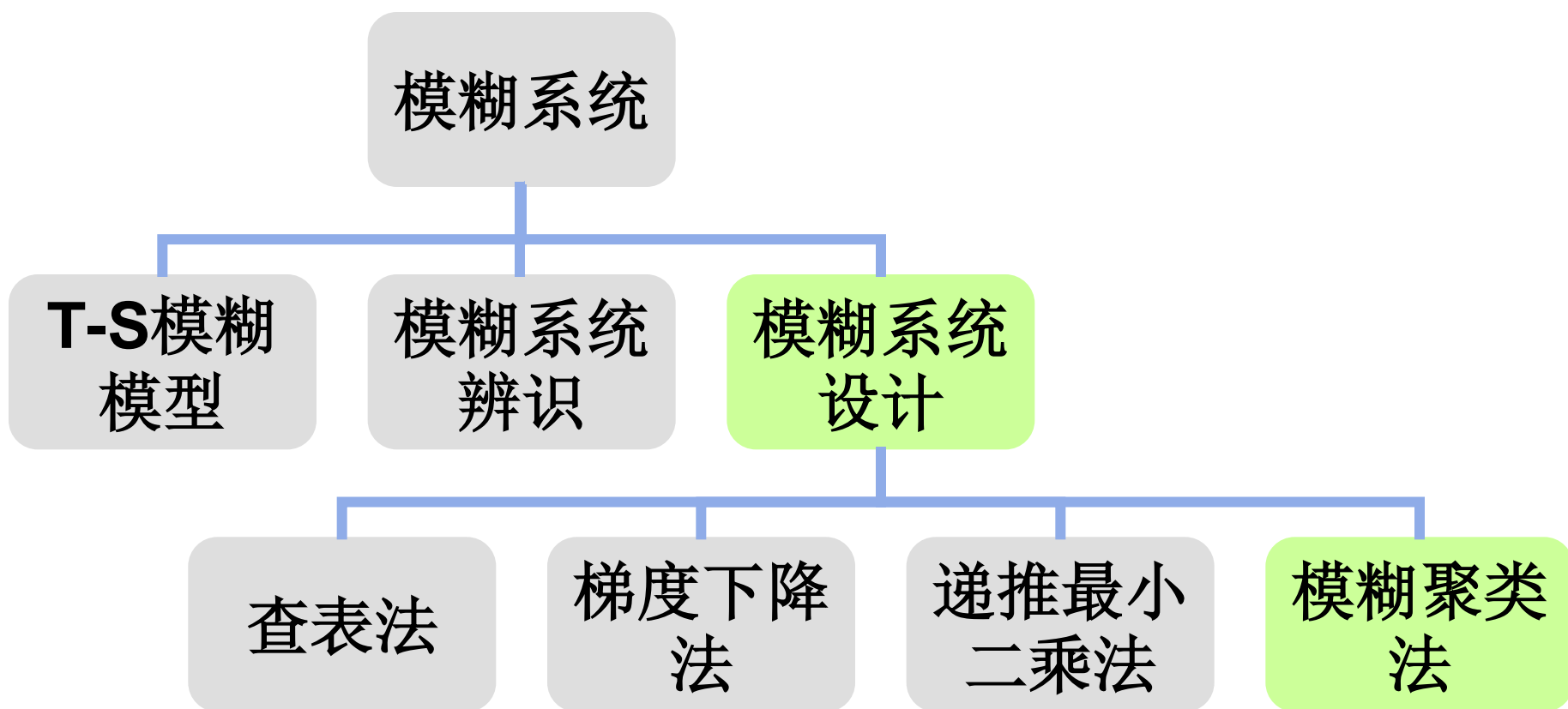
参数矩阵

增益矩阵

协方差矩阵

式中 $\theta(0)$ 是在步骤3中选定的， $P(0)=\sigma I$ （ σ 是一个很大的常数）

3.4 模糊系统设计--模糊聚类法



3.4 模糊系统设计

□ 如何确定模糊系统规则数目？

- 梯度下降法设计模糊系统是在训练前调整规则数目
- 查表法和递推最小二乘法是通过调整if部分的模糊集来依次设定规则数目的边界

□ 一个适当的规则数目是重要的：

- 因为规则太多会令模糊系统变得复杂，并不必须的
- 规则太少会令模糊系统的作用变弱，无效模糊系统无法完成问题目标

3.4 模糊系统设计--模糊聚类法

➤ 将模糊系统规则的数目作为设计参数，并根据输入输出数据对确定规则的数目

基本思想: 将输入输出数据对分成组，一组采用一条模糊规则，即模糊规则的数目等于组的数量

- 构造一个在某种意义上是最优的模糊系统，使其能够以任意精度拟合所有的输入输出数据；
- 即使输入输出数据对的数量少，该模糊系统仍然为有效；
- 采用邻近原则聚类算法将输入输出数据对分组；
- 将每组数据视为一个输入输出数据对，用模糊系统进行拟合。

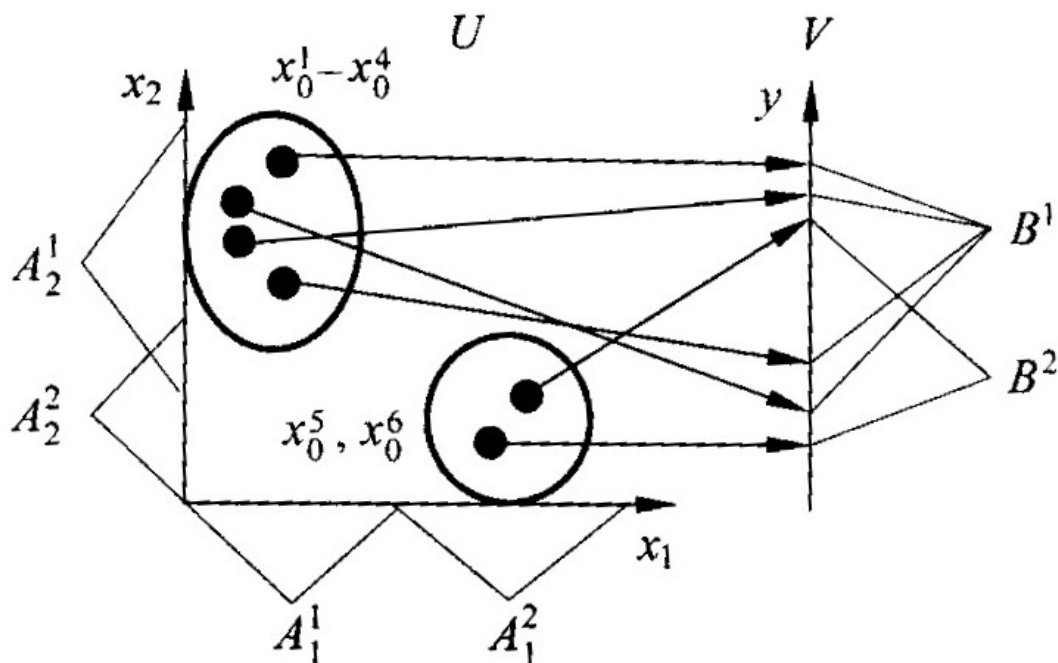
3.4 模糊系统设计--模糊聚类法

□ 聚类——将一个数据集合分割成不相交的子集或组

□ 特点：每组中的数据应具有某些能将它们同其他组数据分开的性质。

□ 方法：

- 首先根据输入点的分布，把输入输出数据对进行分组
- 然后每组仅用一条规则来描述



如果 x_1 为 A_1^1 且 x_2 为 A_2^1 , 则 y 为 B^1

如果 x_1 为 A_1^2 且 x_2 为 A_2^2 , 则 y 为 B^2

注：图中的六对输入—输出数据 $(x_0^1; y_0^1), \dots, (x_0^6; y_0^6)$ 被分为两组，并由此产生上面的两条规则。

3.4 模糊系统设计--模糊聚类法

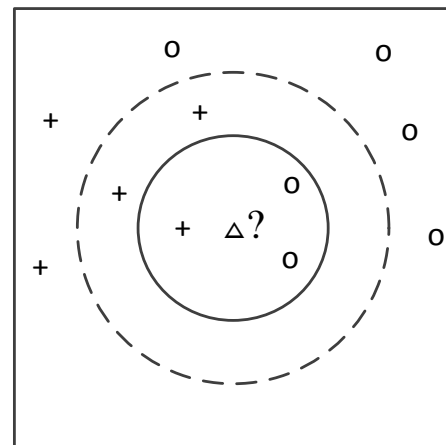
□ 最近邻聚类是一种简单易操作的聚类方法

o和+为已确定类别的训练样本集， Δ 为待测试样本，根据 k 邻近方法可以预测分类结果：

□ 当 $k=3$ 时，与待测试样本邻近的3个训练样本集中，有2个o样本和1个+样本，根据邻近样本中出现最多的类别作为标记时，可划分为o类，如图中实线圆区域；

□ 当 $k=5$ 时，与待测试样本邻近的5个训练样本集中，有2个o样本和3个+样本，根据邻近样本中出现最多的类别作为标记时，应将测试样本划分为+类，如图中虚线圆区域。

□ k 邻近聚类算法中， k 值的选择较为关键，例如，当 $k=11$ 时，邻近对象中有6个o样本和5个+样本，按照最多类别划分原则，测试样本将再次划入当 $k=3$ 时划分的o类。



k 邻近聚类原理示意图

3.4 模糊系统设计--模糊聚类法

基于最近邻聚类法设计模糊系统设计步骤如下

步骤1 从第一个输入输出数据对 $(x_0^1; y_0^1)$ 开始, 把 x_0^1 做为一个聚类中心 x_c^1 并令 $A^1(1) = y_0^1$, $B(1) = 1$, 设半径为 r 。

步骤2 假设考虑第 k 个输入输出数据对 $(x_0^k; y_0^k)$ ($k = 2, 3, \dots$) 时, 已经存在聚类中心分别为 $x_c^1, x_c^2, \dots, x_c^M$ 的 M 个聚类。分别计算 x_0^k 到这 M 个聚类中心的距离 $|x_0^k - x_c^l|$ ($l = 1, 2, \dots, M$)。设这些距离最小的距离为 $|x_0^k - x_c^{l_k}|$, 即 $x_c^{l_k}$ 为 x_0^k 的最近邻原则聚类, 则:

(a) 如果 $|x_0^k - x_c^{l_k}| > r$, 则把 x_0^k 做为新的聚类中心 $x_c^{M+1} = x_0^k$,

令 $A^{M+1}(k) = y_0^k$, $B^{M+1}(k) = 1$, 并令 $A^l(k) = A^l(k-1)$,

$B^l(k) = B^l(k-1)$ ($l = 1, 2, \dots, M$)。

(b) 如果 $|x_0^k - x_c^{l_k}| \leq r$, 则做如下计算:

$A^{l_k}(k) = A^{l_k}(k-1) + y_0^k$ 和 $B^{l_k}(k) = B^{l_k}(k-1) + 1$

当 $l \neq l_k, l = 1, 2, \dots, M$ 时

$A^l(k) = A^l(k-1)$ 和 $B^l(k) = B^l(k-1)$

3.4 模糊系统设计--模糊聚类法



步骤3 如果 x_0^k 并未建立一个新的聚类, 则根据 k 对输入输出数据 $(x_0^i; y_0^i) (i=1, 2, \dots, k)$ 设计模糊系统如下

$$f_k(x) = \frac{\sum_{l=1}^M A^l(k) \exp\left(-\frac{|x - x_c^l|}{\sigma}\right)^2}{\sum_{l=1}^M B^l(k) \exp\left(-\frac{|x - x_c^l|}{\sigma}\right)^2}$$

如果 x_0^k 建立了一个新的聚类, 则所设计的模糊系统为

$$f_k(x) = \frac{\sum_{l=1}^{M+1} A^l(k) \exp\left(-\frac{|x - x_c^l|}{\sigma}\right)^2}{\sum_{l=1}^{M+1} B^l(k) \exp\left(-\frac{|x - x_c^l|}{\sigma}\right)^2}$$



步骤4 令 $k=k+1$, 返回步骤2。

3.4 模糊系统设计--模糊聚类法

- 由于每个输入输出数据对都有可能产生新的聚类，因此所设计的模糊系统中规则的数目在设计中也是不断变化的
- 规则的数目取决于输入输出数据对中输入点的分布以及半径 r
- 半径 r 决定了模糊系统的复杂性
- r 越小，所得到组的数目就越多，从而使得模糊系统的越复杂
- 当 r 较大时，所设计的模糊系统会比较简单但缺乏力度
- 实践过程中，可通过试错法找到一个合适的 r

模糊-神经网络组合应用

模糊-神经网络组合

```
graph TD; A[模糊-神经网络组合] --- B[基于标准模型的FNN]; A --- C[基于T-S模型的FNN];
```

基于标准模型的FNN

基于T-S模型的FNN

模糊系统的主要特点

- 表达基于规则的知识、利用已有的经验知识
- 适合表达那些模糊或定性的知识
- 推理方式类似人类思维模式
- 处理不确定性、非线性和其它问题的有力工具
- 缺乏自学习和自适应能力
- 语言型推理控制

神经网络的主要特点

- 并行计算、分布式信息存贮
- 容错能力强
- 具有自适应学习能力
- 初始权值常取为零或随机数,训练时间较长,易陷入局部极小值
- 映射型

模糊-神经网络组合应用 概述

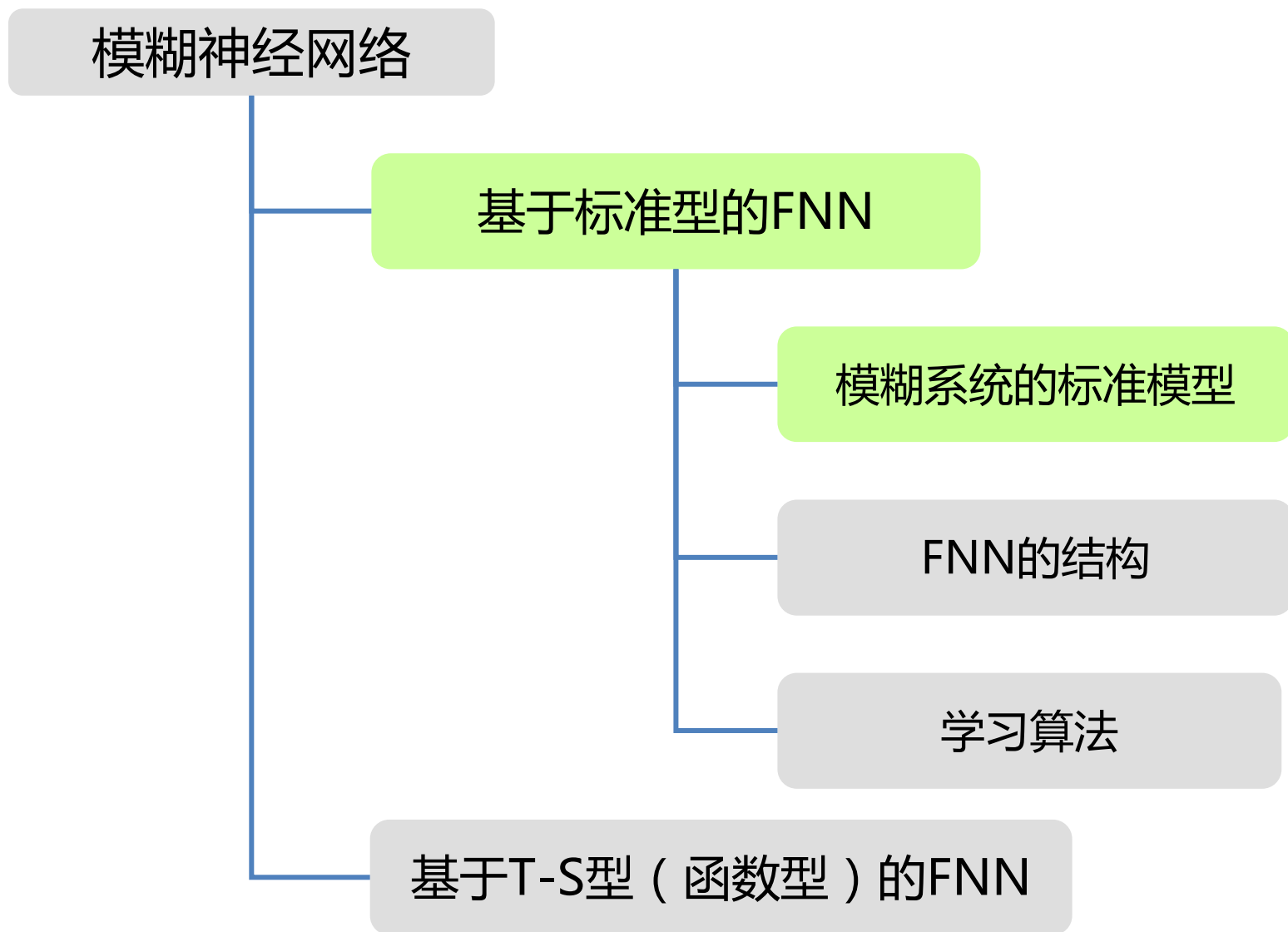
计算智能的主要特征

- 无需精确数学模型
- 强容错能力
- 可硬件实现

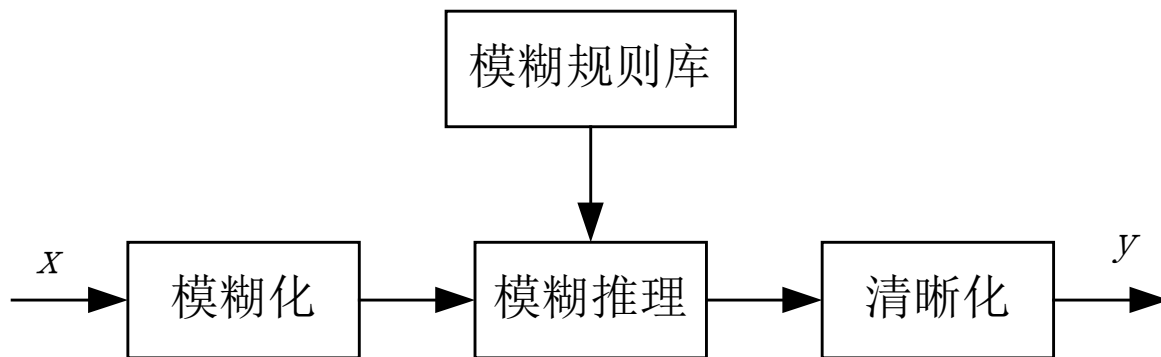
模糊-神经网络组合应用的显著特点：

- 提高了模糊系统的自适应能力
- 改善了神经网络系统的全局性能，加强了可观测性
- 已有知识获得应用
- 强映射能力得到运用

模糊-神经网络组合应用——FNN



模糊系统的标准型



基于标准型的模糊系统原理结构图

模糊-神经网络组合应用 模糊系统的标准型 (2)

设输入向量 $x = [x_1 \ x_2 \ \cdots \ x_n]^T$, 每个分量 x_i 均为模糊语言变量, 并设 $T(x_i) = \{A_i^1, A_i^2, \cdots A_i^{m_i}\}$, $i = 1, 2, \dots, n$.

其中, $A_i^j (j = 1, 2, \dots, m_i)$ 是 x_i 的第 j 个语言变量值, 相应的隶属度函数为 $\mu_{A_i^j} (j = 1, 2, \dots, n; j = 1, 2, \dots, m_i)$.

输出量 y 也是模糊语言变量且 $T(y) = \{B^1, B^2, \dots, B^{m_y}\}$

其中 $B^j (j = 1, 2, \dots, m_y)$ 是 y 的第 j 个语言变量值, 相应的隶属度函数为 $\mu_{B^j}(y)$.

设描述输入输出关系的模糊规则为：

R^i ：如果 x_1 是 A_1^i and x_2 是 A_2^i ...and x_n 是 A_n^i ，则 y 是 B^i ，
其中 $i=1, 2, \dots, m$ 。

若输入量采用单点模糊集合模糊化方法，则对于给定的输入 x ，可以求得对于每条规则的适用度为：

$$\alpha_i = \mu_{A_1^i}(x_1) \wedge \mu_{A_2^i}(x_2) \wedge \dots \wedge \mu_{A_n^i}(x_n)$$

通过模糊推理可得对于每一条模糊规则的输出量模糊集合的隶属度函数为：

$$\mu_{B_i}(y) = \alpha_i \wedge \mu_{B^i}(y)$$

模糊-神经网络组合应用 模糊系统的标准型 (4)

从而输出量总的模糊集合为 $B = \bigcup_{i=1}^m B_i$

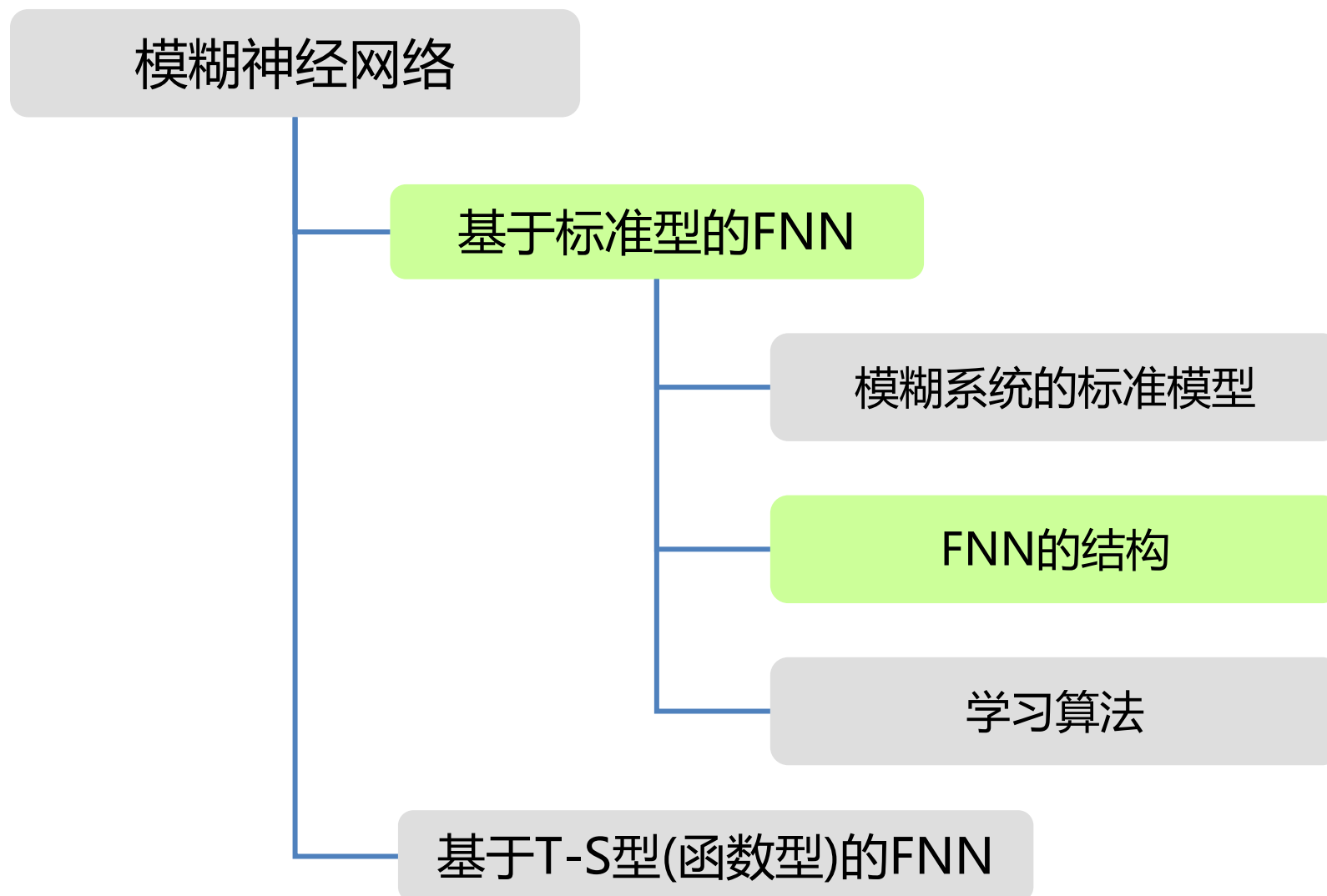
$$\mu_B(y) = \bigvee_{i=1}^m \mu_{B_i}(y)$$

若采用加权平均的清晰化方法，则可求得输出的清晰化量为

$$y = \frac{\sum_{i=1}^m y_{c_i} \alpha_i}{\sum_{i=1}^m \alpha_i}$$

y_{c_i} 是使 $\mu_{B_i}(y)$ 取最大值的点，一般为隶属度函数的中心点。

模糊-神经网络组合应用——FNN

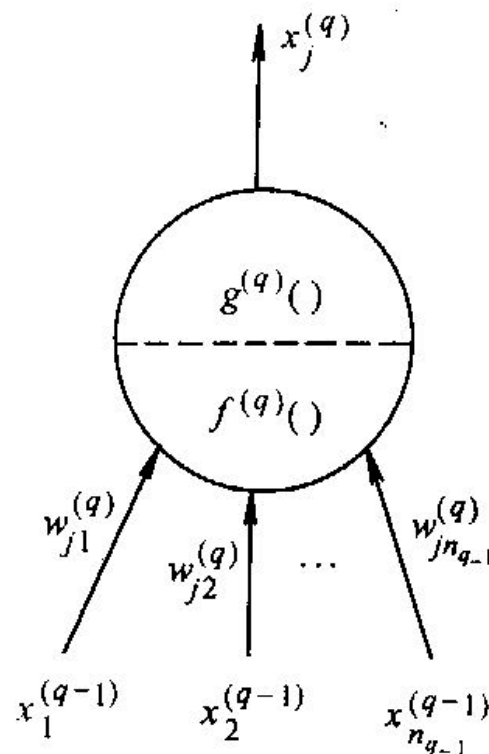


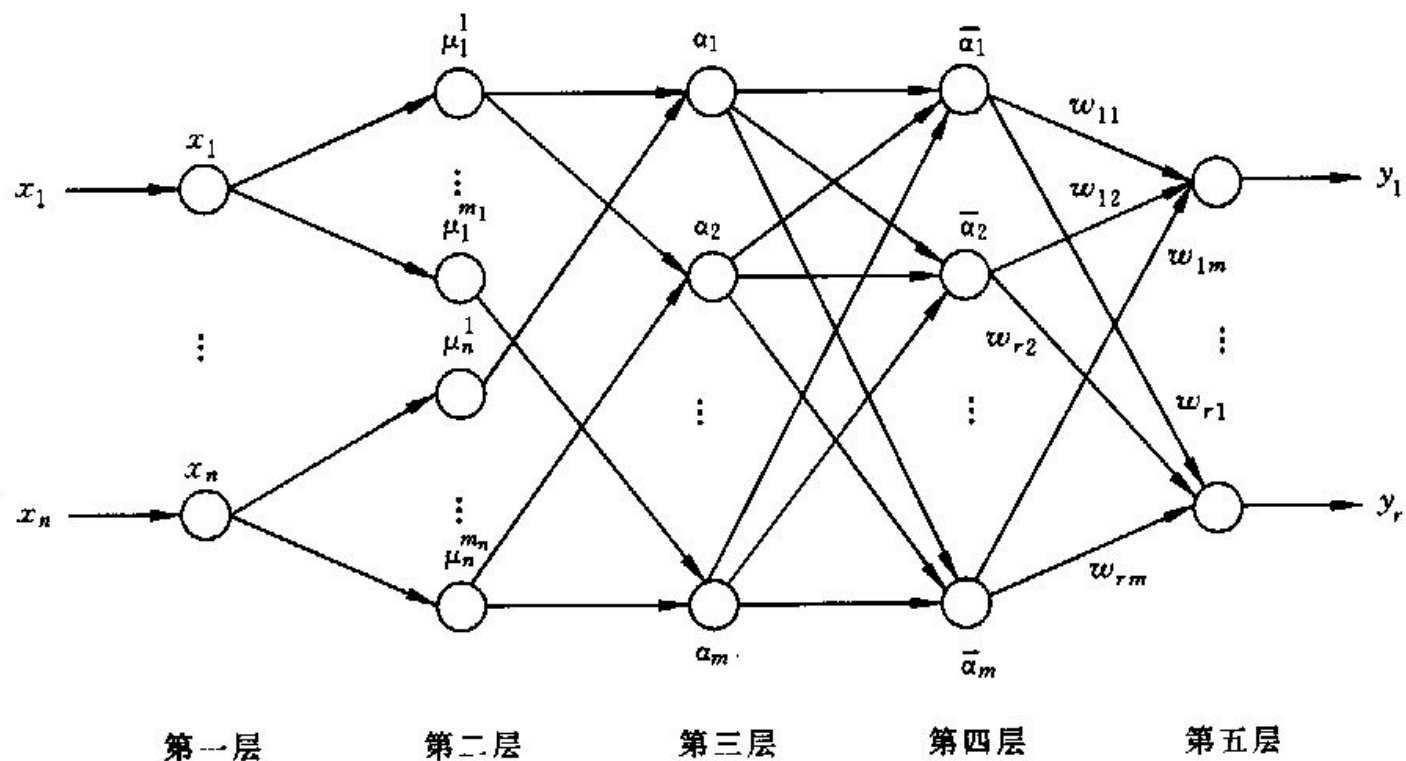
单个神经元结点的基本结构

第 q 层第 j 个结点的输入和输出

$$f_j^{(q)} = \sum_{i=1}^{n_{q-1}} w_{ji}^{(q)} x_i^{(q-1)}$$

$$x_j^{(q)} = g^{(q)}(f_j^{(q)})$$



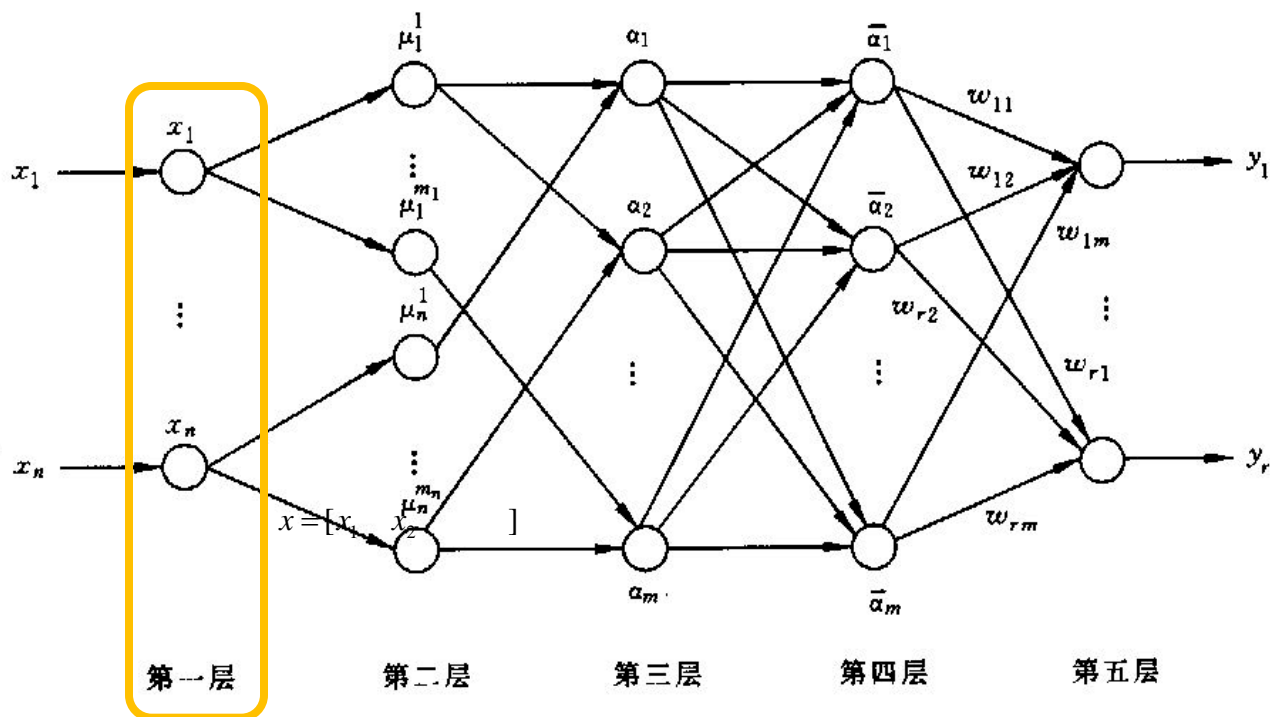


基于标准模型的FNN结构

第一层

- 输入层
- 将输入值 x 传送到下一层
- 节点数为 n

节点函数:



$$f_i^{(1)} = x_i^{(0)} = x_i$$

$$x_i^{(1)} = g_i^{(1)}(f_i^{(1)}) = f_i^{(1)}$$

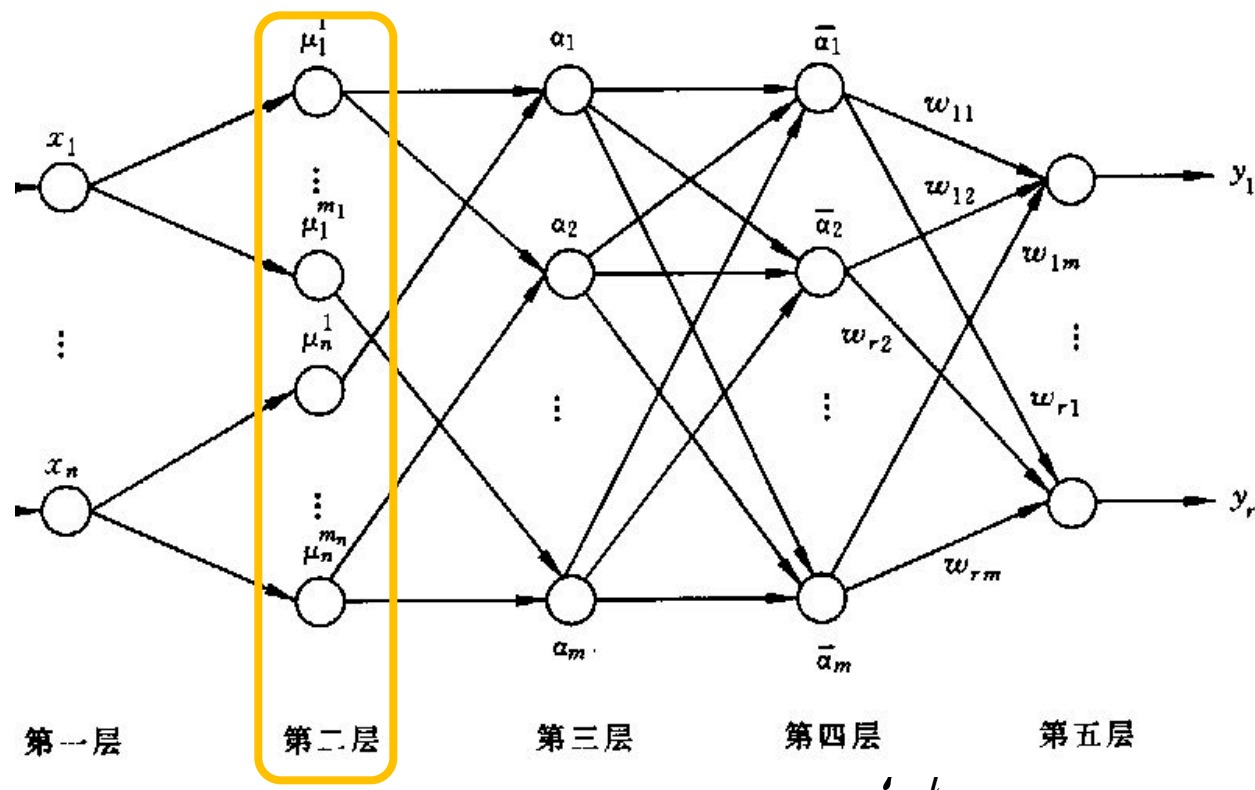
$$i = 1, 2, \dots, n$$

$$x = [x_1 \ x_2 \ \dots \ x_n]^T$$

第二层

- 模糊量化层
- 计算输入值属于模糊集的隶属度
- 可为高斯函数

节点函数:



$$f_{ij}^{(2)} = -\frac{(x_i^{(1)} - c_{ij})^2}{\sigma_{ij}^2}$$

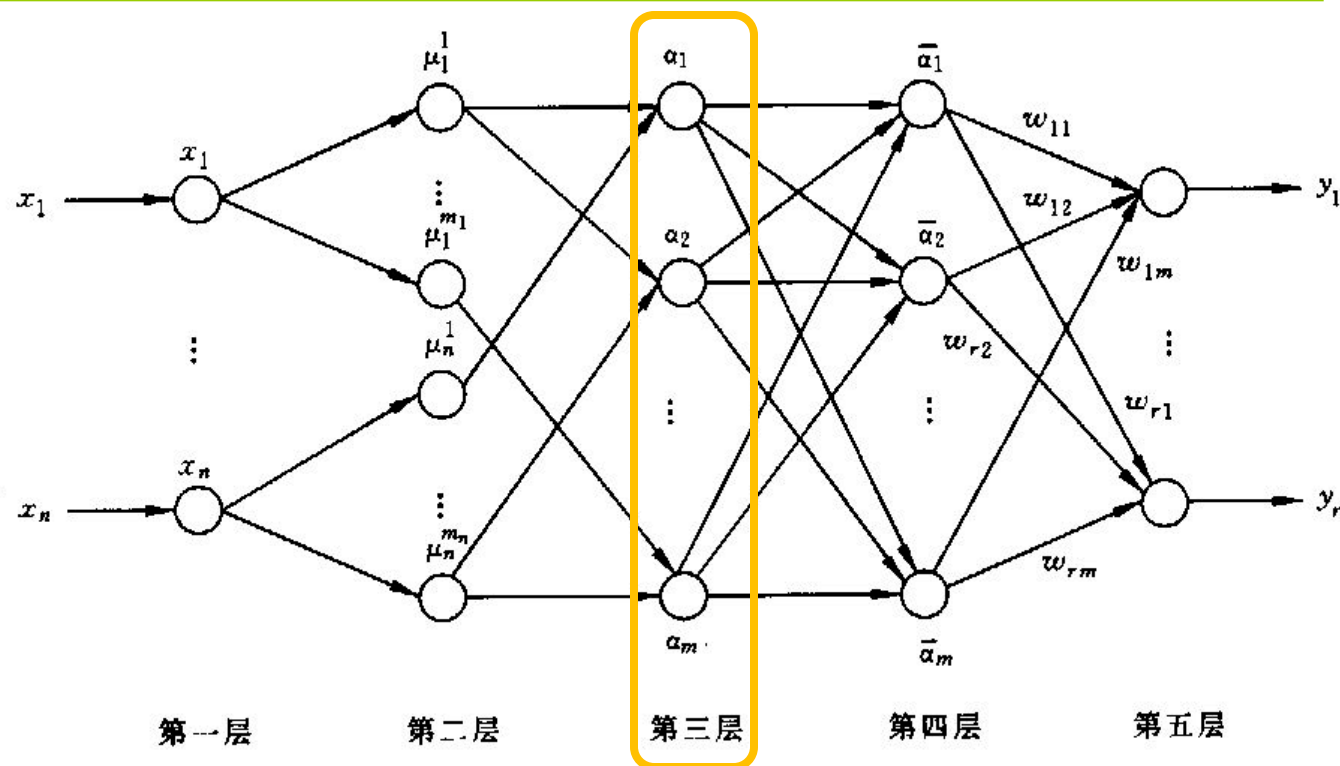
$$x_{ij}^{(2)} = \mu_i^j = g_{ij}^{(2)}(f_{ij}^{(2)}) = e^{f_{ij}^{(2)}} = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}$$

$$i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m_i$$

第三层

- 规则推理层
- 计算每条规则的适用度

节点函数



$$f_j^{(3)} = \min\{\mu_1^{j_1}, \mu_2^{j_2}, \dots, \mu_n^{j_n}\}$$

或
$$f_j^{(3)} = \mu_1^{j_1} \mu_2^{j_2} \dots \mu_n^{j_n}$$

$$x_j^{(3)} = \alpha_j = g_j^{(3)}(f_j^{(3)}) = f_j^{(3)}$$

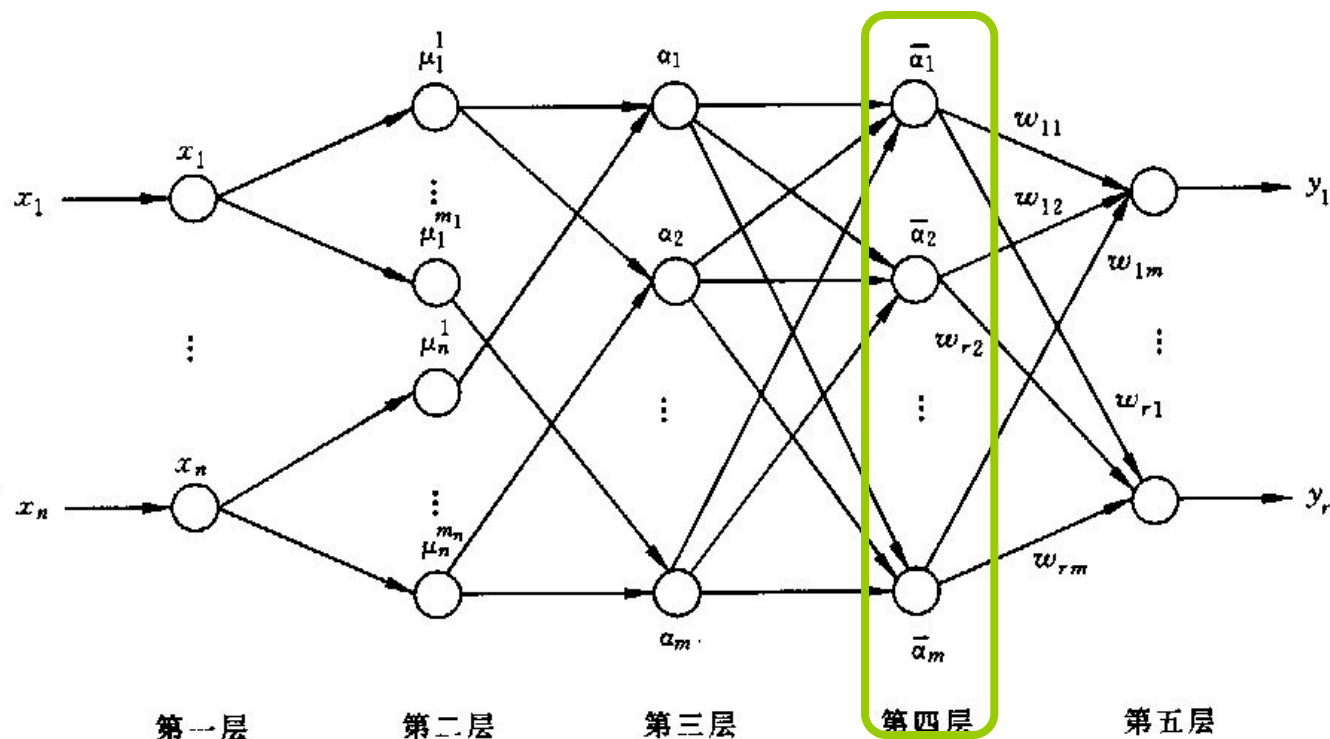
$$j = 1, 2, \dots, m$$

$$m = \prod_{i=1}^n m_i$$

第四层

- 规则合成层
- 实现归一化计算

节点函数



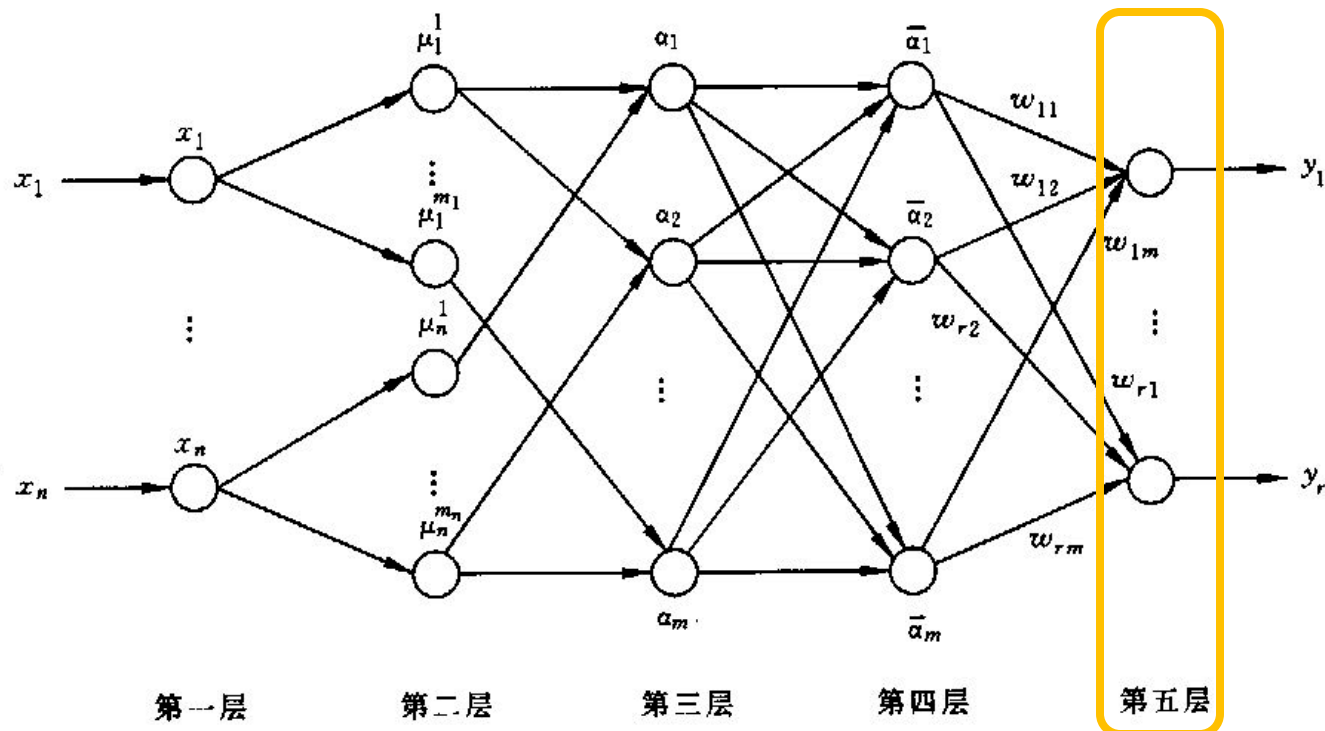
$$f_j^{(4)} = x_j^{(3)} / \sum_{i=1}^m x_i^{(3)} = \alpha_j / \sum_{i=1}^m \alpha_i, \quad \bar{\alpha}_j = \frac{\alpha_j}{\sum_{i=1}^m \alpha_i}, \quad j = 1, 2, \dots, m$$

$$x_j^{(4)} = \bar{\alpha}_j = g_j^{(4)}(f_j^{(4)}) = f_j^{(4)} \quad j = 1, 2, \dots, m$$

第五层

- 输出层
- 实现清晰化计算

节点函数

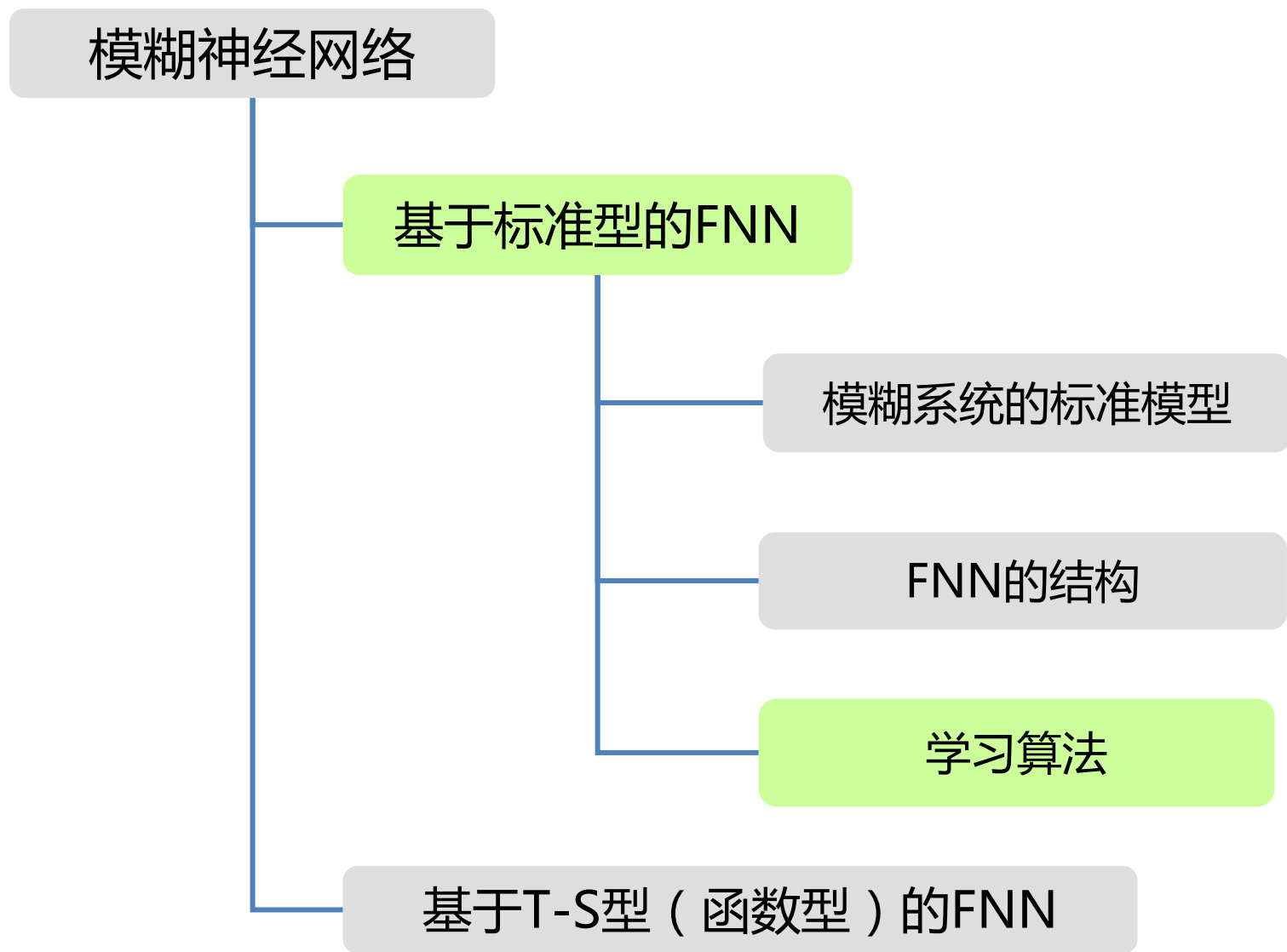


$$f_i^{(5)} = \sum_{j=1}^m w_{ij} x_j^{(4)} = \sum_{j=1}^m w_{ij} \bar{\alpha}_j, \quad y_i = \sum_{j=1}^m w_{ij} \bar{\alpha}_j, i=1, 2, \dots, r$$

$$x_i^{(5)} = y_i = g_i^{(5)}(f_i^{(5)}) = f_i^{(5)}$$

$$i=1, 2, \dots, r$$

模糊-神经网络组合应用——FNN



基于标准模型的FNN--学习算法

□ 设误差代价函数为：

$$E = \frac{1}{2} \sum_{i=1}^r (y_{di} - y_i)^2$$

式中： y_{di} 为期望输出， y_i 为实际输出。

□ 计算第五层反向传播误差：

$$\delta_i^{(5)} = -\frac{\partial E}{\partial f_i^{(5)}} = -\frac{\partial E}{\partial y_i} = y_{di} - y_i$$

权值的一阶梯度为： $\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial f_i^{(5)}} \frac{\partial f_i^{(5)}}{\partial w_{ij}} = -\delta_i^{(5)} x_j^{(4)} = -(y_{di} - y_i) \bar{\alpha}_j$

基于标准模型的FNN--学习算法 (2)

□ 计算其他三层反向传播误差：

$$\delta_j^{(4)} = -\frac{\partial E}{\partial f_j^{(4)}} = -\sum_{k=1}^r \frac{\partial E}{\partial f_i^{(5)}} \frac{\partial f_i^{(5)}}{\partial g_j^{(4)}} \frac{\partial g_j^{(4)}}{\partial f_j^{(4)}} = \sum_{i=1}^r \delta_i^{(5)} w_{ij}$$

$$\delta_j^{(3)} = -\frac{\partial E}{\partial f_j^{(3)}} = -\sum_{i=1}^r \frac{\partial E}{\partial f_i^{(4)}} \frac{\partial f_i^{(4)}}{\partial g_j^{(3)}} \frac{\partial g_j^{(3)}}{\partial f_j^{(3)}} = \frac{1}{\left(\sum_{k=1}^m \alpha_i\right)^2} \left(\delta_j^{(4)} \sum_{\substack{i=1 \\ i \neq j}}^m \alpha_i - \sum_{\substack{k=1 \\ k \neq j}}^m \delta_k^{(4)} \alpha_k \right)$$

$$\delta_{ij}^{(2)} = -\frac{\partial E}{\partial f_{ij}^{(2)}} = -\sum_{k=1}^r \frac{\partial E}{\partial f_i^{(3)}} \frac{\partial f_i^{(3)}}{\partial g_j^{(2)}} \frac{\partial g_j^{(2)}}{\partial f_{ij}^{(2)}} = \sum_{k=1}^m \delta_k^{(3)} s_{ij} e^{f_{ij}^{(2)}} = \sum_{k=1}^m \delta_k^{(3)} s_{ij} e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}$$

基于标准模型的FNN--学习算法 (3)

□ 当 $f^{(3)}$ 采用取小运算，则 μ_i^j 是第 k 个规则结点输入的最小值：

$$S_{ij} = \frac{\partial f_k^{(3)}}{\partial g_{ij}^{(2)}} = \frac{\partial f_k^{(3)}}{\partial \mu_i^j} = 1$$

否则, 当 $f^{(3)}$ 采用相乘运算，则 μ_i^j 是第 k 个规则结点的一个输入：

$$S_{ij} = \prod_{\substack{j=1 \\ j \neq i}}^n \mu_j^{i_j}$$

否则 $S_{ij} = 0$

基于标准模型的FNN--学习算法 (4)

□ 高斯隶属函数参数的一阶梯度为：

$$\frac{\partial E}{\partial c_{ij}} = \frac{\partial E}{\partial f_{ij}^{(2)}} \frac{\partial f_{ij}^{(2)}}{\partial c_{ij}} = -\delta_{ij}^{(2)} \frac{2(x_i - c_{ij})}{\sigma_{ij}^2}$$

$$\frac{\partial E}{\partial \sigma_{ij}} = \frac{\partial E}{\partial f_{ij}^{(2)}} \frac{\partial f_{ij}^{(2)}}{\partial \sigma_{ij}} = -\delta_{ij}^{(2)} \frac{2(x_i - c_{ij})^2}{\sigma_{ij}^2}$$

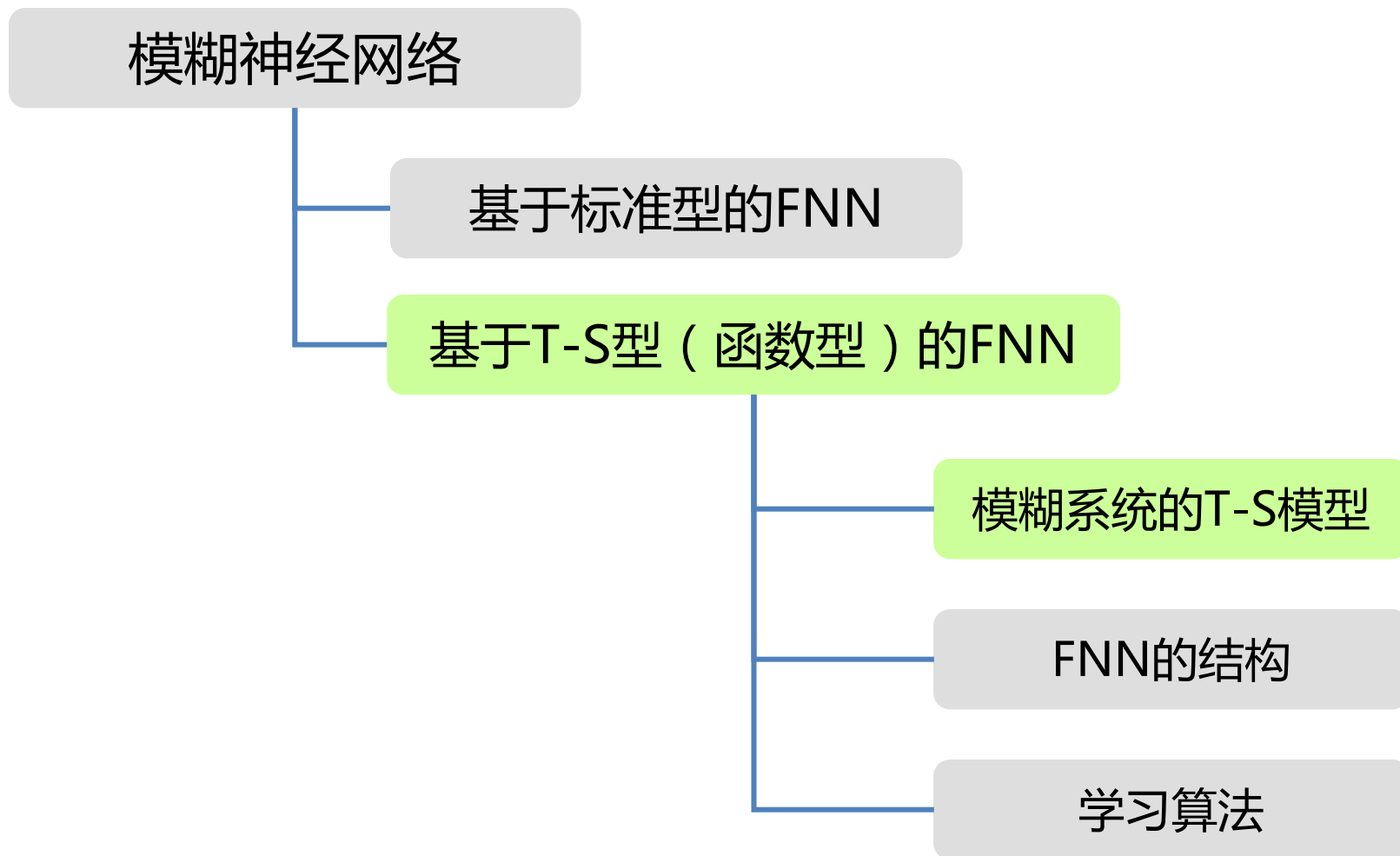
基于标准模型的FNN--学习算法 (5)

➤ 参数调整的学习算法如下：

$$w_{ij}(k+1) = w_{ij}(k) - \beta \frac{\partial E}{\partial w_{ij}} \quad i = 1, 2, \dots, r \quad j = 1, 2, \dots, m$$

$$c_{ij}(k+1) = c_{ij}(k) - \beta \frac{\partial E}{\partial c_{ij}} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m_i$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \beta \frac{\partial E}{\partial \sigma_{ij}} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m_i$$



□ 模糊系统的T-S模型

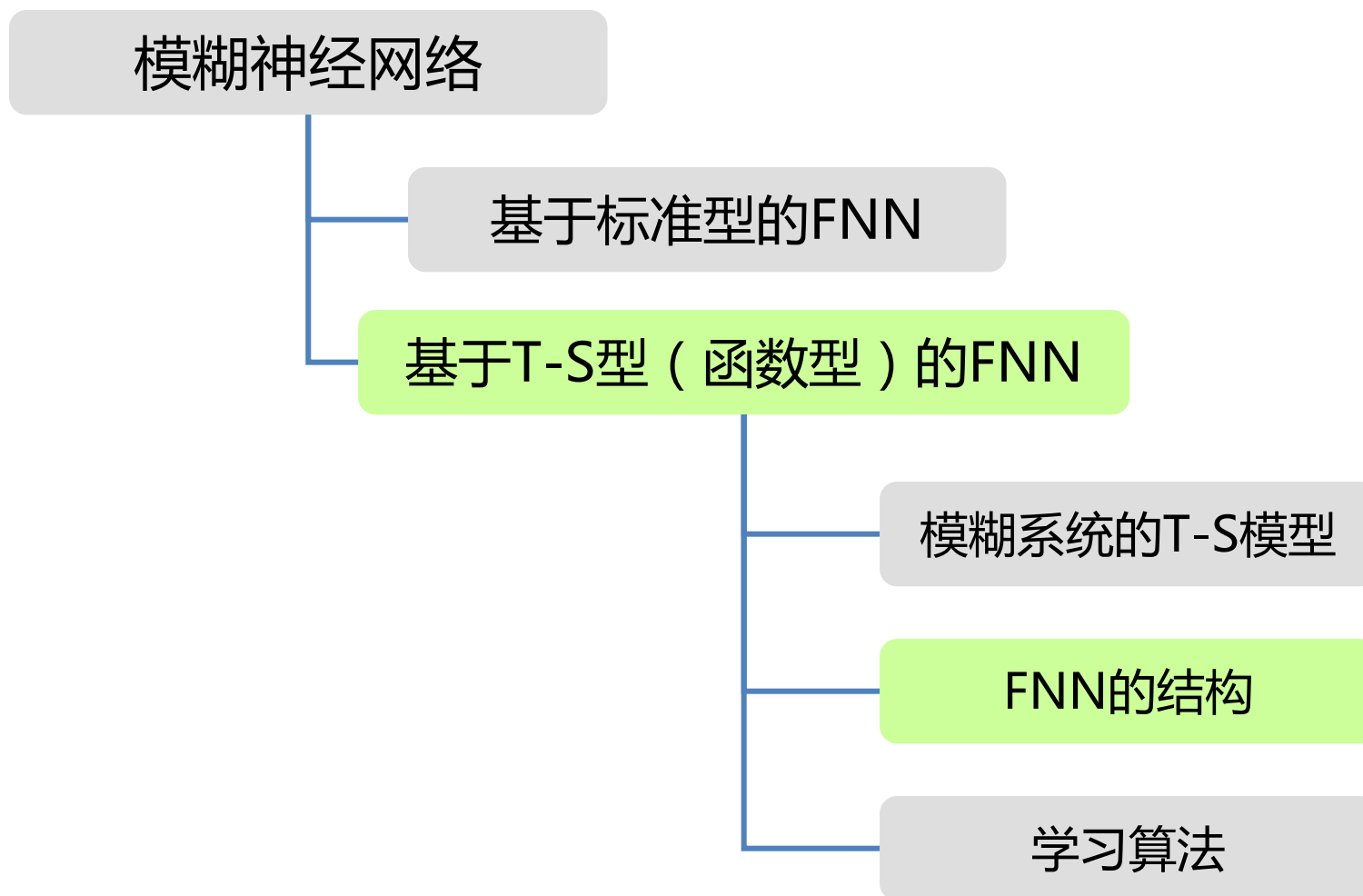
$R^i : \text{If } x_1 \text{ is } A_1^i, \quad x_2 \text{ is } A_2^i, \dots, \quad x_m \text{ is } A_m^i,$

$\text{Then } y^i = P_0^i + P_1^i x_1 + P_2^i x_2 + \dots + P_m^i x_m$

$$\hat{y} = \frac{\sum_{i=1}^n G^i y^i}{\sum_{i=1}^n G^i}$$

其中, $G^i = \prod_{j=1}^m A_j^i(x_j^0)$

Π 为模糊化算子, 通常由取小或代数积计算.

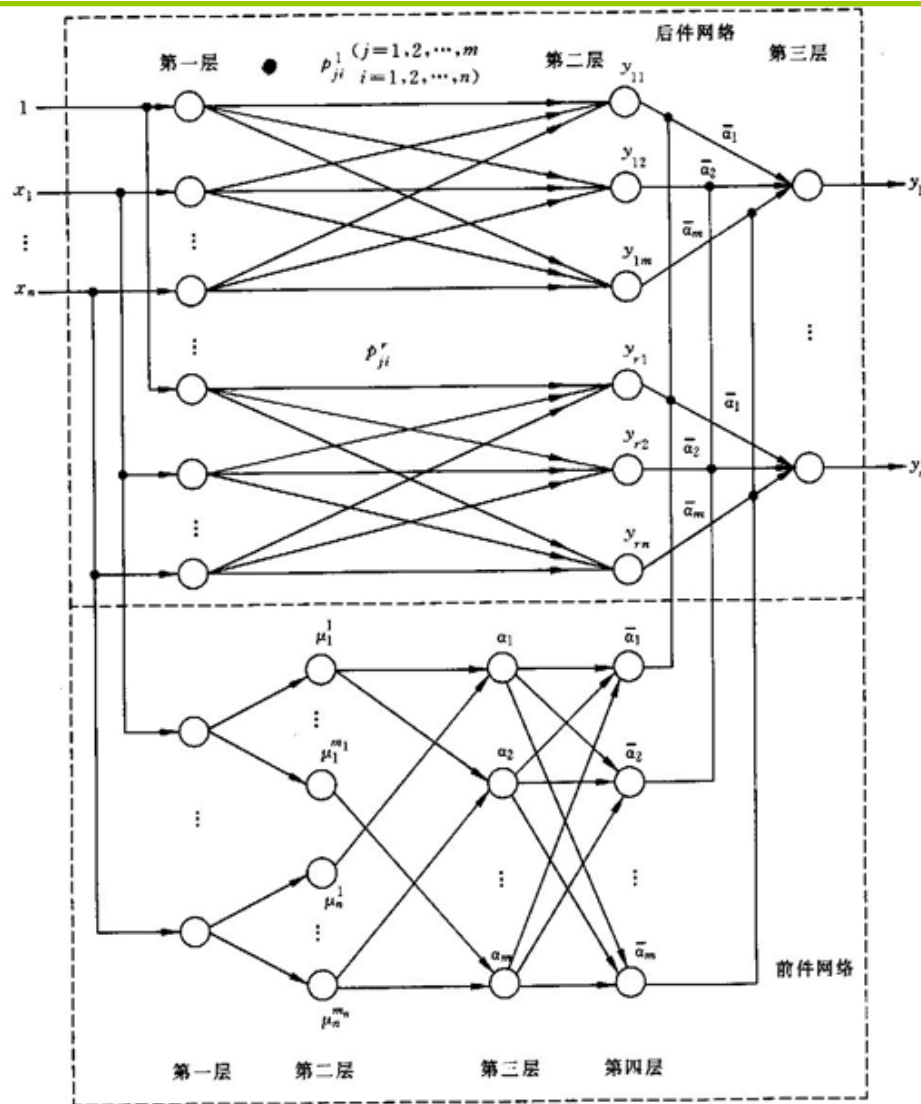


□ T-S型的FNN结构

由前件网络和后件网络两部分组成:

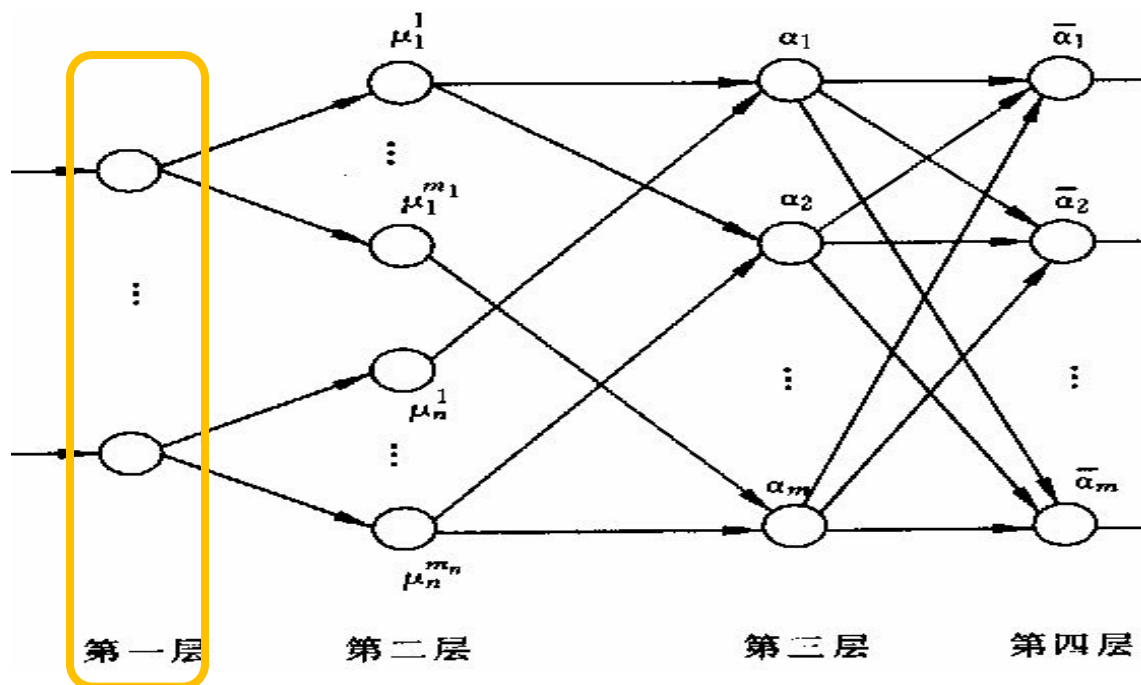
- 前件网络是用来匹配模糊规则的前件
- 后件网络是用来产生模糊规则的后件

模糊-神经网络组合应用 函数型FNN的结构 (2)



基于T-S模型的FNN结构

模糊-神经网络组合应用 函数型FNN的结构 (3)



◆ 前件网络

➤ 第一层

输入层, 该层将输入值 $x = [x_1 \ x_2 \ \cdots \ x_n]^T$ 传送到下一层.
该层的结点数 $N_1 = n$.

➤ 第二层

每个结点代表一个语言变量值, 作用为计算各输入分量属于各语言变量值模糊集合的隶属度函数:

$$\mu_i^j \equiv \mu_{A_i^j}(x_i) \quad i=1, 2, \dots, n, j=1, 2, \dots, m_i$$

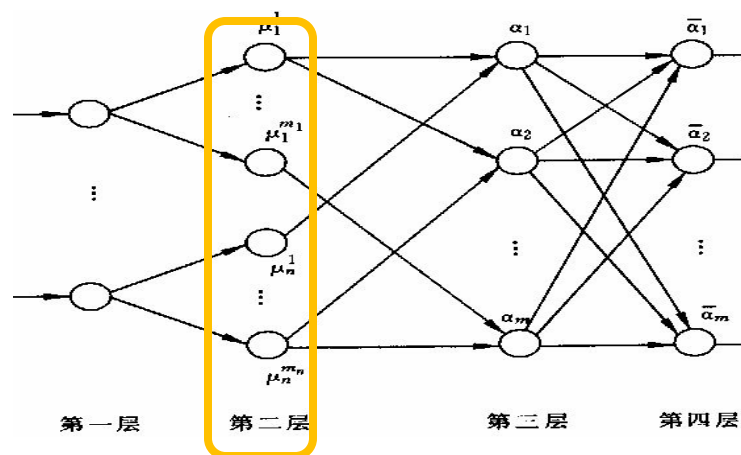
其中, n 是输入量的维数, m_i 是 x_i 的模糊分割数.

若隶属函数采用高斯函数, 则

$$\mu_i^j = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}$$

该层的节点总数:

$$N_2 = \sum_{i=1}^n m_i$$



➤ 第三层

每个结点代表一条模糊规则，用来匹配模糊规则的前件，计算出每条规则的适用度：

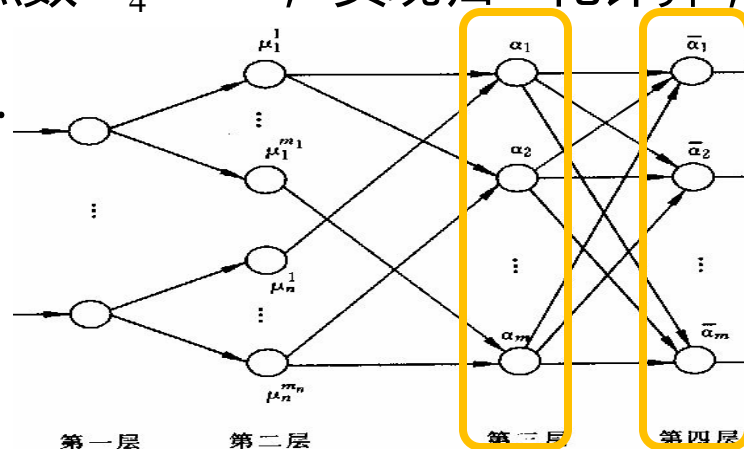
$$\alpha_j = \mu_1^{i_1} \mu_2^{i_2} \cdots \mu_n^{i_n}$$

其中, $i_1 \in \{1, 2, \cdots, m_1\}$, $i_2 \in \{1, 2, \cdots, m_2\}$, $i_n \in \{1, 2, \cdots, m\}$,

$$j = 1, 2, \cdots, m, \quad m = \prod_{i=1}^n m_i$$

该层的结点总数 $N_3 = m$, 第四层的节点数 $N_4 = m$, 实现归一化计算, 即

$$\bar{\alpha}_j = \frac{\alpha_j}{\sum_{i=1}^m \alpha_i}, \quad j = 1, 2, \cdots$$

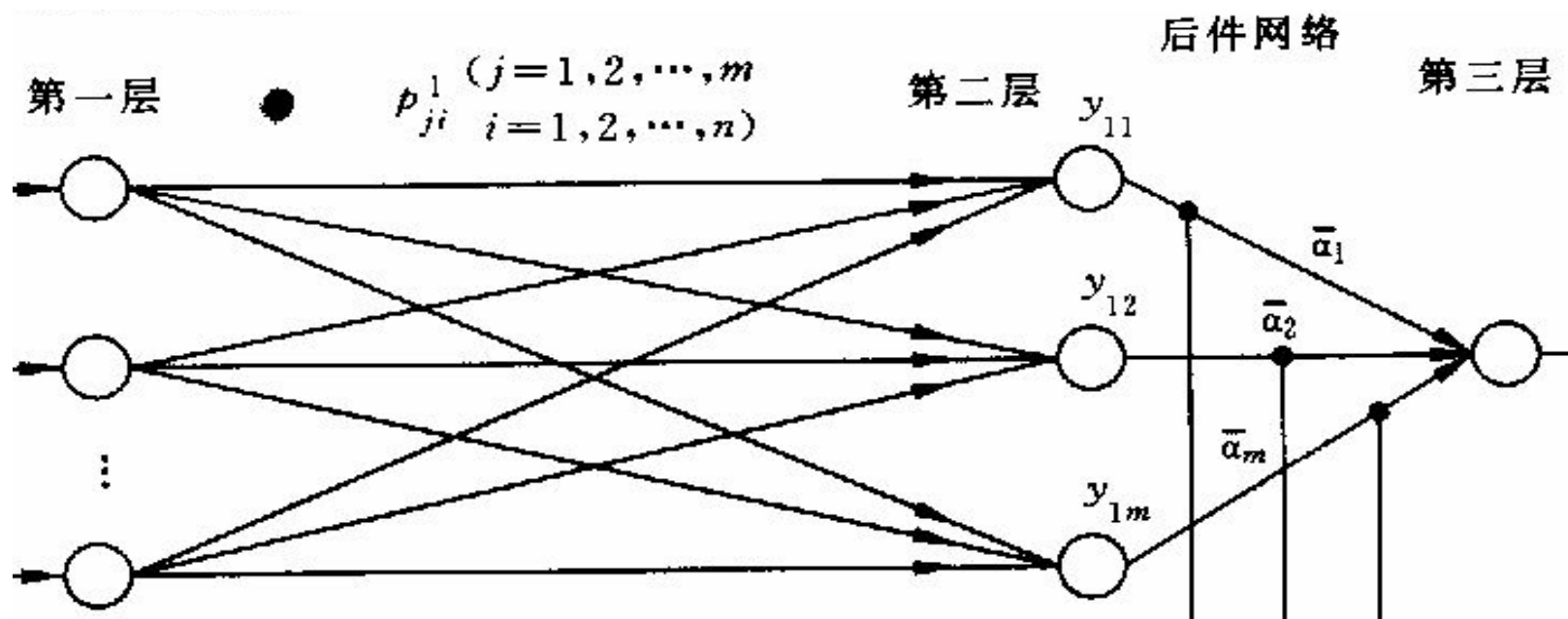


◆ 后件网络

由 r 个结构相同的并联子网络组成，每个子网络产生一个输出量。

➤ 第一层

输入层，将输入变量传送到第二层。输入层中第0个结点的输入值 $x_0 = 1$ ，作用是提供模糊规则后件中的常数项。



➤ 第二层

m 个结点，每个结点代表一条规则，作用为计算每一条规则的结论，即：

$$y_{ij} = p_{j0}^i + p_{j1}^i x_1 + \cdots + p_{jn}^i x_n = \sum_{k=0}^n p_{jk}^i x_k$$

$$i = 1, 2, \cdots, r \quad j = 1, 2, \cdots, m$$

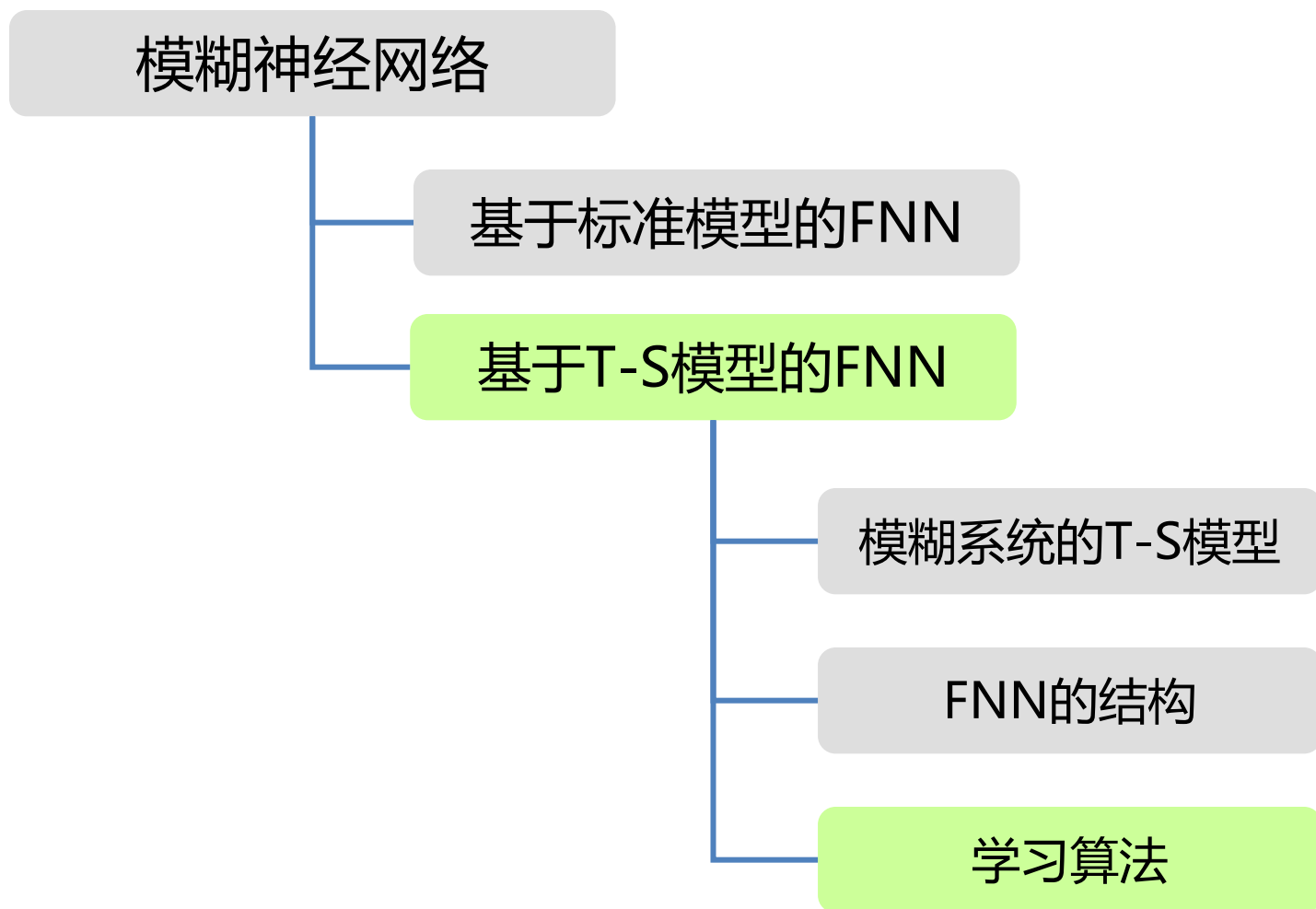
➤ 第三层

计算系统的输出，即：

$$y_i = \sum_{j=1}^m \bar{\alpha}_j y_{ij} \quad i = 1, 2, \cdots, r$$

可见， y_i 是各规则后件的加权和，加权系数为各模糊规则经归一化的适用度，即前件网络的输出用作后件网络第三层的连接权值。

基于T-S型的FNN —— 学习算法



➤ 学习算法

取误差代价函数为

$$E = \frac{1}{2} \sum_{i=1}^r (y_{di} - y_i)^2$$

其中, y_{di} 和 y_i 分别表示期望输出和实际输出。

设各输入分量的模糊分割数是预先确定的, 那么需要学习的参数主要是后件网络的连接权 p_{ji}^k 以及前件网络第二层各结点隶属函数的中心值 c_{ij} 及宽度 σ_{ij} 。

基于T-S型的FNN——学习算法 (3)

首先给出求 p_{ji}^k 的学习算法：

$$\frac{\partial E}{\partial p_{ji}^k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial y_{kj}} \frac{\partial y_{kj}}{\partial p_{ji}^k} = -(y_{dk} - y_k) \bar{\alpha}_j x_i$$

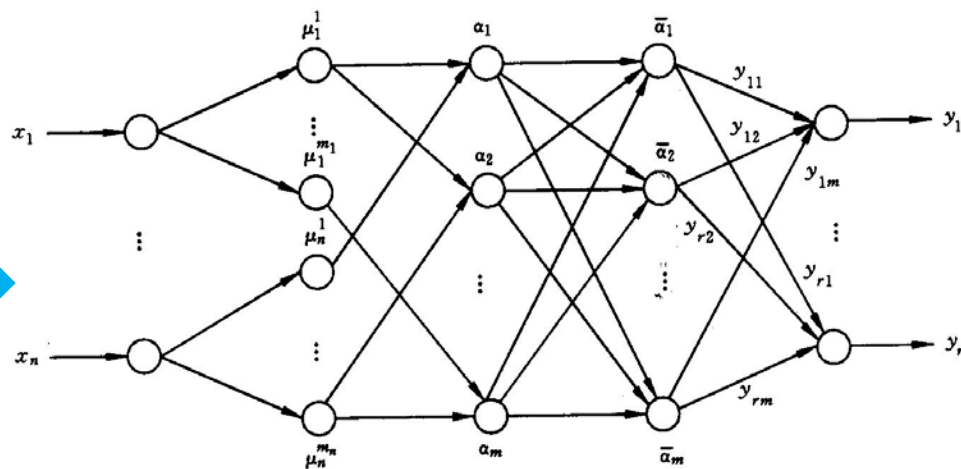
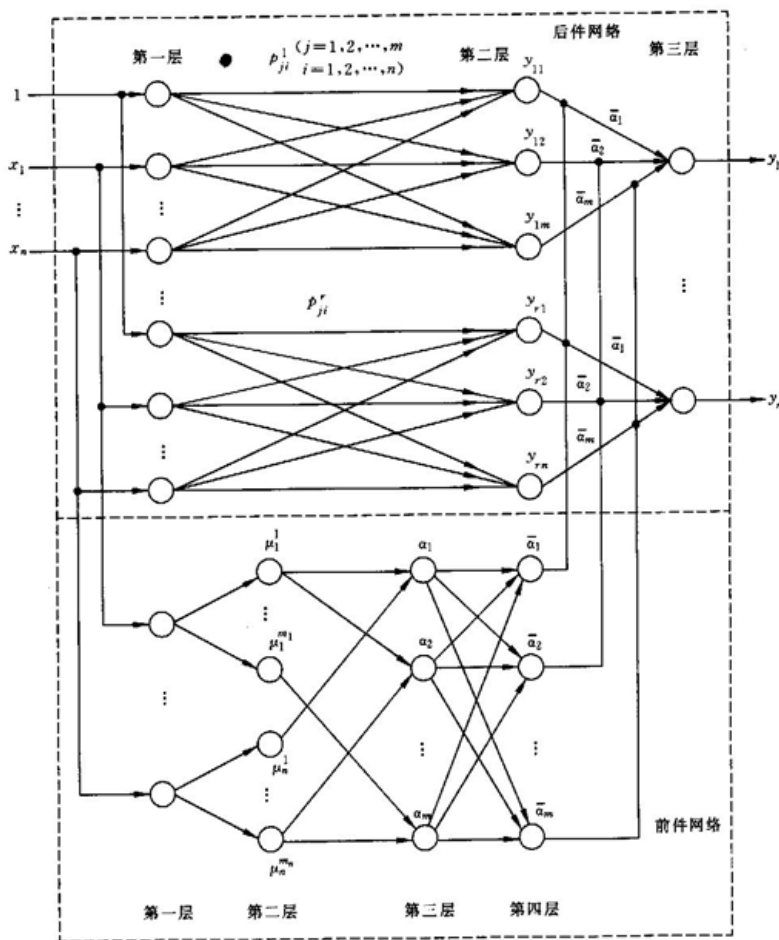
$$p_{ji}^k(l+1) = p_{ji}^k(l) - \beta \frac{\partial E}{\partial p_{ji}^k} = p_{ji}^k(l) + \beta (y_{dk} - y_k) \bar{\alpha}_j x_i$$

其中, $j = 1, 2, \dots, m$; $i = 0, 1, \dots, n$; $k = 1, 2, \dots, r$

基于T-S型的FNN——学习算法 (4)

- 讨论 c_{ij} 及 σ_{ij} 参数的学习，可将参数 p_{ji}^k 设为定值

左图得以简化为右图：



基于T-S模型的FNN简化结构

基于T-S型的FNN——学习算法 (5)

- 简化结构与基于标准模型的FNN具有完全相同的结构
- 只需令最后一层的连接权 $y_{ij} = w_{ij}$ ，则完全可借用前述结果，即有：

$$\delta_i^{(5)} = y_{di} - y_i \quad i = 1, 2, \dots, n$$

$$\delta_j^{(4)} = \sum_{i=1}^r \delta_i^{(5)} y_{ij} \quad j = 1, 2, \dots, m$$

$$\delta_j^{(3)} = \frac{1}{\left(\sum_{k=1}^m \alpha_k\right)^2} \left(\delta_j^{(4)} \sum_{\substack{i=1 \\ i \neq j}}^m \alpha_i - \sum_{\substack{k=1 \\ k \neq j}}^m \delta_k^{(4)} \alpha_k \right)$$

$$\delta_{ij}^{(2)} = \sum_{k=1}^m \delta_k^{(3)} S_{ij} e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m_i$$

基于T-S型的FNN——学习算法 (6)

其中, 当 $f^{(3)}$ 采用取小运算, 则 μ_i^j 是第 k 个规则结点输入的最小值:

$$S_{ij} = 1$$

否则, $S_{ij} = 0$.

当 $f^{(3)}$ 采用相乘运算时, 则 μ_i^j 是第 k 个规则结点的一个输入:

$$S_{ij} = \prod_{\substack{j=1 \\ j \neq i}}^n \mu_j^{i_j}$$

否则, $S_{ij} = 0$.

基于T-S型的FNN——学习算法 (7)

最终可得：

$$\frac{\partial E}{\partial c_{ij}} = -\delta_{ij}^{(2)} \frac{2(x_i - c_{ij})}{\sigma_{ij}^2}$$

$$\frac{\partial E}{\partial \sigma_{ij}} = -\delta_{ij}^{(2)} \frac{2(x_i - c_{ij})^2}{\sigma_{ij}^2}$$

$$c_{ij}(k+1) = c_{ij}(k) - \beta \frac{\partial E}{\partial c_{ij}} \quad i=1, 2, \dots, n \quad j=1, 2, \dots, m_i$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \beta \frac{\partial E}{\partial \sigma_{ij}} \quad i=1, 2, \dots, n \quad j=1, 2, \dots, m_i$$

举例

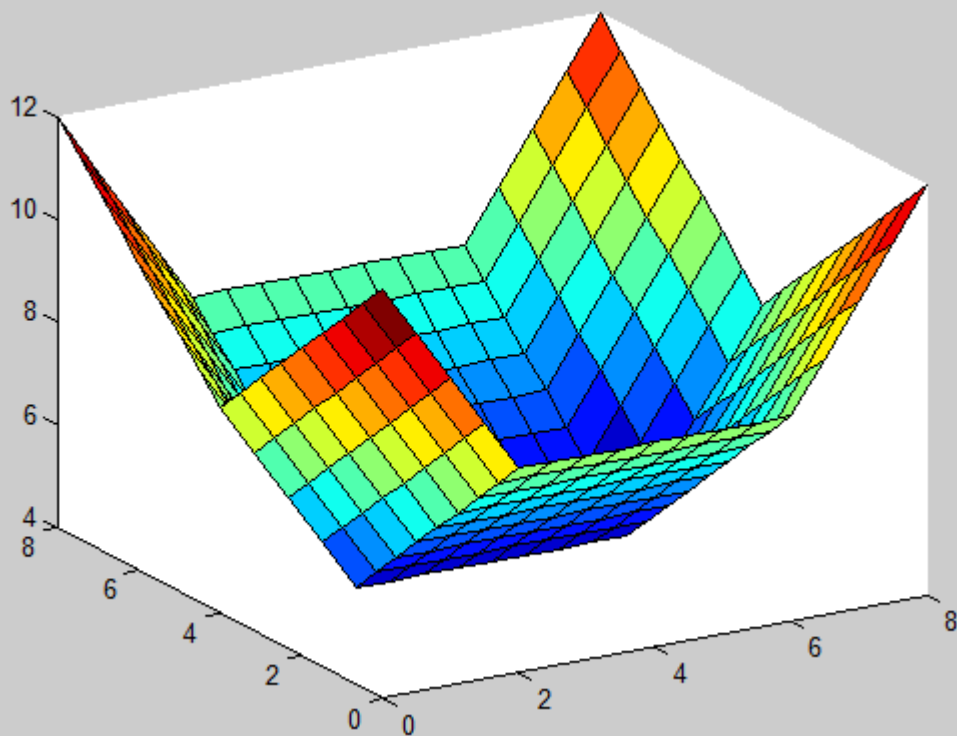
举例

举例

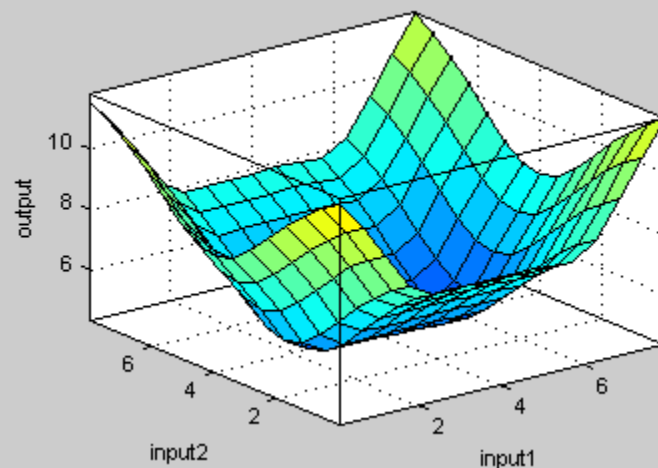
例1.

原系统为 $Z=|X-2|+|X-6|+|Y-4|$

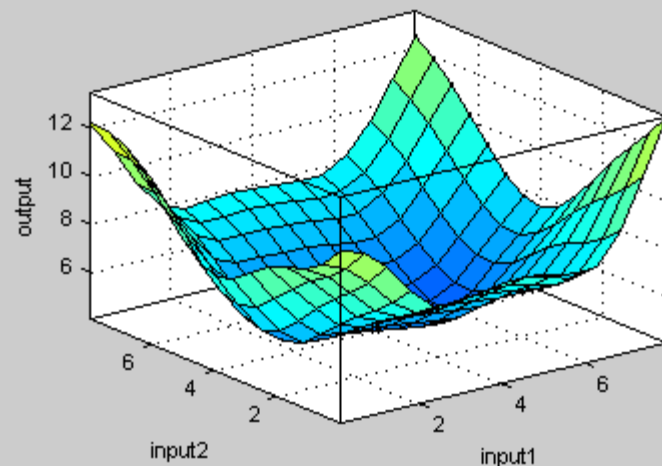
选择400组数据进行系统辨识



无噪声辨识结果



增加标准差为1的高斯噪声后的辨识结果



举例

例2.

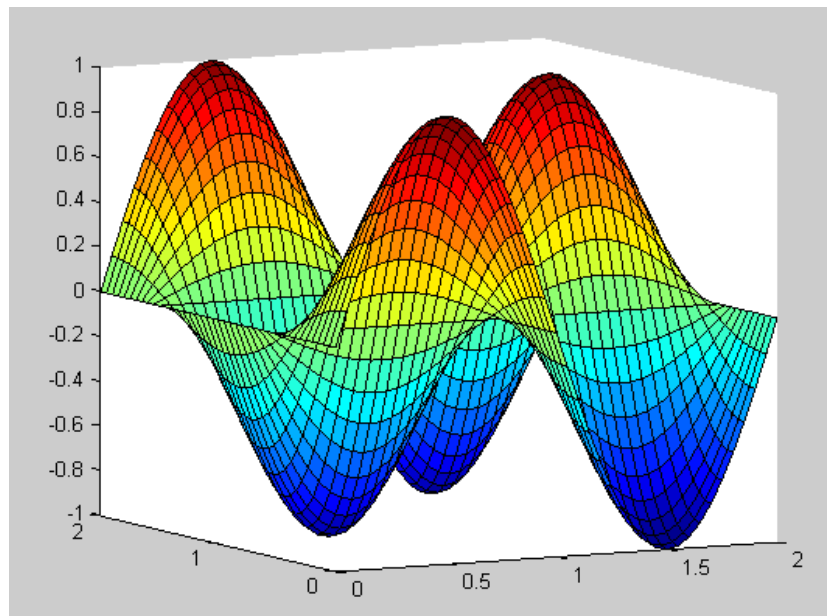
原系统

$$f(x_1, x_2) = \sin(\pi x_1) \cos(\pi x_2)$$

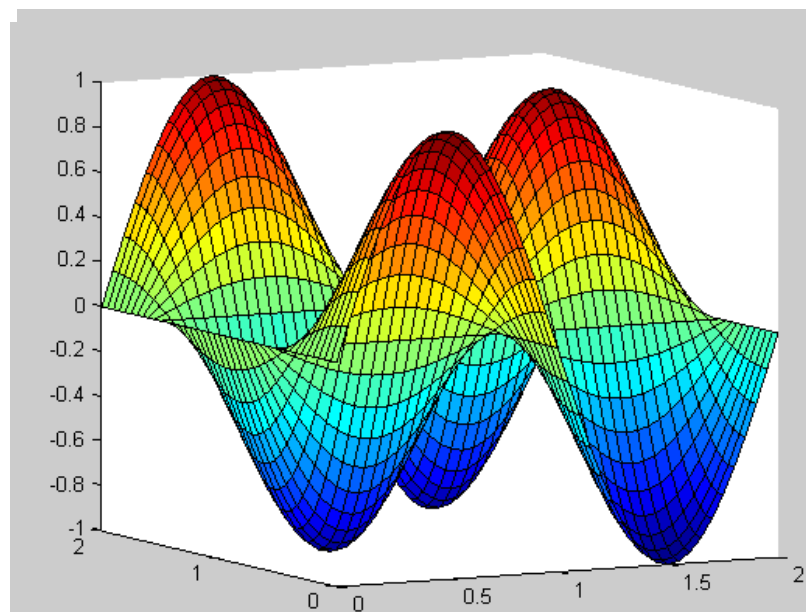
$$x_1 \in [0, 2], x_2 \in [0, 2]$$

选择**420**组数据进行系统辨识。

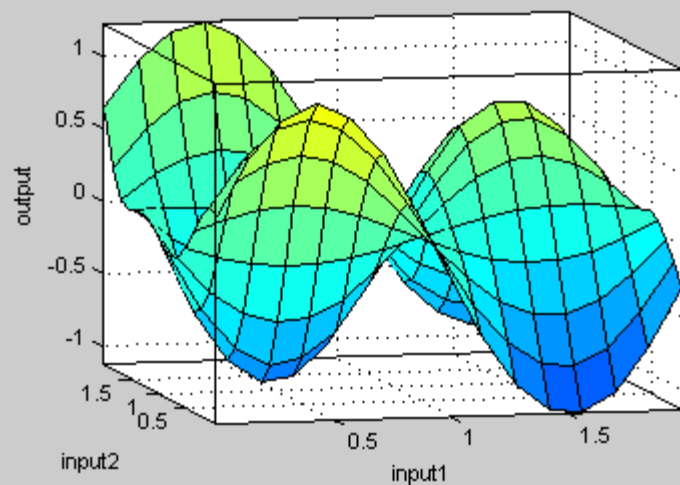
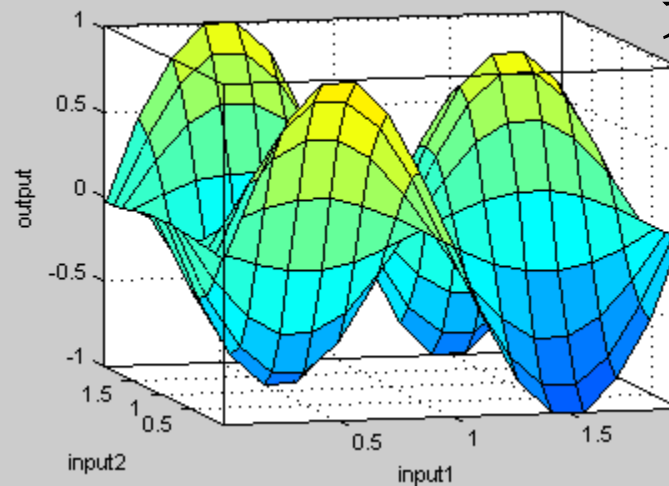
原系统结构：



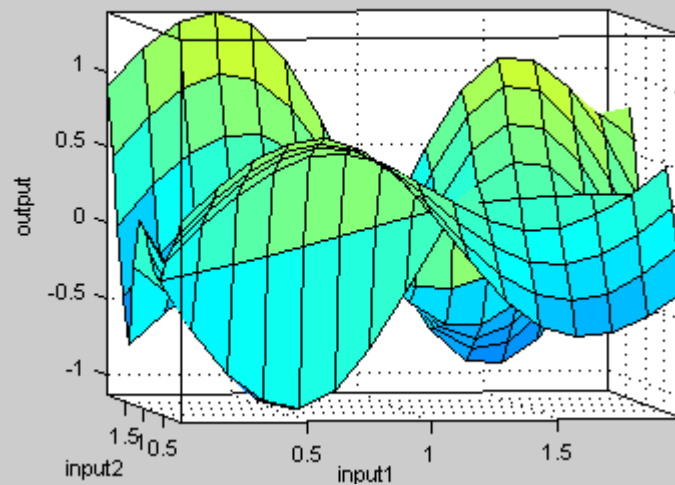
举例



无噪声



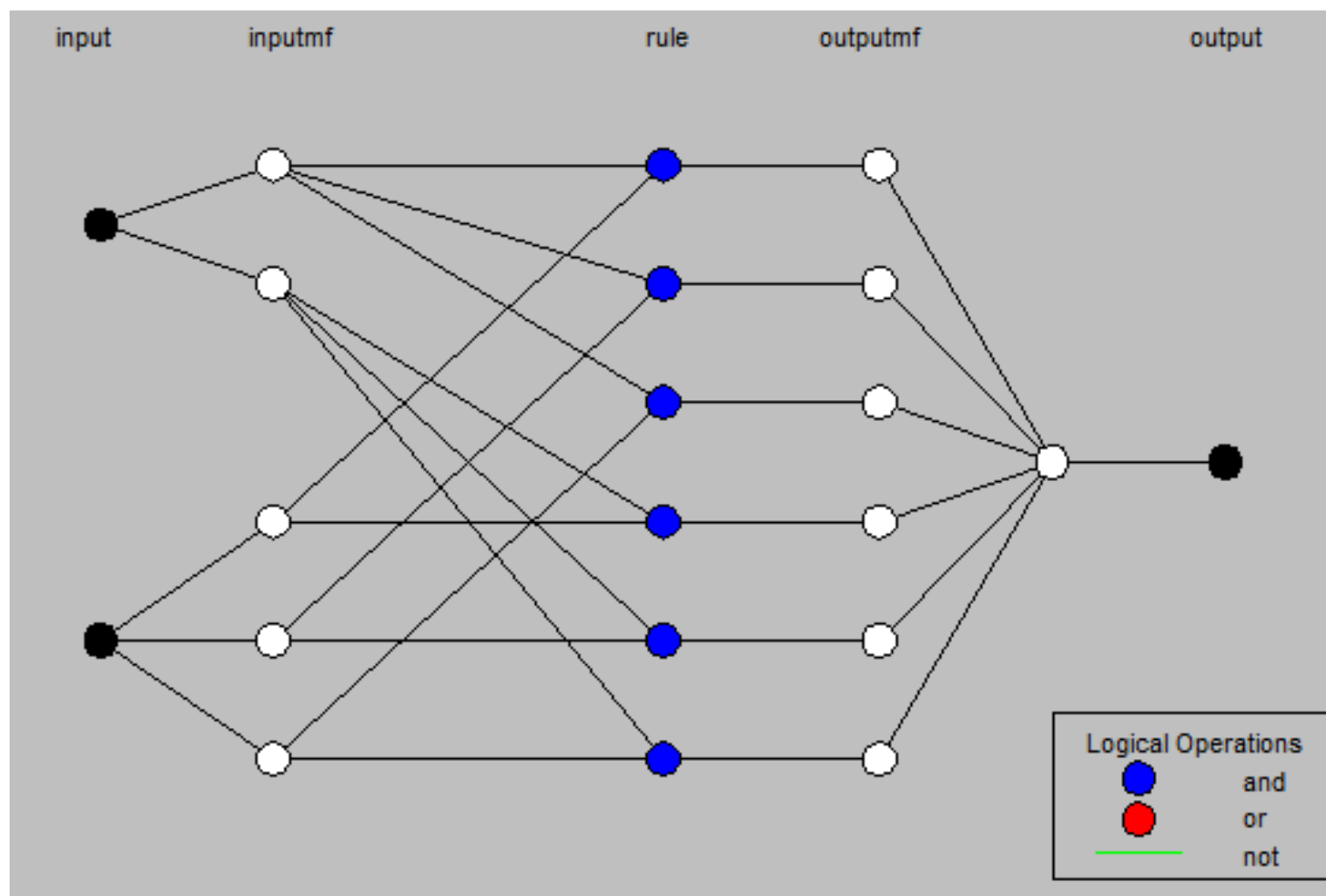
标准差为0.5的高斯噪声



标准差为1的高斯噪声

举例

FNN结构



模糊系统结语

- 模糊智能是应非线性系统处理的需求而出现并发展的；
- 模糊逻辑对推理的依赖正在逐渐减弱，在模糊系统中，算法的设计正在成为主要部分。



作业讨论

选做作业展示

- 各选做内容
- 每位3min , 提问3min
- 重点介绍：结构特点
主要思路
关键步骤
结果讨论

欢迎提问和指正
谢谢!