

# 智能系统设计——Part 1

---

## 总体概述

---

### 专家系统(Expert system)

专家系统是一种旨在模仿特定领域内人类专家推理和决策过程的软件系统。它通常用于解决某些领域的复杂问题。

### 人工神经网络 (Artificial Neural Networks, ANN)

人工神经网络是一种受人脑神经元结构和功能启发的计算模型，能够通过学习从数据中发现模式并解决问题。ANN模拟人类神经元的连接，通过层级结构处理数据（输入层、隐藏层、输出层）。

- 特点：分为有监督学习（需要人工提供标注数据）和无监督学习（自动学习数据特征）

### 模糊系统 (Fuzzy Systems)

模糊系统是一种基于模糊逻辑的计算模型，能够处理“不精确”、“部分真”的信息，以模糊语言（如“高”、“中”、“低”）表达问题。

### 遗传算法 (Genetic Algorithm, GA)

遗传算法是一种基于自然进化过程的优化算法，通过模拟“适者生存”和“自然选择”来寻找问题的最优解。

### 数据挖掘 (Data Mining)

数据挖掘是一种从大规模数据中提取隐藏模式和知识的技术，目的是将原始数据转化为有意义的信息。

## 机器学习

---

### 机器学习的定义

Tom Mitchell (1998) 定义：机器学习是研究能够通过经验 (E) 提升某任务 (T) 性能 (P) 的算法。学习任务可表示为：<P, T, E>。

### 学习的类型

- **监督学习 (Supervised Learning)**：监督学习是一种最常见的机器学习方法，通过**已有的输入数据和对应的标签**来训练模型，目的是让模型学会根据输入预测输出。
  - 常见算法：决策树，线性回归，Logistic回归，支持向量机，
- **无监督学习 (Unsupervised Learning)**：无监督学习的目标是从**没有标签的训练数据**中发现隐藏的模式、结构或关系。其中训练数据仅包含输入数据，没有输出标签
  - 常见算法：聚类算法、降维算法 (PCA)
- **半监督学习 (Semi-supervised learning)**：部分数据有标签，大量数据没有标签。通过利用无标签数据，提升模型性能。
- **强化学习 (Reinforcement Learning, RL)**：强化学习是一种基于**奖励和惩罚机制**的学习方法，模型通过与环境交互来学习如何采取最优行动，以最大化累积奖励。其训练数据没有传统的输入输出数据，而是通过环境的交互获取状态、动作和奖励。

# 机器学习算法包含三个部分

每个机器学习算法都包含三大组成部分：**表示 (Representation)**、**优化 (Optimization)**、**评估 (Evaluation)**

## 表示

表示是关于算法如何表示问题以及数据的方式，决定了模型的能力范围和潜在的解决方案空间。其作用是定义算法如何表示数据和模型，以及决定模型可以学习的模式的复杂程度。

例如：决策树使用分支结构表示决策规则，神经网络使用多层结构表示复杂的模式

- 内容：包括模型的形式和结构
  - **线性模型**：如线性回归、逻辑回归。
  - **树模型**：如决策树、随机森林。
  - **神经网络**：如深度学习模型。
  - **概率模型**：如贝叶斯网络、隐马尔可夫模型 (HMM)。

## 优化

优化是关于如何调整模型参数以最大化或最小化某个目标函数（通常是损失函数）。其作用是找到最佳模型参数，使模型在训练数据上表现最佳。优化包括算法收敛速度、效率和稳定性。优化目标为最小化损失函数或最大化收益函数。

- **优化方法**：
  - **梯度下降法**：如随机梯度下降 (SGD)、动量梯度下降、Adam优化器。
  - **动态规划**：用于序列建模（如HMM学习）。
  - **遗传算法**：如进化计算、神经网络的结构优化。

## 评估

评估是关于如何测量模型的性能，衡量模型是否达到了任务需求。评估衡量模型在训练集和测试集上的表现，帮助选择最佳模型以及检测过拟合或欠拟合问题。

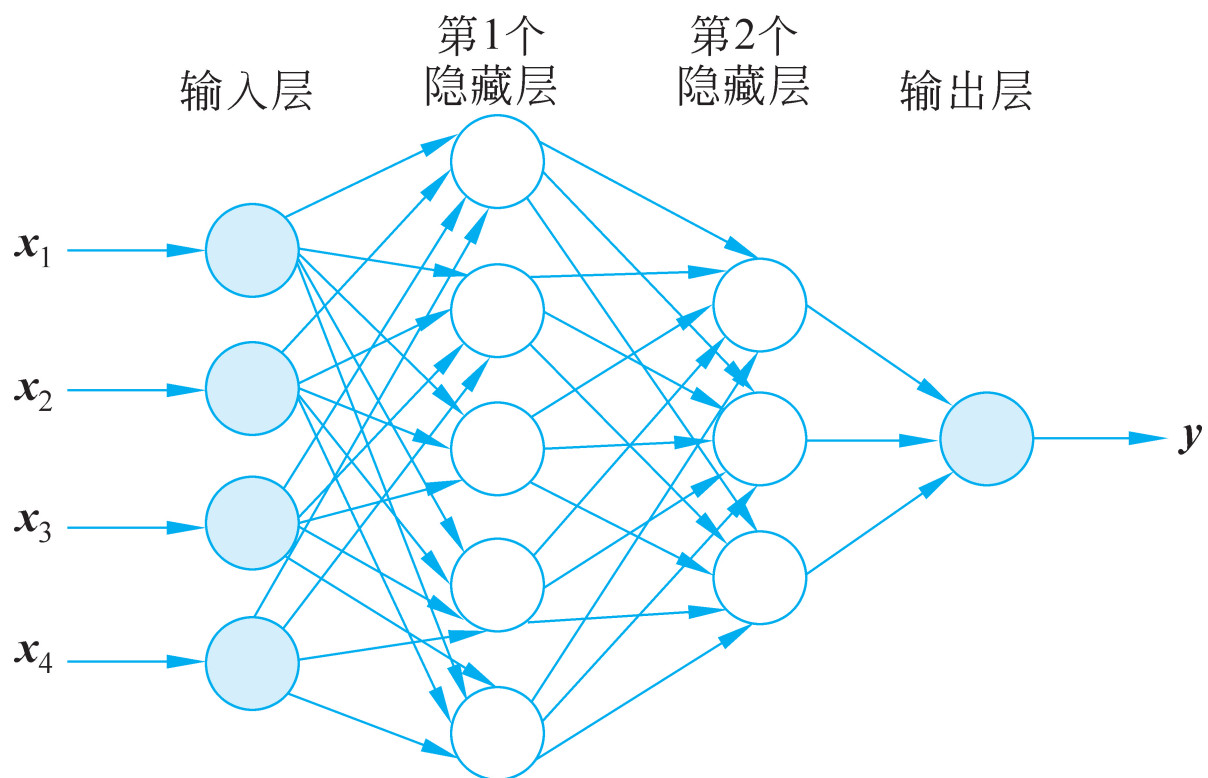
- 评估指标：
  - 准确率 (Accuracy)、精确率 (Precision)、召回率 (Recall)、F1分数
  - 均方误差 (MSE)、绝对误差 (MAE)。
  - 对数似然 (Log Likelihood)，熵 (Entropy)、KL散度

# 人工神经网络 (ANN)

---

## 神经网络定义

人工神经网络是一种受人类大脑启发的系统，可以是算法，也可以是硬件，其设计模仿了人脑的结构和功能。通过调整连接权重，神经网络可以通过示例进行学习，并具有一定的泛化能力。



## 人工神经网络基本结构

人工神经网络由感知器 (Perceptron) 组成，分为**输入层**、**隐藏层**和**输出层**。其中隐藏层的数量可以多于一个，多层隐藏层的网络称为“深度神经网络”。输入层接收外部数据，通过权重调整激活目标输出。

- 关键组成部分：
  - **输入节点**：接收外部输入并传递到隐藏层
  - **权重**：连接的效能，权重值可正（兴奋）或负（抑制），通过学习过程进行调整以实现特定功能。
  - **激活函数**：常用非线性函数如Sigmoid和分段线性函数，用于输出层和隐藏层计算

## 前馈神经网络 (Feed-forward Neural Networks)

最常见的神经网络类型

- **单层前馈网络 (感知机)**：包含输入层和输出层。尽管有两个层，但由于只有输出层参与计算，因此称为单层。
- **多层前馈网络**：包含输入层、一个或多个隐藏层，以及输出层。隐藏层负责进行中间计算并传递信息到输出层。

## 递归神经网络 (Recurrent Neural Networks, RNN)

- 特点
  - 与前馈网络的区别：递归神经网络与前馈神经网络不同，RNN 中至少存在一个反馈回路。这意味着输出可以影响输入（或中间状态），从而具有记忆历史输入的能力。
  - 不一定稳定：由于存在反馈连接，RNN 可能不稳定
  - 可以学习时间序列：递归结构使得 RNN 能够处理和学习时间相关的信息，例如时间序列数据、振荡模式或动态信号。这是 RNN 的核心优势。
  - 模拟生物神经网络的特性：在生物神经网络（如大脑皮层）中，大多数连接是**反馈连接**（反馈来自于本地或远程神经元）。这种反馈机制被认为与记忆和动态处理密切相关。因此，RNN 被认为在某种程度上更贴近生物神经元的实际工作机制。
- RNN优缺点

- 优点：RNN 的内部状态可以记忆过去的信息，适用于时间序列数据（如语音、文本、视频等）的处理。此外，通过反馈回路，可以很好地模拟动态信号。
- 缺点：由于长时间依赖，可能面临梯度消失或梯度爆炸问题，训练困难。需要展开时间维度，增加计算复杂度。

## 反向传播算法（Back-propagation, BP）

### BP神经网络

### 神经网络的衡量方法

- 均方误差（Mean-Squared-Error, MSE）：用于衡量神经网络的预测值和目标值之间的差异

$$MSE = \frac{1}{n} \sum (\text{desired}^2 - \text{actual}^2)$$

- 均方根误差（Root-Mean-Squared-Error, RMSE）：均方误差的平方根形式，用于将误差的量纲与原值一致化

$$RMSE = \sqrt{\frac{\sum (\text{desired} - \text{actual})^2}{n}}$$

### BP神经网络的定义

BP神经网络是一种多层的前馈神经网络，其主要特点是：**信号是前向传播的，而误差是反向传播的**。BP神经网络的过程主要分为两个阶段，第一阶段是信号的前向传播，从输入层经过隐藏层，最后到达输出层；第二阶段是误差的反向传播，从输出层到隐藏层，最后到输入层，依次调节隐藏层到输出层的权重和偏置，输入层到隐藏层的权重和偏置。

### 计算步骤

- 计算每个神经元的输出值

$$net_j = \sum_k W_{jk} \cdot O_k + \text{bias}_j$$

- 输出层误差信号的计算

$$\delta_{pj} = (T_{pj} - O_{pj}) \cdot O_{pj} \cdot (1 - O_{pj})$$

- 其中：
  - $\delta_{pj}$ ：输出层神经元  $j$  的误差信号。
  - $T_{pj}$ ：目标输出值（期望值）。
  - $O_{pj}$ ：实际输出值。
  - $O_{pj} \cdot (1 - O_{pj})$ ：激活函数的导数（假设为 Sigmoid 激活函数）。

- 隐藏层误差信号的计算

$$\delta_{pj} = O_{pj} \cdot (1 - O_{pj}) \cdot \sum_k \delta_{pk} \cdot W_{kj}$$

其中

- $\delta_{pj}$ ：隐藏层神经元  $j$  的误差信号。
- $\sum_k \delta_{pk} \cdot W_{kj}$ ：隐藏层神经元  $j$  所连接的输出层误差信号的加权和。

- 权重调整公式

$$\Delta W_{ji}(t) = \eta \cdot \delta_{pj} \cdot O_{pi}$$

其中 $\eta$ 表示学习率

## 自组织映射 (Self-Organizing Map, SOM)

**自组织映射 (SOM)** 是一种**无监督学习**的神经网络。SOM 的主要作用是将**高维数据映射到低维（通常是二维）空间**，同时保持数据的**拓扑结构**，即映射后相邻的神经元表示高维空间中相似的数据。

- 特点：
  - SOM是无监督学习类型，不需要目标输出或标签数据。可以自动识别输入数据的模式，进行聚类或分类。
  - 竞争学习机制：输入数据被与网络中的每个神经元比较，找出距离最近的神经元（即“胜者”）。只有胜者及其周围的邻居神经元会调整权重

### SOM的工作原理

- 初始化：设置神经元的初始权重（通常为随机值）
- 竞争阶段：对于每个输入数据，计算其与每个神经元权重的相似性（通常用欧几里得距离衡量）。找到最接近的神经元（称为“胜者神经元”）
- 更新阶段：仅调整胜者神经元及其周围邻居神经元的权重，使它们更接近当前输入数据。权重更新公式为：

$$W_i(t+1) = W_i(t) + \alpha(t) \cdot [X(t) - W_i(t)]$$

其中 $W_i$ 是神经元权重， $\alpha$ 是学习率， $X$ 是输入数据

## 模糊逻辑 (Fuzzy logic)

### 定义

模糊逻辑是用来描述模糊性的逻辑体系，与传统布尔逻辑的“真”或“假”不同，它支持“程度”的概念，比如温度高低、美观程度等滑动标尺上的值。模糊集是模糊逻辑的核心概念，它允许元素部分属于一个集合，成员资格用一个从0到1的值表示，而非布尔逻辑中的0或1

### 模糊集合

模糊集合通过表示对集合的归属程度 $[0, 1]$ 这样的隶属(membership)值，尝试表示没有明确界限的模糊概念。这与通常集合的属于还是不属于的Yes/No判断基准的定义相对，由于引入归属的程度，模糊集合相当于允许不明确性。模糊集合可以更好地处理现实中不确定性和模糊性问题，例如“高个子”或“适度温度”。

### 模糊集合的表示方式

模糊集合A通常表示为：

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n}$$

其中

$x_i$ 是模糊集合中的具体元素，属于某个有限的全集 (finite universe of discourse)

$\mu(x)$ 是 $x_i$ 的**隶属度 (grade of membership)**，表示 $x_i$ 属于A的程度

模糊集合的全集可以定义为：

$$A = \{x_1, x_2, x_3, \dots, x_n\}$$

- 全集的作用：它定义了模糊集合可以取值的范围。模糊集合A中的这些元素全部来自于全集

## 模糊集合的运算

### 1. 补集 (Complement)

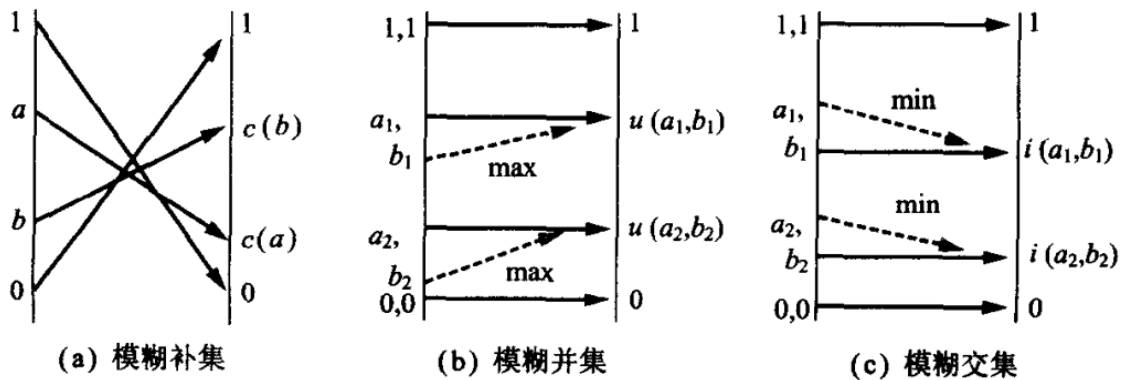
$$\mu_c(x) = 1 - \mu_A(x)$$

### 2. 并集 (Union)

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$$

### 3. 交集 (Intersection)

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$



## 模糊集合的性质

- 等价性(Equality):** 两个模糊集合相等当且仅当其隶属度相同  $\mu_A(x) = \mu_B(x)$
- 包含性(Inclusion):** 模糊集合 A 包含于模糊集合 B (即  $A \subseteq B$ ) , 条件是:  $\mu_A(x) \leq \mu_B(x)$
- 基数 (Cardinality):** 集合中所有元素隶属度的总和。  $card_A = \sum_{i=1}^n \mu_A(x_i)$
- 空集(Empty Fuzzy Set):** 如果集合中所有元素的隶属度均为0, 则为空集
- $\alpha$ -截集 (Alpha-Cut):** 提取隶属度大于某阈值 $\alpha$ 的元素子集。  $A_\alpha = \{x \in X | \mu_A(x) \geq \alpha\}$
- 正则性 (Normality):**
  - 正则集合: 如果一个模糊集合 A 中至少存在一个元素  $x \in X$ , 使得隶属度  $\mu_A(x) = 1$ , 则称该集合为 **正则集合 (Normal Fuzzy Set)**。
  - 次正则集合 (Subnormal Fuzzy Set): 这种集合的所有元素的隶属度都小于 1
  - 高度 (Height): 集合中所有元素的最大隶属度值

$$\text{height}(A) = \max_{x \in X} (\mu_A(x))$$

- 支持度 (Support):** 模糊集合 A 的 **支持度 (Support)** 是一个普通集合, 由所有隶属度大于 0 的元素组成:

$$\text{supp}(A) = \{x \in X | \mu_A(x) > 0\}$$

- 核心 (Core):** 模糊集合 A 的 **核心 (Core)** 是一个普通集合, 由所有隶属度等于 1 的元素组成:

$$core(A) = \{x \in X | \mu_A(x) = 1\}$$

## 模糊集合的数学运算法则

### 1. 线性缩放 (Scaling)

- 运算定义：

$$aA = \{a \cdot \mu_A(x), \forall x \in X\}$$

- $a$ ：一个标量，用于对集合  $A$  的隶属度值进行缩放。
- $A$ ：原模糊集合，其形式为  $\{\mu_A(x_1)/x_1, \mu_A(x_2)/x_2, \dots\}$ 。
- $aA$ ：缩放后的模糊集合，其中**所有隶属度值都乘以  $a$** 。

### 2. 幂运算 (Exponentiation)

- 运算定义：

$$A^a = \{\mu_A(x)^a, \forall x \in X\}$$

- $a$ ：一个标量，用于将模糊集合  $A$  的每个隶属度值取幂。
- $A$ ：原模糊集合，其形式为  $\{\mu_A(x_1)/x_1, \mu_A(x_2)/x_2, \dots\}$ 。
- $A^a$ ：幂运算后的模糊集合，其中**每个隶属度值都取  $a$  次幂**。

## 模糊规则 (Fuzzy Rules)

### 定义

模糊规则是以 **IF-THEN (如果-那么)** 的形式构建的条件语句，用于表达输入与输出之间的关系。其一般形式为：

IF Condition (Antecedent) THEN Action (Consequent)

- Condition (前件/条件) :**
  - 输入条件，用模糊语言变量表示，例如“温度是高的”，“湿度是适中的”。
  - 描述输入的模糊集合。
- Action (后件/结论) :**
  - 输出结果，也用模糊语言变量表示，例如“风速是快的”，“喷洒量是少的”。
  - 描述输出的模糊集合。

例如：

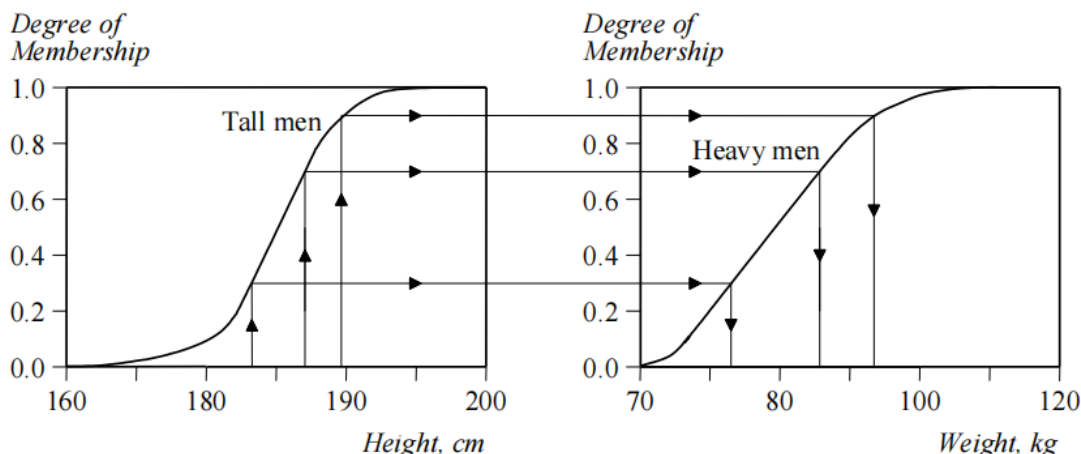
- IF temperature is high THEN fan speed is fast
  - 这是一个模糊规则，其中“temperature is high”是前件，“fan speed is fast”是后件。

### 模糊规则的触发和推理(Firing Fuzzy Rules)

在经典规则中，条件（前件，Antecedent）要么为真（完全触发规则），要么为假（规则不触发）。在模糊规则中，所有规则可以 **部分触发 (Partially Fire)**，即前件的隶属度不是 0 或 1，而是一个介于 0 和 1 之间的值，代表触发程度。**如果前件的隶属度值为某个程度，则后件的隶属度值也会按照相同程度被触发。**

- 触发过程：
  - 前件 (Antecedent) 的计算：**前件的隶属度值通过输入变量的**隶属函数(membership function)**计算得出。例如，“height is tall”的隶属度是根据输入身高值在模糊集合“tall men”中的隶属度来确定的。

- **后件 (Consequent) 的计算**: 后件的隶属度由前件的隶属度决定, 并使用某种推理机制 (如单调选择方法 Monotonic Selection) 。



如该图所示, 前件的触发可以转移到后件

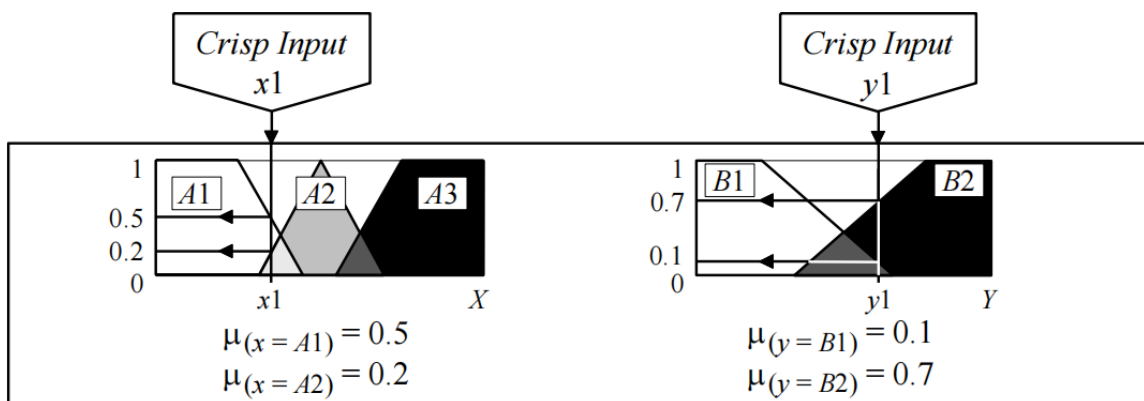
## 模糊推理(Fuzzy inference)

把集合中的模糊性进行扩张, 表示处理逻辑中的模糊性, 就是模糊推理。模糊推理被认为是根据模糊规则进行的近似推理, 是多值逻辑的延伸上的推理。两种主要的模糊推理方法 (**Mamdani 和 Sugeno**)

### Mamdani 模糊推理方法的过程

#### 1. 模糊化 (Fuzzification)

将输入的精确值 (Crisp Input) 转化为模糊集合中对应的隶属度值。根据输入找到对应的隶属度的值。



这里  $x_1$  输入对应  $A_1$  的隶属度值为 0.5, 对应隶属度  $A_2$  的值为 0.2。  $y_1$  同理

#### 2. 规则评估 (Rule Evaluation)

基于模糊规则的前提 (Antecedents), 使用模糊逻辑运算 (如 AND/OR) 计算每条规则的结果 (Truth Value) 。

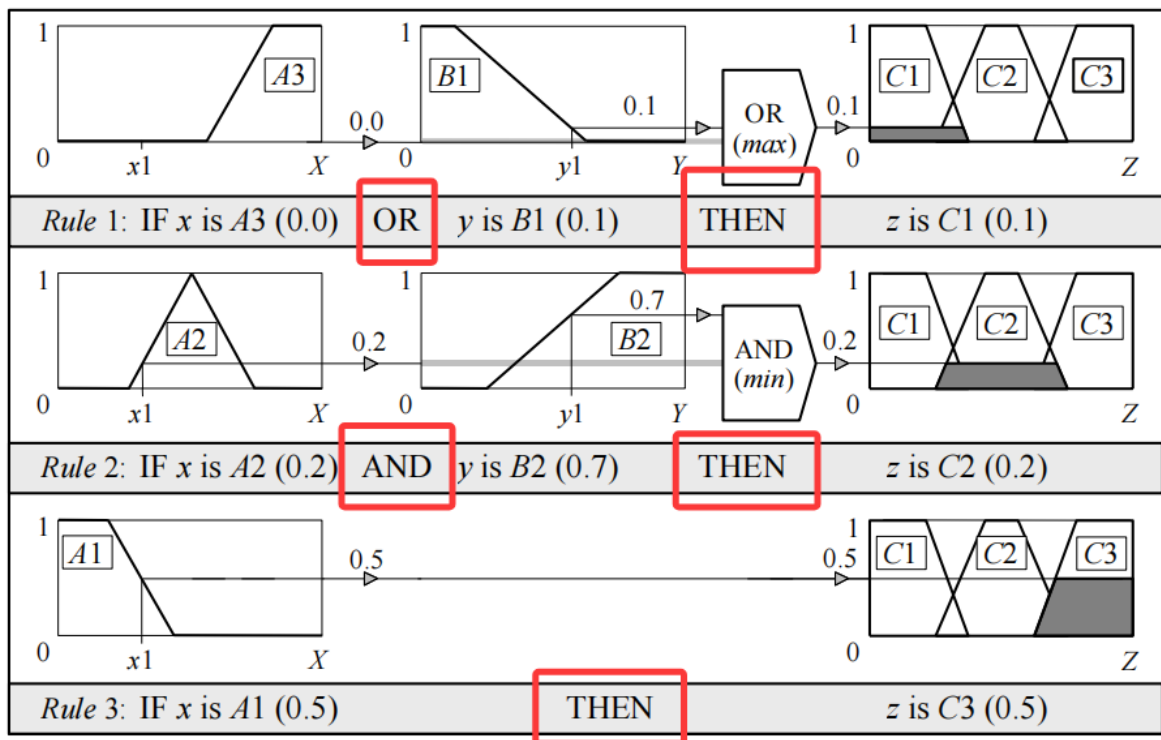
- 通过模糊逻辑运算计算前提条件的结果:
  - 若规则使用 **AND** 运算 (交集), 则取多个隶属度的最小值:

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

- 若规则使用 **OR** 运算 (并集), 则取多个隶属度的最大值:

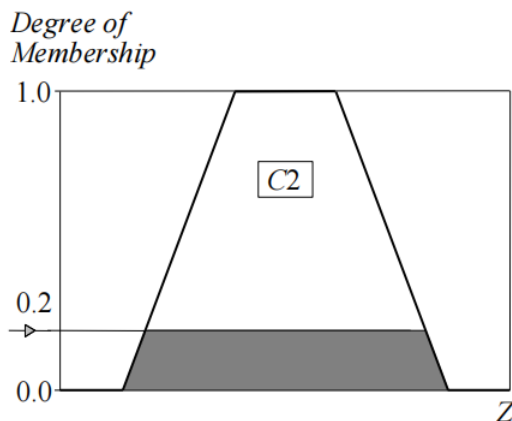
$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$



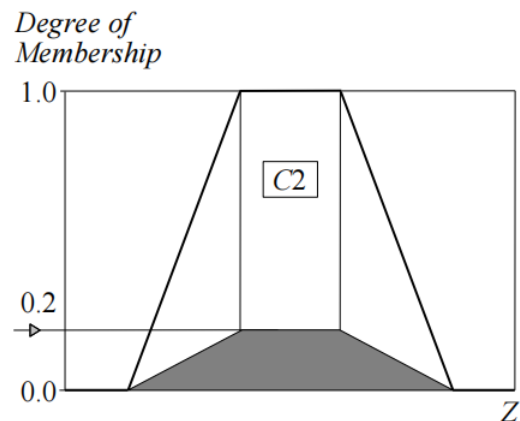


例如该图中，利用模糊规则指定的逻辑表达符号，分别计算最后的 $C_1$ ,  $C_2$ ,  $C_3$

- **对规则后件处理：Clipping (截断) 和 Scaling (缩放)** 两种方法。截断法通过前提的隶属度值对后件模糊集合进行截断，缩放法使用前提的隶属度值对后件模糊集合的所有隶属度进行缩放
  - **Clipping 方法：**将规则前提的隶属度 (Truth Value) 作为一个阈值，直接截断后件模糊集合的顶部。截断后，模糊集合顶部被削去，因此会丢失一些信息，但由于计算简单且效率高，它是最常用的方法之一。
  - **Scaling 方法：**使用规则前提的隶属度值，对后件模糊集合的每个隶属度值进行比例缩放 (乘以该隶属度值)。原始模糊集合的形状得以保留，只是高度被调整。缩放后的模糊集合保留了更多细节，适合高精度的模糊专家系统 (Fuzzy Expert Systems)。



clipping



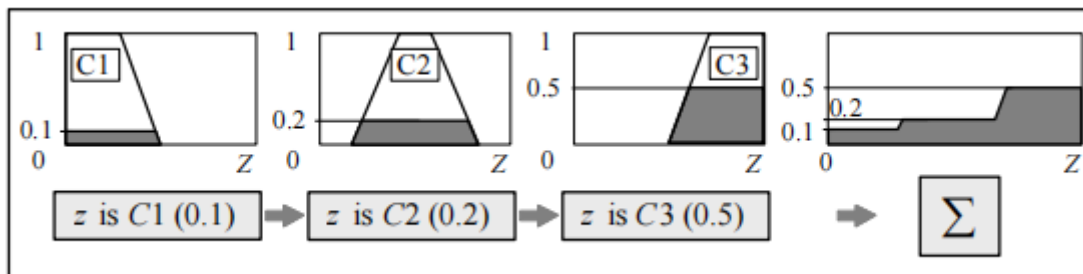
scaling

### 3. 规则输出的聚合 (Aggregation of Rule Outputs)

将所有规则的输出模糊集合整合为一个模糊集合。其中每条规则输出的模糊集合根据规则的前提结果进行剪裁 (Clipping) 或缩放 (Scaling)。

- **聚合方法：**聚合将所有规则的模糊输出合并为一个模糊集合。通常通过取规则输出模糊集合的最大值完成

$$\mu_{\text{aggregated}}(z) = \max(\mu_{\text{reul1}}(z), \mu_{\text{reul2}}(z), \dots)$$

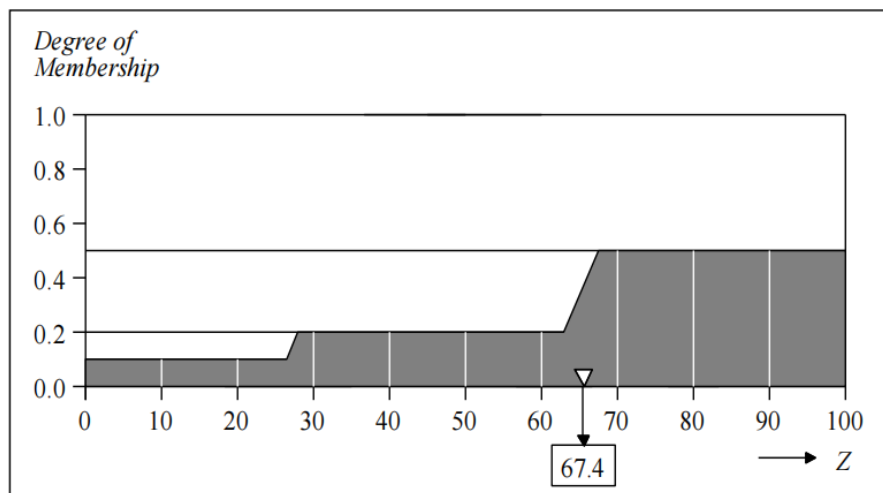


#### 4. 去模糊化 (Defuzzification)

将聚合后的模糊集合转化为清晰的输出值。本课程使用主要方法是 **质心法 (Centroid Method)**。计算模糊集合的重心 (质心)，公式如下：

$$z = \frac{\int_a^b \mu(z) \cdot z dz}{\int_a^b \mu(z) dz}$$

其中： $\mu(z)$ 表示模糊集合的隶属函数； $z$ 表示清晰值。在真实情况中，使用离散的值进行求解。



$$COG = \frac{(0+10+20) \times 0.1 + (30+40+50+60) \times 0.2 + (70+80+90+100) \times 0.5}{0.1+0.1+0.1+0.2+0.2+0.2+0.2+0.5+0.5+0.5+0.5} = 67.4$$

#### Sugeno方法

- Sugeno 模糊推理方法主要的不同点在于**规则的后件 (Consequent)**：
  - Mamdani 方法中，后件是一个**模糊集合**。
  - Sugeno 方法中，后件是一个**数学函数或常数**。
- 两种方法的对比：
  - **Mamdani方法**：适合捕捉专家知识，更加直观且类似于人类思维，但计算复杂度高。
  - **Sugeno方法**：计算效率高，适合控制问题尤其是动态非线性系统。

#### 自适应神经模糊推理系统 (ANFIS, Adaptive Neuro-Fuzzy Inference System)

ANFIS (自适应神经模糊推理系统) 可以看作是一个结合了**神经网络**和**模糊逻辑**的模型，用于解决复杂的预测或分类问题。

## 模糊关系(Fuzzy relationship)

模糊关系是经典关系的推广，允许部分隶属度的存在。例如，模糊关系“朋友”可以描述两个人之间的友谊程度，而不是简单的“是朋友”或“不是朋友”。简单来说，模糊关系就像给人与人之间的关系加了“分数”。在传统的关系里，比如“朋友关系”，要么是朋友（1），要么不是朋友（0）。但在模糊关系里，友谊可以是部分的。

**模糊笛卡尔积**是生成模糊关系的一种方法，**模糊组合**是模糊关系的一种运算方法，用于将多个模糊关系组合起来。

### 模糊笛卡尔积(Cartesian Product)

假设  $\tilde{A}$  是在宇宙  $X$  上的一个模糊集合； $\tilde{B}$  是在宇宙  $Y$  上的一个模糊集合

模糊笛卡尔积 ( $\tilde{A} \times \tilde{B}$ ) **定义**为一个模糊关系  $\tilde{R}$ ，它是笛卡尔空间  $X \times Y$  的子集。该模糊关系通过隶属函数定义，公式如下：

$$\mu_R(x, y) = \min(\mu_A(x), \mu_B(y))$$

- $\mu_A(x)$ :  $x$  在集合  $\tilde{A}$  中的隶属度。
- $\mu_B(y)$ :  $y$  在集合  $\tilde{B}$  中的隶属度。

### 模糊组合 (Fuzzy Composition)

假设：

- $\tilde{R}$  是笛卡尔空间  $X \times Y$  上的一个模糊关系。
- $\tilde{S}$  是笛卡尔空间  $Y \times Z$  上的一个模糊关系。
- $\tilde{T}$  是笛卡尔空间  $X \times Z$  上的模糊关系。

通过模糊组合可以将  $X \rightarrow Y$  和  $Y \rightarrow Z$  的模糊关系组合为  $X \rightarrow Z$  的模糊关系。

- **最大-最小法 (Max-Min Composition) :**

$$\mu_T(x, z) = \bigvee_{y \in Y} (\mu_R(x, y) \wedge \mu_S(y, z))$$

- 对于每对  $(x, z)$ ，计算  $y \in Y$  的所有可能组合的最小值（ $\wedge$  取较小值），然后取最大值（ $\vee$  取最大值）。

- **最大-乘法 (Max-Product Composition) :**

$$\mu_T(x, z) = \bigvee_{y \in Y} (\mu_R(x, y) \cdot \mu_S(y, z))$$

- 对于每对  $(x, z)$ ，计算  $y \in Y$  的所有可能组合的乘积（ $\cdot$  表示乘积），然后取最大值（ $\vee$  取最大值）。

## 模糊图(Fuzzy Graph)

模糊图表示一组输入语言变量与一个输出语言变量之间的功能映射。通俗来说，模糊图是用来描述事物之间关系的一种工具，但它和我们平常用的“清晰的”数学函数图不太一样。它主要处理那些“不确定”或者“模糊”的信息。

## 扩展原则(Extension principle)

扩展原则是一种方法，用于将传统明确 (crisp) 领域的数学表达扩展到模糊 (fuzzy) 领域。具体来说：

- 对于一个明确函数  $f(x)$ ，它将一个输入  $x$  映射到一个唯一的输出  $y$ 。
- 在模糊领域，扩展原则允许我们将输入模糊集合  $A$  转换为输出模糊集合  $B$ 。

### 数学表示

通过扩展原则，输入模糊集合中的所有可能值通过函数  $f(x)$  映射后形成一个新的模糊集合。假设有一个函数  $f$  从集合  $X$  映射到集合  $Y$ ，并且  $A$  是集合  $X$  上的一个模糊集合，其定义为：

$$A = \mu_A(x_1)/(x_1) + \mu_A(x_2)/(x_2) + \cdots + \mu_A(x_n)/(x_n)$$

这里：

- $\mu_A(x_i)$  是模糊集合  $A$  中  $x_i$  的隶属度，表示  $x_i$  属于集合  $A$  的程度（值在 0 到 1 之间）。
- $x_i$  是集合  $X$  中的具体值。

扩展原则表明，通过函数  $f(x)$ ，模糊集合  $A$  的“映射”形成新的模糊集合  $B$ ，其表达式为：

$$B = f(A) = \mu_A(x_1)/(y_1) + \mu_A(x_2)/(y_2) + \cdots + \mu_A(x_n)/(y_n)$$

其中：

- $y_i = f(x_i)$ ，即  $y_i$  是  $x_i$  通过函数  $f(x)$  映射后的值。
- $\mu_B(y)$  是集合  $B$  中  $y$  的隶属度。

### 多对一的映射情况

若函数  $f(x)$  是一个多对一的映射（即多个  $x$  值可能映射到同一个  $y$  值），那么输出模糊集合  $B$  的隶属度计算方式是：

$$\mu_B(y) = \max_{x=f^{-1}(y)} \mu_A(x)$$

通俗解释：

- 在模糊集合  $A$  中，输入值可能具有不同程度的隶属度（介于 0 到 1 之间），映射到输出集合  $B$  后，输出值的隶属度由输入值的隶属度决定。
- 如果一个输出值由多个输入值映射得到，那么该输出值的隶属度取输入值隶属度的最大值。

## 模糊运算(Fuzzy Arithmetic)

**Fuzzy Arithmetic (模糊算术)** 是一种用于处理 **不确定性和模糊性** 的数学工具。它扩展了传统的算术运算（比如加、减、乘、除），让它们能够在 **模糊数**（具有不确定性或模糊边界的数）之间进行操作。它的核心在于利用模糊集合和隶属度函数来描述不确定的值，并对这些值进行运算。

### 算数运算

#### 1. 模糊加法 (Fuzzy Addition)

$$\mu_{A+B}(z) = \bigoplus_{x+y=z} [\mu_A(x) \otimes \mu_B(y)]$$

- $\bigoplus_{x+y=z}$ ：在所有满足  $x + y = z$  的  $(x, y)$  对中取隶属度的最大值。
- $\otimes$ ：表示取隶属度的最小值

#### 2. 模糊减法 (Fuzzy Subtraction)

$$\mu_{A-B}(z) = \bigoplus_{x-y=z} [\mu_A(x) \otimes \mu_B(y)]$$

### 3. 模糊乘法 (Fuzzy Multiplication)

$$\mu_{A \times B}(z) = \bigoplus_{x \times y=z} [\mu_A(x) \otimes \mu_B(y)]$$

### 4. 模糊除法 (Fuzzy Division)

$$\mu_{A/B}(z) = \bigoplus_{x/y=z} [\mu_A(x) \otimes \mu_B(y)]$$

## 遗传算法(Genetic Algorithms)

遗传算法是一种**基于种群的搜索方法**，以解决复杂问题的能力而著称。标准的遗传算法并不直接考虑约束条件，这使得其在处理约束优化问题时需要特别的机制（如约束处理技术）。遗传算法属于“生成-测试”类型的算法（Generate-and-Test Algorithms），通过生成候选解并测试其质量来优化问题。

遗传算法的基本流程为：初始化，评估，选择，重组，变异，评估子代，选择下一代，最后如果发现达到最优解后，算法结束

#### 遗传算法的特点：

- 基于种群**：遗传算法通过多个候选解（种群）进行搜索，而非单一解。
- 信息重组**：利用重组操作（如交叉）将候选解的信息混合生成新的解。
- 随机性**：遗传算法是随机搜索算法，通过随机性避免陷入局部最优。

## 概念解释

### 表示 (Representation)

- Phenotype (表现型)**：在问题上下文中形成可能解的对象（实际解）。
- Genotype (基因型)**：表现型的编码形式，GA中操作的基本单位。

遗传算法包含两个数据转换操作，一个是从表现型到基因型的转换，将搜索空间中的参数或解转换成遗传空间中的染色体或个体，这个过程称为编码(coding)。另一个是从基因型到表现型的转换，即将个体转换成搜索空间中的参数，这个过程称为解码(decode)。

### 变异操作 (Mutation Operator)

变异操作是单一变量操作，用于在现有个体的基础上生成新的个体。通过改变基因型中的某些部分产生变异体 (Mutant)，即子代个体。变异可以产生新的特征，通过探索搜索空间避免局限。

### 交叉操作 (Crossover Operator)

当两个生物机体配对或复制时，它们的染色体相互组合，产生一对由双方基因组成的新的染色体。这一过程称为交叉(crossover)。交叉操作是二元变量操作，通过从两个父代个体中重组基因信息生成一个或两个子代个体。

- 特点**：具有随机性，交叉的具体方式依赖于随机决定。
- 原理**：将父代具有的不同且优良的特性结合，生成更好的子代，从而提高种群的整体适应度。

## 父母选择机制 (Parent Selection Mechanism)

通过区分个体的质量，选择适应度较高的个体成为下一代的父母。高质量个体有更高的概率被选为父母，而低质量个体通常也保留小的被选概率，以避免过于贪婪而陷入局部最优解。

## 生存者选择机制 (Survivor Selection Mechanism)

根据个体的质量决定哪些个体可以进入下一代。由于种群数量不变，因此需要选择哪些个体留下。

- 特点：**基于适应度值 (Fitness-biased)**：优先选择适应度较高的个体。生存者选择通常是**确定性的 (Deterministic)**，与父母选择的随机性不同。具体而言，排序父代和子代的集合，选择适应度最高的一部分个体，或者仅从子代中选择（基于年龄偏向）。

## 初始化和终止条件 (Initialization and Termination Condition)

**初始化：**

- 大多数情况下，初始种群是随机生成的。
- 如果有固有知识，设法把握最优解的分布范围。然后在此分布范围内设定初始群体。

**终止条件：**

- 遗传算法是随机性的，无法保证一定能找到最优解。
- 常见的终止条件：
  - 达到最大计算时间或迭代次数。
  - 适应度评估次数达到设定上限。
  - 适应度在一段时间内改善幅度低于阈值。

## 种群(Population)

种群保存所有可能的候选解，是基因型的多重集合。遗传算法中，种群数量是固定的。过早收敛 (Premature Convergence) 是遗传算法中的常见问题，如果种群多样性过快丧失，算法可能陷入局部最优解。

## 遗传算法的流程

### 种群的初始化

在遗传算法 (GA) 中，初始种群是随机生成的。某些遗传算法会使用特殊技术来生成更高质量的初始种群，以加速进化过程。

### 选择/复制(Reproduction)

**选择操作也被称为复制操作，是从当前群体中按照一定概率选出优良个体。使它们有机会作为父代繁殖下一代子孙。判断个体优良与否的准则是各个个体的适应度值。**

1. 代际繁殖(Generational Reproduction)：循环 $N/2$ 次，其中 $N$ 是种群大小。每次循环中，根据当前的选择程序选择两个染色体，这两个染色体作为父母产生两个子代，最终产生 $N$ 个新的染色体。
2. 稳态繁殖 (Steady-state Reproduction)：这种方法选择两个染色体，对它们进行交叉操作以获得一个或两个子代（可能还会应用变异），然后将结果重新安装回种群中；最不适合的个体将被销毁。

## 亲本选择机制

在遗传算法中，哪个个体被选择进行交叉是按照概率进行的。适应度大小的个体被选择的概率大，但不是一定说能够被选上。使用**适应度比例方法**进行选择：

适应度比例方法中，各个个体被选择的概率和其适应度值成比例。设群体规模大小为 $M$ ，个体 $i$ 的适应度值为 $f_i$ ，则这个个体被选择的概率为：

$$p_{si} = \frac{f_i}{\sum_{i=1}^M f_i}$$

## 选择个体的方法

选择操作是根据个体的选择概率确定哪个个体被选择进行交叉、变异等操作。

### 1. 轮盘赌选择(Roulette Wheel selection)：在遗传算法中使用的最多

在轮盘赌选择方法中，先按个体的选择概率产生一个转盘，转盘每个区的角度与个体的选择概率成比例，然后产生一个随机数，它落入转盘的哪个区就选择对应的个体交叉。

在实际计算时，可以按照个体顺序求出每个个体的**累积概率**，然后产生一个随机数，它落入累积概率的哪个区域就选择相应的个体交叉。例如，表 6.1 所示 11 个个体的适应度、选择概率和累积概率。为了选择交叉个体，需要进行多轮选择。例如，第 1 轮产生一个随机数为 0.81，落在第 5 个和第 6 个个体之间，则第 6 个个体被选中。第 2 轮产生一个随机数为 0.32，落在第 1 个和第 2 个个体之间，则第 2 个个体被选中。依此类推。

表 6.1 个体适应度、选择概率和累积概率

个体	1	2	3	4	5	6	7	8	9	10	11
适应度	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.1
选择概率	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.05	0.03	0.02	0.01
累积概率	0.18	0.34	0.49	0.62	0.73	0.82	0.89	0.94	0.97	0.99	1.00

### 2. 基于排名的选择 (Rank-based selection)：

选择概率基于染色体在种群中的相对排名或位置，而不是绝对适应度。

### 3. 锦标赛选择 (Tournament-based selection)：

- 锦标赛选择是从种群中随机选择  $K$  个父母，然后返回其中最适应的一个。

## 交叉(Crossover)

当两个生物机体配对或者复制时，它们的染色体相互混合，产生一对由双方基因组成的新的染色体。这一过程称为交叉(crossover)或者重组(recombination)。

### 单点交叉(Single-point crossover)

随机生成一个数字（小于或等于染色体长度）作为交叉位置，保持数字前面的比特不变，然后交换两个亲本在交叉位置后的比特

- 亲本1: 7 3 | 7 6 1 3
- 亲本2: 1 7 | 4 5 2 2
- 子代1: 7 3 | 4 5 2 2
- 子代2: 1 7 | 7 6 1 3

## 两点交叉(Two-point crossover)

与单点交叉类似，但需要选择两个位置，并且只交换这两个位置之间的比特。这种交叉方法可以保留染色体的首尾部分，只交换中间部分。

- 亲本1: 7 3 | 7 6 | 1 3
- 亲本2: 1 7 | 4 5 | 2 2
- 子代1: 7 3 | 4 5 | 1 3
- 子代2: 1 7 | 7 6 | 2 2

## 均匀交叉 (Uniform Crossover)

均匀交叉中，第一个亲本的每个基因有0.5的概率与第二个亲本的对应基因交换。例如，对于每个位置，随机生成一个0到1之间的数字，如果生成的数字小于0.5，则子代1从亲本1获得基因，子代2从亲本2获得基因，否则相反。

## 变异(Mutation)

进化机制除了能够改变已有的特征，还能产生新的特征。如果生物繁殖仅仅是上述的交叉过程，那么即使经历成千上万代以后，适应能力最强的成员最强的特征也和初始群体中最强的特征一样。发送变异的概率通常很低，但在经历多代以后变异就会很明显。**在遗传算法中，变异是将个体编码中的一些位进行随机的变化。**变异的主要目的是维持群体的多样性。

## 反转(Inversion)

作用于单个染色体，它在染色体上随机选择两个点，然后反转这两个点之间元素的顺序。

例如：给定一个染色体序列 3 8 4 8 6 7。如果我们随机选择位置 2 和 5 并应用反转操作，那么我们将得到新的序列：3 6 8 4 8 7。

## 位点变异

对于染色体上的基因来说，随机选择一个或者多个基因，并对这些基因的基因值以变异概率 $P_m$ 进行变动

- 例子：假设我们已经使用交叉得到了一个新的字符串：7 3 4 5 1 3。假设变异率为 0.001。对于第一个比特 7，我们随机生成一个0到1之间的数字。如果这个数字小于变异率（0.001），那么第一个比特 7 需要变异。我们再生成一个1到最大值8之间的数字，得到一个数字（例如 2）。现在第一个比特变异为 2。我们对其他比特重复相同的过程。在我们的例子中，如果只有第一个比特变异，而其余的比特没有变异，那么我们将得到一个新的染色体：2 3 4 5 1 3。

## 遗传算法中的约束处理(Constraint)

### 直接处理(Direct Constraint Handling)

不通过适应度函数反映约束条件是否被违反，因此不会对满足约束的个体产生偏好。这需要确保种群中的个体在任何时候都是可行的（满足约束）。

### 间接处理 (Indirect Constraint Handling)

将约束视为惩罚(penalties)，将违反约束的行为加入惩罚项，作为优化目标的一部分。遗传算法通过最小化惩罚值来间接实现约束的满足。通常惩罚项是针对违反约束的情况设定的，某些遗传算法还会为错误初始化的变量或到可行解的距离设置惩罚值。



# 记忆算法(Memetic Algorithm)

记忆算法 (Memetic Algorithm, MA) 是一种遗传算法的改进变体, 由 **Moscato 和 Norman** 在1992年提出。它基于遗传算法中的进化思想, 但不同于仅依赖生物学进化的遗传算法, 记忆算法还融合了**文化进化**或**思想进化**的概念。核心思想为: 不仅通过与其他解的重组改进一个解, 还允许一个解通过自身的适应性调整进行改进。

## Meme (记忆单元)

在记忆算法中, 记忆单元 (Meme) 是信息的基本单位。与基因不同, 记忆单元在个体之间传递时, 可以被持有的个体进一步改进, 而基因在传递过程中通常保持不变。这种“改进”可以理解为个体对当前信息的本地优化。

记忆算法是一种结合了局部搜索技术的进化算法。它通过局部搜索 (local search) 进一步优化个体, 增强种群中解的质量。

## 记忆算法基本流程

遗传算法的基本操作:

- 包括**变异** (mutation) 和**交叉** (crossover) 操作, 用于在解的种群中生成新解。
- 交叉操作倾向于从种群中的基因中寻找最佳解, 而变异操作作为种群注入新的基因。

局部搜索技术 (Local Search) :

- 通常使用**爬山法 (Hill Climbing)** 等技术, 在变异生成的新解的基础上, 进一步优化。
- 局部搜索可以减少搜索空间, 从而更快地找到高质量的解。

Memetic Algorithm 的改进:

- 在遗传算法中, 解的进化主要依赖种群间的交叉和变异。
- 而在记忆算法中, 局部搜索的引入使得解在其局部邻域内进行进一步优化。

## 与遗传算法对比

遗传算法 (GA) :

- 基于生物学的进化模型, 强调种群间的遗传和变异。
- 缺点: 可能陷入局部最优, 且收敛速度较慢。

记忆算法 (MA) :

- 在进化过程中结合了局部搜索优化, 模仿文化或思想的进化过程。
- 优点: 通过局部搜索更快地找到高质量解, 减少陷入局部最优的风险。