集成学习 ensemble learning

集成学习基础知识

集成学习通过构建并结合多个学习器来完成学习任务

有时也被称为多分类器系统、基于委员会的学习等

集成学习先产生一组“个体学习器individual learner” 再用某种策略将它们结合起来

集成学习分同质集成和异质集成。同质集成中的个体学习器由相同的学习算法生成，个体学习器称为基学习器；异质集成中的个体学习器由不同的学习算法深生成，个体学习器称为组件学习器。

集成学习要显著优于单一个体学习器必须满足两个必要条件：1）个体学习器之间应该是相互独立的；2）个体学习器应当好于随机猜测学习器。

- 满足第2个条件往往比较容易，因为在现实任务中，出于种种考虑，比如希望使用较少的个体学习器，或者是希望重用关于常见学习器的一些经验等，人们往往会使用比较强的个体学习器。
- 满足第1个条件往往比较困难，个体学习器是为解决同一个 问题训练出来的，显然不可能互相独立！事实上，个体学 习器的“准确性”和“多样性”本身就存在冲突。一般的， 准确性很高之后，要增加多样性就需要牺牲准确性。
- 因此，如何产生“好而不同”的个体学习器是集成学习研 究的核心！

集成学习常用方法

如何在保持个体学习器足够“好”的前提 下增强多样性呢？ 一般的思路是在学习过程中引入 **随机性**，常用方法主要包括：

训练样本扰动

输入属性扰动

输出标记扰动

算法参数扰动

混合扰动

1）：训练样本扰动

- 训练样本扰动通常是用抽样的方法从原始训练样本 集中产生出不同的样本子集，然后再利用不同的样 本子集训练出不同的个体学习器。比如，在装袋 Bagging中使用自助采样，在提升Boosting中使用序 列采样。
- 此类方法简单高效，使用也最广，但只对不稳定基 学习器有效，比如决策树、神经网络等；对稳定基 学习器效果不明显，比如线性学习器、支持向量机、 朴素贝叶斯、k-最近邻学习器等。

2）：输入属性扰动

- 输入属性扰动通常是从初始属性集中抽取出若干个属性子集，然后利用不同的属性子集训练出不同的 个体学习器。比如，随机子空间[Ho,1998]和随机 森林[Breiman, 2000]。
- 此类方法对包含大量冗余属性的数据集有效，但若 数据集只包含少量属性，或者冗余属性很少，则不 宜使用。

3)：输出标记扰动

- 输出标记扰动通常是对训练样本的类标记稍作变动， 比如，可将原来的多分类问题随机转化多个二分 问题来训练基学习器，纠错输出码[Dietterich and Bakiri, 1995]就是这类方法的典型代表。
- 此类方法对类数足够多的数据集有效，但若数据集 包含的类数较少，则不宜使用

4)：算法参数扰动

- 算法参数扰动通常是通过随机设置不同的参数来训 练差别较大的个体学习器。比如，神经网络的隐层 神经元数、初始连接权值等
- 此类方法对参数较多的算法有效，对参数较少的算 法，可通过将其学习过程中某些环节用其他类似方 式代替?从而达到扰动的目的

5）：混合扰动

- 混合扰动是指在在同一个集成算法中同时使用上述 多种扰动方法。比如，随机森林就同时使用了训练 样本扰动和输入属性扰动。

集成学习结合策略

> - 到此为止，我们都是在讨论：**怎么通过集成学习的常用方法生成“好而不同”的个体学习器？**
> - 剩下来要解决的问题：**怎么结合生成的个体学习器，具体的结合策略有哪些？**
> - 对分类任务来说，最常见的结合策略就是投票法(voting)，具体包括：
> ✓ 绝对多数投票法(majority voting)
> ✓ 相对多数投票法(plurality voting)
> ✓ 加权投票法(weighted voting)

- 绝对多数投票法：即若某标记得票过半数，则分类 为该标记，否则拒绝分类。
- 相对多数投票法：分类为得票最多的标记，若同时 有多个标记获最高票，则从中随机选取一个

- 加权投票法：给每个个体学习器预测的类标记赋一 个权值，分类为权值最大的标记。这里的权值通常 为该个体学习器的分类置信度（类成员概率）。

演化学习

演化学习基础知识

定义：演化学习基于演化算法提供的优化工具设计机器学习算法。演化算法：或称“进化算法”，它 是一个“算法簇 ”，其灵感都来自于大自然的生物进化。演化算法有 很多的版本，比如，有不同的遗传基因表达方式，不 同的交叉和变异算子，以及不同的再生和选择方法

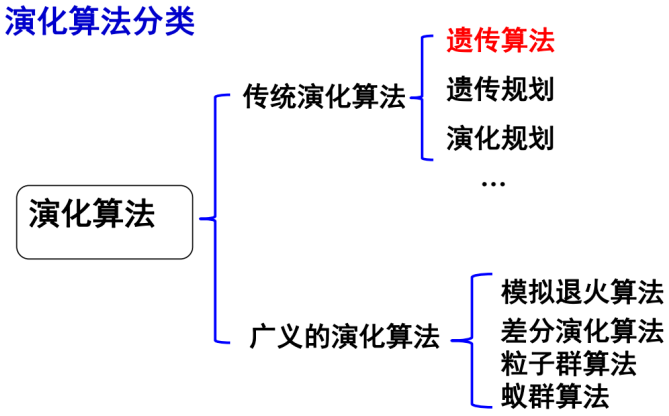
与传统的优化算法相比，演化算法的特点在于

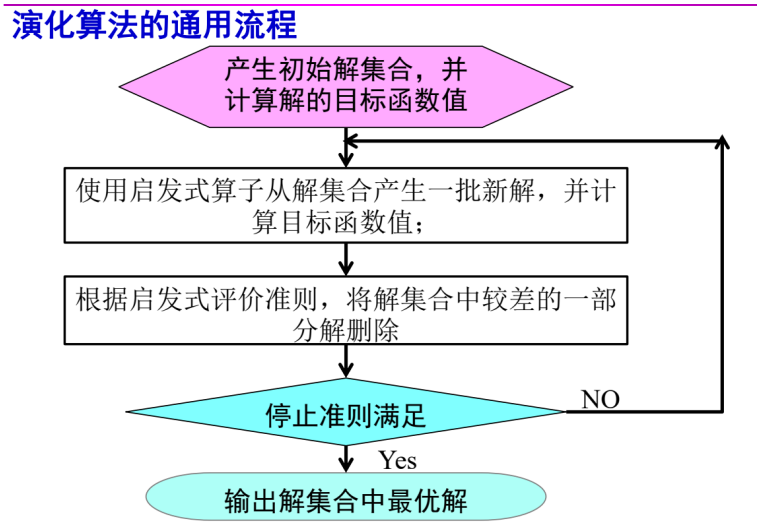
具有高鲁棒性和广泛适应性；

具有自组织、自适应、自学习的特性；

本质并行性；

能够不受问题性质的限制，有效处理传统优化算法难以解决的复杂问题。





遗传算法

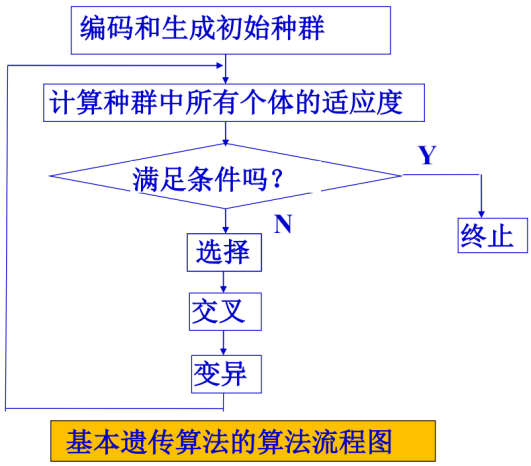
定义：遗传算法是模拟生物在自然环境中的遗传和演化过程而形 成的一种自适应全局优化概率搜索算法。

基本思想

从初始种群出发，采用优胜劣汰、适者生存的自然法则选择个体 ， 并通过杂交、变异来产生新一代种群，如此逐代演化，直到满 足目标为止。

特点

- 遗传算法是从问题解空间多点并行搜索，而非从单个解开始 搜索；
- 遗传算法利用目标函数的适应度这一信息而非利用导数或其它辅助信息来指导搜索；
- 遗传算法利用选择、交叉、变异等算子而不是利用确定性规 则进行操作。



基本遗传算法的算法流程图

遗传算法编码：

（1）二进制编码

二进制编码是将原问题的结构变换为染色体的位串结构。在二进制编码中，首先要确定二进制字符串的长度，该长度与变量的定义域和所求问题的计算精度有关。

例如： 假设变量x的定义域为[-2，5]，其要求精度为10E-6，则我们需要将[-2,5]分成7000000个等长小区间，每个小区间用一个二进制位串来表示，于是二进制位串长度至少23位。这是因为：

$$4194304 = 2^{22} < 7000000 < 2^{23} = 8388608$$

二进制编码存在的主要缺点：汉明悬崖。

例如，7和8的二进制数分别为0111和1000，当算法从7改进到8时，就必须改变所有的位。

（2）实数编码

实数编码是将每个个体的染色体都用某一范围的一个实数（浮点数）来表示，其编码长度等于问题变量的个数。

这种编码方法是将问题的解空间映射到实数空间上，然后在实数空间上进行遗传操作。

实数编码适应于多维、高精度要求的连续函数优化问题。

（3）有序串编码

很多组合优化问题中，目标函数的值不仅与表示解的字符串中各字符的值有关，而且与其所在字符串的位置有关，这时，需要采用独特的有序串编码，比如旅行商优化问题。

适应性度量：

适应度函数是用于对个体的适应性，进行度量的函数。通常，一个个体的适应度值越大，它被遗传到下一代种群中的概 率也就越大。

(1)常用的适应度函数

原始适应度函数：直接将待求解问题的目标函数f(x)定义为遗传算法的适应度函数。

例如，在求解极值问题 $max_{x\in[a,b]}f(x)$ 时，f(x)即为x的原始适应度函数。

采用原始适应度函数

优点：能够直接反映出待求解问题的最初求解目标

缺点：是有可能出现适应度值为负的情况

（2）标准适应度函数

标准适应度函数：在遗传算法中，一般要求适应度函数值非负，并且，适应度值越大越好，这就往往需要对原始适应度函数进行某种变换，将其转换为标准的度量方式，以满足演化操作的要求，这样所得到的适应度函数被称为标准适应度函数f_Normal(x)。

例如：对极小化问题，其标准适应度函数可定义为

$$f_{nomal}(x)=\begin{cases} f_{\max}(x)-f(x) & 当f(x)<f_{\max}(x) \\ 0 & 否则 \end{cases}$$

其中，f_{max} (x)是原始适应函数f(x) 的一个上界。如果f_{max} (x) 未知，则可用当前代或到目前为止各演化代中的f (x) 的最大值来代替。

选择操作

选择操作是指根据选择概率按某种策略从当前种群中挑 选出一定数目的个体，使它们能够有更多的机会被遗传到下 一代。常用的选择策略:比例选择，排序选择，竞技选择。

比例选择：每个个体被选中的概率与其适应度大小成正比。比如在轮盘赌选择算法中，个体被选中的概率取决于该个体的相对适应度。而相对适应度的定义为：

$$P(x_i)=\frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}$$

其中，$P(x_i)$是个体x_i的相对适应度，即个体x_i被选中的概率；$f(x_i)$是个体x_i的原始适应度值；分母是种群的累加适应度值

轮盘赌选择算法的基本思想是：根据每个个体的选择概率$P(x_i)$将一个圆盘分成N个扇区，其中第i个扇区的中心角为：

$$2\pi\frac{f(x_i)}{\sum_{j=1}^{N} f_i(x_j)}=2\pi p(x_i)$$

再设立一个移动指针，选择时，假想转动指针，当指针静止时，若它指向第i个扇区，则选择个体i。

从统计角度看，个体的适应度值越大，其对应的扇区的面积越大，被选中的可能性也越大。

交叉操作

交叉操作是指按照某种方式对选择的父代个体的染色体的 部分基因进行交叉重组，从而形成新的个体。

交叉重组是自然界中生物遗传进化的一个主要环节，也是 遗传算法中产生新的个体最重要的方法之一。根据个体编码方 法的不同，遗传算法中的交叉操作可分为二进制交叉和实值交叉两种类型。

二进制交叉是指二进制编码情况下所采用的交叉操作， 它主要包括单点交叉、两点交叉和均匀交叉等方法。

变异操作

变异是指对选中个体的染色体中的某些基因进行变动，以形成新的个体。变异也是生物遗传和自然演化中的一种基本现象，它可增强种群的多样性。遗传算法中的变异操作增加了算法的局部随机搜索能力，从而可以维持种群的多样性。根据个体编码方式的不同，变异操作可分为二进制变异和实值变异两种类型。

（1）二进制变异：该变异方法是先随机地产生一个变异位，然后将该变异位置上的基因值由“0”变为“1”，或由“1”变为“0”，产生一个新的个体。例如：设变异前的个体为A=0 0 1 1 0 1，若随机产生的变异位置是2，则该个体的第2位由“0”变为“1”。 变异后的新的个体是A'= 0 1 1 1 0 1 。

（2）实值变异

◆基于位置的变异方法

该方法是先随机地产生两个变异位置，然后将第二个变异位置上的基因移动到第一个变异位置的前面。

例 设选中的个体向量C=20 16 19 12 21 30，若随机产生的两个变异位置分别时2和4，则变异后的新的个体向量是：

C'= 20 12 16 19 21 30

◆基于次序的变异

该方法是先随机地产生两个变异位置，然后交换这两个变异位置上的基因。

例 设选中的个体向量D=20 12 16 19 21 30，若随机产生的两个变异位置分别时2和4，则变异后的新的个体向量是：

D'= 20 19 16 12 21 30

演化神经网络

概述

演化神经网络是基于演化计算和神经网络两大研究方向， 将二者有机融合而产生的一种全新神经网络模型。 这种模型把演化计算的自适应机制与神经网络的学习机制 有机的结合在一起，有效地克服了传统人工神经网络的很多缺 点。 演化神经网络模型的一个主要特点就是它对动态环境的自 适应性。这种自适应性过程通过演化的三个等级实现，即连接 权值和阈值、网络结构和学习规则的演化

根据演化神经网络实现的三个等级，演化神经网络模型有以下四种不同的类型：

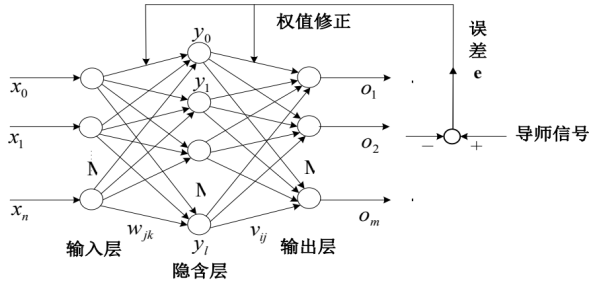
初始权值和阈值演化

网络结构演化

结构和权值阈值同时演化

学习规则演化

初始权值阈值演化---以遗传算法优化BP神经网络的连接权值和阈值为例



BP网络是一类多层的前馈神经网络，目前是人工神经网络中应用最广泛的算法之一，但是存在一些缺陷，比如说学习收敛速度慢、不能保证收敛到全局极小点，网络结构不易确定。

优化目标：用遗传算法优化BP神经网络的初始权值和阈值， 优化后的BP神经网络具有更好的预测精度。

算法的基本思路：

（1）对神经网络的初始权值和阈值进行编码

（2）然后对种群所有个体进行解码，生成多个神经网络

（3）对每个神经网络进行BP训练，然后以均方根误差作为评 价标准，对种群中所有个体进行适应度评价。

（4）进行选择、交叉、变异操作，产生新的种群

（5）判断是否达到停止条件，否则转到（2）运行。

演化学习问题与挑战

存在的问题

- 对演化算法这类随机性启发式优化算法而言， 其理论研究不足，比如优化效率高低、与最优解的逼近程度 如何、启发式算子效用评估等问题难以有严格的答案，这导 致了演化学习也缺乏有效的理论解释。

挑战：

- 近些年，一方面学习模型变得复杂、数据增长迅速、 一方面对模型训练时间有严格约束，如何使得演化学习能够 进行有效、快速地优化，还有待深入的研究。