

先进算法讲义

在本讲义中，我们将着重讲述一些数学建模中常用的算法，包括神经网络算法、遗传算法、模拟退火算法和模糊数学方法。用这些算法可以较容易地解决一些很复杂的，常规算法很难解决的问题。由于这些算法都有着很深的理论背景，因此，本讲义中不可能也没有必要详细地讨论这些算法的理论，我们的目标在于应用，大家只需大概了解这些算法的原理，知道能用这些算法解决一类什么样的问题，并能应用这些算法解决数学建模中的一些问题即可。

因为着眼于应用，所以我们还提供了一些程序代码，使用者只需套用这些程序，便可使问题得到很好的解决。

第一节 神经网络

1. 神经网络的简单原理

人工神经网络是根据人的认识过程而开发出的一种算法。假如我们现在只有一些输入和相应的输出，而对如何由输入得到输出的机理并不清楚，那么我们可以把输入与输出之间的未知过程看成是一个“网络”，通过不断地给这个网络输入和相应的输出来“训练”这个网络，网络根据输入和输出不断地调节自己的各节点之间的权值来满足输入和输出。这样，当训练结束后，我们给定一个输入，网络便会根据自己已调节好的权值计算出一个输出。这就是神经网络的简单原理。

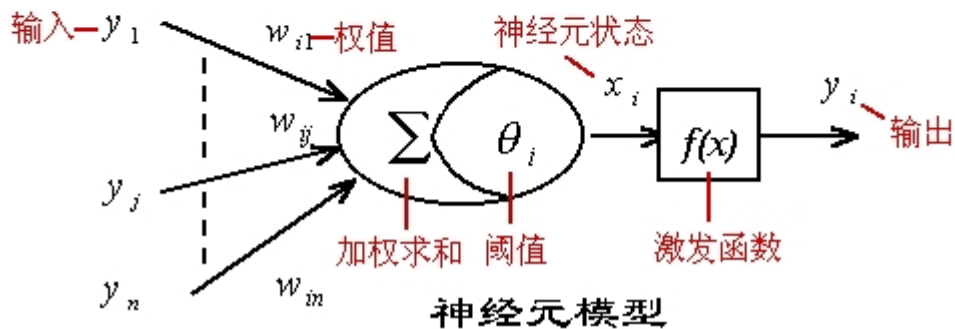
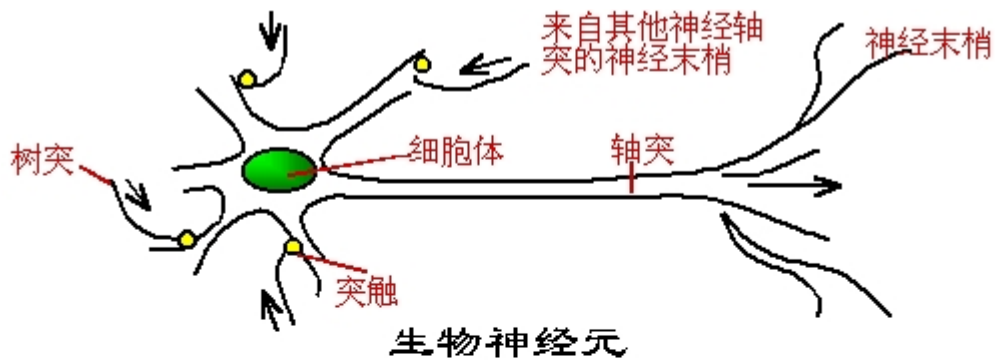
2. 神经元和神经网络的结构

如上所述，神经网络的基本结构如下图所示：



神经网络一般都有多层，分为输入层，输出层和隐含层，层数越多，计算结果越精确，但所需的时间也就越长，所以实际应用中要根据要求设计网络层数。

神经网络中每一个节点叫做一个人工神经元，他对应于人脑中的神经元，两者的结构比较如下图：



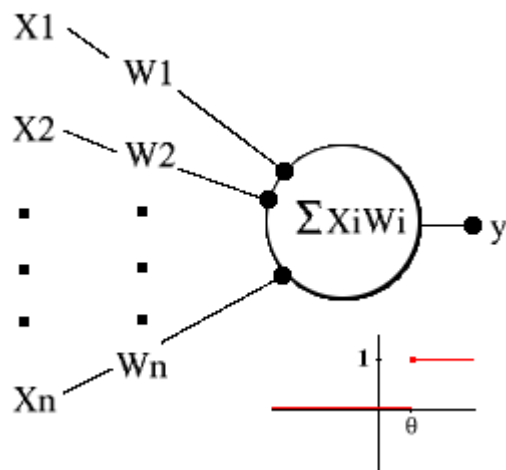
一个人工神经元一般有多个输入和一个输出，另外有一个激发函数，不同的激发函数对应了不同的网络，也决定了网络的用途。

3. 神经网络的分类

神经网络按照网络结构和激发函数的不同可分为许多种，我们在此只是对感知器和 BP 网络进行简介。

感知器：

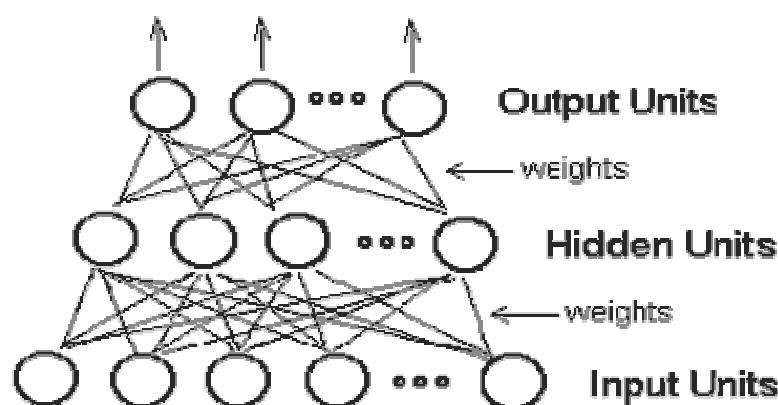
最早也是最简单的一种神经网络，它的神经元激发函数为阶跃函数，其神经元结构如下图：



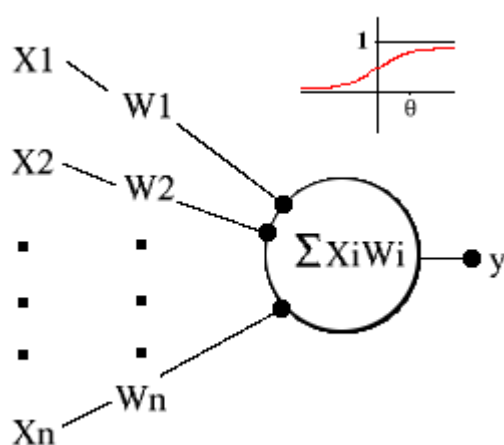
感知器主要用于分类。

BP 网络：

应用得最为广泛，最为重要的一种神经网络。这种网络一般有多层，网络结构如下：



BP 网络的激发函数一般采用 S 型函数，如正切或对数函数，其神经元结构如下：



BP 网络的用途十分广泛，可用于以下方面：

函数逼近：用输入矢量和相应的输出矢量训练一个网络逼近一个函数

模式识别：用一个特定的输出矢量将它与输入矢量联系起来

分类：把输入矢量以所定义的合适方式进行分类

数据压缩：减少输出矢量维数以便于传输或存储

4. 神经网络在数学建模中的应用

数学建模中有很多题目都可以用神经网络加以解决，比较典型的题目有：DNA 序列分类题（2000 年全国赛 A 题），癌症判断题（2001 年北京大学数学建模竞赛），乳房癌的诊断题（2001 年全国大学生数学建模夏令营 C 题）。下面我们使用神经网络的方法解决癌症判断题（2001 年北京大学数学建模竞赛），题目如下：

[附件](#)中的文件给出了一个 114 个基因, 60 个人的基因表达水平的样本. 其中前 20 个是癌症病人的基因表达水平的样本(其中还可能有子类), 其后的是 20 个正常人的基因表达信息样本, 其余的 20 个是待检测的样本(未知它们是否正常).

(1). 试设法找出描述癌症与正常样本在基因表达水平上的区别, 建立数学模型, 及识别方法, 去预测待检测样本是癌症还是正常样本.

(2). 设计图示 (可视化) 方法, 使得在你的数学模型下, 尽量清楚地表现癌症与正常样本在基因表达水平上的区别, 以及癌症样本中是否有子类.

这道题是很典型的用神经网络的分类问题，只需用感知器神经网络便能完成此分类工作，我们用前 40 组数据对网络进行训练，再用训练号的网络来计算后 20 组数据，便能得到分类的结果。详细的程序可参见本讲义附带的 Matlab 程序。

5. 神经网络的程序设计

该讲义附带了如下程序资源：

Matlab 神经网络工具箱函数简介

癌症判断题（2001 年北京大学数学建模竞赛）的 Matlab 程序

第二节 遗传算法

1. 遗传算法的简单原理

遗传算法(Genetic Algorithm, GA)是一种基于自然群体遗传演化机制的高效探索算法，它摒弃了传统的搜索方式，模拟自然界生物进化过程，采用人工进化的方式对目标空间进行随机化搜索。它将问题域中的可能解看作是群体的一个个体或染色体，并将每一个体编码成符号串形式，模拟达尔文的遗传选择和自然淘汰的生物进化过程，对群体反复进行基于遗传学的操作（遗传，交叉和变异），根据预定的目标适应度函数对每个个体进行评价，依据适者生存，优胜劣汰的进化规则，不断得到更优的群体，同时以全局并行搜索方式来搜索优化群体中的最优个体，求得满足要求的最优解。

我们先通过一个例子来了解遗传算法的原理：

假定我们要求函数 $f(x) = x^2$ 的极大值，其中 x 为自然数， $0 \leq x \leq 31$ 。现在，我们将每一个数看成一个生命体，通过进化，我们看谁能最后生存下来，谁就是我们所寻找的数。

①. 编码

我们将每一个数作为一个生命体，那么必须给其赋予一定的基因，这个过程叫做编码。我们可以把变量 x 编码成 5 位长的二进制无符号整数表示形式，比如 $x=13$ 可表示为 01101 的形式，也就是说，数 13 的基因为 01101。

②. 初始群体的生成

由于遗传的需要，我们必须设定一些初始的生物群体，让其作为生物繁殖的第一代，需要说明的是，初始群体的每个个体都是通过随机方法产生的，这样便可以保证生物的多样性和竞争的公平性。

③. 适应度评估检测

生物的进化服从适者生存，优胜劣汰的进化规则，因此，我们必须规定什么样的基因是“优”的，什么样的基因是“劣”的，在这里，我们称为适应度。显然，由于我们要求 $f(x) = x^2$ 的最大值，因此，能使 $f(x) = x^2$ 较大的基因是优

的，使 $f(x) = x^2$ 较小的基因是劣的，因此，我们可以将 $f(x) = x^2$ 定义为适应度函数，用来衡量某一生物体的适应程度。

④. 选择

接下来，我们便可以进行优胜劣汰的过程，这个过程在遗传算法里叫做选择。注意，选择应该是一个随机的过程，基因差的生物体不一定会被淘汰，只是其被淘汰概率比较大罢了，这与自然界中的规律是相同的。因此，我们可以采取赌论的方式来进行选择。

⑤. 交叉操作

接下来进行交叉繁殖，随机选出两个生物体，让其交换一部分基因，这样便形成了两个新的生物体，为第二代。

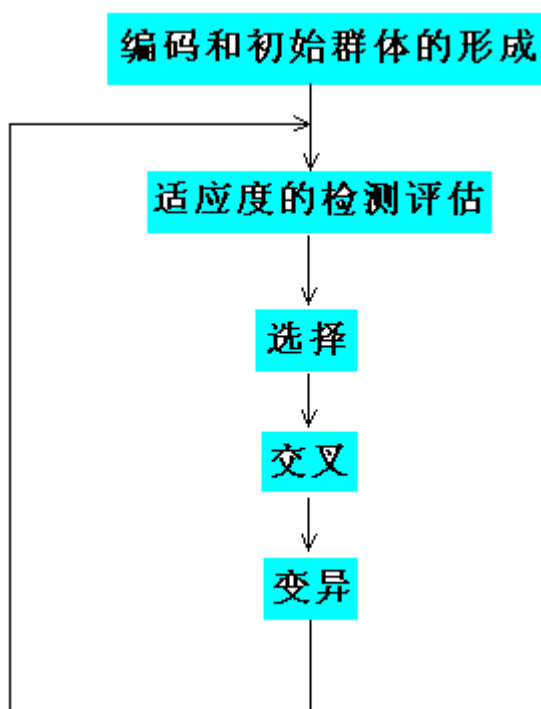
⑥. 变异

生物界中不但存在着遗传，同时还存在着变异，在这里我们也引入变异，使生物体的基因中的某一位以一定的概率发生变化，这样引入适当的扰动，能避免局部极值的问题。

以上的算法便是最简单的遗传算法，通过以上步骤不断地进化，生物体的基因便逐渐地趋向最优，最后便能得到我们想要的结果。

2. 遗传算法的步骤

从上面的例子中，我们便能得到遗传算法的一般步骤，如下图所示：



3. 遗传算法的应用

遗传算法主要是用来寻优，它具有很多优点：它能有效地避免局部最优现象，有及其顽强的鲁棒性，并且在寻优过程中，基本不需要任何搜索空间的知识和其他辅助信息等等。

为了介绍遗传算法的应用，我们将前面的例子进行完，整个过程如下：

初始群体	01101	11000	01000	10011
x 的值	13	24	8	19
适应度 $f(x) = x^2$	169	576	64	361
选择概率	0.14	0.49	0.06	0.31
选择上的计数（来自赌轮）	1	2	0	1
交叉处	0110/0	1100/0	11/000	10/011
下一代群体	01100	11001	11011	10000
x 的值	12	25	27	16
适应度 $f(x) = x^2$	144	625	729	256

4. 遗传算法的程序设计

本讲义提供了遗传算法求 $\sin^2 x$ 的 C 语言程序

第三节 模拟退火算法

1. 模拟退火算法的简单原理

模拟退火算法主要用于解决组合和优化问题，它是模拟物理中晶体物质的退火过程而开发的一种优化算法。

在对固体物质进行模拟退火处理时，通常先将它加温熔化，使其中的粒子可自由运动，然后随着温度的逐渐下降，粒子也逐渐形成了低能态的晶格。若在凝结点附近的温度下降速率足够慢，则固体物质一定会形成最低能态的基态。

对于组合优化问题来说，它也有这样的类似过程。组合优化问题解空间中的每一点都代表一个具有不同目标函数值的解。所谓优化，就是在解空间中寻找目标函数最小（大）解的过程。若把目标函数看成能量函数，某一控制参数视为温度 T ，解空间当作形态空间，那么寻找基态的过程也就是求目标函数极小值的优化过程。

2. 模拟退火算法的应用

如前所述，模拟退火算法主要用于解决组合和优化问题，典型的例子是用模拟退火算法来解决 TSP 问题，本讲义附带了用模拟退火算法解决 TSP 问题的算法，有兴趣的同学可以参考

模拟退火算法也经常用在神经网络中，本讲义也附带了一个模拟退火算法用在神经网络中的程序。

第四节 模糊数学方法

1. 模糊数学的简单原理

①. 模糊集

对于传统集合，一个元素要么属于这个集合，要么不属于这个集合，我们用 1 表示元素属于该集合，用 0 表示元素不属于这个集合。

模糊集合论认为，一个元素可以不完全属于一个集合，用 $[0, 1]$ 之间的一个数来表示一个元素属于一个集合的程度，这个数叫做该元素属于该集合的隶属度。

②. 模糊概念的清晰化

我们可以设定一个数（比如 0.5），当一个元素的隶属度大于这个数时，我们就可以认为该元素属于这个集合，这就是模糊概念的清晰化。

2. 模糊数学的应用举例

模糊数学在数学建模中主要用于模糊决策，这也是决策论中很重要的一部分内容，我们通过下面的例子来说明。

有 7 名同学进行了毕业论文的答辩，有 10 位教授要对同学的答辩做出评分，评分采取等级制，各等级的分数如下：

评分标准	一等	二等	三等	四等	五等	六等
分数	10	8	6	4	3	1

将参加评选的同学进行两两比较评分，例如 $x_1 \rightarrow x_2$ （以先评价的 x_2 为基准，后评价的 x_1 为对象进行相对比较评分）。比如 10 人所给相加的总分为 80 分，则学生 x_1 对

x_2 的优先选择比为 $r_{12} = \frac{80}{100} = 0.80$ (其中分母 100 为 10 个教授都给最高分时的总分)

分) 相应的 x_2 对 x_1 的优先选择比为 $r_{21} = 1 - r_{12} = 1 - 0.8 = 0.2$ 。利用上面的方法便可以得到下表:

	一等	二等	三等	四等	五等	六等	评 选 组 给 被 评 人 总 分	优 先 选 择 比 r_{ij}
	分 数							
	10	8	6	4	3	1		
$x_1 \rightarrow x_2$	3	4	3				80	0.80
$x_1 \rightarrow x_3$	1	5	3	1			72	0.72
$x_1 \rightarrow x_4$	2	5	2	1			76	0.76
$x_1 \rightarrow x_5$	6	2	1	1			86	0.86
$x_1 \rightarrow x_6$	3	1	4	2			70	0.70
$x_1 \rightarrow x_7$	4	3	1	2			78	0.78
$x_2 \rightarrow x_3$	1	4	3	2			68	0.68
$x_2 \rightarrow x_4$	4	2	1	1	1	1	70	0.70
$x_2 \rightarrow x_5$	2	2	3	2		1	63	0.63
$x_2 \rightarrow x_6$	5	2	1	2			80	0.80
$x_2 \rightarrow x_7$	3	2	3	1	1		71	0.71
$x_3 \rightarrow x_4$		2	3	4	1		53	0.53
$x_3 \rightarrow x_5$	2	3		1	2	2	56	0.56
$x_3 \rightarrow x_6$		2	2	1	2	3	41	0.41
$x_3 \rightarrow x_7$	1	2	3	2		2	54	0.54
$x_4 \rightarrow x_6$			2	3	1	4	31	0.31
$x_4 \rightarrow x_6$		1	1	2	3	3	34	0.34
$x_4 \rightarrow x_7$	1		1	2	3	3	36	0.36
$x_5 \rightarrow x_6$	1	2	1	2	1	3	46	0.46
$x_5 \rightarrow x_7$	1		2	3	1	3	40	0.40
$x_6 \rightarrow x_7$	1	2		2	2	3	43	0.43

于是可以得到优先选择矩阵:

$$R^{(1)} = \begin{pmatrix} 1 & 0.80 & 0.72 & 0.76 & 0.86 & 0.70 & 0.78 \\ 0.20 & 1 & 0.68 & 0.70 & 0.63 & 0.80 & 0.71 \\ 0.28 & 0.32 & 1 & 0.53 & 0.56 & 0.41 & 0.54 \\ 0.24 & 0.30 & 0.47 & 1 & 0.31 & 0.34 & 0.36 \\ 0.14 & 0.37 & 0.44 & 0.69 & 1 & 0.46 & 0.40 \\ 0.30 & 0.20 & 0.59 & 0.66 & 0.54 & 1 & 0.43 \\ 0.22 & 0.29 & 0.46 & 0.64 & 0.60 & 0.57 & 1 \end{pmatrix}$$

$\lambda = 0.70$, 得 λ -截矩阵为:

$$R_{0.7}^{(1)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

λ -截矩阵 $R_{0.7}^{(1)}$ 的第一行元素都等于 1, 说明只有 x_1 的优越度超过了 0.70, 所以学生 x_1 为第一优越对象。

除去 $R^{(1)}$ 中第一优越对象 x_1 所在的行和列, 得到新的模糊优先关系矩阵 $R^{(2)}$ 为:

$$R^{(2)} = \begin{pmatrix} 1 & 0.68 & 0.70 & 0.63 & 0.80 & 0.71 \\ 0.32 & 1 & 0.53 & 0.56 & 0.41 & 0.54 \\ 0.30 & 0.47 & 1 & 0.31 & 0.34 & 0.36 \\ 0.37 & 0.44 & 0.69 & 1 & 0.46 & 0.40 \\ 0.20 & 0.59 & 0.66 & 0.54 & 1 & 0.43 \\ 0.29 & 0.46 & 0.64 & 0.60 & 0.57 & 1 \end{pmatrix}$$

取 $\lambda = 0.63$, 得:

$$R_{0.63}^{(2)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix},$$

λ -截矩阵 $R_{0.63}^{(2)}$ 的第一行元素都等于 1, 应取 x_2 为第二优越对象。

除去 $R^{(2)}$ 中第二优越对象 x_2 所在的行与列, 得到新的模糊优先关系矩阵 $R^{(3)}$ 为

$$R^{(3)} = \begin{pmatrix} 1 & 0.53 & 0.56 & 0.41 & 0.54 \\ 0.47 & 1 & 0.31 & 0.34 & 0.36 \\ 0.44 & 0.69 & 1 & 0.46 & 0.40 \\ 0.59 & 0.66 & 0.54 & 1 & 0.43 \\ 0.46 & 0.64 & 0.60 & 0.57 & 1 \end{pmatrix}.$$

取 $\lambda = 0.46$, 得

$$R_{0.46}^{(3)} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

可知 x_7 为第三优越对象。

除去 $R^{(3)}$ 中第二优越对象 x_7 所在的行与列，得到新的模糊优先关系矩阵 $R^{(4)}$ 为

$$R^{(4)} = \begin{pmatrix} 1 & 0.53 & 0.56 & 0.41 \\ 0.47 & 1 & 0.31 & 0.34 \\ 0.44 & 0.69 & 1 & 0.46 \\ 0.59 & 0.66 & 0.54 & 1 \end{pmatrix},$$

取 $\lambda = 0.54$ ，得

$$R_{0.54}^{(4)} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

可知 x_6 为第四优越对象。

类似地，可得 $R^{(5)}$ 为

$$R^{(5)} = \begin{pmatrix} 1 & 0.53 & 0.56 \\ 0.47 & 1 & 0.31 \\ 0.44 & 0.69 & 1 \end{pmatrix},$$

取 $\lambda = 0.53$ ，得

$$R_{0.53}^{(5)} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

可知 x_3 为第五优越对象，同样，

$$R^{(6)} = \begin{pmatrix} 1 & 0.31 \\ 0.69 & 1 \end{pmatrix}$$

取 $\lambda = 0.69$ ，得

$$R_{0.69}^{(6)} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

可知 x_5 为第六优越对象，因此 7 名同学的排名为：

$$x_1, x_2, x_7, x_6, x_3, x_5, x_4$$