

模式线性可分时的一种单层感知器算法

夏东盛 李永涛 张 晓 蔡 创

(陕西工业职业技术学院 咸阳712000)

摘 要 在模式识别时常常需要对模式进行分类,线性可分模式的分类是最基本的一种。常用的线性分类算法是LMSE算法,它们在本质上都属于几何分类法,当模式线性可分时,一般都能达到令人满意的效果。然而考虑到LMSE算法并非是最简单和有效的线性分类算法,本文基于神经网络中单层感知器的概念,利用单层感知器可以把输入空间划分成两个区域来进行输入向量分类的特点,提出了在模式线性可分时用神经网络中单层感知器进行模式划分的一种新算法。然后对该线性分类算法的原理和算法过程进行了阐述,最后用MATLAB实现了这种分类算法,并解决了两个不同类型的线性模式的划分问题。

关键词 层感知器,模式识别,神经网络,线性分类算法

An Algorithm of Single Layer Perceptron for the Linearly Separable Patterns

XIA Dong-Sheng

(Shanxi Polytechnic Institute, Xianyang 712000)

Abstract On pattern recognition there are many patterns that need be classified, among which linearly separable patterns classification are most fundamental. LMSE algorithm that belongs to geometry classification algorithm is generally used, and linearly separable patterns can be classified well. Yet it is not simplest and validest, therefore, in this article, based on conception of single layer perceptron in neural network, in favor of the characteristic of single layer perceptron which gives single layer perceptron the ability to classify input vectors by dividing input space into two regions, a new algorithm of linearly separable patterns to be classified by single layer perceptrons were proposed, and the theory was illustrated. The linear classification algorithm was realized in MATLAB and two different problems of linear pattern classification were solved.

Keywords Single layer perceptrons, Pattern recognition, Neural network, Linear classification algorithm

1 引言

模式识别就是通过计算机用合适的数学方法来对包含大量数据集的模式进行处理和判读。在模式识别中,若一类模式满足下列条件:

1)每一模式类与其它模式类间可用单个判别平面分割;

2)每两类之间都可以分别用判别平面分隔开,则我们可以说这类模式线性可分。也就是说,对于线性可分的模式,我们可以用若干直线或平面将模式分类。比较成熟的线性分类算法有LMSE算法等,它们在本质上都属于几何分类法,当模式线性可分时,一般都能达到令人满意的效果。

感知器神经是网络中的一个概念,由美国计算机科学家罗森布拉特于1957年提出,是一种最简单的神经网络。在神经网络中,最基本的信息处理单元是神经元,一个神经元的模型如图1所示。其中每一个输入分量 p_j ($j=1,2,\dots,r$)通过一个权值分量 w_j 进行加权求和,并作为函数 $f(\cdot)$ 的输入,加入偏差使网络多了一个可调参数。函数 $f(\cdot)$ 叫做激活函

数,此函数的性质决定了网络的功能。

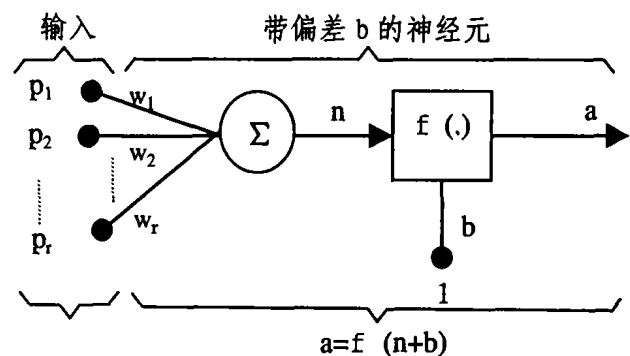


图1 神经元的模型

感知器作为一种简单的神经网络,它的激活函数 $f(\cdot)$ 为一个阈值函数,只有0和1两种状态,其表达式如式(1),函数的图形如图2所示。

$$a = f(n+b) = \begin{cases} 1 & n+b > 0 \\ 0 & n+b < 0 \\ 1 & n+b = 0 \end{cases} \quad (1)$$

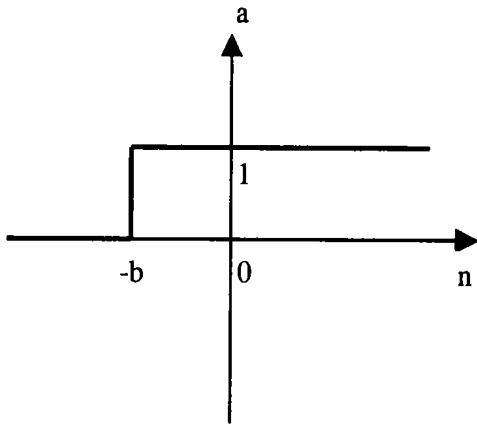


图2 感知器的激活函数的图形

考虑到 LMSE 算法并非是最简单和有效的线性分类算法,本文根据上面介绍的感知器可以很方便地把输入向量划分成两类的特点,提出了用单层感知器去进行线性模式划分的一种新算法,并阐述了其原理,最后用 MATLAB 去实现,并解决了两个不同类型的模式划分问题。

2 感知器分类算法描述

由感知器的激活函数可见,对任意一组输入 p ,感知器的输出 a 取值要么是1,要么是0。假设权矩阵用 W 来表示,输入用矩阵 P 来表示,偏差用 B 来表示,则感知器的输出 a 可用矩阵的形式表示如下:

$$a = (W * P + B) \quad (2)$$

感知器的输出 a 为0还是为1,根据式1可以看出,由 $W * P + B$ 的值决定。 $W * P + B = 0$ 成为了一个决定感知器的输出值 a 的边界函数,称为边界判定函数。当 $W * P + B < 0$ 时, $a = 0$; 而当 $W * P + B \geq 0$ 时, $a = 1$ 。当我们确定了网络的权值 W 和偏差 B 后,在由各输入矢量 $p_j (j = 1, 2, \dots, r)$ 为坐标轴组成的输入矢量空间中,可以画出 $W * P + B = 0$ 的轨迹。对于任意一组输入矢量 P , 要么落在 $W * P + B = 0$ 上, 要么落在 $W * P + B > 0$ 区域, 要么落在 $W * P + B < 0$ 区域, 只可能是这三种情况之一。所以我们可以适当选择参数 W 和 B , 使得对输入矢量能够达到期望的划分。当输入矢量为二维时, 边界判定函数 $W * P + B = 0$ 的轨迹 L 为一条直线, 其图形如图3所示。

通过对感知器的训练,可以不断调整网络的权值矩阵 W 和偏差矩阵 B , 从而实现对模式的分类。感知器的训练过程如下:

在输入矢量 P 的作用下,计算网络的实际输出 A , 并与相应的期望输出 T 进行比较, 检查 A 是否等于 T , 然后用比较后的误差 E , 根据学习规则进行权值和偏差的调整, 重新计算网络在新权值作用下的输入, 重复权值调整过程, 直到网络的输出等于期望的输出或训练次数达到事先设置的最大值时才结

束训练。

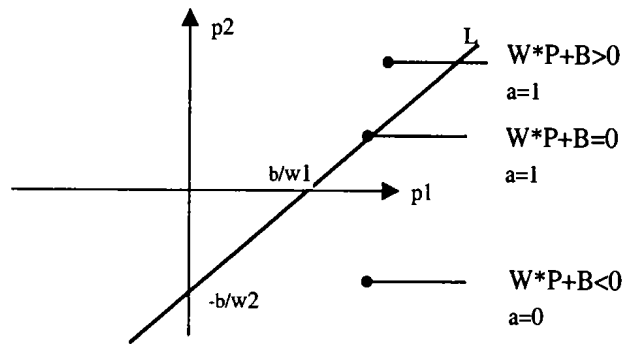


图3 二维输入时边界判定函数的轨迹

图4给出了训练算法的详细步骤。

1. 由 P 和 T 确定网络结构, 得出神经元数目 s
2. 初始化参数
 - W 初始化为 $(-1, 1)$ 之间的随机值
 - 初始化最大训练次数 \max_epoch
3. 由 P 及最新的 W , 计算网络输出 A
4. 判断 A 是否等于 T , 若相等, 则算法结束, 否则判断是否达到最大训练次数, 若达到, 则算法结束, 否则继续。
5. 根据学习规则调整 W 和 B , 并返回3

图4 训练算法详细步骤

3 算法实现及其应用

作为感知器分类算法的应用, 我们首先用 MATLAB 实现了其算法, 然后用其解决了两个线性可分的模式划分问题。下面将详细描述问题及其解决过程。

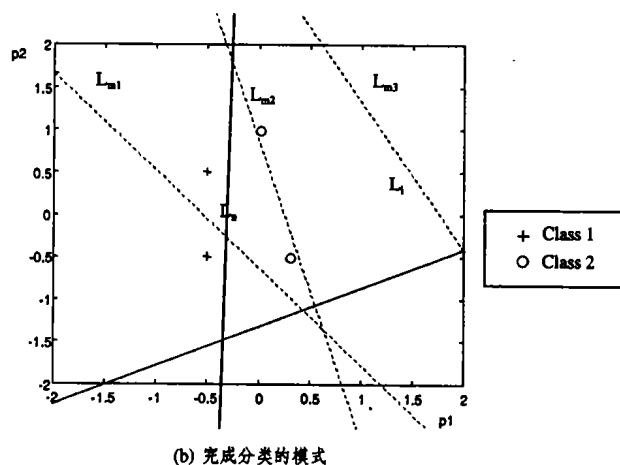
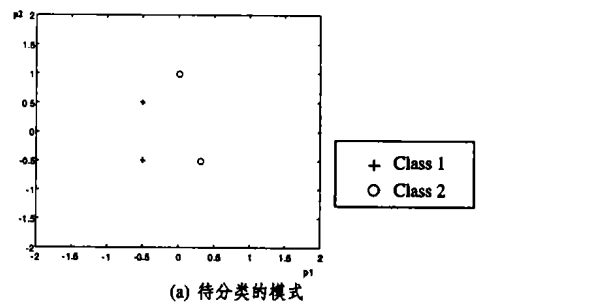


图5 两类模式的划分

注: L_1 代表初始时选择的分类直线; $L_{mi} (i = 1, 2, 3)$ 代表中间训练过程中产生的结果; L_m 代表最终结果

3.1 一个简单的分类问题

假设有一组输入矢量 $P = [-0.5 \quad -0.5 \quad 0.3 \quad 0; -0.5 \quad 0.5 \quad -0.5 \quad 1]$;

其对应的期望输出矢量 $T = [1 \quad 1 \quad 0 \quad 0]$; 要找出一条直线, 将两类模式划分开。

为了直观地看出这个问题所代表的意义, 将输入矢量 P 以及期望矢量 T 表示于图5, 其中用 '+' 代表1, 用 'o' 代表0, 可以看出总共有两类模式, 我们的目的是在图中找出一条直线, 将此两类模式划分开。

应用感知器分类算法, 可以得到图5(b)所示的结果。其中, 黑色的实线 L_n 最终结果, 红色的虚线 L_0 代表初始时选择的分类直线, 即 W 和 B 为随机数时的结果, 兰色的虚线 $L_{mi} (i=1, 2, 3)$ 代表中间训练过程中产生的结果。由图中可以看出, 一开始时, 由于权矩阵和偏差矩阵均为随机选取, 故所产生的结果很不好, 选择的直线不能正确地将两类分开。但是随着训练过程的继续, W 和 B 不断地得到调整, 所以结果向越来越好的方向发展, 直到经过有限次的训练, 产生了正确的结果, 得到的直线能正确地将两类划分开来。从图中也可以看出来, 用感知器分类算法来解决这一类简单的分类问题是相当有效的, 图中只经过了四次训练就收敛到了最好的结果。

3.2 多类情况

假设有一组输入矢量 $P = [0.1 \quad 0.7 \quad 0.8 \quad 0.8 \quad 1.0 \quad 0.3 \quad 0.0 \quad -0.3 \quad -0.5 \quad -1.5; 1.2 \quad 1.8 \quad 1.6 \quad 0.6 \quad 0.8 \quad 0.5 \quad 0.2 \quad 0.8 \quad -1.5 \quad -1.3]$;

其对应的期望输出矢量 $T = [1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$;

要找出若干直线, 将这些模式划分开。

通过观察 T 的元素, 可以发现 T 可以有四种组合, 即 00, 01, 10, 11, 也即模式总共有四类。为了直观地看出这个问题所代表的意义, 将输入矢量 P 以及期望矢量 T 表示于图6(a), 其中用不同的符号代表不同的类别, 可以看出总共有四类模式, 我们需要找出两根直线才能将此四类模式划分开。

应用感知器分类算法, 可以得到图6所示的结果。其中, 黑色的实线 L_{n1} 和 L_{n2} 代表最终的结果, 红色的虚线 L_{i1} 和 L_{i2} 代表刚开始时由随机选择的权矩阵和偏差矩阵得到的结果。问题2比问题1要复杂得多, 但是应用感知器分类算法还是很有效地得到了比较完美的结果。相对于问题1来说, 解决问题2所花费的训练次数要多一些, 但还是在20次以内就收敛到最好结果了。并且需要提供给算法的仍然只是 W 和 P , 所以对算法而言, 解决问题1和问题2并没有太大的差别。

结束语 由于感知器自身结构的限制, 其应用被限制在一定的范围。因为感知器的输出矢量只能

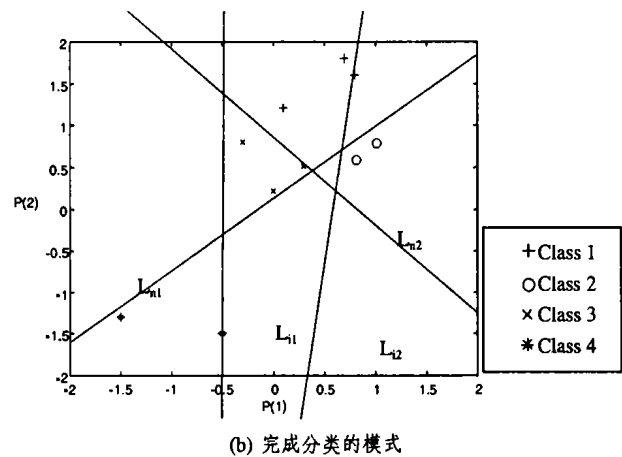
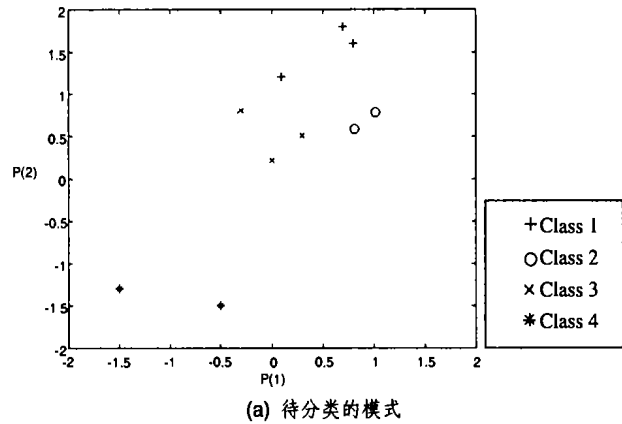


图6

注: L_{i1} 和 L_{i2} 代表初始时选择的分类直线; L_{n1} 和 L_{n2} 代表最终结果

取0和1, 所以只能解决简单的分类问题, 并且只能线性地将输入矢量进行分类。若输入矢量为线性不可分的, 则单层感知器将无能为力。比如, 单层感知器就解决不了简单的“异或”逻辑运算的模式划分。通常将采取多层网络结构, 并且可以采用一些更先进的算法去对网络进行训练, 使问题得到圆满的解决。比如可以采用遗传算法去训练网络的权值, 将权值和偏差作为待整定参数, 而适应度函数可以选为误差, 即网络实际输出的值和期望输出的值的差。这样就可以利用遗传算法的全局搜索能力, 在权值和偏差可能的取值空间去搜索使得误差最小的一组 W 和 P , 作为最终网络的参数。

参考文献

- 1 沈清, 汤霖. 模式识别导论. 国防科技大学出版社, 1998
- 2 从爽. 面向 MATLAB 工具箱的神经网络理论与应用. 中国科技大学出版社, 1998
- 3 陈国良, 王东生. 遗传算法及其应用. 人民邮电出版社, 1996
- 4 Duda R O, Hart P E, Stork D G. 李宏东, 姚天翔译. 模式分类. 机械工业出版社, 2003
- 5 Hagan M T, Demuth H B, Beale M H. 戴葵译. 神经网络设计. 机械工业出版社, 2002