

《神经网络导论》实验一

—Adaline的LMS算法

西安交通大学电子与信息工程学院信息与通信工程系

2012年11月

1 实验目的

1. 通过实验了解Adaline的工作原理;
2. 对比LMS的三种算法, 并通过上机实验掌握具体实现方法;
3. 与采用硬限幅函数的单个神经元模型进行对比, 比较其异同。

2 实验原理

我们已经知道, 采用硬限幅函数的单个神经元, 通过简单的学习算法, 可以成功实现两类线性可分类的分类功能。但对于大多数的非线性可分类来说, 则无法完成分类功能, 为此我们转而采用具有线性功能函数的神经元Adaline(Adaptive Linear Element)方法。

设输入矢量 $\mathbf{X} = [x_1, x_2 \dots x_N]$, 加权矢量 $\mathbf{W} = [w_1, w_2 \dots w_N]$, 则神经元的输出可通过下式来计算:

$$\begin{cases} I = \mathbf{W}\mathbf{X}^T = \mathbf{X}\mathbf{W}^T \\ y = f(I) = \mathbf{W}\mathbf{X}^T = \mathbf{X}\mathbf{W}^T \end{cases} \quad (1)$$

要实现Adaline的分类功能, 按照最小二乘法的统计意义而言, 就是要求所有样本的实际输出值与理想预期值之间误差的均方值最小。设输入观察矢量 \mathbf{X} 的期望输出是 d , 当权向量为 \mathbf{W} 时的实际输出是 y , 定义误差 $\varepsilon = d - y$ 。考虑所有可能出现样本的均方误差:

$$E[\varepsilon^2] = E[(d - y)^2] \quad (2)$$

将(1)式代入, 可得:

$$\begin{aligned} E[\varepsilon^2] &= E[d^2] + \mathbf{W}E[\mathbf{X}^T\mathbf{X}]\mathbf{W}^T - 2E[d\mathbf{X}]\mathbf{W}^T \\ &= E[d^2] + \mathbf{WRW}^T - 2\mathbf{PW}^T \end{aligned} \quad (3)$$

其中, $\mathbf{R} \triangleq E[\mathbf{X}^T\mathbf{X}]$ 是输入向量的自相关矩阵, $\mathbf{P} \triangleq E[d\mathbf{X}]$ 是输入向量与期望输出的互相关向量。由(3)式可知必定存在最佳的加权矢量 \mathbf{W}^* 使均方误差达到最小, 对(3)式求梯度可得:

$$\nabla_{\mathbf{W}} E[\varepsilon^2] = 2\mathbf{WR} - 2\mathbf{P} \quad (4)$$

由(4)式可解出最优权向量:

$$\mathbf{W}^* = \mathbf{PR}^{-1} \quad (5)$$

(5)式给出了求最佳加权矢量的方法, 但是需要做大量的统计计算, 而且当输入矢量 \mathbf{X} 的维数很大时, 需要解决高阶矩阵求逆的问题, 这些都是非常困难的。于是我们给出下面三种递推求解的方法。

2.1 LMS学习问题的严格递推学习算法

1. 任意设置初始加权矢量 $\mathbf{W}(0)$;
2. 对于每一个时序变量 k , 按下式调整权向量 \mathbf{W} :

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \alpha [-\nabla_{\mathbf{W}} E[\varepsilon^2(k)]] , \quad k = 1, 2, \dots \quad (6)$$

(6)式的含义为, 应该向梯度的负方向调整加权向量 $\mathbf{W}(k)$ 。只要选定合适的步幅系数 α 就能保证学习的收敛性。

求出(6)式中的梯度:

$$\nabla_{\mathbf{W}} = -2E[\varepsilon(k)\mathbf{X}(k)] \quad (7)$$

于是(6)式变为:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + 2\alpha E[\varepsilon(k)\mathbf{X}(k)] \quad (8)$$

用这种方法可以保证求得严格的最佳解, 而且避开了矩阵求逆的困难。但学习过程中的每一步仍需完成大量的统计计算, 统计计算的困难尚需解决。

2.2 LMS学习问题的随机逼近算法

将(8)式修正为如下形式:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + 2\alpha\varepsilon(k)\mathbf{X}(k) \quad (9)$$

即得到随机逼近算法。其与严格递推算法的区别在于: 用 $\varepsilon(k)\mathbf{X}(k)$ 代替 $E[\varepsilon(k)\mathbf{X}(k)]$, 由此避免了统计计算的困难, 但同时也给加权矢量的变化趋势带来了随机性。研究表明, 将步幅系数 α 变成一个随时序 k 变化的量, 当 $\alpha(k)$ 满足下列条件时, 学习一定是收敛的。

(1) $\alpha(k)$ 是时序 k 的非增函数;

(2) $\sum_{k=0}^{\infty} \alpha(k) = \infty, \quad \sum_{k=0}^{\infty} \alpha^2(k) < \infty$

2.3 LMS学习问题的基于统计的算法

这是一种具有一定统计特性的学习算法。

假设学习进行到第 k 步时, 可能出现的样本有 P 个, 分别用 $\mathbf{X}_p(k)$ 表示, 下标 $p = 1, 2 \dots P$ 表示第 k 步学习过程中可能出现的不同样本编号。第 p 个样本的理想输出和实际输出分别用 $d_p(k)$ 和 $y_p(k)$ 表示。我们定义“误差平方和” $J(k)$ 如下:

$$J(k) = \frac{1}{P} \sum_p \varepsilon_p^2(k) \quad (10)$$

其中 $\varepsilon_p(k) = d_p(k) - y_p(k)$, $J(k)$ 相对于 \mathbf{W} 的梯度为:

$$\nabla_{\mathbf{W}} J(k) = -\frac{2}{P} \sum_p \varepsilon_p(k) \mathbf{X}_p(k) \quad (11)$$

令误差朝 $J(k)$ 减小的方向调整, 可得如下递推算法:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{2\alpha}{P} \sum_p \varepsilon_p(k) \mathbf{X}_p(k) \quad (12)$$

当 P 的数量非常大, 使上式右端的求和项足以代表输入的统计特性时, (12)式与(8)式(严格递推算法)一致。当 P 取1时, (12)式与(9)式(随机逼近求解算法)一致。

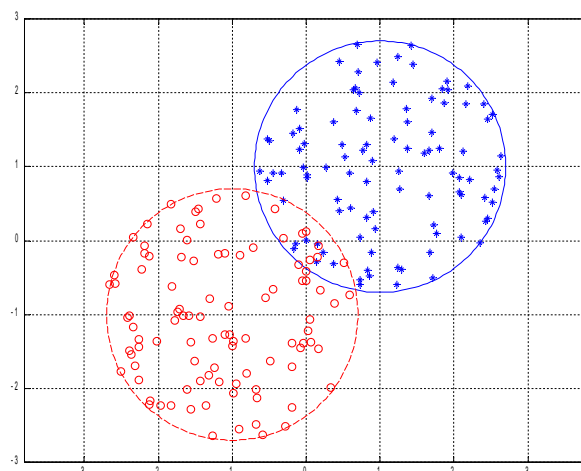


图 1: 非线性可分类

3 实验内容及步骤

1. 阅读Adaline的原理，理解三种LMS算法的思想。并用随机逼近算法和最陡下降算法实现图一所示的分类问题。
2. 样本矢量。输入样本矢量在图1所示的两个圆内均匀分布，圆心分别位于点 $(1, 1)$ 和 $(-1, -1)$ ，半径为1.7。令位于右上方圆内的样本理想输出为1，左下方圆内的样本理想输出为-1。用MATLAB的load命令加载数据文件lms_samp.mat，其中的变量samp是一个 200×3 的矩阵，其第1列为样本的横坐标，第2列为纵坐标，第3列为该样本的理想输出。
3. LMS算法。计算自相关阵 \mathbf{R} 和互相关向量 \mathbf{P} ；在此基础上计算最佳权向量 \mathbf{W}^* 、最小均方误差 $E[\varepsilon^2]_{min}$ （下面用 E_{min} 表示）。（注意：采用具有2个输入的单神经元，并考虑神经元的阈值。）
4. 随机逼近算法。选择步幅系数 $\alpha = 0.01$ 、最小误差 $E_{min} + 0.001$ ，按公式(9)编程实现随机逼近算法，记录学习收敛后得到的权向量；绘出每步学习后的均方误差 $E[\varepsilon^2(k)]$ 曲线。（收敛条件为所有样本的均方误差小于所选的最小误差。）
5. 固定最小误差 $E_{min} + 0.001$ ，分别选择步幅系数 $\alpha = 0.002$ 、0.008、0.02、0.1、0.3，重复第4步。观察步幅系数对收敛速度有什么影响，

并解释所出现的现象。

6. 基于统计的算法。选取步幅系数 $\alpha = 0.02$ 、最小误差 $E_{min} + 0.001$ 、每步随机选取 $P = 5$ 个样本。按公式(12)编程实现最陡下降算法，记录学习收敛后得到的权向量，绘出均方误差 $E[\varepsilon^2(k)]$ 随学习步数的变化曲线。
7. 固定最小误差 $E_{min} + 0.001$ ，步幅系数 $\alpha = 0.02$ ，分别选择 $P = 2, 50$ 。重复步骤6。观察均方误差 $E[\varepsilon^2]$ 的收敛性、收敛速度与每步样本数 P 之间的关系，并对结果进行简要分析。
8. 检验。用load命令加载数据文件lms_tstsamp.mat，其中的矩阵变量tstsamp包括了200个用于测试的样本向量。使用这些测试样本分别对第4步和第6步学习得到的权向量进行分类检验，计算分类正确率。
9. (选做) Widrow 严格递推算法。设所给出的样本即为所有可能的样本，固定最小误差 $E_{min} + 0.001$ ，步幅系数 $\alpha = 0.02$ ，按公式(6)编程实现Widrow的严格递推算法。分别选择步幅系数 $\alpha = 0.05, 0.1, 0.35$ ，记录每步学习后的均方误差曲线。分析所得到的结果。

4 实验思考题

1. 如果想用采用硬限幅函数的单个神经元完成该分类任务，会出现什么样的现象？
2. 通过观察比较随机逼近算法与不同 P 时最陡下降算法的均方误差曲线随时序变量 k 的变化有什么不同，并简要解释原因。

5 实验报告要求

1. 简述实验目的及原理；
2. 按实验步骤附主要实验结果，并进行实验结果分析；
3. 总结实验的主要结论，针对实验中不完善之处提出改进建议；
4. 回答实验思考题；
5. 附软件的主要模块（即每种算法的主体部分）及其说明（包括主要变量及功能模块）。