第二章　感知机(perceptron)

2.1
感知机模型

- 定义2.1：感知机模型
  - 输入空间：$\mathcal{X} \subseteq R^n$
  - 输出空间：$\mathcal{Y} = \{+1, -1\}$
  - 输入特征向量： $x \in \mathcal{X}$
    - 实例的特征向量
  - 输出值：$y \in \mathcal{Y}$
    - 实例类别
  - 输入空间 -> 输出空间：

$$f(x) = sign(w*x + b) \tag{2.1}$$

    其中（2.1）：
    - $w \in R^n$为权值（weight）
    - $b \in R$为偏置（bias）
    - **sign**为符号函数：

$$sign(x) = \begin{cases} +1, x \geq 0 \\ -1, x < 0 \end{cases} \tag{2.2}$$

- 感知机是一种线性分类模型，属于*判别模型*。
  - 其原理近似于二维平面内的一次线性规划问题($l: y = kx + b$) 。
  - 映射到空间中，直线 l 映射为超平面 S ，直线斜率 k 映射为超平面的法向量 w ，直线截距 b 映射为超平面截距 b 。
  - 这个超平面将特征空间分为两部分，具有同一属性的不同特征的样本集合（特征空间）被分为两个部分。因此，超平面S也称为分离超平面。
- 注意：
  - 在平面上的一次线性分类中，可被直线分为两部分的点集中的任意一点（x_n, y_n)均满足：

$$y_n > k * x_n + b || y_n < k * x_n + b$$

    而在感知机公式中，我们设 $y_n = sign(p_n)$不考虑分类结果到分类标识的转化（**sign**），其满足的条件为：

$$p_n > 0 || p_n < 0$$

即：

$$w * x + b > 0 || w * x + b < 0$$

如果将其简单的与线性规划作对比，我们不难发现，他们之间还是存在一些差别的：为什么感知机的判断标准是与 0 比较呢？难道所有的点在二维平面上的函数值都为 0 么？又该如何解释并统一这种差别呢？

- 个人认为，平面上的直线完成的简单线性划分的判断标准不是0而是其他整数，是因为其空间的特征向量是经过旋转的，我们用固定的x轴和y轴来度量所有点的分类；而当上升到超平面与n维特征空间时，我们的度量标准就仅以我们当前的分类方式产生的特征向量为标准。
- 更进一步，我们也可以由此出发，将平面上的线性规划问题转化为与感知机相类似的形式：只需将平面旋转至x轴正向与分类直线的法向量平行，再将坐标系平移至x轴与分类直线重合（或者将平面上所有点都投影到一条与分类直线垂直的直线上），即可得到相同结果（不便插图，请读者自行尝试）。此时的“x轴”即为分类直线，“y轴”即为点的特征取值范围，与分类直线垂直。
- 再回到感知机的公式中，对于 n 维特征空间 N 和 n-1维超平面 α ，我们不便理解旋转这一概念。可以变通一下，将空间 N 中的所有点投影到一个与超平面 α 正交的超平面 β 上，类比上一点的方案2，即可得到共性。

2.2 感知机学习策略

2.2.1　数据集的线性可分性：

- 定义2.2：数据集的线性可分性：
  - 对于数据集

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

其中$x_i \in \mathcal{X}, y_i \in \mathcal{Y} = \{+1, -1\}, i = 1, 2, ..., N$

  - 若存在一个超平面S:

$$w * x + b = 0$$

能够将数据集的正实例点和负实例点完全正确地划分到超平面的两侧，则称数据集 Ç 是**线性可分数据集**，否则称数据集 Ç **线性不可分**。

- 通俗来讲，就是能否一刀将一张饼上的葡萄干全部分开。

2.2.2　感知机学习策略

- 感知机学习目标：
  - 求得一个能够将训练集正实例点和负实例点完全正确分开的分离超平面。
- 学习方式：
  - 定义损失函数，将损失函数极小化

- 损失函数：
  对于输入空间中任意一点 x_0 到超平面 S 的距离为：

$$\frac{1}{||w||}|w*x_0+b|$$

  其中$||w||$是w的L_2范数。（此公式类似于二维平面上点到直线的距离公式）
  对于**误分类**的数据(x_i,y_i)来说，由**sign**可知，

$$-y_i(w*x_i+b)>0$$

  恒成立，故有：

$$|w*x_0+b|=-y_i(w*x_i+b)$$

  得到每个误分类的点到超平面S的距离为：

$$-\frac{1}{||w||}y_i(w*x_i+b)$$

  不考虑$\frac{1}{||w||}$，就得到感知机学习的损失函数：

$$L(w,b)=-\sum_{x_i\in M}y_i(w*x_i+b)$$

  其中M为误分类点的集合，这一函数就是感知机学习的*经验风险函数*。
  - 注：
    - 对于被正确分类的点，$w*x_i+b>0$，$y_i=+1$，但由于参与计算经验风险函数的点都是误分类点，故$w*x_i+b>0$时，$y_i=-1$。
    - 损失函数非负，且误分类点越少，损失函数的值越小。
    - 损失函数对于w，b连续可导

2.3　感知机学习算法

2.3.1　感知机学习算法的原始形式

- 感知机学习算法是求损失函数极小化问题的解的算法
- 具体方法：梯度下降法：
  - 输出: w,b , 感知机模型$f(x)=sign(w*x_b)$
    - 选取初值w_0,b_0
    - 在训练集中选取任意一点(x_i,y_i)
    - 如果$y_i(w*x_i+b)\leq 0$

$$w\leftarrow w+\eta y_i x_i$$

$$b\leftarrow b+\eta y_i$$

其中η是步长,范围$(0,1]$,又称学习率。

- 注：此处的公式及赋值方法，在f（x）中分别对w和b求导就可以得到。 η 和 y_i 分别表示步长和方向。

2.3.2　算法的收敛性

- 所谓收敛，就是在有限步可以得到正确的解。在感知机中，正确的解就是能将数据集完全正确划分的超平面。
- 做一个变换，将矩阵 b 并入矩阵w，得到$\hat{w}=(w^T,b)^T$,同时，为了使运算不便，将x扩充为$\hat{x}=(x^T,1)^T$,这样，公式变为：

$$f(x)=sigh(\hat{w}*\hat{x})$$

定理2.1

设数据集T是线性可分的，其中$x_i\in\mathcal{X}=R^n,y_i\in\mathcal{Y}=\{-1,+1\}$，$i=1,2,...,N$,则

（1）存在$||\hat{w}_{opt}||=1$的超平面将数据集全部分开，且存在$\gamma>0$,对所有$i=1,2,...,N$

$$y_i(\hat{w}_{opt}*\hat{x}_i)=y_i(w_{opt}*x+b_{opt})\geq\gamma \tag{2.8}$$

（2）令$R=max_{1\leq i\leq N}||\hat{x}_i||$,则感知机2.1的误分类次数k满足不等式：

$$k\leq(\frac{R}{\gamma})^2 \tag{2.9}$$

- 注：
  - $\gamma_{max}=min_i\{y_i(w_{opt}*x_i+b_{opt})\}$
  - (2.9）说明误分类的次数是有限的，有限次搜索课找到正确解
  - 同一组数据，可能有多个正确解，这取决于误分类点的选择顺序和初值的选择。
  - 训练集线性不可分时，感知机学习算法不收敛，迭代结果震荡。

2.3.3 感知机算法的对偶形式

- 将 w 和 b 表示成实例 x_i 和 y_i 的线性组合的形式，通过求解系数求w和b。
- 对于算法2.1中，对于一个错分类点，假设其对 w 和 b 修改了 n_i 次，则增量为 $n_i\eta y_ix_i$
- 假设 w 和 b 关于 (x_i,y_i) 的增量分别为

$$\Delta w=\alpha_iy_ix_i=n_i\eta y_ix_i$$

和

$$\Delta b=\alpha_iy_i=n_i\eta y_i$$

这里 $\alpha=n_i\eta$,当 $\eta=1$ 时，α_i 即为因为误分而更新的次数。

具体算法过程：

- 对于 b 的计算与算法2.1相同
- 将参数 w 变为 α_i , 对其求导可得新递推式：

$$\alpha_i\leftarrow\alpha_i+\eta$$

由于 α 和 b 均为变化量，故不是一般性，可将 α 和 b 初值均设为0，此时二者皆为直接系数。

- 代入感知机模型，得：

$$f(x)=sign(w*x+b)=sign(\sum_{j=1}^{N}n_j\eta y_jx_j*x+\sum_{j=1}^{N}n_j\eta y_j) \tag{2.10}$$

注：

- 实例点更新次数越多，意味着它距离分离超平面越近，也就越难正确分类，这样的实例对学习结果影响越大。
    - 这点开始时让笔者十分费解：**不是应该离超平面越远的点对的超平面形状影响越大么?**
    - 的确，但是这里对学习结果的影响并非对超平面形状的影响，而是考虑超平面对实例点的敏感程度。离超平面越近的点，超平面稍微移动，这个点就从正变负或相反。离超平面越近，超平面为适应它而作出的调整也就越频繁，越多。
    - 查阅资料后，这个点很可能是支持向量。
- 实际上，这种对α的学习，也就是对n_i的学习，代表某一实例对超平面的影响程度，实际代码中经常不做区分。
- Gram矩阵：
    - 由（2.10）可知，样本点的特征向量以内积行事存在于感知机的训练算法中；故可以提前计算好所有实例的内积，也就是Gram矩阵，可以大大提高计算速度。

$$G = [x_i * x_j]_{N*N}$$

习题：

2.1　验证感知机为什么不能表示异或。

- 对于除异或的其他运算（以“与”举例）：

	0	**1**
0	0	0
1	0	1

其真值表为线性可分的；

- 而对于异或：

	0	**1**
0	1	0
1	0	1

无法用一条直线将其按照0、1分类，故异或线性不可分，感知机无法将其表示。

2.3　证明以下定理：样本集线性可分的充分必要条件是正实例点集所构成的凸壳与负实例点集所构成的凸壳互不相交

- 凸壳：

- 设集合 $S \subset R^n$ 是由R^n中的 k 个点所组成的集合，即$S = \{x_1, x_2, x_3, ..., x_k\}$. 定义 S 的凸壳 $conv(S)$ 为：

$$conv(S) = \{x = \sum_{i=1}^{k} \lambda_i x_i | \sum_{i=1}^{k} \lambda_i = 1, \lambda_i \geq 0, i = 1, 2, ..., k\}$$

- 证明：
  - 充分性：
    - 反证法： 假设$conv(A) \bigcap conv(B) \neq \emptyset$,且S可被超平面分割，设

$$P = conv(A) \bigcap conv(B)$$

      对于$\forall x \in P$，有：

$$x \in conv(A) \&\& x \in conv(B)$$

      又线性可分定义知，若S为线性可分，则$\forall x \in S, (x \in A) \& (x \in B) = \emptyset$,
      故产生矛盾。
      故结论成立。
  - 必要性：
    - 若样本集S线性可分，设正样例集为A，负样例集为B，则有

$$A \bigcap B = \emptyset$$

  则有

$$\forall x \in S, (x \in A) \& (x \in B) = \emptyset$$

  故交集不存在。