

感知机

Li Liang*

1 感知机

感知机由两层神经元组成，输入层接受外界输入传递给输出层，输出层是M-P 神经元，也称“阈值逻辑单元”。感知机模型是寻找线性超平面，将线性可分的样本点正确的分成两类。

记超平面为 $w\mathbf{X} + b = 0$ ，则分类结果为：

$$y = \text{sign}(w\mathbf{X} + b) = \begin{cases} 1 & w\mathbf{X} + b > 0 \\ -1 & w\mathbf{X} + b \leq 0 \end{cases} \quad (1)$$

若存在某个点 $\{x_i, y_i\}$ 的分类结果错误，则 $y_i(wx_i + b) < 0$ ，据此可以判断点是否误分类。我们采用误分类点到超平面的距离来衡量分类失败的程度，希望误分类点到超平面的距离越小越好。假设所有的误分类点集合为 M ，则损失函数为：

$$L(w, b) = \sum_{x_i \in M} \frac{-y_i(wx_i + b)}{\|w\|} \quad (2)$$

不考虑 $\|w\|$ ，则损失函数为：

$$L(w, b) = \sum_{x_i \in M} -y_i(wx_i + b) \quad (3)$$

不考虑 $\|w\|$ ，相当于对超平面方程进行了规范化，可以理解为最终解算的 w 实际上是 $\frac{w}{\|w\|}$ 。

最终，问题变成最小化损失函数问题。

由于损失函数里面有限定，只有误分类的 M 集合里面的样本才能参与损失函数的优化，所以不能用最普通的批量梯度下降。感知机采用的是随机梯度下降法，每次仅采用一个误分类点来更新梯度。

*<https://github.com/leeliang/>

损失函数的梯度为：

$$\begin{aligned}\nabla_w L(\mathbf{w}, b) &= - \sum_{x_i \in M} y_i x_i \\ \nabla_b L(w, b) &= - \sum_{x_i \in M} y_i\end{aligned}\tag{4}$$

由于采用随机梯度法，采用的迭代公式为：

$$\begin{aligned}w &\leftarrow w + \eta y_i x_i \\ b &\leftarrow b + \eta y_i\end{aligned}\tag{5}$$

上式与西瓜书有一定差别，原因在于 *sign* 函数的定义不一致。