Perceptron

Mia Feng

2018 年 3 月 5 日

1 概述

Perceptron：监督学习，确定性模型。二分类，线性模型。模型虽然简单，但是由 Perceptron 可以很好的过度到理解神经网络 [?]。学习步长的设置是决定结果很重要的因素，神经网络 fine tuning 时也会先调整学习步长。

求解目标：分类超平面，以二维空间为例，求解一条直线

$$y = b + \theta_1 x_1 + \theta_2 x_2 = \theta^{\mathbf{T}} x + b \tag{1}$$

求解思路：最小化误分类点数（由其可以扩展到 SVM 最大化最小几何距离）。Concretely, 通过最小化代价函数 [1]：

$$L(\theta, b) = -\sum_{x_i \in M} y_i(\theta \cdot x_i + b) \tag{2}$$

求解方法：最小化误分类点数，stochastic gradient descent

1.1 推导

误分类点数 当分类结果正确时，类标乘以分类平面 $(\theta \cdot x_i + b)$ 所得符号为正，反之所得符号为负。所以，对类标乘以分类平面值的结果取负，记为误分类的点数目。目标要对其求极小

$$\arg\min_{\theta, b} L(\theta, b) = -\arg\min_{\theta, b} \sum_{x_i \in M} y_i(\theta \cdot x_i + b) \tag{3}$$

Stochastic gradient descent 每次随机选取一个误分类点使其梯度下降，因为对所有样本点的梯度下降是耗时的（这种叫做 Batch Gradient

Descent）。神经网络采用 SGD 与 Batch Gradient Desecent 的中间版，或者这种中间版的改进版。为了降低运算量，采用 mini-batch gradient descent（对数据分批次求极小）。算法在期望上收敛，所以不一定是全局最优。即使代价函数是强凸且光滑，收敛速度也只有 $O\left(\frac{1}{T}\right)$，where T 是样本点数。运用随机梯度下降后，每当一个实例点被误分，即位于超平面错误的一侧时，即调整 θ, b 使得分类平面向该误分类点的一侧移动，以减少该误分类点与超平面的距离，甚至超平面越过该误分类点使其被正确分类 [1]。

$$\begin{aligned} \nabla_\theta L\big(\theta, b\big) &= -\sum_{x_i \in M} y_i x_i \\ \nabla_b L\big(\theta, b\big) &= -\sum_{x_i \in M} y_i \end{aligned} \tag{4}$$

2 算法实现

见 [1]

1. 初始化 θ, b
2. 迭代直至训练集中无误分类点 {
   - a. 从样本集中随机选取数据点 $\big(x_i, y_i\big)$
   - b. 如果 $y_i\big(\theta \cdot x_i + b\big) \le 0$

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

}

3 Implementation

分类测试：数据在 flowers.csv

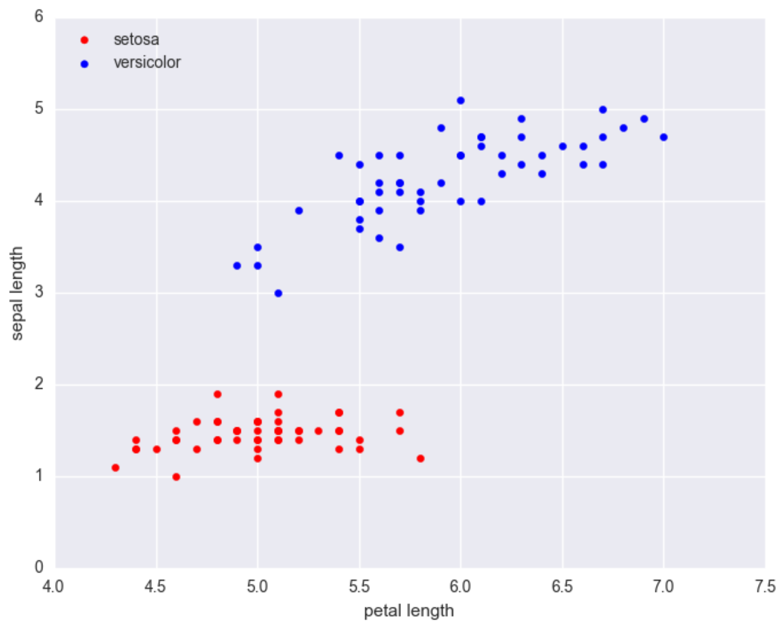
setosa
versicolor
sepal length
petal length
0
1
2
3
4
5
6
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5

图 1: 训练数据

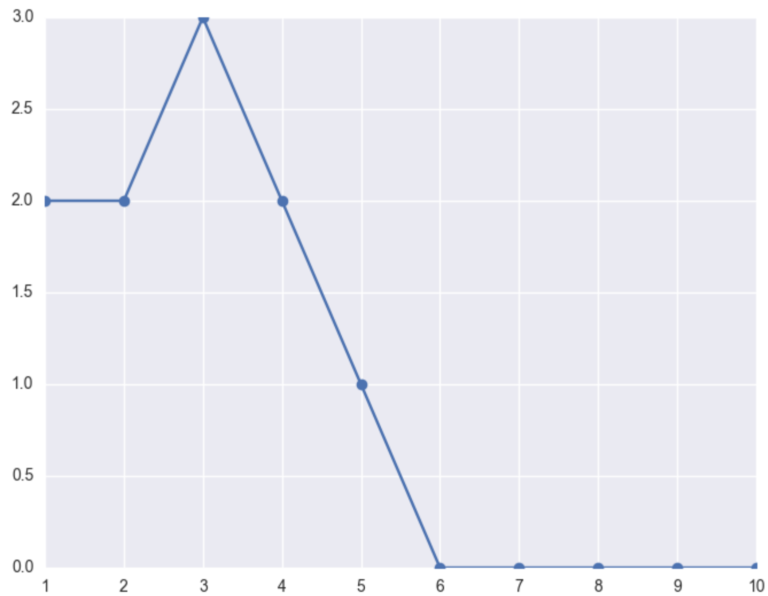

图 2: 每次迭代被错分的数目

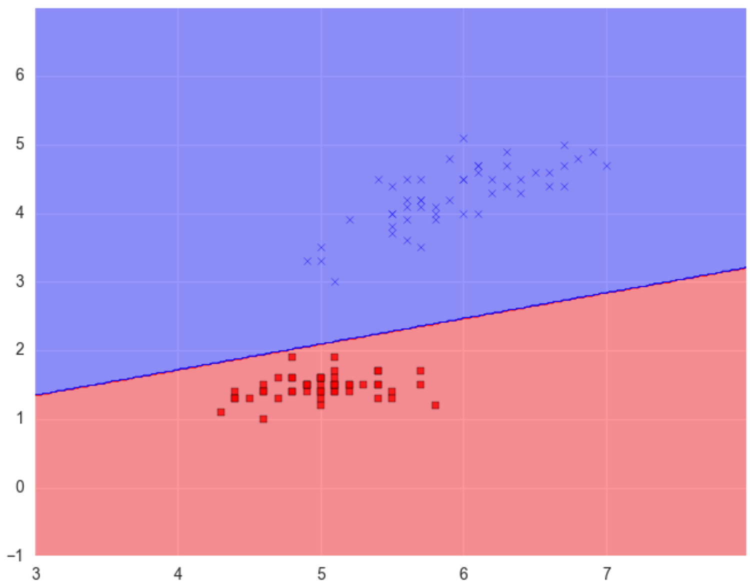

图 3: Perceptron 分类平面

参考文献

[1] 李航. 统计学习方法. 清华大学出版社, 2012.