

机器视觉实验室

面向 2022 级同学招新考察

一、请查阅相关资料，将文献翻译成英文。

- 翻译意思准确即可，不必逐字逐句一一对应翻译
- 公式请用公式编辑器输入（公式编辑器使用方法可以百度）

使用梯度下降法，理论上可以完成逻辑回归的训练过程，但在实际使用中还会出现一个问题。从式 (3.36) 和式 (3.37) 可以看出，无论更新哪个参数都要对所有 N 个数据求和。 N 较小还好，如果 N 非常大，数据就无法一次性放入内存中，计算时间将变得非常久。

可以解决这个问题的方法就是随机梯度下降法 (stochastic gradient descent)。与梯度下降法先计算全部数据的和再去更新参数的做法不同，随机梯度下降法是每次随机选择一个数据去更新参数，即用以下表达式对 N 个数据进行计算。

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \eta(t_n - y_n)\mathbf{x}_n \quad (3.38)$$

$$b^{(k+1)} = b^{(k)} + \eta(t_n - y_n) \quad (3.39)$$

名称中之所以有“随机”二字，是因为它选择数据的顺序是随机的。使用随机梯度下降法，就能以梯度下降法更新一次参数的计算量完成 N 次参数更新，从而高效地找到最优解。不过 N 次学习之后就能让梯度收敛为 0 的情况很少，需要对 N 个数据进行反复训练。反复训练全部数据的过程称为迭代 (epoch)，每次迭代都会先把数据顺序打乱再进行训练，这样训练结果就更不容易出现偏颇，容易得到更优的解。为了方便理解，请看以下以近似 Python 语法编写的随机梯度下降法的伪代码。

二、请谈一谈你对参与竞赛、科研项目的一些想法。进了创新工作室后，有什么样的学习计划？