

Weekly Report (3)

---By Chen Junlin at UESTC

chenjunlin@std.uestc.edu.cn

Week: 2019-2020-1学期 第9周

Time: 2019/10/28 ~ 2019/11/3

本周完成内容

1. 完成了 **CS231n Assignment 1 第 1 部分**。实现了基于 K-Nearest Neighbor 方法的分类器，并采用向量化（Vectorization）方法对分类器的训练进行了加速，针对不同的k的取值对分类器的表现进行评估

(Codes: [knn.ipynb](#)、[k_nearest_neighbor.py](#))

2. 加深了对于 NumPy 一些常用函数的认识（笔记见[Notes](#)），对于Python编程的熟练程度进一步加强。
3. CS231n 课程视频看到 **Lecture 3**（about SVM, softmax and gradient descent）。SVM是较新的知识，而softmax和梯度下降法在之前已有了了解。

困难和挑战

在对 KNN 方法进行向量化实现时，发现需要理清各个矩阵之间的关系较为困难。需要时时刻刻将各个矩阵变量的维度及每个维度的含义记在心中，或者单独列出来放在旁边，否则编程将会一团乱麻。另外，将计算表达式进行向量化的转化需要一定的数学推导，尤其是线性代数中对于矩阵的操作，并且要对Numpy中常见的对矩阵的操作相当熟悉。

例如，对于 L2 距离，

$$dist(i, j) = \sqrt{\sum_k (x_{test}(i, k) - x_{train}(j, k))^2}$$

其中， x_{test} 和 x_{train} 均为 $n \times D$ 维矩阵，对矩阵中相应的行，进行逐像素计算、求和。为了向量化，需进行以下变形：

$$dist(i, j) = \sqrt{\sum_k (x_{test}(i, k))^2 + \sum_k (x_{train}(j, k))^2 - \sum_k 2x_{test}(i, k)x_{train}(j, k)}$$

根据矩阵乘法公式：

$$X \cdot Y = \sum_k X(i, k) \cdot Y(k, j)$$

所以：

$$\sum_k x_{test}(i, k)x_{train}(j, k) = X_{test} \cdot X_{train}^T$$

而另外两项则直接使用逐元素平方、调用 `numpy.sum()` 对 `axis=1` 进行求和。最后，调用广播（Broadcast）机制，实现不同维数矩阵之间的相加减。

p.s. Debug 真心难受啊😓

下周计划

- 由于期中考试临近，计划主要针对已看完的 Lecture 3，以 Assignment 为主，实现相应方法，对不熟悉的知识结合官方 Notes 进行细化，对不熟悉的 Python 和 Numpy 语法进行收集整理。
- 不着急开展新的内容。完成 Assignment 后如果有时间再开展新内容。