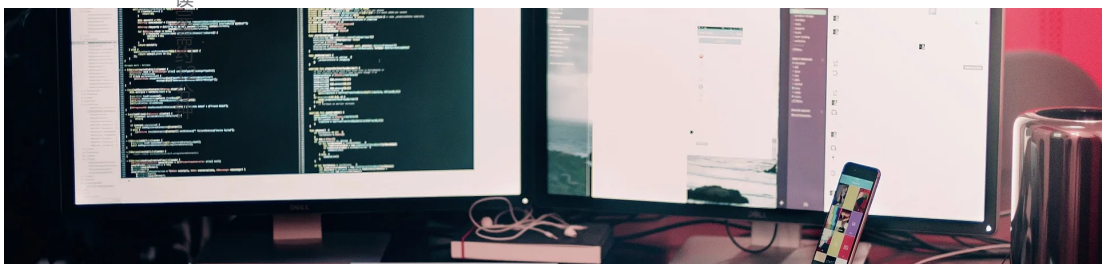
处理数据，大数据甚至更大数据的 17 种策略

作者：Jeff Hale 译者：刘志勇 策划：刘燕

2020-09-10

本文字数：3517 字 阅读



> 如何处理大数据和真正的大数据？

本文最初发表在 Towards Data Science 博客，经原作者 Jeff Hale 授权，InfoQ 中文站翻译并分享。

处理大数据可能很棘手。不会有人喜欢“内存不足”的错误提示。也不会有人愿意等待代码运行。更别说会有人乐意抛弃 Python。

如果你遇到这些问题，请不要绝望！我将在本文中为你提供一些技巧，并介绍一些即将问世的库，帮助你高效地处理大数据。我还会为你提供解决方案，解决那些无法适合内存的代码问题。而这一切都将会在 Python 中进行。

Python是最流行科学和数值计算编程语言。对于清理代码和探索性数据分析来说，Pandas是最受欢迎的。

在 Python 中使用 Pandas，可以让你处理比 Microsoft Excel 或 Google Sheets 更多的数据。

SQL 数据库在存储数据方面非常流行，但是 Python 生态系统在表达性、测试性、可再现性以及快速执行数据分析、统计和机器学习的能力方面，比 SQL 有很多优势。

不幸的是，如果你是在本地工作的话，那么，Pandas 所能处理的数据量将会受到你机器上的物理内存容量限制。但如果你在云端中工作，更多的内存需求又意味着需要耗费更多的资金。

无论你的代码在哪里运行，你都希望操作能够快速进行，这样你就可以完成任务！

总是要做的事情

如果你听说过或看到过关于加速代码的建议，那么你一定看到过警告：**不要过早优化！**

诚然 ，这是一条好建议。但懂得一些技巧也是很聪明的做法，这样你才能写出干净、快速的代码。

对于任何规模的数据集来说，下面三种做法都是良好的编码实践。

刘燕

InfoQ高级技术编辑

最新发布

进入工业大生产阶段，能让 AI 真地的关窍是什么？

19 小时前

马斯克吐槽无用的“阑尾”终于特斯拉移除超声波传感器，All In动驾驶

20 小时前

水滴筹创始人：中国以外不推行节跳动 2021 年净亏 6041 亿，超千亿；；美对芯片实施新的出一周资讯

2022-10-09

推荐阅读

2018 年，20 大 Python 数据科了哪些更新？

Python，架构，深度学习，AI

把嵌套列表作为 Apache Spark选

2019-07-25

Spark 数据倾斜问题处理

2021-01-14

仅用几行代码，让 Python 函数倍

AI，语言 & 开发，Python，方法

我创建了自己的 YouTube 算法

建列表比加载列表的 append 属性并将其作为函数反复调用要快：这要感谢 Stack Overflow Answer（译注：Stack Overflow 网站的一款 App，Stack Overflow 是一个程序设计领域的问答网站）。然而，一般来说，不要为了速度而牺牲清晰度，所以在嵌套列表推导务必要小心。

. 在 Pandas，使用内置的函数向量化。其原理其实和字典推导的原因是一样的。一次将一个函数应用于整个数据结构比重复调用一个函数要快得多。

如果你发现自己正在申请，请想想你是否真的需要这样做。它将会遍历行或列。向量化方法通常速度更快，代码更少，因此它们是一个双赢的方法。

要避免使用其他在数据上循环的 Pandas Series 和 DataFrame 方法：`applymap`、`itterrows`、`ittertuples`。在 DataFrame 上使用`replace`方法，而不是任何其他选项，这样可以节省大量时间。

请注意，这些建议可能不适用于非常少量的数据，但在这种情况下，风险很低，所以谁在乎呢？

这就涉及到我们最重要的规则

如果可以，就继续用 Pandas。这是快乐的源泉。

如果你没有问题，也不希望数据膨胀，就不要担心这些问题。但在某些时候，你会遇到一个大数据集，然后你会想知道该怎么做。让我们来看一些技巧。

处理相当大的数据（大约数百万行）

. 如果你要做机器学习，就用你的数据子集来探索、清理，构建一个基线模型。快速解决 90% 的问题，节省时间和资源。这个技巧可以帮你节省很多时间。

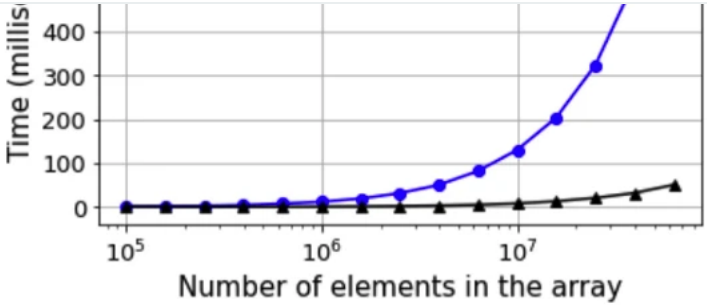
. 在读取 DataFrame 时，只用`usecols`参数加载你需要的列。记住，更少的数据输入意味着胜利！

. 有效地利用 dtype。将数值列向下转换为对`pandas.to_numeric()`有意义的最小 dtypes。将基数较低的列（只有几个值）转换为分类 dtype。请参阅这个关于高效 dtype 的🔗Pandas 指南。

. 在 Scikit-learn 将模型训练并行化，学习尽可能使用更多的处理核心。默认情况下，Scikit-learn 只使用 CPU 的一个核心。许多计算机的 CPU 有 4 个或更多的核心。在使用 GridSearchCV 和许多其他类进行交叉验证时，通过传递参数`n_jobs=-1`，可以将它们全部用于可并行化的任务。

. 将 Pandas DataFrame 保存为 feather 或 pickle 格式，以加快读写速度。感谢 Martin Skarzynski，因为他提供了证据和代码的🔗链接。

. 使用`pd.eval`加快 Pandas 的操作。将你常用的代码以字符串形式传递给函数。它的操作速度更快。下面有一个测试图表，DataFrame 为 100 列。



`df.query`与`pd.eval`基本相同，但它是作为 DataFrame 方法，而不是顶级 Pandas 函数。

由于存在一些问题，请注意查看文档。

Pandas 在幕后使用的是 Numexpr。Numexpr 还可以与 NumPy 一起工作。向 Chris Conlan 致敬，感谢他的著作《Fast Python》（《快速 Python》），正是这本书，我才知道了 Numexpr。Chris 这本书是一本学习如何加速 Python 代码的优秀读物。

处理真正的大数据（大约数千万行以上）

. 使用Numba。如果你在做数学计算，Numba 能给你带来极大的速度提升。安装 Numba 并导入它。然后，当你需要在 NumPy 数据上进行循环，且不能使用向量化方法时，就使用`@numba.jit`装饰器函数。它只对 NumPy 数据有效。在 Pandas DataFrame 上使用`.to_numpy()`将其转换为 NumPy 数组。
. 在有意义的时候使用SciPy 稀疏矩阵。Scikit-learn 通过一些转换器（如 CountVectorizer）自动输出稀疏数组。当数据主要为 0 或缺少值时，可以将列转换为 Panda 中的稀疏 dtype。要了解更多内容请点击此处。
. 使用 Dask 将数据集的读取并行化为 Pandas 的数据块。Dask 还可以跨多台机器执行并行化数据操作。它模仿了 Panda 和 NumPy API 的一个子集。Dask_ML是一个姊妹包，用于跨多台机器并行化机器学习算法。它模仿了 Scikit-learn API。与其他流行的机器学习库如 XGBoost、LightGBM、PyTorch 和 TensorFlow 很好地结合在一起。
. 不管有没有 GPU，都可以使用 PyTorch。正如我在这篇关于排序的文章中所发现的那样，在 GPU 上使用 PyTorch 可以大大提高速度。

未来处理大数据需要注意/尝试的事项

以下三个方案是截止 2020 年年中的前沿方案。预计将会出现配置问题和早期 API。如果你是在本地 CPU 上工作，这些方案不太可能满足你的需求。但它们看起来都很有前途，值得关注。

. 你能使用很多 CPU 核心吗？你的数据是否超过 32 列（到 2020 年年中开始是必需的）？然后考虑一下Modin。它模仿 Pandas 的一个子集，以加快对大型数据集的操作。它在幕后使用的是 Apache Arrow（通过 Ray）或 Dask。Dask 后端是实验性的。在我的测试中，有些事情并不是很快，例如，从 NumPy 数组读取数据就很慢，并且内存管理也是一个问题。
. 你可以使用 Jax 代替 NumPy。Jax 是 Google 开源的一款非常前沿的产品。它通过以下五个底层工具来加速操作：autograd、XLA、JIT、向量化器和并行化器。它可以在 CPU、GPU 或 TPU 上工作，并且可能比使用 PyTor

一个很好的提升。

其他关于代码速度和大数据的知识

计时操作

如果你想在 Jupyter Notebook 中进行计时操作，你可以使用`%time`或`%%timeit`魔法命令。它们都在单行或真个代码单元中工作。

```
%%timeit
a = 1 - 3
```

```
16.4 ns ± 0.334 ns per loop (mean ± std. dev. of 7
runs, 100000000 loops each)
```

`%time`运行一次，而`%%timeit`运行代码多次（默认值为 7）。一定要查看文档，了解其中的微妙之处。

如果你在脚本或 Notebook 中，可以导入时间模块，检查运行某些代码前后的时间，并找出差异。

```
from time import time

start = time()
a = 1 - 3
finish = time()
print(finish - start)
```

```
4.9114227294921875e-05
```

在进行测试时间时，请注意不同的机器和软件版本可能会导致差异。如果要重复测试的话，缓存有时候会产生误导。正如所有的实验一样，要尽可能保持一切不变。

存储大数据

GitHub 的最大文件大小为100MB。如果你想使用 GitHub 对大文件进行版本化，你可以使用Git Large File Storage（Git 大文件存储）扩展。

除非你愿意，否则请确保你没有将文件自动上传到 Dropbox、iCloud 或其他自动备份服务。

想了解更多吗？

Pandas 文档中，有关于提高性能和扩展到大型数据集的部分。本文其中一些想法就是根据这些章节改编而来。

总结

你已经了解了如何编写更快的代码，你也了解了如何处理大数据和真正的大数据。最后，你还了解到了一些新出的库，它们在处理大数据方面可能会继续变得越来越流行。

Jeff Hale，技术撰稿人，撰写数据科学相关的文章，如 Python、SQL、Docker 和其他技术主题。并维护数据科学资源邮件列表：https://dataawesome.com

原文链接：

https://towardsdatascience.com/17-strategies-for-dealing-with-data-big-data-and-even-bigger-data-283426c7d260

发布于：2020-09-10 02:12

刘燕 InfoQ高级技术编辑
发布了 923 篇内容，共 326.8 万次阅读，收获喜欢 1706 次。

大数据 AI 数据处理 最佳实践 方法论

评论

更多内容推荐

开始在 Amazon Web Services 上使用 R

本文将讨论 R 的基本知识和 AWS 上 R 的常见工作负载对。

文化 & 方法，语言 & 开发，亚马逊云科技

我是如何爱上 Julia 编程语言的？

为什么 Julia 很快成了我最喜爱的数据科学编程语言。

AI，语言 & 开发，编程语言，最佳实践，Python，C++

我们在工作中，除了和文字、表格打交道之外，还会经常涉及到批量处理图片和视频的工作需要。

2021-02-19

这 10 个 Python 技能，被低估了

一旦掌握这些技能，我敢说，你将会成为一个更“性感”的数据科学家。

AI，编程语言，Python，方法论

Python 太慢了吗？

“语言本身可能不是瓶颈，而是外部系统的限制。”

语言 & 开发，文化 & 方法，Python

自学 14 天后，我毁掉了自己的数据工程师面试

我是如何毁掉数据工程师面试的？

AI，新基建，AICon，最佳实践，方法论，机器学习，Apache，Python

Word Count：从零开始运行你的第一个 Spark 应用

纸上谈兵不可取，今天用一个小练习为你示范怎样解决统计词频的问题。

2019-05-28

深度学习为什么要选择 PyTorch

PyTorch 为深度学习程序员提供了很多帮助。

文化 & 方法，AI，AICon，数据库，Python，最佳实践，机器学习，方法论，数据处理

MXNet API 入门—第 2 篇

Apache MXNet是一种功能全面、可以灵活编程并且扩展能力超强的深度学习框架，支持包括卷积神经网络(CNN)与长短期记忆网...

语言 & 开发，架构

特征处理：如何利用 Spark 解决特征处理问题？

像动作、喜剧、爱情、科幻这些电影风格，是怎么转换成数值供推荐模型使用的呢？用户的行为历史又是怎么转换成数值特征的呢？

2020-10-11

为.NET 所用的 NumPy 和 SciPy

作为Python Tools for Visual Studio项目的一部分，NumPy和SciPy程序库已经迁移到.NET上了。这项迁移通过本地的C核心组合...

.NET，Python，语言 & 开发，架构

介绍 Gluon 这一易于使用的灵活深度学习编程接口

在 Gluon 中，您可以使用简单、清晰和简洁的代码定义神经网络。

其他，语言 & 开发，亚马逊云科技

实现更好编码的 30 个神奇的 Python 技巧

本文提供了 30 个 Python 技巧，可以将你的逻辑变成更优雅的代码。

AI，语言 & 开发，Python，方法论

Swift 与谷歌的可微编程项目

两年前，谷歌的一个小型团队开始致力于使Swift成为第一种具有一流语言集成可微编程能力的主流语言。

AI，机器学习，Swift，TensorFlow，Python

16 | 数据分析基础篇答疑

我总结了Numpy、Pandas、爬虫以及数据变换相关的问题，精选了大家比较疑惑的点作为解答。

2019-01-17

Koalas：让 pandas 轻松切换 Apache Spark，在大数据中规模应用

4月24日，Databricks在Spark + AI峰会上开源了一个新产品Koalas，它增强了PySpark的DataFrame API，使其与 pandas兼容。

AI，数据库，Spark，数据处理

Embedding 基础：所有人都在谈的 Embedding 技术到底是什么？

为什么我们总说Embedding在推荐系统领域非常重要？最经典的Embedding方法到底长啥样？

2020-10-13

Amazon SageMaker 现已推出：Deep Graph Library

今天，我们很高兴地宣布，为简化图神经网络的实现而构建的开源库 Deep Graph Library 现已在 Amazon SageMaker 上推出。

新基建，其他，亚马逊云科技

发现更多内容