从Pipenv到PyTorch，盘点2017年最受欢迎的十大机器学习Python库

2017年12月28日 ~ WISSENPRESS

1. Pipenv

项目地址：https://github.com/pypa/pipenv

2017 年排名第一的 python 库非 Pipenv 莫属。它在今年初发行，但却影响了每个 Python 开发者的工作流程，尤其是现在它成了用于管理依赖项的官方推荐工具。

Pipenv 源自大牛 Kenneth Reitz 的一个周末项目，旨在把其他软件包管理器的想法整合进 Python。安装 virtualenv 和 virtualenvwrapper，管理 requirements.txt 文件，确保依赖项的依赖项版本的可复现性，以上这些统统不需要。借助 Pipenv，你可以在 Pipfile（通常使用添加、删除或更新依赖项的命令构建它）中指定所有你的依赖项。Pipenv 可以生成一个 Pipfile.lock 文件，使得你的构建成为决定性的，避免了寻找 bug 的困难，因为甚至你也不认为需要一些模糊的依赖项。

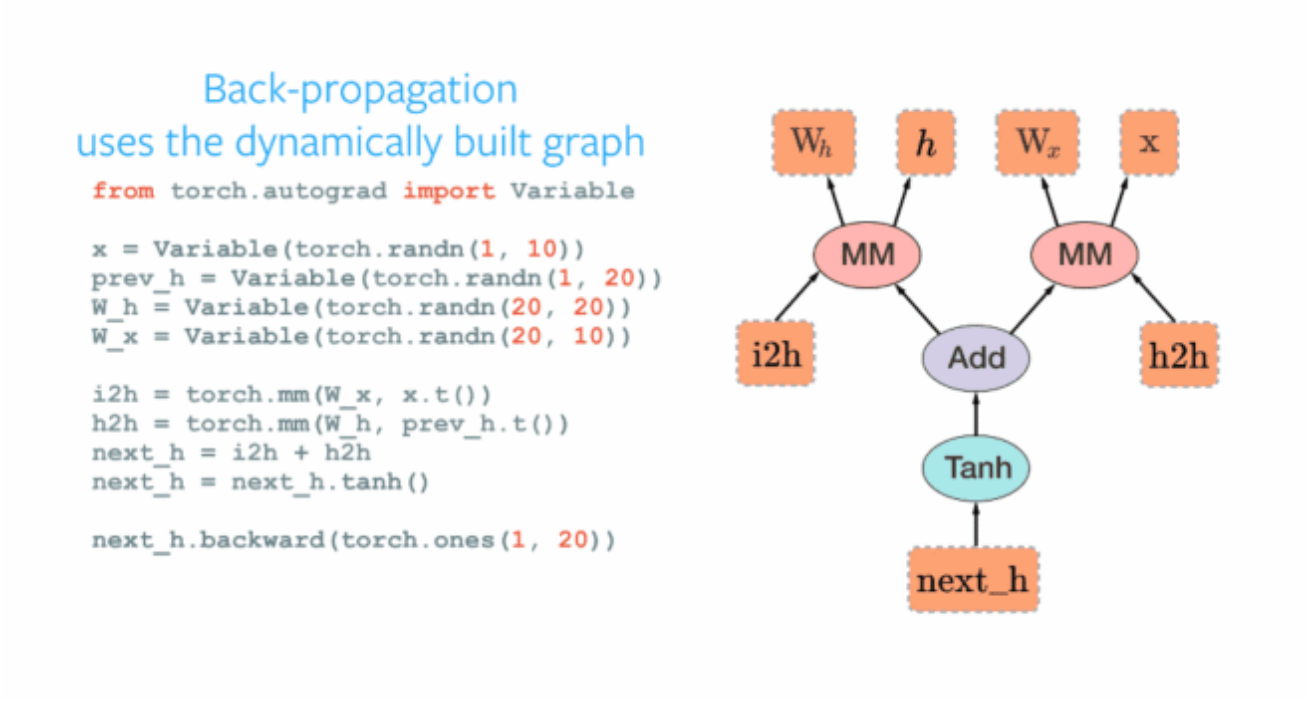
当然，Pipenv 还有很多其他特性，以及很好的文档，因此确保检查完毕，再开始在所有你的 Python 项目上使用它。

2. PyTorch

项目地址：http://pytorch.org/

如果有一个库在今年特别是在深度学习社区中大为流行，那么它很可能是 PyTorch。
PyTorch 是 Facebook 今年推出的深度学习框架。

PyTorch 构建在 Torch 框架之上，并对这个（曾经？）流行框架做了改善，尤其是 PyTorch 是基于 Python 的，这与 Lua 形成鲜明对比。鉴于过去几年人们一直在使用 Python 进行数据科学研究，这为深度学习的普及迈出了重要一步。



最值得注意的是，由于其实现了全新的动态计算图（Dynamic Computational Graph）范式，PyTorch 成为了众多研究者的首选框架之一。当使用其他框架比如 TensorFlow、CNTK、MXNet 编写代码时，必须首先定义一个称之为计算图的东西。该图指定了由我们的代码构建的所有操作与数据流，且它在构建完后会进行编译和利用框架潜在地优化，因此静态计算图能很自然地在 GPU 上实现并行处理。这一范式被称为静态计算图，它很棒，因为你可以利用各种优化，并且这个图一旦建成即可运行在不同设备上（因为执行与构建相分离）。但是，在很多任务中比如自然语言处理中，工作量经常是变动的：你可以在把图像馈送至算法之前把其大小重新调整为一个固定的分辨率，但是相同操作不适用于语句，因为其长度是变化的。这正是 PyTorch 和动态图发挥作用的地方。通过在你的代码中使用标准的 Python 控制指令，图在执行时将被定义，给了你对完成若干任务来说很关键的自由。

当然，PyTorch 也会自动计算梯度（正如你从其他现代深度学习框架中所期望的一样），这非常快，且可扩展，何不试一试呢?

3. Caffe2

项目地址：https://caffe2.ai/

也许这听起来有点疯狂，但是 Facebook 在今年也发布了另外一个很棒的深度学习框架。原始的 Caffe 框架多年来一直被广泛使用，以无与伦比的性能和经过测试的代码库而闻名。但是，最近的深度学习趋势使得该框架在一些方向上停滞不前。Caffe2 正是一次帮助 Caffe 赶上潮流的尝试。

Caffe2 支持分布式训练、部署（甚至在移动端平台）和最新的 CPU、支持 CUDA 的硬件。尽管 PyTorch 更适合于研究，但是 Caffe2 适合大规模部署，正如在 Facebook 上看到的一样。

同样，查看最近的 ONNX 工作。你可以在 PyTorch 中构建和训练你的模型，同时使用 Caffe2 来部署！这是不是很棒?

4. Pendulum

项目地址：https://github.com/sdispater/pendulum

去年，Arrow——一个旨在为你减负同时使用 Python datatime 的库入选了榜单；今年，该轮到 Pendulum 了。

Pendulum 的优点之一在于它是 Python 标准 datetime 类的直接替代品，因此你可以轻易地将其与现有代码整合，并在你真正需要时利用其功能。作者特别注意以确保时间区正确处理，默认每个实例意识到时间区。你也会获得扩展的 timedelta 来简化日期时间的计算。

与其他现有库不同，它努力使 API 具有可预测性行为，因此知道该期望什么。如果你正在做一个涉及 datetime 的重要工作，它会使你更开心。查看该文件获得更多信息：https://pendulum.eustace.io/docs/。

5. Dash

项目地址：https://plot.ly/products/dash/

研究数据科学的时候你可以在 Python 生态系统中使用如 Pandas 和 scikit-learn 等非常棒的工具，还可以使用 Jupyter Notebook 管理工作流程，这对于你和同事之间的协作非常有帮助。但是，当你的分享对象并不知道如何使用这些工具的时候，该怎么办？如何建立一个可以让人们轻松地处理数据并进行可视化的接口？过去的办法是建立一个专业的熟悉 JavaScript 前端设计团队，以建立所需要的 GUI，没有其它办法。

Dash 是几年发布的用于构建网页应用（特别针对于数据可视化的高效利用）的纯 Python 开源库。它建立在 Flask、Plotly.js 和 React 的顶部，可以提供数据处理的抽象层次的接口，从而让我们不需要再学习这些框架，进行高效的开发。该 app 可在浏览器上使用，以后将发布低延迟版本，以在移动设备上使用。

可以在这个网站中查看 Dash 的有趣应用：https://plot.ly/dash/gallery。

6. PyFlux

项目地址：https://github.com/RJT1990/pyflux

Python 中有很多库可以用于研究数据科学和机器学习，但是当你的数据点是随时间演化的度量的时候（例如股价，甚至任何仪器测量值），这就不一样了。

PyFlux 就是一个专用于处理时序数据的开源 Python 库。对时序数据的研究是统计学和经济学的一个子领域，其研究的目的可以是描述时序数据的（关于隐变量或感兴趣特征的）演化行为，也可以是预测时序数据的未来状态。

PyFlux 允许使用概率方法对时序数据建模，拥有多种现代时序数据模型的实现，例如 GARCH。

7. Fire

项目地址：https://github.com/google/python-fire

大多数情况下，我们需要为项目创建一个命令行界面（CLI）。除了传统的 argparse 之外，Python 还有 clik 和 docopt 等很棒的工具。Fire 是今年谷歌发布的软件库，它在解决这个问题上采取了不同的方法。

Fire 是能为任何 Python 项目自动生成 CLI 的开源库。这里的关键点是自动化：我们几乎不需要编写任何代码或文档来构建 CLI。我们只需要调用一个 Fire 方法并把它传递到所希望构建到 CLI 中的目标，例如函数、对象、类、字典或根本不传递参数（这样将会把整体代码导入 CLI）。

一般我们需要阅读该项目下的指导手册，以便通过案例了解它是如何工作的。

8. imbalanced-learn

项目地址：https://github.com/scikit-learn-contrib/imbalanced-learn

在理想的情况中，我们总会有完美的平衡数据集，用它来训练模型将十分舒爽。但不幸的是，在实际中我们总有不平衡的数据集，甚至有些任务拥有非常不平衡的数据。例如，在预测信用卡欺诈的任务中绝大多数交易（99%+）都是合法的，只有极少数的行为需要算法识别为欺诈。如果我们只是朴素地训练 ML 算法，那么算法的性能可能还不如全都预测为占比大的数据，因此在处理这一类问题时我们需要非常小心。

幸运的是，该问题已经经过充分的探讨，且目前存在各种各样的技术以解决不平衡数据。imbalanced-learn 是一个强大的 Python 包，它提供了很多解决数据不平衡的方法。此外，imbalanced-learn 与 scikit-learn 相互兼容，是 scikit-learn-contrib 项目的一部分。

9. FlashText

项目地址：https://github.com/vi3k6i5/flashtext

在大多数数据清理流程或其它工作中，我们可能需要搜索某些文本以替换为其它内容，通常我们会使用正则表达式完成这一工作。在大多数情况下，正则表达式都能完美地解决这一问题，但有时也会发生这样的情况：我们需要搜索的项可能是成千上万条，因此正则表达式的使用将变得十分缓慢。

为此，FlashText 提供了一个更好的解决方案。在该项目作者最初的基准测试中，它极大地缩短了整个操作的运行时间，从 5 天到 15 分钟。FlashText 的优点在于不论搜索项有多少，它所需要的运行时都是相同的。而在常用的正则表达式中，运行时将随着搜索项的增加而线性增长。

```
# FlashText替代关键词


>>> keyword_processor.add_keyword('New Delhi', 'NCR region')


>>> new_sentence = keyword_processor.replace_keywords('I love Big Apple and new delhi.')


>>> new_sentence


>>> # 'I love New York and NCR region.'
```

FlashText 证明了算法和数据结构设计的重要性，即使对于简单的问题，更好的算法也可以轻松超越在最快处理器上运行的朴素实现。

10. Luminoth

项目地址：https://luminoth.ai/

如今图像无处无在，理解图像的内容对于许多应用程序来说都是至关重要的。值得庆幸的是，由于深度学

习的进步，图像处理技术已经有了非常大的进步。

Luminoth 是用于计算机视觉的开源 Python 工具包，它使用 TensorFlow 和 Sonnet 构建，且目前支持 Faster R-CNN 等目标检测方法。此外，Luminoth 不仅仅是一个特定模型的实现，它的构建基于模块化和可扩展，因此我们可以直接定制现有的部分或使用新的模型来扩展它而处理不同的问题，即尽可能对代码进行复用。

它还提供了一些工具以轻松完成构建 DL 模型所需要的工程工作：将数据（图像等）转换为适当的格式以馈送到各种操作流程中，例如执行数据增强、在一个或多个 GPU 中执行训练（分布式训练是训练大规模模型所必需的）、执行评价度量、在 TensorBoard 中可视化数据或模型和部署模型为一个简单的 API 接口等。所以因为 Luminoth 提供了大量的方法，我们可以通过它完成很多关于计算机视觉的任务。

此外，Luminoth 可以直接与 Google Cloud 的 ML 引擎整合，所以即使我们没有强大的 GPU，我们也可以在云端进行训练。

更多优秀的 **Python** 库

除了以上十个非常流行与强大的 Python 库，今年还有一些同样值得关注的 Python 库，包括 PyVips、Requestium 和 skorch 等。

1. PyVips

项目地址：https://github.com/jcupitt/pyvips

你可能还没听过 libvips 库，但你一定听说过 Pillow 或 ImageMagick 等流行的图像处理库，它们支持广泛的格式。然而相比这些流行的图像处理库，libvips 更加快速且只占很少的内存。例如一些基准测试表明它相比 ImageMagick 在处理速度上要快三倍，且还节省了 15 倍的内存占用。

PyVips 是最近发布用于 libvips 的 Python 绑定包，它与 Python 2.7-3.6（甚至是 PyPy）相兼容，它易于使用 pip 安装。所以如果你需要处理图像数据的应用，那么这个库是我们所需要关注的。

2. skorch

项目地址：https://github.com/dnouri/skorch

假设你很喜欢使用 scikit-learn 的 API，但却遇到了需要使用 PyTorch 工作的情况，该怎么办？别担心，skorch 是一个封装，可以通过类似 sklearn 的接口提供 PyTorch 编程。如果你熟悉某些库，就会希望使用相应的直观可理解的句法规则。通过 skorch，你可以得到经过抽象的代码，从而将精力集中于重要的方面。

原文链接：https://tryolabs.com/blog/2017/12/19/top-10-python-libraries-of-2017/ (https://tryolabs.com/blog/2017/12/19/top-10-python-libraries-of-2017/)

POSTED IN 大数据文摘

发布者：wissenpress

查看该作者所有主题 wissenpress

在WORDPRESS.COM的博客.