盘点四大民间机器学习开源框架：Theano、Caffe、Torch 和 SciKit-learn

leiphone.com/news/201701/Lutmxs35U8ZNF7p6.html

2017年1月3日

在上期的谷歌、微软、OpenAI 等巨头的七大机器学习开源项目 看这篇就够了，我们盘点了 TensorFlow，CNTK，SystemML，DeepMind Lab 等各大互联网巨头的开源平台。本期，雷锋网将带领大家来看看诞生于民间（学界）的另外四大开源项目：

1. Theano

Theano 在深度学习框架中是祖师级的存在。它的开发始于 2007，早期开发者包括传奇人物 Yoshua Bengio 和 Ian Goodfellow。

Theano 基于 Python，是一个擅长处理多维数组的库（这方面它类似于 NumPy）。当与其他深度学习库结合起来，它十分适合数据探索。它为执行深度学习中大规模神经网络算法的运算所设计。其实，它可以被更好地理解为一个数学表达式的编译器：用符号式语言定义你想要的结果，该框架会对你的程序进行编译，来高效运行于 GPU 或 CPU。

它与后来出现的 Tensorflow 功能十分相似（或者应该说，Tensorflow 类似 Theano），因而两者常常被放在一起比较。它们本身都偏底层，**同样的，Theano 像是一个研究平台多过是一个深度学习库。你需要从底层开始做许多工作，来创建你需要的模型。**比方说，Theano 没有神经网络的分级。

但随着这些年的发展，大量基于 Theano 的开源深度学习库被开发出来，包括 Keras, Lasagne 和 Blocks。这些更高层级的 wrapper API，能大幅减少开发时间以及过程中的麻烦。甚至，据雷锋网(公众号：雷锋网)所知，很少开发者会使用“裸奔”的 Theano，多数人需要辅助的 API。顺便说一句，**Theano 是一整套生态系统**，别只用它裸奔，然后抱怨不好用。

在过去的很长一段时间内，Theano 是深度学习开发与研究的行业标准。而且，由于出身学界，它最初是为学术研究而设计，这导致深度学习领域的许多学者至今仍在使用 Theano。但随着 Tensorflow 在谷歌的支持下强势崛起，Theano 日渐式微，使用的人越来越少。**这过程中的标志性事件是：创始者之一的 Ian Goodfellow 放弃 Theano 转去谷歌开发 Tensorflow。**

优点：

- Python + NumPy 的组合
- 使用计算图
- RNN 与计算图兼容良好
- 有 Keras 和 Lasagne 这样高层的库

- 不少开发者反映，它的学习门槛比Tensorflow 低

缺点：

- 本身很底层
- 比 Torch 臃肿
- 不支持分布式
- 有的错误信息没什么用
- 大模型的编译时间有时要很久
- 对事先训练过的模型支持不足
- 用的人越来越少

2. Caffe

这又是一个祖师级的深度学习框架，2013 年就已问世。

它的全称是 “Convolution Architecture For Feature Extraction”，意为**“用于特征提取的卷积架构”**，很明白地体现了它的用途。Caffe 的创始人，是加州大学伯克利分校的中国籍博士生贾扬清。当时贾在伯克利计算机视觉与学习中心做研究。博士毕业后，他先后在谷歌和 Facebook 工作。

在 AI 开发者圈子中，Caffe 可以说是无人不知、无人不晓。据 GitHub 最新的机器学习项目热度排名，Caffe 仅位列 Tensorflow 之后，雄踞第二。它是一个被广泛使用的机器视觉库，把 Matlab 执行快速卷积网络的方式带到 C 和 C++。虽然 Caffe 被部分开发者看做是通用框架，但**它的设计初衷是计算机视觉——并不适于其他深度学习应用，比如文字、语音识别和处理时间序列数据。**

Caffe 的主要用途：利用卷积神经网络进行图像分类。这方面它代表了业内一流水平，是开发者的首选。

说到 Caffe，就不得不提 Model Zoo。后者是在 Caffe 基础上开发出的一系列模型的汇聚之地。因此，**开发者使用 Caffe 最大的好处是：**能在 Model Zoo 海量的、事先训练好的神经网络中，选择贴近自己使用需求的直接下载，并立刻就能用。

就雷锋网所知，这些模型中有很多是世界一流的。有很多它们的教程：

- Alex’s CIFAR-10 tutorial with Caffe
- Training LeNet on MNIST with Caffe
- ImageNet with Caffe

业内人士普遍认为，Caffe 适合于以实现基础算法为主要目的的工业应用，有利于快速开发。但对于处理较特殊的任务，它存在灵活性不足的问题——为模型做调整常常需要用 C++ 和 CUDA，虽然 Python 和 Matlab 也能做些小调整。

优点：

非常适合前馈神经网络和图像处理任务

非常适于利用现有神经网络

不写代码也能训练模型

Python 交互界面做得不错

缺点：

需要 C++ 和 CUDA 来编写新 GPU 层级。

在递归神经网络上表现不佳

对于大型神经网络，它十分繁琐（GoogLeNet, ResNet）

没有商业支持

3. Torch

相比其他开源框架，Torch 是一个非主流。

没错，说的就是它的开发语言：**基于1990 年代诞生于巴西的 Lua，而非机器学习界广泛采用的 Python。**其实 Lua 和Python 都属于比较容易入门的语言。但后者明显已经统治了机器学习领域，尤其在学界。而企业界的软件工程师最熟悉的是 Java，对 Lua 也比较陌生。这导致了 Torch 推广的困难。因此，虽然 Torch 功能强大，但并不是大众开发者的菜。

那么它强大在哪里？

- 首先，Torch 非常适用于卷积神经网络。它的开发者认为，Torch 的原生交互界面比其他框架用起来更自然、更得心应手。
- 其次，第三方的扩展工具包提供了丰富的递归神经网络（ RNN）模型。

因为这些强项，许多互联网巨头开发了定制版的 Torch，以助力他们的 AI 研究。**这其中包括 Facebook、Twitter，和被谷歌招安前的 DeepMind。**

与 Caffe 相比，在 Torch 里定义一个新层级比它要容易，因为你不需要写 C++ 代码。和 TensorFlow 和 Theano 比起来，Torch 的灵活度更高，因为它是命令式的；而前两者是陈述式的（declarative），你必须 declare 一个计算图。这使得在 Torch 上进行束搜索（beam search）这样的操作要比它们容易得多。

Torch 的热门应用：在增强学习领域，用卷积神经网络和代理处理图像问题。

兴趣主要在增强学习的开发者， Torch 是首选。

优点：

- 灵活度很高
- 高度模块化
- 容易编写你自己的层级
- 有很多训练好的模型

缺点：

- 需要学 Lua
- 通常需要自己写训练代码
- 不适于循环神经网络
- 没有商业支持

4. SciKit-learn

SciKit-learn 是老牌的开源 Python 算法框架，始于 2007 年的 Google Summer of Code 项目，最初由 David Cournapeau 开发。

它是一个简洁、高效的算法库，提供一系列的监督学习和无监督学习的算法，以用于数据挖掘和数据分析。**SciKit-learn 几乎覆盖了机器学习的所有主流算法，这为其在 Python 开源世界中奠定了江湖地位。**

它的算法库建立在 SciPy (Scientific Python) 之上——你必须先安装 SciPy 才能使用 SciKit-learn 。它的框架中一共包括了：

- NumPy: 基础的多维数组包
- SciPy: 科学计算的基础库
- Matplotlib: 全面的 2D/3D 测绘
- IPython: 改进的交互控制器
- Sympy: 符号数学
- Pandas:数据结构和分析

它命名的由来：SciPy 的扩展和模块在传统上被命名为 SciKits。而提供学习算法的模组就被命名为 scikit-learn。

它与 Python 世界另一大算法框架——TensorFlow 的主要区别是：TensorFlow 更底层。而 SciKit-learn 提供了执行机器学习算法的模块化方案，很多算法模型直接就能用。

优点：

- 经过筛选的、高质量的模型
- 覆盖了大多数机器学习任务
- 可扩展至较大的数据规模
- 使用简单

缺点：

灵活性低

5. MXNet

提到出身学界的开源框架，就不得不提 MXNet。不过，因为亚马逊已将其作为御用平台，因而上期的盘点（谷歌、微软、OpenAI 等巨头的七大机器学习开源项目 看这篇就够了）中已经对其作了介绍。有兴趣的读者请戳链接。