机器学习实践指南

原创2017-09-28译者：Flowsnow[Linux中国](http://mp.weixin.qq.com/s?__biz=MjM5NjQ4MjYwMQ==&mid=2664609513&idx=1&sn=060068ce268569ff1558c5d5ee420538&chksm=bdce8daf8ab904b92a482d249e90d3314128183f22b83c5a2ccca112d4123eebf4584ea6ffa5&mpshare=1&scene=1&srcid=0928eXsteXI50Mnep9x3sJXF##)

在今天的文档中，我会列出 7 个步骤（和 50 多个资源）帮助你开启这个令人兴奋的计算机科学领域的大门，并逐渐成为一个机器学习高手。

-- Karlijn Willems

本文导航

编译自　|　https://medium.freecodecamp.org/how-machines-learn-a-practical-guide-203aae23cafb

　作者　|　Karlijn Willems

　译者　|　Flowsnow

你可能在各种应用中听说过（ML），比如垃圾邮件过滤、光学字符识别（OCR）和计算机视觉。

开启机器学习之旅是一个涉及多方面的漫长旅途。对于新手，有很多的书籍，有学术论文，有指导练习，有独立项目。在这些众多的选择里面，很容易迷失你最初想学习的目标。

所以在今天的文章中，我会列出 7 个步骤（和 50 多个资源）帮助你开启这个令人兴奋的计算机科学领域的大门，并逐渐成为一个机器学习高手。

请注意，这个资源列表并不详尽，只是为了让你入门。 除此之外，还有更多的资源。

1、 学习必要的背景知识

你可能还记得 DataCamp 网站上的学习数据科学[1]这篇文章里面的信息图：数学和统计学是开始机器学习（ML）的关键。 基础可能看起来很容易，因为它只有三个主题。 但不要忘记这些实际上是三个广泛的话题。

在这里需要记住两件非常重要的事情：

◈ 首先，你一定会需要一些进一步的指导，以了解开始机器学习需要覆盖哪些知识点。

◈ 其次，这些是你进一步学习的基础。 不要害怕花时间，有了这些知识你才能构建一切。

第一点很简单：学习线性代数和统计学是个好主意。这两门知识是必须要理解的。但是在你学习的同时，也应该尝试学习诸如最优化和高等微积分等主题。当你越来越深入 ML 的时候，它们就能派上用场。

如果是从零开始的，这里有一些入门指南可供参考：

◈ Khan 学院[2] 对于初学者是非常好的资源，可以考虑学习他们的线性代数和微积分课程。

◈ 在 麻省理工学院 OpenCourseWare[3] 网站上学习线性代数[4] 课程。

◈ Coursera course[5] 网站上对描述统计学、概率论和推论统计学的介绍内容。

*统计学是学习 ML 的关键之一*

如果你更多喜欢阅读书籍，请参考以下内容：

◈ [6]

◈ [7]

◈ [8]

◈ 麻省理工学院在线教材[9]

然而，在大多数情况下，你已经对统计学和数学有了一个初步的了解。很有可能你已经浏览过上面列举的的那些资源。

在这种情况下，诚实地回顾和评价你的知识是一个好主意，是否有一些领域是需要复习的，或者现在掌握的比较好的？

如果你一切都准备好了，那么现在是时候使用 R 或者 Python 应用这些知识了。作为一个通用的指导方针，选择一门语言开始是个好主意。另外，你仍然可以将另一门语言加入到你的技能池里。

为什么这些编程知识是必需的？

嗯，你会看到上面列出的课程（或你在学校或大学学习的课程）将为你提供关于数学和统计学主题的更理论性的介绍（而不是应用性的）。 然而，ML 非常便于应用，你需要能够应用你所学到的所有主题。 所以最好再次复习一遍之前的材料，但是这次需要付诸应用。

如果你想掌握 R 和 Python 的基础，可以看以下课程：

◈ DataCamp 上关于 Python 或者 R 的介绍性课程： Python 语言数据科学介绍[10] 或者 R 语言编程介绍[11]。

◈ Edx 上关于 Python 或者 R 的介绍性课程： Python 语言数据科学介绍[12] 和 R 语言数据科学介绍[13]。

◈ 还有很多其他免费的课程。查看 Coursera[14] 或者 Codeacademy[15] 了解更多。

当你打牢基础知识后，请查看 DataCamp 上的博客 Python 统计学：40+ 数据科学资源[16]。 这篇文章提供了统计学方面的 40 多个资源，这些资源都是你开始数据科学（以及 ML）需要学习的。

还要确保你查看了关于向量和数组的 这篇 SciPy 教程[17]文章，以及使用 Python 进行科学计算的研讨会[18]。

要使用 Python 和微积分进行实践，你可以了解下 SymPy 软件包[19]。

2、 不要害怕在 ML 的“理论”上浪费时间

很多人并不会花很多精力去浏览理论材料，因为理论是枯燥的、无聊的。但从长远来看，在理论知识上投入时间是至关重要的、非常值得的。 你将会更好地了解机器学习的新进展，也能和背景知识结合起来。 这将有助于你保持学习积极性。

此外，理论并不会多无聊。 正如你在介绍中所看到的，你可以借助非常多的资料深入学习。

书籍是吸收理论知识的最佳途径之一。 它们可以让你停下来想一会儿。 当然，看书是一件非常平静的事情，可能不符合你的学习风格。 不过，请尝试阅读下列书籍，看看它是否适合你：

◈ [20]， Tom Mitchell 著，书可能比较旧，但是却很经典。这本书很好的解释介绍了机器学习中最重要的课题，步骤详尽，逐层深入。

◈ （你可以在这里[21]看到这本书的幻灯片版本）：这本书对初学者来说非常棒。 里面讨论了许多实践中的应用程序，其中有一些是在 Tom Mitchell 的书中缺少的。

◈ [22] ：这本书由编写的，仍未完本，但对于那些正在学习 ML 的学生来说，这一定是很好的参考资料。

◈ [23]  由 Jurg Nievergelt 和 Klaus Hinrichs 著。

◈ 也可以参阅 Matthew North 的[24]。 你会发现这本书引导你完成一些最困难的主题。

◈ [25]  由 Alex Smola 和 S.V.N. Vishwanathan 著。

*花些时间看书并研究其中涵盖的资料*

视频和慕课对于喜欢边听边看来学习的人来说非常棒。 慕课和视频非常的多，多到可能你都很难找到适合你的。 下面列出了最知名的几个：

◈ 这个著名的机器学习慕课[26]，是讲的，介绍了机器学习及其理论。 别担心，这个慕课讲的非常好，一步一步深入，所以对初学者来说非常适用。

◈ 麻省理工学院 Open Courseware 的 6034 课程的节目清单[27]，已经有点前沿了。 在你开始本系列之前，你需要做一些 ML 理论方面的准备工作，但是你不会后悔的。

在这一点上，重要的是要将各种独立的技术融会贯通，形成整体的结构图。 首先了解关键的概念：和的区别、分类和回归等。 手动（书面）练习可以派上用场，能帮你了解算法是如何工作的以及如何应用这些算法。 在大学课程里你经常会找到一些书面练习，可以看看波特兰州立大学的 ML 课程[28]。

3、 开始动手

通过看书和看视频了解理论和算法都非常好，但是需要超越这一阶段，就要开始做一些练习。你要学着去实现这些算法，应用学到的理论。

首先，有很多介绍 Python 和 R 方面的机器学习的基础知识。当然最好的方法就是使用交互式教程：

◈ Python 机器学习：Scikit-Learn 教程[29]，在这篇教程里面，你可以学到使用 Scikit-Learn 构建模型的 KMeans 和支持向量机（SVM）相关的知名算法。

◈ 给初学者的 R 语言机器学习教程[30] 用 R 中的类和 caret 包介绍机器学习。

◈ [Keras 教程：Python 深度学习25[31] 涵盖了如何一步一步的为分类和回归任务构建多层感知器（MLP）。

还请查看以下静态的（非互动的）教程，这些需要你在 IDE 中操作：

◈ 循序渐进：Python 机器学习[32]： 一步一步地学习 Scikit-Learn。

◈ 循序渐进：使用 Keras 开发你的第一个神经网络[33]： 按这个教程一步一步地使用 Keras 开发你的第一个神经网络。

◈ 你可以考虑看更多的教程，但是机器学习精要[34]这篇教程是非常好的。

除了教程之外，还有一些课程。参加课程可以帮助你系统性地应用学到的概念。 经验丰富的导师很有帮助。 以下是 Python 和机器学习的一些互动课程：

◈ 用 scikit-learn 做监督学习[35]： 学习如何构建预测模型，调整参数，并预测在未知数据上执行的效果。你将使用 Scikit-Learn 操作真实世界的数据集。

◈ 用 Python 做无监督学习[36]： 展示给你如何从未标记的数据集进行聚类、转换、可视化和提取关键信息。 在课程结束时，还会构建一个推荐系统。

◈ Python 深度学习[37]： 你将获得如何使用 Keras 2.0 进行深度学习的实践知识，Keras 2.0 是前沿的 Python 深度学习库 Keras 的最新版本。

◈ 在 Python 中应用机器学习[38]： 将学习者引入到机器学习实践中，更多地关注技术和方法，而不是这些方法背后的统计学知识。

*理论学习之后，花点时间来应用你所学到的知识。*

对于那些正在学习 R 语言机器学习的人，还有这些互动课程：

◈ 机器学习介绍[39] 可以让你宏观了解机器学习学科最常见的技术和应用，还可以更多地了解不同机器学习模型的评估和训练。这门课程剩下的部分重点介绍三个最基本的机器学习任务： 分类、回归和聚类。

◈ R 语言无监督学习[40] ，用 R 语言从 ML 角度提供聚类和降维的基本介绍。 可以让你尽快获得数据的关键信息。

◈ 实操机器学习[41]涵盖了构建和应用预测功能的基本组成部分，其重点是实际应用。

最后，还有很多书籍以偏向实践的方式介绍了 ML 主题。 如果你想借助书籍内容和 IDE 来学习，请查看这些书籍：

◈ [42]，Sebastian Raschka 著。

◈ [43]，Sebastian Raschka 著。

◈ [44]，Brett Lantz 著。

4、 练习

实践比使用 Python 进行练习和修改材料更重要。 这一步对我来说可能是最难的。 在做了一些练习后看看其他人是如何实现 ML 算法的。 然后，开始你自己的项目，阐述你对 ML 算法和理论的理解。

最直接的方法之一就是将练习的规模做得更大些。 要做一个更大的练习，就需要你做更多的数据清理和功能工程。

◈ 从 Kaggle[45] 开始。 如果你需要额外的帮助来征服所谓的“数据恐惧”，请查看 Kaggle 的 Python 机器学习教程[46] 和 Kaggle 的 R 语言机器学习教程[47]。 这些将带给您快速的提升。

◈ 此后，你也可以自己开始挑战。 查看这些网站，您可以在其中找到大量的 ML 数据集：UCI 机器学习仓库[48]，用于机器学习的公开数据集[49] 和 data.world[50]。

*熟能生巧。*

5、 项目

虽然做一些小的练习也不错，但是在最后，您需要做一个项目，可以在其中展示您对使用到的 ML 算法的理解。

最好的练习是实现你自己的 ML 算法。 您可以在以下页面中阅读更多关于为什么您应该做这样的练习，以及您可以从中学到什么内容：

◈ 为什么有许多先进的 API，比如 tensorflow，还需要自己手动实现机器学习的算法？[51]

◈ 为什么要从头开始实现机器学习算法？[52]

◈ 使用 Python 从头开始实现一个分类器，我能从中学到什么？[53]

接下来，您可以查看以下文章和仓库。 可以从中获得一些灵感，并且了解他们是如何实现 ML 算法的。

◈ 如何实现机器学习算法[54]

◈ 从头开始学习机器学习[55]

◈ 从头开始学习机器学习算法[56]

*开始时项目可能会很难，但是可以极大增加你的理解。*

6、 不要停止

对 ML 的学习永远不能停止，即使你在这个领域工作了十年，总是有新的东西要学习，许多人都将会证实这一点。

例如，ML 趋势，比如现在就很受欢迎。你也可以专注于那些现在不怎么火，但是将来会火的话题上。如果你想了解更多，可以看看这个有趣的问题和答案[57]。

当你苦恼于掌握基础知识时，你最先想到的可能不是论文。 但是它们是你紧跟最新研究的一个途径。 论文并不适合刚刚开始学习的人，但是绝对适合高级人员。

◈ 20 篇最新的机器学习和深度学习领域的顶级研究论文[58]

◈ 机器学习研究杂志[59]

◈ 优秀的深度学习论文[60]

◈ 机器学习的一些最好的研究论文和书籍[61]

其他技术也是需要考虑的。 但是当你刚开始学习时，不要担心这些。 例如，您可以专注于 Python 或 R 语言 （取决于你已经知道哪一个），并把它到你的技能池里。 你可以通过这篇文章来查找一些感兴趣的资源。

如果您还想转向大数据，您可以考虑研究 Spark。 这里有一些有趣的资源：

◈ 在 R 语言中使用 sparklyr 来了解 Spark[62]

◈ Spark 数据科学与工程[63]

◈ 介绍 Apache Spark[64]

◈ Apache Spark 分布式机器学习[65]

◈ 用 Apache Spark 进行大数据分析[66]

◈ 初学者指南：用 Python 操作 Apache Spark[67]

◈ PySpark RDD 速查表[68]

◈ PySpark SQL 速查表[69]

其他编程语言，比如 Java、JavaScript、C 和 C++ 在 ML 中越来越重要。 从长远来看，您可以考虑将其中一种语言添加到学习列表中。 你可以使用这些博客文章来指导你选择：

◈ 机器学习和数据科学最流行的编程语言[70]

◈ 机器学习和数据科学最流行的语言是...[71]

*学无止境。*

7、 利用一切可以利用的资源

机器学习是一个充满难度的话题，有时候可能会让你失去动力。 或者也许你觉得你需要点改变。 在这种情况下，请记住，有很多资源可以让你打消掉这种想法。 查看以下资源：

播客是可以让你继续你的 ML 旅程，紧跟这个领域最新的发展的伟大资源：

◈ 谈论机器[72]

◈ 数据怀疑论者[73]

◈ 线性化[74]

◈ 本周的机器学习及 AI[75]

◈ 机器学习 101[76]

当然，还有更多的播客。

文档和软件包源代码是深入了解 ML 算法的实现的两种方法。 查看这些仓库：

◈ Scikit-Learn[77]：知名的 Python ML 软件包

◈ Keras[78]： Python 深度学习软件包

◈ caret[79]： 非常受欢迎的用于分类和回归训练 R 软件包

可视化是深入 ML 理论的最新也是最流行的方式之一。 它们对初学者来说非常棒，但对于更高级的学习者来说也是非常有趣的。 你肯定会被下面这些可视化资源所吸引，它们能让你更加了解 ML 的工作原理：

◈ 机器学习的可视化介绍[80]

◈ Distill[81] 使 ML 研究清晰，动态和生动。

◈ 如果你想玩下神经网络架构，可以看下 Tensorflow - 神经网络游乐场[82]。

◈ 更多的看这里：机器学习算法最佳的可视化方法是什么？[83]

*学习中的一些变化更加能激励你。*

现在你可以开始了

现在一切都取决于你自己了。学习机器学习是一个持续的过程，所以开始的越早就会越好。 运用你手边的一切工具开始吧。 祝你好运，并确保让我们知道你的进步。

这篇文章是我基于 Quora 问题（小白该如何开始机器学习[84]）给出的答案。

作者简介：

Karlijn Willems，数据科学记者

via: https://medium.freecodecamp.org/how-machines-learn-a-practical-guide-203aae23cafb

作者：Karlijn Willems[85] 译者：Flowsnow 校对：wxy

本文由 LCTT 原创编译，Linux中国 荣誉推出

[阅读原文](http://mp.weixin.qq.com/s?__biz=MjM5NjQ4MjYwMQ==&mid=2664609513&idx=1&sn=060068ce268569ff1558c5d5ee420538&chksm=bdce8daf8ab904b92a482d249e90d3314128183f22b83c5a2ccca112d4123eebf4584ea6ffa5&mpshare=1&scene=1&srcid=0928eXsteXI50Mnep9x3sJXF##)