案例一

作者 | 子实（某科研院程序员）

本人码农，从六月开始正式接触机器学习(其实五年前的本科毕设就是在生物信息领域应用神经网络的项目，但是非常浅薄)，深吸一口气，先要声明“人之患在好为人师”，我用的步骤只是适合我，下面的内容仅供参考。

第一步：复习线性代数。(学渣的线代忘了好多-\_-||)

懒得看书就直接用了著名的——麻省理工公开课：线性代数，深入浅出效果拔群，以后会用到的SVD、希尔伯特空间等都有介绍 - http://open.163.com/special/opencourse/daishu.html

广告：里面有一些deeplearning实验，适合新手学习：https://github.com/826235954/deeplearning-.git

第二步：入门机器学习算法。

还是因为比较懒，也就直接用了著名的——斯坦福大学公开课 ：机器学习课程(http://open.163.com/special/opencourse/machinelearning.html)，吴恩达教授的老版cs229的视频，讲的非常细(算法的目标->数学推演->伪代码)。这套教程唯一的缺点在于没有介绍最近大火的神经网络，但其实这也算是优点，让我明白了算法都有各自的应用领域，并不是所有问题都需要用神经网络来解决。

多说一点，这个课程里详细介绍的内容有：一般线性模型、高斯系列模型、SVM理论及实现、聚类算法以及EM算法的各种相关应用、PCA/ICA、学习理论、马尔可夫系列模型。课堂笔记在：CS 229: Machine Learning (Course handouts) -http://cs229.stanford.edu/syllabus.html，同样非常详细。

广告：里面有一些deeplearning实验，适合新手学习：https://github.com/826235954/deeplearning-.git

第三步：尝试用代码实现算法。

依然因为比较懒，继续直接使用了著名的——机器学习 | Coursera(https://www.coursera.org/learn/machine-learning)，还是吴恩达教授的课程，只不过这个是极简版的cs229，几乎就是教怎么在matlab里快速实现一个模型(这套教程里有神经网络基本概念及实现)。这套课程的缺点是难度比较低，推导过程非常简略，但是这也是它的优点——让我专注于把理论转化成代码。

广告：里面有一些deeplearning实验，适合新手学习：https://github.com/826235954/deeplearning-.git

第四步：自己实现功能完整的模型——进行中。

还是因为比较懒，搜到了cs231n的课程视频 CS231n Winter 2016 - YouTube(https://www.youtube.com/playlist?list=PLkt2uSq6rBVctENoVBg1TpCC7OQi31AlC) ，李飞飞教授的课，主讲还有Andrej Karpathy和Justin Johnson，主要介绍卷积神经网络在图像识别/机器视觉领域的应用（前面神经网络的代码没写够？这门课包你嗨到爆~到处都是从零手写~）。这门课程的作业就更贴心了，直接用Jupyter Notebook布置的，可以本地运行并自己检查错误。主要使用Python以及Python系列的科学计算库（Scipy/Numpy/Matplotlib）。课堂笔记的翻译可以参考 智能单元 - 知乎专栏(https://zhuanlan.zhihu.com/p/22339097)，主要由知友杜客翻译，写的非常好~

在多说一点，这门课对程序员来说比较走心，因为这个不像上一步中用matlab实现的作业那样偏向算法和模型，这门课用Python实现的模型同时注重软件工程，包括常见的封装layer的forward/backward、自定义组合layer、如何将layer组成网络、如何在网络中集成batch-normalization及dropout等功能、如何在复杂模型下做梯度检查等等；最后一个作业中还有手动实现RNN及其基友LSTM、编写有助于调试的CNN可视化功能、Google的DeepDream等等。（做完作业基本就可以看懂现在流行的各种图片风格变换程序了，如 cysmith/neural-style-tf - https://github.com/cysmith/neural-style-tf）另外，这门课的作业实现非常推崇computational graph，不知道是不是我的幻觉……要注意的是讲师A.K的语速奇快无比，好在YouTube有自动生成解说词的功能，准确率还不错，可以当字幕看。

广告：里面有一些deeplearning实验，适合新手学习：https://github.com/826235954/deeplearning-.git

因为最近手头有论文要撕，时间比较紧，第四步做完就先告一段落。后面打算做继续业界传奇Geoffrey Hinton教授的Neural Networks for Machine Learning | Coursera(https://www.coursera.org/learn/neural-networks)，再看看NLP的课程 Stanford University CS224d: Deep Learning for Natural Language Processing(http://cs224d.stanford.edu)，先把基础补完，然后在东瞅瞅西逛逛看看有什么好玩的……

PS：一直没提诸如TensorFlow之类的神器，早就装了一个（可以直接在conda中为Tensorflow新建一个env，然后再装上Jupyter、sklearn等常用的库，把这些在学习和实践ML时所用到的库都放在一个环境下管理，会方便很多），然而一直没时间学习使用，还是打算先忍着把基础部分看完，抖M总是喜欢把最好的留在最后一个人偷偷享受2333333（手动奸笑

PS\*\*2：关于用到的系统性知识，主要有：

线性代数，非常重要，模型计算全靠它~一定要复习扎实，如果平常不用可能忘的比较多；

高数+概率，这俩只要掌握基础就行了，比如积分和求导、各种分布、参数估计等等。（评论中有知友提到概率与数理统计的重要性，我举四肢赞成，因为cs229中几乎所有算法的推演都是从参数估计及其在概率模型中的意义起手的，参数的更新规则具有概率上的可解释性。对于算法的设计和改进工作，概统是核心课程，没有之一。答主这里想要说的是，当拿到现成的算法时，仅需要概率基础知识就能看懂，然后需要比较多的线代知识才能让模型高效的跑起来。比如最近做卷积的作业， 我手写的比作业里给出的带各种trick的fast函数慢几个数量级，作业还安慰我不要在意效率，岂可修！）

需要用到的编程知识也就是Matlab和Numpy了吧，Matlab是可以现学现卖的；至于Python，就看题主想用来做什么了，如果就是用来做机器学习，完全可以一天入门，如果想要做更多好玩的事，一天不行那就两天。（贴一个Python/Numpy的简要教程：Python Numpy Tutorial - http://cs231n.github.io/python-numpy-tutorial/，是cs231n的课堂福利。）

我感觉机器学习的先修就这么点，记得Adobe的冯东大神也说过机器学习简直是21世界的黑科技——因为理论非常简单但是效果惊人的好。

====

既然提到好玩的，墙裂推荐 Kaggle: Your Home for Data Science(https://www.kaggle.com) ，引用维基上的介绍：

Kaggle是一个数据建模和数据分析竞赛平台。企业和研究者可在其上发布数据，统计学者和数据挖掘专家可在其上进行竞赛以产生最好的模型。这一众包模式依赖于这一事实，即有众多策略可以用于解决几乎所有预测建模的问题，而研究者不可能在一开始就了解什么方法对于特定问题是最为有效的。Kaggle的目标则是试图通过众包的形式来解决这一难题，进而使数据科学成为一场运动。