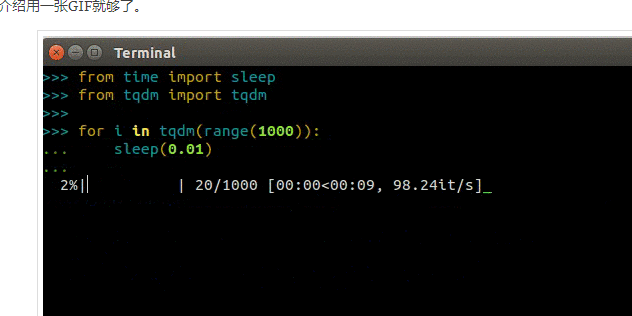
Python基本是我目前工作、计算、数据挖掘的唯一编程语言（除了符号计算用Mathematica外）。当然，基本的Python功能并不是很强大，但它胜在有巨量的第三方扩展库。在选用Python的第三方库时，我都会经过仔细考虑，希望能挑选出最简单的、最直观的一个（因为本人比较笨，太复杂用不了）。在数据处理方面，我用得最多的是Numpy和Pandas，这两个绝对称得上王者级别的库，当然不能不提的是Scipy，但我很少直接用它，一般会通过Pandas间接调用了；可视化方面不用说是Matplotlib了；在建模方面，我会用Keras，直接上深度学习模型，Keras已经成为相当流行的深度学习框架了，如果做文本挖掘，通常还会用到jieba（分词）、Gensim（主题建模，包含了诸如word2vec之类的模型），机器学习库还有流行的Scikit Learn，但我很少用；网络方面，写爬虫我用requests，这是个人性化的网络库，如果写网站，我会用bottle，这是个单文件版的迷你框架，一切由自己定义，当然，我也不会去写什么大型网站，我就写一个简单的的接口那样而已；最后如果要并行的话，一般直接用multiprocessing。

不过，以上都不是本文要推荐的，本文要推荐的是两个可以渗透到日常写代码的库，它实现了我们平时很多时候都需要的功能，但是不用增加什么代码，绝对让人眼前一亮。

1. tqdm #

tqdm的介绍用一张GIF就够了。



tqdm

tqdm

说白了，它就是用来显示进度条的，很漂亮，使用很直观（在循环体里边加个tqdm），而且基本不影响原程序效率。名副其实的“太强大美”了！这样在写运行时间很长的程序时，是该多么舒服啊！

2. retry #

正如它的名字，retry是用来实现重试的。很多时候我们都需要重试功能，比如写爬虫的时候，有时候就会出现网络问题导致爬取失败，然后就需要重试了，一般我是这样写的（每隔两秒重试一次，共5次）：

import time

def do\_something():

xxx

for i in range(5):

try:

do\_something()

break

except:

time.sleep(2)

这样未免有些累赘。有了retry后，只需要。

from retry import retry

@retry(tries=5, delay=2)

def do\_something():

xxx

do\_something()

也就是在函数的定义前，加一句@retry就行了。

Python果然是绝对省心～