

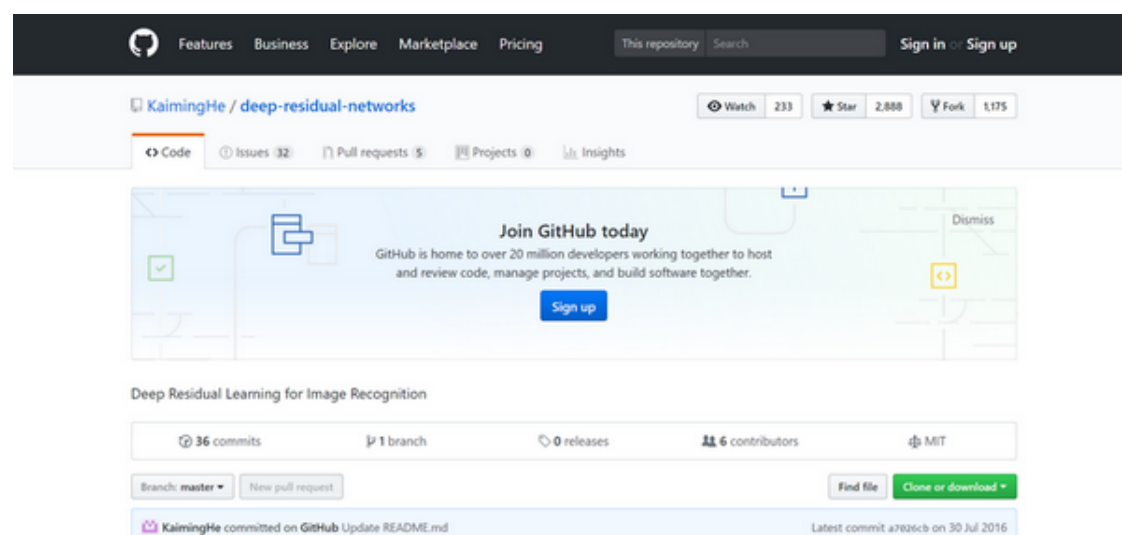
## 2.8 使用开源的实现方案（Using open-source implementations）

你已经学过几个非常有效的神经网络和 **ConvNet** 架构，在接下来的几段视频中我想与你分享几条如何使用它们的实用性建议，首先从使用开放源码的实现开始。

事实证明很多神经网络复杂细致，因而难以复制，因为一些参数调整的细节问题，例如学习率衰减等等，会影响性能。所以我发现有些时候，甚至在顶尖大学学习 AI 或者深度学习的博士生也很难通过阅读别人的研究论文来复制他人的成果。幸运的是有很多深度学习的研究者都习惯把自己的成果作为开发资源，放在像 **GitHub** 之类的网站上。当你自己编写代码时，我鼓励你考虑一下将你的代码贡献给开源社区。如果你看到一篇研究论文想应用它的成果，你应该考虑做一件事，我经常做的就是网络上寻找一个开源的实现。因为你如果能得到作者的实现，通常要比你从头开始实现要快得多，虽然从零开始实现肯定可以是一个很好的锻炼。

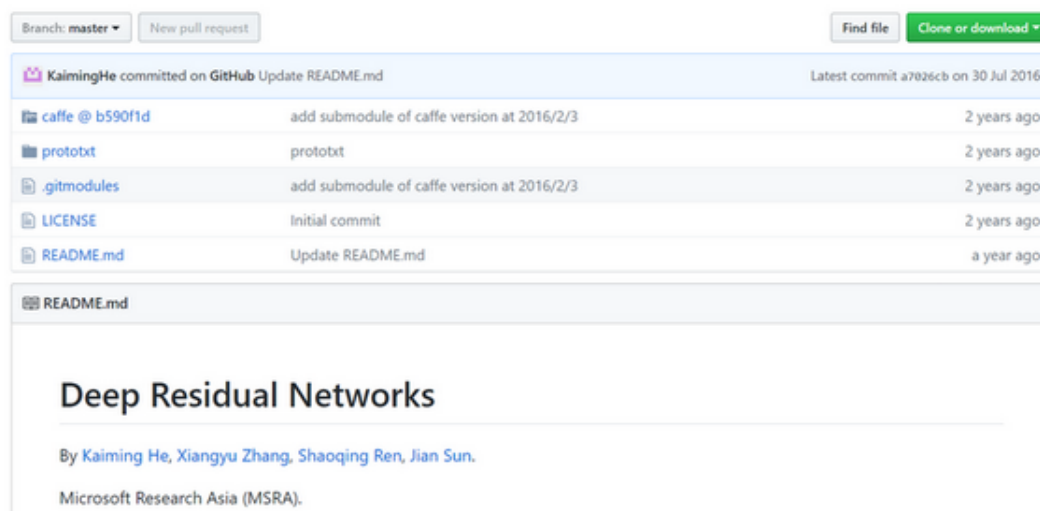
如果你已经熟悉如何使用 **GitHub**，这段视频对你来说可能没什么必要或者没那么重要。但是如果你不习惯从 **GitHub** 下载开源代码，让我来演示一下。

（整理者注：**ResNets** 实现的 **GitHub** 地址 <https://github.com/KaimingHe/deep-residual-networks>）

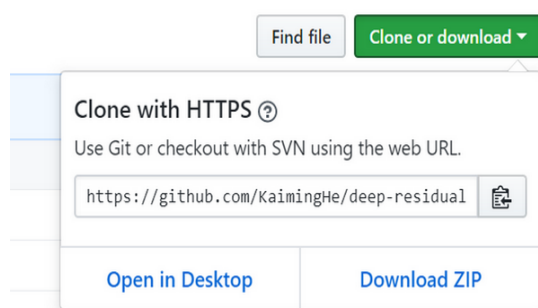


假设你对残差网络感兴趣，那就让我们搜索 **GitHub** 上的 **ResNets**，那么你可以在 **GitHub** 看到很多不同的 **ResNet** 的实现。我就打开这里的第一个网址，这是一个 **ResNets** 实现的 **GitHub** 资源库。在很多 **GitHub** 的网页上往下翻，你会看到一些描述，这个实现的文字说明。

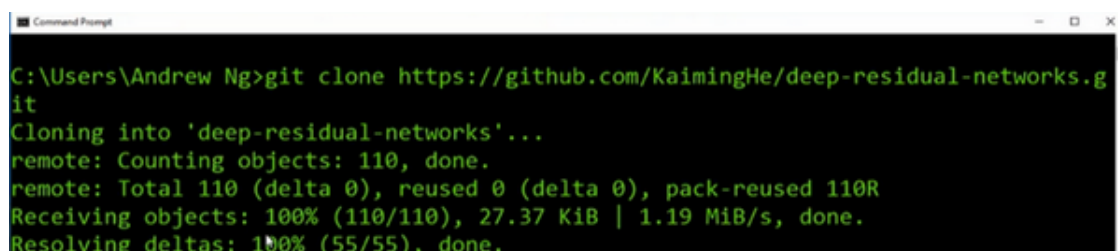
这个 **GitHub** 资源库，实际上是由 **ResNet** 论文原作者上传的。这些代码，这里有麻省理工学院的许可，你可以点击查看此许可的含义，**MIT** 许可是比较开放的开源许可之一。我将下载代码，点击这里的链接，它会给你一个 **URL**，通过这个你可以下载这个代码。



我点击这里的按钮（**Clone or download**），将这个 **URL** 复制到我的剪切板里。



（整理者注：**NG** 此处使用的是 **linux** 系统的 **bash** 命令行，对于 **win10** 系统，可以开启 **linux** 子系统功能，然后在 **win10** 应用商店下载 **ubuntu** 安装，运行 **CMD**，输入命令 **bash** 即可进入 **linux** 的 **bash** 命令行）



接着到这里，接下来你要做的就是输入 **git clone**，接着粘贴 **URL**，按下回车，几秒之内就将这个资源库的副本下载到我的本地硬盘里。

让我们进入目录，让我们看一下，比起 **Windows**，我更习惯用 **Mac**，不过没关系，让我们试一下，让我们进入 **prototxt**，我认为这就是存放这些网络文件的地方。让我们看一下这个文件。因为这个文件很长，包含了 **ResNet** 里 101 层的详细配置。我记得，从这个网页上

看到这个特殊实现使用了 **Caffe** 框架。但如果你想通过其它编程框架来实现这一代码，你也可以尝试寻找一下。

```
C:\Users\Andrew Ng>cd deep-residual-networks

C:\Users\Andrew Ng\deep-residual-networks>dir
Volume in drive C is Windows
Volume Serial Number is 4E6C-6C59

Directory of C:\Users\Andrew Ng\deep-residual-networks

10/15/2017  06:30 PM    <DIR>          .
10/15/2017  06:30 PM    <DIR>          ..
10/15/2017  06:30 PM                96 .gitmodules
10/15/2017  06:30 PM    <DIR>          caffe
10/15/2017  06:30 PM            1,100 LICENSE
10/15/2017  06:30 PM    <DIR>          prototxt
10/15/2017  06:30 PM            7,160 README.md
               3 File(s)              8,356 bytes
               4 Dir(s)  515,663,204,352 bytes free
```

```
C:\Users\Andrew Ng\deep-residual-networks>cd prototxt

C:\Users\Andrew Ng\deep-residual-networks\prototxt>dir
Volume in drive C is Windows
Volume Serial Number is 4E6C-6C59

Directory of C:\Users\Andrew Ng\deep-residual-networks\prototxt

10/15/2017  06:30 PM    <DIR>          .
10/15/2017  06:30 PM    <DIR>          ..
10/15/2017  06:30 PM            69,987 ResNet-101-deploy.prototxt
10/15/2017  06:30 PM            104,809 ResNet-152-deploy.prototxt
10/15/2017  06:30 PM            34,820 ResNet-50-deploy.prototxt
               3 File(s)             209,616 bytes
               2 Dir(s)  515,663,151,104 bytes free
```

```
C:\Users\Andrew Ng\deep-residual-networks\prototxt>more ResNet-101-deploy.prototxt
```

```
name: "ResNet-101"
input: "data"
input_dim: 1
input_dim: 3
input_dim: 224
input_dim: 224

layer {
  bottom: "data"
  top: "conv1"
  name: "conv1"
  type: "Convolution"
  convolution_param {
    num_output: 64
    kernel_size: 7
    pad: 3
    stride: 2
    bias_term: false
  }
  -- More (0%) --
```

如果你在开发一个计算机视觉应用,一个常见的工作流程是,先选择一个你喜欢的架构,或许是你在这门课中学习到的,或者是你从朋友那听说的,或者是从文献中看到的,接着寻找一个开源实现,从 **GitHub** 下载下来,以此基础开始构建。这样做的优点在于,这些网络通常都需要很长的时间来训练,而或许有人已经使用多个 **GPU**,通过庞大的数据集预先训练了这些网络,这样一来你就可以使用这些网络进行迁移学习,我们将在下一节课讨论这些内容。

当然,如果你是一名计算机视觉研究员,从零来实现这些,那么你的工作流程将会不同,如果你自己构建,那么希望你将工作成果贡献出来,放到开源社区。因为已经有如此多计算机视觉研究者为了实现这些架构做了如此之多的工作,我发现从开源项目上开始是一个更好的方法,它也确实是一个更快开展新项目的方法。