

实验研究工作流程详解：如何把你的机器学习想法变成现实

2017年06月05日 13:15:17 机器之心

0| | | |

选自dustintran

作者：DUSTIN TRAN

机器之心编译

参与：李泽南、Smith


从研究思想的提出到实验的具体实现是工程中的基础环节。但是这一过程常常被一些明显的小瑕疵所影响。在学术界，研究生需要辛苦的科研——大量的编写代码，撰写说明以及论文创作。新的工程项目经常需要全新的代码库，而且通常很难把过去应用过的代码直接延伸到这些新项目当中去。

基于此种情况，哥伦比亚大学计算机科学博士生及 OpenAI 研究者 Dustin Tran 从其个人角度概述了从研究思想到实验过程的步骤。其中最关键的步骤是提出新观点，这往往需要大量时间；而且至少对作者来说，实验环节不仅是学习，更是解决无法预测的问题的关键所在。另外，作者还明确说明：这个工作流程仅适用于实验方面的研究，理论研究则需要遵循另外的流程，尽管这两者也有一些共同点。机器之心对该工作流程进行了编译介绍，你有什么想法呢？不妨在评论中与我们分享。

找对问题

在真正开始一个项目之前，如何让你的想法「落地」成为更正式的议题是非常关键的。有时它很简单——就像导师会给你分配任务；或者处理一个特定的数据集或实际问题；又或是和你的合作者进行谈话来确定工作内容。

更为常见的是，研究其实是一系列想法（idea）不断迭代所产生的结果，这些想法通常是通过日常谈话、近期工作、阅读专业内和专业外领域文献和反复研读经典论文所产生的。



机器之心

专业的人工智能媒体与产业服务平台。

- 热文排行
- 日榜周榜月榜
- 1 印度服软了：这件事上，我们还要30年才..

2 乔布斯、马斯克....优步CEO：创始人是...

3 校园贷代理江湖：月入10万！两年建公司..

4 5万亿巨债压顶 800多个大老板命悬一线

5 房价要下跌了？真相远比你想象的要残酷..

6 存款利率超过5%却没人敢存，银行的套...

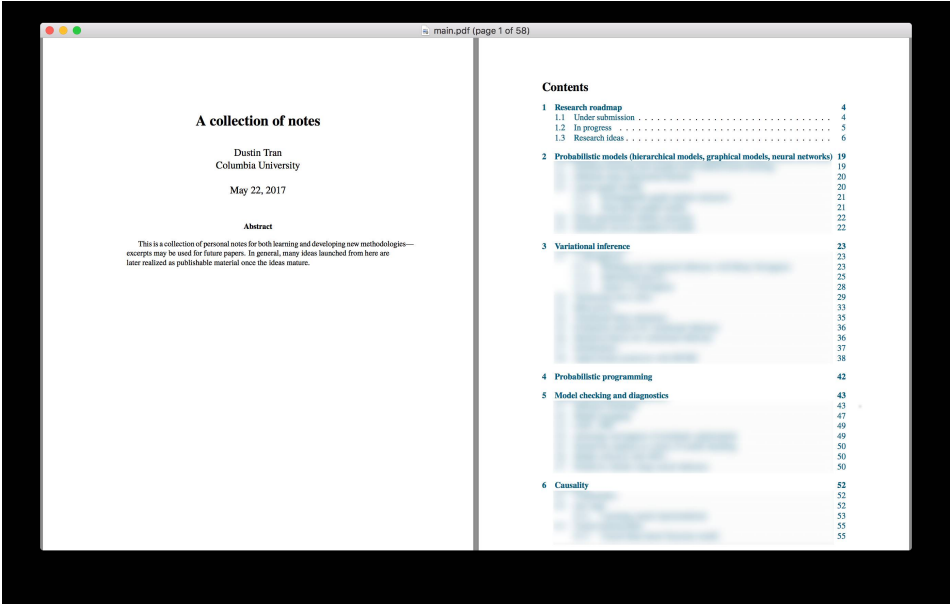
7 楼市入冬！真正的原因，出乎你的意料

8 卡塔尔遭断崖式外交崩盘 中国两大机遇...

9 未来5年，预测美国房价会涨20%，你会...

10 吞了168亿，乐视为何依然病怏怏？





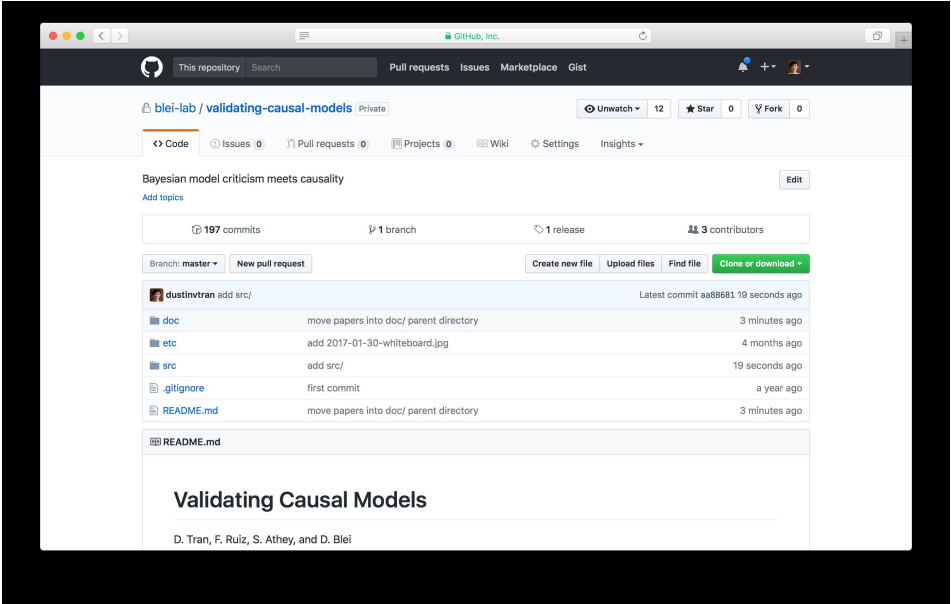
我的所有尚未探索过的研究思想的主文档

我发现了一种方法非常有用——即保持一个单一的主文档（master document），这通常需要很多工作。

首先，它有一个项目列表来排列所有的研究想法、问题和题目。有时它们可以是比较高层面的问题，就像「用于强化学习的贝叶斯/生成方法」、「解决机器学习领域的公平性问题」；也可以是一些很具体的议题，比如「处理 EP 中记忆复杂度的推理网络」、「规模偏置的与对称的 Dirichlet 先验的分析」。我经常努力把项目列表写得更加简明：子内容通过一些链接进行展开。

然后，根据接下来要做的工作来对 idea 清单进行分类。这通常会给我的后续研究指明方向。我也可以根据其方向是否和我的研究观点一致、其必要性和有效性随时修改这些项目的优先级。更重要的是，这个列表清单不仅仅是关于后续观点的，更是关于接下来我更愿意研究什么内容的。从长远角度来考虑，这对于找到重要问题和提出简单新颖的解决方法是有重要贡献的。我经常访问这个清单，重新安排事务，添加新想法，删除不必要的议题。最终当我可以详细说明一个 idea 的时候，它就可以成为一篇比较正式的论文了。一般来说，我发现在同一个位置（同一个格式）迭代 idea 的过程可以使正式论文写作中的衔接和实验过程都变得更加流畅。

管理一个项目



我们为近期的 arXiv 预印本搭建的 repository

我喜欢在 GitHub 存储库中维护研究项目。不管一个「单元」的研究是多少，我都会将其定义成某种相对自我包含的东西；比如，它可能会连接到一篇特定的论文、一个已被应用的数据分析或目前一个特定主题。

GitHub 存储库不仅可用于跟踪代码，而且还可用于跟踪一般的研究进程、论文写作进度或尝试其它合作项目。但项目的组织方式一直以来都是一个痛点。我比较喜欢以下的结构，该结构来自 Dave Blei，可参阅：http://www.cs.columbia.edu/~blei/seminar/2016_discrete_data/notes/week_01.pdf

-- doc/ -- 2017-nips/ -- preamble/ -- img/ -- main.pdf -- main.tex -- introduction.tex--

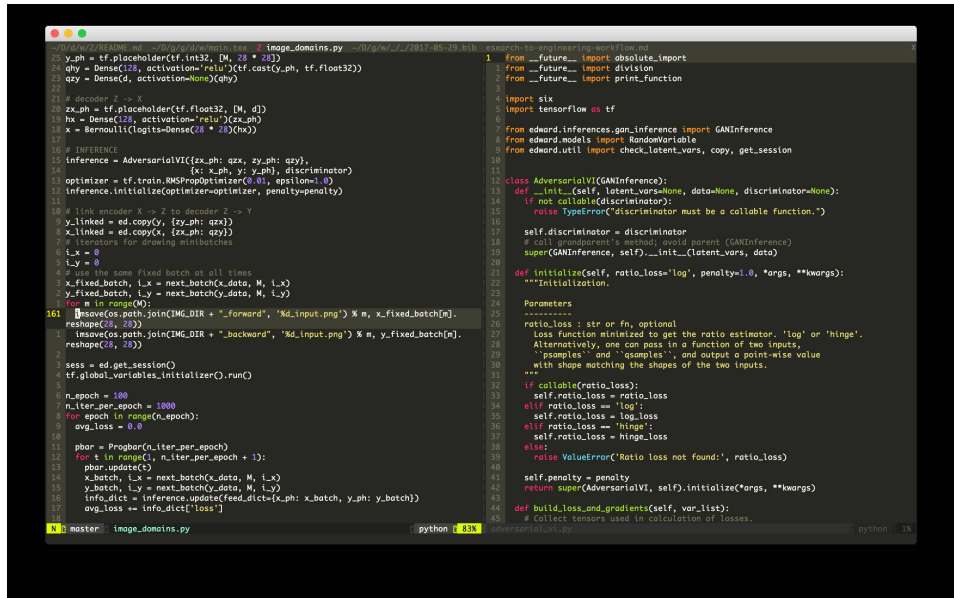
README.md 为自己和合作者保持了一个需要去做的事的列表，这让面临的问题和前进的方向变得明确。

doc/包含所有的记录事项，每个子目录都包含一个会议纪要或是文献提交，main.tex 是主要文档，每一章节都是不同文件，如 introduction.tex，让每个章节分开可以让多人同时处理不同的章节，避免合并冲突。有些人喜欢在主要实验完成后一次写出完整论文，但我更喜欢把论文作为目前想法的记录，并且让它和想法本身一样，随着实验的进展不断推进。

etc/是其他与前面的目录无关的内容。我通常用它来存储项目中讨论留下的白板内容的图片。有时候，我在日常工作中获得了一些灵感，我会将它们都记录在 Markdown 文档中，它也是一个用于处置对于工作的各种评论的目录，如合作者对于论文内容的反馈。

src/是编写所有代码的位置。可运行的脚本都是直接写在 src/上的，类和实用程序写在了 codebase/上。下面我将详细说明一下（还有一个是脚本输出目录）。

编写代码



我现在写所有代码的工具都是 Edward，我发现它是快速实验现代概率模型和算法的最佳框架。

Edward 链接：<http://edwardlib.org/>

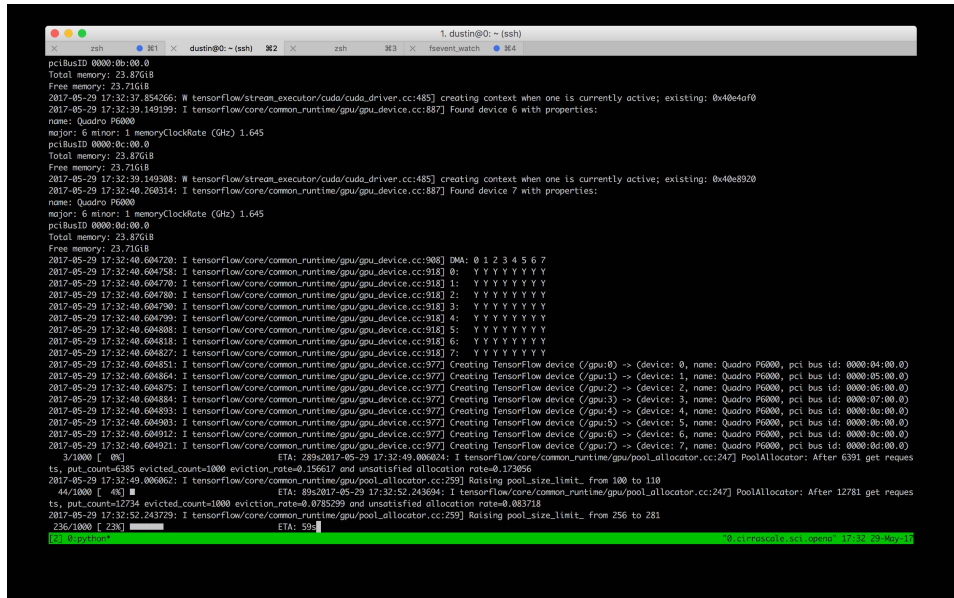
在概念层面上，Edward 的吸引力在于语言遵循数学：模型的生成过程被转化为每行 Edward 代码；随后希望写出的算法被转化为下一行……这种纯净的转换过程免去了在未来试图将代码拓展为自然研究问题时的麻烦：例如，在之前使用了不同的方法，或者调整了梯度估值，或尝试了不同的神经网络架构，或是在大数据集中应用了其他方法等等。

在实践层面上，我总是从 Edward 的现有模型示例（在 `edward/examples` 或 `edward/notebooks`）中受益，我将预置算法源代码（在 `edward/inferences`）作为一个新文件粘贴到我的项目中的 `codebase/` 目录中，然后进行调整。这样从零开始就变得非常简单了，我们也可以避免很多低级细节上的缺失。

在编写代码时，我一直遵循 PEP8（我特别喜欢 pep8 软件包：<https://pypi.python.org/pypi/pep8>），随后尝试从脚本共享的类和函数定义中分离每个脚本；前者被放在 `codebase/` 中以备导入。从第一步开始维护代码质量总是最好的选择，这个过程非常重要，因为项目会随着时间的不断膨胀，同时其他人也会逐渐加入。

Jupyter 记事本。许多人在使用 Jupyter 记事本（链接：<http://jupyter.org/>）用作可交互式代码开发的方法，它也是嵌入可视化和 LaTeX 的简单方法。对于我来说，我并没有将它整合到自己的工作流程中。我喜欢将自己所有的代码写入 Python 脚本中，然后运行脚本。但 Jupyter 等工具的交互性值得称赞。

实验管理



在好的工作站或云服务商做投资是必要的事。GPU 这样的特性基本上普遍可用，而我们应该有权限并行运行许多工作。

我在本地计算机完成脚本编写之后，我主要的工作流是：

1. 运行 rsync 同步我本地计算机的 Github Repository（包含未授权文档）到服务器的 directory。
2. ssh 到服务器。
3. 开始 tmux 并运行脚本。众事驳杂，tmux 能让你超脱此进程，从而不需要等待它的结束才与服务器再次交互。

在脚本可行之后，我开始用多个超参数配置钻研实验。这里有一个有帮助的工具 tf.flags，它使用命令行论证增强一个 Python 脚本，就像下面这样为你的脚本增加一些东西：

```
flags = tf.flagsflags.DEFINE_float('batch_size', 128, 'Minibatch during training')flags.DEFINE_fl
```

然后，你可以运行下面这样的终端命令：

```
python script1.py --batch_size=256 --lr=1e-4
```

这使得提交超参数更改的服务器任务变得容易。

最后，说到管理实验时输出的任务，回想一下前文中 src/目录的结构：

```
-- src/ -- checkpoints/ -- codebase/ -- log/ -- out/ -- script1.py -- script2.py
```

我们描述了每个脚本和 codebase/。其他三个目录被用于组织实验输出：

checkpoints/记录在训练中保存的模型参数。当算法每固定次数迭代时，使用 tf.train.Save r 来保存参数。这有助于维护长时间的实验——你可能会取消一些任务，后来又要恢复参数。每个实验的输出都会存储在 checkpoints/中的一个子目录下，如 20170524_192314_batch_size_25_lr_1e-4/。第一个数字是日期（YYYYMMDD），第二个是时间（HMS），其余的是超参数。

log/存储用于可视化学习的记录。每次实验都有属于自己的和 checkpoints/中对应的子目录。使用 Edward 的一个好处在于，对于日志，你可以简单地传递一个参数 inference.initialize (logdir='log/' + subdir)。被追踪的默认 TensorFlow 摘要可以用 TensorBoard 可视化。

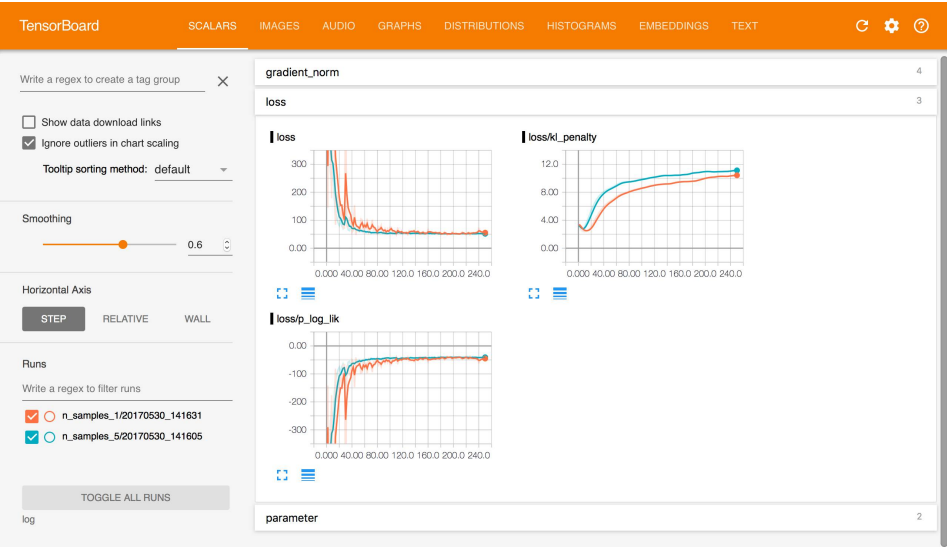
out/记录训练结束后的探索性输出；例如生成的图片或 matplotlib 图，每个实验都有自己的和 checkpoints/中对应的子目录。

软件容器。virtualenv 是管理 Python 安装环境的必备软件，可以减少安装 Python 的困难程度。如果你需要更强大的工具，Docker containers 可以满足你的需要。

Virtualenv 链接：<http://python-guide-pt-br.readthedocs.io/en/latest/dev/virtualenvs/>

Docker containers 链接：<https://www.docker.com/>

探索、调试和诊断



TensorBoard 是可视化和探索模型训练的一种优秀工具。因为 TensorBoard 具有良好的交互性，你会发现它非常易于使用，因为这意味着不需要配置大量 matplotlib 函数来了解训练。我们只需要在代码的 tensor 上加入 tf.summary。

Edward 默认记录了大量摘要，以便可视化训练迭代中损失的函数值、渐变和参数的变化。TensorBoard 还包括经过时间的比较，也为充分修饰的 TensorFlow 代码库提供了很好的计算图。对于无法只用 TensorBoard 进行诊断的棘手问题，我们可以在 out/目录中输出内容并检查这些结果。

调试错误信息。我的调试 workflow 非常糟糕。对此，我在代码中嵌入打印语句并通过消去过程来寻找错误。这种方法非常原始。虽然还没有尝试过，但我听说 TensorFlow 的 debugger 功能非常强大。

提升研究理解

不断考研你的模型与算法，通常，学习过程会让你对自己的研究和模型有更好的了解。这可以让你回到制图板上，重新思考自己所处的位置，寻求进一步提升的方法。如果方法指向成功，我们可以从简单的配置逐渐扩大规模，试图解决高维度的问题。

从更高层级上看，workflow 在本质上就是让科学方法应用到真实世界中。在实验过程中的每一次迭代里，抛弃主要想法都是不必要的。但另一方面，这一切的理论基础必须稳固。

在这个过程中，实验并不是孤立的。合作、与其他领域的专家沟通、阅读论文、基于短期以及长期角度考虑问题、参加学术会议都有助于拓宽你看待问题的思路并能帮助解决问题。

说明

本工作流主要用于实证研究，但其中的一些方法是值得其他任务参考的。

主文档结构的模板可以参考：<https://github.com/dustinvtran/latex-templates>

参考文献

1. Gelman, A., Shalizi, C. R. (2012). Philosophy and the practice of Bayesian statistics. British Journal of Mathematical and Statistical Psychology, 66(1), 8–38.

2. Pearl, J. (2000). Causality. Cambridge University Press.

3. Wainwright, M. J., Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning, 1(1–2), 1–305.




更多有关GMIS 2017大会的内容，请点击「[阅读原文](#)」查看机器之心官网 GMIS 专题↓↓↓



	0			
--	---	--	--	--

作者历史文章


机器之心「AI00」颁奖晚宴闭幕，五月最新榜单发布



机器之心GMIS2017全球机器智能峰会第二日晚，机器之心「AI00」首次举行了颁奖典礼。来自国内外的近20家上榜企业代表接受邀请并领奖。既包括英特尔、英伟达、[\[详细\]](#)

2017年 06月06日 06:15

苹果开发者大会WWDC 2017：首次全面展示苹果的人工智能实力



机器之心报道参与：李亚州、XavierMassa当地时间6月5日，苹果开发者年度盛会WWDC2017在美国加州举行。在这个舞台上，我们看到了苹果软件、硬件有哪些[\[详细\]](#)

2017年 06月06日 06:15

开源 | 浏览器上最快的DNN执行框架WebDNN：从基本特性到性能测评



选自Github机器之心编译参与：蒋思源、晏奇WebDNN是网页浏览器中最快的DNN执行框架，而本文首先简单介绍了WebDNN特征与其框架结构，即表明了为什么W[详细]

2017年 06月05日 13:15

学界 | CMU新研究试图统一深度生成模型：搭建GAN和VAE之间的桥梁



选自arXiv机器之心编译参与：吴攀不同的深度生成模型之间存在怎样的共性？近日，来自CMU和Petuum的四位研究者ZhitingHu、ZichaoYang、R[详细]

2017年 06月05日 13:15

资源 | 《人工智能与游戏》发行初版：从三个方面概述游戏人工智能（附下



选自gameaibook机器之心编译参与：黄小天近日，由马耳他大学副教授、情感计算专家GeorgiosN.Yannakakis和纽约大学副教授、人工智能与游戏研[详细]

2017年 06月05日 13:15

学界 | 深度学习算法全景图：从理论证明其正确性



选自arXiv机器之心编译参与：蒋思源、黄小天论文地址：
<https://arxiv.org/abs/1705.07038> 本论文通过理论分析深度神经网络群体风险（[详细]

2017年 06月04日 14:45

专访 | 英特尔AIPG数据科学主任 Yinyin Liu：英特尔更注重构建整体性端



机器之心原创作者：邱陆陆2016年起，英特尔在人工智能领域接连的大手笔收购引起了业界广泛关注。从Nervana到Movidius和Mobileye，这家半个世纪[详细]

2017年 06月04日 14:45

资源 | 企业应该怎样选择数据科学&机器学习平台？



选自kdnuggets机器之心编译参与：吴攀、黄小天、NurhachuNull一个弹性的数据科学平台（DataSciencePlatform）对于大型企业内的每[详细]

2017年 06月04日 14:45

教程 | 从硬件配置、软件安装到基准测试，1700美元深度学习机器构建指



选自Medium作者：Slav机器之心编译参与：QuantumCheese、LjLinjing、蒋思源在用了十年的MacBookAirs和云服务以后，我现在要搭[详细]

2017年 06月04日 14:45

ACL 2017 杰出论文公布，国内四篇论文入选（附解读）



机器之心报道参与：PaperWeekly、机器之心国际计算语言学协会(ACL ,
TheAssociationforComputationalLinguistics[详细]

2017年 06月04日 14:45

- 1
- 2
- 3
- 4
- 5
-
-