

AI前线

2017年12月刊

A I - F R O N T



关注落地技术，探寻AI应用场景



卷首语

人工智能的未来已来

作者 郭蕾

如月之恒，如日之升。从 1956 年在 Dartmouth 召开的学术会议开始，人工智能发展到今天已经走过了整整一个甲子，我想这中间已有数不清的浮沉和起落，也有很多说不清道不完的故事。而此时此刻，人工智能这位“花甲老人”却再一次以全新的面貌回归到大众视野，站在了浪潮之巅。

有人说，随着计算能力和数据能力的不断夯实，人工智能的发展将会迎来新的拐点。也有人说，人工智能就像是当年工业革命中，电灯对于人类的影响一样，必将深刻和深远。

回望过去的一年，不管是在国家层面，还是在媒体圈，抑或是产学界，人工智能都得到了前所未有的关注。甚至吴恩达还说，一百年前，电可以为很多企业、很多行业带来巨大的交通通讯和农业网络，今天人工智能也会为很多企业带来一样大的改变。他的这句话一直萦绕在我耳边，也让我更为直观的理解了人工智能的意义和价值。

作为一家媒体，从 2016 年开始我们就重点布局人工智能相关的内容，



AI 前线也是在这一时间诞生，到现在已经突破了 10 万的订阅用户。接下来我就从一个媒体编辑的视角来和大家聊聊我看到的人工智能发展趋势和风向。

软银董事长孙正义曾经说过人工智能未来将直接决定国家竞争力。中国政府绝对是全球最早关注人工智能发展的国家之一。从 2015 年开始，国务院以及相关部委就相继发布了多个人工智能相关的指导意见和行动实施方案，甚至在十九大报告中也有提及。12 月 14 日，工业和信息化部又发布了《促进新一代人工智能产业发展三年行动计划（2018–2020 年）》，这一计划更为具体，其中提到了重点落地领域和重点技术，也列出了三年目标，简单来说就是希望能够提速人工智能产业发展，并且做到世界领先。而在今年 7 月，国务院正式印发的《新一代人工智能发展规划》中，也明确指出了新一代人工智能发展三步走的战略目标，这基本就是国家在人工智能领域的规划蓝图。

从这些文件中不难看到，人工智能已经上升到了国家战略层面，不管

是人才培养还是产业建设，国家政策都极为支持。我在这里不细说，感兴趣的朋友可以到相关网站上仔细研读。

再谈谈人工智能的应用层面。前两天和一个老朋友聊天，他拍着我肩膀，激动地说：“不管是在哪个场景里，人工智能都能让你尝到甜头”。对于他的话，我深以为然。几年前，大数据还大红大紫的时候，很多公司其实就已经利用机器学习和自然语言处理等技术来智能化或者自动化自己的系统。到现在，我看到人工智能落地最多的场景还是客服、搜索和推荐，这也是大部分企业里最常遇到的场景了，也最容易见疗效。

单从技术的角度看，今年也有很多“传统技术”和人工智能结合的实践，我举个例子。AIOps 是今年的一个流行词，Gartner 的报告宣称，到 2020 年，将会有近 50% 的企业在他们的业务和 IT 运维方面采用 AIOps，可见其应用范围之广。人工智能和运维的结合，在阿里巴巴、Facebook 这些公司都已经得到了验证。用一句毫不客气的话来说，在这个数字的年代，任何使用传统技术来管理机器数据的组织要么忽略了信息的价值，要么已经让他们的运维团队不堪重负。在运维中落地人工智能，也是迟早的事。

另外，从云计算行业来看，国内外大型的云计算服务商都在努力叠加人工智能的能力。在刚刚结束的 re:Invent 大会上，AWS 就发布了几个 AI 相关的大杀器，相信接下来一年内各大追逐者基本也会沿着这个方向布局自己的 AI 产品。那人工智能和云计算之间是什么关系呢？马化腾说云是数字化升级的基础设施，而人工智能则是云上生长出来的前沿产品，“云 + 人工智能”未来或相当于“电 + 计算机”。沿着这个比喻往深想，你会发现确实很贴切，因为不就是有了计算机，才有了互联网时代吗？

人工智能不是万能良药，但我相信它是未来，而且现在“未来已来”。

助力人工智能落地

2018.01.13 – 01.14 · 北京国际会议中心

根据Gartner的预测，AI在2018年已经不是遥不可及的东西，每家公司都可以碰得到。到2020年，AI将成为CIO首要投资目标，深度神经网络跟机器学习会有100亿美元的市场。

那么，2018年，你是否已经做好准备转战AI了？应该去哪里学习现成的落地案例和实践经验呢？

InfoQ中国团队为大家梳理了目前AI领域的最新动态，并邀请到了来自Amazon、Snap、Etsy、BAT、360、小米、京东等40+公司AI技术负责人前来AiCon分享他们的机器学习落地实践经验，肯定可以给你一些启发和思考。

演讲嘉宾



颜水成

360人工智能研究院
院长及首席科学家



山世光

中科院智能信息处理重点实验室
常务副主任



喻友平

百度
AI技术生态部总经理



徐盈辉

菜鸟
人工智能部资深总监



洪亮劫

Etsy
数据科学主管



老师木

一流科技
创始人



王刚

小米
小爱语音交互系统负责人



林添

Google软件工程师
Tensorflow中国团队成员



张清

浪潮
AI首席架构师



杨建朝

Snap
研究院任主任科学家



胡时伟

第四范式
首席架构师



蔡超

Amazon
中国研发中心首席架构师

精彩案例 先睹为快

Amazon 机器学习在工程项目中的应用实践

第四范式 如何利用大规模机器学习技术解决问题并创造价值

菜鸟 双11：如何运用机器学习等AI技术实现物流优化

TutorABC 如何利用大数据和AI提升学习效果

小米 语音识别和NLP在智能音响中的实践

苏宁 智能机器人平台应用实践

爱奇艺 自然语言处理和视频大数据分析应用

淘宝 智能写手——智能文本生成在双十一的应用

售票倒计时进行中！

截至2018年1月12日前

团购享受更多优惠

售票咨询(电话)：18514549229 (同微信)

售票咨询(邮箱)：gouiao@geekbang.org



售票咨询



扫码关注大会官网



全球软件开发大会2018

主办方 **Geekbang** **InfoQ**
极客邦科技

[北京站]

北京·国际会议中心

演讲：2018年4月20–22日 培训：2018年4月18–19日

纵览20大热门专题

最低优惠7折进行时

现在报名每张立减2040元

移动开发实践 | 大数据与机器学习 | 大规模系统的性能优化

团购享受更多优惠 截至2017年12月31日

编程语言

Java前沿

前端前沿技术

大前端实践

新兴大数据处理技术

人工智能与业务实践

深度学习

运维新趋势

高可用架构

业务架构

微服务架构

大数据平台架构

大会官网：www.qconbeijing.com
访问官网获取更多前沿技术趋势

如有任何问题，欢迎咨询
电话：15110019061
微信：qcon-0410



AI 前线

InfoQ 中文站 AI 月刊 2017 年 12 月

生态评论

8 唐杉：2017，AI 芯片元年

落地实践

18 Netflix 推荐算法，让每个人看到不一样的电影海报

29 基于深度学习的 DGA 恶意域名分类算法

37 阿里巴巴年度技术总结：人工智能在搜索的应用和实践

推荐阅读

45 2017 年回顾：NLP、深度学习与大数据

62 ImageNet 冠军带你入门计算机视觉：卷积神经网络



唐杉：2017，AI 芯片元年

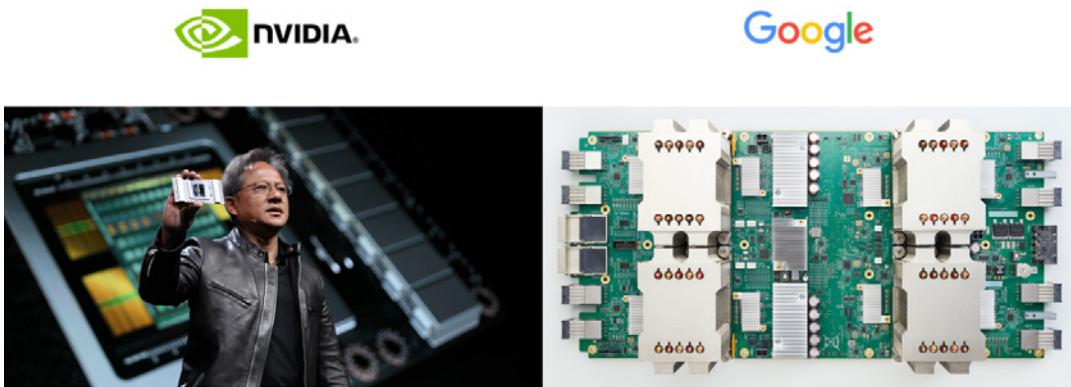
作者 | 唐杉

2017 年，AI 芯片是半导体产业的亮点，而它受到的关注又远远超出半导体的圈子。这一年，从科技巨头到初创公司，新老角色轮番登场，为我们上演了精彩好戏。若干年后，当我们再回头来看，一定可以把 2017 年作为 AI 芯片元年。

Goole vs Nvidia · 巨头之间的“错位战争”

四月初，Google 公布了一篇即将在 ISCA2017 上发表的论文：“In-DatcenterPerformance Analysis of a Tensor Processing Unit”。可以说正是这件“小事”，揭开了一部年度大戏的序幕，而它产生的深远影响甚至可能会持续到很多年之后。其实，在 2016 年 6 月的时候 Google 就透露了自己研发了一款在云端使用的专用 AI 芯片，TPU (Tensor Processing Unit)。Google 做 AI 芯片当然是吸引眼球的新闻，但苦于一直没有公布细节，大家也只能猜测和等待。因此，这篇普通的学术论文，得到了媒体的极大关

注。我也在第一时间写了一篇评论文章：“Google TPU 揭秘”，也是我的公众号阅读量最大的文章之一。对 TPU 高度关注的当然不只我们这些吃瓜群众，还有 AI 芯片领域绝对的统治者 Nvidia。后面就发生了黄教主和 Google 间关于 TPU 的 Benchmark 结果是否合理的口水战。而早在 2016 年 Google 透露 TPU 的时候，Nvidia 就多次表示它对 GPU 在 AI 运算上的统治地位没有什么威胁。



5月11日，Nvidia GTC2017大会，黄教主在Keynote上抛出了最新的GPU Volta (GV100)。Nvidia股票应声大涨，媒体也是大肆报道。AI芯片的焦点似乎又回到了Nvidia这一边。

除了公布了重量级的Volta，GTC上还有一个“小事件”，Nvidia宣布开源它的DeepLearning Accelerator (DLA)，9月正式公开。这个发布，在黄教主的Keynote中是一句话带过，但在业界引起的震动却一点也不小。“Nvidia为什么要搞开源？会开源什么东西？这个开源会不会影响众多初创公司的前景？”对这些问题的讨论一直延续到NVDLA真正开源之后。

没过多久，5月17日，在Google I/O大会上，Google公布了第二代TPU，用媒体的话说“…stole Nvidia's recent Volta GPU thunder…”。虽然TPU2的细节公布的并不多，但指标确实看起来很不错，而且具有非常好的可扩展性。唯一的遗憾就是它并不对外销售，只能以TPU Cloud的方式

供大家使用。

9月下旬，Jeff Dean 这位 Google 的软件大神参加了 HotChip 这个芯片界的重要会议，并在 Keynote “Recent Advances in Artificial Intelligence via Machine Learning and the Implications for Computer System Design”也亲自介绍了 TPU 和 TPU2 的情况，把它们作为新的计算生态中重要的一环。

9月底，NVDLA 在承诺的最后期限之前开源了 NVDLA 的部分硬件代码，同时公布了未来开源更多硬件和软件的路线图。这之后，大家对 NVDLA 也做了各种分析和讨论，试图把它玩起来。从目前来看，NVDLA 的开源好像并没有影响众多初创公司的融资。这个话题我们后面再说。至于 Nvidia 开源 DLA 的原因，官方的说法是让更多人可以更容易的实现 Inference，促进 AI 的推广，特别是在众多嵌入式设备上的应用。但从整个开源的过程来看，这个开源的决定似乎是比较仓促的。DLA 来自 Nvidia 自动驾驶 SoC 中的一个 module，最初并不是以开源 IP 为目的而设计的。而且 9 月的开源也只公开了一部分硬件代码和相应的验证环境，离真正能用起来也还是有较大差距。我们不好判断这个开源的决定是否和 Google TPU（在 Inference 上有比较大的优势）的强势亮相有关系。但基本的推测是，在 Deep Learning 中 Nvidia 的核心利益应该在于 Training（目前 GPU 还是 training 的最好平台）。让 Inference 门槛更低，渗透到更多应用，特别是 Edge 端，从而进一步促进 Training 的需求，应该是符合它的最大利益的。而且 NVDLA 的软件环境还是使用 Nvidia 的 CUDA/TensorRT，还是由 Nvidia 掌控的。

这场从一篇论文开始，几乎贯穿了 2017 年全年的 Google 和 Nvidia 的明争暗斗，对业界的影响可能要远远超过这两家公司本身。我之所以把它称为“错位”的战争，是因为它发生在 Google 这样的传统的软件巨头和 Nvidia 这样的芯片巨头之间。如果换成 Intel vs Nvidia，似乎是再正常不过的。Google 的参战，也许是开启了新的时代。我们可以看到，不仅是 TPU，Google 在 10 月又公布了他们在“Google Pixel 2”手机中使用的定制

SoC IPU (Image Processing Unit)。和 Apple 越来越多的自己定制芯片一样，Google 这样的科技巨头同样有应用（明确知道自己要什么），技术（对相关技术的多年积累），资源（不缺钱，不缺人）上的优势，定制自己的硬件，甚至芯片会变得常态化。同时我们也看到，Google TPU 的示范效应已经显现，更多的科技巨头加入 AI 加速硬件的竞争。Tesla 宣布自己定制自动驾驶芯片；Amazon，Microsoft，以及国内的 BAT，华为都在 Cloud 中提供专门的 FPGA 加速的支持；据称 Big Five 中还有在自己开发芯片的；BAT 也都在组建芯片设计的团队，等等。虽然大家具体的架构和实现方式不同，但都反映出对 AI 专用硬件的极大兴趣。相信未来这一趋势会越来越明显。

同时，传统的芯片巨头当然不会坐视这个巨大的市场被 Nvidia 主宰或者被 Google 们瓜分。Intel 连续收购了 Nervana（云），Movidius（端），Mobileye（自动驾驶），Altera（FPGA），又把 AMD 的 RajaKudori（GPU）招至帐下，甚至还搞了 Loihi（neuromorphic），可以说拿了一手好牌；虽然动作没有大家想象的那么快，但后面的发力还是值得期待的。AMD 也在努力追赶，毕竟他们的 CPU+GPU 有自己绝活，而整个公司也已经逐渐走出了低谷。而且，不管 Tesla 和 AMD 合作自动驾驶芯片的消息到底是真是假，芯片公司这种输出芯片设计能力的模式也是一种不错（或者无奈）的选择。

“以 Deep Learning 为代表的新型计算模式将引领未来芯片的发展方向”，这一观点基本已经是大家的一个共识。越来越多的玩家会关注能够支持新型计算的芯片，其中很多可能之前完全不在半导体这个圈子，也完全不了解芯片是怎么回事。2017 年我们不时能看到一些对比 CPU，GPU，FPGA 和 ASIC 架构的科普文章，甚至有 10W+ 的阅读量，不难看出大家的热情。

初创公司 • 长长的 list

2017 的 AI 芯片大戏中，主角不仅是巨头，初创公司也都粉墨登场，



戏份一点儿都不逊色。更重要的，在初创公司的“表演”中，中国公司不仅毫不怯场，而且非常出彩。我从 8 月份开始在 github 上维护一个 AI 芯片的列表，既包括大公司的产品，又包括初创公司的情况。到 12 月，这个列表中的信息越来越多，世界范围内的初创公司有 30 多家。而且这个列表还只包含了公开信息，还有很多公司处在 stealth 状态并没有收录。我也听到一个说法，在 AI 芯片领域的初创公司可能超过了 100 家，在 TSMC 排队投片也有 30 家。

不管在什么领域，初创公司都会面临很多风险和不确定性，也可能在成长过程中不断调整和变化。AI 芯片当然也不例外。我们看到，在这一年中，很多公司在不断成长，逐渐明确自己的方向和定位，走的越来越坚实。另一方面，从今年初创公司融资的情况来看，这个领域（也包括更大范围的 AI 概念）也明显出现一些泡沫。有些公司，在没有任何实际东西的情况下，就可以实现“PPT 融资”或者“Paper 融资”。有些公司，重心放在了 PR 上面，功夫都是做给 VC 看的，人称“2VC”公司。面对 AI 这个趋势性机会，有泡沫当然也是正常现象，只是希望这些泡沫不要伤害整个市场的发展。

抛开各种烟雾和泡沫，我们逐渐在这个领域初创公司也看到一些“龙

头企业”。比如国内的寒武纪、地平线、深鉴科技和比特大陆，都在 2017 年发布了自己的产品；美国的 Cerebras、Wave Computing、Graphcore 和 Groq（前 GoogleTPU 主要设计者创立），或有雄厚的实力，或有自己特色的技术和比较清晰的产品。在 2017 年，国内也出现一些依托应用开发芯片的 AI 初创公司，这些公司大多以应用牵头研发芯片。我也预期在 2018 年会看到更多这样的情况。当然，很多初创公司并没有公开自己的信息，不排除正在憋大招的可能性。

熟悉半导体产业的朋友可能比较清楚，半导体领域初创公司获得 VC 投资在之前是非常困难的。主要原因是这个产业风险大，门槛高，周期长。但 2017 年，AI 芯片的初创企业却受到了资金追捧。我们可以看看今年的一些公开的融资数据。寒武纪：1 亿美金（估值近 10 亿美金）；深鉴科技：4000 万美金；地平线：近亿美金；Cerabras：6000 万美金（估值 8.6 亿美金）；Graphcore：5000 万美金。在前面我也提到，当 Nvidia 宣布要开源 DLA 的时候，大家感觉会对初创公司的融资和估值有一定影响。但从结果来看，这种情况并没有出现。在 9 月之后，我们又看到很多初创公司成功融资。而投资者的热情似乎一点都没有减弱，只要有一个新的公司出现，立刻会有很多投资机构蜂拥而至。

为什么传统上不愿意碰半导体产业的投资者现在却对 AI 芯片趋之若鹜呢？这是一个有趣的问题。具体的原因可能有很多方面，整个 AI 领域的投资热潮应该是一个主要原因。如果观察这些投资背后的资本，可以看到很多本身就是 AI 领域很活跃的投资者，甚至本身就是把 AI 作为未来重点的科技巨头，比如 BAT。而传统的投资半导体领域的资金倒是比较谨慎一些。从这个角度来说，这些没有太多半导体背景的资本大量进入芯片领域，是会给大家带来新的机会和视野，还是带来风险和不确定性，还是有待观察的。另外，现在所说的 AI 芯片，一般是指 Deep Learning 加速芯片，相对来说，关键算法简单清晰，优化目标非常明确，很多技术（比如矩阵运算的硬件加速）已有多年的研究基础。而对这种硬件加速器的验证，测试和调试也相对容易。如果不进行精细的优化，硬件部分可以由一

个较小团队在较短时间完成。这些技术上的特征比较适合初创公司快速尝试。当然，做一个加速芯片（或者 IP）的硬件只是第一步。要真正做出能被市场接受的产品，则需要很多扎实的工作，产品定义，硬件效能，软件工具，系统测试，现场支持等等，一个短板也不能有。虽然大家都很关心投片的时间，但样片出来之后，脏活累活还多着呢。

2018 • 关注什么

对于 2018，我还是非常期待的。作为一名多年从事芯片架构设计多年的工程师，我首先期待看到一些技术上的创新。2017 年我写了不少分析 AI 芯片相关技术的文章，到年末几乎有点审美疲劳了（相信读者也是一样），似乎新鲜东西越来越少。在 2017 年底，有一个叫 Vathys 的初创公司，一下子开了好几个脑洞，全定制的 Asynchronous Logic，等效的时钟可以到 12GHz (28nm 工艺)；High-densitySRAM (1T-SRAM)，片上存储容量可以达到 1.5GB (28nm)；Wireless 3D Stacking，10,000GBit/S @ ~8 fJ/bit。这几项技术要么是目前还停留在学术研究阶段，要么是曾经昙花一现。一个初创公司一下就祭出这几个大招，又是这么高的指标，真有可能实现吗？所以，当 Vathys 的老板发邮件说应该把他们公司加到我做的 AI 芯片 List 里的时候，我开始是婉拒的。不过，换一个角度来看，即使是他们完全在忽悠，也算是击中了 Deep Learning 处理器的痛点。而且这几项技术目前也都有人在研究，在 AI 的热潮和巨大的资金支持下也许真能搞出来也说不定。所以，我还是希望看到他们或者是其它团队能够在这几项技术上取得突破，让我们真正激动一把。说到技术的突破，我们未来（可能要比 2018 年更远）还可以期待看到在存储技术上的突破，以及由新的存储技术带动的架构上的创新，包括 Neuromorphic 这条技术路线。

接下来，当然是巨头们的下一步动作。Google 的 TPU 是否会卖给自己之外的用户，直接和 Nvidia 展开竞争？目前 ONNX 阵营已经形成和 Google 的对峙，Google 作为生态最完整的厂商，推广 TPU 对巩固自

己的领先地位很有意义。Big Five 和 BAT 哪个会学习 Google 榜样直接自研芯片？阿里达摩院的芯片研究会不会从 AI 开始？Intel 能不能如大家所期待的全面爆发？Nvidia 会如何应对来自各方的挑战，是否会展开更专用的加速芯片，而不是仅仅在 GPU 中加个 Tensor Core？高通什么时候在手机芯片中加上硬件加速器？ARM 下一步会怎么走，会不会横扫嵌入端？。。。随便想想就会有很多值得期待的看点。最近我们也看到，为了对抗 Nvidia，AMD 和 Intel 竟然很罕见的宣布合作。而 IBM 在 Power9 上和 Nvidia 深度合作。2018 年也许我们还能看到业界巨头间更多的合纵连横。

初创公司的命运也是 2018 年最大的看点。我在之前的一篇文章中说过“对于 AI 芯片的 startup 来说，2018 年就算不是毕业大考，也至少到了学期末考试了…”。2018 年，大部分初创公司都将会交出第一次测验的结果（芯片），也会开始小批量的试用。相信到时会有比较公平的 Benchmarking 结果出现，“理论上”的指标会被实际的“跑分”结果取代。虽然对于初创公司来说，犯错误是可以容忍的，第一代芯片也不能完全代表公司未来的前景。但是，做芯片需要巨大资源的持续支持，这个阶段掉队可能非常危险。当然，第一次的淘汰对于真正优秀的企业也是最好的机会。我非常期待看到能够在考试中脱颖而出，并跨上新的台阶（或者直接毕业）的同学；或者，会有我们不认识的面孔，突然惊艳出场。另外，2018 年，在 Edge 端会有更多的传统芯片厂商加入竞争，三星，高通，MTK，展讯等等；而在嵌入端 IP 上有绝对优势的 ARM 应该也会有更大的动作，这些都可能会对初创公司的命运产生重大影响。

最后，是变局的可能。从整体上来讲，AI 整体上在 2018 年会怎么发展是一个大家都非常关注的问题。继续高速增长，还是平稳发展，又或者会遇到问题高开低走？不管是哪种情况，AI 芯片必然会受到大势的影响。比较特殊的是，芯片研发的周期大约在 9 到 18 个月左右，这比软件应用的开发和更新周期要长的多。再加上一些滞后效应，芯片的发展很难和算法和应用的发展节奏同步。芯片开发中一个比较可怕的问题就是未来

的不确定性。相对来说，一个可预期的平稳增长的环境是最有利于芯片研发的，可以让芯片设计者能够更好的规划产品和协调资源。另一种变局情况是，算法层面发生巨大的变化，也就是技术上的不确定性。这几年最成功的 AI 算法就是基于神经网络的深度学习。这正是目前 AI 芯片在需求上的基础，也决定了现在大部分 AI 芯片都是以加速这一类算法为目标的。如果基本算法需求发生变化，会对 AI 芯片的设计产生很大的影响。比如，目前已经有一定应用基础的低精度网络，也就是在 inference 中使用非常低的精度，甚至直接使用二值网络。如果这种 Inference 得到广泛应用，现在的芯片架构则可能得要重新考虑。再比如，如果 Hinton 大神的 capsule networks 得到实用，也可能会需要新的芯片架构来支持。毕竟 AI 领域现在发展很快，所以大家也都必须要时刻盯着应用和算法层面最新的进展。我们也要随时间自己下面的问题（来自 Jeff Dean 在 NIPS2017 的演讲）。

If you start an ASIC machine learning accelerator
design today, ...

Starts to get deployed into production in ~2 years

Must remain relevant through ~5 years from now

**Can We See The Future Clearly Enough?
What should we bet on?**

总结

2017 年马上就要过去，在这几年相对“平淡”的半导体领域，AI 芯片让我们小激动了一下。其实可聊的事情很多，以上文字基本上是想到哪写到哪，也都是个人一点点感想，准确的地方，还请各位多多指正，多多包含。

祝各位读者 2018 年万事如意！更要祝各位奋战在 AI 芯片第一线的各位同仁获得成功！

AI 前线声明 | 本文系唐杉博士原创文章，已经授权 InfoQ 公众号转发传播。

作者简介

唐杉博士具有超过 15 年的芯片设计、专用处理器设计和 SoC 架构设计经验，现在 Synopsys AI Lab 负责 AI 芯片架构和相关技术的研究。欢迎关注唐杉博士的公众号 StarryHeavensAbove。



Netflix 推荐算法， 让每个人看到不一样的电影海报

作者 | Ashok Chandrashekhar 等

译者 | Debra

不久前，Netflix 推出交错测试个性化推荐算法，计算速度提高 100 倍秒杀 A/B 测试的消息引起了不小的轰动。而仅一周后，这家视频网站宣布了他们利用情境 bandits 推荐算法，实现了视频配图的个性化处理。

多年来，Netflix 个性化推荐系统的主要目标，是为用户在合适的时间推荐合适的视频。Nteflix 网站上每个分类页面下有成千上万部影片，用户账号达数十亿，为每个用户推荐最合适的视频是头等要事。但推荐系统能做到的不仅是这些。怎样让用户对你推荐的视频感兴趣？怎样让一个陌生的视频激起用户的兴趣？什么样的视频值得关注？回答这些问题对于帮助用户发现好的内容至关重要，特别是对于不熟悉的视频。

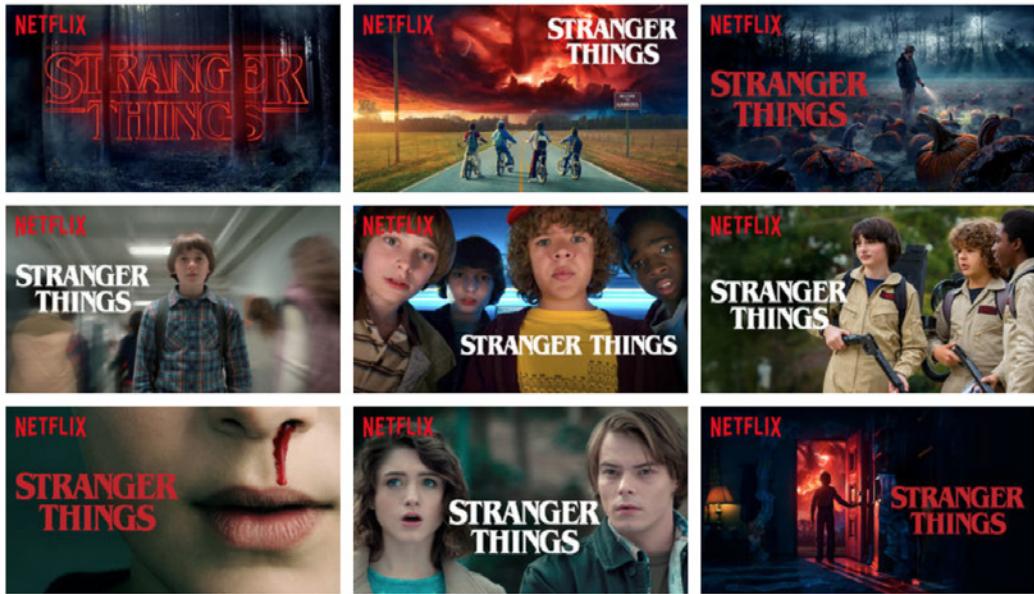
用来描述视频的配图或图像，是可以轻松地解决这个问题的方法之一。如果一张配图对用户有足够的吸引力，比如用户熟悉的演员、让人肾上腺激素飙升的汽车追逐场面，或者一部电影或电视节目精髓的戏剧性场景等信息（一张图片胜过千言万语），就会诱惑用户点开视频。这是 Netflix 与传统媒体产品不同的一点：我们的产品可能超过一亿种，为每个用户提供个性化推荐和个性化的视觉效果。



没有配图的 Netflix 主页

之前，我们讨论过如何做到为所有会员的视频匹配最合适的图片。通过多臂老虎机算法，我们可以为视频找到最合适的配图，以《怪奇物语》为例，这部影片获得了最高用户播放率。但是，鉴于用户的品味和偏好存在巨大差异，如果我们能够找到每个用户偏好的点，并在配图中能呈现出他们最感兴趣的东西，效果不是更好吗？

我们探讨一下配图个性化在哪些场景下具有重要意义。例如，每个用户有不同的观看历史，下图左是三个用户过去看过的视频，箭头右侧是我们为会员推荐的颇受欢迎的电影。



为《怪奇物语》设计的配图，不同的图像涵盖了节目中的不同主题

我们为电影《心灵捕手》设计个性化配图的根据是每个用户对不同类型和主题的偏好。对于看过许多浪漫爱情电影的人，如果他的推荐图片中包含马特·达蒙（Matt Damon）和米妮·司各德（Minnie Driver）的信息，可能他会对比《心灵捕手》感兴趣，而如果是对于看过很多喜剧片的用户，我们在推荐图中包含知名喜剧演员罗宾·威廉斯（Robin Williams）的信息，吸引他的几率可能更大。



另外，个性化配图对喜欢不同演员的用户会产生什么影响呢？以《低俗小说》为例，一位观看过很多乌玛·瑟曼（Uma Thurman）出演电影的用户可能会对包含乌玛（Uma）信息的图片反应更为积极。同理，John

Travolta 的粉丝更可能因为图像中包含 John 而被这部电影吸引。



当然，并不是所有的配图个性化场景都是这么明了的。所以我们并没有穷举这些规则，而是依靠数据来告诉我们应该使用什么图片。总体而言，通过配图个性化处理，我们可以帮助提高每个用户的体验。

克服重重挑战

Netflix 还通过算法对网站做了很多个性化处理，以提高会员体验，包括主页列表选择、列表的标题、展示的图片、发送的消息等等。对于我们来说，每一个方面的个性化处理都是独特的挑战，个性化配图也不例外。其中，图像个性化处理的挑战之一，是每个位置视频的配图只能有一张。相比之下，典型的推荐设置可以向会员提供多个选择，之后我们可以从会员的选择中了解他们的偏好。这意味着图像选择是一个在闭环中操作的鸡与鸡蛋问题：会员选择播放哪个视频的根据只有图片。这就导致一个问题：当我们推出个性化图片时，会不会影响成员播放（或不播放）视频，以及什么情况下是不管我们放了哪张图片，用户仍会播放视频（或不播放）。因此，个性化配图推荐应该结合传统方法与算法才能奏效。当然，为了正确学习配图个性化，我们需要收集大量的数据，来找到能表明哪个配图对于用户更合适的信息。

另一个挑战，是要理解配图变化所产生的影响，是否会降低视频的可识别性，让视频在视觉上难以重新找到？例如，会员之前感兴趣但至今还没有注意到的视频，或者，配图改变是否会让用户改变想法。如果我们找

到更好的图片呈现给会员并不断更换图片，会让会员感到迷惑。另外，改变图像也会引起归因问题，因为我们不清楚究竟是哪张图像引起了会员对视频的兴趣。

接下来，是要理解配图如何与同一个页面或者阶段选择的其他配图进行合理关联。也许主角的大胆特写非常适用于页面上的视频配图，因为与其他作品相比，它显得非常突出。但是，如果整个页面的配图都是这一类型，那么它的效果反而会大打折扣。因此，孤立地看每一幅图片可能还不够，我们需要思考如何在整个页面使用多样化的图像。配图的效果可能还取决于图片之外其他的因素（例如简介、预告片等）。所以，我们的图片选择应该多样化，让每个视频之间都能形成互补。

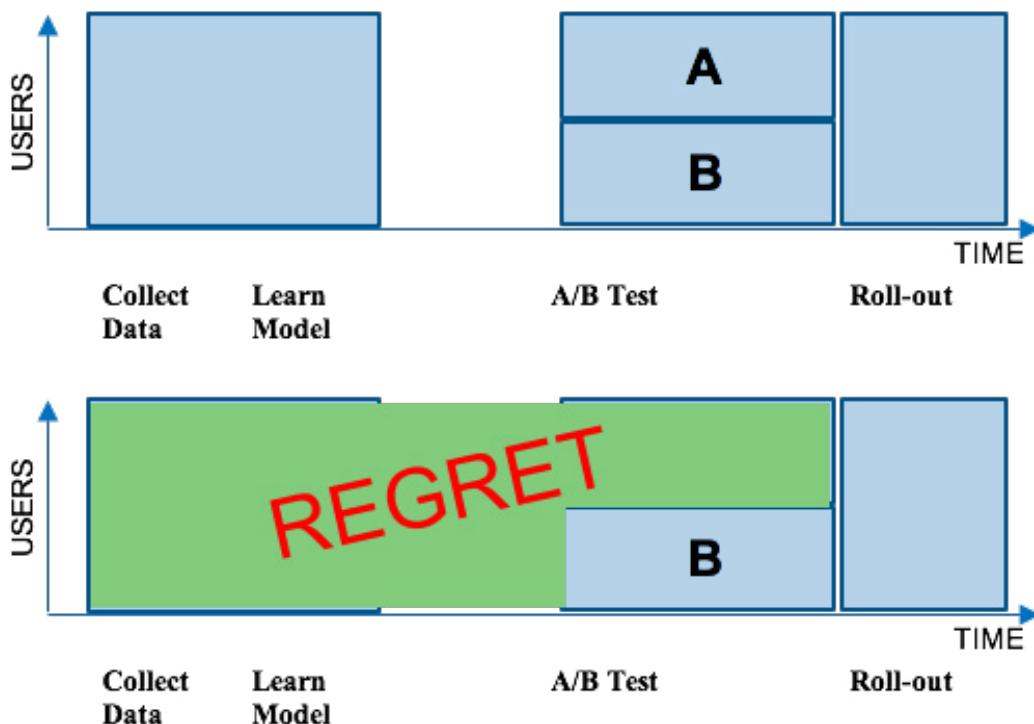
为了实现有效的个性化，我们还需要为每个视频提供优质的作品库。这意味着我们需要多个库存，并且每个库存的图片都是非常有吸引力、信息丰富且与视频契合，但要避免那种“标题诱饵”式的图片。视频的图像集也需要足够多样化，以涵盖对内容的不同角度感兴趣的广大潜在观众。毕竟，一张配图的信息量取决于看到它的个体。因此，我们的配图不仅需要突出视频中的不同主题，还要突出不同的美学。

最后，是大规模个性化配图面临的工程挑战。由于我们的会员体验是视觉化的，包含大量的图像，因此，系统在峰值时需要每秒处理超过 2000 万个低延迟请求。这个系统必须足够强大，因为用户界面不能正确渲染图稿，用户体验会显著下降。而且，个性化算法还需要在视频上传时做出快速响应，这意味着要在冷启动的情况下快速学习个性化。启动后，该算法必须不断进行调试，因为配图的效果可能会随着时间的推移而变化，视频的生命周期不断演变，而且会员的品味也在不断变化。

情境 bandits 推荐个性化配图

Netflix 的大部分推荐引擎都采用机器学习算法。首先，我们会收集一批关于会员如何使用服务的数据，然后在这批数据上运行一个新的机器学习算法。接下来，我们对这种算法在现有生产系统上进行 A / B 测

试。通过在随机子集上进行 A / B 测试，我们了解到新算法是否比现有的生产系统更好。A 组会员代表当前的产品体验，而 B 组代表新算法下的产品体验。如果 B 组中的会员对 Netflix 的参与度更高，那么我们将把这个新算法推广到整个会员群体。不幸的是，这种批处理方式也有缺憾（regret）：许多会员长期以来并没有更好的用户体验，如下图所示：



为了减小这个缺憾，我们放弃了批处理机器学习，而使用在线机器学习。对于图片个性化，我们使用的在线学习框架是情境 bandits (contextual bandits)。情境 bandits 并不是收集整批的数据，进行学习模型训练，直到 A / B 测试结束，而是可以迅速为每个会员找到最合适的个性化图片。简而言之，情境 bandits 是一类在线学习算法，这种算法可以在学习无偏差模型所需的训练数据成本，和将学习模型应用于每个会员的好处之间进行权衡。我们使用非情境 bandits 进行非个性化图像选择，找到不考虑情境的最佳图像。而对于个性化推荐，每个会员均代表不同的情境，因为我们预计不同的会员会对图像做出不同的反应。

情境 bandits 的一个重要属性，是其是为尽量减小缺憾而设计的。在高层次上，我们通过在学习模型的预测中输入受控随机化来获得情境 bandits 的训练数据。随机化方案的复杂性可以从简单的具有均匀随机性的 epsilon-greedy 公式，到随着模型不确定性而自适应地改变随机化程度的闭环方案。我们将这个过程称为数据探索（data exploration）。进行这样的探索，我们需要记录每个配图选择的随机化信息。这种日志记录让我们可以纠正走偏的选择倾向，从而以稍后所述的不偏颇的方式执行离线模型评估。

由于我们可能不会采用情境 bandits 算法预测的最佳图像，所以数据探索可能会产生成本（或缺憾）。这种随机性对会员体验（以及我们的指标）有什么影响呢？我们有超过一亿的会员，通常情况下，探索带来的缺憾非常小，分摊到庞大的会员基数上，每个会员都会为记录提供一部分反馈。这使得每个成员的探索成本可以忽略不计，这也是起码选择情境 bandits 改善会员体验的重要因素。如果探索成本很高，那么使用情境 bandits 进行随机化和数据探索就不太合适。根据我们的在线数据探索方案，不管视频是否被播放，我们都会获得一个记录每个（会员、标题、图像）元组的训练数据集。此外，我们可以控制探索，使图像选择不会经常变化，这使得会员对特定图片的参与度更加清晰。

模型训练

在在线学习中，我们训练情境 bandits 模型根据情境为每个会员选择最合适的图片。通常每个视频最多有几十张候选图片，为了训练选择模型，我们为每个会员的图片进行排名来简化问题。简化之后，我们仍然可以找到会员对视频图像的偏好，因为呈献给用户的每个候选图像，有一部分会引起用户的参与，而另一部分则不会。我们可以对这些偏好进行建模和预测，会员享受高质量参与度的概率会相应提高。这样的模型可以是监督式学习，也可以是汤普森抽样（Thompson Sampling）情境 bandits、LinUCB 或贝叶斯方法（Bayesian）。

潜在的信息

在情境 bandits 中，情境通常表示为模型输入提供的特征向量。我们可以使用许多信息作为特征，尤其是会员的许多属性：他们播放的视频、视频类型、会员对特定视频的参与度、国籍、语言偏好、使用设备、时间等。

另外一个重要的考虑因素，是候选池中一些图片优于其他图片。我们观察数据探索中所有图像的总体转换率（take rates），即高质量播放次数除以印象数量。以前做非个性化图像选择时，我们仅根据总体转换率之间的差异来决定为用户批量选择的最佳图像。而在我们新的情境 bandits 个性化模型中，整体转换了仍然是重要的，并且个性化推荐仍会与非个性化图像排名有一定重合。

图像选择

为会员提供合适图像，实际上是一个从与视频匹配的可用图像池中找到最佳候选图像的选择性问题。模型经过上述训练后，我们用它来对每个情境的图像进行排序，并预测为会员推荐图像会引发播放的概率。我们按这些概率对候选图像集进行排序，并选择出概率最高的图像。

效果评估

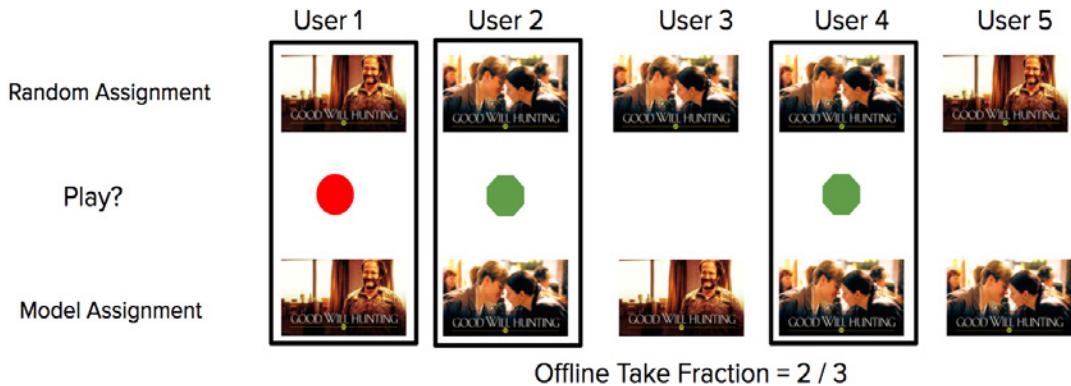
离线

在线上部署之前，我们可以使用一种称为“重播”的离线技术 [1] 对情境 bandits 算法进行评估。这种方法让我们可以根据记录的探索数据来回答反事实问题。换句话说，如果我们在同等条件下使用不同的算法，在不同情境下在线下会发生什么。

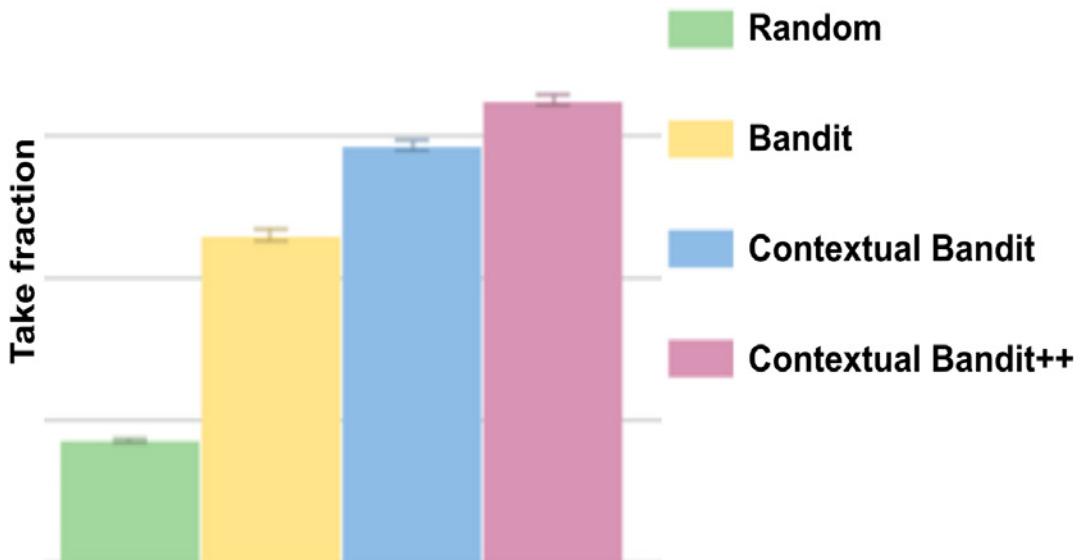
如果我们假设提供的图像是通过新算法选择的，而不是现用的算法，则重播显示出会员对视频的参与度。图 2 显示了与随机选择或非情境 bandits 相比，情境 bandits 如何提高记录中用户的平均参与率。

如下图：根据记录的数据计算重播率的简单示例。为每个成员分配一个随机图像（第一行），系统记录了视频印象以及用户播放了视频（绿

色圆圈) 或没有 (红色圆圈)。通过匹配随机分配和模型分配重合的部分 (黑色方块)，计算该子集的分数来计算新模型的重播指数。



如下图：基于图像探索数据记录中重播率，不同算法选择的图像平均分数（越高越好）。随机（绿色）表示随机选择图像，简单的 Bandit 算法（黄色）选择具有最高分数的图像。情境 bandits 算法（蓝色和粉红色）根据情境为不同的成员选择不同的图像。



如下图：根据用户个人资料进行的情境图像选择示例。Comedy 指主要观看喜剧片的个人资料，Romance 代表看爱情片最多的用户个人资料。情境 bandits 算法为更喜欢喜剧片的会员推荐了带有著名喜剧演员罗宾·威廉姆斯 (Robin Williams) 形象，同时更为浪漫的情侣接吻图片。

Profile Type	Score Image A	Score Image B
Comedy	5.7	6.3
Romance	7.2	6.5



Image A

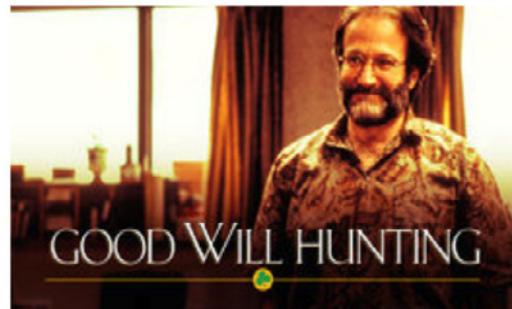


Image B

在线

经过对多种离线模型进行试验之后，我们找到了可以提高重播率的模型，最后进行 A / B 测试，以对个性化情境 bandits 与非个性化 bandits 进行比较。正如我们所料，个性化对核心指标提高起到了重大的作用。我们也看到了线下测量重播率与线上模型之间的合理性关联。在线结果还发现了有趣的现象，例如，在会员之前没有参与的视频，个性化的改善效果更好。这不无理由，因为我们更希望这个算法对用户并不熟悉的视频发挥更大的作用。

结论

现在，我们已经迈出了第一步，在个性化图片推荐和其他服务中采用了这种方法。这改进了用户发现新内容的方法，有史以来，我们不仅对推荐内容进行了个性化，而且对推荐的方式也进行了个性化。但是，这个方法还有很多可以改进的地方，应用的范围也可以进一步扩大，包括通过计算机视觉技术开发能以最快的速度对图像和视频进行个性化处理的算法冷

启动等。另一个机会是可以将这种个性化方法扩展到我们使用的其他类型的配图以及其他视频描述语，例如概要、元数据和预告片中。



基于深度学习的 DGA 恶意域名分类算法

作者 | 曾凤

本文整理自瀚思科技曾凤在上海ICAMIT201上的演讲《基于深度学习的DGA恶意域名分类算法》。瀚思科技是中国第一家大数据安全公司。

一、背景说明

僵尸网络（BotNet）是指采用一种或多种传播手段，将大量僵尸主机（Bot）感染病毒，从而在主控者（Botmaster）和被感染主机之间，通过命令与控制服务器（Command and Control Server, C2 Server），形成的一个可一对多控制的网络。目的是尽可能地感染更多的机器。可以看出，不论是对网络安全运行还是用户数据安全的保护来说，僵尸网络都是极具威胁的隐患。

目前，攻击者操纵僵尸网络通常会使用多个域名的方式来连接至C2服务器，从而达到操控受害者机器的目的。这些域名通常会被编码在恶意程序中，这也使得攻击者具有了很大的灵活性，他们可以轻松地更改这

些域名以及IP。该连接方式最大的优势是用极为简单的代码便可实现，劣势是其极易被政府检测。所以就有了域名生成算法（Domain Generation Algorithms，DGA），通过DGA，攻击者可以在短时间内自动产生成千上万的域名，这样就可有效地避开黑名单列表以及政府的检测。

二、域名生成算法（简称 DGA）介绍以及技术挑战

DGA到底是什么？简而言之，其通过输入一些种子，包含字符串、数字以及日期，利用加密算法，比如异或操作等，从而产生一系列的伪随机字符创，即域名。例如前段时间风靡全球的勒索病毒Cryptolocker，它以邮件附件形式分发，感染计算机并加密近百种格式文件（包括电子表格、数据库、图片等），从而对用户进行勒索。我们对其输入一个日期2014年2月7号，采用DGA算法会产生新的[域名](#)。

安全人员可以通过收集样本以及对DGA进行逆向，来预测哪些域将来会被生成和预注册并将它们列入黑名单中。但DGA可以在短时间内生成成千上万的域，这是一个相当庞大的数量，我们不可能每天都重复收集和更新我们的列表。

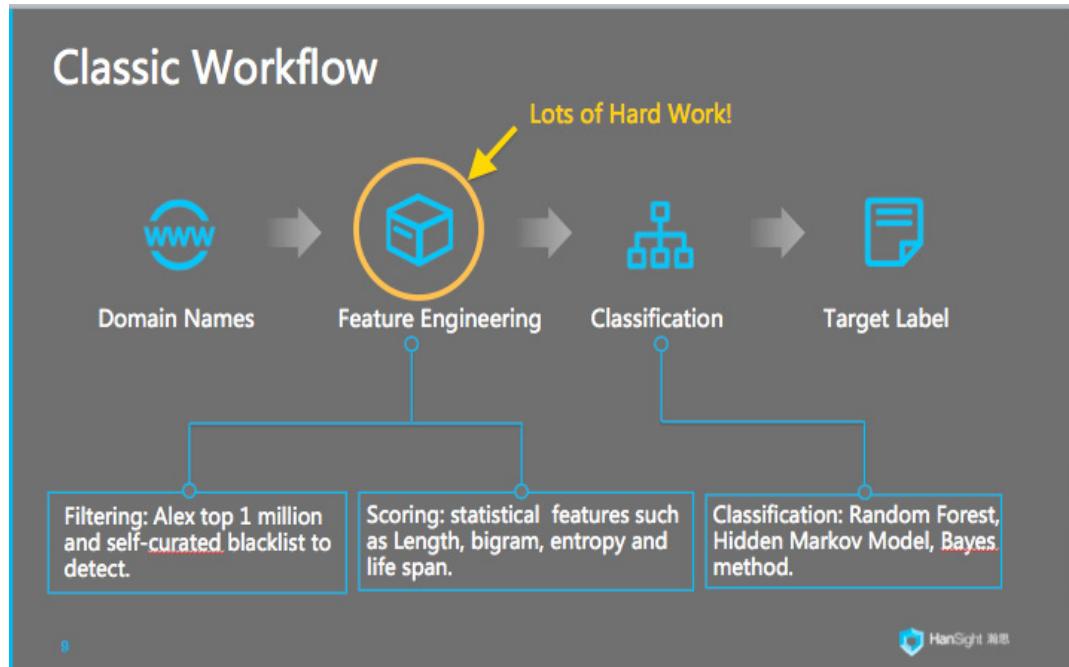
所以我们需要实现的是对DGA产生的恶意域名进行实时检测。

为实现该目标，首先需要了解当前流行的检测方法以及所面临的技术难点。经典的检测技术主要分为两个阶段，特征工程和分类算法。如下图所示，特征工程主要从两个方面入手：

1. 基于过滤的方法，采用Alex前100万个网站和黑名单对域名进行检测；
2. 基于统计特征的方法，例如长度、二元语法、信息熵和生存周期等。

在这之后，需要对采取的特征进行分类，这其中常见的机器学习算法有随机森林、隐马尔科夫模型、贝叶斯方法等。

在整个检测过程中，特征工作最为繁琐，且以上工作流有如下的缺点：



1. 过度依赖人工特征工程，较难实现；
2. 偏低的检测率以及偏高误报率；
3. 速度慢，不能实时检测。

那么，我们该如何解决上述挑战？又为何选择深度学习？

因为深度学习有如下的优点：

1. 学习特征可自动提取特征，完全脱离人工特征；
2. 较高的检测率以及较低的误报率；
3. 速度快，能并行处理大规模数据。

三、深度学习简介

深度学习究竟是什么？

深度学习的概念源于人工神经网络的研究，是机器学习研究中的一个新的领域，其动机在于建立、模拟人脑进行分析学习的神经网络。它通过组合低层特征形成更加抽象的高层表示属性类别或特征，以发现数据的分布式特征表示。

假设有n个点 $((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$ ，我们需要

What's Deep Learning?

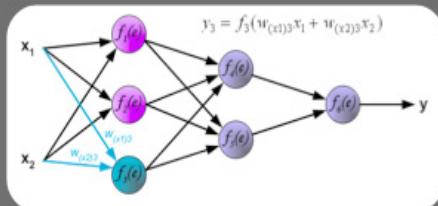


Figure 1. three-layer neural network.

- Given a set of values $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Assume that the true model $y = w^T x$
- Find optimal w to minimize the following quantity

$$\arg \min_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

12

HanSight 漫思

要用一个模型去拟合这n个点，假设这个模型为，

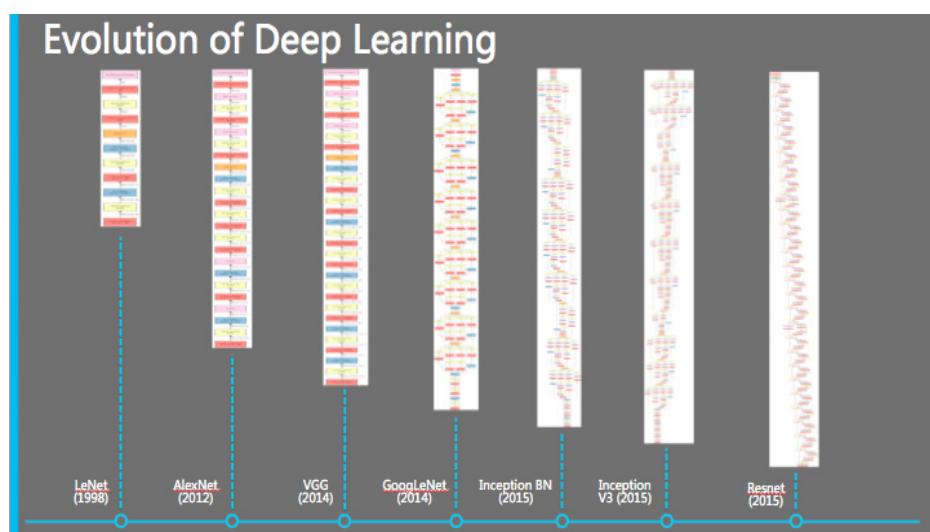
$$y = w^T x$$

则目标函数为

$$\arg \min_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

即寻求最优的权重w使得与y值之间的距离最小。

下面来看当前比较优秀的几个深度学习网络。



上图展示了从1998年到2015年深度学习的一个发展状况，这些网络，例如LeNet, AlexNet, VGG, Inception网络已经在计算机视觉、语音识别、图像分类以及自然语言处理等方面取得了重大成功。如大家所见，这些网络变得越来越复杂，越来越深。那为什么会越来越深？其中最主要的原因在于，越深的网络意味着能更多的非线性函数，即它能从越复杂的函数中提取筛选出更有用的特征表示。

值得一提的是，基于ImageNet数据集已经有了很多优秀的训练好的深度学习模型。Imagenet数据集对深度学习的浪潮起了巨大的推动作用，它有1400多万幅图片，涵盖2万多个类别，其中有超过百万的图片有明确的类别标注和图像中物体位置的标注。

四、我们的模型：基于词嵌入以及迁移学习

那么如何将ImageNet训练好的深度学习模型应用到恶意域名检测问题上？这其中两个难点在于

1. 我们需要分类的是域名（字符类型），它在内容上有别于ImageNet中的图片；
2. 我们需要处理的是百万级的域名数据，这训练起来非常耗时。

针对难点1，我们可采用词嵌入方法，将字符类型的域名转换成图片；

针对难点2，我们期望跳过训练模型这一步，直接将已有的训练好的ImageNe模型运用到域名中进行检测，这需要迁移学习的理论。

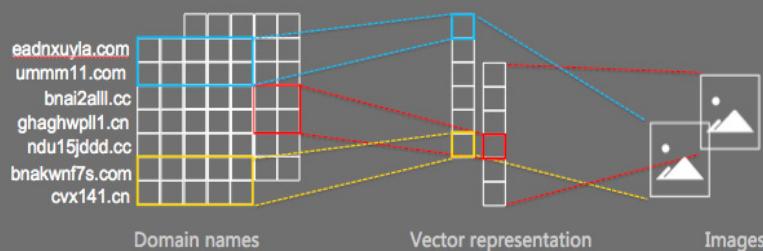
接下来我将分别解释词嵌入和迁移学习。首先来看看什么是词嵌入。

词嵌入是自然语言处理中的名词。从数学上定义为一个映射：从文档空间投影到一个低维的数字型向量空间（一般用的维度可以是几十到几千）。该映射为一个单射函数，即每个Y只有唯一的X对应，反之亦然。它能够将文档进行数值化处理，从而将文档分析问题转化成相对应的数值向量（或者矩阵）问题。

它主要有如下的两个优点：

What's Word Embedding?

a mapping from the **vocabulary** domain to a **low-dimensional vector** representation of real numbers.



- Advantages:**
1. Dimension **Reduction** - it is a more efficient representation.
 2. Contextual **Similarity** - it is a more expressive representation.

1. 降维—更为有效的表征；
2. 文本相似度—更为相近的表征。

接下来我将介绍迁移学习。如下图，左边我们有源数据、源模型以及源标签；右边是目标数据（域名）、目标模型以及目标标签；中间则是“迁移学习”，它将源模型和目标模型连接起来。源模型（比如基于ImageNet训练好的深度学习模型）通过训练得到权重，我们将这些“学好的”权重迁移到我们的新数据集----域名，这就是著名的“迁移学习”。

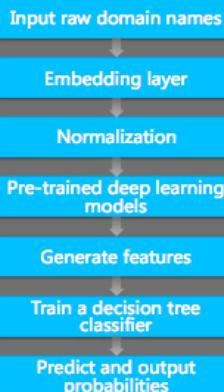
What's Transfer Learning?



到此，我已经介绍完所有需要用到的预备理论。接下来我将为大家介绍，我们如何将这些理论应用到DGA恶意域名检测中的。

1. 对原始域名，采用词嵌入对其数值化；
2. 对1中所得到的向量进行归一化；
3. 输入到已训练好的ImageNet模型，并提取倒数第三层作为特征；
4. 利用特征训练决策树模型，从而达到分类和预测的目的。

Proposed Model



1. Use **embedding layer** to turn domains into vector representation of real numbers.
2. Need **Normalization** to make vector representation more normal or regular.
3. Extract the third layer from last as **features**.
4. Take features to train a **decision tree classifier** for binary-label and prediction.

Experimental Results

Architectures	True Positive Rates	False Positive Rates	Accuracy
AlexNet	0.967086	0.02391	0.97231
VGG16	0.97819	0.02125	0.97296
VGG19	0.97258	0.01714	0.97039
SqueezeNet	0.97461	0.01942	0.97198
Inception-BN-21k	0.97882	0.01831	0.97596
Inception-BN-1k	0.98519	0.0161	0.98196
Inception V4	0.99863	0.01128	0.98568
ResNet152	0.99317	0.01659	0.98273

The best performance is by using **Inception V4**, which achieved **99.86%** true positive rates.

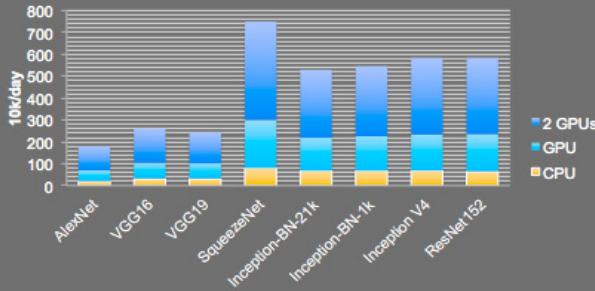
Table 1. True Positive Rates, False Positive Rates and accuracy for Binary Classifiers.

五、实验结果

表1总结了真阳性率，假阳性率和准确性，从这张表可以看出，最好的模型是Inception V4，它能达到99.86%的假阳性率。

图一展示的是不同模型基于CPU和GPU的性能表现。如大家所见，最

Experimental Results



- Speed : 2GPUs > GPU > CPU
- The **fastest** architecture is **SqueezeNet** that can handle almost **five million** domain names per day under 2 GPUs.

Figure 1. Performance of Different Architectures on CPU and GPU for Binary Classifiers.

21

HanSight 漏洞

快的模型为SqueezeNet。当采用只用一个CPU运行时，每天只能处理不到100万的数据；当采用1个GPU运行时，每天能处理200多万的数据；当采用2个GPU运行时，每天能处理大约500的数据量。



阿里巴巴年度技术总结： 人工智能在搜索的应用和实践

作者 | 欧文武

以深度学习为代表的人工智能在图像、语音和NLP领域带来了突破性的进展，在信息检索和个性化领域近几年也有不少公开文献，比如wide & deep实现了深度模型和浅层模型的结合，dssm用于计算语义相关性，deepfm增加了特征组合的能力，deep CF用深度学习实现协同过滤，rnn recommender采用行为序列预估实现个性化推荐等。工业级的信息检索或个性化系统是一个复杂的系统工程，深度学习的工业级应用需要具备三个条件：强大的系统计算能力，优秀的模型设计能力和合适的应用场景，我们梳理了过去一年多搜索在深度学习方向上的探索，概要的介绍了我们在深度学习系统、深度学习算法和搜索应用落地的进展和思考，希望对大家有所启发。

深度学习在搜索的应用概括起来包括 4 个方面：

首先是系统，强大的深度学习训练平台和在线预测系统是深度学习应

用的必要条件，目前我们的离线深度学习框架、在线深度学习框架和在线预测框架统一到tf，并实现了日志处理，特征抽取，模型训练和在线服务部署端到端的流程，极大提升了算法迭代效率；

其次是搜索应用，包括智能交互，语义搜索，智能匹配和智能决策四个技术方向，这四个方向的协同创新实现了搜索全链路的深度学习技术升级，并具备从传统的单场景单目标优化到多场景多目标联合优化的能力；

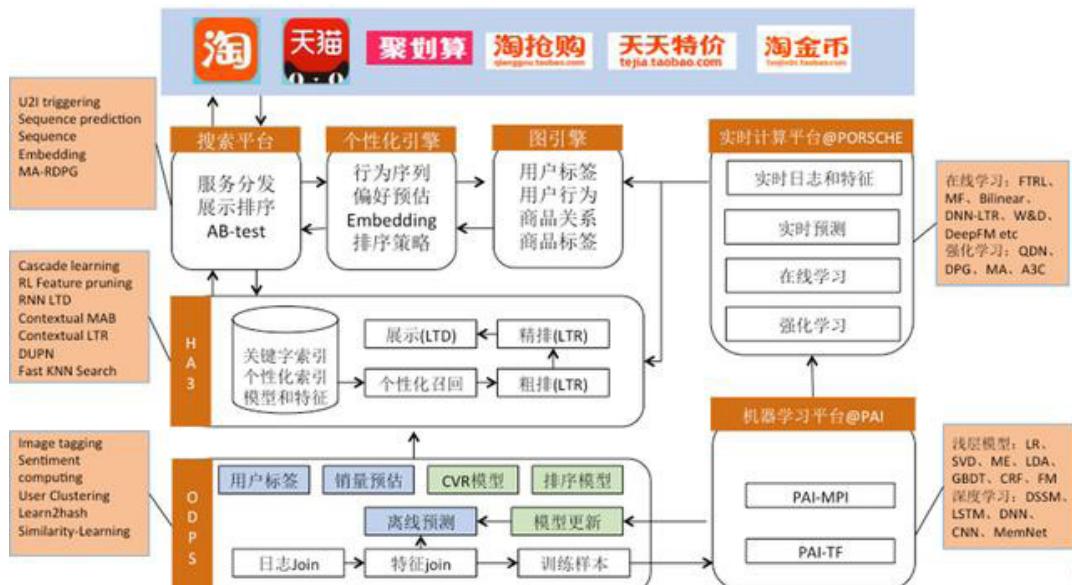
再次是在性能优化上做的工作，包括模型压缩、量化、低秩分解再到二值网络，大量的技术调研和论证，为未来提高深度模型预测性能和软硬件协同优化做了很好的技术铺垫；

最后是排序平台化，实现了PC商品搜索、无线商品搜索、店铺内搜索搜索和店铺搜索的搜索服务统一，通过特征和模型复用，实现了多条业务线技术的快速升级。下面我会简要的概括下在四个方向上取得的主要进展和背后的思考。

下面是搜索系统和算法的简图。系统包括：

- a. 离线数据平台ODPS，负责离线日志join、特征抽取和离线模型预估产出排序特征，时效性不强的特征都是通过离线数据平台产出的，比如用户性别标签，商品关键字等；
- b. 离线机器学习平台PAI，底层是主流的parameter server和TF深度学习框架，平台实现了大部分机器学习算法模型的并行训练和预测，在搜索应用中主要作用是离线模型训练产出离线排序特征模型；
- c. 流式计算和在线学习平台 Porsche，流式计算是基于blink负责实时日志解析和特征join生成实时排序特征，在线学习和离线学习底层框架可以相同，差别主要是依赖数据源和部分优化方法不同，由于用户行为和市场环境变化快，流式计算和在线学习在搜索应用非常广泛，并积累了不少在线学习和强化学习算法；
- d. 在线服务平台，包括引擎、排序服务和搜索平台组成，负责在线的服务分发、索引查询、排序服务和结果合并等功能，搜索的排序策略、相关性、个性化等模型主要通过在线预测服务生效。经过多年发展我们

已经具备了非常完善商品搜索排序算法体系，包括知识图谱、分词、tagging、类目预测、意图预测、拼写纠错、query 推荐、query 语义改写、相关性、商品标签、商品质量、店铺分层、用户profile、用户偏好、用户感知、召回策略、个性化模型、多样性策略、异构服务混排策略、多目标联合优化策略、多场景联合排序策略等，并平台化的方式赋能相关业务团队。



搜索系统和算法简图

系统进展包括机器学习平台和在线预测平台

机器学习平台。搜索训练样本主要来自用户行为，由于用户行为是流式数据，适合做在线深度学习，但当模型参数非常庞大需要海量的样本时在线学习需要很长的时间才能收敛，这时一般是先做离线预训练再结合增量或在线学习，另外有些模型离线预训练后在线只需要对接近输出层的网络做 fine-tuning。搜索在实际应用的有离线机器学习平台 PAI 和在线机器学习平台 Porsche，两个平台深度学习框架目前都统一到了 tf-pai，tf-pai 对原生 tf 做了一些优化，比如底层通讯，稀疏参数存储、优化方法、GPU 显存优化等，比原生 tf 训练深度有较大的提升，训练上千万样本和上百亿参数的深度模型毫无压力。虽然 Porsche 和 PAI 都支持 GPU，但在

搜索应用中 CPU 依然是主流，GPU 应用比较少，原因主要是个性化相对图像或语音简单，特征抽取网络比较浅，维度相对较低，GPU 的稠密矩阵计算能力得不到充分发挥，同时离在线混布后流量低谷期间腾出了大量的在线服务闲置 CPU，把临时闲置的 CPU 利用起来做深度学习训练是一个非常好的思路。

在线预估RTP，搜索排序算分服务。由于每次搜索请求有上千个商品需要计算排序分数，深度模型应用对RTP服务的压力是非常大的，RTP通过采用异构计算，计算算子化和模型分片等方式解决了深度模型inference计算和存储问题，深度模型用GPU，浅层模型用CPU，今年双11期间搜索RTP服务用到了550张GPU卡。另外，RTP还实现了离线/在线训练模型/数据和在线预测服务部署的无缝衔接，算法训练好的模型或数据可以很轻松的部署都在线服务，提升了算法迭代效率。

算法包括智能交互、语义搜索、智能匹配和搜索策略四个方向

智能交互。商品搜索就是带交互的商品推荐，用户通过关键字输入搜索意图，引擎返回和搜索意图匹配的个性化推荐结果，好的交互技术能够帮助到用户更好的使用搜索引擎，目前搜索的交互主要是主动关键字输入和关键字推荐，比如搜索框中的默认查询词和搜索结果中的文字链等，推荐引擎根据用户搜索历史、上下文、行为和状态推荐关键字。

和商品推荐的区别是，关键字推荐是搜索链路的中间环节，关键字推荐的收益除了关键字的点击行为外，还需要考虑对整个购物链路的影响，包括在推荐关键字的后续行为中是否有商品点击、加购和成交或跳转到另外一个关键字的后继行为，这是一个典型的强化学习问题，action 是推荐的关键字候选集合，状态是用户当前搜索关键词、上下文等，收益是搜索引导的成交。除了被动的关键字推荐，我们也在思考搜索中更加主动的交互方式，能够做到像导购员一样的双向互动，主动询问用户需求，挑选个性化的商品和给出个性化的推荐理由，目前阿里搜索团队已经在做智能导购和智能内容方向的技术原型及论证，智能导购在技术上主要是借鉴对话系统，通过引导用户和引擎对话与关键字推荐方式互为补充，包括自然语

言理解，对话策略，对话生成，知识推理、知识问答和商品搜索等模块，功能主要包括：

a. 根据用户搜索上下文生成引导用户主动交互的文本，比如搜索“奶粉”时，会生成“您宝宝多大？0~6个月，6个月到1岁……”引导文案，提示用户细化搜索意图，如果用户输入“3个月”后，会召回相应段位的奶粉，并在后续的搜索中会记住对话状态“3个月”宝宝和提示用户“以下是适合3个月宝宝的奶粉”。

b. 知识导购，包含提高售前知识问答或知识提示，比如“3个月宝宝吃什么奶粉”回答“1段”。目前对话技术正在提高中，尤其是在多轮对话状态跟踪、知识问答和自动评价几个方面，但随着深度学习、强化学习和生成对抗学习等技术在NLP、对话策略、阅读理解等领域的应用，越来越多的训练数据和应用场景，domain specific 的对话技术未来几年应该会突飞猛进。智能内容生成，包括生成或辅助人工生成商品和清单的“卖点”，短标题和文本摘要等，让淘宝商品表达更加个性化和多元化。

语义搜索。语义搜索主要是解决关键字和商品内容之间的语义鸿沟，比如搜索“2~3周岁宝宝外套”，如果按照关键字匹配召回结果会远小于实际语义匹配的商品。

- 语义搜索的范围主要包括：

a. query tagging和改写，比如新品，年龄，尺码，店铺名，属性，类目等搜索意图识别和归一化，query tagging模型是用的经典的序列标注模型 bi-lstm + CRF，而标签分类（归一化）作为模型另外一个任务，将序列标注和分类融合在一起学习。

b. query 改写，主要是计算query之间相似度，把一个query改写成多个语义相似的query，通常做法是先用不同改写策略生成改写候选query集合，比如词替换、向量化后top k、点击商品相似度等，然后在用ltr对后续集合排序找出合适的改写集合，模型设计和训练相对简单，比较难的是如何构建高质量的训练样本集合，线上我们用bandit 的方法探测部分query 改写结果的优劣，离线则用规则和生成对抗网络生成一批质量较高的样

本。

c. 商品内容理解和语义标签，通过商品图片，详情页，评价和同义词，上下位词等给商品打标签或扩充商品索引内容，比如用 image tagging 技术生成图片的文本标签丰富商品内容，或者更进一步用直接用图片向量和文本向量融合，实现富媒体的检索和查询。

d. 语义匹配，经典的DSSM 模型技术把query 和商品变成向量，用向量内积表达语义相似度，在问答或阅读理解中大量用到多层LSTM + attention 做语义匹配，同样高质量样本，特别是高质量负样本很大程度上决定了模型的质量，我们没有采样效率很低的随机负采样，而是基于电商知识图谱，通过生成字面相似但不相关的query及相关文档的方法生成负样本。

从上面可以看到query tagging、query相似度、语义匹配和语义相关性是多个目标不同但关联程度非常高的任务。下一步计划用统一的语义计算框架支持不同的语义计算任务，具体包括

1. 开发基于商品内容的商品表征学习框架，为商品内容理解，内容生成，商品召回和相关性提供统一的商品表征学习框架，重点包括商品标题，属性，详情页和评价等文本信息抽取，图像特征抽取和多模信号融合。

2. query 表征学习框架，为query 类目预测，query改写，query 推荐等提供统一的表征学习框架，重点通过多个query 相似任务训练统一的query 表征学习模型。

3. 语义召回，语义相关性等业务应用模型框架。语义搜索除了增加搜索结果相关性，提升用户体验外，也可以一定程度上遏制淘宝商品标题堆砌热门关键词的问题。

- 智能匹配。这里主要是指个性化和排序。内容包括：

a. ibrain (深度用户感知网络)，搜索或推荐中个性化的特点是用户的理解与表达，基于淘宝的用户画像静态特征和用户行为动态特征，我们基于multi-modals learning、multi-task representation learning以及LSTM的相

关技术，从海量用户行为日志中直接学习用户的通用表达，该学习方法善于“总结经验”、“触类旁通”，使得到的用户表达更基础且更全面，能够直接用于用户行为识别、偏好预估、个性化召回、个性化排序等任务，在搜索、推荐和广告等个性化业务中有广泛的应用场景，感知网络超过10B个参数，已经学习了千亿次的用户行为，并且会保持不间断的增量学习越来越聪明。

b. 多模学习，淘宝商品有文本、图像、标签、id、品牌、类目、店铺及统计特征，这些特征彼此有一定程度的冗余和互补，我们利用多模学习通过多模联合学习方法把多维度特征融合在一起形成统一的商品标准，并在多模联合学习中引入self-attention实现特征维度在不同场景下的差异，比如女装下图片特征比较重要，3C下文本比较重要等。

c. deepfm，相对wide & deep模型，deepfm增加了特征组合能力，基于先验知识的组合特征能够应用到深度学习模型中，提升模型预测精度。

d. 在线深度排序模型，由于行为类型和商品重要性差异，每个样本学习权重不同，通过样本池对大权重样本重复copy分批学习，有效的提升了模型学习稳定性，同时通过融合用户状态深度ltr模型实现了千人千面的排序模型学习。

e. 全局排序，ltr只对单个文档打分然后按照ltr分数和打散规则排序，容易导致搜索结果同质化，影响总页效率，全局排序通过已知排序结果做为上下文预测下一个位置的商品点击概率，有效提升了总页排序效率。

f. 另外工程还实现了基于用户和商品向量的向量召回引擎，相对倒排索引，向量化召回泛化能力更强，对语义搜索和提高个性化匹配深度是非常有价值的。以上实现了搜索从召回、排序特征、排序模型、个性化和重排的深度学习升级，在双11无线商品搜索中带来超过10% (AB-Test)的搜索指标提升。

多智能体协同学习实现智能决策

搜索中个性化产品都是成交最大化，导致的问题是搜索结果趋同，浪费曝光，今年做的一个重要工作是利用多智能体协同学习技术，实现了搜

索多个异构场景间的环境感知、场景通信、单独决策和联合学习，实现联合收益最大化，而不是此消彼长，在今年双11中联合优化版本带来的店铺内和无线搜索综合指标提升12% (AB-Test)，比非联合优化版本高3% (AB-Test)。

性能优化。在深度学习刚起步的时候，我们意识到深度模型inference性能会是一个瓶颈，所以在这方面做了大量的调研和实验，包括模型压缩(剪枝)，低秩分解，量化和二值网络。

通过以上技术，今年双11期间在手淘默认搜索、店铺内搜索、店铺搜索等均取得了10% (AB-Test)以上的搜索指标提升。

阿里巴巴人工智能搜索应用的未来计划

通用用户表征学习。前面介绍的 DUPN 是一个非常不错的用户表征学习模型，但基于 query 的 attention 只适合搜索，同时缺少基于日志来源的 attention，难以推广到其他业务，在思考做一个能够适合多个业务场景的用户表征模型，非搜索业务做些简单 fine tuning 就能取得比较好的效果；同时用户购物偏好受季节和周期等影响，时间跨度非常大，最近 K 个行为序列假设太简单，我们在思考能够做 life-long learning 的模型，能够学习用户过去几年的行为序列。

搜索链路联合优化。从用户进入搜索到离开搜索链路中的整体优化，比如 搜索前的 query 引导（底纹），搜索中的商品和内容排序，搜索后的 query 推荐（锦囊）等场景。

跨场景联合优化。今年搜索内部主搜索和店铺内搜索联合优化取得了很好的结果，未来希望能够拓展在更多大流量场景，提高手淘的整体购物体验；多目标联合优化。搜索除了成交外，还需要承担卖家多样性，流量公平性，流量商业化等居多平台和卖家的诉求，搜索产品中除了商品搜索外还有“穹顶”，“主题搜索”，“锦囊”，“内容搜索”等非商品搜索内容，不同搜索目标和不同内容（物种）之间的联合优化未来很值得深挖。



2017 年回顾：NLP、深度学习与大数据

作者 | 核子可乐、薛命灯

在今天的文章中，我们将回顾 2017 年年内基于深度学习技术所实现的 AI 发展成效。当然，受到篇幅所限，本篇文章不可能涵盖全部科学论文、框架及工具。在这里，我们只希望与大家分享这一年中最振奋人心的成果，同时结合全球 AI 大咖观点，带你回顾过去一年以来，深度学习带来的发展及其意义。

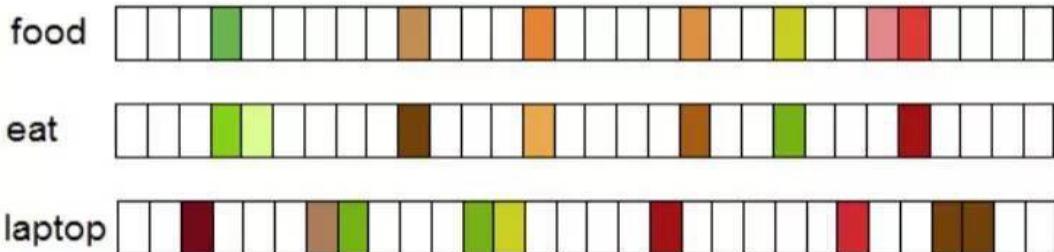
自然语言处理的发展与趋势

2017 年是自然语言处理领域的重要一年，深度学习所发挥的作用正在不断扩大，尤其在某些情况下能够带来惊人的效果——而所有迹象都表明，这一趋势在新的一年中还将持续下去。

从训练 word2vec 到使用预训练模型

可以说，词嵌入是深度学习在自然语言处理领域最为知名的技术之

一。词嵌入源自 Harris 于 1954 年提出的分布假说，他认为具有相似含义的词汇通常会出现在同类语境当中。关于词嵌入的详细解释，这里建议大家参阅 Gabriel Mordecki 发布的这篇精彩文章。



词汇分布向量示例

Word2vec（由 Mikolov 等于 2013 年提出）与 GloVe（由 Pennington 等于 2014 年提出）等算法正是这一领域的先驱性方案——虽然其尚不属于深度学习（word2vec 中的神经网络较为浅表，而 GloVe 则采取基于计数的实现方法），但利用二者训练的模型已经被广泛应用于各类深度学习自然语言处理方案当中。另外需要强调的是，这两种算法确实极具成效，甚至使得词嵌入成为目前最值得肯定的实现方法。

作为起步，对于需要使用词嵌入的特定 NLP 问题，我们倾向于首先使用一套与之相关的大型语料库进行模型训练。当然，这种作法存在一定的入门难度——也正因为如此，预训练模型才开始逐渐普及起来。在利用维基百科、Twitter、谷歌新闻以及 Web 抓取等数据完成训练之后，这些模型将允许大家轻松将词嵌入机制整合至自己的深度学习算法当中。

2017 年的种种实践证明，预训练词嵌入模型已经成为解决 NLP 问题的一类关键性工具。举例来说，来自 Facebook AI Research（简称 FAIR）实验室的 fastText 即提供包含 294 种语言的预训练向量，这无疑给整个技术社区带来了巨大的贡献与推动作用。除了可观的语言支持数量，fastText 还采用字符 N 元模型（即使是来自特定领域的术语等罕见词，其中亦包含同样存在于其它常见词中的 N 元字符组合），这意味着 fastText 能够回避 OOV（即词汇量超出）问题。从这个角度来看，fastText 的表现要

优于 word2vec 以及 GloVe，而且前者在处理小型数据集时同样更胜一筹。

尽管已经实现了一定进展，但这方面仍有大量工作需要完成。举例来说，卓越的 NLP 框架 spaCy 就能够对词嵌入与深度学习模型加以整合，从而以原生方式实现 NER 及依存关系语法分析等任务，使得用户能够更新现有模型或者使用自主训练的模型。

未来应该会出现更多针对特定领域的预训练模型（例如生物学、文学、经济学等），从而进一步降低自然语言处理的实现门槛。届时用户只需要对这些模型进行简单微调，即可顺利匹配自己的实际用例。与此同时，能够适应词嵌入机制的方法也将不断涌现。

调整通用嵌入以适配特定用例

预训练词嵌入方案的主要缺点，在于其使用的训练数据往往与我们的实际数据之间存在着词汇分布差异。假定您面对的是生物学论文、食谱或者经济学研究文献，大家可能没有规模可观的语料库用于嵌入训练；在这种情况下，通用词嵌入方案可能有助于带来相对理想的成果。然而，我们该如何对词嵌入方案进行调整，从而确保其适合您的特定用例？

这种适应性通常被称为 NLP 中的跨领域或领域适应技术，其与迁移学习非常相似。Yang 等人在这方面拿出了非常有趣的成果。今年，他们公布了一套正则化连续跳元模型，可根据给定的源领域词嵌入学习目标领域的嵌入特征。

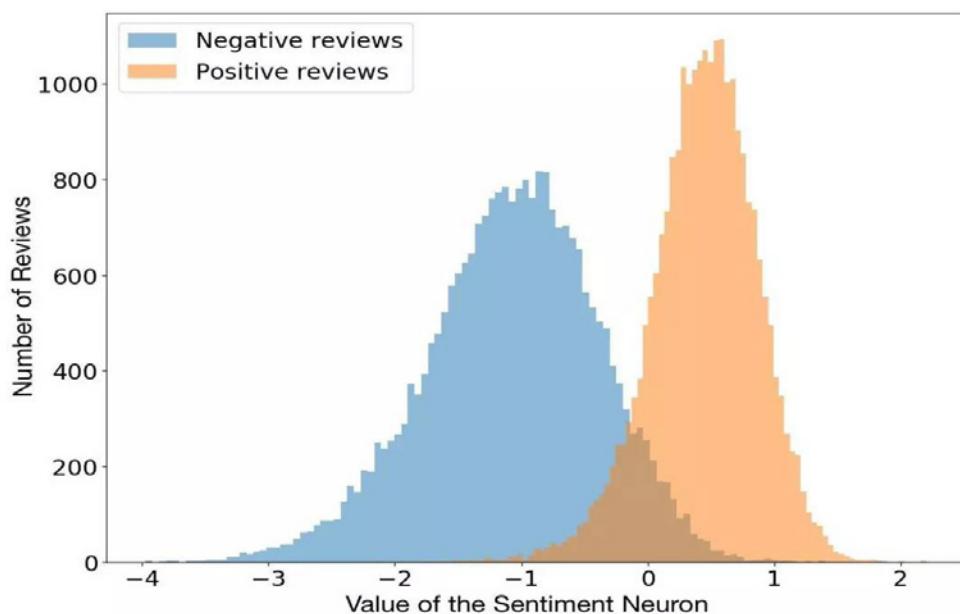
其中的核心思路简单但极富成效。想象一下，假定我们已经在源领域当中知晓词 w 的词嵌入为 w_{sw} 。为了计算 w_{twt} （目标领域）的嵌入，作者在两个领域之间向 w_{sw} 添加了一个特定迁移量。基本上，如果该词在两个领域皆频繁出现，则意味着其语义与领域本身不存在依存关系。在这种情况下，高迁移量意味着该词在两个领域中产生的嵌入结果倾向于彼此相似。但如果该词在特定领域中的出现频率比另一领域明显更高，则迁移量将相应降低。

作为与词嵌入相关的研究议题，这项技术还没有得到广泛关注与探

索——但我相信其会在不久的未来获得应有的重视。

情感分析——令人印象深刻的“副产物”

与青霉素乃至 X 光一样，情感分析同样是一场意外中的惊喜。今年，Radford 等人开始探索字节级递归语言模型的特性，但其本意只是希望预测 Amazon 评论内容中的下一个字符。最终的结论显示，他们训练模型中的某个神经元能够准确预测情感值。是的，这个单一“情感神经元”能够以令人印象深刻的水准将评论内容归类为“正面”或“负面”。



审查极性与神经元的值

在注意到这种现象后，作者们决定利用斯坦福情绪树库对该模型进行进一步测试，并发现其准确性高达 91.8%——优于原有最好成绩 90.2%。这意味着他们的模型能够以无监督方式利用更少实例实现训练，并至少能够立足斯坦福情绪树库这一特定但涵盖范围广泛的数据集之上实现最为先进的情感分析能力。

情感神经元的实际使用

由于该模型立足字符层级运作，因此各神经元会根据文本中的每一字

符作出变更，而最终成效令人印象深刻。

举例来说，在“best”一词之后，该神经元的值会变为强正值。然而这种效果将随着“horrendous”这一负面词语的出现而消失——非常符合逻辑。

```
This is one of Crichton's best books. The characters of Karen Ross, Peter Elliot, Munro, and Amy are beautifully developed and their interactions are exciting, complex, and fast-paced throughout this impressive novel. And about 99.8 percent of that got lost in the film. Seriously, the screenplay AND the directing were horrendous and clearly done by people who could not fathom what was good about the novel. I can't fault the actors because frankly, they never had a chance to make this turkey live up to Crichton's original work. I know good novels, especially those with a science fiction edge, are hard to bring to the screen in a way that lives up to the original. But this may be the absolute worst disparity in quality between novel and screen adaptation ever. The book is really, really good. The movie is just dreadful.
```

情感神经元的行为

生成包含偏向极性的文本

当然，这套训练模型亦是一套行之有效的生成模型，因此能够用于生成类似 Amazon 评论的文本内容。而让我个人感到惊喜的是，大家甚至能够简单覆盖情感神经元的值来选定所生成文本的偏向极性。

情感固定为正面	情感固定为负面
Best hammock ever! Stays in place and holds its shape. Comfy (I love the deep neon pictures on it), and looks so cute.	They didn't fit either. Straight high sticks at the end. On par with other buds I have. Lesson learned to avoid.
Just what I was looking for. Nice fitted pants, exactly matched seam to color contrast with other pants I own. Highly recommended and also very happy!	The package received was blank and has no barcode. A waste of time and money.

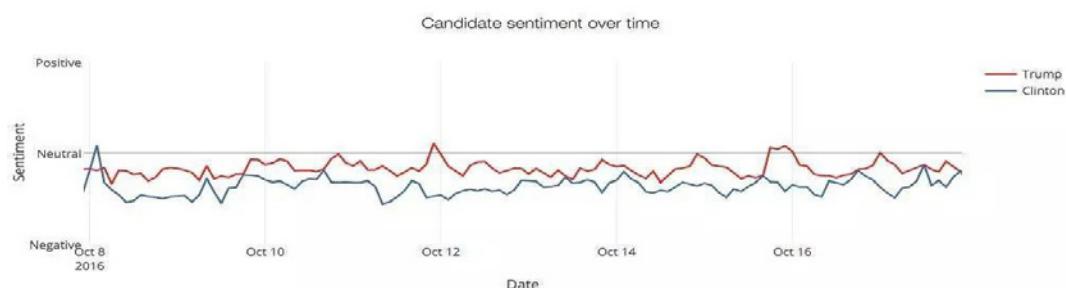
以上为所生成的示例文本

作者们选择了乘法 LSTM（由 Krause 等人于 2016 年发布）作为其神经网络模型，这主要是由于他们发现乘法 LSTM 的超参数设置收敛速度要远高于其它普通 LSTM。其中包含 4096 个单元，且利用 8200 万条 Amazon 评论内容进行训练。

时至今日，我们仍无法理解这套经过训练的模型为何能够以如此精确的方式捕捉到评论内容的情感倾向。当然，大家也可以尝试训练自己的模型并进行实验。再有，如果您拥有充分的时间与 GPU 计算资源，亦可投入一个月利用四块英伟达 Pascal GPU 重现研究人员们的训练过程。

Twitter 中的情感分析

无论是对企业品牌的评价、对营销活动影响作出分析抑或是量化 2016 年美国总统大选中民众对希拉里与特朗普的支持程度，Twitter 中的情感分析一直作为一款强大的工具存在。



特朗普对希拉里：Twitter 上的情感分析

SemEval 2017

Twitter 上的情感分析已经引起了 NLP 研究人员们的广泛关注，同时亦成为政治及社会科学界内的热门议题。也正因为如此，SemEval 自 2013 年以来提出了一项更为具体的任务。

今年，总计 48 支队伍参与到评选当中，这也再次证明了 SemEval 的魅力所在。为了进一步了解 Twitter 公司组织的 SemEval 究竟是什么，我们将首先回顾其今年提出的五项任务：

任务 A：根据给定的一条推文，判断其代表正面、负面抑或中性情感。

任务 B: 根据给定的一条推文与主题, 将与该主题相关的推文内容进行观点二分: 正面与负面。

任务 C: 根据给定的一条推文与主题, 将与该主题相关的推文进行观点五分: 强正面、弱正面、中立、弱负面、强负面。

任务 D: 根据与某一主题相关的一组推文, 估算其中正面与负面情感类别的分布情况。

任务 E: 根据与某一主题相关的一组推文, 立足以下五种类别进行推文内容估算: 强正面、弱正面、中立、弱负面、强负面。

如大家所见, 任务 A 属于最常见的任务, 有 38 个团队参与了这项任务; 但其它任务则更具挑战性。主办方指出, 深度学习方法的使用量已经相当可观并仍在不断增加——今年已经有 20 个团队开始采用卷积神经网络 (简称 CNN) 与长 / 短期记忆 (简称 LSTM) 等模型。此外, 尽管 SVM 模型仍然相当流行, 但已经有一部分参与者将其与神经网络方法或词嵌入特征加以结合。

BB_twtr 系统

今年我还发现了一套纯粹的深度学习系统, 即 BB_twtr 系统 (Cliche, 2017 年), 其在五项任务的英文版本挑战中全部位列第一。该作者将 10 套 CNN 与 10 套 biLSTM 结合起来, 并利用不同超参数以及不同预训练策略对其进行训练。感兴趣的朋友可以查阅链接内论文中对该网络架构的详尽描述。

为了训练这些模型, 作者采用了人类标记推文 (为了让大家体会到其工作量, 单是任务 A 就包含 49693 条此类推文), 同时构建起一套包含 1 亿条推文的未标记数据集。其能够通过简单的字符表情标记——例如: -) ——从这套未标记数据集中提取出独立数据集。这些推文通过小写、标记、URL 以及表情符号等被替换为统一的标记方式, 用于强调证据的重复字符也经过类似的处理 (例如将‘Niiice’与‘Niiiiiiice’统一转换为‘Niice’)。

为了对作为 CNN 及 biLSTM 输入内容的词嵌入进行预训练，该作者采用了 word2vec、GloVe 以及 fastText 对未标记数据集进行训练，且三者皆采用默认设置。在此之后，他利用中立数据集对词嵌入进行微调，旨在添加极性信息；最后再利用人类标记数据集对模型进行再次微调。

利用以往 SemEval 数据集进行实验，他发现 GloVe 会导致成效降低，且并不存在适用于全部数据集的最佳模型。该作者随后将全部模型利用一套软投票策略结合起来。最终得出的模型顺利战胜了 2014 年与 2016 年的获胜模型方案，且与其它几年的优胜者亦相差不多。正是这套方案，在 2017 年的 SemEval 当中获得五项任务的英文版本优胜。

尽管他选择的组合方式并不具备有机性——而仅通过一种简单的软投票策略实现，但这项工作仍然证明了将多种深度学习模型加以结合的可能性。事实上，这次尝试还证明了我们完全能够以端到端方式（即输入内容必须经过预处理）实现超越监督学习方法的 Twitter 情感分析能力。

令人兴奋的抽象概括系统

自动概括与自动翻译一样，皆属于自然语言处理领域的元老级任务之一。目前实现自动概括主要通过两种方法：基于提取型方法，通过从源文本中提取最重要的文本段建立摘要；基于抽象型方法，以抽象方式通过生成文本构建摘要内容。从历史角度来看，基于提取的方法最为常见，这主要是因为其实现难度要远低于基于抽象型方法。

过去几年以来，基于递归神经网络（简称 RNN）的模型开始在文本生成方面取得惊人的进展。其在简短输入与输出文本场景中的表现非常出色，但所生成的长文本却存在着连续性差及重复度高等问题。在工作中，Paulus 等人提出了一种新的神经网络模型以克服上述局限——而结果令人振奋，具体如下图所示。

作者们利用一款 biLSTM 编码器读取输入内容，并利用 LSTM 解码器生成输出结果。他们的主要贡献在于利用一种新的内部关注策略对输入内容以及连续生成的输出结果进行分别关注，同时结合标准监督词语预测与

强化学习机制建立起一种新的训练方法。

The bottleneck is no longer access to information; now it's our ability to keep up.
AI can be trained on a variety of different types of texts and summary lengths.
A model that can generate long, coherent, and meaningful summaries remains an open research problem.

The last few decades have witnessed a fundamental change in the challenge of taking in new information. The bottleneck is no longer access to information; now it's our ability to keep up. We all have to read more and more to keep up-to-date with our jobs, the news, and social media. We've looked at how AI can improve people's work by helping with this information deluge and one potential answer is to have algorithms automatically summarize longer texts. Training a model that can generate long, coherent, and meaningful summaries remains an open research problem. In fact, generating any kind of longer text is hard for even the most advanced deep learning algorithms. In order to make summarization successful, we introduce two separate improvements: a more contextual word generation model and a new way of training summarization models via reinforcement learning (RL). The combination of the two training methods enables the system to create relevant and highly readable multi-sentence summaries of long text, such as news articles, significantly improving on previous results. Our algorithm can be trained on a variety of different types of texts and summary lengths. In this blog post, we present the main contributions of our model and an overview of the natural language challenges specific to text summarization.

模型所生成的摘要内容

内部关注策略

之所以要提出内部关注策略这一概念，主要是为了避免输出结果中的重复性问题。为了达到这项目标，他们在解码过程中使用暂时关注机制查看输入文本中的前一段落，并借此决定下一个将要生成的词汇。这就迫使该模型在生成过程中使用输入内容中的不同部分。此外，作者们还允许模型从解码器当中访问此前曾经存在的隐藏状态。将这两条函数结合起来，即可为摘要输出结果选择最理想的一个单词。

强化学习

在创建同一条摘要时，不同的人往往会展开完全不同的词汇与句子——而这两条摘要可能同样准确有效。因此，良好的摘要并不一定需要尽可能同训练数据集中出现的词汇序列相匹配。以此为前提，作者们决定避免使用标准的指导强迫算法，而是在每个解码步骤内（即生成每个单词时）尽可能减小丢失值。事实证明，他们选择的这一强化学习策略确实非常有效。

来自近端到端模型的出色成果

这套模型接受了 CNN/Daily Mail 数据集的测试，并得到了极为出色的处理结果。除此之外，人类评估者亦对该模型作出了测试，并发现其摘要结果的可读性与质量都有所提升。这些结果令人印象深刻，特别是考虑

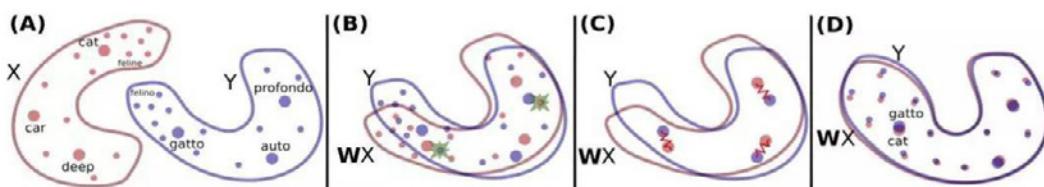
到其仅需要非常基础的预处理过程：对输入文本进行标记与小写化，而后将数字全部替换为“0”，最终将数据集内的部分特定实体彻底移除。

这是否代表着通往无监督机器翻译的第一步？

所谓双语词典归纳，是指利用两种语言的源语与单语语料库实现单词识别与翻译——这是一项历史相当悠久的自然语言处理任务。自动生成的双语词典能够有力支持其它 NLP 类任务，包括信息检索与统计类机器翻译等。然而，此类方法大多高度依赖于某种资源——例如初始版本的双语词典。而这类词典往往并不存在或者很难构建。

随着词嵌入机制的成功，人们开始考虑实现跨语言词嵌入的可能性——其目标在于分配嵌入空间，而非建立词典。遗憾的是，第一批实现方案仍然依赖于双语词典或对等语料库。不过在实践工作当中，Conneau 等人（2018 年）提出了一种极具发展前景的方法，其不依赖于任何特定资源，且在多种语言到语言翻译、句子翻译检索以及跨语言单词相似性类任务当中拥有优于现有监督学习方法的实际成效。

作者们开发出的方法是将所输入的两组词嵌入以单一语言数据为基础进行独立训练，而后学习二者之间的映射关系，从而使得翻译结果在公共空间内尽可能接近。作者们利用 fastText 对维基百科文档进行无监督词汇向量训练，下图所示为这种方法的核心实现思路：



在两套词嵌入空间之间建立映射关系

其中红色的 X 分布为英语单词嵌入，而蓝色的 Y 分布则为意大利语单词嵌入。

作者们首先利用对抗性学习以获取用于执行第一次初始对齐的旋转矩

阵 W。根据 Goodfellow 等 (2014 年) 提出的基本原则，他们构建起一套生成对抗网络（简称 GAN）。若大家希望了解 GAN 的工作原理，推荐各位参阅本篇由 Pablo Soto 撰写的文章。

为了在对抗学习过程中进行问题建模，他们在定义中为鉴别器添加了判定角色，同时随机从 WX 与 Y 中提供某些样本元素（详见上图中的第二列），借以判断这些元素属于哪一种语言。接下来，他们训练 W 以防止鉴别器作出准确的预测。这种作法在我看来简直有才，而其结果也相当令人满意。

在此之后，他们利用两个后续步骤进一步完善映射关系。其一是避免在映射计算中因罕见字的出现而引发问题。其二是构建实际翻译能力，其中主要应用到已经学会的映射关系与距离度量机制。

在某些情况下，这套模型拥有极为先进的处理结果。例如在英语到意大利语的单词翻译过程中，在 P@10 的情况下，其能够以接近 17% 的精度完成源单词翻译（具体数量超过 1500 个）。

	English to Italian			Italian to English		
	P@1	P@5	P@10	P@1	P@5	P@10
<i>Methods with cross-lingual supervision</i>						
Mikolov et al. (2013b) [†]	33.8	48.3	53.9	24.9	41.0	47.4
Dinu et al. (2015) [†]	38.5	56.4	63.9	24.6	45.4	54.1
CCA [†]	36.1	52.7	58.1	31.0	49.9	57.0
Artetxe et al. (2017)	39.7	54.7	60.5	33.8	52.4	59.1
Smith et al. (2017) [†]	43.1	60.7	66.4	38.0	58.5	63.6
Procrustes - CSLS	44.9	61.8	66.6	38.5	57.2	63.0
<i>Methods with cross-lingual supervision (Wiki)</i>						
Procrustes - CSLS	63.7	78.6	81.1	56.3	76.2	80.6
<i>Methods without cross-lingual supervision (Wiki)</i>						
Adv - Refine - CSLS	66.2	80.4	83.4	58.7	76.5	80.9

英语到意大利语单词翻译平均精度

作者们宣称，他们的方法将能够作为无监督机器翻译技术的重要起点。如果实际情况真是如此，那么未来的前景绝对值得期待。当然，我们也希望看到这种新方法能够走得更快、更远。

专用型框架与工具

目前市面上存在大量通用型深度学习框架与工具，其中 TensorFlow、Keras 以及 PyTorch 选项得到了广泛使用。然而，专用型开源 NLP 深度学习框架及工具也开始兴起。2017 年是令人振奋的一年，目前已经有不少非常实用的开源框架被交付至社区手中。而以下三款引起了我的浓厚兴趣。

AllenNLP

AllenNLP 框架是一套构建于 PyTorch 之上的平台，用于在语义 NLP 任务中轻松利用深度学习方法解决问题。其目标是帮助研究人员设计并评估新模型。该框架包含多种常用语义 NLP 任务的参考实验模型，其中包括语义角色标记、文本引用以及共因解析等。

ParlAI

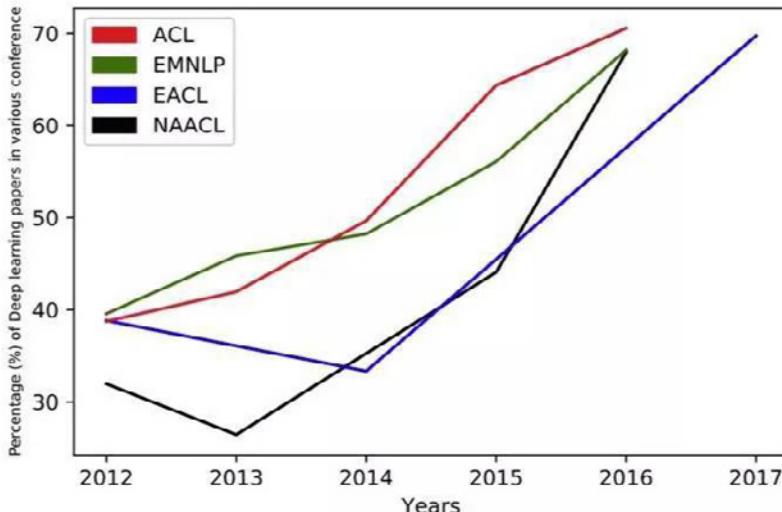
ParlAI 框架是一套开源软件平台，用于进行对话研究。其利用 Python 实现，旨在为对话模型的共享、训练与测试提供一套统一的框架。ParlAI 能够与 Amazon Mechanical Turk 实现轻松集成。另外，其还提供多种流行数据集，并能够支持大量神经模型——包括记忆网络、seq2seq 以及关注型 LSTM 等。

OpenNMT

OpenNMT 工具集是一款通用型框架，专门用于序列到序列类模型。其可用于执行诸如机器翻译、摘要、图像到文本以及语音识别等任务。

写在最后

毫无疑问，用于解决 NLP 类问题的深度学习技术正在不断增加。在这方面的一大证明性指标，在于过去几年来发表在 ACL、EMNLP、EACL 以及 NAACL 等关键性 NLP 会议上的深度学习论文在比例上出现了显著提升。



深度学习论文百分比变化图

然而，真正的端到端学习目前才刚刚开始。我们仍需要完成一些经典的 NLP 任务以筹备数据集，例如对某些实体（包括 URL、数字以及电子邮箱地址等）进行清洗、标记或者统一化调整。我们还在利用各类通用型嵌入，其缺点在于无法捕捉到特定领域术语的重要意义，且在多词表达式的理解方面表现不佳——我在自己的工作项目中已经充分体会到了这些弊端。

对于将深度学习技术应用于 NLP 领域而言，2017 年无疑是伟大的一年。我希望 2018 年能够带来更多端到端学习成果，而各类专用型开源框架也能得到进一步发展。如果您对于本文中提及的各类成果及框架有着自己的看法，或者拥有您支持的方案，请在评论中与大家分享。

机器学习与大数据的发展与趋势

2017 年，我们见证了大数据将 AI 推向了技术浪潮之巅。AI 成为媒体和从业者的注意力焦点，当然这其中包含了正面（各行各业日趋强大的机器学习算法和 AI 应用）和负面（机器将取代人类工作，甚至控制人类世界）的信息。我们也目睹了基于数据的价值创新，包括数据科学平台、深度学习和主要几个厂商提供的机器学习云服务，还有机器智能、规范性分析、行为分析和物联网。

我们综合整理了一些数据科学家、AI 专家对 2017 年机器学习和数据分析发展现状的总结，以及他们对 2018 年发展趋势的预测，由于篇幅有限我们隐去了这些专家的名字，如果需要了解专家的详细信息，请参看文末的参考文章，如果各位读者有其他补充和观点，欢迎在评论区与我们讨论。

2017 的发展状况

AlphaGo Zero 带来了一种新的增强学习方式，或许是 2017 年 AI 领域最重要的研究成果

2017 年，我们看到了 AI 的大踏步发展。尽管之前的深度学习模型需要大量的数据来训练算法，但神经网络和增强学习的应用告诉我们，大数据集并非高效算法的必要条件。DeepMind 使用这些技术创造了 AlphaGo Zero，它的表现已经超出了之前的算法。

企业 AI 成为主流

很多大型公司启动了 AI 或机器学习项目，不过这些项目的目标有一定的局限性。大型厂商的项目日趋走向开源，DIY 项目会越来越多。这意味着企业必须提升数据科学技能。例如：

1. 谷歌发布了第二代 TPU，如果从能量方面来考量，它可以节省数十亿美元。
2. 英伟达发布的 Volta 架构基于特斯拉 GPU，每个 GPU 可以支持 120 万亿次浮点运算。
3. D-Wave 量子计算机炒作风波平息，带有 QISKit 量子编程框架的 20 量子位量子计算机出现。

机器学习被应用在数据集成上

2017 年是智能分析平台的发展元年。从分析机器人到自动化机器学习，数据科学中出现了太多复杂、智能自动化的东西。数据集成和数据预备平台能够智能地处理数据源，自动修复数据管道中的错误，甚至基于通过与人类交互学习而来的知识进行自我维护或完成数据质量处理任务。自动机器学习平台和半自动化的特征工程很快改变了数字分析领域的游戏规则。

数据科学自动化，出现了很多自动化机器学习平台。机器学习解决了数据分析和数据管理的大难题，需要大量人工介入的数据集成被某种程度的自动化方式所取代，为我们节省了大量时间。

保守的公司开始拥抱开源

最为保守的传统公司（如银行、保险、健康医疗）开始主动使用开源的数据分析、AI 和数据管理软件。有些公司鼓励员工抛弃使用具有著作权的工具，有些则只建议在个别项目上使用它们。这其中也有成本方面的考虑，但更好的性能和招聘方面的便利也是重要的考虑因素。

Python、Java 和 R 语言从 2017 年开始成为最为吃香的编程语言

人们对 AI 发展的期待快过其实际发展程度

2018 年趋势预测

AI 将更多应用在商业领域

2018 年，AI 的发展脚步会加快，AI 的价值将在这一年得到体现：

- McAfee 实验室的研究报告表明，对抗机器学习将被用在网络入侵检测、欺诈检测、垃圾检测和木马检测上。
- HPE 将研发标量积引擎，并推出自己的神经网络芯片，用于高性能推理计算，如深度神经网络、卷积神经网络和循环神经网络。
- 无监督学习和自治学习将助力机器人与周围的陆上环境和水下环境互动。
- 机器学习在物联网和边缘计算领域的应用门槛将会降低，空间位置智能将出现突破性的算法，应用在手机、RFID 传感器、UAV、无人机和卫星上。
- 机器学习应用继续扩张领地，比如市场、金融风险、欺诈检测、劳动力优化配置、制造业和健康医疗。
- 深度学习不管在势头上还是在实际应用价值上都蓬勃发展。一系列新型的高级神经网络将机器学习提升到新的高度，以高性能解决大信号输入问题，如图像分类（自动驾驶、医疗图像）、声音

(语音识别、说话者识别)、文本(文本分类)，甚至是“标准”的业务问题。

这一领域的开发内容与 2017 年相比可能不会有太大变化：流程自动化、机器智能、客户服务、个人定制化以及劳动力转型。物联网领域的发展也会更加成熟，包括更加成熟的安全特性、模块化平台、用于访问传感器数据流的 API 以及边缘分析接口。我们也将看到数字化在其他领域成为主流，如制造行业、基础设施领域、工程领域和建筑行业。我们相信，2018 年会有更多的从业者将 AI 的优势带向更广大的领域。

2017 年是星光耀眼的一年，很多甚至跟 AI 都擦不上边的厂商开始提供 AI 产品。2018 年，我们将看到 AI 和机器学习应用在更多的商业领域。为什么这么说？因为那些亟待解决业务问题的大佬们并不关心具体的技术将怎样发展，他们会想方设法加速供应链流动，想知道客户的动向，并向计算机寻求答案。那些能够以最快速度提供预测分析的厂商将成为游戏规则的制定者。

独立 AI 初创公司将走向衰落

在过去几年，风险资本的追捧催生了数百家 AI 初创公司，每家公司都只解决一小部分问题。尽管它们很努力，但要在现有的流程中实现集成将是一个巨大的挑战。因此，现有的公司要么提供易于集成的 AI“微服务”，要么向已经将 AI 嵌入到事务系统中的厂商购买服务。

规则与安全将至关重要

随着 AI 在众多领域的应用，如犯罪审判、金融、教育和职场，我们需要建立算法标准来评估它们的准确性。关于 AI 对社会影响的研究将会持续增长，包括建立 AI 的适用规则（比如避免决策黑盒）以及了解深度学习算法是如何做出决策的。

安全问题将继续升温，企业将在安全方面投入更多的精力，提升区块链可见性是提升公司数据安全性行之有效的方式。期待下一年能够看到自动化 AI 被无缝地集成到更多的分析和决策过程中。欧洲通用数据保护条例的实施确保数据不会被滥用，从而更好地保护个人数据。

量子计算将吸引更多目光

量子机器学习的未来取决于拥有更多状态的量子位，可能是 10 以上，而不是只能支持两种状态的量子位。量子计算和数据科学算法将吸引更多人的眼球，尽管真正的量子计算机还离我们很遥远。

AI 泡沫将持续膨胀

人们从 2017 年开始大肆谈论机器学习、AI 和预测分析，可惜大部分公司或厂商都是在故弄玄虚，他们根本没有真正的实力去做这些事情。这些领域需要时间和人才，实打实的经验是非常重要的！AI 泡沫将继续膨胀，不过我们也会看到沉淀的迹象。AI 仍然会被过度吹捧。

数据科学家群体将扩大

数据分析员和数据科学家需要知道哪些算法可以用来做什么。分析和机器学习的自动化将产生多元化的算法，有可能会出现“人人都是数据科学家”的局面。与此同时，GDPR（欧洲通用数据保护条例）将在 2018 年 5 月 25 号开始实行，这将给数据科学带来重要影响。

2018 年将是数据科学和预测分析领域出现众多领头羊的一年，不只是因为这是大势所趋，根本原因是它们将给我们的业务带来真正的改变。预测招聘可以为你省下数百万美元的招聘经费，AI 和机器学习可以在几秒钟内完全之前需要几天才能完成的事情。

2018 年，实现“人人都是数据科学家”的目标将是头等大事。从专家的经验来看，团队仍然需要保持综合性结构：为不具备数据分析背景的员工和高层提供工具来帮助他们做出决策。更重要的是，团队需要开发出自己的数据模型，要有能够理解模型和特定分析技术局限性的的数据科学家。

参考文章

- <https://tryolabs.com/blog/2017/12/12/deep-learning-for-nlp-advancements-and-trends-in-2017/>
- <https://www.kdnuggets.com/2017/12/data-science-machine-learning-main-developments-trends.html>



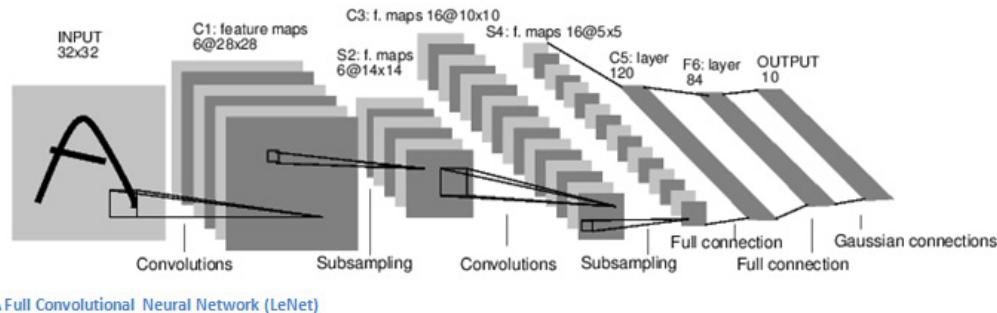
ImageNet 冠军带你入门计算机视觉： 卷积神经网络

作者 | 360 董健

神经网络的发展历史

卷积神经网络（Convolutional Neural Network, CNN）的起源可以追溯到上世纪 60 年代。生物学研究表明，视觉信息从视网膜传递到大脑中是通过多个层次的感受野（Receptive Field）激发完成的，并提出了 Neocognitron 等早期模型。1998 年，深度学习三巨头之一的 Lecun 等，正式提出了 CNN，并设计了如下图所示的 LeNet-5 模型。该模型在手写字符识别等领域取得了不错的成绩。

由于计算资源等原因，CNN 在很长时间内处于被遗忘的状态。二十多年后的 ImageNet 比赛中，基于 CNN 的 AlexNet 在比赛中大放异彩，并引领了 CNN 的复兴，此后 CNN 的研究进入了高速发展期。目前卷积神经网络的发展有两个主要方向：



- 如何提高模型的性能。这个方向的一大重点是如何训练更宽、更深的网络。沿着这一思路涌现出了包括 GoogleNet, VGG, ResNet, ResNext 在内的很多经典模型。
- 如何提高模型的速度。提高速度对 CNN 在移动端的部署至关重要。通过去掉 max pooling, 改用 stride 卷积, 使用 group 卷积, 定点化等方法, 人脸检测、前后背景分割等 CNN 应用已经在手机上大规模部署。

目前, CNN 是计算机视觉领域最重要的算法, 在很多问题上都取得了良好的效果。因为篇幅关系, 本文将主要介绍卷积神经网络的基础知识。

神经网络 vs 卷积神经网络

上篇文章中我们介绍了神经网络。神经网络在大数据处理, 语言识别等领域都有着广泛的应用。但在处理图像问题时会许多问题:

参数爆炸

以 $200 \times 200 \times 3$ 的图像为例, 如果输入层之后的 hidden layer 有 100 个神经元, 那么参数量会达到 $200 \times 200 \times 3 \times 100 = 1200$ 万。显然有如此多参数的模型是难以训练且容易过拟合的。

平移不变性

对很多图像问题, 我们希望模型满足一定的平移不变性。例如对图像分类问题, 我们希望物体出现在图片的任何位置上, 模型都能正确识别

出物体。

局部相关性

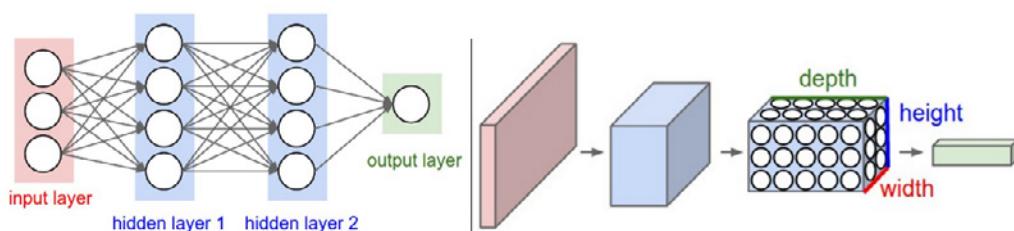
在大数据等问题中，输入维度之间不存在显式的拓扑关系，因此适合使用神经网络（全连接层）进行建模。但对于计算机视觉的问题，输入图片的相邻像素之间存在天然的拓扑关系。例如，判断图片中某个位置是否有物体时，我们只需要考虑这个位置周边的像素就可以了，而不需要像传统神经网络那样将图片中所有像素的信息作为输入。

为了克服神经网络的上述问题，在视觉领域，我们需要一种更合理的网络结构。卷积神经网络，在设计时通过局部连接和参数共享的方式，克服了神经网络的上述问题，因而在图像领域取得了惊人的效果。接下来我们将详细介绍卷积神经网络的原理。

卷积神经网络

网络结构

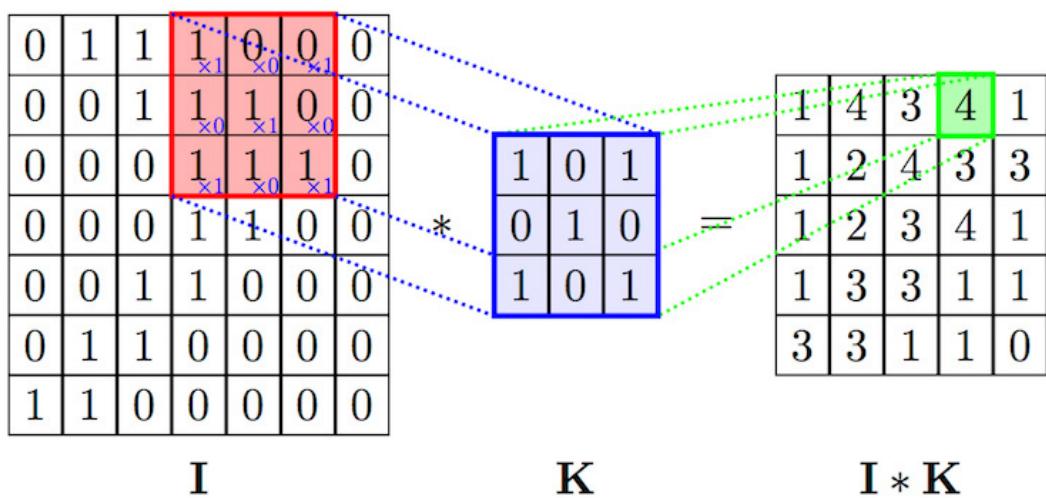
卷积神经网络和传统神经网络的整体结构大致相同。如下图所示，含有 2 层全连接层的传统神经网络和含有 2 层卷积层的卷积神经网络都是由基本单元堆叠而成，前一层的输出作为后一层的输入。最终层的输出，作为模型的预测值。二者的主要差别在于基本单元不同，卷积神经网络使用卷积层代替了神经网络中的全连接层。



和全连接层一样，卷积层中也含有可以学习的参数 weight 和 bias。模型的参数，可以按上一篇文章介绍的方法，在监督学习的框架下定义损失函数，通过反向传播进行优化。

卷积 (Convolution)

卷积层是整个卷积神经网络的基础。2D 卷积操作，可以看作是一个类似模板匹配的过程。如下图所示，将尺寸为 $h \times w \times d$ 的模板，通过滑动窗口的方式和输入进行匹配。滑动过程中，输入中对应位置的值和模板的权重的内积加一个偏移量 b ，作为对应输出位置的值。 w, h 是模板的大小，统称为 kernel size，在 CNN 中， w 和 h 一般会取相同的值。 d 是模板的 channel 数量，和输入的 channel 数相同，例如对 RGB 图像，



channel 数为 3。

模板在卷积神经网络中常被称为卷积核（K）或者过滤器（filter）。在标准卷积中，输出位置 (x,y) 对应的输出值可以表示成：

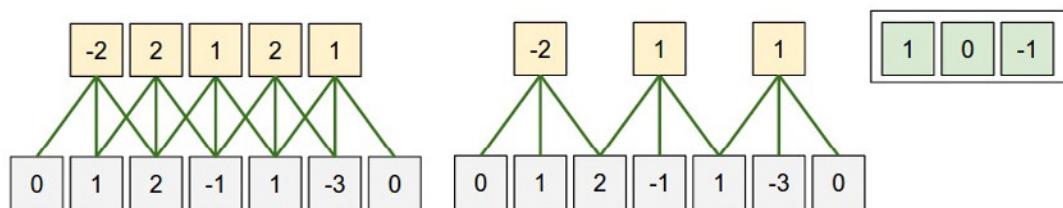
$$conv(I, K)_{xy} = \sum_{i=1}^h \sum_{j=1}^w \sum_{k=1}^d K_{ijk} \cdot I_{x+i-1, y+j-1, k} + b$$

在 CNN 中，除了描述单个 filter 的 h, w, d 这 3 个参数之外，还有 3 个重要的参数 depth, stride 和 padding：

- depth 指的是输出 channel 的数量，对应于卷积层中 filter 的数量

- stride 指的是 filter 每次滑动的步长
- padding 指的是在输入四周补 0 的宽度。使用 padding 主要是为了控制输出的尺寸。如果不添加 padding，使用 kernel size 大于 1 的 filter 会使输出尺寸比输入小。在实际中经常会增加 padding，使得输入和输出的尺寸一致。

如下图所示，对 1D 的情况，假设输入尺寸为 W ，filter 的尺寸为 F ，stride 为 S ，padding 为 P ，那么输出的尺寸为 $(W - F + 2P)/S + 1$ 为。通过设定 $P=(F-1)/2$ ，当 $S=1$ 时，输入和输出的尺寸会保持一致。2D 卷积的计算和 1D 卷积类似。



对比传统神经网络中的全连接层，卷积层实际上可以看成是全连接层的一种特例。首先是局部连接性，通过利用输入自带的空间拓扑结构，卷积神经网络只需考虑在空间上和输出节点距离在 filter 范围内的输入节点，其他边的权重都为 0。此外，对于不同的输出节点，我们强制 filter 的参数完全一致。但通过这种 局部连接 和 参数共享，卷积层可以更好的利用图像中内在的拓扑关系及平移不变形，大大减少了参数，从而得到一个更好的局部最优解，使其在图像问题上有更好的性能。

在 tensorflow 中实现卷积层非常简单，可以直接调用 `tf.nn.conv2d`：

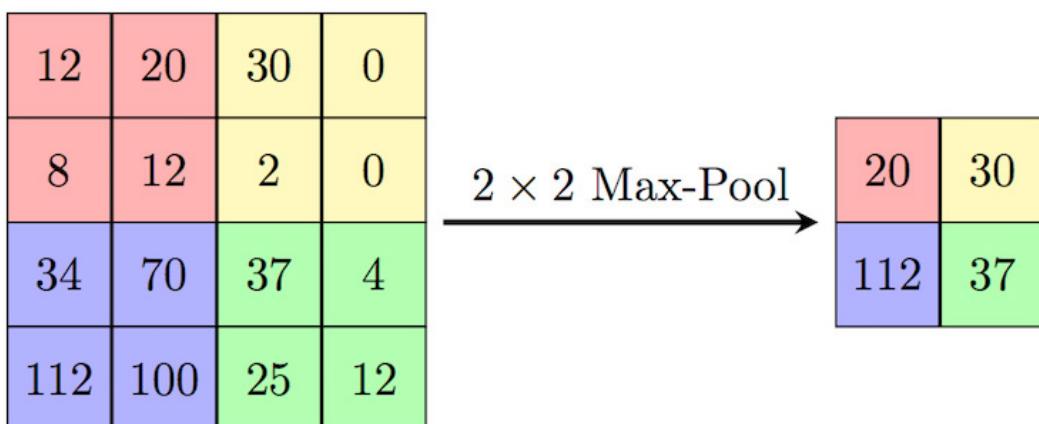
```
weight=tf.get_variable(shape=[kernel_size, kernel_size, input_size, depth])
bias = tf.get_variable(shape=[depth])
conv = tf.nn.conv2d(x, weight, strides=[1, 1, 1, 1], padding='SAME')
conv_relu = tf.nn.relu(conv + bias)
```

池化 (Pooling)

在 CNN 网络中，除了大量的卷积层，我们也会根据需要，插入适量

的池化层。池化层可以用来减少输入的尺寸，从而减少后续网络的参数与计算量。常见的池化操作（如 max pooling, average pooling），通常也可以提供一定的平移不变性。

我们以 max pooling 举例，max pooling 对 kernel size 范围内的所有值取 max，结果作为对应位置的输出。pooling 通常是对每个 channel 单独操作，因此输出的 channel 数和输入相同。池化层和卷积层类似，pooling 操作也可以理解为采用滑动窗口的方式，因此也有和卷积对应的步长 stride 和 padding 等概念。下图所示就是一个 kernel size 和 stride 都为 2 的 max pooling 操作：

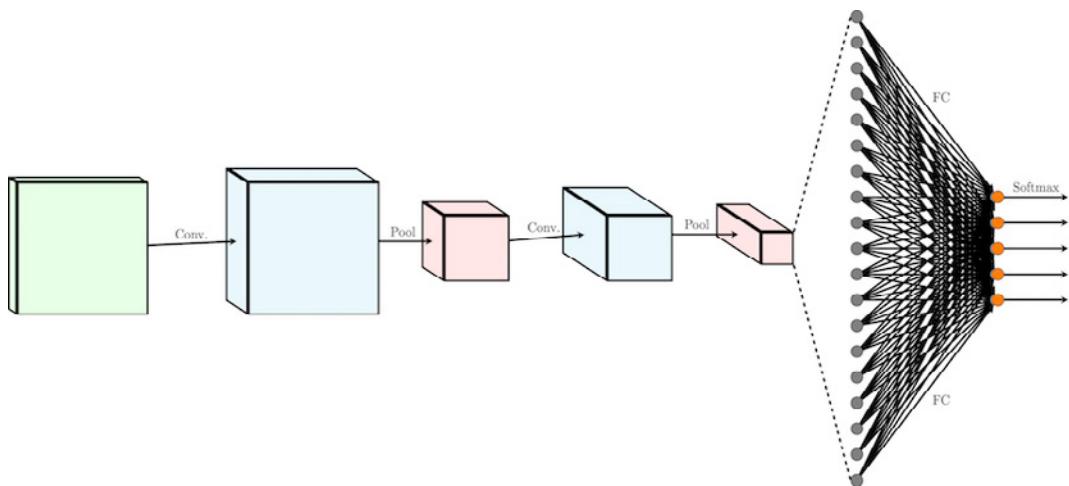


实际当中，池化层的参数有两种比较常见的配置，一种是 kernel size 和 stride 都为 2 的，这种设置池化过程中无重叠区域。另一种是 kernel size 为 3，stride 为 2 的有重叠 pooling。在 tensorflow 中实现池化层也非常简单：

```
tf.nn.max_pool(x, ksize=[1, size, size, 1], strides=[1, stride, stride, 1],
padding='SAME')
```

卷积神经网络的经典网络结构

介绍了卷积神经网络的基本组成模块之后，我们接下来介绍一下卷积神经网络的经典网络结构。从 1998 的 LeNet-5 开始，到 Imagenet 2012 的 AlexNet 模型，再到后来的 VGG 等一系列经典模型，基本都遵从了这个经典结构。



为了清晰，我们省略了卷积和全连接层之后的非线性激活函数。如上图所示，经典的卷积神经网络，可以分为三个部分：

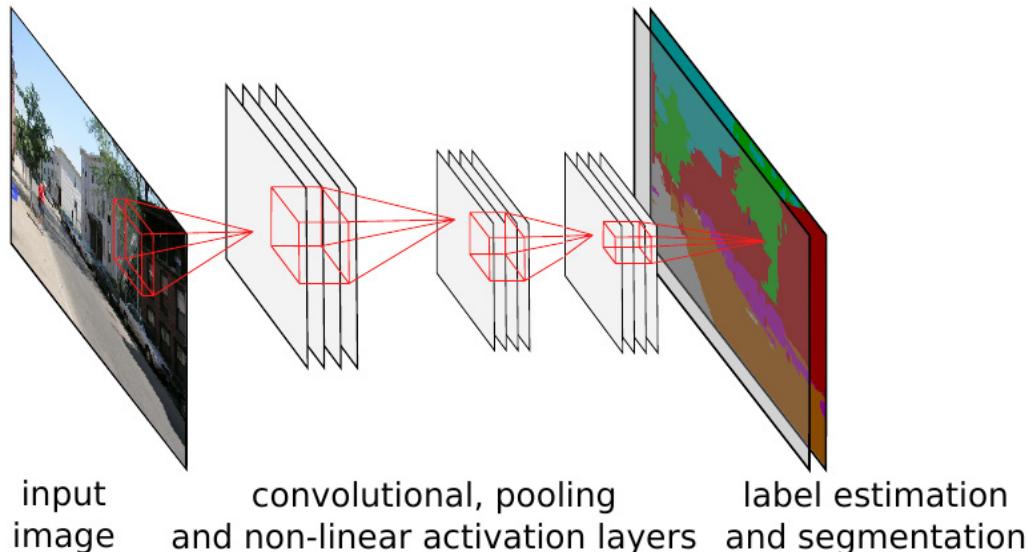
- 一系列级联的 conv+pooling 层（有时会省略掉 pooling 层）。在级联的过程中，输入的尺寸逐渐变小，同时输出的 channel 逐渐变多，完成对信息从低级到高级的抽象。
- 一系列级联的全连接层。在卷积层到全连接层的交界处，卷积层的输出转化成一维的输入送入全连接层。之后根据任务的复杂程度，级联一系列全连接层。
- 最后的输出层，根据任务的需要，决定输出的形式。如多分类问题，最后会接一个 softmax 层。

经典卷积神经网络，可以看作是一个输出尺寸固定的非线性函数。它可以将尺寸为 $H \times W \times 3$ 的输入图片转化为最终的维度为 d 的定长向量。经典卷积神经网络在图像分类、回归等问题上取得了巨大的成功。之后的实战部分，我们会给出一个回归问题的例子。

全卷积网络

经典的卷积神经网络中由于有全连接层的存在，只能接受固定尺寸的图片作为输入，并产生固定尺寸的输出。虽然可以通过使用 adaptive pooling 的方式，接受变长的输入，但这种处理仍然只能产生固定尺寸的

输出。为了克服经典卷积神经网络的这种缺点，在物体分割等输出尺寸可变的应用场景下，我们不再使用全连接层。这种主要计算单元全部由卷积层组成的网络，被称为全卷积网络（FCN）。



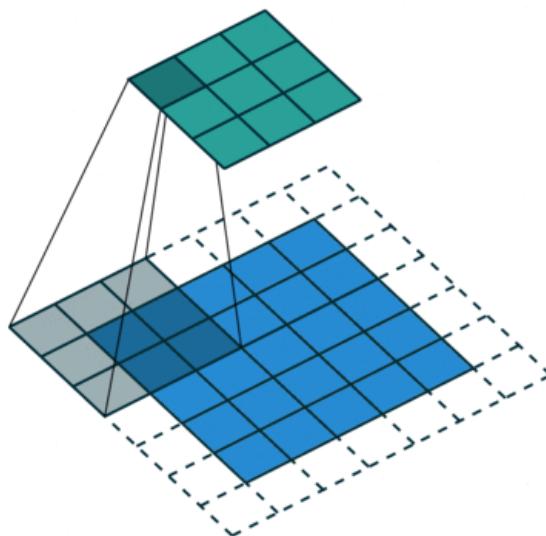
如上图所示，由于卷积操作对输入尺寸无限制，且输出尺寸由输入决定，因此全卷积网络可以很好的处理如分割等尺寸不固定的问题。全卷积网络，可以看成是一种输出尺寸随输入尺寸线性变化的非线性函数。它可以将尺寸为 $H \times W \times 3$ 的输入图片转化为最终维度为 $H/S \times H/S \times d$ 的输出。可以转化为这种形式的监督学习问题，基本都可以在全卷积网络的框架下求解。

反卷积 (Deconvolution)

在全卷积网络中，标准的卷积 + 池化操作，会使输出的尺寸变小。对于很多问题，我们需要输出的尺寸和输入图片保持一致，因此我们需要一种可以扩大输入尺寸的操作。最常用的操作就是反卷积。

反卷积可以理解成卷积的逆向操作。这里我们主要介绍 $\text{stride} > 1$ 且为整数的反卷积。这种反卷积可以理解为一种广义的差值操作。以下图为例，输入是 3×3 的绿色方格，反卷积的 stride 为 2，kernel size 为 3，

padding 为 1。在滑动过程中，对每个输入方格，其输出为对应的 3×3 阴影区域，输出值为输入值和 kernel 对应位置值的乘积。最终的输出为滑动过程中每个输出位置对应值的累加和。这可以看成是一种以 3×3 kernel 值为权重的差值操作。最外边的一圈白色区域无法进行完整的差值操作，因此可以通过设定 padding 为 1，将周围的一圈白色区域去掉，最终的输出尺寸为 5×5 。

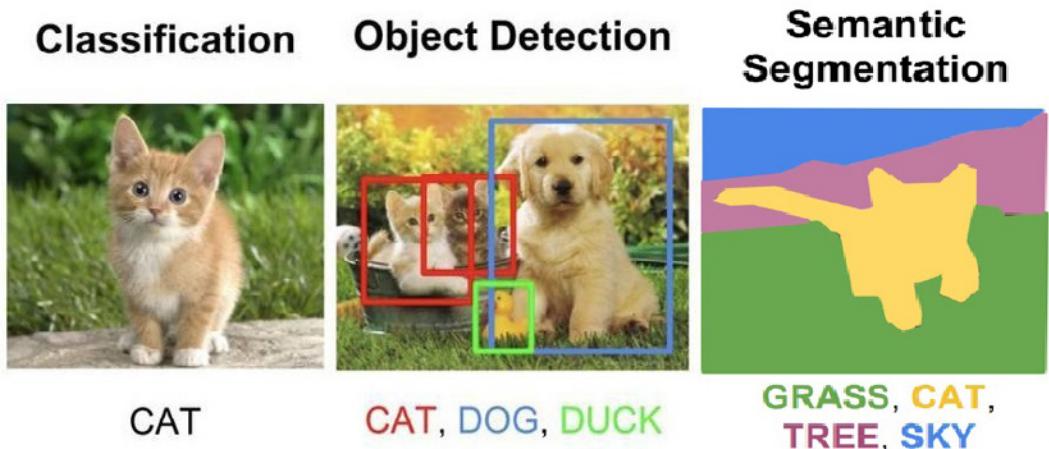


根据上面的描述， $\text{stride} > 1$ 且为整数的反卷积，如果固定反卷积 kernel 的取值为双线性差值 kernel，反卷积可以等价于双线性差值。而通过学习得到反卷积 kernel，相比固定参数的 kernel，可以更好的适应不同的问题，因此反卷积可以看成是传统差值的一种推广。和卷积类似，tensorflow 中已经实现了反卷积模块 `tf.layers.conv2d_transpose`。

卷积神经网络在视觉识别中的应用

CNN 在视觉识别（Visual Recognition）中有着非常广泛的应用。我们接下来以视觉识别中的三大经典问题：分类 / 回归、检测和分割为例，介绍如何用 CNN 解决实际问题。

分类 / 回归（classification/regression）



图像分类是指判别图像属于哪一 / 哪些预先指定的类别，图像回归是指根据图像内容判断图片属性的取值。分类和回归在实际中都有着广泛的应用。从物体分类，人脸识别，再到 12306 的验证码识别等，都可以抽象成标准的分类问题。类似的，人脸的关键点位置预测，人脸的属性预测（如年龄，颜值）等，也都可以抽象为标准的回归问题。目前视觉领域的应用，如果能抽象成输出为定长的分类或者回归问题，在有大量训练数据的情况下，通常都可以采用之前介绍的经典卷积神经网络框架解决。

检测 (detection)

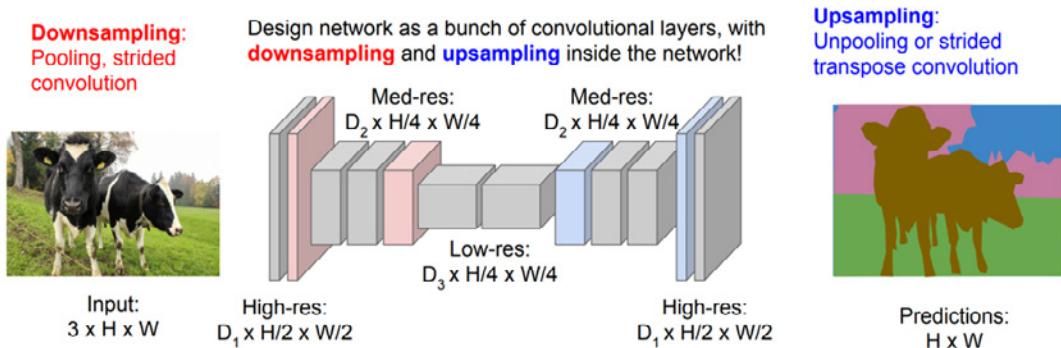
检测问题通常是指判断出图片中是否有物体，以及该物体的位置。检测有 one-stage 和 two-stage 的方法。因为篇幅关系，我们重点介绍在 FCN 框架下的 one-stage 方法。

按之前的介绍，FCN 可以看作是将 $H \times W \times 3$ 的输入图片，转化为 $H/S \times W/S \times d$ 输出的非线性函数。在 FCN 的框架下解决检测问题，我们可以预测每一个输出位置是否有物体，以及物体左上角、右下角相对于当前输入位置的偏移。这样对每个输出位置，需要 5 维的向量来表示是否有物体，即 $d=5$ 。定义了网络的输出之后，我们人工构造出对应的 ground truth，之后在监督学习的框架下，通过定义损失函数 (l2 loss) 并进行反向传播，进行参数的学习。

分割 (segmentation)

分割问题是指给出图片中每个像素点的类别。基于 FCN 的分割方法和上面介绍的 one-stage 的检测方法非常类似。对一个多分类的分割问题，对输出的每一个位置，我们可以判断其所属的类别。在 FCN 的框架下，对于 N 分类问题，输出为 $H/S \times W/S \times N$ 。之后通过反向传播的方式进行训练。分割和检测问题有一个区别是我们有时需要得到和输入图片同样大小的输出 ($H \times W \times N$)，但卷积神经网络为了加速，通常会添加 pooling 层，减小中间卷积层的尺寸。如下图所示，为了保证输出的尺寸满足要求，我们可以在网络的最后添加反卷积层进行补偿，从而获得更大尺寸的输出。

Semantic Segmentation Idea: Fully Convolutional



实战：人脸关键点检测



Facial Keypoints Detection

Detect the location of keypoints on face images
175 teams · 10 months ago

人脸关键点检测是现在视觉领域比较成熟的一项应用，是活体检测，人类美化，人脸识别等高级应用的基础。本文最后通过一个人脸关键点检测的例子，展示如何用 Tensorflow 实现图像回归类的应用。实验数据集采用 Kaggle 比赛中的 [Faical Kerypoints Detection](#) 数据集。该数据集包含了

7094 张训练图片和 1783 张测试图片。数据集中的每一张人脸都有 15 个关键点的标注，图片的尺寸为 96x96。

L2 距离回归

Kaggle 比赛的目标是预测人脸上 15 个关键点的坐标，总共 30 个 float 值，属于标准的回归问题。我们选择采用最常见的 L2 距离，作为优化的目标。和第一篇文章中神经网络模型的代码结构一样，我们将代码分成了 3 个主要模块，分别是 Dataset 模块，Net 模块和 Solver 模块。

模型结构

- `inference` 我们在 `inference` 函数中定义网络的主体结构。因为模型会重复用到全连接层和卷积层，因此我们将他们封装成函数 `linear_relu` 和 `conv_relu`，从而方便复用代码。网络结构上我们采用了比较简单的 3 层卷积，2 层全连接的结构。卷积层的输出通过 `tf.reshape` 转化成了全连接层可以接受的格式。因为是回归问题，我们直接将最后一层全连接层的结果作为输出。
- `loss` 为了简单，对于标准的回归问题，我们使用 `mse` 作为损失函数 `tf.reduce_mean(tf.square(predictions - labels), name='mse')`

测试时，我们依旧使用 tensorflow 提供了 `tf.metrics` 模块，自动完成对每个 batch 的评价，并将所有的评价汇总。在这个例子里，我们是解决回归问题，因此可以使用 `tf.metrics.mean_squared_error` 计算均方误差。

```
def linear(x, output_size, wd=0):
    input_size = x.get_shape()[1].value
    weight = tf.get_variable(
        name='weight',
        shape=[input_size, output_size],
        initializer=tf.contrib.layers.xavier_initializer())
    bias = tf.get_variable(
        'bias', shape=[output_size], initializer=tf.constant_initializer(0.0))
    out = tf.matmul(x, weight) + bias
    if wd != 0:
```

```
weight_decay = tf.multiply(tf.nn.l2_loss(weight), wd, name='weight_loss')
tf.add_to_collection('losses', weight_decay)
return out

def linear_relu(x, output_size, wd=0):
    return tf.nn.relu(
        linear(x, output_size, wd), name=tf.get_default_graph().get_name_scope())

def conv_relu(x, kernel_size, width, wd=0):
    input_size = x.get_shape()[3]
    weight = tf.get_variable(
        name='weight',
        shape=[kernel_size, kernel_size, input_size, width],
        initializer=tf.contrib.layers.xavier_initializer())
    bias = tf.get_variable(
        'bias', shape=[width], initializer=tf.constant_initializer(0.0))
    conv = tf.nn.conv2d(x, weight, strides=[1, 1, 1, 1], padding='SAME')
    if wd != 0:
        weight_decay = tf.multiply(tf.nn.l2_loss(weight), wd, name='weight_loss')
        tf.add_to_collection('losses', weight_decay)

    out = tf.nn.relu(conv + bias, name=tf.get_default_graph().get_name_scope())
    return out

def pool(x, size):
    return tf.nn.max_pool(
        x, ksize=[1, size, size, 1], strides=[1, size, size, 1], padding='SAME')

class BasicCNN(Net):
    def __init__(self, **kwargs):
        self.output_size = kwargs.get('output_size', 1)
        return

    def inference(self, data):
        with tf.variable_scope('conv1'):
```

```
conv1 = conv_relu(data, kernel_size=3, width=32)
pool1 = pool(conv1, size=2)

with tf.variable_scope('conv2'):
    conv2 = conv_relu(pool1, kernel_size=2, width=64)
    pool2 = pool(conv2, size=2)

with tf.variable_scope('conv3'):
    conv3 = conv_relu(pool2, kernel_size=2, width=128)
    pool3 = pool(conv3, size=2)

# Flatten convolutional layers output
shape = pool3.get_shape().as_list()
flattened = tf.reshape(pool3, [-1, shape[1] * shape[2] * shape[3]])

# Fully connected layers
with tf.variable_scope('fc4'):
    fc4 = linear_relu(flattened, output_size=100)

with tf.variable_scope('fc5'):
    fc5 = linear_relu(fc4, output_size=100)
with tf.variable_scope('out'):
    prediction = linear(fc5, output_size=self.output_size)

return {"predictions": prediction, 'data': data}

def loss(self, layers, labels):
    predictions = layers['predictions']
    with tf.variable_scope('losses'):
        loss = tf.reduce_mean(tf.square(predictions - labels), name='mse')
    return loss

def metric(self, layers, labels):
    predictions = layers['predictions']
```

```

with tf.variable_scope('metrics'):

    metrics = {

        "mse": tf.metrics.mean_squared_error(
            labels=labels, predictions=predictions)}

    return metrics

```

Dataset

```

images = np.vstack(df['Image'].values) / 255. # scale pixel values to [0, 1]
images = images.astype(np.float32)

label = df[df.columns[:-1]].values
label = (label - 48) / 48 # scale target coordinates to [-1, 1]
label = label.astype(np.float32)

def parse_example(example_proto):

    features = {

        "data": tf.FixedLenFeature((9216), tf.float32),
        "label": tf.FixedLenFeature((30), tf.float32, default_value=[0.0] * 30),
    }

    parsed_features = tf.parse_single_example(example_proto, features)
    image = tf.reshape(parsed_features["data"], (96, 96, -1))

    return image, parsed_features["label"]

dataset = tf.contrib.data.TFRecordDataset(files)
dataset = dataset.map(self.parse_function)

```

Dataset 部分，我们使用了 tensorflow 推荐的 tfrecord 格式。通过 TFRecordDataset 函数读取 tfrecord 文件，并通过 parse_example 将 tfrecord 转换成模型的输入格式。tfrecord 作为一种定长格式，可以大大加快数据的读取速度。特别在使用 GPU 时，可以防止数据 io 成为性能的瓶颈。

Solver

通过模块化的设计，我们可以完全复用第一篇文章中的 Solver 代码，而不需要任何修改，进而提高代码的复用效率。

实验结果

```

file_dict = {
    'train': os.path.join(args.data_dir, 'train.tfrecords'),
    'eval': os.path.join(args.data_dir, 'test.tfrecords')
}

with tf.Graph().as_default():

    dataset = Dataset(
        file_dict=file_dict,
        split='train',
        parse_function=parse_example,
        batch_size=50)

    net = Net(output_size=30)

    solver = Solver(dataset, net, max_steps=200, summary_iter=10)
    solver.train()

```

封装好借口之后，我们可以通过上面简单的代码，完成模型的训练。

下图是 tensorboard 中可视化的网络结构、loss 的统计以及模型在测试图片上的效果

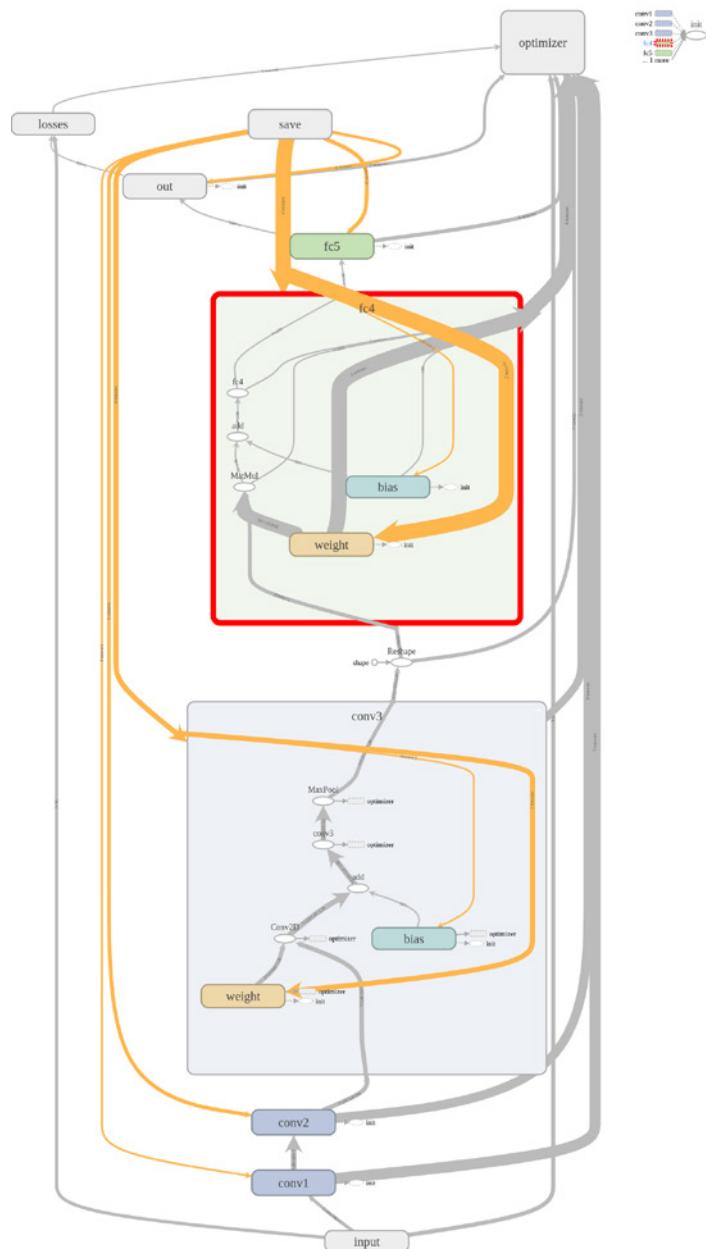
```

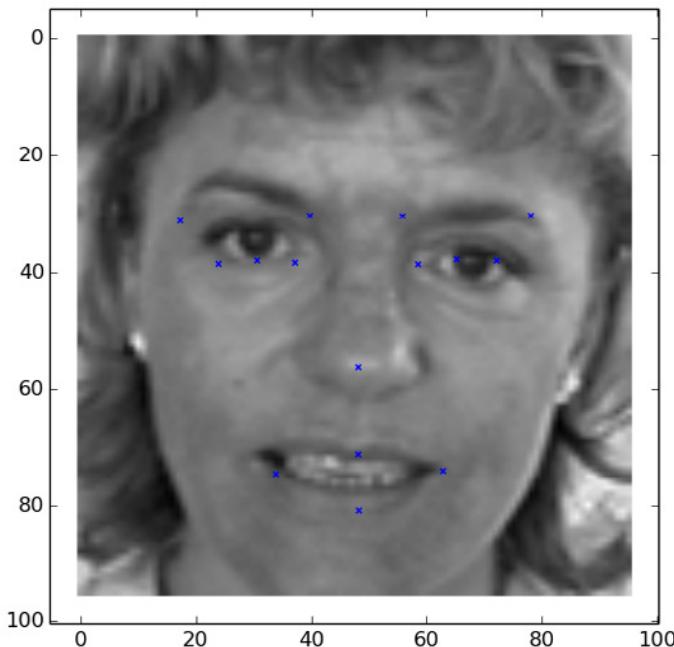
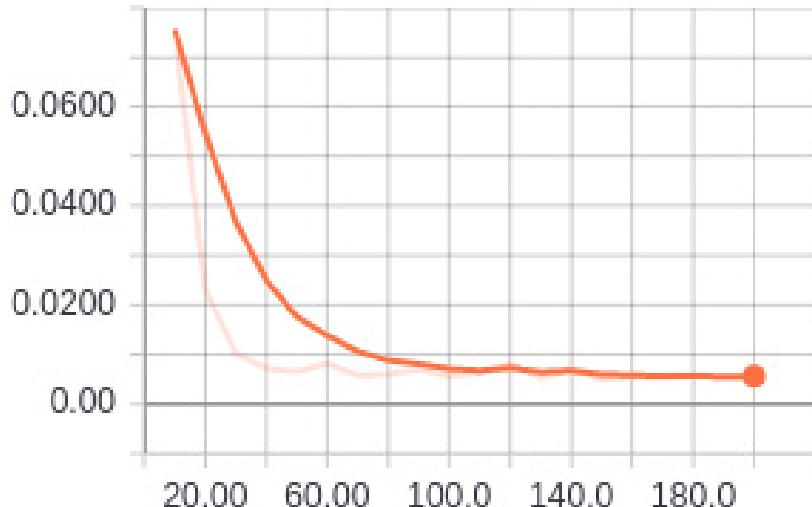
step      10: loss = 0.0756 (136.2 examples/sec)
step      20: loss = 0.0230 (155.2 examples/sec)
step      30: loss = 0.0102 (149.1 examples/sec)
step      40: loss = 0.0071 (125.1 examples/sec)
step      50: loss = 0.0065 (160.9 examples/sec)
step      60: loss = 0.0081 (171.9 examples/sec)
step      70: loss = 0.0058 (148.4 examples/sec)
step      80: loss = 0.0060 (169.4 examples/sec)
step      90: loss = 0.0069 (185.4 examples/sec)
step     100: loss = 0.0057 (186.1 examples/sec)
step     110: loss = 0.0062 (183.8 examples/sec)
step     120: loss = 0.0080 (170.3 examples/sec)
step     130: loss = 0.0052 (185.8 examples/sec)
step     140: loss = 0.0071 (184.3 examples/sec)
step     150: loss = 0.0049 (170.7 examples/sec)

```

```
step 160: loss = 0.0056 (178.7 examples/sec)
step 170: loss = 0.0053 (173.2 examples/sec)
step 180: loss = 0.0058 (172.6 examples/sec)
step 190: loss = 0.0053 (172.5 examples/sec)
step 200: loss = 0.0056 (188.1 examples/sec)
```

mse: 0.140243709087



loss

可以看到，一个 3 层卷积 +2 层全连接的经典卷积神经网络，就可以很好的解决人脸关键点检测的问题。在实际中，我们可以使用更复杂的网络和一些其他 trick 来进一步提高模型性能。

作者简介

董健，360 高级数据科学家，前 Amazon 研究科学家。目前主要关注深度学习、强化学习、计算机视觉等方面的科学和技术创新，拥有丰富的大数据、计算机视觉经验。曾经多次领队参加 Pascal VOC、ImageNet 等世界著名人工智能竞赛并获得冠军。

博士期间在顶级国际学术会议和杂志上发表过多篇学术论文。从 2015 年底加入 360 至今，董健作为主要技术人员参与并领导了多个计算机视觉和大数据项目。

完整代码[下载](#)。



扫码关注InfoQ公众号

Geekbang | **InfoQ**
极客邦科技