

算法工程师 计算机算法设计 深度学习（Deep Learning） 卷积神经网络（CNN）

算法（书籍）

关注者

4,729

被浏览

585,741

在做算法工程师的道路上，你掌握了什么概念或技术使你感觉自我提升突飞猛进？

圆桌收录 不只代码 | 程序员 A/B 面 >

对偶问题：在做程序员的道路上，你掌握了什么概念或技术使你感觉自我提升突飞猛进？

关注问题

写回答

+ 邀请回答

👍 好问题 505


💬 2 条评论

➦ 分享

⋮

收起

^


 登录后你可以

🔒 unlimited 看优质回答

💬 私信答主深度交流

🔖 精彩内容一键收藏

登录

 下载知乎客户端

与世界分享知识、经验和见解

查看全部 87 个回答

 金赢若愚

华盛顿大学 人工智能博士在读

3,500 人赞同了该回答

在科研中训练到的思维方式，是我最宝贵的成长。

一. 反馈的闭环

在一个研究任务中，我首先尝试了方法A，没搞定，于是改用方法B。组里大哥问：为什么改方法B。我说A没效果，或许B能work。大哥继续问：为什么A不work。我说或许A不适合这个问题？大哥说：当你有一个尝试，你一定要知道它为什么work以及为什么不work。每次不work了你就换另一个方法，那另一个方法就能work吗？这不是科研，是赌博，是瞎试。你只有知道为什么一个方法有效或无效，何时有效何时无效，你才能增进对这个问题的理解，然后基于此提出有价值的策略。

我的思考是，既然做了一件事，就一定要得到反馈，要搞清楚哪里做得好哪里不好，这样这个尝试所投入的时间才是有效的。不然就是在碰运气，如同做题不对答案，如同训练模型不算loss不做backprop。

二. 把炼丹技术推广到生活中

机器学习的很多技术都与现实世界的概念相互呼应。

在深度学习里，模型掉入局部最优，就是生活中的内卷，就是在狭窄的赛道上追求极致。摆脱内卷就是跳出局部寻求全局最优的过程。破局之道是尝试新事物，或增加训练数据。

 JumpServer

《新一代堡垒机建设指南》

免费下载



广告

关于作者

 金赢若愚

华盛顿大学 人工智能博士在读

回答

34

文章

4

关注者

17,016

关注他

发私信



算法（书籍）

《算法（第4版）》全面讲述算法和数据结构的必备知识，具...

查看更多 >

被收藏 2,222 次

男士的自我修养389 人关注
不思量 创建

自我修养390 人关注
南山 创建

推荐系统&计算广告310 人关注

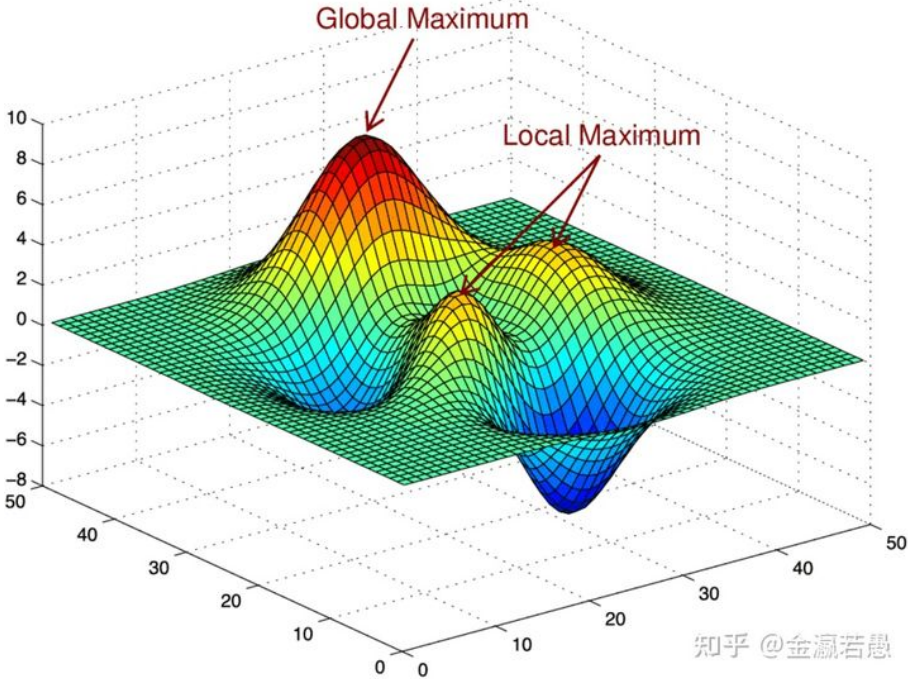
▲ 赞同 3500 ▼

💬 108 条评论

➦ 分享

★ 收藏

♥ 喜欢



与人交流、观察和学习他人可以避免闭门造车。与他人交流就是深度学习里的增加训练数据。进入好的学校好的公司就是提高训练样本的质量：在label准确时，学得轻松。

前面讲的“构建闭环”，就是关注backprop时的梯度。你不能攒了特别多事情后再去反思或寻求反馈，这个反馈链太长，要么梯度消失，要么梯度爆炸，无法有效学习。类似对loss的求导过程：**你必须清晰地看到因果链条，才能做到从结果倒推原因，进而优化自己。**

已经做得很好的事情没必要重复。此时应该挑战新的项目，保持自己在学习的状态。这就是hard data mining. 遇到坏人后就认为全人类没救了，是overfitting。对应在机器学习里，在 unbalanced dataset上学习，要想到用weighted loss。

挫折易使人变得复杂、内心冲突多。如果能用简单的形态存在，我们或许应警惕过分复杂。机器学习里，解决小问题硬上大模型是一种粗暴且没技术含量的办法。用不必要大的模型是对探求事物本质的逃避，科研如此，生活亦然。你以为模型练成了，其实它学的是shortcut，因此有泛化能力差的问题：variance大，不robust（内心不稳）。同时，模型太复杂就不容易理解，遇到bug不容易诊断病因（内心不易平稳和愉悦）。但避免复杂不是要当巨婴，不是抗拒成长。当任务复杂、训练数据也大的时候，就必须要大模型。核心是模型复杂度（心智复杂度）要与任务复杂度和数据量（阅历）匹配，才是健康的。

三. 交流，交流，再交流

组里的女神姐姐教导我：“做research要多和人交流，多去听别人的paper reading，也把你读到的论文和想法讲给别人听。因为在讨论的过程里你会意识到未曾发现的问题。尝试给人讲明白的过程里，你的思路也会越发清晰。”道理我都懂，每次我头点得像敲鼓一样，但也没坚持做到。

为什么做不到？大概是心理包袱，总想着搞出一个牛X闪闪的东西后再展示给别人，不然觉得丢脸。或者总有一种再试一下就能搞定的错觉。这种错觉和我炒股票的风格如出一辙。

直到后来我看到越来越多平淡无奇的项目都逆袭了，我理解到，事物的发展都有必然的过程，要尊重其发展规律，不要总想直接搞个大新闻。高效的科研**要主动的寻求身边的资源，争取让良师益友把时间花在自己身上**，如果自己不主动争取，再照顾你的人也不能像亲妈一样耳提面命，他们毕竟不好逼迫你。具体的，**应该多把自己读懂的论文讲给别人，在你给他人创造价值的同时，你也使他们帮助你这件事更容易了：他们只有懂了你懂的，才能给你最有效的建议，你才能借用他们的头脑思考。**

我发现当我积极主动之后，身边的人都默默支持起我来。稻盛和夫^o说：“心不唤物，物不至”。

四.「焦虑动力」模型

帽子戏法
梅小西 创建

Life online 攻略
尚萌 创建

35 人关

27 人关注

相关问题

- 走算法工程师,机器学习,深度学习方面的工作,不读研好吗? 15 个回答
- 算法工程师的落地能力具体指的是什么? 35 个回答
- 机器学习算法工程师如何自己接项目单干、并赚取比上班多得多的收入? 16 个回答
- 算法工程师真的是调参侠吗? 12 个回答
- 本科。想去做推荐算法或者机器学习算法工程师请问阿胶如何准备明年春招? 7 个回答

相关推荐

- 深度学习之模型设计：核心算法与案例实践
24 人读过 阅读
- 人工智能算法（卷3）：深度学习和神经网络
13 人读过 阅读
- 深度学习案例精粹
艾哈迈德·曼肖伊
31 人读过 阅读

知乎 健康 × 知乎 心理

健康最懂她

北京大学国际医院 × 北京大学首钢医院联合主办

刘看山 · 知乎指南 · 知乎协议 · 知乎隐私保护指引

应用 · 工作 · 申请开通知乎机构号

侵权举报 · 网上有害信息举报专区

京 ICP 证 110745 号

京 ICP 备 13052560 号 - 1

京公网安备 11010802020088 号

互联网药品信息服务资格证书

(京) - 非经营性 - 2017 - 0067

服务热线: 400-919-0001

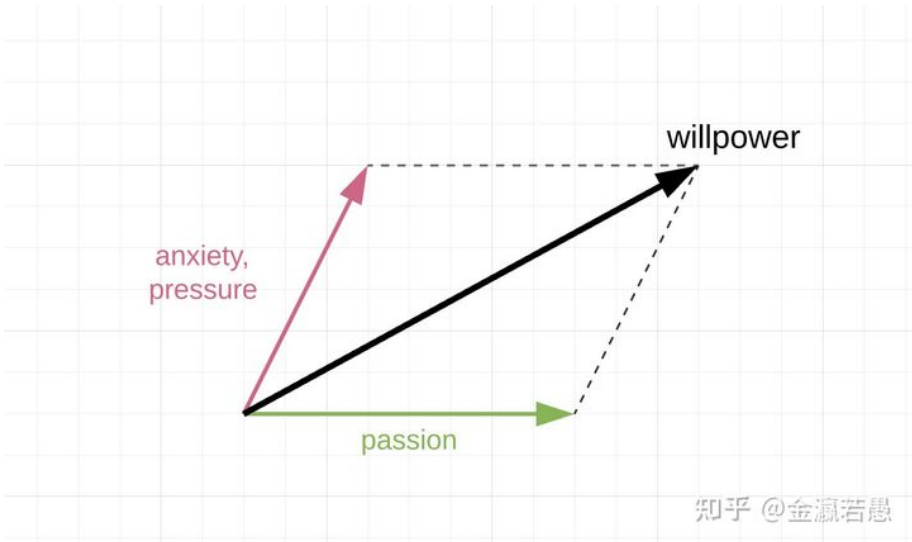
违法和不良信息举报: 010-82716601

举报邮箱: jubao@zhihu.com

儿童色情信息举报专区

信息安全漏洞反馈专区

勤奋上进、自我驱动、有行动力，本质是内心动力充沛的精神状态。动力不足时，我们懒惰、拖延、自控力差。用向量分解的思维模型，我们可以看清「动力」的本质。



Willpower=动力; Anxiety=焦虑; Pressure=压力; Passion=热情

• 动力的分量是 1.焦虑/压力 和 2.热情。动力是二者的合力

带着上面这句话，我们可以举出奋斗者的两种极端状况。一种由焦虑主导，被绩效和deadline催促，常处于不安和压力之中。希望获得他人的认可，如果有了过失，会感到煎熬；另一种是热情主导的。专注、投入，不易为外界干扰和诱惑所动。在做事的过程中，能收获成就感和喜悦，认为工作与个人追求比较重合。

警惕“动力”过分依赖“焦虑和压力”的状况。这个动力来源并不持久和可靠。随着年龄增加，体力和心力会下降，且随着阅历增加，阈值变高，很多事会不再觉得重要。那时就会失去动力。

分量之间存在代偿。盲人的耳朵往往很敏锐。如果动力过多依赖焦虑和压力，则留意到热情和乐趣就更难。可能并不是你不热爱，你只是太紧张了而已。

五. 独立思考

我对独立思考这件事体会最深的就是决定要不要读PhD时。拿到UW的录取后，因为微软组里就很多博士、教授，我自然就去请教他们问要不要读博，毕竟五年是一笔不小的投入。当时有一部分人说值得一读，另一些人说专心做事业可能有更大回报。

有趣的是，当我签下offer后，所有的人，不论之前给了什么建议，都由衷恭喜和认可我的决定，并和我畅想毕业后Dr. Jin[走向人生](#)^Q(996)巅峰的画面。

那之前他们的建议是真心的吗？当然是真心的。签offer后的认可也是真心的。于是我理解到，很多事情，做与不做都能找出道理。**他人给建议时，也会考虑到我们的情绪，让我们不论怎样选择都有台阶下。**因此，我们要保持独立思考，要对自己的决定负全责。

建议的推导逻辑比结论重要。他人建议的正确用法是让自己看问题多个角度，减少信息差，而不是直接取其结论，让他人代为做选择。

六. 把工作当做一个二阶优化过程

不管title是算法工程师还是应用科学家，产出都是代码及其体现的知识产权。我们工作的过程就是优化这个代码及知识产权的过程。但这太basic。

我们不仅要优化产品，也要优化产出产品的过程：一阶优化，是优化代码质量。二阶的优化，是优化工作过程，这个工作过程是代码质量更上一层楼的原因，是原因的原因，是二阶导数。

类比一下：为了走得远，你可以优化速度（路程的原因是速度），更进一步你可以优化加速度（速度的原因是加速度）。不断向上溯源，能解决根本问题、通用问题。





回到优化工作流程上，我们不断问自己：在我工作的过程中，有哪些是重复工作？有哪些可以更高的完成？有哪些可以被拎出来整理出可以复用的，于是后面就不再需要花心思重做或者检查其bug？经过一两年我攒出了一套自己的代码库，很多被重复使用的部分（如用matplotlib画各种图，各种数据预处理）就都可以直接copy paste。这就形成了复利效应——时间越长，这些整理出来的代码片段就创造越高的价值。

七. 学习些销售意识

销售是一门大学问，值得广大猿类学习。

• 像设计产品外包装一样设计履历

一个畅销的产品必须有个特色，这个特色与其他竞品形成了差异化竞争，才能有自己的市场，卖上个好价格。比如[始祖鸟](#)^o这个牌子的衣服颜值很一般，但它的防水面料好，就拿下了户外市场。一个中庸的什么都不差也什么都不突出的产品就很难被你记住。同时，产品的优秀比不上品牌的优秀，如果产品的价值凝练成了品牌，那这个品牌本身就值高价。比如我们会为了logo付费。

我们的职业发展，说得现实些，也是希望自己在市场上有个好价格。为此，我们不需样样精通，但必须有一个具有代表性的，能拿得出手的技能。比如，我就是要和所有人不同，去学远古技术汇编语言。那只要市场出现了对这个技能的需求，你就能有极大的[定价权](#)^o。当我们深耕一个领域很久，又写了很多优质的博客，那你的名字就成为了品牌，可以帮助公司[招贤纳士](#)^o，等等。

我的老板兼导师也曾说：你做paper要想怎样把你的paper卖出去。你要站在消费者角度想，他们为什么要花时间去读你的文章？你的论文有没有创意，能给他们带来什么价值？写作语言，图表美观程度，就是卖相。这和销售很像。

我们的简历就是商品的成分表——在决定选择什么项目时，不妨想想这会在简历里留下一行什么样的记录，会不会帮你抬高自己的职业价值。

• 用产品思维规划项目

科研项目的立项过程和产品策划非常像。第一步都是要做survey，了解清楚目前技术有哪些分支，是什么历史契机促成了某个技术的出现（比如有了新的数据集），不同的技术优缺点在哪（A更准确，B速度更快，C不需要很多训练数据，等等）。然后了解这个任务的定义，metric（关键指标）是什么，也就是搞清楚游戏规则，知道大家在比些什么，头部的玩家都是谁等等。不了解清楚，后面的一切都是错的。

导师们经常问我的一个问题就是“这个task的upper bound是什么”。我说我知道state-of-the-art是什么，还要知道upper bound吗？导师说你在开始一个尝试前，要知道还有多少空间留给你去做。如果SOTA（目前最好的）已经比较接近upper bound了，你就很难再往上去攻了，你可能找个别的任务更有成就感。**你跳进一个赛道前，要对其发展到了什么阶段有些概念。**

八. 关于忙碌

在上面各种催人上进的内容之后，我再来分享一个故事来结尾。

一次公司的一位高层前辈见面，临走前我问他能不能给我个过来人的建议。

我以为他会给我讲一些勤奋工作的态度、人生规划的经验、或者给我打打鸡血这种内容。

而前辈只是指了指办公室书架上的一排没拆封的乐高跟我说，我给你一个建议，就是年轻时多花点时间在自己的嗜好上。我年轻时喜欢乐高，但因为一直忙于工作，并没有花很多时间在我这个小小的嗜好上。现在我时间多了，但对乐高也不再那种热情了，现在只把他们摆起来，包装都没拆。所以年轻的时候，你可以花点时间在自己的爱好上。不能只有忙碌的工作。

很片面，欢迎指正。

编辑于 2021-10-18 06:02

更多回答

▲ 赞同 3500 ▼

● 108 条评论

➤ 分享

★ 收藏

♥ 喜欢



夕小瑶

机器学习话题下的优秀答主

835 人赞同了该回答

我以为回答区会有这张图：



结果竟然没有！知乎果然没有让我失望，还是一如既往的认真、严谨、有逼格。咳咳，作为一枚知乎算法圈的老司机小可爱，感觉自己有必要来添砖加瓦一下。

相信我，深刻理解数据集的意义与数据集构造与迭代技术，你会从算法小白突飞猛进到即将入门！
(认真脸)

首先贴出我的算法工程师升级打怪路线：

1. 最菜的算法工程师靠调学习率
2. 次菜的[算法工程师](#)靠试新模型
3. 合格的算法工程师会懂得做数据
4. 不错的算法工程师还会用新paper优化实际问题
5. 优秀的算法工程师能数据、模型、策略joint design&iterate
6. 顶级的算法工程师能颠覆行业方法论

仅供参考，请勿对号入座！尤其是正在看本文的技术leader们，不要将自己的小弟对号入座，毕竟从1到6的形状是个倒金字塔(´▽`)。

其实从这里可以看出，1和2其实大部分实习生和应届校招生都能做到，而3就是完成学校->工业界的重要转折点，也就是**做数据**。

关于数据

学生思维中的算法工程师日常可能是这样的：

天上掉下来/老板拍过来/网上download下来一个**训练集、测试集**，然后我开始调模型&调参，调好了就上线了。

尤其是那些整天吹嘘算法岗门槛低，以为跑个BERT刷个榜单就入门了nlp的人，不是学生就是半路转行的调参侠。

当你开始意识到数据的重要性了，不敢说技术突飞猛进吧，至少可以承认你马上就要入门这一行了。比如你开始有以下意识：

1. 嗯，我不能只关注离线测试集指标涨不涨，我还要判断测试集靠不靠谱，包括采样/数据分布的线上一致性、时效性、标注正确率、评测方差/置信度等。
2. 嗯，我不能对着同一个测试集做大量没道理的炼丹，比如暴力调参、疯狂改随机种子、**暴力乱加策略**、暴力魔改、暴力增删改查等。我知道大量的无意义超参调整，只会变相的让模型用超参过拟合这个测试集。最终结果往往是，向上汇报猛如虎，线上用户用脚投票。
3. wok，准确率从90%一下子涨到99%了！肯定不是我nb，绝对是出bug或者标签泄漏了！
4. 嗯，新老方法都有其存在的意义和发挥作用的阶段，都是不可缺的。
5. 嗯，从1到2与从0到1的解法有很大区别。

就至少跟200元/天的实习生划清界限了。。。

那么，具体来说，做数据具体要怎么做呢？或者说需要积累哪几

赞同 3500

108 条评论

分享

收藏

喜欢



简单来讲：

1. 标注标准
2. 采样策略

关于标注标准，这其实是个比较吃业务经验的事情。

真实的业务场景数据是非常dirty的，即使是情感分类^o这种看起来非黑即白的NLP任务，在流量大的真实业务场景中，都会面临大量的边界样本^o和人都要分辨半天，甚至需要足够学历、阅历才能分辨出情感极性的样本（想象一下知乎评论区那些“阴阳怪气”的评论，可能你都不知道对方是在骂你还是夸你）。

此外，哪怕是同一条样本，同样的分类任务，业务场景稍有变化，那么你期待的标签可能会完全相反。比如，句子“oppo最新款手机多少钱”与句子“vivo最新款手机多少钱”的文本相关性，在搜索场景，就是妥妥的负例，因为用户的出发点是获取真实知识，你如果给用户返回vivo的价格，那用户会有一种被欺骗感。

但是，在搜索广告场景，却完全可以作为正例，因为本身vivo和oppo的大众认知相对比较近，用户在搜索广告场景的出发点可能是买一部不错的安卓机，并且用户觉得oppo可能是不错的选择，那这时候你给出vivo的信息，用户往往不会反感，甚至可能因为发现vivo更合适而下单。

当然了，这种品牌实体的替换导致的文本相关性结论并不是永久成立的，比如同样搜索广告，用户问“劳莱斯最新款多少钱”，结果你出了一条“五菱最新款价格”，那用户就会感觉受到了侮辱。。所以深刻理解具体业务场景的优化目标，对于制定正确的标注标准是十分必要的。当然，标注标准很大程度上取决于产品标准，而你就是要将这个产品标准深刻理解且转变成众包平台人员也能轻松看懂的标注标准。

一旦标注标准导向错了，坐拥百万标注数据和100层的预训练模型也可能原地踏步。相反地说，如果标准做得好，那将大大提升标注数据的有效性，每周新返回的标注数据带来的增益可能比你花式炼丹^o一个月都来的有效。

关于采样策略，这个则同时跟业务目标、技术选型与模型水平相关，在不同的场景有不同的策略。

比如你到了要端到端解决问题的阶段，又能做到大规模标注，那可以考虑直接对齐线上真实分布做同分布采样+端到端标注。但是大部分复杂业务，往往是pipeline的方式，则要考虑对齐上一级分布。涉及到具体场景和阶段时，又会根据这个大原则去微调采样策略，比如精排^o问题中如何 mining 负例，多路召回时如何均衡采样来优化粗排等，在真实分布的基础上做些微调同样可能带来肉眼可见的提升。

除了标准和采样问题外，数据问题上还可以组合大量预处理策略和训练策略，耦合上模型的特点和输入分布偏好，又是一波空间。总之，在当前的大框架下，数据空间往往比模型空间大得多，这也是当下算法工程师炼丹水平高低的一个重要分水岭。

更进一步

说完了倒金字塔最重要的一层，我们继续往下深入。

3到4的过程相对来说不是很难。如果你有刷paper的习惯，配合着高手速，脑子别太笨，那剩下的20%的疑难问题你也能解的七七八八了。即使不怎么刷paper，在靠谱的算法团队里苟着，通过耳濡目染也能get到不少求解疑难算法问题的高端姿势（误）。能做到这一步的，在大厂核心团队里也称得上不错了。如果再有一些软实力加成，至少不会担心没人要。

而4到5的过程，则需要一些顶层设计能力和工作年限积累。首先要做有挑战性的业务，那些挖个字典、跑跑textcnn、finetune下BERT就完成目标的业务，说难听点都是demo，至少算法肯定不是这个业务的核心竞争力。当然了，这种挑战大的场景主要集中在搜索、推荐、广告等核心赛道上，这也是为什么有钱的中小厂都爱挖大厂核心部门的高T去做算法leader乃至CTO，因为顶层设计能力确实纯靠聪明、看paper和小打小闹的业务是学不来的。怎样的策略方案会预期达到怎样的业务效果，消耗多少成本，有哪些风险，甚至这个事情当下能不能做，应不应该做，前置环节是什么，下个阶段要做什么等，这些问题都要有足够的判断力。毕竟，哪个老板愿意拿着几百上千万的年薪让你去他的命根子业务上成长试错呢？

而5到6的过程，多拍脑袋吧，也需要环境、机遇。如果你做到了友位。。。

▲ 赞同 3500 ▼

● 108 条评论

➤ 分享

★ 收藏

♥ 喜欢

发布于 2021-03-11 23:22

赞同 835

36 条评论

分享

收藏

喜欢

收起 ^

 **DLing** 

一个只在地上走不在天上飘的深度学习者...

564 人赞同了该回答

突飞猛进不敢当，但是从事深度学习图像算法这几年来，却是也有一些感慨。

1. 数据放在第一位，成也数据，败也数据。深刻认识数据的重要性，把数据集维护好，数据量够了，再谈后面的模型优化，数据都不干净，用再好的模型，也不会出好的结果。

2. 启动开发前，多问问自己有没有了解这个业务，目前定的方案还有没有盲点没有考虑到，毕竟启动开发需要准备各种数据集，耗时长且需要一定的人员和经济投入，如果开发过程中或者测试阶段发现方案不合适，这时候推倒重来的话，就DT了。

3. 算法工程师并不只是调包侠，炼丹师，而是一个综合要求很高的岗位。要训的了模型；写的了逻辑；优化的了算法性能，时刻把运行速度，准确率，召回率，显存利用率，显存占用，cpu利用率，内存占用，并发路数等等记在心里；还得深刻了解业务，目前的方案合不合适？产品定的指标，给的需求有没有坑？完成这些需求，要选什么样的硬件最划算，可不可以少几个模型，毕竟看数据也很费眼；

4. 多实验，多记录，多对比，勤讨论，勤汇报，勤迭代。这一行多少还是有点玄学的，很多问题没有很强的理论可以支撑，靠经验的地方很多，这个模型效果好，很多情况也不是推导出来的，而是实验出来的，有时候想破脑袋，也没有动手起几个模型效果来的快。平时多跟同事同行讨论讨论，搞不好费了你好几根头发的问题，就被别人解决过呢？

5. 多看行业顶会论文，多追追大牛的博客，思路打开了，落地也就简单很多。

以上是我从事算法行业几年来的一点点体会，不是具体到看了某一篇文章，学了某一个框架，熟悉了某一个语言给自己带来的提升。但就我而言，这些对岗位认知的更新，做事套路的更新对自己的提升有时候要强于某一项具体的技术。今天把这些体会分享出来，希望能符合题主预期。

发布于 2021-02-22 23:13

赞同 564

27 条评论

分享

收藏

喜欢

查看全部 87 个回答