

# 特别加餐 | “银弹”不可靠，最优的模型结构该怎么找？

---

你好，我是王喆。

推荐模型篇的课程到现在已经接近尾声了，我们也已经学习并且实践了六种深度学习推荐模型。最近，我发现很多同学会在留言区提出类似这样的问题：

1. 老师，我的 Wide&Deep 模型在我的数据集上效果不好怎么办？
2. 老师，是不是 DeepFM 模型的效果会比 NeuralCF 好？
3. 老师，DIEN 模型是不是现在效果最好的模型？

其实，我完全理解同学们提问题的心情，就是希望在工作中通过不断地改进模型找到一个最强的模型，来尽快地提升推荐效果，击败当前的模型。我们团队中所有的新人也几乎都有这样的想法。

但是真的存在一个万能的模型结构，能在所有推荐系统上都达成最好的推荐效果吗？这节课，我希望我们能够放缓脚步，务虚一点，好好思考一下到底什么才是最好的模型结构，以及算法工程师正确的工作方法是什么。

## 有解决推荐问题的“银弹”吗？

在软件工程领域的著作《人月神话》中，作者 Brooks 凭借自己在 IBM 管理 2000 人完成大型项目的经验，得出了一个结论：没有任何技术或管理上的进展，能够独立地许诺十年内使软件系统项目生产率、可靠性或简洁性获得数量级上的进步。

Brooks 用“没有银弹”来形容这个结论。在欧洲的古老传说中，银色的子弹是能够一击杀死狼人这种怪物的特效武器。我们也可以把银弹理解成是最好的解决办法。“没有银弹”这个结论让很多期待寻找大型软件开发“捷径”的管理者和工程师们深感失望。但距离人月神话出版已经 45 年的今天，我们找到“银弹”了吗？

很遗憾，不仅在大型软件开发这个领域，“没有银弹”的观念深入人心，而且我要说的是，在推荐系统领域，也同样不存在能够一劳永逸解决问题的“银弹”，或者说根本不存在一种模型结构，它能够一击解决推荐效果问题，做到总是最优。

为什么这么讲呢？我们拿阿里的 DIEN 模型做一个例子。

我们在[21 讲](#)曾经详细介绍过 DIEN 模型，这里再做一个简要的回顾。DIEN 模型的整体结构是一个加入了 GRU 序列模型的深度学习网络。其中，序列模型部分主要负责利用用户的历史行为序列，来预测用户下一步的购买兴趣，模型的其他部分则是 Embedding MLP 的结构，用来把用户的兴趣向量跟其他特征连接后进行预测。

由于阿里巴巴在业界巨大的影响力，DIEN 模型自 2019 年提出以来，就被很多从业者认为是解决推荐问题的“银弹”，并纷纷进行尝试。

但是在应用的过程中，DIEN 并没有体现出“银弹”的效果。在这个时候，大家又都习惯于从自身上找原因，比如说，“是不是 Embedding 层的维度不够”，“是不是应该再增加兴趣演化层的状态数量”，“是不是哪个模型参数没有调好”等等。包括我自己在实践的过程中，也会因为 DIEN 并没有产生预期的推荐效果提升，而去思考是不是因为我们没有完整的复现 DIEN 模型。

说了这么多，我想强调的是，所有提出类似问题的同行都默认了一个前提假设，就是在阿里巴巴的推荐场景下能够提高效果的 DIEN 模型，在其他应用场景下应该同样有效。然而，这个假设真的合理吗？DIEN 模型是推荐系统领域的“银弹”吗？当然不是。接下来，我就带你一起分析一下。

既然 DIEN 的要点是模拟并表达用户兴趣进化的过程，那模型应用的前提必然是应用场景中存在着“兴趣进化”的过程。阿里巴巴的电场景下，因为用户的购买兴趣在不同时间点有变化，所以有着明显的兴趣进化趋势。

比如说，用户在购买了电脑后，就有一定概率会购买电脑周边产品，用户在购买了某些类型的服装后，就会有一定概率选择与其搭配的其他服装。这些都是兴趣进化的直观例子，也是阿里巴巴的电场景下非常典型的情况。

除此之外，我们还发现，在阿里巴巴的应用场景下，用户的兴趣进化路径能够被整个数据流近乎完整的保留。作为中国最大的电商集团，阿里巴巴各产品线组成的产品矩阵几乎能够完整地抓住用户购物过程中兴趣迁移的过程。当然，用户有可能去京东、拼多多等电商平台购物，从而打断阿里巴巴的兴趣进化过程，但从统计意义上讲，大量用户的购物链条还是可以被阿里巴巴的数据体系捕获的。

这样一来，我们就总结出了 DIEN 有效的前提是应用场景满足两个条件，一是应用场景存在“兴趣的进化”。二是用户兴趣的进化过程能够被数据完整捕获。如果二者中有一个条件不成立，DIEN 就很可能不会带来较大的收益。

为什么这么说呢？还是以我自身的实践经历为例，我现在工作的公司 Roku 是北美最大的视频流媒体平台，在使用的过程中，用户既可以选择我们自己的频道和内容，也可以选择观看 Netflix、YouTube，或者其他流媒体频道的内容。但是，一旦用户进入 Netflix 或者其他第三方应用，我们就无法得到应用中的具体数据了。

在这样的场景下，我们仅能获取用户一部分的观看、点击数据，而且这部分的数据仅占用户全部数据的 10% 左右，用户的整个行为过程我们无法完全获取到，那么谈何构建起整个兴趣进化链条呢？

另外一点也很关键，通过分析我们发现，长视频用户的兴趣点相比电商而言其实是非常稳定的。电商用户可以今天买一套衣服，明天买一套化妆品，兴趣的变化过程非常快。但你很难想象长视频用户今天喜欢看科幻电影，明天就喜欢看爱情片，绝大多数用户喜好非常稳定地集中在一个或者几个兴趣点上。这也是序列模型并不能给我们公司提高效果的另一个主要原因。

总的来说，通过 DIEN 的例子我们可以得出，**到底怎样的模型结构是最优的模型结构，跟你的业务特点和数据特点强相关。因此，在模型结构的选择上，没有“银弹”，没有最优，只有最合适。**

## **在工作中避免学生思维**

那么有没有可供参考的方法论来指导模型结构的选择，或者说更广义一点，指导各种模型参数的调优呢？当然是有的，但在谈这个方法论之前，我想先纠正一个工作中非常有害的思维方式，特别是对于刚工作一两年的同学，我们最应该纠正的就是工作中的学生思维。

**“学生思维”最典型的表现就是总是在寻求一个问题的标准答案。**因为在我们的学生时代，所有的题目都是有标准答案的，会就是会，不会

就是不会。

但在工作中就不一样了。举个例子来说，在讲 Embedding 的部分，很多同学问我 Embedding 到底应该取多少维？在实际的工作中，这就是一个典型的没有标准答案的问题。实际工作中，它应有的决策链条应该是下面这样的：

1. 先取一个初始值，比如说 10 维来尝试一下效果；
2. 以 10 维 Embedding 的效果作为 baseline，进行一定程度的参数调优，比如尝试 5 维和 20 维的 Embedding，比较它跟 10 维的效果，确定更好的维度数；
3. 如果项目时间和硬件条件允许，我们还可以尝试 fine tuning（精细调参），直到找到最优的维度设置；
4. 在上线前再次评估 Embedding 线上存储所需存储量的限制，如果线上存储的可用空间有限，我们可以通过适当降低维度数缩小 Embedding 所需的存储空间。

你从这个过程中肯定能够体会到，所谓 Embedding 的维度到底取多少这个问题。根本没有标准答案，最优的答案跟你的数据量，Embedding 模型本身的结构都有关系，甚至还要受到工程环境的制约。

类似的问题还有“局部敏感哈希到底要选择几个桶”，“召回层的 TopN 中，N 到底应该选择多少”，“Wide&Deep 模型中可不可以加入新的用户特征”，“要不要在模型中加入正则化项，Drop out”等等。这些问题都没有标准答案，你只有通过实践中的尝试才能得到最优的设定。

其实这也是我在讲解这门课时一直秉承的原则，我希望把业界的主流方法告诉你，**期望你建立起来的是一套知识体系和方法论，而不是一**

套能让你一劳永逸的模型参数，因为“银弹”并不存在。

## 算法工程师正确的工作方法

我们刚刚否定了“学生思维”，那该怎么建立起一套正确的算法工程师思维呢？下面是我总结出的一套通用的算法工程师的工作流程，虽然不能说我一定掌握了“正确”的方法，但这些工作方法都是从我多年的工作经验中总结出来的，也都得到了验证，你完全可以借助它们来完善自己的工作方法。

1. **问题提出**：清楚领导提出的问题，或者自己发现问题。
2. **数据和业务探索**：在动手解决这个问题之前，我们一定要花时间弄清楚业务的相关逻辑，并且动手用一些脚本程序弄清楚自己可利用数据的数据量、数据特点、提取出一些特征并分析特征和标签之间的相关性。
3. **初始解决方案**：根据探索结果提出初始解决方案。
4. **解决方案调优**：在初始解决方案之上，进行技术选型和参数调优，确定最终的解决方案。
5. **工程落地调整**：针对工程上的限制调整技术方案，尽量做到能简勿繁，能稳定不冒险。
6. **生产环境上线**：进行最终的调整之后，在生产环境上线。
7. **迭代与复盘**：根据生产环境的结果进行迭代优化，并复盘这个过程，继续发现问题，解决问题。

最后我还想补充一点，我一直认为，**做算法工程师，首先要有扎实全面的技术功底，但更重要的其实是自信和务实的精神，不迷信所谓的权威模型，不试图寻找万能的参数，从业务出发，从用户的真实行为出发，才能够构建出最适合你业务场景的推荐模型。**

## 小结

在解决推荐问题上，我认为没有“银弹”的，特别是在模型结构这个点上，我们必须综合考虑自己的业务特点和数据特点，在实践中不断进行参数调优，才能找到最合适自己业务场景的模型。

事实上，不仅仅是推荐问题，对于其他问题来说，我也不建议同学们去追求所谓的“银弹”。换句话说，我们必须尽量避免学生思维，不要总是试图去寻找标准答案，而是应该尽可能多地掌握业界的主流技术手段，丰富自己的“武器库”，建立自己的知识框架。这样，我们才能在面对实际工作中复杂问题的时候，找到最合适的兵器。

除此之外，作为一名算法工程师，我建议你在工作中按照“问题提出”，“数据和业务探索”，“提出初始解决方案”，“解决方案调优”，“工程落地调整”，“生产环境上线”，“迭代与复盘”的顺序，去完成整个的项目周期。这是能帮助你快速建立起正确方法论的有效途径。

总的来说，算法工程师是一份极有挑战的工作，相比研发岗有非常确定的项目目标，算法的优化工作往往需要我们去找到那个可以提升的点，自己去找到那组最合适的参数，并且可以完成生产环境的工程实现。这给了我们极大的灵活性，也给了我们不小的业绩压力。希望这节课可以帮助你纠正一些误区，与你共勉。

## 课后思考

推荐模型的研究可谓层出不穷，很多同学都热衷于追求实现最前沿的技术，最 fancy 的模型，你这样的吗？你认可这种现象吗？

欢迎把你的想法写在留言区，我们一起讨论！