01｜深度学习推荐系统的经典技术架构长啥样？

2020-09-21 王喆

深度学习推荐系统实战 进入课程 >

讲述：王喆
时长 17:02 大小 15.61M

你好，我是王喆。从今天开始，我们正式开始学习“深度学习推荐系统”了。在开始之前，我想先问你一个问题：当你开始学习一个全新领域的时候，你想做的第一件事情是什么？

当然每个人可能都有自己的答案，但对于我自己来说，我最想搞明白的是两个问题。一个是，这个领域到底要解决什么问题？第二个是，这个领域有没有一个非常高角度的思维导图，让我能够了解这个领域有哪些主要的技术，做到心中有数？

针对“深度学习推荐系统”这个领域啊，可能还会有第三个问题，为什么我们要一直强调“深度学习”，深度学习到底给推荐系统带来了什么革命性的影响？相信听完了这一节课，你心中的这三个问题也都能迎刃而解。

推荐系统要解决的根本问题是什么？

在开篇词中我们提到，推荐系统的应用已经渗透到购物、娱乐、学习等生活的方方面面，虽然商品推荐、视频推荐、新闻推荐这些推荐场景可能完全不同，既然它们都被称为“推荐系统”，解决的本质问题一定是相通的，遵循着共通的逻辑框架。

推荐系统要解决的问题用一句话总结就是，**在“信息过载”的情况下，用户如何高效获取感兴趣的信息。**

因此，推荐系统正是在“浩如烟海的互联网信息”和“用户的兴趣点”之间，搭建起的一座桥梁。那这座桥是怎么一步步搭建起来的呢？下面，我们先来看看，推荐系统比较抽象的逻辑架构是什么样的，再一步步搭建起它的技术架构，让你对推荐系统有一个整体上的印象。

推荐系统的逻辑架构

从推荐系统的根本问题出发，我们可以清楚地知道，推荐系统要处理的其实是“**人**”和“**信息**”之间的关系问题。也就是基于“人”和“信息”，构建出一个找寻感兴趣信息的方法。

这里“信息”的定义非常多样，它在不同场景下的具体含义也千差万别。比如说，在商品推荐中指的是“商品信息”，在视频推荐中指的是“视频信息”，在新闻推荐中指的是“新闻信息”，为了方便，我们可以把它们统称为“物品信息”。

而从“人”的角度出发，为了更可靠地推测出“人”的兴趣点，推荐系统希望能利用大量与“人”相关的信息，这类信息包括历史行为、人口属性、关系网络等，它们可以被统称为“用户信息”。

此外，在具体的推荐场景中，用户的最终选择一般会受时间、地点、用户的状态等一系列环境信息的影响，这些环境信息又可以被称为“场景信息”或“上下文信息”。

清楚了这些信息的定义，推荐系统要处理的问题就可以被形式化地定义为：**对于某个用户*U*（User），在特定场景*C*（Context）下，针对海量的“物品”信息构建一个函数，预测用户对特定候选物品*I*（Item）的喜好程度，再根据喜好程度对所有候选物品进行排序，生成推荐列表的问题。**

这样一来，我们就可以抽象出推荐系统的逻辑框架了。虽然这个逻辑框架还比较简单，但我们正是在此基础上，对各模块进行细化和扩展，才产生了推荐系统的整个技术体系。

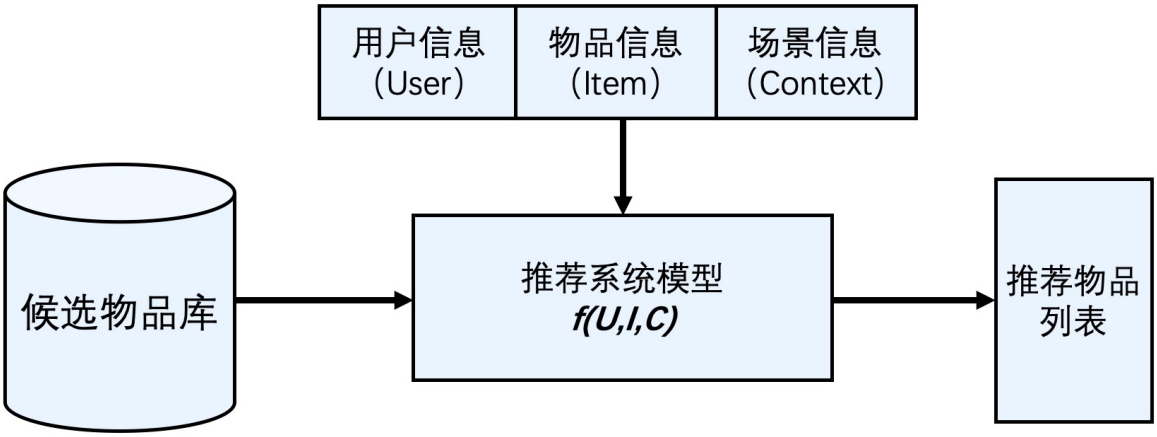

图1 推荐系统逻辑架构

深度学习对推荐系统的革命

有了推荐系统的逻辑架构，我就能回答开头的第三个问题“**深度学习到底给推荐系统带来了什么革命性的影响？**”。

在推荐系统逻辑架构图中，居于中心位置的是一个抽象函数*f(U,I,C)*，它负责“猜测”用户的心，为用户可能感兴趣的物品打分，从而得出最终的推荐物品列表。在推荐系统中，这个函数一般被称为“推荐系统模型”（今后简称“推荐模型”）。

深度学习应用于推荐系统，能够极大地增强推荐模型的拟合能力和表达能力。简单来说，就是让推荐模型“猜的更准”，更能抓住用户的“心”。这么说你可能还没有一个清晰的概念，接下来，我们再从模型结构的角度出发，来比较一下传统机器学习推荐模型和深度学习推荐模型的区别，让你有一个更清晰的认识。

我在下面给出了一张模型结构对比图，它对比了传统的矩阵分解模型和深度学习矩阵分解模型的区别。我们先忽略细节不谈，你第一眼看上去有什么感觉？是不是觉得深度学习模型变得更复杂了，一层又一层，层数增加了很多。

你的感觉一点儿都没错，其实正是**因为深度学习复杂的模型结构，让深度学习模型具备了理论上拟合任何函数的能力**。如果说 *f(U,I,C)*这个推荐函数具有一个最优的表达形式，那传统的机器学习模型只能够拟合出*f(U,I,C)* 这个推荐函数的近似形式，而深度学习模型则可以最大程度地接近这个最优形式。

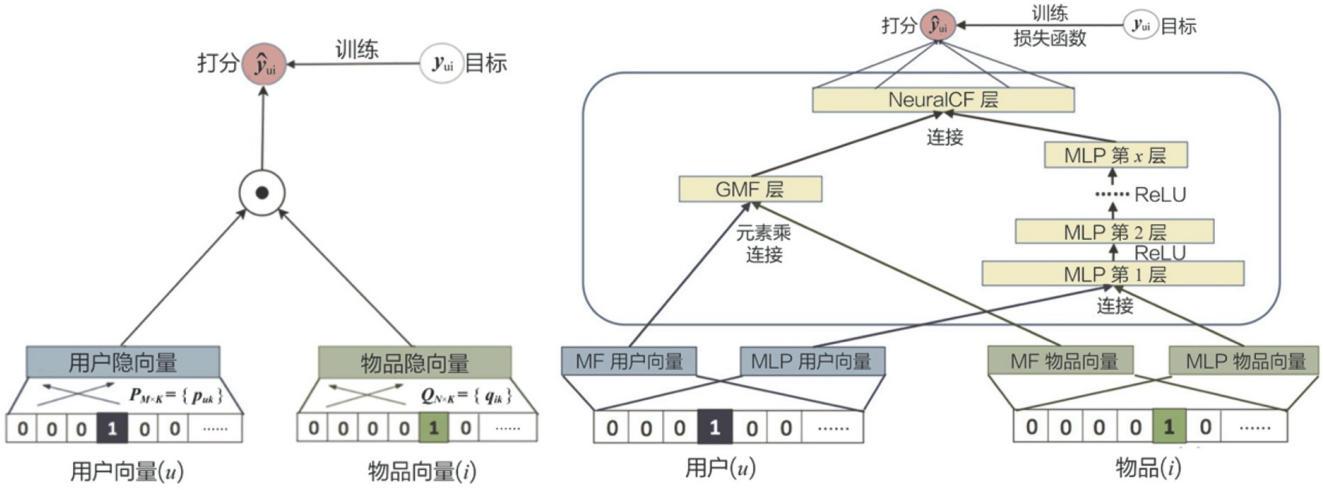

图2 传统的矩阵分解模型和深度学习矩阵分解模型的对比图
来源：《Neural collaborative filtering》

除此之外，深度学习模型非常灵活的模型结构还让它具备了一个无法替代的优势，就是我们可以**让深度学习模型的神经网络模拟很多用户兴趣的变迁过程，甚至用户做出决定的思考过程**。比如阿里巴巴的深度学习模型——深度兴趣进化网络（如图 3），它利用了三层序列模型的结构，模拟了用户在购买商品时兴趣进化的过程，如此强大的数据拟合能力和对用户行为的理解能力，是传统机器学习模型不具备的。

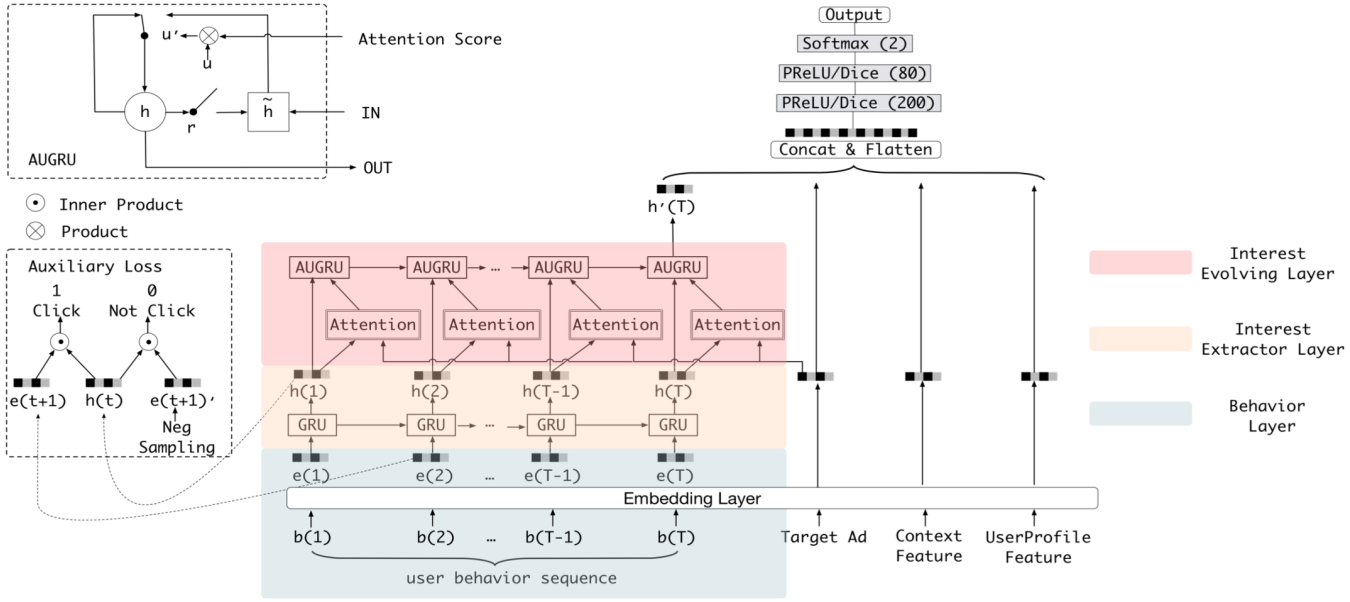

图3 阿里巴巴的深度兴趣进化网络

来源：《Deep Interest Evolution Network for Click-Through Rate Prediction》

但是，深度学习对推荐系统的革命影响还远不止这些。近几年，由于深度学习模型的结构复杂度大大提高，使通过训练使模型收敛所需的数据量大大增加，这也反向推动了推荐系统大数据平台的发展，让推荐系统相关的大数据存储、处理、更新模块也一同迈入了“深度学习时代”。

讲了这么多深度学习对推荐系统的影响，我们似乎还没看到一个完整的深度学习推荐系统架构。别着急，下面，我们就来讲一讲，经典的深度学习推荐系统的技术架构是什么样的。

深度学习推荐系统的技术架构

讲之前啊，我还要说明一点，深度学习推荐系统的架构与经典的推荐系统架构其实是一脉相承的，它对经典推荐系统架构中某些特定模块进行了改进，使之能够支持深度学习的应用。所以，我会先讲经典的推荐系统架构，再讲深度学习对它们的改进。

在实际的推荐系统中，工程师需要着重解决的问题有两类。

一类问题与数据和信息相关，即“用户信息”“物品信息”“场景信息”分别是什么？如何存储、更新和处理数据？

另一类问题与推荐系统算法和模型相关，即推荐系统模型如何训练、预测，以及如何达成更好的推荐效果？

一个工业级推荐系统的技术架构其实也是按照这两部分展开的，其中“数据和信息”部分逐渐发展为推荐系统中融合了数据离线批处理、实时流处理的数据流框架；“算法和模型”部分则进一步细化为推荐系统中，集训练（Training）、评估（Evaluation）、部署（Deployment）、线上推断（Online Inference）为一体的模型框架。基于此，我们就能总结出推荐系统的技术架构图。

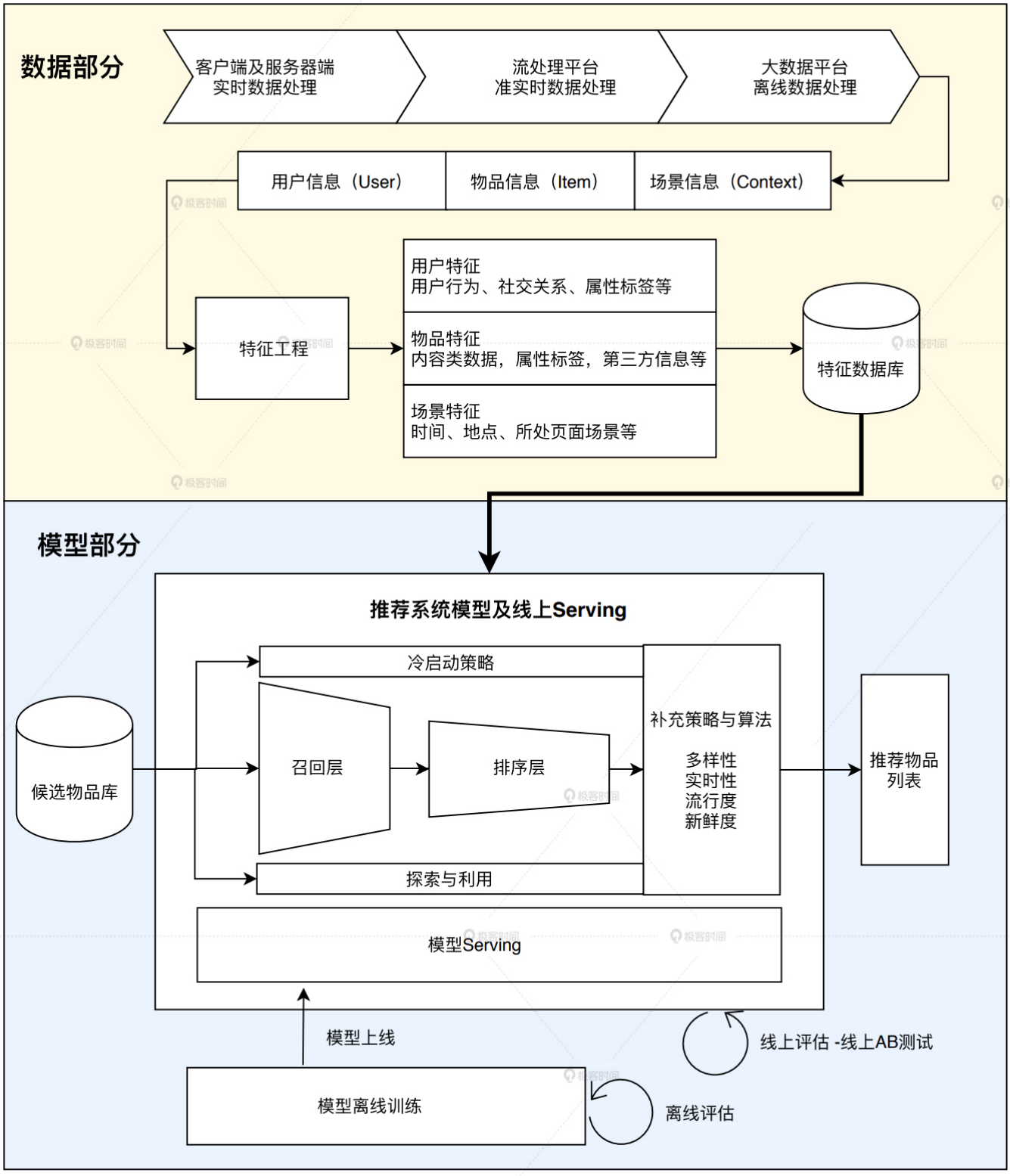

图4 推荐系统技术架构示意图

在图 4 中，我把推荐系统的技术架构分成了“数据部分”和“模型部分”。那它们的工作内容和作用分别是什么呢？深度学习对于这两部分的影响又有哪些呢？下面，我来一一讲解。

第一部分：推荐系统的数据部分

推荐系统的“数据部分”主要负责的是“用户”“物品”“场景”信息的收集与处理。根据处理数据量和处理实时性的不同，我们会用到三种不同的数据处理方式，按照实时性的强弱排序的话，它们依次是**客户端与服务器端实时数据处理、流处理平台准实时数据处理、大数据平台离线数据处理**。

在实时性由强到弱递减的同时，三种平台的海量数据处理能力则由弱到强。因此，一个成熟推荐系统的数据流系统会将三者取长补短，配合使用。我们也会在今后的课程中讲到具体的例子，比如使用 Spark 进行离线数据处理，使用 Flink 进行准实时数据处理等等。

大数据计算平台通过对推荐系统日志，物品和用户的元数据等信息的处理，获得了推荐模型的训练数据、特征数据、统计数据等。那这些数据都有什么用呢？具体说来，大数据平台加工后的数据出口主要有 3 个：

1. 生成推荐系统模型所需的样本数据，用于算法模型的训练和评估。
2. 生成推荐系统模型服务（Model Serving）所需的“用户特征”，“物品特征”和一部分“场景特征”，用于推荐系统的线上推断。
3. 生成系统监控、商业智能（Business Intelligence，BI）系统所需的统计型数据。

可以说，推荐系统的数据部分是整个推荐系统的“水源”，我们只有保证“水源”的持续、纯净，才能不断地“滋养”推荐系统，使其高效地运转并准确地输出。在深度学习时代，深度学习模型对于“水源”的要求更高了，**首先是水量要大**，只有这样才能保证我们训练出的深度学习模型能够尽快收敛；**其次是“水流”要快**，让数据能够尽快地流到模型更新训练的模块，这样才能够让模型实时的抓住用户兴趣变化的趋势，这就推动了大数据引擎 Spark，以及流计算平台 Flink 的发展和应用。

第二部分：推荐系统的模型部分

推荐系统的“模型部分”是推荐系统的主体。模型的结构一般由“召回层”、“排序层”以及“补充策略与算法层”组成。

其中，“召回层”一般由高效的召回规则、算法或简单的模型组成，这让推荐系统能快速从海量的候选集中召回用户可能感兴趣的物品。“排序层”则是利用排序模型对初筛的候选集进行精排序。而“补充策略与算法层”，也被称为“再排序层”，是在返回给用户推荐列表之前，为兼顾结果的“多样性”“流行度”“新鲜度”等指标，结合一些补充的策略和算法对推荐列表进行一定的调整，最终形成用户可见的推荐列表。

从推荐系统模型接收到所有候选物品集，到最后产生推荐列表，这一过程一般叫做“**模型服务过程**”。为了生成模型服务过程所需的模型参数，我们需要通过模型训练（Model Training）确定模型结构、结构中不同参数权重的具体数值，以及模型相关算法和策略中的参数取值。

模型的训练方法根据环境的不同，可以分为“离线训练”和“在线更新”两部分。其中，离线训练的特点是可以利用全量样本和特征，使模型逼近全局最优点，而在线更新则可以准实时地“消化”新的数据样本，更快地反应新的数据变化趋势，满足模型实时性的需求。

除此之外，为了评估推荐系统模型的效果，以及模型的迭代优化，推荐系统的模型部分还包括“离线评估”和“线上 A/B 测试”等多种评估模块，用来得出线下和线上评估指标，指导下一步的模型迭代优化。

我们刚才说过，深度学习对于推荐系统的革命集中在模型部分，那具体都有什么呢？我把最典型的深度学习应用总结成了 3 点：

1. 深度学习中 Embedding 技术在召回层的应用。作为深度学习中非常核心的 Embedding 技术，将它应用在推荐系统的召回层中，做相关物品的快速召回，已经是业界非常主流的解决方案了。
2. 不同结构的深度学习模型在排序层的应用。排序层（也称精排层）是影响推荐效果的重中之重，也是深度学习模型大展拳脚的领域。深度学习模型的灵活性高，表达能力强的特点，这让它非常适合于大数据量下的精确排序。深度学习排序模型毫无疑问是业界和学界都在不断加大投入，快速迭代的部分。
3. 增强学习在模型更新、工程模型一体化方向上的应用。增强学习可以说是与深度学习密切相关的另一机器学习领域，它在推荐系统中的应用，让推荐系统可以在实时性层面更上一层楼。

小结

这节课，我带你熟悉了深度学习推荐系统的技术架构，虽然涉及的内容非常多，但如果没有记住的话，你也完全不用慌张，只需要在心中留下这个框架的印象就可以了。你完全可以把这节课的内容当作整个课程的技术索引，让它成为属于你自己的一张知识图谱。

形象点来说，你可以把这节课程的内容想象成是一颗知识树，它有根，有干、有枝、有叶，还有花。

其中，推荐系统的根就是推荐系统要解决的根本性问题：在“信息过载”情况下，用户怎么高效获取感兴趣的信息。

而推荐系统的干就是推荐系统的逻辑架构：对于某个用户U（User），在特定场景C（Context）下，针对海量的“物品”构建一个函数，预测用户对特定候选物品I（Item）的喜好程度的过程。

枝和叶就是推荐系统的各个技术模块，以及各模块的技术选型。技术模块撑起了推荐系统的技术架构，技术选型又让我们可以在技术架构上实现各种细节，开枝散叶。

最后，深度学习在推荐系统的应用无疑是当前推荐系统技术架构上的明珠，它就像是这颗大树上开出的花，是最精彩的点睛之笔。深度学习的模型结构复杂，数据拟合能力和表达能力更强，能够让推荐模型更好的模拟用户的兴趣变迁过程，甚至是做决定的过程。而深度学习的发展，也推动着推荐系统数据流部分的革命，让它能够更快、更强地处理推荐系统相关的数据。

好了，这节课我们就讲到这里，希望你能牢牢记住深度学习推荐系统的架构，播撒下这一粒种子，然后跟随我后面的课程让它长大成一颗属于你自己的参天大树。

图5 推荐系统的技术架构像树一样生发而出

课后思考

下面是 Netflix 的推荐系统的经典架构图，你能结合本节课讲的推荐系统技术架构，说出 Netflix 架构图中哪些是数据部分，哪些是模型部分吗？

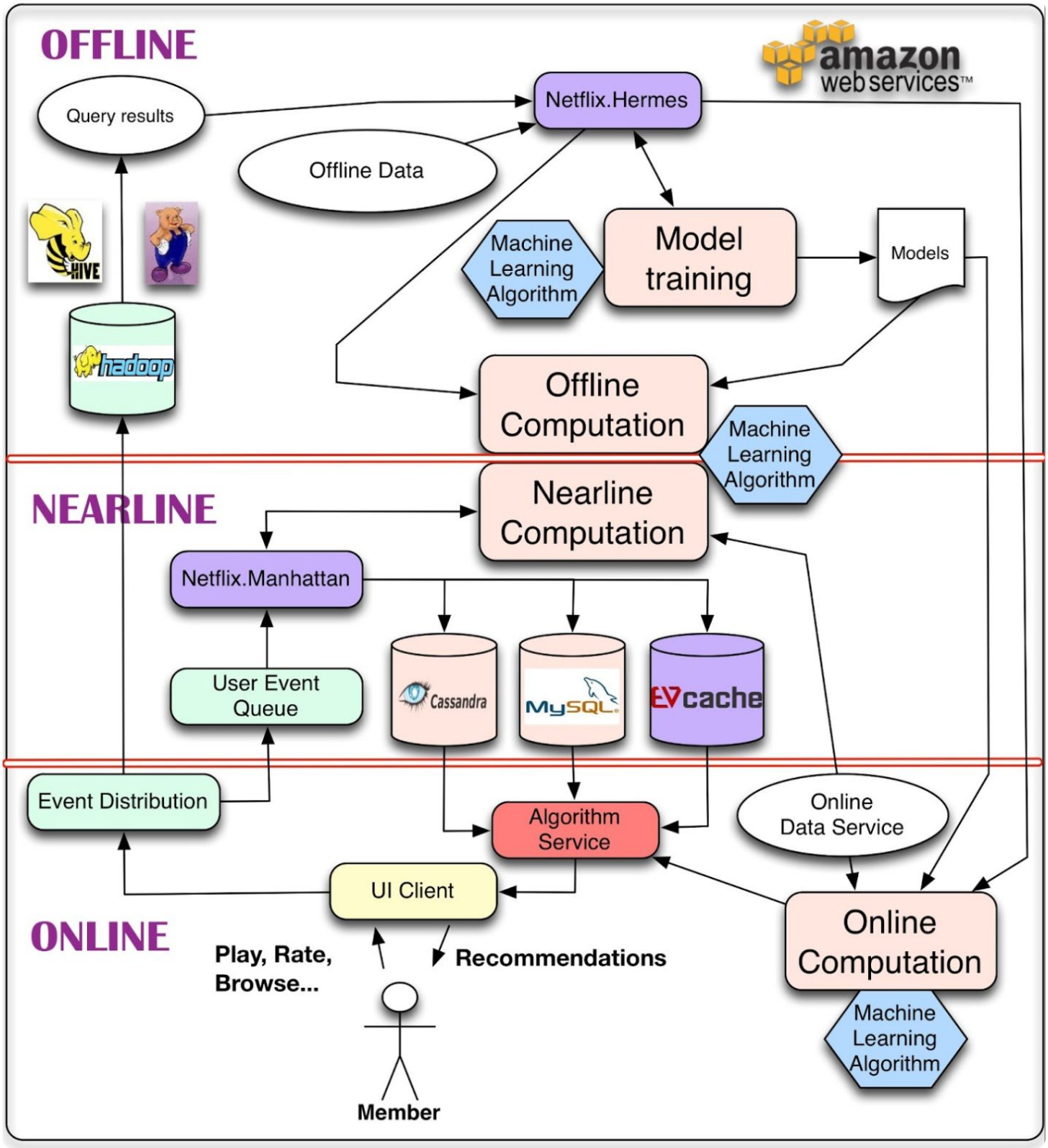

图6 Netflix架构图示意图

这样的深度学习推荐系统和你想的一样吗？如果今天的课程对你有帮助，也欢迎你把它转发出去！我们下节课见！

13 人觉得很赞 | 提建议

上一篇 开篇词｜从0开始搭建一个深度学习推荐系统

下一篇 02｜Sparrow RecSys：我们要实现什么样的推荐系统?

精选留言 (17)

写留言

朱月俊
2020-09-22

数据部分：event distribution, hadoop, query results, netflex.hermes, user event queue, netflix.manhattan.
模型部分：model training, models, online computation, online service, algorithm service.

展开

作者回复: 赞，完全正确。另外nearline computation也属于数据部分，正中央的evcache等几个数据库可以看作数据部分和模型部分的接口。

8

杜军

2020-09-27

请教大神，我们注意到 Flink 最近更新比较频繁，号称可以做到流批一体分析，甚至 ETL 领域好像也可以用起来，是不是可以在系统架构设计的时候直接用 Flink 取代 Spark，ETL 和实时部分统一到一个架构上来是否可行？谢谢

展开

作者回复: 这是个很好的问题。其实也是大数据工程师们一直追求的批流一体的Kappa架构。

但实践中遇到的困难不少。一是一些历史遗留问题，比如当前很多公司的数据体系大部分是建立在spark基础上的，那直接用flink去替代肯定有风险，所以很多公司还沿用着混合的lambda架构。

另外是Spark和Flink发展的问题，Flink在进化的同时Spark也在发展，比如Structured Streaming的提出就是为了跟Flink竞争，而且Spark本身的社区成熟程度和这么多年的积累还是超过目前的Flink的。所以也难说Flink会完全替代Spark。

但毫无疑问，批流一体是未来的方向，大家也都在往这个方向努力。但我个人觉得Spark和Flink会长期共存，共同发展。

5

Four y

2020-09-22

老师，请问关于大数据数据出口的那一部分，请问实时的用户推荐请求也是会先经过大数据处理，生成可供线上推理的数据吗？就是针对文中大数据出口的第二点。

展开

作者回复: 这是个好问题，希望大家多提这样的思考。
在推荐服务器做线上推断时，实时用户请求里面包含的特征一般是直接在服务器内部提取出来的，所以肯定不需要再在数据流中走一遍。

但是线上请求数据最终还是会落盘，生成日志数据，这个过程中，一些流处理，和批处理的平台会对这些数据做进一步处理，生成今后可供使用的特征以及训练用样本。

2 4

高璇璇

2020-09-23

增强学习是指reinforcement learning吗？国内一般叫强化学习

作者回复: 是的，两种可能都有叫，大家能理解就好。

2

李@君
2020-09-23

老师，spark和flink都兼具批处理和流处理的能力，技术选型时为什么不使用其中一个呢。文中所提到的实时处理，又实用的是什么技术呢。模型在线更新时，是使用新产生的的数据再对模型进行训练吗，这样的话会不会太耗时，影响功能。

展开

作者回复: 问题比较多，可能需要在后续课程中深入讨论。不是一两句话能讲清楚。

简单来讲数据流部分是每个公司最复杂的部分，其中有历史原因，也有各平台配合的原因。比如Spark目前来讲相比Flink还是更适用于批量大数据处理，而Flink基于流的思想提出，天然更适合流处理。具体的选型，各平台之间的配合，我们在后续对应章节再详细讨论。

2

Tango
2020-10-22

看其他留言都太牛了，加油学习吧。

展开

作者回复: 加油

1

张弛 Conor
2020-09-28

老师，想向您请教一下召回层和排序层除了结果上的“粗”和“精”是否还有其他的区别？另外就是二者在深度学习技术加持下能否实现端到端的排序呢？

展开

作者回复: 我们在之后的召回层介绍中还会深入讲解召回层，所以期待之后的交流。

关于端到端排序是一个非常好的问题。其实我们理想状态下最好的结果就是不分召回层和排序层，实现端到端排序。但是工程上很难做到，因为排序层往往是复杂模型，大规模候选集情况下

延迟较大。

但是也已经有不少业界团队在探索端到端排序的可能。我觉得是一个很好的值得改进的方向。

1

fsc2016
2020-09-22

老师，这套课程侧重点和您的书《深度学习推荐系统》区别是什么了，更偏实战嘛

作者回复: 是的，专栏的名字是《深度学习推荐系统实战》，所以会更注重理论联系实际，用一套代码SparrowRecSys把所有重要的知识点实现一遍，最后串联成一套成型的推荐系统。

1

wYang
2020-10-26

请问王老师一个概念问题，落盘是什么意思？为什么要用落盘这个词？

作者回复: 落盘一般是指写入磁盘或者一些永久性存储的存储系统。

Yuwei Quan
2020-10-15

我想问nearline computation这一步是在干嘛啊，是把训练好的model 存成artifact准备和online model相结合做deployment吗。我不太懂数据部分和模型部分的接口是什么意思。最后的event distribution存起来有什么好处啊，会复盘吗？然后他们的Netflix Hermes和Manhattan是一个部组的整个代称还是一个数据处理的代称。谢谢

展开

作者回复: nearline computation主要负责准实时特征更新。
数据部分和模型部分的接口 可以关注后续课程的特征和模型服务部分。
event distribution 可以用于异常监控预警和一些特征的生成。
Hermes和Manhattan是Netflix推荐系统的两个模块，不用过分纠结细节。

Geek_07
2020-10-10

老师，我想请教一下，候选物品库大概是怎么做存储管理的呢？像今日头条这种平台可能会有不同类型的物品（比如视频、图文等等），他们的结构化的内容属性可能都是不同的，做召回的时候，召回层又是怎么跟物品库做交互的？

展开

作者回复: 召回层的时候还会做介绍。基本都会采用多路召回进行融合。


夜雨声烦
2020-10-09

推荐系统要解决的三个问题：
1，推荐系统要解决的问题是什么？
在“信息过载”的情况下，用户如何高效获取感兴趣的信息。
2，有没有一个非常高角度的思维导图，让我能够了解这个领域有哪些主要的技术，做到心中有数？...

展开

作者回复: 作业部分回答的非常好。


西北小英雄
2020-10-02

老师您好，我去看了您的知乎专栏对Netflix推荐系统架构图的讲解，对下面部分有点疑问。（可能我是推荐系统新手，没有工程经验对架构不太懂，如果的问题不是很营养，望谅解）

比如从online 到nearline和offline通过用户消息队列（User Event Queue，现在基本都...

展开

作者回复: 1、User Event Queue就拿用kafka举例，跟offline的关系确实不强，唯一的关系就是数据经kafka最终会落盘到离线的存储系统。所以这一点你的理解没错
2、nearline的理解也没错，一般来说nearline处理好的数据、特征会存储到evcache，redis等内存数据库供online service使用。


太子长琴
2020-09-25

老师好，请问下啊，
特征库会存储模型计算后的特征（比如embeddibg）吗，还是只存储基本特征？
实时新数据（比如新用户，新物品）一般是多久或者有什么指标确定需要更新到离线模型呢？
谢谢

展开

作者回复: 一般来说embeeding也完全可以存储到特征数据库中，最终的决定还是应该看各公司具体的技术架构和业务需求。如果embedding数据量比较大，也有单独存储的。

实时新数据一般来说，特征的更新是越快越好，因为特征的改变也直接影响到最终的排序结果。

模型的训练没有统一的指标，理论上来说也是越快越好，但要考虑算力，数据处理延迟等工程限制。大多公司以小时级别更新。

李@君
2020-09-23

Netflix的离线数据处理部分，还在使用hive和pig技术吗。

作者回复: 这张图是netflix 2015年发布的，所以有些技术肯定会更新。在这里仅作为大家练习使用。

军舰
2020-09-22

老师，候选物品库是不是可以这样理解：我现在在看柯南，那么候选物品可能是动画片、侦探题材等？

作者回复: 候选物品库是所有可被推荐的物品集合。比如一个动漫视频网站，他的候选物品可以是柯南，海贼王，七龙珠等等，但动画片，侦探题材这些属于物品上的特征，并不是候选物品。

少刷票圈多读书

2020-09-22

王喆老师对应的知乎专栏文章：https://zhuanlan.zhihu.com/p/114590897

作者回复: 嗯，基本算是课后问题的详细解答了。