推荐系统Embedding向量召回在即刻的工程实践

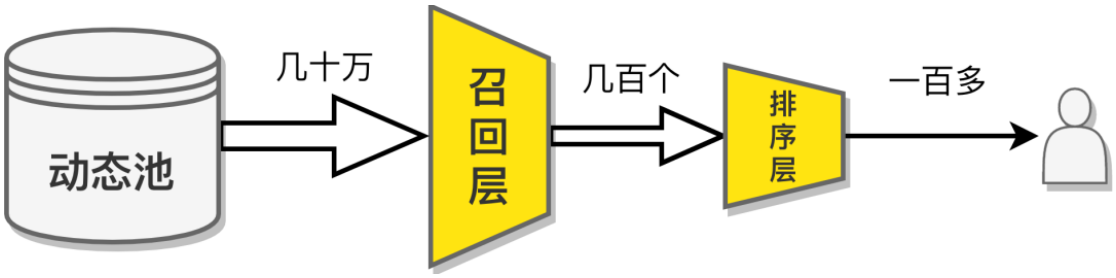
原创 欧承祖 即刻技术团队 7月15日

收录于话题

#推荐系统 135 #召回策略 1 #向量 10

即刻首页的动态广场是一个发现好玩圈子和有趣即友的地方。如何让每一个用户都能在上面看到自己感兴趣的内容，是即刻推荐团队一直以来努力的目标。

去年我们分享了一篇技术文章，介绍如何在线上服务中使用Spark MLlib，解决了排序模型的线上serving问题。过去这一年，即刻推荐系统完成了从Spark+XGBoost到TF+DNN的技术升级。在排序模型深度化之后，我们开始探索深度学习在召回层上的应用。



推荐系统召回层的目标，就是根据用户画像以及过往的消费历史，从百万级的推荐池里粗筛出当前用户最可能感兴趣的几百条内容，并将结果交由排序模型进一步排序。与排序不同的是，召回层面对的候选集量级通常是巨大的。这就导致在召回时，没法将候选集中的每条内容**逐个**与当前用户进行比对计算，挑出用户最可能喜欢的。因此，我们一般会根据不同业务的需求，将候选池中的内容按照某种方式做索引，从而根据某方面的特征挑出用户可能喜欢的一批内容。不同的索引方式，对应着不同的召回策略。多路召回策略并行地工作，总体来看，就是从不同角度去搜集用户可能感兴趣的内容，完成一次召回。不同的业务场景有独特的召回策略，不同的内容类型也有各异的索引方式。

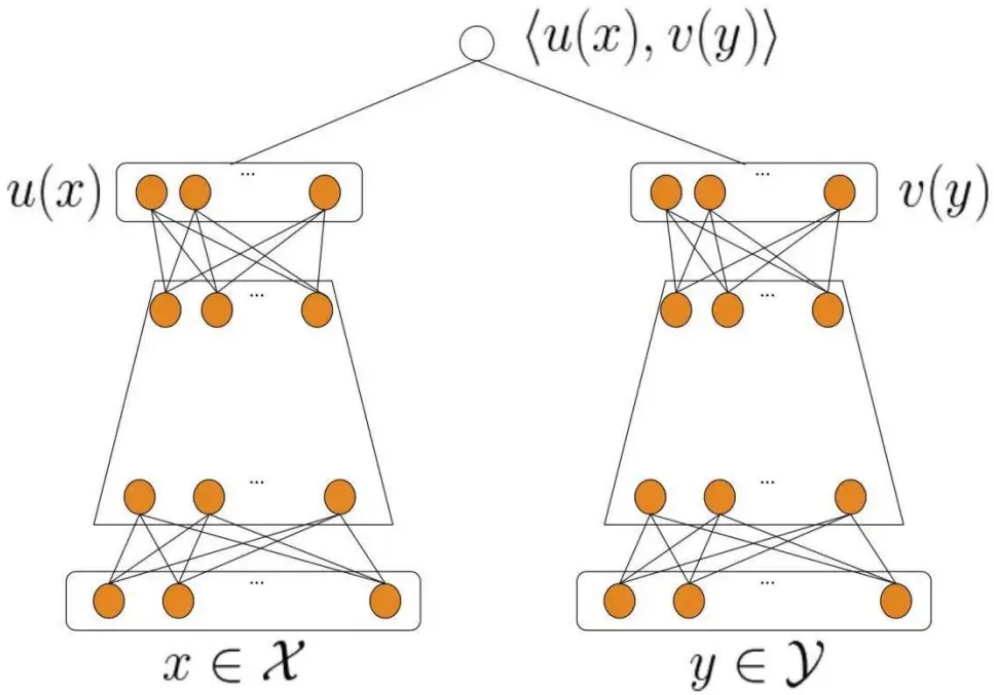
在这些召回策略中，有一类特殊的，被称为基于向量embedding的召回。作为一种既考虑了精确度又兼顾了覆盖面的召回策略，它被广泛运用于各类业务场景中。向量召回的索引方式，就是基于向量来完成的。在训练好一个embedding模型，得到关于物品和用户的向量后，就可以将物品的向量作为索引的key，用户的向量作为查询的query，通过一些模糊最近邻(ANN)的算法快速从海量候选池中找到一批最匹配的物品。由于embedding的普适性，几乎任何业务场景和内容类型都可以尝试使用向量召回。

在即刻，我们为动态广场推荐流的场景制定了一些召回策略，在一定程度上满足了帮助即友发现有趣内容的需求。但随着社区兴趣圈的丰富，用户的喜好也变得更加多元。为了让推荐系统能好满足个性化的需求，我们决定开始尝试在召回策略中加入基于embedding向量的召回。

模型结构

业界成熟的embedding模型有很多：从最原始的Matrix Factorization，到受word2vec启发的item2vec、node2vec；从基于用户点击序列的YouTubeDNN，到基于图的GCN和GraphSAGE。不同的模型复杂度不同，也有不同的应用场景。

考虑到即刻动态流推荐对于实时性的要求，我们受到DSSM的启发，选择了一个简单的DNN双塔模型，并通过有监督的方式训练，优化目标是点击率。



双塔模型的两个塔分别接收用户侧的特征和内容侧的特征，输出用户向量和内容向量。对于单个塔内部的结构，我们选用了最简单的多层全连接的结构：第一层是特征embedding层，将原始输入的多个特征通过embedding得到向量表示，并将所有特征向量concat；最后一层是输出的embedding层，得到用户侧和内容侧的向量表示。得到双侧的向量表示后，直接用一个简单的距离函数，计算两个向量的距离，得到最后的输出为[0,1]之间的一个标量，通过交叉熵的方式与用户点击的反馈做loss计算。考虑到线上召回的实时性要求，最后输出的embedding维度设为64。

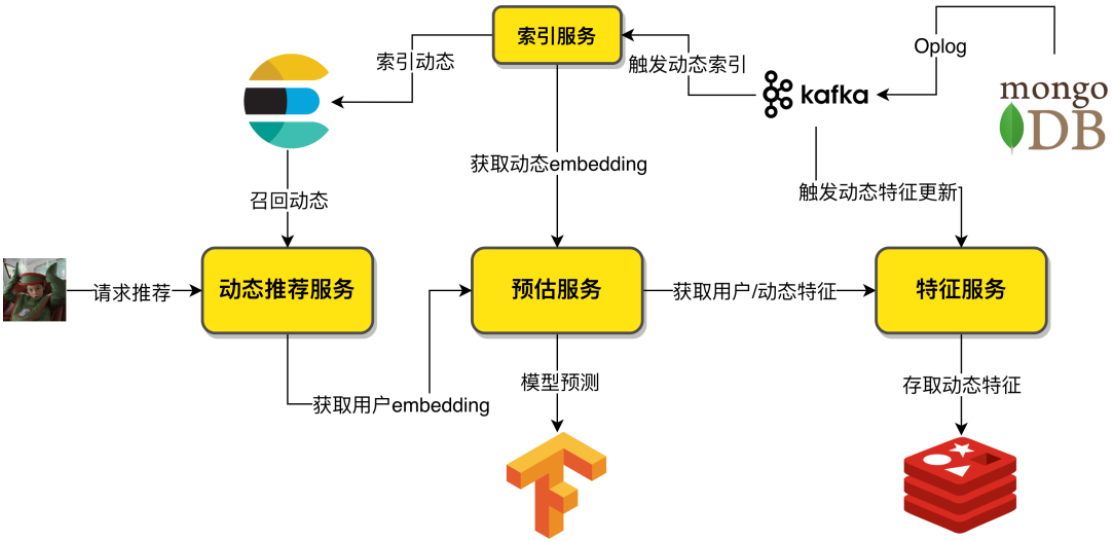
模型部署

在得到一个训练好的模型后，我们需要将模型部署成一个在线服务，供线上的推荐系统使用。由于在实际使用中，内容侧的向量和用户侧的向量是分开计算的。因此在部署服务时，需要将双塔模型的两个塔拆成两个小模型部署：内容模型只接收内容的特征输入，输出一个内容侧向量；用户模型则接收用户侧的特征，输出用户向量。

使用TensorFlow Serving，可以方便地将模型文件直接部署为在线服务。

索引召回架构

将模型部署到线上后，需要在现有的推荐系统架构中整合基于embedding的召回策略。



我们目前的索引、召回流程是基于Elasticsearch完成的，将新增的向量召回通道集成进ES是最省时又方便的做法了。

整个索引召回流程分成两个部分。在动态索引阶段，通过消费MongoDB的operation log，可以实现动态的近实时索引和特征更新。索引服务在接收到索引请求时，通过特征服务获取或计算动态的近实时特征，然后预估服务调用TF-serving计算得到动态embedding向量，最后连同其他字段一起索引进ES。在推荐召回阶段，通过调用特征服务计算用户的实时特征，然后调用TF-serving计算得到用户embedding向量，再去ES中检索用户最可能喜欢的一批动态，完成一次embedding召回。

ES在7.X版本中原生支持dense vector索引和查询，但由于我们目前的ES版本是6.7，还没有原生的向量索引功能，因此使用了阿里云提供的向量索引插件来完成向量的索引和模糊查询。

向量检索的P95耗时基本满足推荐服务的延迟要求。

效果与迭代方向

第一版基于embedding向量的召回策略上线后，即刻动态广场的整体互动率提升了**33.75%**，达到了单次上线最大效果提升。基于embedding的召回策略不仅是所有召回策略里互动率最高的，而且也是分发量最大的。

初版模型只是我们将深度学习应用在召回层的一次小小尝试，这版模型还存在很多问题，无论是模型结构还是推荐架构都有很多优化空间。在架构上，一个亟需解决的是，如何自动地完成embedding模型更新带来的向量版本同步问题。我们知道，embedding模型的一个根本性问题，就是每个维度的语义是未知且易变的，每次训练得到的向量空间都不同，也就是说不同版本的向量之间无法比较。这就为模型迭代优化造成了很多阻碍，怎么持续地完成模型更新集成，是一个很大难题。在模型结构上，双塔模型有其优势，但也有很多不足的地方，比如没有考虑行为序列等。后续我们计划在模型结构和训练方式上尝试更多可能。

在即刻，我们持续研究复杂的前沿机器学习算法，并将其应用于真实的推荐系统中；我们还关注如何建立一套灵活、敏捷的部署流程，方便快速迭代模型。这一切，都是为了帮助即友们发现更多好玩的圈子、认识更多有趣的朋友。即刻推荐团队欢迎优秀的工程师加入我们，一起建设即刻镇。

参考链接：

- https://zhuanlan.zhihu.com/p/143763320

- https://zhuanlan.zhihu.com/p/97821040

- https://zhuanlan.zhihu.com/p/73853438

- https://posenhuang.github.io/papers/cikm2013_DSSM_fullversion.pdf

- https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf

作者：

欧承祖，即刻算法工程师

阅读原文