21 | YouTubeDNN：召回算法的后起之秀（上）

2023-06-02 黄鸿波 来自北京

《手把手带你搭建推荐系统》

你好，我是黄鸿波。

在前面的课程中，我们讲解了几种不同的召回算法，在这节课中我们会继续前面的课程，学习一个新的召回算法——YouTubeDNN 模型。YouTubeDNN 模型因为内容比较多，我把它分成了上下两篇，我们这节课先聚焦 YouTubeDNN 模型的概念和召回原理，在下节课实现一个基于 YouTubeDNN 的召回。

YouTubeDNN 模型的概念及结构

上一节课中说到比较经典的 U2I 模型有两种，一种是 DSSM，另一种就是这节课要讲解的基于 YouTubeDNN 的召回模型。

YoutubeDNN 是 Youtube 用于做视频推荐的落地模型，可以说是最近几年来推荐系统中的经典模型。其大体思路就是召回阶段使用多个简单的模型来进行筛选，这样可以大量地筛除相

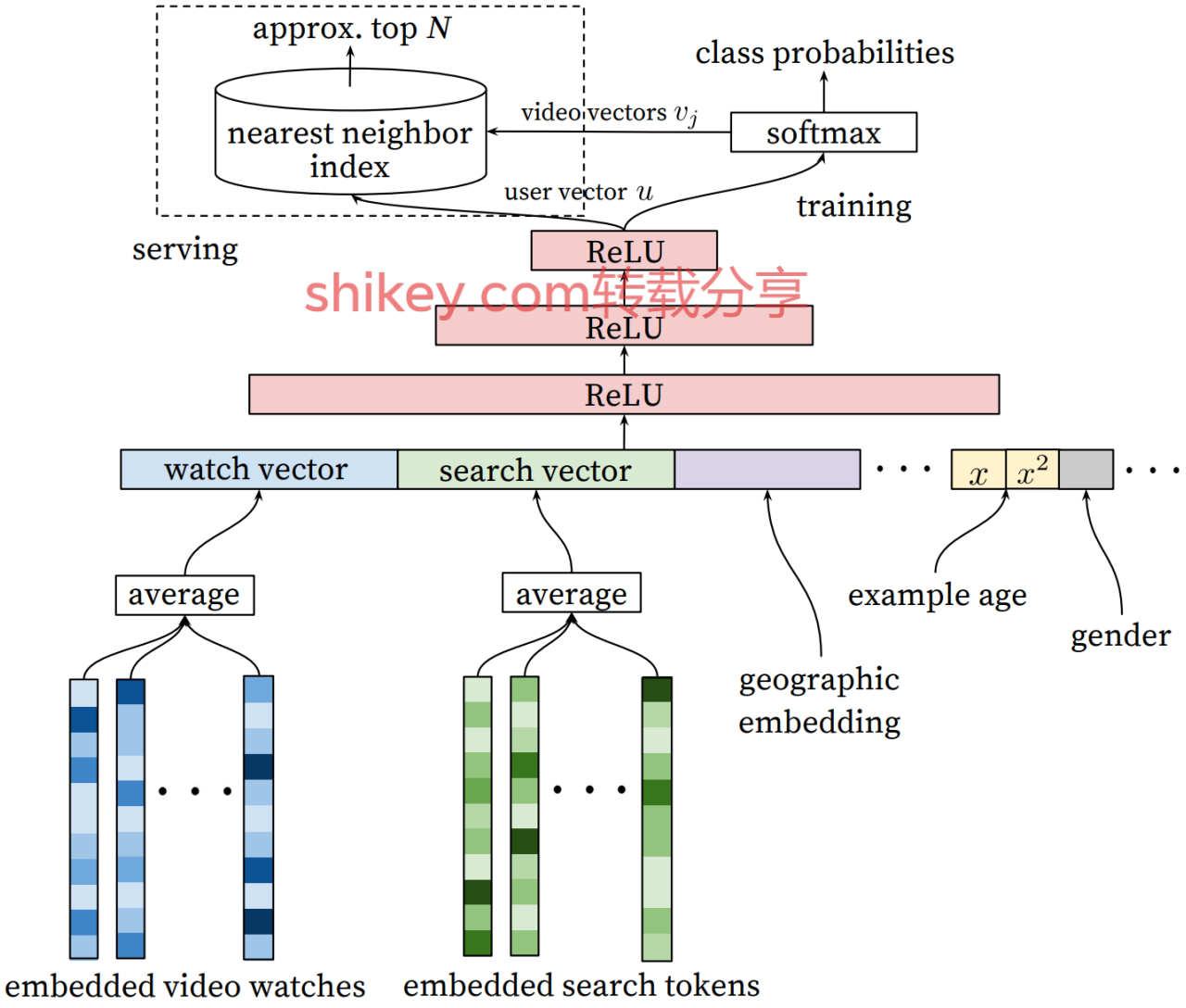
关度较低的内容，而排序阶段则是使用相对复杂的模型来获得精准的推荐结果。因此，YouTubeDNN 实际上包含了两个部分：召回和排序。这里我们主要讲解 YouTubeDNN 模型中的召回部分。

YouTubeDNN 模型的召回主要是完成候选视频的快速筛选（在论文中被称为 Candidate Generation Model），也就是候选集的生成模型。在这一部分中，模型要做的就是将整个 YouTube 数据库中的视频数量由百万级别降到数百级别。

如果用协同过滤算法来处理这百万级别的数据量，显然处理不过来。因为协同过滤的本质是计算两两内容之间的关系矩阵，如果内容越多，矩阵就会越大。我们在计算协同过滤时会把整个矩阵放入内存当中，当矩阵变得越来越大时，会导致 OOM（Out Of Memory）现象，说白了就是内存会爆掉，导致最后无法产生想要的结果。即使有一个无限大内存的机器，对于这种上百万的矩阵计算时间也会非常长，也不是我们想要的。

而 YouTubeDNN 利用了 Embedding 向量加上对负样本的特殊采样处理，巧妙地解决了这一问题。

接下来我们详细看下具体的解决思路，首先是 YouTubeDNN 召回部分的模型结构。



这个结构我们自底向上来看，整个模型由最下面的模型输入层开始，经过了三层 ReLU 神经网络之后得到用户的特征向量。然后经过 Softmax 层进行预测，得到每个视频的观看概率，从而得到了视频的召回层排序。接着，我们来详细地说一下这几层所做的事情。

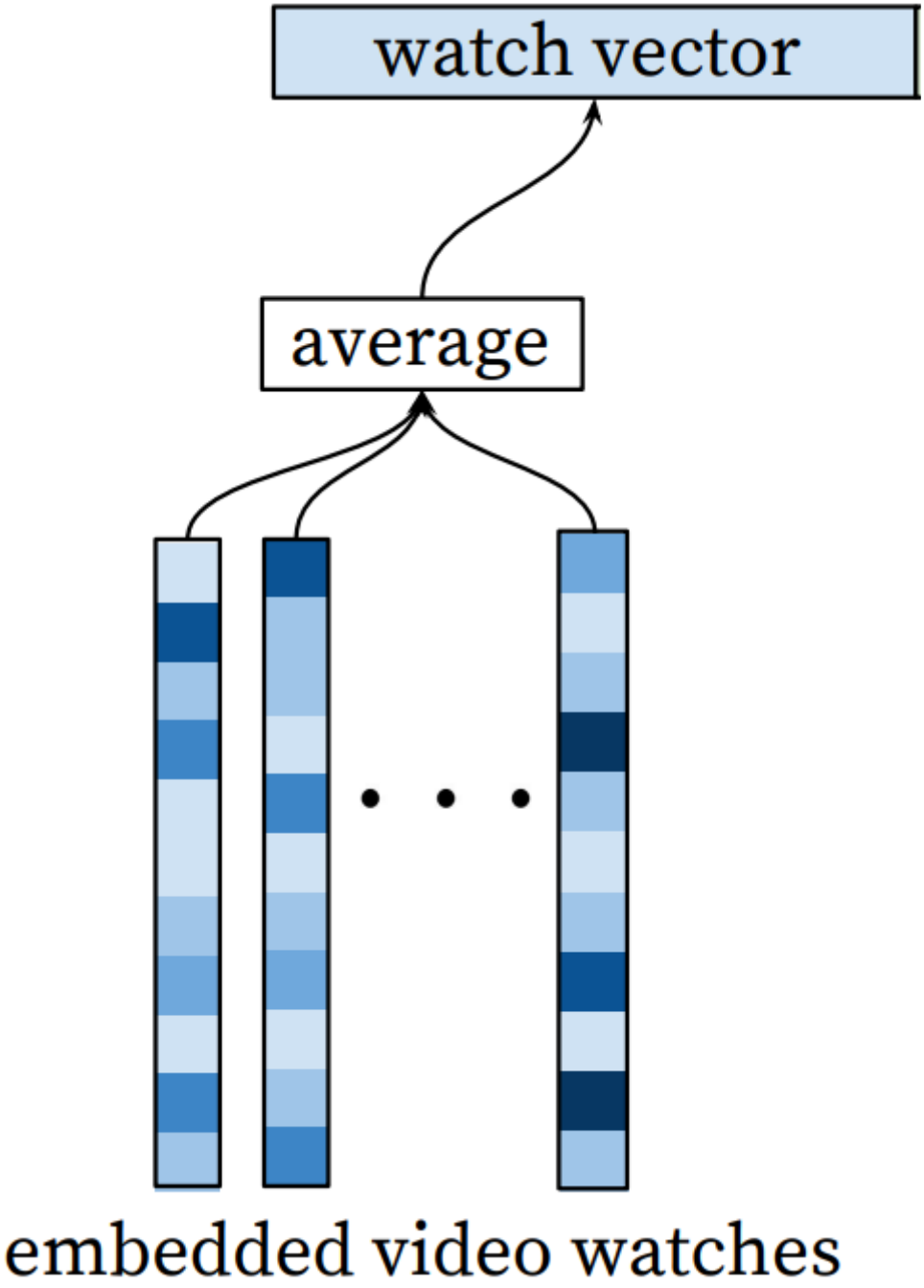
输入层

在 YouTubeDNN 的召回部分中，输入层里面一共取了以下 4 种特征。

1. 用户看过视频的 Embedding，也就是 embedded video watches。
2. 用户搜索的关键词的 Embedding 向量，也就是 embedded search tokens。
3. 用户所在的地理位置的特征，也就是这里面的 geographic embedding。
4. 用户的基本特征，包括年龄、性别等等。

在处理观看的视频序列和搜索词时需要格外注意，每一个人所观看的内容的数量、长度一定不同、所搜索的关键词也一定不同，所产生出来的 Embedding 也会千差万别。**这时我们就需要用多值特征和平均池化进行处理。**

每个人的序列长度不同会导致产生出来的特征长度不同。我们在进行训练时就需要将多值特征中的每一个视频 ID 经过 Embedding Lookup 操作后，得到其对应的 Embedding 向量，然后再经过一层平均池化处理，最后得到这个多值特征所对应的 Embedding 特征，也就是下面这个部分。

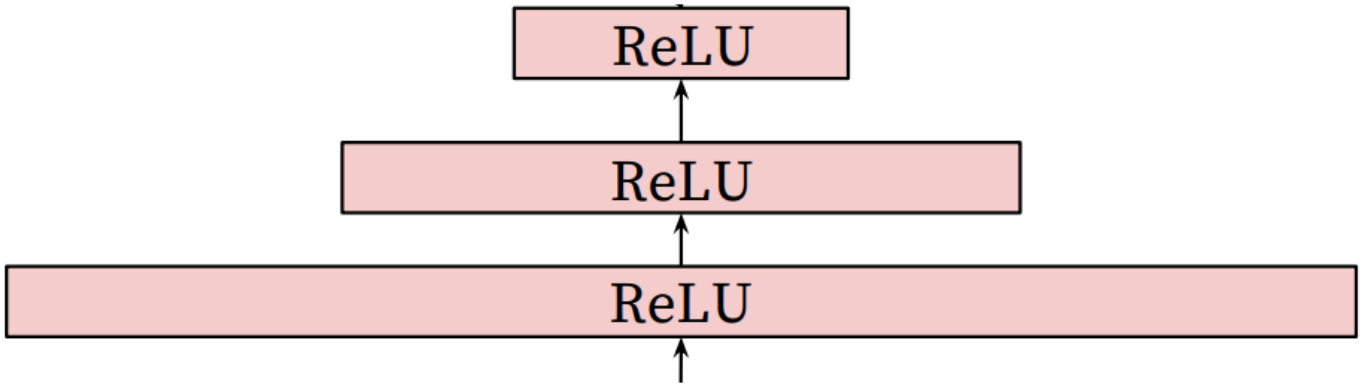


无论是对于观看的视频还是搜索的关键词，都是采用这样的方式，通过平均池化，得到了最后的 Embedding，然后将观看视频的历史、搜索的 Tokens，以及用户的地理信息 Embedding 和用户的其他信息 Embedding 拼接起来形成一个长度很大的 Embedding，这个 Embedding 就是我们最后要输入给模型的 Embedding。

三层神经网络

当通过输入层得到了需要输入到模型的 Embedding 向量后，接下来要做的就是把它们送到模型里进行训练。

YouTubeDNN 的召回层中所使用到的模型结构其实非常简单，就是用了三层 ReLU 结构，最后再接入了一层 Softmax 进行预测，我们来看下这三层 ReLU 结构。



实际上在 YouTubeDNN 中，这三层 ReLU 的作用就是接收输入层特征的 CONCAT，然后使用常见的塔形设计，对自底向上的每一层神经元数目作减半处理，直到得到的输出维度与 Softmax 所要求的输入维度相同（也就是 256 维）。所以在这三层 ReLU 中，我们经历的维度变化就是 1024ReLU、512ReLU、256ReLU。

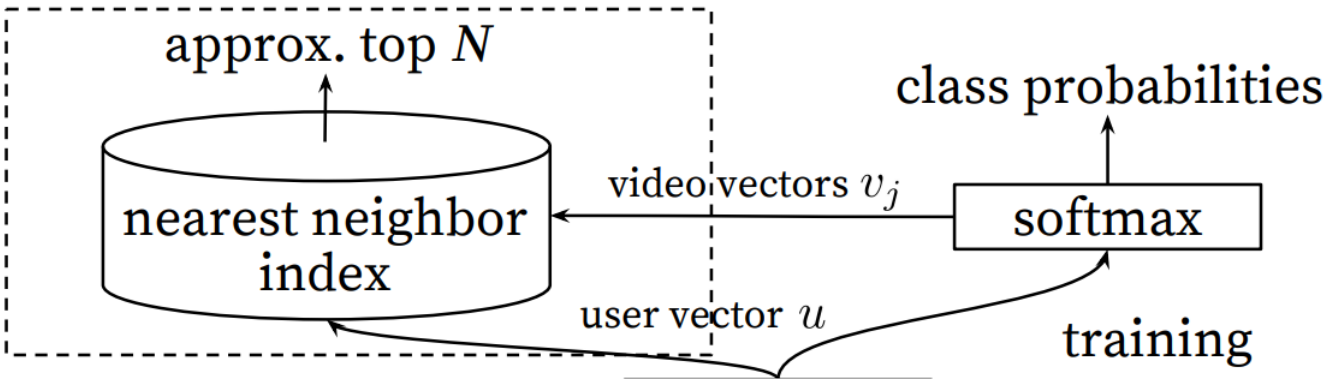
这里简单说一下 ReLU 函数。ReLU 是一个激活函数，在卷积神经网络中使用该函数是为了去除卷积结果中的负值，保留正值不变。ReLU 激活函数只在输入大于 0 时才激活一个节点，此时输出等于输入；当输入小于 0 时，输出为零。

在深度学习网络（DNN）中，由于训练的数据量和神经网络的参数较大，就会产生过拟合的现象。ReLU 会使一部分神经元的输出为 0，这样就造成了网络的稀疏性，并且减少了参数的相互依存关系，缓解了过拟合问题的发生。

在这里之所以使用 DNN，是因为在 DNN 中连续性变量和类别型变量都很容易输入到模型中，并且用户的信息（以及一些可以简单进行二值化的特征和数值特征）都可以很容易地训练 DNN。最后数值型经过归一化之后，再输入到下一层模型中。

Softmax 层

在经过三层 ReLU 之后，召回层使用 Softmax 作为输出层，我们来看一下 Softmax 这一层的结构。



我们知道，传入 Softmax 层的参数是用户的 Embedding 向量，而这里的用户的 Embedding 并不是在输入层里面的用户信息，也就是说，在 Softmax 这一层所传入的用户的 Embedding 并不是提前计算好的，而是根据输入的特征信息实时计算得到，这里面的用户信息实际上就是最后一层 ReLU 的输出。

在 Softmax 这一层中需要注意的是，最终的输出并不是点击率的预估，而是预测用户最终会点击哪个视频。首先看右侧的 Softmax 部分，这一部分实际上是把上一步 ReLU 输出的向量 u 接到了 Softmax 层，得到其概率分布。

YouTubeDNN 把每一个视频当做一个类别，这里的 Softmax 可以理解为每一个视频进行了一个概率上的打分。从论文的角度来说，就是将用户的向量 u 和视频的向量 v 进行内积（这个内积实际上就是求其相似度然后再进行 Softmax），这样就可以得到用户观看每个视频的概率，也就是上面这张图中所展示的 class probabilities。

实际上右侧的 Softmax 部分可以理解为是在做一个离线的 Training，之所以要这么做，其实主要是考虑到工程效率方面的问题，可以加快线上的预测效率。

当得到右侧最终的用户向量之后，为了与离线的训练保持一致，仍然需要对每一个视频，也就是 Item 向量进行内积运算，然后得到概率最高的 N 个结果作为输出。

我们看左面的这个部分中有一个 nearest neighbor index，这个部分就是核心所在。这里 YouTubeDNN 采用最临近搜索的方法去完成 topN 的推荐，通过召回模型得到用户的向量和 Item 的向量做内积，然后再用最临近搜索来得到最后的 topN，这样的效率实际上是最高的。

到这里，我们就整体过了一遍 YouTubeDNN 的召回模型结构。

YouTubeDNN 的一些 trick

讲完召回模型之后，我们来说一下在 YouTubeDNN 中一些值得拿出来探讨的 trick，这些内容更加偏向于工程。

采样问题

YouTubeDNN 在采样中的一个 trick 就是进行了负采样。这里训练的样本来自全部观看记录，也就是说，观看记录包括用户被推荐的内容，再加上用户自己搜索或者在其他地方点击的内容，这样做的好处是可以使新的视频也能够有比较好的曝光。

作者将用户看完的内容作为正样本，再从视频库里随机选取一些样本作为负样本（这里的随机选取一般是用户没有看过或者是没有给用户曝光过的样本）。这里实际上有一个相对取巧的操作：没有曝光过的内容理论上有可能被点击过，但是这里作者把它们全都变成了负样本，使其有一定的随机性。

在采样部分，作者在训练数据中对每个用户选取相同的样本数。这样做的目的是保证用户在损失函数的权重是相等的，这样可以尽可能地减少高度活跃的用户对整个推荐结果的 Loss 的影响。

另外我们要知道，加入负采样的目的是提高训练的速度。为什么这么说呢？因为正常来讲对于每一个样本来说，所有的视频都可能是正样本。而对于 YouTubeDNN 来说，Softmax 实际上对每一个视频都有一个概率。

这样大体量的视频集如果不做负采样将会是一个上千万的分类，这样显然加大了训练难度。所以作者正常提取正样本，把负样本从视频库中根据重要性进行抽样，这样就很容易缓解训练的压力，提高了整体的效率。

特征构造

YouTubeDNN 在特征构造中也有一些 trick。

首先对于一些简单的特征（比如年龄、性别以及一些连续值的特征）没有经过特殊的处理，直接输入，然后做一层归一化后，把最终的结果压缩到[0, 1]范围内。

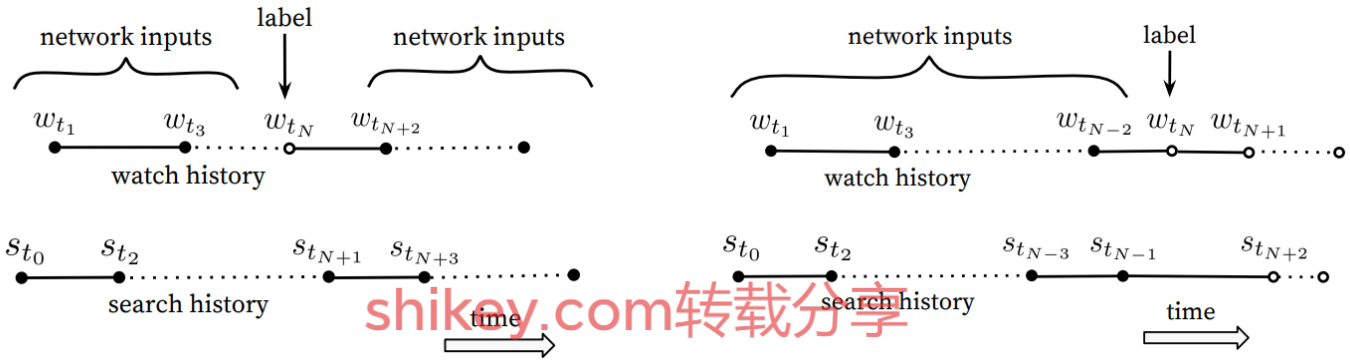
其次，对于用户观看的视频（也就是 Item 这一块）在模型中并没有取视频的特征，只是简单将视频 ID 作为特征传入进来，然后利用 DNN 来自动学习商品的 Embedding 特征。

最后，因为历史信息是一个边长的视频 ID 序列，根据 ID 序列和 Embedding 视频矩阵来获取用户历史观看的 Embedding 向量序列，通过加权、平均或注意力机制等方法，将这个 Embedding 序列映射成一个定长的 Watch Vector，作为输入层的一部分。

上下文的选择

在上下文和标签的选择中，主要利用了序列的方法。用户每次进入 YouTube 一直到退出都有一定的序列，而这个序列大部分情况下会选择范围比较广的视频进行浏览，然后再慢慢地专注到一个比较小的范围内。按照冷启动的 EE 问题来讲，就是从探索到了开采。

这种浏览行为实际上就是一种不对称的浏览行为。因此，作者的做法就是从观看序列中随机抽取一个视频作为 Label，这样就忽略了这种不对称性所产生的影响。大概的示意图如下。



(a) Predicting held-out watch　　(b) Predicting future watch

实际上这张图是作者的一个对比实验。左边这种是许多算法都会用到的方式，就是利用全局观看的信息作为输入，这种方法忽略了观看序列的不对称性。而在 YouTubeDNN 中选用的是右边的方法，就是把历史信息当做输入，用历史来预测未来。这样做的好处在于，模型的测试集往往也是用户最近一次的观看行为，后面的实验可以把用户最后一次的点击放到测试集中，防止信息穿越的问题，这种方式实际上在线上的 AB Test 中表现更佳。

总结

到这里，今天的课程就讲完了，我们来对今天的课程做一个简单的总结，学完本节课你应该知道以下四个要点。

1. 基于 YouTubeDNN 的召回是一种基于 YouTube 推荐系统的深度学习模型，用于预测用户可能感兴趣的视频和其他内容，并将这些内容推荐给用户。这种召回模型使用了一种称为 Deep Neural Network（DNN）的深度学习算法，通过学习过去用户的观看历史、搜索查询和浏览行为等信息，来预测用户的兴趣和偏好，从而推荐相关的视频。
2. 基于 YouTubeDNN 的召回实际上也是双塔模型中的一种。
3. YouTubeDNN 主要分成三层，即输入层、DNN 层和 Softmax 输出层。
4. 我们也知道了 YouTubeDNN 中用到了很多 trick，比如对数据集的采样、负样本的生成加快训练速度、特征构造以及上下文选择等。

思考题

这节课学完了，给你留 2 道思考题。

1. 查阅 YouTubeDNN 的相关资料，了解下还有哪些这节课中没有提到的 trick。
2. 想一想，如果是信息流图文推荐，这里的特征应该怎么做？

期待你的分享，如果今天的内容让你有所收获，也欢迎你推荐给有需要的朋友！

精选留言 (1)

爱极客
2023-06-02 来自广东

老师，这节课的理论，后面会有使用案例吗？