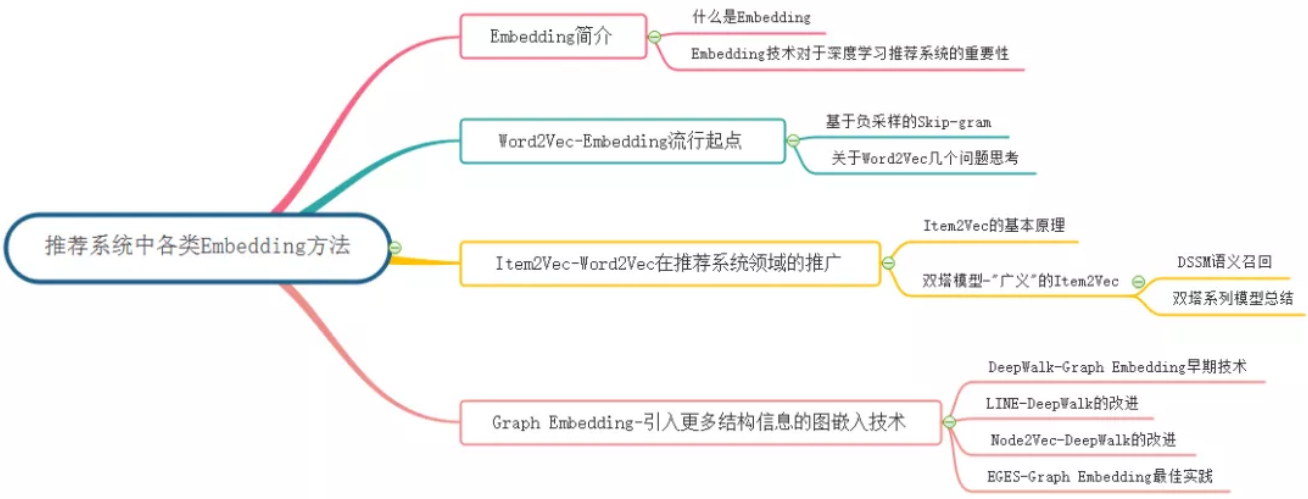
深度学习推荐系统中各类流行的Embedding方法（上）

原创 Microstrong Microstrong 5月11日

收录于话题
#推荐系统原理解析与实践部署

6个

Embedding方法概览：



1. Embedding简介

Embedding，中文直译为“嵌入”，常被翻译为“向量化”或者“向量映射”。在整个深度学习框架中都是十分重要的“基本操作”，不论是NLP（Natural Language Processing，自然语言处理）、搜索排序，还是推荐系统，或是CTR（Click-Through-Rate）模型，Embedding都扮演着重要的角色。

1.1 什么是Embedding

形式上讲，Embedding就是用一个低维稠密的向量“表示”一个对象，这里所说的对象可以是一个词（Word2Vec），也可以是一个物品（Item2Vec），亦或是网络关系中的节点（Graph Embedding）。其中“表示”这个词意味着Embedding向量能够表达相应对象的某些特征，同时向量之间的距离反映了对象之间的相似性。

1.2 Embedding技术对于深度学习推荐系统的重要性

在深度学习推荐系统中，为什么说Embedding技术对于深度学习如此重要，甚至可以说是深度学习的“基本核心操作”呢？原因主要有以下四个：

（1）在深度学习网络中作为Embedding层，完成从高维稀疏特征向量到低维稠密特征向量的转换（比如Wide&Deep、DIN等模型）。 推荐场景中大量使用One-hot编码对类别、Id型特征进行编码，导致样本特征向量极度稀疏，而深度学习的结构特点使其不利于稀疏特征向量的处理，因此几乎所有的深度学习推荐模型都会由Embedding层负责将高维稀疏特征向量转换成稠密低维特征向量。因此，掌握各类Embedding技术是构建深度学习推荐模型的基础性操作。

（2）作为预训练的Embedding特征向量，与其他特征向量连接后，一同输入深度学习网络进行训练（比如FNN模型）。 Embedding本身就是极其重要的特征向量。相比MF等传统方法产生的特征向量，Embedding的表达能力更强，特别是Graph Embedding技术被提出后，Embedding几乎可以引入任何信息进行编码，使其本身就包含大量有价值的信息。在此基础上，Embedding向量往往会与其他推荐系统特征连接后一同输入后续深度学习网络进行训练。

（3）通过计算用户和物品的Embedding相似度，Embedding可以直接作为推荐系统的召回层或者召回策略之一（比如Youtube推荐模型等）。 Embedding对物品、用户相似度的计算是常用的推荐系统召回层技术。在局部敏感哈希（Locality-Sensitive Hashing）等快速最近邻搜索技术应用于推荐系统后，Embedding更适用于对海量备选物品进行快速“筛选”，过滤出几百到几千量级的物品交由深度学习网络进行“精排”。

（4）通过计算用户和物品的Embedding，将其作为实时特征输入到推荐或者搜索模型中（比如Airbnb的Embedding应用）。

2. Word2Vec-Embedding流行起点

关于Word2Vec的入门文章请看我之前的一篇文章：深度浅出Word2Vec原理解析，Microstrong，地址：https://mp.weixin.qq.com/s/zDneR1BU6xvt8cndEF4_Xw

2.1 基于负采样的Skip-gram

这里我单独把基于负采样的Skip-gram模型再详细描述一次，是因为这个模型太重要了，稍后我们讲解的Item2Vec模型和Airbnb论文《Real-time Personalization using Embeddings for Search Ranking at Airbnb 》提出的模型都借鉴了基于负采样的Skip-gram模型的思想。所以，我们务必要把基于负采样的Skip-gram模型理解透彻。

Skip-gram模型是由Mikolov等人提出的。下图展示了Skip-gram模型的过程。该模型可以看做是CBOW模型的逆过程，CBOW模型的目标单词在该模型中作为输入，上下文则作为输出。

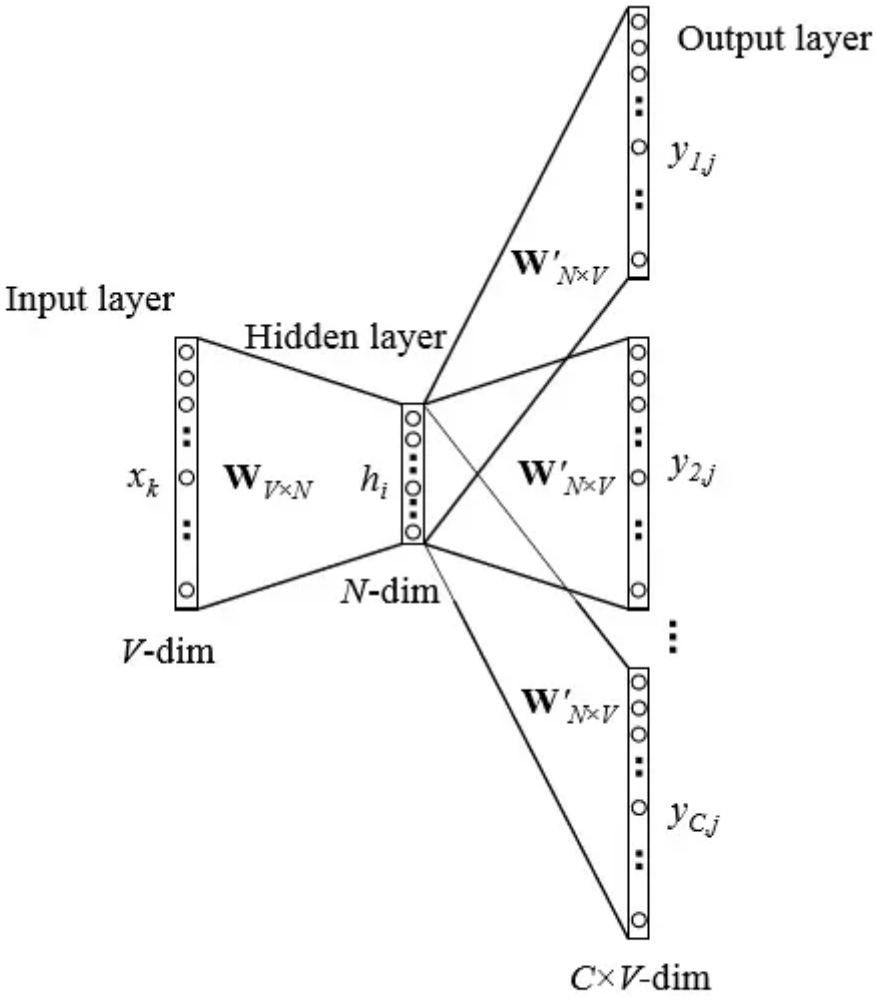


Figure 3: The skip-gram model.

我们使用 v_{wi} 来表示输入层中唯一单词（也叫中心词）的输入向量，所以这样的话，对隐藏层 h 的定义意味着h仅仅只是简单拷贝了输入层到隐藏层的权重矩阵 W 中跟输入单词 w_I 相关的那一行。拷贝公式得到：

$$h = W^T_{(k,\cdot)} := V^T_{w_I}$$

在输出层，我们输出 C 个多项式分布来替代仅输出一个多项式分布。每个输出是由同一个隐藏层到输出层矩阵计算得出的：

$$p(w_{c,j} = w_{O,c}|w_I) = y_{c,j} = \frac{exp(u_{c,j})}{\sum_{j'=1}^{V} exp(u_{j'})}$$

这里$w_{c,j}$是第c个输出面上第j个单词；$w_{O,c}$是中心词对应的目标单词中的第c个单词；w_I 是中心词（唯一输入单词）；$y_{c,j}$是第c个输出面上第j个单元的输出值；$u_{c,j}$是第c个输出面上的第j个单元的输入。因为输出面共享同一权重矩阵，所以有：

$$u_{c,j} = u_j = V^{'T}_{w_j} \cdot h, \quad for \quad c = 1, 2, \cdots, C$$

$V^{'}_{wj}$是词汇表第j个单词的输出向量，可由$W^{'}$ 矩阵中的所对应的一列拷贝得到。

Skip-gram的损失函数可以写为：

$$
\begin{aligned}
E &= -logp(w_{O,1}, w_{O,2}, \dots, w_{O,C}|w_I) \\
&= -log\prod_{c=1}^{C}\frac{exp(u_{c,j_c^*})}{\sum_{j'=1}^{V}exp(u_{j'})} \\
&= -\sum_{c=1}^{C}u_{j_c^*} + C\cdot log\sum_{j'=1}^{V}exp(u_{j'})
\end{aligned}
$$

由于语料库非常的大，直接计算中心词与词库中每个单词的softmax概率不现实。为了解决这个问题，Google提出了两个方法，一个是Hierarchical Softmax，另一个方法是Negative Sampling。Negative Sampling的思想本身源自于对Noise Contrastive Estimation的一个简化，具体的，把目标函数修正为：

$$
log\sigma(v_{w_o}^{'T}v_{w_I}) + \sum_{i=1}^{k}E_{w_i\sim P_n(w)}[log\sigma(-v_{w_i}^{'T}v_{w_I})]
$$

$P_n(w)$是噪声分布 (noise distribution)。即训练目标是使用Logistic regression区分出目标词和噪音词。另外，由于自然语言中很多高频词出现频率极高，但包含的信息量非常小（如'is' 'a' 'the'）。为了平衡低频词和高频词，利用简单的概率丢弃词w_i：

$$
P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}
$$

其中$f(w_i)$是w_i 的词频，t的确定比较trick，启发式获得。实际中t大约在10^{-5} 附近。

推荐阅读Airbnb论文：

【1】Grbovic M, Cheng H. Real-time personalization using embeddings for search ranking at airbnb[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 311-320.

2.2 关于Word2Vec几个问题思考

熟悉了Word2Vec的基本原理后，我们来探究几个关于Word2Vec的经典面试题。

（1） Hierarchical Softmax方法中哈夫曼树是如何初始化生成的？也就是哈夫曼树是如何构建的呢？

答：Hierarchical Softmax依据词频构建Huffman树，词频大的节点离根节点较近，词频低的节点离根节点较远，距离远参数数量就多。

（2）Hierarchical Softmax对词频低的和词频高的单词有什么影响？为什么？

答：词频高的节点离根节点较近，词频低的节点离根节点较远，距离远参数数量就多，在训练的过程中，低频词的路径上的参数能够得到更多的训练，所以Hierarchical Softmax对词频低的单词效果会更好。

（3）Hierarchical Softmax方法中其实是做了一个二分类任务，写出它的目标函数？

答：$L(\theta)=\sum_{i=1}^{m} y_i log h_\theta(x_i)+(1-y_i)log(1-h_\theta(x_i))$，其中 y_i 是真实标签，$h_\theta(x_i)$是模型预测的标签。

（4）Negative Sampling是一种什么采样方式？是均匀采样还是其它采样方法？

答：词典 D 中的词在语料 C 中出现的次数有高有低，对于那些高频词，被选为负样本的概率就应该比较大，反之，对于那些低频词，其被选中的概率就应该比较小。这就是我们对采样过程的一个大致要求，本质上就是一个带权采样问题。

（5）详细介绍一下Word2Vec中负采样方法？

答：先将概率以累积概率分布的形式分布到一条线段上，以 $a=0.2, b=0.3, c=0.5$ 为例， a 所处线段为 $[0,0.2]$，b所处线段为 $[0.2,0.5]$， c 所处线段为 $[0.5,1]$，然后定义一个大小为 M 的数组，并把数组等距离划分为 m 个单元，然后与上面的线段做一次映射，这样我们便知道了数组内的每个单元所对应的字符了，这种情况下算法的时间复杂度为$O(1)$，空间复杂度为 $O(M)$， m 越小精度越大。

最后，我们来聊一聊Word2Vec对Embedding技术的奠基性意义。Word2Vec是由谷歌于2013年正式提出的，其实它并不完全由谷歌原创，对词向量的研究可以追溯到2003年论文《a neural probabilistic language model》，甚至更早。但正是谷歌对Word2Vec的成功应用，让词向量的技术得以在业界迅速推广，使Embedding这一研究话题成为热点。毫不夸张地说，Word2Vec对深度学习时代Embedding方向的研究具有奠基性的意义。

从另一个角度看，在Word2Vec的研究中提出的模型结构、目标函数、负采样方法及负采样中的目标函数，在后续的研究中被重复使用并被屡次优化。掌握Word2Vec中的每一个细节成了研究Embedding的基础。从这个意义上讲，熟练掌握本节内容非常重要。

3. Item2Vec-Word2Vec在推荐领域的推广

在Word2Vec诞生之后，Embedding的思想迅速从自然语言处理领域扩散到几乎所有机器学习领域，推荐系统也不例外。既然Word2Vec可以对词“序列”中的词进行Embedding，那么对于用户购买“序列”中的一个商品，用户观看“序列”中的一个电影，也应该存在相应的Embedding方法，这就是Item2Vec方法的基本思想。

推荐阅读Item2Vec论文：

【1】Barkan O , Koenigstein N . Item2Vec: Neural Item Embedding for Collaborative Filtering[J]. 2016.

3.1 Item2Vec的基本原理

由于Word2Vec的流行，越来越多的Embedding方法可以被直接用于物品Embedding向量的生成，而用户Embedding向量则更多通过行为历史中的物品Embedding平均或者聚类得到。利用用户向量和物品向量的相似性，可以直接在推荐系统的召回层快速得到候选集合，或在排序层直接用于最终推荐列表的排序。正是基于这样的技术背景，微软于2016年提出了计算物品Embedding向量的方法Item2Vec。

微软将Skip-gram with negative sampling（SGNS）应用在用户与物品的交互数据中，因此将该方法命名为Item2Vec。相比Word2Vec利用“词序列”生成词Embedding。Item2Vec利用的“物品集合”是由特定用户的浏览、购买等行为产生的历史行为记录序列变成物品集合。通过从物品序列移动到集合，丢失了空间/时间信息，还无法对用户行为程度建模（喜欢和购买是不同程度的强行为）。好处是可以忽略用户和物品关系，即便获得的订单不包含用户信息，也可以生成物品集合。而论文的结论证明，在一些场景下序列信息的丢失是可忍受的。

Item2Vec中把用户浏览的商品集合等价于Word2Vec中的word的序列，即句子（忽略了商品序列空间/时间信息）。出现在同一个集合的商品被视为positive。对于用户历史行为物品集合$w_1, w_2, \ldots, w_K$，Item2Vec的优化目标函数为：

$$\frac{1}{K}\sum_{i=1}^{K}\sum_{j=1}^{K} logp(w_j|w_i)$$

我们再来看一下，Skip-gram是如何定义目标函数的。给定一个训练序列$w_1, w_2, \ldots, w_T$，模型的目标函数是最大化平均的log概率：

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} logp(w_{t+j}|w_t)$$

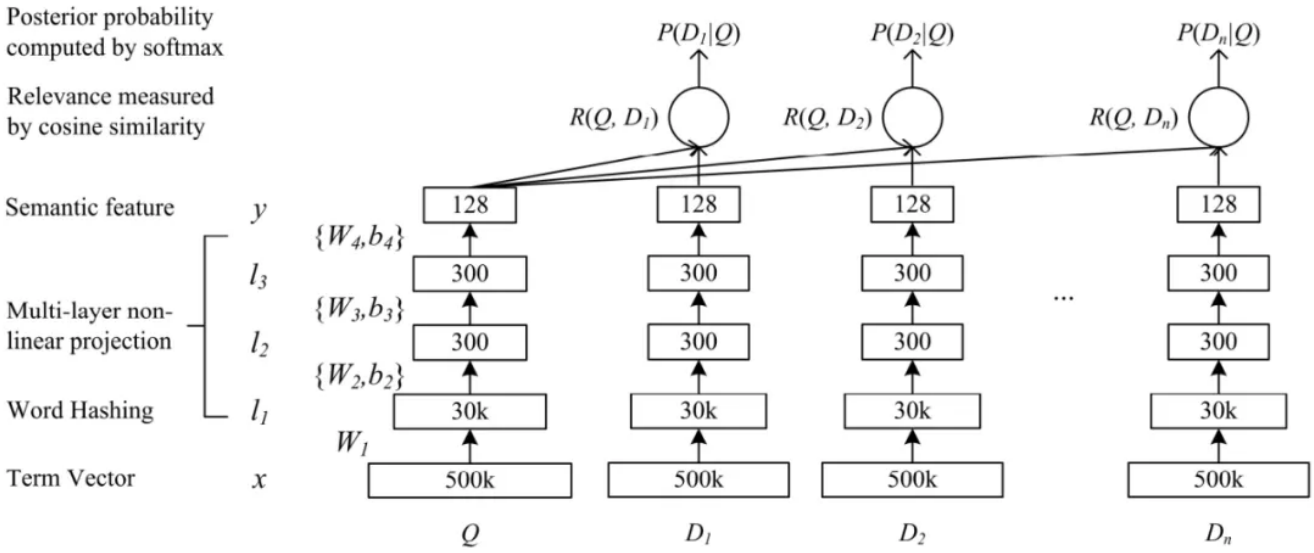
通过观察上面两个式子会发现，Item2Vec和Word2Vec唯一的不同在于，Item2Vec丢弃了时间窗口的概念，认为序列中任意两个物品都相关，因此在Item2Vec的目标函数中可以看到，其是两两物品的对数概率的和，而不仅是时间窗口内物品的对数概率之和。

在优化目标定义好后，Item2Vec训练和优化过程同Word2Vec一样，利用负采样，最终得到每个商品的Embedding representation。利用商品的向量表征计算商品之间两两的cosine相似度即为商品的相似度。

3.2 双塔模型-"广义"的Item2Vec

3.2.1 DSSM语义召回

DSSM模型是微软2013年发表的一个关于query/doc的相似度计算模型，后来发展成为一种所谓”双塔“的框架广泛应用于广告、推荐等领域的召回和排序问题中。我们自底向上来看下图所示的网络结构：



- 首先特征输入层将Query和Doc（One-hot编码）转化为Embedding向量，原论文针对英文输入还提出了一种word hashing的特殊Embedding方法用来降低字典规模。我们在针对中文Embedding时使用Word2Vec类常规操作即可；
- 经过Embedding之后的词向量，接下来是多层DNN网络映射得到针对Query和Doc的128维语义特征向量；
- 最后会使用Query和Doc向量进行余弦相似度计算得到相似度R，然后进行softmax归一化得到最终的指标后验概率P，训练目标针对点击的正样本拟合P为1，否则拟合P为0；

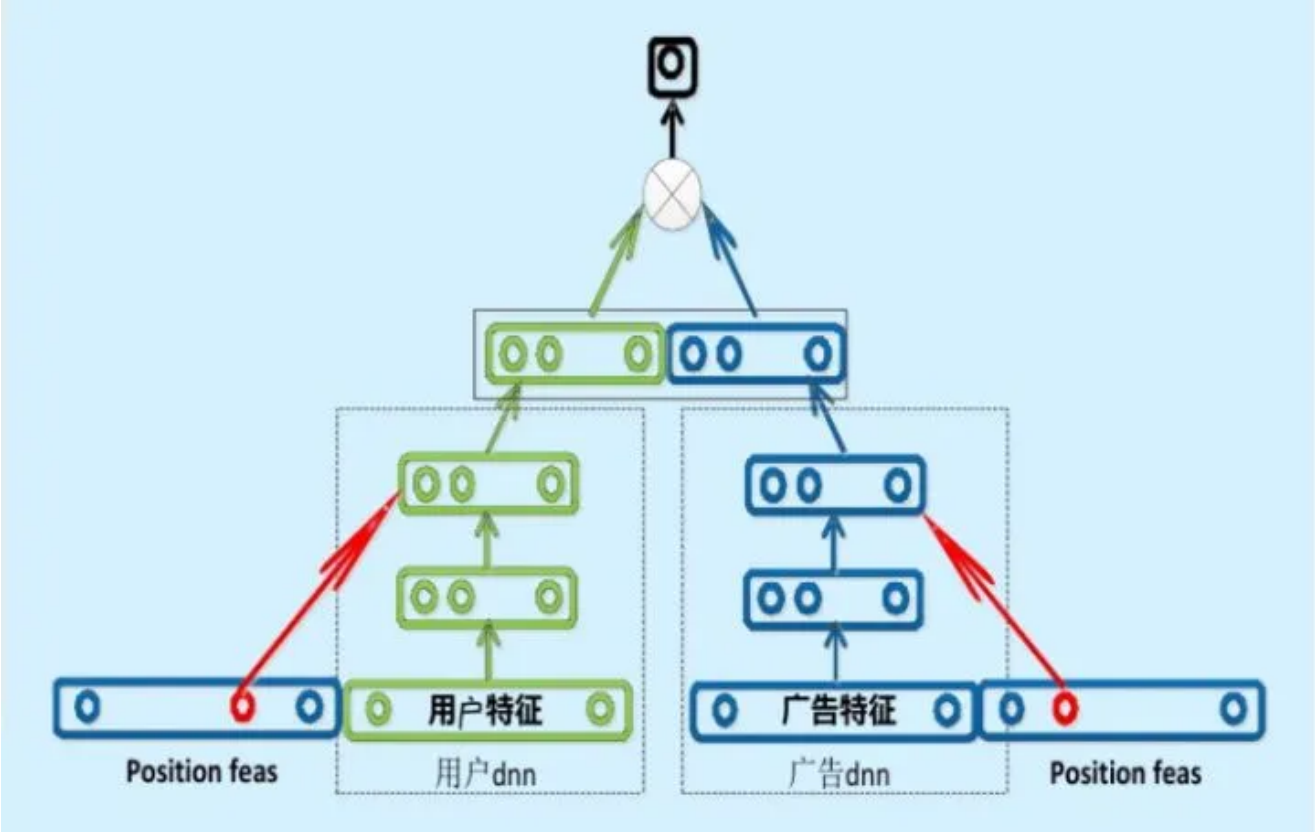
可以看到DSSM的核心思想就是将不同对象映射到统一的语义空间中，利用该空间中对象的距离计算相似度。这一思路被广泛应用到了广告、搜索以及推荐的召回和排序等各种工程实践中。

推荐阅读DSSM论文：

【1】Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data[C]// ACM International Conference on Conference on Information & Knowledge Management. ACM, 2013:2333-2338.

3.2.2 双塔系列模型总结

事实上，Embedding对物品进行向量化的方法远不止Item2Vec。广义上讲，任何能够生成物品向量的方法都可以被称为Item2Vec。典型的例子是曾在百度、Facebook等公司成功应用的双塔模型，如下图所示。



图：百度的“双塔”模型（来源于参考文献11）

百度的双塔模型分别用复杂网络对“用户特征”和“广告特征”进行了Embedding化，在最后的交叉层之前，用户特征和广告特征之间没有任何交互，这就形成了两个独立的“塔”，因此称为双塔模型。

在完成双塔模型的训练后，可以把最终的用户Embedding和广告Embedding存入内存数据库。而在线上inference时，也不用复现复杂网络，只需要实现最后一层的逻辑（即线上实现LR或浅层NN等轻量级模型拟合优化目标），再从内存数据库中取出用户Embedding和广告Embedding之后，通过简单计算即可得到最终的预估结果。

总之，在广告场景下的双塔模型中，广告侧的模型结构实现的其实就是对物品进行Embedding的过程。该模型被称为“双塔模型”，因此以下将广告侧的模型结构称为“物品塔”。那么，“物品塔”起到的作用本质上是接收物品相关的特征向量，经过“物品塔”内的多层神经网络结构，最终生成一个多维的稠密向量。从Embedding的角度看，这个稠密向量其实就是物品的Embedding向量，只不过Embedding模型从Word2Vec变成了更为复杂灵活的“物品塔”模型，输入特征由用户行为序列生成的One-hot特征向量，变成了可包含更多信息的、全面的物品特征向量。二者的最终目的都是把物品的原始特征转变为稠密的物品Embedding向量表达，因此不管其中的模型结构如何，都可以把这类模型称为“广义”上的Item2Vec类模型。

Word2Vec和其衍生出的Item2Vec类模型是Embedding技术的基础性方法，但二者都是建立在“序列”样本（比如句子、用户行为序列）的基础上的。在互联网场景下，数据对象之间更多呈现的是图结构，所以Item2Vec在处理大量的网络化数据时往往显得捉襟见肘，这就是Graph Embedding技术出现的动因。

最后，由于上周时间有限，且不想让本文篇幅过长，我会在下一篇文章中接着给大家详细介绍Graph Embedding的相关内容，请大家持续关注我哈！

双塔模型推荐阅读论文：

【1】Yi X, Yang J, Hong L, et al. Sampling-bias-corrected neural modeling for large corpus item recommendations[C]//Proceedings of the 13th ACM Conference on Recommender Systems. 2019: 269-277.

Reference

【1】《深度学习推荐系统》王喆编著。
【2】Graph Embedding：深度学习推荐系统的"基本操作" - 顾鹏的文章 - 知乎
https://zhuanlan.zhihu.com/p/68247149
【3】第四范式先荐 第5期 图推荐算法在E&E问题上的应用，地址：
https://mp.weixin.qq.com/s/RTAHBIAPQMqCM8WF6trwug
【4】图推荐算法在E&E问题上的应用，地址：
https://mp.weixin.qq.com/s/KSW47hbNLaHTw9Ib0wMO8g
【5】Embedding在推荐算法中的应用总结 - 梦想做个翟老师的文章 - 知乎
https://zhuanlan.zhihu.com/p/78144408
【6】Graph Embedding：深度学习推荐系统的"基本操作" - 顾鹏的文章 - 知乎
https://zhuanlan.zhihu.com/p/68247149
【7】word2vec详解（CBOW，skip-gram，负采样，分层Softmax） - 孙孙的文章 - 知乎
https://zhuanlan.zhihu.com/p/53425736
【8】推荐召回算法之深度召回模型串讲 - 深度传送门的文章 - 知乎
https://zhuanlan.zhihu.com/p/63343894
【9】现阶段各家公司的广告算法使用的主流模型有哪些？ - 付鹏的回答 - 知乎
https://www.zhihu.com/question/352306163/answer/902698330
【10】深度学习技术在美图个性化推荐的应用实践，地址：
https://mp.weixin.qq.com/s/b8DkQWZbUc5-jzWKBd8iUA
【11】如何解决推荐系统工程难题——深度学习推荐模型线上serving？ - 王喆的文章 - 知乎
https://zhuanlan.zhihu.com/p/77664408

长按二维码扫描关注

Microstrong

ID:MicrostrongAI

Microstrong(小强)同学主要研究兴趣是机器学习、深度学习、计算机视觉、智能对话系统相关内容，分享在学习过程中的读书笔记！期待您的关注，欢迎一起学习交流进步！

微信搜一搜

Microstrong