06｜Embedding：所有人都在谈的Embedding技术到底是什么？

2020-10-14 王喆

深度学习推荐系统实战 进入课程 >

讲述：王喆
时长 16:41 大小 15.29M

你好，我是王喆。今天我们聊聊 Embedding。

说起 Embedding，我想你肯定不会陌生，至少经常听说。事实上，Embedding 技术不仅名气大，而且用 Embedding 方法进行相似物品推荐，几乎成了业界最流行的做法，无论是国外的 facebook、Airbnb，还是在国内的阿里、美团，我们都可以看到 Embedding 的成功应用。因此，自从深度学习流行起来之后，Embedding 就成为了深度学习推荐系统方向最火热的话题之一。

但是 Embedding 这个词又不是很好理解，你甚至很难给它找出一个准确的中文翻译，如果硬是翻译成“嵌入”、“向量映射”，感觉也不知所谓。所以索性我们就还是用 Embedding 这个叫法吧。

那这项技术到底是什么，为什么它在推荐系统领域这么重要？最经典的 Embedding 方法 Word2vec 的原理细节到底啥样？这节课，我们就一起来聊聊这几个问题。

什么是 Embedding?

简单来说，**Embedding 就是用一个数值向量“表示”一个对象（Object）的方法**，我这里说的对象可以是一个词、一个物品，也可以是一部电影等等。但是“表示”这个词是什么意思呢？用一个向量表示一个物品，这句话感觉还是有点让人费解。

这里，我先尝试着解释一下：一个物品能被向量表示，是因为这个向量跟其他物品向量之间的距离反映了这些物品的相似性。更进一步来说，两个向量间的距离向量甚至能够反映它们之间的关系。这个解释听上去可能还是有点抽象，那我们再用两个具体的例子解释一下。

图 1 是 Google 著名的论文 word2vec 中的例子，它利用 word2vec 这个模型把单词映射到了高维空间中，每个单词在这个高维空间中的位置都非常有意思，你看图 1 左边的例子，从 king 到 queen 的向量和从 man 到 woman 的向量，无论从方向还是尺度来说它们都异常接近。这说明什么？这说明词 Embedding 向量间的运算居然能够揭示词之间的性别关系！比如 woman 这个词的词向量可以用下面的运算得出：

Embedding(**woman**)=Embedding(**man**)+[Embedding(**queen**)-Embedding(**king**)]

同样，图 1 右的例子也很典型，从 walking 到 walked 和从 swimming 到 swam 的向量基本一致，这说明词向量揭示了词之间的时态关系！

这就是 Embedding 技术的神奇之处。

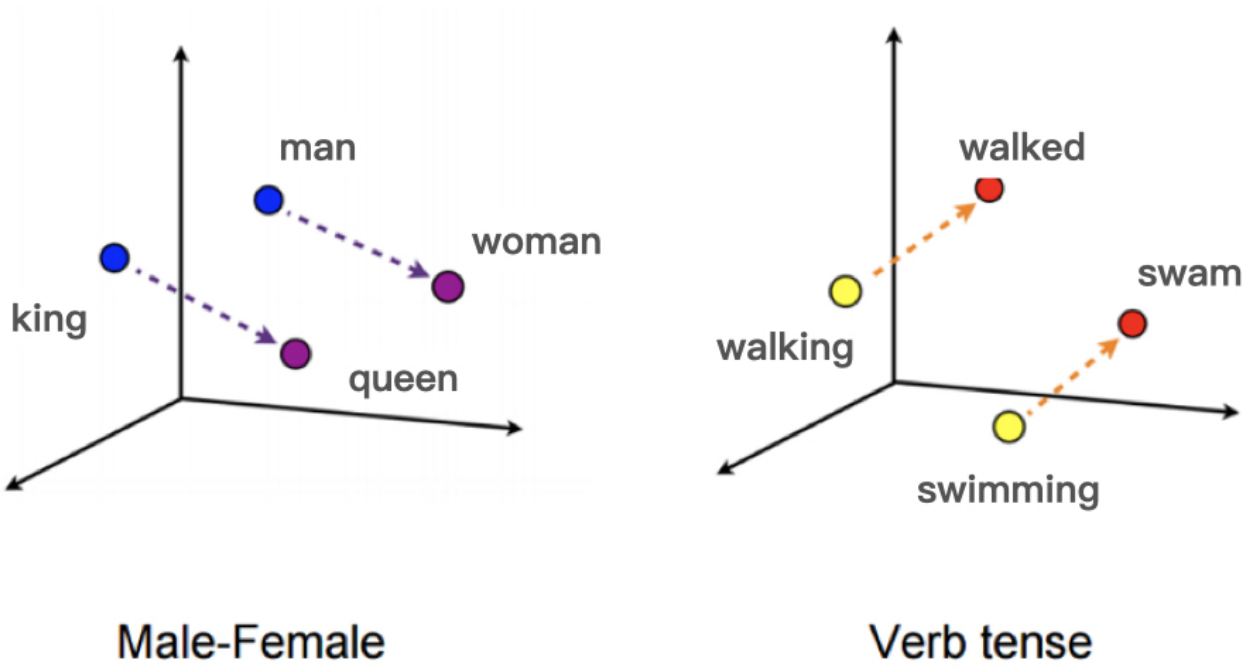

图1 词向量例子

你可能会觉得词向量技术离推荐系统领域还是有一点远，那 Netflix 应用的电影 Embedding 向量方法，就是一个非常直接的推荐系统应用。从 Netflix 利用矩阵分解方法生成的电影和用户的 Embedding 向量示意图中，我们可以看出不同的电影和用户分布在一个二维的空间内，由于 Embedding 向量保存了它们之间的相似性关系，因此有了这个 Embedding 空间之后，我们再进行电影推荐就非常容易了。具体来说就是，我们直接找出某个用户向量周围的电影向量，然后把这些电影推荐给这个用户就可以了。这就是 Embedding 技术在推荐系统中最直接的应用。

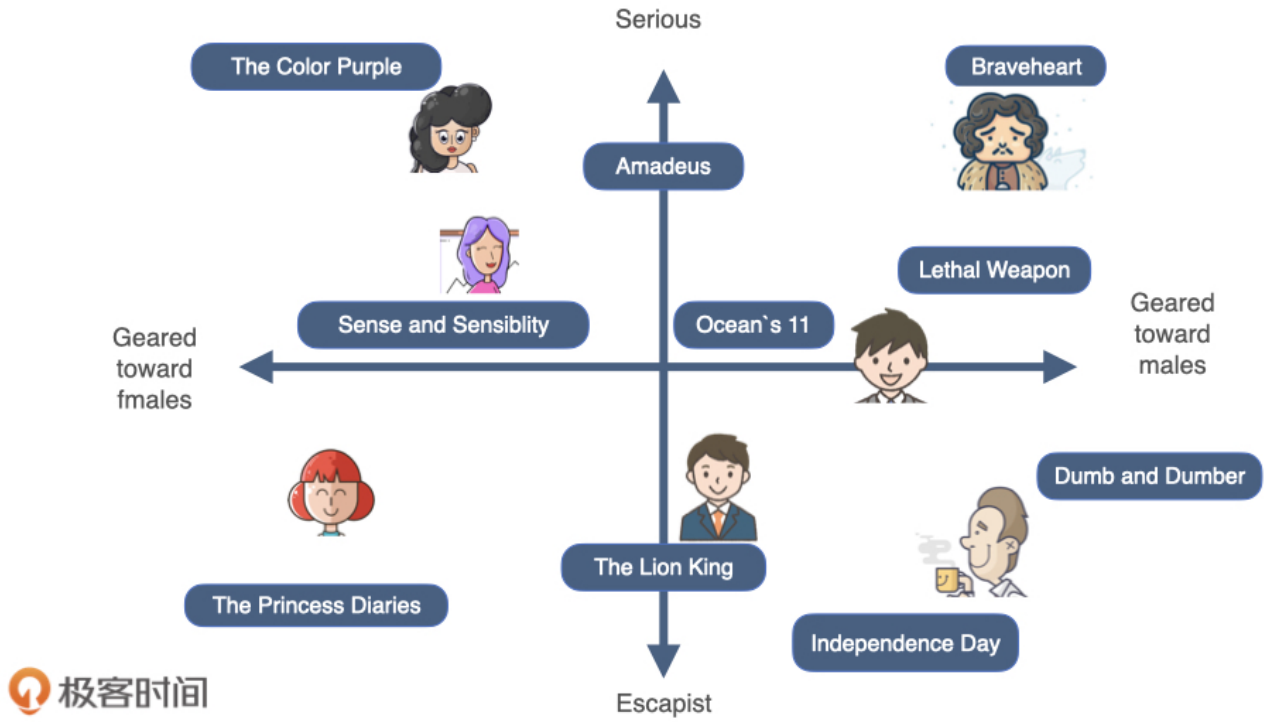

图2 电影-用户向量例子

Embedding 技术对深度学习推荐系统的重要性

事实上，我一直把 Embedding 技术称作深度学习的“基础核心操作”。在推荐系统领域进入深度学习时代之后，Embedding 技术更是“如鱼得水”。那为什么 Embedding 技术对于推荐系统如此重要，Embedding 技术又在特征工程中发挥了怎样的作用呢？针对这两个问题，我主要有两点想和你深入聊聊。

首先，Embedding 是处理稀疏特征的利器。 上节课我们学习了 One-hot 编码，因为推荐场景中的类别、ID 型特征非常多，大量使用 One-hot 编码会导致样本特征向量极度稀疏，而深度学习的结构特点又不利于稀疏特征向量的处理，因此几乎所有深度学习推荐模型都会由 Embedding 层负责将稀疏高维特征向量转换成稠密低维特征向量。所以说各类 Embedding 技术是构建深度学习推荐模型的基础性操作。

其次，Embedding 可以融合大量有价值信息，本身就是极其重要的特征向量。 相比由原始信息直接处理得来的特征向量，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，所以通过预训练得到的 Embedding 向量本身就是极其重要的特征向量。

因此我们才说，Embedding 技术在深度学习推荐系统中占有极其重要的位置，熟悉并掌握各类流行的 Embedding 方法是构建一个成功的深度学习推荐系统的有力武器。**这两个特点也是我们为什么把 Embedding 的相关内容放到特征工程篇的原因，因为它不仅是一种处理稀疏特征的方法，也是融合大量基本特征，生成高阶特征向量的有效手段。**

经典的 Embedding 方法，Word2vec

提到 Embedding，就一定要深入讲解一下 Word2vec。它不仅让词向量在自然语言处理领域再度流行，更关键的是，自从 2013 年谷歌提出 Word2vec 以来，Embedding 技术从自然语言处理领域推广到广告、搜索、图像、推荐等几乎所有深度学习的领域，成了深度学习知识框架中不可或缺的技术点。Word2vec 作为经典的 Embedding 方法，熟悉它对于我们理解之后所有的 Embedding 相关技术和概念都是至关重要的。下面，我就给你详细讲一讲 Word2vec 的原理。

什么是 Word2vec?

Word2vec 是“word to vector”的简称，顾名思义，它是一个生成对“词”的向量表达的模型。

想要训练 Word2vec 模型，我们需要准备由一组句子组成的语料库。假设其中一个长度为 _T_ 的句子包含的词有$w_1,w_2.....w_t$，并且我们假定每个词都跟其相邻词的关系最密切。

根据模型假设的不同，Word2vec 模型分为两种形式，CBOW 模型（图 3 左）和 Skip-gram 模型（图 3 右）。其中，CBOW 模型假设句子中每个词的选取都由相邻的词决定，因此我们就看到 CBOW 模型的输入是 wt 周边的词，预测的输出是 wt。Skip-gram 模型则正好相反，它假设句子中的每个词都决定了相邻词的选取，所以你可以看到 Skip-gram 模型的输入是 wt，预测的输出是 wt 周边的词。按照一般的经验，Skip-gram 模型的效果会更好一些，所以我接下来也会以 Skip-gram 作为框架，来给你讲讲 Word2vec 的模型细节。

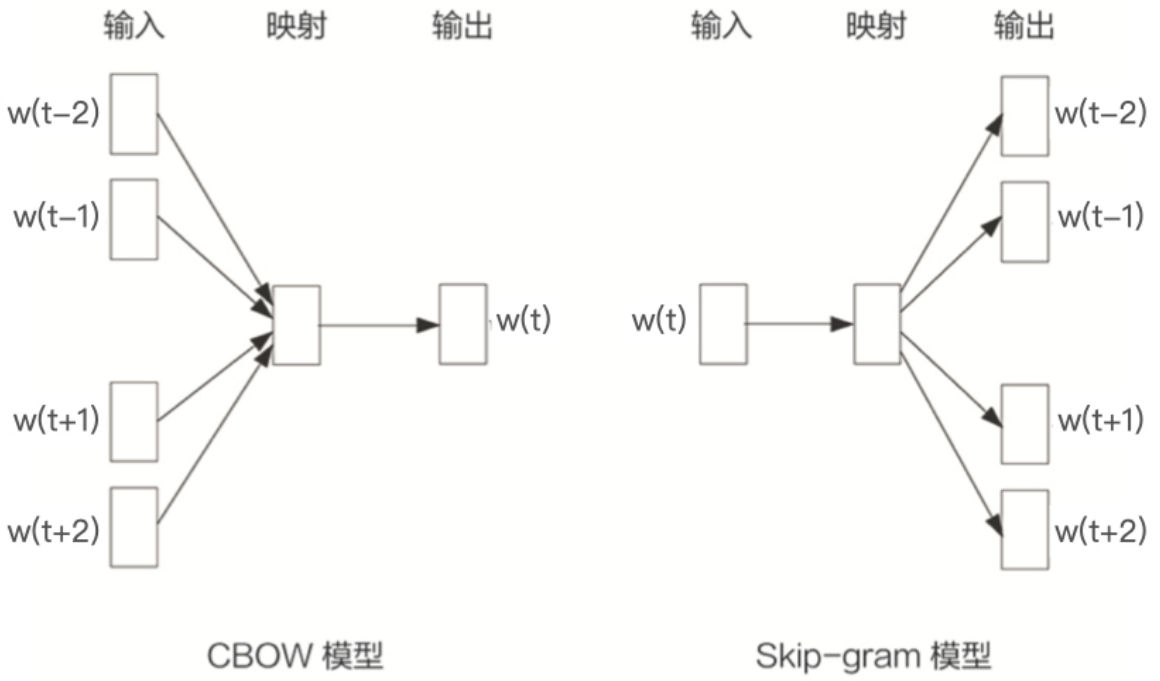

图3 Word2vec的两种模型结构CBOW和Skip-gram

Word2vec 的样本是怎么生成的?

我们先来看看**训练 Word2vec 的样本是怎么生成的。** 作为一个自然语言处理的模型，训练 Word2vec 的样本当然来自于语料库，比如我们想训练一个电商网站中关键词的 Embedding 模型，那么电商网站中所有物品的描述文字就是很好的语料库。

我们从语料库中抽取一个句子，选取一个长度为 2_c_+1（目标词前后各选 _c_ 个词）的滑动窗口，将滑动窗口由左至右滑动，每移动一次，窗口中的词组就形成了一个训练样本。根据 Skip-gram 模型的理念，中心词决定了它的相邻词，我们就可以根据这个训练样本定义出 Word2vec 模型的输入和输出，输入是样本的中心词，输出是所有的相邻词。

为了方便你理解，我再举一个例子。这里我们选取了“Embedding 技术对深度学习推荐系统的重要性”作为句子样本。首先，我们对它进行分词、去除停用词的过程，生成词序列，再选取大小为 3 的滑动窗口从头到尾依次滑动生成训练样本，然后我们把中心词当输入，边缘词做输出，就得到了训练 Word2vec 模型可用的训练样本。

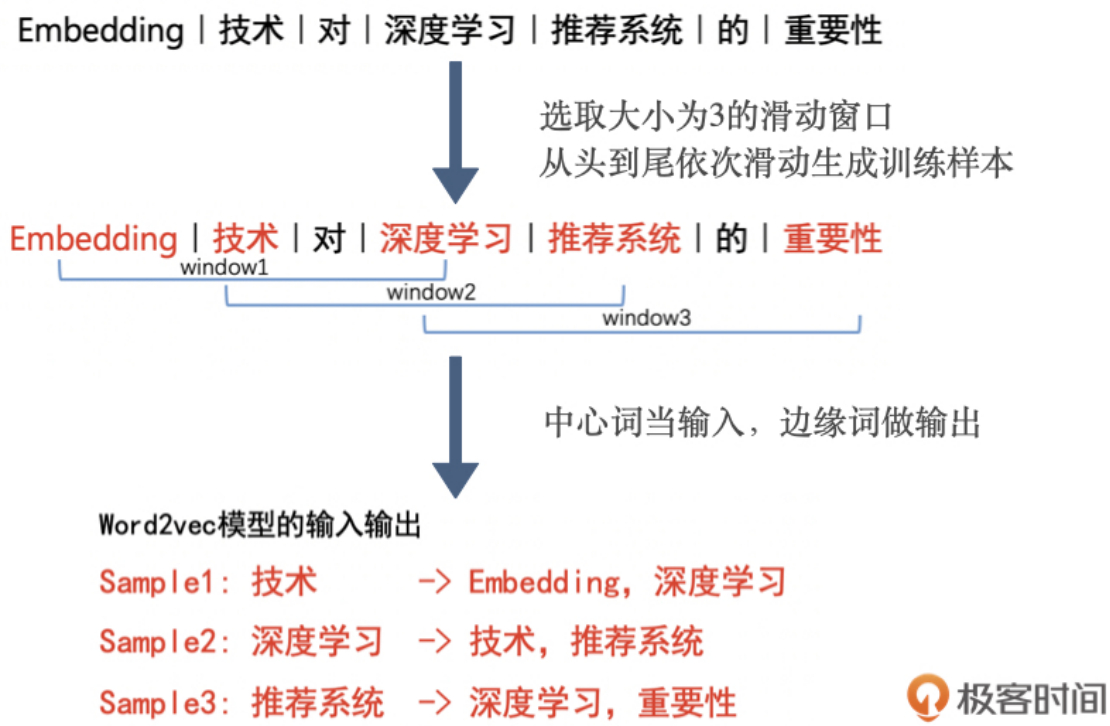

图4 生成Word2vec训练样本的例子

Word2vec 模型的结构是什么样的?

有了训练样本之后，我们最关心的当然是 Word2vec 这个模型的结构是什么样的。我相信，通过第 3 节课的学习，你已经掌握了神经网络的基础知识，那再理解 Word2vec 的结构就容易多了，它的结构本质上就是一个三层的神经网络（如图 5）。

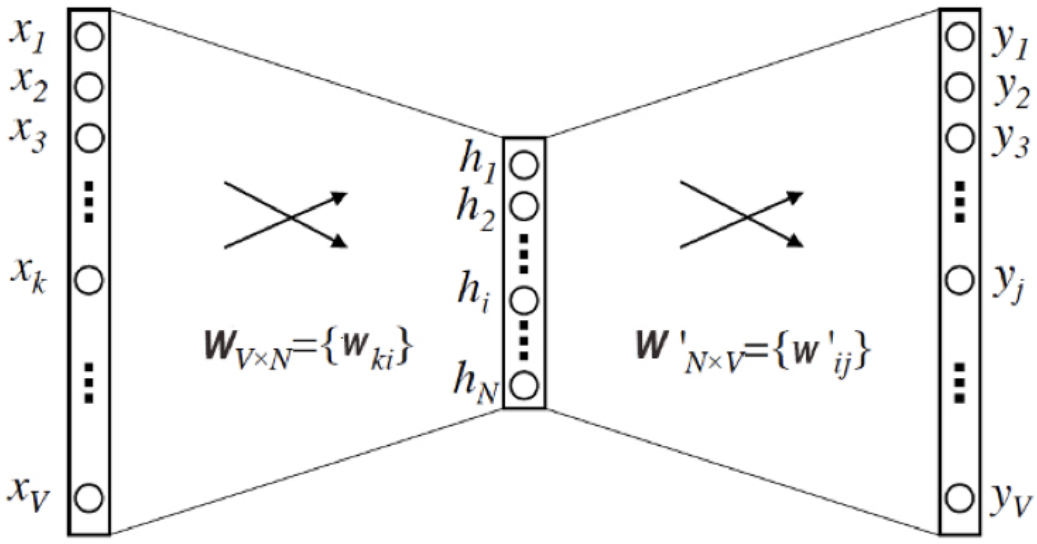

图5 Word2vec模型的结构

它的输入层和输出层的维度都是 V，这个 V 其实就是语料库词典的大小。假设语料库一共使用了 10000 个词，那么 V 就等于 10000。根据图 4 生成的训练样本，这里的输入向量自然就是由输入词转换而来的 One-hot 编码向量，输出向量则是由多个输出词转换而来的 Multi-hot 编码向量，显然，基于 Skip-gram 框架的 Word2vec 模型解决的是一个多分类问题。

隐层的维度是 N，N 的选择就需要一定的调参能力了，我们需要对模型的效果和模型的复杂度进行权衡，来决定最后 N 的取值，并且最终每个词的 Embedding 向量维度也由 N 来决定。

最后是激活函数的问题，这里我们需要注意的是，隐层神经元是没有激活函数的，或者说采用了输入即输出的恒等函数作为激活函数，而输出层神经元采用了 softmax 作为激活函数。

你可能会问为什么要这样设置 Word2vec 的神经网络，以及我们为什么要这样选择激活函数呢？因为这个神经网络其实是为了表达从输入向量到输出向量的这样的一个条件概率关系，我们看下面的式子：

$$p\left(w_O \mid w_I\right)=\frac{\exp\left(v'_{w_O} v_{w_I}\right)}{\sum_{i=1}^{V} \exp\left(v'_{w_i}{}^{\top} v_{w_I}\right)}$$

这个由输入词 WI 预测输出词 WO 的条件概率，其实就是 Word2vec 神经网络要表达的东西。我们通过极大似然的方法去最大化这个条件概率，就能够让相似的词的内积距离更接近，这就是我们希望 Word2vec 神经网络学到的。

当然，如果你对数学和机器学习的底层理论没那么感兴趣的话，也不用太深入了解这个公式的由来，因为现在大多数深度学习平台都把它们封装好了，你不需要去实现损失函数、梯度下降的细节，你只要大概清楚他们的概念就可以了。

如果你是一个理论派，其实 Word2vec 还有很多值得挖掘的东西，比如，为了节约训练时间，Word2vec 经常会采用负采样（Negative Sampling）或者分层 softmax（Hierarchical Softmax）的训练方法。关于这一点，我推荐你去阅读

Word2vec Parameter Learning Explained这篇文章，相信你会找到最详细和准确的解释。

怎样把词向量从 Word2vec 模型中提取出来？

在训练完 Word2vec 的神经网络之后，可能你还会有疑问，我们不是想得到每个词对应的 Embedding 向量嘛，这个 Embedding 在哪呢？其实，它就藏在输入层到隐层的权重矩阵 WVxN 中。我想看了下面的图你一下就明白了。

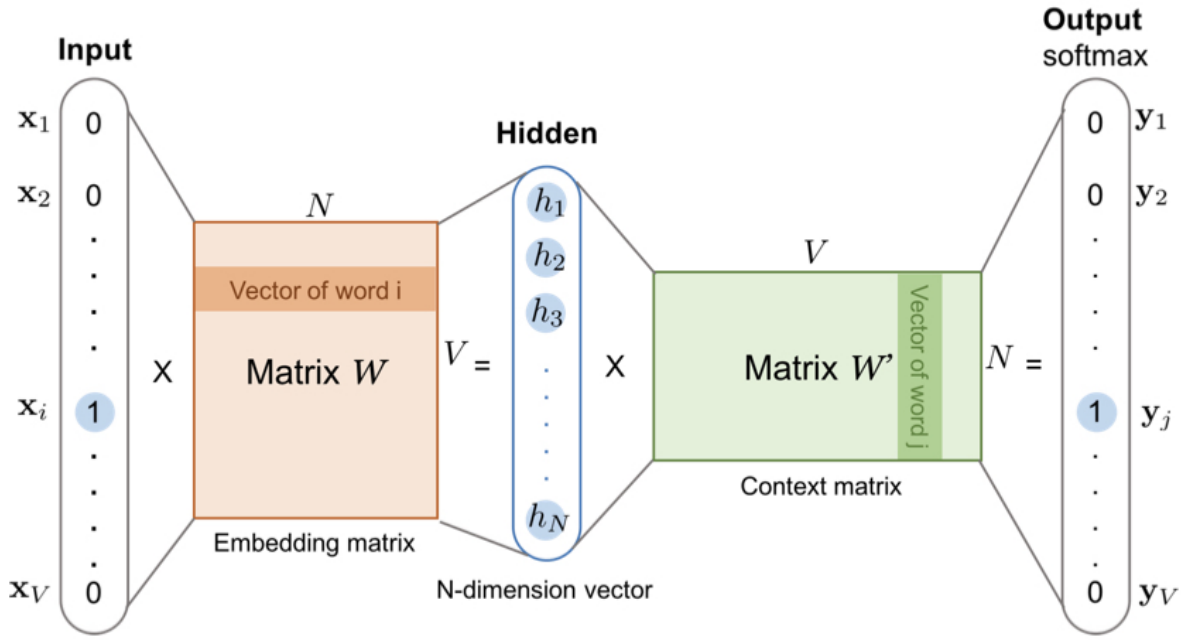

图6 词向量藏在Word2vec的权重矩阵中

你可以看到，输入向量矩阵 WVxN 的每一个行向量对应的就是我们要找的“词向量”。比如我们要找词典里第 i 个词对应的 Embedding，因为输入向量是采用 One-hot 编码的，所以输入向量的第 i 维就应该是 1，那么输入向量矩阵 WVxN 中第 i 行的行向量自然就是该词的 Embedding 啦。

细心的你可能也发现了，输出向量矩阵 W ‘也遵循这个道理，确实是这样的，但一般来说，我们还是习惯于使用输入向量矩阵作为词向量矩阵。

在实际的使用过程中，我们往往会把输入向量矩阵转换成词向量查找表（Lookup table，如图 7 所示）。例如，输入向量是 10000 个词组成的 one hot 向量，隐层维度是 300 维，那么输入层到隐层的权重矩阵为 10000x300 维。在转换为词向量 Lookup table 后，

每行的权重即成了对应词的 Embedding 向量。如果我们把这个查找表存储到线上的数据库中，就可以轻松地在推荐物品的过程中使用 Embedding 去计算相似性等重要的特征了。

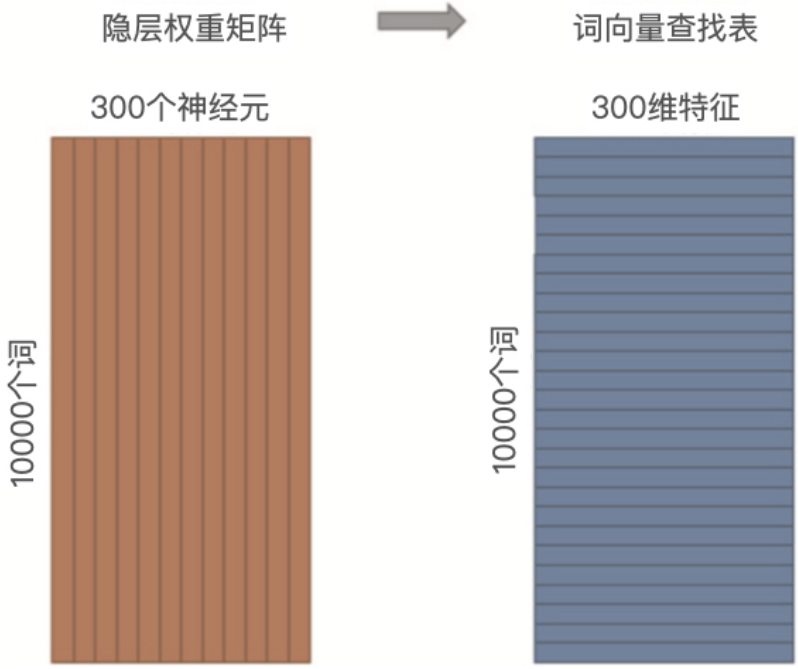

图7 Word2vec的Lookup table

Word2vec 对 Embedding 技术的奠基性意义

Word2vec 是由谷歌于 2013 年正式提出的，其实它并不完全是原创性的，学术界对词向量的研究可以追溯到 2003 年，甚至更早的时期。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，进而使 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。

从另一个角度来看，Word2vec 的研究中提出的模型结构、目标函数、负采样方法、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，熟练掌握本节课的内容是非常重要的。

Item2Vec：Word2vec 方法的推广

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。

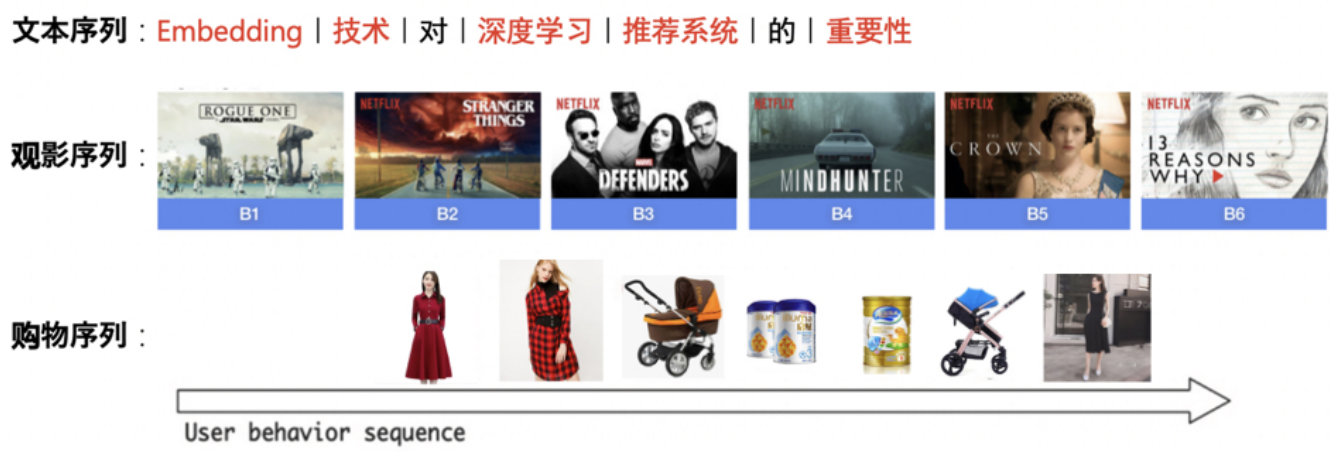

图8 不同场景下的序列数据

于是，微软于 2015 年提出了 Item2Vec 方法，它是对 Word2vec 方法的推广，使 Embedding 方法适用于几乎所有的序列数据。Item2Vec 模型的技术细节几乎和 Word2vec 完全一致，只要能够用序列数据的形式把我们要表达的对象表示出来，再把序列数据“喂”给 Word2vec 模型，我们就能够得到任意物品的 Embedding 了。

Item2vec 的提出对于推荐系统来说当然是至关重要的，因为它使得“万物皆 Embedding”成为了可能。对于推荐系统来说，Item2vec 可以利用物品的 Embedding 直接求得它们的相似性，或者作为重要的特征输入推荐模型进行训练，这些都有助于提升推荐系统的效果。

小结

这节课，我们一起学习了深度学习推荐系统中非常重要的知识点，Embedding。Embedding 就是用一个数值向量“表示”一个对象的方法。通过 Embedding，我们又引出了 Word2vec，Word2vec 是生成对“词”的向量表达的模型。其中，Word2vec 的训练样本是通过滑动窗口一一截取词组生成的。在训练完成后，模型输入向量矩阵的行向

量，就是我们要提取的词向量。最后，我们还学习了 Item2vec，它是 Word2vec 在任意序列数据上的推广。

我把这些重点的内容以表格的形式，总结了出来，方便你随时回顾。

知识点	关键描述
Embedding	Embedding就是用一个数值向量“表示”一个对象（Object）的方法
Word2vec	生成对“词”的向量表达的模型
Word2vec的样本生成方法	通过滑动窗口一一截取词组，把词组内的词转换成训练样本。
Word2vec模型的结构	三层神经网络
把词向量从Word2vec模型中提取的步骤	Word2vec模型中输入向量矩阵的行向量
Item2vec?	Word2vec在任意序列数据上的推广

极客时间

这节课，我们主要对序列数据进行了 Embedding 化，那如果是图结构的数据怎么办呢？另外，有没有什么好用的工具能实现 Embedding 技术呢？接下来的两节课，我就会一一讲解图结构数据的 Embedding 方法 Graph Embedding，并基于 Spark 对它们进行实现。

课后思考

在我们通过 Word2vec 训练得到词向量，或者通过 Item2vec 得到物品向量之后，我们应该用什么方法计算他们的相似性呢？你知道几种计算相似性的方法？

如果你身边的朋友正对 Embedding 技术感到疑惑，也欢迎你把这节课分享给 TA，我们下节课再见！

提建议

上一篇 05 | 特征处理：如何利用Spark解决特征处理问题?

下一篇 07 | Embedding进阶：如何利用图结构数据生成Graph Embedding?

精选留言 (14)

写留言

wolong
2020-10-14

老师您好，我这边有个问题。假如我们是做商品推荐，假如商品频繁上新，我们的物品库会是一个动态的，Embedding技术如何应对?

展开

作者回复: 非常好的冷启动问题，你有什么想法，想先听听你自己的思考。

2 9

Geek_3c29c3
2020-10-23

老师，想问一下，业界利用embedding召回时：
1、是用用户embedding与item embedding的相似性召回还是先计算用户之间的相似性TOPN，然后生成一个user-item矩阵，看看最相似的用户买的多的item得分就更高？；
2、业界用embedding召回如何评价优劣？数据集会划分训练集和验证集吗，来验证购买率，召回率等指标；如果划分，是按照时间划分还是按照用户来划分啊？

展开

作者回复: 1. 一般用前者
2.业界一般直接用线上AB test的结果来衡量embedding召回的优劣。离线指标可以做参考。如果做离线测试，一般采用时间划分，要避免引入未来信息。

2

张弛 Conor
2020-10-15

计算向量间相似度的常用方法：https://cloud.tencent.com/developer/article/1668762

作者回复: 非常好的文章，很全面了，推荐大家学习。

2

Dikiwi
2020-10-14

相似性一般用欧式距离，cosine距离;
线上快速召回一般有用ANN，比如LSH算法进行近似召回。

展开

作者回复: 赞，ANN的方法也是后续的课程内容。

2

杜军
2020-10-14

期待王老师的 Graph Embedding 讲解，能否配合一个 notebook 的示例就更完美了

2

wanlong
2020-10-20

老师您好，非常棒的文章，请教一个问题：
如果要往更深部分的学习，除了朝您说的算法细节，还有对应各个环节实现遇到的工程挑战，并设计相对的优化方向，这样算一个靠谱的掌握了吧？

作者回复: 是的，就是这两个方向，理论细节和实践经验。

特别是实践经验，在数据量大了之后，会遇到很多工程上的问题，需要不断的积累。

1

小眼无神霹雳霹雳猪
2020-10-20

老师你好！我想不太明白的地方是word2vec的输入层维度是词汇库的大小V。word2vec里词汇库的大小V是不怎么会变的。但如果是经常会加新物品进来的item2vec V会动态的变大。item2vec要随着V变大不断变动神经网络的结构并重新训练么？

展开

作者回复: 是的，V如果变化的话必须重新训练。除非采取其他item emb冷启动的方法。

1

张弛 Conor
2020-10-15

老师，有两个问题想请教一下：
1.为什么深度学习的结构特点不利于稀疏特征向量的处理呢？
2.既然输出向量是Multi-hot，那用softmax这种激活函数是否不太好呢？Softmax有输出相加和为一的限制，对于一对多的任务是不是不太合适呢？

展开

作者回复: 第一个问题非常好，我想你能够好好想一下梯度反向传播的过程，再回头看看这个问题。我相信你应该能得出自己的答案。

第二个也非常好，在训练的时候，确实要把最终的label做归一化，比如这样[0, 0.5, 0, 0.5, 0]。这是训练多分类模型的标准做法。另一种做法是把上一个样本拆成两个独立的onehot样本[0, 1, 0, 0, 0] 和 [0, 0, 0, 1, 0]，这样训练也可以，就不存在你说的信息丢失的问题。

2

1

WiFeng

2020-10-15

其实，它就藏在输入层到隐层的权重矩阵 WVxN 中。我想看了下面的图你一下就明白了。

这个愣是没明白。

作者回复: 关注图中橙色的部分，Embedding Matrix

2 1

傻

2020-10-14

如果是线上用到的实时特征，重新计算embedding的话，响应时间是否能满足要求呢?

作者回复: 不满足，embedding需要离线训练。除非是通过一些average pooling，sum pooling的方法在线处理，否则不可能实时生成。

4 1

大龄小学生

2020-10-22

老师，维度过高为什么不用降维，而用embedding

展开

作者回复: 本质上embedding就是降维的一种方式。比如经典的降维方法PCA，其实也就是用主成分分量来表示一个高维向量。

但是现在embedding的使用方式灵活多了，生成embedding的神经网络可以非常灵活，所以深度学习里面还是embedding用的更广泛一些。

1

书豪

2020-10-19

老师可以结合一个小小的案例实战来讲，这样就会更加直观一点！！！

作者回复: 实战在第8讲就会到来，不要着急

鱼_XueTr
2020-10-14

1.向量投影，即余弦相似度，2.计算模比较大小，各种距离，比如欧式距离，3.KNN

作者回复: 很全面了

pedro
2020-10-14

相似性的计算，实质是向量相似性的计算，大家都知道的有余弦相似性，街区距离等，复杂一点的有EMD，还有我也不知道😂

展开