# 【深度学习】深度学习在推荐系统中的应用有哪些?

2018-04-16 刑无刀

推荐系统三十六式 进入课程 >

讲述：黄洲君
时长 12:54 大小 5.91M

时至今日，深度学习已经不是一个新名词了，由于它的出现，计算机视觉、自然语言理解等领域的从业者都过上了好日子，错误率大幅度降低。

尤其是那些不断号称端到端的建模方式，让还在埋头于特征工程的推荐系统从业者们跃跃欲试，想赶紧引入深度学习大显身手。

经过这些年学界和业界的不断尝试，深度学习在推荐系统中已经有了很多成功的应用。

所以我在这个专栏里面理应本着实用落地的原则给你介绍一下，到底深度学习在推荐系统中有些什么应用，以及到底是怎么回事?

## 深度学习与推荐系统

深度学习也就是深度神经网络，并不是一个全新的概念，而是枯木逢春；所以它才能在计算力成本下降、效率提升、数据量陡增的今天得以焕发光彩，原来的浅层模型可以逐渐深入，挖掘出事物背后的更多规律和特征。

因此，深度学习的原理在这里并不做过多涉及，如果需要了解，你可以去专攻一下深度学习。

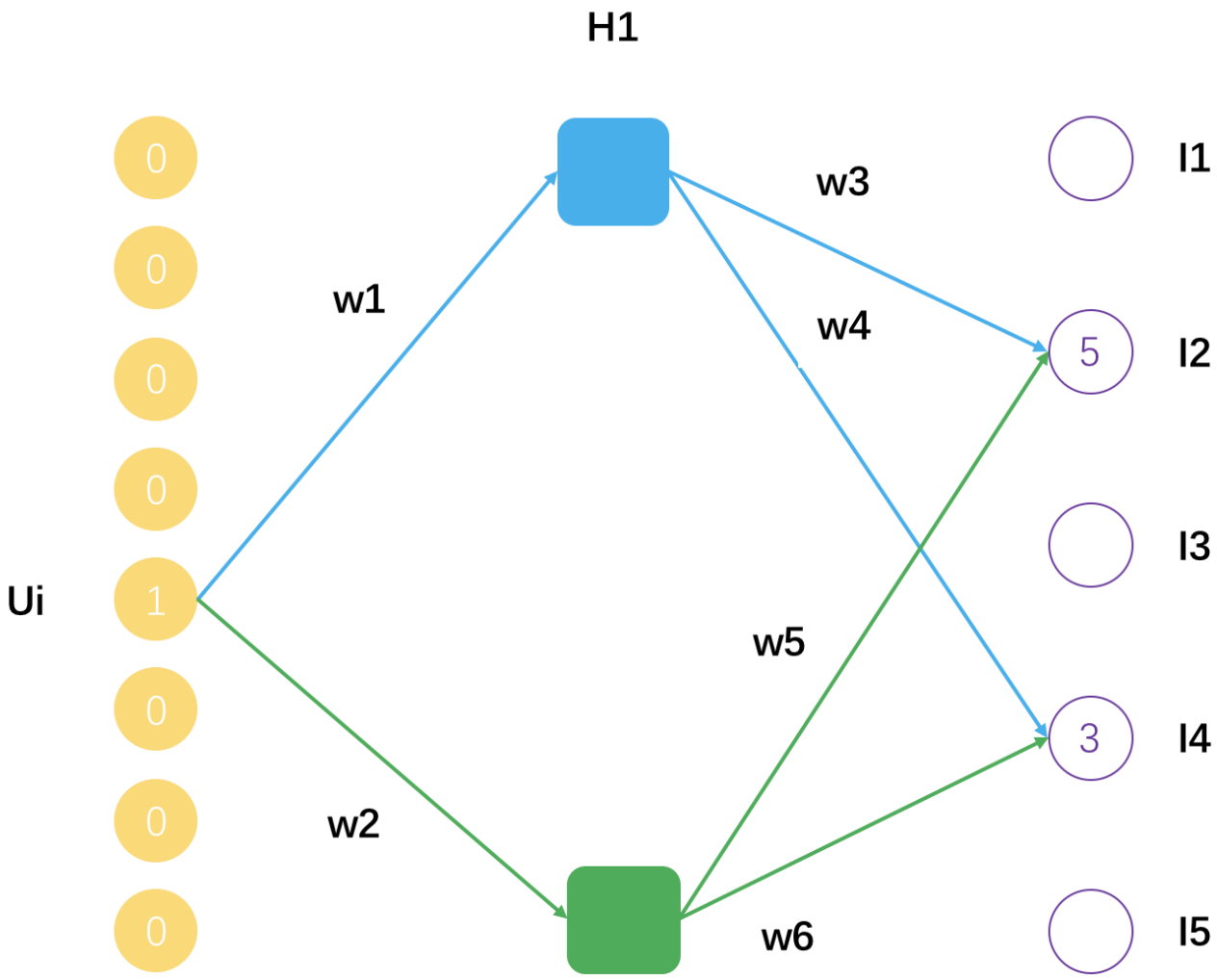
我在这里仅仅用简单的语言力图消除一些概念上的陌生感，在有了一些直观的认识后，直接进入到应用阶段，看看它可以帮助你做什么事。

你还记得矩阵分解吗？矩阵分解是把原来用户和物品之间的大矩阵，分解成了两个小矩阵相乘。这两个小矩阵小在哪？

原始的矩阵中，表示每个用户的向量是物品，表示每个物品的向量是用户，两者向量的维度都特别高不说，还特别稀疏，分解后用户向量和物品向量不但维度变得特别小，而且变稠密了。

业界还把这个稠密的向量叫做隐因子，意图直观说明它的物理意义：用户背后的偏好因子，物品背后的主题因子。

实际上，你完全可以把矩阵分解看成是一种浅层神经网络，只有一层，它的示意图如下。




这个示意图表示了一个用户 Ui，评分过的物品有 I2 和 I4，分解后的矩阵隐因子数量是 2，用户 Ui 的隐因子向量就是 [w1, w2]，物品 I2 的隐因子向量是 [w3, w5]，物品 I4 的隐因子向量是 [w4, w6]。

可以把矩阵分解看成是一个拥有一个隐藏层的神经网络，得到的隐因子向量就是神经网络的连接权重参数。

在前面的专栏中，我第一次提到深度学习时，还建议你把逻辑回归看成一个没有隐藏层的神经网络。因此，深度学习，也就是深度神经网络并不是那么神秘，只是深。这个“深”代表了事物的某些本质属性。

这种对本质属性的挖掘，有两个好处。

1. 可以更加高效且真实地反映出事物本身的样子。对比一下，一张图片用原始的像素点表示，不但占用空间大，而且还不能反应图片更高级的特征，如线条、明暗、色彩，而后

者则可以通过一系列的卷积网络学习而得。
2. 可以更加高效真实地反映出用户和物品之间的连接。对比一下，以用户历史点击过的物品作为向量表示用户兴趣；用这些物品背后隐藏的因子表示用户兴趣，显然后者更高效更真实，因为它还考虑了物品本身的相似性，这些信息都压缩到隐因子向量中了，同时再得到物品的隐因子向量，就可以更加直接平滑地算出用户对物品的偏好程度。

这两个好处，正是深度学习可以帮助推荐系统的地方。第一个叫做 Embedding，就是嵌入，第二个叫做 Predicting，就是预测。

其实两者我在前面的内容都已经有涉及了，矩阵分解得到的隐因子向量就是一种 Embedding，Word2vec 也是一种 Embedding，Wide&Deep 则是用来预测的。关于第二种，具体来说有几个方向：深度神经网络的 CTR 预估，深度协同过滤，对时间序列的深度模型。

下面逐一带你认识。首先就是深度学习的第一种应用。

## 各种 2vec

你还记得在内容推荐那一章里，我跟你提到过，对内容的挖掘怎么深入都不为过，越深入越好，很多时候甚至优于对排序模型的优化。

那里提到了 Word2vec，用于学习词嵌入向量。当把一个词表示成一个稠密的向量后，就可以计算词的相似度，进而可以计算句子的相似度，也可以直接把这个稠密向量作为特征输入给高级的预测模型。

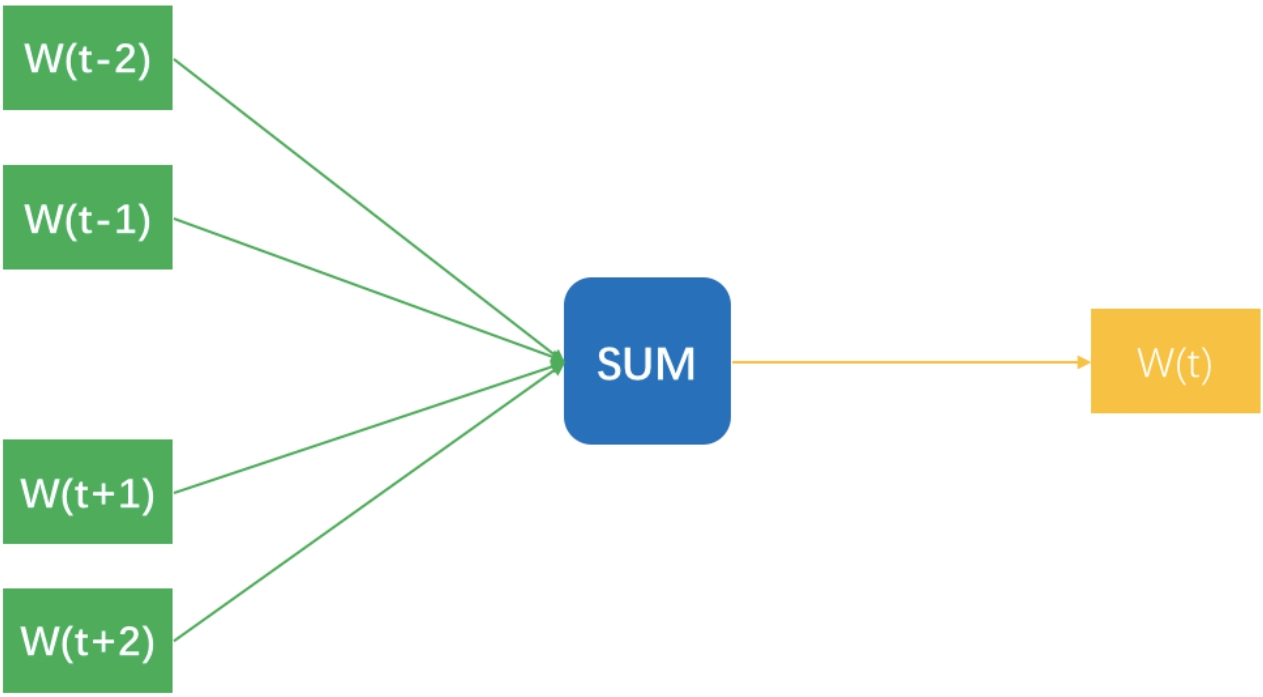
于是，这个 2vec 的思想，就被发扬光大了。首先还是在文本领域，从 Word2vec 到 Sentence2vec，再到 Doc2vec。其实思想都类似甚至会让你觉得有上当受骗的错觉。

简单介绍一下 Word2vec。你知道，Word2Vec 最终是每个词都得到一个稠密向量，十分类似矩阵分解得到的隐因子向量，得到这个向量有两个训练方法。

先说第一个方法，想象你拿着一个滑动窗口，在一篇文档中从左往右滑动，每一次都有 N 个词在这个窗口内，每移动一下，产生 N-1 条样本。

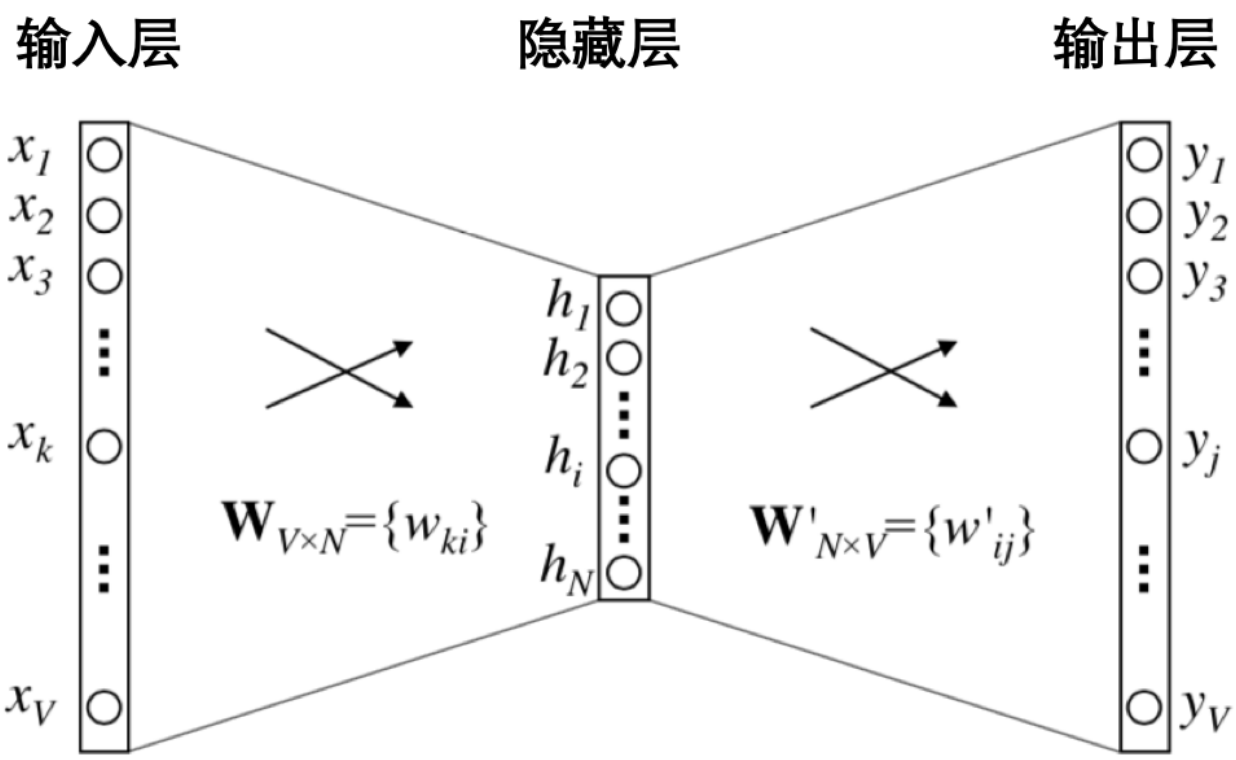
每条样本都是用窗口内一个词去预测窗口正中央那个词，明明窗口内是 N 个词，为什么只有 N-1 条样本呢？因为正中央那个词不用预测它本身啊。这 N-1 条样本的输入特征是词的

嵌入向量，预测标签是窗口那个词。示意图如下所示。




图中把 N-1 个样本放在一起示意的，无法看出隐藏层，实际上，输入时每个词可以用 One-hot 方式表示成一个向量，这个向量长度是整个词表的长度，并且只有当前词位置是 1，其他都是 0。

隐藏层的神经元个数就是最终得到嵌入向量的维度数，最终得到的嵌入向量元素值，实际上就是输入层和隐藏层的连接权重。示意图如下。



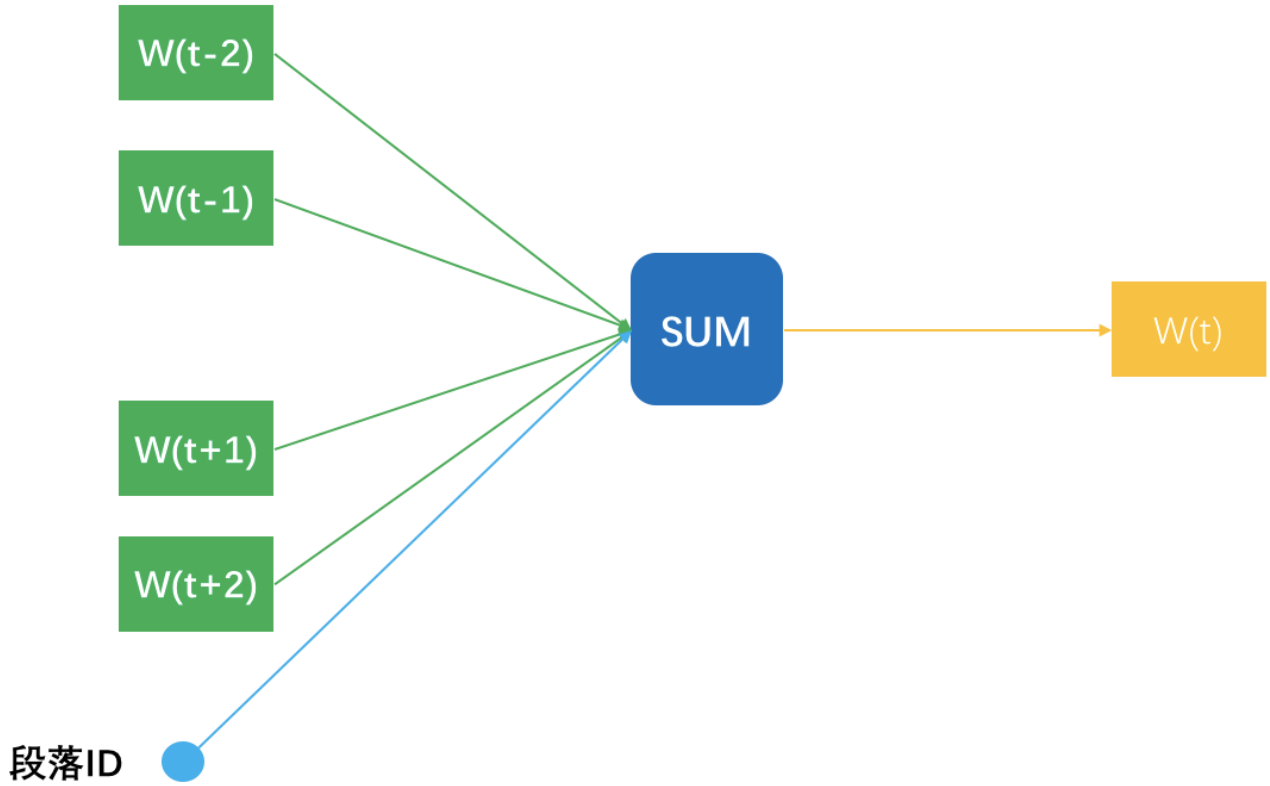

至于 Word2vec 的第二种训练方法，则是把上述的 N-1 条样本颠倒顺序，用窗口中央的词预测周围的词，只是把输入和输出换个位置，一样可以训练得到嵌入向量。

这里注意，看上去 Word2vec 是构造了一个监督学习任务，但实际上并不是为了得到一个预测模型，在实际中用词预测词，是为了得到词的嵌入向量，Embedding 本身就是目的。

我们沿着 Word2vec 这种学习嵌入向量的思路想，既然词可以表示成一个稠密向量干这干那，那不如来个 Sentence2vec，把一个句子表示成一个嵌入向量，通常是把其包含的词嵌入向量加起来就完事了。

而 Doc2vec 则略微一点点不同，说明一点，多个句子构成一个段落，所以这里的 Doc 其实就是段落。Doc2vec 在窗口滑动过程中构建 N-1 条样本时，还增加一条样本，就是段落 ID 预测中央那个词，相当于窗口滑动一次得到 N 条样本。

一个段落中有多少个滑动窗口，就得到多少条关于段落 ID 的样本，相当于这个段落中，段落 ID 在共享嵌入向量。段落 ID 像个特殊的词一样，也得到属于自己的嵌入向量，也就是 Doc2vec。




理解了 Word2vec 之后，它在推荐系统中的应用就有举一反三的应用。第一个就是 Product2vec，你看这名字就知道我要干嘛了，对，就是给商品学习一个嵌入向量。

这简直就是照着词嵌入的做法来，把用户按照时间先后顺序加入到购物车的商品，看成一个一个的词，一个购物车中所有的商品就是一个文档；于是照猫画虎学出每个商品的嵌入向量，用于去做相关物品的推荐，或者作为基础特征加入到其他推荐排序模型中使用。

类似的，如果是应用商场的 App 推荐，也可以依计行事，把用户的下载序列看成文档，学习每个 App 的嵌入向量。

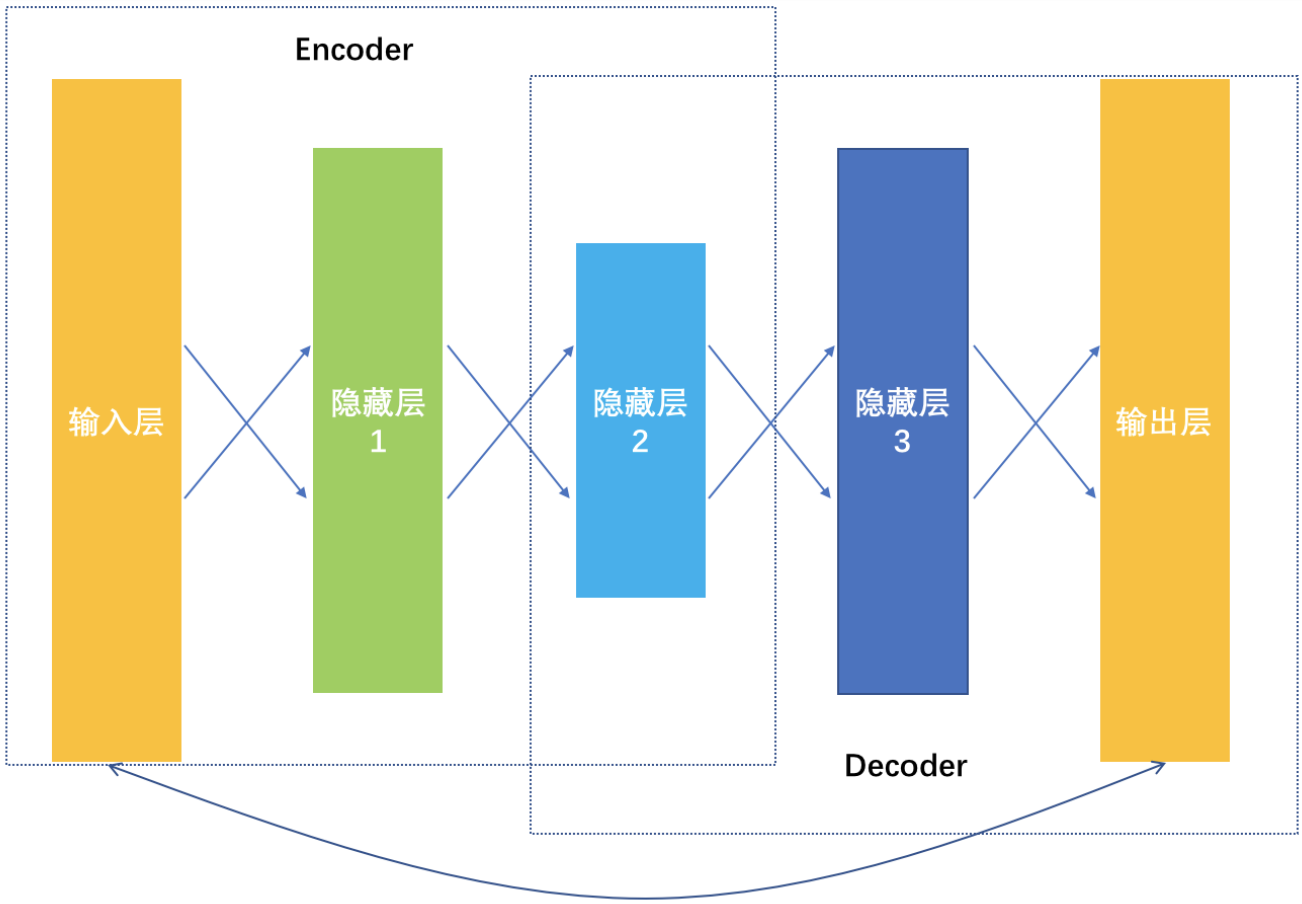
虽然，这个嵌入学习得到的结果，样子和矩阵分解得到隐因子向量一样，但是机制不同，可以两者都有，拼接成一个更大的稠密向量去做你喜欢做的事，比如 CTR 预估。

且慢，各种 2vec 的做法其实还不算深度学习，毕竟隐藏层才一层而已。如果要用更深的模型学习嵌入向量，就是深度学习中的 AutoEncoder。

它是一种输入和输出一模一样的神经网络，这个神经网络就一个目的，更加清楚地认识自己，在这个优化目标指导下，学到的网络连接权重都是不同的嵌入向量，所以也叫做

AutoEncoder，自动编码器。

从输入数据逐层降维，相当于是一个对原始数据的编码过程，到最低维度那一层后开始逐层增加神经元，相当于是一个解码过程，解码输出要和原始数据越接近越好，相当于在大幅度压缩原始特征空间的前提下，压缩损失越小越好。




以上是深度学习如何通过学习更好地表达事物特征，帮助推荐系统取得更好效果的做法。

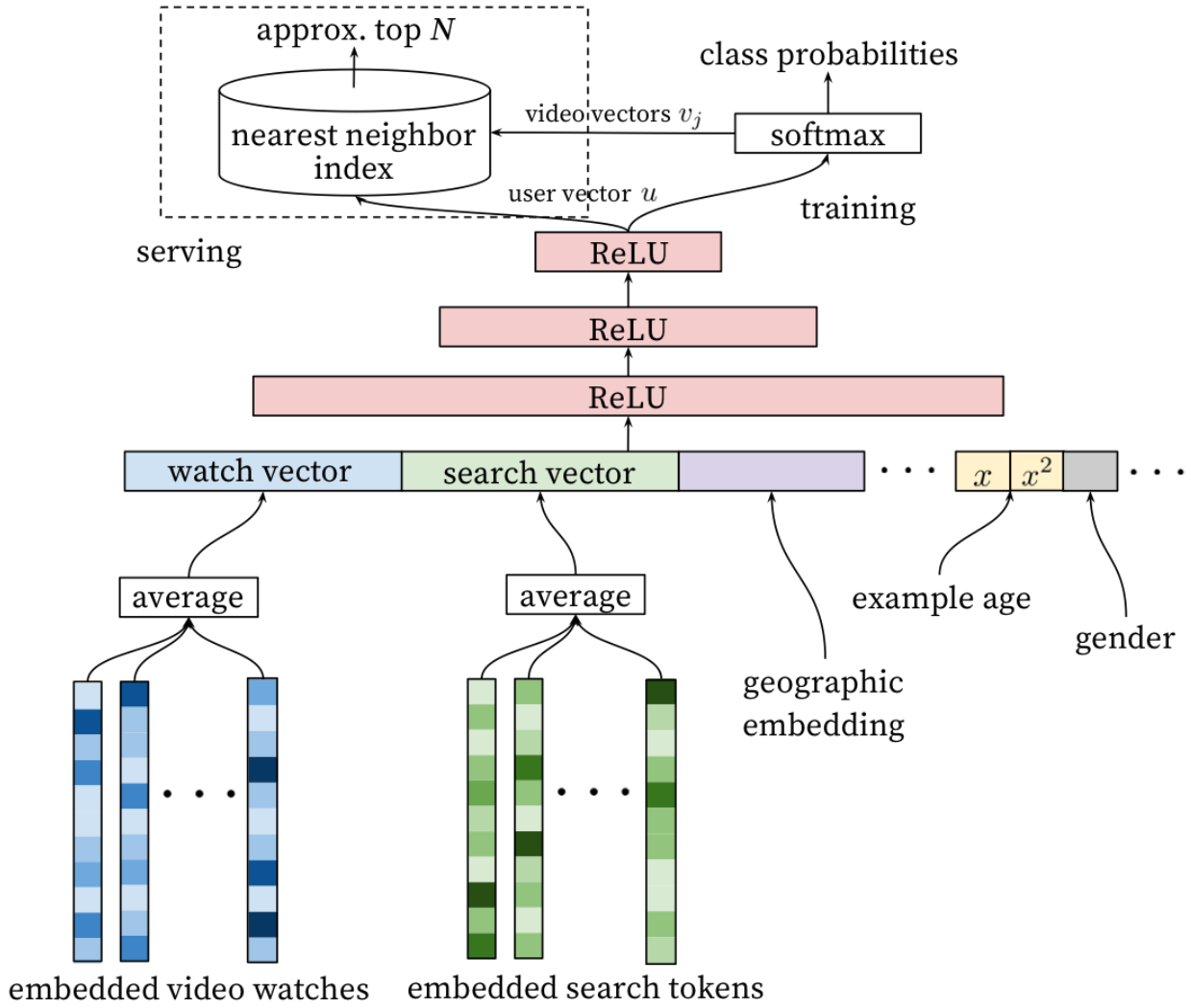
## YouTube 视频推荐

以 YouTube 为例，它们在自己的推荐系统大量用到了深度学习，用于推荐更好的视频给用户。我来给你仔细描述一下，具体到视频推荐场景，深度学习可以在哪些地方用到。

首先，Youtube 把推荐的预测任务看成是一个多分类，这个和之前常规的推荐系统要么预测行为要么预测评分的做法不太一样，而是把候选物品当成多个类别，预测用户下一个会观看哪个视频。

$$P(w_t = i|U, C) = \frac{e^{v_i u}}{\sum_{j \in V} e^{v_j u}}$$

这个公式中 U 是用户 C 是场景，输入是视频的嵌入向量和用户的嵌入向量。这里就涉及了先要使用深度神经网络，从用户历史反馈行为和场景信息中学习物品和用户的嵌入向量。整个推荐排序模型示意图如下。




看这个推荐模型示意图，就可以看到深度学习应用在了哪些地方。

1. 根据观看历史把视频变成了嵌入向量，然后平均后作为输入特征之一，这个和前面的 Product2vec 的思路一致，把观看历史看成文档，观看的视频看成词。
2. 搜索 Query 也变成了嵌入向量，平均之后作为输入特征之二。
3. 人口统计学信息统统都嵌入了。
4. 还加入视频的年龄信息，也就是在预测时，视频上传多久了。
5. 所有这些不同的嵌入向量拼接成一个大的输入向量，经过深度神经网络，在输出层以 Softmax 作为输出函数，预测下一个观看视频。

在模型训练时，以 Softmax 作为输出层，但是在实际线上预测服务时，由于模型关心相对顺序，所以并不需要真的去计算 Softmax，而是拿着用户的特征向量做近似的近邻搜索，只生成最相近的一些推荐结果。

整个推荐系统非常好理解，也比较好落地，所有的模型都可以通过 TensorFlow 快速实现。

## 总结

通过观察 YouTube 的推荐系统中所用到的深度学习来看，在排序方面，深度神经网络已经崭露头角，包括前面讲融合模型时，专门讲到的 Wide&Deep 模型，也是深度学习在排序方面的贡献。

除此之外，深度学习更多发挥作用的地方是特征表达上，各种嵌入技术得以让物品、用户、关系等对象的特征化有更好的输出。

今天主要介绍了深度学习在推荐系统可以发挥哪些作用，同时以 YouTube 为例，介绍了国际大厂在这方面的落地情况。

从 YouTube 的推荐系统可以看出，深度学习主要贡献在于特征表达学习和排序模型上。当然，深度学习纵有千般好，如果你的数据量少得可怜，那么你也只能过过眼瘾，很难真正落地执行。

给你留一个思考题，我们一起交流。为什么 YouTube 排序模型训练时规规矩矩按照 Softmax 输出，以交叉熵作为目标函数，但在线上运行时，却可以按近邻搜索去近似，背后有什么考虑？欢迎留言一起讨论。


© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 【MAB问题】如何将Bandit算法与协同过滤结合使用

下一篇 【深度学习】用RNN构建个性化音乐播单

## 精选留言 (9)

写留言

**异尘** 3
2018-05-14

这里的模型是用作召回的，用softmax的思路同word2vec的预测部分类似，最终目的是得出embeded向量，用过embeded向量可以用很多模糊近邻算法快速召回，后续再应用wide&deep排序

展开

**jacket** 2
2018-04-18

老师，看前面介绍BPR排序模型的时候，我觉得矩阵因子分解的方法，可以预测出同个用户对不同商品的评分，这是同个用户的评分标准，根据评分直接就可以拍个顺序了。这种方法跟BPR比较的话，孰好孰坏那?

展开

**林彦** 2

2018-04-17

我的理解softmax已经线下预测出用户最有可能看的视频，线上预测如另一个评论所述为了从数百万个视频中计算最可能的N个类（视频），用softmax进行评分或校验计算成本比较高，无法满足线上的延时要求。我们假定物以类聚，找出和最可能的视频相近的N个邻居的效果和用softmax验证每个结果并筛选出最可能的N个是比较相近的。这里看原文的是选出的N个(数百个)视频的排序过程中引入了新的特征，然后用一个和候选集训练模型类似...

展开

**刘大猫** 2

2018-04-16

老师介绍的这部分是matching部分，不是很关注具体的排序，计算softmax成本太大，所以召回的时候用近邻检索，提高效率。

**jt120** 1

2018-04-16

公式解析有问题，youtube的那个实现哪里还能找到资料？

展开

作者回复: chenkaijiang001@lianjia.com，你给我联系，我发给你。

**shangqiu86**

2019-05-05

YouTube那篇论文我也看过，也看过一些对该论文解读的部分，感觉论文的大部分介绍的还是特征的选择，希望老师以后可以单独开一些讲解最新最热paper的课，我会继续支持的。

展开

**benying**

2018-11-04

youtude做了个user2vec

展开



**科技狗**

2018-04-16

请问老师，YouTube 的ranking 部分，如何对观看时长预测exp(wx+b)和点击概率加权LR同时输出的?

**jt120**

2018-04-16

越相似熵越小?

展开