Embedding 讨论

leolinuxer

July 16, 2020

Contents

1 Embedding 简介 [1]

1.1 什么是 embedding

简单来说，embedding 就是用一个低维的向量表示一个物体，可以是一个词，或是一个商品，或是一个电影等等。这个 embedding 向量的性质是能使距离相近的向量对应的物体有相近的含义，比如 Embedding(复仇者联盟) 和 Embedding(钢铁侠) 之间的距离就会很接近，但 Embedding(复仇者联盟) 和 Embedding(乱世佳人) 的距离就会远一些。

除此之外 Embedding 甚至还具有数学运算的关系，比如 Embedding（马德里）-Embedding（西班牙）+Embedding(法国) Embedding(巴黎)

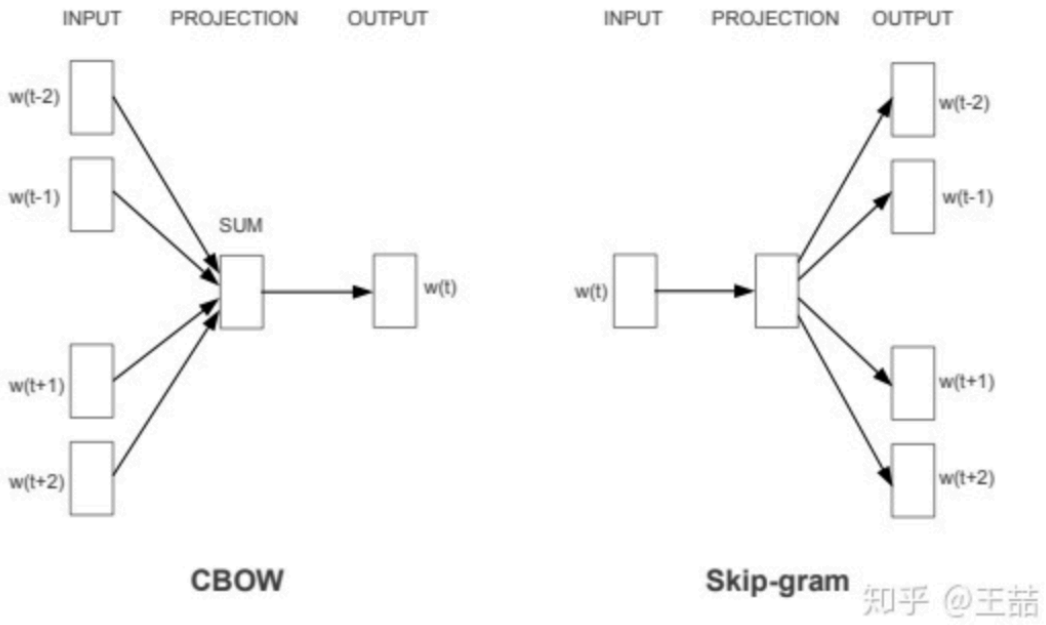
言归正传，Embedding 能够用低维向量对物体进行编码还能保留其含义的特点非常适合深度学习。在传统机器学习模型构建过程中，我们经常使用 one hot encoding 对离散特征，特别是 id 类特征进行编码，但由于 one hot encoding 的维度等于物体的总数，比如阿里的商品 one hot encoding 的维度就至

少是千万量级的。这样的编码方式对于商品来说是极端稀疏的，甚至用 multi hot encoding 对用户浏览历史的编码也会是一个非常稀疏的向量。而深度学习的特点以及工程方面的原因使其不利于稀疏特征向量的处理。因此如果能把物体编码为一个低维稠密向量再喂给 DNN，自然是一个高效的基本操作。

1.2 使 embedding 空前流行的 word2vec

对 word 的 vector 表达的研究早已有之，但让 embedding 方法空前流行，我们还是要归功于 google 的 word2vec。我们简单讲一下 word2vec 的原理，这对我们之后理解 AirBnB 对 loss function 的改进至关重要。

既然我们要训练一个对 word 的语义表达，那么训练样本显然是一个句子的集合。假设其中一个长度为 T 的句子为 $w_1, w_2, \cdots, w_T$ 。这时我们假定每个词都跟其相邻的词的关系最密切，换句话说每个词都是由相邻的词决定的（CBOW 模型的动机），或者每个词都决定了相邻的词（Skip-gram 模型的动机）。如下图，CBOW 的输入是 w_t 周边的词，预测的输出是 w_t，而 Skip-gram 则反之，经验上讲 Skip-gram 的效果好一点，所以本文从 Skip-gram 模型出发讲解模型细节。



word2vec的两种模型结构CBOW和Skip-gram

那么为了产生模型的正样本，我们选一个长度为 $2c+1$（目标词前后各选 c 个词）的滑动窗口，从句子左边滑倒右边，每滑一次，窗口中的词就形成了我们的一个正样本。

有了训练样本之后我们就可以着手定义优化目标了，既然每个词 w_t 都决定了相邻词 w_{t+j}，基于极大似然，我们希望所有样本的条件概率 $p(w_{t+j}|w_t)$ 之积最大，这里我们使用 log probability。我们的目标函数有了：

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

接下来的问题是怎么定义 $p(w_{t+j}|w_t)$ ，作为一个多分类问题，最简单最直接的方法当然是直接用

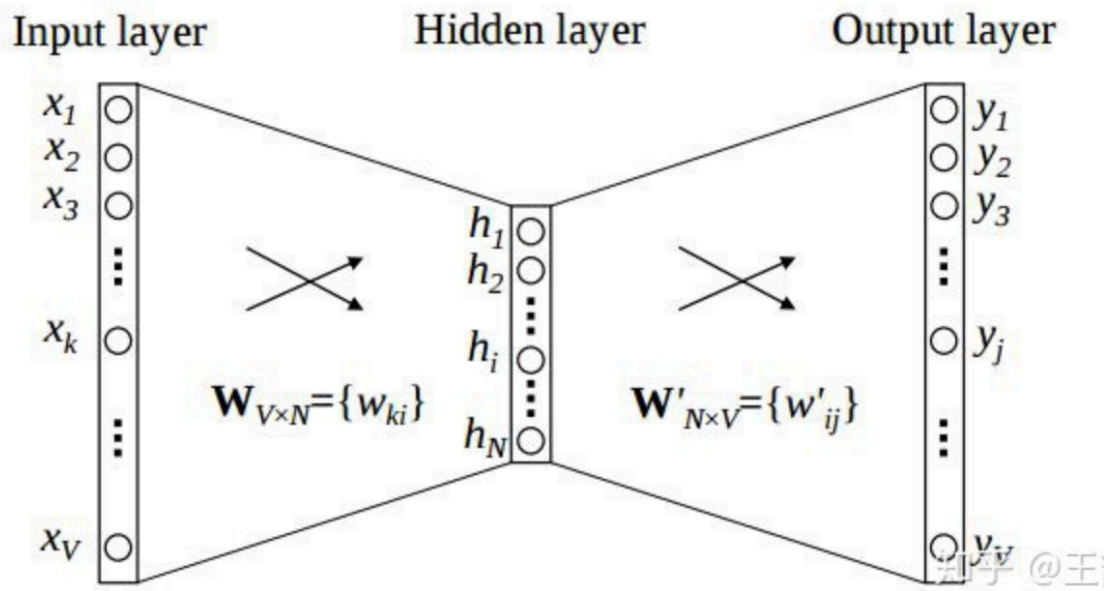
softmax 函数，我们又希望用向量 w_t 表示每个词 w，用词之间的距离 $v_i^T v_i$ 表示语义的接近程度，那么我们的条件概率的定义就可以很直观的写出。

$$P(w_O|w_I) = \frac{\exp({v'}_{w_O}^T v_{w_I})}{\sum_{w=1}^{W} \exp({v'}_{w_O}^T v_{w_I})}$$

看到上面的条件概率公式，很多同学可能会习惯性的忽略一个事实，就是

我们用 w_t 去预测 w_{t+j} ，但其实这二者的向量表达并不在一个向量空间内。

就像上面的条件概率公式写的一样，v'_w 和 v_w 分别是词 w 的输出向量表达和输入向量表达。那什么是输入向量表达和输出向量表达呢？我们画一个 word2vec 的神经网络架构图就明白了。

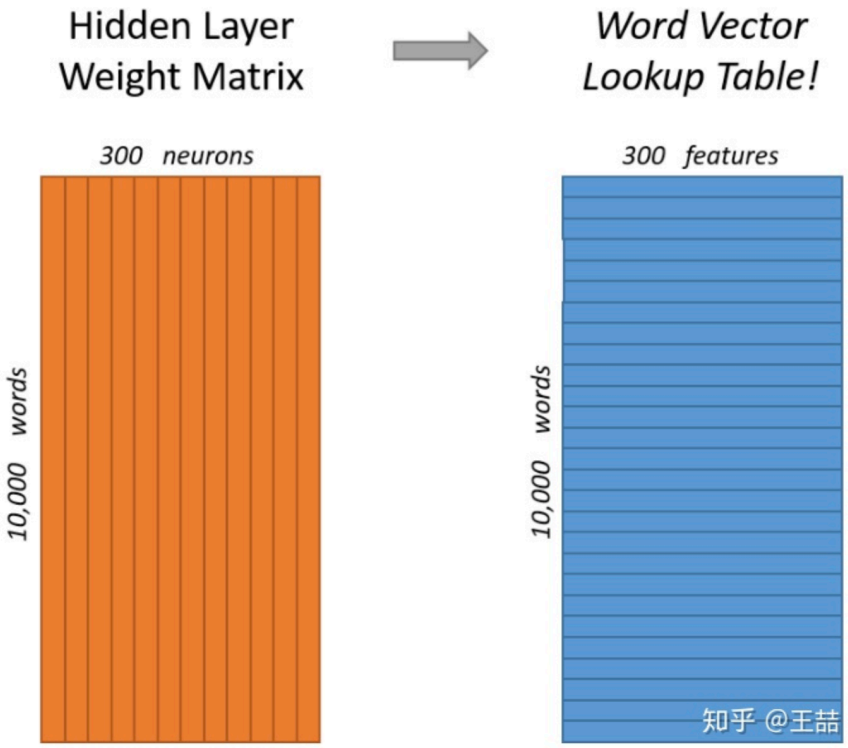


word2vec的算法架构

根据 $p(w_{t+j}|w_t)$ 的定义，我们可以把两个 vector 的乘积再套上一个 softmax 的形式转换成上面的神经网络架构（需要非常注意的一点是 hidden layer 的激活函数，大家要思考一下，到底是 sigmoid 函数还是普通的线性函数，为什么?）。在训练过程中我们就可以通过梯度下降的方式求解模型参数了。那么上文所说的输入向量表达就是 input layer 到 hidden layer 的权重矩阵 $W_{V\times N}$ ，而输出向量表达就是 hidden layer 到 output layer 的权重矩阵 $W'_{N\times V}$ 。

那么到底什么是我们通常意义上所说的词向量 v_w 呢？

其实就是我们上面所说的输入向量矩阵 $W_{V\times N}$ 中每一行对应的权重向量。于是这个权重矩阵自然转换成了 word2vec 的 lookup table。



当然在训练 word2vec 的过程中还有很多工程技巧，比如用 negative sampling 或 Hierarchical Softmax 减少词汇空间过大带来的计算量，对高频词汇进行降采样避免对于这些低信息词汇的无谓计算等。在具体实现的时候最好参考 Google 的原文 Distributed Representations of Words and Phrases and their Compositionality

1.3 从 word2vec 到 item2vec

在 word2vec 诞生之后，embedding 的思想迅速从 NLP 领域扩散到几乎所有机器学习的领域，我们既然可以对一个序列中的词进行 embedding，那自然可以对用户购买序列中的一个商品，用户观看序列中的一个电影进行 embedding。而广告、推荐、搜索等领域用户数据的稀疏性几乎必然要求在构建 DNN 之前对 user 和 item 进行 embedding 后才能进行有效的训练。

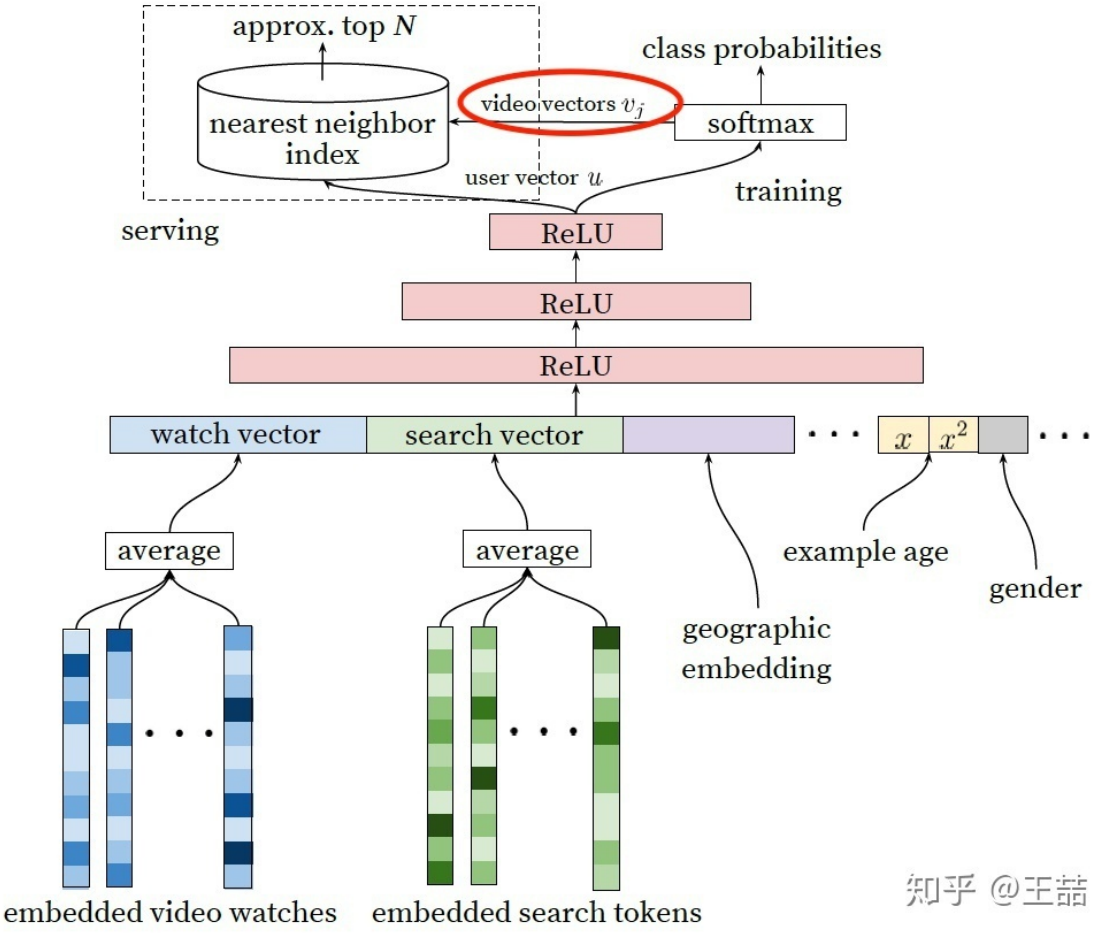
具体来讲，如果 item 存在于一个序列中，item2vec 的方法与 word2vec 没有任何区别。而如果我们摒弃序列中 item 的空间关系，在原来的目标函数基础上，自然是不存在时间窗口的概念了，取而代之的是 item set 中两两之间的条件概率：

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

具体可以参考 item2vec 的原文 Item2Vec:Neural Item Embedding for Collaborative Filtering

但 embedding 的应用又远不止于此，事实上，由于我们也可以把输出矩阵的列向量当作 item embedding，这大大解放了我们可以用复杂网络生成 embedding 的能力。读过我专栏上一篇文章 YouTube 深度学习推荐系统的十大工程问题的同学肯定知道，YouTube 在 serve 其 candidate generation model 的时候，只将最后 softmax 层的输出矩阵的列向量当作 item embedding vector，而将 softmax 之

前一层的值当作 user embedding vector。在线上 serving 时不用部署整个模型，而是只存储 user vector 和 item vector，再用最近邻索引进行快速搜索，这无疑是非常实用的 embedding 工程经验，也证明了我们可以用复杂网络生成 user 和 item 的 embedding。



KDD 2018 best paper Real-time Personalization using Embeddings for Search Ranking at Airbnb 也介绍了 Airbnb 的 embedding 最佳实践。

2 Embedding 在深度推荐系统中的 3 大应用方向 [2]

在深度学习推荐系统中，Embedding 有三个主要的应用方向：

- 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；
- 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；
- 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或计算广告系统的召回层或者召回方法之一；

2.1 深度学习网络中的 Embedding 层

由于高维稀疏特征向量天然不适合多层复杂神经网络的训练，因此如果使用深度学习模型处理高维稀疏特征向量，几乎都会在输入层到全连接层之间加入 Embedding 层完成高维稀疏特征向量到低维稠

密特征向量的转换。典型的例子是微软的 Deep Crossing 模型和 Google 的 Wide&Deep 模型的深度部分。

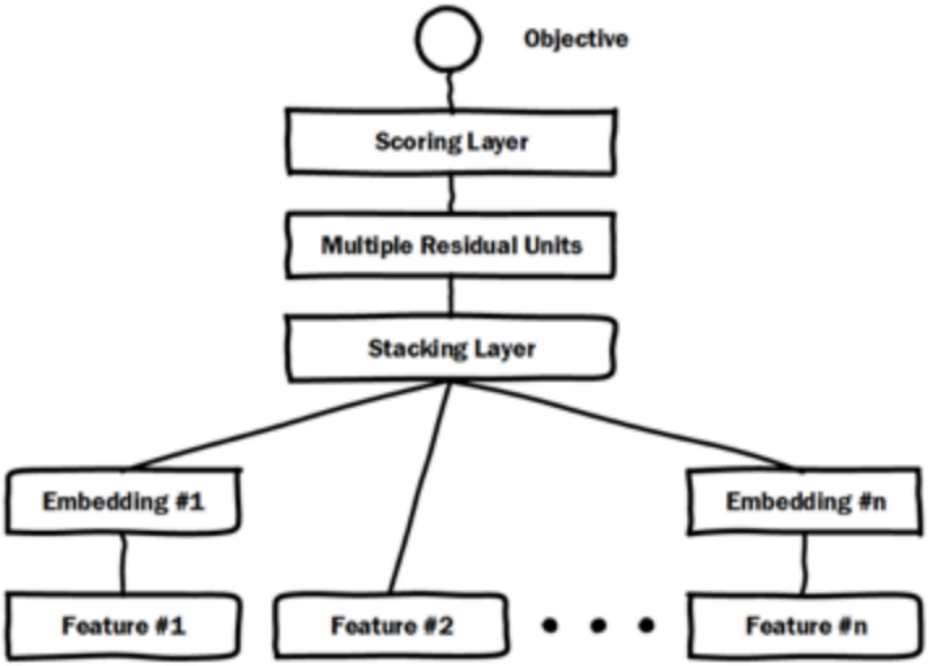
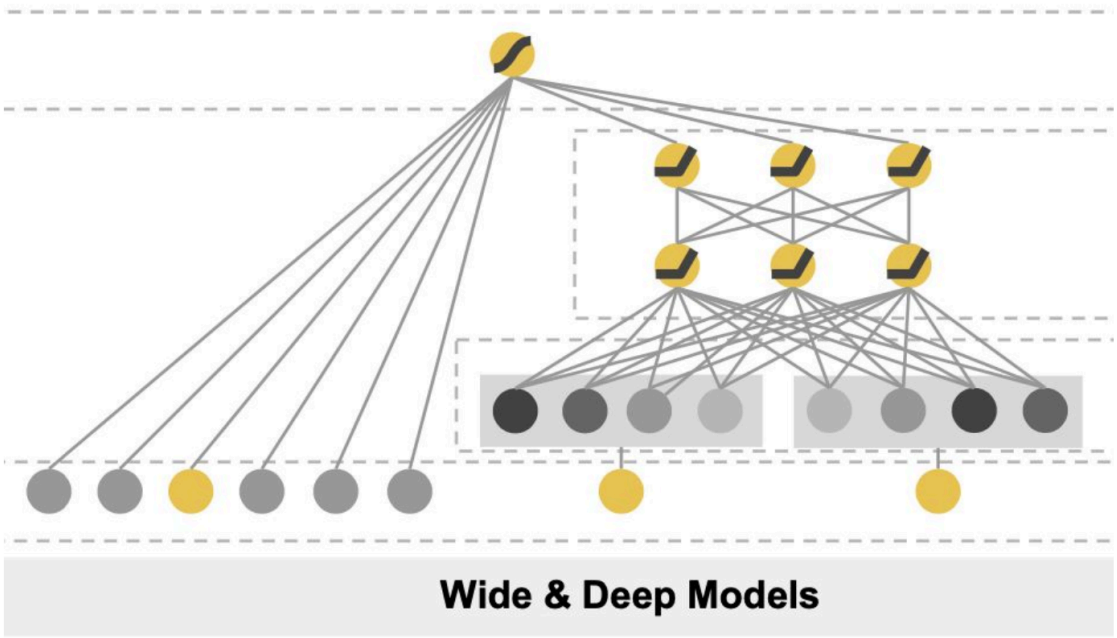


图1 微软Deep Crossing模型



Wide&Deep模型示意图

图 1 中可以清晰的看到 Deep Crossing 模型中的 Embedding 层将每一个 Feature 转换成稠密向量，图 2Wide&Deep 模型中 Deep 部分的 Dense Embeddings 层同样将稀疏特征向量进行转换。广义来说，Embedding 层的结构可以比较复杂，只要完成高维向量的降维就可以了，但一般为了节省训练时间，深

度神经网络中的 Embedding 层是一个高维向量向低维向量的直接映射（如图 3）。

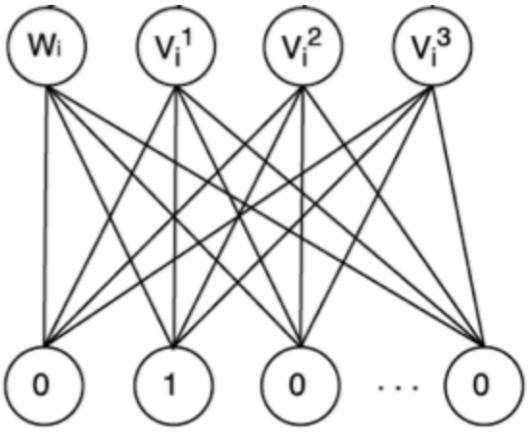


图3 稀疏one hot向量向稠密向量的映射

一般来说，推荐系统的输入向量中包含大量稀疏的 one hot 特征，图 3 展示了典型的稀疏向量向稠密 embedding 向量的最简单的 embedding 层结构。

用矩阵的形式表达 Embedding 层，本质上是求解一个 m（输入高维稀疏向量的维度）x n（输出稠密向量的维度）维的权重矩阵的过程。如果输入向量是 one-hot 特征向量的话，权重矩阵中的列向量即为相应维度 one-hot 特征的 embedding 向量。

将 Embedding 层与整个深度学习网络整合后一同进行训练是理论上最优的选择，因为上层梯度可以直接反向传播到输入层，模型整体是自洽和统一的。但这样做的缺点同样显而易见的，由于 Embedding 层输入向量的维度甚大，Embedding 层的加入会拖慢整个神经网络的收敛速度。

> 这里可以做一个简单的计算。假设输入层维度是 100,000，embedding 输出维度是 32，上层再加 5 层 32 维的全连接层，最后输出层维度是 10，那么输出层到 embedding 层的参数数量是 32*100,000= 3,200,000，其余所有层的参数总数是（32*32）*4+32*10=4416。那么 embedding 层的权重总数占比是 3,200,000 / (3,200,000 + 4416) = 99.86%。

也就是说 embedding 层的权重占据了整个网络权重的绝大部分。那么训练过程可想而知，大部分的训练时间和计算开销都被 Embedding 层所占据。正因为这个原因，Embedding 层往往采用预训练的方式完成。

2.2 Embedding 的预训练方法

通过上面对 Embedding 层的介绍，同学们肯定已经知道 Embedding 层的训练开销是巨大的。为了解决这个问题，Embedding 的训练往往独立于深度学习网络进行。在得到稀疏特征的稠密表达之后，再与其他特征一起输入神经网络进行训练。典型的采用 Embedding 预训练方法的模型是 FNN（如图 4）。

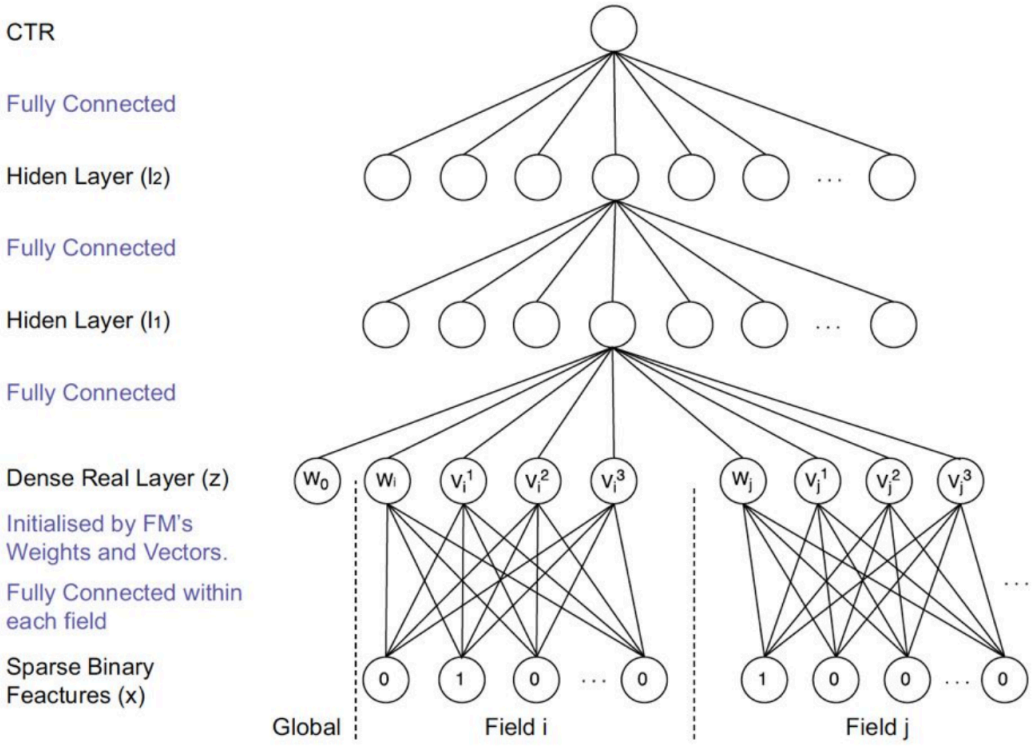


图4 FNN模型结构

FNN 利用了 FM 训练得到的物品向量，作为 Embedding 层的初始化权重，从而加快了整个网络的收敛速度。在实际工程中，直接采用 FM 的物品向量作为 Embedding 特征向量输入到后续深度学习网络也是可行的办法。

再延伸一点讲，Embedding 的本质是建立高维向量到低维向量的映射，而“映射”的方法并不局限于神经网络，实质上可以是任何异构模型，这也是 Embedding 预训练的另一大优势，就是可以采用任何传统降维方法，机器学习模型，深度学习网络完成 embedding 的生成。

典型的例子是 2013 年 Facebook 提出的著名的 GBDT+LR 的模型，其中 GBDT 的部分本质上也是完成了一次特征转换，可以看作是利用 GBDT 模型完成 Embedding 预训练之后，将 Embedding 输入单层神经网络进行 CTR 预估的过程。

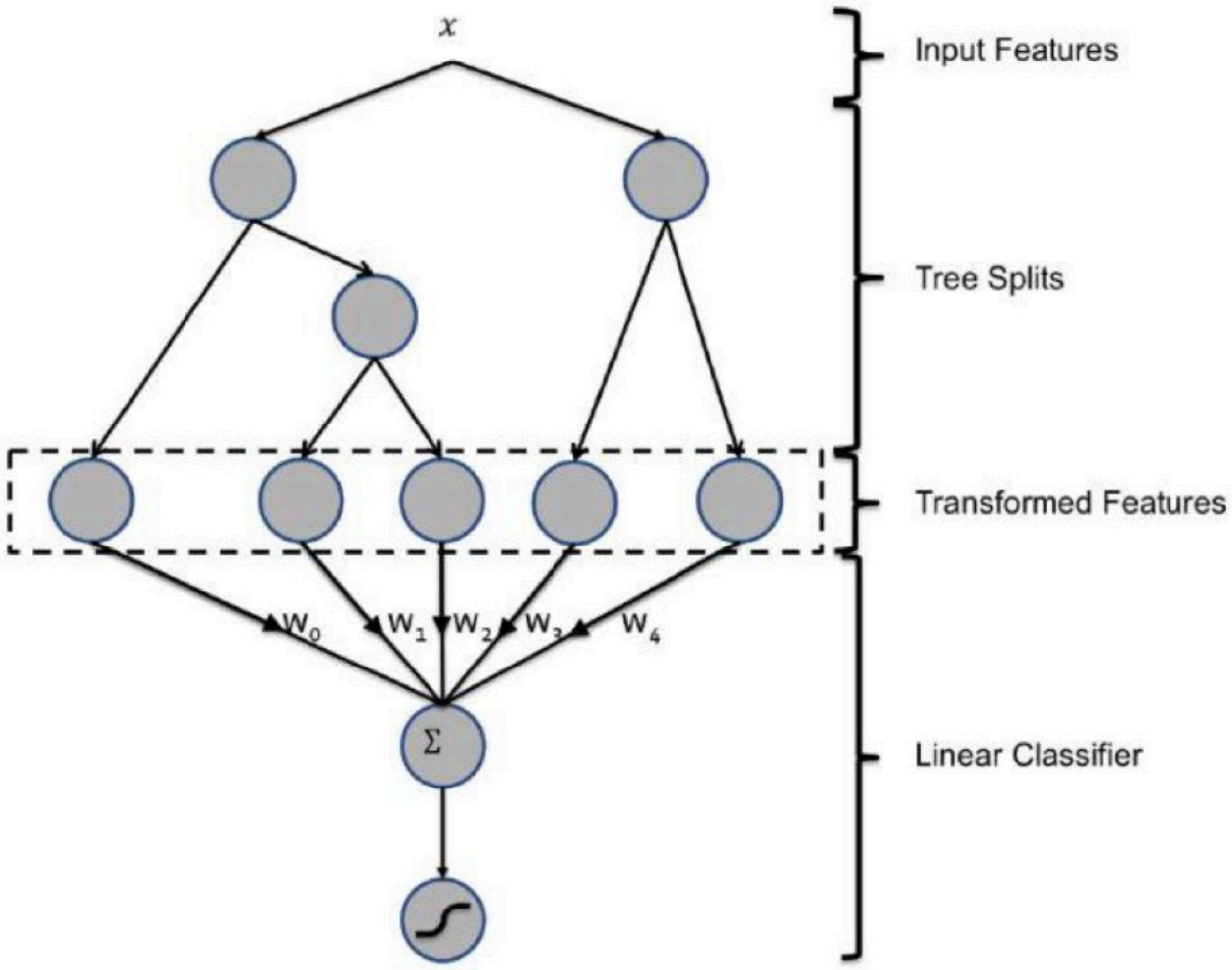


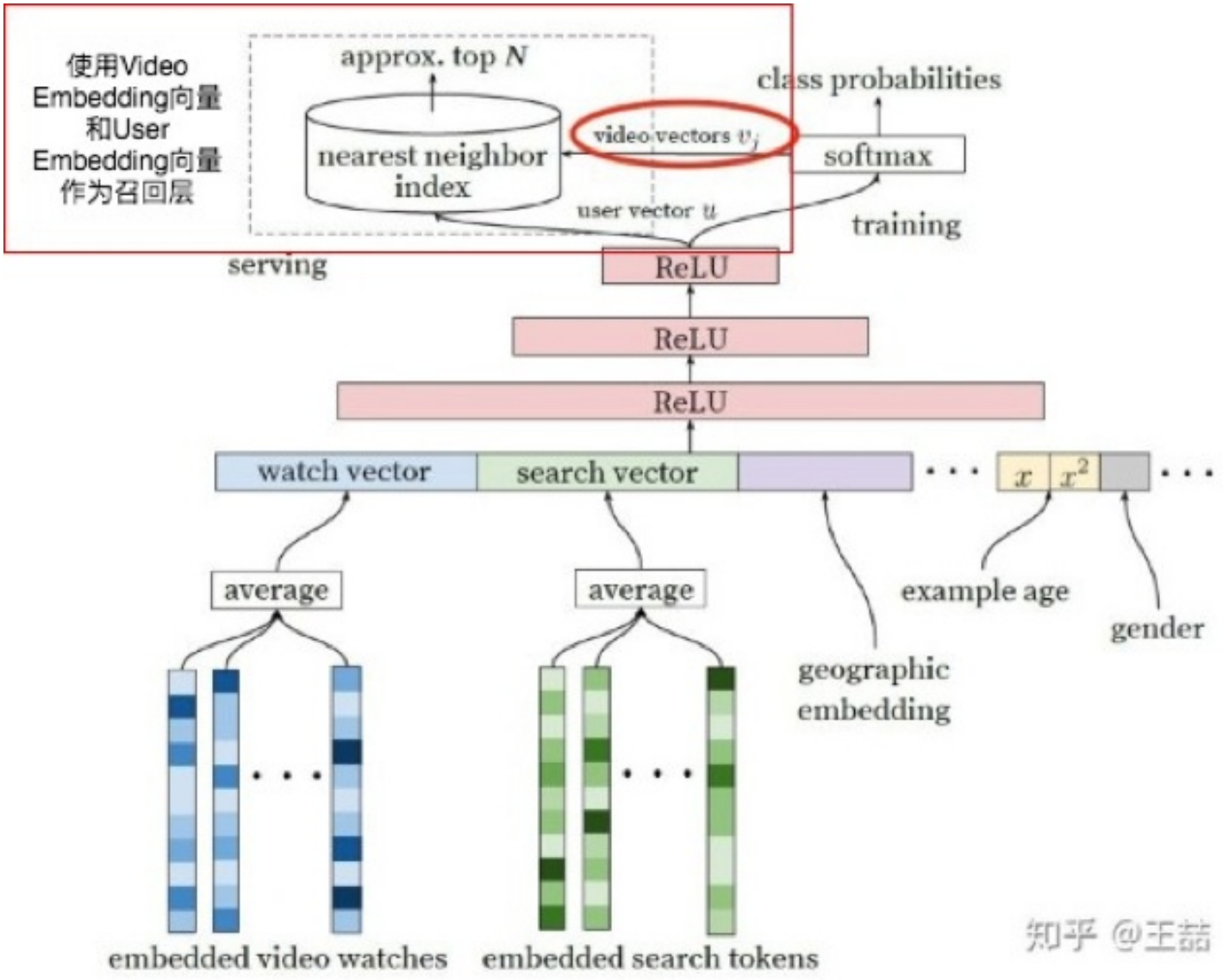
图5 GBDT+LR模型 GBDT完成Embedding过程

2015 年以来，随着大量 Graph Embedding 技术的发展，Embedding 本身的表达能力进一步增强，而且能够将各类特征全部融合进 Embedding 之中，这使 Embedding 本身成为非常有价值的特征。这些特点都使 Embedding 预训练成为更被青睐的技术途径。

诚然，将 Embedding 过程与深度网络的训练过程割裂，必然会损失一定的信息，但训练过程的独立也带来了训练灵活性的提升。举例来说，由于物品或用户的 Embedding 天然是比较稳定的（因为用户的兴趣、物品的属性不可能在几天内发生巨大的变化），Embedding 的训练频率其实不需要很高，甚至可以降低到周的级别，但上层神经网络为了尽快抓住最新的正样本信息，往往需要高频训练甚至实时训练。使用不同的训练频率更新 Embedding 模型和神经网络模型，是训练开销和模型效果二者之间权衡后的最优方案。

2.3 embedding 作为推荐系统或计算广告系统的召回层

随着 Embedding 技术的进步，Embedding 自身的表达能力也逐步增强，利用 Embedding 向量的相似性，直接将 Embedding 作为推荐系统召回层的方案越来越多的被采用。其中 Youtube 推荐系统召回层的解决方案是典型的做法。



我曾经在文章《重读 Youtube 深度学习推荐系统论文，字字珠玑，惊为神文》中介绍过了 Youtube 利用深度学习网络生成 Video Embedding 和 User Embedding 的方法。利用最终的 Softmax 层的权重矩阵，每个 Video 对应的列向量就是其 Item Embedding，而 Softmax 前一层的输出就是 User Embedding。在模型部署过程中，没有必要部署整个深度学习网络来完成从原始特征向量到最终输出的预测过程，只需要将 User Embedding 和 Item Embedding 存储到线上内存数据库，通过内积运算再排序的方法就可以得到 item 的排名。这大大加快了召回层的召回效率。

2.4 总结

事实上，除了上述的三种主要的 Embedding 应用方向，业界对于 Embedding 的创新性研究不仅没有停止，而且有愈演愈烈之势，阿里的 EGES，Pinterest 的 GNN 应用，Airbnb 基于 Embedding 的搜索模型等大量表达能力非常强的 Embedding 方法的诞生，使 Embedding 本身就已经成为了优秀的 CTR 模型和推荐系统模型。作为计算广告和推荐系统领域的从业者，无论如何强调 Embedding 的重要性都不过分。

3 扩展阅读

Embedding 从入门到专家必读的十篇论文：https://zhuanlan.zhihu.com/p/58805184

References

[1] “万物皆 embedding，从经典的 word2vec 到深度学习基本操作 item2vec.” [Online]. Available: https://zhuanlan.zhihu.com/p/53194407

[2] “Embedding 在深度推荐系统中的 3 大应用方向.” [Online]. Available: https://zhuanlan.zhihu.com/p/67218758