知乎 首发于 机器学习深度学习应用

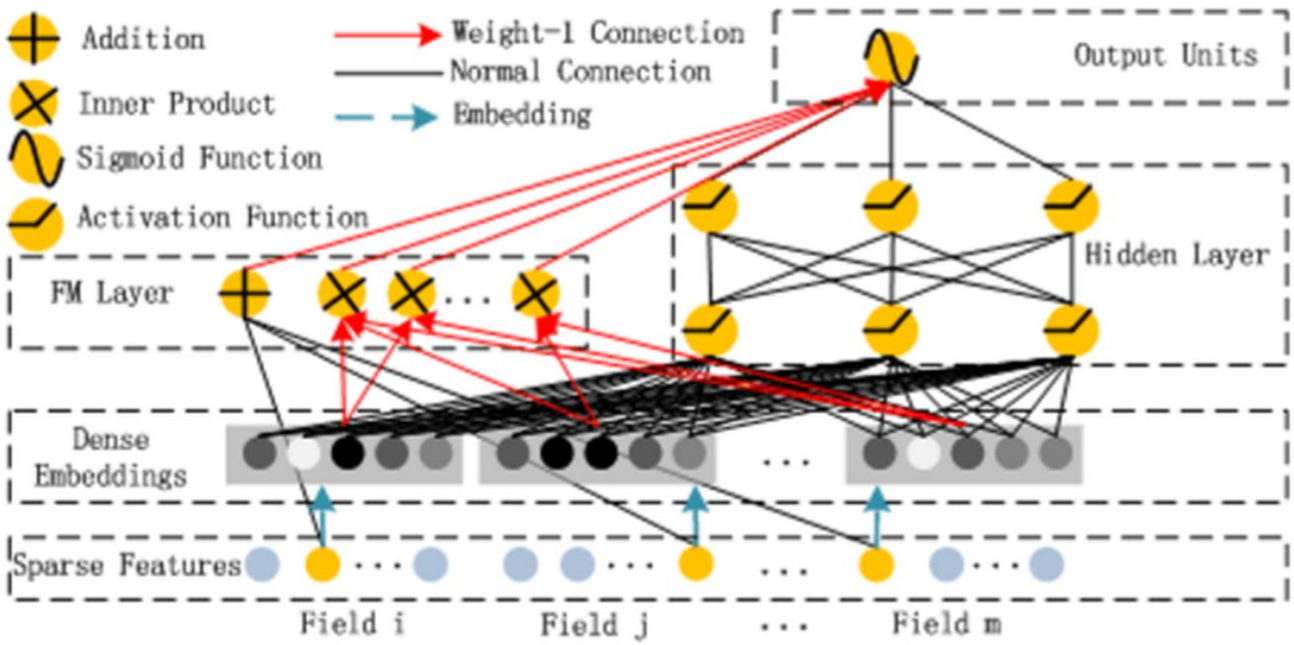


Figure 1: Wide & deep architecture of DeepFM. The wide and deep component share the same input raw feature vector, which enables DeepFM to learn low- and high-order feature interactions simultaneously from the input raw features.

深入浅出DeepFM

奔奔
机器学习、深度学习、推荐算法、反作弊、nlp

关注他

前言

由于DeepFM算法以及工程化实现内容太多，所以拆分成两个部分来介绍。具体工程实现参考：推荐算法注意点和DeepFM工程化实现。介绍DeepFM有点炒冷饭了，不过是自己做的工程化的实现，原理还是复习一遍，用理论指导实践，实践再反哺理论会有新发现。发现了一个新的路子FM-->FFM-->DeepFM-->DCN-->xDeepFM，以后会持续更新相关内容。

DeepFM简介

DeepFM是华为和哈工大在2017发表的论文，论文名称《DeepFM: A Factorization-Machine based Neural Network for CTR Prediction》。

感觉DeepFM是在Wide&Deep结构的基础上，使用FM取代Wide部分的LR，这样可以避免人工构

知乎 首发于 机器学习深度学习应用

以对比二阶更高阶的特征组合进行建模，实际上由于计算复杂度的原因，一般只用到二阶特征组合。其实可以理解为FM=LR+低阶特征的两两组合。

对于高阶的特征组合，很自然想到用DNN。稀疏的onehot可以通过dense层转到隐藏层，这样低阶特征和高阶特征的组合隐含的体现在隐藏层中。

最终低阶组合特征融合高阶组合特征，将DNN与FM进行一个合理的融合，就是DeepFM了。DeepFM=DNN+FM，下面就从FM开始介绍，顺带延伸一下FFM；再稍微介绍DNN部分，这样双拼就完成了。

FM部分

FM模型在2010年由Rendle提出的，最近几年才真正在各大厂大规模在CTR预估和推荐领域广泛使用。原文从两个角度来讲FM模型的重要性，从特征组合的衍化和矩阵分解的角度来讲，FM处于两者的交汇地带。

特征组合(从LR到SVM再到FM)

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i \quad \text{..... (a)}; \quad \sigma(\hat{y}) = \frac{1}{1+e^{-\hat{y}}} = \frac{1}{1+e^{-w^T x}} \quad \text{.....(b)}$$

上图a是线性模型的公式， w_0 是偏置项， x_i 是一条样本的特征 i 的值，w_i 是特征 i 的权重；上图b是LR的预测公式，也就是线性模型 $\hat{y}$ 加上一个sigmoid激励函数。

LR用于特征组合是线性模型+人工构造非线性特征，优点是可解释性强，模型简单，上线速度快；缺点是人工构造特征费时费力且不易构造。

SVM 引入核函数的概念来学习交叉特征，为两两的特征组合分配一个权重参数。这些新的权重参数和原始特征对应的参数一样，交给模型在训练阶段学习。预测函数如下：

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} x_i x_j$$

其中 $w_{i,j}$ 是特征 x_i 和特征 x_j 乘积的权重。这就是核函数为二阶多项式核的 SVM 模型。这种二阶笛卡尔乘积的暴力组合，会导致特征维度上的灾难，两两特征不一定都有效，并且特征非常稀疏，没法更新学习参数。由于 $w_{i,j}$ 的取值取决于 x_i 和 x_j 的乘积，在数据稀疏的场景下，可能存在训练集中 $x_i x_j$ 始终为零的情况，这样的话，模型就无法有效的更新权重 $w_{i,j}$ 了；在预测阶

知乎 首发于 机器学习深度学习应用

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < V_i, V_j > x_i x_j$$

其中：$< V_i, V_j >$ 表示特征向量 V_i 和特征向量 V_j 的内积；V_i 是 x_i 特征的隐因子向量；V_j 是 x_j 特征的隐因子向量。通过隐因子向量的维度来表征特征之间的关系，这本质上是在对特征进行embedding化表征。它认为两个特征之间，即使没有共同出现在一条样本中，也是有间接联系的。例如，特征A和特征B曾在一些样本中一起出现过，特征B和特征C曾在一些样本中出现过，那么特征A和特征C无论是否在样本中一起出现过，仍然是有些联系的。

既然二阶特征组合可以学到隐因子向量，那么三阶、四阶、五阶的特征是否也可以呢？答案是可以但是复杂度增加了很多，一般只用到二阶就可以了。

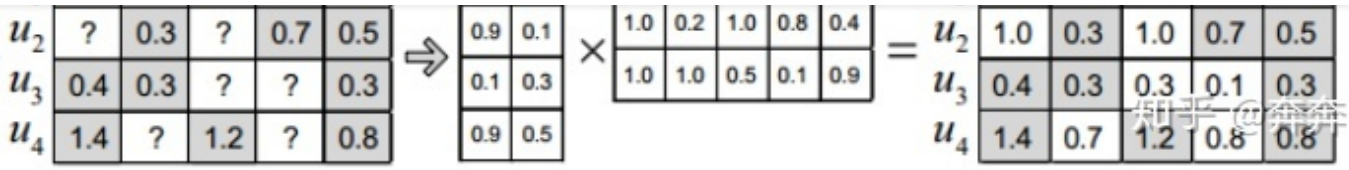
矩阵分解

上面讲了这么多，那么和矩阵分解有什么关系呢？先说结论因子分解机是把矩阵分解和逻辑回归融合了。下面具体阐述一下:

上图中每一条稀疏样本都记录了用户对电影的评分。最右边的y是评分；左边的特征有五种，用户ID、当前评分的电影ID、曾经评过其他电影得分、评分时间、上一次评分的电影。

基础的矩阵分解（MF),将评分矩阵R分解为用户矩阵U和项目矩阵S， 通过不断的迭代训练使得U和S的乘积越来越接近真实矩阵，$R = U^T S$ 矩阵分解过程如图:

知乎 首发于 机器学习深度学习应用



如果构建如上图的用户ID和电影ID的评分矩阵，那么就是标准的矩阵分解。

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < V_i, V_j > x_i x_j$$

对照着FM的公式来看。如果用户ID和电影ID，在一条样本中，特征维度只有1，其他为0，也就是说特征值在自己的维度为1，其他维度为0，直接是 w_u 和 w_i；二阶部分特征乘积也只剩下了一个1，其他都为0了。那么FM公式变为：

$$\hat{y} = w_0 + w_u + w_i + < V_u, V_i >$$

其中： w_0 整个网站的平均分， w_u :用户的评分偏置， w_i ：电影的评分偏置。这样就变成了带有偏置信息的SVD了。

带有偏置信息SVD基础上加上用户历史评过分的电影ID,就是SVD++；带有偏置信息SVD基础上加上时间信息就是time-SVD。

本质上，FM可以看做包含除了User ID和Item ID外，还带有很多其他特征信息的MF；也相当于融合了逻辑回归和矩阵分解（FM=LR+低阶特征的两两组合）。因此，MF是FM的特例。

FM 效率问题

从FM的原始数据公式看，二阶特征两两组合，如果隐因子向量的维度是k，特征维度为n,那么时间复杂度就是 $O(kn^2)$ 。如果这样的复杂度就不是一个实用化的模型了，所以需要化简：

$$
\begin{aligned}
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j\rangle\, x_i\, x_j - \frac{1}{2}\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_i\rangle\, x_i\, x_i \qquad 1\\
&= \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{j,f}\, x_i\, x_j - \sum_{i=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{i,f}\, x_i\, x_i\right) \qquad 2\\
&= \frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)\left(\sum_{j=1}^{n} v_{j,f}\, x_j\right) - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right) \qquad 3\\
&= \frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right) \qquad 4
\end{aligned}
$$

整个化简过程

讲一下化简的过程：

第一步： $j = i + 1$ 变成了从 $j = 1$ 开始，相当于把矩阵的上三角扩展成全部，为了还原成上三角，所以要除以2；中间的 $< v_i, v_i >$ 是不算的，在前半部分多算了，所以要减去。

第二步：在第一步中的 v_i 和 v_j 是k维向量，把点积公式展开。

第三步：把第二步中的k维向量共同的点积部分提取到外部。

第四步：第三步中的前半段i=1和j=1都是求和，下标一样，值也相同，合并改成平方项。

此时，复杂度从 $O(kn^2)$ 降低到 $O(kn)$ 。

模型训练

FM用来做CTR预估，目标是一个二分类，输出也要经过sigmoid变换，因此损失函数也就是常用的logistic loss。

$$loss(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^i log(\sigma(\hat{y})) + (1 - y^i) log(1 - \sigma(\hat{y}))]$$

公式中 $\sigma(\hat{y})$ 是因子分解机的预测输出后，经过sigmoid函数变换得到的预估CTR， y^i 是真实样本的类别标记，正样本是1，负样本是0，m是样本总数。对这个损失目标函数使用梯度下降或者随

知乎 首发于 机器学习深度学习应用

$$\frac{\partial}{\partial\theta}\hat{y}(\mathbf{x}) = \begin{cases} x_i, & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta \text{ is } v_{i,f} \end{cases}$$

如果 θ 是 w_0 ，那么导数为1；如果 θ 是 w_i ，那么导数为 x_i ；如果 θ 是 $v_{i,f}$ 那么导数为 $x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2$ ；由于 $\sum_{j=1}^{n} v_{j,f} x_j$ 只与 f 有关，独立于 i ，可以提前计算。通常每一次梯度计算的时间复杂度为 $O(1)$ ，所有参数都被更新的复杂度为 $O(kn)$ 。综上所述，FM可以在线性时间训练和预测，是一种非常高效的模型。

FM的优点

1. 高维稀疏特征下交叉仍能估计，可以泛化未被观察的交叉，避免大量构建人工特征，费时费力。
2. 训练时参数的学习和模型的预测，时间复杂度是线性的。

FFM

稍微说一下FFM(Field-aware Factorization Machines)，不但认为特征和特征之间有关系，还认为特征和特征类型有关系。特征类型就是某些特征是来自数据的同一个字段。因子分解机模型的是这样：

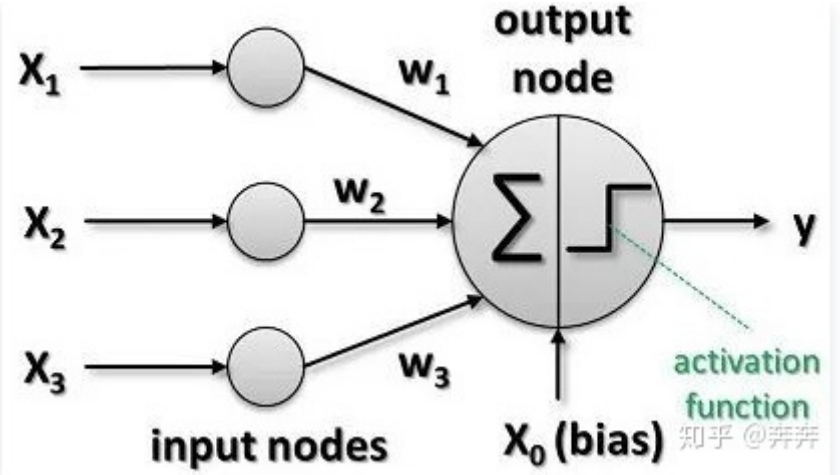
$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < V_i, V_j > x_i x_j$$

FFM改进的是二阶组合那部分，改进的模型认为每个特征有f个隐因子向量，这里的f就是特征一共来自多少个字段（Field），二阶组合部分改进后如下：

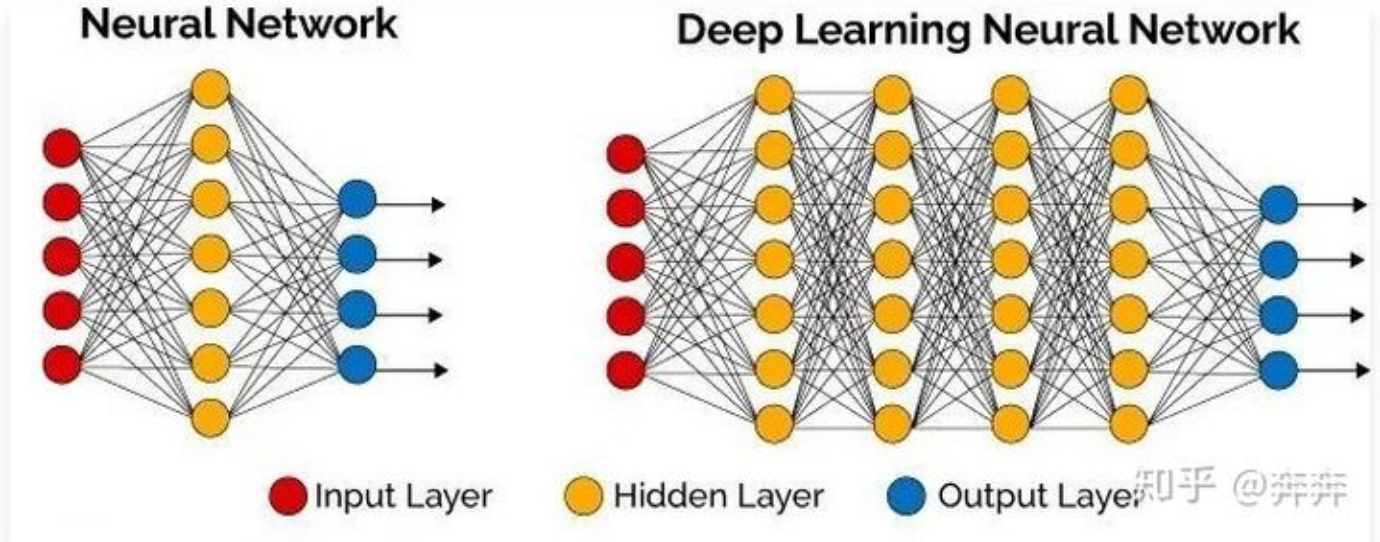
$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} < V_{i,fj}, V_{j,fi} > x_i x_j$$

FFM模型也常用来做CTR预估。FFM是FM的一个特例，它更细致地刻画了字段特征。首先它做了任意两个特征组合，但是区别在于，怎么刻划字段特征？FM只有一个隐因子向量，但FFM现在有两个向量，也就意味着同一个特征，要和不同的fields进行组合的时候，会用不同的embedding去组合，它的参数量更多。优点是效果好于FM，缺点是参数扩大了f倍，内存消耗大，不利于实用化。

不是和LR很像？。



2006，Hinton 发表了经典论文“Reducing the Dimensionality of Data with Neural Networks”。这篇论文提出了预训练（Pre-training）的方法，如下图：



加入了隐藏层，可以是多层，输出层也可以不止一个，多个输出，扩展了激活函数，每层都可以有激活函数。层与层之间是全连接，也就是说，第i层的任意一个神经元一定与第i+1层的任意一个神经元相连。虽然DNN看起来很复杂，但是从小的局部模型来说，还是和感知机一样，即一个线性关系 $z = \sum w_i x_i + b$，加上一个激活函数 $\sigma(z)$。

通过前向传播建立目标函数，反向传播通过链式求导法则，优化损失函数，减小误差更新参数。如果是二分类的话，最后一层通常是Sigmoid激活函数+logistic loss；如果是多分类的话，最后一层通常是Softmax+交叉熵损失函数。其实logistic loss是交叉熵损失的特殊情况。

Softmax 函数如下：

知乎 首发于 机器学习深度学习应用

$\sum_i e^{V_i}$

其中，V_i 是分类器前级输出单元的输出。i 表示类别索引，总的类别个数为 C。S_i 表示的是当前元素的指数与所有元素指数和的比值。Softmax 将多分类的输出数值转化为相对概率。

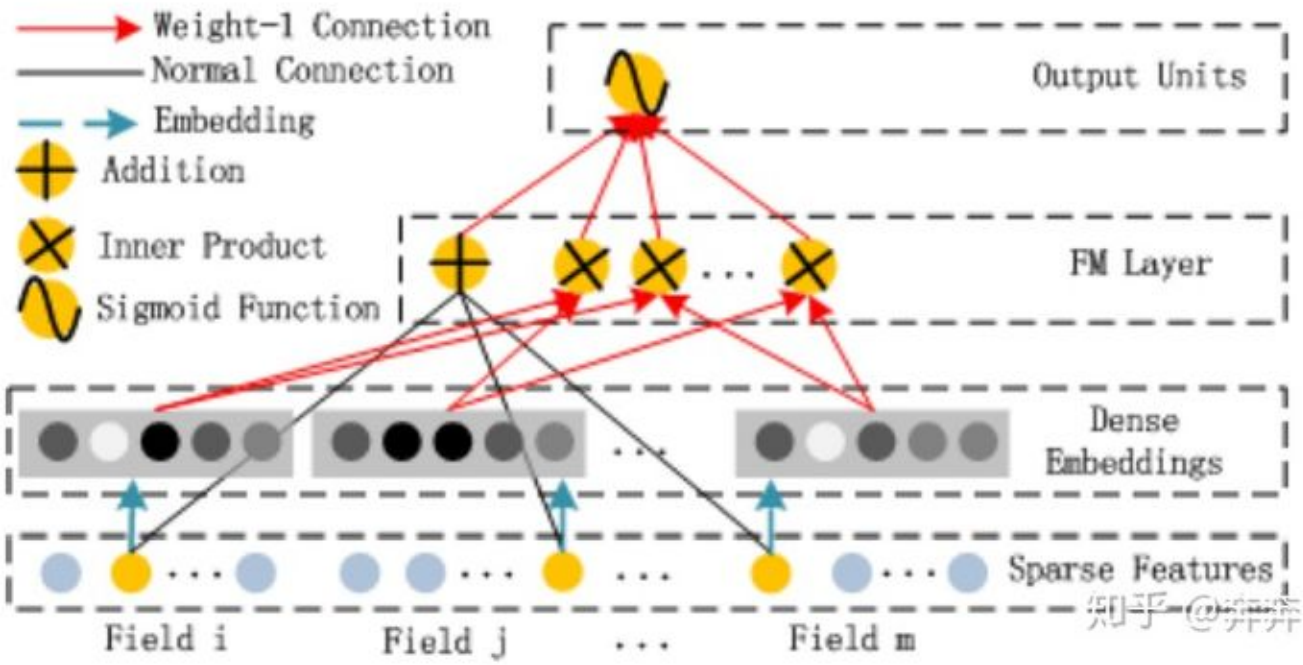
交叉熵损失函数如下：

$$H(p,q) = -\sum_{i=1}^{n} p(x_i) log(q(x_i))$$

其中，$p(x_i)$ 表示真实值，$q(x_i)$ 表示求出的softmax值。

DeepFM模型

下面介绍一下论文中的结构图以及预测的函数。FM部分结构图如下：

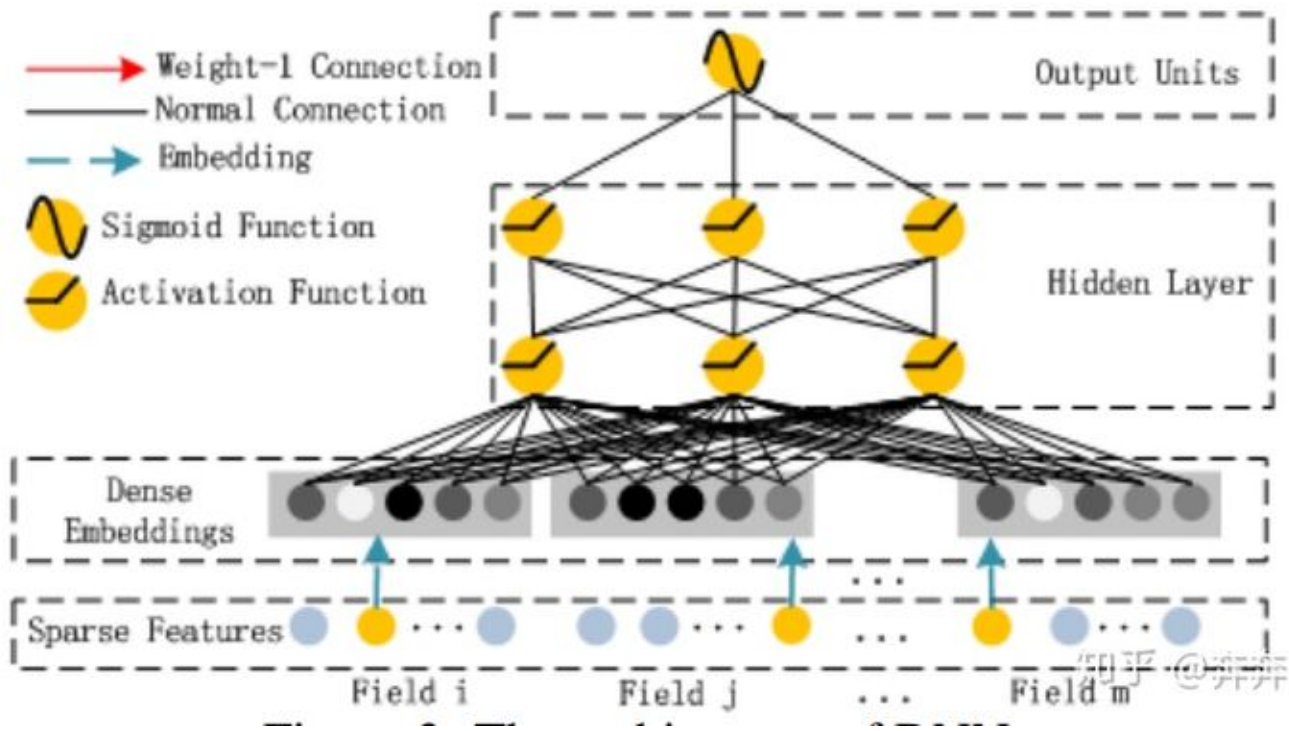


FM部分

预测函数为：

$$y_{FM} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < V_i, V_j > x_i x_j$$

知乎 首发于 机器学习深度学习应用



DNN部分

下面具体介绍一下DNN部分的预测函数：

$$a^{(0)} = [e_1, e_2, \ldots, e_m]$$

e_i ： 第 i 个特征的embedding, m ： 表示特征总数， $a^{(0)}$ ： 初始化神经网络。

$$a^{l+1} = \sigma(W^l a^l + b^l)$$

l ： 网络层的深度， σ :激活函数， a^l ： l 层的输出， W^l ： l 层权重， b^l ： l 层的偏置项。

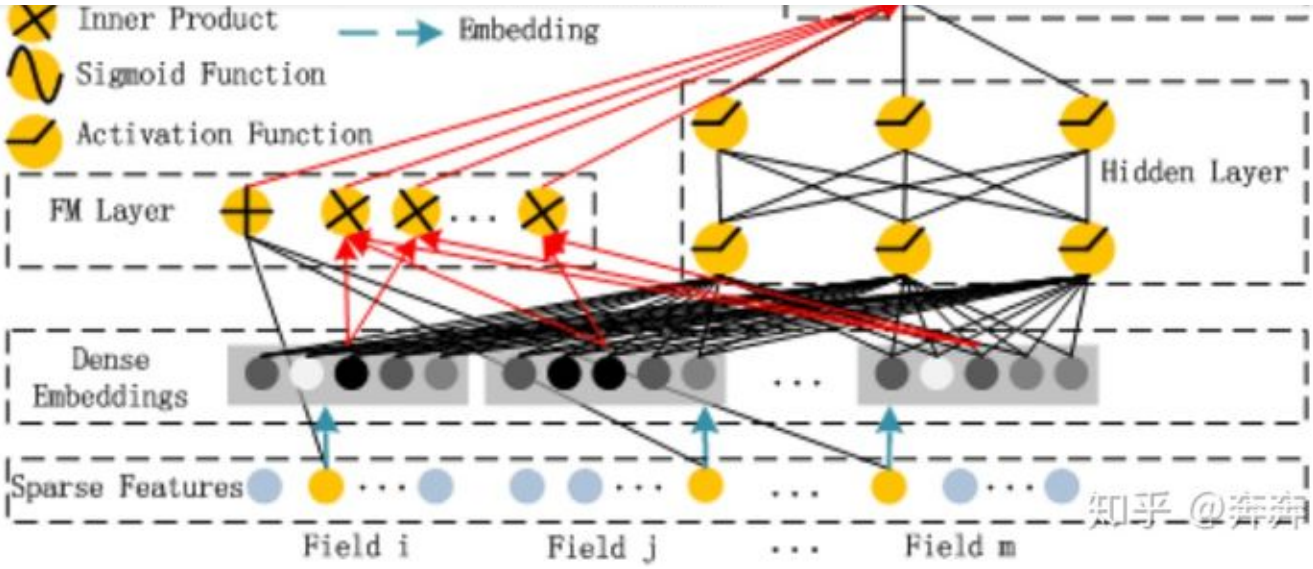
最后输入sigmoid函数中的预测：

$$y_{DNN} = W^{|H|+1} \cdot a^{|H|} + b^{|H|+1}$$

其中 $|H|$ 表示隐藏层的个数。

上面分别介绍了FM和DNN,下面把他们融合起来。典型网络融合有两种方式，一种是并行结构，一种是串行结构，DeepFM采用的是并行的方式。

知乎 首发于 机器学习深度学习应用



DeepFM架构图

DeepFM 包含两部分：神经网络部分与因子分解机部分，这两部分共享同样的输入向量，这使得DeepFM可从输入的原始特征中同时学习低阶和高阶特征交互。

DeepFM 预测函数：

$$\hat{y} = sigmoid(y_{FM} + y_{DNN})$$

损失函数为交叉熵损失函数，在CTR预估中就是logistic loss。

效果

论文中对点击率预估做了对比实验，具体实验结果如下图所示：

知乎 首发于 机器学习深度学习应用

	AUC	LogLoss	AUC	LogLoss
LR	0.8641	0.02648	0.7804	0.46782
FM	0.8679	0.02632	0.7894	0.46059
FNN	0.8684	0.02628	0.7959	0.46350
IPNN	0.8662	0.02639	0.7971	0.45347
OPNN	0.8657	0.02640	0.7981	0.45293
PNN*	0.8663	0.02638	0.7983	0.45330
LR & DNN	0.8671	0.02635	0.7858	0.46596
FM & DNN	0.8658	0.02639	0.7980	0.45343
DeepFM	**0.8715**	**0.02619**	**0.8016**	**0.44985**

算法对比优点

- 逻辑回归(LR)：利用线性特征，缺少低阶交叉特征和高阶特征。低阶交叉特征和高阶特征需要自己构造，费时费力还有点玄学。
- DNN：考虑高阶特征，缺少对于低阶交叉特征和线性特征的考虑。
- CNN：考虑近邻特征的关系，较单一，适合图片分类。
- RNN：考虑更多的是数据时序性，较单一。
- FM：考虑线性和低阶交叉特征，缺少高阶特征。
- Wide&Deep：考虑了线性特征和高阶特征，低阶交叉特征需要手动交叉生成。
- DeepFM：具有线性特征、低阶交叉特征和高阶特征，且不需要人为做低阶交叉特征，FM模块和Deep模块共享Feature Embedding部分，可以更快的训练，以及更精确的训练学习。

具体推荐算法实践见：

奔奔：推荐算法注意点和DeepFM工程化实现
zhuanlan.zhihu.com

推荐系统多目标学习：

https://zhuanlan.zhihu.com/p/183760759
zhuanlan.zhihu.com