FM算法

- 什么是CTR

  - Click-Through-Rate（点击通过率）用户点击某个广告的概。我是渣渣辉。。。
  - 说人话就是：预测一个广告会不会被点击，可以当做一个二分类问题
  - 好做吗？不太容易，主要是特征层面的问题
  - 什么算法合适呢？咱们主要唠FM系列

FM算法

CTR算法

- 最早用线性模型来做，看起来十分简单，却不太好用！
- 线性模型单独考虑特征和权重之前的关系，没有组合信息
  （例如哈尔滨烤冷面，俩特征分别是：哈尔滨，烤冷面，组合起来才是那个味）
- 特征工程：不就是搞特征嘛，我就使劲想哪些能组合，整合上去
- 一般数据挖掘问题也确实都是偏人工的，但是推荐系统中，特征指标太多了，人工做特征工程这个事也太麻烦了，而且人是善变的！

FM算法

- CTR算法

  - 利用树模型来做，树模型本身就有特征选择的功能，再结合集成算法

  - 二项式特征我全要，穷举所有可能组合都往上整
    （行是行，但是数据层面会遇到一些问题，一会举例）

  - FM算法（Factorization Machine），解决了特征层面的问题

  - 可以先来想一想，推荐系统中涉及的数据会有什么样的特点呢?

FM算法

- 特征带来的问题

  - 用户的特征，商品的特征，行为的特征，能有的信息实在太多了

  - 不仅特征维度高，主要还涉及很多离散型特征，通常都用one-hot来处理

  - 编码后的结果：

$$\underbrace{[0,1,0,0,0,0,0]}_{\text{Weekday=Tuesday}}\quad\underbrace{[0,1]}_{\text{Gender=Male}}\quad\underbrace{[0,0,1,0,\ldots,0,0]}_{\text{City=London}}$$

  - 你以为这就完了？咱们刚才不是说要考虑特征之间的组合嘛。。。
    （如果仅考虑二阶情况，这特征量都有点吓人了）

FM算法

- 如何解决高维且非常稀疏的特征?

  - 还记得特征组合的意义吧：（宅男，游戏）（中年人，保健品）

  - 二阶多项式：$$y(X) = \omega_0 + \sum_{i=1}^{n} \omega_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} x_i x_j$$

  - 前面还是普通的线性模型，只不过多了二阶的部分

  - 目的肯定是要求解权重参数，你也肯定知道要用梯度下降，有没有啥问题?

FM算法

- 如何解决高维且非常稀疏的特征?

  - 参数量有多少呢？线性的还好，但是二阶的出现了平方项

  - 二阶多项式： $y(X) = \omega_0 + \sum_{i=1}^{n} \omega_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} x_i x_j$

  - 一般难道的特征本来都很稀疏了，二阶得啥样？那不更稀疏了嘛!

  - 要训练权重参数，二阶的地方必须要两个特征都非零才可以!

FM算法

- 先来看个定理

  - N阶实对阵矩阵可以分解为：$A = Q\Lambda Q^T$ （PCA课里曾说过的对角化）
  - 想一想咱们的二阶权重参数矩阵：$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\omega_{ij}x_i x_j$ （同样是对称阵，公式为一半）
  - 如果对W进行分解：$W = V^T V$ （其中的向量V到底表示什么呢？）

$$W^{\star} = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2n} \\ \dots & \dots & \dots & \dots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nn} \end{bmatrix} = V^T V = \begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_n \end{bmatrix} \times [V_1, V_2, \dots, V_n] = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1k} \\ v_{21} & v_{22} & \dots & v_{2k} \\ \dots & \dots & \dots & \dots \\ v_{n1} & v_{n2} & \dots & v_{nk} \end{bmatrix} \times \begin{bmatrix} v_{11} & v_{21} & \dots & v_{n1} \\ v_{12} & v_{22} & \dots & v_{n2} \\ \dots & \dots & \dots & \dots \\ v_{1k} & v_{2k} & \dots & v_{nk} \end{bmatrix}$$

FM算法

- 先来看个定理

- Xi与Xj的二项系数其实就它俩各自的隐向量的内积，即：$w_{ij} = \langle v_i, v_j \rangle$

- 隐向量表示什么呢？向量的维度如何定义呢？

- 可以把隐向量当做是潜意识特征，比如宅男打游戏的向量表示，维度通常较低

$$W^{\star} = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2n} \\ \dots & \dots & \dots & \dots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nn} \end{bmatrix} = V^T V = \begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_n \end{bmatrix} \times [V_1, V_2, \dots, V_n] = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1k} \\ v_{21} & v_{22} & \dots & v_{2k} \\ \dots & \dots & \dots & \dots \\ v_{n1} & v_{n2} & \dots & v_{nk} \end{bmatrix} \times \begin{bmatrix} v_{11} & v_{21} & \dots & v_{n1} \\ v_{12} & v_{22} & \dots & v_{n2} \\ \dots & \dots & \dots & \dots \\ v_{1k} & v_{2k} & \dots & v_{nk} \end{bmatrix}$$

FM算法

- 求解过程

  - 重点还是在二阶：$\hat{y}(X) := \omega_0 + \sum_{i=1}^{n} \omega_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j$
  - 借鉴：$ab + ac + bc = \frac{1}{2}\left[(a+b+c)^2 - (a^2 + b^2 + c^2)\right]$
  - 从而：$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} < v_i, v_j > x_i x_j - \frac{1}{2} \sum_{i=1}^{n} < v_i, v_i > x_i x_i$
  - 其实也就是对称阵W的上半部分

FM算法

- 求解过程

原式：$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}<v_i,v_j>x_ix_j=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}<v_i,v_j>x_ix_j-\frac{1}{2}\sum_{i=1}^{n}<v_i,v_i>x_ix_i$$

内积展开：$$=\frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{f=1}^{k}v_{i,f}v_{j,f}x_ix_j-\sum_{i=1}^{n}\sum_{f=1}^{k}v_{i,f}v_{i,f}x_ix_i\right)$$ k表示隐向量的维度

合并同类项：$$=\frac{1}{2}\sum_{f=1}^{k}\left[\left(\sum_{i=1}^{n}v_{i,f}x_i\right)\cdot\left(\sum_{j=1}^{n}v_{j,f}x_j\right)-\sum_{i=1}^{n}v_{i,f}^2x_i^2\right]$$

合并成平方项：$$=\frac{1}{2}\sum_{f=1}^{k}\left[\left(\sum_{i=1}^{n}v_{i,f}x_i\right)^2-\sum_{i=1}^{n}v_{i,f}^2x_i^2\right]$$

FM算法

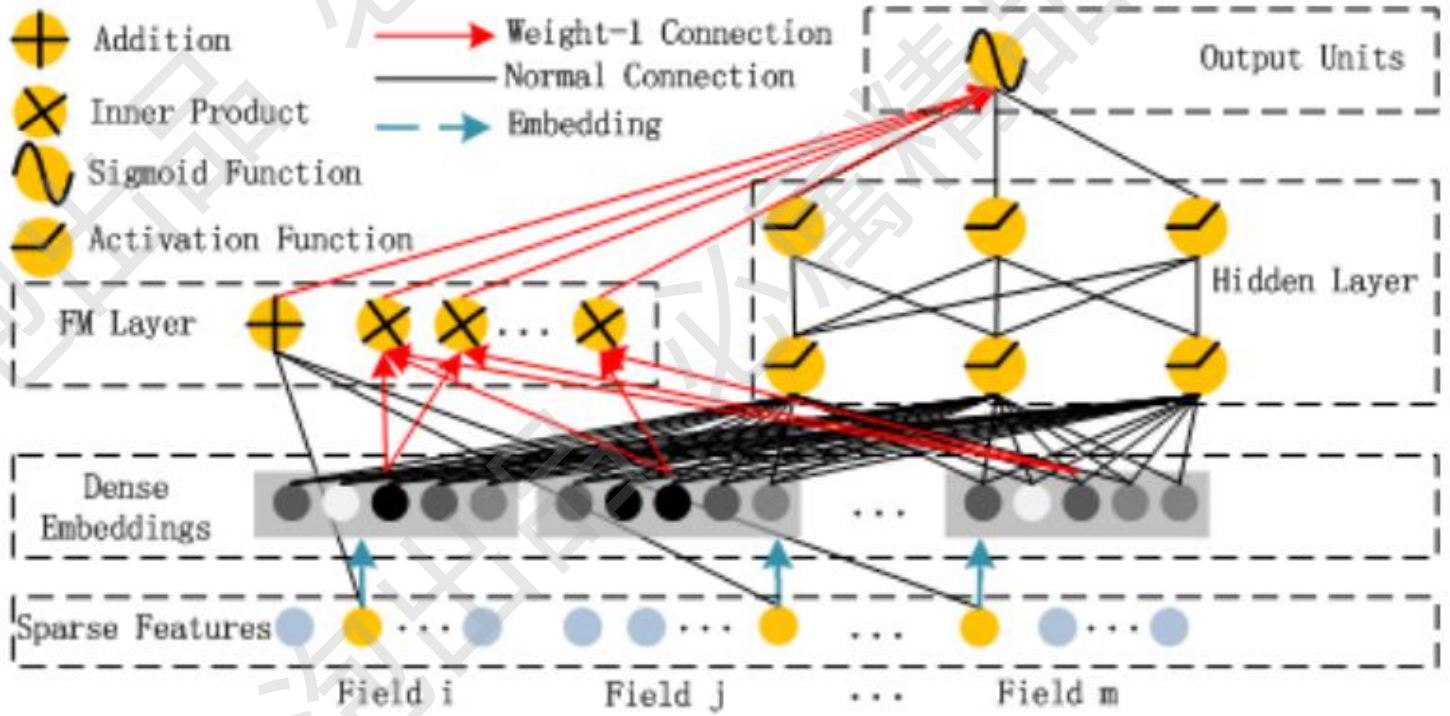
- 总结一下

  - 原公式二阶项复杂度o(n^2)，现在只需要kn（n个隐向量，维度为k）
  - V如何表示出来的呢？好像还得预训练，挺麻烦个事，一会咱们直接DeepFM
  - 特征组合较多的时候，自然会想到一个好兄弟：神经网络！
  - 接下来咱们瞅瞅DeepFM是干啥的

DeepFM

- FM融合到神经网络中

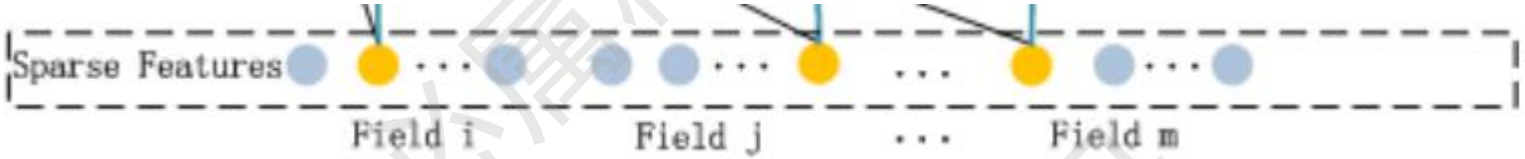
- 迪哥真理：不管啥算法，但凡用深度学习做，都更简单了
- 整体架构：FM+DNN
- embedding是FM重点
- 还是要进行特征组合



FM算法

- 输入层

  - 输入的数据长什么样：



  - Field表示一个特征，因为one-hot把特征展开了，所以一个filed有多个值

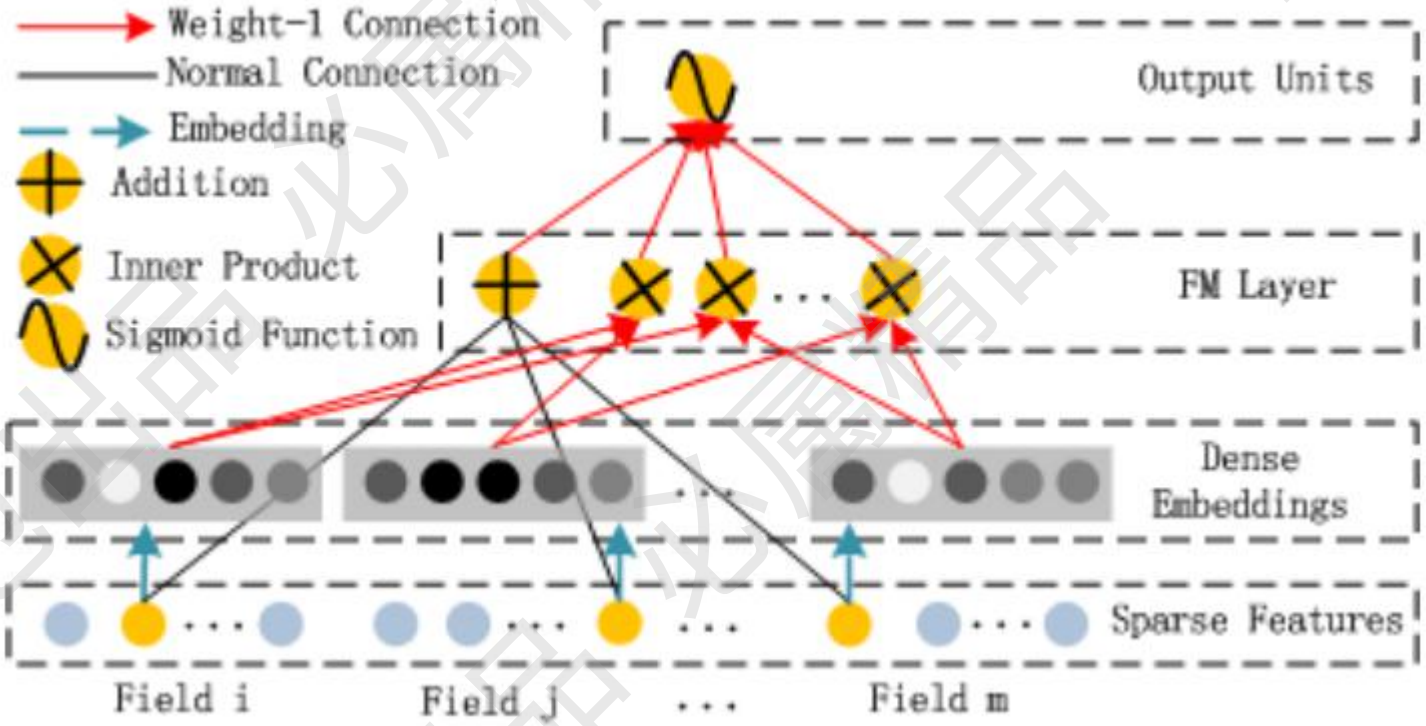
  - 因为一会要做embedding，传入的并不是实际数据，而是索引

  - 例如（男生/女生；哈尔滨/沈阳/长春；打篮球/不打篮球）
    （男生，哈尔滨，打篮球：特征索引为 0，2，5；特征值为1，表示取当前特征）

FM算法

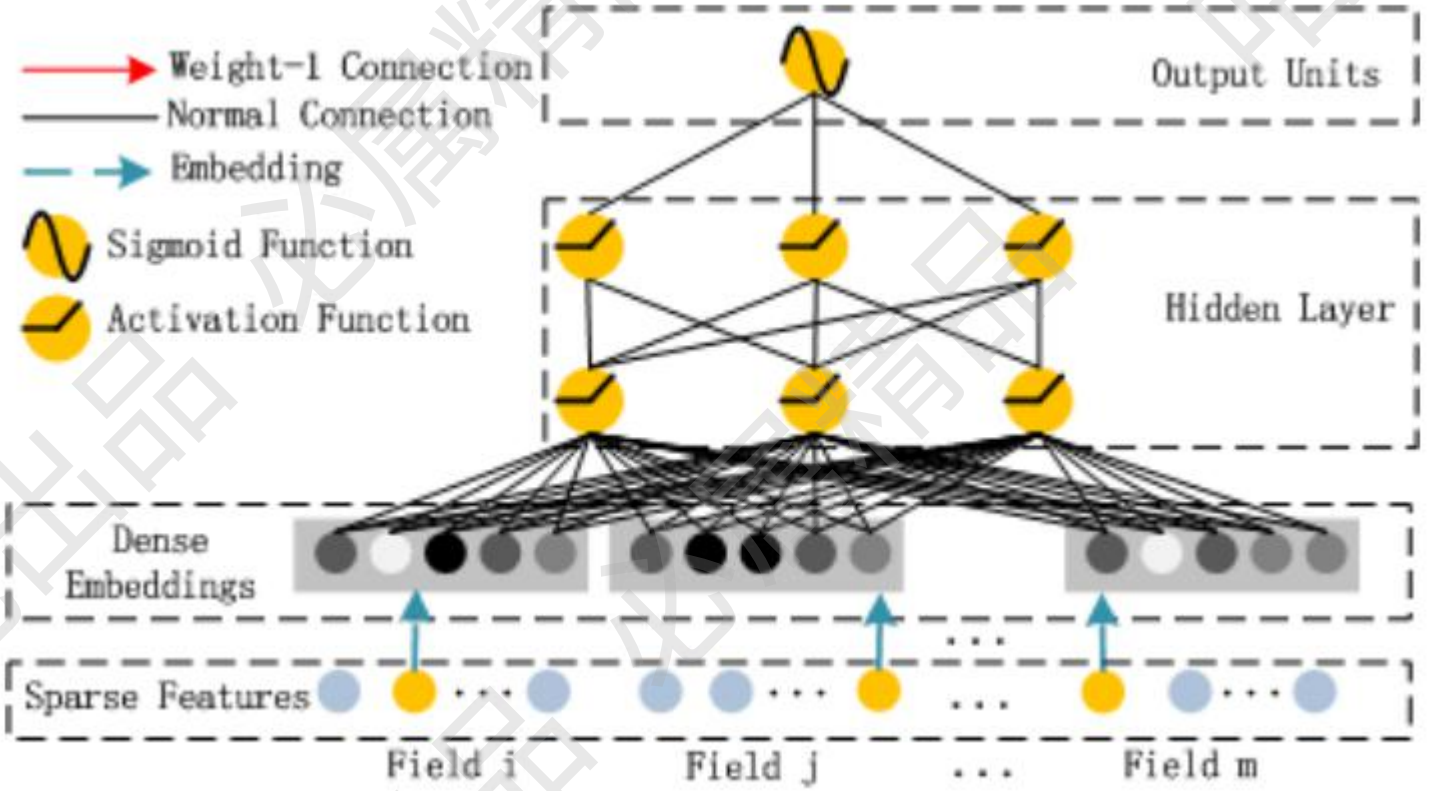
- Embedding层

  - 其实是设计多组权重
  - 其中包括了一阶和二阶
  - 权重参数就是embedding
  - 源码见!



FM算法

- DNN层
  - 就是传统神经网络
  - 全连接就搞定了
  - 最终输入sigmoid结果
  - 输出：



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