CTR综述

- CTR综述
  - CTR的应用场景
  - CTR的难点
  - 现有方法的研究方向
  - 提取交叉特征的方法
  - CTR的主流技术
  - CT预测总体流程
  - 参考文献

CTR的应用场景

CTR主要应用在广告系统和推荐系统当中，其目的在于预估出用户点击相应广告/推荐物品的概率。

CTR在推荐系统中的应用方式主要有两种：

i. 推荐列表依据 CTR 的值进行排序

ii. 推荐列表依据 $CTR * bid$ 的值进行排序（ bid是指如果用户点击该推荐，商家可以获得的收益）

CTR在广告系统中的应用也是类似，平台会根据广告收益和CTR的值对广告进行排序。

CTR的难点

i. 难以发掘有效的交叉特征：

- 高阶特征本身发掘难度较大，只能通过ML学习
- 低阶特征在特征维度较多时同样难以发现

现有方法的研究方向

- 更充分的提取低阶和高阶交叉特征；
- 自动提取交叉特征，很多线性模型依然需要大量的专业业务领域的知识进行人工构造；

提取交叉特征的方法

- FM
- Ploylearn
- DNN
- GBDT

CTR的主流技术

i. Factorization machine [1], 2010

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

> FM（Factorization Machine）是由Konstanz大学Steffen Rendle（现任职于Google）于2010年最早提出的，旨在解决稀疏数据下的 自动特征组合 问题。
>
> **Code** :
> Sklearn实现了FM和ploylearn（另外一种提取低阶特征的方法）
> https://github.com/scikit-learn-contrib/polylearn
> FastFMlib: https://github.com/ibayer/fastFM
> 作者Rendle在2012年用C++实现了LibFM，并发表论文 [14]

ii. Field-aware Factorization Machine, 2012

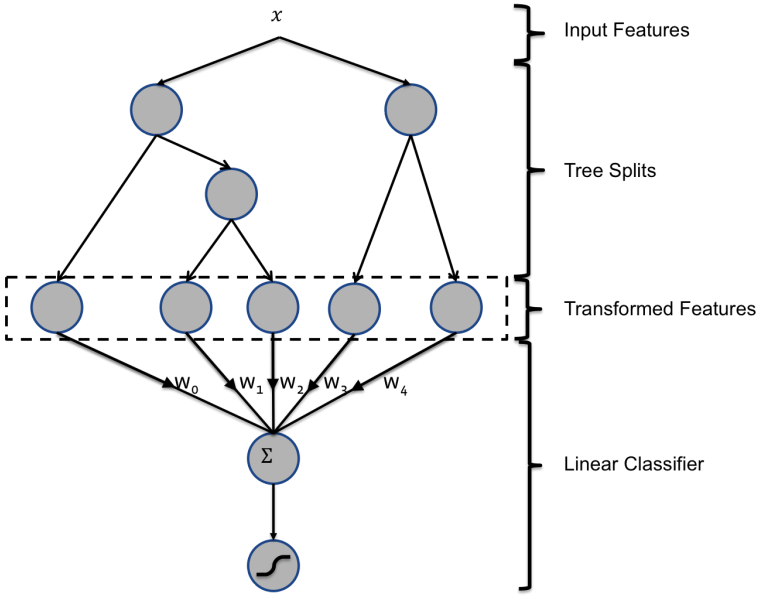
$$\phi_{FFM}(w, x) = \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (w_{j_1,f_2} \cdot w_{j_2,f_1}) x_{j_1} x_{j_2}$$

> FFM最初的概念来自Yu-Chin Juan（阮毓钦，毕业于中国台湾大学，现在美国Criteo工作）与其比赛队员，是他们借鉴了来自Michael Jahrer的论文 [3] 中的field概念提出了FM的升级版模型。通过引入field的概念，FFM把相同性质的特征归于同一个field。
>
> **Code**
> 台大'3idiots'用GBDT+FFM赢了criteo: https://github.com/guestwalk/kaggle-2014-criteo
> 台大'4idiots'（+MJ）用FFM赢了avazu:
> https://github.com/guestwalk/kaggle-avazu

iii. Follow the Regularied Leader, 2013

> FTRL是一种基于逻辑回归的 在线学习算法 ,能够学习出有效且稀疏的模型。FTRL 是由 Google 的 H. Brendan McMahan 在 2010 年提出的 [4],2013 年又和 Gary Holt, D. Sculley, Michael Young 等人发表了一篇关于 FTRL 工程化实现的论文 [5]。 FTRL 算法融合了 RDA 算法能产生稀疏模型的特性和 SGD 算法能产生更有效模型的特性。它在处理诸如 LR 之类的带非光滑正则化项(例如 1 范数,做模型复杂度控制和稀疏化)的凸优化问题上性能非常出色,国内各大互联网公司都已将该算法应用到实际产品中。
>
> **Code** :
> python版： https://github.com/fmfn/FTRLp
> 多线程版： https://github.com/bingzhengwei/ftrl_proximal_lr

iv. GBDT + LR, 2014



> 2014 年 Facebook 发表了一篇在 CTR 领域极具影响力的论文 [6],该论文尝试提出一种解决特征组合问题的方案,基本思路是利用树模型的组合特性来自动做特征组合,结合 GBDT 训练出一些组合特征,然后再传入 LR 进行分类,该方法取得极大成功,但也有很大程度的过拟合风险,所以必须要采取相应的防止过拟合的措施。
>
> **evaluation** : AUC/NE(Normalized Entropy)/calibration(the ratio of the average estimated CTR and empirical CTR)
>
> **Code** : https://github.com/neal668/LightGBM-GBDT-LR

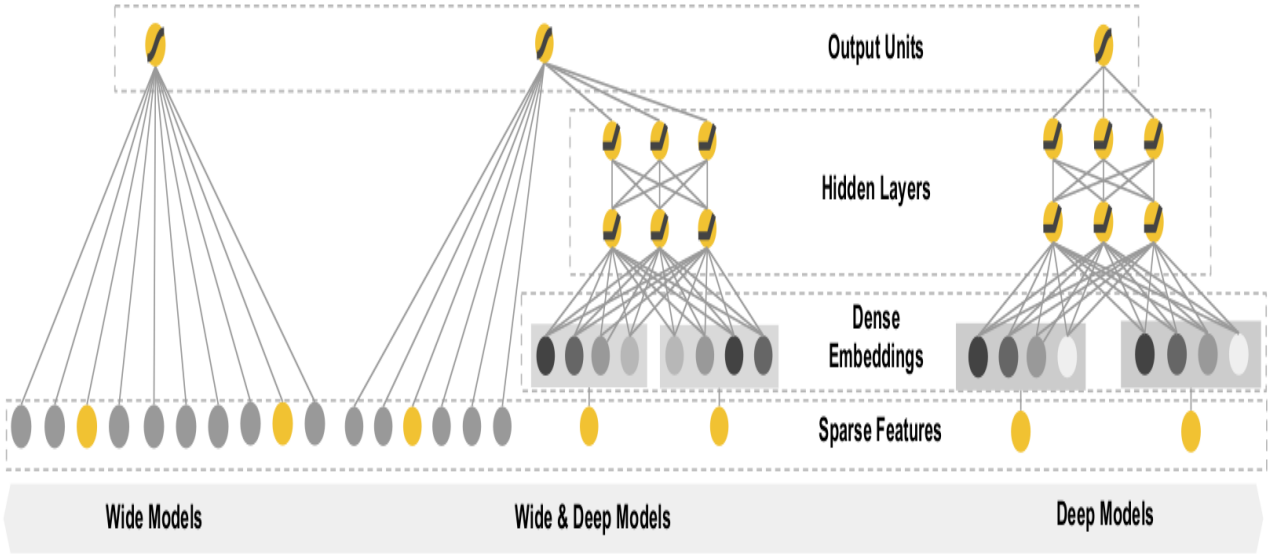
v. RNN(2014)

> 更倾向于序列数据 [7]
>
> **evaluation** : AUC/RIG compare with LR / NN

vi. CNN(2015) [8]

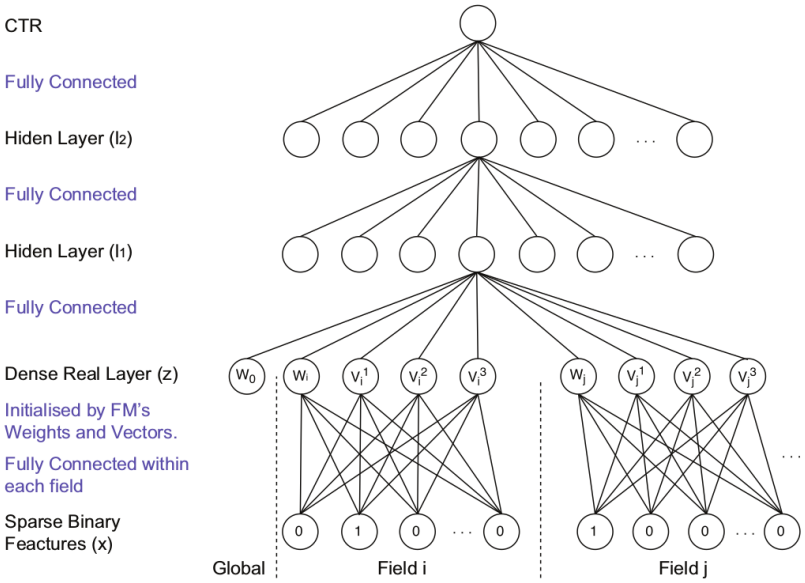
> 更倾向于由相邻特征之间的相互作用而产生的交叉特征.
>
> **dataset** :
> avazu: https://www.kaggle.com/c/avazu-ctr-prediction/data
> Yoochoose: http://recsys.yoochoose.net
>
> **evaluation** : logloss compare with LR / FM / RNN
>
> **Code** : https://github.com/neal668/LightGBM-GBDT-LR

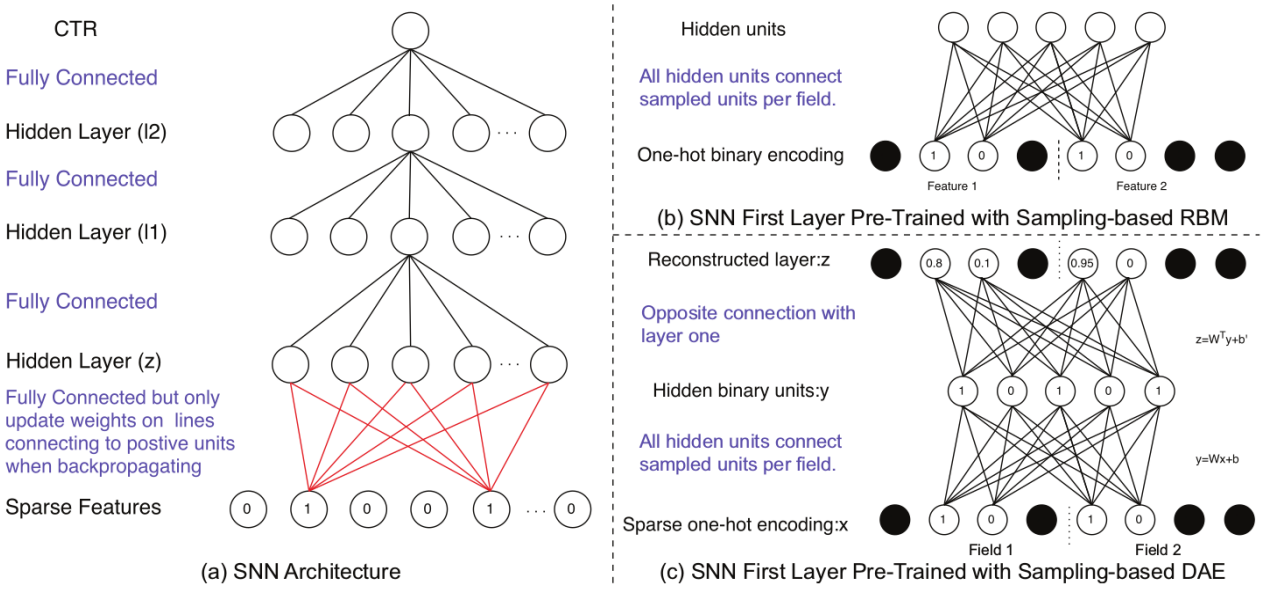
vii. Wide & Deep model(2016):



> 是 2016 年 Google 开源于 TensorFlow 的一种混合网络结构 [2],包括 wide 和 deep 两部分,其中 wide 是一个线性模型,deep 是一个深度模型, 两部分所需的输入不同,wide 依然需要预先的特征工程, 而 deep 部分不需要;其主要缺陷就在于 wide 部分依然需要依赖预先的特征工程；
>
> **dataset** :
> Google Play game center
>
> **evaluation** : AUC / online test compare with wide / deep / wide & deep
>
> **Code** ： wide & deep model:
> https://www.tensorflow.org/tutorials/wide_and_deep

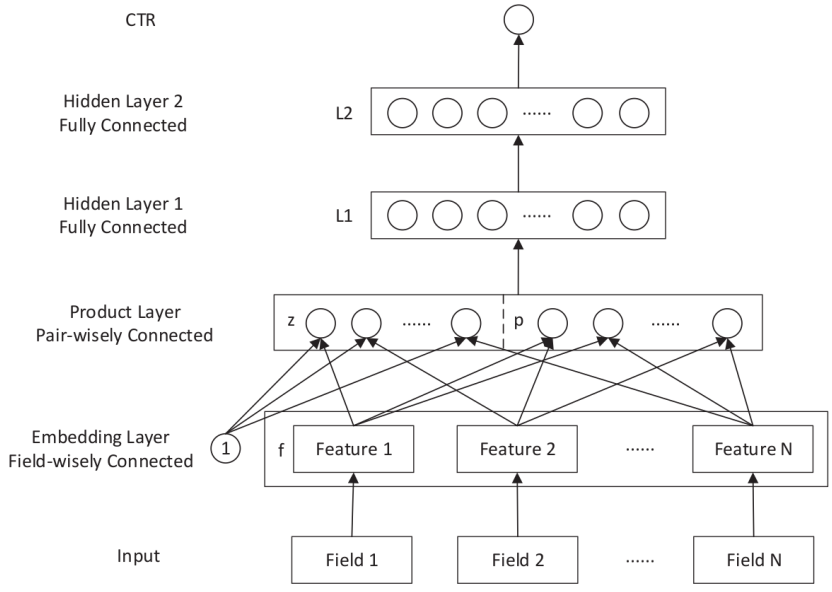
viii. Factorization-machine supported Neural Network(FNN [9], 2016)





> FNN 先训练 FM,然后以 FM 训练的参数对 DNN 进行初始化,继而训练 DNN 模型,该方法的缺陷在于后续 DNN 的训练可能会覆盖掉第一步 FM的预训练成果,从而导致低阶交叉特征的学习能力较差。
>
> **dataset** :
> iPinYou: 19.5M instances(14.79k positive), 937.67K features,
>
> **evaluation** : AUC compare with LR/FM/FNN/SNN*
>
> **Code** ： FNN&SNN: https://github.com/wnzhang/deep-ctr

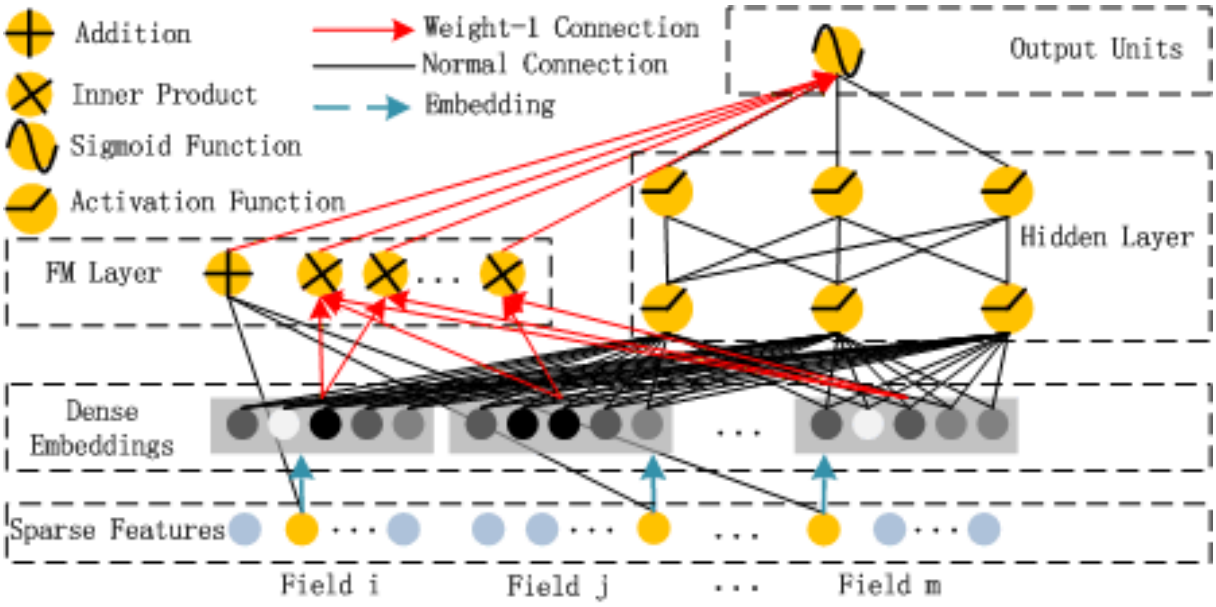
ix. Product-based Neural Network(PNN, 2016)



> PNN(Product based Neural Network)是由上海交通大学的 YaYanru Qu 和伦敦大学的Ying Wen 等人发表的论文 [10]中提出的, PNN 在 DNN 的基础之上作出改进,在 embeddinglayer 和 fully connected layer 之间增加了一个 product layer,使 PNN 具备高阶交叉特征的学习能力。
>
> **dataset** :
> criteo: 1T数据，选取7天训练，1天测试(79.38M instances, 1.64M features)

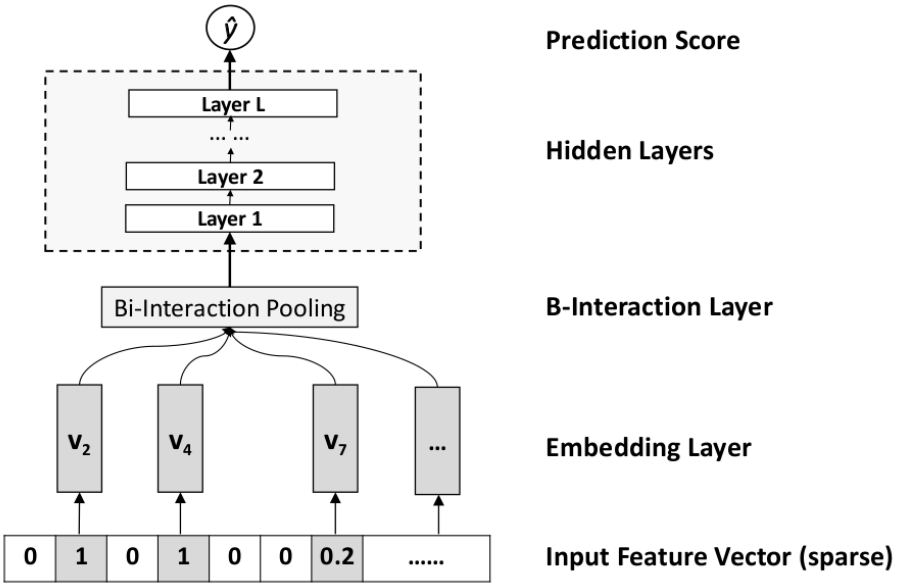
> iPinYou: 19.5M instances, 937.67M features, 一共10天数据，7/3分
>
> **evaluation** : AUC & RIG compare with LR/FM/FNN/CCPM/IPNN/OPNN/PNN*
>
> **Code** : PNN: https://github.com/Atomu2014/product-nets

x. Deep Factorization Machine(2017)



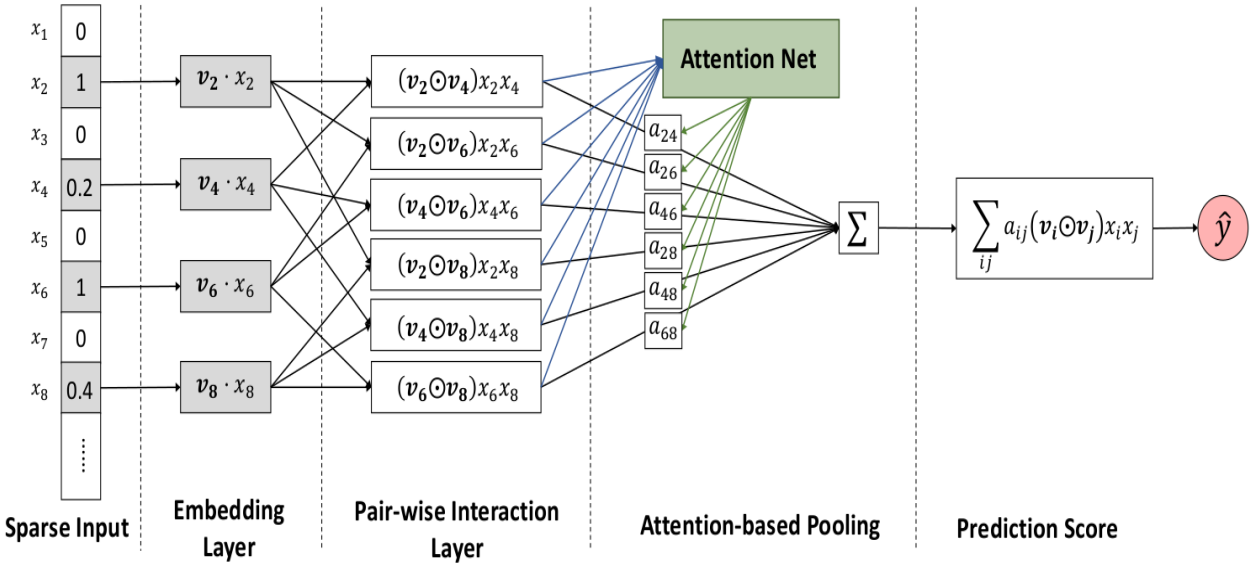
> DeepFM [11]是2017年华为诺亚方舟实验室提出的一种新的FM+DNN模型，该模型主要分为deep和wide两部分，deep模块使用deep learning学习高阶交叉特征，wide模块使用FM提取低阶交叉特征，deep和wide两个模块共享输入和embeddings，最终通过output layer进行结合，输出CTR。
>
> **dataset** ：
> Criteo(4.5million records, 13连续特征, 26离散特征)， 9：1划分；
> Company(1 billion, 7天训练, 1天测试)
>
> **evaluation** ： AUC & Logloss compare with LR/FM/FNN/PNN/WDL/DeepFM
>
> **Code** : DeepFM/CCFNet/BMF: https://github.com/Leavingseason/OpenLearning4DeepRecsys

xi. Neural Factorization Machines(2017)



> NFM [12]是一种新型的DNN，其本质是用DNN实现的更高阶的FM，从而增强了其非线性表达能力。
> **dataset** ：
> Frappe: (288609 instances, 5382 features)；
> MovieLens: (2006859 instances, 90445 features)
> train: validation:test 9:2:1
> **evaluation** ： RMSE compare with FM/High order FM/Wide&Deep/DeepCross

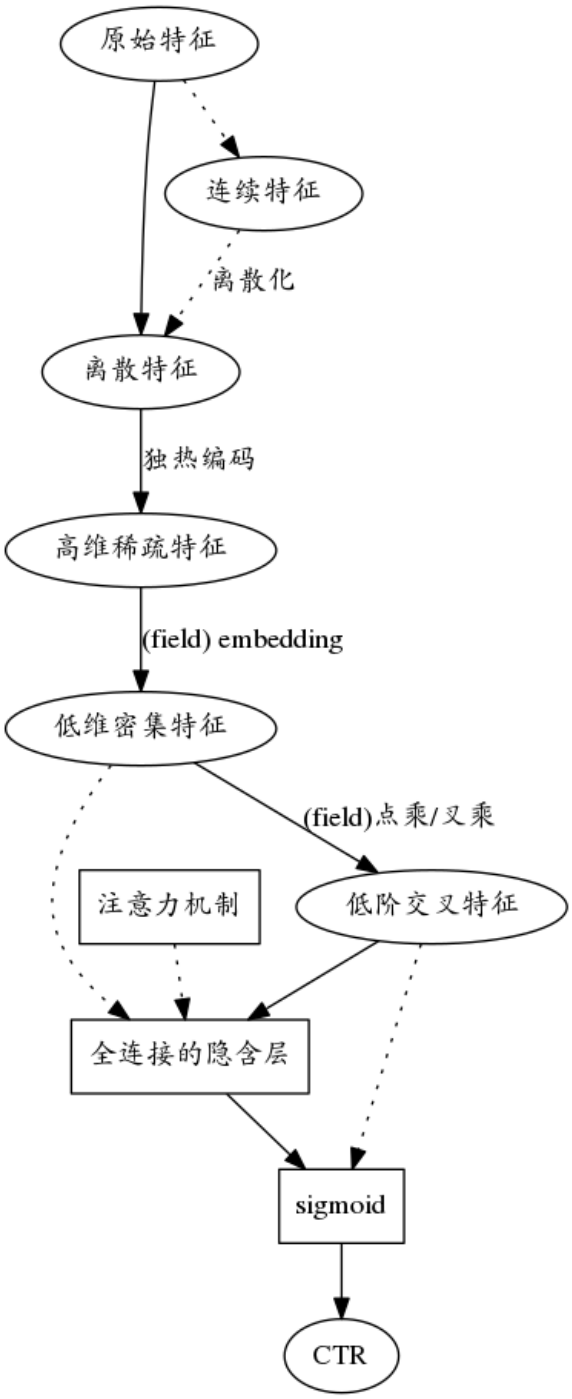
xii. Attention Factorization Machines(2017)



> AFM模型 [13]是NFM的一种改进模型。在传统FM模型中，使用二阶交叉特征得到非线性表达能力，但是不是所有的特征交叉都会有预测能力，很多无用的特征交叉加入后反而会相当于加入了噪声，因此，在AFM模型中，加入了Attention Net机制，旨在去降低噪声特征的干扰。
>
> **dataset** ：
> Frappe: (288609 instances, 5382 features)；

> MovieLens: (2006859 instances, 90445 features)
> train: validation:test 9:2:1
> **evaluation** ： RMSE compare with FM/High order FM/Wide&Deep/DeepCross
>
> **CODE** : https://github . com/hexiangnan/attentional factorization machine

CT预测总体流程



通常， CTR预测包括如下流程：

- 特征预处理
  - 连续特征标准化

    - 连续特征离散化(离散特征更易于提取交叉特征)
    - 离散特征二值化
- embedding layer
    - 借鉴FM/FFM的思路，将每个feature/field转化为embedding向量(降维同时变稀疏为密集)
    - 或者 hash trick(FFM), 但是这种方法，没有embedding的密集优势
- feature interaction
  仅仅是原始特征很难达到好的效果，所以需要挖掘交叉特征，常见的挖掘交叉特征的方法包括：
    - FM
    - FFM
    - GBDT(可以学习到高阶交叉，但缺陷在于不能joint training)
    - Product layer(inner product / outer product)(这里可以借鉴**FFM**的思路)
- hidden layer
  鉴于神经网络强悍的特征表示能力，一般都会在最后加几层全连接层学习难以挖掘的高阶交叉特征
    - 可以添加 attention net
    - 抑制过拟合，可以结合**SNN**的**dropout?**或者**early stopping?**
    - 选择合适的optimizer
- output layer
  output layer一般都是sigmoid function

参考文献

i. Rendle, Steffen. “Factorization machines with libfm.” ACM Transactions on Intelligent Systems and Technology (TIST) 3.3 (2012): 57. ↩

ii. CHENG H-T, KOC L, HARMSEN J, et al. Wide & deep learning for recommender systems[C] // Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. 2016 : 7 – 10. ↩

iii. Jahrer, Michael, et al. “Ensemble of collaborative filtering and feature engineered models for click through rate prediction.” KDDCup Workshop. 2012. ↩

iv. MCMAHAN H B, STREETER M. Adaptive bound optimization for online convex optimization[J]. arXiv preprint arXiv:1002.4908, 2010. ↩

v. MCMAHAN H B, HOLT G, SCULLEY D, et al. Ad click prediction: a view from thetrenches[C] // Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 2013 : 1222 – 1230. ↩

vi. HE X, PAN J, JIN O, et al. Practical lessons from predicting clicks on ads at facebook[C]// Proceedings of the Eighth International Workshop on Data Mining for Online

Advertising.2014 : 1 – 9. ↩

vii. Zhang, Yuyu, et al. “Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks.” AAAI. 2014. ↩

viii. Liu, Qiang, et al. “A convolutional click prediction model.” Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 2015. ↩

ix. ZHANG W, DU T, WANG J. Deep learning over multi-field categorical data[C] // Europeanconference on information retrieval. 2016 : 45 – 57. ↩

x. QU Y, CAI H, REN K, et al. Product-based neural networks for user response prediction[C]// Data Mining (ICDM), 2016 IEEE 16th International Conference on. 2016 : 1149 – 1154. ↩

xi. GUO H, TANG R, YE Y, et al. DeepFM: A Factorization-Machine based Neural Networkfor CTR Prediction[J]. arXiv preprint arXiv:1703.04247, 2017. ↩

xii. HE X, CHUA T-S. Neural Factorization Machines for Sparse Predictive Analytics[J], 2017. ↩

xiii. XIAO J, YE H, HE X, et al. Attentional factorization machines: Learning the weight of feature interactions via attention networks[J]. arXiv preprint arXiv:1708.04617, 2017. ↩

xiv. Steffen Rendle (2012): Factorization Machines with libFM, in ACM Trans. Intell. Syst. Technol., 3(3), May. ↩