Wide & Deep Learning for Recommender Systems

摘要

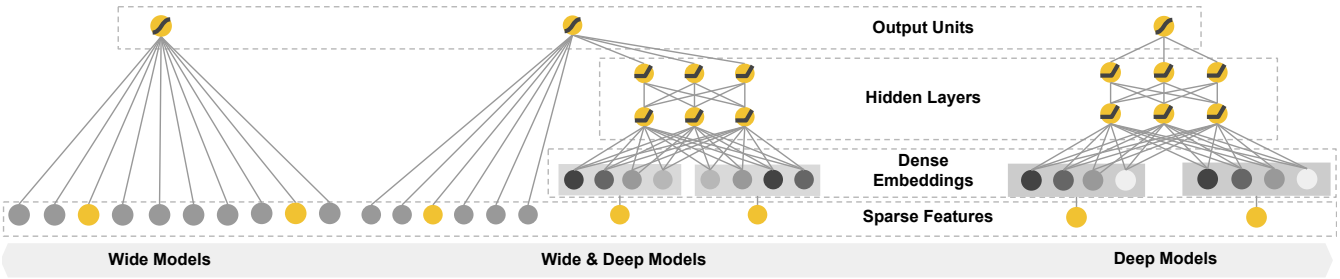
原文地址 https://arxiv.org/abs/1606.07792

- 大量数据的regression/classification
- 减少特征工程
- 同时训练简单的wide linear model和deep NN (embeddings)组合成新的模型

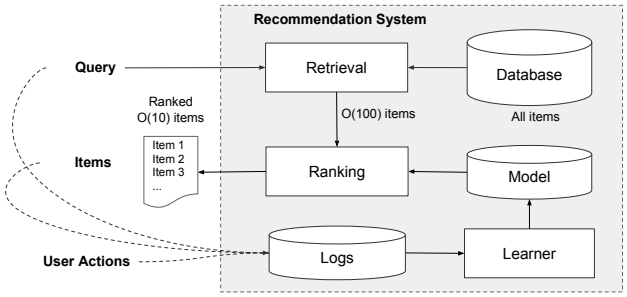
1 推荐系统特性

- memorization 在历史数据上学习/记忆频繁的关联 typical
- generalization 在关联传递的基础上找历史数据里没出现的新特征组合 提升diversity
- massive – sacle online 出于简单选择LR
  - binarized sparse features → one – hot encoding
  - ↑cross product 体现特征关联
  - 无法包含应训练集没出现过的特征对
- embedding – based (*e.g.* factorization machine)
  - 学习query和iterm的低维dense embedding可以用于没见过的特征对
  - query – item matrix稀疏 + 高rank,缺少合适的低维embedding效率表示
  - 例如用户有特殊的偏好,item很小众,与大部分query – item没有交集,dense embedding会导致对所有的query – item给出非零预测，推荐给用户的结果过于general，但cross – product的LR可以避免

2 框架



- embedding → feed – forward NN
- sparse inputs → feature transform → linear model
- 同时训练



- query: various user + contextual features
- ML models + human − defined rules
- rank $P(y|x)$

3 wide部分

cross − product transform

$$\phi_k(x) = \prod x_i^{c_{k_i}} \quad c_{k_i} \in \{0,1\}$$

仅当两个特征都为1 的时候为1

4 deep部分

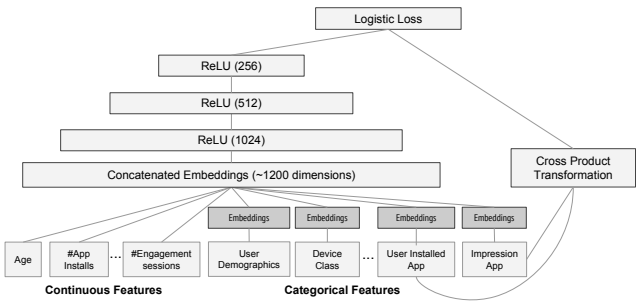
- feature strings: sparse, high-dimensional categorical → dense low-dimensional
  - embeddings $O(10) \sim O(100)$
  - 随机初始化
- forward feed NN

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

  f : activation function = ReLU

5 模型连接

- 用对数的加权和作为预测结果
- 再把结果放进一个logistic loss，



- （这点存疑）不同于ensemble：
  - joint training 同时优化所有参数，每个部分都很小
  - ensemble 分开训练，每个单独的模型都需要大一些，只在训练后结合结果

- 同时BP 来自两个部分的输出的梯度
- 实验配置：deep部分用AdaGrad wide部分用L_1 regularization

6 源码演示

- https://medium.com/tensorflow/predicting-the-price-of-wine-with-the-keras-functional-api-and-tensorflow-a95

这是一个回归问题，这里挑选了

- description 个性化的词汇描述（很长）转换成一系列words vector作为wide feature,
- variety 酒的品类（unique value有限的categorical feature）进行一次one-hot 编码作为wide feature
- description embed 成为 integerized description vectors 作为深度特征