2018 用户兴趣建模大赛解决方案

小小码农队

1、 EDA

1.1 Interaction

⑴总用户个数：37,821；
⑵总视频个数：9,204,230；
⑶训练集总交互数：52,134,536；
⑷测试集总交互数：8,180,616；
⑸训练集中，所有点击和未点击的比例为 1:4，说明问题为类别不平衡问题；
⑹训练集和测试集是同一批用户，训练集和测试集的视频没有交集；
⑺数据具有周期性，一共有四个周期，训练集占三个周期，测试集占一个周期；
⑻训练集中，每个用户的总点击次数最少为 80 次，平均为 278 次，说明给定的这批用户为活跃用户；

1.2 Face

⑴训练集中有人脸的视频数占视频总数的 38.8%；
⑵测试集中有人脸的视频数占视频总数的 38.0%；

1.3 Text

⑴训练集中有文字的视频数占视频总数的 46.3%；
⑵测试集中有文字的视频数占视频总数的 46.5%；

2、 问题思考

给定的数据可以分为两大类，用户和视频的交互信息、视频的内容信息（face,text,visual）。题目没有给出用户侧的信息，可以利用用户和视频的交互信息构建用户社交网络和视频社交网络，并使用 deepwalk 算法训练得到社交网络中个体节点的表征，即为 user.emb 和 photo.emb。对于视频侧的信息，我们要充分利用给定的视频内容信息，face 特征中每个封面的人脸个数不同，可能要人工构造一些类别特征，将每个封面的特征映射到类别特征上；text 特征可以用 tfidf 算法提取；由于 visual 特征比较大，考虑先采样后聚类。由于给定的用户为活跃用户，可以使用统计特征，如平均点击率、平均播放时长。

3、 特征工程

3.1 User.emb 和 Photo.emb

由于社交网络与语言模型存在一定的相似性，我们利用用户和视频的交互信息分别构建用户社交网络和视频社交网络。网络为无向图，对用户社交网络而言，图中的每个节点代表一个用户，如果两个用户被推荐了相同的视频，则两个用户节点有连接边，否则没有连接边；对视频社交网络而言，图中的每个节点代表一个视频，如果两个视频被推荐给了同一用户，则两个视频节点有连接边，否则没有连接边。然后，通过 deepwalk 算法训练得到社交网络中个体节点的表征，即为 user.emb 和 photo.emb。

deepwalk 算法思想：以任意节点为起点构建随机游走路径作为 document，其中随机游走的节点可抽象的认为是 word，生成大量语料训练集，使用 word2vec 算法训练得到

节点的词向量表征。

3.2 Face_Feature

题目给定的人脸信息包含四个属性，人脸占整个图片的比例(proportion)、人脸性别(sex)、人脸年龄(age)、相貌属性(look)。我们将每个属性按照取值进行分段，并统计每个视频对应每个属性分段的人脸个数。这样可以通过这些属性分段特征刻画出用户的偏好，喜欢看长得好看的女性、喜欢看小孩等等。

下面是每个属性的分段，最终得到的人脸特征有 35 维：

proportion: 0.0-0.1,0.1-0.2,0.2-0.3,0.3-0.4,0.4-0.5,0.5-0.6,0.6 以上（共 7 维）

sex: 0,1（共 2 维）

age:0-2,2-3,3-5,5-6,6-8,8-10,10-12,12-13,13-15,15-16,16-17,17-20,20-22,22-25,25-27,27-30,30-33,33-35,35-40,40 以上（共 20 维）

look: 20-36,36-45,45-60,60-75,75-85,85 以上（共 6 维）

3.3 Text_Feature

对于文字描述信息，我们首先通过统计得到停用词，对去除停用词后的文字描述使用 tfidf 算法 (n-gram=1)，再将得到的 tfidf 特征进行非负矩阵分解 (nmf)，设定主题个数为 20，没有文字的视频算作一个主题，从而得到视频封面文字描述的主题分布。

下面详细阐述一下停用词的提取方法：统计每个用户被推荐的有封面文字描述的视频列表，进而统计每个用户的词表，词表包含单词 id (word_id)、用户被推荐视频封面文字描述中包含该单词的视频个数 (photo_num)、单词在用户被推荐视频封面文字描述中出现的频数 (word_fre)，以 photo_num 和 word_fre 为第一和第二关键字对用户词表从大到小排序，取 top100。将所有用户的 top100 单词合并为一个列表 w，在 w 中出现次数大于 nb_users/2 (nb_users 为用户个数)的单词即为停用词，最终的停用词表记为 stop_words。

3.4 视觉特征聚类

在 10% 的视觉特征上训练 kmeans 聚类算法，并对所有视觉特征进行聚类，聚类个数分别为 100 和 500。

3.5 Interaction_Feature

⑴用户侧特征 (4 维)

每个用户的平均点击率、平均播放时长，oof 的统计 (out of fold)；

每个用户被推荐的视频总数，oof 的统计；

每个用户被推荐的视频总数；

⑵视频侧特征 (4 维)

每个视频类别 (100 or 500) 的平均点击率、平均播放时长，oof 的统计；

⑶用户和视频的组合特征 (4 维)

每个用户在每个视频类别 (100 or 500) 的平均点击率、平均播放时长，oof 的统计；

缺失的用每个用户的平均点击率、平均播放时长填充，oof 的统计；

⑷时间戳和作品时长 (2 维)

4、 模型

一共有 8 个模型，模型 1-4 为一组，模型 5-8 为一组，它们是一一对应的（模型 1 对应模型 4，以此类推）。它们的区别在于：⑴训练集和提前停止的方式不同，模型 1-4 按时间序列划分训练集和验证集，取 80%的数据用于训练，使用验证集的 auc 进行提前

停止；模型 5-8 取全量数据集用于训练，使用测试集和模型 1-4 集成结果的 spearman 系数进行提前停止。⑵特征不同，模型 1-4 没有使用 interaction_feature；模型 5-8 在 sigmoid 层前面拼接了 interaction_feature。模型 5-8 的示意图如图 1-4 所示，模型 1-4 只是去掉了图中的 Interaction_feature。

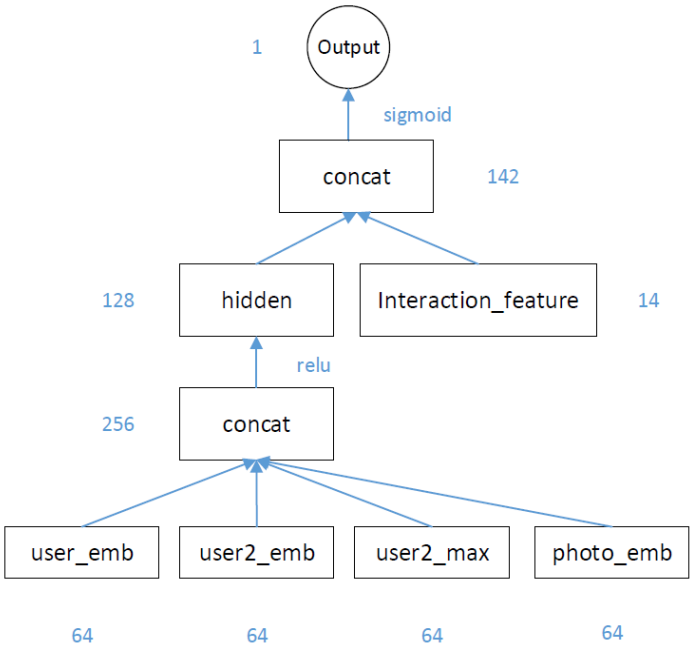

图 1：模型 5

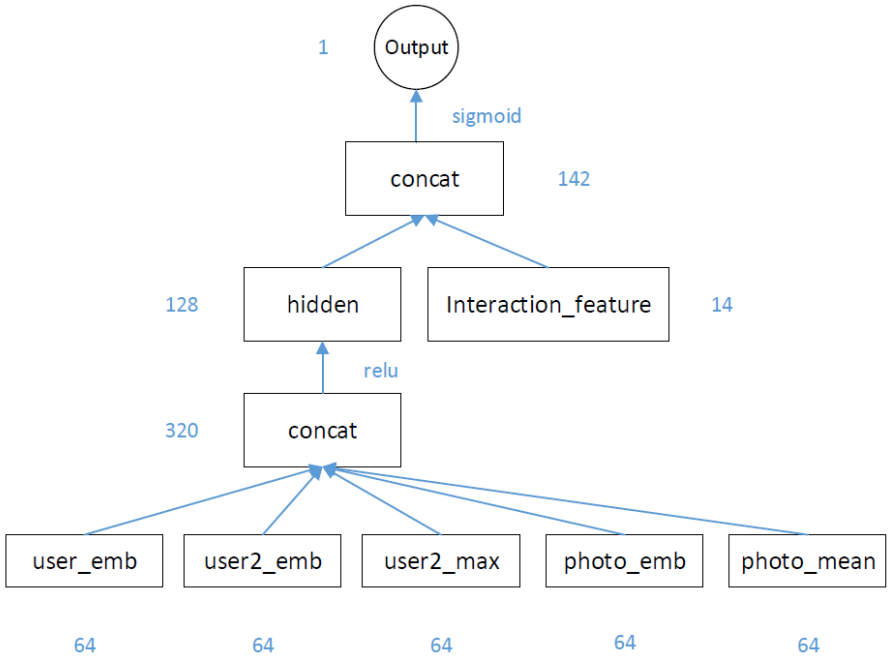

图 2：模型 6

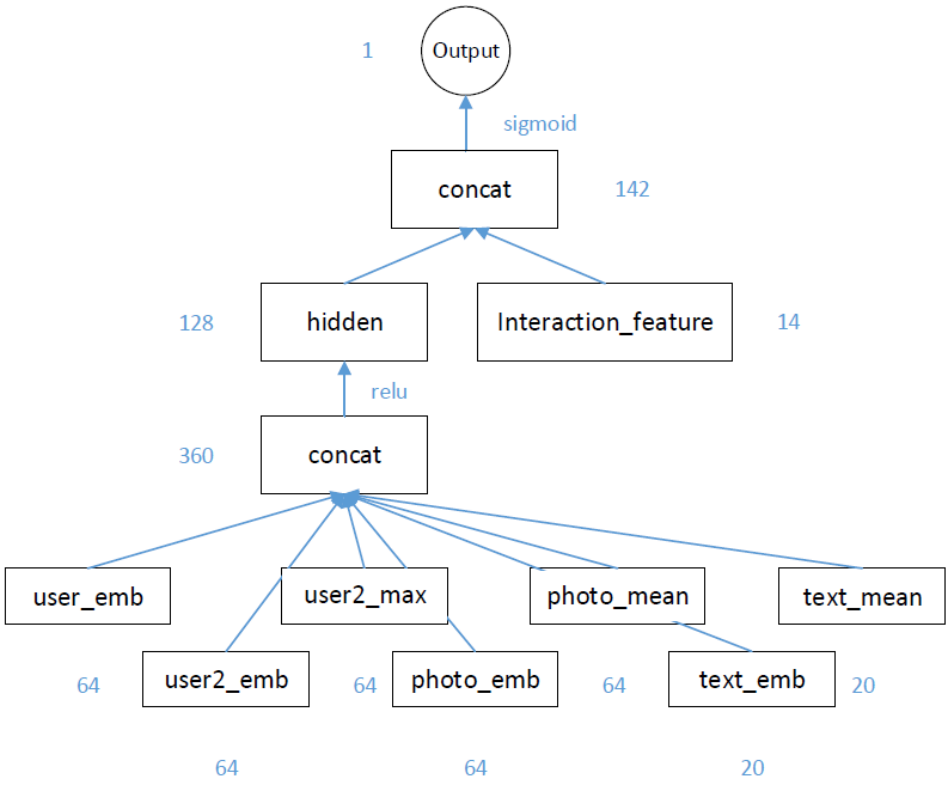

图 3：模型 7

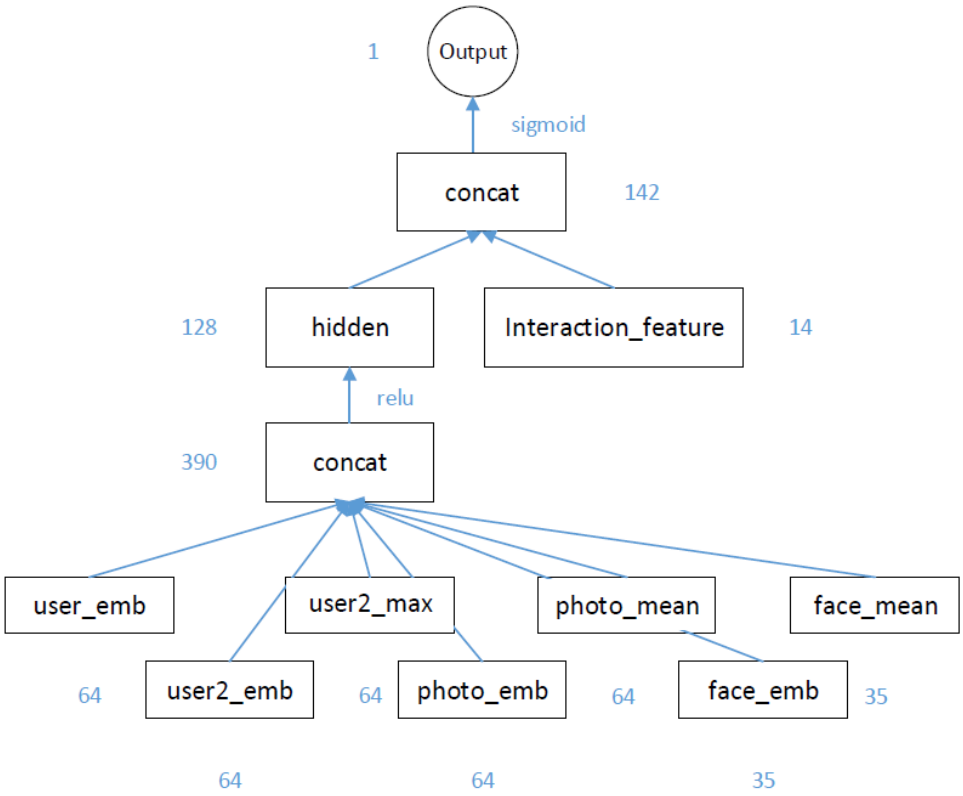

图 4：模型 8

下面对模型中的输入模块进行统一解释：

⑴ user_emb: 见 3.1 节，用户的 embedding 特征，不随网络一起训练，64 维；
⑵ photo_emb: 见 3.1 节，视频的 embedding 特征，不随网络一起训练，64 维；
⑶ user2_emb: 使用均匀分布随机初始化的用户 embedding 特征，随网络一起训练，64 维；
⑷ face_emb: 见 3.2 节，人脸属性分段特征，不随网络一起训练，35 维；
⑸ text_emb: 见 3.3 节，视频封面文字描述的主题分布，不随网络一起训练，20 维；
⑹ user2_max: 每个视频对应的用户列表，先使用 user2_emb 转换为 embedding 形式，再做 GlobalMaxPooling1D，代表了和当前用户最相似用户的 embedding，64 维；
⑺ photo_mean: 每个用户都有一个按时间戳排列的被推荐视频列表，取每个用户位于当前视频前面和后面的 15 个视频，先使用 photo_emb 转换为 embedding 形式，再做 GlobalAveragePooling1D，代表了用户的兴趣，64 维；
⑻ face_mean: 每个用户都有一个按时间戳排列的被推荐视频列表，取每个用户位于当前视频前面和后面的 15 个视频，先使用 face_emb 转换为 embedding 形式，再做 GlobalAveragePooling1D，代表了用户的兴趣，64 维；
⑼ text_mean: 每个用户都有一个按时间戳排列的被推荐视频列表，取每个用户位于当前视频前面和后面的 15 个视频，先使用 text_emb 转换为 embedding 形式，再做 GlobalAveragePooling1D，代表了用户的兴趣，64 维；
⑽ Interaction_feature: 见 3.5 节，根据用户和视频的交互信息计算出来的统计特征，14 维；

我们线上的最终结果是模型 1-4 的集成结果和模型 5-8 的集成，线上最终成绩为 0.75241018。模型 1-4 的集成结果取得的线上成绩为 0.74910139，模型 5 取得的线上成绩为 0.74886836。

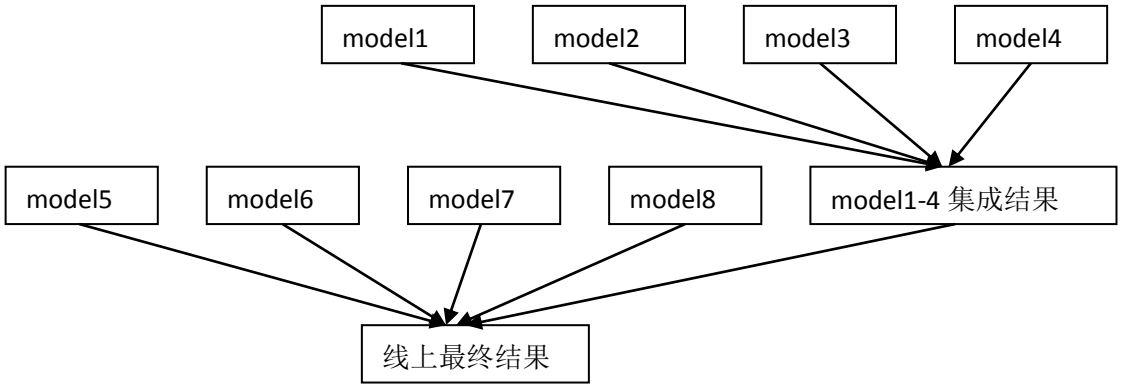

图 5：模型集成

5、 总结

我和队友的研究方向都不是数据挖掘和推荐系统，初赛阶段我们费了很大力气搞了个 rnn 模型对用户历史兴趣进行建模，线上结果 0.55，完全就是随机呀！随后通过看群里各方大佬的讨论，发现可以使用规则、统计，要好好挖掘特征，还有什么 lr、fm、ffm、xgboost，于是我们开始上网恶补机器学习、推荐系统方面的知识。至此，彻底抛弃之前的 rnn 模型，开始致力于挖掘视频的内容特征，这算是从一个大坑跳进了一个小坑，想法很多，能实现的很少，实现了在线下验证集上有用的更少，最终线上提分的简直是九牛一毛。虽然比赛的过程很艰难，但我们从未想过放弃，最终也算是取得了一个比较满意的成绩。最后非常感谢中国多媒体大会、北京快手科技有限公司共同举办的这次大

赛，让我们在算法、数据、编程能力上都有了提升，希望以后能继续参加这样的比赛，生命不息，比赛不止。