概览

这个项目使用了 tensorflow 框架，对用户数据和内容数据进行特征提取，保存用户特征和内容特征到文件，然后进行基于用户相似性、基于内容相似性和用户所喜欢类型的推荐。写这个报告的目的是整理一下基于 tensorflow 推荐算法的流程，以及总结一下可以借鉴的地方。

数据预处理

这个算法使用了 movielens 这个数据集，里面包括了用户数据、内容数据和用户行为数据。对应到 minitrill 中的数据中，也需要用户数据、视频数据和用户行为数据，但是里面的数据有所区别。

用户数据

数据中的格式：UserID::Gender::Age::Occupation::Zip-code

- Gender is denoted by a "M" for male and "F" for female
- Age is chosen from the following ranges:
  - 1: "Under 18"
  - 18: "18-24"
  - 25: "25-34"
  - 35: "35-44"
  - 45: "45-49"
  - 50: "50-55"
  - 56: "56+"

可以看到上面的数据格式中，分别有用户 ID、性别、年龄、职业和邮政编码，但是在用户的特征，不需要邮政编码。在数据预处理的时候，性别转化为 1 和 0，分别表示 M 和 F；然后年龄转化为 0 到 7 的数据分别代表不同的年龄段。而这个数据集中，有 20 种职业，也分别用 20 个数字来表示，这个已经不需要转换了，数据集中已经有了表示。

每一个数据都会经过一个嵌入层，转换成为各自代表的特征：

- 性别特征
- 年龄特征

- 职业特征
- 用户 ID 特征

- Occupation is chosen from the following choices:
  - 0: "other" or not specified
  - 1: "academic/educator"
  - 2: "artist"
  - 3: "clerical/admin"
  - 4: "college/grad student"
  - 5: "customer service"
  - 6: "doctor/health care"
  - 7: "executive/managerial"
  - 8: "farmer"
  - 9: "homemaker"
  - 10: "K-12 student"
  - 11: "lawyer"
  - 12: "programmer"
  - 13: "retired"
  - 14: "sales/marketing"
  - 15: "scientist"
  - 16: "self-employed"
  - 17: "technician/engineer"
  - 18: "tradesman/craftsman"
  - 19: "unemployed"
  - 20: "writer"

Movielens 中的数据仅仅包含了人口统计学信息。而 minitrill 项目中，用户数据有多个部分组成：人口统计学信息、社交信息、兴趣信息，在提取用户特征的时候，可以在建立多个层来表示这些不同类别的数据。

所以在拥有用户行为数据的时候，就可使用深度网络训练求得用户特征，可以生成比较小的网络参数，代表了用户的特征。但是对新用户的冷启动问题就很难解决了，因为没有对应的行为数据可以用于训练，求出用户特征。

内容数据

数据中的格式：MovieID::Title::Genres

- Titles are identical to titles provided by the IMDB (including year of release)
- Genres are pipe-separated and are selected from the following genres:
  - Action
  - Adventure
  - Animation
  - Children's
  - Comedy
  - Crime
  - Documentary
  - Drama
  - Fantasy
  - Film-Noir
  - Horror
  - Musical
  - Mystery
  - Romance
  - Sci-Fi
  - Thriller
  - War
  - Western

内容数据要处理的只有 title 和 genres 两个部分的数据，而这 2 个数据都是文本数据。类型是有限集合，那么就可以很简单转换成为数字集合。因为电影的类型是多样的，不是唯一的，所以需要一个列表来存储这些数据，如何拥有这个类型，那么就在上面填写这个类型的数字索引，如果没有，那么用一个约定的字符来填写空白。

Minitrill 中的视频数据除了类型或者标签，还有一些统计数据，例如点赞数量、转发数量等等，可以反应视频的热度、新颖性等等的信息，那么这些数据属于不同的特征，应该也需要分开处理。

用户行为数据

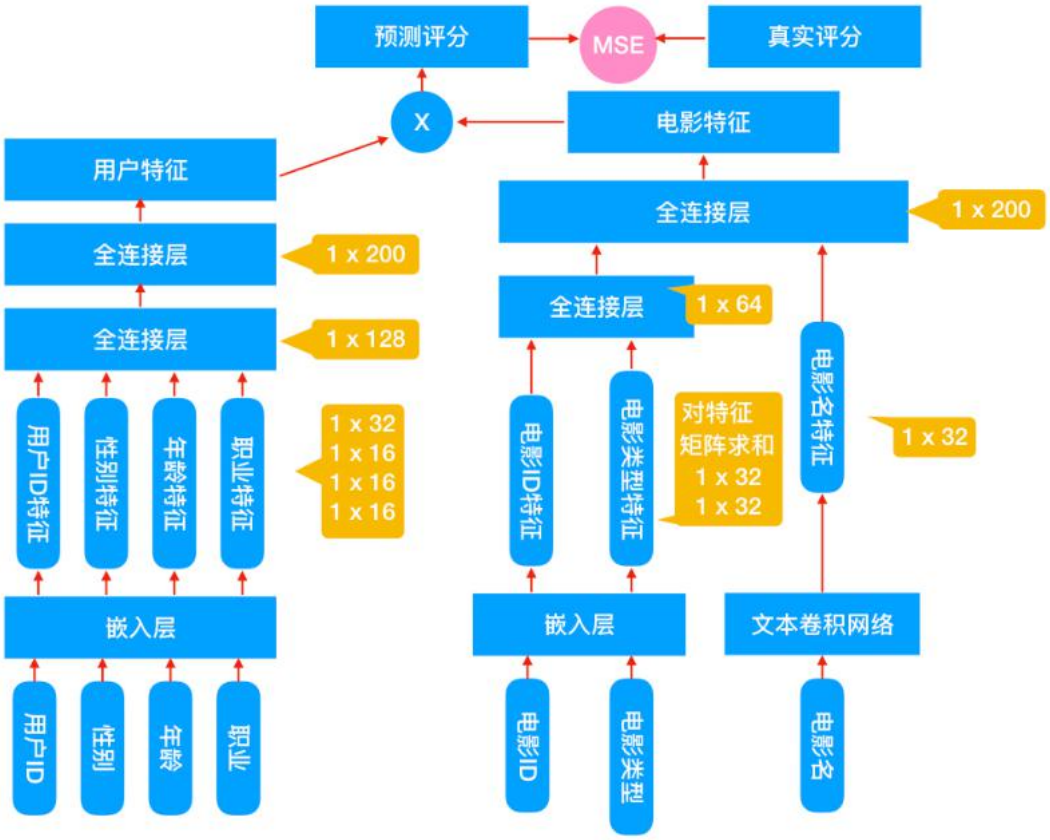
Movielens 的用户行为数据包含了用户 ID、电影 ID、评分和时间戳，这个深度网络的训练依赖于用户行为数据，因为用户行为数据相当于将用户和内容关联在一起，反应出了用户画像和内容画像。而在这个开源项目中，不使用时间戳，仅使用用户 ID、电影 ID 和评分，用评分来作为 Y，优化损失函数，让用户特征、内容特征的关联接近评分。

	UserID	MovieID	Rating
0	1	1193	5
1	1	661	3
2	1	914	3
3	1	3408	4
4	1	2355	5

而 minitrill 中可收集到的用户行为数据有点赞、评论、转发、观看时间等等的信息，可以从多个维度表示用户和视频的特征。我们可以从 2 个方面来思考一下，用户产生这些行为的想法：

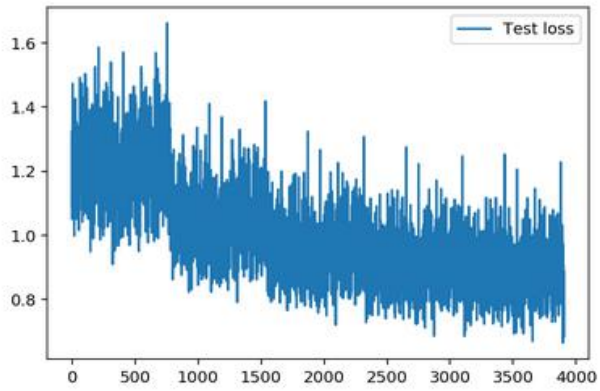
- 喜爱：点赞、赞美的评论、转发，以及长的观看时间，都能够从某种程度代表了用户对这个视频的喜爱程度；
- 讨厌：如果用户不点赞，出现了恶意、厌恶的评论，转发给朋友进行批评性的讨论，那么则代表了用户不喜爱这个视频。但是可能又不代表用户讨厌这类视频。

特征提取



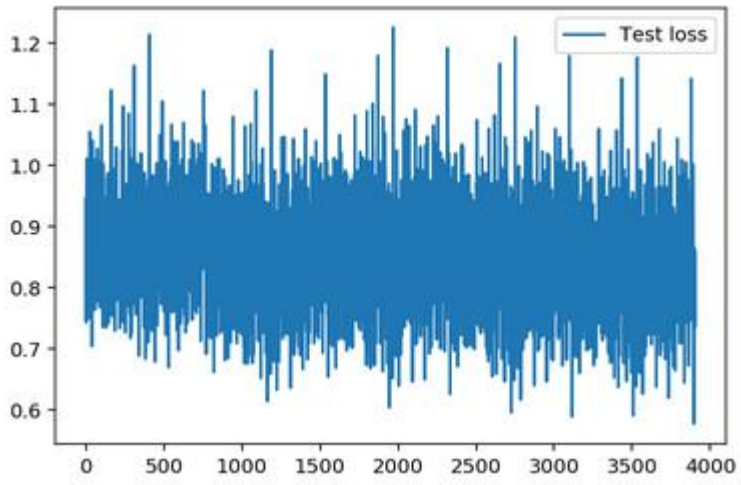
这个项目可以在笔记本的配置训练，所以会在预处理的时候将数据都转换成代表数字，然后输入嵌入层，维度是（1，32）和（N，32），减小了计算量。将特征输入到全连接层，最后都成为（1，200）的用户特征和电影特征。

使用 MSE（最小二乘）计算损失，然后进行迭代优化，训练出用户特征和电影特征，存到文件中。在训练的时候，按照原来的 learning_rate 为 0.0001，测试集的损失会出现非线性递减的特性：



```
(20, 3883)
[1133 1194 1133 1133 3269 1133 3269 1133 1133 3269 1133 3269 3269 1133
 1133 3269 3269 3269 3269 1133]
```

在推荐相似用户所喜欢的电影时只有 2 个，所以这一部分是需要注意的，测试集的损失的均值要集中在一个区间内。修改了学习率 learning_rate 后，就能够正常推荐了，计算出正确的用户特征和电影特征。



所以在以后的 minitrill 用户特征和视频特征的训练都需要注意测试集上损失的均值要保持在一个范围内，不能递增也不能递减。同时在深度网络的设计上要借鉴这个项目。