**文件目录结构说明**

**根目录下文件夹说明：**

**Datasets**

该目录主要存放原始数据经过预处理后的训练数据

该目录下有四个文件其中：

**History.mat**为VM-ip通讯矩阵

**Cluster.mat** 为VM的相似邻居VM的矩阵。在我们的实验中，每一个VM为其输入了30个邻居节点。

**Dataset.pkl** 为经过预处理的训练集，验证集以及测试集

**List.pkl** 为经过预处理的每一个VM的历史交互信息以及相似VM集合

**Output**

该目录主要存放模型训练过程中产生的数据，包括：

**Output.txt** train loss 与 validation loss 的记录文件

**Best\_checkpoint.pth.tar** 训练过程中最优loss对应的参数保存

**Latest\_checkpoint.pth.tar** 训练过程中最后一个epoch对应的参数保存

**根目录下python函数说明：**

**preprocess.py**

数据预处理，包括从History.mat提取历史信息，以及从Cluster.mat中提取聚类信息。同时，按照指定的比例划分训练集，验证集以及测试集

**model.py**

推荐模型主要实现代码，主要实现了三个类：

\_UserModel: VM的模型表达

\_ItemModel: IP\Config的模型表达

ConfigRec: 利用VM建模与Config建模，输出0-1的分数。分数越高证明越有可能被推荐

另外包含两个辅助类：

\_MultiLayerPercep：三层MLP的实现代码

\_Aggregation：聚合层的实现代码

**utils.py**

包含batch内数据采样的辅助函数，其中实现了batch内负采样

**dataloader.py**

划分数据集的辅助函数，继承torch.utils.data.Dataset类，用于batch内数据采样，调用了utils.py中的函数

**main.py**

代码的主函数，实现了训练以及验证，以及测试的步骤。需要先运行preprocess.py 预处理函数

**代码执行流程：**

* python preprocess.py –-test\_prop 0.1

预处理代码，命令行参数test\_prop为测试集的划分比例，该函数会读取datasets中的history.mat与 cluster.mat的数据，并将处理结果转化为pickle文件也存储在datasets文件夹中，用于后续的模型训练

* python main.py -- epoch 30 -- lr 0.0001 --embed\_dim 64 –threshold 0.5

模型训练代码，命令行参数包括epoch的次数，学习率，embedding的大小以及threshold的大小。该函数每一个batch会输出当前loss，并在每100个batch将当前loss与最优loss记录在文件中。

每一个epoch结束后，都会将当前的参数以及最优loss对应的参数保存在output文件夹中

* python main.py -- test --embed\_dim 64 –threshold 0.5

模型测试代码，命令行参数表示当前为测试部分，不进行训练。需要输入对应的embedding size 与 threshold 参数。在测试时，会将输出的结果与threshold进行比较，如果结果大于threshold则推荐该样本。并计算总体的测试准确率，召回率，F1值以及总推荐的样本数。