## 文件说明

材料中的压缩包主要包括两大部分：一个为code\_project文件夹，另外一个就是说明文档。其中，code\_project文件夹中包括三个部分：code文件夹，environment.yaml文件，以及README.md文件。其中，code文件夹存放比赛用到的源代码，environment.yaml文件结合anaconda可以创建代码所需的虚拟环境，README.md文件是对文件结构、环境配置、代码运行等方面的说明。对于code这个文件夹，该文件夹下的**generate.sh**为主脚本，想要复现结果，运行该脚本即可。

## 依赖环境

本代码当时运行的硬件环境如下（仅供参考）：

CPU：Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz

GPU: NVIDIA 1080Ti 11GB\*4

RAM：32GB

软件运行环境如下：

操作系统：Ubuntu 18.04.3 LTS

编程语言：Python 2.7（主体）/Python 3.6（极小部分代码用到）

环境配置：Anaconda 4.5.4 / 4.8.3

部分环境依赖如下：

python=2.7.18,pytorch=1.0.0,cudatoolkit=10.2.89,torchvision=0.2.2,sympy=1.5,pandas=0.24.2,scipy=1.2.1等等

**[推荐]** 我们制作了一个环境配置文件，可以基于anaconda，使用environment.yaml来快速创建复现实验结果所需的环境，具体指令如下：

conda env create -f environment.yaml

创建了这个环境后，通过输入以下指令，进入该环境。

conda activate epi\_gpu

**注意：**如果创建上述环境过程中遇到了prefix相关的问题，请考虑修改environment.yaml文件中的name内容或者prefix路径，使之与您本地的路径保持一致。

## 运行说明

运行“code\_project/code/generate.sh”脚本即可复现结果，进入到上述目录后，输入如下指令：

chmod +x ./generate.sh

./generate.sh

其中最后的结果文件存放在./predict/submission\_final.csv中

**注意：**

1. Markdown格式的代码说明在“**code\_project/code/README.md**”中，有关依赖环境、代码运行、注意事项等内容，也可以参考上述README.md文件。

2. 在复现实验结果的时候，使用的是已经生成好的邻接矩阵，不需要人为再生成，只需要激活environment.yaml文件中的依赖即可复现结果。

3. 如果需要单独生成邻接矩阵，由于使用到的依赖与environment.yaml环境中的依赖存在冲突，需要创建另外一个环境该环境的相关环境依赖如下：

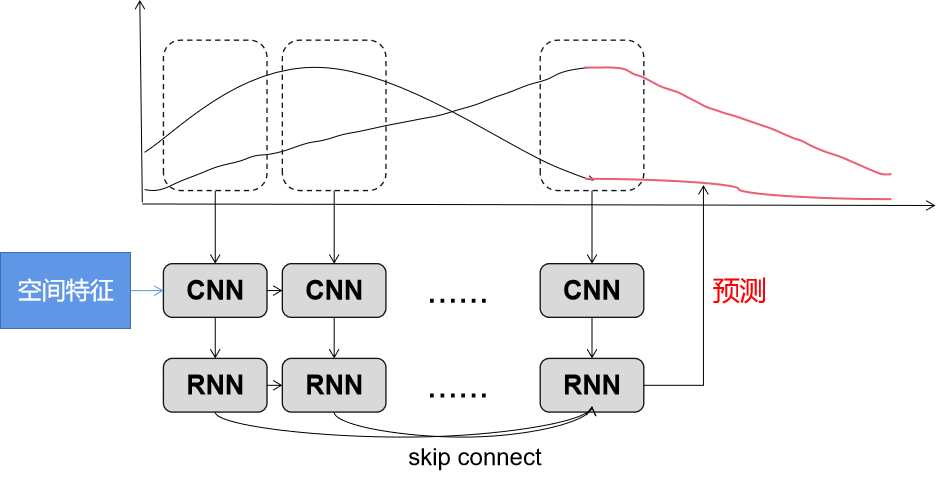
Python=3.6, pandas=1.0.3, numpy=1.18.2, scipy=1.4.1

在新建了以上新的环境后，可以通过调用并运行“code/adj\_generate”目录下的“final\_adjacent\_matrix.py”文件，得到邻接矩阵，对应的邻接矩阵文件保存在“code/adj\_generate/src4”文件夹中。

## 主要思路说明

我们的思路主要是利用空间和时间的多尺度特征来进行预测，其中density数据用于分析城市的管控情况；transfer和migration用于分析城市间的联系迁徙情况；grid\_attr用于分析各区域的地理邻接情况；这些特征与infection时间特征进行多尺度融合，实现基于空间和时间的多尺度特征预测。

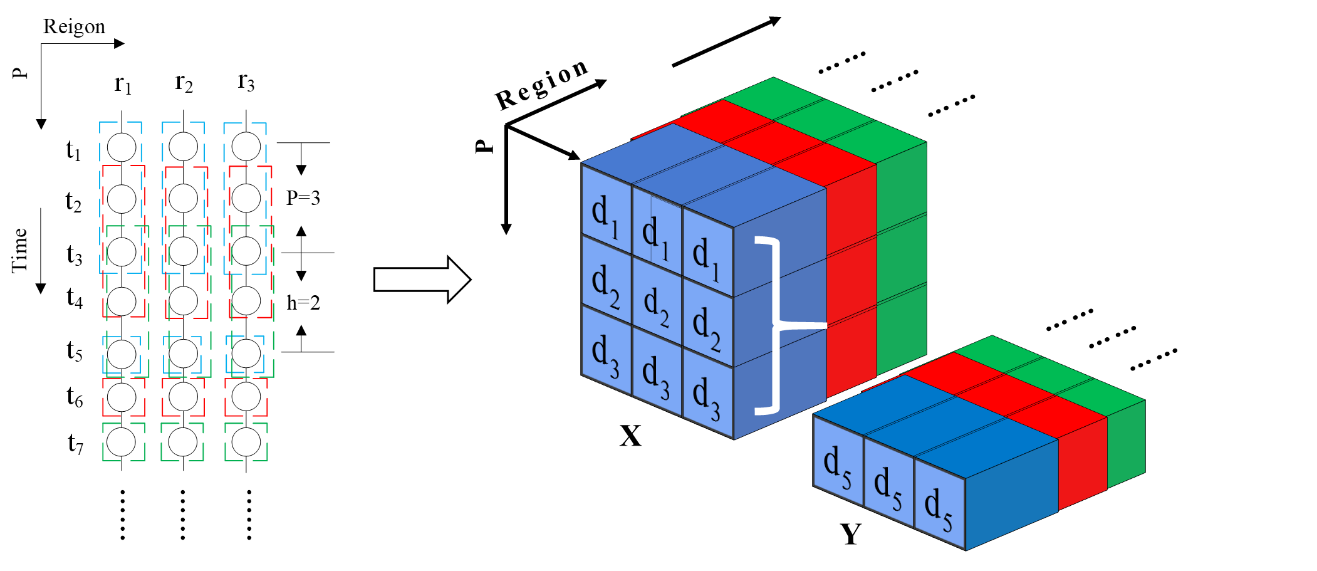
基于上述思路，本模型主要采用CNN+RNN的预测方式（如下图所示）。其中CNN网络用于提取各个区域之间空间上的关联特征，并将空间特征与时间数据进行融合。RNN用于将融合后的多尺度特征进行时间序列上的预测。辅助模型为AR时间序列自回归模型等。



对于空间特征的构造，我们将城市与城市之间的关联基于人口迁徙指数，分析每个城市人口流动最高的区域作为每个城市的枢纽点，类似大型交通集散区等，作为城市之间直接关联的首发区域。对于density管控不同的城市，我们采取不同的枢纽开放比例（如下图所示）。

|  |  |
| --- | --- |
|  |  |
|  | 8260100a221ec5b756e41cef412b37a |

模型的输入数据一共有两组，一组是反应空间特征关系的邻接矩阵，另一组是时间数据，也就是每个地区每天的病例新增情况。利用滑动窗口的方式，每次取一个窗口大小的时间数据进行训练，最终，得到一个关于未来30天流行病情况的预测。



# 参考&引用业界的论文

1. Yuexin Wu et al., 2018. Deep Learning for Epidemiological Predictions. SIGIR’18, July 8-12, 2018, Ann Arbor, MI, USA.
2. Balachandar Vellingiri et al., 2020. COVID-19: A promising cure for the global panic. Science of the Total Environment.
3. Alex Krizhevsky et al., 2017. ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM.
4. Kyunghyun Cho et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014).
5. Vinay Kumar Reddy Chimmula et al., 2020. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. Chaos, Solitons and Fractals.
6. Bing Yu et al., 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence.
7. Dey, Rahul & Salem, Fathi. (2017). Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks.
8. Kingma D P, Ba J A. A method for stochastic optimization. arXiv 2014[J]. arXiv preprint arXiv:1412.6980, 2019, 434.
9. Bengio Y, CA M. Rmsprop and equilibrated adaptive learning rates for nonconvex optimization[J]. corr abs/1502.04390, 2015.