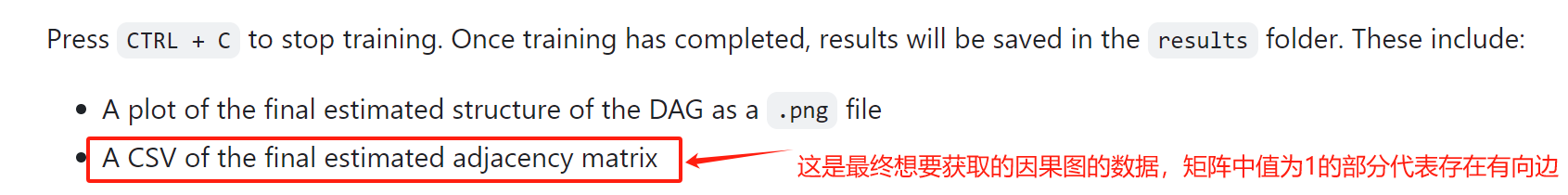
1、选定结局之后的前top个元素一起组成的df，将其存到 /backend/DAG\_from\_GNN\_main目录下的datasets文件夹[./backend/DAG\_from\_GNN\_main/datasets]下，命名为'causal\_related\_data.csv'

df.to\_csv('causal\_related\_data.csv', index=False)

2、更新获取了上述'causal\_related\_data.csv'之后，运行dag-gnn



import subprocess

import signal

import time

from django.http import JsonResponse

def run\_command\_with\_timeout(command, timeout\_seconds):

# 此处因为dag-gnn代码不会自动停止，所以需手动设置定时启用 Ctrl+C 以停止代码运行，获取最终的数据

process = subprocess.Popen(command, stdout=subprocess.PIPE, stderr=subprocess.PIPE, cwd='./DAG\_from\_GNN\_main', universal\_newlines=True)

try:

# 等待命令执行

process.wait(timeout=timeout\_seconds)

except subprocess.TimeoutExpired:

# 如果超时，发送 Ctrl + C 信号

process.send\_signal(signal.SIGINT)

time.sleep(1) # 等待一秒钟确保处理 Ctrl + C 信号

# 终止进程

process.terminate()

# 等待进程退出

process.wait()

# 读取输出结果

result = process.communicate()

return result

def your\_view(request):

# 此处的your\_view指的是获取因果图时的那个函数

try:

# 调用本地的Python脚本

# 使用方式

result = run\_command\_with\_timeout(['python', '-m', 'DAG\_from\_GNN'], timeout\_seconds=60) # 此处的60待定，因为实际运行时间远远长于这个时间

print(result)

except Exception as e:

print(f"Error: {e}")

# 获取生成的 final\_adjacency\_matrix.csv，值为1 的代表有有向边存在

dataset = pd.read\_csv('final\_adjacency\_matrix.csv')

# 该矩阵类似于dagma算法中的结构模型 W\_est



以上两步之后，可得到经过 dag-gnn 算法计算而得到的因果图的节点和有向边；