Enhancing the accuracy of Autism detection using fMRI images with Graph Autoencoder and Graph Neural Network

MSc Research Project
Data Analytics

Sumit Rai
Student ID: x20207603

School of Computing
National College of Ireland

Supervisor: Hicham Rifai

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sumit Rai
Student ID:	x20207603
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Hicham Rifai
Submission Due Date:	15/08/2022
Project Title:	Enhancing the accuracy of Autism detection using fMRI images with Graph Autoencoder and Graph Neural Network
Word Count:	670
Page Count:	5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	15th August 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing the accuracy of Autism detection using fMRI images with Graph Autoencoder and Graph Neural Network

Sumit Rai
x20207603

1 Introduction

This document gives a detailed description of the steps that need to be followed to run the code and replicate the results mentioned in the project report. It covers the tools required for the code execution.

2 Environmental setup

The research is about modelling the deep neural networks for enhancing the accuracy of classification of Autism using resting-state functional magnetic resonance imaging (rs-FMRI). The programming language used is Python and Pytorch library is used for the modelling. The model is trained on a laptop computer with the following configuration:

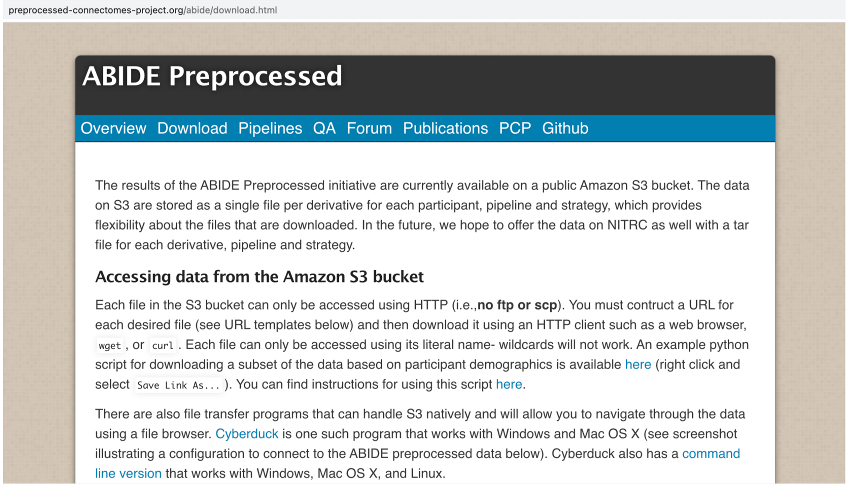
Table 1: System configuration

Operating System	MacOS Monterey Version 12.5
RAM	16 GB
CPU	Apple M1 Pro

3 Data preparation

The preprocessed neuroimaging publicly open source dataset known as ABIDE-I dataset is to be downloaded from the web which is available at the link ABIDE - I Preprocessed as part of part of the Preprocessed Connectome Project (PCP).

The dataset is available as a set of subject files and a phentotypic file having information on subjects' phenotypic attributes. The subject files are available as tab delimited files and are 1101 in number. The additional preprocessing is performed automatically by the code during execution.



preprocessed-connectomes-project.org/abide/download.html

ABIDE Preprocessed

Overview Download Pipelines QA Forum Publications PCP Github

The results of the ABIDE Preprocessed initiative are currently available on a public Amazon S3 bucket. The data on S3 are stored as a single file per derivative for each participant, pipeline and strategy, which provides flexibility about the files that are downloaded. In the future, we hope to offer the data on NITRC as well with a tar file for each derivative, pipeline and strategy.

Accessing data from the Amazon S3 bucket

Each file in the S3 bucket can only be accessed using HTTP (i.e.,**no ftp or scp**). You must contruct a URL for each desired file (see URL templates below) and then download it using an HTTP client such as a web browser, `wget` , or `curl` . Each file can only be accessed using its literal name- wildcards will not work. An example python script for downloading a subset of the data based on participant demographics is available here (right click and select `Save Link As...`). You can find instructions for using this script here.

There are also file transfer programs that can handle S3 natively and will allow you to navigate through the data using a file browser. Cyberduck is one such program that works with Windows and Mac OS X (see screenshot illustrating a configuration to connect to the ABIDE preprocessed data below). Cyberduck also has a command line version that works with Windows, Mac OS X, and Linux.

Figure 1: ABIDE Preprocessed

4 Code execution

4.1 Unzip the code

Before the code can be run, it need to placed at a path and unzipped. Extract the zip x20207603.zip at a path.

```
    > unzip x20207603.zip
```

The unzipped folder x20207603.zip contains two folders Research_Project_x20207603.zip for first model and Research_Project_x20207603_II.zip for second model which also need to be unzipped.

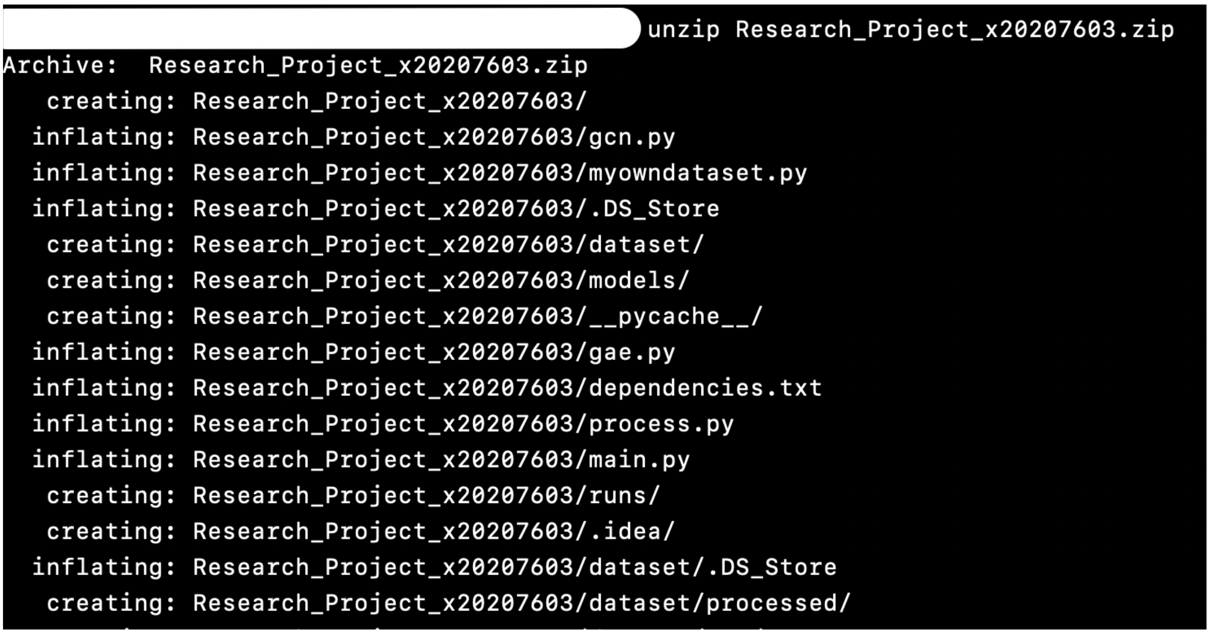
```
    > unzip Research_Project_x20207603.zip
> unzip Research_Project_x20207603_II.zip
```

Each of them contains the following folders

- dataset
- models
- runs

and following files:

- dependencies.txt
- gcn.py
- main.py
- myowndataset.py
- process.py



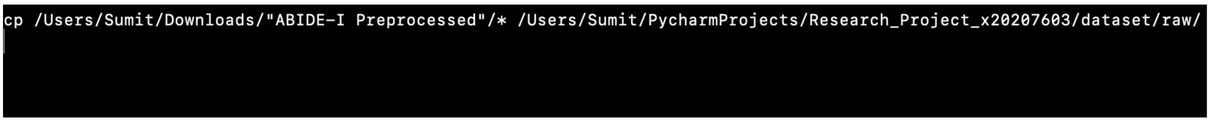
```
[redacted] unzip Research_Project_x20207603.zip
Archive:  Research_Project_x20207603.zip
   creating: Research_Project_x20207603/
  inflating: Research_Project_x20207603/gcn.py
  inflating: Research_Project_x20207603/myowndataset.py
  inflating: Research_Project_x20207603/.DS_Store
   creating: Research_Project_x20207603/dataset/
   creating: Research_Project_x20207603/models/
   creating: Research_Project_x20207603/__pycache__/
  inflating: Research_Project_x20207603/gae.py
  inflating: Research_Project_x20207603/dependencies.txt
  inflating: Research_Project_x20207603/process.py
  inflating: Research_Project_x20207603/main.py
   creating: Research_Project_x20207603/runs/
   creating: Research_Project_x20207603/.idea/
  inflating: Research_Project_x20207603/dataset/.DS_Store
   creating: Research_Project_x20207603/dataset/processed/
```

Figure 2: Unzipping the code

4.2 Copy dataset

To run the first model, copy the dataset downloaded in 3 and paste it in *dataset/raw* folder inside base code folder *Research_Project_x20207603*

> cp <Full path to downloaded files> <Full path to unzipped code folder>/dataset/raw



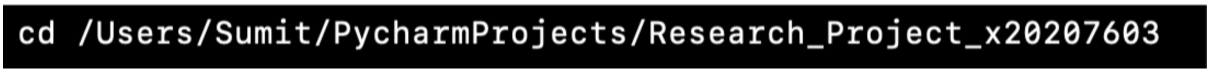
```
cp /Users/Sumit/Downloads/"ABIDE-I Preprocessed"/* /Users/Sumit/PycharmProjects/Research_Project_x20207603/dataset/raw/
```

Figure 3: Copy dataset

4.3 Installing the dependencies

The Pytorch library is used for the modelling. The package names are provided in a file dependencies.txt which is present in base directory Research_Project_x20207603. It is recommended to create a new environment by using conda or pip.

1. Open Terminal on mac os or command prompt on windows os.
2. Move to the base code directory:
   cd <Full path>/Research_Project_x20207603



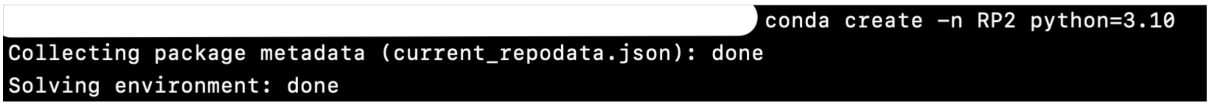
```
cd /Users/Sumit/PycharmProjects/Research_Project_x20207603
```

Figure 4: Change directory

3. To create a new environment:

   conda create -n <new_env> python=3.10

   conda activate <new_env>

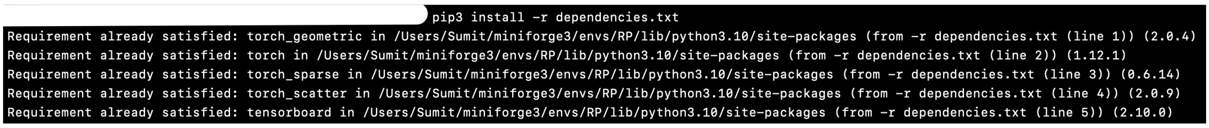


```
[redacted] conda create -n RP2 python=3.10
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Figure 5: Creation of new environment

4. To install the dependencies, follow the steps below:

   pip install -r dependencies.txt



```
[redacted] pip3 install -r dependencies.txt
Requirement already satisfied: torch_geometric in /Users/Sumit/miniforge3/envs/RP/lib/python3.10/site-packages (from -r dependencies.txt (line 1)) (2.0.4)
Requirement already satisfied: torch in /Users/Sumit/miniforge3/envs/RP/lib/python3.10/site-packages (from -r dependencies.txt (line 2)) (1.12.1)
Requirement already satisfied: torch_sparse in /Users/Sumit/miniforge3/envs/RP/lib/python3.10/site-packages (from -r dependencies.txt (line 3)) (0.6.14)
Requirement already satisfied: torch_scatter in /Users/Sumit/miniforge3/envs/RP/lib/python3.10/site-packages (from -r dependencies.txt (line 4)) (2.0.9)
Requirement already satisfied: tensorboard in /Users/Sumit/miniforge3/envs/RP/lib/python3.10/site-packages (from -r dependencies.txt (line 5)) (2.10.0)
```

Figure 6: Install dependencies

4.4 Run the code

There are a total of 5 script files (*main.py, myowndataset.py, gcn.py and process.py*). The *main.py* is main script which calls other script files and generates the results. The script *myowndataset.py* performs the additonal pre-processing, *gcn.py* defines the graph auto-enoder neural network for latent representation learning and graph neural netwrok used for classification and *process.py* performs the training and testing steps. To run the code use the following command:

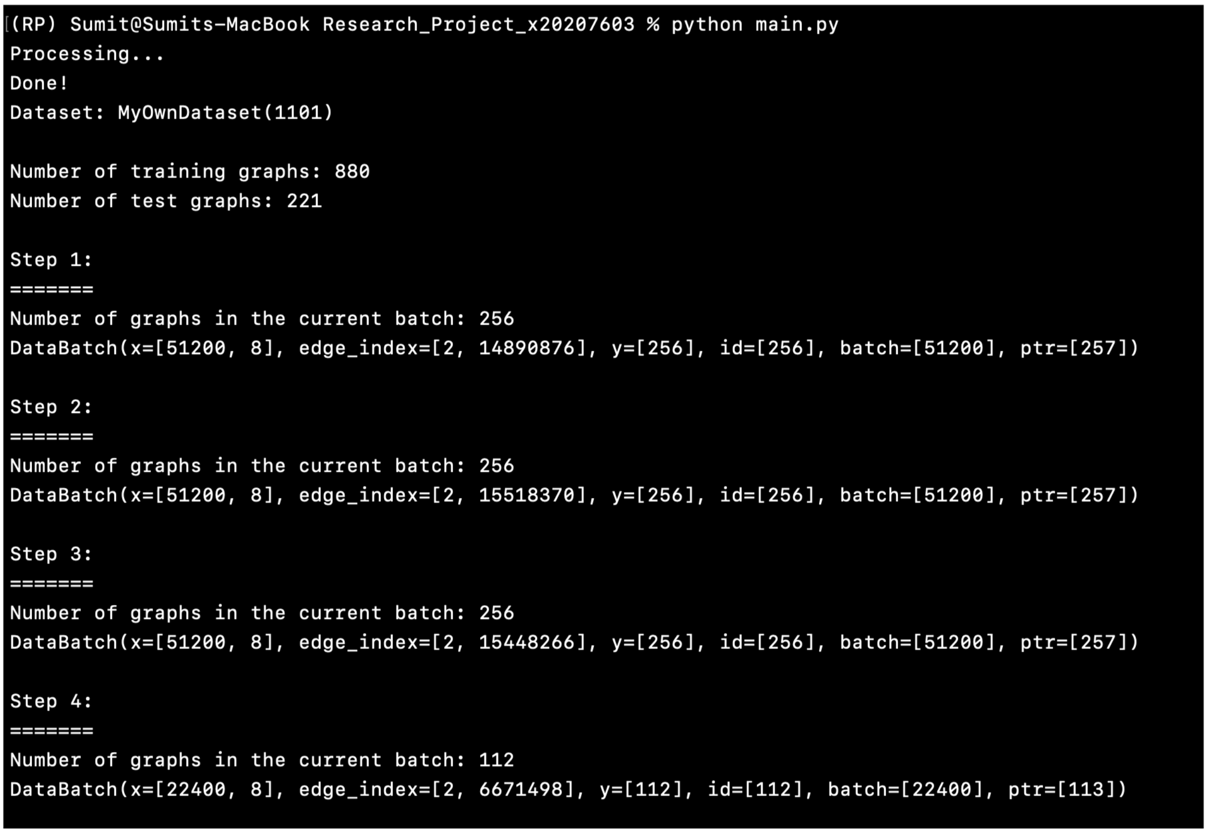
> python main.py

4.5 After completion of the run, see the graphs

The models are saved in the folder runs after the training. The graphs are generated during running of the code using *SummaryWriter* from *torch.utils.tensorboard*. After completion of run, the graphs can be visualized using the command below: tensorboard –logdir=runs

The server address started by the tensorboard is provided in the result of the above command. Open the address in a browser to analyse the graphs created after training.

Similarly, the steps from 4.2 till 4.5 is to be followed for running the second model *Research_Project_x20207603_II.zip* replacing Research_Project_x20207603 with Research_Project_x20207603_II.



```
[(RP) Sumit@Sumits-MacBook Research_Project_x20207603 % python main.py
Processing...
Done!
Dataset: MyOwnDataset(1101)

Number of training graphs: 880
Number of test graphs: 221

Step 1:
=======
Number of graphs in the current batch: 256
DataBatch(x=[51200, 8], edge_index=[2, 14890876], y=[256], id=[256], batch=[51200], ptr=[257])

Step 2:
=======
Number of graphs in the current batch: 256
DataBatch(x=[51200, 8], edge_index=[2, 15518370], y=[256], id=[256], batch=[51200], ptr=[257])

Step 3:
=======
Number of graphs in the current batch: 256
DataBatch(x=[51200, 8], edge_index=[2, 15448266], y=[256], id=[256], batch=[51200], ptr=[257])

Step 4:
=======
Number of graphs in the current batch: 112
DataBatch(x=[22400, 8], edge_index=[2, 6671498], y=[112], id=[112], batch=[22400], ptr=[113])
```

Figure 7: Running main.py script

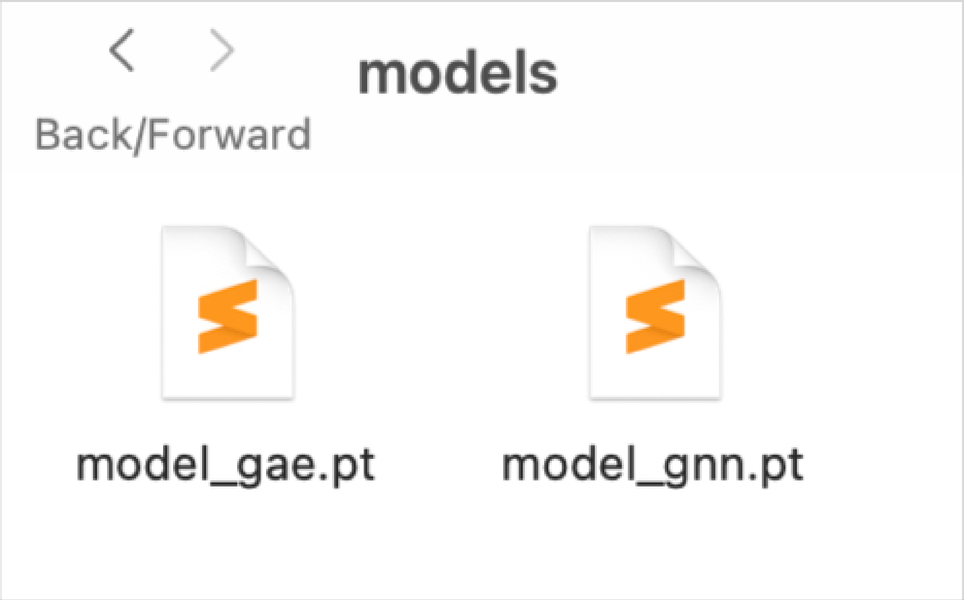
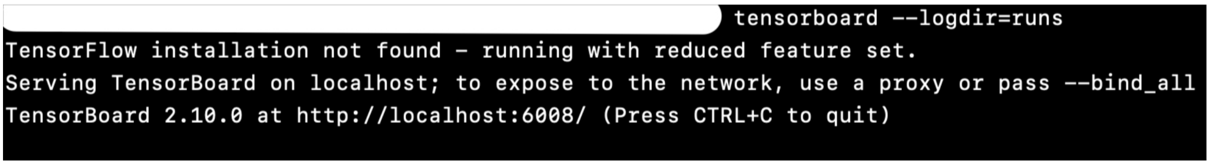


Figure 8: Model saved after training



```
[redacted] tensorboard --logdir=runs
TensorFlow installation not found - running with reduced feature set.
Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all
TensorBoard 2.10.0 at http://localhost:6008/ (Press CTRL+C to quit)
```

Figure 9: Running tensorboard to visualize graphs