ASD detection on the ABIDE dataset using a population-based GCN

Mattia D'Agostini
University of Potsdam, Germany
August 2025
mattia.dagostini@uni-potsdam.de

Abstract

For the final project of the SoSe 2025 class on Convolutional Graph Neural Networks, I have developed and trained a population-based GNN to classify preprocessed fMRI scans from the ABIDE dataset into ASD or Healthy Control group. The model was trained on a subset of the ABIDE dataset, and its performance was evaluated in terms of AUC and F1 scores. The results were compared with a similar CNN-based model. The graph-based model achieved slightly better performance, highlighting the benefits of a graph-oriented approach when it comes to model architecture. The project code can be found on Github [1].

1 Introduction

Functional magnetic resonance imaging or functional MRI (fMRI) is a medical technique that measures brain activity by detecting changes associated with blood flow. This particular technique is widely used in various medical fields, being employed in brain mapping tasks as well as in diagnostic tasks. Nowadays, data from fMRI scans is often used in academic research related to medical diagnosis. Recent developments in deep learning have led to this technology being used in combination with neural networks. Models are trained to diagnose a neural condition based on fMRI scans. The idea behind this approach is not to substitute a doctor's opinion, since a medical diagnosis is a process based on multiple tests and needs a human validation before being presented to the patient. Neural networks-based models can rather be used to estimate the likelihood of a patient condition when an ordinary procedure cannot be performed. This is often used to estimate the likelihood of various conditions, such as ASD (Autism Spectrum Disorder) by using prenatal data in a fetus. Additionally, machine learning approaches can be used to determine the biomarkers associated to a specific condition, since it's possible to trace the statistical reasons that led to a diagnosis made by a model. In this context, Graph Neural Networks [15] can be considered a useful model architecture, as the graph structure allows the data to be represented according to the brain parcellation employed during data acquisition.

[1] https://github.com/matitooo/gnn-asd

2 Related Work

As presented in Luo et al. [9], ASD diagnosis is one of the main medical tasks in which GNN models are applied. GNN can be individual based or population based [1]. In the first approach an individual graph is built for every subject, by considering brain ROIs (Regions of interest) as nodes and connections between ROIs measured during the fMRI scan as edges. When it comes to population-based approaches, a single graph is built for all the subjects, in which every subject represents one node (hence population) and edges are computed by comparing subjects similarity based on various phenotypical features. In this experiment, a population-based GCN was built, based on Parisot et al. [12].

3 Data

Training and test data was entirely extracted from the ABIDE (Autism Brain Imaging Data Exchange) dataset [5]. More specifically, data used in the experiment was taken from a preprocessed version of the ABIDE dataset [4], which allows researchers to perform experiments using the ABIDE dataset without the neuroscientific knowledge required to pre-process it. Additional pre-processing was executed on the training and test data, and it will be described in the Methodology and Experiment section.

4 Methodology and Experiment

4.1 fMRI Data Extraction and Preprocessing

Data is initially downloaded using the ABIDE preprocessed module [4]. The fMRI data has already been parcellated into 200 ROIs using the Harvard-Oxford brain atlas [6]. Alongside the fMRI data, phenotypic features for each subject are downloaded and used to construct the population-based graph. Next, for each ROI, using code from Parisot et al. [12], the mean fMRI value is computed and used to calculate the Pearson correlation between ROIs. This process is repeated for every subject, resulting in a connectivity matrix for each individual. This matrix, after being preprocessed with Fisher Z transformation and vectorized, will serve as the node feature in the final graph architecture.

4.2 Graph Creation

Following the approach used in Parisot et al. [12], a population graph is built by connecting nodes (subjects) with an arc if they share the same sex and if their fMRI scores come from the same dataset (ABIDE is a collection of multiple pre-existing datasets). The choice of treating the site of origin as a phenotypic feature is common in research where graphs are built using a dataset consisting of a sub-datasets collection. This is done to avoid differences in data pre-processing between different sub-datasets leading to biases in subject classification. Arc connections are weighted, being stronger if two subjects share both features. By using this approach, the population graph was built. Using a similar code and the Phenotypic Features CSV, it would be possible to build a more complex graph, which would require a deeper knowledge of correlation between phenotypic features and ASD diagnosis.

4.3 Graph processing

Next, the graph is processed to reduce the number of connections, in order to reduce noise in the data and computation required in the training and evaluation phase by obtaining a sparser graph. To achieve this, a correlation-based distance metric is used to compute the functional similarity between subjects' fMRI connectivity vectors, resulting in a matrix representing the subject similarities in terms of fMRI scans. The functional similarity is computed by first evaluating the pairwise correlation distance, which is next converted into a similarity using a Gaussian kernel through the following equation.

$$S_{ij} = \exp\left(-\frac{\text{dist}(X_i, X_j)^2}{2\sigma^2}\right) \quad (1)$$

Where $dist$ is the correlation spatial distance from the Scipy [14] module and σ is the distance mean. The graph matrix and the functional similarity matrix are then multiplied, resulting in a more detailed graph in terms of subject similarities. Finally, a threshold is applied to the graph, keeping connections only between nodes featuring an edge weight higher than 0.7. This process returns the final graph that will be used to train the GCN model.

4.4 GCN model architecture

The GCN [8] model architecture used in this experiment was based on the one described in Parisot et al. [12]. The model architecture consists of two GCN layers, using Chebyshev polynomials to approximate graph convolutions. This is done by expanding the graph Laplacian filter as a series of Chebyshev polynomials and allows information extracted from the fMRI to be propagated better through the model layers. The approach and its parameter tuning, the maximum polynomial degree (K), were taken from Parisot et al. [12]. Each layer is followed by a layer dropout [13], which is used to improve the model performance, avoiding overfitting. Finally, a fully connected layer is used to predict the class associated to the graph node (1 for ASD and 0 for Healty Control Group).

4.5 Baseline for comparison

To compare the graph-based model to a non-graph-based one, a two layer 1D CNN [11] model with a final fully connected layer was developed. This model only operates on the vectorized fMRI features without using any graph information. The model hyper-parameters were chosen to mimic as close as possible the GCN one in terms of dropout rate and hidden layer size.

4.6 Training

The model was trained using an Adam optimizer [7] and Cross-Entropy loss [10]. Hyper-parameters sweeping was done using grid search and maximizing the AUC score [3]. The hyper-parameters training sweeps were done using Weights & Biases Biewald [2] and returned the following parameters as the best performing.

Table 1: Best performing parameters

Parameter	Value
Dropout	0.5
Weight decay	0.0005
Hidden units	32
Learning rate	0.0001
Epochs	200
Maximum polynomial degree (K)	3
Threshold	0.7

4.7 Evaluation

Models were evaluated using two metrics: AUC score [3] and F1 score. F1 score was preferred to accuracy, which was used as the reference metric in Parisot et al. [12], as it is not conditioned by class imbalances in datasets and takes into account both precision and recall. For each model, multiple F1 scores were computed by varying the classification threshold. The best score was kept as the final model score.

5 Results

The two evaluated models obtained the following scores In Parisot et al. [12], the equivalent GCN model achieved an AUC of approximately 0.75, which is similar to the performance obtained by the model proposed in this experiment. On the other hand, it is important to note that the CNN-based model returned comparable scores, suggesting that the additional pre-processing applied to the fMRI scans played an important role. Given the similar results, the GCN model is to be preferred as it requires less computational power. These results could potentially be improved by constructing the graph using additional phenotypical features. However, for most phenotypical features available in the ABIDE dataset, it is not straightforward to define a similarity measure between two entries of the same feature.

Table 2: Models Test Performance

Model	Test AUC	Test F1	Threshold
GCN	0.74	0.75	0.35
CNN	0.72	0.74	0.28

References

[1] David Ahmedt-Aristizabal, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. Graph-based deep learning for medical diagnosis and analysis: Past, present and future. 21(14):4758.

[2] Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

[3] Andrew P. Bradley. 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159.

[4] Cameron Craddock, Yassine Benhajali, Carlton Chu, François Chouinard, Alan Evans, András Jakab, Budhachandra S. Khundrakpam, John D. Lewis, Qingyang Li, Michael Milham, Chaogan Yan, and Pierre Bellec. 2013. The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives. *Frontiers in Neuroinformatics*, 7:41. Conference Abstract: Neuroinformatics 2013, Stockholm, Sweden, 27 Aug - 29 Aug, 2013.

[5] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto, B. Deen, S. Delmonte, I. Dinstein, B. Ertl-Wagner, D. A. Fair, L. Gallagher, D. P. Kennedy, C. L. Keown, C. Keysers, and 23 others. 2014. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. 19(6):659–667.

[6] D. N. Kennedy, C. Haselgrove, B. Fischl, J. Breeze, J. Frazier, L. J. Seidman, and J. M. Goldstein. 2025. Harvard - oxford cortical structural atlas. `http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases`. RRID: $SCR_0$01476.

[7] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

[8] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *Preprint*, arXiv:1609.02907.

[9] Xuexiong Luo, Jia Wu, Jian Yang, Shan Xue, Amin Beheshti, Quan Z. Sheng, David McAlpine, Paul Sowman, Alexis Giral, and Philip S. Yu. 2024. Graph neural networks for brain graph learning: A survey. *Preprint*, arxiv:2406.02594 [cs].

[10] Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2023. Cross-entropy loss functions: Theoretical analysis and applications. *Preprint*, arXiv:2304.07288.

[11] Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *Preprint*, arxiv:1511.08458 [cs].

[12] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrerro Moreno, Ben Glocker, and Daniel Rueckert. 2017. Spectral graph convolutions for population-based disease prediction. *Preprint*, arxiv:1703.03020 [stat].

[13] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting.

[14] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 16 others. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

[15] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *CoRR*, abs/1812.08434.