MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

Medical data processing with graph neural networks

Master's Thesis

Author
Anna Gergály

Advisor
dr. Luca Szegletes

March 31, 2025

Contents

Kivonat i

Abstract ii

1 Introduction 1

1.1 Motivation 1

1.2 My approach 1

1.3 Structure 1

2 Background 2

2.1 Graph Neural Networks 2

2.1.1 Neural Networks 2

2.1.2 Working on Graph Data 3

2.1.2.1 Graph convolutions 4

2.1.2.2 Common tasks 4

2.1.3 Graph Neural Network Types 5

2.1.3.1 Graph Convolutional Networks 5

2.1.3.2 Graph Attention Networks 5

2.2 Medical data 5

2.2.1 Medical imaging 5

2.2.2 MRI and fMRI imaging 5

2.2.2.1 Technical background 6

2.2.2.2 Processing 6

2.2.2.3 Clinical significance 6

3 Technical background 7

3.1 Deep Learning Frameworks 7

3.1.1 Pytorch 7

3.1.2 DGL 7

3.1.3 Sklearn 7

3.1.4 Tsfresh . . . 7
3.1.5 Xgboost . . . 7
3.1.6 Supporting packages . . . 7
3.2 Using fMRI data . . . 7
3.2.1 Data formats . . . 7
3.2.2 Pre-processing . . . 7
3.2.3 Extracting ROI Values and Connectomes . . . 7

4 Methods and Implementation 8
4.1 Graph Neural Networks on Connectomes . . . 8
4.1.1 Dataset . . . 8
4.1.2 Methodology . . . 8
4.1.3 Results . . . 9
4.1.4 Conclusions . . . 9
4.2 Connectome-based diffusion . . . 9
4.2.1 Dataset . . . 9
4.2.2 Methodology . . . 9
4.2.3 Results . . . 9
4.2.4 Conclusions . . . 9
4.3 Minimising device-to-device differences . . . 9
4.3.1 Dataset . . . 9
4.3.2 Methodology . . . 9
4.3.3 Results . . . 9
4.3.4 Conclusions . . . 9

5 Future work 10
5.1 Future work . . . 10
5.2 Conclusions . . . 10

Acknowledgements 11

Bibliography 12

HALLGATÓI NYILATKOZAT

Alulírott *Gergály Anna*, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2025. március 31.

Gergály Anna
hallgató

Kivonat

Jelen dokumentum egy diplomaterv sablon, amely formai keretet ad a BME Villamosmérnöki és Informatikai Karán végző hallgatók által elkészítendő szakdolgozatnak és diplomatervnek. A sablon használata opcionális. Ez a sablon LaTeX alapú, a *TeXLive* TeX-implementációval és a PDF-LaTeX fordítóval működőképes.

Abstract

This document is a LaTeX-based skeleton for BSc/MSc theses of students at the Electrical Engineering and Informatics Faculty, Budapest University of Technology and Economics. The usage of this skeleton is optional. It has been tested with the *TeXLive* TeX implementation, and it requires the PDF-LaTeX compiler.

Chapter 1

Introduction

1.1 Motivation

1.2 My approach

1.3 Structure

Chapter 2

Background

2.1 Graph Neural Networks

Many real life problems can be best modelled with graphs and these representations can code a lot of information if assessed correctly. By developing techniques that work to extract this information and extrapolate from it effectively, we can build powerful systems that can predict, classify and advise based on graph data.

Graph algorithms have a long history in mathematics and there is a rich variety of graph related problems that are best solved using these classical discrete mathematical methods: pathfinding algorithms, breadth or depth-first search algorithms are used every day both in everyday IT applications and infrastructure, and also in research. But in certain cases these methods may not be well-equipped to handle the task at hand. With the large amount of data collected every day and the specialized tasks that need to be fulfilled there is reason to bring machine learning into graphs and the discipline has exploded in popularity in the last five years or so.

2.1.1 Neural Networks

Neural networks are an especially interesting and useful area of machine learning and data analysis: as referenced in their name, they were created in resemblance of the human brain's structure, mimicking the interconnected neurons. In recent years they became the flagship of AI research because of their ability to work on incredibly large datasets effectively and produce never seen before results.

The basis of a neural network are neurons which sum up incoming ”signals” multiplied by weights and apply an activation function to their output. The network learns by updating these weights and biases based on the training data, to match the desired output to each piece of input. The activation function adds non-linearity to the model, making it capable of learning more complicated relationships in the data.

Neurons are typically organized into layers in a neural network and the models usually benefit from having a large number of layers: modern models for more complex tasks can get very 'deep' which lead to the popularization of the term deep learning when talking about such models. But since having more layers makes a model more computationally intensive, leading to higher costs and slower inference, researchers are often looking to prune models or find architectures that deliver similar results while having a lower parameter count.

Parameters in a model are updated through a process called backpropagation. Here, for labeled training data, the error of the prediction of the model is calculated through the use of a loss function. This should be representative of how far off the model was from the ground truth and also easy to work with derivation-wise and numerically. This is then ”propagated back” through the model by deriving what each of the model’s parameters contributed to this loss. This is what’s known as the gradient and it is used to perform gradient descent: based on the resulting derivative we have a direction to move in to lower the loss.

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{i,j}}, \tag{2.1}$$

In case of larger datasets it is impossible to calculate the gradient for the entire dataset in one go, so a process called stochastic gradient descent is used, where randomly selected subsets (batches) of the dataset are used to calculate it during training. Having a gradient, the other part needed is the learning rate: how much to change weight in the given direction. This can be considered hyperparameter of the model, but modern optimizers (such as ADAM) use adaptive learning rate optimization techniques (such as momentum and RMSProp) to adjust learning rate during training to ensure a smooth and fast convergence.

From the most fundamental concepts of the original perceptron and the breakthrough idea of using backpropagation, neural networks became a widespread and varied phenomenon: there are architectures for different types of data optimized for different kinds of tasks; these models have proved to be applicable in almost any area.

A very active and widely used branch of neural networks are convolutional neural networks (CNNs). These are most commonly used on image data and work by convolving learnable kernels with the input to extract features. This is useful for capturing spacial relationships allowing for better pattern detection while also greatly decreasing the number of trainable parameters compared to a fully connected feed-forward network. Attention-based networks are another important model class that became very successful in recent times. Transformer architectures are used in many disciplines and are capable of state-of-the-art results. Both of these concepts turn up in graph neural networks as well.

2.1.2 Working on Graph Data

Graphs are a very versatile mathematical concept because of the way they lend themselves very neatly as a generalization. They have great expressive power when it comes to describing relations, groups and things with a rich inner structure. We can find graph-based datasets in a variety of domains: molecules in chemistry, interaction graphs in social sciences, knowledge graphs and computer networks. The level of abstraction they provide makes them suitable for use in many fields, as connections between items or molecules or people are vital in understanding complex natural systems.

One area that is particularly rich with such examples is biology and medicine. Interactions between drugs, relationships between species and contact tracing in epidemiology are all important facets that can benefit from the ability to analyse graph datasets. We can also find interesting networks to analyse inside of organisms: the focus of this thesis is the analysis of network formed by brain regions and the interactions that happen in the human brain.

Graphs are mathematically defined as a set of nodes (or vertices) and edges (or links) which run between two nodes. They can be categorized by their edge structure: directed graphs specifically code the information of a starting and end node, while undirected graphs do not. In terms of graph neural networks this is a very important distinction as it fundamentally alters how the network's meaning. Certain types of networks are designed with a specific type of graph in mind and might need special normalisation.

Certain types of graphs with special properties can be especially interesting in certain areas. Trees (which contain no cycles) can be useful for describing containment hierarchies or sentence structures, for example. Based on semantic meaning we can also talk about heterogeneous graphs were nodes may represent entirely different things, purchase or interaction graphs in a recommender system where a vertex could be a user or an item to be purchased. In this case it is important to make this information available to the network.

A graph dataset often codes much more information then just the mathematical structure itself. Ideally some information is available of each node; a feature descriptor vector for each node supplies more information for the network and also helps identify the node it is attached to: since graphs are order agnostic the model needs another way to identify which node is which.

The contents of such a descriptor vector must be domain specific. For example in case of a social network it would code information about a given person: age, gender, height. In case of a molecule prediction scenario it could be atomic weight or covalent radius. For the brain fMRI analysis case most relevant to this thesis, it could be information about a given brain region, average activation or even an activation timeseries.

In case there is no suitable information available for the node feature vectors it is common practice to use rows of the identity matrix: in absence of additional data this can be useful to serve as the identification tool the network needs. In some cases information might be available in other ways, for example as edge weights or edge feature vectors. These could be descriptive of the users' interactions in a social network example.

Node feature vectors can be concatenated together to form a matrix and together with the adjacency matrix (or edge weight matrix) of the graph they form a very neat representation of the graph. This makes the highly variable structure of the graph containable within a constant, well-structured manner that is crucial for machine learning applications.

But since graphs can be used to represent complex structures and hierarchies, where this inner construction is more pronounced and relevant then in other cases, they require specialized methods when it comes to machine learning and neural networks. A lot of these methods take after image recognition or object detection solutions.

The reason for this is that while images do not have explicitly structured data in the same way as graphs do, their contents can be in complex spatial relations with each other. A computer vision model needs to take these into account when performing complex tasks such as segmentation or creating a full description of what is happening in a photo. Even for simple object detection, more complex object have hierarchical structures that the model needs to 'understand' in some way.

2.1.2.1 Common tasks

The motivation behind working with graph datasets is of course to perform some sort of inference using the patterns extracted from the training data. Since these datasets are diverse both in semantics and in structure this could mean a lot of different types of tasks. In this section the most common of these will be described with examples and commonly used methods for solving them.

Most graph architectures work by assigning a vector to each node on their output. This is often called a node embedding: an n-dimensional vector is attached to the node that attempts to convey all available information about the node and its role in the graph structure, effectively embedding it in an n-dimensional space. Node embeddings are not strictly only generated in deep learning models, there are for example random-walk based methods for this purpose, such as node2vec.

In case of deep learning models the creation of node embedding is an iterative process as there is a new vector attached to the node after every layer. In this process the original feature vectors can be thought of as a sort of 0th iteration of the embedding, purely representing information about the node itself. With more and more layers aggregating information on a larger and larger neighbourhood of a node this slowly transforms into containing information of the structure of the network as well.

Node prediction. In this

Since the network output is representative of a nodes in the first place, this is the most straightforward type of task to solve using a deep learning model.

Graph prediction.

Edge prediction.

Clustering.

2.1.2.2 Graph convolutions

The Laplacian of a graph is a square matrix $n \times n$ in size (where n is the number of nodes in the graph). It can be calculated from the adjacency matrix and diagonal node degree matrix:

$$L = D - A,$$

where L is the Laplacian, D is the diagonal node degree matrix and A is the adjacency matrix. In the diagonal node degree matrix there is similar to an identity matrix but in each row the instead of one we have the degree of the node corresponding to said row. The Laplacian can be used to build polynomials that can be used on node features.

$$p_w = w_0 I_n + w_1 L + w_2 L^2 + \cdots + w_d L^d = \sum_{i=0}^{d} w_i L$$

These polynomials can be thought of as analogues of 'filters' in CNNs, and the coefficients w_i as the weights of the 'filters'. There exists a type of network that built directly on the concept of Laplacian polynomials: ChebNet used Chebyshev polynomials and normalized Laplacians to create a more numerically stable and 'stackable'.

ChebNet was a big step in ushering research into graph networks, as it motivated many to think about graph convolutions from a different perspective. Current graph neural networks utilize graph convolutions in ways that make use of 'computational shortcuts':

bypassing costly operations such as eigenvalue calculations can make a model much more scalable, both in terms of data throughput and model size.

These shortcuts often mean calculating local approximations of the graph convolutions. Local graph operations are often so called 'message passing' operations: every node in the graph sends a 'message' to its neighbours based on its own data and then each node aggregates the received messages. In practice this means each node vector is updated in a layer based on its neighbours.

2.1.3 Graph Neural Network Types

2.1.3.1 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) expand on the idea of CNNs, which are commonly used in image processing. Observing from a graph perspective, an image can be considered a very special case of a graph, where pixels are nodes and their neighbours can be determined from the grid. This type of network generalizes the concept of local convolutions from the image domain to general graphs.

2.1.3.2 Graph Attention Networks

2.2 Medical data

We can call anything medical data that relates to human healthcare in one way or another. This most commonly means data collected by healthcare institutions about patients using sensor equipment, but it could also be information about a patient's habits or general environment, data from drug test trials or location data in case of epidemiological contact tracing.

This kind of data is often very personal and regarded as sensitive data: people are entitled to equal treatment regardless of medical status and as such information about the health status of an individual is protected. Many patients do not want employers or other third parties who could use such information against them to know details about their conditions and their treatment.

For this reason doctors are required to not give out patient information unless it is strictly necessary and/or they have the informed consent of the person. This puts medical researchers in an interesting position; medical researchers using machine learning even more so. Data collected specifically for experiments where subjects can consent to their data being used is a straightforward situation, but there is data collected every day, worldwide in hospitals that could be very useful for furthering medicine, but doing so poses data privacy concerns. This is especially important in case of machine learning and neural network research as these disciplines require a very large amount of data that is very hard to collect using only organized experiments.

2.2.1 Medical imaging

Non-invasive medical imaging techniques, such as MRI, X-ray, and CT scans, are essential tools in medicine, allowing physicians to understand the internal structures and functions of the body without the need for surgery. These techniques are invaluable for their ability to produce high-resolution images of different types of tissues, making it ideal for assessing the state and function of internal organs.

2.2.2 MRI and fMRI imaging

Magnetic Resonance Imaging (MRI) produces its high-fidelity images, while using no ionizing radiation, making possible its repeated use in patients.

2.2.2.1 Technical background

2.2.2.2 Processing

2.2.2.3 Clinical significance

Chapter 3

Technical background

3.1 Deep Learning Frameworks

In my work I used Python both for implementing machine learning solutions and for creating pipelines; downloading, sorting and preparing data, as well as various scripting needs. For the purpose of organizing the used Python packages and ensuring a consistent environment I have used a virtual conda environment created with miniconda.

For machine learning tasks I have employed various libraries commonly used for such tasks to avoid reimplementing standard solutions, both in support of my own final solutions and for creating baseline solutions for comparison and for data exploration and visualization purposes.

3.1.1 Pytorch

3.1.2 DGL

3.1.3 Sklearn

3.1.4 Tsfresh

3.1.5 Xgboost

3.1.6 Supporting packages

3.2 Using fMRI data

3.2.1 Data formats

3.2.2 Pre-processing

3.2.3 Extracting ROI Values and Connectomes

Chapter 4

Methods and Implementation

4.1 Graph Neural Networks on Connectomes

4.1.1 Dataset

The ABIDE (Autism Brain Imaging Data Exchange) dataset focuses on the furthering research into ASD (autism spectrum disorder) through neuro-imaging and neuroscience. The dataset contains in total 1112 resting state fMRIs of subjects both with ASD (539 individuals) and typically developing controls (573 individuals).

The imaging data has been collected by 16 institutions, who collaborated to create a publicly accessible anonymised dataset to allow the broader scientific community to take part in ASD related research while preserving the privacy of the participants. As such, datasets do not contain any protected health information.

Since fMRI preprocessing techniques are very complex image processing operations and heavily resource intensive, an initiative also formed to provide a preprocessed version of the dataset.

Since there is not a widely accepted best practice method for preprocessing fMRI images, the participating five teams have all used their preferred methods to clean the data. The participants have also provided a ROI activation timeseries calculated using 7 different brain atlases from each type of the preprocess method.

The very close ratio of affected vs control subjects (539 to 573) means the dataset is well-balanced in terms of a classification task on the diagnosis of subjects.

4.1.2 Methodology

In the paper [**?**] by Wang et al. researchers worked with the ABIDE dataset to create a graph convolutional network based architecture for predicting the diagnosis of a patient based on the resting state fMRI.

4.1.3 Results

4.1.4 Conclusions

4.2 Connectome-based diffusion

4.2.1 Dataset

4.2.2 Methodology

4.2.3 Results

4.2.4 Conclusions

4.3 Minimising device-to-device differences

4.3.1 Dataset

4.3.2 Methodology

4.3.3 Results

4.3.4 Conclusions

Chapter 5

Future work

5.1 Future work

5.2 Conclusions

Acknowledgements

Ez nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

Bibliography