Skye Purchase

Extracting Concepts from Simplified Graph Neural Networks

Computer Science Tripos – Part II

Peterhouse

February 19, 2023

Declaration of originality

I, Skye Purchase of Peterhouse, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose. I am content for my dissertation to be made available to the students and staff of the University.

Signed Skye Purchase

Date February 19, 2023

Proforma

Candidate Number:	**123456**
Project Title:	**Extracting Concepts from Simplified Graph Neural Netw**
Examination:	**Computer Science Tripos – Part II, 2023**
Word Count:	**...**[1]
Code Line Count:	**...**
Project Originator:	123456
Supervisor:	Charlotte Magister and Pietro Babiero

[NB: Should I include Pietro Lio for the idea of concepts and Charlotte Magister for the paper?**]**

Original Aims of the Project

[NB: This is a first draft I am not sure what specifically should go here**]**

This project analyses the effect that simplifying graph neural networks has on the graph concepts that are extracted from a trained model. It focuses specifically on the widely aclaimed Simplified Graph Convolution *Wu et al.* [1][SGC] which removes the non-linearity between layers from previous graph neural network approaches reducing the problem of graph representational learning to a precomputation on the graph structure and a single linear regressor. Using the graph concept extraction method and metrics proposed in *Magister et. al* [3] the project aims to compare the concepts from trained SGC models to a baseline of Graph Convolution Network models. Further extensions then build on SGC to see which techniques can improve accuracy and concept metrics.

Work Completed

The project was a success. It fulfilled all the success criteria and further demonstrated that the Simplified Graph Convolution (SGC) does not produce as rich concepts as the Graph Convolution Network under the metrics of concept purity and completeness. Furthermore, it demonstrated that in the case of highly synthetic data SGC is unable to match the performance of GCN by a significant margin. My extensions demonstrate that in the case of real-world graph datasets, where graph structure is important, SGC continues to underperform compared to GCN and does not produce comparable concepts. **[NB:** I have not fully completed the results gathering from all of my extensions. Specifically the SGC+GCN (SGCN) and JumpNet SGC (JSGC).**]**

[1]This word count was computed by `texcount`

I additionally set out to create a new parameterised dataset to further analyse the shortcomings of SGC when dealing with graph structure. **[Error: This is still not complete but I am undertaking this task.]**

Special Difficulties

None

Contents

Chapter 1

Introduction

This dissertation explores the effect that simplifying graph neural network architecture has on the explainability of the trained model. Specifically it focuses on the ideas of graph concepts which are extracted from the trained model to provide a visual demonstration of which subspaces of the input influence the which label is chosen. These concepts are compared to the original Graph Convolution Network (GCN) using the metrics of concept purity and completeness proposed in [3]. The specific simplified model chosen is the Simplified Graph Convolution (SGC) proposed in [1] due to claims that it matches the performance of GCN.

1.1 Motivation

1.2 Related Work

1.3 Contributions

Chapter 2

Preparation

2.1 Background

2.2 Graph Representational Learning

2.2.1 Graph Convolutional Network

2.2.2 Simplified Graph Convolution

2.3 Explainability

2.3.1 Concepts

2.3.2 Graph Concept Explainer

2.4 Datasets

2.4.1 Synthetic

2.4.2 Real-World

2.5 Tools Used

2.6 Software Methodology

2.7 Testing

2.8 Licensing

2.9 Starting Point

Chapter 3

Implementation

3.1 Machine Learning Pipeline

3.1.1 Reproducibility

3.1.2 Experimentation

3.2 Models & Datasets

3.3 Concept Extraction and Evaluation

3.3.1 Extraction

3.3.2 Evaluation

3.3.3 Visualisation

3.4 Pytorch Geometric

3.5 Repository

Chapter 4

Evaluation

4.1 Success Criteria

4.2 Methodology

4.3 Results Reproduction

4.4 Comparison of Accuracy

4.5 Comparison of Concept Scores

4.6 Extensions

4.6.1 SGC Graph Classification

4.6.2 SGC and GCN Mixed Model

4.6.3 JumpNet style SGC

4.6.4 Tunable Dataset

Chapter 5

Conclusion

Appendix A

Proposal

A.1 Description

A.1.1 Introduction

Within the area of geometric deep learning there have been recent ablation studies looking into the effectiveness of Graph Neural Networks (GNNs). The majority of these studies question the effectiveness of the deep neural network approach of multiple layers separated by non-linear function passes when working with geometric datasets (graphs). [1] introduce a new approach, Simplified Graph Convolution (SGC), which remove these non-linear functions from the network. This reduces the problem to a pre-computation on the graph adjacency matrix and a simple linear regression using a single weight matrix. The pre-computation on the graph adjacency matrix encodes information about message passing between nodes in the graph. [2] introduce further variations on SGC that use the same underlying concept of a pre-computation but deal with the parameters differently allowing for more complex associations. In both cases the results show that removing the non-linearity does not hinder the performance of the network and can in fact improve performance.

Similarly, the has been a lot of interest into explainable artificial intelligence (XAI) to move away from the black box nature of AI models. There exists multiple methods within this field of machine learning and I will specifically focus on the idea of Concepts. Concepts focuses on relating specific outputs of a model to subspaces within its input space, this gives an indication of what the model is using within the input space to infer the given output. The collections of these subspaces are what are known as concepts. This approaches allows a human actor to get a better understanding of the model's inference as they can compare their own intuition of the input to the concepts the model uses to produce the given result. [3] introduce GCExplainer which adapts prior techniques to extract high-level concepts from GNNs. The paper focuses on extracting concepts from a Graph Convolutional Network (GCN, [4]) model.

Both SGC and GCExplainer are research papers but both including detailed sections on experiment setup and available github repositories. In the case of GCExplainer the author of the paper is one of my supervisors and can therefore clear up any setup details and help with common pitfalls in the training process.

A.1.2 Extensions

Further extensions on the core goal will look at the concepts extracted from variations on SGC with potential new approaches outside of existing literature being implemented. Approaches outside of literature include mixing SGC "layers" within GCN layers, allowing for non-linearity to be added later in the forward pass. Equally using multiple SGC steps and using Jump Knowledge [5] to help combat the common problem of over-smoothing in GNNs. Further extensions may be found based on results during the project.

These extensions can also look into how accurate the new models are outside of the concept metrics and provide further comparison between accuracy and concept purity & completeness.

A.1.3 Substance and Structure

With the rise in popularity of simplified GNNs due to their simple nature and effective representational learning it is important to evaluate how they fit into explainability frameworks. This project therefore looks to extract concepts from SGC trained on the same datasets using in [3] to compare against GCN. This work will provide useful insight into how these simplified networks operate as well as providing further comparison between previous GNN techniques (GCN) and new simplified GNN techniques (SGC). Comparison between the two GNN techniques will focus on concept purity and completeness as described in [3].

This is broken down into reproducing both papers using the setup and structure described in each to compare the results that I achieve with those published in each paper. The code used for both of these can then be combined to extract concepts from the SGC model. This will allow comparison between a simplified GNN (SGC) and a standard GNN (GCN). Further extensions can then be carried out based on these results to further investigate different GNN approaches in regard to concept extraction.

A.2 Starting Point

I have worked with Pytorch and the extension for geometric deep learning, Pytorch Geometric, over the summer culminating in a research paper. During this summer project I worked with building and testing graph datasets, building new GNN models, and testing/comparing GNN models from previous models to the new models. During the project I also briefly learnt about and used convolutional neural networks and transformers. I do not have experience with XAI or the area of Concepts.

No code has been written for this project beyond the code that will be used within the Pytorch and Pytorch Geometric libraries.

A.3 Success Criteria

The project will be deemed a success if I

- Implement SGC and extract the concepts used for each of the datasets
- Implement GCN and extract the concepts to use as a baseline
- Compare the concepts between SGC and GCN using the metrics of concept completeness and concept purity

A.4 Work Plan

A.4.1 Interval 1: 13/10 - 26/10

Preparatory Work: Download datasets (both SGC and GCExplainer), preliminary tests on datasets to check all correct, setup up environment (required libraries are up-to-date etc.), test training on local machine and servers to determine extra resources. Read and annotate the three papers being implemented.

Start work on implementing Simplified Graph Convolution (SGC).

A.4.2 Interval 2: 27/10 - 9/11

Complete implementation of SGC and train on SGC paper datasets, compare against papers results to confirm implementation is correct. Train SGC on GCExplainer datasets.

Milestone: Trained SGC model.

Category Theory 1 Deadline: 4/11 (25%)

A.4.3 Interval 3: 10/11 - 23/11

Implement Graph Convolution Network (GCN, from GCExplainer) and train on GCExplainer datasets.

A.4.4 Interval 4: 24/11 - 7/12

Implement GCExplainer concept extraction on GCN and compare concept metrics to confirm implementation is correct.

Milestone: Working GCExplainer.

Category Theory 2 Deadline: 2/12 (75%)

A.4.5 Interval 5: 8/12 - 21/12

Carry out concept extraction on the trained SGC model(s) and produce concept metrics for SGC model(s).

Milestone: SGC concept metrics (**Core Project Complete**)

A.4.6 Interval 6: 22/12 - 4/1

Buffer. Christmas and New Year

A.4.7 Interval 7: 5/1 - 18/1

Extension: Research (read papers and annotate) one of the following techniques

- Mixing SGC and GCN layers
- Adding knowledge jumps
- Other simplified graph convolutions

A.4.8 Interval 8: 19/1 - 1/2

Extension: Implement and train the chosen technique from Appendix A.4.7 on the GCExplainer Datasets.

A.4.9 Interval 9: 2/2 - 15/2

Extension: Carry out GCExplinaer concept extraction on the trained model from Appendix A.4.7.

Deadline: Progress Report 3/2

A.4.10 Interval 10: 16/2 - 1/3

Write Dissertation: Introduction and Implementation chapters, and share with supervisors.

Deep Neural Networks 1 Deadline: 17/02 (30%)

A.4.11 Interval 11: 2/3 - 15/3

Buffer. If not required and extension runs smoothly carry out a second technique from Appendix A.4.7.

A.4.12 Interval 12: 16/3 - 29/3

Write Dissertation: Evaluation of different approaches and their concept metrics, and share with supervisors.

Milestone: First draft

A.4.13 Interval 13: 30/3 - 12/4

Write Dissertation: Respond to feedback on first draft and share with supervisors.

Milestone: Second draft

Deep Neural Networks 2 Deadline: 17/03 (70%)

A.4.14 Interval 14: 13/4 - 27/4

Write Dissertation: Respond to feedback on second draft and share with supervisors.

Milestone: Final draft

Deadline: Dissertation submission 12/5

A.5 Supervisors

- Lucie Charlotte Magister
- Pietro Babiero
- Pietro Lio

I will have joint weekly meetings at 5pm on Wednesday will all supervisors (barring temporary unavailability) to report on the progress of the project and help with any problems that have occurred in the current interval.

A.6 Resource Declaration

My own machine (AMD Ryzen 7 5700U with Radeon Graphics (16) @ 1.8GHz, 16GB RAM, and AMD ATI Lucienne) for the primary implementation of the project and remote work on any provided servers used. The majority of the project should be completed on this machine. I will also use git version control storing a remote repository on a private github repository in the eventuality of my own machine malfunctioning. Regular remote pushes at the end of the day or major project milestones will be carried out. I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure.

High performance cluster (HPC) as intensive training may be required. The required hours will remain within the SL3 bracket so no funding for SL2 will be required.

Local CL server access (idun, heimdall, etc) for less costly iterations and testing of GPU models before utilising HPC if my own machine is not sufficient. I am able to acquire kerberos tickets to use these services.

Storage space available on my own machine will be sufficienet for the datasets used. Though space on local CL and HPC servers would be required if I am using these for model training.

I will require other software packages, PyTorch and PyTorch Geometric, to help with the framework of building, testing and training my models.

Bibliography

[1] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019. 3, 7, 17

[2] Sudhanshu Chanpuriya and Cameron Musco. Simplified graph convolution with heterophily. *arXiv preprint arXiv:2202.04139*, 2022. 17

[3] Lucie Charlotte Magister, Dmitry Kazhdan, Vikash Singh, and Pietro Liò. Gcexplainer: Human-in-the-loop concept-based explanations for graph neural networks. *arXiv preprint arXiv:2107.11889*, 2021. 3, 7, 17, 18

[4] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 17

[5] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018. 18