Complex Networks [CS60078] 2023-24

Assignment 2: Node classification

Deadline for submission: 22 March 2024, 23:59 IST

General Instructions

1. In the assignment we will use the CORA network (description provided in the next page).

2. Your codes should print out exactly what is asked, and in the specified format.

3. How and what to submit: Submit one .zip or .tar.gz file containing a compressed folder that should contain all source codes, all files to be submitted (as per the task descriptions given below) and an instructions file (see next point). Name the compressed file the same as your roll number. Example: name the compressed file “19CS60R00.zip” or “19CS60R00.tar.gz” if your roll number is 19CS60R00.

4. Along with the source codes and files asked in the tasks, also submit an additional text file called “instructions.txt” where you should state how to run your codes as well as any additional information you want to convey, such as the version of Python. The instructions.txt file should also contain your name and roll number.

5. We should be able to run your submitted code in a computer with a reasonable configuration (for instance 2GB or more RAM) by following your submitted instructions. If any part of your code takes a long time to run (e.g., more than 10 minutes) report that in the instruction file with an estimate of time required.

6. The assignment should be done individually by each student. You should not copy any code from one another, or from any web source. Plagiarized codes will be awarded zero for the whole assignment. **You should not share your codes with anyone even after the submission deadline has passed.**

7. **Submit your response here: https://forms.gle/QzvtSNxUc6QyhReh6**

Dataset Description

This directory contains a selection of the Cora dataset. Unlike the previous assignment, the “**.cites**” file is split into **Train** and **Test.**

The Cora dataset consists of Machine Learning papers. These papers are classified into one of the following seven classes:

Case_Based
Genetic_Algorithms
Neural_Networks
Probabilistic_Methods
Reinforcement_Learning
Rule_Learning
Theory

The papers were selected in a way such that in the final corpus every paper cites or is cited by at least one other paper. There are 2708 papers in the whole corpus.
After stemming and removing stopwords we are left with a vocabulary of size 1433 unique words. All words with document frequency less than 10 were removed.

THE DIRECTORY CONTAINS TWO FILES:

The .content file contains descriptions of the papers in the following format:

<paper_id> <word_attributes>+ <class_label>

The first entry in each line contains the unique string ID of the paper followed by binary values indicating whether each word in the vocabulary is present (indicated by 1) or absent (indicated by 0) in the paper. Finally, the last entry in the line contains the class label of the paper.

The .cites file contains the citation graph of the corpus. Each line describes a link in the following format:

<ID of cited paper> <ID of citing paper>

Each line contains two paper IDs. The first entry is the ID of the paper being cited and the second ID stands for the paper which contains the citation. **The direction of the link is from right to left. If a line is represented by "paper1 paper2" then the link is**

"paper2->paper1". Note that your results will be wrong if you do not assume the correct convention.

Task Description:

Part 1: Node2Vec and Logistic Regression

Objective:

Implement the Node2Vec algorithm for the CORA graph and perform 7-class classification of the nodes using the generated node embeddings as features. Utilize logistic regression (LR) for the classification task.

Tasks:

1. Implement the Node2Vec algorithm for generating node embeddings. Use the provided CORA graph dataset for this purpose. Code Node2Vec from scratch.
2. Utilize the generated node embeddings as features for 7-class classification of the nodes.
3. Train a logistic regression model on the node embeddings on the **train data** to perform the classification on the **test data**. Can use built in library functions for LR.
4. Evaluate the performance of the LR model using appropriate evaluation metrics such as accuracy, precision, recall, and macro-F1-score.

Deliverables:

1. Python code implementing the Node2Vec algorithm for the CORA graph.
2. Python code for performing 7-class classification using logistic regression on the generated node embeddings.
3. A pdf report containing the evaluation results of the LR model and any observations or insights gained during the process.

Part 2: Graph Convolutional Networks (GCN)

Objective:

Implement the Graph Convolutional Network (GCN) based on Kipf's original ICLR 2017 paper and perform the same 7-class node classification task using the CORA graph dataset.

Tasks:

1. Study Kipf's original GCN paper (https://arxiv.org/abs/1609.02907) to understand the architecture and operations involved in GCNs.
2. Implement the GCN architecture (get help from the paper) with two GCN layers with 16 units each, RelU activation function and dropout rate = 0.5 .
3. Train the GCN model on the CORA graph dataset for the node classification task(Optimizer=Adam and Learning Rate = 0.01).
4. Evaluate the performance of the GCN model using the same evaluation metrics as in Part 1.

Deliverables:

1. Python code implementing the GCN architecture based on Kipf's paper.
2. A pdf report detailing the implementation approach, training process, and evaluation results of the GCN model.
3. In the same pdf report a comparison between the performance of the LR model (from Part 1) and the GCN model, along with any insights gained from the comparison.

Please note the following carefully:

1. Your code files should be in two separate folders named LR and GCN and there should be two python code files LR.py and GCN.py in respective folders, and should NOT take any input arguments. Include the dataset in your submission.

2. The py files should output a text file each, which contains the evaluation metrics. Name the output files “lr_metrics.txt” and “gcn_metrics.txt”. Generate the files inside the folders called LR and GCN respectively.

3. Report your insights during the training and evaluation process and comparisons of LR and GCN performance in a separate pdf file called “Analysis.pdf”.