Mini-project 3

Advanced Machine Learning (02460)
Technical University of Denmark
Mikkel N. Schmidt

April 19, 2024

1 Formalities

This is the project description for the third mini project in *Advanced Machine Learning (02460)*. The project is part of the course exam and will count towards your final grade.

Deadline You must submit your report as a group electronically via DTU Learn by 2 May 2024 at 12:00 (noon).

Groups You must do the project in groups of 3–4 people. You need an exception to deviate from this group size. You do not need to document individual contributions.

What should be handed in? You must hand in a single report in PDF format and your code in a single file (either a zip or tar archive).

Length The report must follow the published LaTeXtemplate and consist of:

1. A single page with the main text, including figures and tables. This page must include names, student numbers, course number and the title "Mini-project 3" (so do not include a front page).
2. Unlimited pages of references.
3. A single page of well-formatted code snippets.

You must use at least font size 10pt and margins of at least 2cm. Any content violating the space limitation will not be evaluated.

Code You may use all code you were given during weeks 1–11 in the course. If you use code from elsewhere, it must be documented in the report.

2 Project description

In this project you will implement a generative model for graphs based on the techniques discussed in the course. The model must be able to generate graph adjacency matrices (not node features). You may or may not use the node features in your solution.

2.1 Dataset

We will use the *MUTAG dataset* introduced by Debnath et al.: a collection of nitroaromatic compounds (molecular graphs) and the task is to predict their mutagenicity on Salmonella typhimurium (graph-level binary classification). Vertices represent atoms and edges represent bonds, and the 7 discrete node labels represent the atom type (one-hot encoded). There are a total of 188 graphs in the dataset.

2.2 Baseline

As a baseline, you must implement an Erdös-Rényi model in the following way:

1. Sample the number of nodes N from the empirical distribution of the number of nodes in the training data.
2. Compute the link probability r as the graph density (number of edges divided by total possible number of edges) computed from the training graphs with N nodes.
3. Sample a random graph with N nodes and edge probability r according to the Erdös-Rényi model.

2.3 Deep generative model

Implement at least one deep generative model. Among the models discussed in the course, this could be a VAE with node-level latents, a VAE with a graph-level latent, or a GAN. As a component of your implementation, you must use either a message passing graph neural network or a graph convolution neural network.

The report must include a technical description of your model, with enough information to replicate your results. Include code snippets of central components of your implementation.

2.4 Sample evaluation metrics

You must sample 1000 graphs from the **baseline** and from the **deep generative model**.

Novelty and uniqueness: First, you must include a table in which you compare the baseline and the deep generative model in terms of the following metrics:

1. *Novel:* The percentage of the sampled graphs that are different from the training graphs. This metric measures to which degree your generative model is overfitted to the training data.
2. *Unique:* The percentage of the sampled graphs that are unique. This metric measures if your generative model is capable of generating a diverse set of graphs

3. *Novel and unique:* The percentage of the sampled graphs that are novel and unique.

	Novel	Unique	Novel+unique
Baseline	42%	42%	42%
Deep generative model	42%	42%	42%

To compute these metrics, you need to compare graphs to see if they are isomorphic. To do this, use the Weisfeiler-Lehman algorithm, e.g. using the Python package NetworkX.

Graph statistics: Second, you must compare statistics of the generated graphs to the statistics of the empirical distribution of the training graphs. You must do this by plotting histograms of the following statistics for the baseline, deep generative model, and the empirical distribution.

1. Node degree
2. Clustering coefficient
3. Eigenvector centrality

The histograms should be aligned so that it easy to make a visual comparison.