Topics in Deep Learning: Assignment 2

Deadline: 23rd March 2022 (11:59 pm)

Charu Sharma & Makarand Tapaswi

Total marks: 100

Instructions:

1. This assignment is an individual submission, NOT a group activity.
2. Total Marks of the assignment is 100 with duration of two weeks.
3. The submission should be in a pdf file format (preferably in latex). Handwritten figures and equations can be added.
4. Binary evaluation will be conducted for subparts (like 1.1, 1.2, etc.) of the questions.
5. For queries, tag and reach out to TAs via Course MS Teams Group Channel – Assignment Queries. Please tag instructors while asking them questions.
6. Submissions should be done via Moodle only.
7. Plagiarism will lead to an F grade in the course.

Question 1: 25 marks

Take a graph G = (V, E) of your choice (with at least 8 nodes). In order to get the best node embeddings, we want to run three random walk approaches on G i.e., DeepWalk, Node2Vec and Struct2Vec. Please write all your assumptions clearly. You are expected to draw the graph and solve the problem without running any kind of code.

1. Run DeepWalk, Node2Vec and Struct2Vec on G for two to five iterations (show all steps). Characterize the vector representations of the nodes. 15 marks
2. With the help of 1, analyze and find out which algorithm will give better representation and why? Mention which algorithm captures what. Explain for each algorithm, why it works or why it doesn't work for G. 5 marks
3. Now, considering random walk as a matrix on G, provide transition matrix T for all nodes in G and initial distribution vector v. Compute stationary distribution vector v with the help T and initial distribution. 5 marks

Question 2: 25 marks

Take a graph G = (V, E) of your choice (with at least 10 nodes). In order to get the qualitative node representations, we want to run *n*-layer GNN (GCN or GraphSAGE or any variant) for G i.e., 1-layer, 2-layer and 3-layer GNN. Please write all your assumptions clearly. You are expected to draw the graph and solve the problem without running any kind of code.

1. Run 1-layer, 2-layer and 3-layer GNN (GCN or GraphSAGE or any) for a few iterations (at least three). Show all the steps, write flow of the model and vector representations of nodes. Please mention if you are considering or ignoring any parameters for simplicity. 15 marks
2. Write your observations with reasons about the model which is best for your graph. What is the problem with the other n-layer models? Please note that the best model should be able to

capture structure and semantics of the graph while maintaining unique representation for each node. 5 marks

3. Elaborate over-smoothing problem and receptive field for your graph G. At which point over-smoothing occurs for G (you can show steps where a model leads to over-smoothing and has similar representations for nodes in G)? What is the optimal size of receptive field for G and why? Does skip-connection help in your *n*-layer model for G? If yes, how? (Please provide an answer specifically for your graph) 5 marks

Question 3: 50 marks

Implement and evaluate abstractions of a GNN. You may use the base libraries of pytorch or tensorflow, however, directly using graph libraries for implementing the models (e.g., pytorch-geometric) is not allowed. You may use the graph libraries for helping with data loading or other non-model specific parts. Training some of the models below may take some computation time, so get started sooner than later.

1. Implement a base class that can support GNNs of different kinds with the 4-step approach (initialization, aggregation, combination, output) that we discussed in class. Build your code such that it is re-usable for some of the questions below. 10 marks
2. Implement the Graph Convolutional Network (GCN) by inheriting from the base class above. Train and evaluate using the standard protocol on the semi-supervised setup of the Citeseer citations dataset (see the GCN paper) [A, B]. Analyze and report on the difference in performance for three types of adjacency matrix normalizations mentioned in class: row, column and symmetric. Discuss what causes the differences. 10 marks
3. Compare the GCN above against another GNN variant – take your pick from models that are covered (GraphSAGE) or other models that will be covered soon (GIN, GAT, etc.). Analyze and report on how the performance compares against GCN. Also play with a few options (more/deeper layers or wider layers) to change the number of learnable parameters and analyze (report and discuss) impact on performance. 10 marks
4. Implement a vanilla Recurrent Neural Network (RNN) as an abstraction of the same class from part 1. Evaluate the model's performance on the IMDb sentiment classification task. This blog [C] may be a useful starting point. 10 marks
5. Upload the assignment code as a self-explanatory and readable zip/targz package such that TAs can easily run your scripts and replicate results. There should be 3 commands that can replicate questions 2, 3, and 4. Set random seeds, download the dataset, do everything that is necessary for reproducibility. Provide the necessary python packages in a requirements.txt file. Include all this info in a markdown README. For an example of such a README.md, look at [D]. (This is practice for all your future research activities including the project as publicly releasing research code has become standard practice in ML). 10 marks

[A] Link to Citeseer dataset: https://linqs.soe.ucsc.edu/data
[B] GCN paper (which I hope everyone has already read): https://arxiv.org/abs/1609.02907
[C] Blog post on Sentiment classification on IMDb https://towardsdatascience.com/a-beginners-guide-on-sentiment-analysis-with-rnn-9e100627c02e
[D] GCN code repository https://github.com/tkipf/gcn