ECS 189G-001

Deep Learning

Winter 2024

Course Project: Stage 5 Report Example

Team Information

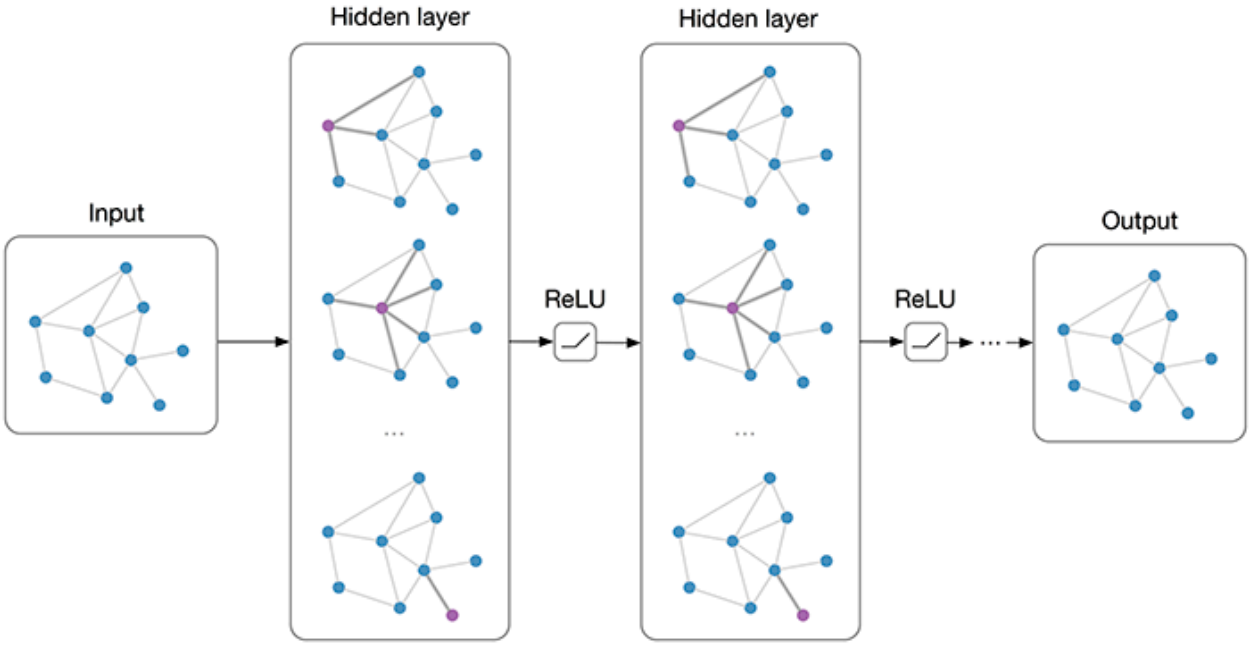
Enter Your Team Name Here (delete the extra rows if your team has less than 4 students)		
Denis Savytski	920000274	dsavytski@ucdavis.edu

Section 1: Task Description

Perform node classification task using Graph Convolutional Network (GCN) on 3 different datasets. Find optimal parameters for better performance.

Section 2: Model Description

The models for all of the datasets use standard GCN models in their base case. (More complicated models are explored in the last section of this document).



In the picture the input is represented as a graph, in reality, it is a distinct set of nodes and an adjacency matrix. We utilize 2 Graph Convolutional layers (denoted as hidden layers). ReLU AC is used between the layers, dropout is utilized after the first ReLU() function, before the last hidden layer. The output layer AC is logarithmic softmax. The dimensions of the hidden layers will differ depending on the dimensions of the input and output layers.

Section 3: Experiment Settings

3.1 Dataset Description

All of the datasets have a similar structure. They are divided into two files: node, and link. The prior has features and labels inside of them, each observation has its own line, each feature divided by space. The very first feature is the node's ID, which is unique and is also used in the link file. All the next values, excluding the very last one, are features (whose number differs depending on the dataset). The last value represents the label, that we want to classify. Link file is a file that represents the edges, by their ID. Each row is a space-separated edge that contains the link between the vertices. All of the datasets have been partitioned into train/test datasets, each containing 20 observations of every category.

Every dataset represents scientific publications classified into different classes. Publications are taken from their respective publication platforms. Feature in all datasets indicates the presence or absence of a specific word in the publication and takes a value of 1 or 0.

Cora:

The dataset contains 2708 publications, 1433 features for every observation, and 7 labels.

CiteSeer:

3312 publications, 3703 features, and 6 labels.

PubMed:

19712 publications, 500 features, and 3 labels.

3.2 Detailed Experimental Setups

For all of the models:

- Learning rate = 0.01
- Number of epochs = 100
- CrossEntropy as Loss function
- Hidden layer AC: ReLU
- Output Layer AC: log_softmax
- Hidden layers dimensions: [num_of_features, 16], [16, num_of_labels]
- Full batch GD is used for every model
- ADAM optimizer

3.3 Evaluation Metrics

In the experiment, we will use Accuracy, Precision, Recall, F1 score as our evaluation metrics.

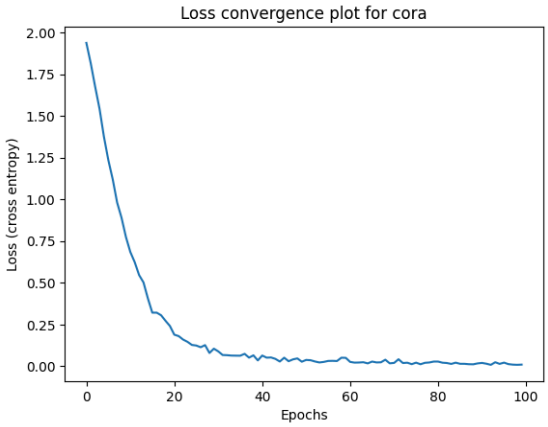
3.4 Source Code

GitHub Link

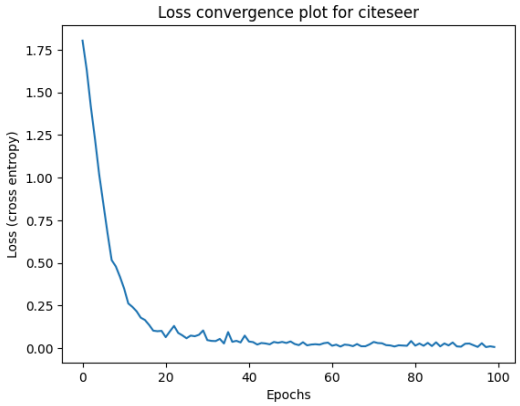
3.5 Training Convergence Plot

All plots are presented for the basic models, with 2 HL.

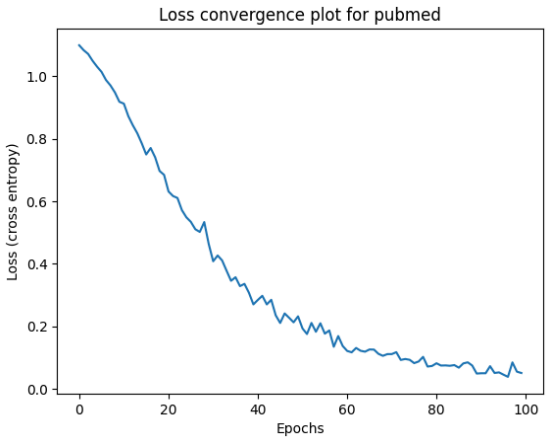
- **Cora**



- **CiteSeer**



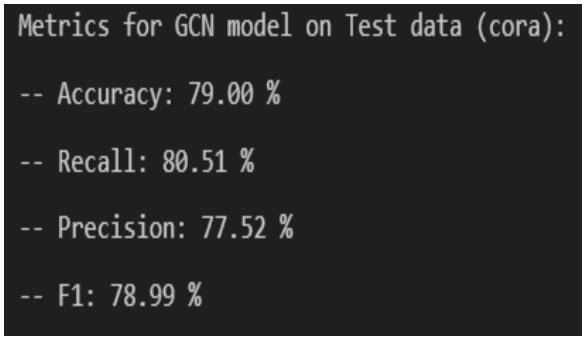
- **PubMed**

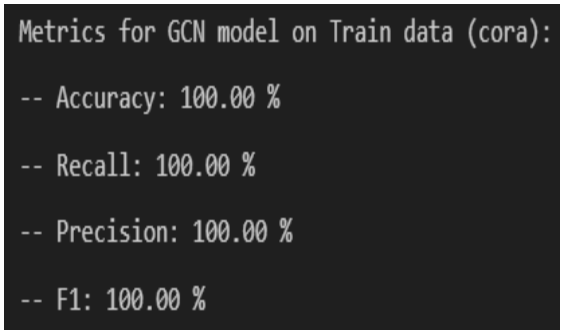


3.6 Model Performance

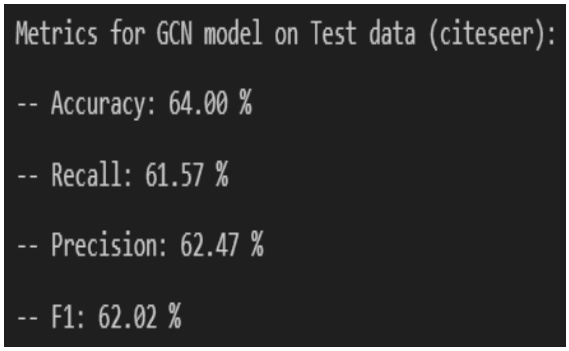
All the data presented for the basic models, with 2 HL.

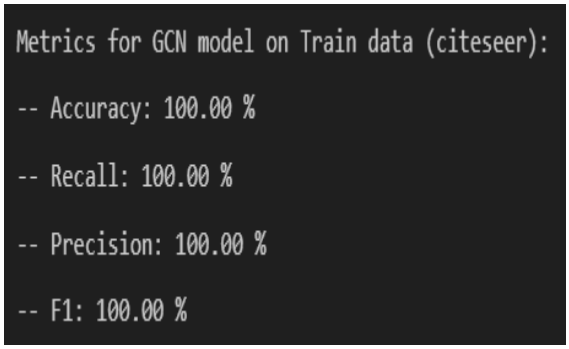
- **Cora**



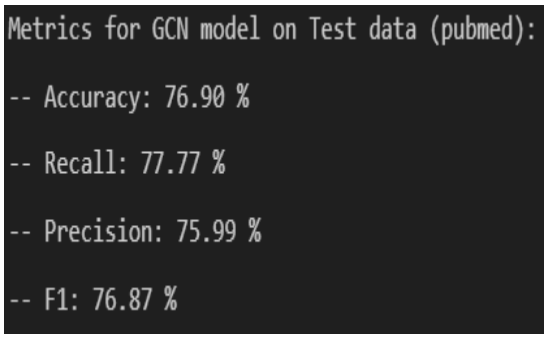


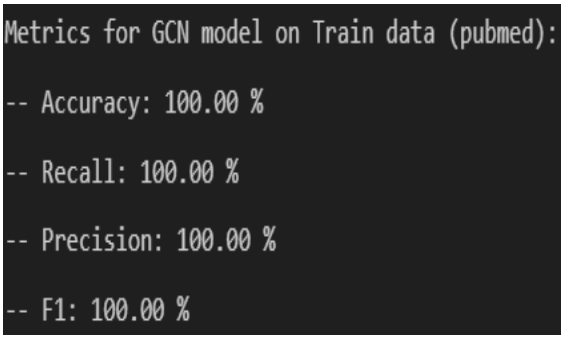
- **CiteSeer**





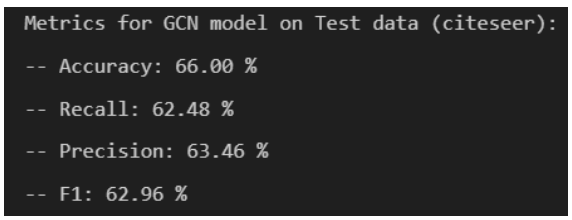
- **PubMed**





3.7 Ablation Studies

We will not change the batch size and will perform full batch in all the other models. As our first change, we would like to change loss function to NLLLoss. This action does not change significantly the accuracy of the models, but it able to slightly increase performance for the CiteSeer dataset, getting above 65%:



In further studies, we will utilize cross-entropy lf, and try to improve performance on the two remaining datasets. We will first add another layer, and change AF to sigmoid and SoftMax. This yielded a 75% accuracy on Cora and 77% on PubMed. After laying around more, and adding a weight decay of 5e-4 to the Adam optimizer, got the Cora dataset above 80% in the default configuration, using NLLLoss instead of CrossEntropy.

```
Metrics for GCN model on Test data (cora):
-- Accuracy: 80.10 %
-- Recall: 82.09 %
-- Precision: 78.22 %
-- F1: 80.11 %
```

Same parameters also increased accuracy of pubmed to

```
Metrics for GCN model on Test data (pubmed):
-- Accuracy: 79.30 %
-- Recall: 79.31 %
-- Precision: 78.85 %
-- F1: 79.08 %
```