ECS 189G-001

Deep Learning

Winter 2024

Course Project: Stage 5 Report Example

Team Information

NeuralNinjas		
Student 1: Srihita Ramini	Student 1: 919221527	Student 1: sramini@ucdavis.edu
Student 2: Trishna Sharma	Student 2: 918135782	Student 2: tksharma@ucdavis.edu
Student 3: Sai Sindura Vuppu	Student 3: 918091017	Student 3: svuppu@ucdavis.edu
Student 4: Olivia Shen	Student 4: 919293157	Student 4: oshen@ucdavis.edu

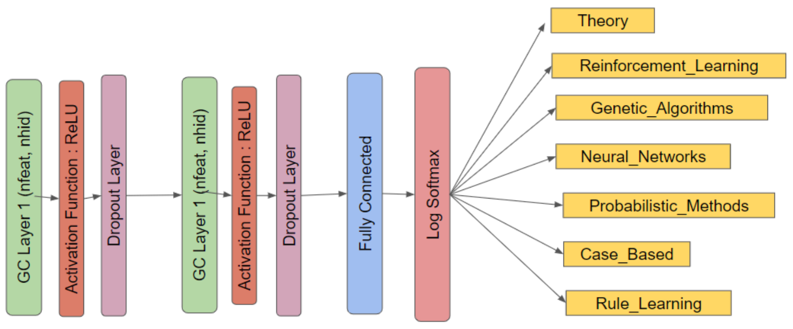
Section 1: Task Description

For Stage 5, we are asked to train three graph datasets by building a GCN model for each. To do this we build a graph embedding model and train it with the graph to classify the nodes. Then we apply it to a testing set and measure the accuracy.

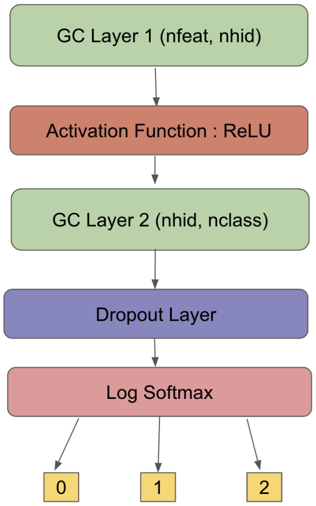
We work with three directed datasets, namely Cora, Citeseer, and Pubmed. These are graph datasets that are used to test and benchmark a model's performance. Each dataset contains a node and a link file.

Section 2: Model Description

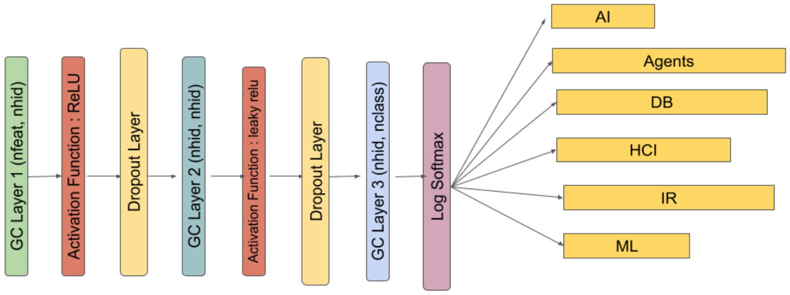
Cora: 2-layer GCN + FC output layer



Pubmed: 2-layer GCN



Citeseer: 3-layer GCN



Section 3: Experiment Settings

3.1 Dataset Description

Each dataset contains a node and a link file. The node file contains all the nodes in the graph, the node ID, the features, and the classification label of the node. The link file contains the directed links between the nodes of the graph.

Cora: The Cora dataset is a graph where each node represents a scientific paper and each edge where one paper cites another. It has 2708 nodes, each with 1433 features. There are 7 unique labels for nodes. It also has 5429 directed edges. We create a balanced training set of 140 nodes by sorting indices by class, applying random permutation, and selecting the indices [:20] of each shuffled class list. Then we select indices [21: 21+150] to create a balanced testing set of 1050 nodes. We use set random seed to 42 for reproducibility.

Citeseer: The Citeseer dataset is the same concept as Cora. The node file has 3312 nodes each with 3703 features. There are 6 unique types of nodes. The link file contains 4715 directed edges. We took 120 random training samples (with 20 from each of the 6 classes), and selected another 1200 (200 instances of each of the 6 classes) for our testing set.

Pubmed: The Pubmed dataset contains papers about 3 types of diabetes. The node file has 19717 nodes, each with 500 features. The link file contains 44324 directed edges. We randomly sampled 60 instances for our training set (20 instances from each of the 3 classes), and 600 nodes (200 instances of each of the 3 classes) for our testing set.

3.2 Detailed Experimental Setups

For our Graph Convolution Network (GCN) models, we use Graph Convolution implementation from PyGCN (https://github.com/tkipf/pygcn).

Cora: We use a 2-layer GCN (graph convolution network) architecture with one fully connected output layer. GC layer 1 has size (1433,100) and GC layer 2 has size (100,100). Each GC layer is followed by a ReLU activation function and 0.5 dropout. The FC layer has size (100,7) and is followed by a log softmax function to produce a probability distribution across 7 classes. The model is trained with NLL Loss for 80 epochs at a learning rate of

1e-3 using Adam optimizer with weight decay of 5e-4. All random seeds are fixed to 42 for consistent results. For testing, we set the model to evaluation mode to disable dropout.

Citeseer: We use a 3-layer GCN architecture. GC layer 1 has size (3703,128), GC layer 2 has size (128,128), and GC layer 3 has size (128,6). Each GC layer is followed by a ReLU activation function and 0.7 dropout, except for the last layer which uses log softmax function to produce a probability distribution across 6 classes. The model is trained with NLL Loss for 60 epochs at a learning rate of 1e-3 using Adam optimizer with weight decay of 5e-4. All random seeds are fixed to 16 for consistent results.

PubMed: We use a 2-layer GCN architecture. GC layer 1 has size (500,16) and GC layer 2 has size (16,3). GC layer 1 is followed by a ReLU activation function and 0.5 dropout. GC layer 2 is followed by log softmax function to produce a probability distribution across 3 classes. The model is trained with NLL Loss for 70 epochs at a learning rate of 2e-2 using Adam optimizer with weight decay of 5e-4. All random seeds are fixed to 42 for consistent results.

3.3 Evaluation Metrics

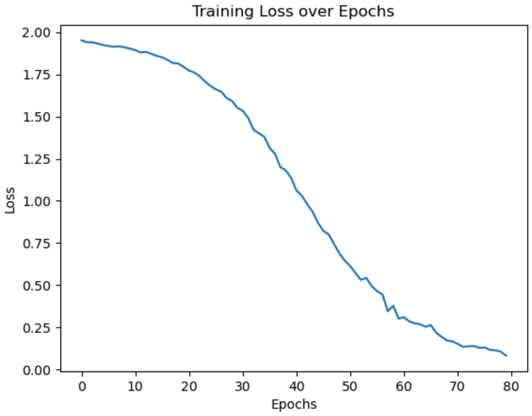
In the experiment, we will use Accuracy, Precision, Recall, F1 score as our evaluation metrics.

3.4 Source Code

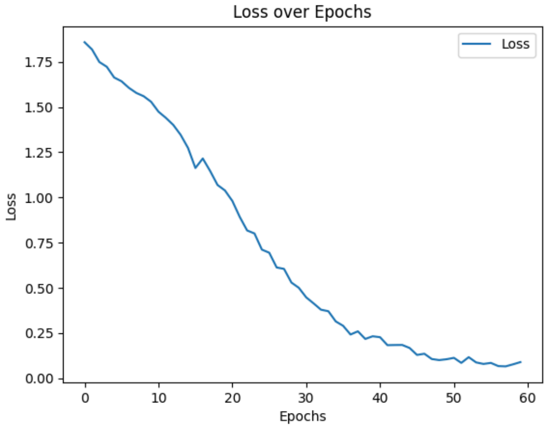
Source code: https://github.com/srihita123/ECS-189G-Deep-Learning

3.5 Training Convergence Plot

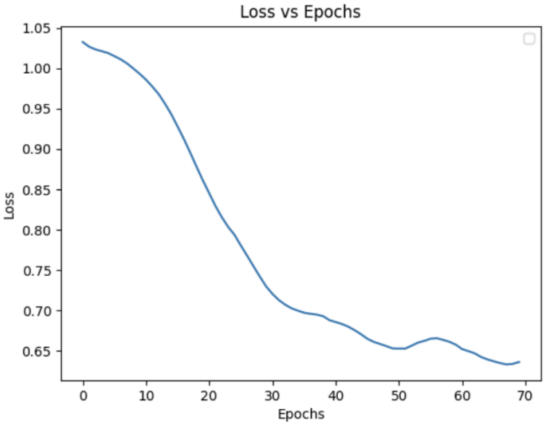
Cora:



Citeseer:



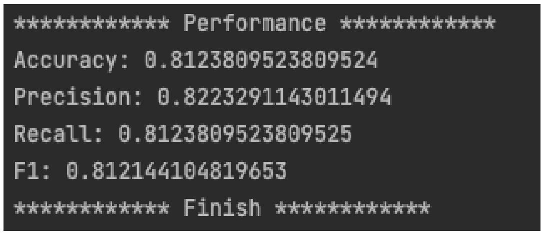
Pubmed:



3.6 Model Performance

Cora:

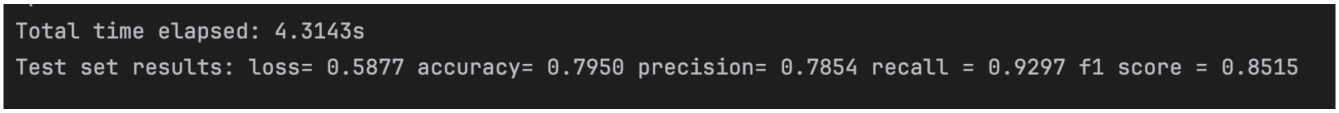
The Cora dataset performed well with 2 GCN layers, for 80 epochs, learning rate of 1e-3 and weight decay of 5e-4 with Adam optimizer. All seeds were set to 42. It gave an accuracy of 81.2%



```
************ Performance ************
Accuracy: 0.8123809523809524
Precision: 0.8223291143011494
Recall: 0.8123809523809525
F1: 0.812144104819653
************ Finish ************
```

PubMed:

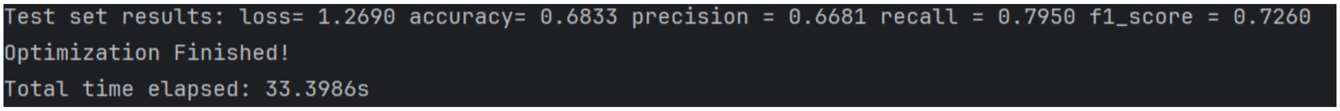
The Pubmed dataset performed with an accuracy of 79.5%. With 2 GCN layers, with 70 epochs, learning rate of 0.02, weight decay of 5e-4 using Adam Optimizer.



```
Total time elapsed: 4.3143s
Test set results: loss= 0.5877 accuracy= 0.7950 precision= 0.7854 recall = 0.9297 f1 score = 0.8515
```

Citesser:

The citeseer dataset performed with an accuracy of 68.3% with 3 GCN layers, 200 epochs, learning rate of 0.001, L2 regularization with weight decay of 5e-4.



```
Test set results: loss= 1.2690 accuracy= 0.6833 precision = 0.6681 recall = 0.7950 f1_score = 0.7260
Optimization Finished!
Total time elapsed: 33.3986s
```

3.7 Ablation Studies

Cora:

Changes Made	Accuracy	Precision	Recall	F1 Score
Dropout = 0.0	0.795	0.8	0.795	0.792

Learning rate = 5e-2	0.788	0.792	0.788	0.787
Output activation = softmax Loss function = cross entropy	0.78	0.799	0.78	0.783
GC 1 output size = 50 GC 2 output size = 50	0.52	0.547	0.52	0.5
Epochs = 40	0.58	0.757	0.58	0.535
GC 1 activation = Sigmoid GC 2 activation = Sigmoid	0.279	0.587	0.287	0.232

PubMed:

Change Made	Accuracy	Other Metrics

Changed activation function to Leaky ReLU	0.792	Precision= 0.8810
		Recall = 0.9250
		F1 Score = 0.9024

Changed epochs to 30	0.756	Precision= 0.8605
		Recall = 0.9343
		F1 Score = 0.8959

Changed dropout rate to 0.5	0.786	Precision= 0.8960
		Recall = 0.9188
		F1 Score = 0.9073

Citeseer:

Changes Made	Accuracy	Other Metrics
Learning rate = 0.01, dropout = 0.5	0.6133	precision= 0.7037, recall = 0.6650, f1 score = 0.6838
Hidden Units = 200, dropout = 0.5	0.6233	precision= 0.6746, recall = 0.7050, f1 score = 0.6895
Random seed = 42, learning rate = 0.002	0.6250	precision= 0.6940, recall = 0.6350, f1 score = 0.6632
Input Layer Activation Function = Tanh	0.6533	precision= 0.7461, recall = 0.7200, f1 score = 0.7328
Input Layer Activation Function = Tanh Hidden Layer Activation Function = ReLU	0.6183	precision= 0.4624, recall = 0.6450, f1 score = 0.5386