**DELHI TECHNOLOGICAL UNIVERSITY**

****

**SYNOPSIS**

**CWS PROJECT**

# **Graph Convolutional Networks**

DISCRETE STRUCTURES

Code:- IT-205

Faculty :- Mrs. Swati Sharda

**Submitted by:**

**Rahul Jain (2K19/IT/103)**

**Satvik Dixit (2K19/IT/116)**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**DELHI TECHNOLOGICAL UNIVERSITY**

**(Formerly Delhi College of Engineering)**

**Bawana Road, Delhi-110042**

**CERTIFICATE**

I hereby certify that the project dissertation titled “Graph Convolutional Networks” which is submitted by Rahul Jain (2K19/IT/103) and Satvik Dixit (2K19/IT/116) in Information Technology, Delhi Technological University, Delhi in partial fulfilment of the requirement for the completion of the third semester of their degree, is a record of the project work carried out by the students under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any other project

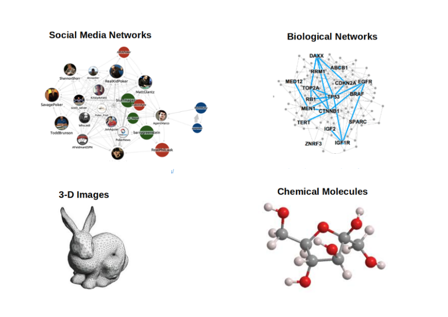
Date: 1 st December 2020

**Mrs . Swati Sharda**

**Supervisor**

**Introduction**

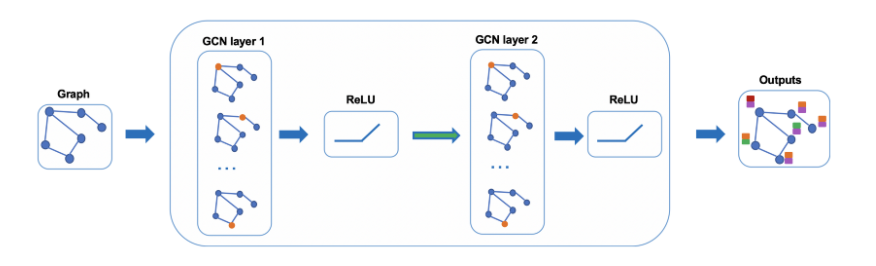
Many problems are graphs in true nature. In our world, we see many data are graphs, such as molecules, social networks, and paper citation networks. For example, in a chemical molecule that consists multiple atoms, the **atoms** can be defined as **nodes** and the **bond between atoms** can be defined as **edges**.Another example is document citation networks. The **nodes** represent individual **documents** and each **edge** represents **whether that document is cited by the other.**How the edges link the nodes allows us to distinguish between a directed vs an undirected graph. Simply put, in a directed graph, direction matters, and edges cannot be used in the other direction. Undirected graphs behave in the opposite manner, the edges follow no direction and can be used interchangeably.



## Tasks on Graphs that can be performed are:

* Node classification: Predict a type of a given node
* Link prediction: Predict whether two nodes are linked
* Community detection: Identify linked clusters of nodes
* Network similarity: How similar are two (sub)networks

**GCN** is a type of **convolutional neural network** that **can work directly on graphs** and take advantage of their structural information.It solves the problem of classifying nodes (such as documents) in a graph (such as a citation network), where labels are only available for a small subset of nodes (semi-supervised learning).



**PROBLEM STATEMENT**

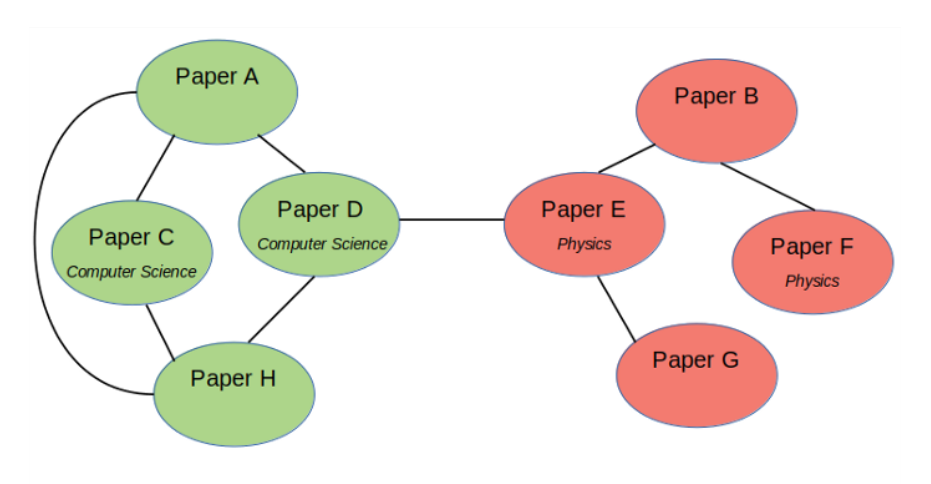
**AIM:** Training Graph Convolutional Networks on Node Classification Task with CORA, PubMed and Citeseer Dataset.

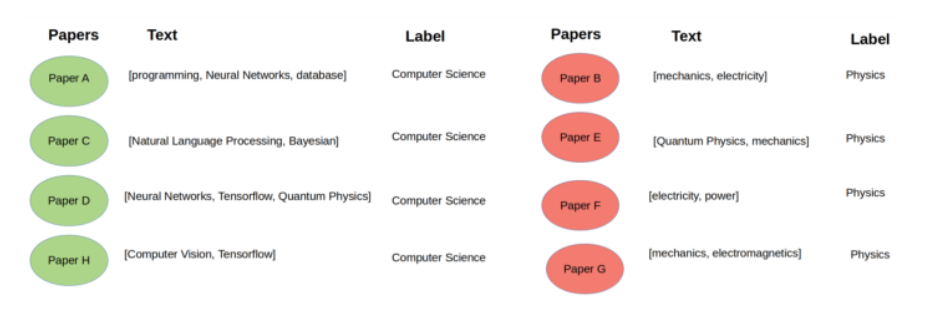
**DATASET OVERVIEW:**

**CORA :-** CORA citation network dataset consists of 2708 nodes, where each node represents a document or a technical paper. The node features are bag-of-words representation that indicates the presence of a word in the document

**PubMed :-** The PubMed Diabetes dataset consists of 19717 scientific publications from PubMed database pertaining to diabetes classified into one of three classes. The citation network consists of 44338 links. Each publication in the dataset is described by a TF/IDF weighted word vector from a dictionary which consists of 500 unique words.

**Citeseer :-** The CiteSeer dataset consists of 3312 scientific publications classified into one of six classes. The citation network consists of 4732 links, although 17 of these have a source or target publication that isn't in the dataset and only 4715 are included in the graph. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 3703 unique words.



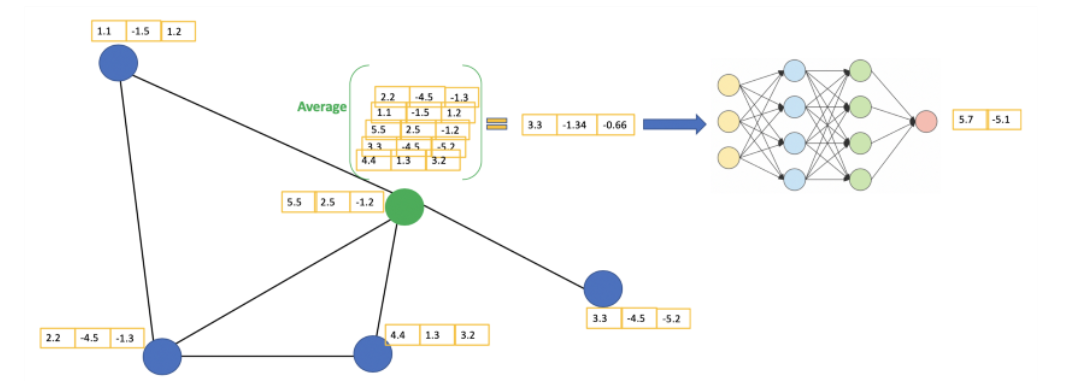


CORA citation network dataset consists of **2708 nodes,** where each node represents a document or a technical paper. The node features are bag-of-words representation that indicates the presence of a word in the document. The vocabulary — hence, also the node features — contains **1433** words.

We will treat the dataset as an **undirected graph** where the edge represents whether one document cites the other or vice versa. There is no edge feature in this dataset. The goal of this task is to classify the nodes (or the documents) into 7 different classes which correspond to the papers’ research areas. This is a single-label multi-class classification problem with **Single Mode** data representation setting.

**METHODOLOGY**

The general idea of GCN: For each node, we get the feature information from all its neighbors and of course, the feature of itself. Assume we use the average() function. We will do the same for all the nodes. Finally, we feed these average values into a neural network.



In the following figure, we have a simple example with a citation network. Each node represents a research paper, while edges are the citations. We have a pre-process step here. Instead of using the raw papers as features, we convert the papers into vectors (by using NLP embedding, e.g., tf–idf).

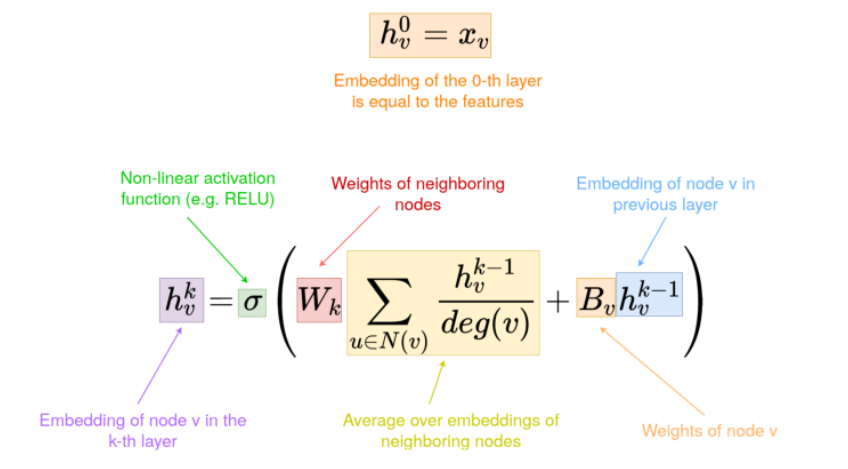
Let’s consider the green node. First off, we get all the feature values of its neighbors, including itself, then take the average. The result will be passed through a neural network to return a resulting vector.

In practice, we can use more sophisticated aggregate functions rather than the average function. We can also stack more layers on top of each other to get a deeper GCN. The output of a layer will be treated as the input for the next layer. So, how does all of this come together as a neural network? Let’s take an example of a social circle. We can first of all look just at one person itself. Then we can compile information about the friends of a person. Then information about friends of friends and so on. This is basically the idea of a graph net: we aggregate information of neighbors, and neighbors of neighbors, etc. of one node.

The major difference between graph data and “normal” data we encounter in other machine learning tasks is that we can derive knowledge from two sources:

1. Just like in other machine learning applications every node has a set of **features**. For example, when we look at a social network every node can be a person with a certain age, gender, interests, political views, etc.
2. Information is also encoded in the **structure of the graph.** By looking at friends of a person it is often possible to get some insight into this person.

**FEED FORWARD FORMULA OF GRAPH CONVOLUTIONAL NETWORKS:**

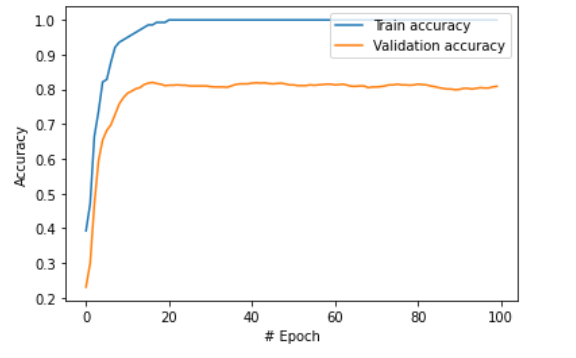
****

Intuitively, the cited papers are likely to belong to similar research area. In this citation network dataset, we want to leverage the citation information from each paper in addition to its own textual content. Hence, the dataset has now turned into a network of papers.

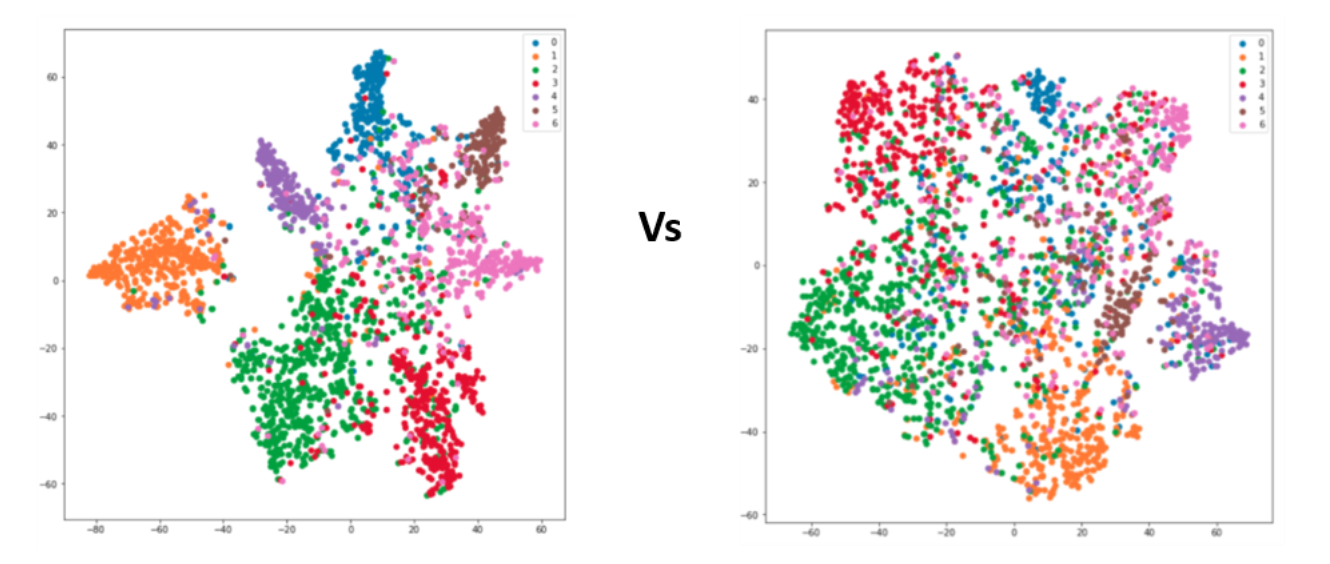
Using this configuration, we can utilize Graph Neural Networks, such as Graph Convolutional Networks (GCNs), to build a model that learns the documents interconnection in addition to their own textual features. The GCN model will learn the nodes (or documents) hidden representation not only based on its own features, but also its neighboring nodes’ features. Hence, we can reduce the number of necessary labeled examples and implement semi-supervised learning utilizing the **Adjacency Matrix (A)** or the nodes connectivity within a graph.

Another case where Graph Neural Networks might be useful is when each example does not have distinct features on its own, but the relations between the examples can enrich the feature representations.

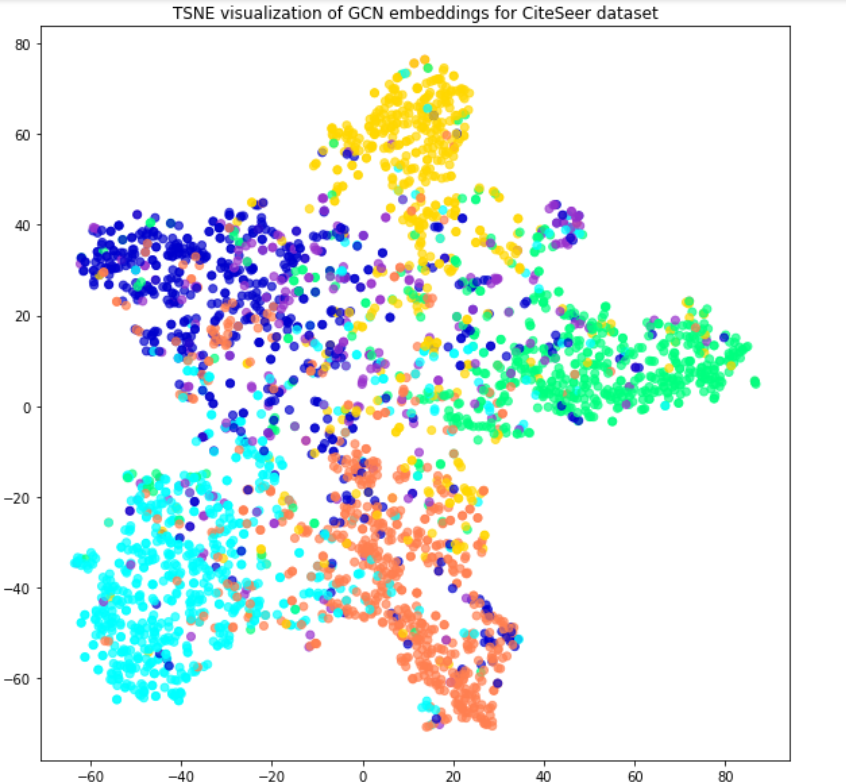
**RESULTS :-**

****

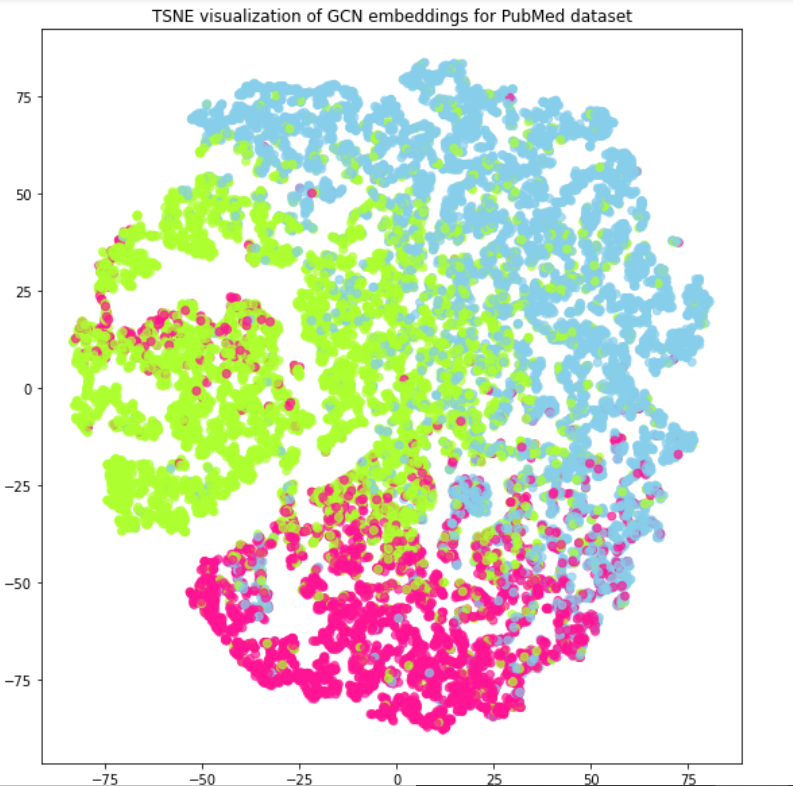
**Comparison of GCN with Fully Connected Neural Network**

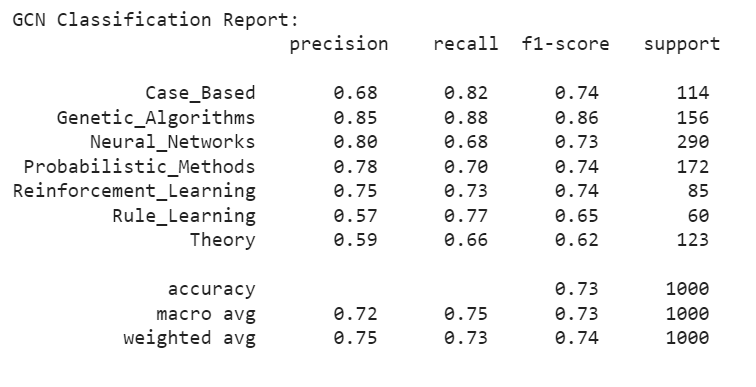
****

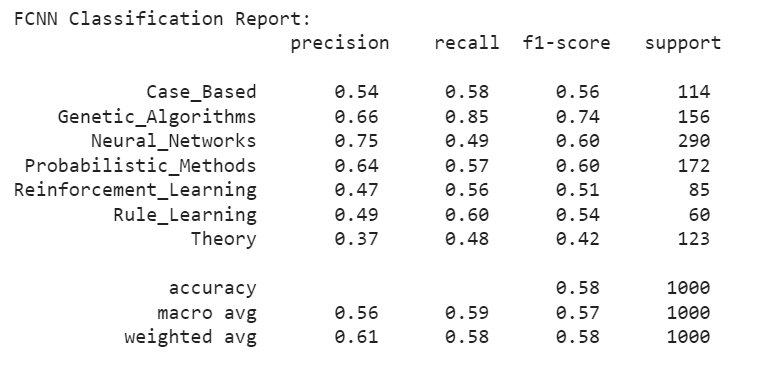
**Citeseer Dataset:**

****

**PubMed Dataset**

****

****

****

#### **From the results above, it is clear that GCN significantly outperforms FCNN with macro average F1-score is only 55%. The t-SNE visualization plot of FCNN hidden layer representations is scattered, which means that FCNN can’t learn the features representations as well as GCN.**

**REFERENCES :**

**[1] T. Kipf and M. Welling,** [**Semi-Supervised Classification with Graph Convolutional Networks**](https://arxiv.org/pdf/1609.02907.pdf) **(2017). arXiv preprint arXiv:1609.02907. ICLR 2017**

**[2]** [**http://web.stanford.edu/class/cs224w/**](http://web.stanford.edu/class/cs224w/)

**[3]**[**https://towardsdatascience.com/how-to-do-deep-learning-on-graphs-with-graph-convolutional-networks-7d2250723780**](https://towardsdatascience.com/how-to-do-deep-learning-on-graphs-with-graph-convolutional-networks-7d2250723780)