IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

Research Article Classification using Graph Convolutional Neural Network

Isha Shrivastava[1], Arun Jhapate[2]

SIRT, Bhopal, (M.P), India[1,2]

Abstract: The thesis presents a mechanism for research paper classification based on a relational graph that represents the interrelationships among papers, such as citations, authors, common references, etc. The proposed method is a semi-supervised learning that have used graph convolution neural network in learning the relations. The GCNN captures the spatial relation i.e. neighbors of a node in feature vector creation. The Proposed method makes use of message passing system, where node sends their feature values i.e. word embedding to their neighbors node so that every node can create its own feature vector based on the content of its own article and its neighboring articles. This method has better performances it tries to predict the class based on the neighboring classes and the content of its neighbor which is common between them. Comparison with the previous methods, the proposed method has performed well with a significant margin.

Keywords: GCNN, Article Classification

I. INTRODUCTION

With the growth of online research articles, classification systems play a key role in deciding the category of the document so that the document can be sent as a recommendation based on their indentified category. Due to the importance of classification systems, there have always been emerging works in this field.

The research article cites the other articles which are on same subject or used similar methodology. The research article database can be consider as a graph, in which an article act as a node and the citation act as a link between two article. We consider 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 as graph-based semi-supervised learning.

The potential applications of for research article classification are as follows:

- **News article classification:**

  News article gets generated with a massive rate, their classification is very much required as only relevant news needs to be given to a user, therefore an automated mechanism is required for classification. [1]

- **Opinion mining:**

  It is very important to analyze the information on opinions, sentiment, and subjectivity in documents with a specific topic [2].

  Analysis results can be applied to various areas such as website evaluation, the review of online news articles, opinion in blog or SNS, etc. [3].

- **Email classification and spam filtering:**

  Its area can be considered as a document classification problem not only for spam filtering, but also for classifying messages and sorting them into a specific folder [4].

Previously in this research area, a dictionary-based system is introduced that represents items in a dictionary, and based on the topics extracted by LDA.

Furthermore, it has used TF-IDF scheme that extracts the subject words frequency and applies the K Mean clustering [5-8] algorithm, that clusters the papers having similar subject based on TF-IDF value.

The system is highly scalable as it uses HDFS [9,10] that can process big data rapidly. Moreover, the system used map reduce frame work for TFIDF calculation from the abstract of each paper [9,10].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

II. BACKGROUND

This section presents a survey of previous methods that has performed the research article classification. The problem in research article classification is to assign a one or more predefined classes to a given research article.

Previously several techniques have been published for document classification [11]. The document classification is divided into types of methods, which are supervised and unsupervised [12-14]. In the supervised way, the classification of documents done by supervised learning methods.

In this we are using some methods or we can say we are using some ways to analyze the data (i.e., pair data of predefined input–output) and it will create an function which is an inferred function by which we will map other examples. And in different scenario, in this without any predefined criterion, an unsupervised classification groups documents, which will be depend on the similarity of the documents. Already there have been developed different different algorithms like Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), TF-IDF, Naive Bayes classifier and so on and these all algorithms are based on automatic document classification.

On the other hand, Bravo-Alcobendas presented a algorithm which extracts the behaviour of documents by Non-negative matrix factorization (NMF) which is basically based on a document clustering algorithm this work is based on paper classification and that groups documents by K-means clustering algorithm. This work is not mainly focuses on a sophisticated classification if we discuss about different words, we mainly focuses on the word count in documents by which high dimensional vector formed so for the reduction of high dimensional vector which is present in the documents which is formed due to word count.

In a previous work, a method is proposed by Hanyurwimfura et al for paper classification based common content and the title of research paper [15]. In one of the work, a method is proposed that group papers based on the extracted keywords of from the research objective and background of the paper.

These methods considered only the subjects, objectives and background information which is useful one. However, the works did not consider frequently occurred keywords in classification task. The title of a paper, objective and background information gives only the limited information about a paper class, which leads to a wrong decision.

In one of the article [16] of Nguyen et al. used a bag of word technique and K nearest algorithm for paper classification. In this method, topics are extracted from the contents of all paper. It is highly computationally inefficient when the volume of the data is very high.

In comparison with the previous method, our proposed method uses keywords which are extracted from the abstract of the paper, and the topics which are extracted by the LDA Scheme.

The extracted keywords are used in calculation of TF-IDF of each article. Then for classification we used K means, that clusters the article based on TF-IDF.

Furthermore, we incorporated recent advancements of deep learning methods for this graph structured like data. A graph neural network propagates the node information in iteration to other connected nodes for generating activation values of hidden layer neuron output. The process gets repeated again and again until we get output.

It was first outlined in [17] and further elaborated on in [18]. [19] proposed a graph convolution based on spectral graph theory. Following on from this work, a number of authors have proposed improvements, extensions, and approximations of these spectral convolutions [6,21] which have proved the effectiveness on node classification and link prediction, as well as recommender systems [22]. These approaches have consistently outperformed techniques based upon matrix factorization or random walks. However, the learnt filters in the spectral approaches depend on the Laplacian eigenbasis and spectral decomposition, which is prohibitively expensive on large graphs.

Moreover there are some other methods which followed non spectral approaches [23-26], that defined convolutions directly on the graph [27]. The method introduced a Graph-SAGE, a method that computes the node representation in an inductive manner. The method is performed very well on large scale inductive benchmark. However the method can only tackle only a fixed size neighborhood of each node.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

III. PROPOSED METHODOLOGY

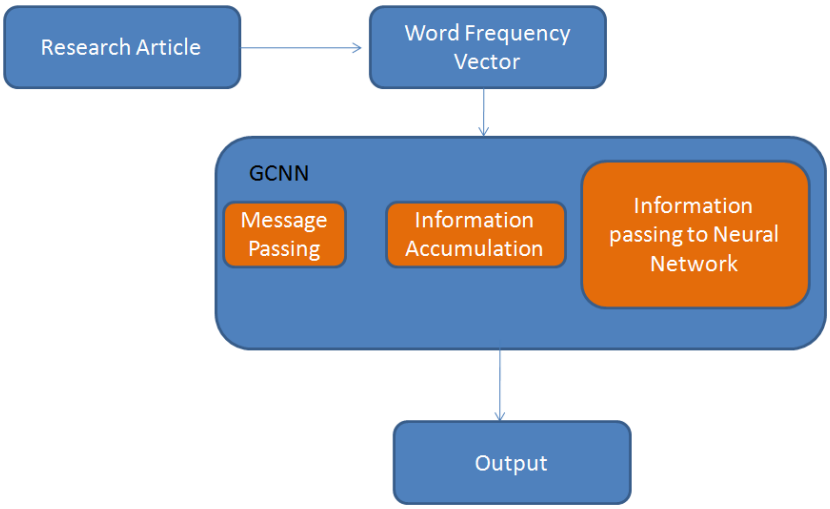
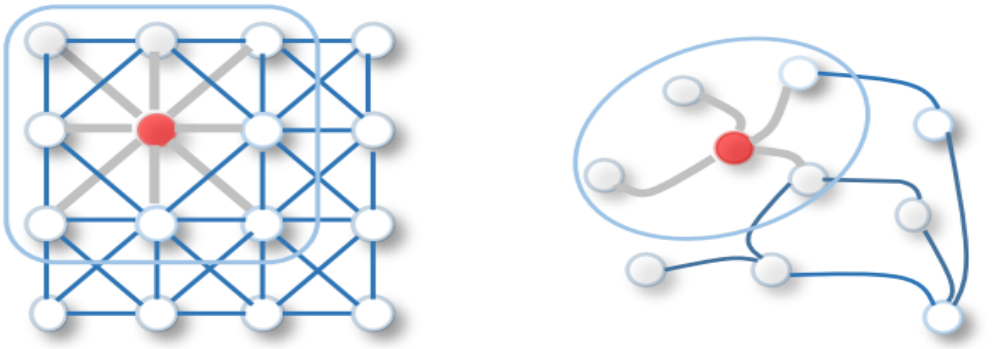


Figure 1: **Proposed Methodology for research article classification**

In this section we present a proposed mechanism for research article classification. The mechanism is shown in Figure 1. Firstly we calculate the feature vector from the content of the given documents. The feature vector contains the frequency of each individual word. The length of the vector is equal to the number of unique words. Next the word frequency feature vector and its corresponding class then passed to the GCNN for training for classification. The functionality of GCNN (graph convolutional neural networks) is discussed in below subsection.

3.1 GCNN

The graph convolutional neural network [28-39,41-48] processes a graph, that consists of nodes and edges. For node classification , spatial information needs to be captured like CNN, where each node need to capture the information of their neighbors by message passing system and Aggregates the collected information and passes the information to the neural network for classification.



(a) 2D Convolution. Analogous to a graph, each pixel in an image is taken as a node where neighbors are determined by the filter size. The 2D convolution takes the weighted average of pixel values of the red node along with its neighbors. The neighbors of a node are ordered and have a fixed size.

(b) Graph Convolution. To get a hidden representation of the red node, one simple solution of the graph convolutional operation is to take the average value of the node features of the red node along with its neighbors. Different from image data, the neighbors of a node are unordered and variable in size.

Figure 2: **How GCNN is different from CNN**

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

3.1.1 Final node representation

Applying the aggregation and update operations layer by layer generates the representations of nodes for each depth of GNN. The overall representations of users and items are required for the final prediction task. A mainstream approach is to use the node vector in the last layer as the final representation. However, the representations obtained in different layers emphasize the messages passed over different connections [40]. Specifically, the representations in the lower layer select the individual feature more while those in the higher layer select the neighbor feature more. To take advantage of the connections expressed by the output of different layers, recent studies employ different methods to integrate the messages from different layers.

- Mean-pooling
- Sum-pooling
- Weighted-pooling
- Concatenation

Compared to mean-pooling and sum-pooling, weighted pooling allows more flexibility to differentiate the contribution of different layers. Among these four methods, the former three all belong to linear operation, and only concatenation operation preserves information from all layers.

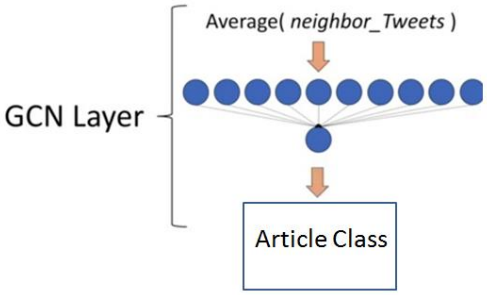


Figure 3: Averaging Feature Vector for Classification

3.1.2 Graph Construction

The most straightforward way is to directly use the original user-item bipartite graph. If some nodes have few neighbors in the original graph, it would be beneficial to enrich the graph structure by either adding edges or nodes. When dealing with large-scale graphs, it is necessary to sample the neighborhood for computational efficiency. Sampling is a trade-of between effectiveness and efficiency, and a more effective sampling strategy deserves further study.

• Neighbor Aggregation.
When neighbors are more heterogeneous, aggregating neighbors with attentive weights would be preferable to equal weights and degree normalization; otherwise, the latter two are preferable for easier calculation. Explicitly modeling the influence among neighbors or the affinity between the central node and neighbors might bring additional benefits, but needs to be verified on more datasets.

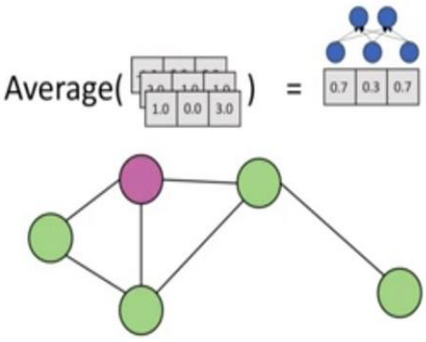


Figure 4: **Neighbor Aggregation**

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

• Information Update.

Compared to discarding the original node, updating the node with its original representation and the aggregated neighbor representation would be preferable. Recent works show that simplifying the traditional GCN by removing the transformation and non-linearity operation can achieve better performance than the original ones.

• Final Node Representation.

To obtain overall user/item representation, utilizing the representations from all layers is preferable to directly using the last layer representation. In terms of the function of integrating the representations from all layers, weighted-pooling allows more flexibility, and concatenation preserve information from all layers.

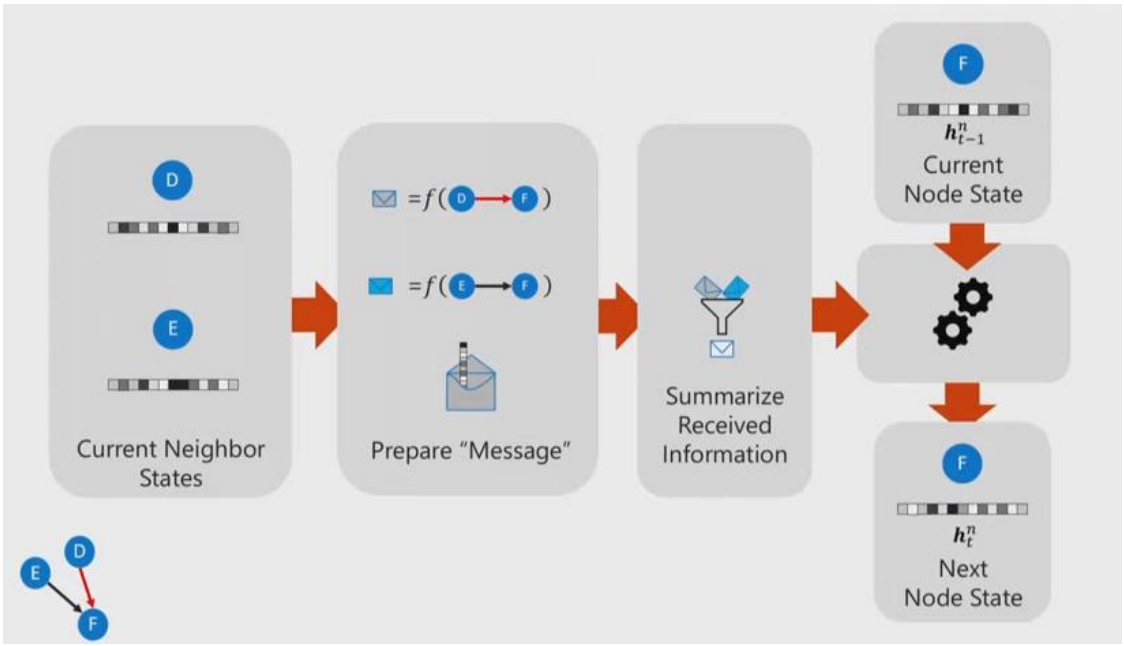


Figure 3.5: Message Passing

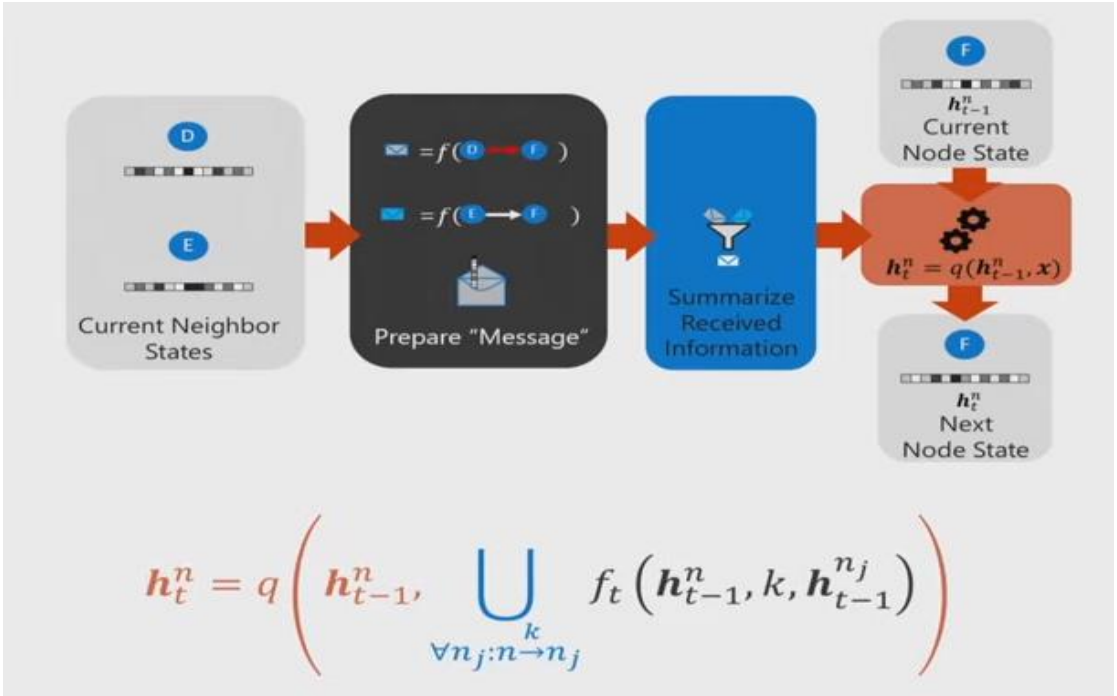


Figure 3.6: Message Summarization

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

Steps for research article classification:

1. Firstly extract the words from articles.
2. Remove stop word.
3. Construct dictionary by adding each individual unique word in the dictionary.
4. Now construct feature vector using word frequency.
5. Next construct a graph based on citation: Two articles having citation of one another will be connected with each other, as shown in figure 3.4.
6. Feature vector passing among the neighbor nodes.
7. Accumulation of feature vectors using averaging operator.
8. Next pass the accumulated feature vector to the input layer of the neural network .
9. Next pass the input vector to next hidden layer to generate activation pattern of hidden layer.
10. To generate final node representation, again share the hidden layer activation pattern to the neighboring nodes.
11. Next pass the final node representation to the next layer to generate output.

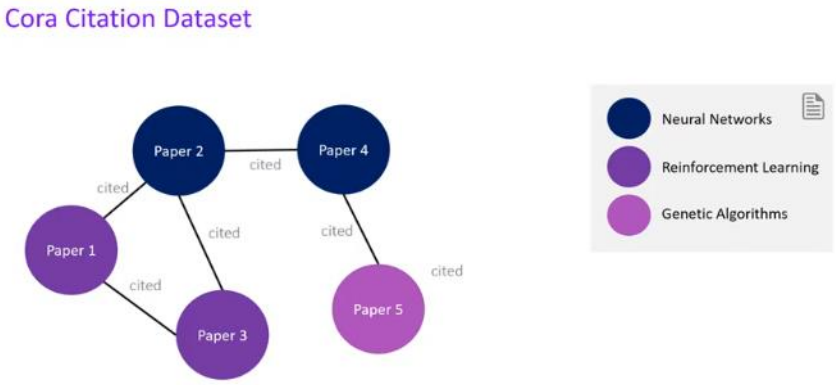


Figure 3.7: **Relational Graph of Research Article**

Here each article carries a word embedding vector that actually carries the information of words present in the document. The length of the vector is equal to the total number of words present in the dictionary.

The value in the feature vector represents the frequency of words in a document. Now the nodes sends their embedding to its connected node so that the node can accumulate the information of its neighbor. The accumulated information then passed to neural network to calculate the activation for the next hidden layer.

Each node now have own activation values which is again passed to its neighbor nodes. Now this time each node would accumulate the information of neighbors neighbor. Again the accumulated information will be passed to the next layer of neural networks to calculate the final output.

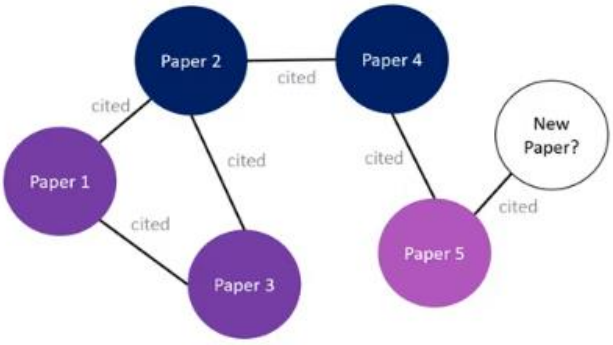


Figure 3.8: **Research article with unknown label**

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed method the model is tested on Citation network dataset. In the citation network datasets—Citeseer, Cora and Pubmed [24]—nodes are documents and edges are citation links. Label rate denotes the number of labeled nodes that are used for training divided by the total number of nodes in each dataset.

Table 4.1: Graph Datasets [29]

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003

4.1 Citation Network datasets

We consider three citation network datasets: Citeseer, Cora and Pubmed [24]. The datasets contain sparse bag-of-words feature vectors for each document and a list of citation links between documents. We treat the citation links as (undirected) edges and construct a binary, symmetric adjacency matrix A. Each document has a class label. For training, we only use 20 labels per class, but all feature vectors.

4.1.1 CORA

The CORA dataset which is citation network. The cora dataset is having 2708 scientific research articles which are classified in to seven different categories. Each article is represented by 0/1 binary vector that represents the presence of a word from a corresponding dictionary. The dictionary consists of 1433 unique words.

Nodes=Research Article
Edge=Citation
Node Features= Word Vectors

Labels= Publication type(Neural Netwoks, Rule Learning, Reinforcement Learning…..)

Node features are represented by real valued vector where each value lie between 0 and 1. Firstly we calculated the frequency of each word and then normalizes the frequency to standardize the features. Now each paper having word embedding of length 1433 as shown in figure 4.2.

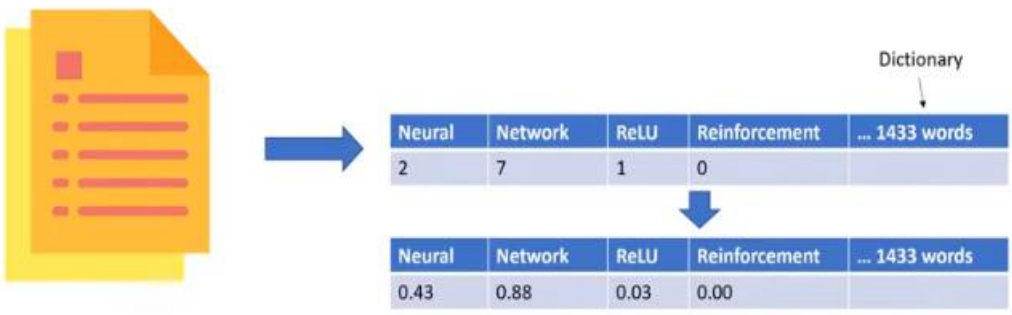


Figure 4.1: Bag of words representation

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

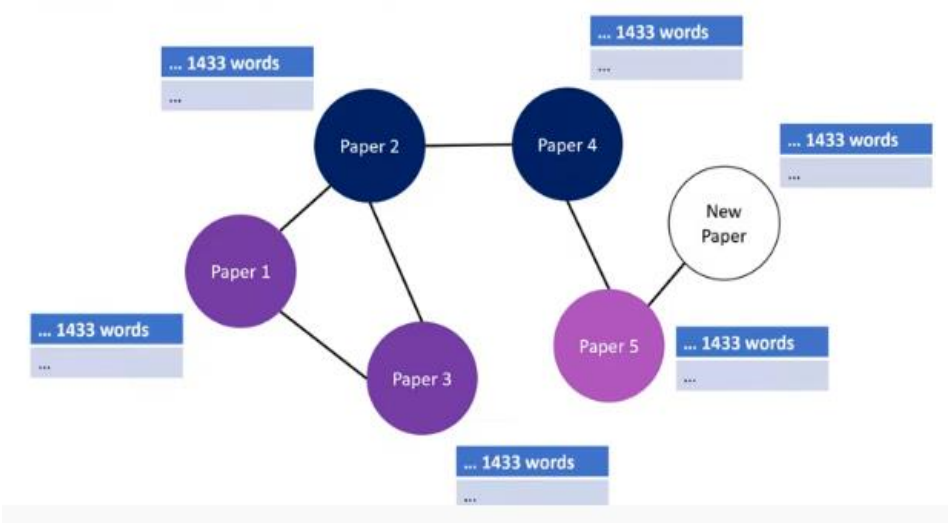


Figure 4.2: bag of word representation

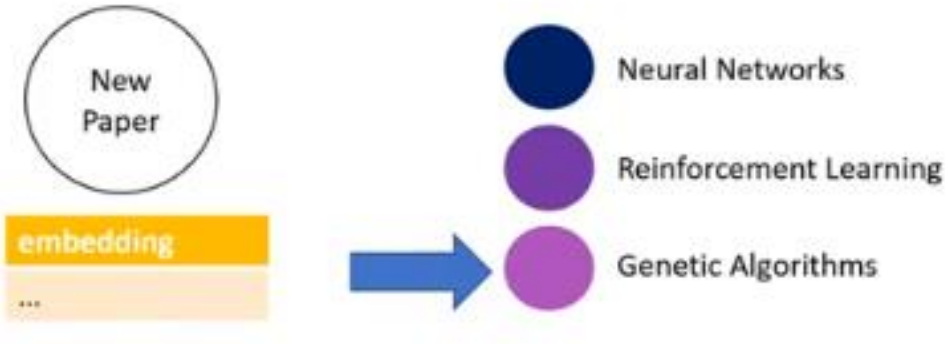


Figure 4.3: Classification based on embedding

4.1.2 Citation networks

We consider three citation network datasets: Citeseer, Cora and Pubmed [24]. The datasets contain sparse bag-of-words feature vectors for each document and a list of citation links between documents. We treat the citation links as (undirected) edges and construct a binary, symmetric adjacency matrix A. Each document has a class label. For training, we only use 20 labels per class, but all feature vectors.

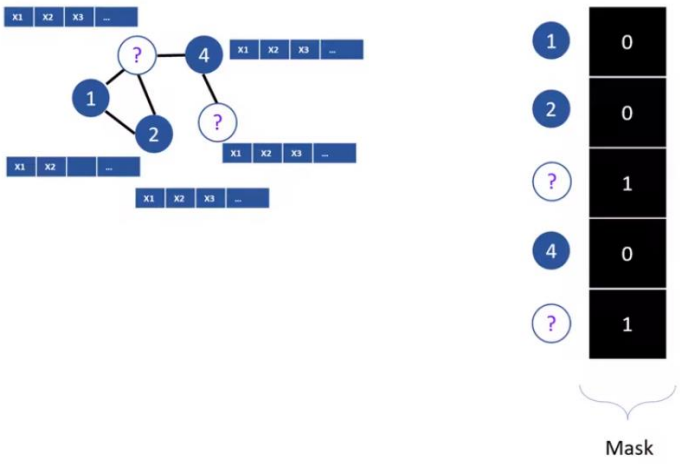


Figure 4.4 : Binary Mask for Node Level Prediction

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

4.2 Vanilla Deep Neural Network:

For classification we used vanilla deep neural network, that consists of input layer, one hidden layer and output layer. For optimization of neural weights we used ADAM optimizer [41] with learning rate of 0.001 and 1000 epoch. The loss with respect to different epoch is shown in figure 4.4.

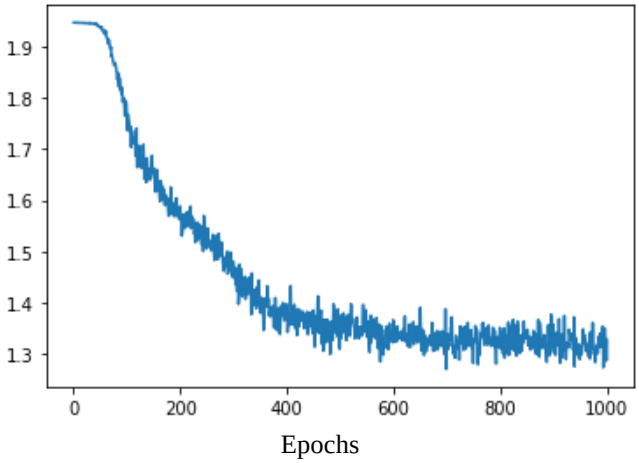


Figure 4.5: Loss With respect to different epochs

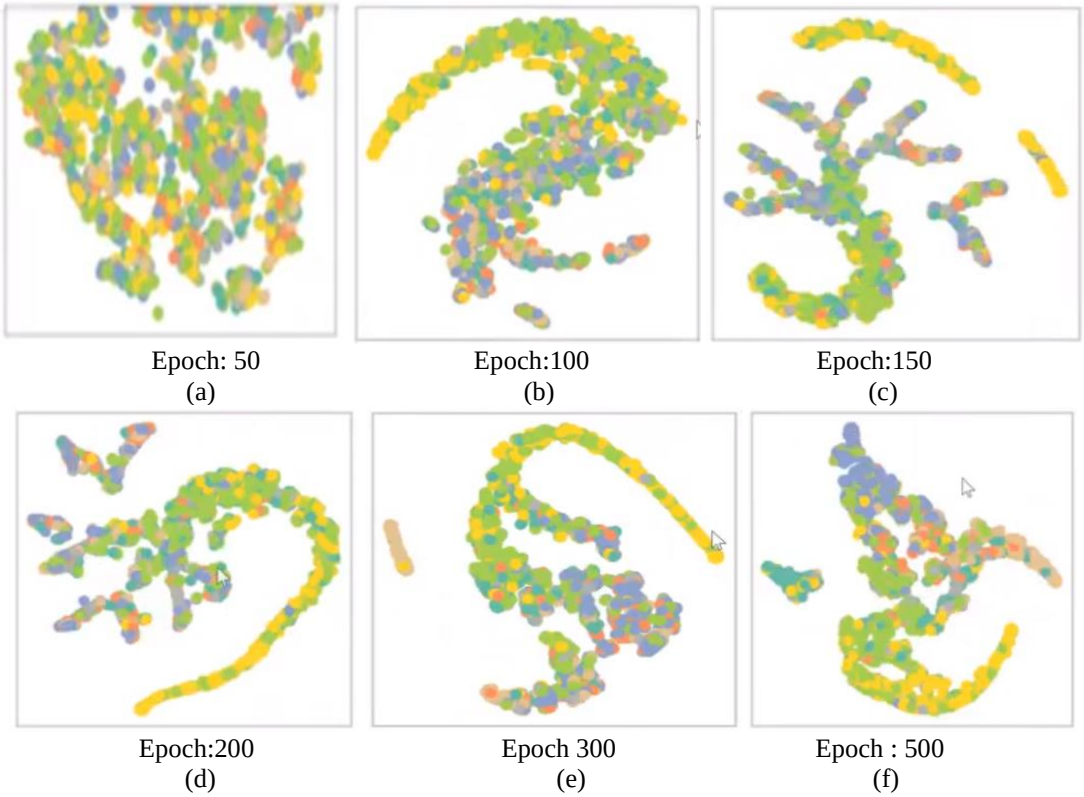


Figure 4.6: **Embedding Plot over different epochs**

We have reduced the dimension of our embedding to a two dimension so that we can visualize it in a simple 2d plot. Here in the plot we have 2708 nodes, each node with the same color belongs to a same class.

As you can see in the first plot in Figure 4.4(a), everything is spread all over the place so our embedding are completely spread but over the time or can say over different epochs our GNN improves the embedding and we can see that there

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

are some clusters so asically classes with the same embedding appear is the same area and this is actually what the goal is to have a perfect clustering which will eventually for new data points easily allow us to predict their class.

Table 4.2: **Comparative analysis of accuracy with respect to different datasets**

Dataset / Model	CORA	Citeseer	Pubmed
Proposed method	89.3	91.4	88.4
Planetoid [39]	74.8	78.45	77.34
Chebnet[40]	81.3	84.29	85.31

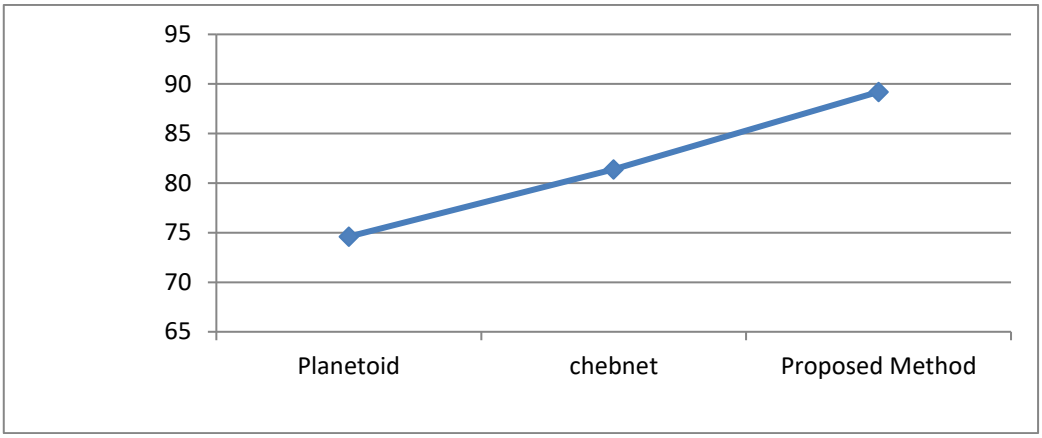


Figure 4.7: **Accuracy comparison with the previous works**

V. CONCLUSION AND FUTURE WORKS

In this thesis we presented a work on article classification. The work has used graph convolutional neural network as a classifier. Since the input is present in the form of a graph therefore we used GCNN to learn spatial relationship of citation. The GCNN has learned spatial connectivity as CNN learns and makes use of Deep Learning in making association between words frequency and document class. The proposed mechanism has performed well in comparison with the previous published works with a significant margin.

Graph-Laplacian regularization based method [3, 32,] are having drawback due to the assumption that edges encode mere similarity of nodes. On the other side, Skip-gram based methods are having limitation that they are based on a multi-step pipeline which is difficult to optimize.

Our proposed model has overcome both the issues.Propagation of feature vector to neighbor nodes improves performance of classification in comparison to methods like ICA [18], where only label information is aggregated.

In future we can add attention mechanism in graph neural network, that would learn how much attention is to given to a word while performing aggregation of feature vector. Definitely the accuracy would be increased.

The proposed mechanism is highly inefficient in time. With the increase of nodes in the graph the processing gets increase exponentially. To remove this drawback ,

we can use hadop distributed files system and distributed processing so that the each individual document gets processed parallel independent of other documents. The distributed processing on GPU will save lots of time in execution.

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022
DOI: 10.17148/IJARCCE.2022.111248

REFERENCES

[1] Mart´ın Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.
[2] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In Advances in neural information processing systems (NIPS), 2016.
[3] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of machine learning research (JMLR), 7(Nov):2399–2434, 2006.
[4] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. IEEE Transactions on Knowledge and Data Engineering, 20(2):172–188, 2008.
[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In International Conference on Learning Representations (ICLR), 2014.
[6] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr, and Tom M. Mitchell. Toward an architecture for never-ending language learning. In AAAI, volume 5, pp. 3, 2010.
[7] Michael Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on ¨ graphs with fast localized spectral filtering. In Advances in neural information processing systems (NIPS), 2016.
[8] Brendan L. Douglas. The Weisfeiler-Lehman method and graph isomorphism testing. arXiv preprint arXiv:1101.5211, 2011.
[9] David K. Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan´ Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In Advances in neural information processing systems (NIPS), pp. 2224–2232, 2015.
[10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, volume 9, pp. 249–256, 2010.
[11] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In Proceedings. 2005 IEEE International Joint Conference on Neural Networks., volume 2, pp. 729–734. IEEE, 2005.
[12] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016. 9 Published as a conference paper at ICLR 2017
[13] David K. Hammond, Pierre Vandergheynst, and Remi Gribonval. Wavelets on graphs via spectral ´ graph theory. Applied and Computational Harmonic Analysis, 30(2):129–150, 2011.
[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[15] Thorsten Joachims. Transductive inference for text classification using support vector machines. In International Conference on Machine Learning (ICML), volume 99, pp. 200–209, 1999.
[16] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR), 2015.
[17] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In International Conference on Learning Representations (ICLR), 2016.
[18] Qing Lu and Lise Getoor. Link-based classification. In International Conference on Machine Learning (ICML), volume 3, pp. 496–503, 2003.
[19] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of Machine Learning Research (JMLR), 9(Nov):2579–2605, 2008.
[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (NIPS), pp. 3111–3119, 2013.
[21] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In International Conference on Machine Learning (ICML), 2016.
[22] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710. ACM, 2014.
[23] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. IEEE Transactions on Neural Networks, 20(1):61–80, 2009.
[24] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. AI magazine, 29(3):93, 2008.
[25] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research (JMLR), 15(1):1929–1958, 2014.

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 11, Issue 12, December 2022

DOI: 10.17148/IJARCCE.2022.111248

[26] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077. ACM, 2015.

[27]Boris Weisfeiler and A. A. Lehmann. A reduction of a graph to a canonical form and an algebra arising during this reduction. Nauchno-Technicheskaya Informatsia, 2(9):12–16, 1968.

[28] Jason Weston, Fred´ eric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi- ´ supervised embedding. In Neural Networks: Tricks of the Trade, pp. 639–655. Springer, 2012.

[29]Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In International Conference on Machine Learning (ICML), 2016.

[30] Wayne W. Zachary. An information flow model for conflict and fission in small groups. Journal of anthropological research, pp. 452–473, 1977.

[31] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Scholkopf. ¨ Learning with local and global consistency. In Advances in neural information processing systems (NIPS), volume 16, pp. 321–328, 2004.

[32]Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In International Conference on Machine Learning (ICML), volume 3, pp. 912–919, 2003.

[33]F.Scarselli, M.Gori, "The graph neural network model," *IEEE Transactions on Neural Networks, 2009*

[34]T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. of ICLR*, 2017.

[35]Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, Philip S. Yu, "A Comprehensive Survey on Graph Neural Networks"

[36]D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proc. of CVPR*, vol. 2, 2017

[37]J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *Proc. of CVPR*, 2018

[38]X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *CVPR 201*.

[39]Yang, Zhilin, William Cohen, and Ruslan Salakhudinov. "Revisiting semi-supervised learning with graph embeddings."*International conference on machine learning*. PMLR, 2016.

[40]Monti, Federico, Karl Otness, and Michael M. Bronstein. "Motifnet: a motif-based graph convolutional network for directed graphs."*2018 IEEE Data Science Workshop (DSW)*. IEEE, 2018.

[41]Zhang, Zijun. "Improved adam optimizer for deep neural networks." *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. Ieee, 2018.

[42]Li, Ruoyu, et al. "Adaptive graph convolutional neural networks." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. No. 1. 2018.

[43]Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907* (2016).

[44]Zhang, Si, et al. "Graph convolutional networks: a comprehensive review." *Computational Social Networks* 6.1 (2019): 1-23.

[45]Pope, Phillip E., et al. "Explainability methods for graph convolutional neural networks." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

