Educational Administration: Theory and Practice
2024, 30(6), 3946-3950
ISSN: 2148-2403
https://kuey.net/

Research Article

Developing Deep Learning Models For Analyzing And Understanding Complex Graph Structures

C Ramesh Kumar[1*], Dr. Sabeena Kazi[2], Dr Archana Shukla[3], Dr. T. Muthukani Vairavel[4], Dr. Amol R Madane[5], Mr. A. Durai Ganesh[6]

[1*]Professor, Department of Applied Mathematics, Rungta College of Engineering College, Durg, Bhilai, C. G., Email: ramesh@rungta.ac.in
[2]Assistant professor, Department of Basic Sciences and Humanities, College of Computer and Information Sciences, Majmaah University, Riyadh, Al Majmaah, Riyadh, Email: s.badesaheb@mu.edu.sa
[3]Assistant professor (HoD), Department Mathematics, Nehru Gram bharati (deemed to be) University, Prayagraj, Utter Pradesh, Email: archanashuklaald@gmail.com
[4]Assistant Professor, Department of Mathematics, Kalasalingam Academy of Research and Education, Krishnankoil, Srivilliputtur, Virudhunagar, Tamil Nadu, Email: muthukanivairavel@gmail.com
[5]Solution Architect, Department of Computer Consultancy, Tata Consultancy Services Ltd Pune, Pune, Maharashtra, Email: amol.madane@gmail.com
[6]Assistant Professor, Department of Mathematics, PET Engineering College, Vallioor, Tamil Nadu, Email: aduraiganesh25@gmail.com

Citation: C Ramesh Kumar et al. (2024), Developing Deep Learning Models For Analyzing And Understanding Complex Graph Structures, *Educational Administration: Theory and Practice*, *30*(6), 3946-3950
Doi: 10.53555/kuey.v30i6.6356

ARTICLE INFO

ABSTRACT

In this research, we construct deep learning (DL) networks to comprehend and analyze difficult graph structures. Due to DL's remarkable rise in popularity in recent years, difficult tasks in difficult domains like visual analysis and linguistic processing are resolved with great efficiency. In light of this achievement, the deep neural network (DNN) replicates lower-level neural processes. It is challenging to describe such visualizations and they aren't particularly helpful for comprehending the decision-making process. By incorporating Recurrent Neural Network (RNN) in co-activation graphs and analyzing their capacity to improve understanding, we develop prior investigations in this work. To better understand the connection among sets of neuron layers from hidden layers and output categories, the co-activation graph accumulates relationships among the activation levels of the neurons. This research is expanded to take into account various data sets that were gathered from public sources to validate the validity of the findings. By examining behaviors and strong nodes and utilizing graph imaging techniques, this work showed an understanding of the RNN function and the understanding of process modeling requirements.

Keywords: Graph structures, Deep Learning, Nodes, Recurrent Neural Network (RNN)

1. Introduction

Graphs are established to visualize a wide range of real-world interaction systems, including social media, citation, and bibliographic graphs. Learning efficient graph representations and to solve challenges down the line is especially crucial given their extensive use [1]. The node embedding in a Graph Neural Network (GNN) is created by combining and modifying the embeddings of its neighbors, according to a message passing architecture. GNNs are employed for a variety of analytical tasks, such as recommender systems, link prediction, and node categorization, because of their strong performance [2]. Graphs are an effective tool for representing information generated by both natural and artificial processes. A graph is relational as the linkages that define its structure indicate communication among the linked entities, and compositional since the made up of discrete information components [3]. The hidden layers are typically not addressed by these methods, and some of them are not relevant to convolutional layers. However, a number of models explain essentially the impact of feature area inputs on a model's output, while others application with the predictions of the neural network as an oracle to develop more decision tree-style models for machine learning (ML). Network managers can see possible bottlenecks, security risks, and other problems with graphs to monitor network traffic in real time [4]. In this work, we build on earlier studies by integrating RNN in co-activation graphs and assessing their ability to enhance comprehension.

2. Related work

To address different issues in traffic jobs, these architectures are made up of many deep learning (DL) approaches. The traffic network is broken down into grids using convolution neural networks (CNNs) to simulate spatial dependence. The offered mechanical classification framework and linear discriminant analysis (LDA)-based network analysis approach are an efficient means to enable machine-assisted comprehension of text-based accident reports [5]. CNN [6] model is trained to identify accident tales and automatically extract text characteristics without requiring human feature processing. They contend that these graph topologies are more appropriate than sequences for learning tasks related to compiler optimization. They demonstrate that none of the approaches can provide a performance gain overall on the job of thread coarsening factor prediction [7]. A description of DL approaches at a high level and looked further into a thorough description of current advances in DL for atomistic modeling, materials imaging, spectrum analysis, and natural language processing (NLP) [8]. They presented an Artificial Intelligence (AI) based approach that bridges the gap between the design space and material performance through the application of a conditional generative adversarial neural network (cGAN) [9] and game theory. This communication proposes an effective method for predicting radiation patterns for antenna arrays with varying physical geometries: a residually connected complex-valued GNN. The antenna array's topology and excitation are represented graph-wise by the model, which employs GNN [10] as its structural backbone. According to the experimental results, using five different types of antenna arrays, the simulation's average absolute percent error and mean root-mean-square error averages are 0.032 and 1.86%, respectively. This study presented Temporal Graph Networks (TGNs) [11], a general and effective structure for DL on dynamic graphs that are time-series representations of graphs.

3. Materials

The study outlines of training procedure, elaborates on model parameterization, and presents the suggested RNN algorithm.

3.1 Dataset

Two significant datasets that we obtained from Kaggle were employed in our study: MNIST fashion (https://www.kaggle.com/datasets/zalando-research/fashionmnist) and MNIST handwritten digits (https://www.kaggle.com/datasets/hojjatk/mnist-dataset). Ten classes, with 70,000 images, comprise both datasets. The handwritten digits dataset has classes related to numbers 0-9, whereas the fashion dataset has classes related to clothes.

3.2 Co-activation graph

This research aims to better understand the functioning of the hidden component of the model by extracting and representing knowledge from trained RNN. With the extraction and representation of information from trained RNN, this study requires providing insights into operation the hidden part of network. Since the connections between its nodes show that their activation levels are connected, we refer to this network as a co-activation graph. Since the latter are easier for humans to understand, the primary goal of a co-activation structure is to establish a relationship among neurons in the neurons in all hidden layer depths and the characteristic input space of the result classes. The three procedures listed below can be used to create a co-activation graph given a trained RNN:

- **Determine activation values**

Extracting activation values for every prediction is the initial step in feeding the RNN with data samples. Because each neuron produces a single activation value, this technique is simple for thick layers. Conversely, because convolutional layers' filters have distinct activation values for every area in the input, they will produce a range of values.

- **Establish and determine edge weights**

Determining the strength of the interactions with neurons in every layer comes after the activation levels for each layer's neurons have been gathered. We compute the statistical correlation among every pair of neurons using the activity values.

- **Construct and evaluate the co-activation graph**

The co-activation graph may be constructed and examined in this last and third stage. Weighted edges show the correlation between the activation levels of the neurons represented by the nodes, which can be neurons in any layer of the RNN.

3.3 Recurrent neural network (RNN)

The RNN has several fixed activation function units, one for each time step. Each entity have an internal condition that's referred to as the unit's hidden state. This hidden state represents the network's current level of knowledge from the past at a specific time step. Let $\{w_s\}_{s\in}N_p$ represent a series of N-dimensional data

points$w_s \epsilon R^M$. From this sequence, information is extracted by RNN in the form of a hidden state variable $y_s \in R^M$. A nonlinear map is used to learn the states $\{w_s\}_{s\in}N_0$from the sequence y_s. It receives as inputs the current data point w_s and the previous hidden state w_{s-1} and outputs the updated hidden state w_s. The parameters for this map are Equation (1),

$$y_s = \sigma(Bw_s + Aw_{s-1}) \quad (1)$$

For a computational graph, let $\sigma: R \to R$ be linear operators and point-wise nonlinearity and $B \in R^{M\times M}, A \in R^{M\times M}$. Although it's not required, for the sake of simplicity of explanation, we'll assume that y_s and w_s have the same dimensions.

The target representation Y, which is often viewed as a more suitable illustration of $\{w_s\}$for the current job, is usually provided with the sequence$\{w_s\}$. The components of Y get another sequence $\{z_s\}_{s\in N_p}, z_s \epsilon R^N$, as is the case in automated voice recognition, and could be a single value $z \in Y$ that summarizes information from the whole series, such as a sentiment characterizing a tweet. RNN use a second nonlinear map $\Phi(y_s) = \rho(Dy_s$, to apply the hidden state to estimate Y. When a sequence serves as the target representation, this map is parameterized as Equation (2),

$$\Phi(y_s) = \rho(Dy_s) \quad (2)$$

Where the point-wise nonlinearity is utilized to compute the output, $D \epsilon R^{N\times M}$is represented by$\rho: R \to R$, and the linear output map$[\rho(w)]_j = \rho([w]_j)$. We might calculate s from the situation at the end U of the series $\Phi(y_s) = \rho(Dy_s$, in situations where a single result s is connected to the sequence$\{w_s\}$.

The optimum linear mappings D, E, and F are achieved by minimizing a loss function $L(\Phi(y_s), Y) or\ L(\Phi(y_S), Y)$ over a training set, which consists of multiple sequencesy_S and their corresponding representations y. With the help of the available training examples, this learning framework determines which sequential information is relevant to keep in the hidden statey_s, allowing the hidden state to adapt to the job at hand.

The idea that several parameters in linear operators D, E, and F are independent of time index s is crucial to the performance of RNNs. The series is executed using the same linear operators. Two primary benefits of this parameter-sharing system across the time dimension are that it permits learning compared to sequences of varied length while also controlling the number of parameters. This is in line with our hidden state recurrent approximation model, in which every learned state depends only on the input received at that moment and its predecessor. The updated state is constantly the same, regardless of starting time s_0, provided that the input's current value and its prior state are equal. The ability to determine which neurons have a significant influence on the decisions made by the model aligns a representation of co-activation graph with data through trained RNN.

4. Results

This section includes a series of in-depth tests that demonstrate the suggested RNN method beats a wide range of forecasting techniques, including one that picks up on a hidden graph structure.

4.1 Visualization of co-activation graph

Figure 1 shows that the visualization strategy inserts specific groups close to one distinct while keeping other classes spread out on both graphs. This might suggest that there is a chance for co-activation graphs to include a community structure. Before carrying out this kind of graph analysis, it's crucial to determine if the information from each RNN is represented in the graphs. The blue nodes in the buried layers indicate neurons, while those in the final layer are represented by maroon nodes.

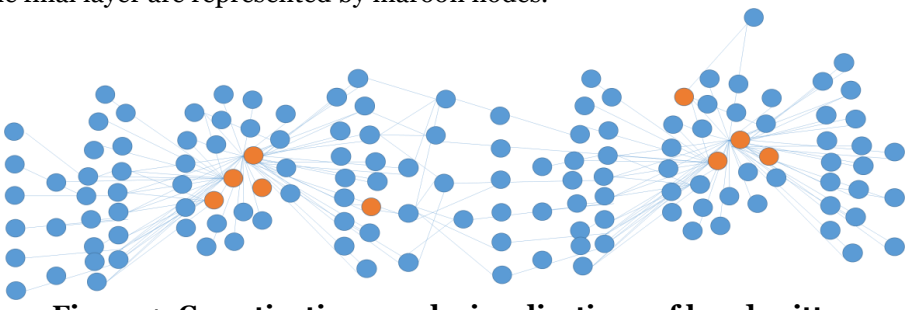

Figure 1: Co-activation graph visualizations of handwritten

4.2 Analysis of community structure

The RNN information samples from the testing set as a result could determine each neuron's activation value. The examined community structure in this graph, based on the idea shown in Figure 2, determines whether it is useful in identifying comparable classes from the perspective of the RNN.

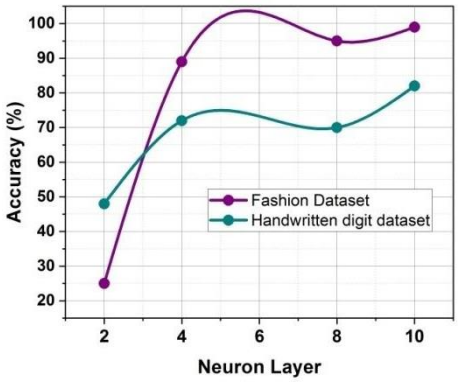

Figure 2: Extracting neuron layers in co-activation graph

The method is used to the connections among nodes in hidden layers that represent neurons and the nodes in an outcome layer that represent classes. The identified communities and classifications found in each of them are displayed in Table 1. It is shown from the fashion dataset that classes within the same groups have semantic meanings that are comparable. This is evident from the placement of classes such as coats and pullovers in one community and sandals and sneakers in another. Since the meaning of the handwritten numerals is less obvious. It is noteworthy that the modularity for both datasets was more than 0.4, indicating denser sections within the communities that were observed, but not completely distinct.

Table 1: Division and section in two datasets

	Handwritten digits	**Fashion**
Section	division	division
Class 1	0,2,4,6	pullover, T-shirt/Top, coat, shirt
Class 2	5,7,8,9	dress, Trouser
Class 3	1,3	bag, sneaker, ankle boot, Sandal
Modularity	0.45	0.413

5. Conclusion

In this research, we formalize and test a unique method for analyzing and elucidating the inner workings of deep learning models. The suggested approach makes use of the formalized concept of a co-activation graph, which first appeared to extract and express information from a developed RNN. Our evaluation showed that identify neurons for significantly contribute to the connection of certain classes by examining the graph structure. This is important to understand since we can analyze the neurons' strong influence on certain classes' prediction values to improved understand the decisions the model made. Graph visualization approaches were demonstrated to be an effective means of providing further transparency by elucidating the outcomes of individual analyses conducted on co-activation graphs. A potential approach for future research is examining the function of our effective framework for early stopping.

Reference

1. Zhao, J., Wang, X., Shi, C., Hu, B., Song, G., and Ye, Y., 2021, May. Heterogeneous graph structure learning for graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, No. 5, pp. 4697-4705).
2. Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S. and Tang, J., 2020, August. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 66-74).
3. Munikoti, S., Agarwal, D., Das, L., Halappanavar, M. and Natarajan, B., 2023. Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications. *IEEE transactions on neural networks and learning systems*.
4. Ye, J., Zhao, J., Ye, K., and Xu, C., 2020. How to build a graph-based deep learning architecture in traffic domain: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *23*(5), pp.3904-3924.
5. Zhong, B., Pan, X., Love, P.E., Ding, L. and Fang, W., 2020. Deep learning and network analysis: Classifying and visualizing accident narratives in construction. *Automation in Construction*, *113*, p.103089.
6. Brauckmann, A., Goens, A., Ertel, S. and Castrillon, J., 2020, February. Compiler-based graph representations for deep learning models of code. In *Proceedings of the 29th International Conference on Compiler Construction* (pp. 201-211).

7. Choudhary, K., DeCost, B., Chen, C., Jain, A., Tavazza, F., Cohn, R., Park, C.W., Choudhary, A., Agrawal, A., Billinge, S.J. and Holm, E., 2022. Recent advances and applications of deep learning methods in materials science. npj Computational Materials, 8(1), p.59.
8. Yang, Z., Yu, C.H. and Buehler, M.J., 2021. Deep learning model to predict complex stress and strain fields in hierarchical composites. *Science Advances*, *7*(15), p.eabd7416.
9. Wong, F., Zheng, E.J., Valeri, J.A., Donghia, N.M., Anahtar, M.N., Omori, S., Li, A., Cubillos-Ruiz, A., Krishnan, A., Jin, W. and Manson, A.L., 2024. Discovery of a structural class of antibiotics with explainable deep learning. *Nature*, *626*(7997), pp.177-185.
10. Jin, J., Su, Q., Xu, Y., He, Z. and Lu, Y., 2022. Efficient radiation pattern prediction of array antennas based on complex-valued graph neural networks. *IEEE Antennas and Wireless Propagation Letters*, *21*(12), pp.2467-2471.
11. Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F. and Bronstein, M., 2020. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*.
12. G.Venkata Subbaiah, Nalini S. Patil, C. Sugandhi, A. Durai Ganesh, M. Sindhu, 2024, For Optimizing Network Resilence Through Advanced Graph Theory Algorithms, The International Journal of Interdisciplinary Organizational Studies, 19(1), p.426-435.