Graph Convolutional Neural Networks Based Strain Estimation

GIULIA MARASCO, SOHEILA SADEGHI ESHKEVARI
and SHAMIM N. PAKZAD

ABSTRACT

Infrastructural owners have to manage and control a huge number of structures that are subject to many phenomena that can alter their performance. During their service life, infrastructure like bridges are exposed to cycles of dynamic loads, causing potential fatigue failure. For the life-cycle assessment aim, most procedures require the use of strain data. The acquisition of strain data is not easy, especially in critical positions, and the high costs would restrict the wide use of such procedures. This study proposes a technique based on the graph theory to obtain strain from acceleration data. As shown in neuroscience and biological fields, this type of architecture is able to achieve good outcomes not relying on handcrafted design parameters. Besides, their capacity of grasping spatio-temporal dependencies turns out to be a key point in the damage detection framework. A numerical case study involving a two-span beam has been investigated to explore the potentiality of this architecture for acceleration-strain conversion. Here the graph nodes represent the sensor locations and the combination of geometric and signals aided to create a spatial-temporal GCN architecture. The obtained outcomes are promising and make the framework extendable to real case studies.

INTRODUCTION

In order to control the structural performance of infrastructure like bridges that are exposed to dynamic loads and deteriorating materials over time, structural health monitoring (SHM) strategies are crucial [1, 2].

The vibration-based SHM techniques that acquire the structural response in terms of accelerations are widely used [3]. Changes in the modal parameters stemming from the raw signals are frequently connected to the damage detection task. Nevertheless, the success of such procedures is jeopardized by several factors like data volume, monitoring time, and environmental factors like temperature [4, 5] that hinder the early deterioration state detection. As a result, despite significant research efforts in this field, operational and environmental factors continue to impose risks to large-scale structural evaluations,

Giulia Marasco, Soheila Sadeghi Eshkevari, Shamim N. Pakzad, Department of Civil and Environ- mental Engineering, Lehigh University, Bethlehem, PA 18015, U.S.A. Email: gim223@lehigh.edu, sos318@lehigh.edu, snp208@lehigh.edu

and these techniques have proven to be more effective for model updating tasks.

Another research strand aims to control the structural status of bridges by evaluating the recorded structural response in terms of strain or stress. In addition to fault detection task [6], the acquisition of such structural properties makes it possible to perform fatigue evaluation and estimate the structural remaining useful life for the infrastructure. Such evaluation is of paramount relevance for structures like bridges that, throughout their service life, are repeatedly subjected to dynamic loads and therefore exposed to the fatigue phenomenon [7]. The acquisition of strain response is not straightforward due to the difficult accessibility to critical areas as well as the required high costs and laboriousness [8,9]. To overcome the limitations linked to the use of the strains, deep learning techniques able to link the strain and the acceleration response can be exploited [10]. Recently, Gulgec et al. [11] presented a study on strain signal estimation for fatigue assessment based on accelerometer-measured data leveraging LSTM. Although the latter is a useful tool for tackling tasks involving sequence prediction, it is not a suitable non-linear model simulator [12] and is also regarded as a computationally expensive method.

The idea that data can be organized on a graph structure was first introduced in other fields, such as protein interactions in biology, social networks in recommender system, traffic prediction, and computer vision. The use of this concept at the structural level is very promising because, by expressing the structure as a graph, it allows us to provide the model along with its spatial properties and information regarding the connections between the elements. Such further information will aid the model's understanding of the structural response. Right now, the studies carried out in the civil engineering field mostly focus their attention on damage classification or static structural response prediction problems. Dang et al. [13] proposed a graph convolutional framework directly using vibration data for damage classification. The node features contain node coordinates and time histories of observed accelerations. In this sense, a node's features are not only a series of discrete, separate values but also a sequence with an embedded temporal link. Whalen and Mueller [14] developed a graph-based approach for trusses. The geometry, supports, and loads are encoded by the graph that predicts the displacement fields from static loads. Song et al. proposed the use of graph neural networks for elastic structural analysis [15] in a self-supervised learning scheme.

The present study aims at creating an architecture based on a graph convolution neural network (GCN) in a supervised learning approach for the purpose of conversion of acceleration signal to strain at each structural node. Therefore, the regression problem to be solved should consider the dynamic of the structures. The GCN allows managing both spatial data and multiple time-series data due to the convolution operation. To validate the approach, synthetic data have been used and the obtained results are promising.

DATA REPRESENTATION USING GRAPH STRUCTURES

Graph neural networks (GNNs) are recent deep learning techniques improving the capabilities of traditional neural networks like Convolutional (CNNs) [16] and Recurrent (RNNs) [17, 18] to handle non-Euclidean input. The main assumption is that the order of the graph nodes doesn't have an impact on the prediction of the target. As a result, the sharing of the GNN parameters across the whole graph is unaffected by the ordering of the nodes. A graph is typically represented by 1:

$$G = (V, E) \tag{1}$$

where V denotes the set of nodes (vertices) and E indicates the set of the edges. In the structural case, the nodes can symbolize the structural joints, and the edges are the structural components linking the joints [19]. We also need to define the adjacency matrix A. In this work, we define the matrix values, A_{ij}, to be 1 if nodes i and j are linked together and 0 otherwise. The sensor locations are treated as graph nodes in this work. The vibration data are the nodes' multidimensional properties and the geometrical layout of the structure inherently yields information on the set of edges E and adjacency matrix A [13]. The types of edge orientation and the level of prediction aid to define the graph structure. In terms of edge orientation, graph structures can be either direct or undirected. The latter have been used in this study. Generally, there are three levels of prediction tasks on a graph: graph level, nodes level, and edge level. The aim of the present study is to convert acceleration signals to strain signals at each node. Consequently, the regression problem is addressed at the node level.

In the literature, there are several types of GNN. Graph auto-encoder , graph attention model, graph generative model, graph recurrent neural network, and graph convolutional network are some examples. The present study is focused on the last.

Graph convolutional network models are a class of neural network which can make use of the graph's structure and convolutionally aggregate node information from the neighborhoods. They have been successful in several tasks and have a high capacity to learn the graph representations [20].

The method in which the CNN and the GCN combine vibration signals from multiple sensors is the main distinction between them. Once CNN's kernel combines the signals, there isn't the possibility to determine which sensor a time series belongs to. On the other hand, after the GCN operation a new version of the node's features which only combines the signals of nodes that are directly linked is obtained. Consequently, one may distinguish the data from each sensor following a GCN layer.

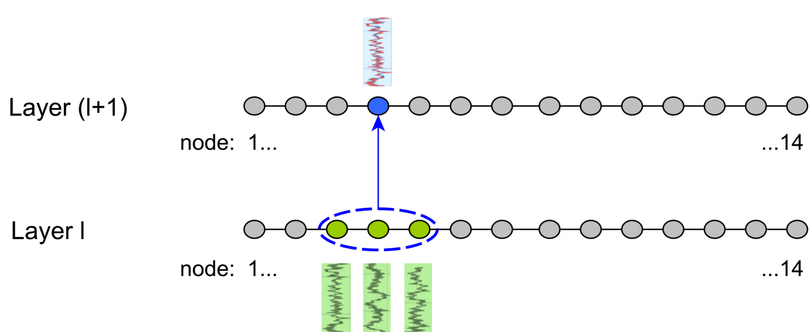

Figure 1. GCN convolution process.

Figure 1 shows the convolutional operation in GCN. The information stemming from the neighbors aids in improving the prediction accuracy at the nodes due to the spatial information given by the close nodes. Furthermore, unlike CNNs which integrate data based on data organisation, GCNs take advantage of real mechanical interactions. GCNs are able to take into account that adjacent data may not directly interact mechanically. The GCN layer tipically computes the representation of the node X^{l+1} at layer (l+1) starting from X^l, the one at level l. Equation 2 shows such operation.

$$X^{l+1} = \hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}X^lW + b \tag{2}$$

Here, $\hat{A} = A + I$ is the self-loops adjacency matrix, $\hat{D}$ is its degree matrix, I is the identity matrix, and W is a weight matrix. In this study the graph filter by means of the degree matrix has not been applied to normalize the input adjacency matrix.

CASE STUDY

To implement and test the graph architecture from the perspective of fatigue assessment, a case study involving a finite element two-span beam subjected to traveling loads simulating the vehicles loads on a bridge has been used.

Each span length is 7.3 meters and each is divided to 8 equal length elements. The Finite Element Software SAP2000, which is well known for its reliability and effectiveness, is used to simulate the beam. Figure 2 shows the geometry and the sensor's location (blue rectangles). To match a bridge dataset as nearly as feasible, the natural frequencies, damping, and moving load repetition have all been tuned for the vibration dataset. The frequencies of the first three modes are 1.9, 3, and 7.5 Hz. Seven nodes for each span, as highlighted in Figure 2, are used to capture the structural response in terms of accelerations and strains. The accelerations and strains time series are recorded for 4 minutes of vehicle moving load at a sampling frequency of 0.01 seconds.

After then, noise with an SNR of 15 has been added. By subtracting the average value and dividing by the standard deviation for each node separately, the signals are normalized. The signals are subsequently split into 4s, or 400 samples each, with 20-sample steps.

As a result, for each of the 14 nodes of the graph/sensor locations, a dataset composed of 1181 signals with 400 data points each has been created.

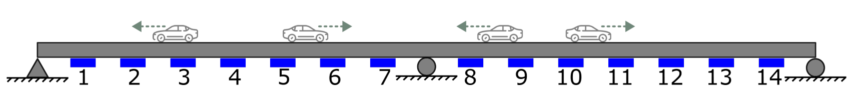

Figure 2. Two-span simulated beam.

GRAPH NETWORK CONFIGURATION AND TRAINING

The database was handled as a collection of 1181 graphs to make it compatible with the use of GCN. Each graph represents one time interval in which signals are recorded for all nodes. Each graph has a feature matrix X $\in \Re^{(n \times d)}$ and an adjacency matrix A $\in \Re^{(n \times n)}$, where n is the number of nodes and d is the dimension of node features. In this study, n is equal to 14 and d is equal to 403. The feature vector for each node is composed of the acceleration signal (400 samples), the node coordinates (x,y), and one feature that reflects the first mode shape.

Table I shows the detailed configuration of the models and the number of parameters for each layer. Leaky-ReLU and Linear are the activation functions used for the first and second Dense layers, respectively.

The dataset has been split considering 70%, 20%, and 10% of the entire dataset for the training, validation, and testing, respectively. The Adam optimizer and the hyper-parameters' values displayed in Table II have been used for the training network. As a loss function, the Mean Squared Error has been used to be minimized throughout the optimization process of training.

The network has been implemented in Python with the Spektral library [21].This latter is based on TensorFlow 2 and the Keras API, and it offers a versatile framework for GNNs that enables the implementation of sophisticated applications with a manageable level of complexity.

TABLE I. Model summary.

Layer Type	Parameters
GCN	80,800
Normalization	400
Dense	80,800
Dropout	0
Dense	160,400

TABLE II. Hyperparameters configuration.

Hyperparameters	Values
Learning rate	0.0001
Epochs	3,000
Patience	500
Batch size	32

RESULTS

Figure 3 displays the values of the loss function over the epochs for the training and validation dataset. The trend shown by the loss function points out a stable convergence during the training of the network. Figure 4 shows the prediction of one normalized signal. As can be seen, the target trend, namely the normalized real signal, is captured by the model.

Time–response assurance criterion (TRAC) is the metric used to evaluate the capacity of the model to predict correctly the strain signals. Within its formulation (see Equation 3) X_t and $\bar{X}_t$ are the measured strain signal and the predicted strain signal, respectively.

$$TRAC = \frac{\left[X_t^T \bar{X}_t\right]^2}{\left[X_t^T X_t\right]\left[\bar{X}_t^{\,T} \bar{X}_t\right]} \tag{3}$$

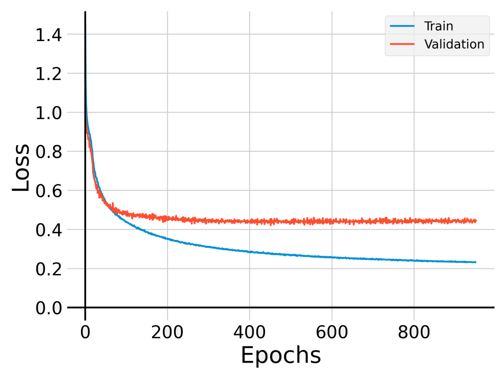

Figure 3. Loss function for training and validation dataset.

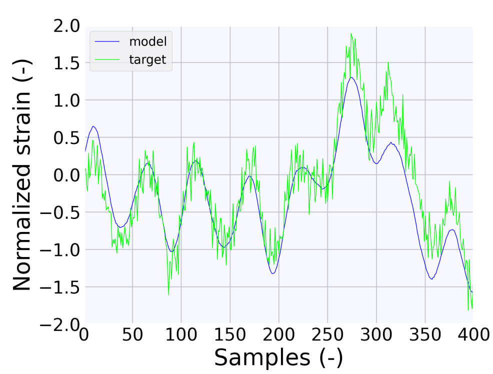

Figure 4. Normalized signal prediction.

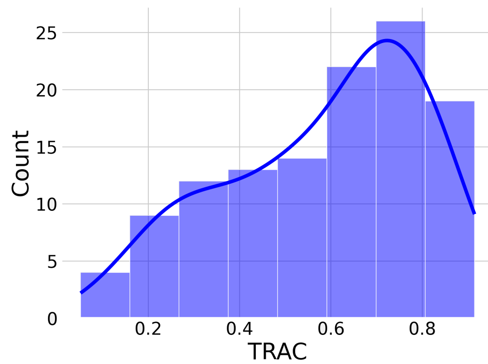

Figure 5. TRAC distribution for the test dataset.

Figure 5 shows the values of the TRAC for the test dataset. The TRAC mode value on the test dataset is 0.72. Therefore, the results are considered very promising. Further improvements could be obtained making the network deeper by stacking numerous GCN layers.

CONCLUDING REMARKS

This study investigated the use of the graph convolutional neural network to convert acceleration to strain signals for the life-cycle assessment. To validate the approach, a two-span beam has been used as case study. The framework showed a good accuracy for the prediction task even using a not deep network and the outcomes are promising for future developments of the work. Despite more research needs to be carried out before moving on the real-world applications, the presented approach is suited for generalization. It illustrates the potentialities of the graph theory in the structural engineering field and will be extended to more challenging case study.

REFERENCES

1. Gulgec, N. S., G. S. Shahidi, T. J. Matarazzo, and S. N. Pakzad. 2017. “Current challenges with bigdata analytics in structural health monitoring,” in *Structural Health Monitoring & Damage Detection, Volume 7: Proceedings of the 35th IMAC, A Conference and Exposition on Structural Dynamics 2017*, pp. 79–84.
2. Yao, R., S. N. Pakzad, and P. Venkitasubramaniam. 2017. “Compressive sensing based structural damage detection and localization using theoretical and metaheuristic statistics,” *Structural Control and Health Monitoring*, 24(4):e1881.
3. Cronin, L. M., S. S. Eshkevari, D. Sen, and S. N. Pakzad. 2021. “Transfer learning for input estimation of vehicle systems,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7953–7957.
4. Wu, C., P. Wu, J. Wang, R. Jiang, M. Chen, and X. Wang. 2021. “Critical review of data-driven decision-making in bridge operation and maintenance,” *Structure and Infrastructure Engineering*, 18(1):47–70.

5. Chiaia, B., G. Marasco, G. Ventura, and C. Zannini Quirini. 2020. “Customised active monitoring system for structural control and maintenance optimisation,” *Journal of Civil Structural Health Monitoring*, 10(2):267–282.
6. Marasco, G., F. Oldani, B. Chiaia, G. Ventura, F. Dominici, C. Rossi, F. Iacobini, and A. Vecchi. 2022. “Machine learning approach to the safety assessment of a prestressed concrete railway bridge,” *Structure and Infrastructure Engineering*:1–15.
7. Fatemi, A. and L. Yang. 1998. “Cumulative fatigue damage and life prediction theories: a survey of the state of the art for homogeneous materials,” *International journal of fatigue*, 20(1):9–34.
8. Kim, S., S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. 2007. “Health monitoring of civil infrastructures using wireless sensor networks,” in *Proceedings of the 6th international conference on Information processing in sensor networks*, pp. 254–263.
9. Childers, B. A., M. E. Froggatt, S. G. Allison, T. C. Moore Sr, D. A. Hare, C. F. Batten, and D. C. Jegley. 2001. “Use of 3000 Bragg grating strain sensors distributed on four 8-m optical fibers during static load tests of a composite structure,” in *Smart structures and materials 2001: Industrial and commercial applications of smart structures technologies*, SPIE, vol. 4332, pp. 133–142.
10. Gulgec, N. S., M. Takáč, and S. N. Pakzad. 2019. “Innovative sensing by using deep learning framework,” in *Dynamics of Civil Structures, Volume 2: Proceedings of the 36th IMAC, A Conference and Exposition on Structural Dynamics 2018*, pp. 293–300.
11. Gulgec, N. S., M. Takáč, and S. N. Pakzad. 2020. “Structural sensing with deep learning: Strain estimation from acceleration data for fatigue assessment,” *Computer-Aided Civil and Infrastructure Engineering*, 35(12):1349–1364.
12. Wang, Y. 2017. “A new concept using lstm neural networks for dynamic system identification,” in *2017 American control conference (ACC)*, pp. 5324–5329.
13. Dang, V.-H., T.-C. Vu, B.-D. Nguyen, Q.-H. Nguyen, and T.-D. Nguyen. 2022. “Structural damage detection framework based on graph convolutional network directly using vibration data,” in *Structures*, vol. 38, pp. 40–51.
14. Whalen, E. and C. Mueller. 2022. “Toward reusable surrogate models: Graph-based transfer learning on trusses,” *Journal of Mechanical Design*, 144(2).
15. Song, L.-H., C. Wang, J.-S. Fan, and H.-M. Lu. 2022. “Elastic structural analysis based on graph neural network without labeled data,” *Computer-Aided Civil and Infrastructure Engineering*.
16. Gulgec, N. S., M. Takáč, and S. N. Pakzad. 2019. “Convolutional neural network approach for robust structural damage detection and localization,” *Journal of computing in civil engineering*, 33(3):04019005.
17. Eshkevari, S. S., M. Takáč, S. N. Pakzad, and M. Jahani. 2021. “DynNet: Physics-based neural architecture design for nonlinear structural response modeling and prediction,” *Engineering Structures*, 229:111582.
18. Eshkevari, S. S., L. Cronin, S. S. Eshkevari, and S. N. Pakzad. 2022. “Input estimation of nonlinear systems using probabilistic neural network,” *Mechanical Systems and Signal Processing*, 166:108368.
19. Scarselli, F., M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. 2008. “The graph neural network model,” *IEEE transactions on neural networks*, 20(1):61–80.
20. Zhang, S., H. Tong, J. Xu, and R. Maciejewski. 2019. “Graph convolutional networks: a comprehensive review,” *Computational Social Networks*, 6(1):1–23.
21. Grattarola, D. and C. Alippi. 2021. “Graph neural networks in tensorflow and keras with spektral [application notes],” *IEEE Computational Intelligence Magazine*, 16(1):99–106.