Graph Convolutional Neural Networks

Introduction

A growing number of DoD data problems are graph problems: the data from sources such as sensor feeds and web traffic require graphs to represent mathematically. Machine learning seems like a perfect tool for such datasets, but machine learning approaches for the irregularly structured data of graph problems are sharply limited.

We use graph signal processing formalisms to create new tools for graph convolutional neural networks (GCNNs), extending deep learning into the irregular world of graph problems.

Learning on Graphs

We address two classes of graph learning problems.

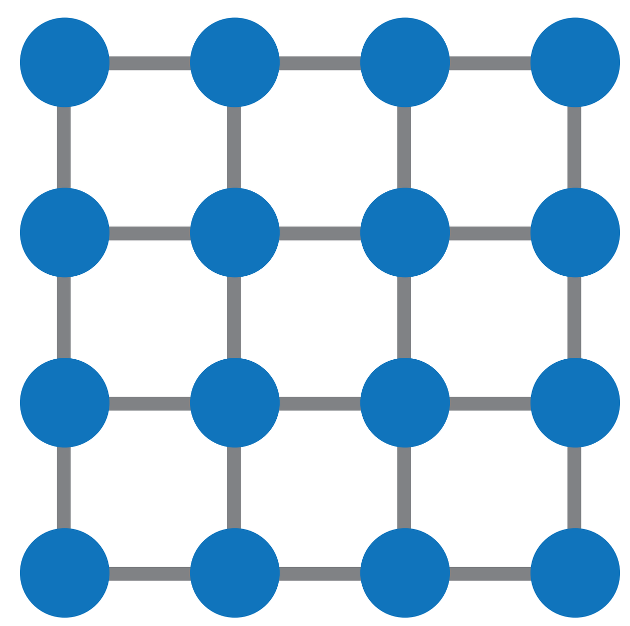
- **Node classification:** Predict information about unlabeled nodes in a graph, based on labeled nodes.
- **Graph classification:** Predict information about new graphs, based on labeled graphs. This is like image classification in computer vision.

Our Approach

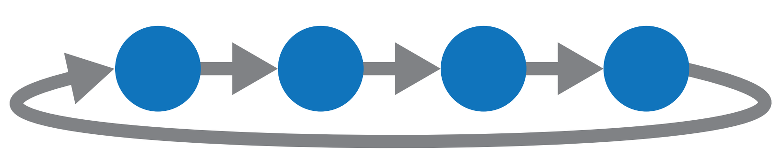
We built GCNNs using graph signal processing theory, yielding implementations that are computationally cheaper and empirically more accurate than other approaches. We also conducted experiments with pooling and sampling to further improve the state of the art and better understand when graph convolution is useful. These results will be detailed in forthcoming publications at the end of 2019.

We apply **graph signal processing** formalisms to create new tools for **deep learning on graphs.**

Regular Data Structures

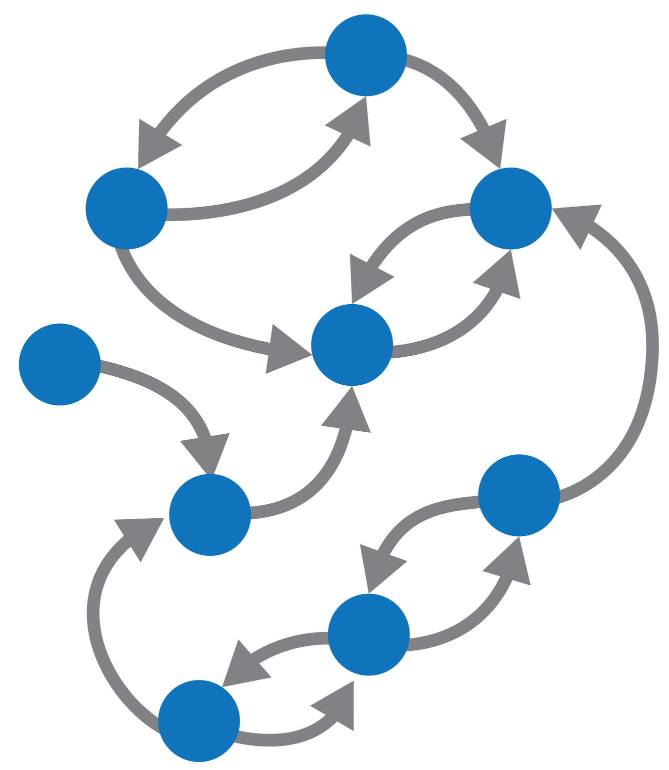


Images



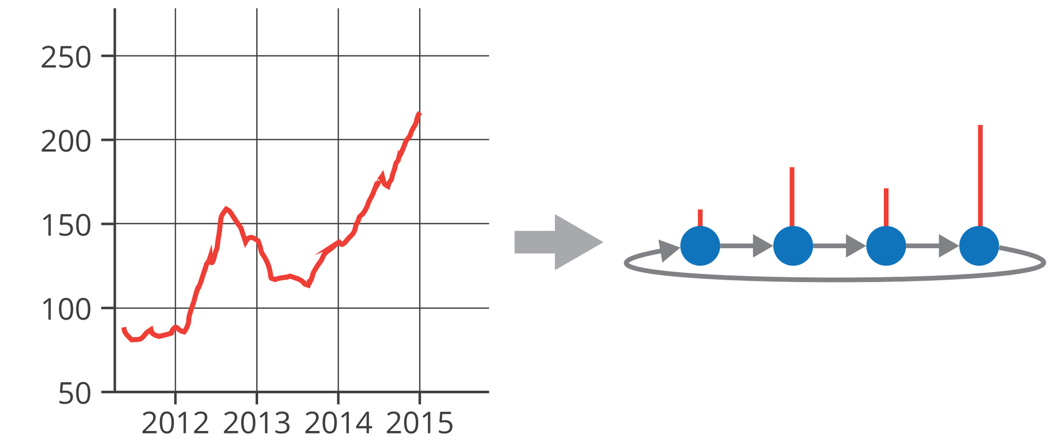
Time Series

Irregular Data Structures

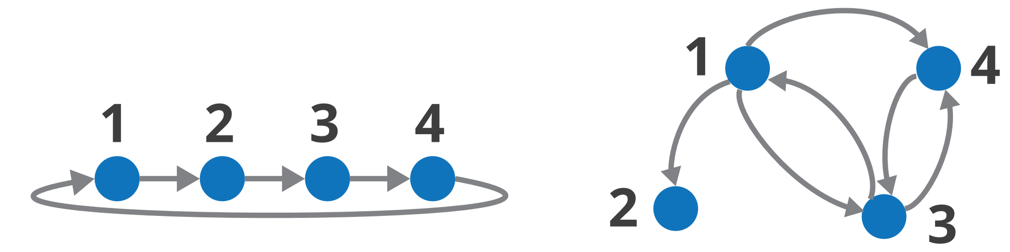


Social Networks
World Wide Web
Telecom Networks
Supply Chains
Biological Systems
Chemical Models
State Machines
Call Graphs
•••

Most deep learning techniques only work well on Euclidean data structures (i.e., data with a uniform, grid-like structure, as shown on the left) and don't extend to data with non-Euclidean structures (as shown on the right). GCNNs are an effort to extend the techniques that perform so well on regular data structures to irregular data structures.



Regular data structures like time series and images can be modeled as graphs, and the signal processing operations that work on these regular data structures can be considered special cases of a more generalized graph signal processing.



Time series General graph

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

A graph interpretation of a time series shows that the circular shift matrix $\mathbf{C}$ is also the graph's adjacency matrix. This dual role of the adjacency matrix lets us develop a graph signal processing that can be applied to other graphs, such as the one on the right with its graph shift $\mathbf{A}$.

$$\mathbf{G} = \sum_{k=0}^{K} g_k \mathbf{A}^k \qquad (1)$$

$$\mathbf{x}^{(\ell+1)} = \sigma\left(\mathbf{G}\mathbf{x}^{(\ell)} + \mathbf{b}\right) \qquad (2)$$

It can be shown in signal processing theory that convolution is polynomial in the shift, therefore, we can express the graph convolution $\mathbf{G}$ as a polynomial in the graph shift $\mathbf{A}$, as shown in equation (1). This graph convolution can then be used as the kernel in a convolutional layer of a neural network, as shown in equation (2).

Carnegie Mellon University
Software Engineering Institute

Oren Wright | owright@sei.cmu.edu
Dr. José Moura | moura@andrew.cmu.edu