Graph Neural Networks

Alejandro Ribeiro

Dept. of Electrical and Systems Engineering

University of Pennsylvania

Email: aribeiro@seas.upenn.edu

Web: alelab.seas.upenn.edu

August 31, 2020

Course Objectives

- This professor is very excited today. Happy to have a captive audience to talk about his research

- Graph Neural Networks (GNNs) are exciting tools with broad applicability and interesting properties

These are, therefore, the two objectives of this course

Develop the ability to use Graph Neural Networks in practical applications	Understand the fundamental properties of Graph Neural Networks

- Identify situations where GNNs have potential. Formulate problems with GNNs. Develop solutions.

Are These Course Goals Relevant?

- Definitely! ⇒ GNNs are the tool of choicefor performing machine learning on graphs

⇒ Authorship attribution ⇒ Identify author of anonymous text. See [Segarra et al '14]

⇒ Recommendation systems ⇒ Predict product ratings of different customers. See [Ruiz et al '18]

⇒ Resource allocation in wireless communication networks See [Eisen-Ribeiro '19]

⇒ Learning Decentralized Controllers in Collaborative Autonomy. See [Tolstaya et al '19]

- My mom would always ask me about my first day of school. All good mothers are equal.
- I would hate it if you don't have anything interesting to tell your moms when you call them later tonight
- Thus, allow me to use the next half hour to tell you some interesting things for this conversation

(I) The why $\Rightarrow$ Graphs appear in scores of settings. $\Rightarrow$ They are pervasive models of structure

(II) The how $\Rightarrow$ We should use a neural network $\Rightarrow$ Fully connected neural networks do not scale

$\Rightarrow$ Convolutions (in time or graphs) are the key to scalable machine learning

(III) Convolutional filters in Euclidean space and convolutional filters on graphs

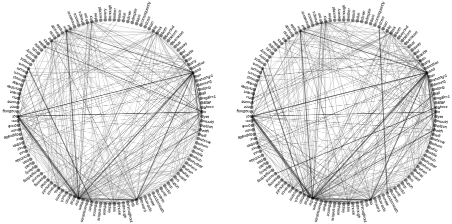
(IV) Convolutional neural networks and Graph (convolutional) neural networks

Machine Learning on Graphs: The Why

Why Are Graphs so Common in Information Processing?

- Graphs are generic models of signal structure that can help to learn in several practical problems

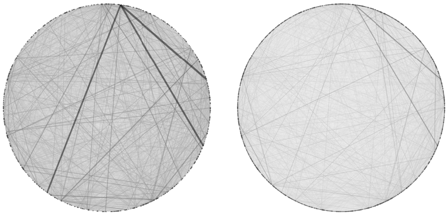
Authorship Attribution



Identify the author of a text of unknown provenance

Segarra et al '16, arxiv.org/abs/1805.00165

Recommendation Systems



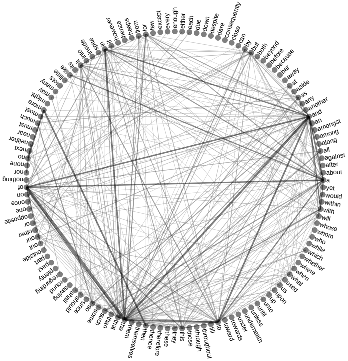
Predict the rating a customer would give to a product

Ruiz et al '18, arxiv.org/abs/1903.12575

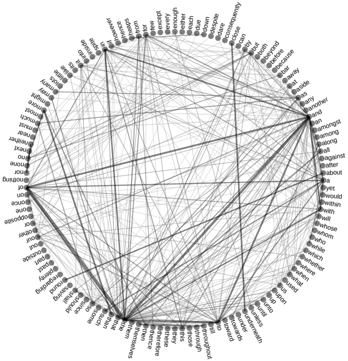
- In both cases there exists a graph that contains meaningful information about the problem to solve

- **Nodes** represent different **function words** and **edges how often words appear close** to each other
  - ⇒ A proxy for the different ways in which different authors use the English language grammar

William Shakespeare



Christopher Marlowe



- WAN differences **differentiate the writing styles of Marlowe and Shakespeare** in, e.g., Henry VI

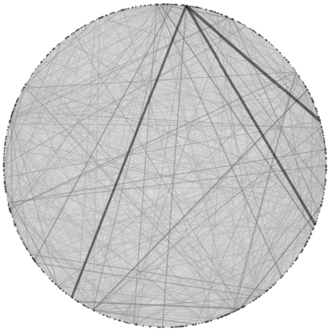
Segarra-Eisen-Egan-Ribeiro, *Attributing the Authorship of the Henry VI Plays by Word Adjacency*, Shakespeare Quarterly 2016, doi.org/10.1353/shq.2016.0024

Recommendation System with Collaborative Filtering

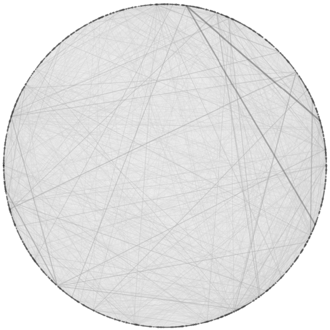
- Nodes represent different customers and edges their average similarity in product ratings

  $\Rightarrow$ The graph informs the completion of ratings when some are unknown and are to be predicted

Variation Diagram for Original (sampled) ratings



Variation Diagram for Reconstructed (predicted) ratings



- Variation energy of reconstructed signal is (much) smaller than variation energy of sampled signal

Ruiz-Gama-Marques-Ribeiro, *Invariance-Preserving Localized Activation Functions for Graph Neural Networks*, arxiv.org/abs/1903.12575

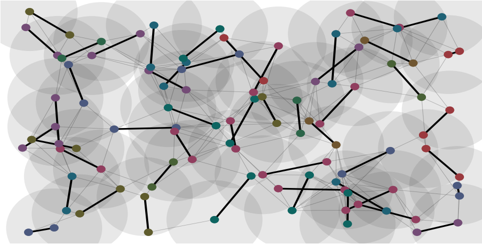
- Graphs are more than data structures $\Rightarrow$ They are models of physical systems with multiple agents

Decentralized Control of Autonomous Systems

Coordinate a team of agents without central coordination

Tolstaya et al '19, arxiv.org/abs/1903.10527

Wireless Communications Networks



Manage interference when allocating bandwidth and power

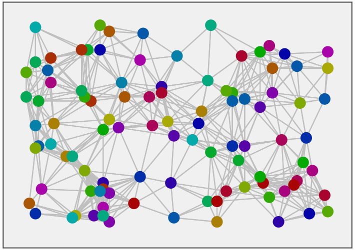
Eisen-Ribeiro '19, arxiv.org/abs/1909.01865

- The graph is the source of the problem $\Rightarrow$ Challenge is that goals are global but information is local

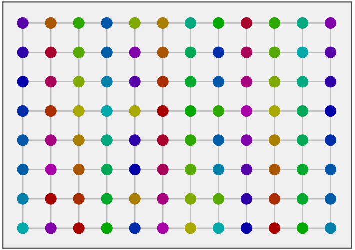
Machine Learning on Graphs: The How

- There is overwhelming empirical and theoretical justification to choose a neural network (NN)

Challenge is we want to run a NN over this



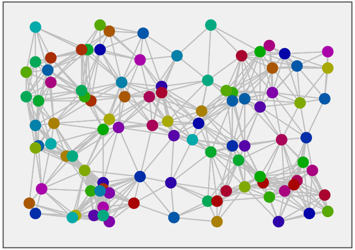
But we are good at running NNs over this



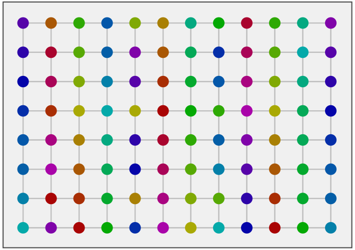
- Generic NNs do not scale to large dimensions $\Rightarrow$ Convolutional Neural Networks (CNNs) do scale

- CNNs are made up of layers composing convolutional filter banks with pointwise nonlinearities

Process graphs with graph convolutional NNs



Process images with convolutional NNs



- Generalize convolutions to graphs $\Rightarrow$ Compose graph filter banks with pointwise nonlinearities
- Stack in layers to create a graph (convolutional) Neural Network (GNN)

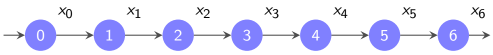
Convolutions in Time, in Space, and on Graphs

- How do we generalize convolutions in time and space to operate on graphs?

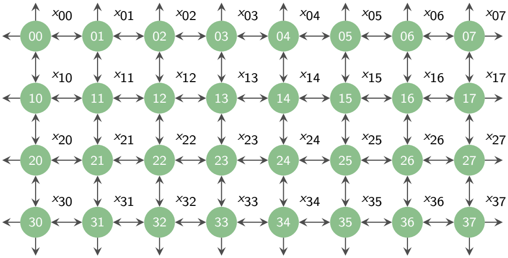
$\Rightarrow$ Even though we do not often think of them as such, convolutions are operations on graphs

- We can describe discrete time and space using graphs that support time or space signals

Description of time with a directed line graph



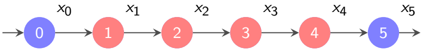
Description of images (space) with a grid graph



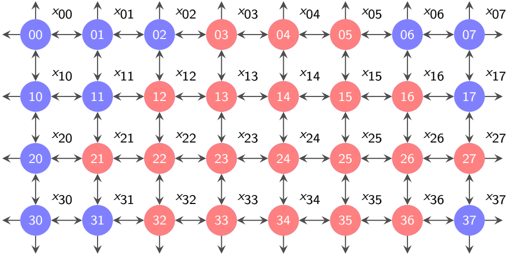
- Line graph represents adjacency of points in time. Grid graph represents adjacency of points in space

- Use line and grid graphs to write convolutions as polynomials on respective adjacency matrices $\mathbf{S}$

Description of time with a directed line graph



Description of images (space) with a grid graph

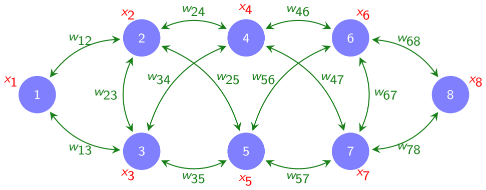


- Filter with coefficients h_k $\Rightarrow$ Output $\mathbf{z} = h_0\,\mathbf{S}^0\mathbf{x} + h_1\,\mathbf{S}^1\mathbf{x} + h_2\,\mathbf{S}^2\mathbf{x} + h_3\,\mathbf{S}^3\mathbf{x} + \ldots = \sum_{k=0}^{\infty} h_k\,\mathbf{S}^k\mathbf{x}$

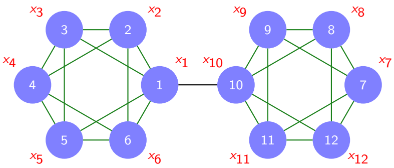
- Time and Space are pervasive and important, but still a (very) limited class of signals

- Use graphs as generic descriptors of signal structure with signal values associated to nodes and edges expressing expected similarity between signal components

A signal supported on a graph



Another signal supported on another graph

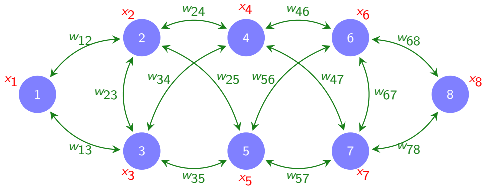


- Nodes are customers. Signal values are product ratings. Edges are cosine similarities of past scores

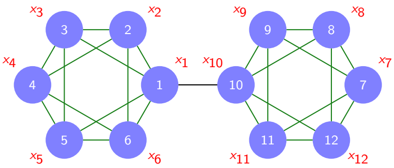
- Time and Space are pervasive and important, but still a (very) limited class of signals

- Use graphs as generic descriptors of signal structure with signal values associated to nodes and edges expressing expected similarity between signal components

A signal supported on a graph



Another signal supported on another graph

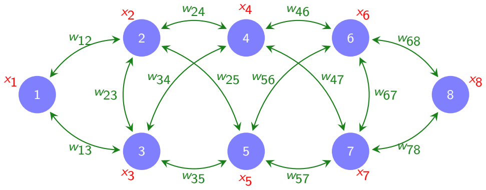


- Nodes are drones. Signal values are velocities. Edges are sensing and communication ranges

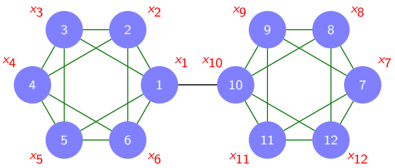
- Time and Space are pervasive and important, but still a (very) limited class of signals

- Use graphs as generic descriptors of signal structure with signal values associated to nodes and edges expressing expected similarity between signal components

A signal supported on a graph



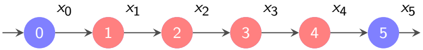
Another signal supported on another graph



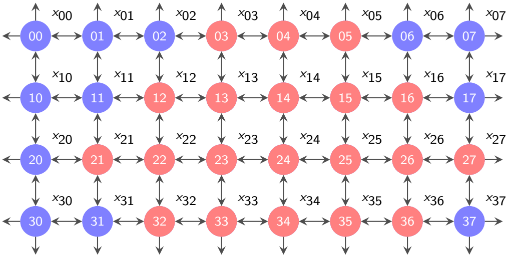
- Nodes are transceivers. Signal values are QoS requirements. Edges are wireless channels strength

- We've already seen that convolutions in time and space are polynomials on adjacency matrices

Description of time with a directed line graph



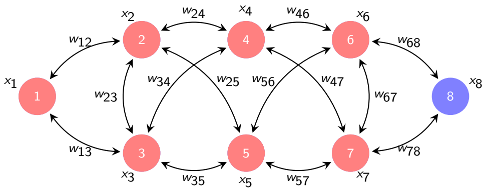
Description of images (space) with a grid graph



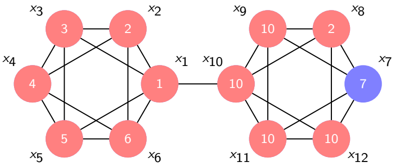
- Filter with coefficients h_k $\Rightarrow$ Output $\mathbf{z} = h_0\, \mathbf{S}^0\mathbf{x} + h_1\, \mathbf{S}^1\mathbf{x} + h_2\, \mathbf{S}^2\mathbf{x} + h_3\, \mathbf{S}^3\mathbf{x} + \ldots = \sum_{k=0}^{\infty} h_k\, \mathbf{S}^k\mathbf{x}$

- For graph signals we define graph convolutions as polynomials on matrix representations of graphs

A signal supported on a graph



Another signal supported on another graph



- Filter with coefficients h_k $\Rightarrow$ Output $\mathbf{z} = \ h_0\,\mathbf{S}^0\mathbf{x} \ + h_1\,\mathbf{S}^1\mathbf{x} \ + h_2\,\mathbf{S}^2\mathbf{x} \ + h_3\,\mathbf{S}^3\mathbf{x} \ + \ldots = \sum_{k=0}^{\infty} h_k\,\mathbf{S}^k\mathbf{x}$

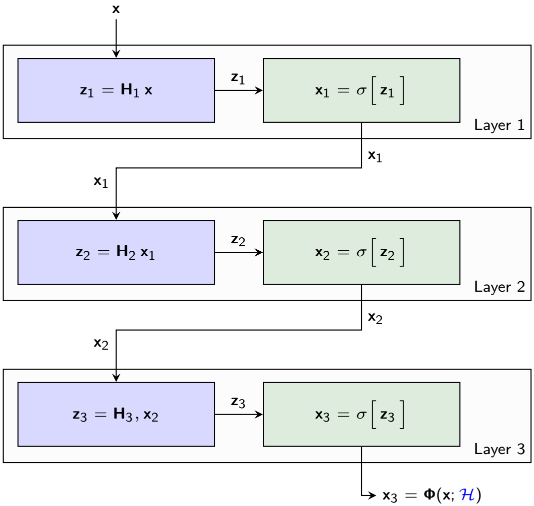
- Graph convolutions share the locality of conventional convolutions. Recovered as particular case

Convolutional Neural Networks and Graph Neural Networks

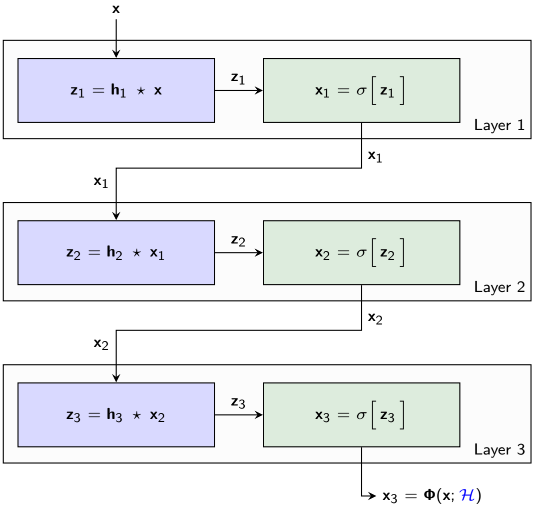
- CNNs and GNNe are minor variations of linear convolutional filters

  $\Rightarrow$ Compose filters with pointwise nonlinearities and compose these compositions into several layers

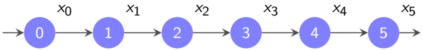
- A neural network composes a cascade of layers
- Each of which are themselves compositions of linear maps with pointwise nonlinearities
- Does not scale to large dimensional signals $\mathbf{x}$



- A convolutional NN composes a cascade of layers
- Each of which are themselves compositions of convolutions with pointwise nonlinearities
- Scales well. The Deep Learning workhorse
- A CNNs are minor variation of convolutional filters
  - $\Rightarrow$ Just add nonlinearity and compose
  - $\Rightarrow$ They scale because convolutions scale

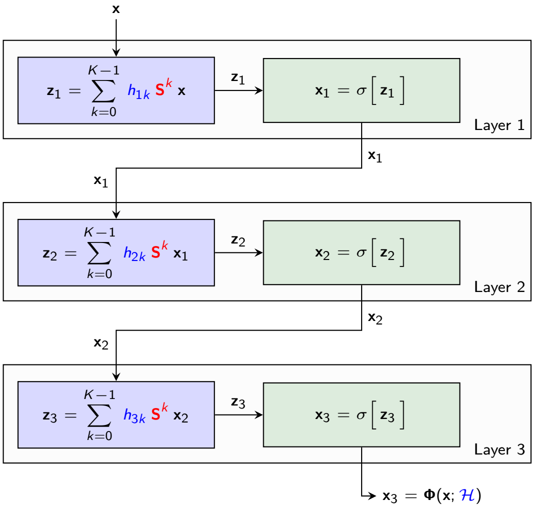


- Those convolutions are polynomials on the adjacency matrix of a line graph



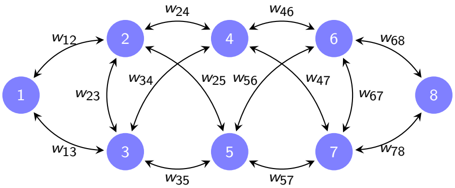
- Just another way of writing convolutions and Just another way of writing CNNs

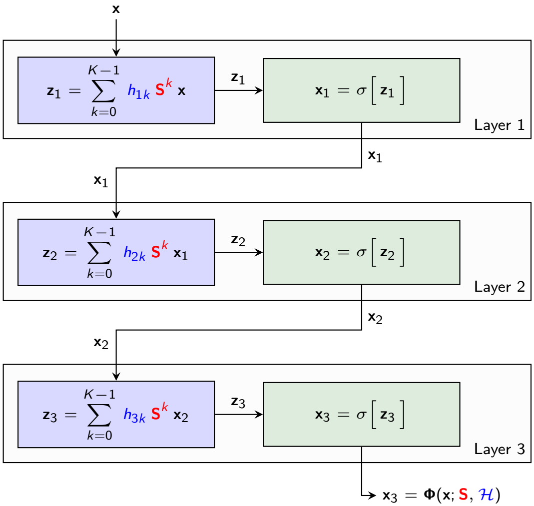
- But one that lends itself to generalization



Graph Neural Networks (GNNs)

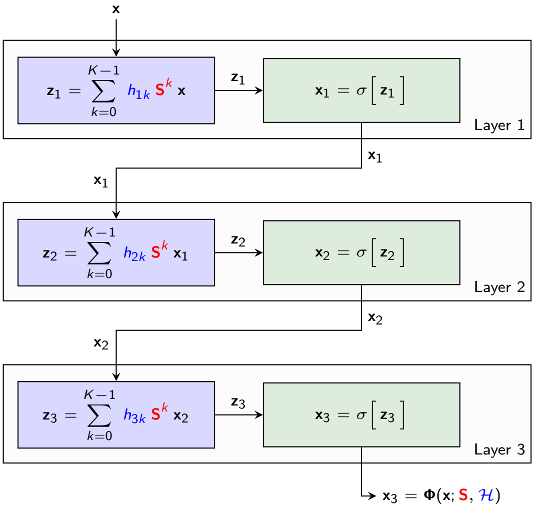
- The graph can be any arbitrary graph
- The polynomial on the matrix representation $\mathbf{S}$ becomes a graph convolutional filter





Gama-Marques-Leus-Ribeiro, *Convolutional Neural Network Architectures for Signals Supported on Graphs*, TSP 2019, arxiv.org/abs/1805.00165

- A graph NN composes a cascade of layers
- Each of which are themselves compositions of graph convolutions with pointwise nonlinearities
- A NN with linear maps restricted to convolutions
- Recovers a CNN if $\mathbf{S}$ describes a line graph



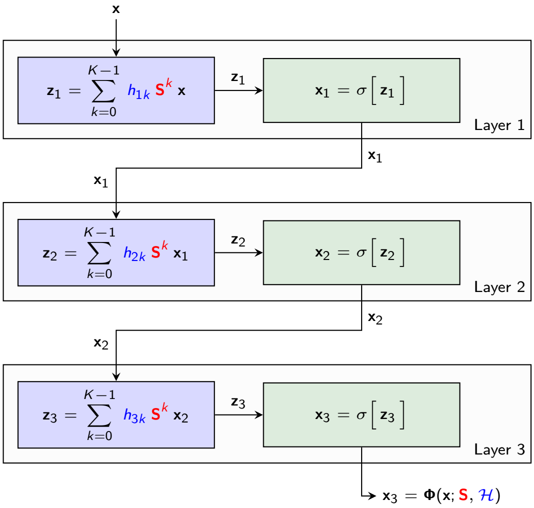
Gama-Marques-Leus-Ribeiro, *Convolutional Neural Network Architectures for Signals Supported on Graphs*, TSP 2019, arxiv.org/abs/1805.00165

- There is growing evidence of scalability.

- A GNN is a minor variation of a graph filter

  $\Rightarrow$ Just add nonlinearity and compose

- Both are scalable because they leverage the signal structure codified by the graph



Gama-Marques-Leus-Ribeiro, *Convolutional Neural Network Architectures for Signals Supported on Graphs*, TSP 2019, arxiv.org/abs/1805.00165

The Road Ahead

Course Objectives Revisited

We said there were two objectives in this course but there is obviously a third one

Develop the ability to use Graph Neural Networks in practical applications

Understand the fundamental properties of Graph Neural Networks

Define Graph Neural Network Architectures

- I told you a lot about architectures today in the form of convolutions. Just to give you a taste.
  - $\Rightarrow$ Don't worry if you didn't understand. Will revisit graph filters and graph neural networks
  - $\Rightarrow$ We will also study graph recurrent neural networks
- Can't use blindly $\Rightarrow$ GNNs have properties that explain why they work. And why they don't
  - $\Rightarrow$ Permutation Equivariance. Stability to deformations. Transferability

Ruiz-Gama-Ribeiro, *Graph Neural Networks: Architectures, Stability and Transferability,* PIEEE 2020, arxiv.org/pdf/2008.01767

- Labs will focus on building practical skills

Lab 1: Statistical and Empirical Risk Minimization. Just a warmup

Lab 2: Recommendation Systems

Huang-Marques-Ribeiro, *Rating Prediction via Graph Signal Processing*, TSP 2018, DOI: 10.1109/TSP.2018.2864654

Lab 3: Learning Controllers in Distributed Collaborative Intelligent Systems

Tolstaya-Gama-Paulos-Pappas-Kumar-Ribeiro, *Learning Decentralized Controllers for Robot Swarms with Graph Neural Networks*, arxiv.org/pdf/1903.10527

Lab 4: Learning Resource Allocations in Wireless Communication Networks

Eisen-Ribeiro, *Optimal Wireless Resource Allocation with Random Edge Graph Neural Networks*, arxiv.org/pdf/1909.01865

Lab 5: Multirobot Path Planning

Li-Gama-Ribeiro-Prorok, *Graph Neural Networks for Decentralized Multi-Robot Path Planning*, arxiv.org/pdf/1912.06095

Looking forward to Working with You!