Graph Neural Networks

Introductory lecture

Advanced ML using NN

spring 2021

Introduction to Graph Neural Networks (GNN)

Mats Granath
2/3 2021

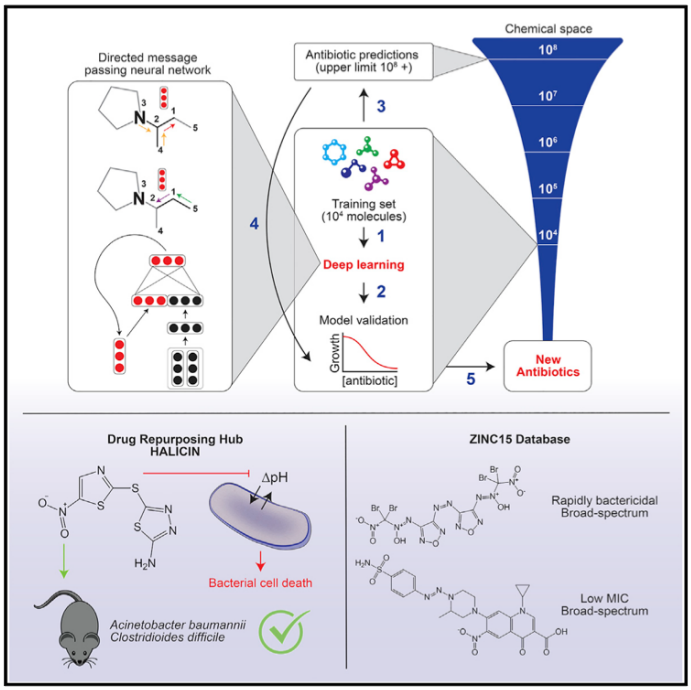
Recent GNN application

Cell

Article

A Deep Learning Approach to Antibiotic Discovery

Graphical Abstract



Authors

Jonathan M. Stokes, Kevin Yang, Kyle Swanson, ..., Tommi S. Jaakkola, Regina Barzilay, James J. Collins

Correspondence

regina@csail.mit.edu (R.B.), jimjc@mit.edu (J.J.C.)

In Brief

A trained deep neural network predicts antibiotic activity in molecules that are structurally different from known antibiotics, among which Halicin exhibits efficacy against broad-spectrum bacterial infections in mice.

Highlights

- A deep learning model is trained to predict antibiotics based on structure
- Halicin is predicted as an antibacterial molecule from the Drug Repurposing Hub
- Halicin shows broad-spectrum antibiotic activities in mice
- More antibiotics with distinct structures are predicted from the ZINC15 database

Stokes et al., 2020, Cell *180*, 688–702
February 20, 2020
https://doi.org/10.1016/j.cell.2020.01.021

Outline

- what is a graph, how describe
- Graph learning tasks
- GNN vs. CNN
- GNN layers
- Homework

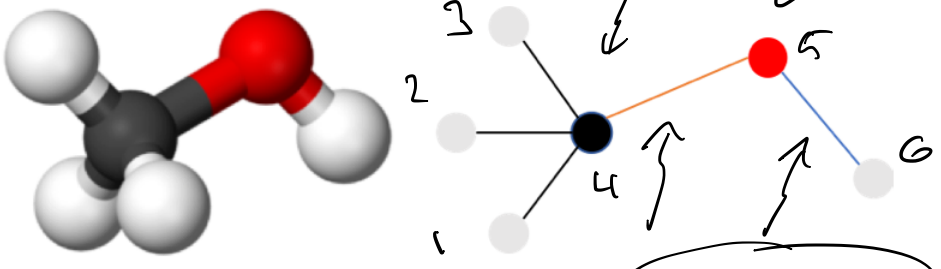
• Watch the Youtube videos for more general discussion!

What is a graph?

It's a collection of nodes and edges

$i \in \{1, \dots n\}$

$\{(i \to j)\}$



(b) molecule

$n = 6$

Structure given by Adjacency matrix ↕ $n \times n$ matrix

$A_{ij} = 1$ if edge $i \to j$

$A_{ij} = 0$ if no edge

We will consider undirected graphs $A_{ji} = A_{ij}$

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

← symmetric

But there will be more information than the structure

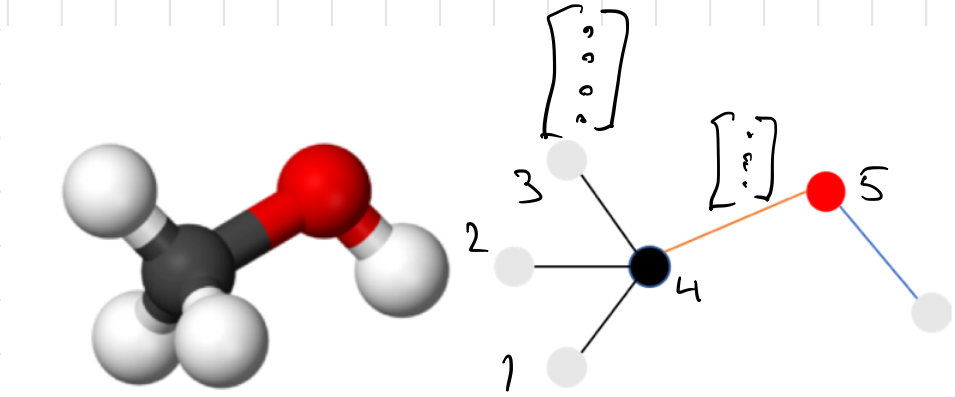
Graph Neural Networks:
A Review of Methods and Applications

Jie Zhou*, Ganqu Cui*, Zhengyan Zhang*, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, Maosong Sun

Feature vectors on nodes and edges

$\vec{e}_{ij}$ → edges

$\vec{x}_i$ → feature vector



(b) molecule

Ex. c_{45} = distance 4 to 5 (scalar, not vector)

x_3 = label of type of molecule

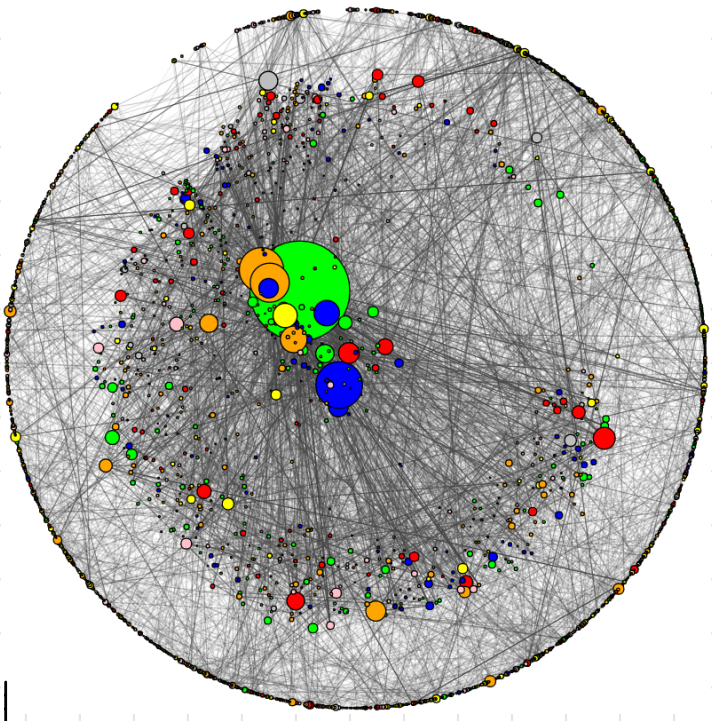
Example Cora dataset

(citation network of papers.
↑
nodes

2708 nodes

1433 dim. feature vector
↑
based on keywords

7 classes of papers

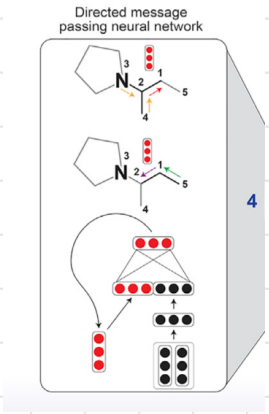


Aleksa Gordic, 2020

Machine learning tasks on graphs

Graph classification

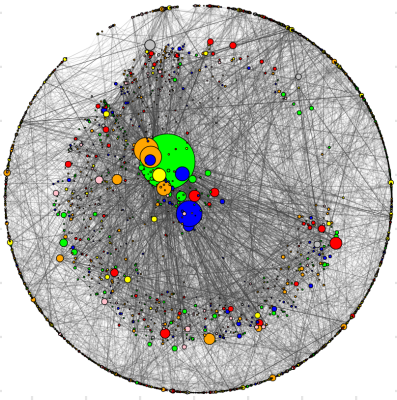
E.g. Does the molecule have atibiotic properties or not →



many graphs

Node classification

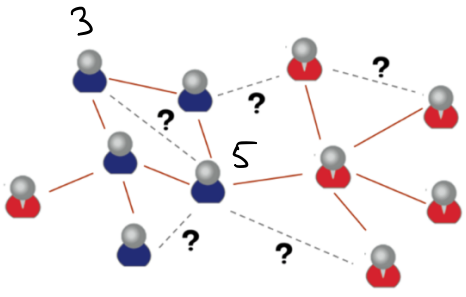
E.g. is the paper in class A, B, C, etc.? -



single graph

Link prediction

E.g. Is there a link (edge) between node 3 and 5?

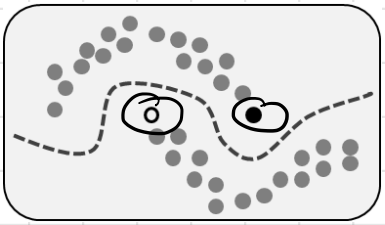


(e) social network

Supervised learning:

Graph classification is typically supervised:
set of labeled graphs, use for training.

Semi-supervised learning:



Ex: two labeled data points, many unlabeled
use clustering to extract info. from unlabeled data points.

Node prediction is typically of this type.

Ex. Cora 140 labeled nodes out of 2708 nodes
Use info about how papers are related (citation graph)

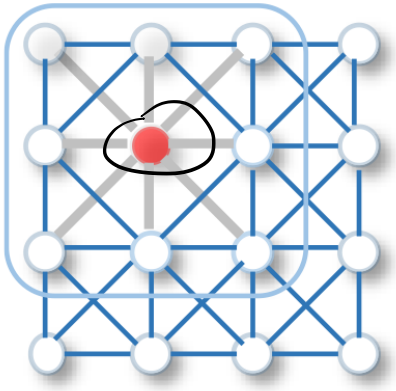
E.g. homophilous graph: adjacent nodes more likely to share properties.

Unsupervised:

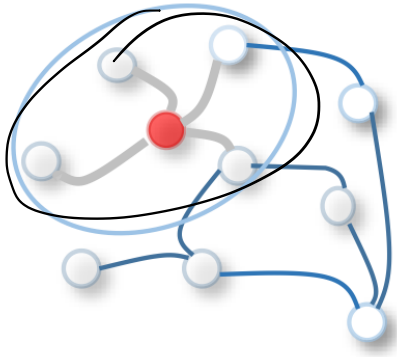
GNN do clustering even when not trained.

Convolutions on graphs

CNN filter



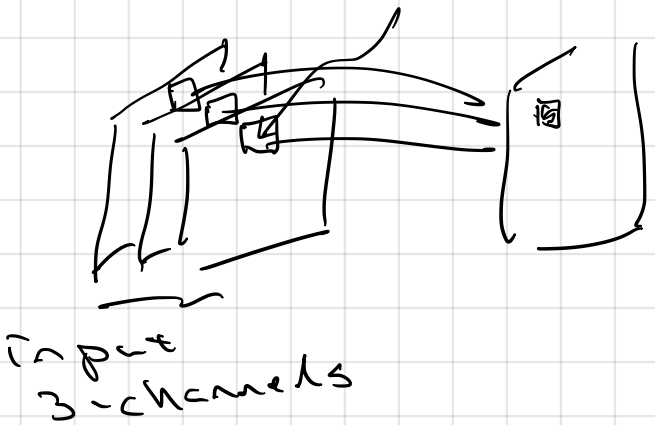
GNN filter



A Comprehensive Survey on Graph Neural Networks

Zonghan Wu, Shirui Pan, *Member, IEEE*, Fengwen Chen, Guodong Long, Chengqi Zhang, *Senior Member, IEEE*, Philip S. Yu, *Fellow, IEEE*

Standard CNN work on grids, regular structure: each "node" has fixed number of neighbors



different filter: weights to all pixels

3×3 filter has 9 trainable weights

How to set up similar object on graph?

Clearly: We want to use adjacency matrix A_{ij}

Problem: not fixed number of adjacent nodes, how many weights?

Graph convolutional layers, some examples

Basic graph convolution

SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

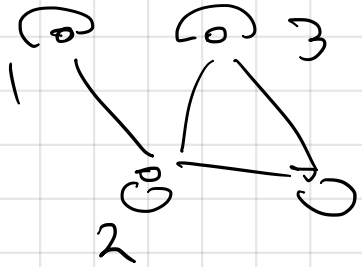
Thomas N. Kipf
University of Amsterdam
T.N.Kipf@uva.nl

Max Welling
University of Amsterdam
Canadian Institute for Advanced Research (CIFAR)
M.Welling@uva.nl

add "self-loops" to A

$\tilde{A} = A + I$

$\tilde{A} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ \vdots & & & \end{pmatrix}$



its degree matrix

$\tilde{D}$: diagonal $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij} = \tilde{d}_i$

$D_{ij} = 0 \quad i \neq j$

- Convolution takes a graph to an isomorphic graph, same structure, i.e. same A_{ij}.

Gives new node feature vectors

$\vec{x}_i \rightarrow \vec{x}'_i$

$\vec{x}_i$: dimension d; $\vec{x}'_i$: dimension d'

make feature matrix $X = \begin{pmatrix} - x_1 - \\ - x_2 - \\ - x_3 - \end{pmatrix}$ $n \times d$

$$X' = \sigma\left(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} X W\right)$$

X': $n \times d'$; σ: non-linearity, ReLU, tanh.; $\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$: $n \times n$; X: $n \times d$; W: weight matrix, $d \times d'$

- Same weights on all nodes!

equivalently

$$\vec{x}_i' = \sigma\left(\sum_j \frac{\tilde{A}_{ij}}{\sqrt{\tilde{d}_i \tilde{d}_j}} W^T \vec{x}_j \right)$$

collect from adjacent neighbors

linear transform to d' dim vector

just like a single dense NN layer

(collects information isotropically from neighbors.

"one-hop" per layer

Warning! Too many layers may smear out information over the graph.

No edge info used except for A

- Simple variation that uses an edge weight e_{ij} (scalar edge feature)

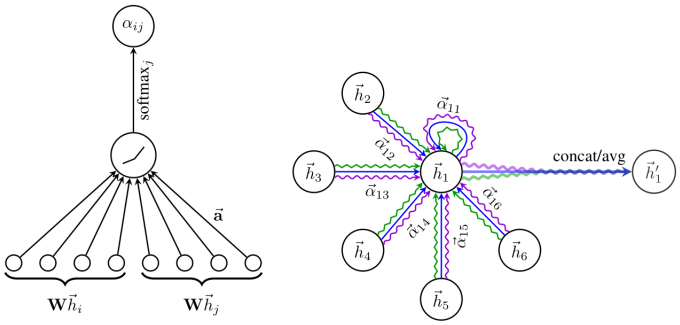
$$\vec{x}_i' = \sigma\left(w_1 \vec{x}_i + w_2 \sum_j e_{ij} \vec{x}_i \right)$$

"heavy" edges weighed more

Graph Attention layers

The basic layer is isotropic, all adjacent nodes are treated the same.

We may want to be more selective.



GRAPH ATTENTION NETWORKS

Petar Veličković*
Department of Computer Science and Technology
University of Cambridge
petar.velickovic@cst.cam.ac.uk

Guillem Cucurull*
Centre de Visió per Computador, UAB
gcucurull@gmail.com

Arantxa Casanova*
Centre de Visió per Computador, UAB
ar.casanova.8@gmail.com

Adriana Romero
Montréal Institute for Learning Algorithms
adriana.romero.soriano@umontreal.ca

Pietro Liò
Department of Computer Science and Technology
University of Cambridge
pietro.lio@cst.cam.ac.uk

Yoshua Bengio
Montréal Institute for Learning Algorithms
yoshua.umontreal@gmail.com

Attention gives selective feedback from adjacent nodes.

- $\vec{x}'_i = \sigma\left(\sum_j \alpha_{ij} W \vec{x}_j\right)$

  d' ← $\vec{x}'_i$; d ← $\vec{x}_j$; W: $d' \times d$ weight matrix; α_{ij}: attention coefficients

- $\alpha_{ij} = \text{softmax}(\Sigma_{ij}) = \dfrac{e^{\Sigma_{ij}}}{\sum_k e^{\Sigma_{ik}}}$

- $\Sigma_{ij} = \sigma'\left(\vec{a} \cdot [W\vec{x}_i \| W\vec{x}_j]\right) \tilde{A}_{ij}$

  $\vec{a}$: attention vector, dim. $2d'$, trainable weights

  $[W\vec{x}_i \| W\vec{x}_j]$: concatenation; d' + d' = $2d'$

- One can also have multiple attention heads

  W^k a^k $k = 1, \ldots K$ ← $\vec{x}'_i = \Big\|_k^K \vec{x}'^k_i$ concatenation over heads

Pooling layers

Convolutional layers: graph → same graph (new features)

- How do we get output for graph classification? ← single label
- We may want to decrease or unify graph size.

Use pooling layers: Pool nodes together.

- Global pooling ⇔ graph → 1 node (single feature vector)

E.g. global mean pool

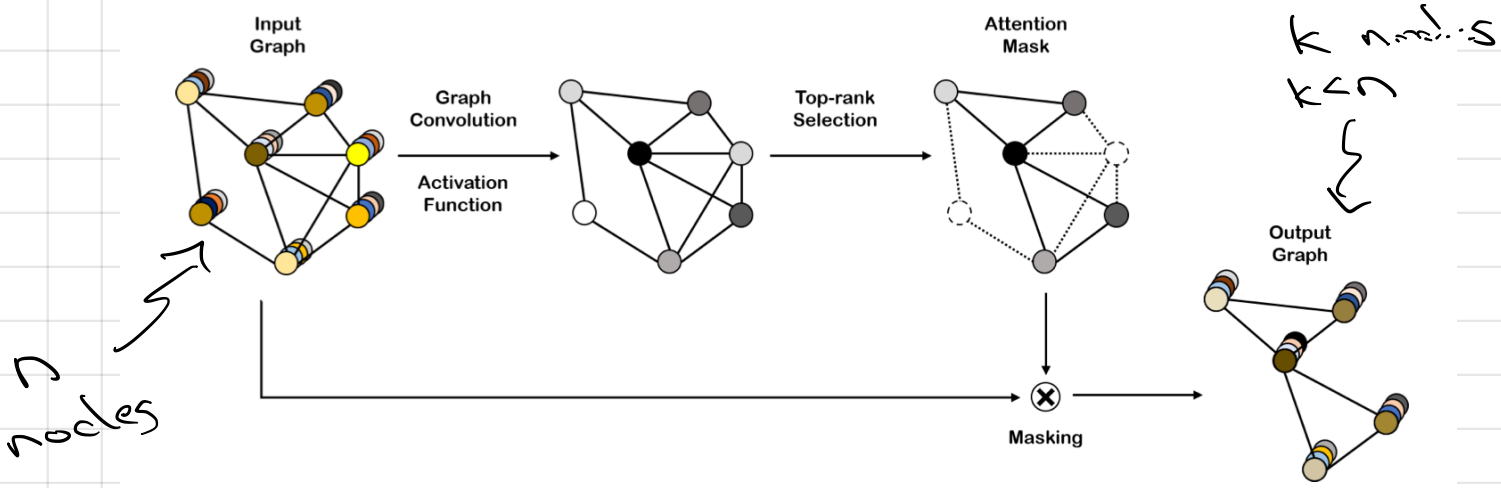
$$\vec{x}^{1} = \frac{1}{n} \sum_{i} \vec{x}_{i}$$

MLP → classifier

Example –

Self-Attention Graph Pooling

Junhyun Lee [*1] **Inyeop Lee** [*1] **Jaewoo Kang** [1]



Select the k most important nodes of

Self-attention $Z_i = \sigma\left(\sum_j \tilde{A}_{ij} \frac{\vec{a} \cdot \vec{x}_j}{\sqrt{\tilde{d}_i \tilde{d}_j}}\right)$

Z_i = scalar value of each node; $\vec{a}$ = attention vector

Keep k nodes with highest Z-value

$A_{ij} \rightarrow A_{mask(i,j)}$ ($n \times n$ → $k \times k$)

$\vec{X}'_i = \vec{X}_i Z_i$

Bottom line: Put together conv. layers + pooling layers depending on the task.

Beware of overdoing.

Homework A

- Walk through

Graph Neural Networks, Assignment A

Course: Advanced machine learning using neural networks, TIF360/FYM360
Contact: Mats Granath, mats.granath@physics.gu.se
David Fitzek, davidfi@chalmers.se

- Recommended GNN library for the h.w.