Stanford Graph Learning Workshop

Jure Leskovec
We are in the middle of a big revolution...
AI Renaissance


Perceptron
Rosenblatt

Backprop
Werbos

First NIPS

RNN / LSTM
Schmidhuber

CNN
LeCun

Autoencoder
LeCun, Hinton

IMAGENET

DeepMind
AlphaGo

Microsoft
Speech recognition

Visual cortex
Hubel&Wiesel

Neurocognitron
Fukushima

SVM
Vapnik

First GPU

ImageNet
breakthrough
Krizhevsky

TensorFlow
OpenAI

TESLA
Autonomous cars

AI BIRTH

DARK AGES

RENAISSANCE

Jure Leskovec (@jure), Stanford University
The Deep Learning Revolution

Breakthroughs in image recognition fueled by Convolutional Neural Networks.
The Deep Learning Revolution

Breakthroughs in natural language understanding fueled by Transformers.
Representation Learning

Classical computer vision: hand-crafted features (e.g. SIFT) 
+ simple classifier (e.g. SVM)

Modern computer vision: data-driven end-to-end systems
But…
Most Data Scientists still…

Machine Learning Lifecycle:
This feature, that feature.
Every single time!

Raw Data → Structured Data/Signals → Learning Algorithm → Model

Feature Engineering
Two Pain Points: One

Data Scientist’s paint point today (#1):

Data scientists still **hand encode features** to solve prediction problems.

Hand encoding graph features is…

… complex and involves expensive queries

… error prone

… suboptimal

… labor intensive
Data Scientist’s pain point today (#2):

Data is often incomplete.
Address Books, Follows, Interests, Protein Protein Interaction, Ancestry Predictions often entail completing the “missing information”.
Relational structure is often not leveraged due to scalability issues.
Why is that?
Modern ML Toolbox
Modern deep learning toolbox is designed for sequences & grids
Not everything can be represented as a sequence or a grid.

How can we develop neural networks that are much more broadly applicable?

New frontiers beyond classic neural networks that only learn on images and sequences.
Graphs are the new frontier of deep learning
The hottest subfield in ML

ICLR Keyword Growth 2018-2020

- Graph neural network
- Adversarial robustness
- Robustness
- Meta-learning
- Transformer
- Neural architecture search
- Self-supervised learning
- BERT
- NLP
- Continual learning

0.0000 0.0025 0.0050 0.0075 0.0100
% of keywords
Graphs around us!

- Molecules
- Knowledge
- Information
- Brain/neurons
- Genes
- Communication
- Software
- Social
Applications of DL on Graphs

- Recommender system
- Neutrino detection
- LHC
- Fake news detection
- Drug repurposing
- Chemistry
Applications of DL on Graphs

Computer graphics
Virtual/augmented reality
Robotics
Autonomous driving
Medicine
Drug design
We Deployed Our Technology
This Workshop

Representation Learning for Graph Data
This Workshop

- Live stream on Youtube: https://youtu.be/NKZdqCi5fVE
- Over 7000 registered attendees!
Exciting Program

- Tools and Frameworks
- Short talks about a wide range of application domains:
  - Graphics and Vision
  - Fraud and intrusion detection
  - Financial networks
  - Knowledge Graphs and Reasoning
  - Natural Language Processing
  - Biomedicine
- Panels with leading industry experts
Exciting Program

- Tools and Frameworks

**PyG 2.0**
Advanced Representation Learning on Graphs

tu dortmund university
Stanford University

**GraphGym**
Easy design and training of new GNNs

https://ogb.stanford.edu
/snap-stanford/ogb
Workshop program

08:00 - 09:00  Registration & Breakfast
09:00 - 09:30  Jure Leskovec, Stanford -- *Welcome and Overview of Graph Representation Learning*
09:30 - 10:15  Matthias Fey, TU Dortmund -- *PyG 2.0: Advanced Representation Learning on Graphs*
10:15 - 10:45  Break
10:45 - 12:00  Industry panel
  - Andrew Zhai, Pinterest
  - Jaewon Yang, LinkedIn
  - Benedek Rozemberczki, AstraZeneca
  - Hatem Helal, GraphCore
  - Nadia Fawaz, Pinterest (moderator)
12:00 - 13:00  Lunch
13:00 - 13:20  Jan Eric Lenssen, TU Dortmund -- *Applications to Graphics and Vision*
13:20 - 13:40  Rex Ying, Stanford -- *Applications to Fraud and Intrusion Detection*
13:40 - 14:00  Jiaxuan You, Stanford -- *Applications to Financial Networks*
14:00 - 14:20  Hongyu Ren, Stanford -- *Application to Knowledge Graphs*
14:20 - 14:40  Antoine Bosselut, Stanford -- *Applications in Natural Language Processing*
14:40 - 15:00  Maria Brbic, Stanford -- *Applications in Biomedicine*
15:00 - 15:30  Break
15:30 - 15:50  Jiaxuan You, Stanford -- *GraphGym: Easy-to-use Platform for Graph Learning*
15:50 - 16:10  Weihua Hu, Stanford -- *Open Graph Benchmark: Large-Scale Challenge*
16:15 - 17:00  Industry panel
  - Kim Branson, GlaxoSmithKline
  - Marinka Zitnik, Harvard University
  - Naren Chitter, JP Morgan Chase
  - Yu Liu, Facebook AI
  - Hema Raghavan, LinkedIn (moderator)
17:00  Concluding remarks
This Workshop

What is Representation Learning for Graphs?
Goal: Representation Learning

Map nodes to d-dimensional embeddings such that similar nodes in the network are embedded close together.

Node $u$ \rightarrow \mathbb{R}^d$ Learn a neural network

Feature representation, embedding

$\mathbb{R}^d$
Deep Learning in Graphs

Input: Network

Predictions: Node labels, New links, Generated graphs and subgraphs
Why is it Hard?

Networks are complex!

- Arbitrary size and complex topological structure (i.e., no spatial locality like grids)
- No fixed node ordering or reference point
- Often dynamic and have multimodal features

Image vs. Text

Networks

Images

Text
Problem Setup

- Assume we have a graph $G$:
  - $V$ is the vertex set
  - $A$ is the adjacency matrix (assume binary)
  - $X \in \mathbb{R}^{m \times |V|}$ is a matrix of node features
  - $\nu$: a node in $V$; $N(\nu)$: the set of neighbors of $\nu$.

- Node features:
  - Social networks: User profile, User image
  - Biological networks: Gene expression profiles, gene functional information
  - When there is no node feature in the graph dataset:
    - Indicator vectors (one-hot encoding of a node)
    - Vector of constant 1: $[1, 1, \ldots, 1]$
A Naïve Approach

- Join adjacency matrix and features
- Feed them into a deep neural net:

Issues with this idea:
- \( O(|V|) \) parameters
- Not applicable to graphs of different sizes
- Sensitive to node ordering

\[
\begin{bmatrix}
A & B & C & D & E \\
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix}
\]
Idea: Convolutional Networks

CNN on an image:

Goal is to generalize convolutions beyond simple lattices
Leverage node features/attributes (e.g., text, images)
Real-World Graphs

But our graphs look like this:

- There is no fixed notion of locality or sliding window on the graph
- Graph is permutation invariant
Single Convolutional neural network (CNN) layer with 3x3 filter:

Idea: transform information at the neighbors and combine it:

- Transform "messages" $h_i$ from neighbors: $W_i h_i$
- Add them up: $\sum_i W_i h_i$
**Key idea:** Network is a computation graph

Learn how to propagate information across the network
Graph Neural Networks

Each node defines a computation graph

- Each edge in this graph is a transformation/aggregation function

Deep Model: Many Layers

- Model can be of arbitrary depth:
  - Nodes have embeddings at each layer
  - Layer-0 embedding of node $u$ is its input feature, $x_u$
  - Layer-$k$ embedding gets information from nodes that are $K$ hops away
Neighborhood Aggregation

\[ h_v^{(l+1)} = \sigma(W_l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|} + B_l h_v^{(l)}) \]
Train on a set of nodes, i.e., a batch of compute graphs
Inductive Capability

- The same aggregation parameters are shared for all nodes:
  - The number of model parameters is sublinear in $|V|$ and we can generalize to unseen nodes!
Inductive Capability

Generate embeddings for nodes as needed

Even for nodes we never trained on!
Key Benefits

- No manual feature engineering needed
- End-to-end learning results in optimal features.
- Any graph machine learning task:
  - Node-level, link-level, entire graph-level prediction
- Scalable to billion node graphs!
Key Benefits

- GNNs adapt to the **shape** of data
  - Other Deep Learning architectures assume fixed input (matrix, sequence)
  - GNN makes not such assumptions
How to Stay in Touch

- Navigate to http://pyg.org
  - Join us on Slack
  - Documentation and demos
  - GitHub

conda install pyg -c pyg
<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:00 - 09:00</td>
<td>Registration &amp; Breakfast</td>
</tr>
<tr>
<td>09:00 - 09:30</td>
<td>Jure Leskovec, Stanford -- Welcome and Overview of Graph Representation Learning</td>
</tr>
<tr>
<td>09:30 - 10:15</td>
<td>Matthias Fey, TU Dortmund -- PyG 2.0: Advanced Representation Learning on Graphs</td>
</tr>
<tr>
<td>10:15 - 10:45</td>
<td>Break</td>
</tr>
<tr>
<td>10:45 - 12:00</td>
<td>Industry panel</td>
</tr>
<tr>
<td></td>
<td>- Andrew Zhai, Pinterest</td>
</tr>
<tr>
<td></td>
<td>- Jaewon Yang, LinkedIn</td>
</tr>
<tr>
<td></td>
<td>- Benedek Rozemberczki, AstraZeneca</td>
</tr>
<tr>
<td></td>
<td>- Hatem Helal, GraphCore</td>
</tr>
<tr>
<td></td>
<td>- Nadia Fawaz, Pinterest (moderator)</td>
</tr>
<tr>
<td>12:00 - 13:00</td>
<td>Lunch</td>
</tr>
<tr>
<td>13:00 - 13:20</td>
<td>Jan Eric Lenssen, TU Dortmund -- Applications to Graphics and Vision</td>
</tr>
<tr>
<td>13:20 - 13:40</td>
<td>Rex Ying, Stanford -- Applications to Fraud and Intrusion Detection</td>
</tr>
<tr>
<td>13:40 - 14:00</td>
<td>Jiaxuan You, Stanford -- Applications to Financial Networks</td>
</tr>
<tr>
<td>14:00 - 14:20</td>
<td>Hongyu Ren, Stanford -- Application to Knowledge Graphs</td>
</tr>
<tr>
<td>14:20 - 14:40</td>
<td>Antoine Bosselut, Stanford -- Applications in Natural Language Processing</td>
</tr>
<tr>
<td>14:40 - 15:00</td>
<td>Maria Brbic, Stanford -- Applications in Biomedicine</td>
</tr>
<tr>
<td>15:00 - 15:30</td>
<td>Break</td>
</tr>
<tr>
<td>15:30 - 15:50</td>
<td>Jiaxuan You, Stanford -- GraphGym: Easy-to-use Platform for Graph Learning</td>
</tr>
<tr>
<td>15:50 - 16:10</td>
<td>Weihua Hu, Stanford -- Open Graph Benchmark: Large-Scale Challenge</td>
</tr>
<tr>
<td>16:15 - 17:00</td>
<td>Industry panel</td>
</tr>
<tr>
<td></td>
<td>- Kim Branson, GlaxoSmithKline</td>
</tr>
<tr>
<td></td>
<td>- Marinka Zitnik, Harvard University</td>
</tr>
<tr>
<td></td>
<td>- Naren Chittar, JP Morgan Chase</td>
</tr>
<tr>
<td></td>
<td>- Yu Liu, Facebook AI</td>
</tr>
<tr>
<td></td>
<td>- Hema Raghavan, LinkedIn (moderator)</td>
</tr>
<tr>
<td>17:00</td>
<td>Concluding remarks</td>
</tr>
</tbody>
</table>
PyG 2.0
Advanced Representation Learning on Graphs

Matthias Fey
matthias@pyg.org
https://pyg.org
/pyg-team/pytorch-geometric

conda install pyg -c pyg