Deep Graph Networks

INTELLIGENT SYSTEMS FOR PATTERN RECOGNITION (ISPR)

DAVIDE BACCIU – DIPARTIMENTO DI INFORMATICA - UNIVERSITA' DI PISA

DAVIDE.BACCIU@UNIPI.IT
Lecture Outline

- Motivations
- Formalization of the learning task: graph prediction, induction, transduction and generation
- Historical perspective: contractive and contextual models
- A view on modern deep learning for graphs
- Research directions & applications
Introduction
Why Graphs?
Context is fundamental for the correct interpretation of information
Graph Structured Data

Oriented edge/arc $e_{vu}$ possibly with label $l_{vu}$

Node/vertex $u$

Undirected edge

Vectorial node label $x_v$

Cycle

Structures are useful because allow to represent relationships in the data
A Nomenclature Nightmare

- Deep learning for graphs
- Graph neural networks
- Deep Graph Networks
- Graph CNN
- Geometric deep learning
- Neural networks for graphs
- CNN for/on graphs
- Learning graph/node embedding
- Graph Convolutional Networks
Hierarchical representation learning allows to efficiently diffuse information through graph structure.

Node representation depends on its context (shorter first-longer later).
Predictive Tasks

Network data

Structure classification/regression
Transductive tasks

Given a vectorial and/or structured input

\[ y \sim P(y|x) \]

Learn to generate a structured prediction
An Hystorical (and Geographical) Perspective

Early neural network approaches to deal with cyclic graphs of varying topology date back to 2005-2009

(Sperduti & Starita, TNN 1997)

A. Micheli, TNN 2009

Scarselli et al, TNN 2009
Contractive - Graph Neural Networks (GNN)

- Extend the Recurrent/Recursive Neural Network approach to cyclic graphs
- Handle loops through fixed points
- Impose dynamic weight constraints to yield a contractive state mapping

Scarselli et al, TNN 2009

https://sailab.diism.unisi.it/gnn/
Contextual - Neural Networks for Graphs (NN4G)

- A feedforward approach to process graphs
- Handle loops through layering
- Uses context from frozen earlier layers compute the state on the node at current layer
- Layerwise training

A. Micheli, TNN 2009
Deep Graph Networks

- Encode vertices and the graph itself into a vector space by means of an adaptive (learnable) mapping
- Use the learned encodings to solve predictive and explorative tasks
A Survey of Recent Approaches

❖ Convolutional Neural Networks for Graphs
  ❖ Spectral
  ❖ Spatial

❖ Recurrent Graph Processing
  ❖ Fast graph reservoir networks

❖ Contextual Graph Processing
  ❖ Contextual Graph Convolutions
  ❖ Node embeddings
  ❖ Unsupervised and generative approaches
Convolutional Neural Networks for Graphs
How to Perform Convolutions on Graphs?

**SPATIAL DOMAIN**

What is the equivalent of sliding a kernel to aggregate local spatial information?

**SPECTRAL DOMAIN**

Exploit the Convolution Theorem and Fourier analysis to perform convolutions in the spectral domain.

\[ \mathcal{F}(f \ast g) = \mathcal{F}(f) \times \mathcal{F}(g) \]

Decompose a function \( f \) as a combination of vectors \( e_k \) from an orthonormal basis.
The Spectral Scenario

- Single weighted undirected graph
  - $w_{ij} > 0$ weight of the i-j edge
- Functions $f_i$ attaching values (i.e. labels/signals $x_i$) to nodes $i$
- Task: process the signals defined on the graph structure
Spectral Graph Convolution in 1 Slide

- Given a graph $G$, the eigendecomposition of its Laplacian provides an orthonormal basis $U$ which allows to compute the graph convolution of its node signals $f$ with a filter

$$ (f *_G g) = \mathcal{F}^{-1} \left( \mathcal{F}(f) \mathcal{F}(g) \right) = U \mathbf{W}(\lambda) U^T f $$

Convolutional filter $g$ in spectral domain

Graph equivalent of the learnable CNN filter matrix $W$

Spectral convolution matrix $W$ contains information on the graph Laplacian
A Graph View on (Image) Convolutions

Visual convolutions are graph convolutions on a regular grid

Plus some **key assumptions** which make it difficult to directly apply them to graphs

- Regular neighborhood
- Existence of a total node ordering
Node Neighborhoods

Example of 4-neighborhoods

Neighborhoods depend on node ordering: how can I get coherent node ordering across multiple graphs?
Leverage graph labelling techniques (e.g. Weisfeiler-Lehman) to determine a coherent ordering within the graph and between the graphs.

Determining a coherent ordering to match nodes to filter parameters in NP complete (graph normalization).
Contractive Graph Processing
Graph embedding by learning-free neurons

- Each vertex in an input graph is encoded by the hidden layer

\[
\mathbf{h}(\mathbf{v}) = \tanh(\mathbf{V}\mathbf{x}(\mathbf{v}) + \sum_{\mathbf{v}' \in N(\mathbf{v})} \mathbf{W}\mathbf{h}(\mathbf{v}'))
\]
Deep Reservoirs for Graphs

Trained in closed-form (e.g., pseudo-inversion, ridge regression)

\[ y(g) = W_0 \sum_{v \in V_g} h(v) \]
Contextual Graph Processing
Neighborhood Aggregation & Layering
What is inside of the Box?

○ A learning model of course (e.g. a neural network) including an aggregation function to handle size-varying neighborhoods

A simple model

$$h^1_v = \sigma(W_{lAGG} \{ h^l_{i-1} : i \in N(v) \}, \overline{W}_l h^l_{v-1})$$
The graph convolutional layer

\[ h^\ell+1_v = \Phi^\ell+1 \left( h^\ell_v, \Psi \left( \{ \psi^\ell+1_u(h^\ell_u) \mid u \in \mathcal{N}_v \} \right) \right) \]

**Variants/extensions:**
- Edge-aware convolution
- Attention over neighbors
- Laplacian-normalized

**MLP/Linear**

<table>
<thead>
<tr>
<th>Model</th>
<th>Neighborhood Aggregation ( h^{\ell+1}_v )</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNAG [88]</td>
<td>( \sigma \left( w^{\ell+1}<em>v x_v + \sum</em>{u \in \mathcal{N}<em>v} \sum</em>{i} w^{\ell+1}_{uv} h^\ell_u \right) )</td>
</tr>
<tr>
<td>GNN [104]</td>
<td>( \sum_{u \in \mathcal{N}<em>v} \text{MLP}^{\ell+1}(h^\ell_v, x_u, a</em>{uv}, h^\ell_u) )</td>
</tr>
<tr>
<td>GraphESN [41]</td>
<td>( \sigma \left( w^{\ell+1}<em>v x_v + \sum</em>{i} w^{\ell+1}_{uv} h^\ell_u \right) )</td>
</tr>
<tr>
<td>GCN [72]</td>
<td>( \sigma \left( w^{\ell+1}<em>v \sum</em>{u \in \mathcal{N}<em>v} a</em>{uv} h^\ell_u \right) )</td>
</tr>
<tr>
<td>GAT [120]</td>
<td>( \sigma \left( w^{\ell+1}<em>v + \sum</em>{u \in \mathcal{N}<em>v} w^{\ell+1}</em>{uv} h^\ell_u \right) )</td>
</tr>
<tr>
<td>ECC [111]</td>
<td>( \sigma \left( \sum_{u \in \mathcal{N}<em>v} \text{MLP}^{\ell+1}(a</em>{uv}, h^\ell_u) \right) )</td>
</tr>
<tr>
<td>R-GCN [105]</td>
<td>( \sigma \left( \sum_{u \in \mathcal{N}_v} \frac{1}{\sqrt{\mathcal{N}<em>v}} w^{\ell+1}</em>{uv} h^\ell_u \right) )</td>
</tr>
<tr>
<td>GraphSAGE [84]</td>
<td>( \sigma \left( w^{\ell+1}<em>v \left( \sum</em>{u \in \mathcal{N}_v} h^\ell_u \right) \right) )</td>
</tr>
<tr>
<td>CGMM [3]</td>
<td>( \sigma \left( \sum_{u \in \mathcal{N}<em>v} w^{\ell+1}</em>{uv} \left( \sum_{i} w^{\ell+1}_{uv} h^\ell_u \right) \right) )</td>
</tr>
<tr>
<td>GIN [131]</td>
<td>( \text{MLP}^{\ell+1}(1 + \epsilon^{\ell+1}) h^\ell_v + \sum_{u \in \mathcal{N}_v} h^\ell_u )</td>
</tr>
</tbody>
</table>
Graph Isomorphism Network (a.k.a. sum is better)  

Xu et al, ICLR 2019

- A study of GNN expressivity w.r.t. WL test of graph isomorphism
- Choice of aggregation functions influences what structures can be recognized
- Propose a simple aggregation and concatenation model

\[
\begin{align*}
    h_v^{(k)} &= \text{MLP}^{(k)} \left( (1 + \epsilon^{(k)}) \cdot h_v^{(k-1)} + \sum_{u \in N(v)} h_u^{(k-1)} \right) \\
    h_G &= \text{CONCAT} \left( \text{READOUT} \left( \{ h_v^{(k)} \mid v \in G \} \right) \mid k = 0, 1, \cdots, K \right)
\end{align*}
\]

Basically the NN4G approach
Graph Attention

Learning to weight contribution of other nodes when aggregating to form the node embedding

\[ \tilde{h}_i' = \sigma \left( \sum_{j \in \mathcal{N}_i} \alpha_{ij} W \tilde{h}_j \right) \]

[Image of a graph with nodes and edges, illustrating the concept of multihead attention and the formula for calculating the updated node embedding.]

Velickovic et al, ICLR 2018
Using Node Embedding

Aggregate all node embeddings to compute graph level predictions

\[ h_G = \sum_{i \in G} h_i^L \]

Typically embedding from top layer

Train node level predictors

Works also for inductive learning
Training the Embedding

- Backpropagate from the (graph or node level) error computed from the top layer embeddings to the early layers

Loss function on the graph-level or node-level predictions

Can also be unsupervisedly trained by using structure induced notions of node similarity (e.g. Node2Vec)
Unsupervised Structure Embeddings
Generative learning for graphs

❖ General, efficient and scalable architecture

❖ Handle arbitrary structure (directed, undirected or mixed), labelled edges and nodes

❖ Learn in both supervised and unsupervised way

Contextual Graph Markov Model (CGMM)
Contextual Graph Markov Model (CGMM)

Contextual Graph Markov Model (CGMM)
Incremental Construction

1. Map the graph to the model (base case)

2. Perform inference and freeze states

3. Add a new layer and use frozen states as observed variables in the graphical model

Go back to step 2
Computing embedding

✓ Finding the most likely **state assignment**

\[
\max_i P(y_u|Q_u = i) P(Q_u = i|q_N(u))
\]

✓ The inferred latent states are used as observable variables in subsequent layers

✓ A **fixed-size vector of states frequencies** as graph encoding

![Inferred states](as colors)

# Hidden states = 4

![Frequency vector]

Inferred states (as colors)

2 1 0 2

Frequency vector
CGMM Layer Training

A maximum likelihood approach to learning

\[
\mathcal{L} = \prod_{g \in G} \prod_{u \in g} \sum_i P^l(y_u|Q_u = i) P^l(Q_u = i|q^l_{\mathcal{N}(u)}(g))
\]

Assumption: i.i.d. graphs

- Emission distrib.
- Switching Parents distrib.
- Transition distrib.

Split by layer and by arc

\[
= \prod_{g \in G} \prod_{u \in g} \sum_i P^l(y_u|Q_u = i) \sum_{l \in L} P(L_u = \tilde{l}) \sum_{a = 1}^A P^l(S_u = a) \frac{1}{|\mathcal{N}^l(a(u))|} \sum_j P^l(a(Q_u = i|Q_{\ast}^l = j) \sum_{v \in \mathcal{N}^l(a(u))} q^l(v|j)
\]

Trained by Expectation-Maximization

Davide Bacciu - Università di Pisa
CGMM – Depth Matters...

...possibly more than width
Interpreting CGMM

Thanks to the probabilistic approach

Bacciu, Errica, Micheli, JMLR 2020
Advanced Topics
What About Pooling?

❖ Standard aggregation operates of **predefined node subsets**
❖ Ignore **community/hierarchical structure** in the graph
❖ Need graph coarsening (pooling) operators
  ❖ Differentiable
  ❖ Graph theoretical
  ❖ Graph signature

Rex Ying et al, NIPS 2018
Bacciu et al, AAAI 2023
The Complete Picture – Graph Convolutions & Graph Pooling

DAVIDE BACCIU - ISPR COURSE
Scaling up to large graphs

❖ Dealing with large-scale graphs
❖ Social networks
❖ Recommendation systems
❖ Biomedical network data

❖ How?
❖ Sampling
❖ Modularization (communities)
❖ Active learning
❖ HPC on graphs
Graph Generation

Generate a prediction that is itself a graph

Adjacency-based

Structure-based
Graph Variational Autoencoder

GraphVAE generates adjacency matrix up to k vertices

Sample molecules by latent space interpolation

Simonovsky, Komodakis, ICLR-WS 2018
Language-Based Graph Generation

Generate a graph node-by-node and edge-by-edge through a sequential approach.

Bacciu, Micheli, Podda, Neurocomputing 2020
Explaining Deep Graph Networks

Identify relevant substructures and features for the prediction of a molecule.

Explain predictions locally with counterfactuals.

R. Xing et al, NeurIPS 2019

Bacciu et al, NeurIPS WS 2020
Software

You can find most of the foundational models in this tutorial implemented here.

PyTorch geometric

DeepGraphLibrary

Our Python library for Deep Graph Networks

github.com/diningphil/PyDGN
Data (Benchmarks)

- Pytorch Geometric and DGL integration
- Standardized splits and evaluators + leader-board
- Node, link and graph property prediction tasks

- Standardise assessment of existing benchmarks rather than inventing new ones
- Chemical, social, vision, synthetic, bioinformatics (with leader-board)
- Pytorch Geometric and DGL integration
Applications
Predicting Properties of Chemical Compounds

\[ \text{Toxicity} \]
\[ \text{Quantum mechanical properties} \]
\[ \sim 10^{-2} \text{ seconds} \]

\[ \text{Solubility} \]
\[ \sim 10^{3} \text{ seconds} \]

Micheli et al, JCICS 2001
Duvenaud, Maclaurin et al, NIPS 2015
Gilmer et al, ICML 2017

Simulation methods

DAVIDE BACCIU - ISPR COURSE
Generating Molecules

Fragment-based deep molecule generation

Podda, Bacciu, Micheli, AISTATS 2020
Knowledge graphs

A natural way of representing known entities and relationships in a domain

Node/link embeddings are numerical encodings of entities and relationships
Side Effects of Drug Combinations

Analyzing a multimodal graph of interactions
• Drug-drug
• Drug-protein
• Protein-protein

Zitnik, Agrawal, Leskovec, Bioinformatics 2018
Recommendation Systems

...and other kinds of social network analyses
Relational Stock Learning
Point Clouds – Semantic Segmentation

Build point cloud graphs and train semantic class predictors based on vertex embeddings

Landrieu, Simonovsky, CVPR 2018
Analysis of ICT systems
Wrap-up
Advertisement Time

A tutorial paper reviewing the deep learning for graph area
D. Bacciu, F. Errica, A. Micheli, M. Podda, A Gentle Introduction to Deep Learning for Graphs, Neural Networks, 2020, Arxiv

Our Python library for Deep Graph Networks
github.com/diningphil/PyDGN

Upcoming events
Deep learning for graphs, special session at IEEE IJCNN 2023, July 2023
Deep learning for graphs, tutorial at ECML-PKDD 2023, September 2023
Deep learning for graphs, special session at ESANN 2023, October 2023
Our Learning4Graphs Community

IEEE NNTC Task Force on Learning for Structured Data
Chair: Shirui Pan – ViceChairs: Filippo Maria Bianchi, Lorenzo Livi
www.learning4graphs.org

Promote events, research and dissemination activities for the community working on machine learning for structured data.
Conclusions

- Deep learning for graphs is a research topic that is entering its consolidation phase
  - Many works sharing same underlying idea (adjacency, contractive, contextual)
  - Much early work left unacknowledged and reinvented
- What should we focus on?
  - Theoretical characterization and properties of operators (machine learning + graph theory)
  - Efficiency and efficacy of context creation and propagation (unsupervised, gradient issues, reinforcement learning & graphs)
  - Research directions (pooling, generative, transduction, expressivity, scalability, interpretability)
  - Applications (biomedical, software and ICT systems, large scale interaction networks)
- A candidate AI model for the integration of symbolic knowledge, numerical data and algorithmic reasoning
Bibliography (I)

Spectral Domain Convolutions

1. Joan Bruna, Wojciech Zaremba, Arthur Szlam, Yann LeCun, Spectral Networks and Locally Connected Networks on Graphs, ICLR 2014
2. Mikael Henaff, Joan Bruna, Yann LeCun, Deep Convolutional Networks on Graph-Structured Data, Arxiv 2015
4. Thomas N. Kipf, Max Welling, Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Spatial Domain Convolutions

1. David Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, Ryan P. Adams, Convolutional Networks on Graphs for Learning Molecular Fingerprints, NIPS 2015
Contextual Approaches

2. Yujia Li, Daniel Tarlow, Marc Brockschmidt, Richard Zemel, Gated Graph Sequence Neural Networks, ICLR 2016
4. Xu et al.: How Powerful are Graph Neural Networks?, ICLR 2019
5. Davide Bacciu, Federico Errica, Alessio Micheli, Contextual Graph Markov Model: A Deep and Generative Approach to Graph Processing, ICML 2018
6. Veličković et al, Deep Graph Infomax, ICLR 2019
7. Davide Bacciu, Federico Errica, Alessio Micheli, Probabilistic Learning on Graphs via Contextual Architectures, JMLR 2020
Bibliography (III)

Miscellanea

1. Davide Bacciu, Federico Errica, Alessio Micheli, Marco Podda, A Gentle Introduction to Deep Learning for Graphs, Neural Networks, 2020 (Tutorial)
3. Martin Simonovsky, Nikos Komodakis, GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders, NIPS Workshop, 2017 (Graph Generation)
5. Davide Bacciu, Alessio Micheli, Marco Podda, Edge-based sequential graph generation with recurrent neural networks, Neurocomputing 2020 (Graph Generation)
6. Marco Podda, Davide Bacciu, Alessio Micheli, A Deep Generative Model for Fragment-Based Molecule Generation, AISTATS 2020 (Molecule Generation)
Bibliography (IV)

Miscellanea

1. Ying et al.: Hierarchical Graph Representation Learning with Differentiable Pooling, NeurIPS 2018 (Pooling)
2. Zhang et al, An End-to-End Deep Learning Architecture for Graph Classification, AAAI 2018 (Pooling)
3. D. Bacciu et al, K-plex Cover Pooling for Graph Neural Networks, NeurIPS WS 2020 (Pooling)
5. Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, Yoshua Bengio, Graph Attention Networks, ICLR 2018 (Attention)
6. Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, Le Song, Adversarial Attack on Graph Structured Data, ICML 2018 (Adversarial Attacks)
7. R. Ying et al, GNNExplainer: Generating Explanations for Graph Neural Networks, NeurIPS 2019 (Interpretability)
8. F Errica, D Bacciu, A Micheli, Graph Mixture Density Networks, ICML 2021 (Multimodal)