Graph Neural Networking
19 June 2020

ITU
AI/ML in 5G
Challenge
Applying machine learning in communication networks
ai5gchallenge@itu.int
Graph Neural Networking Challenge 2020

Ref: ITU-ML5G-PS-014

José Suárez-Varela

Barcelona Neural Networking center
Universitat Politècnica de Catalunya
June 19th 2020
What are Graph Neural Networks?
What are Graph Neural Networks?

- Graph Neural Networks (GNN) is a neural network family designed to **learn from graph-structured data**
- GNN have been recently promoted and popularized by Google DeepMind *et al.*
- Extensively used in other fields where data is fundamentally represented as graphs (e.g., chemistry)

### State-of-the-art Neural Network Models

<table>
<thead>
<tr>
<th>Type of NN</th>
<th>Information Structure</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Connected NN</td>
<td>Arbitrary</td>
<td>Generic classification, non-linear regression</td>
</tr>
<tr>
<td>Convolutional NN</td>
<td>Spatial</td>
<td>Images and video</td>
</tr>
<tr>
<td>Recurrent NN</td>
<td>Sequential</td>
<td>Text and voice</td>
</tr>
<tr>
<td>Graph NN</td>
<td>Relational</td>
<td>Graphs (molecules, maps, networks)</td>
</tr>
</tbody>
</table>
Current status of Graph Neural Networks

Many AI applications rely on graphs:
- Chemistry (e.g., molecules)
- Biology
- Physics
- Logistics
- Social networks
- Computer Networks
- ...

Currently, research efforts are being devoted to develop the theoretical foundations of GNN.

The networking community is starting to investigate its applications.

Google Trends: “Graph Neural Networks”

*Must-read papers on GNN: [https://github.com/thunlp/GNNPapers](https://github.com/thunlp/GNNPapers)
How can Graph Neural Networks be applied to networking?
What is a digital twin?

- A digital twin is a mathematical representation of a physical and/or logical object.
- In networking, it is a network model.
- It can be used for network optimization:
  - What will be the performance if I change this configuration? (e.g., routing)
  - What will happen if there is a failure? (e.g., on links)
Given a network configuration, can you predict the resulting performance?

- A digital twin for networks is fundamentally this box:

  ![Diagram](#)
  
  - Traffic
  - Topology
  - Configuration

  **Digital Twin**

  Performance prediction:
  E.g., per-flow delay, Jitter, loss
How to build a digital twin for networks?

- Networks are fundamentally represented as graphs:
  - Topology
  - Routing
  - Traffic flowing along nodes and links
  - ...

- Traditional neural networks (NN) are not suited to learn from graphs (e.g., Fully connected NN, Convolutional NN, Recurrent NN, etc.)

- Traditional NN-based approach → Feature engineering
  - Ad-hoc solutions for specific problems, usually transforming the problem to prevent learning graphs
  - Limited performance, not applicable to complex real-world scenarios
  - Unable to generalize to other networks!
How can we design deployable ML solutions for networking?

NETWORKING LAB

ML model (e.g., neural network)

+ Controlled testbed

FINAL PRODUCT

Training

Deployment

Customer’s network

19/06/2020
Generalization problem of traditional ML solutions for networks

- Main problem to achieve deployable ML-based solutions for networks:
  - Traditional ML solutions are not able to generalize to other networks
Main problem to achieve deployable ML-based solutions for networks:

- Traditional ML solutions are not able to generalize to other networks
- It is unfeasible to train ML-based optimization tools directly on customer’s network:
  - It would require costly network instrumentation and might cause service disruption due to possible misconfiguration!
  - The same applies to transfer learning (need re-training on customers’ networks)
• Main problem to achieve deployable ML-based solutions for networks:
  • Traditional ML solutions are not able to generalize to other networks
  • It is unfeasible to train ML-based optimization tools directly on customer’s network:
    - It would require costly network instrumentation and might cause service disruption due to possible misconfiguration!
    - The same applies to transfer learning (need re-training on customers’ networks)

The ML product fails to operate in the customer’s network

Need for ML models able to generalize to other networks not seen during training
• Graph Neural Networks is the only ML-based technique that is able to generalize over networks

• Non-ML alternatives to build digital twins:
  - Network simulation → Accurate, but computationally expensive
  - Queuing theory → Unable to model complex real-world networks

• Advantages of GNN with respect to state-of-the-art solutions:
  - Fast (low computational cost)
  - High accuracy
  - Deployability → Unlike other ML-based solutions, it generalizes to other networks!
GNN applied to networking

• GNNs learn the underlying relationships between network elements represented in the form of graphs

• As a result, they can model accurately other networks not seen during the training phase

• Standard GNNs (e.g., chemistry) are not directly suitable for computer networks

• Need for custom GNN models adapted to operate on different networking use cases

GNN is a generic toolbox to build solutions for networking
Looking at other fields: Computer Vision

- Convolutional Neural Networks (CNN) led to a breakthrough in applications and services
- CNNs are well suited to model spatially structured data (e.g., images)

Facial Recognition

Self-driving Cars

Graph Neural Networks are to computer networks what Convolutional Neural Networks are for computer vision
Graph Neural Networking challenge 2020

https://bnn.upc.edu/challenge2020
Problem overview:

- **Input:**
  - Network topology
  - Source-destination traffic matrix
  - Network configuration:
    - Routing
    - Queue scheduling policy on nodes (Strict priority, Weighted Fair Queueing and Deficit Round Robin)

- **Output:**
  - Mean per-packet delay on each source-destination flow
• Generated with the OMNet++ packet-accurate network simulator

• Thousands of simulation samples with topologies, routings, queue scheduling configurations, and traffic (large range of traffic intensities)
Queue scheduling configurations:

- All network nodes have three queues associated to three different traffic classes (different priorities)

- Samples of four different scenarios (25% samples each one):
Queue scheduling configurations:

• All network nodes have three queues associated to three different traffic classes (different priorities)

• Samples of four different scenarios (25% samples each one):
  • **Scenario 1** → All nodes implement Weighted Fair Queuing (WFQ) with fixed weights on queues (60 for Queue #1, 30 for Queue #2, and 10 Queue #3)
Queue scheduling configurations:

• All network nodes have three queues associated to three different traffic classes (different priorities)

• Samples of four different scenarios (25% samples each one):
  • **Scenario 1** → All nodes implement Weighted Fair Queuing (WFQ) with fixed weights on queues (60 for Queue #1, 30 for Queue #2, and 10 Queue #3)
  • **Scenario 2** → All nodes implement WFQ with variable weights assigned to queues
Queue scheduling configurations:

- All network nodes have three queues associated to three different traffic classes (different priorities)

- Samples of four different scenarios (25% samples each one):
  - **Scenario 1** → All nodes implement Weighted Fair Queuing (WFQ) with fixed weights on queues (60 for Queue #1, 30 for Queue #2, and 10 Queue #3)
  - **Scenario 2** → All nodes implement WFQ with variable weights assigned to queues
  - **Scenario 3** → Nodes can implement Strict Priority (SP), WFQ, or Deficit Round Robin (DRR). WFQ and DRR include variable weights on nodes
Queue scheduling configurations:

- All network nodes have three queues associated to three different traffic classes (different priorities)

- Samples of four different scenarios (25% samples each one):
  - **Scenario 1** → All nodes implement Weighted Fair Queuing (WFQ) with fixed weights on queues (60 for Queue #1, 30 for Queue #2, and 10 Queue #3)
  - **Scenario 2** → All nodes implement WFQ with variable weights assigned to queues
  - **Scenario 3** → Nodes can implement Strict Priority (SP), WFQ, or Deficit Round Robin (DRR). WFQ and DRR include variable weights on nodes
  - **Scenario 4** → Similar to scenario 3 but it defines different traffic profiles for the three traffic classes
• Public data sets:
  • Training and validation datasets
  • See all the details at: https://challenge.bnn.upc.edu/dataset

• Python API to easily read and process the dataset
  https://github.com/knowledgedefinednetworking/datanetAPI
• RouteNet* learns the relations between topology, traffic, routing and how these elements affect the resulting network performance (e.g., delay)

• Generalizes to unseen topologies, routing configurations and traffic

*Rusek, K., Suárez-Varela, J., Mestres, A., Barlet-Ros, P. and Cabellos-Aparicio, A., 2019. Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN. In ACM SOSR 2019
• Open source implementation in TensorFlow 2.1
  
  https://github.com/knowledgedefinednetworking/RouteNet-challenge

- RouteNet* is not designed to model the impact of different queue scheduling policies on nodes

*Rusek, K., Suárez-Varela, J., Mestres, A., Barlet-Ros, P. and Cabellos-Aparicio, A., 2019. Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN. *In ACM SOSR 2019*
**Objective:** Test the generalization capabilities of neural network solutions:

- Training dataset → Samples simulated in two network topologies
- Validation and Test datasets → Samples simulated in a third topology

- The test data set will be released at the end of the challenge (Sep 11\textsuperscript{th}) and the evaluation phase will start just after that

- We will evaluate the capability of the proposed solutions to make good delay predictions in the test dataset
• The test dataset will be unlabeled (i.e., no delay measurements)

• Participants have to label this dataset with their neural network models and send the results in CSV format

• Evaluation score $\rightarrow$ MAPE (Mean Absolute Percentage Error)

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right|$$

Lower is better!
Guidelines for participants

• Participants are encouraged to update RouteNet or design their own neural network architectures

• How to update RouteNet:
  • Modify the neural network architecture to model different queue scheduling policies on nodes
  • Hyper-parameter tuning, normalization...

• We provide a tutorial on how to run RouteNet and modify the code
  
  [https://github.com/knowledgedefinednetworking/RouteNet-challenge](https://github.com/knowledgedefinednetworking/RouteNet-challenge)

19/06/2020
Graph Neural Networking challenge 2020

- **Organized as part of the ITU AI/ML in 5G Challenge** (Ref: ITU-ML5G-PS-014)
  Special thanks to ITU for making this possible!

- **Target audience:**
  - Networking community
  - AI community (GNN is a hot topic!)

- **Main resources:**
  - Baseline model and tutorial → RouteNet*
  - **API** to easily read and process the datasets
  - **Mailing list** to engage participants


Incentives for participants

• Good opportunity to be introduced in the application of GNN for networking
  
  This is the first competition in the world on GNN applied to networks!

• Access to the global round of the ITU AI/ML in 5G challenge:
  • Top candidates will be considered by the ITU judging committee
  • Awards and presentation at the final conference (Nov-Dec 2020)
  • More details at: ITU AI/ML 5G Challenge: Participation Guidelines
Incentives for participants

• Good opportunity to be introduced in the application of GNN for networking
  
  **This is the first competition in the world on GNN applied to networks!**

• Access to the global round of the ITU AI/ML in 5G challenge:
  • Top candidates will be considered by the ITU judging committee
  • Awards and presentation at the final conference (Nov-Dec 2020)
  • More details at: [ITU AI/ML 5G Challenge: Participation Guidelines](#)

• Top 3 teams will be recognized in the challenge website and will receive certificates of appreciation

• Presentation of the winning solution at the BigDama workshop (tentatively co-located with ACM CoNEXT 2020 – Dec 2020)

• Possibility to publish a paper co-authored with the challenge organizers
Organizing team

José Suarez-Varela  Albert López  Miquel Ferriol  Guillermo Bernárdez  Paul almasan  Krzysztof Rusek

Prof. Pere Barlet-Ros  Prof. Albert Cabellos

Barcelona Neural Networking Center

19/06/2020
Main dates

Graph Neural Networking Challenge 2020

See all the details at: https://bnn.upc.edu/challenge2020

- **Registration is now open to all participants** (teams up to 4 members)
- **Challenge duration** → May 22nd-Oct 21st (≈ 5-month duration)
- **Registration deadline** → Jun 30th
- **Evaluation phase** → Sep 11th-Sep 25th
- **Winners (top 3) official announcement** → Oct 21st
- **ITU final conference and awards** → Nov-Dec 2020

ITU registration link: [here]
Slack channel: [here]
ITU ML5G-PS-012: ML5G-PHY
(Universidade Federal do Pará, Brazil)
26 June 2020

ITU
AI/ML in 5G Challenge

Applying machine learning in communication networks

ai5gchallenge@itu.int