Certifiable Robustness of Graph Convolutional Networks under Structure Perturbations

Daniel Zügner, Stephan Günnemann
Technical University of Munich, Germany
KDD 2020
Deep Learning on Graphs

Graphs are ubiquitous
- Social networks
- Web graphs
- Knowledge graphs

Graph neural networks (GNNs): state of the art on tasks such as
- **Semi-supervised node classification** (our work’s focus)
- Link prediction
- Unsupervised representation learning (node embeddings)
Background: GNN Node Classification

Given: Graph $G = (A, X)$ with connectivity $A$ and node features $X$

Predict a node’s class by using its neighborhood.

\[ \hat{Y} = \text{softmax}(\hat{A} \sigma(\hat{A} X W^{(1)} + b^{(1)})W^{(2)} + b^{(2)}) \]

Pre-processed adjacency matrix
Graph Neural Networks are not robust w.r.t. adversarial attacks.

[Dai et al. 2018]
[Wang et al. 2018]
[Zügner et al. 2018], [Zügner and Günnemann 2019]

Adversarial attacks on GNNs:
  • Node feature perturbations
  • **Graph structure perturbations** (this work’s focus)
Graph Neural Networks are not robust

GNNs are not robust w.r.t. adversarial attacks.

[Dai et al. 2018]
[Wang et al. 2018]
[Zügner et al. 2018], [Zügner and Günnemann 2019]

Adversarial attacks on GNNs:

- Node feature perturbations
- **Graph structure perturbations** (this work’s focus)
High-level approach

- We assume that the present graph can contain up to a certain number of adversarial edges, i.e. an attacker has inserted edges.
- In other words, the clean (unknown) graph is reachable by removing edges from the present graph according to the adversarial budget.
- That is, by inserting edges the adversary could have changed the model’s prediction.

Certifiable Robustness of Graph Convolutional Networks under Structure Perturbations – Daniel Zügner and Stephan Günnemann.
Finding the worst-case perturbation

- **Robustness certification**: check whether there exists a graph reachable by removing edges for which the node has a different predicted label.

  ➔ **Certify** robustness if it is guaranteed that the prediction **does not change**.

- Since **enumerating** all reachable graphs is **intractable**, we propose to **bound** the maximum change in logits (**“worst-case”** perturbation).

- **$L_0$-bounded** perturbations on the graph structure
  - At most $q$ adversarial edges per node
  - At most $Q$ adversarial edges in total (over all nodes)
Optimization problem view

• Finding the “worst-case” perturbation as an optimization problem:

\[
\tilde{m}^t(y^*, y) := \min_A f(A, X)_{ty^*} - f(A, X)_{ty} \quad \text{subject to budget constraints}
\]

• \( \tilde{m}^t(y^*, y) > 0 \) for all classes \( y \): predicted class cannot be changed (robust).
• \( \tilde{m}^t(y^*, y) < 0 \) for any class \( y \): prediction is not robust.
Optimization problem challenges

Challenges for optimization:

- $f$ has **nonlinear activation functions**
  - can be addressed via linear relaxation.

- Optimization over **binary variable** ($A \in \{0,1\}^{N \times N}$)

- Problem contains **nonconvex products** of variables

\[
\tilde{m}^t(y^*, y) := \min_A f(A, X)_{ty^*} - f(A, X)_{ty}
\]

*subject to budget constraints*

Certifiable Robustness of Graph Convolutional Networks under Structure Perturbations – Daniel Zügner and Stephan Günnemann.
Challenge: binary variables

Addressing the challenge of optimization over **binary variable** \( A \in \{0,1\}^{N\times N} \)

- Naïve solution: **continuous relaxation**, i.e. \( A \in [0,1]^{N\times N} \)
- However, GCN performs **preprocessing** on \( A \):

  Pre-processed matrix:  
  \[
  \widehat{A}_{ij} = \begin{cases} 
  \frac{1}{\sqrt{(\text{deg}(i)+1)\cdot(\text{deg}(j)+1)}} & \text{If nodes } i \text{ and } j \text{ are connected (or } i=j) \\
  0 & \text{else}
  \end{cases}
  \]

  Values \( \widehat{A}_{ij} \) are **not convex** in the variables \( (A_{ij}) \)
Challenge: binary variables

Our approach:

• directly optimize the continuous-valued message passing matrix $\hat{A} \in \mathbb{R}^{N \times N}$

• Derive constraints on $\hat{A}$ that are induced by the budget constraints on $A$ and the preprocessing procedure.

• E.g.: node degrees cannot increase.

$\Rightarrow \hat{A}_{ij}$ must be in $\left[0, \frac{1}{\sqrt{(\deg(i)+1-q) \cdot (\deg(j)+1-q)}}\right]$

• In the paper we derive induced constraints on an element-, row-, and global level.
Challenge: nonconvex variable products

• At each GCN layer we perform message passing in the form of multiplying by $\widehat{A}$:

$$H^{(l)} = \sigma(\widehat{A} H^{(l-1)} W^{(l)} + b^{(l)})$$

  \hspace{1cm}\text{nolinear terms}

  \hspace{1cm}\text{depends on $\widehat{A}$}

• This introduces non-convex bilinear terms of the variables.

• A naïve solution is to relax these terms using the reformulation-linearization (RLT) technique. However, this leads to loose approximations.
Challenge: nonconvex variable products

- We show how to phrase the problem as a *jointly constrained bilinear program*:

  \[
  \min_{x, z} \; x^T z 
  \quad \text{subject to } \; x + z \leq c 
  \]

  (and other constraints)

- Can be solved *exactly* via *branch-and-bound (B&B)* techniques.

- We design a custom *B&B-scheme* for *provably robust GCN*:
  - converges *faster* and
  - produces *tighter bounds* (compared to standard B&B).
Optimization properties

- At each B&B iteration we solve a **linear program**, for which we can use highly-optimized **off-the-shelf solvers**.

- We **refine** upper and lower bounds at each iteration.

- Finding the **global optimum** can take an infinite number of iterations.

- Since we are only interested in the **sign of the optimum**, we can stop when either the upper or lower bound **crosses zero**.
Optimization

Once the upper or lower bound crosses zero, we can stop optimization.

**Lower bound** > 0: provably robust
**Upper bound** < 0: no decision

In about 80% of the cases, our algorithm terminates within 50 iterations.

Dataset: Cora-ML
Comparison to linear relaxation

- Our **branch-and-bound** method can prove robustness for many more nodes than a linear relaxation baseline.

- Even a **single edge perturbation** can change the predicted labels of up to 11% of nodes.
Robust training comparison

- We compare various robust training schemes in terms of their robustness share.
- In line with [Dai et al 2018], adversarial training does not seem to help.
- Interestingly, the only consistently more robust training method is [Zügner+ 2019]’s method for improved robustness for attribute attacks.
• **Robustness certification** of GCN for perturbations of the graph structure

• Our novel **branch-and-bound** algorithm substantially outperforms a linear relaxation approach

• **Adversarial training** does not seem to improve provable robustness. This highlights the need for **novel robust training** schemes.

• Paper, code & more: [www.daml.in.tum.de/robust-gcn](http://www.daml.in.tum.de/robust-gcn)