Robustness & Graph (Convolutional) Neural Networks

Tim Bohne

25. Februar 2021

Machine Learning Seminar 20/21

Übersicht

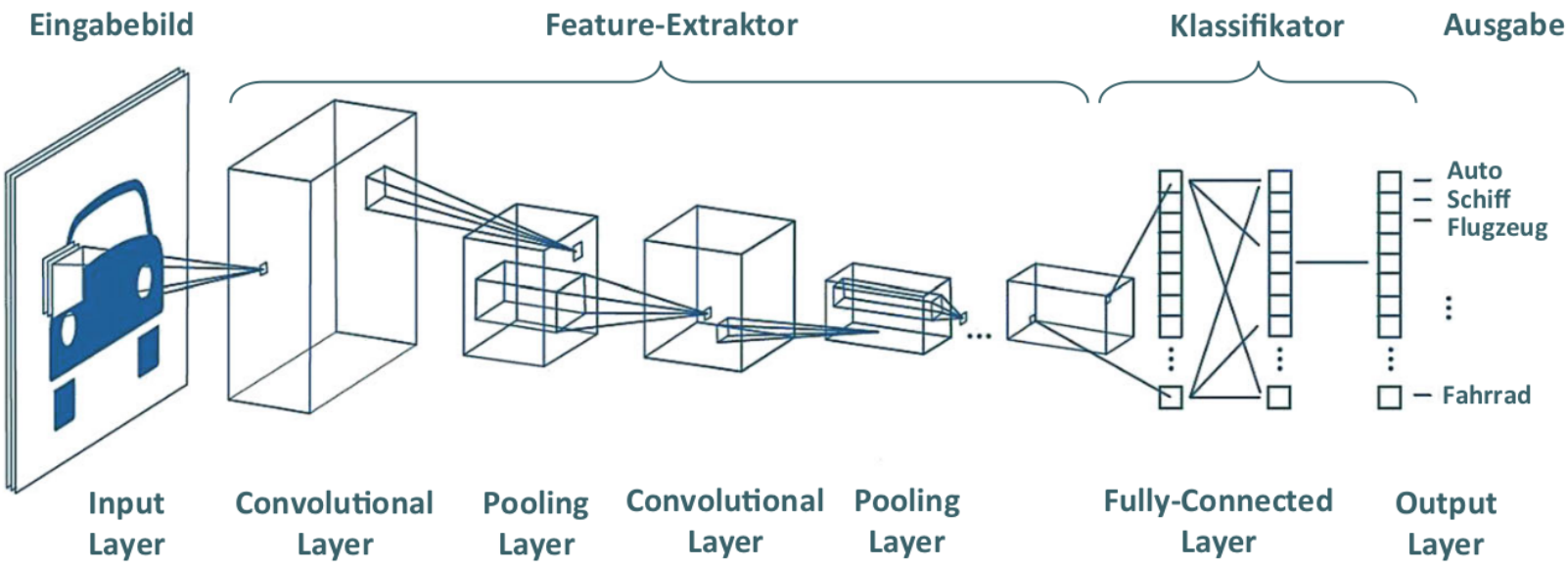
1. Motivation
2. Graph Neural Networks (GNNs)
3. Robustheit
   - GNNs: Manipulation der Knoten-Attribute
   - GNNs: Manipulation der Graph-Struktur
4. Fazit / Ausblick

Motivation

Motivationen für Graph Neural Networks:

1. Convolutional Neural Networks (CNNs)
2. Graph Embedding

Convolutional Neural Networks (CNNs)



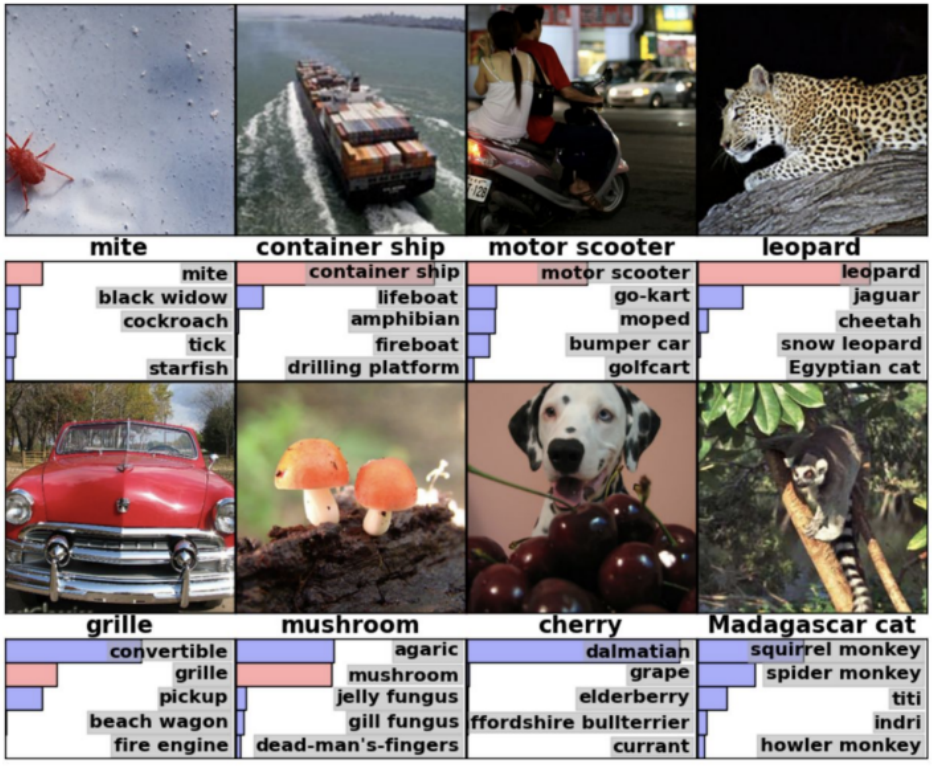
Convolutional Neural Network [9]

- **Feature-Learning**
  Sequenz aus Convolutional- und Pooling-Layern
- **Classification**

Convolutional Neural Networks (CNNs)

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

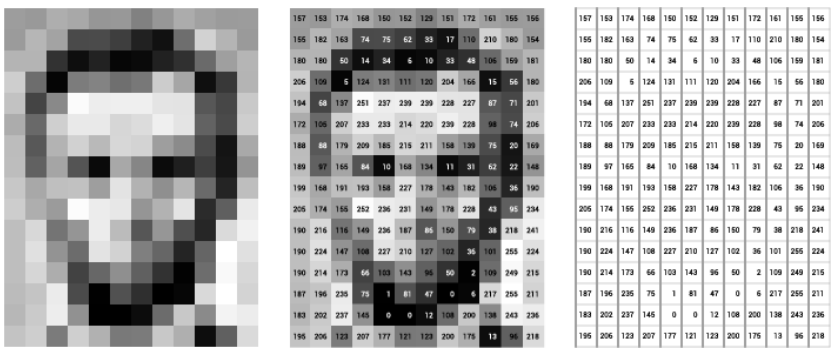


ImageNet Challenge [6]

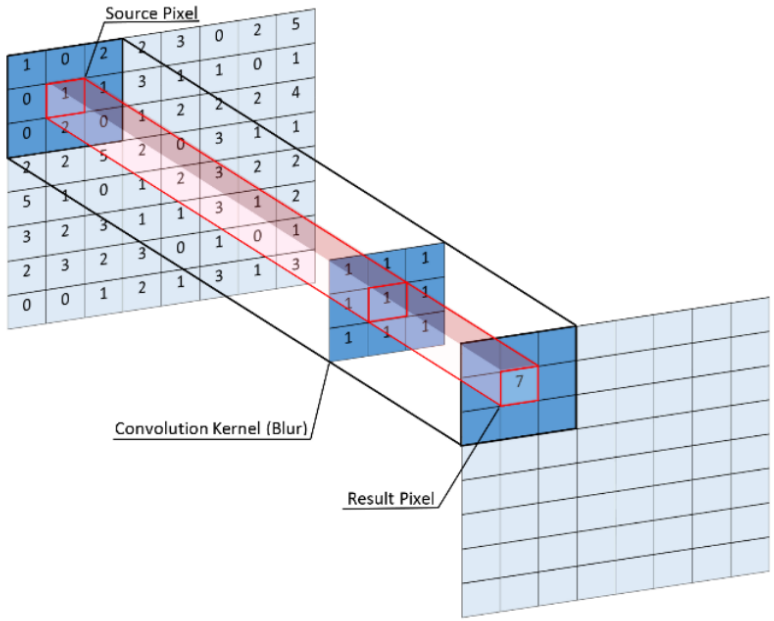
Hier sind CNNs sehr erfolgreich!

Convolutional Neural Networks (CNNs)

Problem: CNNs funktionieren lediglich mit Euklidischen Datenstrukturen (Bilder, Text), nicht mit Graphen!



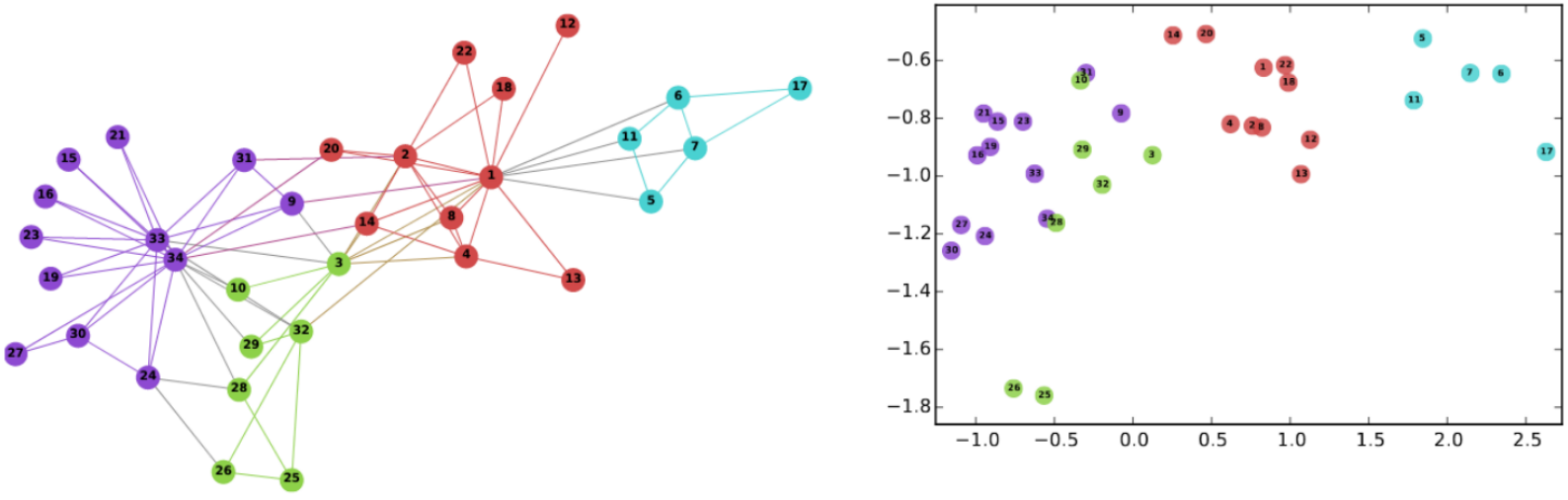
Pixelmatrix [8]



Convolution-Operation [4]

Graph Embedding

Übersetze Graph-Struktur in niedrigdimensionalen Vektor

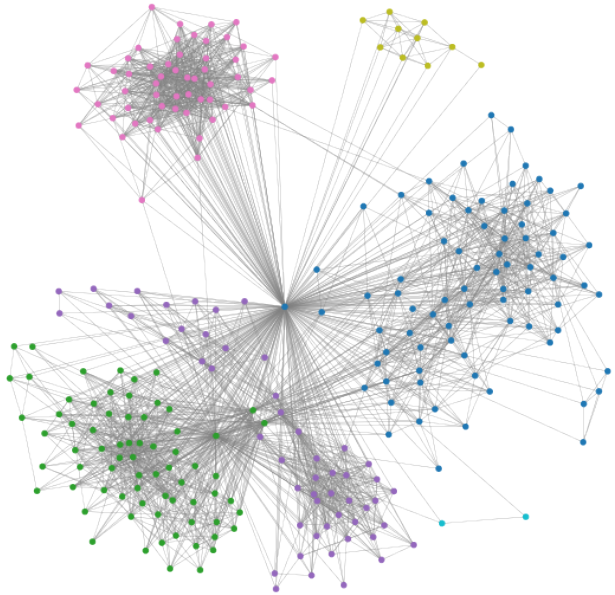


2-dim. Output für jeden Knoten [7]

Graph Neural Networks (GNNs)

Ermöglichen Machine Learning auf Graphen, z.B.:

- Modellierung physikalischer Systeme
- Lernen molekularer Fingerabdrücke
- Analyse sozialer Netzwerke (Vorhersagen / Empfehlungen)
- Empfehlungssysteme



Freunde-Netzwerk einer Person [1]

Graph Neural Networks (GNNs)

Typische Aufgaben für GNNs:

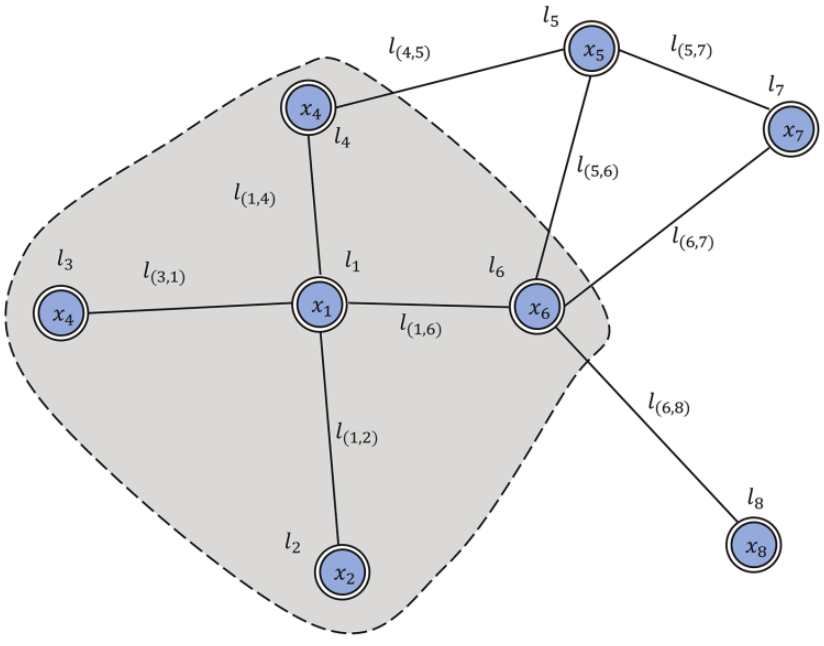
- Semi-Supervised Node Classification
- Graph Classification
- Link Prediction
- Clustering

Graph Neural Networks (GNNs)

Graph $\boldsymbol{G = (X, A)}$

$\boldsymbol{X}$: Knoten - Personen in sozialem Netzwerk

$\boldsymbol{A}$: Kanten - Beziehungen zwischen Personen



Graph basierend auf Scarselli et al. [5]

Ziel: Lerne „Node Embedding“ $\boldsymbol{h_v}$ für jeden Knoten $\boldsymbol{v} \in \boldsymbol{X}$ um einen Output $\boldsymbol{o_v}$ zu generieren, z.B. das vorhergesagte Label

Graph Neural Networks (GNNs)

2 wichtige Funktionen (x: Input Feature, h: Hidden State):

1. **Node Embedding:** $h_v = f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]})$
2. **Output Embedding:** $o_v = g(h_v, x_v)$

Lerne Parameter von f und g:

Loss Term: $\sum_{i=1}^{p}(t_i - o_i)$

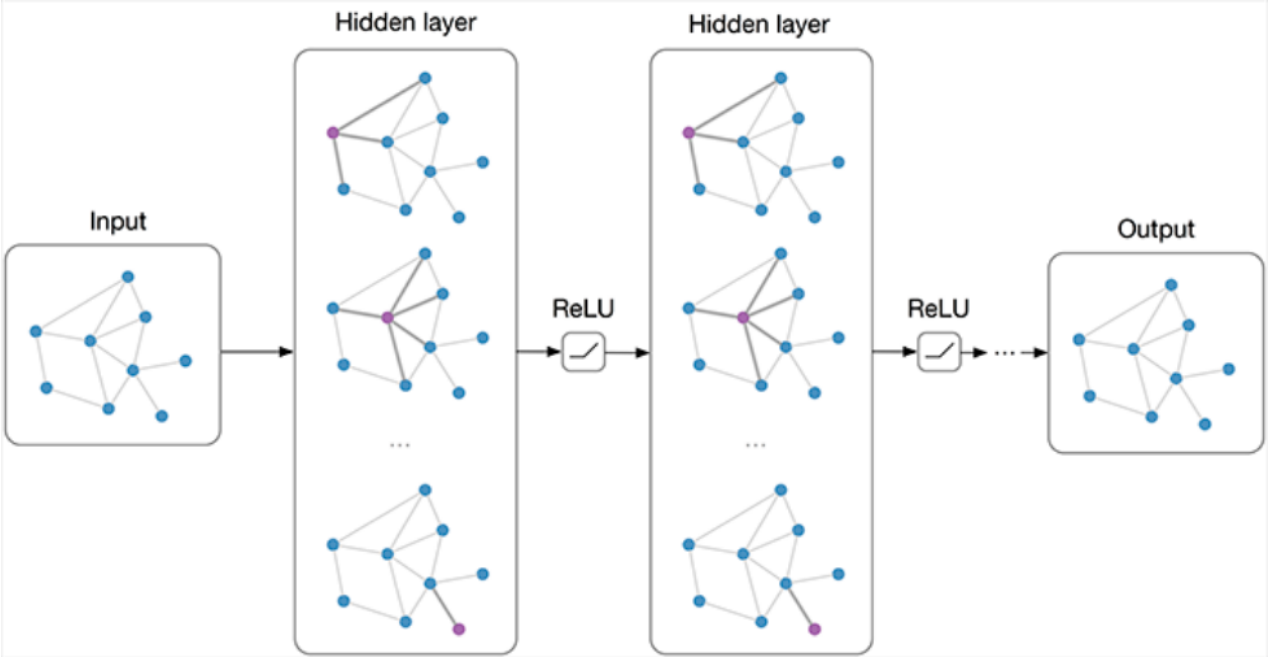
Lernmethode: Gradient Descent

- Iteratives Update: $H^{t+1} = F(H^t, X)$
- Gradient wird basierend auf dem Loss berechnet
- Gewichte W werden basierend auf dem Gradienten adaptiert

Graph Convolutional Networks (GCNs)

Input: Features X, Adjazenzmatrix A

Output: Feature-Vektor für jeden Knoten



Multi-Layer GCN [3]

Anschließend „Node Classification“ basierend auf den Output-Features

Graph Convolutional Networks (GCNs)

Standard-Convolution kann nicht für Graphen definiert werden!

Lösung: Embeddings (spectral / spatial)

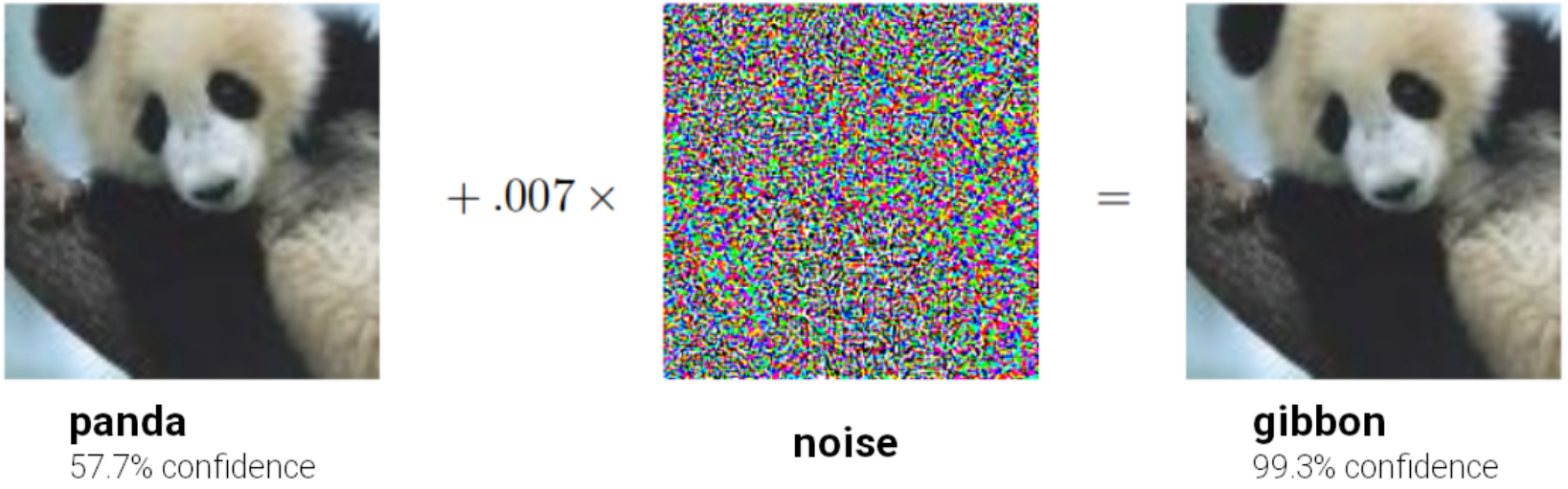
Approximierte „Graph-Convolution“ (Kipf et al. [3]):

$$H^{(l+1)} = \sigma[\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}]$$

Sehr erfolgreich in praktischen Anwendungen eingesetzt!

Robustheit von Machine Learning Modellen

Machine Learning Modelle sind i.A. anfällig für „Adversarial Attacks“

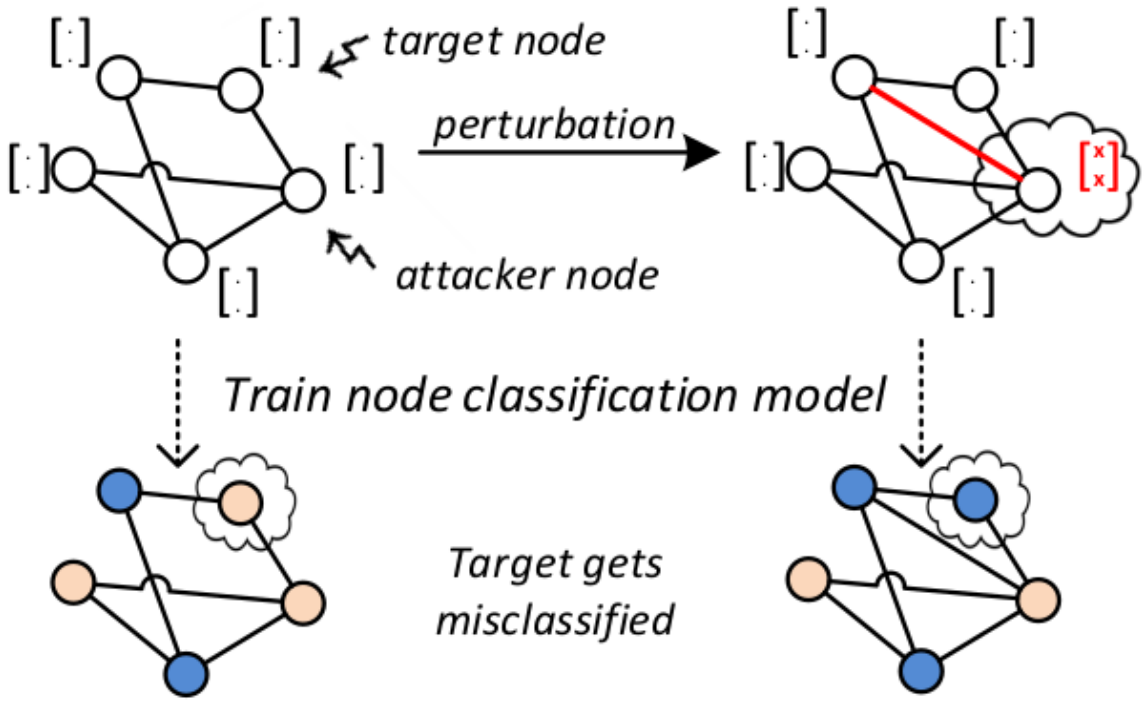


Adaptiert aus Goodfellow et al. [2]

Robustheit von GNNs

Dies gilt auch für GNNs, wobei sich zwei Arten von Angriffen unterscheiden lassen:

- **Manipulation der Knoten-Attribute**
- **Manipulation der Graph-Struktur**



Zügner et al. [10]

Robustheit von GNNs

3 Phasen in der Literatur:

1. GNNs anfällig für „Adversarial Attacks“
2. Verteidigungsmechanismen für konkrete Szenarien
3. Beweisbare Garantien für die Robustheit bestimmter Modelle

Semi-Supervised Node Classification

Graph $\boldsymbol{G} = (\boldsymbol{A}, \boldsymbol{X})$ Zielknoten $\boldsymbol{t}$ Trainierbare Parameter $\boldsymbol{\theta}$

$\boldsymbol{A}$: Adjazenzmatrix, $\boldsymbol{X}$: Knoten-Features

$\boldsymbol{V}$: Knotenmenge

$\boldsymbol{V_L} \subseteq \boldsymbol{V}$: Teilmenge der gelabelten Knoten

$\boldsymbol{\mathcal{T}(A)}$: „Message-Passing-Matrix“ - wie Aktivierungen durch das GNN propagiert werden $\rightarrow$ Transformation der Adjazenzmatrix

Cross-Entropy-Loss Minimierung:

Lerne $\boldsymbol{\theta}$ unter Verwendung der $\boldsymbol{V_L}$

Ziel: Label der verbleibenden Knoten ohne Label vorhersagen

Robustheit von GNNs

Zertifizierte Robustheit gegenüber Manipulationen der Knoten-Attribute - Zügner et al. [11]

Wie lässt sich sicherstellen, dass kleine Änderungen an den Attributen keine dramatischen Auswirkungen auf den Output haben?

Zertifizierung der Robustheit bereits trainierter GNNs

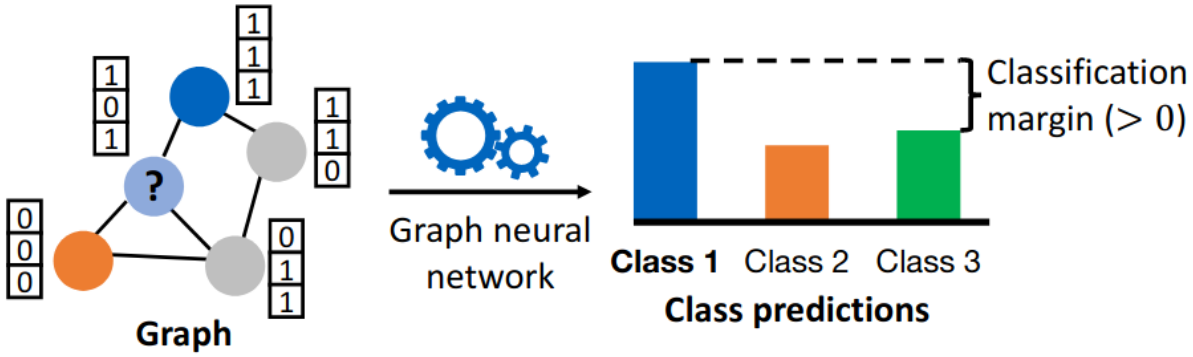
Ziel: Zertifikat für Knoten t bedeutet, dass die Vorhersage für t sich nach zulässigen Manipulationen nicht ändert
$\rightarrow$ Definiertes Angriffsmodell

Idee: „**Worst-Case-Margin**“ m^t für Knoten t zwischen den Klassen y und y^* bzgl. zulässiger Manipulationen:

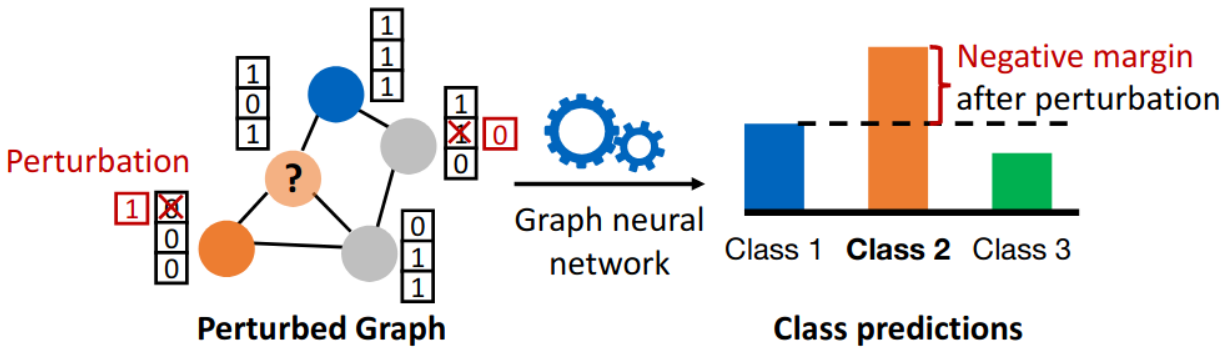
$$m^t(y^*, y) := \min_{\tilde{X}} f_\theta^t(\tilde{X}, A)_{y^*} - f_\theta^t(\tilde{X}, A)_y$$
$$s.t. \quad \tilde{X} \in zul.Manipulationen$$

$m^t > 0 \, \forall \, y \neq y^* \rightarrow$ Modell robust bzgl. Knoten t

High-Level Idee



Vor der Manipulation [11]



Nach der Manipulation [11]

Lösung des Optimierungsproblems

Optimierungsproblem nicht effizient lösbar:
Diskrete Daten + nicht-konvexe Aktivierungsfunktion

Lösung:

- Konvexe ReLU Relaxation: Transformiert das GNN in ein effizient lösbares LP
- Kontinuierliche Relaxation der ganzzahligen Knotenattribute

$\rightarrow$ Effizient berechenbare Schranken für den „Worst-Case-Margin“
$\rightarrow$ Ggf. „false negatives“, jedoch keine „false positives“

Training mit dem Ziel der Robustheit

Ziel: Erreiche beweisbare Robustheit durch Training

$\rightarrow$ Optimiere bezüglich Robustheit

Robust Hinge Loss

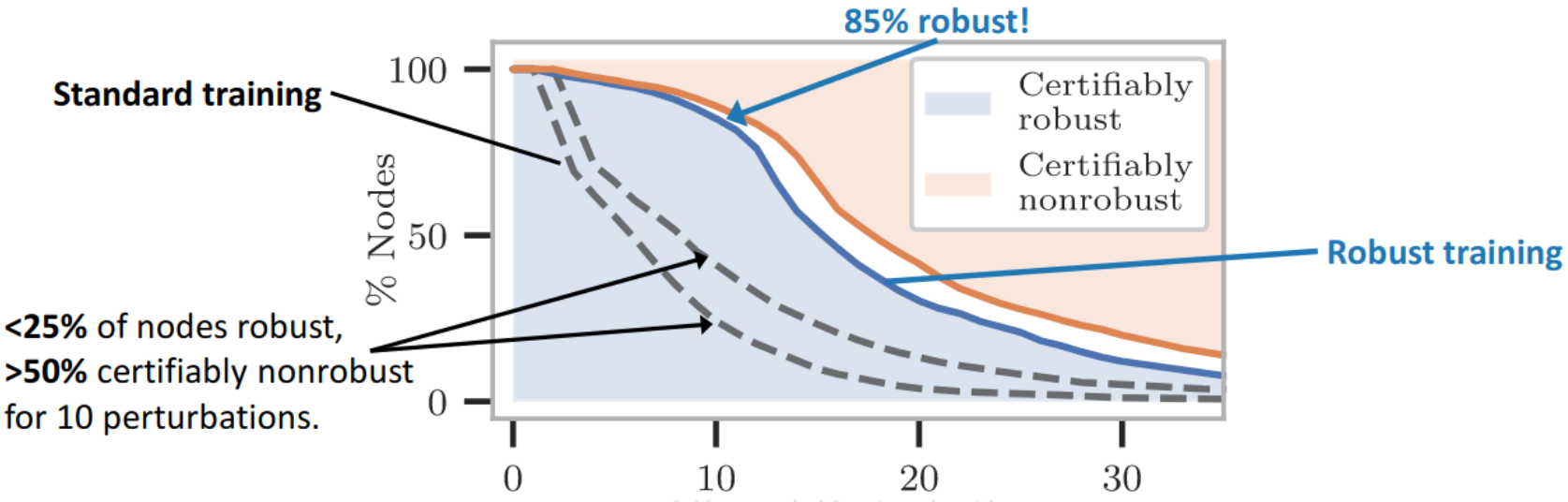
$$\min_{\theta} \sum_{v \in V_L} \mathcal{L}(p_v, y_v) + \sum_{v \in V_L} \hat{\mathcal{L}}_{M_L}(-m_v^*, y_v) + \sum_{v \in V \setminus V_L} \hat{\mathcal{L}}_{M_U}(-m_v^*, \hat{y}_v)$$

- Relaxierte Version um Robustheit sicherzustellen
- Exakte Version für die Klassifizierung

$\rightarrow$ **Mit Standardsoftware robuste GNNs trainieren**

$\rightarrow$ **Qualität der Klassifizierungsergebnisse nicht schlechter**

Training mit dem Ziel der Robustheit

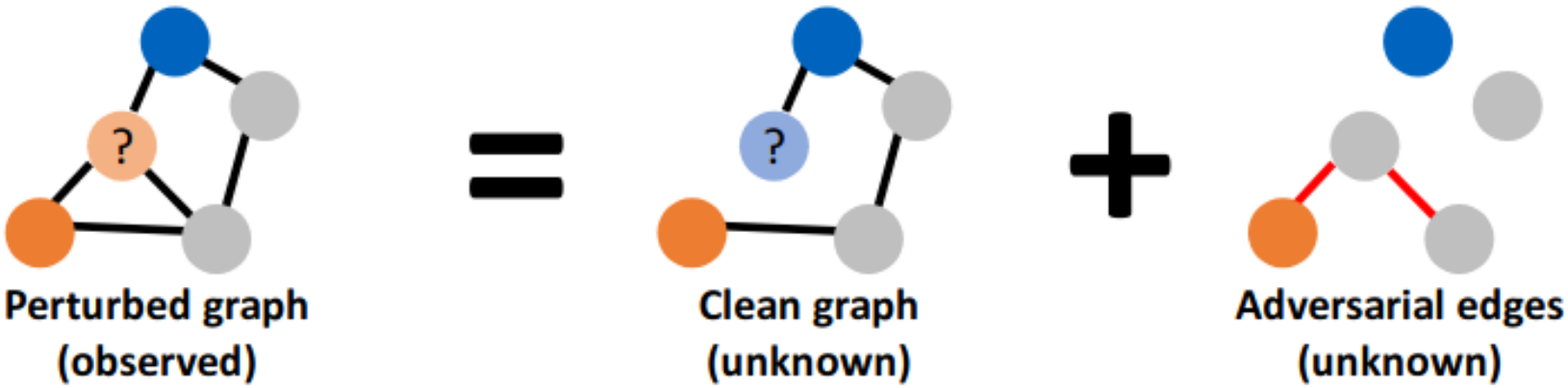


„Robustheits-Training“ vs. Standard-Training [11]

- Nur wenige Knoten nicht zertifizierbar
- Standard-Training führt zu GNNs, die nur robust bzgl. sehr weniger Manipulationen sind

Zertifizierte Robustheit gegenüber Manipulationen der Graph-Struktur - Zügner et al. [12]

Angreifer können neue Kanten in den Graphen einfügen, z.B. Likes in Social Media:



High-Level Idee [12]

Robustheit von GNNs

Einfügen neuer Kanten kann die Vorhersagen des Modells ändern!

Ziel: Ein Zertifikat bedeutet, dass sich die Vorhersage für einen Knoten nicht ändert, wenn ein Angreifer Kanten hinzufügt

Idee: Prüfe, ob es einen Graphen gibt, der durch Entfernen von Kanten erreichbar ist und die Vorhersage ändert

Angriffsmodell: Praktisch unmöglich, alle Graphen zu enumerieren $\rightarrow$ Limitiere Anzahl an „Adversarial Edges“

Optimierungsproblem

- Bereits trainiertes GNN mit Parametern $\boldsymbol{\theta}$
- Input-Graph ($\boldsymbol{A}$) möglicherweise manipuliert
- $\boldsymbol{A}$ ist von der „sauberen“ Variante $\boldsymbol{A}^*$ durch eine Menge zulässiger Manipulationen erreichbar

Worst-Case-Margin:

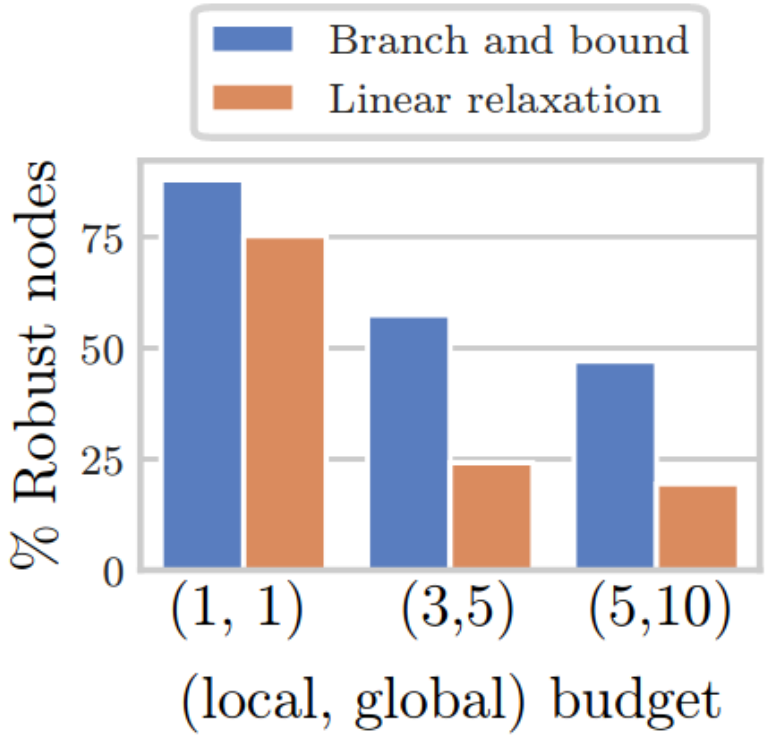
$$\boldsymbol{m^t(y^*, y) := \min_{\tilde{A}} f_\theta^t(X, \tilde{A})_{y^*} - f_\theta^t(X, \tilde{A})_y}$$

$$\boldsymbol{s.t. \quad \tilde{A} \in zul.Manipulationen}$$

$\boldsymbol{m^t > 0 \, \forall \, y \neq y^*} \rightarrow$ Modell robust bzgl. Knoten $\boldsymbol{t}$

Ergebnis

- Effiziente Zertifizierung der Robustheit gegenüber Manipulationen der Graph-Struktur
- Bereits wenige Manipulationen führen zu einem Label-Wechsel eines beträchtlichen Anteils der Knoten



B&B vs. Lineare Relaxation [12]

Fazit

- GNNs sind sehr **erfolgreich** in einer Vielzahl von praktischen Anwendungen einsetzbar
- GNNs sind **nicht robust** - anfällig für Manipulationen
  - der Knoten-Attribute
  - der Graph-Struktur
- Um GNNs in (sicherheitskritischen) praktischen Anwendungen nutzen zu können, ist ein bestimmtes Maß an **Robustheit nötig**
- **Beweisbare Garantien** für die (Nicht-)Robustheit von GNNs sind möglich und nötig - erste Ansätze gesehen

Ausblick

- Spielraum für **Generalisierungen** (z.B. Angriffsmodelle)
- Trainingsmethoden, die **Robustheit explizit als Optimierungsziel** enthalten
- **Robustheitsgarantien** für Attribut- und Struktur-Manipulationen
- **Grundsätzliches Verständnis** - was macht Manipulationen schädlich?

Ausblick

Ideen / Ansätze

- Graphen, die reale Probleme repräsentieren, besitzen bestimmte strukturelle Gemeinsamkeiten, die manipulierte Graphen verletzen
- Statistisch signifikante strukturelle Attribute in Manipulationen entdecken
- Wahrscheinlichkeit dafür, dass bestimmte Manipulationen schädlich sind

$\rightarrow$ **Allgemeinere Angriffsmodelle**
$\rightarrow$ **Manipulationen erkennen + verhindern**
$\rightarrow$ **Robustheit**

Yihui Fan.
Creating and Analysing Facebook Friend Network Graphs Using Python.
https://www.databentobox.com/2019/07/28/facebook-friend-graph/.
Abgerufen am: 15.02.2021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy.
Explaining and harnessing adversarial examples.
In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, San Diego, CA, USA, May 7-9, Conference Track Proceedings*, 2015.

Thomas N. Kipf and Max Welling.
Semi-supervised classification with graph convolutional networks.
In *5th International Conference on Learning Representations, Toulon, France, April 24-26, Conference Track Proceedings*. OpenReview.net, 2017.


Ahmad Kiswani.
Convolution and Normalized Cross Correlation on Kepler Architecture.
https://sipl.eelabs.technion.ac.il/projects/convolution-and-normalized-cross-correlation-on-kepler-architectur

Abgerufen am: 15.02.2021.


Zhiyuan Liu and Jie Zhou.
Introduction to graph neural networks.
Synthesis Lectures on Artificial Intelligence and Machine Learning, 14:1–127, 03 2020.


X. Giro o Nieto.
Image classification on Imagenet (D1L4 2017 UPC Deep Learning for Computer Vision).
https://www.slideshare.net/xavigiro/image-classification-on-imagenet-d1l4-2017-upc-deep-learning-for-c

Abgerufen am: 15.02.2021.


Bryan Perozzi, Rami Al-Rfou, and Steven Skiena.
Deepwalk.
Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, Aug 2014.


Thomas Smits and Melvin Wevers.
The visual digital turn: Using neural networks to study historical images.
Digital Scholarship in the Humanities, 35, 01 2018.


Patrick Zschech, Christoph Sager, Philipp Siebers, and Maik Pertermann.
Mit computer vision zur automatisierten qualitätssicherung in der industriellen fertigung: Eine fallstudie zur klassifizierung von fehlern in solarzellen mittels elektrolumineszenz-bildern.
HMD Praxis der Wirtschaftsinformatik, 07 2020.


Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann.
Adversarial attacks on neural networks for graph data.
In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, August 19-23*, pages 2847–2856. ACM, 2018.


Daniel Zügner and Stephan Günnemann.
Certifiable robustness and robust training for graph convolutional networks.
In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, August 4-8*, pages 246–256. ACM, 2019.


Daniel Zügner and Stephan Günnemann.
Certifiable robustness of graph convolutional networks under structure perturbations.
In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27*, pages 1656–1665. ACM, 2020.