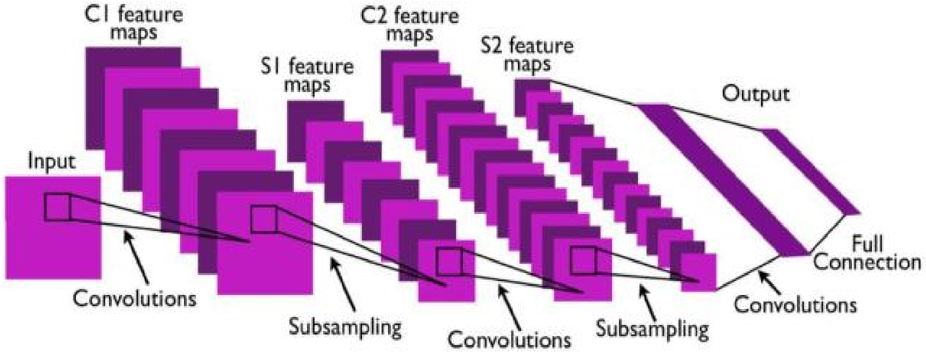
Adversarial Attacks on Neural Networks for Graph Data

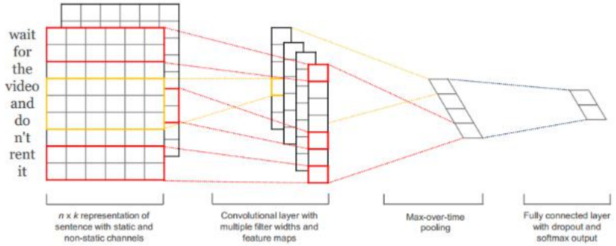
Deepboy

2018.11.30

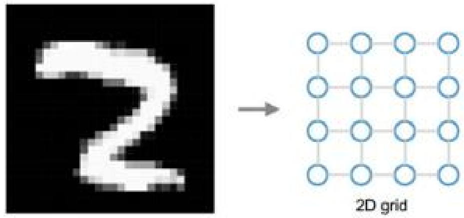
Background of GCN



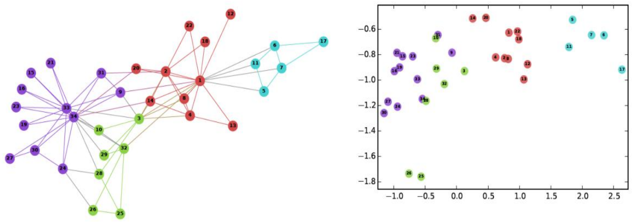
Convolutional neural network



NLP: CNN for Sentence Classification

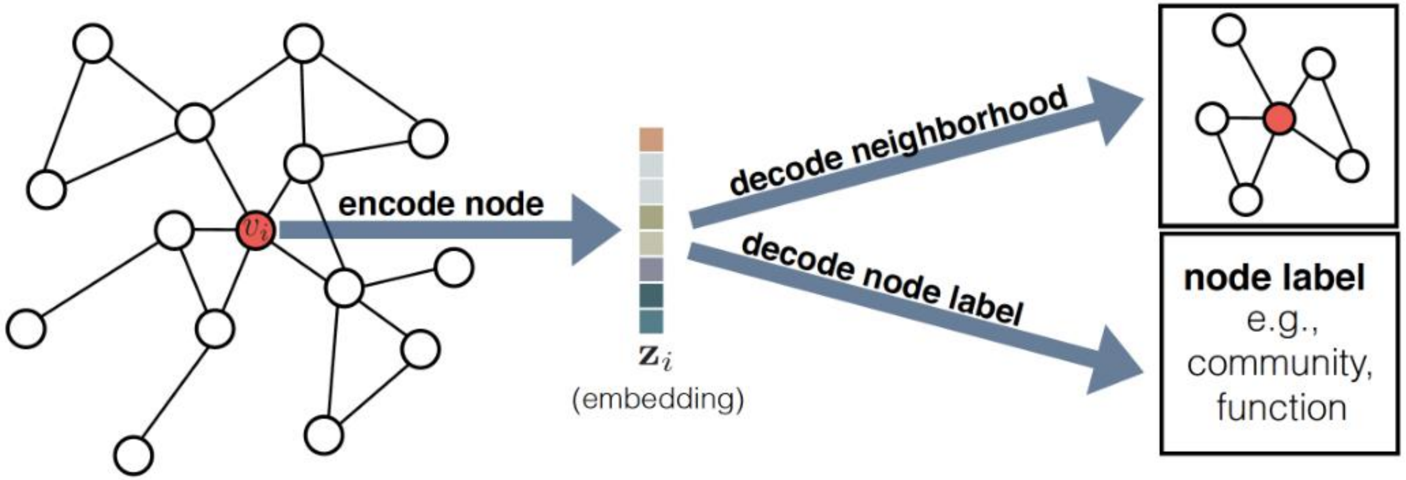


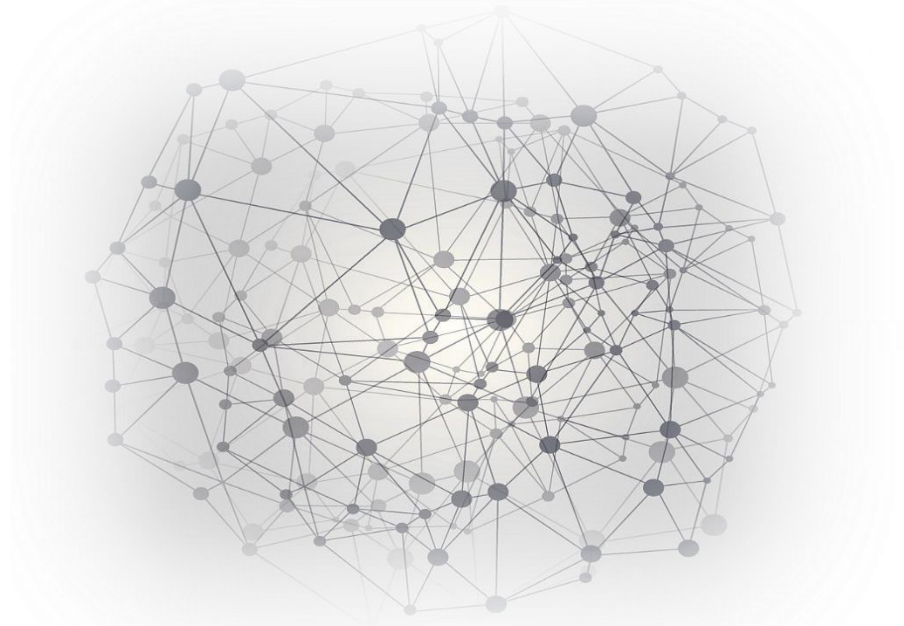
CV: CNN for Image Classification



Graph: Graph embedding

Background of GCN



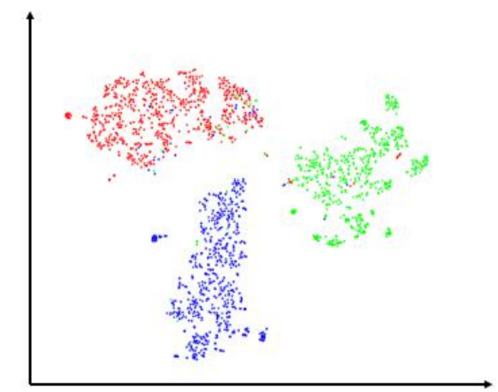


Input :

	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	1	1
C	1	0	0	1	0
D	1	1	1	0	1
E	0	1	0	1	0

Output :

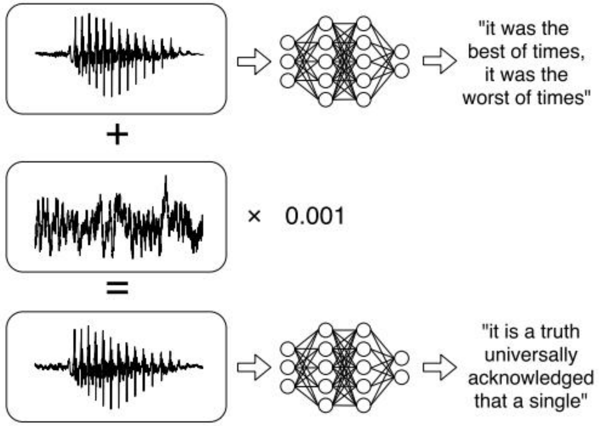
(i = 0,..,n)



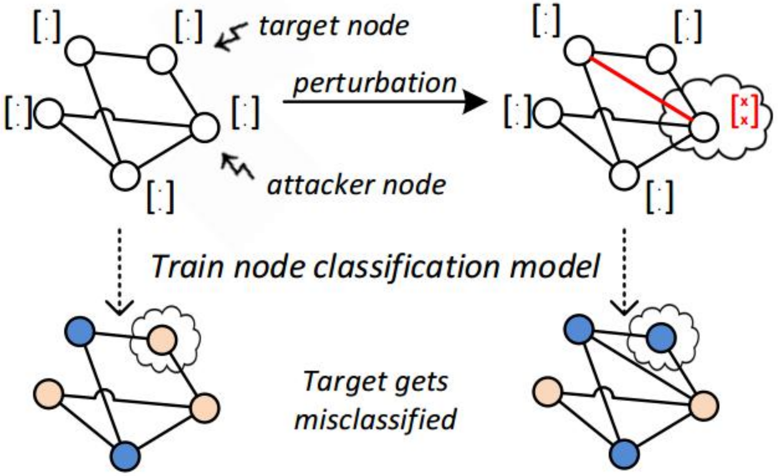
Adversarial Attacks on Neural Networks

Implementation in different fields:

Computer Vision



Audio



Graph

Adversarial Attacks on Graph Convolutional Networks

Latest reference works

- 2018-KDD-Adversarial Atacks on Neural Networks for Graph Data

  Preprints-Adversarial Attacks on Node Embeddings

  Technical University of Munich, Germany

- 2018-ICML-Adversarial Attack on Graph Structured Data

  1. Georgia Institute of Technology
  2. Ant Financial

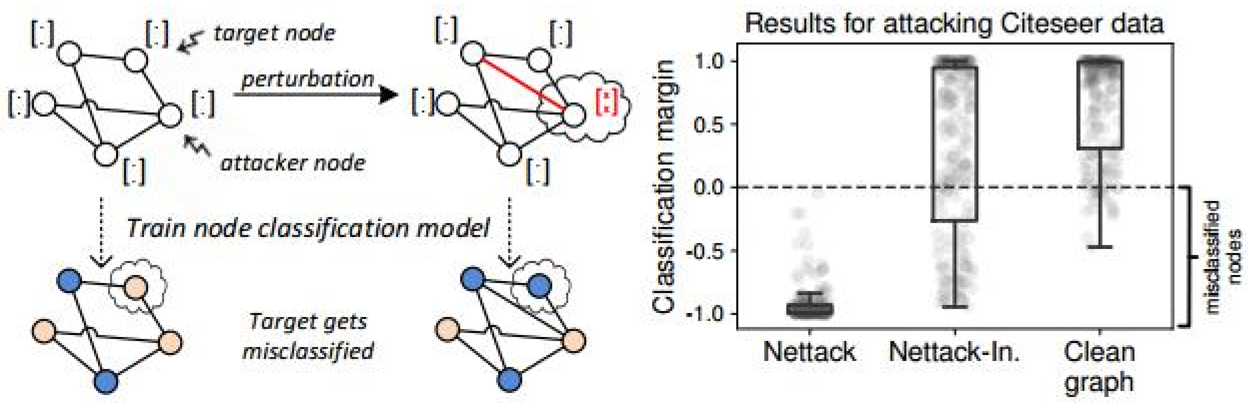
- 2018-AAMAS-Adversarial Classification on Social Networks

Adversarial Attacks on Neural Networks for Graph Data

Main Contribution:

- The first study of adversarial attacks on attributed graphs;
- Implement attacks at test time and poisoning / causative attacks at training phase
- The attacks are transferable.

Main Content:



adversarial attacks against node classification tasks

Adversarial Attacks on Neural Networks for Graph Data

Node classification:

Semi-supervised task

Let **G** = (A, X) be an attributed graph: the adjacency matrix **A** ∈ $\{0,1\}^{N\times N}$ and node's features matrix X ∈ $\{0,1\}^{N\times D}$.

Classification algorithm:

$$Z = f_\theta(A, X) = \mathrm{softmax}\left(\hat{A}\,\sigma\left(\hat{A}XW^{(1)}\right)W^{(2)}\right)$$

Attack model:

Attack goal

Original graph $\mathbf{G}^{(0)} = (\mathbf{A}^{(0)}, \mathbf{X}^{(0)}) \xrightarrow{\text{perturbations}}$ Adversarial graph $\mathbf{G}' = (\mathbf{A}', \mathbf{X}')$

Change $A^{(0)}$ — **Structure attacks**

Change $X^{(0)}$ — **Feature attacks**

Adversarial Attacks on Neural Networks for Graph Data

Attack model:

Target vs. Attackers.

Attack a specific target node v_0 , aim to change v_0's prediction.

1. Perturb v_0
2. Change other nodes

Limitations

Ability: $X'_{ui} \neq X^{(0)}_{ui} \Rightarrow u \in \mathcal{A} \quad , \quad A'_{uv} \neq A^{(0)}_{uv} \Rightarrow u \in \mathcal{A} \vee v \in \mathcal{A}$

Budget: $$\sum_u \sum_i |X^{(0)}_{ui} - X'_{ui}| + \sum_{u<v} |A^{(0)}_{uv} - A'_{uv}| \leq \Delta$$

$v_0 \notin \mathcal{A}$: Influencer attack

$v_0 \in \mathcal{A}$: Direct attack

Adversarial Attacks on Neural Networks for Graph Data

Unnoticeable Perturbations:

Difficulties

(i) The graph structure is discrete preventing to use infinitesimal small changes

(ii) Sufficiently large graphs are not suitable for visual inspection

Solution

Core idea is to allow only those perturbations that preserve specific inherent properties for the input graph

Degree distribution

Feature statistics preserving

Adversarial Attacks on Neural Networks for Graph Data

Experiments

Dataset

Dataset	N_{LCC}	E_{LCC}
Cora-ML [23]	2,810	7,981
CiteSeer [30]	2,110	3,757
Pol. Blogs [1]	1,222	16,714

Attacks on the surrogate model

(1) Nettack、Nettack-In

(2) FGSM

(3) RND

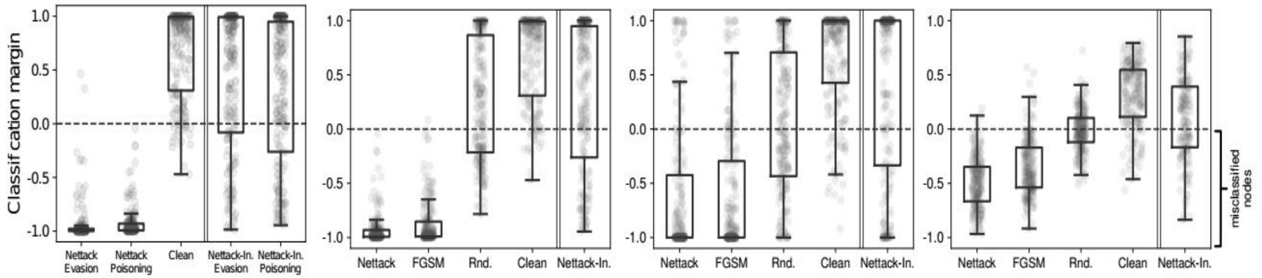
Transferability of attacks

(1) Evasion vs. Poisoning Attack

(2) Base model: GCN 、CLN and unsupervised model DeepWalk

(3) Limited Knowledge

Partial results



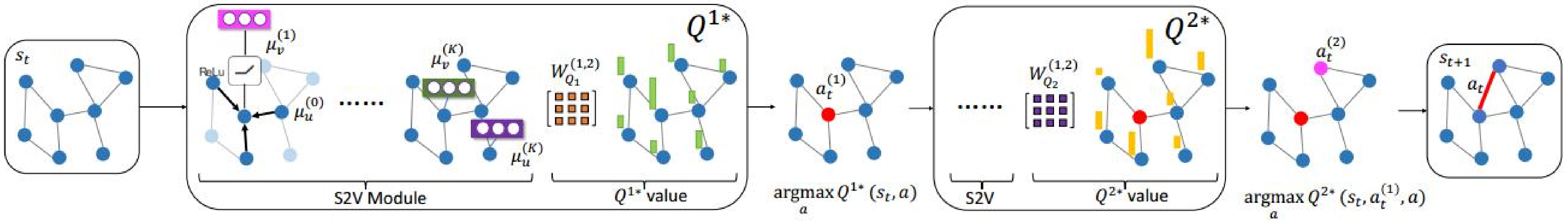
(a) Evasion vs. poisoning for GCN (b) Poisoning of GCN (c) Poisoning of Column Network (d) Poisoning of DeepWalk

Adversarial Attack on Graph Structured Data

Main Contribution:

- First propose a RL based attack, while only requiring prediction labels.
- Propose attack based on GA and gradient descent where additional prediction confidence or gradients are available.

Main Content:



Adversarial Attacks on Neural Networks for Graph Data

Attack model:

Attacker's goal

Attack a specific target node v_0 , aim to change v_0's prediction.

(1) Add edges

(2) Delete edges

$$\max_{\tilde{G}} \quad \mathbb{I}(f(\tilde{G},c) \neq y)$$
$$s.t. \quad \tilde{G} = g(f,(G,c,y))$$
$$\mathcal{I}(G,\tilde{G},c) = 1.$$

Equivalency indicator

$$\mathcal{I}(G,\tilde{G},c) = \mathbb{I}(f^*(G,c) = f^*(\tilde{G},c))$$

Small modifications

$$\mathcal{I}(G,\tilde{G},c) = \mathbb{I}(|(E-\tilde{E}) \cup (\tilde{E}-E)| < m) \cdot \mathbb{I}(\tilde{E} \subseteq \mathcal{N}(G,b))).$$

Adversarial Attacks on Neural Networks for Graph Data

Base attack model

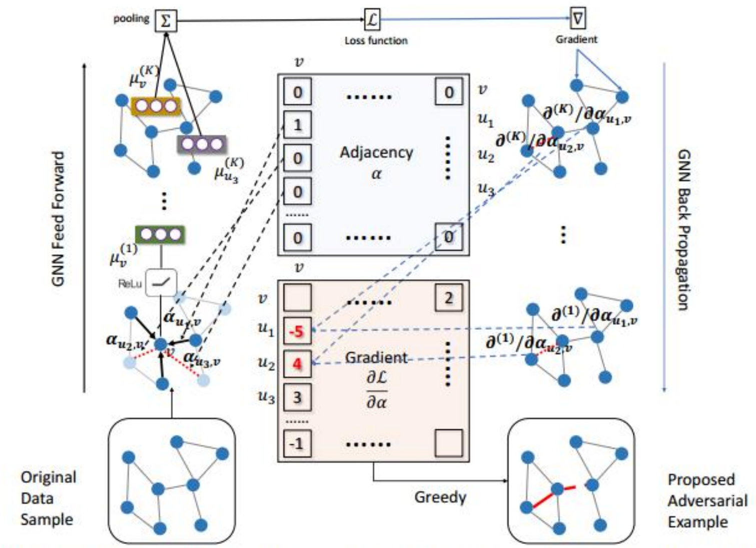


Figure 2. Illustration of graph structure gradient attack. This white-box attack adds/deletes the edges with maximum gradient (with respect to α) magnitudes.

Gradient-based white box attack

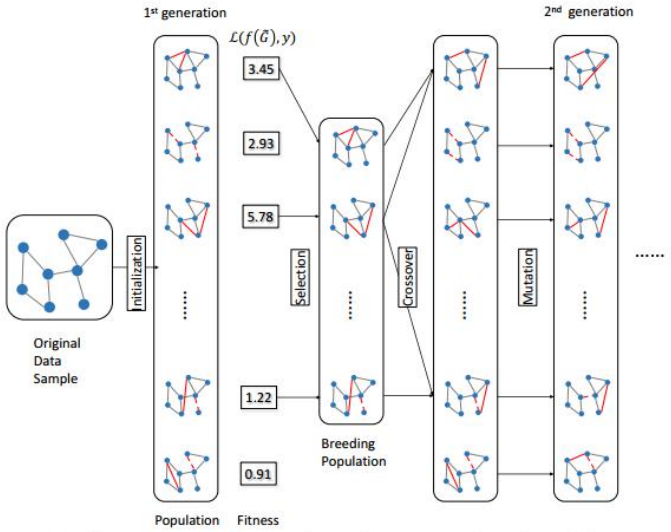


Figure 3. Illustration of attack using genetic algorithm. The population evolves with selection, crossover and mutation operations. Fitness is measured by the loss function.

Genetic algorithm

Adversarial Attacks on Neural Networks for Graph Data

Experiments

Dataset

15000 graphs generated with Erdos-Renyi random graph model

Table 3. Statistics of the graphs used for node classification.

Dataset	Nodes	Edges	Classes	Train/Test I/Test II
Citeseer	3,327	4,732	6	120/1,000/500
Cora	2,708	5,429	7	140/1,000/500
Pubmed	19,717	44,338	3	60/1,000/500
Finance	2,382,980	8,101,757	2	317,041/812/800

Partial results

Method	Citeseer	Cora	Pubmed	Finance
(unattacked)	71.60%	81.00%	79.90%	88.67%
RBA, *RandSampling*	67.60%	78.50%	79.00%	87.44%
WBA, *GradArgmax*	63.00%	71.30%	72.4%	86.33%
PBA-C, *GeneticAlg*	63.70%	71.20%	72.30%	85.96%
PBA-D, *RL-S2V*	62.70%	71.20%	72.80%	85.43%
Exhaust	62.50%	70.70%	71.80%	85.22%