Unlearnable Graph Examples

Zhaoning Yu, Tiancheng Zhou

Background

The volume of “free” data has been key to the current success of deep learning. However, it also raises privacy concerns about the unauthorized exploitation of personal data. It is thus crucial to develop methods to prevent unauthorized data exploitation

Goal: DNNs trained on unlearnable examples will have a performance equivalent to random guessing on normal test examples.

- Making an example unlearnable should not affect its quality for normal usage.
- The development of unlearnable examples should use the weaknesses of DNNs.

Research Problem

Recent studies have focused on the development of Unlearnable Examples which aim to make the original data unlearnable by adding imperceptible but deceptive perturbations to data samples.

We seek to answer the question of how to create unlearnable graphs, in order to enhance the robustness and security of GNN models against unauthorized graph exploitation.

Method

- A graph is represented by nodes (entities) and edges (relationships)
- They can capture complex relationships and are inherently non-Euclidean
- The invariance and equivariance properties make graph learning more challenging to work with
- Node information
- Topology information

Method - Node feature

Generate a noise for each feature each node

- Noise is learnable
- Noise is close to 0
- Introduce specific noise to each individual node based on node features
- Noise is zero mean

$$\text{Mean } (\mu) = \frac{1}{N}\sum_{i=1}^{N}(X \cdot W)_i$$

$$\text{Standard Deviation } (\sigma) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((X \cdot W)_i - \mu)^2}$$

$$\text{noise } (\epsilon) = \frac{X \cdot W - \mu}{\sigma}$$

Method - Topology information

Perturb only one edge for each graph (mask the edge)

- Initialize a learnable query vector
- Generate edge features by combining the features of the two nodes connected by the edge
- Calculate the similarity between each edge in a graph with the query vector
- Select the edge with the highest score
- Mask the edge

Experiments

Datasets: Mutagenicity, AIDS, PROTEINS

Baselines: Graph Convolutional Network

Optimizer: Adam

Learning rate: 0.01

Split: 0.8/0.2

Results

	Mutagenicity		AIDS		PROTEINS	
	Train	Test	Train	Test	Train	Test
GCN	0.7760	0.7539	0.8565	0.8633	0.7044	0.7136
GCN + (feature noise)	0.7611	0.5663	0.7841	0.1700	0.7067	0.6808
GCN + (structure noise)	0.7783	0.7145	0.8659	0.8267	0.7022	0.6901
GCN + (Feature & structure)	0.7580	0.4564	0.8200	0.2000	0.7156	0.6432

Thank you for your listening!