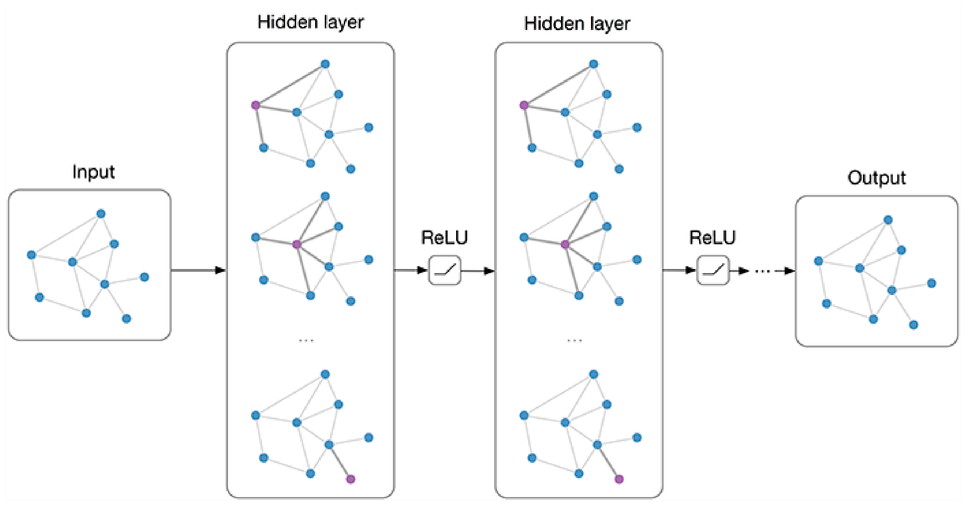
Methods and Optimizations for Node Classification

Yifan Ren (ivan-ren@sjtu.edu.cn)
ACM Class, Shanghai Jiao Tong University

Background

Node Classification

Graph Neural Network



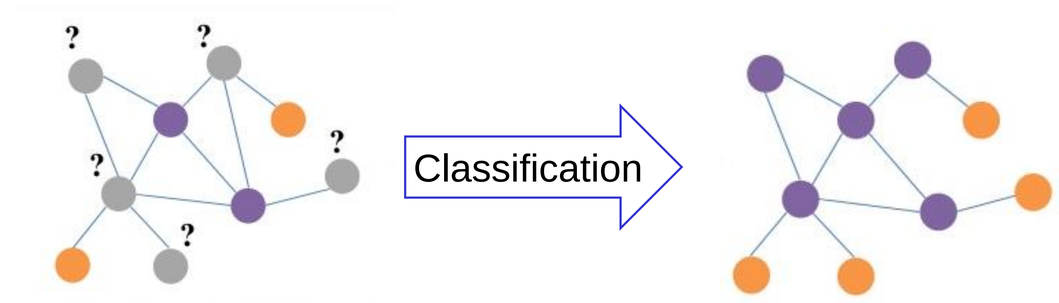
Terminologies

$H^{(l+1)} = \sigma\left(f\left(H^{l}\right)\right)$: The propagation rule using between two adjacent matrixes of activations in the l^{th} layer.
$\hat{A} = A + I_N$: The adjacency matrix of the undirected graph G with added self-connections.
α_{ij}: a shared attentional mechanism function between v_i and v_j.

Task

Original Graph

True Result



Predict unknown node labels by node features and the known labels of some specific nodes.

Challenges

Graph Convolutional Networks: has a poor scalability, and cannot simply increase the layers number.
Graph Attention Networks: the lack of training methods, sampling nodes also have room to improve.
Graph SAmple and aggreGatE: the number of sampling nodes increases exponentially with the number of layers, resulting in the poor performance of
the model on time per batch.
These models all have some defects, so we want to give some optimizations or mix them up to make them perform better.

Our Approach

Graph Convolutional Networks

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

Graph Attention Networks

$$\vec{h_i'} = \sigma(\frac{1}{K}\Sigma_{k=1}^{K}\Sigma_{j\in N(i)}\alpha_{ij}^{k}W^{k}\vec{h_j})$$

GraphSAGE

$$h_v^k = \sigma(W \cdot MEAN(\{h_v^{k-1}\} \cup \{h_u^{k-1}, \forall u \in N(v)\}))$$

GCN with Jump Knowledge Networks

All layers jump to the last layer and aggregate, so that nodes can adaptively choose the size of receptive domain.

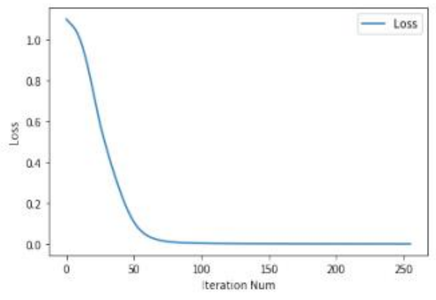


Figure 1: GCN

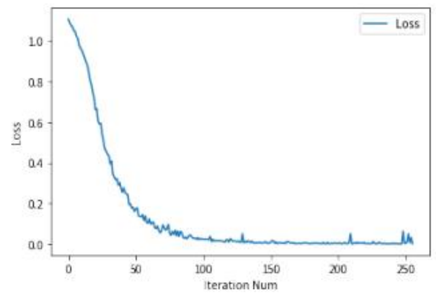


Figure 2: GAT

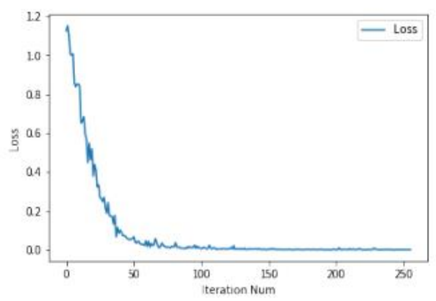


Figure 3: GraphSAGE

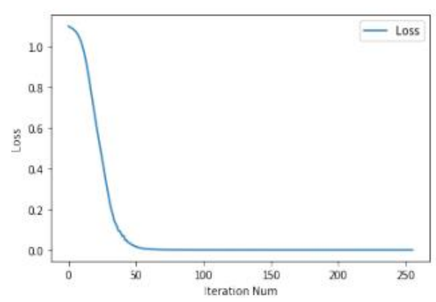


Figure 4: GCN_JKN

Average pooling and Max pooling

Using average pooling or max pooling to integrate these results.
$\text{logits}_{\text{pooling}} = \text{softmax}(\text{f}(\text{Logits}))$

Experiments

Setup

Dataset: Cora(2708 nodes), Citeseer(3327 nodes) and Pubmed(19717 nodes).
Metrics: to measure the correct rate between the predict labels with the true labels on each test mask.

Compared Models

- 3-layer GCN : uses the Laplacian matrix when propagation.
- 3-layer GAT : add attention mechanism.
- 3-layer GraphSAGE: learning of aggregator.
- 6-layer GCN with JKN optimization.

Results

Method	Cora	Citeseer	Pubmed
GCN	**77.3**	**63.1**	**75.4**
GAT	76.0	61.9	73.7
GraphSAGE	75.0	61.6	73.0
GCN with JKN	75.0	57.8	74.8
Average Pool	**80.3**	**65.4**	**76.3**
Max Pool	77.8	61.6	74.9