Graph Neural Networks

Script based on CS224W

Unity05

December 23, 2021

1 Message Passing Graph Neural Networks

1.1 Introduction

The idea behind message passing GNNs is **k - hop neighborhood aggregation**. One single GNN layer can be looked at as one single hop.

A GNN layer mainly consists of two parts: **Message Computation** and **Aggregation**.

1.1.1 Message Computation

Each node computes a message based on it's embedding in the previous layer.

$$m_u^{(l)} = \phi^{(l)} \left(h_u^{(l-1)} \right)$$

$m_u^{(l)}$	...	message of node u in layer l
$\phi^{(l)}$	...	message computation function of layer l
$h_u^{(l-1)}$	...	node u's embedding in layer l - 1

1.1.2 Aggregation

Node v's new embedding is computed by aggregating its own message as well as all of its neighbor node's messages.

$$h_v^{(l)} = \sigma \left(\square^{(l)} \left(\{ m_u^{(l)} \mid u \in N(v) \}, m_v^{(l)} \right) \right)$$

σ	...	nonlinear activation function
$h_v^{(l)}$	...	node v's new embedding in layer l - 1
$\square^{(l)}$	...	aggregation function of layer l
$m_u^{(l)}$	...	message of node u in layer l
$N(v)$	...	neighborhood of node v

1.2 Popular GNN Layers

1.2.1 Graph Convolutional Networks (GCN)

Message Computation
Embeddings are passed through a linear layer (transformation with weight matrix). Normalized by node degrees.

$$m_u^{(l)} = \left(W^{(l)} \cdot h_u^{(l-1)}\right)$$

Aggregation

$$h_v^{(l)} = \sigma\left(\sum_{u \in N(v) \cup \{v\}} \frac{1}{\sqrt{deg(u)} \cdot \sqrt{deg(v)}} \cdot m_u^{(l)}\right)$$

1.2.2 Graph Attention Networks (GAT)

Message Computation
No difference to GCN.

$$m_u^{(l)} = \left(W^{(l)} \cdot h_u^{(l-1)}\right)$$

Aggregation
Weighted summation of messages normalized by attention weights.

$$h_v^{(l)} = \sigma\left(\sum_{u \in N(v) \cup \{v\}} \alpha_{vu} \cdot m_u^{(l)}\right)$$

Computation of α_{vu} with attention mechanism a:

$$\alpha_{vu} = \frac{exp(e_{vu})}{\sum_{k \in N(v)} exp(e_{vk})}$$

$$e_{vu} = a\left(W^{(l)} h_u^{(l-1)}, W^{(l)} h_v^{(l-1)}\right)$$