

# Lecture 17: Graph Neural Networks

## CS480/680 Intro to Machine Learning

2023-3-14

Pascal Poupart  
David R. Cheriton School of Computer Science



# Graph Neural Networks

- Generalization of
  - Convolutional neural networks
  - Transformers
- Applications:
  - Recommender systems
  - Social networks, financial networks
  - Biology, Chemistry and Physics (proteins, molecules)
  - Combinatorial optimization

# Embeddings

- Neural network that computes embeddings for nodes (and edges) in a graph by passing messages along the edges of the graph
- The embedding of a node captures information about its context

# Message Passing

- Graph: Nodes ( $V = \{v\}$ ) and edges ( $E = \{e\}$ )
  - Initial node embedding:  $h_0(v)$
  - Message passing:  
$$h_t(v) \leftarrow \text{combine}(h_{t-1}(v), \text{aggregate}(\{h_{t-1}(u) \mid u \in \text{neighbors}(v)\}))$$

# Convolutional Neural Network

- CNN that preserves size is a special type of GNN
  - Initial node embedding:  $h_0(v) = \text{pixel intensities}$

# Convolutional Neural Network

- CNN that preserves size is a special type of GNN

- Initial node embedding:  $h_0(v) = \text{pixel intensities}$

- $n \times n$  convolutional layer (stride=1, padding=same):

$$m_{ij} \leftarrow \text{aggregate}(\{h_{t-1}(v_{i'j'}) \mid i' \neq i \text{ or } j' \neq j\}) = \sum_{i' \neq i \text{ or } j' \neq j} w_{i'j'} h_{t-1}(v_{i'j'})$$

$$h_t(v_{ij}) \leftarrow \text{combine}(h_{t-1}(v_{ij}), m_{ij}) = \sigma(w_{ij} h_{t-1}(v_{ij}) + m_{ij})$$

- $n \times n$  pooling layer (stride=1, padding=same):

$$m_{ij} \leftarrow \text{aggregate}(\{h_{t-1}(v_{i'j'}) \mid i' \neq i \text{ or } j' \neq j\}) = \max_{i' \neq i \text{ or } j' \neq j} h_{t-1}(v_{i'j'})$$

$$h_t(v_{ij}) \leftarrow \text{combine}(h_{t-1}(v_{ij}), m_{ij}) = \max\{h_{t-1}(v_{ij}), m_{ij}\}$$

# Graph Convolutional Neural Network

- GNN that does the same operations as a CNN on arbitrary graphs
  - Initial node embedding:  $h_0(v)$
  - Message passing:  $H_t \leftarrow \sigma \left( \hat{D}^{-1} \hat{A} H_{t-1} W \right)$

# Transformer

- Transformer is a special type of fully connected GNN
  - Initial node embedding:  $h_0(v) = \text{token embedding} + \text{positional encoding}$
  - Message passing:
$$m_v \leftarrow \text{aggregate}(\{h_{t-1}(u) \mid u \in \text{neighbors}(v)\}) = \sum_u a_{vu} W_V h_{t-1}(u)$$
$$h_t(v) \leftarrow \text{combine}(h_{t-1}(v), m_v) = \text{norm} \left( \text{ff}(\text{norm}(a_{vv} W_V h_{t-1}(v) + m_v)) \right)$$

where  $a_{vu}$  is the attention weight of  $u$  with respect to  $v$

$$a_{vu} = \frac{\exp \left( \text{sim} \left( W_Q h(v), W_K h(u) \right) \right)}{\sum_{u'} \exp \left( \text{sim} \left( W_Q h(v), W_K h(u') \right) \right)}$$



# Recommender System

- Movie recommendation (edge completion)
- Bipartite graph:
  - Nodes: users and movies
  - Edges:  $e_{uv}$  = user  $u$  watched movie  $v$

# Recommender System

- Movie recommendation (edge completion)
- Bipartite graph:
  - Nodes: users and movies
  - Edges:  $e_{uv}$  = user  $u$  watched movie  $v$
- Messages:  $h_t(v) \leftarrow \text{comb}_\phi(h_{t-1}(v), \text{agg}_\theta(\{h_{t-1}(u) | u \in nb(v)\}))$ 
  - Example:  $h_t(v) \leftarrow \sigma(\phi h_{t-1}(v) + \sum_u \theta h_{t-1}(u))$
- Edge prediction:  $P(e_{uv}) = f_W(u, v)$ 
  - Example:  $P(e_{uv}) = \sigma(h(u)^T W h(v))$
- Train:  $\max_{W, \theta, \phi} \sum_{e_{uv} \in E} \log P(e_{uv})$

# Boolean Satisfiability

- Example:  $(v_1 \vee \sim v_2 \vee v_3) \wedge (v_2 \vee \sim v_3 \vee \sim v_4)$
- Graph:
  - Nodes: literals  $V = \{v\}$  and clauses  $U = \{u\}$
  - Edges:  $e_{vu}$  = literal  $v$  appears in clause  $u$   
 $e_{v\sim v}$  = special edge from literal  $v$  to its complement  $\sim v$

# Boolean Satisfiability

- Example:  $(v_1 \vee \sim v_2 \vee v_3) \wedge (v_2 \vee \sim v_3 \vee \sim v_4)$
- Graph:
  - Nodes: literals  $V = \{v\}$  and clauses  $U = \{u\}$
  - Edges:  $e_{vu}$  = literal  $v$  appears in clause  $u$   
 $e_{v\sim v}$  = special edge from literal  $v$  to its complement  $\sim v$
- Messages:  $h_t(v) \leftarrow \text{comb}_\phi(h_{t-1}(v), \text{agg}_\theta(\{h_{t-1}(u) \mid u \in \text{nb}(v)\}))$ 
  - Example:  $h_t(v) \leftarrow \sigma(\phi h_{t-1}(v) + \sum_u \theta h_{t-1}(u))$
- Clause classification:  $P(u) = f_w(u)$ 
  - Example:  $P(u) = \sigma(w^T h(u))$
- Train:
  - Satisfiable:  $\max_{w, \theta, \phi} \sum_{u \in \text{posClauses}} \log P(u)$
  - Unsatisfiable:  $\min_{w, \theta, \phi} \sum_{u \in \text{posClauses}} \log P(u)$