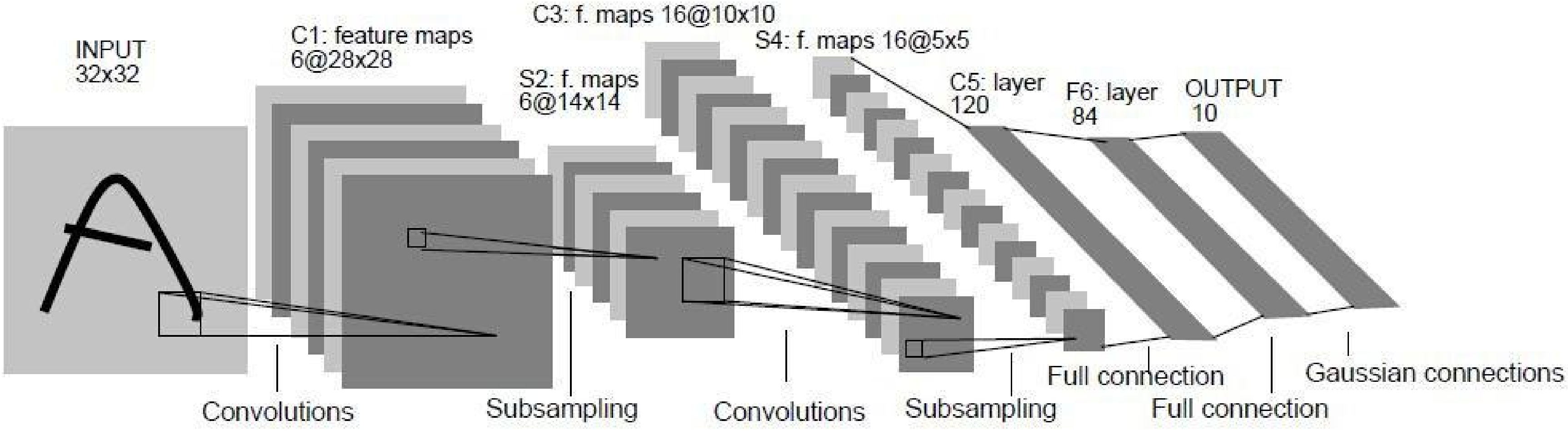
GNN

- Graph Neural Networks

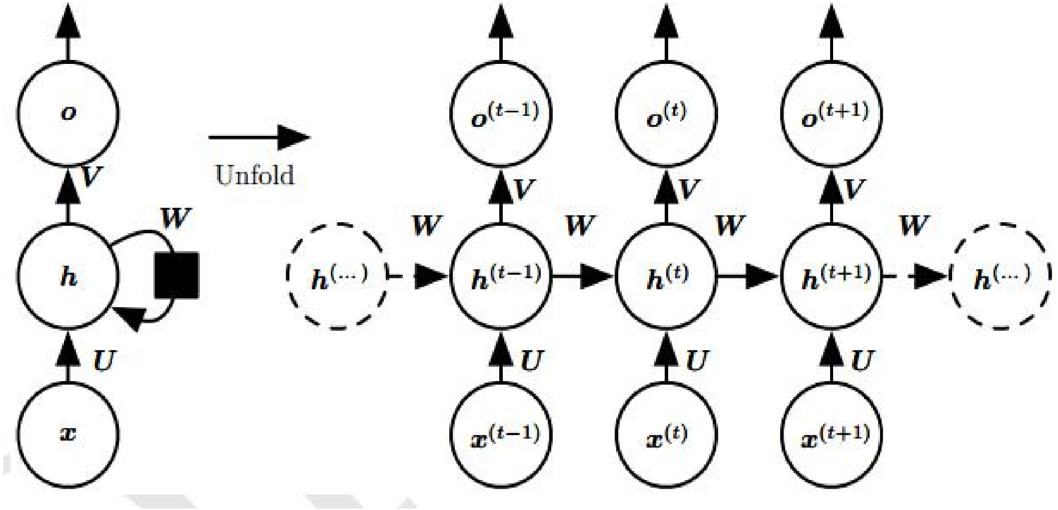
Overview

- GNN
- GCN（with GNN Report）
- DGL library

From CNN&RNN to GNN

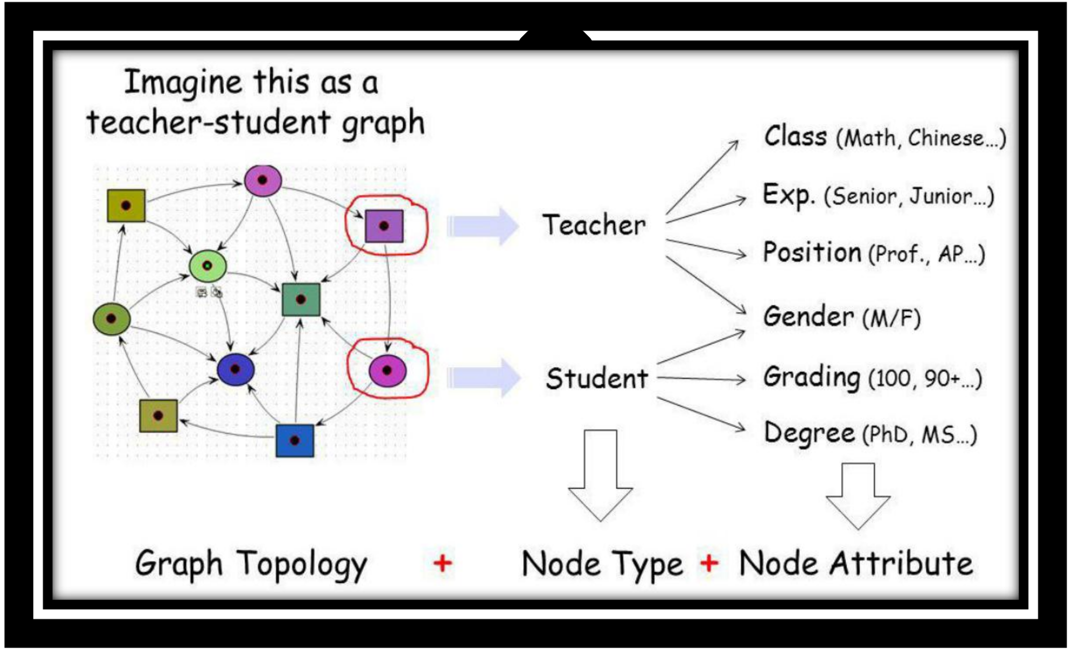


From CNN&RNN to GNN



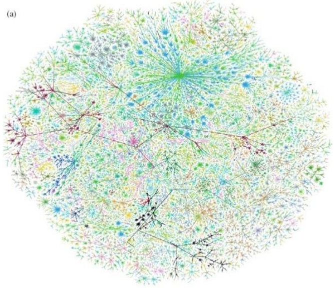
Graphs: A Universal Language

- Graphs are a general language for describing and modeling complex systems

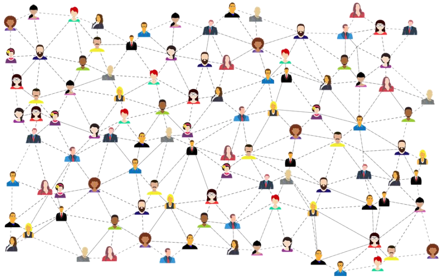


Graph!

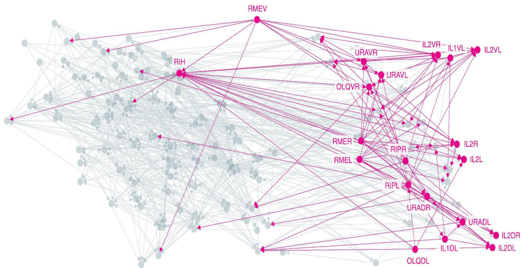
Graph-structured Data Are Ubiquitous



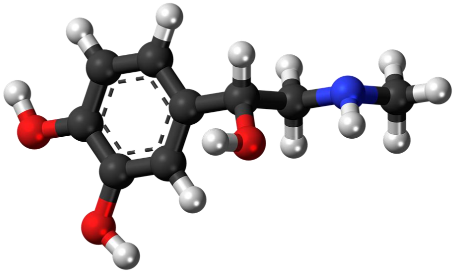
Internet



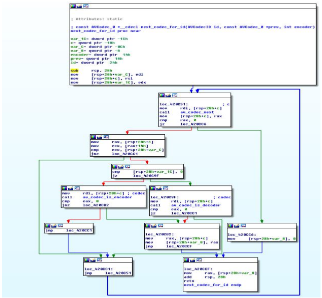
Social networks



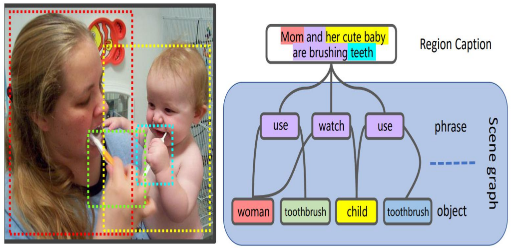
Networks of neurons



Biomedical graphs



Program graphs



Why Graphs? Why Now?

- Universal language for describing complex data
  - Networks/graphs from science, nature, and technology are more similar than one would expect
- Shared vocabulary between fields
  - Computer Science, Social science, Physics, Biology, Economics
- Data availability (+ computational challenges)
  - Social/internet, text, logic, program, bio, health, and medical
- Impact
  - Social networking, Social media, Drug design, Event detection, Natural language processing, Computer vision, and Logic reasoning

Machine(Deep) Learning with Graphs

Classical ML tasks in graphs:

- Node classification
  - Predict a type of a given node
- Link prediction
  - Predict whether two nodes are linked
- Community detection
  - Identify densely linked clusters of nodes
- Graph similarity
  - How similar are two (sub)graphs

Recent ML tasks in graphs:

- Graph classification
  - Predict a type of a given graph
- Graph generation
  - Generate graphs from learned distribution
- Graph structure learning
  - Identify densely linked clusters of nodes
- Graph-to-XXX learning
  - Graph Inputs – XXX out

Graph Representation Learning (GNNs)

- Graph Neural Networks (GNNs) extends the well known CNN and RNN on graphs, from Euclidean data to Graphs and Manifolds

- RNN-based GNNs:
  - Graph neural networks (Scarselli et al., 2009)
  - Gated graph sequence neural networks (GGS-NNs) (Li et al., ICLR 2016)

- CNN-based GNNs:
  - Graph Convolutional Networks (GCN) (Kipf & Welling, ICLR 2017)

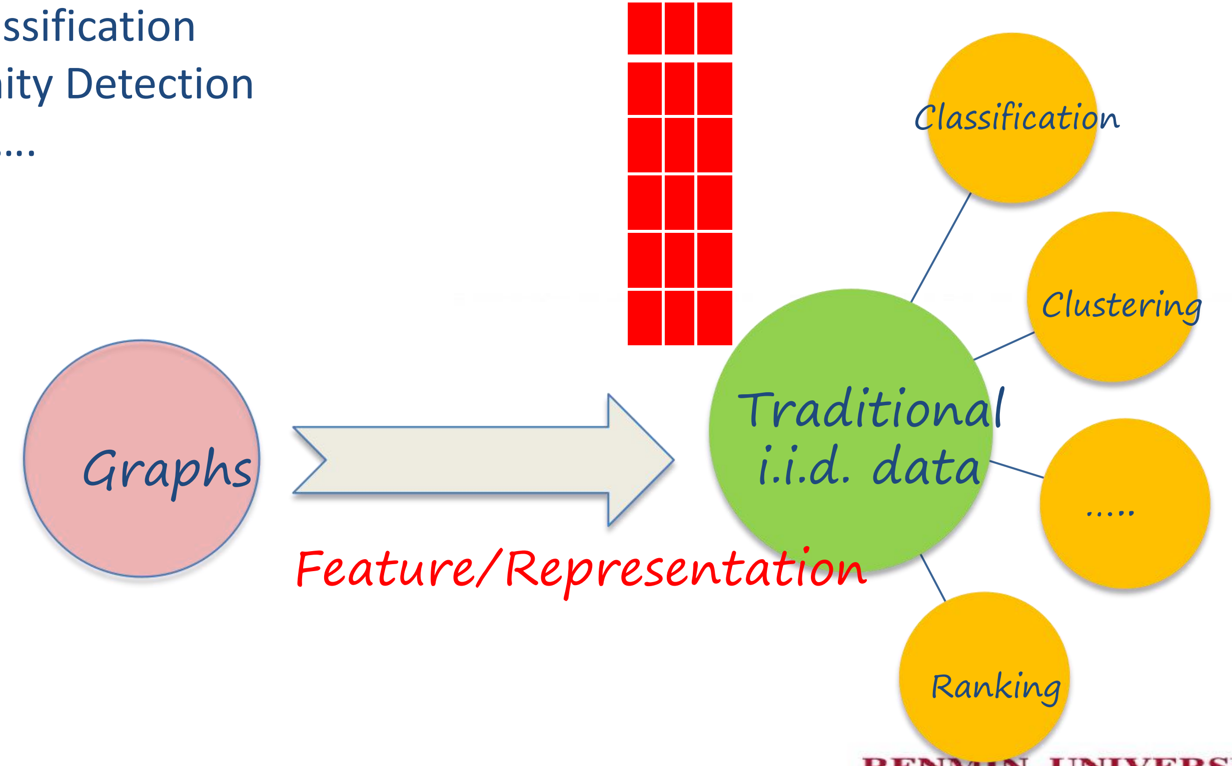
– Message Passing-based GNNs:
  - GraphSAGE (Hamilton & Ying & Leskovec, NIPS 2017)
  - Graph Attention Networks (GAT) (Velickovic et al., ICLR 2018)
  - MPNN (Gilmer et al., ICML 2017)

ML on Graphs

Numerous real-world problems can be summarized as a set of tasks on graphs

- Link prediction
- Node Classification
- Community Detection
- Ranking ....

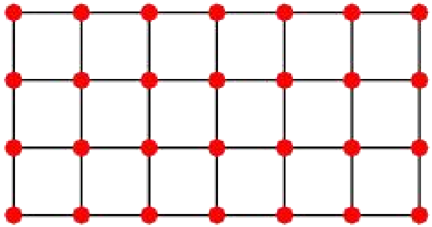
ML solutions

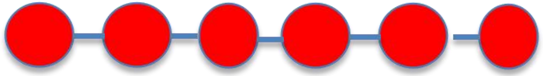


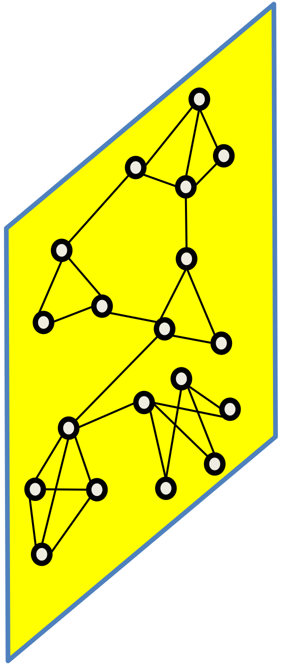
Deep Learning Meets Graphs: Challenges

Traditional DL is designed for simple grids or sequences

- CNNs for fixed-size images/grids
- RNNs for text/sequences



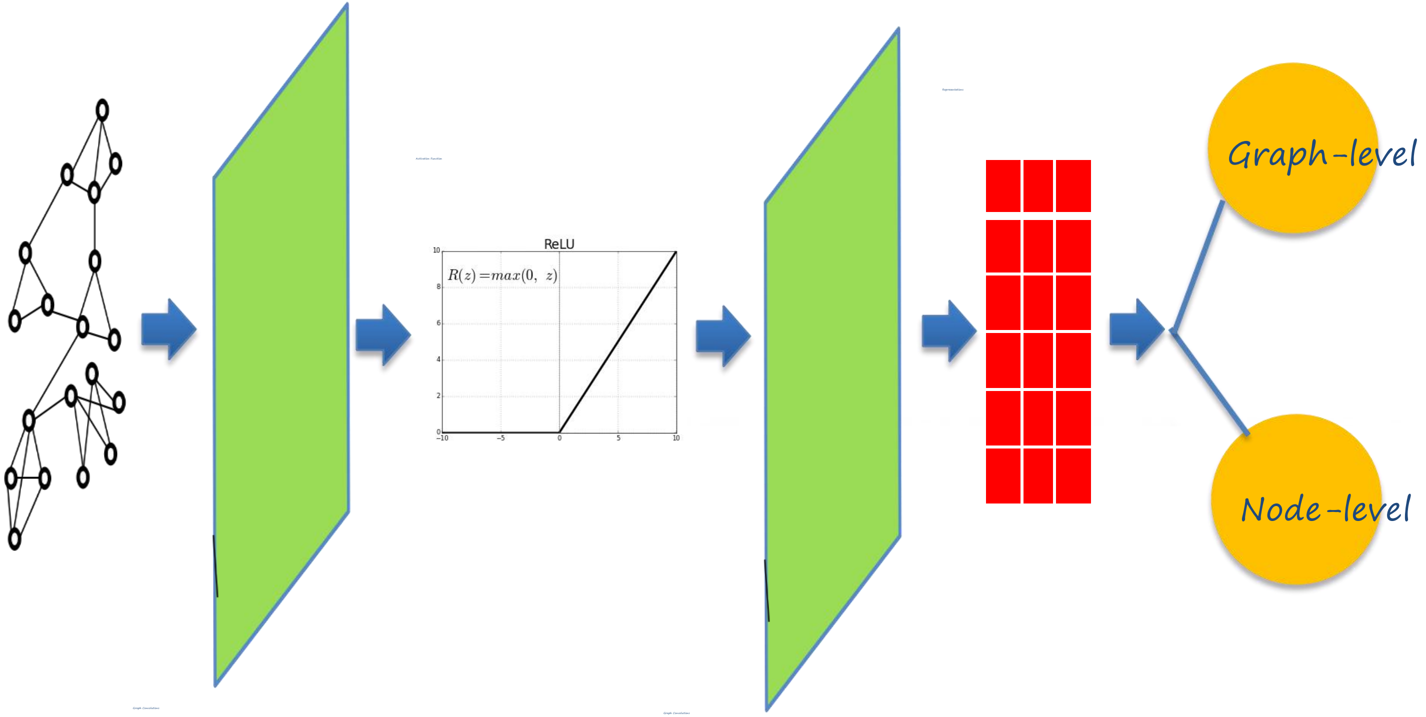




But nodes on graphs have different connections

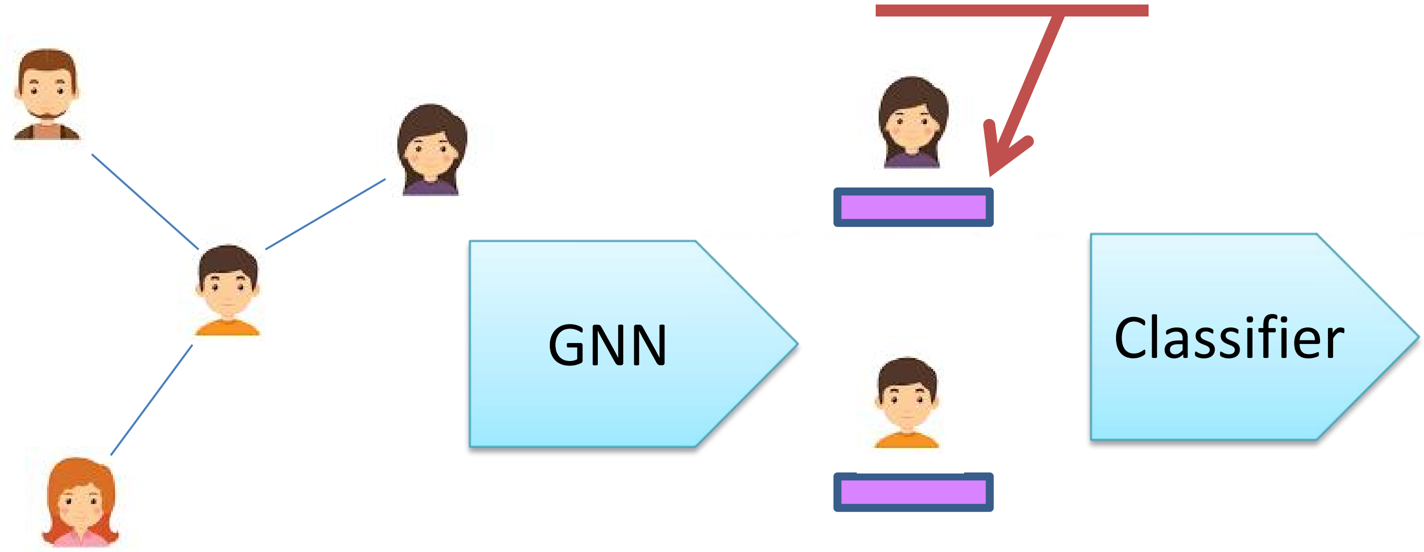
- Arbitrary neighbor size
- Complex topological structure
- No fixed node ordering

Graph Neural Networks



How GNN solves this problem?

- Graph Neural Network (GNN): learn embedding on nodes/edges/graphs

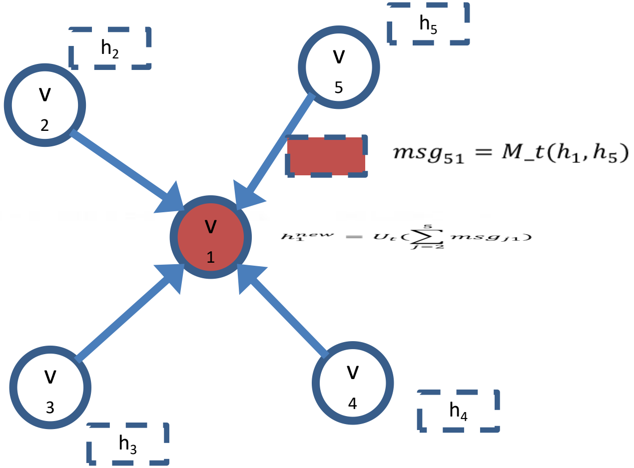


How GNN learns embeddings

- GNN is based on message passing

$$m_v^{t+1} = \boxed{\sum_{w \in N(v)}} \boxed{M_t(h_v^t, h_w^t, e_{vw})} \qquad (1)$$

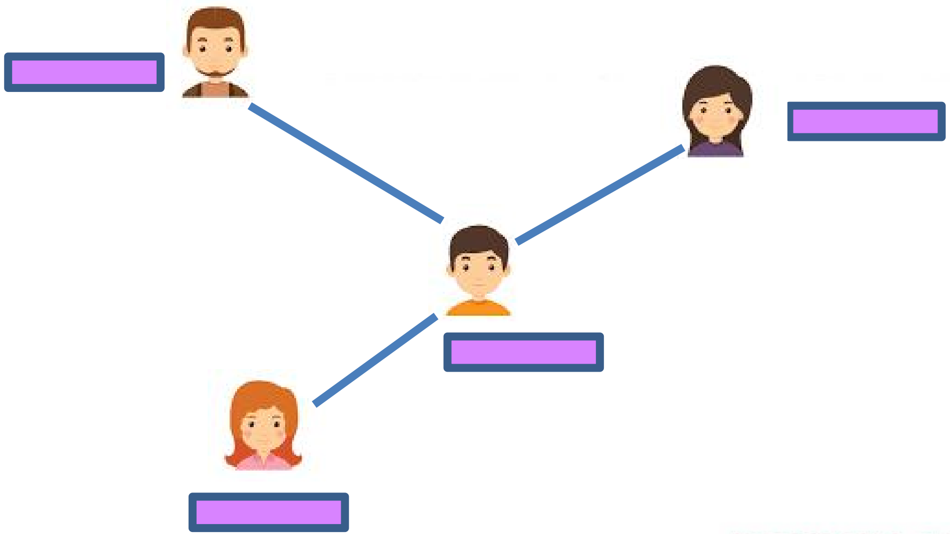
$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \qquad (2)$$



How GNN is trained

- Step 0:

Associate each node / edge with input features (an embedding vector)



How GNN is trained (cont'd)

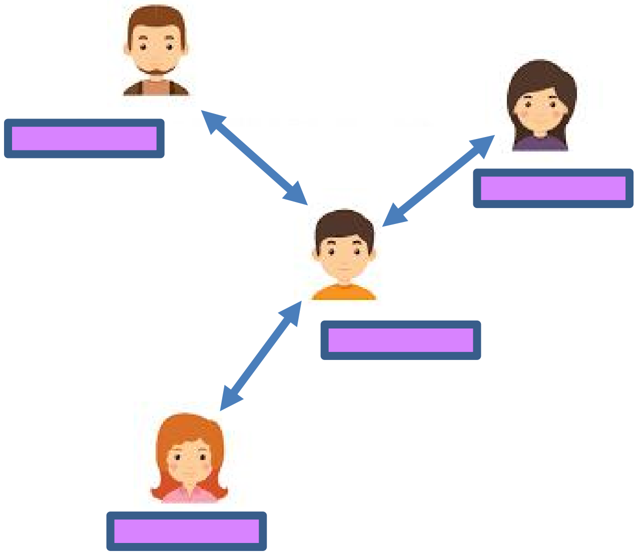
- Step 1:

Perform message passing for a fixed number (a pre-defined constant) of iterations

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

Both are parameterized neural network modules



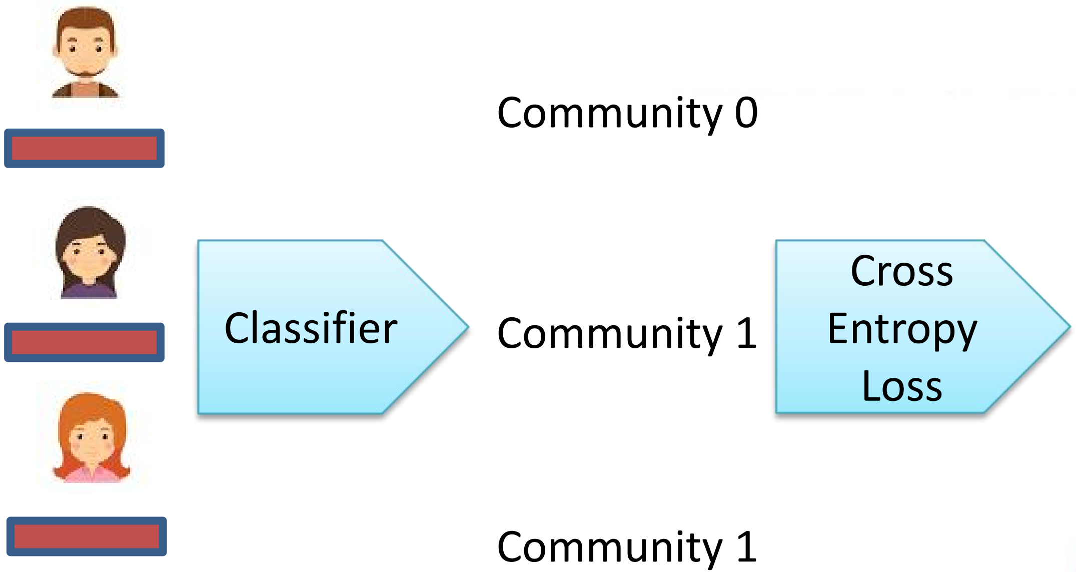
How GNN is trained (cont'd)

- Step 2:

Use the GNN-outputted embedding for training tasks to calculate loss

- Step 3:

Minimize loss with backward propagation



GCN

Video report from bilibili

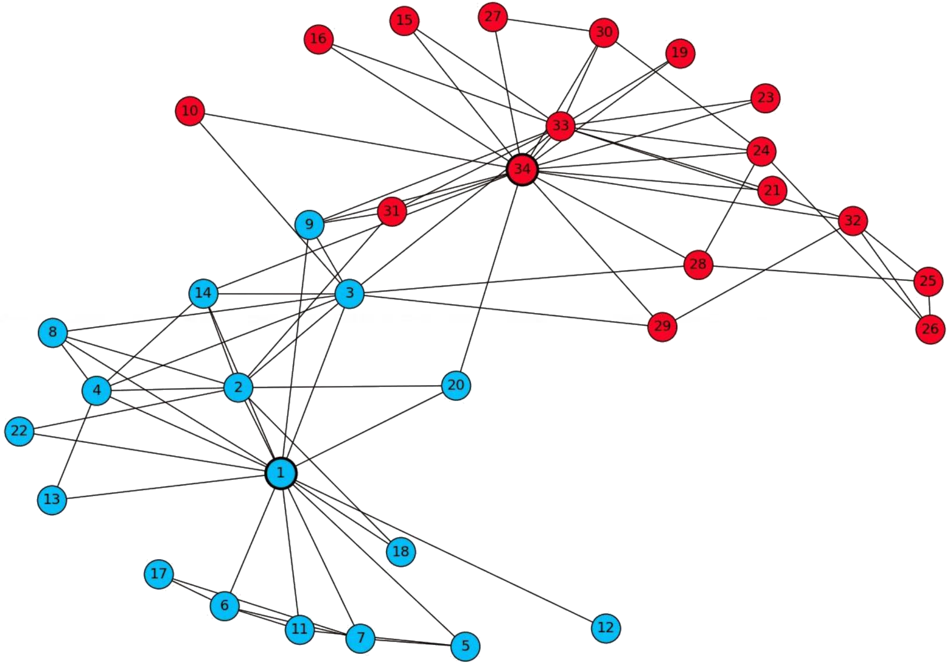
Overview of DGL

Deep Graph Library (DGL) is a Python package built for easy implementation of graph neural network model family, on top of existing DL frameworks (e.g. PyTorch, MXNet, Gluon etc.). DGL reduces the implementation of graph neural networks into declaring a set of functions. In addition, DGL provides:

- Versatile controls over message passing, ranging from low-level operations such as sending along selected edges and receiving on specific nodes, to high-level control such as graph-wide feature updates.
- Transparent speed optimization with automatic batching of computations and sparse matrix multiplication.
- Seamless integration with existing deep learning frameworks.
- Easy and friendly interfaces for node/edge feature access and graph structure manipulation.
- Good scalability to graphs with tens of millions of vertices.

Zachery's Karate Club

- Nodes: club members
  - Node 1 is club instructor
  - Node 34 is club president
- Edges: interaction outside club between members
- The task: predict for each member which community (instructor or president) he belongs to



Code

。 。 。