DGL

**DGL表示点的方式：**

1. 节点 id
2. 边 (id1,id2）id1->id2 有方向 有边id
3. 多个节点 一维的整型张量（如，PyTorch的Tensor类，TensorFlow的Tensor类或MXNet的ndarray类）

**节点和边的特征**

1. 特征类型

g.ndata['x'] = th.ones(g.num\_nodes(), 3) *# 长度为3的节点特征*

g.edata['x'] = th.ones(g.num\_edges(), dtype=th.int32) *# 标量整型特征*

g.ndata['y'] = th.randn(g.num\_nodes(), 5)

1. 获取特征

>>>g.ndata['x'][1] *# 获取节点1的特征*

tensor([1., 1., 1.])

**>>>** g.edata['x'][th.tensor([0, 3])] *# 获取边0和3的特征*

tensor([1, 1], dtype=torch.int32)

常用特征:

# 边 0->1, 0->2, 0->3, 1->3

edges = th.tensor([0, 0, 0, 1]), th.tensor([1, 2, 3, 3])

weights = th.tensor([0.1, 0.6, 0.9, 0.7]) # 每条边的权重

g = dgl.graph(edges)

g.edata['w'] = weights # 将其命名为 'w

Graph(num\_nodes=4, num\_edges=4,

ndata\_schemes={}

edata\_schemes={'w' : Scheme(shape=(,), dtype=torch.float32)})

**从外部源创建图**

1. 从用于图和稀疏矩阵的外部Python库（NetworkX 和 SciPy）创建而来
2. 从磁盘加载图数据。

**异构图**

在DGL中，一个异构图由一系列子图构成，一个子图对应一种关系。每个关系由一个字符串三元组 定义 (源节点类型, 边类型, 目标节点类型) 。由于这里的关系定义消除了边类型的歧义，DGL称它们为规范边类型

消息传递范式