1.3 Node and Edge Features

(中文版)

The nodes and edges of a `DGLGraph` can have several user-defined named features for storing graph-specific properties of the nodes and edges. These features can be accessed via the `ndata` and `edata` interface. For example, the following code creates two node features (named `'x'` and `'y'` in line 8 and 15) and one edge feature (named `'x'` in line 9).

```
1   >>> import dgl
2   >>> import torch as th
3   >>> g = dgl.graph(([0, 0, 1, 5], [1, 2, 2, 0])) # 6 nodes, 4 edges
4   >>> g
5   Graph(num_nodes=6, num_edges=4,
6         ndata_schemes={}
7         edata_schemes={})
8   >>> g.ndata['x'] = th.ones(g.num_nodes(), 3)              # node feature of length 3
9   >>> g.edata['x'] = th.ones(g.num_edges(), dtype=th.int32)  # scalar integer feature
10  >>> g
11  Graph(num_nodes=6, num_edges=4,
12        ndata_schemes={'x' : Scheme(shape=(3,), dtype=torch.float32)}
13        edata_schemes={'x' : Scheme(shape=(,), dtype=torch.int32)})
14  >>> # different names can have different shapes
15  >>> g.ndata['y'] = th.randn(g.num_nodes(), 5)
16  >>> g.ndata['x'][1]                  # get node 1's feature
17  tensor([1., 1., 1.])
18  >>> g.edata['x'][th.tensor([0, 3])]  # get features of edge 0 and 3
19      tensor([1, 1], dtype=torch.int32)
```

Important facts about the `ndata` / `edata` interface:

- Only features of numerical types (e.g., float, double, and int) are allowed. They can be scalars, vectors or multi-dimensional tensors.
- Each node feature has a unique name and each edge feature has a unique name. The features of nodes and edges can have the same name. (e.g., 'x' in the above example).
- A feature is created via tensor assignment, which assigns a feature to each node/edge in the graph. The leading dimension of that tensor must be equal to the number of nodes/edges in the graph. You cannot assign a feature to a subset of the nodes/edges in the graph.
- Features of the same name must have the same dimensionality and data type.
- The feature tensor is in row-major layout – each row-slice stores the feature of one node or edge (e.g., see lines 16 and 18 in the above example).

For weighted graphs, one can store the weights as an edge feature as below.

```
>>> # edges 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])  # weight of each edge
>>> g = dgl.graph(edges)
>>> g.edata['w'] = weights  # give it a name 'w'
>>> g
Graph(num_nodes=4, num_edges=4,
      ndata_schemes={}
      edata_schemes={'w' : Scheme(shape=(,), dtype=torch.float32)})
```

See APIs: `ndata`, `edata`.