

1.4 从外部源创建图

(English Version)

可以从外部来源构造一个 `DGLGraph` 对象，包括：

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

本节不涉及通过转换其他图来生成图的函数，相关概述请阅读API参考手册。

从外部库创建图

以下代码片段为从SciPy稀疏矩阵和NetworkX图创建DGL图的示例。

```
>>> import dgl
>>> import torch as th
>>> import scipy.sparse as sp
>>> spmat = sp.rand(100, 100, density=0.05) # 5%非零项
>>> dgl.from_scipy(spmat) # 来自SciPy
Graph(num_nodes=100, num_edges=500,
      ndata_schemes={},
      edata_schemes={})

>>> import networkx as nx
>>> nx_g = nx.path_graph(5) # 一条链路0-1-2-3-4
>>> dgl.from_networkx(nx_g) # 来自NetworkX
Graph(num_nodes=5, num_edges=8,
      ndata_schemes={},
      edata_schemes={})
```

注意，当使用 `nx.path_graph(5)` 进行创建时，`DGLGraph` 对象有8条边，而非4条。这是由于 `nx.path_graph(5)` 构建了一个无向的NetworkX图 `networkx.Graph`，而 `DGLGraph` 的边总是有向的。所以当将无向的NetworkX图转换为 `DGLGraph` 对象时，DGL会在内部将1条无向边转换为2条有向边。使用有向的NetworkX图 `networkx.DiGraph` 可避免该行为。

```
>>> nxg = nx.DiGraph([(2, 1), (1, 2), (2, 3), (0, 0)])
>>> dgl.from_networkx(nxg)
Graph(num_nodes=4, num_edges=4,
      ndata_schemes={},
      edata_schemes={})
```

! Note

DGL在内部将SciPy矩阵和NetworkX图转换为张量来创建图。因此，这些构建方法并不适用于重视性能的场景。

相关API: `dgl.from_scipy()`、`dgl.from_networkx()`。

从磁盘加载图

有多种文件格式可储存图，所以这里难以枚举所有选项。本节仅给出一些常见格式的一般情况。

逗号分隔值 (CSV)

CSV是一种常见的格式，以表格格式储存节点、边及其特征：

nodes.csv

age, title

43, 1

23, 3

...

edges.csv

src, dst, weight

0, 1, 0.4

0, 3, 0.9

...

许多知名Python库(如Pandas)可以将该类型数据加载到python对象(如 `numpy.ndarray`)中，进而使用这些对象来构建DGLGraph对象。如果后端框架也提供了从磁盘中保存或加载张量的工具(如 `torch.save()`，`torch.load()`)，可以遵循相同的原理来构建图。

另见：[从成对的边 CSV 文件中加载 Karate Club Network 的教程。](#)

JSON/GML 格式

如果对速度不太关注的话，读者可以使用NetworkX提供的工具来解析 [各种数据格式](#)，DGL可以间接地从这些来源创建图。

DGL 二进制格式

DGL提供了API以从磁盘中加载或向磁盘里保存二进制格式的图。除了图结构，API也能处理特征数据和图级别的标签数据。 DGL也支持直接从S3/HDFS中加载或向S3/HDFS保存图。参考手册提供了该用法的更多细节。

相关API： `dg1.save_graphs()`、 `dg1.load_graphs()`。