

## 第5章：训练图神经网络

(English Version)

### 概述

本章通过使用 [第2章：消息传递范式](#) 中介绍的消息传递方法和 [第3章：构建图神经网络 \(GNN\) 模块](#) 中介绍的图神经网络模块，讲解了如何对小规模的图数据进行节点分类、边分类、链接预测和整图分类的图神经网络的训练。

本章假设用户的图以及所有的节点和边特征都能存进GPU。对于无法全部载入的情况，请参考用户指南的 [第6章：在大图上的随机（批次）训练](#)。

后续章节的内容均假设用户已经准备好了图和节点/边的特征数据。如果用户希望使用DGL提供的数据集或其他兼容 `DGLDataset` 的数据(如 [第4章：图数据处理管道](#) 所述)，可以使用类似以下代码的方法获取单个图数据集的图数据。

```
import dgl

dataset = dgl.data.CiteseerGraphDataset()
graph = dataset[0]
```

注意: 本章代码使用PyTorch作为DGL的后端框架。

### 异构图训练的样例数据

有时用户会想在异构图上进行图神经网络的训练。本章会以下面代码所创建的一个异构图为例，来演示如何进行节点分类、边分类和链接预测的训练。

这个 `hetero_graph` 异构图有以下这些边的类型：

- `('user', 'follow', 'user')`
- `('user', 'followed-by', 'user')`
- `('user', 'click', 'item')`
- `('item', 'clicked-by', 'user')`
- `('user', 'dislike', 'item')`
- `('item', 'disliked-by', 'user')`

```

import numpy as np
import torch

n_users = 1000
n_items = 500
n_follows = 3000
n_clicks = 5000
n_dislikes = 500
n_hetero_features = 10
n_user_classes = 5
n_max_clicks = 10

follow_src = np.random.randint(0, n_users, n_follows)
follow_dst = np.random.randint(0, n_users, n_follows)
click_src = np.random.randint(0, n_users, n_clicks)
click_dst = np.random.randint(0, n_items, n_clicks)
dislike_src = np.random.randint(0, n_users, n_dislikes)
dislike_dst = np.random.randint(0, n_items, n_dislikes)

hetero_graph = dgl.heterograph({
    ('user', 'follow', 'user'): (follow_src, follow_dst),
    ('user', 'followed-by', 'user'): (follow_dst, follow_src),
    ('user', 'click', 'item'): (click_src, click_dst),
    ('item', 'clicked-by', 'user'): (click_dst, click_src),
    ('user', 'dislike', 'item'): (dislike_src, dislike_dst),
    ('item', 'disliked-by', 'user'): (dislike_dst, dislike_src)})

hetero_graph.nodes['user'].data['feature'] = torch.randn(n_users, n_hetero_features)
hetero_graph.nodes['item'].data['feature'] = torch.randn(n_items, n_hetero_features)
hetero_graph.nodes['user'].data['label'] = torch.randint(0, n_user_classes, (n_users,))
hetero_graph.edges['click'].data['label'] = torch.randint(1, n_max_clicks, (n_clicks,)).float()
# 在user类型的节点和click类型的边上随机生成训练集的掩码
hetero_graph.nodes['user'].data['train_mask'] = torch.zeros(n_users,
dtype=torch.bool).bernoulli(0.6)
hetero_graph.edges['click'].data['train_mask'] = torch.zeros(n_clicks,
dtype=torch.bool).bernoulli(0.6)

```

## 本章路线图

本章共有四节，每节对应一种图学习任务。

- [5.1 节点分类/回归](#)
- [5.2 边分类/回归](#)
- [5.3 链接预测](#)
- [5.4 整图分类](#)
- [guide\\_cn-training-graph-eweight](#)