

## 2.5 在异构图上进行消息传递

(English Version)

异构图（参考用户指南 [1.5 异构图](#)）是包含不同类型的节点和边的图。不同类型的节点和边常常具有不同类型的属性。这些属性旨在刻画每一种节点和边的特征。在使用图神经网络时，根据其复杂性，可能需要使用不同维度的表示来对不同类型的节点和边进行建模。

异构图上的消息传递可以分为两个部分：

- [1. 对每个关系计算和聚合消息。](#)
- [2. 对每个结点聚合来自不同关系的消息。](#)

在DGL中，对异构图进行消息传递的接口是 `multi_update_all()`。 `multi_update_all()` 接受一个字典。这个字典的每一个键值对里，键是一种关系，值是这种关系对应 `update_all()` 的参数。 `multi_update_all()` 还接受一个字符串来表示跨类型整合函数，来指定整合不同关系聚合结果的方式。这个整合方式可以是 `sum`、`min`、`max`、`mean` 和 `stack` 中的一个。以下是一个例子：

```
import dgl.function as fn

for c_etype in G.canonical_etypes:
    srctype, etype, dsttype = c_etype
    Wh = self.weight[etype](feat_dict[srctype])
    # 把它存在图中用来做消息传递
    G.nodes[srctype].data['Wh_%s' % etype] = Wh
    # 指定每个关系的消息传递函数: (message_func, reduce_func).
    # 注意结果保存在同一个目标特征“h”，说明聚合是逐类进行的。
    funcs[etype] = (fn.copy_u('Wh_%s' % etype, 'm'), fn.mean('m', 'h'))
# 将每个类型消息聚合的结果相加。
G.multi_update_all(funcs, 'sum')
# 返回更新过的节点特征字典
return {ntype : G.nodes[ntype].data['h'] for ntype in G.ntypes}
```