BIGNNITION: 一个GNN的快速原型化框架

1 Problems：现阶段，实现GNN模型非常繁琐，需要强大的ML专业知识，不利于非专业人员使用GNN完成相应的工作；现有的GNN系统，要么繁琐，要么不支持非标准的model。

2 Idea：基于TensorFlow实现一个GNN框架，user可以通过yaml文件定义GNN model，以隐藏GNN的张量操作和数学公式；提供极大地灵活性，适用于任何GNN model（包括非标准的）。

3 贡献：通过提出一个MSMP（多级消息传递）图抽象实现了idea，我的理解是，这个MSMP是将GNN中的NN部分，通过Entity之间的消息传递去定义一个NN，定义出每个消息传递的消息聚合方式、更新函数即可；此外，还提供了高效的debug功能，结合交互式调试可视化和高级检错机制。

4 框架实现：包括模型描述接口、数据集接口、core engine、debug部分

* 模型描述：（即编写YAML文件）

1. 实体定义：实体可以表示为node或edge或子图。（我理解是，比如2分类问题就可以描述为两个实体；最小路径搜索的问题也可以描述为一个实体等）。
2. 消息传递阶段定义：即实体之间的联系，包括源实体和目的实体，这里就是定义消息传递模型（包括消息聚合方式，更新函数等）。
3. Readout：传递完消息，读出消息，获取node、graph级别的输出，完成实用的任务（比如预测结果）。结合本文的官网，我认为这一部分是将NN的运算结果进行比如池化之类的操作降维，以便进行后续的任务。
4. NN定义：提供一个直接映射的接口，定义[2][3]的中的具体NN操作，如下图所示。

* 数据集：IGNNITION需要json形式的数据集结构，提供数据集转化的接口
* 核心引擎：将YAML文件转化为TensorFlow代码，读取数据集，进行模型训练，只需三行代码即可运行YAML定义的GNN model。
* Debug：交互式debug（我并没有使用，文中说是可以指出YAML文件的错误）

5 实验：通过使用IGNNITION和TensorFlow实现RouteNet和GQNN，验证两种实现的准确率与性能相差无几。

