# 🎓Neural Subgraph Isomorphism Counting

1. **论文信息**
   1. 论文名：Neural Subgraph Isomorphism Counting
   2. 领域：Theory of computation→ Pattern matching; • Information systems → Information integration.
   3. 关键词：Subgraph Isomorphism, Dynamic Memory, Neural Network
   4. 发表年份：2020
   5. 会议/期刊名：ACM SIGKDD
   6. 级别：A
2. **INTRODUCTION**
   1. **目前存在的问题和解决方法**
      1. **效率：**While the learning based approach is inexact, we are able to generalize to count large patterns and data graphs in linear time compared to the exponential time of the original NP-complete problem.
      2. **数据问题：**
         1. 为该问题生成足够的训练数据：典型数据集大小有限，远不足以训练图神经网络。与其他图学习问题（如最短路径或中心性）相比，hard to get ground truth。
      3. **寻找合适的inductive bias(归纳偏置)**：
         1. 应用更强和更复杂的表示学习技术：一般情况都需要考虑异构类型的节点和边，为了表示和区分具有邻域结构的节点和边。
         2. 模型设计中必须考虑两个图之间的相互作用：任务不是在单个图上执行，而是涉及额外的模式。
         3. 考虑计算效率：该模型将被训练并应用于具有大图和大量样本的数据集，考虑到可能模式的多样性。
   2. **本文方法和创新点**
      1. **创新点：**
         1. This is the first work to model the subgraph isomorphism counting problem as a learning task，training and inference enjoy linear time complexities, scalable for pattern/graph sizes and the number of data examples.
         2. 为序列模型和图模型提供了通用的编码方法。
         3. 引入了一个Dynamic Intermedium Attention Memory(DIAMNet)来解决计数的更全局的推理问题。
         4. 实验结果表明，基于学习的子图同构计数比传统算法VF-2快10-1000倍，在可接受的误差范围内。
      2. **数据集：**
         1. 实验开发两个大规模数据集：包括一个小图（每个图≤ 1024个计数）和一个大图（每个图≤ 4096个计数）
         2. 基准数据集：MUTAG
      3. **方法：**
         1. 序列模型建模：将图的边定义为五元组(u,v,X(u),y,X(v))，称为code, 其中X(u)是起点标签,X(v)是终点标签。参考gSpan中定义的顺序来比较code的字典序。图与一个minimum code或者五元组的列表对应。再将minimum code编码为multi-hot matrix。然后使用序列建模的一般策略来学习图中边之间的依赖关系。Convolutional Neural Networks (CNNs)、Long Short-Term Memory (LSTM)、Transformer-XL (TXL)。
         2. 图模型建模：图模型中，每个顶点都有一个特征向量，每个边都用于将信息从源传递到目标。在图神经网络中设置编码后的顶点标签作为顶点特征，并通过边传播混合消息。
            1. Relational Graph Convolutional Network (RGCN): 每个关系对应于变换矩阵，将关系特定信息从源变换到目标。选择block-diagonal-decomposition来解决参数数量随关系数量的快速增长，并遵循原始设置的mean aggregator 。
            2. Graph Isomorphism Network (GIN): 在多层感知器(MLP)和sum aggregator的帮助下更好地捕获homogeneous graph structures。在每个relational graph convolutional layer之后添加MLPs，并使用sum aggregator来聚合消息，将此变体命名为RGIN。
         3. 从序列模型或者图模型中得到列向量为d-dimensional的 graph representation 和 pattern representation。将它们输入interaction layers，然后提取pattern和graph之间的correlated context。
            1. 问题：如果使用attention mechanism 建模 graph representation 和 pattern representation 之间的 interactions 以及 图自身的interactions，时间复杂度过高：序列模型：O(|EP | · |EG| + |EG|^2) 图模型：O(|VP | · |VG| + |VG|^2)
            2. 解决方法：提出了一种新的网络：Dynamic Intermedium Attention Memory(DIAMNet)：*using an external memory as an intermedium to attend both the pattern and the graph in order。*时间复杂度降低至 O(T · M · (|EP | + |EG|))或者 O(T · M · (|VP | + |VG|))，M是内存大小 ，T是递归步数。
   3. **实验**
      1. **实现了五个不同的Representation Models：**
         1. CNN：3-layer convolutional layers followed by max-pooling layers
         2. LSTM：simple 3-layer LST model
         3. TXL：6-layer Transformer encoder with additional memory.
         4. RGCN：3-layer RGCN with the block-diagonal-decomposition.
         5. RGIN：combination of GIN and RGCN: adding a 2-layer MLP after each relational convolutional layer and using the sum aggregator.
      2. **Interaction Networks：**
         1. Pool：A simple pooling operation is directly applied for the output of neural models for classification or regression.
         2. MemAttn: Consider to use memory for the key and the value.Finally, get the mean-pooled representations for the pattern and the graph and feed the same combination in Pool to the FC layers.
         3. DIAMNet: mean-pooling is fast and performs well. Feed the whole memory with size information into the next FC layers.
            1. The memory in DIAMNet is regarded as the query, initialized once and updated **t** times
            2. MemAttn initializes memory **2t** times, takes memory as the key and the value, and updates graph representations **|graph representation| · t** times.
         4. **实验设置：**
            1. 使用均方误差(MSE)来训练模型，并根据dev集的结果选择最佳模型。
            2. For the small and large datasets, first train models on the small and further use curriculum learning on the large.
         5. **实验结果：**
            1. 在不同表示模块下，选择最佳的池化方式得到的结果进行对比： 1. 选择预测全为0和全为平均值作为baseline 2. 与传统算法VF-2的速度进行对比
            2. 结果显示：

graph model 优于大多数 sequence model，CNN模型在图同构计数问题表现最差，

LSTM models with different interaction networks are constantly better than TXL.

RGIN is much better than RGCN and other sequence models, which shows that MLPs and the sum aggregator are good at vertex representation learning and structure modeling in this task.

实验使用了不同的池化方法和注意力机制，包括 SumPool、MeanPool、MaxPool、MemAttn 和 DIAMNet，对比它们在prediction layer的表现。For all representation architectures, DIAMNet 表现最好。

DIAMNet 的内存和计算成本随着输入大小线性增长，在处理具有数千甚至数百万个顶点和边的实际数据时，效率优化明显。

虽然 MemAttn 也可以将二次计算成本降低到线性，但其性能不稳定。

CNN with DIAMNet is the best among sequence models.

通过使用CNN模型比较了MaxPool和DIAMNet的性能，在CNN上两者差异最明显

1. **RELATED WORK**
   1. **subgraph isomorphism problem:**
      1. **Most subgraph isomorphism algorithms are based on backtracking**：
         1. Ullmann，VF2、GraphQL：first obtain a series of candidate vertices and update a mapping table，then recursively revoke their own subgraph searching functions to match one vertex or one edge at a time.。
            1. problem：It is still hard to perform search when either the pattern or the data graph grows since the search space grows exponentially as well.
      2. **Some other algorithms are designed based on graph-index:**
         1. gIndex：can be used as filters to prune out many unnecessary graphs.
            1. problem: However, graph-index based algorithms' time and space in indexing also increase exponentially with the growth of the graphs.
      3. **Add weak rules:**
         1. TurboISO and VF3: Add some weak rules to find candidate subregions and then call the recursive match procedure on subregions. These weak rules can significantly reduce the searching space in most cases
      4. **some approximation techniques for triangle counting:**
         1. hard to generalized to large patterns
      5. **Random walks 、Graph simulation...**
   2. **Graph Representation Learning:**
      1. Most of them focus on generalizing the idea of convolutional neural networks for general graph data structures or relational graph structures with multiple types of relations
      2. **graph isomorphism network (GIN)、recurrent neural networks (RNNs)**
      3. With external memory, sequence models can work well on complicated tasks such as language modeling and shortest path finding on graphs
      4. **graph kernels:** also convert graph isomorphism to a similarity learning problem
         1. limit: usually work on small graphs and do not focus on subgraph isomorphism identification or counting problems.