Systematic Generalization with Edge Transformers

Relevant Resources:

paper: https://arxiv.org/pdf/2112.00578

这篇文章发表于NeurIPS 2021，提出了**Edge Transformer**，它更自然地适应关系推理

Motivation:

Nevertheless, systematic (or compositional) generalization benchmarks remain challenging for this class of models, including large instances with extensive pre-training (Keysers et al., 2020; Tsarkov et al., 2020; Gontier et al., 2020; Furrer et al., 2020). Similarly, despite their increasing application to a variety of reasoning and inference problems, the ability of Graph Neural Networks' (GNN) (Gori et al., 2005; Scarselli et al., 2009; Veli ˇckovi'c et al., 2018) to generalize systematically has also been recently called into question (Sinha et al., 2019, 2020). Addressing these challenges to systematic generalization is critical both for robust language understanding and for reasoning from other kinds of knowledge bases.

系统（或组建）泛化基准对这类模型仍然具有挑战性。

系统推理概括能力存在质疑。

直觉是经典transformer可以被解释为类似于Prolog逻辑编程语言子集的连续放松的一种推理系统，在本工作中，作者提出了Edge Transformer，它是Transformer模型的一种推广，使用了一种新的三角形注意机制，该机制受到了更一般的推理规则家族的启发。

Intuitions:

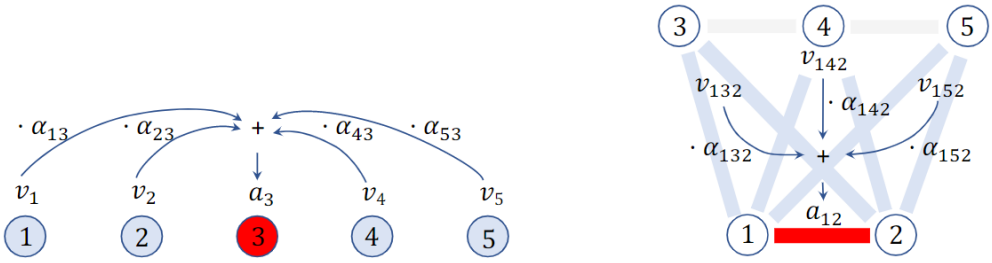

Figure 1: Left: Transformer self-attention computes an update a_3 for the node 3 by mixing value vectors v_i associated with each node i with attention weights α_i. Right: triangular attention computes an update a_{12} for the edge $(1, 2)$ by mixing value vectors v_{1l2} computed for each triangle $(1, l, 2)$ with attention weights α_{1l2}. Values v_3, v_{112}, v_{122} and their contributions are not shown for simplicity.

Transformer操作一组n个实体，例如句子中的单词，将其表示为图1左侧显示的图中的节点。每个节点都与i的d维节点状态X(i)相关联，它可以被认为是与节点应用某个函数相关联的输出，该函数表示节点属性。在变压器中，计算通过顺序地将每个节点与l个节点状态关联起来，每个状态通过注意机制从上一层的节点状态更新而来。

采用类似prolog的表示法，我们可以将转换器架构实现的基本推理过程编写为：

$$X^{l+1}(1) \vdash_A X^l(1), X^l(2), \ldots, X^l(n).$$

在逻辑编程中，旋转门符号 ⊢ 意味着当表达式的右边为真时，左边也必须为真。用A 表示的推理规则是通过注意机制A 学习的，$X^l(\cdot)$也是如此。

对于许多实际的推理和NLU任务系统，必须学习关系之间的依赖关系，例如家庭关系:MOTHER(x,y)和MOTHER(y,z)意味着GRANDMOTHER(x,z)。这样一个一般的推理问题可以用类似prolog的符号表示如下:

$$X^{l+1}(1,2) \vdash_A X^l(1,1), X^l(1,2), \ldots, X^l(2,1), X^l(2,2).$$

经典的变压器也有可能捕捉到这样的推理模式，但要做到这一点，每个变压器状态$X^{l+1}(\cdot)$必须编码关系本身的属性，即MOTHER(·，·)和GRANDMOTHER(·，·)，以及关系中节点1所在的所有目标节点x，即MOTHER(1,x)。换句话说，要对关系进行编码，一个经典的变压器必须将其状态表示用于两个不同的目的。这种担忧的混合给学习带来了沉重的负担。

为了解决这个问题，作者建议模型表示的基本对象不应该是与节点关联的状态，而是与边关联的状态，就像图1右边显示的那些状态。这样的边缘表示是基于对其他边缘的关注机制而更新的。这种机制的灵感来自于逻辑编程中的统一过程。

例如，GRANDMOTHER(x, y) $\vdash_A$ MOTHER(x,z) MOTHER(z,y)是指这个规则右边的两个谓词共享一个变量z。成功推断出GRANDMOTHER(x,y)，其中涉及找到z的绑定，满足MOTHER(z,y)和MOTHER(x,z)，这是一个在逻辑编程中通过统一处理的过程。

Model:

Edge Transformers

边缘变压器运行在一个具有n 个节点和n^2 个有向边(我们允许自连接)标记为d维特征向量的完整图上。

因此，edge transformer状态是一个三维张量$X \in R^{n,n,d}$，它由每条边$(i,j), i,j \in [1,n]$对应的边状态$X(i,j)$组成。对于每个边的状态，我们也会写出x_{ij}。Edge transformer层计算:

$$X' = FFN(LN(X + TriangularAttention(LN(X)))).$$

这里，FFN（X）是一个完全连接的前馈网络，其一个隐藏层为4·D单元，其独立地应用于每个边缘状态X_{ij}，LN代表层归一化机制（Ba等，2016）在Edge状态XIJ中的出现时重新激活每个特征$f \in [1，d]$。它遵循变压器的熟悉结构（Vaswani等，2017）计算具有两个重要差异的计算：（a）它使用3D张力状态而不是矩阵状态;（b）它利用了我们描述的新型三角形关注机制。

对于单边状态x_{ij}，单头三角注意更新输出向量a_{ij}，计算方法如下:

$$a_{ij} = W^o \sum_{l \in [1,n]} \alpha_{ilj} v_{ilj},$$

$$\alpha_{ilj} = \underset{l \in [1,n]}{\text{softmax}}\, q_{il} k_{lj} / \sqrt{d},$$

$$q_{il} = W^q x_{il},$$

$$k_{lj} = W^k x_{lj},$$

$$v_{ilj} = V^1 x_{il} \odot V^2 x_{lj},$$

⊙代表元素相乘，W^Q，W^K，W^O，V^1，$V^2 \in R^{d,d}$，是用于生成query，key，output和value向量k_{il}，q_{lj}，a_{ij}和v_{ilj}的矩阵。

非正式地说，与边(i,j)相关的更新通过聚合共享节点l的所有对边的信息进行，即所有对(i,l)和(l,j)。更新后的边值a_{ij}是每对边贡献的注意加权混合。图1显示了变压器和边缘变压器计算之间的一些关键区别。注意边(i,j)，(i,l)和(l,j)在图中形成了一个三角形，因此我们的注意机制得名。

我们定义了上述机制的一个多头泛化，其方式类似于vanilla transformer中多头注意的实现.具体来说，每个头h [1,m]将执行Edge Attention 使用较小的对应矩阵W^Q，W^K，V^1，$V^2 \in R^{d,d}$，并通过乘以W^O的头部输出的连接来计算联合输出a_{ij}：

$$a_{ij} = W^O[a_{1ij}; \ldots; a_{mij}].$$

Discussion:

作者提出了边缘转换器，这是神经模型可能受益于扩展的3d张量状态的一个实例，它更自然地适应关系推理。实验结果令人鼓舞，在三个组合泛化基准测试中，Edge Transformer的性能明显优于竞争基线。

边缘变压器计算和存储器使用为$O(n^3)$，对于输入大小n。虽然这使得缩放到更大的投入具有挑战性，但在我们的经验中，现代GPU硬件允许培训边缘变压器大小的输入到达一百个实体。使用更大输入的应用中使用边缘变压器的方法是仅为输入的最突出或重要组成部分创建边缘变压器节点。例如，在语言建模上下文中，这些可以只是名词短语的头和/或仅命名实体。未来的工作可以在原始文本混合模型上构建和预拉力，该模型结合了基于变压器和边缘变压器的处理，后者仅针对小节点的小子集执行。