



神经符号系统: 非确定性管理的视角

李泽南^{1,2}, 姚远^{1,2}, 马晓星^{1,2*}, 吕建^{1,2}

1. 南京大学计算机软件新技术全国重点实验室, 南京 210023

2. 南京大学计算机学院, 南京 210023

* 通信作者. E-mail: xxm@nju.edu.cn

收稿日期: 2024-09-25; 修回日期: 2024-11-18; 接受日期: 2024-11-30; 网络出版日期: 2025-01-07

国家自然科学基金 (批准号: 62025202) 和江苏省软件新技术与产业化协同创新中心资助项目

摘要 作为一种智能化软件形态, 融合了神经网络和符号规则的神经符号系统正受到越来越多的关注, 相关研究爆炸性增长. 然而, 现有技术仍然难以有效支持较大规模神经符号系统“端到端”的自动构建. 神经网络训练和符号规则学习过程中对非确定性的不兼容的处理方式是造成神经符号学习效果欠佳的一个关键因素. 本文旨在以非确定性管理的新视角, 较为系统、一致、严谨地讨论分析“端到端”神经符号学习的技术挑战, 并以此视角整理讨论相关工作的思路、进展和当前仍然存在的问题. 在此基础上, 展望下一步的发展方向.

关键词 神经符号学习, 神经符号接地, 推理捷径, 符号约束合成

1 引言

作为一种智能化软件形态, “神经符号系统” (neuro-symbolic systems) 融合了神经网络和符号规则, 重要性日增. 首先, 从经验中学得知识是人类智能的体现. 相应地, 无需显式编程, 直接从数据中归纳出以符号规则的形式来表达的知识, 长期以来一直是计算机科学的重要挑战问题^[1~4]. 从认知科学的角度来看, 神经符号系统与丹尼尔·卡尼曼 (Daniel Kahneman) 所提出的“系统 1 & 系统 2”人类思维模式相符^[5]. 在人类智能中, 系统 1 推理是快速、直觉, 却不准确的, 这与神经网络模型的工作方式非常类似; 而系统 2 推理是慢速、逻辑, 但准确的, 这与基于符号规则的演绎推理行为相似. 因此, 许多著名人工智能学者认为, 以神经符号学习构建神经符号系统, 是走向下一代人工智能的必经之路^[6~9]. 从软件工程的角度来看, 神经符号系统已经在自动驾驶系统^[10]、智能医疗设备^[11]、机器人语音操控^[12]等许多领域展现出巨大应用前景; 更重要的是, 神经符号学习这一框架融合了“智能化软件的开发”和“软件的智能化开发”两方面的内容, 其基本原理和算法的研究具有特别重要的方法论意义.

引用格式: 李泽南, 姚远, 马晓星, 等. 神经符号系统: 非确定性管理的视角. 中国科学: 信息科学, 2025, 55: 1–31, doi: 10.1360/SSI-2024-0163

Li Z N, Yao Y, Ma X X, et al. Neuro-symbolic systems: a perspective of uncertainty management. Sci Sin Inform, 2025, 55: 1–31, doi: 10.1360/SSI-2024-0163

神经符号系统涉及的相关研究相当广泛,但现有技术仍然难以有效支持大规模系统“端到端”^{[13]1)}的自动构建^[14~17].已有研究多关注其中一个侧面.例如,在人工智能领域,归纳逻辑编程^[18,19]将归纳学习过程与逻辑程序结合,基于一系列样例以及预先定义好的逻辑原子,自动组合出完整的逻辑规则;而在软件工程领域,基于示例的程序合成^[20,21]和面向演示的程序合成^[22,23]的技术也被广泛研究,实现了在一些领域特定语言(domain specific language, DSL)^[24]上编程的某种程度的自动化.然而,这些技术均依赖于规则模板或者DSL的定义,也就是说,将现实物理世界的现象映射到抽象符号世界的概念这一关键步骤仍然需要人类专家来完成.另一方面,DeepProbLog^[25], Semantic Loss^[26], DL2^[27]等方法试图将符号知识引入神经网络训练过程,以提高网络的预测性能.然而,这些符号知识同样需要人类专家来提供,而且神经网络模型也难以真正学得这些知识^[28].除了人工负担的问题之外,这些技术更深层的不足在于未能实现神经网络与符号规则的“相互促进”,从而导致整个系统的学习和推理的效率与效果受限.

“非确定性”(uncertainty)是理解高效融合神经符号之内在技术挑战的一个关键.非确定性是神经符号系统在构建过程中无法回避的因素:一方面,系统面对的物理世界问题本身常常是天然不确定的,例如天气预报问题;另一方面,现代神经网络的学习过程中也会引入随机性,例如随机梯度下降、Dropout技术等^[29,30].神经网络和符号规则两类软件实体对非确定性的处理方式截然不同.前者基于学习泛化理论,把非确定性隐式地包含于期望损失最小化的学习目标之中.而且主流的神经网络学习技术常常牺牲对于非确定性的精确管理,并以此来换取更好的预测准确性^[31,32].后者的构建过程中,人类开发者负责抽象掉现实世界问题的非确定性,以换取符号规则推理的可计算性、可解释性和可组合性.然而,在结合神经网络模块与符号规则模块,实现“端到端”神经符号学习时,这两个相互割裂的非确定性处理方式带来了很大的问题.一方面,神经网络训练忽略显式化的非确定性管理,直接追求最小化经验损失,导致在后续神经符号接地过程中,极易容易陷入逻辑推理捷径(见第4节),使得整个神经符号系统的性能不佳;另一方面,人工排除了非确定性后的符号演绎系统,难以通过梯度下降的方式和神经网络模块共同训练、相互促进,极大地影响了神经符号学习的效率.因此,可以说,“端到端”神经符号学习需要一套新的非确定性管理方案.

本文旨在以非确定性管理的新视角,较为系统、一致、严谨地讨论分析“端到端”神经符号学习的技术挑战.基于作者在前期工作^[33~35]中所获得的认识,我们首先使用标准的数学框架将神经符号学习建模成一个双层优化问题,用以讨论“端到端”学习的优越性和必要性,以及当前实现“端到端”学习的两个主要技术瓶颈:“神经符号接地”以及“可微符号合成”.在这一框架下,我们综述了现有的神经符号学习工作,以非确定性管理的视角解读它们的贡献和当前仍然存在的问题.在此基础上,我们展望了下一步的发展方向.神经符号学习领域的研究文献数量爆炸性增长,详尽综述已不现实且无必要.较之神经符号学习方向近期已有的一些综述^[14~17,36],我们期待本文通过上述框架,能够为读者提供一个足够全面深入、更为系统一致,同时也更有利于理解掌握的路线指引.

本文的组织结构如下.第2节介绍神经符号系统、“端到端”神经符号学习,以及相应的双层优化建模框架.第3节讨论神经符号接地问题及其主要技术路线:模型计数方法和模糊逻辑方法.我们还将分析已有方法如何引入非确定性,以及利用非确定性加速神经符号接地.第4节讨论推理捷径问题,通过理论分析来解释非确定性管理对推理捷径产生的影响.我们还将讨论在一般监督学习中和在神经符号接地中的非确定性管理方法之间的联系.第5节详细介绍可微符号合成问题,对已有工作按符号约束结构的不同进行归类,分别解释其非确定性引入方法.第6节讨论当前所需解决的关键挑战,并展望未来工作的分析.最后第7节总结全文.

1) 一般而言,神经符号系统因其应用目的的不同而有不同的构建方式,未必是端到端的.例如,在符号求解中引入深度学习,使用符号知识改善深度学习系统等^[13].但本文围绕端到端的神经符号学习展开讨论,这不仅是出于理论框架的纯粹性考虑,也是因为它在实现上涵盖了非端到端构建时用到的大多数关键技术,更重要的是它将迫使我们更深入地理解神经符号融合交互的核心问题.

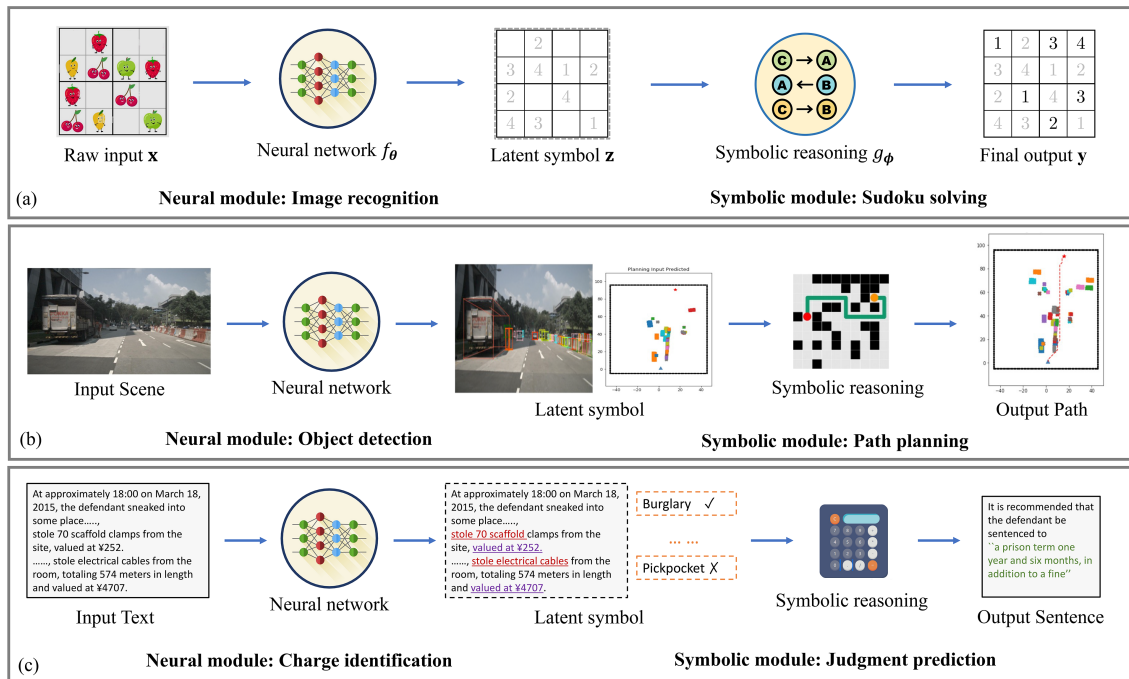


图 1 (网络版彩图) 神经符号系统的 3 个实例, 分别完成 (a) 视觉数独求解, (b) 自动驾驶, 以及 (c) 智慧司法任务.

Figure 1 (Color online) Three examples of neuro-symbolic systems, performing (a) visual Sudoku solving, (b) self-driving, and (c) smart justice tasks, respectively.

2 神经符号系统与神经符号学习

2.1 神经符号系统

为了聚焦于神经-符号交互的基本问题, 本文考虑一种最简单常见的神经符号系统形式^{[37,38]2)}: 系统顺序地包含一个“神经网络模块”和一个“符号规则模块”. 其中, 神经网络模块可以是一个神经网络或者是多个神经网络的集成, 它通常负责从现实物理世界中获取信息, 并将其映射到某个符号空间. 换句话说, 神经网络模块的主要功能是完成那些难以精确描述的感知任务, 例如图像分类、语音识别、自然语言理解等^[39~41]. 符号规则模块以神经网络在符号空间的感知结果作为输入, 并通过一系列精确的逻辑推理, 演绎计算出最终的结果. 符号规则模块可有多种实现方式, 例如使用模型推理系统、SMT 求解器, 或人类专家编制的程序等^[42~44]. 图 1 给出了几个神经符号系统的例子. 其中图 1(a) 数独问题常被用作一个神经符号学习的简单示例和性能算法评估的实验对象. 图 1(b) 和 (c) 则分别展示了神经符号系统在自动驾驶和智慧司法领域的应用场景. 此外, 一些神经符号系统的其他典型应用可以参考文献 [45].

神经符号系统充分发挥了神经网络和符号规则的不同优势, 因而在许多混合了感知与推理的任务中能够表现出更好的性能. 通过这种结合, 系统在处理大规模、高维度的感知任务时, 可利用神经网络模块进行有效的特征提取和模式识别, 例如图像分类和语音识别; 同时, 在需要执行复杂决策和推理的任务中, 符号规则模块能够提供明确的逻辑演绎处理能力, 如在自动驾驶任务中进行路径规划. 因此, 相比较于完全的符号推理系统, 神经符号系统节省了人类专家建模现实物理世界的成本, 也具有更好

2) 神经符号系统中神经网络部件和符号规则部件可有多种组合方式. 近期有报告介绍了 6 种神经符号系统结构^[37]. 但对于本文所考虑的非确定性管理问题而言, 这 6 种结构形式均可以归结为前述简单形式的不同侧重点的变种. 其中一个特别值得一提的结构形式来自于文献 [38]: 神经网络模块在提供感知预测的同时也提供快速且不精确的推理结果. 这使得我们能够采用一种投机策略来加速神经符号系统的推理速度. 本在第 6 节对其进行讨论.

的泛用性;而相比较于完全的深度学习系统,神经符号系统可具有更好的鲁棒性和可解释性.

2.2 神经符号学习

神经符号学习的目的就是以数据驱动的方式,将神经符号系统的构建过程自动化.形式上,定义 $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ 分别为(原始)输入空间,(最终)输出空间,以及中间的(隐)符号空间.那么,神经符号系统可以表示为从 $\mathcal{X}$ 到 $\mathcal{Y}$ 的映射,其中神经网络模块实现一部分映射 $f_{\theta}: \mathcal{X} \rightarrow \mathcal{Z}$,而符号规则模块实现另一部分 $g_{\phi}: \mathcal{Z} \rightarrow \mathcal{Y}$.

隐符号空间连接了神经网络模块与符号规则模块,因此它在神经符号系统中至关重要.可以看到,除样本 $x \in \mathcal{X}$ 和最终标签 $y \in \mathcal{Y}$ 可以被观测以外,如果中间隐符号空间对应的标签 $z \in \mathcal{Z}$ 同样可以获得,那么神经网络模块和逻辑推理模块可以独立地实现自动构建:对于神经网络模块 f_{θ} ,隐符号空间的标注 z 提供了该模块的强监督信号.结合对应输入空间的样本 x ,通用的监督学习技术就可以训练出要求的深度学习模型;对于符号规则模块 g_{ϕ} ,隐符号空间的标注 z 实际上是该模块输入空间的样本.结合对应输出空间的标注 y ,经典的归纳逻辑编程技术^[19]以及程序合成技术^[46]同样可以得到要求的逻辑推理系统.换句话说,在隐符号空间可获得的情形下,我们可以使用两个独立优化问题来表示神经网络模块的训练和符号规则模块的学习:

$$\theta = \arg \min_{\theta \in \Theta} \mathbb{E}_{(x,z) \sim \mathcal{D}_1} [\ell_1(f_{\theta}(x), z)], \quad \phi = \arg \min_{\phi \in \Phi} \mathbb{E}_{(z,y) \sim \mathcal{D}_2} [\ell_2(g_{\phi}(z), y)],$$

其中 $\ell_1(\cdot, \cdot)$ 和 $\ell_2(\cdot, \cdot)$ 分别表示神经网络模块和符号规则模块的训练损失.

然而,隐符号空间的标签往往代价昂贵且需要较强的领域知识.例如,在图 1(b) 中的自动驾驶任务中,粗略标记每一张驾驶图片需要 7 min;如果精确标记图片(像素点级别),每一张图片甚至需要 1.5 h^[47].对于图 1(c) 中的判罪估算任务,将相应的法规落地到具体的数值,需要经验丰富的律师,并且结合案例具体分析得到结果,这也是代价非常昂贵的.相比较而言,输出空间的标记通常较为简单.对于前述的两个例子,在自动驾驶任务中,输出空间的标记是驾驶轨迹,这只需要从人类驾驶员合法开出的路径中进行采集,这是非常廉价的.而在判罪估算任务中,输出空间的标记是最终的判决结果,这同样也可以直接从网络上收集得到.“端到端”的神经符号学习正是一种能够避免昂贵的隐符号空间标注的关键技术.

“端到端”的神经符号学习的另外一个重要优势在于,耦合两个模块的构建过程,可以相互促进二者从而提高神经符号整体的学习效果和效率.一方面,对隐符号空间的人工标注虽然使得神经网络模型的训练变得很容易,但也使得网络训练过程独立于下游任务,因此并不能很好地适配后续符号推理模块;另一方面,人工标注的数据往往多样性不足,这使得符号推理模块难以处理模型由于非确定性导致的异常输出.此外,一系列的现有理论和实验^[35, 48~50]表明,将两个模块的训练结合,可以实现二者性能乃至神经符号系统整体性能的提高:在网络训练过程中引入符号推理的知识,能够提高神经网络模块的准确率;而利用神经网络模块的输出,也可以加速符号规则的学习,并且使得学得符号规则模块更加鲁棒.

综上,通过引入隐符号变量 z ,我们将神经符号学习任务建模为这两个模块的联合训练问题:

$$\begin{aligned} \theta &= \arg \min_{\theta \in \Theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell_1(f_{\theta}(x), z)], & \phi &= \arg \min_{\phi \in \Phi} \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell_2(g_{\phi}(z), y)], \\ \text{s.t.} \quad g_{\phi}(z) &= y, \quad z \in \mathcal{Z}, & \text{s.t.} \quad z &= f_{\theta}(x), \quad z \in \mathcal{Z}. \end{aligned} \quad (\text{P})$$

隐符号变量 z 耦合了这两个模块的训练:在神经网络模块的训练过程中,监督信号 z 由符号规则模块确定;而在符号规则模块的训练过程中,输入样本 z 是神经网络模块的输出.

2.3 神经符号学习的关键挑战

在上述建模下,神经符号学习问题 (P) 是一个双层优化问题^[51, 52].从优化算法的角度来看,该问题最直觉的解决方案就是交替最小化^[53, 54]这两个优化目标.具体地说,在每一步迭代过程中,固定

一个模块, 最小化另外一个模块, 依次交替执行直至最终收敛. 但该方法的收敛性至今存疑, 例如, 在最简单的双线性双层优化问题 ($\min_{\mathbf{x}} \mathbf{x}^T \mathbf{y}$, s.t. $\mathbf{y} = \arg \min_{\mathbf{y}} -\mathbf{x}^T \mathbf{y}$) 上, 交替最小化都会陷入循环问题^[55~57]³⁾. 此外, 其每一轮迭代都需要计算两个最小化的子问题, 导致算法的整体效率过低. 而且, 实践经验表明, 迭代中的最优模型往往造成梯度信息丢失, 影响最终的性能. 例如, 从对抗生成网络^[58, 59]的训练中就可以看出, 如果在训练过程中直接最小化判别器的损失函数, 得到最优判别器反而不能提供足够的梯度信息, 从而导致生成器的模式坍塌^[60~62].

在可微的情形下, 求解双层优化问题更有效且高效的方法是交替梯度下降^[63, 64]. 首先, 在合适的假设下, 它和交替最小化具有相同的线性收敛速率^[65~67]. 而且, 交替梯度下降只需要计算损失函数的梯度, 并不要求解额外的子优化问题, 因此总体效率更高. 更重要的是, 近期一系列研究以及实验表明, 该方法及其变种反而具有更好的性能^[68~70]. 对此, 目前最合理且最有可能的一个解释是: 与交替最小化相比, 交替梯度下降增强了两个模块之间的交互, 因而能够在训练过程中引入更多的损失函数和梯度信息, 确保最终收敛到了更好的结果^[71~73].

但要将交替梯度下降法应用于神经符号学习问题 (P), 有两个困难需要解决: (i) 如何枚举巨大的可行符号标记空间以监督神经网络训练, 以及 (ii) 如何将离散的符号约束空间连续化以支持梯度下降.

要理解困难 (i), 考虑神经网络模块的训练问题:

$$\theta = \arg \min_{\theta \in \Theta} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\ell_1(f_{\theta}(\mathbf{x}), \mathbf{z})], \quad \text{s.t.} \quad g(\mathbf{z}) = \mathbf{y}, \quad \mathbf{z} \in \mathcal{Z}. \quad (1)$$

该问题的核心计算困难在于符号空间 $\mathcal{Z}$ 通常是离散的, 并且满足 $g(\mathbf{z}) = \mathbf{y}$ 的符号执行解并不唯一. 在已有文献中, 该问题经常被称为“神经符号接地问题”(symbol grounding problem)^[74, 75]. 直观上看, 我们可以对每一个训练样本穷举所有的可行符号标记, 再基于每一种可能的标记结果计算损失函数, 以及整个训练集上的期望损失, 进而计算梯度. 但是, 这样的候选符号接地空间大小约为 k^N , 其中 N 表示训练样本的个数, 而 k 表示可行符号标记的数量. 因此, 这在复杂度上是完全不可接受的, 且即使穷举单个样本的可行符号标记的算法复杂度本身就是 #P-hard^[76~78].

为了缩小符号接地问题的穷举空间, 已有的研究均试图在训练过程中引入网络模型对于符号标记的非确定性作为权重, 来引导和加速符号标记的搜索. 但是, 最新的一系列研究发现, 过于依赖模型非确定性又会导致符号接地过程陷入“推理捷径问题”(reasoning shortcuts)^[33, 79~81]. 在后续章节中, 我们将通过简单的理论分析来说明当前已有的神经符号学习工作难以有效地解决推理捷径问题. 因此, 我们认为: 神经网络模块训练的关键挑战在于如何管理非确定性, 即既需要利用非确定性加速符号接地的计算, 又必须防止由于过度依赖模型非确定性而陷入推理捷径.

要理解困难 (ii), 考虑符号规则模块的训练问题:

$$\phi = \arg \min_{\phi \in \Phi} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\ell_2(g_{\phi}(\mathbf{z}), \mathbf{y})], \quad \text{s.t.} \quad \mathbf{z} = f(\mathbf{x}), \quad \mathbf{z} \in \mathcal{Z}, \quad (2)$$

其中, 每一个参数 $\phi \in \Phi$ 实际上表示一条候选逻辑推理规则. 换句话说, 符号规则模块的训练问题本质上是候选逻辑推理规则的搜索问题. 该问题的核心计算困难在于参数空间 Φ 是离散的, 因此无法直接计算关于 ϕ 的梯度. 同样地, 已有的工作试图穷举所有的候选逻辑规则, 并且为每一条规则引入相应的非确定性 (也称为权重或者分数). 在这样的情况下, 关于候选逻辑规则空间的离散搜索问题就被转化为了关于非确定性系数的连续优化问题, 使得梯度下降法能够被直接实现.

我们将这一可微化 – 梯度优化过程称为“可微符号约束合成”(differentiable symbolic constraint synthesis). 但是, 在可微符号约束合成中, 刻画参数空间 Φ 通常非常困难. 如果参数空间太大, 那么将无法穷举所有的候选逻辑规则, 影响训练效率以及最终的效果. 反之, 如果参数空间过小, 那么符号推理模块的表达力不够, 同样无法得到良好的结果. 另外, 任意的可微化会导致符号推理的精确性丢失.

3) 一个更形象的比喻: 石头剪刀布游戏中, 玩家若仅根据上一轮信息“最优化”决策, 就可能陷入“剪刀 – 石头 – 布 – 剪刀 – ...”的循环.

所以,当前的研究针对多种不同的逻辑规则结构,例如,霍尔子句、合取/析取表达式,以及整数线性约束等,设计不同的可微化方案. 综上,我们认为:非确定性管理也是符号规则高效合成的关键. 一方面要针对不同的逻辑规则结构,以恰当的形式引入非确定性以可微化逻辑规则的搜索问题,提高其计算效率;另一方面又要有办法最终消除非确定性,以得到精确的符号规则.

3 神经符号接地

我们从神经网络模块的训练问题开始,在确定符号规则模块 g 的情形下,我们有

$$\theta = \arg \min_{\theta \in \Theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell_1(f_\theta(x), z)], \quad \text{s.t.} \quad g(z) = y, \quad z \in \mathcal{Z}. \quad (\text{P}_1)$$

给定样本对 (x, y) 时, $g(z) = y$ 的解并不唯一并且解空间是高度结构化且离散的. 因此,神经网络模块的训练问题 (P_1) 也被称为神经符号接地问题 (symbol grounding problem): 找到映射 $f_\theta: \mathcal{X} \rightarrow \mathcal{Z}$, 使得对于分布 $\mathcal{D}$ 上采样的样本 (x, y) , 映射后 $z = f_\theta(x)$ 能够满足符号约束 $\mathcal{S}_y = \{z \mid g(z) = y\}$. 此外,我们将符号空间的标记 z 称作伪标签 (pseudo-label) 或者伪符号接地 (pseudo-grounding), 并且将满足符号约束的伪符号接地 $z \in \mathcal{S}_y$ 称为可行伪符号接地 (feasible pseudo-grounding).

符号接地问题的解决方案主要分为两种: (1) “模型计数” (model counting) [82, 83] 和 (2) “模糊逻辑” (fuzzy logic) [84, 85]. 两种方法的核心思想都是使用一种点-集合 (即 $f_\theta(x)$ 到 $\mathcal{S}_y$ 的距离或散度来替代原本的点-点 (即 $f_\theta(x)$ 与 z) 损失函数. 更具体地说,二者均使用模型的预测非确定性 $P_\theta(z|x)$ 代替离散的预测结果 $f_\theta(x)$, 其中前者引入并最大化 $P_\theta(z|x)$ 在 $\mathcal{S}_y$ 上的后验概率似然; 而后者试图定义并最小化 $P_\theta(z|x)$ 到 $\mathcal{S}_y$ 的距离.

3.1 模型计数方法

对于模型计数方法,我们一般将问题 (P_1) 改写为

$$\theta = \arg \min_{\theta \in \Theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(\theta; x, y)], \quad \ell(\theta; x, y) := -\log \left(\sum_{z \in \mathcal{S}_y} P_\theta(z|x) \right), \quad (3)$$

其中 $P_\theta(z|x)$ 表示神经网络模型对 z 的条件概率估计,也称为模型预测非确定性. 不难看出,修改后的损失函数是满足条件概率分布 $P_\theta(z|x)$ 的伪符号接地 z 能够蕴含符号约束 $\mathcal{S}_y$ 的负对数似然. 因此,在有些文献 [86] 中,式 (3) 也被称为蕴含中学习 (learning from entailment). 在该负对数似然损失的计算中,需要穷举符号约束 $\mathcal{S}_y$ 内所有元素,这通常被称为模型计数.

在经典神经符号学习框架中 (例如, DeepProbLog [25] 和 Semantic Loss [26]), 实现模型计数的方案是将符号约束 $\mathcal{S}_y$ 预先编译成对应的二分决策图 (binary decision diagram, BDD) [87, 88] 或者句式决策图 (sentential decision diagram, SDD) [89, 90]. 图 2 以逻辑表达式 $(A \wedge B) \vee (C \wedge D)$ 为例,给出了 BDD 和 SDD 的具体表出形式. 编译完成后,在训练过程中,对于任意 y , 所有的可行符号接地 $z \in \mathcal{S}_y$ 都可以根据决策图快速查询得到. 但是,在实际情况下,受限于决策图的规模大小,决策图的快速查询往往难以实现. 具体而言,决策图的规模取决于 (1) 编译算法的有效性. 尽管这方面已有广泛的研究,提出了一系列启发式约减决策图的方法 [91], 但它们的规模扩展性仍然非常有限. (2) 约束结构的复杂性. 随着变量增多,约束的表示形式变复杂,即使决策图已约减好,规模仍然过大,带来不菲的推理代价.

为了提高决策图的查询效率,许多研究工作关注于利用神经网络模型的预测概率加速决策图的推理. 例如, Scallop [92] 根据模型预测概率选出 top- k 的可行符号接地代替 $\mathcal{S}_y$ 的完全穷举; DPLA* [93] 在 top- k 选择的基础上,引入 A* 算法启发式的搜索这些高概率符号接地结果; A-NeSI [94] 提出训练一

4) 需要注意的是,这里的“模型”并不是指深度学习模型,而是指满足符号约束的解.

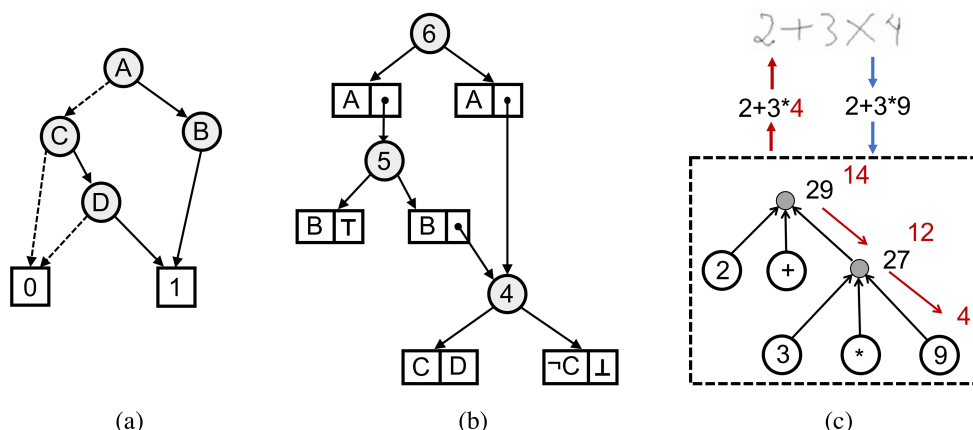


图 2 (网络版彩图) 符号约束的决策图及其上的搜索. 逻辑表达式 $(A \wedge B) \vee (C \wedge D)$ 对应的 (a) 二分决策图和 (b) 句式决策图; (c) 基于 (错误) 模型预测结果 $2 + 3 \times 9 = 29$ 在四则运算语法树上的后向搜索.

Figure 2 (Color online) Decision diagrams and search of symbolic constraints. (a) The binary decision diagram and (b) the sentential decision diagram corresponding to the logical expression $(A \wedge B) \vee (C \wedge D)$; (c) the back-search of the arithmetic expression grammar tree corresponding to the (incorrect) model prediction $2 + 3 \times 9 = 29$.

个新的神经网络模型, 用以直接估计后验概率 $\sum_{z \in \mathcal{S}_y} P_{\theta}(z|x)$. 进一步, NGS^[95] 基于模型预测概率局部展开决策图, 再使用后向搜索 (back-search) 找出可行解. 图 2 中也给出了 NGS 方法在手写公式识别中的一个示例. 其中, $2 + 3 \times 4 = 14$ 被错误地预测成了 $2 + 3 \times 9 = 29$. 此时, NGS 通过局部的展开语法树, 搜索到了最“接近”的修正 $9 \rightarrow 4$, 从而得到了正确的符号接地.

另一类实现模型计数的方案是利用布尔可满足性 (satisfiability, SAT)^[96,97] 以及可满足性模理论 (satisfiability modulo theories, SMT)^[98,99] 求解器. 例如, NeuroASP^[100] 和 SLASH^[101,102] 使用的回答集编程工具 Clingo^[103,104], 其利用 SAT 求解器以及一系列取否操作来实现模型计数^[105]; 除此以外, ABL^[106] 和 ABL-meta^[107] 分别使用无梯度优化工具 RACOS^[108], 以及 Prolog 约束逻辑编程工具 CLP^[109], 实现符号约束解的高效搜索. 另外, SMTLayer^[110] 将后向搜索和最大可满足性模理论 (MaxSAT)^[111,112] 相结合, 基于神经网络模型当前的预测结果, 利用 MaxSAT 求解器搜索可行解.

但以上这两类加速的方案均存在显著缺陷. 一方面, 虽然引入启发式策略能够显著提高效率, 但也在搜索过程中带来了很大的偏差. 另一方面, 尽管目前的约束求解器已经非常高效, 但随着问题结构变得更加复杂, 精确地穷举出所有的接地结果仍然是几乎不可能的. 为此, Li 等^[34] 的近期工作中提出使用投影马尔可夫链蒙特卡洛采样技术^[113,114] (projected Markov Chain Monte Carlo, projected MCMC), 将 MCMC 采样和 SMT 求解器相结合, 实现近似模型计数. 图 3 给出了投影马尔可夫链蒙特卡洛采样的一个例子^[34]. 其中, 模型计数的符号约束为 $\mathcal{S}_y = \{z \mid \text{eval}(z) = y\}$, $z \in \mathcal{Z}$ 表示任意 4 个个位数字以及加减乘除组成的四则运算公式, $\text{eval}(\cdot)$ 表示四则运算的计算函数. 在该例子中计算结果 $y = 42$. 从一个初始可行符号接地开始, 例如 $z = [4; \times; 9; +; 3; +; 3]$. 为了在符号约束中实现 MCMC 采样, 使用投影 $\Pi([z_1; \dots; z_7]) = [z_2; \dots; z_6]$, 即丢弃第一个和最后一个数字. 那么, 当前状态 $z = [4; \times; 9; +; 3; +; 3]$ 会被映射到 $\Pi(z) = [\times; 9; +; 3; +]$. 投影使得符号接地空间具有较好的连通性^[115] (connectivity). 接下来, 随机修改投影空间的一个运算或数字, 例如, 将“9”更新到“8”. 最后, 使用 SMT 求解器实现逆投影, 即计算出剩余的部分 (例如 $4 \times 8 + 3 + 7 = 42$), 从而得到一个新的可行符号接地.

3.2 模糊逻辑方法

不同于模型计数方法将 $\ell(\theta; x, y)$ 定义为所有可行符号接地的负对数似然, 模糊逻辑方法旨在定义一个从模型预测概率 $P_{\theta}(z|x)$ 到符号约束 $\mathcal{S}_y$ 的可微距离. 具体而言, 对于以一阶逻辑形式表达的

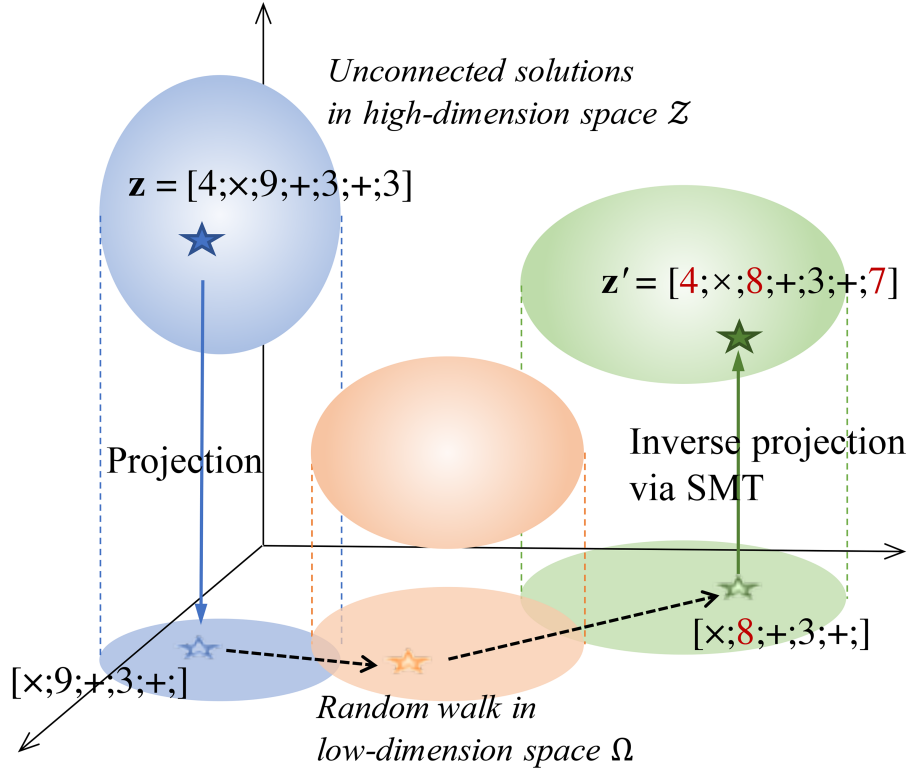


图 3 (网络版彩图) 基于投影的马尔可夫链蒙特卡洛采样技术.

Figure 3 (Color online) Projected MCMC sampling technique.

符号约束 S_y , 模糊逻辑方法根据一套编码规则, 将所有的逻辑原子 (logical atom) 转换为可微隶属度函数 (membership function) [116, 117], 并把逻辑连接词 (logical connective) 和量词 (logical quantifier) 转换为可微运算符. 其中, 逻辑原子的隶属度函数通常直接用神经网络模型的预测概率来计算, 例如, 将 $a := (z = [4; x; 9; +; 3; +; 3])$ 这样的逻辑原子转化为 $\tilde{a} := P_\theta(z = [4; x; 9; +; 3; +; 3] \mid x)$.

对于逻辑连接词的翻译, 已有的神经符号学习方法大多使用基于三角范数 (t-norm) 函数 [118, 119] 的模糊逻辑. 例如, NeuPSL [120] 和 DiffLogic [121] 基于概率软逻辑 (probabilistic soft logic, PSL) [122], 使用 Lukasiewicz 三角范数转换逻辑连接词; 但是这类函数存在梯度消失的问题, 影响后续神经网络模型训练的效率, 因此, DL2 [27] 采用了乘积三角范数⁵⁾, 并将析取连接词转换简化为了简单加法. 但是, 对于包含长合取链的符号约束, 乘积三角范数同样存在梯度消失的问题, 因此 SBR [123] 另外引入了最大/最小三角范数 (也称为 Gödel 三角范数). 综上所述, 将模糊逻辑转换逻辑连接词的方式总结如下:

$$\begin{array}{ll} \text{Lukasiewicz 三角范数} & a_1 \tilde{\wedge} a_2 := \max(0, \tilde{a}_1 + \tilde{a}_2 - 1), \quad a_1 \tilde{\vee} a_2 := \min(1, \tilde{a}_1 + \tilde{a}_2), \\ \text{乘积三角范数} & a_1 \tilde{\wedge} a_2 := \tilde{a}_1 \times \tilde{a}_2, \quad a_1 \tilde{\vee} a_2 := \tilde{a}_1 + \tilde{a}_2, \\ \text{最大/最小三角范数} & a_1 \tilde{\wedge} a_2 := \min(\tilde{a}_1, \tilde{a}_2), \quad a_1 \tilde{\vee} a_2 := \max(\tilde{a}_1, \tilde{a}_2). \end{array}$$

对于逻辑量词, 模糊逻辑方法通常将其转化为最小化/最大化算子, 即

$$\tilde{\forall} u. (u \in U \rightarrow a(u)) := \min_{u \in U} \tilde{a}(u), \quad \tilde{\exists} v. (v \in V \rightarrow a(v)) := \max_{v \in V} \tilde{a}(v).$$

5) 在原始乘积三角范数中, 析取连接词的转化被定义为 $a_1 \tilde{\vee} a_2 := \tilde{a}_1 + \tilde{a}_2 - \tilde{a}_1 \times \tilde{a}_2$. DL2 将其简化成了 $a_1 \tilde{\vee} a_2 := \tilde{a}_1 + \tilde{a}_2$. 该简化使得合取与析取的转化不一致, 但因后续工作均沿用这一修改, 并为简便起见, 我们将 DL2 修改之后的版本记作乘积三角范数.

$$\begin{aligned}
\mathcal{S}_{\mathbf{y}}(\mathbf{z}) = & \underbrace{\bigwedge_{m,n=1,\dots,4} \left(\bigvee_{v=1,\dots,4} ([\mathbf{z}, \mathbf{y}]_{mn} = v) \right)}_{\text{each cell should be in } \{1, \dots, 4\}} \\
& \underbrace{\bigwedge_{m,v=1,\dots,4} \left(\bigvee_{n=1,\dots,4} ([\mathbf{z}, \mathbf{y}]_{mn} = v) \right)}_{\text{each row should fill } \{1, \dots, 4\}} \\
& \underbrace{\bigwedge_{n,v=1,\dots,4} \left(\bigvee_{m=1,\dots,4} ([\mathbf{z}, \mathbf{y}]_{mn} = v) \right)}_{\text{each column should fill } \{1, \dots, 4\}} \\
& \underbrace{\bigwedge_{r,t=1,2} \left(\bigvee_{v=1,\dots,4} \bigwedge_{(m,n) \in B_{rt}} ([\mathbf{z}, \mathbf{y}]_{mn} = v) \right)}_{\text{each block should fill } \{1, \dots, 4\}}
\end{aligned}$$

1	2	3	4
3	4	1	2
2	1	4	3
4	3	2	1

Sudoku matrix $[\mathbf{z}, \mathbf{y}]$, where $\mathbf{z}$ represents the symbol grounding result, and $\mathbf{y}$ denotes the observed results.

$B_{rt} = \{(2r + i, 2t + j) \mid i, j = 1, 2\}$

图4 数独 (4×4) 的符号约束 $\mathcal{S}_{\mathbf{y}}(\mathbf{z})$. 原子命题 $[\mathbf{z}, \mathbf{y}]_{mn} = v$ 表示第 m 行第 n 列的元素取值是否为 v ; 符号约束分别表示每一个元素必须在 $\{1, \dots, 4\}$ 以内, 以及每一行/列/块均必须填满 $\{1, \dots, 4\}$.

Figure 4 Symbolic constraint $\mathcal{S}_{\mathbf{y}}(\mathbf{z})$ of Sudoku (4×4). The atomic proposition $[\mathbf{z}, \mathbf{y}]_{mn} = v$ indicates whether the element in the m -th row and n -th column is equal to v . The symbolic constraint ensures that each element must be within the set $\{1, \dots, 4\}$, and each row, column, and block must contain the numbers $\{1, \dots, 4\}$ exactly once.

在 SBR^[123] 中, 全称量词对应的最小化算子被改成均值计算, 以加速神经网络模型的训练:

$$\tilde{\forall}u.(u \in U \rightarrow a(u)) := \left(\frac{1}{|U|} \sum_{u \in U} \tilde{a}(u) \right).$$

这使得全称量词后的每一个约束都能够被均匀地优化. 在此基础上, LTN^[124, 125] 进一步将一般均值替代为广义均值, 并且对存在量词做出了相应的修改以保持一致:

$$\tilde{\forall}u.(u \in U \rightarrow a(u)) := \left(\frac{1}{|U|} \sum_{u \in U} \tilde{a}(u)^p \right)^{\frac{1}{p}}, \quad \tilde{\exists}v.(v \in V \rightarrow a(v)) := 1 - \left(\frac{1}{|V|} \sum_{v \in V} (1 - \tilde{a}(v))^p \right)^{\frac{1}{p}}.$$

以 4×4 数独任务为例, 使用乘积三角范数, 即 $a_1 \tilde{\wedge} a_2 = \tilde{a}_1 \times \tilde{a}_2$, $a_1 \tilde{\vee} a_2 = \tilde{a}_1 + \tilde{a}_2$, 解释模糊逻辑方法如何定义神经网络模型训练的可微损失函数. 如图4所示, 数独任务中的符号约束可以写成合取范式 (conjunctive normal form, CNF) 的形式, 其中原子命题 $[\mathbf{z}, \mathbf{y}]_{mn} = v$ 表示第 m 行第 n 列的元素取值是否为 v , 其对应的隶属度函数由神经网络的概率表示 $P_{\theta}([\mathbf{z}, \mathbf{y}]_{mn} = v \mid \mathbf{x})$. 然后, 模糊逻辑方法将该符号规则转化为如下损失函数:

$$\begin{aligned}
\ell(\theta; \mathbf{x}, \mathbf{y}) = & \underbrace{\prod_{m,n=1,\dots,4} \left(\sum_{v=1,\dots,4} P_{\theta}([\mathbf{z}, \mathbf{y}]_{mn} = v \mid \mathbf{x}) \right)}_{\text{每一格必须在 } 1 \sim 4 \text{ 之内}} \times \underbrace{\prod_{m,v=1,\dots,4} \left(\sum_{n=1,\dots,4} P_{\theta}([\mathbf{z}, \mathbf{y}]_{mn} = v \mid \mathbf{x}) \right)}_{\text{每一行必须填满 } 1 \sim 4} \\
& \times \underbrace{\prod_{n,v=1,\dots,4} \left(\sum_{m=1,\dots,4} P_{\theta}([\mathbf{z}, \mathbf{y}]_{mn} = v \mid \mathbf{x}) \right)}_{\text{每一列必须填满 } 1 \sim 4} \times \underbrace{\prod_{r,t=1,2,v=1,\dots,4} \left(\sum_{(m,n) \in B_{rt}} P_{\theta}([\mathbf{z}, \mathbf{y}]_{mn} = v \mid \mathbf{x}) \right)}_{\text{每一块必须填满 } 1 \sim 4}, \\
& B_{rt} = \{(2r + i, 2t + j) \mid i, j = 1, 2\}.
\end{aligned} \tag{4}$$

此外, 还有一系列工作研究如何加速模糊逻辑转化后的损失函数计算, 这些工作都尝试引入更多的并行化计算, 以加速模糊逻辑方法的推理. 例如, DeepStochLog^[126], ENT^[127] 和 SPL^[128] 引入概率 AND-OR 环路 (circuits)^[129] 来实现模糊逻辑计算的并行化, 并且将整个环路作为计算图直接嵌入神经网络模块, 以支持统一的前向推理和反向传播. 图5给出了一个将逻辑表达式 $(A \wedge B) \rightarrow C$ 编码为 AND-OR 环图的示例. 此外, STE^[130] 引入了一个额外的状态矩阵, 将式(4)中的连乘计算消解成矩阵的乘积和点积, 利用 GPU 并行加速计算; 但状态矩阵的大小随问题变量的规模指数级增长, 为解决

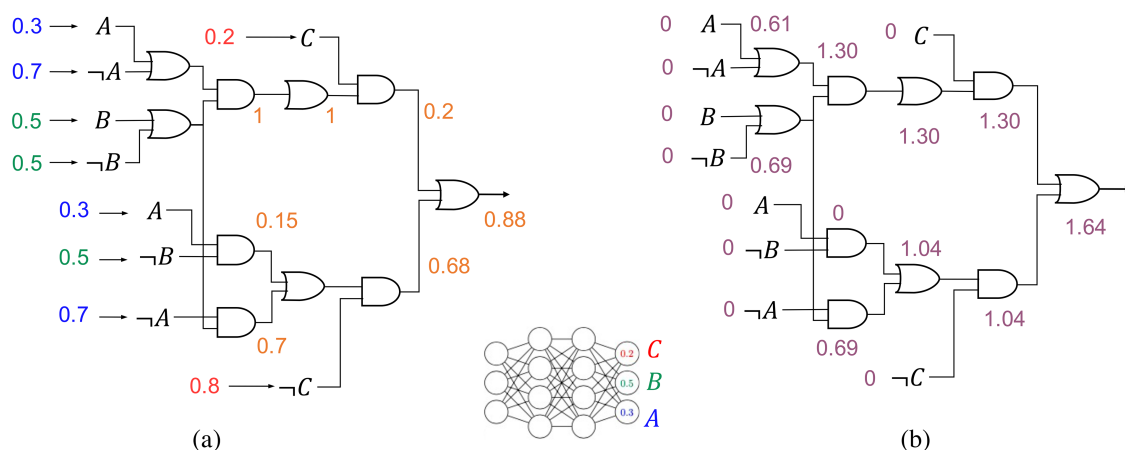


图5 (网络版彩图) 概率 AND-OR 环图及其对应的信息熵. 逻辑表达式 $(A \wedge B) \rightarrow C$ 对应的 (a) 概率 AND-OR 环图, 其中逻辑原子 A, B, C 的隶属度由神经网络预测概率 $P_{\theta}(z|x)$ 决定; 基于概率环图, 我们可以递归地计算该模糊逻辑表达式的 (b) 信息熵.

Figure 5 (Color online) Probabilistic AND-OR circuit and information entropy. (a) The probabilistic AND-OR circuit of the logical expression $(A \wedge B) \rightarrow C$, where the membership degrees of the logical atoms A, B , and C are determined by the neural network prediction $P_{\theta}(z|x)$; then, (b) the corresponding information entropy is recursively computed based on the probabilistic circuit.

该问题, LogicMP^[131] 将状态矩阵隐式地聚合在 Einsum 操作中, 这既节约了 GPU 存储, 同时能够在大规模任务中实现快速推理.

3.3 小结

总体上说, 模糊逻辑方法与模型计数方法均将网络的预测非确定性建模到了神经符号接地过程中, 以提高神经网络模块训练的效率. 但实际应用上, 模糊逻辑方法与模型计数方法各有优劣. 对于模糊逻辑方法, 一方面, 模糊逻辑避免了可行符号接地的穷举问题, 因而大幅提高了训练的效率. 但另一方面, 模糊逻辑并没有完全保证符号约束的精确性, 因而可能影响最终的系统性能. 例如, 当前的模糊逻辑方法均不能保证逻辑一致性^[132]. 以 $(a \vee a) \rightarrow a$ 为例, 理想情况下, 模糊逻辑应该将该重言式转化为常值 1, 但实际上, 上述 3 种三角范数函数均不能保证该性质. 而模型计数方法并不会丢失符号约束的精确性, 并且理论上更符合第一性原理^[26]. 但是, 由于训练过程中每一步梯度下降的计算都涉及可行符号接地的穷举, 因此其训练效率远不如模糊逻辑方法.

除以上权衡以外, 模型计数方法和模糊逻辑方法均存在另一个相同的问题, 即无法有效地避免推理捷径. 我们在第 4 节中给出详细的分析.

4 推理捷径缺陷

无论是模型计数方法还是模糊逻辑方法, 其训练目标都是将模型的预测结果限制在符号约束之内. 这里的一个关键问题是, 给定存在多个可行符号接地的符号约束, 如何保证模型在训练过程中学得那些符合预期的符号接地结果. 已有理论研究和实验结果均表明, 神经网络模型能够拟合任意的训练标签. 现实的情况是, 在神经符号接地任务中, 总存在某些推理捷径, 即该符号接地结果是可行的, 并且很容易被网络学得, 但却不是实际所期望的符号接地结果, 也不具有良好的泛化性能. Li 等^[33] 的研究工作中首先明确提出并研究了这一问题, 此后一系列相关工作^[79~81, 133] 进行了进一步的讨论.

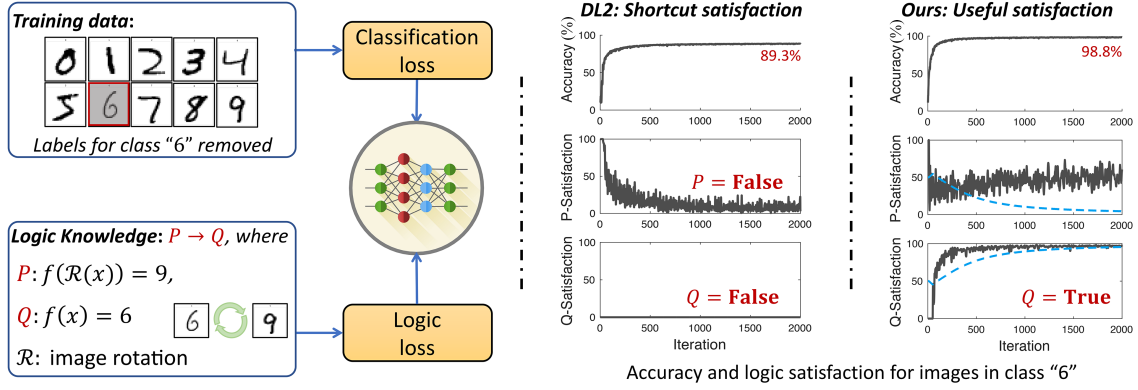


图 6 (网络版彩图) 手写识别的半监督学习分类任务. 我们删除训练图像中的“6”类标签, 但引入符号推理规则 $P := (f(\mathcal{R}(x)) = 9) \rightarrow Q := (f(x) = 6)$ 来预测“6”, 其中 $\mathcal{R}(x)$ 代表将图像 x 旋转 180° . 对于“6”类而言, 存在两种可行符号接地: (1) $P = F$, 将旋转图片 $\mathcal{R}(x)$ 识别成非“9”结果; (2) $(P, Q) = (T, T)$, 将旋转后图片 $\mathcal{R}(x)$ 识别成“9”, 并且将原始图片 x 识别成“6”. 理想的符号接地结果应该是 $(P, Q) = (T, T)$; 然而, 已有的模型计数方法和模糊逻辑方法都会通过“捷径” ($P = F$) 来满足规则, 导致“6”类图片几乎都分类错误.

Figure 6 (Color online) Semi-supervised learning classification task for handwritten digit recognition. We remove the labels of training images in class “6”, but introduce a logical rule $P := (f(\mathcal{R}(x)) = 9) \rightarrow Q := (f(x) = 6)$ to predict “6”, where $\mathcal{R}(x)$ stands for rotating the image x by 180° . For the “6” class, there are two possible symbol grounding outcomes. The ideal symbol grounding outcome should be $(P, Q) = (T, T)$; however, existing model counting methods and fuzzy logic methods often take the shortcut $P = F$ to satisfy the rule, resulting in nearly all images of “6” being incorrectly classified.

4.1 示例构造

先构造一个半监督手写数字识别任务解释推理捷径问题^[33]. 在该任务中, 移除训练数据中所有“6”类图像的标签, 但引入了一条简单的符号推理规则进行代替: $P := (f(\mathcal{R}(x)) = 9) \rightarrow Q := (f(x) = 6)$, 即“如果旋转 180° 之后的图片被识别为‘9’, 那么该图片应该被识别为‘6’”. 损失函数以及学习曲线如图 6 所示. 具体而言, 对于所有图片, 采用神经符号学习的方式, 即利用该符号规则对应的模型计数损失或模糊逻辑损失进行训练. 而对于非“6”类图片, 由于有标签, 可以使用通常监督学习方法的分类损失.

理想情况下, 虽然缺少“6”类的图片标注, 但神经网络模型能够正确分类“9”类图片, 因此通过符号规则 $P \rightarrow Q$ 理应能够学得“6”类图片的分类. 然而, 已有的神经符号学习框架大多不能够得到该预期结果. 这是因为, 此时有两种可行符号接地: (1) $P = F$, 将旋转图片 $\mathcal{R}(x)$ 识别成非“9”结果; (2) $(P, Q) = (T, T)$, 将图片 x 识别成“6”. 显而易见, 相比较于后者, 前者是更容易被满足的符号接地. 因此, 模型更容易利用该“捷径”推理: 将“6”类图片的旋转识别为非“9”结果.

将推理捷径问题定义为: 深度学习模型错误地拟合到不符合预期 (semantically unintended) 的符号接地结果, 引起神经符号系统的性能下降. 当前, 已经有一些研究分析推理捷径发生的原因及其解决方案. 比较主流的解释是: 模型计数方法和模糊逻辑方法中的一些近似计算引入了一些不合理的归纳偏好, 导致训练过程最终收敛到了错误的符号接地结果^[79, 81, 133]. 此外, NeSy-RS^[80] 从数据、模型架构, 以及损失函数多个角度出发, 提供了一系列解决方案. 总结为两点: (1) 增加领域知识和数据量, 提高数据覆盖率; (2) 引入自编码器损失与解耦架构, 帮助区分不同概念.

但这些理论解释与解决方法的有效性仍然存疑. 一方面, 在我们构造的半监督学习示例中, 即使没有近似计算, 神经网络模型仍然会陷入推理捷径之中; 另一方面, 尽管 NeSy-RS 中提出的一系列解决方案能够有效改善神经符号系统的性能, 但其实验同样表明, 这些策略并不能够从根本上解决推理捷径问题. 第 4.2 小节从非确定性的视角分析与解释推理捷径发生的原因.

4.2 模型计数方法的推理捷径

首先, 考虑模型计数方法, 其对应的损失函数 (及其梯度) 可以写为

$$\ell(\boldsymbol{\theta}; \mathbf{x}, \mathbf{y}) = -\log \left(\sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) \right), \quad \nabla \ell(\boldsymbol{\theta}; \mathbf{x}, \mathbf{y}) = -\frac{1}{\sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})} \left(\sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} \nabla P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) \right).$$

接下来, 引入一个额外的参数分布 $Q_{\boldsymbol{\vartheta}}(\mathbf{z})$ 将符号接地“软化”, 用以显示刻画符号接地的非确定性. 并且, 记 $H(Q) := -\sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} Q_{\boldsymbol{\vartheta}}(\mathbf{z}) \log Q_{\boldsymbol{\vartheta}}(\mathbf{z})$ 为分布 $Q_{\boldsymbol{\vartheta}}$ 的信息熵 (entropy), $\text{supp}(Q_{\boldsymbol{\vartheta}})$ 为分布 $Q_{\boldsymbol{\vartheta}}$ 的支撑集 (support set). 那么, 有如下定理.

定理1 (模型计数方法的非确定性控制) 给定原始输入 $\mathbf{x}$, 以及最终输出 $\mathbf{y}$, 对于模型计数的损失函数 $\ell(\boldsymbol{\theta}; \mathbf{x}, \mathbf{y})$, 其最小化问题等价于 ($\gamma = 1$)

$$\begin{aligned} \min_{\boldsymbol{\theta}, \boldsymbol{\vartheta}} \quad & \hat{\ell}(\boldsymbol{\theta}, \boldsymbol{\vartheta}) := \sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} (-Q_{\boldsymbol{\vartheta}}(\mathbf{z}) \log P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) + \gamma Q_{\boldsymbol{\vartheta}}(\mathbf{z}) \log Q_{\boldsymbol{\vartheta}}(\mathbf{z}), \\ \text{s.t.} \quad & \text{supp}(Q_{\boldsymbol{\vartheta}}) \subseteq \mathcal{S}_{\mathbf{y}}. \end{aligned}$$

也就是说, 这两个最小化问题对于 $\boldsymbol{\theta}$ 有相同的最优解, 并且有相同的梯度下降动态.

证明 先考虑梯度 $\nabla \ell(\boldsymbol{\theta})$, 可以将其改写为

$$\nabla \ell(\boldsymbol{\theta}) = -\sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} \frac{1}{\sum_{\mathbf{z}' \in \mathcal{S}_{\mathbf{y}}} P_{\boldsymbol{\theta}}(\mathbf{z}'|\mathbf{x})} \nabla P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = -\sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} \frac{P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}{\sum_{\mathbf{z}' \in \mathcal{S}_{\mathbf{y}}} P_{\boldsymbol{\theta}}(\mathbf{z}'|\mathbf{x})} \nabla \log P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}).$$

对于优化问题 $\min_{\boldsymbol{\theta}, \boldsymbol{\vartheta}} \hat{\ell}(\boldsymbol{\theta}, \boldsymbol{\vartheta})$, 给定 $\boldsymbol{\theta}$ 时, 最优解 $\boldsymbol{\vartheta}^*$ 满足

$$Q_{\boldsymbol{\vartheta}^*}(\mathbf{z}) = \frac{P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}{\sum_{\mathbf{z}' \in \mathcal{S}(\mathbf{y})} P_{\boldsymbol{\theta}}(\mathbf{z}'|\mathbf{x})}.$$

因此 $\nabla_{\boldsymbol{\theta}} \hat{\ell}(\boldsymbol{\theta}, \boldsymbol{\vartheta}^*)$ 可以被重写为

$$\nabla_{\boldsymbol{\theta}} \hat{\ell}(\boldsymbol{\theta}, \boldsymbol{\vartheta}^*) = -\sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} Q_{\boldsymbol{\vartheta}^*}(\mathbf{z}) \nabla \log P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = -\sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} \frac{P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}{\sum_{\mathbf{z}' \in \mathcal{S}_{\mathbf{y}}} P_{\boldsymbol{\theta}}(\mathbf{z}'|\mathbf{x})} \nabla \log P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}).$$

在此简单补充一下, $Q_{\boldsymbol{\vartheta}^*}$ 是 $\min_{\boldsymbol{\vartheta}} \ell(\boldsymbol{\theta}, \boldsymbol{\vartheta})$ 在给定 $\boldsymbol{\theta}$ 时最优解的证明. 具体而言, 内问题的拉格朗日 (Lagrange) 函数可以写为

$$\mathcal{L}(\boldsymbol{\vartheta}; \lambda) = \sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} Q_{\boldsymbol{\vartheta}}(\mathbf{z}) \log \left(\frac{Q_{\boldsymbol{\vartheta}}(\mathbf{z})}{P_{\boldsymbol{\theta}}(\mathbf{z})} \right) - \lambda \left(\sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} Q_{\boldsymbol{\vartheta}}(\mathbf{z}) - 1 \right).$$

计算关于 $Q_{\boldsymbol{\vartheta}}(\mathbf{z})$ 的梯度, 并令其为 0, 那么

$$\log Q_{\boldsymbol{\vartheta}^*}(\mathbf{z}) + 1 - \log P_{\boldsymbol{\theta}}(\mathbf{z}) - \lambda = 0$$

对任意 $\mathbf{z} \in \mathcal{S}_{\mathbf{y}}$ 成立. 因此, 有

$$Q_{\boldsymbol{\vartheta}^*}(\mathbf{z}) = e^{\lambda-1} P_{\boldsymbol{\theta}}(\mathbf{z}), \quad \sum_{\mathbf{z} \in \mathcal{S}_{\mathbf{y}}} e^{\lambda-1} P_{\boldsymbol{\theta}}(\mathbf{z}) = 1.$$

将这两个等式结合, 最终得到

$$Q_{\boldsymbol{\vartheta}^*}(\mathbf{z}) = \frac{P_{\boldsymbol{\theta}}(\mathbf{z})}{\sum_{\mathbf{z}' \in \mathcal{S}_{\mathbf{y}}} P_{\boldsymbol{\theta}}(\mathbf{z}')}.$$

根据该定理, 并利用系数 $\gamma = 1$, 可以转而考察最小化 KL 散度问题: $\min_{\theta, \vartheta} \hat{\ell}(\theta, \vartheta) := \text{KL}(Q_{\vartheta} \| P_{\theta})$, 并且有支撑集约束 $\text{supp}(Q_{\vartheta}) \subseteq \mathcal{S}_y$. 其中, P_{θ} 较为简单, 其最小化的目标是拟合对偶分布 Q_{ϑ} ; 而对于 Q_{ϑ} , 利用 KL 散度分解 $\text{KL}(Q_{\vartheta}, P_{\theta}) = H(Q_{\vartheta}, P_{\theta}) - H(Q_{\vartheta})$, 除支撑集约束 $\text{supp}(Q_{\vartheta}) \subseteq \mathcal{S}_y$ 以外, 有以下两个部分进行非确定性管理:

- (1) 交叉熵 $H(Q_{\vartheta}, P_{\theta}) = \sum_{z \in \mathcal{S}_y} (-Q_{\vartheta}(z|x) \log P_{\theta}(z|x))$. 鼓励 Q_{ϑ} 收敛到独热 (one-hot) 分布;
- (2) 负信息熵 $-H(Q_{\vartheta}) = \sum_{z \in \mathcal{S}_y} Q_{\vartheta}(z) \log Q_{\vartheta}(z)$. 熵正则化控制 Q_{ϑ} 的尖锐度.

基于以上分解, 可以解释推理捷径发生的原因: 首先, 对于推理捷径所对应的符号接地结果, 神经网络模型的预测概率较大. 例如, 在前面构造的半监督学习任务中, 推理捷径 (即旋转之后不是数字“9”) 的预测概率 $P_{\theta}(f(\mathcal{R}(x)) \neq 9|x)$ 远大于正确符号接地 (即旋转后图片是“9”, 并且原始图片是“6”) 的预测概率 $P_{\theta}(f(\mathcal{R}(x)) = 9 \wedge f(x) = 6|x)$. 在这样的情况下, 缺少非确定性管理导致 (1) 交叉熵过度强化了 Q_{ϑ} , 使其快速收敛到了推理捷径对应的符号接地. 然而, (2) 信息熵的惩罚系数 ($\gamma = 1$) 太小, 不足以控制 Q_{ϑ} 的尖锐度, 因此难以摆脱推理捷径.

实际上, 绕开推理捷径, 以及实现正确的符号接地, 二者是完全一致的. 具体而言, 在未知正确符号接地的情况下, 为了绕开推理捷径, 必须先将 Q_{ϑ} 约束成 $\mathcal{S}_y$ 上的均匀分布 (即降低交叉熵, 提高负信息熵), 以降低不同符号接地概率之间的差异; 而在找到正确符号接地之后, 同样需要将 Q_{ϑ} 的分布密度更多导向正确结果 (即提高交叉熵, 降低负信息熵), 从而确保符号接地过程的最终收敛. 因此, 可以说, 整个符号接地过程的核心就是管理非确定性, 以确保符号接地的表达力以及最终的效果.

4.3 模糊逻辑方法的推理捷径

类似地, 对于模糊逻辑方法, 同样可以将其与对偶分布 Q_{ϑ} 上的非确定性控制问题等价. 具体地说, 有如下定理成立.

定理2 (模糊逻辑方法的非确定性控制) 给定原始输入 x , 以及最终输出 y , 对于三角范数模糊逻辑的损失函数 $\ell(\theta; x, y)$, 其最小化问题等价于

$$\begin{aligned} \min_{\theta, \vartheta} \quad & \hat{\ell}(\theta, \vartheta) := \sum_{z \in \mathcal{S}_y} (-Q_{\vartheta}(z) \log P_{\theta}(z|x)) + \gamma Q_{\vartheta}(z) \log Q_{\vartheta}(z), \\ \text{s.t.} \quad & \text{supp}(Q_{\vartheta}) \subseteq \mathcal{S}_y. \end{aligned}$$

也就是说, 这两个最小化问题关于 θ 有相同的最优解, 并且有相同的梯度下降动态. 此外, 3 种不同的三角范数分别对应常系数 $\gamma = 1, 1, 0$.

证明 我们可以将符号约束 $\mathcal{S}_y$ 重写为析取式 $\vee_{z' \in \mathcal{S}_y} (z = z')$. 那么, 模糊逻辑方法中 3 种三角范数对应的损失函数分别为

$$\text{Lukasiewicz 三角范数} \quad \max_{\theta} \ell(\theta; x, y) := \min \left(1, \sum_{z \in \mathcal{S}_y} P_{\theta}(z|x) \right) = \sum_{z \in \mathcal{S}_y} P_{\theta}(z|x) \quad (5)$$

$$\Rightarrow \min_{\theta} \ell(\theta; x, y) = -\log \left(\sum_{z \in \mathcal{S}_y} P_{\theta}(z|x) \right), \quad (6)$$

$$\text{乘积三角范数} \quad \max_{\theta} \ell(\theta; x, y) := \sum_{z \in \mathcal{S}_y} P_{\theta}(z|x) \quad (7)$$

$$\Rightarrow \min_{\theta} \ell(\theta; x, y) = -\log \left(\sum_{z \in \mathcal{S}_y} P_{\theta}(z|x) \right), \quad (8)$$

$$\text{最大/最小三角范数} \quad \max_{\theta} \ell(\theta; x, y) := \max_{z \in \mathcal{S}_y} (P_{\theta}(z|x)) \quad (9)$$

$$\Rightarrow \min_{\theta} \ell(\theta; \mathbf{x}, \mathbf{y}) = \min_{\mathbf{z} \in S_{\mathbf{y}}} -\log(P_{\theta}(\mathbf{z}|\mathbf{x})), \quad (10)$$

其中, 引入负对数变换, 将最大化问题转化为最小化问题. 不难看出, Lukasiewicz 三角范数与乘积三角范数的损失函数就是模型计数的损失函数; 而最大/最小三角范数的损失函数等价于 $\gamma = 0$, 即没有熵正则化项, Q_{θ} 直接收敛到独热分布的情形.

总体而言, 与模型计数方法一致, 模糊逻辑方法同样很容易陷入推理捷径. 发生推理捷径的根本原因同样是对 $P_{\theta}(\mathbf{z}|\mathbf{x})$ 以及 $Q_{\theta}(\mathbf{z})$ 的信息熵控制不够, 限制了符号接地的表达力.

4.4 神经符号学习中的非确定性管理

从上文的讨论中可以看到, 负信息熵的控制神经符号学习中至关重要. 如果负信息熵的惩罚太弱, 那么符号接地很容易陷入推理捷径, 因而收敛到不符合预期的结果; 但是, 如果负信息熵的惩罚太强, 那么符号接地又难以收敛到确定的结果, 因而影响性能. 由于负信息熵是符号接地分布 P_{θ} 以及 Q_{θ} 的非确定性度量, 因此也将其称为“神经符号学习的非确定性管理”(uncertainty management).

深度学习领域中, 已经有许多关于非确定性管理的研究, 其主要目标是信心校准^[134]: 对于目标符号接地 $\mathbf{z}$, 基于预测概率 $P_{\theta}(\mathbf{z}|\mathbf{x})$, 将信心区间 $[0, 1]$ 划分为 M 个子区间:

$$S_m = \{\mathbf{x} \mid a_m < P_{\theta}(\mathbf{z}|\mathbf{x}) \leq a_{m+1}\}, \quad m = 0, \dots, M.$$

信心校准要求, 对于分布 $\mathcal{D}$ 上, 并且落在 S_m 内的样本, 其预测概率 $P_{\theta}(\mathbf{z}|\mathbf{x})$ 的期望等于准确率:

$$\mathbb{E}_{\mathbf{x} \sim \mathcal{D}, \mathbf{x} \in S_m} [P_{\theta}(\mathbf{z}|\mathbf{x})] = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}, \mathbf{x} \in S_m} [\mathbb{I}(f_{\theta}(\mathbf{x}) = \mathbf{z})],$$

其中 $\mathbb{I}(\cdot)$ 为示性函数, 即当 $f_{\theta}(\mathbf{x}) = \mathbf{z}$ 时取值为 1, 否则为 0. 信心校准的目的是提高深度神经网络的可靠性, 例如, 在天气预报的预测结果里, 信心校准要求所有预测明天下雨概率等于 80% 的场景中, 真正下雨的次数占比约为 80%. 这使得用户能够信赖模型的预测结果.

对于深度神经网络而言, 已有的工作表明, 模型往往预测信心过高 (overconfident). 这是因为, 神经网络模型在训练过程中往往通过牺牲信心校准性能, 来获取更高的准确率^[31]. 因此, 许多研究工作试图在训练过程中添加额外正则化项, 以防止模型的“过拟合”, 主要包括标签光滑化^[135]、间隔标签光滑^[136]、Mixup 增广^[137]、AugMix 增广^[138]、焦点损失^[32]、Logit 正则化^[139, 140]、成对约束损失^[141], 以及能量模型^[142~144]等. 可以说, 这些工作均显式地或隐式地将信息熵正则化引入损失函数中, 我们以最经典的标签光滑化和焦点损失为例进行说明.

(1) 标签光滑化将 K -分类交叉熵损失中的独热标签 $\mathbf{y}$ 软化, 用 $\tilde{\mathbf{y}} = (1 - \varepsilon)\mathbf{y} + \varepsilon/K$ 代替. 令 $\mathcal{L}_{\text{CE}}$ 表示交叉熵损失, 那么标签光滑化得到的损失函数为

$$\mathcal{L}_{\text{LS}}(\tilde{\mathbf{y}}, P_{\theta}(\mathbf{y}|\mathbf{x})) = (1 - \varepsilon)\mathcal{L}_{\text{CE}}(\mathbf{y}, P_{\theta}(\mathbf{y}|\mathbf{x})) + \varepsilon\mathcal{L}_{\text{CE}}(\mathbf{u}, P_{\theta}(\mathbf{y}|\mathbf{x})), \quad \mathbf{u} = [1/K, \dots, 1/K].$$

标签光滑化等价于添加了一个最大熵惩罚项, 鼓励模型预测接近均匀分布, 以防止模型信心偏高.

(2) 焦点损失 $\mathcal{L}_{\text{focal}}(\mathbf{y}, P_{\theta}(\mathbf{y}|\mathbf{x})) = (1 - P_{\theta}(\mathbf{y}|\mathbf{x}))^{\gamma} \mathcal{L}_{\text{CE}}(\mathbf{y}, P_{\theta}(\mathbf{y}|\mathbf{x}))$ 最初是为了解决目标检测中的类别不平衡问题而提出的^[145]. 但近期, 有研究者指出, 焦点损失满足以下性质:

$$\mathcal{L}_{\text{focal}}(\mathbf{y}, P_{\theta}(\mathbf{y}|\mathbf{x})) \geq \mathcal{L}_{\text{CE}}(\mathbf{y}, P_{\theta}(\mathbf{y}|\mathbf{x})) - \gamma H(P_{\theta}(\mathbf{y}|\mathbf{x})).$$

因此, 焦点损失隐式地在交叉熵损失上添加了最大熵正则化项, 缓解模型输出信心偏高的问题.

已有实验研究还表明, 信心校准与网络性能之间是权衡 (trade-off) 关系, 并且神经网络在一般监督学习中通常会以牺牲校准性的方式来提高准确率^[31, 32]. 但是, 正如前面所分析的, 在神经符号接地中, 信心过高会引发推理捷径, 降低模型性能, 因此对于神经符号系统而言, 二者并没有明显的权衡关

系. 因此, 可以说, 非确定性管理在神经符号学习中更为重要: 它既解决推理捷径问题, 提高神经符号系统的性能; 也实现模型信心校准, 提高网络模型本身的可靠性.

这些用于信心校准的熵正则化的方法同样可以用在神经符号接地问题中. 例如, ENT^[127] 和 SS^[146] 在乘积三角范数的基础上, 递归地引入了合取运算符与析取运算符的信息熵正则化以及互信息增强. 以 ENT 为例, 其中合取表达式 $a_1 \wedge a_2$ 和析取表达式 $a_1 \vee a_2$ 的信息熵分别被计算为

$$\begin{aligned}\text{Ent}(a_1 \wedge a_2) &= \text{Ent}(a_1) + \text{Ent}(a_2), \\ \text{Ent}(a_1 \vee a_2) &= \tilde{a}_1 \log \tilde{a}_1 + \tilde{a}_2 \log \tilde{a}_2 + \tilde{a}_1 \text{Ent}(a_1) + \tilde{a}_2 \text{Ent}(a_2).\end{aligned}$$

此外, LwRS^[33] 引入对偶变量 τ , 将最大/最小三角范数改进为

$$a_1 \tilde{\wedge} a_2 := \min_{\tau_1, \tau_2} \{ \tau_1 \tilde{a}_1 + \tau_2 \tilde{a}_2 \mid \tau_1 + \tau_2 = 1 \}, \quad a_1 \tilde{\vee} a_2 := \max_{\tau_1, \tau_2} \{ \tau_1 \tilde{a}_1 + \tau_2 \tilde{a}_2 \mid \tau_1 + \tau_2 = 1 \}.$$

对偶变量 τ 与前面定理中的对偶分布 Q_θ 是一致的. 再通过对偶梯度上升, 利用梯度下降的步长隐式地控制 Q_θ 的信息熵. 进一步地, Li 等^[34] 的工作直接从原始优化问题出发:

$$\begin{aligned}\min_{\theta, \vartheta} \quad & \hat{\ell}(\theta, \vartheta) := \sum_{z \in \mathcal{S}_y} (-Q_\vartheta(z) \log P_\theta(z|x)) + \gamma Q_\vartheta(z) \log Q_\vartheta(z), \\ \text{s.t.} \quad & \text{supp}(Q_\vartheta) \subseteq \mathcal{S}_y.\end{aligned}$$

将模拟退火策略与 projected MCMC 相结合, 在神经网络模型训练过程中缓慢降低 γ 的值. 这确保神经符号学习能够在训练初期, 探索足够多的可行符号接地, 而随着训练损失函数的下降, 逐渐收敛到确定的符号接地结果. 值得一提的是, 当前最新的理论研究指出^[81], 符号接地空间的不连通性也是加剧推理捷径发生的原因. 而 Li 等^[34] 的工作中使用投影技术提高了符号接地空间的连通性, 与该理论结果一致, 这可能是其能够有效摆脱推理捷径的另一原因.

4.5 小结

本节通过一个简单统一的理论框架解释了 (基于模型计数方法和模糊逻辑方法的) 神经符号接地过程中推理捷径发生的原因: 缺乏神经网络的有效非确定性管理, 导致训练过程容易收敛到那些信心过高, 不符合预期的符号接地结果. 紧接着, 分析一般深度学习中的非确定性管理, 并且将其与神经符号学习中的非确定性管理联系: 前者是为了信心校准, 即确保预测概率等于真实发生概率; 而后者不仅仅为了信心校准, 也为了避免推理捷径. 并且, 从信息熵正则化的角度, 我们指出两种非确定性管理问题与方法的一致性, 解释了当前各种缓解神经符号接地中推理捷径问题技术的合理性.

尽管神经符号学习中的推理捷径非常普遍^[80], 但单独针对推理捷径的研究仍然非常有限. 一方面, 从方法上讲, 已有方法均在探索如何通过信息熵实现更好的非确定性管理, 但其他可行方案 (例如, 通过引入因果模型^[147, 148]) 缺乏进一步探索; 此外, 信息熵正则化的系数 γ 很难确定. 正如前文所说, 太小的 γ 正则化效果不够, 难以避免推理捷径; 而太大的 γ 影响训练效率, 甚至导致符号接地难以收敛. 因此, 一个高效的并且动态调整信息熵正则化的方案同样需要研究. 另一方面, 从理论上说, 推理捷径的产生原因仍然缺乏完整理论分析^[81]. 而且, 尽管在神经网络的监督学习领域, 使用最大熵正则化实现信心校准的应用非常广泛, 但仍然缺少理论支撑^[149]. 因此, 最大熵正则化对于神经符号学习中非确定性管理的有效性同样缺少更深层次的理论结果.

5 可微符号约束合成

本节关注符号演绎模块的学习问题:

$$\phi = \arg \min_{\phi \in \Phi} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\ell_2(g_\phi(\mathbf{z}), \mathbf{y})], \quad \text{s.t.} \quad \mathbf{z} = f(\mathbf{x}), \quad \mathbf{z} \in \mathcal{Z}. \quad (\text{P}_2)$$

该问题中, 参数空间 Φ 通常表示一系列候选逻辑规则, 往往是离散的, 因此不能进行梯度计算, 更无法执行梯度下降. 实际上, 在早期机器学习文献中^[3], 符号演绎模块的建模与训练可以分为“归纳逻辑编程”(inductive logic programming, ILP)^[19]和“统计关系学习”(statistical relational learning, SRL)^[150]这两种方法. 其中, 归纳逻辑编程利用一阶霍尔子句 $(a_1 \wedge \dots \wedge a_n) \rightarrow c$ 组成的集合来表达符号推理规则; 其训练过程主要是对被合取逻辑原子的搜索^[151]. 统计关系学习通常使用概率图模型, 例如, Bayesian 网络, Markov 随机场, 以及 (隐) Markov 模型等, 来建模逻辑原子之间的因果或依赖关系; 其训练过程通常是 Monte Carlo 采样或者变分推断来近似优化最大似然^[152,153]. 可以看出, 这些经典机器学习方法的建模以及训练大多都不支持梯度下降法, 这不仅阻碍了符号推理模块与神经网络模块的联合训练, 也极大降低了大量数据下符号演绎模块本身的训练效率.

因此, 近期大量神经符号学习的研究关注于如何将符号推理模块的可微化, 以方便使用梯度下降法进行训练. 我们将其统一称为“可微符号约束合成”. 总的来说, 这些方法都可以理解为: 为所有预先定义好的候选符号约束赋予非确定性系数, 再使用梯度下降法优化该系数. 但符号约束的结构非常多样, 不同的结构需要不同的非确定性建模方法, 以保证最终的效率和效果. 接下来, 我们将简要总结常用的符号约束结构及其对应的建模方法.

5.1 霍尔子句结构

首先, 一类典型的工作就是在已有的归纳逻辑编程框架上, 引入非确定性系数来实现候选符号约束选择. 例如, 考虑到 ILP 学得的目标逻辑规则为 m 条霍尔子句 $(a_{i1} \wedge \dots \wedge a_{in}) \rightarrow c_i, i = 1, \dots, m$ 组成的集合, ∂ILP ^[154] 给所有的候选霍尔子句分配了相应权重 $w_1, \dots, w_m$, 利用梯度下降优化, 选择权重最高的一个或多个逻辑规则作为最终符号约束. 但为了保证效率, ∂ILP 将霍尔子句中逻辑原子的个数严格限制为 2, 以减少候选符号约束的规模. 这也降低了整体框架的表达力: 当目标符号约束较为复杂时, 通常需要引入更高阶的逻辑原子, 以确保目标约束能够被候选符号约束集合所覆盖.

为了解决 ∂ILP 的表达力不足问题, DILP-ST ^[155] 和 αILP ^[156] 均放宽了对候选符号约束的限制, 并提出使用束搜索 (beam-search) 来启发式地选择能够提高推理性能的霍尔子句, 以组成候选符号约束. 另外, DiffLog ^[157] 提出结合用户提供的先验知识, 使用主动学习中的委员会查询策略^[158], 以交互式的方式构建候选符号约束. 而 ∂ILP 的另一个问题是大批量数据下的训练效率较低. 这是因为随着候选符号约束的增多, 训练时符号推理的代价往往非常高. 鉴于此, 一系列研究基于知识图谱推理, 将一阶逻辑推理 $a_1 \wedge a_2 \rightarrow c$ 由稀疏矩阵乘法 $\mathbf{A}_1 \cdot \mathbf{A}_2 \approx \mathbf{C}$ 近似. 例如, NeuralLP ^[159] 定义了 TensorLog 运算^[160]: 对于独热编码的输入向量 $\mathbf{z}$, 定义 $a_1(y, u)$ 和 $a_2(u, z)$ 对应的关系矩阵为 $\mathbf{A}_1$ 和 $\mathbf{A}_2$, 那么 $\mathbf{s} = \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \mathbf{z}$ 实现了链式结构推理 $a_1(y, u) \wedge a_2(u, z) \rightarrow c(y, z)$, 其中 $\mathbf{s}$ 表示了满足推理规则的 y 取值集合. 在此基础上, Neural-Num-LP ^[161] 以及 NLIL ^[162] 进一步拓展了框架的表达力, 使得符号约束能够表达数值关系 (例如, $\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{y}_1 \geq c$) 以及非链式结构 (例如, $a_1(y, u) \wedge a_2(z) \rightarrow c(y)$). 图 7 给出了 TensorLog 运算的一个例子^[163], 其将候选符号约束的推理转化为逻辑原子的概率乘积.

除此以外, NeuralLP 还使用带注意力机制的长短时记忆网络 (long short-term memory, LSTM) 代替简单的权重向量, 来计算每一条候选符号约束的分数. 在此基础上, DRUM ^[164] 将其拓展为了双向循环网络 (bidirectional recurrent neural network, bidirectional RNN). RLogic ^[165] 通过表示学习, 引入了关系嵌入来提升分数预测的性能; NCRL 引入循环注意力元来提高霍尔子句的组合型, 并且根据语义一致性来给逻辑规则评分; RNNLogic ^[166] 额外引入了一个 RNN 规则生成器, 并提出使用 EM 算法将权重预测和规则生成这两个 RNN 模型同步训练.

不同于 ∂ILP 将分数定义给候选霍尔子句, NTP ^[167] 选择直接给每一个逻辑原子分配权重. 具体地说, NTP 同样使用连续的嵌入空间来编码逻辑原子, 并且使用 RBF 核合一运算 (unification) 来定义每一个逻辑原子赋值或者替换的分数, 再定义 AND 和 OR 运算计算完整霍尔子句的分数. 以图 7 为例, $\text{grandPa}(\text{RICK}, \text{MORTY})$ 由候选规则 $p(x, y) \wedge p(y, z) \rightarrow g(x, z)$ 证明的分数将会由 $S' = \text{unify}_\phi(H, G, S)$

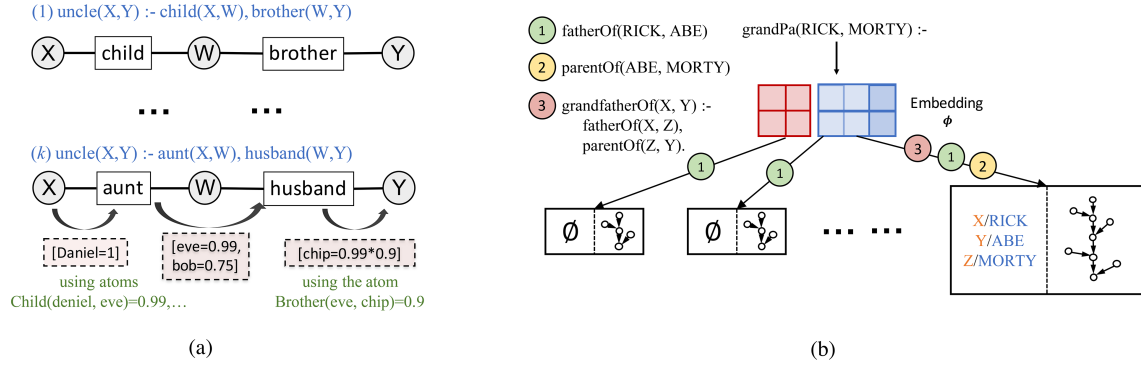


图 7 (网络版彩图) 霍尔子句符号约束合成方法 TensorLog 和 NTP. (a) TensorLog 基于原子命题的概率, 通过矩阵乘法提高候选子句的推理效率; (b) NTP 展开所有的逻辑分支, 穷举所有的候选符号约束, 每一个候选子句的分数由相应逻辑原子的合一运算得到.

Figure 7 (Color online) Symbolic constraint synthesis of Horn clause: TensorLog and NTP. (a) TensorLog is based on the probabilities of atomic propositions and improves the inference efficiency of candidate clauses through matrix multiplication; (b) NTP expands all logical branches and exhaustively enumerates all candidate symbolic constraints. The score of each candidate clause is derived from the unification operation of the corresponding logical atoms.

递归地计算得到^[168]. 其中, S 表示当前状态, 用于存储所有的变量替换, 以及当前状态下合一成功的可能性; H 和 G 为待合一的逻辑原子 (例如, $p(x, y)$ 和 $\text{pa}(\text{RICK}, \text{BETH})$), 其中的非变量符号均对应于 ϕ 中的嵌入表示 (例如, $\phi_p, \phi_{\text{pa}}, \phi_{\text{RICK}}, \phi_{\text{BETH}}$).

类似于 ∂ILP , NTP 需要预先将所有的逻辑推理分支全部展开. 随着逻辑原子, 以及符号约束的增多, 该展开结构过于复杂, 并且很多分支是多余的, 这严重阻碍了 NTP 的学习与推理效率. 为了缓解该问题, GNTTP^[169] 提出贪心选择一个极小的集合作为候选集合; CTP^[170] 提出额外再引入一个选择模块, 即参数化并训练出一个注意力分布, 基于该分布采样出候选集合. 该分布同样由神经网络刻画, 与其嵌入表示一起, 使用梯度下降进行优化. 此外, LRI^[171] 仍然使用逻辑原子的嵌入表示, 但利用给定的模板约束搜索空间; 为了降低模板设计的依赖, HRI^[172] 使用层次结构, 允许堆叠多个霍尔子句, 以提高符号约束学习的表达力.

5.2 合取范式结构

SATNet^[173] 关注于合取范式 (CNF) 形式的符号约束, 即 $\bigwedge_{j=1}^m \bigvee_{i=1}^n a_{ij}$. 给定逻辑原子组成的集合 $\{a_i \mid i = 1, \dots, n\}$, SATNet 引入辅助变量 $s_{ij} \in \{0, 1\}$ 来标记是否将原子命题 a_i 放入第 j 个析取子句, 得到候选符号约束集合 $\{\bigwedge_{j=1}^m \bigvee_{i=1}^n s_{ij} a_i \mid s_{ij} \in \{0, 1\}\}$. 再使用 MaxSAT 转换得到如下形式的优化问题进行符号约束合成:

$$\max_{\tilde{s} \in \{-1, 0, 1\}^{m \times n}} \sum_{j=1}^m \bigvee_{i=1}^n \mathbb{I}(\tilde{s}_{ij} \tilde{a}_i > 0),$$

其中 $\tilde{s}_{ij}$ 的取值集合被拓展到 $\{-1, 0, 1\}$, 以表示取否运算; 相应地用 $\tilde{a}_i \in \{-1, 1\}$ 代替 $a_i \in \{\mathbf{T}, \mathbf{F}\}$. 该优化目标在给定逻辑原子的情形下, 寻找最大可满足的子句结构. 进一步地, SATNet 引入半定松弛 (semidefinite relaxation, SDR)^[174~176] 技术, 将该问题放松为一个可微优化问题, 从而能够使用梯度下降优化去学习松弛的约束矩阵 $\tilde{\mathbf{S}} \in \mathbb{R}^{m \times n}$.

在 SATNet 的基础上, Sym-SATNet^[177] 提出了一个框架, 自动发掘潜在符号约束的置换对称性, 并把该对称性作为 SATNet 的参数矩阵的条件, 从而使得其学习符号的约束满足该置换对称性. 相比与原本的 SATNet, 通过强制参数矩阵的对称性, SymSATNet 极大地减少了参数量, 使其能快速学习到符号约束. 另一方面, 半定松弛的过程并不可逆, 之后得到的约束矩阵 $\tilde{\mathbf{S}}$ 并不能返回计算原先的 $\{-1, 0, 1\}$ 上的示性矩阵, 因此实际上无法得到其合成的符号约束. 为此, ME-SATNet^[178] 提出利用

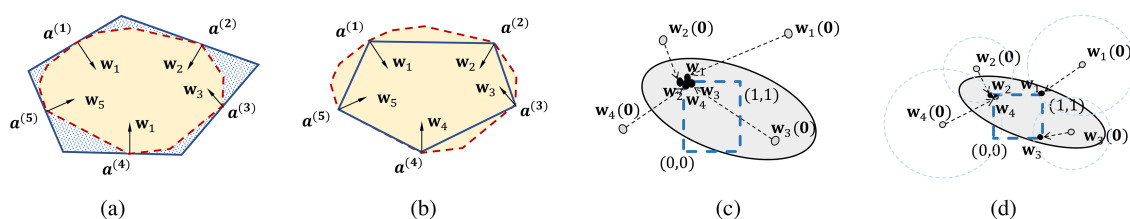


图8 (网络版彩图) 整数线性符号约束合成方法 IntLP 和 PBL. (a) 和 (b) 分别表示外多胞体和内多胞体, IntLP 利用这两个集合所包夹的部分刻画符号约束; (c) 表示整数线性结构可微符号约束合成的退化问题, 即所有的权重向量均收敛到了相同的结果, 使得学出的逻辑规则退化. (d) 引入了信赖域约束, 给每一个权重向量添加了搜索范围, 这样不同的权重向量就能收敛到不同的结果, 确保学得逻辑规则的多样性.

Figure 8 (Color online) Integer linear symbolic constraint synthesis methods IntLP and PBL. (a) and (b) represent the outer and inner polyhedra, respectively. IntLP uses the regions enclosed by these two sets to characterize symbolic constraints. (c) illustrates the degeneration problem in integer linear structure-differentiable symbolic constraint synthesis, where all weight vectors converge to the same result. (d) introduces a trust-region penalty, adding a search range for each initial weight vector. This allows different weight vectors to converge to different results, ensuring the diversity of the learned logical rules.

Kronecker δ 函数, 并且将 SATNet 约束矩阵 $\tilde{S}$ 的训练目标重新建模成了最大化变量相等 (maximum equality), 以确保能够将最终学得的约束矩阵 $\tilde{S}$ 舍入到 $\{-1, 0, 1\}^{m \times n}$. 除此以外, 后续研究发现^[179], 将符号推理模块与神经网络模块联合训练时, SATNet 的性能严重依赖于预训练模型的质量. 为了弥补这一问题, SATNet*^[180] 提出先使用无监督学习技术得到高质量预训练模型, 对输入做聚类并利用聚类结果作为标签数据来预训练神经网络模块.

5.3 整数线性结构

另一类研究较多的符号约束结构是整数线性约束, 即 $w^T a \leq b$, 其中 $w \in \mathbb{Z}^n, b \in \mathbb{Z}$. 选择整数线性约束的原因主要在于: (1) 当前的整数规划求解器的效率较高, 因此无需担心逻辑推理时的效率问题; (2) 很多符号约束合成问题确实可以用整数线性约束去表示^[181]. 为了获得足够精确的符号约束, IntLP^[182] 学出外多胞体 (outer polytope) 和内多胞体 (inner polytope) 两组约束. 如图8所示, 外多胞体与内多胞体所包夹的部分被作为最终学得的整数线性约束, 其中外多胞体和内多胞体需要使用整数线性规划求解器得到. CombOptNet^[183] 同样使用整数线性规划求解器, 但用其来求解 $w^T a \leq b$ (关于 a) 的可行解. 由于整数线性规划问题是分段线性的, 因而通常存在梯度值为0的情形, 为此, CombOptNet 定义了一个替代梯度, 以方便梯度下降法更新参数 w 和 b . 为了避免求解器的调用, Neural-ILP^[184] 基于欧几里得距离定义可满足度来代替具体的求解结果, 进行损失函数计算.

整数线性约束形式的约束合成问题比其他的方法具有更好的可拓展性, 这是由良好的连续性结构, 以及强大的求解器所保证的. 但另一方面, 整数线性约束的可解释性弱于一般的逻辑程序, 其学得的约束仍然难以被人类专家所理解. 为此, PBL^[35] 提出使用差分凸优化 (difference-of-convex) 技术^[185, 186], 在保证梯度下降可行性的同时, 将权重系数 w 限制在布尔类型 $\{0, 1\}^n$. 该限制使得学出的符号约束为伪布尔类型 (pseudo-boolean), 确保了其与合取范式之间的等价转换^[187, 188]. 除此以外, 整数线性的候选符号约束空间是无穷大的, 因此符号约束合成过程很容易退化 (degeneracy), 即学出无意义或者冗余的规则. 例如, 假设目标逻辑约束是 $a_1 + a_2 + a_3 \geq 2$ 和 $a_3 = 1$, 退化将导致最终只得到单一约束 $a_1 + a_2 + a_3 \geq 2$ 或者一个平凡约束 $a_3 \geq 0$. 为解决该问题, Neural-ILP 提出使用协方差正则化来保证系数 $w_1, \dots, w_k$ 之间的低相关性; 而 PBL 利用信赖域约束. 如图8所示, 信赖域约束将每一条约束的搜索空间分开, 以确保学得的约束互不相同. 值得一提的是, 退化问题与推理捷径问题存在许多相似之处, 二者均需要引入非确定性来实现对可行的符号接地/符号约束的广泛搜索, 并成功收敛到预期结果.

5.4 逻辑网络结构

最后一类符号约束结构直接使用神经网络结构来近似一阶或高阶的逻辑规则. 这一看似不合理的方式实际上有着严格的理论支撑: 已有理论研究表明, 仅使用多层感知机 (multilayer perceptron, MLP) 就可以拟合任意的布尔函数^[189,190]. 基于此, NLM^[191] 使用张量表示逻辑原子 $\tilde{a}_i \in \{0, 1\}, i = 1, \dots, n$, 并通过置换运算 (permute) 以及 Sigmoid 激活函数的 MLP 来拟合 And 以及 Or 的逻辑运算, 再对张量使用扩展 (expansion) 和收缩 (reduction) 的操作使其能够近似符号规则中的存在和任意的量词. 但是, 多层感知机拟合任意布尔函数仅仅是理论结果, 对于实际训练得到的多层感知机, 往往很难计算出其对应的布尔函数并提取出相应的逻辑规则.

为解决该问题, dNL^[192] 放弃使用多层感知机, 而是将模糊逻辑的计算方式嵌入到神经网络中. 具体地说, dNL 使用乘积三角范数来表示神经网络“合取”层 $F_{\wedge}(\tilde{a}_1, \dots, \tilde{a}_n) = \prod_{i=1}^n w_i \tilde{a}_i$, 以及神经网络“析取”层 $F_{\vee}(\tilde{a}_1, \dots, \tilde{a}_n) = 1 - \prod_{i=1}^n (1 - w_i \tilde{a}_i)$, 其中权重 $w_i \in [0, 1]$ 表示原子命题 $\tilde{a}_i$ 是否被放入析取/合取式. 然而, dNL 仅使用两层网络, 即一层析取层和一层合取层, 来表示合取范式或者析取范式, 这使得所需逻辑原子的数量较多. DLM^[193] 将其扩展到多层, 并将权重 $w_i, i = 1, \dots, n$ 离散化为分类分布, 以确保能够得到最终的符号约束. 然后, DLM 使用 Gumbel-Softmax 对权重 w 进行重参数化, 以支持梯度下降训练. 另一种常被使用的模糊逻辑是 Lukasiewicz 三角范数, LNN^[194,195] 和 LRNN^[196] 均使用 $F_{\wedge}(\tilde{a}_1, \dots, \tilde{a}_n) = \text{Sigmoid}(b - \sum_{i=1}^n w_i(1 - \tilde{a}_i))$ 和 $F_{\vee}(\tilde{a}_1, \dots, \tilde{a}_n) = \text{Sigmoid}(\sum_{i=1}^n w_i \tilde{a}_i + 1 - b)$ 的形式作为“合取/析取”层. 注意, Lukasiewicz 三角范数中的 min/max 运算被 Sigmoid 激活函数结合一个偏置项 b 所代替, 以防止梯度消失.

另外一类直接基于神经网络的工作选择放弃对逻辑规则的拟合, 而是直接使用递归网络进行近似逻辑推理. RRN^[197] 是其中的典型, 其提出的递归关系网络利用图结构上的消息传递来学习逻辑规则. 除此以外, DRNet^[198] 以及其改进版本 CLR-DRNet^[199] 引入一个推理模块来限制隐空间的表示, 再使用生成式解码器根据包含推理信息的表示进行推断. 但是, RTNet^[50] 表明, 这些方法都无法实现神经网络模块和符号推理模块的共同训练, 而且, 直接使用递归变换器 (recurrent transformer), 即使用递归结构来鼓励变换器实现多步推理, 能够显著地超过已有方法. 值得注意的是, 这可能是将大语言模型与神经符号系统结合的一个有效方法.

5.5 小结

本节讨论了多种针对不同约束结构的符号约束合成方法, 特别是其可微化技术. 这些技术所解决的关键问题在于如何引入和表示一个候选符号约束的非确定性评估. 例如, ∂ ILP 将非确定性赋予所有的候选霍尔子句, 而 NTP 将非确定性计算迁移到了逻辑原子上; ∂ ILP 使用一个简单的权重表示非确定性, 而 NeuralLP 利用网络预测得到非确定性. 但在应对大规模的符号约束合成任务时, 现有方法往往效率过低, 因而大多并不能直接适配“端到端”的神经符号学习任务. 所以, 如何针对不同的符号约束结构更高效地引入非确定性, 仍然是当前需要解决的问题.

除此以外, 当前已经证明支持“端到端”神经符号学习的框架也仍有许多缺陷. 例如, SATNet 已经能够完成 4×4 视觉数独求解任务的“端到端”神经符号学习, 但它的候选符号约束的非确定性由松弛后的系数矩阵表示, 导致逻辑规则无法被提取出来, 因而缺乏可解释性; PBL 相比 SATNet 更为高效, 甚至可以完成简化设置下自动驾驶任务的“端到端”神经符号学习, 但其只能学得伪布尔符号约束, 逻辑的阶次 (order) 太低, 将这些规则解释成人们可读的形式仍然非常困难. 另外, 随着问题维度的升高, 太低阶次的符号约束也导致在合成过程中极易产生退化问题. 总之, 如何管理非确定性, 确保其能够得到精确的逻辑规则, 以及如何实现升阶 (lifting) 来降低搜索空间, 同样是开放研究问题.

6 未来工作展望

我们从完善理论基础、提升规模扩展性、实现工程化、结合大语言模型以及拓展实际应用场景 5 个方面展望神经符号系统和神经符号学习进一步的工作方向.

6.1 神经符号系统的理论基础

概括而言, 符号逻辑规则, 包括软件程序代码, 以逻辑蕴含意义的正确性, 实现知识驱动的演绎推理, 其行为可解释、可组合、可信任; 而机器学习, 特别是深度学习, 以概率近似正确的方式, 实现数据驱动的归纳推理, 在处理涉及高维对象的非确定问题方面展示了强大的能力. 如何建立一个统一的理论框架融合二者之长, 是长期以来的挑战问题. 近期 Zhou 等^[48, 48, 200]提出的反绎学习, 把演绎反过来嵌入到归纳中去, 通过“机器学习 – 逻辑推理 – 机器学习 – 逻辑推理 – ...”的迭代循环, 实现二者的共同促进, 成为了一个值得关注的途径.

在神经符号学习的理论研究方面, 相关研究仍然非常匮乏. 首先, 对于神经符号接地问题, 在模型计数方法中, 当前的启发式搜索算法 (例如, top- k 近似) 均引入了偏差, 无法直接套用已有关于收敛性以及泛化误差界的理论结果^[201]. 因此, 未来研究工作需要对该偏差建模, 并且提供相应的理论分析结果. 模糊逻辑方法与之类似, 但除收敛性和泛化误差界的分析以外, 模糊逻辑方法还缺少不同转换策略之间的广泛比较, 以及相应的理论分析结果^[132]. 其次, 推理捷径问题的数学定义不够精确, 也缺少更深层次的理论分析讨论推理捷径发生的条件^[80]. 这也导致当前推理捷径的解决方案缺乏严格的理论证明作为支撑. 最后, 在可微符号约束合成问题中, 尽管目标约束结构非常清晰, 但学习算法本身真正的表达力 (例如, 能否学得任意满足条件的目标约束) 大多欠缺精确的理论刻画.

对于“端到端”的神经符号学习而言, 尽管交替梯度下降的收敛性分析已经有较多研究成果^[67], 然而, 对于神经符号学习这一具体问题, 仍需进行更多研究, 以提供更准确的算法前提描述、更紧的收敛速率结果以及更细致的收敛解性质分析. 此外, 相比较于模块独立的训练方式, “端到端”神经符号学习的优越性和鲁棒性需要新的理论框架和结果来支撑. 相对应地, 非确定性管理在“端到端”神经符号学习中的必要性, 以及如何引入非确定性管理以最大化性能提升, 也是当前值得研究的理论问题.

6.2 神经符号学习的规模扩展性

在神经符号学习中, 神经符号接地和可微符号约束合成都会受到维数灾难^[202]的影响. 在神经符号接地问题中, 随着变量数的升高和符号空间的增大, 符号求解工具效率会大幅降低; 而与之相对的, 可行符号接地的分布将变得更加稀疏, 因此已有的约束外搜索方法的效率同样受到很大影响. 即便使用 MCMC 这样的采样技术实现近似计数, 更高的问题维度同样影响到其随机游走的效率. 类似的问题同样发生在可微符号约束合成任务中. 具体而言, 随着变量数的增多, 候选符号约束的数量呈指数级上升, 这也增加了最优符号约束搜索和优化的难度. 因此, 如何在神经符号学习中应用和适配更高效的高维搜索或者采样算法 (例如, HMC 采样^[203], GFlowNet^[204]等) 是当前需要进一步研究的问题.

此外, 高维复杂任务中, 推理效率问题是限制神经符号系统应用的主要瓶颈. 其最主要的原因在于符号规则模块的计算效率过低. 因此, 关于提高符号规则模块效率的研究至关重要. 例如, 一个较为可靠的, 缓解该问题的策略是拓展神经符号系统的结构^[38], 使得神经网络模块在提供感知预测的同时也提供快速且不精确的推理结果. 这使得我们能够采用一种投机策略在加速神经符号系统的推理速度: 符号规则模块一般只需要验证神经网络模块的不精确推理是否正确, 这是快速且高效的; 而只有在判定出预测结果不正确时, 符号规则模块才会执行完全的逻辑推理, 得到纠正后的答案. 以图 1(a) 中的水果图片数独问题为例, 神经网络模块不仅仅分类预测图片对应的数字, 也会预测所需填空数字; 符号规则模块将快速判定所预测的结果是否符合数独规则, 并仅修改错误的部分.

6.3 神经符号系统的工程化实现

当前,神经符号系统的工程化实现仍然面临许多挑战:(1)从需求角度看,考虑到神经符号学习和推理的高代价,需要谨慎识别真正需要神经与符号系统协同工作的情况,按需部署神经符号系统。(2)从开发角度看,神经网络模型与符号推理系统的开发方式天然不同,因此需要构建新型软件支撑环境,更好地支持神经网络的训练、神经符号交互等关键环节。(3)从质量角度看,受限于神经网络模型非确定性的存在,神经符号系统可能面临着更复杂的鲁棒性、安全性等方面的挑战。(4)从效率角度看,实际工程实现时需要考虑符号系统与神经系统的适配性,例如,双方的计算资源配置、推理速度是否匹配等。

特别地,针对神经符号系统,设计特定的网络结构以及构建符号推理工具是系统工程化实现的关键基础。首先,通用的网络结构往往不适用于后续的符号推理任务,引起性能上的损失。当前许多研究使用图神经网络模型^[198,199],原因是该结构理论上能够更好地利用符号推理帮助神经网络训练,在推理时神经网络也能更方便地与符号规则模块协同工作,从而提高系统整体性能。然而,近期有研究表明,这种结构其实际效果提升有限,推理效率也不高,不足以应用到复杂任务上^[50]。因此,仍然需要进一步研究和探索更适合神经符号系统的网络结构。

其次,符号规则模块也需要相应的改进,以确保能够在符号推理过程中充分利用神经网络模型的预测信息。当前,大多符号推理模块都基于已有的,较为成熟的符号推理引擎,例如 Z3, ASP, CLP 等。由于这些系统内部实现十分复杂精巧,贸然修改其代码实现引入额外的预测信息常常反而会导致推理性能的下降。因此,已有的研究工作仅限于把符号推理引擎当作黑盒,基于预测结果修改输入,或者修改推理工具的配置参数,来达到提高神经符号系统推理效率的目的^[38]。相较而言,灵活性、适配性较高的定理证明器,例如 Isabelle, Lean, Coq 等,更适合用于神经符号系统中的符号推理引擎。特别地,Isabelle 本身就包含机器学习模块,用以辅助定理证明搜索,加速符号规则的证明。已有工作,例如 Thor 和 Baldur^[205,206],已经成功将神经网络模型嵌入定理证明器之中以加速符号推理。

6.4 神经符号系统与大语言模型结合

大语言模型在语言理解、代码生成,以及数学问题求解等许多场景下已经被证明有极好的表现^[207~209]。但是,人们普遍认为,逻辑推理能力是大语言模型的一个重要缺陷^[210~212]。将大语言模型嵌入神经符号范式也许是解决该问题的一个有效方案。这方面已有研究进行了初步探索。例如,Logic-LM^[213]提出利用大语言模型将推理问题形式化,再使用符号求解工具(例如,SMT 求解器)去进行逻辑推理;DTV^[214]利用自动形式化技术^[215],将大语言模型的数学推理过程映射到 Isabelle/HOL 语言,再使用自动定理证明器去验证其正确性^[216]。最值得一提的是 AlphaGeometry^[217],它利用大语言模型进行几何问题的辅助点生成,再使用符号演绎引擎来实现 IMO 级别几何问题的定理证明。这些进展提示我们,把大语言模型嵌入神经符号系统,提高大语言模型自身性能乃至进一步提高神经符号系统性能,是当前最具有潜力的研究方向之一。

另一方面,利用大语言模型提高原先神经符号学习任务的效果和效率同样值得研究。换句话说,神经符号系统的两个模块,神经网络模块和符号规则模块,均可以使用大语言模型来替代或者帮助构建。对于神经网络模块,将大语言模型用于替代传统的机器学习/深度学习模型不仅仅可以提高模块的性能,也能够帮助整个符号接地的建立^[218,219]。例如,在神经符号接地中,已有工作通过提示大语言模型来提供可行符号接地,以及加速符号接地的搜索过程^[218,219]。在可微符号约束合成中,大语言模型可以从收集到的数据中提出候选符号约束,并且利用输出的对数概率来引导符号约束合成^[220,221]。另外,上述一系列自动形式化的工作表明^[213~215],大语言模型能够很好地完成自动形式化的任务,这极大地扩展了神经符号系统的应用场景。另一方面,对于符号规则模块,虽然大语言模型的逻辑推理能力仍需要进一步探索,但将其与符号推理工具结合,在保证精确度的同时提高推理速度已经被证明是

可行的 [38, 218]. 而且大语言模型还能够生成符号规则的自然语言解释, 甚至去形式化, 将符号约束翻译回自然语言 [222, 223], 提高规则的可读性以及可理解性.

6.5 神经符号系统的实际应用场景

迄今为止, 神经符号学习主要在一些简单, 或者较为纯粹的实验性任务上取得较好的效果. 将神经符号学习技术应用到实际场景, 在生产环节中部署神经符号系统仍有许多挑战. 一个特别典型, 并且值得探索的场景是自动驾驶. 从理论上说, 我们可以利用神经符号学习技术, 不仅自动地从人类的驾驶行为中学会驾驶, 而且自动地合成符号化的驾驶规则, 并确保其符合交通法规和安全驾驶的要求, 从而提高自动驾驶系统的性能与鲁棒性 [224~226]. 但实际上, 对于学习驾驶场景对应的时序逻辑规则, 以及将这些复杂的时序逻辑规则引入神经符号接地过程支持模型训练, 当前的神经符号学习技术大多无能为力.

此外, 在许多程序代码生成场景下, 神经符号系统也有着良好的理论可行性和应用潜力. 以视觉编程 (visual programming) [227~229] 为例, 该任务要求神经符号系统按照一系列的代码模版, 合成出符合要求的视频剪辑程序, 并执行得到最终的剪辑结果. 但是该任务非常依赖于专家设计模版, 如何自动地生成代码模版, 以及拓展到别的代码生成的生产任务中, 当前仍需进一步研究. 神经符号系统在数学定理证明任务中也有较大潜力. 例如, Leandojo [230] 结合神经网络模型和符号搜索与推理引擎 Aesop [231], 大幅提高了当前自动定理证明工具的能力. Tao [232] 预期到 2026 年, 集成了形式证明验证器、网络搜索、符号数学库的人工智能将成为数学研究中可信赖的共同作者.

7 结论

非常抽象地说, 学习本身就是依据经验在非确定性中寻找确定性的过程, 而在此过程中坚持“知之为知之, 不知为不知”, 亦即对非确定性进行显式地建模和管理, 是实现有效的神经符号学习, 融合神经计算和逻辑推理的一个关键. 基于这样的认识, 本文以非确定性管理的视角, 为“端到端”神经符号学习建立了一个双层优化框架, 进而在此框架下, 对神经符号接地以及可微符号合成两个步骤, 以及其间必须克服的捷径满足问题, 进行了分析和讨论, 讨论了已有相关工作的进展和局限, 并从理论基础、规模扩展性、工程化实现、与大语言模型结合以及应用场景等方面展望了未来的工作方向.

结合知识驱动与数据驱动的途径构建功能强大的智能化软件系统正在走向主流. 神经符号系统作为其微观基础, 其理论、方法和技术的研究远未充分. 对其中最具挑战性的“端到端”神经符号学习的基本理论和算法的研究有望带来重要的启示. 希望本文能为这方面的探索提供有益的指引.

参考文献

- 1 Valiant L G. A theory of the learnable. *Commun ACM*, 1984, 27: 1134–1142
- 2 Angluin D. Queries and concept learning. *Machine Learn*, 1988, 2: 319–342
- 3 de Raedt L, Passerini A, Teso S. Learning constraints from examples. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018
- 4 Gulwani S. AI-assisted programming: applications, user experiences, and neuro-symbolic techniques (keynote). In: *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022
- 5 Kahneman D. *Thinking, Fast and Slow*. New York City: Farrar, Straus and Giroux, 2011
- 6 Yoshua B, Gary M. AI Debate: the best way forward for AI. 2020. <https://montrealartificialintelligence.com/aidebate/>
- 7 LeCun Y. A Path Towards Autonomous Machine Intelligence. *Open Review*, 2022. <https://openreview.net/pdf?id=BZ5a1r-kVsf>

- 8 Hochreiter S. Toward a broad AI. *Commun ACM*, 2022, 65: 56–57
- 9 Zhang B, Zhu J, Su H. Toward the third generation artificial intelligence. *Sci Sin Inform*, 2020, 50: 1281–1302 [张钹, 朱军, 苏航. 迈向第三代人工智能. *中国科学: 信息科学*, 2020, 50: 1281–1302]
- 10 Yurtsever E, Lambert J, Carballo A, et al. A survey of autonomous driving: common practices and emerging technologies. *IEEE Access*, 2020, 8: 58443–58469
- 11 Pesapane F, Volonté C, Codari M, et al. Artificial intelligence as a medical device in radiology: ethical and regulatory issues in Europe and the United States. *Insights Imag*, 2018, 9: 745–753
- 12 Shridhar M, Manuelli L, Fox D. CLIPort: what and where pathways for robotic manipulation. In: *Proceedings of Conference on Robot Learning*, 2022. 894–906
- 13 Hitzler P, Eberhart A, Ebrahimi M, et al. Neuro-symbolic approaches in artificial intelligence. *Natl Sci Rev*, 2022, 9: nwac035
- 14 Wang W, Yang Y, Wu F. Towards data-and knowledge-driven artificial intelligence: a survey on neuro-symbolic computing. 2022. *ArXiv:2210.15889*
- 15 Marra G, Dumančić S, Manhaeve R, et al. From statistical relational to neurosymbolic artificial intelligence: a survey. *Artif Intell*, 2024, 328: 104062
- 16 Wan Z, Liu C K, Yang H, et al. Towards cognitive AI systems: a survey and prospective on neuro-symbolic AI. 2024. *ArXiv:2401.01040*
- 17 Manduchi L, Pandey K, Bamler R, et al. On the challenges and opportunities in generative AI. 2024. *ArXiv:2403.00025*
- 18 Lavrac N, Dzeroski S. Inductive logic programming. In: *Proceedings of Wafer Level Package Seminar*, 1994. 146–160
- 19 Cropper A, Dumančić S, Evans R, et al. Inductive logic programming at 30. *Mach Learn*, 2022, 111: 147–172
- 20 Halbert D C. *Programming by Example*. Berkeley: University of California, Berkeley, 1984
- 21 Gulwani S. Programming by examples. *Depend Softw Syst Eng*, 2016, 45: 3–15
- 22 Lau T, Wolfman S A, Domingos P, et al. Programming by demonstration using version space algebra. *Machine Learn*, 2003, 53: 111–156
- 23 Calinon S. Learning from demonstration (programming by demonstration). In: *Encyclopedia of Robotics*. Berlin: Springer, 2018
- 24 Mernik M, Heering J, Sloane A M. When and how to develop domain-specific languages. *ACM Comput Surv*, 2005, 37: 316–344
- 25 Manhaeve R, Dumancic S, Kimmig A, et al. DeepProbLog: neural probabilistic logic programming. In: *Proceedings of Advances in Neural Information Processing Systems*, 2018
- 26 Xu J, Zhang Z, Friedman T, et al. A semantic loss function for deep learning with symbolic knowledge. In: *Proceedings of International Conference on Machine Learning*, 2018. 5502–5511
- 27 Fischer M, Balunovic M, Drachler-Cohen D, et al. DL2: training and querying neural networks with logic. In: *Proceedings of International Conference on Machine Learning*, 2019. 1931–1941
- 28 Kambhampati S. Can large language models reason and plan? *Ann New York Acad Sci*, 2024, 1534: 15–18
- 29 Donoho D L, Huo X. Uncertainty principles and ideal atomic decomposition. *IEEE Trans Inform Theor*, 2001, 47: 2845–2862
- 30 Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: *Proceedings of International Conference on Machine Learning*, 2016. 1050–1059
- 31 Guo C, Pleiss G, Sun Y, et al. On calibration of modern neural networks. In: *Proceedings of International Conference on Machine Learning*, 2017. 1321–1330
- 32 Mukhoti J, Kulharia V, Sanyal A, et al. Calibrating deep neural networks using focal loss. In: *Proceedings of Advances in Neural Information Processing Systems*, 2020. 33: 15288–15299
- 33 Li Z, Liu Z, Yao Y, et al. Learning with logical constraints but without shortcut satisfaction. In: *Proceedings of the 11th International Conference on Learning Representations*, 2022
- 34 Li Z, Yao Y, Chen T, et al. Softened symbol grounding for neuro-symbolic systems. In: *Proceedings of the 11th International Conference on Learning Representations*, 2022
- 35 Li Z, Huang Y, Li Z, et al. Neuro-symbolic learning yielding logical constraints. In: *Proceedings of Advances in*

- Neural Information Processing Systems, 2024
- 36 Garcez A d, Bader S, Bowman H, et al. Neural-symbolic learning and reasoning: a survey and interpretation. In: *Neuro-Symbolic Artificial Intelligence: The State of the Art*. Amsterdam: IOS Press, 2022. 342: 327–327
 - 37 Kautz H A. The third AI summer: AAAI Robert S. Englemore Memorial Lecture. *AI Mag*, 2022, 43: 105–125
 - 38 Cornelio C, Stuehmer J, Hu S X, et al. Learning where and when to reason in neuro-symbolic inference. In: *Proceedings of the 11th International Conference on Learning Representations*, 2022
 - 39 Rawat W, Wang Z. Deep convolutional neural networks for image classification: a comprehensive review. *Neural Comput*, 2017, 29: 2352–2449
 - 40 Yu D, Deng L. *Automatic Speech Recognition*. Berlin: Springer, 2016
 - 41 Otter D W, Medina J R, Kalita J K. A survey of the usages of deep learning for natural language processing. *IEEE Trans Neural Netw Learn Syst*, 2021, 32: 604–624
 - 42 Shapiro E Y. The model inference system. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981. 1064–1064
 - 43 De Moura L, Bjørner N. Z3: an efficient SMT solver. In: *Proceedings of International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2008. 337–340
 - 44 Barbosa H, Barrett C, Brain M, et al. Cvc5: a versatile and industrial-strength SMT solver. In: *Proceedings of International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2022. 415–442
 - 45 Cai Y H, Yang H, An X, et al. Study on neurosymbolic learning and its applications. *Chin J Intell Sci Technol*, 2022, 4: 560–570 [蔡莹皓, 杨华, 安璇, 等. 神经符号学及其应用研究. *智能科学与技术学报*, 2022, 4: 560–570]
 - 46 Lau T A, Domingos P M, Weld D S. Version space algebra and its application to programming by demonstration. In: *Proceedings of International Conference on Machine Learning*, 2000. 527–534
 - 47 Cordts M, Omran M, Ramos S, et al. The cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3213–3223
 - 48 Zhou Z H, Huang Y X. Abductive learning. In: *Neuro-Symbolic Artificial Intelligence: The State of the Art*. Amsterdam: IOS Press, 2021
 - 49 Duan X, Wang X, Zhao P, et al. DeepLogic: joint learning of neural perception and logical reasoning. *IEEE Trans Pattern Anal Mach Intell*, 2022, 45: 4321–4334
 - 50 Yang Z, Ishay A, Lee J. Learning to solve constraint satisfaction problems with recurrent transformer. 2023. ArXiv:2307.04895
 - 51 Bard J F. *Practical Bilevel Optimization: Algorithms and Applications*. New York: Springer Science & Business Media, 2013
 - 52 Dempe S, Zemkoho A. Bilevel optimization. In: *Topological Aspects of Nonsmooth Optimization*. Berlin: Springer, 2020
 - 53 Gunawardana A, Byrne W, Jordan M I. Convergence theorems for generalized alternating minimization procedures. *J Machine Learn Res*, 2005, 6: 2049–2073
 - 54 Bezdek J C, Hathaway R J. Some notes on alternating optimization. In: *Proceedings of AFSS International Conference on Fuzzy Systems Calcutta*, 2002. 288–300
 - 55 Powell M J D. On search directions for minimization algorithms. *Math Programming*, 1973, 4: 193–201
 - 56 Daskalakis C, Panageas I. The limit points of (optimistic) gradient descent in min-max optimization. In: *Proceedings of Advances in Neural Information Processing Systems*, 2018
 - 57 Pethick T, Latafat P, Patrinos P, et al. Escaping limit cycles: global convergence for constrained nonconvex-nonconcave minimax problems. 2023. ArXiv:2302.09831
 - 58 Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. *Commun ACM*, 2020, 63: 139–144
 - 59 Creswell A, White T, Dumoulin V, et al. Generative adversarial networks: an overview. *IEEE Signal Process Mag*, 2018, 35: 53–65
 - 60 Zhou Z, Liang J, Song Y, et al. Lipschitz generative adversarial nets. In: *Proceedings of International Conference on Machine Learning*, 2019. 7584–7593
 - 61 Wu Y, Zhou P, Wilson A G, et al. Improving gan training with probability ratio clipping and sample reweighting. In: *Proceedings of Advances in Neural Information Processing Systems*, 2020. 33: 5729–5740

- 62 Sajeeda A, Hossain B M M. Exploring generative adversarial networks and adversarial training. *Int J Cogn Comput Eng*, 2022, 3: 78–89
- 63 Boyd N, Schiebinger G, Recht B. The alternating descent conditional gradient method for sparse inverse problems. *SIAM J Optim*, 2017, 27: 616–639
- 64 Boş I, Böhm A. Alternating proximal-gradient steps for (stochastic) nonconvex-concave minimax problems. *SIAM J Optim*, 2023, 33: 1884–1913
- 65 Ha W, Barber R F. Alternating minimization and alternating descent over nonconvex sets. 2017. ArXiv:1709.04451
- 66 Lee J, Cho H, Yun C. Fundamental benefit of alternating updates in minimax optimization. 2024. ArXiv:2402.10475
- 67 Chen T, Sun Y, Yin W. Closing the gap: tighter analysis of alternating stochastic gradient methods for bilevel problems. In: *Proceedings of Advances in Neural Information Processing Systems*, 2021. 25294–25307
- 68 Ji K, Yang J, Liang Y. Bilevel optimization: convergence analysis and enhanced design. In: *Proceedings of International Conference on Machine Learning*, 2021. 4882–4892
- 69 Liu B, Ye M, Wright S, et al. Bome! Bilevel optimization made easy: a simple first-order approach. In: *Proceedings of Advances in Neural Information Processing Systems*, 2022. 35: 17248–17262
- 70 Shen H, Chen T. On penalty-based bilevel gradient descent method. In: *Proceedings of International Conference on Machine Learning*, 2023. 30992–31015
- 71 Schäfer F, Anandkumar A. Competitive gradient descent. In: *Proceedings of Advances in Neural Information Processing Systems*, 2019
- 72 Fiez T, Chasnov B, Ratliff L. Implicit learning dynamics in Stackelberg games: equilibria characterization, convergence analysis, and empirical study. In: *Proceedings of International Conference on Machine Learning*, 2020. 3133–3144
- 73 Li Z, Usman M, Tao R, et al. A systematic survey of regularization and normalization in GANs. *ACM Comput Surv*, 2023, 55: 1–37
- 74 Harnad S. The symbol grounding problem. *Phys D-Nonlinear Phenom*, 1990, 42: 335–346
- 75 Taddeo M, Floridi L. Solving the symbol grounding problem: a critical review of fifteen years of research. *J Exp Theor Artif Intell*, 2005, 17: 419–445
- 76 Valiant L G. The complexity of enumeration and reliability problems. *SIAM J Comput*, 1979, 8: 410–421
- 77 Karp R M, Luby M, Madras N. Monte-Carlo approximation algorithms for enumeration problems. *J Algorithms*, 1989, 10: 429–448
- 78 Creignou N, Hermann M. Complexity of generalized satisfiability counting problems. *Inf Comput*, 1996, 125: 1–12
- 79 He H Y, Dai W Z, Li M. Reduced implication-bias logic loss for neuro-symbolic learning. *Mach Learn*, 2024, 113: 3357–3377
- 80 Marconato E, Teso S, Vergari A, et al. Not all neuro-symbolic concepts are created equal: analysis and mitigation of reasoning shortcuts. In: *Proceedings of Advances in Neural Information Processing Systems*, 2024
- 81 van Krieken E, Minervini P, Ponti E M, et al. On the independence assumption in neurosymbolic learning. 2024. ArXiv:2404.08458
- 82 Chavira M, Darwiche A. On probabilistic inference by weighted model counting. *Artif Intell*, 2008, 172: 772–799
- 83 Kimmig A, van den Broeck G, de Raedt L. Algebraic model counting. *J Appl Logic*, 2017, 22: 46–62
- 84 Hájek P. *Metamathematics of Fuzzy Logic*. Berlin: Springer Science & Business Media, 2013
- 85 van Krieken E, Acar E, van Harmelen F. Analyzing differentiable fuzzy logic operators. *Artif Intell*, 2022, 302: 103602
- 86 Raedt L D, Kersting K, Natarajan S, et al. Statistical relational artificial intelligence: logic, probability, and computation. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Cham: Springer, 2016. 10: 1–189
- 87 Akers. Binary decision diagrams. *IEEE Trans Comput*, 1978, 100: 509–516
- 88 Bryant R E. Graph-based algorithms for Boolean function manipulation. *IEEE Trans Comput*, 1986, 100: 677–691
- 89 Darwiche A. SDD: a new canonical representation of propositional knowledge bases. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011
- 90 Choi A, Darwiche A. Dynamic minimization of sentential decision diagrams. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2013. 187–194

- 91 Darwiche A, Marquis P. A knowledge compilation map. *J Artif Intell Res*, 2002, 17: 229–264
- 92 Huang J, Li Z, Chen B, et al. Scallop: from probabilistic deductive databases to scalable differentiable reasoning. In: *Proceedings of Advances in Neural Information Processing Systems*, 2021, 34: 25134–25145
- 93 Manhaeve R, Marra G, de Raedt L. Approximate inference for neural probabilistic logic programming. In: *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, 2021. 475–486
- 94 van Krieken E, Thanapalasingam T, Tomczak J, et al. A-NeSI: a scalable approximate method for probabilistic neurosymbolic inference. In: *Proceedings of Advances in Neural Information Processing Systems*, 2024
- 95 Li Q, Huang S, Hong Y, et al. Closed loop neural-symbolic learning via integrating neural perception, grammar parsing, and symbolic reasoning. In: *Proceedings of International Conference on Machine Learning*, 2020. 5884–5894
- 96 Gu J, Purdom P W, Franco J V, et al. Algorithms for the satisfiability (SAT) problem: a survey. In: *Handbook of Combinatorial Optimization*. Oxford: Oxford University Press, 1996. 35: 19–152
- 97 Gong W, Zhou X. A survey of SAT solver. In: *Proceedings of AIP Conference Proceedings*, 2017
- 98 de Moura L, Dutertre B, Shankar N. A tutorial on satisfiability modulo theories: (invited tutorial). In: *Proceedings of International Conference on Computer Aided Verification*, 2007. 20–36
- 99 Monniaux D. A survey of satisfiability modulo theory. In: *Proceedings of the 18th International Workshop on Computer Algebra in Scientific Computing*, 2016. 401–425
- 100 Yang Z, Ishay A, Lee J. NeurASP: embracing neural networks into answer set programming. 2023. ArXiv:2307.07700
- 101 Skryagin A, Stammer W, Ochs D, et al. SLASH: embracing probabilistic circuits into neural answer set programming. 2021. ArXiv:2110.03395
- 102 Skryagin A, Stammer W, Ochs D, et al. Neural-probabilistic answer set programming. In: *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, 2022. 463–473
- 103 Gebser M, Kaminski R, Kaufmann B, et al. Theory solving made easy with Clingo 5. In: *Proceedings of Technical Communications of the 32nd International Conference on Logic Programming*, 2016
- 104 Lifschitz V. *Answer Set Programming*. Cham: Springer, 2019
- 105 Gebser M, Kaufmann B, Neumann A, et al. Conflict-driven answer set enumeration. In: *Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning*, 2007. 136–148
- 106 Dai W Z, Xu Q, Yu Y, et al. Bridging machine learning and logical reasoning by abductive learning. In: *Proceedings of Advances in Neural Information Processing Systems*, 2019
- 107 Dai W Z, Muggleton S H. Abductive knowledge induction from raw data. 2020. ArXiv:2010.03514
- 108 Yu Y, Qian H, Hu Y Q. Derivative-free optimization via classification. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016
- 109 Triska M. The Boolean constraint solver of SWI-prolog (system description). In: *Proceedings of International Symposium on Functional and Logic Programming*, 2016. 45–61
- 110 Wang Z, Vijayakumar S, Lu K, et al. Grounding neural inference with satisfiability modulo theories. In: *Proceedings of Advances in Neural Information Processing Systems*, 2024
- 111 Hassan M, Urban C, Eilers M, et al. MaxSMT-based type inference for Python 3. In: *Proceedings of the 30th International Conference on Computer Aided Verification*, 2018. 12–19
- 112 Li C M, Many F. MaxSAT, hard and soft constraints. In: *Handbook of Satisfiability*. Amsterdam: IOS Press, 2021. 903–927
- 113 Feng W, He K, Yin Y. Sampling constraint satisfaction solutions in the local lemma regime. In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021. 1565–1578
- 114 Feng W, Guo H, Yin Y, et al. Fast sampling and counting k -SAT solutions in the local lemma regime. *J ACM*, 2021, 68: 1–42
- 115 Wigderson A. *Mathematics and Computation: A Theory Revolutionizing Technology and Science*. Princeton: Princeton University Press, 2019
- 116 Dombi J. Membership function as an evaluation. *Fuzzy Sets Syst*, 1990, 35: 1–21
- 117 Jain A, Sharma A. Membership function formulation methods for fuzzy logic systems: a comprehensive review. *J Crit Rev*, 2020, 7: 8717–8733

- 118 Gupta M M, Qi J. Theory of T-norms and fuzzy inference methods. *Fuzzy Sets Syst*, 1991, 40: 431–450
- 119 Klement E P, Mesiar R, Pap E. *Triangular Norms*. Berlin: Springer Science & Business Media, 2013
- 120 Pryor C, Dickens C, Augustine E, et al. NeuPSL: neural probabilistic soft logic. 2022. ArXiv:2205.14268
- 121 Shengyuan C, Cai Y, Fang H, et al. Differentiable neuro-symbolic reasoning on large-scale knowledge graphs. In: *Proceedings of Advances in Neural Information Processing Systems*, 2024
- 122 Kimmig A, Bach S, Broecheler M, et al. A short introduction to probabilistic soft logic. In: *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, 2012. 1–4
- 123 Diligenti M, Gori M, Sacca C. Semantic-based regularization for learning and inference. *Artif Intell*, 2017, 244: 143–165
- 124 Serafini L, d'Avila Garcez A S. Learning and reasoning with logic tensor networks. In: *Proceedings of Conference of the Italian Association for Artificial Intelligence*, 2016. 334–348
- 125 Badreddine S, d'Avila Garcez A, Serafini L, et al. Logic tensor networks. *Artif Intell*, 2022, 303: 103649
- 126 Winters T, Marra G, Manhaeve R, et al. DeepStochLog: neural stochastic logic programming. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 10090–10100
- 127 Ahmed K, Wang E, Chang K W, et al. Neuro-symbolic entropy regularization. In: *Proceedings of Uncertainty in Artificial Intelligence*, 2022. 43–53
- 128 Ahmed K, Teso S, Chang K W, et al. Semantic probabilistic layers for neuro-symbolic learning. In: *Proceedings of Advances in Neural Information Processing Systems*, 2022. 29944–29959
- 129 Vergari A, Choi Y, Peharz R, et al. Probabilistic circuits: representations, inference, learning and applications. In: *Proceedings of Tutorial at the 34th AAAI Conference on Artificial Intelligence*, 2020
- 130 Yang Z, Lee J, Park C. Injecting logical constraints into neural networks via straight-through estimators. In: *Proceedings of International Conference on Machine Learning*, 2022. 25096–25122
- 131 Xu W, Wang J, Xie L, et al. LogicMP: a neuro-symbolic approach for encoding first-order logic constraints. 2023. ArXiv:2309.15458
- 132 Grespan M M, Gupta A, Srikumar V. Evaluating relaxations of logic for neural networks: a comprehensive study. 2021. ArXiv:2107.13646
- 133 Marconato E, Teso S, Passerini A. Neuro-symbolic reasoning shortcuts: mitigation strategies and their limitations. 2023. ArXiv:2303.12578
- 134 Flach P A. Classifier calibration. In: *Proceedings of Encyclopedia of Machine Learning and Data Mining*, 2016
- 135 Müller R, Kornblith S, Hinton G E. When does label smoothing help? In: *Proceedings of Advances in Neural Information Processing Systems*, 2019
- 136 Liu B, Ayed I B, Galdran A, et al. The devil is in the margin: margin-based label smoothing for network calibration. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 80–88
- 137 Zhang H, Cisse M, Dauphin Y N, et al. Mixup: beyond empirical risk minimization. 2017. ArXiv:1710.09412
- 138 Hendrycks D, Mu N, Cubuk E D, et al. AugMix: a simple data processing method to improve robustness and uncertainty. 2019. ArXiv:1912.02781
- 139 Joo T, Chung U. Revisiting explicit regularization in neural networks for well-calibrated predictive uncertainty. 2020. ArXiv:2006.06399
- 140 Wei H, Xie R, Cheng H, et al. Mitigating neural network overconfidence with logit normalization. In: *Proceedings of International Conference on Machine Learning*, 2022. 23631–23644
- 141 Cheng J, Vasconcelos N. Calibrating deep neural networks by pairwise constraints. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 13709–13718
- 142 Wang Y, Li B, Che T, et al. Energy-based open-world uncertainty modeling for confidence calibration. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 9302–9311
- 143 Grathwohl W, Wang K C, Jacobsen J H, et al. Your classifier is secretly an energy based model and you should treat it like one. 2019. ArXiv:1912.03263
- 144 Liu W, Wang X, Owens J, et al. Energy-based out-of-distribution detection. In: *Proceedings of Advances in Neural Information Processing Systems*, 2020. 33: 21464–21475
- 145 Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection. In: *Proceedings of the IEEE International*

- Conference on Computer Vision, 2017. 2980–2988
- 146 Ahmed K, Chang K W, van den Broeck G. Semantic strengthening of neuro-symbolic learning. In: Proceedings of International Conference on Artificial Intelligence and Statistics, 2023. 10252–10261
- 147 Bengio Y, Deleu T, Rahaman N, et al. A meta-transfer objective for learning to disentangle causal mechanisms. 2019. ArXiv:1901.10912
- 148 Yao L, Chu Z, Li S, et al. A survey on causal inference. *ACM Trans Knowl Discov Data*, 2021, 15: 1–46
- 149 Filho T S, Song H, Perello-Nieto M, et al. Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Mach Learn*, 2023, 112: 3211–3260
- 150 Getoor L, Taskar B. *Introduction to Statistical Relational Learning*. Cambridge: MIT Press, 2007
- 151 Kearns M, Li M, Valiant L. Learning Boolean formulas. *J ACM*, 1994, 41: 1298–1328
- 152 Blei D M, Kucukelbir A, McAuliffe J D. Variational inference: a review for statisticians. *J Am Statistical Assoc*, 2017, 112: 859–877
- 153 Luengo D, Martino L, Bugallo M, et al. A survey of Monte Carlo methods for parameter estimation. *EURASIP J Adv Signal Process*, 2020, 2020: 25
- 154 Evans R, Grefenstette E. Learning explanatory rules from noisy data. *J Artif Intell Res*, 2018, 61: 1–64
- 155 Shindo H, Nishino M, Yamamoto A. Differentiable inductive logic programming for structured examples. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 5034–5041
- 156 Shindo H, Pfanschilling V, Dhami D S, et al. α ILP: thinking visual scenes as differentiable logic programs. *Mach Learn*, 2023, 112: 1465–1497
- 157 Si X, Raghothaman M, Heo K, et al. Synthesizing datalog programs using numerical relaxation. 2019. ArXiv:1906.00163
- 158 Seung H S, Oppor M, Sompolinsky H. Query by committee. In: Proceedings of the 5th Annual Workshop on Computational Learning Theory, 1992. 287–294
- 159 Yang F, Yang Z, Cohen W W. Differentiable learning of logical rules for knowledge base reasoning. In: Proceedings of Advances in Neural Information Processing Systems, 2017
- 160 Cohen W W. TensorLog: a differentiable deductive database. 2016. ArXiv:1605.06523
- 161 Wang P W, Stepanova D, Domokos C, et al. Differentiable learning of numerical rules in knowledge graphs. In: International Conference on Learning Representations, 2019
- 162 Yang Y, Song L. Learn to explain efficiently via neural logic inductive learning. 2019. ArXiv:1910.02481
- 163 Wang W Y, Cohen W. Scalable statistical relational learning for NLP. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts, 2016. 14–16
- 164 Sadeghian A, Armandpour M, Ding P, et al. DRUM: end-to-end differentiable rule mining on knowledge graphs. In: Proceedings of Advances in Neural Information Processing Systems, 2019
- 165 Cheng K, Liu J, Wang W, et al. RLogic: recursive logical rule learning from knowledge graphs. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022. 179–189
- 166 Qu M, Chen J, Xhonneux L P, et al. RNNLogic: learning logic rules for reasoning on knowledge graphs. 2020. ArXiv:2010.04029
- 167 Rocktäschel T, Riedel S. End-to-end differentiable proving. In: Proceedings of Advances in Neural Information Processing Systems, 2017
- 168 Minervini P, Bosnjak M, Rocktäschel T, et al. Towards neural theorem proving at scale. 2018. ArXiv:1807.08204
- 169 Minervini P, Bošnjak M, Rocktäschel T, et al. Differentiable reasoning on large knowledge bases and natural language. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 5182–5190
- 170 Minervini P, Riedel S, Stenetorp P, et al. Learning reasoning strategies in end-to-end differentiable proving. In: Proceedings of International Conference on Machine Learning, 2020. 6938–6949
- 171 Campero A, Pareja A, Klinger T, et al. Logical rule induction and theory learning using neural theorem proving. 2018. ArXiv:1809.02193
- 172 Glanois C, Jiang Z, Feng X, et al. Neuro-symbolic hierarchical rule induction. In: Proceedings of International Conference on Machine Learning, 2022. 7583–7615
- 173 Wang P W, Donti P, Wilder B, et al. SATNet: bridging deep learning and logical reasoning using a differentiable

- satisfiability solver. In: Proceedings of International Conference on Machine Learning, 2019. 6545–6554
- 174 Wang P W, Kolter J Z. Low-rank semidefinite programming for the MAX2SAT problem. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 1641–1649
- 175 Wang P W, Chang W C, Kolter J Z. The mixing method: coordinate descent for low-rank semidefinite programming. 2017. ArXiv:1706.00476
- 176 Luo Z Q, Ma W K, So A, et al. Semidefinite relaxation of quadratic optimization problems. IEEE Signal Process Mag, 2010, 27: 20–34
- 177 Lim S, Oh E G, Yang H. Learning symmetric rules with SATNet. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 13251–13262
- 178 Li Z, Guo J, Jiang Y, et al. Learning reliable logical rules with SATNet. In: Proceedings of Advances in Neural Information Processing Systems, 2024
- 179 Chang O, Flokas L, Lipson H, et al. Assessing SATNet’s ability to solve the symbol grounding problem. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 1428–1439
- 180 Topan S, Rolnick D, Si X. Techniques for symbol grounding with SATNet. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 20733–20744
- 181 Cook W J. In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation. Princeton: Princeton University Press, 2015
- 182 Meng T, Chang K W. An integer linear programming framework for mining constraints from data. In: Proceedings of International Conference on Machine Learning, 2021. 7619–7631
- 183 Paulus A, Rolínek M, Musil V, et al. CombOptNet: fit the right NP-hard problem by learning integer programming constraints. In: Proceedings of International Conference on Machine Learning, 2021. 8443–8453
- 184 Nandwani Y, Ranjan R, Singla P, et al. A solver-free framework for scalable learning in neural ILP architectures. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 7972–7986
- 185 Dinh T P, Thi H A L. Convex analysis approach to D.C. programming: theory, algorithms and applications. Acta Math Vietnam, 1997, 22: 289–355
- 186 Thi H A L, Dinh T P. DC programming and DCA: thirty years of developments. Math Program, 2018, 169: 5–68
- 187 Eén N, Sörensson N. Translating pseudo-Boolean constraints into SAT. J Satisfiability, Boolean Modeling Comput, 2006, 2: 1–26
- 188 Bailleux O, Boufkhad Y, Roussel O, et al. A translation of pseudo-Boolean constraints to SAT. J Satisfiability, Boolean Modeling Comput, 2006, 2: 191–200
- 189 Steinbach B, Kohut R. Neural networks – a model of Boolean functions. In: Proceedings of the 5th International Workshop on Boolean Problems, 2002. 223–240
- 190 Payani A, Fekri F. Learning algorithms via neural logic networks. 2019. ArXiv:1904.01554
- 191 Dong H, Mao J, Lin T, et al. Neural logic machines. 2019. ArXiv:1904.11694
- 192 Payani A, Fekri F. Inductive logic programming via differentiable deep neural logic networks. 2019. ArXiv:1906.03523
- 193 Zimmer M, Feng X, Glanois C, et al. Differentiable logic machines. 2021. ArXiv:2102.11529
- 194 Riegel R, Gray A, Luus F, et al. Logical neural networks. 2020. ArXiv:2006.13155
- 195 Sen P, de Carvalho B W, Riegel R, et al. Neuro-symbolic inductive logic programming with logical neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2022. 8212–8219
- 196 Sourek G, Aschenbrenner V, Zelezny F, et al. Lifted relational neural networks: efficient learning of latent relational structures. J Artif Intell Res, 2018, 62: 69–100
- 197 Palm R, Paquet U, Winther O. Recurrent relational networks. In: Proceedings of Advances in Neural Information Processing Systems, 2018
- 198 Chen D, Bai Y, Zhao W, et al. Deep reasoning networks: thinking fast and slow. 2019. ArXiv:1906.00855
- 199 Bai Y, Chen D, Gomes C P. CLR-DRNets: curriculum learning with restarts to solve visual combinatorial games. In: Proceedings of the 27th International Conference on Principles and Practice of Constraint Programming, 2021
- 200 Zhou Z-H. Abductive learning: towards bridging machine learning and logical reasoning. Sci China Inf Sci, 2019, 62: 076101
- 201 Wang K, Tsamoura E, Roth D. On learning latent models with multi-instance weak supervision. In: Proceedings of

- Advances in Neural Information Processing Systems, 2024
- 202 Köppen M. The curse of dimensionality. In: Proceedings of the 5th Online World Conference on Soft Computing in Industrial Applications, 2000. 4–8
- 203 Neal R M. MCMC using hamiltonian dynamics. In: Handbook of Markov Chain Monte Carlo. Boca Raton: Chapman & Hall/CRC, 2011
- 204 Zhang D, Malkin N, Liu Z, et al. Generative flow networks for discrete probabilistic modeling. In: Proceedings of International Conference on Machine Learning, 2022. 26412–26428
- 205 Jiang A Q, Li W, Tworkowski S, et al. Thor: wielding hammers to integrate language models and automated theorem provers. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 8360–8373
- 206 First E, Rabe M N, Ringer T, et al. Baldur: whole-proof generation and repair with large language models. In: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2023. 1229–1241
- 207 Hendrycks D, Burns C, Basart S, et al. Measuring massive multitask language understanding. 2020. ArXiv:2009.03300
- 208 Chen M, Tworek J, Jun H, et al. Evaluating large language models trained on code. 2021. ArXiv:2107.03374
- 209 Lightman H, Kosaraju V, Burda Y, et al. Let's verify step by step. 2023. ArXiv:2305.20050
- 210 Chu Z, Chen J, Chen Q, et al. A survey of chain of thought reasoning: advances, frontiers and future. 2023. ArXiv:2309.15402
- 211 Parmar M, Patel N, Varshney N, et al. Towards systematic evaluation of logical reasoning ability of large language models. 2024. ArXiv:2404.15522
- 212 Wan Y, Wang W, Yang Y, et al. A & B == B & A: triggering logical reasoning failures in large language models. 2024. ArXiv:2401.00757
- 213 Pan L, Albalak A, Wang X, et al. Logic-LM: empowering large language models with symbolic solvers for faithful logical reasoning. 2023. ArXiv:2305.12295
- 214 Zhou J P, Staats C, Li W, et al. Don't trust: verify-grounding llm quantitative reasoning with autoformalization. 2024. ArXiv:2403.18120
- 215 Wu Y, Jiang A Q, Li W, et al. Autoformalization with large language models. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 32353–32368
- 216 Blanchette J C, Bulwahn L, Nipkow T. Automatic proof and disproof in Isabelle/HOL. In: Proceedings of the 8th International Symposium on Frontiers of Combining Systems, 2011. 12–27
- 217 Trinh T H, Wu Y, Le Q V, et al. Solving olympiad geometry without human demonstrations. Nature, 2024, 625: 476–482
- 218 Cornelio C, Diab M. Recover: a neuro-symbolic framework for failure detection and recovery. 2024. ArXiv:2404.00756
- 219 Kwon J, Levine S, Tenenbaum J B. Neuro-symbolic models of human moral judgment: LLMs as automatic feature extractors. In: Proceedings of the 40th International Conference on Machine Learning, 2023
- 220 Zhang H, Huang J, Li Z, et al. Improved logical reasoning of language models via differentiable symbolic programming. 2023. ArXiv:2305.03742
- 221 Luo L, Ju J, Xiong B, et al. ChatRule: mining logical rules with large language models for knowledge graph reasoning. 2023. ArXiv:2309.01538
- 222 Gao G, Ju H, Jiang J, et al. A semantic search engine for Mathlib4. 2024. ArXiv:2403.13310
- 223 Azerbayev Z, Piotrowski B, Avigad J. ProofNet: a benchmark for autoformalizing and formally proving undergraduate-level mathematics problems. In: Proceedings of the 2nd MATH-AI Workshop, 2022
- 224 Gilpin L H, Ilievski F. Neuro-symbolic reasoning in the traffic domain. J AI Res, 2021, 15: 123–145
- 225 Sun J, Sun H, Han T, et al. Neuro-symbolic program search for autonomous driving decision module design. In: Proceedings of Conference on Robot Learning, 2021. 21–30
- 226 Sharifi I, Yildirim M, Fallah S. Towards safe autonomous driving policies using a neuro-symbolic deep reinforcement learning approach. 2023. ArXiv:2307.01316
- 227 Myers B A. Visual programming, programming by example, and program visualization: a taxonomy. SIGCHI Bull, 1986, 17: 59–66

- 228 Gupta T, Kembhavi A. Visual programming: compositional visual reasoning without training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 14953–14962
- 229 Cho J, Zala A, Bansal M. Visual programming for text-to-image generation and evaluation. 2023. ArXiv:2305.15328
- 230 Yang K, Swope A, Gu A, et al. LeanDojo: theorem proving with retrieval-augmented language models. In: Proceedings of Advances in Neural Information Processing Systems, 2024
- 231 Limperg J, From A H. Aesop: white-box best-first proof search for lean. In: Proceedings of the 12th ACM SIGPLAN International Conference on Certified Programs and Proofs, 2023. 253–266
- 232 Tao T. Embracing change and resetting expectations. AI Anthology, 2023. <https://unlocked.microsoft.com/ai-anthology/terence-tao/>

Neuro-symbolic systems: a perspective of uncertainty management

Zenan LI^{1,2}, Yuan YAO^{1,2}, Xiaoxing MA^{1,2*} & Jian LÜ^{1,2}

1. State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210023, China

2. School of Computer Science, Nanjing University, Nanjing 210023, China

* Corresponding author. E-mail: xxm@nju.edu.cn

Abstract As an intelligent software paradigm, neuro-symbolic systems – integrating neural networks and symbolic rules – have garnered increasing attention, with explosive growth in related research. However, existing approaches still struggle with the effective and efficient construction of large-scale neuro-symbolic systems. One key factor contributing to the unsatisfactory performance of neuro-symbolic learning is the incompatible management of uncertainty during both neural network training and symbolic rule learning processes. This paper aims to provide a systematic, consistent, and rigorous perspective to facilitate the discussion and analysis of the technical challenges in “end-to-end” neuro-symbolic learning. Based on this perspective, the paper organizes and reviews the ideas, progress, and current issues in related work, while also looking ahead to future directions in the development of this field.

Keywords neuro-symbolic learning, neural symbol grounding, reasoning shortcuts, constraint synthesis