Logic Tensor Networks

逻辑张量网络

Samy Badreddine [a,b,*] , Artur d′ Avila Garce z[c] , Luciano Serafini [d] , Michael Spranger [a,b]
Samy Badreddine [a,b,*] , Artur d′ Avila Garce z[c] , Luciano Serafini [d] , Michael Spranger [a,b]
[a] Sony Computer Science Laboratories Inc,3-14-13 Higashigotanda,141-0022, Tokyo, Japan

[a] 索尼计算机科学研究所，东京都港区东五反田 3-14-13，141-0022，日本

[b] Sony AI Inc,1-7-1 Konan,108-0075, Tokyo, Japan

[b] 索尼 AI 公司，东京都港区海岸 1-7-1，108-0075，日本

[c] City, University of London, Northampton Square, EC1V 0HB, London, United Kingdom

[c] 伦敦城市大学，Northampton Square，EC1V 0HB，伦敦，英国

[d] Fondazione Bruno Kessler, Via Sommarive 18,38123, Trento, Italy

[d] Bruno Kessler 基金会，Via Sommarive 18，38123，特伦托，意大利

Abstract

摘要

Attempts at combining logic and neural networks into neurosymbolic approaches have been on the increase in recent years. In a neurosymbolic system, symbolic knowledge assists deep learning, which typically uses a sub-symbolic distributed representation, to learn and reason at a higher level of abstraction. We present Logic Tensor Networks (LTN), a neurosymbolic framework that supports querying, learning and reasoning with both rich data and abstract knowledge about the world. LTN introduces a fully differentiable logical language, called Real Logic, whereby the elements of a first-order logic signature are grounded onto data using neural computational graphs and first-order fuzzy logic semantics. We show that LTN provides a uniform language to represent and compute efficiently many of the most important AI tasks such as multi-label classification, relational learning, data clustering, semi-supervised learning, regression, embedding learning and query answering. We implement and illustrate each of the above tasks with several simple explanatory examples using TensorFlow 2. The results indicate that LTN can be a general and powerful framework for neurosymbolic AI.

近年来，将逻辑与神经网络结合成神经符号方法的尝试不断增加。在神经符号系统中，符号知识协助深度学习，后者通常使用子符号分布式表示，以在更高层次的抽象上进行学习和推理。我们提出了逻辑张量网络（LTN），这是一个支持使用丰富数据和关于世界的抽象知识进行查询、学习和推理的神经符号框架。LTN 引入了一种完全可微分的逻辑语言，称为实数逻辑，通过使用神经网

络计算图和一阶模糊逻辑语义，将一阶逻辑签名中的元素映射到数据上。我们展示了 LTN 提供了一种统一的语言，可以有效地表示和计算许多最重要的 AI 任务，如多标签分类、关系学习、数据聚类、半监督学习、回归、嵌入学习和查询回答。我们使用 TensorFlow 2 实现了上述每个任务，并通过几个简单的解释性示例进行了说明。结果表明，LTN 可以是神经符号 AI 的一种通用而强大的框架。

Keywords: Neurosymbolic AI, Deep Learning and Reasoning, Many-valued Logics.

关键词：神经符号 AI，深度学习与推理，多值逻辑。

1. Introduction

1. 引言

Artificial Intelligence (AI) agents are required to learn from their surroundings and reason about what has been learned to make decisions, act in the world, or react to various stimuli. The latest Machine Learning (ML) has adopted mostly a pure sub-symbolic learning approach. Using distributed representations of entities, the latest ML performs quick decision-making without building a comprehensible model of the world. While achieving impressive results in computer vision, natural language, game playing, and multimodal learning, such approaches are known to be data inefficient and to struggle at out-of-distribution generalization. Although the use of appropriate inductive biases can alleviate such shortcomings, in general, sub-symbolic models lack comprehensibility. By contrast, symbolic AI is based on rich, high-level representations of the world that use human-readable symbols. By rich knowledge, we refer to logical representations which are more expressive than propositional logic or propositional probabilistic approaches, and which can express knowledge using full first-order logic, including universal and existential quantification $(\forall x \text{and} \exists y)$, arbitrary n -ary relations over variables, e.g. $R(x, y, z, \ldots)$, and function symbols, e.g. father $\operatorname{Of}(x), x+y$, etc. Symbolic AI has achieved success at theorem proving, logical inference, and verification. However, it also has shortcomings when dealing with incomplete knowledge. It can be inefficient with large amounts of inaccurate data and lack robustness to outliers. Purely symbolic decision algorithms usually have high computational complexity making them impractical for the real world. It is now clear that the predominant approach to ML, where learning is based on recognizing the latent structures hidden in the data, is insufficient and may benefit from symbolic AI [17]. In this context, neurosymbolic AI, which stems from neural networks and symbolic AI, attempts to combine the strength of both paradigms (see [16, 40, 54] for recent surveys). That is to say, combine reasoning with complex representations of knowledge (knowledge-bases, semantic networks, ontologies, trees, and graphs) with learning from complex data (images, time series, sensorimotor data, natural language). Consequently, a main challenge for neurosymbolic AI is the grounding of symbols, including constants, functional and relational symbols, into real data, which is akin to the longstanding symbol grounding problem [30].

人工智能（AI）代理需要从周围环境中学习并对其所学内容进行推理，以做出决策、在世界上行动或对各种刺激做出反应。最新的机器学习（ML）主要采用纯符号下学习的方法。通过使用实体的分布式表示，最新的 ML 能够快速做出决策，而无需构建一个可理解的世界模型。尽管在计算机视觉、自然语言、游戏和多模态学习方面取得了令人印象深刻的结果，但这些方法被认为在数据效率和分布外泛化方面存在困难。尽管使用适当的归纳偏置可以缓解这些不足，但一般来说，符号下模型缺乏可理解性。相比之下，符号 AI 基于丰富的、高级的世界表示，使用人类可读的符号。通过丰富的知识，我们指的是比命题逻辑或命题概率方法更具表现力的逻辑表示，它可以使用完整的一阶逻辑来表达知识，包括全称和存在量化 $(\forall x \text{and} \exists y)$ ，任意的 n 元关系，例如 $R(x, y, z, \ldots)$ ，以及函数符号，例如父亲 $\mathrm{Of}(x), x+y$ 等。符号 AI 在定理证明、逻辑推理和验证方面取得了成功。然而，在处理不完整知识时也存在不足。它可能在处理大量不准确的数据时效率低下，并且对异常值缺乏鲁棒性。纯符号决策算法通常具有很高的计算复杂度，使其在现实世界中不切实际。现在很明显，当前 ML 的主要方法，即基于识别数据中隐藏的潜在结构的学习，是不够的，并可能从符号 AI 中受益 [17]。在这种背景下，神经符号 AI，源自神经网络和符号 AI，试图结合这两种范式的优势（参见 [16, 40, 54] 的最近调查）。也就是说，结合推理与复杂知识表示（知识库、语义网络、本体、树和图）的学习，以及从复杂数据（图像、时间序列、传感器运动数据、自然语言）中学习。因此，神经符号 AI 的一个主要挑战是将符号，包括常量、函数和关系符号，映射到实际数据中，这类似于长期存在的符号映射问题 [30]。

Logic Tensor Networks (LTN) are a neurosymbolic framework and computational model that supports learning and reasoning about data with rich knowledge. In LTN, one can represent and effectively compute the most important tasks of deep learning with a fully differentiable first-order logic language, called Real Logic, which adopts infinitely many truth-values in the interval $[0,1]$ [22, 25] . In particular, LTN supports the specification and computation of the following AI tasks uniformly using the same language: data clustering, classification, relational learning, query answering, semi-supervised learning, regression, and embedding learning.

逻辑张量网络（Logic Tensor Networks，LTN）是一种神经符号框架和计算模型，支持使用丰富的知识对数据进行学习和推理。在 LTN 中，可以使用完全可微的一阶逻辑语言——实数逻辑（Real Logic）来表示和有效计算深度学习的最重要任务，该语言在区间 $[0,1]$ [22, 25] 中采用无限多的真值。特别是，LTN 支持使用同一种语言统一规范和计算以下 AI 任务：数据聚类、分类、关系学习、查询回答、半监督学习、回归和嵌入学习。

*Corresponding author
* 通讯作者
Email addresses: badreddine. samy@gmail.com (Samy Badreddine), a. garcez@city.ac.uk (Artur d'Avila Garcez),
电子邮件地址：badreddine.samy@gmail.com (Samy Badreddine), a.garcez@city.ac.uk (Artur d'Avila Garcez),
serafini@fbk.eu (Luciano Serafini), michael . spranger@sony . com (Michael Spranger)
serafini@fbk.eu (Luciano Serafini), michael.spranger@sony.com (Michael Spranger)

LTN and Real Logic were first introduced in [62]. Since then, LTN has been applied to different AI tasks involving perception, learning, and reasoning about relational knowledge. In [18, 19], LTN was applied to semantic image interpretation whereby relational knowledge about objects was injected into deep networks for object relationship detection. In [6], LTN was evaluated on its capacity to perform reasoning about ontological knowledge. Furthermore, [7] shows how LTN can be used to learn an embedding of concepts into a latent real space by taking into consideration ontological knowledge about such concepts. In [3], LTN is used to annotate a reinforcement learning environment with prior knowledge and incorporate latent information into an agent. In [42], authors embed LTN in a state-of-the-art convolutional object detector. Extensions and generalizations of LTN have also been proposed in the past years, such as LYRICS [47] and Differentiable Fuzzy Logic (DFL) [68,69]. LYRICS provides an input language allowing one to define background knowledge using a first-order logic where predicate and function symbols are grounded onto any computational graph. DFL analyzes how a large collection of fuzzy logic operators behave in a differentiable learning setting. DFL also introduces new semantics for fuzzy logic implications called sigmoidal implications, and it shows that such semantics outperform other semantics in several semi-supervised machine learning tasks.

LTN 和 Real Logic 最初在 [62] 中引入。自那时起，LTN 已被应用于涉及感知、学习和关系知识推理的不同 AI 任务。在 [18, 19] 中，LTN 被应用于语义图像解释，其中关于对象的关系知识被注入深度网络以进行对象关系检测。在 [6] 中，评估了 LTN 在执行关于本体知识的推理方面的能力。此外，[7] 展示了如何使用 LTN 通过考虑关于这些概念的本体知识来学习概念嵌入到潜在实数空间中。在 [3] 中，LTN 被用于用先验知识注释强化学习环境并将潜在信息融入代理中。在 [42] 中，作者将 LTN 嵌入到最先进的卷积对象检测器中。在过去的几年里，也提出了 LTN 的扩展和泛化，例如 LYRICS [47] 和可微分模糊逻辑（DFL）[68,69]。LYRICS 提供了一个输入语言，允许使用一阶逻辑定义背景知识，其中谓词和函数符号被映射到任何计算图上。DFL 分析了大量模糊逻辑操作符在可微分学习环境中的行为。DFL 还引入了模糊逻辑蕴涵的新语义，称为 sigmoidal 蕴涵，并展示了这种语义在几个半监督机器学习任务中优于其他语义。

This paper provides a thorough description of the full formalism and several extensions of LTN. We show using an extensive set of explanatory examples, how LTN can be applied to solve many ML tasks with the help of logical knowledge. In particular, the earlier versions of LTN have been extended with: (1) Explicit domain declaration: constants, variables, functions and predicates are now domain typed (e.g. the constants John and Paris can be from the domain of person and city, respectively). The definition of structured domains is also possible (e.g. the domain couple can be defined as the Cartesian product of two domains of persons); (2) Guarded quantifiers: guarded universal and existential quantifiers now allow the user to limit the quantification to the elements that satisfy some Boolean condition, e.g. $\forall x : \mathrm{age}\,(x) < 10\ (\mathrm{playsPiano}\ (x) \to \mathrm{enfantProdige}\ (x)\ )$ restricts the quantification to the cases where age is lower than 10; (3) Diagonal quantification: Diagonal quantification allows the user to write statements about specific tuples extracted in order from n variables. For example, if the variables capital and country both have k instances such that the i -th instance of capital corresponds to the i -th instance of country, one can write $\forall$ Diag(capital, country) capitalOf(capital, country).

本文详细描述了 LTN 的完整形式主义及其几个扩展。我们通过一组广泛的解释性示例展示了如何利用逻辑知识将 LTN 应用于解决许多机器学习任务。特别是，LTN 的早期版本已经扩展了以下内容：（1）显式域声明：常量、变量、函数和谓词现在具有域类型（例如，常量 John 和 Paris 可以分别属于人和城市域）。也可以定义结构化域（例如，可以将伴侣域定义为两个人域的笛卡尔积）；（2）守卫量词：守卫的全称和存在量词现在允许用户将量化限制为满足某些布尔条件的元素，例如 $\forall x : \text{age}\,(x) < 10$ (playsPiano $(x) \rightarrow$ enfantProdige (x)) 将量化限制在年龄小于 10 的案例中；（3）对角量化：对角量化允许用户编写关于从 n 变量中按顺序提取的特定元组的陈述。例如，如果变量 capital 和 country 都有 k 实例，使得 capital 的第 i 个实例对应于 country 的第 i 个实例，则可以编写 $\forall$ Diag(capital, country) capitalOf(capital, country)。

Inspired by the work of [69], this paper also extends the product t-norm configuration of LTN with the generalized mean aggregator, and it introduces solutions to the vanishing or exploding gradient problems. Finally, the paper formally defines a semantic approach to refutation-based reasoning in Real Logic to verify if a statement is a logical consequence of a knowledge base. Example 4.8 proves that this new approach can better capture logical consequences compared to simply querying unknown formulas after learning (as done in [6]).

受 [69] 的工作启发，本文还扩展了 LTN 的乘积 t-范数配置，加入了广义均值聚合器，并引入了解决梯度消失或爆炸问题的方法。最后，本文正式定义了一种基于反驳的推理的语义方法，用于在实逻辑中验证一个陈述是否是知识库的逻辑后果。示例 4.8 证明了这种方法与在学习后简单地查询未知公式（如 [6] 中所做）相比，能更好地捕获逻辑后果。

The new version of LTN has been implemented in TensorFlow 2 [1]. Both the LTN library and the code for the examples used in this paper are available at https://github.com/ logictensornetworks/logictensornetworks

新版本的 LTN 已在 TensorFlow 2 中实现 [1]。LTN 库以及本文中使用的示例代码可在 https://github.com/logictensornetworks/logictensornetworks 获取。

The remainder of the paper is organized as follows: In Section 2, we define and illustrate Real Logic as a fully-differentiable first-order logic. In Section 3, we specify learning and reasoning in Real Logic and its modeling into deep networks with Logic Tensor Networks (LTN). In Section 4 we illustrate the reach of LTN by investigating a range of learning problems from clustering to embedding learning. In Section 5, we place LTN in the context of the latest related work in neurosymbolic AI. In Section 6 we conclude and discuss directions for future work. The Appendix contains information about the implementation of LTN in TensorFlow 2, experimental set-ups, the different options for the differentiable logic operators, and a study of their relationship with gradient computations.

本文的其余部分组织如下：在第 2 节中，我们定义并说明了作为全微分一阶逻辑的 Real Logic。在第 3 节中，我们指定了在 Real Logic 中的学习和推理方法，以及将其建模为深度网络中的逻辑张量网络（LTN）。在第 4 节中，我们通过研究从聚类到嵌入学习的一系列学习问题，展示了 LTN 的应用范围。在第 5