Logic Tensor Networks: Deep Learning and Logical Reasoning from Data and Knowledge*

逻辑张量网络：基于数据与知识的深度学习与逻辑推理*

Luciano Serafini [1] and Artur d’Avila Garcez [2]
Luciano Serafini [1] 和 Artur d’Avila Garcez [2]

[1] Fondazione Bruno Kessler,Trento,Italy,serafini@fbk.eu [2] City University London,UK,a.garcez@city.ac.uk
[1] 意大利特伦托布鲁诺 · 凯斯勒基金会，serafini@fbk.eu [2] 英国伦敦城市大学，a.garcez@city.ac.uk

Abstract. We propose Logic Tensor Networks: a uniform framework for integrating automatic learning and reasoning. A logic formalism called Real Logic is defined on a first-order language whereby formulas have truth-value in the interval $[0, 1]$ and semantics defined concretely on the domain of real numbers. Logical constants are interpreted as feature vectors of real numbers. Real Logic promotes a well-founded integration of deductive reasoning on a knowledge-base and efficient data-driven relational machine learning. We show how Real Logic can be implemented in deep Tensor Neural Networks with the use of Google's TEN-SORFLOW ™ primitives. The paper concludes with experiments applying Logic Tensor Networks on a simple but representative example of knowledge completion.
摘要。我们提出逻辑张量网络（Logic Tensor Networks）：一个整合自动学习与推理的统一框架。定义了一种称为实值逻辑（Real Logic）的逻辑形式体系，基于一阶语言，其中公式的真值位于区间$[0, 1]$内，语义具体定义在实数域上。逻辑常量被解释为实数特征向量。实值逻辑促进了基于知识库的演绎推理与高效数据驱动的关系机器学习的良好融合。我们展示了如何利用谷歌的TENSORFLOW ™ 原语在深度张量神经网络中实现实值逻辑。本文最后通过一个简单但具有代表性的知识补全示例，展示逻辑张量网络的应用实验。

Keywords: Knowledge Representation, Relational Learning, Tensor Networks, Neural-Symbolic Computation, Data-driven Knowledge Completion.
关键词：知识表示，关系学习，张量网络，神经符号计算，数据驱动的知识补全。

1 1 Introduction

2 1 引言

The recent availability of large-scale data combining multiple data modalities, such as image, text, audio and sensor data, has opened up various research and commercial opportunities, underpinned by machine learning methods and techniques [5,12, 17, 18]. In particular, recent work in machine learning has sought to combine logical services, such as knowledge completion, approximate inference, and goal-directed reasoning with data-driven statistical and neural network-based approaches. We argue that there are great possibilities for improving the current state of the art in machine learning and artificial intelligence (AI) thought the principled combination of knowledge representation, reasoning and learning. Guha's recent position paper [15] is a case in point, as it advocates a new model theory for real-valued numbers. In this paper, we take inspiration from such recent work in AI, but also less recent work in the area of neural-symbolic integration [8, 10, 11] and in semantic attachment and symbol grounding [4] to achieve a vector-based representation which can be shown adequate for integrating machine learning and reasoning in a principled way.
近年来，结合多种数据模态（如图像、文本、音频和传感器数据）的海量数据的可用性，开启了由机器学习方法和技术支撑的多种研究与商业机会[5,12,17,18]。特别是，机器学习领域的最新工作致力于将逻辑服务（如知识补全、近似推理和目标导向推理）与数据驱动的统计及神经网络方法相结合。我们认为，通过知识表示、推理与学习的原则性结合，有望显著提升当前机器学习和人工智能（AI）的技术水平。Guha最近的立场论文[15]即为一例，主张为实值数构建新的模型理论。本文从此类AI最新研究中汲取灵感，同时借鉴神经符号集成（neural-symbolic integration）

[8,10,11]及语义附着与符号锚定[4]领域的较早研究，旨在实现一种基于向量的表示，该表示被证明适合以原则性方式整合机器学习与推理。

- The first author acknowledges the Mobility Program of FBK, for supporting a long term visit at City University London. He also acknowledges NVIDIA Corporation for supporting this research with the donation of a GPU.
- 第一作者感谢FBK的流动项目支持其在伦敦城市大学的长期访问。同时感谢NVIDIA公司通过捐赠GPU支持本研究。

This paper proposes a framework called Logic Tensor Networks (LTN) which integrates learning based on tensor networks [26] with reasoning using first-order many-valued logic [6],all implemented in TENSORFLOW ™ [13]. This enables,for the first time, a range of knowledge-based tasks using rich knowledge representation in first-order logic (FOL) to be combined with efficient data-driven machine learning based on the manipulation of real-valued vectors [1] . Given data available in the form of real-valued vectors, logical soft and hard constraints and relations which apply to certain subsets of the vectors can be specified compactly in first-order logic. Reasoning about such constraints can help improve learning, and learning from new data can revise such constraints thus modifying reasoning. An adequate vector-based representation of the logic, first proposed in this paper, enables the above integration of learning and reasoning, as detailed in what follows.
本文提出了一个名为逻辑张量网络（Logic Tensor Networks, LTN）的框架，将基于张量网络[26]的学习与基于一阶多值逻辑[6]的推理结合起来，所有实现均基于TENSORFLOW ™ [13]。这首次使得利用一阶逻辑（FOL）中丰富的知识表示进行一系列基于知识的任务与基于实值向量操作的高效数据驱动机器学习相结合成为可能。给定以实值向量形式存在的数据，可以在一阶逻辑中紧凑地指定适用于向量某些子集的逻辑软约束和硬约束及其关系。对这些约束的推理有助于提升学习效果，而从新数据中学习则可修正这些约束，从而影响推理。本文首次提出的适当的基于向量的逻辑表示，实现了上述学习与推理的整合，具体内容如下所述。

We are interested in providing a computationally adequate approach to implementing learning and reasoning [28] in an integrated way within an idealized agent. This agent has to manage knowledge about an unbounded, possibly infinite, set of objects $O=\{o_1,o_2,\ldots\}$. Some of the objects are associated with a set of quantitative attributes,represented by an n -tuple of real values $\mathcal{G}(o_i)\in\mathbb{R}^n$,which we call grounding. For example, a person may have a grounding into a 4-tuple containing some numerical representation of the person's name, her height, weight, and number of friends in some social network. Object tuples can participate in a set of relations $\mathcal{R}=\{R_1,\ldots,R_k\}$, with $R_i\subseteq O^{\alpha(R_i)}$,where $\alpha(R_i)$ denotes the arity of relation R_i . We presuppose the existence of a latent (unknown) relation between the above numerical properties, i.e. groundings,and partial relational structure $\mathcal{R}$ on O . Starting from this partial knowledge, an agent is required to: (i) infer new knowledge about the relational structure on the objects of O ; (ii) predict the numerical properties or the class of the objects in O .
我们致力于提供一种计算上充分的方法，以在理想化智能体中以集成方式实现学习和推理[28]。该智能体必须管理关于一个无界、可能无限的对象集合$O=\{o_1,o_2,\ldots\}$的知识。其中一些对象关联着一组定量属性，由一个实数n元组$\mathcal{G}(o_i)\in\mathbb{R}^n$表示，我们称之为基底（grounding）。例如，一个人可能有一个4元组的基底，包含该人的姓名的某种数值表示、身高、体重以及某社交网络中朋友的数量。对象元组可以参与一组关系$\mathcal{R}=\{R_1,\ldots,R_k\}$，其$R_i\subseteq O^{\alpha(R_i)}$，其中$\alpha(R_i)$表示关系$R_i$的元数。我们假设上述数值属性（即基底）之间存在潜在（未知）关系，以及在O上的部分关系结构$\mathcal{R}$。基于这些部分知识，智能体需要：（i）推断关于O对象的关系结构的新知识；（ii）预测O中对象的数值属性或类别。

Classes and relations are not normally independent. For example, it may be the case that if an object x is of class $C, C(x)$,and it is related to another object y through relation $R(x, y)$ then this other object y should be in the same class $C(y)$. In logic: $\forall x \exists y((C(x) \wedge R(x, y)) \rightarrow C(y))$. Whether or not $C(y)$ holds will depend on the application: through reasoning,one may derive $C(y)$ where otherwise there might not have been evidence of $C(y)$ from training examples only; through learning,one may need to revise such a conclusion once examples to the contrary become available. The vectorial representation proposed in this paper permits both reasoning and learning as exemplified above and detailed in the next section.
类别和关系通常不是独立的。例如，若对象x属于类别$C, C(x)$，且通过关系$R(x, y)$与另一对象y相关联，则该对象y应属于同一类别$C(y)$。用逻辑表示为：$\forall x \exists y((C(x) \wedge R(x, y)) \rightarrow C(y))$。是否成立$C(y)$取决于具体应用：通过推理，可以导出$C(y)$，即使仅凭训练样本可能没有该证据；通过学习，一旦出现相反的样本，可能需要修正该结论。本文提出的向量表示允许如上所示的推理和学习，具体细节见下一节。

The above forms of reasoning and learning are integrated in a unifying framework, implemented within tensor networks, and exemplified in relational domains combining data and relational knowledge about the objects. It is expected that, through an adequate integration of numerical properties and relational knowledge, differently from the immediate related literature [9,2,1] ,the framework introduced in this paper will be capable of combining in an effective way first-order logical inference on open domains with efficient relational multi-class learning using tensor networks.
上述推理和学习形式被整合在一个统一框架中，该框架通过张量网络实现，并在结合对象数据和关系知识的关系域中得到示例。预计通过数值属性与关系知识的充分整合，有别于现有相关文献[9,2,1]，本文引入的框架能够有效结合开放域上的一阶逻辑推理与利用张量网络的高效关系多类别学习。

The main contribution of this paper is two-fold. It introduces a novel framework for the integration of learning and reasoning which can take advantage of the representational power of (multi-valued) first-order logic, and it instantiates the framework using tensor networks into an efficient implementation which shows that the proposed vector-based representation of the logic offers an adequate mapping between symbols and their real-world manifestations, which is appropriate for both rich inference and learning from examples.
本文的主要贡献有两方面。其一，提出了一个学习与推理整合的新框架，能够利用（多值）一阶逻辑的表达能力；其二，通过张量网络实例化该框架，形成高效实现，展示了所提基于向量的逻辑表示在符号与其现实表现之间提供了恰当映射，适用于丰富的推理和基于示例的学习。

[1] In practice,FOL reasoning including function symbols is approximated through the usual iterative deepening of clause depth.
[1] 实际中，包含函数符号的一阶逻辑推理通过通常的子句深度迭代加深进行近似。

The paper is organized as follows. In Section 2, we define Real Logic. In Section 3 , we propose the Learning-as-Inference framework. In Section 4, we instantiate the framework by showing how Real Logic can be implemented in deep Tensor Neural Networks leading to Logic Tensor Networks (LTN). Section 5 contains an example of how LTN handles knowledge completion using (possibly inconsistent) data and knowledge from the well-known smokers and friends experiment. Section 6 concludes the paper and discusses directions for future work.
本文结构如下。第2节定义实逻辑（Real Logic）。第3节提出学习即推理框架。第4节通过展示实逻辑如何在深度张量神经网络中实现，形成逻辑张量网络（Logic Tensor Networks, LTN），实例化该框架。第5节以著名的吸烟者与朋友实验为例，展示LTN如何利用（可能不一致的）数据和知识完成知识补全。第6节总结全文并讨论未来工作方向。

3 2 Real Logic

4 2 实逻辑

We start from a first order language $\mathcal{L}$,whose signature contains a set $\mathcal{C}$ of constant symbols,a set $\mathcal{F}$ of functional symbols,and a set $\mathcal{P}$ of predicate symbols. The sentences of $\mathcal{L}$ are used to express relational knowledge,e.g. the atomic formula $R\left(o_1, o_2\right)$ states that objects o_1 and o_2 are related to each other through binary relation $R; \forall xy. \left(R\left(x, y\right) \rightarrow R\left(y, x\right)\right)$ states that R is a symmetric relation,where x and y are variables; $\exists y. R\left(o_1, y\right)$ states that there is an (unknown) object which is related to object o_1 through R . For simplicity,without loss of generality,we assume that all logical sentences of $\mathcal{L}$ are in prenex conjunctive,skolemised normal form [16],e.g. a sentence $\forall x\left(A\left(x\right) \rightarrow \exists yR\left(x, y\right)\right)$ is transformed into an equivalent clause $\neg A\left(x\right) \vee R\left(x, f\left(x\right)\right)$,where f is a new function symbol.
我们从一阶语言$\mathcal{L}$开始，其符号包含一组常量符号$\mathcal{C}$，一组函数符号$\mathcal{F}$，以及一组谓词符号$\mathcal{P}$。$\mathcal{L}$的句子用于表达关系知识，例如原子公式$R\left(o_1, o_2\right)$表明对象o_1和o_2通过二元关系$R; \forall xy. \left(R\left(x, y\right) \rightarrow\right.$相关联；$R\left(y, x\right))$表明$R$是对称关系，其中$x$和$y$是变量；$\exists y. R\left(o_1, y\right)$表明存在一个（未知的）对象通过$R$与对象$o_1$相关联。为简化起见，且不失一般性，我们假设所有$\mathcal{L}$的逻辑句子均为前束范式合取范式，且已斯科勒姆化[16]，例如句子$\forall x\left(A\left(x\right) \rightarrow \exists yR\left(x, y\right)\right)$被转换为等价子句$\neg A\left(x\right) \vee R\left(x, f\left(x\right)\right)$，其中$f$是一个新的函数符号。

As for the semantics of $\mathcal{L}$,we deviate from the standard abstract semantics of FOL, and we propose a concrete semantics with sentences interpreted as tuples of real numbers. To emphasise the fact that $\mathcal{L}$ is interpreted in a "real" world,we use the term (semantic) grounding,denoted by $\mathcal{G}$,instead of the more standard interpretation ^{2} .
关于$\mathcal{L}$的语义，我们偏离了标准一阶逻辑（FOL）抽象语义，提出了一种具体语义，将句子解释为实数元组。为强调$\mathcal{L}$在“真实”世界中的解释，我们使用术语（语义）基础，记为$\mathcal{G}$，而非更常用的解释^2。

- $\mathcal{G}$ associates an n -tuple of real numbers $\mathcal{G}\left(t\right)$ to any closed term t of $\mathcal{L}$; intuitively $\mathcal{G}\left(t\right)$ is the set of numeric features of the object denoted by t .
- $\mathcal{G}$将一个n维实数元组$\mathcal{G}\left(t\right)$关联到$\mathcal{L}$的任意封闭项$t$；直观上，$\mathcal{G}\left(t\right)$是由$t$所指对象的数值特征集合。

- $\mathcal{G}$ associates a real number in the interval $[0, 1]$ to each clause ϕ of $\mathcal{L}$. Intuitively, $\mathcal{G}\left(\phi\right)$ represents one's confidence in the truth of ϕ ; the higher the value,the higher the confidence.
- $\mathcal{G}$将区间$[0, 1]$内的实数关联到$\mathcal{L}$的每个子句ϕ。直观上，$\mathcal{G}\left(\phi\right)$表示对$\phi$真实性的置信度；数值越高，置信度越大。

A grounding is specified only for the elements of the signature of $\mathcal{L}$. The grounding of terms and clauses is defined inductively, as follows.
基础仅为$\mathcal{L}$符号的元素指定。项和子句的基础通过归纳定义，如下所示。

Definition 1. A grounding $\mathcal{G}$ for a first order language $\mathcal{L}$ is a function from the signature of $\mathcal{L}$ to the real numbers that satisfies the following conditions:
定义1. 一阶语言$\mathcal{L}$的基础$\mathcal{G}$是从$\mathcal{L}$的符号映射到实数的函数，满足以下条件：

1. $\mathcal{G}\left(c\right) \in \mathbb{R}^n$ for every constant symbol $c \in \mathcal{C}$;
2. 对每个常量符号$c \in \mathcal{C}$，$\mathcal{G}\left(c\right) \in \mathbb{R}^n$；

2. $\mathcal{G}\left(f\right) \in \mathbb{R}^{n \cdot \alpha(f)} \rightarrow \mathbb{R}^n$ for every $f \in \mathcal{F}$;
3. 对每个$f \in \mathcal{F}$，$\mathcal{G}\left(f\right) \in \mathbb{R}^{n \cdot \alpha(f)} \rightarrow \mathbb{R}^n$；

[2] In logic,the term "grounding" indicates the operation of replacing the variables of a term/formula with constants. To avoid confusion, we use the term "instantiation" for this.
[2] 在逻辑中，“基础”一词指用常量替换项/公式中的变量的操作。为避免混淆，我们使用“实例化”一词来表示此操作。

---

3. $\mathcal{G}(P) \in \mathbb{R}^{n \cdot \alpha(R)} \to [0,1]$ for every $P \in \mathcal{P}$;
4. 对每个$P \in \mathcal{P}$，$\mathcal{G}(P) \in \mathbb{R}^{n \cdot \alpha(R)} \to [0,1]$；

A grounding $\mathcal{G}$ is inductively extended to all the closed terms and clauses,as follows:
一个赋值$\mathcal{G}$通过归纳方式扩展到所有封闭项和子句，具体如下：

$$\mathcal{G}(f(t_1,\ldots,t_m)) = \mathcal{G}(f)(\mathcal{G}(t_1),\ldots,\mathcal{G}(t_m))$$

$$\mathcal{G}(P(t_1,\ldots,t_m)) = \mathcal{G}(P)(\mathcal{G}(t_1),\ldots,\mathcal{G}(t_m))$$

$$\mathcal{G}(\neg P(t_1,\ldots,t_m)) = 1 - \mathcal{G}(P(t_1,\ldots,t_m))$$

$$\mathcal{G}(\phi_1 \vee \cdots \vee \phi_k) = \mu(\mathcal{G}(\phi_1),\ldots,\mathcal{G}(\phi_k))$$

where μ is an s-norm operator,also known as a t-co-norm operator (i.e. the dual of some t-norm operator). [3]
其中μ是一个s-范数算子，也称为t-余范数算子（即某个t-范数算子的对偶）。[3]

Example 1. Suppose that $O = \{o_1, o_2, o_3\}$ is a set of documents defined on a finite dictionary $D = \{w_1, \ldots, w_n\}$ of n words. Let $\mathcal{L}$ be the language that contains the binary function symbol $\mathsf{concat}(x,y)$ denoting the document resulting from the concatenation of documents x with y . Let $\mathcal{L}$ contain also the binary predicate Sim which is supposed to be true if document x is deemed to be similar to document y . An example of grounding is the one that associates to each document its bag-of-words vector [7]. As a consequence, a natural grounding of the concat function would be the sum of the vectors, and of the Sim predicate, the cosine similarity between the vectors. More formally:
示例1. 假设$O = \{o_1, o_2, o_3\}$是一组定义在有限词典$D = \{w_1, \ldots, w_n\}$上的文档集合，词典包含n个词。设$\mathcal{L}$为包含二元函数符号$\mathsf{concat}(x,y)$的语言，该符号表示由文档x与y连接而成的新文档。$\mathcal{L}$还包含二元谓词Sim，当文档x被认为与文档y相似时，该谓词为真。一个赋值的例子是将每个文档关联到其词袋向量[7]。因此，concat函数的自然赋值是向量的加和，Sim谓词的自然赋值是向量间的余弦相似度。更正式地：

$-\mathcal{G}(o_i) = \langle n_{w_1}^{o_i}, \ldots, n_{w_n}^{o_i} \rangle$,where n_w^d is the number of occurrences of word w in document d ;
$-\mathcal{G}(o_i) = \langle n_{w_1}^{o_i}, \ldots, n_{w_n}^{o_i} \rangle$，其中$n_w^d$是词$w$在文档$d$中的出现次数；

- if $\mathbf{v}, \mathbf{u} \in \mathbb{R}^n, \mathcal{G}(\text{concat})(\mathbf{u}, \mathbf{v}) = \mathbf{u} + \mathbf{v}$;
- 如果$\mathbf{v}, \mathbf{u} \in \mathbb{R}^n, \mathcal{G}(\text{concat})(\mathbf{u}, \mathbf{v}) = \mathbf{u} + \mathbf{v}$；

- if $\mathbf{v}, \mathbf{u} \in \mathbb{R}^n, \mathcal{G}(\text{Sim})(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$.
- 如果$\mathbf{v}, \mathbf{u} \in \mathbb{R}^n, \mathcal{G}(\text{Sim})(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$。

For instance,if the three documents are $o_1 =$ "John studies logic and plays football", $o_2 =$ "Mary plays football and logic games", $o_3 =$ " John and Mary play football and study logic together",and $W = \{$ John,Mary,and,football,game,logic,play,study,together $\}$ then the following are examples of the grounding of terms, atomic formulas and clauses.
例如，若三个文档分别是$o_1 =$“John学习逻辑并踢足球”，$o_2 =$“Mary踢足球和玩逻辑游戏”，$o_3 =$“John和Mary一起踢足球并学习逻辑”，且$W = \{$ John,Mary,and,football,game,logic,play,study,together $\}$，则以下是项、原子公式和子句赋值的示例。

$$\mathcal{G}(o_1) = \langle 1,0,1,1,0,1,1,1,0\rangle$$

$$\mathcal{G}(o_2) = \langle 0,1,1,1,1,1,1,0,0\rangle$$

$$\mathcal{G}(o_3) = \langle 1,1,2,1,0,1,1,1,1\rangle$$

$$\mathcal{G}(\text{concat}(o_1,o_2)) = \mathcal{G}(o_1) + \mathcal{G}(o_2) = \langle 1,1,2,2,1,2,2,1,0\rangle$$

$$\mathcal{G}\bigg(\text{Sim}(\text{concat}(o_1,o_2),o_3) = \frac{\mathcal{G}(\text{concat}(o_1,o_2)) \cdot \mathcal{G}(o_3)}{\|\mathcal{G}(\text{concat}(o_1,o_2))\| \cdot \|\mathcal{G}(o_3)\| \cdot} \approx \frac{13}{14.83} \approx 0.88$$

$$\mathcal{G}(\text{Sim}(o_1,o_3) \vee \text{Sim}(o_2,o_3)) = \mu_{\max}(\mathcal{G}(\text{Sim}(o_1,o_3),\mathcal{G}(\text{Sim}(o_2,o_3))$$

$$\approx \max(0.86, 0.73) = 0.86$$

[3] Examples of t-norms which can be chosen here are Lukasiewicz,product,and Gödel. Lukasiewicz s-norm is defined as $\mu_{Luk}(x,y) = \min(x+y,1)$; Product s-norm is defined as $\mu_{Pr}(x,y) = x+y-x\cdot y$; Gödel s-norm is defined as $\mu_{\max}(x,y) = \max(x,y)$.
[3] 此处可选的t-范数示例包括Lukasiewicz、乘积和Gödel。Lukasiewicz s-范数定义为$\mu_{Luk}(x,y) = \min(x+y,1)$；乘积s-范数定义为$\mu_{Pr}(x,y) = x+y-x\cdot y$；Gödel s-范数定义为$\mu_{\max}(x,y) = \max(x,y)$。

---

5 3 Learning as approximate satisfiability

6 3 学习作为近似可满足性

We start by defining ground theory and their satisfiability.
我们首先定义赋值理论及其可满足性。

Definition 2 (Satisfiability). Let ϕ be a closed clause in $\mathcal{L}, \mathcal{G}$ a grounding,and $v \leq w \in [0,1]$. We say that $\mathcal{G}$ satisfies ϕ in the confidence interval $[v,w]$,written $\mathcal{G} \models_v^w \phi$, if $v \leq \mathcal{G}(\phi) \leq w$.
定义2（可满足性）。设ϕ为赋值$\mathcal{L}, \mathcal{G}$中的封闭子句，且$v \leq w \in [0,1]$。若$\mathcal{G}$在置信区间$[v,w]$内满足ϕ，记作$\mathcal{G} \models_v^w \phi$，当且仅当$v \leq \mathcal{G}(\phi) \leq w$。

A partial grounding,denoted by $\widehat{\mathcal{G}}$,is a grounding that is defined on a subset of the signature of $\mathcal{L}$. A grounded theory is a set of clauses in the language of $\mathcal{L}$ and partial grounding $\widehat{\mathcal{G}}$.
部分赋值，记作$\widehat{\mathcal{G}}$，是定义在$\mathcal{L}$符号子集上的赋值。一个赋值理论是语言$\mathcal{L}$和部分赋值$\widehat{\mathcal{G}}$中的子句集合。

Definition 3 (Grounded Theory). A grounded theory (GT) is a pair $\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$ where $\mathcal{K}$ is a set of pairs $\langle [v,w], \phi(\mathbf{x}) \rangle$,where $\phi(\mathbf{x})$ is a clause of $\mathcal{L}$ containing the set $\mathbf{x}$ of free variables,and $[v,w] \subseteq [0,1]$ is an interval contained in $[0,1]$,and $\mathcal{G}$ is a partial grounding.
定义3（赋值理论）。赋值理论（GT）是一个对$\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$，其中$\mathcal{K}$是一组对$\langle [v,w], \phi(\mathbf{x}) \rangle$，$\phi(\mathbf{x})$是包含自由变量集合$\mathbf{x}$的$\mathcal{L}$的子句，$[v,w] \subseteq [0,1]$是包含于$[0,1]$的区间，$\mathcal{G}$是部分赋值。

Definition 4 (Satisfiability of a Grounded Theory). A GT $\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$ is satisfiabile if there exists a grounding $\mathcal{G}$,which extends $\widehat{\mathcal{G}}$ such that for all $\langle [v,w], \phi(\mathbf{x}) \rangle \in \mathcal{K}$ and any tuple $\mathbf{t}$ of closed terms, $\mathcal{G} \models_v^w \phi(\mathbf{t})$.
定义4（赋值理论的可满足性）。若存在扩展$\widehat{\mathcal{G}}$的赋值$\mathcal{G}$，使得对所有$\langle [v,w], \phi(\mathbf{x}) \rangle \in \mathcal{K}$及任意闭合项元组$\mathbf{t}$，$\mathcal{G} \models_v^w \phi(\mathbf{t})$，则GT$\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$是可满足的。

From the previous definiiton it follows that checking if a GT $\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$ is satisfiable amounts to seaching for an extension of the partial grounding $\widehat{\mathcal{G}}$ in the space of all possible groundings,such that all the instantiations of the clauses in $\mathcal{K}$ are satisfied w.r.t. the specified interval. Clearly this is unfeasible from a practical point of view. As is usual, we must restrict both the space of grounding and clause instantiations. Let us consider each in turn: To check satisfiability on a subset of all the functions on real numbers, recall that a grounding should capture a latent correlation between the quantitative attributes of an object and its relational properties [4] . In particular,we are interested in searching within a specific class of functions, in this paper based on tensor networks, although other family of functions can be considered. To limit the number of clause instantiations,which in general might be infinite since $\mathcal{L}$ admits function symbols, the usual approach is to consider the instantiations of each clause up to a certain depth [3].
由前述定义可知，检查GT$\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$的可满足性即在所有可能赋值空间中搜索部分赋值$\widehat{\mathcal{G}}$的扩展，使得$\mathcal{K}$中所有子句的实例化均满足指定区间。显然，从实际角度看这是不可行的。通常，我们必须限制赋值空间和子句实例化。分别考虑：为检查实数函数子集上的可满足性，回想赋值应捕捉对象定量属性与其关系属性[4]间的潜在相关性。特别地，本文关注基于张量网络（tensor networks）的特定函数类，尽管也可考虑其他函数族。为限制子句实例化数量（通常可能无限，因为$\mathcal{L}$允许函数符号），常用方法是考虑每个子句至一定深度的实例化[3]。

When a grounded theory $\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$ is inconsitent,that is,there is no grounding $\mathcal{G}$ that satisfies it, we are interested in finding a grounding which satisfies as much as possible of $\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$. For any $\langle [v,w], \phi \rangle \in \mathcal{K}$ we want to find a grounding $\mathcal{G}$ that minimizes the satisfiability error. An error occurs when a grounding $\mathcal{G}$ assigns a value $\mathcal{G}(\phi)$ to a clause ϕ which is outside the interval $[v,w]$ prescribed by $\mathcal{K}$. The measure of this error can be defined as the minimal distance between the points in the interval $[v,w]$ and $\mathcal{G}(\phi)$:
当赋值理论$\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$不一致，即不存在满足它的赋值$\mathcal{G}$时，我们关注寻找尽可能满足$\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$的赋值。对于任意$\langle [v,w], \phi \rangle \in \mathcal{K}$，我们希望找到最小化可满足性误差的赋值$\mathcal{G}$。当赋值$\mathcal{G}$对子句$\phi$赋值$\mathcal{G}(\phi)$超出$\mathcal{K}$规定的区间$[v,w]$时，发生误差。该误差的度量可定义为区间$[v,w]$内点与$\mathcal{G}(\phi)$的最小距离：

$$\operatorname{Loss}(\mathcal{G}, \langle [v, w], \phi \rangle) = |x - \mathcal{G}(\phi)|, v \leq x \leq w \tag{1}$$

Notice that if $\mathcal{G}(\phi) \in [v, w], \operatorname{Loss}(\mathcal{G}, \phi) = 0$.
注意如果$\mathcal{G}(\phi) \in [v, w], \operatorname{Loss}(\mathcal{G}, \phi) = 0$。

---

[4] For example,whether a document is classified as from the field of Artificial Intelligence (AI) depends on its bag-of-words grounding. If the language $\mathcal{L}$ contains the unary predicate $AI(x)$ standing for " x is a paper about AI" then the grounding of $AI(x)$,which is a function from bag-of-words vectors to $[0, 1]$,should assign values close to 1 to the vectors which are close semantically to AI . Furthermore,if two vectors are similar (e.g. according to the cosine similarity measure) then their grounding should be similar.
[4] 例如，文档是否被归类为人工智能（Artificial Intelligence, AI）领域，取决于其词袋（bag-of-words）基础。如果语言$\mathcal{L}$包含一元谓词$AI(x)$，表示“x是一篇关于AI的论文”，那么$AI(x)$的基础映射（从词袋向量到$[0, 1]$的函数）应当对语义上接近AI的向量赋予接近1的值。此外，如果两个向量相似（例如根据余弦相似度度量），那么它们的基础映射也应相似。

The above gives rise to the following definition of approximate satisfiability w.r.t. a family $\mathbb{G}$ of grounding functions on the language $\mathcal{L}$.
上述内容引出了关于语言$\mathcal{L}$上基础函数族$\mathbb{G}$的近似可满足性的以下定义。

Definition 5 (Approximate satisfiability). Let $\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$ be a grounded theory and $\mathcal{K}_0$ a finite subset of the instantiations of the clauses in $\mathcal{K}$,i.e.
定义5（近似可满足性）。设$\langle \mathcal{K}, \widehat{\mathcal{G}} \rangle$为一个有基础的理论，$\mathcal{K}_0$为$\mathcal{K}$中子句实例的有限子集，即

$$K_0 \subseteq \{\langle [v, w], \phi(\mathbf{t}) \rangle\} \mid \langle [v, w], \phi(\mathbf{x}) \rangle \in \mathcal{K} \text{and} \mathbf{t} \text{is any} n \text{-tuple of closed terms.}\}$$

Let $\mathbb{G}$ be a family of grounding functions. We define the best satisfiability problem as the problem of finding an extensions $\mathcal{G}^*$ of $\widehat{\mathcal{G}}$ in $\mathbb{G}$ that minimizes the satisfiability error on the set $\mathcal{K}_0$,that is:
设$\mathbb{G}$为一族基础函数。我们将最佳可满足性问题定义为在$\mathbb{G}$中寻找$\widehat{\mathcal{G}}$的扩展$\mathcal{G}^*$，使得在集合$\mathcal{K}_0$上的可满足性误差最小化，即：

$$\mathcal{G}^* = \underset{\widehat{\mathcal{G}} \subseteq \mathcal{G} \in \mathbb{G}}{\operatorname{argmin}} \sum_{\langle [v, w], \phi(\mathbf{t}) \rangle \in \mathcal{K}_0} \operatorname{Loss}(\mathcal{G}, \langle [v, w], \phi(\mathbf{t}) \rangle)$$

7 4 Implementing Real Logic in Tensor Networks

8 4 在张量网络中实现实数逻辑

Specific instances of Real Logic can be obtained by selectiong the space $\mathbb{G}$ of ground-ings and the specific s-norm for the interpretation of disjunction. In this section, we describe a realization of real logic where $\mathbb{G}$ is the space of real tensor transformations of order k (where k is a parameter). In this space,function symbols are interpreted as linear transformations. More precisely,if f is a function symbol of arity m and $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^n$ are real vectors corresponding to the grounding of m terms then $\mathcal{G}(f)(\mathbf{v}_1, \ldots, \mathbf{v}_m)$ can be written as:

实数逻辑的具体实例可以通过选择基础空间$\mathbb{G}$和用于析取解释的特定s-范数获得。本节中，我们描述一种实数逻辑的实现，其中$\mathbb{G}$是实数张量变换空间，阶数为k（k为参数）。在该空间中，函数符号被解释为线性变换。更具体地，如果f是一个元数为m的函数符号，且$\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^n$是对应于$m$个项的基础的实向量，则$\mathcal{G}(f)(\mathbf{v}_1, \ldots, \mathbf{v}_m)$可表示为：

$$\mathcal{G}(f)(\mathbf{v}_1, \ldots, \mathbf{v}_m) = M_f \mathbf{v} + N_f$$

for some $n \times mn$ matrix M_f and n -vector N_f ,where $\mathbf{v} = \langle \mathbf{v}_1, \ldots, \mathbf{v}_n \rangle$.
对于某个$n \times mn$矩阵M_f和n维向量N_f，其中$\mathbf{v} = \langle \mathbf{v}_1, \ldots, \mathbf{v}_n \rangle$。

The grounding of m -ary predicate $P, \mathcal{G}(P)$,is defined as a generalization of the neural tensor network [26] (which has been shown effective at knowledge compilation in the presence of simple logical constraints),as a function from $\mathbb{R}^{mn}$ to $[0,1]$,as follows:
元数为m的谓词$P, \mathcal{G}(P)$的基础定义为神经张量网络[26]的推广（该网络已被证明在存在简单逻辑约束时对知识编译有效），作为从$\mathbb{R}^{mn}$到$[0,1]$的函数，定义如下：

$$\mathcal{G}(P) = \sigma\left(u_P^T \tanh\left(\mathbf{v}^T W_P^{[1:k]} \mathbf{v} + V_P \mathbf{v} + B_P\right)\right) \tag{2}$$

where $W_P^{[1:k]}$ is a 3-D tensor in $\mathbb{R}^{mn \times mn \times k}, V_P$ is a matrix in $\mathbb{R}^{k \times mn}$,and B_P is a vector in $\mathbb{R}^k$,and σ is the sigmoid function. With this encoding,the grounding (i.e. truth-value) of a clause can be determined by a neural network which first computes the grounding of the literals contained in the clause, and then combines them using the specific s-norm. An example of tensor network for $\neg P(x,y) \rightarrow A(y)$ is shown in Figure 1. This architecture is a generalization of the structure proposed in [26], that has been shown rather effective for the task of knowledge compilation, also in presence of simple logical constraints. In the above tensor network formulation, W_*, V_*, B_* and u_* with $* \in \{P, A\}$ are parameters to be learned by minimizing the loss function or, equivalently,to maximize the satisfiability of the clause $P(x,y) \rightarrow A(y)$.
其中 $W_P^{[1:k]}$ 是一个三维张量，$\mathbb{R}^{mn \times mn \times k}, V_P$ 是一个矩阵，$\mathbb{R}^{k \times mn}$ 是一个向量，B_P 是 sigmoid 函数。通过这种编码，子句的赋值（即真值）可以由神经网络确定，该网络首先计算子句中包含的文字的赋值，然后使用特定的 s-范数将它们组合起来。$\neg P(x,y) \rightarrow A(y)$ 的张量网络示例如图1所示。该架构是文献[26]中提出结构的推广，已被证明在知识编译任务中相当有效，即使在存在简单逻辑约束的情况下也是如此。在上述张量网络的表述中，W_*, V_*, B_* 和 u_* 以及 $* \in \{P, A\}$ 是通过最小化损失函数或等价地最大化子句 $P(x,y) \rightarrow A(y)$ 的可满足性来学习的参数。

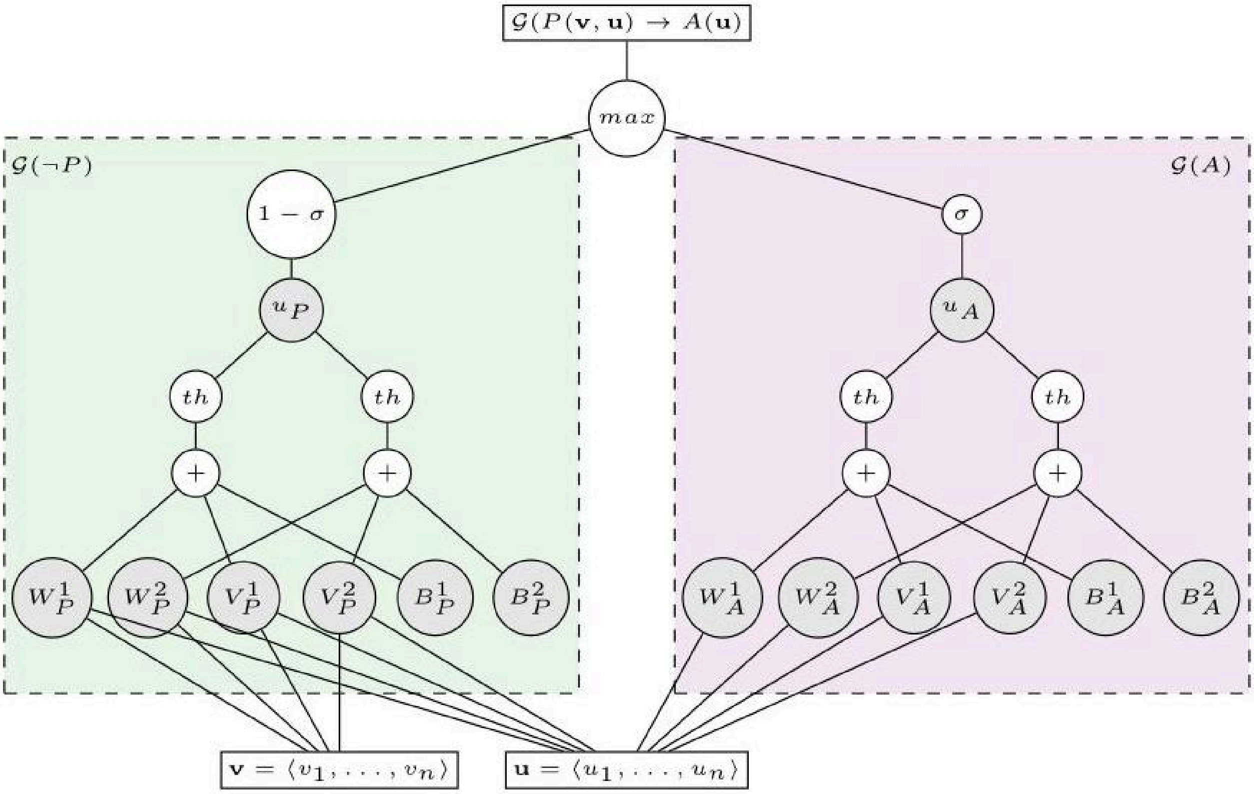


Fig. 1. Tensor net for $P(x,y) \to A(y)$,with $\mathcal{G}(x) = \mathbf{v}$ and $\mathcal{G}(y) = \mathbf{u}$ and $k = 2$.
图1. 用于 $P(x,y) \to A(y)$ 的张量网络，包含 $\mathcal{G}(x) = \mathbf{v}$、$\mathcal{G}(y) = \mathbf{u}$ 和 $k = 2$ 。

9 5 An Example of Knowledge Completion

10 5 知识补全示例

Logic Tensor Networks have been implemented as a Python library called 1tn using Google's TENSORFLOW ™ . To test our idea,in this section we use the well-known friends and smokers [5] example [24] to illustrate the task of knowledge completion in ltn. There are 14 people divided into two groups $\{a, b, \ldots, h\}$ and $\{i, j, \ldots, n\}$. Within each group of people we have complete knowledge of their smoking habits. In the first group, we have complete knowledge of who has and does not have cancer. In the second group, this is not known for any of the persons. Knowledge about the friendship relation is complete within each group only if symmetry of friendship is assumed. Otherwise,it is imcomplete in that it may be known that,e.g., a is a friend of b ,but not known whether b is a friend of a . Finally,there is also general knowledge about smoking, friendship and cancer, namely, that smoking causes cancer, friendship is normally a symmetric and anti-reflexive relation, everyone has a friend, and that smoking propagates (either actively or passively) among friends. All this knowledge can be represented by the knowledge-bases shown in Figure 2.

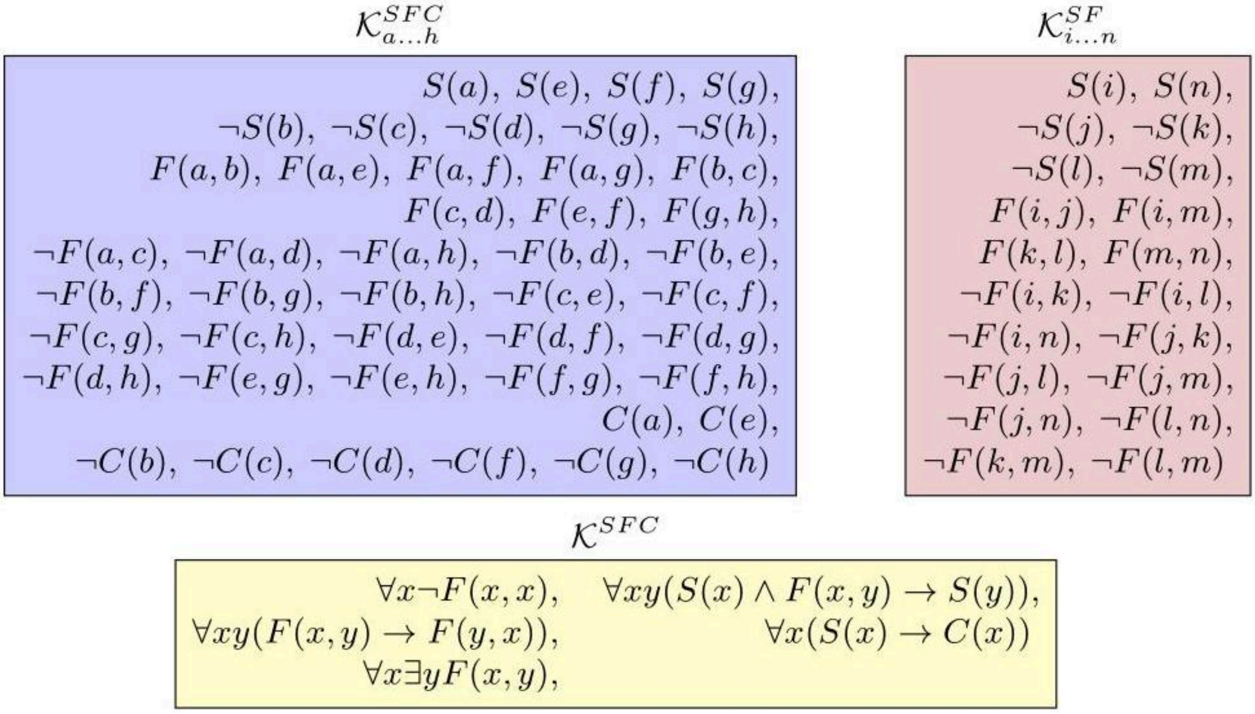
逻辑张量网络（Logic Tensor Networks）已作为一个名为 1tn 的 Python 库实现，基于谷歌的 TENSORFLOW ™ 。为了验证我们的想法，本节使用著名的“朋友与吸烟者”[5] 示例[24]来说明 ltn 中的知识补全任务。共有14人，分为两组 $\{a, b, \ldots, h\}$ 和 $\{i, j, \ldots, n\}$。在每组内，我们对他们的吸烟习惯有完整的了解。在第一组中，我们对谁患癌症和谁未患癌症有完整的知识。第二组中，这些信息对任何人都未知。关于友谊关系的知识仅在假设友谊对称的情况下在每组内是完整的。否则，它是不完整的，例如，可能知道 a 是 b 的朋友，但不知道 b 是否是 a 的朋友。最后，还有关于吸烟、友谊和癌症的一般知识，即吸烟导致癌症，友谊通常是对称且反自反的关系，每个人都有朋友，且吸烟在朋友间传播（无论是主动还是被动）。所有这些知识都可以通过图2所示的知识库表示。

The facts contained in the knowledge-bases should have different degrees of truth, and this is not known. Otherwise, the combined knowledge-base would be inconsistent (it would deduce e.g. $S(b)$ and $\neg S(b)$). Our main task is to complete the knowledge-base (KB), that is: (i) find the degree of truth of the facts contained in KB, (ii) find a truth-value for all the missing facts,e.g. $C(i)$,(iii) find the grounding of each constant symbol $a, \ldots, n.$[6] To answer (i)-(iii),we use 1tn to find a grounding that best approximates the complete KB. We start by assuming that all the facts contained in the knowledge-base are true (i.e. have degree of truth 1). To show the role of background knolwedge in the learning-inference process, we run two experiments. In the first (exp1),we seek to complete a KB consisting of only factual knowledge: $\mathcal{K}_{exp1} = \mathcal{K}^{SFC}_{a \ldots h} \cup \mathcal{K}^{SF}_{i \ldots n}$. In the second (exp1),we also include background knowledge,that is: $\mathcal{K}_{exp2} = \mathcal{K}_{exp1} \cup \mathcal{K}^{SFC}$.
知识库中包含的事实应具有不同的真实性程度，但这一点尚不明确。否则，合并后的知识库将会不一致（例如会推导出$S(b)$和$\neg S(b)$）。我们的主要任务是完善知识库（KB），即：（i）确定KB中包含事实的真实性程度，（ii）为所有缺失的事实找到一个真值，例如$C(i)$，（iii）找到每个常量符号$a, \ldots, n.$[6]的赋值。为了解决（i）-（iii），我们使用1tn来寻找一个最能近似完整KB的赋值。我们首先假设知识库中所有事实都为真（即真实性程度为1）。为了展示背景知识在学习推理过程中的作用，我们进行了两次实验。第一次（exp1）中，我们试图完善仅包含事实知识的KB：$\mathcal{K}_{exp1} = \mathcal{K}^{SFC}_{a \ldots h} \cup \mathcal{K}^{SF}_{i \ldots n}$。第二次（exp1）中，我们还包括了背景知识，即：$\mathcal{K}_{exp2} = \mathcal{K}_{exp1} \cup \mathcal{K}^{SFC}$。

---

[5] Normally,a probabilistic approach is taken to solve this problem,and one that requires instantiating all clauses to remove variables, essentially turning the problem into a propositional one; 1th takes a different approach.
[5] 通常，解决此问题采用概率方法，并且需要实例化所有子句以消除变量，实质上将问题转化为命题问题；而1th采取了不同的方法。

[6] Notice how no grounding is provided about the signature of the knowledge-base.
[6] 注意这里没有提供关于知识库签名的任何赋值信息。



$\mathcal{K}^{SFC}_{a\ldots h}$

$$\begin{array}{r} S(a),\ S(e),\ S(f),\ S(g),\\ \neg S(b),\ \neg S(c),\ \neg S(d),\ \neg S(g),\ \neg S(h),\\ F(a,b),\ F(a,e),\ F(a,f),\ F(a,g),\ F(b,c),\\ F(c,d),\ F(e,f),\ F(g,h),\\ \neg F(a,c),\ \neg F(a,d),\ \neg F(a,h),\ \neg F(b,d),\ \neg F(b,e),\\ \neg F(b,f),\ \neg F(b,g),\ \neg F(b,h),\ \neg F(c,e),\ \neg F(c,f),\\ \neg F(c,g),\ \neg F(c,h),\ \neg F(d,e),\ \neg F(d,f),\ \neg F(d,g),\\ \neg F(d,h),\ \neg F(e,g),\ \neg F(e,h),\ \neg F(f,g),\ \neg F(f,h),\\ C(a),\ C(e),\\ \neg C(b),\ \neg C(c),\ \neg C(d),\ \neg C(f),\ \neg C(g),\ \neg C(h) \end{array}$$

$\mathcal{K}^{SF}_{i\ldots n}$

$$\begin{array}{r} S(i),\ S(n),\\ \neg S(j),\ \neg S(k),\\ \neg S(l),\ \neg S(m),\\ F(i,j),\ F(i,m),\\ F(k,l),\ F(m,n),\\ \neg F(i,k),\ \neg F(i,l),\\ \neg F(i,n),\ \neg F(j,k),\\ \neg F(j,l),\ \neg F(j,m),\\ \neg F(j,n),\ \neg F(l,n),\\ \neg F(k,m),\ \neg F(l,m) \end{array}$$

$\mathcal{K}^{SFC}$

$$\begin{array}{rr} \forall x \neg F(x,x), & \forall xy(S(x) \wedge F(x,y) \to S(y)),\\ \forall xy(F(x,y) \to F(y,x)), & \forall x(S(x) \to C(x))\\ \forall x \exists y F(x,y), & \end{array}$$

Fig. 2. Knowledge-bases for the friends-and-smokers example.
图2. 朋友与吸烟者示例的知识库。

We confgure the network as follows: each constant (i.e. person) can have up to 30 real-valued features. We set the number of layers k in the tensor network to 10,and the regularization parameter $^7\lambda = 1^{-10}$. For the purpose of illustration,we use the Lukasiewicz t-norm with s-norm $\mu(a,b) = \min(1, a+b)$,and use the harmonic mean as aggregation operator. An estimation of the optimal grounding is obtained after 5,000 runs of the RMSProp learning algorithm [27] available in TENSORFLOW $^{\text{TM}}$.
我们将网络配置如下：每个常量（即人）最多可以有30个实值特征。我们将张量网络中的层数k设置为10，正则化参数$^7\lambda = 1^{-10}$。为了说明，我们使用带有s-范数$\mu(a,b) = \min(1, a+b)$的Lukasiewicz t-范数，并使用调和平均作为聚合算子。经过5000次RMSProp学习算法[27]（在TENSORFLOW中提供）运行后，获得了最优赋值的估计$^{\text{TM}}$。

The results of the two experiments are reported in Table 1. For readability, we use boldface for truth-values greater than 0.5 . The truth-values of the facts listed in a knowledge-base are highlighted with the same background color of the knowledge-base in Figure 2. The values with white background are the result of the knowledge completion produced by the LTN learning-inference procedure. To evaluate the quality of the results, one has to check whether (i) the truth-values of the facts listed in a KB are indeed close to 1.0, and (ii) the truth-values associated with knowledge completion correspond to expectation. An initial analysis shows that the LTN associated with $\mathcal{K}_{\text{exp1}}$ produces the same facts as $\mathcal{K}_{exp1}$ itself. In other words,the LTN fits the data. However,the LTN also learns to infer additional positive and negative facts about F and C not derivable from $\mathcal{K}_{exp1}$ by pure logical reasoning; for example: $F(c,b), F(g,b)$ and $\neg F(b,a)$. These facts are derived by exploiting similarities between the groundings of Table 1. the constants generated by the LTN. For instance, $\mathcal{G}(c)$ and $\mathcal{G}(g)$ happen to present a high cosine similarity measure. As a result,facts about the friendship relations of c affect the friendship relations of g and vice-versa,for instance $F(c,b)$ and $F(g,b)$. The level of satisfiability associated with $\mathcal{K}_{exp1} \approx 1$,which indicates that $\mathcal{K}_{exp1}$ is classically satisfiable.
两个实验的结果报告见表1。为了便于阅读，我们对大于0.5的真值使用加粗显示。知识库中列出的事实的真值在图2中以与知识库相同的背景色突出显示。白色背景的数值是由LTN学习推理过程产生的知识补全结果。为了评估结果的质量，需要检查(i)知识库中列出的事实的真值是否确实接近1.0，以及(ii)与知识补全相关的真值是否符合预期。初步

分析表明，关联$\mathcal{K}_{\mathrm{exp1}}$的LTN产生了与$\mathcal{K}_{exp1}$本身相同的事实。换句话说，LTN拟合了数据。然而，LTN还学会推断关于F和C的额外正负事实，这些事实无法通过纯逻辑推理从$\mathcal{K}_{exp1}$推导出；例如：$F(c,b), F(g,b)$和$\neg F(b,a)$。这些事实是通过利用表1中LTN生成的常量的基底之间的相似性得出的。例如，$\mathcal{G}(c)$和$\mathcal{G}(g)$恰好表现出较高的余弦相似度。因此，关于c的友谊关系的事实会影响g的友谊关系，反之亦然，例如$F(c,b)$和$F(g,b)$。与$\mathcal{K}_{exp1} \approx 1$相关的可满足性水平表明$\mathcal{K}_{exp1}$在经典意义上是可满足的。

[7] A smoothing factor $\lambda \parallel \boldsymbol{\Omega}\|_2^2$ is added to the loss function to create a preference for learned parameters with a lower absolute value.
在损失函数中加入了一个平滑因子$\lambda \parallel \boldsymbol{\Omega}\|_2^2$，以偏好绝对值较小的学习参数。

	S	C	F							
			a	b	C	d	e	f	g	h
a	1.00	1.00	0.00	1.00	0.00	0.00	1.00	1.00	1.00	0.00
b	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
C	0.00	0.00	0.00	0.82	0.00	1.00	0.00	0.00	0.00	0.00
d	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00
e	1.00	1.00	0.00	0.33	0.21	0.00	0.00	1.00	0.00	0.00
f	1.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00
g	1.00	0.00	0.03	1.00	1.00	1.00	0.11	1.00	0.00	1.00
h	0.00	0.00	0.00	0.23	0.01	0.14	0.00	0.02	0.00	0.00

T1

	S	C	F							
			a	b	C	d	e	f	g	h
a	1.00	1.00	0.00	1.00	0.00	0.00	1.00	1.00	1.00	0.00
b	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
C	0.00	0.00	0.00	0.82	0.00	1.00	0.00	0.00	0.00	0.00
d	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00
e	1.00	1.00	0.00	0.33	0.21	0.00	0.00	1.00	0.00	0.00
f	1.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00
g	1.00	0.00	0.03	1.00	1.00	1.00	0.11	1.00	0.00	1.00
h	0.00	0.00	0.00	0.23	0.01	0.14	0.00	0.02	0.00	0.00

T1

	S	C	F					
			i	j	k	l	m	n
i	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
j	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
k	0.00	0.00	0.10	1.00	0.00	1.00	0.00	0.00
l	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
m	0.00	0.03	1.00	1.00	0.12	1.00	0.00	1.00
n	1.00	0.01	0.00	0.98	0.00	0.01	0.02	0.00

	S	C	F					
			i	j	k	l	m	n
i	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
j	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
k	0.00	0.00	0.10	1.00	0.00	1.00	0.00	0.00
l	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
m	0.00	0.03	1.00	1.00	0.12	1.00	0.00	1.00
n	1.00	0.01	0.00	0.98	0.00	0.01	0.02	0.00

Learning and reasoning on $\mathcal{K}_{exp1} = \mathcal{K}^{SFC}_{a...h} \cup \mathcal{K}^{SF}_{i...n}$

在$\mathcal{K}_{exp1} = \mathcal{K}^{SFC}_{a...h} \cup \mathcal{K}^{SF}_{i...n}$上的学习与推理

	S	C	F							
			a	b	C	d	e	f	g	h
a	0.84	0.87	0.02	0.95	0.01	0.03	0.93	0.97	0.98	0.01
b	0.13	0.16	0.45	0.01	0.97	0.04	0.02	0.03	0.06	0.03
C	0.13	0.15	0.02	0.94	0.11	0.99	0.03	0.16	0.15	0.15
d	0.14	0.15	0.01	0.06	0.88	0.08	0.01	0.03	0.07	0.02
e	0.84	0.85	0.32	0.06	0.05	0.03	0.04	0.97	0.07	0.06
f	0.81	0.19	0.34	0.11	0.08	0.04	0.42	0.08	0.06	0.05
g	0.82	0.19	0.81	0.26	0.19	0.30	0.06	0.28	0.00	0.94
h	0.14	0.17	0.05	0.25	0.26	0.16	0.20	0.14	0.72	0.01

	S	C	F							
			a	b	C	d	e	f	g	h
a	0.84	0.87	0.02	0.95	0.01	0.03	0.93	0.97	0.98	0.01
b	0.13	0.16	0.45	0.01	0.97	0.04	0.02	0.03	0.06	0.03
C	0.13	0.15	0.02	0.94	0.11	0.99	0.03	0.16	0.15	0.15
d	0.14	0.15	0.01	0.06	0.88	0.08	0.01	0.03	0.07	0.02
e	0.84	0.85	0.32	0.06	0.05	0.03	0.04	0.97	0.07	0.06
f	0.81	0.19	0.34	0.11	0.08	0.04	0.42	0.08	0.06	0.05
g	0.82	0.19	0.81	0.26	0.19	0.30	0.06	0.28	0.00	0.94
h	0.14	0.17	0.05	0.25	0.26	0.16	0.20	0.14	0.72	0.01

	S	C	F					
			i	j	k	l	m	n
i	0.83	0.86	0.02	0.91	0.01	0.03	0.97	0.01
j	0.19	0.22	0.73	0.03	0.00	0.04	0.02	0.05
k	0.14	0.34	0.17	0.07	0.04	0.97	0.04	0.02
l	0.16	0.19	0.11	0.12	0.15	0.06	0.05	0.03
m	0.14	0.17	0.96	0.07	0.02	0.11	0.00	0.92
n	0.84	0.86	0.13	0.28	0.01	0.24	0.69	0.02

	S	C	F					
			i	j	k	l	m	n
i	0.83	0.86	0.02	0.91	0.01	0.03	0.97	0.01
j	0.19	0.22	0.73	0.03	0.00	0.04	0.02	0.05
k	0.14	0.34	0.17	0.07	0.04	0.97	0.04	0.02
l	0.16	0.19	0.11	0.12	0.15	0.06	0.05	0.03
m	0.14	0.17	0.96	0.07	0.02	0.11	0.00	0.92
n	0.84	0.86	0.13	0.28	0.01	0.24	0.69	0.02

	$a,\ldots ,h,i,\ldots ,n$
$\forall x\neg F\left({x,x}\right)$	0.98
$\forall {xy}\left({F\left({x,y}\right) \rightarrow F\left({y,x}\right) }\right)$	0.900.90
$\forall x\left({S\left(x\right) \rightarrow C\left(x\right) }\right)$	0.77
$\forall x\left({S\left(x\right) \land F\left({x,y}\right) \rightarrow S\left(y\right) }\right)$	0.96 0.92
$\forall x\exists y\left({F\left({x,y}\right) }\right)$	1.0

	$a,\ldots ,h,i,\ldots ,n$
$\forall x\neg F\left({x,x}\right)$	0.98
$\forall {xy}\left({F\left({x,y}\right) \rightarrow F\left({y,x}\right) }\right)$	0.900.90
$\forall x\left({S\left(x\right) \rightarrow C\left(x\right) }\right)$	0.77
$\forall x\left({S\left(x\right) \land F\left({x,y}\right) \rightarrow S\left(y\right) }\right)$	0.96 0.92
$\forall x\exists y\left({F\left({x,y}\right) }\right)$	1.0

Learning and reasoning on $\mathcal{K}_{exp2} = \mathcal{K}_{a\ldots h}^{SFC} \cup \mathcal{K}_{i\ldots n}^{SF} \cup \mathcal{K}^{SFC}$

在$\mathcal{K}_{exp2} = \mathcal{K}_{a\ldots h}^{SFC} \cup \mathcal{K}_{i\ldots n}^{SF} \cup \mathcal{K}^{SFC}$上的学习与推理

The results of the second experiment show that more facts can be learned with the inclusion of background knowledge. For example,the LTN now predicts that $C\left(i\right)$ and $C\left(n\right)$ are true. Similarly,from the symmetry of the friendship relation,the LTN concludes that m is a friend of i ,as expected. In fact,all the axioms in the generic background knowledge $\mathcal{K}^{SFC}$ are satisfied with a degree of satisfiability higher than 90% , apart from the smoking causes cancer axiom - which is responsible for the classical inconsistency since in the data f and g smoke and do not have cancer -,which has a degree of satisfiability of 77% .

第二次实验的结果表明，加入背景知识后可以学习到更多事实。例如，LTN现在预测$C\left(i\right)$和$C\left(n\right)$为真。同样地，根据友谊关系的对称性，LTN推断出m是i的朋友，符合预期。事实上，通用背景知识$\mathcal{K}^{SFC}$中的所有公理的满足度均高于90%，除了“吸烟导致癌症”这一公理——该公理引发了经典不一致性，因为数据中f和g吸烟但未患癌症——其满足度为77%。

11 6 Related work

12 6 相关工作

In his recent note, [15], Guha advocates the need for a new model theory for distributed representations (such as those based on embeddings). The note sketches a proposal, where terms and (binary) predicates are all interpreted as points/vectors in an n -dimensional real space. The computation of the truth-value of the atomic formulae $P(t_1,\ldots,t_n)$ is obtained by comparing the projections of the vector associated to each t_i with that associated to P_i . Real logic shares with [15] the idea that terms must be interpreted in a geometric space. It has, however, a different (and more general) interpretation of functions and predicate symbols. Real logic is more general because the semantics proposed in [15] can be implemented within an $1tn$ with a single layer $(k=1)$,since the operation of projection and comparison necessary to compute the truth-value of $P(t_1,\ldots,t_m)$ can be encoded within an $nm\times nm$ matrix W with the constraint that $\langle \mathcal{G}(t_1),\ldots,\mathcal{G}(t_n)\rangle^T W \langle \mathcal{G}(t_1),\ldots,\mathcal{G}(t_n)\rangle \leq \delta$,which can be encoded easily in 1tn.
在他最近的论文[15]中，Guha主张需要为分布式表示（如基于嵌入的表示）建立新的模型理论。该论文提出了一个方案，其中术语和（二元）谓词均被解释为n维实数空间中的点/向量。原子公式$P(t_1,\ldots,t_n)$的真值通过比较与每个t_i相关联的向量投影与与P_i相关联的向量投影来计算。实逻辑（Real Logic）与[15]共享术语必须在几何空间中解释的观点，但对函数和谓词符号的解释不同且更为通用。实逻辑更为通用，因为[15]中提出的语义可以在单层$(k=1)$的$1tn$中实现，因为计算$P(t_1,\ldots,t_m)$真值所需的投影和比较操作可以编码在带有约束$\langle \mathcal{G}(t_1),\ldots,\mathcal{G}(t_n)\rangle^T W \langle \mathcal{G}(t_1),\ldots,\mathcal{G}(t_n)\rangle \leq \delta$的$nm\times nm$矩阵$W$中，而该约束可以在1tn中轻松编码。

Real logic is orthogonal to the approach taken by (Hybrid) Markov Logic Networks (MLNs) and its variations [24, 29, 22]. In MLNs, the level of truth of a formula is determined by the number of models that satisfy the formula: the more models, the higher the degree of truth. Hybrid MLNs introduce a dependency from the real features associated to constants, which is given, and not learned. In real logic, instead, the level of truth of a complex formula is determined by (fuzzy) logical reasoning, and the relations between the features of different objects is learned through error minimization. Another difference is that MLNs work under the closed world assumption, while Real Logic is open domain. Much work has been done also on neuro-fuzzy approaches [19]. These are essentially propositional while real logic is first-order.
实逻辑与（混合）马尔可夫逻辑网络（MLNs）及其变体[24, 29, 22]的方法正交。在MLNs中，公式的真值水平由满足该公式的模型数量决定：模型越多，真值程度越高。混合MLNs引入了与常量相关的实特征依赖，该依赖是给定的而非学习得到的。而在实逻辑中，复杂公式的真值水平由（模糊）逻辑推理决定，不同对象特征之间的关系通过误差最小化学习得到。另一个区别是MLNs基于封闭世界假设，而实逻辑是开放域。神经模糊方法[19]也有大量研究，这些方法本质上是命题逻辑，而实逻辑是一级逻辑。

Bayesian logic (BLOG) [20] is open domain, and in this respect similar to real logic and LTNs. But, instead of taking an explicit probabilistic approach, LTNs draw from the efficient approach used by tensor networks for knowledge graphs, as already discussed. LTNs can have a probabilistic interpretation but this is not a requirement. Other statistical AI and probabilistic approaches such as lifted inference fall into this category, including probabilistic variations of inductive logic programming (ILP) [23], which are normally restricted to Horn clauses. Metainterpretive ILP [21], together with BLOG, seem closer to LTNs in what concerns the knowledge representation language, but do not explore the benefits of tensor networks for computational efficiency.
贝叶斯逻辑（BLOG）[20]是开放域的，在这方面与实逻辑和LTNs相似。但LTNs并未采用显式的概率方法，而是借鉴了知识图谱张量网络的高效方法，如前所述。LTNs可以有概率解释，但这不是必需的。其他统计AI和概率方法如提升推理也属于此类，包括通常限制于Horn子句的归纳逻辑编程（ILP）[23]的概率变体。元解释ILP[21]与BLOG在知识表示语言方面似乎更接近LTNs，但未探索张量网络在计算效率上的优势。

An approach for embedding logical knowledge onto data for the purpose of relational learning, similar to Real Logic, is presented in [25]. Real Logic and [25] share the idea of interpreting a logical alphabet in an n -dimensional real space. Terminologically, the term "grounding" in Real Logic corresponds to "embeddings" in [25]. However, there are several differences. First, [25] uses function-free langauges, while we provide also groundings for functional symbols. Second, the model used to compute the truth-values of atomic formulas adopted in [25] is a special case of the more general model proposed in this paper (as described in Eq. (2)). Finally, the semantics of the universal and existential quantifiers adopted in [25] is based on the closed-world assumption (CWA), i.e. universally (respectively, existentially) quantified formulas are reduced to the finite conjunctions (respectively, disjunctions) of all of their possible instantiations; Real Logic does not make the CWA. Furthermore, Real Logic does not assume a specific t-norm.
文献[25]提出了一种将逻辑知识嵌入数据以实现关系学习的方法，类似于Real Logic。Real Logic与[25]共享将逻辑字母表解释为n维实数空间的思想。术语上，Real Logic中的“grounding”对应于[25]中的“embeddings”。然而，两者存在若干差异。首先，[25]使用无函数语言，而我们还提供了函数符号的groundings。其次，[25]中用于计算原子公式真值的模型是本文提出的更一般模型（如公式(2)所述）的特例。最后，[25]采用的全称和存在量词语义基于封闭世界假设（CWA），即全称（存在）量化公式被简化为其所有可能实例的有限合取（析取）；而Real Logic不采用CWA。此外，Real Logic不假设特定的t-范数。

As in [11], LTN is a framework for learning in the presence of logical constraints. LTNs share with [11] the idea that logical constraints and training examples can be treated uniformly as supervisions of a learning algorithm. LTN introduces two novelties: first, in LTN existential quantifiers are not grounded into a finite disjunction, but are scolemized. In other words, CWA is not required, and existentially quantified formulas can be satisfied by "new individuals". Second, LTN allows one to generate data for prediction. For instance,if a grounded theory contains the formula $\forall x \exists y R(x, y)$,LTN generates a real function (corresponding to the grounding of the Skolem function introduced by the formula) which for every vector $\mathbf{v}$ returns the feature vector $f(\mathbf{v})$,which can be intuitively interpreted as being the set of features of a typical object which takes part in relation R with the object having features equal to $\mathbf{v}$.
如文献[11]所述，LTN是一个在逻辑约束存在下进行学习的框架。LTN与[11]共享将逻辑约束和训练样本统一视为学习算法监督信号的思想。LTN引入了两项创新：首先，在LTN中，存在量词不被ground为有限析取，而是进行斯科勒姆化（skolemization）。换言之，不需要CWA，存在量化公式可以通过“新个体”来满足。其次，LTN允许生成用于预测的数据。例如，如果一个grounded理论包含公式$\forall x \exists y R(x, y)$，LTN会生成一个实值函数（对应于该公式引入的斯科勒姆函数的grounding），该函数对每个向量$\mathbf{v}$返回特征向量$f(\mathbf{v})$，直观上可解释为与特征为$\mathbf{v}$的对象在关系R中参与的典型对象的特征集合。

Finally, related work in the domain of neural-symbolic computing and neural network fibring [10] has sought to combine neural networks with ILP to gain efficiency [14] and other forms of knowledge representation, such as propositional modal logic and logic programming. The above are more tightly-coupled approaches. In contrast, LTNs use a richer FOL language, exploit the benefits of knowledge compilation and tensor networks within a more loosely- coupled approach, and might even offer an adequate representation of equality in logic. Experimental evaluations and comparison with other neural-symbolic approaches are desirable though, including the latest developments in the field, a good snapshot of which can be found in [1].
最后，神经符号计算和神经网络融合领域的相关工作[10]试图将神经网络与归纳逻辑编程（ILP）结合以提高效率[14]，以及结合其他知识表示形式，如命题模态逻辑和逻辑编程。上述方法属于更紧耦合的方式。相比之下，LTN使用更丰富的一阶逻辑（FOL）语言，利用知识编译和张量网络的优势，采取更松耦合的方式，甚至可能提供逻辑中等式的适当表示。然而，仍需进行实验评估并与其他神经符号方法进行比较，包括该领域的最新进展，相关综述可见文献[1]。

13 7 Conclusion and future work

14 7 结论与未来工作

We have proposed Real Logic: a uniform framework for learning and reasoning. Approximate satisfiability is defined as a learning task with both knowledge and data being mapped onto real-valued vectors. With an inference-as-learning approach, relational knowledge constraints and state-of-the-art data-driven approaches can be integrated. We showed how real logic can be implemented in deep tensor networks, which we call Logic Tensor Networks (LTNs), and applied efficiently to knowledge completion and data prediction tasks. As future work, we will make the implementation of LTN available in TENSORFLOW $^{\text{TM}}$ and apply it to large-scale experiments and relational learning benchmarks for comparison with statistical relational learning, neural-symbolic computing, and (probabilistic) inductive logic programming approaches.

我们提出了Real Logic：一个统一的学习与推理框架。近似可满足性被定义为一个学习任务，知识和数据均映射为实值向量。通过推理即学习的方法，可以整合关系知识约束与最先进的数据驱动方法。我们展示了如何在深度张量网络中实现Real Logic，称之为逻辑张量网络（Logic Tensor Networks, LTN），并高效应用于知识补全和数据预测任务。未来工作中，我们将发布基于TENSORFLOW$^{\text{TM}}$的LTN实现，并将其应用于大规模实验和关系学习基准测试，以便与统计关系学习、神经符号计算及（概率）归纳逻辑编程方法进行比较。

15 References

16 参考文献

1. Cognitive Computation: Integrating Neural and Symbolic Approaches, Workshop at NIPS 2015, Montreal, Canada, April 2016. CEUR-WS 1583.

2. Cognitive Computation: Integrating Neural and Symbolic Approaches, Workshop at NIPS 2015, Montreal, Canada, April 2016. CEUR-WS 1583.

2. Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches, AAAI Spring Symposium, Stanford University, CA, USA, March 2015.

3. Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches, AAAI Spring Symposium, Stanford University, CA, USA, March 2015.

3. Dimitris Achlioptas. Random satisfiability. In Handbook of Satisfiability, pages 245-270. 2009.

4. Dimitris Achlioptas. Random satisfiability. In Handbook of Satisfiability, pages 245-270. 2009.

4. Leon Barrett, Jerome Feldman, and Liam MacDermed. A (somewhat) new solution to the variable binding problem. Neural Computation, 20(9):2361-2378, 2008.

5. Leon Barrett, Jerome Feldman, and Liam MacDermed. A (somewhat) new solution to the variable binding problem. Neural Computation, 20(9):2361-2378, 2008.

5. Yoshua Bengio. Learning deep architectures for ai. Found. Trends Mach. Learn., 2(1):1-127, January 2009.

6. Yoshua Bengio. Learning deep architectures for ai. Found. Trends Mach. Learn., 2(1):1-127, January 2009.

6. M. Bergmann. An Introduction to Many-Valued and Fuzzy Logic: Semantics, Algebras, and Derivation Systems. Cambridge University Press, 2008.

7. M. Bergmann. An Introduction to Many-Valued and Fuzzy Logic: Semantics, Algebras, and Derivation Systems. Cambridge University Press, 2008.

7. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. J. Mach. Learn. Res., 3:993-1022, March 2003.

8. David M. Blei, Andrew Y. Ng, 和 Michael I. Jordan. 潜在狄利克雷分配（Latent Dirichlet Allocation）。机器

学习研究杂志（J. Mach. Learn. Res.），3:993-1022，2003年3月。

8. Léon Bottou. From machine learning to machine reasoning. Technical report, arXiv.1102.1808, February 2011.
9. Léon Bottou. 从机器学习到机器推理。技术报告，arXiv.1102.1808，2011年2月。

9. Artur S. d'Avila Garcez, Marco Gori, Pascal Hitzler, and Luís C. Lamb. Neural-symbolic learning and reasoning (dagstuhl seminar 14381). Dagstuhl Reports, 4(9):50-84, 2014.
10. Artur S. d'Avila Garcez, Marco Gori, Pascal Hitzler, 和 Luís C. Lamb. 神经符号学习与推理（dagstuhl研讨会14381）。Dagstuhl报告，4(9):50-84，2014年。

10. Artur S. d'Avila Garcez, Luís C. Lamb, and Dov M. Gabbay. Neural-Symbolic Cognitive Reasoning. Cognitive Technologies. Springer, 2009.
11. Artur S. d'Avila Garcez, Luís C. Lamb, 和 Dov M. Gabbay. 神经符号认知推理。认知技术。施普林格出版社，2009年。

11. Michelangelo Diligenti, Marco Gori, Marco Maggini, and Leonardo Rigutini. Bridging logic and kernel machines. Machine Learning, 86(1):57-88, 2012.
12. Michelangelo Diligenti, Marco Gori, Marco Maggini, 和 Leonardo Rigutini. 架起逻辑与核机器的桥梁。机器学习，86(1):57-88，2012年。

12. David Silver et al. Mastering the game of go with deep neural networks and tree search. Nature, 529:484-503, 2016.
13. David Silver 等. 利用深度神经网络和树搜索掌握围棋游戏。自然（Nature），529:484-503，2016年。

13. Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
14. Martín Abadi 等. TensorFlow：异构系统上的大规模机器学习，2015年。软件可从tensorflow.org获取。

14. Manoel V. M. França, Gerson Zaverucha, and Artur S. d'Avila Garcez. Fast relational learning using bottom clause propositionalization with artificial neural networks. Machine Learning, 94(1):81-104, 2014.
15. Manoel V. M. França, Gerson Zaverucha, 和 Artur S. d'Avila Garcez. 使用底层子句命题化结合人工神经网络的快速关系学习。机器学习，94(1):81-104，2014年。

15. Ramanathan Guha. Towards a model theory for distributed representations. In 2015 AAAI Spring Symposium Series, 2015.
16. Ramanathan Guha. 面向分布式表示的模型理论。载于2015年AAAI春季研讨会系列，2015年。

16. Michael Huth and Mark Ryan. Logic in Computer Science: Modelling and Reasoning About Systems. Cambridge University Press, New York, NY, USA, 2004.
17. Michael Huth 和 Mark Ryan. 计算机科学中的逻辑：系统建模与推理。剑桥大学出版社，美国纽约，2004年。

17. Jeffrey O. Kephart and David M. Chess. The vision of autonomic computing. Computer, 36(1):41-50, January 2003.
18. Jeffrey O. Kephart 和 David M. Chess. 自主计算的愿景。计算机（Computer），36(1):41-50，2003年1月。

18. Douwe Kiela and Léon Bottou. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In Proceedings of EMNLP 2014, Doha, Qatar, 2014.
19. Douwe Kiela 和 Léon Bottou. 使用卷积神经网络学习图像嵌入以提升多模态语义。载于2014年EMNLP会议论文集，卡塔尔多哈，2014年。

19. Bart Kosko. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

20. Bart Kosko. 神经网络与模糊系统：面向机器智能的动力系统方法。Prentice-Hall出版社，美国新泽西州Upper Saddle River，1992年。

20. Brian Milch, Bhaskara Marthi, Stuart J. Russell, David Sontag, Daniel L. Ong, and Andrey Kolobov. BLOG: probabilistic models with unknown objects. In IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005, pages 1352-1359, 2005.

21. Brian Milch, Bhaskara Marthi, Stuart J. Russell, David Sontag, Daniel L. Ong, 和 Andrey Kolobov. BLOG：具有未知对象的概率模型。载于IJCAI-05，第十九届国际人工智能联合会议论文集，英国爱丁堡，2005年7月30日至8月5日，页1352-1359，2005年。

21. Stephen H. Muggleton, Dianhuan Lin, and Alireza Tamaddoni-Nezhad. Meta-interpretive learning of higher-order dyadic datalog: predicate invention revisited. Machine Learning, 100(1):49-73, 2015.

22. Stephen H. Muggleton, Dianhuan Lin, 和 Alireza Tamaddoni-Nezhad. 高阶二元Datalog的元解释学习：谓词发明的再探讨。机器学习，100(1):49-73，2015年。

22. Aniruddh Nath and Pedro M. Domingos. Learning relational sum-product networks. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA., pages 2878-2886, 2015.

23. Aniruddh Nath 和 Pedro M. Domingos. 学习关系和积网络。载于第二十九届AAAI人工智能会议论文集，2015年1月25-30日，美国德克萨斯州奥斯汀，页2878-2886，2015年。

23. Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2016.

24. Luc De Raedt, Kristian Kersting, Sriraam Natarajan 和 David Poole. 统计关系人工智能：逻辑、概率与计算. 人工智能与机器学习综述讲义. Morgan & Claypool 出版社, 2016.

24. Matthew Richardson and Pedro Domingos. Markov logic networks. Mach. Learn., 62(1- 2):107-136, February 2006.

25. Matthew Richardson 和 Pedro Domingos. 马尔可夫逻辑网络. 机器学习, 62(1-2):107-136, 2006年2月.

25. Tim Rocktaschel, Sameer Singh, and Sebastian Riedel. Injecting logical background knowledge into embeddings for relation extraction. In Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), June 2015.

26. Tim Rocktaschel, Sameer Singh 和 Sebastian Riedel. 将逻辑背景知识注入嵌入以进行关系抽取. 载于北美计算语言学协会年会 (NAACL), 2015年6月.

26. Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning With Neural Tensor Networks For Knowledge Base Completion. In Advances in Neural Information Processing Systems 26. 2013.

27. Richard Socher, Danqi Chen, Christopher D. Manning 和 Andrew Y. Ng. 使用神经张量网络进行知识库补全推理. 载于神经信息处理系统进展第26卷, 2013年.

27. T. Tieleman and G. Hinton. Lecture 6.5 - RMSProp, COURSERA: Neural networks for machine learning. Technical report, 2012.

28. T. Tieleman 和 G. Hinton. 讲座6.5 - RMSProp, COURSERA：机器学习中的神经网络. 技术报告, 2012年.

28. Leslie G. Valiant. Robust logics. In Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing, STOC '99, pages 642-651, New York, NY, USA, 1999. ACM.

29. Leslie G. Valiant. 鲁棒逻辑. 载于第三十一届ACM计算理论研讨会论文集, STOC '99, 页642-651, 纽约, 美国, 1999年. ACM.

29. Jue Wang and Pedro M. Domingos. Hybrid markov logic networks. In Proceedings of the Twenty-Third AAAI

Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008, pages 1106-1111, 2008.

30. Jue Wang 和 Pedro M. Domingos. 混合马尔可夫逻辑网络. 载于第二十三届美国人工智能协会年会论文集, AAAI 2008, 芝加哥, 伊利诺伊州, 美国, 2008年7月13-17日, 页1106-1111, 2008年.