简单人工网络逻辑解释的三元概念分析

Ingo Schmitt[0000−0002−4375−8677]

科特布斯-森夫滕贝格勃兰登堡工业大学，德国 schmitt@b-tu.de
https://www.b-tu.de/fg-dbis

arXiv:2601.06229v1 [cs.LG] 9 Jan 2026

摘要。 人工神经网络（ANN）是一种用于解决复杂分类问题的数值方法。由于分类能力强，ANN 方法在准确率方面往往优于其他分类方法。然而，与采用符号范式的方法相比，人工神经网络模型缺乏可解释性。我们的思路是，从一个以输入对象的最小项值训练而成的简单人工神经网络模型中推导出符号表示。基于 ReLU 节点，人工神经网络模型被划分为多个单元。我们将人工神经网络模型转换为基于单元的三维位张量。将形式概念分析理论应用于该张量，可得到以逻辑树表示的概念，这些逻辑树表达了可解释的属性交互。对这些概念进行评估后，其分类能力仍与初始人工神经网络模型相当。

关键词： 解释 · 人工神经网络 · 形式概念分析 · 逻辑

1 引言

除了准确率高之外，分类模型还应具备可解释性。人工神经网络模型（简称 ANN 模型）在许多分类任务中都能取得较高的准确率，但可解释性较低。ANN 模型这类数值方法会隐藏内部决策逻辑，导致人们缺乏信任 [13,9]。因此，人工神经网络常被称为黑箱解决方案。

在医学诊断或对人类生命有重大影响的决策等关键应用中，分类模型的可解释性是一项关键要求。与解释相关的一项任务是说明针对给定输入对象得出的分类结果。

许多研究者（例如 [10,14], 中引用的研究者）致力于让人工神经网络模型的黑箱行为更加透明、易于理解。这些方法大多不以逻辑为基础。例如，夏普利值 [22]及其改进方法 SHAP [12] 会通过加性度量计算单个输入属性的重要性。SHAP 方法将局部可解释模型无关解释（LIME） [16] 加以推广。另一种方案是在提取广义加性模型的基础上，提取岭函数和形状函数，以说明属性的影响。用于衡量输入属性重要性的方法

可以推广为衡量多个属性之间交互作用强度的方法；例如，参见 [15]。

请注意，包括其衍生方法 [23] 和 Tsetlin 机 [8] 在内的决策树分类器，均基于布尔逻辑决策。一般而言，逻辑表达式被认为比人工神经网络模型等数值型（有时也称为亚符号型）分类器更易于解释。 [2,6,3] 的研究是较早尝试将神经网络（亚符号范式）与逻辑（符号范式）衔接起来的工作。该研究定义了表示神经网络状态的图式，以及类似模糊逻辑的运算（合取和析取）。然而，这些运算并不遵循布尔代数定律。

[7] 中的研究讨论了信息处理的符号范式与亚符号范式衔接语境下的“不相容性”和“实现”概念，并将这两种范式都与量子力学的概念联系起来。

一种有趣的方法 [1,25] 是通过使用神经网络模拟决策树，将决策树形式的逻辑与深度神经网络衔接起来。在学习并组合属性值的分箱后，为每种分箱组合训练线性网络，以学习分类模型。相比之下，我们的工作以一个给定的、经过训练的简单人工神经网络模型为出发点，尝试在训练 后对其进行解释。此外，[1,25] 中的分区基于由 Softmax 函数模拟的决策树分裂节点，而我们的方法则基于 ReLU 节点。

我们方法的主要思想是利用概率论、形式概念分析和受量子逻辑启发的决策树（QLDT [21]），将训练后的人工神经网络模型映射为非布尔逻辑树。我们得到逻辑树的线性组合，从而有助于解释给定人工神经网络模型的语义。这种方法类似于 RuleFit 方法 [4], ，后者依赖于由决策树导出的规则的线性组合。然而，决策树基于布尔逻辑，并且对于连续输入值缺乏平滑性。

对于我们的解释方法，我们定义了针对一个使用平衡训练数据 TR 训练完成的单类人工神经网络分类器模型的一些限制条件，这两者都是我们的出发点：

– 人工神经网络由线性层（例如卷积、平均池化和全连接层）、一个仅包含 ReLU 节点的层以及一个输出节点组成。– 分类的目标输出为：类别决策为 1，否则为 0。目标值通过对输出节点应用阈值算子 τ 得到。– 输入对象属性的数量 $\#atts$ 很少。每个属性的值都位于单位区间 $[0,1] \in \mathbb{R}$ 内，表示某个人类可理解属性的满足程度。例如，参见表1左侧的两个训练对象（x_1, y_1），（x_2, y_2）以及两个属性a_1, a_2及其属性值 $x_i[1], x_i[2]$。– 人工神经网络具有 $2^{\#atts}$ 个输入节点，用于接收输入对象的最小项值（定义如下）。

表1. 两个示例对象（x_1，y_1），（x_2，y_2）在两个属性 a_1，a_2 上的取值，其中 $x_i =$（$x_i[1]$, $x_i[2]$）（左侧部分）以及四个导出的最小项（右侧部分）：$mt_i[0] =$（$1 - x_i[1]$）·（$1 - x_i[2]$）$mt_i[1] =$（$1 - x_i[1]$）$\cdot x_i[2]$，$mt_i[2] = x_i[1] \cdot$（$1 - x_i[2]$）$mt_i[3] = x_i[1] \cdot x_i[2]$

对象	属性 取值		目标	导出的最小项			
i	$x_i[1]$	$x_i[2]$	y_i	$mt_i[0]$	$mt_i[1]$	$mt_i[2]$	$mt_i[3]$
1	0.8	0.1	1	0.18	0.02	0.72	0.08
2	0.5	0.6	0	0.2	0.3	0.2	0.3

要对人工神经网络模型进行逻辑解释，我们需要将 $2^{\#atts}$ 最小项值$mt_i[k]$而非 $\#atts$ 值 $x_i[j]$ 作为输入。最小项值是取反（$1 - x_i[j]$）和未取反的属性值$x_i[j]$的合取（表示为乘积）。每个最小项值都表示所有属性之间的一种特定交互作用。根据 [21,20], ，我们定义：

$$mt_i[k] := \Pi_{j=1}^{\#atts} \left(1 - x_i[j]\right)^{1-b_j} \cdot x_i[j]^{b_j}$$

其中整数 $k = 0 \ldots 2^{\#atts} - 1$ $b =$,,是 $b_1 \ldots b_{\#atts}$ $k \quad k = \sum_{j=0}^{\#atts-1} b_{\#atts-j} \cdot 2^j$ 最小项标识符，也是（ ）的位编码。可以证明 $\sum_k mt_i[k] = 1$ 成立。例如，参见表1右侧部分的 2^2最小项值 $mt_i[k]$。

通过我们的方法，可以得到逻辑树。图1展示了三个非常简单的示例及其算术求值。稍后我们将更详细地介绍这类逻辑树。

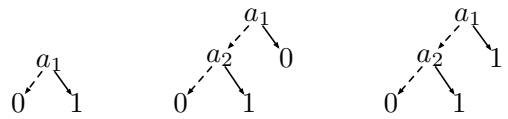


图1。 逻辑树示例：实线表示未取反的属性求值，虚线表示取反的属性求值。逻辑树的算术求值如下：$[a_1]^i = x_i[1]$（左）、$[a_1 \wedge a_2]^i =$（$1 - x_i[1]$）$\cdot x_i[2]$（中）以及$[a_1 \vee$（$a_1 \wedge a_2$）$] = x_i[1] +$（$1 - x_i[1]$）$\cdot x_i[2]$（右）。

2 人工神经网络分区单元

人工神经网络中的 ReLU 节点由以下函数定义

$$ReLU(z) = \begin{cases} z \text{ if } z \geq 0 \\ 0 \; otherwise. \end{cases}$$

假设给定的人工神经网络模型有 l 个 ReLU 节点，记为 $ReLU_1$，…，$ReLU_l$。对于每个输入对象，ReLU 节点的状态要么是激活，对应于

恒等函数；要么是非激活，对应于零函数。下文中，我们用比特 1 表示 ReLU 节点处于激活状态，用比特 0 表示节点处于非激活状态。由于 ReLU 节点下方的线性层，其激活与非激活状态之间的分界对应于输入空间 $[0,1]^{2^{\#atts}}$ 中的一个超平面。

对于输入对象 x_i，我们将所有 l ReLU 节点的状态位序列解释为一个整数，并称之为其分区编号 p_i。对于给定的分区编号 p，我们将 $active(p)$ 定义为激活的 ReLU 节点集合。分区编号用于标识输入空间 $\mathbb{R}^{2^{\#atts}}$中的划分单元：具有相同分区编号的输入对象属于同一个划分单元。由于每个 ReLU 节点都对应一个超平面，因此每个划分单元都会形成 $\mathbb{R}^{2^{\#atts}}$ 的一个凸子集。所有划分单元彼此不相交。由于人工神经网络模型中除 ReLU 节点外的所有层都是线性的，因此对于给定的划分单元 p，我们可以将这些线性层合并为一个线性映射 mw^p，从输入最小项值映射到输出节点值。这里，$mw^p[k]$ 表示最小项k的最小项权重。表 2 给出了一个关于 2^2 个最小项的划分单元线性映射示例。

表 2. 最小项 $k=0$ 的示例值 $mw^p[k]$，...，3

划分	最小项权重			
p	$mw^p[0]$	$mw^p[1]$	$mw^p[2]$	$mw^p[3]$
1	-8	3	6	2
⋮	⋮	⋮	⋮	⋮

对于给定的人工神经网络模型，最多有 2^l 个划分单元。如果某个单元 p 完全位于输入空间 $[0,1]^{2^{\#atts}}$ 之外，则不存在任何输入对象 x_i 使得 $p_i = p$ 成立。所有划分单元共同构成一个关于相应集合 $active(p)$ 的原子布尔子集格。图 2 展示了一个包含三个 ReLU 节点和八个划分单元的示例格。

对于每个划分单元，我们使用以下信息：

– 训练对象的数量 i，其中 $p_i = p$– 1对象和 0对象的数量，
其中 $p_i = p$– 最小项权重：mw^p

根据这些信息，并非所有划分单元对于解释人工神经网络模型都至关重要。没有分配训练对象的单元，或只包含 1对象或只包含 0对象的单元，相关性较低，因为它们无助于区分输入对象。包含大量训练对象，且 1对象和 0对象出现次数均衡的单元则具有较高相关性。

由于我们的人工神经网络架构，可以通过将某个分区单元所对应的原子划分单元中的权重相加，推导出该分区单元的所有最小项权重。A

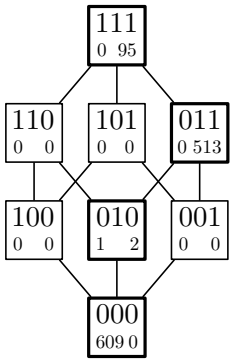


图 2. 以三个 ReLU 节点的状态位码标识的划分单元格格示例：左侧的小数字表示 TR 中零对象的数量（$y_i = 0$），右侧的小数字表示一对象的数量（$y_i = 1$）。粗体划分表示关注的划分（非空）。

当且仅当恰有一个 ReLU 节点处于激活状态时，该分区单元被视为原子单元：

$$mw^p = \sum_{i:b_i=1} mw^{2^{l-i}}$$

其中，$b_1 b_2 \ldots b_l$ 是编号为 p 的分区单元的位编码。推导示例为 mw^{101} : $= mw^{100} + mw^{001}$。这种推导非常有用，因为只需存储原子划分单元格的最小项值，即 $l \cdot 2^{\#atts}$ 值。在以下各节中，我们假设 P 是本质划分单元格的集合。

3 将最小项权重转换为位张量

给定所有分区单元 $p \in P$ 在最小项 k 上的最小项权重 $mw^p[k]$，例如表 2 所示。人工神经网络模型的输入是一个对象 x_i，其每个最小项 k 都有对应的最小项值 $mt_i[k]$，并被分配到分区单元 $p = p_i$，例如表1所示。得分 ANN(x_i) 可写为标量积 ANN(x_i) := $\langle mt_i,\ mw^p \rangle = \sum_k mt_i[k] \cdot mw^p[k]$。例如，使用表 1 中的对象（$p_1 = p_2 = 1$）以及表 2 中的最小项权重，可得得分 ANN($x_1$) = 3.1 和 ANN($x_2$) = 1.1。

如果我们假设所有对象的得分值各不相同，那么它们的得分就可以严格排序。对于我们的解释方法，我们对最小项权重 $mw^p[k]$应用严格单调递增线性函数 f，将其转换为位值。由于标量积对其参数是线性的，对最小项权重应用 f 会保留对象的严格排序：

$$\langle mt_i, f(mw^p) \rangle \ = \ f\left(\langle mt_i, mw^p \rangle\right)$$
$$\langle mt_{i_1}, mw^p \rangle > \langle mt_{i_2}, mw^p \rangle \Leftrightarrow f\left(\langle mt_{i_1}, mw^p \rangle\right) > f\left(\langle mt_{i_2}, mw^p \rangle\right)$$

因此，将修改后的阈值 $f(\tau)$应用于得分$f(\text{ANN}(x_i))$，可以保留原有的类别判定。

接下来，我们从映射后的最小项值 $f(mw^p[k])$ 中去除小数部分，并将剩余的整数编码为一个由 $\#bits$ 位组成的二进制数。去除小数部分会引入截断误差。因此，如果我们将截断后的最小项权重应用于输入对象，得分不同的对象可能会获得相同的得分，这意味着严格分数排序会变成弱序。

为了确定线性函数 f，我们推导出一个最小项值区间 $[a,b]$，其中 $a = \min_{p\in P,\ k} mw^p[k]$ 以及 $b = \max_{p\in P,\ k} mw^p[k]$。我们将每个最小项权重 $v := mw^p[k]$ 编码为一个具有 $\#bits$ 位的整数 v'：

$$v' := \lfloor f(v) \rfloor := \left\lfloor \frac{v-a}{b-a} * \frac{2^{\#bits}}{1+\epsilon} \right\rfloor \in \{0,1,\ldots,2^{\#bits}-1\}. \tag{1}$$

分母 $1+\epsilon$（其中 ϵ 是一个较小的值，例如 $2^{-(\#bits+4)}$）是获得 $2^{\#bits}-1$ 而非 $2^{\#bits}$ 所必需的，对于 $v := b$ 而言如此。该映射如图3所示。

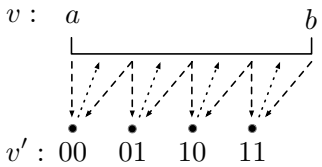


图 3. 虚线：最小项权重到二进制数的有损映射（$\#bits = 2$）；点线：重构函数

表3展示了我们引入的示例最小项权重的映射。将映射后的最小项值应用于表1中的示例对象，可得分数 $\text{ANN}(x_1)' = 2.38$ 和 $\text{ANN}(x_2)' = 2.1$。

表3. 分区 $p = 1$ 中所有最小项 $k = 0,\ldots,3$ 在表 2 中映射得到的值 $mw^p[k]$，其中 $\#bits = 2$；最后两行显示了 $\lfloor f(mv^p[k]) \rfloor$ 相对于两个位级的位值

	最小项 k			
	0	1	2	3
$mw^p[k]$	-8	3	6	2
$f(mv^p[k])$	0	3.14	3.99	2.85
$\lfloor f(mv^p[k]) \rfloor$	0	3	3	2
$bl = 0$: 2^0	0	1	1	0
$bl = 1$: 2^1	0	1	1	1

接下来，我们定义一个线性重构函数，尝试从 v' 重构v ，并使平均重构误差较小。在 $v'' \approx v$的情况下，用于从 v'得到 v'' 的线性重构函数如下：

$$v'' := \left(v' * \frac{1+\epsilon}{2^{\#bits}} + 2^{-(\#bits+1)} \right) * (b-a) + a. \tag{2}$$

为获得较小的重构误差，术语 $2^{-(\#bits+1)}$ 选择截断区间的中点，如图3所示。假设最小项值服从均匀分布，则平均重构误差为：

$$E(|v - v''|) = \frac{b-a}{2^{\#bits+2}}$$

上界为：

$$|v - v''| \leq \frac{b-a}{2^{\#bits+1}}.$$

由于重构误差随位数增加而呈指数级下降，因此参数 $\#bits$ 可以较小，从而确保重构误差较小。

映射所有最小项值后，还需要映射类别决策阈值 τ。最小项值的截断会使得分值产生 1/2 的平均误差。因此，我们选择 $\tau' = f(\tau) - 1/2$。

在我们的示例中，假设初始值为 $\tau = 2$ ，并得到 $\tau' = 2.36$。因此，经过最小项权重映射后，阈值 τ' 能够有效地区分对象 x_1 和 x_2 （$\mathrm{ANN}(x_1) = 3.1$、$\mathrm{ANN}(x_2) = 1.1$，以及 $\mathrm{ANN}(x_1)' = 2.38$、$\mathrm{ANN}(x_2)' = 2.1$）。

值 $v' = \lfloor f(mw^p[k]) \rfloor$ 被视为二进制数。例如，请参见表3中的最后两行。由训练后的人工神经网络模型得出的每一位取决于分区单元 $p \in P$、最小项 $k \in \{0, 1, \ldots, 2^{\#atts} - 1\}$和位级别 $bl \in \{0, 1, \ldots, \#bits - 1\}$的组合：

$$bt(p, k, bl) := \lfloor f(mw^p[k]) \rfloor \,\&\, 2^{bl}$$

其中， bt是三维比特张量， $'\&'$ 表示按位与。因此，人工神经网络模型的语义可以通过一个由位值组成的三维张量 $bt[p, k, bl]$ 近似表示。

根据这一构造，我们可以将输入对象 x_i 的 $p_i = p$ 的 $f(\mathrm{ANN}(x_i))$ 近似形式化为：

$$f(\mathtt{ANN}(x_i)) = \sum_{k:=0}^{2^{\#atts}-1} mt_i[k] \cdot f(mw^p[k]) \approx \sum_{k:=0}^{2^{\#atts}-1} mt_i[k] \cdot \lfloor f(mw^p[k]) \rfloor$$

and

$$\begin{aligned} \sum_{k:=0}^{2^{\#atts}-1} mt_i[k] \cdot \lfloor f(mw^p[k]) \rfloor &= \sum_{k:=0}^{2^{\#atts}-1} mt_i[k] \cdot \sum_{bl:=0}^{\#bits-1} bt(p,k,bl) \cdot 2^{bl} \\ &= \sum_{bl:=0}^{\#bits-1} 2^{bl} \cdot \sum_{k:=0}^{2^{\#atts}-1} mt_i[k] \cdot bt(p,k,bl) \end{aligned}$$

给定一个位张量 bt，我们计算其能量：

$$power(bt) := \sum_{p\in P}\sum_{k:=0}^{2^{\#atts}-1}\sum_{bl:=0}^{\#bits-1} bt[p,k,bl]\cdot 2^{bl} = \sum_{p\in P}\sum_{k:=0}^{2^{\#atts}-1} \lfloor f(mw^p[k])\rfloor.$$

能量可以理解为所有分区单元中位编码最小项权重的总贡献。它近似于给定人工神经网络模型中所有最小项权重之和。

4 形式概念分析与三元概念分析

形式概念分析（FCA）是由 Ganter 和 Wille[11,5,24]提出的一种理论。在我们的方法中，我们使用形式概念分析将张量 bt 转换为紧凑表示，随后再将其解释为逻辑树。

形式概念分析基于二元关系，而三元概念分析（TCA）可视为其向三元关系的推广。下面，我们定义形式概念分析的主要组成部分：二元背景是一个三元组（G，M，I），其中 G 是对象集合，M 是属性集合[1]。$I \subseteq G \times M$ 是定义在 G 和 M 上的二元关系。$(g,\ m) \in I$，也写作 gIm，表示对象 g（属于 G）具有属性 m（属于 M）。背景可以用交叉表表示，例如表 4 左侧所示，其中交叉符号表示 gIm：$= (g,\ m) \in I$。列代表对象，行代表属性。对于子集 $A \subseteq G$，我们定义函数 $'$：$2^G \to 2^M$，其定义为 $A' :=\{m \in M | gIm$，对所有 $g \in A\}$.均成立。类似地，对于子集 $B \subseteq M$，我们定义函数 $'$：$2^M \to 2^G$，其定义为B'：$= \{g \in G | gIm$，对所有 $m \in B\}$. 均成立。基于这些函数，背景的一个概念是一个二元组（A，B），其中 $A \subseteq G$，$B \subseteq M$，$A' = B$且 $B' = A$。换言之，$A \times B \subseteq I$（笛卡尔积规则）成立，并且在不违反笛卡尔积规则的前提下，A 和 B 均无法扩展。若将背景表示为交叉表，则概念（A，B）对应于经过某种列和行排列后、由交叉符号填满的一个极大矩形，例如表 4 中的第一个概念。

表 4. 示例背景及若干概念；粗体叉号对应第一个概念

	g_1	g_2	g_3
m_1		×	
m_2		×	×
m_3	×	×	×

$$concepts = \left\{ \begin{array}{c} (\boldsymbol{\{g_2, g_3\}, \{m_2, m_3\}}) \\ (\{g_1, g_2, g_3\}, \{m_3\}) \\ (\{g_2\}, \{m_1, m_2, m_3\}) \end{array} \right\}$$

对于给定的背景，最多可以存在 $2^{\min\{|G|,|M|\}}$ 个概念。所有概念共同构成偏序集 $(\{(A'',\ A') | A \subseteq$上的布尔子集格

[1] “对象”和“属性”是抽象术语，并不指代我们分类任务中的输入对象及其属性值。

$G\}$, $\trianglelefteq$) 与 $(A_1'', A_1') \trianglelefteq (A_2'', A_2') \Leftrightarrow A_1'' \subseteq A_2''$。概念可以重叠，例如，$(\{g_2\}, M)$ 和 $(G, \{m_3\})$ 与 $g_2 I m_3$ 重叠，这意味着它们可以共享交叉部分。

我们从单元素集合推导出一组概念（不一定包含所有概念），如下所示：

$$\texttt{concepts} := \{(\{a\}'', \{a\}') | a \in G\} \cup \{(\{b\}', \{b\}'') | b \in M\}.$$

由此得到的概念不超过 $|G|+|M|$ 个。注意，每个交叉项 $(g, m) \in I$至少被概念集合中的一个概念覆盖。

现在，我们将形式概念分析推广到三元概念分析（TCA）：三元背景是一个四元组 (K_1, K_2, K_3, Y)，其中 K_1 是对象集合，K_2 是属性集合，K_3 是条件集合。$Y \subseteq K_1 \times K_2 \times K_3$ 是定义在 K_1、K_2 和 K_3 上的三元关系。表达式 $(g, m, b) \in Y$ 表示对象 g 在条件 b 下具有属性 m。对于三元背景、$X_2 \subseteq K_2$ 和 $X_3 \subseteq K_3$，我们定义函数 $^1: 2^{K_2} \times 2^{K_3} \to 2^{K_1}$: $(X_2, X_3)^1 := \{x_1 \in X_1 | (x_1, x_2, x_3) \in Y$，对所有 $(x_2, x_3) \in X_2 \times X_3\}$. 均成立。类似地，我们定义函数 $(X_1, X_3)^2$ 和 $(X_1, X_2)^3$，分别用于计算 X_2 和 X_3。三元概念是一个三元组 (X_1, X_2, X_3)，满足 $X_1 = (X_2, X_3)^1$、$X_2 = (X_1, X_3)^2$，和 $X_3 = (X_1, X_2)^3$。三元概念可视为满足笛卡尔积规则 $X_1 \times X_2 \times X_3 \subseteq Y$ 的极大长方体。从定义可知，给定两个集合 X_i、X_j 后，三元概念便唯一确定。对于给定的三元背景 (K_1, K_2, K_3, Y)，我们按切片逐一计算三元概念，方法如下：

$$\texttt{triconcepts} := \bigcup_{x_1 \in K_1} \{(X_2, X_3)^1, X_2, X_3) | (X_2, X_3) \in \texttt{concepts}^{x_1}\},$$

其中，概念^{x_1} 是从由三元背景派生的 x_1切片二元背景（G, M, I）中计算得到的二元概念：$(x_2, x_3) \in I \Leftrightarrow (x_1, x_2, x_3) \in Y$以及 $G := K_2$, $M := K_3$.

5 从互斥三元概念生成逻辑树

在第3节中，我们从人工神经网络模型中推导出了一个三维张量 bt。我们将其解释为一个三元背景（K_1, K_2, K_3, Y），其中包含 $K_1 = P$、$K_2 = \{0, 1, \ldots, 2^{\#atts}-1\}$、$K_3 = \{0, 1, \ldots, \#bits - 1\}$以及 $Y = \{(p, k, bl) | bt[p, k, bl] = 1\}$。例如，我们可以将表3中该分区的最后两行视为一个三元背景：$K_1 = \{p\}$, $K_2 = \{0, 1, 2, 3\}$, $K_3 = \{0, 1\}$，并且

$$Y = \{(p, 1, 0), (p, 2, 0), (p, 1, 1), (p, 2, 1), (p, 3, 1)\}.$$

我们将三元概念计算为 $\{c_j(X_1, X_2, X_3)\} := $ 三元概念。根据表3中的示例，我们得到两个概念：$c_1 := c_1(\{p\}, \{1, 2, 3\}, \{1\})$ 和 $c_2 := c_2(\{p\}, \{1, 2\}, \{0, 1\})$。概念可以看作是一个完全由置位比特填充的极大长方体。类似于上下文的能量，我们现在将概念 c_j 的能量 定义为其分区单元数量、其最小项数量以及其位编码层级上的

其最小项的数量，以及其位编码层级上的 $powersum := \sum_{x_3 \in X_3} 2^{x_3}$ ：

$$power(c_j(X_1, X_2, X_3)) := |X_1| * |X_2| * \sum_{x_3 \in X_3} 2^{x_3}$$

两个示例概念的能量为： $power(c_1) = power(c_2) = 6$. 结合上下文能量，我们将相对概念能量 定义为$relpower(c) := power(c)/power(bt)$[2].

我们将三元概念的最小项 X_2 解释为逻辑表达式，其相对功率值由 X_3 计算得出，并且适用于 X_1中分区单元的对象。正如上一节所述，不同三元概念的长方体可能相互重叠，进而产生重叠的逻辑表达式。我们的目标是获得能量较高且互不重叠的逻辑表达式。因此，我们从一个三元背景中计算出一个数组 ex_c ，其中互斥三元概念按相对能量降序排列。于是， $\sum_{xc \in ex_c} relpower(xc) = 1$ 成立；换句话说， bt 中的每个置位都恰好包含在一个互斥三元概念 xc中。所需数组的计算方法见算法 4。根据表3中的示例，我们得到： $ex_c[0] = c_0(\{p\}, \{1,2\}, \{0,1\})$，其$power(ex_c[0]) = 6$，以及 $ex_c[1] = c_1(\{p\}, \{3\}, \{1\})$，其 $power(ex_c[1]) = 2$。

```
1  函数 排除_三元概念（三元背景 bt）：数组 三元概念的
2    ex_三元概念: = []  // 空结果数组
3    C: = triconcepts(bt)
4    xc: =arg max_{c∈C} relpower(c) //幂值最高的概念
5    ex_triconcepts.append(xc)
6    // 通过幂值最高的概念简化三元背景
7    对于 x_1 ∈xc.X_1
8      对于 x_2 ∈xc.X_2
9        对于 x_3 ∈xc.X_3
10         bt[x_1, x_2, x_3]:= 0 // 无交叉
11   if 幂（bt） > 0  // 三元背景是否非空
12     转到 线 3
13   return ex_三元概念
```

图 4. 提取互斥三元概念

在算法中，我们在第 4 行使用了 $relpower$ 来寻找最佳三元概念。不过，这只是四种可能方法中的第一种：

1. M1：相对功率（#$partitions$ * #$minterms$ * $powersum$）
2. M2：#$partitions$ * $powersum$3. M3： $powersum$

[2] 或者，对于两个能量公式，我们也可以使用所覆盖划分单元中的 TR 对象数（支持度），而不是 $|X_1|$ 单元数。

4. M4：覆盖的划分单元中对象的准确率

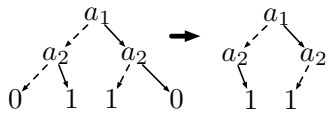
我们将在第7节比较它们对逻辑树生成的影响。

来自三元概念 $c_j(X_1,\ X_2,\ X_3)$ 的最小项标识符集合 X_2 并不是逻辑表达式的直观表示。因此，我们从 X_2 构建受量子逻辑启发的决策树（QLDT [21]），并将其呈交专家进行解释。QLDT 是由 X_2 定义的逻辑表达式的紧凑表示。主要构建思路是将活跃和非活跃最小项的位码用作训练数据

$$\{(\text{bit-code}(k),1)|k\in X_2\}\cup\{(\text{bit-code}(k),0)|k\in K_2\setminus X_2\}$$

以训练经典决策树。最小项 k 的位编码中的各位表示所有 n 原始属性，可以是取反或未取反。生成的决策树中靠近根节点的属性，比距离根节点较远的属性更能有效地区分活跃最小项和非活跃最小项。

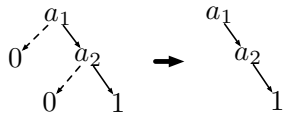
根据示例三元概念 $ex_c[0]$ 及 $X_2=\{1,2\}$，我们得到训练数据：{(00，0)，(01，1)，(10，1)，(11，0)}。由 X_3 得出的 $powersum$ 是$2^0+2^1=3$。最终的逻辑树如图5所示。



$$3*[(a_1\wedge\overline{a}_2)\vee(\overline{a}_1\wedge a_2)]^i$$
$$:=3*(x_i[1]*(1-x_i[2])+(1-x_i[1])*x_i[2])$$

图5. $ex_c[0]$的示例逻辑树；实线表示未取反，虚线表示取反求值，可省略通往0叶节点的路径；下方为对象 x_i 的算术求值（见第6节）

图6展示了来自 $ex_c[1]$ 且具有 $X_2=\{3\}$的逻辑树，训练数据为{(00，0)，(01，0)，(10，0)，(11，1)}和 $powersum$ $2^1=2$。



$$2*[a_1\wedge a_2]^i:=2*x_i[1]*x_i[2]$$

图6. $ex_c[1]$的示例逻辑树；实线表示未取反，虚线表示取反求值，可省略通往0叶节点的路径；下方为对象 x_i的算术求值

6 逻辑树的评估与解释

三元概念的最小项可视为析取范式逻辑表达式的最小项。由于输入对象的最小项值是连续的，按照我们的方法评估逻辑表达式并非基于布尔值。然而，如 [18,19],中所证明的，底层可交换量子逻辑在统计独立的属性值上遵循布尔代数和概率论的规则。评估结果是一个连续得分值，通过阈值算子（阶跃函数）映射为类别决策 τ。

逻辑树是基于最小项对逻辑表达式进行紧凑表示的一种形式，具有良好的可解释性。内部树节点根据单个属性进行分支，每条指向子节点的边都对应于对该属性进行取反或未取反的求值。与决策树类似，叶节点表示类别决策。对于我们的单类分类问题，我们忽略 0 叶节点，只关注 1 叶节点。[3] 令 L_1 为从根节点到 1 叶节点的所有路径的集合。给定路径 $l \in L_1$上的所有文字，无论取反与否，都以合取方式组合。 L_1路径上的每个属性最多出现一次。内部节点依据某个属性分支为两个子节点，其中一个取反，另一个未取反，这对应于异或。

根据概率论的规则，我们使用以下三条规则评估逻辑树：

1. 取反事件的概率值通过用 1 减去该事件的概率来计算。2. 独立事件合取的概率值通过算术乘积计算。3. 异或的概率值通过算术求和计算。

由此，我们得到逻辑树 $qldt$ 针对对象 x_i 的求值结果，如下所示：
$= \sum_{l \in L_1} \prod_{e_{lj} \in path(l)} [e_{lj}]^i$，其中，对于未取反属性 a_j， $[e_{lj}]^i := x_i[j]$ 成立；对于取反属性， $[e_{lj}]^i := 1 - x_i[j]$ 成立。假设有一个特殊的输入对象，其属性值仅为 0 和 1。在这种情况下， $qldt$求值与布尔逻辑一致。因此，我们的求值公式$[qldt]^i$ 可视为布尔逻辑的一种推广。

由若干互斥概念，我们得到若干逻辑树。要对输入对象 x_i求值，我们选择覆盖该对象的概念所对应的逻辑树：

$$QLDT(i) = \{qldt(c) | c \in \texttt{excl_triconcepts(bt)} \wedge p_i \in c.X_1\},$$

将它们的加权树评估值相加，得到得分值：

$$score(x_i) = \sum_{qldt(c) \in QLDT(i)} powersum(c) * [qldt(c)]^i.$$

[3] 关注0叶节点等同于逻辑否定关注1叶节点。

如果 $score(x_i) > \tau'$，那么我们得到类别值1；否则，我们得到0。

由于我们将位张量映射到逻辑树，因此对于具有 $p = p_i$ 的输入对象 x_i，可得到以下等价关系：

$$\sum_{qldt(c)\in QLDT} powersum(c) * [qldt(c)]^i = \sum_{bl:=0}^{\#bits-1} 2^{bl} \cdot \sum_{k:=0}^{2^{\#atts}-1} mt_i[k] \cdot bt(p,k,bl).$$

对于我们的示例对象，在 $ex_c[0]$ 和 $ex_c[1]$ 上进行加权和可得到 $score(x_1) = 2.38$ 以及 $score(x_2) = 2.1$。也就是说，对逻辑树进行加权和可以近似最小项求值。

最小项权重和逻辑树的解释

在进行解释时，领域专家与逻辑和分类专家应共同协作。下文将他们称为 专家。

根据分区单元的最小项权重，我们可以使用沙普利公式 [17] 计算各个属性在这些最小项中的贡献：

$$Sh_i = \sum_{S\subseteq N\setminus\{i\}} \frac{(n-1-|S|)!|S|!}{n!} \left(v(S\cup\{i\}) - v(S)\right).$$

核心思想是将最小项的所有未取反属性解释为一个集合。计算夏普利值的输入是对象属性 a_1、.....、 a_n 的索引集合 $N := \{1,\ldots,n\}$，以及最小项值 $v(S) := mw^p[k]$，其中$k = \sum_{i\in S} 2^{i-1}$ 由被视为最小项 k的未取反属性的属性组成。请注意，在我们的例子中， v 不一定是单调的，并且 $v(\emptyset) = mw^p[0]$ 不必为零。将所有夏普利值相加，可得：

$$\sum_i Sh_i = v(N) - v(\emptyset) = mw^p[2^n - 1] - mw^p[0].$$

按上述方法计算出的夏普利值与对象无关。如果我们希望它们依赖于输入对象 x_i，则计算 $v(S) := mt_i[k] \cdot mw^p[k]$. 夏普利值的一个普遍缺点是无法体现属性之间的交互作用。

我们提出的解释方法从人工神经网络模型中推导出逻辑树，因此是一种依赖于模型且内在的解释方法。逻辑树表示用于解决分类问题的属性之间复杂的交互作用。

树背后的逻辑表达式可以与专家提出的另一逻辑表达式（即一个假设）进行比较。为了进行比较，可以分析其底层最小项集合之间的交集。例如，如果假设中的每个最小项都包含在树表达式中，那么该假设就蕴含树表达式。有关更深入的讨论，请参见 [21]。

如果推导出的逻辑树 $qldt(c)$ 过于复杂，难以供人理解，就可以对其通往叶节点 $l \in L_1$ 的各条路径分别进行解读。根据树求值可知，各条路径求值只需

相加即可。每条叶路径都可以解读为文字的合取。作为树的一部分，叶路径 $l \in L_1$ 具有以下性质，这些性质独立于 TR，且对解读很有价值：

– 叶深度：树中的深度 $depth(l)$对应路径属性的数量，并说明覆盖了多少个最小项： $2^{n-depth(l)}$.此处， n 是属性总数。– 树的权重： $powersum(c)$– 分区单元数量： $|c.X_1|$

对于给定的输入对象 x_i 及其 $p = p_i$，路径评估显示其对得分的贡献 $[l]^i$: $= \prod_{j \in path(l)} [a_{lj}]^i$ 。最后，如果我们取训练（或测试）数据 T，就可以计算值 $target = 0$ 或 $target = 1$ 对得分的平均路径贡献 $\frac{1}{|T|} \sum_{(x_i, target) \in T} [l]^i$，以及准确率、召回率和精确率等指标。接下来将给出实验场景中的示例。

7 人工神经网络模型的实验解释

实验数据集为输血服务中心数据集[4]。我们需要开发一个分类器，以预测某人是否会献血。为了对每个人作出判断，我们有以下信息：近期性 r（距上次献血的月数）、频率 f（献血总次数）、血液捐献量 m（捐献的血液总量，单位为 c.c.）以及时间 t（距首次献血的月数）。在平衡情况下，178 人献过血，另有 178 人未献血。

我们训练了一个包含 5 个 ReLU 节点、16 个输入节点和 1 个输出节点的人工神经网络模型。最终决策阈值 τ 通过选择能够达到最佳准确率的训练得分值来确定。输血问题是一个困难的分类问题，而我们的模型达到了 74% 的准确率。我们从网络中推导出了其划分单元。图 7 展示了其非空划分单元所构成的偏序集的哈斯图。我们选取划分单元（1、5、17、20、28）作为关键单元：总计 356 人中有 305 人属于这些关键单元。

表5. 根据本质划分单元格的最小项计算得出的四个属性的 Shapley值；绝对值最大的项以粗体标出

分区单元	近期性	频率	血液捐献量	time
1	**0.044**	−0.026	−0.026	0.008
5	**-0.075**	0.041	0.032	0.021
17	**0.062**	0.001	0.008	0.049
20	**-0.101**	0.095	0.093	0.054
28	0.081	**-0.119**	0.011	0.005

[4] https://archive.ics.uci.edu/ml/datasets.php

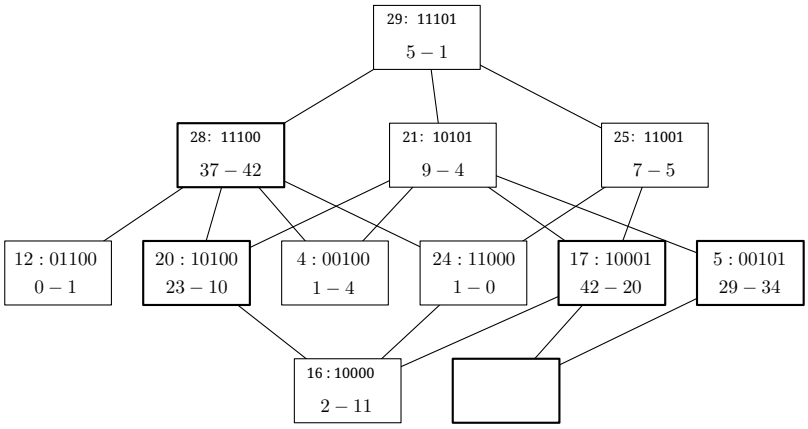


图7。 输血：非空划分单元的偏序集；位编码显示 ReLU节点的活动状态；活动 ReLU节点集合之间的子集关系构成一个偏序集；左下方的数值显示 1对象的数量；右下方的数值显示 0对象的数量；选定的单元（1、5、17、20、28）具有较高的支持度（下方数值之和），并以粗线框标示

表5展示了关键划分单元的夏普利值。属性“时间”在所有关键划分单元中均为正值，但影响较小。属性“近期性”和“频率”的影响较大；然而，它们在一些划分单元中为正值，在另一些划分单元中则为负值。这表明，这些属性根据其他属性的不同而发挥不同作用（交互作用较强），并且不同单元中人们的行为并不具有同质性。

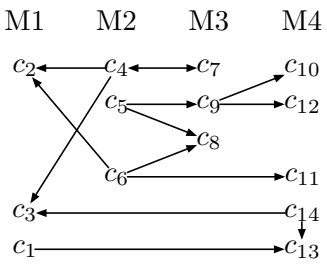


图8。 输血：概念及其蕴含关系（最小项集合的子集和划分单元集合的超集）

根据最小项值，我们计算出一个 7 位的位张量。对所得概念进行测试，准确率为 74%，这表明截断误差足够小。表6展示了采用不同方法选择最佳概念后得到的顶层概念。比较不同方法得出的概念，可以发现它们之间存在蕴含关系，如图8所示。一个概念

表6. 使用方法 M1、M2、M3 和 M4 得出的顶级互斥概念

概念划分单元	最小项		位级
M1：$\#partitions * \#minterms * powersum$			
c_1	$\{1, 5, 17\}$	$\{0, 2-7, 9, 15\}$	$\{1\}$
c_2	$\{28\}$	$\{0-2, 4, 8, 9, 11, 13\}$	$\{0\}$
c_3	$\{20, 28\}$	$\{1, 6-8, 14, 15\}$	$\{1\}$
M2：$\#partitions * powersum$			
c_4	$\{5, 20, 28\}$	$\{8\}$	$\{0, 4\}$
c_5	$\{5, 17, 20\}$	$\{14\}$	$\{0\}$
c_6	$\{1, 17, 28\}$	$\{1\}$	$\{0\}$
M3：$powersum$			
c_7	$\{5, 20, 28\}$	$\{8\}$	$\{0, 4\}$
c_8	$\{17\}$	$\{1, 14\}$	$\{0, 6\}$
c_9	$\{5, 20\}$	$\{14\}$	$\{0, 3\}$
M4：准确率			
c_{10}	$\{20\}$	$\{2, 4, 5, 9-14\}$	$\{5\}$
c_{11}	$\{17\}$	$\{1, 2, 4, 6-13, 15\}$	$\{2\}$
c_{12}	$\{5\}$	$\{2, 4-6, 10-14\}$	$\{3\}$
c_{13}	$\{1\}$	$\{0, 2-9, 14, 15\}$	$\{1\}$
c_{14}	$\{1, 20, 28\}$	$\{6\}$	$\{3\}$

蕴含另一个概念，当且仅当它们的最小项集合之间存在子集关系，且它们的划分单元集合之间存在超集关系。尽管概念 c_{13} 和 c_{14} 因其位级不同而互斥，但概念 c_{14} 蕴含概念 c_{13}，因为它们的最小项集合之间存在子集关系。

接下来，我们将考察表7中推导出的顶层概念在 $relpower$、精确率、召回率、准确率和支持度（所覆盖训练对象的数量）方面的属性。其中既有非常具体、精确率高而召回率低的概念（c_6，c_{14}），也有非常一般、召回率高而精确率低的概念（c_4，c_7，c_{10}，c_{13}）。由于选择最佳概念的方法不同，方法 M1 会将 $relpower$ 较高的概念集中在顶层，而方法 M4 则倾向于选择准确率较高的概念。最终，专家必须决定采用哪种方法。

聚焦于 M4 概念可以发现，概念 c_{10}，c_{11}，c_{12}，c_{14} 覆盖了所有本质划分单元格。同时，它们的准确率均不低于 70%，接近网络 74% 的准确率。其支持度为总计 356 个对象中的 305 个。这意味着，只需四个概念就足以近似网络的准确率。

现在我们来考察这些概念的逻辑解释。概念c_{14}只有一个最小项，非常简单：$r \wedge f \wedge m \wedge t$。图9展示了概念 c_{10} 和 c_{11}的逻辑树。由于这些树难以解释，我们考察它们通往1叶节点的路径，如表8所示。对于每条路径，我们列出精确率、召回率、准确率、所覆盖的最小项数（#mt），以及分别针对0对象（0-avg）和1对象（1-avg）的平均路径求值。其差异

表7。概念特征

概念 $relpower$	精确率	召回率	准确率	支持度	
c_1	**19%**	58%	26%	55%	**193**
c_2	11%	51%	54%	54%	79
c_3	8%	64%	26%	52%	112
c_4	4%	53%	98%	54%	175
c_5	4%	75%	50%	60%	158
c_6	4%	**100%**	1%	52%	209
c_7	4%	53%	98%	54%	175
c_8	2%	75%	73%	66%	62
c_9	3%	63%	84%	65%	96
c_{10}	0.4%	71%	**100%**	**72%**	33
c_{11}	4%	77%	83%	**72%**	62
c_{12}	1%	76%	55%	71%	63
c_{13}	1%	70%	97%	70%	68
c_{14}	7%	**100%**	9%	70%	180

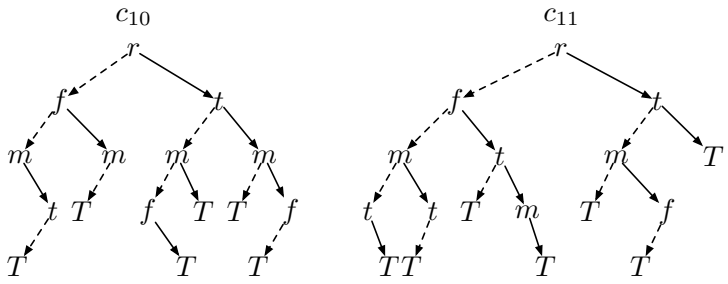


图9。 输血：概念 c_{10} 和 c_{11} 在关键单元上的逻辑树， T 代表输血类别，虚线表示对前一个属性的取反求值

表8。 首批概念逻辑树的叶路径及其特征；粗体数字表示高值

叶路径	精确率	召回率	准确率	#mt	0-平均	1-平均
c_{10}						
$\overline{r} \wedge \overline{f} \wedge m \wedge \overline{t}$	75%	65%	60%	1	0.049	0.047
$\overline{r} \wedge f \wedge \overline{m}$	**100%**	13%	39%	2	0.125	0.112
$r \wedge f \wedge \overline{m} \wedge \overline{t}$	71%	**100%**	**72%**	1	0.023	0.025
$r \wedge m \wedge \overline{t}$	83%	43%	54%	2	0.080	0.085
$r \wedge \overline{m} \wedge t$	**100%**	13%	39%	2	0.056	0.047
$r \wedge \overline{f} \wedge m \wedge t$	71%	**100%**	**72%**	1	0.040	0.030
c_{11}						
$\overline{r} \wedge \overline{f} \wedge \overline{m} \wedge t$	73%	71%	62%	1	0.025	0.023
$\overline{r} \wedge \overline{f} \wedge m \wedge \overline{t}$	72%	64%	59%	1	0.027	0.020
$\overline{r} \wedge f \wedge \overline{t}$	67%	**100%**	67%	2	0.074	0.050
$\overline{r} \wedge f \wedge m \wedge t$	**100%**	11%	40%	1	0.271	0.193
$r \wedge \overline{m} \wedge \overline{t}$	75%	73%	66%	2	0.065	0.131
$r \wedge \overline{f} \wedge m \wedge \overline{t}$	76%	69%	64%	1	0.034	0.054
$r \wedge t$	75%	71%	64%	4	0.336	0.382

0-avg 与 1-avg 之间的差异可视为区分能力的一种度量。值得注意的是， c_{10} 的第三条和第六条叶路径很有意思，因为它们的准确率为 72%，对于分类而言非常有效。

表9展示了表8所列叶路径之间的蕴含关系。这些蕴含关系与表8共同说明，向一个概念添加最小项会如何影响其特征。

表9. 概念 c_{10} 与 c_{11}的叶路径之间的蕴含关系

c_{10} 叶	蕴含 c_{11} 叶	
$\overline{r} \wedge \overline{f} \wedge m \wedge \overline{t}$	$\longleftrightarrow$	$\overline{r} \wedge \overline{f} \wedge m \wedge \overline{t}$
$r \wedge f \wedge \overline{m} \wedge \overline{t}$	$\longrightarrow$	$r \wedge \overline{m} \wedge \overline{t}$
$r \wedge m \wedge \overline{t}$	$\longleftarrow$	$r \wedge \overline{f} \wedge m \wedge \overline{t}$
$r \wedge \overline{m} \wedge t$	$\longrightarrow$	$r \wedge t$
$r \wedge \overline{f} \wedge m \wedge t$	$\longrightarrow$	$r \wedge t$

8 结论

我们通过在人工神经网络模型与逻辑表达式之间构建桥梁，并将这些表达式可视化为逻辑树（QLDT），解决了使用逻辑表达式解释人工神经网络模型的问题。因此，我们的解释方法将人工神经网络模型分解为逻辑树。

人工神经网络中的 ReLU 节点为网络引入非线性，这对于解决非线性分类任务至关重要。我们的方法利用 ReLU 条件对训练后的人工神经网络模型进行分区，从而为每个分区单元获得线性映射。这些线性映射的最小项权重被解释为位数。基于所得的三维比特张量，我们应用三元概念分析中的技术提取三元概念，并将其进一步解释为逻辑树。

我们的方法主要优势在于能够运用逻辑解释人工神经网络模型。逻辑为分析属性之间的交互提供了强大的工具集。

我们的方法依赖于将所有最小项取值用作网络输入。由于 n 个对象属性共有 2^n 个最小项，因此 n 被限制在相对较小的范围内。在未来的工作中，我们将努力使此方法适用于更复杂的神经网络。

参考文献

1. Balestriero, R.：神经决策树。arXiv 预印本 arXiv:1702.07360 (2017) 2. Balkenius, C., Gärdenfors, P.：神经网络中的非单调推理。载于：KR。第 32–39 页。Citeseer (1991) 3. Blutner, R.：非单调推理与神经网络。Synthese **142**，143–174 (2004) 4. Friedman, J.H., Popescu, B.E.：通过规则集成进行预测学习。应用统计年刊 **2**，916–954 (2008)，https://api.semanticscholar.org/CorpusID:16884658 5. Ganter, B., Wille, R.：应用格理论：形式概念分析。达姆施塔特工业大学数学系 (1997) 6. Gärdenfors, P.：逻辑如何从信息动力学中涌现。隆德大学 (1992) 7. beim Graben, P.：物理符号系统的不兼容实现。Mind and Matter **2**(2)，29–51 (2004) 8. Granmo, O.C., Glimsdal, S., Jiao, L., Goodwin, M., Omlin, C.W., Berge, G.T.：卷积 Tsetlin 机。arXiv 预印本 arXiv:1905.09688 (2019) 9. Kaur, D., Uslu, S., Rittichier, K.J., Durresi, A.：可信人工智能：综述。ACM计算调查 (CSUR) **55**(2)，1–38 (2022) 10. Kraus, M., Tschernutter, D., Weinzierl, S., Zschech, P.：可解释广义加性神经网络。欧洲运筹学杂志 (2023) 11. Lehmann, F., Wille, R.：形式概念分析的三元方法。载于：概念结构：应用、实现与理论：第三届国际概念结构会议，ICCS'95，美国加利福尼亚州圣克鲁斯，1995 年 8 月 14–18 日，论文集 3。第 32–43 页。施普林格 (1995) 12. Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.I.：从局部解释到全局理解：利用面向树模型的可解释人工智能。Nature Machine Intelligence **2**(1)，56–67 (2020) 13. Miller, T.：人工智能中的解释：来自社会科学的见解。Artificial Intelligence **267**，1–38 (2019) 14. Molnar, C.：可解释机器学习。Lulu.com (2020)

15. Murofushi, T., Soneda, S.：模糊测度的读取技术 (iii)：交互指数。载于：第九届模糊系统研讨会论文集，札幌。第 693–696 页 (1993) 16. Ribeiro, M.T., Singh, S., Guestrin, C.：「为什么要信任你？」解释任意分类器的预测。载于：第 22 届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集。第 1135–1144 页 (2016) 17. Roth, A.E.：夏普利值简介。载于《夏普利值》，第 1–27 页 (1988) 18. Schmitt, I.：Qql：一种数据库与信息检索查询语言。VLDB期刊 **17**(1)，39–56 (2008) 19. Schmitt, I.：将权重纳入基于量子逻辑的查询语言。载于《信息检索与决策的类量子模型》，第 129–143 页。施普林格 (2019) 20. Schmitt, I.：QLC：一种受量子逻辑启发的分类器。载于：Desai, B.C., Revesz, P.Z. (编)，IDEAS'22：国际数据库工程应用研讨会，匈牙利布达佩斯，2022 年 8 月 22–24 日。第 120–127 页。ACM (2022)。 https://doi.org/10.1145/3548785.3548790, https://doi.org/10 3548790 21 1145 3548785 . / . . Schmitt, I.：QLDT：一种基于量子逻辑的决策树。载于：Chiusano, S., Cerquitelli, T., Wrembel, R., Nørvåg, K., Catania, B., Vargas-Solar, G., Zumpano, E. (编)，数据库与信息系统新趋势——ADBIS 2022 短文、博士生论坛及研讨会：DOING、K-GALS、MADEISD、MegaData、SWODCH，意大利都灵，2022 年 9 月 5–8 日，论文集。计算机与信息科学通信，第 1652 卷，第 299–308 页。施普林格 (2022)。 https://doi.org/10.1007/978-3-031-15743-1_28, https://doi.org/ 10 1007 978 3 031 15743 1 _ 28 . / - - - - . Shapley, L.S., 等：n人博弈22的一个值。普林斯顿大学出版社，普林斯顿 (1953) 23. Suthaharan, S., Suthaharan, S.：决策树学习。载于《大数据分类的机器学习模型与算法：通过实例实现有效学习》，第 237–269 页 (2016) 24. Wei, L., Qian, T., Wan, Q., Qi, J.：三元概念分析研究综述。国际机器学习与控制论杂志 **9**，699–712 (2018) 25. Yang, Y., Morillo, I.G., Hospedales, T.M.：深度神经决策树。arXiv 预印本 arXiv:1806.06988 (2018)