用于函数可视化的可解释架构神经网络

Shengtong Zhang，西北大学工业工程与管理科学系；
Daniel W. Apley，西北大学工业工程与管理科学系

arXiv:2303.03393v1 [cs.LG] 3 Mar 2023

摘要

在许多科学研究领域，了解并可视化黑箱函数中所有输入变量的影响至关重要。现有的可视化工具无法同时展示所有输入变量的影响。虽然可以选取一个或两个输入变量，通过二维或三维图进行可视化，同时固定其他变量，但这会让人对模型产生过度简化且不完整的认识。为克服这一不足，我们提出一种新的可视化方法，利用可解释架构神经网络（IANN）直接、同步地展示所有输入变量的影响。我们提出了两种可解释结构，每种结构都可以通过特定的 IANN 方便地表示，并讨论了若干可能的扩展。我们还提供了一个 Python 软件包，用于实现所提出的方法。补充材料可在线获取。

关键词 ： 函数可视化；可解释机器学习；神经网络 k

1 引言

科学家和工程师常常希望将黑箱函数的输出可视化，以了解并解释输入变量对输出（即响应）变量的影响。例如，复杂计算机模拟的大多数代理模型都以黑箱特性明显、缺乏可解释性而闻名。将黑箱模型可视化有助于了解输入如何影响预测的响应变量、预测关系是否符合直觉，以及如何基于模型做出决策。

我们将黑箱函数记为 $f(\boldsymbol{x})$，其中 $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)$ 是一个 d 维输入变量向量，而 $f(\boldsymbol{x})$ 是标量响应。 $f(\boldsymbol{x})$ 作为 $\boldsymbol{x}$ 的函数，通常称为响应面。实际上，从大型数据集或复杂的计算机模拟模型中估计出的许多函数都包含大量输入变量，这使得可视化所有输入变量的联合效应颇具挑战。

在现有旨在解释黑箱模型的研究中，或许最常见且与本研究最密切相关的是“部分依赖”（PD）图（Friedman，2001年）和“个体条件期望”（ICE）图（Goldstein等人，2015年）。每张 PD 图通过对省略的输入变量的边际分布进行积分，展示一个或两个选定输入变量的边际效应；而每张 ICE 图则展示一组曲线，这些曲线是所关注的单个选定输入变量的函数，每条曲线对应省略的 $d-1$ 输入变量的一种固定组合。累积局部效应（ALE）图（Apley 和 Zhu，2020年）改进了 PD 图：当输入变量高度相关时，它的计算速度快得多，结果也更准确。与 ICE 图密切相关的格架图（Becker等人，1996年、1994年）由一系列图组成，展示在省略变量保持固定时，响应变量如何随一对选定输入变量变化；省略变量的每种固定组合都对应一张单独的图。尽管格架图通常能呈现清晰且相当完整的图景

对于 $f(\boldsymbol{x})$ 在 $d=3$ 情况下的情形， $d>3$ 格架图会变得难以处理，因为需要考虑的省略变量 $d-2$ 固定组合过多。PD、ICE、格架图和 ALE 图各自着重于可视化一个或两个变量的效应，并且在 d 较大时，无法清晰呈现所选输入变量与省略输入变量之间的交互作用；相比之下，我们的方法旨在可视化所有输入变量的联合效应。此外，Agarwal 等人（2021年）将深度神经网络与广义加性模型相结合，以提升可解释性。主要区别在于，为便于可视化，Agarwal 等人（2021年）中的神经网络架构被限制为如下形式的加性函数关系： $f(\boldsymbol{x}) = f_1(x_1) + f_2(x_2) + \cdots + f_d(x_d)$ ，因此无法用于表示和可视化输入变量之间的交互作用。

其他关联性较弱的研究则侧重于计算每个输入变量的变量重要性度量。Breiman（2001年）提出了随机森林模型中基于置换的特征重要性这一思想，随后该方法被扩展到一般的黑箱模型（Fisher 等，2019年；K¨onig 等，2021年）。基于博弈论概念，Lundberg 和 Lee（2017年）使用夏普利值计算特征重要性。然而，特征重要性度量只能为每个输入变量的重要性提供一个标量数值评分，却无法揭示这些变量如何影响响应变量。

一些方法不再单独考察原始变量/特征的影响，而是旨在可视化 f 的拓扑结构（$\boldsymbol{x}$）。这些方法主要关注识别 f 的局部极小值和极大值的数量及位置，随后识别输入空间中连接极小值与极大值的路径。Gerber 等人（2010年）使用 Morse-Smale 复形将 d 维输入变量空间划分为若干非重叠区域，在这些区域内 f 是单调的，并找出 f 的局部极小值和极大值。为了可视化 f 在各个划分区域中的拓扑结构，他们构建了特定的回归曲面，并将其嵌入二维空间，以简化表示各区域中的 f 。该方法应用于核反应堆安全分析

和可视化（Maljovec 等人，2013年）。Harvey 等人（2012年）进一步利用 Reeb图打散 Morse-Smale 复形产生的环，并提供拓扑结构更简单的可视化。尽管可以从 Morse-Smale 复形中每个分段对应的嵌入回归曲线解读输入变量的局部影响，但全局影响却难以解读，尤其是在分段数量增加时。此外，该方法并不适合可视化一组输入之间的全局交互作用。

f的可视化方法基于以下近似表示:

$$f(x_1, x_2, \cdots, x_d) \approx g(x_j, h(\boldsymbol{x}_{\setminus j})) \quad \text{for some } j \in \{1, 2, \cdots, d\}. \tag{1}$$

符号“$\boldsymbol{x}_{\setminus j}$”表示除 x_j 之外的所有输入变量。为简便起见，我们聚焦于定义在笛卡尔积 $\prod_{j=1} I_j$ 上的函数 f，其中每个 $I_j \in \mathbb{R}$ 都是一个闭区间。在第 2 节中，我们将证明，对于每个 $j \in \{1, 2, \cdots, d\}$，$[0,1]^d$ 上的任意连续 f，都可以用结构 (1) 任意精确地近似，其中的函数 g 和 h 均为连续函数（不过，对于某些 j，所得函数 g 和 h 的性质会优于其他 j 对应的函数）。我们还将介绍一种估计 g 和 h 的方法，以及一种选择索引 j 的方法，从而获得良好的近似。

要通过 (1) 可视化 f，并突出显示某个特定输入变量 x_j，可以绘制 f 随 x_j 和 h 变化的三维图。举例来说，我们来考虑物理学中的谐波函数

$$f(\boldsymbol{x}) = x_1 * \sin[(2\pi/x_2) * x_3 + x_4]. \tag{2}$$

这里 f 表示波上某一点的位移，$x_1 \in [0.5, 2]$ 表示振幅，$x_2 \in [0.5, 2]$ 表示波长，$x_3 \in [0, 1]$ 表示该点的位置（例如，距波源的距离），$x_4 \in [0, \pi]$ 表示波的相位。f 可表示为式（1），其中单独列出 x_1：$f(\boldsymbol{x}) = x_1 * \sin[h(\boldsymbol{x}_{\setminus 1})]$, 以及 $h(\boldsymbol{x}_{\setminus 1}) = (2\pi/x_2) * x_3 + x_4$。相应的三维可视化图（$f$ 与 x_1 和 h的关系）见图1(a)。这种图可以直观地展示 f 如何取决于单一变量 x_j，包括 x_j 如何与某个潜在函数（h）相互作用；该函数能够体现 x_j与其他 $d-1$ 输入变量交互作用的全部情况。

本文主要使用静态三维图来可视化估计得到的函数。然而，考虑到静态三维图可能因视觉错觉而容易被误读（Cleveland 和 McGill，1984年），我们建议用户采用自己偏好的可视化渲染方式来绘制函数，例如使用可交互旋转三维图的软件，以更好地避免误读或遗漏图中的特征；也可以在三维图旁附上带有等高线的二维热图，和/或使用高级阴影效果与透视视角。图1中，我们还展示了同一函数的二维热图，以及使用软件 *rayshader* 制作的阴影透视图（Morgan-Wall，2022年）。

为了理解 x_j 如何与其他单个输入变量相互作用，并更广泛地理解所有 d 输入变量对 f的联合效应，我们的方法采用层次化方式，并用一种类似于式 (1) 的结构来近似 h，其中突出显示第二个输入变量，依此类推。我们在第 3.1 节介绍这一方法，并将其称为“原始变量层次”（OVH）结构。此外，我们还考虑一种更一般的结构，该结构假设层次化 h 函数是原始输入的某些线性组合的函数，这可以进一步增强可视化效果。我们将其称为“不相交活跃子空间层次”（DASH）结构，并在第 3.2 节加以介绍。

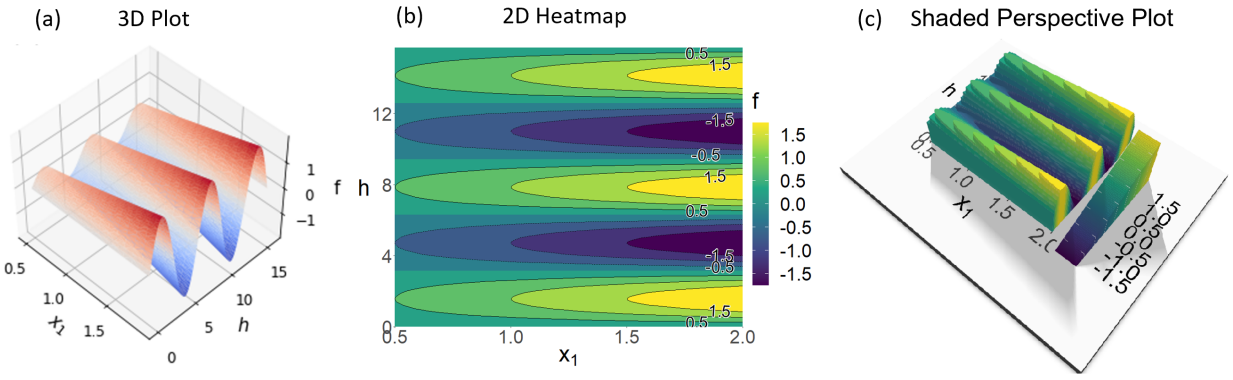


图1：谐波函数 f按结构（1）进行可视化的示例，包括：（a）三维图；（b）同一函数的二维热图；以及（c）阴影透视图。彩色图请参见在线版本。

本文其余部分安排如下。第 2 节给出一个定理，说明任意连续函数 f 都可以由具有更易解释结构 (1) 的模型任意逼近；此外，我们还介绍用于估计式 (1) 中函数 g 和 h 的可解释架构神经网络（*IANN*）模型。第 3 节构建 OVH 和 DASH 结构，通过对潜在函数 h进行层次分解来可视化并解释所有变量的联合效应；这两种结构都可以用特定的 IANN 模型方便地表示。需要指出的是，对于评估成本高昂的复杂黑箱模拟响应面，应首先根据模拟数据拟合代理模型（Sacks等人，1989年；Kennedy 和 O'Hagan，2000年，2001年），然后将代理模型用作 f。第 4 节介绍用于确定层次分解中输入变量适当排序的算法。第5节提供更多数值示例，以展示如何使用 IANN 方法进行函数可视化。第6节讨论 IANN 方法的一些潜在扩展。

2 IA第一层次的神经网络结构

本节介绍如何使用 IANN 模型通过结构 (1) 近似 f，以及该模型如何促进对 f 的可视化。在第 2.1 节中，我们提出定理 2.1，以说明对于某些连续函数 g 和 h，所提出的结构 (1) 可以在 $[0,1]^d$ 上近似任意连续函数。为了估计 g 和 h，我们在第 2.2 节中引入IANN架构，以便针对每个单独选取的变量 x_j 近似 f。在第2.3节中，我们比较了部分依赖图、个体条件期望图和 IANN可视化图，并建立了它们之间的联系，以说明后者具有能够更清晰地解释 f 的优势。如果针对每个输入以非层次方式重复这一分解，则可以将其作为一种独立方法，用于可视化每个输入的影响（如图1所示），即针对每个输入 x_j，对于 $j = 1, 2, \cdots, d$。这种可视化中缺失的主要信息是

所有输入联合效应的解耦，这正是第3节的主题。为此，本节中的分解提供了基本构件。

2.1 IANN近似定理

以下定理（证明见附录 D）保证，我们在（1）中提出的可解释结构能够近似任意定义在 $[0,1]^d$ 上的连续函数 f。

定理 2.1。设 $f:\ [0,1]^d \to \mathbb{R}$ 是连续函数。对于任意 $\epsilon > 0$ 和 $j \in\{1,2,\cdots,d\}$，存在连续函数 g 和 h，满足：

$$\left|f(\boldsymbol{x}) - g(x_j, h(\boldsymbol{x}_{\setminus j}))\right| < \epsilon, \quad \forall \boldsymbol{x} = (x_j, \boldsymbol{x}_{\setminus j}) \in [0,1]^d. \tag{3}$$

更一般地，如果 f 定义在各个 x_j 中区间的笛卡尔积上，则上述定理仍然成立。

该定理的重要意义在于，结构（1）可用于以任意高的精度近似连续函数 f，进而利用这种近似来可视化 f，如图1所示。这种可视化与个体条件期望图有关，但能提供更清晰的呈现，具体将在第 2.3 节讨论。此外，由于 h 是连续的，我们可以再次应用定理 2.1，证明 h 可以用同一结构（1）近似得十分精确；这一过程还可以重复进行，从而得到第3节所述的层次分解。

请注意，尽管有一项需要指出的注意事项，定理 2.1 适用于每个 $j \in \{1,2,\cdots,d\}$。对于某些 j，相应的 h 可能极其复杂，以至于无法在后续层级中进行实际估计和可视化。然而，对于我们所考虑的所有实函数 f，我们都能针对至少一个输入变量 x_j，利用一个性质良好的 h 对 f 进行准确近似，并且随后可在下一层级通过相同的结构 (1) 对其进行近似。此外，我们的算法（见第 4 节）会自动找到输入变量的良好层次排序，从而得到一个

对 f 的准确近似，并且该近似的准确性可以轻松量化，以检验所讨论的函数 f 是否确实能够由 IANN结构近似。

2.2 IANN架构

为估计函数 g 和 h，我们采用图2所示的定制神经网络架构，并将其称为 IANN。

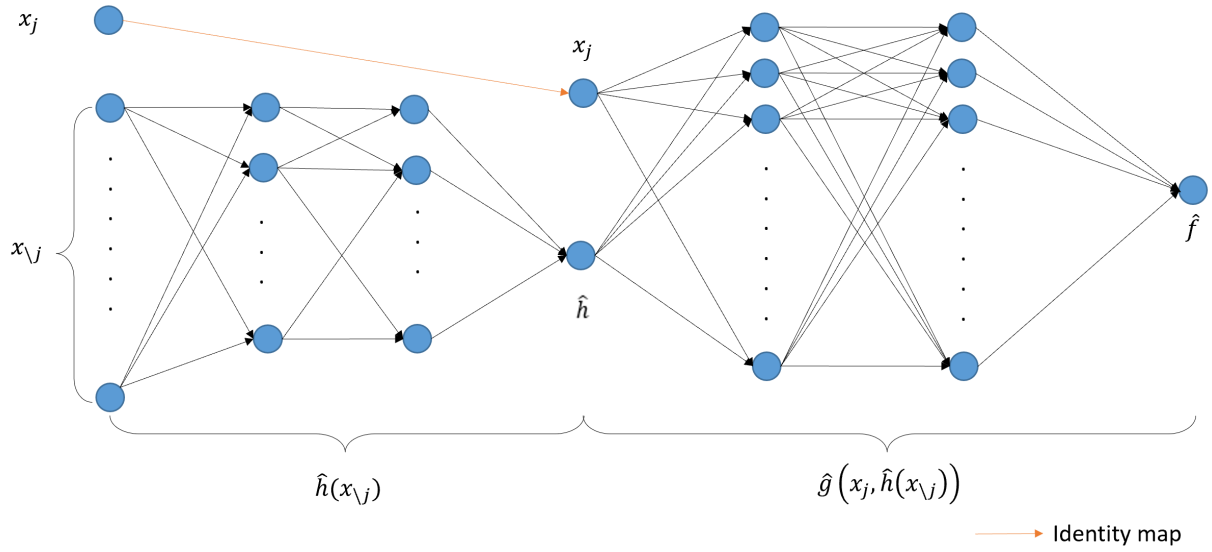


图2：在顶层层次上表示公式（1）的 IANN架构。中间的瓶颈层仅由两个节点组成，分别表示 x_j 和 h（$\boldsymbol{x}_{\setminus j}$）。

输入层包含 d 个节点，用于表示 d 输入变量；输出层中的单个节点则是原函数 f（$\boldsymbol{x}$）的近似。根据定理 2.1以及神经网络的通用逼近定理（Hornik等，1989年；Cybenko，1989年；Hornik，1991年），只要具有足够多的层和节点，图2中的 IANN就能够通过结构（1）近似任意连续函数 f，其中函数 g 和 h 是在拟合 IANN模型的过程中估计得到的。

ˆ 中间的瓶颈层只有两个节点，分别表示 x_j 和 h。 x_j 是

ˆ 通过恒等映射直接连接到第一个输入节点，而 h 是式 (1) 中潜在函数 $h(\boldsymbol{x}_{\setminus j})$ 的估计值。在瓶颈层左侧的各层中，我们通过一个记作 $h(\boldsymbol{x}_{\setminus j})$ 的全连接神经网络，将表示其余输入变量 $\boldsymbol{x}_{\setminus j}$ 的 $d-1$ 个节点连接到节点 hˆ。在瓶颈层右侧的各层中，我们使用另一个全连接神经网络（记作 ˆg，其输入为 x_jˆ和 h），将函数 g 表示为 $f(\boldsymbol{x})$ 的近似。

将训练数据中的输入/响应观测值记为 $\{(x_1^i,\ x_2^i,\ \cdots,\ x_d^i,\ f^i)\ \ 1 \leq i \leq N\}$，将 IANN 的响应预测记为 ˆ $f^i = \hat{g}(x_j^i,\ \hat{h}(\boldsymbol{x}_{\setminus j}^i))$，其中 N是训练集中的观测值数量。训练数据采用附录 B 中所述的定制拉丁超立方抽样生成。请注意，即使 f 表示一个评估成本高昂的仿真响应面，N 也可以取较大的值，因为首先会用拟合于模拟数据的代理模型替代 f，而从代理模型生成响应观测值的成本很低。为使 IANN 拟合训练数据（即估计权重和偏置），我们采用标准平方误差损失:

$$Loss = \frac{1}{N}\sum_{i=1}^{N}\left|f^i - \hat{f}^i\right|^2. \tag{4}$$

拟合 IANN 后，可以通过绘制 ˆgˆ 关于 x_j 和 h的函数图，对 f 的近似进行可视化，如图1所示。拟合后的 IANN 可作为 f的近似。为评估该近似是否充分，我们在输入空间中生成一个大型随机样本作为测试集，并通过比较 IANN 的 ˆ 测试预测值 f 与测试输入处的实际函数值 f，计算测试 r^2 。

回到式（2）中定义的谐波函数，我们（以非层次化方式）依次单独选取四个输入变量中的每一个作为 x_j，并拟合图2中的 IANN架构。由此得到的四幅三维图（分别展示四种不同 x_j下，g（x_j，h）作为x_j 和 h 的函数）见图3。左上角的可视化图具有最高的测试 r^2 （99.8%），这表明单独选取振幅变量 x_1后，谐波函数可以由结构（1）很好地近似。

这与式（2）中 f 的结构一致（实际应用中，它通常会是一个更复杂、也更难直接解释的函数，例如拟合某个复杂计算机模拟输出的代理模型），因为式（2）可以写成 f（$\boldsymbol{x}$）$= x_1 *$ 正弦 $\left[h\ (\boldsymbol{x}_{\backslash 1})\ \right]$，其中 $h\ (\boldsymbol{x}_{\backslash 1})\ =\ (2\pi/x_2)\ * x_3 + x_4$。从左上图可以清楚看出，对于任意固定的$\boldsymbol{x}_{\backslash 1}$，$x_1$ 的影响都是线性的；而 x_1 的影响则强烈依赖于$\boldsymbol{x}_{\backslash 1}$的取值。例如，对于某些 $h\ (\boldsymbol{x}_{\backslash 1})$，$x_1$ 的影响为正（线性趋势的斜率为正）；而在其他取值下，其影响为负。此外，斜率随 $h\ (\boldsymbol{x}_{\backslash 1})$ 呈正弦函数般的周期性变化。显然，$h\ (\boldsymbol{x}_{\backslash 1})$ 的IANN表示能够很好地捕捉实际函数 $(2\pi/x_2)\ * x_3 + x_4$，这一点从稍后第5.2节介绍的层次化IANN可视化中看得更清楚。图3左上图还展示了该方法如何用于理解输入之间的交互，因为 x_1 与 $h\ (\boldsymbol{x}_{\backslash 1})$ 显然存在很强的交互作用：随着 $h\ (\boldsymbol{x}_{\backslash 1})$ 变化，x_1 的影响会由正转负，反之亦然。

对于另外三幅图，测试 r^2 值相对较低，这表明在 IANN 中单独考察这些变量无法很好地近似结构 (1)。因此，这些图的可靠性较低，在解释输入的影响时应谨慎使用。在这种情况下，我们建议仅在顶层使用 $g(x_1, h(\boldsymbol{x}_{\backslash 1}))$的左上图（测试$r^2 = 99.8\%$），然后使用第3节中的层次分解来可视化 $\{x_2, x_3, x_4\}$的影响。我们在第5.2节中通过延续此示例来说明这一做法。如定理 2.1 之后所指出的，对于某个特定的 x_j，潜在函数 h 可能过于复杂，无法通过层数和节点数有限的神经网络进行估计（而且在后续层次中也难以直接可视化）；这一点会从其对应的测试 r^2值较低体现出来。但对于我们考察过的所有实际示例，至少有一个 x_j 对应的 $g(x_j, h(\boldsymbol{x}_{\backslash j}))$三维可视化图具有足够高的测试 r^2 值，因此可视化结果是可靠的。如果多幅图的测试 r^2值都较高，用户可以依靠其中任意一幅或全部图来解释输入的影响。

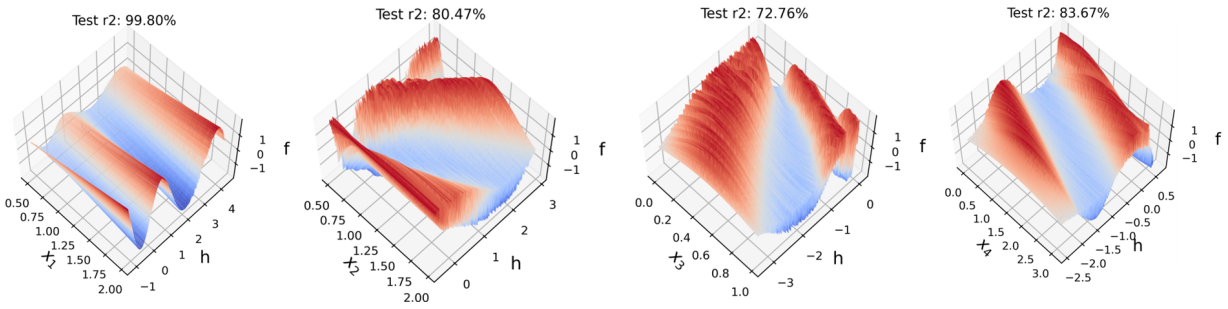


图3：式（2）中谐波函数的顶层（非层次化）可视化图，每次分别突出显示一个输入。每幅图上方显示测试 r^2 ，用于评估该图所表示的 IANN近似的准确性。

2.3 与 PD 和 ICE 图的联系

可视化输入变量影响的最常用方法是部分依赖图（PD 图），该方法最早由 Friedman（2001年）提出。随后，Goldstein等人（2015年）提出了个体条件期望图（ICE 图），通过针对被省略输入变量的每种固定组合展示一组曲线，改进了部分依赖图。下面我们使用谐波函数进行说明，并比较 IANN、部分依赖图和个体条件期望图之间的联系。

图4(a)使用开源Python软件包*scikit-learn* （Pedregosa等人，2011；Buitinck等人，2013）展示了变量 x_1 的部分依赖图和个体条件期望图。请注意，以虚线表示的部分依赖图是个体条件期望图中所有曲线的平均值。我们的IANN图与累积局部效应图之间并没有紧密、直接的联系。二者唯一的联系是IANN图与部分依赖图之间的联系；同时，累积局部效应图旨在生成类似于部分依赖图的结果，只是计算效率高得多，并且当输入高度相关时，不易出现问题。由于谐波波示例中的输入几乎不相关，累积局部效应图（为简洁起见未展示）与图4(a)中的部分依赖图非常相似。此示例的个体条件期望图都是斜率不同的直线，这正确地表明，对于固定的$\boldsymbol{x}_{\setminus 1}$， x_1 的效应是线性的，并且 x_1 与

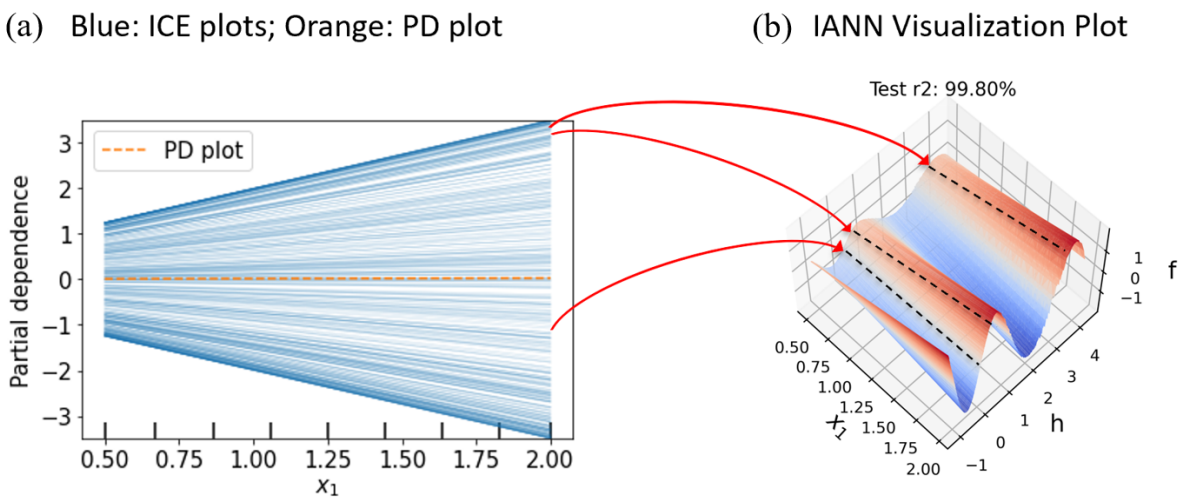


图4: （a）现有方法（个体条件期望图和部分依赖图）与（b）我们提出的 IANN 在可视化谐波函数中振幅变量 x_1 的影响方面的联系。IANN可视化可以理解为将个体条件期望图中的单条曲线平滑地拼接起来，从而形成更易于理解的三维曲面。（b）中的三条虚线曲线分别对应（a）中由连接箭头标出的三条个体条件期望图曲线。

其他变量存在交互（因为斜率各不相同）。相比之下，图4(b)展示了同一示例中 g（x_1, h（$\boldsymbol{x}_{\setminus 1}$））的IANN 图。请注意，个体条件期望图中的每条单独曲线都表示在$\boldsymbol{x}_{\setminus 1}$固定时，$f$ 作为 x_1 的函数。IANN 图可以看作是将所有个体条件期望图曲线平滑地拼接起来，从而形成一个三维曲面；该曲面为 x_1 对 f 的影响，以及 x_1 如何与其余变量$\boldsymbol{x}_{\setminus 1}$的某个函数（$h$）相互作用，提供了一种结构更清晰、可解释性更强的可视化。在图4(b)中，虚线曲线展示了 h（$\boldsymbol{x}_{\setminus 1}$）取三个固定值时对应的三条单独曲线；图4(a)中的箭头则标出了这三个$\boldsymbol{x}_{\setminus 1}$ 值对应的个体条件期望图曲线。与二维个体条件期望图可视化相比，此例中的三维IANN可视化更清晰地呈现了函数 f 。

3 用于可视化所有输入联合效应的层次化 IANN

除了用于独立可视化所选输入x_j的效应之外，第 2 节中介绍的 IANN结构还可以采用层次化方式，类似地对 h进行分解和可视化，依此类推。我们提出了两个版本的层次分解，每个版本都旨在可视化所有 d 输入的联合效应。在第 3.1 节中，我们介绍原始变量层次化（OVH）结构；在第 3.2 节中，我们介绍不相交主动子空间层次化（DASH）结构。从某种意义上说，该结构是 OVH结构的推广，当所涉及的各种函数是输入的某些线性组合的函数时，可以简化可视化。

3.1 原始变量层次化（OVH）结构

在结构（1）中， h 是$\boldsymbol{x}_{\setminus j}$中的所有输入的函数。为了可视化 h 如何依赖于$\boldsymbol{x}_{\setminus j}$，我们可以通过以下方式对函数进行层次分解（在 g 和 h 上添加下标，以表示层次级别）：

$$
\begin{aligned}
f(\boldsymbol{x}) &\approx g_1(x_{j_1}, h_1(\boldsymbol{x}_{\setminus j_1})) \\
h_{i-1}(\boldsymbol{x}_{\setminus (j_1,\cdots,j_{i-1})}) &\approx g_i(x_{j_i}, h_i(\boldsymbol{x}_{\setminus (j_1,\cdots,j_i)})), \quad i = 2, \cdots, d-1,
\end{aligned} \tag{5}
$$

其中 $h_{d-1}(x_{j_d}) \overset{\text{定义}}{=} x_{j_d}$。这里， $\{j_1, j_2, \cdots, j_d\}$ 表示输入索引 $\{1, 2, \cdots, d\}$ 的排列顺序，该顺序将在拟合层次化 IANN 之前作为预处理步骤确定。确定输入排序的算法将在第 4 节中介绍。与记号 $\boldsymbol{x}_{\setminus j}$ 类似，记号 $\boldsymbol{x}_{\setminus (j_1,\cdots,j_i)}$ 表示除 $\{x_{j_1}, \cdots, x_{j_i}\}$ 之外的输入变量。每个函数 g_i 都是函数 h_{i-1} 的近似，该函数有两个输入：x_{j_i} 和 $h_i(\boldsymbol{x}_{\setminus (j_1,\cdots,j_i)})$。每个函数 h_i 都取决于前几次迭代中所选输入的顺序 $\{j_1, j_2, \cdots, j_i\}$，但为简洁起见，我们在记号中省略了这一点

。由于定理 2.1 中的函数 h 也是连续的，因此我们可以在每一层反复对函数 h_{i-1} 应用定理 2.1，以证明层次分解 (5) 的合理性。

我们将式 (5) 中的第一个方程称为第 *1* 层表示，将其对应的三维图或二维热图（$f(\boldsymbol{x})$ 关于 x_{j_1} 和 $h_1(\boldsymbol{x}_{\setminus j_1})$ 的函数图）称为第 *1* 层图。类似地，我们将式 (5) 中的 i^{th} 方程称为第 i 层表示，将其对应的三维图或二维热图（h_{i-1}关于 x_{j_i}和 h_i 的函数图）称为第 i 层图，其中 $i = 2, \cdots, d-1$。这种方法总共生成 $d-1$ 个层次可视化图，可用于整体理解所有 d 个输入的联合效应。

为了同时估计所有 h_i 函数，我们采用具有多个瓶颈层的 IANN架构，如图 5 所示，其中展示的是输入变量数为 $d = 5$ 的情形。虚线箭头表示恒等映射。为简化记号，我们省略 IANN架构中 h_i上的帽号。在图 5 中，IANN架构总共包含 5 个输入变量和 4 个层级。第 4 层只是一个全连接神经网络，它有两个输入变量，输出表示函数$h_3(x_{j_4}, x_{j_5})$ 的估计值。同样，第 3 层有一个瓶颈输入层，包含两个输入（x_{j_3} 和h_3），输出为 h_2；第 2 层有一个瓶颈输入层，包含两个输入（x_{j_2}和 h_2），输出为 h_1；第 1 层的两个瓶颈输入为 x_{j_1} 和 h_1，输出为估计/近似得到的 f。

与公式 (4) 中定义的损失函数类似，我们使用 L_2 损失来最小化 IANN 输出 f 与原始函数 f之间的均方误差。通过同时估计原始函数 $f(\boldsymbol{x})$ 和所有潜函数 h_i，当 d 较大时，误差不易在不同层级之间累积。

为说明 OVH IANN 方法，考虑以下示例:

$$f(\boldsymbol{x}) = (5x_1 + x_2 + x_3 + x_4 + x_5 - 4.5)^2, \qquad \boldsymbol{x} \in [0,1]^5. \tag{6}$$

在这种情况下，满足结构 (5) 的输入排序 $\{j_1, j_2, \cdots, j_5\}$ 并不唯一。事实上，输入变量的任何排列都满足 (5)。例如，如果我们选择

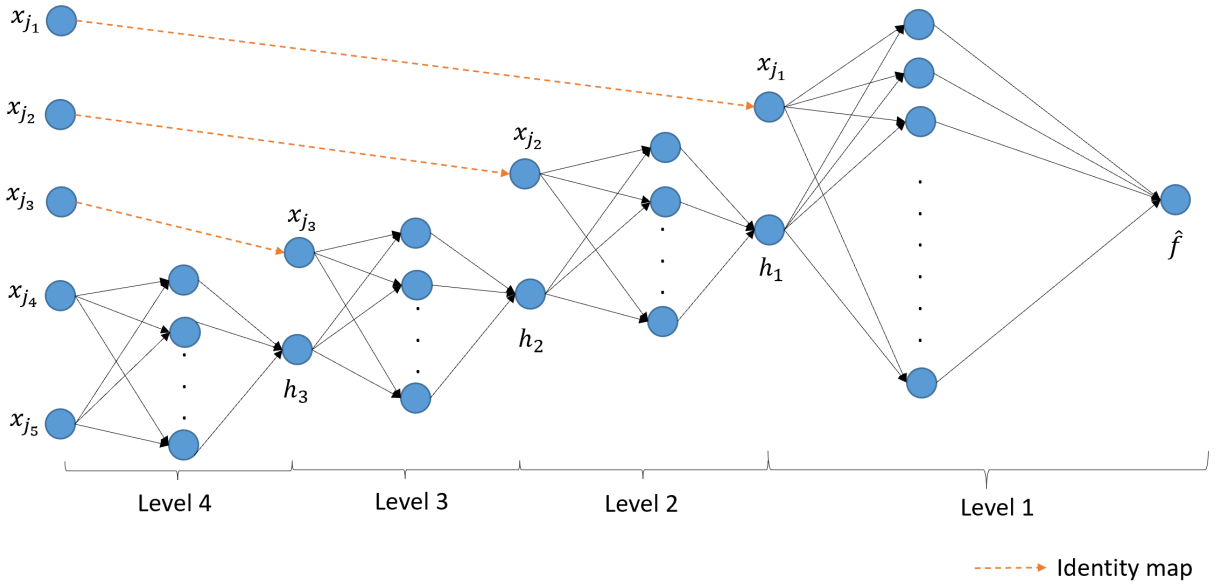


图 5：针对含有 $d = 5$ 个输入变量的 OVH结构的 IANN架构。

$j_1 = 2$，则响应可以表示为：

$$f(\boldsymbol{x}) = g_1(x_2, h_1(\boldsymbol{x}_{\setminus 2})) = (x_2 + h_1(\boldsymbol{x}_{\setminus 2}))^2, \tag{7}$$

其中 $h_1(\boldsymbol{x}_{\setminus 2}) = 5x_1 + x_3 + x_4 + x_5 - 4.5$。按照相同的逻辑，无论选择哪个剩余输入作为 x_{j_2}，h_1 都可以类似地进行分解，依此类推。请注意，如果将五个输入中的任意一个选作 x_{j_1}，上述论证仍然成立。下文采用排序 $(j_1, j_2, j_3, j_4, j_5) = (1, 5, 4, 2, 3)$，该排序由第 4 节所述的算法确定。三维可视化图如图6所示。测试 r^2 为 99。97%，表明 OVH结构 (5) 能够很好地近似 f。

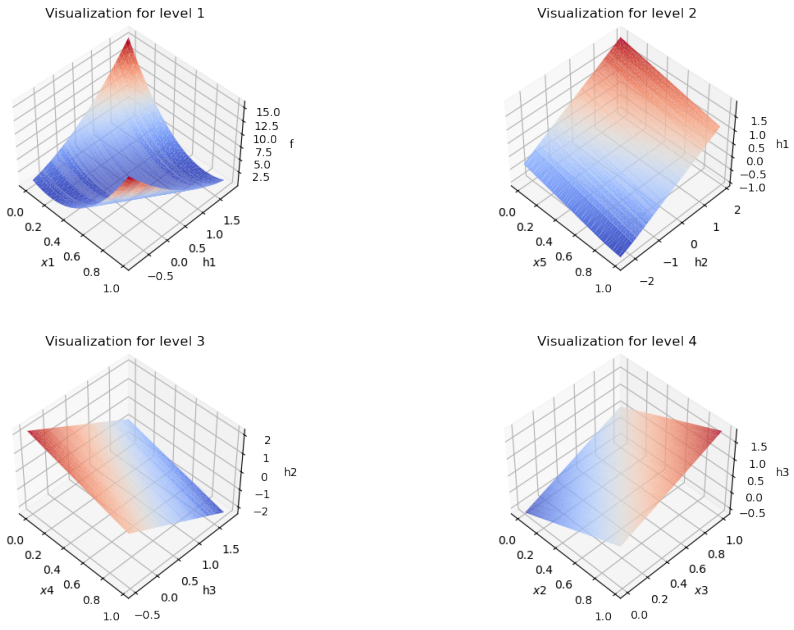


图6：示例 f （见公式（6））的 IANN可视化图。应按照层级的数值顺序，从左上角到右下角阅读这些图。

以下是从图6中的 IANN可视化图 得出的重要结论。从第 1 层图可以看出，当固定 h_1 时，函数 $f(\boldsymbol{x})$ 关于 x_1 呈二次形式，这与真实函数 $f(\boldsymbol{x}) = (5x_1 - h_1)^2$ 一致，如果取 $h_1 = -(x_2 + x_3 + x_4 + x_5 - 4.5)$。式 (7) 中的函数 g_1 和 h_1 当然并非唯一，因为可以将加性常数和/或乘性常数并入 h_1 ，并相应地修改 g_1 ，而不改变函数 $g_1(x_1, h_1(\boldsymbol{x}_{\setminus 1}))$。然而，这并不会改变对输入变量联合效应的解释。进一步就第 1 层图的解释而言，函数 f 在 x_1 处达到最小值，而该取值会随着 h_1 的增加而线性增加。

为了理解其他输入 $\boldsymbol{x}_{\setminus 1}$ 对 f的影响，我们必须从图6的图中辨别两点：(i) h_1 对 f 的影响，以及 (ii) $\boldsymbol{x}_{\setminus 1}$ 中的变量对 h_1的影响。关于前者，从第 1 层图可以看出，对于固定的 x_1 ，函数 f 关于 h_1 $(\boldsymbol{x}_{\setminus 1})$ 大致呈二次关系。为了理解 $\boldsymbol{x}_{\setminus 1}$的影响

对 h_1的影响，必须查看后续层级的图。从图6的第2层、第3层和第4层图可以看出，每个函数 h_i 关于其两个参数 $x_{j_{i+1}}$ 和 h_{i+1}大致呈线性关系，这意味着 h_1 关于 $\boldsymbol{x}_{\setminus 1}$呈线性关系。

为了进一步说明 IANN 层级图如何帮助理解 f 的行为，假设我们想选出使 f 最大化的 $\boldsymbol{x}$ 值。根据第 1 层图，最大值出现在两个角点 $(x_1, h_1) = (0, 1.5)$ 或 $(1, -1)$ 处。若聚焦于前者，根据第二层图，$h_1 = 1.5$ 在 $(x_5, h_2) = (0, 2)$ 处取得最大值。接着，根据第 3 级图，$h_2 = 2$ 在 $(x_4, h_3) = (0, -0.5)$ 处取得最大值。最后，根据第 4 级图，$h_3 = -0.5$ 在 $(x_2, x_3) = (0, 0)$ 处取得最大值。因此，f 在 $(x_1, h_1) = (0, 1.5)$ 处的最大值对应于 $x_1 = x_2 = \cdots = x_5 = 0$。按照相同的步骤考察另一个最大值 $(x_1, h_1) = (1, -1)$，我们发现它对应于 $x_1 = x_2 = \cdots = x_5 = 1$。这两种情况都对应于式 (6) 的真实最大值。

此外，由于从第 2、3 和 4 层级的图中可以看出，$h_1(\boldsymbol{x}_{\setminus 1})$ 只是其余变量 x_2、x_3、x_4、x_5的线性函数，并且 $f(\boldsymbol{x})$ 关于 $h_1(\boldsymbol{x}_{\setminus 1})$ 和 x_1 是二次的，因此我们可以得出结论：f 关于每个变量都是二次的。

3.2 不相交输入变量分层（DASH）结构

在本节中，我们提出一种可视为 OVH结构推广的替代分层结构。DASH结构假设分层函数 h_i's 是以下输入变量的某些不相交线性组合的函数：

$$
\begin{aligned}
f(\boldsymbol{x}) &\approx g_1(v_1, h_1(\boldsymbol{v}_{\setminus 1})), \\
h_{i-1}(\boldsymbol{v}_{\setminus (1,\cdots,i-1)}) &\approx g_i(v_i, h_i(\boldsymbol{v}_{\setminus (1,\cdots,i)})), \qquad i = 2, \cdots, p-1, \\
&\text{where} \quad v_i = \boldsymbol{\beta}_i^T \boldsymbol{x}_{J_i}, \quad i = 1, 2, \cdots, p \le d.
\end{aligned} \tag{8}
$$

p这里的 $\{J_1, \cdots, J_p\}$ 是互不相交的指标集，即 $J_i \bigcap J_j = \emptyset, \quad \forall 1 \le i \ne j \le p$、$\bigcup J_i =_{i=1}\{1, 2, \cdots, d\}$，而$\boldsymbol{x}_{J_i}$表示索引属于 J_i的输入变量。注意，DASH

结构（8）在 $p = d$时会退化为OVH结构。

定理 2.1 同样适用于结构 (8)：将 x 替换为 v 即可，因为输入的不相交线性组合（ v）取值于闭区间。与 OVH 结构的证明类似，可以对 f 和 h_i 反复应用该定理，从而证明，对于某些连续函数 g_i 和 h_i，结构 (8) 可以任意精确地近似定义在 $[0, 1]^d$ 上的任意连续函数 f。我们在附录A中描述的算法会自动确定不相交线性组合的数量 p、(8) 中的 v，以及 p 个不相交线性组合的排列顺序（v_1，v_2，$\cdots$，v_p），使得 (8) 能够良好地近似 f。

与 OVH 结构的处理方式类似，我们可以绘制一系列三维图，将 (8) 中的 f 可视化为 v 的函数；注意，v 易于理解为输入的函数，因为它们是输入的不相交线性组合。更具体地说，在第 1 层图中，我们将 f 可视化为 v_{j_1} 和 h_1 的函数，然后在 i^{th} 层进一步将 h_{i-1} 可视化为 v_{j_i} 和 h_i 的函数，其中 $i = 2, 3, \cdots, p-1$。对于我们考察过的许多实际示例，取 $p << d$ 的式 (8) 都能很好地近似 f。这样做的优点是，我们将分层三维可视化图的数量从 $d - 1$ 减少到 $p - 1$，从而在 $p << d$ 时大大简化了解释。

为了用 IANN架构 表示 (8)，我们在 OVH结构 的第一层前添加一个线性层，以学习 p 不相交线性组合 (v_1，v_2，$\cdots$，v_p) 及其底层的不相交输入组；图 7 展示了 $p = 5$ 的情况。每个线性组合 $v_i = \boldsymbol{\beta}_i^T \boldsymbol{x}_{J_i}$ 在该层中由一个单独的节点表示，该节点没有偏置，并使用线性激活函数。(8) 中的系数 $\boldsymbol{\beta}_i$ 是这个线性层中的估计权重。其余架构与图 5 所示用于 OVH结构 的 IANN 相同，只是后续各层的输入是不相交线性组合。为便于可视化，我们在 h_{i-1} 关于 v_{j_i} 和 h_i 的三维可视化图下方显示系数 $\boldsymbol{\beta}_i$。

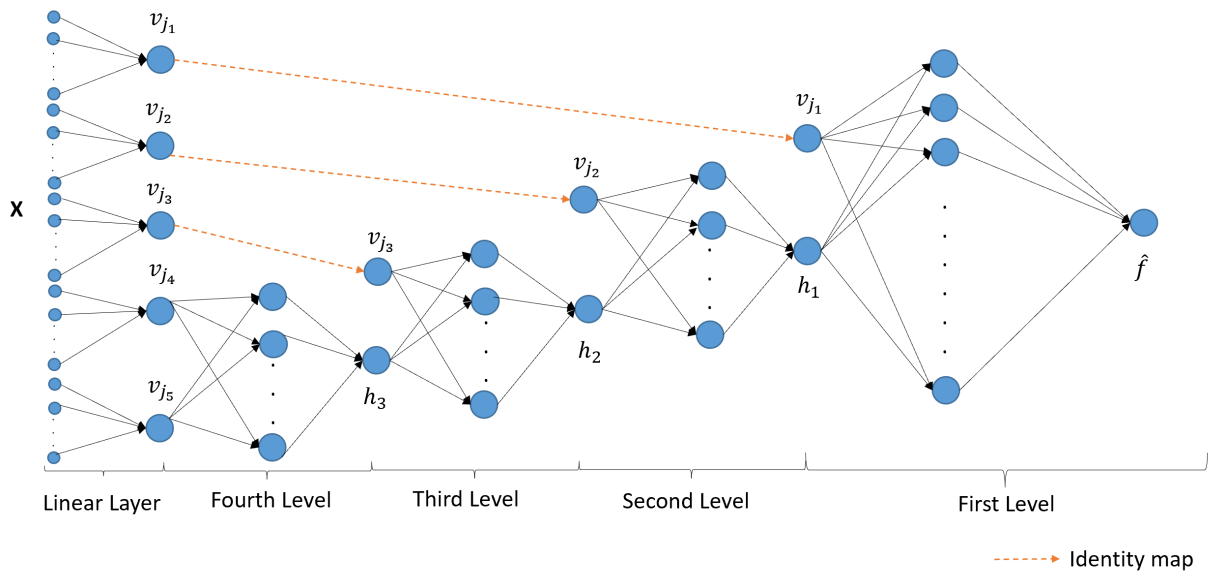


图 7: 采用 $p =$ 的 DASH结构 IANN架构示意图 5.

例如，考虑以下函数：

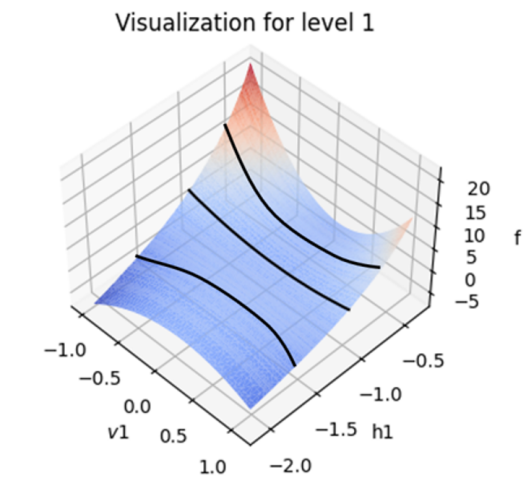
$$f(\boldsymbol{x}) = \left[7\exp\left(-4(1.5x_1 + x_2 - 2x_3)^2\right) + 2x_4 - 1.5x_5 + 0.7x_6 - 1.5\right](x_7 - 1.5x_8 + 0.7x_9 - 0.3)^2, \tag{9}$$

对于 $\boldsymbol{x} \in [0,1]^9$，可表示为式 (8)，其中 $p = 3$ 和

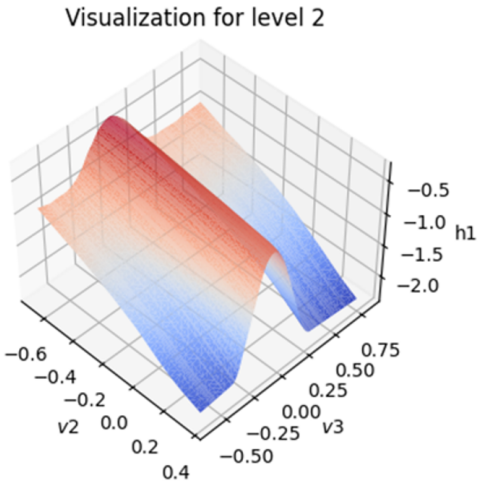
$$\begin{aligned} v_1 &= x_7 - 1.5x_8 + 0.7x_9, \\ v_2 &= 2x_4 - 1.5x_5 + 0.7x_6, \\ v_3 &= 1.5x_1 + x_2 - 2x_3, \quad \text{and} \\ f(\boldsymbol{x}) &= (v_1 - 0.3)^2 * h_1(v_2, v_3), \end{aligned} \tag{10}$$

其中 $h_1(v_2, v_3) = [7\exp(-4v_3^2) + v_2 - 1.5]$。我们将 DASH结构 IANN 拟合到该函数生成的数据上（三个不相交线性组合由附录A中所述的算法自动确定，而非预先指定），结果如图8所示。测试 r^2 为 99。98%，表明 DASH结构 (8) 对 f 提供了良好的近似。

与使用八个三维图来可视化和解释函数 $f(\boldsymbol{x})$ 不同，如同



$\{\beta_{1,7}, \beta_{1,8}, \beta_{1,9}\}$ = {0.667, -1.001, 0.467}



$\{\beta_{2,4}, \beta_{2,5}, \beta_{2,6}\}$ = {-0.506, 0.379, -0.176}
$\{\beta_{3,1}, \beta_{3,2}, \beta_{3,3}\}$ = {0.548, 0.365, -0.730}

图8：式（9）中 f 的 DASH结构 IANN可视化。左图中的实线曲线表示固定 $h_1 = -1.5$, -1.0 和 -0.5 时 f 随 v_1 的变化。

OVH结构一样，用八个三维图来可视化并解释函数 $\beta_i()$；而使用图8中的DASH结构时，我们只需两个三维图，就能可视化所有输入变量的影响。我们在每个IANN可视化图下方展示各线性组合中的输入系数 $\beta_{i,j}\, x_j \;\; v_i$ 。从图8中可以看出，每个 $\boldsymbol{\beta}_i$ 中的估计系数比值都与式（10）定义的真实比值相近。例如，真实的 $v_1 = x_7 - 1.5x_8 + 0.7x_9$，而拟合 IANN 得到的估计$v_1$ 为 $v_1 = 0.667x_7 - 1.001x_8 + 0.467x_9$，除常数乘法因子外，两者几乎完全相同。

关于如何解读这些图，首先，从图8的上方图中可以看出，当 h_1 固定时，函数 $f(\boldsymbol{x})$ 关于 v_1 呈非单调变化（大致为二次变化）。由于 v_1 是三个输入变量 x_7, x_8, x_9 的线性组合，因此可以得出结论：这些输入变量中的每一个都会对 $f(\boldsymbol{x})$ 产生非单调影响，至少在该线性组合中另外两个变量取某些固定值时如此。从同一张图可以看出，当 v_1 固定时，随着 h_1 增加，函数 $f(\boldsymbol{x})$ 以近乎线性的方式单调递增。为了可视化 h_1，第二层图表明，对于每个固定的 v_3， h_1 关于 v_2 单调且近乎线性，但关于 v_3 呈非单调变化，适用于

每个固定的 v_2。由此可以得出一个结论：当 v_1 固定时， f 关于 x_4、 x_5 和 x_6 （涉及 v_2 的输入）近乎线性，但关于其他输入变量则呈非线性。请注意， f 关于 h_1 的轻微凹曲率，往往会抵消 h_1 关于 v_2 的轻微凸曲率，因此当 v_1 固定时， f 关于 v_2 近乎线性。

此外，从图中也可以辨别输入变量之间的交互作用。在解读模型时，理解输入之间的交互作用（而非可加性）往往很重要。我们可以从第1层IANN图判断 v_1 中的某个输入变量是否与其余变量交互作用。由于其余变量位于 v_1 或 h_1中，我们考虑两类交互作用：(i) v_1中的变量与 h_1中的变量之间的交互作用，以及 (ii) v_1内部变量之间的交互作用。如果该输入变量不存在这两类交互作用，我们就可以断定它与所有其余变量都没有交互作用。

关于 v_1 与 h_1之间的交互作用，如果 v_1 对 f 的影响（例如，图8左上图中的实线曲线）仅随 h_1 的变化而改变一个加性常数，那么根据定义， v_1 与 h_1 之间不存在交互作用，因此输入 $\boldsymbol{x}_{J_1}$ 与输入$\boldsymbol{x}_{\setminus J_1}$之间也不存在交互作用。关于 v_1内部的交互作用，如果对于每个 h_1 取值， v_1 对 f 的影响都仅为线性，那么 v_1 中的输入变量彼此之间不存在交互作用。反之，如果对于至少某些 h_1 取值， v_1 对 f 的影响是非线性的，那么根据以下论证， v_1 内部存在交互作用。在这种情况下，对于使 v_1 的影响呈非线性的那些 h_1 取值， $\frac{\partial f_{(v_1,\ h_1)}}{\partial v_1} = \tilde{g}(v_1,\ h_1)$ 对某个函数 $\tilde{g}$ 成立，该函数会随 v_1变化而变化；此时，所有具有非零 $\boldsymbol{\beta}$ 系数的输入变量 $\boldsymbol{x}_{J_1}$ 彼此之间都存在交互作用。

从图8中的 IANN可视化可以看出，第 1 层图中的三条实线曲线表明， v_1 与 h_1之间存在很强的交互作用。由于第二层图显示， v_2 和 v_3 都会影响 h_1 ，因此我们得出结论： $\boldsymbol{x}_{J_1}$ 中的输入与 $\boldsymbol{x}_{J_2}$ 和 $\boldsymbol{x}_{J_3}$中的输入相互作用。此外，由于第 1 层图中的实线曲线呈非线性， $\boldsymbol{x}_{J_1}$ 中的输入彼此相互作用。

作为 IANN图用途的另一个示例，我们来看看所谓的稳健参数设计（RPD）问题（Taguchi，1986；Robinson等人，2004）。在这一问题中，一些输入变量是可控的系统/产品设计变量，另一些则是噪声变量，会在系统运行或产品使用过程中不可控地变化。RPD 的目标是为可控输入找到设计值，使函数 f 对不可控输入的变化最不敏感。举例来说，假设构成 v_1 的输入 $\boldsymbol{x}_{J_1}=\{x_7, x_8, x_9\}$ 是噪声变量，而其他输入是可控的设计变量。从图8中的第 1 层图可以看出，当 $h_1 \approx -1.0$时， f 对 v_1的依赖最弱。因此，如果选择$\boldsymbol{x}_{\setminus J_1}$ （ h_1 取决于这些输入），使得 $h_1 \approx -1.0$ （例如，利用图8中的第二层图），那么 f 对噪声变量 $\boldsymbol{x}_{J_1}$的变化将最不敏感。

4 IANN算法细节

本节介绍如何确定分层IANN分解中输入变量的顺序（即每一层出现哪个输入变量），以及算法的其他细节。这里，我们主要介绍用于 OVH IANN结构的相关算法。用于 DASH IANN结构的算法与此类似，其细节见附录A。

为使用 OVH结构最佳近似 f 并使可视化图更便于解读，我们按照顺序确定输入变量的排列：在 OVH结构 (5) 的每一层，选择能带来最佳近似精度的变量。为了解我们这一方法背后的直觉，请注意，如果在 (5) 的第一层中$f(\boldsymbol{x}) = g_1(x_{j_1}, h_1(\boldsymbol{x}_{\setminus j_1}))$，那么根据链式法则，

$$\frac{\partial f}{\partial \boldsymbol{x}_{\setminus j_1}}(\boldsymbol{x}) = \frac{\partial g_1}{\partial \boldsymbol{x}_{\setminus j_1}}(\boldsymbol{x}) = \frac{\partial g_1}{\partial h_1}(\boldsymbol{x}) \nabla h_1(\boldsymbol{x}_{\setminus j_1}), \tag{11}$$

其中 ∇ 表示 　是函数相对于其输入参数的梯度。由于

— $\frac{\partial g_1}{\partial h_1}(\boldsymbol{x})$ 是标量，如果固定 $\boldsymbol{x}_{\setminus j_1}$，并考虑 $\frac{\partial f}{\partial \boldsymbol{x}_{\setminus j_1}}(\boldsymbol{x}) \in \mathbb{R}^{d-1}$ 在许多不同的 x_{j_1} 值下的情况，它们都近似共线，仅在大小和符号上有所不同。因此，作为 x_{j_1} 的候选项，我们使用主成分分析（PCA）来衡量共线程度，并考虑梯度最共线的输入。

我们注意到，尽管 IANN 结构中的函数 g 和 h 出现在公式 (11)（以及下文的公式 (16)）中，但它们并未用于确定 IANN 输入排序的算法。输入排序是一个预处理步骤，它直接使用 f 的梯度，无需拟合 IANN。公式 (11) 和 (16) 仅用于说明我们采用输入排序方法的原理。我们还注意到，确定输入 $\boldsymbol{x}$ 的排序要求函数 f 可微，因为我们需要计算 $f(\boldsymbol{x})$ 关于 $\boldsymbol{x}$的梯度。

以下介绍第1层梯度投影算法的详细步骤。**梯度投影算法：**（对每个输入变量 x_j，j =1、2，$\cdots$，d重复步骤 1–3）

1. 对于输入变量 x_j，在 $\boldsymbol{x}_{\setminus j}$空间中抽取 $N_{\setminus j}$ 个拉丁超立方设计（LHD）样本，记为 $\{\boldsymbol{x}_{\setminus j}^l;\ \ l=1,\ \cdots,\ N_{\setminus j}\}$。然后，在 x_j 空间中抽取 N_j 个均匀分布于 x_j 取值范围内的点，记为 $\{x_j^m;\ \ m=1,\ \cdots N_j\}$。

2. 对于 $l=1,\ 2,\ \cdots,\ N_{\setminus j}$，计算梯度向量 $\boldsymbol{a}^{m,\ l}_{\ j} = \nabla_{\boldsymbol{x}_{\setminus j}} f(x_j^m,\ \boldsymbol{x}_{\setminus j}^l) \in \mathbb{R}^{d-1},\ \ m=1,\ 2,\ \cdots,\ N_j$，并将它们堆叠成 $N_j \times (d-1)$ 矩阵：

$$A_j^l = \begin{bmatrix} \left(\boldsymbol{a}_j^{1,l}\right)^T \\ \vdots \\ \left(\boldsymbol{a}_j^{N_j,l}\right)^T \end{bmatrix}, \quad \text{for } l=1,2,\cdots,N_{\setminus j}. \tag{12}$$

3. 使用 PCA 找出与$\left(A_j^l\right)^T A_j^l$的最大特征值对应的特征向量（$\boldsymbol{z}_j^l$），针对每个 $l=1,\ 2,\ \cdots,\ N_{\setminus j}$，并对特征向量进行归一化，使其满足 $\left|\left|\boldsymbol{z}_j^l\right|\right|$ =1。计算梯度向量 $\boldsymbol{a}^{m,}_{\ j}{}^{l}$ 与其在$\boldsymbol{z}_j^l$上的投影之间的误差，定义为

$\boldsymbol{z}_j^l$，定义为

$$E_j^l = \sum_{m=1}^{N_j} \left|\left| \boldsymbol{a}_j^{m,l} - \left[\boldsymbol{z}_j^l \left(\boldsymbol{z}_j^l \right)^T \right] \boldsymbol{a}_j^{m,l} \right|\right|^2, \quad \text{for } l = 1, 2, \cdots, N_{\setminus j}. \tag{13}$$

然后，通过以下方式计算 x_j 的梯度投影误差（E_j）：

$$E_j = \frac{1}{N_{\setminus j}} \sum_{l=1}^{N_{\setminus j}} \frac{E_j^l}{\Omega_j^l}, \tag{14}$$

其中

$$\Omega_j^l = \frac{1}{N_j} \sum_{m=1}^{N_j} \left|\left| \boldsymbol{a}_j^{m,l} \right|\right|^2 \qquad \text{for } l = 1, 2, \cdots, N_{\setminus j}. \tag{15}$$

请注意，我们在式（14）中使用归一化因子 Ω_j^l，以相对于该 l 的梯度向量平均平方长度来表示投影误差E_j^l。

4. 最后，我们选择投影误差最小的输入变量 x_{j_1}，即 $j_1 = \arg\min\{E_j\colon\ j = 1,\ 2,\ \cdots,\ d\}$.

本文使用最大最小准则（Johnson 等人，1990年；McKay 等人，2000年）生成拉丁超立方体抽样，旨在最大化拉丁超立方设计中任意两个样本之间的最小距离，以增强其空间填充性。还请注意，整个 IANN 建模流程在两种不同用途下使用拉丁超立方设计：首先，在上述梯度投影算法中确定输入排序；随后再次使用它生成用于拟合 IANN 模型的训练数据（见附录 B）。对于前一种用途，我们发现标准拉丁超立方设计效果良好；对于后一种用途，我们发现附录 B 中的改进型拉丁超立方设计始终比标准拉丁超立方设计效果更好。

为后续每一层选择输入的算法类似：假设我们已确定前几层中输入变量的顺序，$\{x_{j_1}, \cdots, x_{j_{i-1}}\}$对于某个 $i > 1$。与寻找 x_{j_1}的过程类似，对于每个候选输入索引 $j_i \in \{1, 2, \cdots, d\} \setminus \{j_1, \cdots, j_{i-1}\}$，我们考虑 $f(\boldsymbol{x})$ 关于剩余输入 $\boldsymbol{x}_{\setminus(j_1,\cdots,j_i)}$的梯度。根据链式法则，若结构 (5) 成立且取等号，则该梯度为：

$$\frac{\partial f}{\partial \boldsymbol{x}_{\setminus(j_1,\cdots,j_i)}}(\boldsymbol{x}) = \frac{\partial g_1}{\partial h_1}(\boldsymbol{x})\frac{\partial h_1}{\partial h_2}(\boldsymbol{x}_{\setminus j_1})\cdots\frac{\partial h_{i-1}}{\partial h_i}(\boldsymbol{x}_{\setminus(j_1,\cdots,j_{i-1})})\nabla h_i(\boldsymbol{x}_{\setminus(j_1,\cdots,j_i)}). \tag{16}$$

接着，固定 $\boldsymbol{x}_{\setminus(j_1,\cdots,j_i)}$，并在 $\boldsymbol{x}_{(j_1,\cdots,j_i)}$空间中取一些拉丁超立方体抽样，使 $(x_{j_1},\cdots,x_{j_i})$ 变化。这样，式 (16) 中的向量 $\nabla h_i(\boldsymbol{x}_{\setminus(j_1,\cdots,j_i)})$ 是一个常向量，因此梯度向量$\frac{\partial f}{\partial \boldsymbol{x}_{\setminus(j_1,\cdots,j_i)}}(\boldsymbol{x})$ 共线，且仅大小和符号不同。因此，我们将上述相同的梯度投影算法应用于后续各层：用 $\boldsymbol{x}_{\overline{\setminus(j_1,\cdots,j_i)}}$ 替换 $\boldsymbol{x}_{\setminus j_1}$，并根据与式 (14) 类似的梯度投影误差度量，将 x_{j_i} 选为剩余输入中在式 (16) 中对应梯度向量共线程度最高的输入。为了避免在每一层反复生成拉丁超立方体抽样，我们只在 d维输入空间中抽样一次，然后在需要时将这些样本投影到较低维空间。

5 数值实验

本节提供两个函数可视化示例，这些函数具有封闭形式的“真实值”表达式，可用于与我们的 IANN 可视化进行比较。补充材料部分提供了一个示例，其中的函数是一个复杂数值模拟的实际代理模型，该模拟针对受应变材料样本的势能进行建模。

5.1 用于钻孔函数的 IANN

为了说明 IANN 图如何帮助函数的可视化与解释，我们使用钻孔函数。

$$f(\boldsymbol{x}) = 2\pi x_1\left(x_4 - x_6\right)\left[\log\left(\frac{x_2}{x_3}\right)\left(1 + 2\frac{x_7 x_1}{\log(\frac{x_2}{x_3})x_3^2 x_8} + \frac{x_1}{x_5}\right)\right]^{-1}, \tag{17}$$

该函数用于模拟两个含水层之间的水流。虽然式（17）的函数形式已知，但在评估代理建模方法时，通常将其视为黑箱函数（An 和 Owen，2001年；Harper 和 Gupta，1983年；Morris 等，1993年；Surjanovic 和

Bingham，2013年）。响应 f 是含水层之间的水流量。此处， $d = 8$、$x_1 \in [0.05, 0.15]$ 和 $x_2 \in [100, 50\,000]$ 分别表示钻孔半径及其影响范围， $x_3 \in [63\,070, 115\,600]$ 和 $x_5 \in [63.1, 116]$ 分别表示上部和下部含水层的导水系数， $x_4 \in [990, 1110]$ 和 $x_6 \in [700, 820]$ 分别表示上部和下部含水层的测压水头， $x_7 \in [1120, 1680]$ 表示钻孔长度，$x_8 \in [9855, 12\,045]$ 表示钻孔的水力传导系数。

由于钻孔函数有八个输入，且每个输入的尺度各不相同，因此在拟合 IANN 模型时，我们首先使用最小-最大归一化，将输入范围重新缩放至 $[0, 1]$ 。OVH结构将以层次方式生成七幅三维可视化图。相比之下，我们使用DASH结构，通过四个线性组合及其层次排序来解释所有输入变量的影响；这些组合和排序由附录A中所述的算法自动确定，该算法与第 4 节中的算法类似。

就计算开销而言，该算法用时 96 秒确定钻孔示例的输入排序。主要的计算开销来自确定输入排序后拟合单个 IANN，这在钻孔示例中耗时 245 秒。拟合单个 IANN 的计算开销与拟合参数数量相当的标准神经网络相当。可以通过交叉验证为 IANN 模型选择最优超参数，其计算开销与拟合单个 IANN 所需的时间成正比。

如果用户希望理解并解释某一个特定输入的影响，可以手动选择线性组合组的排序，使第一组包含所关注的输入变量。表1展示了算法为 DASH IANN 结构生成的排序分组，其中约束条件是第一组必须包含特别关注的输入。八行分别展示了将八个输入中的每一个指定为特别关注对象时所得到的排序。这样，便可以通过第 1 层图更直接地可视化该输入对 f 的影响。其余线性组合的排序

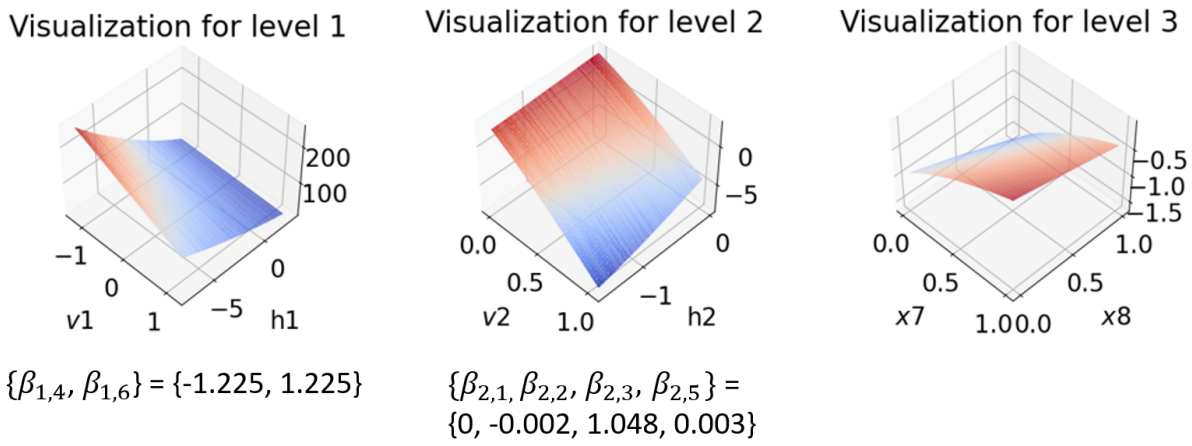
显示在表1的第二列中。对于表1中的每种排序，我们还展示了相应的测试 r^2。图9展示了测试 r^2 最高的 IANN可视化图。

作为比较，我们使用全局敏感性分析（Herman和Usher，2017年）计算 $\{x_1, x_2, \cdots, x_8\}$的总敏感性指数，结果为 $\{0.0, 0.0, 0.174, 0.26, 0.0, 0.261$,0。258， $0.063\}$。这表明 x_1， x_2和 x_5对钻孔函数影响很小，这一点也反映在图9的 IANN 图中。

索引 j	输入组和排序	测试 r^2
1	[[1, 2, 3, 5], [4, 6], [7], [8]]	99.97%
2	[[1, 2, 3, 5], [4, 6], [7], [8]]	99.97%
3	[[1, 2, 3, 5], [4, 6], [7], [8]]	99.97%
4	[[4, 6],[[1, 2, 3, 5], [7], [8]]	99.99%
5	[[1, 2, 3, 5], [4, 6], [7], [8]]	99.97%
6	[[4, 6],[[1, 2, 3, 5], [7], [8]]	99.99%
7	[[7],[4, 6],[[1, 2, 3, 5], [8]]	99.94%
8	[[8],[4, 6],[[1, 2, 3, 5], [7]]	99.99%

表1：对于假设采用 DASH 结构的钻孔函数，列出了$p = 4$ 互不相交的线性组合组及其排序。这些内容由 DASH IANN 结构的算法自动确定，并满足第一组包含指定的特别关注输入变量这一约束；该变量显示在第1列。

我们使用 34，000 个训练样本来拟合 IANN，并使用 10^6 随机生成的测试输入样本来计算测试 r^2。钻孔函数的测试 $r^2 = 99.99\%$。每个线性组合中归一化输入的系数列在图9中各图的下方。由于输入已归一化至 $[0, 1]$, ，比较



$\{\beta_{1,4}, \beta_{1,6}\}$ = {-1.225, 1.225}

$\{\beta_{2,1}, \beta_{2,2}, \beta_{2,3}, \beta_{2,5}\}$ = {0, -0.002, 1.048, 0.003}

图9：公式（17）中钻孔函数的 DASH结构 IANN可视化 f。图中坐标轴以及图下方列出的 β 系数对应归一化输入。

各系数的大小可以判断每个线性组合中所涉及输入的相对重要性。例如，从图9的第 1 层图可以看出，v_1 由两个输入变量组成，即 x_4 和 x_6，相应系数的绝对值相同、符号相反。这与公式（17）中的钻孔函数公式一致，该公式可写为 $f(\boldsymbol{x}) = (x_4 - x_6)h_1(\boldsymbol{x}_{\setminus(4,6)})$，其中 $h_1 = 2\pi x_1 \left[\log\binom{x_2}{x_3}\left(1 + 2\binom{x_7 x_1}{x_2 x_3}\log() x_3^2 x_8 + \binom{x_1}{x_5}\right)\right]$。从第二层图可以看出，除输入变量 x_3 外，系数几乎为零。这也与全局敏感性分析的结果一致，该分析表明 x_1，x_2，x_5 对响应 f 几乎没有影响。

作为比较参考，针对钻孔函数拟合的线性模型的测试 r^2 为 94.68%，这表明钻孔函数在指定输入域上几乎是线性的。后续章节中的示例将展示 IANN 可视化对于非线性程度更高的函数的可解释性。

5.2 谐波波函数的 IANN

重新考虑式 (2) 中的谐波波函数 $f(\boldsymbol{x})$，图 1 和图 3 仅展示了第 1 层的 IANN 可视化。我们首先考虑采用 DASH结构来近似 f，但除了每个 v_i 均为单个输入变量且 $p = d$ 为平凡情形外，它未能找到其他相关的不相交线性组合。因此，我们采用 OVH结构。更一般地说，我们建议用户先尝试 DASH结构；如果它未能产生任何相关的线性组合，使测试r^2达到可接受的高水平，那么用户就应采用 OVH结构。

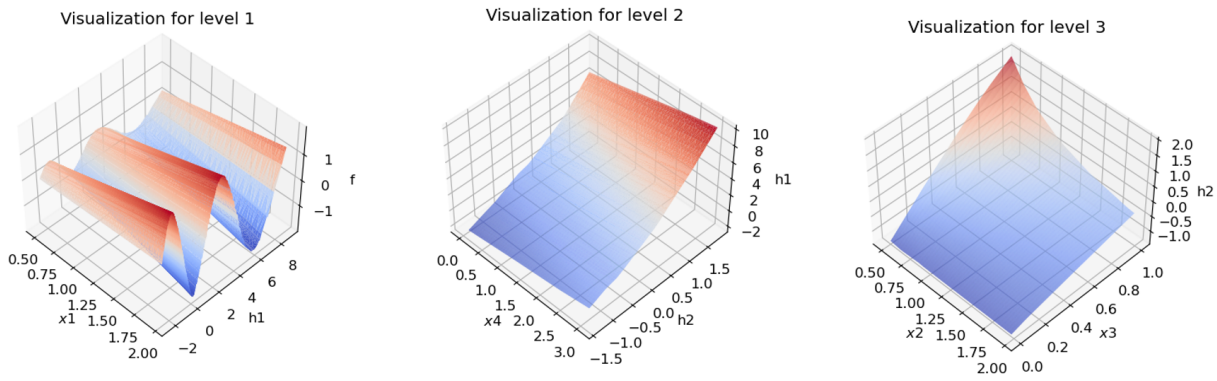


图1：式 (2) 中谐波函数的 OVH结构IANN可视化。测试 $r^2 = 99.95\%$。

从图10中的第 1 层图（与图1相似）可以看出，在其他变量固定时，x_1 对 f（$\boldsymbol{x}$）呈线性影响，而对于每个固定的 x_1值，f（$\boldsymbol{x}$）均为正弦的，这与公式（2）一致。从第2层图可以看出，h_1（$\boldsymbol{x}_{\backslash 1}$）关于 x_4 和 h_2呈线性，且 x_4 和 h_2 之间的交互作用很小，这表明存在加性结构h_1（$\boldsymbol{x}_{\backslash 1}$）$= \alpha_1 h_2$（$\boldsymbol{x}_{\backslash (1,4)}$）$+ \alpha_2 x_4$，其中 α_1 和 α_2为常数。因此，x_4 反映了谐波函数中的相位角。从第 3 级图可以看出，对于每个固定的 x_2，h_2 关于 x_3 呈线性；而对于某些固定的 x_3，h_2 关于 x_2 呈非线性，尤其是在 x_3 较高时。此外，从第 3 级图可以清楚地看出，当 $x_3 = 0$时，x_2 对 h_2 没有影响；而当 $x_3 = 1$时，x_2 对 h_2 有很大的非线性影响，这也与“真实的”

公式（2）中 h_2 （$= 2\pi\frac{x_3}{x_2}$，相差某些常数）的表达式一致。同样，从第 3 级图可以看出，随着 x_2减小， x_3 对 h_2的影响大得多，因而对h_1 （来自第2层图）以及 f （来自第 1 层图）的影响也大得多。由此可知，对于较小的 x_2，当 x_3 在其整个取值范围内变化时，f 会经历更多个正弦周期；这与公式（2）中的观察一致，即 x_2 表示正弦函数的波长。从这个意义上说，IANN 图确实反映了输入变量的物理解释。

在附录 C 中可以找到更多示例，这些示例采用我们的 IANN 方法来可视化实际的黑箱函数。这些函数代表了针对物理系统黑箱计算机模拟输出拟合得到的代理模型。

6 结论与潜在扩展

本文介绍了用于可视化和解释黑箱函数的 IANN 方法。我们从理论上证明， $[0,1]^d$ 上的任意连续函数都可以通过我们提出的、更具可解释性的结构进行任意精度的近似；这种结构可以方便地用所提出的 IANN架构表示。为可视化所有输入变量的影响，我们开发了两种分层结构（OVH 和 DASH），每种结构都可以用特定的 IANN架构表示。我们还开发了算法，以自动确定输入变量的排序，旨在为每种分层结构提供对原始函数的最佳近似。我们通过多个示例展示了 IANN 方法在可解释性方面的优势。

我们设想了若干种对 IANN 方法进行扩展的可能方式，以增强对原始函数 f 的解读，和/或扩大 IANN 能够提供良好近似的 f 类别。增强 IANN可视化图的可解释性的一种方式是开发定制的图形用户界面（GUIs），用户可以通过滑块控件更改每个输入变量 $\{x_1, x_2, \cdots, x_d\}$ 的值，并交互式地可视化 IANN 可视化中对应的点 $\{f, h_1, \cdots, h_{d-2}\}$

在各层绘制的曲面。此外，为了便于创建$\boldsymbol{x}_{\setminus(j,l)}$的一组固定值下 f 与$\{x_j, x_l\}$ 的三维格状图（当 d 较大时，通常需要谨慎选择许多固定的$\boldsymbol{x}_{\setminus(j,l)}$ 值），可以修改 IANN架构，以拟合以下形式的模型

$$f(\boldsymbol{x}) = g(x_j, x_l, h(\boldsymbol{x}_{\setminus(j,l)})), \tag{18}$$

此时，用户只需为标量 h($\boldsymbol{x}_{\setminus(j,l)}$)选择少量固定值，而不必为高维的 $\boldsymbol{x}_{\setminus(j,l)}$选择许多固定值。

人们还可以考虑取消 DASH结构中对线性组合的互不相交限制，这样每个输入就可以出现在多个组合中。我们没有采取这一做法的主要原因是，相较于 IANN 方法的通用性，我们更重视单个输入的影响的可解释性。线性组合之间存在重叠意味着同一个输入变量会出现在多个线性组合中，这会使对该单个输入变量的解释变得不够清晰。鉴于随着线性组合数量的增加，DASH结构中的互不相交集合限制会变得越来越宽松、越来越通用（在极限情况下，每个线性组合都是一个单独的输入，此时 DASH结构会退化为 OVH结构），我们认为，互不相交限制在通用性与可解释性之间构成了合理的权衡。作为未来的工作，我们计划进一步探索这一问题，但目前尚不清楚如何在存在线性组合重叠的情况下保持可解释性。

一种相关的扩展方法是采用修改后的架构 (18)，并直接绘制 f 与 $\{x_j, x_l\}$的关系图，其中 h($\boldsymbol{x}_{\setminus(j,l)}$)由滑动条控制。这种方法的解释方式与三维格状图类似，但完全避免了为省略变量$\boldsymbol{x}_{\setminus(j,l)}$选择任何固定值。类似于第 3.1 节中的 OVH结构，还可以进一步将式 (18) 中的 h 分解为一个关于两个额外输入和第二个潜在函数的函数；在第2层图中，这个潜在函数同样由滑动条控制，图中展示 h 与这两个额外输入的关系，后续层次也依此类推。这种扩展大约可以将所需层数减少

一半，同时还会扩大该方法的适用范围，使其不仅适用于可由 (1) 表示的函数，也适用于可由 (18) 表示的函数。

另一个潜在的扩展是采用二分树 IANN 结构，其第 1 层图为 $f(\boldsymbol{x}) = g_1(h_{1,1}(\boldsymbol{x}_{J_{1,1}}) h_{1,2}(\boldsymbol{x}_{J_{1,2}}))$ 对 $h_{1,1}(\boldsymbol{x}_{J_{1,1}})$ 和 $h_{1,2}(\boldsymbol{x}_{J_{1,2}})$，其中$J_{1,1}$和 $J_{1,2}$是对 $\{1, 2, \cdots, d\}$ 的不相交划分。随后，每个潜在 h 函数都可以在后续层中以类似方式分解，从而生成一组树状嵌套的三维图。为确定每一层的索引组划分，我们预计可以开发出与第 4 节中的 IANN 算法类似的方法。此扩展的主要优势在于，它扩大了我们的 IANN 能够提供良好近似的 f 类别（式（5）中的 IANN 结构是树形 IANN 结构的一个特例，其中每组输入都被划分为一个单独的输入和其余输入），并且有望降低 IANN 可视化中树的深度。

另一个可能的扩展方向，是从当前对黑箱仿真函数进行可视化，拓展到对拟合观测数据的通用监督学习模型进行可视化，以解释输入变量的影响。主要挑战在于，与整个（矩形）输入空间都有意义的黑箱仿真函数不同，监督学习模型的输入训练数据往往高度相关。使用当前的 IANN 方法可视化黑箱监督学习模型，需要外推到输入空间中数据稀少的区域，这会使得解释不可靠。

补充材料

补充材料 此处将介绍多个主题，包括 DASH IANN 结构的算法、定制的 LHD 抽样技术、其他数值示例，以及定理 2.1 的证明。（IANN-supplementary.pdf）**IANN 的 Python 软件包：** “IANN”Python 软件包，其中包含用于执行

文章中所述 IANN 方法的代码以及相关数据集。（iann-codes.zip，IANN 的压缩代码包）

利益披露声明

作者声明不存在需要披露的竞争性利益。

致谢

本研究部分由美国空军科学研究办公室资助，资助编号为 FA9550-18-1-0381，在此谨致谢意。

参考文献

Agarwal, R., L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, 和 G. E. Hin- ton (2021年)。神经加性模型：基于神经网络的可解释机器学习。神经信息处理系统进展 *34*，4699–4711。

An, J. 和 A. Owen（2001年）。准回归。复杂性杂志 *17*(4), 588–607。

Apley, D. W. 和 J. Zhu（2020年）。可视化预测变量在黑箱监督学习模型中的影响。英国皇家统计学会杂志：*B*辑（统计方法学）*82*(4), 1059–1086。

Becker, R. A., W. S. Cleveland 和 M.-J. Shyu（1996年）。格架显示的视觉设计与控制。计算与图形统计学杂志 *5*(2), 123–155。

Becker, R. A., W. S. Cleveland、M.-J. Shyu 和 S. P. Kaluzny（1994年）。格架显示：用户指南。统计研究报告 *10*。

Breiman，L.（2001年）。随机森林。机器学习 *45*（1），5–32。

Buitinck，L.、G. Louppe、M. Blondel、F. Pedregosa、A. Mueller、O. Grisel、V. Niculae、P. Prettenhofer、A. Gramfort、J. Grobler、R. Layton、J. VanderPlas、A. Joly、B. Holt 和 G. Varoquaux（2013年）。机器学习软件的 API 设计：来自 scikit-learn 项目的经验。载于*ECML PKDD* 研讨会：数据挖掘与机器学习语言，第 108–122 页。

Cleveland, W. S. 和 R. McGill（1984年）。图形感知：理论、实验及其在图形方法发展中的应用。美国统计学会会刊 *79*(387), 531–554。

Cybenko, G.（1989年）。通过 S 形函数的叠加进行近似。控制、信号与系统数学 *2*(4), 303–314。

Fisher, A., C. Rudin 和 F. Dominici（2019）。所有模型都是错的，但许多模型都有用：通过同时研究整类预测模型来学习变量的重要性。机器学习研究期刊 *20*(177), 1–81。

Friedman, J. H.（2001年）。贪心函数逼近：梯度提升机。统计年刊, 1189–1232。

Gerber, S., P.-T. Bremer, V. Pascucci 和 R. Whitaker（2010）。高维标量函数的可视化探索。*IEEE* 可视化与计算机图形学汇刊 *16*(6), 1271–1280。

Goldstein、A.、A. Kapelner、J. Bleich 和 E. Pitkin（2015年）。窥探黑箱内部：使用个体条件期望图可视化统计学习。计算与图形统计学杂志 *24*(1)，44–65。

Harper, W. V. 和 S. K. Gupta（1983）。采用拉丁超立方抽样和确定性敏感性方法对钻孔情景进行比较的敏感性/不确定性分析。核废物隔离办公室，Battelle纪念研究所，俄亥俄州哥伦布。

Harvey, W.、O. R¨ubel、V. Pascucci、P.-T.Bremer 和 Y. Wang（2012）。通过Reeb图分裂增强拓扑敏感型聚类。载于数据分析与可视化中的拓扑方法 *II*，第 77–90 页。施普林格。

Herman, J. 和 W. Usher（2017年1月）。SALib：用于敏感性分析的开源 Python 库。开源软件期刊 *2*(9)。

Hornik, K.（1991年）。多层前馈网络的逼近能力。神经网络 *4*(2), 251–257。

Hornik, K.、M. Stinchcombe 和 H. White（1989年）。多层前馈网络是通用逼近器。神经网络 *2*(5), 359–366。

Johnson, M. E.、L. M. Moore 和 D. Ylvisaker（1990年）。极小极大与极大极小距离设计。统计规划与推断期刊 *26*(2), 131–148。

Kennedy, M. C. 和 A. O'Hagan（2000年）。在可获得快速近似时预测复杂计算机代码的输出。*Biometrika 87*(1), 1–13。

Kennedy, M. C. 和 A. O'Hagan（2001年）。计算机模型的贝叶斯校准。皇家统计学会期刊：*B*辑（统计方法学）*63*(3), 425–464。

K¨onig, G., C. Molnar, B. Bischl 和 M. Grosse-Wentrup（2021年）。相对特征重要性。载于*2020*年 第*25*届国际模式识别会议（*ICPR*），第9318–9325页。IEEE。

Lundberg, S. M. 和 S.-I. Lee（2017年）。一种解释模型预测的统一方法。载于第*31*届神经信息处理系统国际会议论文集，第4768–4777页。

Maljovec, D., B. Wang, V. Pascucci, P.-T. Bremer, M. Pernice, D. Mandelli, and R. Nour- galiev（2013年）。用于核反应堆安全的高维标量函数探索

分析与可视化。在核科学与工程应用数学和计算方法国际会议。Citeseer。

McKay, M. D.、R. J. Beckman 和 W. J. Conover（2000年）。计算机代码输出分析中输入变量取值选择的三种方法比较。*Technometrics 42*(1)，55–61。

Morgan-Wall, T.（2022年）。*rayshader*：创建地图并将数据可视化为二维和三维。https://www.rayshader.com，https://github.com/tylermorganwall/rayshader，https://www.rayshader.com/。

Morris, M. D.、T. J. Mitchell 和 D. Ylvisaker（1993年）。贝叶斯设计与计算机实验分析：在曲面预测中使用导数。*Technometrics 35*(3)，243–255。

Pedregosa, F.、G. Varoquaux、A. Gramfort、V. Michel、B. Thirion、O. Grisel、M. Blon- del、P. Prettenhofer、R. Weiss、V. Dubourg、J. Vanderplas、A. Passos、D. Cournapeau、M. Brucher、M. Perrot 和 E. Duchesnay（2011年）。Scikit-learn：Python 中的机器学习。机器学习研究杂志 *12*，2825–2830。

Robinson, T. J., C. M. Borror, 和 R. H. Myers（2004）。稳健参数设计：综述。国际质量与可靠性工程 *20*(1), 81–101。

Sacks, J., W. J. Welch, T. J. Mitchell, 和 H. P. Wynn（1989）。计算机实验的设计与分析。统计科学 *4*(4), 409–423。

Surjanovic, S. 和 D. Bingham（2013）。仿真实验虚拟库：测试函数和数据集。访问日期：2022年1月4日，网址：http://www.sfu.ca/ ssurjano。

Taguchi, G.（1986）。质量工程导论：将质量融入产品和工艺流程。技术报告。