

文章编号:1006-2467(2022)07-0840-10

DOI: 10.16183/j.cnki.jsjtu.2021.191

基于深度学习的智能全局灵敏度分析

吴庶宸¹, 戚宗锋², 李建勋¹

(1. 上海交通大学 电子信息与电气工程学院, 上海 200240;
2. 电子信息系统复杂电磁环境效应国家重点实验室, 河南 洛阳 471003)

摘要: 提出一种将深度学习与灵敏度分析结合的端到端方法, 对深度模型的结构和激活函数进行特殊设计以适应后续灵敏度的计算, 深度模型训练的同时对保存的权重信息进行反向传播计算灵敏度. 在波士顿房价数据集、航迹融合数据集和 G 函数上的实验结果表明, 所提方法相比于 Sobol' 法等经典方法在参数分布不均匀时准确性更高, 具备更强的鲁棒性, 相比于使用传统神经网络的方法准确性更高. 此外, 通过实验验证了基于深度学习模型的样本参数灵敏度可以优化模型的输出结果.

关键词: 卷积神经网络; Inception 模块; 灵敏度分析; Sobol' 方法; 非线性系统分析; 深度学习
中图分类号: TB 112 **文献标志码:** A

Intelligent Global Sensitivity Analysis Based on Deep Learning

WU Shuchen¹, QI Zongfeng², LI Jianxun¹

(1. School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; 2. State Key Laboratory of Complex Electromagnetic Environment Effects on Electronic Information System, Luoyang 471003, Henan, China)

Abstract: This paper proposes an end-to-end method that combines deep learning and sensitivity analysis, which can perform gradient back propagation calculation sensitivity on the saved weight information while training the model. The structure and activation function of the depth model are specially designed to adapt to the subsequent sensitivity calculation. The experimental results conducted on a Boston house prices dataset, a track information fusion dataset, and the G function show that the proposed method is more accurate than classical methods such as Sobol' method when the parameter distribution is uneven, and has a stronger robustness. Compared with the traditional neural network method, the accuracy of the proposed method is higher. The experiment proves that the sample parameter sensitivity obtained by the deep learning model can be used to optimize the model output.

Key words: convolutional neural network; Inception module; sensitivity analysis; Sobol' method; nonlinear system analysis; deep learning

收稿日期: 2021-07-22

基金项目: 国家自然科学基金(61673265), 国家重点研发计划(2020YFC1512203), 民用飞机专项研究(MJ-2017-S-38), 电子信息系统复杂电磁环境效应国家重点实验室基金(2019K0302A)资助项目

作者简介: 吴庶宸(1996-), 男, 北京市人, 硕士生, 主要研究方向为参数灵敏度分析.

通信作者: 李建勋, 男, 教授, 博士生导师; E-mail: lijx@sjtu.edu.cn.

灵敏度分析根据输入参数对输出影响的大小来评估模型输入参数的重要性. 通过研究灵敏度分析结果,研究员可以将重要性较小的输入因素设为定值或者直接去掉,大幅减少模型和问题的复杂性;研究员还可以通过灵敏度结果,优先利用有限的资源来改进重要性较大的输入因素,有针对性地提升模型表现,并得到针对模型输出的归因分析结果^[1].

灵敏度分析包括全局灵敏度分析和局部灵敏度分析. 其中,局部灵敏度分析实现简单、计算量少,但只能反映局部点的单项不确定因素的灵敏度信息,一般适用于线性模型^[2]. 全局灵敏度分析方法可以反映不确定因素在整个变化空间内对输出的影响,适用于非线性、非单调的复杂系统^[3-4].

Sobol' 法是一种基于方差的全局灵敏度计算方法,其将系统输出方差分解为由各输入变量所决定的方差之和,从而定量分析输入变量对系统输出的影响^[5]. 文献[6]对比分析了 Sobol' 法和其他全局灵敏度分析法,发现 Sobol' 法是评价集总式系统单个参数灵敏度和多参数相互作用灵敏度的最有效方法之一. 但是, Sobol' 全局灵敏度分析法需要另外构建系统模型,因此灵敏度的计算精度和准确性很大程度上受到系统模型的影响. 由于 Sobol' 法的计算核心是 Monte Carlo 方法^[6],假设参数在定义域内均匀分布,这导致 Sobol' 法的局限性很强,定义区间如果受到脏数据或者离散点的影响,准确率将大大下降,在实际问题中平均分布的假设很难满足,计算准确性难以保证,并且无法处理未知模型^[7-8]. 文献[9]在 Sobol' 法的基础上提出使用多维 Fourier 变换来降低计算成本的 Fourier 幅度灵敏度检验 (FAST) 法. 文献[10]提出了一种结合 FAST 和随机平衡设计的方法. 文献[11]对这些方法进行了整合和改进. 但是这些方法并没有从根本上解决 Sobol' 法存在的问题,计算量依然很大且有和 Sobol' 法一样的局限和缺点. 文献[12-13]提出基于人工神经网络的灵敏度分析方法,通过把人工神经网络与灵敏度分析结合,计算不同气象条件下蒸发过程的灵敏度. 文献[14]提出使用神经网络结合数据处理 (GMDH) 组合方法计算 Sobol' 灵敏度. 使用神经网络在解决复杂非线性系统的模拟问题的同时,计算出系统全局灵敏度的方法在当下得到了比较多的关注,由于不需 Monte Carlo 采样,这种方法计算效率更高. 但是传统神经网络本身存在训练参数太多、无法提取高维深度相关特征和局部特征以及网络层数不宜过深的问题,而且传统神经网络在建模的时候并没有考虑后续求灵敏度这一步骤,

二者存在割裂,导致求得的灵敏度区分度低,该方法实现效果并不理想.

针对上述问题,将深度学习方法引入全局灵敏度的计算中,利用深度学习方法对交叉特征的提取能力同时提升模型的表现和全局灵敏度分析的准确性,创新性地提出 SInception-卷积神经网络 (CNN) 模型. 该模型的创新点包括:① 通过针对灵敏度分析设计的特征打散机制和 SInception 网络结构增加模型的局部特征组合能力;② 提出串行修正线性单元 (CReLU) 激活函数增加灵敏度计算结果的稀疏性;③ 在训练深度学习网络时额外引入 L1 正则化,以增加网络参数的稀疏性^[15],提升后续灵敏度计算的区分度和准确性. 考虑到研究者多希望利用样本灵敏度有计划、有效率地调节模型输出,所以提出利用训练好的深度学习模型求得每个样本的参数灵敏度,建立系统输出与输入参数的近似函数关系方程以优化系统输出. 在 Kaggle 波士顿房价数据集、工业航迹融合评估数据集和 G 函数 3 个公开数据集中进行测试并与 Sobol' 法和传统神经网络法对比,证明了所提方法的有效性和鲁棒性,得到了准确性指标.

1 Sobol' 法

Sobol' 法基于 Monte Carlo 抽样技术和模型分解思想,能够定量计算系统各输入参数的 1 阶灵敏度和高阶灵敏度以及总灵敏度^[16]. 假设模型可以表示为

$$Y = f(X) = f(x_1, x_2, \cdots, x_k) \tag{1}$$

式中: Y 为模型输出; $X = \{x_1, x_2, \cdots, x_k\}$ 为输入参数; k 为模型输入参数个数. Sobol' 法将函数 $f(X)$ 分解为单个参数或几个参数相互结合的子项之和. 设各参数的空间域为 $\Omega \in [0, 1]$, 则有 $f(x_1, x_2, \cdots, x_k) =$

$$f_0 + \sum_{i=1}^k f_i(x_i) + \sum_{1 \leq i < j \leq k} f_{ij}(x_i, x_j) + \cdots + f_{1, 2, \dots, k}(x_1, x_2, \cdots, x_k) \tag{2}$$

式中: $f_i, f_{ij}, f_{1, 2, \dots, k}$ 为分解后的子项函数,如果输入参数独立,且每项按 0 均值选择、平方可积,那么式中 f_0 为常量,等于输出的期望值. Sobol 证明上式的分解唯一且相互正交,可以通过多重积分求得^[16]:

$$f_0 = \int f(x) \, dx \tag{3}$$

$$f_i(x_i) = \int f(x) \prod_{s \neq i} dx_s - f_0 \tag{4}$$

$f_{ij}(x_i, x_j) =$

$$\int f(x) \prod_{s \neq i, j} dx_s - f_0 - f_i(x_i) - f_j(x_j) \quad (5)$$

用上述方法可以求出所有项. $f(X)$ 的总方差为

$$D = \int f^2(x) dx - f_0^2 \quad (6)$$

其子项方差可以通过 $f(x)$ 的每一子项计算得出:

$$D_{i_1, i_2, \dots, i_s} = \iint \dots \int f_{i_1, i_2, \dots, i_s}^2(x_{i_1}, x_{i_2}, \dots, x_{i_s}) dx_{i_1} dx_{i_2} \dots dx_{i_s} \quad (7)$$

式中: $1 \leq i_1 < \dots < i_s \leq k, s = 1, 2, \dots, k$, 由于 $f_i, f_{ij}, f_{i_1, i_2, \dots, k}$ 子项相互正交, 对式(2)两边平方, 并移项合并可得到下式:

$$D = \sum_{i=1}^k D_i + \sum_{i < j} D_{ij} + \sum_{i < j < l} D_{ijl} + \dots + D_{1, 2, \dots, k} \quad (8)$$

通过方差比可以求得 Sobol' 灵敏度为

$$S_{i_1, i_2, \dots, i_s} = \frac{D_{i_1, i_2, \dots, i_s}}{D} \quad (9)$$

当 $s = i$ 时, 计算结果为参数 x_i 的 1 阶灵敏度系数, 表示 x_i 对输出的主要影响^[4].

在实际工程中, 系统模型函数 $f(X)$ 往往难以获得, Sobol' 法利用 Monte Carlo 法的思想, 通过采样并利用估算式求积分. 具体估算式如下式所示:

$$\hat{D} = \frac{1}{2(c-1)} \sum_{M=1}^{2c} f^2(x_M^{1,2}) - \hat{f}_0^2 \quad (10)$$

$$\hat{D}_i = \frac{1}{c-1} \sum_{M=1}^c f(\mathbf{X}_{(\sim i)M}^1, x_{iM}^1) f(\mathbf{X}_{(\sim i)M}^2, x_{iM}^1) - \hat{f}_0^2 \quad (11)$$

式中: “ \sim ” 表示此变量为估计值; c 为采样个数; $\mathbf{X}_{(\sim i)M} = [x_{1M} \ x_{2M} \ \dots \ x_{(i-1)M} \ x_{(i+1)M} \ \dots]$; x_M 为第

M 个采样点; 上标 1, 2 为不同采样组数.

Sobol' 法有十分明显的缺点, 当模型输入参数个数为 k 时, 每个参数都需要计算 2^k 个 Monte Carlo 积分, 当模型较大、参数较多时, 参数分布不均匀导致需要的采样次数较多, Sobol' 法的计算量十分庞大. 由于 Sobol' 法求得的值为基于采样的估计值, 所以求得的灵敏度精度在很大程度上受到采样精度和样本数量的影响, 尤其在模型样本数量较小时, Sobol' 法无法取得理想效果^[10].

2 基于 SInception-CNN 的特征提取和全局灵敏度分析

2.1 SInception-CNN 模型

SInception-CNN 与传统 CNN 一样包含卷积层、池化层和全连接层 3 种模块. 卷积层通过不同的核函数实现对输入特征的提取; 池化层的作用是减少维度, 对特征进行压缩并提取主要特征; 全连接层进行最终的特征组合. 本文提出通过打散特征的方法, 将输入特征组合为不同的输入特征矩阵, 通过这种方法把结构化的数据转换为“图”, 从而为不同特征组合在一起输入同一个卷积核提供了机会. 一个样本的处理过程如图 1 所示. 其中: q 为一个大小为 2×2 的卷积核提取局部特征后进行非线性变换的函数.

神经网络(NN)作为全连接层受限于不能独立处理局部特征, CNN 的卷积核受限于只能处理输入参数周围空间的局部特征^[17]. 对结构化数据的回归问题和灵敏度分析问题来说, 考虑特征间的相互作用十分重要, 通过引入这种打散输入样本的方式, 引入了特征的多种邻居组合, 为后面的 CNN 卷积核带来多种组合的可能, 引入更多特征间的相互作用.

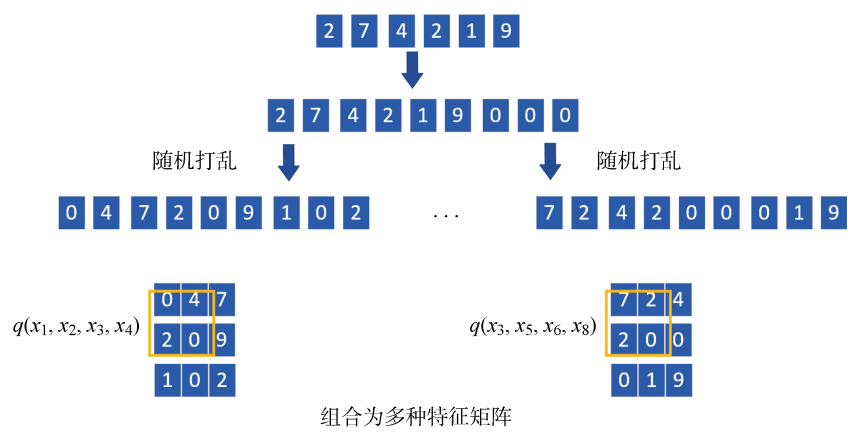


图 1 SInception-CNN 中数据的处理过程
Fig. 1 Data processing in SInception-CNN

SIception-CNN 参考了文献[18]的并行卷积核设计,并使用了丰富的卷积核尺寸,考虑了不同数量、不同空间关系的特征相互作用. SIception-

CNN 的模型结构如图 2 所示. 其中:卷积核 1×1 、 2×2 等卷积核后的数字为卷积后的输出维度; CReLU 为本次卷积操作使用的激活函数.

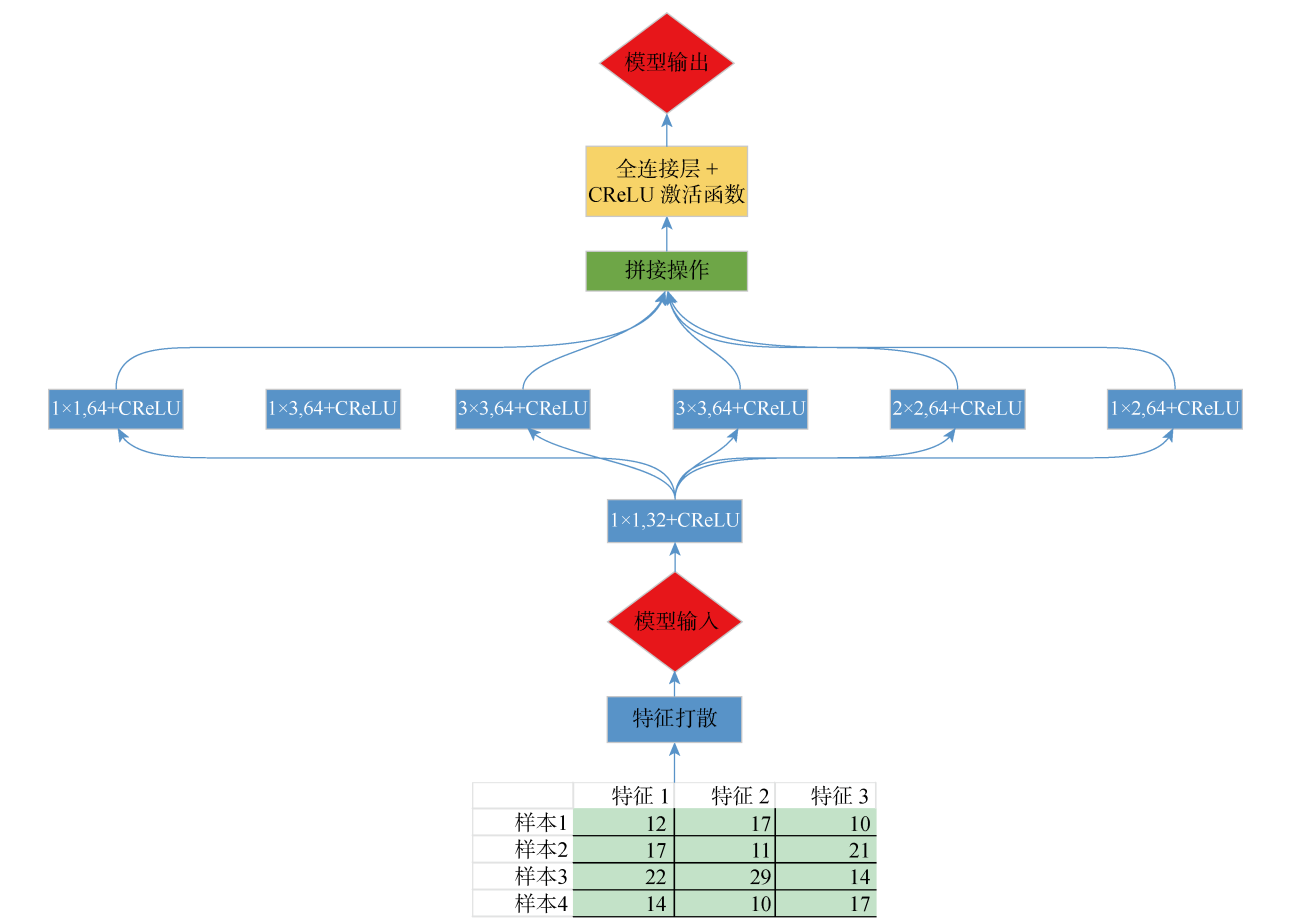


图 2 SIception-CNN 模型结构
Fig. 2 Structure of SIception-CNN model

2.2 CReLU 激活函数

目前,使用最为广泛的 ReLU 激活函数如图 3 所示. 其中: y 为上一层输出; F 为激活函数.

重,达到特征筛选的目的,从而增加灵敏度计算的区分度. CReLU 的激活函数 F_{CReLU} 如图 4 所示. 自适应阈值计算公式如下式所示:

$$U = \frac{1}{H} \sum_{a=1}^H y_a$$

(12)

式中: H 为上一层节点数量; U 为激活函数的自适应阈值; y_a 为上一层第 a 个神经元的输出. 后续实验表明这种激活函数达到了预期效果且没有减少模型的拟合能力.

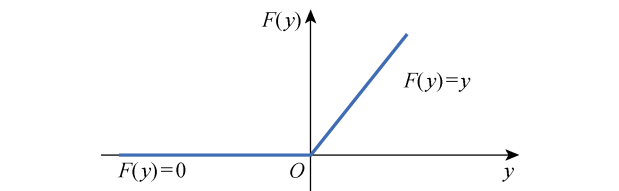


图 3 ReLU 激活函数
Fig. 3 ReLU activation function

如果在使用 ReLU 激活函数的网络进行灵敏度分析,会导致灵敏度结果区分度低,影响准确性. 因为 ReLU 在设计之初的考虑是在引入非线性能力的同时减少计算量、减少梯度消失,并未考虑后续的灵敏度计算. 所以,本文参考文献[19]设计了一种新的激活函数 CReLU,可以自适应杀死小权值的权

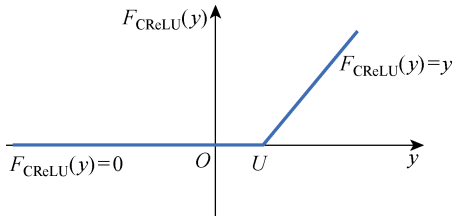


图 4 CReLU 激活函数
Fig. 4 CReLU activation function

2.3 SInception-CNN 模型与全局灵敏度分析的结合

总的来说,全局灵敏度的计算可以分为 3 类方法:基于回归的方法、基于方差的方法和基于矩独立的方法. Sobol'法是基于方差的方法,而神经网络和 SInception-CNN 是基于回归的方法. 基于回归方法的原理是通过建立输入与输出的近似映射,得到输入因子的标准回归系数作为灵敏度指数^[20].

SInception-CNN 是一种深度学习建模方法,全局灵敏度分析是一种数据或者模型的分析方法. 通过两者的结合,既使得深度学习方法可以得到其对输入参数的灵敏度信息,不再是一个完全的“黑盒”,又把深度学习很强的特征提取能力赋予了全局灵敏度分析. 通过引入深度学习,全局灵敏度分析具有了智能,不再需要人工对特征进行组合提取,不再需要对参数的定义域进行人为选定. 基于深度学习的全局灵敏度分析是一种智能的端到端的分析方法.

基于深度学习的智能全局灵敏度分析方法先通过数据训练深度学习模型 SInception-CNN,再利用其权重信息的反向传播求模型的全局灵敏度. 本方法既可以看作是一种对深度学习模型的分析方法,又是一种新的分析全局灵敏度的思路. 虽然本方法对灵敏度的计算模式类似于局部灵敏度方法中的求导法,但是得益于卷积和神经网络的参数共享和全连接结构,本方法所求得的结果为全局灵敏度,即包含了各个特征之间相互作用的计算结果. 特征参数在卷积特征图中输入,经过卷积结构和全连接层,在输出神经元输出,这个过程中包含了大量的共用权重计算,而这些权重值是由所有样本的所有特征参数训练而得出的,因此用本方法求得的灵敏度结果包含了全局信息,更符合全局灵敏度的定义.

3 SInception-CNN 网络计算灵敏度

3.1 全连接层神经元灵敏度的计算

全连接层一般置于整个网络的末尾,因此在反向传播算法中先计算全连接层神经元的灵敏度. 典型的 3 层全连接层如图 5 所示. 其中: z_m 为输入层 z 的第 m 个神经元; h_n 为隐含层 h 的第 n 个神经元; o_p 为输出层 o 的第 p 个神经元; b_1 、 b_2 为偏置系数; w_1 为 z 与 h 之间的连接权重; w_2 为 h 与 o 的连接权重.

前两层间的前向传播公式如下所示:

$$h_n = F_{\text{CReLU}} \left(\sum_{i=1}^H w_{1i}^n z_m + b_n \right) \quad (13)$$

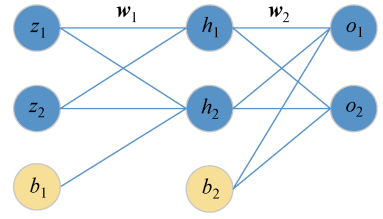


图 5 全连接层结构图

Fig. 5 Fully connected layer structure

式中: w_1^m 为 w_1 中连接输入层第 m 个节点和隐含层第 n 个节点的权重值; b_n 为第 n 个节点的偏置系数.

对如图 5 所示的全连接层模型,应用 BP 算法的思想,如果想知道输入层 z 的每个神经元对输出层 o 的神经元产生了多少影响,可以用 o 对 z 求导得出,类似于局部灵敏度分析中的直接求导法,具体如下式所示:

$$\frac{\partial o}{\partial z} = \left(\sigma_2 \odot \frac{\partial o}{\partial h} \right) \left(\sigma_1 \odot \frac{\partial h}{\partial z} \right) \quad (14)$$

$$\frac{\partial o}{\partial h} = w_1, \quad \frac{\partial h}{\partial z} = w_2 \quad (15)$$

式中: σ_L ($L=1,2$) 为第 L 层的激活函数 F_{CReLU} 的导数矩阵,每个神经元输出对输入的导数为 1 项;符号 \odot 为矩阵对应项相乘; $\partial o / \partial z$ 为输入层 z 对输出层 o 的灵敏度; $\partial o / \partial h$ 为输入层 z 对隐含层 h 的灵敏度; $\partial h / \partial z$ 为隐含层 h 对输入层 z 的灵敏度. 输入层的每个神经元对模型输出 E (输出层 o 的下一层,只有 1 个输出神经元) 的灵敏度计算公式如下式所示:

$$\frac{\partial E}{\partial z} = \frac{\partial o}{\partial z} \delta^\circ = (w_1 \odot \sigma_1) (w_2 \odot \sigma_2) \delta^\circ \quad (16)$$

式中: δ° 为输出层 o 对模型输出 E 的灵敏度.

3.2 CNN 灵敏度的计算

CNN 网络包含卷积层和池化层,卷积层和池化层结构,如图 6 所示,其中: d 为通道总数.

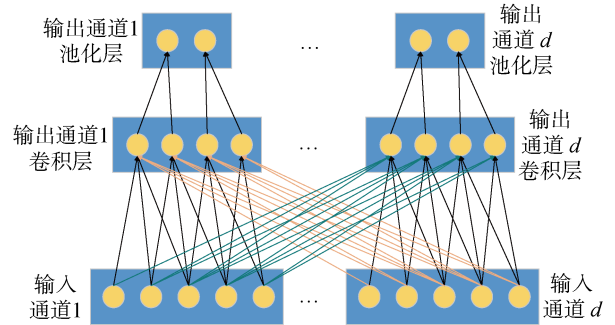


图 6 卷积层和池化层结构图

Fig. 6 Structures of pooling layer and convolutional layer

假设卷积步长为 1,核的尺寸为 1×3 . 卷积层中, x_{dm}^{out} 为输出通道 d 的第 m 个神经元值, x_{vm}^{in} 为输

入通道 v 的第 n 个神经元值, b_d 为输出通道 d 的神经元的偏置系数, $w_{v(n)d(m)}$ 为连接输入通道 v 的第 n 个神经元与输出通道 d 的第 m 个神经元的权重. 前向传播公式如下式所示:

$$x_{dm}^{\text{out}} = F_{\text{ReLU}}(x_{vm}^{\text{in}} w_{v(n)d(m)} + x_{v(n+1)d(m)}^{\text{in}} w_{v(n+1)d(m)} + \cdots + b_d) \quad (17)$$

第 l 卷积层的前向传播公式的矩阵形式如下:

$$\mathbf{x}^{\text{out}} = F_{\text{ReLU}}(\mathbf{w}_l \mathbf{x}^{\text{in}} + \mathbf{b}_l) \quad (18)$$

式中: \mathbf{x}^{out} 为第 l 个卷积层的输出结果; \mathbf{x}^{in} 为第 l 个卷积层的输入; \mathbf{w}_l 为相应卷积核参数; \mathbf{b}_l 为第 l 个卷积层的偏置系数.

则求得当前卷积层输入神经元对模型输出 E 的灵敏度的矩阵形式为

$$\frac{\partial E}{\partial \mathbf{x}^{\text{in}}} = \frac{\partial \mathbf{x}^{\text{out}}}{\partial \mathbf{x}^{\text{in}}} \boldsymbol{\delta}^{\text{out}} = \boldsymbol{\delta}^{\text{out}} \text{rot180}(\mathbf{w}_l) \odot \boldsymbol{\sigma}_l \quad (19)$$

式中: $\boldsymbol{\delta}^{\text{out}}$ 为下一层 \mathbf{x}^{out} 对模型输出 E 的灵敏度; rot180 为将卷积参数矩阵翻转 180° ; $\boldsymbol{\sigma}_l$ 为第 l 个卷积层激活函数的导数矩阵.

3.3 池化层灵敏度的计算

池化方法分为均值池化和最大池化. 池化的目的是通过减少计算图的大小减少计算参数, 并再次提取有价值特征^[21]. 最大池化因为适合分离稀疏特征被更广泛的使用, 并且它可以减少卷积层参数误差造成均值的偏移, 提取出纹理特征^[22]. 最大池化的前向传播是把计算图中的最大的特征变量传递给下一层. 反向传播是把梯度直接传给前一层最大特征的位置. 所以通过保存的池化层输入中对应尺寸的最大值位置即可建立池化层梯度矩阵. 池化层输入 \mathbf{C}^{in} 对模型输出 E 的灵敏度如下式所示:

$$\frac{\partial E}{\partial \mathbf{C}^{\text{in}}} = \frac{\partial \mathbf{C}^{\text{out}}}{\partial \mathbf{C}^{\text{in}}} \boldsymbol{\delta}^{\text{out}} = \boldsymbol{\delta}^{\text{out}} \odot s(\mathbf{C}^{\text{in}}) \quad (20)$$

式中: \mathbf{C}^{out} 为池化层输出; $\boldsymbol{\delta}^{\text{out}}$ 为池化层输出对模型输出 E 的灵敏度; s 函数根据正向传播的池化逻辑为灵敏度重新分配权重导数.

3.4 标准化对灵敏度的影响

神经网络模型在输入时需要标准化处理, 设标准化过程为

$$x^{\text{after}} = \frac{x^{\text{row}} - \mu(x^{\text{row}})}{\text{std}(x^{\text{row}})} \quad (21)$$

式中: $\mu(x^{\text{row}})$ 为原始输入 x^{row} 的平均值; $\text{std}(x^{\text{row}})$ 为的原始输入 x^{row} 的标准差; x^{after} 为标准化后的模型输入层. 则 x^{row} 对模型输出 E 的灵敏度为

$$\frac{\partial E}{\partial x^{\text{row}}} = \text{std}(x^{\text{row}}) \frac{\partial E}{\partial x^{\text{after}}} = \text{std}(x^{\text{row}}) \delta^{\text{row}} \quad (22)$$

式中: δ^{row} 为标准化后输入 x^{row} 对模型输出的灵敏度.

4 实验结果和分析

在一个工业数据集、一个公开的真实数据集和用 G 函数生成的仿真数据分别进行了实验, 以验证本方法的有效性. 工业数据集为工业航迹融合结果评估数据集; 真实数据集为 Kaggle 波士顿房价数据集. 本文通过真实数据集的对比 Sobol'法和神经网络法验证本方法的鲁棒性和有效性, 通过 G 函数生成数据集量化本方法的准确性和广泛适用性.

4.1 Kaggle 波士顿房价数据集进行灵敏度分析

数据集包含 37 个特征和 1 个标签, 其特征包括房屋的大小、位置、层数等信息, 标签是房屋价格, 将数据集输入模型.

在对比试验中, NN、CNN 和 SInception-CNN 使用了相同的训练参数数量, 模型评价指标体系如表 1 所示, 使用了均方误差 (MSE), 平均绝对误差 (MAE) 和平均误差率 3 个指标. 由表 1 可以看出, SInception-CNN 模型在测试集的效果最优.

表 1 Kaggle 波士顿数据集中的不同模型评价体系
Tab. 1 Evaluation index of different models in Kaggle Boston dataset

模型	测试集 MSE	测试集 MAE	测试集平均 误差率/%
NN	0.268	2.21	13.50
CNN	0.095	1.62	9.90
SInception-CNN	0.072	1.54	9.41

神经网络方法和 SInception-CNN 方法灵敏度分析结果如图 7 所示, 其中: S 为输入特征的灵敏度计算值. 由图 7 可以看出, 由于 SInception-CNN 特

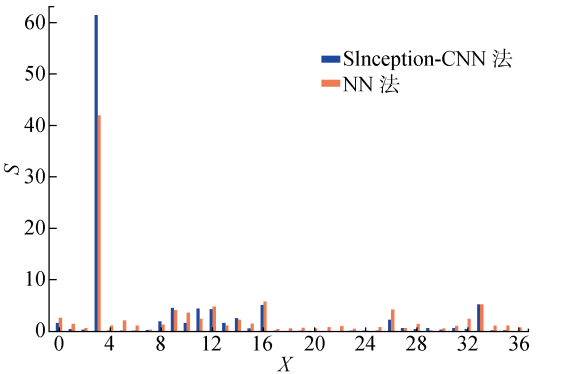


图 7 Kaggle 波士顿数据集神经网络法和 SInception-CNN 的计算结果对比

Fig. 7 Comparison of NN method and SInception-CNN method on calculation results in Kaggle Boston dataset

殊的激活函数设计,以及加入 L1 正则化,得到的灵敏度结果更具有稀疏性,筛选掉无效特征.

灵敏度计算结果鲁棒性对比如表 2 所示,其中:括号内的值为当前特征的灵敏度排序,本方法在多

维特征数据集中可以取得与神经网络方法相近的结果但更具稀疏性. 在数据集中引入一个脏样本,这个人工生成样本的特征 3 取值为正常值的 3 倍,改变了特征 3 的参数区间.

表 2 SInception-CNN 和 Sobol’法鲁棒性对比
Tab. 2 Comparison of robustness between SInception-CNN and Sobol’ methods

参数	SInception-CNN		参数	Sobol’法	
	正常	加入脏样本		正常	加入脏样本
房屋总面积	61.52(1)	64.07(1)	房屋总面积	64.25(1)	93.64(1)
地上居住面积	5.22(2)	5.49(2)	房屋一层面积	10.52(2)	0.13(8)
杂项物品总价值	5.05(3)	4.37(3)	地下室已装修面积	9.51(3)	0.55(5)
地下室已装修面积	4.47(4)	3.96(4)	杂项物品总价值	3.96(4)	1.15(3)
地下室未装修面积	4.36(5)	3.88(5)	车库面积	3.05(5)	0.12(9)

与 Sobol’法的鲁棒性对比中,在加入脏数据(使参数区间产生变化)后 Sobol’法的计算结果变动非常大,除了第 1 位灵敏度参数,排序都发生了变化,而 SInception-CNN 法所受到的影响很小,排序结果没有发生变化,这是由于 SInception-CNN 法通过学习输入与输入参数的非线性关系即模型结构计算灵敏度,少量样本不会影响基本非线性映射,所以鲁棒性高. 而 Sobol’法依赖于参数区间提供采样范围,当脏数据影响采样范围时,计算结果将产生很大偏差.

4.2 工业航迹信息融合结果评估数据集的灵敏度分析

工业航迹信息融合结果评估数据集包含 25 个特征和一个标签. 这个数据集的内容是对一套中航工业的航迹融合算法的评估数据. Sobol’法、神经网络法和 SInception-CNN 法求得的灵敏度结果如图 8 所示.

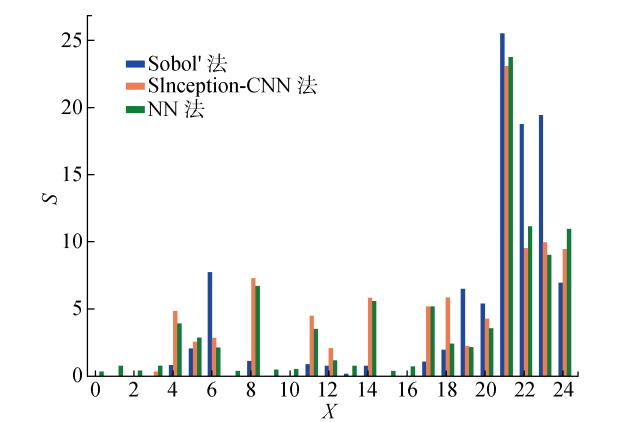


图 8 航迹信息融合数据集的灵敏度结果对比
Fig. 8 Comparison of calculation results in track information fusion dataset

由图 8 可知,Sobol’法在 21、22、23 这 3 个特征上获得了较大的灵敏度,这是因为这 3 个变量的取值范围很大,但分布非常不平衡,90%的样本都在取值范围的右端,这显然对依靠变量取值范围进行采样的 Sobol’法造成了很大的影响. 但是 SInception-CNN 方法和 NN 方法不会产生这种问题,而且 SInception-CNN 方法又在结果的稀疏性上强于 NN 法.

4.3 G 函数的灵敏度分析

针对 SInception-CNN 法的准确性问题,使用 G 函数进行测试. G 函数被广泛应用于各种数值统计方法(灵敏度分析方法),因为其相当复杂,且参数灵敏度可以计算^[23]. G 函数形式如下所示:

$$G(X_1, X_2, \dots, X_N, \beta_1, \beta_2 \cdots \beta_N) = \prod_{j=1}^N g_j \tag{23}$$

式中: X_j 为模型输入参数; β_j 为决定参数灵敏度的调整项; J 为输入参数个数,且 $J = 1, 2, \dots, N$; N 为输入参数总数,则 g_j 可以表示为

$$g_j = \frac{|4X_j - 2| + \beta_j}{1 + \beta_j} \tag{24}$$

G 函数的真实灵敏度在文献[23]中求出,计算方法在文献[24]中有详细介绍. 本实验中使用 6 个输入参数, $a_k = \{0, 0.5, 3, 6, 9, 99, 99\}$, 假设所有参数的空间域为 $[0, 1]$.

实验使用 Sobol’序列生成随机数,并构造关于 G 函数的数据集. 分别使用 Sobol’法和 SInception 法计算模型灵敏度,并与真实值对比,求二者的平均误差 L_{MAE} 可以表示为

$$L_{MAE} = \frac{1}{10} \sum_{r=1}^{10} \left[\frac{\sum_{j=1}^N |\hat{T}_j - T_j|}{N} \right]_r \tag{25}$$

式中: \hat{T}_J 为参数灵敏度的真实值; T_J 为求得的灵敏度; r 为实验次数, 10 次取平均误差. 实验结果如图 9 所示.

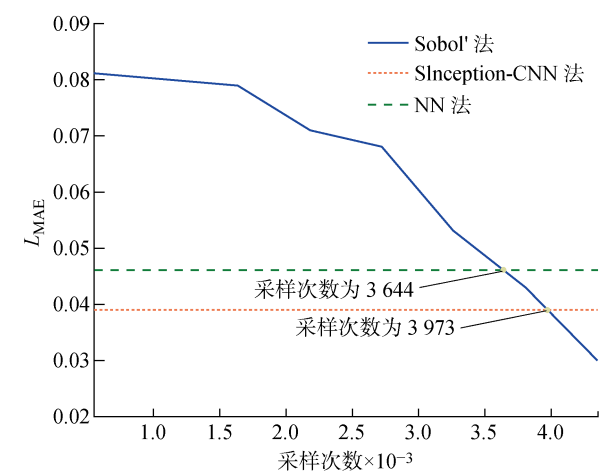


图 9 G 函数中 3 种方法准确率对比
Fig. 9 Accuracy comparison of three methods in G function

NN 法和 SInception-CNN 法作为基于回归的方法,数据来源于数据集,因此不需要额外采样. 由图 9 可以看出,针对 G 函数,当 Sobol' 法的采样次数小于 3 644 时, NN 灵敏度分析方法的精确度大于 Sobol' 法; 当 Sobol' 法的采样次数小于 3 973 时, SInception-CNN 法的精确度大于 Sobol' 法. 因为深度学习模型是一个黑盒系统,其内部存在一些噪声,所以根据深度学习模型来求灵敏度时计算结果不可避免地包含了一些噪声. 基于深度学习的全局灵敏度分析方法在理想生成数据中的准确性低于 Sobol' 法. SInception-CNN 方法更适用于模型未知、参数分布不均匀或者无法进行大量采样的情况.

5 对模型输出的组合提升

输入特征与输出之间的关系以 SInception-CNN 卷积网络权重的形式保存了下来. 因此使用 SInception-CNN 方法可以求得每个样本的参数灵敏度. 针对单个样本的参数灵敏度相当于输入参数对系统输出的近似梯度. 使用参数灵敏度可以得到输入与输出间的自适应线性关系方程,继而选择最有效的参数组合来提升输出.

数据集使用的 Sklearn 波士顿房价数据集,有 13 个输入变量,1 个输出变量. 各个参数的实际意义如表 3 所示.

将 69 号样本输入训练好的 SInception-CNN 模型,并根据输入对模型进行反向传播求梯度,结果如图 10 所示.

表 3 Sklearn 波士顿数据集中参数的实际意义
Tab. 3 Actual meaning of features in Sklearn Boston dataset

特征	定义
CRIM	城镇人均犯罪率
ZN	占地 2 322. 576 平方米以上住宅用地比例
INDUX	每个城镇工业用地的比例
CHAS	查尔斯河(如果临近河流则为 1;否则为 0)
NOX	一氧化氮浓度
RM	住宅平均房间数
AGE	1940 年建造的房屋比例
DIS	与 5 个就业中心的距离加权和
RAD	公路通达性
TAX	资产税率
PTRATIO	师生比例
B	黑人比例
LSTAT	低收入人口比例

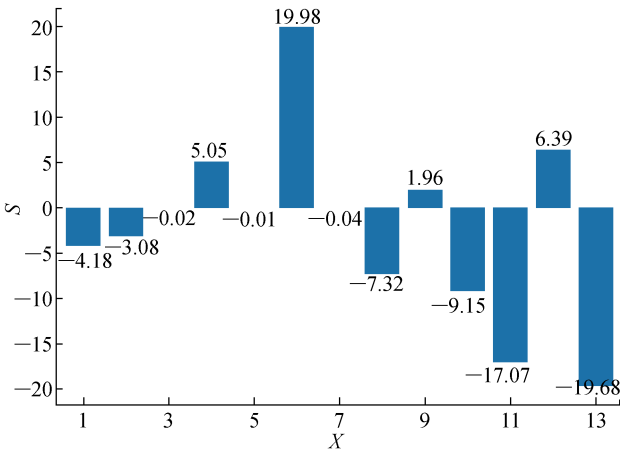


图 10 第 69 号样本的参数灵敏度
Fig. 10 Sensitivity of the 69th sample

经验证,求得结果可以真实反映参数的实际意义. 例如,参数 1(人均犯罪率),其值越高则房价越低,那么求得的近似梯度为负,其他参数都可合理解释.

以其中第 69 号样本为例,选取梯度为正的几个参数将其均提高 0.1(标准化后),输入 SInception-CNN 模型,输出从 20.18 提高到 20.46,提升了 0.28,依据灵敏度对预测差值的计算为 0.253,误差为 0.14%. 利用 SInception-CNN 法计算的样本参数灵敏度可以得到最优调整策略,并得到预期变化量.

6 结语

本文结合深度学习和全局灵敏度分析提出了一种智能全局灵敏度分析方法. 使用 SInception-CNN 以模型权重形式学习输入与输出间的复杂关系, 通过梯度反向传播计算系统参数灵敏度和全局灵敏度, 在建模的同时完成全局灵敏度的计算. 这种新颖的结合既使得深度学习可以得到其对输入参数的灵敏度信息, 不再是一个完全的“黑盒”, 又把深度学习很强的特征提取能力赋予了全局灵敏度分析. 全局灵敏度分析不再需要人工对特征进行组合提取, 不再需要对参数的定义域进行人为选定, 可以直接根据数据得出结果, 成为了一种端到端的智能方法.

在两个真实世界数据集中, 证明了 SInception-CNN 相比于经典方法 Sobol' 法有更强的鲁棒性且不依赖已有模型, 相比于 NN 法有更强的准确性和稀疏性. 通过对 G 函数进行灵敏度分析, 得到了 SInception-CNN 在理想生成数据中的准确性指标. 此外, 本文还提供了一种根据智能全局灵敏度分析方法优化模型输出的方法.

参考文献:

- [1] SALTELLI A, ALEKSANKINA K, BECKER W, *et al.* Why so many published sensitivity analyses are false: A systematic review of sensitivity analysis practices[J]. **Environmental Modelling and Software**, 2019, 114: 29-39.
- [2] ZHANG X Y, TRAME M, LESKO L, *et al.* Sobol sensitivity analysis: A tool to guide the development and evaluation of systems pharmacology models[J]. **CPT: Pharmacometrics and Systems Pharmacology**, 2015, 4(2): 69-79.
- [3] 周云峰, 周永潮, 郑春华, 等. 采用 Sobol 方法的暴雨径流管理模型参数灵敏度分析[J]. **浙江大学学报(工学版)**, 2019, 53(2): 347-354.
ZHOU Yunfeng, ZHOU Yongchao, ZHENG Chunhua, *et al.* Sensitivity analysis of parameters of storm water management model with Sobol method[J]. **Journal of Zhejiang University (Engineering Science)**, 2019, 53(2): 347-354.
- [4] 王丰华, 王劭菁, 苏磊, 等. 采用 Sobol' 算法分析变电站接地网关键参数的灵敏度[J]. **高电压技术**, 2017, 43(1): 300-306.
WANG Fenghua, WANG Shaojing, SU Lei, *et al.* Sensitivity analysis of substation grounding grid parameters based on Sobol' method [J]. **High Voltage Engineering**, 2017, 43(1): 300-306.
- [5] 何琨, 严正, 徐潇源, 等. 基于 Sobol' 法的孤岛微电网潮流全局灵敏度分析[J]. **电力系统自动化**, 2018, 42(14): 99-106.
HE Kun, YAN Zheng, XU Xiaoyuan, *et al.* Sobol' method based global sensitivity analysis of power flow in islanded microgrid[J]. **Automation of Electric Power Systems**, 2018, 42(14): 99-106.
- [6] TANG Y, REED P, WAGENER T, *et al.* Comparing sensitivity analysis methods to advance lumped watershed model identification and evaluation [J]. **Hydrology and Earth System Sciences**, 2007, 11(2): 793-817.
- [7] ROSOLEM R, GUPTA H V, SHUTTLEWORTH W J, *et al.* A fully multiple-criteria implementation of the Sobol' method for parameter sensitivity analysis [J]. **Journal of Geophysical Research: Atmospheres**, 2012, 117: 1-18.
- [8] KHORASHADI ZADEH F, NOSSENT J, SARRAZIN F, *et al.* Comparison of variance-based and moment-independent global sensitivity analysis approaches by application to the SWAT model[J]. **Environmental Modelling and Software**, 2017, 91: 210-222.
- [9] CUKIER R I, LEVINE H B, SHULER K E. Non-linear sensitivity analysis of multiparameter model systems[J]. **Journal of Computational Physics**, 1978, 26(1): 1-42.
- [10] TARANTOLA S, GATELLI D, MARA T A. Random balance designs for the estimation of first order global sensitivity indices [J]. **Reliability Engineering and System Safety**, 2006, 91(6): 717-727.
- [11] TISSOT J Y, PRIEUR C. Bias correction for the estimation of sensitivity indices based on random balance designs [J]. **Reliability Engineering and System Safety**, 2012, 107: 205-213.
- [12] NOURANI V, SAYYAH FARD M. Sensitivity analysis of the artificial neural network outputs in simulation of the evaporation process at different climatologic regimes [J]. **Advances in Engineering Software**, 2012, 47(1): 127-146.
- [13] JAIN S K, NAYAK P C, SUDHEER K P. Models for estimating evapotranspiration using artificial neural networks, and their physical interpretation [J]. **Hydrological Processes**, 2008, 22(13): 2225-2234.
- [14] SONG S F, WANG L. Modified GMDH-NN algorithm and its application for global sensitivity analysis [J]. **Journal of Computational Physics**, 2017, 348: 534-548.
- [15] PARK M Y, HASTIE T. L1-regularization path al-

gorithm for generalized linear models[J]. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, 2007, 69(4): 659-677.

[16] NOSSENT J, ELSEN P, BAUWENS W. Sobol' sensitivity analysis of a complex environmental model [J]. **Environmental Modelling and Software**, 2011, 26(12): 1515-1525.

[17] HADJI I, WILDES R P. What do we understand about convolutional networks? [EB/OL]. (2018-05-23) [2021-06-11]. <https://arxiv.org/abs/1803.08834>.

[18] LIN M, CHEN Q, YAN S. Network in network [EB/OL]. (2013-12-16) [2021-06-11]. <https://arxiv.org/abs/1312.4400>.

[19] BA, JIMMY LEI, JAMIE RYAN KIROS. Layer normalization[EB/OL] (2016-07-21) [2021-06-13]. <https://arxiv.org/abs/1607.06450>.

[20] CONFALONIERI R, BELLOCCHI G, BRGAGLIO S, *et al.* Comparison of sensitivity analysis techniques: A case study with the rice model WARM [J]. **Ecological Modelling**, 2010, 221(16): 1897-1906.

[21] BHATTACHARJEE K, PANT M, ZHANG Y D, *et al.* Multiple Instance Learning with Genetic Pooling for medical data analysis[J]. **Pattern Recognition Letters**, 2020, 133: 247-255.

[22] GRAHAM B. Fractional max-pooling [EB/OL]. (2014-12-18) [2021-06-13]. <https://arxiv.org/abs/1412.6071>.

[23] MARREL A, IOOSS B, LAURENT B, *et al.* Calculations of Sobol indices for the Gaussian process metamodel [J]. **Reliability Engineering and System Safety**, 2009, 94(3), 742-751.

[24] ARCHER G E B, SALTELLI A, SOBOL I M. Sensitivity measures, anova-like techniques and the use of bootstrap [J]. **Journal of Statistical Computation and Simulation**, 1997, 58(2): 99-120.

(本文编辑:石易文)