用于时间序列双向降维的一种函数型方法

Aniruddha Rajendra Rao Haiyan Wang 工业人工智能实验室，日立美国有限公司研发部
工业人工智能实验室，日立美国有限公司研发部 加利福尼亚州圣克拉拉 加利福尼亚州圣克拉拉

Chetan Gupta 工业人工智能实验室，日立
美国有限公司研发部 加利福尼亚州圣克拉拉

{Aniruddha.Rao、Haiyan.Wang、Chetan.Gupta}@hal.hitachi.com

arXiv:2301.00357v1 [cs.LG] 1 Jan 2023

关键词： 函数型数据分析、深度学习、自编码器、函数型神经网络、时间序列、降s维。

摘要

数据的增长使降维技术的需求日益增加，尤其是在非标量变量领域，包括时间序列、自然语言处理和计算机视觉。本文通过函数型数据分析，专门研究时间序列的降维问题。当前函数型数据的降维方法包括函数型主成分分析和函数型自编码器，但这些方法仅适用于线性映射，或只能为时间序列提供标量表示，效率较低。在真实数据应用中，数据的性质要复杂得多。我们提出一种非线性函数对函数方法，由函数型编码器和函数型解码器组成，通过由连续神经元构成的连续隐藏层学习函数型数据内在的结构，从而解决现有方法中的上述问题。我们的方法通过减少函数特征的数量以及观测函数的时间点数量，得到低维潜在表示。我们通过多项模拟和真实数据示例展示了所提模型的有效性。

1 引言

如今，随着信息技术的快速发展，许多领域都在以惊人的频率持续收集数据，例如安装在工业设备上的传感器、经济因素、个人健康以及环境暴露。数据量呈指数级增长，给数据的存储、传输和分析带来了挑战。

降维通过以有意义的方式压缩数据，在解决这一问题中发挥着关键作用。降维的目标是通过将数据投影到信息损失最小的低维子空间，在数学上降低数据的维度。所学习的数学映射不仅能有效地梳理出哪些变量较为重要，还能揭示变量之间的相互作用方式。降维还有助于应对维数灾难、促进信息传输、降低存储需求并缩短计算时间。降维的实用价值使其成为一个活跃的研究领域。尤其是在过去几年中，构建降维模型已被证明有益于许多领域，例如时间序列、文本、图像和视频。这吸引了科学界研究人员的关注，尤其是在非标量类型的数据日益普及的情况下 [27, 13, 24, 8, 21]。遗憾的是，时间序列数据的降维仍有许多工作要做。

本文旨在探讨时间序列数据降维问题。这类数据常见于工业和研究领域的诸多方向，如经济学、气象学、金融、健康、传感器等。例如，每周气温、每小时传感器数据、每日股票收益，以及患者每月的血压读数。这个问题既困难又有趣，因为时间序列由同一组变量在不同时间戳上的随机观测组成，而由于复杂的时间相关性，其中的信息既非独立，也非线性。从数学角度来说，我们要构建一个映射，将某个时间区间 $\mathcal{S}$ 内按时间顺序测量的多个数值变量，映射为同一时间区间 $\mathcal{S}$内按时间顺序测量的少数几个数值变量。

本文考虑一种类似于自编码器的通用设定，其中输入和输出时间序列相同，并利用潜在空间表示学习时间序列的降维形式。主成分分析（PCA）和自编码器（AE）等标准机器学习降维方法，旨在处理在规则间隔的时间点记录且位于有限维空间中的数据。这些

方法具有以下　　局限性：忽略复杂的时间相关性，无法捕捉复杂关系　　，
时间序列数据的标量表示　列数据 [2]。

函数型数据分析（FDA） [15, 4, 9] 将这些时间序列信息视为连续统上的函数，这些函数本质上具有无限维。函数主成分分析（FPCA） [15, 4, 9] 是多元分析领域中主成分分析（PCA）对应的函数型方法。FPCA 是一种实用的降维工具，常作为许多函数分析方法 [3, 5] 的关键组成部分。它将整个时间序列视为单个函数样本，从而能够对时间序列数据进行无监督学习，获得低维表示。FPCA 有助于克服上述一些局限，但它们学习的是数据的线性表示，因此当底层映射较为复杂时，容易出现欠拟合。函数型数据分析中还有一个充分降维分支 [10, 11] ，它能够学习数据的非线性低维潜在表示，但这些方法属于有监督方法，因此适用范围有限。此外，在存在多个函数特征的情况下，这些方法要么无法使用，要么效果不佳。

深度学习方法在过去几年中变得非常流行。它们为学习多种类型数据的非线性表示提供了一些最先进的方法 [23, 7]。近年来，函数型数据中的深度学习 [18, 26, 25, 22, 17] 获得了很大发展。函数型自编码器（FAE） [8] 受 Rossi 等人启发，[19, 20], 提出了利用深度学习对函数型数据进行降维的方法。FAE 将自编码器应用于函数型环境。他们表明，在某些条件下，函数主成分分析（FPCA）是 FAE 的一个特例。尽管 FAE 已被证明优于许多最先进的方法 [8], ，但遗憾的是，它并未捕捉到函数型数据的真实本质，因为只有部分层具有权重函数，并且时间序列数据的低维潜在表示是标量，这具有一定的局限性。到目前为止，我们讨论的所有方法都存在某种结构性限制；据我们所知，现有工作要么致力于减少函数特征的数量，要么致力于减少观测这些曲线的时间点数量，但同时解决这两个问题的研究非常有限。

本文从函数型数据分析的视角，对时间序列问题中的降维进行建模。我们创新性地识别出函数特征之间的一般数学映射，并基于此提出一种双向非线性降维方法。具体而言，我们提出了双函数型自编码器（BFAE），这是将传统自编码器推广到函数型数据场景的一种新型方法。本文的贡献总结如下：

- 我们提出一种从多变量时间变量到其自身的一般函数映射，该映射将函数型自编码器（FAE）作为一种特殊情形涵盖在内。
- 我们提出一种新模型，将广为人知的神经网络自编码器推广到函数型数据场景，同时保留数据的函数特性，以解决时间序列中的降维问题。
- 我们阐释了该方法如何在双向（特征数量和时间点）降维的同时捕捉非线性关系，因此将其称为双函数型自编码器（BFAE）。
- 我们通过仿真实验和两个真实案例，证明了所提出的方法在学习时间序列低维非线性表示方面的有效性。

2 预备知识

2.1 符号说明和现有技术

本文旨在利用相关变量内部及变量之间的时间依赖关系，开发一种方法，以紧凑且高效的降维方式映射并表示多个时间序列变量（特征）。我们先介绍一些有用的符号，讨论现有技术和问题定义，然后详细介绍双函数型自编码器（BFAE）。假设对于 i^{th} （$i \in \{1, 2, ..., N\}$）个独立受试者，我们有 R 个特征，这些特征在紧凑的时间区间 $\mathcal{S} \subseteq$ R内连续记录。具体而言，第 i^{th} 个受试者的 r^{th} 特征的观测时间点由一个 $M_s^{(i,r)}$维向量S$^{(i,r)} = [S_1^{(i,r)}, ..., S_j^{(i,r)}, ..., S_{M_i}^{(i,r)}(, _s{}^r]^T)$ 表示，其中$M_s^{(i,r)}$表示时间序列中的时间点数量，且 $S_j^{(i,r)} \in \mathcal{S}$对应 $i = 1, ..., n$； $r = 1, ..., R$； $j = 1, ..., M_s^{(i,r)}$。相应的时间序列记为X$^{(i,r)} = [X_1^{(i,r)}, ..., X_j^{(i,r)}, ..., X_{M_i}^{(i,r)}(, _s{}^r]^T)$。下标 $M_s^{(i,r)}$表示测量时间戳可能因特征和受试者而异。为了适用主成分分析（PCA）和自编码器（AE）等现有方法，数据必须是规则时间序列，因此我们假设所有特征和受试者的时间点数量均为 M。观测数据可记为 $\{$X$^{(i,1)}, ...,$ X$^{(i,R)}\}_{i=1}^N$。接着，我们将 R 个时间特征串联为X$^{(i)} = [$X$^{(i,1)^T}, ...,$ X$^{(i,R)^T}]^T$。给定样本 $\{$X$^{(i)}\}_{i=1}^N$，问题可表述为一个无监督非线性多维函数表示学习问题，正式定义如下：

$$\mathbf{Z}^{(i)} = F(\mathbf{X}^{(i)}) \tag{1}$$

其中，$\mathbf{Z}^{(i)} = \{\mathbf{Z}^{(i,1)}, ...\mathbf{Z}^{(i,R')}\}_{i=1}^N$ 是时间序列$\mathbf{X}^{(i)}$ 的潜在表示，其中 $R' < R$。

这种问题表述存在一些缺点。在对数据进行预处理、使每个时间序列的时间点数量一致时，可能会引入偏差；而且，这些广泛使用的现有技术在假设时间序列数据为标量值时，解决时间序列数据降维问题也各有局限。这些多变量方法无法捕捉 $\mathbf{X}^{(i,r)}, r = 1, ...,$ R之间的时间依赖关系。

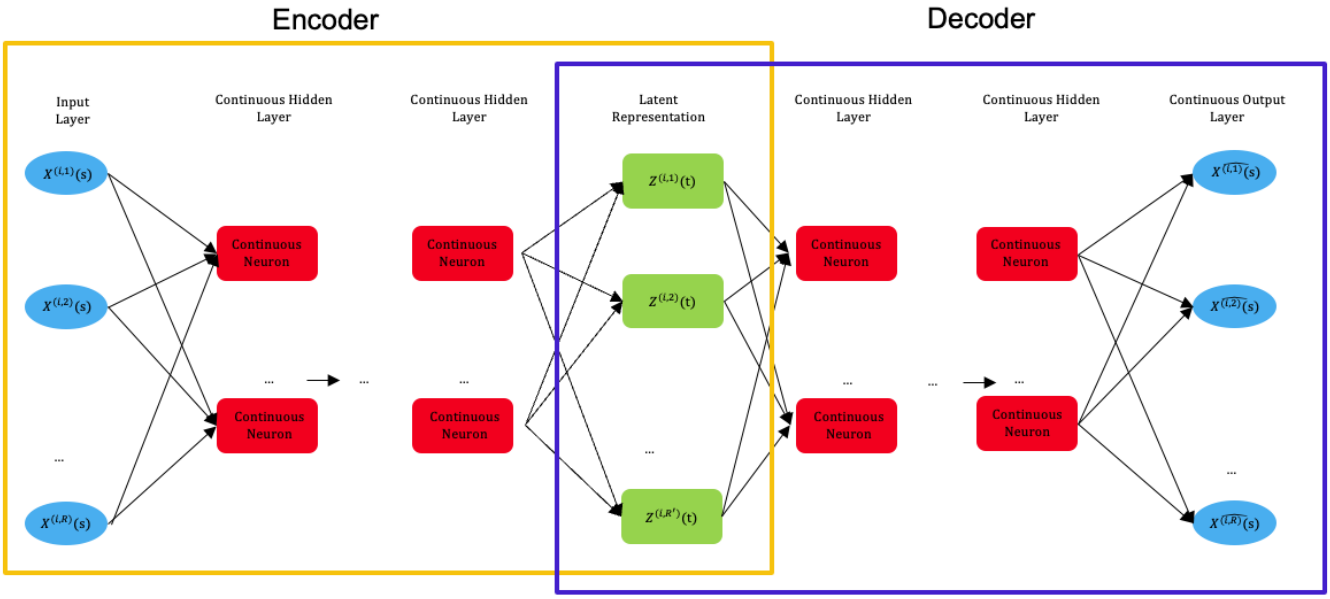


图 1：我们提出的双函数自编码器的总体架构

2.2 函数型数据分析与另一种表述

在本文前文中，我们从多元时间序列的角度探讨了降维问题。现在，我们来看看采用函数型数据分析（函数型数据分析）的一种替代性问题表述。在函数型数据分析中，我们从随机函数中学习，这些函数是在连续区间上动态变化的数据。函数型数据分析在许多领域都有广泛应用，例如时间序列、传感器数据、图像和空间数据 [14, 9]。我们可以利用函数型数据分析的这一能力来处理连续底层曲线 $X^{(i,r)}(s)$ $s \in \mathcal{S}$，这些曲线是在一个时间区间内的离散时间点上观测得到的。在函数分析中，输入或输出必须是函数曲线，或者二者皆是。对于给定的函数特征 $\{X^{(i,1)}(s).....,\ X^{(i,R)}(s)\ s \in \mathcal{S}\}_{i=1}^{N}$，我们寻找数据潜在表示的问题可表述如下：

$$\mathbf{Z}^{(i)}(t) = F(\mathbf{X}^{(i)}(s)). \tag{2}$$

其中，$\mathbf{Z}^{(i)}(t) = \{\mathbf{Z}^{(i,1)}(t)..., \mathbf{Z}^{(i,R')}(t)\ t \in \mathcal{S}\}_{i=1}^{N}$ 是时间序列$\mathbf{X}^{(i)(s)}$ 的潜在表示，其中 $R' < R$，且每个时间序列均在 M' 个时间点（$M' \leq M$）上观测得到。

随着近期该研究领域热度不断攀升，函数型数据分析在时间序列应用中的流行程度显而易见。正如 [28], 所示，函数型数据分析中的平滑性假设可能有所偏离，这使我们能够直接将不同的函数型数据分析技术应用于时间序列数据。此外，与随时间学习固定参数的常见序列模型（循环神经网络、长短期记忆网络）相比，函数型数据分析使我们能够学习随区间 $\mathcal{S}$ 变化的特征效应，并构建更高效的模型 [25]。在函数型领域中，我们有函数主成分分析 [6, 1]，它能够将信息编码到低维潜在空间中，但其映射是线性的。这可能成为一种限制因素，因为在许多现实应用中，多个函数特征之间不仅彼此关系复杂，在不同时间点之间也存在复杂关联。

人们对深度学习和函数型数据分析（FDA）的兴趣日益浓厚，促成了多个相继提出的深度函数模型 [19, 26, 16]，用于不同的分析任务。FAE [8] 是一种用于降维的深度学习方法，它将自编码器从向量空间推广到函数型环境。在这种网络中，只有第一层和最后一层能够借助 [19, 20], 给出的函数型神经元处理函数，其余网络则由传统神经元组成。因此，FAE 只能以标量形式给出函数型数据的潜在表示，这限制了它的应用。函数特征之间存在复杂关系的情况十分常见，而这些关系无法嵌入标量形式。因此，函数型数据的标量表示并不充分。

在下一节中，我们将提出一种非线性函数对函数自编码器，它利用全连接神经网络的优势，同时保留时间序列的函数特性。

3 双函数自编码器模型

3.1 我们的方法

本文的主要目标是学习一个映射 $\mathbf{f}(\cdot)$，将时间序列数据映射到低维潜在空间，如图 1 所示。为实现这一目标，我们采用自编码器和函数神经网络 [19, 16, 17] 的思想。自编码器（AE）是一种广受欢迎的降维技术，它借助编码器，能够从 R 维向量值输入空间 $\mathbf{R}^R$ （其中 $R' < R$）中学习 R' 维空间 $\mathbf{R}^{R'}$ 内的非线性潜在表示。

自编码器的解码器部分有助于将学习到的潜在表示映射回 R 维向量值输入空间 $\mathbf{R}^R$。自编码器的输入和输出相同，我们通过测量输出相对于输入的重构误差来进行学习。我们希望运用这一思想，但将其置于函数型环境中：在此环境下，编码器将 R 维函数空间映射到 R' 维函数空间，而解码器则将 R' 维函数空间映射回 R 维函数空间。我们采用 [16, 17] 中定义的连续神经元，以便在函数型环境中完成这一任务。连续神经元接收函数型输入并生成函数型输出，从而保留数据的函数型特性。

如图 1 所示，我们的双函数自编码器（BFAE）模型框架通过使用多个连续隐藏层，构建从一层到另一层的连续映射；这些隐藏层由多个连续神经元组成，其定义见 [16, 17]。图 1 展示了三种类型的层：输入层、连续隐藏层和连续输出层。输入层是第一层，用于接收函数特征；连续隐藏层由多个连续神经元组成；连续输出层则利用连续神经元重建输入函数特征。模型借助随时间（或其他连续域）连续变化的函数型权重进行学习。这些函数型权重包含单变量函数和双变量函数，也就是说，我们学习的不仅是函数，还有曲面。这里的核心思想是，在整个网络的重建过程中保留数据的函数特性。这样，在利用函数特征的领域信息和连续性的同时，就能以更丰富的结构将数据转换为低维潜在表示。我们将 l^{th} 连续隐藏层及其 r^{th} 连续神经元定义如下：

$$H_{(l)}^{(i,r)}(s) = \sigma\Big(b_{(l)}^{(r)}(s) + \sum_{j=1}^{J} \int w_{(l)}^{(r,j)}(s,t) H_{(l-1)}^{(i,j)}(t) dt\Big) \tag{3}$$

其中 $l = 1$、2、3，...，L，$H^{(i,r)}{}_{(0)}(s) = X^{(i,r)}(s)$、$H^{(i,r)}{}_{(L)}(s) = \widehat{X^{(i,r)}}(s)$、$b^{(r)}{}_{(l)}(s) \in \mathcal{L}^2(\mathcal{S})$ 是未知的单变量截距函数，$w^{(r,j)}{}_{(l)} \in \mathcal{L}^2(\mathcal{S} \times \mathcal{S})$ 是二元参数函数，对应于 l^{th} 连续隐藏层中的第 r^{th} 个连续神经元，其来源是（$l-1$）th 连续隐藏层中的第 j^{th} 个连续神经元；$\sigma(\cdot)$ 是标准激活函数。J 的值可视为每个连续隐藏层中的特征数量；对于输入层和连续输出层，该值固定为 R。对于其余连续隐藏层，我们可以将 J 调整为任意标量值。同样，连续神经元输出的观测时间点对于输入层和连续输出层是固定的，而对于其余连续隐藏层中的其他连续神经元，则可以进行调整。

现在我们来探讨网络的工作原理。我们使用公式 3 定义的连续神经元，从输入层逐步得到输入的降维形式，并将其表示在一个中间连续隐藏层中（例如，该层是 l'^{th} 层）。这是 BFAE 方法中的编码器部分，称为函数型编码器。我们可以在 l'^{th} 个连续隐藏层中获得输入的低维表示：$\mathsf{Z}^{(i)}(s) = H^{(i)}(l')(s)$，其中 $\mathsf{Z}^{(i)}(s) = \{\mathsf{Z}^{(i,1)}(s) ..., \mathsf{Z}^{(i,R')}(s)\ s \in \mathcal{S}\}_{i=1}^{N}$，且 $H^{(i)}(l')(s) = \{H_{l'}^{(i,1)}(\,)(s) ..., H^{(i,R')}(l')(s)\ s \in \mathcal{S}\}_{i=1}^{N}$，$R' < R$，函数 $\mathsf{Z}^{(i)}$ 在 $M' \leq M$ 个时间点处被观测到。我们从输入的这一潜在表示出发，利用连续隐藏层中的连续神经元，逐步映射到连续输出层，从而获得重构后的函数型输入值。这部分方法称为函数型解码器。因此，我们利用函数型编码器建立了从 R 维函数空间到 R' 维函数空间的映射，而函数型解码器则将 R' 维函数空间映射回 R 维函数空间。潜在表示 $\mathsf{Z}^{(i)}(s)$ 非常灵活，我们可以根据具体任务，将函数特征数量设为 R'，将观测到的时间点数量设为 M'。

为简单起见，假设图 1 的输入层有 R 个函数特征、L 个连续隐藏层，每层各有 J 条输入连接和 R 条输出连接。BFAE 本质上与自编码器相似，但对于函数型数据，我们必须定义连续隐藏层的数量、每个连续隐藏层中的连续神经元数量，以及每个连续神经元对应的函数被观测到的时间点数量。连续神经元所使用的激活函数为 ReLU、双曲正切、Sigmoid/逻辑函数或线性函数（连续输出层）。借助公式 3，网络的前向传播很直接。在反向传播阶段，我们直接利用函数曲面（二元）和函数截距（一元）进行学习。我们需要函数梯度来学习 BFAE 网络的函数参数。这些函数梯度衡量的是：当函数所依赖的另一个函数发生变化时，该函数随之发生的变化。

我们使用下面给出的函数梯度来优化网络的函数参数。我们采用的优化方法是传统的梯度下降。必要的假设和数学工具将在 [19, 26, 12] 中讨论。我们使用变分法中的 Fr´echet 导数来计算函数梯度。反向传播所需的偏导数如下：

$$
\begin{aligned}
\frac{\partial H_{(l)}^{(i,r)}}{\partial b_{(l)}^{(r)}}(s) &= \sigma'\Big(b_{(k)}^{(r)}(s) + \sum_{j=1}^{J}\int w_{(l)}^{(r,j)}(s,t)H_{(l-1)}^{(i,j)}(t)dt\Big) \\
\frac{\partial H_{(l)}^{(i,r)}}{\partial w_{(l)}^{(r,j)}}(s,t) &= \sigma'\Big(b_{(l)}^{(r)}(s) + \sum_{j'=1}^{J}\int w_{(l)}^{(r,j')}(s,t')H_{(l-1)}^{(i,j')}(t')dt'\Big) \\
&\quad \times \int \frac{\partial}{\partial w_{(l)}^{(r,j)}(s,t)} w_{(l)}^{(r,j)}(s,t)H_{(l-1)}^{(i,j)}(t)dt \\
&= \sigma'\Big(b_{(l)}^{(r)}(s) + \sum_{j'=1}^{J}\int w_{(l)}^{(r,j')}(s,t')H_{(l-1)}^{(i,j')}(t')dt'\Big) \\
&\quad \times H_{(l-1)}^{(i,j)}(t)
\end{aligned}
\tag{4}
$$

其中 $l = 1, 2, 3, \ldots\ldots, L$, $H^{(i,r)}{}_{(0)}(s) = X^{(i,r)}(s)$, $H^{(i,r)}{}_{(L)}(s) = \widehat{X^{(i,r)}}(s)$ 和 $\sigma'(\cdot)$ 表示 $\sigma(\cdot)$ 的一阶导数。在反向传播阶段，我们从连续输出层向输入层反向遍历网络，并计算连续神经元相对于双变量函数权重和函数截距的偏导数。我们要最小化的损失函数为 $\mathcal{L}(\theta)$，其中 θ 是 BFAE 网络中所有函数参数的集合。损失函数如下:

$$
\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{r=1}^{R}\int \Big(X^{(i,r)}(s) - \widehat{X^{(i,r)}(s)}\Big)^2 dt \tag{5}
$$

连续隐藏层的数量、每个连续隐藏层中的连续神经元数量，以及每个连续神经元对应的观测时间点数量，都可以视为超参数。我们还可以调整方法，以适应不规则函数型数据或标量变量。所有这些都为我们的方法提供了很大的灵活性，使其能够处理不同类型的问题，并根据预测、分类、聚类、预测等不同分析任务的下游流程进行调整。

3.2 我们的方法与现有技术的联系

一些现有的降维方法可以表示为 BFAE 的特例。众所周知，采用线性激活函数的单层自编码器（AE）与主成分分析（PCA）类似。同样，我们可以调整模型参数，使其实现函数主成分分析（FPCA）的功能。当模型满足以下条件时，FPCA 是 BFAE 的特例：采用线性激活函数、只有一个连续隐藏层，且函数型权重受到正交归一约束。《函数型自编码器》（FAE） [8] 论文讨论了 AE 如何成为 FAE 的特例，因为 FAE 将 AE 中的标量权重和内积替换为函数型权重和内积。同理，FAE 也是我们的方法 BFAE 的特例。如果我们指定模型中的函数型编码器学习函数特征的标量（$M' = 1$）潜在表示，那么我们的方法本质上就实现了 FAE。因此，我们的方法是对 PCA、FPCA、AE 和 FAE 等大多数现有方法的推广。

4 数值实验

本节将所提出的模型应用于多个模拟场景和两个实际问题。我们将结果与几种最先进的方法进行比较，例如主成分分析、AE 和函数主成分分析，以展示 BFAE 的有效性。通过结果，我们展示了以下目标：1）我们的方法利用推导出的函数梯度进行了有效训练；2）BFAE 能够学习函数特征之间的关系；3）我们通过优于其他方法展示了所提方法的有效性。

4.1 模拟

在第一个实验中，我们考虑一个预测函数（即 $R = 1$），该函数在稠密的规则网格上观测。我们从均值为 0、协方差如下所示的高斯过程中生成 $N = 100$, 1000 条独立同分布随机曲线 $\Big\{X^{(1)}(t) \cdots, X^{(n)}(t)\Big\}$:

$$
C_X(t,s) = \frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}|t-s|}{\rho}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}|t-s|}{\rho}\right). \tag{6}
$$

其中，这是 Mat´ern 协方差函数， K_ν 是第二类修正贝塞尔函数。$\rho = 0.5, \nu = 5/2$ 和 $\sigma^2 = 1.$ 的取值如下。这些曲线在区间 $[0, 1]$ 上等间距的时间点处实现，其中 $M = 50$,共有 250 个时间点。我们特意选择了 $N < M$ 的情况，以检验它是否会给任何方法带来问题。

使用高斯过程生成曲线后，我们向其中添加一些随机噪声（ϵ_i（t））。结果取 100 次模拟的平均值，并将 100 个样本划分为 80 个训练样本和 20 个测试样本。对于 1000 个样本的情况，也采用相同比例进行划分。我们报告由公式（7）给出的函数

M	50	250	50	250
方法/N	100		1000	
PCA	0.483	0.418	0.469	0.398
AE	0.409	0.443	0.390	0.406
FPCA	0.390	0.351	0.356	0.343
BFAE	0.301	0.265	0.102	0.098
基函数自编码器（M'）	0.331	0.274	0.117	0.104

表 1：不同方法捕捉单个时间序列函数的均方根误差比较。

特征重建的均方根误差（RMSE）；对于标量方法，我们则只报告标准 RMSE，不考虑时间维度。在第一个实验中，我们的方法采用连续隐藏层（$L=1$，3）的多种组合，且仅使用一个连续神经元（$J=1$）。连续神经元中潜在表示的网格（s）值为 M 或 $M'=M/5$。我们使 AE 网络与 BFAE 相似；对于主成分分析和函数主成分分析，则遵循 99%方差解释规则。请注意，我们可以将 M' 设为 $(1, M)$ 范围内的任意标量值，但我们选择这个特定值（$M'=M/5$）来展示我们方法的灵活性。

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\sum_{r=1}^{R}\int\left(X^{(i,r)}(s)-\widehat{X^{(i,r)}(s)}\right)^2 dt} \tag{7}$$

从表 I 可以看出，主成分分析和自编码器在时间曲线上表现相似。函数主成分分析的性能有所提升，但无论样本量（N）和时间点数量（M）如何，我们的方法表现都最好。随着样本量和时间点数量增加，性能差异也会扩大。由于我们的方法是一种深度学习模型，拥有更多信息时学习效果更好。当我们将观测时间点数量设为 M'时，由于将真实数据限制在更小的潜在空间中，我们的方法性能有所下降。但即使在这种情况下，与使用 50 或 250 个时间点相比，改用 10 或 50 个时间点表示真实曲线，我们的方法仍优于所有其他方法。

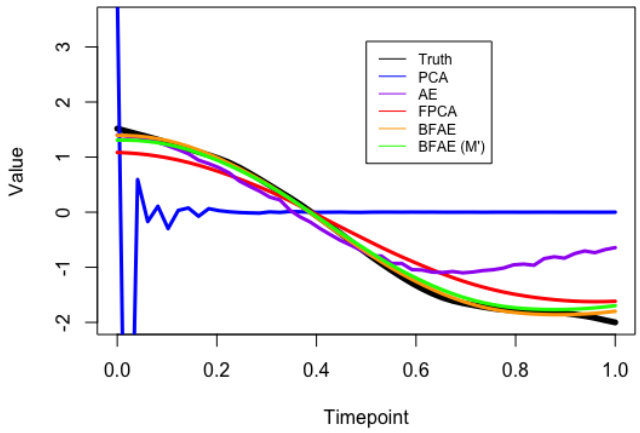


图 2：当 $N=1000$、$R=1$ 和 $M=50$ 时，不同方法的曲线重建比较

图 2 展示了在 $N=1000$、$R=1$ 和 $M=50$ 的模拟设置下，采用不同方法从随机样本中重建的曲线。与表 I 类似，主成分分析的表现较差，重建曲线的形状与真实曲线相差很大。主成分分析是一种线性函数方法，无法捕捉数据中的时间信息。自编码器和函数主成分分析的表现优于主成分分析，能够重建出大致形状，但与真实曲线仍相距甚远。无论观测到的时间点数量如何，我们的方法 BFAE 都能很好地重建真实曲线，并在整个时间区间内准确捕捉曲线形状。

我们将第一个实验扩展到包含更多函数特征。在第二个实验中，我们将特征数量增加到 $R=10$。我们采用相同的流程，针对 N 和 M 的同一组取值生成这 10 条曲线。以下讨论的结果是在加入随机噪声后，使用相同的训练集和测试集划分进行 100 次模拟所得的平均值。对于我们的方法，我们考虑连续隐藏层数量的多种组合（$L=1$、3）和连续神经元数量的多种组合（$J=1$、2、4），并设定 $R'=4$。连续神经元中潜在表示的网格（s）值为 M 或 $M'=M/5$。

表 II 显示，函数特征的增加会导致主成分分析的性能下降。尽管另外两种方法的表现优于主成分分析，但无论样本量（N）如何，我们的方法都表现最佳

M	50	250	50	250
方法/N	100		1000	
PCA	1.527	1.525	1.518	1.518
AE	0.471	0.490	0.489	0.484
FPCA	0.386	0.417	0.365	0.404
BFAE	0.328	0.292	0.109	0.132
基函数自编码器（M’）	0.359	0.314	0.147	0.169

表 2：比较不同方法在捕捉多个（$R = 10$）时间序列函数时的均方根误差。

方法/数据	音素	阿德莱德
PCA	14.960	18.633
AE	1.790	2.354
FPCA	1.242	0.636
BFAE	1.126	0.581
基函数自编码器（M’）	1.128	0.615

表 3：比　　较均方根误差　不同方法捕捉预测　　真实数据的函数　　.

以及时间点的数量（M）。随着样本量和时间点数量的增加，性能差异也随之扩大。我们观察到，与上一张表相比，由于增加了更多特征，误差略有增加。将时间点数量减少到 M' 的情况也有类似表现。

4.2 真实数据应用

我们考察两个真实数据集。第一个是 TIMIT 的语音识别数据（可在 http://statweb.stanford.edu/tibs/Ele-mStatLearn/ 获取），其中包含不同音素的语音信号。音频信息十分丰富，但采集频率极高，因此需要较大的存储容量。在工业中，这类数据的低维潜在表示会非常有用。另一个例子涉及澳大利亚阿德莱德市电力需求与温度之间的关系。由于电力存储成本高昂，这类数据十分重要；了解温度对需求的影响，有助于制定基于数据的运营措施。

对于语音识别数据，我们按照 [10, 16], 的方式设置实验，使用两个音素（响应变量）的语音信号（R=1），其转写如下: “aa”是“dark”中的元音，“ao”是“water”中的第一个元音。对语音信号进行降维后，我们借助函数线性模型构建分类器模型。图 3 显示这两个组难以区分；图中展示了通过 150 个点的对数周期图计算得到的两个音素曲线（$M = 150$）。每个音素的信息均记录在 150 个点上，减少这类数据的维度大有裨益，尤其是考虑到这种低维表示有望扩展至数量多得多的语音信号。我们共有 800 个函数样本，将其分为 640 个训练样本和 160 个测试样本。

阿德莱德数据（可在 R 软件的 fds 包中获取）包含从 1997年7月6日到 2007年3月31日（508 周）以半小时为间隔每日测量的温度和电力需求记录（$M = 48$）。我们将每周的每一天视为一个特征（R=7），并尝试使用函数线性模型（FLM）[4, 9, 14] 来刻画温度与需求之间的关系。在使用函数型数据分析对这种关系建模之前，我们先用 BFAE 将温度曲线表示为紧凑的低维形式。我们将这些数据划分为 400 个训练样本和 108 个测试样本。图 4 展示了全部 508 周的半小时温度和电力需求（兆瓦）数据。从图像中可以看出，每周各天的温度曲线遵循相似的模式；学习这种模式可以在尽可能少损失信息的前提下降低维度。

从表 III 可以看出，主成分分析在降低两个真实数据集的信息维度方面仍然存在困难。自编码器和函数主成分分析的表现优于主成分分析，而 BFAE 的表现最佳。对于音素数据，我们甚至将点数降至 $M' = 30$，BFAE 的表现仍优于其他方法。BFAE 在时间点为 M 与 M' 时的表现差异可以忽略不计，这表明其低维表示包含了丰富的信息。对于阿德莱德数据，我们将特征数降至 4（$R' = 4$），并将时间点数降至 $M' = 12$；采用时间点为 M 的设置时，我们的方法取得了最佳结果。图 4 和表 III 的结果验证了我们减少温度特征的直觉。

建模结果见表 IV。我们为音素数据构建了一个分类器，用于预测语音信号属于“aa”还是“ao”。对于阿德莱德数据，我们将一周 7 天的温度信息映射到同一周 7 天的电力需求进行预测。这两个模型均借助原始方法数据以及使用 BFAE 学习得到的低维潜在表示构建。由于这里的目标是展示我们的方法能够通过捕捉信号，取得至少与原始方法数据一样好的建模结果，因此我们忽略了其他方法。从表 IV 可以看出，我们的方法不仅具有良好的建模效果，性能也与原始方法数据相当，并且由于包含的噪声更少，过拟合倾向更低。在两种情况下，随着我们将时间点减少到 M'，误差都会增加，但性能仍然具有竞争力。

总体而言，与原始数据相比，我们的方法在将信息压缩到低维潜在空间的同时，仍能在不同任务的建模中保持有竞争力的性能，在这方面表现最佳。我们能够表示

真实数据	Data	原始方法	BFAE	BFAE（M'）
音素	训练集	0.175	0.185	0.205
	test	0.200	0.190	0.210
阿德莱德	训练集	188.363	164.105	199.758
	test	220.739	184.518	215.294

表 4：不同方法对响应变量建模的误差比较（音素的分类误差和阿德莱德的均方根误差）。

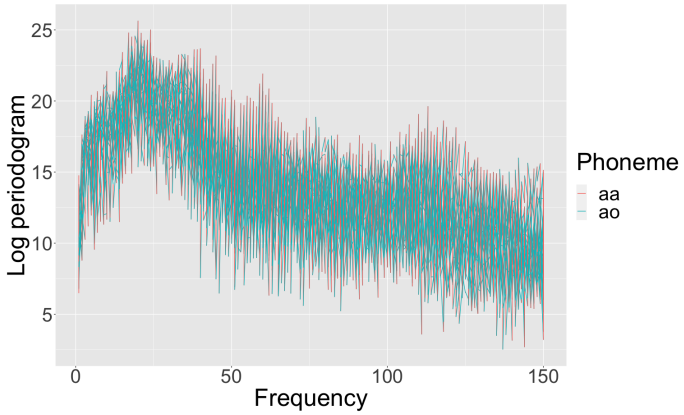


图 3：音素数据的语音信号曲线

信息，并以紧凑且富含数据的方式进行呈现，同时还能将其重构回原始尺度，而不会造成太多信息损失。

5 结论

本文提出了一种用于多元时间序列降维的新型模型。我们的方法通过减少特征数量以及观测时间序列的时间点数量，实现双向降维。所提出的双函数自编码器（BFAE）利用函数型编码器将输入压缩为低维潜在表示，并通过函数型解码器重建信息。与现有方法相比，我们的方法具有诸多优势，包括能够灵活地表示时间序列数据、捕捉特征间随时间变化的相关性、处理不同类型的数据以及捕捉非线性关系。除了取得优异的模拟结果外，我们还通过两个真实案例，将我们的方法与现有技术中的几种常见做法进行了比较，展示了该方法的出色表现。我们的方法在重建中产生的误差更小，并且对数据的建模效果与原始信息相当。我们预计，该模型将在各种实际问题中得到广泛应用，例如减少大量时间序列数据的传输负载、有效存储大量时间序列信息，以及在保持性能的同时降低各类分析任务的计算成本。

参考文献

[1] Daniel Backenroth、Jeff Goldsmith、Michelle D. Harran、Juan C. Cortes、John W. Krakauer 和 Tomoko Kitago。使用异方差函数型主成分分析对运动学习进行建模。 *Journal of the American Statistical Association* 113 523 1003 1015, 2018 30416231 , ( ): –。

PMID: 。[2] Wei Chen、Hythem Sidky 和 Andrew L. Ferguson。时间滞后自编码器在动力系统中发现慢模态的能力与局限。 *The Journal of Chemical Physics*，151(6)：064123，2019年。[3] Jeng-Min Chiou、Yu-Ting Chen 和 Ya-Fang Yang。多元函数型主成分分析：一种归一化方法。 *Statistica Sinica*，第1571–1596页，2014年。[4] F. Ferraty 和 Y. Romain。 *The Oxford Handbook of Functional Data Analysis*。牛津大学出版社，2011年1月。[5] Clara Happ 和 Sonja Greven。针对不同（维度）定义域上观测数据的多元函数型主成分分析。 *Journal of the American Statistical Association*, 113(522)：649–659，2018年。[6] Clara Happ 和 Sonja Greven。针对不同（维度）定义域上观测数据的多元函数型主成分分析。 *Journal of the American Statistical Association*，113(522)：649–659，2018年。[7] G. E. Hinton 和 R. R. Salakhutdinov。利用神经网络降低数据维度。*Science*，313 5786 504 507，2006年 ( )：–。

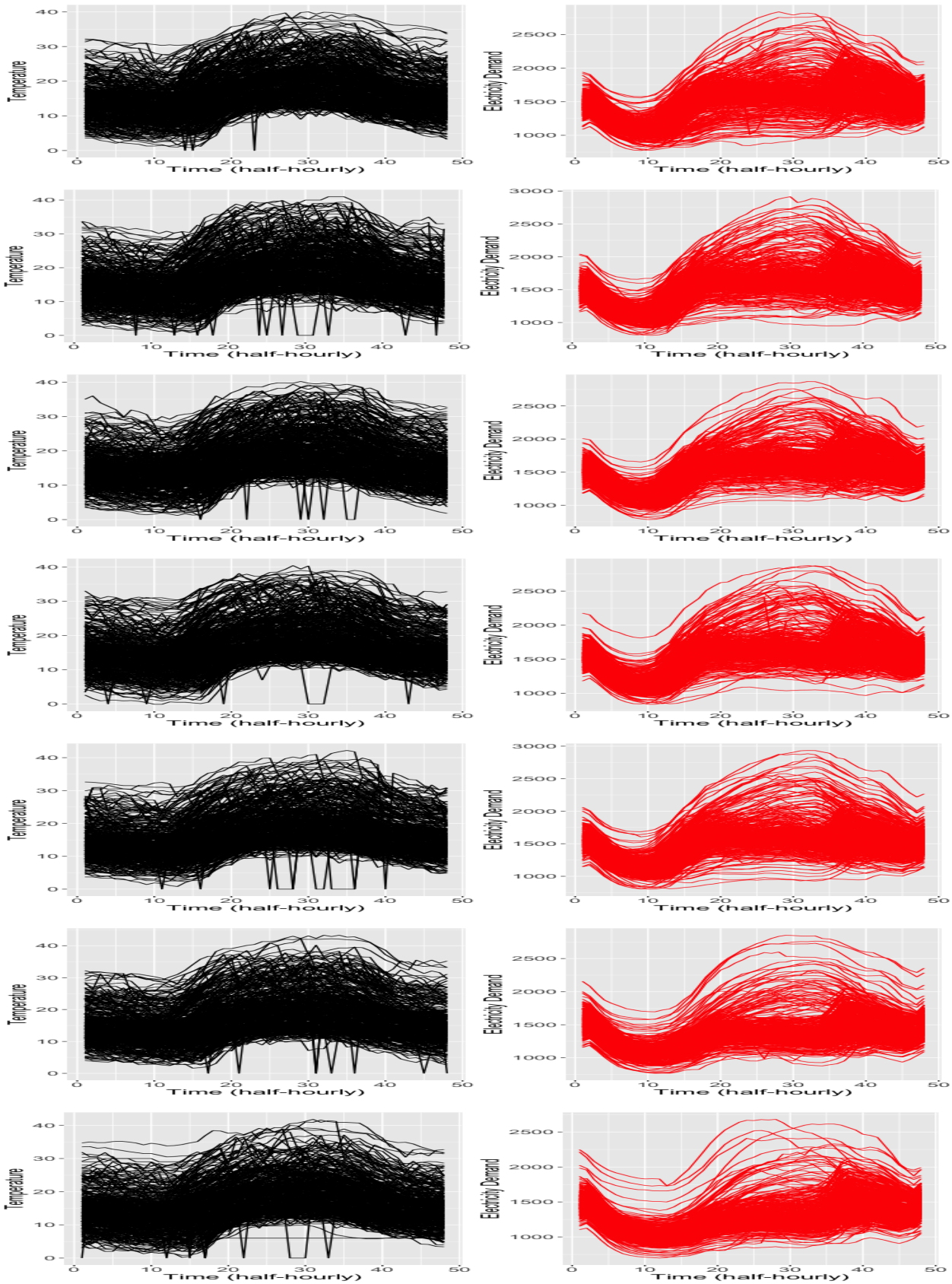


图 4：日曲线　　阿德莱德温度和电力需求的日曲线（从上到下依次为星期一至星期日）　　de

[8] Tsung-Yu Hsieh、Yiwei Sun、Suhang Wang 和 Vasant Honavar。 *Functional Autoencoders for Functional Data Representation Learning* 666 674 2021 ，第 –

页。arXiv，。[9] Piotr Kokoszka 和 Matthew Reimherr。 *Introduction to Functional Data Analysis*。纽约：Chapman and Hall/CRC，2018年。[10] Bing Li 和 Jun Song。函数型数据的非线性充分降维。 *Annals of Statistics*，45(3):1059–1095，2017年6月。版权：版权所有 2017年 爱思唯尔，保留所有权利。[11] Bing Li 和 Jun Song。基于弱条件矩的函数型数据降维。 *The Annals of Statistics* 50 1 107 128, 2022 ，( ):

– 。[12] Peter J. Olver。变分法导论。载于 *Introduction to the Calculus of Variations*，2014 年。[13] Stefan Petscharnig 、Mathias Lux 和 Savvas Chatzichristofis。利用深度学习和自编码器对图像特征进行降维。载于 *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*，CBMI '17，美国纽约州纽约市，2017年。美国计算机协会。[14] James O Ramsay。 *Functional data analysis*。Wiley Online Library，2006年。[15] J.O. Ramsay 和 B.W. Silverman。 *Functional Data Analysis*。施普林格统计学丛书。施普林格，1997年。[16] Aniruddha Rajendra Rao 和 Matthew Reimherr。现代非线性函数对函数回归，2021年。[17] Aniruddha Rajendra Rao 和 Matthew Reimherr。使用神经网络进行非线性函数建模，2021年。[18] Aniruddha Rajendra Rao、Qiyao Wang、Haiyan Wang、Hamed Khorasgani 和 Chetan Gupta。时空函数型神经网络。 *arXiv preprint arXiv:2009.05665*，2020年。[19] Fabrice Rossi、Brieuc Conan-Guez 和 Fran.cois Fleuret。使用多层感知机进行函数型数据分析。载于 *Proceedings of IJCNN*，第2843–2848页。Citeseer，2002年。[20] Fabrice Rossi、Nicolas Delannay、Brieuc Conan-Guez 和 Michel Verleysen。神经网络中的函数型数据表示。 *Neurocomputing*，64:183–210，2005年。[21] Nitish Srivastava、Elman Mansimov 和 Ruslan Salakhutdinov。使用长短期记忆网络无监督学习视频表示。载于 *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* 15, 843 852 2015 ，

ICML' 页码 – 。JMLR.org，。[22] Barinder Thind、Kevin Multani 和 Jiguo Cao。函数型输入的深度学习，2020年。[23] Michael Tschannen、Olivier Frederic Bachem 和 Mario Lu˘ci´c。基于自编码器的表示学习最新进展。载于 *Bayesian Deep Learning Workshop, NeurIPS*，2018年。[24] Laurens van der Maaten、Eric Postma 和 H. Herik。降维：比较综述。 *Journal of Machine Learning Research - JMLR* 10 01 2007 ， ，

。[25] Qiyao Wang 、 Haiyan Wang 、 Chetan Gupta 、 Aniruddha Rajendra Rao 和 Hamed Khorasgani。用于时间序列数据回归的非线性函数对函数模型。载于 *2020 IEEE International Conference on Big Data (Big Data)*，第232–239页，2020年。[26] Qiyao Wang、Shuai Zheng、Ahmed Farahat、Susumu Serita、Takashi Saeki 和 Chetan Gupta。用于稀疏函数型数据的多层感知机。载于 *2019 International Joint Conference on Neural Networks (IJCNN)*，第1–10页。IEEE，2019年。[27] Yasi Wang、Hongxun Yao 和 Sicheng Zhao。基于自编码器的降维。 *Neurocomputing*，184:232–242，2016年。RoLoD：用于计算机视觉的鲁棒局部描述符，2014年。[28] Fang Yao、Hans-Georg M ¨ *Journal of the American Statistical Association* 100 470 577 590, 2005

uller 和 Jane-Ling Wang。稀疏纵向数据的函数型数据分析。， ( ): – 。