

# 基于卷积神经网络的短时傅里叶变换

顾亚鑫, 武国宁

中国石油大学北京理学院, 北京

收稿日期: 2024年1月25日; 录用日期: 2024年2月22日; 发布日期: 2024年2月29日

## 摘要

本研究提出了一种基于卷积神经网络(CNN)的短时傅里叶变换方法, 用于揭示非稳态信号的频谱随时间的变化规律。我们设计了一个卷积神经网络, 采用双层结构, 并随机初始化网络权重系数, 不包含偏置系数。输入数据为随机生成的一维信号, 其傅里叶变换作为标签数据。通过使用平方误差作为损失函数, 并运用梯度下降法对网络进行训练, 网络逐渐学得输入信号到其傅里叶变换的映射规则。同时, 我们观察到网络权重系数在迭代过程中逐渐逼近傅里叶变换的核函数。基于学到的核函数, 我们进行了信号的时频分析。数值试验结果表明, 以通过训练获得的核函数作为基函数的短时傅里叶变换能取得与传统窗口傅里叶变换相一致的结果, 证明了该方法的有效性。这一基于卷积神经网络的短时傅里叶变换方法为处理非稳态信号提供了一种新颖而有效的途径。

## 关键词

短时傅里叶变换, 卷积神经网络, 时频分析, 非稳态

# Short-Time Fourier Transform Based on Convolutional Neural Network

Yaxin Gu, Guoning Wu

College of Science, China University of Petroleum Beijing, Beijing

Received: Jan. 25<sup>th</sup>, 2024; accepted: Feb. 22<sup>nd</sup>, 2024; published: Feb. 29<sup>th</sup>, 2024

## Abstract

This study proposes a windowed Fourier transform method based on the Convolutional Neural Network (CNN) to reveal the temporal evolution of the spectrum of non-stationary signals. We designed a CNN with a dual-layer structure, randomly initialized network weight coefficients, and no bias terms. The input data consists of randomly generated one-dimensional signals, with their Fourier transforms serving as label data. Using the mean square error as the loss function and ap-

plying gradient descent for network training, the network gradually learns the mapping rules from input signals to their Fourier transforms. Simultaneously, we observed that the network weight coefficients gradually approximate the Fourier transform kernel during the iterative process. Based on the learned kernel function, we conducted a time-frequency analysis of signals. Numerical experimental results demonstrate that the windowed Fourier transform obtained through the learned kernel function as basis functions achieve consistent results with the traditional windowed Fourier transform, confirming the effectiveness of the proposed method. This CNN-based windowed Fourier transform method provides a novel and effective approach for processing non-stationary signals.

## Keywords

Short-Time Fourier Transform, Convolutional Neural Network, Time-Frequency Analysis, Non-Stationary

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

傅里叶变换将信号从时间域变换到频率域,是信号处理和微分方程求解的重要方法[1] [2]。离散傅里叶变换(DFT)作为一种特殊的线性变换,以揭示信号中不同频率成分的存在和强度,其核函数构成了范德蒙满秩矩阵[3]。尽管离散傅里叶变换运算量大,限制了其广泛应用[4],但随着快速傅里叶变换(FFT)的引入,傅里叶变换在实际中变得更加普及,成为信号处理领域中不可或缺的工具[5]。傅里叶变换适用于分析平稳信号的频谱特征,对于非平稳信号,基于傅里叶变换的短时傅里叶变换通过加窗,分段时移对信号进行频谱分析[6],将一维信号分解为二维时频谱,能够有效表征信号的频谱随时间的变化规律,因此成为非稳态信号处理的重要方法之一[7] [8]。

神经网络作为一种模仿生物神经网络结构和功能的数学模型,具备对函数进行近似表示的能力。典型的神经网络由多层神经元结构组成,每个神经元拥有输入和输出[9]。其自适应性学习能力使得神经网络能够根据给定的输入数据和标签数据学得输入和输出之间的关系。通过大量数据的训练,复杂的神经网络甚至能够逼近无限维空间到无限维空间的映射关系[10]。神经网络强大的学习能力和丰富的特征表示使其在自然语言处理、图像分类和识别、目标跟踪、地震信号噪声压制和地震参数反演等领域得到了广泛应用[11] [12] [13] [14]。其在这些领域的成功运用为传统问题注入了新的发展动力。

近些年,学者们将快速傅里叶变换成功应用于卷积神经网络领域,以来提高算法的速度[15] [16] [17] [18]。其中,一些学者将快速傅里叶变换应用于对抗神经网络,以加速训练过程并减轻梯度消失和爆炸的相关问题[19] [20]。此外,傅里叶变换还被用于近似网络中的全连接线性层,以减小算法的复杂性[21] [22]。傅里叶变换在深度学习中的成功运用凸显了该方法的重要性。尽管傅里叶变换在神经网络领域的应用较多[23] [24],但以神经网络为基础对傅里叶变换进行研究相对较少。本文设计了一个双层的全连接网络,其中不设置偏执系数,初始权重采用服从正态分布的随机数。此外,选择线性激活函数。建立网络拓扑结构后,我们随机生成一系列信号作为网络的输入数据,对这些随机信号进行快速傅里叶变换,得到相应的标签数据。利用标签数据对网络进行训练,使损失函数下降到预定的阈值。在训练过程中,网络学习输入数据到其标签数据的映射规则,从而使得该网络的权重系数逼近傅里叶变换的核函数。通过训练

得到核函数后, 我们能够通过将数据输入神经网络, 得到相应的输出数据, 即输入数据的傅里叶变换。最后, 我们将通过训练获得的核函数应用于短时傅里叶变换, 以分析信号的频谱随时间的变换规律。数值实验表明, 该方法能够取得和常规短时傅里叶变换相一致的结果。本文提出的方法是将神经网络应用到傅里叶变换的一次尝试。

论文结构安排如下: 第一部分为理论部分, 阐述基于神经网络的傅里叶变换的实现过程; 第二部分为网络结构及参数设置的介绍; 第三部分为实验部分, 主要包括模拟数据、实际地震数据实验; 最后一部分为结果分析和展望。

## 2. 理论原理

离散傅里叶变换是信号分析的常用方法之一, 对于有限长离散序列  $[x_0, x_1, \dots, x_{N-1}]$ , 其傅里叶变换为:

$$X_k = \sum_{i=0}^{N-1} x_i e^{-j2\pi \frac{ik}{N}} \quad (1)$$

这里  $j^2 = -1$ ,  $N$  为信号的长度。公式(1)也可写为矩阵形式:

$$X_k = [x_0, x_1, \dots, x_{N-1}] \begin{bmatrix} e^{-j2\pi \frac{0k}{N}} \\ e^{-j2\pi \frac{1k}{N}} \\ e^{-j2\pi \frac{2k}{N}} \\ \vdots \\ e^{-j2\pi \frac{(N-1)k}{N}} \end{bmatrix} \quad (2)$$

从而有:

$$[X_0, X_1, \dots, X_{N-1}] = [x_0, x_1, \dots, x_{N-1}] \begin{bmatrix} e^{-j2\pi \frac{0}{N}} & e^{-j2\pi \frac{0}{N}} & \dots & e^{-j2\pi \frac{0}{N}} \\ e^{-j2\pi \frac{0}{N}} & e^{-j2\pi \frac{1}{N}} & \dots & e^{-j2\pi \frac{1(N-1)}{N}} \\ e^{-j2\pi \frac{0}{N}} & e^{-j2\pi \frac{2}{N}} & \dots & e^{-j2\pi \frac{2(N-1)}{N}} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-j2\pi \frac{0}{N}} & e^{-j2\pi \frac{(N-1)}{N}} & \dots & e^{-j2\pi \frac{(N-1)^2}{N}} \end{bmatrix} \quad (3)$$

若令  $\mathbf{W}$  表示公式 (3) 中的傅里叶变换矩阵, 则公式(3)可表示为:

$$[X_0, X_1, \dots, X_{N-1}] = [x_0, x_1, \dots, x_{N-1}] \mathbf{W} \quad (4)$$

傅里叶变换作为一种特殊的线性变换, 可以采用双层神经网络来表示(如图 1 所示), 其中  $[x_0, x_1, \dots, x_{N-1}]$  为网络的输入,  $[X_0, X_1, \dots, X_{N-1}]$  为网络的输出。该网络无需设置神经元的偏执系数, 激活函数为线性函数  $\mathbf{y} = \mathbf{x}$ 。由于神经网络的权重为实数, 所以对于公式(3)做进一步处理。由于  $e^{-j2\pi \frac{ik}{N}} = \cos 2\pi \frac{ik}{N} - j \sin 2\pi \frac{ik}{N}$ , 所以

$$\text{Real}[X_k] + j\text{Imag}[X_k] = \sum_{i=0}^{N-1} x_i \cos 2\pi \frac{ik}{N} + (-j) \sum_{i=0}^{N-1} x_i \sin 2\pi \frac{ik}{N} \quad (5)$$

这里  $\text{Real}[X_k]$  表示  $X_k$  的实部,  $\text{Imag}[X_k]$  表示  $X_k$  的虚部。由(5)式, 得到傅里叶变换另可写为:

$$\bar{\mathbf{X}} = \mathbf{x} [\mathbf{W}_{11}, \mathbf{W}_{12}] \quad (6)$$

其中:  $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]$ ,

$$\bar{\mathbf{X}} = [\text{Real}[X_0], \text{Real}[X_1], \dots, \text{Real}[X_{N-1}], \text{Imag}[X_0], \text{Imag}[X_1], \dots, \text{Imag}[X_{N-1}]],$$

$$\mathbf{W}_{11} = \begin{bmatrix} \cos \frac{0}{N} & \cos \frac{0}{N} & \cdots & \cos \frac{0}{N} \\ \cos \frac{0}{N} & \cos \frac{2\pi}{N} & \cdots & \cos \frac{2\pi(N-1)}{N} \\ \cos \frac{0}{N} & \cos \frac{2\pi}{N} 2 & \cdots & \cos \frac{2\pi(N-1)}{N} 2 \\ \vdots & \vdots & \ddots & \vdots \\ \cos \frac{0}{N} & \cos \frac{2\pi(N-1)}{N} & \cdots & \cos \frac{2\pi(N-1)^2}{N} \end{bmatrix},$$

$$\mathbf{W}_{12} = \begin{bmatrix} \sin \frac{0}{N} & \sin \frac{0}{N} & \cdots & \sin \frac{0}{N} \\ \sin \frac{0}{N} & \sin \frac{2\pi}{N} & \cdots & \sin \frac{2\pi(N-1)}{N} \\ \sin \frac{0}{N} & \sin \frac{2\pi}{N} 2 & \cdots & \sin \frac{2\pi(N-1)}{N} 2 \\ \vdots & \vdots & \ddots & \vdots \\ \sin \frac{0}{N} & \sin \frac{2\pi(N-1)}{N} & \cdots & \sin \frac{2\pi(N-1)^2}{N} \end{bmatrix},$$

为了便于求傅里叶逆变换, 进一步作如下拓展

$$\bar{\mathbf{X}} = [\mathbf{x}, \mathbf{O}] \begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ -\mathbf{W}_{12} & \mathbf{W}_{11} \end{bmatrix} \quad (7)$$

其中  $\mathbf{O} = [0, 0, \dots, 0]$  为零矩阵, 其维数为  $N \times N$ 。  $-\mathbf{W}_{12}$  表示  $\mathbf{W}_{12}$  矩阵的负矩阵, 即  $\mathbf{W}_{12}$  中的每一个元素取相反数。  $\begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ -\mathbf{W}_{12} & \mathbf{W}_{11} \end{bmatrix}$  为傅里叶变换的核函数。

本文设计了一个双层的全连接网络结构, 通过训练数据及其标签学习该核函数。

### 3. 网络结构及其参数设置

常规傅里叶变换给定输入和运算规则(核函数)得到信号的傅里叶变换。基于卷积神经网络的傅里叶变换给定输入数据和标签数据, 设置合适的网络拓扑结构和损失函数, 可以通过梯度下降法学习从输入数据到标签数据的映射规则。

本文设计的网络为两层的全连接神经网络(如图 1 所示)。假设输入长度为  $N$  的实信号  $\mathbf{x}$ , 在其后面补充  $N$  个零, 得到长度为  $2N$  的输入信号  $\bar{\mathbf{x}} = [\mathbf{x}, \mathbf{O}]$ , 扩展后的信号  $\bar{\mathbf{x}}$  作为网络的输入信号。对实信号  $\mathbf{x}$  进行傅里叶变换得到  $\mathbf{X}$ , 将  $\mathbf{X}$  中每个元素的实部作为前  $N$  个元素,  $\mathbf{X}$  中每个元素的虚部作为后  $N$  个元素, 组成长度为  $2N$  的向量  $\bar{\mathbf{X}}$ , 该向量作为网络的标签数据。网络不设置偏执系数, 激活函数为  $y = x$ 。损失函数为平方误差函数

$$L(\bar{\mathbf{x}}, \bar{\mathbf{X}}) = \|\mathbf{F}(\bar{\mathbf{x}}) - \bar{\mathbf{X}}\|_2^2 \quad (8)$$

这里  $F(\bar{x})$  为网络的输出向量。网络采用亚当(Adam)优化算法, 输入数据为随机生成服从正态分布 10000 条长度为  $N$  的数据, 标签数据为其傅里叶变换。信号的长度, 分别取 8, 16, 32, 64, 128, 256, 训练 100 个轮次(Epoch), 每批次大小(Batch size)为 100。图 2 展示了不同长度输入信号的损失函数随迭代次数变化示意图。从图中可以看出, 损失函数随着迭代次数的增加而减小, 当输入信号长度越长时下降需要的迭代次数越多。

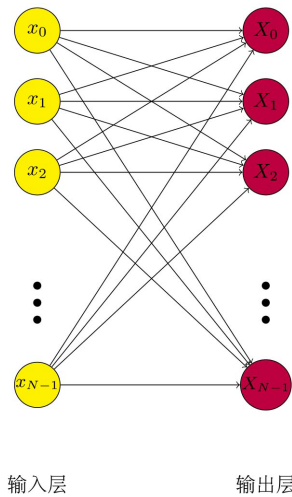


Figure 1. Fourier transform neural network topology diagram  
图 1. 傅里叶变换网络拓扑结构示意图

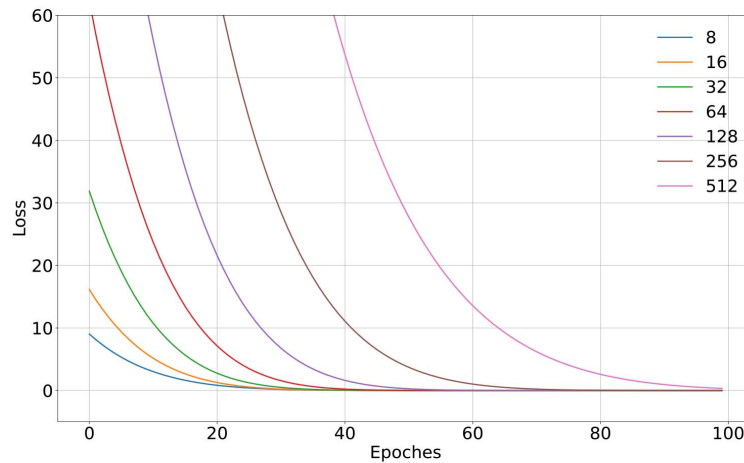


Figure 2. Graph of loss function versus iteration numbers  
图 2. 损失函数随迭代次数变化示意图

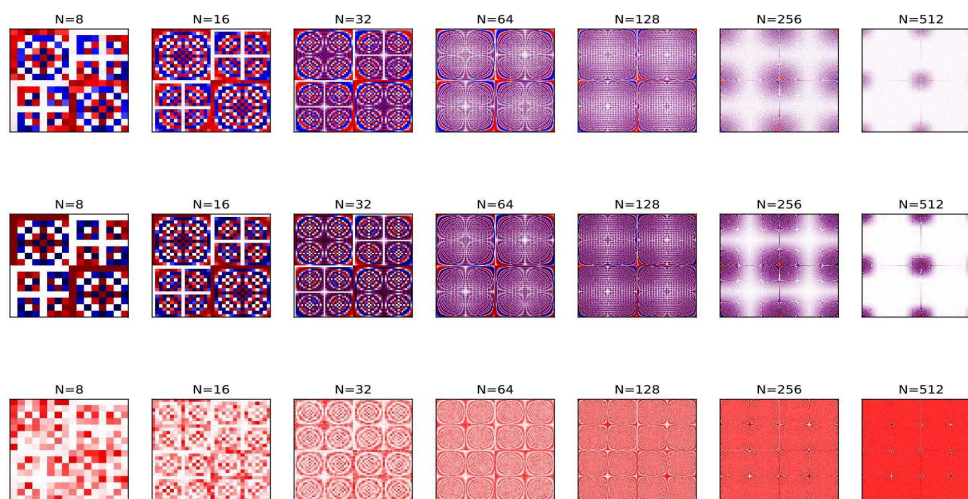
## 4. 数值试验

### 4.1. 核函数分析

图 3 展示了信号长度分别取 8, 16, 32, 64, 128, 256, 512 时, 核函数随着迭代次数的变化。其中第一行为迭代 20 次网络学习到的核函数。第二行为公式(7)对应的核函数(理论核函数)。第三行为理论核函数与学习到的核函数之间的误差。从图中可以看出, 迭代次数较小时, 网络学习到的核函数与真实的

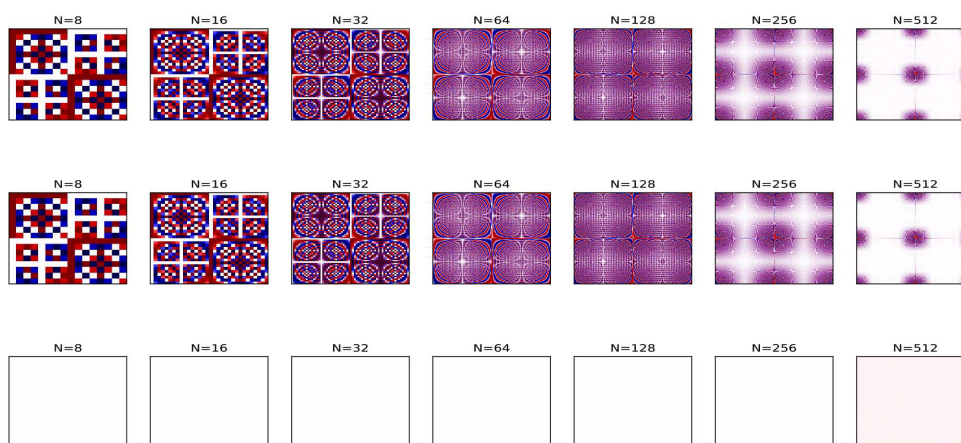
核函数之间有一定的误差。图 4 为迭代得到的核函数、理论核函数及其两者的误差。第一行为迭代 100 次学习到的核函数，第二行为公式(7)对应的核函数，第三行为两者的误差。从图 4 可以看出，当迭代次数为 100 时，两者的误差相对较小。

为更好的分析网络学习的效果，我们对比了信号长度为 128 时，迭代 10 次和 100 次得到的核函数(图 5(a)和图 5(b))。从图中可以看出，迭代次数较小时，学习得到核函数还没有较好的逼近理论核函数。当迭代次数增加到 100 时，学习的得到的核函数较好的逼近了理论核函数。图 6(a)为经过 10 次迭代学习得到的核函数中的前 10 个基函数(蓝色表示核函数的实部，红色表示虚部)。图 6(b)为经过 100 次迭代学习得到的前 10 个基函数。从图中可以看出，随着迭代次数的增加，基函数逼近于傅里叶变换的理论基函数。



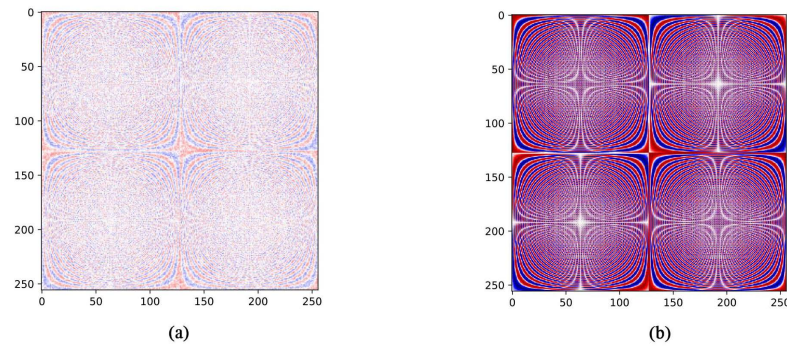
**Figure 3.** Kernel functions corresponding to Fourier transforms with different lengths. The first row shows the kernel functions obtained after 20 iterations of learning, the second row represents the theoretical kernel functions, and the third row illustrates the error between the two

图 3. 不同长度信号傅里叶变换对应的核函数。第一行为迭代 20 次学习到的核函数，第二行为理论核函数，第三行为两者之间的误差

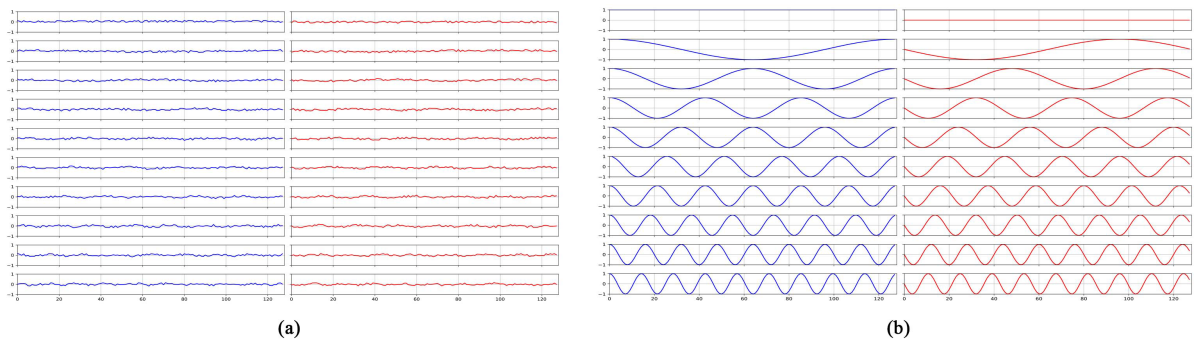


**Figure 4.** Kernel functions corresponding to Fourier transforms with different lengths. The first row shows the kernel functions obtained after 100 iterations of learning, the second row represents the theoretical kernel functions, and the third row illustrates the error between the two

图 4. 不同长度信号傅里叶变换对应的核函数。第一行为迭代 100 次学习到的核函数，第二行为理论核函数，第三行为两者之间的误差



**Figure 5.** The learned kernel function (input signal length is 128). (a) After 10 iterations; (b) After 100 iterations  
**图 5.** 输入信号长度为 128, 学习得到的核函数: (a) 迭代 10 次; (b) 迭代 100 次



**Figure 6.** Learned basis functions (Input signal length is 128), where blue represents the real part, and red represents the imaginary part of the basis function. (a) After 10 iterations; (b) After 100 iterations  
**图 6.** 学习到的基函数(输入信号长度为 128), 蓝色表示基函数的实部, 红色表示虚部。(a) 迭代 10 次; (b) 迭代 100 次

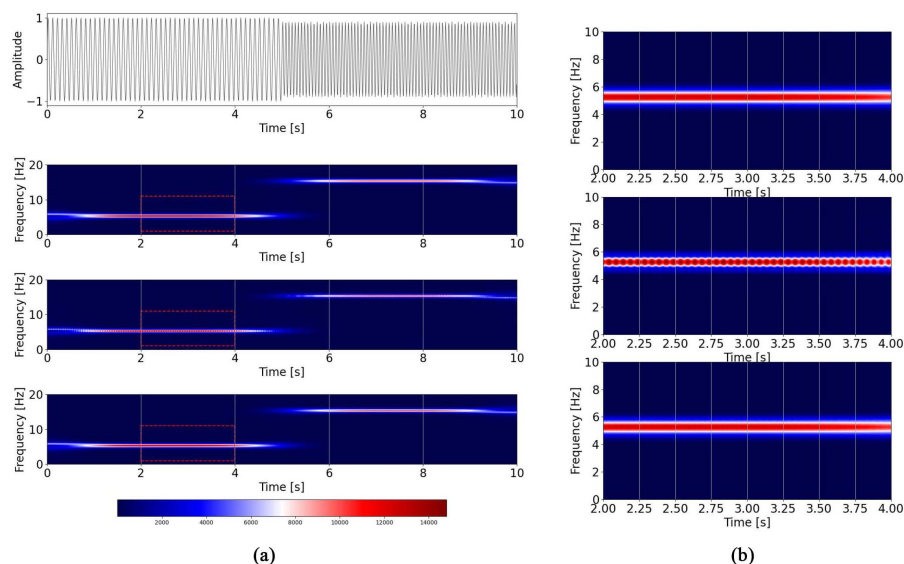
## 4.2. 信号时频分析

图 7(a)为一模拟信号, 该信号中含有 5 赫兹, 15 赫兹两个成份。图 7(a)图从上至下依次为合成信号, 短时傅里叶变换时频图, 迭代 20 次和 100 次得到核函数预测的时频图。图 7(b)图为图 7(a)图中红色虚线框对应的放大图。从图中可以看出, 迭代 20 次预测的时频图位于中间的部分, 连续性较差。这是由于迭代次数较少, 网络还没有充分学习变换的规律。当迭代次数增加到 100 时, 网络学习到核函数较好地逼近理论核函数, 其时频分析预测图与短时傅里叶变换得到的时频图较好吻合。

图 8, 图 9 为两个合成信号。图 8(a), 图 9(a)从上至下依次为采用常规短时傅里叶变换(窗函数长度为 128 的高斯窗), 经过 20 次和 100 次迭代得到核函数的时频预测图。图 8(b), 图 9(b)分别为图 8(a), 图 9(a)红色虚线框的放大图。从图中可以看出, 经过 100 次迭代, 基于神经网络学习得到的核函数的时频分析结果与基于常规短时傅里叶变换的结果基本一致。

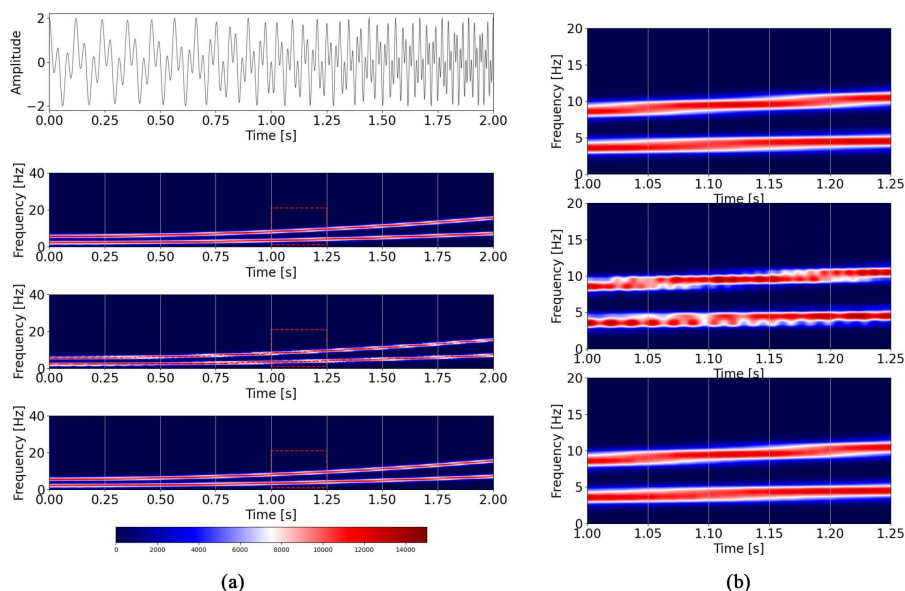
图 10(a)为一道海上地震数据及其时频分析结果图。图 10(a)图从上至下分别为采用常规短时傅里叶变换(窗函数长度为 128 的高斯窗), 经过 20 次和 100 次迭代得到的核函数的时频图。图 10(b)图为图 10(a)图红色虚线框的放大图。从图中可以看出, 当迭代次数为 20 时, 本文提出方法的时频分析结果不如常规傅里叶变换的结果。经过 100 次迭代, 该变换取得的结果与常规短时傅里叶变换的结果几乎相同。

图 11(a)为一个二维地震数据, 该数据有 471 道, 采样时间为 0.002。图 11(b)~(d)分别为常规短时傅里叶变换(长度为 128 的高斯窗)、经过 20 次和 100 次迭代得到的核函数的时频分析结果。三个数据体



**Figure 7.** Comparison of time-frequency analysis for simulated signal. (a) Represents the synthetic signal and its spectrogram. From top to bottom: Synthetic signal, short-time Fourier transform (STFT) spectrogram. Spectrogram using the proposed method (kernel function after 20 iterations). Spectrogram using the proposed method (kernel function after 100 iterations); (b) Is an enlarged view of the red dashed box in (a)

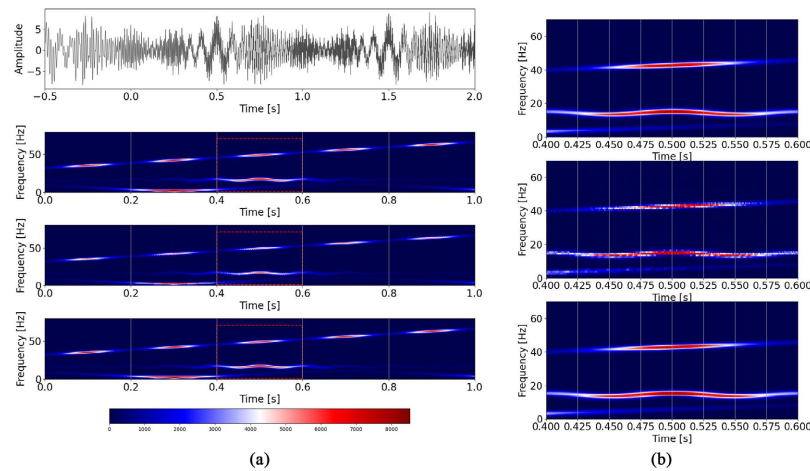
**图 7.** 模拟信号时频分析对比。(a) 表示合成信号及其频谱图。从上至下依次为: 合成信号, 短时傅里叶变换频谱, 本文提出的方法频谱(核函数为迭代 20 次得到), 本文提出的方法频谱(核函数为迭代 100 次得到); (b) 为(a)中红色虚线框的放大图



**Figure 8.** Comparison of time-frequency analysis for simulated signal. (a) Represents the synthetic signal and its spectrogram. From top to bottom: Synthetic signal, short-time Fourier transform (STFT) spectrogram, spectrogram using the proposed method (kernel function after 20 iterations). Spectrogram using the proposed method (kernel function after 100 iterations); (b) Is an enlarged view of the red dashed box in (a)

**图 8.** 模拟信号时频分析对比。(a) 表示合成信号及其频谱图。从上至下依次为: 合成信号, 短时傅里叶变换频谱, 本文提出的方法频谱(核函数为迭代 20 次得到), 本文提出的方法频谱(核函数为迭代 100 次得到); (b) 为(a)中红色虚线框的放大图

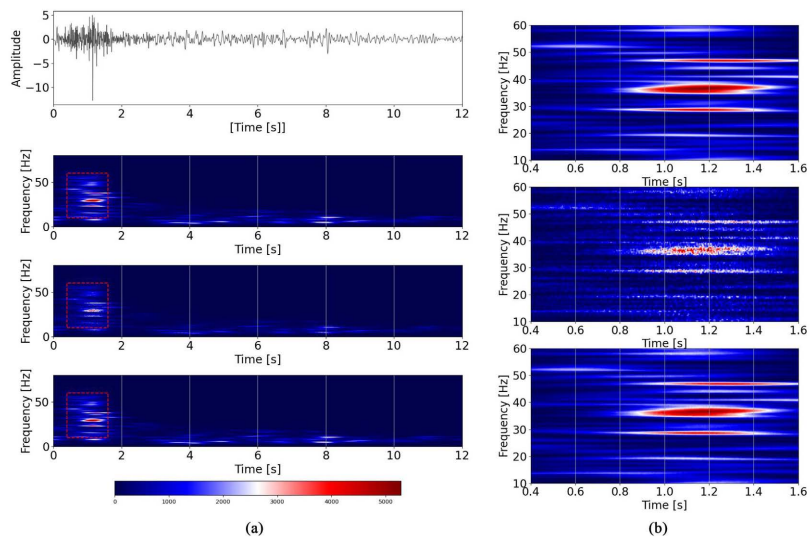




**Figure 9.** Comparison of time-frequency analysis for simulated signal. (a) Represents the synthetic signal and its spectrogram. From top to bottom: Synthetic signal, short-time Fourier transform (STFT) spectrogram, spectrogram using the proposed method (kernel function after 20 iterations). Spectrogram using the proposed method (kernel function after 100 iterations); (b) Is an enlarged view of the red dashed box in (a)

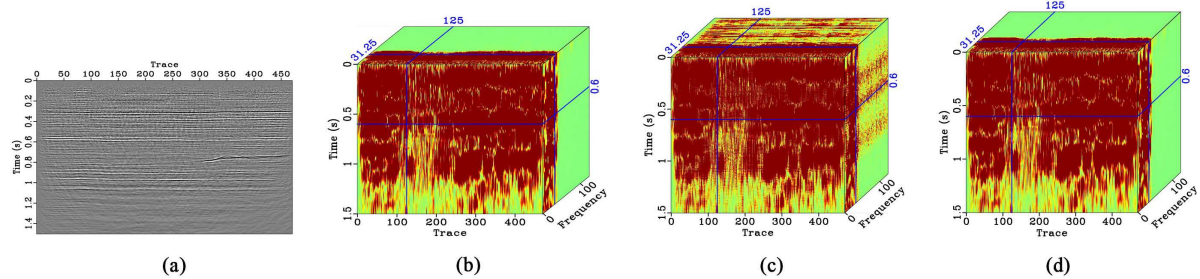
**图 9.** 模拟信号时频分析对比。(a) 表示合成信号及其频谱图。从上至下依次为: 合成信号, 短时傅里叶变换频谱, 本文提出的方法频谱(核函数为迭代 20 次得到), 本文提出的方法频谱(核函数为迭代 100 次得到); (b) 为(a)中红色虚线框的放大图

的正前方展示的是 31.25 Hz 的时频切片, 右侧方展示的是第 125 道的时 - 频谱, 正上方为深度为 0.6 秒的时 - 频谱。从图中可以看出, 迭代次数较小的时频数据体, 能量没有较好的聚焦。当迭代次数增加到 100 时, 学习得到核函数的时频数据体与常规傅里叶变换时频数据体结果基本一致。



**Figure 10.** Comparison of time-frequency analysis for real signal. (a) Represents the real signal and its spectrogram. From top to bottom: signal, short-time Fourier transform (STFT) spectrogram, spectrogram using the proposed method (kernel function after 20 iterations), spectrogram using the proposed method (kernel function after 100 iterations); (b) Is an enlarged view of the red dashed box in (a)

**图 10.** 真实信号时频分析对比。(a) 表示真实信号及其频谱图。从上至下依次为: 信号, 短时傅里叶变换频谱, 本文提出的方法频谱(核函数为迭代 20 次), 本文提出的方法频谱(核函数为迭代 100 次); (b) 为(a)中红色虚线框的放大图



**Figure 11.** Comparison of time-frequency analysis for seismic data. (a) Seismic data; (b) Time-frequency cube using windowed Fourier transform; (c) Time-frequency cube of the kernel function obtained after 20 iterations; (d) Time-frequency cube of the kernel function obtained after 100 iterations

**图 11.** 地震数据时频分析对比。(a) 地震数据; (b) 短时傅里叶变换时频图; (c) 迭代 20 次得到的核函数时频图; (d) 迭代 100 次得到的核函数时频图

## 5. 结论与展望

神经网络通过训练能够获得输入数据到标签间的映射关系。当网络结构足够复杂时, 甚至能够表征无限维空间到无限维空间的映射关系。本文设计了一个简单的双层网络, 通过训练能够渐近逼近数据到其傅里叶变换的线性映射关系。在此基础上, 我们基于学习到的核函数进行了窗口时频分析。验证了, 当迭代次数较大时, 本文提出的方法能够取得和常规短时傅里叶变换相一致的结果。后续研究应关注: 如何设计网络结构, 使其能够快速有效地学习数据间的映射关系。如何从较少的数据中学习数据间的映射关系。

## 致 谢

感谢油气资源与探测国家重点实验室“1158”项目的支持(No.PRP/DX-2305)。

## 参考文献

- [1] 武国宁, 曹思远, 马宁, 刘建军. S 变换的时频分析特性及其改进[J]. 地球物理学进展, 2011, 26(5): 1661-1667.
- [2] 武国宁, 曹思远, 孙娜. 基于复数道地震记录的匹配追踪算法及其在储层预测中的应用[J]. 地球物理学报, 2012, 55(6): 2027-2034.
- [3] 周剑雄, 陈付彬, 石志广, 付强. 补零离散傅里叶变换的插值算法[J]. 信号处理, 2007(5): 690-694.
- [4] 张德干, 郝先臣, 高光来, 赵海. 一种基于快速傅里叶变换的小波变换方法[J]. 东北大学学报, 2000(6): 598-601.
- [5] 孟小红, 郭良辉, 张致付, 李淑玲, 周建军. 基于非均匀快速傅里叶变换的最小二乘反演地震数据重建[J]. 地球物理学报, 2008, 51(1): 235-241.
- [6] Liu, Y. and Fomel, S. (2013) Seismic Data Analysis Using Local Time-Frequency Decomposition. *Geophysical Prospecting*, **61**, 516-525. <https://doi.org/10.1111/j.1365-2478.2012.01062.x>
- [7] Liu, G.C., Fomel, S. and Chen, X.H. (2011) Time-Frequency Analysis of Seismic Data Using Local Attributes. *Geophysics*, **76**, 23-34. <https://doi.org/10.1190/geo2010-0185.1>
- [8] Wu, G.N., Fomel, S. and Chen, Y.K. (2018) Data-Driven Time-Frequency Analysis of Seismic Data Using Non-Stationary Prony Method. *Geophysical Prospecting*, **66**, 85-97. <https://doi.org/10.1111/1365-2478.12530>
- [9] EI-Bakry, H.M. and Zhao, Q.F. (2004) Fast Object/Face Detection Using Neural Networks and Fast Fourier Transform. *International Journal of Signal Processing*, **1**, 182-187.
- [10] Li, Z.Y., Kovachki, N., Azizzadenesheli, K., et al. (2020) Fourier Neural Operator for Parametric Partial Differential Equations. arXiv: 2010.08895.
- [11] 杨庭威, 曹丹平, 杜南樵, 崔荣昂, 南方舟, 徐亚, 梁策. 基于深度学习的接收函数横波速度预测[J]. 地球物理学报, 2022, 65(1): 214-226.
- [12] 张逸伦, 喻志超, 胡天跃, 何川. 基于 U-Net 的井中多道联合微地震震相识别和初至拾取方法[J]. 地球物理学报, 2022, 65(1): 214-226.

- 2021, 64(6): 2073-2085.
- [13] 宋欢, 毛伟建, 唐欢欢. 基于深层神经网络压制多次波[J]. 地球物理学报, 2021, 64(8): 2795-2808.
- [14] 赵明, 陈石, Yuen, D. 基于深度学习卷积神经网络的地震波形自动分类与识别[J]. 地球物理学报, 2019, 62(1): 374-382.
- [15] Mathieu, M., Henaff, M. and LeCun, Y. (2013) Fast Training of Convolutional Networks through FFTs. arXiv: 1312.5851.
- [16] Highlander, T. and Rodriguez, A. (2016) Very Efficient Training of Convolutional Neural Networks Using Fast Fourier Transform and Overlap-and-Add. arXiv: 1601.06815.
- [17] Lin, S., Liu, N., Nazemi, M., Li, H.J., Ding, C.W., Wang, Y.Z. and Pedram, M. (2018) FFT-Based Deep Learning Deployment in Embedded Systems. 2018 *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, 19-23 March 2018, 1045-1050. <https://doi.org/10.23919/DATE.2018.8342166>
- [18] Chitsaz, K., Hajabdollahi, M., Karimi, N., Samavi, S. and Shirani, S. (2020) Acceleration of Convolutional Neural Network Using FFT-Based Split Convolutions. arXiv: 2003.12621.
- [19] Koplun, R. and Sontag, E.D. (1997) Using Fourier-Neural Recurrent Networks to Fit Sequential Input/Output Data. *Neurocomputing*, **15**, 225-248. [https://doi.org/10.1016/S0925-2312\(97\)00008-8](https://doi.org/10.1016/S0925-2312(97)00008-8)
- [20] Zhang, J., Lin, Y.B., Song, Z. and Dhillon, I.S. (2018) Learning Long Term Dependencies via Fourier Recurrent Units. arXiv: 1803.06585.
- [21] Zhang, Y. and Chan, L.W. (2000) ForeNet: Fourier Recurrent Networks for Time Series Prediction. *7th International Conference on Neural Information Processing*, 576-582. <https://api.semanticscholar.org/CorpusID:17884060>
- [22] Cheng, Y., Yu, F.X., Feris, R.S., Kumar, S., Choudhary, A. and Chang, S.F. (2015) An Exploration of Parameter Redundancy in Deep Neural Networks with Circulant Projections. *IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 2857-2865. <https://doi.org/10.1109/ICCV.2015.327>
- [23] Choromanski, K., Likhoshesterov, V., Dohan, D., Song, X.Y., Davis, J., Sarlos, T., Belanger, D., Colwell, L. and Weller, A. (2020) Masked Language Modeling for Proteins via Linearly Scalable Long-Context Transforms. arXiv: 2006.03555.
- [24] Lee-Thorp, J., Ainslie, J., Eckstein, I. and Ontanon, S. (2021) FNet: Mixing Tokens with Fourier Transforms. arXiv: 2015.03824.