课程大作业报告

1. 小组基础信息

小组队名	无言以队
小组成员	**谢浩志+ZY2406222** 黄星阳+ZY2406437
科学任务	FEDformer 求解时间序列预测问题
论文标题	FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting
科学领域	时序

2. 背景介绍

(1) 科学任务背景

长时间序列预测是许多实际应用中的关键任务，被广泛应用于能源、气候、交通、经济等领域。准确的长时间序列预测对于资源分配、决策指定和风险管理等方面具有重要意义。然而，由于长时间序列的数据存在时间跨度长、特征复杂多样、数据维度高等特点，使得预测任务极具挑战性。

近年来，深度学习方法，尤其是循环神经网络(RNN)等方法在时间序列预测中取得了显著进展，但在处理长时间序列时常常受到梯度爆炸或梯度消失问题的限制。随着相关论文的发表，Transformer 架构也被引入到时间序列预测中，并取得了较好的成果。但是较高的计算复杂度和内存需求使得 Transformer 难以应用于长时间序列预测。虽然目前已经有大量研究致力于降低其的计算成本的工作，但基于 Transformer 的时间序列预测方法往往无法捕捉到时间序列的整体特征，导致预测结果与真实值仍然存在较大的差异。

(2) 主要挑战

- 计算复杂度高

标准 Transformer 架构的自注意力机制具有二次的时间复杂度 $O(N^2)$ 。例如对于长度为 L 的序列，计算复杂度就达到了 $O(L^2)$，这意味着当序列变长时，计算开销会急剧增加，严重限制了 Transformer 架构在长时间序列上的使用。

- 内存开销大

由于标准 Transformer 架构的自注意力机制需要维护整个序列的注意力矩阵，

空间复杂度达到了 $O(N^2)$，当序列较长时，所需的内存开销也会急剧增加，难以在有限的硬件资源上处理较长的序列，也限制了 Transformer 架构在长时间序列上的使用。

- 整体特征捕捉困难

受限于 Transformer 架构的逐点注意力机制，随着序列长度的增加，基于 Transformer 架构的模型捕捉到的远距离依赖信息可能变得稀疏且不准确。在实际处理长时间序列这类特别长的序列时，模型在捕捉局部特征时往往具有较好的表现，但忽略了全局的上下文信息，导致模型无法保持时间序列的整体分布特性，进而严重影响了模型的预测结果。

3. 问题定义

长时序预测问题旨在基于已有的历史观测数据预测未来一段时间内的时序演化。该任务可形式化为一个序列到序列的映射过程。设输入的时间序列为 $X = x_1, x_2, ..., x_I \in \mathbb{R}^{I \times D}$，其中 I 表示输入序列的时间步长，$D$ 表示每个时间步的特征维度。模型的目标是预测未来 O 个时间步的序列，记为 $Y = \{y_{I+1}, y_{I+2}, ..., y_{I+O}\} \in \mathbb{R}^{O \times D}$ 其中 O 为预测长度，对应的预测结果为 $\hat{Y} = \{\hat{y}_{I+1}, \hat{y}_{I+2}, ..., \hat{y}_{I+O}\} \in \mathbb{R}^{O \times D}$，为模型生成的输出序列。

具体而言，在给定输入序列 X 的条件下，学习一组参数化的模型 $Enc(\cdot)$ 和$Dec(\cdot)$，由编码器 $Enc(\cdot)$ 将输入序列 X 映射为一个潜在的隐藏状态序列 $H = \{h_1, h_2, ..., h_I\} \in \mathbb{R}^{I \times D'}$，其中 D' 为隐藏状态的表示维度，编码过程可表示为 $H = Enc(X)$，随后，解码器$Dec(\cdot)$以隐藏状态序列 H 为输入，生成预测序列 $\hat{Y} = Dec(H)$，其最终学习目标是使预测序列 $\hat{Y}$ 尽可能接近真实序列 Y，即使得预测值与真实值间的均方误差

整个模型由编码器与解码器联合建模，其最终学习目标是使预测序列 $\hat{Y}$ 尽可能接近真实序列 Y。为此，定义损失函数为预测值与真实值之间的均方误差（Mean Squared Error, MSE）为 $\mathcal{L}(\theta) = \frac{1}{O}\sum_{t=1}^{O} ||\hat{y}_{I+t} - y_{I+t}||_2^2$，其中 θ 表示模型的参数集合，包含编码器和解码器部分的参数，即 $\theta = \{\theta_{Enc(\cdot)}, \theta_{Dec(\cdot)}\}$，同时预

测输出满足 $\widehat{Y} = Dec(Enc(X;\theta_{Enc});\theta_{Dec})$。

因此，长时序预测问题可归结为以下最优化问题的求解：$\min_{\theta} \mathcal{L}(\theta)$，该优化目标旨在最小化模型预测输出与真实未来序列之间的距离，以提升模型对未来时序演化趋势的拟合与泛化能力。

4. 研究现状

长时间序列预测作为一项长期受到关注的研究课题，预测模型主要经历了从传统统计方法、循环神经网络（RNN）系列模型，到近年来 Transformer 架构的广泛应用与创新。

(1) 传统时间序列预测模型

早期的时间序 列预测多采用统计建模方法，ARIMA[1][2]是最早的统计建模方法之一，适用于马尔科夫过程，但难以处理非线性和非平稳序列。

伴随深度学习的发展，RNN 被广泛应用于序列数据的处理，例如 LSTM[3]和 GRU[4]通过引入门控机制解决了梯度消失与爆炸的问题。DeepAR[5]在序列预测中引入了概率建模，Attention based RNN[6]使用临时注意力机制增强乐长依赖建模能力。但这些模型本身难以并行化，同时又难以处理长依赖关系。TCN[7]虽然利用卷积结构实现了高效建模，但受限于卷积核的感受野，仍难以捕捉长时间序列的全局特征。

(2) 基于 Transformer 的时间序列预测

由于 Transfomer[8]的强大性能，原本应用于自然语言处理（NLP）和计算机视觉（CV）任务中 Transfomer 架构的模型也被逐步广泛应用于时间序列预测。但该架构也带来了较高的时间复杂度和空间复杂度，因此研究者们也采取了各种方法尝试解决这一问题。

其中一种方式是通过预先指定稀疏连接策略降低复杂度，例如 Longformer[9]采用滑动窗口稀疏注意力，LogTrans[10]提出了 log-sparse attention，将复杂度降为$O(N\,log^2\,N)$，H-Transformer[11],采用层次化注意力结构，进一步实现了$O(N)$ 复杂度。

Informer[12]和 Autofomer[13]则采用了其他方式。Informer 通过 KL 散度选择注意力矩阵中的 top-k，降低复杂度至 $O(N\,log\,N)$。Autofomer 替换标准注意力为自相关模块，捕捉子序列之间的重复模式，结合快速傅里叶变换（FFT）实现高效的子序列匹配，也实现了 $O(N\,log\,N)$ 的复杂度。

此外，通过数学方法采用注意力矩阵的低秩近似也是一种新兴的方式。Linformer[14]将注意力矩阵通过线性变换降维，实现了$O(N)$ 复杂度。Luna[15]进一步提出嵌套线性结构。Nyströformer[16]借助 Nyström 方法对注意力矩阵近似。Performer[17]则引入乐随机特征方法，模拟核注意力机制。

(3) 傅里叶变换在深度学习中的应用

傅里叶变换将将时间域信号映射到频率域，有助于揭示信号中的周期成分，同时，在频域中，卷积操作也可以简化为乘法从而大幅提高效率。由于快速傅里叶变换的提出大幅降低了傅里叶变换的时间复杂度，将傅里叶变换引入深度学习以提高效率中也因此成为了可能。

Mathieu[18]等人将 FFT 应用于卷积神经网络的加速，RobustPeriod[19]使用 FFT 计算自相关函数，提升效率。 Autoformer[13]将 FFT 引入 Auto-Correlation 模块，用于序列模式匹配。

Li[20]和 Gupta[21] 等人最早将傅里叶变换引入求解偏微分方程（PDE）的问题中。Rahimi[22]]等人将输入映射到随机傅里叶特征空间，进而加速了核方法。RF-Softmax[23]利用随机傅里叶实现了近似 softmax 的高效计算。

FNet[24]将 Transformer 中的自注意力层用标准傅里叶变换替代，以几乎为零的性能损失实现了更快的训练速度。

5. 论文方法

FEDformer 作为一种面向长时间序列预测任务的 Transformer 变体，其整体结构如下图所示：

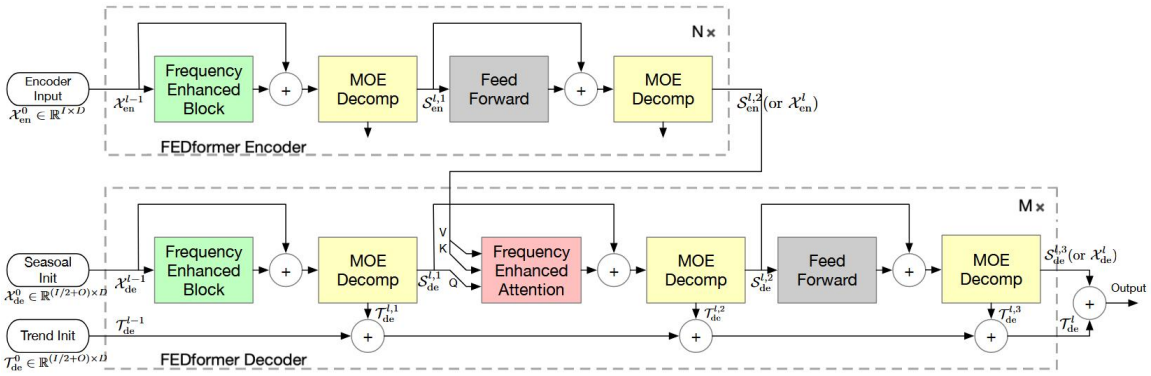

图 1 FEDformer 整体架构图

模型整体采用编码器-解码器框架，并在此基础上引入了序列分解机制与频域建模模块，以增强模型对长周期、多尺度动态特征的建模能力。输入序列首先经过嵌入层处理，结合时间戳信息转换为高维特征表示，随后通过序列分解模块将原始序列拆解为趋势项与季节项两个分量。趋势项反映长期缓变趋势，季节项则保留周期性波动成分，分别交由不同子模块处理。编码器部分主要负责对季节项进行表征建模，采用由多个频域注意力层构成的堆叠结构以提取周期信号中的时序关联信息。解码器则在接收编码器输出的同时，引入时间步补齐与交叉注意力机制，生成未来时间步的季节项预测结果，并与趋势项拼接得到最终输出序列。该结构以“分解-频域建模-重组”的方式，有效缓解了序列长度增长带来的建模难度，并提升了对整体时序动态的拟合能力。

为应对标准 Transformer 在长序列预测中面临的高计算复杂度问题，FEDformer 引入了基于频域的注意力机制，替代了原始 Transformer 中的全局点对点自注意力计算。具体而言，模型采用傅里叶变换（Fourier Transform）或多尺度小波变换（Multi-Wavelet Transform）将序列映射至频率域，并在主频模式选择策略下，仅保留固定数量的频域分量用于注意力建模，从而显著降低了计算复杂度。相比传统自注意力需要在长度为 L 的序列上计算 $O(L^2)$ 的注意力矩阵，FEDformer 仅对选定的频域模式执行线性投影与交互操作，整体复杂度降低至$O(L)$，可扩展至更长时间跨度的数据建模任务。此外，频域处理还具备天然的周期性建模优势，使模型更易捕捉到隐藏在时序数据中的规律性信号。

面对 Transformer 难以捕捉时间序列全局趋势信息的挑战，FEDformer 采用“趋势-季节”双路径解耦机制作为结构基础，通过滑动平均实现输入序列的趋势项提取，并在解码阶段将历史趋势的均值部分直接外推作为未来时间段的趋势

估计值。此举不仅避免了全局趋势建模中的学习不稳定性，也减少了解码器对冗余信息的关注负担。同时，在编码与解码阶段对季节项采用频域处理，使其专注于建模残差中的高频波动模式，进而提升对局部周期性成分的表达能力。这种结构化的表征方式确保了模型同时具备全局趋势拟合与局部细节捕捉的能力，实现了更具鲁棒性的长时间序列预测性能。

6. 复现结果

本次实验基于 MindSpore 昇思深度学习框架复现 FEDformer 模型，目标是在保障模型结构与原论文保持一致的前提下完成实验验证。具体而言，我们在不改动网络主体结构的基础上，将模型中的 FFT 模块、序列分解模块、频域注意力模块等关键组件逐一迁移至 MindSpore，并实现了包括 FourierBlock、FourierCrossAttention 在内的核心模块复现。在训练与推理过程中，充分利用 MindSpore 支持的动态图机制进行调试验证，同时结合原项目配置重构了数据加载与训练流程，实现在多数据集上训练与评估。

在复现过程中，出现三个主要问题。

（1）由于昇腾 AI 处理器当前尚不支持 ops.FFTWithSize 等复数傅里叶变换相关算子，因此我们无法直接在昇腾平台完成全流程训练，最终将实验平台迁移至具备 NVIDIA Tesla P100 GPU 的环境进行运行。

（2）MindSpore 在自动微分过程中不支持复数类型数据的梯度反传运算，导致涉及频域卷积或复数乘法的模块无法直接进行训练。为此我们将相关复杂计算模块以实部和虚部分离的方式进行改写。具体来说，我们将模型结构中频域卷积的权重分为了实部和虚部分别存储如下：

```
# 只存储实数权重
self.weights1_real = ms.Parameter(self.scale * ops.randn((8, in_channels//8, out_channels//8, len(self.index))))
self.weights1_imag = ms.Parameter(self.scale * ops.randn((8, in_channels//8, out_channels//8, len(self.index))))

self.rfft = ops.FFTWithSize(signal_ndim=1, inverse=False, real=True)
self.irfft = ops.FFTWithSize(signal_ndim=1, inverse=True, real=True, signal_sizes=(seq_len,))
```

之后在继续卷积计算时将实部和虚部分开，实现代码如下：

```python
def complex_mul(a_real, a_imag, b_real, b_imag, mode):
    """使用实数表示执行复数乘法"""
    einsum = ops.Einsum(mode)
    real = einsum((a_real, b_real)) - einsum((a_imag, b_imag))
    imag = einsum((a_real, b_imag)) + einsum((a_imag, b_real))
    return real, imag  # 返回实部和虚部的实数张量
```

（3）在图模式（GRAPH MODE）下，MindSpore 对于部分高阶函数与复合控制流的支持仍不完善，导致训练中出现算子不支持或类型推导失败的情况。为此最终实验在 PYNATIVE（动态图）模式下进行，硬件设置部分的代码如下：

```python
class Exp_Basic:
    def __init__(self, args):
        self.args = args
        self._set_device()
        self.model = self._build_model()

    def _set_device(self):
        if self.args.use_gpu:
            context.set_context(mode=context.PYNATIVE_MODE, device_target="GPU", device_id=self.args.gpu)
            # context.set_context(mode=context.GRAPH_MODE, device_target="GPU", device_id=self.args.gpu)
            # context.set_context(mode=context.PYNATIVE_MODE, device_target="Ascend", device_id=self.args.gpu)
            print(f"Using GPU: {self.args.gpu}")
        else:
            context.set_context(mode=context.PYNATIVE_MODE, device_target="CPU")
            print("Using CPU")
```

本次实验选择了 FEDformer、Autoformer、Informer 与 Transformer 四种模型结构作为对比对象，在公开的 Electricity、Exchange、Traffic 与 Weather 四个数据集上进行了复现实验。考虑到设备资源限制及整体实验开销，实验仅在预测步长为 96 的场景下进行了 Multivariate 设定（输入为多个变量组成的多维时间序列）下的测试，最终实验结果如下表所示：

表 1 FEDformer 实验结果表

方法	指标\数据集	Electricity	Exchange	Traffic	Weather
Transformer	MAE	**0.830**	1.418	**0.911**	1.580
	RMSE	**1.014**	1.784	**1.315**	2.066
	MAPE	**3.238**	12.247	**5.847**	25.460
Informer	MAE	0.938	1.589	0.963	1.743
	RMSE	1.133	1.954	1.353	2.233
	MAPE	5.320	9.693	7.462	31.488
Autoformer	MAE	1.698	**0.278**	1.941	2.379
	RMSE	2.016	**0.385**	2.265	3.797
	MAPE	13.925	**1.691**	25.638	87.032
FEDformer	MAE	1.326	0.278	1.795	**0.319**
	RMSE	1.562	0.386	2.101	**0.501**
	MAPE	10.921	1.699	24.045	**18.864**

从结果来看，FEDformer 在部分指标上仍表现出领先优势，尤其在

Exchange 与 Weather 两个数据集上取得了较低的 MAE 和 RMSE 值，验证了其频域建模和季节性模式的捕捉，在季节趋势强的数据集上产生一定优势。但总体而言，复现性能与原论文报告存在一定差异，表现为各模型在不同数据集上的数值波动较大，尤其在 Autoformer 和 Informer 两个模型上，误差值相比原文显著升高。造成这一现象的可能原因包括硬件平台不同导致的浮点计算偏差、MindSpore 框架下实现与 PyTorch 存在的细节差异、图模式向动态模式切换所引入的部分不可训练模块、训练轮数与初始化权重未能完全复刻等。此外，复数处理方式的近似化也可能弱化了频域建模模块的表达能力。未来若可基于 MindSpore 原生支持的复数与 FFT 算子进一步优化实现，有望获得与原文更一致甚至更优的实验性能。

参考文献

[1] Box G E P, Jenkins G M. Some recent advances in forecasting and control[J]. Journal of the Royal Statistical Society. Series C (Applied Statistics), 1968, 17(2): 91-109.

[2] Box G E P, Pierce D A. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models[J]. Journal of the American statistical Association, 1970, 65(332): 1509-1526.

[3] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[4] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[5] Salinas D, Flunkert V, Gasthaus J, et al. DeepAR: Probabilistic forecasting with autoregressive recurrent networks[J]. International journal of forecasting, 2020, 36(3): 1181-1191.

[6] Qin Y, Song D, Chen H, et al. A dual-stage attention-based recurrent neural network for time series prediction[J]. arXiv preprint arXiv:1704.02971, 2017.

[7] Sen R, Yu H F, Dhillon I S. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting[J]. Advances in neural information processing systems, 2019, 32.

[8] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[9] Beltagy I, Peters M E, Cohan A. Longformer: The long-document transformer[J]. arXiv preprint arXiv:2004.05150, 2020.

[10] Li S, Jin X, Xuan Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting[J]. Advances in neural information processing systems, 2019, 32.

[11] Zhu Z, Soricut R. H-transformer-1d: Fast one-dimensional hierarchical attention for sequences[J]. arXiv preprint arXiv:2107.11906, 2021.

[12] Zhou H, Zhang S, Peng J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(12): 11106-11115.

[13] Wu H, Xu J, Wang J, et al. Autoformer: Decomposition transformers with

auto-correlation for long-term series forecasting[J]. Advances in neural information processing systems, 2021, 34: 22419-22430.

[14] Wang S, Li B Z, Khabsa M, et al. Linformer: Self-attention with linear complexity[J]. arXiv preprint arXiv:2006.04768, 2020.

[15] Ma X, Kong X, Wang S, et al. Luna: Linear unified nested attention[J]. Advances in Neural Information Processing Systems, 2021, 34: 2441-2453.

[16] Xiong Y, Zeng Z, Chakraborty R, et al. Nyströmformer: A nyström-based algorithm for approximating self-attention[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(16): 14138-14148.

[17] Choromanski K, Likhosherstov V, Dohan D, et al. Rethinking attention with performers[J]. arXiv preprint arXiv:2009.14794, 2020.

[18] Mathieu M, Henaff M, LeCun Y. Fast training of convolutional networks through ffts[J]. arXiv preprint arXiv:1312.5851, 2013.

[19] Wen Q, He K, Sun L, et al. RobustPeriod: Robust time-frequency mining for multiple periodicity detection[C]//Proceedings of the 2021 international conference on management of data. 2021: 2328-2337.

[20] Li Z, Kovachki N, Azizzadenesheli K, et al. Fourier neural operator for parametric partial differential equations[J]. arXiv preprint arXiv:2010.08895, 2020.

[21] Gupta G, Xiao X, Bogdan P. Multiwavelet-based operator learning for differential equations[J]. Advances in neural information processing systems, 2021, 34: 24048-24062.

[22] Rahimi A, Recht B. Random features for large-scale kernel machines[J]. Advances in neural information processing systems, 2007, 20.

[23] Rawat A S, Chen J, Yu F X X, et al. Sampled softmax with random fourier features[J]. Advances in Neural Information Processing Systems, 2019, 32.

[24] Lee-Thorp J, Ainslie J, Eckstein I, et al. Fnet: Mixing tokens with fourier transforms[J]. arXiv preprint arXiv:2105.03824, 2021.