时序预测模型 Fedformer 研究报告

丘云芸

摘要

摘要：本文通过对 Alibaba Cluster Trace 2018 数据集进行预处理，采用深度学习的技术，基于 FEDformer 来实现云资源指标的预测，提高对云资源的调度能力。本文主义通过频率低秩近似和专家混合分解的注意机制实现控制分布转移，通过频率增强结构解耦输入序列长度和注意矩阵维数，实现预测模型的线性复杂度，在 Alibaba Cluster Trace 2018 数据集基础上，完成对云资源指标的预测。

关键词：时间序列预测；注意力机制；云资源预测

1 引言

时间序列预测在众多领域中（例如电力、能源、天气、交通等）都有广泛的应用。时间序列预测问题极具挑战性，尤其是长程时间序列预测（long-term series forecasting）。在长程时间序列预测中，需要根据现有的数据对未来做出较长时段的预测。在部分场景中，模型输出的长度可以达到 1000 以上，覆盖若干周期。该问题对预测模型的精度和计算效率均有较高的要求。且时间序列往往会受到分布偏移和噪音的影响，使得预测难度大大增加。

针对时间序列问题，传统的 RNN、LSTM 等 Recurrent 模型，在训练时容易受到梯度消失和爆炸的影响，尤其是面对更加长程的序列时。且这类 Recurrent 的模型无法并行计算，限制了其在大规模问题上的应用。

2 相关工作

基于 Transformer 的时间序列预测，通过 Attention 机制捕捉 point-wise 的关系，能够在时序预测中取得较好效果，但仍存在较大不足。Informer、Autoformer 等文章对传统 Attention 机制进行了改进，在提高计算效率的同时能够取得较好的效果。传统 Transformer 为平方复杂度，Autoformer (NeurIPS’21)、Informer (AAAI’21 Best paper)、Reformer (ICLR’2020) 等模型能够达到 log-线性复杂度，而 FEDformer 因使用了 low-rank approximation 而可以达到线性复杂度，并在精度上大幅超越 SOTA（state-of-the-art）结果。

3 本文方法

3.1 本文方法概述

Transformer 在 CV、NLP 等领域取得了很好的效果，但在时间序列预测问题上，情况会更复杂。例如在图片分类问题中，训练集和测试集的图片基本采样自相同的分布。然而在时间序列预测问题中，序列的分布可能随时间轴的推进不断变化，这就需要模型具备更强的外推能力。如下图所示，因为模型输入和真实值的分布差异较大，导致模型的预测值不准确。

为了解决这个问题，FEDformer 提供了两种思路：1. 通过周期趋势项分解（seasonal-trend decomposition）降低输入输出的分布差异；2. 提出了一种在频域应用注意力机制的模型结构，以增加对噪声

的鲁棒性。FEDformer 的主体结构采用编码-解码器结构，内部包括四种子模块：频域学习模块（Frequency Enhanced Block）、频域注意力模块（Frequency Enhanced Attention）、周期-趋势分解模块（MOE Decomp）、前向传播模块（Feed Forward）。

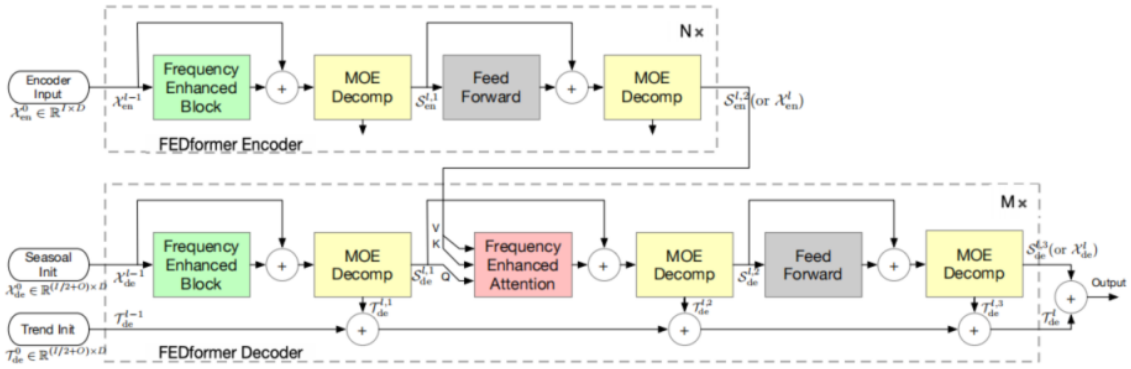

图 1: FEDformer 架构图

3.2 Series Decomposition Block

这个模块主要目的是将时间序列分解成趋势项和季节项。在最基础的时间序列分析领域，一个时间序列可以被视为趋势项、季节项、周期项和噪声。对于这 4 个因素的拆解，有加法模型、乘法模型等，其中加法模型认为这 4 个因素相加构成了当前时间序列。本文采用了加法模型，认为时间序列由趋势项 + 季节项构成。为了提取出季节项，本文采用了滑动平均法，通过在原始输入时间序列上每个窗口计算平均值，得到每个窗口的趋势项，进而得到整个序列的趋势项。同时，根据加法模型，将原始输入序列减去趋势项，即可得到季节项。模型的输入结合 Series Decomposition Block 模块。Encoder 部分输入历史时间序列，Decoder 部分的输入包括趋势项和季节项两个部分。趋势项由两部分组成，一部分是历史序列经过 Series Decomposition Block 分解出的趋势项的后半部分，相当于用历史序列近期的趋势项作为 Decoder 的初始化；趋势项的另一部分是 0 填充的，即目前尚不知道的未来序列的趋势项，用 0 进行填充。季节项和趋势项类似，也是由两部分组成，第一部分为 Encoder 分解出的近期季节项，用于初始化；第二部分为 Encoder 序列均值作为填充。

Encoder 部分的主要目的是对复杂的季节项进行建模。通过多层的 Series Decomposition Block，不断从原始序列中提取季节项。这个季节项会作为指导 Decoder 在预测未来时季节项的信息。Decoder 部分也是类似的结构，利用 Encoder 信息和 Decoder 输入进行预测。

3.3 Auto-Correlation Mechanism

Auto-Correlation Mechanism 的核心思路是利用时间序列的自相关系数，寻找时间序列最相关的片段。时间序列的自相关系数计算时间序列和其滑动一个步长后的时间序列的相关系数。举例来说，如果一个时间序列是以年为周期，那么序列平移 365 天后，原序列和平移后的序列相关系数是很高的。AutoFormer 利用了这个性质，计算各个滑动步长的自相关系数，并选择相关系数 top k 的滑动步长。

在具体实现上，Auto-Correlation Mechanism 替代了 Transformer 中的 self-attention 结构，其公式如下。首先，将输入的时间序列通过全连接映射成 Q、K、V，这和 multi-head attention 相同。接下来，计算 Q 和 K 之间各个周期的相关系数，选择相关系数最高的 top k，这 k 个周期代表着 Q 和 K 的高相关性周期。这个过程可以理解为计算 attention，与以往 attention 不同的是，这里计算的是片段的相似关系而非点的相似关系。最后，利用 softmax 得到每个周期归一化的权重，对 V 的对应周期平移结果进

行加权求和。

4 复现细节

4.1 与已有开源代码对比

本工作引用了 Fedformer[1]提供的代码，实现了 FEDformer 的整体框架，对 Alibaba Cluster Trace 2018 数据集进行预处理，基于 FEDformer 来实现云资源指标的预测。

4.2 实验环境搭建

实验环境：Python 3.6、PyTorch 1.9.0

准备工作长时间序列预测是一个序列到序列的问题，我们用 I 表示输入长度，O 表示输出长度，D 表示序列的隐状态。编码器的输入为 $I \times D$，而解码器的输入为 $(I/2+O) \times D$ 。

```
enc_modes = int(min(configs.modes, configs.seq_len//2))
dec_modes = int(min(configs.modes, (configs.seq_len//2+configs.pred_len)//2))
print('enc_modes: {}, dec_modes: {}'.format(enc_modes, dec_modes))
```

图 2: 输入处理

在编码器中，输入经过两个 MOE Decomp 层，每层会将信号分解为 seasonal 和 trend 两个分量。其中，trend 分量被舍弃，seasonal 分量交给接下来的层进行学习，并最终传给解码器。

```
self.encoder = Encoder(
    [
        EncoderLayer(
            AutoCorrelationLayer(
                encoder_self_att,
                configs.d_model, configs.n_heads),

            configs.d_model,
            configs.d_ff,
            moving_avg=configs.moving_avg,
            dropout=configs.dropout,
            activation=configs.activation
        ) for l in range(configs.e_layers)
    ],
    norm_layer=my_Layernorm(configs.d_model)
)
```

图 3: 编码器

在解码器中，编码器的输入同样经过三个 MOE Decomp 层并分解为 seasonal 和 trend 分量。其中，seasonal 分量传递给接下来的层进行学习，其中通过频域 Attention（Frequency Enhanced Attention）层对编码器和解码器的 seasonal 项进行频域关联性学习，trend 分量则进行累加最终加回给 seasonal 项以还原原始序列。

```
self.decoder = Decoder(
    [
        DecoderLayer(
            AutoCorrelationLayer(
                decoder_self_att,
                configs.d_model, configs.n_heads),
            AutoCorrelationLayer(
                decoder_cross_att,
                configs.d_model, configs.n_heads),
            configs.d_model,
            configs.c_out,
            configs.d_ff,
            moving_avg=configs.moving_avg,
            dropout=configs.dropout,
            activation=configs.activation,
        )
        for l in range(configs.d_layers)
    ],
    norm_layer=my_Layernorm(configs.d_model),
    projection=nn.Linear(configs.d_model, configs.c_out, bias=True)
)
```

图 4: 解码器

4.3 界面分析与使用说明

在 run.py 文件中给出了各类参数的设置说明，可以对模型、数据集、输入维数等进行简便的设置。

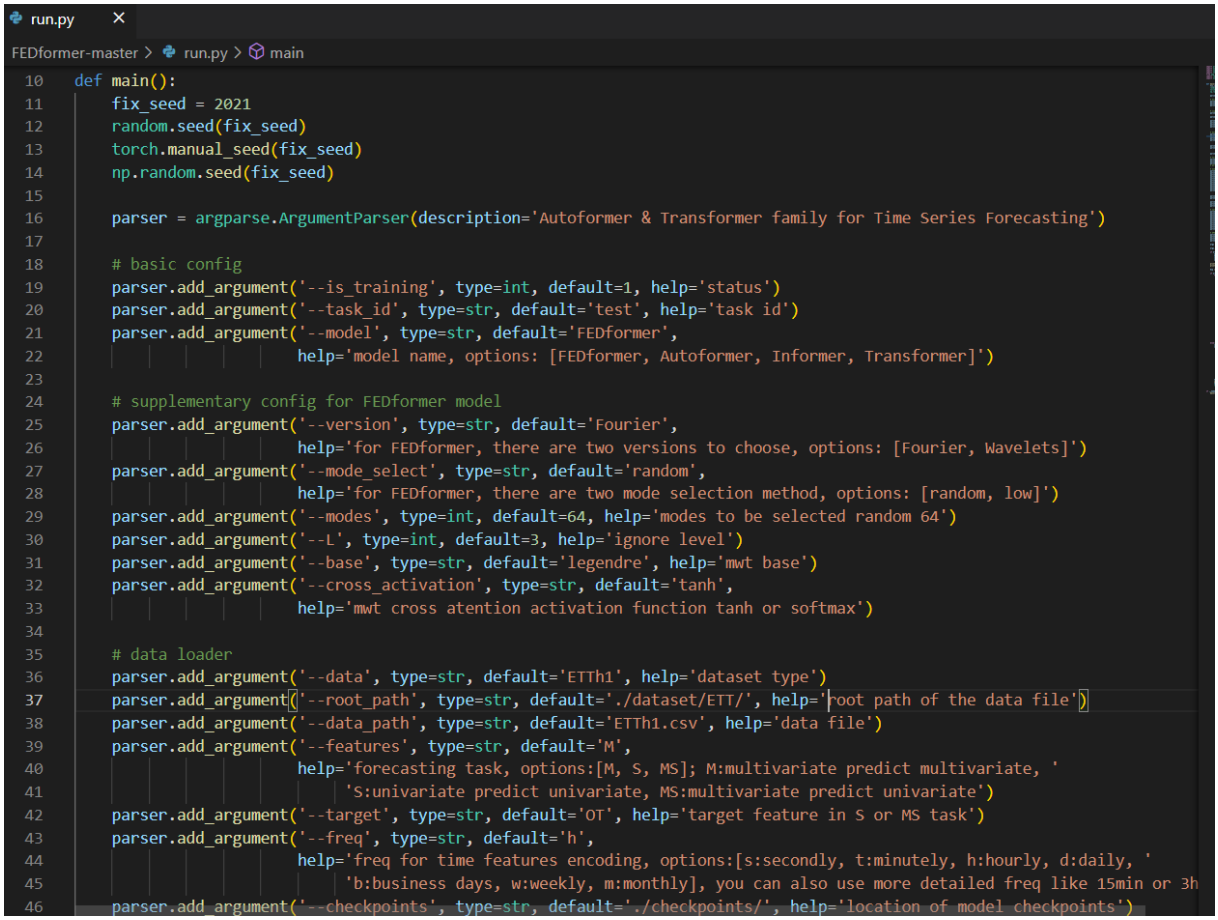

图 5: 操作界面示意

5 实验结果分析

本部分对实验所得结果进行分析，详细对实验内容进行说明，实验结果进行描述并分析。

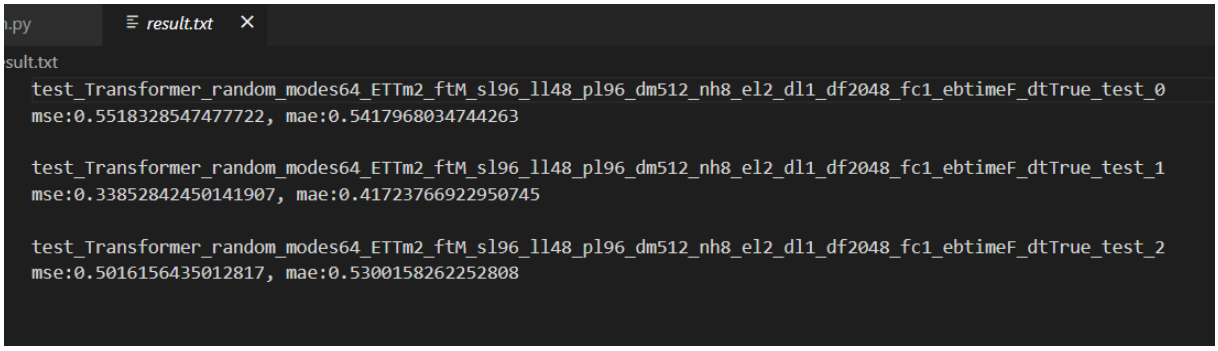

图 6: 复现实验结果示意

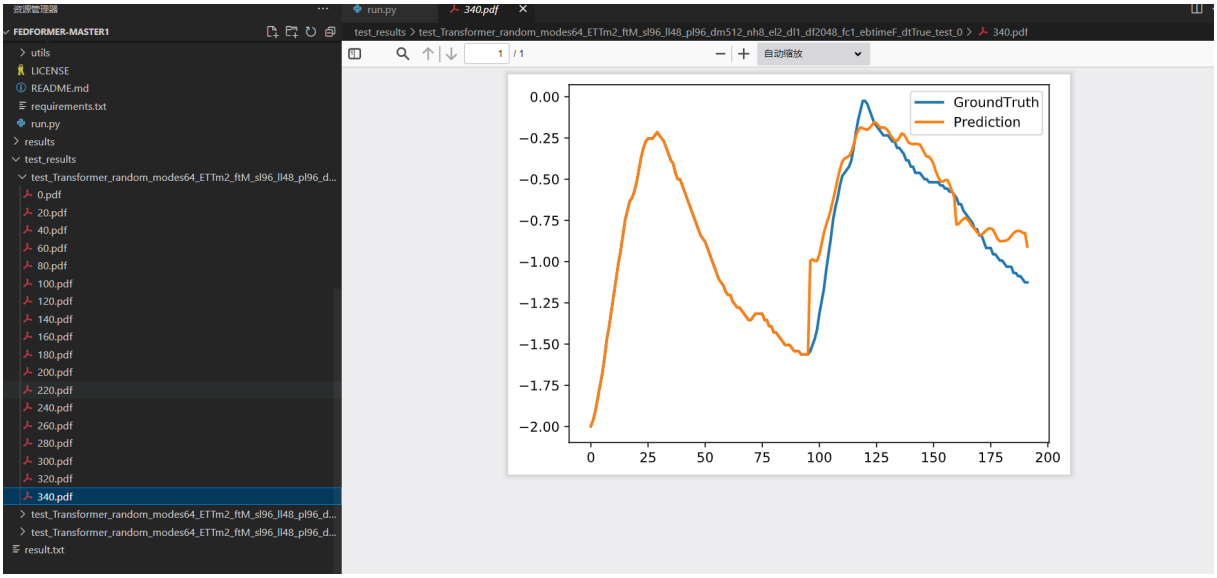

图 7: 复现实验结果示意

我们可以看到在论文原有数据集上预测效果是比较好的，但在云资源数据集上，由于云资源时间序列具有高度非平稳性、高度非线性、变化极快的特点，预测结果不太理想。

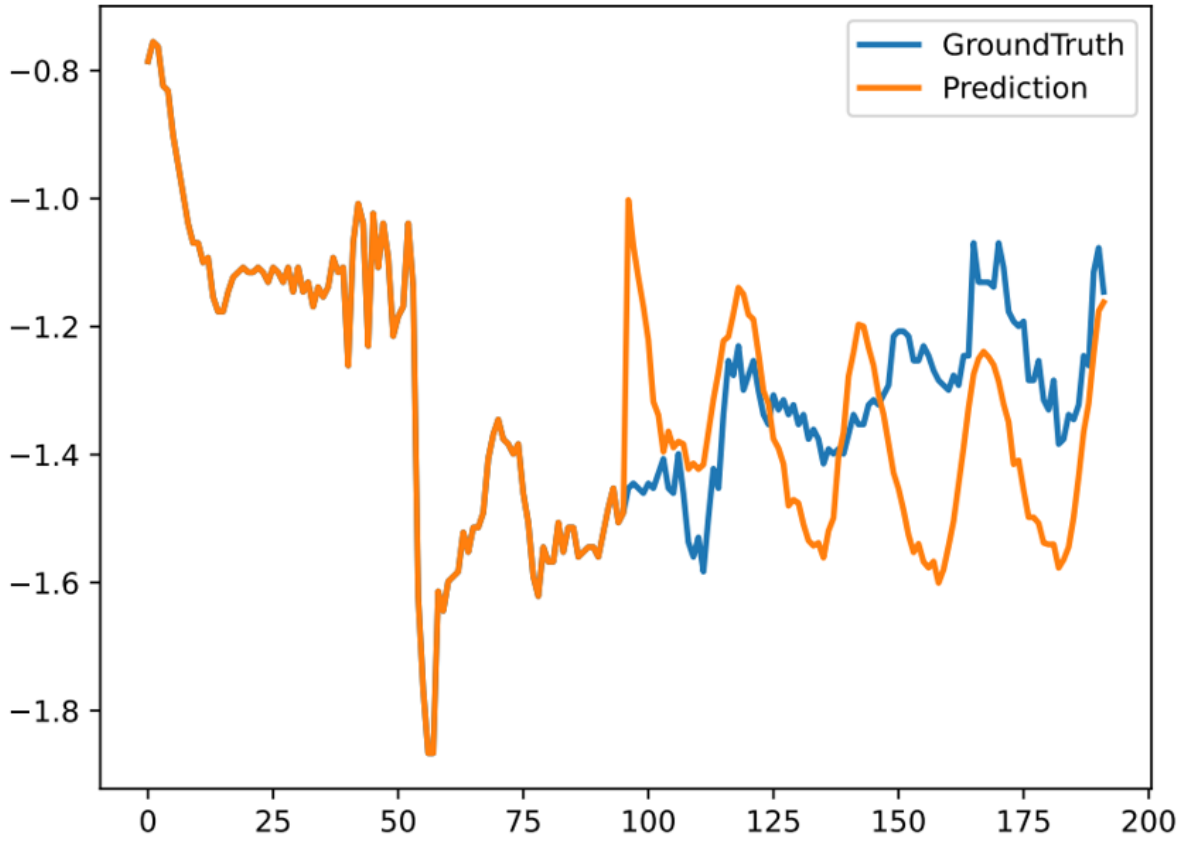

图 8: 云资源预测实验结果示意

6 总结与展望

本部分对整个文档的内容进行归纳并分析目前实现过程中的不足以及未来可进一步进行研究的方向。通过对 Fedformer 的复现加深了对 Attention 机制的了解。在复现过程中我们发现 Autoformer 和 FEDformer 结构非常相似，但是 Autoformer 是将序列分解为多个时域子序列进行特征提取，FEDformer 则是采用频率变换将序列分解为多个频域模态来提取特征。特别的是，FEDformer 在子序列选择中不使用选择性方法，而是所有的频率特征都是从整个序列中计算出来的，这种全局特性使模型对长序列

具有更好的性能。本文使用 FEDformer 进行云资源的预测，由于云资源时间序列具有高度非平稳性、高度非线性、变化极快的特点，因此难以实现准确预测。本文认为未来可以使用滤波器对处理后的任务时间序列进行平滑处理，消除可能的异常值和噪声，以此达到更高的准确性。

参考文献

[1] ZHOU T, MA Z, WEN Q, et al. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting[C]//Proc. 39th International Conference on Machine Learning (ICML 2022). Baltimore, Maryland, 2022.