A TIME SERIES IS WORTH 64 WORDS: LONG-TERM FORECASTING WITH TRANSFORMERS

Yuqi Nie,Nam H. Nguyen ,Phanwadee Sinthong,Jayant Kalagnanam

摘要

本文提出了一种基于 Transformer 的模型的有效设计，用于多元时间序列预测和自监督表示学习。它基于两个关键组件：(i) 将时间序列分割为子系列级别的补丁，作为 Transformer 的输入标记； (ii) 通道独立性，其中每个通道包含单个单变量时间序列，该序列在所有序列中共享相同的嵌入和 Transformer 权重。补丁设计自然具有三重好处：嵌入中保留局部语义信息；给定相同的回溯窗口，注意力图的计算和内存使用量会成倍减少；并且该模型可以参加更长的历史。与基于 SOTA Transformer 的模型相比，本文的通道独立补丁时间序列 Transformer (PatchTST) 可以显着提高长期预测精度。该模型被应用于自监督预训练任务，并获得出色的微调性能，其性能优于大型数据集上的监督训练。将一个数据集上的屏蔽预训练表示转移到其他数据集也能产生 SOTA 预测准确性。

关键词：长期序列预测；Transformer

1 引言

预测是时间序列分析中最重要的任务之一。随着深度学习模型的快速发展，有关该主题的研究工作数量显着增加。深度模型不仅在预测任务上表现出了出色的性能，而且在表示学习上也表现出了出色的性能，其中可以提取抽象表示并将其转移到各种下游任务（例如分类和异常检测）以获得最先进的性能。在深度学习模型中，Transformer 在自然语言处理（NLP）、计算机视觉（CV）、语音等各个应用领域取得了巨大成功。以及最近的时间序列，受益于其注意力机制，可以自动学习序列中元素之间的连接，因此成为顺序建模任务的理想选择。 Informer [1]、Autoformer [2]和 FEDformer [3]是成功应用于时间序列数据的 Transformer 模型的最佳变体。不幸的是，尽管基于 Transformer 的模型设计很复杂，但最近的论文表明，一个非常简单的线性模型可以在各种常见基准上优于所有以前的模型，并且它面临着挑战Transformer 对于时间序列预测的有用性。在本文中，通过提出一个通道无关的补丁时间序列变换器（PatchTST）模型来回答这个问题。

2 相关工作

2.1 基于 Transformer 模型中的Patch。

Transformer已经在不同的数据模式上展示了巨大的潜力。在所有应用中，当本地语义信息很重要时，Patch是必不可少的一部分。在 NLP 中，BERT虑基于子词的标记化，而不是执行基于字符的标记化。在 CV 中，Vision Transformer (ViT)是一项里程碑式的工作，它在输入 Transformer 模型之前将图像分割成 16×16 的块。

2.2 基于 Transformer 的长期时间序列预测。

近年来，有大量工作试图应用 Transformer 模型来预测长期时间序列。我们在这里总结了其中一些。 LogTrans使用具有 LogSparse 设计的卷积自注意力层来捕获局部信息并降低空间复杂度。 Informer提出了一种 ProbSparse 自注意力机制，通过蒸馏技术来有效地提取最重要的密钥。 Autoformer借鉴了传统时间序列分析方法的分解和自相关的思想。 FEDformer使用傅立叶增强结构来获得线性复杂度。 Pyraformer应用具有尺度间和尺度内连接的金字塔注意力模块，也获得了线性复杂度。

2.3 时间序列表征学习。

除了监督学习之外，自监督学习也是一个重要的研究课题，因为它显示出学习下游任务的有用表示的潜力。近年来提出了许多非基于 Transformer 的模型来学习时间序列的表示。与此同时，Transformer 被认为是基础模型和学习通用表示的理想候选者。

3 本文方法

3.1 本文方法概述

给定以下问题：给定具有回溯窗口 L 的多元时间序列样本集合：$(x_1, ..., x_L)$，其中时间步 t 的每个 x_t 是维度 M 的向量，我们希望预测 T 的未来值$(x_{L+1}, ..., x_{L+T})$。本文的PatchTST如图 1所示，其中该模型使用普通 Transformer 编码器作为其核心架构。

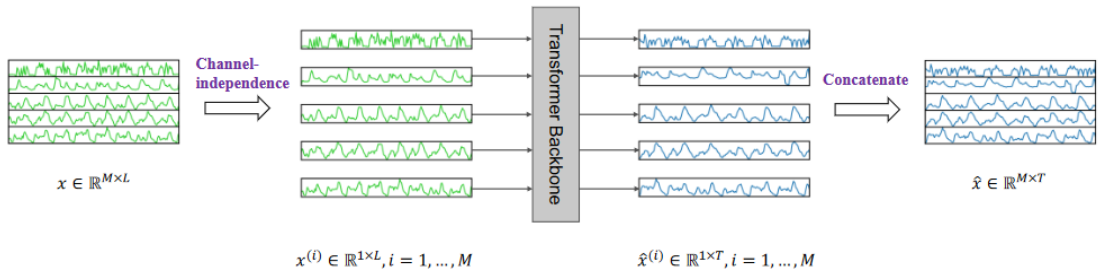

图 1. PatchTST模型示意图

3.2 Patching

每个输入单变量时间序列 $x^{(i)}$ 首先被划分为重叠或不重叠的Patch。将 patch 长度表示为 P，将步幅（两个连续 patch 之间的非重叠区域）表示为 S，则修补过程将生成一系列 patch $x_p^{(i)} \in \mathbb{R}^{PXN}$，其中 N 是 patch 的数量，$N = \lfloor (L-P)/S \rfloor + 2$。这里，我们在修补之前将最后一个值$x_L^{(i)} \in \mathbb{R}$ 的 S 个重复数填充到原始序列的末尾。通过使用Patch，输入标记的数量可以从 L 减少到大约 L/S。这意味着注意力图的内存使用和计算复杂度以 S 为二次方降低。因此，在训练时间和 GPU 内存的限制下，Patch设计可以让模型看到更长的历史序列，从而可以显着提高预测能力性能。

3.3 Transformer编码器

我们使用普通的 Transformer 编码器如图 2所示,将观察到的信号映射到潜在的表示。这些补丁通过可训练的线性投影$W_p \in \mathbb{R}^{DXP}$映射到维度 D 的 Transformer 潜在空间，并应用可学习的附加位置编码$W_pos \in \mathbb{R}^{DXN}$ 来监视补丁的时间顺序：$x_d^{(i)} = W_p x_p^{(i)} + W_{pos}$，其中 $x_d^{(i)} \in \mathbb{R}^{DXN}$ 表示将馈送到 Transformer 编码器的输入。

3.4 实例标准化

这是最近提出的技术，以帮助减轻训练数据和测试数据之间的分布偏移效应。它只是用零均值和单位标准差对每个时间序列实例 x(i) 进行归一化。本质上，我们在修补之前对每个 x(i) 进行归一化，并将平均值和偏差添加回输出预测中。

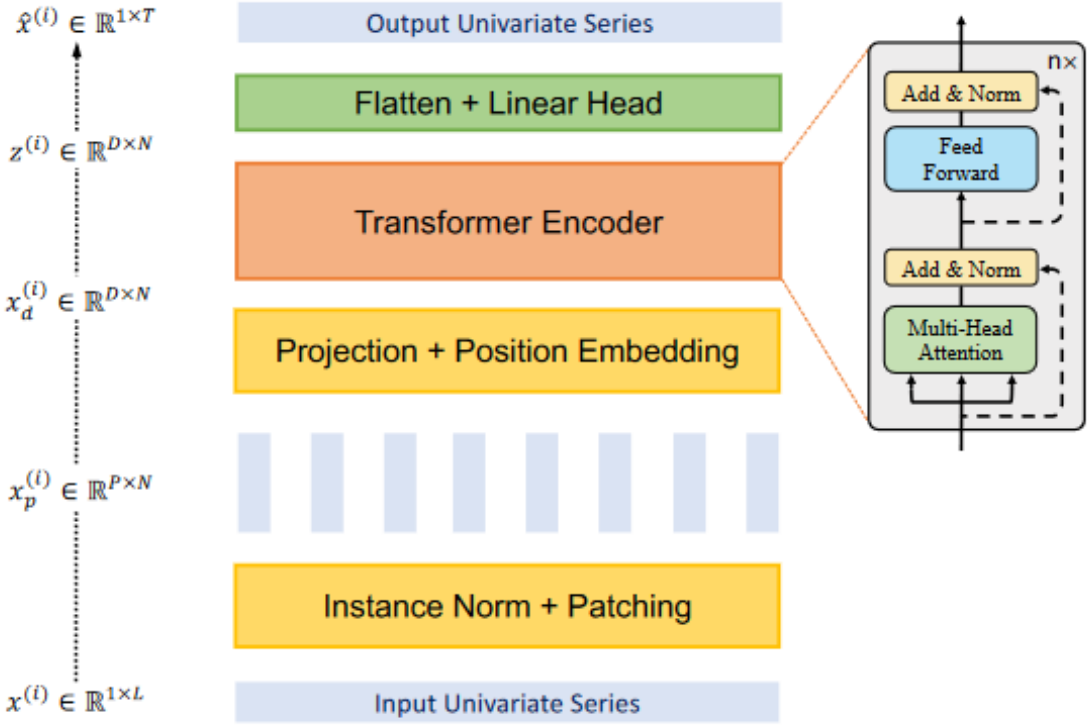

图 2. Tranformer Backbone

4 复现细节

4.1 与已有开源代码对比

本工作引用了 PatchTST 提供的代码，实现了 PatchTST 的整体框架。在框架的基础上修改了数据加载类，增加了一些数据处理方法，使其能够加载更多的数据集，更好地处理数据；原模型在时间复杂度上进行了优化，但实际运行起来在时间上的花费仍然很多，于是对模型架构进行了一些修改，尝试在Encoder，Decoder中更多地使用简单的线性层；对它的标准化方法也进行了一些修改，使其能更好地适应模型，能更多地适应各种数据集。

代码参考开源地址：https://github.com/yuqinie98/PatchTST

4.2 实验环境搭建

操作系统为 64 位 windows10，开发平台为 pycharm 2023 x64，使用 pycharm 远程服务器功能连接至服务器，服务器 python 环境 3.8，显卡配置 A40，设置生成随机数的种子参数为 2021，使用 torch 框架实现实验，torch 版本为 1.13.0+cu117 环境，使用 scikit-learn 工具包版本 1.02。

4.3 创新点

- Patching:将时间序列分割为子系列级别的补丁，作为 Transformer 的输入标记。
- 通道独立性:每个通道包含一个单变量时间序列，该序列在所有序列中共享相同的嵌入和 Transformer 权重。

5 实验结果分析

本次复现在ETTh1,ETTh2,Electricity三个数据集上进行了多次的实验如表1所示，ori_PatchTST为论文中原模型，PatchTST为复现模型。

1. ETT(Electricity Transformer Temperature)数据集：ETT 是电力长期部署的关键指标。该数据集采集了中国的两个独立县城为期 2 年的数据。其中 ETTh1 和 ETTh2 数据集为 1 小时采样一次。训练集/验证集/测试集分别按 12/4/4 个月来划分。

2. Electricity数据集：该数据是从澳大利亚新南威尔士州电力市场收集的。数据集的每个示例指的是30分钟的时间段，即一天的每个时间段有48个实例。数据集上的每个示例都有 5 个字段：星期几、时间戳、新南威尔士州电力需求、维多利亚州电力需求、各州之间的预定电力传输和类别标签。类别标签标识新南威尔士州相对于过去 24 小时移动平均值的价格变化

该实验结果表明：（1）在大多数情况下，PatchTST 和 ori_PatchTST 的 mse 和 mae 值比较接近，差异不大。这表明该复现在模型性能上与论文中的模型相当。（2）对于不同的数据集（例如 ETTh1、Electricity、ETTh2），两个模型在mse和mae上的表现相对一致。（3）在ETT数

Models	patchTST		ori_PatchTST	
Metric	mse	mae	mse	mae
ETTh1_96	0.375	0.399	0.37	0.4
ETTh1_192	0.413	0.421	0.413	0.429
ETTh1_336	0.431	0.435	0.422	0.44
ETTh1_720	0.449	0.465	0.447	0.468
Electricity_96	0.304	0.293	0.129	0.222
Electricity_192	0.442	0.319	0.147	0.24
Electricity_336	0.575	0.343	0.163	0.259
Electricity_720	0.375	0.399	0.197	0.29
ETTh2_96	0.291	0.338	0.274	0.337
ETTh2_192	0.388	0.409	0.341	0.382
ETTh2_336	0.414	0.427	0.329	0.384
ETTh2_720	0.423	0.444	0.379	0.422

表 1. 复现结果

据集的表现上，复现模型在某些情况下比论文中的模型更有效。而在Electricity数据集上，复现模型普遍弱于论文中的模型。

总体而言，该复现在各种设置下都能够捕捉到模型的性能。

此外，本工作还尝试去验证论文中的创新点：通道独立性，使用一个新的海洋数据集进行单通道或多通道的输出，并与Informer模型进行对比。使用的新的数据集是用于海表预测的，该数据集对6个变量进行预测，分别是海表面高度(adt),显著波高(swh),海表温度（mwsst）,纬向风(uwnd),经向风(vwnd)，海表盐度（sss）。该实验验证了论文中的通道独立性。但是该实验表明，在该数据集上，PatchTST的效果不如Informer，可能需要进一步设置PatchTST的相关参数才能得到更好的效果。

Model	**Informer**				**PatchTST**			
	单通道输出		多通道输出		单通道输出		多通道输出	
adt	0.27	0.349	0.275	0.355	0.456	0.462	0.395	0.427
swh	0.59	0.517	0.589	0.517	0.924	0.654	0.933	0.65
mwsst	0.0007	0.015	0.002	0.027	1.144	0.703	1.133	0.701
uwnd	16.951	2.973	16.891	2.967	18.637	3.161	17.845	3.085
vwnd	18.1	3.097	17.969	3.081	19.145	3.208	18.617	3.153
sss	0.024	0.0846	0.026	0.096	0.42	0.445	0.389	0.428

表 2. 实验结果

6 总结与展望

本文针对时间序列预测的挑战，复现了一种基于 Transformer 的 PatchTST 模型。通过将时间序列数据划分为补丁，该模型在多个时间序列数据集上展现出了显著的预测精度和计算效率。PatchTST 模型利用其独特的补丁方法和通道独立特性，成功地捕获了时间序列数据中的长期依赖关系和局部特征，有效地提升了预测的准确性。

在实验中，对比了 PatchTST 模型与当前先进的时间序列预测模型。结果表明，无论是在短期还是长期预测任务中，PatchTST 都能取得更好的性能，尤其是在处理具有复杂模式和噪声的时间序列数据时表现出了优越的鲁棒性。

尽管 PatchTST 模型在多个方面表现优异，但仍存在一些局限性和挑战。首先，模型的参数调整对于不同数据集来说是一项挑战，需要通过大量的实验来寻找最优的补丁长度和步长。其次，对于极端情况下的时间序列数据，如非常不规则的数据，模型的预测精度仍有待提高。

未来的研究可以从以下几个方面进行：

1. 参数优化：研究更高级的算法来自动调整补丁长度和步长，以提高模型在不同类型的时间序列数据上的适应性和精确度。

2. 模型结构改进：探索将 PatchTST 模型与其他深度学习方法结合的可能性，例如集成学习或多任务学习，以进一步提升模型的性能和泛化能力。

3. 应用范围拓展：探索 PatchTST 在其他领域的应用，如金融市场分析、气候模式预测等，以验证和扩展其应用范围。

4. 自监督学习的运用：研究如何利用自监督学习技术来提前训练 PatchTST 模型，以便更好地处理大规模的未标记时间序列数据。

随着技术的进步和数据量的增加，时间序列预测的方法和应用将持续发展。PatchTST 模型作为一种新兴的方法，有望在未来的研究中发挥更大的作用，尤其是在解决实时预测和处理大规模数据集的问题上。随着人工智能和机器学习技术的不断进步，可以预期 PatchTST 模型将在准确性和效率上得到进一步的提升。

参考文献

[1] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[2] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[3] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pages 27268–27286. PMLR, 2022.