A Time Series is Worth 64 Words: Long-term Forecasting with Transformers

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

1) Introduction

The sequential nature of time series data makes Transformers a promising architecture for time series forecasting. Previous research papers have led to the development of Transformer-based forecasting models like FEDformer, Autoformer, and Informer. However, a recent paper (Zeng et al., 2022) shows that a very simple linear model can **outperform** the previously mentioned Transformer-based models in terms of MSE and MAE when applied to benchmark time series datasets covering electricity consumption, illnesses, traffic, and more. This challenges the usefulness of Transformer-based forecasting models, as they are computationally expensive and perform worse than simple linear models. Guided by this problem, our paper aims to **find improvements to Transformer-based models so that they outperform simple linear models**.

There are two key designs proposed in **PatchTST** that improve performance and reduce computational complexity. The first design choice is **patching**, where time series data is segmented into subseries-level patches that serve as input tokens to the Transformer. This allows the model to capture local semantic information – something overlooked by previous works, which only use point-wise input tokens. The second design feature is **channel-independence**, where a multivariate time series is broken into multiple univariate time series and fed individually into the transformer. Note, however, that each univariate time series shares the same embedding and Transformer weights to create a model that generalizes better.

2) Chosen Result

We aim to reproduce the results of applying **supervised** PatchTST/64 to five datasets from the paper. Specifically, we will only replicate the rows labeled with 96 (prediction length=96) or 24 for the ILI dataset. The prediction length for the ILI dataset is shortened since it is much smaller than the other datasets.

Models		PatchTST/64		PatchTST/42		DLinear		FEDformer		Autoformer		Informer		Pyraformer		LogTrans	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	96	**0.360**	**0.249**	0.367	0.251	0.410	0.282	0.576	0.359	0.597	0.371	0.733	0.410	2.085	0.468	0.684	0.384
	192	**0.379**	**0.256**	0.385	0.259	0.423	0.287	0.610	0.380	0.607	0.382	0.777	0.435	0.867	0.467	0.685	0.390
	336	**0.392**	**0.264**	0.398	0.265	0.436	0.296	0.608	0.375	0.623	0.387	0.776	0.434	0.869	0.469	0.734	0.408
	720	**0.432**	**0.286**	0.434	0.287	0.466	0.315	0.621	0.375	0.639	0.395	0.827	0.466	0.881	0.473	0.717	0.396
Electricity	96	**0.129**	**0.222**	0.130	**0.222**	0.140	0.237	0.186	0.302	0.196	0.313	0.304	0.393	0.386	0.449	0.258	0.357
	192	**0.147**	**0.240**	0.148	**0.240**	0.153	0.249	0.197	0.311	0.211	0.324	0.327	0.417	0.386	0.443	0.266	0.368
	336	**0.163**	**0.259**	0.167	0.261	0.169	0.267	0.213	0.328	0.214	0.327	0.333	0.422	0.378	0.443	0.280	0.380
	720	**0.197**	**0.290**	0.202	0.291	0.203	0.301	0.233	0.344	0.236	0.342	0.351	0.427	0.376	0.445	0.283	0.376
ILI	24	**1.319**	**0.754**	1.522	0.814	2.215	1.081	2.624	1.095	2.906	1.182	4.657	1.449	1.420	2.012	4.480	1.444
	36	1.579	0.870	**1.430**	**0.834**	1.963	0.963	2.516	1.021	2.585	1.038	4.650	1.463	7.394	2.031	4.799	1.467
	48	**1.553**	**0.815**	1.673	0.854	2.130	1.024	2.505	1.041	3.024	1.145	5.004	1.542	7.551	2.057	4.800	1.468
	60	**1.470**	**0.788**	1.529	0.862	2.368	1.096	2.742	1.122	2.761	1.114	5.071	1.543	7.662	2.100	5.278	1.560
ETTh1	96	**0.370**	0.400	0.375	**0.399**	0.375	**0.399**	0.376	0.415	0.435	0.446	0.941	0.769	0.664	0.612	0.878	0.740
	192	0.413	0.429	0.414	0.421	**0.405**	**0.416**	0.423	0.446	0.456	0.457	1.007	0.786	0.790	0.681	1.037	0.824
	336	**0.422**	0.440	0.431	**0.436**	0.439	0.443	0.444	0.462	0.486	0.487	1.038	0.784	0.891	0.738	1.238	0.932
	720	**0.447**	0.468	0.449	**0.466**	0.472	0.490	0.469	0.492	0.515	0.517	1.144	0.857	0.963	0.782	1.135	0.852
ETTm1	96	0.293	0.346	**0.290**	**0.342**	0.299	0.343	0.326	0.390	0.510	0.492	0.626	0.560	0.543	0.510	0.600	0.546
	192	0.333	0.370	**0.332**	0.369	0.335	**0.365**	0.365	0.415	0.514	0.495	0.725	0.619	0.557	0.537	0.837	0.700
	336	0.369	0.392	**0.366**	0.392	0.369	**0.386**	0.392	0.425	0.510	0.492	1.005	0.741	0.754	0.655	1.124	0.832
	720	**0.416**	**0.420**	0.420	0.424	0.425	0.421	0.446	0.458	0.527	0.493	1.133	0.845	0.908	0.724	1.153	0.820

Like the paper, our goal is to obtain a smaller MSE and MAE on these datasets than both the simple linear model (DLinear) and the Transformer-based models proposed by previous work. Doing so would prove that PatchTST outperforms all previous forecasting models while reducing computational complexity compared to the other Transformer-based models.

3) Methodology

Our re-implementation of supervised PatchTST/64 is **nearly identical** to the model architecture described in the paper. Channel splitting is first applied, turning a multivariate time series into multiple univariate time series. Instance norm and patching (patch length P=16) are then applied to each univariate time series. The resulting patches are mapped to the Transformer latent space, and a learnable positional encoding is added. We then feed the tokens into the Transformer encoder consisting of 3 layers. Finally, the outputs are fed through a flatten layer and each channel is concatenated to recreate a multivariate prediction. The predictions are fed through the MSE loss function to train the model.

The five datasets we used entailed electricity consumption for 321 households (Electricity), influenza patients (ILI), road occupancy rates in San Francisco (Traffic), and powerload data from China (ETTh1, ETTm1). Because the datasets varied in size, the transformer architecture varied between datasets. The ILI dataset was much **smaller** than the others, leading the paper to use only 4 heads in the encoder layer compared to the 16 heads used in the other PatchTST/64 models.

We display MSE and MAE to compare PatchTST/64 with DLinear and other Transformer-based models.

4) Results & Analysis

		Our PatchTST/64		Paper PatchTST/64		DLinear		FEDformer		Autoformer		Informer		Pyroformer	
Dataset	**Pred Length**	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	96	0.421	0.295	**0.360**	**0.249**	0.410	0.282	0.576	0.359	0.597	0.371	0.733	0.410	2.085	0.468
Electricity	96	0.156	0.258	**0.129**	**0.222**	0.140	0.237	0.186	0.302	0.196	0.313	0.304	0.393	0.386	0.449
ILI	24	2.071	0.974	**1.319**	**0.754**	2.215	1.081	2.624	1.095	2.906	1.182	4.657	1.449	1.420	2.012
ETTh1	96	0.427	0.452	**0.370**	0.400	0.375	**0.399**	0.376	0.415	0.435	0.446	0.941	0.769	0.664	0.612
ETTm1	96	0.326	0.372	**0.293**	0.346	0.299	**0.343**	0.326	0.390	0.510	0.492	0.626	0.560	0.543	0.510

* Smallest errors across all models are **bolded**. Datasets on which our PatchTST/64 outperformed all models except the paper's PatchTST/64 are underlined.

The original paper found that PatchTST/64 outperforms DLinear and the other Transformer-based models on nearly all datasets across MSE and MAE. Our re-implementation results show similar findings. While our PatchTST/64 does not outperform DLinear on as many datasets, we achieved smaller losses than the Transformer-based models on almost all datasets. We attribute the slightly lower performance of our PatchTST/64 to **compute limitations** that we faced during training. On the Traffic and Electricity models, we trained for **only 3 epochs** compared to the 100 epochs in the paper. For the ETTh1 and ETTm1 models, we trained for **only 25 epochs** compared to the 100 epochs in the paper. As for the ILI model, we trained the same number of epochs as the paper (100 epochs). **Note that this is the only model we implemented that outperformed DLinear and the other Transformer-based models.**

Our results show that including patching and channel-independence noticeably improves the performance of Transformer-based forecasting models. With these new design features, Transformers can outperform all previous forecasting models while reducing computational complexity. Although the paper written by Zeng et al. casts doubt on the usefulness of Transformers for time series forecasting, the results from our re-implementation show promise in this area.

5) Reflections

The biggest lesson learned from this project is that **RAM storage** is a large concern when training models on small GPUs. Training the Transformer with even a few encoder layers demanded significant computation due to the sheer number of parameters to learn. We often found ourselves constrained by computational limitations, which forced us to experiment with simpler models and fewer training epochs. Once we began training the models, we noticed that **regularization techniques**, specifically dropout, early stopping, and channel-independence, noticeably increased the performance of our models. We were surprised by the extent to which these techniques helped reduce testing loss – a key takeaway from our re-implementation effort.

As for potential next steps, we would like to investigate the **transfer learning** capabilities of PatchTST. Our paper experiments with pre-training the model on the Electricity dataset before fine-tuning the pre-trained model for the other datasets. Their experiments show that forecasting accuracy is improved while significantly **reducing computation time**. Given more time to experiment, we would like to reproduce these results.

6) References

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. *arXiv:2211.14730, 2023.*

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers Effective for Time Series Forecasting? *arXiv preprint arXiv:2205.13504, 2022.*

Trindade, A. (2015). ElectricityLoadDiagrams20112014 [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C58C86.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). *Informer: Beyond efficient transformer for long sequence time-series forecasting*. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 12, pp. 11106–11115)