Transformer-based time series classification versus alternative approaches with a focus on interpretability

Juan Kostelec
juank@student.ethz.ch

Raffael Künzi
rakuenzi@student.ethz.ch

Nicole Sanche
nsanche@student.ethz.ch

Abstract

Many models for time series classification (TSC) or prediction rely on shallow models, often leveraging domain expertise to extract features. One of the reasons for choosing these traditional approaches remains interpretability. Only recently (Fawaz et al. 2019) conducted an extensive review of state-of-the-art DL methods for TSC and benchmarked them on the UCR/UEA archive (Dau et al. 2019). Overall best performance could be achieved with Residual Networks (ResNet) and Fully Convolutional Neural Networks (FCN) which, due to the use of global average pooling, allow the use of 1D Class Activation Maps (CAMs) to highlight the areas of the TS that contributed most to a certain classification. Transformer-based models have hardly been considered for TSC so far. This paper aims to close this gap by benchmarking two slightly different transformer-based DL architectures against ResNet and FCN for univariate time series. A specific focus is put on explaining differences in the performance of the investigated architectures using visualization techniques like CAMs (Zhou et al. 2016) and Attention Maps (AMs) (Clark et al. 2019).

1 Introduction

Time series (TS) data is prevalent in just about every business and scientific field. Basically any classification task requiring ordering can be modeled as a TSC problem. In the past TS classification and prediction was dominated by shallow models requiring extensive data preprocessing and feature engineering. Only recently, helped by visualization techniques mitigating their black-box effect (Fawaz et al. 2019), end-to-end DL models like ResNet and FCN gained acceptance as state-of-the-art data agnostic baselining approaches for TSC. Transformer-based architectures are popular in natural language processing (NLP) but rather new to time series analysis. Recent work rather focuses on TS forecasting (Li et al. 2019; Wu et al. 2020) than classification (Rußwurm and Körner 2020). In this paper we build on the foundation work of Fawaz et al. 2019; Wang, Yan, and Oates 2017 by adding two different transformer-based architectures to their benchmark study of DL methods on the UCR univariate time series archive (`https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`).

In summary, our contributions are as follows:

- We adapted two different transformer-based architectures, one with Post-Layer Normalization (Post-LN) (Vaswani et al. 2017) and one with Pre-Layer Normalization (Pre-LN) (Xiong et al. 2020), to the use case of univariate TSC.
- We benchmarked the transformer-based architectures against ResNet and FCN on a subset of the UCR TS archive leveraging the approach applied by Fawaz et al. 2019; Wang, Yan, and Oates 2017.
- We used data set characteristics (e.g. theme, length of time series, size of training set) and visualization techniques (CAMs and AMs) to interpret and explain differences in the performance of the investigated architectures for selected data sets.

- Our results do not support the hypothesis that transformer-based models perform comparatively well in classification of TS with long range dependencies. This might be due to the specific nature of the datasets in the UCR archive which are short in length compared e.g. to the satellite data used by Rußwurm and Körner 2020 for their work on transformer-based TSC models.

2 Background on models and methods

The following section provides a high level overview of the investigated DL models and the chosen experimental set-up. A more detailed description of the individual models can be found in Fawaz et al. 2019; Wang, Yan, and Oates 2017 for ResNet and FCN and in Vaswani et al. 2017; Xiong et al. 2020 for the different transformer-based models.

2.1 Fully Convolutional Neural Networks (FCN)

In our setting the FCN is made up of three basic one-layer convolutional blocks (batch normalization layer and Relu activation layer, filters per block {128, 256, 128}) followed by a global average pooling (GAP) and a softmax layer. The detailed experimental set-up is as follows: batch size: 16, number or epochs: 2000, optimizer: Adam with learning rate 1e-3).

2.2 Residual Networks (ResNet)

The ResNet implementation has by far the deepest structure of all applied models. Three residual blocks, each consisting of three identical convolutional layers (batch normalization layer and Relu activation layer, filters per block {64, 128, 128}), are followed by a global average pooling (GAP) and a softmax layer. The detailed experimental set-up is as follows: batch size: 16, number of epochs: 1500, optimizer: Adam with learning rate 1e-3).

2.3 Post-Layer Normalization Transformer (TPost-LN)

The applied Post-Layer transformer (TPost-LN) relies on the original encoder stack architecture for a single block transformer originally introduced by Vaswani et al. 2017. The first sublayer consists of a single-head self-attention mechanism, the second one of a fully connected feed-forward network. The residual connection applied to each of the two sublayers is followed by layer normalization which explains the naming. The actual transformer block is followed by a max pooling and a softmax layer. The detailed experimental set-up is as follows: batch size: 16, number of epochs: 2000, optimizer: Adam with with a learning rate warm-up of 100 epochs and square root decay after).

2.4 Pre-Layer Normalization Transformer (TPre-LN)

As a sanity check and mitigation approach for potential warm-up stage issues with the TPost-LN, we also implemented an alternative transformer architecture with Pre-Layer normalization (Pre-LN). The set-up is basically similar to the TPost-LN with the exception that the layer normalization is done inside the residual connections of the self-attention and feed-forward network sublayers (see Xiong et al. 2020). The detailed experimental set-up is as follows: batch size: 16, number of epochs: 2000, Adam with learning rate 1e-3).

2.5 Class Activation (CAMs) and Attention Maps (AMs)

The global average pooling layers of FCN and ResNet allow for the use of CAMs to identify the contributing regions in the raw data for a specific label and thus to facilitate the visualization of the predicted class score. The class activation map M_c for a given class c is given by

$$M_c = \sum_c \omega_k^c S_k(x) \tag{1}$$

with $S_k(x)$ representing the activation of filter k in the last convolutional layer at temporal location x and ω_k^c the weight of the final softmax function for filter k and class c (Wang, Yan, and Oates 2017).

The natural choice for visualization of the outputs of the transformer-based models are attention maps (AMs). The self-attention head computes attention weights $\alpha_{i,j}$ between all pairs of input vectors as softmax-normalized dot products between the query (q) and key vectors (k). The output o of the attention head is given by the weighted sum of the value vectors (v) (equation (2)), (Clark et al. 2019). Each point on the derived AM corresponds to the average attention a particular head is putting toward a specific input vector (meaning a specific point in the time series).

$$\alpha_{i,j} = \frac{exp(q_i^T k_j)}{\sum_{l=1}^{n} exp(q_i^T k_l)} \quad \text{and} \quad o_i = \sum_{j=1}^{n} \alpha_{i,j} v_j \tag{2}$$

3 Experiments and results

3.1 Performance on UCR TS Archive

We tested all of the above described four DL architectures on a subset of the UCR TS archive. We trained each model five times for the above specified number of epochs and chose the testing accuracy of the epoch with the lowest training loss per run. The only preprocessing step applied to all raw data was an upfront z-normalization per time series.

Table 1 provides a summary of all test results including the name of the respective data set, the number of classes, the number of samples in the training/test set, the length of the time series and achieved testing accuracy (mean and standard deviation) per DL architecture. ResNet achieved highest testing accuracy on 32 out of the 44 datasets tested. The two transformer-based models significantly fell behind ResNet and FCN on testing accuracy. Additional experiments with a tuned TPost-LN with four attention heads showed slightly higher testing accuracies (on average 4% above the accuracy of the TPost-LN with one attention head) but still significantly below the ResNet and FCN results. TPost-LN achievd in average lower accuracies than TPre-LN, but using a fixed instead of a scheduled learning rate on the TPost-LN lead to comparable results.

3.2 Visualization of feature importance

In (Fig. 1) CAMs for ResNet and AMs for TPre-LN are visualized to highlight the contribution of each time series region to the class identification on the 'Coffee' dataset. This particular dataset has been chosen as illustration of properties observed for most of the datasets. Transformer-based models typically focus on a single or very few regions of the signal (in extreme cases a region can be one single time step), whereas ResNet and FCN use multiple regions for feature extraction as highlighted by the respective CAMs. For the 'Coffee' dataset the TPre-LN seems to attend mostly to the region where the main differences between the two classes are located, whereas the CAMs for ResNet also highlight some additional regions. This might explain why ResNet achieved 100% accuracy whereas TPre-LN only achieved 95.7%.

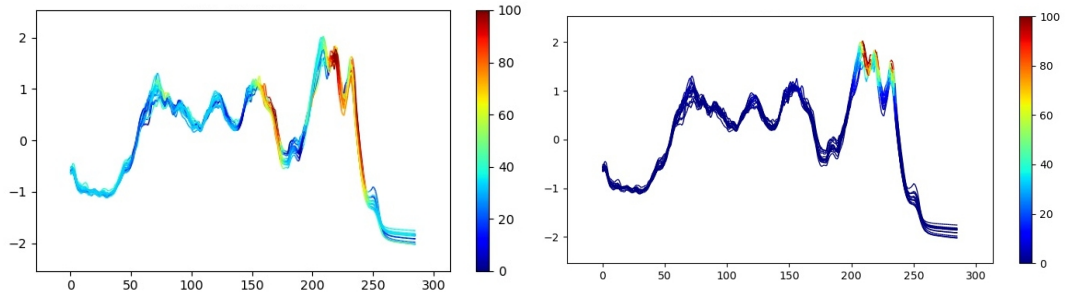
In (Fig. 2) a subset of the 'OSULeaf' dataset is visualized. While ResNet scores 98.7%, TPre-LN achieves only 52.0%. The high attention on one small area of the TPre-LN seems not to be sufficient, contrary to the ResNet which found features across the whole length of the signal.

4 Discussion and summary

We demonstrated that basic transformer-based models are inferior to best-in-class DL models (ResNet and FCN) for short-term time series forecasting on a subset of the UCR data archive. Adding additional complexity to the transformer-based models (via increasing the number of attention heads for the TPost-LN from one to four heads) slightly improved testing accuracy but scores remained still significantly below those of ResNet and FCN. These results are in line with the findings of Vaswani et al. 2017 who also allude to the fact that increasing the number of attention heads and transformer blocks doesn't always result in the desired performance improvements. CAMs and AMs proved to be effective visualization tools to explain the performance differences of the tested models. While transformers are known to handle long range relation in the data well (by their very construction they attend to the whole time series at the same time), our results do not show any consistent relative out-performance of the transformer in the longer TS. Suggestion for future research would be to create synthetic data of different characteristics (e.g. noisy, seasonal, ...) to further explore the

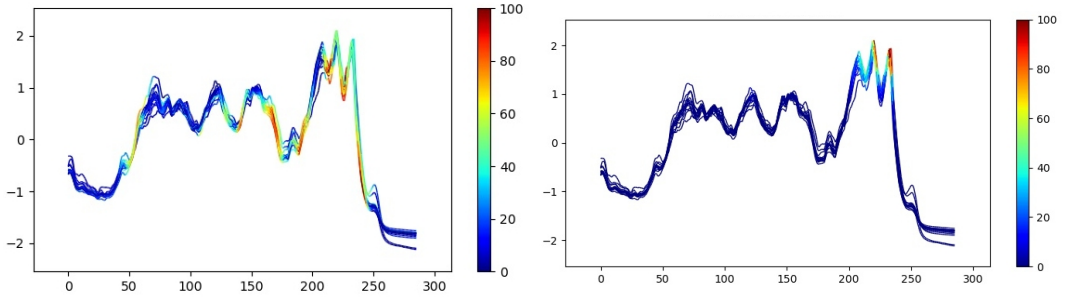
Table 1: Average testing accuracy on 44 UCR time series dataset

Test set characteristics							
Name	C	Train/Test	Lgth	FCN	ResNet	TPost-LN	TPre-LN
50words	50	450/455	270	66.1(0.5)	**74.0**(1.7)	59.0(3.8)	68.7(1.9)
Adiac	37	390/391	176	**84.7**(0.5)	81.1(1.2)	41.3(1.9)	66.0(4.1)
Beef	5	30/30	470	**82.7**(6.0)	79.3(6.0)	72.7(4.9)	68.0(1.8)
CBF	3	30/900	128	98.7(0.8)	**99.6(0.3)**	96.5(0.8)	96.6(3.0)
ChlorineCon.	3	467/3840	166	83.3(0.5)	**83.5**(0.8)	70.4(8.8)	76.8(2.7)
CinCECGtorso	4	40/1480	1639	84.7(1.0)	86.2(2.0)	**87.0**(4.0)	82.6(6.1)
Coffee	2	28/28	286	**100.0**(0.0)	**100.0**(0.0)	97.9(2.0)	95.7(3.0)
Cricket_X	12	390/390	300	78.2(1.2)	**79.8**(1.4)	49.7(4.2)	60.2(2.5)
Cricket_Y	12	390/390	300	77.4(1.5)	**80.4**(0.8)	51.1(3.5)	60.4(2.3)
Cricket_Y	12	390/390	300	78.8(1.1)	**80.3**(1.7)	52.9(3.1)	62.3(1.8)
DiatomSizeR	4	16/306	345	93.3(0.5)	93.2(1.0)	**94.5**(2.6)	92.8(1.7)
ECGFiveDays	2	23/861	136	98.6(0.5)	**99.8**(0.4)	85.1(3.0)	85.8(8.6)
Fish	7	175/175	463	95.7(1.2)	**98.4**(0.5)	72.6(4.1)	80.2(3.4)
Face (all)	14	560/1690	131	**94.8**(0.6)	83.2(4.1)	73.6(4.5)	74.3(2.3)
Face (four)	4	24/88	350	94.3(0.8)	**95.2**(0.5)	85.9(5.4)	86.4(4.3)
FacesUCR	14	200/2050	131	94.2(0,5)	**95.2**(0.3)	80.3(1.9)	82.4(2.1)
Gun-Point	2	50/150	150	**99.9**(0.3)	99.3(0,5)	79.7(1.2)	93.5(5.0)
Haptics	5	155/308	1092	49.7(0.0)	**54.3**(1.3)	41.9(1.7)	42.0(1.7)
Inlineskate	7	100/550	1882	**42.5**(2.4)	41.9(8.0)	28.2(1.0)	31.7(2.3)
ItalyPowDem	2	67/1029	24	96.1(0.2)	**96.3**(0.6)	94.9(0.6)	95.4(0.9)
Lightning2	2	60/61	637	71.8(0.7)	**76.1**(2.5)	70.8(2.9)	70.2(3.2)
Lightning7	7	70/73	31	81.6(4.1)	**82.5**(2.5)	69.6(6.5)	71.0(4.4)
MALLAT	8	55/2345	1024	96.6(0.5)	**97.0**(0.4)	89.9(4.1)	90.8(4.3)
MedicalImg	10	381/760	99	77.6(0.6)	**77.9**(1.1)	69.7(2.6)	71.4(2.6)
MoteStrain	2	20/1252	84	91.8(1.4)	**92.6(1.6)**	86.4(2.5)	87.3(2.3)
NonInvECG1	42	1800/1965	750	**96.3**(0.2)	93.7(0.3)	78.9(1.7)	89.5(1.8)
NonInvECG2	42	1800/1965	750	**95.6**(0.3)	94.0(0.4)	86.9(2.3)	92.2(0.4)
OSULeaf	6	200/242	427	97.5(0.4)	**98.7**(0.3)	45.6(6.9)	52.0(3.6)
Olive	4	30/30	570	73.3(9.7)	**86.0**(1.5)	69.3(7.6)	70.0(7.1)
SonyAIBO	2	20/601	70	96.8(1.2)	**97.4**(0.7)	74.2(4.5)	75.7(5.3)
SonyAIBOII	2	27/953	65	**97.8**(0.7)	97.4(0.7)	76.7(5.3)	77.4(4.7)
StarLightC	3	1000/8236	1024	96.1(0.4)	**97.5**(0.2)	94.7(0.6)	95.8(0.3)
SwedishLeaf	15	500/625	128	**96.6**(0.4)	95.6(0.3)	80.3(3.3)	86.4(2.0)
Symbols	6	25/995	398	97.1(1.3)	**97.4**(1.5)	84.1(5.3)	86.9(2.2)
Trace	4	100/100	275	**100**(0.0)	**100**(0.0)	99.8(0.4)	98.8(0.8)
TwoLeadECG	2	23/1139	82	**100**(0.0)	99.9(0.1)	80.6(6.9)	82.4(9.4)
TwoPatterns	4	1000/4000	128	88.3(0.3)	**100**(0.0)	98.7(0.3)	99.3(0.6)
WordSynonyms	25	267/638	270	56.1(0.9)	**63.2**(1.1)	53.4(3.5)	56.2(2.6)
SynthControl	6	300/300	60	98.9(0.1)	**99.4**(0.4)	97.3(1.0)	98.6(0.8)
uWaveGstLib_X	6	896/3582	315	75.8(0.5)	**78.7**(0.2)	75.4(1.8)	76.7(0,4)
uWaveGstLib_Y	6	896/3582	315	65.2(0.3)	**68.4**(0.7)	65.8(0.5)	67.2(0.7)
uWaveGstLib_Z	6	896/3582	315	73.1(0.4)	**75.6**(0.9)	66.8(0.7)	69.2(0.8)
Wafer	2	1000/6174	152	99.7(0.0)	**99.8**(0.1)	99.4(0.2)	99.5(0.1)
Yoga	2	300/3000	426	85.5(0.9)	**85.5**(0.5)	75.3(2.9)	81.3(1.7)
Number wins				12	32	2	0
AVG rank(arith.)				1.932	1.273	3.705	3.045



(a) ResNet on Coffee: Class 1

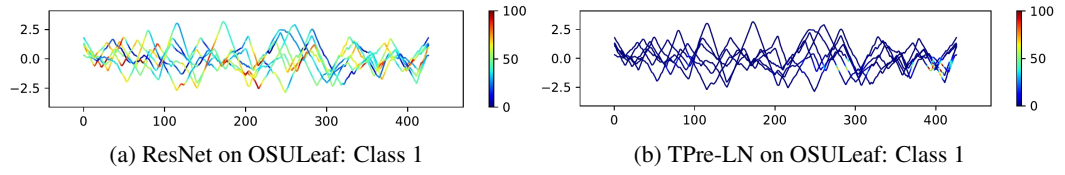
(b) TPre-LN on Coffee: Class 1



(c) ResNet on Coffee: Class 2

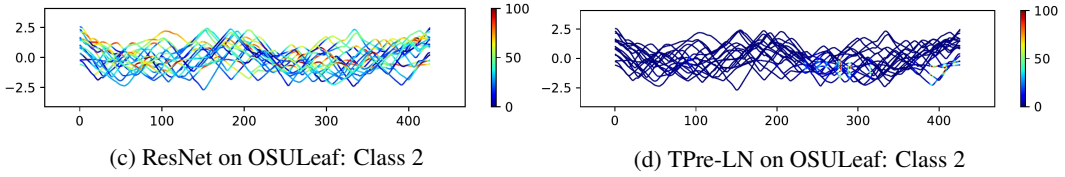
(d) TPre-LN on Coffee: Class 2

Figure 1: Using Class Activation Map (for ResNet) and the attention values (for TPre-LN) to highlight the importance of different parts of the signal on the class identification. Red corresponds to high contribution and blue to lower contribution.



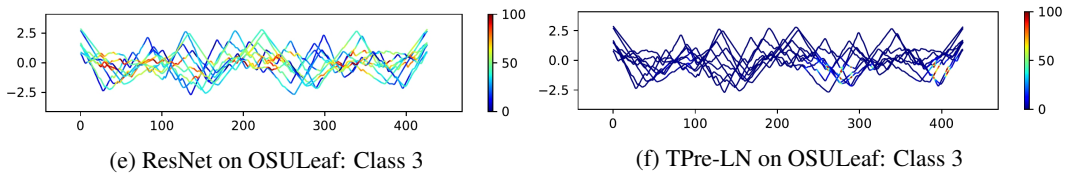
(a) ResNet on OSULeaf: Class 1

(b) TPre-LN on OSULeaf: Class 1



(c) ResNet on OSULeaf: Class 2

(d) TPre-LN on OSULeaf: Class 2



(e) ResNet on OSULeaf: Class 3

(f) TPre-LN on OSULeaf: Class 3

Figure 2: Class Activation Map (for ResNet) and attention values (for TPre-LN) on a subset of the OSULeaf dataset. Displayed are only 3 of the 6 classes.

behaviour of transformers under different regimes. Indeed, the tendency of transformer-based models to attend to only one or very few areas of the underlying time series might turn out to be a benefit in the case of very noisy raw data. Finally, training the transformer models has been a bit challenging, with the training loss either exploding during training or converging to a local optimum, so a further topic to explore would be the impact of the learning rate schedule on the transformer's performance.

References

[Cla+19] Kevin Clark et al. "What does bert look at? an analysis of bert's attention". In: *arXiv preprint arXiv:1906.04341* (2019).

[Dau+19] Hoang Anh Dau et al. "The UCR time series archive". In: *IEEE/CAA Journal of Automatica Sinica* 6.6 (2019), pp. 1293–1305.

[Faw+19] Hassan Ismail Fawaz et al. "Deep learning for time series classification: a review". In: *Data Mining and Knowledge Discovery* 33.4 (2019), pp. 917–963.

[Li+19] Shiyang Li et al. "Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019, pp. 5243–5253. URL: https://proceedings.neurips.cc/paper/2019/file/6775a0635c302542da2c32aa19d86be0-Paper.pdf.

[RK20] Marc Rußwurm and Marco Körner. "Self-attention for raw optical Satellite Time Series Classification". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 169 (2020), pp. 421–435. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2020.06.006. URL: http://www.sciencedirect.com/science/article/pii/S0924271620301647.

[Vas+17] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[Wu+20] Neo Wu et al. "Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case". In: *arXiv preprint arXiv:2001.08317* (2020).

[WYO17] Zhiguang Wang, Weizhong Yan, and Tim Oates. "Time series classification from scratch with deep neural networks: A strong baseline". In: *2017 International joint conference on neural networks (IJCNN)*. IEEE. 2017, pp. 1578–1585.

[Xio+20] Ruibin Xiong et al. "On Layer Normalization in the Transformer Architecture". In: *arXiv e-prints*, arXiv:2002.04745 (Feb. 2020), arXiv:2002.04745. arXiv: 2002.04745 [cs.LG].

[Zho+16] Bolei Zhou et al. "Learning Deep Features for Discriminative Localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.