Towards More Accurate Time Series Forecasting with Variance-aware Loss Functions in Transformers

Collin Drake[1] and **Jack Cerullo**[1]

[1]Peak to Peak Charter School

March 2024

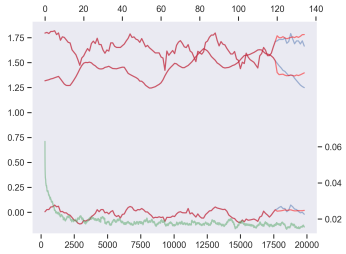
Abstract

When forecasting time series data, transformer models predict sequences lacking in volatility, exhibiting significant bias. We hypothesize that transformer models do so because of their loss functions. More specifically, we posit that the mean component of mean squared error and mean absolute error cause this behavior. We propose two alternative loss functions – Variance-weighted Maximum Squared Error and Variance Weighted Absolute Error – which, crucially, do not incorporate averaging and output variance in the error calculation. We do so to prevent our transformer from converging at a minimum wherein it reduces loss by merely forecasting a time series devoid of volatility.

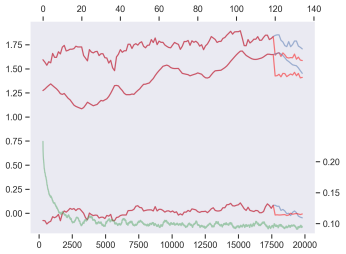
1 Introduction

In recent years, there has been a surge in efforts to adapt transformers for a wide array of tasks [7]. Landmark models such as the Informer, Autoformer, ETSFormer, and FEDFormer have significantly advanced the field by improving forecast length and accuracy [11, 9, 8, 12]. Despite notable progress, the application of transformers to time series forecasting remains largely limited, with only a handful of successful implementations such as Google's MetNet-3 [1]. Questions persist regarding their overall effectiveness in time series forecasting [10].

Motivated by these concerns, we comprehensively explored and evaluated various transformer-based models. Our investigation revealed a common issue across all models: a tendency towards underfitting, resulting in predominantly highly biased, "linear" predictions regardless of the transformer architecture employed. Fig.(1) shows this underfitting.



(a) MSE



(b) MAE

Figure 1: The model's predictions (red) exhibit insufficient volatility compared to the actual values (blue). The green subplots display MSE and MAE on the right and left respectively.

Such behavior is detrimental to transformers in any sort of application wherein a certain volatility is expected of the forecasted sequence. Admittedly, the prevalence of this underfitting across models perplexed us for some time. Eventually, we concluded that the existence of a minimum that punished deviation from the actual sequence's mean was, naturally, owed to the mean component of mean squared error and mean absolute error.

Armed with this information, we then took a step back to examine MSE and MAE. We noticed two things: firstly, metrics that optimize for a mean lead to underfitting; secondly, these metrics consider each time step to be equally important when in reality, the final value of a time series is often most crucial.

Our proposed loss functions seek to mitigate the observed bias by optimizing for the shape of our time series instead. Hereafter, we'll explore various alternatives to averaging the residuals, and we'll also explore weighting and other heuristics.

2 Related Works

2.1 Loss Functions

As it stands, the prevailing issue amongst MSE and other such L_p norm variants is such that they're insensitive to the shape of data. By nature of mean being their key component, they will, of course, aim to reduce the mean error. This results in the characteristically flat prediction shape when MSE or MAE are applied to time series forecasting. In 2024, Lee et al. attempted to mitigate this issue with TILDE-Q, but, whilst innovative, achieves results only slightly better than their benchmark [5]; although, it's an excellent paper if you're interested in the subject, we highly recommend giving it a thorough read.

Computer vision has also seen some innovative loss functions which optimize for depth estimation and generative image synthesis [2].

2.2 Transformer Architectures

Time series analysis has seen many new model architectures in recent years [7]. Many architectures optimize for context length [11, 9, 4, 6, 3], and others optimize for computational efficiency [3]. Despite their differences, they share in their loss functions, as such models will very seldom utilize anything but MSE or MAE.

3 Methodology

3.1 Maximum Error

$$\begin{aligned}\text{MaxAE} &= \max\left(|\mathbf{y} - \hat{\mathbf{y}}|\right) \\ &= \max\left(\forall i \in ||\mathbf{y}||\ |\mathbf{y}_i - \hat{\mathbf{y}}_i|\right)\end{aligned}$$

$$\begin{aligned}\text{MaxSE} &= \max\left((\mathbf{y} - \hat{\mathbf{y}})^2\right) \\ &= \max\left(\forall i \in ||\mathbf{y}||\ \ (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2\right)\end{aligned}$$

3.2 Variance-weighted Maximum Error

We incorporate the absolute difference in variances to ensure that $\hat{\mathbf{y}}$ has a similar variance to that of $\mathbf{y}$, ensuring they have the same shape. We raise e to the power of our maximum differences for two reasons: firstly, to raise our differences above 1; and secondly, to penalize larger differences in variance. Larger differences in variance can be thought of as larger differences in shape.

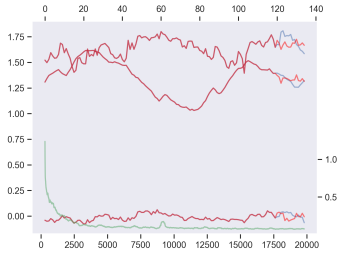
$$\begin{aligned}\text{VMaxAE} &= \exp\left(|\text{var}(\mathbf{y}) - \text{var}(\hat{\mathbf{y}})|\right) \cdot \max\left(|\mathbf{y} - \hat{\mathbf{y}}|\right) \\ &= \exp\left(|\text{var}(\mathbf{y}) - \text{var}(\hat{\mathbf{y}})|\right) \cdot \max\left(\forall i \in ||\mathbf{y}||\ |\mathbf{y}_i - \hat{\mathbf{y}}_i|\right)\end{aligned}$$

$$\begin{aligned}\text{VMaxSE} &= \exp\left(|\text{var}(\mathbf{y}) - \text{var}(\hat{\mathbf{y}})|\right) \cdot \max\left((\mathbf{y} - \hat{\mathbf{y}})^2\right) \\ &= \exp\left(|\text{var}(\mathbf{y}) - \text{var}(\hat{\mathbf{y}})|\right) \cdot \max\left(\forall i \in ||\mathbf{y}||\ \ (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2\right)\end{aligned}$$

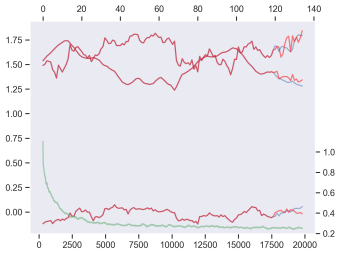
4 Results

5 Conclusion

The code for the transformer model used in this project can be found at cldrake01/sibyl.



(a) VMaxSE



(b) VMaxAE

Figure 2: The model's predictions (red) exhibit insufficient volatility compared to the actual values (blue). The green subplots display MSE and MAE on the right and left respectively.

6 Acknowledgements

We would like to thank our teachers who helped us with this project, specifically Mr. Robert Hettmansperger, Mr. Jake Lehr, and Mx. Seonjoon-young.

References

[1] Marcin Andrychowicz et al. *Deep Learning for Day Forecasts from Sparse Observations*. 2023. arXiv: 2306.06079 [physics.ao-ph].

[2] Jonathan T. Barron. "A General and Adaptive Robust Loss Function". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[3] William Brandon et al. *Striped Attention: Faster Ring Attention for Causal Transformers*. 2023. arXiv: 2311.09431 [cs.LG].

[4] Cristian Challu et al. *N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting*. 2022. arXiv: 2201.12886 [cs.LG].

[5] Hyunwook Lee et al. *TILDE-Q: A Transformation Invariant Loss Function for Time-Series Forecasting*. 2024. arXiv: 2210.15050 [cs.LG].

[6] Hao Liu, Matei Zaharia, and Pieter Abbeel. *Ring Attention with Blockwise Transformers for Near-Infinite Context*. 2023. arXiv: 2310.01889 [cs.CL].

[7] Qingsong Wen et al. "Transformers in time series: A survey". In: *International Joint Conference on Artificial Intelligence(IJCAI)*. 2023.

[8] Gerald Woo et al. *ETSformer: Exponential Smoothing Transformers for Time-series Forecasting*. 2022. arXiv: 2202.01381 [cs.LG].

[9] Haixu Wu et al. *Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting*. 2022. arXiv: 2106.13008 [cs.LG].

[10] Ailing Zeng et al. "Are Transformers Effective for Time Series Forecasting?" In: *Proceedings of the AAAI Conference on Artificial Intelligence* 37.9 (June 2023), pp. 11121–11128. DOI: 10.1609/aaai.v37i9.26317. URL: https://ojs.aaai.org/index.php/AAAI/article/view/26317.

[11] Haoyi Zhou et al. *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*. 2021. arXiv: 2012.07436 [cs.LG].

[12] Tian Zhou et al. "FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting". In: 2022.