Variance-Aware Loss: Addressing Underfitting in Transformer-Based Time Series Forecasting

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

[1]Peak to Peak Charter School

March 1st, 2024

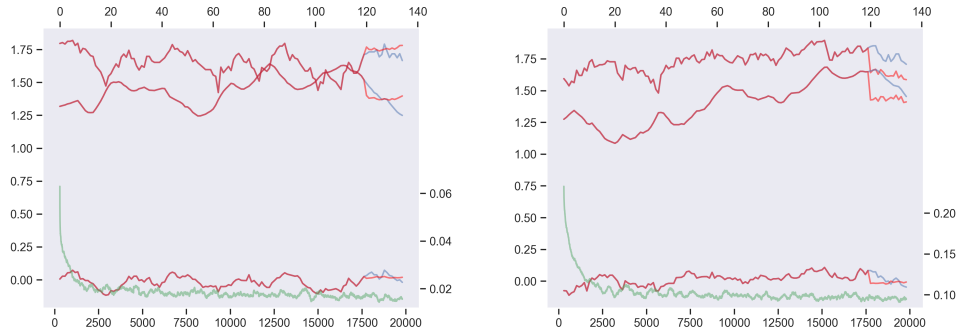
Abstract

When forecasting time series, transformer models predict sequences lacking in volatility. 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 causes this behavior. We propose two alternative loss functions: Variance-weighted Maximum Squared Error and Variance-weighted Maximum Absolute Error, which, crucially, do not incorporate averaging. 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, helping time series transformer models continue to train without the risk of underfitting towards the mean. PyTorch implementations of the models used in this project can be found at github.com/cldrake01/sibyl.

1 Introduction

In recent years, there has been a surge in efforts to adapt transformers for a wide array of tasks [17]. Landmark models such as the Informer, Autoformer, ETSFormer, and FEDFormer have significantly advanced the field by improving forecast length and accuracy [21, 19, 18, 22]. 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 [20].

Motivated by these concerns, we comprehensively explored and evaluated various transformer-based models. Our investigation revealed a common issue across all models: a tendency



(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. In other words, the model is severely underfitted. From top to bottom, the indicators measured are RSI, ADX, and ROC.

towards underfitting, resulting in predominantly "linear" predictions regardless of the transformer architecture employed. Fig.(1) shows this underfitting.

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 Mean Squared Error (MSE) and Mean Absolute Error (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 "linearization" by optimizing for the variance, or "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 Work

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 invariant, or flat in "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, this method achieves results only marginally better than their benchmark [10].

Computer vision has also seen some innovative loss functions which optimize for depth

estimation and generative image synthesis [2].

2.2 Transformer Architectures

Researchers and practitioners have long been interested in autoregressive models for their ability to forecast a wide array of trends, the applications of which include weather forecasting, demand forecasting, and quantitative analysis [6, 1, 8, 7]. Autoregressive models, including ARIMA, RNNs, LSTMs, as well as N-BEATS [13] and N-HITS [5] more recently, have been extensively used in forecasting tasks but often struggle with capturing long-term dependencies, especially in seemingly stochastic time series data [15, 11, 9, 16, 4, 14]. Transformers, however, are known for their wildly successful application to Natural Language Processing (NLP) tasks, wherein they are required to attend to the relationships of many interdependent tokens. It was for their ability to capture long-term dependencies and complex relationships that researchers began applying transformers to time series forecasting tasks. Consequently, researchers have seen many new transformer architectures in recent years, with many of them having been conceived with time series forecasting in mind [21, 19, 12, 3]. While transformers have come a long way, each architecture that we tested exhibited significant underfitting when tasked with forecasting stochastic, financial data.

3 Methodology

3.1 Existing Loss Functions

It's worth noting that actual values, $\mathbf{y}$, and the predicted values, $\hat{\mathbf{y}}$, are always assumed to be identical in their lengths, e.g., $n = \dim(\mathbf{y}) = \dim(\hat{\mathbf{y}})$. For regression, the most widely used loss functions are

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \hat{\mathbf{y}}_i)^2,$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\mathbf{y}_i - \hat{\mathbf{y}}_i|.$$

3.2 Maximum Error

Maximum Error functions are practically the same function as Mean Error Functions (MAE, MSE), with the only difference being that they take the maximum difference within the residuals as opposed to a mean difference. Furthermore, both means and maximums are permutation invariant. We use this function to create more of a "moving target" for our model to target, as opposed to solely minimizing a mean of the residuals.

$$\begin{aligned}\text{MaxSE} &= \max\left(\mathbf{y}-\hat{\mathbf{y}}\right)^2 \\ &= \max_i\ \left(\mathbf{y}_i-\hat{\mathbf{y}}_i\right)^2,\end{aligned}$$

$$\begin{aligned}\text{MaxAE} &= \max\left|\mathbf{y}-\hat{\mathbf{y}}\right| \\ &= \max_i\left|\mathbf{y}_i-\hat{\mathbf{y}}_i\right|.\end{aligned}$$

3.3 Variance-weighted Maximum Error

We incorporate the absolute difference in variances to ensure that $\hat{\mathbf{y}}$ has a similar variance to that of $\mathbf{y}$. 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. Differences in variance can be thought of as differences in "shape".

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

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

4 Results

Across all metrics, the VMaxSE and VMaxAE performed very similarly to their counterparts.

Table 1: Performance Metrics for All Loss Functions on Alpaca

Metric	VMaxSE	MSE	VMaxAE	MAE
Bias	0.56074	0.55505	0.55493	0.55569
Variance	0.55281	0.55258	0.5513	0.55768
Std	0.74342	0.74318	0.74204	0.74665
MSE	0.01556	0.01545	0.01548	0.01567
MAE	0.08986	0.08917	0.08869	0.09039

Table 2: Performance Metrics for All Loss Functions on ELD

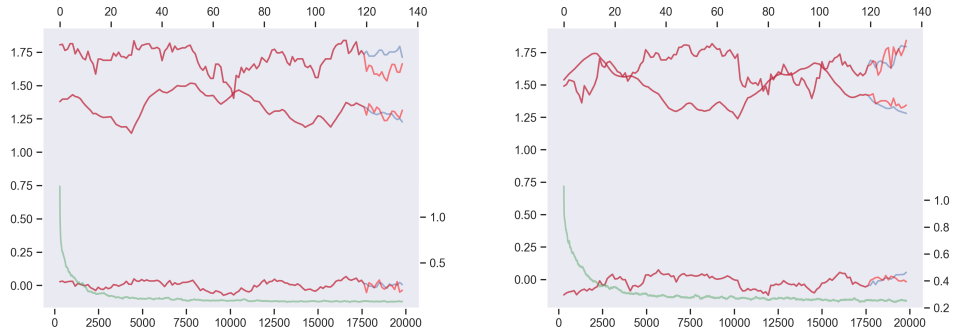
Metric	VMaxSE	MSE	VMaxAE	MAE
Bias	0.32125	0.31772	0.32909	0.32533
Variance	0.10593	0.1032	0.21569	0.22624
Std	0.32499	0.32078	0.46438	0.47549
MSE	0.22077	0.21572	0.24866	0.25734
MAE	0.33776	0.34332	0.29828	0.3058

Table 3: Performance Metrics for All Loss Functions on ETT

Metric	VMaxSE	MSE	VMaxAE	MAE
Bias	0.2996	0.27243	0.28104	0.28554
Variance	0.08054	0.07948	0.90563	0.09168
Std	0.326	0.28137	0.32442	0.30242
MSE	0.1961	0.19511	0.2286	0.22082
MAE	0.29399	0.29715	0.26623	0.26367

5 Conclusion

All in all, our study addressed the issue of underfitting in transformer-based time series forecasting models, attributing it to the mean component in traditional loss functions like MSE and MAE. To counter this, we introduced two variance-weighted loss functions, VMaxSE and VMaxAE, which prioritize capturing the "shape" of the time series over averaging. This underscores the necessity of rethinking loss functions in time series forecasting, especially in transformer models, for more accurate predictions. We anticipate further exploration of innovative loss functions and architectural improvements to address underfitting by considering the temporal characteristics of data.



(a) VMaxSE (b) VMaxAE

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

6 Acknowledgements

We would like to thank and acknowledge Mr. Robert Hettmansperger and Mr. Jake Lehr.

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] Jian Cao, Zhi Li, and Jian Li. "Financial time series forecasting model based on CEEMDAN and LSTM". In: *Physica A: Statistical Mechanics and its Applications* 519 (2019), pp. 127–139. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2018.11.061. URL: https://www.sciencedirect.com/science/article/pii/S0378437118314985.

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

[6] Jamal Fattah et al. "Forecasting of demand using ARIMA model". In: *International Journal of Engineering Business Management* 10 (2018), p. 1847979018808673. DOI: 10.1177/1847979018808673. eprint: https://doi.org/10.1177/1847979018808673. URL: https://doi.org/10.1177/1847979018808673.

[7] Tingyu Guo and Boping Tian. "The Study of Option Pricing Problems based on Transformer Model". In: *2022 International Conference on Information Science and Communications Technologies (ICISCT)*. 2022, pp. 1–5. DOI: 10.1109/ICISCT55600.2022.10146913.

[8] S.L. Ho and M. Xie. "The use of ARIMA models for reliability forecasting and analysis". In: *Computers Industrial Engineering* 35.1 (1998), pp. 213–216. ISSN: 0360-8352. DOI: https://doi.org/10.1016/S0360-8352(98)00066-7. URL: https://www.sciencedirect.com/science/article/pii/S0360835298000667.

[9] Zahra Karevan and Johan A.K. Suykens. "Transductive LSTM for time-series prediction: An application to weather forecasting". In: *Neural Networks* 125 (2020), pp. 1–9. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2019.12.030. URL: https://www.sciencedirect.com/science/article/pii/S0893608020300010.

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

[11] Benjamin Lindemann et al. "A survey on long short-term memory networks for time series prediction". In: *Procedia CIRP* 99 (2021). 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020, pp. 650–655. ISSN: 2212-8271. DOI: https://doi.org/10.1016/j.procir.2021.03.088. URL: https://www.sciencedirect.com/science/article/pii/S2212827121003796.

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

[13] Boris N. Oreshkin et al. "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting". In: *CoRR* abs/1905.10437 (2019). arXiv: 1905.10437. URL: http://arxiv.org/abs/1905.10437.

[14] Alaa Sagheer and Mostafa Kotb. "Time series forecasting of petroleum production using deep LSTM recurrent networks". In: *Neurocomputing* 323 (2019), pp. 203–213. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.09.082. URL: https://www.sciencedirect.com/science/article/pii/S0925231218311639.

[15] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. "The Performance of LSTM and BiLSTM in Forecasting Time Series". In: *2019 IEEE International Conference on Big Data (Big Data)*. 2019, pp. 3285–3292. DOI: 10.1109/BigData47090.2019.9005997.

[16] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. "A Comparison of ARIMA and LSTM in Forecasting Time Series". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018, pp. 1394–1401. DOI: 10.1109/ICMLA.2018.00227.

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

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

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

[20] 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.

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

[22] Tian Zhou et al. *FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting*. 2022. arXiv: 2201.12740 [cs.LG].