

A first exploration of the new minGRU models for time series analysis

Václav Tran

ČVUT - FIT

tranvacl@fit.cvut.cz

December 27, 2024

1 Introduction

This work focuses on three time-series related tasks. The first task involves parameter inference, where the goal is to determine key characteristics of a sinusoidal wave, such as amplitude and frequency, directly from a set of observed data points. The second task is time-series forecasting, where the objective is to predict the future evolution of a sinusoidal wave based on historical data. The third task extends the application of time-series forecasting to stock market data.

To address these tasks, we evaluated the performance of the minGRU architecture [2], a simplification of GRU architecture [1]. We compare minGRU with other established architectures, assessing their strengths and limitations in these two tasks. Additionally, in the second task, training time versus dataset size was evaluated for minGRU, GRU, and LSTM [3] to compare their computational efficiency.

2 Input Data

The datasets used in this exploration are tailored to the specific requirements of each task. Each dataset was divided into an 80/20 split for training and testing across all tasks.

2.1 Parameter Inference

For the parameter inference task, a custom synthetic dataset was generated, where each sample consists of 100 data points representing a sinusoidal wave with varying amplitude and frequency.

2.2 Time-Series Forecasting (Sinusoidal Waves)

For the time-series forecasting of sinusoidal waves, another custom synthetic dataset was generated. Each sample in this dataset represents a sinusoidal wave, consisting of 100 data points as input and one additional target data point for prediction.

2.3 Stock Market Forecasting

For the stock market forecasting task, real-world stock data for IBM was sourced using the `yfinance` library. The dataset spans from 2014-01-01 to 2024-04-01 and includes daily stock prices. Pre-processing focused on the closing prices.

3 Methods

The methods employed in this study are tailored to the requirements of each task. Different evaluation metrics and model comparisons are used to ensure a quality assessment. All computations were performed on NVIDIA A100 GPU, 20GB MIG instance using MetaCentrum's GPU clusters.

3.1 Parameter Inference

For the parameter inference task, the models are evaluated using the Mean Absolute Error (MAE). The performance of the following models is compared:

- GRU
- LSTM
- Transformer-encoder [5]
- minGRU

The GRU, LSTM, and Transformer-encoder models were implemented using the PyTorch `nn` library, while the minGRU architecture was implemented from scratch based on its original paper.

3.2 Time-Series Forecasting (Sinusoidal Waves)

For time-series forecasting of sinusoidal waves, models are evaluated using the normalized Root Mean Squared Error (nRMSE). The comparison includes the following models:

- minGRU
- Transformer encoder-decoder

- N-BEATS [4]
- Exponential Smoothing

All models except minGRU are implemented and evaluated using the DARTS time-series forecasting library. This task focuses on comparing the ability of the models to predict future values of sinusoidal waves based on historical data. For minGRU, autoregressive prediction was used, where each model used its previous predictions as inputs to forecast future values. During training, a sliding window approach was employed, where the model used the previous 100 data points to predict the next data point.

During evaluation of training time versus dataset size in time-series forecasting of sinusoidal waves, a different implementation of LSTM and GRU was used to account for the high optimization of PyTorch’s default implementation [6].

3.3 Stock Market Forecasting

For forecasting stock market data, the models are also evaluated using nRMSE. The following models are compared:

- minGRU
- Transformer encoder-decoder
- N-BEATS
- Exponential Smoothing
- Prophet

As in Task 2, the DARTS library is used for implementing and evaluating most models, with Prophet included as an additional comparison. Once again, for minGRU, autoregressive prediction was used. During training, a sliding window approach was employed, where the model used the previous 60 time steps to predict the subsequent 60 time steps.

4 Results

4.1 Parameter Inference

For the parameter inference of sinusoidal waves, models were evaluated using the MAE for amplitude and frequency estimation, along with percentage errors relative to the true values. The results calculated on 10000 samples are presented below. The percentage error indicates how large the error is compared to the actual value.

The GRU model achieved the lowest error rates for both amplitude and frequency estimation. MinGRU and LSTM show comparable performance,

Table 1: Performance of models on amplitude estimation.

Model	MAE (units)	Percentage (%)
MinGRU	0.13	1.32
GRU	0.12	1.17
LSTM	0.15	1.51
Transformer Encoder	0.22	2.24

Table 2: Performance of models on frequency estimation.

Model	MAE (units)	Percentage (%)
MinGRU	0.15	1.46
GRU	0.05	0.48
LSTM	0.12	1.20
Transformer Encoder	0.21	2.09

while the Transformer had the highest errors in both categories.

4.2 Time-Series Forecasting (Sinusoidal Waves)

The dataset used for this task consisted of 5000 samples, where each sample had 100 data points plus one target point. The maximum amplitude and frequency were both set to 10. The models were trained on 4000 samples for 100 epochs and their performance was evaluated on test set, comprised of the remaining 1000 samples. The averaged results of the 1000 samples are presented in Table 3.

Table 3: Performance of models on sinusoidal wave forecasting (nRMSE).

Model	Average nRMSE
minGRU	1.2603
N-BEATS	1.7368
Transformer	1.9388
Exponential Smoothing	17.0372

4.2.1 Training Time and Dataset Size

The evaluation of time-series forecasting models for sinusoidal waves includes a comparison of training times across different dataset sizes.

As shown in Figure 1, the minGRU consistently demonstrates faster training times compared to the LSTM and GRU, particularly as the dataset size increases. This efficiency makes minGRU a suitable choice for large-scale time-series datasets.

4.3 Stock Market Forecasting

For stock market data forecasting, the models were evaluated using nRMSE. The results on the test set for IBM stock data are shown in Figure 2.

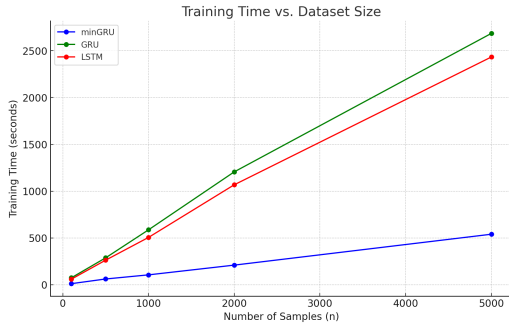


Figure 1: Training time vs. dataset size for minGRU, GRU, and LSTM

4.4 Discussion

The results demonstrate that the minGRU architecture effectively balances forecasting accuracy and computational efficiency, outperforming or matching traditional models such as GRU and LSTM in sinusoidal forecasting tasks while offering faster training times.

However, GRU surprisingly outperforms minGRU by a somewhat large margin in estimating the frequency of the sinusoidal wave. This discrepancy may be attributed to GRU’s more complex gating mechanisms, which could be required for more accurate frequency estimation in sinusoidal data.

In the stock market forecasting task, minGRU achieved competitive performance compared to established models, although specialized models like Prophet still exhibited superior accuracy.

Overall, these findings suggest that minGRU is a viable and efficient alternative for time-series analysis, particularly in scenarios where computational resources and training speed are critical.

5 Conclusion

The minGRU architecture offers a significantly more efficient and scalable alternative to traditional GRU models, demonstrating competitive performance in time-series forecasting tasks.

This analysis supports the adoption of minGRU in various time-series forecasting scenarios, where it consistently matches or outperforms other architectures, such as Transformer and N-BEATS, especially in environments with limited computing resources.

Further analysis could explore the application of minGRU in diverse domains such as weather prediction and anomaly detection, as well as evaluate its scalability and performance in natural language processing tasks to determine whether it can compete effectively with Transformer-based architectures.

References

- [1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.
- [2] Leo Feng, Frederick Tung, Mohamed Osama Ahmed, Yoshua Bengio, and Hossein Hajimirsadeghi. Were rnns all we needed?, 2024.
- [3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
- [4] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting, 2020.
- [5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.
- [6] George Yiasemis. Recurrent neural networks from scratch using pytorch. <https://github.com/georgeyasemis/Recurrent-Neural-Networks-from-scratch-using-PyTorch>, 2023.



Figure 2: Stock market prediction visualizations.