Time Series Prediction Using Neural Networks

Yukai Song

December 18, 2024

Introduction and Data Preparation

This report examines four neural network architectures—Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM), Gated Recurrent Units (GRU), and Transformers—for predicting two time series signals, Ot and Rt, sampled every 15 minutes over 11 months. The goal is to develop, train, and evaluate these models, comparing their performance to identify the best predictors based on Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2.

Data Preparation:

1. **Splitting:** 80% for training and 20% for testing.
2. **Sequence Creation:** Generated input sequences using a sliding window to provide temporal context.
3. **Scaling:** Normalized features using the MinMaxScaler provided by the scikit-learn library to enhance model convergence.

Model Implementation and Training

We implemented and trained four models using TensorFlow:

- **RNN:** Baseline model with simple recurrent connections.
- **LSTM:** Captures long-term dependencies via gating mechanisms.
- **GRU:** Simplified alternative to LSTM, balancing complexity and efficiency.
- **Transformers:** Utilize self-attention to capture global temporal relationships.

Hyperparameter Choices and Tuning

For the RNN, LSTM, and GRU models, the number of units was set to **300**, the random seed to **0**, the learning rate to 1×10^{-3}, the dropout rate to **0.3**, and the number of layers to **2**. In contrast, the Transformer model was configured with **4 attention heads**, a model dimensionality (d_{model}) of **128**, **3072** feed-forward units (ff_units), a learning rate of 1×10^{-3}, a dropout rate of **0.3**, and **2 layers**. The sequence length was set to **32**, the total number of epochs was set to **100**, and the early stopping strategy was implemented by monitoring the validation loss for **10** epochs.

After training these models for the Ot and Rt signals, it was found that the training loss curve drops quickly in the beginning and then decreases slowly, while the validation loss curve oscillates as the number of epochs increases (these curves can be found in the attached notebook). Based on this observation, this might be because the learning rate 1×10^{-3} is too large, preventing the models from learning the trends of the Rt and Ot signals effectively. Therefore, the learning rate was set to 1×10^{-4}, 5×10^{-5}, or 1×10^{-5} during hyperparameter tuning for all models. Additionally, the number of units was set to **250** or **350**, and the number of layers to **2** or **3** to observe their influences in RNN, LSTM, and GRU during tuning. For tuning the Transformer, the number of heads was set to **2** or **4**, the model dimensionality (d_{model}) to **128** or **256**, the number of feed-forward units to **2048** or **3072**, and the number of layers to **2** or **3**. Other hyperparameter choices remained the same as described in the previous paragraph. The results of training and hyperparameter tuning can be found in the next section.

Comparison and Discussions

The performance on the testing set for the Ot and Rt signals of the original models before hyperparameter tuning is summarized in the table 1.

1. **Ot Signal:**
   - **MSE:** The Transformer (0.0426) slightly outperforms the GRU (0.0433).
   - **MAE:** The GRU achieves a marginally lower MAE (0.1366) than the Transformer (0.1495).
   - **R^2:** The Transformer has a higher R^2 (0.3711) compared to the GRU (0.3599).
2. **Rt Signal:**
   - **MSE:** The Transformer leads with the lowest MSE (0.0079).
   - **MAE:** The Transformer also has the best MAE (0.0521).
   - **R^2:** The Transformer explains 62.93% of the variance, surpassing the GRU (61.79%).

Model	Target	MSE	MAE	R^2
RNN	Ot	0.0433	0.1434	0.3599
	Rt	0.0089	0.0579	0.5829
LSTM	Ot	0.0442	0.1498	0.3464
	Rt	0.0134	0.0713	0.3699
GRU	Ot	0.0433	0.1366	0.3599
	Rt	0.0081	0.0565	0.6179
Transformer	Ot	0.0426	0.1495	0.3711
	Rt	0.0079	0.0521	0.6293

Table 1: Evaluation Metrics for Ot and Rt Predictions on the testing set

3. **Overall:**
   - The GRU and Transformer models show strong performance across both signals.
   - The Transformer excels in capturing global dependencies, especially for Rt.
   - The LSTM underperforms relative to the GRU and Transformer.

In the hyperparameter fine-tuning stage, the best model was selected based on the Mean Squared Error (MSE). It was found that the best model for the Ot signal is an LSTM with the number of units set to **250**, a learning rate of 1×10^{-4}, a dropout rate of **0.3**, and **3 layers**, resulting in an MSE of **0.0422**. Similarly, the best model for the Rt signal is an LSTM with the number of units set to **250**, a learning rate of 5×10^{-5}, a dropout rate of **0.3**, and **3 layers**, resulting in an MSE of **0.0076**.

- **Recurrent Neural Network (RNN)**
  - **Strengths:**
    - **Simplicity:** RNNs have a straightforward architecture, making them easy to implement and understand.
  - **Weaknesses:**
    - **Vanishing Gradient Problem:** Struggles with long-term dependencies.
- **Long Short-Term Memory (LSTM)**
  - **Strengths:**
    - **Handling Long-Term Dependencies:** LSTMs are designed to capture long-range dependencies in data, mitigating the vanishing gradient issue inherent in standard RNNs.
    - **Best Performance:** LSTMs achieved the best MSE in the hyperparameter fine-tuning stage.
  - **Weaknesses:**
    - **Complexity:** More computationally intensive with more parameters.
- **Gated Recurrent Unit (GRU)**
  - **Strengths:**
    - **Efficiency:** Simpler than LSTMs with fewer parameters, leading to faster training and reduced computational resource requirements.
  - **Weaknesses:**
    - **Slight Inferiority to Transformers:** While robust, GRUs do not outperform Transformers, particularly in tasks where attention mechanisms provide significant advantages.
- **Transformer**
  - **Strengths:**
    - **Attention Mechanism:** Effectively captures complex and long-range dependencies through self-attention, enhancing predictive accuracy and explanatory power.
  - **Weaknesses:**
    - **Computational Resources:** Transformers typically require more computational power and memory, which can be a limitation in resource-constrained environments.
    - **Data Requirements:** They perform best with large datasets, and their effectiveness may diminish with smaller datasets. During the hyperparameter fine-tuning stage, the Transformer performed slightly worse than the LSTM, with a best MSE of **0.0077** for the Rt signal and **0.0424** for the Ot signal (compared to **0.0076** and **0.0422** for the LSTM, respectively).

Conclusion

- The **Transformer** model ranks second for its superior performance, leveraging attention mechanisms to handle complex dependencies effectively.
- The **LSTM** offers a balanced alternative with strong performance and efficiency, resulting in the best MSE for both Ot and Rt signals.
- **RNNs** and **GRUs**, while foundational and capable of handling sequential data, lag behind in this comparison due to performance limitations and higher complexity, respectively.