LUT School of Engineering Sciences
Computer Vision and Pattern Recognition
BM20A6100 - Advanced Data Analysis and Machine Learning
Professor – & Teaching Assistant: Lasse Lensu and Akseli Suutari

Forecasting the electric power consumption for a house

Week 5 - Next Day Forecasting Model

Sunday 30th November, 2025
Umme Tanjuma Haque
Chamath Wijerathne
Nada Rahali

Contents

List of Figures

1 Introduction

This week's assignment is focused on developing a multivariate next-day forecasting model utilizing a Transformer neural network. The idea is to adapt the Transformer architecture to daily household energy data and assess its forecasting performance. Multivariate forecasting requires capturing interactions between multiple electrical measurements and understanding how they collectively affect the target variable. Hence, this task is both more informative and more complex.

The dataset, as discussed in previous tasks, is the household electric power consumption dataset, originally provided at one-minute resolution. This task's work involved preprocessing, daily resampling, scaling, sequence creation, model training, evaluation of the test dataset, and interpretation of the results.

2 Data Preparation

2.1 Loading and Preprocessing

The loading and preprocessing of the data are consistent with the previous task. After preprocessing, the dataset contained no missing values and included all required energy-related features.

2.2 Daily Resampling and Feature Matrix Construction

The minute-level data was resampled into daily averages to provide a smoother, lower-noise sequence suitable for Transformer forecasting.

The multivariate feature matrix includes:

- Global_active_power
- Global_reactive_power
- Voltage
- Global_intensity
- Sub_metering_1
- Sub_metering_2
- Sub_metering_3
- y (target)

In total, the daily dataset contained **1,442 days × 8 features**.

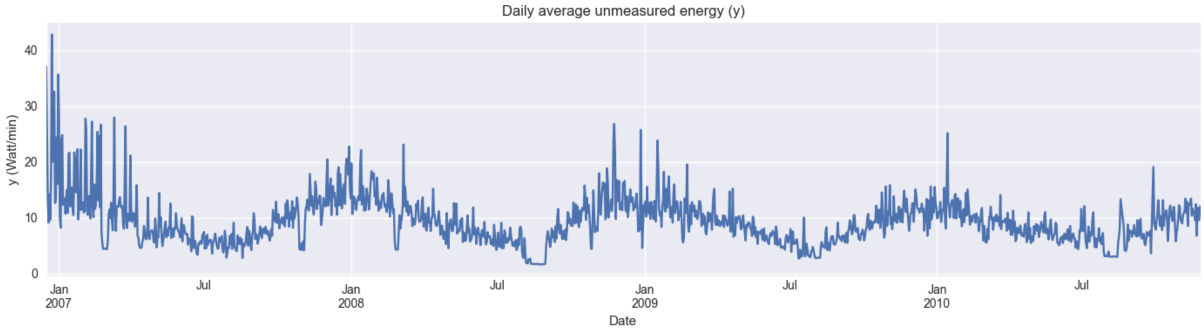

Figure 1: Daily average after resampling

2.3 Train Test Split and Scaling

An 80/20 time-based split was applied to preserve temporal order, with the first 80% of days used for training and the last 20% used for testing.

A MinMaxScaler was fit on the training set only to avoid data leakage, and then applied to both train and test sets.

2.4 Sequence Generation

The training of the Transformer model was done through the creation of fixed-sized input sequences, where a windowing method was applied such that each example was provided 30 days of multivariate data (30 × 8 features) along with the corresponding value of variable y for the next day. This led to a total of **1,123 training instances** and **259 test instances**. The training data was then further split into **898 instances for training** and **225 instances for validation**. Finally, all these instances were encapsulated within PyTorch Dataset and DataLoader objects.

3 Transformer Model Architecture

A custom PyTorch Transformer architecture was implemented.

Input projection layer: Maps the 8-dimensional feature vector into a 64-dimensional embedding.

Transformer Encoder:

2 encoder layers, 4 attention heads, feedforward dimension = 128, dropout = 0.1, and batch_first=True.

This allows the model to learn temporal relationships across the 30-day window.

Sequence Aggregation:

The representation of the final time step is extracted:

$$z = x[:, -1, :]$$

Output layer:

A final linear layer maps the encoded vector to a single regression value (forecast).

Training Setup:

Loss: MSE

Optimizer: Adam (learning rate = 1e-3)

Epochs: 30

Batch size: 32

4 Results

4.1 Training and Validation Loss

After 30 epochs of training, the training loss and validation loss converged quickly during the first five to ten epochs; after that, they gradually stabilized. It was also observed that the validation loss remained approximately at par with, or even lower than, the training loss, without any observed signs of overfitting. This means that the Transformer model was able to learn the temporal relationships effectively, that the selected hyperparameters were appropriate, and that the regularization techniques used, such as dropout, effectively prevented overfitting.

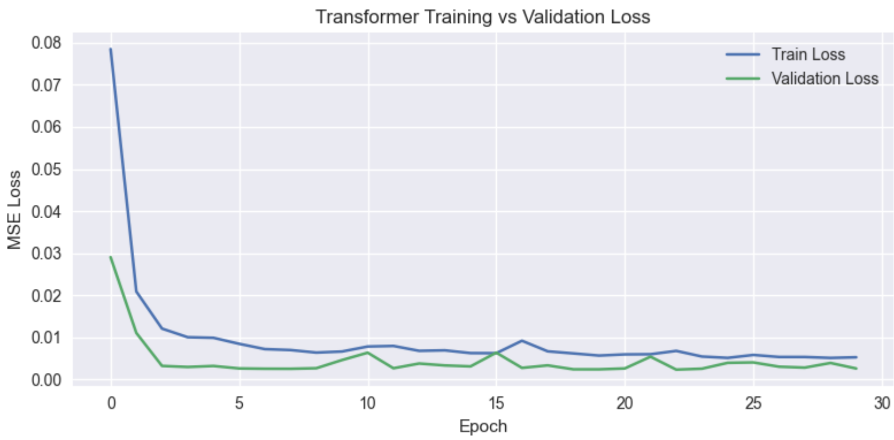

Figure 2: Training vs Validation Loss

4.2 Test Set Metrics

After training, the model was tested for its accuracy on a test set of data that was unseen, and the predictions made by the model were converted back to the original units of the target variable. The RMSE and MAE of the model for this test data were found to be 1.883 and 1.368, respectively. Both these errors are much lower than the variability of the target variable.

4.3 Forecast Visualization

By comparing the actual and predicted series for the first 150 test days, the predicted series was able to track the trend and seasonality of the actual target series. The points of peak and trough were predicted relatively well, although sudden peaks or troughs had a tendency to be slightly smoothed, as is to be expected for sequence-based models. Also, there was no sign of biased or drifting patterns. The Transformer was able to generalize well from the training to the test data.

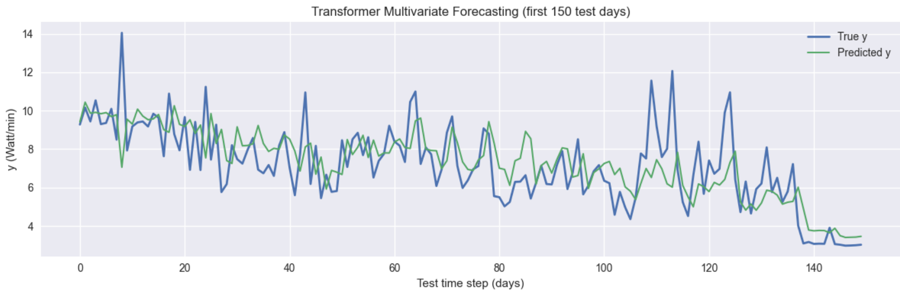

Figure 3: Forcasting First 150 test days

5 Complexity Considerations

Handling multivariate time series adds to the complexity. It is necessary to learn inter-variable relationships for eight different variables, along with their combined effect on the output variable, while the attention mechanism must take into consideration the relationships from the perspectives of both time and feature dimensions. Moreover, it was important to pay attention to the preprocessing of the training data, especially because the output variable is a derived value.

6 Optimization and Tuning Plan

Based on this training and validation process, various techniques can be applied to optimize this Transformer model. These techniques include testing the size of the window for input, such as a 14-day window, a 45-day window, or even a 60-day window, to determine if a shorter or longer window is optimal.

Another important domain to focus on is hyperparameter tuning. The model capacity can be controlled by the embedding dimension parameter or d_model (ranging from 32 to 128), the number of encoder layers from 1 to 4, the number of attention mechanisms from 2 to 8, and the dimension of the feedforward layer from 128 to 512. The learning rate or schedules such as ReduceLROnPlateau can be tested.

Techniques of regularization can be improved by exploring different values of dropout parameters (from 0.05 to 0.2) or adding weight decay to the optimizer. Validation techniques can be improved by using time series cross-validation techniques, such as a rolling window method, and adding early stopping can also be beneficial.

It might be possible to make further enhancements through feature engineering, for example, incorporating calendar-based features or lag variables. There is also the possibility of forming a hybrid model that includes the Transformer along with other components such as MLP layers. Finally, the model can be developed to incorporate multi-step forecasting, where predictions for multiple days in the future can be obtained simultaneously.

Taken altogether, these tuning methods form a roadmap that can be applied to improve the accuracy, stability, and generalization capacity of the Transformer model for multivariate time series forecasting.

7 Conclusion

This task effectively applied a multivariate Transformer model for forecasting energy for the next day, depending on the characteristics of household energy consumption on a daily basis. The whole procedure–from preprocessing to daily resampling, formation of a sequence, training, and evaluation–was carried out effectively, and accurate forecasts were obtained. Additionally, the fact that the Transformer model was capable of learning long-term dependencies shows that further improvements in its capacity, sequence length, and optimization methods could lead to even better forecasts.