Michael Chase Allen
CSCI 575 – Adv. Machine Learning
5/1/2024

Predicting Energy Consumption with LSTMs

Problem Statement

Predicting energy consumption is vital for effective energy management, enabling energy-saving measures, and enhancing the efficiency of energy distribution systems. It supports the integration of renewable energy sources into the grid, aligning energy supply with demand, and reducing reliance on non-renewable, environmentally harmful sources. Accurate forecasts can lead to cost savings for both energy producers and consumers through optimized production and consumption strategies. Using data from the Individual Household Electric Power Consumption dataset, Machine Learning models like LSTMs can be created to predict the energy consumption over days, hours, or even minutes.

Solution

To predict the energy consumption for future timestamps, the product first checks if an 'outputs' directory exists, and if not, it creates one. This directory is used to store the outputs of the script, such as plots and models.

Then the script uses a DataLoader class to load the data from the specified file. The DataLoader class is responsible for loading the data, cleaning it, and providing it in a format that can be used for analysis and modeling. Provided the intention was to create a generalizable product, this class could be easily adapted to load other time series datasets or alternate versions of the Individual Household Electric Power Consumption dataset.

If the -eda flag is provided, the script performs exploratory data analysis (EDA) on the dataset using a DataFramePlotter class. This class is responsible for creating plots of the data, which can help to understand the patterns and trends in the data. This step is not specific to energy consumption data and could be used with any time series data.

Data preprocessing is done to convert the data into a format that can be ingested by TensorFlow's LSTM architecture. This includes the introduction of lag features, where a user can specify the number of time steps they want to include as lag features. Other noteworthy preprocessing steps include MinMax normalization and resampling to a longer time period (by default, the dataset is in minutes which is too granular for the LSTM to pick up definitive trends).

Finally, a MyLSTM object is instantiated which: takes the preprocessed data frame, splits the data into training and testing sets based on the split ratio, reshapes the input data to be 3D (required input shape for LSTM models in Keras), instantiates a LSTM model with 300 units and a Dense output layer with a single neuron (predicted value), fits the model on the training data, and plots the predictions on the testing data. This class also has support for plotting loss metrics and saving the serialized model for incremental learning.

Overall, the product was designed to be flexible and adaptable. By changing the data loading and preprocessing steps, it could be used to forecast other time series data. The use of command-line arguments allows a user to easily customize the behavior of the script.

Demo

To try out this product, check the Readme and follow the code execution steps. Alternatively, here is a quick live demo showing how the product would work for an end user.

Assumptions, Constraints, and Implications

One limiting assumption made was the dataset column names when parsing/cleaning the dataset, which may preclude the generalizability of this product and requires further exploration for how it could be implemented to extend beyond the current dataset. In this vein, the dataset reshaping steps for prediction and scaling may require further exploration to ensure that new datasets can be transformed (scaled and inverse scaled) and concatenated without changing the code and command line inputs.

Developing a Solution

As mentioned previously, the dataset used was the Individual Household Electric Power Consumption dataset. The primary feature engineering in this product is the introduction of lag features and resampling, which users can control to potentially improve model performance. The model trained was a 300 unit LSTM model with rectified linear unit (relu) activation, adam optimizer, and mean squared error (mse) loss.

Michael Chase Allen
CSCI 575 – Adv. Machine Learning
5/1/2024

Summary

This project demonstrates the use of Long Short-Term Memory (LSTM) networks to predict energy consumption from the Individual Household Electric Power Consumption dataset. The process begins with the creation of an 'outputs' directory for storing script outputs like plots and models. Data is loaded and preprocessed, including normalization and resampling to make it suitable for LSTM analysis. A key feature is the introduction of lag features that help the model understand temporal dependencies. The script, designed with flexibility in mind, can adapt to various datasets through modifications in data loading and preprocessing steps. The trained model, consisting of a 300-unit LSTM with a relu activation function, demonstrates promising results with an RMSE of approximately 0.55 on the test set. Overall, the solution provides a robust framework for energy consumption forecasting, which could be extended to other time series forecasting tasks. Further explorations and adjustments are suggested for enhancing generalizability and performance across different datasets.

Appendix

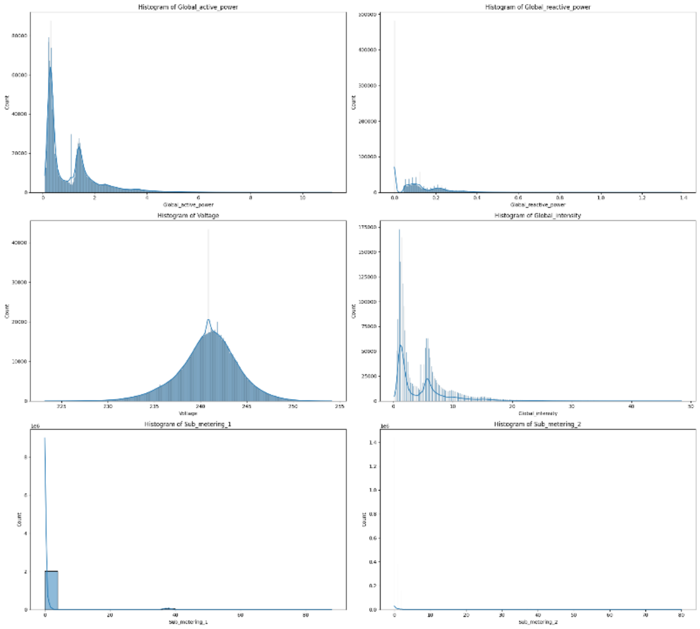

Figure A1: Output of EDA step, including histograms of categorical variables

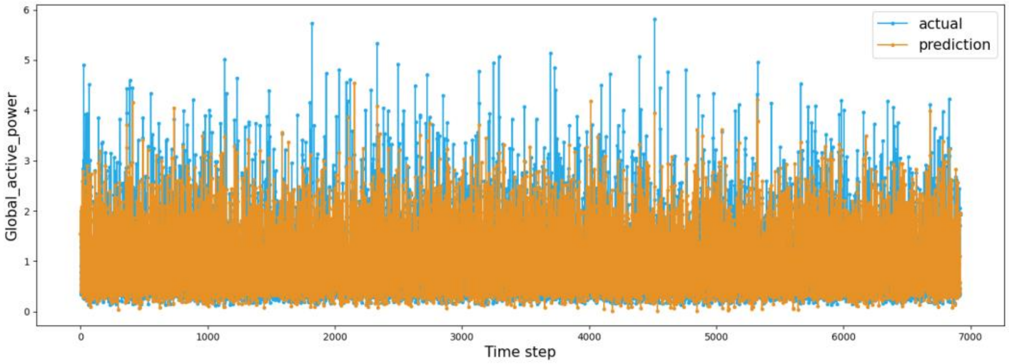

Figure A2: Model predictions on testing set, with an RMSE of ~.553

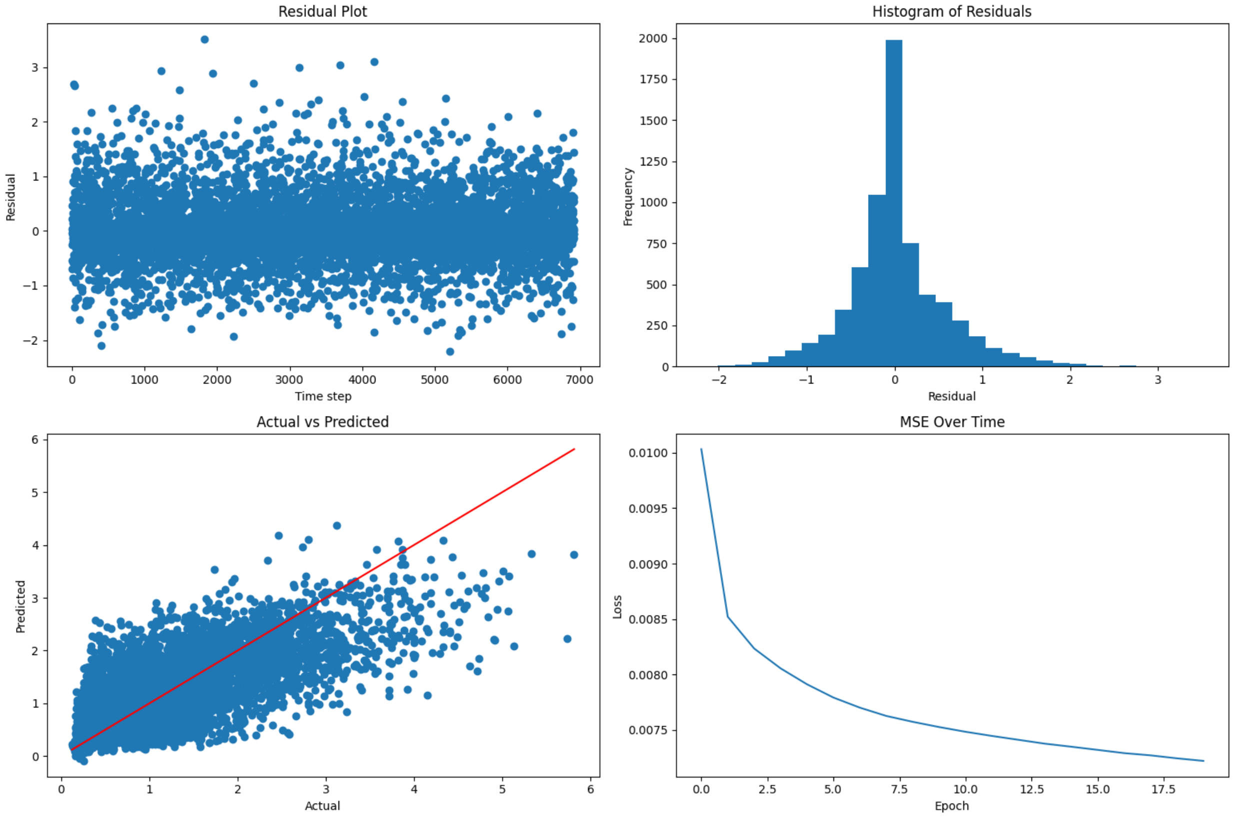

Figure A3: Loss and Residual metrics for LSTM training and output