Recurrent Neural Network Architectures for Time series Forecasting

Kshitij Aggarwal
Department of Robotics
University of Maryland
College Park, United States
kshitij2@umd.edu

Nirvan Mishra
Department of Robotics
University of Maryand
College Park, United States
nmishra1@umd.edu

Chaitanya Kulkarni
Department of Robotics
University of Maryland
College Park, United States
chaitkul@umd.edu

Raajith Gadam
Department of Robotics
University of Maryland
College Park, United States
raajithg@umd.edu

Abstract

To model various suitable Machine Learning models for Time-Series forecasting, and compare their performances to arrive at the best possible architecture for the prediction problem. Here, RNN, LSTM, and 1-D CNN have been implemented and compared to each other. Inline with our expectations, LSTM performs best among the three with RNN being a close second.

Furthermore, the best-performing model of the LSTM is then used to predict the future electricity consumption by an average household over time.

1. Introduction

Long Short-Term Memory (LSTM) is a form of recurrent neural network (RNN) architecture designed to handle the difficulties associated with learning long-term dependencies in sequential input. Sepp Hochreiter and Jürgen Schmidhuber introduced LSTM in 1997, and it has since become an essential tool in the field of deep learning, particularly in applications containing sequential information, such as natural language processing, speech recognition, and time series analysis.

The vanishing gradient problem makes it difficult for traditional RNNs to capture and store information over lengthy sequences. By adding a specific memory cell, gating mechanisms, and a forget gate, LSTM was created to bypass these constraints. These elements allow LSTMs to selectively store, update, and retrieve information, allowing them to sustain a more stable learning process.

An input, output, and forget gate are included in every memory cell that makes up the unique LSTM architecture. These gates regulate the information flow, making it easier for the network to store and remove data as needed. LSTMs are well-suited for jobs requiring an extensive hold of context and temporal patterns because of their complicated design, which enables them to model intricate connections within sequential data.

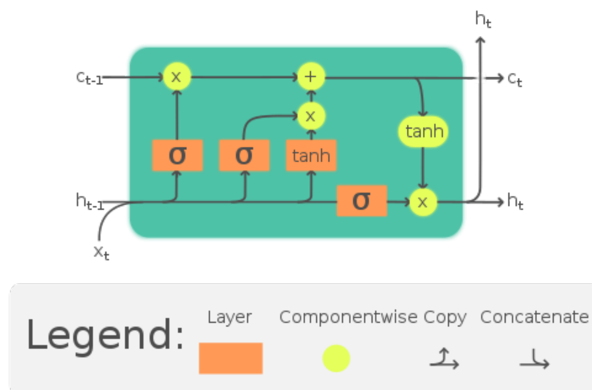

Figure 1. [1]A sample LSTM cell

LSTM are evidently known to perform better than RNN for time-series forecasting, therefore we have tested this hypothesis by implementing a pipeline for predicting energy consumption in the near future. We chose this particular problem for mainly two reasons: Electrical energy price per unit is constant for a state with time, and not varying, which is the case for the US, and majority of EU. Therefore, this presented an interesting opportunity to explore this phenomenon a little deeper with the help of our models.

Additionally, the prices fluctuate based on the demand/suppy scale. Therefore, if we can accurately predict the energy demand in the future, then government organizations can plan on selling redundant electricity and buying at the right time from their neighboring countries, participating in a collective energy-sharing pool, such as *EEX(European Energy Exchange)*[4], thus possibly minimizing energy loss, and financially effective trade.

2. Related Work

The application of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) in time forecasting has been thoroughly investigated in earlier research. To improve time series prediction accuracy, researchers have taken advantage of LSTM's capacity to record long-term dependencies and RNNs' sequential information processing. Promising outcomes from the integration of these neural network architectures have shown how well they model and predict temporal patterns for a variety of applications.

[5] In this paper, the effectiveness of Gaussian Process (GP) regression for short-term interval electricity price forecasting is evaluated. It performs a performance analysis of different covariance functions, except those based on Automatic Relevance Determination (ARD), because of computational limitations. The main contribution is the forecasting model based on GP regression that is proposed and found to perform better than individual and averaging-based schemes. Additionally, the comparative evaluation of non-ARD covariance functions provides valuable insights.

[8]The authors of this study use a modular day-ahead electricity price forecasting approach that makes use of the capabilities of long short-term memory (LSTM) networks and recurrent neural networks (RNN). Utilizing a time series framework based on a combination of convolutional, RNN, and LSTM neural networks, the proposed model makes use of historical price data, grid weather information, and load forecasts. Precise hourly estimates for day-ahead electricity prices in the German bidding zone are made possible by this complex architecture. Furthermore, the model uses a specialized convolutional neural network to incorporate forecasts of solar and wind power generation. With the help of the sophisticated capabilities of RNN and LSTM models, load estimation, renewable energy forecasts, and historical data are combined to produce a reliable day-ahead prediction of electricity prices.

[2]This study addresses the critical need for accurate energy consumption forecasting and compares various machine learning models using Weka on hourly and daily household energy consumption datasets from Kaggle in this comparative analysis. Multi-layer perceptron, K-Nearest Neighbor Regression, Support Vector Regression, Linear Regression, and Gaussian Processes are among the models tested. Furthermore, ARIMA and VAR models are implemented in Python for forecasting energy consumption in South Korean households with and without weather data. The results show that Support Vector Regression outperforms, followed by Multilayer Perceptron and Gaussian Process Regression, demonstrating their efficacy in predicting precise energy consumption.

[7]This paper introduces a deep recurrent neural network (DRNN) for forecasting day-ahead electricity prices in deregulated markets, which effectively captures the complex dependence structure in multivariate models. The DRNN learns indirect relationships between electricity prices and external factors through its diverse functions and multi-layer structure. The DRNN outperforms single support vector machines (SVM) by 29.71 percent and hybrid SVM networks by 21.04 percent in mean absolute percentage error, providing valuable insights for accurate electricity price predictions in deregulated environments.

[3] This study handles the understudied area of precise long-term electricity price forecasting with high resolution, despite the inherent difficulties brought on by the extreme volatility of electricity prices and their intricate relationships with influencing factors. Hourly electricity prices in Hungary were predicted over short and long-term periods using deep neural network experiments, one of which used a deep neural network with a single ConvLSTM encoder. Promising results were found when methodically comparing different network structures and evaluating the influence of environmental factors like date/time and meteorological data. This demonstrates how sophisticated neural network architectures may be useful for handling the complex dynamics involved in long-term electricity price prediction.

3. Dataset

For our project, we have used the `"AEP_hourly"`[6] dataset. This dataset is provided and reported by AEP (Americal Electric Power), which is a big investor-led electricity utilities management company based in the United States.

This data consists of Electric Power consumption (in MW), on an hourly basis, from the past fifteen years.

3.1. Dimensions and size

The dataset has (121273) entries, with two columns. Each row signifies one datapoint for the electricity consumption(in MW), for that hour.

3.2. Data preprocessing

1. Conversion from a single Datetime column to separate columns for Year, Month, Week, Day, Date, and Time, so as to get a better understanding of the variation with respect to each time scale.
Input: (121273, 2)
Output: (121273, 7)

2. Resampling down from 1,21,273 to 5055 entries, by

averaging values throughout the duration of 24 hours, into a single day.

Input: (121273, 7)

Output: (5055, 4)

4. Exploratory Data Analysis

Some useful statistical metrics gathered from the dataset:

1. Mean power consumption: 15499 MW
2. Standard Deviation: 2591.4 MW
3. Minimum consumption: 2591.399065 MW
4. Lower Quartile: 13630 MW
5. Upper Quartile: 17200 MW
6. Max power consumption: 25695 MW

Steps ensure consistency in the Dataset:

1. Checking for duplicate values: None
2. Checking for null (or zero) values: None

Some insights and local trends:

1. YEARLY TREND:

(Fig. 2) shows the variation of energy consumption per year. Peak electricity consumption is noticed for the 2007-2008 time period.

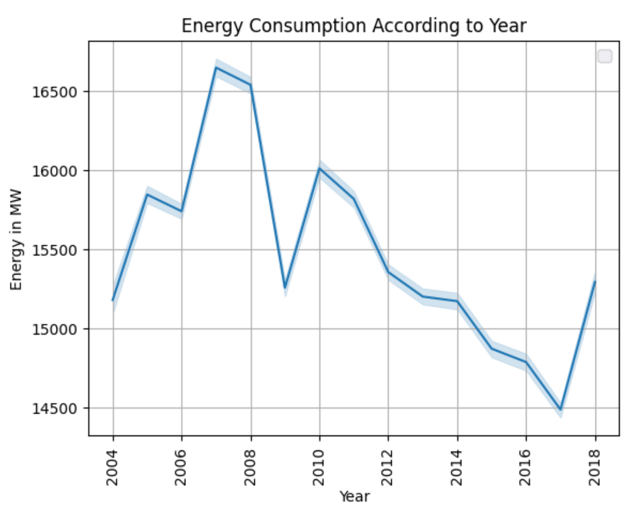

Figure 2. Yearly Trend

2. MONTHLY TREND: (Fig. 3) shows the variation in energy consumption during a year. Peak electricity consumption can be clearly seen for the months of January and December, accounting for colder weather, and increased requirement for space heaters.

Similarly, a dip in consumption can be observed for April, and October, due to the moderate climate, hence less requirement for thermostats.

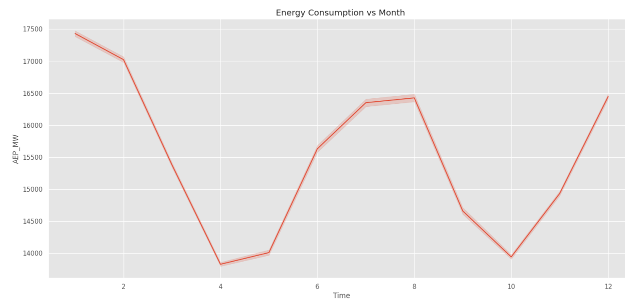

Figure 3. Monthly trend

3. DAILY TREND:

(Fig. 4) shows the trend during the day per hour. It is observed that the demand increases at around 11:00 and stays relatively high until 22:00. This trend could be a reflection of workspaces, and businesses opening their operations, and their closing times.

Similarly, a peak minimum is observed at 05:00, and this could be due to the lowest activity pertaining to energy consumption since a majority of the population is at rest during this time.

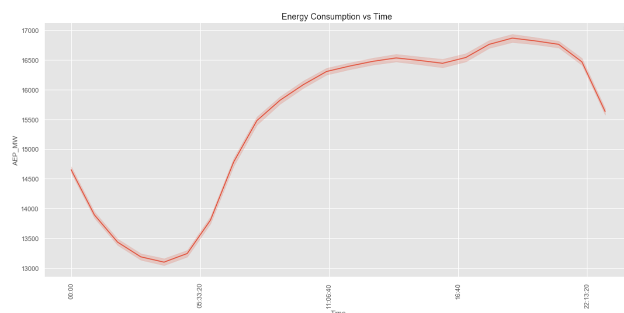

Figure 4. Daily Trend

5. Approach

To compare and reach at a conclusive best Machine Learning model for Time series forecasting, performance between RNN and LSTM has been compared with relatively similar architectures and optimization techniques.

Furthermore, to explore alternative approaches to the problem, a 1-D CNN has also been designed to test how well it performs in comparison, for the said task.

The following sections underline all three model architectures, and the relative performance among these three.

6. Model Architecture

Three separate Machine Learning models were created to compare the performance and infer the most effective architecture for time series forecasting applications.

6.1 Simple RNN

6.2 Long-Short-Term Memory (LSTM) RNN

6.3 1-D Convolutional Neural Network

6.1. Simple RNN

A recurrent neural network (RNN) comprising four layers of basic RNNs and a dense output layer. The activation function employed is the hyperbolic tangent (tanh). To mitigate overfitting, a dropout layer with a 0.2 rate is incorporated. During compilation, the Adam optimizer, mean squared error as the loss function, and accuracy as the evaluation metric are utilized. The model is fitted to the training data for 20 epochs with a batch size of 2. The model architecture's summary includes the number of parameters in each layer and the overall count of parameters in the entire model.

Layer (type)	Output Shape	Param #
simple_rnn_20 (SimpleRNN)	(None, 60, 50)	2600
dropout_11 (Dropout)	(None, 60, 50)	0
simple_rnn_21 (SimpleRNN)	(None, 60, 50)	5050
simple_rnn_22 (SimpleRNN)	(None, 60, 50)	5050
simple_rnn_23 (SimpleRNN)	(None, 50)	5050
dense_15 (Dense)	(None, 1)	51

Total params: 17801 (69.54 KB)

Figure 5. RNN Model Architecture

6.2. LSTM

A 4-layer LSTM model, with additional dropout layers, was implemented. Each layer consists of 50 perceptrons until the final Dense layer coalesces to provide with a single output. Figure 6 shows the model summary below.

Alternating with the LSTM layers, Dropout layers have been incorporated with a Dropout value of 0.1, to overcome over-fitting, if any. Mean Squared Error is used as the loss function, and two optimizers were used, namely: Adam, and SGD.

Layer (type)	Output Shape	Param #
lstm_16 (LSTM)	(None, 60, 50)	10400
dropout_4 (Dropout)	(None, 60, 50)	0
lstm_17 (LSTM)	(None, 60, 50)	20200
dropout_5 (Dropout)	(None, 60, 50)	0
lstm_18 (LSTM)	(None, 50)	20200
dropout_6 (Dropout)	(None, 50)	0
dense_4 (Dense)	(None, 1)	51

Total params: 50851 (198.64 KB)
Trainable params: 50851 (198.64 KB)
Non-trainable params: 0 (0.00 Byte)

Figure 6. LSTM Model Architecture

6.3. 1-D CNN

The first layer is the Conv1D layers which has 64 filters with kernel size 3 and rectified linear unit (ReLU) activation. The input data is transformed by this convolutional procedure, yielding an output tensor of shape (58, 64). Then, the spatial dimensions are reduced by applying a MaxPooling1D layer with a pool size of 2, which produces an output tensor of (29, 64). To avoid overfitting, the network includes a Dropout layer with a 0.2 dropout rate that facilitates regularization.

After these pooling and convolutional procedures, the tensor is reshaped into a one-dimensional vector using a flatten layer. The mean squared error is used as the loss function in the overall architecture, which is assembled using the Adam, SGD and RMSProp Optimizers and their performances are compared.

Layer (type)	Output Shape	Param #
conv1d_5 (Conv1D)	(None, 58, 64)	256
max_pooling1d_5 (MaxPooling1D)	(None, 29, 64)	0
dropout_5 (Dropout)	(None, 29, 64)	0
flatten_5 (Flatten)	(None, 1856)	0
dense_5 (Dense)	(None, 1)	1857

Total params: 2113 (8.25 KB)
Trainable params: 2113 (8.25 KB)
Non-trainable params: 0 (0.00 Byte)

Figure 7. CNN Model Architecture

7. Experiments

Deploying various combinations of the number of layers, optimizers, and dropout rates, here we compare the performance of each of the three models individually to arrive at the best hyperparameters for all three architectures.

7.1. RNN

1.The model performance is similar for the model under SGD,Adam optimizer and the base model.The loss function for both Adam and SGD followed a similar trend.Vanilla RNN performance is not boosted by the optimizers.

2.The base model performs efficiently and better without optimizer's.(Fig.8)

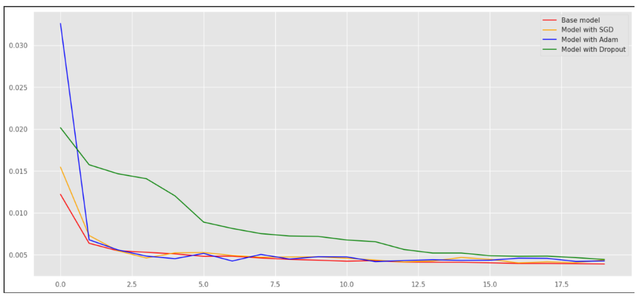

Figure 8. Loss function vs epochs for different Simple RNN configurations

7.2. LSTM

A slight performance boost can be observed when switching from SGD, to Adam as the optimizer. This can be due to that fact that unlike SGD, Adam dynamically computes and changes the learning rates, based on the previous gradients.

Also, it is observed that the model performs better without Dropout. A possible reason for this could be that the model is not experiencing any over-fitting, since that is where Dropout regularization helps.

Figure 9 shows the loss over time for all configurations as discussed earlier.

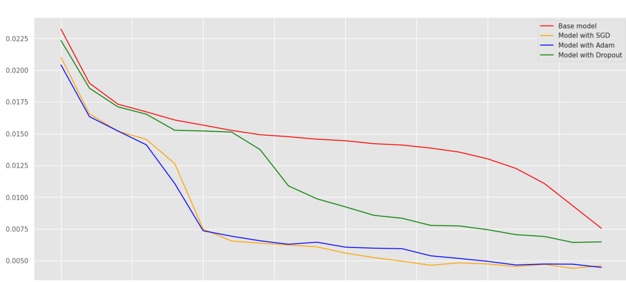

Figure 9. Loss function vs epochs for different LSTM configurations

7.3. 1D CNN

Among the employed optimizers namely SGD, RMSProp, and Adam, it is observed that Adam outperforms SGD and exhibits a slight advantage over RMSProp. This could be due to Adam's combination of adaptive learning rates and momentum.

The model performs better at lower rates of dropout. This observation indicates that the model is likely complex enough to capture and generalize patterns in the data without the need for high regularization.

Figure 10 shows the loss over time for all configurations as discussed earlier.

Figure 10. Loss function vs epochs for different CNN configurations

8. Results

Comparing the best versions of the three architectures, namely: RNN, LSTM, and 1-D CNN, based on the loss function, and how quickly they converge, we conclude that LSTM performs better than RNN marginally, while outperforming CNN with a bigger margin.

Figure 11 shows the final predicted electricity consumption, v/s the actual output on the test dataset. It is evident that the model is able to predict and replicate the daily, monthly, and seasonal variations, effectively as visible from the peaks and troughs. There are slight inconsistencies partially due to the societal variations, but the general trend is essentially captured with our LSTM model.

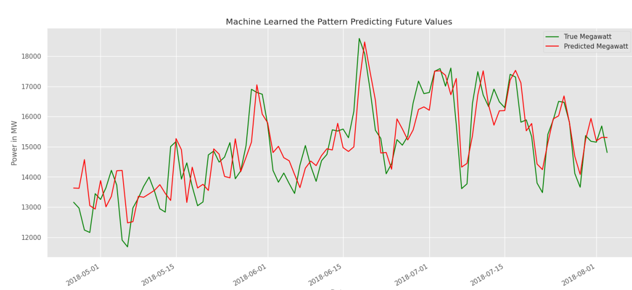

Figure 11. LSTM: Predicted vs actual electricity consumption

9. Limitations

Despite the better performance of LSTM over RNNs and 1-D CNN, a major limiting factor is that LSTM traditionally requires larger datasets to learn the behavior of the time series effectively, thus might not be applicable in situations with limited dataset.

Furthermore, the second limitation comes from the computational power required. In comparison to RNNs, training a LSTM model can be more computationally expensive and, hence could be a barrier in certain applications.

10. Conclusions

In conclusion, the performance was compared between RNN, 1-D CNN, and LSTM, for time-series forecasting applications. In particular, to predict future electricity consumption. It is observed that LSTM performs almost comparable to that of the traditional RNN. This could be due to the lack of feature variation with time for our particular dataset, hence there were minimal behaviors and patterns to learn and predict from.

However, one interesting observation came from the performance of 1-D CNN. Despite it's less prevalent use for time series forecasting, we achieved relatively comparable results between itself, and LSTM. After tuning the hyperparameters, the loss was comparable with less time taken to train the model. Therefore, this could be beneficial in practical applications, with limited resources.

11. Future Work

We suggest comparing the performance of LSTM against other intriguing designs, such as Random Forest and XGBoost, in order to enhance the thoroughness of our work. Important insights into the advantages and disadvantages of each model in forecasting future electricity usage will be provided by this comparative analysis.

We also acknowledge that improving the LSTM model could lead to better prediction accuracy. This methodology guarantees a more sophisticated comprehension of the prognostic potential of distinct architectures and lays the foundation for improving the precision of energy usage projections.

References

[1] Long short-term memory.

[2] M. Bilal, H. Kim, M. Fayaz, and P. Pawar. Comparative analysis of time series forecasting approaches for household electricity consumption prediction, 2022.

[3] G. Dombi and T. Dulai. Hourly electricity price forecast for short-and long-term, using deep neural networks. *Acta Universitatis Sapientiae, Informatica*, 14:208–222, 02 2023.

[4] Europex. Europex energy markets.

[5] N. Mehmood and N. Arshad. Interval forecasting of hourly electricity spot prices using rolling window based gaussian process regression, 2020.

[6] Robikscube. Hourly energy consumption dataset. Accessed: December 15, 2023.

[7] C. Zhang, R. Li, H. Shi, and F. Li. Deep learning for day-ahead electricity price forecasting. *IET Smart Grid*, 3(4):462–469, 2020.

[8] W. Zhang, F. Cheema, and D. Srinivasan. Forecasting of electricity prices using deep learning networks, 2018.