# RNN 을 이용한 태양광 에너지 생산 예측

리아크 무사다르, 변영철, 이상준
제주대학교 컴퓨터공학과
e-mail : liaq192@gmail.com, ycb@jejunu.ac.kr, sjlee@jejunu.ac.kr

# Solar Energy Prediction using Environmental Data via Recurrent Neural Network

Mudassar Liaq, Yungcheol Byun, Sang-Joon Lee (Corresponding Author)
Dept. of Computer Engineering, Jeju National University

요 약

Coal and Natural gas are two biggest contributors to a generation of energy throughout the world. Most of these resources create environmental pollution while making energy affecting the natural habitat. Many approaches have been proposed as alternatives to these sources. One of the leading alternatives is Solar Energy which is usually harnessed using solar farms. In artificial intelligence, the most researched area in recent times is machine learning. With machine learning, many tasks which were previously thought to be only humanly doable are done by machine. Neural networks have two major subtypes i.e. Convolutional neural networks (CNN) which are used primarily for classification and Recurrent neural networks which are utilized for time-series predictions. In this paper, we predict energy generated by solar fields and optimal angles for solar panels in these farms for the upcoming seven days using environmental and historical data. We experiment with multiple configurations of RNN using Vanilla and LSTM (Long Short-Term Memory) RNN. We are able to achieve RSME of 0.20739 using LSTMs.

**Index terms**: Machine learning, Vanilla RNN, LSTM, Solar

## I. INTRODUCTION

Energy is primary requirement for any household now a days. There are many sources from which its harnessed. Most common ones are coal with 40.8 and natural gas with 21.6 percent [1]. Both of these resources are not clean and they produce content which is harmful to environment as by product. Due to these factors voice for clean energy is raised regularly. Many governments across the globe have started to work on replacing these facilities with clean and viable alternatives. Biggest contender in this regard is solar energy which is both clean and hygienic. To harness solar energy, solar farms are created where large number of solar panels are concentrated which produce high volumes of energy. To get the best from these solar fields, angles of solar plate to sun becomes critical.

In terms of predicting solar energy few relevant works are done. In [2] authors work to create a tool to address the complexity of tools required to manage the complexity of a solar energy plant. Their tool utilizes Artificial Neural Networks (ANNs) to predict the parameters that are involved in solar energy prediction. They achieved accuracy of 0.5 to 9 percent in their work. Next most relevant work is [3] in which authors have used Genetic Algorithms to decide initial weights of ANN. This ANN is then used to predict the power generation of the grid. Them achieve good performance for value of R2 grater then 0.9. In [4] authors have predicted wind energy using neural networks. They experimented with multiple models and came up with a model with best results of error around 9 percent. They also analyzed the effect of uneven distribution of wind speed and direction on energy prediction in this paper. In [5] authors utilized multiple machine learning algorithms (Support vector machine and Gradient Boosting) with different feature selection algorithms (Linear, RileafF, and local information analysis). They predicted the solar energy based on latitude and longitude among other factors. They predicted the energy for location with existing history i.e. existing solar energy farms as well as new locations i.e. prospect location for new solar farms.

In this work we try to predict the phenomenon of optimal angles for a specific solar field along with expected energy prediction using local and environmental features.

## II. Methodology

For this project we received dataset with 20 different files with data spanning from different solar fields. These files had different formats of data. There were 8 generic fields in each file, with some files containing some extra fields as well. These common fields were date, Client, Angle1, Angle2, temperature, humidity, total sunshine hours and energy production for that day. Out of these fields Client was not related to energy production. Rest of the fields had a direct or indirect effect on energy generation. In field "Date" contained the date for which readings were recorded. Angle1 was the measurement for the vertical angle of plate i.e. angle of solar panel with respect to Y axis. Angle2 was measurement of horizontal angle i.e. angle with respect to x-axis. Total Sunshine hours were the time solar panels received sunlight. All the files spanned the data for multiple solar fields starting from 1st January 2017 till 31st December 2017. We converted dataset for all these files into a single file with uniform format containing aforementioned fields. The combination file resulted in 87000 readings of solar panels spanning 1 complete year. All this data is summarized in Table-1.

**Table 1 Raw Solar Logs Summary**

| Feature | Value |
|---|---|
| Basic Data Input Type | Car Entry Logs |
| Data Format | Date, Client Name, Angle1, Angle2, Temperature, Humidity, Total Sunshine Hours, Energy Produced. |
| Logs Start Date | 1st January 2017 |
| Logs End Date | 31st December 2017 |
| Total Records | 83000 Entries |
| Records Span | 1 Years (365 Days) |
| Aggregation Level | Days |
| Train Data | 70 % |
| Test Data | 30 % |
| Normalization Formula | = (Feature – Min) / (Max – Min + e). |

On this data we applied feature normalization. Purpose of feature normalization is to make sure that each feature is scaled in uniform scale ([0,1] in our scenario) as it helps prevent initial bias of neural network to any specific feature. We used following equation for feature normalization.

*Normalized Feature = (Feature - Minimum Value) / (Maximum Value – Minimum Value + e).*

*Where 'e' is Euler's constant added to avoid DEVISION BY 0 error.*

For a neural network to work the dataset is usually divided into two uneven chunks. Major one is used for training the neural network and minor is utilized for testing. For this purpose, we also divided our dataset into two parts as well on seventy-thirty ratio. 70 percent data was used for training and 30 percent for testing.

Neural Networks are majorly divided into two major types. First are Convolutional Neural Networks. These networks are mostly used for classification problem where input needs to be classified into predefined categories. Second type is Recurrent Neural Networks which are mostly used for prediction purposes. They are especially useful in case of time series data where data is not only dependent upon its current state but also on some other values in time. To simply state RNNs are good in capturing dependencies in data with respect to timeline. For very same reason we chose to use RNNs for this problem. We used two different variations of RNNs as for this project. First is Vanilla RNN which has standard memory cell. Second one is LSTM (Long Short-Term Memory) which has improved memory element with capability to remember selective knowledge instead of most recent one.

**Table 2 RNN Configurations**

| Feature | Value |
|---|---|
| RNN Type | Vanilla, LSTM |
| Input Layer | 1 * [7 days * 8 features] = 56 neurons |
| Output layer | 1 * [7 days * 3 features] = 21 neurons |
| Hidden Layers | 3,5,7,10,12,14 |
| Learning Rate | 0.1,0.01, 0.001, 0.001, 0.005, 0.0001, 0.00001 |
| Sequence length | 7 days (7 * 8 features = 56) |
| Iterations | 1,000, 2,000, 5,000, 10,000, 20,000, 30,000 |

We inputted 7 days of data as input to RNN and output was Optimal angles and energy for upcoming seven days. We experimented with multiple variations of hidden layers from 3 to 14. We also explored many different learning rates ranging from 0.1 to 0.00001. We chose Root Mean Square Error (RMSE) as our error metric. We experimented with MSE (mean square error) but it quickly grew towards with very large numbers eventually becoming NAN (not a number in python). We also varied epochs for training ranging from 1000 to 30,000. All this information is summarized in Table-2.

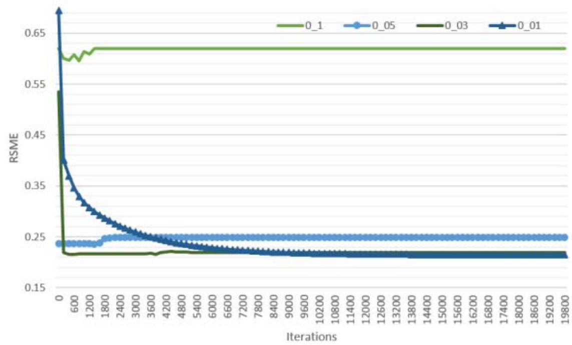


**Figure 1: RSME of LSTM with different configurations**

We started with vanilla RNNs. They were generating RSME of around 0.45 at best with learning rates of 0.0001. Best results were generated using 20,000 iterations. After this point data starts to overfit. The training RMSE drops but increases on Testing data. best results for vanilla RNN were generated with 7 hidden layers.

**Table 3. Vanilla RNN vs LSTMs.**

| Configurations | Vanilla RNN | LSTM RNN |
|---|---|---|
| Iterations | **20,000** | **20,000** |
| Memory | **Standard** | **LSTM** |
| Layers (Optimal) | **7** | **7** |
| Learning Rate (Optimal) | **0.00001** | **0.01** |
| RSME(Optimal) | **0.29793** | **0. 20739** |

For LSTM we got better results around 0.20 using learning rate of 0.01 as shown in Figure-1. Optimal hidden layers were again 7 with same epochs as vanilla RNNs. We also experimented with number of input days, but they were no substantial improvement beyond 7 days. These conclusions are summarized into Table-3.

## III. CONCLUSION

In this paper we proposed an approach to predict the solar energy generation for next seven days based on historical data and weather conditions. We predicted the optimal angles that would produce maximum energy and the expected energy generated using those angles. We were able to reduce RSME to 0.20 which is better than other works (to best of my knowledge). Results show that vanilla RNN was outperformed by LSTM by 0.09 and achieved 0.2043. The reason that LSTM performed better then Vanilla RNN due to its memory component as it helped LSTM to capture time dependencies better. In the future works we would extend this work by incorporating more factors with our dataset along with some architectural improvements.

## ACKNOWLEDGEMENT

This research was financially supported by the Ministry of SMEs and Startups(MSS), Korea, under the “Regional Specialized Industry Development Program(R&D or non-R&D, Project Number: P0003167)” supervised by the Korea Institute for Advancement of Technology (KIAT). This work was supported by 'Jeju Industry-University Convergence Foundation' funded by the Ministry of Trade, Industry, and Energy(MOTIE, Korea). [Project Name: 'Jeju Industry-University convergence Foundation / Project Number: N0002327]

## REFERENCES

[1] Council, W. (2018). [online] Worldenergy.org. Available at: https://www.worldenergy.org/wp-content/uploads/2016/10/World-Energy-Resources-Full-report-2016.10.03.pdf [Accessed 20 Nov. 2018].

[2] Rodríguez, F., Fleetwood, A., Galarza, A. and Fontán, L. (2018). Predicting solar energy generation through artificial neural networks using weather forecasts for microgrid control. Renewable Energy, 126, pp.855-864.

[3] Sameti, M., Jokar, M. and Astaraei, F. (2017). Prediction of solar Stirling power generation in smart grid by GA-ANN model. International Journal of Computer Applications in Technology, 55(2), p.147.

[4] Świątek, B. and Dutka, M. (2015). Wind power prediction for onshore wind farms using neural networks. Renewable Energy and Power Quality Journal, pp.333-338.

[5] Martin, R., Aler, R., Valls, J., & Galvan, I. (2015). Machine learning techniques for daily solar energy prediction and interpolation using numerical weather models. Concurrency And Computation: Practice And Experience, 28(4), 1261-1274. doi: 10.1002/cpe.3631