5-2018

Deep Neural Network Regression and Sobol Sensitivity Analysis for Daily Solar Energy Prediction Given Weather Data

Yixuan Sun
Purdue University

DEEP NEURAL NETWORK REGRESSION AND SOBOL SENSITIVITY ANALYSIS FOR DAILY SOLAR ENERGY PREDICTION GIVEN WEATHER DATA

A Thesis

Submitted to the Faculty

of

Purdue University

by

Yixuan Sun

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Mechanical Engineering

May 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Guang Lin, Chair

School of Mechanical Engineering

Dr. Ilias Bilionis

School of Mechanical Engineering

Dr. Carlo Scalo

School of Mechanical Engineering

Approved by:

Jay P. Gore

Head of the School Graduate Program

This is dedicated to my mentors, my friends and my parents.

ACKNOWLEDGMENTS

I would like to thank Professor Lin for his expert advise and encouragement, as well as Professor Bilionis and Professor Scalo for their help and suggestions. I would also like to thank my fellow students and friends, Hongshan Li, and Pranev Jain for their contribution to this research. I am grateful to all of those whom I have had pleasure to work with during my master program. This work would not be possible if I had not had their support and inspiration.

My friends, Dr. Fei Han, Zhiyao Yang, Gonghao Sun and Lifeng Chen have been such an important part of the pursuit of this research. I would like to thank them for sharing their experiences, and their support and caring. Most importantly, I want to thank my supportive and loving parents for the emotional and financial support, which allows me to work toward my master degree.

PREFACE

This research is the master's thesis as a conclusion of my master program at school of Mechanical Engineering, Purdue University. The basis of this research stemmed from my interest in the applications of deep learning techniques in real-world engineering problems. With the suggestion from my advisor, Dr. Guang Lin, and my experience of dealing with solar energy harnessing during undergraduate study, the idea that using weather forecasting information to predict solar energy came to my mind.

Solar energy, as one of the most popular sustainable energies, has been utilized in many ways. Its dependency on weather condition makes it unstable as an energy source. Accurate forecasts for solar energy are increasingly important for management of electricity grid and solar energy trade. In this research, combined with deep learning regression technique, a predictive model has been developed, only taking weather forecasting information as input.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SYMBOLS

y	target value
$\hat{y}$	predicted value
X	feature vector
b	bias vector
D	variance/partial variance
f_0	mean of the model output
S	Sobol sensitivity indices
n	sample size
x^{stand}	standardized feature
μ	sample mean for a certain feature
σ	sample standard deviation for a certain feature
θ^t	weight for current iteration
θ^{t+1}	weight for the next iteration
w_{ij}^k	weight for node j in layer l_k for incoming node i
b_i^k	bias for node i in layer l_k
a_i^k	product sum plus bias (activation) for node i in layer l_k
o_i^k	output for node i in layer l_k
r_k	number of nodes in layer l_k
r^2	coefficient of determination

ABBREVIATIONS

MLP	multi-layer proceptron
ANN	artificial neural network
GEFS	global ensemble forecasting system
FAST	Fourier amplitude test
EFAST	extended Fourier amplitude test
MAE	mean absolute error
MSE	mean squared error
PV	photovoltaic

NOMENCLATURE

sigmoid	sigmoid function
tanh	hyperbolic tangent function
ReLU	rectified linear unit

GLOSSARY

Activation function	allowing neural networks to learn complex decision boundaries.
Backpropogation	an algorithm to efficiently calculate the gradients in a neural network.
Batch normalization	a technique that normalizes layer inputs per mini-batch.

ABSTRACT

Sun, Yixuan M.S.M.E., Purdue University, May 2018. Deep Neural Network Regression and Sobol Sensitivity Analysis for Daily Solar Energy Prediction Given Weather Data. Major Professor: Guang Lin, School of Mechanical Engineering.

Solar energy forecasting plays an important role in both solar power plants and electricity grid. The effective forecasting is essential for efficient usage and management of the electricity grid, as well as for the solar energy trading. However, many of the existing models or algorithms are based on real physical laws, where tons of calculations, step-by-step modification, and many inputs are required. In this research, a novel deep Multi-layer Perceptron (MLP) based regression approach for predicting solar energy is proposed, in which the inputs are only ensemble weather forecasting data. The results demonstrate that our proposed deep Multi-layer Perceptron based regression approach for solar energy forecasting is efficient as well as accurate enough. A Sobol sensitivity analysis is performed over the trained model, determining the most important variables in the weather forecasting model data. The first-order and the total order Sobol sensitivity indices for quantifying feature importance, are calculated for each model input parameter. With using the process of feature removal, the result of Sobol sensitivity analysis is verified.

1. INTRODUCTION

Deep learning has been widely used in supervised and unsupervised problems, where it does not require much of the careful feature engineering and considerable domain expertise over the raw data. Instead, deep learning methods are representation-learning methods with multiple levels of representation, where there are simple but non-linear modules that transform the representation from the raw input data into a representation at a higher, slightly more abstract level, layer after layer.[6]. Deep neural networks have outstanding performance of solving problems in many domains of science, business and government[12], and it has been proved to have promising results in classification and regression problems. A standard neural network consists of neurons, which are connected processors. In each neuron, there will be a sequence of real-value activation produced. Normally, the calculation from the previous layer to the next layer of deep neural network can be represented as follow:

$$y = \mathbf{w}^T X + b \tag{1.1}$$

where X is the input vector, w is the weight matrix for input X, b is the bias vector, y is the output vector. In each node, there is a non-linear activation function that can approximate arbitrarily complex functions. There are three commonly used activation functions: `sigmoid`, `tanh`, and `ReLU`. Among those, `sigmoid` and `tanh` will squash the input to a range of (0,1) and (-1,1), respectively. `ReLU` does not activate all the neurons at the same time, where only the input with negative values will be activated. The fact that limited number of neurons need activating makes the network sparse, resulting in higher computational efficiency. `ReLU` has been applied in solar energy forecasting network in the following sections.

The input data consists of multiple features, where some features affect the output more than the others. Thus, another aspect in this research is to find out which

parameter or set of parameters has the most significant influence on the output. This is usually done by sensitivity analysis. A technique called Sobol sensitivity analysis is conducted in this work.

A feed forward neural network with fully connected layers has been used for daily solar energy prediction in this research, followed by a Sobol sensitivity analysis over all the input variables. The following sections are the introductions of solar energy prediction technique and the mechanism of Sobol sensitivity analysis.

1.1 Solar energy forecasting techniques

Renewable energy sources has been proved to have many environmental advantages over traditional fossil fuels for generating electricity. But the energy such as solar and wind fluctuate with the changing weather conditions. The accurate forecasting of solar energy a location is able to receive is vital for electric utility companies to make adjustment in advance to have the right balance of renewable and fossil fuels available. Errors in the forecasting could lead to large expenses for the utility from excess fuel consumption or emergency purchases of electricity from neighboring utilities. Power forecasts typically are derived from numerical weather prediction models, but statistical and machine learning techniques are increasingly being used in conjunction with the numerical models to produce more accurate forecasts.

The effective prediction of solar energy for certain area is essential for photovoltiac power plants and electricity grids. The efficient usage and management of electricity grid and the solar energy trading would benefit from accurate solar energy forecasting. PV power forecasting is a great concern for operators and designers of power systems because of its variable and volatile features [19]. Some common forecasting methods include regressive or stochastic learning models and remote-sensing or local-sensing based physical models [2]. Hybrid models integrating stochastic learning and local sensing techniques have been applied these years for intra-hour forecasting. In addition, the application of artificial neural networks (ANN) has made contribution

to solar energy forecasting. Mellit et al. proposed an MLP model to forecast the solar irradiance on a base of 24-h using the present values of the mean daily solar irradiance and air temperature [13]. A recurrent neural network and an MLP network are used in [20] to generate solar radiation synthetic series. However, these approaches did not take weather forecasting models into account, especially the daily cloud cover of specific locations. Generalized deterministic solar radiation models are introduced, where latitudes, longitudes, cloud cover and sky clearness index are inputs, resulting that the error in calculated insolation values are within 20% of the measured values[11].

In this research, a solar energy forecasting method based on deep fully connected neural network regression has been developed, using location information (latitudes and longitudes) and ensemble weather forecasting models as inputs. A 2-dimensional interpolation is employed during the data preporcessing stage to get the solar energy for each location that has the corresponding weather data.

1.2 Sensitivity Analysis

The weather data used in this research contains 15 different weather variables and there are 5 reports for each day. We want to know which individual variable or groups of variables would affect the output the most. Sensitivity analysis is able to reveal the identification of the model parameter (weather variables in this case) or set of parameters that have the greatest influence on the model output. Sensitivity analysis gives insights about how much certain input parameter or set of input parameters contributes to the variability of the model output. It is widely applied in many fields, such as business, economics, and engineering. The application of sensitivity analysis can be summarized as: (i) understanding the relationship between inputs and output. (ii) determining how much the uncertainty in the model parameters contributes to the output variability. (iii) analyzing the significant parameters that take the lead in model outputs and magnitudes [16][17][8]. Sensitivity analysis is also useful for

determining the uncertainty in input parameters, as well as the model structure, by which we can gain additional confidence in the model[22].

In the sensitivity analysis process, we can regard the model as a black-box, that is, the model output is a function of the inputs $y = f(x)$, where x can be a high dimensional vector. In general, sensitivity analysis can be classified into two categories: local sensitivity analysis, and global sensitivity analysis.

1.2.1 Local sensitivity analysis

Local sensitivity analysis aims to investigate how small variations in a single input parameter would affect the model output. The local sensitivity indices are usually described mathematically as the first order partial derivatives of model outputs respect to the model input parameters.

$$A_i = \frac{\partial y}{\partial x_i}(x_1, x_2, ..., x_p) \tag{1.2}$$

where y is the model output, $x_1, x_2, ..., x_p$ are the model input parameters, and p is the dimension of the input space. Local sensitivity analysis only addresses the sensitivity relative to point estimates chosen, rather than the entire parameter distribution[7], so the interaction between input parameters can not be evaluated by it. Global sensitivity analyses are able to characterize the interaction between input parameters and quantify the effect it imposes on the model outputs, overcoming the limitation brought by local sensitivity analysis.

1.2.2 Global sensitivity analysis

In a global sensitivity analysis, the permutation can happen simultaneously over all the input parameters, allowing the sensitivity evaluation of each individual parameter as well as the interactions between parameters at the same time. It offers the possibility of investigating how the variance of the model output in influenced by the relative impact of a single model parameter and the interactions between

parameters[9]. The common global sensitivity analysis are regression analysis, screening, the Fourier amplitude sensitivity test (FAST), extended Fourier amplitude sensitivity test (EFAST) and variance based methods.

Global sensitivity analysis has been applied to various fields. For example, [3] studies the use of global sensitivity analysis for design optimization of shell and tube heat exchangers; in[4], the extended Fourier amplitude sensitivity test is used for the selection of input variables of neural networks; in [10], one can evaluate and optimize the performance of probabilistic neural network using Sobol, FAST and EFAST global sensitivity analysis methods; Sobol sensitivity analysis is used in the evaluation and parameter selection for flow simulations of the river Kleine Nete in[14].

Among those techniques, both FAST or EFAST and Sobol's methods are variance-based, decomposing the variance of the output of the model or system into fractions which can be attributed to inputs or sets of inputs. The difference is that in FAST method, a sinusoidal function is used for the pattern search for multidimensional integration, while a Monte Carlo integration method is employed in Sobol sensitivity analysis[22].

In this research, a Sobol sensitivity analysis is used to investigate the importance of each variable as well as the interactions between variables in weather data, respect to the trained neural network's output.

1.2.3 Sobol sensitivity analysis

Sobol sensitivity analysis is based on the decomposition of model output variance into summands of variances of the input parameters in increasing dimensionality[18][21]. It determines the contribution of each input parameter and their interactions to the overall model output variance.

Sobol sensitivity analysis is applied in quantifying how much variability in the model output is depending on each of the input parameter or the interactions between parameters. It can be achieved by computing the first-order, second-order,

higher-order, and the overall sensitivity indices. Below, we show the mathematical description of the analysis process.

Let $\mathbf{x} = (x_1, x_2, ..., x_N)$ be a set of input parameters that are mutually independent. Each parameter has a finite interval that can be [0,1] after rescaling. We can think of each parameter as a random variable which is uniformly distributed over the interval of [0,1]. Consider an integrable function $f(x)$ whose sensitivity to the input parameters $x_1, x_2, ..., x_N$ defined in I^N, where I denotes [0,1] interval and I^N is a N-dimensional hypercube. Then, we have:

$$f(x) = f_0 + \sum_{i=1}^{N} f_i(x_i) + \sum_{i=1}^{N}\sum_{i\neq j}^{N} f_{ij}(x_i, x_j) + f_{1...N}(x_1, ..., x_N) \tag{1.3}$$

where f_0 is the mean value of $f(x)$, and the expression of $f_i(x_i)$ and $f_{ij}(x_i, x_j)$ are listed below.

$$f_0 = \int_0^1 f(x)dx$$

$$f_i(x_i) = \int_0^1 f(x)\prod_{k\neq i} dx_k - f_0$$

$$f_{ij}(x_i, x_j) = \int_0^1 f(x)\prod_{k\neq i,j} dx_k - f_0 - f_i(x_i) - f_j(x_j)$$

The procedure is performed until the last term $f_{1...N}(x_1, ..., x_N)$ is determined. $f(x)$ in Eq. 1.3 is called the analysis of variance representation when the condition shown in Eq. 1.4 is satisfied[21]:

$$\int_0^1 f_{i_1,...,i_N}(x_{i_1}, ..., x_{i_N})dx_k = 0 \tag{1.4}$$

where $k = i_1, ..., i_N$. Because of this property, if we square both sides of Eq. 1.3 and integrate it:

$$D = \int_0^1 f^2(x)dx - f_0^2 = \sum_{i=1}^{k} D_i + \sum_{i<j} D_{ij} + \sum_{i<j<l} D_{ijl} + ... + D_{1,2,...,k}$$

where D is the model output variance, and $D_{i_1...i_N} = \int_0^1 f^2_{i_1...i_s}(x_{i1}, ..., x_{i_N})dx_{i1}, ..., x_{i_N}$ is called the partial variance corresponding to the subset of parameters $x_{i1}, ..., x_{i_N}$. The Sobol sensitivity indices of that subset of parameters is defined as:

$$S_{i_1...i_N} = \frac{D_{i_1...i_N}}{D} \tag{1.5}$$

The integer N is called the order or the dimension of the index. For instance, $S_i = \frac{D_i}{D}$ is the first-order contribution of the ith parameter to the model output; $S_{ij} = \frac{D_{ij}}{D}$ is the second-order contribution of the interaction between ith and jth parameter to the model output, and so on. Finally, the total sensitivity indices are defined as the sum of all the sensitivity indices. For the ith parameter, the total sensitivity index is $S_{Ti} = S_i + S_{ij_{i\neq j}} + ... + S_{1...i...s}$, quantifying the overall effect of the ith parameter on the model output.

Note that since all the indices are nonnegative, if we sum all the sensitivity indices of all the parameters, we have

$$\sum_{i=1}^{k} S_i + \sum_{i<j} S_{ij} + \sum_{i<j<l} S_{ijl} + ... + S_{1,2,...,k} = 1 \tag{1.6}$$

In summary, the first-order sensitivity index measures the contribution of an individual parameter to the model output variance. The first-order index of parameter i can be regarded as the fraction of the model output variance that would disappear on average if parameter i is fixed[14]. The overall or total-order sensitivity indices evaluate the contribution to the output variance over a full range of parameter space.

In this research, the first-order and the total-order Sobol sensitivity indices are employed over 75 model parameters to evaluate how important each individual parameter as well as the interactions between certain parameter and the rest parameters is to the model output.

2. SOLAR ENERGY FORECASTING

2.1 Problem settings

The principal idea behind this research is to create a generic model that can predict solar energy received given location in an efficient way, meanwhile maintaining reasonable accuracy. In other words, we want to only take the latitude and the longitude of a specific location along with the data from weather forecasting model as inputs to predict the solar energy that it will receive for the day. The weather may affect solar energy in many different ways, such as cloud cover, precipitation, temperature, wind speed, etc; however, it is hard to find the precise correlation between each weather feature and the solar energy received. Instead, we aim to create a direct mapping from the weather forecasting data to the solar energy. Deep neural networks have the ability to find linear as well as non-linear relationship between features and targets without much of feature engineering or domain knowledge. We intend to use a deep fully connected neural network to find the aforementioned mapping by training the network as a regression problem. Now, let's take a look at the dataset.

2.1.1 Dataset

The dataset used in this work is named GEFS in netCDF4 files. There are 15 netCDF4 files, each representing a particular variable from the ensemble weather data. Every netCDF4 file holds the grids for the variable with time step, stored in a multidimensional array. The first dimension is the total number of days during which the weather data is collected. The second dimension is the ensemble member that the forecast comes from, where there are 11 members with perturbed initial conditions. The third dimension is the times that the weather model runs in a day, which happens once every 3 hours from 12 to 24 hours. The fourth and the fifth dimension are the

latitudes and longitudes of locations in a uniform spatial grid. All the variables used are listed in the table below.

Table 2.1. Variables from weather model.

Number	Name
1	3-Hour accumulated precipitation at the surface
2	Downward long-wave radiative flux average at the surface
3	Downward short-wave radiative flux average at the surface
4	Air pressure at mean sea level
5	Precipitable Water over the entire depth of the atmosphere
6	Specific Humidity at 2 m above ground
7	Total cloud cover over the entire depth of the atmosphere
8	Total column-integrated condensate over the entire atmosphere.
9	Maximum Temperature over the past 3 hours at 2 m above the ground
10	Minimum Temperature over the past 3 hours at 2 m above the ground
11	Current temperature at 2 m above the ground
12	Temperature of the surface
13	Upward long-wave radiation at the surface
14	Upward long-wave radiation at the top of the atmosphere
15	Upward short-wave radiation at the surface

Besides the 15 variables from the ensemble weather model, for each location, there is information of latitudes, longitudes and time steps, encoded in a 5-dimensional array.

2.1.2 Metrics

Mean absolute error

Mean absolute error (MAE) is used to evaluate the performance of the regression analysis in this research. MAE is chosen as it clearly indicates how much the predicted value deviates from the true value. The MAE is defined given below:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{2.1}$$

where n is the number of samples, y_i is the true target value of sample i, $\hat{y}_i$ is the predicted target value of sample i.

Coefficient of determination

Besides the mean absolute error, the coefficient of determination is used to evaluate both the trained model and the baseline model. It is a measure of how well the model replicates or predicts the true values from the data, denoted by r^2. It indicates the proportion of variance in the model output that is predictable from the input parameters. r^2 can be mathematically described as follow:

$$r^2 = \frac{SS_{regression}}{SS_{total}} \tag{2.2}$$

where

$$SS_{regression} = \sum_i (f_i - \bar{y})^2$$

$$SS_{total} = \sum_i (y_i - \bar{y})^2$$

in which f_i is the model output, y_i is the true value in the data, and $\bar{y}$ is the mean value of true outcome in the data.

2.1.3 Data standardization

Due to the 15 variables representing different physical values, the scale of different variables varies. In order to have a more generalized and accurate deep neural network

model, one is required to feed in the data where all features have the similar or the same scale, so that certain feature will not be dominant just because of the large magnitude. The normalization process is shown as follow:

$$x_{i,j}^{stand} = \frac{x_{i,j} - \mu_j}{\sigma_j}, \forall i, j \tag{2.3}$$

where $x_{i,j}$ represents the value of the jth feature in the ith sample ; μ_j is the mean value of the jth feature in the training set; σ_j is the standard deviation of the jth feature in the training set. Note that when apply normalization to validation and testing sets, the mean value μ_j and the standard deviation σ_j are both from the training set, preventing any information leak from the validation set and testing set to the training process.

2.1.4 Two-dimensional linear interpolation

The locations with available weather data do not coincide with the locations with true solar energy measurement. As it is shown in figure 2.1, the blue points are the locations with corresponding weather data, but the red points are the locations that have actual measured solar energy data. Thus, a two-dimensional linear interpolation is used to find the right matching between the target value measured and the weather data.

We assume that each variable to each GEFS grid point among the 15 aforementioned variables in the weather data is linearly depending on the latitude and longitude. In two-dimensional linear interpolation, we perform linear interpolation in one direction first, and again in the other direction. The solution to a bilinear interpolation problem is stated as follow:

$$f(x, y) \approx a_0 x + a_1 y + a_2 xy \tag{2.4}$$

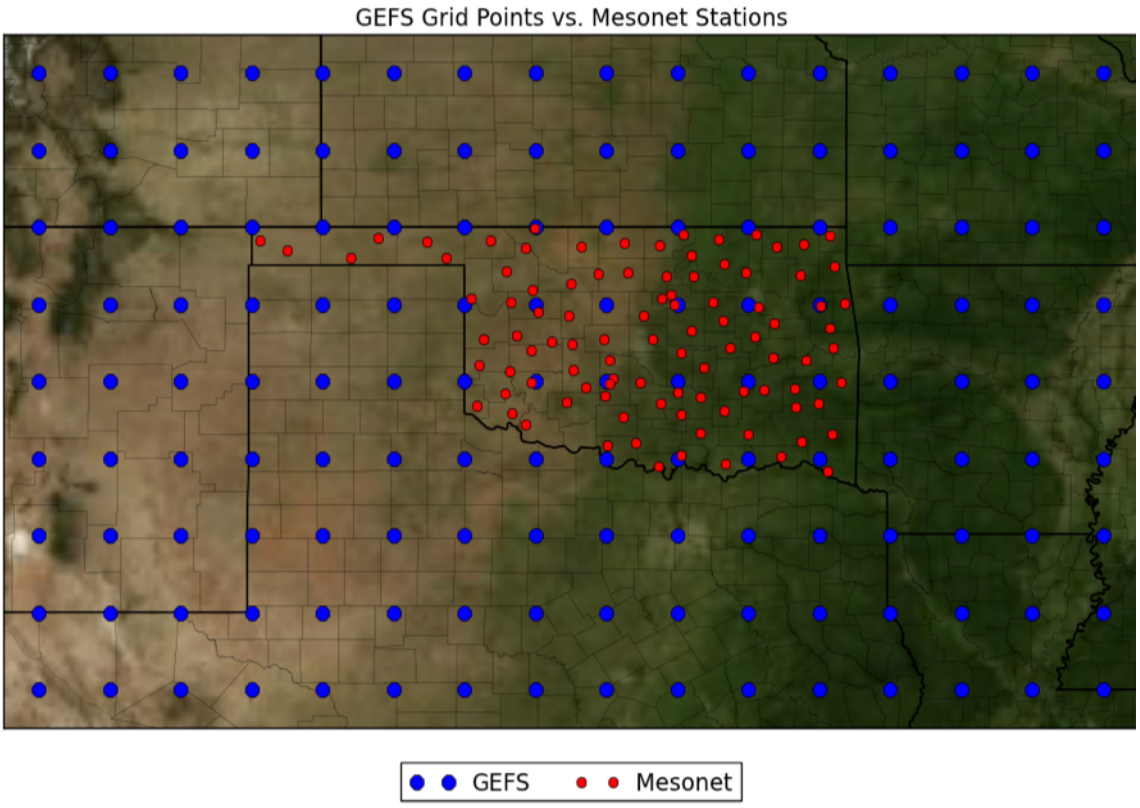


Figure 2.1. Points with GEFS data and points where the measurement taken.

where a_0, a_1, a_2 are weight vectors for matrices x,y, and xy. They can be found by solving:

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_1 & y_2 & x_1y_2 \\ 1 & x_2 & y_1 & x_2y_1 \\ 1 & x_2 & y_2 & x_2y_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f(Q_{11}) \\ f(Q_{12}) \\ f(Q_{21}) \\ f(Q_{22}) \end{bmatrix} \tag{2.5}$$

In this dataset, there are GEFS 96 grid points with corresponding latitudes and longitudes values. We run a two-dimensional interpolation using `Scipy` through the 96 points for each of the 15 weather data variables, and based on the latitudes and longitudes of the 98 Mesonet points, we exterpolated the weather data for each point, getting their weather variables to construct the trainable dataset.

2.2 Deep Neural Network

This section represents the architecture of deep fully connected neural network for solar energy prediction task. The inputs are normalized weather variables calculated from the two-dimensional interpolation. The target values are measured daily solar energy received at Mesonet points.

Deep fully connected neural networks have the potential to identify the nonlinear relationship between input features and the targets. It takes on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations. The deeper it goes, the higher level representation it is able to identify, so that it establishes a proper mapping from input features to their target.

2.2.1 Network structure

The proposed the network is a deep fully connected neural network consisting of 1 input layer, 1 output layer, and 4 hidden layers. We introduce some notations used in the neural network. An instance fed in the neural network is a two-dimensional

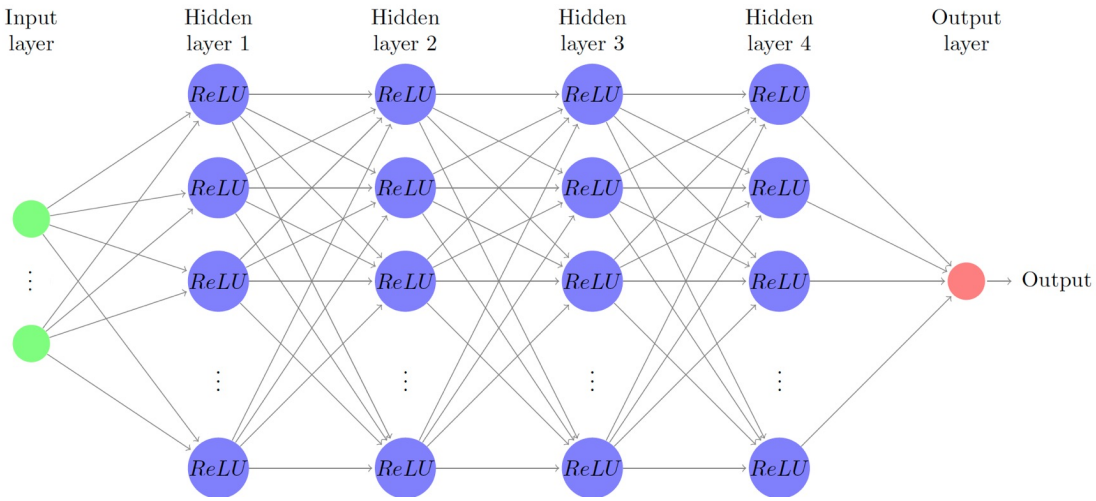


Figure 2.2. The structure of the feed forward neural network.

matrix, where there are N (also known as batch size) data samples and D features. In our case, batch size N needs specifying during training, and the number of features D is 75 due to the shape of the input, the input layers contains 75 nodes, each representing a variable from the weather data. The first hidden layer has the number of nodes of 300, which is sufficiently large to capture all the information contained in the input layer. For the next 3 hidden layers, the numbers of nodes in each layer are 150, 80, 30, respectively. In the output layer, there is only one node since for the regression problem, only one output value per sample is needed. The structure is shown as in Figure 2.2:

For each node in hidden layers, an activation function called rectified linear function ($ReLU$) has been used. The purpose of using activation functions is to introduce non-linearity into the network, so that each layer is able to extract high level representations from the input data.

2.2.2 Feed Forward

The first step during training process is feed forward, where the information moves in only one direction, layer by layer in the network. The basic concept of feed forward

neural network is quite simple: the network is supplied with both a set of input data to be learned and the desired output response for each data sample[15]. When the input information enters a layer, it will be parameterized by weights, which are essentially numbers, or vectors or matrices. The path operation between layers is to add up the output times weights of each node from the previous layer and pass it to each node from the subsequent layer, which can be mathematically described as below:

$$y_j = \sum_{i=1}^{n} (w_{ij} x_i + b_i) \tag{2.6}$$

where n is the number of nodes in the previous layer, x_i is the output from the previous layer (x_i is a feature if the previous layer is the input layer), w_i is the weight, b_i is the bias term. The output y_j can be considered as the input value for the node in the subsequent layer.

The operation inside of the nodes is called activation process, where an activation function is applied to the node's input. In this research, sigmoid function is used. The activation process is shown below:

$$y'_j = ReLU(y_j) \tag{2.7}$$

where y'_j is the output of a node after activation, and

$$ReLU(x) = \begin{cases} 0, & \text{if } x < 0. \\ x, & \text{otherwise.} \end{cases} \tag{2.8}$$

With this choice, only partial nodes in each layer will be activated, not being squashed into certain range, which is widely used for regression problem. The derivative of the $ReLU$ function is in form like this:

$$\frac{d}{dx} ReLU = \begin{cases} 0, & \text{if } x < 0. \\ 1, & \text{otherwise.} \end{cases} \tag{2.9}$$

The derivative of $ReLU$ function is continuous and can be easily computed, being suitable for backpropagation process.

2.2.3 Loss function, Gradient Descent and Backpropagation

Loss function: One of the most important objectives of training period in supervised learning, for example, in this research, is to minimize the difference between the true target value and the output value of the network. As in traditional MLP training for regression task, the mean squared error loss (MSE) is used in the training stage. The MSE is a measure of the quality of the network, indicating the average squared deviations between true value and predicted value. It is always non-negative, and values closer to zero are better. The mathematical expression of MSE is as follow:

$$MSE = \frac{1}{2N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.10}$$

where y_i is the true target value, and $\hat{y}_i$ is the output value of the network.

Gradient descent: In order to minimize the MSE, after each iteration, we need to update the weights in the network to make the predicted value closer to its true value. In this research, gradient descent optimization algorithm is used. The weight update rule is shown as follow:

$$\theta^{\mathbf{t+1}} = \theta^{\mathbf{t}} - \alpha\frac{\partial MSE(X,\theta)}{\partial\theta} \tag{2.11}$$

where θ denotes the parameters (weights and biases) of the neural network at iteration t in gradient descent, α is the learning rate, controlling the step size of updating the weights.

Backpropagation: After each iteration from the feed forward stage, we will have the error value, with the mean squared error loss function. We need the predicted error to propagate back through the network in order to update the weight matrices for each layer. The chain rule of computing derivatives is applied here during the process of getting the minimum value of MSE. There are some notations can be used to describe the backpropagation in the feed forward network:

Using the notation above, backpropagation aims to minimize the MSE by calculating the value of $\frac{\partial E}{\partial w_{ij}^k}$, for each weight w_{ij}^k, where $E = \frac{1}{2}(y_i - \hat{y}_i)^2$. The partial

Table 2.2. Notations in backpropagation.

Notation	Explanation
w_{ij}^k	weight for node j in layer l_k for incoming node i
b_i^k	bias for node i in layer l_k
a_i^k	product sum plus bias (activation) for node i in layer l_k
o_i^k	output for node i in layer l_k
r_k	number of nodes in layer l_k

derivative of MSE with respect to w_{ij}^k can be written as follow according to chain rule and Equation (2.9):

$$\frac{\partial MSE(X,\theta)}{\partial w_{ij}^k} = \frac{1}{N}\sum_{d=1}^{N}(y_i - \hat{y}_i)\frac{\partial \hat{y}_i}{\partial w_{ij}^k} \tag{2.12}$$

where,

$$\frac{\partial \hat{y}_i}{\partial w_{ij}^k} = \frac{\partial ReLU(a_i^k)}{\partial a_i^k}\frac{\partial a_i^k}{\partial w_{ij}^k}, \qquad a_i^k = b_i^k + \sum_{j=1}^{r_k-1} w_{ij}^k o_j^{k-1} \tag{2.13}$$

According to Equation (2.8), $\frac{dReLU(a_i^k)}{da_i^k} = \begin{cases} 0, & \text{if } a_i^k < 0. \\ 1, & \text{otherwise.} \end{cases}$. Thus, the partial derivative has the final form as:

$$\frac{\partial MSE(X,\theta)}{\partial w_{ij}^k} = \begin{cases} 0, & \text{if } a_i^k < 0. \\ \frac{1}{N}\sum_{d=1}^{N}(y_i - \hat{y}_i), & \text{otherwise.} \end{cases} \tag{2.14}$$

This part is the gradient of the network, which will be applied in weight update in Equation (2.10). When it is the input layer, o_i becomes the input variable x_i.

2.3 Experiment and Result

In this section, we have performed an experiment using the previously proposed feed forward neural network, where there are an input layer, 4 hidden layers, and an

output layer. The network is trained and tested using data given. After interpolation, the feature vector of each sample is reshaped into a 75-dimensional vector, because with 5 sets of weather data a day, and 15 weather variables in each set, a sample essentially has 75 features for a day. Thus, the input layer contains 75 nodes, each representing a feature in a sample. The first hidden layers have 300 nodes, 150 nodes, 80 nodes, and 30 nodes, respectively. The output layer only has 1 node due to the regression problem. The *ReLU* activation functions are applied to each node in the hidden layers. The experiment result is compared with the result of a baseline model.

2.3.1 Baseline: linear regression

A linear regression model has been implemented as the baseline model for predicting daily solar energy given weather data in the study. It is a linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X[5]. The linear regression model has the form of:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \tag{2.15}$$

in which $\mathbf{X}$ is the feature vector, 75 variables in this case, y is the target vector, β is the weight matrix and ϵ is the bias vector. The implementation of the linear model is achieved by a Python machine learning library `scikit-learn`. The mean absolute error over the testing set is 0.2254, which indicates that on average, the predicted values using linear regression deviate from the true value by 13.64%. Figure 2.2 shows the testing result of 100 samples in the baseline model, in which it is obvious that the predicted values basically capture the true target value; however, there are some distinct differences around turning points , and the trends of some middle points are completely toward different directions. Based on the result plot, we can clearly see that the baseline model is able to fit well around the mean value but the model performs poorly at the points that are away from the mean value.

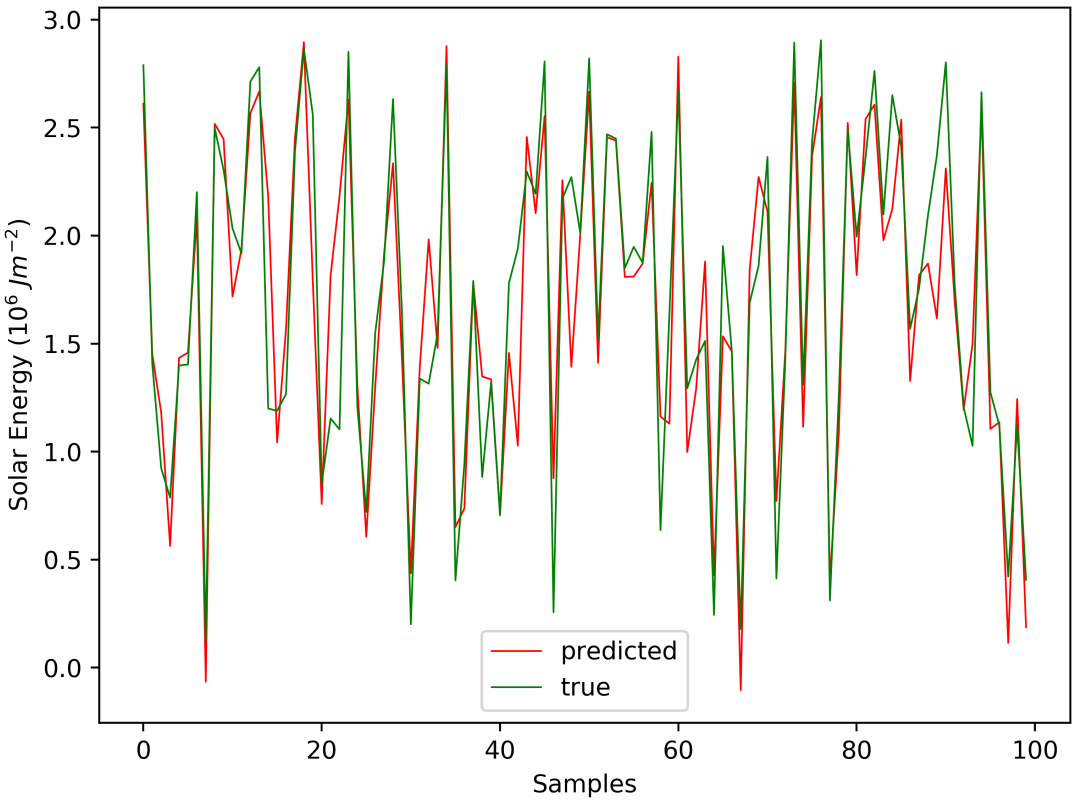


Figure 2.3. Testing result of linear regression baseline model.

2.3.2 Deep neural network

The training procedure can be summarized as follow:

1. Divide the data into training, validation and testing set.
2. Select network architecture and training parameters.
3. Train the model using the training set for each of K-fold (K = 5).
4. Evaluate the model using the validation set for each of K-fold (K = 5).
5. Repeat steps 2–4 using different training parameters.
6. Determine the final training parameters.

The construction and running of the network happen in Python programming language and a deep learning framework called Pytorch. As mentioned before, prior to feeding in the data, the data standardization process is required, where each feature in the raw data is standardized to have mean value of 0 and standard deviation of 1. The target values, daily solar energy, attain their original distribution, but being rescaled (divided by 10^6) to have a reasonable magnitude range. During the training and parameter tuning process, a 5-fold cross validation has been used. The training set in the cross validation is nearly evenly divided into 5 parts, making 4 parts as the training set and 1 part as the validation set. We train the network on the training set, and test it on the validation set. This process is repeated 5 times until the model has been tested on every single part of the 5 split parts.

2.3.3 Result

Figure 2.2 shows the training error and the cross-validation error plots against epochs. In the training process, 100 epochs and batch size 100 have been used when feeding in the training samples. With the learning rate of 0.05, the model converges around epoch 50 when validation error starts increasing, resulting in 0.0432 training error and 0.0572 validation error. Target value has been rescale linearly during the

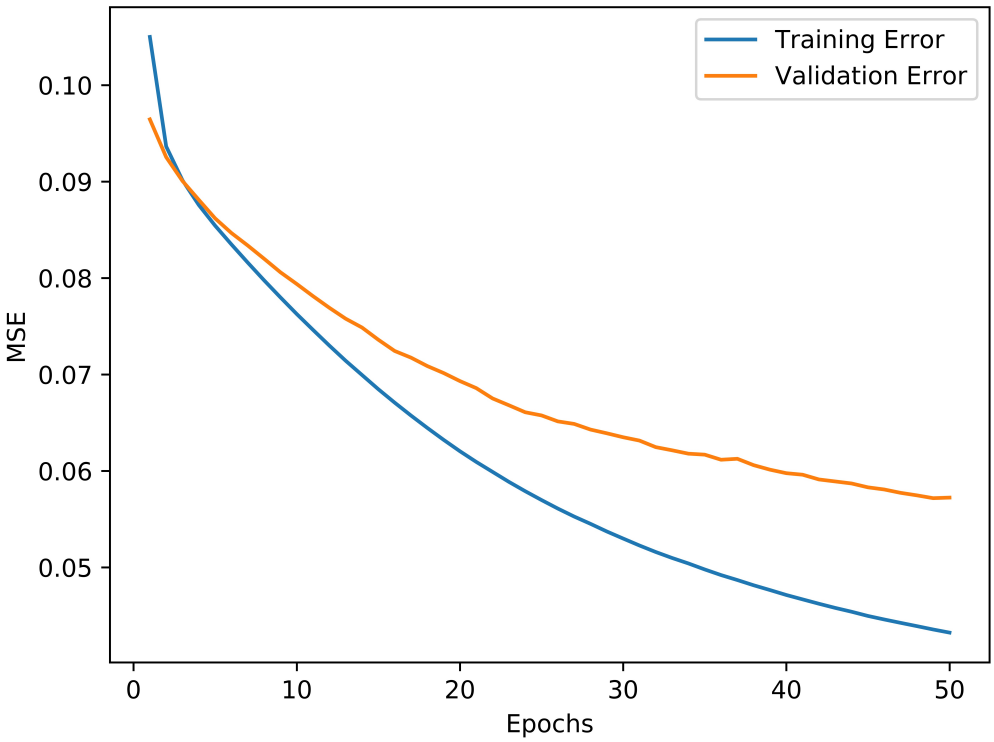


Figure 2.4. The mean squared error for training and validation processes.

Table 2.3. Performance of baseline and trained models.

	Baseline	Trained NN
MAE	0.2254	0.1492
Testing Accuracy	86.36%	90.97%
r^2	0.8363	0.9156

training process due to their large magnitude. After rescaling, the mean value of the target is 1.6526. The metric used to evaluate this model is mean absolute error, as shown in Equation (2.1) and Equation (2.2). The model performance on testing set using aforementioned metrics gives the testing error of 0.1492, which means on average, the predicted value deviates from the true value by 9.03%. The trained model gets a r^2 score of 0.9156, while the baseline model has a r^2 score of 0.8363, indicating that for the trained neural network work, there is 91.56% variability between the true value and the predicted outcome can be accounted for. The comparison between the predicted value and the true value out of 400 testing samples is shown in figure 2.3, where the blue curves represent the predicted value, and the orange curves represent the measured value. Overall, the blue and the orange lines overlap for the most of the part, whereas at some changing points and peak values, there are some observable differences. The trained neural network outperforms the baseline linear regression model. The comparison results are shown in the Table 2.3:

2.4 Conclusion

A feed forward neural network consisting of 4 fully connected hidden layers has been developed as the regressor for daily solar energy prediction given weather information. The mean squared error loss function and gradient descent algorithm are used in training stage. The trained network can achieve a prediction accuracy of 90.97% using mean absolute error as the metric. The trained model outperforms the

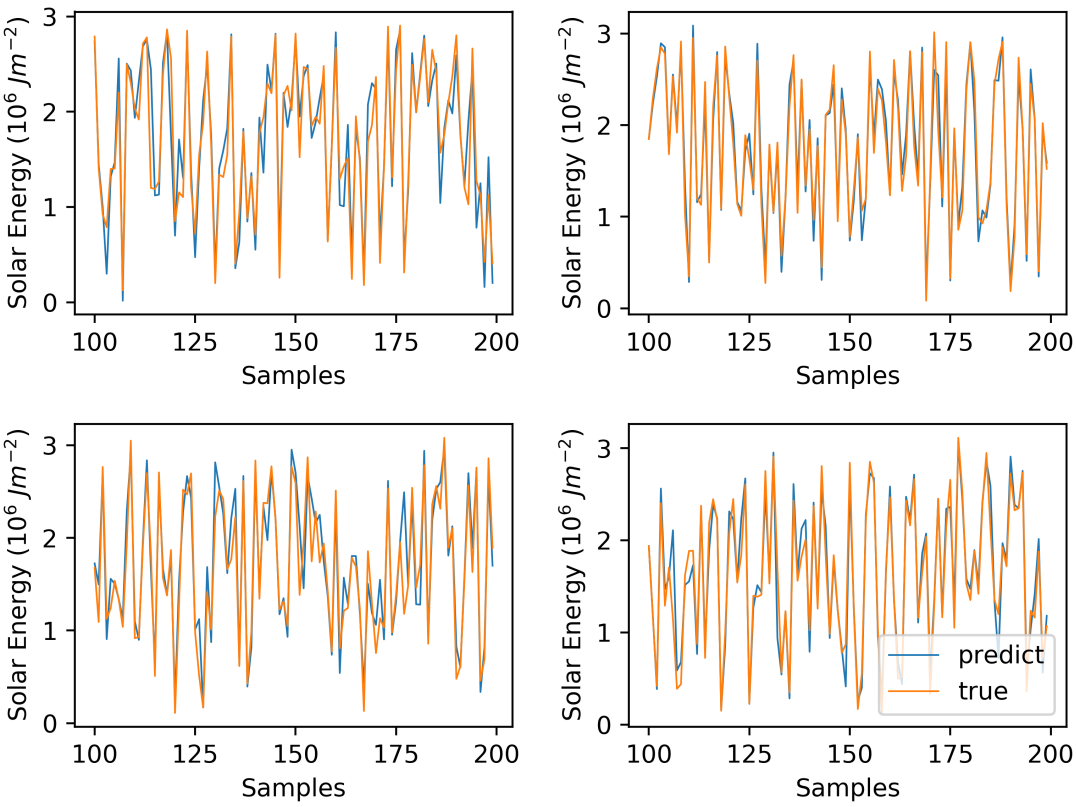


Figure 2.5. Comparison between predicted and measured solar energy.

linear regression baseline model, proving that the trained neural network is accurate and efficient enough to prediction daily solar energy. According to figure 2.4, the predicted values can capture most true values except for some peak areas. As a result, with given well-developed weather forecasting data of a location, consisting of the 15 types of variables used in this research, a reasonably accurate prediction of the amount of daily solar energy received at that location can be made. This model can be useful in solar energy forecasting in order to contribute to the management of electric power grid.

3. SOBOL SENSITIVITY ANALYSIS

The trained model in the previous chapter takes all 75 weather variables into account. However, we want to know which are the most important ones among 75 variables that have more significant contributions to the model output. The Sobol sensitivity indices, both the first-order indices and the total-order indices in Eq.1.5 are calculated for each of the 75 variables.

3.1 Steps for Sobol Sensitivity Analysis

The Sobol sensitivity analysis in this research is conducted using a sensitivity library in Python programming language called `SALib`. The general steps of Sobol analysis in `SALib` are listed below:

- Decide the parameters in the model that need to vary
- Determine the parameter range including the lower and the upper bounds.
- Generate samples based on the problem setting in the previous two steps.

Note

In order to perform a reliable Sobol analysis, the generated sample size is usually depending on the number of parameters evaluated and how complex the model is[22]. There is no, however, a general protocol to follow to determine the sample size. The rule is that the more parameters need to be evaluated and the more complex the model is, the more samples are required. For the Sobol sensitivity analysis of the trained solar energy prediction model, a sample size of 400,000 is used.

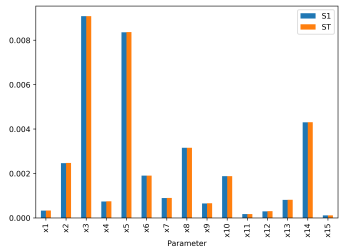
- Feed the generated samples in the trained model and get the model output.
- The output will be used to calculate the first-order and the total-order sensitivity indices.

In this case, the number of parameter in the analysis is 75, because all 75 weather variables are considered to be evaluated. Due to the standardization process in Eq.2.2, the range of all the parameters is between -1 and 1. The number of parameter sets generated is subject to $2n \times (p + 1)$, where n is the initial sample size, p is the number of parameters evaluated. Thus, the number of parameter sets generated for this particular model is 60,800,000.

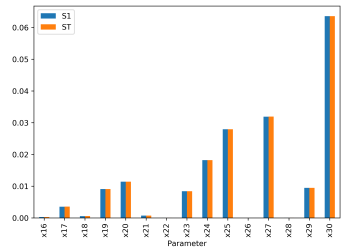
3.2 Sensitivity Analysis Results

The Sobol sensitivity analysis of the trained model is to identify the key weather variables that drive the changes in predicted daily solar energy. The Sobol sensitivity analysis results are presented and interpreted in the following section.

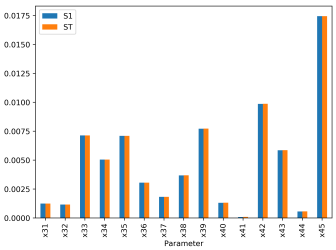
As mentioned in 3.1, the outputs of this Sobol sensitivity analysis are the first-order and the total-order sensitivity indices for all 75 weather variables, shown in figure 3.2. The x-axis in the figure represents 75 parameters evaluated written in $x1...x75$ due to convenience. For every 15 parameters from $x1$ to $x75$, they represent the aforementioned weather variables, respectively, and there are 5 sets of parameters in total. The y-axis is the magnitude of indices for both the first-order and the total-order. In Sobol sensitivity analysis, the parameters with sensitivity indices greater than 0.05 are considered significant. In the result of our analysis, in decreasing order, only $x65$, $x71$, $x73$, $x30$ and $x64$ are greater than 0.05 threshold. They represent, in order, the fifth report of the precipitable water over the entire depth of the atmosphere, the fifth report of the current temperature at 2m above the ground, the fifth report of the upward long-wave radiation at the surface, the second report of the upward short-wave radiation at the surface, and the fifth report of the air pressure at mean sea level. Notice that if we look at all the 75 variables together, most of the dominant variables are in the last weather data report. The precipitable over the entire depth of the atmosphere in the fifth report plays the most dominant part among all variables.



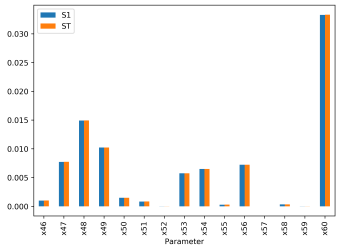
1 First set of weather data.



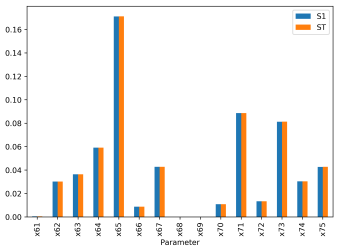
2 Second set of weather data.



3 Third set of weather data.



4 Fourth set of weather data.



5 Fifth set of weather data.

Figure 3.1. Sobol first-order and total-order sensitivity indices for 5 sets of weather data reported. 1. Sensitivity indices of variables in the first set of weather data. 2. Sensitivity indices of variables in the second set of weather data. 3. Sensitivity indices of variables in the third set of weather data. 4. Sensitivity indices of variables in the fourth set of weather data. 5. Sensitivity indices of variables in the fifth set of weather data.

The result of Sobol sensitivity indices of 5 times of weather data reports separately is shown in figure 3.1. For the first report of the day, the downward short wave radiative flux average at the surface, the precipitable water over the entire depth of the atmosphere, and the upward long-wave radiation on top of the atmosphere are the most three importance parameters. For the second report, the upward short-wave radiation at the surface, the temperature 2m above the ground, and the minimum temperature over the past 3 hours at 2m above the ground are the most significant ones. The third time report shows the similar situation, where the upward short-wave radiation at the surface is most dominant one, followed by the temperature at the surface, and the maximum temperature over the past 3 hours at 2m above the ground. For the fourth report, the upward short-wave radiation at the surface, the downward short-wave radiative flux average at the surface, and the air pressure at mean sea level take the lead. In the fifth time of weather data report, the precipitable water over the entire depth of the atmosphere, the current temperature at 2m above the ground, and the upward long-wave radiation at the surface are relatively more important.

3.2.1 Discussion

Based on the sensitivity analysis results, each of the 5 weather forecasting records of the day, the most important variable varies. This might have been caused by the fact that 5 times of records are fed into the model altogether with 75 input features. This might eliminate the effect of contribution of each individual record to the final output. For the first four weather forecasting records, the Sobol sensitivity indices are less than 0.05 threshold, indicating the first four records are much less important than the fifth one. According to the data description, the first record is the weather forecasting for 7 or 8 AM local time, and the following records are the weather forecasting of every 3 hours increment. Thus, the fifth record is supposed to

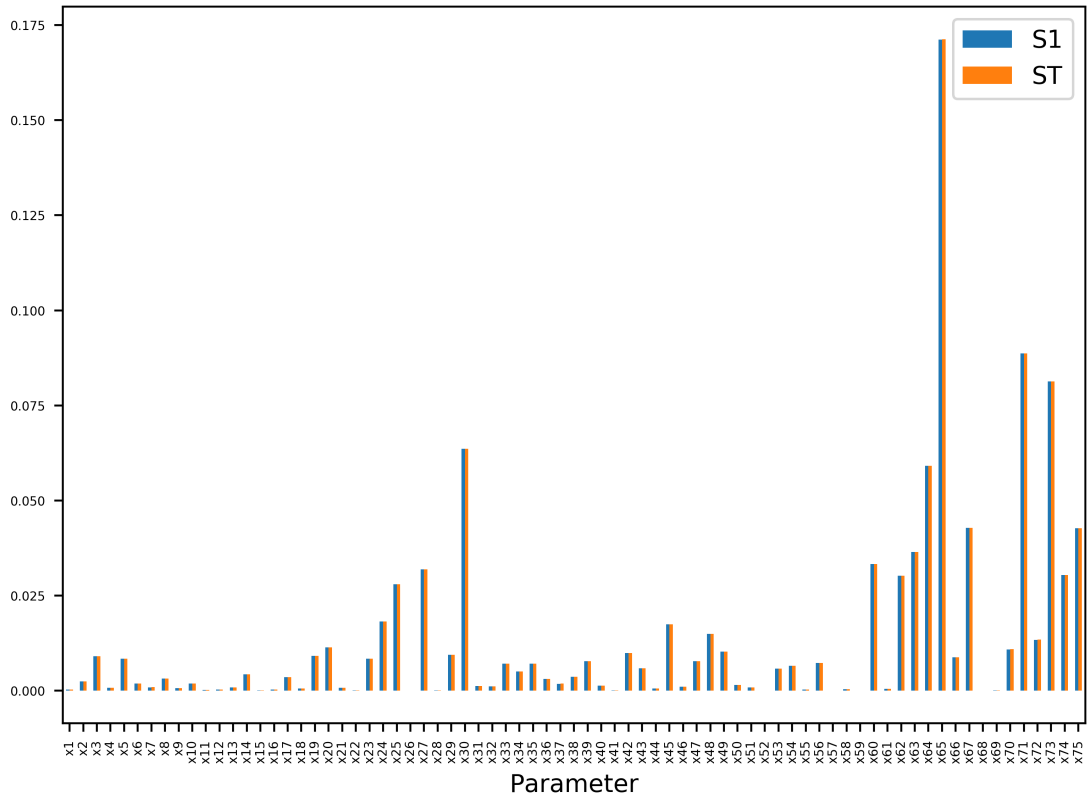


Figure 3.2. The first-order and the total-order Sobol sensitivity indices. "S1" means the first-order indices and "ST" means the total-order indice.

be the weather forecasting for 7 or 8 PM local time, yet having the most importance among all. The precise cause and underlying mechanism still need exploring.

3.2.2 Results with Bootstrap Confidence Interval

The estimate of sensitivity is useful only with an estimate of its sampling variability[1]. The Sobol sensitivity analysis indices are evaluated by bootstrap confidence interval. A 95% confidence level is applied in the process, so there will be an error bar for each index computed.

3.2.3 Bootstrap Confidence Interval

The bootstrap is based on the law of large numbers, which says that with enough data the empirical distribution will be a good approximation of the true distribution. In Sobol sensitivity analysis, the Monte Carlo sampled values are resampled with replacement many times, at each stage and for each parameter the Sobol sensitivity S_i is recalculated, leading to a bootstrap estimate of the sampling distribution of the sensitivity indices.

The basic idea of bootstrap is to use the statistics from the resampled data to approximate the statistics from the originally sampled data in order to have a well-approximated value of the variation of parameters of the true underlying distribution. For example, $x_1, ..., x_n$ is a data sample drawn from a distribution P, and u is a statistic computed from the sample. A resampling or an empirical distribution of the data P^* is obtained. The next step is to draw samples from the empirical distribution, say $x_1^*, ..., x_n^*$, and the corresponding statistic u^* can be calculated from the resample. In general, if we want to evaluated the variation of a statistics,such as the sample mean $\bar{x}$ computed from a sample, the underlying distribution needs to be known to find the distribution of the variation $\delta = \bar{x} - \mu$, where μ is the population mean of the underlying distribution. The bootstrap principle offers a practical approach to estimating the distribution of $\delta = \bar{x} - \mu$ when the underlying distribution is unknown. It says that we can approximate it by the distribution of

$$\delta^* = \bar{x} - \bar{x^*} \tag{3.1}$$

where $\bar{x^*}$ is the mean value of an empirical bootstrap sample. After obtaining many enough bootstrap samples, the critical value of δ^* based on the confidence level chosen, can be calculated by the corresponding percentiles.

In this Sobol sensitivity analysis, every sensitivity index computed has a bootstrap confidence interval with it. We can decide if we have chosen the appropriate number of evaluations based on the magnitude of the bootstrap confidence intervals. In general,

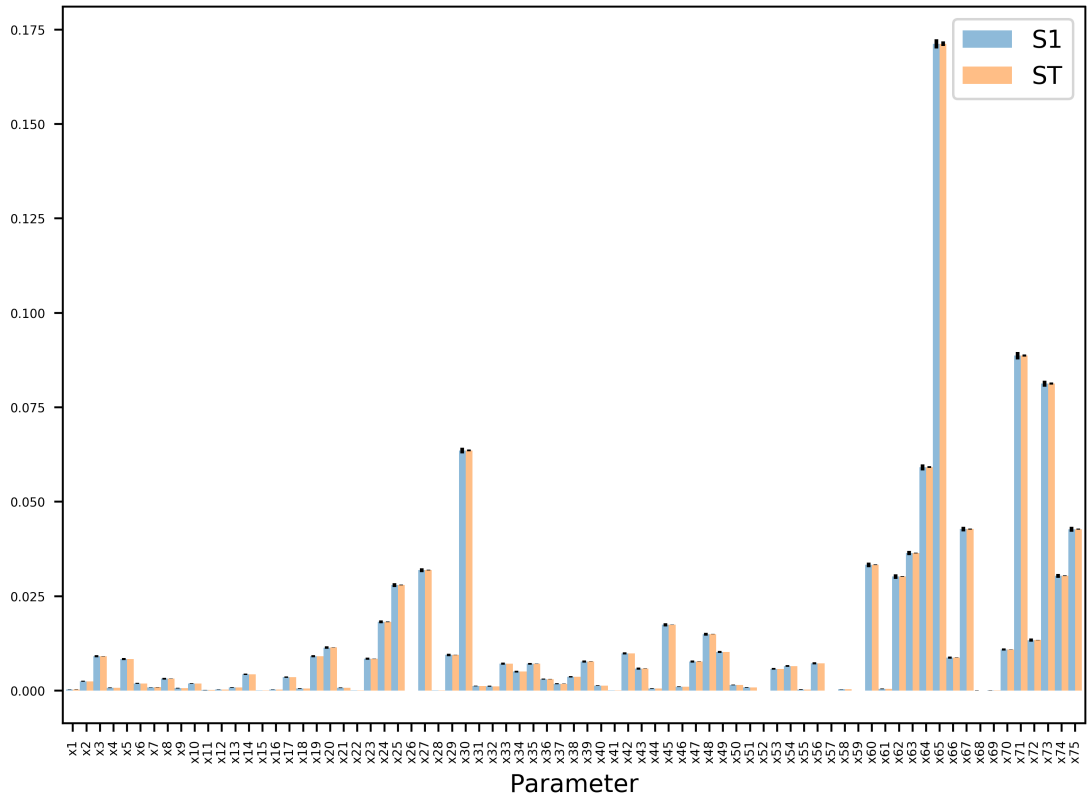


Figure 3.3. The first-order and the total-order Sobol sensitivity indices with 95% bootstrap confidence interval.

if the confidence interval of the dominant parameter is less than 10% of itself, we can say the initial sampling number is appropriately enough.

3.2.4 Results

The results of all 75 evaluated parameters are shown in figure 3.3. The error bars in the figure represent the bootstrap confidence intervals, which are all less than 10% of their corresponding Sobol sensitivity indices. Thus, we can say that we have chosen the proper number of the initial sample size and the Sobol sensitivity indices have well converged. We can also observe that the error bars for the first-order indices are generally greater than them of the total-order indices.

Table 3.1. Testing results after feature removal.

	Set 1	Set 2	Set 3	Set 4	Set 5
MAE	0.2164	0.2132	0.2283	0.2374	0.2277
Testing Accuracy	86.91%	87.10%	86.19%	85.63%	86.22%

3.3 Sobol sensitivity analysis verification

The analysis in the previous section indicates that the 5 most dominant or significant parameters among all the 75 evaluated parameters are $x65$, $x71$, $x73$, $x30$ and $x64$. In order to verify the reliability of the analysis results, the same trained model is evaluated after taking out the most significant parameters, compared with 5 less important parameters taken out. This process is repeated 5 times with different sets of less important parameters taken out each time. The results are shown in Table 3.1:

Meanwhile, the mean absolute error and the testing accuracy of taking out 5 most significant variables are 0.4193, and 74.63%, which indicates a much worse predictive ability than taking out less important features. The reliability of this Sobol sensitivity analysis is verified.

3.4 Discussion

According to the results of Sobol sensitivity analysis, the top most important variables contribute more to the model output variation. Most of the important variables coincide with the general knowledge on solar energy harnessing. The precipitable water over the entire depth of the atmosphere in this analysis contributes the most to the model output variation, possibly due to its strong correlation with the cloud distribution and refractive index of the atmosphere. The second significant variable is the current temperature at 2m above the ground The possible reason is that because

temperature reflects the effect of solar radiation, especially the temperature at some distance above the ground which indicates the amount of solar radiation is reflects. Among the top 5 most important variables, the effect of air pressure at mean sea level on harnessing solar energy needs further study and exploration. Evidently, based on the sensitivity analysis, the significant variables mostly come from the fifth run of the weather model of the day. The weather forecasting model runs start at 00 UTC, so they will always correspond to the same universal time. Thus, the fifth run of the weather model could have happened during the peak solar radiation period of time at Mesonet area.

4. SUMMARY

In this research, a fully connected feed forward neural network has been constructed and trained for daily solar energy prediction given weather forecasting model data. Due to the inconsistency of between the GEFS weather data points and the Mesonet measured solar energy points, a two-dimensional linear interpolation is employed to obtain the corresponding weather data for the Mesonet measurement points. The proposed model has a testing accuracy of 90.97% for daily solar energy forecasting, outperforming the linear regression baseline model of which the testing accuracy is 86.36%.

In order to explore the importance of the input weather variables, the Sobol sensitivity analysis is performed. The Sobol sensitivity indices, both the first-order and the total order, indicate that if we look at the 75 variables together, the top 5 important features are the fifth report of precipitable water over the entire atmosphere, the fifth report of the current temperature at 2m above the ground, the fifth report of upward short-wave radiation at the surface, the second report of upward short-wave radiation at the surface , and the fifth report of the air pressure at mean sea level. In each of the 5 weather data reports of a day, the most significant features are downward short-wave radiative flux average at the surface, the upward short-wave radiation at the surface, the upward short-wave radiation at the surface, the upward short-wave radiation at the surface, and the precipitable water over the entire atmosphere, respectively. The reliability of the Sobol analysis is verified by feature removal testing method. This research provides an accurate and efficient enough way to predict the daily solar energy only with weather forecasting model data. In addition, the importance of features have been studied, providing meaningful information for the choice of locations for solar power plants and their design.

REFERENCES

REFERENCES

[1] G. Archer, A. Saltelli, and I. Sobol. Sensitivity measures, anova-like techniques and the use of bootstrap. *Journal of Statistical Computation and Simulation*, 58(2):99–120, 1997.

[2] Y. Chu, M. Li, and C. F. Coimbra. Sun-tracking imaging system for intra-hour dni forecasts. *Renewable Energy*, 96:792–799, 2016.

[3] M. Fesanghary, E. Damangir, and I. Soleimani. Design optimization of shell and tube heat exchangers using global sensitivity analysis and harmony search algorithm. *Applied Thermal Engineering*, 29(5-6):1026–1031, 2009.

[4] E. Fock. Global sensitivity analysis approach for input selection and system identification purposes—a new framework for feedforward neural networks. *IEEE transactions on neural networks and learning systems*, 25(8):1484–1495, 2014.

[5] D. A. Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.

[6] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[7] D. Hamby. A review of techniques for parameter sensitivity analysis of environmental models. *Environmental monitoring and assessment*, 32(2):135–154, 1994.

[8] A. Kiparissides, S. Kucherenko, A. Mantalaris, and E. Pistikopoulos. Global sensitivity analysis challenges in biological systems modeling. *Industrial & Engineering Chemistry Research*, 48(15):7168–7180, 2009.

[9] P. A. Kowalski and M. Kusy. Determining the significance of features with the use of sobol method in probabilistic neural network classification tasks. In *Computer Science and Information Systems (FedCSIS), 2017 Federated Conference on*, pages 39–48. IEEE, 2017.

[10] P. A. Kowalski and M. Kusy. Sensitivity analysis for probabilistic neural network structure reduction. *IEEE transactions on neural networks and learning systems*, 2017.

[11] R. Kumar and L. Umanand. Estimation of global radiation using clearness index model for sizing photovoltaic system. *Renewable energy*, 30(15):2221–2233, 2005.

[12] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[13] A. Mellit and A. M. Pavan. A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected pv plant at trieste, italy. *Solar Energy*, 84(5):807–821, 2010.

[14] J. Nossent, P. Elsen, and W. Bauwens. Sobol'sensitivity analysis of a complex environmental model. *Environmental Modelling & Software*, 26(12):1515–1525, 2011.

[15] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga. Ensemble deep learning for regression and time series forecasting. In *Computational Intelligence in Ensemble Learning (CIEL), 2014 IEEE Symposium on*, pages 1–6. IEEE, 2014.

[16] A. Saltelli, K. Chan, E. M. Scott, et al. *Sensitivity analysis*, volume 1. Wiley New York, 2000.

[17] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[18] A. Saltelli, S. Tarantola, and K.-S. Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, 1999.

[19] M. J. Sanjari and H. Gooi. Probabilistic forecast of pv power generation based on higher order markov chain. *IEEE Transactions on Power Systems*, 32(4):2942–2952, 2017.

[20] A. Sfetsos and A. Coonick. Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques. *Solar Energy*, 68(2):169–178, 2000.

[21] I. M. Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001.

[22] X.-Y. Zhang, M. Trame, L. Lesko, and S. Schmidt. Sobol sensitivity analysis: a tool to guide the development and evaluation of systems pharmacology models. *CPT: pharmacometrics & systems pharmacology*, 4(2):69–79, 2015.