Scientific Research Publishing

Journal of Computer and Communications, 2019, 7, 31-36
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Linear Regression and Gradient Descent Method for Electricity Output Power Prediction

Yuanliang Liao

Tsinghua International School, Campus of Tsinghua High School, Beijing, China
Email: cutegirl2007@163.com

How to cite this paper: Liao, Y.L. (2019) Linear Regression and Gradient Descent Method for Electricity Output Power Prediction. *Journal of Computer and Communications*, **7**, 31-36.
https://doi.org/10.4236/jcc.2019.712004

Received: November 19, 2019
Accepted: December 13, 2019
Published: December 16, 2019

Abstract

Regulating the power output for a power plant as demand for electricity fluctuates throughout the day is important for both economic purpose and the safety of the generator. In this work, gradient descent method together with regularization is investigated to study the electricity output related to vacuum level and temperature in the turbine. Ninety percent of the data was used to train the regression parameters while the remaining ten percent was used for validation. Final results showed that 99% accuracy could be obtained with this method. This opens a new window for electricity output prediction for power plants.

Keywords

Machine Learning, Linear Algebra, Linear Regression, Gradient Descent, Error Analysis

1. Introduction

The power output of a power plant typically has a complicated relation with respect to the physical parameters, such as temperature, vacuum level, relative humidity, and exhaust steam pressure, etc. [1] [2] [3] [4] [5]. Attempts have been made to resolve the relations using methods such as Bagging algorithm [2], neural network [3], etc. However, either the algorithm itself is complicated or it involves other non-intuitive algorithms such as particle swarm optimization. A simple prediction method without consuming much computing resources is highly desired.

With the advent of computer science, specifically machine learning, methods have been established to build mathematical models based on training data to make predictions or decisions. These methods do not involve explicit programs

to perform the task. Using different mathematical algorithms, computers are able to make accurate predictions. Wide applications have been implemented in our daily life, for example, pattern recognition [6] [7], speech recognition [8] [9], text categorization [10] [11], autonomous driving [12] [13], medical diagnosis [14] [15], computational biology [16] [17], etc.

In machine learning, gradient descent is a very popular method for regression. It is an optimization algorithm used to find the values of coefficients of a function that minimizes a cost function. Gradient descent and cost function are methods and functions that help to analyze sets of data [18] [19]. By combining these two, it enables us to estimate values base on previous records. In this work, we developed a predictive model, that can predict output power of a power plant given the temperature and vacuum level. This is an attempt in using cost function and linear descent after courses of machine learning.

2. Methods

Obtaining the predicted value requires the theta of the linear equation. In order to validate the result, we use ninety percent of the data to predict the result and the remaining ten percent will be used to verify the validity of the predicted data. Graphing will reflect the range of the predicting value. As last, there will be calculation of rate of deviation, which is the percent of error comparing the predicted value and actual value.

Linear regression is a linear approach to modeling the relationship between a scalar response or dependent variable and one or more explanatory variables or independent variables. Simply, we have a set of data. Each dependent value corresponds to another independent variable. We may call these two represent dependent variable and independent variable; our purpose is to find the relationship between the dependent variable and the independent variables. Nevertheless, unlike any "pretty" functions we familiar the most, the variable does not have a direct relationship such as linear or exponential. This can be easily explained: the numbers are authentic data from real life, which means most numbers will not perfectly match to each other. Because there are many other uncertainties in real life, causing the change of dependent variable unstable and without pattern, people cannot use equation to explain the relationship of two variables. People can, however, plot the data to a graph and draw line of best fit. It looks easy when we usually directly get that by plugging data into excel, but the method of actually drawing the line is not so simple. The word "draw" is not very appropriate because it requires rigid calculations. Assume the linear function is expressed as the following.

$$h_\theta(x) = \theta_0 + \theta_1 x$$

We want it to fit the data we gathered. That means when we plug in x value (the size) the output of the function is closest to the real value in the dataset. There are several points, so we need a line that the average deviation from real value is the smallest. Therefore, we write the equation.

$$\sum_{i=1}^{m}\left[h_\theta\left(x^{(i)}\right)-y^{(i)}\right]^2$$

This simply means the sum of the difference of the estimate value to real value. In order to make it smaller, we take the derivative of the equation and then search for the critical point (1/2 m and square is added for simplifying the calculation process):

$$\begin{aligned}\frac{\partial}{\partial\theta_i}J\left(\theta_0,\theta_1\right)&=\frac{\partial}{\partial\theta_i}\left\{\frac{1}{2m}\sum_{i=1}^{m}\left[h_\theta\left(x^i\right)-y^{(i)}\right]^2\right\}\\&=\frac{\partial}{\partial\theta_i}\left\{\frac{1}{2m}\sum_{i=1}^{m}\left[\theta_0+\theta_1x^{(i)}-y^{(i)}\right]^2\right\}\end{aligned}$$

In addition, we need Gradient Descent to calculate the minimum of the function. Gradient Descent is a formula to find the minimum of a function,

$$\theta_j:=\theta_j-\frac{\partial}{\partial\theta_i}J\left(\theta_0,\theta_1\right),\ j=0,1$$

For the linear function, we can finally get the slope and the intersection point of the function.

$$\text{temp0}=\theta_0-\frac{\partial}{\partial\theta_i}J\left(\theta_0,\theta_1\right)$$

$$\text{temp1}=\theta_1-\frac{\partial}{\partial\theta_i}J\left(\theta_0,\theta_1\right)$$

$$\theta_0=\text{temp0}$$

$$\theta_1=\text{temp0}$$

3. Results

The code starts with loading given data file by csvread, which is used for reading excel data. The distribution of the power output with respect to vacuum level and temperature is shown in **Figure 1**. The absolute value of power output is reflected in its corresponding size and color during plotting. Then we declare the amount of the data we will be used to predict, which is 90% percent, and set the scale features to zero mean. The first step is used to set up the matrix that required in gradient descent formula. Part two is plugging in the matrix we just set up in step one. Gradient descent will provide the two largest constant (theta) of the linear function, which is an essential for prediction. The cost function is plotted in **Figure 2** versus number of iterations. It clearly shows that the cost function decreased by four order of magnitude with gradient descent algorithm. Then we use cross validation to find the predicted value. Basically, theta is the ratio of the current data to the next data. So we use the current data times theta to gain the prediction. Finally, we use the remaining ten percent data to find the error of the prediction in **Figure 3**. Ten percent stands for the actual data and we compare it with our prediction to find the error. Results show that we can get less than 1% average error with this method.

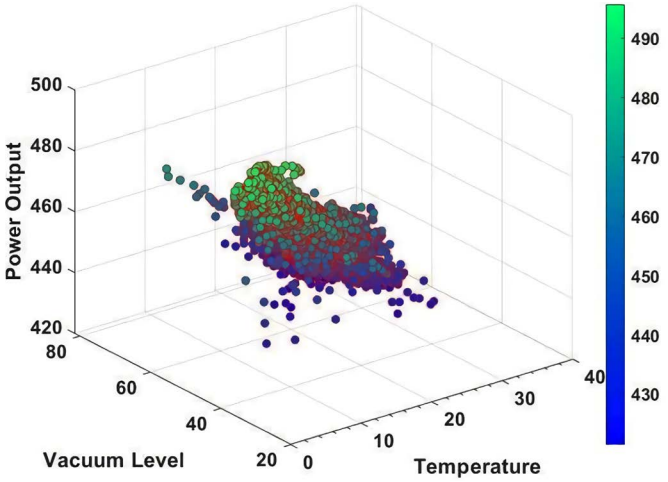

Figure 1. Power output versus vacuum level and temperature.

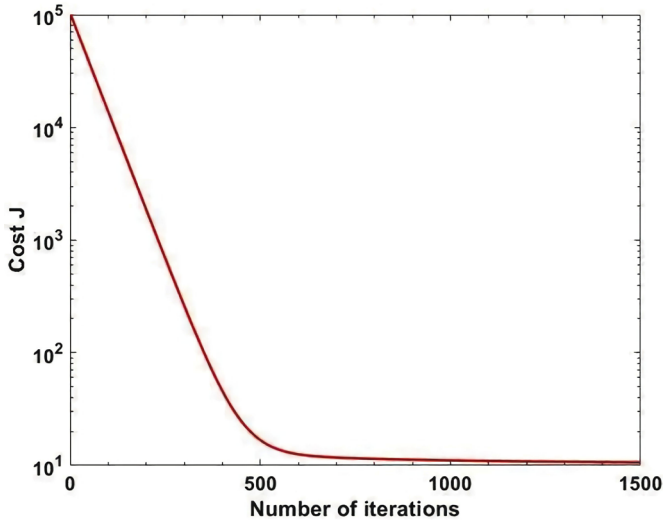

Figure 2. Cost function J decreases with number of iterations.

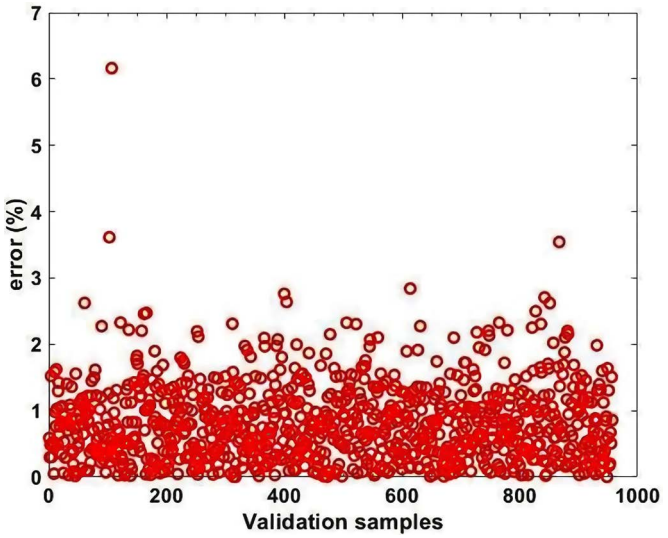

Figure 3. Validation with the 10% samples. Results show that less than 1% error is obtained.

The difficulty of the program is declaring the data and set the matrix. Because gradient descent and cross validation are only formula. The input of the formula comes from the matrix, which required some manipulation to make the input fit the requirement of the formulas.

4. Conclusion

In this paper, we employed gradient descent method combined with cost function to predict the power output based on the input of vacuum level and temperature in a power plant. Less than 1% prediction error has been achieved. Although this is a preliminary study, with more complicated gradient method by incorporating more physical parameters, more accurate results could be anticipated. Moreover, we believe this method could be extended to other areas.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] El-Sharkh, M.Y., *et al.* (2004) A Dynamic Model for a Stand-Alone PEM Fuel Cell Power Plant for Residential Applications. *Journal of Power Sources*, **138**, 199-204. https://doi.org/10.1016/j.jpowsour.2004.06.037

[2] Tüfekci, P. (2014) Prediction of Full Load Electrical Power Output of a Base Load Operated Combined Cycle Power Plant Using Machine Learning Methods. *International Journal of Electrical Power & Energy Systems*, **60**, 126-140. https://doi.org/10.1016/j.ijepes.2014.02.027

[3] Quan, H., Srinivasan, D. and Khosravi, A. (2013) Short-Term Load and Wind Power Forecasting Using Neural Network-Based Prediction Intervals. *IEEE Transactions on Neural Networks and Learning Systems*, **25**, 303-315. https://doi.org/10.1109/TNNLS.2013.2276053

[4] Na, M.G., Jung, D.W., Shin, S.H., Jang, J.W., Lee, K.B. and Lee, Y.J. (2005) A Model Predictive Controller for Load-Following Operation of PWR Reactors. *IEEE Transactions on Nuclear Science*, **52**, 1009-1020. https://doi.org/10.1109/TNS.2005.852651

[5] Liu, J., Fang, W.L., Zhang, X.D. and Yang, C.X. (2015) An Improved Photovoltaic Power Forecasting Model with the Assistance of Aerosol Index Data. *IEEE Transactions on Sustainable Energy*, **6**, 434-442. https://doi.org/10.1109/TSTE.2014.2381224

[6] Weiss, S.M., Kapouleas, I. and Shavlik, J.W. (1990) An Empirical Comparison of Pattern Recognition, Neural Nets and Machine Learning Classification Methods. In: Shavlik, J.W. and Dietterich, T.G., Eds., *Readings in Machine Learning*, Morgan Kaufmann Publishers, San Mateo, 177-183.

[7] Riesen, K. and Bunke, H. (2008) IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. In: *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition* (*SPR*) *and Structural and Syntactic Pattern Recognition* (*SSPR*), Springer, Berlin, Heidelberg, 287-297. https://doi.org/10.1007/978-3-540-89689-0_33

[8] Deng, L. and Li, X. (2013) Machine Learning Paradigms for Speech Recognition: An

Overview. *IEEE Transactions on Audio, Speech, and Language Processing*, **21**, 1060-1089. https://doi.org/10.1109/TASL.2013.2244083

[9] Liu, Y., *et al.* (2006) A Study in Machine Learning from Imbalanced Data for Sentence Boundary Detection in Speech. *Computer Speech & Language*, **20**, 468-494. https://doi.org/10.1016/j.csl.2005.06.002

[10] Sebastiani, F. (2002) Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, **34**, 1-47. https://doi.org/10.1145/505282.505283

[11] Joachims, T. (1998) Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: *European Conference on Machine Learning*, Springer, Berlin, Heidelberg, 137-142. https://doi.org/10.1007/BFb0026683

[12] Dogan, Ü., Edelbrunner, J. and Iossifidis, I. (2011) Autonomous Driving: A Comparison of Machine Learning Techniques by Means of the Prediction of Lane Change Behavior. *IEEE International Conference on Robotics and Biomimetics*, Phuket Island, 7-11 December 2011, 1837-1843. https://doi.org/10.1109/ROBIO.2011.6181557

[13] Sallab, A.E.L., *et al.* (2017) Deep Reinforcement Learning Framework for Autonomous Driving. *Electronic Imaging*, **19**, 70-76. https://doi.org/10.2352/ISSN.2470-1173.2017.19.AVM-023

[14] Foster, K.R., Koprowski, R. and Skufca, J.D. (2014) Machine Learning, Medical Diagnosis, and Biomedical Engineering Research-Commentary. *Biomedical Engineering Online*, **13**, 94. https://doi.org/10.1186/1475-925X-13-94

[15] Kononenko, I. (2001) Machine Learning for Medical Diagnosis: History, State of the Art and Perspective. *Artificial Intelligence in Medicine*, **23**, 89-109. https://doi.org/10.1016/S0933-3657(01)00077-X

[16] Ben-Hur, A., *et al.* (2008) Support Vector Machines and Kernels for Computational Biology. *PLoS Computational Biology*, 4, e1000173. https://doi.org/10.1371/journal.pcbi.1000173

[17] Angermueller, C., *et al.* (2016) Deep Learning for Computational Biology. *Molecular Systems Biology*, **12**, 878. https://doi.org/10.15252/msb.20156651

[18] Zinkevich, M., *et al.* (2010) Parallelized Stochastic Gradient Descent. *Advances in Neural Information Processing Systems*, **23**, 2595-2603.

[19] Andrychowicz, M., *et al.* (2016) Learning to Learn by Gradient Descent by Gradient Descent. *Proceedings of the* 30*th International Conference on Neural Information Processing Systems*, Barcelona, 5-10 December 2016, 3988-3996.