Short-Term Forecasting of Wind Power Generation by Adaptive Neural Networks

Hong Li
New York City Colleg e of Technology of City University of New York
hli@citytech.cuny.edu

Ali Setoodehnia
ECPI University
asetoodehnia@ecpi.edu

Abstract

This paper presents a method of short term forecasting of wind power generation using Feed Forward Multilayer Perceptron (FMP) with an adaptive learning algorithm. Wind power, as an alternative to burning fossil fuels, is clean, renewable and widely distributed in conjunction with other electric power sources to give reliable supply. It is important to have accurate forecasting for power management. A neural network model is designed with learning algorithm following analysis of convergence of learning process based on the Backpropagation algorithm. The analysis leads to conditions of learning factors to guarantee the convergence. The conditions are further extended to a feasible formula that defines an adaptive learning factor at iteration of learning process. The result of simulations using wind power data from European countries sourced from Open Power System platform are presented.

Introduction

Wind power has been used as long as humans put sails into the wind. As an alternative to burning fossil fuels, it uses air flow through wind turbines to mechanical power generators for electricity. Wind power is renewable, produces no greenhouse gas emissions during operation, uses no water, and uses little lands. The effects on the environment are far less problematic than those of nonrenewable power sources. Worldwide there are now thousands of wind turbines operating. World wind generation capacity more than quadrupled between 2000 and 2006, doubling about every three years. Wind power capacity has expanded rapidly and wind energy production was around 4% of total worldwide electricity usage, and growing rapidly. Rapid growth in wind power, as well as increase on wind generation, requires serious research in various fields. Because wind power is weather dependent, it is variable and intermittent over various time-scales. Thus accurate forecasting of wind power is recognized as a major contribution for reliable large-scale wind power integration. Wind power forecasting methods can be used to plan unit commitment, scheduling and dispatch by system operators, and maximize profit by electricity traders. Numbers of wind power forecasting models were used in many research [1]. The Artificial Neural Networks (ANNs) have been proven in many real world applications to be useful in various tasks of modeling nonlinear systems, such as signal processing, pattern recognition, optimization, weather forecasting, to name a few. It has drawn many researchers in power generation forecasting.

The ANN approach was applied to provide short-term wind power forecasting based on historical wind power data from Portugal [2]. Research [3] proposed a new fuzzy-based cost function with the purpose of having more freedom and flexibility in adjusting NN parameters used for construction of PIs. In comparison with the other cost functions in the literature, this new formulation allows the decision-makers to apply their preferences for satisfying the PI coverage probability and PI normalized average width individually.

The ANN is a set of processing elements (neurons or perceptrons) with a specific topology of weighted interconnections between these elements and a learning law for updating the weights of interconnection between two neurons. The FMP networks have been shown to obtain successful results in system identification and control [4]. The Lyapunov theorem was used to provide stability analysis of Backpropagation training algorithm of such network in [5-7]. However, the training process can be very sensitive to initial condition such as number of neurons, number of layers, and value of weights, and learning factors which are often chosen by trial and error. This paper presents a detailed analysis of the FMP architecture and its stability. The Backpropgation algorithm is used for learning – that is, weight adjusting. The Least Square error function is defined and verified satisfying the Lyapunov condition which guarantees the stability of the system. In the work [8-9], the analysis carries out a method that defines a range of learning factor. Selecting the learning factor within the range at each iteration ensures the condition for stability is satisfied. This is a simple back-propagation network of three layers, and it is trained and tested on a high volume of historical wind power generation data. The challenge here is not in the network architecture itself, but instead in the choice of variables and the information used for training. In this research, historical data were chosen from previous every 3 hours of 24 hours. In simulation, instead of selecting a learning factor by trial and error, author defines an adaptive learning factor which satisfies the convergence condition and adjust connection weight accordingly. The simulation results are presented to demonstrate the performance.

Basic Principle of FMP

A system identification problem can be outlined as follow: a set of data is collected from the system including input data and corresponding output data observed, or measured as target output of the identification problem. The set is often called “training set”. A neural network model with parameters, called weights, is designed to simulate the system. When the output from neural network are calculated, an error representing the difference between target output and calculated output from the system is generated. The learning process of neural network is to modify the network to minimize the error.

Consider a system with N inputs $X = \{X_1, \dots, X_N\}$ and M output units Y =$\{Y_1, \dots , Y_M\}$. A recurrent FMP network combines number of neurons, called nodes, feed forward to next layer of nodes. Suppose N_l is number of nodes in lth layer, each output from the l-1th layer will be used as input for next layer. A system of a single layer with M outputs can be expressed in form of

$$Y_{jp} = f(Z) = f\left(\sum_{i=1}^{N} w_{ij} X_{ip} + \sum_{i=1}^{D} v_{ij} Y_{j(p-i)}\right) \qquad (1)$$

where w_{ij} is called connection weight from input X_i to output Y_j;vij is called connection weight of local feedback at jth node with ith delay; $f(\,)$ is a nonlinear sigmoid function

$$f(Z) = \frac{1- e^{-\theta Z}}{1+ e^{-\theta Z}} \tag{2}$$

with constant coeficient θ, called slope; p = 1, …, T, T is number of patterns, D is number of delay used in local feedback.

The Backpropagation algorithm has become a common algorithm used for training feed-forward multilayer perceptron. It is a generalized the Least Mean Square algorithm that minimizes the mean squared error between the target output and the network output with respect to the connection weights. The algorithm looks for the minimum of the error function in weight space using the method of gradient descent. The combination of weights which minimizes the error function is considered to be a solution of the learning problem. A proof of the Backprpagation algorithm was presented in [10] based on a graphical approach.

The total error E of the network training set is defined as

$$E = \frac{1}{T}\sum_{k=1}^{N_L}\sum_{p=1}^{T} e_k^2(p) \tag{3}$$

where $e_k^2(p)$ is the error associated with *pth* pattern at the *kth* node of output layer,

$$e_k^2(p) = (\, d_k(p) - Y_k^L(p))^2 \tag{4}$$

where $d_k(p)$ is the target at kth node and $Y_k^L(p)$ is the output of network at the kth node. The learning rule is chosen following gradient descent method to update the network connection weights iteratively,

$$\Delta W_j = -\mu\frac{\partial E}{\partial W_j}; j = 1, \dots, M \tag{5}$$

$$\Delta \nu_j = -\mu\frac{\partial E}{\partial \nu_j}; j = 1, \dots, D \tag{6}$$

where $W_j = (w_{1j}, \dots, w_{Nj})$ *and* $\nu_j = (\nu_{j1}, \dots, \nu_{Dj})$ are weight vectors in jth node; μ is a constant called learning factor.

Adaptive Learning Factor

There are no inclusive general concepts of stability for nonlinear systems. The behavior of a system may depend drastically on inputs and disturbances. However, Lyapunov theory has been used in many researches to examine the stability of nonlinear systems.

The definition of the Lyapunov function and the Lyapunov theorem are quoted below [11]:

Definition (Lyapunov function): A scalar function V(x) is a Lyapunov function for the system

$$x(t+1) = f\big(\, x(t)\big),\; f(0) = 0 \tag{7}$$

if the following conditions hold:

1. $V(0) = 0$ and $V(0)$ is continuous in x
2. $V(x)$ is positive definite, that is, $V(x) \geq 0$ with $V(x) = 0$ only if $x = 0$
3. $\Delta\, V(x) = V(f\big(x(t)\big) - V(x(t))$ is negative definite, that is, $V(f\big(x(t)\big) - V(x(t)) \leq 0$ with $\Delta V(x) = 0$ only if $x = 0$;

Theorem 1 (Lyapunov Theorem): The solution $x(t) = 0$ for the system given by (7) is asymptotically stable if there exists a Lyapunov function of the system in x.

The stability of the learning process in an identification approach leads to better modeling and a convergent process. According to the Lyapunov theorem, determination of stability depends on selection and verification of a positive definite function. For the systems defined in (1) – (2), assume that the Backpropagation learning rule is applied and the error function and weights updating rule are defined in (5) - (6), then define

$$V(t) = \frac{1}{N_L T} \sum_{j=1}^{N_L} \sum_{p=1}^{T} e_k^2(t) \tag{8}$$

The following theorem was proved in [8] that the V (t) satisfies the Lyapunov conditions.

Theorem 2: Assume that the system with one hidden layer can be represented in the form of:

$$Y_{jp} = f\big(\, Z_{jp}^1\big) = f\big(\, \sum_{i=1}^{N} w_{ij}^o X_{ip}^1 + \sum_{i=1}^{D} v_i^o Y_{j(p-i)}\,\big) \tag{9}$$

$$X_{jp}^1 = f\big(Z_{ip}\big) = f\big(\, \sum_{i=1}^{N} w_{ij}^h X_{ip} + \sum_{i=1}^{D} v_i^h X_{j(p-i)}^1\big) \tag{10}$$

the gradient descent rule

$$\Delta W_o^j = -\mu \frac{\partial E}{\partial W_o^j},\;\; j = 1, \ldots, M \tag{11}$$

$$\Delta v_o^j = -\mu \frac{\partial E}{\partial v_o^j},\;\; j = 1, \ldots, D \tag{12}$$

$$\Delta W_h^j = -\mu \frac{\partial E}{\partial W_h^j},\;\; j = 1, \ldots, M \tag{13}$$

$$\Delta v_h^j = -\mu \frac{\partial E}{\partial v_h^j},\;\; j = 1, \ldots, D \tag{14}$$

Where

$$W_o^j = (w_{o1}^j, \ldots, w_{oN}^j)^T,\;\; v_o^j = (v_{o1}^j, \ldots, v_{op}^j)^T$$
$$W_h^j = (w_{h1}^j, \ldots, w_{hH}^j)^T,\;\; v_h^j = (v_{h1}^j, \ldots, v_{hD}^j)^T$$

are weight vectors in jth node in output layer and hidden layer respectively. The system is stable when the learning factor in (11) - (14) satisfies the condition given below:

$$\mu < \frac{TM}{2\sum_{j=1}^{M}\sum_{p=1}^{T}\left[\left\|\frac{\partial Y_{pj}}{\partial W_o^j}\right\|^2+\left\|\frac{\partial Y_{pj}}{\partial v_o^j}\right\|^2+\left\|\frac{\partial Y_{pj}}{\partial W_h^j}\right\|^2+\left\|\frac{\partial Y_{pj}}{\partial v_h^j}\right\|^2\right]} \quad (15)$$

Proof: given in [8].

In simulation, the learning factor was generally predefined constant whose value was selected by trial and error. The simulation performance differs from different values of learning factor. The learning process may converge or may not reach a satisfactory threshold with various learning factors. From the above theorem, convergence is guaranteed if an adaptive learning factor is selected at each iteration satisfying the stability condition. For purpose of simplifying the simulation, instead of calculating all $\frac{\partial Y_{pj}}{\partial W^l}$ and $\frac{\partial Y_{pj}}{\partial v^l}$ for l = 1, … L; j = 1, … N_L, the following corollary provides an extended condition with more restriction but less calculation.

Consider infinite norm notation for any vector $X = \{x_1, x_2, \dots, x_n\}$ that $\|X\|_\infty = \max_{1<i<n}\{x_i\}$, for simplicity, use notation $\|*\|$ in this paper representing $\|X\|_\infty$). Applying infinite norm in (15) and notation $\left|v_j^l\right| = \sum_{d=1}^{Dv} v_{jd}^l$, extended conditions can be driven as follow:

Corollary 1: The system defined in (9) – (10) converges if the learning factor in (11) – (14) satisfies the following conditions:

$$2 - \theta\,|v^o| > 0\,, \quad (16)$$

$$\mu < \frac{\left(2-\theta\left|v_j^o\right|\right)^2}{4\theta(2+\theta(|w^o|-\left|v_j^o\right|} \quad (17)$$

Simulation

The changing energy landscape requires rigorous analysis to support robust investment and policy decisions. Power systems are complex, hence researchers and analysts often rely on large numerical computer models for a variety of purposes, ranging from price projections to policy advice and system planning. Such models include unit commitment, dispatch, and generation expansion models. These models require a large amount of input data, such as information about existing power stations, interconnector capacity, and yearly electricity consumption, and ancillary service requirements, but also (hourly) time series of load, wind and solar power generation, and heat demand. Fortunately, most of these data are publicly available from sources such as transmission system operators, regulators, or industry associations. The Open Power System Data platforms provide free and open data of the European power system with restricted use for non-commercial applications. The Open Power System Data is implemented by four institutions, DIW Berlin, Europa-Universität Flensburg, Technical University of Berlin, and Neon Neue Energieökonomik and funded by the German Federal Ministry for Economic Affairs and Energy. The simulations presented here used a data package which contains time-series data relevant for power system modeling. The data include hourly electricity consumption for 36 European countries, wind and solar power generation from German transmission system operators. The German wind power generation data available from 2005- 2015 are used for ANN training and mainly demonstrating that the enhanced learning algorithm may avoid many trial and errors for selection of learning factors.

In general, the initial weights of a neural network model are randomly selected, a learning factor is predefined. The performance of the learning can sometime very volatile due to the selection of the learning factor. To find the optimal fit, the trial and error is common practice to run the simulation with different values of learning factor. In this research, an upper boundary of learning factors (17) is derived from the theory of convergence. At iteration of network training, the norm of weights is calculated and a learning factor is defined to satisfy the convergence condition (17).

A three layer neural network structure was selected with 8 inputs, 6 and 4 nodes in the hidden layer, and one outputs. For hourly wind power generation as output, 8 inputs are wind power generation of every three hours in previous 24 hours period. Data from 2013 and 2014 are used to train the ANN model. The data from 2015 are used to test the ANN model performance. Input and output data are normalized to range from 0 to 1. After the ANN network is trained, the forecast of wind power generation is calculated using the ANN network and then denormalized compared with the actual wind power generation in 2015.

With the constant learning factor, two values were used for the learning trials: 0.05 and 0.1. After several attempts, with slope set as 0.7, learning factor set as constant 0.05, momentum term set as 0.1, and random generated initial weights, the system reached to absolute error 0.028 after 100000 iterations. The error behaviors are shown respectively in Figure 1 and Figure 2.

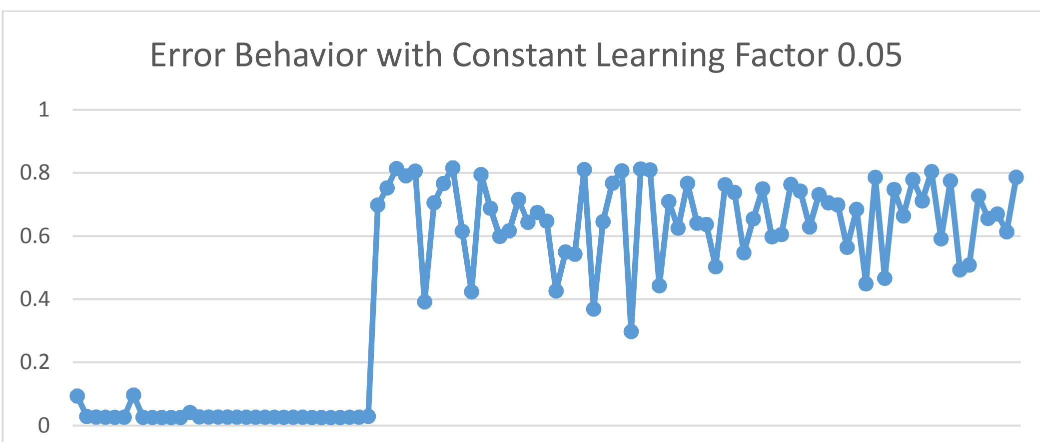

Figure 1. Error Behavior of Training with Constant Learning Factor 0.05

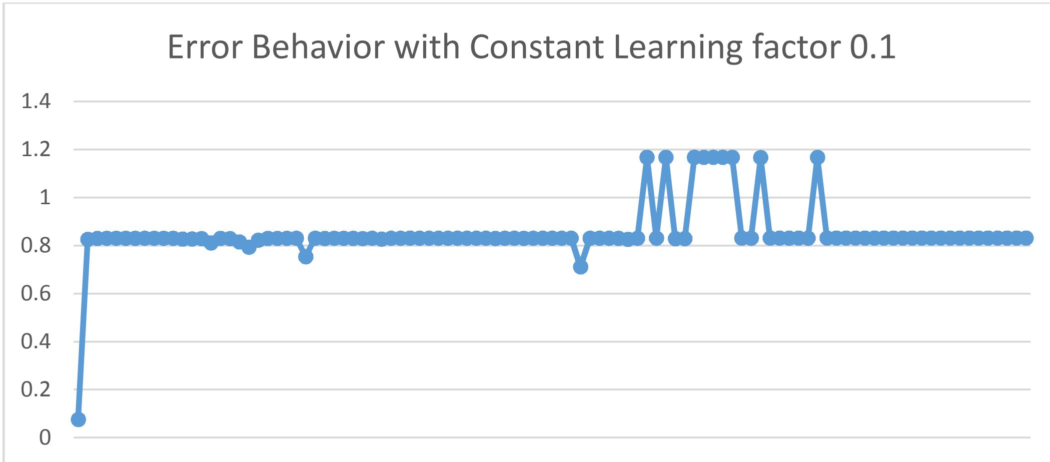

Figure 2. Error Behavior of Training with Constant Learning Factor 0.1

Applying the same values of slope, 0.7 and momentum, 0.1 with initial learning factor 0.1 and 0.05, the error of the neural network learning steadily decreases when an adaptive learning factor was applied at each iteration, Figure 3 and Figure 4 demonstrate the error behaviors of learning with initial learning factors, 0.05 and 0.1 respectively. It is observed that error behaviors do not differ with different initial values of learning factors. However, the constant learning factor can cause volatile performance of training.

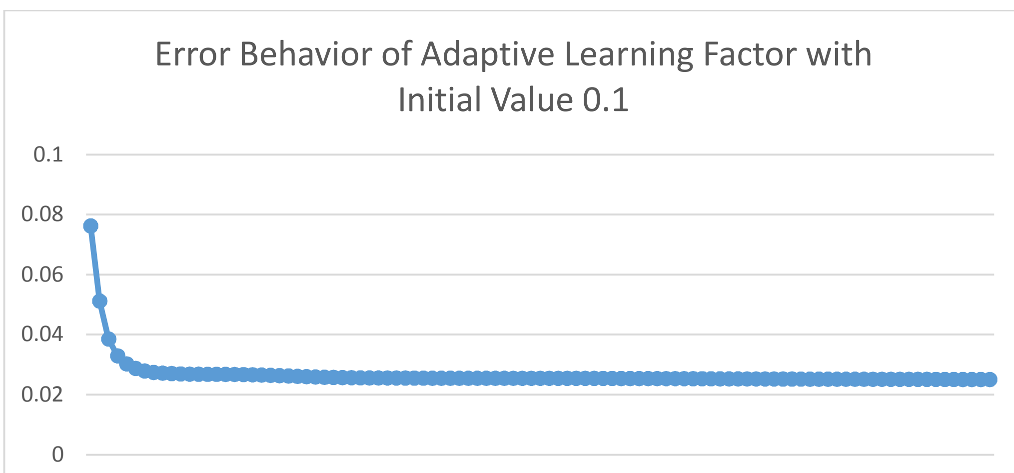

Figure 3. Error Behavior of Training with Adaptive Learning Factor with initial 0.1

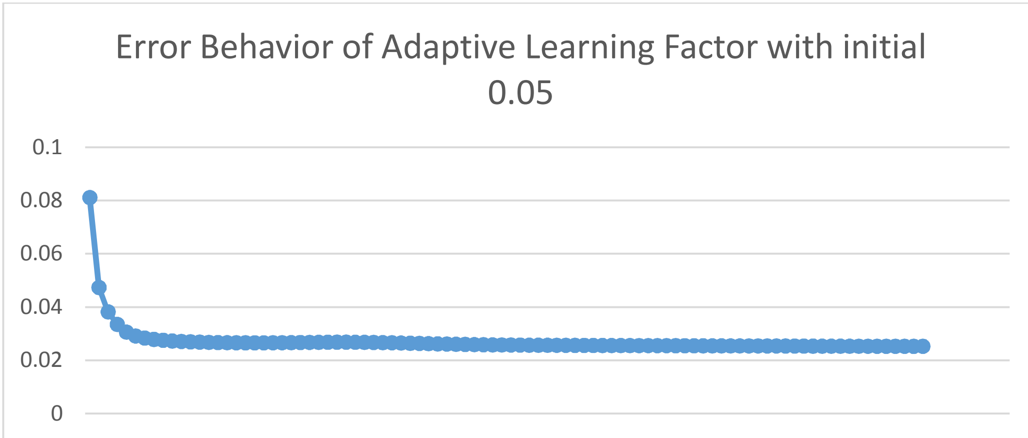

Figure 4. Error Behavior of Training with adaptive Learning Factor with initial 0.05

The Figure 5 – Figure 10 present the wind power forecasting for year 2015 from the neural network trained with slope 0.7, momentum 0.1, and adaptive learning factor starting with 0.05.

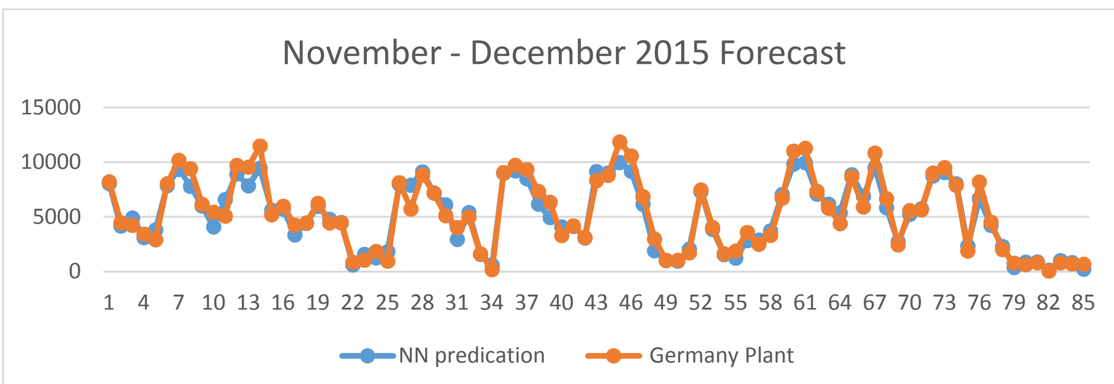

Figure 5. NN Predication and German Plant for November – December 2015

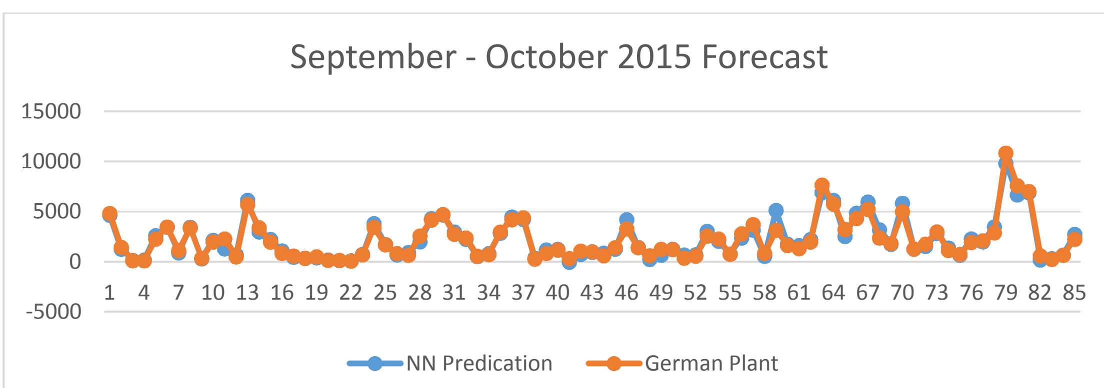

Figure 6. NN Predication and German Plant for September – October 2015

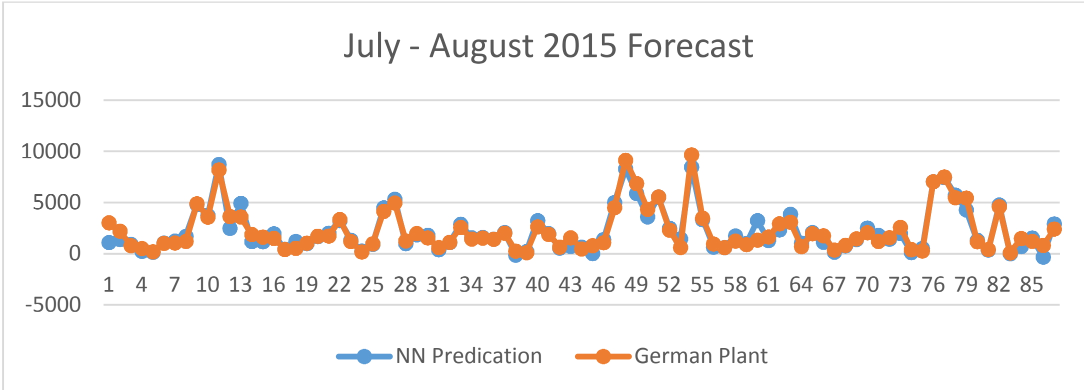

Figure 7. NN Predication and German Plant for July – August 2015

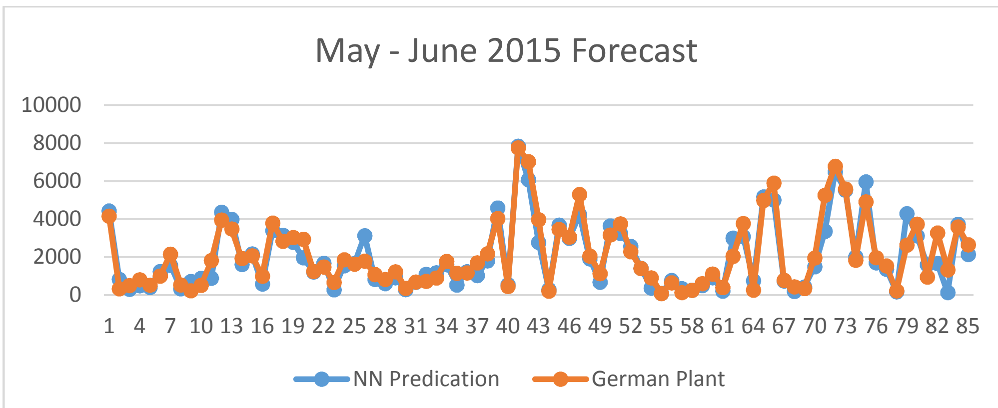

Figure 8. NN Predication and German Plant for May – June 2015

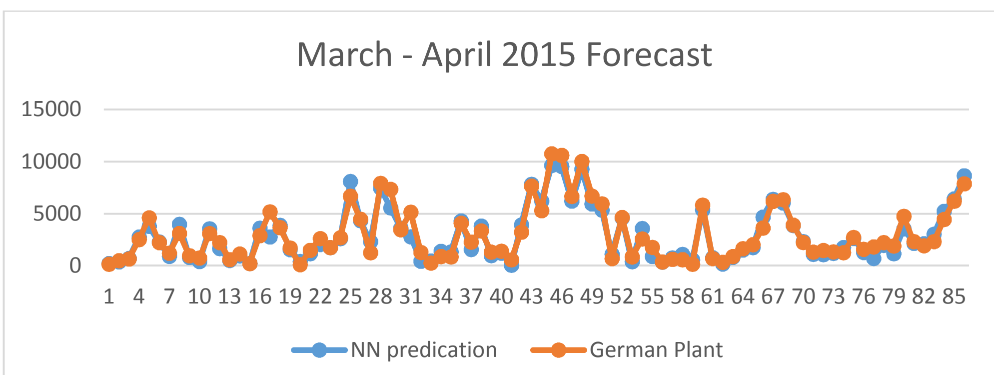

Figure 9. NN Predication and German Plant for March-April 2015

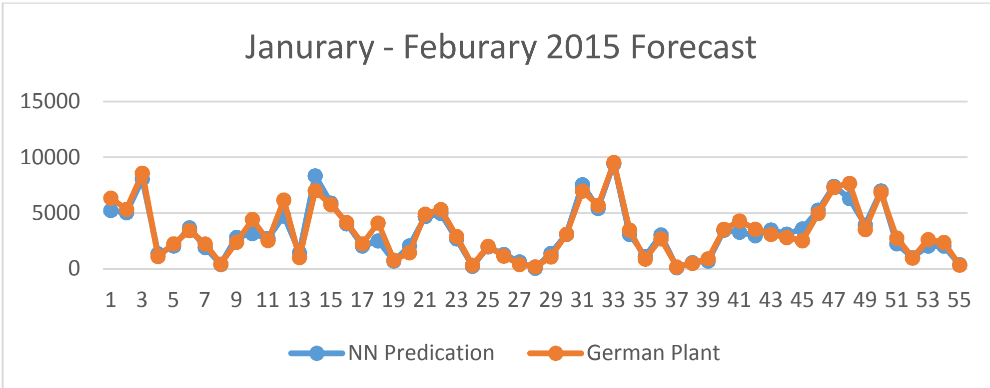

Figure 10. NN Predication and German Plant for January – February 2015

Conclusion

This research focuses on wind power generation forecasting using neural network. The historical data of time-series wind power generation are sourced from Open Power System platform. The data are available from 2005, hourly wind power generation. An adaptive learning factor are defined at iteration of training following the analysis of convergence theory. The analysis results in a condition which provides an upper boundary of the learning factor. Furthermore, a more simplified condition was used to provide a feasible implementation of the adaptive learning factor. An adaptive learning factor derived from analysis of stability is selected at each iteration of learning process satisfying convergence condition to avoid unstable phenomena. The simulation result is based on the data of German wind power plant. The error behaviors were demonstrated for training with an adaptive learning factor as well as with a selected constant learning factor. The comparison demonstrated that an arbitrarily chosen learning factor leads to an unstable identification of the considered system; however, an adaptive learning factor satisfying the conditions ensures the stability of the identification system. The neural network is trained with historical data from 2013 – 2014 and tested for forecasting performance with data from 2015.

References

[1] Wang, X. Guo, P. and Huang, X.(2011). A review of wind power forecasting models, Energy Procedia Elsevier, 12(2011) 770-778.

[2] Catalao, J., Pousinho, H. and Medes, V.,(2009) An artificial neural network approach for short term wind power forecasting in Portugal, Engineering intelligent system1:5-11 2009.

[3] Kavousi-Fard, A., Khosravi, A. and Nahavandi, S., (2016) *A New Fuzzy-Based Combined Prediction Interval for Wind Power Forecasting*, IEEE Transcation on Power Systems, Volume: 31, Issue: 1, pp 18-26.

[4] Korkobi, T., Djemel, M. & Ctourou, M. (2008). Stability Analysis of Neural Netorks-based System Identification. Modeling and Simulation in Engineering, Volume 2008, Article ID 343940

[5] Akea, H., Alassar,R., & Covachev, V. (2005). Stability of Neural Networks with Time Varying Delays in the Presence Impulses. Advances in Dynamical Systems and Application,. 1(1), 1-15.
[6] Choon Ki Ahn, H. (2012) Stability Conditions for Fuzzy Neural Networks, Advances in Fuzzy Systems Volumn Articale ID 281821
[7] Liang, G. (2010). Global Asymptotically Stability of Cellular Neural Networks with Time-varying Delay, 2010 8th World Congress on Intelligent Control and Automation , (pp 5031-5036)
[8] Hong Li (2014). Adaptive Learning Factor of Backpropagation in Feed Forward Neural Networks, International Journal of Modern Engineering, Vol. 14, No 2, Spring/Summer 2014, pp 47 – 53.
[9] Hong Li and Ali Setoodehnia (2014). Convergence Analysis of Adaptive Recurrent Neural Network, International Journal of Engineering Research and Applications (IJERA), Vol. 4, Issue 6(Version 1), pp.48-53.
[10] R. Rojas, *Neural Networks*, Springer-Verlag, 1996
[11] Sontag, E.D., *Mathematical Control Theory*, Springer-Verlag, New York

Biographies

HONG LI is an Associate Professor and Chairperson of Computer Systems Technology at New York City College of Technology of the University of New York. She earned her Ph.D. in Mathematics from the University of Oklahoma. Her research interests are mathematical modeling, applications for artificial neural networks, and software development. Dr. Li may be reached at hli@citytech.cuny.edu.

ALI SETOODEHNIA is faculty of Electrical Engineering Technology at ECPI University. He earned his Ph.D. in Electrical Engineering from the University of Oklahoma. Dr. Setoodehnia's research interests are artificial neural networks, power system, and system identification. Dr. Setoodehnia may be reached at asetoodehnia@ecpi.edu.