Neural Networks for Wind Speed Prediction Along the Oregon-Washington Border

Teal Hobson-Lowther
Colorado School of Mines
December 10, 2015

Abstract—**Neural networks are evaluated for wind speed prediction. A summary of previous work in the field is given. The performance of our networks are compared across two different locations along the Washington-Oregon border. Forecasting capabilities of trained neural networks are determined for intervals of thirty minutes, one hour, and one day for each location. Optimal network architectures are determined experimentally. A discussion of the results ensues.**

Index Terms—**Biddle Butte, Butler Grade, machine learning, non-linear autoregressive neural networks, Oregon, renewable energy, time-series prediction, Washington, wind prediction**

I. Introduction

As we progress into a time where global climate change threatens to alter the fate of the earth and the human race along with it, the need for reliable forms of renewable energy becomes imperative. Of the several forms of renewable energies being explored and exploited by our species, solar, wind, and hydro-electric are at the forefront. These show the most promise as economically viable solutions for alternative energies. Because of their relative novelty, the large-scale development of grid-incorporated power plants is still an active area of research. Each has their own drawbacks and limitations with respect to grid integration.

This paper will focus on wind-energy in particular. The main drawback of wind energy generation lies in the lack of predictability of the resource. Wind follows very distinct daily and yearly trends. To be more precise, the wind resource is generally higher during the night hours, and also tends to be higher during the winter months. The wind resource, as a rule, is modeled as a statistical distribution called the Weibull. The parameters of the Weibull that properly model the wind vary depending on location (urban, rural, mountainous, plains, coast, etc.).

Although the yearly statistical distribution of wind speed for a given area is predictable to a high degree, the average speeds over a particular day or even a month still remain relatively hard to predict. This presents several problems for the incorporation of wind power to the main grid.

Incorporating wind energy into the grid requires an accurate prediction for several reasons. The first reason develops before the renewable system is built.

How much power will be generated, over the lifetime of your turbine array, in a particular area? The answer depends on the prediction of net wind speeds in that area over the lifetime of a given wind system. Indeed, to analyze the cost-effectiveness of a wind-turbine installation, total lifetime generation must be a relatively accurate metric.

Will the power system be able to provide adequate power accommodations to a given user area? The answer to this question, again, relies on wind forecasting ability. Because wind energy is not constant throughout the day, accurate prediction of the amount of power generated (and, perhaps more importantly, stored) during peak wind hours is necessary to ensure that power needs are met throughout the day.

How can we ensure the most efficient wind turbines? The application of adaptive control systems to wind turbine characteristics such as blade pitch, yaw, and rotor speed has proven to be an exciting area of research, and quite fruitful in increasing the overall efficiency of turbines. That being said, without accurate forecasting of wind speeds, control systems are forced into a purely reactive state. Much research has shown that control systems which alter the states of a turbine in a predictive manner are more efficient. The optimal performance of these control systems can therefore be seen to rely heavily on a model's ability to predict wind speeds with a higher accuracy.

This paper will explore the predictive capabilities of Neural Networks applied to wind speed data. The first section (II) will review previous work done in the field, as well as provide a relatively in-depth look at a particular paper on which this research is loosely based. The second section (III) will give a description of the data utilized in the experiments conducted herein. The third section (IV) will give the reader a brief description of the theory behind time-series prediction using neural networks, as well as their applications. The fourth section (V) delineates the several experiments performed, and their results are discussed. A discussion of these results, and a brief conclusion, will be found in (VII) and (IX), resp.

II. Previous Work

There have been several papers published in the last 30 years regarding the application of neural networks to wind speed prediction. There is an even larger volume of research available for the general theory and application of neural networks to time series prediction.

The main paper I have drawn insight from [1] is rather old, but still contains an exemplary description of Non-Linear Auto-Regressive Neural Networks (NARNN) as well as a solid comparison between general linear auto-regressive (AR) models and NARNNs. The predictive ability is measured in terms of root mean squared error, and RMSE values of 1.24 and 2.88 for daily and monthly prediction, resp.

A three-layer, feed-forward neural network trained with back-propagation is applied to wind speed prediction in [2]. The authors of this paper demonstrate significant performance increases over the persistence and autoregressive integrated moving average predictive models on data taken from Portugal. Additionally, their analysis is easily implemented in MATLAB using the "nntool" for time-series prediction. Sum of squared error, mean absolute percentage error, and standard deviation of error are the metrics they used to evaluate network performance.

In [4], both feed-forward and recurrent (Jorden) network architectures were applied to 12 years of daily, weekly, and monthly wind speed data in Mumbai, India. Feed-forward networks were the best at predicting daily values, while cascading correlation networks were most accurate at weekly predictions. Jorden recurrent architectures were most accurate for monthly predictions. The network structures that yielded best results are shared.

Reading [3] gives insight into the different performances of three types of neural networks for hourly forecasting. Back propagation, RBF (radial basis functions), and ADALINE (adaptive linear element) networks were compared by three different metrics (MAE, RMSE, MAPE).

Nogay et. al. applied a single hidden layer, 60 neuron back-propagation NN to mean daily wind speed data in Turkey, and achieved a MSE of .378088 on the testing data. They did not use a time-series approach, rather used several other meteorological data such as soil temperature and relative humidity as input data and wind speed as output data. This was the only paper that tried to estimate wind speed using something other than a time-series approach, and (although it is hard to compare across disparate data sets) got the best RMSE prediction results.

III. Data

The data analyzed in this paper originally comes from the Bonneville Power Administration (BPA). They provide meteorological data from several "Met Sites" spanning the Oregon-Washington border. A labeled map of locations for which the BPA provides data is shown in Figure (1). Dr. Mandy Herring at the Colorado School of Mines was kind enough to let me use a pre-processed version of the BPA data, with imputed data. This paper will focus on predicting wind speed from both Biddle Butte and Butler Grade meteorological stations. These two stations were chosen because they have very distinct (normalized) variances: 0.0867 and 0.1033 for Biddle Butte/Butler Grade daily, resp. To be explicit, the various neural network architectures were trained on the normalized data from 2012 at various intervals: 30 minute, hourly, and daily. The various network's performances on new data were evaluated using the normalized 2013 data of the same sample rates. The normalization is a factor that can be justified, because there is a distinct range of speeds in which the wind is most likely to occur ($\sim [0, 25]\frac{m}{s}$)

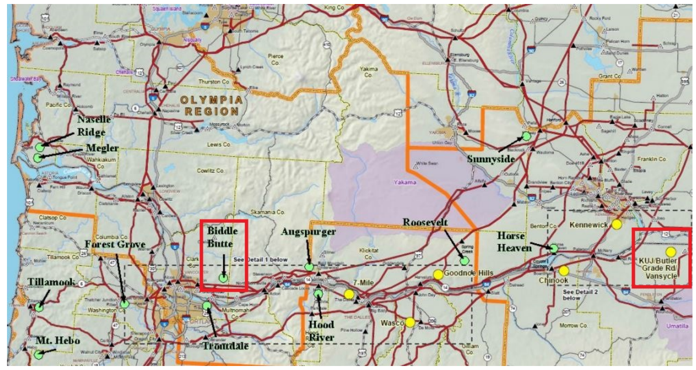

Fig. 1: Met Sites along the Washington-Oregon border. The meteorological data from the Biddle Butte and Butler Grade sites are compared in this paper

IV. Time-Series Prediction Using Neural Networks

The predictive modeling of time-series data using neural networks is an active research area. There are several different network architectures that allow the user to discern future values of a time series from previous training data. The approach used in this paper is a Non-Linear Autoregressive (NAR) model.

A. Theory

The theory of a basic feed-forward multi-layer perceptron, trained using back propagation, will suffice as the basis for NAR models. The most simple neural processing unit is called a neuron. The neuron's name derives from its ability to mimic the activity of the most simple processing unit in the brain, the neuron.

The input to a neural network is a vector **x** of N input elements $x_i, i = 1, ..., N$. A weighted linear combination of these inputs is presented to each neuron, such that this combination, presented to the mth neuron (v_m), is given by:

$$v_m = \sum_{i=1}^{N} w_{im} x_i - w_{0m}$$

where w_{im} denotes the weight from input i to neuron m, and w_{0m} is a bias term associated with a neuron, and acts to center the neuron's response about a particular critical value.

The activity (output) of each neuron is the result of running the v_m through a non-linear *activation function* $\phi()$, such that the response of the mth neuron, y_m, is given by:

$$y_m = \phi(v_m).$$

The most common activation function by far is the logistic sigmoid, $\sigma()$, where

$$\sigma(q) = \frac{1}{1 + \exp^{-q}}.$$

This paper will be using the logistic sigmoid as the activation function for all neurons in the network.

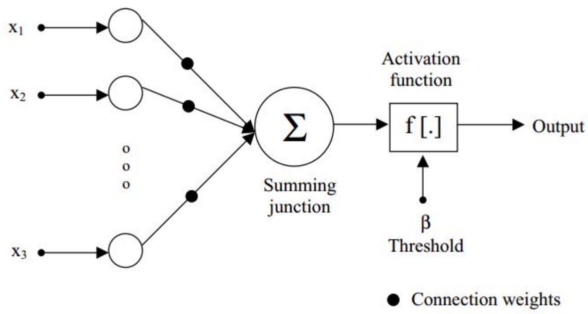

Fig. 2: Working of a single neuron [4]

The understanding of a single neuron's activity is the basis for the understanding of all of the network. Indeed, the network simply consists of different architectures linking the inputs/outputs of these neurons in interesting ways. The most common linkage method is that of the Multi-Layer Perceptron (MLP), whose structure is shown in Figure 2. Several neurons are linked to the inputs in parallel, and form the *hidden layer* of the network. Then, the outputs of each neuron in the hidden layer are used as inputs to another set of parallel neurons in the *output layer*. The number of neurons in the output layer is equal to the size of the desired network output.

In the case of time-series prediction, the hidden layer can be any size, while the output layer is generally size 1, corresponding to the desired output of y_t. The inputs are a series of past samples from the time series. For example, a network with time lag α would consist of inputs $y_{t-1}, y_{t-2}, ..., y_{t-\alpha}$, and the output of the network would be a single value, y_t. A general MLP architecture for wind speed prediction is found in Figure 3.

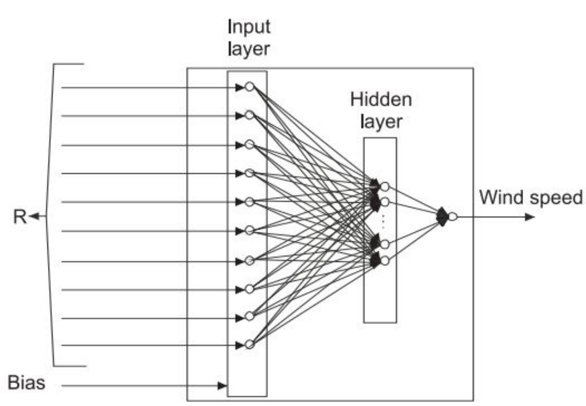

Fig. 3: Generic MLP architecture for wind speed prediction [5]

The flexibility of the network comes from the way the weights are determined. Initially, the weights connecting each element of the network are randomly initialized. Then, a series of input and output pairs are presented to the network. When a training sample pair is presented, the network attempts to minimize the error between its predicted output based solely on the input vector, and the actual known value of the output. The weights of the network are updated automatically based on the minimization of this error, and a new sample is presented. Training occurs until a specified lower bound for error is reached by the network. The method through which the weights update is known as back-propagation, discussed below. The true beauty of the network is that, through automatic processes, it determines structure *on its own* , and therefore no underlying knowledge of the system is required to get accurate predictions with a net. For this reason, many people regard NNs as a "black box" method that yields little to no insight into the actual problem. It gets results but the interpretability of the results is a bit vague.

Back-propagation is a way of transmitting the error of a network from the output back through the previous layers, and updating the weights such that an error is reduced. In this case, the error function that was minimized was the MSE:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (1)$$

where y_i is the true time-series value, $\hat{y}_i$ is the NN output, and n is the number of time-steps. Stochastic Gradient Descent is the most basic algorithm for performing back-propagation. Concise description of the basic idea can be found in [4] and [1] However, there are several variations that allow for different convergence characteristics of the network. For this paper, the Levenberg-Marquardt algorithm for error minimization was implemented using Matlab's code included in the Neural Network Toolkit (7).

B. Application

In this paper, different architectures of neural networks are applied to the data and different results are achieved. The hyper-parameters of the networks used, (number of time-lag samples, number of neurons in hidden layer, number of hidden layers) are found experimentally, by changing until a minimum RMSE on the testing data set was obtained. The RMSE can be calclated as: $RMSE = \sqrt{MSE}$ where MSE is that found in Equation 1. The programs were implemented using Matlab's Neural Network Toolkit, with custom coding to break away from the limitations of the Toolkit GUI. The GUI tends to push the user on a one-way track for different types of prediction/classification problems, and does not allow for much flexibility in the parameters. For example, the desire to have the output layer of a NAR time-series network have logistic sigmoid activation functions forced a movement out of the GUI. Training these algorithms was relatively quick across the board due to the quick convergence rate of the Levenberg-Marquardt back-prop algorithm (seconds total training + testing time).

V. Experiments

As stated before, several neural networks of various architectures were experimentally developed for the lowest RMSE performance on data taken with differing sample rates. The data were first pre-processed by normalizing and then subtracting the mean. Neural networks were trained on the data from their respective stations for the year of 2012, and had their predictive capabilities evaluated on the data from 2013. Across the board, RMSE on the training set was lower than that of the testing set, which was to be expected. In general, the short-term forecasting required more time-lag samples to be used compared to long-term predictive networks.

A. 30 Minute Forecast

Biddle Butte bi-hourly forecast was most effective with a neural network containing 15 time lags and 30 hidden neurons. Butler Grade bi-hourly forecast was most effective with a neural network containing 5 time lags and 30 hidden neurons. Details showing the predictive capabilities for Biddle Butte and Butler Grade are shown in Figure 4.

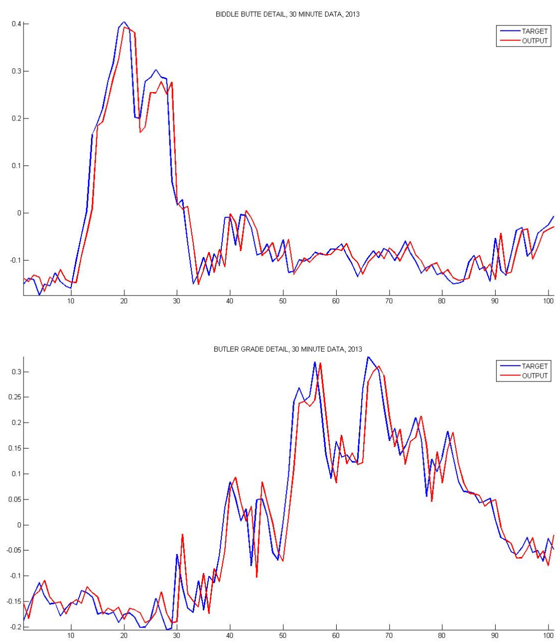

Fig. 4: **Top:** Detail of network response to the bi-hourly Biddle Butte 2013 data. **Bottom:** Detail of network response to the bi-hourly Butler Grade 2013 data.

B. Hourly Forecast

Biddle Butte hourly forecast was most effective with a neural network containing 10 time lags and 30 hidden neurons. Butler Grade hourly forecast was most effective with a neural network containing 4 time lags and 30 hidden neurons. Details showing the predictive capabilities for Biddle Butte and Butler Grade are shown in Figure 5.

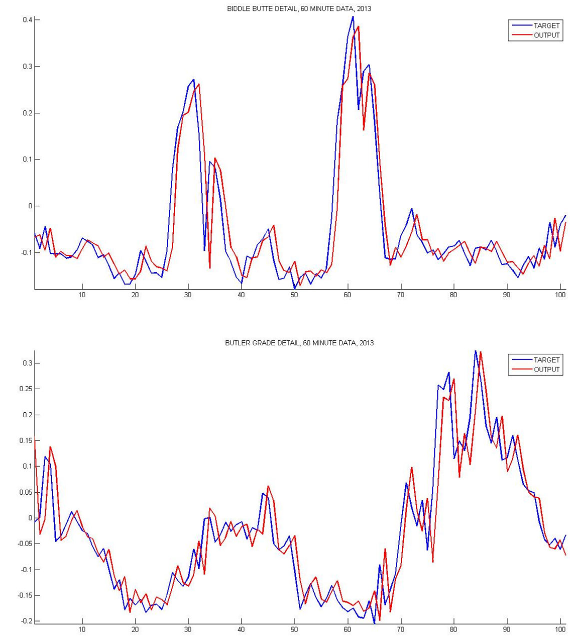

Fig. 5: **Top:** Detail of network response to the hourly Biddle Butte 2013 data. **Bottom:** Detail of network response to the hourly Butler Grade 2013 data.

C. Daily Forecast

Biddle Butte daily forecast was most effective with a neural network containing 1 time lags and 24 hidden neurons. Butler Grade daily forecast was most effective with a neural network containing 1 time lags and 24 hidden neurons as well. Details showing the predictive capabilities for Biddle Butte and Butler Grade are shown in Figure 6.

VI. Summary of Results

A summary of the experiments performed, as well as their RMSE, are shown in Table I.

Analysis	RMSE
BB 30 Min.	0.0279
BG 30 Min.	0.0360
BB Hourly	0.0355
BG Hourly	0.0453
BB Daily	0.1861
BG Daily	0.2503

TABLE I: Table of RMSE on normalized 2013 data for different predictive horizons. BB = Biddle Butte, BG = Butler Grade

VII. Discussion

A. 30 Minute Forecast

Bi-hourly forecast results were the most accurate of all time divisions investigated in Section V. The RMSE in terms of normalized wind speed found in Table I were 2.79% and 3.6% for Biddle Butte and Butler Grade, resp. This translates to a wind speed of about $.69\frac{m}{s}$ and $.9\frac{m}{s}$. A difference in predictive power of under $1\frac{m}{s}$ is tolerable for a single turbine application, considering there is not a massive difference in power for just one machine. That being said, in a large scale application this difference could be intolerable.

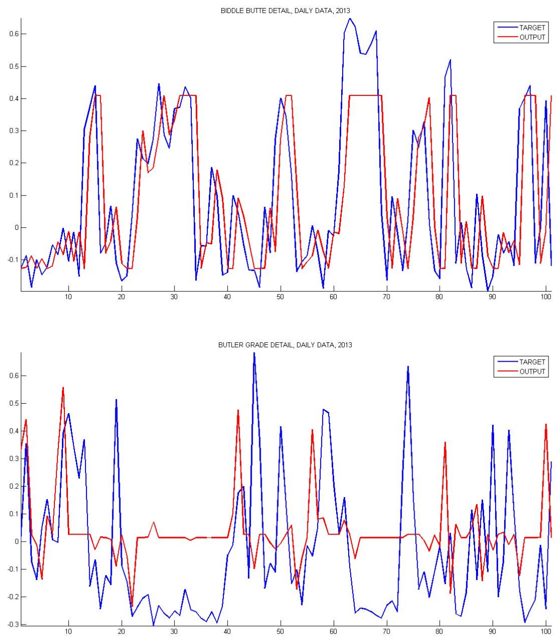

Fig. 6: **Top:** Detail of network response to the daily Biddle Butte 2013 data. **Bottom:** Detail of network response to the daily Butler Grade 2013 data.

B. Hourly Forecast

The hourly forecasts were slightly less impressive than those of the bi-hourly experiment. Again referring to Table I, the RMSEs in terms of normalized wind speed are 3.55% (Biddle Butte) and 4.53% (Butler Grade). This translates to a speed difference of about $.8875\frac{m}{s}$ and $1.1325\frac{m}{s}$. The overall difference in wind speeds is, again, perhaps tolerable in small-scale grid applications, but gets to be undesirable in large-scale generation schemes.

C. Daily Forecast

Across the board, daily forecasting provided poor results. In particular, RMSEs in Table I show daily predictive errors of 18.61% (Biddle Butte) and 25.03% (Butler Grade). This translates to a wind speed error of approximately $4.6525\frac{m}{s}$ and $6.257\frac{m}{s}$. This daily error is considerably higher than that of the hourly or bi-hourly forecasts. Predictive error this high would introduce significant difficulties in energy generation forecasting.

VIII. Future Work

Many avenues of investigation are opened via this paper. Next steps involve experimenting with some preliminary feature extraction, rather than training the neural networks on raw data. New network architectures, using external inputs such as temperature, humidity, or other meteorological data could be implemented.

IX. Conclusion

The methodology applied in this paper generated results that were significantly better in terms of RMSE than those presented in [1] with bi-hourly and hourly wind speed prediction, but fell far behind when it came to the daily prediction. Experimentation led to several distinct optimal configurations of nonlinear autoregressive neural networks, depending on the wind speed sample rate and variance. In general, more frequent sampling required architectures that accounted for more past samples. Additionally, adding more hidden layers did not tend to increase the network's accuracy, although altering the number of neurons in the hidden layer had significant effects on the predictive RMSE. It makes sense that the field of wind prediction is still an active area of research; wind forecasting remains difficult even for complex machine processing structures such as neural networks. Despite the incredible flexibility of these systems, finding patterns in the significant nonlinearities of the wind remains an elusive task.

References

[1] M. Mohandes, S. Rehman and T. Halawani, 'A neural networks approach for wind speed prediction', Renewable Energy, vol. 13, no. 3, pp. 345-354, 1998.
[2] J. Catalo, H. Pousinho and V. Mendes, 'An artificial neural network approach for short-term wind power forecasting in Portugal', Engineering Intelligent Systems, vol. 17, no. 1, 2015.
[3] G. Li and J. Shi, 'On comparing three artificial neural networks for wind speed forecasting', Applied Energy, vol. 87, no. 7, pp. 2313-2320, 2010.
[4] A. More and M. Deo, 'Forecasting wind with neural networks', Indian Institute of Technology, Mumbai, India, 2015.
[5] H. Nogay, T. Akinci and M. Eidukeviciute, 'Application of artificial neural networks for short term wind speed forecasting in Mardin, Turkey', Journal of Energy in Southern Africa, vol. 23, no. 4, 2012.
[6] MATLAB and Statistics Toolbox Release 2014a, The MathWorks, Inc., Natick, Massachusetts, United States.
[7] MATLAB and Neural Network Toolbox Release 2014a, The MathWorks, Inc., Natick, Massachusetts, United States.