Short-term wind speed forecasting by combination of neural networks

Presenting author: Ramón Velo. University of Santiago de Compostela. Spain. ramon.velo@usc.es.

Co-authors: Mª Dolores Fernández. Manuel R. Rodríguez. University of Santiago de Compostela. Spain.

1. Introduction

Short-term predictions are usually based on time series analysis to complex structures as computational fluid dynamic, neural networks and fuzzy logic. Over recent years, in order to improve the prediction of the wind speed in the next few hours, it is using new artificial neural networks (ANN). One of the advantages of the ANNs is its ability to model the problems conditioned by multiple factors and complex relationships between variables to furnish nonlinear relationships between them. Different types of neural networks compete with statistical methods and hybrid models.

The multilayer perceptron network (MLP) is a feed forward widely employed, formed of information units called neurons which are connected to each other and arranged in layers. Each neuron outputs an activation function which is applied to the weighted sum of its inputs. The logistic sigmoid function is a differentiable nonlinear activation function that performs a smooth thresholding suitable for artificial neural networks. In supervised learning is presented to the network a set of patterns together with the target output. Their weights are adjusted iteratively until the output tends to be desired, using detailed information on the error committed at every step.

Self-Organizing Maps (SOM) are neural networks that present an unsupervised learning system but competitive, in which the output neurons compete amongst themselves to be activated, with the result that only one is activated at any one time. As a result neurons are obliged to organize, forming a two-dimensional map.

Learning Vector Quantization (LVQ) is an adaptive method for data classification based on training data with the desired class information. LVQ is a nearest neighbour pattern classifier based on competitive learning and is a supervised learning algorithm of ANN.

2. Approach

The aim of this work is to improve the prediction of wind speed in the coming hours using MLP neural networks. The use of neural networks SOM and LVQ as a method to choose the input values is evaluated.

3. Main body of abstract

The structure of a neural network consists of two SOM layers, an input layer and output layer, so that the neurons are connected through adjustable weights or network parameters. The input layer consists of 2 neurons one for each input variable, 10-min values of wind speed and direction on the day before the target. They are chosen for this arrangement only the values of wind speed above 4 m/s. The neighbouring function is Gaussian, learning rate is lineal $\alpha(t) = \frac{1}{t}$ and the maximum number of interactions that allowed was 500. The obtained class is used as the criterion for the LVQ network but whenever greater number of values and always with a percentage above 40%.

The LVQ networks are composed of three layers, an input layer, a competitive intermediate layer learns to classify input vectors into subclasses and the output layer transforms the competitive layer classes in target classifications. The two classes have been previously defined by SOM neural network are represented by a feature vector. During training a pairs of sample points x and their class labels are shown to the learner, samples are shown one at a time. The closest feature vector v_w, winning feature vector is updated according to one of the following equations, depending on if feature vector v_w classifies sample point x correctly or not.

If x and v_w belong to the same class: $v_w(t+1) = v_w(t) + \alpha(t)[\,x - v_w(t)]$

If x and v_w belong to different class: $v_w(t+1) = v_w(t) - \alpha(t)[\,x - v_w(t)]$

The term *α(t)* controls how large the movement of the feature vector should be in the update. The value of *α(t)* is updated after each training step as follows: $\alpha(t+1) = \frac{\alpha(t)}{1+p\cdot\alpha(t)}$

Where *p*=1 if sample point is classified correctly, If *p*=-1 in another way.

α(t) decrease when a sample is classified correctly and increase when an incorrect classification is made by the winning feature vector. Training parameters were: 0.1 is chosen as α(0), 500 epochs, 2-20 of subclasses range, 120 s of maximum time,

Training vectors is presented to the network as input, and the Euclidean distance from the input vector to each of the feature vector is computed. Output neurons compete for activated and only one is active at given input information to the network. The winning neuron is the one whose weight vector is closest to the input vector.

The architecture of the multilayer perceptron neural network was the same for all the simulations: 2 neurons in the input layer (10-min values of wind speed and direction), 5 or 10 neurons in the hidden layer and 1 neuron in the output layer (10-min values of wind speed). For a perceptron, the network's outputs are given by next equation:

$$y_i = \sum_j \omega_{ij}^2 \tan h \sum_l \left(\omega_{jl}^1 X_l + \theta_j^1\right) + \theta_i^2$$

The function's parameters are given by the set of weights. $\Omega = \left\{\omega_{jk}^1, \omega_{ij}^2, \theta_j^1, \theta_i^2 \right\}$ Where k runs over the networks input (1 and 2), j runs over the hidden layer (5 and 10) and i runs over the network's outputs (1). Therefore, ω_{jk}^1 are the weights associated with the connections between the network's inputs and the hidden layer and θ_j^1 the bias added to the input of the hidden layer's neurons. Similarly, ω_{jl}^2 are the weights associated with the connections between the hidden layer and the network's outputs and θ_j^2 the bias added to the input of the neurons of the hidden layer.

The utilization of MLP neural networks is divided into two phases, a training phase, during which a set of training patterns is used to determine the weights that define the neural model. For network training ten-minute values of wind speed and direction belonging to a period of 30 days is chosen. The selected range of data is provided LVQ network.

The period has a class containing values higher than 40% is chosen. We observed a relationship expressed by the following equation: $RMSE = 101{,}8e^{-0{,}19x}$

Where is x the number of elements that constitute the class of the chosen period.

In the second phase, named test, the patterns are processed and its input values are the speed and the direction of the day previous to the target day.

The training algorithm used was backpropagation and the learning is performed by modifying weights in successive iterations by applying gradient descent method until the error is minimized. Is is chosen a maximum number of epochs to train: 1000, learning rate: 0,1. The initialization of the weights of all neurons is carried up to a random value.

The results obtained are compared with those provided by the persistent pattern (which states that the expected value k time steps ahead is equal to the most recent value. $P_t = P_{t+k}$) and ARIMA (autoregressive integrated moving average) time series models more parsimonious. Previously to identify the model should take into account the restriction of stationarity must be fulfilled. The time series must be stationary in mean, variance, and autocovariance.

The data selected correspond to stations located in Northwest Spain (Galicia): Ons (42,38º; -8,93º) at an altitude of 121 m and Punta Candelaria (43,71º; -8,05º) at an altitude of 254 m (Meteogalicia). Neural Toolbox™ software of MATLAB was used for the creation and resolution of the ANN models.

The performance of the models in forecasting accuracy was assessed using the following evaluation criteria: Pearson´s correlation coefficient (ρ), Root Mean Square Error (RMSE), Index of Agreement (IOA), Mean Average Error (MAE). Table 1 shows the average values of the some of the errors calculated at 50 simulations.

Table 1. Root Mean Square Errors for different models used to simulate the wind speed at the next 6 and 24 hours.

Input (ten-minute speed and direction values)	Model	RMSE	
		24 h	6 h
30 previous days to the target	Persistence	3,36	2,51
	ARIMA (2 1 0)	2,62	1,22
	2_5_1	4,22	3,12
	2_10_1	4,40	3,02
30 days chosen by means LVQ and SOM (>40 %)	2_5_1	2,83	2,49
	2_10_1	2,75	2,57

It is notable the significant drop in RMSE (above 30 %) using the method of electing the proposed training period. In the prediction for six hours, the error is less between 6 and 12% respect to 24 hours. No significant differences were observed between the neural networks used.

4. Conclusion

In this work proposes a model to predict the wind speed in the coming hours. The results indicate that using networks LVQ and SOM for input selection improves prediction of the MLP neural networks.

5. Learning objectives

- Reduce uncertainty in predicting the wind speed by the combined use of several artificial neural networks.
- Compare the efficiency in predicting short-term wind speed of different methodologies.
- Highlight the strengths of SOM and LVQ neural networks.
- Compare the efficiency of different types of ANNs.