Resilient Propagation for Multivariate Wind Power Prediction

Jannes Stubbemann, Nils Andre Treiber and Oliver Kramer
Department of Computer Science, University of Oldenburg, 26111 Oldenburg, Germany

Keywords: Neural Networks, Resilient Propagation, Wind Power Prediction.

Abstract: Wind power prediction based on statistical learning has the potential to outperform classical physical weather prediction models. Neural networks have been successfully applied to wind prediction in the past. In this paper, we apply neural networks to the spatio-temporal prediction model we proposed in the past. We concentrate on a comparison between classical backpropagation and the more advanced resilient propagation (RPROP) variants. The analysis is based on time series data from the NREL western wind data set. The experimental results show that RPROP+ and iRPROP+ significantly outperform the classical backpropagation variants.

1 INTRODUCTION

While physical models for wind power predictions have advantages in long-term range, statistical methods perform well for short-term predictions (Lei et al., 2009). With this motivation, we have built a spatio-temporal prediction model and employed linear, k-nearest neighbor and support vector regression techniques in the past (Kramer et al., 2013; Treiber et al., 2013). In this paper, we experimentally compare resilient propagation (RPROP) variants to enhance the methodological repertoire of regression methods for wind power prediction based on the spatio-temporal model.

The paper is structured as follows. In Section 2, we introduce the wind power prediction problem as regression problem and present related work. RPROP and its variants are shortly sketched in Section 3. The experimental comparison is presented in Section 4. Results are summarized and discussed in Section 5.

2 WIND POWER PREDICTION

The power grid is moving from a stable supply of comparatively few centralized power plants to a heterogenous grid with thousands of entities. Their power is fluctuating and depending on wind and sun. For a stable integration of renewables into the grid, a precise prediction is important. Since a couple of years data-driven models have shown to deliver competitive results in wind power prediction.

2.1 Wind Prediction as Regression Problem

We treat the wind power prediction problem as regression problem. Given a training set of pattern-label observations $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subset \mathbb{R}^d$, we seek for a regression model $f : \mathbb{R}^d \to \mathbb{R}$ that learns reasonable predictions for power values of a target turbine given unknown patterns $\mathbf{x}'$. We define the patterns as wind power features $\mathbf{x}_i \in \mathbb{R}^d$ of a target turbine and its neighbors at time t (and the past) and the labels as target power values y_i at time $t + \lambda$. Such wind power features $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,d})^T$ can consist of present and past measurements, i.e., $p_j(t), \ldots, p_j(t-\mu)$ of $\mu \in \mathbb{N}^+$ time steps of turbine $j \in \mathcal{N}$ from the set $\mathcal{N}$ of employed turbines (target and its neighbors) or wind power changes $p_j(t-1) - p_j(t), \ldots, p_j(t-\mu) - p_j(t-\mu+1)$. In the experimental evaluation of this work, we construct each pattern with $\mu = 2$ past measurements and only consider absolute values (and no differences).

2.2 Related Work

Neural networks have been applied to data-based wind prediction in the past. For example, Mohandes et al. compared an autoregressive model with a classical backpropagation network for wind speed prediction and demonstrated the superior accuracy of the neural network model (Mohamed A. Mohandes and Halawani, 1998). In a similar line of research, Catalao et al. trained a three-layered feedforward network

Stubbemann J., Andre Treiber N. and Kramer O..
Resilient Propagation for Multivariate Wind Power Prediction.
DOI: 10.5220/0005284403330337
In *Proceedings of the International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2015), pages 333-337
ISBN: 978-989-758-077-2

with the Levenberg-Marquardt algorithm for short-term wind power forecasting, which outperformed the persistence model[1] and ARIMA approaches (Catalao et al., 2009). Han et al. focused on an ensemble method of neural networks for wind power prediction (Han et al., 2011).

In our preliminary work, we proposed the spatio-temporal approach for support vector regression in (Kramer et al., 2013) that has been introduced in the previous section. In (Treiber et al., 2013), we concentrated on the aggregation of features with nearest neighbors and support vector regression on the level of a wind park. Recently, we proposed an ensemble approach based on statistical learning methods (Heinermann and Kramer, 2014). An evolutionary method for feature selection has been proposed by (Treiber and Kramer, 2014).

3 RESILIENT PROPAGATION

RPROP has been introduced as variant of backpropagation to allow faster learning and avoiding oscillating around local optima.

3.1 Backpropagation

The backpropagation learning algorithm for training of feedforward networks has originally been introduced by Rumelhart *et al.* (Rumelhart et al., 1986). It is based on fitting the network output o_{net} to the target values y_i given pattern $\mathbf{x}_i$. This can be written as error function[2] E_{net}:

$$E_{net} = \frac{1}{2}\sum_{i=0}^{N}((y_i - o_{net}(\mathbf{x}_i)^2) \quad (1)$$

In backpropagation, the weights of the network w_{ij} are adapted via gradient descent

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \quad (2)$$

with

$$\Delta w_{ij}^{(t)} = -\rho \cdot \frac{\partial E}{\partial w_{ij}}^{(t)} \quad (3)$$

and learning rate ρ. As a variant, backpropagation with momentum (BPMom) considers the last weight update in the current update

$$\Delta w_{ij}^{(t)} = -\rho \cdot \frac{\partial E}{\partial w_{ij}}^{(t)} + \alpha \cdot \Delta w_{ij}^{(t-1)} \quad (4)$$

to reduce oscillations around local optima.

[1]The naive persistence model assumes that the wind will not change within time horizon λ.

[2]The definition corresponds to the mean squared error (MSE) of the prediction.

3.2 RPROP

RPROP has been introduced as extension of classical backpropagation by Riedmiller and Braun (Riedmiller and Braun, 1992). The idea of RPROP is to replace the constant learning rate ρ by adaptive step sizes Δ_{ij} that are controlled during learning for each weight w_{ij}. The step sizes Δ_{ij} are updated as follows

$$\Delta_{ij}^{(t)} = \begin{cases} \Delta_{ij}^{(t-1)} \cdot \eta^{+} & \text{if } \frac{\partial E}{\partial w_{ij}}^{(t)} \cdot \frac{\partial E}{\partial w_{ij}}^{(t-1)} > 0 \\ \Delta_{ij}^{(t-1)} \cdot \eta^{-} & \text{if } \frac{\partial E}{\partial w_{ij}}^{(t)} \cdot \frac{\partial E}{\partial w_{ij}}^{(t-1)} < 0 \\ \Delta_{ij}^{(t-1)} & \text{else} \end{cases} \quad (5)$$

with parameters $0 < \eta^{-} < 1 < \eta^{+}$, which specify the step size adaptation magnitude, if the sign of the gradient $\frac{\partial E}{\partial w_{ij}}^{(t-1)}$ changes. The weights are updated with

$$\Delta w_{ij}^{(t)} = -\text{sign}(\frac{\partial E}{\partial w_{ij}}^{(t)}) \cdot \Delta_{ij}^{(t)} \quad (6)$$

Two RPROP variants that include backtracking are introduced in the following.

for each w_{ij} **do**
  if $\frac{\partial E}{\partial w_{ij}}^{(t-1)} \cdot \frac{\partial E}{\partial w_{ij}}^{(t)} > 0$ **then**
    $\Delta_{ij}^{(t)} := \min(\Delta_{ij}^{(t-1)} \cdot \eta^{+}, \Delta_{\max})$
    $\Delta w_{ij}^{(t)} := -\text{sign}(\frac{\partial E}{\partial w_{ij}}) \cdot \Delta_{ij}^{(t)}$
    $w_{ij}^{(t+1)} := w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$
  elseif $\frac{\partial E}{\partial w_{ij}}^{(t-1)} \cdot \frac{\partial E}{\partial w_{ij}}^{(t)} < 0$ **then**
    $\Delta_{ij}^{(t)} := \max(\Delta_{ij}^{(t-1)} \cdot \eta^{-}, \Delta_{\min})$
    $w_{ij}^{(t+1)} := w_{ij}^{(t)} - \Delta w_{ij}^{(t)}$
    $\frac{\partial E}{\partial w_{ij}}^{(t)} := 0$
  elseif $\frac{\partial E}{\partial w_{ij}}^{(t-1)} \cdot \frac{\partial E}{\partial w_{ij}}^{(t)} = 0$ **then**
    $\Delta w_{ij}^{(t)} := -\text{sign}(\frac{\partial E}{\partial w_{ij}}) \cdot \Delta_{ij}^{(t)}$
    $w_{ij}^{(t+1)} := w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$

Algorithm 1: Pseudocode of RPROP+, oriented to Igel and Hüsken (Igel and Hüsken, 2003). The weight update is reverted in case the sign of the partial derivative has changed.

3.3 RPROP+

RPROP+ is an extension of RPROP and has been introduced by Igel and Hüsken (Igel and Hüsken, 2003). The idea of RPROP+ is to revert the weight update step in case the sign of the partial derivative has changed. Algorithm 1 shows the pseudocode of RPROP+. If the sign of the partial derivative has not changed in the last iteration, the step size Δ_{ij} of weight w_{ij} increases. The step size is limited by the maximum value $\Delta_{\max}$. If the sign of the gradient has changed in the last iteration, step size $\Delta_{ij}^{(t)}$ is decreased (again limited by the minimum value Δ_{min}) and the last weight update is reverted. Last, the current gradient is reset to 0 to enforce the last condition (Line 12), which conducts a weight update with the new (reduced) step size.

Figure 1 illustrates with working principle of RPROP+. An increase of the step size in case the sign of the partial derivative has not changed is reasonable

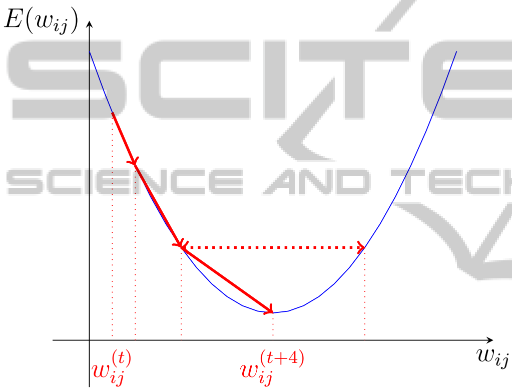


Figure 1: Illustration of gradient descent with RPROP+.

to accelerate the walk into the direction of the optimum (left two solid arrows). In case the optimum is missed and the sign of the partial derivates have changed (dotted arrow), the following gradient descent step is performed from the previous position with a decreased step size (w^{t+4}).

3.4 iRPROP+

A further method we test is the improved resilient propagation with backtracking (iRPROP+) (Igel and Hüsken, 2003), which is an extension of RPROP+. The difference to RPROP+ is that the weight update is only reverted, if it led to an increased error, i.e., if $E^{(t)} > E^{(t-1)}$. In pseudocode 1, Line 9 must be replaced by

$$\text{IF} \quad E^{(t)} > E^{(t-1)} \quad \text{THEN} \quad w_{ij}^{(t+1)} := w_{ij}^{(t)} - \Delta w_{ij}^{(t)}$$

The variants are experimentally compared in the next section.

4 EXPERIMENTAL ANALYSIS

In this section, we compare standard backpropagation, backpropagation with momentum, RPROP+, and iRPROP+ experimentally. For this sake, the four methods are run for 2000 iterations on test data sets in turbines from Casper, Las Vegas, Reno, and Tehachapi for a prediction horizon of $\lambda = 3$ steps (30 minutes). We use each 5th pattern of the wind time series data of year 2004. The resulting data set consists of 10512 patterns, of which 85% are used for training and 15% are randomly drawn for the validation set. Each training process is repeated three times. The topologies of the neural networks depend on the number of employed neighboring turbines, which determine the dimensionality of patterns $\mathbf{x}_i$:

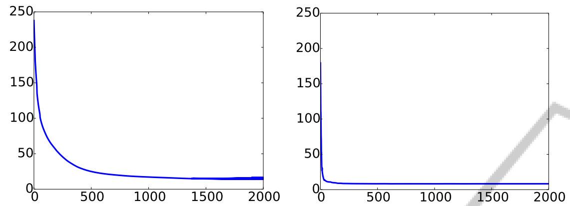
- Casper: 33 input neurons (10 neighboring turbines, 1 target turbine, 3 time steps), 34 hidden neurons
- Cheyenne: 33 input neurons (10 neighboring turbines, 1 target turbine, 3 time steps), 34 hidden neurons
- Las Vegas: 30 input neurons (9 neighboring turbines, 1 target turbine, 3 time steps), 31 hidden neurons
- Reno: 30 input neurons (9 neighboring turbines, 1 target turbine, 3 time steps), 31 hidden neurons
- Tehachapi: 21 input neurons (6 neighboring turbines, 1 target turbine, 3 time steps), 22 hidden neurons

For the classical backpropagation variants, the following parameters are chosen: $\rho = 3 \cdot 10^{-7}$ and $\alpha = 1 \cdot 10^{-8}$ for BPMom. For RPROP, the following parameters are chosen: $\Delta_{\min} = 1 \cdot 10^{-6}$, $\Delta_{\max} = 50, \eta^- = 0.5$, and $\eta^+ = 1.2$. Table 1 shows the experimental results. The figures show the validation error in terms of MSE. RPROP+ and iRPROP clearly outperform the two classical backpropagation variants.

In the following, we analyze and compare the learning curves of BP and RPROP. Figure 2 shows the validation error development in terms of MSE in the course of backpropagation and iRPROP+ training for the Tehachapi data sets. The plots show that RPROP+ achieves a significantly faster training error reduction than backpropagation. A closer look at the learning curves (in terms of validation error) offers Figure 3. Each three runs of backpropagation show a smooth approximately linear development. iRPROP+ based training reduces the errors faster, but also suffers from slight deteriorations during the learning process. However, the situation changes at later stages of

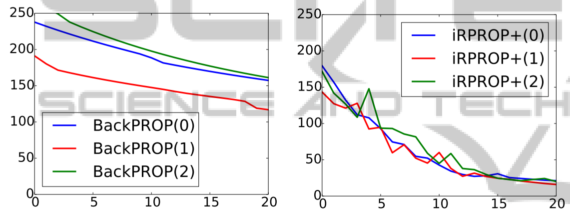
Table 1: Comparison of standard backpropagation (BackPROP), backpropagation with momentum (BackPROPMom), RPROP+ and iRPROP+ in terms of validation error (MSE).

Algorithm	Casper	Cheyenne	Las Vegas	Reno	Tehachapi	∅
BackPROP	19.51(8.73%)	19.32(15.17%)	14.82(9.64%)	16.9(9.34%)	17.66(13.36%)	17.64(11.25%)
BackPROPMom	19.45(8.3%)	20.62(13.41%)	14.32(6.2%)	16.35(3.45%)	18.02(9.33%)	17.75(8.14%)
RPROP+	10.57(3.87%)	8.37(4.12%)	10.83(3.97%)	12.32(3.32%)	7.62(3.04%)	9.94(3.66%)
iRPROP+	10.76(5.08%)	8.56(3.83%)	10.92(3.52%)	12.47(3.11%)	7.81(3.9%)	10.1(3.89%)



(a) BP (b) iRPROP+

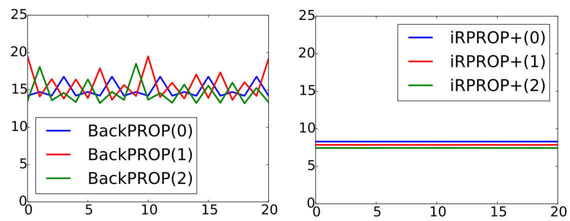
Figure 2: Learning curve for training of (a) backpropagation and (b) iRPROP+ training on Tehachapi for 2000 iterations.



(a) BP (b) iRPROP+

Figure 3: Learning curve for training of (a) backpropagation and (b) iRPROP+ training on Tehachapi for the first 20 training iterations.

learning, see Figure 4, which shows the last 20 iterations. Here, the backpropagation learning curves fluctuate, while the iRPROP+ training curves converge smoothly to constant values.



(a) BP (b) iRPROP+

Figure 4: Learning curve for training of (a) backpropagation and (b) iRPROP+ training on Tehachapi for the last 20 training iterations.

The analyzed methods reach the optimum faster than in 2000 iterations. Table 2 analyzes how fast the optimum is reached, i.e. the number of training iterations until the optimal validation error has been reached. The experiments show that RPROP+ and iRPROP+ reach their optima faster than the backpropagation variants. On Casper, Las Vegas and Reno, they need far less than 1000 iterations for an optimal training. iRPROP+ is on average 37% faster than classic BackPROP and 32% faster than BackPROPMom.

Table 2: Average number of iterations until minimum validation error has been reached for all four neural networks on five parks.

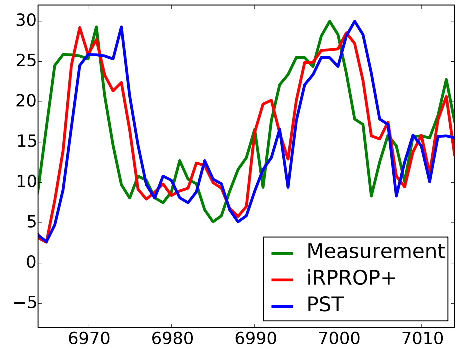
Park	BP	BPMom	RPROP+	iRPROP+
Casper	1090	1025	936	919
Chey.	997	702	1316	803
L.V.	1339	1421	908	735
Reno	1999	1993	650	551
Teh.	1220	1015	1645	1190
∅	1328	1231	1091	839

The results confirm that both RPROP variants outperform backpropagation. Further, iRPROP+ is up to one third faster than RPROP+.

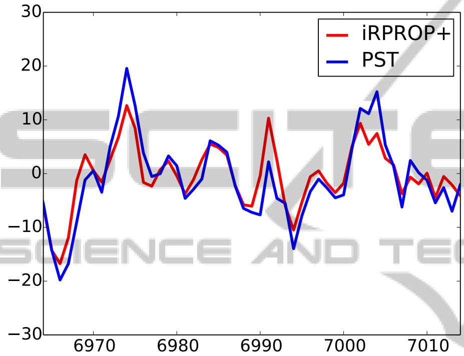
In the last part of the experimental analysis, we show the results of RPROP in time series prediction scenarios. The prediction results of iRPROP+ on a short exemplary interval are shown in the following. Figure 5(a) shows a comparison between the predicted wind power of iRPROP+, the persistence model PST, and the real measurements y. The figure shows that the curve of iRPROP+ is closer to the curve of y. This observation is confirmed by the plot of deviations, see Figure 5(b). The iRPROP+ deviation from the real wind power is in most cases smaller than the deviation than PST. Interestingly, ramp-up events are over-estimated, i.e., they are predicted earlier than they occur.

5 CONCLUSIONS

In this work, we applied neural networks to the spatio-temporal time series regression approach for the first time. The experimental analysis has shown that the two RPROP variants RPROP+ and iRPROP+ have outperformed the standard backpropagation algorithms. The conditional acceptance of gradient descent steps turns out to be advantageous for learning the short term wind power prediction. In the future, we will extend the experimental results to get statistically significant results and enlarge the number of test data sets.



(a) prediction with iPROP+



(b) deviation from measurement

Figure 5: Prediction of wind power for short time interval with iRPROP+ on Tehachapi (a) prediction and (b) deviation from target values y_i

In further experiments, we observed that the backpropagation variants outperform the persistence model, nearest neighbor regression, and linear regression. We expect to improve the RPROP results with an adaptation of the network architecture, e.g. with GPROP (Castillo et al., 2000), which tunes the network topology and initial parameters with genetic algorithms. Further, we plan to apply deep learning to the spatio-temporal prediction scenario.

ACKNOWLEDGEMENTS

We thank the National Renewable Energy Laboratory (NREL) for the western wind data set.

REFERENCES

Castillo, P. A., Merelo, J. J., Prieto, A., Rivas, V., and Romero, G. (2000). G-prop: Global optimization of multilayer perceptrons using GAs. *Neurocomputing*, 35:149–163.

Catalao, J. P. S., Pousinho, H. M. I., and Mendes, V. M. F. (2009). An artificial neural network approach for short-term wind power forecasting in portugal. In *Intelligent System Applications to Power Systems, 2009. ISAP '09. 15th International Conference on*, pages 1–5.

Han, S., Liu, Y., and Yan, J. (2011). Neural network ensemble method study for wind power prediction. In *Power and Energy Engineering Conference (APPEEC), 2011 Asia-Pacific*, pages 1–4.

Heinermann, J. and Kramer, O. (2014). Precise wind power prediction with SVM ensemble regression. In *Artificial Neural Networks and Machine Learning - ICANN 2014 - 24th International Conference on Artificial Neural Networks, Hamburg, Germany, September 15-19, 2014. Proceedings*, pages 797–804.

Igel, C. and Hüsken, M. (2003). Empirical evaluation of the improved rprop learning algorithm. *Neurocomputing*, 50:105–123.

Kramer, O., Gieseke, F., and Satzger, B. (2013). Wind energy prediction and monitoring with neural computation. *Neurocomputing*, 109:84–93.

Lei, M., Shiyan, L., Chuanwen, J., Hongling, L., and Yan, Z. (2009). A review on the forecasting of wind speed and generated power. *Renewable and Sustainable Energy Reviews*, 13(4):915 – 920.

Mohamed A. Mohandes, S. R. and Halawani, T. O. (1998). A neural networks approach for wind speed prediction. *Renewable Energy*, 13(3):345 – 354.

Riedmiller, M. and Braun, H. (1992). Rprop - a fast adaptive learning algorithm. *Proc. of ISCIS*, VII.

Rumelhart, D., Hintont, G., and Williams, R. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536.

Treiber, N. A., Heinermann, J., and Kramer, O. (2013). Aggregation of features for wind energy prediction with support vector regression and nearest neighbors. In *European Conference on Machine Learning, DARE Workshop*.

Treiber, N. A. and Kramer, O. (2014). Evolutionary turbine selection for wind power predictions. In *KI 2014: Advances in Artificial Intelligence - 37th Annual German Conference on AI, Stuttgart, Germany, September 22-26, 2014. Proceedings*, pages 267–272.