Neural Networks Based Local Weather Prediction System

Ján Adamčák, Rudolf Jakša, and Ján Liguš

Technical University Košice
Letná 9, Košice, Slovakia
jadamcak@gmail.com, jaksa@neuron.tuke.sk

Abstract. In this paper we describe how to build a fully autonomous system for collection, prediction and presentation of single-position meteorological data - the local weather prediction system. By employing non-linear statistics with neural network predictor on meteorological time-series data we were able to achieve good results for the one-day weather prediction. This novel local statistical approach to weather prediction is different compare to standard methods which are based on the air mass movement modelling. Main objective of this paper is to describe whole system for local weather prediction including technology, software, methods and parameters, and also experimental results.

1 Introduction

Our local weather prediction system with neural networks is based on approximation of weather function by black-box model from weather data collected in particular local region. This will create the weather model for single position on the map. Weather forecast agencies produce forecasts for bigger regions or even for continents or whole globe. The local prediction method is not practical for agency forecasts. However, for local applications, where we are interested in local weather course, this approximation-based weather prediction can be useful. It is able to produce prediction with short 15 minutes period. The local weather prediction can be used as an extension and refinement of agency weather forecasts. In our papers [3][4] we documented our previous work with local weather prediction for district heating company application. In that work we used longer historical data and wider selection of meteorological and technology variables for training the predictor. In this work we will address the prediction from data from standard meteorological station.

2 Weather Prediction System

Whole system consists from meteostation, control server and the public server with internet connection. We use standard meteostation Davis Vantage Pro 2 with simple control panel. It is capable to measure main weather parameters as air temperature, humidity, dew point, wind chill, heat index, air pressure,

wind speed, wind degree, rain rate and solar radiation. Important is that we can control it remotely from the Linux box. Connection between meteostation and Linux box is provided via USB cable and the dwt.c application. Connection between Linux box as a control server and public server is based on 2.4GHz WiFi IEEE 802.11g connection secured via WPA2-PSK with AES cipher. Public server is connected to internet. Using WiFi connection between control and public server, these are galvanically isolated. This overall scheme is on the Figure 1. Control and public servers operate several subsystems:

1. meteostation access subsystem to measure and log data from meteostation,
2. the timer for sequential task run,
3. prediction subsystem to predict weather in near future,
4. visualization subsystem to convert numerical data to graphical data,
5. web subsystem to publish visualizations to internet,
6. and finally for right cooperation of all subsystems we need data transformation subsystem.

2.1 Meteostation Access Subsystem

For the low-level control and data access we use free Linux application dwt.c by Con Tassios [6]. This software is available under GPL 2 license. We wrote a patch for dwt.c to match it better with our meteostation. We distribute this patch under the same license as the first author.

Patch does include changed bus specification from RS232 to USB. It also includes new methods for measurement of a voltage of included battery, for measurement of a solar radiation and for measurement of evapotranspiration. Biggest change is addition of method for diagnostic messages. These messages

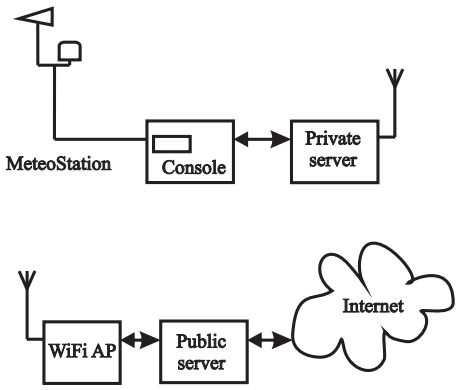


Fig. 1. Overall physical structure of weather prediction system, meteostation and two servers isolated with the WiFi connection.

Table 1. Log values and their types

VALUES	MEASURAND	UNITS	VALUES	MEASURAND	UNITS
00:16:55	TIME	HH:MM:SS	0.0	RAIN_MONTH	mm
1-10-2013	DATE	D-M-YYYY	863.6	RAIN_YEAR	mm
2.8	OUT_TEMP	°C	0.0	RAIN_RATE	mm/h
89	OUT_HUMIDITY	%	0.0	RAIN_STORM	mm
1.1	DEW_POINT	°C	65535	START_STORM	unrecognized value
2.8	WIND_CHILL	°C	622	SUNRISE	HMM
2.8	HEAT_INDEX	°C	1615	SUNSET	HMM
1023.8	BAROMETER	hPa	0.0	SOLAR_RAD	W/m(2)
0	WIND_SPEED	km/h	0.0	ET_TODAY	mm
12	WIND_DEGREE	°	0.0	ET_MONTH	mm
0	AVG_WIND	km/h	77.3	ET_YEAR	mm
0.0	RAIN_TODAY	mm	9	FORECAST_RULE	8bit number

translate 200 forecast rules for next 4 hours created by vendor of meteostation. Measured data are written into text file line by line, where every line is single measurement. Output format of line is of the type: VALUE (space) VALUE. All values and their types are in the Table 1. File header is second column starting with the character #. Measured data stream is in the first column.

2.2 Timer Subsystem Structure

Timer subsystem structure consist of standard Linux timer cron. This tool is our synchronization clock for the right timing of all other subsystems. Its setup is simple and flexible. Main activity is running every fifteen minutes. This main activity starts measurement, logging, parsing logs, prediction, parsing predictions, generation of visualizations and copying all public data to webserver.

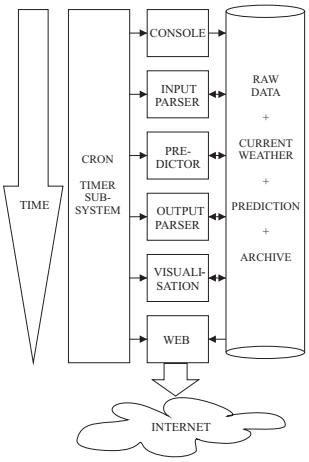
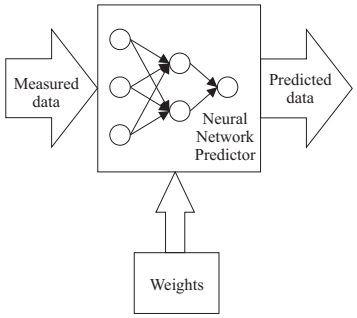


Fig. 2. Timer subsystem structure - time flow.

The order of tasks is very important to generate a correct prediction. The whole scheme of timer subsystem is on the Figure 2. The tasks sequence is defined by time of running tasks and input and output data for these tasks.

1. First running task is measurement task. This task write a new line with measurement at the end of file data.log. This task is defined in Meteostation Access Subsystem section.
2. Second running task is first parser task. This task transforms data.log file to predictor input file. This transformation is needed for fast run of predictor.
3. Third running task is the predictor. This subsystem use as input parsed data.log file to generate a prediction of outdoor temperature for next 24 hours. This prediction is written in predictor.log at the end of file.
4. Fourth running task is second parser task. This task does transform output of first parser task and output of predictor task to series of files for the visualization task.
5. Visualization task is running as fifth task in this sequence. Its main role is generate plots via gnuplot tool from preprocessed measurement and prediction outputs. These plots are saved in JPEG format for simple viewing on all devices.
6. After generation of plots is running Web subsystem task for publishing outputs of measurement, prediction and visualization. This task in first step does upload all relevant outputs to public webserver. Second step of this task is nonstop running Apache webserver with uploaded data. This webserver with configured webpage provides distribution of data on user requests. Public webpage does include simple javascript for automatic reload after fifteen minutes from last download to keep content actualized.



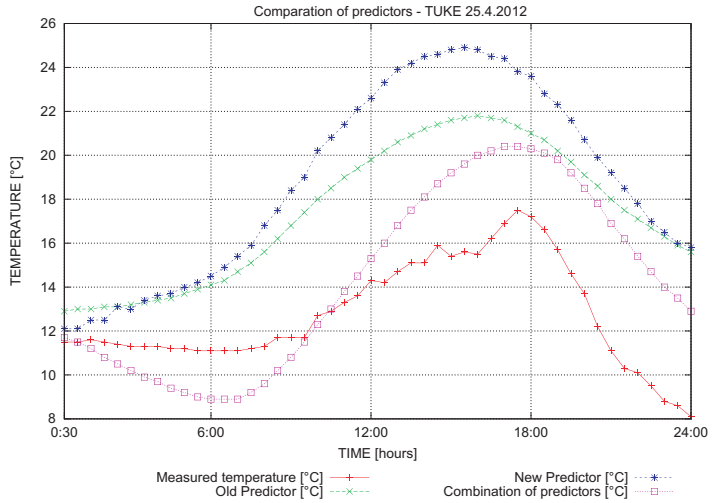
Parameter	Value
Number of input neurons	866
Number of hidden layers	1
Number of hidden neurons	300
Number of output neurons	48
Activation function	Sigmoid
Output transform. function	((4*x)-2)*20
Sigmoid parameter alfa	0.5
Bias value of neuron	-1

Fig. 3. Prediction subsystem structure and configuration of neural network predictor. Neural network weights are trained and stored externally.

2.3 Prediction Subsystem

Prediction subsystem consists of empty model of feed-forward neural network with stable configuration and weights file. Neural network configuration is shown in Table 3. For the correct function of predictor this network needs trained weight file. This file represents knowledge learned during learning phase of neural network. We used standard multilayer perceptron neural network with one hidden layer and backpropagation learning algorithm (see [1][2]).

For training we used data obtained between 7/2011 and 3/2012. These data are public on web page of this project [7]. Input of the prediction subsystem is time window generated from the last 288 measurements in 15 minutes intervals but predictor does use 144 measurements in 30 minutes intervals. This setup is copied from [5], where additional sources of data with 30 minutes sampling were used. As the input for predictor we used distance from the midnight and distance from the new year and additional six variables: Air Temperature, Dew Point, Humidity, Barometer, Wind speed, Wind Direction.



Name of experiment	Old predictor	New predictor	Combination
Sum of errors in °C	224.2	291.9	125.6
Average error per neuron in °C	4.67	6.08	2.62

Fig. 4. Results of the predictor on example day 25.4.2012. Errors in the table are from the same day as the figure. Sum of errors is value after summing all 24 hours errors in one prediction. Average of errors is value of sum divided by number of output neurons (48). For this day, the best predictor is combination predictor. Predictor error is caused by short training set obtained from meteostation and differences in weather between meteostation place and the airport, which was main source of training data.

Output of predictor is logged into archive for later statistical processing to evaluate quality of predictions. Also the last prediction from this file is used for the visualization subsystem. Results of the predictor are visible on webpage of the project [7] and on the Figure 4. These results were obtained with following settings of experiment:

1. Old predictor is winner of series of experiments based on data from KSC airport in Košice. These data are from years 2000 to 2011. From these data we used 170 000 examples splitted to training set of 140 000 examples and testing set of 30 000 examples. The learning process was stopped after stabilization of learning error.
2. New predictor is winner of series of experiments based on data from meteostation used in this project in Košice. These data were measured from July 2011 to March 2012. All data contained 29 000 examples for learning. The learning process was stopped after stabilization of learning error.
3. Combination of predictors is the winner of series of experiments when we used the worst predictor from the old predictors and this predictor is relearned using new data from our meteostation. This relearning was stopped after stabilization of learning error.

Predictor can be extended to predict other meteorological values as Dew Point, Humidity, Barometer, Wind Speed and Direction (see [5]). All we need for this upgrade are additional trained weight files for prediction.

2.4 Data transform subsystem

Data transform subsystem consists of some simple programs to convert measured data to the format needed by predictor and visualization subsystem (input parser) and programs to convert predicted data for visualization (output parser & gnuplot) and programs to generate the archive for public.

Input parser is simple program written in C language. Its function is to transform measured data from archive of all measurements to time window for predictor. This time window is written to two files. First file is hist.txt file for next processing by output parser. Second file is priklad.txt, where the data after scaling for neural network are written. Scaling is simple - all data are multiplied with a constant to normalize input to interval from -2 to +2. it is also used to transform the output of predictor back to real units.

Output parser is simple program written in C language, too. Its function is transform output of predictor to format for visualization subsystem. The input is hist.txt file from input parser and the output of predictor.

Current weather iframe is generated by input parser, too. This iframe contains only last measurement and is useful as current weather numerical information. This iframe is written in html.

Archive of measured data is split into months. It's made from full measurement data.log file every month after the midnight automatically by the timer subsystem. This mini archive files are controlled for consistency by administrator of system. After the check the files are copied to public web archive using SCP Linux command.

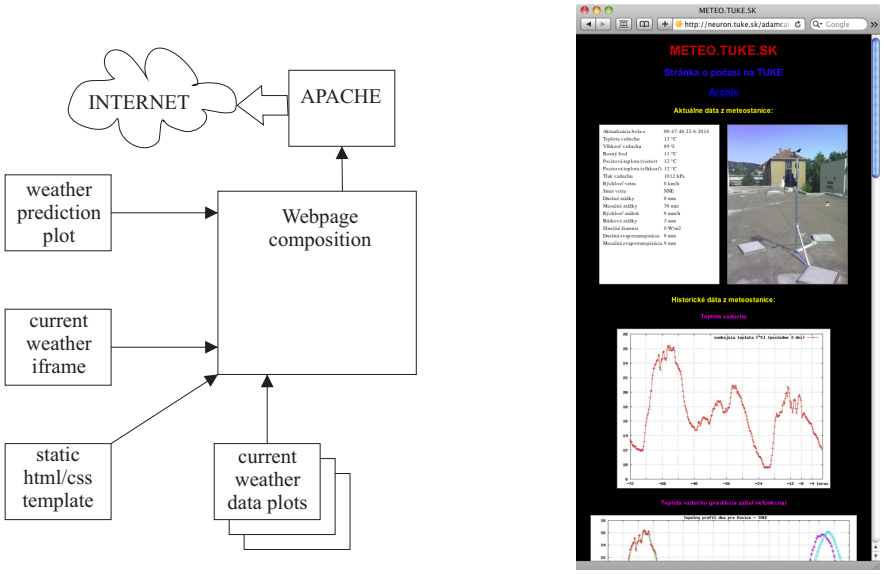


Fig. 5. The web subsystem and the screenshot of webpage with included visualizations from gnuplot. Top-left frame on webpage is iframe with numerical weather data, embedable into external webpages.

2.5 Visualization Subsystem

Main results of whole system - visualizations - are generated by visualization subsystem. This subsystem is based on GNUPLOT program to generate plots of measured and predicted data. As input of data we use hist.txt file to draw a current weather data plots. But also we draw predicted data. These data are combined with current weather data plots to visualize full time window of 72 hours back and 24 hours into future.

2.6 Web Subsystem

Webpage composition is very simple. All images and current data iframe are simply copied into static webpage template. This is done regularly every 15 minutes as defined by the timer subsystem. Remote copying is realized using simple SCP linux command. Webpage template includes simple code for automatic refresh of client window every 15 minutes.

3 Conclusion

Our experience with neural networks based local weather prediction and our experiments with the topic show promising results - neural networks sliding

window prediction can be used as an alternative or a complement to standard weather forecasts.

Achieved accuracies are technologically useful and beneficial, but the historical data for the training of predictor have to be several years long. We recommend 10 years or more for the predictor training.

Our predictor structure for one day ahead hourly prediction is based on seven meteorological variables inputs from past three days and encoded date/time information. We used shallow topology with single hidden layer.

We want to emphasize the availability of local weather prediction as an alternative and extension of regional weather forecasts. In the paper we provided all details necessary to build local weather prediction system with own meteostation, predictor and public web output. It shows relative affordability and simplicity of such system.

Acknowledgment

Research supported by the "Center of Competence of knowledge technologies for product system innovation in industry and service", with ITMS project number: 26220220155 for years 2012-2015.

References

1. S. Haykin, *Neural Networks (A Comprehensive Foundation)*, Macmillan, 1994.
2. B. Krose, and P. van der Smagt, *An Introduction to Neural Networks*, University of Amsterdam, 1996.
3. J. Koščák, R. Jakša, R. Sepeši, P. Sinčák, *Daily Temperature Profile Prediction for the District Heating Application*, in Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems, Springer, ISBN:978-3-319-00541-6, pp. 363-371, 2013.
4. J. Koščák, R. Jakša, P. Sinčák, *Prediction of Temperature Daily Profile by Stochastic Update of Backpropagation Through Time Algorithm*, in ISF 2011 Proceedings, International Symposium on Forecasting, International Institute of Forecasters, ISSN 1997-4124, 2011.
5. J. Adamčák, *Predikcia počasia s využitím neurónových sietí (in slovak, En.: Weather Forecast Using Neural Networks)*, Technical University Košice, 2012.
6. Con Tassios, *Davis Weather Tool - dwt.c*, version 1.0.1, FREE software licensed under the GNU General Public License (GPL) version 2, Swinburne University of Technology, Melbourne, 2008.
7. J. Adamčák, Project webpage, http://neuron.tuke.sk/adamcak/0/, 2012.