The Use of Artificial Neural Networks in Predicting Air Pollution in Cities – Hardware Implementation Issues

Marzena Banach, Zofia Długosz, Rafał Długosz and Tomasz Talaśka

Abstract— **The paper presents a concept of an air pollution monitoring system based on a wireless sensor network, in which particular sensors are able to predict future pollution levels within a specific time horizon. In the solutions reported in state-of-the art literature, forecasting the changes in pollution levels is usually based on the use of artificial neural networks (ANNs), globally trained for a given area, e.g. a city. In contrast, we propose to equip the sensors with hardware-implemented ANNs that could make the prediction based on data that the network obtains from the place where particular sensor is installed. Such networks require appropriate hardware optimization to maintain low energy consumption of the overall sensor. For this reason, in this paper we present a method of optimizing one of more complex components that is involved in determining the value of neuron weights in the network training process. In this work, we focus on the backpropagation (BP) learning algorithm, which results from the widespread use of this type of network in predicting air pollution levels.**

I. Introduction

One of significant problems in urban areas is a growing number of traffic-related emission [1], which may harm air quality and impact human health. Among air pollution emitted by vehicles are volatile organic compounds (NMVOC), particulate matter (PM), sulphur oxides (SOx), nitrogen oxides (NOx), ammonia (NH_3), non-methane volatile organic compounds (NMVOC) and others [2].

From a technical point of view, the process of pollution monitoring is a relatively uncomplicated task. It needs a sensor network installed in a given area. The sensors operate in parallel, measuring levels of selected pollutants and other parameters, and send the readings to a base station that process them and creates a pollution map, actualized in specified time intervals. Such solutions are already offered by a growing number of companies. They include, for example, Airly [3], LookO2 [4] and Luftdaten [5].

M. Banach in with Poznan University of Technology, Institute of Architecture and Spatial Planning, Nieszawska 13C, 61-021, Poznan, Poland; E-mail: marzena.banach@put.poznan.pl

Z. Długosz, R. Długosz and T. Talaśka are with Bydgoszcz University of Science and Technology, Faculty of Telecommunication, Computer Science and Electrical Engineering, ul. Kaliskiego 7, 85-796, Bydgoszcz, Poland, E-mail: zosia.dlugosz@gmail.com, rafal.dlugosz@gmail.com, tomasz.talaska@gmail.com

R. Długosz is with Aptiv Services Poland, Podgórki Tynieckie 2, 30-399, Kraków, Poland E-mail: rafal.dlugosz@aptiv.com

A substantially different problem than the one described comes with the ability to reliably predict pollutant levels over a given time horizon. Such information is particularly crucial for city users who are potentially the most affected by air pollution (cyclists, pedestrians, etc.). Based on a short-term prediction map of pollution levels, they can optimize their way through the city, taking into account health issues.

A. Related works

Various techniques are used for the estimation of air pollutant emissions in cities [6], [7]. However, artificial neural networks (ANNs) are frequently used tools in this task. There is a growing interest in recent years on the use of ANNs in predicting and forecasting ambient air pollution. In [8], for example, an application of the BP ANN was proposed for the forecasting a daily mean of the PM_{10} pollutant one day ahead. The [9] proposes the use of BP ANN in the forecasting of the concentration of PM_{10}, SO_2 and CO pollutants, on the basis of seven input parameters, mostly the weather ones. Similar systems based on Multi-Layer Perceptrons Backpropagation Artificial Neural Network (BP ANN) are also proposed in [10] and [11].

In the state-of the art solutions, briefly described above, the learning data sets used to train the ANN are in most cases composed of major meteorological factors, that include temperature (T), relative humidity (RH), wind speed (WS), wind direction (WD), as well as selected levels of air pollution factors (typically PM_{10}). In reality, additional various factors related to urban development in cities play a crucial role. In [12], for example, selected factors related to local urban conditions, such as street width and building height were included in the learning data set.

The paper is organized as follows. In next section we highlight problems that may affect prediction quality. As we observe, the prediction results may substantially differ even for similar values of the weather conditions frequently used. This is influenced by local conditions in particular locations in the cities (microclimate, urban development, etc.). For this reason, in the following section we present a contribution to systems, in which hardware implemented ANNs integrated with particular wireless sensors distributed over the city are trained

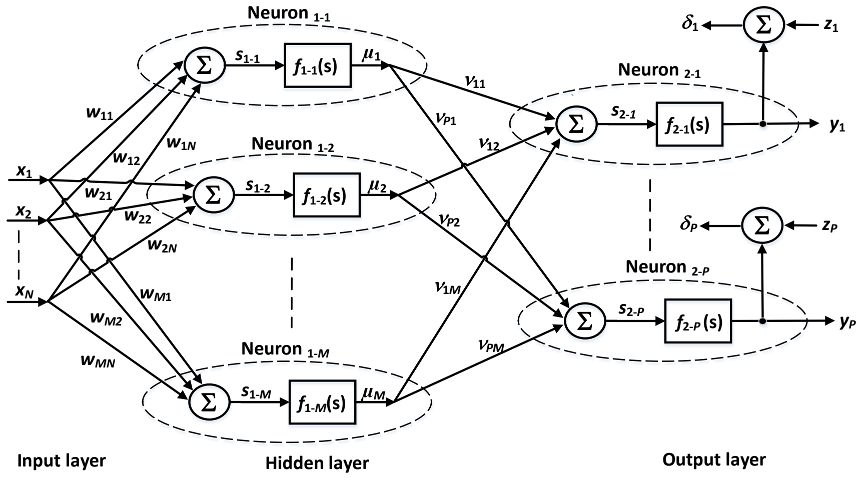


Fig. 1. Block diagram of the Multi-Layer Perceptrons Backpropagation Artificial Neural Network (with a single hidden layer).

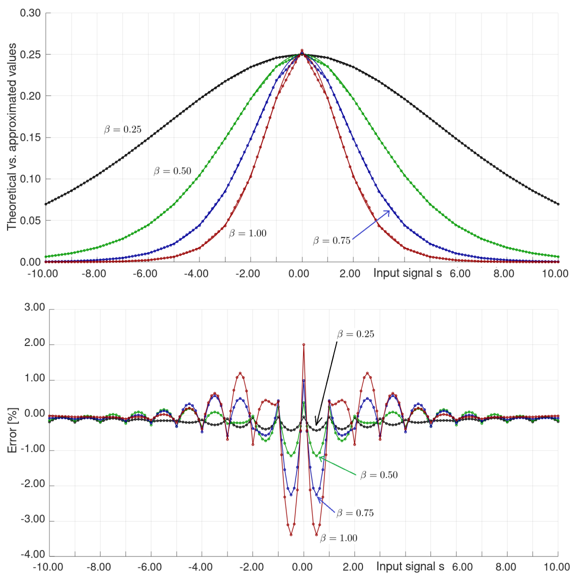
(a)
(b)


Fig. 2. Illustration of: (a) an approximation of the $f(s)\cdot[1-f(s)]$ term with the use of polylines, (b) a resultant error between the theoretical and the approximated values, for selected values of the β factor. The results are shown for an 8-bit signal s and address to the LUT stored on 5 MSBs.

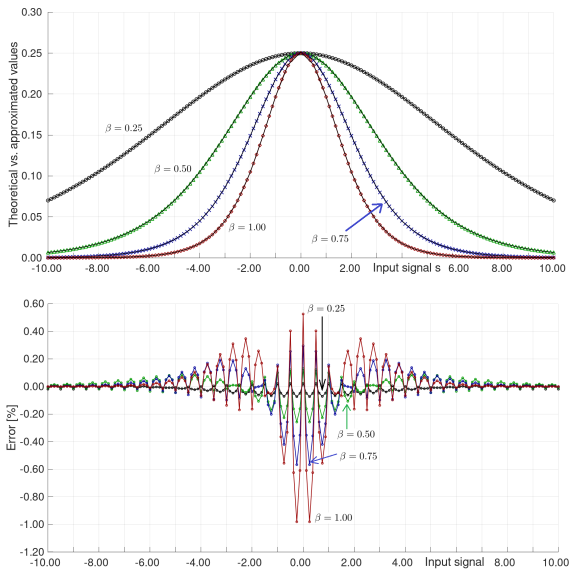
(a)
(b)


Fig. 3. Results similar to the ones shown in Fig. 2 for an 8-bit signal s and address to the LUT stored on 6 MSBs.

locally in the city. This is a significant difference from what is being proposed in comparison with other state-of-the-art works. As a result, the ANNs being components of particular wireless sensors do not need to communicate with the base station during the learning process. Since BP ANNs are frequently used in prediction of air pollution levels, in following Section we provide some ideas how to simplify its implementation in hardware. Finally, the conclusions are drawn in last Section.

II. Impact of environmental conditions on prediction results and proposed distributed approach

It is worth to notice that for similar values of T, RH, WS, WD, PM_{10} parameters but for different environmental conditions, predicted pollution levels may differ significantly. Selected factors of this type are briefly characterized below.

Geographical location of the city, e.g. by the river / sea, climate zone, etc.

Density and height of buildings – so-called aerodynamic roughness of the terrain – ventilation abilities. The wind speed in the city is on average lower by 20 %

at night and 30 % in the day. There may be also an increase in the wind speed in the city – the so-called tunnel effect (air flow in accordance with the street route).

The phenomenon of so-called urban heat island – the air above the city center is warmer than outside the city, hence as lighter it rises and causes local pressure reduction causing air suction from the areas surrounding the city.

Building geometry has an impact on the degree of inhibition of natural air movement in the city. This factor most strongly affects the nature of aerodynamic phenomena in cities, but it is difficult to predict.

Roads and streets which, due to their function and the vehicles moving on them, make up the sources of pollution, they also depend on the type of surface of the space (problem of possible sealing of the ground).

The specificity of tightly built-up urban interiors – streets or squares / courtyards where there may be a phenomenon of air stagnation or air circulation around without replacing it, especially when the width between buildings is less than 1.5 times the height. It is about the so-called Local Climate Zones LCZ, which were selected and developed for selected cities in Sri Lanca [13].

All these factors not only vary daily, not only in different spaces (local zones), but also depending on the time of day or season. In some seasons of the year the relative temperature difference between the city center and its outskirts increases, which results from heating apartments. Buildings in the center are sometimes taller and more dense (in accordance with the recommended "compact city"), and its organization and formation has a direct impact on local conditions and air quality [14].

The functions often arranged in it and the transport network necessary for operation are one of the main sources of air pollution in the city. However, the aforementioned development of buildings in the so-called down-town contributes to the "stopping" of these harmful substances, additionally hindering their removal (blocking the ventilation of cities).

To sum up, within the structure of a modern city it is difficult to find a small group of representative urban zones, in which a model prediction process could be carried out. In practice, each sensor in the city responsible for prediction should be trained on the basis of an individual data set, consisting of a specific vector of the input signals as well as a desired response of the ANN. Current solutions are able to provide only average responses in a specific area. Taking this into account, we propose a solution in which particular wireless sensors are equipped with ANNs trained individually.

III. Implementation and Results

In this Section we focus on one of the most critical block of the BP ANN, when looking from the hardware implementation point of view. First we provide an overview of the math standing behind such ANNs, pointing on the aspects that are the subject of the investigations presented in this paper.

An example block diagram of the BP ANN, with N inputs, a single hidden layer composed of M neurons and the output layer composed of P neurons is shown in Fig. 1. In the learning process of this type of NN particular iterations of the learning process are divided into two phases. In one of them the network error δ is computed on the basis of the input data X. This error is computed separately for each neuron in the output layer. This error is defined as a difference between an expected value, d, and the output signal from a given output neuron, y.

The described computation chain is the error backpropagation core procedure. In the next step, on the basis of these errors the ANN performs the adaptation of the weights of own neurons, according to formulas 1 and 2, for the first and the second layer, respectively. These formulas are given for selected neurons Neuron_{1-1} and Neuron_{2-1}, for $n = 1 \dots N$ and $m = 1 \dots M$, as follows:

$$w_{1n}^{k+1} = w_{11}^{k} + \eta \cdot \delta_{11}^{k} \cdot x_{n}^{k} \cdot \frac{df_{11}(s)}{ds} \tag{1}$$

$$\vartheta_{1m}^{k+1} = \vartheta_{1m}^{k} + \eta \cdot \delta_{21}^{k} \cdot \mu_{m}^{k} \cdot \frac{df_{21}(s)}{ds} \tag{2}$$

In the formulas above, η is the learning rate. It is set to the value close to 1 at the beginning of the learning process, and then it diminishes toward 0. When it reaches 0, the learning process is completed.

In the proposed approach sigmoid activation function, f, has been applied based on a hardware implementation described in [15]. This function is described as:

$$f(s) = \frac{1}{1 + e^{-\beta \cdot s}} \tag{3}$$

The derivative of this function, $\frac{df(s)}{ds}$, present in Eqs. 1 and 2 can be expressed as follows:

$$\frac{df(s)}{ds} = \beta \cdot f(s) \cdot [1 - f(s)] \tag{4}$$

Taking the 4 into account, Eq. 1 may be expressed as 5, while 2 as 6, below.

$$w_{11}^{k+1} = w_{11}^{k} + \eta \cdot \delta_{11}^{k} \cdot x_{1}^{k} \cdot \beta \cdot f(s) \cdot [1 - f(s)] \tag{5}$$

$$\vartheta_{11}^{k+1} = \vartheta_{11}^{k} + \eta \cdot \delta_{21}^{k} \cdot \mu_{1}^{k} \cdot \beta \cdot f(s) \cdot [1 - f(s)] \tag{6}$$

An interesting aspect is the way of implementation of the sigmoid activation function in equations above. In fact, the overall term $f(s) \cdot [1 - f(s)]$ can be determined in a single step. In the proposed approach this term is approximated with the use of a polyline, composed of line segments with different slopes and offsets (in vertical axis), stored in a look-up table (LUT).

The use of the proposed approach is illustrated in Figs. 2 and 3 for selected values of the β factor. The input signal s is a multi-bit signal, with the number of bits being one of important parameters here. Another parameter is the size of the LUT, which is equals to the number of the slots of the input domain. In the proposed implementation, the number of slots is always one of the powers of 2, which simplifies the hardware implementation. Fig. 2 presents selected results for the input signal, s, stored on 8 bits (256 values). The domain of the input signal is in this case divided into 32 slots, i.e. 5 most significant bits (MSB) of this signal are used as a pointer to the LUT. The remaining 3 bits (8 values) are used as offsets within particular slots. Fig. 3 presents similar results, however for the size of the LUT increased to 6 MSBs. In this case, the remaining 2 bits (4 values) are used as offsets within particular slots.

The values of the slopes and the offsets used to generate the waveforms in Fig. 2 are computed using a program written by us that additionally allows to minimize the error between the theoretical and the approximated waveforms. The values of these factors are represented by fixed-point numbers, in order to simplify the multiplication and summing operations in hardware. In the presented example, the largest slope equals 1023. To enable a direct comparison with a theoretical waveform, in Fig. 2 we have normalized the approximated waveforms so that to fit it to the theoretical waveform. In real implementation the normalization may be performed by division operation, by a factor which is one of the powers of 2. Selecting such values allows to realize the division by a simple operation of bit shifting. Since the waveforms in Fig. 2 are symmetric, the size of the LUT may be reduced by half. Our investigations with an example BP ANN (software model) show that the errors at the levels visible in Fig. 2 (b) do not affect the learning process. The visible error can be further reduced, by increasing the number of slots, as shown in Fig. 3 (b).

IV. Conclusions

In the paper we presented a method of implementing in hardware a non-linear activation function of neurons. In theory, this function requires complex mathematical operations, such as divisions and calculating an exponential function. In the conventional approach, in addition, there are many multiplication operations of floating point numbers, including the signal taken from the output of the activation function. Analyzing this function, and the way it is used in determining the weights of the neurons, we noticed that many complex operations can be replaced by simple operations that include multiplication and addition. The proposed method allows for reducing the number of multiplication operations. The proposed solution requires the use of a LUT of a small size. The LUT stores the values of the slopes and offsets for the function approximating the activation function. These values are computed earlier before staring the learning process of the ANN. The advantage of the proposed solution is that fixed-point numbers may be used. This significantly simplifies the implementation of the overall neural network.

References

[1] Liaquat, A.M., Kalam, M.A., Masjuki, H.H., Jayed, M.H., "Potential emissions reduction in road transport sector using biofuel in developing countries", *Atmospheric Environment*, Vol. 44, Iss. 22, 2010, pp. 3869-3877.

[2] Amato, F., Cassee, F.R., Denier van der Gon, H.A.C., et al., "Urban air quality: The challenge of traffic non-exhaust emissions", *Journal of Hazardous Materials*, Vol. 275, 2014, pp. 31-36.

[3] Airly net of air pollution sensors: `https://airly.eu/en/`

[4] LookO2 net of air pollution sensors: `https://www.looko2.com/heatmap.php`

[5] Luftdaten net of air pollution sensors: `https://luftdaten.info/en/home-en/`

[6] Stamenković, L.J., Antanasijević, D.Z., Ristić, M.Đ., et al., "Estimation of NMVOC emissions using artificial neural networks and economical and sustainability indicators as inputs", *Environmental Science and Pollution Research*, Vol. 23, 2016, pp.10753-10762.

[7] Stamenković, L.J., Antanasijević, D.Z., Ristić, M.Đ., et al., "Prediction of nitrogen oxides emissions at the national level based on optimized artificial neural network model", *Air Quality, Atmosphere & Health*, Vol. 10 (1), (2017), pp.15-23

[8] Hooyberghs, J., Mensink, C., Dumont, G., Fierens, F., Brasseur, O., "A neural network forecast for daily average PM_{10} concentrations in Belgium", *Atmospheric Environment*, Vol. 39, Iss. 18, 2005, pp. 3279-3289.

[9] Kurt, A., Gulbagci, B., Karaca, F., Alagha, O., "An online air pollution forecasting system using neural networks".*Environment International*, Vol. 34(5), 2008, pp. 592-598.

[10] Dedovic, M. M., "Forecasting PM_{10} concentrations using neural networks and system for improving air quality". *International Symposium on Telecommunications* (BIHTEL), Sarajevo, 2016, pp.1-6.

[11] Biancofiore, F., Busilacchio, M., Verdecchia, M. "Recursive neural network model for analysis and forecast of PM_{10} and $PM_{2.5}$", *Atmosphere Pollution Research*, Vol. 8, 2016, pp. 652-659.

[12] Fang, M., Zhu, G., Zheng, X., Yin, Z., "Study on air fine particles pollution prediction of main traffic route using artificial neural network", *International Conference on Computer Distributed Control and Intelligent Environmental Monitoring*, CDCIEM 2011, pp. 1346-1349

[13] Perera N., Emmanuel R., Mahanama P.K.S., "Projected urban development, changing 'Local Climate Zones' and relative warming effects in Colombo, Sri Lanka", *Proceedings of International conference on Cities, People and Places*, Colombo, Sri Lanka, 2013, ISSN: 2345-9530

[14] Chao Li, Zhanyong Wang, Bai Li, Zhong-Ren Peng, Qingyan Fu, "Investigating the relationship between air pollution variation and urban form", *Building and Environment*, Vol. 147, 2019, pp.559-568.

[15] Z. Długosz, R. Długosz, "Nonlinear Activation Functions for Artificial Neural Networks Realized in Hardware", *International Conference Mixed Design of Integrated Circuits and Systems* (MIXDES), Gdynia, Poland, 2018, pp.381-384.