Calibration of Low-Cost Particulate Matter Sensors with Elastic Weight Consolidation (EWC) as an Incremental Deep Learning Method

Rainer Schlund[1], Johannes Riesterer[1], Marcel Köpke[1], Michal Kowalski[2], Paul Tremper[1], Matthias Budde[1(✉)], and Michael Beigl[1]

[1] TECO/Pervasive Computing Systems, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
budde@teco.edu

[2] Helmholtz Zentrum München, German Research Center for Environmental Health (HMGU), Neuherberg, Germany
https://www.teco.edu

Abstract. Urban air quality is an important problem of our time. Due to their high costs and therefore low spacial density, high precision monitoring stations cannot capture the temporal and spatial dynamics in the urban atmosphere, low-cost sensors must be used to setup dense measurement grids. However, low-cost sensors are imprecise, biased and susceptible to environmental influences. While neural networks have been explored for their calibration, issues include the amount of data needed for training, requiring sensors to be co-located with reference stations for extensive periods of time. Also re-calibrating them with new data can lead to catastrophic forgetting. We propose using Elastic Weight Consolidation (EWC) as an incremental calibration method. By exploiting the Fisher-Information-Matrix it enables the network to compensate for different sources of error, both pertaining to the sensor itself, as well as caused by varying environmental conditions. Models are pre-calibrated with data of 40 h measurement on a low-cost SDS011 PM sensor and then re-calibrated on another SDS011 sensor. Our evaluation on 1.5 years of real world data shows that a model using EWC with a time period of data of 6 h for re-calibration is more precise than models without EWC, even those with longer re-calibration periods. This demonstrates that EWC is suitable for on-the-fly collaborative calibration of low-cost sensors.

Keywords: Incremental learning · Elastic Weight Consolidation · Sensor calibration · Particulate matter · Air quality

1 Introduction

Urban air quality has become one of the most important problems of our time. The air we breathe is directly related to the quality of people's lives. Particulate

S. Paiva et al. (Eds.): SmartCity360° 2020, LNICST 372, pp. 596–614, 2021.
https://doi.org/10.1007/978-3-030-76063-2_40

matter pollution is one of the largest risks to health worldwide and also contributes to environmental problems such as acid rain, ozone layer depletion and global climate change [23,30].

In industrialized countries, there is on average one official monitoring station per 100,000 inhabitants in cities, whereas in developing countries with high levels of air pollution there is one monitoring station per millions of residents [35]. These classic measurement grids are unsuitable to capture the spatial and temporal dynamics of air pollution in the urban atmosphere [27]. More than ten years ago, the first projects emerged that aimed to bridge this gap using mobile and/or low-cost sensors: the *MESSAGE* project [29,33], *Common Sense* [11], or *OpenSense* [16,20] to name a few.

However, low-cost sensors have some inherent disadvantages and challenges that still have not been adequately addressed. They are generally much more sensitive to changes of environmental conditions such as temperature, wind, humidity and others. They often exhibit sensor drift and may lose sensitivity over time due to aging of their components and therefore need to be re-calibrated frequently [6,8,16]. An established approach to mitigate systematic errors is calibration. Neural networks can be trained to minimize deviations in comparison to reference instruments [23]. The disadvantage of neural networks is the large amount of data required for training in order to later yield sufficient quality [13], which usually means a long time span for data acquisition. This is especially true for Particulate Matter (PM) sensing, as many different environmental factors influence measurement [5]. Additionally, individual low-cost sensors often a-priori exhibit significant inconsistencies between one another [5]. Therefore, learned models cannot simply be transferred from one sensor to another. Instead, each sensor must be calibrated individually [7] over a long period of time to learn how it is influenced by meteorological factors like wind, temperature, humidity, etc., which strongly limits practical application.

This paper presents an approach that uses Elastic Weight Consolidation (EWC) in order to mitigate this limitation. EWC is an algorithm for the training of neural networks which enables them to learn similar tasks incrementally. The calibration of different individual sensors can be considered as a sequence of similar tasks. We show that neural networks can be trained incrementally with the EWC algorithm in order to transfer their calibration to further sensors. By transferring the calibration from one sensor to others the required amount of data and the length of training periods can be significantly reduced. Thus mitigating the limitations described above, when training each sensor individually.

2 Related Work

Spinelle et al. [36] compared the performance of different methods for field calibration of O_3 and NO_2 sensors and found that simple methods such as linear and multivariate linear regression were sufficient for the O_3 sensors, but an artificial neural network yielded better results for NO_2. They included wind and air pressure in addition to temperature and humidity. The neural network was trained with the hourly data of one week (168 data sets).

Yamamoto et al. [37] calibrated low-cost temperature sensors installed at three different locations, but they were not in the immediate vicinity of the references. For the experiment, hourly recorded data were collected over a period of 305 days. A neural net with a hidden layer was used, which had as input parameters the measured temperature, the radiation value of the sun, the humidity, the azimuth value and the altitude above sea level. In the evaluation the neural network shows clear advantages over a simple linear regression and is even able to compensate for measurements influenced by direct solar radiation.

Hojaiji et al. [17] describe the calibration of a fine dust sensor under artificial conditions simulating indoor and outdoor climate. They achieve good results with a simple correction algorithm, which is supposed to compensate influences of temperature and humidity. However, the algorithm was only tested over a period of one day, so that no statements can be made about the long-term behaviour. Also how the calibration behaves under real outdoor conditions has not been investigated.

A calibration using a sensor array for ozone is described in [1]. The array is calculated to a virtual calibrated sensor using linear regression. The influences of temperature and humidity were also taken into account. After removing outliers, 450 hourly measurements from a period of 3–4 weeks were used for calibration. Since some nonlinearities are shown in scatterplots, it is pointed out under Future work, that the root of the mean square error (RMSE) can possibly be further improved with the use of nonlinear calibration models, to which neural networks belong.

In [8] the sensors were first calibrated in the vicinity of a reference instrument. Various Matlab functions for nonlinear regression were used, which processed about 13000 data recorded in minute intervals. The calibrated sensor was then validated at another location over a period of several months. However, the validation was only performed with an hourly resolution, but showed good performance compared to the reference.

The ability of a system to transfer knowledge and acquired skills from one task to another is called transfer learning. Traditional machine learning methods learn tasks always anew without relying on existing knowledge. Transfer learning techniques, on the other hand, attempt to transfer what has been learned from one task or domain to another, assuming the tasks have similarities between each other [22]. Prahm et al. [34] show that thigh prostheses can be controlled by simple myoelectric signals from muscle groups that are still present. However, this control is susceptible to electrode displacement, sweat, fatigue and other influences. This means that continuous recalibration of the control signals is necessary. The prostheses are first trained in an interference-free system. In the application they are then incrementally recalibrated daily with small data sets of less than one minute, based on the initial training, by transfer learning.

In recent years some new learning techniques by transferring known knowledge using neural networks have been developed. One of them is the Elastic Weight Consolidation Algorithm [18]. This algorithm is the foundation of this work.

In order to effectively build sensor networks, [24] proposes an approach using a cross sensor calibration based on sensor rendezvous. If two sensors fall below a certain distance from each other, a calibration process is initiated provided that one of the two sensors has sufficient validity for its measurement data. Various regression methods are used, for which neural networks could be a potential alternative.

Cheng et al. [9] follow a cloud-based approach in which the network's sensors send their data to a central database via the Internet. From there, the data is calibrated using a neural network and can then be transferred to apps for health care, for example.

3 Design

An aspect of neural networks is their ability to learn different tasks by training them simultaneously. This leads to problems if not all tasks are already previously known [18,31]. Sequential learning can be achieved if the data of a training session is stored in an episodic memory and presented to the network again for the next task. However, the data from the memory must be presented to the network again in each training run [19,25].

There is evidence that biological systems anchor what they have learned by consolidating the synapses that are essential for a learned task, which means that they are less plastic and therefore remain stable over longer periods of time [2,12]. Following theses insights, the Elastic Weight Consolidation (EWC) algorithm developed by Kirkpatrick et al. [18] consolidates the weights which are important for a specific task. For a neural network, there is not only one optimal configuration of the weights, but many that lead to comparable results. This set of weights can be viewed as a subset of the space of all possible weights. This suggests that for two similar tasks A and B, such as calibrating two sensors, there are weight distributions θ_A and θ_B that are similar and suitable for both. The EWC algorithm forces the development of the weights in training for task B into the (if available) intersection $\theta_A \cap \theta_B$, shown schematically in Fig. 1. After training, the newly calculated weights are therefore in an area that performs optimally for both tasks A and B [18].

3.1 Mathematical Foundations of the EWC-Algorithmus

Looking at neural networks from a probability theory point of view, a training is the search for the weights that best describe a given data set $\mathcal{D}$. In the case of a regression this means that the expected value of the unknown distribution $p(\theta|\mathcal{D})$ is a minimum for a loss function [18,28]. For this conditional distribution, Bayes theorem in logarithmic form can be applied:

$$\log p(\theta|\mathcal{D}) = \log p(\mathcal{D}|\theta) + \log p(\theta) - \log p(\mathcal{D}) \tag{1}$$

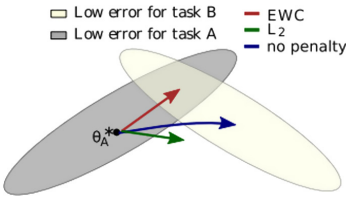


Fig. 1. The EWC algorithm ensures that the weights learned in task B are still suitable for task A by rewarding movements in the direction of the red arrow. The green arrow shows that uniform forcing in any direction is too restrictive. The blue arrow shows the development of the weights without EWC [18]. (Color figure online)

So if one wants to find the optimal weights for two consecutive tasks, then $\mathcal{D} = \mathcal{D}_A \cup \mathcal{D}_B$ with $\mathcal{D}_A \cap \mathcal{D}_B = \emptyset$ and we compute using the log version of Bayes theorem:

$$\begin{aligned}
\log p(\theta|\mathcal{D}) &= \log p(\theta|\mathcal{D}_A \cup \mathcal{D}_B) \\
&= \log p(\mathcal{D}_A \cup \mathcal{D}_B|\theta) + \log p(\theta) - \log p(\mathcal{D}_A \cup \mathcal{D}_B) \\
&= \log(p(\mathcal{D}_A|\theta) * p(\mathcal{D}_B|\theta)) + \log p(\theta) - \log(p(\mathcal{D}_A) * p(\mathcal{D}_B)) \\
&= \log p(\mathcal{D}_B|\theta) - \log(p(\mathcal{D}_B)) + \underbrace{\log p(\mathcal{D}_A|\theta) + \log p(\theta) - \log(p(\mathcal{D}_A))}_{=\log p(\theta|\mathcal{D}_A)} \\
&= \log p(\mathcal{D}_B|\theta) + \log p(\theta|\mathcal{D}_A) - \log(p(\mathcal{D}_B))
\end{aligned} \tag{2}$$

The distribution for the model of task A is the only term which dependents on $\mathcal{D}_A$ is. This means that all information about task A is contained in this distribution. Assuming that some weights θ_A^* are close to the optimal weights, they are a good estimator for the expected value of $\log p\,(\theta|\mathcal{D}_A)$. An estimator for a parameter, on the other hand, is the better the smaller the variance of its distribution function is. In the example of the normal distribution, a small variance leads to a narrower and higher bell curve, so the interval for values with high probability becomes narrower. What the EWC algorithm wants to achieve is that only weights that are good estimators for task A are chosen for task B [18].

The a posteriori distribution for task A can be estimated by a Laplace approximation

$$p\,(\theta|D_A) = \frac{1}{Z} e^{-E(\theta)}, \tag{3}$$

where $E(\theta) = \log p\,(\theta, D_A)$ is called energy function with $Z = p(\mathcal{D}_A)$ [28]. If the Taylor series of the energy function around some weights θ_A^* is formed up to the second degree, the first gradient of the approximation is zero, since θ_A^* is a minimum for task A and we get:

$$p\,(\theta|\mathcal{D}_A) = e^{-E(\theta_A^*)} e^{\left[-\frac{1}{2}(\theta-\theta_A^*)^T H(\theta-\theta_A^*)\right]} \tag{4}$$

where H is the Hesse matrix of the energy function. The negative value of H is known as the observed information matrix F and its expected value $\mathbb{E}(F)$ is

known as the Fisher matrix [28]. If we assume $e^{-E(\theta_A^*)}$ to be of constant value λ, which will be the case if task A has already been trained, and apply the logarithm to Eq. (4), we get:

$$\log p\left(\theta|\mathcal{D}\right) = \log p\left(\mathcal{D}_B|\theta\right) - \frac{\lambda}{2}\left(\theta - \theta_A^*\right)^T F_{\theta_A^*}\left(\theta - \theta_A^*\right) + C \tag{5}$$

where $C := -\log(p(\mathcal{D}_B)$ is constant with respect to the weights. To reduce the computational effort for computing the Fisher matrix, it is assumed to be a diagonal matrix (which also might be achieved by a change of basis) and in this case Eq. (5) simplifies to

$$\log p\left(\theta|\mathcal{D}\right) = \log p\left(\mathcal{D}_B|\theta\right) - \frac{\lambda}{2} F_{\theta_A^*}\left(\theta - \theta_A^*\right)^2 + C. \tag{6}$$

The Fisher matrix has two interesting properties. It can be used to assess the quality of an estimator and the inner product $\left(\theta - \theta_A^*\right)^T F_{\theta_A^*}\left(\theta - \theta_A^*\right)$ defines a metric that can be used to calculate a local distance between estimators [32]. In this metric, all optimal estimators for task A have a small distance between them. If the training for task B results in an estimator becoming less optimal for task A, the distance between these two estimators increases. This distance calculated with $\frac{\lambda}{2} F_{\theta_A^*}\left(\theta - \theta_A^*\right)^2$ is therefore suitable as a penalty term within the loss function of task B defined by

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \frac{\lambda}{2} F_{\theta_A^*}(\theta - \theta_A^*)^2. \tag{7}$$

$\frac{\lambda}{2}$ is thereby a proportionality factor (EWC-factor), which is trained as a hyperparameter during training and determines how large the resistance is, for a change of the weights determined by the first task [28] and [18,21].

A neural network is described by the output function $t = y\left(x, \theta\right)$ with $x \in \mathcal{D}_A$. In case of a linear activation function, for the distribution $p(\mathbf{t}|\mathbf{x})$ $\mathbf{x} \in D_A$ one can assume a normal distribution centered around $y(x, \theta)$ with a variance β^2 [18]. This distribution coincides with the distribution of $p(\theta|\mathcal{D}_A)$, since $p(t|x) = p(\theta|x)$. Assuming t_i to be conditionally independent we conclude:

$$p_\theta\left(t|x\right) = \prod_{i=1}^{o} N\left(t_i|y(\mathbf{x}, \theta)_i, \beta^2\right) \tag{8}$$

The definition of the Fisher matrix is:

$$I(\theta) := \mathrm{E}_\theta(S_\theta^2) \tag{9}$$

In this case the scoring function S_θ is the first derivative of the log-likelihood function of the normal distribution in 8. Thus, following [32] the Fisher can be computed by:

$$F = \beta^2 \mathbb{E}\left[\sum_{i=1}^{0}\left(\frac{\partial y_i}{\partial \theta}\right)^T \left(\frac{\partial y_i}{\partial \theta}\right)\right] \tag{10}$$

The Fisher matrix is thus a diagonal matrix whose diagonal contains the sum of the derivatives of the output function of the net $y(x, \theta)$ with respect to all data points. The Fisher matrix describes an expected value. Since the average of the results obtained from a large number of trials approaches the mean value, the matrix is calculated by taking the mean value for a selected set of n data points from $\mathcal{D}_A$.

3.2 Construction of the Neural Networks

Deeper neural networks have a higher parameter efficiency, i.e. they can model complex functions with exponentially fewer neurons than flat ones and can therefore be trained more quickly [13]. Hence a simple backpropagation net with seven input parameters, three hidden layers and one output layer with one neuron was chosen as model with a linear activation function as output layer. But as these configuration is not chosen by profound experience, this must be further evaluated in future work. The input parameters are the sensor reading in $\mu g/m^3$, the temperature in degrees Celsius, the relative humidity in percent, the wind speed in meters per second, the precipitation in millimeters, the air pressure in millibar and the day of the year (1-365). As minimization function the mean squared error (MSE) was used because it' higher sensitive to outliers (unusually high or low values) [26]. Hyper parameters of the network are number of neurons per layer, activation function per layer, optimization function, learning rate, batch size. The values of all parameters were deliberately restricted as little as possible, since there is hardly any experience to date in the selection of hyperparameters with regard to the EWC algorithm for a regression task. The number of neurons is selected from the interval $[10, 1000]$. The following activation functions are available: Sigmoid function, Rectified linear unit ($Relu$), Leaky_Relu, Exponential linear unit (Elu) and Tangent hyperbolic function. The selection of optimization functions consists of Adam, Adagrad and RMSProp. The learning rate can take values between 0.0001 and 0.2, the batch size values from 8 to 128.

3.3 Bayesian Optimization

For a defined model, the Bayesian algorithm searches exploratively and exploitatively for the best possible values for the hyperparameters which minimize the loss function for training runs on the data [3]. To limit the time required for the search, the model is trained with a smaller number of epochs using the MSE as a loss function. The relationship between exploitation and exploration is determined by the so-called acquisition functions. The one that was chosen (UCB) is the one that works with the upper limit of the confidence interval for the prediction. The ratio between exploitation and exploration is determined with the parameter Kappa, which can have values from 0.1 to 10. A mean value of 1.0 was chosen for this parameter. Other parameters that are set are the number of randomly selected points used as a starting point for the exploration and the

number of iterations of a search. In total five optimization runs with different parameter combinations were performed.[1]

3.4 Trial Procedure

The entire test series consists of three consecutive steps (see also Table 1):

Bayesian Optimization. In this step a pre-selection of models suitable for further training is made. A total of 252 different models have been generated in five runs. For each sensor/data combination, the six models with the lowest MSE value were included in the second step.

Pre-selection. The nets selected in the Bayesian optimization are trained in the pre-selection with a larger number of epochs and more repetitions on differently sized training data sets. On the basis of the ratio between the MSE for the training data, the MSE for the test data and the absolute values of the MSE for the test data, an estimation should be made of the minimum amount of data with which a successful training for the net can be performed. Of all the runs performed on all the training data, the three models with the lowest MSE are selected again for the final training with the EWC algorithm.

Final Training with EWC. Finally, the initial training and then the subsequent training are carried out for the networks with and without EWC according to the encoding Table 2. Before the nets are trained, they are evaluated on the test data from the sensors for the follow-up training to determine how well the predictions improve in the follow-up training.

All final training are carried out in the follow-up training with different amounts of data in order to estimate, as in the pre-selection, with which minimum amount of data a successful training for the networks can be carried out. These data are then used to evaluate the test. Table 1 shows an overview of the parameter values used for the training runs in the different test sections.

4 Evaluation

We evaluated our approach on the data of six NovaFitness SDS011 PM1 sensors, which were operated for more than 1.5 years under real world conditions on the roof on an air quality and weather measurement station (see Fig. 2). In laboratory tests (cmp. Budde et al. [5]), these sensors showed good linear behaviour, but with a systematic misinterpretation of the actual concentration. It was also shown that the measured values in certain areas are strongly dependent on environmental factors such as relative humidity. The SDS011 recorded data with a

[1] As basis for our implementation, we used the GitHub repositories https://github.com/fmfn/BayesianOptimization (Bayesian Optimization) and https://github.com/ariseff/overcoming-catastrophic (EWC algorithm).

Table 1. Overview of the value ranges of the training parameters

Bayesian search				
Run	Exploitation/exploration coefficient Kappa	Initial points	Iterations	Epochs
1	1	20	40	400
2	1	20	40	400
3	1	20	40	500
4	1	200	400	700
5	1	400	500	700
Backpropagation network				
Hidden layer 1		10–1000		
Hidden layer 2		10–1000		
Hidden layer 3		10–1000		
Activation function layer 1		sigmoid, tanh, elu, relu, leaky_relu		
Activation function layer 2		sigmoid, tanh, elu, relu, leaky_relu		
Activation function layer 3		sigmoid, tanh, elu, relu, leaky_relu		
Optimization functions		RMSProp, Adam, Adagrad		
Pre-selection and test for required data volume				
# selected models from bayesian optimization		6		
# datasets (1 dataset equals 74 min of data)		8, 16, 64, 200, 400, 600, all		
# epochs		500, 1500		
# training runs per model		8		
Final training				
# selected models from pre-selection		3		
# datasets for EWC algorithm		5, 20, 80, all		
# epochs audited		50, 100, 500, 800, 2000		
# epochs for training the initial sensor		1500		
# training runs per model		3		
EWC factor		150000		

frequency of 1 reading every 3 min in several separate periods between October 22nd, 2017 and April 1st, 2019. A total of approx. 100,000 usable data points, spanning all four seasons, was recorded. There was no removal of incorrect measured values or outliers. There are areas where the sensors underestimate as well as overestimate values, which indicates a nonlinear behaviour of the sensors.

A Grimm EDM180 particle counter was used as reference device [14]. Additionally, every minute one value measuring temperature, humidity, air pressure, precipitation and wind speed each was recorded. The day of the year was added to these values as a further date in order to be able to adjust for whether this seasonal information had an effect on the quality of the calibration.

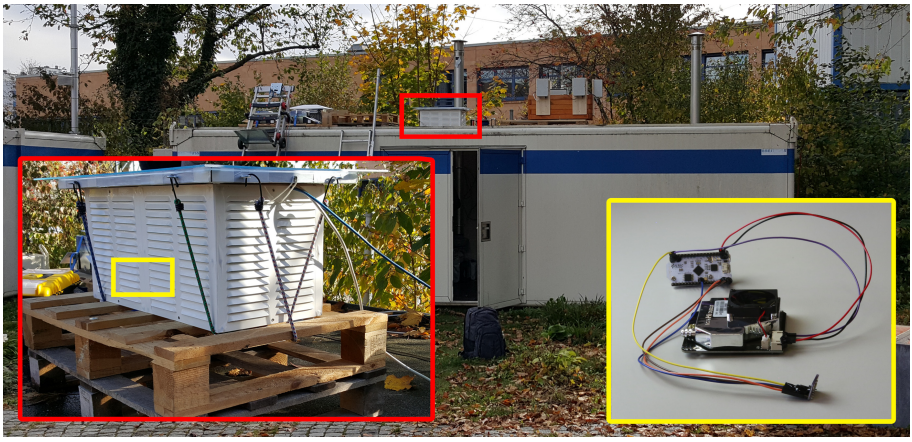


Fig. 2. Data collection setup: Six NodeMCU platforms with an SDS011 PM sensor and a BME280 temperature/humidity sensor (yellow rectangle) were operated for a period of 1.5 years. They were installed on top of an air quality measurement container in Augsburg, Germany (as part of the SmartAQnet project [4]) within a well-ventilated instrument shelter (red rectangle). (Color figure online)

Since the data points were collected at different frequencies, the values from the reference station were combined to form three-minute averages. For the precipitation data, the sum was calculated instead. In order to be able to assign the two time series to each other, the values of the reference values were replenished to minute cycles, so that three copies of each measured value were made. The two time series were merged again by a join operation. Since there was a small period of time in which the reference did not record any values, this was subsequently removed. At the end of the process, there is a measurement in the data again for every third minute, except in the periods in which the sensors did not measure.

Measurements were taken for the two particle sizes PM10 and PM2.5. Since neural networks often work better on normalized data all data sets of the SDS011 sensors have been normalized. No normalized files were generated from the reference values, in this case the output layer maps back to the original value range. In the Table 2 the color coding for the trained neural networks and the different data sets of the sensors is shown. The six sensors available for the experiment are divided into two groups. With the first group (sensors 1, 2 and 3) the nets are trained in the initial training without using the EWC (marked black in Table 2).

The nets are then trained with the data from the second group of sensors (sensors 4, 5 and 6) with and without EWC in a subsequent training (blue markings in Table 2). For the training, all data sets used have been split in the ratio $\frac{4}{9}$ training, $\frac{3}{9}$ test, and $\frac{2}{9}$ validation. The nets are first trained on training data and then evaluated with the test data. Good models with low mean square error are selected based on the results they produce on the test data. Since the

Table 2. Color coding of all data-sensor combinations of the neural networks trained in the experiments.

		Sensors		
		Initial Training	Follow-up Training without EWC	Follow-up Training with EWC
Data	PM10			
	PM2.5			

selection is made in several successive training runs, the trained models are no longer completely independent of the test data and have to be tested once on completely unknown data, the validation data set, in the end.

Before looking at the evaluation data, we present the mean squared error (MSE) of the raw (unprocessed) measurements of the six SDS011 sensors compared to the reference in Fig. 3. It is noticeable that the values for the particle size PM10 shown significantly higher MSE values and also a greater range of variation than the values for PM2.5. This is in line with findings from related work that showed that the SDS011 is much more susceptible to errors in the PM10 channel [5].

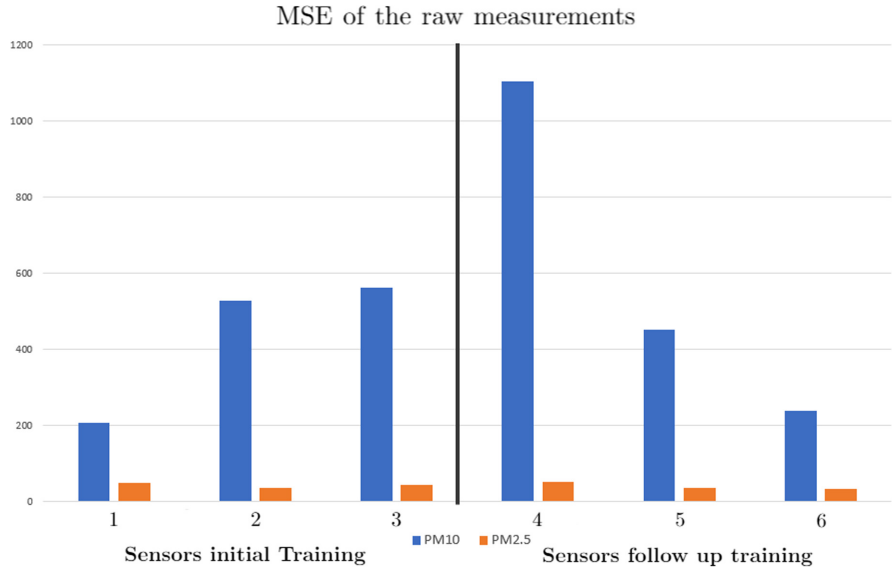


Fig. 3. Mean squared error (MSE) values for all sensors. MSE in $(\mu g/m^3)^2$

In total 7,290 models were trained. 2,430 of them were trained on initial sensors (sensors 1, 2, 3 marked black) without EWC (see column *Initial training* in Table 2). Originating from those all models were subsequently trained on follow-up sensors with and without EWC (columns *Follow-up Training without EWC* and *Follow-up Training with EWC* in Table 2 respectively).

4.1 Description of Evaluation Data

All following results are carried out as mean values aggregated over all sensors, since the basic properties of the EWC should not be dependent on a selection of sensors. In order to get an estimation of the amount of data needed for the training with EWC, we trained with different amount of data and also with different number of epochs. Furthermore, the evaluations is only presented for networks that meet a given quality standard. In Table 3, only nets with an MSE lower than one hundred are considered. The evaluation is divided into two sections.

1. The initial training . The trained nets are evaluated on:
   (a) the validation data of the initial training for PM10 and PM2.5
   (b) the test data of the sensors of the follow-up training for PM10 and PM2.5
2. The follow-up training for the networks initialized in 1. once with and once without EWC . The nets are evaluated in each case on:
   (a) the validation data of the follow-up training for PM10 and PM2.5
   (b) the validation data for initial training for PM10 and PM2.5 .

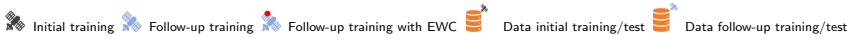
During training, the EWC tries to keep the weights in an optimal range for both tasks. To estimate how well the data of the initial training will be evaluated after the follow-up training, one has to compare the MSE from (1.a) with the MSE from (2.b). This shows how well the networks can continue to predict the measured values of the initial sensor after the follow-up training. (2.a) and (2.b) shows how well the nets can predict the measured values for both sensors. The comparison between 2. With and without EWC shows how well the nets trained with EWC perform compared to those without.

4.2 Validation

1. Networks trained with and without EWC: It is shown that the nets trained with EWC are able to make very good predictions for both sensors, with significantly better values than the nets trained without EWC (comparison of columns 2 and 3 respectively columns 4 and 5).
2. Prediction for initial sensor: Also only slight losses are shown in the comparison of the nets trained with EWC on both sensors compared to the nets trained only on the data of the initial sensor (columns 1 and 3).

Table 3. MSE for PM10 over all sensor pairs. Selection of the networks: MSE for follow-up training on initial data in the validation with (column3) or without (column 2) EWC less than 100. Models trained with EWC show clearly better results than those without for both evaluation data sets.

Data fraction	# epochs	training on / evaluation on				
5	50	63	82	76	71	64
	200	63	86	77	73	68
	500	60	87	79	74	67
	800	60	86	79	75	68
	1200	59	86	79	72	67
20	50	62	81	74	70	63
	200	61	81	74	63	60
	500	69	82	77	63	61
	800	61	80	75	66	64
	1200	62	81	76	64	63
80	50	61	83	74	66	60
	200	61	80	73	64	59
	500	63	78	75	60	58
	800	63	77	75	60	59
	1200	62	79	75	59	59



3. Amount of data: Looking at the results for the different data sets in columns 3 and 5, it can be seen that the results for five and ten data units respectively could be of good enough quality for sensor training. This corresponds to a period between 6 h and 12 h for the measurements. This value is still below the value in test section 2 Fig. 4 estimated value of 32 units = 38 h.

The same picture can be seen in Table 4 for the particle size PM2.5, for which the quality measure is an MSE smaller than forty.

To investigate whether there is a significant difference in the mean values of MSE in training with and without EWC on the validation data of the initial and the follow-up sensors, only models with a maximum MSE of 100 for PM10 are considered for both qualities, for PM2.5 the limit is 40.

Table 5 shows that the mean values of the MSE on nets trained with EWC are better for both particle sizes than without. The standard deviation and standard error are of a similar order of magnitude on the compared nets.

Table 6 shows the correlation between the nets trained with and without EWC. Finally, we need to check if the differences in the MSE are significant. Since this test involves dependent samples with parameterized data, Student's t-test for dependent samples is used for the test. A prerequisite for the test is a normal distribution of the MSE values, but this can be neglected for large data sets. What is given when the sample size is over 500 values for both PM2.5 and PM10 [10].

Table 4. MSE for PM2.5 over all sensor pairs. Selection of the networks: MSE for follow-up training on initial data in the validation with (column 3) or without (column 2) EWC less than 40. Models trained with EWC show clearly better results than those without for both evaluation data sets.

Data fraction	# epochs	training on / evaluation on					
5	50		16	21	16	23	18
	200		16	24	19	25	20
	500		16	25	20	25	20
	800		16	25	20	25	21
	1200		16	26	21	26	21
20	50		16	21	16	21	17
	200		16	21	17	21	18
	500		16	21	18	21	18
	800		16	21	18	21	18
	1200		16	22	18	21	19
80	50		16	20	16	20	17
	200		16	19	16	19	17
	500		16	18	16	17	16
	800		17	19	16	18	16
	1200		16	18	16	17	16

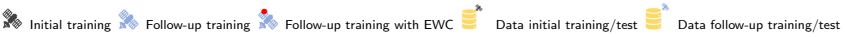


Table 5. Descriptive statistics on the nets trained with and without EWC on task 2 for the two particle sizes PM2.5 and PM10. The upper half shows the evaluation on the data of the initial sensors, the lower half on the data of the subsequent sensors.

Sensors	Class	Training	Validation	N	mean	stddev	stderr
	PM2.5			760	21.90	5.03	0.77
				760	18.09	3.40	0.66
	PM10			603	82.02	9.57	3.34
				603	76.04	9.38	3.10
	PM2.5			760	21.67	5.59	0.77
				760	18.45	4.36	0.66
	PM10			603	66.50	17.47	3.34
				603	62.82	16.04	3.10

The values in Table 6 confirm that the differences between the MSE values of nets trained with EWC on the data of the initial sensors (task 1) and the differences on the data of the subsequent sensors (task 2) are significant. This means that the nets trained with EWC are better able to calibrate the data of the initial sensors as well as the data of the subsequent sensors.

Table 6. Pearson correlation (PCC) and t-Test on the significance of the differences in the mean values of the MSE (each for the nets trained with and without EWC, for both particle sizes on the data of the initial and subsequent sensors). (* significance level 5%)

Sensors	Class	Training	Validation	Pearson correlation PCC	Pearson correlation Significance*	t-Tests Test statistics	t-Tests p-Value*
	PM2.5			0.69	<0.05	-28.85	<0.05
	PM10			0.69	<0.05	-19.80	<0.05
	PM2.5			0.75	<0.05	-24.36	<0.05
	PM10			0.93	<0.05	-14.10	<0.05

Initial training Follow-up training Follow-up training with EWC Data initial training/test Data follow-up training/test

No trends can be identified for the hyperparameters trained in the optimization. The values for the batch size are usually above 80, and there is no recognizable trend for the activation functions of the hidden layers and the learning rate. The number of neurons per layer is usually over 100, but there are also individual nets that manage with a total number of less than 100 neurons and provide good values. Also for the different optimizers no preference of a particular one can be recognized.

All in all, the optimization indicates that there are many different parameter combinations that are suitable for calibrating the sensors in the experimental setup chosen here. From the Bayesian optimization, the 24 best models per sensor were taken over into the preselection.

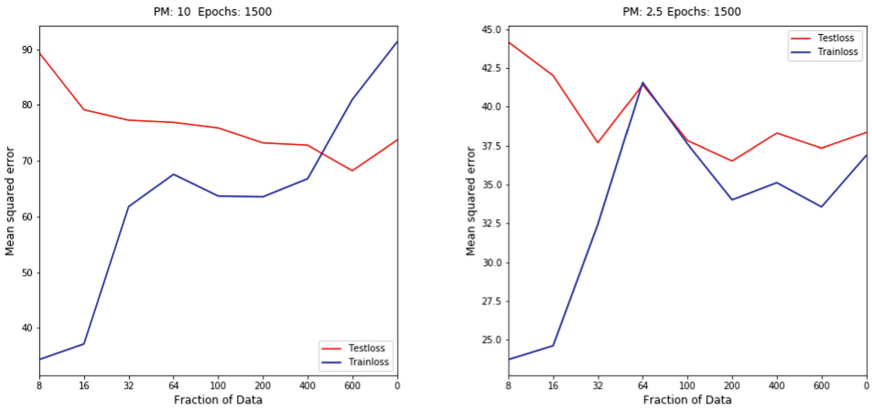
4.3 Preselection and Data Volume

Test runs that do not have a value better than 100 in at least one of the MSE values for test and training data, or a value greater than 300 in one of the two, were rejected as unsuccessful training runs. The test runs were performed with 1500 and 500 epochs. Since the results are almost identical for both, only the data for 1500 epochs are presented here.

To estimate the amount of data required, the mean value of the MSE for training was calculated for all sensors. The values are shown separately for the two particle sizes PM10 and PM2.5 in Fig. 4.

The curves for both particle sizes show a large difference between the MSE for the test data and the MSE for the training data up to 32 data units. This is an indication that the network is overtrained for these values. In addition, it can be seen that the MSE on the test data tends to improve further with larger amounts of data.

These are good indications that the smallest amount of data required is at least greater than 32 units. 32 units correspond to 768 measurements performed at a frequency of three minutes, i.e. about 38 h.



(a) Particle size PM10 (b) Particle size PM2.5

Fig. 4. Comparison of the development of the MSE for test- and training data, the data set 0 corresponds to a training with all data.

It is noticeable that the mean of the MSE for the training data of the particle size PM10 is significantly higher for the data units 600 and 0 (0 = all data) than for the test data, although for a mean value over all training runs it could rather be expected that the opposite case is realized. Investigations of the reference data have shown that some unusually high readings occurred in these areas that the network could not compensate for through training and explain this difference.

5 Conclusion and Future Work

In this work, we presented a method of transferring calibration models from one low-cost particle sensor to another. By exploiting Elastic Weight Consolidation (EWC), we demonstrated that this can be done using substantially less training data – and thereby time – than with previous approaches.

For this, we trained neural network based calibration models on a low cost sensor and subsequently continue to train them on a second one with and without using the EWC algorithm. In order to study the ability of the two approaches to learn the second calibration while recognizing the first, we evaluated various models on several pairs of sensors that collected data under real-world conditions for more than 1.5 years. It turned out that even with a small data set containing only 120 measured values (approx. 6h) for the second sensor, the training with EWC shows very good results and in comparison to training without EWC the MSE values are significantly better on both sensors.

The exploration of models in this work was quite restricted. For example, the EWC factor is not trained as a hyperparameter, but with a fixed value, since it has shown itself to be relatively insensitive to changes over a wide range of values in test runs. The number of hidden layers was also not optimized as a hyperparameter. The potential of a detailed optimization of the hyperparameters

could therefore still be investigated. Possibilities for improvement might also lie in the selection of a different net typology. LSTM nets, for example, are particularly suitable for time series, such as measurement data from sensors [13]. It might also be interesting to check whether an initial training of the networks on several instead of only one sensor can lead to further improvements.

In this paper was not examined, how far a sensor can be moved away from the reference station and still delivers reliable data and how long the measurements remain reliable over time. These are points that should be addressed in future work.

The original work on the EWC itself sees room for improvement by using a point estimator for the a posteriori estimation of variance, which could be further improved by using a Bayesian neural network [18].

In [21] it is described how the performance of the EWC can be further improved by a reparameterization that leads to a rotation of the parameter space.

In this context, it could be examined to what extent such networks are suitable for approaches like on-the-fly or collaborative calibration [15,24]. Therefore it might be possible to train sensors cumulatively using the presented method with even smaller amounts of data and thus keep the individual length of stay at the reference device small.

Acknowledgements. This work has been partially funded by the German Federal Ministry for Traffic and Digital Infrastructure (BMVI) as part of project SmartAQnet [4] (grant number 19F2003B).

References

1. Barcelo-Ordinas, J.M., Garcia-Vidal, J., Doudou, M., Rodrigo-Muñoz, S., Cerezo-Llavero, A.: Calibrating low-cost air quality sensors using multiple arrays of sensors. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6. IEEE (2018)
2. Benna, M.K., Fusi, S.: Computational principles of biological memory. arXiv preprint arXiv:1507.07580 (2015)
3. Brochu, E., Cora, V.M., De Freitas, N.: A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599 (2010)
4. Budde, M., et al.: SmartAQnet: remote and in-situ sensing of urban air quality. In: Proceedings of SPIE Remote Sensing of Clouds and the Atmosphere XXII, vol. 10424, p. 104240C (2017)
5. Budde, M., et al.: Potential and limitations of the low-cost SDS011 particle sensor for monitoring urban air quality. ProScience **5**, 6–12 (2018)
6. Budde, M., Zhang, L., Beigl, M.: Distributed, low-cost particulate matter sensing: scenarios, challenges, approaches. ProScience **1**, 230–236 (2014)
7. Castell, N., et al.: Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? Environ. Int. **99**, 293–302 (2017)
8. Cavaliere, A., et al.: Development of low-cost air quality stations for next generation monitoring networks: calibration and validation of PM2.5 and PM10 sensors. Sensors **18**(9), 2843 (2018)

9. Cheng, Y., et al.: AirCloud: a cloud-based air-quality monitoring system for everyone. In: Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, pp. 251–265. ACM (2014)
10. Diaz-Bone, R.: Statistik für Soziologen. UTB GmbH (2018)
11. Dutta, P., et al.: Common sense: participatory urban sensing using a network of handheld air quality monitors. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys 2009, pp. 349–350. Association for Computing Machinery, New York (2009). https://doi.org/10.1145/1644038.1644095
12. Fusi, S., Drew, P.J., Abbott, L.F.: Cascade models of synaptically stored memories. Neuron **45**(4), 599–611 (2005)
13. Géron, A.: Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc., Newton (2017)
14. Grimm Aerosol Technik: Model EDM180. https://www.grimm-aerosol.com/products-en/environmental-dust-monitoring/approved-pm-monitor/edm180/
15. Hasenfratz, D., Saukh, O., Sturzenegger, S., Thiele, L.: Participatory air pollution monitoring using smartphones. Mob. Sens. **1**, 1–5 (2012)
16. Hasenfratz, D., Saukh, O., Thiele, L.: On-the-fly calibration of low-cost gas sensors. In: Picco, G.P., Heinzelman, W. (eds.) EWSN 2012. LNCS, vol. 7158, pp. 228–244. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28169-3_15
17. Hojaiji, H., Kalantarian, H., Bui, A.A., King, C.E., Sarrafzadeh, M.: Temperature and humidity calibration of a low-cost wireless dust sensor for real-time monitoring. In: 2017 IEEE Sensors Applications Symposium (SAS), pp. 1–6. IEEE (2017)
18. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. Proc. Natl. Acad. Sci. **114**(13), 3521–3526 (2017)
19. Kumaran, D., Hassabis, D., McClelland, J.L.: What learning systems do intelligent agents need? Complementary learning systems theory updated. Trends Cogn. Sci. **20**(7), 512–534 (2016)
20. Li, J.J., Faltings, B., Saukh, O., Hasenfratz, D., Beutel, J.: Sensing the air we breathe—The OpenSense Zurich dataset. In: Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
21. Liu, X., Masana, M., Herranz, L., Van de Weijer, J., Lopez, A.M., Bagdanov, A.D.: Rotate your networks: Better weight consolidation and less catastrophic forgetting. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 2262–2268. IEEE (2018)
22. Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., Zhang, G.: Transfer learning using computational intelligence: a survey. Knowl.-Based Syst. **80**, 14–23 (2015)
23. Maag, B., Zhou, Z., Thiele, L.: A survey on sensor calibration in air pollution monitoring deployments. IEEE Internet Things J. **5**(6), 4857–4870 (2018)
24. Markert, J.F., Budde, M., Schindler, G., Klug, M., Beigl, M.: Private rendezvous-based calibration of low-cost sensors for participatory environmental sensing. In: Proceedings of the Second International Conference on IoT in Urban Space, pp. 82–85. ACM (2016)
25. McClelland, J.L., McNaughton, B.L., O'Reilly, R.C.: Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. Psychol. Rev. **102**(3), 419 (1995)
26. Mertens, P., Rässler, S.: Prognoserechnung. Springer, Heidelberg (2005). https://doi.org/10.1007/b138143

27. Monn, C.: Exposure assessment of air pollutants: a review on spatial heterogeneity and indoor/outdoor/personal exposure to suspended particulate matter, nitrogen dioxide and ozone. Atmos. Environ. **35**(1), 1–32 (2001)
28. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge (2012)
29. North, R., Richards, M., Cohen, J., Hoose, N., Hassard, J., Polak, J.: A mobile environmental sensing system to manage transportation and urban air quality. In: 2008 IEEE International Symposium on Circuits and Systems (2008)
30. World Health Organization: Who releases country estimates on air pollution exposure and health impact (2016). https://goo.gl/G4uqFE
31. Parisotto, E., Ba, J.L., Salakhutdinov, R.: Actor-mimic: deep multitask and transfer reinforcement learning. arXiv preprint arXiv:1511.06342 (2015)
32. Pascanu, R., Bengio, Y.: Revisiting natural gradient for deep networks. arXiv preprint arXiv:1301.3584 (2013)
33. Polak, J.: Mobile environmental sensor systems across a grid environment-the message project. ERCIM News **2007**(68) (2007)
34. Prahm, C., Paassen, B., Schulz, A., Hammer, B., Aszmann, O.: Transfer learning for rapid re-calibration of a myoelectric prosthesis after electrode shift. In: Ibáñez, J., González-Vargas, J., Azorín, J.M., Akay, M., Pons, J.L. (eds.) Converging Clinical and Engineering Research on Neurorehabilitation II. BB, vol. 15, pp. 153–157. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-46669-9_28
35. Rai, A.C., et al.: End-user perspective of low-cost sensors for outdoor air pollution monitoring. Sci. Total Environ. **607**, 691–705 (2017)
36. Spinelle, L., Gerboles, M., Villani, M.G., Aleixandre, M., Bonavitacola, F.: Field calibration of a cluster of low-cost available sensors for air quality monitoring. Part A: ozone and nitrogen dioxide. Sens. Actuators, B Chem. **215**, 249–257 (2015)
37. Yamamoto, K., Togami, T., Yamaguchi, N., Ninomiya, S.: Machine learning-based calibration of low-cost air temperature sensors using environmental data. Sensors **17**(6), 1290 (2017)