Talanta 117 (2013) 112–118

Contents lists available at ScienceDirect

Talanta

journal homepage: www.elsevier.com/locate/talanta

Extended artificial neural networks: Incorporation of *a priori* chemical knowledge enables use of ion selective electrodes for *in-situ* measurement of ions at environmentally relevant levels

Amy V. Mueller*, Harold F. Hemond

Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, 77 Massachusetts Ave. 48-102, Cambridge, MA 02139, United States

ARTICLE INFO

Article history:
Received 1 July 2013
Received in revised form
26 August 2013
Accepted 27 August 2013
Available online 4 September 2013

Keywords:
Chemometrics
Artificial Neural Network (ANN)
Ion Selective Electrode (ISE) array
Electronic tongue
Charge balance
Real-time chemical measurements
In-situ environmental sensors

ABSTRACT

A novel artificial neural network (ANN) architecture is proposed which explicitly incorporates *a priori* system knowledge, i.e., relationships between output signals, while preserving the unconstrained non-linear function estimator characteristics of the traditional ANN. A method is provided for architecture layout, disabling training on a subset of neurons, and encoding system knowledge into the neuron structure. The novel architecture is applied to raw readings from a chemical sensor multi-probe (electric tongue), comprised of off-the-shelf ion selective electrodes (ISEs), to estimate individual ion concentrations in solutions *at environmentally relevant concentrations* and *containing environmentally representative ion mixtures*. Conductivity measurements and the concept of charge balance are incorporated into the ANN structure, resulting in (1) removal of estimation bias typically seen with use of ISEs in mixtures of unknown composition and (2) improvement of signal estimation by an order of magnitude or more for both major and minor constituents relative to use of ISEs as stand-alone sensors and error reduction by 30–50% relative to use of standard ANN models. This method is suggested as an alternative to parameterization of traditional models (e.g., Nikolsky–Eisenman), for which parameters are strongly dependent on both analyte concentration and temperature, and to standard ANN models which have no mechanism for incorporation of system knowledge. Network architecture and weighting are presented for the base case where the dot product can be used to relate ion concentrations to both conductivity and charge balance as well as for an extension to log-normalized data where the model can no longer be represented in this manner. While parameterization in this case study is analyte-dependent, the architecture is generalizable, allowing application of this method to other environmental problems for which mathematical constraints can be explicitly stated.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Artificial neural networks (ANNs), introduced by Marvin Minsky in 1954 [1] as a model for knowledge storage and learning, have grown into a versatile tool for non-linear function estimation in a variety of fields, including image and natural language processing, robotics and controls, and chemometrics. Within the last of these, ANNs are used both as a predictive (or pattern recognition) tool for analysis of chemical or environmental data (e.g., [2]) and as a post-processing module for sensor data [3–11]. This work addresses customization of the standard ANN architecture for sensor post-processing to take advantage of chemical understanding of these systems which may be available *a priori* and which may otherwise be neglected.

Specifically, this work targets quantification of the charge balance of natural water samples (i.e., $>99\%$ of the ion content) through use of relatively inexpensive, off-the-shelf sensors such as ion selective electrodes (ISEs). While ISEs may be only partially or poorly selective for their target analyte, they have great promise for use in *in-situ* portable sensor systems if:

1. Cross-ion interferences can be addressed;
2. Electrode responses can be adequately characterized without the need for specification of the large number of coefficients used in traditional models, e.g., Nernst, Nikolsky–Eisenman, which are dependent on temperature [12] as well as the concentrations of both the target and the interfering analytes (the response to which may not be Nernstian) [12–14];
3. ISE responses can be quantified even in the non-linear range where natural concentrations may lie (<10–$100\ \mu M$).

In this context, where the majority of ions are quantified, one can explicitly constrain results using conductivity measurements

* Corresponding author. Tel.: +1 857 445 8226.
E-mail address: amym@alum.mit.edu (A.V. Mueller).

0039-9140/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.talanta.2013.08.045

(via a weighted sum using molar conductances) and the concept of charge balance. The proposed ANN architecture implements these constraints for the estimation of the following ions (both major and minor constituents of oxygenated natural waters): Na^+, K^+, Ca^{2+}, Mg^{2+}, NH_4^+, Cl^-, NO_3^-, SO_4^{2-}, CO_3^{2-}/HCO_3^-, and H^+/OH^- (pH system). Note that ammonium (NH_4^+) and nitrate (NO_3^-) are included in particular as minor but extremely important nutrient constituents whose concentrations may provide information about a wide range of both natural and anthropogenic processes. Simultaneous measurement of relevant environmental parameters (i.e., temperature and conductivity) are further included in the ANN input data set to provide additional information about the chemical state of the system.

The combined hardware/software system was validated on a set of simulated environmental samples, based on New England water composition and comprised of mixes of the target analytes *at environmental levels* and *in environmentally representative mixtures*. This paper presents the software architecture development, applications, and results. For further discussion of the hardware components and chemical implications the reader is referred to [15].

2. Materials and methods

2.1. Artificial neural network (ANN) setup

Artificial neural networks are conceptually modeled on a simple understanding of human neural structure in which initial neuron interconnections are specified by the system designer but weights of interconnections are adapted during training ("learn"). The number of hidden layers, number of nodes per layer (need not be constant between layers), transfer function between layers (often different for input, hidden, and output layers), learning rate, momentum, and training algorithm must also be specified and will vary between applications. Significant variation among neural networks produced with different tuning parameter values has been shown, and it is important to note that ANN problems have no closed form solution.

Sensitivity to choice of parameter values has been analyzed relative to chemical applications [16–18,7], and it is possible to identify parameter values favored in the relevant literature as a starting point for analysis of any new problem [19,7]. It is, however, typically necessary to explore the space of possible ANNs through trial-and-error to find an optimal system on an application-by-application basis as exhaustive tests have shown resulting weights and biases can be determined more strongly by training parameters than by training data [20]. This search may be automated [21,22], e.g., using quality of estimation for the test set data as the goodness heuristic, though the quality of an ANN does not necessarily vary monotonically with each parameter. Significantly, in cases where genetic algorithms have been used to search the space of possible ANNs for chemical applications [17,23], the overall search time was shortened but the resulting neural networks performed at approximately the same level as those created by trial-and-error. As such, a human-directed optimization search is followed in this work.

A prototypical neural network neuron is shown in Fig. 1, with inputs p_i, weights w_i, bias b, transfer function f, and output a. The output a as a function of the inputs p_i is shown for the case where inputs are combined with the *dot product* function, the standard method proposed for ANNs. The method described here takes advantage of the dot product relationship of chemical concentrations to both total conductivity and charge balance to implement these functions within the "standard" neural network architecture.

Network inputs and outputs are typically normalized before training, to equalize the effect of each channel on the output RMSE calculation, e.g., to values between ± 1. For chemical applications, concentration data is often log-transformed prior to this normalization. While ultimately this was not shown to improve results of this work, the derived technique is presented for implementation of chemical constraints for both mathematical forms.

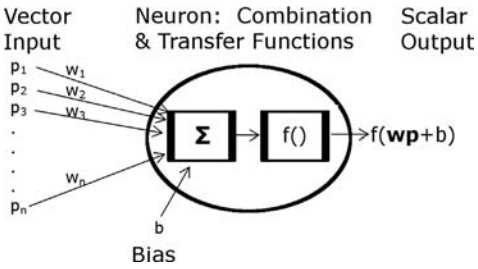

Fig. 1. Prototypical neuron component of a neural network. The input vector **p** is combined with weight vector **w** (usually with the dot product, as shown), biased, and processed using a transfer function (i.e., linear, sigmoid, etc.) to produce the scalar neuron output.

Table 1
Range of parameters explored for design of neural networks.

Parameter	Values
Hidden layer size	6, 9, 12, 15, 18
μ	0.001, 0.1
μ_{inc}	1.5, 10, 50
μ_{dec}	0.1, 0.5, 0.9
Training goal	10^{-6}
Number of hidden layers	1, 2, 3
Max number of epochs	10,000
Hidden layer transfer function	tansig
Output layer transfer function	purelin
Training fraction	0.7
Validation fraction	0.15
Testing fraction	0.15

A range of neural network architectures was constructed and trained using the data sets described above (11-ISE array, temperature, and conductivity sensor data vs. 12 known ionic concentrations (Na^+, K^+, Ca^{2+}, Mg^{2+}, NH_4^+, Cl^-, NO_3^-, SO_4^{2-}, carbonate and pH systems)). Design of the architecture and test set took place at two levels, which will be referred to here as *external* (layout of constraint layers, target data normalization) and *internal* (setting of individual parameters to control the ANN training, number of hidden layers or nodes per hidden layer, etc.). The full range of internal ANN options was explored for each combination of external options during optimization; for clarity, these two design stages are discussed independently. Training of networks followed standard back propagation methods splitting the full data set into training (70%), validation (15%), and testing (15%) components. Implementation of all neural network functionality was done using the Matlab Neural Network Toolbox V7.0 (R2010b), and key scripts can be found in [15].

2.2. Internal ANN parameter space

The parameter space explored for the internal ANN settings was informed by initial simulation experiments [15]. Table 1 shows the values for each parameter. In cases where previous experience had already identified the superior parameter choice, only the single selected value is listed.

This set of parameterizations resulted in creation and training of 2790 independent ANNs for each of the external architectures specified. Resulting ANNs were ranked on several metrics, including MSE (mean squared error), NRMSE (normalized root mean squared error), and MRE (mean relative error). Ultimately ranking based on the NRMSE of the target ion concentrations (excluding

pH and carbonate systems) was identified to correlate most closely with the desired system behavior, i.e., a balance of accuracy on all channels, including low-concentration nitrogen ion species. Recall that, despite the relatively smaller contribution of these ions to the total charge balance of natural waters, both nitrate and ammonium are key nutrients and tracers of a number of both natural and anthropogenic processes such as eutrophication (nutrient pollution), nitrification/denitrification (emission of greenhouse gasses), or accidental or intentional discharge of industrial or wastewater treatment facility effluent.

2.3. External ANN parameter space

Design of the ANN architecture (number of inputs and outputs, constraints, data normalization) was done independently of selection of internal parameters. Options explored, discussed further in the subsequent section, included the following:

- Outputs = 12 ion concentrations only.
- Outputs = 12 ion concentrations + charge balance (CB) constraint.
- Outputs = 12 ion concentrations + electrical conductivity (EC) constraint.
- Outputs = 12 ion concentrations + both CB and EC constraints.
- Target data form: absolute concentrations vs. log-transformed.

Use of the *logsin* transfer function between the last hidden layer and the output layer (also requiring use of the range [0,1] for the target *mapminmax* function) was explored as an alternative to log-transformation of the data to ensure that all outputs are non-negative (and thus physically relevant in the case of concentration values), however results were consistently inferior to those produced using the log-transformation.

2.4. Adding constraints in the ANN architecture

To incorporate chemical constraints into the neural network architecture, a non-traditional format must be explored. The possibility of using feedback, i.e., analogous to use of an op-amp feedback circuit, was rejected due to the lack of software to support zero-delay feedback loops. Instead, an architecture is proposed in which *certain weights are prevented from being altered during training*. This allows an explicit calculation of weighted sum functions of analyte concentrations, which can then be trained *as an additional output*.

To explore this further, let us consider a neural network with n concentration outputs $\{C_1, .., C_n\}$. Let us assume also that there is measurable quantity A whose relationship to individual concentrations is governed by a weighted sum (dot product) relationship as follows:

$$A = \sum_{i=1}^{n} a_i \cdot C_i \tag{1}$$

In this case a single neuron, using the dot product function to combine its inputs and with weights appropriately set (to the a_i coefficient values), can be used to represent this relationship *as long as the weights are not altered during training*. It is significant to note that both conductivity and charge neutrality take this form, implying that both chemical constraints can, in concept, be easily integrated into a relatively simple neural network structure. Both quantities are mathematically well-defined (and/or easily measured), allowing for the creation of accurate target outputs for training purposes. Because the calculations are mutually independent (the value for electrical conductivity is not required to complete the charge balance calculation and vice versa), a system can use either constraint exclusively or may incorporate both in parallel. The specific formulations, as well as discussion of the practical details associated with input/output scaling and using $\log_{10}(C_i)$ values as targets, are explored in more detail below.

Implementation of these constraints is visualized here for an example network with the following characteristics:

- 13 input signals (here: sensor outputs).
- Two hidden layers with 6 and 12 neurons with, respectively, tansig and purelin transfer functions.
- 12 target outputs (here: ion concentrations).

Fig. 2 shows the Matlab-formatted representation for this network (hidden *Layers* contain **w** weight and **b** bias vectors and tansig or linear transfer function; numbers under boxes indicate the number of nodes in the layer), along with an identical network with one or two additional constraint layers appended as specified above. One will notice that (1) the basic network structure is unchanged by appending constraint layers and (2) the weights and bias values for the constraint layers are all marked as *untrained*. Yellow nodes mark points at which inputs are normalized/outputs are un-normalized, i.e., data is scaled per channel such that the network operates on a set of vectors containing values in the range $[-1,1]$.

2.5. Mathematics of chemical constraint implementation

Because constraint vectors take their input from a point in the network *where signals are still normalized*, assignment of weight and bias values to these untrained layers must take into account both the natural chemical relationship being modeled and the normalization of the data incorporated into the neural network training algorithm. This section derives appropriate values for the electrical conductivity relationship; implementation of a charge balance constraint follows an identical process.

Recall that electrical conductivity (EC, temperature corrected to match standard values at 25 °C) can be related to the concentrations C_i of ions in solution via the following relationship [24], where the γ_i coefficients represent the molar conductance in (μS/cm)/(mmol/L) of species i (see Table 2).

$$\mathrm{EC} = \sum_{i=1}^{n} \gamma_i \cdot C_i. \tag{2}$$

For a network without input/output normalization, the weight values would be identical to the γ_i coefficients and the corresponding bias signals would be zero. Because of the magnitude range of output signals across channels, however, such a network would be likely to favor accurate estimates of large-magnitude signals (and not necessarily accurately estimate small-magnitude signals). As such, inputs and outputs are normalized such that the values on each channel occupy the same range (e.g., using Matlab's built-in *mapminmax* function) before training as follows. For input vector $\mathbf{C_i}$ (e.g., $[Ca^{2+}]$ for the entire suite of training samples):

$$\tilde{\mathbf{C}}_\mathbf{i} = \frac{(y_{\max} - y_{\min})}{(C_{i,\max} - C_{i,\min})} \cdot (\mathbf{C_i} - C_{i,\min}) + y_{\min} \tag{3}$$

where $y_{\min}$ and $y_{\max}$ are chosen by the user (typically ± 1, and which can be set independently for each signal), $C_{i,\min}$ is the minimum value in the $\mathbf{C_i}$ vector, and $C_{i,\max}$ is the corresponding maximum value. By defining C_i and y ranges, this can be rewritten as

$$\tilde{\mathbf{C}}_\mathbf{i} = \frac{y_{range}}{C_{i,range}} \cdot (\mathbf{C_i} - C_{i,\min}) + y_{\min} \tag{4}$$

which can be inverted to yield the original concentration value as

$$\begin{aligned} \mathbf{C_i} &= (\tilde{\mathbf{C}}_\mathbf{i} - y_{\min}) \cdot \frac{C_{i,range}}{y_{range}} + C_{i,\min} \\ &= \alpha_i \cdot \tilde{\mathbf{C}}_\mathbf{i} + \beta_i \end{aligned} \tag{5}$$

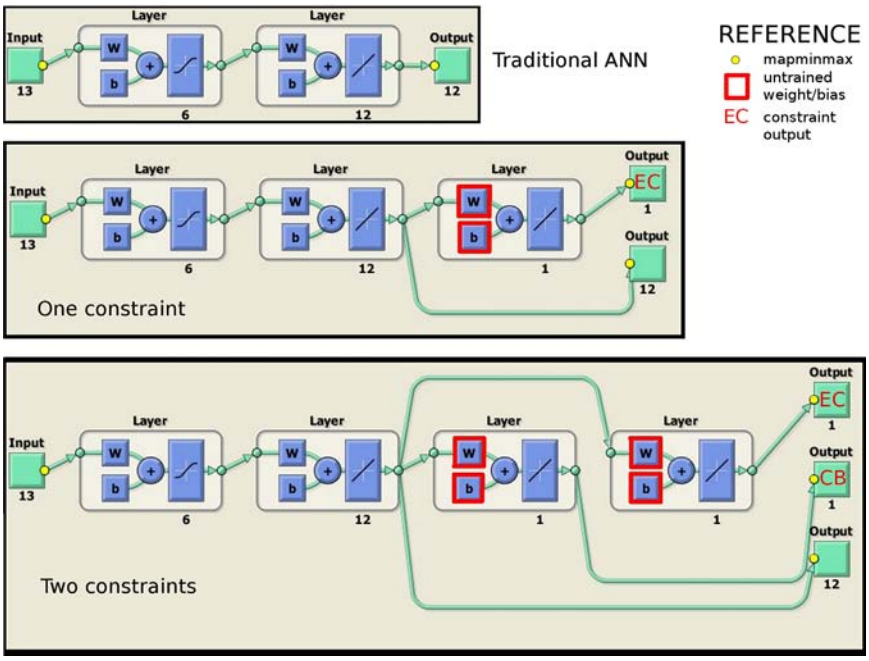

Fig. 2. Matlab-formatted representation of three neural network architectures: a traditional structure (top), one with a single constraint layer (middle), and one with two constraint layers (bottom). (Input and output layers shown as green boxes, hidden layers as a box surrounding **w** weight vector, **b** bias vector, and transfer function.) Note that the middle case constraint output is labeled EC for the Electrical Conductivity case but could represent either of the chemical constraints discussed here. Weights and biases omitted from training in the non-traditional architectures are boxed in red, while nodes where normalization (range scaling of a single variable to $[-1,1]$ using Matlab's *mapminmax* function) or inverse normalization is implemented are highlighted in yellow.

Table 2
Charge balance and conductivity constraint multipliers used for calculation of non-trained neuron weights. Conductance values adapted from [25,26].

Analyte	γ_{CB} []	γ_{EC} [(μS/cm)/(mmol/L)]
Na^+	+1	50.10
K^+	+1	73.5
Ca^{2+}	+2	119
NH_4^-	+1	73.5
Mg^{2+}	+2	106
SO_4^{2-}	−2	160
Cl^-	−1	76.46
NO_3^-	−1	71.46
H^+	0[a]	349.6
OH^-	0[a]	199.1
HCO_3^-	−1	44.5
CO_3^{2-}	−2	138.6
HSO_4^-	−1	52.0

[a] Note that the charge balance constraint has been formulated such that the balance of all other ions is trained to the net contribution from H^+ and OH^- as explained in the text.

where α_i and β_i are defined by the known constants $y_{\min}$, y_{range}, and $C_{i,range}$. Recalling that the EC calculation is summed over the n ionic constituents, the calculation for electrical conductivity (for all m samples) thus becomes

$$\mathbf{EC} = \sum_{i=1}^{n} \gamma_i \cdot \alpha_i \cdot \tilde{\mathbf{C}}_\mathbf{i} + \sum_{i=1}^{n} \gamma_i \cdot \beta_i \tag{6}$$

It is, however, necessary to account for the normalization on the EC channel itself as the calculation performed by the neuron must output $\tilde{\mathbf{EC}}$, the normalized EC signal (the actual EC signal is subsequently restored by the inverse mapping).

$$\tilde{\mathbf{EC}} = \frac{(y_{\max} - y_{\min})}{(EC_{\max} - EC_{\min})} \cdot (\mathbf{EC} - EC_{\min}) + y_{\min}$$
$$= \frac{y_{range}}{EC_{range}} \cdot (\mathbf{EC} - EC_{\min}) + y_{\min}$$
$$= \alpha_{EC} \cdot \mathbf{EC} + \beta_{EC} \tag{7}$$

Combining this with Eq. (6) and distributing coefficients to separate terms yields

$$\tilde{\mathbf{EC}} = \sum_{i=1}^{n} \alpha_{EC} \cdot \gamma_i \cdot \alpha_i \cdot \tilde{\mathbf{C}}_\mathbf{i} + (\beta_{EC} + \sum_{i=1}^{n} \alpha_{EC} \cdot \gamma_i \cdot \beta_i)$$
$$= \sum_{i=1}^{n} w_i \cdot \tilde{\mathbf{C}}_\mathbf{i} + B \tag{8}$$

From Eq. (8) it can be seen that the individual weights for connections from each signal $\tilde{\mathbf{C}}_\mathbf{i}$ to the EC layer should be set to the product $\alpha_{EC} \cdot \gamma_i \cdot \alpha_i$ while the bias connection to the EC layer should be set to $(\beta_{EC} + \sum_{i=1}^{n} \alpha_{EC} \cdot \gamma_i \cdot \beta_i)$.

The corresponding calculation for a charge balance layer progresses identically, however it is important to determine exactly what relationship is being encoded with this constraint. It is possible to calculate a full charge balance, i.e.,

$$\sum_i |z_i| \cdot [C_i^{z_i+}] - \sum_i |z_i| \cdot [C_i^{z_i-}] \approx 0 \tag{9}$$

However, this may not be the most appropriate formulation for the constraint due to the manner in which neural nets are implemented, i.e., use of the equation in this form requires that the target be a zero vector, and consequently the normalization stage would result in undefined values. It is instead better to move a few terms (e.g., those corresponding to the carbonate system or pH) to the right-hand side of the equation to ensure non-zero values. For example:

$$([Na^+] + [K^+] + [NH_4^+] + [2 \cdot Ca^{2+}] + [2 \cdot Mg^{2+}])$$
$$-([Cl^-] + [NO_3^-] + [2 \cdot SO_4^{2-}] + [HCO_3^-] + [2 \cdot CO_3^{2-}])$$
$$= [OH^-] - [H^+] \tag{10}$$

This particular formulation has been chosen for this work as the pH of the system is relatively well known (i.e., the pH electrode is nearly interference free in waters of relatively weak ionic strength) and should thus be a reasonable signal with which to constrain the rest of the system. Because each channel is normalized (e.g., to the range $[-1,1]$), the small magnitude of the right-hand side does not adversely affect system training. The coefficient values required to calculate the weight and bias values for the charge balance constraint layer are correspondingly given in Table 2.

2.6. Weight constraints with logarithmic targets

Normalization of input and output data has been described above as a method for 'fairly' weighting contributions of all signals to error calculations (and thus to ANN training algorithms), however it should be noted that the linear transformation performed by *mapminmax* may not result in optimal weighting in cases where signals vary over several orders of magnitude *on a single input or output channel*. In this case it may be advisable to perform a log-transformation on the data before passing it to the *mapminmax* function, however transformation of the data in this manner is not directly compatible with the constraint methodology mapped out above. Because the dot product will no longer correctly represent the chemical relationship in this form, it is necessary to both recalculate appropriate weight values *and to create a new method for calculating weighted inputs* in this case. For an input vector $\mathbf{C_i}$ with all elements >0, two stages of normalization are required:

$$\mathbf{L_i} = \log_{10}\mathbf{C_i} \tag{11}$$

$$\tilde{\mathbf{L_i}} = \frac{(y_{\max}-y_{\min})}{(L_{i,\max}-L_{i,\min})}\cdot(\mathbf{L_i}-L_{i,\min})+y_{\min} \tag{12}$$

where the signals available internal to the neural network (previously $\tilde{\mathbf{C}}_\mathbf{i}$) correspond in this case to $\tilde{\mathbf{L}}_\mathbf{i}$. The EC calculation, however, must still be made using $\mathbf{C_i}$, so it is necessary to manipulate these equations such that $\mathbf{C_i}$ is given as a function of other system parameters. *Note*: *the mathematics derived here apply to a network for which concentration values are both log-transformed and magnitude normalized while the constraint* (*EC or CB*) *values are only magnitude normalized*.

$$\begin{aligned}\tilde{\mathbf{L_i}} &= \frac{y_{range}}{L_{i,range}}\cdot(\mathbf{L_i}-L_{i,\min})+y_{\min}\\ &= \alpha_i\cdot\mathbf{L_i}+\beta_i\\ &= \alpha_i\cdot\log_{10}\mathbf{C_i}+\beta_i\end{aligned} \tag{13}$$

Inverting this equation gives $\mathbf{C_i}$ as a function of other system parameters:

$$\begin{aligned}\mathbf{C_i} &= 10^{\tilde{\mathbf{L_i}}-\beta_i/\alpha_i}\\ &= 10^{\tilde{\mathbf{L_i}}/\alpha_i}\cdot 10^{-\beta_i/\alpha_i}\\ &= \delta_i\cdot 10^{\tilde{\mathbf{L_i}}/\alpha_i}\end{aligned} \tag{14}$$

where the notation δ_i has been introduced to represent a function of coefficients α_i and β_i. Substituting this into the known EC relationship (Eq. (2)) and correcting for the magnitude normalization on the EC data (Eq. (7)) yields

$$\mathbf{EC} = \sum_{i=1}^{n}\gamma_i\cdot\delta_i\cdot 10^{\tilde{\mathbf{L_i}}/\alpha_i} \tag{15}$$

$$\begin{aligned}\tilde{\mathbf{EC}} &= \sum_{i=1}^{n}\alpha_{EC}\cdot\gamma_i\cdot\delta_i\cdot 10^{\tilde{\mathbf{L_i}}/\alpha_i}+\beta_{EC}\\ &= \sum_{i=1}^{n} w_{1,i}\cdot 10^{w_{2,i}\cdot\tilde{\mathbf{L_i}}}+B\end{aligned} \tag{16}$$

As is clear from this formulation, *two* weights are now required for each input signal to calculate EC as a function of the normalized $\tilde{\mathbf{L}}_\mathbf{i}$, and the calculation is no longer in dot product form. This is a non-standard format for a weighting function and will not be available in most ANN software libraries, however one can easily be created and interfaced with standard ANN software. This custom function must include the following:

- Size of weight vector w required for input vector (to this layer) p of size $r\times s$.
- Output z as a function of inputs p and weights w
- dz/dp as a function of w, p
- dz/dw as a function of w, p

A function of this sort was created in Matlab and is provided in [15] as *cbsum.m*. (No weighting function of the desired form is available from the Matlab Neural Network Toolbox, however the custom weighting function is simply assigned as the *layer.WeightFcn* within the standard neural network framework.) Note that while the last two items (partial derivatives) are required components of the weighting function, they are used exclusively for training and will thus not affect neural network training in cases where training is disabled on the corresponding weights.

3. Results

Ion concentrations were estimated using (1) ISE calibration curves, assuming ISEs were used as stand-alone sensors (i.e., response to interferences were not modeled) and (2) approximately 90,000 different ANNs based on the parameter sets outlined above. The ISE-only results are presented as a baseline case against which to measure improvements in estimation accuracy (representing the concentration values which would be estimated through use of a single ISE for measurement of a single analyte in the unprocessed ion mixture). ANN results are evaluated and compared to the baseline to ascertain level of improvement. Abbreviated results are provided here, while a complete characterization and evaluation of the hardware/software system is provided in [15].

As stated above, the minimum sum of NRMSE for the target ions (excluding pH and carbonate systems) was used to select the optimal ANN, the characteristics of which are provided in Table 3.

It is significant to note that networks trained to absolute concentration data (i.e., without log-normalization) produced lower total NRMSEs by a factor of 1.5–3 compared to other configurations, which was unexpected due to the wide range of concentrations (several orders of magnitude) and because log-normalization of data is a common practice within the field of chemometrics (see Fig. 3). In this case, inclusion of both feedback channels improves results up to 30–50%, however it is interesting to note that corresponding gains were not seen when constraint channels were added independently.

Table 3
Parameterization for best ANN (chosen using NRMSE metric).

Architecture	
Outputs	12 ions+CB+EC
Normalization	None
Parameterization	
Hidden layers	[6, 18, 15]
μ	0.001
μ_{dec}	0.1
μ_{inc}	10

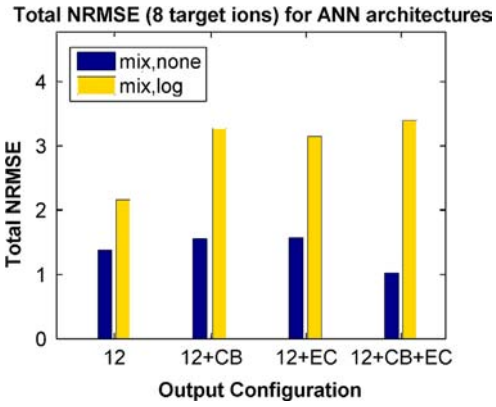

Fig. 3. Total NRMSE as a function of ANN architecture. Horizontal axis shows output configuration while vertical bars represent options for data used in training (mix data only/mix data plus single-salt standards) and data normalization (none/log).

The optimal ANN configuration was then used to estimate the individual concentrations of the 8 target ions as well as the pH and carbonate species. Example scatter plots of these results – with ISE estimations overlain for reference – are shown in Fig. 4, and results for the two constraint channels are provided in Fig. 5. ANN concentration estimates (blue) generally fall along the 1:1 lines (red), having *successfully removed the bias encountered* with the ISEs as stand-alone sensors. For ANN results the 95% confidence interval on the slope contains 1 (perfect agreement) for all target analytes except SO_4^{2-} and CO_3^{2-} (for which the system did not contain a specific ISE), for which it is slightly less than 1. Intercepts are only statistically significantly different from zero for Cl^- and SO_4^{2-}. These facts essentially identify ANN estimates as an unbiased estimator (i.e., *accurate*), and therefore both the R^2 and RMSE values provide information about the magnitude of scatter around the targets (*precision*). RMSE values are $< 10\ \mu M$ for all analytes excepting Na^+ (the ion present at highest concentrations, up to mM range), even for those analytes for which no specific sensor is used (Mg^{2+}, SO_4^{2-}, OH^-, and the carbonate species).

4. Discussion

Artificial neural networks provide a powerful tool for the creation of non-traditional network structures, including those which can incorporate *a priori* knowledge of system characteristics on top of standard ANN non-linear function estimation capabilities. When combined with custom weighting functions and the ability to omit certain weights from alteration during training, network architectures implementing known chemical constraints can be designed using existing software tools (e.g., standard Matlab functions). Assignment of weights and bias values based on known constraint equations requires attention to data normalization as well as inherent chemical relationships, however this work has laid out the fundamental requirements for implementing both conductivity and charge balance constraints in a neural network outputting absolute or log concentrations of the majority of ions found in fresh waters. Implementation of these constraints substantially improves predictive capability relative to use of ISEs as stand-alone sensors – entirely removing signal bias and reducing error by an order of magnitude or more – while the network architecture implementing both constraints outperformed even traditional neural networks, reducing NRMSE by 30–50%.

Extension of such constraint-incorporation techniques has the potential to add ANNs to the growing toolbox of methods available for analysis of environmental datasets. ANNs have an inherent

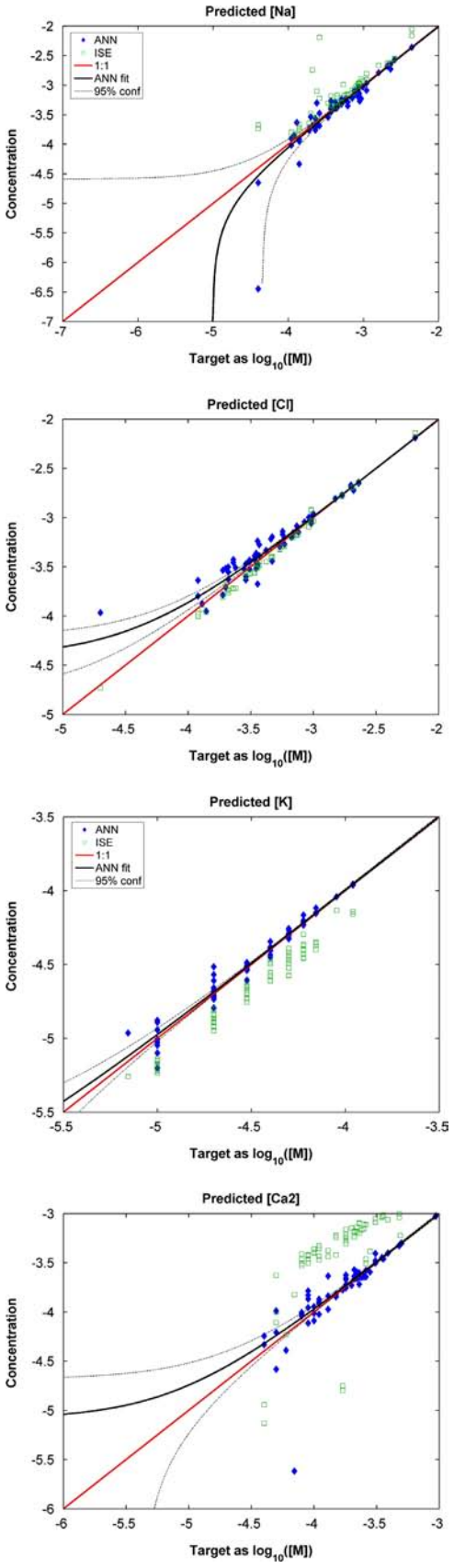

Fig. 4. Scatter plots of Na^+, Cl^-, K^+ and Ca^{2+} ion concentrations estimated using the optimal ANN as a function of target concentration. One-to-one line shown in red; regression of estimates against targets (concentration data) and 95% confidence interval on the linear fit shown in black. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

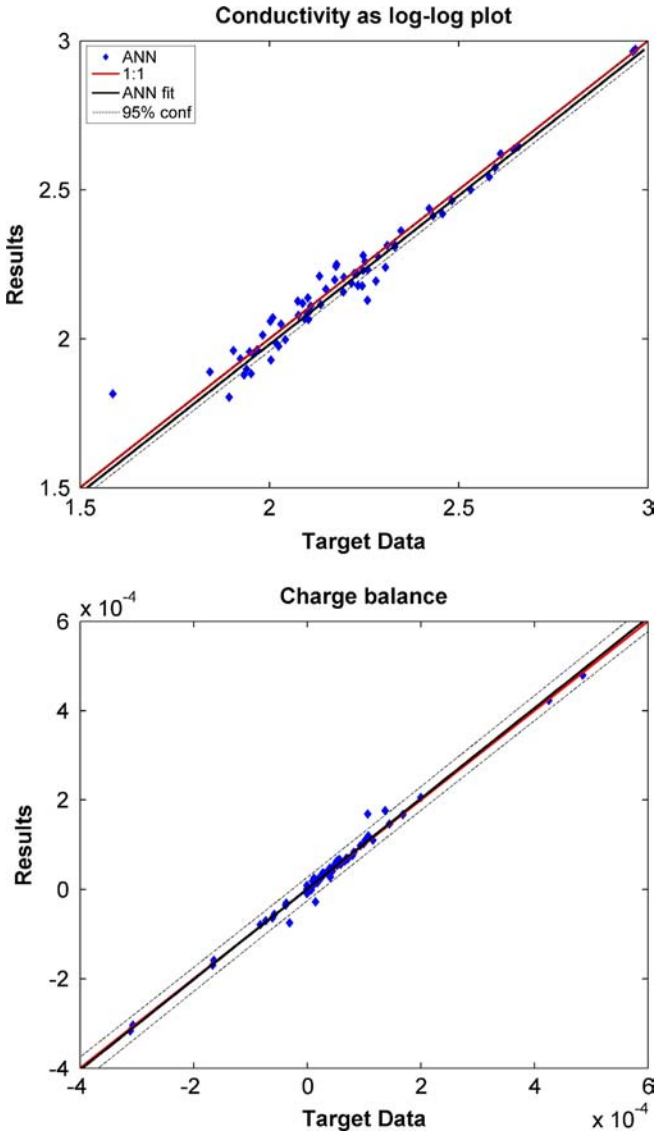

Fig. 5. Scatter plot of constraint estimations of optimal ANN as a function of target value. One-to-one line shown in red; regression of estimates against targets (data before log-transformation) and 95% confidence interval on the linear fit shown in black.

advantage in cases where (1) the relationship between the environmental factor and measurement is poorly understood or poorly constrained, (2) the relationship is expected to be non-linear, and (3) collection of sample data is less costly/time-consuming than the experiments required to constrain relationship parameters. In cases where multiple factors are of interest, e.g., chemical speciation within a closed system, and additional knowledge is available at the system level, e.g., mass balance for specific atoms, water flows, etc., this novel ANN formulation may provide a straightforward way to incorporate *a priori* knowledge with available sample data to further improve estimation capabilities.

Acknowledgments

Funding for this work has come from: Sea Grant NA10OAR4170086, a GSA Graduate Student Research Grant, an MIT-Xerox Fellowship, a Martin Foundation Sustainability Fellowship, an NSF Graduate Research Fellowship, NSF EAR 0330272, the National Research Foundation – Singapore (Agreement Dated 09/18/07), and the MIT William E Leonhard Chair.

References

[1] M. Minsky, Neural Nets and the Brain Model Problem, Ph.D. Thesis, Princeton University, Princeton, New Jersey, 1954.
[2] H. Maier, G. Dandy, Water Resour. Res. 32 (2010) 1013–1022.
[3] C. Di Natale, F. Davide, J. Brunink, A. D'Amico, Y. Vlasov, A. Legin, A. Rudnitskaya, Sensors Actuators B 34 (1996) 539–542.
[4] J. Mortensen, A. Legin, A. Ipatov, A. Rudnitskaya, Y. Vlasov, K. Hjuler, Anal. Chim. Acta 403 (2000) 273–277.
[5] A. Rudnitskaya, A. Ehlert, A. Legin, Y. Vlasov, S. Buttgenbach, Talanta 55 (2001) 425–431.
[6] C. Di Natale, A. Macagnano, F. Davide, A. D'Amico, A. Legin, Y. Vlasov, A. Rudnitskaya, B. Selezenev, Sensors Actuators B 44 (1997) 423–428.
[7] J. Gallardo, S. Alegret, R. Muoz, M. De-Romn, L. Leija, P. Hernndez, M. del Valle, Anal. Bioanalytical Chem. 377 (2003) 248–256.
[8] M. Baret, D. Massart, P. Fabry, F. Conesa, C. Eichner, C. Menardo, Talanta 51 (2000) 863–877.
[9] M. Cortina-Puig, X. Munoz-Berbel, M. Alonso-Lomillo, F. Munoz-Pascual, M. del Valle, Talanta 72 (2007) 774–779.
[10] A. Legin, A. Rudnitskaya, Y. Vlasov, C. Di Natale, E. Mazzone, A. D'Amico, Electroanalysis 11 (1999) 814–820.
[11] A. Legin, A. Rudnitskaya, L. Lvova, Y. Vlasov, C. Di Natale, A. D'Amico, Anal. Chim. Acta 484 (2003) 33–44.
[12] E. Zahran, V. Gavalas, M. Valiente, L. Bachas, Anal. Chem. 82 (2010) 3622–3628.
[13] W. Morf, The Principles of Ion-Selective Electrodes and of Membrane Transport, Elsevier Scientific Publishing Company and Akademiai Kiado, The Publishing House of the Hungarian Academy of Sciences, The Netherlands and Budapest, Hungary, 1981.
[14] E. Bakker, E. Pretsch, P. Buhlmann, Anal. Chem. 72 (2000) 1127–1133.
[15] A. Mueller, Development of a Combined Multi-Sensor/Signal Processing Architecture for Improved In-Situ Quantification of the Charge Balance of Natural Waters, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2012.
[16] M. Cortina, A. Gutes, S. Alegret, M. del Valle, Talanta 66 (2005) 1197–1206.
[17] E. Richards, C. Bessant, S. Saini, Chemometrics Intelligent Lab. Syst. 61 (2002) 35–49.
[18] J. Gallardo, S. Alegret, M. de Roman, R. Munoz, P. Hernandez, L. Leija, M. del Valle, Anal. Lett. 36 (2003) 2893–2908.
[19] F. Despagne, D. Massart, Analyst 123 (1998) 157–178.
[20] R. Srivastav, K. Sudheer, I. Chaubey, Water Resour. Res. 43 (2007) 1–12.
[21] L. Almeida, T. Ludermir, Neurocomputing 73 (2010) 1438–1450.
[22] A. Abraham, Neurocomputing 56 (2004) 1–38.
[23] E. Richards, C. Bessant, S. Saini, Sensors Actuators B 88 (2003) 149–154.
[24] American Public Health Association, Standard Methods for Examination of Water and Waste Water, 13th edition, American Public Health Association, Inc., New York, 1971.
[25] P. Atkins, J. de Paula, Physical Chemistry, 7th edition, Oxford University Press, Oxford, UK, 2003.
[26] P. Vanysek, CRC Handbook of Chemistry and Physics, 81st edtion, CRC Press, Boca Raton, FL.