A Study of Artificial Neural Networks for Electrochemical Data Analysis

Kan-Lin Hsueh (藤広琳)
Department of Energy and Resources, National United University, Maioli, Taiwan, R.O.C.

Artificial neural network has been demonstrated the capability of identifying the chemical composition from various analytical methods. This study tested the feasibility of using a single network to calculate data obtained from polarography, linear scanning voltammetry (LSV), and electrochemical impedance spectroscopy (EIS). Computer generated data were used for network learning and testing. The network property of a single layer network was calculated at various learning cycles, learning rates, and β values of the neuron activation function. This study also evaluated two-layer, three-layer, and four-layer networks. Test results conclude that a single artificial neural network could analyze various analytical data by using corresponding weight matrices. To produce accurate results, the network requires high training cycle, small β value of sigmoid function, and small learning rate. Network accuracy increase as the number of neuron increased to certain extents. Network structure plays minimum effects on the accuracy of network for a given number of neurons.

Keywords: Neural network; Polarization; Linear scanning voltammetry; Electrochemical impedance spectroscopy.

INTRODUCTION

Artificial neural network

Electrochemical analyses, such as polarography, linear scanning voltammetry (LSV), and electrochemical impedance spectroscopy (EIS), produce various plots or diagrams. Parameter values of these plots can be calculated by various numerical methods.1-3 Artificial neural network (ANN) analyzes the electrochemical signals. ANN is a numerical model that network structure is mimicking the neural connection of human brain. It had been use for data integration, data analysis, and database updating.4 It is capable of recognizing patterns and classifying data into various categories. It is also capable of updating its database through “training.” There are many types of artificial neural networks. Considering the signal flow direction, the networks can be classified into three categories, the feed-forward, the recurrent, and the hybrid network. In the feed-forward network, the signal flows from the inputs to the output direction. In the recurrent network, the output signals feed back as part of the input signals. The hybrid network contains different types of network. The outputs of one type of the network feed into other types of network.

ANN as an analytical tool

The ANN has been applied on the controlling or optimization of fabrication processes,5-9 such as reactive ion etching (RIE), chemical vapor deposition (CVD), and plasma etching, etc. The feasibility of using ANN for peak identification or classification had been studied in various spectroscopy areas,10-18 such as ultraviolet (UV) spectrum, chromatography, and cyclic voltammogram. Mittermayr et al.10 used 200 measured UV spectra to test the library search ability of artificial neural network. Network performed well on noisy spectral data. However, it failed to classify unknown spectra as unknown. Principle component artificial neural network was applied to determine the iron and cobalt in micellar medium from spectrophotometric data.11 Rowe et al.12 used 380 computer generated chromatography peak profiles to evaluate the efficiency of several artificial neural network architectures. They found that the accuracy of optimized artificial neural network (in 5.6 seconds) was similar to a human expert (in 8 hours) in the classification of 396 chromatography peak profiles. Wavelet neural network was used to evaluate the retention indices of gas chromatographic data.13,14 Overlapped voltammetric and differential pulse polarographic (DPP) peaks
could be resolved by using Elman recurrent artificial neural network,\textsuperscript{15} or back-propagation network,\textsuperscript{16} or principal component network.\textsuperscript{17} A wavelet artificial neural network was used to resolve highly overlapped anodic stripping voltammetry.\textsuperscript{18} ANN was also used in many electrochemical signals processing, such as glucose sensor,\textsuperscript{19} dual-genic impedimetric sensor for DNA hybridization detection,\textsuperscript{20} impedance data of binary mixture of microorganisms,\textsuperscript{21} LSV signal of adenine and cytosine interfered with hydrogen evolution,\textsuperscript{22} polarography curve of metal-ligand,\textsuperscript{23,24} piezoelectric quartz crystal sensor for organic pollutants.\textsuperscript{25} These studies were aiming to develop an artificial intelligence that can identify the chemical composition from the measured data, resolve high overlapped peaks, and calculates parameter values. An open-source code, EChem++ containing ANN is also available and can be freely obtained.\textsuperscript{26-28}

**Scope of this study**

The objective of this study was to test the concept of using a single artificial neural network on various electrochemical analyses. This study evaluated the feasibility of using artificial neural network to calculate parameter values from data of different electrochemical methods. The electrochemical techniques described here were polarography, LSV, and EIS. All tests used simulated data. Polarography used current density ($i$) versus electrode potential ($\varphi$) data of a nickel sulfamate plating bath, the LSV used cell voltage ($E$) versus current density ($i$) data of a proton exchange membrane (PEM) fuel cell, and the EIS used Bode plot data of an R($Rc$) equivalent circuit. Parameters calculated from polarography data were species concentrations. The exchange current density, Tafel slope, and internal resistance were calculated from the LSV data. Values of resistance and capacitor were calculated from the EIS data. Several network structures were also tested in this study.

**NUMERICAL PRINCIPAL AND PROCEDURE**

The artificial neural network used here was trained first with a given set of data. The network was then tested with another set of data. Following section describes the methods and equations used to generate both training data and testing data. This section describes (1) method to generate the simulated data for polarography, LSV, and EIS, (2) neuron, network structure, and neural input strategy used for this study, (3) the numerical training procedure, and (4) the numerical testing procedure.

**Generation of training and testing data**

All the data used in this study were computer generated. Equation (1) derived by Parry and Osteryoung,\textsuperscript{29} was used to generate the polarography data.

$$i_p = \sum_j \frac{C_j}{2b_j} e^{-\frac{a_j}{b_j} \varphi}$$  \hspace{1cm} (1)

Here “$i_p$” was the polarographic current density, $\varphi$ was the electrode potential, $C_j$ was the concentration of species “$j$”. Both $a_j$ and $b_j$ were the characteristic parameters of species “$j$”. For a given species in a given solution, the value of $a_j$ and $b_j$ were constants.

The LSV model used equation (2) to simulate the output cell voltage ($E_{cell}$) at various output current densities ($i_{cell}$) of a PEM fuel cell.\textsuperscript{2}

$$E_{cell} = k_1 - k_2 \log(i_{cell}) - k_3 i_{cell}$$  \hspace{1cm} (2)

The “$k_1$” and “$k_2$” were the kinetic parameter of Tafel equation. The value of “$k_1$” included the cell open voltage and exchange current density. The $k_3$ was the Tafel slope. The “$k_3$” represented the internal resistance of the cell.

The EIS model uses equations (3 – 6) to simulate the EIS data of a R($Rc$) circuit. Both real part ($Z_{re}$) and imaginary part ($Z_{im}$) of the EIS data were calculated as a function of frequency ($f$) in equation (3) and (4). The absolute value of impedance ($|Z|$) and phase angle ($\theta$) were then calculated from equations (5) and (6).

$$Z_{re} = R_{cell} + \frac{R_{e}}{1 + (2\pi f R_{e} C_{el})^2}$$  \hspace{1cm} (3)

$$Z_{im} = \frac{2\pi f R_{e} C_{el}}{1 + (2\pi f R_{e} C_{el})^2}$$  \hspace{1cm} (4)

$$|Z| = \sqrt{Z_{re}^2 + Z_{im}^2}$$  \hspace{1cm} (5)

$$\theta = \tan^{-1}(Z_{im} / Z_{re})$$  \hspace{1cm} (6)

**Neuron, network, and inputs in this work**

In most network applications, the neural outputs were bi-level outputs (1/0, or +1/-1). This study used network to determine the value of individual parameters. Depending on the type of electrochemical signal, this required output signal to be any number ranged from 0 to $\infty$. The sigmoid
function was used as the activation function for all the neurons. The sigmoid function \( g(h_i) \) was defined as

\[
g(h_i) = \frac{1}{1 + e^{-\beta h_i}}
\]  

(7)

Here \( h_i \) was the sum of all the inputs to neuron “i.” The \( \beta \) was an adjustable parameter. The shape of \( g(h_i) \) was strongly dependent on the value of \( \beta \). The sigmoid function became a step function at large \( \beta \) and became a linear function at small \( \beta \). The linear function had the same sensitivity over the entire input data range so that the function was not too sensitive to the input in certain ranges and under sensitive to the input in other ranges. However, a linear function could not confine the output value within 0 and 1. A sigmoid function at small \( \beta \) value had the properties closer to linear function and had the properties that confine the output between 0 and 1.

A single-layer feed-forward network was studied first to evaluate the influences of training cycle and activation function on the network properties. This network contained 101 inputs, 3 neurons, and 3 outputs. The structure of this network was abbreviated as 101-3. The input/output number chosen so that all the simulated data contained 101 data points calculated from the polarography, LSV, and EIS equations (1-6). The network calculated the value of three parameters from polarography, LSV, or EIS data. Backpropagation was used as the learning rule. Sigmoid function was used as the activation function for each neuron. This study also evaluated other network structures, they were two-layer networks (101-71-3, 101-51-3, 101-31-3), three-layer network (101-51-51-3, 101-71-31-3, 101-31-71-3), and a four-layer network (101-71-51-21-3). For each model (polarography, LSV, and EIS), ten data sets were used to train and to test all networks. The other ten data with 10% of noise level were used to test these networks mentioned.

Each set of training data contains input data and answers. For a training set \((\mu)\), the input data \( (i_{i}^{\mu}) \) were generated from equations (1-6). The answers \( (A_{1}, A_{2}, A_{3}) \) were the parameter used in equations (1-6). They were \( C_{1}, C_{2}, \) and \( C_{3} \) for polarography, \( k_{1}, k_{2}, \) and \( k_{3} \) for LSV, \( R_{c1}, R_{c2}, \) and \( C_{21} \), for EIS. Fig. 1 depicted the inputs for polarography, LSV, and EIS. Polarography input data were current densities \( "i_{p}" \) in equation (1) at predefined electrode potentials \( \phi \) as shown on Fig. 1a. Network calculated the polarography answers, the species concentrations \( C_{1}, C_{2}, \)
and C3. The LSV input data were cell voltages “E_{cell}” in equation (2) at predefined current densities as shown on Fig. 1b. Network calculated the LSV answers, the kinetic parameters of k1, k2, and k3. The EIS input data were the impedances |Z| and phase angles \( \theta \) in equation (5-6) at predefined frequency as shown on Fig. 1c. Network calculated the EIS answers, the resistance and capacitance, R_{e1}, R_{e2}, and C_{e1}, of the equivalent circuit R(RC).

Fig. 2a-2d were typical input data with various parameter values for polarography (Fig. 2a), for LSV (Fig. 2b), and for EIS (Fig. 2c and 2d). For polarography, curves marked #1, 2, 3, and 4 on Fig. 2a were simulated data measured of species 1, 2, and 3 in nickel sulfate plating bath at different concentrations. For LSV, curves marked #1, 2, 3, and 4 on Fig. 2b were simulated data of PEM fuel cell at different operating conditions, such as temperatures, humidification, gas flow rate, etc. For EIS, curves marked #1, 2, 3, and 4 on Fig. 2c and 2d were simulated data of an equivalent circuit with different R and C values.

**Training procedure**

The purpose of network training was to adjust the weights of the network so that the network outputs match the corrected answers. The network used many sets of training data so that the weights in the network had their proper values. Training used back propagation algorithm. Two-layer artificial neural network as shown on Fig. 3, was used here as an example to describe the training procedure. For a given training set \((\mu)\), it contained both the input data \((i_1^{\mu} \text{ to } i_{101}^{\mu})\) and the answers \((A_1^{\mu}, A_2^{\mu}, A_3^{\mu})\).

Initially, the values of network weight were random number between -0.5 and 0.5. The output from the neurons in layer 1 \((V_j^{\mu})\) were calculated for a given set \((\mu)\) of input data \((i_k^{\mu})\).

\[
V_j^{\mu} = g(h_j), \quad h_j = \sum_k w_{jk} i_k^{\mu}
\]  

(8)

Here \(w_{jk}\) were the weights of the first layer. Outputs from the neurons in layer 2 \((O_i^{\mu})\) were then calculated from \(V_j^{\mu}\).
Here \( w_{ij} \) were the weights of the second layer.

After the outputs of all the data sets were calculated, the weights of the last layer neurons were then adjusted with \( w_{ij} \) by following equation.

\[
\delta_j^m = g'(h_j^m) \sum_i w_{ij} \delta_i^m
\]  
(13)

Here \( g'(h_j^m) \) was the derivative of the activation function. The \( h_j^m \) was the sum of all the products of \( i_k^m \) and \( w_{ik} \). The \( i_k^m \) were the inputs. After all the weights were adjusted, the network outputs \( O_j^m \) were then calculated based on these new weights in the forward direction (solid lines in Fig. 3). The new outputs were then used to calculated the weight adjustments \( \Delta w_{ij} \) and \( \Delta w_{jk} \) in the backward sequence (dashed lines in Fig. 3). This procedure repeated until the values of weight adjustments \( \Delta w_{ij} \) and \( \Delta w_{jk} \) were reduced to a certain tolerance. When both the \( \Delta w_{ij} \) and \( \Delta w_{jk} \) were reduced to the tolerance level, we say the network was trained (or converged).

During the training cycle, network minimized an objective function, the network square error \( H \), error among outputs and corrected outputs. It was defined as equation (14).

\[
H = \frac{1}{2} \sum_{i} \sum_{j} (O_j^m - A_j^m)^2
\]  
(14)

Here \( O_j^m \) was the output from neuron “i” of input data set \( j \). The \( A_j^m \) was the parameter value used by the model to calculate the input data set \( j \). The network square error was the sum of the network output errors \( (O_j^m - A_j^m) \) from all neurons for all input data sets. A zero network square error means that the network output were perfectly match the parameter values. When \( H = 0 \), no weight adjustment was needed. The goal of network training was to adjust the network weights so that the network square error, \( H \) was minimized.

Testing procedure

The performance of trained network was evaluated with ten testing data sets. A 10% noise level generated with a random number was added to the each one of the testing data. These noisy testing data were used to evaluate the accuracy of the network.

RESULTS AND DISCUSSION

Network square error

The contour of the network square error (square error vs. \( h_j \)) is important in the minimization process (or training
A steep and narrow minimum contour means that the training was very sensitive to where the training process started (the initial values of the network weights). If it starts away from the minimum, it might not find the minimum. A broad and shallow square error contour means that the initial values of the network weights were not critical. The network training eventually and slowly reaches the minimum.

A simple 2-1 network as shown on Fig. 4a was used to examine the dependence of network square error on its weight. Disregard the actual values of the inputs \(i_1\) and \(i_2\) and the weights \(w_1\) and \(w_2\), the network square error was examined as a function of \(h_i\) at different values of \(\beta\). The region under consideration was marked by the dashed box on Fig. 4a. To simplify the study, we assumed only one set of input data and one output. For an output of 0.7, the network square error was \(\left[ g(h_i) - 0.7 \right]^2 / 2\).

For a given set of input, \(h_i\) was depending on the value of network weights. Fig. 4b shows the dependence of square error on \(h_i\). A large value of \(\beta\) generated a steep square error well and a small \(\beta\) generated a shallow square error well. Since the value of \(h_i\) was unknown, a steep square error well could cause the network training to be trapped at a plateau and never reached the square error minimum. A small \(\beta\) would ensure that the network converged to a square error minimum.

**Effects of training parameters on network square error**

After examined a simple 2-1 network, the properties of a 101-3 network was examined. This network had 101 inputs, 3 outputs, and 3 neurons. The dependence of network square error on training parameters was examined first. Training parameters include training cycle, the \(\beta\) value of the sigmoid function in equation (7), and the learning rate, \(\eta\) value of equation (10).

**Training Set**

Training data sets of polarography, LSV, and EIS were used to examine the 101-3 network properties. Network training used ten sets of data each for polarography, LSV, and EIS, respectively. Equations mentioned in section II generated these data sets. Table 1 listed parameter values used by these equations. The first network property examined was the training cycle. The training cycle was the number of back-propagation performed before the network was converged (trained). The network square error was the indicator to examine the convergence of network.

**Training Cycle**

Fig. 5 was the plot of square error of the 101-3 network for polarography, LSV, and EIS. The square error was plotted as a function of the logarithm of the training cycle. Network energies were evaluate at the \(\beta\) value of 0.2 and at the learning rate (\(\eta\)) of 1, 5, and 10. Fig. 5a and 5b were the results from polarography and LSV training data, respectively. Fig. 5c was the results from EIS training data. Fig. 5d was the results from EIS training data with modification. Typical EIS training data contained two sets of information as shown on Fig. 1c. One was the phase angle, and the other one was the absolute impedance. The value of phase angle might be out weight the value of impedance. The EIS modification was made by multiplying all the data of phase angle with a factor of 0.01. This placed the value of phase angle and impedance in the same order of magnitude.

In general, the network square error decreased as the training cycle increased. It reduced to a satisfactory level with a training cycle greater than 100,000. Only exception was the results from EIS training data without any modification (Fig. 5c). Large gap between the phase angle and im-
The Effects of $\beta$ and $\eta$ on Network Square Error

Fig. 6 presented the effects of $\beta$ and $\eta$ on the square error of the 101-3 network. Fig. 6a, 6b, and 6c were the results for polarography, LSV, and modified EIS data, respectively. The training cycle was kept at $1 \times 10^6$ for all the calculations. For a given $\eta$, network reached lower square error at small $\beta$ than at large $\beta$. This was because of the network square error contour at the vicinity of minimum becomes very steep and narrow when the $\beta$ was large. In gen-

<table>
<thead>
<tr>
<th>Data Set</th>
<th>C_1</th>
<th>C_2</th>
<th>C_3</th>
<th>K_1</th>
<th>K_2</th>
<th>K_3</th>
<th>R_{e_1}</th>
<th>C_{e_1}</th>
<th>R_{e_2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.15</td>
<td>0.05</td>
<td>1</td>
<td>0.82</td>
<td>0.035</td>
<td>0.164</td>
<td>0.08</td>
<td>0.3</td>
<td>0.03</td>
</tr>
<tr>
<td>2</td>
<td>0.10</td>
<td>0.10</td>
<td>1</td>
<td>0.7</td>
<td>0.035</td>
<td>0.164</td>
<td>0.08</td>
<td>0.1</td>
<td>0.03</td>
</tr>
<tr>
<td>3</td>
<td>0.05</td>
<td>0.15</td>
<td>1</td>
<td>0.766</td>
<td>0.05</td>
<td>0.164</td>
<td>0.16</td>
<td>0.3</td>
<td>0.03</td>
</tr>
<tr>
<td>4</td>
<td>0.15</td>
<td>0.10</td>
<td>1</td>
<td>0.766</td>
<td>0.02</td>
<td>0.164</td>
<td>0.16</td>
<td>0.1</td>
<td>0.03</td>
</tr>
<tr>
<td>5</td>
<td>0.15</td>
<td>0.15</td>
<td>1</td>
<td>0.766</td>
<td>0.035</td>
<td>0.2</td>
<td>0.08</td>
<td>0.3</td>
<td>0.01</td>
</tr>
<tr>
<td>6</td>
<td>0.10</td>
<td>0.05</td>
<td>1</td>
<td>0.766</td>
<td>0.035</td>
<td>0.12</td>
<td>0.08</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td>7</td>
<td>0.10</td>
<td>0.15</td>
<td>1</td>
<td>0.08</td>
<td>0.035</td>
<td>0.12</td>
<td>0.16</td>
<td>0.3</td>
<td>0.01</td>
</tr>
<tr>
<td>8</td>
<td>0.05</td>
<td>0.05</td>
<td>1</td>
<td>0.7</td>
<td>0.05</td>
<td>0.22</td>
<td>0.16</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td>9</td>
<td>0.05</td>
<td>0.10</td>
<td>1</td>
<td>0.75</td>
<td>0.045</td>
<td>0.2</td>
<td>0.16</td>
<td>0.3</td>
<td>0.05</td>
</tr>
<tr>
<td>10</td>
<td>0.183</td>
<td>0.074</td>
<td>1</td>
<td>0.766</td>
<td>0.035</td>
<td>0.164</td>
<td>0.08</td>
<td>0.1</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Fig. 5. Square error of a 101-3 network as a function of training cycle. (a) for polarography training data, (b) for LSV training data, (c) for EIS training data without modification, and (d) for EIS training data with modification.
eral, the β value smaller than 0.1 did not reduce the square error further.

The learning rate, η changed from 0.5 to 10 on Fig. 6. For a large value of η (η = 10), the network square error reached convergence only at small β value (~ 0.1). For a small value of η (η = 0.5), the network square error reached convergence at large β value (0.3-0.5). A small η and a small β value were preferred for network convergence.

**Effects of Neuron Number on Network Square Error**

Fig. 7 compared the of two-layer artificial neural networks with various neuron numbers in the hidden layer. These networks included 101-3-3, 101-5-3, 101-11-3, 101-21-3, 101-31-3, 101-51-3, and 101-71-3. One-layer network 101-3 was also included for comparison. The 101-3, 101-3-3, 101-5-3 networks had higher square error than others. Network energies were less than $1 \times 10^{-3}$ for neuron number in the hidden layer equal or greater than 11. Networks 101-11-3, 101-21-3, 101-31-3, 101-51-3, and 101-71-3 had lower square error than others. However, there was no defined trend among those networks.

**Network Square Error with Different Network Structure**

The network square error of eight network structures was also calculated. They were two-layer networks (101-71-3, 101-51-3, 101-31-3), three-layer network (101-51-51-3, 101-71-31-3, 101-31-71-3), and two four-layer network (101-71-51-21-3 and 101-41-31-30-3).

Each one of the network was learned with ten sets of learning data first. Weight matrices of individual network structures were calculated during learning. Learning was carried out with a training cycle of $1 \times 10^6$, a learning rate of 1.0, and β of 0.2. Then the network square error was calculated with another ten sets of testing data. These testing data were the same as the learning data, except all the testing data were superimposed with ± 10% of noise. Fig. 8 was the calculated network square error results of polarography, LSV, and modified EIS for various network structures. The square error for polarography was higher than the square error for LSV and EIS. To plot all the square error together on one diagram, the upper part scale was dif-

---

**Fig. 6. Square error of a 101-3 network as a function of training cycle. (a) for polarography training data, (b) for LSV training data, (c) for EIS training data without modification, and (d) for EIS training data with modification.**
different from the lower part scale on Fig. 8.

First group of comparison was two-layer network with different number of neuron in the hidden layer (101-71-3, 101-51-3, and 101-31-3). Difference of the two-layer network was different neuron number at the first layer (71, 51, and 31). The 101-51-3 network had the lowest square error for polarography and LSV. The 101-31-3 had the lowest network square error for EIS. Second group of comparison was three-layer network with same total number of neuron (102) in the hidden layer (101-51-51-3, 101-71-31-3, 101-31-71-3). The 101-71-31-3 had the lowest square error for LSV. The 101-51-51-3 had the lowest network square error for polarography and for EIS. Third group of comparison was the four-layer network versus three-layer network. The total neuron number in the hidden layer was a constant of 102 for the four-layer network (101-41-31-30-3), and three-layer network.

In both Figs. 7 and 8, the square errors of LSV were much higher than the square error of polarography and EIS data. This was due to higher average input value of LSV than average input value of polarography and EIS data as given on Fig. 2. The input value of LSV data was ranged between 0.3 and 1.0. The input value of polarography data was ranged between 0 and 1; however half of the data was zero. The EIS phase angle data were in the range between 0 and 90. They were multiplied by a factor of 0.01 before input to the network. Among all the networks on Fig. 8, the network with lowest square error was 101-51-3 for polarography, 10-51-3 for LSV, and 101-71-51-21-3 for EIS, respectively. There was no consistent result among the network structure.

CONCLUSIONS

It is feasible of using a single artificial neural network
to analyze various analytical data by using corresponding weight matrices. To produce accurate results, the network requires high training cycle, small $\beta$ value of sigmoid function, and small learning rate ($\eta$). The EIS data contains two types of input, the phase angle and the absolute impedance at predefined frequencies. In order for the network to reach convergence, values of these two inputs need to adjust into the same order of magnitude. The network accuracy increases as the number of network neuron increased. Network structure plays minimum effects on the network accuracy.

ACKNOWLEDGEMENTS

The author wishes to thank the National Science Council of Taiwan (NSC-99-2623-E-239-001-ET) for the financial support of publishing this work.

Received March 22, 2010.

REFERENCES