Artificial neural network study of the electrical conductivity of mould flux

How to cite:

© 2021 Rongshan Qin

https://creativecommons.org/licenses/by/4.0/

Link(s) to article on publisher’s website:

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data policy on reuse of materials please consult the policies page.
Artificial neural network study of the electrical conductivity of mould flux

Rongshan Qin

To cite this article: Rongshan Qin (2021) Artificial neural network study of the electrical conductivity of mould flux, Materials Science and Technology, 37:18, 1476-1482, DOI: 10.1080/02670836.2021.2016269

To link to this article: https://doi.org/10.1080/02670836.2021.2016269

© 2021 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

Published online: 21 Dec 2021.

Submit your article to this journal

Article views: 606

View related articles

View Crossmark data
Artificial neural network study of the electrical conductivity of mould flux

Rongshan Qin

School of Engineering and Innovation, The Open University, Milton Keynes, UK

ABSTRACT
The electrical conductivity of mould flux with chemical constitution of CaO-SiO₂-Al₂O₃-NaO-K₂O-MgO-CaF₂-Cr₂O₃-FeO-MnO has been investigated. The assessed database contains one unitary, five binary, nine ternary, four quaternary, two quinary, two senary and one octonary subsystems. Each constitutional component is in connection to another via some direct or indirect links. A multilayer artificial neural network method was developed and implemented in the database. The work provides a method to calculate the relationships between the composition, temperature and electrical property of the mould flux within the defined parameter ranges. The results have been validated against those experimental data that are not included in the training of the neural networks.

ARTICLE HISTORY
Received 24 August 2021
Revised 29 November 2021
Accepted 5 December 2021

KEYWORDS
Electrical conductivity; mould flux; multilayer neural network; continuous casting

Introduction
Many steelmaking companies are using a mould flux with chemical constitution of CaO-SiO₂-Al₂O₃-NaO-K₂O-MgO-CaF₂-Cr₂O₃-FeO-MnO to cast stainless steels in continuous casting mould. The mould slag contains high concentration of fluorine to reduce viscosity and solidification temperature [1]. Cr₂O₃-FeO-MnO can either pre-exist in mould powder or enter to liquid slag from oxidation of alloying elements in liquid stainless steel at casting mould and tundish [2]. The electrical conductivity of this system has never been assessed systematically although the data for its subsystems are notably rich.

The primary driving force to study the electrical properties of mould powder is for electroslag remelting processing [3,4]. This has, therefore, attracted considerable experimental measurement activities [3–6], theoretical modelling [7,8] and data assessment [9] for several subsystems. The recent environment regulation on carbon neutral steelmaking promotes the electrification of continuous casting. Electric field affects materials segregation [10], viscosity [11], distribution of oxide inclusions [12] and surface roughness in the cast mould [13]. This demands an analytical expression to represent the relationship between the chemical constitution, temperature and electrical conductivity of the system. The aim of the present work was to provide a mean to calculate the constitution and temperature-dependent electrical conductivity of the mould flux materials.

This work uses artificial neural network to approach the target. The method relies on available data to extrapolate values in the unknown parameters’ range [14]. Ideally, a relationship between the electrical conductivity, constitution and temperature should be derived from micro-mechanisms [15]. The previous theoretical modelling for the electrical conductivity of mould slag has been based largely on an assumption that electrical conduction is carried out by the moving ionised atoms. Obviously, viscosity affects the mobility of ionised atoms and hence plays an important role in electrical conduction. The basicity of mould slag affects the amount and length of silicate chains. This lays a foundation for the optical basicity model to calculate the electrical conductivity of mould slag [7]. On the other side, the interactions between constitutional components influence the mobility of atoms. This forms a basis to use particle interaction to calculate electrical conductivity [8]. For the oxides which can conduct electricity by electronic means, such as FeO, the above-mentioned theoretical models stray away [2]. The data-based artificial neural network has proved to be an effective solution to provide alternative solution [9].

The artificial neural network is different from data fitting such as the least square method. The former prevents overfitting but the latter seeks best fitting to data. The recent development of artificial intelligent learning enables the method to reduce its dimension according to the conservative laws that are hidden in
the data [16,17]. Artificial neural network has potential to indicate some physics natures buried in the big data. This work intends to provide a method to calculate the constitution and temperature-dependent electrical conductivity of the mould flux.

**Artificial neural network**

A supervised multilayer artificial neural network with back propagation learning algorithm has been coded for the present purpose. The network has an input layer containing 11 units to record the constitutional compositions and temperature. The output layer has 1 unit to provide the computational result for electrical conductivity. There are T-1 hidden layers each containing numerous units. The architecture of 2-layer network is illustrated schematically in Figure 1, where the hidden layer has n-units. The mapping function for 2-layer neural network in the present work is defined as

\[ x_j = \sum_{i=1}^{10} \omega_{ji}^{(1)} c_i + \omega_{ji}^{(1)} T + b_{ji}^{(1)} \]  
\[ u_j = 1/(1 + e^{-5}) \]  
\[ y = \sum_{j=1}^{n} \omega_{j}^{(2)} u_j + b_{j}^{(2)} \]  
\[ \sigma = 1/(1 + e^{-y}) \]  

The mapping function for > 2 layers neural network can be manipulated in the same way. A code package to calculate up to 4-layer neural network has been developed by the author. \( \omega_{ji}^{(k)} \) is the weight factor between \( j^{th} \) unit in \( k^{th} \) layer and \( i^{th} \) unit in \( (k-1)^{th} \) layer, where \( 0^{th} \) layer is the input layer. \( b_{ji}^{(k)} \) is the bias for \( j^{th} \) unit in \( k^{th} \) layer. \( x_j \) is the activation of \( j^{th} \) unit in 1st layer \( u_i \) is its activation function. \( y \) is the activation of output layer.

The electrical conductivity \( \sigma \) is the activation function of \( y \).

The activation functions in Equations (1.2) and (1.4) are nonlinear and with output values between 0 and 1. Many other neural network models choose other functions such as \( f(x) = \tanh(x) \) [14], which has value between \(-1\) and \(1\), or Gaussian distribution [16]. It is important to normalise the data according to the format of activation functions. In the present model, all the electrical conductivity data are normalised to a value between 0 and 1. The numerical results are denormalised to get the true value.

Following the standard method in neural network calculation [18], the total difference is defined as

\[ E_D = \frac{1}{m} \sum_{l=0}^{m} (\sigma_{c}^{(l)} - \sigma_{c}^{(l)})^2 \]  

where \( m \) is the total number of sets of training data. \( \sigma_{c}^{(l)} \) and \( \sigma_{c}^{(l)} \) are the \( l^{th} \) training value and calculated value, respectively. To prevent neural network optimisation from overfitting the noise in training data, a regularisation is defined to minimise the total value of weight factor square as \( E_W = \sum_{i,j,k} (w_{ji}^{(k)})^2 \). The overall target function is defined as [18]

\[ E = \frac{1}{2} \beta E_D + \frac{1}{2} \alpha E_W \]  

where \( \beta \) and \( \alpha \) are coefficients. Their ratio reflects a balance between accuracy and simplicity. The intelligent learning is to perform a gradient descent minimisation of the total difference via [19]

\[ \Delta w_{ji}^{(k)} = -\eta \partial E/\partial w_{ji}^{(k)} \]  

where \( \eta \) is the learning rate. Equation (4) defines one of the learning methods that always works but unnecessarily the most effective method. In the following section, other learning methods will be used and compared with Equation (4). The analytical expression for each weight factor and bias can be obtained by back propagation of the partial difference of target function according to Equation (4).

**Data assessment**

The composition in the database is defined to be atomic percentage. The weight percentage data from literature has been converted according to their molar weight [3,5,6,21]. Renormalisation has been performed for those data that the original total composition does not come to 100% [22]. This is not rare in steel company become the initial constitutions of C and CO2 in mould powder are not counted in liquid slag. However, some data in literature has a total composition in excess of 100%. Those data are ignored entirely. For the data contain tiny fraction (3.00 wt-%) of NaO and K2O but without detailed information about their

![Figure 1. Schematic diagram illustrates 2-layer neural network.](image-url)
ratio [21]. An assumption of 50:50 is applied. This assumption does not affect the specification of the individual contribution from NaO and K₂O to the electrical conductivity because a significant amount of NaO and K₂O data from other literature has been applied to justify the assumption [23,24]. For the data that have been assessed previously [8], the assessed data instead of large amount of raw experimental data has been adopted to minimise noise. Some data are reported as scattered points in plotted figure. To get the values from the plotted figure with high accuracy, a java-based code package has been developed by the author to convert the figure to data. Some figures have been plotted artificially with scales in coordinates not proportional [22]. Only those data with clear indication of their values have been adopted. For the plotted continuous curves, only those points accompanied with experimental values are adopted. After the critical assessment, a database contains 752 sets of data with one unitary, five binary, nine ternary, four quaternary, two quinary, two senary and one octonary subsystems has been built up. The detailed subsystems are listed in Table 1. The parameter ranges are shown in Table 2.

The unit of temperature is Kelvin. The unit of electrical conductivity is Ω⁻¹·cm⁻¹.

### Neural network computation and results

The activation functions, as illustrated in Equations (1.2) and (1.4), have an output value between 0 and 1 for the activation between -∞ and +∞. However, σ = 0.066693 at y = -5 and σ = 0.993307 at y = 5, which indicate an extremely slow approximation to either 0 or 1. Based on this consideration, the training data for electrical conductivity is not normalised by the true minimum and maximum values in the database but multiplied by 0.7 to the minimum value and 1.3 to the maximum value. The input parameters for composition and temperature are all normalised to a value between 0 and 1 to ensure every input parameter has the same weight of contribution. The denormalisation and normalisation procedure followed the following equations

\[
\tilde{\sigma} = \frac{\sigma - 0.7\sigma_{\text{min}}}{1.3\sigma_{\text{max}} - 0.7\sigma_{\text{min}}} \\
\tilde{c}_i = \frac{c_i - c_{i,\text{min}}}{c_{i,\text{max}} - c_{i,\text{min}}} \quad (5.1)
\]

where \(\tilde{\sigma}\) and \(\tilde{c}_i\) are the normalised electrical conductivity and composition for \(i < 11\) and temperature when \(i = 11\). The initial values for all the weight factors and biases are assigned to a random float value between -5 and 5. A high-quality random number generator was coded according to a probability theory developed by Marsaglia et al. [25].

In neural network calculation, it has been noted that the artificial learning by means of Equation (4) in every time iteration does not help to find the weight factors and biases to achieve minimum total differences between the target values and calculated value. The weight factors soon adjust their values to minimise the overall target function \(E\) rather than the total difference \(E_D\). To overcome this problem, the regularisation term \(E_W\) is not included in each time iteration but replaced at the final step assessment. It is also found that the convergence rate is almost doubled by the following artificial learning mechanism, which agrees with the suggestion from Rumelhart et al. [19]

\[
\Delta w_{ji}^{(k)}(t + 1) = -\eta \frac{\partial E}{\partial w_{ji}^{(k)}} + \delta \Delta w_{ji}^{(k)}(t) \quad (6)
\]

where \(\delta\) is a coefficient. For the 2-layer neural network calculation with \(n = 16\) and \(\alpha = \beta = \eta = \delta = 0.5\), the

<table>
<thead>
<tr>
<th>Value</th>
<th>CaO</th>
<th>SiO₂</th>
<th>Al₂O₃</th>
<th>NaO</th>
<th>K₂O</th>
<th>MgO</th>
<th>CaF₂</th>
<th>Cr₂O₃</th>
<th>FeO</th>
<th>MnO</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>471</td>
</tr>
<tr>
<td>Max</td>
<td>72.24</td>
<td>60.81</td>
<td>43.24</td>
<td>29.21</td>
<td>6.00</td>
<td>20.00</td>
<td>0.523</td>
<td>34.80</td>
<td>20.00</td>
<td>2039</td>
<td></td>
</tr>
</tbody>
</table>
Table 3. The optimised weight factors and biases obtained by artificial neural network calculations.

<table>
<thead>
<tr>
<th>i</th>
<th>$\omega_{(1)}^{(1)}$</th>
<th>$\omega_{(2)}^{(1)}$</th>
<th>$\omega_{(3)}^{(1)}$</th>
<th>$\omega_{(4)}^{(1)}$</th>
<th>$\omega_{(5)}^{(1)}$</th>
<th>$\omega_{(6)}^{(1)}$</th>
<th>$\omega_{(7)}^{(1)}$</th>
<th>$\omega_{(8)}^{(1)}$</th>
<th>$\omega_{(9)}^{(1)}$</th>
<th>$\omega_{(10)}^{(1)}$</th>
<th>$\omega_{(11)}^{(1)}$</th>
<th>$\omega_{(12)}^{(1)}$</th>
<th>$\omega_{(13)}^{(1)}$</th>
<th>$\omega_{(14)}^{(1)}$</th>
<th>$\omega_{(15)}^{(1)}$</th>
<th>$\omega_{(16)}^{(1)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>7.042</td>
<td>1.479</td>
<td>1.048</td>
<td>-1.703</td>
<td>-6.276</td>
<td>3.248</td>
<td>-6.630</td>
<td>-2.839</td>
<td>-0.906</td>
<td>-4.354</td>
<td>-0.171</td>
<td>-3.031</td>
<td>0.807</td>
<td>1.810</td>
<td>-0.352</td>
<td>-5.788</td>
</tr>
<tr>
<td>3</td>
<td>-1.304</td>
<td>-2.414</td>
<td>-3.978</td>
<td>0.182</td>
<td>2.548</td>
<td>2.945</td>
<td>-1.225</td>
<td>-0.627</td>
<td>-3.282</td>
<td>-2.201</td>
<td>-1.870</td>
<td>-1.810</td>
<td>-1.620</td>
<td>-5.388</td>
<td>3.190</td>
<td>-3.816</td>
</tr>
<tr>
<td>4</td>
<td>-0.267</td>
<td>-3.132</td>
<td>-3.209</td>
<td>-7.334</td>
<td>-3.269</td>
<td>5.318</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.376</td>
<td>-2.656</td>
<td>-0.931</td>
<td>1.841</td>
<td>-0.312</td>
<td>3.175</td>
<td>-6.724</td>
<td>-1.493</td>
<td>-0.382</td>
<td>-2.201</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.860</td>
<td>-8.368</td>
<td>-5.901</td>
<td>6.286</td>
<td>-1.250</td>
<td>0.786</td>
<td>2.651</td>
<td>0.438</td>
<td>2.259</td>
<td>0.032</td>
<td>6.445</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-2.727</td>
<td>5.957</td>
<td>3.358</td>
<td>2.482</td>
<td>2.415</td>
<td>7.337</td>
<td>-1.195</td>
<td>-8.983</td>
<td>2.609</td>
<td>-0.273</td>
<td>-3.009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-0.862</td>
<td>1.249</td>
<td>5.116</td>
<td>5.106</td>
<td>-2.449</td>
<td>0.498</td>
<td>-3.733</td>
<td>-3.322</td>
<td>4.180</td>
<td>0.328</td>
<td>-1.869</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1.730</td>
<td>0.858</td>
<td>-0.612</td>
<td>-2.560</td>
<td>-0.201</td>
<td>-1.836</td>
<td>1.620</td>
<td>-6.329</td>
<td>-2.162</td>
<td>-1.012</td>
<td>1.505</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$b_{(1)}^{(1)}$</td>
<td>1.172</td>
<td>0.645</td>
<td>0.321</td>
<td>-2.985</td>
<td>4.332</td>
<td>-5.004</td>
<td>2.231</td>
<td>6.241</td>
<td>6.158</td>
<td>4.916</td>
<td>-2.561</td>
<td>3.300</td>
<td>0.997</td>
<td>5.489</td>
<td>-3.383</td>
<td>3.552</td>
</tr>
<tr>
<td>$b_{(2)}^{(1)}$</td>
<td>-6.009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. The evolution of total difference ($E_D$), regularisation term ($E_W$) and target function ($E$) vs. iteration steps for 2-layer neural network with hidden layer containing 16 units.

Figure 3. The evolution of total difference ($E_D$), regularisation term ($E_W$) and target function ($E$) vs. number of units in hidden layer after $1.6 \times 10^7$ iteration steps.

evolution of total difference, regularisation term and target function vs iteration steps are demonstrated in Figure 2. It shows that total difference drops sharply in the early stage (labelled by A), followed by slow drops (labelled by B) until a flat stage (labelled by C) to fluctuate around a minimum value. However, the regularisation term was increased slowly but monotonically until stage C. This is due to the early mentioned decision of not to include the minimisation of regularisation term in the time iteration. The target function has been reduced monotonically until the flat stage.

To determine the optimum number of units in the hidden layer in 2-layer neural network calculation, one has calculated the change of differences until $1.6 \times 10^7$ iteration steps for various number of units. The results are plotted in Figure 3. Although $E_D$ decreases when the number of units increases, $E_W$ demonstrates some optimised values. $E_W$ increases sharply when the number of units is away from the optimised one. The target

function, $E$, reveals an optimised value for the number of units in the hidden layer of the 2-layer neural network. Based on the results, $n = 16$ is chosen. It is worth mentioning that the local minimum at $n = 16$ for the curve of $E_D$ is out of expectation. To double check whether it is a numerical coincidence, the code and parameters were run at three different workstations but the results were very similar, given the fact that the initialisation of weight factors involves a random number generator which should be different at different computers. The smallest $E_D$ appeared at $9,012,000^{th}$ iteration step. The values for the weight factors and biases at this optimised condition are listed in Table 3. These values can be used to calculate the electrical conductivity of the system at any composition and temperature in the parameters' range.

The optimised weight factors and bias values have been implemented to calculate the electrical conductivity for 752 sets of compositions and temperature.
The results are plotted in Figure 4 and compared with the values in the database. Figure 4(a) shows the comparison at linear scale coordinates. The 45° line indicates the perfect agreement between the artificial neural network computational results and the value in database, where majority data are from experimental measurement and the rest from assessment based on experimental values. The figure shows good agreement. Owing to the wide range distribution of the electrical conductivity values from 0.016 to 23.771, which across three orders of magnitude, the comparison in logarithmic scale is shown in Figure 4(b). The data shows some almost evenly distribution around the 45° line, majority with absolute error below 5%. The largest absolute discrepancy appears in the lowest electrical conductivity end, as is circled in Figure 4(b). Those data are found

Figure 4. The electrical conductivity from numerical results by artificial neural network calculation vs. the value in the database: (a) in linear scale plotting, and (b) in logarithmic scale plotting.

Figure 5. Validation of neural network calculation by experimental data reported in literature [6,26,27]. (a) CaF$_2$-Al$_2$O$_3$ at 1773K, (b) CaF$_2$-Al$_2$O$_3$ at 1973K, (c) CaO-CaF$_2$ at 1773K, and (d) CaO-CaF$_2$ at 1873K.
all belong to CaO-SiO₂-Al₂O₃ subsystem at a temperature either in 1623 K or 1673 K, and was reported in one paper.

To validate the artificial neural network calculations, the optimised weight factors and bias values have been implemented to calculate two binary systems CaF₂-Al₂O₃ and Cao-CaF2 at different temperature. The results have been compared with the experimental results reported in various literature [6,26,27]. Figure 5 presents the results for (a) CaF₂-Al₂O₃ at 1773K, (b) CaF₂-Al₂O₃ at 1973K, (c) Cao-CaF2 at 1773K and (d) Cao-CaF2 at 1873K. These experimental data are not in database during the training of neural network. Figure 5 shows that the electrical conductivity obtained in the neural network calculations are within the fluctuation of various experimental measurements. It proves that the artificial neural network prediction for the electrical conductivity can be used to predict the change of electrical conductivity at various subsystems in different compositions and temperatures.

The artificial neural network and machine learning have many potential applications in steel metallurgy [28]. In the future, more works will be done to include other components to the system, such as NiO, TiO₂, MgF₂, BaF₂, BaO, ZrO, CaS. The availability of the new experimental measurement method for electrical conductivity enables to get more accurate data in other systems [29], which will help to build up database for training and validation of the neural networks. The future work can, hopefully, also include the effort to use the data and machine learning method to identify the main oxides that control the electrical conductivity of mould flux and the influence of temperature on the electrical properties, and to compare the results with the theoretical predictions [7,8,15].

Conclusions

- An electrical conductivity database for CaO-SiO₂-Al₂O₃-NaO-K₂O-MgO-CaF₂-Cr₂O₃-FeO-MnO liquid mould slag system has been built up.
- The database has been implemented to train an artificial neural network. It is found that the two-layer neural network with 16 units in hidden layer provides the minimum difference in target function. The optimised weight factors and bias values can be used to calculate the electrical conductivity of the system in a wide range of compositions and temperature.
- The numerical prediction has been validated by the experimental results reported in literature. Excellent performance of artificial neural network derivation has been proved.

Acknowledgements

This work was financially supported by the European Commission RFCS (No. 847269) and Engineering and Physical Sciences Research Council in UK (No. EP/R029598/1).

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/R029598/1]; European Commission RFCS (Research Fund for Coal and Steel) [grant number 847269].

ORCID

Rongshan Qin http://orcid.org/0000-0002-4464-9616

References


