Huuhtanen, Timo; Jung, Alex

**Anomaly Location Detection with Electrical Impedance Tomography Using Multilayer Perceptrons**

*Published in:*

*DOI:*
10.1109/MLSP49062.2020.9231818

Published: 23/09/2020

*Document Version*
Peer reviewed version

Please cite the original version:
ANOMALY LOCATION DETECTION WITH ELECTRICAL IMPEDANCE TOMOGRAPHY USING MULTILAYER PERCEPTRONS

Timo Huuhtanen, Alexander Jung

Aalto University, Department of Computer Science, Espoo, Finland

ABSTRACT

Electrical impedance tomography (EIT) does imaging by solving a nonlinear ill-posed inverse problem. Recently, there has been an increasing interest in solving this problem with artificial neural networks. However, a systematic understanding of the optimal neural network architecture for this problem is still lacking. This paper compares the performance of different multilayer perceptron algorithms for detecting the location of an anomaly on a sensing surface by solving the EIT inverse problem. We generate synthetic data with varying anomaly sizes/locations and compare a wide range of multilayer perceptron algorithms by simulations. Our results indicate that increasing the dimensions of the perceptron improves performance, but this improvement saturates soon. The best performance is achieved when using the multilayer perceptron for regression and Gaussian noise addition as the regularization method.

Index Terms— Electrical impedance tomography, multilayer perceptron, anomaly detection

1. INTRODUCTION

Electrical impedance tomography (EIT) [1] determines the internal structure of a 3-dimensional body or a 2-dimensional surface based on its conductivity distribution. EIT is important for a wide range of medical and industrial applications. It is also a potential technology for sensing surface applications, because of its non-intrusive nature and low implementation cost. Sensing surfaces are used for observing human activity [2], implementing human-to-machine interface (HMI) [3], or detection of failures of construction materials [4]. A typical task for a sensing surface is to find the location of an anomaly in the conductivity of the surface.

The EIT inverse problem is to find a function, which maps the measured voltages to the conductivity distribution, thus reconstructing the image formed by the conductivity values. This problem has traditionally been solved by linearization [1]. However, as the problem is nonlinear and severely ill-posed, linearized solutions do not give good results if the conductivity differs significantly from the conductivity in the point of linearization. As artificial neural networks (ANN) are universal function approximators [5], several studies of using ANNs for EIT have been published in recent years [6].

The EIT inverse problem has been solved with multilayer perceptrons (MLP) [7], [8], [9], [10], and with radial basis functions (RBF) [11], [12]. An alternative approach is to solve the inverse problem with traditional methods and use ANNs as post-processing stage [13], [14].

Although a large amount of research on ANNs applied to EIT has been carried out, a systematic study comparing the impact of ANN hyperparameters for the performance is still lacking. Most research papers aim at image reconstruction with good quality; to date, there is little published data on studies using EIT to solve more dedicated problems directly, such as detection of the anomaly location in our case. The authors of [9] use MLP with one hidden layer to estimate the location of an object, but their study is limited to only one ANN architecture and one set of hyperparameters.

In the present study, we select the quintessential class of ANN architectures, MLP, in the context of solving the EIT inverse problem applied to anomaly location detection. We study the impact of width and depth of the MLP into the performance of location detection and compare different regularization techniques. Additionally, we compare regression and classification approaches for this problem. This study attempts to find optimal ANN architecture and hyperparameter values for the EIT inverse problem.

The remaining part of this paper is organized as follows. Section 2 formulates the EIT inverse problem and the linearized solution. Section 3 describes the neural network architectures to be compared. The numerical experiments are discussed in Section 4. Section 5 presents a conclusion and discusses opportunities for future research.

2. EIT ALGORITHMS

In this section, we formulate the EIT inverse problem and describe a standard solution [1], [15] to be used as reference. EIT measures the distribution of electrical conductivity $\sigma$ of surface $\Omega$ using $n_e$ electrodes connected to the boundary of $\Omega$. The electrodes are used to inject currents and measure voltages. We expect the opposite current pattern to be used as illustrated in Fig. 1 (a); i.e. the electrodes on opposite sides
of the surface are used as current source and sink. The rest of the electrodes are used for voltage measurements. The roles of the electrodes are then changed, and measurements are repeated until all the electrodes have been cycled through. With this arrangement, the total number of voltage measurements in one frame becomes \( n_M = n_e (n_e - 4) \). The measured voltages are collected into a vector \( v \in \mathbb{R}^{n_M} \).

EIT forward problem amounts to calculating the estimates for the electric potential in \( \Omega \) assuming \( \sigma \) is known. The calculation can be done by modeling the surface as a finite element mesh with \( n_N \) triangular elements as illustrated in Fig. 2. Conductivity distribution is approximated as being constant within each element. The electric potential values for each element can then be calculated using a finite element model (FEM). From these values, we can calculate \( f(\sigma) \in \mathbb{R}^{n_M} \), which is the concatenation of voltage differences between consecutive electrodes calculated from the model; here \( \sigma \in \mathbb{R}^{n_N} \) is the vector of stacked conductivities for each mesh element. The EIT inverse problem amounts to recovering \( \sigma \) from \( v \) and \( f(\sigma) \) as

\[
\arg \min_{\sigma \in \mathbb{R}^{n_N}} \| v - f(\sigma) \|_2^2.
\]  

(1)

A feasible approach in anomaly detection is difference EIT, where the conductivity of an empty surface has been measured, and the difference between that and the conductivity with an anomaly on the surface is used to determine the anomaly location. The function \( f \) can then be linearized around a reference conductivity \( \sigma_0 \) and measurement values \( v_0 \) as

\[
f(\sigma) = v_0 + J(\sigma - \sigma_0).
\]  

(2)

Here, \( J \) (\( J_{ij} = \frac{\partial f_i}{\partial \sigma_j} \)) is the sensitivity matrix or Jacobian, which we also get from FEM. We denote the differential conductivity \( \sigma - \sigma_0 \) at time \( t \) by \( x(t) \). The entry \( x_i(t) \) corresponds to the conductivity difference value at the \( i \)th triangular element of the inverse mesh. Inserting (2) into (1) and denoting \( y = v - v_0 \) allows to reformulate the EIT inverse problem as

\[
\arg \min_{\sigma \in \mathbb{R}^{n_N}} \| y - Jx \|_2^2.
\]  

(3)

This system is typically underdetermined \((n_N > n_M)\) and regularization is required. Standard choices for regularization are \( L_1 \) and \( L_2 \) and total variation (TV) regularization [16], [17]. Here we are interested in finding the location and do not care about the preservation of the shape or size of the anomaly in the image reconstructed from the values of \( x \). So,
we select $L_2$, which fits well into the detection of the center of the anomaly. The regularized version of (3) then becomes

$$\arg\min_{x \in \mathbb{R}^{nN}} \|y - Jx\|_2^2 + \lambda \|Rx\|_2^2.$$  
(4)

Here $\lambda$ is a hyperparameter, and $R$ is a regularization matrix containing some prior information about $x$. In the simulations, we use empirically found value $\lambda = 0.01$. Here we select the regularization proposed in [1] with $R = (\text{diag}(J^T J))^{0.5}$.

Solving the problem (4) with one step of the Gauss-Newton (GN) algorithm [1] yields

$$\hat{x} = (J^T J + \lambda \text{diag}(J^T J)^{0.5})^{-1} J^T y \quad (5)$$

Our actual problem is to find the location of the center of the anomaly, $r = [r_h, r_v]$ as illustrated in Fig. 1 (b). We can estimate $r$ by finding $\arg\max_i \{\hat{x}_i\}$, where $\hat{x}_i$ is the $i$:th component of $\hat{x}$, and taking the Euclidean coordinates of the corresponding mesh element of the inverse mesh.

3. MLP AND EIT INVERSE PROBLEM

3.1. EIT inverse problem as a regression problem

The EIT inverse problem aims at finding a function, which maps voltage measurements $v$ to the 2-dimensional coordinates of the anomaly center, $r$. Thus, we can treat the inverse problem as a regression or function approximation problem, where $v$ is the input, and $r$ is the output. Multilayer perceptrons with at least one hidden layer and linear output layer are known to be universal function approximators [5]. So, we try solving the function approximation problem directly by training a multilayer perceptron with simulated training data. To this end, we simulate MLPs varying the number of hidden layers and the number of neurons per layer to find the optimal regression architecture. The architecture is illustrated in Fig. 3, and the hyperparameters used in our trials are listed in Table 1.

Table 1: Key hyperparameters for MLP regression

| Input layer width | 192 |
| Hidden layer width | 20, ..., 320 |
| Hidden layer activation | ReLU |
| Output layer width | 2 |
| Output type | Linear |
| Optimizer | Adaptive moments (ADAM) |
| Loss function | Mean squared error (MSE) |
| Epochs | 1000 (with early stopping) |

3.2. EIT inverse problem as a classification problem

An alternative approach to solving the location of an anomaly is to formulate the problem as a classification problem. The surface $\Omega$ has been split into a finite number of mesh elements as described in Section 2 and illustrated in Fig. 2, and the center of the anomaly is located in one of these mesh elements. We can then consider the problem of finding the location as a classification problem of finding the mesh element where the anomaly center is located. There will be then $n_N$ classes, where each class corresponds to the situation that the anomaly center is located within that mesh element. The Euclidean coordinates of the corresponding mesh element can then be used as the estimate of $r$.

The architecture for the simplest possible perceptron classification for anomaly center detection is the single-layer perceptron (SLP) illustrated in Fig. 4; the used hyperparameters are listed in Table 2. The classification gets the voltage measurements $v$ as input. The classification has $n_N$ outputs, each of which is a softmax function indicating the probability that the anomaly center is located in the corresponding inverse mesh element. The location of the anomaly is achieved by finding the output with maximum value, taking the Euclidean coordinates of the center of that element, and using them as an estimate for the anomaly location.

Table 2: Key hyperparameters for SLP classification

| Input layer width | 192 |
| Output layer width | 287 |
| Output type | Softmax |
| Optimizer | Adaptive moments (ADAM) |
| Loss function | Sparse categorical cross-entropy |
| Epochs | 1000 (with early stopping) |

3.3. Regularization

Regularization is used for the training of ANNs to avoid overfitting. We compare different regularization techniques as
part of the present study. A standard approach is to use $L_1$ or $L_2$ regularization in the hidden layers of MLP, and control the regularization by the weight of the penalty term as the regularization-controlling hyperparameter. An alternative technique is dropout, where units of the neural network are randomly dropped out during the training phase. Dropout is controlled by the rate of the dropping. Regularization can also be achieved by adding Gaussian noise in the input data and using the standard deviation (SD) of the added noise as hyperparameter. Principal component analysis (PCA) can also be used for dimensionality reduction and thus for reduction of the risk of overfitting [7], [8]. The controlling hyperparameter is the number of principal components retained in the data after PCA. In the present study, we compare these regularization techniques with controlling hyperparameter values summarized in Table 3.

**Table 3: Hyperparameters controlling regularization**

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1/L_2$ penalty weight</td>
<td>0, 0.1, 0.01, 0.001</td>
</tr>
<tr>
<td>Dropout rate</td>
<td>0, 0.2, 0.5</td>
</tr>
<tr>
<td>SD of added noise</td>
<td>0, 0.01, 0.0316, 0.1, 0.316</td>
</tr>
<tr>
<td>Nr. of PCA comp.</td>
<td>10, 20, 30, 40, 60, 80, 120, 160, 192</td>
</tr>
</tbody>
</table>

4. RESULTS AND DISCUSSION

We compared the algorithms by simulations using pyEIT [15] and Keras [18] libraries with generated, synthetic test data. These data consisted of conductivity values on a unit-circle-shaped sensing surface with homogenous conductivity $\sigma_0 = 1$, and a circular anomaly with conductivity $\sigma = 10$ as illustrated in Fig. 5. As the selected inverse mesh determines the resolution for anomaly location detection, we placed the center of the anomaly in one of the elements of the inverse mesh and then varied the location to go through all the elements of the inverse mesh. The anomaly diameter was randomly varied in the range $d \in [0.03, 0.5]$. This set of conductivity values was run through pyEIT simulation model for the EIT forward problem to generate a frame of voltage measurements, $v$. This voltage frame generation procedure was repeated 200 times for each anomaly location to create a dataset of $200 \times 287 = 57400$ frames. The frames were then split (80%/20%) to training and validation datasets and used with the known anomaly locations as labels to train and compare the algorithms. To compare the performance of different MLP algorithms, we trained MLP regression and classification algorithms using various hyperparameter values and selected the combinations resulting in the lowest validation loss (MSE). For these MLP algorithms, we calculated the average position error (PE) [1], which is the Euclidean distance between the estimated and actual anomaly locations, to be used as the final measure of performance.

We compared the regularization techniques by first training a 4-layer MLP (layer widths 320/320/320/80) with noise-free training data and then tested the performance using the validation data with added noise (-20dB compared to the input signal). We repeated this with the regularization techniques mentioned in Section 3.3 and all the combinations of hyperparameter values listed in Table 3. Fig. 6 illustrates the cases where only one regularization technique at a time is in use with the best hyperparameter value found. The results show that $L_1$, $L_2$, and dropout with any hyperparameter value yield worse performance when compared to the non-regularized case. Dimensionality reduction with PCA using the best hyperparameter value did improve the performance only slightly, but the best result was achieved by adding Gaussian noise with the standard deviation 0.1 giving 25% improvement to the validation loss (MSE) compared to the non-regularized case. Hence, we selected noise addition as the regularization to be used in the rest of the comparisons in this study.

We studied the impact of the number of layers and layer widths by simulating MLP regression with 1-4 hidden layers and varying the layer widths according to Table 1. In Fig. 7, the validation loss is depicted as a function of the number of neurons for an MLP regression algorithm with a single hidden layer. The validation loss and PE values for the best MLP regression algorithms with 1-4 hidden layers (R1, R2, R3, R4)
Fig. 6: Comparison of regularization techniques: validation loss of an MLP regression (4 hidden layers; widths 320/320/320/80) using different regularization techniques with the best hyperparameter values found.

as well as those of the SLP classification (CL) are shown in Fig. 8. Table 4 summarizes the performance results of the best MLP algorithms as well as those of the traditional GN algorithm (5), which we used as a reference.

Fig. 7: Validation loss as a function of the width of the single hidden layer in a MLP.

![Graph showing validation loss as a function of the width of the single hidden layer in a MLP.]

Fig. 8: MSE and PE on the validation set as a function of the number of layers (MLP regression).

![Graph showing MSE and PE on the validation set as a function of the number of layers (MLP regression).]

Table 4: Performance of the best MLP architectures.

<table>
<thead>
<tr>
<th>Algorithm (hidden layer widths)</th>
<th>PE (train)</th>
<th>Loss (validation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GN</td>
<td>0.1112</td>
<td>0.0310</td>
</tr>
<tr>
<td>R1 (200/200/-/-)</td>
<td>0.0469</td>
<td>0.0049</td>
</tr>
<tr>
<td>R2 (280/240/-/-)</td>
<td>0.0333</td>
<td>0.0043</td>
</tr>
<tr>
<td>R3 (320/320/80/-)</td>
<td>0.0273</td>
<td>0.0039</td>
</tr>
<tr>
<td>R4 (320/320/320/80)</td>
<td>0.0203</td>
<td>0.0038</td>
</tr>
<tr>
<td>CL</td>
<td>0.0298</td>
<td>0.0099</td>
</tr>
</tbody>
</table>

Overall, all the compared ANN architectures can outperform the traditional GN algorithm. The best regression and classification MLPs yield at least 50% smaller PE and validation loss values compared to the standard GN method.

Trials with the regression algorithms show that increasing the number of layers and layer widths improves the results, but the improvement saturates quite soon. As seen in Fig. 7, increasing the number of neurons improves the performance up to 200 neurons. With multiple hidden layers, the impact of layer width is less clear: the same performance can be achieved with several combinations of layer widths, and the performance improvement of increasing the layer width saturates after about 100 neurons. From Fig. 8, adding hidden layers to the MLP improves the performance, but the impact is negligible after 2 hidden layers (when observing MSE) or after 4 hidden layers (when observing PE).

When comparing the validation losses of regression and classification approaches, it can be observed that regression with 1-4 hidden layers yields better performance. However, the PE results of a SLP classification are better than those of the best regression algorithms with 1 or 2 hidden layers. One possible explanation for the result is that our dataset has true anomaly locations at centers of the inverse mesh elements, which results in zero PE if the classification result is correct. As PE essentially calculates the mean absolute error of the location, this criterion favors this sparse nature of the classification error data.

5. CONCLUSIONS

We have applied ANNs to solve the EIT inverse problem. This approach outperforms traditional methods that are based on linearization. The use of ANNs works for regression and classification tasks associated with locating anomalies on a 2D sensing surface. We stress that our study was limited to detecting a single circularly-shaped anomaly, so a natural pro-
gression of this work is to extend the analysis detecting several anomalies with varying shapes. The gain achieved with dimensionality reduction also suggests considering alternative training strategies for the ANN such as autoencoding.

6. REFERENCES


