A Spatial Adaptation of the Time Delay Neural Network for Solving ECGI Inverse Problem

Amel Karoui, Mostafa Bendahmane, Nejib Zemzemi

A Spatial Adaptation of the Time Delay Neural Network for Solving ECGI Inverse Problem

Amel Karoui[1,2,3], Mostafa Bendahmane[1,2,3], and Nejib Zemzemi[1,2,3]

[1] University of Bordeaux, IMB, Bordeaux, France
[2] National Institue of Mathematics and Informatics, Inria Bordeaux, France
[3] IHU-Lyric , Bordeaux, France

Abstract. The ECGI inverse problem is still a common area of research. Since the results in the state of the art are not yet satisfactory, exploring new methods for the resolution of the inverse problem of electrocardiography is the main goal of this paper. To this purpose, we suggest to use temporal and spatial constraints to solve the inverse problem using neural networks methods. First, we use a time-delay neural network initialized with the spatial adjacency operator of the heart surface mesh. Then, we suggest a new approach to reconstruct the heart surface potential from the body surface potential using a spatial adaptation of time delay neural network. It consists on taking into account temporal and spatial dependence between potential measures. This allows to exploit the local and dynamic potential propagation properties. We test these approaches on simulated data. Results show that the new approach outperforms the classic time-delay neural network and has considerable improvements with respect to the state-of-the-art methods.

Keywords: Time-delay neural network · Spatial adaptation · Adjacency matrix · Inverse problem · Electrocardiography.

1 Introduction

The non-invasive electrocardiographic imaging (ECGI) is the procedure carried out nowadays to reconstruct the heart surface potential (HSP) from the body surface potential (BSP) measurements. It provides cardiac information that allows the cardiologists to make better diagnosis of some heart diseases such as atrial and ventricular fibrillations. To date, the state of the art proposes mostly to model the relation between the heart electrical activity and the BSP measurements by a partial differential equation which is solved using a variety of methods based on a transfer matrix. This problem is called inverse problem and known to be ill-posed. In fact, a little perturbation in BSPs can strongly affects the solution. To overcome this problem, several techniques were used especially the Tikhonov regularization. More details about these methods can be found in [3] . In recent years, some papers introduced a new vision of the inverse problem by using machine learning algorithms [2,5,8,9,1]. In this paper, we will introduce a new approach to solve the inverse problem using a neural network inspired from

the work of Jiaqiu Wang et al [7] for travel time prediction. Neural network training and assessment are performed using simulated data.

2 Building the spatial adaptation of time-delay neural network

In this section, we first describe the basic architecture of an artificial neural network (ANN) and a time-delay neural network (TDNN). Then we propose the new approach using the spatial adjacency operator.

2.1 The basic artificial neural network: ANN

A basic ANN has 3 layers : An input layer that reads the input data, a hidden layer, which consists of neurons that learn the relationship between the input and the target and an output layer which produces the output data. A basic neural network can be written as follows :

$$\widehat{O} = \begin{bmatrix} \widehat{O}_1 \\ \vdots \\ \widehat{O}_M \end{bmatrix} = \begin{bmatrix} W_{11} & \cdots & W_{1N} \\ \vdots & \ddots & \vdots \\ W_{M1} & \cdots & W_{MN} \end{bmatrix} \begin{bmatrix} I_1 \\ \vdots \\ I_N \end{bmatrix} + \begin{bmatrix} b_1 \\ \vdots \\ b_M \end{bmatrix} \tag{1}$$

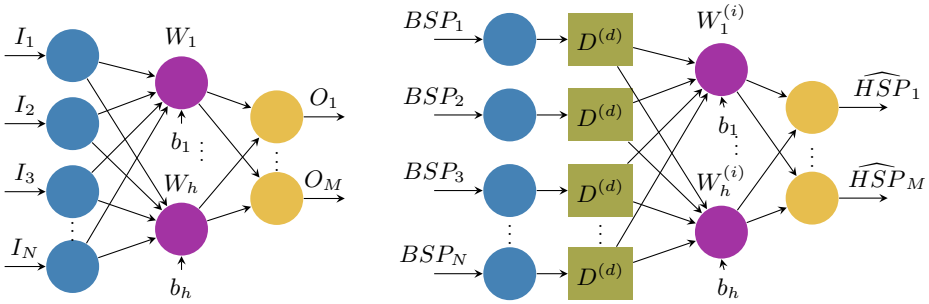
where $\widehat{O}$ is the output of the ANN, I is the input data and W is the weight matrix estimated by the ANN. The vector b is the bias. Figure 1a shows a basic architecture of one hidden layer neural network.
The training process consists of tuning the connection weights W_{ij} between the inputs I_j and the targets O_i using an optimization algorithm to get the optimal prediction $\widehat{O}$.

2.2 Time-delay neural network : TDNN

The main idea is that the body surface potential at a timestep t is highly dependent with its values at previous timesteps $t-1, t-2, \ldots$. Thus, TDNN is a good candidate to get use of this dependence. In fact, each neuron in the TDNN uses the current and its d previous values of the BSP input to estimate the HSP target map at the given timestep t as shown in Figure 1b where $D^{(d)}$ represents the time delay operation. The value of d is the chosen time-delay window size. Hence, for some timestep t the estimated output $\widehat{HSP}(t)$ is given by :

$$\widehat{HSP}(t) = \sum_{i=0}^{d} W^{(i)} BSP(t-i) + b \tag{2}$$

Here, $W^{(i)}$ is the weight matrix associated with the input BSP at timestep $t-i$, $i = 0 \cdots d$.



(a) ANN (b) TDNN

Fig. 1: Architecture diagrams of a ANN and a TDNN models

2.3 Injecting the spatial adjacency operator into the TDNN

Similarly to the temporal correlation, we suppose that the heart surface potential in a given point P is strongly dependent on its recorded values at the adjacent points. Hence, we use the spatial adjacency matrix as a representation of the relation between the target spatial location and its adjacent locations. In our case, this matrix is used in two different ways:
(1) SATDNN-LL LL is for Linear Layer): We add a second linear layer to the TDNN model and we initialize its weight matrix by the spatial adjacency matrix.
(2) SATDNN-AT (AT is for Adjacency Transformation): This model is made with two hidden layers. The first layer is identical to the TDNN expressed by Eq.2. Then, we perform an element-wise multiplication of the first layer output by the first order adjacency matrix $Adj^{(1)}$. This allows, for each point, to only keep the weights corresponding to its adjacent points and reduces the others to zero. The architecture of this model is provided in Figure 2. One can see that non adjacent points are not considered (dashed arrows) in the second linear layer. The principle of this method is close to the space-time delay neural network developed for lipreading [4] and traffic forecasting [7]. But to the best of our knowledge, it has not been introduced before for the inverse problem in electrocardiography. The network of Figure 2 can be written as :

$$\widehat{HSP}(t) = W_2 \left[Adj^{(1)} \left(\sum_{i=0}^{d} W_1^{(i)} BSP(t-i) + b \right) \right] + c \tag{3}$$

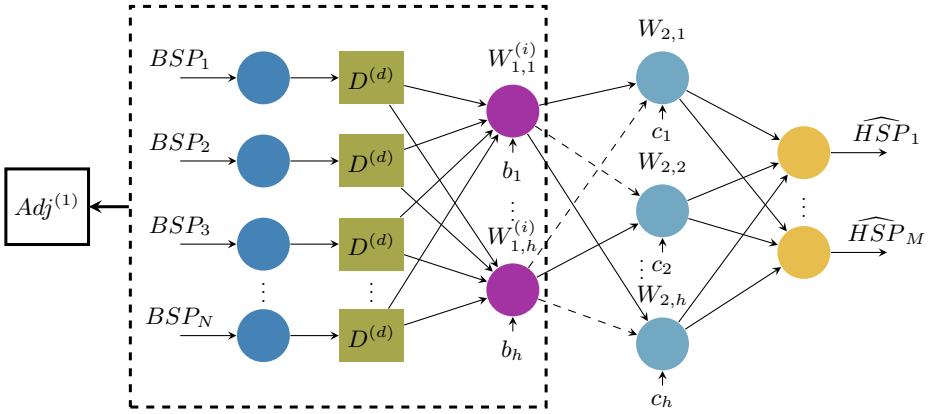


Fig. 2: Architecture diagram of the spatial adaptation of the time-delay neural network (SATDNN-AT).

where b and c are biases. A more detailed matrix form is expressed as :

$$\widehat{HSP}(t) = \begin{bmatrix} \widehat{HSP}_1(t) \\ \widehat{HSP}_2(t) \\ \vdots \\ \vdots \\ \widehat{HSP}_M(t) \end{bmatrix} = \begin{bmatrix} \left[Adj^{(1)} \left(\sum_{i=0}^{d} W^{(i)}_{1,1} BSP(t-i) + b_1\right)\right] W_{2,1} + c_1 \\ \left[Adj^{(1)} \left(\sum_{i=0}^{d} W^{(i)}_{1,2} BSP(t-i) + b_2\right)\right] W_{2,2} + c_2 \\ \vdots \\ \vdots \\ \left[Adj^{(1)} \left(\sum_{i=0}^{d} W^{(i)}_{1,h} BSP(t-i) + b_h\right)\right] W_{2,h} + c_h \end{bmatrix} \tag{4}$$

3 Data

3.1 Simulated data

Simulated data is obtained by considering a realistic 3D heart-torso geometry segmented from CT-Scan images (see [3] for more details). The propagation of the electrical wave was computed using the monodomain reaction-diffusion model. The transmembrane currents used to compute the extracellular potential distribution throughout the torso were computed by solving a static bidomain problem in an homogeneous, isotropic torso model. Synchronized electrical potential on the epicardium and on the body surface were extracted on coarse meshes in order to test the inverse methods. The torso mesh contains 2873 nodes and the heart mesh 519 nodes.

3.2 Model implementation

The different models are implemented using Python and the machine learning library PyTorch [6] reference for Pytorch. The spatial adjacency operator is implemented using VTK library. To train and test the neural network, the dataset is split into 3 subsets : a training dataset for training the model, a validation dataset to avoid overfitting during the training process and a testing one to evaluate the trained model performance. They correspond respectively to 75%, 20% and 5% of the whole dataset. In each epoch of the training process, the trained model is assessed using the validation dataset to avoid overfitting. The training phase stops when the maximum number of epochs is reached. Finally, the trained model is applied to the testing dataset and evaluated using error and correlation metrics.
The three models are implemented using the mean squared error as an optimization criterion and the stochastic gradient descent with momentum as an optimization algorithm. The hyperparameters of this latter are fixed empirically by executing the training process with different values and choosing the best one. Table 1 contains the chosen values of the learning rate and the momentum for training the different models. It is important to mention that we use the same partition of data randomly generated for the assessment and comparison of the three different models.

	TDNN	SATDNN-LL	SATDNN-AT
Learning rate	0.01	0.001	0.01
Momentum	0.8	0.8	0.8

Table 1: Hyperparameters' values used to train the different models : TDNN, SATDNN-LL and SATDNN-AT

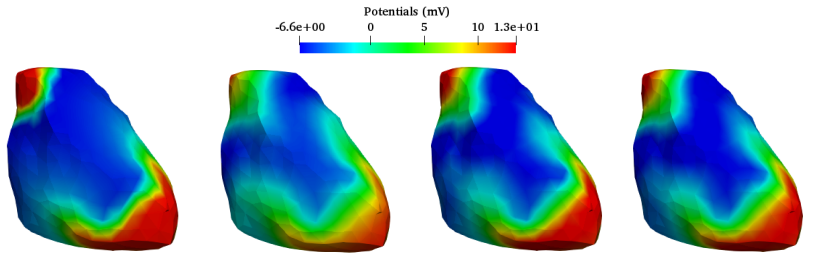
4 Results

In this section, we present the results obtained after training and then testing the three models using the simulated dataset.

4.1 Heart surface potential reconstruction

Results of the training and testing processes in terms of relative error (RE) and correlation coefficient (CC) are reported in Figures 3 and 4. Figure 3 shows HSP maps reconstructed using the three models at the timestep $t = 418$ chosen randomly of the total simulation time. We notice that the SATDNN-AT gives the minimum $RE = 0.23$ and $CC = 0.97$ while the SATDNN-LL comes second with $RE = 0.25$ and $CC = 0.96$ compared to TDNN ($RE = 0.33$, $CC = 94$).

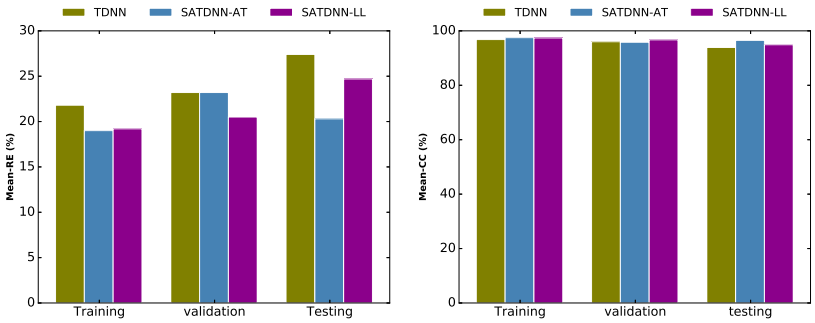
This approves that our approach improves the performance of the classic TDNN. Figure 4a shows that the SATDNN-AT outperforms the two other models in the



(a) t=418 (b) RE=0.33,CC=0.94 (c) RE=0.23,CC=0.97 (d) RE=0.25,CC=0.96

Fig. 3: Simulated (a) and estimated heart surface potential maps with : (b) TDNN, (c) SATDNN-AT, (d) SATDNN-LL at the timestep t=418 of the simulation

testing phase with a mean relative error equal to 20% compared to 24% for the SATDNN-LL which yields the second best result and 27% for the TDNN. Concerning the correlation coefficient, the Figure 4b shows a little distinction between the three models.



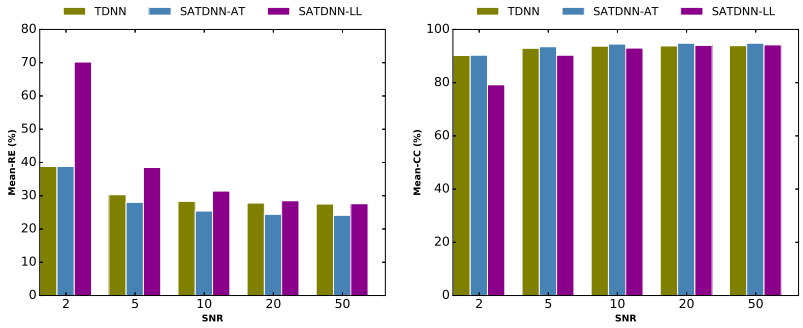
(a) Mean of REs (b) Mean of CCs

Fig. 4: Means of RE and CC of the reconstructed BSPs during the training, validation and testing phases.

4.2 Robustness analysis

The sensitivity issue is of crucial importance in this study due to the ill-posedness of the ECGI inverse problem caused by the measurement errors. To assess the

robustness of the three models, we tested them using noisy dataset with different SNR values going from 2 dB to 50 dB. Figure 5a shows the means of the relative errors obtained by using the SNRs 2dB, 5dB, 10dB, 20dB and 50dB. We observe that the behavior of the TDNN and SATDNN-AT with respect to the SNR is consistent, except for the lowest value 2dB. However, the SATDNN-LL shows a high degradation for low SNRs. For example, it goes from 38% to 70% for 5dB and 2dB respectively.
In terms of correlation coefficient, we observe in figure 5b that all the models presented a similar response to the variation of SNR.



(a) REMean (b) CCMean

Fig. 5: Means of RE and CC of the reconstructed HSPs with respect to the SNR between the original BSPs and the noisy ones.

4.3 Convergence analysis

Figure 6 represents the training and validation loss in terms of mean squared error as a function of the training number of epochs. It shows that the SATDNN-LL initialized with the adjacency matrix converges after 200 iterations compared to 350 iterations for the SATDNN-LL initialized with a random matrix generated by a Gaussian distribution (SATDNN-LL-RI) and 400 or more for the TDNN. First, this indicates that adding a hidden linear layer improves the performance of the neural network in terms of convergence. It also proves that the SATDNN-LL initialized with the adjacency matrix reaches stable training and validation errors faster than the others. The observed spike in the SATDNN-LL curves is due to the use of stochastic gradient descent with momentum as an optimization algorithm.

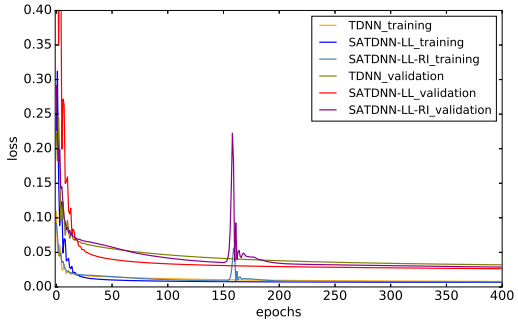


Fig. 6: Evolution of training and validation losses with respect to the number of epochs during the training process using TDNN, SATDNN-LL and SATDNN-LL-RI(random initialization)

5 Discussion and conclusion

In this paper, we presented a proof of concept showing the ability of machine learning techniques to solve the ECGI inverse problem. We presented three different approaches. The first is based on a pure time delayed neural network. The second and the third methods are variants of a spatial adaptation taking into account the influence of adjacent points. Numerical results show that SATDNN-AT is more accurate in terms of RE and that the three methods are almost equivalent in terms of CC. We also tested the robustness of the methods by adding noise on the data. We have seen that the third model SATDNN-LL is more sensitive to the noise and seems to be unstable for low values of SNR. Compared to the state-of-the-art methods, neural network based methods show an improvement in terms of RE and CC evaluated in [3]. This confirms the usefulness of machine learning approach for solving the ECGI inverse problem for few reasons. In one hand, it provides a better generalization of the problem since it doesn't depend on a transfer matrix and its induced errors. In the other hand, machine learning methods are less time-consuming than traditional methods since they are trained once and then can be used multiple times. However, we have to say that introducing this method into real life applications would be challenging. First we will need a training dataset collected on patients containing the BSPs and their correspondent HSPs. We also need to standardize the geometries between patients and map the electrical information on a template geometry. This is one of the challenges of using machine learning in this ECGI application. Furthermore, in order to be efficient the data base should be also sufficiently rich, which means that it should contain data for different heart conditions. The last limitation is related to the high computational cost of the training, because of the number of degrees of freedom in the heart geometry and the number of electrodes collecting the potential data on the body surface. The last one could be easily reduced by selecting a subset of the electrodes. However,

reducing the number of the nodes describing the heart geometry will have an impact on the resolution of the reconstructed electrical information. This would be subject of future works.

Acknowledgements

This work was supported by the French National Research Agency, grant references ANR-10-IAHU04- LIRYC and ANR-11-EQPX-0030.

References

1. Alawad, M., Wang, L.: Learning domain shift in simulated and clinical data: Localizing the origin of ventricular activation from 12-lead electrocardiograms. IEEE transactions on medical imaging (2018)
2. Giffard-Roisin, S., Delingette, H., Jackson, T., Webb, J., Fovargue, L., Lee, J., Rinaldi, C.A., Razavi, R., Ayache, N., Sermesant, M.: Transfer learning from simulations on a reference anatomy for ecgi in personalized cardiac resynchronization therapy. IEEE Transactions on Biomedical Engineering **66**(2), 343–353 (2019)
3. Karoui, A., Bear, L., Migerditichan, P., Zemzemi, N.: Evaluation of fifteen algorithms for the resolution of the electrocardiography imaging inverse problem using ex-vivo and in-silico data. Frontiers in Physiology **9** (2018)
4. Lin, C.T., Nein, H.W., Lin, W.C.: A space-time delay neural network for motion recognition and its application to lipreading. International Journal of Neural Systems **9**(04), 311–334 (1999)
5. Malik, A., Peng, T., Trew, M.L.: A machine learning approach to reconstruction of heart surface potentials from body surface potentials. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 4828–4831. IEEE (2018)
6. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS-W (2017)
7. Wang, J., Tsapakis, I., Zhong, C.: A space–time delay neural network model for travel time prediction. Engineering Applications of Artificial Intelligence **52**, 145–160 (2016)
8. Zemzemi, N., Dubois, R., Coudiere, Y., Bernus, O., Haissaguerre, M.: A machine learning regularization of the inverse problem in electrocardiography imaging. In: Computing in Cardiology 2013. pp. 1135–1138. IEEE (2013)
9. Zemzemi, N., Labarthe, S., Dubois, R.D., Coudière, Y.: From body surface potential to activation maps on the atria: A machine learning technique. In: 2012 Computing in Cardiology. pp. 125–128. IEEE (2012)