From Surface To Depth – thinking the next step of the inverse problem of electrocardiography

Roland Stenger[1,2], **Baltasar Rüchardt**[1,3], **Stefan Luther**[1,3,4], **and Ulrich Parlitz**[1,2,3,*]

[1]Max Planck Institute for Dynamics and Self-Organization, Am Fassberg 17, 37077 Göttingen, Germany
[2]Institute for the Dynamics of Complex Systems, University of Göttingen, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany
[3]German Center for Cardiovascular Research (DZHK), partner site Göttingen, Robert-Koch-Str. 42a, 37075 Göttingen, Germany
[4]Institute of Pharmacology and Toxicology, University Medical Center Göttingen, Robert-Koch-Str. 40, 37075, Göttingen, Germany
[*]ulrich.parlitz@ds.mpg.de

ABSTRACT

Example Abstract. Abstract must not include subheadings or citations. Example Abstract. Abstract must not include subheadings or citations. Example Abstract. Abstract must not include subheadings or citations. Example Abstract. Abstract must not include subheadings or citations. Example Abstract. Abstract must not include subheadings or citations. Example Abstract. Abstract must not include subheadings or citations. Example Abstract. Abstract must not include subheadings or citations. Example Abstract. Abstract must not include subheadings or citations.

1 Introduction

In this article we examine how deep we can ’look’ into the heart based on surface information on 2 different sides on a cube, where the dynamics have been learned by neuronal networks.

2 Methods

In this section we first introduce in section **??** the mathematical model for simulating excitable media. Furthermore, the individual steps which are necessary for generating both regimes, are explained. Following, in section **??** we introduce the machine learning model for facing the task: reconstructing excitations under the surface of the cube.

2.1 Simulation of excitable media

Einleitende Worte...

Barkley model

There are various models to describe the heart dynamic with differential equations in terms of voltage current. Since the experiments goal is to reconstruct hidden regions under the surface qualitatively, the model for excitable media does not have to provide a realistic approximation of the heart’s electrical activity in terms of current voltage. The Barkley model is a model for excitable media and is a system of two coupled differential equations with the variables u and v, which build a reaction diffusion system. It was proposed by Barkley et al. in 1990[?].

The model is given by the equations

$$\begin{aligned}\frac{\partial u}{\partial t} &= D\cdot\nabla^2 u + \frac{1}{\varepsilon}(1-u)\left(u-\frac{v+b}{a}\right),\\ \frac{\partial v}{\partial t} &= u^\alpha - v,\end{aligned} \qquad (1)$$

where u is a fast variable, while the variable v is slower and inhibiting[?]. The parameter a, b and ε are positive constants. A bigger value for a increased the excitation duration while an increasing fraction between b and a, $\frac{b}{a}$, results in a larger excitability threshold[?].

Characteristic variables

To have a more intuitive understanding about temporal and spatial sizes, the following characteristic variables are introduced:

- Time T_c: The characteristic time T_c is defined by average period length, which is estimated by identifying peaks within the time series of individual voxels from simulations of regime A. As described in section **??**, a part of the simulated data consists of 2048 recordings from simulations which last, after an initial phase, over a time span of 512 time steps, which are recorded in intervals of 16 time steps. When two peaks are identified within the time series of a voxel, the gap between them is then saved as a single period length. These lengths, calculated from each voxel on each simulation are used to compute the mean period length, which is considered as characteristic time.
- Length λ_c: An other part of the simulation data, as described in section **??**, consists of 2048 recordings from simulations at a certain time step, which include the data till a depth of 32 discrete spatial units. Therefore, the data can be represented by a tensor with the dimensionality of (32,120,120). From this data, the values of the voxels along vectors in x- and y-direction along the axes with length 120 are used. From each individual simulation, 120*32 vectors of length 120 follow, which consist of the respective u-values of the simulations. In these series, the positions of the peaks at which u reaches a local maximum are determined, and then the distance to the next peak is calculated. A histogram of the collection of all distances from each simulation is shown in figure **??**. The first maximum of this histogram is regarded as characteristic length. The error of this value is estimated with $0.5\Delta s$.

Figure 1. Histogram from the distances between two peaks of the u-value, which are recorded along vectors in x- and y-direction on the simulated cube geometry till a depth of 32.

The resulting values for λ_c and T_c are shown in table **??**. Since in both regimes the discretization of time and space (Δt and Δs) are used, the characteristic variables, scaled with the discretizations are used in the further process. However, due to the chaotic behaviour of regime B, no spiral waves are formed on which such a determination of the characteristic variables would be possible.

Table 1. Characteristic variables for the Barkley regime A.

λ_c	$(16.2 \pm 0.5) \cdot \Delta s$
T_c	$(74.93 \pm 5.45) \cdot \Delta t$

From the two variables λ_c and T_c, a characteristic velocity can be extracted, which results to

$$v_c = \frac{\lambda_c}{T_c} = (0.22 \pm 0.02)\frac{\Delta s}{\Delta t}. \tag{2}$$

A wave-front which propagates along the normal of the surface in a depth of λ_c, needs approximately 74.93 discrete time steps to be recognisable at the surface, since it is estimated to propagate with a speed of v_c.

Starting conditions

In the following, the procedure of generating unique starting conditions in each new simulation is explained, where the characteristic properties for regime A (periodic dynamics) and regime B (chaotic dynamics) remain the same.

2.1.1 Regime A, periodic spiral waves

To initiate spiral waves, the cube in which the simulation takes place, is firstly divided into 8 sub cubes of equal size. The following procedure is the same for two of the 8 sub cubes. In the 6 remaining, the values for u and v are set to 0 at every voxel.

First, the sub cube is *filled* up till a random height (as shown in the left graphic of figure **??**), where the u-value is set to 1.0 at every voxel inside. The same procedure takes place for the v-value, but rotated by 90 degree, whose values are set to 0.5. Visualized is this procedure in the left figures of **??** and **??**. In the next step, this structure is rotated around all three axes randomly, but remains the same for u- and v-value for one sub cube. As third step, the two sub cubes on which this procedure was performed, are placed diagonally to each other, so that these, with the remaining 6 sub cubes, form the shape of the original cube. This is the initial situation with which the simulation starts.The height till which the sub cubes gets *filled* is random, as well as the rotations. The left plot in figure **??** shows an example of a single voxel within the simulation whose u-value is plotted against the time. The periodicity behaviour (after an initial phase) can be seen here.

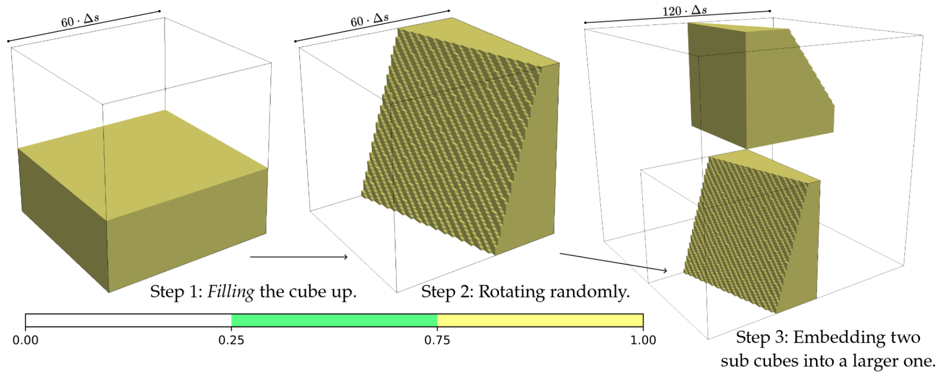

60 · Δs
60 · Δs
120 · Δs
Step 1: *Filling* the cube up.
Step 2: Rotating randomly.
Step 3: Embedding two sub cubes into a larger one.
0.00
0.25
0.75
1.00

Figure 2. Procedure of generating starting conditions for the u- and v-value. Visualized is the u-value at each step within the procedure.

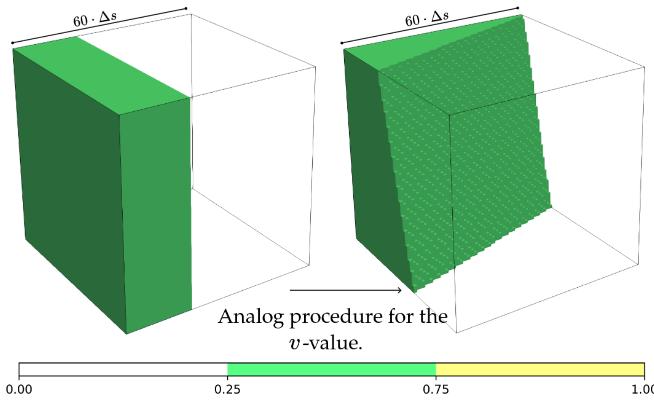


Figure 3. Procedure of generating starting conditions for the u- and v-value. Visualized is the v-value at the first two steps within the procedure.

2.1.2 Regime B, chaos

To generate chaos, not only the Barkley parameter a, b, ε and α are critical, but also the starting conditions. The cube is firstly divided into equilateral cubes of length $6 \cdot \Delta s$ in each dimension, where every voxel in a single sub cube has the same u- and v-value of the Barkley model. The values for u are randomly chosen between 0 and 1 from an equal distribution. If the value exceeds 0.4 in a sub cube, v is set to 0 in the same sub cube. If it is not the case, v is set to 1.0. The right plot in figure **??** shows an example of a single voxel within the simulation whose u-value is plotted against the time. It can be seen that there is no periodicity as in the left graphic.

2.2 Machine learning methods

In this section, we introduce the spatio temporal long-short-term-memory (ST-LSTM)[?], a special kind of recurrent neural network (RNN), which is applied to perform the reconstruction task as defined in section [XXX], a kind of sequence-to-sequence (seq2seq) problem.

2.2.1 Recurrent neural networks

Recurrent neural networks (RNNs) are prominent when it comes to seq2seq problems, where the network is trained to convert a sequence to another sequence. A famous task in this domain is machine translation where a sequence of words in one language is transformed to another sequence of words of a different language. Other important tasks are for example speech-recognition where an encoded voice recording has to be converted into a sequence of words, or next-frame-prediction, a task to predict the future frames of a video. In each example, recurrent networks were considered as state-of-the-art, at least for a while[???].

A recurrent neural network is a form of neural network which is, in combination with an iterative update loop of its internal state (memory), able to process through an input sequence with varying length. With every iteration, an internal state, the *hidden state*, which has a fixed dimensionality, is evolving to be dependent on every previous time step. The *hidden state* can be calculated regardless of the length of the input sequence and provides a representation of the sequence.

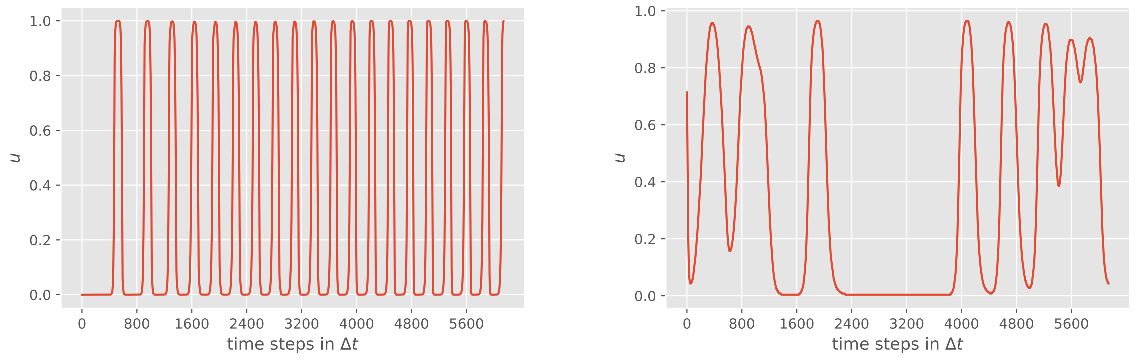


Figure 4. Time development of the u-value of a single voxel from a simulation in the (left) periodic regime A and (right) the chaotic regime B.

A characteristic that is generally seen with seq2seq problems is that both the input sequence and the output sequence are of variable length. For example, a translation model has to be able to take a sequence with varying length of words to output a translation as a sequence with a varying length as well. Furthermore, a strong translation model should be able to translate single words as well as a long sequences of hundreds of words. In this work, the problem of the numerical experiment also faces the characteristic of a seq2seq problem, where a recurrent neural network is used as well.

2.2.2 Long short-term memory (LSTM)

The following equations describe the computations within the probably most famous recurrent neural network, the Long short-term memory (LSTM). Given a sequence of inputs $(x_1, x_2, ..., x_T)$, the LSTM not only evolves a hidden state h_t, but also a cell state C_t such as

$$\begin{aligned}
g_t &= \tanh\left(\mathbf{W}_{cx} \cdot x_t + \mathbf{W}_{ch} \cdot h_t + b_g\right), \\
i_t &= \sigma\left(\mathbf{W}_{ix} \cdot x_t + \mathbf{W}_{ih} \cdot h_t + b_i\right), \\
f_t &= \sigma\left(\mathbf{W}_{fx} \cdot x_t + \mathbf{W}_{fh} \cdot h_t + b_f\right), \\
o_t &= \sigma\left(\mathbf{W}_{ox} \cdot x_t + \mathbf{W}_{oh} \cdot h_t + b_o\right), \\
C_{t+1} &= f_t \circ C_t + i_t \circ g_t, \\
h_{t+1} &= o_t \circ \tanh(C_t),
\end{aligned} \tag{3}$$

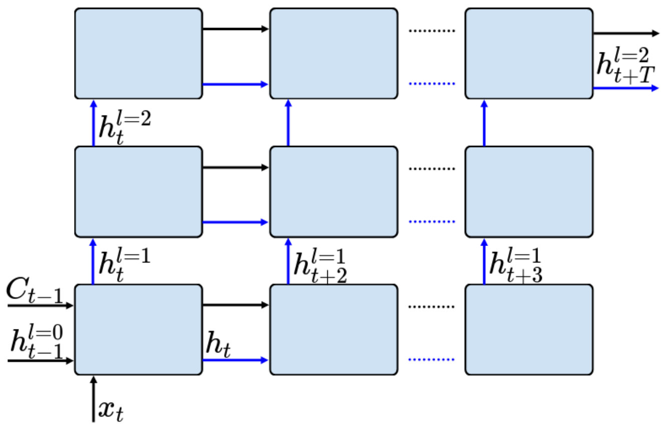
Furthermore, it provides an output sequence $(o_0, o_1, ..., .o_T)$ of the same length as the input sequence. The function tanh is the hyperbolic tangent, and σ is the logistic function. The symbol $\circ$ denotes the Hadamard product, also known as pointwise multiplication.

The neural network to face the seq2seq problem in this experiment is based on the LSTM design with three adjustments, that are made address specific problems in this special task, which are described in the following:

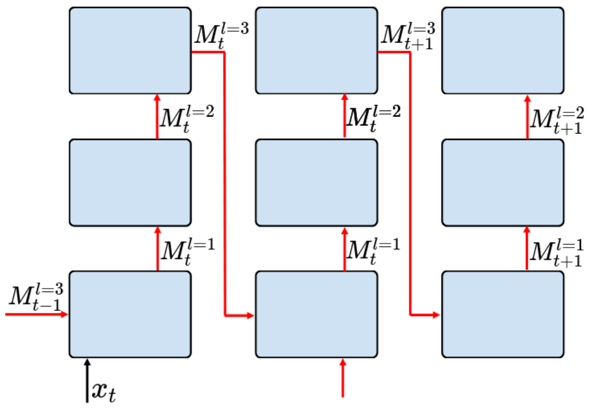
To address the problem mentioned in the last point, Y. Wang et al. introduce in their paper *PredRNN:Recurrent Neural Networks for Predictive Learning using Spatio-temporal LSTMs*[?] a variation of the LSTM and its integration into a stacked recurrent neural network. They do not define how to measure the equality in processing the two aspects (spatial correlations and temporal dynamics) but they formulate it as follows:

> *[...] spatial representations are encoded layer by layer, with hidden states being delivered from bottom to top. However, the memory cells that belong to these [...] layers are mutually independent and updated merely in time domain. Under these circumstances, the bottom layer would totally ignore what had been memorized by the top layer at the previous time step.*

To address this problem they invented a variation of a LSTM, the spatio-temporal LSTM (ST-LSTM), which is defined by:



(a) Long-short-term-memory (LSTM)



(b) Spatio-temporal LSTM (ST-LSTM)

Figure 5. Comparison of the computational flow within a stacked LSTM and a stacked ST-LSTM. The red arrows mark the additional computations of the M-state

$$
\begin{aligned}
g_t &= \tanh\left(\mathbf{W}'_{cx} * x_t + \mathbf{W}'_{ch} * h_{t-1} + b_g\right), \\
i_t &= \sigma\left(\mathbf{W}'_{ix} * x_t + \mathbf{W}'_{ih} * h_{t-1} + b_i\right), \\
f_t &= \sigma\left(\mathbf{W}'_{fx} * x_t + \mathbf{W}'_{fh} * h_{t-1} + b_f\right), \\
C_t &= f_t \circ C_{t-1} + i_t \circ g_t, \\
g'_t &= \tanh\left(\mathbf{W}''_{cx} * x_t + \mathbf{W}'_{cm} * M_t^{l-1} + b'_g\right), \\
i'_t &= \sigma\left(\mathbf{W}''_{ix} * x_t + \mathbf{W}'_{im} * M_t^{l-1} + b'_i\right), \\
f'_t &= \sigma\left(\mathbf{W}''_{fx} * x_t + \mathbf{W}'_{fm} * M_t^{l-1} + b'_f\right), \\
M_t^l &= f'_t \circ M_t^{l-1} + i'_t \circ g'_t \\
o_t &= \sigma\left(\mathbf{W}'_{ox} * x_t + \mathbf{W}'_{oh} * h_{t-1} + \mathbf{W}_{om} * M_t^l + b_o\right), \\
h_t &= o_t \circ \tanh(\mathbf{W}_{1\times 1} * \left[C_t^l, M_t^l\right]).
\end{aligned} \tag{4}
$$

The difference to a normal LSTM is that here the states (C, h and additionally M) are not exclusively evolved in time and then transmitted to the next layer. First, the cell state M evolves through the layers of the stacked ST-LSTM, and is then passed to the input of the next time step at the bottom of the network. The idea of this LSTM adaptation is visualized in at **(b)** of figure **??**. The first figure is a visualization of the computations in a enfolded LSTM network with three stacked layers, while the second one shows an unfolded ST-LSTM. The red line is the computational flow of the cell state M in the computational graph.

Although the special kind of LSTM have been mentioned so far, it is not yet clear how exactly this iterative process can be used to face a (seq2seq) problem such as it is posed in this numerical experiment.

2.2.3 Seq2Seq-models

A common design for a LSTM-neural network to address a seq2seq problem is an encoder-decoder model. It consists of two LSTMs, one for encoding the sequence to a *thought vector* of fixed dimension, and the other for generating an output sequence with varying length from the thought vector. The thought vector consists of the respective states, which are the hidden state h, and cell state C, with an additional state M, in case of a ST-LSTM. In the example of an encoder-decoder network (figure **??**), each time step of the input sequence is processed by an encoder independently, such as the output sequence is processed by the decoder (purple parts in the figure). The input sequence of the decoder LSTM (blue parts at the right side of the graphic) is built by a sequence of the same h-value from the thought vector, for every step. Therefore, every iteration in the LSTM takes into account the same representation of the input sequence while evolving another h- and C-value (and probably additionally M) with every iteration.

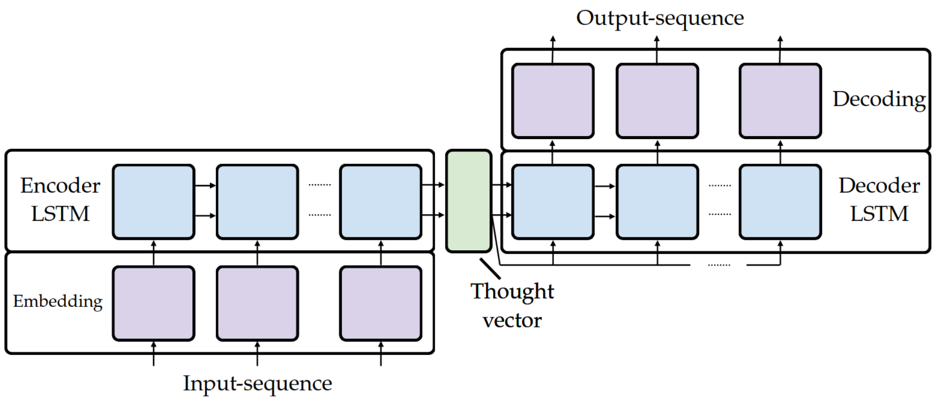


Figure 6. Schematic sequence-to-sequence model (seq2seq) with an encoder-decoder structure.

3 Results

To quantify the reconstruction of the neural networks, we calculate the point wise absolute error between the predicted values of u of the voxels and their true values. The errors are then averaged layer wise, that one gets a mean absolute error per predicted layer. In section **??** we show the results of several trained neural networks of the same type, where the input sequence length is varied. It is tested with the two regimes (A and B), where A has periodic dynamic and B show chaotic behaviour. The best model, considering our error measurement from this experiment, is then taken for further examination in the section afterwords, where the reconstruction target changes. Furthermore, its results are compared to another neural network, which is based on convolutions and a conditional random field.

3.1

Figures:

- Full 3d cube with prediction
- Gridplot, different T
- MAE comparison with different Depths
- Comparison with Sebastian's model

Discussion

The Discussion should be succinct and must not contain subheadings.
LaTeX formats citations and references automatically using the bibliography records in your .bib file, which you can edit via the project menu. Use the cite command for an inline citation, e.g.[?].

For data citations of datasets uploaded to e.g. *figshare*, please use the `howpublished` option in the bib entry to specify the platform and the link, as in the `Hao:gidmaps:2014` example in the sample bibliography file.

Acknowledgements (not Xsub = Xcompulsory)

Acknowledgements should be brief, and should not include thanks to anonymous referees and editors, or effusive comments. Grant or contribution numbers may be acknowledged.

Author contributions statement

Must include all authors, identified by initials, for example: A.A. conceived the experiment(s), A.A. and B.A. conducted the experiment(s), C.A. and D.A. analysed the results. All authors reviewed the manuscript.

Additional information

To include, in this order: **Accession codes** (where applicable); **Competing interests** (mandatory statement).

The corresponding author is responsible for submitting a competing interests statement on behalf of all authors of the paper. This statement must be included in the submitted article file.