Transformation of ReLU-based recurrent neural networks from discrete-time to continuous-time

Zahra Monfared [*1] **Daniel Durstewitz** [*1 2]

Abstract

Recurrent neural networks (RNN) as used in machine learning are commonly formulated in discrete time, i.e. as recursive maps. This brings a lot of advantages for training models on data, e.g. for the purpose of time series prediction or dynamical systems identification, as powerful and efficient inference algorithms exist for discrete time systems and numerical integration of differential equations is not necessary. On the other hand, mathematical analysis of dynamical systems inferred from data is often more convenient and enables additional insights if these are formulated in continuous time, i.e. as systems of ordinary or partial differential equations (ODE/ PDE). Here we show how to perform such a translation from discrete to continuous time for a particular class of ReLU-based RNN. We prove three theorems on the mathematical equivalence between the discrete and continuous time formulations under a variety of conditions, and illustrate how to use our mathematical results on different machine learning and nonlinear dynamical systems examples.

1. Introduction

Recurrent neural networks (RNN) are popular devices in machine learning and AI for tasks that require processing and prediction of temporal sequences, like machine translation (Sutskever et al., 2014), natural language processing (Kumar et al., 2016; Zaheer et al., 2017), or tracking of moving objects in videos (Milan et al., 2017). More recently, in the natural sciences, biology and physics in particular, RNN were also introduced as powerful tools for approximating the unknown nonlinear dynamical system (DS) that produced a set of empirically observed time series, i.e. for identifying the data-generating nonlinear DS in a completely data-driven, bottom-up way (Durstewitz, 2017; Koppe et al., 2019; Razaghi & Paninski, 2019; Vlachas et al., 2018; Zhao & Park, 2017). Theoretically, it has been proven that (continuous) RNN can approximate the flow field of any other nonlinear DS to arbitrary precision on compact sets of the real space under some mild conditions (Funahashi & Nakamura, 1993; Hanson & Raginsky, 2020; Kimura & Nakano, 1998; Trischler & D'Eleuterio, 2016).

RNN, in the form most widely used in machine learning, constitute discrete-time DS defined by a recursive transition rule (difference equation) which maps the network's activation states among consecutive time steps, $z_t = F_\theta(z_{t-1}, s_t)$, where θ are parameters of the system and $\{s_t\}$ is a sequence of external inputs. This formulation is highly advantageous for training RNN on observed data sequences since efficient variational inference and Expectation-Maximization algorithms, which do not require numerical integration of nonlinear ODE, exist for discrete-time systems (Durstewitz, 2017; Koppe et al., 2019; Razaghi & Paninski, 2019; Zhao & Park, 2017). In many scientific contexts, like neuroscience (Koch & Segev, 2003), ODE systems are often highly nonlinear and stiff and thus require more involved implicit numerical integration schemes to achieve accurate and stable solutions (Koch & Segev, 2003; Ozaki, 2012; Press et al., 2007). Furthermore, empirical data are always sampled at *discrete* time points, such that in most cases this assumption may not be too limiting for the purpose of model inference.

On the other hand, natural systems evolve in continuous time, and hence most mathematical theories in physics and biology are formulated in *continuous* time. Thus, for the purpose of theoretical analysis of models inferred from experimental data a continuous-time formulation of the system with biologically or physically meaningful time constants would be preferable. Moreover, a continuous time ODE system enables to analyze properties of the DS under study that are much more difficult or impossible to assess in discrete time. For instance, a continuous time

[*]Equal contribution [1]Department of Theoretical Neuroscience, Central Institute of Mental Health, Medical Faculty Mannheim, Heidelberg University, Mannheim, Germany [2]Faculty of Physics and Astronomy, Heidelberg University, Heidelberg, Germany. Correspondence to: Zahra Monfared <zahra.monfared@zi-mannheim.de>, Daniel Durstewitz <daniel.durstewitz@zi-mannheim.de>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

DS comes with a flow field that enables to visualize the system's dynamics more easily (e.g. Fig. 3), and it enables to smoothly interpolate between observed data points and thus to assess the solution at arbitrary time points. This is of advantage in particular when observations come at irregular event times (Chen et al., 2018). Also, separating DS into slow and fast subsystems (separation of time scales) is a powerful analysis tool (Durstewitz & Gabriel, 2007; Rossetto et al., 1998) that is not readily available for discrete-time systems. More generally, the fact that an ODE system is usually smooth almost everywhere eases the mathematical study of many phenomena, like those of periodic and non-periodic solutions, stable and unstable manifolds of fixed points, bifurcations, or stability of solutions more generally (Absil, 2006). In fact, finding explicit solutions is often impossible even for 1-dimensional recursive maps, such that one often has to revert to graphical methods like cobwebs (Absil, 2006; Haschke, 2004).

It would therefore be highly desirable to find ways in which discrete time RNN models could be embedded into continuous time systems without changing their phase space. Such an embedding requires that for a given discrete-time system $\{\phi_t\}_{t\in\mathbb{T}_d}$ there exists a continuous-time system $\{\psi_t\}_{t\in\mathbb{T}_c}$ such that $\phi_t = \psi_t$ for $t \in \mathbb{T}_d$ (Haschke, 2004). In general, finding such an embedding is not possible for nonlinear discrete-time systems, while the reverse problem, although not trivial (Ozaki, 2012), is much easier and there are different ways for obtaining a discrete time system from a continuous one[1]. The present paper will address this issue using a specific formulation of RNN, namely piecewise-linear RNN (PLRNN) that employ rectified-linear units (ReLU) as their activation function. PLRNN are universal in terms of their dynamical repertoire (Koiran et al., 1994; Siegelmann & Sontag, 1995; Lu et al., 2017), can extract long-term dependencies in sequential data just like LSTMs can (Schmidt et al., 2020), and – in particular – have been used previously to infer nonlinear DS from time series data (Durstewitz, 2017; Koppe et al., 2019). We show that for this specific class of RNN models mathematically equivalent ODE systems, in the sense defined above, can be derived under almost all conditions. We will exemplify these results on a couple of machine learning and DS models, including an ODE solution to the well-known 'addition problem' (Hochreiter & Schmidhuber, 1997), limit cycle and chaotic dynamics, and on a PLRNN inferred from empirical time series (human functional magnetic resonance imaging [fMRI] data; (Koppe et al., 2019)).

[1]The most important of such methods is the Poincaré map, where a continuous time will be reduced to a discrete time system by successive intersections of the flow of the continuous system with the Poincaré section, such that most dynamical properties of the original system will be preserved.

2. Related work

In some recent papers (de Brouwer et al., 2019; Jordan et al., 2019) continuous-time ODE 'approximations' of discrete RNN were sought based on 'inverting' the forward Euler rule for numerically solving continuous ODE systems. A related idea is that of 'Neural ODE' (Chen et al., 2018), where the flow is given by a (deep) neural network (cf. (Pearlmutter, 1989)) to yield a system continuous across 'space' (layers) or time (see also (Abarbanel et al., 2018) for closely related ideas). In (Chang et al., 2019) the reverse approach is taken for obtaining a discrete time formulation that preserves certain properties of the ODE system. None of this work, however, explicitly considered the problem of finding an equivalent continuous-time description of a discrete-time RNN. In fact, apart from the fact that a simple forward Euler rule is known to be rather inaccurate and unstable for integrating stiff ODE systems (Press et al., 2007), a naïve 'inversion' will generally *not* result in a mathematically equivalent ODE system in the sense defined further above, i.e. the resulting system usually will not have the same phase space and temporal behavior. Consequently, much of the previous work was less aimed at finding a mapping between discrete and continuous time networks, but rather to formulate the problem of neural network training in continuous time and/or space to begin with to exploit advantages of an ODE formulation in one way or the other.

Rather, a 'true' translation of a discrete into a continuous RNN may be achieved by taking the continuous time limit of $x_t = F(x_{t-1})$, $\lim_{dt\to 0}\left[\frac{x_t - x_{t-1}}{dt} = \frac{F(x_{t-1}) - x_{t-1}}{dt}\right]$, which, however, can be highly nontrivial or impossible for nonlinear systems (Ozaki, 2012). (Ozaki, 2012), while focusing on the continuous-to-discrete case, also briefly discusses some ideas on this reverse direction of a discrete-to-continuous mapping for nonlinear DS. Only approximations are considered, however, that may work well only for certain cases (RNN, in particular, were not discussed), while here we seek exact equivalence according to the definition further above. In (Nykamp, 2019) a discrete logistic equation is transformed to a continuous-time system by taking the temporal limit. It is, however, not possible to apply such a method for all discrete DS, and even if a transformation is possible, the discrete DS may have different dimensions than the continuous time equivalent; for instance, chaotic behavior is possible in a 1-dimensional recursive map but requires 3 dimensions in an ODE system (Strogatz, 2015). Specifically for *linear* systems, the necessary and sufficient conditions for embedding a discrete-time homogeneous linear system in a continuous-time system have been studied in (Reitmann, 1996). Here, we are interested more generally in discrete-time *non-homogeneous* systems which

are *piecewise linear*, i.e. piecewise-linear (ReLU-based) recurrent neural networks (PLRNN). We will show how to embed a PLRNN into an equivalent continuous-time ODE system, and how the dynamics of the PLRNN is directly connected to the dynamics of the corresponding ODE system. To the best of our knowledge this is the first time a method is introduced for converting discrete into continuous time RNN in a mathematically exact way, i.e., such that the systems are mathematically and dynamically equivalent without using any approximations or numerical techniques.

3. Preliminaries

In the following we will collect some results which we will need for our derivations.

Theorem 1. *Consider the non-homogeneous system*

$$\dot{x} \;=\; Ax + b. \tag{1}$$

Then $\phi(t) = e^{At}$ (with $\phi(0) = I$) is the fundamental matrix solution for linear system $\dot{x} = Ax$, and the solution of system (1) has the form

$$x(t) \;=\; e^{At}\, x_0 + e^{At} \int_0^t e^{-A\tau}\, b\, d\tau, \qquad x(0) = x_0. \tag{2}$$

Proof. See (Perko, 1991). □

Proposition 1. *The matrix $B = \int_0^T e^{At}\, dt$ is invertible iff for every eigenvalue λ of matrix A, we have: $\lambda T \notin 2i\pi\mathbb{Z}^*$ ($\mathbb{Z}^* = \mathbb{Z} \setminus \{0\}$).*

Proof. The Taylor expansion of matrix B has the form:

$$B \;=\; IT + A\frac{T^2}{2!} + A^2\frac{T^3}{3!} + A^3\frac{T^4}{4!} + \cdots, \tag{3}$$

and

$$\text{Spectrum}(B) \;=\; \big\{s(\lambda)\,|\,\lambda \in \text{Spectrum}(A)\big\}, \tag{4}$$

such that

$$s(\lambda) \;=\; \begin{cases} \frac{e^{\lambda T}-1}{\lambda}; & \lambda \neq 0 \\ T; & \lambda = 0 \end{cases}. \tag{5}$$

B is invertible iff it does not have any zero eigenvalue. So, by (4)-(5), B is invertible iff $\lambda T \notin 2i\pi\mathbb{Z}^*$. □

Logarithm of real matrices. For a complex matrix a logarithm (not necessarily unique) will exist iff it is invertible (Higham, 2008). Real matrices do not always have a real logarithm. However, the following theorem guarantees the existence of a real logarithm for a real matrix.

Theorem 2. *A real matrix $A \in \mathbb{R}^{n\times n}$ has a real logarithm if and only if*

(I) A is invertible, and

(II) every $k \times k$ Jordan block associated with a negative eigenvalue occurs an even number of times in the Jordan form of A.

Proof. See (Nunemacher, 1989; Sherif & Morsy, 2008). □

Corollary 2.1. *Due to theorem 2 a real nonsingular 2×2 matrix A with two eigenvalues λ_1 and λ_2 will have a real logarithm in the following cases:*

(1) λ_1 and λ_2 are complex conjugate, i.e. $\lambda_{1,2} = a \pm ib, b \neq 0$. In this case A and $log(A)$ have the Jordan forms $\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$ and $\begin{pmatrix} \alpha & \beta \\ -\beta & \alpha \end{pmatrix}$, respectively, where $e^{\alpha \pm i\beta} = a \pm ib$.

(2) λ_1 and λ_2 are both real and positive.

(3) $\lambda_1 = \lambda_2 = -\lambda$ with $\lambda > 0$. In this case A and $log(A)$ have the Jordan forms $\begin{pmatrix} -\lambda & 0 \\ 0 & -\lambda \end{pmatrix}$ and $\begin{pmatrix} log(\lambda) & \pi \\ -\pi & log(\lambda) \end{pmatrix}$, respectively.

For more details see (Nunemacher, 1989; Sherif & Morsy, 2008).

Remark 2.1. *Suppose that A is a real nonsingular 2×2 matrix which has two equal and negative eigenvalues $\lambda_1 = \lambda_2 < 0$. If A has the Jordan form $\begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_2 \end{pmatrix}$, then it will not have a real logarithm.*

4. Conversion of discrete- into continuous-time PLRNN

4.1. Discrete-time RNN model

Consider a piecewise-linear RNN (PLRNN) of the generic form

$$Z_{t+1} \;=\; A\, Z_t + W\phi(Z_t) + h \tag{6}$$

where $\phi(Z_t) = \max(Z_t, 0)$ is the element-wise rectified linear unit (ReLU) transfer function, $Z_t = (z_{1t}, \cdots, z_{Mt})^T \in \mathbb{R}^M$ denotes the neural state vector at time $t = 1 \cdots T$, the diagonal entries of $A = diag(a_{11}, \cdots, a_{MM}) \in \mathbb{R}^{M\times M}$ represent (linear) auto-regression weights, $W \in \mathbb{R}^{M\times M}$ is a matrix of connection weights (sometimes assumed to be off-diagonal,

i.e. with diagonal elements equal to zero, e.g. (Koppe et al., 2019)), and h is a bias term (Durstewitz, 2017; Koppe et al., 2019).

The discrete-time PLRNN (6) can be represented in the form

$$Z_{t+1} = (A + W D_{\Omega(t)}) Z_t + h, \tag{7}$$

where

$$D_{\Omega(t)} := \text{diag}(d_{\Omega(t)}),$$

with

$$d_{\Omega(t)} := \big(d_1(t), d_2(t), \cdots, d_M(t)\big),$$

such that $d_i(t) = 0$ if $z_{it} \leq 0$ and $d_i(t) = 1$ if $z_{it} > 0$, for $i = 1, 2, \cdots, M$. There are 2^M different configurations for matrix $D_{\Omega(t)}$, depending on the sign of the components of Z_t. That is, the phase space of system (7) is separated into 2^M sub-regions by $M 2^{M-1}$ hyper-surfaces which form discontinuity boundaries. Now, indexing the 2^M different configurations of $D_{\Omega(t)}$ as D_{Ω^k} for $k \in \{1, 2, \cdots, 2^M\}$, we define 2^M matrices

$$W_{\Omega^k} := A + W D_{\Omega^k}, \tag{8}$$

such that in each sub-region the dynamics are governed by a different linear map (cf. Fig. S1), i.e.

$$Z_{t+1} = W_{\Omega^k} Z_t + h, \qquad k \in \{1, 2, \cdots, 2^M\}. \tag{9}$$

All sub-regions S_{Ω^i} corresponding to (9) together with all switching boundaries $\Sigma_{ij} = \bar{S}_{\Omega^i} \cap \bar{S}_{\Omega^j}$ between every pair of successive sub-regions S_{Ω^i} and S_{Ω^j}, with $i, j \in \{1, 2, \cdots, 2^M\}$, are formally defined in Suppl. sect. 7. Note that map (7) is continuous, but has many discontinuities in the Jacobian across the switching boundaries Σ_{ij} (for more details please see Suppl. sect. 7). It is easy to see that for every pair of matrices D_{Ω^k} which differ only in one diagonal entry, their corresponding matrices W_{Ω^k} will differ in only one column.

4.2. Transformation from discrete to continuous-time

As noted in the Introduction, measurements of physical or biological systems are always carried out at discrete times separated by finite time steps Δt (often with a constant sampling rate), and efficient algorithms are available for inferring discrete RNN models from such data (Durstewitz, 2017; Koppe et al., 2019; Razaghi & Paninski, 2019; Zhao & Park, 2017). However, the state of natural dynamical systems is usually better described in continuous time, and - furthermore - continuous-time formulations enjoy a number of key advantages (Chen et al., 2018; Haschke, 2004): First, since for ODE systems trajectories are continuous curves rather than collections of single points, some dynamical properties can be determined more easily. For instance, in discrete-time systems it is not possible to distinguish quasiperiodic from periodic orbits with a large period. Likewise, for ODE systems we have a continuous phase portrait (almost) everywhere, and solutions are defined for any arbitrary time point. Finally, some types of analysis are much easier to do in continuous rather than discrete time. For example, performing a change of variables to compute probability distributions within normalizing flows can be more convenient in continuous- rather than discrete-time systems (Chen et al., 2018). [2]

In the following we will state a set of theorems which show how to convert a discrete into a continuous PLRNN by transforming discrete-time system (9) on every sub-region into an equivalent continuous-time system. In this way, a system of piecewise ordinary differential equations will be assigned to the discrete-time system (9) on $\mathbb{R}^M$. Here we will just state our major results, while all details of the proofs will be given in Suppl. sect. 7. Note that the following theorems settle the problem for one time step Δt taken by the PLRNN, from which, however, results for more time steps immediately follow.

Theorem 3. *Consider discrete-time system (9) on S_{Ω^k}, $k \in \{1, 2, \cdots, 2^M\}$, i.e. the system*

$$Z_{t+1} = F(Z_t) = W_{\Omega^k} Z_t + h, \tag{10}$$

with

$$W_{\Omega^k} := A + W D_{\Omega^k}, \; Z_t \in S_{\Omega^k}, \tag{11}$$

and time step Δt. Suppose that W_{Ω^k} is invertible and has no eigenvalue equal to one, i.e. $P_{W_{\Omega^k}}(1) \neq 0$, where $P_{W_{\Omega^k}}$ denotes the characteristic polynomials of W_{Ω^k}.

(1) There exists a continuous-time system

$$\dot{\zeta} = G(\zeta) = \tilde{W}_{\Omega^k} \zeta(t) + \tilde{h}, \tag{12}$$

which is equivalent to (10) on $[t_0, \; t_0 + \Delta t]$ in the sense that

$$Z_{t_0} = \zeta(t_0), \quad Z_{t_0 + \Delta t} = W_{\Omega^k} Z_{t_0} + h = \zeta(t_0 + \Delta t). \tag{13}$$

Moreover, in this case $\tilde{W}_{\Omega^k}$ is also an invertible matrix and has no eigenvalue equal to one, i.e. $P_{\tilde{W}_{\Omega^k}}(1) \neq 0$.

[2] According to (Chen et al., 2018), while for discrete systems defined by a bijective map, say F, the change in densities due to the mapping by F is given by the determinant of the Jacobian of F, for continuous systems the derivative of the logarithm of the probability density with respect to time is given by the *trace* of the Jacobian, which is easier and numerically more robustly to compute.

Also,

$$\begin{cases} \tilde{W}_{\Omega^k} = \frac{1}{\Delta t} log(W_{\Omega^k}) \\ \tilde{h} = -\frac{1}{\Delta t} log(W_{\Omega^k}) \left[I - W_{\Omega^k}\right]^{-1} h \end{cases} . \quad (14)$$

Furthermore, if for W_{Ω^k} each of its Jordan blocks associated with a negative eigenvalue occurs an even number of times, then $\tilde{W}_{\Omega^k}$ will be a real matrix.

(2) If W_{Ω^k} is both invertible and diagonalizable, then $\tilde{W}_{\Omega^k}$ will be invertible and diagonalizable too.

Proof. See Suppl. sect. 7 (subsection 7.3). □

Corollary 3.1. *The results of theorem 3 are also true if W_{Ω^k} is a positive-definite matrix with $P_{W_{\Omega^k}}(1) \neq 0$.*

Proof. Let W_{Ω^k} be a positive-definite matrix. Then its determinant is positive, which implies that it is invertible, thus satisfying the conditions of theorem 3. Note that if W_{Ω^k} is also Hermitian (or symmetric for real matrices), all eigenvalues of W_{Ω^k} are real and it is diagonalizable as well (Bhatia, 2007). □

In theorem 3 it is assumed that W_{Ω^k} has no eigenvalue equal to one. But there are some PLRNN with interesting computational properties in the form of system (10), for which W_{Ω^k} has at least one eigenvalue equal to one (see Example 2). Hence, more generally we are also interested in converting such neural networks from discrete- to continuous-time. The next two theorems are stated to address this problem.

Theorem 4. *Consider system (10) and assume that W_{Ω^k} is invertible, diagonalizable, and has at least one eigenvalue equal to 1, i.e. $P_{W_{\Omega^k}}(1) = det(I - W_{\Omega^k}) = 0$. Then, there exists an equivalent, in the sense defined in equation (13), continuous-time system (12) for (10) on $[t_0,\ t_0 + \Delta t]$ such that $\tilde{W}_{\Omega^k}$ is diagonalizable, but not invertible, and*

$$\begin{cases} \tilde{W}_{\Omega^k} = \frac{1}{\Delta t} log(W_{\Omega^k}) \\ \tilde{h} = -\frac{1}{\Delta t} \left[\begin{pmatrix} I_n & 0 \\ 0 & O \end{pmatrix} + log(W_{\Omega^k}) \right] \\ \qquad \times \left[\begin{pmatrix} O_{n\times n} & 0 \\ 0 & I \end{pmatrix} - W_{\Omega^k} \right]^{-1} h \end{cases} , \quad (15)$$

where n represents the number of eigenvalues of W_{Ω^k} which are equal to 1 (or the number of eigenvalues of $\tilde{W}_{\Omega^k}$ equal to zero). Also, if each Jordan block of W_{Ω^k} associated with a negative eigenvalue occurs an even number of times, then $\tilde{W}_{\Omega^k}$ will be real.

Proof. See section 7 (subsection 7.4). □

Remark 4.1. *For $n = 0$ in (15), we have*

$$\begin{pmatrix} I_n & 0 \\ 0 & 0 \end{pmatrix} = 0, \qquad \begin{pmatrix} 0_{n\times n} & 0 \\ 0 & I \end{pmatrix} = I. \quad (16)$$

Thus for $n = 0$, i.e. when W_{Ω^k} has no eigenvalue equal to 1, relations (14) and (15) become identical.

W_{Ω^k} in theorem 4 must be diagonalizable, but for some computationally interesting PLRNN W_{Ω^k} is not diagonalizable. The following theorem is stated and proved to address this issue.

Theorem 5. *Let W_{Ω^k} in system (10) be invertible and $P_{W_{\Omega^k}}(1) = det(I - W_{\Omega^k}) = 0$. Then, there exists an equivalent, in the sense of equation (13), continuous-time system (12) for (10) on $[t_0,\ t_0 + \Delta t]$ such that $\tilde{W}_{\Omega^k}$ is not invertible, and*

$$\begin{cases} \tilde{W}_{\Omega^k} = \frac{1}{\Delta t} log(W_{\Omega^k}) \\ \tilde{h} = \left(W_{\Omega^k} \int_0^{\Delta t} e^{-\frac{\tau}{\Delta t} log(W_{\Omega^k})}\ d\tau \right)^{-1} h \end{cases} . \quad (17)$$

Further suppose that each Jordan block of W_{Ω^k} associated with a negative eigenvalue occurs an even number of times, then $\tilde{W}_{\Omega^k}$ is a real matrix.

Proof. See Suppl. section 7 (subsection 7.5). □

5. Application examples

In the following we will illustrate how to use our mathematical results for four specific PLRNN systems of relevance in dynamical systems theory and machine learning. These include examples for a nonlinear oscillator (limit cycle), the 'addition problem' introduced by (Hochreiter & Schmidhuber, 1997) to probe long short-term-memory capacities of RNN, an example of a chaotic system (Lorenz attractor), and a PLRNN inferred from empirical (human fMRI) time series. Matlab code for all these examples is available at github.com/DurstewitzLab/contPLRNN.

Example 1. *Consider a discrete-time PLRNN emulation of the nonlinear van-der-Pol oscillator, derived by training a discrete PLRNN with $M = 10$ units on time series generated by the van-der-Pol equations (taken from (Koppe et al., 2019), provided online at github.com/DurstewitzLab). The Jacobian matrix of this system is always invertible and has no eigenvalue equal to one in any of the sub-regions S_{Ω^i}. Hence we can use theorem 3 to convert this system from discrete- to continuous-time. Fig. 1A illustrates time graphs overlaid for the discrete (blue circles) and continuous (red curves) PLRNN, while Fig. 1B depicts a 2d section of the system's continuous phase space with corresponding flow field. Note that there is a perfect agreement between the discrete and continuous solutions for the set of times at which*

discrete-PLRNN outputs are defined, while at the same time the continuous PLRNN smoothly interpolates between the discrete-time values. Also note that this agreement continues across different subregions S_{Ω^k} induced by the ReLU function.

As an example of a specific DS analysis that is much easier in the continuous than in the discrete time system we consider a special type of bifurcation (i.e., a point in the system's parameter space where the dynamic abruptly changes), the so-called grazing bifurcation of periodic orbits. It occurs in piecewise smooth continuous-time systems when a periodic orbit tangentially intersects ('grazes') with a switching boundary. A related bifurcation, the so-called border-collision bifurcation, also occurs in discrete-time systems when a k-cycle collides with one border. However, in the discrete case, finding the specific bifurcation point can be very challenging, especially in high dimensions, since it amounts to solving highly nested nonlinear equations of the general form $F^k(Z,b)-Z=0$ for large k (where F^k is the k-times iterated map and b a bifurcation parameter), and determining among the solutions that particular point that agrees with the conditions of the bifurcation. In the continuous case, in contrast, one can relatively straightforwardly solve the implied system of equations (see Suppl. section 7 (subsection 7.6) for more details), and hence converting the discrete into the continuous PLRNN offers a big advantage. An example for a grazing bifurcation in the continuous-time PLRNN emulation of the van-der-Pol system is shown in Fig. 2. Bifurcation phenomena like these are of great practical importance and can also have fundamental implications for training RNN (Doya, 1992), since they may imply a sudden switch in the temporal structure of the system's behavior as the bifurcation point is crossed.

Example 2. *Here we consider a 2-unit RNN solution (adapted from (Schmidt et al., 2020)) to the 'addition problem' introduced in (Hochreiter & Schmidhuber, 1997). The RNN receives two streams of inputs, one stream of uniform random numbers $s_{1t} \in [0,1]$, and one series of indicator bits $s_{2t} \in \{0,1\}$ which are mostly 0 except for two 6-step time intervals $[t_1, t_1+5]$ and $[t_2, t_2+5]$ where $s_{2,t_1:t_1+5} = s_{2,t_2:t_2+5} = 1$. Inputs were accommodated by adding a term Cs_t to eq. (6), with $C = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$. The network's task is to produce as an output the sum of all the inputs in s_1 that correspond to the two time intervals $[t_1, t_1+5]$ and $[t_2, t_2+5]$. A simple discrete-time 2-unit PLRNN which (approximately) solves this task is the one with parameters*

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0.01 \end{pmatrix}, \quad W = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad h = \begin{pmatrix} 0 \\ -0.995 \end{pmatrix}.$$

Applying definition (8), $W_{\Omega^k} := A + WD_{\Omega^k}$, we have

$$W_{\Omega^1} = W_{\Omega^2} = \begin{pmatrix} 1 & 1 \\ 0 & 0.01 \end{pmatrix},$$

$$W_{\Omega^3} = W_{\Omega^4} = \begin{pmatrix} 1 & 0 \\ 0 & 0.01 \end{pmatrix}.$$

Hence, every W_{Ω^k} is invertible and diagonalizable, but has one eigenvalue equal to one, satisfying the conditions of theorem 4. Translating this system into continuous time using theorem 4, Fig. 3A displays the two system variables in both continuous and discrete time. Inputs were treated as constant across a time step Δt of the discrete-time PLRNN, such that for the continuous-time PLRNN the term $c = Cs_t$ could be transformed in the same way as the bias term h in eq. (15). Note that while z_2 directly responds to the random inputs, its activity will be integrated (summed) by z_1 whenever the second (indicator) inputs are on, i.e. $s_2 = 1$, which is the case here for the two temporal intervals $[100, 105]$ and $[400, 405]$, as can be seen by z_2 crossing the black dashed line (i.e., when $z_2 > 0$). Visualizing the continuous system's phase space gives some insight into how the RNN solved the addition problem (Fig. 3B): The ζ_1-line forms a line attractor for $\zeta_2 = -0.995/(1-0.01)$, with the flow converging toward this line from all directions, and zero flow right on this line, thus yielding a dimension in state space along which arbitrary values could be stored (Durstewitz, 2003; Seung, 1996). A series of sufficiently strong inputs to the RNN (i.e., whenever $s_2 = 1$) will push the system away from its current to a new position on the line attractor where it will remain until the second suprathreshold series of inputs arrives, in this manner integrating the s_1 inputs accompanied by 1's in s_2 (see also (Schmidt et al., 2020)). In this specific example, the first series of s_1 inputs sums up to ≈ 2.7 (as marked by the left green circle on the ζ_1-line at $\zeta_2 \approx -1$) and the second to ≈ 3.4, and the PLRNN correctly reports the total sum of inputs (right green circle) in its final position on the line attractor.

Example 3. *As an example for a system with chaotic dynamics we chose a PLRNN emulation ($M = 10$) of the 3d Lorenz system, i.e. a PLRNN trained to reproduce the dynamics of the Lorenz equations within the chaotic regime (taken from (Koppe et al., 2019)). In this case, we could apply theorem 3 to accomplish the transformation to continuous time, as all matrices W_{Ω^k} as defined in (8) were invertible with no eigenvalue equal to 1. Fig. 4 confirms that the continuous-time PLRNN agrees with the discrete-time solution also in this case of chaotic behavior.*

Example 4. *As a final example we applied the continuous-time transform developed here to a PLRNN inferred from empirical time series, namely human fMRI data. A discrete PLRNN with $M = 10$ latent states was used for this purpose that had been trained on multivariate ($N = 20$) time*

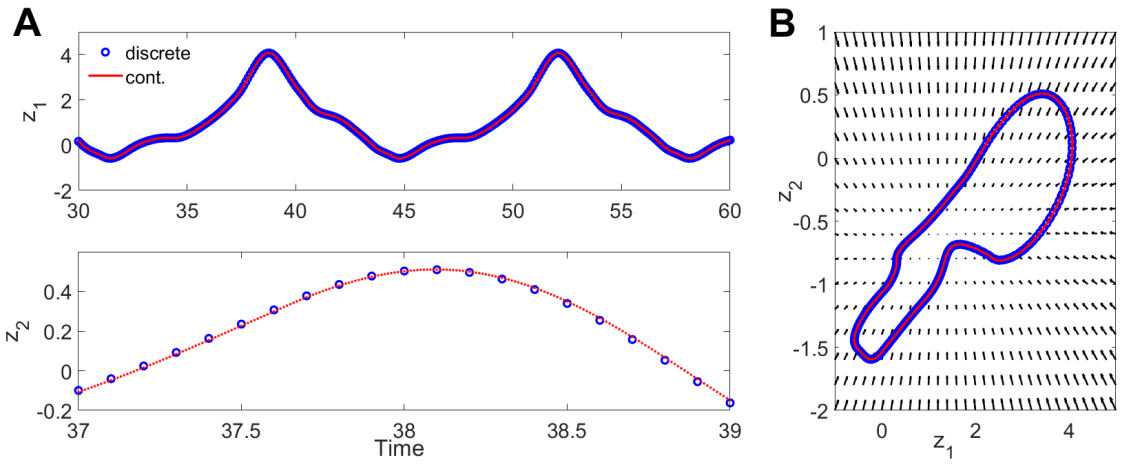

Figure 1. Transformation of a discrete PLRNN emulating the nonlinear van-der-Pol oscillator into a continuous-time ODE system. A) Time graphs for two of the system's 10 variables (unit activations). A zoom-in is provided for z_2 to better highlight how the continuous solution interpolates between the discrete time points. Blue = discrete PLRNN, red = continuous PLRNN. B) Continuous 2d-subspace of the 10-dimensional state space corresponding to the two variables shown in A, with flow fields (black arrows) and the system's trajectory on the limit cycle (red = continuous, blue circles = discrete); note that since this is only a 2d section of a 10-variable system, convergence to the limit cycle cannot be fully assessed from the (z_1, z_2) vector field.

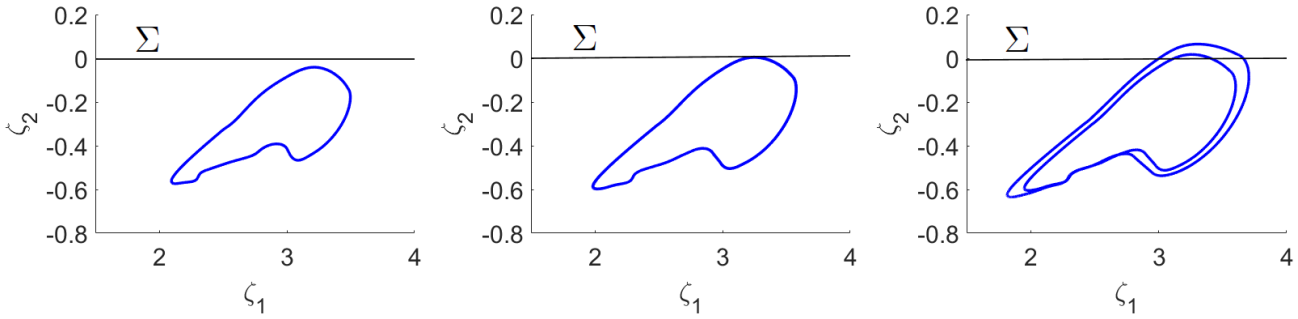

Figure 2. Grazing bifurcation in the continuous PLRNN derived from the van-der-Pol oscillator (Example 1) in the (ζ_1, ζ_2) subspace. The system undergoes a bifurcation as weight parameters $\tilde{w}_{21}^{(1)}$ and $\tilde{w}_{21}^{(2)}$ are decreased from $\tilde{w}_{21}^{(1)} > \tilde{w}_{bif}^{(1)}$ and $\tilde{w}_{21}^{(2)} > \tilde{w}_{bif}^{(2)}$ to $\tilde{w}_{21}^{(1)} < \tilde{w}_{bif}^{(1)}$ and $\tilde{w}_{21}^{(2)} < \tilde{w}_{bif}^{(2)}$, where the system's trajectory tangentially touches the border Σ (center panel) and the behavior changes from simple-periodic (left panel) to a period-2 limit cycle (right panel).

series of Blood Oxygenation Level Dependent (BOLD) signals recorded from human subjects performing a cognitive task while lying in a fMRI scanner. Details on the experimental procedure and task and on PLRNN training can be found in (Koppe et al., 2019) (briefly, an Expectation-Maximization algorithm was used for PLRNN training to maximize the evidence-lower-bound [ELBO] of the data log-likelihood). Theorem 3 could be applied in this case to translate the discrete-time system to continuous time, as all system's matrices met the conditions of this theorem. As Fig. 5A demonstrates, the continuous-time PLRNN (red curves) smoothly interpolates between the predictions produced by the discrete PLRNN (blue circles), and thus smoothly extrapolates among the observed data points when started from an initial condition inferred from the experimental data (Fig. 5B, green circles).

6. Conclusions

The aim of the present article was to show how to convert discrete-time into mathematically equivalent continuous-time RNN. As pointed out in the Introduction, sect. 1, and sect. 4.2, while RNN are usually trained in discrete time, a continuous-time description enjoys multiple advantages when it comes to analyzing the inferred systems and linking them to scientific theories. Here we examined such a transformation for a particular class of RNN based on ReLU activation functions. ReLU transfer functions are by now the most common choice in the deep learning community due to their piecewise constant gradients (Goodfellow et al., 2016; Lin & Jegelka, 2018; Montúfar et al., 2014), easing gradient-descent based algorithms. ReLU-based RNN are computationally and dynamically universal (Kimura &

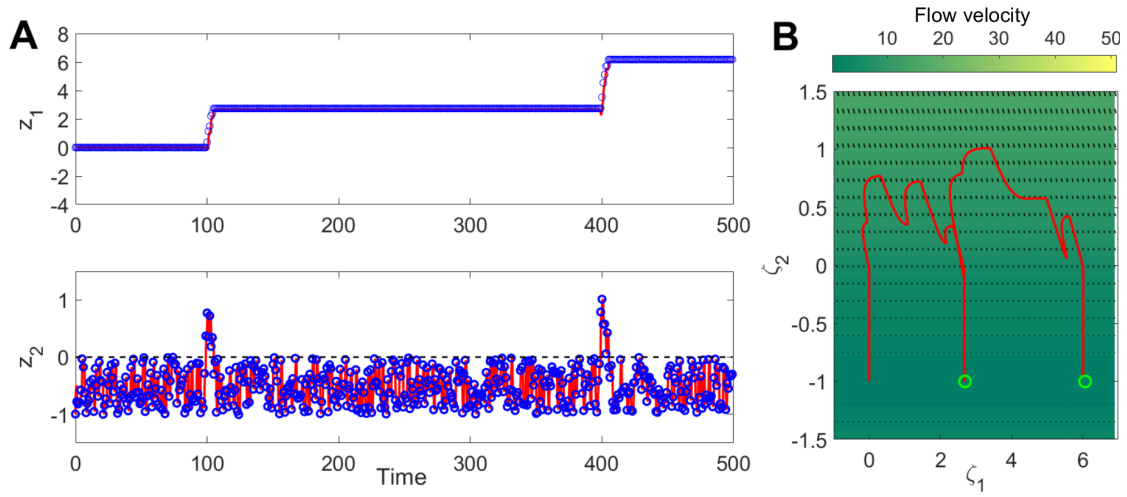

Figure 3. Discrete and corresponding continuous time PLRNN solution to the addition problem (Hochreiter & Schmidhuber, 1997). A) Time graphs for the 2 units of the discrete (blue circles) and continuous (red curves) PLRNN, where z_1 sums up the inputs conveyed through z_2 whenever its activity crosses 0 (dashed black line). B) State space of continuous-time PLRNN with flow field (black arrows) and trajectory (red) on the addition task. Color coding indicates magnitude of flow (vector length; lighter colors = steeper gradients). Green circles mark the true sums of inputs s_1 after the first and second time interval where $s_2 = 1$. Note that the system's final state on the ζ_1-axis correctly reports the total sum.

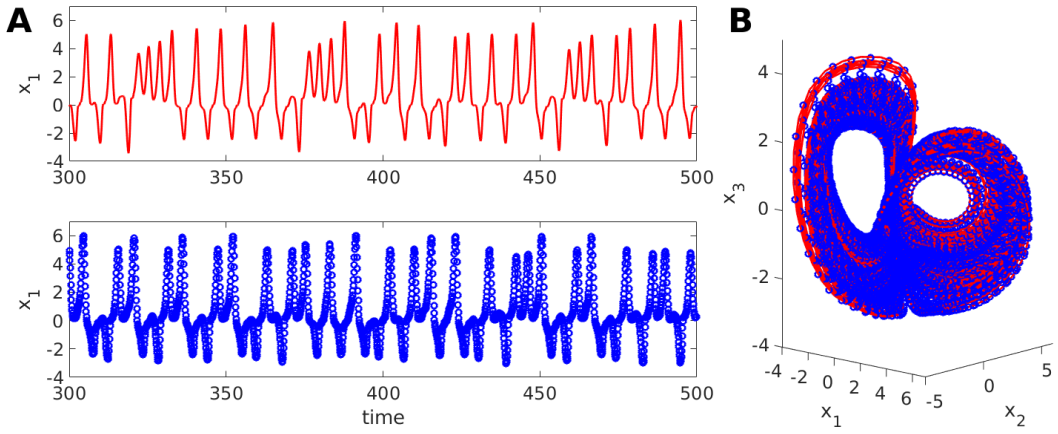

Figure 4. Transformation of a discrete PLRNN emulating the Lorenz equations within the chaotic regime into a continuous-time ODE system. A) Time graphs of one of three observation variables produced by projecting the system's 10 latent variables (unit activations) into a 3d observation space. Top: Continuous time solution (red); bottom: discrete time solution (blue). Note that since this system is chaotic, time graphs will never precisely overlap since any small numerical difference will lead to exponential divergence of trajectories (here, the recurrence matrices were only close to real for the continuous system and imaginary parts were set to 0). B) 3d-projection of the 10-dimensional PLRNN state space corresponding to the three observation variables, exposing the 'butterfly-wing-type' structure of the chaotic Lorenz attractor in discrete (blue circles) and continuous (red lines) time.

Nakano, 1998; Koiran et al., 1994; Lu et al., 2017), can outperform LSTMs on long short-term memory problems (Le et al., 2015; Schmidt et al., 2020), and can be efficiently inferred from time series data for approximating the underlying (nonlinear) dynamical system (Koppe et al., 2019). Hence our focus on the class of piecewise-linear RNN is not very restrictive, but instead encompasses a whole powerful family of RNN architectures and algorithms. Specifically, we proved that such a conversion from discrete to continuous time is possible under a variety of conditions. These include situations where one or more of the eigenvalues of the system's Jacobian are equal to 1, as required for long short-term maintenance (Example 2), or when we have cycles in the discrete case, leading to complex eigenvalues in the continuous case, as we illustrated in our specific applicational examples. Future work may address the problem of

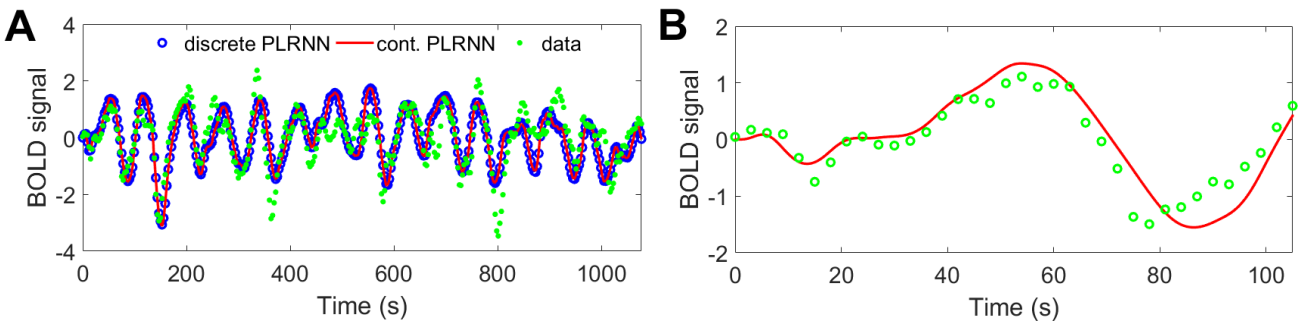

Figure 5. Application of continuous-time transform of PLRNN to human fMRI data. Blue circles in A) show predictions generated from a trained discrete PLRNN (taken from (Koppe et al., 2019)) started from an initial condition inferred from the data (green circles). Red curves show smooth inter-/extrapolations produced by the equivalent continuous-time PLRNN. The PLRNN also received inputs (simulating external stimuli) that were translated the same way from discrete to continuous time as described in Example 2 (for more details see accompanying code and (Koppe et al., 2019)). B) provides a zoom into the initial prediction period, where continuous-PLRNN extrapolations were produced at 10 times finer resolution than provided by the experimental data (one scan obtained every 3s).

discrete-to-continuous-time conversion more generally, for arbitrary nonlinear activation functions, or may attempt to find useful solutions for those cases where a direct translation is not easily possible, e.g. when the recurrence matrix W_{Ω^k} is not invertible, when the ODE system corresponding to the recursive map would need to have different dimensionality (as may occur, e.g., for low-dimensional chaotic maps like the logistic map), or when the time step Δt is not constant. Another valuable extension, especially in the context of latent variable models, would be to *stochastic* DS.

Acknowledgements

This work was funded by the German Science Foundation (DFG) through individual grant Du 354/10-1 to DD, and via the Excellence Cluster 'Structures' at Heidelberg University (EXC-2181/1 – 390900948). We thank Dr. Georgia Koppe for kindly lending us the fMRI data and corresponding PLRNN parameters used for the empirical example in Fig.5.

References

Abarbanel, H. D. I., Rozdeba, P. J., and Shirman, S. Machine learning: Deepest learning as statistical data assimilation problems. *Neural Comput.*, 30:2025–2055, 2018.

Absil, P.-A. Continuous-time systems that solve computational problems. *IJUC*, 2:291–304, 2006.

Bhatia, R. *Positive definite matrices*. Princeton Univ. Press, 2007.

Chang, B., Chen, M., Haber, E., and Chi, E. H. Antisymmetricrnns: a dynamical system view on recurrent neural networks. *ICLR 2019 Conference*, 2019.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in Neural Information Processing Systems 31 (NIPS 2018)*, 2018.

de Brouwer, E., Simm, J., Arany, A., and Moreau, Y. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *Neural Information Processing Systems Conference (NIPS)*, 2019.

di Bernardo, M. and Hogani, S. Discontinuity-induced bifurcations of piecewise smooth dynamical systems. *Philos. Trans. R. Soc*, A 368:4915–4935, 2010.

Doya, K. Bifurcations in the learning of recurrent neural networks. In *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 2777–2780, San Diego, CA, 1992.

Durstewitz, D. Self-organizing neural integrator predicts interval times. *Journal of Neuroscience*, 23(12):5342–5353, 2003.

Durstewitz, D. A state space approach for piecewiselinear recurrent neural networks for reconstructing nonlinear dynamics from neural measurements. *PLoS Computational Biology*, 13(6):e1005542, 2017.

Durstewitz, D. and Gabriel, T. Dynamical basis of irregular spiking in nmda-driven prefrontal cortex neurons. *Cereb Cortex.*, 17(4):894–908, 2007.

Funahashi, K. and Nakamura, Y. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6(6):801–806, 1993.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Hanson, J. and Raginsky, M. Universal simulation of stable dynamical systemsby recurrent neural nets. *Proceedings of Machine Learning Research*, 120:1–9, 2020.

Haschke, R. *Bifurcations in discrete-time neural networks: controlling complex network behaviour with inputs*. PhD thesis, Bielefeld University, 2004.

Higham, N. *Functions of matrices: Theory and computation*. SIAM, 2008.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–80, 1997.

Jordan, I. D., Sokol, P. A., and Park, I. M. Gated recurrent units viewed through the lens of continuous time dynamical systems. *arXiv preprint, abs/1906.01005*, 2019.

Kimura, M. and Nakano, R. Learning dynamical systems by recurrent neural networks from orbits. *Neural Networks*, 11(9):1589–1599, 1998.

Koch, C. and Segev, I. *Methods in Neuronal Modeling*. MIT Press, 2nd edition, 2003.

Koiran, P., Cosnard, M., and Garzon, M. H. Computability with low-dimensional dynamical systems. In *Proceedings of Machine Learning Research*, pp. 113–128, 1994.

Koppe, G., Toutounji, H., Kirsch, P., Lis, S., and Durstewitz, D. Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fmri. *PLoS Computational Biology*, 15(8):e1007263, 2019.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., and Socher, R. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of the 33th International Conference on Machine Learning, PMLR 48*, pp. 1378–1387, 2016.

Le, Q. V., Jaitly, N., and Hinton, G. E. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint:1504.00941*, 2015.

Lin, H. and Jegelka, S. Resnet with one-neuron hidden layers is a universal approximator. *arXiv preprint:1806.10909*, 2018.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. *Advances in Neural Information Processing Systems*, 30: 6231–6239, 2017.

Milan, A., Rezatofighi, S. H., Dick, A., Reid, I., and Schindler, K. Online multi-target tracking using recurrent neural networks. *Proceedings of the thirty-first AAAI conference on artificial intelligence (AAAI-17)*, pp. 4225–4232, 2017.

Monfared, Z. and Durstewitz, D. Existence of n-cycles and border-collision bifurcations in piecewise-linear continuous maps with applications to recurrent neural networks. *Nonlinear dynamics, in press*, 2020. doi: 10.1007/s11071-020-05841-x.

Monfared, Z., Afsharnezhad, Z., and Esfahani, J. A. Flutter, limit cycle oscillation, bifurcation and stability regions of an airfoil with discontinuous freeplay nonlinearity. *Nonlinear dynamics*, 90:1965–1986, 2017.

Montúfar, G., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. *arXiv preprint:1402.1869v2*, 2014.

Nunemacher, J. Which real matrices have real logarithms? *Mathematics Magazine*, 62(2):132–135, 1989.

Nykamp, D. Q. From discrete dynamical systems to continuous dynamical systems. *From Math Insight*, 2019. URL http://mathinsight.org/from_discrete_to_continuous_dynamical_systems.

Ozaki, T. *Time Series Modeling of Neuroscience Data*. CRC Press, 2012.

Pearlmutter, B. A. Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1:263–269, 1989.

Perko, L. *Differential Equations and Dynamical Systems*. Springer-Verlag, 1991.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes: The art of scientific computing (Third edition)*. Cambridge University Press, third edition, 2007.

Razaghi, H. S. and Paninski, L. Filtering normalizing flows. *4th workshop on Bayesian Deep Learning (NeurIPS 2019)*, 2019.

Reitmann, V. *Regular and chaotic dynamics*. Mathematics for engineers and scientists book series (MFIN), Teubner, 1996.

Rossetto, B., Lenzini, T., Ramdani, S., and Suchey, G. Slow-fast autonomous dynamical systems. *International Journal of Bifurcation and Chaos*, 8:2135–2145, 1998.

Schmidt, D., Koppe, G., Beutelspacher, M., and Durstewitz, D. Inferring dynamical systems with long-range dependencies through line attractor regularization. *arXiv:1910.03471v2*, 2020.

Seung, H. S. How the brain keeps the eyes still. *Proceedings of the National Academy of Sciences*, 93(23): 13339–13344, 1996.

Sherif, N. and Morsy, E. Computing real logarithm of a real matrix. *International Journal of Algebra*, 2(3):131–142, 2008.

Siegelmann, H. T. and Sontag, E. D. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132–150, 1995.

Strogatz, S. H. *Nonlinear Dynamics and Chaos: Applications to Physics, Biology, Chemistry, and Engineering: With Applications to Physics, Biology, Chemistry and Engineering*. CRC Press, 2015.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems (NIPS'14)*, pp. 3104–3112, 2014.

Trischler, A. P. and D'Eleuterio, G. M. Synthesis of recurrent neural networks for dynamical system simulation. *Neural Networks*, 80:67–78, 2016.

Vlachas, P., Byeon, W., Wan, Z., Sapsis, T., and Koumoutsakos, P. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474:20170844, 2018.

Zaheer, M., Ahmed, A., and Smola, A. J. Latent lstm allocation joint clustering and non-linear dynamic modeling of sequential data. *Proceedings of the* 34^{rd} *International Conference on Machine Learning, PMLR 70*, pp. 3967–3976, 2017.

Zhao, Y. and Park, I. M. Variational joint filtering. *arXiv:1707.09049*, 2017.