The effect of the forget gate on bifurcation boundaries and dynamics in Recurrent Neural Networks and its implications for gradient-based optimization

1st Alexander Rehmer
Department of Measurement and Control
University of Kassel
Kassel, Germany
alexander.rehmer@mrt.uni-kassel.de

2nd Andreas Kroll
Department of Measurement and Control
University of Kassel
Kassel, Germany
andreas.kroll@mrt.uni-kassel.de

***Abstract*—Recurrent Neural Networks (RNNs) are an internal dynamics approach to identify models from time series data. They have been successfully applied, e.g. in natural language, speech and video processing [1] and the identification of nonlinear state space models [2]. Conventional RNNs, such as the Elman-RNN, are notoriously hard to optimize, since they are highly initialization dependent, prone to slow convergence, and tend to converge to poor local minima. In recent years, the *vanishing/exploding gradient* phenomenon, which arises when employing gradient-based optimization techniques such as Backpropagation Through Time (BPTT), has been identified as the root cause of these difficulties. This led to the development of several new RNN-architectures, such as the Long Short-Term Memory (LSTM) and the Gated Recurrent Unit (GRU), which were intended to prevent the vanishing-gradient problem and surpassed conventional RNNs in all areas of application. However, it has been shown that the gradient also vanishes in Gated Units [3] and there is no work showing, that the rate of decay is lower than in Elman-RNN. This suggests that the underlying mechanisms responsible for their success are at least in part not yet fully understood. The purpose of this paper is to provide an alternative explanation for the superior performance of Gated Units by viewing them as nonlinear dynamical systems and studying the stability of their fixed points. This work expands on the work of Doya et al. [4] and Pascanu et al. [5], who studied the effects of bifurcation boundaries in the parameter space of Elman-RNNs with one internal state on gradient-based learning.**

***Index Terms*—RNN, LSTM, GRU, bifurcations, nonlinear dynamics**

I. Introduction

The training of recurrent model structures, such as Elman-RNNs, is a notoriously hard optimization problem. The vanishing/exploding-gradient phenomenon caused by the nested application of a single recurrent activation function and its weights has been identified as the root-cause of these difficulties [QUELLE]. This lead to the development of recurrent model structures trying to avoid the nesting as far as possible, most prominently the LSTM [6] and GRU [7]. More recent work by v. d. Westhuizen and Lasenby [8] has empirically shown that the *forget gate*, one of four gates in the LSTM, is solely responsible for its superior performance over the Elman-RNN. The forget gate is however precisely the part in the LSTM network, that causes its internal state to vanish, and therefore also the gradient during BPTT. This clearly contradicts the hypothesis, that Gated Units are easier to train because they solve the vanishing-gradient problem. An alternative explanation for the difficulty of training Elman-RNN was provided by Pascanu [5]. By viewing Elman-RNNs as nonlinear dynamical discrete-time systems

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k; \boldsymbol{\theta}\right) \tag{1}$$

with internal state $\boldsymbol{x}_k$ and system parameters $\boldsymbol{\theta}$, it was shown that bifurcation boundaries exist in the parameter-space of the Elman-RNN. When crossing these boundaries, the dynamics of the Elman-RNN change completely, which is associated with a large local gradient. If bifurcation boundaries are encountered during gradient-based optimization, the large local gradient will disturb the training process. Unfortunately, these bifurcation boundaries are located in a domain of the parameter space of the Elman-RNN, where parameters are typically initialized. Crossing them during training is therefore highly likely. As will be shown in this paper, bifurcation boundaries also exist in Gated Units. Depending on the parameterization of the forget gate, these boundaries are however pushed outside of the domain where parameters are typically initialized, making them less likely to become a problem during training. The paper is organized as follows: First codimension-1 bifurcations [9], simply denoted as bifurcations from here on, and their necessary conditions are introduced. Secondly, the analysis of bifurcations in Elman-RNNs conducted in [5] will be expanded on by not only considering networks with one state but also with two states and by considering bifurcations with regard to all network parameters, not only the biases. Subsequently, the LSTM and its forget gate only version, denoted Gated Unit (GU), will be introduced. It will be shown that the parameterization of the forget gate determines the location of the bifurcation boundaries of the GU and that the

bifurcation boundaries of Elman-RNN and GU are identical for the limit case of a fully closed forget gate. From this analysis it can then be concluded, that (at least for reasonable parameterizations of the forget gate) the bifurcation boundaries of Gated Units are located so far from the origin of the parameter space, that are very unlikely to disturb the training process.

II. Related Work

The existence of codimension-1 bifurcations in Elman-RNN and their potentially problematic implications were acknowledged by Doya et al. [4] without analyzing where bifurcations occur in the parameter space. The matter was elaborated on by Pascanu et al. [5], who showed that locally large gradients exist in the parameter space of the Elman-RNN by example. A more systematic treatment of bifurcation boundaries in RNN was conducted by Haschke and Steil [9]. As opposed to the other works, the input to the RNN and not its parameters were treated as the bifurcation parameter. Marichal et al. [10] studied fold bifurcations in Elman-RNNs with two internal states, period-doubling bifurcations were not considered. To the best of the authors knowledge there is no work encompassing an analysis of all types of codimension-1 bifurcations in Elman-RNNs as well as Gated Units. Also the superior performance of Gated Units compared to Elman-RNNs has never been linked to their difference regarding the location or existence of bifurcation boundaries.

III. Codimension-1 Bifurcations

Naturally, the behaviour of a dynamic system like a RNN changes depending on its parameters $\boldsymbol{\theta}$, i.e. its weights and biases. Usually, an infinitesimal change in the systems parameters leads to an infinitesimal change in its dynamic behaviour. This is however not true at a bifurcation point, which is a set of critical parameters $\boldsymbol{\theta}_{\mathrm{crit}}$ that marks the transition between qualitatively different dynamic behaviors, i.e. from a stable to an unstable system. In this case, an infinitesimal change in the system parameters will lead to fundamentally different dynamics. Such bifurcation points form manifolds in the parameter space, i.e. bifurcation boundaries, which separate qualitatively different dynamic behaviors [9]. These bifurcation boundaries have been identified by [5] to be detrimental for gradient-based parameter optimization. Since fundamentally different dynamics will usually lead to vastly different values of the loss function, bifurcation boundaries are usually associated with high local gradients. Encountering a bifurcation boundary during optimization can therefore throw the optimizer off course, since the magnitude of the update is proportional to the gradient.

Local bifurcations occur, when a change in the system parameters causes the stability of an equilibrium or fixed point $\boldsymbol{x}_{\mathrm{F}}$ to change. A local codimension-1 bifurcation of a fixed point $\boldsymbol{x}_{\mathrm{F}}$ is defined by the fixed point condition

$$\boldsymbol{x}_{\mathrm{F}} = \boldsymbol{f}\left(\boldsymbol{x}_{\mathrm{F}}; \boldsymbol{\theta}\right) \tag{2}$$

TABLE I: Codimension-1 fixed point bifurcation types and necessary eigenvalue conditions

eigenvalue	bifurcation type	defintion
$\lambda_i = +1$	Fold	Change in system's parameters leads to sudden creation of a pair of fixed points
$\lambda_i = -1$	Period-doubling	Change in system's parameters causes a periodic trajectory, in cases analysed in the following with period two, to emerge
$\lambda_{i,i+1} = e^{\pm j\omega}$	Neimark-Sacker	Change in system's parameters cause a fixed point to loose its stability and a periodic solution arises

and an additional condition on the eigenvalues λ_i of the Jacobian evaluated at $\boldsymbol{x}_{\mathrm{F}}$.

$$\boldsymbol{J}\left(\boldsymbol{x}\right) = \left.\frac{\partial \boldsymbol{f}\left(\boldsymbol{x}; \boldsymbol{\theta}\right)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_{\mathrm{F}}} \tag{3}$$

A bifurcation occurs, i.e. $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathrm{crit}}$, if the eigenvalues of $\boldsymbol{J}\left(\boldsymbol{x}_{\mathrm{F}}\right)$ leave the unit circle due to variation of $\boldsymbol{\theta}$. Depending on where the eigenvalues cross the unit circle, three different types of bifurcations can be distinguished, see Tbl. I.

IV. Bifurcation boundaries in Elman-RNN

The Elman-RNN, as depicted in Fig. 1, is a straightforward realization of a discrete time nonlinear state space model:

$$\boldsymbol{x}_{k+1} = \mathbf{tanh}\left(\boldsymbol{W}_{\mathrm{c}}\boldsymbol{x}_k + \boldsymbol{W}_{\mathrm{u}}\boldsymbol{u}_k + \boldsymbol{b}_{\mathrm{c}}\right) \tag{4}$$

The internal dynamics are realized via a single layer of n_x recurrent neurons that map the input $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$ and the internal state from the previous time step $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ to the next internal state $\boldsymbol{x}_{k+1}$. Nonlinear behaviour is realized via $\tanh$-activation functions $\boldsymbol{f}_{\mathrm{h}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$. The model parameters are $\boldsymbol{W}_{\mathrm{c}} \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{W}_{\mathrm{u}} \in \mathbb{R}^{n_x \times} n_u$ and $\boldsymbol{b}_{\mathrm{h}} \in \mathbb{R}^{n_x}$. Usually a layer of feedforward neurons maps the internal state to the output. This layer is omitted here, since it does not affect the internal dynamics of the Elman-RNN, which are the subject of investigation. Furthermore, the input $\boldsymbol{u}_k$ does not affect the location or stability of the fixed points $\boldsymbol{x}_{\mathrm{F}}$ of (4) and will therefore be neglected, leading to the state equation of the autonomous Elman-RNN:

$$\boldsymbol{x}_{k+1} = \mathbf{tanh}\left(\boldsymbol{W}_{\mathrm{c}}\boldsymbol{x}_k + \boldsymbol{b}_{\mathrm{c}}\right) \tag{5}$$

According to (2) a fixed point $\boldsymbol{x}_{\mathrm{F}}$ of the Elman-RNN fulfils the condition

$$\boldsymbol{x}_{\mathrm{F}} = \mathbf{tanh}\left(\boldsymbol{W}_{\mathrm{c}}\boldsymbol{x}_{\mathrm{F}} + \boldsymbol{b}_{\mathrm{c}}\right). \tag{6}$$

The Jacobian $\boldsymbol{J}^{\mathrm{RNN}}(\boldsymbol{x})$ of the Elman-RNN (5) is

$$\boldsymbol{J}^{\mathrm{RNN}}(\boldsymbol{x}) = \mathbf{tanh}'(\cdot)\boldsymbol{W}_{\mathrm{c}} \tag{7}$$

with $\mathbf{tanh}'(\cdot) = \mathrm{diag}\left(\mathbf{1} - \mathbf{tanh}^2\left(\boldsymbol{W}_{\mathrm{c}}\boldsymbol{x} + \boldsymbol{b}_{\mathrm{c}}\right)\right)$. By inserting (6) in (7) this can be further simplified to

$$\boldsymbol{J}^{\mathrm{RNN}}(\boldsymbol{x}_{\mathrm{F}}) = \begin{bmatrix} (1 - x_{\mathrm{F},1}^2)w_{\mathrm{c},11} & (1 - x_{\mathrm{F},1}^2)w_{\mathrm{c},12} \\ (1 - x_{\mathrm{F},2}^2)w_{\mathrm{c},21} & (1 - x_{\mathrm{F},2}^2)w_{\mathrm{c},22} \end{bmatrix}. \tag{8}$$

In the following, bifurcations in Elman-RNNs with a single internal state, i.e. $\dim\left(\boldsymbol{x}\right) = 1$ are studied. In this setting the

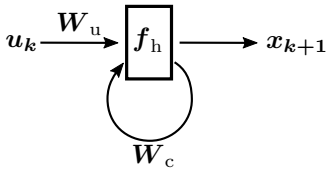


Fig. 1: Elman-RNN: Layers of neurons are represented as rectangles.

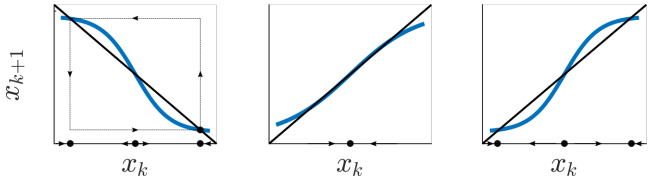


Fig. 2: Graphs of a 1d Elman-RNN's state equation depending on the recurrent weight w_c. Fixed points are marked by filled cirlces and their stability is indicated by arrows. Left: For $w_\mathrm{c} \leq -1$ the RNN exhibits one unstable equilibrium at the origin and a stable oscillation with period 2 (limit cycle). Middle: For $|w_\mathrm{c}| \leq 1$ the RNN exhibits one globally asymptotic stable equilibrium at the origin. Right: For $w_\mathrm{c} \geq 1$ the RNN exhibits one unstable equilibrium at the origin and two stable equilibria.

parameter space is merely two dimensional, which enables a visualization of the bifurcation boundaries. Afterwards the analysis is extended to Elman-RNN with two internal states, i.e. $\dim(\boldsymbol{x}) = 2$.

A. Bifurcation boundaries in 1d Elman-RNN

An Elman-RNN with a single internal state hase a two-dimensional parameter space: $\boldsymbol{\theta} = [w_\mathrm{c}, b_\mathrm{c}]$ with $w_\mathrm{c} \in \mathbb{R}$ and $b_\mathrm{c} \in \mathbb{R}$. To begin with, the bias b_c is set to zero, i.e. $b_c = 0$. In this case the only fixed point is the origin of the state space $x_\mathrm{F} = 0$ and the eigenvalue of the Jacobian amounts to

$$\lambda = w_\mathrm{c} \tag{9}$$

which implies a fold bifurcation point at $\boldsymbol{\theta}^\mathrm{f}_\mathrm{crit} = [1, 0]$ and a period-doubling bifurcation point at $\boldsymbol{\theta}^\mathrm{pd}_\mathrm{crit} = [-1, 0]$. The dynamic behaviour of the Elman-RNN before and after crossing either bifurcation point can be qualitatively assessed by examining the graph of the r.h.s of its state equation (5), which is shown in Fig. 2. The graph in the middle is representative for all parameterizations $|w_\mathrm{c}| \leq 1$, which produce a system with a single stable fixed point at the origin. The graph on the left resp. right shows the additional fixed points created after crossing the period-doubling bifurcation point $\boldsymbol{\theta}^\mathrm{pd}_\mathrm{crit} = [-1, 0]$ resp. the fold bifurcation point $\boldsymbol{\theta}^\mathrm{f}_\mathrm{crit} = [1, 0]$. In the general case, i.e. $b_\mathrm{c} \neq 0$, the fixed point at the origin is shifted to the left resp. right, which also affects the critical values of $w_\mathrm{c,crit}$ at which bifurcations occur. In order to map out the bifurcation boundaries in the general case, for a given parameterization $\boldsymbol{\theta} = [w_\mathrm{c}, b_\mathrm{c}]$, the fixed point is calculated by solving (6) numerically. By subsequently calculating the eigenvalues of $\boldsymbol{J}^\mathrm{RNN}(\boldsymbol{x}_\mathrm{F})$ the stability of the

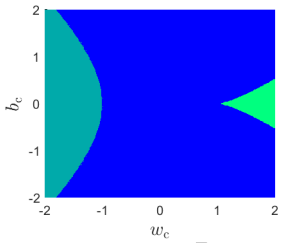


Fig. 3: Regions of qualitatively different behaviour of the 1d autonomous Elman-RNN with $b_\mathrm{c} = 0$: RNN has one globally asymptotically stable equilibrium ($|\lambda| \leq 1$), RNN has three equilibria ($\lambda > 1$), RNN exhibits an oscillation with period 2 ($\lambda < 1$).

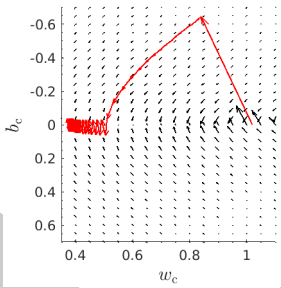


Fig. 4: Steps taken by gradient-based optimization of an Elman-RNN when initialized near a bifurcation point. Initial parameters: $w_\mathrm{c,init} = 1.02$, $b_\mathrm{c,init} = 0$. Optimal parameters: $w_\mathrm{c,opt} \approx 0.53$, $b_\mathrm{c,opt} \approx 0$

fixed point and hence the dynamical behavior of the Elman-RNN can be evaluated. This can be seen in Fig. 3, which shows the parameter space of the 1d Elman-RNN in the range $w_\mathrm{c} = [-2, 2]$, $b_\mathrm{c} = [-2, 2]$ and its resulting dynamic behavior. Three regions of qualitatively different dynamics can clearly be distinguished, which are seperated by bifurcation boundaries. In order to visualize the effect of bifurcation boundaries on the local gradient and therefore their relevance in gradient-based optimization, the gradient $\frac{\partial x}{\partial \boldsymbol{\theta}}$ and its norm $\left|\frac{\partial x}{\partial \boldsymbol{\theta}}\right|$ were calculated. Fig. 5 shows that large gradients occur along the bifurcation boundaries, while the gradients in the remainder of the parameter space are rather small. Hitting a bifurcation boundary during optimization can therefore result in being thrown out into the low-gradient regions of the parameter space from which it takes a large number of iterations to return. This effect is shown in Fig. 4, which shows the optimization of an Elman-RNN which was initialized near a bifurcation point with vanilla gradient descent.

B. Bifurcation boundaries in 2d Elman-RNN

The parameter space of an Elman-RNN with two internal states is six-dimensional with $\boldsymbol{\theta} = [\boldsymbol{W}_\mathrm{c}, \boldsymbol{b}_\mathrm{c}]$, $\boldsymbol{W}_\mathrm{c} \in \mathbb{R}^{2\times 2}$, $\boldsymbol{b}_\mathrm{c} \in \mathbb{R}^2$. A straighforward visualization of the bifurcation boundaries as in the 1d case is therefore not possible.

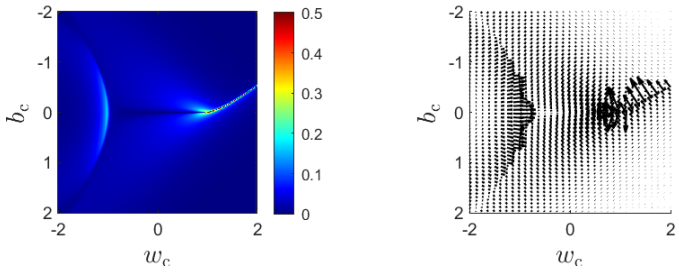


Fig. 5: Magnitude of the gradient $|\frac{\partial \boldsymbol{x}(\boldsymbol{\theta_i})}{\partial [w_c, b_c]}|$ (left) and gradient $\frac{\partial \boldsymbol{x}(\boldsymbol{\theta_i})}{\partial [w_c, b_c]}$ (right) of a 1d Elman-RNN w.r.t. its parameters.

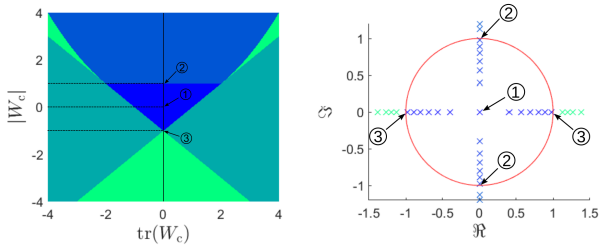


Fig. 6: Left: Regions of qualitatively different behaviour of the 2d autonomous Elman-RNN with $\boldsymbol{b}_c = \boldsymbol{0}$: $|\lambda_{1,2}| \leq 1$: one globally asymptotically stable equilibrium, $\lambda_{1,2} = e^{j\omega}, |\lambda_{1,2}| > 1$: equilibrium is a periodic oscillation, $|\lambda_{1,2}| > 1$: fixed point lost stability in both directions of $\boldsymbol{x}$, $(|\lambda_1| \leq 1 \wedge |\lambda_2| > 1) \vee (|\lambda_1| \leq 1 \wedge |\lambda_2| > 1)$ fixed point lost stability in at least one direction. Right: Eigenvalues of $\boldsymbol{J}^{\mathrm{RNN}}$ depending on $|\boldsymbol{W}_c|$ for $\mathrm{tr}(\boldsymbol{W}_c) = 0$.

In the two-dimensional case the Elman-RNNs eigenvalues of its Jacobian (7) evaluated at a fixed point $\boldsymbol{x}_F$ are

$$\lambda_{1,2} = \frac{\mathrm{tr}(\mathbf{tanh}'(\cdot)\boldsymbol{W}_c)}{2} \pm \sqrt{\frac{\mathrm{tr}(\mathbf{tanh}'(\cdot)\boldsymbol{W}_c)^2}{4} - |\mathbf{tanh}'(\cdot)|\,|\boldsymbol{W}_c|} \tag{10}$$

$|\cdot|$ denotes the determinant of a matrix.

To begin with the bias is set to zero $\boldsymbol{b}_c = \boldsymbol{0}$ so one can examine how the stability of the fixed point $\boldsymbol{x}_F = \boldsymbol{0}$ changes depending on $\boldsymbol{W}_c$. In this case $\mathbf{tanh}'(\cdot) = \boldsymbol{I}$ and the eigenvalues of $\boldsymbol{J}^{\mathrm{RNN}}(\boldsymbol{x}_F)$ become:

$$\lambda_{1,2} = \frac{\mathrm{tr}(\boldsymbol{W}_c)}{2} \pm \sqrt{\frac{\mathrm{tr}(\boldsymbol{W}_c)^2}{4} - |\boldsymbol{W}_c|}. \tag{11}$$

I.e. in this simplified case the eigenvalues of $\boldsymbol{J}^{\mathrm{RNN}}(\boldsymbol{0})$ can be expressed in terms of the trace and determinant of the weight matrix $\boldsymbol{W}_c$. By evaluating (11) at any point in the $[\mathrm{tr}(\boldsymbol{W}_c), |\boldsymbol{W}_c|]$-space the stability of the fixed-point $\boldsymbol{x}_F$ and therefore the dynamic behavior of the Elman-RNN can be determined. Fig. 6 (left) shows which regions in the $[\mathrm{tr}(\boldsymbol{W}_c), |\boldsymbol{W}_c|]$-space result in which dynamic behavior by color-coding. From Fig. 6 it is apparent, that the region in the parameter space, which results in a single stable fixed point, is relatively small, compared to the parameterizations

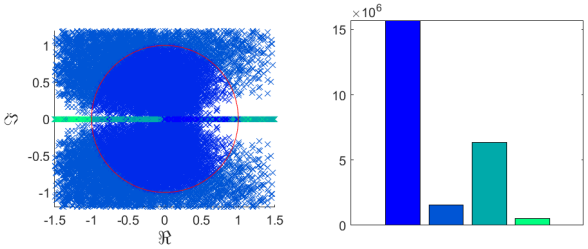


Fig. 7: Left: Regions of qualitatively different behavior of the 2d autonomous Elman-RNN with $\boldsymbol{b}_c \neq \boldsymbol{0}$. Right: Histogram of parameterizations resulting in respective dynamic behavior.

that produce either limit-cycles (after crossing period-doubling or Neimark-Sacker bifurcation points) or render the origin an unstable equilibrium (after crossing fold bifurcation points). This is clearly a shortcoming of the Elman-RNN: Most real-world systems dynamics follow the law of a decaying internal state that approaches a certain equilibrium, i.e. they are stable. Systems that exhibit limit-cycles, on the other hand, are rather rare. The fact that most parameterizations of the Elman-RNN yield a model with dynamics that will only in very rare cases match that of the target system is a clear case of model-system mismatch. Also, the considered domain of the parameter space is interspersed with bifurcation boundaries. If the model is initialized in one dynamic regime but the optimium lies in another, a bifurcation boundary has to be crossed. Due to the large gradient associated with the bifurcation boundary, it is conceivable that the optimizer takes a step so large, that the regime containing the optimum is skipped entirely. This supports the hypothesis, that bifurcation boundaries pose a significant problem during gradient-based optimization.

In the general case, i.e. $\boldsymbol{b}_c \neq \boldsymbol{0}$, the fixed point in the origin is shifted, which affects the eigenvalues of $\boldsymbol{J}^{\mathrm{RNN}}$ and therefore the values of $\boldsymbol{W}_c$ at which bifurcations occur. Unfortunately, in this case a compact representation in terms of the trace and determinant of $\boldsymbol{W}_c$ is not possible since $\mathbf{tanh}'(\cdot) \neq \boldsymbol{I}$ and therefore the product in the trace operator cannot be seperated. Instead, a complete enumeration of the parameter space $\boldsymbol{\theta}$ was performed with all parameters varied in the range $[-1.6, 1.6]$ with an increment of 0.2 yielding a total of approximately $24 \cdot 10^6$ different parameterization. For each parameterization the fixed point and the corresponding eigenvalues $\lambda_{1,2}$ of $\boldsymbol{J}^{\mathrm{RNN}}(\boldsymbol{x}_F)$ were calculated and plotted in the complex plane in order to evaluate the stability of the fixed point. While this presentation does not allow to assess where in the parameter space the bifurcation boundaries occur, it is still possible to empirically evaluate how many of the examined parameterizations produce a certain dynamic behavior. From this can be concluded how large the corresponding areas in the parameter space are. The eingenvalues in the complex plane and a histogram of the occurences of the respective dynamics are given in Fig. 7.

As can be seen from Fig. 7 a considerable portion ($\approx 40\%$) of the tested parameterizations lie outside of the globally

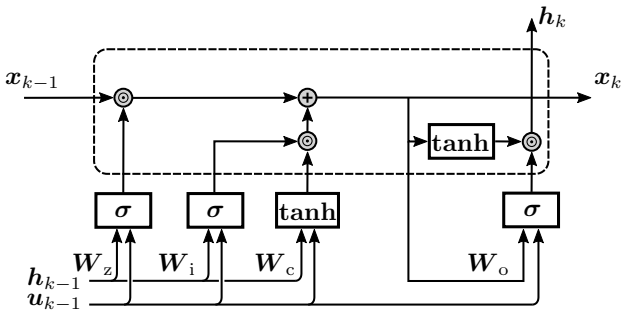


Fig. 8: Long short-term memory (LSTM): Gates are depicted as rectangles with their respective activation functions.

asymptotically stable domain.

V. Bifurcation boundaries in Gated Units

A. LSTM & GRU model structures

Gated Units, most notably the LSTM and GRU, are Recurrent Neural Networks whose architecture follows a different design paradigma than Elman-RNNs. Since the vanishing gradient, caused by the repeated mapping of the internal state through a set of reccurent weights and a reccurent activation function, was thought to be the root cause of the difficulties associated with training Elman-RNNs, Gated Units update the internal state in a different way that was intended to prevent the gradient from vanishing. Instead of obtaining the updated internal state by squashing the previous internal state through a nonlinear activation function, Gated Units only allow additive and multiplicative operations on the internal state through so called *gates*, which are one-layered neural networks. Most notable among all Gated Units is the LSTM, of which many variants were proposed in the last decade. Its model equations are given below and its structure is depicted in Fig. 8.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{z}_k \odot \boldsymbol{x}_k + \boldsymbol{i}_k \odot \mathbf{tanh}\left(\boldsymbol{W}_{\mathrm{c}} \tilde{\boldsymbol{x}}_k + \boldsymbol{b}_{\mathrm{c}}\right) \\ \boldsymbol{h}_{k+1} &= \boldsymbol{o}_k \odot \mathbf{tanh}\left(\boldsymbol{x}_{k+1}\right) \end{aligned} \tag{12}$$

with

$$\begin{aligned} \boldsymbol{z}_k &= \boldsymbol{\sigma}\left(\boldsymbol{W}_{\mathrm{z}} \cdot \left[\boldsymbol{x}_k^T, \boldsymbol{u}_k^T\right]^T + \boldsymbol{b}_{\mathrm{z}}\right), \\ \boldsymbol{i}_k &= \boldsymbol{\sigma}\left(\boldsymbol{W}_{\mathrm{i}} \cdot \left[\boldsymbol{x}_k^T, \boldsymbol{u}_k^T\right]^T + \boldsymbol{b}_{\mathrm{i}}\right), \\ \boldsymbol{o}_k &= \boldsymbol{\sigma}\left(\boldsymbol{W}_{\mathrm{o}} \cdot \left[\boldsymbol{x}_{k+1}^T, \boldsymbol{u}_k^T\right]^T + \boldsymbol{b}_{\mathrm{o}}\right), \end{aligned} \tag{13}$$

with $\tilde{\boldsymbol{x}}_k = \boldsymbol{f}_{\mathrm{r}} \odot \boldsymbol{x}_k$ and $\boldsymbol{W}_{\mathrm{r}}, \boldsymbol{W}_{\mathrm{z}}, \boldsymbol{W}_{\mathrm{c}} \in \mathbb{R}^{n_{\mathrm{x}} \times n_{\mathrm{x}} + n_{\mathrm{u}}}$, $\boldsymbol{b}_{\mathrm{r}}, \boldsymbol{b}_{\mathrm{z}}, \boldsymbol{b}_{\mathrm{c}} \in \mathbb{R}^{n_{\mathrm{x}}}$ and $\boldsymbol{f}_{\mathrm{r}}, \boldsymbol{f}_{\mathrm{z}}, \boldsymbol{f}_{\mathrm{c}} : \mathbb{R}^{n_{\mathrm{x}}} \to \mathbb{R}^{n_{\mathrm{x}}}$. $\boldsymbol{\sigma}(\cdot)$ denotes the logistic function.

While each gate is thought to have a certain purpose such as protecting the internal state from irrelevant inputs or forgetting information that has become irrelevant [6] there is no evidence to support that these gates actually perform these tasks. In the end, all gates are merely nonlinear functions of the internal state and input, that together constitute the r.h.s of the state equation. How easily and well this function can be adapted to the system to be modeled will determine model fit. In fact, research conducted in recent years showed that most of the gates can be removed without compromising performance. Of all things the forget gate $\boldsymbol{z}_k$ was found to be essential for the LSTMs performance, a fact that was not further discussed by the authors [11]. A forget gate only version of the LSTM even outperformed the original LSTM, having only halve the number of parameters [8]. This result is somewhat surprising: The LSTM was invented to prevent the gradient from vanishing. The forget gate however is exactly the part of the LSTM, which causes its gradient to vanish by multiplying the internal states with the activations $\boldsymbol{z}_k \in (0, 1)$ of logistic neurons, see (12).

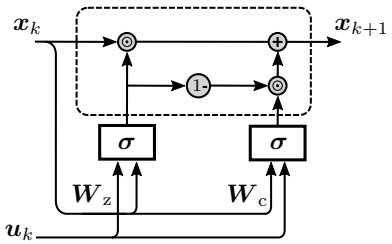


Fig. 9: Forget gate only version of the LSTM, denoted Gated Unit (GU).

In the following, the forget gate only version of the LSTM will be introduced. It will be shown, that the forget gate modifies the r.h.s of the Gated Units state equation in a way, that affects its bifurcation boundaries by shifting them away from the center to the outside of the parameter space where they are less likely to disturb the training process. Again, the input $\boldsymbol{u}_k$ does not change the location or stability of the fixed points, since they depend only on the parameters, and will therefore be neglected.

The state equation of the forget gate only version of the LSTM is

$$\boldsymbol{x}_{k+1} = \boldsymbol{z}_k \odot \boldsymbol{x}_k + (\mathbf{1} - \boldsymbol{z}_k) \cdot \mathbf{tanh}\left(\boldsymbol{W}_{\mathrm{c}} \cdot \boldsymbol{x}_k + \boldsymbol{b}_{\mathrm{c}}\right). \tag{14}$$

This RNN was named JANET in [8]. Since it is also identical to a forget gate only version of the GRU [7] and therefore representative for both LSTM and GRU, it will simply be denoted Gated Unit (GU) from here on. The GU's structure is depicted in Fig. 9. The Jacobian $\boldsymbol{J}^{\mathrm{GU}}(\boldsymbol{x})$ is

$$\begin{aligned} \boldsymbol{J}^{\mathrm{GU}}(\boldsymbol{x}) = \operatorname{diag}(\boldsymbol{x})\boldsymbol{Z}'(\cdot)\boldsymbol{W}_{\mathrm{z}} + \boldsymbol{Z}(\cdot) + \mathbf{tanh}'(\cdot)\boldsymbol{W}_{\mathrm{c}} \\ + \boldsymbol{Z}'(\cdot)\mathbf{tanh}(\cdot)\boldsymbol{W}_{\mathrm{z}} + \boldsymbol{Z}(\cdot)\mathbf{tanh}'(\cdot)\boldsymbol{W}_{\mathrm{c}} \end{aligned} \tag{15}$$

with $\boldsymbol{Z}(\cdot) = \operatorname{diag}(\boldsymbol{z})$ and $\boldsymbol{Z}'(\cdot) = \operatorname{diag}(\boldsymbol{z}(\mathbf{1} - \boldsymbol{z}))$.

B. 1d Gated Unit

A GU with one internal state has a four-dimensional parameter space with $\boldsymbol{\theta} = [w_{\mathrm{c}}, b_{\mathrm{c}}, w_{\mathrm{z}}, b_{\mathrm{z}}] \in \mathbb{R}^4$. To begin with, the bias b_c is set to zero, i.e. $b_c = 0$. In this case $x_{\mathrm{F}} = 0$ is again the only fixed point and $\boldsymbol{J}^{\mathrm{GU}}$ becomes

$$J(0) = (1 - z(0))\, w_{\mathrm{c}} + z(0) \tag{16}$$

From (16) the conditions for the occurence of a fold or period-doubling bifurcation can be derived. A fold bifurcation occurs if

$$w_{\mathrm{c}} = \frac{1 - z\,(0)}{1 - z\,(0)} = 1,$$

which is identical to the Elman-RNN. However, the condition for a period-doubling bifurcation becomes:

$$w_{\mathrm{c}} = -\frac{1 + z\,(0)}{1 - z\,(0)}. \tag{17}$$

Since $z\,(0) = \sigma\,(b_{\mathrm{z}})$ the occurence of a period doubling bifurcation depends on the bias b_{z} of the forget gate. This condition states, that for increasing values of b_{z} the recurrent weight w_{c} must assume exponentially increasing (negative) values in order for a period-doubling bifurcation to occur, i.e. the bifurcation boundary is shifted to the left in the parameter space. This is aided by the fact, that it is common practice to initialize the bias of the forget gate with large positive values, e.g. 1 or 2 [12, 13]. This places the bifurcation at $\boldsymbol{w}_{\mathrm{c}} \in [-7, -15]$, which is so far to the left it is highly unlikely that it will have any effect on the training process. This initialization procedure is supposed to help learn *long-term dependencies* by letting the internal state decay as slow as possible (which corresponds to a slow vanishing of the gradient). In fact, this practice increases the domain in the parameter space that corresponds to a globally asymptotically stable dynamic behavior while simultaneously decreasing the domains that result in qualitatively different dynamic behaviour or at least pushing them outside the relevant parameter domain. This is desirable not only because bifurcations can be associated with large gradients, but also because only very few real-world systems have limit cycles, which is the dynamic behaviour exhibited by the RNN/GU after crossing the period-doubling bifurcation boundary.

In the general case, i.e. $b_{\mathrm{z}} \neq 0$, the origin is not a fixed point. In order to enable a comparison between the bifurcation boundaries of the 1d Elman-RNN and the 1d GU, the forget gate weight was fixed to $w_{\mathrm{z}} = 1$ and the forget gate bias was set to $b_{\mathrm{z}} \in \{-10, -1, 0\}$. With these values fixed (14) was solved numerically for x_{F}. By calculating $J^{\mathrm{GU}}(x_{\mathrm{F}})$ the stability of the fixed point can be evaluated. This was done for all possible parameterizations $w_{\mathrm{c}}, b_{\mathrm{c}} \in [-2, 2]$ with an increment of 0.001. Fig. 10, which shows how the bifurcation boundaries of the GU are first identical to the Elman-RNN (see Fig. 3). For increasing activations of the forget gate (realized through the increasing bias b_{c}) the period-doubling bifurcation boundary is pushed to the left. For $b_{\mathrm{z}} = 0$ the bifurcation boundary is already outside of the considered domain of the parameter space.

C. 2d Gated Unit

The parameter space of a GU with two internal states has twelve dimensions with $\boldsymbol{\theta} = [\boldsymbol{W}_{\mathrm{c}}, \boldsymbol{b}_{\mathrm{c}}, \boldsymbol{W}_{\mathrm{z}}, \boldsymbol{b}_{\mathrm{z}}]$, $\boldsymbol{W}_{\mathrm{c}}, \boldsymbol{W}_{\mathrm{z}} \in \mathbb{R}^{2\times 2}$, $\boldsymbol{b}_{\mathrm{c}}, \boldsymbol{b}_{\mathrm{z}} \in \mathbb{R}^2$. In order to enable a direct comparison with the results on the Elman-RNN, $\boldsymbol{b}_{\mathrm{c}}$ is set to zero, $\boldsymbol{b}_{\mathrm{c}} = \boldsymbol{0}$,

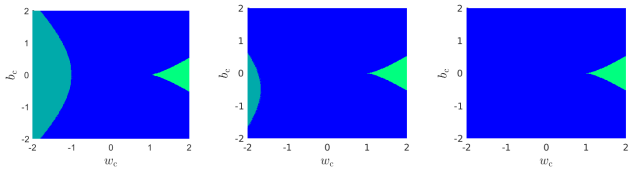


Fig. 10: Regions of qualitatively different behaviour of the 1d autonomous GU with $b_{\mathrm{c}} = 0$.

which again renders the origin a fixed point, i.e. $\boldsymbol{x}_{\mathrm{F}} = \boldsymbol{0}$. This simplifies (15) considerably:

$$\boldsymbol{J}^{\mathrm{GU}}(\boldsymbol{x}_{\mathrm{F}} = \boldsymbol{0}) = \mathrm{diag}(\frac{1}{2}\boldsymbol{I}) + \boldsymbol{W}_{\mathrm{c}} + \boldsymbol{Z}(\boldsymbol{0})\boldsymbol{W}_{\mathrm{c}}. \tag{18}$$

Calculating the eigenvalues of this expression yields

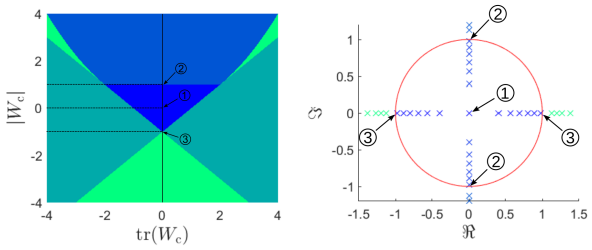
$$\lambda_{1,2} = \frac{\mathrm{tr}((\boldsymbol{I} - \boldsymbol{Z}(\boldsymbol{0}))\boldsymbol{W}_{\mathrm{c}} + \boldsymbol{Z}(\boldsymbol{0}))}{2} \pm \sqrt{\frac{\mathrm{tr}((\boldsymbol{I} - \boldsymbol{Z}(\boldsymbol{0}))\boldsymbol{W}_{\mathrm{c}} + \boldsymbol{Z}(\boldsymbol{0}))^2}{4} - |\boldsymbol{Z}(\boldsymbol{0}) + (\boldsymbol{I} - \boldsymbol{Z}(\boldsymbol{0}))\boldsymbol{W}_{\mathrm{c}}|}. \tag{19}$$

From the equation above two conclusion can be drawn:

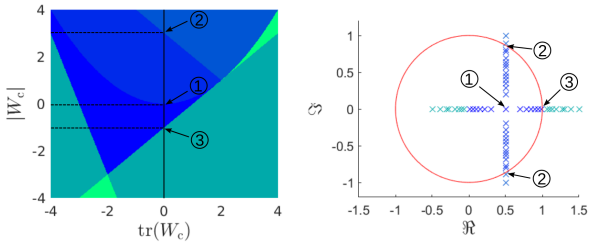
- The eigenvalues and therefore the bifurcation boundaries of the Jacobians of Elman-RNN (11) and GRU (19) are identical, if $\boldsymbol{Z}(\boldsymbol{0}) \to \boldsymbol{0}$.
- Only the bias $\boldsymbol{b}_{\mathrm{z}}$ of the forget gate influences the location of the eigenvalues through $\boldsymbol{Z}(\boldsymbol{0})$ and therefore the stability of the fixed point $\boldsymbol{x}_{\mathrm{F}} = \boldsymbol{0}$.

In order to evaluate the influence of the forget gate on the bifurcation-boundaries, $\boldsymbol{Z}(\boldsymbol{0})$ (which depends only on $\boldsymbol{b}_{\mathrm{z}}$ in this case) was set to $\boldsymbol{Z}(\boldsymbol{0}) \in \{0{\cdot}\boldsymbol{I}, 0.5{\cdot}\boldsymbol{I}, 0.7{\cdot}\boldsymbol{I}, 1{\cdot}\boldsymbol{I}\}$. Then the eigenvalues of $\boldsymbol{J}^{\mathrm{GU}}(\boldsymbol{0})$ can be expressed in terms of the trace and determinant of the weight matrix $\boldsymbol{W}_{\mathrm{c}}$. By evaluating (19) at any point in the $[\mathrm{tr}(\boldsymbol{W}_{\mathrm{c}}), |\boldsymbol{W}_{\mathrm{c}}|]$-space the stability of the fixed-point $\boldsymbol{x}_{\mathrm{F}} = \boldsymbol{0}$ and therefore the dynamic behavior of the GU can be determined. Fig. 11 maps out the $[\mathrm{tr}(\boldsymbol{W}_{\mathrm{c}}), |\boldsymbol{W}_{\mathrm{c}}|]$-space by corresponding dynamic behavior. From Fig. 11 it can be clearly seen, that for increasing values of $\boldsymbol{b}_{\mathrm{z}}$ the region in the parameter space that produces a globally asymptotic stable system increases significantly in size. This in effect pushes the regions corresponding to qualitatively different dynamics so far outside the parameter space, that they will likely never be visited. This phenomenon explains, why Gated Units seem to be less initialization dependent and easier to train than Elman-RNN: It is very likely to initialize a Gated Unit in the parameter space corresponding to globally asymptotic stable dynamics and very unlikely to cross bifurcation boundaries during optimization, since they lie in the outskirts of the parameter space. From a dynamic viewpoint this means also, that Gated Units are a simpler model approach than the Elman-RNN, since they are practically incapable to represent the more complicated dynamics of the Elman-RNN.

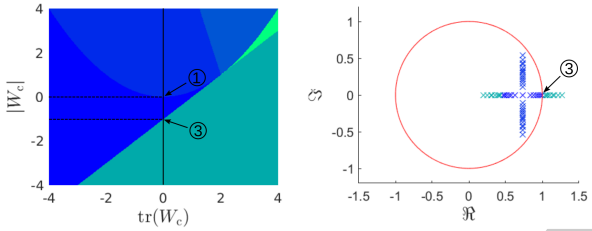
In the general case, i.e. $\boldsymbol{b}_{\mathrm{c}} \neq \boldsymbol{0}$, the fixed point in the origin is shifted, which affects the eigenvalues of $\boldsymbol{J}^{\mathrm{GU}}$ and therefore the values of $\boldsymbol{W}_{\mathrm{c}}$ at which bifurcations occur. Again, a compact representation of the bifurcation boundaries in terms of the trace and determinant of $\boldsymbol{W}_{\mathrm{c}}$ is not possible. Instead, the



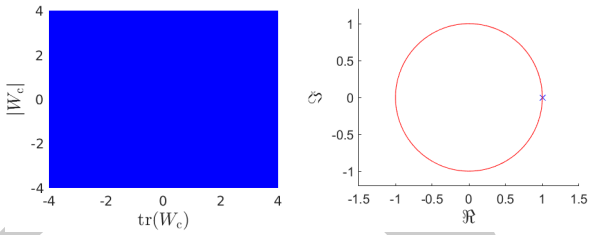
(a) $\lim_{b_z \to -\infty} \boldsymbol{Z}(\mathbf{0}) = 0 \cdot \boldsymbol{I}$



(b) $\boldsymbol{Z}(\mathbf{0}; \boldsymbol{b}_z = \mathbf{0}) = 0.5 \cdot \boldsymbol{I}$



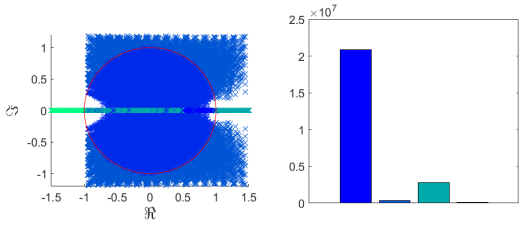
(c) $\boldsymbol{Z}(\mathbf{0}; \boldsymbol{b}_z \approx \mathbf{1}) = 0.75 \cdot \boldsymbol{I}$



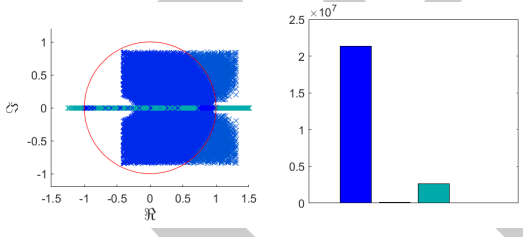
(d) $\lim_{b_z \to \infty} \boldsymbol{Z}(\mathbf{0}) = 1 \cdot \boldsymbol{I}$

Fig. 11: Left: Regions of qualitatively different behaviour of the 2d autonomous GU with $\boldsymbol{b}_c = \mathbf{0}$. Right: Eigenvalues of $\boldsymbol{J}^{\mathrm{GU}}$ depending on $|\boldsymbol{W}_c|$ for $\mathrm{tr}(\boldsymbol{W}_c) = 0$.

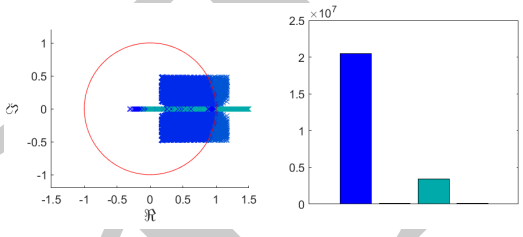
forget gate weight matrix was set to $\boldsymbol{W}_z = \boldsymbol{I}$ and the forget gate bias was set to $\boldsymbol{b}_z \in \{-\mathbf{1}, \mathbf{0}, \mathbf{1}\}$. With these parameters fixed, a complete enumeration of all possible combinations of the remaining parameters $[\boldsymbol{W}_c, \boldsymbol{b}_c]$ was performed. All parameters were varied in the range $[-1.6, 1.6]$ with an increment of 0.2 yielding a total of approximately $25 \cdot 10^6$ different parameterizations. For each parameterization the fixed point and corresponding eigenvalues $\lambda_{1,2}$ of $\boldsymbol{J}^{\mathrm{GU}}(\boldsymbol{x}_F)$ were calculated and plotted in the complex plane in order to evaluate the stability of the fixed point. The eingenvalues in the complex plane and a histogram of the occurences of the respective



(a) $\boldsymbol{b}_z = [-1, -1]^T$



(b) $\boldsymbol{b}_z = [0, 0]^T$



(c) $\boldsymbol{b}_z = [1, 1]^T$

Fig. 12: Regions of qualitatively different behaviour of the 2d autonomous GU with $\boldsymbol{b}_c \neq \mathbf{0}$. Right: Histogram of parameterizations resulting in respective dynamic behavior.

dynamics are given in Fig. 12.

Fig. 12 shows that only about 12% of the considered parameterizations lie outside of the globally asymptotically stable domain. This is a drastic reduction compared to the Elman-RNN, see Fig. 7 and supports and further generalizes the previous findings made under the assumption $\boldsymbol{b}_c = \mathbf{0}$.

VI. Implications for Initialization

The purpose of these investigations was to map out the most relevant region the parameter space of Recurrent Neural Networks as detailed as possible. Therefore the parameter space was sampled equidistantly throughout all experiments. In practice, the choice of initialization procedure affects how likely it is to land in a region associated with a certain dynamic behavior. Fig. 13 shows determinant and trace of $\boldsymbol{W}_c$ drawn from a standard normal distribution (10^6 samples) in a heat map. About 66 % of the sampled matrices would have rendered an Elman-RNN ($\boldsymbol{b}_c = \mathbf{0}$) outside of the globally asymptotically stable domain. Although a comprehensive investigation regarding the effect of different initialization procedures was not part of this work, it might already disclose a different approach to initializing Recurrent Neural Networks. Instead of aiming at initializations that keep the gradient constant, the goal could be to initialize the RNN in the domain

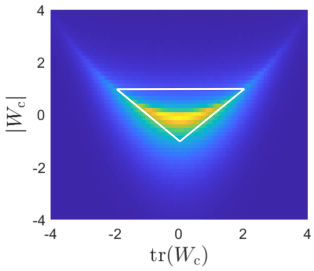


Fig. 13: Heat map of occurences of determinant and trace of 10^7 randomly sampled ($\boldsymbol{W}_c$) drawn from a standard normal distribution. The white triangle marks the region of globally asymptotically stable Elman-RNNs, compare Fig. 6.

of the parameter space that produces the same dynamics as the system to be modeled.

VII. Conclusions

A comprehensive analysis of codimension-1 bifurcations in Elman-RNNs and forget gate only Gated Units was conducted. It was shown, that large domains in the parameter space of the Elman-RNN correspond to very complicated nonlinear dynamic behavior such as limit cycles, that are very rare in real-world systems. Bifurcation boundaries, that mark the transitions between domains of qualitatively different dynamic behavior, are located at the very center of the Elman-RNNs parameter space. These two drawbacks of the Elman-RNN, i.e. large domains of the parameter space produce dynamic behavior that is likely to be irrelevant for the modeling-task at hand and these regions are seperated by very large local gradients, are remedied by Gated Units such as the LSTM and GRU. It is shown, that the forget gate pushes the bifurcation boundaries and thereby the parameterizations that produce the rather unwanted dynamics to the periperal regions of the parameter space. This in turn almost guarantees that a Gated Unit is a globally asymptotically stable system and makes it very unlikely for the optimizer to encounter bifurcation boundaries during gradient-based optimization. The findings in this paper suggest, that Gated Units are not easier to train than Elman-RNNs, because they solve the vanishing gradient problem. Instead, they are less sensitive to initialization, because almost their entire parameter space produces a single dynamic behavior. The existence of mainly one dynamic behavior also means, that there are no infinitely high local gradients, i.e. bifurcation boundaries, in the parameter space that could disturb the training process. Lastly, Gated Units are a great general model candidate because the one dynamic behaviour that they do represent, is the one that most real-world systems exhibit.

Acknowledgment

Funded by the federal state of Hesse and the European Regional Development Fund (ERDF 2014-2020), Project: Digital Twin of Injection Molding (DIM) FKZ: 0107/20007409

References

[1] I. D. Jordan, P. A. Sokol, and I. M. Park. “Gated recurrent units viewed through the lens of continuous time dynamical systems”. In: *arXiv preprint arXiv:1906.01005* (2019).

[2] A. Rehmer and A. Kroll. “On Using Gated Recurrent Units for Nonlinear System Identification”. In: *Preprints of the 18th European Control Conference (ECC)*. IFAC. Naples, Italy, 2019, pp. 2504–2509.

[3] A. Rehmer and A. Kroll. “On the vanishing and exploding gradient problem in Gated Recurrent Units”. In: *21st IFAC World Congress*. Berlin, Germany, 2020.

[4] K. Doya. “Bifurcations of Recurrent Neural Networks in Gradient Descent Learning”. In: *IEEE Transactions on Neural Networks* 1 (1993), pp. 75–80.

[5] R. Pascanu, T. Mikolov, and Y. Bengio. “On the difficulty of training recurrent neural networks”. In: *International conference on machine learning*. 2013, pp. 1310–1318.

[6] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[7] K. Cho, B. v. Merrienboer, and C. Gulcehre. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, 1724–1734.

[8] J. v. d. Westhuizen and J. Lasenby. “The unreasonable effectiveness of the forget gate”. In: *CoRR* abs/1804.04849 (2018).

[9] R. Haschke and J. Steil. “Input space bifurcation manifolds of recurrent neural networks”. In: *Neurocomputing* 64 (2005), pp. 25–38.

[10] R. Marichal, J.D. Piñeiro, and E. González. “Study of fold bifurcation in a discrete recurrent neural network”. In: *Proceedings of the World Congress on Engineering and Computer Science*. Vol. 2. 2009, pp. 995–1000.

[11] K. Greff et al. “LSTM: A Search Space Odyssey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10 (2017), 2222–2232. ISSN: 2162-2388.

[12] F. A. Gers, J. Schmidhuber, and F. Cummins. “Learning to forget: Continual prediction with LSTM”. In: *Neural computation* 12.10 (2000), pp. 2451–2471.

[13] R. Jozefowicz, W. Zaremba, and I. Sutskever. “An empirical exploration of recurrent network architectures”. In: *International conference on machine learning*. PMLR. 2015, pp. 2342–2350.