Adjoint Dynamics of Stable Limit Cycle Neural Networks

Piotr A. Sokół, Ian Jordan, Eben Kadile, Il Memming Park*
Department of Neurobiology and Behavior
Stony Brook University, NY, USA
{piotr.sokol, ian.jordan, eben.kadile, memming.park}@stonybrook.edu

Abstract

Exploding and vanishing gradient are both major problems often faced when an artificial neural network is trained with gradient descent. Inspired by the ubiquity and robustness of nonlinear oscillations in biological neural systems, we investigate the properties of their artificial counterpart, the stable limit cycle neural networks. Using a continuous time dynamical system interpretation of neural networks and backpropagation, we show that stable limit cycle neural networks have non-exploding gradients, and at least one effective non-vanishing gradient dimension. We conjecture that limit cycles can support the learning of long temporal dependence in both biological and artificial neural networks.

1. Introduction

Due to the long, cascaded function compositions of the forward computation in artificial neural networks, the gradient signal often loses information as it is propagated backwards through the network. This phenomenon, known as the vanishing and exploding gradient problem [1, 2], is exacerbated for deep feedforward neural networks (FNNs) and recurrent neural networks (RNNs). Additionally, forward computation of FNN/RNNs are typically implemented with stable dynamics, which leads to perturbations being forgotten after a short period of time (or layers of network) [3].

1.1. Proposed solutions to the vanishing and exploding gradient problem

Many approaches, notably gradient clipping [2], batch normalization [4], and special activation functions such as ReLU, have been proposed to alleviate the exploding gradient problem. At the same time, architectures such as ResNet [5] and Neural ODE [6] and tailored recurrent units such as LSTM [7, 8] and GRU [9] have been used to tame the vanishing gradient problem. More recently, recurrent neural networks with special norm-preserving weight constraints have also been proposed [10, 11]. However, these approaches do not completely ensure the numerical and dynamical stability of the backpropagating gradients.

1.2. Fine tuning problem in neural systems

A recurring idea for taming the vanishing gradient problem is to mimic a continuous attractor that does not forget. For example, LSTM without the forgetting gate stores information in its cell state that implements a continuous attractor, and the linear part of the unitary RNNs preserves the magnitude of the state vector. Similarly, many population dynamics models in theoretical neuroscience, such as models of working memory are designed to have long temporal memory. But unfortunately, the required fine tuning of parameters makes neural networks that implement continuous attractor using inherently nonlinear biophysical neurons brittle [12, 13].

On the other hand, (nonlinear) oscillations can be found throughout neural systems at multiple temporal and spatial scales. Single neurons can show oscillations of multiple time scales and recurrent connections with delays and time constants generate oscillations in networks of a few neurons to large-scale oscillations detectable in the field potentials. This ubiquity of oscillation suggests that it is a robust dynamical phenomena that does not require fine tuning. In this paper, we argue that these spontaneous nonlinear oscillation dynamics may provide a mechanism for long temporal network state memory.

2. Background

2.1. Continuous-time neural networks

An n-dimensional RNN or an FNN with constant width n can be written as,

$$\mathbf{x}_{t+1} = \tilde{f}(\mathbf{x}_t, \theta) \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ denotes the state vector (for RNN) or the activations (for FNN), and θ denotes the parameters of

*This work was supported by NSF IIS-1734910 and Institute for Advanced Computational Sciences (IACS) Jr. Researcher Award.

the network. For the FNN, $\mathbf{x}_0$ corresponds to the input of the network, and $\mathbf{x}_T$ corresponds to the output where T corresponds to the number of layers. If we instead take $\mathbf{x}$ to be a function of a continuous time-variable, then the above system is the Euler approximation of the following ordinary differential equation (ODE) with step-size 1 [14, 6]:

$$\dot{\mathbf{x}} = \tilde{f}(\mathbf{x}(t), \theta) - \mathbf{x}(t) =: f(\mathbf{x}(t), \theta) \tag{2}$$

Let $\mathcal{D}f(\mathbf{x}, \theta)$ be the Jacobian of f evaluated at $\mathbf{x}$ and θ, and $\phi(t)$ be a solution to equation (2) on the time interval $[0, T]$. The time evolution of a perturbation δ of the system at time 0 defines the **forward sensitivity** of $\mathbf{x}$ corresponding to the trajectory ϕ:

$$\dot{\delta} = \mathcal{D}f(\phi(t), \theta)\delta \tag{3}$$

The forward sensitivity will prove essential to our study of continuous-time backpropagation.

2.2. Adjoint dynamics and backpropagation

To study the backpropagating gradient, we define the **adjoint system** of the trajectory ϕ as,

$$\dot{\psi} = -\mathcal{D}f(\phi(t), \theta)^\top \psi. \tag{4}$$

Note that both (3) and (4) are time varying linear dynamical systems. In [15], it was shown that, for all $t \in [0, T]$, the scalar $\psi(t)^\top \delta(t)$ remains constant. This reveals that $\mathbf{x}$ and ψ jointly follow Hamiltonian dynamics, where the Hamiltonian is given by $\mathcal{H} = \psi^\top f(\mathbf{x}, \theta)$.

Importantly, the adjoint dynamics strongly relates to the backpropagating gradient. Suppose we have a cost function $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$, which we evaluate on the terminal point $\phi(T)$, then we may compute the derivative of $\mathcal{L}(\phi(T))$ with respect to the initial condition $\phi(0)$ using the adjoint system. More precisely, if we have the initial condition $\phi(0) = \alpha$ and we obtain the terminal condition $\beta = \phi(T)$ by solving (2), we may compute $\nabla_\alpha \mathcal{L}(\beta)$ by setting the terminal condition $\psi(T) = \nabla_{\mathbf{x}} \mathcal{L}(\beta)$, where $\nabla_{\mathbf{x}}$ denotes the standard gradient, and then integrating *backwards in time* to obtain $\psi(0)$. This integration of the adjoint system backwards in time is the continuous time analog of backpropagating the gradient of an error function through a network. In general,

$$\frac{d\mathcal{L}}{d\theta} = \psi^\top(0)\frac{\partial f(\phi(0))}{\partial \theta} + \int_0^T \psi^\top(t)\frac{\partial f(\phi(t))}{\partial \theta} dt \tag{5}$$

where $\psi(T) = \partial C / \partial \phi(T)$. Hence, if the adjoint integrated backwards in time vanishes or explodes, so does the gradient.

2.3. Forward sensitivity and adjoint dynamics

As hinted above, the dynamics of the forward sensitivity and backward integrated adjoint dynamics behave equivalently. First consider the n-dimensional, inhomogeneous, time varying, linear system

$$\dot{\mathbf{z}} = A(t)\mathbf{z} + \mathbf{w}(t), \quad \mathbf{z}(0) = \mathbf{c} \tag{6}$$

where $A(t)$ is bounded. A classic result shows that the solution of 6 can be written given the solutions of the corresponding homogeneous system [16].

Lemma 1 (Solution of a linear inhomogeneous system)**.** *For a system of the form* (6)*, with a matrix differential equation describing the homogeneous dynamics*

$$\dot{Y} = A(t)\, Y \tag{7}$$

the solution of the inhomogeneous dynamics is given by $\mathbf{z}(t) = Y(t)\,\mathbf{c} + \int_0^t Y(t)\, Y^{-1}(\tau)\, w(\tau) d\tau$.

Note that $Y(t)$ is non-singular, allowing us to study its matrix inverse. Using $d(Y^{-1}) = -Y^{-1}(dY)Y^{-1}$, denoting $\Psi(t) = \left(Y^{-1}\right)^\top$ we have,

$$\dot{\Psi} = -A^\top(t)\,\Psi \tag{8}$$

which corresponds to (4). Any product of the form $Y(t)Y^{-1}(\tau)$ can be equivalently be expressed as $\left(\Psi(\tau)\Psi(t)^{-1}\right)^\top$. Furthermore, we can rewrite the solution to equation (6) in the following equivalent form:

$$\mathbf{z}(t) = Y(t)\,\mathbf{c} + \int_0^t Y(t)\,\Psi^\top(\tau)\, w(\tau) d\tau \tag{9}$$

This reveals that Ψ and Y have equivalent dynamics. Running $Y^{-1}(t)$ forward in time is the same as running $Y(t)$ backwards in time. We now identify the two linear time-varying systems with forward sensitivity and the adjoint systems, which allows us to state the following:

Proposition 2. *Given the dynamical system* $\dot{\mathbf{x}} = f(\mathbf{x}, \theta)$*, for any trajectory, the dynamics of the forward sensitivity,* $\delta(t)$*, are equivalent to the dynamics of the adjoint running backwards in time,* $\psi(-t)$*.*

3. Result

Using the adjoint we can express the gradient of the loss function with respect to the parameters as defined in (5). As noted in [2], the difficulty in optimizing this object primarily stems from computing the adjoint (as a continuous-time analogue for the Jacobian matrices). Therefore we investigate the applicability of using limit cycle initializations to circumvent adjoint instability.

From proposition 2, it is easy to verify the conclusions of [1] that stable fixed points induce vanishing gradients, since the time-reversed dynamics of the adjoint is also stable. The converse also holds for expanding dynamics and exploding gradients. Although the continuous-time versions of many RNNs, e.g. tanh RNNs and GRU-RNN [14], are ultimately bounded in

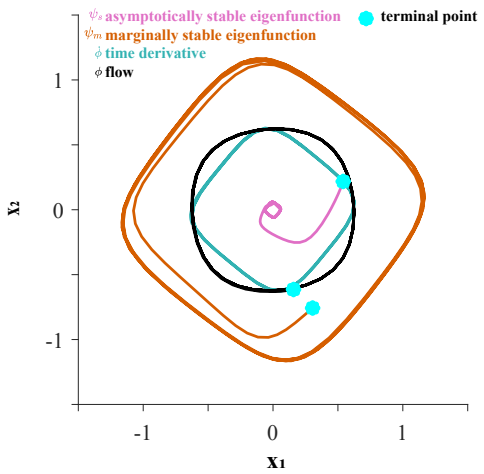

Figure 1: Phase portrait and adjoint state trajectories of a 2D limit cycle RNN. The flow ϕ of $\dot{\mathbf{x}}$ along with its time derivative $\dot{\phi}$, and the two eigenfunctions of the adjoint Ψ.

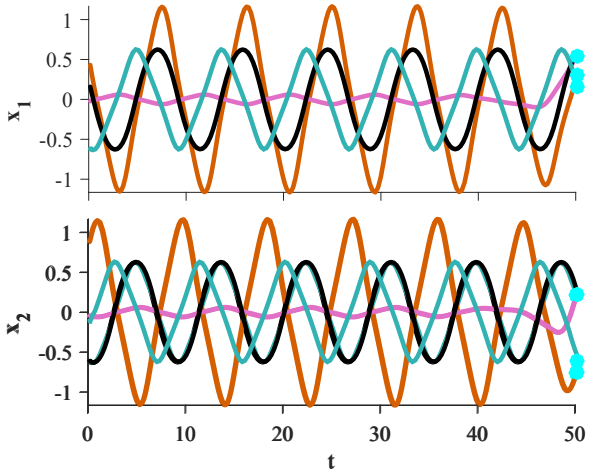

Figure 2: Time series of the trajectories presented in Fig. 1. Recall that the adjoint dynamics ψ are solved backward in time.

the state space, the corresponding time-reversed adjoint dynamics can be unbounded. However, if the dynamics of forward computation forms a stable limit cycle, we can show that the time-reversed adjoint dynamics is nontrivially periodic (hence bounded).

Extensions of Lyapunov's direct method can be used to show that the stability of limit cycles depends on parts of the spectrum of the forward sensitivity [17]. Unlike in the analysis of equilibria of differential equations, the eigenvalues that determine the stability of the periodic orbit are the ones associated with eigenvectors lying in a plane traverse to the flow. To see this, let us define the mondromy matrix C of a T-periodic limit cycle, $\phi(t)$ as [16]:

$$C := \int_{t_0}^{t_0+T} \mathcal{D}f(\phi(\tau))d\tau \tag{10}$$

and let $\Delta(t)$ be the fundamental matrix solution to the forward sensitivity, with $\Delta(t_0) = \mathbb{I}$. Then Δ satisfies $\Delta(t_0+T) = C\Delta(t_0)$. Importantly, we also have that the second time derivative of the flow of (2) satisfies

$$\ddot{\phi}(t) = \mathcal{D}f(\phi(t))\dot{\phi}(t). \tag{11}$$

Thus the function $\dot{\phi}(t)$ is a trajectory of the forward sensitivity dynamics (3). The fact that $\dot{\phi}(t)$ is T-periodic and the identity from (11) allow us to write

$$\dot{\phi}(t) = \dot{\phi}(t+T) = C\dot{\phi}(t)$$

which in turn implies that $\dot{\phi}(t)$ is an eigenfunction of C with eigenvalue 1. We can now state our two main theorems.

Theorem 3 (Andronov-Witt [17])**.** *Let $\phi(t)$ be a non-trivial, periodic solution of* (2) *with period T. If the eigenvalue of C which is equal to unity, has algebraic multiplicity one and if the absolute values of all remaining characteristic multipliers of C are less than unity, then the solution $\phi(t)$ is Lyapunov stable.*

Moreover, for hyperbolic systems Theorem 3 defines an equivalent condition on the stability of the forward sensitivity.

Theorem 4. *If the periodic trajectory associated with a forward-stable limit cycle is hyperbolic, then the periodic, non-autonomous system defining the forward sensitivity* (3) *is bounded-input bounded-output stable. Moreover, the flow of the forward sensitivity φ has one marginally stable [18] eigenfunction which is the time derivative $\dot{\phi}(t)$ of the flow of the original dynamics $\dot{\mathbf{x}} = f(\mathbf{x})$. The remaining $n-1$ eigenfunctions are asymptotically stable.*

Consider a globally stable limit cycle neural network such that without any input the state evolution always converges to a limit cycle. Near the limit cycle, the neural network is governed by Theorem 4 which implies that the backpropagating gradient converges to a periodic orbit. Hence, it does not vanish nor explode.

As a demonstration, we take a vanilla 2-dimensional tanh-RNN that exhibits a globally attracting limit cycle [14, 19]. In Fig. 1, the black trajectory in the state space represents the stable limit cycle. Its derivative and corresponding adjoint dynamics have same asymptotic behavior: oscillating periodically as seen in Fig. 2. If the adjoint is initialized on the asymptotically stable manifold, it quickly decays to the origin (magenta in Figs). Although the adjoint is non-zero in both cardinal dimensions, interestingly, effectively only

one dimension (associated with the marginally stable eigenfunction) conveys the gradient.

One practical usage could be to use them as initialization, akin to the ideas of critical or orthogonal initialization [20]. To construct a larger dimensional system, one can take a direct sum of independent 2-D stable limit cycle neural networks. If the problem has long temporal dependence, the phase of nonlinear oscillators can retain the information over much longer (theoretically infinite) time interval. However, during training, the network may quickly bifurcate out of stable limit cycle behavior [21].

4. Discussion

Inspired by neural oscillations, we have proposed stable limit cycle neural network as a new component in designing FNN/RNN systems, for instance, but not limited to, as an initialization scheme. This adds to the prior research on phasor neural network [22].

One brain region where such putative computation may occur is the olivo-cerebellar loop. This system is known to play a strong role in motor learning, as well as more abstract temporal sequence learning tasks. Moreover, the inferior olivary neurons exhibit non-linear oscillations. There are a natural abundance of tasks with periodicity and long temporal dependence, including motor behaviors, such as walking, as well as tasks requiring well-timed responses. We conjecture that stable limit cycle dynamics could allow for temporal learning in the cerebellum.

References

[1] Yoshu Bengio, Patrice Simard, and Paolo Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, 1994.

[2] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, Feb. 2013, pp. 1310–1318.

[3] Wolfgang Maass and Eduardo D Sontag, "Neural systems as nonlinear filters," *Neural Comput.*, vol. 12, no. 8, pp. 1743–1772, Aug. 2000.

[4] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems 31*, S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, Eds., pp. 6571–6583. Curran Associates, Inc., 2018.

[7] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[8] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins, "Learning to forget: continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[9] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN Encoder-Decoder for statistical machine translation," June 2014.

[10] Martin Arjovsky, Amar Shah, and Yoshua Bengio, "Unitary evolution recurrent neural networks," Nov. 2015.

[11] Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, and Chris Pal, "On orthogonality and learning recurrent networks with long term dependencies," Jan. 2017.

[12] David MacNeil and Chris Eliasmith, "Fine-tuning and the stability of recurrent neural networks," *PLoS One*, vol. 6, no. 9, pp. e22885, Sept. 2011.

[13] Joel Zylberberg and Ben W Strowbridge, "Mechanisms of persistent activity in cortical circuits: Possible neural substrates for working memory," *Annu. Rev. Neurosci.*, vol. 40, pp. 603–627, July 2017.

[14] Ian D Jordan, Piotr Aleksander Sokol, and Il Memming Park, "Gated recurrent units viewed through the lens of continuous time dynamical systems," June 2019.

[15] Lev Semenovich Pontryagin, Vladimir Grigorevich Boltyanskii, Revaz Valerianovich Gamkrelidze, and Evgenii Frolovich Mishchenko, *Mathematical theory of optimal processes*, Macmillan company, 1964.

[16] Carmen Chicone, *Ordinary Differential Equations with Applications*, Springer Science & Business Media, Sept. 2006.

[17] Lev Semenovich Pontryagin, *Ordinary Differential Equations: Adiwes International Series in Mathematics*, Pergamon, May 2014.

[18] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini, *Feedback Control of Dynamic Systems*, Pearson, Boston, 7 edition edition, May 2014.

[19] Randall D Beer, "On the dynamics of small Continuous-Time recurrent neural networks," *Adapt. Behav.*, vol. 3, no. 4, pp. 469–509, Mar. 1995.

[20] Piotr Sokol and Il Memming Park, "Information geometry of orthogonal initializations and training," (under review), Nov. 2019.

[21] Kenji Doya, "Bifurcations in the learning of recurrent neural networks," in *[Proceedings] 1992 IEEE International Symposium on Circuits and Systems*, May 1992, vol. 6, pp. 2777–2780.

[22] André J Noest, "Phasor neural networks," in *Neural Information Processing Systems*, D Z Anderson, Ed. 1988, pp. 584–591, American Institute of Physics.