ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Self-organized dynamic attractors in recurrent neural networks

Benedikt Vettelschoss, Matthias Freiberger, Joni Dambre

Ghent University - imec - IDLab
Technologiepark-Zwijnaarde 126, B-9052 Ghent, Belgium

Abstract. Recurrent neural networks usually rely on either transient or attractor dynamics to implement working memory, and some studies suggest that it requires a combination of the two. These studies introduce attractor states by the supervised training of a network's feedback weights. In this work we report the creation of comparable memory states through unsupervised learning. We introduce attractor dynamics into an echo state network in a self-organized way by applying a differential Hebbian rule to it's feedback weights. We find that this yields periodic and quasiperiodic attractors in most cases. We analyse the linearized system after the learning phase to understand the origin of these attractors, and connect these findings to other results concerning the dynamical changes induced by neural plasticity.

1 Introduction

Working memory (WM) is the ability to "transiently hold and manipulate goal-related information to guide forthcoming actions" [1]. Thus, in contrast to short-term memory, it does not mean simple short-term storage, but implies the ability to incorporate information on the fly in neural computations. These two aspects have given rise to two principles thought to implement WM in artificial neural networks: *attractor* and *transient* dynamics [1, 2, 3].

Attractor memory mechanisms encode information by increasing the activity of a subset of neurons. Information is usually stored by introducing a fixed point attractor into the network state space. Upon presentation of the corresponding stimulus, the network state is pushed to its basin of attraction and then converges to a fixed pattern. This mechanism provides means to store information for a biologically realistic amount of time and is supported by several theoretical and experimental studies [2]. However, it takes a long time for the network state to actually reach an attractor. Furthermore, even when the stable state is reached, fixed points provide no computationally useful dynamics other than the state the network settles into, given the initial conditions [3].

Models which implement WM through transient dynamics have been introduced under the umbrella term *reservoir computing (RC)* in the context of machine learning [4] and computational neuroscience [5]. A reservoir computer is composed of a nonlinear dynamical system (the reservoir) and a set of input and output weights. The input weight matrix is used to project an input to the reservoir and, as the reservoir itself, remains unchanged. Learning is restricted to the output weight matrix that is used to linearly combine a target signal from the

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

reservoir states. The dynamics of the reservoir are used to provide sufficiently diverse nonlinear transformations of the input. A suitable reservoir is one that has only a single globally attracting fixed point in its state space. This property is known as the *echo state property* (ESP) [4]. Ensuring the ESP implies that the performance of a reservoir computer is weak on tasks that require persistent memory. However, persistent memory can be attained by training additional units that provide feedback to the reservoir [6, 7, 8]. With the added feedback loops, the combined system no longer obeys the ESP and fixed point attractors are introduced into the reservoir state space by means of supervised learning.

From a biological point of view, it seems plausible that the mammalian brain indeed makes use of attractor and transient dynamics. However, several studies show that biological neural networks display complex (quasi-)periodic and chaotic dynamics [9] and do not rely solely on fixed point attractors. A plausible scheme for the development of such complex memory states is *self-organization*. Self-organization, a systems property of "[...]being strictly determinate in its actions and yet demonstrate a self-induced change in organization" [10], excludes supervised learning methods.

In this contribution we report the self-organized emergence of a complex attractor by providing feedback to a random recurrent neural network. In the spirit of reservoir computing, the RNN serves here as a placeholder for an unmodified dynamical system. Hence, the proposed mechanism is potentially applicable to introduce persistent memory states into many of the physical devices that have recently been proposed as candidate technologies for future neuromorphic computing devices under the umbrella term *physical reservoir computing*.

2 Related work

While there are many studies concerning memory mechanisms in RNNs (e.g. [11]), we base our method of attractor generation on the work of Der & Martius. In [12] they propose a rule termed *differential extrinsic plasticity*(DEP) to drive a robotic agent through a series of self-sustained motor behaviours. DEP makes use of an inverse model that amplifies perturbations by estimating the causing motor command. Once the inverse model is trained, its output is correlated with past sensor values with a differential Hebbian rule [13]. For robots whose sensor dynamics follow very closely the applied motor commands, Der & Martius set the inverse model to a one-to-one mapping. Here, we are in a similar situation as we apply their formalism to an RNN receiving only very weak inputs. Thus, we omit the inverse model entirely and arrive at differential Hebbian learning, laid out below. Additional related work has been performed by Ceni et al. [14], who analyse a network of fixed-point attractors extracted from RNNs. Contrary to our approach, this approach is not self-supervised.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

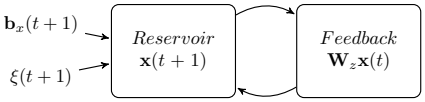

Fig. 1: Schematic illustration of the proposed system.

3 Methods

We consider echo state networks (ESNs) as a reservoir and equip them with feedback connections that are trained using differential Hebbian learning (DHL). ESNs and DHL are quickly reviewed in this section.

3.1 Echo state networks (ESNs) with trainable feedback weights

Echo state networks [4] have been introduced as models for networks of rate-coded neurons with a tanh activation function. In the discussion below, we omit the input and output weights, as we are not combining the reservoir states to form a target signal and the only input that is applied are noise and bias terms that are injected directly into the neurons. Training is restricted to additional feedback weights that project from the states and back into the reservoir. Contrary to most approaches to trained feedback in RC, our feedback weights have the same dimensions as the reservoir weights in order to keep the setting of [12].

The update equation for the system is thus given by

$$\mathbf{x}(t+1) = \tanh\left(\mathbf{W}_x\ \mathbf{x}(t) + \mathbf{W}_z\mathbf{x}(t) + \mathbf{b}_x\right) + \xi(t+1). \tag{1}$$

where $\mathbf{W}_x$ is the $N_x \times N_x$ reservoir weight matrix that is randomly initialized and $\mathbf{W}_z$ are the $N_x \times N_x$ feedback weights initialized to $\mathbf{0}$. $\mathbf{b}_x \sim \mathcal{N}(0, 10^{-3})$ is the vector of N_x biases that are applied to the neurons in each time step. ξ is N_x-dimensional Gaussian noise drawn i.i.d. from $\mathcal{N}(0, 10^{-4})$ at each time-step.

To keep the time it takes for an attractor to emerge low and ease computation we chose a small reservoir size of $N_x = 20$ neurons, drawing $\mathbf{W}_x$ from the unit normal distribution $\mathcal{N}(0, 1)$. We also make the network sparse and randomly set 90% of the weight matrix to 0. To ensure the echo state property, the weights were rescaled such that their spectral radius was $\rho(\mathbf{W}_x) = 0.95$.

3.2 Differential Hebbian learning (DHL)

Differential Hebbian learning [13] correlates the time derivatives of neuronal activations at successive timesteps. Hence,

$$\Delta\mathbf{W}(t+1) = \varepsilon\dot{\mathbf{x}}(t+1)\dot{\mathbf{x}}(t)^T, \tag{2}$$

where ε is a learning rate, $\mathbf{x}(t)$ is the vector of neuronal activations at time t and $\dot{\mathbf{x}}(t)$ are their time deriatives. Here, as we are dealing with a time-discrete system they are the finite differences $\dot{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}(t-1)$. To prevent the feedback

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

weights from growing unboundedly we follow [12] and use weight decay, such that

$$\Delta \mathbf{W}(t+1) = \varepsilon \left(\dot{\mathbf{x}}(t+1)\dot{\mathbf{x}}(t)^T - \mathbf{W}(t) \right) \tag{3}$$

and in addition normalize the weights to have a Frobenius norm of 1:

$$\mathbf{W} \leftarrow \frac{\mathbf{W}}{\|\mathbf{W}\| + 10^{-12}}. \tag{4}$$

4 Results

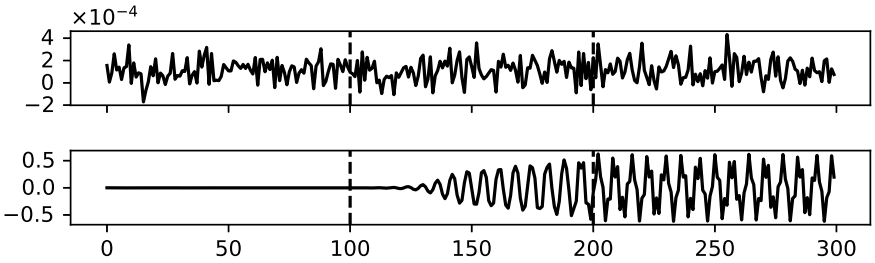

Fig. 2: **Top** Input current to a selected reservoir neuron (noise + bias). **Bottom** Activation of the selected reservoir neuron.

We trained 50 ESNs using the proposed learning rule.Each ESN was run for $T = 10000$ time steps, of which the first 100 were discarded to wash out initial conditions. Adaptation of the feedback weights was switched on until time step 7500. Then the trained system was run for another 2500 timesteps with the input noise still applied and activations were inspected to determine the characteristics of the induced attractor.

Half of the 50 ESNs displayed periodic or quasi-periodic activations after training. 21 showed 2-periodic cycles in which the neuron activations are alternating between -1 and 1. Four networks still displayed fixed point dynamics.

Each attractor can be associated with an eigenvalue of the combined systems dynamics linearized around the origin - the systems Jacobian. For standard ESNs the Jacobian reduces to the reservoir weight matrix $\mathbf{W}_x$. By linearization of the ESN with trained feedback connections we thus obtain the linear system:

$$\mathbf{x}(t+1) = \left[\mathbf{W}_x + \mathbf{W}_z \right] \mathbf{x}(t) + \mathbf{b}_x. \tag{5}$$

Figure 3 shows the eigenspectrum of reservoir and feedback weights, as well as their sum for selected experiments. The spectrum clearly shows characteristics of two types of bifurcations leading to the observed periodic activations: (a) a Neimark-Sacker bifurcation, the discrete version of a Hopf bifurcation, in which a fixed point changes stability and gives rise to a limit cycle via a pair of complex conjugate eigenvalues crossing the unit circle. (b) A period-doubling bifurcation

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

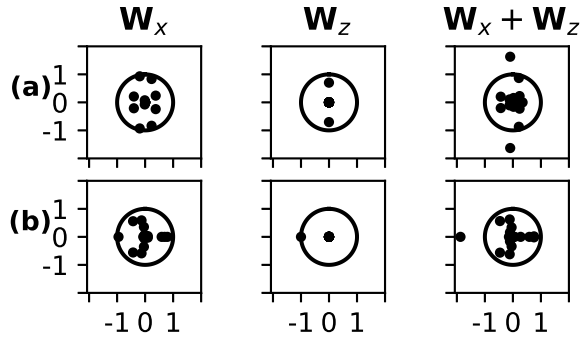

Fig. 3: Eigenvalues of $\mathbf{W}_x$, $\mathbf{W}_z$ and $\mathbf{W}_x + \mathbf{W}_z$ in the complex plane for ESNs with (quasi-)periodic **(a)** and 2-periodic **(b)** dynamics after adaptation of $\mathbf{W}_z$. The observed periods larger than 2 were $3, 4$ and 6.

in which a stable fixed point - a cycle with period one - doubles its period and turns into a 2-periodic orbit [15].

To see how the learning rule introduces the observed dynamics it helps to write the derivatives of neural activations as finite differences in Equation (2). Expanding, we get

$$\Delta\mathbf{W}(t+1) = \varepsilon \left[\mathbf{x}(t+1)\mathbf{x}(t)^T + \mathbf{x}(t)\mathbf{x}(t-1)^T - \mathbf{x}(t)\mathbf{x}(t)^T - \mathbf{x}(t+1)\mathbf{x}(t-1)\right], \tag{6}$$

which we identify as Hebbian terms with different signs. The terms with positive sign are temporally asymmetric, correlating activations at successive timesteps. Learning rules of this kind have been found to initiate a quasiperiodic route to chaos in RNNs[16]. The first term with a negative sign is a classical anti-Hebbian term. These rules are used to decorrelate neural activity by decreasing the connection strengths of simultaneously active neurons [17] and have been linked to criticality in [18]. This combination of terms gives a hint at DHL's destabilizing effect. It increases synaptic strength between successively active neurons while decreasing synaptic strength between those active simultaneously, rendering the two sets largely disjoint. This behaviour likely leads to oscillations which grow until they are stabilized by the tanh. A central role is taken by the normalization, which not only keeps the feedback weights bounded but also scales them up significantly during the early stages of learning. This leads to an immediate loss of the origin's stability. It is remarkable though, that stability is lost via a *single* pair of eigenvalues and not a general scaling of the spectrum as would be the case if we just increased the spectral radius.

5 Conclusion

We applied a variant of the DHL rule proposed by Der & Martius [12] on the feedback weights of ESNs and investigated the resulting system dynamics. We found the rule to result in the emergence of (quasi-)periodic attractors in the

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

majority of cases. These correspond to a bifurcation connected to an eigenvalue of the system's Jacobian, which makes them in a sense natural to the system. Self-organized memory states are beneficial, because they exploit the natural dynamics of a system at hand - be it an RNN, or a physical system - while imposing as few constraints as possible on it. Our approach therefore offers promise for introducing natural dynamic attractors as memory-providing mechanisms in physical substrates used in reservoir computing setups. A more in-depth study of the system evolution with the DHL rule may also provide insights into how the human brain achieves such vast memory spans and fast computations.

References

[1] D. Durstewitz et al. Neurocomputational models of working memory. *Nature Neuroscience*, 3(11s):1184, 2000.

[2] O. Barak et al. Working models of working memory. *Current opinion in neurobiology*, 25:20–24, 2014.

[3] M. Rabinovich et al. Transient dynamics for neural processing. *Science*, pages 48–50, 2008.

[4] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

[5] W. Maass et al. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

[6] R. Pascanu et al. A neurodynamical model for working memory. *Neural networks*, 24(2):199–207, 2011.

[7] W. Maass et al. Computational aspects of feedback in neural circuits. *PLoS computational biology*, 3(1):e165, 2007.

[8] T. Nachstedt et al. Working memory requires a combination of transient and attractor-dominated dynamics to process unreliably timed inputs. *Scientific Reports*, 7(1):2473, 2017.

[9] H. Korn et al. Is there chaos in the brain? ii. experimental evidence and related models. *Comptes rendus biologies*, 326(9):787–840, 2003.

[10] W. Ross-Ashby. Principles of the self-organizing dynamic system. *Journal of General Psychology*, 37:125–128, 1947.

[11] H. Jaeger. Controlling recurrent neural networks by conceptors. *arXiv preprint arXiv:1403.3369*, 2014.

[12] R. Der et al. Novel plasticity rule can explain the development of sensorimotor intelligence. *PNAS*, 112(45):E6224–E6232, 2015.

[13] B. Kosko. Differential hebbian learning. In *AIP Conference proceedings*, volume 151, pages 277–282. AIP, 1986.

[14] Andrea Ceni, Peter Ashwin, and Lorenzo Livi. Interpreting recurrent neural networks behaviour via excitable network attractors. *Cognitive Computation*, pages 1–27, 2019.

[15] Y. Kuznetsov. *Elements of applied bifurcation theory*, volume 112. Springer Science & Business Media, 2013.

[16] C. Molter et al. The road to chaos by time-asymmetric hebbian learning in recurrent neural networks. *Neural computation*, 19(1):80–110, 2007.

[17] P. Földiak. Forming sparse representations by local anti-hebbian learning. *Biological cybernetics*, 64(2):165–170, 1990.

[18] M. Magnasco et al. Self-tuned critical anti-hebbian networks. *Physical review letters*, 102(25):258102, 2009.