

Reservoir Computing by Discretizing ODEs

Claudio Gallicchio

Department of Computer Science, University of Pisa
Largo Bruno Pontecorvo 3 - 56127 Pisa, Italy

Abstract. We draw connections between Reservoir Computing (RC) and Ordinary Differential Equations, introducing a novel class of models called Euler State Networks (EuSNs). The proposed approach is featured by system dynamics that are both stable and non-dissipative, hence enabling an effective transmission of input signals over time. At the same time, EuSN is featured by untrained recurrent dynamics, preserving all the computational advantages of RC models. Through experiments on several benchmarks for time-series classification, we empirically show that EuSN can substantially narrow the performance gap between RC and fully trainable recurrent neural networks.

1 Introduction

The study of neural network architectures from the perspective of dynamical systems is recently attracting increasing research attention. The fundamental intuition is that the computation performed by certain types of neural networks, e.g. Residual Networks, can be interpreted as the numerical solution of an ordinary differential equation (ODE) through discretization [1]. This observation leads to the possibility of enforcing desirable properties into the neural network behavior by imposing specific conditions on the corresponding ODE. Stability plays a fundamental role in this regard, being connected to the propagation of both input signals, during inference, and gradients, during training.

In this work, we focus on Recurrent Neural Network (RNN) architectures, and especially on Reservoir Computing (RC) [2]. The latter defines a particularly appealing approach for designing and training of RNNs, where the hidden recurrent *reservoir* layer is left untrained, leaving the burden of training only to the output layer. As such, stability of the forward signal propagation through the reservoir is of great importance. As the parameters of the temporal transformation developed by the reservoir are left untrained, some form of constraining is necessary to avoid instabilities when the network is put in operation with driving input. This aspect is connected to the well known fading memory property of RC networks, which makes it difficult to preserve the input signal information in the state dynamics across several time steps. Despite this limitation, the RC approach has become increasingly popular due to the formidable trade-off between predictive performance and efficiency of training algorithms. As a consequence, it is the paradigm of choice, e.g., when it comes to implementations in embedded systems [3] or in neuromorphic hardware. [4]. Still, the performance gap with state of the art fully trainable RNNs is sometimes unsatisfactory and leaves room for improvements.

In this paper, we introduce a novel RC technique that draws inspiration from numerical solutions of ODEs. The proposed architecture is achieved by forward Euler discretization of an ODE, and as such it is named *Euler State Network* (EuSN). The stability condition imposed to the ODE naturally leads to EuSN state dynamics that are both stable and non-dissipative, hence alleviating the lossy transmission of input signals over time. We comparatively validate our proposal with literature models on several time-series classification tasks.

2 Euler State Networks

In the context of sequence processing, we focus on ODEs of the form:

$$\mathbf{h}' = \tanh(\mathbf{W} \mathbf{h} + \mathbf{W}_x \mathbf{x} + \mathbf{b}), \quad (1)$$

where $\mathbf{h} \in \mathbb{R}^n$ is the state, $\mathbf{x} \in \mathbb{R}^m$ is the input, $\mathbf{W} \in \mathbb{R}^{n \times n}$ is a recurrent weight matrix, $\mathbf{W}_x \in \mathbb{R}^{n \times m}$ is an input weight matrix, and $\mathbf{b} \in \mathbb{R}^n$ is the bias. Ideally, to avoid both instability and dissipation of the state content over time, it is desirable that the Jacobian of (1) has eigenvalues with ≈ 0 real part. A simple way of achieving such a critical condition is to use an antisymmetric recurrent weight matrix, i.e. of the form $\mathbf{W} = \mathbf{W}_h - \mathbf{W}_h^T$. Relevantly, in this latter case, the eigenvalues of the Jacobian of the transformation in (1) are all imaginary [5], irrespective of the choice of the weight values. Hence, the critical behavior of the involved dynamics is not required to be learned from the data, rather it is an intrinsic property of the system.

Using the forward Euler method to numerically solve (1), and considering an antisymmetric recurrent weight matrix, leads to the following discretization:

$$\mathbf{h}_t = \mathbf{h}_{t-1} + \epsilon \tanh((\mathbf{W}_h - \mathbf{W}_h^T - \gamma \mathbf{I})\mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}), \quad (2)$$

which describes the state evolution of a neural network layer with n recurrent neurons over discrete time steps t . In (2), ϵ and γ are both small positive scalars, representing respectively the step size of integration and a diffusion coefficient for stabilizing the solution. We see (2) under the prism of RC, keeping the parameters in \mathbf{W}_h , \mathbf{W}_x and \mathbf{b} untrained. As the reservoir state of the system evolves as the forward Euler solution of an ODE, we call the resulting model Euler State Network (EuSN).

Interestingly, using standard RC arguments to analyze (2), i.e. studying the linearized system around the null state and for null input, reveals that the effective spectral radius of EuSN is given by the maximum among the absolute eigenvalues of $\tilde{\mathbf{W}} = (1 - \epsilon\gamma)\mathbf{I} + \epsilon(\mathbf{W}_h - \mathbf{W}_h^T)$. As ϵ and γ are both small positive values, the spectral radius of $\tilde{\mathbf{W}}$ is by construction confined in a small neighborhood around 1.

Differently from RC networks, the state evolved by EuSNs is not fading over time, i.e., the reservoir is not forgetting previous inputs during forward propagation. Hence, stability is intrinsic and there is no scaling condition to impose at initialization time. In other words, we see the stability condition

imposed to the ODE by antisymmetric force fields of the Jacobian in terms of an architectural bias for the resulting recurrent neural network. However, to balance between previous state and current input contributions in the state computation, we draw the weights in \mathbf{W}_h from a uniform distribution over $[-\alpha, \alpha]$, and those in \mathbf{W}_x and \mathbf{b} from $[-\beta, \beta]$, treating α and β as hyper-parameters.

As in standard RC approaches, the reservoir system described by (2) is coupled by a readout layer that is the only component of the architecture to undergo the training process.

2.1 Related Models

Within the umbrella of RC, (2) bears similarities with the state transition function of the popular Echo State Network (ESN) model [6] with leaky integrator neurons [7]. However, differently from the case of ESNs, the previous state term in the right hand side of 2 is not modulated by the leakage term. Moreover, the state evolution in (2) makes explicit use of an antisymmetric recurrent weight matrix. In the broader landscape of RNN research, a related model is the Antisymmetric RNN (A-RNN) [5], whose state dynamics are described by (2) as in EuSN. Crucially, while in A-RNN all the internal weights, i.e. \mathbf{W}_h , \mathbf{W}_x and \mathbf{b} , are trainable, in EuSN they are kept fixed. Under this perspective, the study of EuSN can be seen under the lens of randomized neural networks [8], emphasizing the natural abilities of stable RNN architectures based on forward Euler discretization and antisymmetric weight matrices even in the absence (or prior to) training of the internal connections.

3 Experiments

3.1 Datasets

We assessed EuSN on several classification tasks on time-series of diverse nature, taken from the literature [9]. We considered a variety of cases, including both uni-variate and multi-variate time-series, and both binary and multi-class classification. Table 1 gives an overview of the adopted datasets.

3.2 Settings

All datasets were used with the original training-test split. For model selection, we further derived a division into training and validation, with the latter containing 33% of the data. The values of hyper-parameters were tuned on the validation set using Hyperband [10]. In our experiments with EuSNs, we explored the following configurations: number of reservoir neurons $n \in [5, 200]$, recurrent and input scaling coefficients $\alpha, \beta \in [0.1, 1.5]$, step size and diffusion coefficient $\epsilon, \gamma \in [10^{-5}, 10^{-1}]$. As output readout classifier we used a single dense layer trained by RMSprop, with learning rate $\eta \in [10^{-5}, 10^{-1}]$, for a maximum

Name	# Seq.	Max. Length	# Feat.	# Classes
SpokenArabicDigits	8798	93	13	10
CharacterTrajectories	2858	182	3	20
Libras	360	45	2	15
Wafer	7164	152	1	2
ECG5000	5000	140	1	5
Epilepsy	275	206	3	4

Table 1: Overview of the used datasets, including the total number of sequences (# Seq), the maximum length of a sequence (Max. Length), the number of input features per time-step (# Feat.), and the number of target classes (# Classes).

number of 200 epochs, using early stopping with patience of 10. After model selection, we trained 5 models with the selected hyper-parametrization on the whole training set, assessing the accuracy on the test set, averaging the results (and computing standard deviations) on the 5 runs.

For comparison, we performed the same experiments with ESN, A-RNN and GRU. In these cases we used a similar model selection and performance assessment process as outlined for EuSN, exploring the same values for the number of recurrent units and for the hyper-parameters of the training algorithm. For ESN, we configured the reservoir exploring values of spectral radius and input scaling in $[0.1, 1.5]$, and leaking rate in $[0.1, 1]$. For EuSN and ESN the training algorithm is restricted to the output layer, while for A-RNN and GRU it operates on all the network parameters. We ran our experiments on a MacBook Pro laptop with a 2,8 GHz Quad-Core Intel Core i7 processor and 16 GB of RAM.

3.3 Results

The achieved results are given in Table 2. For every dataset and model, we report the set accuracy (Acc), the number of trainable parameters of the selected hyper-parametrization (Params), indicating also the time (in minutes) required for its training and testing (Time). Finally, we also indicate the time required for the whole process of model selection (MS Time).

From the results, it is apparent that EuSN outperforms ESN in all the explored cases. In tasks where ESN already achieves very good accuracy (such as Wafer and ECG5000), EuSN leads to an even improved result, with smaller standard deviations. On tasks in which ESN performs poorly, EuSN is able to reach the same level of accuracy of the fully trainable models A-RNN and GRU, or in any case to significantly bridge the performance gap. Intriguingly, the boost in classification accuracy is not paid in terms of computational resources. In fact, in terms of time, EuSN is as efficient as ESN. Leveraging the untrained dynamics results in a dramatic advantage compared to A-RNN and GRU, as confirmed by the results in Table 2.

SpokenArabicDigits				
	<i>Acc</i>	<i>Params</i>	<i>Time (min.)</i>	<i>MS Time</i>
EuSN (ours)	0.956 (± 0.005)	1850	0.270 (± 0.075)	00h 17m 50s
ESN	0.878 (± 0.013)	1020	0.108 (± 0.018)	00h 17m 27s
A-RNN	0.979 (± 0.004)	43547	4.799 (± 0.337)	06h 46m 52s
GRU	0.978 (± 0.004)	116382	5.010 (± 0.645)	05h 39m 53s
CharacterTrajectories				
	<i>Acc</i>	<i>Params</i>	<i>Time (min.)</i>	<i>MS Time</i>
EuSN (ours)	0.979 (± 0.002)	2640	0.145 (± 0.005)	00h 09m 28s
ESN	0.904 (± 0.013)	2040	0.062 (± 0.006)	00h 08m 55s
A-RNN	0.979 (± 0.003)	7445	1.472 (± 0.257)	01h 31m 16s
GRU	0.966 (± 0.006)	99108	2.058 (± 0.379)	02h 17m 00s
Libras				
	<i>Acc</i>	<i>Params</i>	<i>Time (min.)</i>	<i>MS Time</i>
EuSN (ours)	0.620 (± 0.008)	2130	0.145 (± 0.005)	00h 06m 24s
ESN	0.269 (± 0.019)	2655	0.037 (± 0.001)	00h 06m 41s
A-RNN	0.619 (± 0.009)	4159	0.254 (± 0.016)	00h 10m 39s
GRU	0.772 (± 0.046)	19881	0.152 (± 0.010)	00h 15m 21s
Wafer				
	<i>Acc</i>	<i>Params</i>	<i>Time (min.)</i>	<i>MS Time</i>
EuSN (ours)	0.990 (± 0.002)	189	0.223 (± 0.035)	00h 08m 41s
ESN	0.983 (± 0.008)	197	0.089 (± 0.009)	00h 08m 11s
A-RNN	0.993 (± 0.001)	17029	2.645 (± 0.190)	01h 21m 56s
GRU	0.987 (± 0.005)	19033	1.321 (± 0.152)	02h 04m 53s
ECG5000				
	<i>Acc</i>	<i>Params</i>	<i>Time (min.)</i>	<i>MS Time</i>
EuSN (ours)	0.935 (± 0.000)	945	0.118 (± 0.014)	00h 07m 29s
ESN	0.913 (± 0.011)	565	0.064 (± 0.011)	00h 06m 57s
A-RNN	0.933 (± 0.002)	545	0.448 (± 0.069)	00h 50m 17s
GRU	0.926 (± 0.007)	57397	0.619 (± 0.148)	01h 35m 49s
Epilepsy				
	<i>Acc</i>	<i>Params</i>	<i>Time (min.)</i>	<i>MS Time</i>
EuSN (ours)	0.868 (± 0.010)	468	0.076 (± 0.003)	00h 07m 10s
ESN	0.709 (± 0.033)	708	0.060 (± 0.009)	00h 06m 37s
A-RNN	0.872 (± 0.020)	2389	0.684 (± 0.095)	00h 30m 31s
GRU	0.872 (± 0.047)	576	0.174 (± 0.022)	00h 31m 15s

Table 2: Results on time-series classification problems achieved by EuSN, compared to ESN, A-RNN and GRU. For every dataset we report the accuracy on the test set (Acc), the number of trainable parameters (Params), the time for training and test (Time), and for the whole model selection (MS Time).

4 Conclusions

In this paper, we have introduced a novel class of Reservoir Computing (RC) neural networks, called Euler State Networks (EuSNs), inspired by the numerical solution of Ordinary Differential Equations (ODE). The proposed approach is based on forward Euler discretization, and makes use of antisymmetric recurrent weight matrix to bias the eigenspectrum of the resulting Jacobian near zero real parts. As a result, the employed reservoir dynamics are neither lossy nor unstable, and can effectively preserve information across time steps.

The proposed EuSN approach combines the ability to model long-term propagation of input signals with the typical efficiency of RC. Experiments on several classification problems show that EuSN consistently outperforms ESN in terms of accuracy while keeping similar execution times. At the same time, EuSN can substantially reduce the accuracy gap of RC with respect to state of the art fully trainable RNNs, while offering an evident computational advantage.

The study presented in this paper is empirical and, though preliminary, very promising, putting forward EuSNs as an effective alternative to ESNs for classification problems. Future works will go more in-depth in analyzing ODE-inspired RC architectures and their theoretical properties.

References

- [1] Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.
- [2] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.
- [3] Mauro Dragone, Giuseppe Amato, Davide Bacciu, Stefano Chessa, Sonya Coleman, Maurizio Di Rocco, Claudio Gallicchio, Claudio Gennaro, Hector Lozano, Liam Maguire, et al. A cognitive robotic ecology approach to self-configuring and evolving aal systems. *Engineering Applications of Artificial Intelligence*, 45:269–280, 2015.
- [4] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, 2019.
- [5] Bo Chang, Minmin Chen, Eldad Haber, and Ed H Chi. Antisymmetricrnn: A dynamical system view on recurrent neural networks. *arXiv preprint arXiv:1902.09689*, 2019.
- [6] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004.
- [7] Herbert Jaeger, Mantas Lukoševičius, Dan Popovici, and Udo Siewert. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural networks*, 20(3):335–352, 2007.
- [8] Claudio Gallicchio and Simone Scardapane. Deep randomized neural networks. *Recent Trends in Learning From Data*, pages 43–68, 2020.
- [9] Anthony Bagnall, Jason Lines William Vickers, and Eamonn Keogh. The uea & ucr time series classification repository. www.timeseriesclassification.com.
- [10] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.