Residual Integration Neural Network

Said Ouala, Ananda Pascual, Ronan Fablet

RESIDUAL INTEGRATION NEURAL NETWORK

Said Ouala [1]*, Ananda Pascual* [2]*, Ronan Fablet* [1]

(1) IMT Atlantique; Lab-STICC, Brest, France
(2) IMEDEA, UIB-CSIC, Esporles, Spain

ABSTRACT

In this work, we investigate residual neural network representations for the identification and forecasting of dynamical systems. We propose a novel architecture that jointly learns the dynamical model and the associated Runge-Kutta integration scheme. We demonstrate the relevance of the proposed architecture with respect to learning-based state-of-the-art approaches in the identification and forecasting of chaotic dynamics when provided with training data with low temporal sampling rates.

***Index Terms*—** Dynamical systems, Data-driven models, Neural networks, Forecasting, Runge-Kutta methods

1. PROBLEM STATEMENT AND RELATED WORK

The modeling of physical dynamics is a critical issue. While model-driven strategies based on the definition of ordinary differential equations (ODE) governing the observable phenomena are the classic frameworks to address such a problem [1]. Limitations in terms of numerical complexity [2] as well as the ability to better relate models and observation data for poorly-resolved processes [3] open the venue for data-driven representations as an appealing alternative [4], where one can determine a representation of processes of interest directly from data, especially with the increasing availability of representative data collections.

Regarding the data-driven identification of dynamical systems, one may distinguish representations based on physical priors and machine learning. The first category comprises polynomial representations [5] and sparse regression frameworks [6]. They are particularly appealing in explicitly relating the learnt representation and the associated physical operator. Such methods may, however, fail in representing complex non-linear systems for which strong priors are not available. Machine learning methods can greatly broaden modeling capabilities often however at the expense of a lack of interpretability. A large interest has for instance recently emerged in analog methods, which are based on nearest neighbours [7], in ocean and atmosphere science. Though leading to significant advances for simulation and reconstruction issues, the lack of physical interpretability of such approaches advocates for frameworks bridging the physical paradigm and the statistical paradigm underlying machine learning. In this respect, neural networks, especially Residual Networks (ResNet), are of key interest [8]. More specifically, the discretized numerical integration schemes of an ODE can be stated as a ResNet [9, 10], which allows for the data-driven identification of the dynamical operator governing a dynamical system of interest. To our knowledge, previous works mostly focused on the parameterization of the dynamical operator using or not physical priors (e.g., bilinear setting [10], physics-informed parameterization [11]). Besides the identification of the dynamical operator, the selection of the numerical integration scheme (e.g., explicit Eurler and Runge-Kutta schemes) may significantly affect modeling and forecasting performance, especially when the data available for training may not involve fine-scale time sampling with respect to the characteristics time scales of the considered processes, as shown in a previous work [10]. Such issues appear critical when addressing the identification of dynamical systems from observation data (e.g., satellite earth observation data).

In this work, we address the joint data-driven identification of the dynamical operator governing a dynamical process of interest and the associated numerical integration scheme. We propose a residual integration neural network which jointly learns a dynamical model and an explicit Runge-Kutta integration scheme with an arbitrary number of stages. From an insight on high order numerical integration schemes, we demonstrate the relevance of the proposed architecture for identification and forecasting purposes when considering large integration step. We make explicit the relationship between the number of residual blocks and the order of a given integration scheme in terms of truncation error and show that increasing the number of residual layers in our architecture results in a behaviour similar to the integration of an ODE depending on the integration time-step. Overall, our key contributions are three-fold : i) we propose a new neural network architecture for the joint identification of dynamical systems and their corresponding integration scheme ; ii)

This work was supported by GERONIMO project (ANR-13-JS03-0002), Labex Cominlabs (grant SEACS), Region Bretagne, CNES (grant OSTST-MANATEE), Microsoft (AI EU Ocean awards) and by MESR, FEDER, Région Bretagne, Conseil Général du Finistère, Brest Métropole and Institut Mines Télécom in the framework of the VIGISAT program managed by "Groupement Bretagne Télédétection" (BreTel).

we make explicit the link between the considered residual architecture and high-order integration schemes in terms of truncation error, iii) we demonstrate the relevance of the proposed architecture for Lorenz-63 and Lorenz-96, which are representative of chaotic geophysical dynamics.

2. NUMERICAL INTEGRATION AND RUNGE-KUTTA METHODS

This section briefly introduces Runge-Kutta numerical integration schemes, that will provide the basis for the definition and analysis of the proposed residual integration architecture.

Let us consider a dynamical system, whose time-varying state X_t is governed by the following ordinary differential equation (ODE) :

$$\frac{dX_t}{dt} = F(t, X_t) \tag{1}$$

where F is the dynamical operator. Most of the time, this ODE cannot be solved analytically and numerical integration techniques using discrete approximations are implemented.

Assuming that we are provided with an initial condition X_{t_0}, we aim to solve the ODE for an interval $t \in [t_0, t_f]$. Given a discretization of the interval using a time-step $h > 0$ as $h = \frac{t_f - t_0}{N}$ and $t_n = t_0 + nh$, where $0 < n < N$ an integer and N is the number of grid points, it comes to approximate the value of variable X_t at each grid point : $X_{t_1}, ..., X_{t_n}, ..., X_{t_N}$. Explicit and implicit numerical integration schemes may be considered [12]. In this work, we focus on explicit integration schemes. A one-step explicit integration scheme is defined as :

$$X_{t_{n+1}} = X_{t_n} + h\Phi(t_n, X_{t_n}, h) \tag{2}$$

with $\Phi(t_n, X_{t_n}, h)$ a numerical integration operator. Here, we aim to learn a prediction operator based on Φ so that the forecasting error (typically, a one-step-ahead error) is minimized. From an integration point of view, one may rather consider the truncation error to characterize the numerical integration scheme. The truncation error is defined with respect to the true analytic solution X_t^T as follows :

$$e_n = X^T_{t_{n+1}} - X_{t_{n+1}}(t_n, X^T_{t_n}, h) \tag{3}$$

$$e_n = X^T_{t_{n+1}} - X^T_{t_n} - h\Phi(t_n, X^T_{t_n}, h) \tag{4}$$

A p-order numerical resolution method can be derived, using the Taylor development of the analytic solution X_t^T up to the order $p+1$. Assuming that F is a $\mathcal{C}^p$ function (p times derivable with a continuous p^{th} derivative), we can write the Taylor expansion as :

$$\begin{aligned} X^T_{t_{n+1}} = X^T_{t_n} &+ \sum_{k=1}^{p} h^k \frac{1}{k!} F^{k-1}(t_n, X^T_{t_n}) \\ &+ h^{p+1} \frac{1}{(p+1)!} F^p(t_n, X^T_{t_n}) + O(h^{p+2}) \end{aligned} \tag{5}$$

The corresponding p-order numerical integration scheme can be derived by replacing $\Phi(t_n, X_{t_n}, h)$ in Equation 2 such as :

$$\Phi(t_n, X_{t_n}, h) = \sum_{k=1}^{p} h^{k-1} \frac{1}{k!} F^{k-1}(t_n, X_{t_n}) \tag{6}$$

The corresponding truncation error of the p-order method can be deduced by neglecting the term $O(h^{p+2})$ in equation 5 as :

$$e_n = h^{p+1} \frac{1}{(p+1)!} F^p(t_n, X_{t_n}) \tag{7}$$

The explicit Euler method corresponds to $p = 1$ and its truncation error is proportional to h^2. To use a first-order method like Euler, the integration time step should be small enough which is not always possible for complex systems due to computational issues. Higher-order techniques are more robust to the integration time step [12]. However, the computation of high-order derivatives becomes quickly expensive which may limit their use in practice. Runge-Kutta integration schemes were introduced as an efficient trade-off between high-order approximations and computational complexity. It relies on the following recurrent update :

$$X_{t_{n+1}} = X_{t_n} + \sum_{i=1}^{s} \beta_i k_i \tag{8}$$

where s is the number of stages of the method, $k_i = F(t_n + c_i h, X_{t_n} + h(\sum_{j=1}^{i-1} \alpha_{i,j} k_j))$ with $0 < j < i \leq s$ and $\sum_{i=1}^{s} \beta_i = 1$, $0 < c_i < 1$, $\sum_{j=1}^{i-1} \alpha_{i,j} = c_i$. When $s = 1$, it simply corresponds to the explicit Euler method. For a given number of stages s, Runge-Kutta method coefficients need to satisfy some extra conditions (by matching it to the corresponding Taylor series) to get a given order p [13]. Formally, the Runge-Kutta method order p is always inferior or equal to the number of stages s. For $s = 4$, we retrieve the well-known Runge-Kutta-4 method. For $p > 4$, we need more integration stages s to truly reach a given error order p.

3. RESIDUAL INTEGRATION NETWORK

This section introduces the proposed residual integration neural network (RINN) framework. We first introduce the proposed architecture and the associated learning scheme. We then analyze the characteristics of RINNs in terms of forecasting error.

3.1. Proposed Neural Network architecture

Let us assume we are provided with representative time series $\{X\}$ with a given time sampling rate h, which are governed by an unknown ODE. For the sake of simplicity, we consider below a single time series of length $N+1$, $\{X_0, X_2, \ldots, X_N\}$. The same applies for a dataset formed by different time series possibly of varying lengths. We aim to identify the unknown dynamical operator F (Equation (1)) from time series $\{X\}$ when sampling rate h may be high. As illustrated in the reported experiments, in such situations, Euler-based learning schemes [6, 10] may fail in providing relevant forecasts.

Motivated by the effectiveness of high-order integration schemes in solving differential equations with relatively high time-step h, we propose a novel architecture based on residual networks and Runge-Kutta to effectively identify dynamical systems when provided with observations with low time sampling rates. The proposed architecture involves a residual neural network architecture. A residual block F_{NN} is shared upon all the residual layers up to the predefined stage S. This residual block is the neural-network parameterization of the dynamical operator F in Equation (1). Our architecture mimics a Runge-Kutta numerical integration scheme with S stages, which imposes the following constraints on weighing parameters $\{\beta_i\}_i$, $\{\alpha_{i,j}\}_i$ and $\{c_i\}_i$:

$$\sum_{i=1}^{\hat{s}} \beta_i = 1, \ \ \forall i, \ 0 < c_i < 1 \text{ and } \sum_{j=1}^{i-1} \alpha_{i,j} = c_i \tag{9}$$

Overall, two main components need to be defined to specify a RINN :

— The parametrization chosen for the residual block F_{NN} approximating our true dynamical model F in terms of neural network structures. It may rely on physics-informed parameterizations [6, 14, 10, 11].;

— The number of stages S of our residual integration network.

The learning procedure is stated as the minimization of the forecasting error subject to constrain (9) :

$$\min_{\theta_{NN},c,\beta,\alpha} \sum_{n=1}^{N} \|X_{t_n}^T - \Psi(X_{t_{n-1}}^T, \theta_{NN}, c, \beta, \alpha)\| \tag{10}$$

$$\text{subject to} (9)$$

where Ψ is the output of the RINN obtained by applying the Runge-Kutta recursion (Equation 8) based on the approximate model F_{NN} and the weights c, α and β introduced above. θ_{NN} are the parameters of operator F_{NN}. We implement the considered architecture and learning criterion under pytorch framework. As optimization solver, we consider the ADAM algorithm [15]. The constrained optimization is solved by clipping the integration weights after each training epoch.

3.2. Performance of the learnt RINN

Assuming that the S-stage RINN corresponds to an $\hat{p}$-order numerical integration scheme, the loss function of the RINN relates to the truncation error of the learnt integration scheme :

$$\hat{e_n}^2 = (X_{t_{n+1}}^T - \hat{X}_{t_{n+1}})^2 \tag{11}$$

where $\hat{X}_{t_{n+1}}$ is the output of our RINN.

Using the Taylor expansion given by equation 5 over the true state $X_{t_{n+1}}^T$ up to the order $p+1$, the training error of the learnt $\hat{p}$-order numerical integration scheme is given by :

$$\begin{aligned} \hat{e_n}^2 = (&\sum_{k=1}^{p} h^k \frac{1}{k!} F^{k-1}(t_n, X_{t_n}^T) \\ &+ h^{p+1} \frac{1}{(p+1)!} F^p(t_n, X_{t_n}^T) \\ &- \sum_{k=1}^{\hat{p}} h^k \frac{1}{k!} \hat{F}^{k-1}(t_n, X_{t_n}^T))^2 \end{aligned} \tag{12}$$

This squared truncation error depends on two learnt parameters θ_{NN} (*i.e.*, the parameters of dynamical operator F_{NN}) and $\hat{p}$. It reaches a minimum for $F_{NN} = F$ and $\hat{p} = p$. Hence, a theoretic lower bound of the training loss function of the RINN is given by the truncation error of our true dynamical model :

$$\hat{e_n}^2 > (h^{p+1} \frac{1}{(p+1)!} F^p(t_n, X_{t_n}^T)))^2 \tag{13}$$

Equations 12 and 13 illustrate two main characteristics about learning neural network representations of dynamical models : (i) one cannot expect a training error lower then a theoretical lower bound represented by the truncation error of the true dynamical model (assuming p to be high enough to properly integrate the true dynamical system so that terms proportional to h^{p+2} or higher are negligible), (ii) we may jointly tune F_{NN} and $\hat{p}$ in the RINN architecture to lower the training loss function. Assuming that the integration time step h is set by the temporal sampling of our training data, one may improve the approximation F_{NN} of the true dynamical model F as mostly studied in the data-driven community [6, 10]. One may also decrease the training loss function through a greater order $\hat{p}$ of the integration scheme reproduced by the RINN, parameterized through the number of stages S. This clearly motivates the development of residual networks with several residual layers. Our previous work [10] can be regarded as an illustration of this aspect. We showed that a residual dynamical model reproducing the Runge-Kutta-4 scheme outperforms the model reproducing the Euler setting and this even with a low integration time-step. Those results clearly relate to Equation 12.

4. NUMERICAL EXPERIMENTS

In this section, we evaluate the proposed framework and demonstrate its relevance to identify and forecast dynamical systems governed by an unknown ODE when only provided with data with a low sampling rate. As case studies, we consider two models widely studied in geophysics as examples of chaotic patterns in ocean-atmosphere science [16].

4.1. Case studies

Lorenz-63 system : The Lorenz 63 dynamical system is a 3-dimensional model governed by the following ODE :

$$\begin{cases} \frac{dX_{t,1}}{dt} &= \sigma\left(X_{t,2} - X_{t,2}\right) \\ \frac{dX_{t,2}}{dt} &= \rho X_{t,1} - X_{t,2} - X_{t,1}X_{t,3} \\ \frac{dX_{t,3}}{dt} &= X_{t,1}X_{t,2} - \beta X_{t,3} \end{cases} \tag{14}$$

Under parameterization $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$, this system involves chaotic dynamics with two attractors [16].

Lorenz-96 system : The Lorenz 96 dynamical system is a 40-dimensional system. It involves propagation-like dynamics governed by :

$$\frac{dX_{t,i}}{dt} = (X_{t,i+1} - X_{t,i-2})X_{t,i-1} + A \tag{15}$$

with periodic boundary conditions (*i.e.* $X_{t,-1} = X_{t,40}$ and $X_{t,41} = X_{t,1}$) and $A = 8$. Lorenz-96 system provides means to demonstrate the relevance of the proposed framework for higher-dimensional systems.

We simulate Lorenz-63 (resp. Lorenz-96) state sequences using the LOSDA ODE solver [17] with an integration step of 0.01 (resp. 0.05). We then subsample the simulated sequences to different timesteps while making sure we don't exceed the characteristic time-scale.

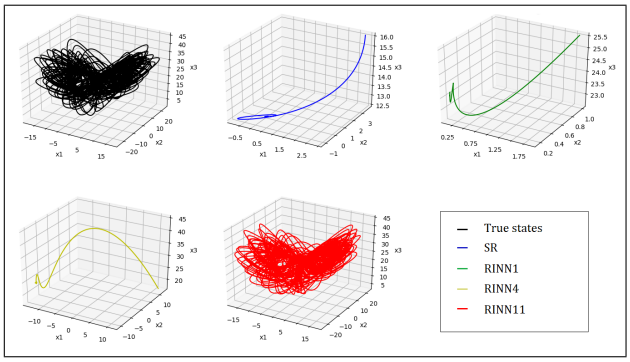

Fig. 1: ***Generated time series of the proposed models.*** *Generated time series of the proposed models for $h = 0.4$. For visualization purpose, the time series were interpolated to an $h = 0.01$ grid using a cubic interpolation.*

4.2. Results

In this section, we compare several Residual Integration neural network performances in predicting the Lorenz dynamics from a given initial state. For benchmarking purpose, the following models were tested :

— **Sparse regression model** [6] (SR) : This model computes a sparse regression over an augmented states vector based on second order polynomial representations of the Lorenz states. The learnt dynamical model is then integrated to compute forecasts using the LOSDA ODE solver [17].

Model		h=0.3	h=0.4	h=0.5
SR	$t_0 + h$	11.10	12.56	7.48
	$t_0 + 4h$	9.64	12.51	57.90
RINN1	$t_0 + h$	11.36	10.64	3.83
	$t_0 + 4h$	8.09	9.96	8.60
RINN4	$t_0 + h$	2.24	7.66	2.80
	$t_0 + 4h$	8.33	10.79	8.64
RINN11	$t_0 + h$	**0.23**	**0.54**	**0.41**
	$t_0 + 4h$	**1.01**	**2.22**	**2.06**

Table 1: ***Forecasting performance of data-driven models for Lorenz-63 dynamical model*** *: mean RMSE for different forecasting time steps.*

Model		h=0.3	h=0.4	h=0.5
RINN4	$t_0 + h$	1.30	2.89	2.76
	$t_0 + 4h$	2.69	3.31	**3.05**
RINN11	$t_0 + h$	**0.02**	**0.68**	**2.06**
	$t_0 + 4h$	**0.09**	**2.48**	3.34

Table 2: ***Forecasting performance of data-driven models for Lorenz-96 dynamical model*** *: mean RMSE for different forecasting time steps.*

— **Residual Integration Neural Network 1** (RINN1) : the proposed residual architecture with a number of stages equal to one. This corresponds to the first order Euler integration method.

— **Residual Integration Neural Network 4** (RINN4) : the proposed residual architecture with a number of stages equal to four. This comprises the fourth-order Runge-Kutta 4 integration technique with integration parameters $\{\beta_i\}_i$, $\{\alpha_{i,j}\}_i$ and $\{c_i\}_i$ set to the true Runge Kutta 4 parameters.

— **Residual Integration Neural Network 11** (RINN11) : Proposed residual architecture with a number of stages equal to 11. In this architecture, the weights of the integration scheme are learnt as explained in section 3.1.

In all these reported experiments, the parameterization used for the neural-network approximation F_{NN} of the dynamical operation F is a bilinear architecture as proposed in [10]. This bilinear architecture ensures that the true model truly lies within the space of possible model parameterizations.

We report the forecasting performances in Tab. 1 and 2. Figure 1 illustrates the trajectories generated using the trained data-driven models on the Lorenz-63 system with $h = 0.4$. The proposed residual integration neural network with 11 residual layers leads to the best performances comparing to identification techniques based on low order integration schemes. This gain clearly motivates the investigation of such representations in data-driven dynamical modeling.

5. CONCLUSION

In this work, we demonstrate the relevance of the residual integration neural network in the identification of dynamical systems. Through the representation of residual networks as high order numerical integration schemes, we prove that high order residual networks can allow the learning of dynamical models even when provided with training data with low temporal sampling. Further works could investigate a relationship between the number of stages in our residual network and the corresponding order of the learnt numerical integration scheme.

6. REFERENCES

[1] Geir Evensen, *Data Assimilation*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[2] Song-You Hong and Jimy Dudhia, "Next-generation numerical weather prediction : Bridging parameterization, explicit clouds, and large eddies," *Bulletin of the American Meteorological Society*, vol. 93, no. 1, pp. ES6–ES9, 2012.

[3] van Leeuwen P. J., "Nonlinear data assimilation in geosciences : an extremely efficient particle filter," *Quarterly Journal of the Royal Meteorological Society*, vol. 136, no. 653, pp. 1991–1999, dec 2010.

[4] Pierre Tandeo, Pierre Ailliot, Bertrand Chapron, Redouane Lguensat, and Ronan Fablet, "The analog data assimilation : application to 20 years of altimetric data," in *International Workshop on Climate Informatics*, Boulder, United States, sep 2015, pp. 1 – 2.

[5] Johan Paduart, Lieve Lauwers, Jan Swevers, Kris Smolders, Johan Schoukens, and Rik Pintelon, "Identification of nonlinear systems using Polynomial Nonlinear State Space models," *Automatica*, vol. 46, no. 4, pp. 647–656, Apr. 2010.

[6] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016.

[7] Redouane Lguensat, Pierre Tandeo, Pierre Ailliot, Manuel Pulido, and Ronan Fablet, "The Analog Data Assimilation," *Monthly Weather Review*, aug 2017.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," *arXiv :1512.03385 [cs]*, december 2015, arXiv : 1512.03385.

[9] Yi-Jen Wang and Chin-Teng Lin, "Runge-Kutta neural network for identification of dynamical systems in high accuracy," *IEEE Transactions on Neural Networks*, vol. 9, no. 2, pp. 294–307, mar 1998.

[10] Ronan Fablet, Said Ouala, and Cedric Herzet, "Bilinear residual Neural Network for the identification and forecasting of dynamical systems," *SciRate*, dec 2017.

[11] Emmanuel de Bezenac, Arthur Pajot, and Patrick Gallinari, "Deep learning for physical processes : Incorporating prior scientific knowledge," *CoRR*, vol. abs/1711.07970, 2017.

[12] Isaac Fried, *Numerical Solution of Differential Equations*, Academic Press, Inc., Orlando, FL, USA, 1979.

[13] J. C. Butcher, "Coefficients for the study of runge-kutta integration processes," *Journal of the Australian Mathematical Society*, vol. 3, no. 2, pp. 185–201, 1963.

[14] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Multistep Neural Networks for Data-driven Discovery of Nonlinear Dynamical Systems," *ArXiv e-prints*, Jan. 2018.

[15] Diederik P. Kingma and Jimmy Ba, "Adam : A Method for Stochastic Optimization," *arXiv :1412.6980 [cs]*, Dec. 2014, arXiv : 1412.6980.

[16] Edward N. Lorenz, "Deterministic Nonperiodic Flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, Mar. 1963.

[17] A. C. Hindmarsh, "ODEPACK, a systematized collection of ODE solvers," *IMACS Transactions on Scientific Computation*, vol. 1, pp. 55–64, 1983.