Bilinear Residual Neural Network for the Identification and Forecasting of Geophysical Dynamics

Ronan Fablet [1], *Said Ouala* [1], *Cédric Herzet* [1,2]

(1) IMT Atlantique; Lab-STICC, Brest, France
(2) INRIA Bretagne-Atlantique, Fluminance, Rennes, France

ABSTRACT

Due to the increasing availability of large-scale observations and simulation datasets, data-driven representations arise as efficient and relevant computation representations of geophysical systems for a wide range of applications, where model-driven models based on ordinary differential equations remain the state-of-the-art approaches. In this work, we investigate neural networks (NN) as physically-sound data-driven representations of such systems. Viewing Runge-Kutta methods as graphical models, we consider a residual NN architecture and introduce bilinear layers to embed non-linearities which are intrinsic features of geophysical systems. From numerical experiments for synthetic and real datasets, we demonstrate the relevance of the proposed NN-based architecture both in terms of forecasting performance and model identification.

***Index Terms*—** Dynamical systems, neural networks, Bilinear layer, Forecasting, ODE, Runge-Kutta methods

1. PROBLEM STATEMENT AND RELATED WORK

Model-driven strategies have long been the classic framework to address forecasting and reconstruction of geophysical systems [1]. The ever increasing availability of large-scale observations and simulation datasets make more and more appealing the development of data-driven strategies [2] especially when dealing with computationally-demanding models or high modeling uncertainties [1].

In this context, data-driven schemes typically aim to identify computational representations of the dynamics of a given state from data, *i.e.* the time evolution of the variable of interest. Physical models usually describe this time evolution through an ordinary differential equation (ODE). One may distinguish two main families of data-driven approaches. A first category involves global parametric representations derived from physical principles [3]. Polynomial representations are typical examples [4]. The combination of such representations with sparse regression recently opened new research avenues. A second category of approach adopts a machine learning point of view and states the considered issue as a regression problem for a predefined time step dt, *i.e.* the regression of the state at time $t + dt$ given the state at time t.

A variety of machine learning regression models have been investigated, among which neural networks and nearest-neighbor models (often referred to as analog forecasting models in geoscience) are the most popular ones [5, 6]. Such approaches offer more modeling flexibility to optimize forecasting performance, at the expense however of a lack of interpretability of the learnt representation. Regarding neural network representations, a variety of recurrent neural networks [7] have been proposed to address non-linear dynamics. They can lead to very accurate forecasting performance, but they do not provide an explicit representation of the systems of interest in terms of physically-interpretable differential operators. With a view to jointly uncovering the governing equations of geophysical processes and optimizing forecasting performance, one may investigate neural network representations designed as numerical integration schemes of ordinary differential equations [8]. In [8] however, the authors only embed classic non-linear activation functions. Though such activations may theoretically represent any type of non-linearities, this representations lead to overcomplex approximations witch have trouble to efficiently encode bilinear non-linearities frequently encountered in geophysical dynamics.

In this work, we investigate such residual neural network representations for geophysical dynamics. We aim to derive computationally-efficient and physically-sound representations. Our contribution is three-fold : i) we introduce a NN architecture with bilinear layers to embed intrinsic non-linearities depicted by the dynamical systems, ii) we make possible the physical interpretation of some NN models based on bilinear non-linearities, iii) we demonstrate the relevance of the proposed NN architecture with respect to state-of-the-art models both in terms of model identification and forecasting for synthetic datasets, namely Lorenz-63 and Lorenz-96 dynamics [9] which are representative of ocean-atmosphere dynamics and for real sea surface temperature anomaly data.

This paper is organized as follows. Section 2 describes

This work was supported by Labex Cominlabs (grant SEACS), GERONIMO project (ANR-13-JS03-0002) and CNES (grant OSTST-MANATEE).

the proposed NN-based architecture for dynamical systems. Section 3 presents numerical experiments. We further discuss our contributions in Section 4.

2. NEURAL NET ARCHITECTURES FOR DYNAMICAL SYSTEMS

We present in this section the proposed NN architecture to represent and forecast a dynamical system governed by an unknown ODE. We first point out the graphical representation of Runge-Kutta methods as residual neural nets as shown in [8]. Based on this graphical representation, we introduce the proposed bilinear NN. We then discuss training issues and applications to forecasting and reconstruction problems.

2.1. Runge-Kutta methods as residual neural nets

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

$$\frac{dX_t}{dt} = F\left(X_t, \theta\right) \tag{1}$$

where F is the dynamical operator and θ some parameters. The fourth-order Runge-Kutta integration scheme is among the most classical ones for simulating state dynamics from a given initial condition $X(t_0)$. It relies on the following sequential update for a predefined integration time step dt :

$$X_{t_0+(n+1)dt} = X_{t_0+n\cdot dt} + \sum_{i=1}^{4} \alpha_i k_i \tag{2}$$

$\{k_i\}$ are defined as follows : $k_i = F\left(X_{t_0+\beta_i k_{i-1} dt}, \theta\right)$ with $k_0 = 0$, $\alpha_1 = \alpha_4 = 1/6$, $\alpha_2 = \alpha_3 = 2/6$, $\beta_1 = \beta_4 = 1$ and $\beta_2 = \beta_3 = 1/2$.

Runge-Kutta integration scheme (2) may be restated using a graphical model as illustrated in the bottom panel of Fig.1. Assuming we are given an approximate model $\hat{F}$ of the true dynamical operator F, the fourth-order runge-Kutta scheme can be regarded as a recurrent network with a four-layer residual net [8], each layer sharing the same operator $\hat{F}$. In this architecture, coefficients $\{\alpha_i\}_i$ refers to the relative weights given to the outputs of the four repeated blocks $\hat{F}$. The same holds for coefficients β_i which refer to the weight given to the output from block $i-1$ when added to input X_t and feeded to block i.

Based on this representation of numerical integration (2) as a residual net, we may state the identification of dynamical operator F in (1) as the learning of the parameters of our recurrent residual NN block $\hat{F}$ stated as in Fig. 1. The other parameters, namely coefficients $\{\alpha_i\}_i$ and $\{\beta_i\}_i$, may be set to the values used in the fourth-order Runge-Kutta scheme or learnt from data. Overall, the key aspect of the considered residual NN is the architecture and parameterization chosen for the shared block $\hat{F}$ that approximates the dynamical operator F. We may also stress that the fourth-order architecture sketched in Fig.1 may be extended to any lower- or higher-order scheme. As a special case, the explicit Euler scheme leads to a one-block architecture.

2.2. Proposed bilinear neural net architecture

Neural net architectures classically exploit convolutional, fully-connected and non-linear activation layer [10]. Following this classic framework, operator F may be approximated as a combination of such elementary layers. It may be noted that dynamical systems, as illustrated for instance by Lorenz dynamics (3) and (4), involve non-linearities, which might not be well-approximated by the combination of a linear transform of the inputs of non-linear activation functions. Especially physical dynamical systems often involve bilinear non-linearities, which express some multiplicative interaction between two physical variables [3, 11]. Among classic physical models, one may cite for instance advection-diffusion dynamics or shallow water equations. Polynomial decompositions then appear as natural representation of dynamical systems for instance for model reduction issues [4].

These considerations motivate the introduction of a bilinear neural net architecture. As illustrated in Fig.1, we can combine fully-connected layers and an element-wise product operator to embed a second-order polynomial representation for operator $\hat{F}$ in the proposed architecture. High-order polynomial representation might be embedded similarly. In Fig.1, we illustrate an architecture where operator $\hat{F}$ can be represented as the linear combination of three linear terms (i.e., linear combination of the input variables) and three bilinear terms (i.e., products between two linear combination of the input variables). In this architecture, the parameterization of the architecture initially relies on the definition of the number of linear and non-linear terms, which relate to the number of hidden nodes in the fully-connected layers, respectively FC_1 and $FC_{2,3}$. The calibration of the proposed architecture then comes to learning the weights of the different fully-connected layers. It may be noted that bilinear NN architectures have also been proposed in other contexts [12, 13].

2.3. Dynamical model interpretation

A huge advantage of using bilinear neural network as a building block of the RKNN introduced in [8] is the interpretability of our dynamical operator when used to identify models with only bilinear nonlinearities. While classical feedforward networks rely on non-linear activation functions to capture non-linear behavior in dynamical systems, making them difficult to be interpreted physically. The bilinear neural network can approach very efficiently systems like Lorenz-63 using only linear activation functions sins it's governed by bilinear nonlinearities. This configuration allows to learn a physically interpretable approximation as the output of the model

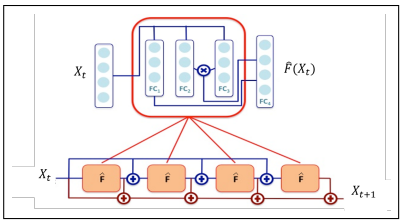

Fig. 1. ***Proposed bilinear residual architecture for the representation of a dynamical system represented by (1).*** We illustrate an architecture associated with a 4^{th}-order Runge-Kutta-like numerical integration for an elementary time-step $dt = 1$. It involves a four-layer residual neural net with an elementary network $\hat{F}$ repeated four times. The output of this elementary network involves a fully-connected layer FC_4 whose inputs are the concatenation of the output of the fully-connected layer FC_1 and the element-wise product between the outputs of fully-connected layers FC_2 and FC_3.

will be a linear combination of linear and bilinear combinations of the inputs.

2.4. Training issues

Given the proposed architecture and a selected parameterization, *i.e.* the number of nodes of the fully-connected layers $FC_{1,2,3}$ the number of $\hat{F}$ blocks, the learning of the model aims primarily to learn the weights of the fully-connected layers associated with block $\hat{F}$. As stated previously, coefficients $\{\alpha_i\}_i$ and $\{\beta_i\}_i$ from (2) may be set *a priori* or learned from the data. Given a dataset $\{X_{t_n}, X_{t_n+dt}\}_n$, corresponding to state time series for a given time resolution dt, the loss function used for training is the root mean square error of the forecasting at one time step dt. Given the relationship between the number of elementary blocks $\hat{F}$ in the considered architecture and the order of the underlying integration scheme, one may consider an incremental strategy, where we initially consider a one-block architecture, *i.e.* an explicit Euler integration scheme prior to increasing the number of $\hat{F}$-blocks for a higher-order numerical scheme.

Regarding initialization aspects, the weights of the fully-connected layers $FC_{1,2,3,4}$ are set randomly and coefficients $\{\alpha_i\}_i$ and $\{\beta_i\}_i$ are set to those of the associated Runge-Kutta scheme. We use Keras framework with Tensorflow backend to implement the proposed architecture. During the learning step, we impose a hard constraint that the different $\hat{F}$-blocks share the same parameters after each training epoch.

2.5. Application to forecasting and dynamical model identification

In this study, we first consider forecasting of the evolution of state X from a given initial condition X_{t_0}. For a trained NN architecture, two strategies may be considered : i) the use of the trained architecture as a recurrent neural net architecture to forecast a time series for a number of predefined time steps dt, ii) the plug-an-play use of the trained operator $\hat{F}$ in a classic ordinary differential equation solver. It may be noted that, for a trained operator $\hat{F}$, the fourth-order architecture sketched in Fig. 1 is numerically equivalent to a fourth-order Runge-Kutta solver.

We also explore the potential of the proposed NN representation for the identification of the underlying dynamical model with bilinear non-linearities. Given a time series $\{X_{t+kdt}\}_k$ that we assume is governed by an unknown ODE that only involves linear and bilinear terms. The approximate dynamical operator $\hat{F}$ will be constructed using linear and bilinear blocks feed into a fully connected layer with linear activation functions.

This architecture allows us to reconstruct the physical learned equations by propagating symbolic inputs through the learnt NN block $\hat{F}$.

3. NUMERICAL EXPERIMENTS

This section presents the numerical experiments we perform to demonstrate the relevance of the proposed bilinear NN architecture. We consider two categories of experiments : using synthetic datasets issued from the numerical integration of classic dynamical systems and using a satellite-derived SST (Sea Surface Temperature) dataset.

3.1. Synthetic case-studies

We first evaluate the proposed approach for 2 classic dynamical system, namely Lorenz-63 and Lorenz-96 dynamics, for which we generate time series exemplars from the numerical integration of the ODE which governs each system. We use the Shampine and Gordon solver [14].

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

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

Under parameterization $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$, Lorenz-63 system involves chaotic dynamics with two attractors. The integration time step dt is set to 0.01.

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

$$\frac{dX_{t,i}}{dt} = (X_{t,i+1} - X_{t,i-2})X_{t,i-1} + A \tag{4}$$

with periodic boundary conditions (*i.e.* $X_{t,-1} = X_{t,40}$ and $X_{t,41} = X_{t,1}$). Time step h is set to 0.05 and $A = 8$.

For each synthetic system, we generate a time series of 50000 time steps to create our training dataset and a time series of 1000 time steps for the test dataset.

In our experiments, we evaluate the forecasting performance as the Root Mean Square error (RMSE) for an integration time of h, $4h$ and $8h$, where h is the integration time step of the simulated time series. For benchmarking purposes, we compare the proposed bilinear residual NN representation to the following data-driven representation :

— a sparse regression model [3] referred to as SR. It combines an augmented bilinear state as regression variable and a sparsity-based regression ;

— an analog forecasting operator [6] referred to as AF. It applies locally-linear operators estimated from nearest neighbors, retrieved according to a Gaussian kernel as in [6] ;

Several NN representations are evaluated :

— the proposed bilinear residual architecture using a one-block version (Euler-like setting), referred to as Bi-NN(1), and a four-block version (Runge-Kutta-like setting) with shared layers, referred to as Bi-NN-SL(4). We use 3-dimensional (resp. 40-dimensional) fully-connected layers for the linear and bilinear layers $FC_{1,2,3}$ for Lorenz-63 system (resp. Lorenz-96 system).

— a neural network architecture similar to the above four-block one but replacing the proposed bilinear block by a classic MLP [8]. From cross-validation experiments, we consider a MLP with 5 hidden layers (resp. 11 hidden layers) and 6 nodes in each layer (resp. 80 nods in each layer) for Lorenz-63 model (resp. Lorenz-96 model). This architecture is referred to as MLP-SL(4) ;

— a MLP architecture trained to predict directly state at time $t + h$ from the state at time t. From cross-validation experiments, we consider a MLP with 5 hidden layers (resp. 10 hidden layers) and 6 nodes in each layer (resp. 80 nodes) for the Lorenz-63 model (resp. the Lorenz-96 model). This architecture is referred to as MLP.

Learning from noise-free training data : in this experiment, we compare the quality of the forecasted state trajectories generated using the models described above (see Tab.1). The learning of the data-driven models is carried using noise-free time series computed using the analytical dynamical models.

Model identification : we investigate model identification performance for Lorenz-63 dynamics in Tab.2. We report the performance in terms of model parameter estimation for the three data-driven schemes whose parameterization explicitly relates to the true physical equations, namely SR, Bi-NN(1) and Bi-NN(4)-SL. Bi-NN(4)-SL leads to a better estimation of model parameters, which explains better forecasting performance presented in Tab.1.

Table 1. ***Forecasting performance of data-driven models for Lorenz-63 and Lorenz-96 dynamics*** : mean RMSE for different forecasting time steps for the following models, AF (A), SR (B), MLP (C), MLP-SL(4) (D), Bi-NN(1) (E), Bi-NN-SL(4) (F). See the main text for details.

	A	B	C	D	E	F
Lorenz-63						
$t_0 + h$	0.001	0.002	0.114	0.009	0.002	**1.37E-5**
$t_0 + 4h$	0.004	0.008	0.172	0.035	0.006	**4.79E-5**
$t_0 + 8h$	0.007	0.014	0.197	0.071	0.013	**8.17E-5**
Lorenz-96						
$t_0 + h$	0.242	0.031	0.827	0.731	0.049	**0.012**
$t_0 + 4h$	0.580	0.086	1.623	1.870	0.140	**0.035**
$t_0 + 8h$	0.988	0.147	2.215	2.752	0.246	**0.064**

Table 2. ***MSE in the estimation of Lorenz-63 parameters for SR, Bi-NN(1) and Bi-NN(4)-SL models.*** See the main manuscript for details.

	Parameter value	SR	Bi-NN(1)	Bi-NN(4)-SL
σ	10	9.97	10.10	$\mathbf{10 \pm E - 4}$
ρ	28	$\mathbf{28 \pm E - 13}$	27.74	$28 \pm E - 4$
β	$8/3$	2.65	2.58	$\mathbf{2.667}$
MSE		0.0387	0.31	$\mathbf{4.6E - 4}$

3.2. SST case-study

We also evaluate the relevance of the proposed NN architecture for the forecasting and reconstruction of sea surface dynamics, and more particularly sea surface temperature. We consider an experimental setup similar to [15]. We use as SST data the OSTIA product [16] delivered by the UK met Office with a $0.05°$ spatial resolution from January 2008 to December 2015 with a temporal resolution $h = 1$ day. We here focus on the SST anomaly below 100km using a Gaussian filtering to remove the large-scale component. As case-study region, we consider a region off south Africa located on longitude $5°E$ to $75°E$ and latitude $25°S$ to $55°S$.

Our goal is to model and SST anomaly time series using the proposed bilinear NN architecture. Following [15], we adopt a patch-based representation with 20×20 patches combined with a patch-level PCA decomposition so that each patch is represented by a 50-dimensional vector. This PCA decomposition accounts for 95% of the total patch-level variance. Overall, the proposed bilinear NN architecture is applied to 50-dimensional time series. The data from 2008 to 2014 were used as training data and we tested our approach on the 2015 data.

We report forecasting performance on two patches which involves complex ocean dynamics in Tab.3. Similarly to the experiments with synthetic data, the proposed bilinear residual NN architecture (60-dimensional fully-connected layers for the layers $FC_{1,2,3}$) outperforms both MLP architectures

(10 hidden layers and 50 nodes in each layer), the sparse regression model and locally-linear analog operators. We performed additional experiments (not included due to the page limit) for the interpolation of missing data in satellite-derived SST image series [15]. Using the learnt NN model as a dynamical operator in a Kalman-based assimilation scheme [15], these experiments led to similar conclusions with a significant improvement of the interpolation performance w.r.t. the best competing methods (up to 40% in terms of MSE).

Table 3. ***Forecasting performance of data-driven models for SST anomaly dynamics*** : mean RMSE for different forecasting time steps for the following models, AF (A), SR (B), MLP (C), MLP-SL(4) (D), Bi-NN(1) (E), Bi-NN-SL(4) (F).

	A	B	C	D	E	F
Patch 1						
$t_0 + h$	1.58	1.15	1.73	1.08	1.11	**1.08**
$t_0 + 2h$	2.08	1.75	2.27	1.58	1.62	**1.56**
$t_0 + 3h$	2.92	2.69	3.08	2.41	2.41	**2.34**
Patch 2						
$t_0 + h$	0.75	0.52	1.24	0.37	0.36	**0.36**
$t_0 + 2h$	1.03	0.93	1.26	0.59	0.51	**0.51**
$t_0 + 3h$	1.67	1.49	1.34	0.89	0.74	**0.74**

4. CONCLUSION

In this work, we demonstrated the relevance of a residual bilinear NN representation for the modeling and identification of geophysical dynamics. Our NN-based representation relies on the representation of classic numerical schemes of differential equations as a multi-layer recurrent network. Importantly, it embeds bilinear nonlinearities which are key features of geophysical dynamics. We demonstrated the relevance of the proposed NN representation in terms of identification and forecasting performance for both synthetic and real datasets. Such NN representation opens new research avenues for the exploitation of machine-learning-based and physically-sound strategies for the modeling, identification and reconstruction of geophysical systems.

5. REFERENCES

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

[2] R. Lguensat, P. Huynh Viet, M. Sun, G. Chen, T. Fenglin, B. Chapron, and R. Fablet, "Data-driven Interpolation of Sea Level Anomalies using Analog Data Assimilation," Tech. Rep., Oct. 2017.

[3] S. L. Brunton, J. L. Proctor, and J. N. 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.

[4] J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens, and R. Pintelon, "Identification of nonlinear systems using Polynomial Nonlinear State Space models," *Automatica*, vol. 46, no. 4, pp. 647–656, Apr. 2010.

[5] Z. Zhao and D. Giannakis, "Analog Forecasting with Dynamics-Adapted Kernels," *arXiv :1412.3831 [physics]*, Dec. 2014, arXiv : 1412.3831.

[6] R. Lguensat, P. Tandeo, P. Aillot, and R. Fablet, "The Analog Data Assimilation," *Monthly Weather Review*, 2017.

[7] D. C. Park and Yan Zhu, "Bilinear recurrent neural network," in *, 1994 IEEE International Conference on Neural Networks, 1994. IEEE World Congress on Computational Intelligence*, June 1994, vol. 3, pp. 1459–1464 vol.3.

[8] 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.

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

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[11] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, "Model selection for dynamical systems via sparse regression and information criteria," *Proc. R. Soc. A*, vol. 473, no. 2204, pp. 20170009, Aug. 2017.

[12] T. Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN Models for Fine-Grained Visual Recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1449–1457.

[13] D.C. Park and T.-K. Jeong, "Complex-bilinear recurrent neural network for equalization of a digital satellite channel," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 711–725, May 2002.

[14] R. Ashino, M. Nagase, and R. Vaillancourt, "Behind and beyond the Matlab ODE suite," *Computers & Mathematics with Applications*, vol. 40, no. 4, pp. 491–512, Aug. 2000.

[15] R. Fablet, P. H. Viet, and R. Lguensat, "Data-driven Models for the Spatio-Temporal Interpolation of satellite-derived SST Fields," *IEEE Transactions on Computational Imaging*, 2017.

[16] C. J. Donlon, M. Martin, J. Stark, J. Roberts-Jones, E. Fiedler, and W. Xindong, "The Operational Sea Surface Temperature and Sea Ice Analysis (OSTIA) system," *Remote Sensing of Environment*, vol. 116, pp. 140–158, Jan. 2012.