A toy model to investigate stability of AI-based dynamical systems

Blanka Balogh, David Saint-Martin, Aurélien Ribes

CNRM, Université de Toulouse, Météo-France, CNRS, Toulouse, France.
blanka.balogh@meteo.fr

April 30, 2021

Context – 1/3

In an atmospheric model (aimed at deriving tendencies for atmospheric state variables $\mathbf{x}$):

$$\frac{\partial \mathbf{x}}{\partial t} = \mathcal{P}(\mathbf{x}) + \mathcal{D}(\mathbf{x})$$

Where $\mathcal{D}$ is the dynamics (resolved processes) of the model, $\mathcal{P}$ are the physical parameterizations (unresolved scale processes).

The machine learning (ML) problem of learning physical parameterizations can be expressed:

$$f(\mathbf{x}) = \mathcal{P}(\mathbf{x}) \equiv \dot{\mathbf{x}}_{\mathcal{P}} \quad (1)$$

Context – 2/3

Once implemented inside the model, most of current AI-based parameterizations faces numerical instabilities. After a few integrations, the model explodes.

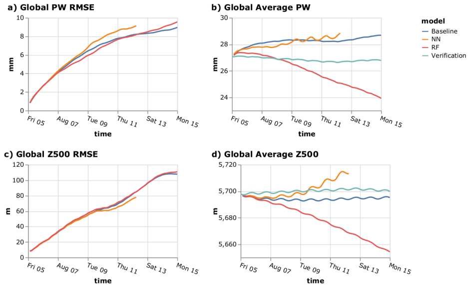


Figure 2: Online skill of the NN and RF. Compares the forecast accuracy in terms of RMSE for precipitable water (a) and 500 mb geo-potential height (c). The respective global averages are shown in (b) and (d).

Figure 1: An example of divergence with an AI-based parameterization scheme. Brenowitz et al., 2020 [3]

Context – 3/3

Currently, no toy model is able to replicate such numerical instability issues. In particular, Lorenz toy models are not complex enough to manifest any instability.

Stephan Rasp

Senior data scientist @ClimateAi

- Munich, Germany
- Email
- Twitter
- GitHub
- Researchgate
- Google Scholar
- LinkedIn

The need for a more realistic toy model

Parameterizing sub-grid processes is the big challenge for climate modeling. Modern machine learning approaches could be one way to make real progress. Over the last two years, first studies (here and here and here) have demonstrated that it is generally feasible to build a ML parameterization. But, as I've summarized in my recent paper on online learning, there are several fundamental obstacles to overcome before ML parameterizations can actually improve weather and climate predictions. Stability, physical consistency and tuning are just some of them.

[...]

All of these challenges cry out for an intermediate step between L96 and full-complexity models.

https://raspstephan.github.io/blog/lorenz-96-is-too-easy/

Objectives

Objectives.

1. Development of an extended version of Lorenz'63 (hereafter, L63) toy model replicating instability issues when system dynamics are learnt with neural networks (hereafter, NN) ;
2. Understanding the origin of this instability + solution to address it.

References

[1] Rasp, S.: Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: general algorithms and Lorenz 96 case study (v1.0), Geosci. Model Dev., 13, 2185–2196, https://doi.org/10.5194/gmd-13-2185-2020, 2020.

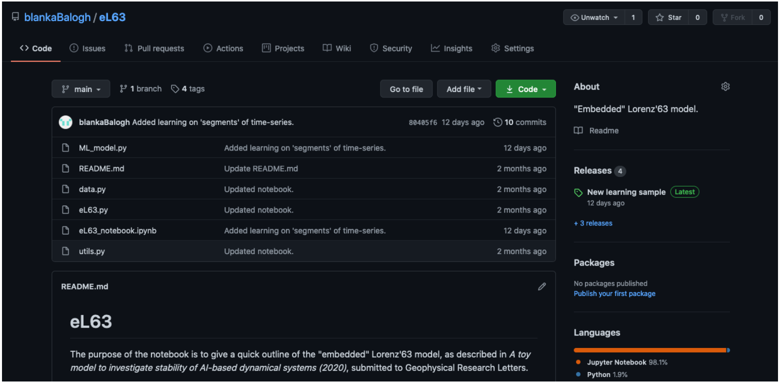
[2] Brenowitz, N. D., Beucler, T., Pritchard, M., Bretherton, C. S. (2020). Interpreting and Stabilizing Machine-Learning Parametrizations of Convection, Journal of the Atmospheric Sciences, 77(12), 4357-4375. Retrieved Feb 24, 2021, from https://journals.ametsoc.org/view/journals/atsc/77/12/jas-d-20-0082.1.xml

[3] Brenowitz, N. D., Henn, B., McGibbon, J., Clark, S. K., Kwa, A., Perkins, W. A., Watt-Meyer, O., Bretherton, C. S. (2020). Machine Learning Climate Model Dynamics: Offline versus Online Performance. https://arxiv.org/abs/2011.03081

[4] Balogh, B., Saint-Martin, D., Ribes, A. (2021). A toy model to investigate stability of AI-based dynamical systems. Geophysical Research Letters, 48, e2020GL092133. https://doi.org/10.1029/2020GL092133

Code & notebook

Code & notebook: https://github.com/blankaBalogh/eL63.



Learning L63 – 1/3

Objective. Learning L63 dynamics with NNs.

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1), \\ \dot{x}_2 &= x_1(\rho - x_3) - x_2, \\ \dot{x}_3 &= x_1 x_2 - \beta x_3 \end{aligned} \tag{2}$$

With (σ, ρ, β) fixed, the ML problem of learning L63 can be written:

$$\dot{\mathbf{x}} = f(\mathbf{x}) \tag{3}$$

In the following, (σ, ρ, β) values are set to $(10, 28, 8/3)$.

Learning L63 – 2/3

Generating an 'orbit' of length 500 Model Time Units (MTU ; 1 MTU $= 20 \times \mathrm{d}t$ where $\mathrm{d}t = 0.05$) or 10^4 integrations

Building the learning sample:

- Timeseries of L63 state variables: $\mathbf{x}(t)$
- Associated time derivatives: $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t))$

In the following, a learning sample will be noted:

$$[\mathbf{x}] = \{(\mathbf{x}_n, f(\mathbf{x}_n))\}_{n=1,\ldots,N}$$

with $\mathbf{x}_n = \mathbf{x}(t_n)$.

Learning L63 – 3/3

Learning of (3) is performed with a very simple NN; the resulting function is $\widehat{f}$ such that:

$$\widehat{\dot{\mathbf{x}}} = \widehat{f}(\mathbf{x}) \qquad (4)$$

Loss function: mean squared error (MSE).
R^2 score over test dataset: 0.999

Model: "functional_1"

Layer (type)	Output Shape	Param #
X_data (InputLayer)	[(None, 3)]	0
dense (Dense)	(None, 256)	1024
dense_1 (Dense)	(None, 128)	32896
dense_2 (Dense)	(None, 128)	16512
dense_3 (Dense)	(None, 128)	16512
dense_4 (Dense)	(None, 128)	16512
dense_5 (Dense)	(None, 128)	16512
dense_6 (Dense)	(None, 64)	8256
dense_7 (Dense)	(None, 32)	2080
predictions (Dense)	(None, 3)	99

Total params: 110,403
Trainable params: 110,403
Non-trainable params: 0

Validation strategy – 1/2

The validation strategy consists in generating long orbits of length 1000 MTU or 20.10^3 integrations, by integrating $\widehat{f}$. Validation orbits generated with the trained NN-model will then be compared with the direct integration of L63 equations.

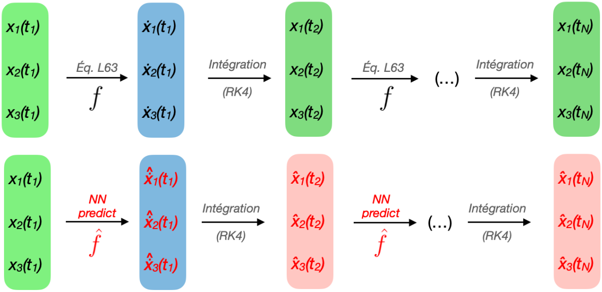


Figure 2: Validation strategy.

Validation strategy – 2/2

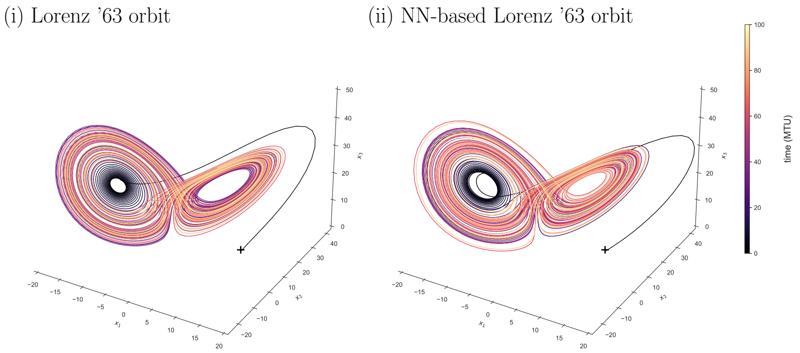


Figure 3: Validation orbits resulting from the integration of: (i) L63 equations and (ii) of the NN model that learnt eq. (3). Only the first 100 MTUs are represented.

Conclusion – Learning L63

Fit to learning sample is great (see validation R2 score), but the resulting NN model does not allow the study of instabilities.

Need of a more complex toy model to study instabilities encountered when developing NN-based parameterizations.
How to conceive a higher dimension ($d > 3$) version of L63 ?

'embedded' Lorenz'63 model (eL63) – 1/2

Objective. Extend L63 to dimension $d > 3$ to create a toy model replicating numerical instabilities.

1. $\mathbf{z} \in \mathcal{B}_z$: dynamics are easy to express explicitly.

$$\begin{aligned} \dot{z}_1 &= \sigma(z_2 - z_1), \\ \dot{z}_2 &= z_1(\rho - z_3) - z_2, \\ \dot{z}_3 &= z_1 z_2 - \beta z_3, \\ \dot{z}_j &= -\kappa z_j, \quad \forall j > 3. \end{aligned} \tag{5}$$

   For sake of simplicity, $\kappa = 1$ hereafter.

2. $\mathbf{x} \in \mathcal{B}_x$ ('learning' space) after random rotation P

$$\mathbf{x} = P\mathbf{z} \tag{6}$$

'embedded' Lorenz'63 model (eL63) – 2/2

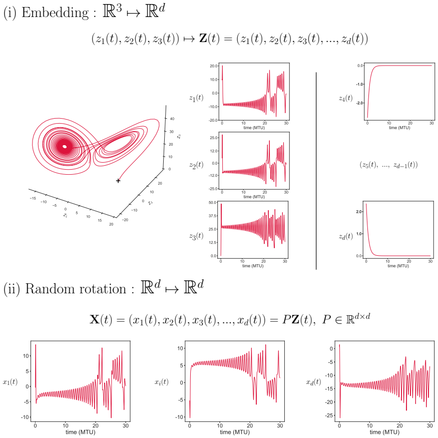


Figure 4: 'embedding' of L63 model. The NN will only see variables from $\mathcal{B}_x$, after applying 'hidden constraint' P.

Learning eL63

- Learning is performed in $\mathcal{B}_x$, in the same configuration than it has been presented for 'original' L63 model (e.g., NN architecture, metrics, learning sample size). The resulting NN-dynamics function is noted $\widehat{f}_{orb}$
- Validation R^2 score: > 0.999.

Stability criterion.
A validation orbit is considered 'stable' if:
$\forall\ 1 \leq i \leq d,\ \forall\ 1 \leq n \leq N \equiv 1000$ MTU,

$$m_i - 3(M_i - m_i) \leq x_i^n \leq M_i + 3(M_i - m_i) \qquad (8)$$

where m_i, M_i maximal and minimal values of variables at coordinate i in the validation orbit.

Evaluation of the NN model – 1/3

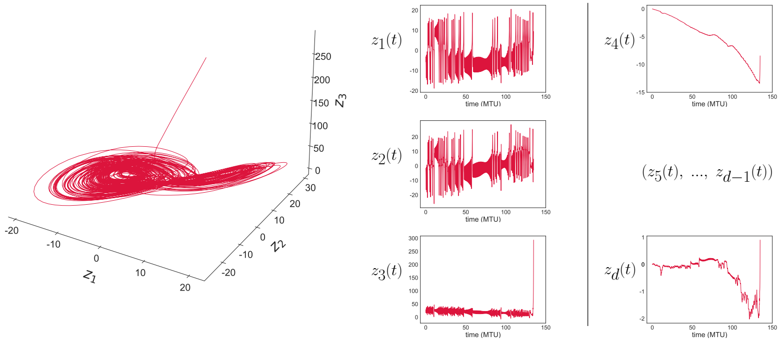


Figure 5: Example of a diverging validation orbit, generated by integration of tendencies from $\widehat{f}_{orb}$. The subsequent NN model of eL63 will be assessed in $\mathcal{B}_z$, facilitating comparison without the random rotation P.

Evaluation of the NN model – 2/3

'Global' stability of the model is assessed, with dimension d fixed, as described below:

1. Train 100 different NNs (= 100 different rotation matrices P): 100 $\widehat{f}_{orb}$ models.
2. With each $\widehat{f}_{orb}$ model, generate validation orbits of length 1000 MTUs from 30 different initial conditions.
3. Apply the stability criterion (see previous slide) to assess stability over $100 \times 30 = 3000$ validation orbits.

Evaluation of the NN model – 3/3

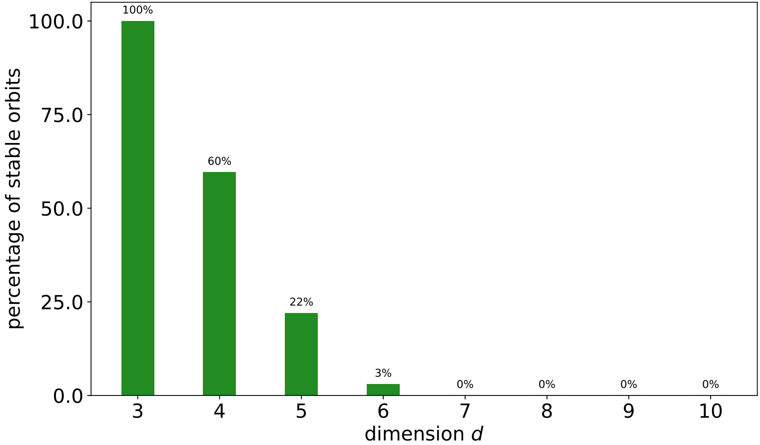


Figure 6: Percentage of stable validation orbits as a function of the embedding dimension d. Validation orbits were generated following the steps described previously.

Conclusions – 'Embedded' L63 model

1. Even with minimal embedding (e.g., $d = 4$), 40% of NN-generated eL63 validation orbits are unstable (or exploding).
2. Stability decays quickly when d increases.
3. With $d > 6$, none of the validation orbits are stable.

eL63 is an extended version of L63, which allows the replication of numerical instabilities when learned by (feedforward) NNs upon a single orbit.

How to make NN models of eL63 stable ?

A new learning sample ? – 1/2

$\widehat{f}_{orb}$ is not precise enough in some regions of the eL63 phase space: could the instability be a **problem of sampling** ? Is it possible to prevent instabilities by designing a specific learning sample ?

A new learning sample ? – 2/2

The new learning sample is built with Latin Hypercube Sampling (LHS). **LHS sampling is performed in $\mathcal{B}_x$.**

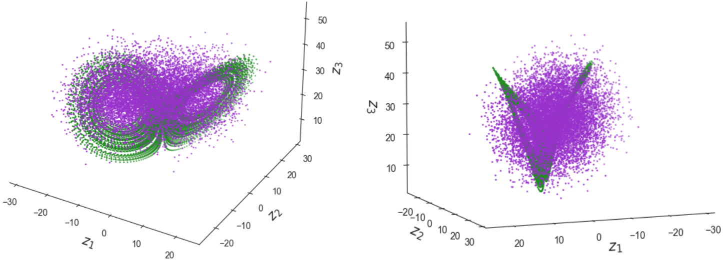


Figure 7: The first 3 components of the state variable **z** from the 'orbital' learning sample (green) and the 'LHS' learning sample (purple). Both samples contains 10^4 individuals.

Fitting NN to the LHS sample

- NN is fitted to the LHS learning sample in the same configuration as previously described (e.g., same NN architecture, same metrics). The size of the LHS learning sample matches those of the orbital learning sample. However, train/test split is done on a random partition basis.
- The resulting NN model is noted $\widehat{f}_{\mathrm{LHS}}$.
- Validation R^2: $\approx 0.997...$
  fit is slightly less accurate than in the previous cases.

Validation

Validation. 100 $\widehat{f}_{\mathrm{LHS}}$, 30 random initial conditions.

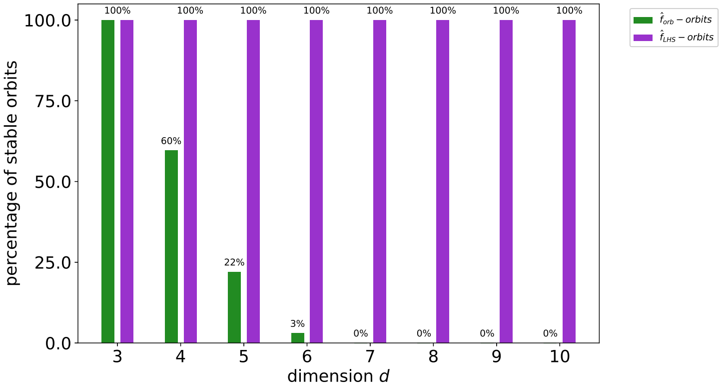


Figure 8: Percentage of stable validation orbits, generated with $\widehat{f}_{orb}$ (green) and $\widehat{f}_{\mathrm{LHS}}$ (purple). NN model is stabilized when fitted to the LHS learning sample (regarding the stability criterion described above).

Conclusion

- L63 model is not complex enough to manifest numerical instabilities when its dynamics is learnt with NNs.
- 'embedded' L63 model, an extended version of L63 to $d > 3$ succeeds in replicating numerical instability issues, when the NN model is fitted to a learning sample consisting in a single orbit.
- The NN model is stabilised (with $d < 11$ at least) when fitted to a specifically designed learning sample. This learning sample was built with LHS.

An LHS sampling is difficult to perform in the case of climate models. This study underlines how much the learning sample is important to grant stability to NN models.