Deep Learning in Multi-step Forecasting of Chaotic Dynamics

Matteo Sangiorgio

Abstract The prediction of chaotic dynamical systems' future evolution is widely debated and represents a hot topic in the context of nonlinear time series analysis. Recent advances in the field proved that machine learning techniques, and in particular artificial neural networks, are well suited to deal with this problem. The current state-of-the-art primarily focuses on noise-free time series, an ideal situation that never occurs in real-world applications. This chapter provides a comprehensive analysis that aims at bridging the gap between the deterministic dynamics generated by archetypal chaotic systems, and the real-world time series. We also deeply explore the importance of different typologies of noise, namely observation and structural noise. Artificial intelligence techniques turned out to provide robust predictions, and potentially represent an effective and flexible alternative to the traditional physically-based approach for real-world applications. Besides the accuracy of the forecasting, the domain-adaptation analysis attested the high generalization capability of the neural predictors across a relatively heterogeneous spatial domain.

1 Introduction

Machine learning techniques are nowadays widely used in time series analysis and forecasting, especially in those characterized by complex nonlinear behaviors. A classical example are the meteorological processes, whose nonlinearity often generates chaotic trajectories. In such context, the machine learning algorithms proved to outperform the traditional methodologies, mainly relying on linear modelling techniques [1, 2].

Artificial neural networks are by far the most widespread technique used for time series prediction. Two different neural architectures can be adopted in this context. The first are the feed-forward (FF) neural networks, so-called because their structures are described by an acyclic graph (i.e., without self-loops). FF architectures are thus

M. Sangiorgio (✉)

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, 20133, Milano, Italy

e-mail: matteo.sangiorgio@polimi.it

© The Author(s) 2022

L. Piroddi (ed.), *Special Topics in Information Technology*,

PoliMI SpringerBriefs, https://doi.org/10.1007/978-3-030-85918-3_1

static and different strategies can be used to adapt them to the intrinsically dynamical nature of time series. A second alternative is represented by recurrent neural networks (RNNs), so-called because of the presence of self-loops that make them dynamical models. This feature should make, at least in principle, the RNNs naturally suited for sequential tasks as time series forecasting.

In this chapter, we investigate the predictive accuracy of different purely autoregressive neural approaches with both FF and recurrent structures. Our analysis takes into account various chaotic systems, spanning from the archetypal examples of deterministic chaos to two real-world time series of ozone concentration and solar irradiance. To quantify the effect of different noise typologies on the forecasting capabilities of the neural predictors, we set up two numerical experiments by perturbing the deterministic dynamics of the archetypal chaotic systems with artificially-generated noise.

Finally, we extend the idea of domain adaptation to time series forecasting: the neural predictors trained to forecast the solar irradiance on a given location (the source domain) are then used, without retraining, as predictors for the same variable in other locations (the target domains).

The rest of this chapter is organized as follows. Section 2 introduces the feed-forward and recurrent neural predictors, describing their structures and their identification procedures. Section 3 reports the results obtained in a deterministic environment. In Sect. 4, the impact of different noise typologies on the forecasting accuracy is evaluated. Section 5 presents two real-world applications. Finally, in Sect. 6, some concluding remarks are drawn.

2 Neural Predictors for Time Series

2.1 Predictors' Identification

The forecasting of a time series is a typical example of supervised learning task: the training is performed by optimizing a suitable metric (the loss function), that assesses the distance between N target samples, $\mathbf{y} = [y(1), y(2), \dots, y(N)]$, and the corresponding predictions, forecasted i steps ahead, $\hat{\mathbf{y}}^{(i)} = [\hat{y}(1)^{(i)}, \hat{y}(2)^{(i)}, \dots, \hat{y}(N)^{(i)}]$. A widely used loss function is the mean squared error (MSE), that can be computed for the i^{th} step ahead:

$$\text{MSE}(\mathbf{y}, \hat{\mathbf{y}}^{(i)}) = \frac{1}{N} \sum_{t=1}^N \left(y(t) - \hat{y}(t)^{(i)} \right)^2. \quad (1)$$

One-step predictors are optimized by minimizing $\text{MSE}(\mathbf{y}, \hat{\mathbf{y}}^{(1)})$. Conversely, a multi-step predictors can be directly trained on the loss function computed on the entire h -step horizon:

$$\frac{1}{h} \sum_{i=1}^h \text{MSE}(\mathbf{y}, \hat{\mathbf{y}}^{(i)}). \quad (2)$$

Following the classical neural nets learning framework, the training is performed for each combination of the hyper-parameters (mini-batch size, learning rate and decay factor), selecting the best performing on the validation dataset. The hyper-parameters specifying the neural structure (i.e., the number of hidden layers and of neurons per layer) are fixed: 3 hidden layers of 10 neurons each. Once the predictor has been selected, it is evaluated on the test dataset (never used in the previous phases) in order to have a fair performance assessment and to avoid overfitting on both the training and the validation datasets.

A well-known issue with the MSE is that the value it assumes does not give any general insight about the goodness of the forecasting. To overcome this inconvenient, the R^2 -score is usually adopted in the test phase:

$$R^2(\mathbf{y}, \hat{\mathbf{y}}^{(i)}) = 1 - \frac{\text{MSE}(\mathbf{y}, \hat{\mathbf{y}}^{(i)})}{\text{MSE}(\mathbf{y}, \bar{\mathbf{y}})}. \quad (3)$$

Note that $\bar{\mathbf{y}}$ is the average of the data, and thus the denominator corresponds to the variance, $\text{var}(\mathbf{y})$. For this reason, the R^2 -score can be seen as a normalized version of the MSE. It is equal to 1 in the case of a perfect forecasting, while a value equal to 0 indicates that the performance is equivalent to that of a trivial model always predicting the mean value of the data. An R^2 -score of -1 reveals that the target and predicted sequences are two trajectories with the same statistical properties (they move within the same chaotic attractor) but not correlated [3, 4]. In other words, the predictor would be able to reproduce the actual attractor, but the timing of the forecasting is completely wrong (in this case, we would say that we can reproduce the long-term behavior or the climate of the attractor [5, 6]).

2.2 Feed-Forward Neural Networks

The easiest approach to adapt a static FF architecture to time series forecasting consists in identifying a one-step predictor, and to use it recursively (FF-recursive predictor). Its learning problem requires to minimize the MSE in Eq. (1) with $i = 1$ (Fig. 1, left), meaning that only observed values are used in input.

Once the one-step predictor has been trained, it can be used in inference mode to forecast a multi-step horizon (of h steps) by feeding the predicted output as input for the following step (Fig. 1, right).

Alternative approaches based on FF networks are the so-called FF-multi-output and FF-multi-model. In the multi-output approach, the network has h neurons in the output layer, each one performing the forecasting at a certain time step of the horizon. The multi-model approach requires to identify h predictors, each one specifically

Fig. 1 FF-recursive predictor in training (**left**) and inference (**right**) modes. For the sake of simplicity, we considered $d = 2$ lags and $h = 2$ leads, and a neural architecture with 2 hidden layers of 3 neurons each

Fig. 2 RNNs trained with (**left**) and without (**right**) teacher forcing. Note that both LSTM-TF and LSTM-no-TF follow the scheme in panel b in inference mode, because at time t , the actual value at $t + 1$ is not available. For the sake of simplicity, we considered $d = 2$ lags and $h = 2$ leads, and a neural architecture with a single hidden layers of 3 cells

trained for a given time step. In this chapter, we limit the analysis to the FF-recursive predictor. A broad exploration including the other FF-based approaches can be found in the literature [7, 8].

2.3 Recurrent Neural Networks

Recurrent architectures are naturally suited for sequential tasks as time series forecasting and allow to explicitly take into account temporal dynamics. In this work, we make use of RNNs formed by long short-term memory (LSTM) cells, which have been successfully employed in a wide range of sequential task, from natural language processing to policy identification.

When dealing with multi-step forecasting, the RNNs are usually trained with a technique known as teacher forcing (TF). It requires to always feed the actual values, instead of the predictions computed at the previous steps, as input (Fig. 2, left).

TF proved to be necessary in almost all the natural language processing tasks, since it guides the optimization ensuring the convergence to a suitable parameterization. For this reason, it became the standard technique implemented in all the deep learning

libraries. TF simplifies the optimization at the cost of making training and inference modes different. In practice, at time t , we cannot use the future values $y(t + 1)$, $y(t + 2)$, $\dots$ because such values are unknown and must be replaced with their predictions $\hat{y}(t + 1)$, $\hat{y}(t + 2)$, $\dots$ (Fig. 2, right).

This discrepancy between training and inference phases is known under the name of exposure bias in the machine learning literature [9]. The main issue with TF is that, even if we optimize the average MSE over the h -step horizon in Eq. (2), we are not really doing a multi-step forecasting (we can say that it is a sequence of h single-step forecasting, similarly to what happens in FF-recursive). In other words, the predictor is not specifically trained on a multi-step task.

We thus propose to train the LSTM without TF (LSTM-no-TF) so that the predictor's behavior in training and inference coincides (Fig. 2, right).

3 Forecasting Deterministic Chaos

We initially consider the forecasting of noise-free dynamics derived from some archetypal chaotic systems. The first is the logistic map, a one-dimensional system traditionally used to model the population dynamics:

$$y(t + 1) = r \cdot y(t) \cdot (1 - y(t)), \quad (4)$$

where the parameter r represents the growth rate at low density. We then consider the Hénon map in its generalized m -dimensional version:

$$y(t + 1) = 1 - a \cdot y(t - m + 2)^2 + b \cdot y(t - m + 1). \quad (5)$$

Two versions of the Hénon map with m equal to 2 and 10 are implemented. As shown in Eqs. (4) and (5), these systems can be easily rewritten as nonlinear autoregressions with m lags.

Figure 3 reports the accuracy of the predictors in terms of R^2 -score, with the predictive horizon also rescaled in terms of Lyapunov times, i.e. the inverse of the largest Lyapunov exponent measuring the system's chaoticity (horizontal axis on top). This allows to standardize the performances taking into account the chaoticity of the considered chaotic map.

The LSTM-no-TF predictor provides the best performance (it forecasts with R^2 -score higher than 0.9 for 6–7 Lyapunov times), followed by the LSTM-TF and by the FF-recursive. Even if the ranking is the same in all the systems, the distance between the three predictors' performances is case-specific. In particular, the LSTM-TF has almost the same accuracy of LSTM-no-TF for the logistic map, and becomes closer to the FF-recursive predictor for the 10D Hénon. In this system, the LSTM-TF also shows a counterintuitive trend: its accuracy is not monotonically decreasing for increasing values of the lead time as one would expect. This is related to the training

Fig. 3 R^2 -score trends across the multi-step horizon for the three considered chaotic systems in a noise-free environment

with TF, which does not allow the predictor to properly propagate the information across time (the issue can be solved by training without TF).

4 Evaluating the Effect of Noise

This chapter extends the analysis to noisy and time-varying environments representing one of the first works, together with [5, 10, 11], that tries to bridge the gap between the ideal deterministic case and the practical applications.

4.1 Observation Noise

The type of noise usually taken into account is that originated by the uncertainty on the observations. Observation noise can be modeled as an additive white disturbance sampled from a gaussian distribution with a null average. Two levels of noise (i.e., two values of the noise standard deviation) are considered: 0.5% and 5% of the noise-free process standard deviation, respectively.

The results of forecasting these noisy time series are reported in Fig. 4 consistently with the deterministic case in Fig. 3. The ranking obtained in both the noise levels is the same as the noise-free one. The most performing predictor, LSTM-no-TF,

Fig. 4 R^2 -score trends across the multi-step horizon for the three chaotic systems. Two different levels of observation noise (0.5 and 5%) are considered

provides a high accuracy (R^2 -score > 0.9) for about 3 and 1 Lyapunov times with noise levels of 0.5% and 5%, respectively. When the real system is used as predictor, it provides a perfect forecasting in the noise-free case (see Fig. 3), but its accuracy is almost identical to that of the FF-recursive and LSTM-TF predictors in a noisy environment.

Figure 4 also shows that the predictors have two distinct behaviors in the final part of the forecasting horizon. The R^2 -scores of FF-recursive and LSTM-TF predictors tend to -1 , meaning that the neural networks reproduce fairly well the chaotic attractors' climate. This happens because these two predictors are trained on a single step

Fig. 5 Dataset obtained from the simulation of the logistic map affected by structural noise

Fig. 6 R^2 -score trends across the multi-step horizon for the logistic map affected by structural noise

ahead, and thus focus on system identification. The fact that the real systems used as predictors tend to -1 confirms that the two predictors correctly reproduce the chaotic systems' regime dynamics. Conversely, the LSTM-no-TF predictor's R^2 -scores tend to 0 because it specifically focuses on the multi-step forecasting. When it reaches its predictability limit, it becomes a trivial predictor always forecasting the dataset average value till the end of the h -step horizon.

4.2 Structural Noise

Another type of noise which frequently affects the real-world processes is the so-called structural noise, i.e., the presence of underlying dynamics that make the considered system non-stationary. We implement the structural noise by periodically varying the logistic parameter r in Eq.(4). As it happens with the annual periodicity of the meteorological variables, the variation of r is much slower than that of y . The resulting slow-fast dynamics showed in Fig.5 assumes stable, periodic, and chaotic behaviors, thus representing a challenging testing ground for the neural predictors.

Figure6 reports the forecasting accuracy obtained with the three predictors. The wide gap between the two LSTM nets and the FF-recursive predictor indicates that a recurrent structure is more appropriate than a static one for the considered non-stationary task. Once again, a training without TF further increases the predictive accuracy of the LSTM predictor.

Fig. 7 R^2 -score trends across the 48-h horizon for the ozone concentration (left) and solar irradiance (right) datasets

5 Real-World Applications

In the end, we evaluated the performances of the three neural predictors on two real-world time series, which are affected by both observation and structural noise (i.e., the uncertainty on the measurement and the presence of yearly and daily periodic dynamics).

5.1 Ozone Concentration

The first practical application considers a dataset of ground-level ozone concentration recorded in Chiavenna (Italy). The time series covers the period from 2008 to 2017 and is sampled at an hourly time step. The physical and chemical processes which generates the tropospheric ozone are strongly nonlinear and its dynamics is chaotic, as demonstrated by the positive value we estimated for the largest Lyapunov exponent (0.057).

The results obtained on a 48-h predictive horizon are reported in Fig. 7 (left). The predictive accuracy of the three predictors is almost identical in the first 6h. After that, a relevant gap emerges confirming the rank of the previous numerical experiments: LSTM-no-TF ensures the best performance, followed by the LSTM-TF and the FF-recursive predictors.

5.2 Solar Irradiance

The second real-world case study is an hourly time series of solar irradiance, recorded at Como (Italy) from 2014 to 2019. Its largest Lyapunov exponent is equal to 0.062, indicating that also its dynamic is chaotic.

Fig. 8 R^2 -score trends across the 48-h horizon obtained with the LSTM-no-TF predictor trained on the Como dataset. Source domain). Casatenovo, Bigarello, and Bema are the target domains

The R^2 -scores of the predictors reported in Fig. 7 (right) show that the FF-recursive predictor provides almost the same accuracy of the LSTM-no-TF, especially in the first 24 h of the horizon. Unlike in the other cases, the LSTM-TF predictor performance is so low that it is practically useless after few hours ahead (for comparison, the clear-sky model has R^2 -score equal to 0.58).

We also use the solar irradiance dataset to test the generalization capability of the neural predictors in terms of domain adaptation. The LSTM-no-TF predictor trained on the data recorded at the Como station, is used, without retraining, to forecast the solar irradiance in three other sites with quite heterogeneous geographical conditions and in different years: Casatenovo (2011), Bigarello (2016), and Bema (2017). Figure 8 reports the results of the domain-adaptation analysis. The R^2 -scores obtained are almost identical in the four domains (note that the vertical axis has been zoomed to spot the small differences between the four trends). This is an interesting result since it tells us that we can identify a neural predictor with a (hopefully large) dataset recorded at a certain location, and then use it in another location where an appropriate dataset for training is not available.

6 Conclusion

We tackled the problem of chaotic dynamics forecasting by means of artificial neural networks, implementing both feed-forward and recurrent architectures. A wide range of numerical experiments have been performed. First, we forecasted the output of some archetypal chaotic processes in a deterministic environment. The analysis is then extended to noisy dynamics, taking into account both observation (white additive disturbance) and structural (non-stationarity) noise. Finally, we considered two real-world time series of ozone concentration and solar irradiance, which have chaotic behaviors.

Whatever the system or the type of noise, our results showed that LSTM-no-TF is the most performing multi-step predictor. The distance in the performances obtained with LSTM-no-TF and the other predictors (FF-recursive and LSTM-TF) is essentially task dependent.

The better predictive power of the LSTM-no-TF predictor is due to the fact that it is specifically trained for a multi-step forecasting task. Conversely, the FF-recursive

and LSTM-TF predictors are optimized solving a system identification task. They are somehow more generic models, suitable for the mid-short-term forecasting and also able to replicate the long-term climate of the chaotic attractor (they can be used, for instance, to perform statistical analyses and for synthetic time series generation).

Besides the accuracy of the forecasting, the domain-adaptation analysis attested the high generalization power of the neural predictors across a relatively heterogeneous domain. The techniques presented in this chapter are specifically developed for multi-step forecasting. Therefore, they could be particularly interesting in the context of receding-horizon control schemes such as model predictive control.

Acknowledgements The author would like to thank Prof. Giorgio Guariso and Prof. Fabio Dercole that supervised this research and offered deep insights into this study.

References

1. M. Sangiorgio, S. Barindelli, R. Biondi, E. Solazzo, E. Realini, G. Venuti, G. Guariso, Improved extreme rainfall events forecasting using neural networks and water vapor measures, in *6th International conference on Time Series and Forecasting*, pp. 820–826 (2019)
2. M. Sangiorgio, S. Barindelli, V. Guglieri, R. Biondi, E. Solazzo, E. Realini, G. Venuti, G. Guariso, A comparative study on machine learning techniques for intense convective rainfall events forecasting, in *Theory and Applications of Time Series Analysis* (Cham, 2020. Springer), pp. 305–317
3. F. Dercole, M. Sangiorgio, Y. Schmirander, An empirical assessment of the universality of ANNs to predict oscillatory time series. *IFAC-PapersOnLine* **53**(2), 1255–1260 (2020)
4. M. Sangiorgio, Deep learning in multi-step forecasting of chaotic dynamics. Ph.D. thesis, Department of Electronics, Information and Bioengineering, Politecnico di Milano (2021)
5. D. Patel, D. Canaday, M. Girvan, A. Pomerance, E. Ott, Using machine learning to predict statistical properties of non-stationary dynamical processes: system climate, regime transitions, and the effect of stochasticity. *Chaos* **31**(3), 033149 (2021)
6. J. Pathak, Z. Lu, B.R. Hunt, M. Girvan, E. Ott, Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos* **27**(12), 121102 (2017)
7. G. Guariso, G. Nunnari, M. Sangiorgio, Multi-step solar irradiance forecasting and domain adaptation of deep neural networks. *Energies* **13**(15), 3987 (2020)
8. M. Sangiorgio, F. Dercole, Robustness of LSTM neural networks for multi-step forecasting of chaotic time series. *Chaos Solitons Fractals* **139**, 110045 (2020)
9. T. He, J. Zhang, Z. Zhou, J. Glass, Quantifying exposure bias for neural language generation (2019). *arXiv preprint arXiv:1905.10617*
10. J. Brajard, A. Carrassi, M. Bocquet, L. Bertino, Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the Lorenz 96 model. *J. Comput. Sci.* **44**, 101171 (2020)
11. P. Chen, R. Liu, K. Aihara, L. Chen, Autoreervoir computing for multistep ahead prediction based on the spatiotemporal information transformation. *Nat. Commun.* **11**(1), 1–15 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.