Neural Epistemology in Dynamical System Learning

Chapter · June 2018

DOI: 10.1007/978-981-13-8950-4_20

Pietro Barbiero
Politecnico di Torino

Giansalvo Cirrincione
Université de Picardie Jules Verne

Maurizio Cirrincione
University of the South Pacific

Elio Piccolo
Politecnico di Torino

Neural Epistemology in Dynamical System Learning

Pietro Barbiero[1], Giansalvo Cirrincione[2], Maurizio Cirrincione[2], Elio Piccolo[3], and Francesco Vaccarino[1]

Politecnico di Torino, Department of Mathematical Sciences, 10126 Torino, Italy
pietro.barbiero@studenti.polito.it
University of Picardie Jules Verne, Lab. LTI, Amiens, France. University of South Pacific, Suva, Fiji
exin@u-picardie.fr
Politecnico di Torino, Department of Control and Computer Engineering, 10126 Torino, Italy
elio.piccolo@polito.it

Abstract. In the last few years neural networks are effectively applied in different fields. However, the application of empirical-like algorithms as feed-forward neural networks is not always justified from an epistemological point of view [1]. In this work the assumptions for the appropriate application of machine learning empirical-like algorithms to dynamical system learning are investigated from a theoretical perspective. A very simple example shows how the suggested analyzes are crucial in corroborating or discrediting machine learning outcomes.

Keywords: epistemology, time series learning, feed-forward neural networks, machine learning, forecasting, Poincaré recurrence theorem, bifurcation theory

1 The Need for an Epistemology

In the last few years machine learning and neural networks are effectively applied in different fields. Experiments often show great results both on structured and unstructured data. However, when they obtain poor results, it is not straightforward from a human being point of view to understand why. In some respects, this need can seem absurd. We typically employ machine learning to solve tasks which are not solvable by human beings. Nevertheless, we require to understand how they make decisions and, above all, we want to understand the circumstances in which they struggle to get good results. Generally, when we observe such situations, the spontaneous reaction is to charge the architecture, the hyperparameter tuning or the optimization algorithm. This is the typical "scapegoat" answer where the impeachment is charged to the algorithm. *However it does not question the use of the algorithm itself.* Can we really be so sure? In this work, we try to collect some justifications that can support the use of machine learning algorithms to dynamical system learning. The kind of problem we are stating is epistemological. Epistemology is a branch of phylosophy which investigates the origin, nature, methods, and limits of knowledge. Given the strict correlation between science and knowledge, epistemology is probably the closest branch of philosophy to science. Apart from the diatribes between justificationism and critical rationalism, machine learning needs epistemology for two main reasons. On one hand, researchers will be aware in advance about the limits of machine learning algorithms before they dive into a new problem. On the other hand, epistemology can have a social impact in increasing the confidence of external users to machine learning.

2 Mathematical Models of Dynamical Systems

From Newton and Leibniz on, the standard way of mathematicians to represent real world phenomena is based on calculus [2]. More specifically, time-variant phenomena are usually described either by differential equations or by recurrence relations. Both approaches represent physical quantities q and their rate of change q_r by means of equations:

$$\boldsymbol{q} = \boldsymbol{f}(\boldsymbol{q}_r) \tag{1}$$

This kind of representations are very straightforward to interpret by humans. Thus, they have emerged as references both in purely theoretical subjects and in making hypotheses for new experiments. However, their compact representation hides the global understanding of the phenomenon. One of the most used approach consists in solving the system of equations and drawing the corresponding phase space [3]. The phase space is a space in which all possible states of a system are represented, with each possible state corresponding to one unique point in the phase space. The solution of the dynamical equations can be represented as a trajectory in the phase space. The resulting plot gives qualitative but global information about the system behavior. In the following we will focus on two trajectory characteristics: boundness and periodicity. Periodic trajectories are curves that repeat their values in regular intervals. A bounded trajectory, instead, is a curve which can be represented in a finite volume. Dynamical systems having bounded or periodic solutions are fully observable (at least theoretically) i.e. we can plan an experiment in order to observe each possible state of the system or, for periodic ones, we can collect enough data to infer the system behavior outside the experiment window.

3 The Poincaré recurrence theorem

The Poincaré recurrence theorem points out an interesting property of dynamical systems. Indeed, it gives the assumptions under which a dynamical system returns close to previous states in a finite time interval.

Theorem 1. *If a dynamical system has a bounded domain in the phase space, then for each open neighborhood I_P of a point P in the phase space there exists a point $P' \in I_P$ that the system will encounter in a finite time* [2] [4].

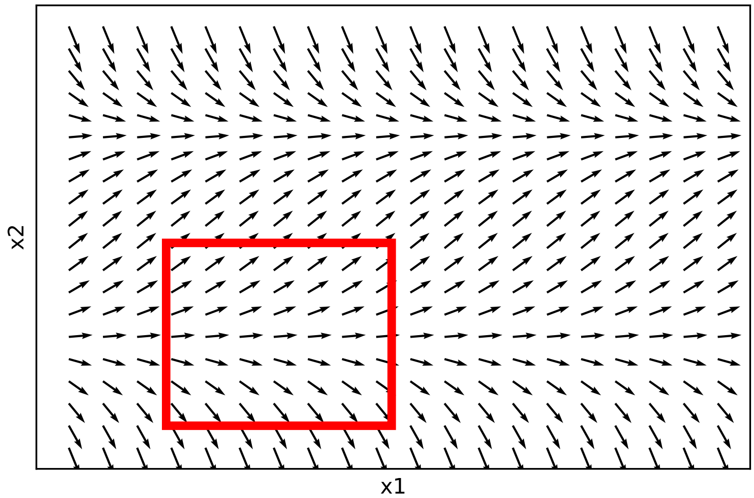

Fig. 1: Phase space of the system of equations: $x_1' = 1$ and $x_2' = x_2(1 - x_2)$. The arrows represent possible trajectories of the dynamical system. The red box represents a possible bound to trajectories. Observe that it does not exist a box which can bound these trajectories. The system behavior cannot be inferred by analyzing the states inside the experiment window.

The impact of this theorem on machine learning epistemology is enormous. Ensuring that *in a finite time* the system will return near a previous state, the theorem guarantees the existence of recurrence which is the building block of statistics and machine learning which literally learn associations (see for example the i.i.d. assumptions in [5]). This recurrence property of bounded phase spaces can be extended to periodic ones (even if unbounded) in the sense that recurrence is implicit in repetition. By analyzing the solutions of the dynamical equations, we can have an insight about what could be the limits of machine learning algorithms when applied to the problem.

4 What can go wrong

Despite the fact that most of real world phenomena have a bounded phase space, there are some aspects that can get machine learning into trouble. First of all, experiments may (and often have) practical limits in observing phenomena. Measuring instruments have limited ranges and/or recurrences may occur in undetectable time intervals. Thus, the retrieved data may lack in fundamental associations needed by algorithms. Another kind of problem may arise in real time

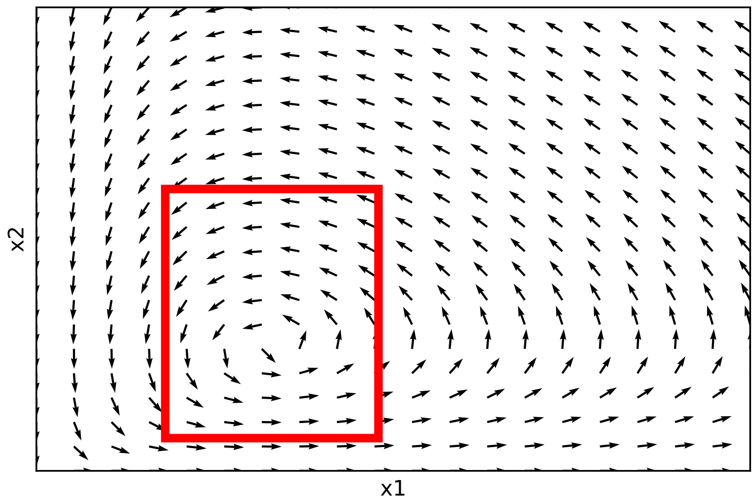

Fig. 2: Phase space of the system of equations: $x_1' = ax_1 - bx_1x_2$ and $x_2' = dx_1x_2 - cx_2$. This system is the Lotka-Volterra model also known as the predator–prey model. The arrows represent possible trajectories of the dynamical system. Observe that the red square is a bound for some trajectories.

applications when the algorithms process online data. Suppose you have modeled a phenomenon, you have carried out an experiment and used the output data to train an algorithm. Then you want to apply the trained algorithm to process a stream of new data. In this case, if the original model is not perfect or does not take into account some important variables, then the data used to train the algorithm may have a different distribution from real time ones (violating the i.i.d. assumptions [5]). In such cases machine learning algorithms try to extrapolate inferring outside the training domain. Unfortunately the results are usually decidedly worse. Finally, dynamical systems may be affected by bifurcations [6] [1]. Bifurcations occur when a small smooth change made to the state of the system causes a sudden change in its behaviour. In such cases, most machine learning algorithms struggle to get meaningful results. Thus, researchers should analyze both the model and the experiment carefully before applying machine learning in such contexts.

5 How Machine Learning See Dynamical Systems

Having modeled a dynamical phenomenon and designed the corresponding experiment, the outcome is a sequence of data usually organized as a time series

[7]. A time series is an ordered sequence of data points, where the order refers to time. In practice, a certain quantity (e.g. the temperature) is sampled at successive (often) equally spaced time instants. More formally a unidimensional time series can be described as:

$$a_n = (x_0, x_1, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_T) \qquad x_i \in \mathbb{R} \qquad \forall i \in [0, T] \tag{2}$$

where the data point x_i is a real number and it refers to the i^{th} time instant. The most common use of such data is forecasting analysis. Forecasting refers to the use of a mathematical model that tries to predict future values based on observed ones [8]. More formally, having defined a time window $w \in \mathbb{N}$, a time series forecasting model is a function:

$$f : \mathbb{R}^w \to \mathbb{R} \tag{3}$$

that uses the observed points $(x_i, \dots, x_{i+w-1})$ in order to predict the value of x_{i+w}. From a machine learning point of view, each observed point corresponds to a feature and the future value is the corresponding target. Following this perspective, it is possible to reshape the time series in order to build a more suitable database for machine learning algorithms:

$$(x_0, \dots, x_{w-1}) \to y_0 = x_w \tag{4}$$
$$(x_1, \dots, x_w) \to y_1 = x_{w+1} \tag{5}$$
$$(x_2, \dots, x_{w+1}) \to y_2 = x_{w+2} \tag{6}$$
$$\vdots \tag{7}$$

In practice, a machine learning algorithm should learn the associations from a training set of the above database.

6 A Song Experiment: The Frère Jacques Song

"Frère Jacques" is a famous French child-song [9]. Fig. 3 shows the music sheet of the song.

For the purpose of the experiments, assume that the music sheet is repeated from the beginning several times, thus simulating a bounded and periodic dynamical system. For the sake of visualization, initially consider a time window $w = 2$. Indeed, with such a choice it is possible to visualize predictions in a 2-dimensional plane. In order to set-up the database, the following one-hot encoding is chosen:

C = 1000000
D = 0100000
E = 0010000
F = 0001000
G = 0000100
A = 0000010
B = 0000001

Fig. 3: The Frère Jacques music sheet.

Given the encoding choice, it is possible to build a database iteratively in the following way [7] [8]:

$$x_1 = (C, D) \to y_1 = (E) \tag{8}$$
$$x_2 = (D, E) \to y_2 = (C) \tag{9}$$
$$\vdots \tag{10}$$

where the note symbols are replaced with their corresponding binary codes. The phase state of the described problem is limited and its trajectories are bounded in a finite volume as shown in fig. 4.

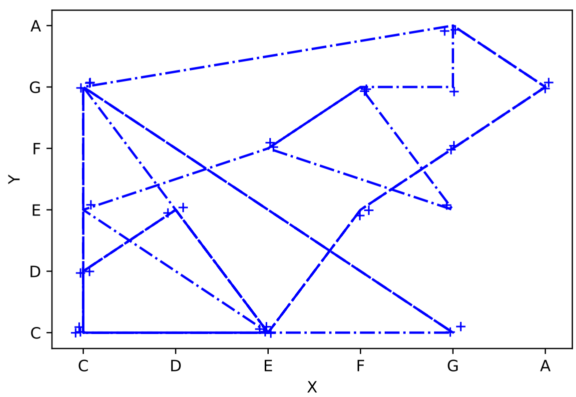

Fig. 4: Phase space of the Frère Jacques song (window size $w = 1$). Observe that the trajectories along which the dynamical system evolves are bounded in a finite volume.

Indeed, the system is periodic and exactly returns in a previous state after a finite (and short) time interval. Since the learning assumptions are fulfilled, then any machine learning algorithm can be correctly applied to the problem. In particular, we are interested in how Multi-Layer Perceptron (MLP) tackles time series forecasting [10] [11] [12]. For the purpose of this experiment, the MLP is equipped with the cross-entropy error function and a SoftMax output layer in order to perform multi-class recognition. The optimization algorithm used is the Adaptive momentum estimation optimizer (Adam). It is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [13]. The objective is forecasting the next note given the previous ones. So, each unit in the output layer correspond to a note and the network prediction corresponds to the output unit with the highest activation. The MLP architecture is composed of the input layer with 2 units (equal to the time window size), a hidden layer and an output layer with 7 units (one for each note). The experiments are carried out with 30 units in the hidden layer.

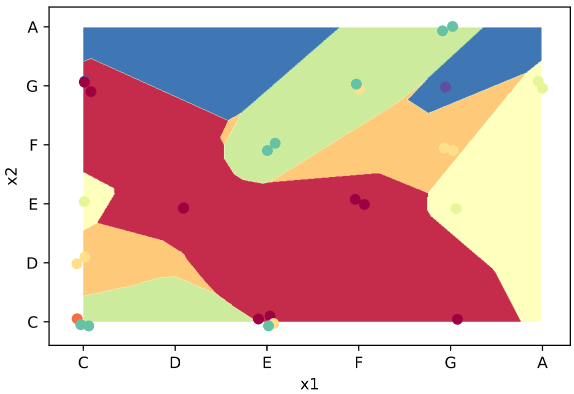

Fig. 5: Decision boundaries learned by a MLP for the Frère Jacques song with a window size $w = 2$.

Fig. 5 shows the input space x_1-x_2 having chosen a window of size $w = 2$. In the figure there are 7 regions with different colors. They are delimited by the estimated decision boundaries learned by the network. The MLP tries to delimit the input space with straight lines (or hyperplanes) in order to label each region of the space. Each point of a region is identified by the label of the region it belongs. This label is compared with the true target in order to evaluate the mismatch. If the predicted labels correspond to targets then the cost is low, otherwise it increases for each mislabeled sample. The experiment is repeated 100 times for different window sizes. The training accuracy increases accordingly to the window size as shown in table 1.

window size	training accuracy
$w = 1$	$39.4\% \pm 1.9\%$
$w = 2$	$82.9\% \pm 1.1\%$
$w = 3$	$86.2\% \pm 0.0\%$
$w = 4$	$92.9\% \pm 0.0\%$
$w = 5$	$96.3\% \pm 0.0\%$
$w = 6$	$96.2\% \pm 0.0\%$
$w = 7$	$100.0\% \pm 0.0\%$

Table 1: Training accuracy of the Frère Jacques song experiment for different window sizes.

These results can be explained by analyzing the phase space of the song. For example, when $w = 6$ there exists 2 sequences $X = [G, A, G, F, E, C]$ ("*Sonnez les matines!*") in the training set. However, the first one has $Y = G$ as target (the next note after the sequence), while the second one has $Y = C$. This is the only sequence in which the MLP fails with this window size. Indeed, the two sequences are the same, but with different targets. Hence, MLP has no information to disambiguate the forecast. This can be regarded as a discrete bifurcation. The experiment window is not enough to allow the disambiguation and the MLP fails the prediction even if the phase space is bounded. Carrying out the same analysis for all the window sizes, it turns out that the training performances listed above correspond to the performances of a Bayes classifier, i.e. they are the optimal ones.

7 Conclusion

In this work we pointed out the need for an epistemology for machine learning. To begin with, we focused on dynamical systems. We described the typical scientific process used to approach the problem from the design of mathematical models to the application of machine learning on the experiment outcomes. We observed how the Poincaré recurrence theorem could be a reliable building block for the foundation of such epistemology. Finally, we underlined how further analyzes should go with the application of machine learning to dynamical systems. Specifically, they are related to: the boundness of the trajectories with regard to the experiment window, the differences between the expected data distribution and the real one, and the presence of bifurcations. We illustrate with a simple experiment how the underlined problems may emerge. The example shows how the suggested analyzes are crucial in corroborating or discrediting machine learning outcomes.

Appendix: Source Code

The source code of the Frère Jacques song experiment is downloadable at `http://www.pietrobarbiero.eu/`.

References

[1] Kristian Barry Ellingsen. "The Epistemology of Learning in Artificial Neural Networks". In: (1995). ISSN: 0803-6489.
[2] Luis Barreira. "Poincaré recurrence: old and new". In: *XIVth International Congress on Mathematical Physics* (2006). DOI: 10.1142/9789812704016_0039.
[3] Morris W. Hirsch, Stephen Smale, and Robert L. Devaney. "Differential Equations, Dynamical Systems, and an Introduction to Chaos". In: (2012).
[4] Henri Poincaré. "Sur le problème des trois corps et les équations de la dynamique". In: *Acta Mathematica* (1890). URL: https://projecteuclid.org/euclid.acta/1485881725.
[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.
[6] Henri Poincaré. "L'Équilibre d'une masse fluide animée d'un mouvement de rotation". In: *Acta Mathematica* (1885). URL: http://smf4.emath.fr/Publications/Gazette/2005/104/smf_gazette_104_71-76.pdf.
[7] Thomas G. Dietterich. "Machine Learning for Sequential Data: A Review". In: *Structural, Syntactic, and Statistical Pattern Recognition*. Springer Berlin Heidelberg, 2002, pp. 15–30. ISBN: 978-3-540-70659-5. DOI: 10.1007/3-540-70659-3_2.
[8] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. "Machine Learning Strategies for Time Series Forecasting". In: *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures*. Springer Berlin Heidelberg, 2013, pp. 62–77. DOI: 10.1007/978-3-642-36318-4_3.
[9] *Frère Jacques Song*. URL: https://en.wikipedia.org/wiki/Fr%5C%C3%5C%A8re%5C_Jacques.
[10] Christopher Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, 1995. ISBN: 978-0198538646.
[11] *Tensorflow*. URL: https://www.tensorflow.org/.
[12] Ken-ichi Funahashi and Yuichi Nakamura. "Neural Networks, Approximation Theory, and Dynamical Systems". In: *Structure And Bifurcations Of Dynamical Systems-Proceedings Of The Rims Conference* (1992).
[13] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference for Learning Representations* (2014).
[14] Neil Gershenfeld. "The Nature of Mathematical Modeling". In: *Cambridge University Press* (2000).