International Journal of Advanced Research in Engineering Technology & Science
ISSN: 2349-2819
Email: editor@ijarets.org Volume-6, Issue-12 December- 2019 www.ijarets.org

A STUDY ON MODELLING CHAOTIC TIME SERIES WITH A REFERENCE OF CHAOS & MAXIMUM LYAPUNOV EXPONENT AND NEURAL NETWORKS

M. T. Kolhe

Assistant professor (Department of Mathematics)

Arts,Commerce and Science College Arvi Di-Wardha Maharashtra, India

Abstract

As a protostellar core accumulates additional gas from its environmental elements, the underlying breakdown starts with a given measure of rakish momentum. Since the rakish momentum is rationed, the rotation will increment causing the infalling gas to ultimately straighten and form a growth plate. The outward acceleration from the rotation will oppose the outspread gravitational power from the focal item, however breakdown can in any case happen along the upward direction.

Nonstop aggregation of mass can anyway prompt an unsound circle that outcomes in a progression of blasts during the accumulation stage. Modeling such dynamics should be possible by tackling the hydrodynamical equations for mass, momentum, and energy transport. The result is normally an exceptionally chaotic time series comprising of a few explodes (or spikes) delegate of the rambling way of behaving experienced all through the flimsiness.

Keywords:

Continuous, accumulation, mass

Introduction

Modeling chaotic systems have the component of capriciousness. That is, the dynamics are seriously delicate to the underlying circumstances. On the off chance that two introductory circumstances vary by a sum δx, after a time t they will contrast by δxeλt encountering remarkable detachment. Most chaotic systems can't be directly modeled by dynamical systems and are just accessible through time series.

Reenacting heavenly development starts with the breakdown of a gravitationally contracting core and go on into the period of star formation. This includes settling the hydrodynamical equations:

$$\frac{\partial \Sigma}{\partial t} = -\nabla_{\rm p} \cdot (\Sigma v_{\rm p})$$

$$\frac{\partial}{\partial t}(\Sigma v_{\rm p}) + [\nabla \cdot (\Sigma v_{\rm p} \otimes v_{\rm p})]_{\rm p} = -\nabla_{\rm p} P + \Sigma g_{\rm p} + (\nabla \cdot \Pi)_{\rm p}$$

$$\frac{\partial e}{\partial t} + \nabla_{\rm p} \cdot (e v_{\rm p}) = -P(\nabla_{\rm p} \cdot v_{\rm p}) - \Omega + \Gamma + (\nabla \cdot v)_{\rm pp'} : \Pi_{\rm pp'}$$

which relate to mass, momentum, and energy transport separately. The equations characterize Σ to be the surface mass density, e is the inward energy per surface region, P is the in an upward direction integrated pressure, vp = vrbr + vφφb is the velocity over the plane of the circle and ∇p = br∂/∂r + φbr −1∂/∂φ is the planar inclination administrator that takes the angle along the plane of the plate.

Vorobyov et al. [1] characterize the gravitational acceleration in the plate plane gp to consider the self gravity of the circle and the gravity of the formed protostar [2]. Finally, Π is the thick pressure tensor, the radiative cooling Ω and the warming function Γ are taken to be:

$$\Omega = F_c \sigma T_{\rm mp}^4 \frac{\tau}{1+\tau^2}, \qquad \Gamma = F_c \sigma T_{\rm irr}^4 \frac{\tau}{1+\tau^2}$$

where σ is the Stefan-Boltzmann constant, Tmp is the midplane temperature of gas, Tirr is the illumination temperature and τ is the optical profundity. The enormous episodes of gradual addition are because of bunches that form inside the plate because of gravitational unsteadiness and migrate to the focal point of the core.

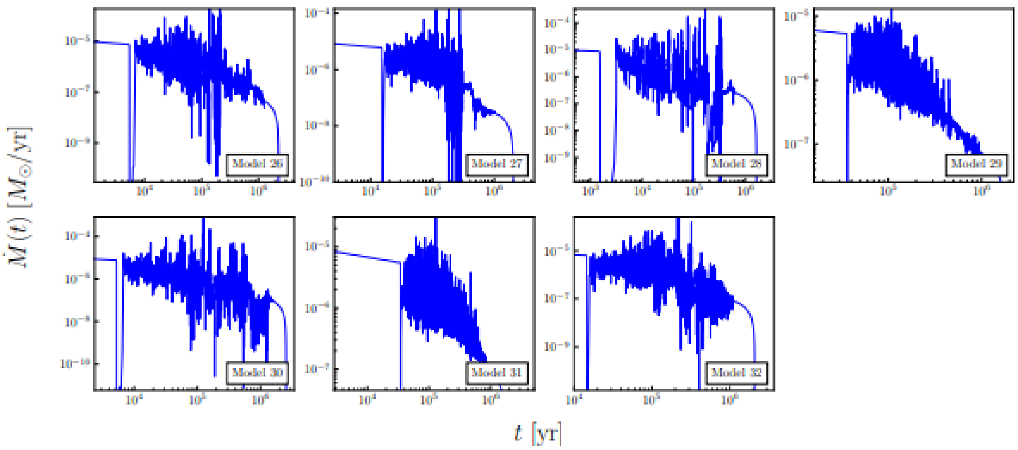

Figure 1: Hydrodynamic simulation outputs

Every one of the models that are utilized to reproduce the growth data are described by the underlying spiral profiles for the gas surface density Σ and precise velocity ω taking the form

$$\Sigma = \frac{r_0 \Sigma_0}{\sqrt{r^2 + r_0^2}}, \qquad \omega = 2\omega_0 \left(\frac{r_0}{r}\right)^2 \left[\sqrt{1 + \left(\frac{r}{r_0}\right)^2} - 1\right]$$

for Σ0 and ω0 being the surface mass density and rakish velocity at the focal point of the core and r0 = √ Ac2 s/πGΣ0 being the focal level range taking on A = 1.2 . [1]

The model is two-dimensional, accepts no magnetic field and takes the underlying gas temperature in falling cores to be 10 K. To generate gravitationally unsound shortened cores, each model is portrayed by the proportion defeat/r0 = 6 where defeat is the core's external range. Whenever defeat has been chosen, r0 can be found by means of the proportion set between the two amounts and the focal surface density can then be found utilizing r0 = √ Ac2 s/πGΣ0. At last, the cloud core mass Mcl can be observed involving the underlying outspread profile for the gas surface density Σ. The amount ω0 is chosen to such an extent that the proportion of rotational to gravitational energy β falls inside ∼ 10−4 and 0.07 [3].

MODELLING CHAOTIC TIME SERIES WITH CHAOS & MAXIMUM LYAPUNOV EXPONENT AND NEURAL NETWORKS

The dynamics of the growth rate in figure 1 demonstrate what is accepted to be exceptionally chaotic way of behaving. This mayhem can be evaluated in a constant Λ known as the Lyapunov example. The Lyapunov type successfully portrays the rate of partition between close directions in stage space. Regularly, the most extreme Lyapunov type is taken to describe a time series and the data can be deciphered as:

$$\Lambda_{\max} > 0, \qquad \text{Chaotic Dynamics}$$
$$\Lambda_{\max} = 0, \qquad \text{Regular Dynamics}$$
$$\Lambda_{\max} < 0, \qquad \text{Fixed-Point Dynamics}$$

where Λmax is taken to be the maximum Lyapunov exponent. In phase space, two signals that differ by an amount δX0 will diverge at a rate given by:

$$|\delta \mathbf{X}(t)| \sim e^{\Lambda t} |\delta \mathbf{X}_0|$$

Neural networks have been demonstrated to accurately inexact any nonstop function and are regularly alluded to as general approximators.

This capacity has given neural networks huge abilities to concentrate and gauge the hidden dynamical interaction happening for a time series. In its most vanilla form, neural networks feed input data from an info layer into one or many secret layers before a forecast is made at the result layer. Each layer is comprised of a progression of neurons that contain data and their particular transformations starting with one layer then onto the next depicted by a bunch of loads.

The single layer perceptron is frequently alluded to as the central unit of a neural network and its engineering schematically demonstrated in figure 2. The perceptron is a parallel direct classifier where the loads are recursively refreshed after each result until the cycle mistake ε (I) is under a client characterized edge.

Given the constraints of a direct model like the single layer perceptron, it is normal to lift to non-straight models like the multi-facet perceptron or all the more by and large, the multilayered neural network. Considering that this examination manages time series, accept that the info data currently comprises of a time series x(t) = [x1(t), x2(t), • • • , xn(t)]T .

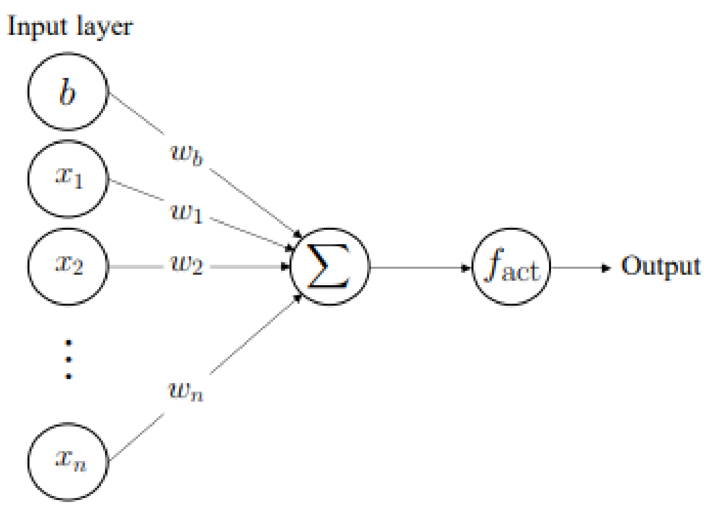

Figure 3: Schematic figure of a one layer perceptron

The multilayered network engineering is given in figure 4 and demonstrates a feed forward network where the sources of info are taken care of through the network in one direction. The idea is like the single layer perceptron where the availability of each layer is relegated a framework W(l) that addresses the loads associating a given network layer l. The secret layers apply non-direct transformations to inputs through initiation functions and proliferate the data forward through the network.

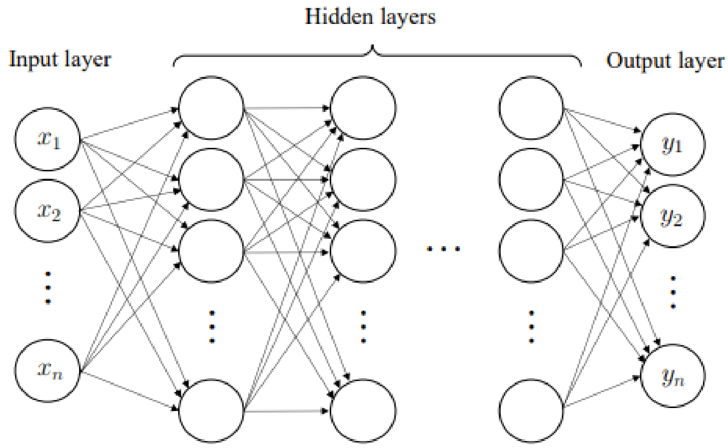

Figure 4: Schematic figure of a multilayered neural network

Like the single layer perceptron, data is taken care of forward through the network through a forward engendering approach and a mistake metric is determined to measure the network's performance. The forward engendering calculation is commonly given as:

$$\mathbf{h}^{(l+1)} = f_{\text{act}}(\mathbf{z}^{(l+1)}) = f_{\text{act}}((\mathbf{W}^{(l)})^T \cdot \mathbf{h}^{(l)})$$

for h (l) being the result at layer l where h (1) = x(t) and h (L) = xˆ(t) for a network having L layers. At the result layer, the mean-squared blunder (or another misfortune) is given as:

$$\text{MSE} = \sum_i \frac{(\hat{x}_i - x_i)^2}{N}$$

where N is the length of the info vector.

In the event that the mistake is over a client characterized edge, the loads are recursively refreshed by a back spread calculation. This normally includes changing the loads to limit an expense function, C given regarding some mistake metric.

To limit, the subsidiary of the expense function is taken as for the info:

$$\frac{dC}{d\mathbf{a}^{(L)}} \cdot \frac{d\mathbf{a}^{(L)}}{d\mathbf{z}^{(L)}} \cdot \frac{d\mathbf{z}^{(L)}}{d\mathbf{a}^{(L-1)}} \cdot \frac{d\mathbf{a}^{(L-1)}}{d\mathbf{z}^{(L-1)}} \cdot \frac{d\mathbf{z}^{(L-1)}}{d\mathbf{a}^{(L-2)}} \cdots \frac{d\mathbf{a}^{(1)}}{d\mathbf{z}^{(1)}} \cdot \frac{\partial \mathbf{z}^{(1)}}{\partial \mathbf{x}(t)}$$

and so the weights are updated as:

$$\mathbf{W}_{j+1}^{(l)} = \mathbf{W}_{j}^{(l)} - \gamma \frac{\partial C}{\partial \mathbf{W}_{j}^{(l)}}$$

for a learning rate γ.

To additionally sum up the network engineering, one can present a progression of criticism circles inside the network to permit information to move in the two directions. These types of networks are called recurrent neural networks (RNN) and have the property of holding 'memory' of past information states. RNN's are appropriate for time series on the grounds that the inner state of the network is kept up with starting with one time step then onto the next. The essential design is given in figure 5 and demonstrates the criticism arrangement for a fundamental 3-layered RNN. RNN's will fill in as the establishment for the reverberation state neural network (ESN) utilized in this examination.

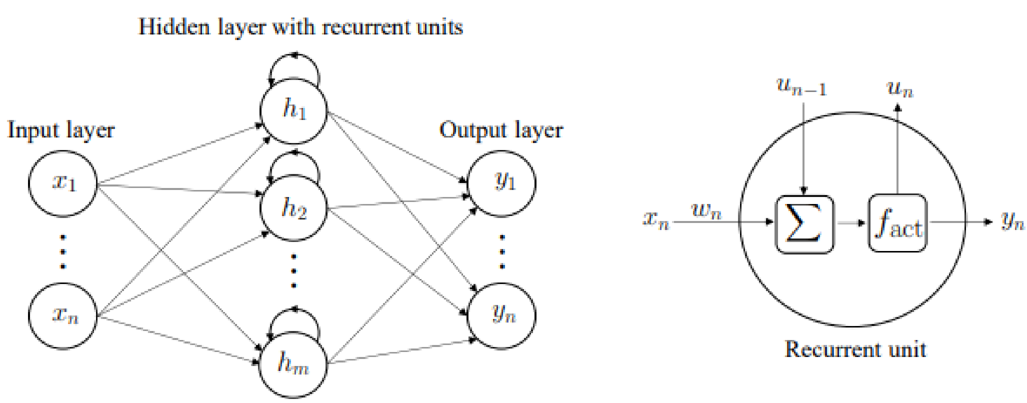

Figure 5: 3-layer RNN where the hidden layer neurons are recurrent

The reservoir processing system is basic to the functionality advantages of reverberation state neural networks. The system maps an info sign to a higher dimensional computational space. The reservoir comprises of meager, haphazardly associated neurons with a client characterized availability having fixed, non-direct dynamics. One of the key advantages comes from computational productivity. Dissimilar to a conventional neural network that trains through a progression of forward and in reverse spread through the network, the reservoir registering system just requires yield loads to be prepared. A reverberation state neural network is comprised of an information layer, a reservoir and a result layer having weight grids Winput, Wx and Woutput individually.

Conclusion

At long last, the late protostellar stage gave the system to an acquaintance likewise with how neural network models can be utilized as a hearty indicator for time series. Another form of recurrent neural networks were created and demonstrated promising fundamental outcomes in modeling the chaotic state encompassing the verbose gradual addition associated with late protostellar development. Such models are meant to being a compelling instrument for data that comes up short on successful fundamental dynamical system model.

References

[1] Eduard I Vorobyov, Vardan Elbakyan, Takashi Hosokawa, Yuya Sakurai, Manuel Guedel, and Harold Yorke. Effect of accretion on the pre-main-sequence evolution of low-mass stars and brown dwarfs. Astronomy & Astrophysics, 605:A77, 2017.

[2] Eduard I Vorobyov and Shantanu Basu. The burst mode of accretion and disk fragmentation in the early embedded stages of star formation. The Astrophysical Journal, 719(2):1896, 2010.

[3] Paola Caselli, Priscilla J Benson, Philip C Myers, and Mario Tafalla. Dense cores in dark clouds. xiv. n2h+ (1-0) maps of dense cloud cores. The Astrophysical Journal, 572(1):238, 2002.

[4] D. Ruelle J. P. Eckmann, S. Oliffson Kamphorst and S. Ciliberto. Liapunov exponents from time series. Phys. Rev. A 34, 4971, 1986.

[5] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. Physical review letters, 120(2):024102, 2018.

[6] Claudio Gallicchio and Alessio Micheli. Deep echo state network (deepesn): A brief survey. arXiv preprint arXiv:1712.04323, 2017.

[7] Jingpei Dan, Wenbo Guo, Weiren Shi, Bin Fang, and Tingping Zhang. Deterministic echo state networks based stock price forecasting. 2014, 2014.

[8] Xiao-chuan Sun, Hong-yan Cui, Ren-ping Liu, Jian-ya Chen, and Yun-jie Liu. Modeling deterministic echo state network with loop reservoir. Journal of Zhejiang University SCIENCE C, 13(9):689–701, 2012.

[9] Marc Audard, Péter Abrahám, Michael M Dunham, Joel D Green, Nicolas Grosso, Kenji Hamaguchi, Joel H Kastner, Agnes Kóspál, Giuseppe Lodato, Marina M Romanova, et al. Episodic accretion in young stars. Protostars and Planets VI, 387(1401.3368), 2014.