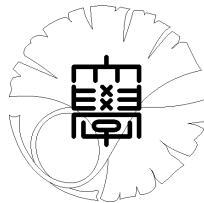


数理科学実践研究レター 2018-16 August 27, 2018

**State-space model estimation of a noisy time series
data using recurrent neural network**

by

Md. Maruf Hossain



UNIVERSITY OF TOKYO

GRADUATE SCHOOL OF MATHEMATICAL SCIENCES

KOMABA, TOKYO, JAPAN

State-space model estimation of a noisy time series data using recurrent neural network

リカレントニューラルネットワークを用いた雑音を含んだ時系列データの状態空間モデル推定

Md. Maruf Hossain¹ (Dept. of Physics, The University of Tokyo)
ホサイン モハンマド マルフ (東京大学大学院物理学専攻)

Abstract

Neural networks provide a flexible way of capturing the underlying dynamics of a system from observed time series data. Correct model identification becomes difficult because increasing or decreasing the nodes and layers of a neural network may overfit or underfit the noisy data, ending up with a wrong model. To address this issue, the independent and identically distributed noise from the observed data is removed by optimizing the autocorrelation coefficients at different lags. It is showed that a recurrent neural network can detect the correct state-space model of the system using the denoised data.

1 Introduction

To describe natural phenomena from observations, normally we start with a model. We first develop a theory for explaining our observations and then derive equations to describe the underlying system. With the advent of neural networks, we can reverse this process. As neural networks are universal approximators [1], we can prepare a network of arbitrary complexity and learn to fit our data without assuming any prior model. As long as the observed data contains no noise, this strategy will work, and we will get the correct model after learning. But, observed data always comes with noise. We may end up overfitting or underfitting our noisy data by starting the learning process with a very complex network or an excessively simplified network.

Recurrent neural networks (RNN) have been used in various applications to predict time series data [2, 3, 4]. Time series data can be modeled with differential equations, and an RNN can simulate the state-space representation of a system of differential equations [2, 4]. As a result, it is possible to detect the differential equations that a system follows by starting with an arbitrarily complex RNN. In this case as well, overfitting and underfitting of noisy data would not allow us to predict the correct complexity of the trained RNN, ending up with a wrong system of differential equations [5]. To avoid this problem, provided that the noise is independent and identically distributed (i.i.d.), we can minimize the autocorrelation coefficients of the time series at different lags. We will have to train the RNN in such a way that the fitting loss is the minimum and the autocorrelation coefficients at different lags are zero. In this paper, denoising is achieved by implementing a Gaussian kernel moving average filter. The appropriate kernel width is found by optimizing the autocorrelation of the residuals. A recurrent neural network is then used to learn the state-space model of the system from this filtered signal.

2 State-space model from noisy data

Let us consider linear differential equations and concentrate on output noise only. For a linear differential equation of the form:

$$a_n \frac{d^n \Phi(t)}{dt^n} + a_{n-1} \frac{d^{n-1} \Phi(t)}{dt^{n-1}} + \dots + a_1 \frac{d\Phi(t)}{dt} + a_0 \Phi(t) = u(t), \quad (1)$$

¹maruf@light.phys.s.u-tokyo.ac.jp

the state-space model can be derived by treating $\frac{d^i \Phi}{dt^i}$, $i = n, n-1, \dots, 0$ as independent basis functions.

$$\Phi_{k+1} = A_k \Phi_k + B_k u_k, \quad (2)$$

$$y_k = C_k \Phi_k + v_k. \quad (3)$$

Here, Φ_k is the state vector, u_k is the input vector, and y_k is the observed output. The measurement noise v_k is assumed to be an i.i.d. Gaussian noise with zero mean and finite variance. A_k, B_k are the time-independent matrices that linearly relate the current state Φ_k and the current input u_k to the next state Φ_{k+1} . C_k is the matrix that gives us the observable y_k from the state Φ_k .

As v_k is an i.i.d. noise, the autocorrelation of v_k at any lag τ should be zero up to some numerical error. We can use this property of the noise to distinguish it from a signal generated from (2).

2.1 De-noising based on autocorrelation

To experiment with this property, we have produced a signal of the form $\tilde{y}_t = [y_0 \cos(\frac{t}{2} \sqrt{-\frac{a_1^2}{a_2^2} + \frac{4a_0}{a_2}}) + \frac{a_1 y_0 + 2a_2 \dot{y}_0}{a_2 \sqrt{-\frac{a_1^2}{a_2^2} + \frac{4a_0}{a_2}}} \sin(\frac{t}{2} \sqrt{-\frac{a_1^2}{a_2^2} + \frac{4a_0}{a_2}})] e^{-\frac{a_1 t}{2a_2}}$. To generate the noisy data y_i , we have added i.i.d Gaussian noise with mean, $\mu = 0$, and standard deviation, $\sigma \sim \frac{A}{2}$ to \tilde{y}_t . \tilde{y}_t is the general solution for the 2nd order differential equation: $a_2 \frac{d^2 y(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_0 y(t) = 0$, $y(0) = y_0$, $\frac{dy(t)}{dt}|_{t=0} = \dot{y}_0$. Our goal is to detect this differential equation from the noisy data by considering autocorrelation. This task is not equivalent to finding the parameters a_2, a_1, a_0 because we assume that we do not have any prior knowledge about the form of the model equation. To achieve this, we will first denoise the generated data with a Gaussian kernel moving average filter:

$$\hat{y}_i = \sum_{j=-\infty}^{\infty} \frac{y_j}{k\sqrt{2\pi}} e^{-\frac{(y_j - y_i)^2}{2k^2}} \quad (4)$$

Here, k is the kernel width of our moving average filter. After applying the filter of (4), we have taken the residuals $r_i = y_i - \hat{y}_i$ and calculated their autocorrelations $R_r(\tau)$ at different lags.

$$R_r(\tau) = \frac{E[(r_t - \mu_r)(r_{t+\tau} - \mu_r)]}{\sigma_r^2} \quad (5)$$

The optimization is done by taking the total absolute autocorrelation for all possible lags:

$$\sum_{\tau} |R_r(\tau)| \quad (6)$$

Fig. 1(a) shows that an optimal kernel width k exists (around 9). In figure 1(b), we have plotted the noisy signal (blue dots), the original signal (black dashes) and the filtered signal (orange line) with $k = 9$. In figure 1(c) and 1(d), we have done the same, this time with $k = 2$ and $k = 15$. Comparing these figures, we see that the reconstruction has the minimum error when $k = 9$. This result means that it is possible to use autocorrelation as an objective function for reconstructing the original signal from a noisy data. But this assumption is valid when we consider i.i.d observational noise only.

2.2 Model Identification with Recurrent Neural Network

We have used the denoised data \hat{y}_i from fig. 1(b) to train a recurrent neural network. As shown in Fig. 2(a), The network has 3 layers to start with, each layer having 3 nodes. To represent the state-space transition as described in (2), any nonlinear transformation function has not been used for the outputs of the nodes. The output of of each node is a weighted sum of the values at other nodes connected to it. After training,

as shown in Fig. 2(b), the network reduces to a system with one input node, 2 nodes representing the state-space of our 2nd order differential equation and the output node. We have used the values $a_2 = 5, a_1 = 0.1, a_0 = 7$. The corresponding transition matrix $A_k = \exp\left(\begin{bmatrix} 0 & 1 \\ -a_0/a_2 & -a_1/a_2 \end{bmatrix} T\right)$ is given by $\begin{bmatrix} A_{k11} & A_{k12} \\ A_{k21} & A_{k22} \end{bmatrix} = \begin{bmatrix} 0.999 & 0.04 \\ -0.056 & 0.998 \end{bmatrix}$. Here, $T = 0.04$ s is the sampling period. After training the model of 3 nodes for the transition matrix with the denoised signal, we obtain the following result.

$$\begin{bmatrix} A_{k11} & A_{k12} & A_{k13} \\ A_{k21} & A_{k22} & A_{k23} \\ A_{k31} & A_{k32} & A_{k33} \end{bmatrix} = \begin{bmatrix} 0.999 & 0.039 & 3.6 \times 10^{-3} \\ -0.058 & 0.999 & -4.6 \times 10^{-3} \\ -1.4 \times 10^{-3} & -1.9 \times 10^{-3} & 0.993 \end{bmatrix} \quad (7)$$

The transition matrix in (7) has elements $|A_{k13}|, |A_{k23}|, |A_{k31}|, |A_{k32}| < 5 \times 10^{-3}$. We can use this value as a threshold and prune these connections in the neural network. All the connections from the 3rd state will be pruned. Hence, the remaining matrix will represent a 2-states model. The values of the matrix elements are also the same up to some computational error as the original matrix that we used to generate the noisy data. This means that although we have used a 3-states model for training a neural network with noisy output which is originally described by a 2-states system, the denoising method based on autocorrelation allows us to predict the correct complexity of the model without any overfitting or underfitting. To further improve the identification performance, we can use more efficient methods for denoising such as wavelet transformations, kernel ridge regression, etc. (see [6]). Whatever the strategy is, we can always use the denoising method based on autocorrelation as described in this paper.

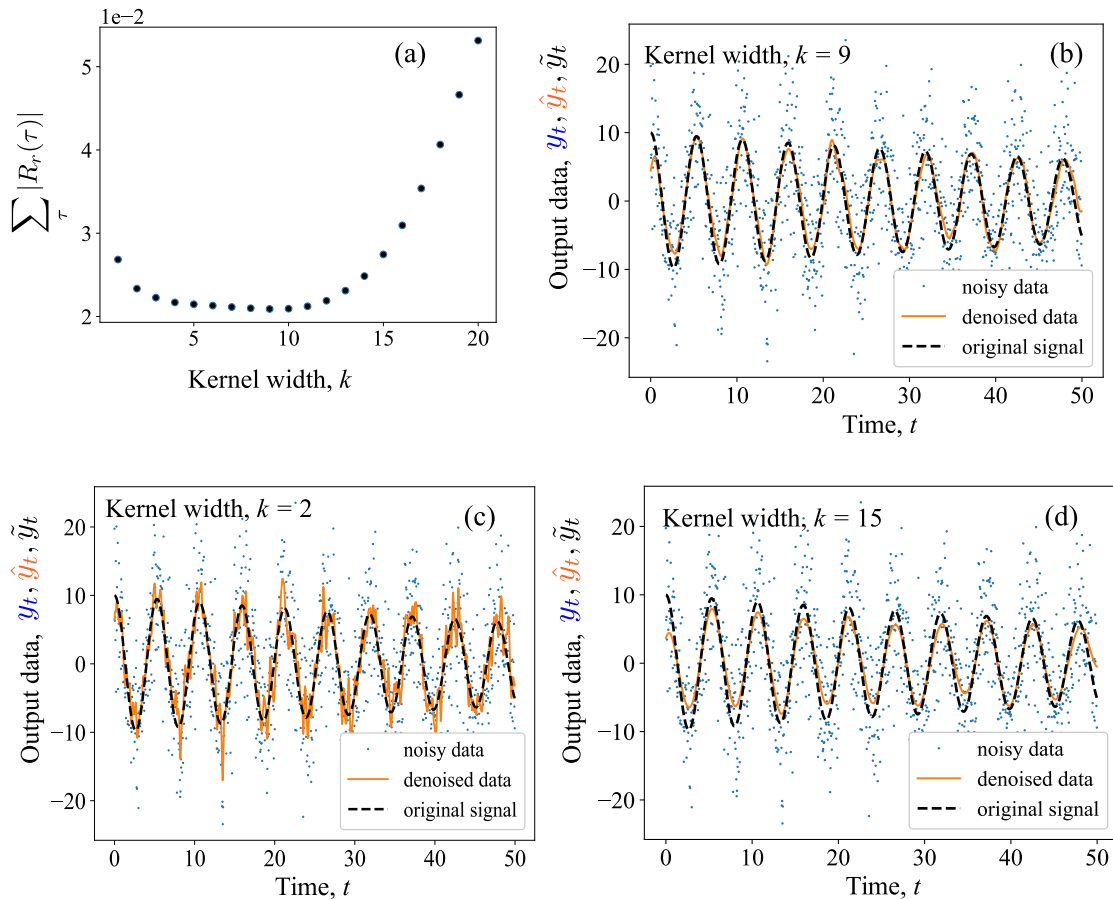


Fig. 1: Denoising with a Gaussian kernel moving average filter which minimizes the autocorrelation

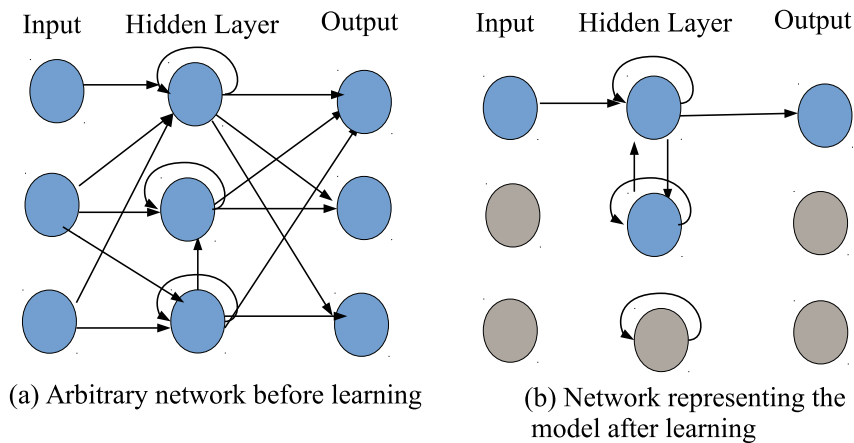


Fig. 2: Model identification of the denoised data with a recurrent neural network

3 Conclusion

The removal of noise from the observed data is a crucial step in identifying the correct model with neural networks. As long as the noise is i.i.d., the autocorrelation coefficients at different lags for the residuals after fitting should be zero. This property of i.i.d. noise has been availed in this paper to identify the state-space model with a recurrent neural network. This strategy works as long as the noise does not have a memory and the system dynamics is deterministic, i.e., it is defined by a linear system of ordinary differential equations, a non-linear differential equation, or a partial differential equation. For the nonlinear case, we may either use an appropriate nonlinear function at the output node of the RNN or find the linearized version of it around an attractor. A separable partial differential equation can be expressed with a system of ordinary differential equations, so finding the latter is equivalent to finding the former. However, the current method of minimizing the autocorrelation needs to be extended for cases when the noise has a memory, or the dynamics of the system is stochastic.

The author is indebted to Prof. Peter Tino, Prof. Masahiro Yamamoto, and Dr. Shunsuke Tsuchioka for their comments and advice during the discussions. This research has been supported by the FMSP leading graduate course of the University of Tokyo.

References

- [1] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks*, 4(2) (1991), 251-257.
- [2] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," in *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240-254, Mar 1994.
- [3] T. Guo, Z. Xu, X. Yao, H. Chen, K. Aberer, and K. Funaya, "Robust Online Time Series Prediction with Recurrent Neural Networks," *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Montreal, QC, 2016, pp. 816-825.
- [4] P. Tino and G. Dorffner, "Predicting the future of discrete sequences from fractal representations of the past," *Machine Learning*, 45(2), pp. 187-218, 2001.
- [5] R. G. Krishnan, D. Liang, and M. D. Hoffman, "On the challenges of learning with inference networks on sparse, high-dimensional data," *The 21st International Conference on Artificial Intelligence and Statistics*, 2018.
- [6] A. Grossmann and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape," *Soc. Int. Am. Math. (SIAM), J. Math. Analys.*, 15(1984), 723-736.