Explainable nonlinear modelling of multiple time series with invertible neural networks*

Luis Miguel Lopez-Ramos**[0000−0001−8072−3994], Kevin Roy**, and Baltasar Beferull-Lozano[0000−0002−0902−6245]

[1] SFI Offshore Mechatronics Center, University of Agder
[2] Intelligent Signal Processing and Wireless Networks (WISENET) Center
[3] Department of ICT, University of Agder, Grimstad, Norway

Abstract. A method for nonlinear topology identification is proposed, based on the assumption that a collection of time series are generated in two steps: i) a vector autoregressive process in a latent space, and ii) a nonlinear, component-wise, monotonically increasing observation mapping. The latter mappings are assumed invertible, and are modeled as shallow neural networks, so that their inverse can be numerically evaluated, and their parameters can be learned using a technique inspired in deep learning. Due to the function inversion, the backpropagation step is not straightforward, and this paper explains the steps needed to calculate the gradients applying implicit differentiation. Whereas the model explainability is the same as that for linear VAR processes, preliminary numerical tests show that the prediction error becomes smaller.

Keywords: Vector autoregressive model · nonlinear · network topology inference · invertible neural network

1 Introduction

Multi-dimensional time series data are observed in many real-world systems, where some of the time series are influenced by other time series. The interrelations among the time series can be encoded in a graph structure, and identifying such structure or topology is of great interest in multiple applications [1]. The inferred topology can provide insights about the underlying system and can assist in inference tasks such as prediction and anomaly detection.

In real-world applications such as neuroscience and genomics, signal interrelations are often inherently nonlinear [2–4]. In these cases, using linear models may lead to inconsistent estimation of causal interactions [5]. We propose deep learning based methods by applying feed-forward invertible neural networks. This project proposes a low-complexity nonlinear topology identification method that is competitive with other nonlinear methods explaining time series data from a heterogeneous set of sensors.

* The work in this paper was supported by the SFI Offshore Mechatronics grant 237896/O30.

** Equal contribution in terms of working hours.

1.1 State of the art and contribution

The use of linear VAR models for topology identification have been well-studied. A comprehensive review of topology identification algorithms was recently published [1], where the issue of nonlinearity is discussed together with other challenges such as dynamics (meaning estimating time-varying models).

In [6], an efficient algorithm to estimate linear VAR coefficients from streaming data is proposed. Although the linear VAR model is not expressive enough for certain applications, it allows clear performance analysis, and is subject to continuous technical developments, such as a novel criterion for automatic order selection [7], VAR estimation considering distributions different to the Gaussian, such as Student's t [8], or strategies to deal with missing data [8, 9, 6].

Regarding non-linear topology identification based on the VAR model, kernels are used in [10, 11] to linearize the nonlinear dependencies by mapping variables to a higher-dimensional Hilbert space. The growth of computational complexity and memory requirements (a.k.a. "curse of dimensionality") associated with kernel representations is circumvented in [10, 11] by restricting the numeric calculation to a limited number of time-series samples using a time window, which results in suboptimal performance. A semiparametric model is proposed for the same task in [12].

A different class of nonlinear topology identification methods are based on deep feedforward or recurrent NNs [5, 13] combined with sparsity-inducing penalties on the weights at one layer, labeled as "Granger-causality layer".

Recent work [14] considers a nonlinear VAR framework where the innovations are not necessarily additive, and proposes estimation algorithms and identifiability results based on the assumption that the innovations are independent.

All the aforementioned nonlinear modeling techniques are based on estimating nonlinear functions that predict the future time series values in the measurement space, which entails high complexity and is not amenable to predicting multiple time instants ahead. The main contribution of our work is a modeling assumption that accounts for mild nonlinear relations that are independent of the (linear) multivariate structure, and reduces the complexity associated with long-term predictions, as explained in detail in Sec. 3.

2 Background

2.1 Graph Topology Identification

Estimating topology of a system means finding the dependencies between network data time series. These dependencies may not be physically observable; rather, there can be logical connections between data nodes that are not physically connected, but which may be (indirectly) logically connected due to, e.g. control loops. Topology inference has the potential to contribute to the algorithmic foundations to solve important problems in signal processing (e.g. prediction, data completion, etc..) and data-driven control.

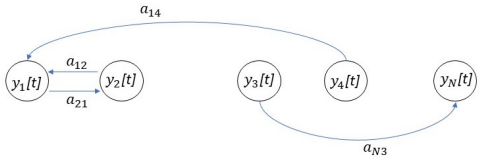

Fig. 1. Illustration of an N-node network with directed edges

While the simplest techniques such as correlation graphs [15] cannot determine the direction of interactions, one may also employ to this end structural equation models (SEM) or Bayesian networks [16]. However, such methods account only for memory-less interactions. On the other hand, causality in the Granger [17] sense is based on the idea that the cause precedes the effect in time, and knowledge about the cause helps predicting the effect more accurately. The way Granger causality is defined makes it interesting, from a conceptual point of view, for understanding data dependencies; however, it is often computationally intractable. Thus, alternative causality definitions, such as those based on vector autoregressive (VAR) models [17, 6] are typically preferred in practical scenarios. The simplest possible VAR model is a linear VAR model.

Consider a collection of N sensors, where $y_n[t]$ denotes the measurement of the n-th sensor at time t. A P-th order linear VAR model can be formulated as

$$y[t] = \sum_{p=1}^{P} A_p y[t-p] + u[t], \qquad P \leq t \leq T \tag{1}$$

where $y[t] = [y_1[t], ..., y_N[t]]^T$, $A_p \in R^{N \times N}$, p = 1,. . . , P, are the matrices of VAR parameters (see Fig. 2) , T is observation time period, and $u[t] = [u_1[t], ..., u_N[t]]$ is an innovation process typically modeled as a Gaussian, temporally white random process. With $a_{n,n'}^{(p)}$ being the (n, n') entry of the matrix A_p, the r.h.s above takes the form:

$$y_n[t] = \sum_{n'=1}^{N} \sum_{p=1}^{P} a_{n,n'}^{(p)} y_{n'}[t-p] + \ u_n[t], \qquad P \leq t \leq T \tag{2}$$

for $n = 1, \ldots, N$, The problem of identifying a linear VAR causality model reduces to estimating the VAR coefficient matrices $\{A_p\}_{p=1}^{P}$ given the observations $\{y[t]\}_{t=0}^{T-1}$. The VAR causality [18] is determined from the support of the VAR matrix parameters and can be interpreted as a surrogate (yet not strictly equivalent) for Granger causality[4].

[4] Notice that VAR models encode lagged interactions, and other linear models such as structural equation models (SEM) or structural VAR (SVAR) are available if interactions at a small time scale are required. In this paper, for the sake of simplicity, we focus on learning non-linear VAR models. However, our algorithm designs can also accomodate the SEM and SVAR frameworks without much difficulty.

2.2 Nonlinear function approximation

The main advantages of linear modeling are its simplicity, the low variance of the estimators (at the cost of a higher bias compared to more expressive methods), and the fact that linear estimation problems often lead naturally to convex optimization problems, which can be solved efficiently.

However, there are several challenges related to inferring linear, stationary models from real-world data. Many instances such as financial data, brain signals, industrial sensors, etc. exhibit highly nonlinear interactions, and only nonlinear models have the expressive capacity to capture complex dependencies (assuming that those are identifiable and enough data are provided for the learning). Some existing methods have tried to capture nonlinear interactions using kernel-based function approximators (see [4, 11] and references therein). In the most general non-linear case, each data variable $y_n[t]$ can be represented as a non-linear function of several multi-variate data time series as:

$$y_n[t] = h_n(y_{t-1}, \ldots, y_{t-P}) + u_n[t], \tag{3}$$

where $y_{t-p} = [y_1[t-p], y_2[t-p], ..., y_N[t-p]]^T$, and $h(\cdot)$ is a non-linear function.

However, from a practical perspective, this model is too general to be useful in real applications, because the class of possible nonlinear functions is unrestricted and, therefore, the estimators will suffer from high variance. Notice also that learning such a model would require in general an amount of data that may not be available in realistic scenarios, and requiring a prohibitive complexity. A typical solution is to restrict the the modeling to a subset of nonlinear functions, either in a parametric (NN) or nonparametric (kernel) way.

Our goal in this paper is to learn nonlinear dependencies with some underlying structure making it possible to learn them with limited complexity, with an expressive slightly higher than linear models.

3 Modelling

The linear coefficients in (1) are tailored to assessing only linear mediating dependencies. To overcome this limitation, this work considers a non-linear model by introducing a set of node dependent nonlinear functions $\{f_i\}_{i=1}^N$. Previous works on nonlinear topology identification [4, 5, 11] estimate nonlinear multivariate models without necessarily assuming linear dependencies in an underlying space; rather, they directly estimate non-linear functions from and into the real measurement space without assuming an underlying structure. In our work, we assume that the multivariate data can be explained as the nonlinear output of a set of observation functions $\{f_i\}_{i=1}^N$ with a VAR process as an input. Each function f_i represents a different non-linear distortion at the i-th node.

Given data time series, the task is to jointly learn the non-linearities together with a VAR topology in a feature space which is linear in nature, where the outputs of the functions $\{f_i\}_{i=1}^N$ belong to. Such functions are required to be invertible, so that sensor measurements can be mapped into the latent feature

space, where the linear topology (coefficients) can be used to generate predictions, which can be taken back to the real space through $\{f_i\}_{i=1}^{N}$. In our model, prediction involves the composition of several functions, which can be modeled as neural networks. The nonlinear observation function at each node can be parameterized by a NN that is in turn a universal function approximator [19]. Consequently, the topology and non-linear per-node transformations can be seen in aggregation as a DNN, and its parameters can be estimated using appropriate deep learning techniques.

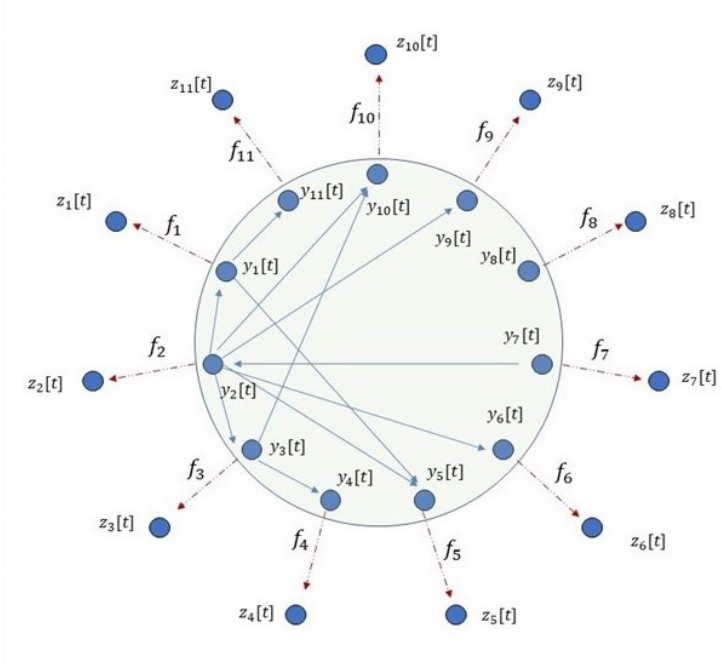

Fig. 2. Causal dependencies among a set of time series are linear in the latent space represented by the green circle. However, the variables in the latent space are not available, only nonlinear observations (output of the functions f_i) are available.

The idea is illustrated in Figure 2. The green circle represents the underlying latent vector space. The exterior of the circle is the space where the sensor measurements lie, which need not be a vector space. The blue lines show the linear dependency between the time series inside the latent space. The red line from each time series shows the transformation to the measurement space. Each sensor is associated with a different nonlinear function. Specifically, if $y_i[t]$ denotes the i-th time series in the latent space, the measurement (observation) is modeled as $z_i[t] = f_i(y_i[t])$. The function f_i is parameterized as a neural network layer with M units, expressed as follows:

$$f_i(y_i) = \sum_{j=1}^{M} \alpha_{ij} \sigma(w_{ij} y_i - k_{ij}) + b_i \tag{4}$$

For the function f_i to be monotonically increasing (which guarantees invertibility), it suffices to ensure that α_{ij} and w_{ij} are positive $\forall j$. The pre-image of f_i is the whole set of real numbers, but the image is an interval $(\underline{z}_i, \bar{z}_i)$, which is in accordance ro the fact that sensor data are usually restricted to a dynamic range. If the range is not available a priori but sufficient data is available, bounds for the operation interval can be easily inferred.

Let us remark three important advantages in the proposed model:

- It is substantially more expressive than the linear model, while capturing non-linear dependencies with lower complexity than other non-linear models.
- It allows to predict with longer time horizons ahead within the linear latent space. Under a generic non-linear model, the variance of a long-term prediction explodes with the time horizon.
- Each non-linear nodal mapping can also adapt and capture any possible drift or irregularity in the sensor measurement, thus, it can directly incorporate imperfections in the sensor measurement itself due to, e.g. lack of calibration.

3.1 Prediction

Given accurate estimates of the nonlinear functions $\{f_i\}_{i=1}^N$, their inverses, and the parameters of the VAR model, future measurements can be easily predicted. Numerical evaluation of the inverse of f_i as defined in (4) can easily be done with a bisection algorithm.

Let us define $g_i = f_i^{-1}$. Then, the prediction consists of three steps, the first one being mapping the previous samples back into the latent vector space:

$$\tilde{y}_i[t-p] = g_i\left(z_i[t-p]\right) \tag{5a}$$

Then, the VAR model parameters are used to predict the signal value at time t (also in the latent space):

$$\hat{y}_i[t] = \sum_{p=1}^{p} \sum_{j=1}^{n} a_{ij}^{(p)} \tilde{y}_j[t-p] \tag{5b}$$

Finally, the predicted measurement at each node is obtained by applying f_i to the latent prediction:

$$\hat{z}_i[t] = f_i\left(\hat{y}_i[t]\right) \tag{5c}$$

These prediction steps can be intuitively visualized as a neural network. The next section formulates an optimization problem intended to learn the parameters of such a neural network. For a simple example with 2 sensors, the network structure is shown in Figure 3.

4 Problem formulation

The functional optimization problem consists in minimizing $\|z[t] - \hat{z}\|_2^2$ (where $z[t]$ is a vector collecting the measurements for all N sensors at time t), subject

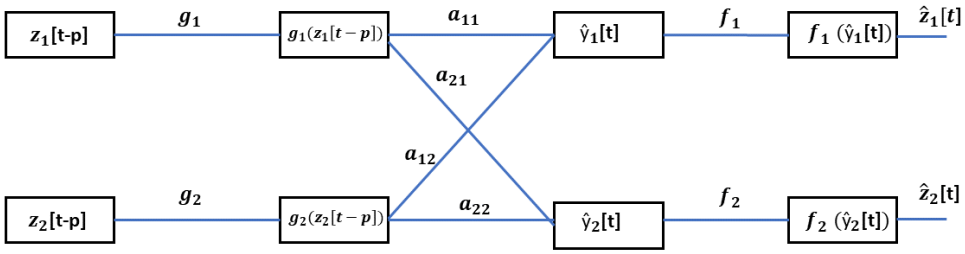

Fig. 3. Schematic for modeling Granger causality for a toy example with 2 sensors.

to the constraint of f_i being invertible $\forall i$, and the image of f_i being $(z_i, \bar{z}_i)$. The saturating values can be obtained from the nominal range of the corresponding sensors, or can be inferred from data.

Incorporating equation (1), the optimization problem can be written as:

$$\min_{f,A} \ \|z[t] - f\Big(\sum_{p=1}^{p} A^{(p)}\left[g(z[t-p])\right]\Big)\|_2^2 \tag{6a}$$

$$\text{s. to:} \ \sum_{j=1}^{M} \alpha_{ij} = \bar{z}_i - \underline{z}_i \ \forall i \tag{6b}$$

$$b_i = \bar{z}_i \ \forall i \tag{6c}$$

$$\alpha_{ij} \geq 0 \ \forall i, j \tag{6d}$$

$$w_{ij} \geq 0 \ \forall i, j \tag{6e}$$

The functional optimization over f_i is tantamount to optimizing over α_{ij},w_{ij}, k_{ij} and b_i. The main challenge to solve this problem is that there is no closed form for the inverse function g_i. This is addressed in the ensuing section.

5 Learning algorithm

Without a closed form for g, we cannot directly obtaining gradients with automatic differentiation such as Pytorch,as is typically done in deep learning with a stochastic gradient-based optimization algorithm. Fortunately, once $\{g_i(\cdot)\}$ is numerically evaluated, the gradient at that point can be calculated with a relatively simple algorithm, derived via implicit differentiation in Sec. 5.2. Once that gradient is available, the rest of the steps of the backpropagation algorithm are rather standard.

5.1 Forward equations

The forward propagation equations are given by the same steps that are used to predict next values of the time series z:

$$\tilde{y}_i[t-p] = g_i\left(z_i[t-p], \theta_i\right) \tag{7a}$$

$$\hat{y}_i[t] = \sum_{p=1}^{p}\sum_{j=1}^{n} a_{ij}^{(p)}\tilde{y}_j[t-p] \tag{7b}$$

$$\hat{z}_i[t] = f_i\left(\hat{y}_i[t], \theta_i\right) \tag{7c}$$

$$C[t] = \sum_{n=1}^{N}\left(z_n[t] - \hat{z}_n[t]\right)^2. \tag{7d}$$

Here, the dependency of the nonlinear functions with the neural network parameters is made explicit, where

$$\theta_i = \begin{bmatrix}\alpha_i \\ w_i \\ k_i \\ b_i\end{bmatrix} \text{ and } \alpha_i = \begin{bmatrix}\alpha_{i1} \\ \alpha_{i2} \\ \vdots \\ \alpha_{iM}\end{bmatrix}, w_i = \begin{bmatrix}k_{i1} \\ k_{i2} \\ \vdots \\ k_{iM}\end{bmatrix}, k_i = \begin{bmatrix}k_{i1} \\ k_{i2} \\ \vdots \\ k_{iM}\end{bmatrix}.$$

5.2 Backpropagation equations

The goal of backpropagation is to calculate the gradient of the cost function with respect to the VAR parameters and the node dependent function parameters θ_i.

The gradient of the cost is obtained by applying the chain rule as following:

$$\begin{gathered}\frac{dC[t]}{d\theta_i} = \sum_{n=1}^{N}\frac{\partial C}{\partial \hat{z}_n[t]}\frac{\hat{z}_n[t]}{\partial\theta_i} \\ \text{where } \frac{\partial C}{\partial \hat{z}_n[t]} = 2(\hat{z}_n[t] - z_n[t]) = S_n\end{gathered} \tag{8}$$

$$\frac{\partial \hat{z}_n[t]}{\partial\theta_i} = \frac{\partial f_n}{\partial \hat{y}_n}\frac{\partial \hat{y}_n}{\partial\theta_i} + \frac{\partial f_n}{\partial\theta_n}\frac{\partial\theta_n}{\partial\theta_i} \tag{9}$$

$$\text{where } \frac{\partial\theta_n}{\partial\theta_i} = \begin{cases} I, n = i \\ 0, n \neq i\end{cases}$$

Substituting equation (8) into (9) yields

$$\frac{dC[t]}{d\theta_i} = \sum_{n=1}^{N} S_n\left(\frac{\partial f_n}{\partial \hat{y}_n}\frac{\partial \hat{y}_n}{\partial\theta_i} + \frac{\partial f_n}{\partial\theta_n}\frac{\partial\theta_n}{\partial\theta_i}\right). \tag{10}$$

Equation(10) can be simplified as:

$$\frac{dC[t]}{d\theta_i} = S_i\frac{\partial f_i}{\partial\theta_i} + \sum_{n=1}^{N} S_n\frac{\partial f_n}{\partial \hat{y}_n}\frac{\partial \hat{y_n}}{\partial\theta_i}. \tag{11}$$

The next step is to derive $\frac{\partial \hat{y}_n}{\partial\theta_i}$ and $\frac{\partial f_i}{\partial\theta_i}$ of equation (11):

$$\frac{\partial \hat{y}_n[t]}{\partial \theta_i} = \sum_{p=1}^{P} \sum_{j=1}^{N} a_{nj}^{(p)} \frac{\partial}{\partial \theta_j} \tilde{y}_j[t-p] \frac{\partial \theta_j}{\partial \theta_i}. \tag{12}$$

With $f_i'(\hat{y}) = \frac{\partial f_i(\hat{y},\theta_i)}{\partial(\hat{y})}$, expanding $\tilde{y}_j[t-p]$ in equation (12) changes (11) to:

$$\frac{dC[t]}{d\theta_i} = S_i \left(\frac{\partial f_i}{\partial \theta_i} \right) + \sum_{n=1}^{N} S_n \left(f_n'(\hat{y}_n[t]) \sum_{p=1}^{P} a_{ni}^{(p)} \frac{\partial}{\partial \theta_i} g_i\left(z_i[t-p], \theta_i\right) \right). \tag{13}$$

Here, the vector

$$\frac{\partial f_i\left(\hat{y}, \theta_i\right)}{\partial \theta_i} = \left[\frac{\partial f_i\left(\hat{y}, \theta_i\right)}{\partial \alpha_i} \frac{\partial f_i\left(\hat{y}, \theta_i\right)}{\partial w_i} \frac{\partial f_i\left(\hat{y}, \theta_i\right)}{\partial k_i} \frac{\partial f_i\left(\hat{y}, \theta_i\right)}{\partial b_i} \right]$$

can be obtained by standard or automated differentiation via, e.g., Pytorch [20].

However, (13) involves the calculation of $\frac{\partial g_i(z,\theta_i)}{\partial \theta_i}$, which is not straightforward to obtain. Since $g_i(z)$ can be computed numerically, the derivative can be obtained by implicit differentiation, realizing that the composition of f_i and g_i remains invariant, so that its total derivative is zero:

$$\frac{d}{d\theta_i}\left[f_i\left(g_i\left(z, \theta_i\right), \theta_i\right)\right] = 0 \tag{14}$$

$$\Rightarrow \frac{\partial f_i\left(g_i\left(z, \theta_i\right), \theta_i\right)}{\partial g\left(z, \theta_i\right)} \frac{\partial g\left(z, \theta_i\right)}{\partial \theta_i} + \left. \frac{\partial f_i\left(\hat{y}, \theta_i\right)}{\partial \theta_i} \right|_{\hat{y}=g_i(z,\theta_i)} = 0 \tag{15}$$

$$\Rightarrow f_i'(g_i(z,\theta_i)) \frac{\partial g\left(z, \theta_i\right)}{\partial \theta_i} + \left. \frac{\partial f_i\left(\hat{y}, \theta_i\right)}{\partial \theta_i} \right|_{\hat{y}=g_i(z,\theta_i)} = 0 \tag{16}$$

$$\text{Hence } \frac{\partial g_i\left(z, \theta_i\right)}{\partial \theta_i} = -\left\{ f_i'(g_i(z,\theta_i)) \right\}^{-1} \left(\left. \frac{\partial f_i\left(\hat{y}, \theta_i\right)}{\partial \theta_i} \right|_{\hat{y}=g_i(z,\theta_i)} \right). \tag{17}$$

The gradient of C_T w.r.t. the VAR coefficient $a_{ij}^{(p)}$ is calculated as follows:

$$\frac{dC[t]}{da_{ij}^{(p)}} = \sum_{n=1}^{N} S_n \frac{\partial f_n}{\partial \hat{y}_n} \frac{\partial \hat{y}_n}{\partial a_{ij}^{(p)}} \tag{18}$$

$$\frac{\partial \hat{y}_n[t]}{\partial a_{ij}^{(p)}} = \frac{\partial}{\partial a_{ij}^{(p)}} \sum_{p'=1}^{P} \sum_{q=1}^{N} a_{nq}^{(p')} \tilde{y}_q[t-p]$$

$$\text{where } \frac{\partial a_{nq}^{(p')}}{\partial a_{ij}^{(p)}} = \begin{cases} 1, n = i, p = p', \text{ and } q = j \\ 0, \text{otherwise} \end{cases} \tag{19}$$

$$\frac{dC[t]}{da_{ij}^{(p)}} = S_i f_i'\left(\hat{y}_i[t]\right) \tilde{y}_j[t-p]. \tag{20}$$

Even though the backpropagation cannot be done in a fully automated way, it can be realized by implementing equations (17) and (13) after automatically obtaining the necessary expressions.

5.3 Parameter optimization

The elements in $\{A^{(p)}\}_{p=1}^{P}$, and $\{\theta_i\}_{i=1}^{N}$ can be seen as the parameters of a NN. Recall from Fig. 3 that the prediction procedure resembles a typical feedforward NN as it interleaves component-wise nonlinearities with multidimensional linear mappings. The only difference is that one of the layers computes the inverse of a given function, and its backward step has been derived. Moreover, the cost function in (6) is the mean squared error (MSE).

The aforementioned facts support the strategy of learning the parameters using state-of-the-art NN training techniques. A first implementation has been developed using stochastic gradient descent (SGD) and its adaptive-moment variant Adam [21]. Constraints (6b)-(6e) are imposed by projecting the output of the optimizer into the feasible set at each iteration.

The approach is flexible enough to be extended with neural training regularization techniques such as dropout or adding a penalty based on the L1 or L2 norm of the coefficients, to address the issue of over-fitting and/or promote sparsity. The batch normalization technique can be proposed to improve the training speed and stability.

6 Experiments

The experiments described in this section, intended to validate the proposed method, can be reproduced with the Python code which is available in GitHub at https://github.com/uia-wisenet/NonlinearVAR

A set of $N = 10$ sensors is simulated, and an underlying VAR process of order $P = 2$. The VAR parameter matrices are generated by drawing each weight i.i.d from a standard Gaussian distribution. Matrices $\{A^{(p)}\}_{p=0}^{P}$ are scaled down afterwards by a constant that ensures that the VAR process is stable [18].

The underlying process samples $\{y[t]\}_{t=1}^{T}$, where T = 1000, are generated as a realization of the aforementioned VAR process, and the simulated sensor observed values $\{z[t]\}_{t=1}^{T}$ are obtained as the output of nonlinear observation functions that are also randomly generated.

The proposed nonlinear VAR estimator is analyzed in a stationary setting, and compared to the VAR estimator of the same order. The training and test curves are shown in Fig. 4. It can be observed that, despite the overfitting, the proposed nonlinear model can explain the time series data with significantly lower error.

7 Conclusion

A method for inferring nonlinear VAR models has been proposed and validated. The modeling assumption that the observed data are the outputs of nodal nonlinearities applied to the individual time series of a linear VAR process lying in an unknown latent vector space. Since the number of parameters that determine the topology does not increase, the model interpretability remains the same as that

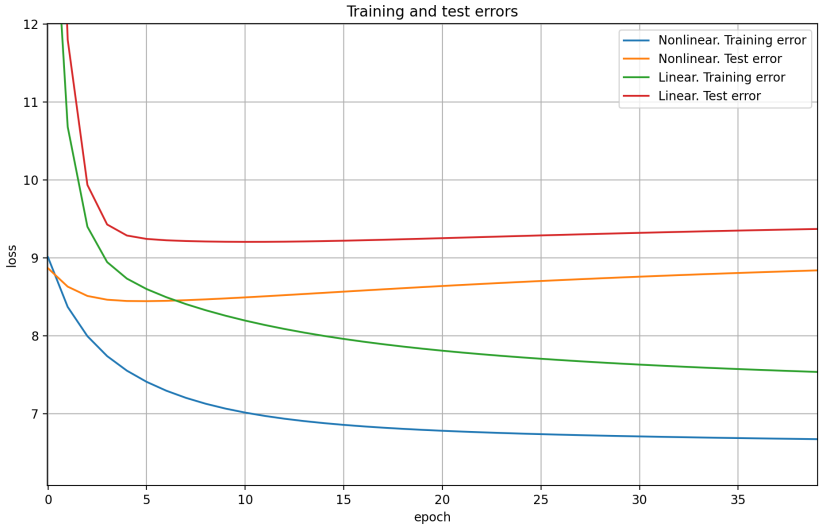

Fig. 4. Comparison of the proposed method (M=5, P=3) vs. a linear VAR model.

with linear VAR modeling, making the proposed model amenable for Granger causality testing and network topology identification. The optimization method, similar to that of DNN training, can be extended with state-of-the-art tools to accelerate training and avoid undesired effects such as convergence to unstable points and overfitting.

Acknowledgement: The authors would like to thank Emilio Ruiz Moreno for helping us manage a more elegant derivation of the gradient of $g_i(\cdot)$.

References

1. G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 787–807, 2018.
2. Z. Chen and S. V. Sarma, "Dynamic neuroscience," *Chen and SV Sarma, Eds. Cham: Springer International Publishing AG*, 2018.
3. A. Fujita, P. Severino, J. R. Sato, and S. Miyano, "Granger causality in systems biology: Modeling gene networks in time series microarray data using vector autoregressive models," in *Advances in Bioinformatics and Computational Biology*, C. E. Ferreira, S. Miyano, and P. F. Stadler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 13–24.
4. Y. Shen, G. B. Giannakis, and B. Baingana, "Nonlinear structural vector autoregressive models with application to directed brain networks," *IEEE Transactions on Signal Processing*, vol. 67, no. 20, pp. 5325–5339, 2019.

5. A. Tank, I. Cover, N. J. Foti, A. Shojaie, and E. B. Fox, "An interpretable and sparse neural network model for nonlinear granger causality discovery," *arXiv preprint arXiv:1711.08160*, 2017.
6. B. Zaman, L. M. L. Ramos, D. Romero, and B. Beferull-Lozano, "Online topology identification from vector autoregressive time series," *IEEE Transactions on Signal Processing*, 2020.
7. F. Nassif and S. Beheshti, "Automatic order selection in autoregressive modeling with application in eeg sleep-stage classification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5135–5139.
8. R. Zhou, J. Liu, S. Kumar, and D. P. Palomar, "Parameter estimation for student's t var model with missing data," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5145–5149.
9. V. N. Ioannidis, Y. Shen, and G. B. Giannakis, "Semi-blind inference of topologies and dynamical processes over dynamic graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2263–2274, 2019.
10. Y. Shen and G. B. Giannakis, "Online identification of directional graph topologies capturing dynamic and nonlinear dependencies," in *2018 IEEE Data Science Workshop (DSW)*, 2018, pp. 195–199.
11. R. Money, J. Krishnan, and B. Beferull-Lozano, "Online non-linear topology identification from graph-connected time series," *arXiv preprint arXiv:2104.00030*, 2021.
12. R. Farnoosh, M. Hajebi, and S. J. Mortazavi, "A semiparametric estimation for the nonlinear vector autoregressive time series model." *Applications & Applied Mathematics*, vol. 12, no. 1, 2017.
13. A. Tank, I. Covert, N. Foti, A. Shojaie, and E. B. Fox, "Neural granger causality," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 01, pp. 1–1, mar 2021.
14. H. Morioka, H. Hälvä, and A. Hyvarinen, "Independent innovation analysis for nonlinear vector autoregressive process," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 1549–1557.
15. M. Jin, M. Li, Y. Zheng, and L. Chi, "Searching correlated patterns from graph streams," *IEEE Access*, vol. 8, pp. 106 690–106 704, 2020.
16. F. Yanuar, "The estimation process in bayesian structural equation modeling approach," *Journal of Physics: Conference Series*, vol. 495, p. 012047, 04 2014.
17. W. Granger Clive, "Some recent developments in a concept of causality [j]," *Journal of Econometrics*, 1988.
18. H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*. Springer, 2005.
19. G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.
20. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
21. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.