Sparse Bayesian deep learning for dynamic system identification

Zhou, Hongpeng; Chahine, I.; Zheng, Wei Xing; Pan, Wei

DOI
10.1016/j.automatica.2022.110489

Publication date
2022

Document Version
Final published version

Published in
Automatica

Automatica 144 (2022) 110489

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Sparse Bayesian deep learning for dynamic system identification[☆]

Hongpeng Zhou[a], Chahine Ibrahim[a], Wei Xing Zheng[b], Wei Pan[c,a,*]

[a] Department of Cognitive Robotics, Delft University of Technology, Delft, 2628 CD, The Netherlands
[b] School of Computer, Data and Mathematical Sciences, Western Sydney University, Sydney, NSW 2751, Australia
[c] Department of Computer Science, The University of Manchester, Manchester, M13 9PL, United Kingdom

ARTICLE INFO

Article history:
Received 27 July 2021
Received in revised form 14 February 2022
Accepted 31 May 2022
Available online xxxx

Keywords:
Regularised system identification
Deep neural networks
Group sparsity
Sparse Bayesian learning

ABSTRACT

This paper proposes a sparse Bayesian treatment of deep neural networks (DNNs) for system identification. Although DNNs show impressive approximation ability in various fields, several challenges still exist for system identification problems. First, DNNs are known to be too complex that they can easily overfit the training data. Second, the selection of the input regressors for system identification is nontrivial. Third, uncertainty quantification of the model parameters and predictions are necessary. The proposed Bayesian approach offers a principled way to alleviate the above challenges by marginal likelihood/model evidence approximation and structured group sparsity-inducing priors construction. The identification algorithm is derived as an iterative regularised optimisation procedure that can be solved as efficiently as training typical DNNs. Remarkably, an efficient and recursive Hessian calculation method for each layer of DNNs is developed, turning the intractable training/optimisation process into a tractable one. Furthermore, a practical calculation approach based on the Monte-Carlo integration method is derived to quantify the uncertainty of the parameters and predictions. The effectiveness of the proposed Bayesian approach is demonstrated on several linear and nonlinear system identification benchmarks by achieving good and competitive simulation accuracy. The code to reproduce the experimental results is open-sourced and available online.

1. Introduction

System identification (SYSID) has a long history in natural and social sciences (Ljung, 1999b). Various approaches have been proposed for both linear/nonlinear systems and static/dynamical processes (Ayala, da Cruz, Freire, et al., 2014; Chen, Andersen, Ljung, et al., 2014; Chiuso & Pillonetto, 2012; M. Brunot & Carrillo, 2017). Among these, neural networks (NNs) are prominent black-box models and recently regained research interest in the SYSID community (Beintema, Toth, & Schoukens, 2021; Forgione & Piga, 2021; Gedon, Wahlström, Schön, & Ljung, 2021; Ljung, Andersson, Tiels, et al., 2020), thanks to the boom of deep learning.

The deep neural network (DNN) models have their advantages and disadvantages. An early paper on feed-forward NNs proved the universal approximation capabilities of any measurable function, using one hidden layer on a compact set (Hornik, Stinchcombe, & White, 1989). The training of DNN is mainly based on data which does not require much prior information about the system (LeCun et al., 2015). Several works also achieved competitive results by using feed-forward NNs (Leshno, Lin, Pinkus, et al., 1993) and recurrent neural networks (RNNs) (Delgado, Kambhampati, & Warwick, 1995; Weber & Gühmann, 2021) in the context of dynamical systems. However, it is not easy to design a proper NN structure. First of all, the trade-off between the model complexity and (simulation) prediction accuracy should be considered. An over-simplified model cannot reveal the underlying relation between input and output data. On the other hand, an over-complex model may overfit the training data, thus reducing its generalisation ability. Besides, the inevitable (non-Gaussian and non-additive) noise and non-smooth characteristics of some nonlinear processes may also cause the overfitting problem. Furthermore, NNs can also be underspecified by the data and constitute a large space of hypotheses for high-performing models (Wilson, 2020). Another challenging problem for SYSID is input regressor selection, which is defined as follows: given input regressors $z(t+1) = [u(t+1), u(t), \dots, u(t-l_u), y(t), y(t-1), \dots, y(t-l_y)]^\top \in \mathbb{R}^{l_u+l_y+1}$ with l_u and l_y denoting respectively the input and output lag, the most relevant input regressor features, which can explain the intrinsic phenomenon of the system,

[☆] This work is supported by China Scholarship Council No. 201706120017 (H.Zhou) and Huawei (W.Pan). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Dario Piga under the direction of Editor Alessandro Chiuso.

[*] Corresponding author at: Department of Computer Science, The University of Manchester, Manchester, M13 9PL, United Kingdom.

E-mail addresses: h.zhou-3@tudelft.nl (H. Zhou), ibrahim-chahine@outlook.com (C. Ibrahim), W.Zheng@westernsydney.edu.au (W.X. Zheng), wei.pan@manchester.ac.uk (W. Pan).

https://doi.org/10.1016/j.automatica.2022.110489
0005-1098/© 2022 Published by Elsevier Ltd.

are selected (Castellano & Fanelli, 2000). An effective input regressor selection can improve the prediction performance, and generalisation ability of the identified model.

For these challenges, the sparse Bayesian learning method offers a principled way to tackle them simultaneously: (a) A more efficient exploration of the hypothesis space (corresponding to saddle points) of NN models is possible (Wilson, 2020; Zhou, Yang, Wang, et al., 2019); (b) Over-fitting can be alleviated, and model redundancies can be eliminated through marginalisation and by choice of sparsity inducing prior distribution over parameters (MacKay, 1992); (c) Important input variables can be selected automatically by imposing structured sparsity on the NN; (d) Model parameters and prediction uncertainties can be quantified, which is particularly useful in decision making and safety-critical applications such as autonomous driving and structural health monitoring (Huang, Shao, Wu, et al., 2019).

Diverse Bayesian SYSID methods have been developed in the last decades. To name a few, a practical sparse Bayesian approach to state-space identification of nonlinear systems was proposed in Pan, Yuan, Gonçalves, et al. (2016) in the context of biochemical networks. A Bayesian identification algorithm of nonlinear autoregressive exogenous (NARX) models using variational inference with a demonstration on the electroactive polymer was introduced in Jacobs, T. Baldacchino, et al. (2018). A framework for identifying the governing interactions and transition logics of subsystems in cyber–physical systems was presented in Yuan, Tang, Zhou, et al. (2019) by using Bayesian inference and predefined basis functions. A variational expectation maximisation approach to SYSID when the data includes outliers was developed in Lindfors and Chen (2020). Two approaches to SYSID using Bayesian networks were proposed in Chiuso and Pillonetto (2012). The first one combines kernel-based stable spline and group Least Angle Regression while the other combines stable splines with the hyper-prior definition in a fully Bayesian model. However, this work did not discuss how to apply the Bayesian approach to the NN model. Another typical probabilistic nonparametric modelling method is the Gaussian process (GP), which can perform excellently for linear and nonlinear SYSID tasks, but suffers from the high computational burden for large datasets and cannot conduct input regressor selection efficiently. Overall, specific to the use of NNs as a model form, little attention has been given to the identification of dynamic systems in a Bayesian framework.

Several approaches have been proposed to treat the NNs in a Bayesian manner, e.g., Laplace approximation, expectation propagation, variational inference, etc. Among these methods, the Laplace approximation is an approximated inference approach that can only represent local properties but is closer in efficiency to maximum a posteriori (MAP) (MacKay, 1992). However, to update the posterior variance of parameters, the Laplace approximation method requires computing the inverse Hessian of log-likelihood, which is infeasible for large-scale NNs. To address this issue, a fast Hessian calculation technique was devised for convolutional NNs and successfully obtained an impressive image classification performance (Zhou et al., 2019).

In this paper, a companion technique for recurrent layers is also developed. Specifically, by unfolding a recurrent layer with its equivalent Fully Connected (FC) layers, the Hessian calculation of a recurrent layer can be treated as the Hessian calculation for the FC layers. Besides, since the Hessian is a diagonally dominant matrix (Martens & Grosse, 2015), we develop a recursive and efficient method to compute the diagonal blocks of the Hessian matrix. Each block represents the Hessian diagonal entries of each layer and can be calculated recursively along with a backward propagation through time (BPTT) process. It should be noted that the Hessian is a necessity for the Laplace approximation method and can accelerate the optimisation process. In this paper, by incorporating the Hessian information to update the loss function, it can be observed that the proposed Bayesian approach can converge faster than the conventional optimisation method without capturing the Hessian information. Similar rapid convergence is also observed in the previous works related to the second-order optimisation methods (A. Botev & Barber, 2017; Botev, 2020; Boyd & Vandenberghe, 2004; Nocedal, 1980).

In addition, a sparse Bayesian approach is proposed to address several challenges for system identification based on deep neural networks, including overfitting the training data, the selection of input regressors, and the uncertainty quantification of model parameters. We will consider two typical DNNs, i.e., Multi-Layer Perceptron (MLP) and Long Short-Term Memory networks (LSTM). The simulation error is adopted as the evaluation metric, which is a more challenging criterion compared with one-step-ahead prediction. The simulation error is equivalent to the N-step-ahead prediction error, with N denoting a user-defined temporal horizon. In order to identify the system in a Bayesian framework, the group priors are introduced over network parameters to induce structured sparsity, and the Laplace approximation is used to approximate the intractable integral of the evidence. The main contributions of this paper have four folds:

- A practical iterative algorithm using Bayesian deep learning is proposed for SYSID. The first identification cycle of the algorithm is equivalent to the conventional sparse group lasso regularisation method. This algorithm can be used with both MLP and LSTM networks for linear and nonlinear processes.
- An efficient Hessian calculation method is proposed for each layer of DNNs, both for MLPs and RNNs. By calculating the block-diagonal entry of the Hessian, the proposed method can turn an intractable training/optimisation procedure into a tractable one. The sparsification process is also accelerated by recursively updating the Hessian information.
- The structured sparsity is incorporated in the Bayesian formulation of the identification problem to alleviate the overfitting issue and select the input regressor. As a consequence, the number of hidden neurons in both MLP and LSTM networks can be significantly reduced.
- The proposed algorithm achieves good and competitive simulation accuracy on five benchmark datasets. The datasets of three linear processes are provided in the MATLAB System Identification Toolbox,[1] including the Hairdryer, Heat exchanger, and the Glass Tube manufacturing process. The datasets of two nonlinear processes are provided on the Nonlinear System Identification Benchmarks website,[2] including the Cascaded Tanks (Schoukens, Mattson, Wigren, et al., 2016) and Coupled Electric Drives (Wigren & Schoukens, 2017).

The organisation of this paper is as follows. Section 2 formulates the identification problem using DNNs and introduces the Bayesian approach. Section 3 presents the iterative procedure of the proposed sparse Bayesian learning algorithm and a recursive Hessian computation method. The illustration of structured sparsity regularisation, uncertainty quantification, and the proposed training algorithm are introduced in Section 5. The identification results and detailed analysis are given in Section 6. Section 7 concludes the paper. A discussion on the limitations and future work are also included in Section L of Appendix.

[1] https://nl.mathworks.com/help/ident/examples.html.

[2] https://sites.google.com/view/nonlinear-benchmark/.

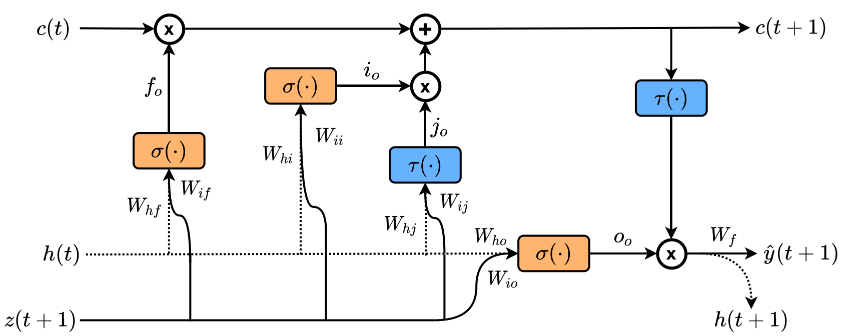


Fig. 1. Single layer long short term memory network.

2. Preliminaries

2.1. Problem formulation

The chosen mathematical model structure is generated by training the network $\mathtt{Net}(\mathcal{W}, z)$, where $\mathcal{W}$ represents an array of the weights in the network and z represents the input regressors of size $1 \times (l_y + l_u + 1)$. These are best defined by the prediction model:.

$$\hat{y}(t+1) = \mathtt{Net}(\mathcal{W}, z(t+1), \epsilon) \tag{1}$$

where ϵ represents the noise term. It should be noted that the ϵ can be in any distribution of exponential family. And the model parameter can be identified with a maximum likelihood method in the case of Gaussian noise (see Chapter 7.3 in Ljung (1999b)). The input regressor of the model is defined as a combination of lagged elements of the system input u and outputs y. The input lag is denoted l_u and output lag l_y, resulting in the expression $z(t+1) = [u(t+1), u(t), \dots, u(t-l_u), y(t), y(t-1), \dots, y(t-l_y)]^\top$. With such a network model, we aim to address two typical problems in SYSID. First, how to promote the sparsity of $\mathcal{W}$ to relieve the overfitting issue of DNNs? Second, how to select the input regressors automatically by identifying and removing the redundant features from $z(t+1)$?

The first DNN model considered is the LSTM network, a type of RNN. Benefiting from the advantages of processing sequential data and memorising information, LSTM can also be applied straightforwardly for dynamic SYSID (Delgado et al., 1995). The BPTT method is used to train LSTM, where the network is unfolded in time and weights are updated based on an accumulation of gradients across time steps (see Fig. 1).

The second DNN model considered is the MLP, a type of feedforward NN. Backpropagation with stochastic gradient descent algorithm and variations are often used to train a MLP network.

2.2. Learning in a Bayesian framework

Given a dataset $\mathcal{D} = (U, Y)$ where the input $U = [u(1), u(2), \dots, u(T)]$ and output $Y = [y(1), y(2), \dots, y(T)]$ with T referring to the number of samples, the posterior estimation for network weights $\mathcal{W}$ is given by Bayes' rule:

$$p(\mathcal{W}|\mathcal{D}, \mathcal{H}) = \frac{p(\mathcal{D}|\mathcal{W}, \mathcal{H})p(\mathcal{W}, \mathcal{H})}{p(\mathcal{D}|\mathcal{H})} \tag{2}$$

$p(\mathcal{D}|\mathcal{W}, \mathcal{H})$ designates the likelihood function, where $p(\mathcal{W}, \mathcal{H})$ denotes the prior over the weights $\mathcal{W}$ and $p(\mathcal{D}|\mathcal{H})$ is the evidence of the hypothesis $\mathcal{H}$ given $\mathcal{D}$. The hypothesis generally incorporates model and inference assumptions. For simplicity of notations, the hypothesis term is dropped in the rest of the paper. Assume that the likelihood function belongs to the exponential family:

$$\begin{aligned} p(\mathcal{D}|\mathcal{W}, \theta) &= g(\theta)\exp\Big\{\sum_{s=1}^{S} \eta_s(\mathcal{W}, \sigma)T_s(\sigma) + B(\mathcal{W})\Big\} \\ &= g(\theta)\exp\big\{-\mathbf{E}(\mathcal{W}, \theta)\big\} \end{aligned} \tag{3}$$

where $g(\cdot)$, $T_s(\cdot)$, $\eta_s(\cdot)$ and $B(\cdot)$ are known functions corresponding to a specific exponential family distribution, θ is the parameter of the family, and $\mathbf{E}(\mathcal{W}, \theta)$ denotes an energy function.

The prior probabilities $p(\mathcal{W})$ takes a Gaussian relaxed variational form $p(\mathcal{W}) \geq p(\mathcal{W}, \psi) = \mathcal{N}(\mathcal{W}|0, \Psi)\ \phi(\psi)$, where $\phi(\psi)$ represents the hyperprior probability of $\psi \triangleq [\psi^1_{11}, \dots, \psi^1_{n_1 1}, \dots, \psi^1_{n_1 n_2}, \dots, \psi^L_{11}, \dots, \psi^L_{n_{L-1}n_L}]$ and $\Psi \triangleq \mathrm{diag}(\psi)$. With the principle of minimising the misaligned probability mass, the hyper-parameter ψ can be obtained by

$$\begin{aligned} \hat{\psi} &= \arg\min_{\psi \geq 0} \int p(\mathcal{D}|\mathcal{W}, \theta)|p(\mathcal{W}) - p(\mathcal{W}, \psi)|\ d\mathcal{W} \\ &= \arg\max_{\psi \geq 0} \int p(\mathcal{D}|\mathcal{W}, \theta)p(\mathcal{W}, \psi)\ d\mathcal{W}. \end{aligned} \tag{4}$$

The resulting problem is known as a type II maximum likelihood. The integration is intractable and can be obtained by the Laplace approximation method, which is explained in detail in Section 3.1.

3. Sparse Bayesian deep learning

3.1. Laplace approximation

The Laplace approximation method is adopted to compute the intractable integral in Eq. (4). $\mathbf{L}(\mathcal{W}, \theta)$ can be approximated by a second-order Taylor series expansion around a set of connection weights $\mathcal{W}^*$ with the operator $\Delta\mathcal{W} = \mathcal{W} - \mathcal{W}^*$, so we have

$$\mathbf{E} \approx \mathbf{L}(\mathcal{W}^*, \theta) + \frac{1}{2}\Delta\mathcal{W}^T\mathbf{H}\Delta\mathcal{W} + \Delta\mathcal{W}^T\mathbf{g}. \tag{5}$$

The resulting expression for the likelihood in a compact form is given by

$$\begin{aligned} &p(\mathcal{D}|\mathcal{W}, \theta) = \mathbf{A}\big(\mathcal{W}^*, \theta\big)\exp\Big\{-\frac{1}{2}\mathcal{W}^T\mathbf{H}\mathcal{W} - \mathcal{W}^T\hat{\mathbf{g}}\Big\} \\ &\hat{\mathbf{g}}\big(\mathcal{W}^*, \theta\big) = \mathbf{g}\big(\mathcal{W}^*, \theta\big) - \mathbf{H}\big(\mathcal{W}^*, \theta\big)\mathcal{W}^* \end{aligned} \tag{6}$$

where $\mathbf{H}\big(\mathcal{W}^*, \theta\big)$ and $\mathbf{g}\big(\mathcal{W}^*, \theta\big)$ are respectively the Hessian and the gradient of the loss function $\mathbf{E}$ with respect to $\mathcal{W}$ at $\mathcal{W}^*$. Eq. (6) is obtained by grouping elements independent of the target variable $\mathcal{W}$ in $\mathbf{A}\big(\mathcal{W}^*, \theta\big)$. The approximated likelihood is an exponent of a quadratic function corresponding to the Taylor series expansion of the energy loss. This form can be recast into a Gaussian function. In effect of the conjugacy of the prior and posterior, the posterior $p(\mathcal{W}|\mathcal{D})$ is Gaussian given by:

$$p(\mathcal{W}|\mathcal{D}) = \mathcal{N}(\mathcal{W}|\mu_{\mathcal{W}}, \Sigma_{\mathcal{W}}) \tag{7}$$

$$\mu_{\mathcal{W}} = \Sigma_{\mathcal{W}}\hat{\mathbf{g}}, \quad \Sigma_{\mathcal{W}} = \left[\mathbf{H} + \Psi^{-1}\right]^{-1} \tag{8}$$

A more detailed derivation of the Laplace approximation is given in Appendix A (Zhou, Ibrahim, Zheng, & Pan, 2022).

3.2. Evidence maximisation

The evidence in Eq. (4) attempts to find the volume of the product $p(\mathcal{D}|\mathcal{W},\theta)p(\mathcal{W},\psi)$, which is Gaussian and proportional to the posterior. Thus, one can approximate the evidence as the volume around the most probable value (here posterior $\mu_{\mathcal{W}}$).

$$\hat{\psi} = \arg\max_{\psi\geq 0} \int p(\mathcal{D}|\mathcal{W},\theta)p(\mathcal{W}|\psi)p(\psi)d\mathcal{W} \tag{9}$$

$$\approx \arg\max_{\psi\geq 0} \underbrace{p(\mathcal{D}|\mu_{\mathcal{W}},\theta)}_{\text{Best Fit Likelihood}} \underbrace{p(\mu_{\mathcal{W}}|\psi)|\Sigma_{\mathcal{W}}|^{\frac{1}{2}}}_{\text{Occam Factor}}. \tag{10}$$

In David Mackay's words, the evidence is approximated by the product of the data likelihood given the most probable weights and the Occam factor (MacKay, 1992). It can also be interpreted as a Riemann approximation of the evidence, where the best-fit likelihood represents the peak of the evidence. And the Occam's factor is the Gaussian curvature around the peak.

By realising that the posterior mean $\mu_{\mathcal{W}}$ maximises $p(\mathcal{D}|\mathcal{W},\theta)$ $p(\mathcal{W}|\psi)$, Eq. (10) can be rewritten into a joint maximisation in $\mathcal{W}$ and ψ. By applying the $-2\log(\cdot)$ operation, the evidence maximisation in Eq. (4) can be recast into a joint minimisation of an objective function $\mathcal{L}(\mathcal{W},\psi,\theta)$ given by:

$$\begin{aligned}\mathcal{L}(\mathcal{W},\psi,\theta) =& \mathcal{W}^T\mathbf{H}\mathcal{W} + 2\mathcal{W}^T\hat{\mathbf{g}} + \mathcal{W}^T\Psi^{-1}\mathcal{W} + \log|\Psi| \\ &+ \log|\mathbf{H} + \Psi^{-1}| - T\log(2\pi\theta)\end{aligned} \tag{11}$$

For a more thorough mathematical derivation that leads to Eq. (11) and insight into the Laplace approximation, please refer to Appendix A and B (Zhou et al., 2022).

3.3. Convex–concave procedure

The objective function in Eq. (11) can be seen as a sum of a convex u and concave v functions in ψ with:

$$u(\mathcal{W},\psi) = \mathcal{W}^T\mathbf{H}\mathcal{W} + 2\mathcal{W}^T\hat{\mathbf{g}} + \mathcal{W}^T\Psi^{-1}\mathcal{W} \tag{12}$$

$$v(\psi) = \log|\Psi| + \log|\mathbf{H} + \Psi^{-1}|. \tag{13}$$

$\mathcal{W}^T\Psi^{-1}\mathcal{W}$ is positive definite, since $\psi > 0$. Thus, u is convex in Ψ. v can be reformulated as a log-determinant of an affine function of Ψ. By using the Schur complement determinant identity:

$$|\Psi||\mathbf{H} + \Psi^{-1}| = \begin{vmatrix}\mathbf{H} & \\ & -\Psi\end{vmatrix} = |\mathbf{H}||\mathbf{H}^{-1} + \Psi| \tag{14}$$

and taking the log of Eq. (14),

$$\log|\Psi| + \log|\mathbf{H} + \Psi^{-1}| = \log|\mathbf{H}| + \log|\mathbf{H}^{-1} + \Psi|$$

one finds an equivalent expression of v that is concave in Ψ. The minimisation problem can therefore be reformulated as a Convex–concave procedure (CCCP) (Chen et al., 2014). $\mathcal{W}$ and ψ are obtained by the iterative minimisation of Eqs. (15)–(16).

$$\mathcal{W}(k+1) = \arg\min_{\mathcal{W}} u\big(\mathcal{W}, \psi(k)\big) \tag{15}$$

$$\psi(k+1) = \arg\min_{\psi\geq 0} u\big(\mathcal{W}(k+1), \psi\big) + \alpha(k)\cdot\psi \tag{16}$$

where $\alpha(k) = \nabla_{\psi} v\big(\psi(k)\big)^T$ is the gradient of v evaluated at the current iterate $\psi(k)$. Using the chain rule, its analytical form is given by:

$$\begin{aligned}\alpha(k) =& \nabla_{\psi}\Big(\log|\Psi| + \log|\mathbf{H} + \Psi^{-1}|\Big)\Big|_{\psi=\psi(k)} \\ =& -\operatorname{diag}\Big(\Psi^{-1}(k)\Big) \odot \operatorname{diag}\Big(\big(\mathbf{H} + \Psi^{-1}(k)\big)^{-1}\Big) \\ & \odot \operatorname{diag}\Big(\Psi^{-1}(k)\Big) + \operatorname{diag}\Big(\Psi^{-1}(k)\Big)\end{aligned} \tag{17}$$

$\odot$ is the point-wise Hadamard product. Since Ψ is a diagonal matrix, Eq. (16) can be expressed per connection independently. With $\Sigma_{W^l_{ab}}(k)$ being the connection weight posterior variance, the analytical form for α is:

$$\Sigma_{\mathcal{W}}(k) = \big(\mathbf{H}(k) + \Psi(k)^{-1}\big)^{-1} \tag{18}$$

$$\alpha^l_{ab}(k) = -\frac{\Sigma_{W^l_{ab}}(k)}{\psi^l_{ab}(k)^2} + \frac{1}{\psi^l_{ab}(k)}. \tag{19}$$

The optimisation step in Eq. (16) for ψ^l_{ab} becomes

$$\psi^l_{ab}(k+1) = \arg\min_{\psi\geq 0} \frac{W^l_{ab}(k+1)^2}{\psi} + \alpha^l_{ab}(k)\cdot\psi. \tag{20}$$

By noting that

$$\frac{{W^l_{ab}}^2}{\psi} + \alpha^l_{ab}\cdot\psi \geq 2\left|\sqrt{\alpha^l_{ab}}\cdot W^l_{ab}\right| \tag{21}$$

the analytical solution is given by

$$\psi^l_{ab}(k+1) = \frac{|W^l_{ab}(k+1)|}{\omega^l_{ab}(k)}$$

where $\omega^l_{ab}(k) = \sqrt{\alpha^l_{ab}(k)}$.

For the second part, finding $\mathcal{W}$ can be done with stochastic gradient descent on Eq. (15), which can be reformulated as the minimisation of a regularised loss function as follows:

$$\begin{aligned}\mathcal{W}(k+1) =& \arg\min_{\mathcal{W}} \mathbf{L} = \arg\min_{\mathcal{W}} \mathcal{W}^T\mathbf{H}\mathcal{W} + 2\mathcal{W}^T\hat{\mathbf{g}} \\ &+ \sum_{l=1}^{L}\sum_{a=1}^{n_{l-1}}\sum_{b=1}^{n_l} \|\omega^l_{ab}\cdot W^l_{ab}\|_{l_1}\end{aligned} \tag{22}$$

$$\approx \arg\min_{W} \mathbf{E}(\cdot) + \lambda\sum_{l=1}^{L}\rho(\omega^l, W^l). \tag{23}$$

$\mathbf{E}(\cdot)$ designates the energy loss function defined in Eq. (3) and $\rho(\cdot)$ is the regularisation term.

4. Hessian computation

4.1. Definitions and properties of the hessian

For a DNN model, the Hessian of a weight matrix $W \in \mathbb{R}^{m\times n}$ is a square matrix of the second-order of partial derivatives of the loss function and can be formulated as:

$$\mathbf{H}_W = \begin{bmatrix} \frac{\partial^2\mathbf{L}}{\partial\vec{W}_1^2} & \frac{\partial^2\mathbf{L}}{\partial\vec{W}_1\partial\vec{W}_2} & \cdots & \frac{\partial^2\mathbf{L}}{\partial\vec{W}_1\partial\vec{W}_{mn}} \\ \frac{\partial^2\mathbf{L}}{\partial\vec{W}_2\partial\vec{W}_1} & \frac{\partial^2\mathbf{L}}{\partial\vec{W}_2^2} & \cdots & \frac{\partial^2\mathbf{L}}{\partial\vec{W}_2\partial\vec{W}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\mathbf{L}}{\partial\vec{W}_{mn}\partial\vec{W}_1} & \frac{\partial^2\mathbf{L}}{\partial\vec{W}_{mn}\partial\vec{W}_2} & \cdots & \frac{\partial^2\mathbf{L}}{\partial\vec{W}_{mn,mn}^2} \end{bmatrix} \tag{24}$$

So the (i,j) element of $\mathbf{H}_W$ is:

$$[\mathbf{H}_W]_{ij} = \frac{\partial^2\mathbf{L}}{\partial\vec{W}_i\partial\vec{W}_j} \tag{25}$$

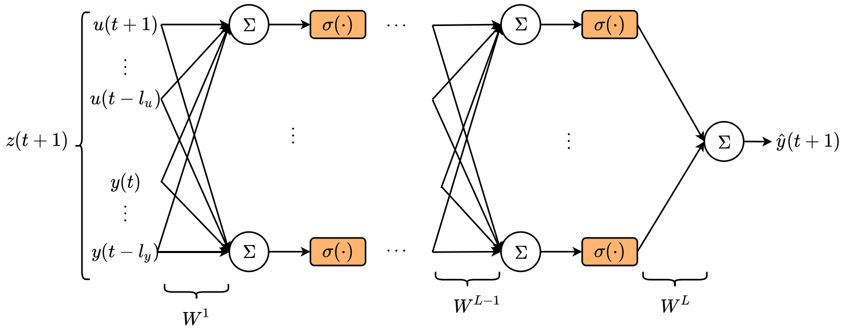


Fig. 2. Multi-Layer Perceptron with L layers.

where $\vec{W} \in \mathbb{R}^{mn}$ is the vectorisation of the multi-dimensional weight matrix $W \in \mathbb{R}^{m\times n}$. As the dimension of the Hessian is the square of the number of unknown parameters ($\mathbf{H}_W \in \mathbb{R}^{mn\times mn}$), it would be convenient to conduct the Hessian calculation by treating the matrix as a vector (the vectorisation operator is defined in Definition 1 of Appendix D (Zhou et al., 2022)).

The Hessian information can benefit the training of DNNs from two aspects. First, it can accelerate the optimisation process. Several previous works on second-order optimisation methods (e.g., the Quasi-Newton methods (Boyd & Vandenberghe, 2004; Nocedal, 1980)) have presented that by incorporating the Hessian information in the optimisation process, the rapid convergence can be obtained without a lot of tuning work (A. Botev & Barber, 2017; Botev, 2020). Besides, Martens et al. (2010) demonstrated that the Hessian information, also known as curvature matrix, could address the typical pathological curvature problem, where the first-order optimisation method often falls into the "canyon" with large varying curvature because of their lack of ability to capture the curvature information (Dauphin et al., 2014; Martens & Grosse, 2015). Second, the Hessian of the weight matrix is a required component for the Laplace approximation method. The Hessian is not only used to calculate the posterior distribution of weight parameters as in Eq. (8) but also used to update the loss function in each cycle (see Eqs. (17)– (23).)

However, as the dimension of the Hessian is the square of the number of parameters, the calculation and storage of the Hessian for large-scale neural networks are infeasible considering their millions of parameters or more (Botev, 2020). To address this problem, an efficient Hessian calculation method for a FC layer was in presented (A. Botev & Barber, 2017; Botev, 2020). The proposed method therein can compute the diagonal blocks of the Hessian, where each block represents the diagonal entries of the Hessian in each layer and can be calculated recursively along with the back-propagation process using Kronecker products.

Inspired by this method (A. Botev & Barber, 2017; Botev, 2020) and the diagonal dominant feature of the Hessian (Martens & Grosse, 2015), we develop two efficient and recursive block-diagonal calculation methods for the Hessian computation of FC layer and recurrent layer in this section.

4.2. Compute the Hessian of fully-connected layer

Given a MLP as shown in Fig. 2, the output of the hidden layer l can be calculated as:

$$h^l = W^l a^{l-1} + b^l, \quad a^l = \sigma(h^l) \tag{26}$$

where b^l is the bias, $\sigma(\cdot)$ is the nonlinear activation function. The superscript l denotes the layer index. a^l and h^l represent the activation value and the pre-activation value, respectively. With these definitions, the proposed Hessian calculation method for a FC layer is summarised in Lemma 1.

Lemma 1. *For a fully-connected layer, given the activation function $\sigma(\cdot)$, the activation value a^l, a^{l-1} and the pre-activation value h^l, the Hessian of the weight matrix W^l is calculated recursively as follows:*

$$\mathbf{H}^l = \mathrm{diag}((a^l)^2 \otimes H^l) \tag{27}$$

where $\otimes$ stands for Kronecker product. H^l is the pre-activation Hessian and is updated as:

$$H^l = (B^l)^2 \odot \left(\left((W^{l+1})^\top\right)^2 H^{l+1}\right) + D^l \tag{28}$$

in which B^l and D^l are defined as:

$$B^l = \sigma'(h^l), \quad D^l = \sigma''(h^l) \odot \frac{\partial L}{\partial a^l} \tag{29}$$

where $\odot$ represents the element-wise multiplication.

The above procedures can be calculated along with a backward propagation process.

Remark 1. It should be noted that Lemma 1 is a modification of the Hessian calculation method proposed in A. Botev and Barber (2017). The proposed approach can be computed more efficiently. Specifically, if the Hessian of a FC layer is computed as Eq. (27)–(29), then the multiply accumulate operation (MACs) for the pre-activation Hessian H and Hessian $\mathbf{H}$ could be reduced from $n(2m^2 + 2n^2 + 4mn + 3m - 1)$ to $n(2 + 4\ m)$ with $W \in \mathbb{R}^{m\times n}$ (e.g., if $n = 100, m = 100$, then the original method requires 107.97×10^6 MACs compared with only 0.04×10^6 MACs for the approximate method.). Lemma 1 is also the inspiration of the proposed Hessian calculation method for a recurrent layer. We will revisit Lemma 1 many times in the following.

4.3. Compute the Hessian of recurrent layer

The challenge of the Hessian calculation for a recurrent layer comes from the recurrent operation, where the weight matrices in a RNN cell will be revisited iteratively through time (Martens, Ba, & Johnson, 2018). This behaviour is different to the FC layer, where the weight matrices only join once through the operation in a forward propagation process. Since a LSTM cell is a special form of the RNN, for the convenience of explanation, we use a simplified RNN structure to illustrate the Hessian calculation process. As shown in Fig. 3, we denote $z(t)$, $h(t)$ and $y(t)$ as the input, hidden state and output of the time step t, respectively. The behaviour of this RNN layer can be described by

$$h(t) = \sigma\left(\bar{h}(t)\right) = \sigma(W_i z(t) + W_h h(t-1)) \tag{30}$$

$$y(t) = g(W_o h(t)) \tag{31}$$

where W_i, W_h and W_o represent the weight matrix of the input layer, hidden layer and output layer, respectively, and σ is the activation function.

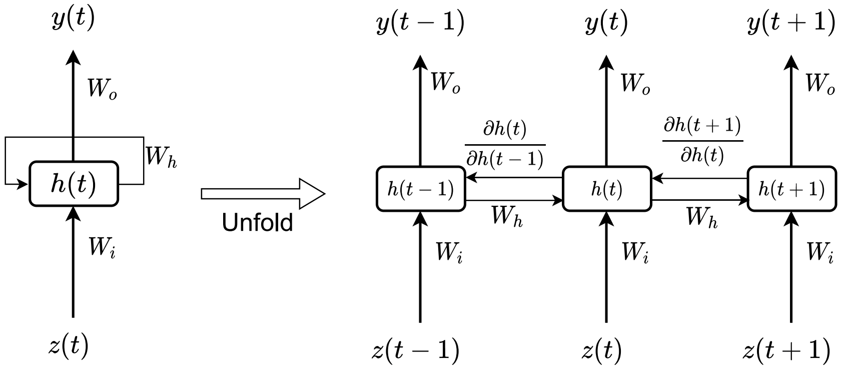


Fig. 3. An unrolled RNN layer.

It can be found that an unrolled RNN layer can be unfolded as several FC layer. Therefore the Hessian calculation for a recurrent layer can be regarded as the calculation of its equivalent FC layer. Inspired by Lemma 1, we propose a recursive and efficient method to compute the Hessian of a recurrent layer as follows.

Lemma 2. *For a recurrent layer, given σ representing the activation function, τ representing backward propagation time horizon, T representing the number of data samples, $z(t)$, $h(t)$ and $y(t)$ representing the input, hidden state and output at the time step t, W_i, W_h and W_o representing the weight matrix of the input layer, hidden layer and output layer, the Hessian of W_i, W_h, W_o within the RNN layer is calculated as follows:*

(1) The Hessian for W_o is:

$$\mathbf{H}_o = \frac{1}{T}\sum_{t=1}^{T} \mathbf{H}_o^{\top}, \quad \mathbf{H}_o^{\top} = h(t)^2 \otimes H_o^{\top} \tag{32}$$

where $H_o^{\top}$ is the pre-activation Hessian.

(2) The Hessian for W_h is:

$$\mathbf{H}_h = \mathbb{E}\left(\sum_{t=1}^{T} \sum_{j=\max(1,t-\tau+1)}^{t} \mathbf{H}_h^{t,j}\right) \tag{33}$$

$$\mathbf{H}_h^{t,j} = h(j-1)^2 \otimes H_h^{t,j} \tag{34}$$

where $\mathbf{H}_h^{t,j}$ and $H_h^{t,j}$ represent the Hessian and the pre-activation Hessian, respectively. In particular, $H_h^{t,j} = B_h^2 \odot \left(\left(W_h^{\top}\right)^2 H_h^{t,j+1}\right) + D_h$, where $B_h = \sigma'(\bar{h}(j))$, $D_h = \sigma''(\bar{h}(j)) \odot \frac{\partial L}{\partial h(j)}$.

(3) The Hessian for W_i is:

$$\mathbf{H}_i = \mathbb{E}\left(\sum_{t=1}^{T} \sum_{k=\max(1,t-\tau+1)}^{t} \mathbf{H}_i^{t,k}\right) \tag{35}$$

$$\mathbf{H}_i^{t,k} = (z(k))^2 \otimes H_i^{t,k} \tag{36}$$

where $H_i^{t,k} = \prod_{j=k+1}^{t} B_i^2 \odot \left(\left((W_i)^{\top}\right)^2 H_i^{j-1,j}\right)$ with $B_i = \sigma'(\bar{h}(j))$.

The above procedures can be calculated along with a BPTT process.

It should be noted that Lemmas 1 and 2 elaborate the detailed procedures to calculate the Hessian with respect to a single data sample (i.e., $T = 1$). If the number of data points is more than 1 (i.e., $T > 1$), the Hessian is calculated by averaging the Hessian of an individual data sample. The detailed proof of Lemmas 1 and 2 are given in Section D.1 and Section D.2 of Appendix D (Zhou et al., 2022).

5. Regularised identification algorithm

5.1. Input regressor selection and structured sparsity regularisation

As illustrated in Section 2.1, the input regressor is $z(t+1) = [u(t+1), u(t), \ldots, u(t-l_u), y(t), y(t-1), \ldots, y(t-l_y)]^{\top} \in \mathbb{R}^{l_u+l_y+1}$. The feature selection means identifying and removing the redundant features from $z(t+1)$. The proposed method can select the input regressors automatically by imposing structured sparsity regularisation on the DNN.

Specifically, the iterative procedure derived in Section 3 includes an assumption on the independence and non-stationarity of connection weights, resulting in a shape-wise regularisation as shown in Fig. 4(a). This drives the individual connection weight to 0. In some applications, one may want to enforce more structured sparsity by pre-defining groups and re-expressing the regularisation term as a function of these groups (Zhou et al., 2019). This paper uses a structured regularisation of rows and columns (Fig. 4(b–d)). The benefits of such an approach, specific to this paper, are obtaining compact sparse models and the suppression of input nodes in z that are deemed less pertinent without loss of accuracy. The reduction in the dimensionality of the input vector z represents the selection of input regressors.

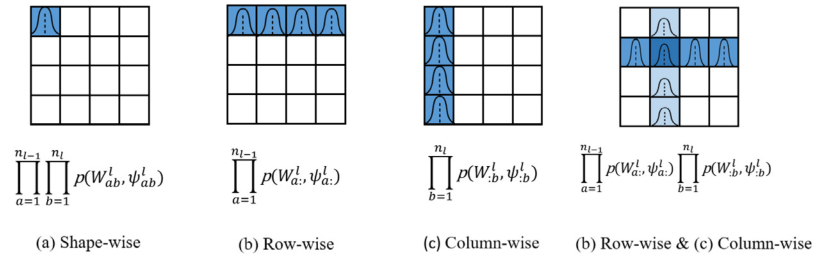
To extend this approach to the Bayesian framework, one has to revisit the prior formulation. The prior of a weight matrix is formulated based on the designated group of weight matrices (row or column or both). These groups are considered independent, but the connection weights of a specific group share the same prior Gaussian relaxation (see Fig. 4(b–d)). This results in a slightly different iterative update rule for the identification algorithm.

For each of the cases shown in Fig. 4, the update rules for ψ, ω and the regularisation function ρ are given in Appendix C Table C.1 (Zhou et al., 2022). There also stands more insight into how the adopted group priors slightly change the regularisation update rules on group Lasso regularisers (Simon, Friedman, Hastie, et al., 2013).

5.2. Algorithm

A pseudocode for the iterative procedure is given by Algorithm 1.

Remark 2. We now give some clarifications on the definition of cycle and epoch in Algorithm 1. One identification "cycle" has $E_{\max}$ epochs. One "epoch" refers to that the entire dataset is processed forward and backward by the NN for one time. In the first identification cycle, the regularisation is conventional ($\omega(0) = \mathbb{1}$).


$\prod_{a=1}^{n_{l-1}}\prod_{b=1}^{n_l} p(W^l_{ab}, \psi^l_{ab})$
$\prod_{a=1}^{n_{l-1}} p(W^l_{a:}, \psi^l_{a:})$
$\prod_{b=1}^{n_l} p(W^l_{:b}, \psi^l_{:b})$
$\prod_{a=1}^{n_{l-1}} p(W^l_{a:}, \psi^l_{a:}) \prod_{b=1}^{n_l} p(W^l_{:b}, \psi^l_{:b})$

(a) Shape-wise (b) Row-wise (c) Column-wise (b) Row-wise & (c) Column-wise

Fig. 4. Priors for structured sparsity of weight matrices. l is the layer index. a and b denote the row and column index of a 2-D weight matrix, respectively.

Algorithm 1 Identification Algorithm

```
Input:  • Collect input–output data $u(t)$ and $y(t)$ for $t = 1, 2, \cdots, T$.
        • Arrange input regressors according to the chosen lags $l_u, l_y$.
        • Set regularisation parameter $\lambda$ (empirically tuned) and DNN pruning thresholds $\kappa_\psi, \kappa_W$ ($\approx 10^{-3}$).
        • Set the number of repeated experiments $M$, identification cycles $C_{\max}$ and the number of epochs in each cycle $E_{\max}$.
        • Initialise hyper-parameters $\Psi(0) = \mathbf{I}$ and $\omega(0) = \mathbb{1}$.
Output: Return the set of connection weights $\mathcal{W}$
  for $m = 1$ to $M$ do
    for $c = 1$ to $C_{\max}$ do
      for $e = 1$ to $E_{\max}$ do
        1 Stochastic Gradient Descent with loss function ($\rho$ is defined in Table C.1):
          $$\mathcal{W}(k+1) = \min_{\mathcal{W}} \mathbf{E}(\cdot) + \lambda \sum_{i=1}^{N} \boldsymbol{\rho}(\omega^l, W^l) \qquad (37)$$
      end for
      2 Update $\alpha$ according to Eqs. (18)-(19)
      3 Update $\psi$ and $\omega$ according to Table C.1
      4 Dynamic pruning:
      if $\psi^l_{ab}(k) < \kappa_\psi$ or $|W^l_{ab}(k)| < \kappa_W$ then
        prune $W^l_{ab}(k)$
      end if
    end for
    Simulate on the validation data and choose the model with the smallest root mean square error (RMSE).
  end for
```

That is, the first obtained model is a sparse model corresponding to the conventional sparse group lasso regularisation method (as shown in (Eq. (37))), and sparser models are expected to result from the subsequent identification cycles.

Remark 3. The proposed algorithm shares the local convergence properties (local minima, saddle point) of the adopted stochastic gradient descent method (M. Zhou & Jin, 2021). This is because the Laplace approximation is a local approximation method and includes an assumption on the unimodality of the posterior. However, the pruning and regularisation techniques introduced are heuristics that help speed up the algorithm and improve convergence and optimality. Nonetheless, the identification experiments are run multiple times with different initialisations. The identified model with the best simulation accuracy is chosen.

5.3. Making predictions with uncertainties

In the Bayesian procedure, predictions are made using the posterior predictive distribution, which is given by:

$$p(\hat{y}|z, \mathcal{D}) = \int p(\hat{y}|\mathcal{W}, z)\; p(\mathcal{W}|\mathcal{D}) d\mathcal{W}. \tag{38}$$

The first term of the integral is the likelihood of the prediction conditional on the network parameters. The second term is the inferred posterior distribution over the weights $\mathcal{W}$, which can be calculated as Eq (7). The expected value of the prediction is:

$$\begin{aligned}\mathbb{E}\big[\hat{y}\big] &= \int \hat{y}\; p(\hat{y}|z, \mathcal{D}) d\hat{y} \\ &= \int \Big(\int \hat{y}\; p(\hat{y}|\mathcal{W}, z) d\hat{y}\Big)\; p(\mathcal{W}|\mathcal{D}) d\mathcal{W} \\ &= \int \mathtt{Net}(\mathcal{W}, z)\; p(\mathcal{W}|\mathcal{D}) d\mathcal{W}\end{aligned} \tag{39}$$

Using the inferred posterior distribution over the weights, one can approximate this integral by the Monte-Carlo sampling method. An unbiased estimate of the prediction is given by the average predictions using $\mathcal{W}$ sampled by the posterior M times as below:

$$\mu_{\hat{y}} \approx \frac{1}{M} \sum_{m=1}^{M} \mathtt{Net}(\mathcal{W}(m), z). \tag{40}$$

In an analogous way, to estimate the variance in the posterior predictive distribution, the expected value $\mathbb{E}\big[\hat{y}^T\hat{y}\big]$ is analytically derived as follows:

$$\begin{aligned}\mathbb{E}\big[\hat{y}^T\hat{y}\big] &= \int \hat{y}^T\hat{y}\; p(\hat{y}|z, \mathcal{D}) d\hat{y} \\ &= \int \Big(\int \hat{y}^T\hat{y}\; p(\hat{y}|\mathcal{W}, z) d\hat{y}\Big)\; p(\mathcal{W}|\mathcal{D}) d\mathcal{W} \\ &= \int \Big(\zeta + \mathtt{Net}(\mathcal{W}, z)^2\Big)\; p(\mathcal{W}|\mathcal{D}) d\mathcal{W}.\end{aligned} \tag{41}$$

where ζ represents the aleatoric uncertainty. An unbiased estimate of the variance is given by Monte-Carlo integration methods (Gal, 2016), with M samples from the inferred posterior distribution of $\mathcal{W}$ as below:

$$\boldsymbol{\Sigma}_{\hat{y}} \approx \zeta + \frac{1}{M} \sum_{m=1}^{M} \mathtt{Net}(\mathcal{W}(m), z)^2 - \mu^T_{\hat{y}}\mu_{\hat{y}}. \tag{42}$$

This variance (Eq. (42)) represents the model uncertainty in the prediction. It is approximated by the sum of an aleatoric uncertainty and epistemic uncertainty. The aleatoric uncertainty is generally known to be irreducible corresponding to the noise covariance of the measurement and is generally incorporated in the likelihood form (Gal, 2016). For example, if the likelihood

Table 1
Models are trained to identify linear and nonlinear processes with validation information.

Process-Model	Layers-Units	Lags	$\mathtt{RMSE}_{\mathtt{val}}$ ($\mu \pm \sigma$)	$\mathtt{RMSE}_{\mathtt{val}}$ ($\mathtt{min}$)	Sparsity	Appendix
Hairdryer-MLP	1–50	5	0.074 ± 0.0005	0.073	88.1%	Appendix F
Hairdryer-LSTM	1–10	5	0.093 ± 0.0166	0.081	93.5%	Appendix F
Heat Exchanger-MLP	1–50	150	0.086 ± 0.0002	0.086	99.3%	Appendix G
Heat Exchanger-LSTM	1–10	150	0.114 ± 0.0299	0.088	96.4%	Appendix G
GT Manufacturing-MLP	1–50	5	0.660 ± 0.0013	0.657	97.8%	Appendix H
GT Manufacturing-LSTM	1–10	5	0.671 ± 0.0019	0.669	99.0%	Appendix H
Cascaded Tanks-MLP	3–10	20	0.428 ± 0.1032	0.257	84.5%	Appendix I
Cascaded Tanks-LSTM	1–50	20	0.500 ± 0.1012	0.362	60.3%	Appendix I
CED-MLP	2–50	10	0.187 ± 0.0285 0.134 ± 0.0192	0.149 0.120	78.4%	Appendix J
CED-LSTM	1–10	10	0.155 ± 0.0257 0.126 ± 0.0201	0.121 0.097	72.8%	Appendix J

is given as Gaussian distribution, then ζ should represent the noise variance. The epistemic uncertainty corresponds to the model's uncertainty in a prediction that is often called reducible uncertainty (Gal, 2016) and grows when moving away from the training data (Wilson, 2020).

6. Experiments

An overview of the simulation accuracy of our experiments compared with other methods can be found in Tables E.1–E.2 in Appendix E (Zhou et al., 2022). The code to reproduce the experimental results is open-sourced and available online.[3]

6.1. Dataset and experiment setup

This section is to summarise the identification experiments of three linear processes and two nonlinear processes using the proposed algorithm. For linear systems, the identification procedure is repeated $M = 20$ times with $C_{\max} = 6$ identification cycles. For nonlinear systems, the identification is also repeated $M = 20$ times but with $C_{\max} = 10$ identification cycles each. Table 1 provides a summary of the model structure used for identification as well as the mean, standard deviation, and minimum validation $\mathtt{RMSE}$ of the M best-identified models and the percentage of sparse parameters in the best-identified model. In Appendix F–Appendix J (Zhou et al., 2022), the benchmarks are described more thoroughly with sparsity plots, simulation plots, and posterior predictive mean and uncertainty plots corresponding to the best-identified model.

Three linear processes are identified, the Hairdryer, Heat exchanger and Glass Tube (GT) manufacturing process. The datasets of these processes are provided by Matlab in the corresponding tutorials (https://nl.mathworks.com/help/ident/examples.html) on linear SYSID. The chosen best validated models are compared to the methods used in the corresponding tutorials. Additional model structures used for the identification of the Hairdryer are taken from Chapter 17.3 of Ljung (1999a) and run in Matlab. The comparisons are in Appendix E Table E.1 (Zhou et al., 2022).

Two nonlinear processes, the Cascaded Tanks (Schoukens et al., 2016), Coupled Electric Drives (Wigren & Schoukens, 2017) are also identified. Information and datasets of these benchmarks are compiled on the web page of the Workshop on Nonlinear System Identification Benchmarks (https://sites.google.com/view/nonlinear-benchmark/). The cascaded tank system is a fluid level control system consisting of two tanks with free outlets fed by a water pump (Schoukens et al., 2016). The fluid levels of these two tanks are adjusted by the input signal that controls the water pump. The coupled electric drive is a system that drives a pulley by controlling a flexible belt. Two electric motors provide the driving force, and the spring is used to fix the pulley. A more detailed description of the system and datasets of these benchmarks are compiled on the web page of the Nonlinear System Identification Benchmarks. The models with the best validation performance are compared with the best models obtained using conventional NN methods for multiple experiments ($M = 20$) and the previous works in the literature for every benchmark in Appendix E Table E.2 (Zhou et al., 2022).

6.2. Analysis of experimental results

In this subsection, the results will be discussed and analysed concerning the claims made on sparsity, uncertainty quantification, and simulation accuracy.

Sparsity: In most cases, the obtained networks are sparse models with structured sparsity. For example, Fig. 5 is a sparsity plot of the Heat Exchanger identified LSTM model, where half of the weight matrices related to hidden states are removed from the input gate (W_{hi}, W_{hj}) and forget gate (W_{hf}). According to Table 1, sparsity is more prominent in the identified linear systems than in nonlinear systems. This demonstrates that the nonlinearity that the data exhibits requires a higher complexity than in the linear case.

Starting with the linear systems, one can note that structured sparsity induces a recognised transport delay in the Heat Exchanger MLP and LSTM models, which characterises this system. Furthermore, the LSTM models for linear systems have complete operators pruned. This means that the cell state can be well regulated with fewer parameters than imposed by the initialised model structure in the Heat Exchanger case. Similar behaviour is seen across linear benchmarks.

Structured sparsity is also observed in the identified networks for nonlinear systems (Table 1). In addition to that, similar to LSTM models identified for linear systems, a lot of parameters involving the hidden states are pruned. A possible explanation for this behaviour is that the hidden states of LSTM units attempt to retain short-term information from the time series that is also available as lagged elements in the input regressor. The simulation result further shows that the input regressor with lagged elements can achieve better simulation performance for a LSTM model (see Appendix E Table E.1-E.2 (Zhou et al., 2022)). Another observation related to the structured sparsity regularisation is the effect of input regressor selection. As shown in Fig. H.2a in Appendix H (Zhou et al., 2022), the number of input regressors is reduced from 40 to 2 after applying the sparse Bayesian algorithm with row-wise and column-wise prior as shown in Appendix C Table C.1. The redundant input regressors are also identified for

[3] https://github.com/hongpengzhou/Deep-Bayesian-System-Identification.

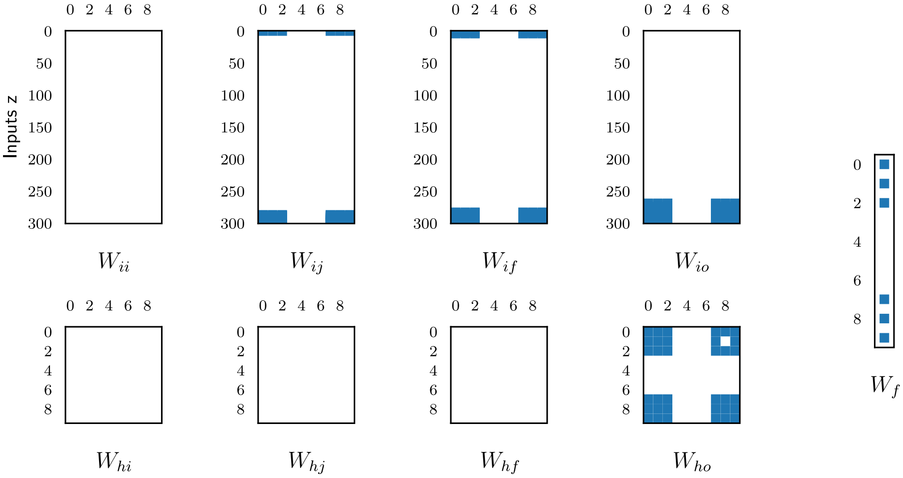


Fig. 5. Sparsity plot of the identified LSTM for Heat Exchanger. (*Blue represents non-pruned connection weights*). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

other benchmarks and removed from the NN, thereby reducing the model complexity.

We also find that DNNs (MLP models) with more hidden layers are necessities to approximate complex systems. For example, the optimal MLP model of the nonlinear cascaded tanks system includes three hidden layers. In contrast, the optimal MLP model of the linear hairdryer system only has one hidden layer. The MLP model with only one hidden layer and 10 hidden neurons is also applied for the cascaded tank system. However, the obtained simulation error is around 0.663, which is worse than the MLP model with three hidden layers (0.257 as in Table 1). It should also be noted that although the number of hidden layers of the MLP models is not reduced in these experiments, the number of hidden neurons is reduced, which provides a more suitable network structure for different systems. For example, as shown in Fig. G.2a, the number of hidden neurons in the MLP model of the Heat Exchanger model is reduced from 50 to 7.

Predictive distributions: The posterior predictive distributions for each model result from the forward propagation of the parameters' posterior uncertainty obtained with the estimation data. Hence, if the validation data holds information that the model does not learn from the estimation data, the posterior predictive distribution could spread a bigger range of predictions (Wilson, 2020).

In some cases, the identified models show an unevenly distributed predictive uncertainty related to nonlinearities or disturbances characteristics of the process and regions where the model can be improved. Fig. 6 shows that the identified model for Cascaded Tanks makes less robust predictions when overflow occurs. The Heat Exchanger shows evenly distributed predictions with uncertainty possibly coming from the ambient temperature disturbance. Furthermore, the model type also affects the predictive distribution. Examples include the LSTM models identified for the Glass Tube Manufacturing Process and Cascaded Tanks. In these benchmarks, the identified MLP model provides more robust predictions than the identified LSTM model.

Free run simulation performance: The free run simulation is a good measure of the model's approximation ability to represent a dynamic process by propagating a model's prediction error while forecasting. In this paper, we select the simulation error as the evaluation metric. It is important to note that, for the studied linear processes, a non-regularised LSTM performs worse when compared to other identification methods. This supports

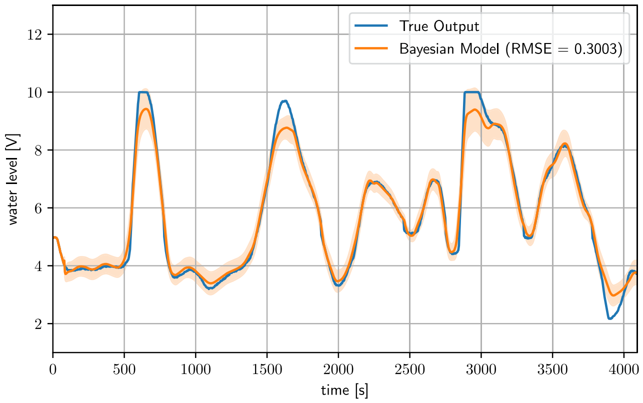


Fig. 6. Posterior predictive $\mu_y \pm 2\sigma_y$ of the identified LSTM on Cascaded Tanks benchmark.

the previous concerns made on using LSTM for the identification of linear systems. The Bayesian MLP model outperforms the Bayesian LSTM model in most presented applications except for the Coupled Electric Drive.

Table 1 displays the mean and standard deviation of the validation simulation errors and the minimum corresponding to the best-chosen model. The minimum is seen to fall close to the range of one standard deviation from the mean. In addition, the variance of validation errors for linear systems is overall less than for nonlinear systems. A possible explanation is that the added complexity in identifying nonlinear processes and the usage of more complex nonlinear structures (LSTM in this case), increases the likelihood of convergence towards saddle points. This is mainly because the Laplace method adopted is a local approximation of the evidence, which is a limitation of the proposed method and justifies running the identification experiment M times.

The Bayesian approach to the identification of each benchmark constitutes an improvement over the conventional MLP and LSTM methods in simulation errors and pushes these methods to perform competitively with other literature (see Table E.1–E.2). Besides, we also make a comparison with the well-known Gaussian process (GP) in machine learning by exploring different kernels (i.e., squared exponential kernel, rational quadratic kernel). However, the GP method cannot perform input regressor selection efficiently, i.e., all regressors flow into the black box model without any priority.

7. Conclusion

In this paper, we combined sparse Bayesian learning and deep learning for SYSID. An iterative procedure for dynamic SYSID has been derived and evaluated with datasets of three linear and two nonlinear dynamic processes. The Bayesian approach in this paper has used the Laplace approximation to approximate the model evidence/marginal likelihood. The structured sparsity regularisation has been implemented on NNs by enforcing group-sparsity inducing priors. An efficient Hessian calculation method for the recurrent layer has been presented by calculating the block-diagonal value of the Hessian. The identified models for the dynamic systems are sparse models that have contributed to input regressor selection and performed competitively with other used SYSID methods in a free run simulation setting. In addition, uncertainties in the inferred predictions and connection weights have been quantified using Monte-Carlo integration methods.

Acknowledgements

We thank Jens Kober and Martijn Wisse from Delft University of Technology for helpful discussions.

References

A. Botev, H. R., & Barber, D. (2017). Practical Gauss-Newton optimisation for deep learning. In *ICML'17*, *Proceedings of the 34th international conference on machine learning - Volume 70* (pp. 557–565). JMLR.org.

Ayala, H. V. H., da Cruz, L. F., Freire, R. Z., et al. 2014. Cascaded free search differential evolution applied to nonlinear system identification based on correlation functions and neural networks. In *Proceedings of the 2014 IEEE symposium on computational intelligence in control and automation (CICA)*, (pp. 1–7).

Beintema, G., Toth, R., & Schoukens, M. (2021). Nonlinear state-space identification using deep encoder networks. In *Learning for dynamics and control* (pp. 241–250). PMLR.

Botev, A. (2020). *The Gauss-Newton matrix for deep learning models and its applications* (Ph.D. thesis), UCL (University College London).

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Castellano, G., & Fanelli, A. M. (2000). Variable selection using neural-network models. *Neurocomputing*, *31*(1–4), 1–13.

Chen, T., Andersen, M. S., Ljung, L., et al. (2014). System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques. *IEEE Transactions on Automatic Control*, *59*(11), 2933–2945.

Chiuso, A., & Pillonetto, G. (2012). A Bayesian approach to sparse dynamic network identification. *Automatica*, *48*(8), 1553–1565.

Dauphin, Y., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. arXiv preprint arXiv:1406.2572.

Delgado, A., Kambhampati, C., & Warwick, K. (1995). Dynamic recurrent neural network for system identification and control. *IEE Proceedings D (Control Theory and Applications)*, *142*(4), 307–314.

Forgione, M., & Piga, D. (2021). Continuous-time system identification with neural networks: model structures and fitting criteria. *European Journal of Control*, *59*, 69–81.

Gal, Y. (2016). *Uncertainty in deep learning* (Ph.D. thesis), University of Cambridge.

Gedon, D., Wahlström, N., Schön, T. B., & Ljung, L. (2021). Deep state space models for nonlinear system identification. *IFAC-PapersOnLine*, *54*(7), 481–486.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Huang, Y., Shao, C., Wu, B., et al. (2019). State-of-the-art review on Bayesian inference in structural system identification and damage assessment. *Adv. Struct. Eng.*, *22*(6), 1329–1351.

Jacobs, W. R., T. Baldacchino, T. D., et al. (2018). Sparse Bayesian nonlinear system identification using variational inference. *IEEE Transactions on Automatic Control*, *63*(12), 4172–4187.

LeCun, Y., et al. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Leshno, M., Lin, V. Y., Pinkus, A., et al. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, [ISSN: 0893-6080] *6*(6), 861–867.

Lindfors, M., & Chen, T. (2020). Regularized LTI system identification in the presence of outliers: A variational EM approach. *Automatica*, [ISSN: 0005-1098] *121*, Article 109152.

Ljung, L. (1999a). *System identification: (2nd ed.): Theory for the user 2nd ed*. USA: Prentice Hall PTR.

Ljung, L. (1999b). System identification. In *Wiley encyclopedia of electrical and electronics engineering* (pp. 1–19). Wiley Online Library.

Ljung, L., Andersson, C., Tiels, K., et al. (2020). Deep learning and system identification. *IFAC-PapersOnLine*, *53*(2), 1175–1181.

M. Brunot, A. J., & Carrillo, F. (2017). Continuous-time nonlinear systems identification with output error method based on derivative-free optimisation. *IFAC-PapersOnLine*, *50*(1), 464–469.

M. Zhou, R. G., & Jin, C. (2021). A local convergence theory for mildly over-parameterized two-layer neural network. arXiv preprint arXiv:2102.02410.

MacKay, D. (1992). Bayesian interpolation. *Neural Computation*, *4*(3), 415–447.

Martens, J., Ba, J., & Johnson, M. (2018). Kronecker-factored curvature approximations for recurrent neural networks. In *International conference on learning representations*.

Martens, J., & Grosse, R. (2015). Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning* (pp. 2408–2417). PMLR.

Martens, J., et al. (2010). Deep learning via Hessian-free optimization. *27*, In *ICML* (pp. 735–742).

Nocedal, J. (1980). Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, *35*(151), 773–782.

Pan, W., Yuan, Y., Gonçalves, J. S., et al. (2016). A sparse Bayesian approach to the identification of nonlinear state-space systems. *IEEE Transactions on Automatic Control*, *61*(1), 182–187.

Schoukens, M., Mattson, P., Wigren, T., et al. (2016). Cascaded tanks benchmark combining soft and hard nonlinearities. In *Workshop on nonlinear system identification benchmarks* (pp. 20–23).

Simon, N., Friedman, J., Hastie, T., et al. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*, *22*(2), 231–245.

Weber, D., & Gühmann, C. (2021). Non-autoregressive vs autoregressive neural networks for system identification. arXiv preprint arXiv:2105.02027.

Wigren, T., & Schoukens, M. (2017). *Coupled electric drives data set and reference models*. Department of Information Technology, Uppsala Universitet.

Wilson, A. G. (2020). The case for Bayesian deep learning. arXiv preprint arXiv:2001.10995.

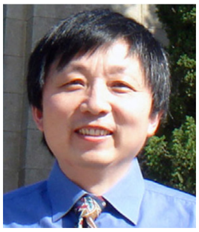
Yuan, Y., Tang, X., Zhou, W., et al. (2019). Data driven discovery of cyber physical systems. *Nature Communications*, *10*(1), 1–9.

Zhou, H., Ibrahim, C., Zheng, W. X., & Pan, W. (2022). Sparse Bayesian deep learning for dynamic system identification. arXiv preprint arXiv:2107.12910.

Zhou, H., Yang, M., Wang, J., et al. (2019). Bayesnas: A Bayesian approach for neural architecture search. In *Proceedings of the 36th international conference on machine learning* (pp. 7603–7613). PMLR.

Hongpeng Zhou received the B.Sc. degree in Detection guidance and control technology from Harbin Institute of Technology, Harbin, China in 2015, the M.Sc. degree in Control Science and Engineering from Harbin Institute of Technology, Harbin, China in 2017, and the Ph.D. degree in Mechanical Engineering from Delft University of Technology, the Netherlands in 2022. He is currently working as a research associate in the University of Manchester, United Kingdom. His main research interests include system identification, deep learning, machine learning, Bayesian optimisation and bioinformatics.

Chahine Ibrahim was born in Brussels, Belgium in 1994. He received a BE degree in Mechanical Engineering from the American University of Beirut in 2017 and an M.Sc. in Systems and Control from the Delft University of Technology in 2020. He is currently working as an R&D engineer at Ampelmann Operations in the Netherlands. His current research focuses on the application of modelling and system identification on offshore mechatronic solutions.



Wei Xing Zheng received the B.Sc. degree in Applied Mathematics in 1982, the M.Sc. degree in Electrical Engineering in 1984, and the Ph.D. degree in Electrical Engineering in 1989, all from Southeast University, Nanjing, China. He is currently a University Distinguished Professor with Western Sydney University, Sydney, Australia. Over the years he has also held various faculty/research/visiting positions at several universities in China, UK, Australia, Germany, USA, etc. He has served as an Associate Editor of Automatica (2011–present), IEEE Transactions on Automatic Control (2004–2007 and 2013–2019), IEEE Transactions on Control of Network Systems (2017–present), and several other flagship journals. He has been an IEEE Distinguished Lecturer of IEEE Control Systems Society. He is a Fellow of IEEE.

Wei Pan received the Ph.D. degree in Bioengineering from Imperial College London in 2017. He is currently an Associate Professor in Machine Learning at the Department of Computer Science, the University of Manchester, UK. Before that, he was an Assistant Professor in Robot Dynamics at the Department of Cognitive Robotics and co-director of Delft SELF AI Lab, Delft University of Technology, Netherlands and a Project Leader at DJI, China. He is the recipient of Dorothy Hodgkin's Postgraduate Awards, Microsoft Research PhD Scholarship and Chinese Government Award for Outstanding Students Abroad. He is on the editorial board of CoRL, ICRA, IROS, IEEE Robotics and Automation Letters. He has a broad interest in robot control using Bayesian machine learning and the principles of dynamic control.