Deep State Space Models for Nonlinear System Identification*

UPPSALA UNIVERSITET

Daniel Gedon[1], Niklas Wahlström[1], Thomas B. Schön[1], Lennart Ljung[2]

May 17, 2021

[1]Deptartment of Information Technology, Uppsala University, Sweden,
[2]Division of Automatic Control, Linköping University, Sweden.

* partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation and the Swedish Research Council, contracts 2016-06079 and 2019-04956.

Outline

Introduction

UPPSALA UNIVERSITET

- System identification: well-established field of automatic control
- Deep learning: successful for high dimensional and nonlinear problems emerging in diverse areas.
- State-Space Models (SSMs) given by:

$$\mathbf{h}_t = f_\theta(\mathbf{h}_{t-1}, \mathbf{u}_t, \mathbf{y}_t).$$
$$\hat{\mathbf{y}}_t = g_\theta(\mathbf{h}_t).$$

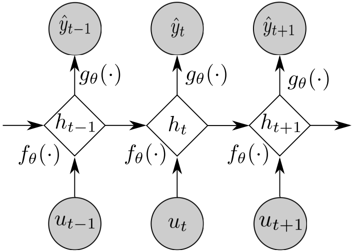
- Deep SSMs:
  - Extension of classic SSM with flexible Neural Networks (NNs); $f_\theta(\cdot)$ and $g_\theta(\cdot)$ as deep mappings.
  - More expressive than feed forward NN for temporal data.
  - Can capture system uncertainty.
- Goal of the paper:
  1. Show usefulness of deep SSMs for nonlinear system identification.
  2. Evaluate six deep SSMs on standard nonlinear system identification benchmarks.

Outline

UPPSALA UNIVERSITET

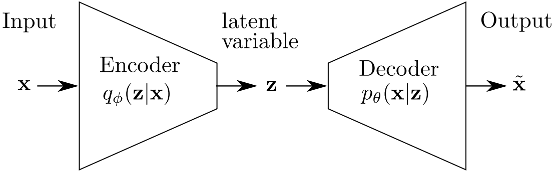
Recurrent Neural Network (RNN)

- Recursive propagation of hidden state.
- Dirac delta function as state transition distribution.



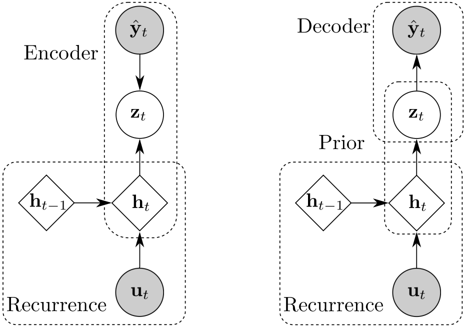
Variational Autoencoder (VAE)

- Decoder: $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}^{dec}, \boldsymbol{\sigma}^{dex}),$
  $[\boldsymbol{\mu}^{\mathrm{dec}}, \boldsymbol{\sigma}^{\mathrm{dec}}] = \mathrm{NN}_\theta^{\mathrm{dec}}(\mathbf{z}).$
- Prior: $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}).$
- Encoder: $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}|\boldsymbol{\mu}^{\mathrm{enc}}, \boldsymbol{\sigma}^{\mathrm{enc}}\right),$
  $[\boldsymbol{\mu}^{\mathrm{enc}}, \boldsymbol{\sigma}^{\mathrm{enc}}] = \mathrm{NN}_\phi^{\mathrm{enc}}(\mathbf{x}).$



UPPSALA UNIVERSITET

- Require a temporal extension of the VAE.
  $\rightarrow$ combine RNN and VAE
- Prior: update with RNN output,
  $p_\theta(\mathbf{z}_t|\mathbf{h}_t) = \mathcal{N}\left(\mathbf{z}_t|\boldsymbol{\mu}_t^{\text{prior}}, \boldsymbol{\sigma}_t^{\text{prior}}\right)$,
  $[\boldsymbol{\mu}_t^{\text{prior}}, \boldsymbol{\sigma}_t^{\text{prior}}] = \text{NN}_\theta^{\text{prior}}(\mathbf{h}_t)$.

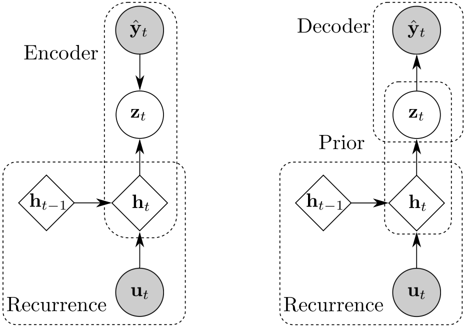


(a) Inference network **(b)** Generative network

Figure 1: Graphical model of VAE-RNN.

We study six different deep SSMs, specifically:

- **VAE-RNN**

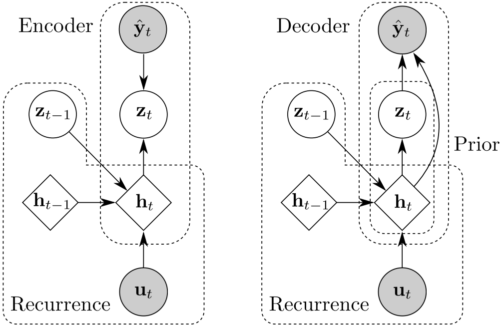


(a) Inference network **(b)** Generative network

Figure 2: Graphical model of VAE-RNN.

We study six different deep SSMs, specifically:

- **VAE-RNN**
- **Variational RNN (VRNN)**: recurrence additionally uses the previous latent variable $\mathbf{z}_{t-1}$.

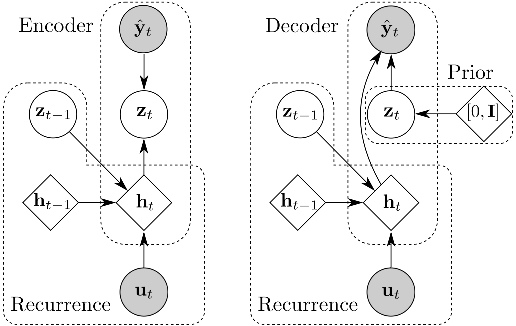


(a) Inference network **(b)** Generative network

Figure 3: Graphical model of VRNN.

We study six different deep SSMs, specifically:

- **VAE-RNN**
- **Variational RNN (VRNN)**: recurrence additionally uses the previous latent variable $\mathbf{z}_{t-1}$.
- **VRNN-I**: VRNN but with static prior.



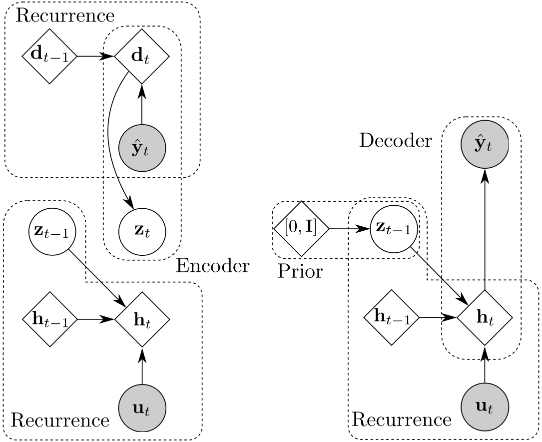
(a) Inference network **(b)** Generative network

Figure 4: Graphical model of VRNN-I.

We study six different deep SSMs, specifically:

- **VAE-RNN**
- **Variational RNN (VRNN)**: recurrence additionally uses the previous latent variable $\mathbf{z}_{t-1}$.
- **VRNN-I**: VRNN but with static prior.
- **Stochastic RNN (STORN)**: Based on VRNN-I. Additional RNN in inference.

Additionally for VRNN and VRNN-I with Gaussian mixture output distribution.



(a) Inference network **(b)** Generative network

Figure 5: Graphical model of STORN.

Outline

VAE parameter learning:

- Maximum likelihood $\mathcal{L}(\theta) = \sum_{i=1}^{N} \log p_\theta(x_i) = \sum_{i=1}^{N} \mathcal{L}_i(\theta)$ intractable.
- Approximate with Evidence Lower Bound (ELBO): $\mathcal{L}_i(\theta) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \widetilde{\mathcal{L}}_i(\theta, \phi)$.
- Rewrite with KL-divergence: $\widetilde{\mathcal{L}}_i(\theta, \phi) = \mathbb{E}_{q_\phi}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})\right)$.
- Maximize $\widetilde{\mathcal{L}}(\theta, \phi) = \sum_{i=1}^{N} \widetilde{\mathcal{L}}_i(\theta, \phi)$.

Deep SSM requires temporal extension of VAE parameter learning:

- ELBO as $\widetilde{\mathcal{L}}(\theta, \phi) = \mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\mathbf{y}_{1:T},\mathbf{z}_{1:T},\mathbf{h}_{1:T}|\mathbf{u}_{1:T},\mathbf{h}_0)}{q_\phi(\mathbf{z}_{1:T},\mathbf{h}_{1:T}|\mathbf{y}_{1:T},\mathbf{u}_{1:T},\mathbf{h}_0)}\right]$.
- Factorize to obtain the following which is maximized

$$\widetilde{\mathcal{L}}(\theta, \phi) = \sum_{t=1}^{T} \mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\mathbf{y}_t|\mathbf{z}_t)p_\theta(\mathbf{z}_t|\mathbf{h}_t)}{q_\phi(\mathbf{z}_t|\mathbf{y}_t,\mathbf{h}_t)}\right] =$$
$$= \sum_{t=1}^{T} \mathbb{E}_{q_\phi}\left[\log p_\theta(\mathbf{y}_t|\mathbf{z}_t)\right] - \mathrm{KL}\left(q_\phi(\mathbf{z}_t|\mathbf{y}_t, \mathbf{h}_t) || p_\theta(\mathbf{z}_t|\mathbf{h}_t)\right),$$

Outline

Toy Problem: Linear Gaussian System

Problem setup:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0.7 & 0.8 \\ 0 & 0.1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} -1 \\ 0.1 \end{bmatrix} \mathbf{u}_k + \mathbf{v}_k,$$

$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k,$$

$$\mathbf{v}_k \sim \mathcal{N}(0, 0.5 \cdot \mathbf{I}), \; \mathbf{w}_k \sim \mathcal{N}(0, 1).$$

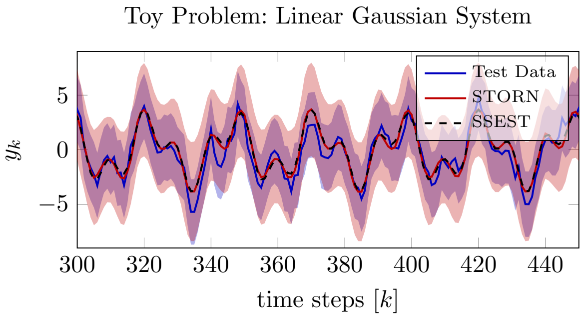


Table 1: Results for linear Gaussian toy problem.

Model	RMSE	NLL
VAE-RNN	1.56	1.95
VRNN-Gauss-I	1.48	1.82
VRNN-Gauss	1.47	1.85
VRNN-GMM-I	1.45	1.80
VRNN-GMM	1.43	1.79
STORN	1.43	1.79
SSEST	1.41	1.78
True lin. model	1.34	-

Narendra-Li Benchmark (1/2)

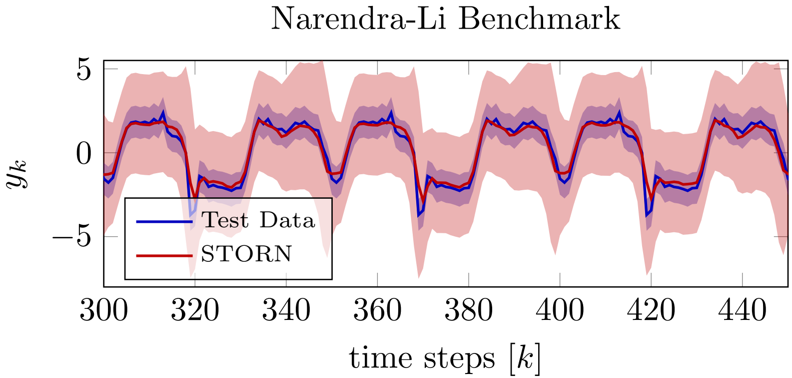
Narendra-Li Benchmark: A highly nonlinear but non-physical, fictional system.

Table 2: Results for the Narendra-Li benchmark.

Model	RMSE	NLL	Samples
VAE-RNN	0.84	1.34	50 000
VRNN-Gauss-I	0.89	1.31	60 000
VRNN-Gauss	0.85	1.28	30 000
VRNN-GMM-I	0.87	1.29	20 000
VRNN-GMM	0.87	1.30	50 000
STORN	0.64	1.20	60 000
Multivariate adaptive regression splines	0.46	-	2 000
Adapt. hinging hyperplanes	0.31	-	2 000
Model-on-demand	0.46	-	50 000
Direct weight optimization	0.43	-	50 000
Basis function expansion	0.06	-	2 000

Narendra-Li Benchmark (2/2)

Results on time evaluation between test data and STORN 1-step ahead prediction.



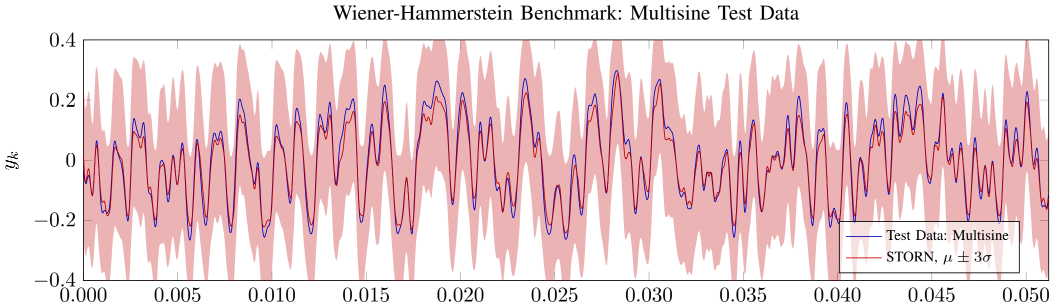
Wiener-Hammerstein Process Noise Benchmark (1/2)

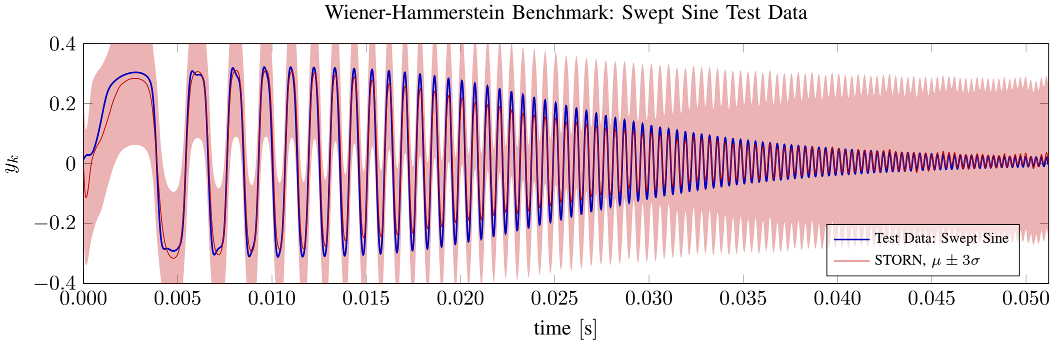
Benchmark: Measured input-output data from an electric circuit.
Nonlinearity between two linear dynamic systems.

Table 3: Results in RMSE for WH benchmark.

Model	swept sine	multisine
VAE-RNN	0.050	0.059
VRNN-Gauss-I	0.076	0.076
VRNN-Gauss	0.082	0.079
VRNN-GMM-I	0.066	0.067
VRNN-GMM	0.076	0.074
STORN	0.034	0.051
NOBF	$\approx$0.2	$<$0.3
NFIR	$<$0.05	$<$0.05
NARX	$<$0.05	$\approx$0.05
PNLSS	0.022	0.038
Best Linear Approx.	-	0.035
ML	-	0.016
SMC	0.014	0.015

Wiener-Hammerstein Process Noise Benchmark (2/2)





Conclusion

- Introduction to deep SSMs as an extension to SSMs
- Six deep SSMs are implemented and applied to nonlinear system identification benchmarks
- Results indicate that the class of deep SSMs is competitive to classic identification methods.

  $\Rightarrow$ toolbox of nonlinear identification methods is extended by a new model class based on deep learning

Thank you!

Reproducible research:

https://github.com/dgedon/DeepSSM_SysID

Contact:

Daniel Gedon, Uppsala University

daniel.gedon@it.uu.se

Supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation