Non-linear State Space Models

Ali Izadi

Table of Contents

1- state space models + RNN

- adding latent variable to RNN
- adding RNN to state space

1) Adding latent Variable to RNN

[1] Bayer, J. and Osendorfer, C., 2014. **Learning stochastic recurrent networks**. *arXiv preprint arXiv:1411.7610*.

[2] Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. and Bengio, Y., 2015. **A recurrent latent variable model for sequential data**. *arXiv preprint arXiv:1506.02216*.

RNN:

$$\mathbf{h}_t = f_\theta\left(\mathbf{x}_t, \mathbf{h}_{t-1}\right)$$

$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid \mathbf{x}_{<t})$$

$$p(\mathbf{x}_t \mid \mathbf{x}_{<t}) = g_\tau(\mathbf{h}_{t-1}),$$

Variational RNN:

$$\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{0,t}, \mathrm{diag}(\boldsymbol{\sigma}^2_{0,t}))$$

$$\text{where } [\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}] = \varphi_\tau^{\text{prior}}(\mathbf{h}_{t-1}),$$

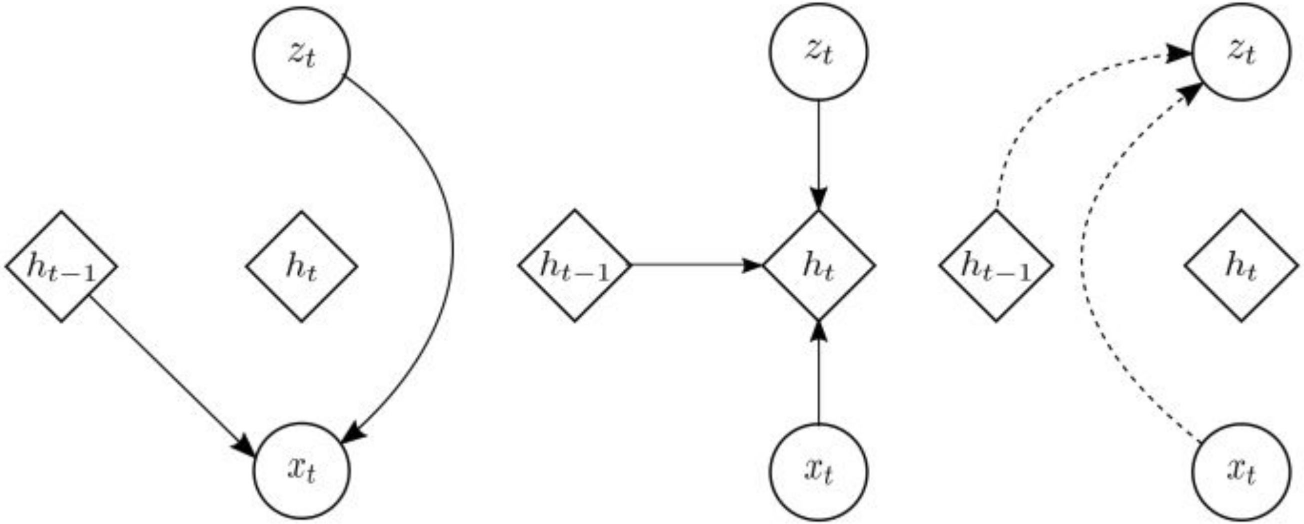
$$\mathbf{x}_t \mid \mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{x,t}, \mathrm{diag}(\boldsymbol{\sigma}^2_{x,t}))$$

$$\text{where } [\boldsymbol{\mu}_{x,t}, \boldsymbol{\sigma}_{x,t}] = \varphi_\tau^{\text{dec}}(\varphi_\tau^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1}),$$

$$\mathbf{h}_t = f_\theta\left(\varphi_\tau^{\mathbf{x}}(\mathbf{x}_t), \varphi_\tau^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1}\right),$$

1) Adding latent Variable to RNN

Variational RNN



(b) Generation (c) Recurrence (d) Inference

2) Adding RNN to state space

[3] Krishnan, R.G., Shalit, U. and Sontag, D., 2015. **Deep kalman filters**. *arXiv preprint arXiv:1511.05121*.

[4] Krishnan, R., Shalit, U. and Sontag, D., 2017, February. **Structured inference networks for nonlinear state space models.** In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 31, No. 1).

Gaussian State Space model:

$$z_t \sim \mathcal{N}(G_\alpha(z_{t-1}, \Delta_t), S_\beta(z_{t-1}, \Delta_t)) \quad \textit{(Transition)} \quad (1)$$

$$x_t \sim \Pi(F_\kappa(z_t)) \quad \textit{(Emission)} \quad (2)$$

linear State Space model:

$$G_\alpha(z_{t-1}) = G_t z_{t-1}, S_\beta = \Sigma_t, F_\kappa = F_t z_t,$$

2) Adding RNN to state space

posterior distribution $p(z_1, \ldots z_T | x_1, \ldots, x_T, u_1, \ldots, u_T)$ becomes intractable to compute.

Stochastic Backpropagation: (VAE)

$$
\begin{aligned}
\log p_\theta(x) &= \log \int_z \frac{q_\phi(z|x)}{q_\phi(z|x)} p_\theta(x|z) p_0(z) \mathrm{d}z \geq \int_z q_\phi(z|x) \log \frac{p_\theta(x|z) p_0(z)}{q_\phi(z|x)} \mathrm{d}z \\
&= \mathop{\mathbb{E}}_{q_\phi(z|x)} [\log p_\theta(x|z)] - \mathrm{KL}(\, q_\phi(z|x) || p_0(z)\,) = \mathcal{L}(x; (\theta, \phi)),
\end{aligned}
$$

2) Adding RNN to state space

[5] Salinas, D., Flunkert, V., Gasthaus, J. and Januschowski, T., 2020. **DeepAR: Probabilistic forecasting with autoregressive recurrent networks**. *International Journal of Forecasting*, *36*(3), pp.1181-1191.

[6] Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y. and Januschowski, T., 2018. **Deep state space models for time series forecasting.** *Advances in neural information processing systems*, *31*, pp.7785-7794.

adding state space to rnn

Datasets	Methods	2-weeks p50Loss	2-weeks p90Loss	3-weeks p50Loss	3-weeks p90Loss	4-weeks p50Loss	4-weeks p90Loss
electricity	auto.arima	0.283	0.109	0.291	0.112	0.30	0.11
	ets	0.121	0.101	0.130	0.110	0.13	0.11
	DeepAR	0.153	0.147	0.147	0.132	0.125	0.080
	DeepState	**0.087**	**0.05**	**0.085**	**0.052**	**0.085**	**0.057**
traffic	auto.arima	0.492	0.280	0.492	0.289	0.501	0.298
	ets	0.621	0.650	0.509	0.529	0.532	0.60
	DeepAR	0.177	0.153	**0.126**	**0.096**	0.219	0.138
	DeepState	**0.168**	**0.117**	0.170	0.113	**0.168**	**0.114**

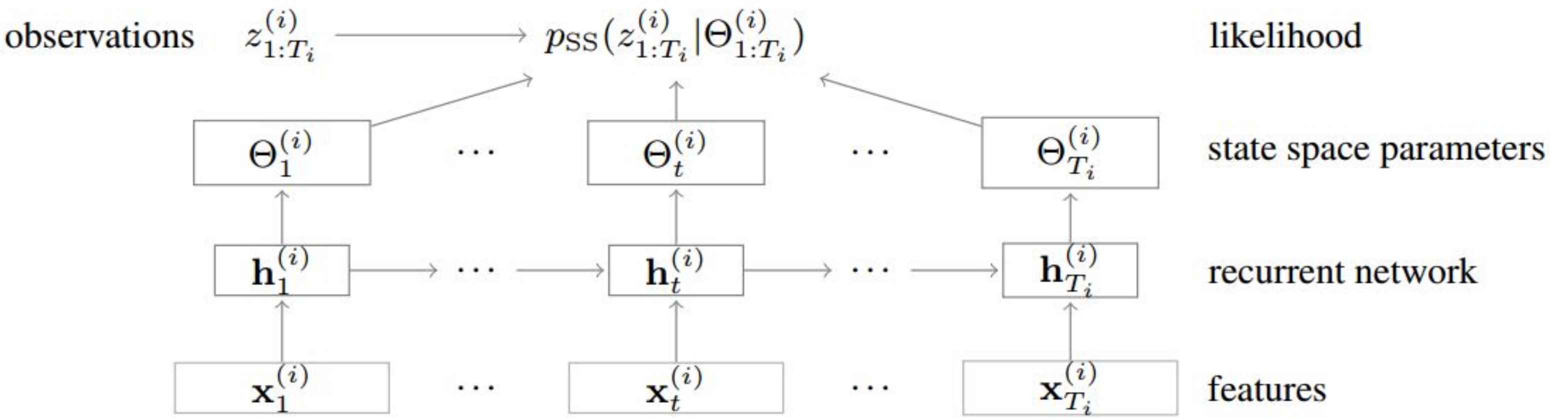
Table 1: Data efficiency. Evaluation on `electricity` and `traffic` datasets with increasing training range. The forecast is evaluated on 7 days.

2) Adding RNN to state space

Parameter learning:

The state space model is fully specified by the parameters:

$$\Theta_t = (\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \boldsymbol{F}_t, \boldsymbol{g}_t, \boldsymbol{a}_t, b_t, \sigma_t)$$

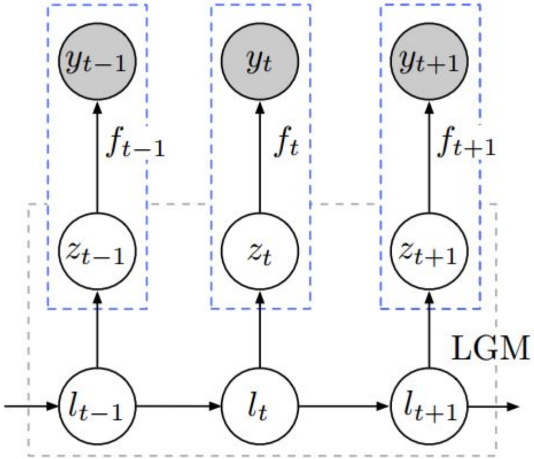


2-Normalizing Flow

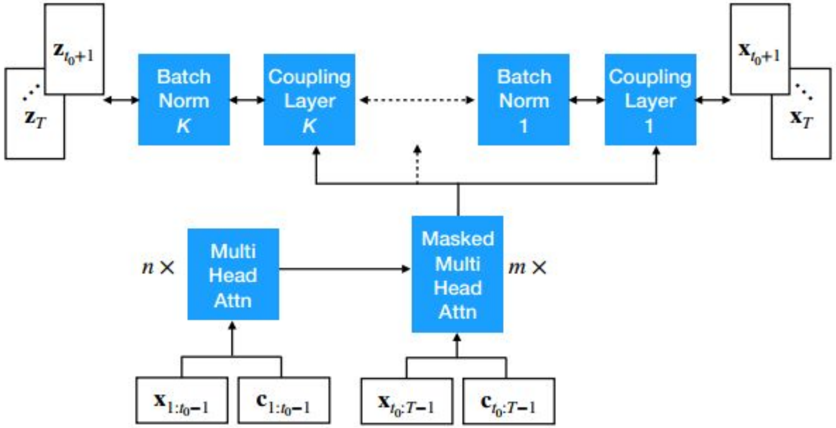
- normalizing kalman filter
- normalizing flow + RNN

[7] de Bézenac, E., Rangapuram, S.S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., Hasson, H., Gallinari, P. and Januschowski, T., 2020. **Normalizing Kalman Filters for Multivariate Time Series Analysis.** *Advances in Neural Information Processing Systems*, *33*.

[8] Rasul, K., Sheikh, A.S., Schuster, I., Bergmann, U. and Vollgraf, R., 2020. **Multi-variate probabilistic time series forecasting via conditioned normalizing flows.** *arXiv preprint arXiv:2002.06103*.



[7]



[8]

3- why combination of state space models + RNN is beneficial?

1) The only source of variability in RNN is conditional output variability.
This is a problem for highly variable, highly structured (high signal-to-noise ratio).

2) Incorporation of structural assumptions + data efficient.

3) Long-term forecast with effective uncertainty estimation.

4- Ideas

- non-linear state space models using normalizing flow

[6] Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y. and Januschowski, T., 2018. **Deep state space models for time series forecasting.** *Advances in neural information processing systems*, *31*, pp.7785-7794.

+ = **Exact Inference**

[7] de Bézenac, E., Rangapuram, S.S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., Hasson, H., Gallinari, P. and Januschowski, T., 2020. **Normalizing Kalman Filters for Multivariate Time Series Analysis.** *Advances in Neural Information Processing Systems*, *33*.

[3] Krishnan, R.G., Shalit, U. and Sontag, D., 2015. **Deep kalman filters**. *arXiv preprint arXiv:1511.05121*.

ELBO

[7] de Bézenac, E., Rangapuram, S.S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., Hasson, H., Gallinari, P. and Januschowski, T., 2020. **Normalizing Kalman Filters for Multivariate Time Series Analysis.** *Advances in Neural Information Processing Systems*, *33*.

[9] Kingma, D.P. and Welling, M., 2013. **Auto-encoding variational bayes.** *arXiv preprint arXiv:1312.6114*.

[10] Rezende, D. and Mohamed, S., 2015, June. **Variational inference with normalizing flows.** In *International Conference on Machine Learning* (pp. 1530-1538). PMLR.

- approximate posterior distributions using normalizing flows

- An interesting question is whether using the idea of the universality of normalizing flows one can transform q(z|x) to be equal (or arbitrarily close) to p(z|x) and thus attain equality in the lower bound. Such a result leads to a consistency result for the learned model, stemming from the consistency of maximum likelihood

Thank you.