Deep latent space models for time-series generation

Linqi Zhou Michael Poli Winnie Xu Stefano Massaroli Stefano Ermon

Sobihan Surendran & Armand Violle

Introduction

Generative models for time-series data are frequently constructed using methods based on ordinary differential equations (ODEs). Existing ODE-based model revealed numerical and computational challenges. In recent work, increasing the dimension of the latent space has proven to be efficient to achieve better performances for Time-Series generation, with latent state space models like S4 [1]. Based on the latter, LS4 [2] introduces a latent space with an increasing dimension, which is said to outperform previous continuous time-generative models. This model is based on an SSM structure for the ODE, and a VAE for the network's architecture. We introduced these concepts, the model and existing experiments in this poster.

State-Space Models (SSMs)

A state-space model maps two single-input signals x and z to a hidden space h and then projecting it into a single-output y, and it can be defined by the following continuous system :

$$\begin{cases} \dot{h}_t &= Ah_t + Bx_t + Ez_t \\ y_t &= Ch_t + Dx_t + Fz_t \end{cases} \tag{1}$$

To fit a time-series framework, we discretize the system (1) using an implicit scheme (Backward Euler) with a time-step $\Delta = t_{k+1} - t_k$, giving us a better formulation. To simplify the notation, we write x_k instead of x_{t_k}, where $x_k = x(k\Delta)$.

$$\begin{cases} h_k &= \overline{A}h_{k-1} + \overline{B}x_k + \overline{E}z_k \\ y_k &= Ch_k + Dx_k + Fz_k \end{cases} \tag{2}$$

with $\overline{A} = (I - \frac{\Delta}{2}A)^{-1}(I + \frac{\Delta}{2}A)$ et $\overline{B} = (I - \frac{\Delta}{2}A)^{-1}\Delta B$ (reps. same definition for $\overline{E}$). From this we established the following recurrence formulas :

$$h_k = \sum_{i=0}^{k} \overline{A}^i \overline{B} x_{k-i} + \sum_{i=0}^{k} \overline{A}^i \overline{E} z_{k-i} \tag{3}$$

For more simplicity, we used the notation $h_k = H_\beta(x, z, h_0, t_k)$, where $\beta = (A, B, E)$ and t_k is the time-step at which we want to evaluate our model.

Variational Auto-Encoders (VAEs) for sequences

VAEs enable to learn latent representations of data inputs, thus able to model all kind of distributions. We're considering the joint distribution $p_\theta(x, z)$ with the *prior* $p(z)$ for the latent distribution, and the *conditional* $p_\theta(x|z)$ used to generate samples from generated latent variables (usually it is a NN). The posterior $p_\theta(z|x)$ being untractable, we approximate it by $q_\phi(z|x)$, enable us to define the ELBO criterion :

$$\text{ELBO} = \mathbb{E}_{q_\phi}\left[\log p_\theta(x|z)\right] - D_{KL}\left(q_\phi(z|x) \,||\, p_\lambda(z)\right) \tag{4}$$

We're considering an observed sequence $x_{\leq T} = (x_0, ..., x_T)$ and a sequence of latent variables $z_{\leq T} = (z_0, ..., z_T)$ sampled from a VAE defined auto-regressively (z_t depends on $z_{<t}$). Both sequences being sampled iid-ly and independently to the other, we can consider the following joint and conditional distributions :

$$\begin{cases} p_\theta(x_{\leq T}, z_{\leq T}) &= \prod_{k=0}^{T} p_\theta(x_k|x_{<k}, z_{\leq k}) p_\lambda(z_k|z_{<k}) \\ q_\phi(z_{\leq T}|x_{\leq T}) &= \prod_{k=0}^{T} q_\phi(z_k|x_{\leq k}) \end{cases} \tag{5}$$

After simplification, we get a new definition of the ELBO criterion, in the case of our discrete framework :

$$\text{ELBO}_{discrete} = \sum_{k=0}^{T} \mathbb{E}_{q_\phi}\left[\log p_\theta(x_k|x_{<k}, z_{\leq k})\right] - D_{KL}\left(q_\phi(z_k|x_{\leq k}) \,||\, p_\lambda(z_k|z_{<k})\right) \tag{6}$$

Latent Space 4 Method

We are using 3 kinds of different variables $x_k \in \mathbb{R}^p$ of the k-th input, $z_k \in \mathbb{R}^q$ the corresponding latent variable and $h_k \in \mathbb{R}^H$ the hidden variable (sub-latent variables). The principle of **LS4 nets** is the following : after mapping input/latent variables into their corresponding hidden variables, we apply N linear layers to map them into a space of dimension $2^N H$ (see left part of figure 1). We thus reach a representation $\tilde{h}_{k+1}^N$ (notation in the figure) belonging to a *very deep* space.

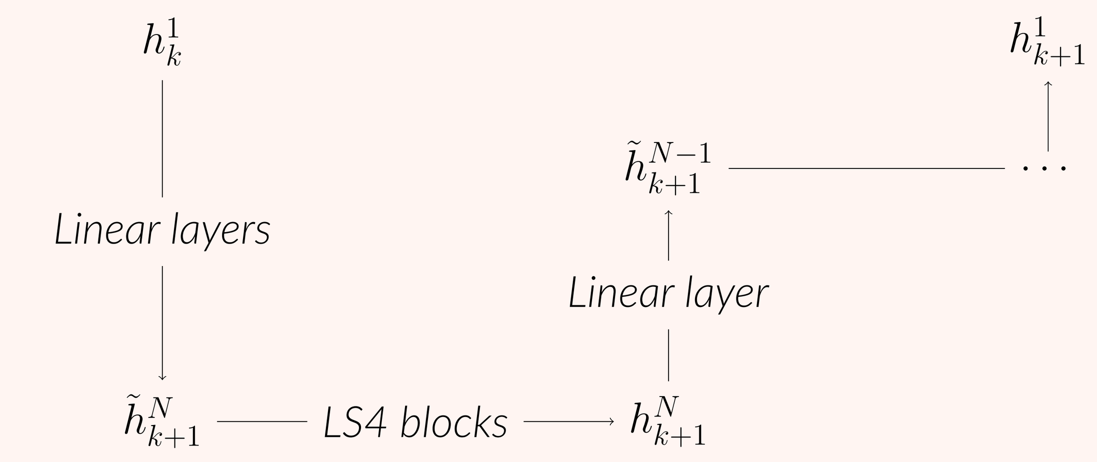

Figure 1. Latent structure of LS4 nets

Then, to generate the next state, we get back to our initial hidden space by applying several smaller networks consisting of B LS4 blocks followed by a linear mapping to reduce the dimension, as described int the figure above. We finally apply a linear layer to map the hidden state h_{k+1}^1 into the desired output. In the next sections, we describe the 3 types of LS4 blocks used in the model.

Prior Block

Like in VAEs, we want to approximate the prior distributions, which are $p_\lambda(z_k|z_{<k})$ in our case. Thanks to the auto-regressive definition of z, the prior network can process using only the latent variables as follows :

$$\begin{aligned} h_{k-1} &= H_{\beta_1}(0, z_{\leq k-1}, 0, k-1) \\ y_{z,k} &= \text{GELU}(C_{y_k} H_{\beta_2}(0, 0, h_{k-1}, k) + F_{y_k} z_{k-1} \\ \textbf{LS4}_{\text{prior}} &= \text{LayerNorm}(G_{y_z} y_{z,k} + b_{y_z}) + z_{k-1} \end{aligned} \tag{7}$$

Using once again the auto-regressive structure, this is equivalent to approximating $\mathcal{N}(\mu_0^z, \sigma_0^z)$, which which is done by reparametrizing. The subsequent z_k are generated auto-regressively.

Generative Block

Here, the goal is to approximate the distributions $p_\theta(x_k|x_{<k}, z_{\leq k})$ of equation (5) that we assume to be gaussian with a fixed covariance, i.e $\mathcal{N}\left(\mu_k^x(x_{<k}, z_{\leq k}), \sigma_x^2\right)$. Unlike in the prior, both latent and input variables are considered in the generative network to produce x_k :

$$\begin{cases} h_{k-1} = H_{\beta_3}(x_{\leq k-1}, z_{\leq k-1}, 0, k-1) \\ \quad h_k = H_{\beta_4}(0, z_{k-1}, h_{k-1}, k) \end{cases}$$

$$\begin{cases} g_{x,k} = \text{GELU}(C_{g_x} h_k + D_{g_x} x_{k-1} + F_{g_x} z_k)\ , & g_{z,k} = \text{GELU}(C_{g_z} h_k + D_{g_z} x_{k-1} + F_{g_z} z_k) \\ \hat{g}_{x,k} = \text{LayerNorm}(G_{g_x} g_{x,k} + b_{g_x}) + x_{k-1})\ , & \hat{g}_{z,k} = \text{LayerNorm}(G_{g_z} g_{z,k} + b_{g_z}) + z_k) \end{cases}$$

$$\textbf{LS4}_{\text{gen}} = (\hat{g}_{x,k}, \hat{g}_{z,k}) \tag{8}$$

Consequently to the nature of our latent variables and the definition of $x_k|x_{<k}, z_{\leq k}$, the goal of the generative block is to approximate the initial distribution for x_0 : $\mathcal{N}\left(\mu_0^x(z_0), \sigma_x^2\right)$. The subsequent x_k are generated auto-regressively.

Inference Block

In this framework, we assume that the approximated posterior $q_\phi(z_k|x_{\leq k})$ is Gaussian of the form $\mathcal{N}\left(\mu_k^z(x_{\leq k}, \sigma^2(x_{\leq k})\right)$. The inference block can be assimilated to the encoder of a VAE, enabling to find the latent representation of the input given only the input. The Inference block is defined as follows :

$$\begin{aligned} h_{k-1} &= H_{\beta_5}(x_{\leq k}, 0, 0, k-1) \\ \hat{y}_{z,k} &= \text{GELU}(C_{\hat{y}_z} h_{k-1} + D_{\hat{y}_z} x_k) \\ \textbf{LS4}_{\text{inf}} &= \text{LayerNorm}(G_{\hat{y}_z} \hat{y}_{z,k} + b_{\hat{y}_z}) + x_k \end{aligned} \tag{9}$$

Given an input $x = (x_0, ..., x_T)$, we are able to compute all z_k in parrallel, allowing faster execution time compared to the prior and generative networks.

Experiments

Data	Metric	RNN-VAE	GP-VAE	ODE^{2}VAE	Latent ODE	TimeGAN	SDEGAN	SaShiMi	**LS4 (Ours)**
FRED-MD	Marginal ↓	0.132	0.152	0.122	0.0416	0.0813	0.0841	0.0482	**0.0221**
	Class. ↑	0.0362	0.0158	0.0282	0.327	0.0294	0.501	0.00119	**0.544**
	Prediction ↓	1.47	2.05	0.567	**0.0132**	0.0575	0.677	0.232	0.0373
NN5 Daily	Marginal ↓	0.137	0.117	0.211	0.107	0.0396	0.0852	0.0199	**0.00671**
	Class. ↑	0.000339	0.00246	0.00102	0.000381	0.00160	0.0852	0.0446	**0.636**
	Prediction ↓	0.967	1.169	1.19	1.04	1.34	1.01	0.849	**0.241**
Temp Rain	Marginal ↓	0.0174	0.183	1.831	**0.0106**	0.498	0.990	0.758	0.0834
	Class. ↑	0.00000212	0.0000123	0.0000319	0.0000419	0.00271	0.0169	0.0000167	**0.976**
	Prediction ↓	159	2.305	1.133	145	1.96	2.46	2.12	**0.521**
Solar Weekly	Marginal ↓	0.0903	0.308	0.153	0.0853	0.0496	0.147	0.173	**0.0459**
	Class. ↑	0.0524	0.000731	0.0998	0.0521	0.6489	0.591	0.00102	**0.683**
	Prediction ↓	1.25	1.47	0.761	0.973	0.237	0.976	0.578	**0.141**

Figure 2. Performances

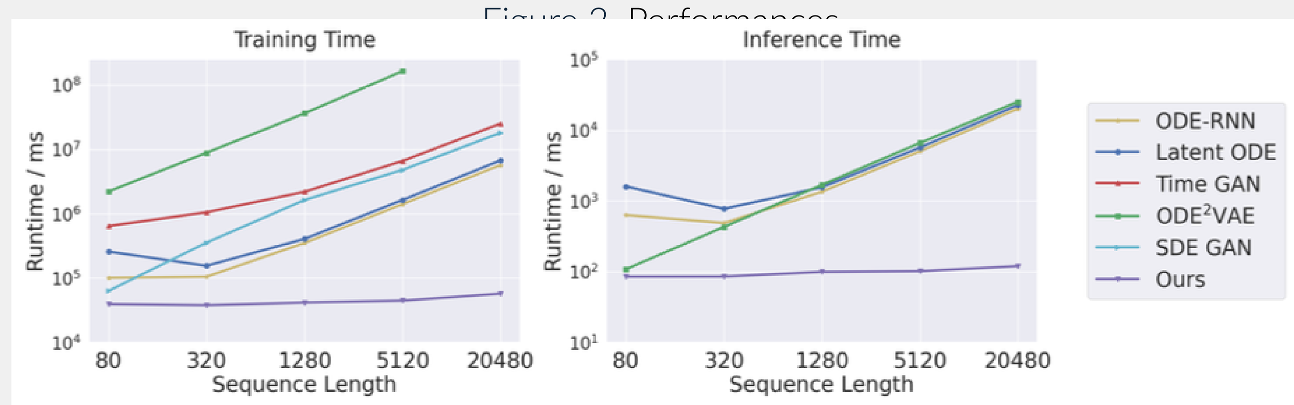

Figure 3. Performances

Conlusion

- **Theoretical fundations** : lack of clarity and precision in the mathematical definitions.
- **Reproductibility** : very difficult because the explanations and the *code* given are vague.

References

[1] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[2] Linqi Zhou, Michael Poli, Winnie Xu, Stefano Massaroli, and Stefano Ermon. Deep latent state space models for time-series generation. *arXiv preprint arXiv:2212.12749*, 2022.