Dynamical Variational Autoencoders

Xavier Alameda-Pineda
RobotLearn Team, Inria, Univ. Grenoble Alpes, CNRS, LJK

joint work with

Laurent Girin[1], **Xiaoyu Bie**[2], **Julien Diard**[3], **Thomas Hueber**[1], **Simon Leglaive**[4], **Radu Horaud**[2]

[1] Univ. Grenoble Alpes, CNRS, Grenoble-INP, GIPSA-lab

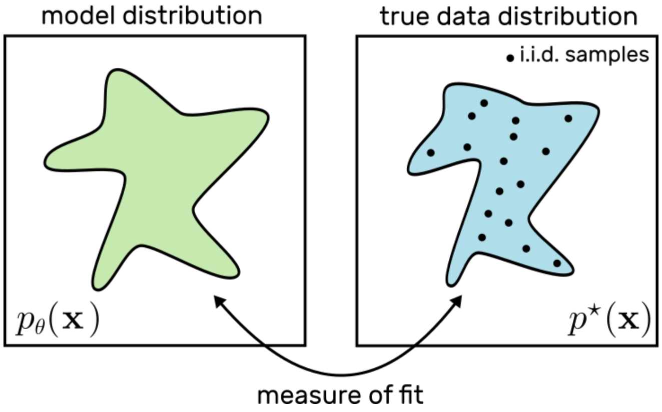
[2] Inria, Univ. Grenoble Alpes, CNRS, LJK

[3] Univ. Grenoble Alpes, CNRS, LPNC

[4] CentraleSupélec, IETR

Introduction

Unsupervised learning



Find θ so that $p(\mathbf{x};\theta)$ is the closest to the $p^\star(\mathbf{x})$: minimize the Kullback-Leibler (KL) divergence.

$$\min_\theta D_{\mathrm{KL}}(p^\star(\mathbf{x}) \parallel p(\mathbf{x};\theta))$$

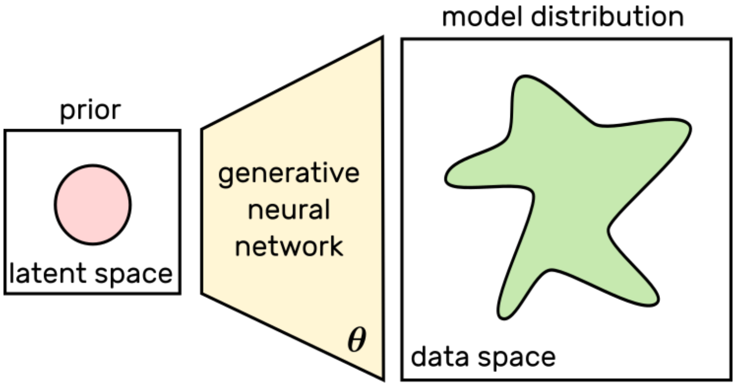
$$= \min_\theta \mathrm{E}_{p^\star(\mathbf{x})}[\ln p^\star(\mathbf{x}) - \ln p($$

$$= \max_\theta \mathrm{E}_{p^\star(\mathbf{x})}[\ln p(\mathbf{x};\theta)]$$

Unknow true data distribution $p^\star(\mathbf{x})$, **but** we have access to a dataset of i.i.d samples $\mathrm{D} = \{\mathbf{x}_i \overset{i.i.d}{\sim} p^\star(\mathbf{x})\}_{i=1}^N$. Replace the intractable expectation by a Monte Carlo estimate:

Latent-variable-based DGMs define the model distribution as a marginal distribution, by introducing a low-dimensional latent random vector:

$$p(\mathbf{x};\theta) = \int p(\mathbf{x}|\mathbf{z};\theta)p(\mathbf{z})d\mathbf{z}.$$



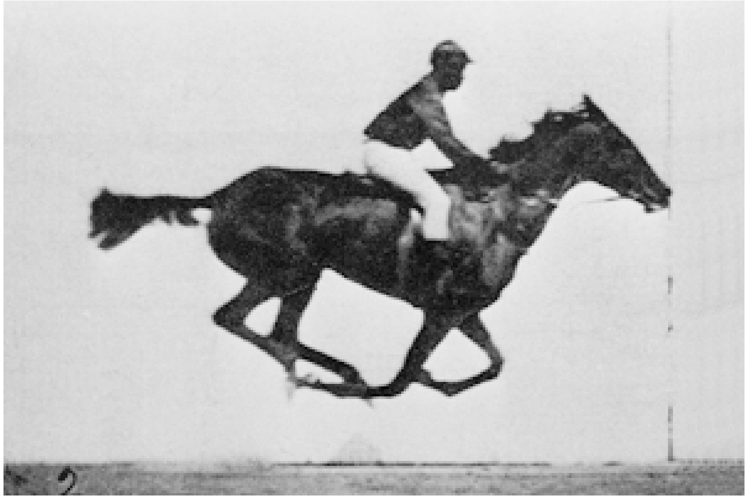
GANs and VAEs are two examples of latent-variable-based DGM, but $p(\mathbf{x}|\mathbf{z};\theta)$ is only defined explicitely (i.e. analytically) for VAEs.

Sequential data

Most of the focus of deep generative modeling has been on static data (e.g. images).

But many data have an inherent sequential/temporal nature (e.g. videos, music, speech, text).

We may be interested in learning the temporal dependencies between the data (observed and latent) at different time frames.



"The horse in motion", pictures made by Eadweard Muybridge in 1887.

Today's presentation

Objective:

Introduce and discuss a class of models called Dynamical Variational Autoencoders (DVAEs) that encompass a large subset of temporal extensions of VAE that have been proposed in the literature.

Outline:

1. Variational autoencoders
2. Dynamical variational autoencoders

This presentation is mostly based on:

L. Girin et al., "Dynamical Variational Autoencoders: A Comprehensive Review", arXiv preprint arXiv:2008.12595, 2020.

S. L. et al., "A Recurrent Variational Autoencoder for Speech Enhancement", IEEE ICASSP, 2020.

Variational autoencoders

- **Generative model**
- Inference model and training
- Application examples

D.P. Kingma and M. Welling, Auto-Encoding Variational Bayes, ICLR 2014.

D.J. Rezende et. al, Stochastic backpropagation and approximate inference in deep generative models, ICML 2014.

Generative model

Let $\mathbf{x} \in \mathrm{R}^D$ and $\mathbf{z} \in \mathrm{R}^K$ be two random vectors (typically $K \ll D$). The generative model is defined by:

$$p(\mathbf{x};\theta) = \int p(\mathbf{x}|\mathbf{z};\theta)p(\mathbf{z})d\mathbf{z}.$$

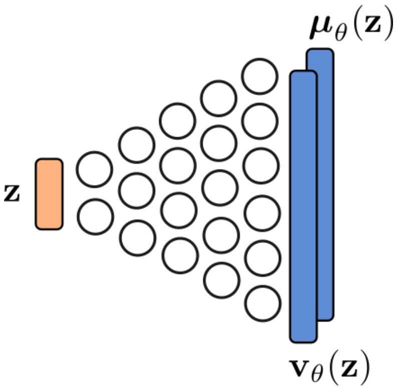
- The prior is a standard Gaussian distribution:

$$p(\mathbf{z}) = \mathrm{N}\left(\mathbf{z};\mathbf{0},\mathbf{I}\right).$$

- The likelihood is parametrized with a generative/decoder neural network, e.g.

$$p(\mathbf{x}|\mathbf{z};\theta) = \mathrm{N}\left(\mathbf{x};\boldsymbol{\mu}_\theta(\mathbf{z}),\mathrm{diag}\left\{\mathbf{v}_\theta(\mathbf{z})\right\}\right),$$

where θ denotes the parameters of the decoder network.


$\boldsymbol{\mu}_\theta(\mathbf{z})$
$\mathbf{z}$
$\mathbf{v}_\theta(\mathbf{z})$

Parameters estimation

- Direct maximum marginal likelihood estimation is intractable due to non-linearities.

- For any distribution $q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})$, we have (Neal and Hinton, 1999; Jordan et al. 1999)

$$\ln p(\mathbf{x};\theta) = \mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta) + D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}) \parallel p(\mathbf{z}|\mathbf{x};\theta)),$$

  where $\mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta)$ is the evidence lower bound (ELBO), defined by

$$\mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta) = \mathrm{E}_{q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})}[\ln p(\mathbf{x},\mathbf{z};\theta) - \ln q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})].$$

R.M. Neal and G.E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants", in M. I. Jordan (Ed.), *Learning in graphical models,* Cambridge, MA: MIT Press, 1999.

M.I. Jordan et al., "An introduction to variational methods for graphical models", Machine learning, 1999.

Parameters estimation

- Direct maximum marginal likelihood estimation is intractable due to non-linearities.

- For any distribution $q(\mathbf{z}|\mathbf{x}; \boldsymbol{\phi})$, we have (Neal and Hinton, 1999; Jordan et al. 1999)

$$\ln p(\mathbf{x}; \theta) = \mathrm{L}(\mathbf{x}; \boldsymbol{\phi}, \theta) + D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x}; \boldsymbol{\phi}) \parallel p(\mathbf{z}|\mathbf{x}; \theta)),$$

where $\mathrm{L}(\mathbf{x}; \boldsymbol{\phi}, \theta)$ is the evidence lower bound (ELBO), defined by

$$\mathrm{L}(\mathbf{x}; \boldsymbol{\phi}, \theta) = \mathrm{E}_{q(\mathbf{z}|\mathbf{x}; \boldsymbol{\phi})}[\ln p(\mathbf{x}, \mathbf{z}; \theta) - \ln q(\mathbf{z}|\mathbf{x}; \boldsymbol{\phi})].$$

Problem #1

$$\max_{\theta} \mathrm{L}(\mathbf{x}; \boldsymbol{\phi}, \theta),$$

where $\mathrm{L}(\mathbf{x}; \boldsymbol{\phi}, \theta) \leq \ln p(\mathbf{x}; \theta)$

R.M. Neal and G.E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants", in M. I. Jordan (Ed.), *Learning in graphical models*, Cambridge, MA: MIT Press, 1999.

M.I. Jordan et al., "An introduction to variational methods for graphical models", Machine learning, 1999.

Parameters estimation

- Direct maximum marginal likelihood estimation is intractable due to non-linearities.

- For any distribution $q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})$, we have (Neal and Hinton, 1999; Jordan et al. 1999)

$$\ln p(\mathbf{x};\theta) = \mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta) + D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}) \parallel p(\mathbf{z}|\mathbf{x};\theta)),$$

where $\mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta)$ is the evidence lower bound (ELBO), defined by

$$\mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta) = \mathrm{E}_{q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})}[\ln p(\mathbf{x},\mathbf{z};\theta) - \ln q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})].$$

Problem #1

$$\max_{\theta} \mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta),$$

Problem #2

$$\max_{\boldsymbol{\phi}} \mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta)$$

where $\mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta) \leq \ln p(\mathbf{x};\theta)$ $\Leftrightarrow \min_{\boldsymbol{\phi}} D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}) \parallel p(\mathbf{z}|\mathbf{x};\theta))$

R.M. Neal and G.E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants", in M. I. Jordan (Ed.), *Learning in graphical models*, Cambridge, MA: MIT Press, 1999.

M.I. Jordan et al., "An introduction to variational methods for graphical models", Machine learning, 1999.

To fully define the objective function, we need to specify the inference model $q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})$.

Variational autoencoders

- Generative model
- **Inference model and training**
- Application examples

D.P. Kingma and M. Welling, Auto-Encoding Variational Bayes, ICLR 2014.

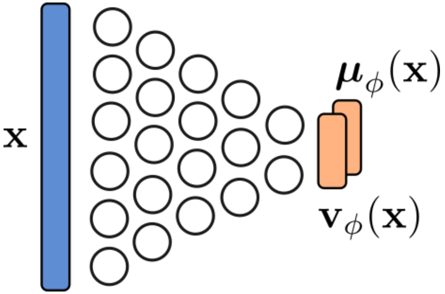
D.J. Rezende et. al, Stochastic backpropagation and approximate inference in deep generative models, ICML 2014.

Inference model

The inference model (approximate posterior) is typically defined by:

$$q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}) = \mathrm{N}\left(\mathbf{z};\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \mathrm{diag}\left\{\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x})\right\}\right),$$

where the mean and variance vectors are provided by the encoder neural network.


$\mathbf{x}$
$\boldsymbol{\mu}_\phi(\mathbf{x})$
$\mathbf{v}_\phi(\mathbf{x})$

ELBO

The ELBO is now fully defined:

$$\begin{aligned}\mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta) &= \mathrm{E}_{q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})}\left[\ln p(\mathbf{x},\mathbf{z};\theta) - \ln q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})\right] \\ &= \underbrace{\mathrm{E}_{q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})}\left[\ln p(\mathbf{x}|\mathbf{z};\theta)\right]}_{\text{reconstruction accuracy}} - \underbrace{D_{\mathrm{KL}}\left(q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}) \parallel p(\mathbf{z})\right)}_{\text{regularization}}.\end{aligned}$$

- prior: $p(\mathbf{z}) = \mathrm{N}\left(\mathbf{z};\mathbf{0},\mathbf{I}\right)$
- likelihood model: $p(\mathbf{x}|\mathbf{z};\theta) = \mathrm{N}\left(\mathbf{x};\boldsymbol{\mu}_\theta(\mathbf{z}),\mathrm{diag}\left\{\mathbf{v}_\theta(\mathbf{z})\right\}\right)$
- inference model: $q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}) = \mathrm{N}\left(\mathbf{z};\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}),\mathrm{diag}\left\{\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x})\right\}\right)$

ELBO

The ELBO is now fully defined:

$$\begin{aligned}\mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta) &= \mathrm{E}_{q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})}\left[\ln p(\mathbf{x},\mathbf{z};\theta) - \ln q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})\right] \\ &= \underbrace{\mathrm{E}_{q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})}\left[\ln p(\mathbf{x}|\mathbf{z};\theta)\right]}_{\text{reconstruction accuracy}} - \underbrace{D_{\mathrm{KL}}\left(q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}) \parallel p(\mathbf{z})\right)}_{\text{regularization}}.\end{aligned}$$

- prior: $p(\mathbf{z}) = \mathrm{N}\left(\mathbf{z};\mathbf{0},\mathbf{I}\right)$
- likelihood model: $p(\mathbf{x}|\mathbf{z};\theta) = \mathrm{N}\left(\mathbf{x};\boldsymbol{\mu}_\theta(\mathbf{z}),\mathrm{diag}\left\{\mathbf{v}_\theta(\mathbf{z})\right\}\right)$
- inference model: $q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}) = \mathrm{N}\left(\mathbf{z};\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}),\mathrm{diag}\left\{\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x})\right\}\right)$

The reconstruction accuracy term is approximated with a Monte Carlo estimate:

$$\mathrm{E}_{q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})}\left[\ln p(\mathbf{x}|\mathbf{z};\theta)\right] \approx \frac{1}{R}\sum_{r=1}^{R}\ln p(\mathbf{x}|\tilde{\mathbf{z}}_r;\theta), \qquad \tilde{\mathbf{z}}_r \sim q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}).$$

ELBO

The ELBO is now fully defined:

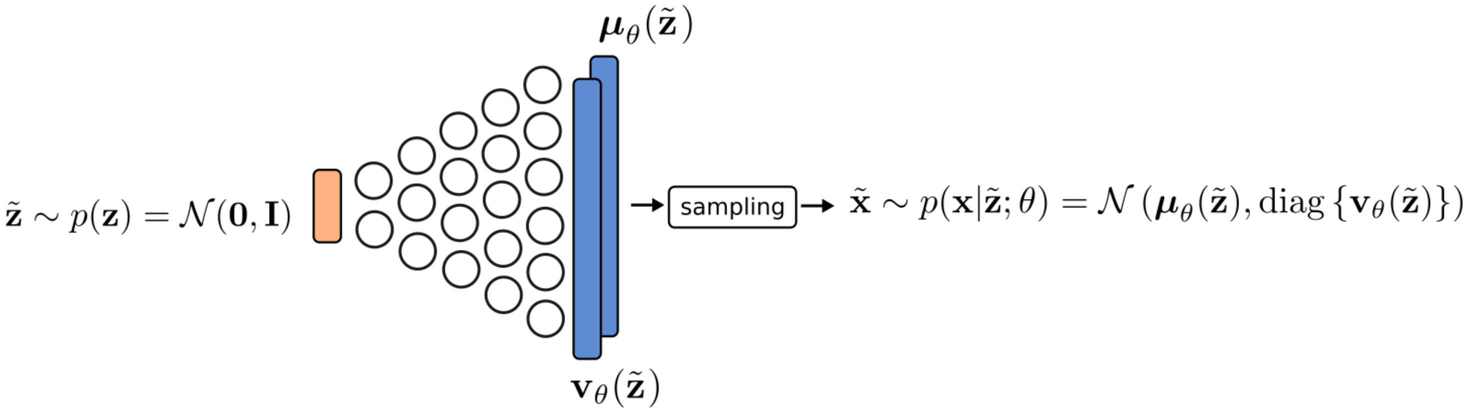
$$\begin{aligned}\mathrm{L}(\mathbf{x};\boldsymbol{\phi},\theta) &= \mathrm{E}_{q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})}\left[\ln p(\mathbf{x},\mathbf{z};\theta) - \ln q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})\right] \\ &= \underbrace{\mathrm{E}_{q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})}\left[\ln p(\mathbf{x}|\mathbf{z};\theta)\right]}_{\text{reconstruction accuracy}} - \underbrace{D_{\mathrm{KL}}\left(q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}) \parallel p(\mathbf{z})\right)}_{\text{regularization}}.\end{aligned}$$

- prior: $p(\mathbf{z}) = \mathrm{N}\left(\mathbf{z};\mathbf{0},\mathbf{I}\right)$
- likelihood model: $p(\mathbf{x}|\mathbf{z};\theta) = \mathrm{N}\left(\mathbf{x};\boldsymbol{\mu}_\theta(\mathbf{z}),\operatorname{diag}\left\{\mathbf{v}_\theta(\mathbf{z})\right\}\right)$
- inference model: $q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}) = \mathrm{N}\left(\mathbf{z};\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}),\operatorname{diag}\left\{\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{x})\right\}\right)$

The reconstruction accuracy term is approximated with a Monte Carlo estimate, using the so-called reparametrization trick:

$$\mathrm{E}_{q(\mathbf{z}|\mathbf{x};\boldsymbol{\phi})}\left[\ln p(\mathbf{x}|\mathbf{z};\theta)\right] \approx \frac{1}{R}\sum_{r=1}^{R}\ln p(\mathbf{x}|\tilde{\mathbf{z}}_r;\theta), \qquad \begin{cases}\boldsymbol{\epsilon}_r & \sim \mathrm{N}\left(\mathbf{0},\mathbf{I}\right) \\ \tilde{\mathbf{z}}_r & = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}) + \operatorname{diag}\{\mathbf{v}_{\boldsymbol{\phi}}(\end{cases}$$

Remarks


$\boldsymbol{\mu}_\theta(\tilde{\mathbf{z}})$
$\tilde{\mathbf{z}} \sim p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$
sampling
$\tilde{\mathbf{x}} \sim p(\mathbf{x}|\tilde{\mathbf{z}};\theta) = \mathcal{N}\left(\boldsymbol{\mu}_\theta(\tilde{\mathbf{z}}), \mathrm{diag}\left\{\mathbf{v}_\theta(\tilde{\mathbf{z}})\right\}\right)$
$\mathbf{v}_\theta(\tilde{\mathbf{z}})$

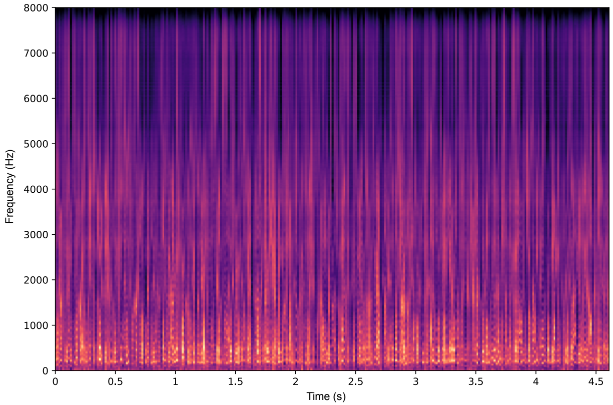
- Note that the encoder was only introduced in order to estimate the parameters of the decoder.
- We do not need the encoder for generating new samples.
- But it is useful if we need to do inference.

Variational autoencoders

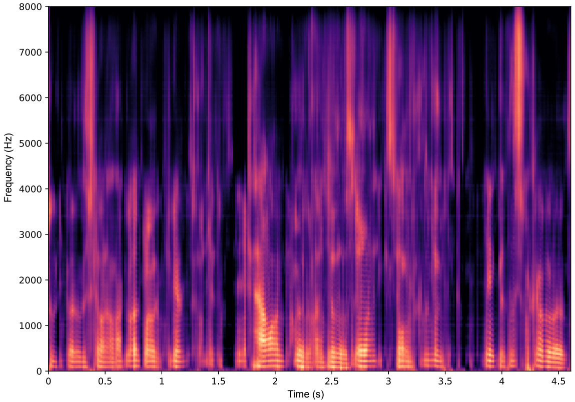
- Generative model
- Inference model and training
- **Application examples**

Generation of a speech signal

VAE



DVAE



The lack of temporal modeling is clearly a problem for VAE. In DVAE we observe: phoneme structure, voiced/unvoiced phonemes, coarticulation and silences.

Phase estimated with Griffin-Lim algorithm.

Dynamical VAEs

- **Generative model**
- Inference model and training
- Applications

Modeling sequential data

We are now interested in modeling sequential data:

- Observed sequence $\mathbf{x}_{1:T} = \{\mathbf{x}_t \in \mathrm{R}^D\}_{t=1}^T$
- Latent sequence $\mathbf{z}_{1:T} = \{\mathbf{z}_t \in \mathrm{R}^L\}_{t=1}^T$

Generative modeling consists in defining the joint distribution with temporal dependencies:

$$p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}),$$

rather than the frame-wise joint distribution (as a vanilla VAE does):

$$p_\theta^{\mathrm{VAE}}(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = \prod_{t=1}^{T} p_\theta(\mathbf{x}_t, \mathbf{z}_t)$$

.

Chain rule

Using the chain rule we can write the joint distribution as a product of conditionals:

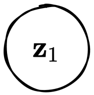
$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1) \prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1})p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})$$

Chain rule

Using the chain rule we can write the joint distribution as a product of conditionals:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1)\prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1})p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})$$

Causal generative process:

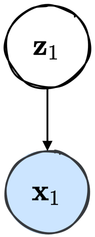


Chain rule

Using the chain rule we can write the joint distribution as a product of conditionals:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1) \prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1})p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})$$

Causal generative process:

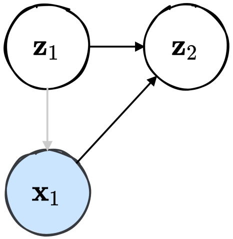


Chain rule

Using the chain rule we can write the joint distribution as a product of conditionals:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1) \prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1})p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})$$

Causal generative process:

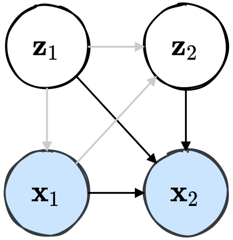


Chain rule

Using the chain rule we can write the joint distribution as a product of conditionals:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1) \prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1})p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})$$

Causal generative process:

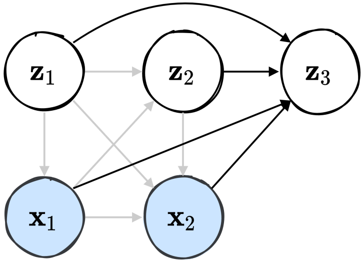


Chain rule

Using the chain rule we can write the joint distribution as a product of conditionals:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1) \prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1})p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})$$

Causal generative process:

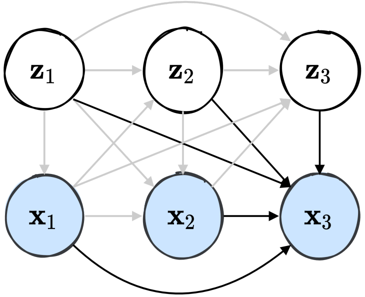


Chain rule

Using the chain rule we can write the joint distribution as a product of conditionals:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1) \prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1})p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})$$

Causal generative process:

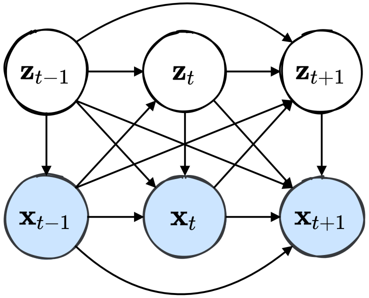


Chain rule

Using the chain rule we can write the joint distribution as a product of conditionals:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1) \prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1})p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})$$

Causal generative process:

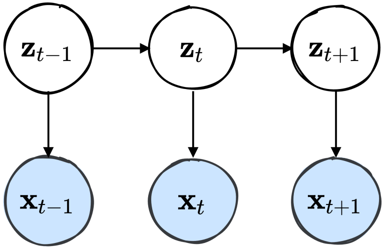


We haven't made any assumption so far.

The state-space model family

Conditional independence assumptions:

- $p(\mathbf{z}_t | \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1}) = p(\mathbf{z}_t | \mathbf{z}_{t-1})$
- $p(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}) = p(\mathbf{x}_t | \mathbf{z}_t)$



The joint distribution simplifies as:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{z}_1) p(\mathbf{x}_1 | \mathbf{z}_1) \prod_{t=2}^{T} p(\mathbf{z}_t | \mathbf{z}_{t-1}) p(\mathbf{x}_t | \mathbf{z}_t).$$

This is the family of state-space models (SSMs) introduced by Kalman in 1960.

- Linear Gaussian SSMs, with a continuous state.
- Hidden Markov models, with a discrete state.

R.E. Kalman, A New Approach to Linear Filtering and Prediction Problems, Transactions of the ASME – Journal of Basic Engineering, 1960.

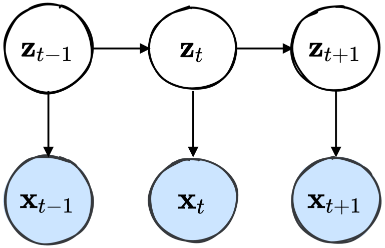
Linear Gaussian SSM

In the linear Gaussian SSM, the two conditional distributions are defined by:

- Transition distribution:

$$p_{\theta_{\mathbf{z}}}(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathrm{N}(\mathbf{z}_t; \boldsymbol{\mu}_{\theta_{\mathbf{z}}}(\mathbf{z}_{t-1}), \boldsymbol{\Sigma}_{\theta_{\mathbf{z}}}(\mathbf{z}_{t-1})),$$

  where $\boldsymbol{\mu}_{\theta_{\mathbf{z}}}(\mathbf{z}_{t-1}) = \mathbf{A}_t\mathbf{z}_{t-1}$ and $\boldsymbol{\Sigma}_{\theta_{\mathbf{z}}}(\mathbf{z}_{t-1}) = \mathbf{Q}_t$.



- Emission distribution:

$$p_{\theta_{\mathbf{x}}}(\mathbf{x}_t|\mathbf{z}_t) = \mathrm{N}(\mathbf{x}_t; \boldsymbol{\mu}_{\theta_{\mathbf{x}}}(\mathbf{z}_t), \boldsymbol{\Sigma}_{\theta_{\mathbf{x}}}(\mathbf{z}_t)),$$

  where $\boldsymbol{\mu}_{\theta_{\mathbf{x}}}(\mathbf{z}_t) = \mathbf{B}_t\mathbf{z}_t$ and $\boldsymbol{\Sigma}_{\theta_{\mathbf{x}}}(\mathbf{z}_t) = \mathbf{R}_t$.

Tractable exact posterior inference using Kalman filter/smoother.

Deep Markov Model

- Transition distribution:

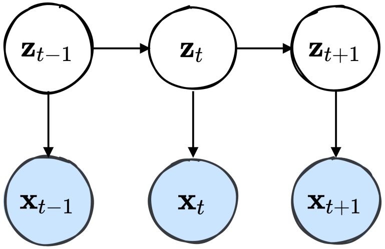
$$p_{\theta_{\mathbf{z}}}(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathrm{N}(\mathbf{z}_t; \boldsymbol{\mu}_{\theta_{\mathbf{z}}}(\mathbf{z}_{t-1}), \boldsymbol{\Sigma}_{\theta_{\mathbf{z}}}(\mathbf{z}_{t-1})),$$

  where $\boldsymbol{\mu}_{\theta_{\mathbf{z}}}$ and $\boldsymbol{\Sigma}_{\theta_{\mathbf{z}}}$ are non-linear functions of $\mathbf{z}_{t-1}$.

- Emission distribution:

$$p_{\theta_{\mathbf{x}}}(\mathbf{x}_t|\mathbf{z}_t) = \mathrm{N}(\mathbf{x}_t; \boldsymbol{\mu}_{\theta_{\mathbf{x}}}(\mathbf{z}_t), \boldsymbol{\Sigma}_{\theta_{\mathbf{x}}}(\mathbf{z}_t)),$$

  where $\boldsymbol{\mu}_{\theta_{\mathbf{x}}}$ and $\boldsymbol{\Sigma}_{\theta_{\mathbf{x}}}$ are non-linear functions of $\mathbf{z}_t$.



The Deep Markov Model corresponds to a VAE with a 1st-order Markov model on the latent state.

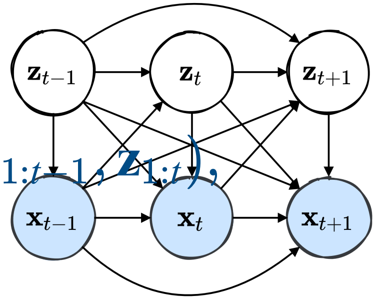
Similar model except that the distributions are now parametrized by a neural network.

R. Krishnan, U. Shalit, D. Sontag, Deep Kalman Filters, NeurIPS Workshops on Advances in Approximate Bayesian Inference & Black Box Inference, 2015.

R. Krishnan, U. Shalit, D. Sontag, Structured Inference Networks for Nonlinear State Space Models, AAAI 2017.

Definition of DVAEs generative model

General umbrella for all (causal) DVAEs:



$$p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = \prod_{t=1}^{T} p_{\theta_\mathbf{z}}(\mathbf{z}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1}) p_{\theta_\mathbf{x}}(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}),$$

where

- $p_{\theta_\mathbf{z}}(\mathbf{z}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1}) = \mathrm{N}(\mathbf{z}_t; \boldsymbol{\mu}_{\theta_\mathbf{z}}(\ldots), \mathrm{diag}\{\mathbf{v}_{\theta_\mathbf{z}}(\ldots)\});$
- $p_{\theta_\mathbf{x}}(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}) = \mathrm{N}(\mathbf{x}_t; \boldsymbol{\mu}_{\theta_\mathbf{x}}(\ldots), \mathrm{diag}\{\mathbf{v}_{\theta_\mathbf{x}}(\ldots)\});$

and $\{\boldsymbol{\mu}_{\theta_\mathbf{z}}, \mathbf{v}_{\theta_\mathbf{z}}\}$, and $\{\boldsymbol{\mu}_{\theta_\mathbf{x}}, \mathbf{v}_{\theta_\mathbf{x}}\}$ are non-linear functions of $\{\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1}\}$ and $\{\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}\}$, respectively.

To simplify notations, we do not distinguish between the cases $t = 1$ and $t > 1$ when writing the chain rule.

RNN parametrization of DVAE generative model

Recall the DVAE generative model:

$$p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = \prod_{t=1}^{T} p_{\theta_\mathbf{z}}(\mathbf{z}_t | \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1}) p_{\theta_\mathbf{x}}(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}).$$

The conditional distributions are parametrized by an RNN, for instance:

- $p_{\theta_\mathbf{z}}(\mathbf{z}_t | \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1}) = \mathrm{N}(\mathbf{z}_t; \boldsymbol{\mu}_{\theta_\mathbf{z}}(\mathbf{h}_t), \mathrm{diag}\{\mathbf{v}_{\theta_\mathbf{z}}(\mathbf{h}_t)\});$
- $p_{\theta_\mathbf{x}}(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}) = \mathrm{N}(\mathbf{x}_t; \boldsymbol{\mu}_{\theta_\mathbf{x}}(\mathbf{z}_t, \mathbf{h}_t), \mathrm{diag}\{\mathbf{v}_{\theta_\mathbf{x}}(\mathbf{z}_t, \mathbf{h}_t)\});$

where $\mathbf{h}_t = \sigma(\mathbf{W}_{xh}\mathbf{x}_{t-1} + \mathbf{W}_{zh}\mathbf{z}_{t-1} + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h)$.

In this example, one single RNN internal state variable $\mathbf{h}_t$ is used to generate both $\mathbf{x}_t$ and $\mathbf{z}_t$. This is totally arbitrary and we could use

Gated extensions of RNNs such as LSTM or GRU networks can also be used.

Dynamical VAEs

- Generative model
- **Inference model and training**
- Applications

Inference in DVAEs

- We are interested in computing the posterior distribution:

$$p_\theta(\mathbf{z}_{1:T} | \mathbf{x}_{1:T}) = \prod_{t=1}^{T} p_\theta(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:T}).$$

  Except for "simple" models such as the linear-Gaussian SSM, this posterior is intractable.

Inference in DVAEs

- We are interested in computing the posterior distribution:

$$p_\theta(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} p_\theta(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:T}).$$

  Except for "simple" models such as the linear-Gaussian SSM, this posterior is intractable.

- As for standard VAEs, we need an inference model:

$$q_\phi(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} q_\phi(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:T}),$$

  where typically we have:

$$q_\phi(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:T}) = \mathrm{N}(\mathbf{z}_t; \widetilde{\boldsymbol{\mu}}_\phi(\ldots), \mathrm{diag}\{\tilde{\mathbf{v}}_\phi(\ldots)\}),$$

  and $\widetilde{\boldsymbol{\mu}}_\phi$, $\tilde{\mathbf{v}}_\phi$ are non-linear functions of $\mathbf{z}_{1:t-1}, \mathbf{x}_{1:T}$.

RNN parametrization of DVAE inference model

One possible parametrization of the conditional posterior of $\mathbf{z}_t$ is given as follows:

$$q_\phi(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:T}) = \mathrm{N}(\mathbf{z}_t; \tilde{\boldsymbol{\mu}}_\phi(\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{g}}_t), \mathrm{diag}\{\tilde{\mathbf{v}}_\phi(\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{g}}_t)\}),$$

where

- $\overrightarrow{\mathbf{h}}_t = \sigma(\overrightarrow{\mathbf{W}}_{xh}\mathbf{x}_{t-1} + \overrightarrow{\mathbf{W}}_{zh}\mathbf{z}_{t-1} + \overrightarrow{\mathbf{W}}_{hh}\overrightarrow{\mathbf{h}}_{t-1} + \overrightarrow{\mathbf{b}}_h)$

  encodes the causal dependencies (we may or may not use the same RNN as for the generative model),

- $\overleftarrow{\mathbf{g}}_t = \sigma(\mathbf{W}_{xg}\mathbf{x}_t + \overleftarrow{\mathbf{W}}_{gg}\overleftarrow{\mathbf{g}}_{t+1} + \overleftarrow{\mathbf{b}}_h)$

  encodes the non-causal dependencies.

  We can compute the ELBO as in VAE. However, both the reconstruction and regularization terms involve an intractable

Different models in the literature belong to the DVAE family (different conditional assumptions):

STORN	J. Bayer and C. Osendorfer, Learning Stochastic Recurrent Networks, arXiv preprint arXiv:1411.7610, 2014
VRNN	J. Chung et al., A recurrent latent variable model for sequential data, NeurIPS, 2015
SRNN*	M. Fraccaro et al., "Sequential neural models with stochastic layers", NeurIPS 2016
DMM*	R. Krishnan et al., Structured Inference Networks for Nonlinear State Space Models, AAAI, 2017
DSAE	Y. Li and S Mandt, Disentangled sequential autoencoder, ICML, 2018
(N)C-RVAE*	S. L. et al., A recurrent variational autoencoder for speech enhancement, IEEE ICASSP, 2020
HIT-DVAE	X. B. et al., HiT-DVAE: Human Motion Generation via Hierarchical Transformer Dynamical VAE, arXiv, 2022

In (Girin et al., 2020), we:

- review and discuss several DVAE models with unified notations,

L. Girin, S. L., X. Bie, J. Diard, T. Hueber, X. Alameda-Pineda, "Dynamical Variational Autoencoders: A Comprehensive Review", arXiv preprint arXiv:2008.12595, 2020.

Dynamical VAEs

- Generative model
- Inference model and training
- **Applications**

Applications & Remarks

So far we have successfully applied DVAEs to:

- Speech (power spectrogram) analysis-resynthesis
- Unsupervised speech enhancement (noise distribution learned at test time)
- Human motion modeling and prediction (HIT-DVAE)
- Single and multiple (bounding-box) object tracking

Open research questions:

- How do interpret $\mathbf{z}$?
- How to ensure $\mathbf{z}$ carries information (specially in auto-regressive models)?
- Can DVAEs handle multiple modalities? How to design the latent spaces?

Thank you

Online material available at:

https://team.inria.fr/robotlearn/dvae/

https://dynamicalvae.github.io/

https://sleglaive.github.io/