

VARIATIONAL AUTOENCODERS

Vincent Barra
 LIMOS, UMR 6158 CNRS, Université Clermont Auvergne



LIMOS
 LABORATOIRE D'INFORMATIQUE,
 DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES



GENERATIVE MODELS

Objective

Learning a model that represents the distribution of data, with input training samples

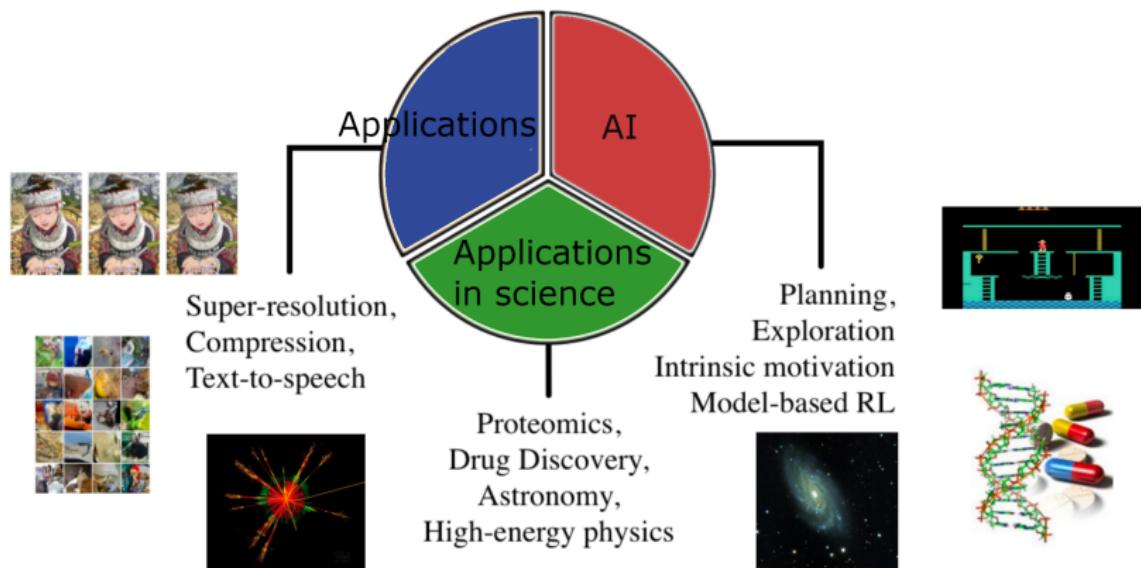
$$\mathcal{P}_{model}(x) \sim \mathcal{P}_{data}(x)$$

0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9 9



7	5	3	2
6	0	4	2
5)	1	3
3	1	6	0

GENERATIVE MODELS

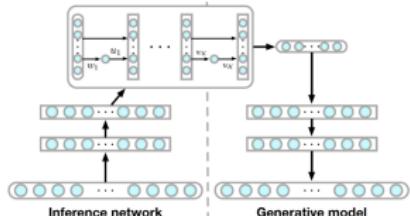


Source: G Louppe

GENERATIVE MODELS

A lot of methods:

- ▶ Boltzmann generators
- ▶ Normalizing flows
- ▶ ...



Rezende and Mohamed 2015.

Latent variable models

- ▶ Variational Autoencoders (today)
- ▶ Generative Adversarial Networks (next lecture)

LATENT MODELS

A latent variable model relates a set of observable variables $\mathbf{x} \in X$ to a set of latent variables $\mathbf{h} \in H$

$$\mathbf{p}(\mathbf{x}, \mathbf{h}) = \mathbf{p}(\mathbf{x}|\mathbf{h})\mathbf{p}(\mathbf{h})$$

if \mathbf{h} are causal factors for $\mathbf{x} \Rightarrow$ sampling from $\mathbf{p}(\mathbf{x}|\mathbf{h})$ = generative process from H to X .

Inference

Inference: given $\mathbf{p}(\mathbf{x}, \mathbf{h})$, compute

$$\mathbf{p}(\mathbf{h}|\mathbf{x}) = \frac{\mathbf{p}(\mathbf{x}|\mathbf{h})\mathbf{p}(\mathbf{h})}{\mathbf{p}(\mathbf{x})}$$

Intractable

LATENT MODELS

Variational Inference

- ▶ $\mathbf{p}_\phi(\mathbf{h}|\mathbf{x})$: family of distributions approximating $\mathbf{p}(\mathbf{h}|\mathbf{x})$
- ▶ ϕ is optimized to minimize the “distance” between both distributions.
- ▶ Among all similarity measures: Kullback Leibler divergence

$$\begin{aligned} KL(\mathbf{p}_\phi(\mathbf{h}|\mathbf{x}) || \mathbf{p}(\mathbf{h}|\mathbf{x})) &= \mathbb{E}_{\mathbf{p}_\phi(\mathbf{h}|\mathbf{x})} \left[\log \frac{\mathbf{p}_\phi(\mathbf{h}|\mathbf{x})}{\mathbf{p}(\mathbf{h}|\mathbf{x})} \right] \\ &= \mathbb{E}_{\mathbf{p}_\phi(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}_\phi(\mathbf{h}|\mathbf{x})) - \log(\mathbf{p}(\mathbf{x}, \mathbf{h}))] + \textcolor{red}{\log(\mathbf{p}(\mathbf{x}))} \end{aligned}$$

Still intractable



LABORATOIRE D'INFORMATIQUE,
DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES



LATENT MODELS

But...

$$\begin{aligned}\min_{\phi} KL(\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x}) || \mathbf{p}(\mathbf{h}|\mathbf{x})) &= \min_{\phi} \log(\mathbf{p}(\mathbf{x})) - \mathbb{E}_{\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}(\mathbf{x}, \mathbf{h})) - \log(\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x}))] \\ &= \max_{\phi} \underbrace{\mathbb{E}_{\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}(\mathbf{x}, \mathbf{h})) - \log(\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x}))]}_{ELBO(\mathbf{x}, \phi)}\end{aligned}$$

LATENT MODELS

But...

$$\begin{aligned}\min_{\phi} KL(\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x}) || \mathbf{p}(\mathbf{h}|\mathbf{x})) &= \min_{\phi} \log(\mathbf{p}(\mathbf{x})) - \mathbb{E}_{\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}(\mathbf{x}, \mathbf{h})) - \log(\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x}))] \\ &= \max_{\phi} \underbrace{\mathbb{E}_{\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}(\mathbf{x}, \mathbf{h})) - \log(\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x}))]}_{ELBO(\mathbf{x}, \phi)}\end{aligned}$$

ELBO

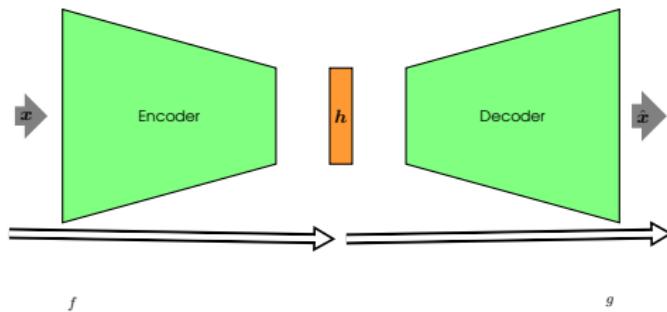
ELBO: Evidence Lower Bound Objective

$$\begin{aligned}ELBO(\mathbf{x}, \phi) &= \mathbb{E}_{\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}(\mathbf{x}, \mathbf{h})) - \log(\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x}))] \\ &= \mathbb{E}_{\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}(\mathbf{x}|\mathbf{h}))\mathbf{p}(\mathbf{h}) - \log(\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x}))] \\ &= \mathbb{E}_{\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x})} [\textcolor{blue}{\log(\mathbf{p}(\mathbf{x}|\mathbf{h}))}] - \textcolor{orange}{KL(\mathbf{p}_{\phi}(\mathbf{h}|\mathbf{x}) || \mathbf{p}(\mathbf{h}))}\end{aligned}$$

Maximizing $ELBO(\mathbf{x}, \phi)$:

- ▶ The first term encourages distributions to be centered on configurations of latent variables \mathbf{h} explaining the observed data
- ▶ The second term enforces distributions to be close to the prior.

AUTOENCODERS

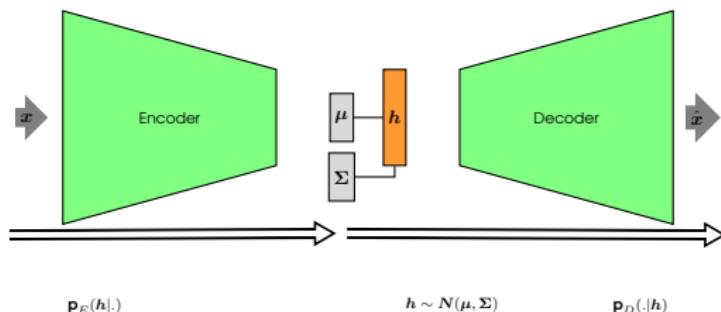


Key ideas

- ▶ A neural network trained using unsupervised learning
- ▶ Trained to copy its input to its output
- ▶ Learns an embedding h

$$\hat{x} = g[f(\mathbf{x})] \quad h = f(\mathbf{x})$$

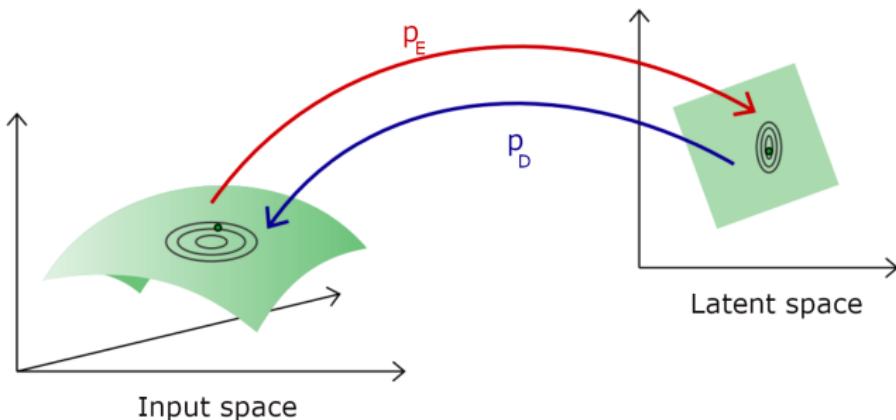
VARIATIONAL AUTOENCODERS



Probabilistic twist on AE

- ▶ $f \implies \mathbf{p}_E(\mathbf{h}|\mathbf{x})$
- ▶ $g \implies \mathbf{p}_D(\mathbf{x}|\mathbf{h})$
- ▶ Sample from μ and Σ to compute latent samples

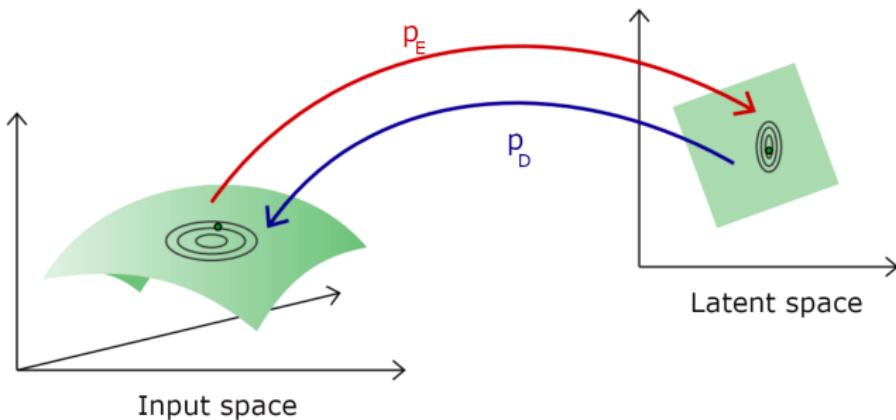
VARIATIONAL AUTOENCODERS



Probabilistic twist on AE

- ▶ $f \implies p_E(\mathbf{h}|\mathbf{x})$
- ▶ $g \implies p_D(\mathbf{x}|\mathbf{h})$
- ▶ Sample from μ and Σ to compute latent samples

VARIATIONAL AUTOENCODERS



Probabilistic twist on AE

- ▶ $\mathbf{p}_E(\mathbf{h}|\mathbf{x}) = \mathbf{p}_\phi(\mathbf{h}|\mathbf{x})$
- ▶ $\mathbf{p}_D(\mathbf{x}|\mathbf{h}) \implies \mathbf{p}_\theta(\mathbf{x}|\mathbf{h})$
- ▶ $\mathbf{p}(\mathbf{h})$: Fixed prior on the latent distribution

Loss:

$$\min_{\phi, \theta} (-ELBO(\mathbf{x}, \phi)) = -\mathbb{E}_{\mathbf{p}_\phi(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}_\theta(\mathbf{x}|\mathbf{h}))] + KL(\mathbf{p}_\phi(\mathbf{h}|\mathbf{x}) || \mathbf{p}(\mathbf{h}))$$

ELBO

Loss

$$\min_{\phi, \theta} (-ELBO(\mathbf{x}, \phi)) = -\mathbb{E}_{\mathbf{p}_\phi(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}_\theta(\mathbf{x}|\mathbf{h}))] + KL(\mathbf{p}_\phi(\mathbf{h}|\mathbf{x}) || \mathbf{p}(\mathbf{h}))$$

- ▶ Given θ , optimize ϕ so that latent variable distribution explains the observed data, while remaining close to the prior
- ▶ Given ϕ , optimize θ so that observed data is well explained by the latent variables.

VARIATIONAL AUTOENCODERS

Encoder/decoder

Both encoder and decoder map to a Gaussian with diagonal covariance.

- ▶ $\mathbf{p}_\phi(\mathbf{h}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}^f(\mathbf{X}), \text{diag}(\boldsymbol{\sigma}^f(\mathbf{X})))$: Inference network (encoder)
- ▶ $\mathbf{p}_\theta(\mathbf{x}|\mathbf{h}) = \mathcal{N}(\boldsymbol{\mu}^g(\mathbf{h}), \text{diag}(\boldsymbol{\sigma}^g(\mathbf{h})))$: Generative network (decoder)

VARIATIONAL AUTOENCODERS

Encoder/decoder

Both encoder and decoder map to a Gaussian with diagonal covariance.

- ▶ $\mathbf{p}_\phi(\mathbf{h}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}^f(\mathbf{X}), \text{diag}(\boldsymbol{\sigma}^f(\mathbf{X})))$: Inference network (encoder)
- ▶ $\mathbf{p}_\theta(\mathbf{x}|\mathbf{h}) = \mathcal{N}(\boldsymbol{\mu}^g(\mathbf{h}), \text{diag}(\boldsymbol{\sigma}^g(\mathbf{h})))$: Generative network (decoder)

Prior $\mathbf{p}(\mathbf{h})$

- ▶ Impose a distribution for the random latent variable.
- ▶ Train the decoder so that its output matches the training data.
- ▶ Common choice: $\mathbf{h} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}^2)$

VARIATIONAL AUTOENCODERS

Let $\mathbf{x}_1 \dots \mathbf{x}_n$ be the training inputs.

$$KL(\mathbf{p}_\phi(\mathbf{h}|\mathbf{x}) || \mathbf{p}(\mathbf{h})) = \frac{1}{n} \left[-\frac{1}{2} \sum_{i=1}^d \left(1 + \log \sigma_i^f(\mathbf{x}) - (\mu_i^f(\mathbf{x}))^2 - \sigma_i^f(\mathbf{x}) \right) \right]$$

VARIATIONAL AUTOENCODERS

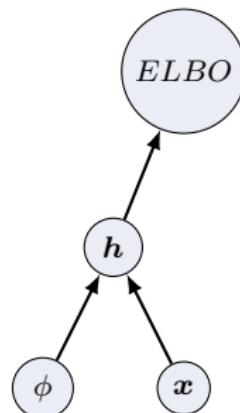
Let $\mathbf{x}_1 \cdots \mathbf{x}_n$ be the training inputs.

$$KL(\mathbf{p}_\phi(\mathbf{h}|\mathbf{x}) || \mathbf{p}(\mathbf{h})) = \frac{1}{n} \left[-\frac{1}{2} \sum_{i=1}^d \left(1 + \log \sigma_i^f(\mathbf{x}) - (\mu_i^f(\mathbf{x}))^2 - \sigma_i^f(\mathbf{x}) \right) \right]$$

$$\min_{\phi, \theta} (-ELBO(\mathbf{x}, \phi)) = -\mathbb{E}_{\mathbf{p}_\phi(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}_\theta(\mathbf{x}|\mathbf{h}))] + KL(\mathbf{p}_\phi(\mathbf{h}|\mathbf{x}) || \mathbf{p}(\mathbf{h}))$$

Optimization

- ▶ Gradients w.r.t $\theta \implies$ easy to obtain
- ▶ Gradients w.r.t $\phi \implies$ more difficult.



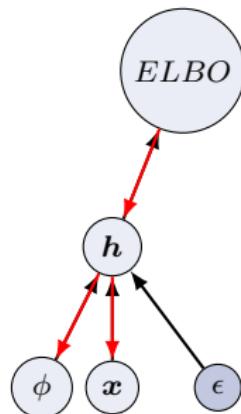
Cannot backpropagate through \mathbf{h} to compute $\nabla_\phi(ELBO)$

REPARAMETERIZATION TRICK

Trick

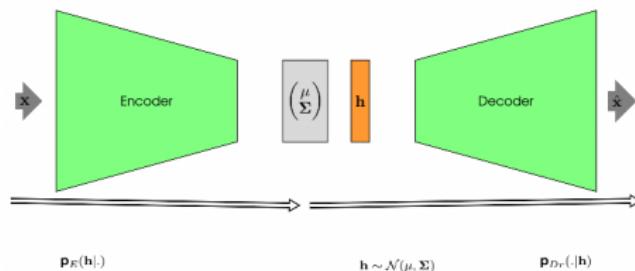
Expressing \mathbf{h} as some differentiable and invertible transformation of another random variable ϵ given \mathbf{x} and ϕ .

$$\mathbf{h} = \mu(\mathbf{x}, \phi) + \sigma(\mathbf{x}, \phi) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$



Gradient can flow out of any random variable \Rightarrow backpropagation is possible.

SUMMARY

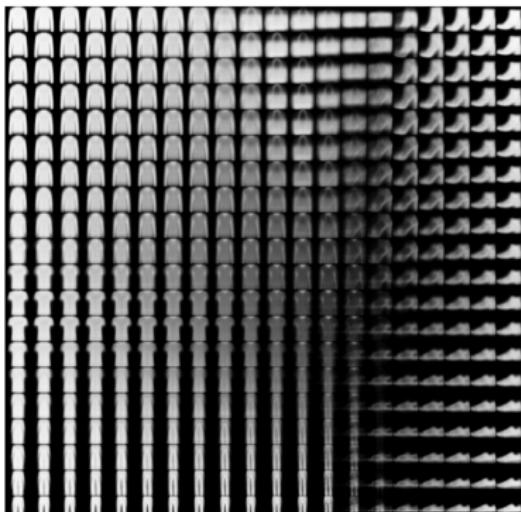
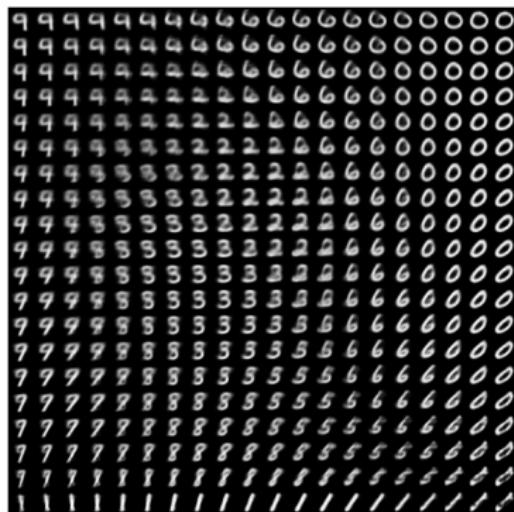


VAE

- 1 Define Encoder and Decoder
- 2 Define the latent space distribution using the reparametrization trick
- 3 Define the ELBO loss
- 4 Train and play with VAE !

EXPLORATION

- ▶ Changing one single variable h_i and keep all other h_j fixed.
- ▶ Dimensions of \mathbf{h} encode different interpretable latent features.



DISENTANGLEMENT



LIMOS
LOGIC MODEL SYSTEM

Kingma et al, 2014

DISENTANGLEMENT



Homogeneous skin color, pose

VS



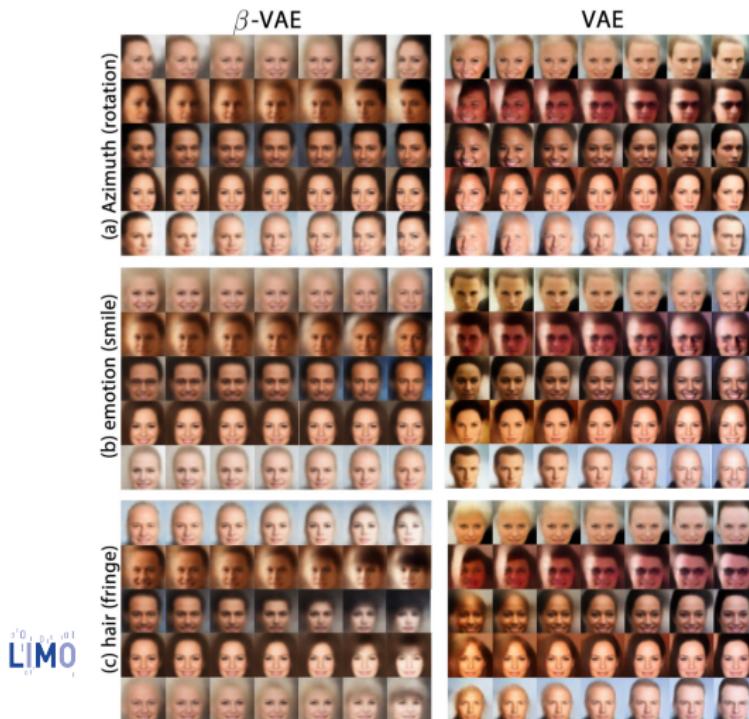
Diverse skin color, pose, illumination

Amini and Soleimany, 2019

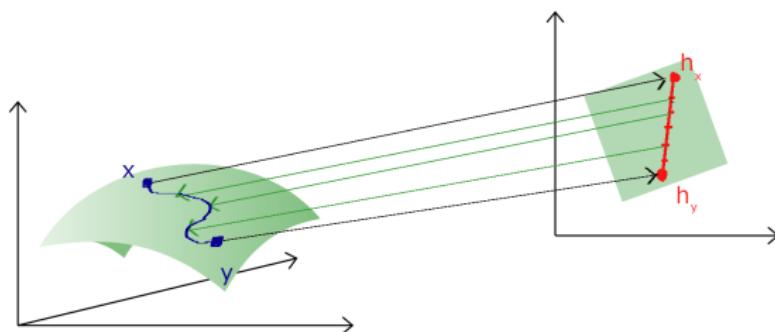
DISENTANGLEMENT

$\beta - VAE$: enforces disentanglement

$$\min_{\phi, \theta} (-ELBO(\mathbf{x}, \phi)) = -\mathbb{E}_{\mathbf{p}_\phi(\mathbf{h}|\mathbf{x})} [\log(\mathbf{p}_\theta(\mathbf{x}|\mathbf{h}))] + \beta KL(\mathbf{p}_\phi(\mathbf{h}|\mathbf{x}) || \mathbf{p}(\mathbf{h}))$$



EXPLORATION



Interpolation in the image space



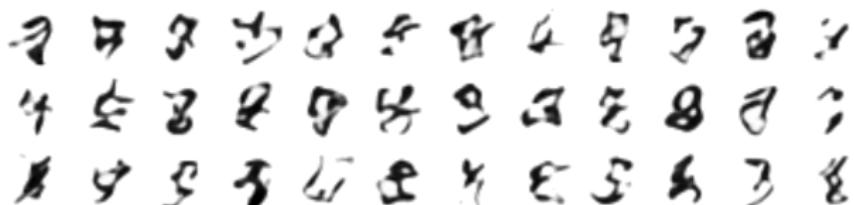
LABORATOIRE D'INFORMATIQUE,
DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES



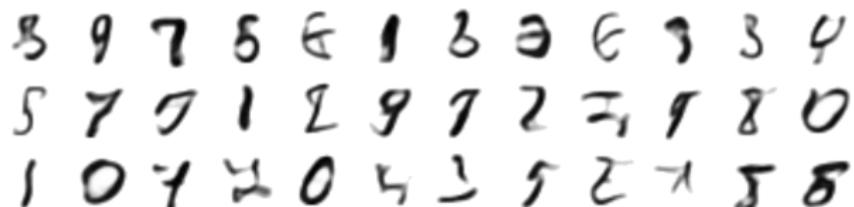
Interpolation in the latent space

SAMPLING

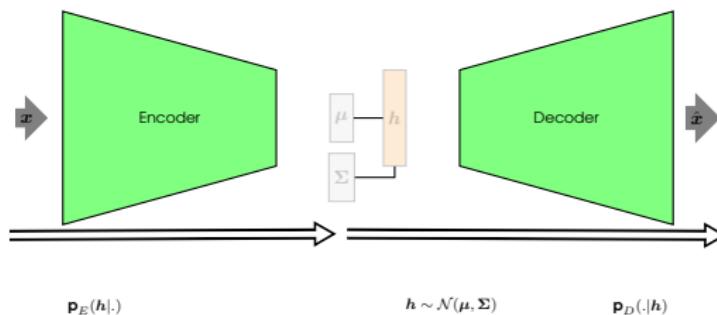
AE (d=32)



VAE (d=32)



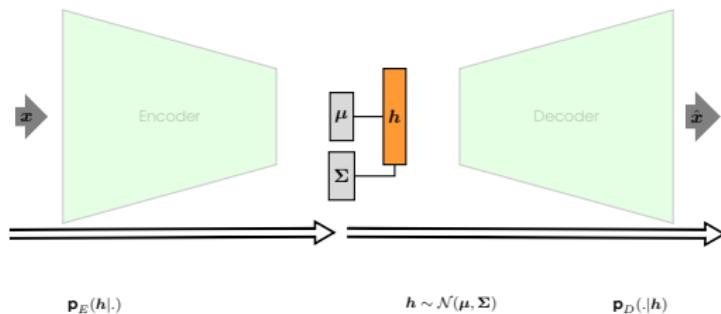
ENCODER-DECODER



Encoder-Decoder

- ▶ MLP
- ▶ CNN
- ▶ RNN

REPARAMETERIZATION TRICK



$$\mathbf{h} = \mu(\mathbf{x}, \phi) + \sigma(\mathbf{x}, \phi) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

```
z_mean, z_log_var = inputs  
batch_size = K.shape(z_mean)[0]  
epsilon = K.random_normal(shape=(batch_size, latent_dim), mean=0., stddev=1.)
```

KL DIVERGENCE

$$KL(\mathbf{p}_\phi(\mathbf{h}|\mathbf{x})||\mathbf{p}(\mathbf{h})) = \frac{1}{n} \left[-\frac{1}{2} \sum_{i=1}^d \left(1 + \log \sigma_i^f(\mathbf{x}) - (\mu_i^f(\mathbf{x}))^2 - \sigma_i^f(x) \right) \right]$$

```
z_mean, z_log_var = encoder(x) of  
kl_loss = - 0.5 * K.sum(1 + z_log_var - K.square(z_mean) - K.exp(z_log_var), axis=-1)
```

AND NOW GATHER ALL THE STUFF

```
def VAE(input_shape, encoder, decoder, sampling_layer,beta):  
  
    x = Input(shape=input_shape, name="input")  
    z_mean, z_log_var = encoder(x)  
    z = sampling_layer([z_mean, z_log_var])  
    x_decoded_mean = decoder(z)  
    vae = Model(x, x_decoded_mean)  
  
    # Loss function : Evidence Lower Bound (ELBO)  
    xent_loss = dim * metrics.binary_crossentropy(K.flatten(x), K.flatten(x_decoded_mean))  
    kl_loss = - 0.5 * K.sum(1 + z_log_var - K.square(z_mean) - K.exp(z_log_var), axis=-1)  
    vae_loss = K.mean(xent_loss + beta*kl_loss)  
  
    vae.add_loss(vae_loss)  
    vae.compile(optimizer='adam')  
    return vae
```



LIMOS
LABORATOIRE DINFORMATIQUE,
DE MODÉLISATION ET OPTIMISATION DES SYSTÈMES

