Day 4: Probabilistic ML

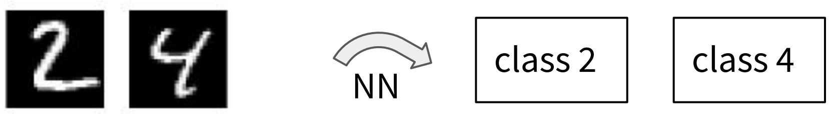
and considerations for physical data

Vanessa Boehm, March 10 2022
LSSTC-DSFP Session 14

*exceptions apply

Deep Neural Models

Deep Classification Networks



Learn a *classification* task
generally supervised and non-probabilistic*

*exceptions apply

Deep Neural Models

Deep Classification Networks

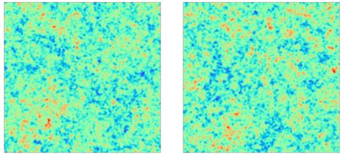
NN

class 2

class 4

Learn a *classification* task
generally supervised and non-probabilistic*

Deep Regression Networks



NN

S_8=0.55

S_8=0.78

Learn a *regression* task
generally supervised and non-probabilistic*

*exceptions apply

Deep Neural Models

Deep Classification Networks

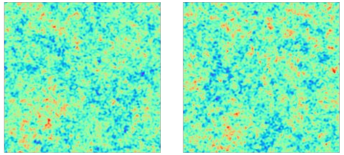
NN

class 2

class 4

Learn a *classification* task
generally supervised and non-probabilistic*

Deep Regression Networks



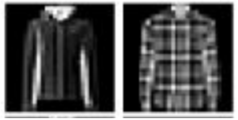
NN

S_8=0.55

S_8=0.78

Learn a *regression* task
generally supervised and non-probabilistic*

Deep Generative Networks



NN

density estimation

log p = -254

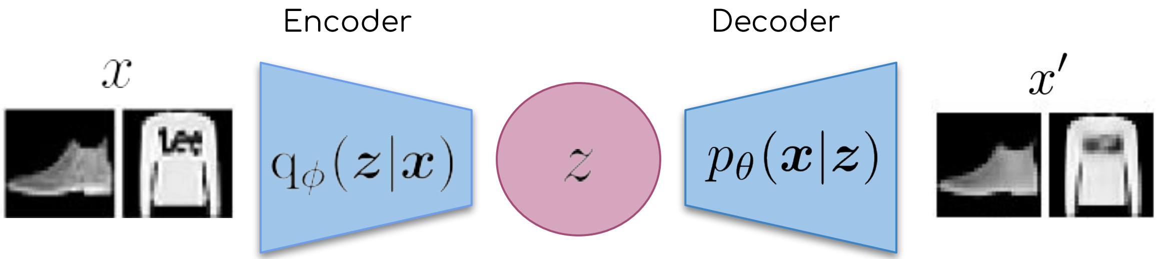
log p = -332

data generation

Density estimation and *data generation*
generally unsupervised and fully probabilistic

Example: Variational Autoencoder

Kingma & Welling 2013, Rezende 2014 + countless variants

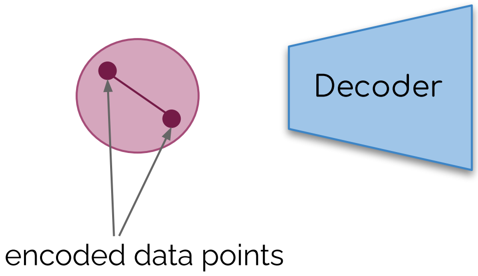


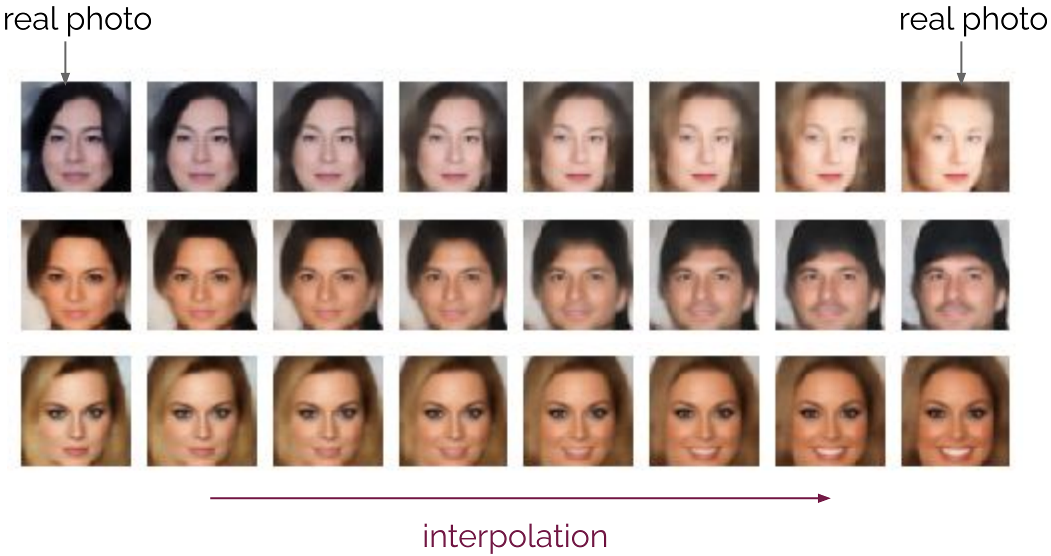
density estimation :

$$
\begin{aligned}
\ln p_\theta(\boldsymbol{x}) &= \ln \int d\boldsymbol{z}\, p(\boldsymbol{z}) p(\boldsymbol{x}|\boldsymbol{z}) \\
&\geq \underbrace{\mathbb{E}_{q_\phi(z|x)} \left[\ln p_\theta(\boldsymbol{x}|\boldsymbol{z})\right]}_{\text{reconstruction error}} - \underbrace{D_{\mathrm{KL}} \left[q_\phi(\boldsymbol{z}|\boldsymbol{x}) || p(\boldsymbol{z})\right]}_{\text{regularization}}
\end{aligned}
$$

Gaussian prior

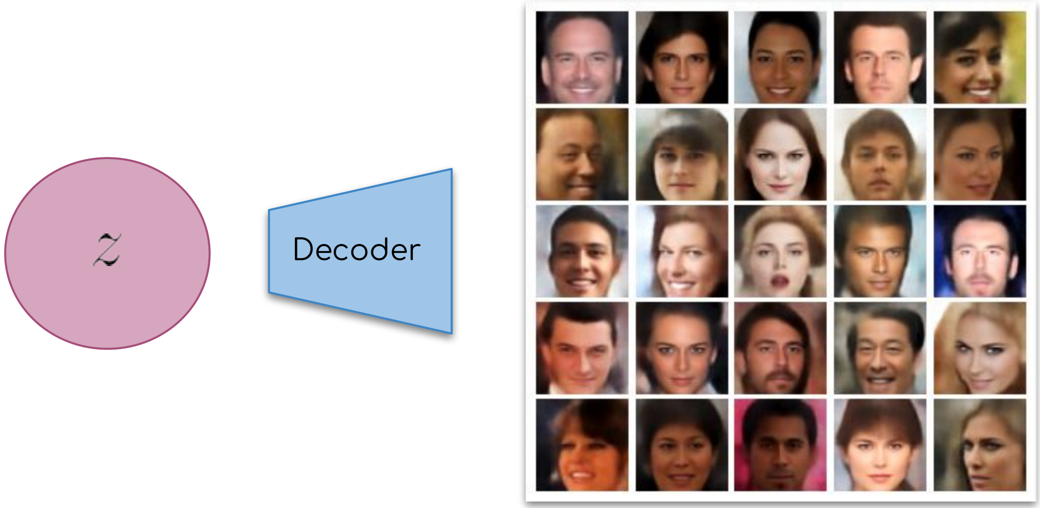
Applications of VAEs: Data Interpolation





Applications of VAEs: Data Generation

- Sample from Gaussian prior and decode

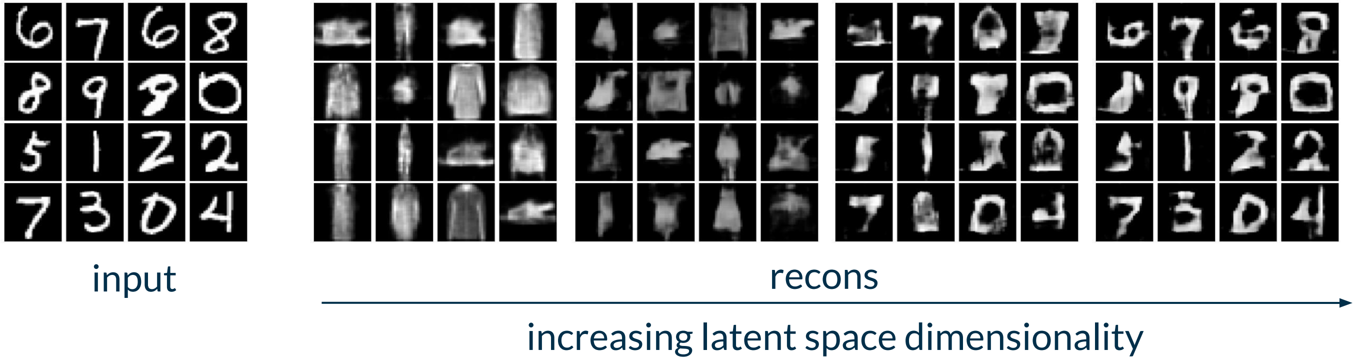


Applications of VAEs: Anomaly detection

- Option 1: Use reconstruction error as anomaly metric
    - Problem high dimensional latent spaces and powerful decoders can result in small reconstruction errors even for anomalous data points

Applications of VAEs: Anomaly detection

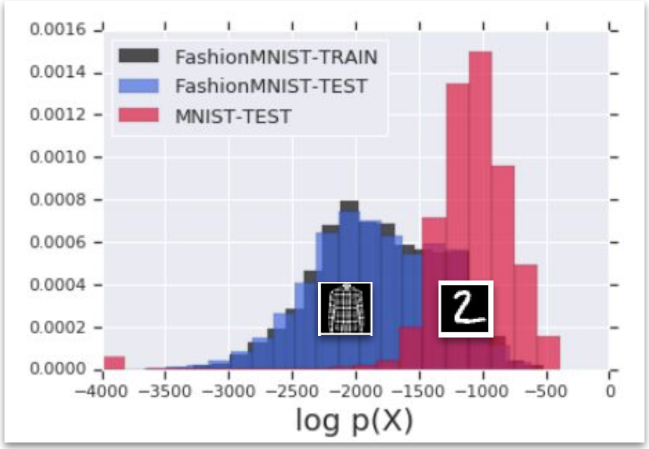
- Option 1: Use reconstruction error as anomaly metric
  - Problem high dimensional latent spaces and powerful decoders can result in small reconstruction errors even for anomalous data points



Applications of VAEs: Anomaly detection

Option 2: Use p(x) estimate (ELBO for VAE) as anomaly metric

- Problem: Tends to fail for various reasons, sometimes *catastrophically*
  (Choi et al 2018, Nalisnick et al 2019a, Hendrycks et al 2019)



Applications of VAEs: Reconstruction of Corrupted Data

Applications of VAEs: Reconstruction of Corrupted Data

observed data

noise

$$\boxed{y = Ax + n}$$

corruption operator

uncorrupted data/signal

$$p(x|y) = p(y|x)p(x)$$

high dimensional posterior

unknown prior/ data distr.

Reconstruction of Corrupted Data

1. Train a Variational Autoencoder on uncorrupted data
2. Replace ***x*** by it's generative process ***g(z)***
3. The new, exact prior distribution is Gaussian

observed data

noise

$$y = A(g_\theta(z)) + n'$$

corruption operator

generative model

low dimensional posterior

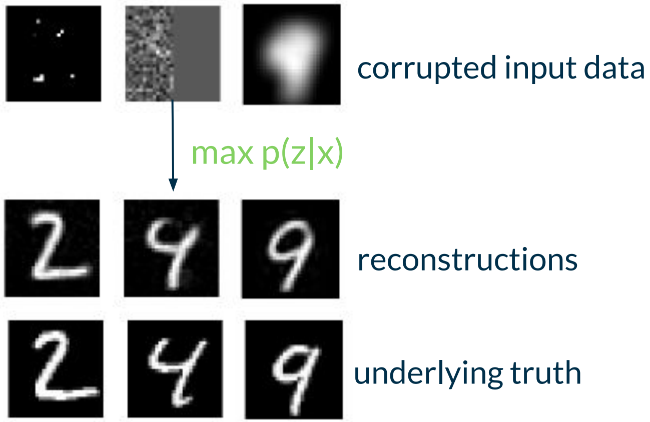
known prior

$$p(z|y) = p(y|z)p(z)$$

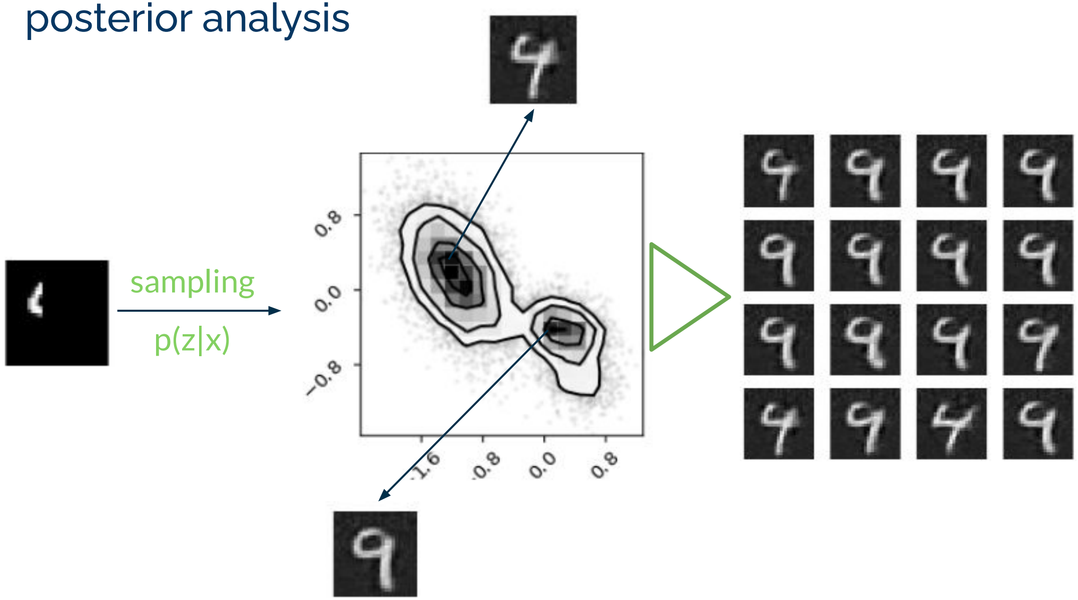
Reconstruction of Corrupted Data

e.g. Boehm et al. 2019

reconstruction



posterior analysis



Problems with VAEs

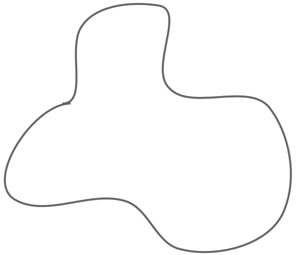
- VAEs often struggle to maximize both terms in the ELBO at the same time
    - The encoded distribution is often not perfectly Gaussian
    - The approximate posterior distribution is often a bad approximation to the true one. Don't use it! (exercise)
    - Lots of hyperparameters need to be optimized to obtain the desired results (e.g. sample size in the training, form of likelihood etc)
    - Vast literature on how to improve VAEs… E.g. beta-VAE, where a scalar parameter beta is used to up- or downweight the KL-term.

Another density estimator: Normalizing Flows

NFs are bijective models. No data compression only transformation!

e.g. RealNVP (Dinh et al. 2019), Glow (Kingma et al 2018), MAF (Papamakarios 2017), NSF (Durcan 2019), SINF (Dai et al 2021)

data space distribution



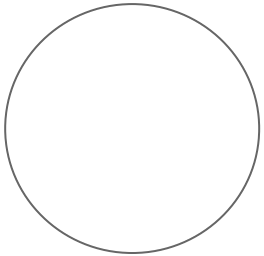
encoder

$$\boldsymbol{z} = \boldsymbol{b}_\theta(\boldsymbol{x})$$

$$\boldsymbol{x} = \boldsymbol{b}_\theta^{-1}(\boldsymbol{z})$$

decoder

latent space distribution



density estimation : $\ln p_\theta(\boldsymbol{x}) = \ln q(\boldsymbol{z}) + \ln |\nabla_{\boldsymbol{x}} \boldsymbol{b}_\theta(\boldsymbol{x})|$

Jacobian of transformation

Gaussian distribution

Another density estimator: Normalizing Flows

NFs are bijective models -> no data compression only transformation!

e.g. RealNVP (Dinh et al. 2019), Glow (Kingma et al 2018), MAF (Papamakarios 2017), NSF (Durcan 2019), SINF (Dai et al 2021)

data space distribution

encoder

$$\boldsymbol{z} = \boldsymbol{b}_\theta(\boldsymbol{x})$$

$$\boldsymbol{x} = \boldsymbol{b}_\theta^{-1}(\boldsymbol{z})$$

decoder

latent space distribution

density estimation : $\ln p_\theta(\boldsymbol{x}) = \ln q(\boldsymbol{z}) + \ln |\nabla_{\boldsymbol{x}} \boldsymbol{b}_\theta(\boldsymbol{x})|$

Jacobian of transformation

Gaussian distribution

Today's exercise

- Use a normalizing flow to improve the VAE training. If we use an NF as prior it helps the VAE achieve a Gaussian prior distribution!

- Find out how well $q(z|x)$ matches $p(z|x)$

- Reconstruct corrupted data by maximizing the posterior $p(z|x)$