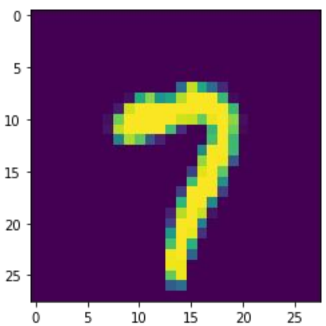
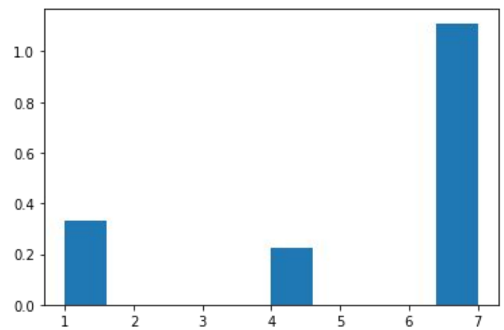
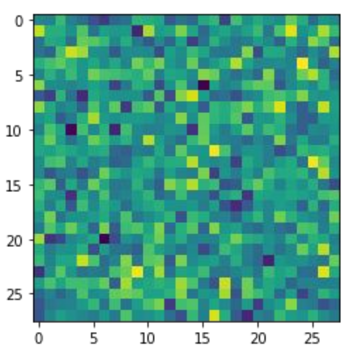
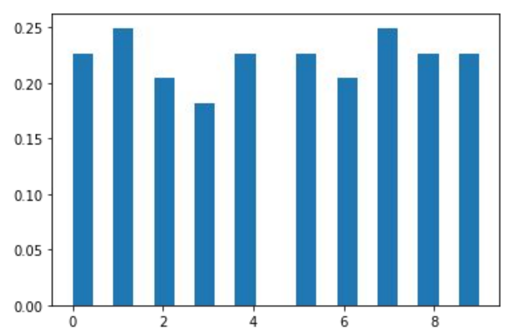
Bayesian Neural Networks and VAE

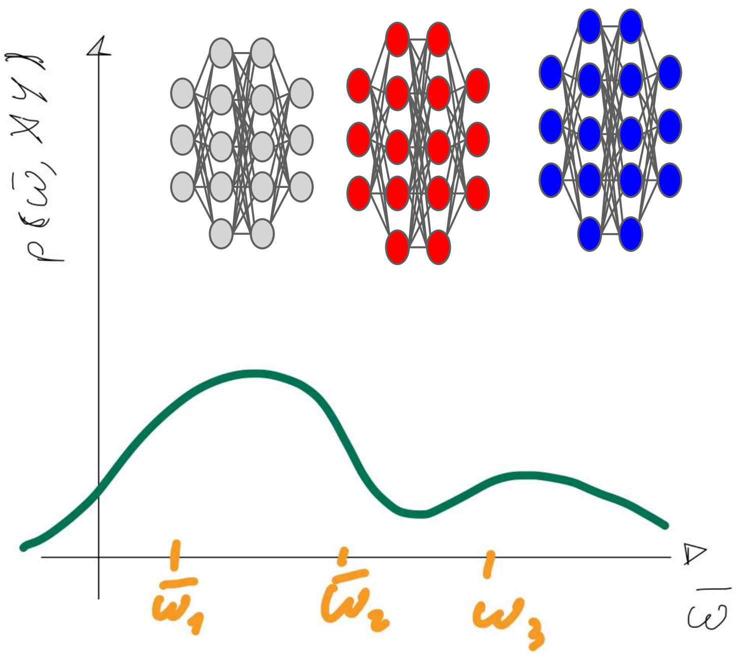
Prof. Dr. Nico Serra - University of Zurich



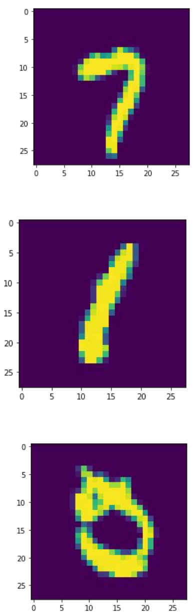


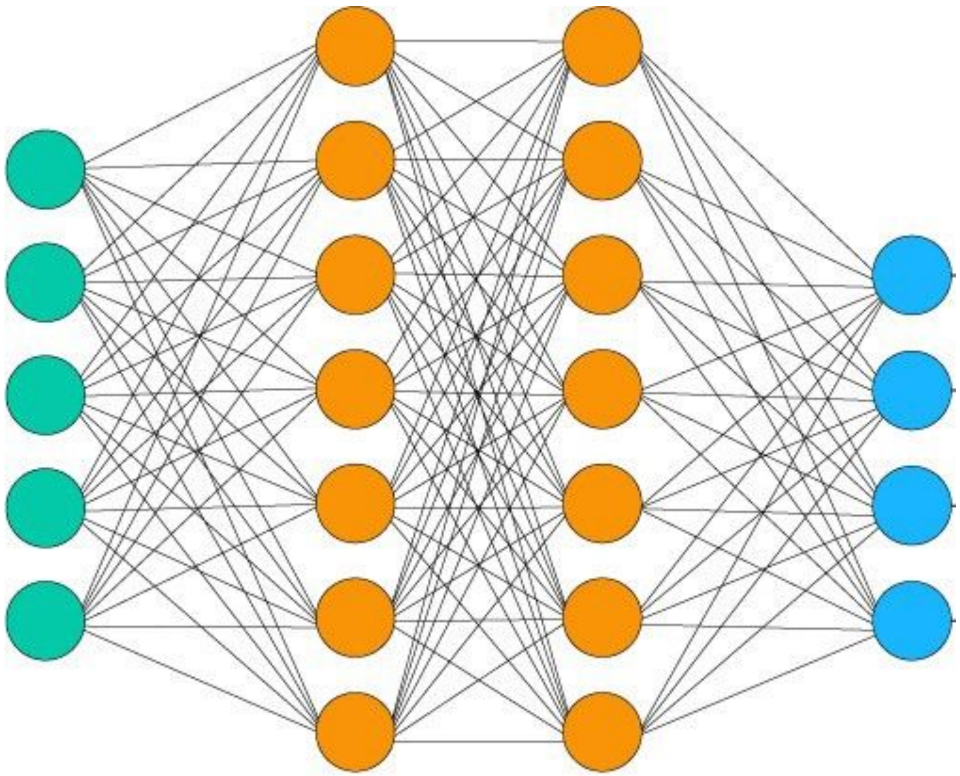


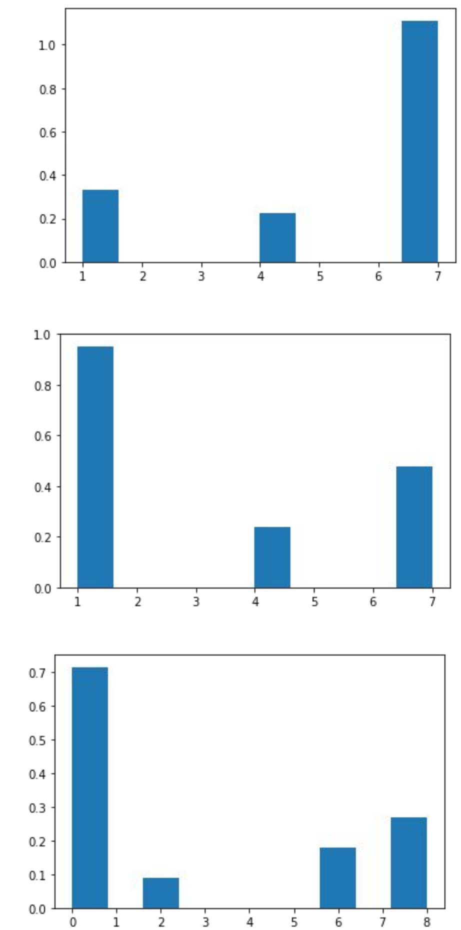




Introduction







$$max_{\varpi} \sum_i \log\left\{p\left(y_i|x_i, \varpi\right)\right\}$$

MISIS National University of Science and Technology

University of Zurich[UZH]

Optimization NN

Training:

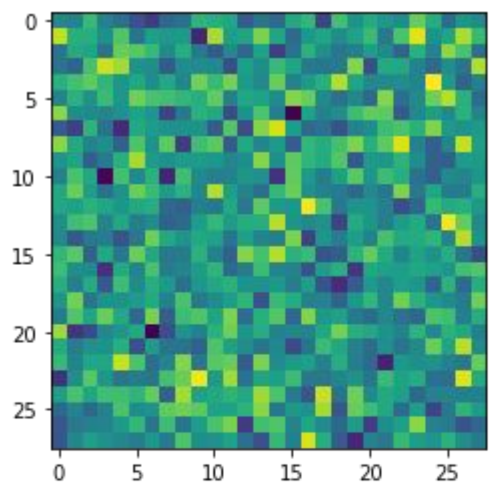
- Forward pass with X
- Calculate error with respect to y
- Back propagation and stochastic gradient descent

$$\{x_j\} \sim Unif\left(x_1, x_2, x_3, \ldots, x_N\right)$$

$$\omega_J \rightarrow \omega_J - \eta \frac{\partial}{\partial \omega_J} \sum_i \log\left\{p\left(y_i | x_i, \varpi\right)\right\}$$

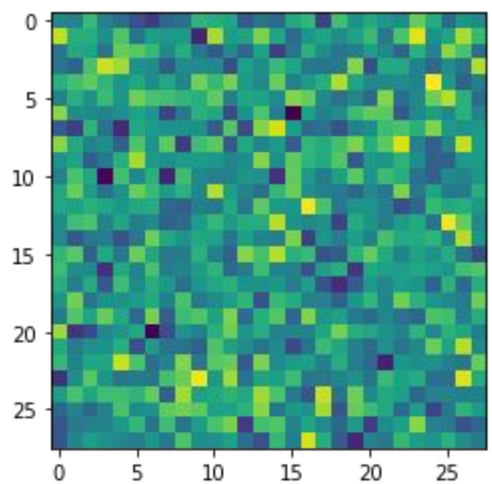
Why Bayesian Networks

- Suppose now you have an image not belonging to any of the class
- How would you want your network to classify it?

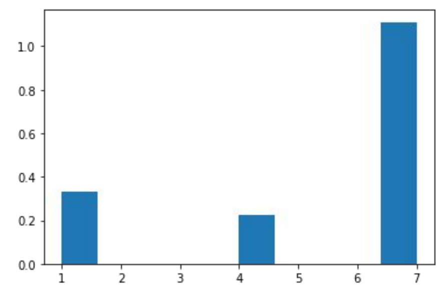


Why Bayesian Networks

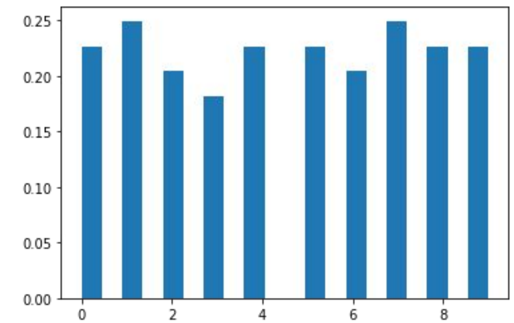
- Suppose now you have an image not belonging to any of the class
- How would you want your network to classify it?



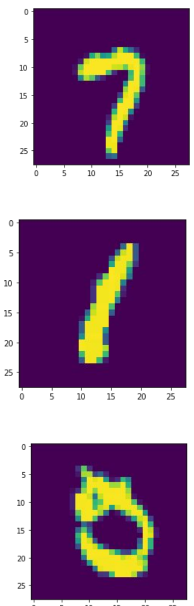
Overconfident Answer

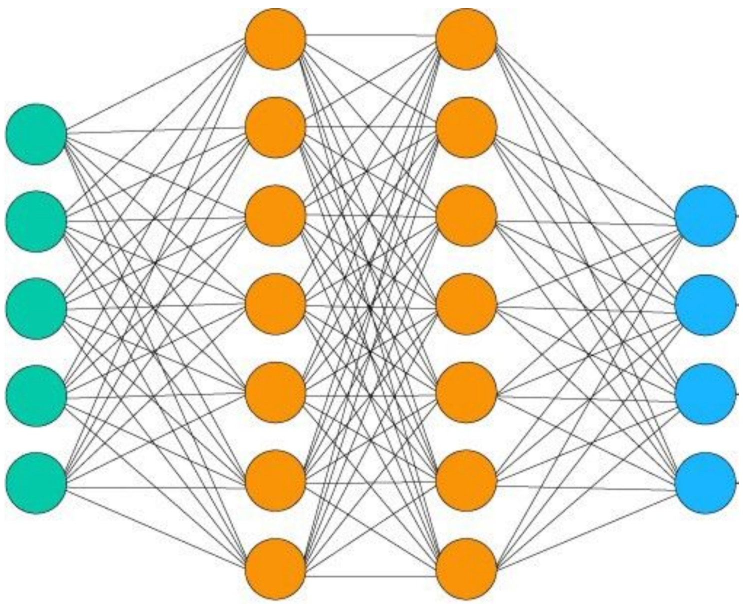


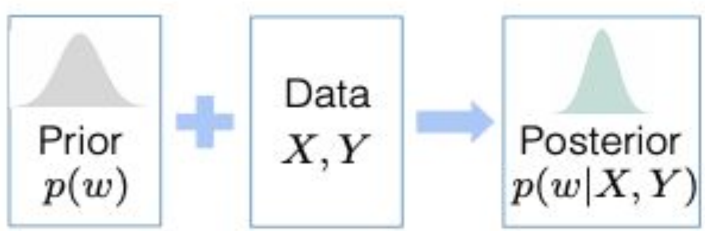
Desired Answer



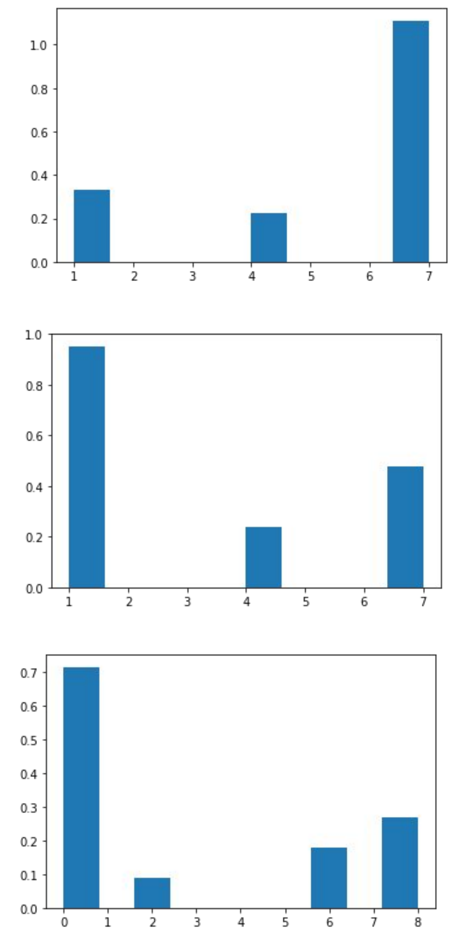
Bayesian NN







$$\mathbb{E}_{p(w|X,Y)} p(y_* | x_*, w)$$

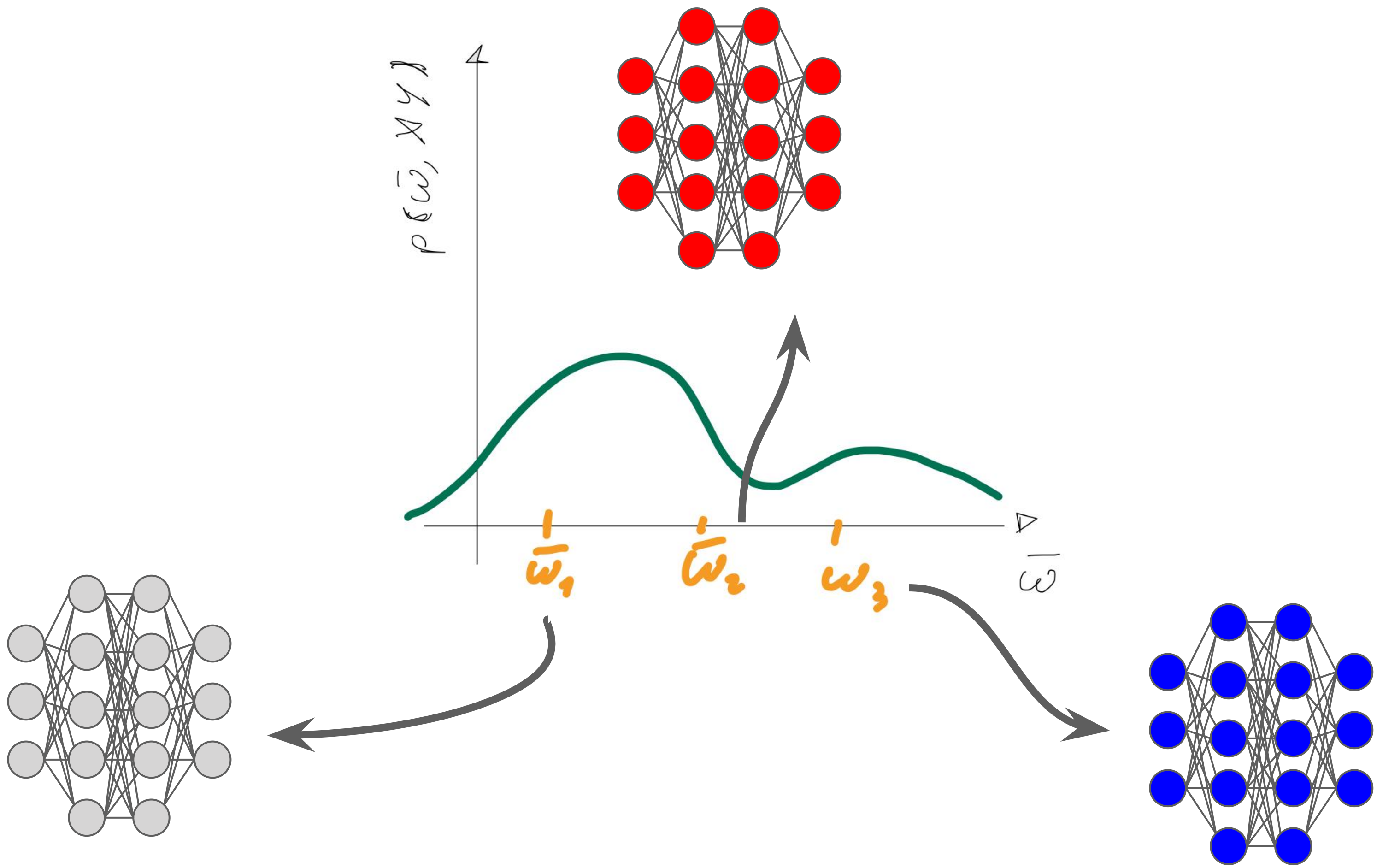


Bayesian Neural Network Optimization

- Bayesian neural networks can be seen as an ensemble of neural networks
- The training consists of finding $p(\omega|X, Y)$
- The prediction give by

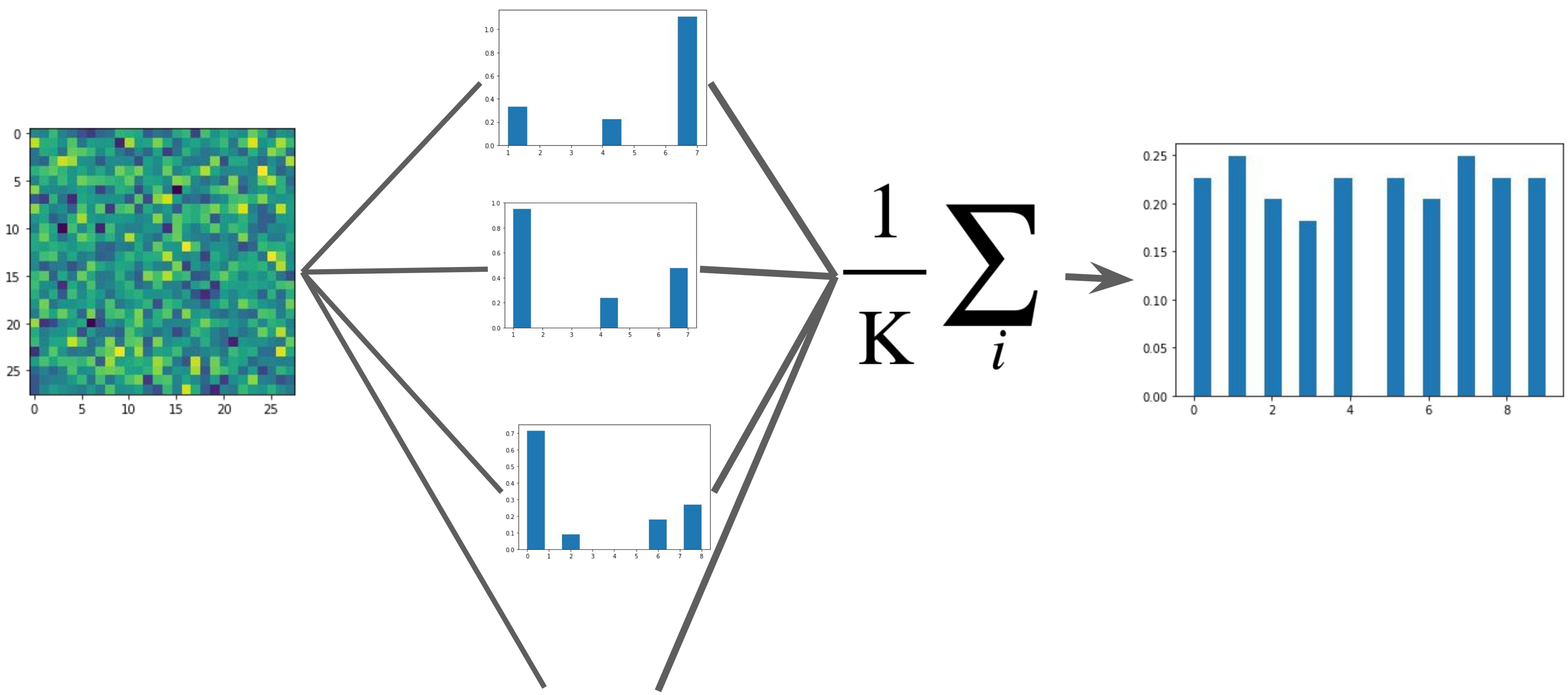
$$\mathbb{E}_{p(w|X,Y)} p(y_*|x_*, w) \approx \frac{1}{K}\sum_{k=1}^{K} p(y_*|x_*, w^k), \quad w^k \sim p(w|X,Y)$$

Bayesian Neural Networks



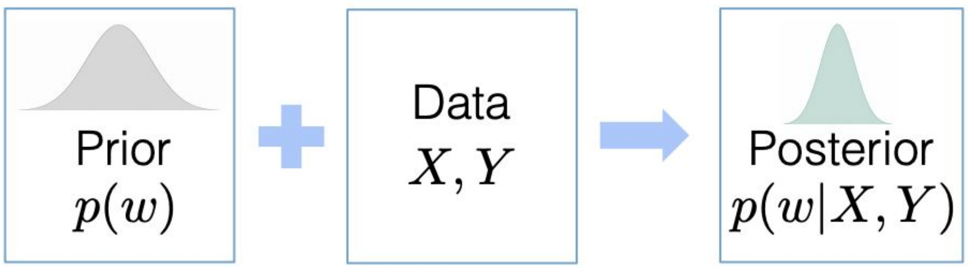
Why Bayesian Networks

- Out-of-domain point



Training BNN

Bayesian Inference:



- Generally BNN have too many parameters to use efficiently MCMC, suitable for Variational Inference

$$\min_{\lambda} KL(q(\omega|\lambda) || p(\omega|X, Y))$$

MISIS National University of Science and Technology

University of Zurich UZH

Training BNN

$$\max_{\lambda} \sum_{i=1}^{N} \underbrace{\mathbb{E}_{q(w|\lambda)} \log p(y^i | x^i, w)}_{\text{Data term}} - \underbrace{KL(q(w|\lambda)||p(w))}_{\text{Regularizer}}$$

Sample mini-batch from data

Sample weight from q (using reparametrization trick)

$$\sum_{j=1}^{m} \log p(y^{i_j} | x^{i_j}, w = f(\lambda, \epsilon^j)), \quad \epsilon^j \sim p(\epsilon) \quad i_j \sim \mathrm{Unif}(1, \ldots, N)$$

Training BNN

$$q(w|\lambda) = \mathcal{N}(\mu, \sigma^2), \quad \lambda = \{\mu, \sigma\}$$

Property of normal distribution:

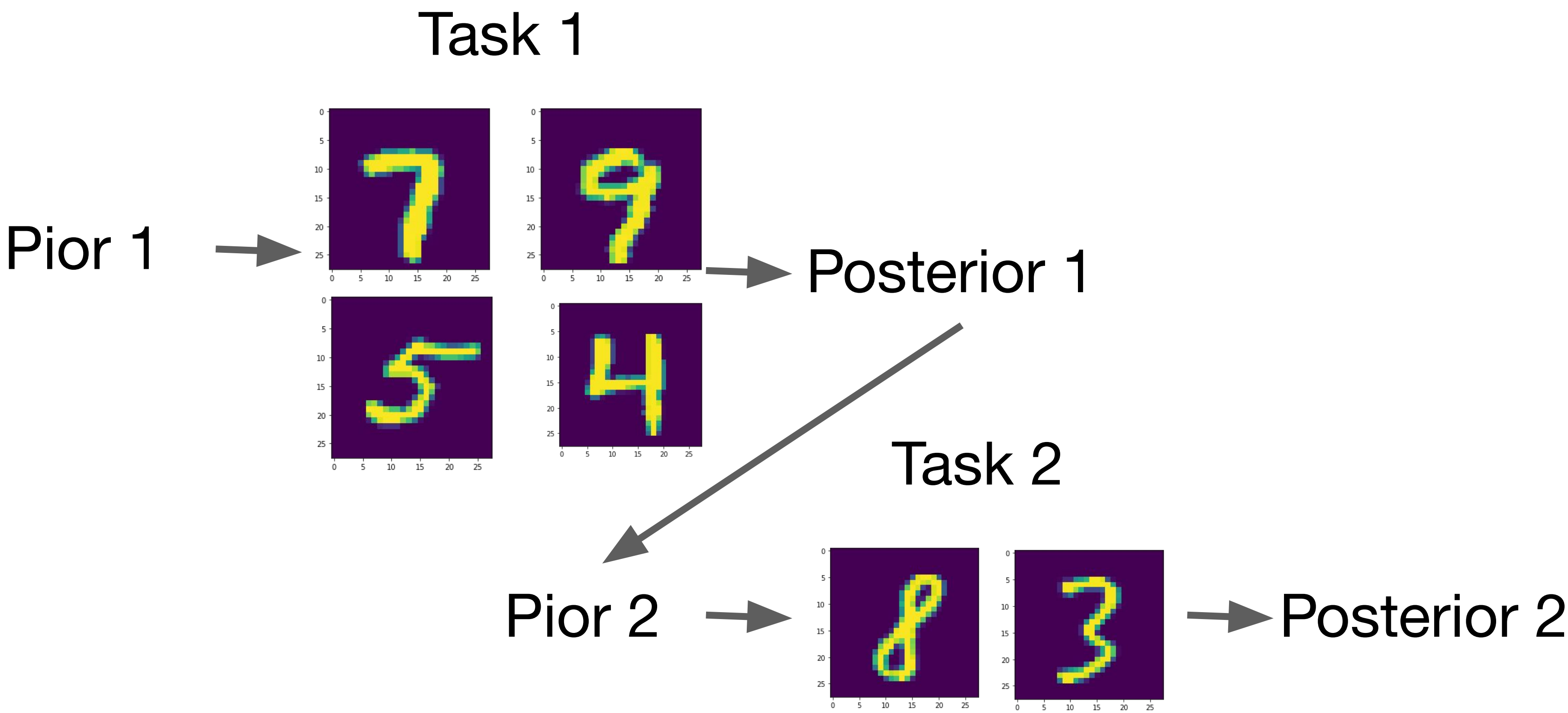
$$w \sim \mathcal{N}(\mu, \sigma^2) \quad \Leftrightarrow \quad w = \mu + \sigma\epsilon, \;\; \epsilon \sim \mathcal{N}(0, 1)$$

Gradient Descent

$$\lambda^{new} = \lambda^{old} + \eta \frac{\partial}{\partial \lambda} \sum_{j=1}^{m} \log p(y^{i_j} | x^{i_j}, w = f(\lambda^{old}, \epsilon^j)), \quad i_j \sim \mathrm{Unif}(1, \ldots, N)$$

$$\epsilon^j \sim p(\epsilon)$$

Continue Learning



- BNN tend to keep better memory of previous task when retrained for new tasks

Advantages of BNN

- Prior of BNN can be use to encode desired properties of the network
- Ensambling provides stability in the training
- Uncertainty estimation
- Better performance for online learning

MISIS National University of Science and Technology

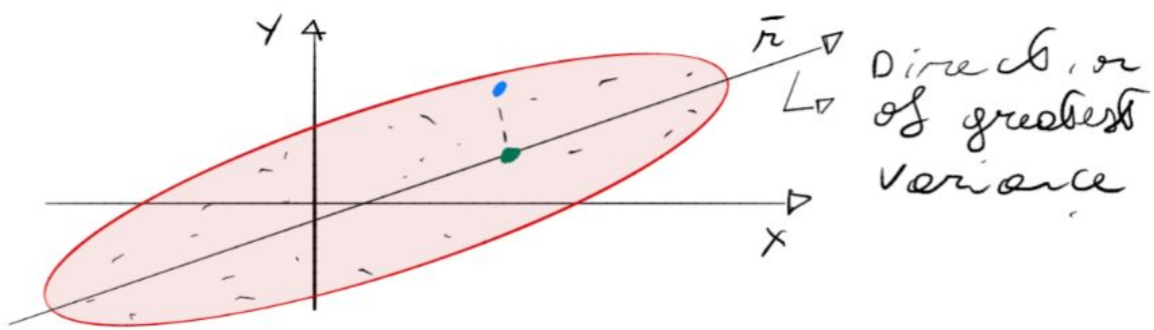
University of Zurich UZH

Variational Autoencoders

PCA

Unsupervise Learning:

- Learn the structure of data
- Learn features in data
- Learn probability distribution of data
- Compress data



PCA: Suppose that I want to represent my data with a single number I chose the direction of greatest variance

MISIS
National University of Science and Technology

University of Zurich UZH

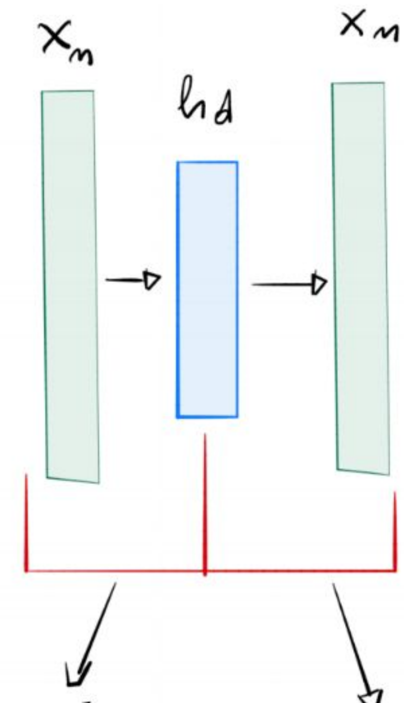
PCA

- The Principal Component Analysis (PCA) is a way of compressing the data
- If data are located on a linear manifold, it is convenient to "get rid" of reduntant dimensions
- In order to find the best representation of data in d-dimensions ($d < n$), we choose the d dimensions with greatest variance
- PCA consists of finding the d orthogonal dimensions with greatest variance, equivalent to diagonalise an n-dimension matrix and take the d-dimensional sub-matrix

PCA

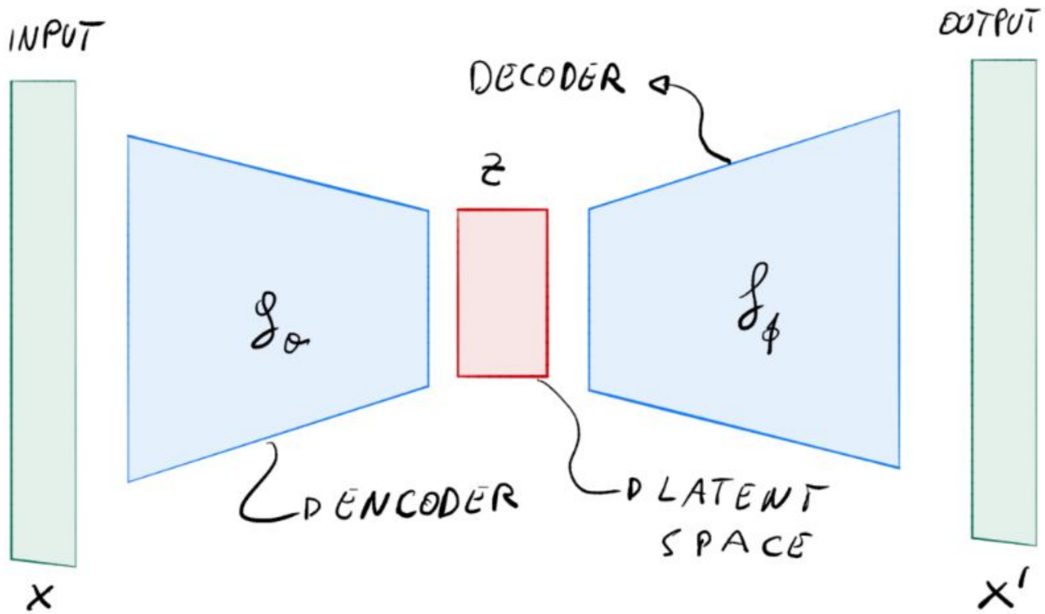
A PCA-like method can be applied with a simple ANN

- ANN with 1 hidden layer and no (linear) activation function
- The dimension of the hidden layer is d<n
- The loss consists in minimising the square error
- The hidden layer spans the same space at PCA, but the h_d neurons are NOT orthogonal
- ANN is not an efficient way to apply PCA



$F : X_n \rightarrow h_d$ $\quad$ $F^{-1} : h_d \rightarrow X_n$

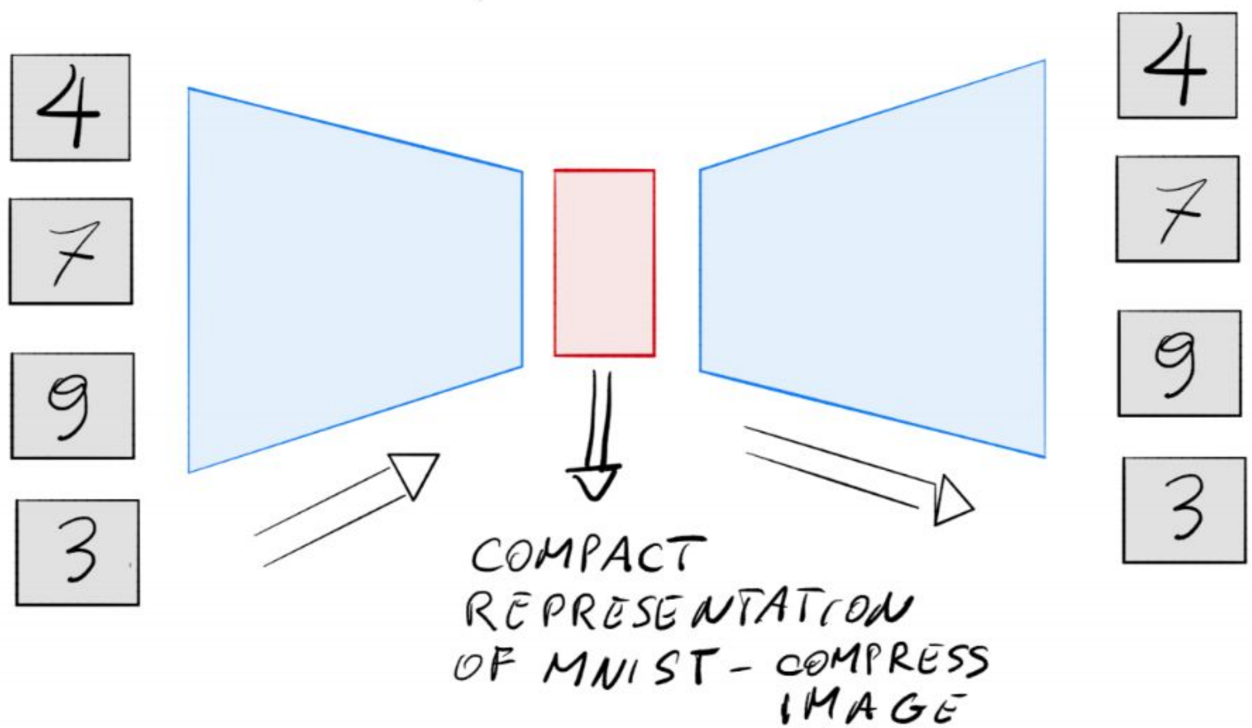
Autoencoders



$$X' = f_\phi \left[g_\theta(X) \right] \qquad Loss : \mathscr{L}(\phi, \theta) = \frac{1}{N} \sum_{i=1}^{N} \left[X_i - X_i' \right]^2$$

Autoencoders (AE) are trained to reproduce the input

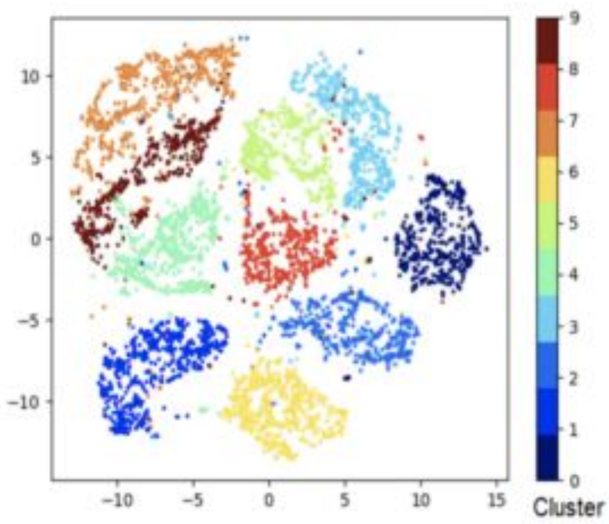
AE example

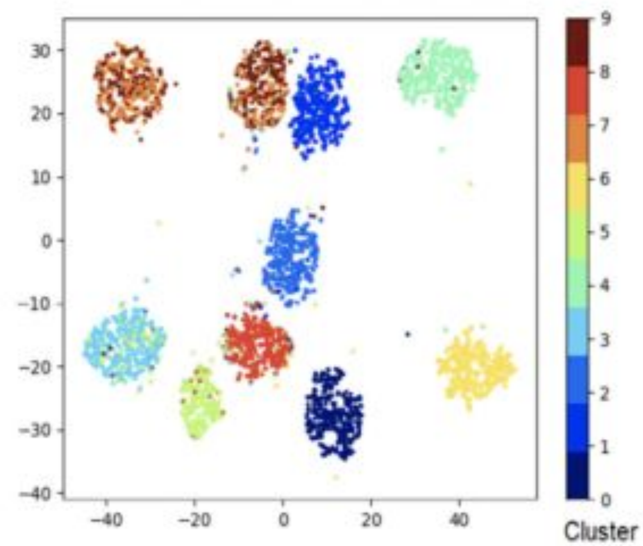


- For instance we can train the AE with the MNIST dataset to reproduce the input
- The latent space is a compact representation of the MNIST dataset

AE Latent Space

- We can visualise the latent space (in this case was a 2-d space)
- This is after training with MNIST, the color represent the different numbers





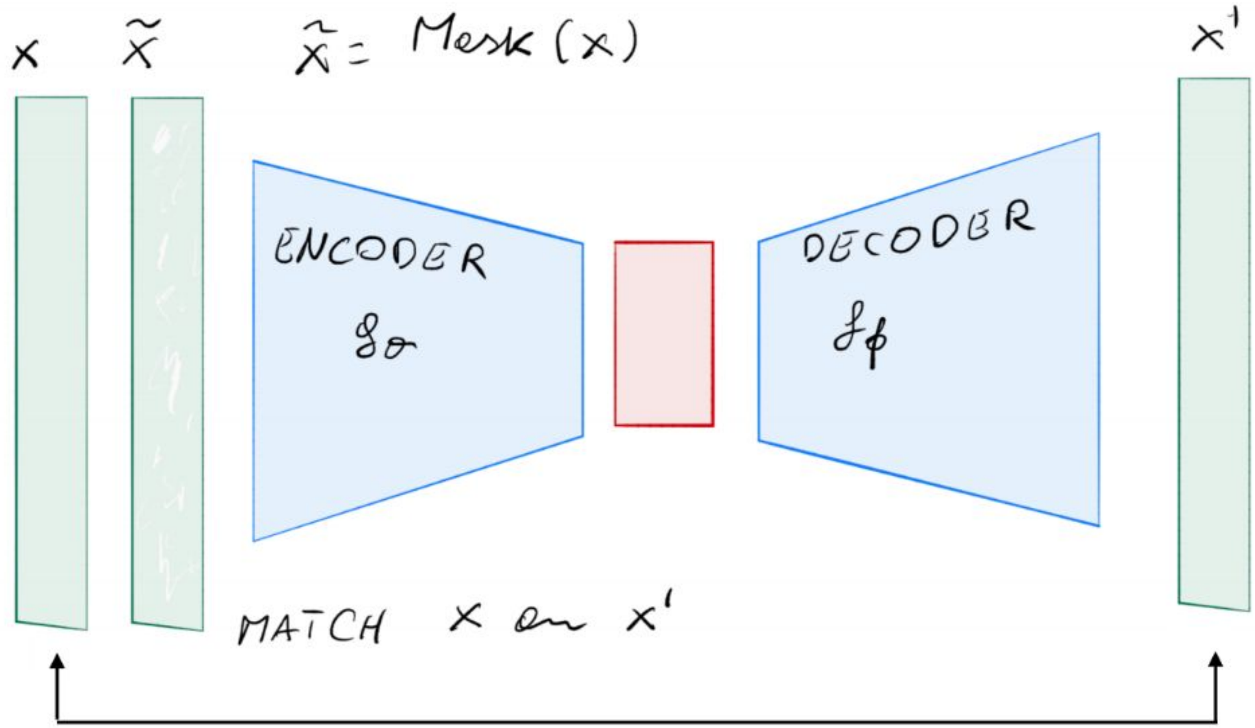
arXiv:1801.07648

Denoising Autoencoders

We can use AE to denoise the input:

- We apply a Mask (simulates noise)
- We predict X, diving as input $\tilde{X}$

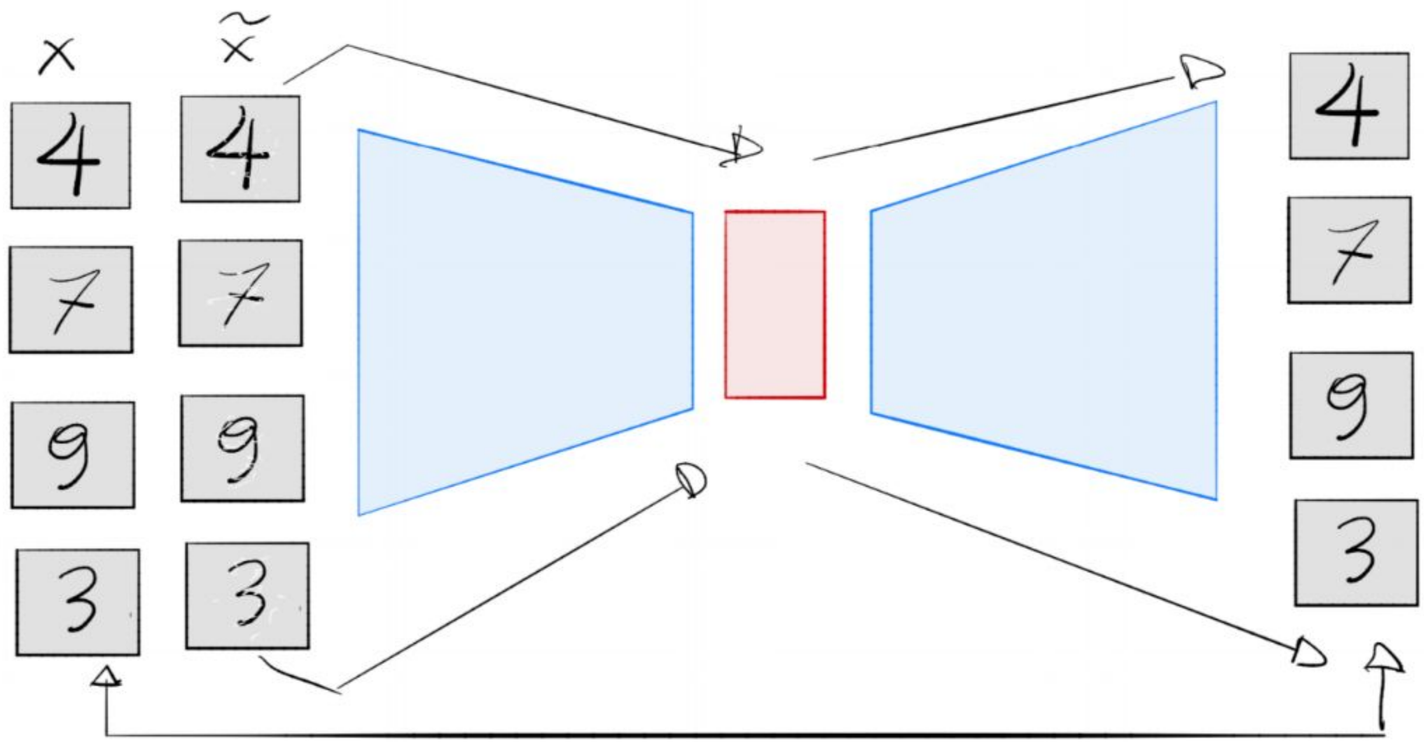
$$Loss : \mathscr{L}(\phi, \theta) = \frac{1}{N} \sum_{i=1}^{N} \left[X_i - X_i'\right]^2$$



MISIS National University of Science and Technology

University of Zurich[UZH]

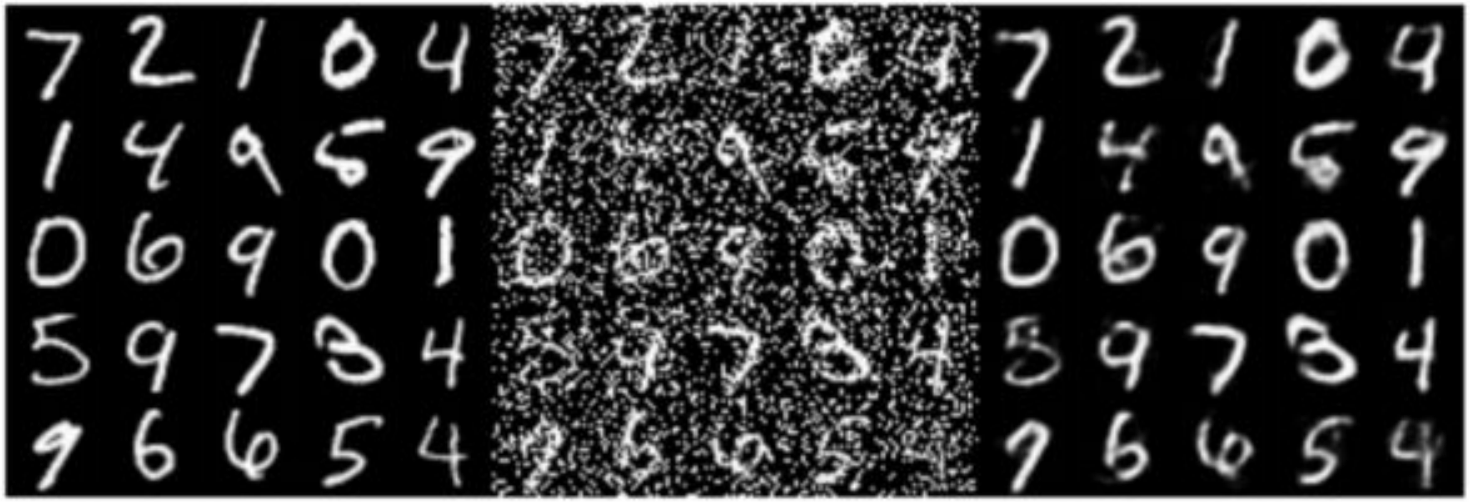
AE Example



$$Loss : \mathscr{L}(\phi, \theta) = \frac{1}{N}\sum_{i=1}^{N}\left[X_i - X_i'\right]^2$$

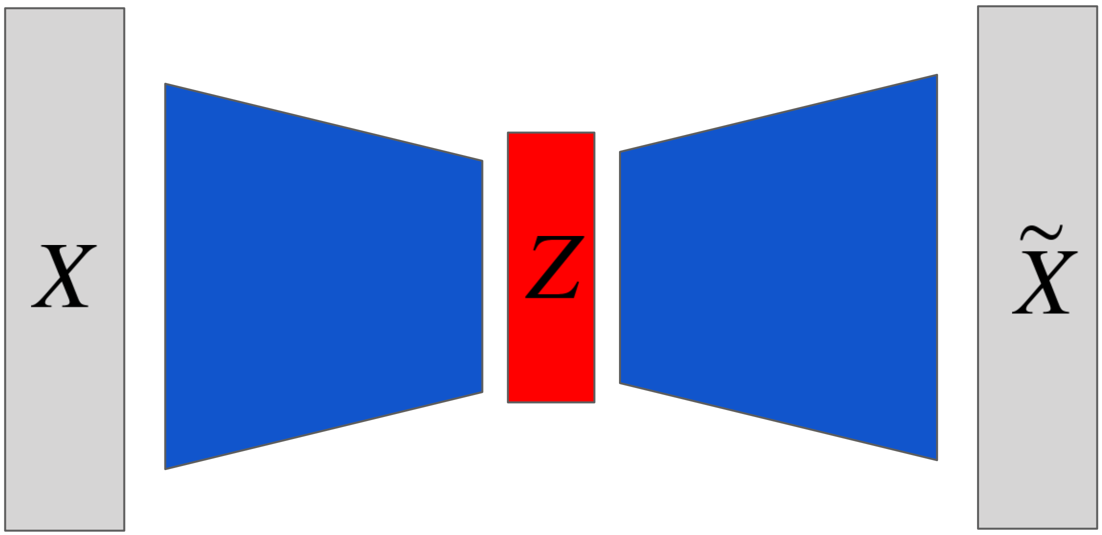
Denoising with AE

Application of denoising AE to corrupted MNIST sample

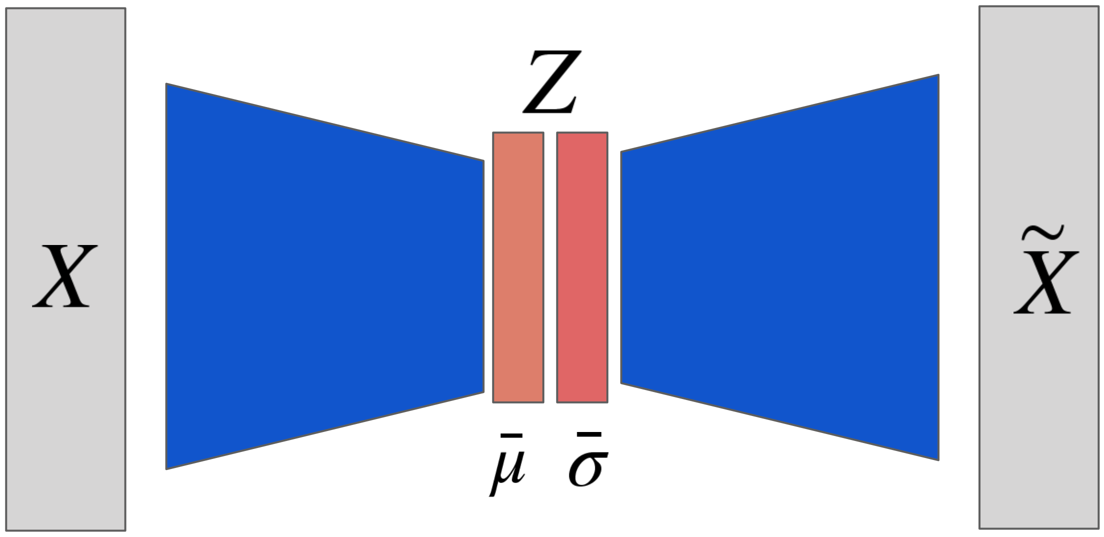


Original input, corrupted data, reconstructed data

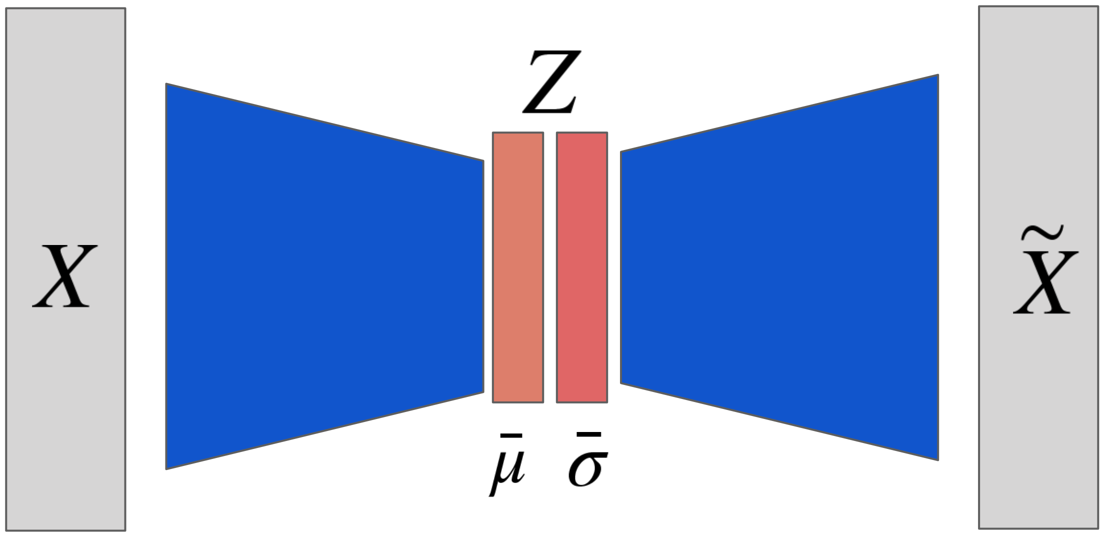
From AE to Variational Autoencoders



VAE as variational inference



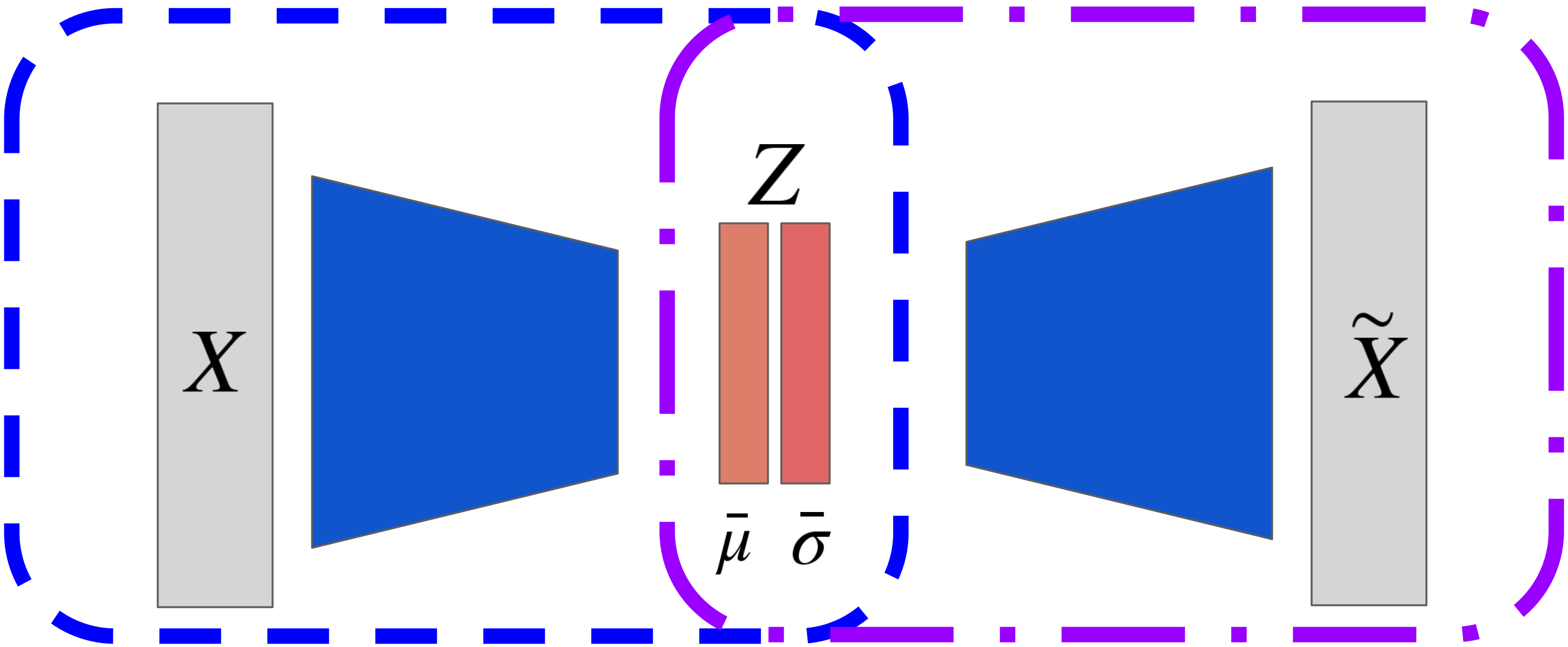
VAE as variational inference



NB: When lifting a NN to be a bayesian one, you do not need to make every single layer probabilistic, having a fewer Bayesian layers is often better for stability

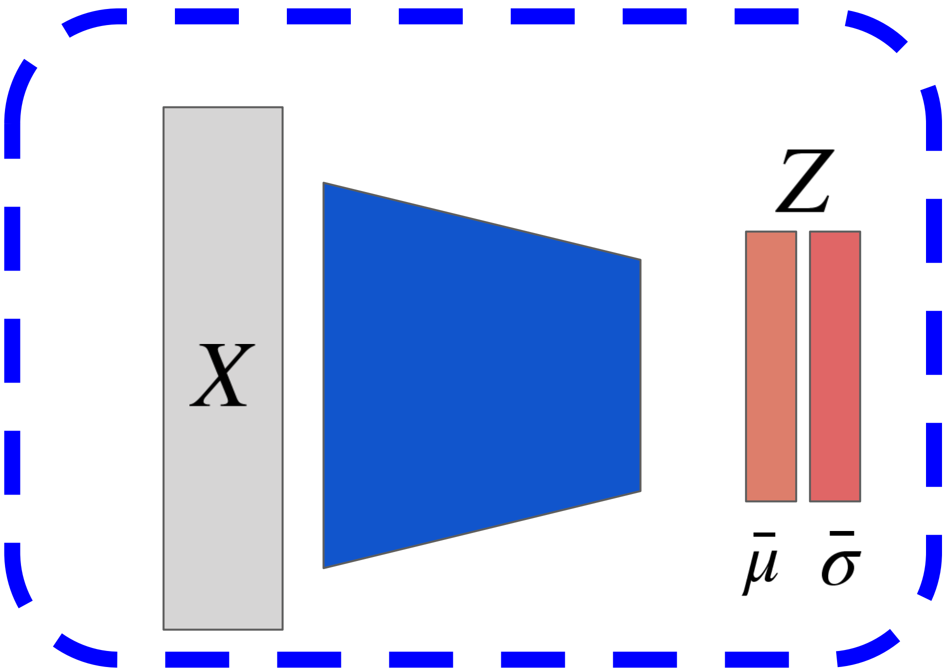
University of Zurich[UZH]

VAE as variational inference

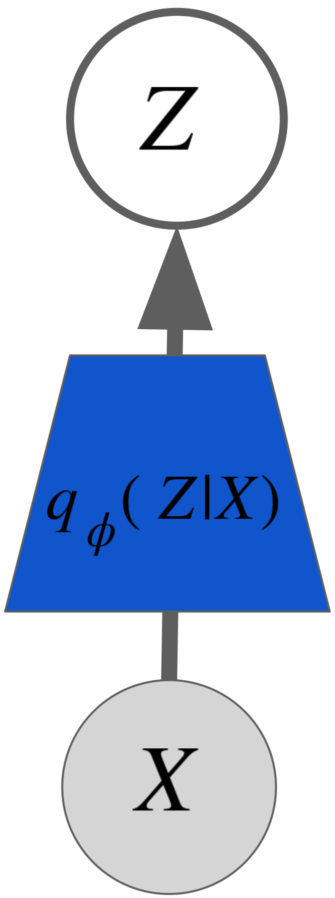


N. Serra

VAE as variational inference

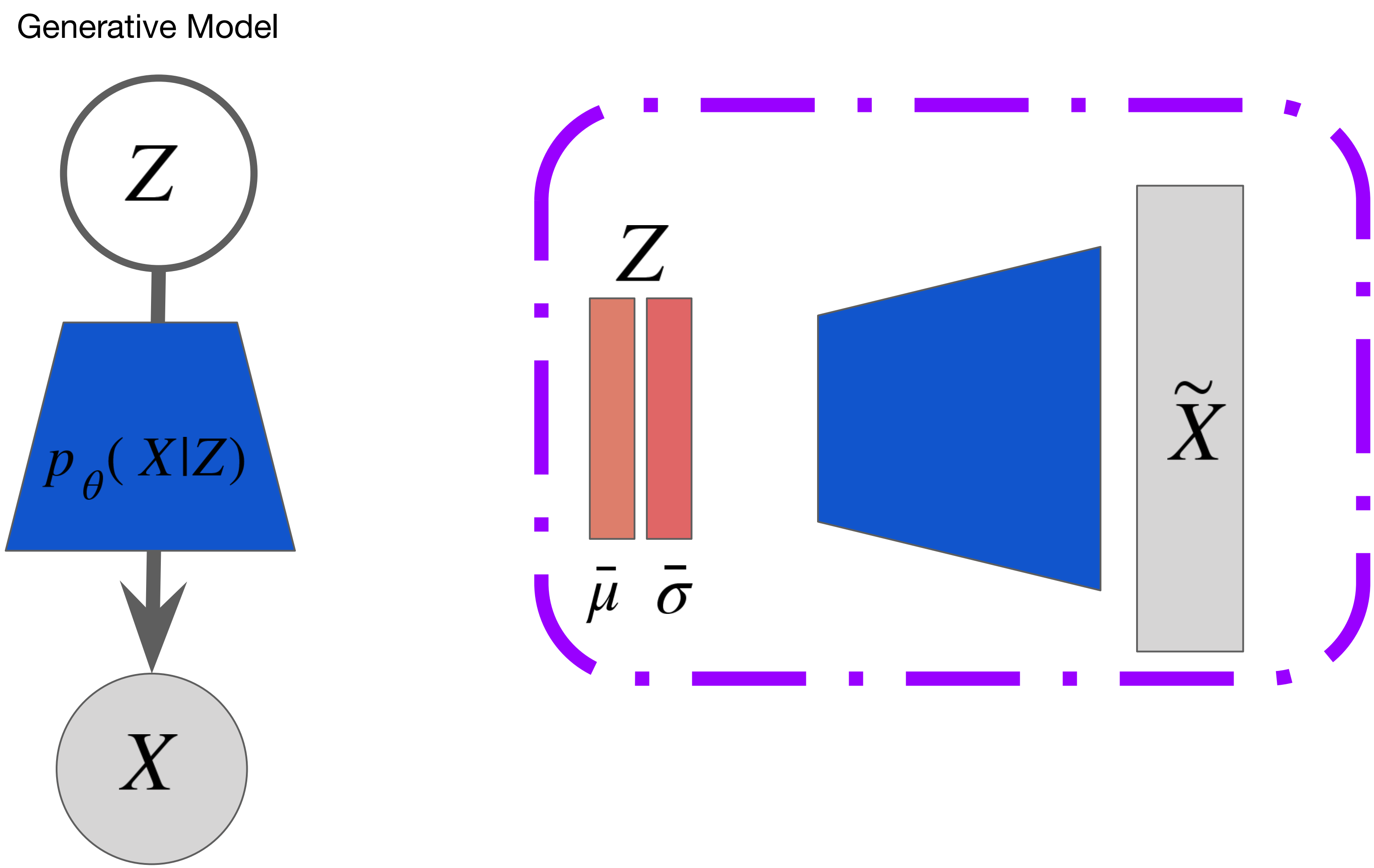


Inference Model



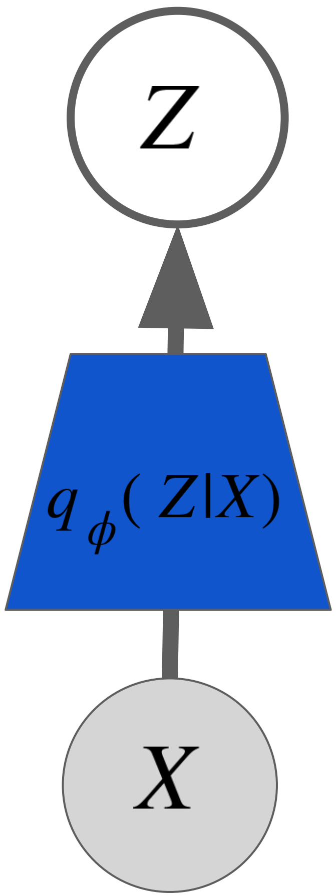
N. Serra

VAE as variational inference

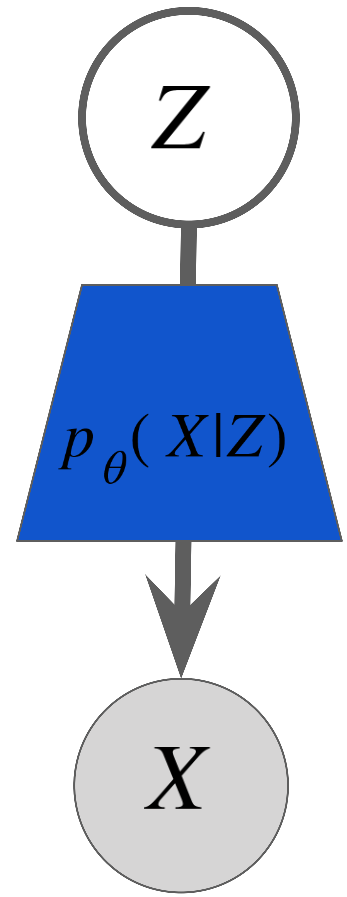


VAE as variational inference

Inference Model

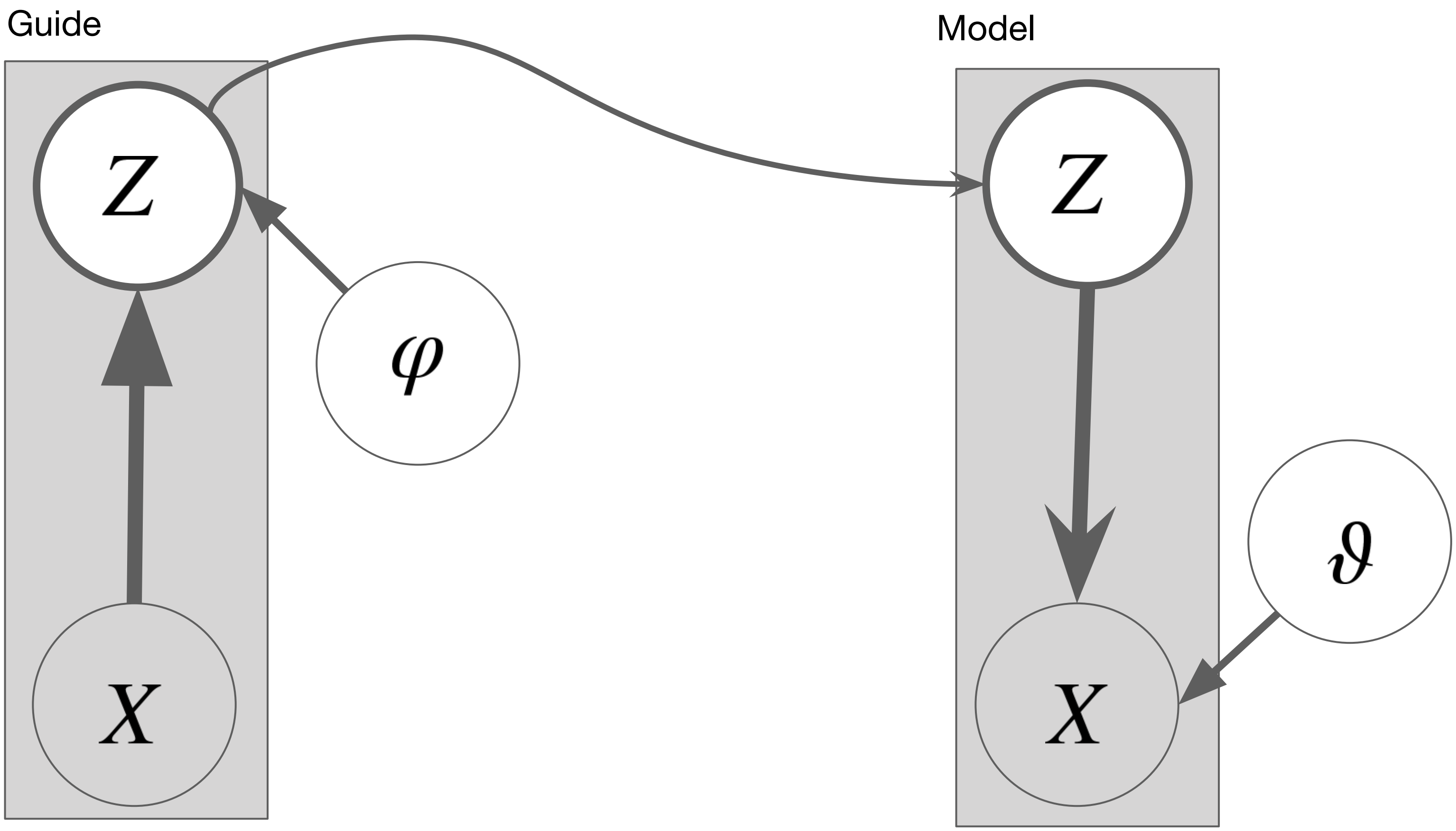


Generative Model



N. Serra

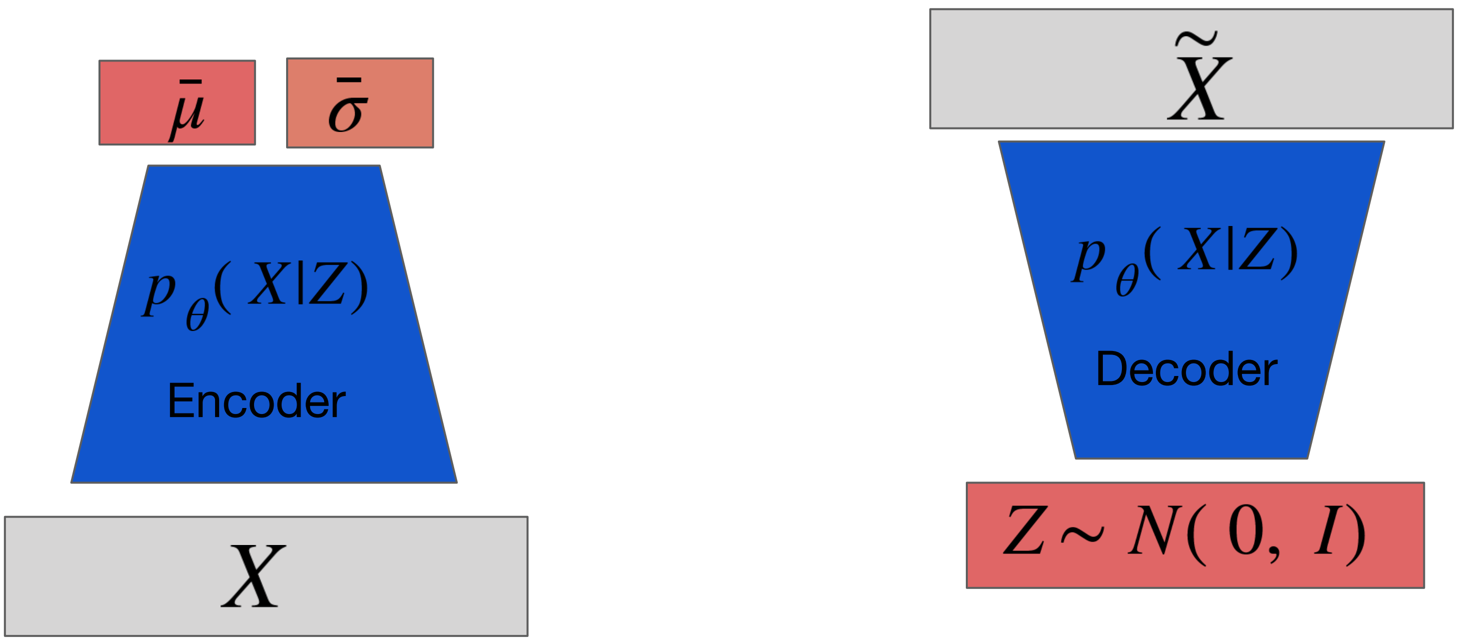
pyro.ai



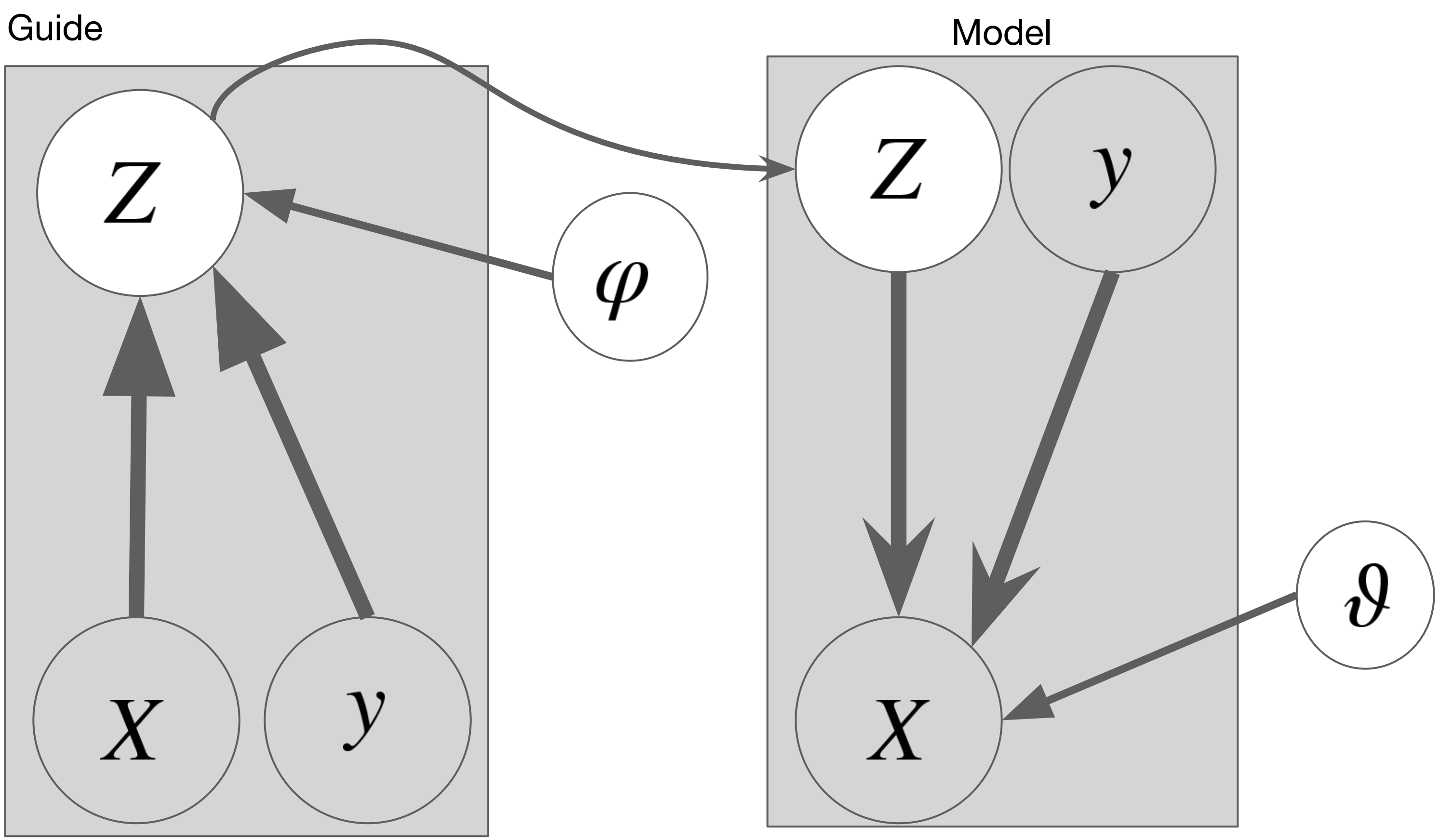
Variational Autoencoder Loss

$$KL(N\{\mu(X),\Sigma(X)\}||N(0,I))$$

$$|\tilde{X}-X|^2$$



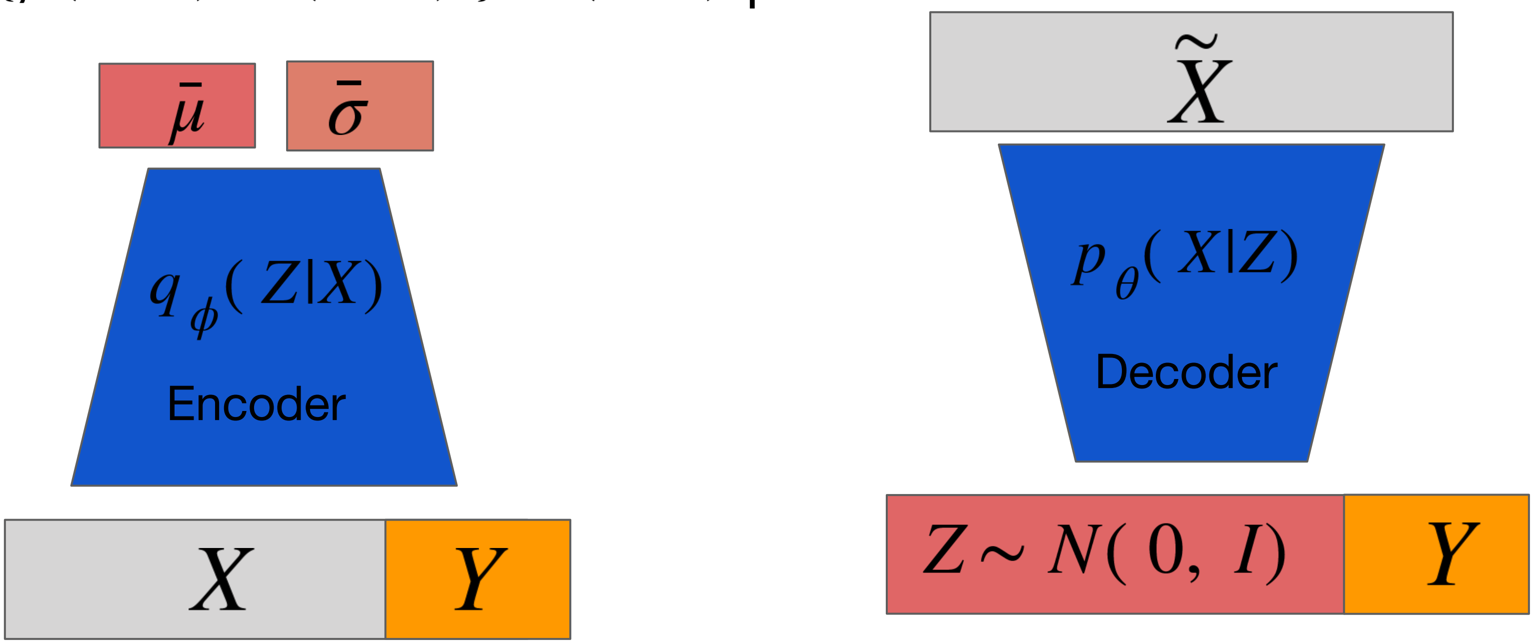
Conditional VAE



Conditional VAE

$$KL\left|N\{\mu(X,Y),\Sigma(X,Y)\}||N(0,I)\right|$$

$$\left|\tilde{X}(X,Y)-X\right|^2$$



University of Zurich[UZH]

VAE results

GANs:

VAE July 2020
(arxiv.org/abs/2007.03898):

N. Serra

Literature

Auto-Encoding Variational Bayes

Diederik P Kingma, Max Welling

Stochastic Backpropagation and Approximate Inference in Deep Generative Models

Danilo Jimenez Rezende, Shakir Mohamed, Daan Wierstra

Variational Encoders and Autoencoders : Information-theoretic Inference and Closed-form Solutions

Karthik Duraisamy