BOSTON UNIVERSITY

Variational Autoencoders (VAEs)

DL4DS – Spring 2026

DS542 Gardos
Prince, *Understanding Deep Learning*,
Rocca, "Understanding Variational Autoencoders (VAEs)", 2019
Other Content Cited

Diederik P. Kingma

2016 OpenAI, founding member
2017 PhD U. of Amsterdam
2018– Google DeepMind

Auto-Encoding Variational Bayes

Diederik P. Kingma
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

Diederik P. Kingma

FOLLOWING

Other names

Research Scientist, Google Brain
Verified email at google.com - Homepage

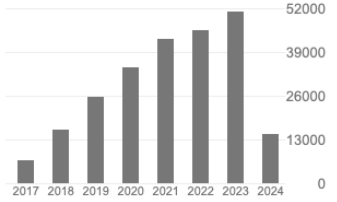
Machine Learning Deep Learning Neural Networks
Generative Models Artificial Intelligence

GET MY OWN PROFILE

TITLE	CITED BY	YEAR
Adam: A method for stochastic optimization DP Kingma, J Ba arXiv preprint arXiv:1412.6980	180174	2014
Auto-Encoding Variational Bayes DP Kingma, M Welling arXiv preprint arXiv:1312.6114	35092	2013
Semi-Supervised Learning with Deep Generative Models DP Kingma, S Mohamed, DJ Rezende, M Welling Advances in Neural Information Processing Systems, 3581-3589	3431	2014
Score-based generative modeling through stochastic differential equations Y Song, J Sohl-Dickstein, DP Kingma, A Kumar, S Ermon, B Poole arXiv preprint arXiv:2011.13456	3126	2020
Glow: Generative Flow with Invertible 1x1 Convolutions DP Kingma, P Dhariwal Advances in Neural Information Processing Systems, 10215-10224	3097	2018
An Introduction to Variational Autoencoders DP Kingma, M Welling Foundations and Trends® in Machine Learning 12 (4), 307-392	2433	2019

Cited by

	All	Since 2019
Citations	241353	214654
h-index	37	36
i10-index	39	39



Public access VIEW ALL

0 articles 3 articles
not available available

Based on funding mandates

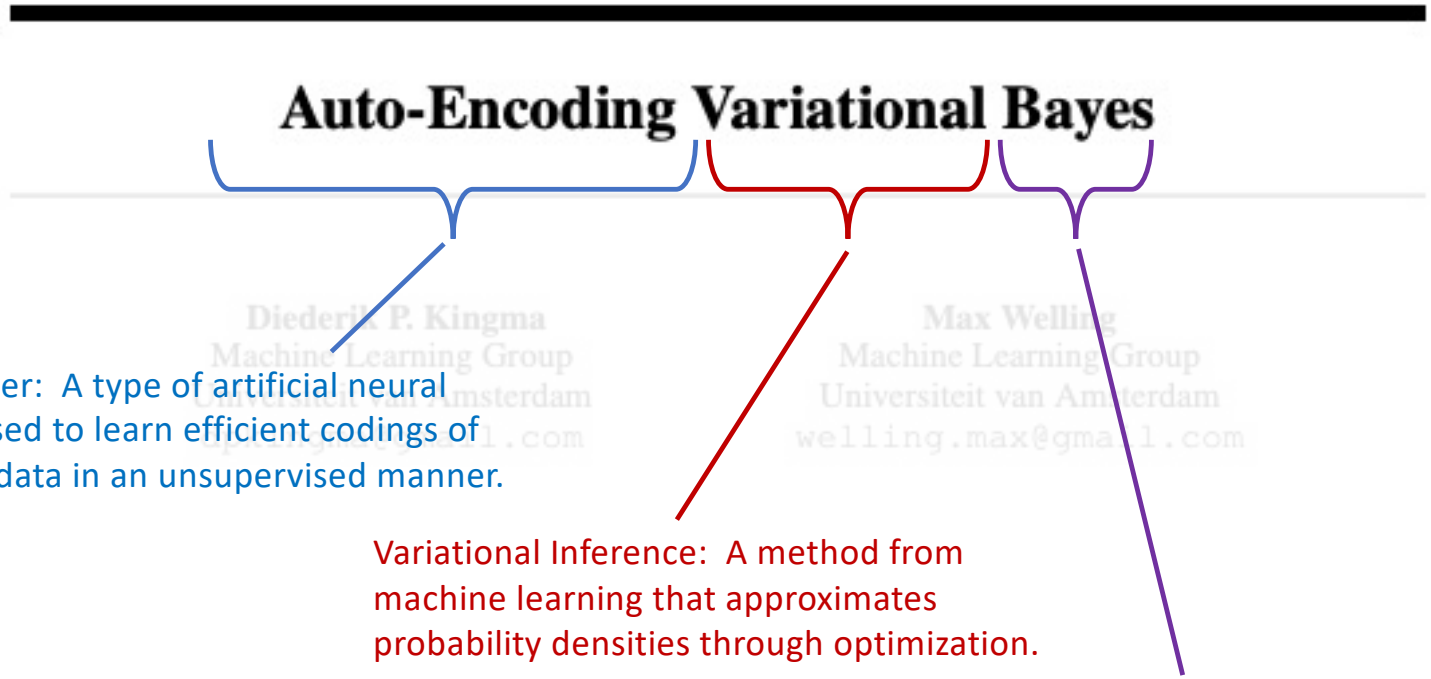
Variational Autoencoder

Variational Inference: A method from machine learning that approximates probability densities through optimization.

Autoencoder: A type of artificial neural network used to learn efficient codings of unlabeled data in an unsupervised manner.

VAE is an autoencoder whose encodings distribution is regularized during the training to ensure that its latent space has good properties allowing us to generate new data.

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019



Autoencoder: A type of artificial neural network used to learn efficient codings of unlabeled data in an unsupervised manner.

Variational Inference: A method from machine learning that approximates probability densities through optimization.

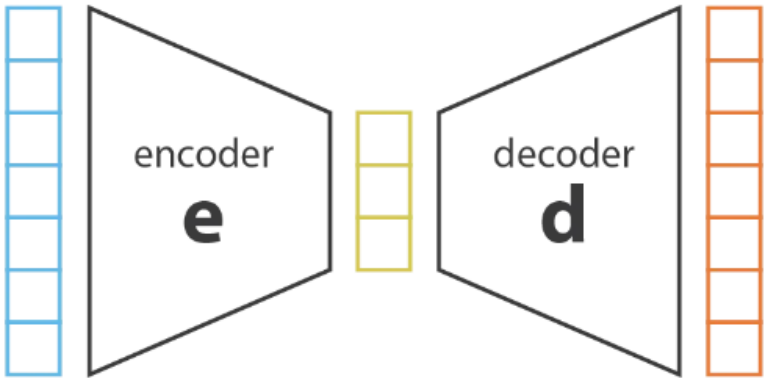
Bayesian since joint density is decomposed into prior and posterior density distributions using Bayes Rule:

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

Outline

- Autoencoder and its limitations
- Intuition behind VAEs
- Derivation of VAE
- Example applications

Dimensionality reduction with an autoencoder



x — **initial data** in space R^n

e(x) — **encoded data** in latent space R^m (with m<n)

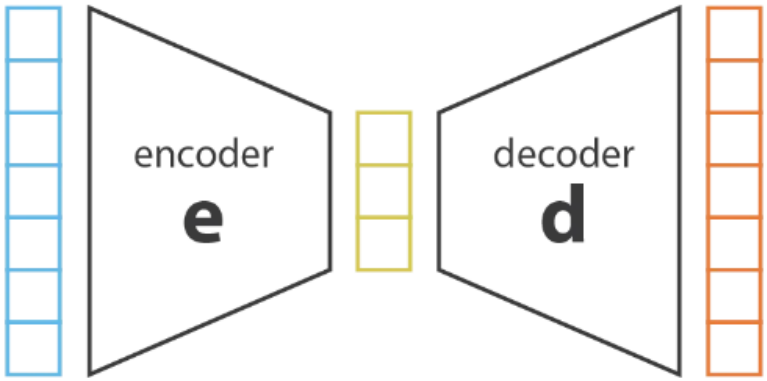
d(e(x)) — **encoded-decoded data** back in the initial space R^n

$x = d(e(x))$ → **lossless encoding**
no information is lost when reducing the number of dimensions

$x \neq d(e(x))$ → **lossy encoding**
some information is lost when reducing the number of dimensions and can't be recovered later

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Dimensionality reduction with an autoencoder



x

initial data
in space R^n

e(x)

encoded data
in latent space R^m (with m<n)

d(e(x))

encoded-decoded data
back in the initial space R^n

We want to find the best encoder, **e**, and decoder, **d**, to minimize the error between x and d(e(x)).

$$(e^*, d^*) = \underset{(e,d)\in E\times D}{\operatorname{argmin}}\ \epsilon(x, d(e(x)))$$

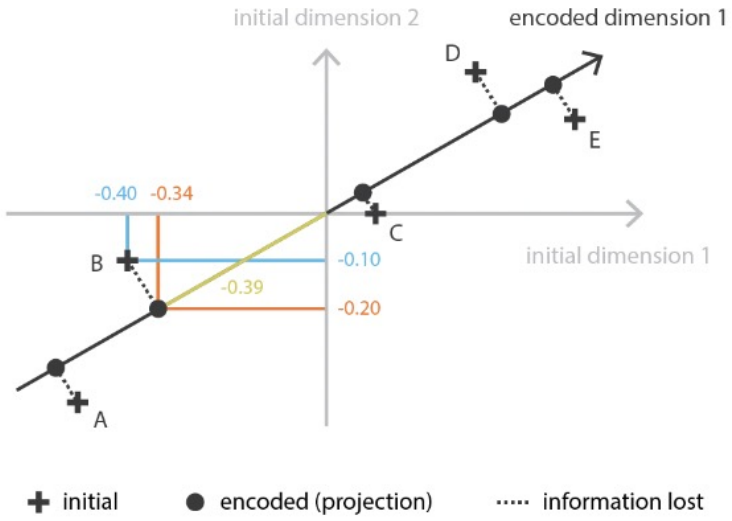
where

$$\epsilon(x, d(e(x)))$$

is the reconstruction error.

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Dimensionality reduction with Principal Component Analysis (PCA)



$n_d = 2 \;\; n_e = 1$

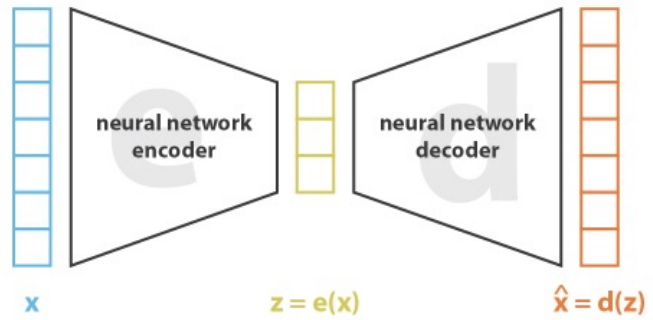
Point	Initial	Encoded	Decoded
A	(-0.50, -0.40)	-0.63	(-0.54, -0.33)
B	(-0.40, -0.10)	-0.39	(-0.34, -0.20)
C	(0.10, 0.00)	0.09	(0.07 0.04)
D	(0.30, 0.30)	0.41	(0.35, 0.21)
E	(0.50, 0.20)	0.53	(0.46, 0.27)

- Project the n_d-dimensional features onto an *orthogonal* n_e-dimensional subspace that minimizes Euclidean distance.
- PC1 points in the direction of greatest variance in the original data.

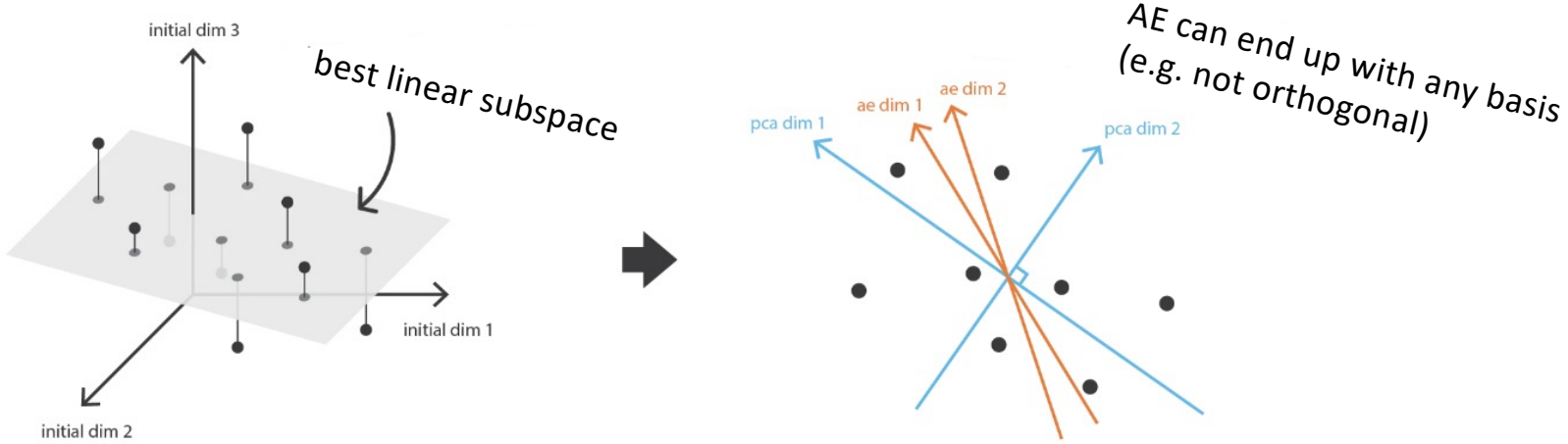
Linear Transformation!!

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Neural Network Autoencoder – 1 Linear Layer

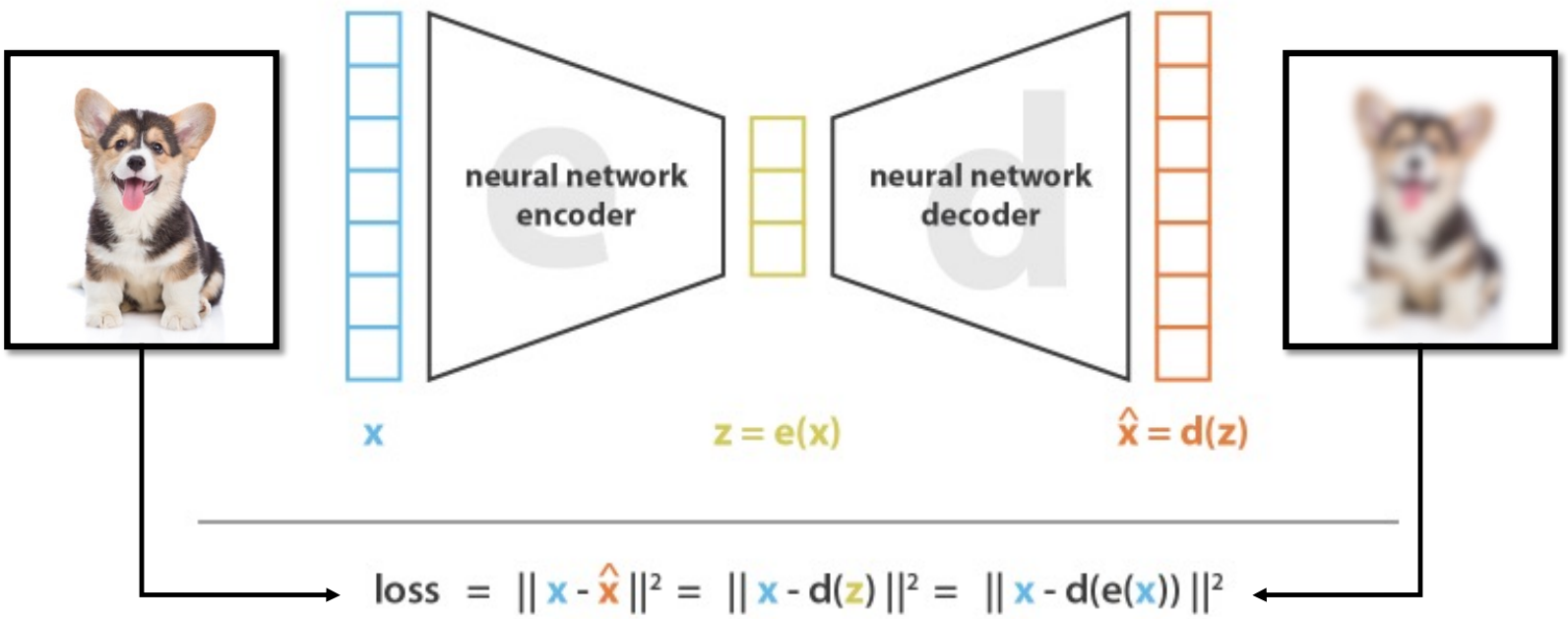


We could define encoder and decoder to each have one linear layer (no activation function), but it wouldn't necessarily converge during training to PCA solution.



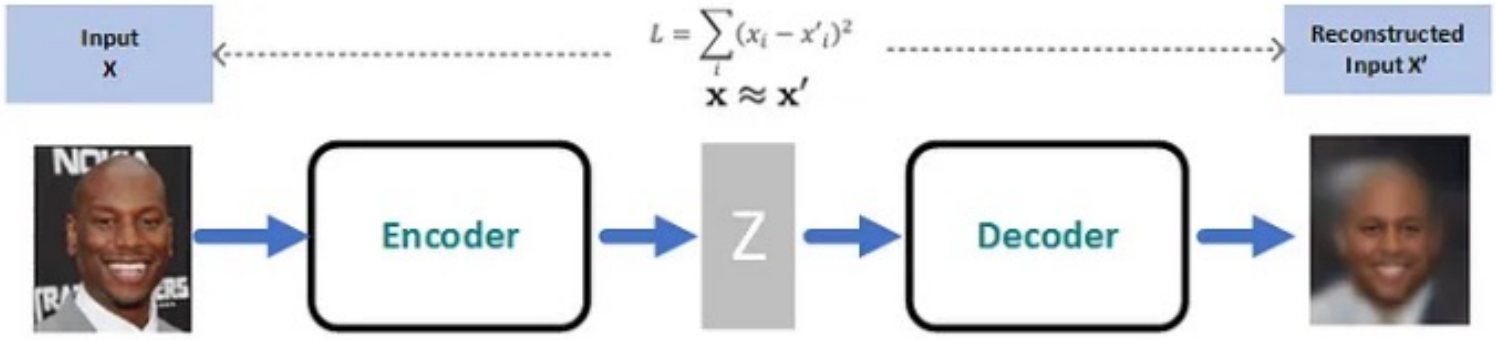
Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Neural Network Autoencoder



Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Autoencoder Reconstruction

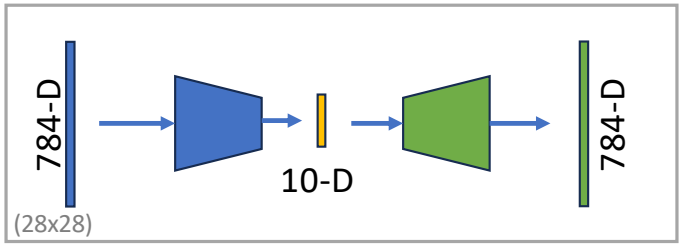


Trained on CelebA dataset.

Kana, "Variational Autoencoders (VAEs) for Dummies -- Step by Step Tutorial", 2020

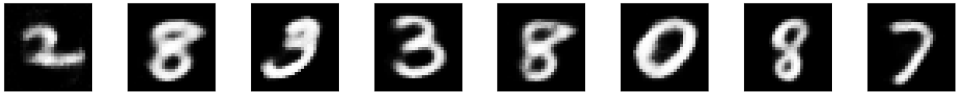
Auto Encoder Code Examples

1. Autoencoder with MNIST



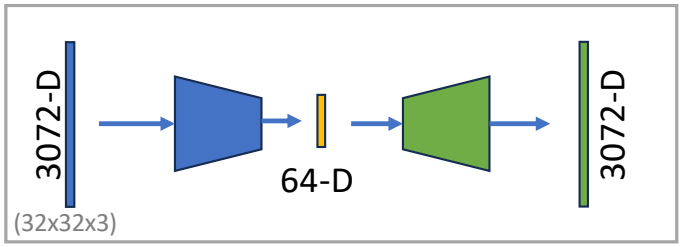
Original

Reconstructed

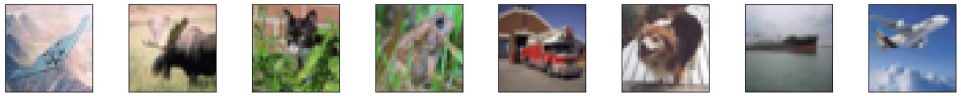


10 epochs @ 60 iterations each, training loss 0.08 → 0.02

2. Autoencoder with CIFAR10



Original

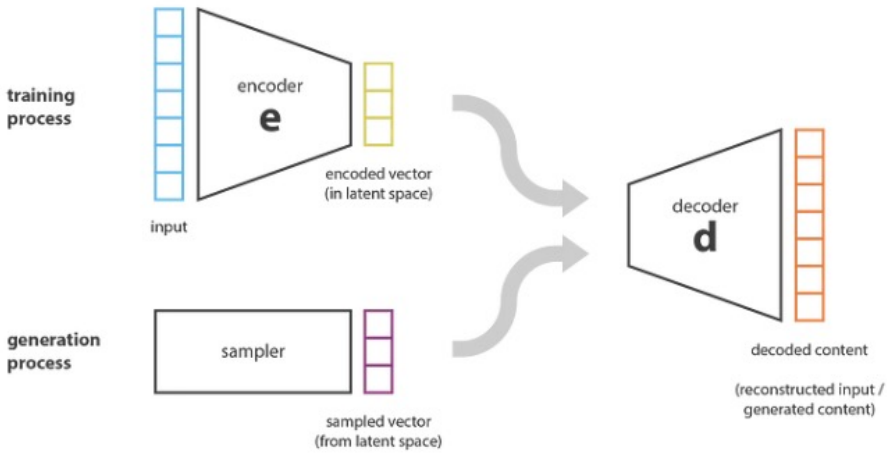


Reconstructed

50 epochs @ 50 iterations each, training loss 0.04 → 0.01

https://github.com/DL4DS/sp2026/tree/main/lecture_collateral/vae

Can we generate new samples with autoencoder?



Train encoder and decoder as autoencoder.

Randomly select a different point in the latent space.

Provide as input to the decoder to generate an output.

Will this produce a good quality output? Why?

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

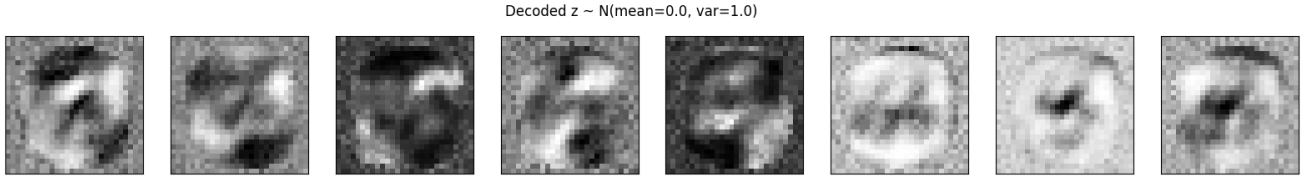
Can we generate new samples with autoencoder?

Train encoder and decoder as autoencoder.

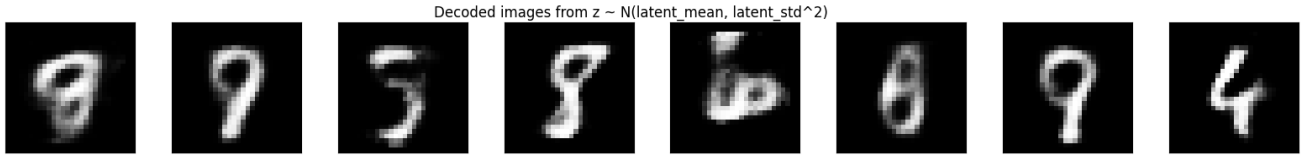
Randomly select a different point in the latent space.

Provide as input to the decoder to generate an output.

01_autoencoder_mnist.ipynb



Reconstruct from standard normal

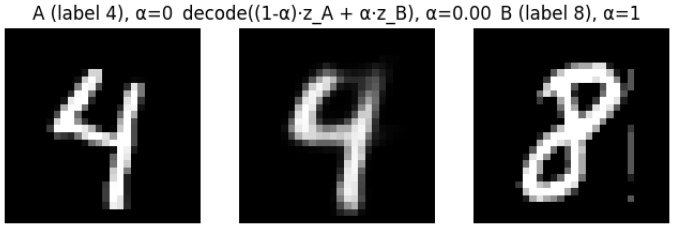


Reconstruct from gaussian with mean and variance from training data

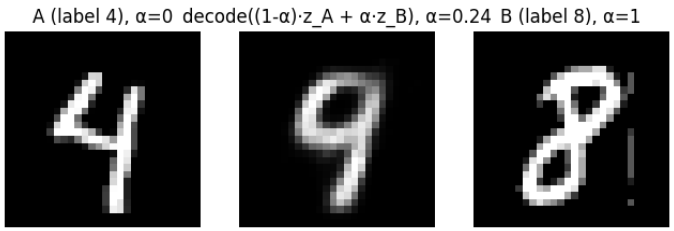
Can we generate new samples with autoencoder?

Linearly interpolate between 2 training data latent vectors.

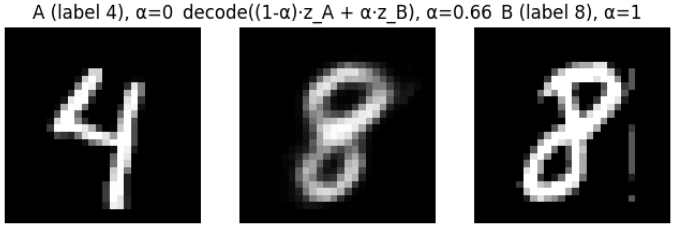
$$\text{decode}((1-\alpha)z_A + \alpha z_B)$$



$\alpha = 0$



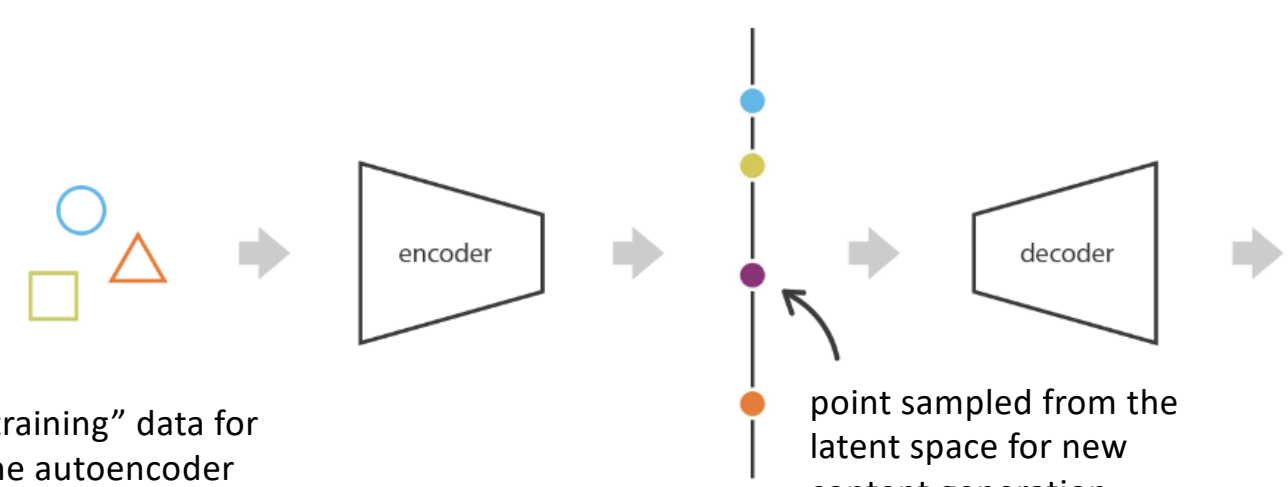
$\alpha = 0.24$



$\alpha = 0.66$

01_autoencoder_mnist.ipynb

Extreme case: Memorization



"training" data for the autoencoder

point sampled from the latent space for new content generation

Perfectly decoded samples.

without regularization the decoded output can be meaningless

Encoder and decoder are so powerful that they can fully memorize the data.

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Outline

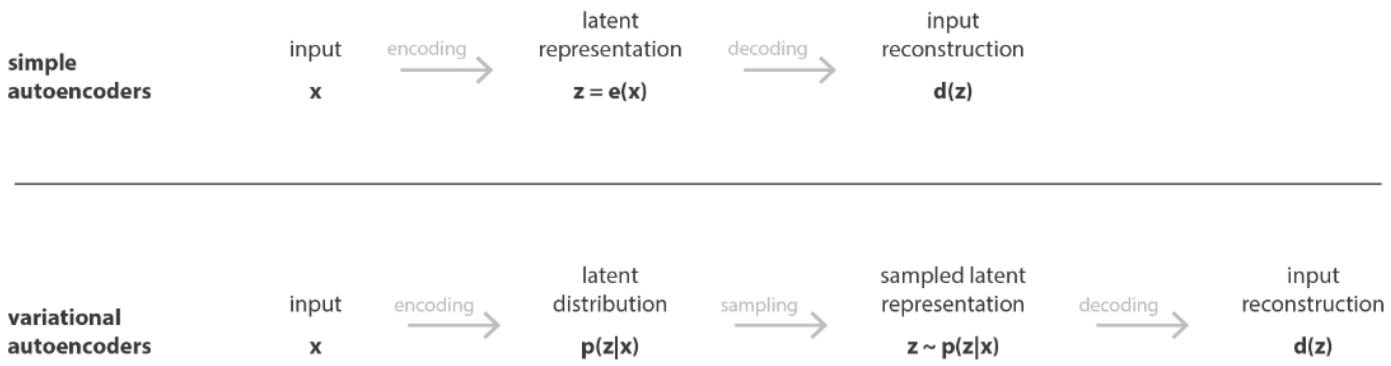
- Autoencoder and its limitations
- Intuition behind VAEs
- Derivation of VAE
- Example applications

Variational Autoencoder…

…is an autoencoder whose training is *regularized* to avoid overfitting and ensure that the *latent space has good properties* that enable generative process.

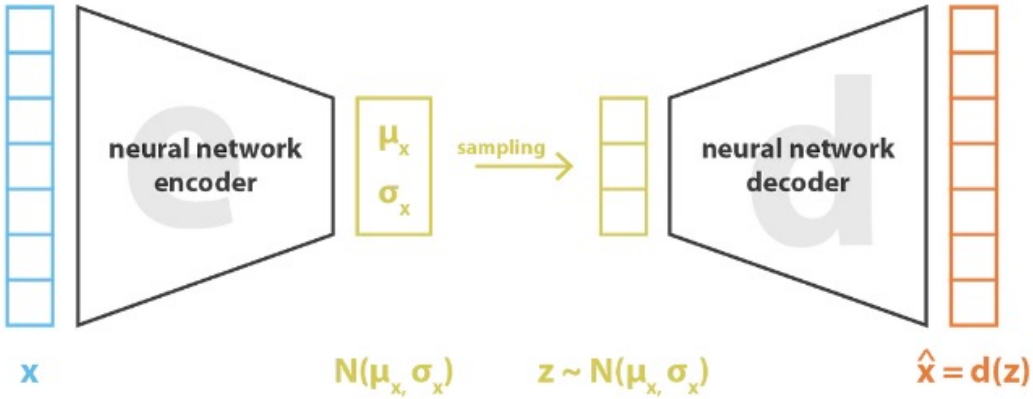
Instead of encoding as a *single point*, encode it as a *distribution* over the latent space.

Assume distributions are normal.



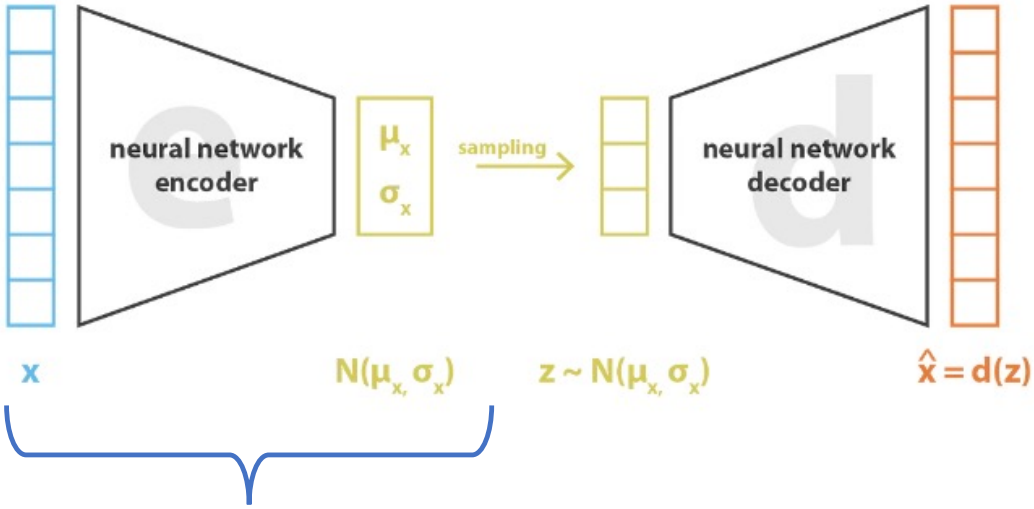
Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Variational Autoencoder



Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

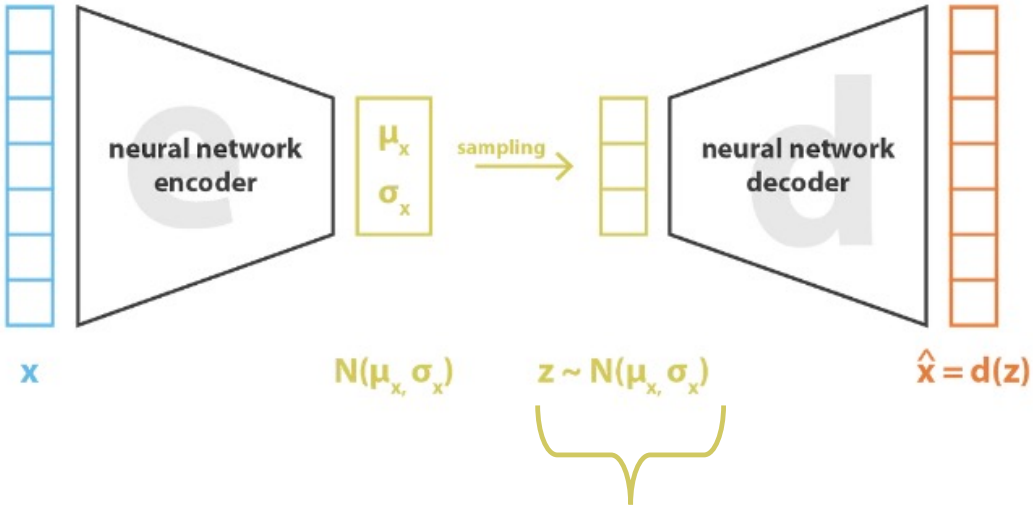
Variational Autoencoder



Encoder is emitting μ_x vector and σ_x diagonal vector for independent gaussians densities.

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

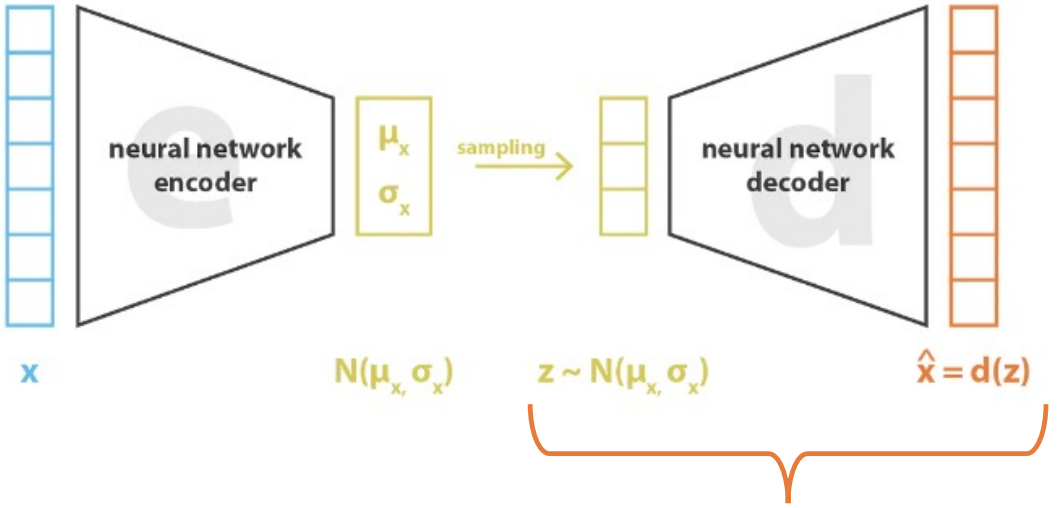
Variational Autoencoder



We then sample z from the multivariate Normal.

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

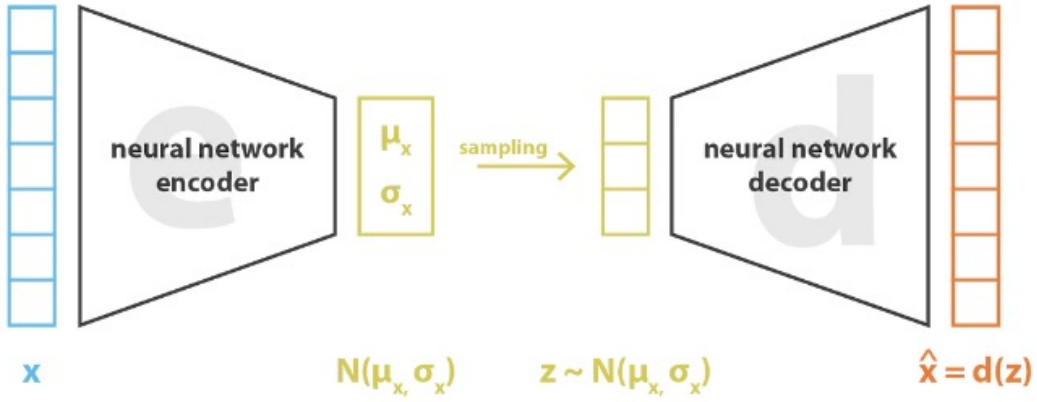
Variational Autoencoder



Then input z to the decoder network to produce output.

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Variational Autoencoder



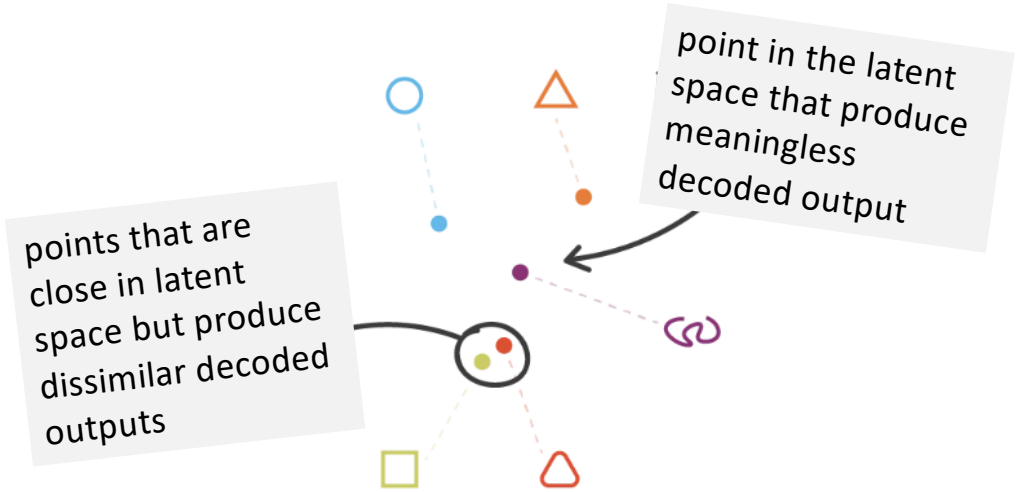
$$\text{loss} = ||x - \hat{x}||^2 + KL[N(\mu_x, \sigma_x), N(0, I)] = ||x - d(z)||^2 + KL[N(\mu_x, \sigma_x), N(0, I)]$$

L2 Loss — Kulback-Leibler divergence

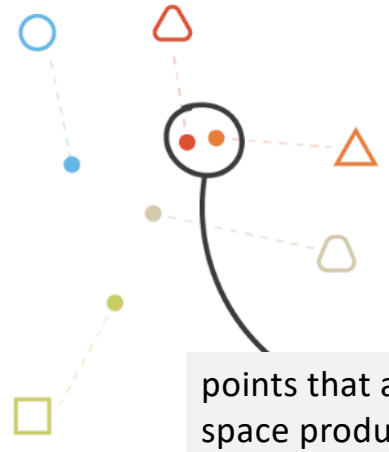
The loss is now the L2 loss as with the autoencoder, but with an additional KL-divergence term as regularizer.

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Intuitions about Regularization



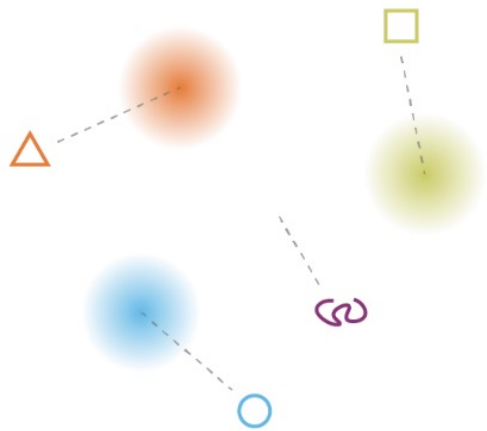
irregular latent space

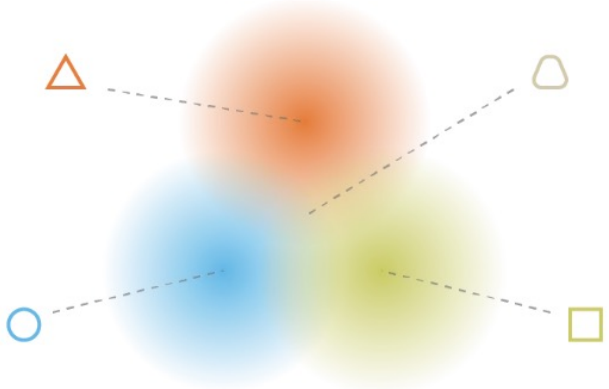


regular latent space

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Encoding to Normal distributions is not enough





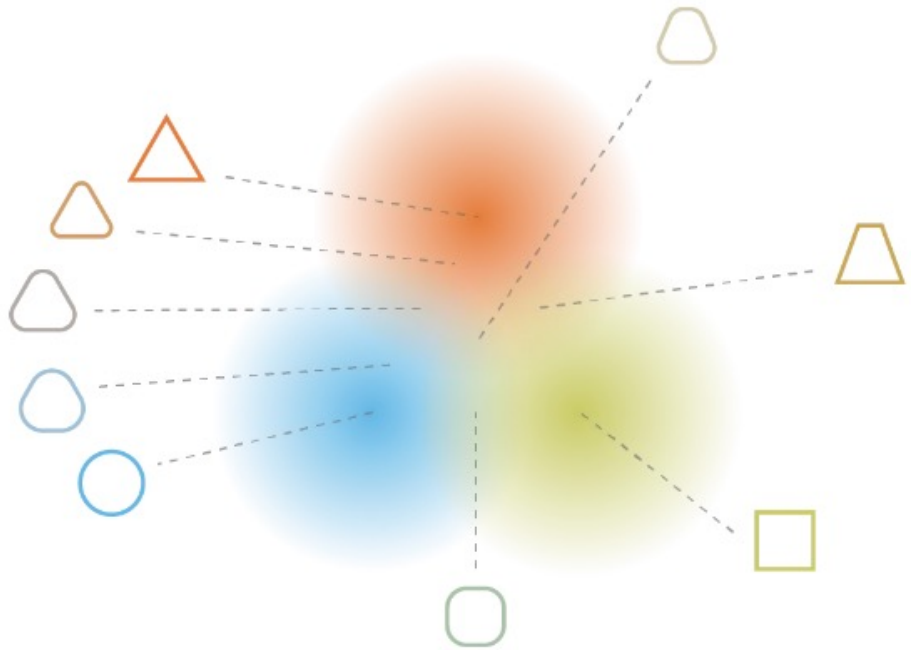
without regularization ✗ ✓ with regularization

We have to regularize the means and the covariances too!
Regularize to a standard normal.

$$\text{loss} = \|x - \hat{x}\|^2 + KL[\,N(\mu_x, \sigma_x), N(0, I)\,]$$

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Benefit of regularization



The continuity and completeness obtained from regularization tends to create a “gradient” over the information encoded in latent space.

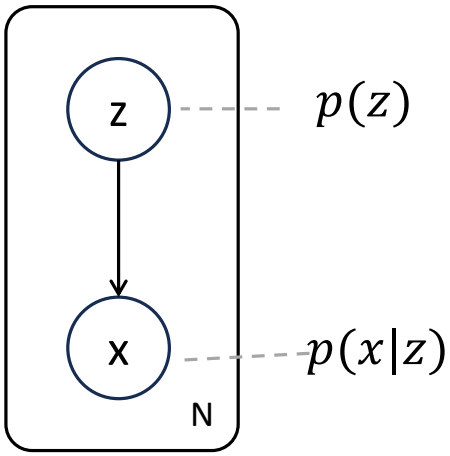
Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Dall-E 3

Outline

- Autoencoder and its limitations
- Intuition behind VAEs
- Derivation of VAE
- Example applications

Preliminaries: Bayesian Models

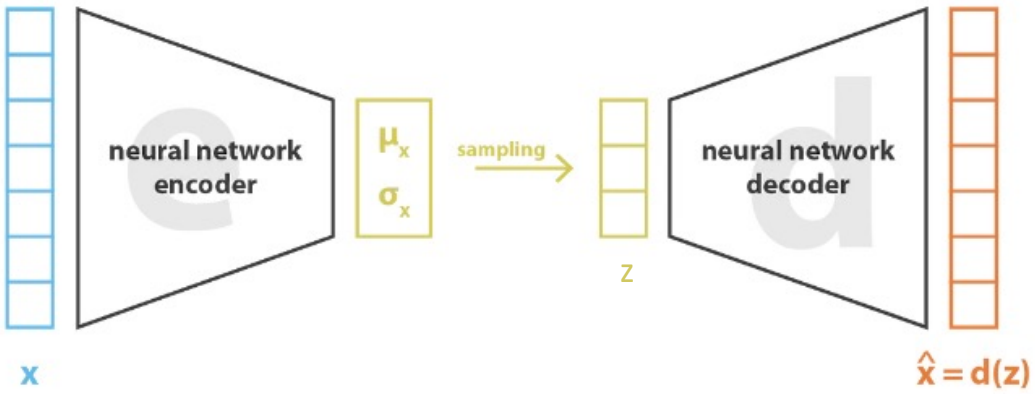


prior – prior knowledge or belief about z

likelihood – probability of a sample given z

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Bayesian Inference

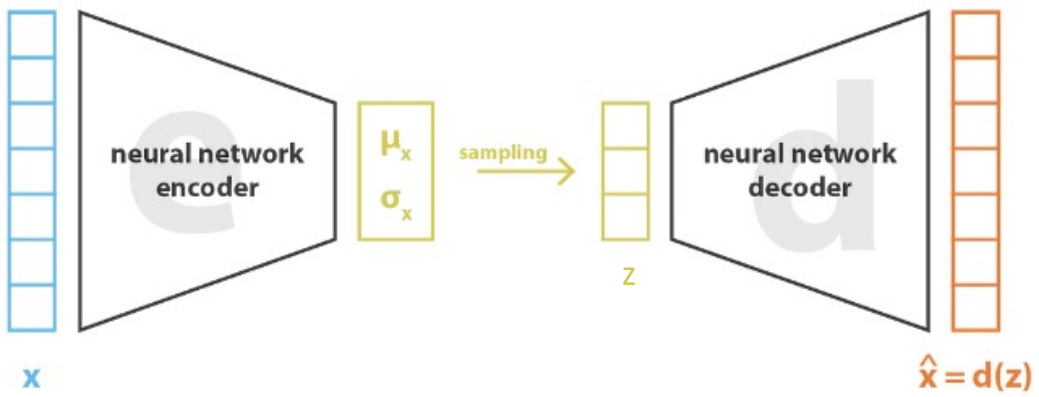


Probabilistic Encoder
$p(z|x)$

Probabilistic Decoder
$p(x|z)$

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Bayesian Inference



Probabilistic Encoder
$p(z|x)$

posterior – update our knowledge of z given a new sample

Probabilistic Decoder
$p(x|z)$

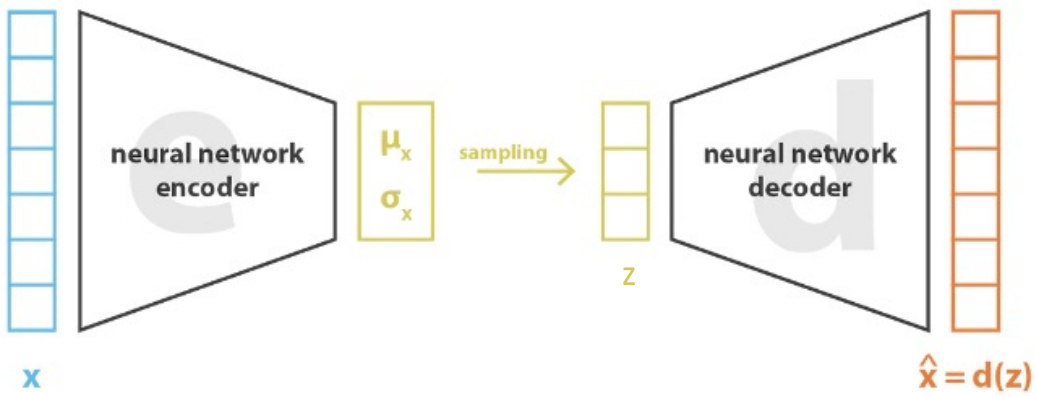
likelihood – probability of a sample given z

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

We can relate the *posterior* to the *likelihood* via **Bayes Theorem.**

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Bayesian Inference



Probabilistic Encoder

$p(z|x)$

posterior – update our knowledge of z given a new sample

Probabilistic Decoder

$p(x|z)$

likelihood – probability of a sample given z

$$\underbrace{p(z|x)}_{\text{posterior}} = \frac{\overbrace{p(x|z)}^{\text{likelihood}}\overbrace{p(z)}^{\text{prior}}}{\underbrace{p(x)}_{\text{evidence}}}$$

prior – prior knowledge or belief about z

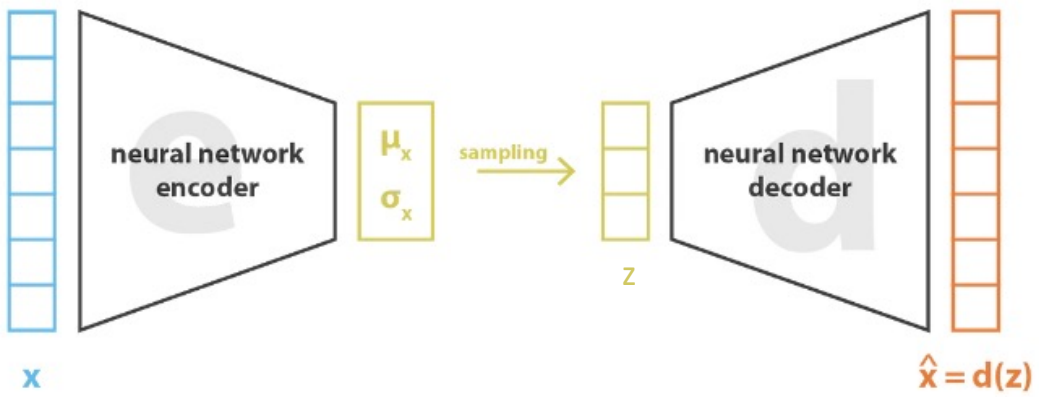
likelihood

posterior

evidence – probability distribution of our observed data

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Bayesian Inference



Probabilistic Encoder

$p(z|x)$

posterior – update our knowledge of z given a new sample

Probabilistic Decoder

$p(x|z)$

likelihood – probability of a sample given z

prior – prior knowledge or belief about z

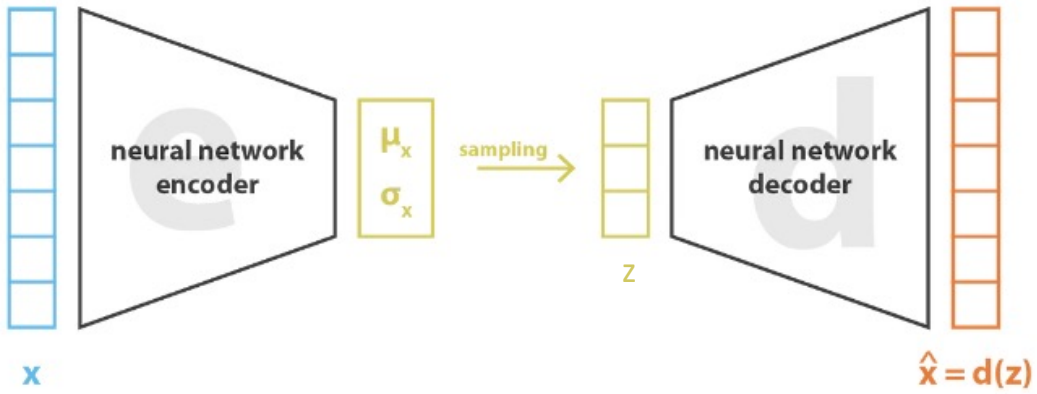
$$\underbrace{p(z|x)}_{\text{posterior}} = \frac{\overbrace{p(x|z)}^{\text{likelihood}}\overbrace{p(z)}^{\text{prior}}}{p(x)}$$

$$= \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz}$$

We can't calculate the integral directly, but we can approximate it using *variational inference*

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Simplifying Assumptions



Probabilistic Encoder

$p(z|x)$

posterior

Probabilistic Decoder

$p(x|z)$

likelihood

Assume that the *prior* is a standard Gaussian

$$p(z) \equiv \mathcal{N}(0, I)$$

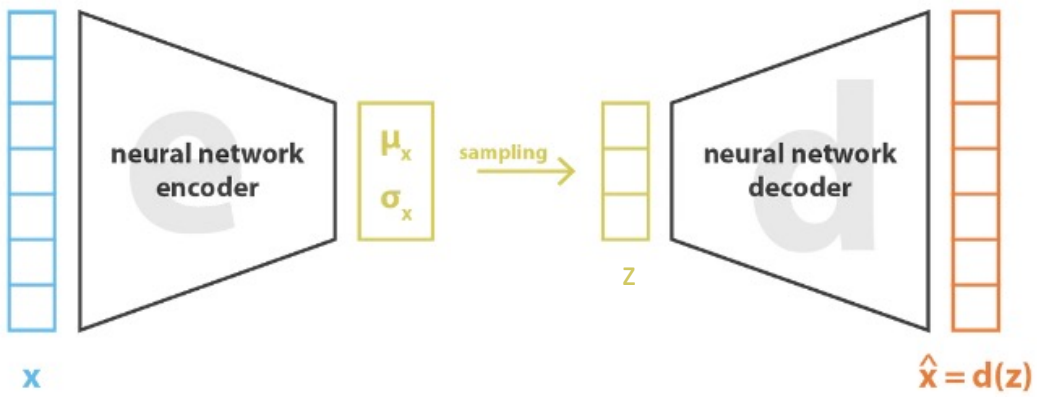
And *likelihood* is a Gaussian

$$p(x|z) \equiv \mathcal{N}(f(z), cI)$$

where $f \in F$ is a family of functions we will specify later and $c > 0$.

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Variational Inference Formulation



Probabilistic Encoder
$p(z|x)$

posterior

Probabilistic Decoder
$p(x|z)$

likelihood

We are going to approximate *posterior* to parameterized set of Gaussians.

Approximate $p(z|x)$ by a Gaussian $q_x(z)$.

$$q_x(z) \equiv \mathcal{N}(g(x), h(x))$$

where $g \in G$ and $h \in H$ are a family of functions we will define shortly.

Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

$q_x(z) \equiv \mathcal{N}(g(x), h(x))$

Variational Inference

$$(g^*, h^*) = \underset{(g,h)\in G\times H}{\arg\min}\ KL(q_x(z), p(z|x))$$

We want to find the best functions, g and h, to minimize the KL-divergence from the posterior $p(z|x)$.

C.5.1 Kullback-Leibler divergence

The most common measure of distance between probability distributions $p(x)$ and $q(x)$ is the *Kullback-Leibler* or KL divergence and is defined as:

$$D_{KL}\left[p(x)||q(x)\right] = \int p(x)\log\left[\frac{p(x)}{q(x)}\right]dx. \tag{C.28}$$

$q_x(z) \equiv \mathcal{N}(g(x), h(x))$

Variational Inference

$$(g^*, h^*) = \underset{(g,h)\in G\times H}{\arg\min}\ KL(q_x(z), p(z|x))$$

$$= \underset{(g,h)\in G\times H}{\arg\min} \left(\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}\left(\log \frac{p(x|z)p(z)}{p(x)}\right)\right)$$

- Rewriting KL divergence as Expectation,
- log of division is difference of the logs
- substituting for the posterior using Bayes Theorem

$q_x(z) \equiv \mathcal{N}(g(x), h(x))$

Variational Inference

$$
\begin{aligned}
(g^*, h^*) &= \underset{(g,h)\in G\times H}{\arg\min}\ KL(q_x(z), p(z|x)) \\
&= \underset{(g,h)\in G\times H}{\arg\min} \left(\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}\left(\log \frac{p(x|z)p(z)}{p(x)}\right)\right) \\
&= \underset{(g,h)\in G\times H}{\arg\min} \left(\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}(\log p(z)) - \mathbb{E}_{z\sim q_x}(\log p(x|z)) + \mathbb{E}_{z\sim q_x}(\log p(x))\right)
\end{aligned}
$$

- log of product becomes sum of logs
- log of division becomes difference of logs

$q_x(z) \equiv \mathcal{N}(g(x), h(x))$

Variational Inference

$$
\begin{aligned}
(g^*, h^*) &= \underset{(g,h)\in G\times H}{\arg\min}\ KL(q_x(z), p(z|x)) \\
&= \underset{(g,h)\in G\times H}{\arg\min} \left(\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}\left(\log \frac{p(x|z)p(z)}{p(x)}\right)\right) \\
&= \underset{(g,h)\in G\times H}{\arg\min}\ (\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}(\log p(z)) - \mathbb{E}_{z\sim q_x}(\log p(x|z)) + \mathbb{E}_{z\sim q_x}(\log p(x))) \\
&= \underset{(g,h)\in G\times H}{\arg\max}\ (\mathbb{E}_{z\sim q_x}(\log p(x|z)) - KL(q_x(z), p(z)))
\end{aligned}
$$

- negating and converting from argmin to argmax
- collecting terms to form KL divergence

$q_x(z) \equiv \mathcal{N}(g(x), h(x))$

Variational Inference

$$
\begin{aligned}
(g^*, h^*) &= \underset{(g,h)\in G\times H}{\arg\min}\ KL(q_x(z), p(z|x)) \\
&= \underset{(g,h)\in G\times H}{\arg\min}\ \left(\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}\left(\log \frac{p(x|z)p(z)}{p(x)}\right)\right) \\
&= \underset{(g,h)\in G\times H}{\arg\min}\ (\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}(\log p(z)) - \mathbb{E}_{z\sim q_x}(\log p(x|z)) + \mathbb{E}_{z\sim q_x}(\log p(x))) \\
&= \underset{(g,h)\in G\times H}{\arg\max}\ (\underbrace{\mathbb{E}_{z\sim q_x}(\log p(x|z))}_{\text{Maximize the expected log likelihood.}} - \underbrace{KL(q_x(z), p(z))}_{\text{Minimize the difference between the approximate posterior and the prior.}})
\end{aligned}
$$

$q_x(z) \equiv \mathcal{N}(g(x), h(x))$

Variational Inference

$$
\begin{aligned}
(g^*, h^*) &= \underset{(g,h)\in G\times H}{\arg\min}\ KL(q_x(z), p(z|x)) \\
&= \underset{(g,h)\in G\times H}{\arg\min}\left(\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}\left(\log \frac{p(x|z)p(z)}{p(x)}\right)\right) \\
&= \underset{(g,h)\in G\times H}{\arg\min}\left(\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}(\log p(z)) - \mathbb{E}_{z\sim q_x}(\log p(x|z)) + \mathbb{E}_{z\sim q_x}(\log p(x))\right) \\
&= \underset{(g,h)\in G\times H}{\arg\max}\left(\mathbb{E}_{z\sim q_x}(\log p(x|z)) - KL(q_x(z), p(z))\right) \\
&= \underset{(g,h)\in G\times H}{\arg\max}\left(\underbrace{\mathbb{E}_{z\sim q_x}\left(-\frac{||x - f(z)||^2}{2c}\right)} - KL(q_x(z), p(z))\right)
\end{aligned}
$$

Log of the Gaussian likelihood $p(x|z) \equiv \mathcal{N}(f(z), cI)$.

This brings our function, f, into the equation, so…

$q_x(z) \equiv \mathcal{N}(g(x), h(x))$

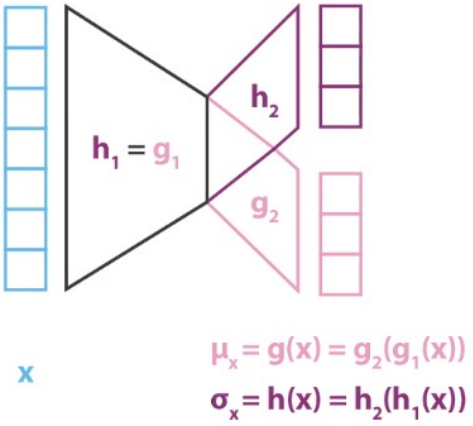
Variational Inference

We are looking for optimal f^*, g^* and h^* such that

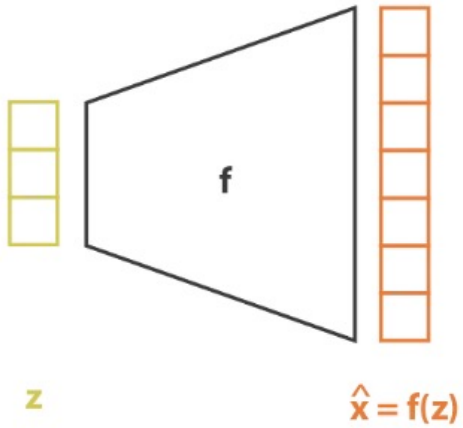
$$(f^*, g^*, h^*) = \underset{(f,g,h) \in F \times G \times H}{\arg\max} \left(\mathbb{E}_{z \sim q_x} \left(-\frac{||x - f(z)||^2}{2c} \right) - KL(q_x(z), p(z)) \right)$$

Note that the constant, c, determines the balance between reconstruction error and the regularization term given by KL divergence.

Enter the Neural Networks

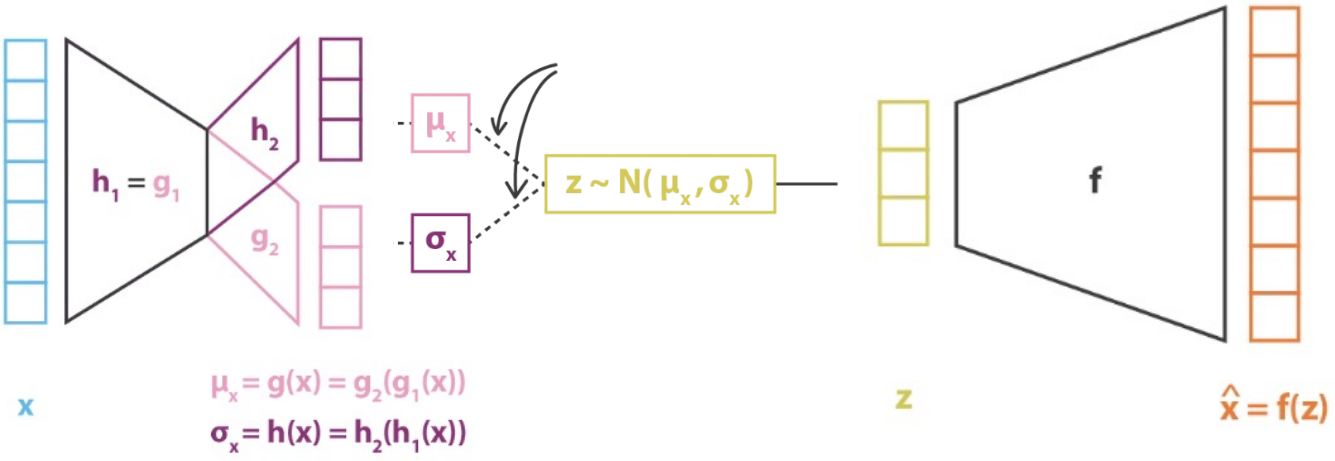


Encoder produces the mean and variance.



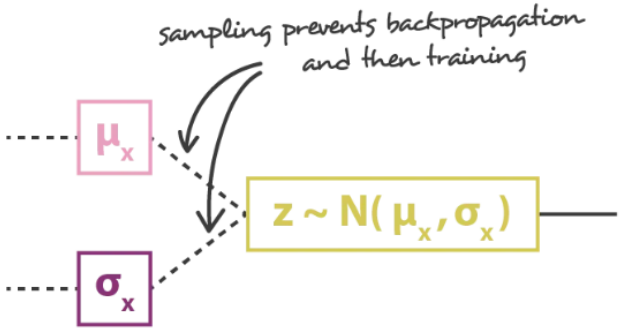
Decoder reconstructs the input (during training)

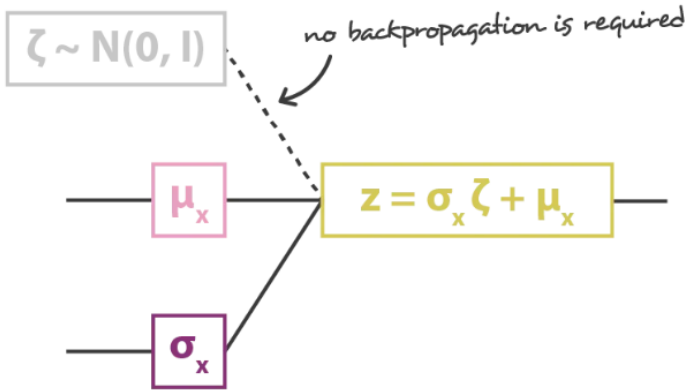
But one more problem to solve



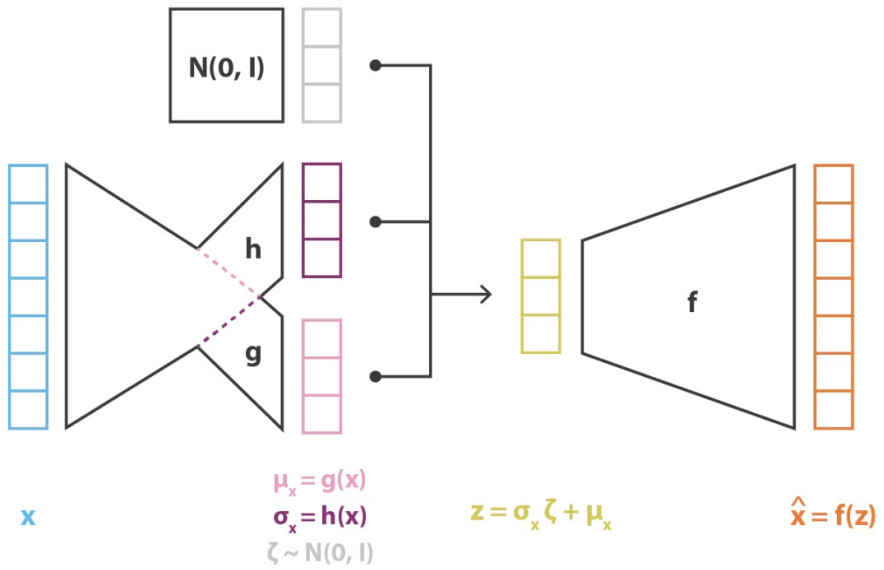
We can't backpropagate through the sampling step.

Use the reparameterization trick





Putting it all together



$$\text{loss} = C \,||\, x - \hat{x} \,||^2 + KL[\, N(\mu_x, \sigma_x), N(0, I) \,] = C \,||\, x - f(z) \,||^2 + KL[\, N(g(x), h(x)), N(0, I) \,]$$

We have as trainable neural network!

We use a Monte-Carlo approximation to the expectation of reconstruction loss

Convert C = 1/(2c).

Probability Distribution Divergence Measures

C.5.1 Kullback-Leibler divergence

The most common measure of distance between probability distributions $p(x)$ and $q(x)$ is the *Kullback-Leibler* or KL divergence and is defined as:

$$D_{KL}\big[p(x)||q(x)\big] = \int p(x)\log\left[\frac{p(x)}{q(x)}\right]dx. \tag{C.28}$$

C.5.2 Jensen-Shannon divergence

The KL divergence is not symmetric (i.e., $D_{KL}[p(x)||q(x)] \neq D_{KL}[q(x)||p(x)]$). The Jensen-Shannon divergence is a measure of distance that is symmetric by construction:

$$D_{JS}\Big[p(x)\Big|\Big|q(x)\Big] = \frac{1}{2}D_{KL}\left[p(x)\middle|\middle|\frac{p(x)+q(x)}{2}\right] + \frac{1}{2}D_{KL}\left[q(x)\middle|\middle|\frac{p(x)+q(x)}{2}\right]. \tag{C.30}$$

It is the mean divergence of $p(x)$ and $q(x)$ to the average of the two distributions.

Dall-E 3

VAE Code Examples

1. VAE with random data  

2. VAE with MNIST  

3. VAE with CIFAR10  

https://github.com/DL4DS/sp2026/tree/main/lecture_collateral/vae

Outline

- Autoencoder and its limitations
- Intuition behind VAEs
- Derivation of VAE
- Example applications

Generating high quality images

Vahdat & Kautz (2020) “NVAE: A deep hierarchical variational autoencoder”

Resynthesizing real data with changes

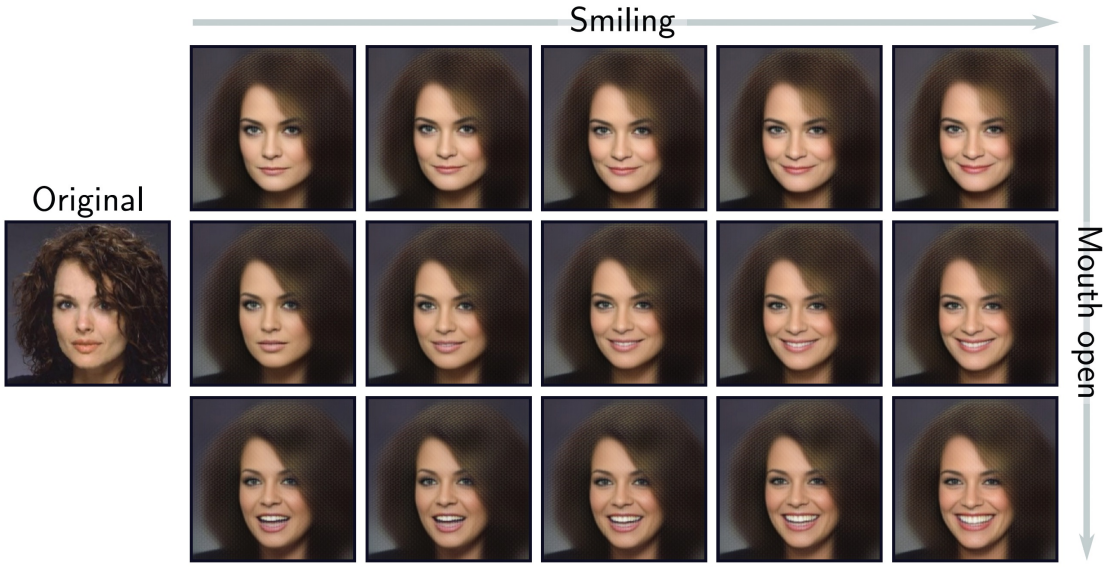
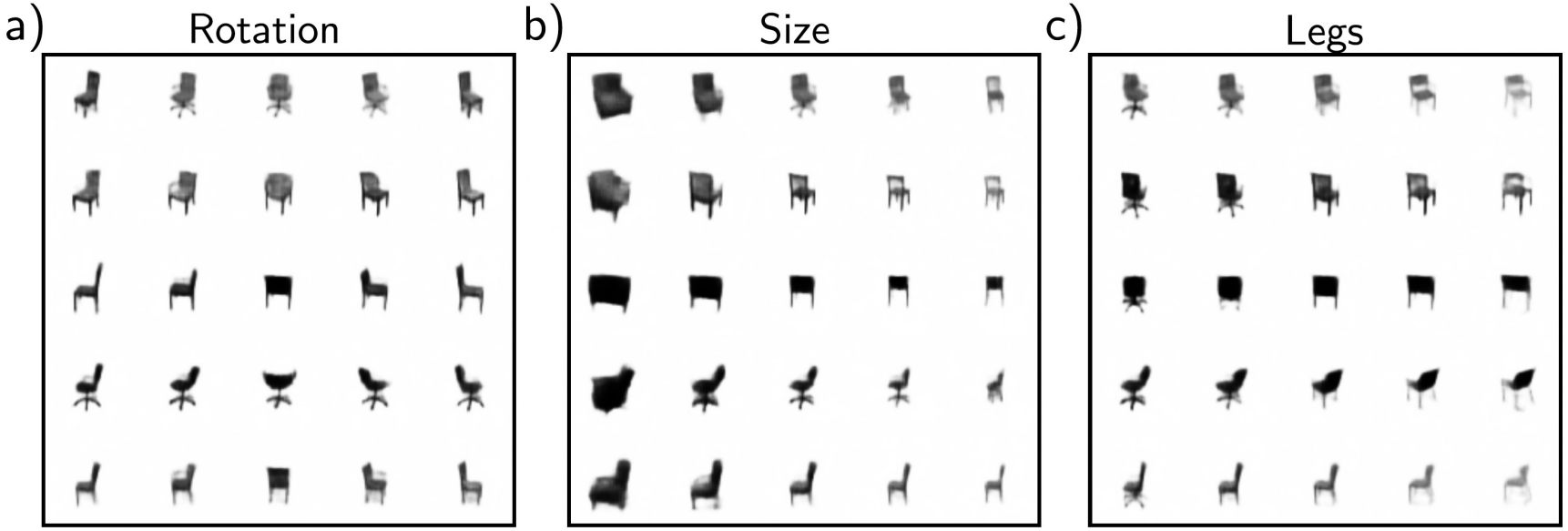
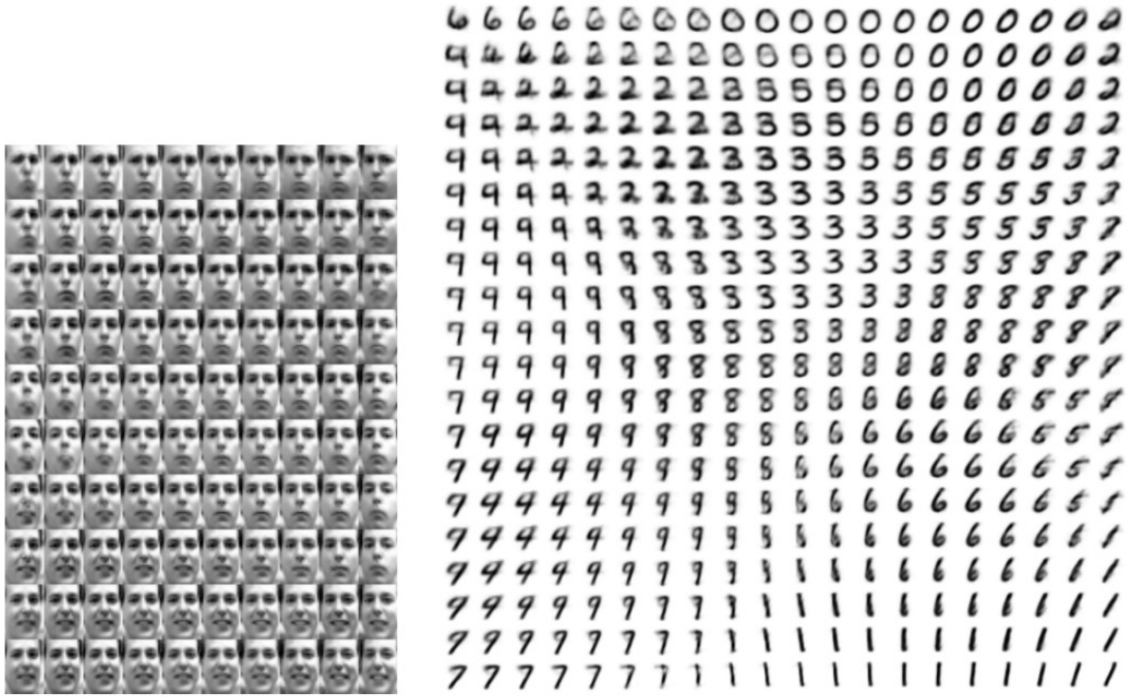


Figure 17.13 Resynthesis. The original image on the left is projected into the latent space using the encoder, and the mean of the predicted Gaussian is chosen to represent the image. The center-left image in the grid is the reconstruction of the input. The other images are reconstructions after manipulating the latent space in directions representing smiling/neutral (horizontal) and mouth open/closed (vertical). Adapted from White (2016).

Disentanglement of the latent space



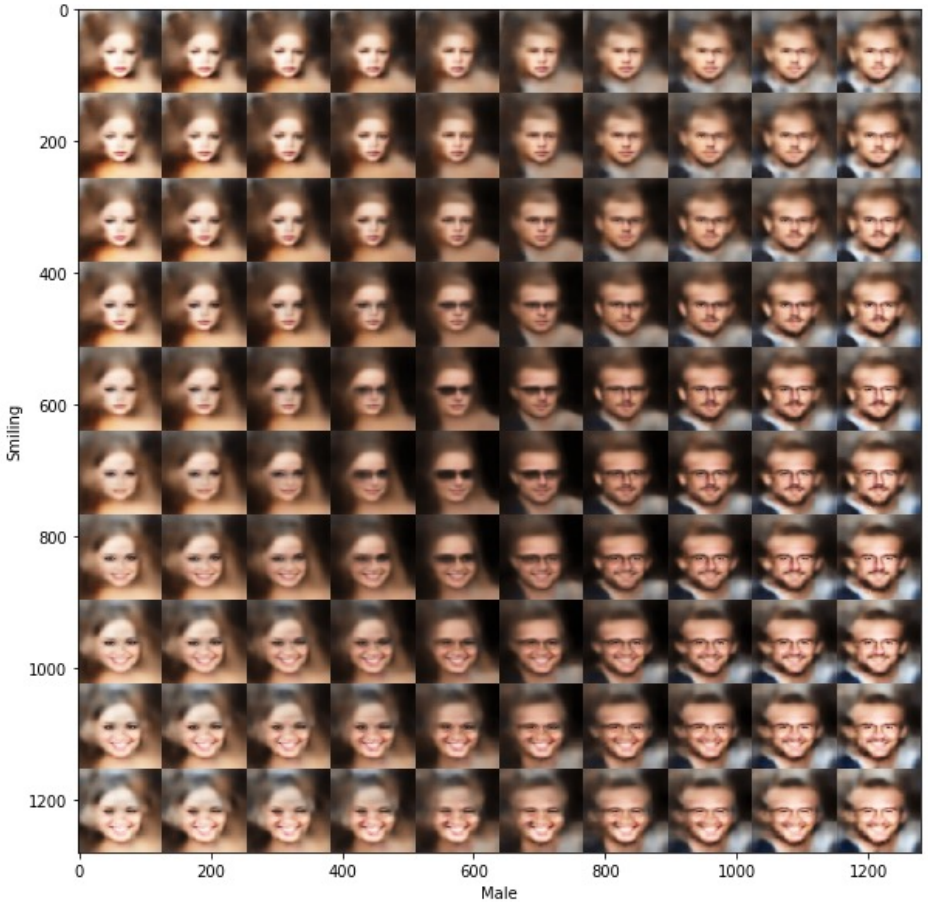
Chen et al (2021) “Cross-layer distillation with semantic calibration.”



(a) Learned Frey Face manifold (b) Learned MNIST manifold

Figure 4: Visualisations of learned data manifold for generative models with two-dimensional latent space, learned with AEVB. Since the prior of the latent space is Gaussian, linearly spaced coordinates on the unit square were transformed through the inverse CDF of the Gaussian to produce values of the latent variables $\mathbf{z}$. For each of these values $\mathbf{z}$, we plotted the corresponding generative $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ with the learned parameters $\boldsymbol{\theta}$.

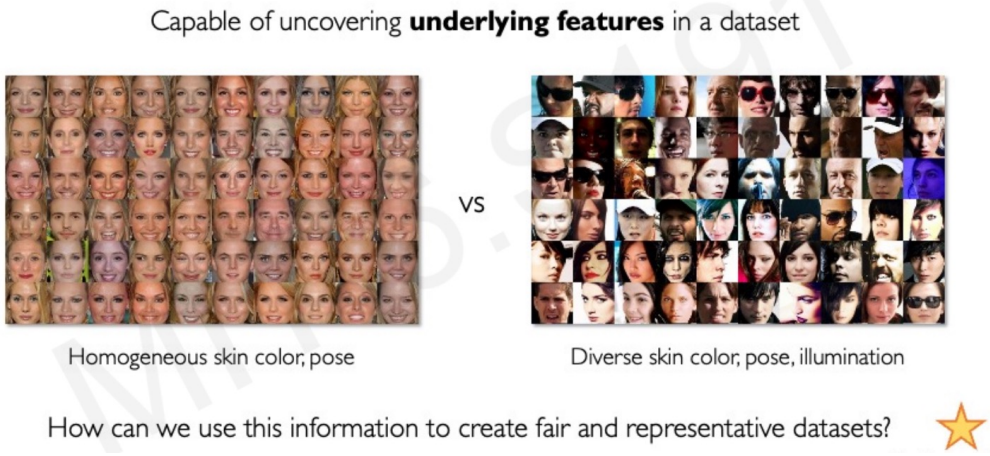
Conditional VAEs



Example from https://medium.com/data-science/variational-autoencoders-vaes-for-dummies-step-by-step-tutorial-69e6d1c9d8e9

Debiasing

Fuses the original learning task with a variational autoencoder to learn the latent structure within the dataset and then adaptively uses the learned latent distributions to re-weight the importance of certain data points while training.



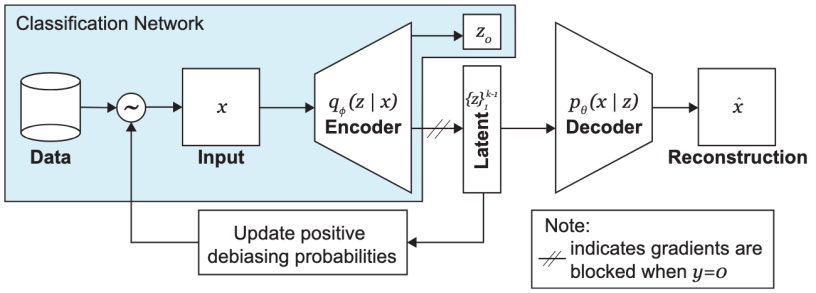
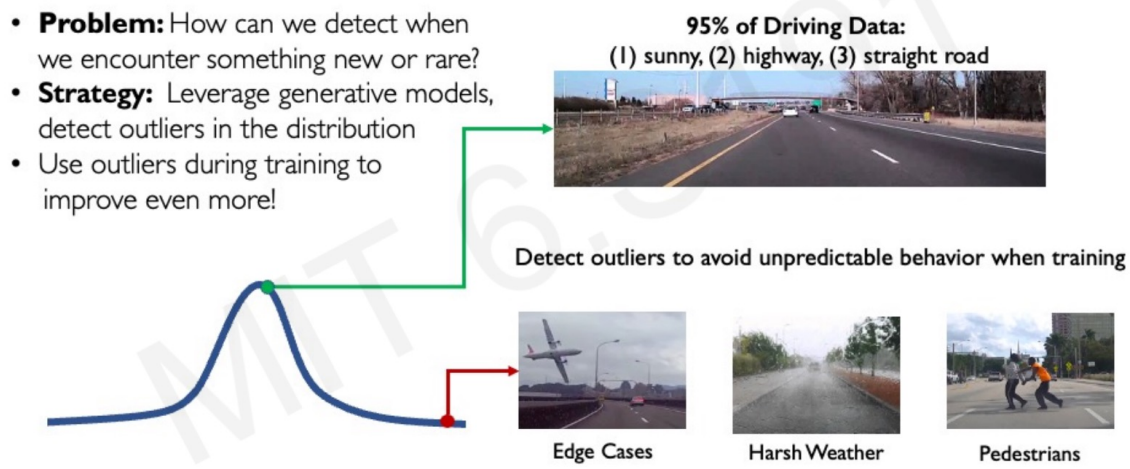


Figure 2: Debiasing Variational Autoencoder. Architecture of the semi-supervised DB-VAE for binary classification (blue region). The unsupervised latent variables are used to adaptively resample the dataset while training.

Amini et al, "Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure," 2019

Outlier Detection

We propose and evaluate a novel architecture for self-supervised learning of latent variables to detect the insufficiently trained situations. Our method also addresses training data imbalance, by learning a set of underlying latent variables that characterize the training data and evaluate potential biases.



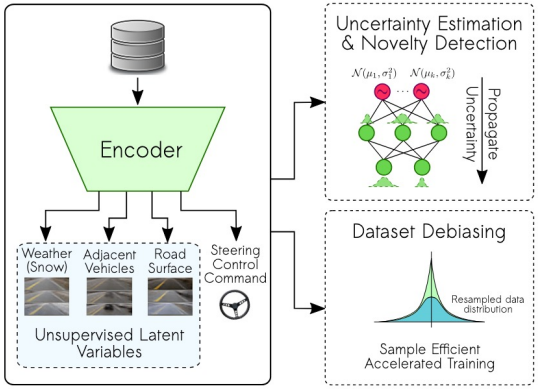
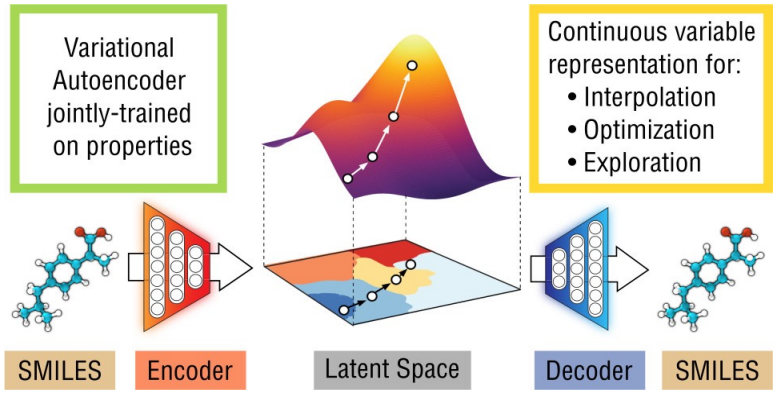


Fig. 1: **Semi-supervised end-to-end control.** An encoder neural network is trained to learn a *supervised* control command as well as various other *unsupervised* outputs that qualitatively describe the image. This enables two key contributions of novelty detection and dataset debiasing.

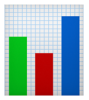
A. Amini et al, "Variational Autoencoder for End-to-End Control of Autonomous Driving with Novelty Detection and Training De-biasing," *2018 (link)*

VAE Applications: 🧬Bioinformatics

- **Drug Discovery & Molecular Generation**
  - Learn latent representations of molecular structures; generate novel drug candidates
  - *Gómez-Bombarelli et al., ACS Central Science, 2018 (link)*
- **Single-Cell RNA Sequencing Analysis**
  - scVI: model gene expression, handle batch effects, impute missing values
  - *Lopez et al., Nature Methods, 2018 (link)*
- **Protein Structure & Sequence Generation**
  - Generate plausible protein sequences; predict function from learned latent space
  - *Greener et al., Scientific Reports, 2018 (link, GitHub)*



Gómez-Bombarelli et al., ACS Central Science, 2018

VAE Applications: Social Sciences 

- **Text Generation & Topic Modeling**
  - Discover latent topics in survey responses, interview transcripts, or social media data
  - *Srivastava & Sutton, ICLR, 2017 (ProdLDA) (link)*
- **Missing Data Imputation**
  - Learn to fill in incomplete survey or census data by modeling the joint distribution
  - *Nazábal et al., Pattern Recognition, 2020 (HI-VAE) (link)*
- **Synthetic Data for Privacy**
  - Generate realistic but anonymized datasets that preserve statistical properties of sensitive health/social data
  - *Pfitzner & Arnrich, arXiv, 2022 (DPD-fVAE) (link)*

VAE Applications: Business 💼

- **Anomaly & Fraud Detection**
  - Credit card fraud, cybersecurity intrusion detection, manufacturing defect identification
  - *An & Cho, SNU Tech Report, 2015 (link)*
- **Recommendation Systems**
  - Collaborative filtering via VAEs to model user preferences in latent space
  - *Liang et al., WWW, 2018 (link)*
- **Customer Segmentation**
  - Use the learned latent space to discover meaningful customer clusters from behavioral data
  - *Jiang et al., IJCAI, 2017 (VaDE) (link)*

Upcoming Topics

- Diffusion Models
- Graph Neural Networks

Feedback

https://forms.gle/pXHM5nx1Ti9aFmpw6