Variational Autoencoders (VAEs)

DL4DS – Spring 2024

DS598 B1 Gardos

Prince, *Understanding Deep Learning,*

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

Other Content Cited

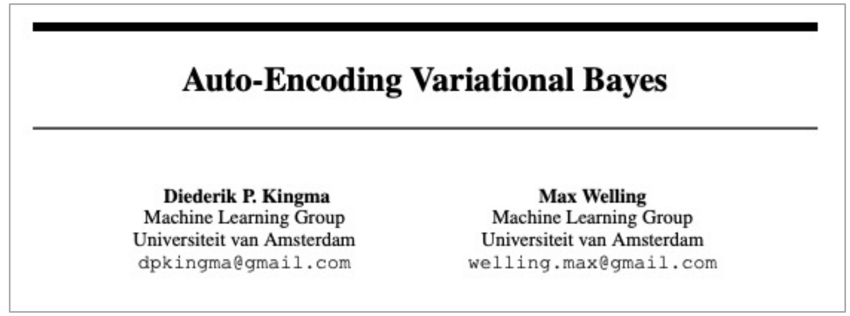
April Dates

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
	April 1	2	3	4 **GANs**	5	6
7	8	9 **VAEs**	10 **Discussion**	11 **Diffusion Models**	12	13
14	15	16 **Graph Neural Nets (VizWiz Leaders Share)**	17 **Discussion**	18 **Reinforcement Learning**	19	20
21	22	23 **TBD/Overflow (JEPA Models)**	24 **Discussion**	25 **★ Project Presentations 1 ★**	26	27
28	29	30 **★ Project Presentations 2 ★**	May 1 **Discussion??**	2 Study Period	3 Study Period	4
5	6 Final Exams	7	8 Final report & Repo **	9	10	11

** Might be earlier. Depends on when grades are due.

Diederik P. Kingma

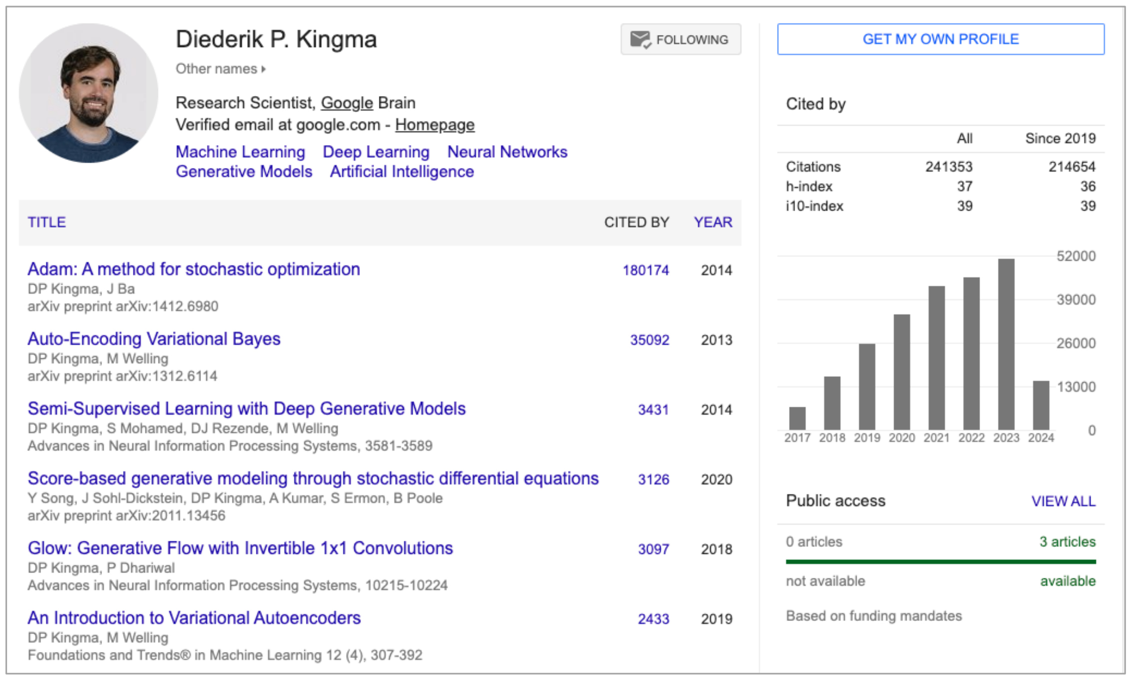
2016 OpenAI, founding member
2017 PhD U. of Amsterdam
2018– Google DeepMind



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
Other names
Research Scientist, Google Brain
Verified email at google.com - Homepage
Machine Learning Deep Learning Neural Networks Generative Models Artificial Intelligence

FOLLOWING

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

GET MY OWN PROFILE

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

$\underbrace{\text{Variational}}$ $\underbrace{\text{Autoencoder}}$

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

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

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.

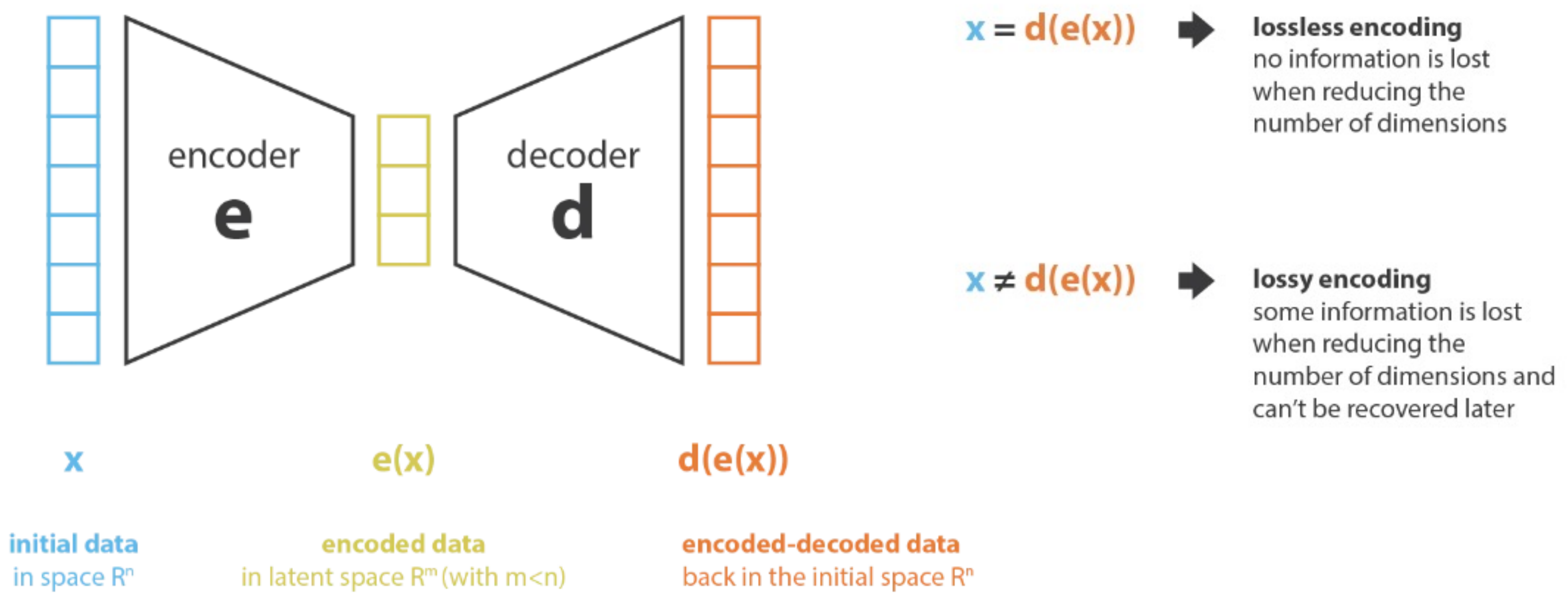
Bayesian since joint density is decomposed into prior and posterior density distributions using Bayes Rule:

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

Outline

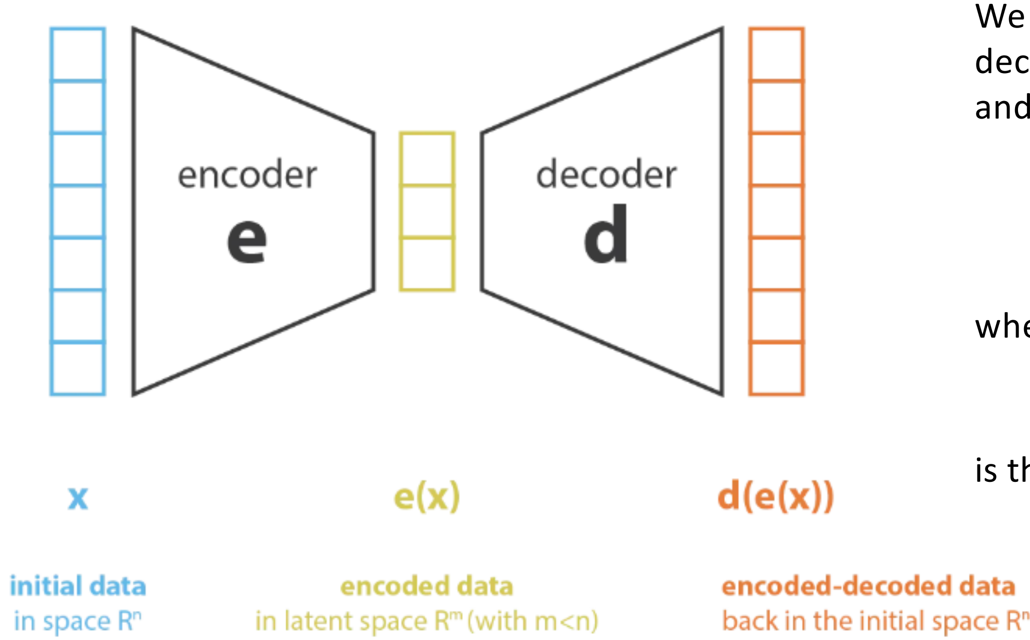
- Autoencoder and its limitations
- Intuition behind VAEs
- Derivation of VAE
- Example applications

Dimensionality reduction with an autoencoder



[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

Dimensionality reduction with an autoencoder



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

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

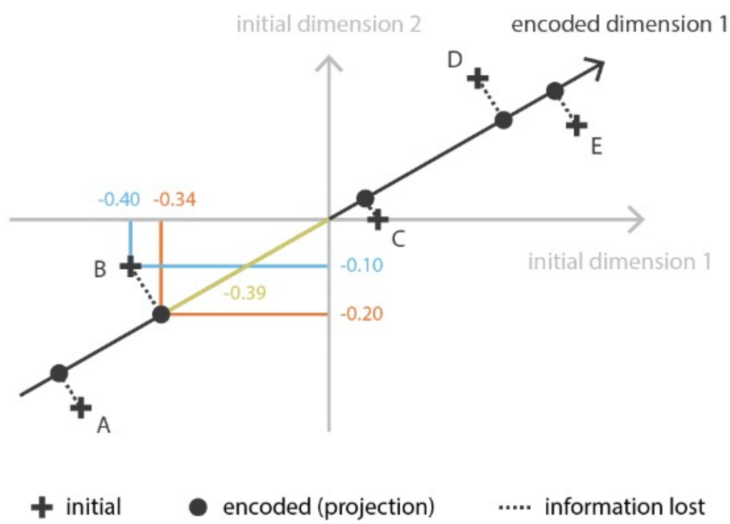
where

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

is the reconstruction error.

[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

Dimensionality reduction with Principal Component Analysis (PCA)



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

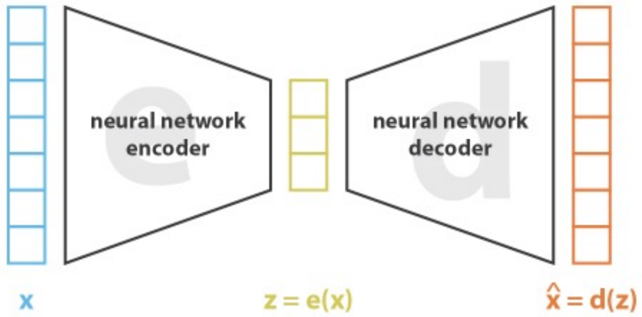
Point	Initial	Encoded	Decoded
A	(-0.50, -0.40)	-0.63	(-0.54, -0.33)
B	(-0.40, -0.10)	-0.39	(-0.34, -0.20)
C	(0.10, 0.00)	0.09	(0.07 0.04)
D	(0.30, 0.30)	0.41	(0.35, 0.21)
E	(0.50, 0.20)	0.53	(0.46, 0.27)

Project the n_d-dimensional features onto an orthogonal n_e-dimensional subspace that minimizes Euclidean distance.

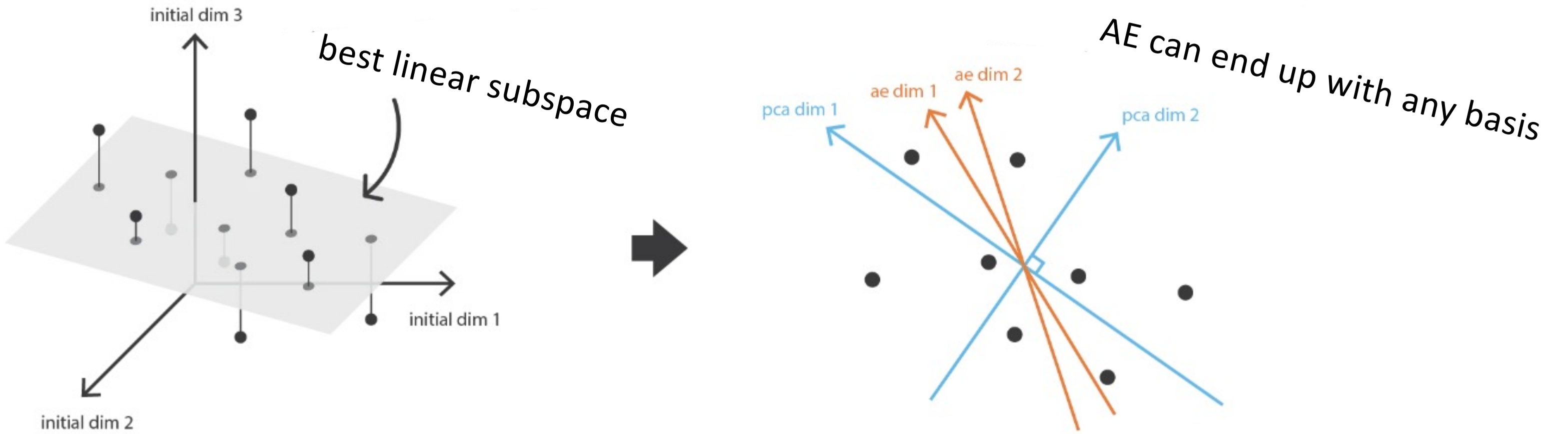
Linear Transformation!!

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

Neural Network Autoencoder – 1 Linear Layer

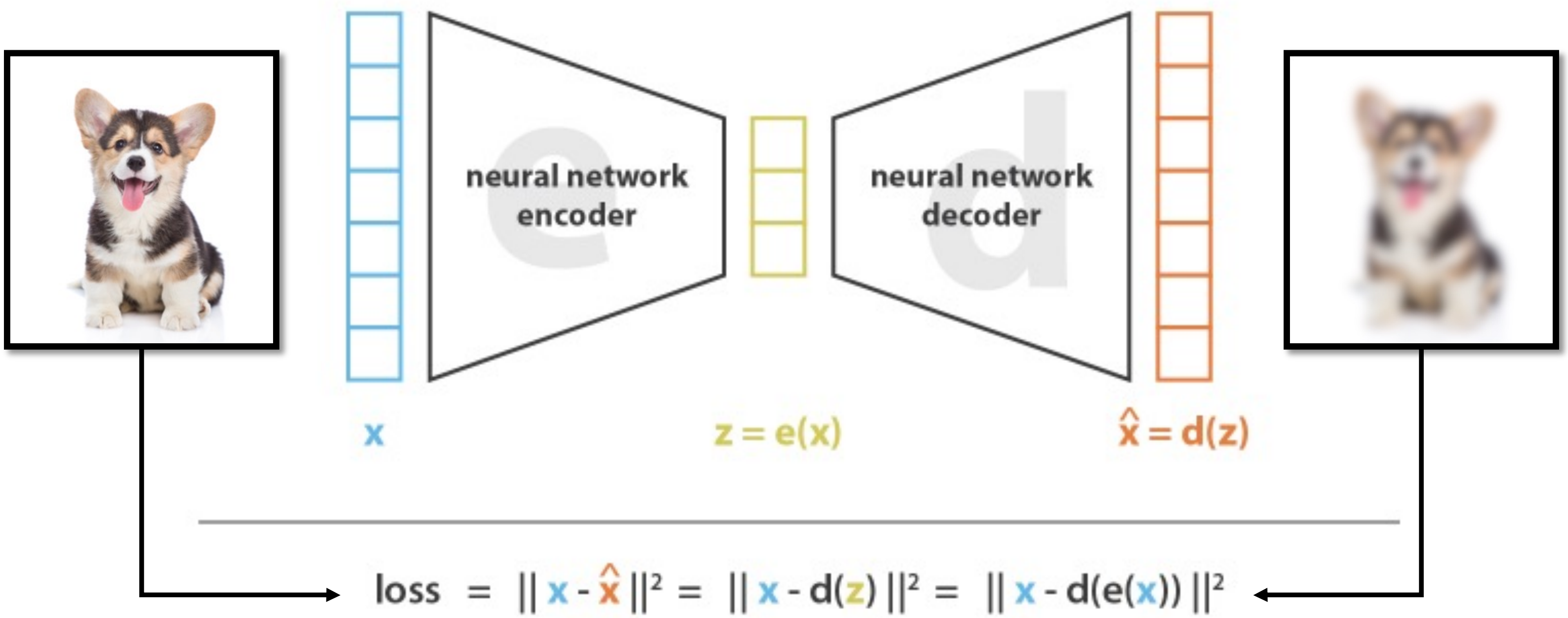


We could define encoder and decoder to each have one linear layer (no activation function), but it wouldn't necessarily converge during training to PCA solution.



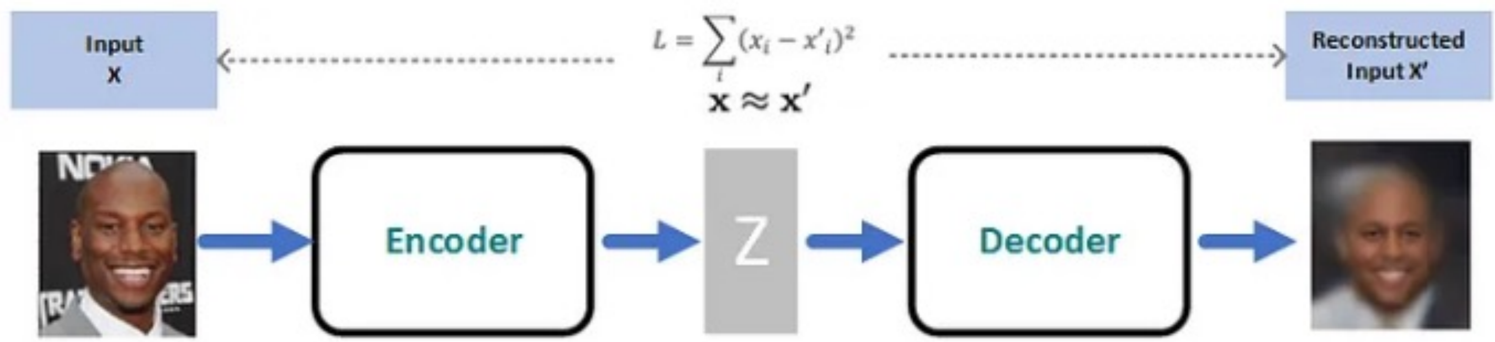
Rocca, "Understanding Variational Autoencoders (VAEs)", 2019

Neural Network Autoencoder



[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

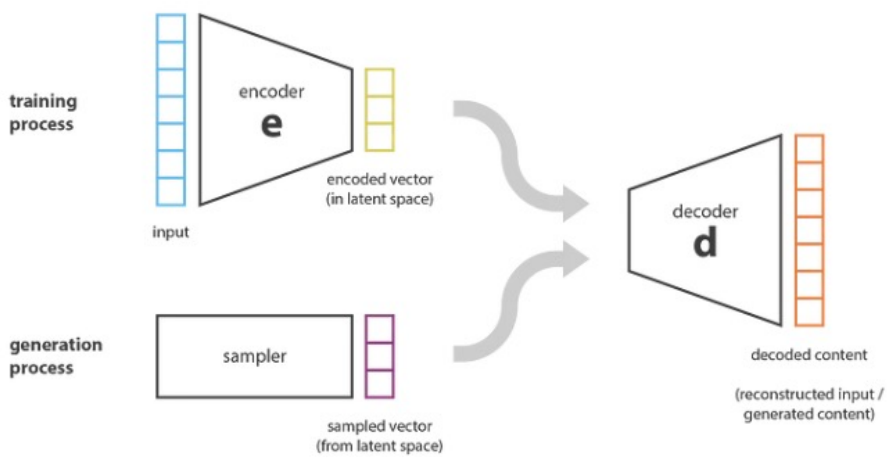
Autoencoder Reconstruction



Trained on CelebA dataset.

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

Can we generate new samples with autoencoder?



Train encoder and decoder as autoencoder.

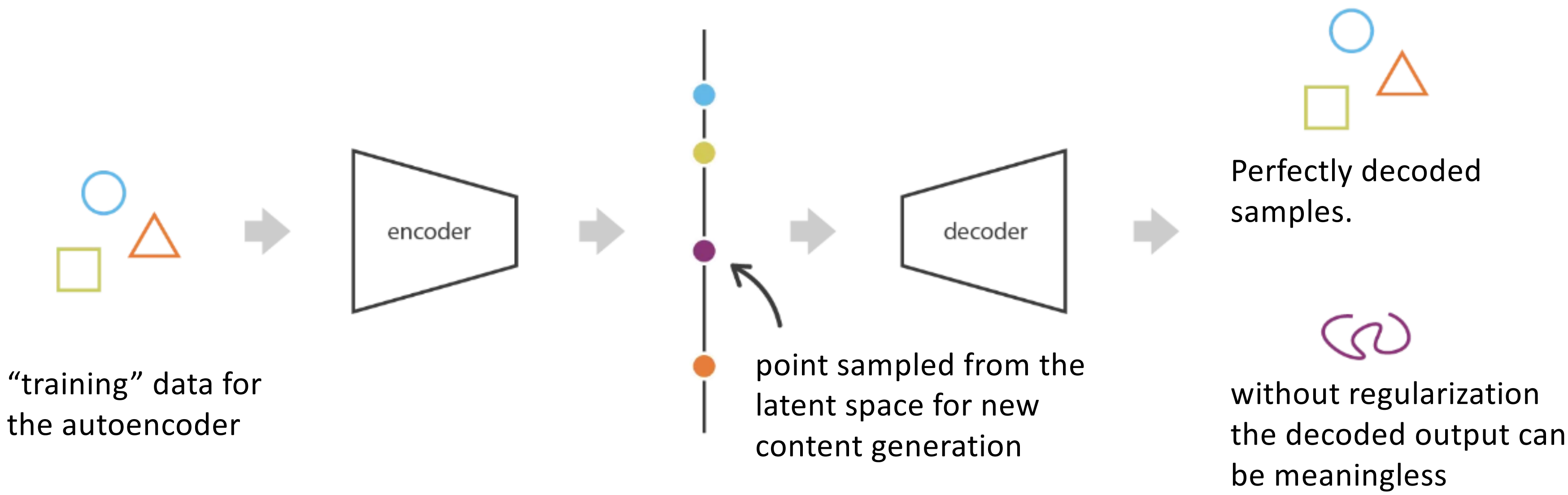
Randomly select a different point in the latent space.

Provide as input to the decoder to generate an output.

Will this produce a good quality output? Why?

[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

Extreme case: Memorization



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

[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

Outline

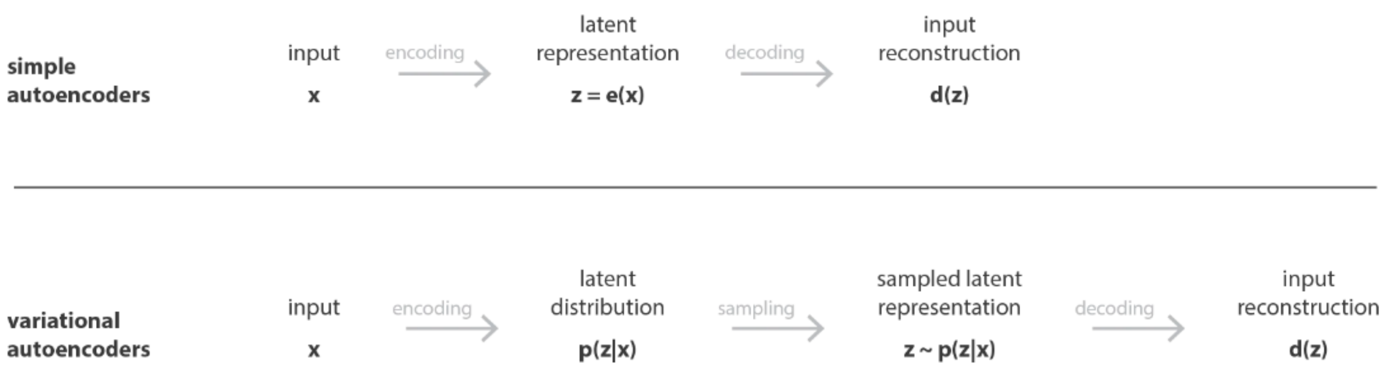
- Autoencoder and its limitations
- Intuition behind VAEs
- Derivation of VAE
- Example applications

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.

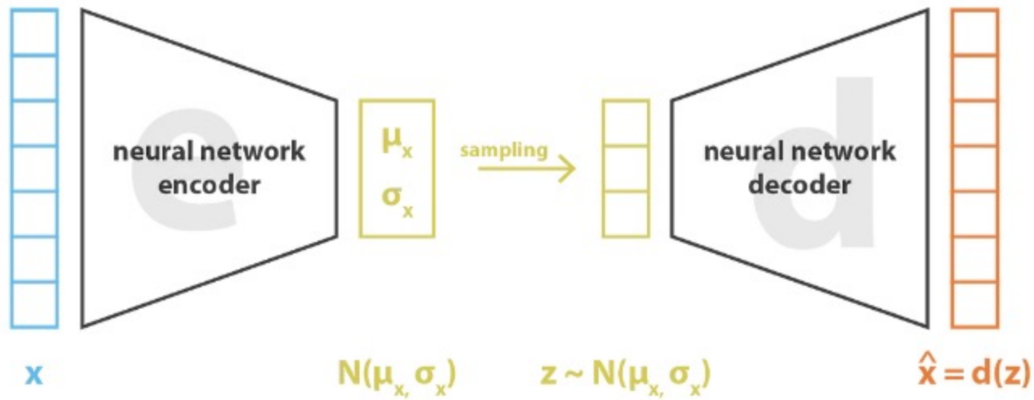
Instead of encoding as a *single point*, encode it as a *distribution* over the latent space.

Assume distributions are normal.



[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

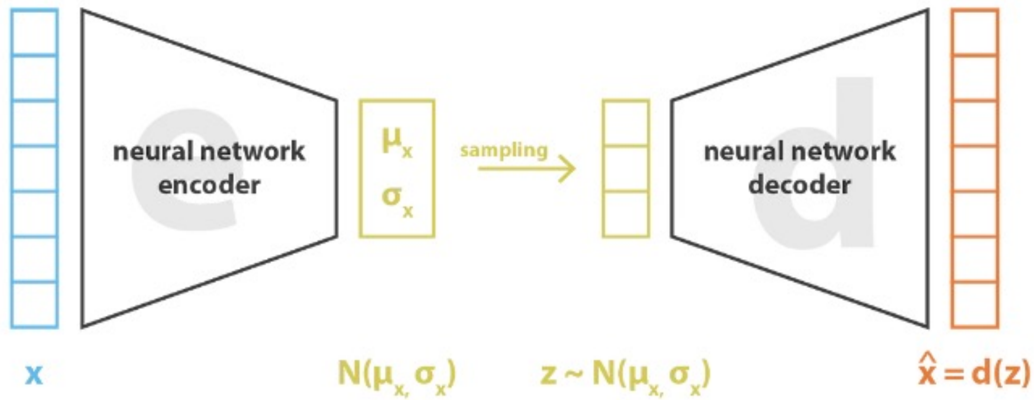
Variational Autoencoder



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

[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

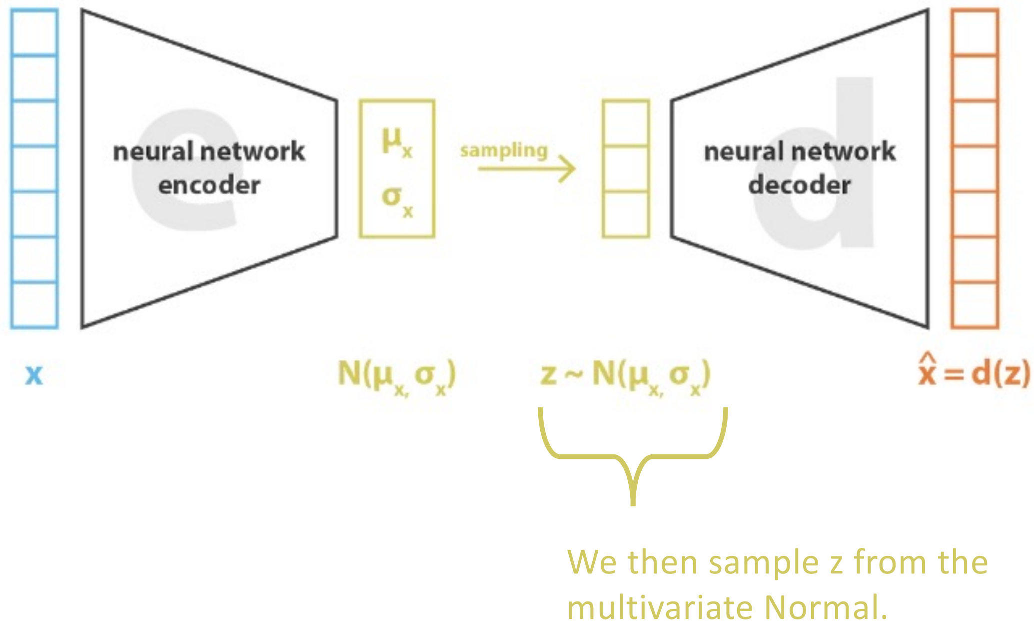
Variational Autoencoder



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

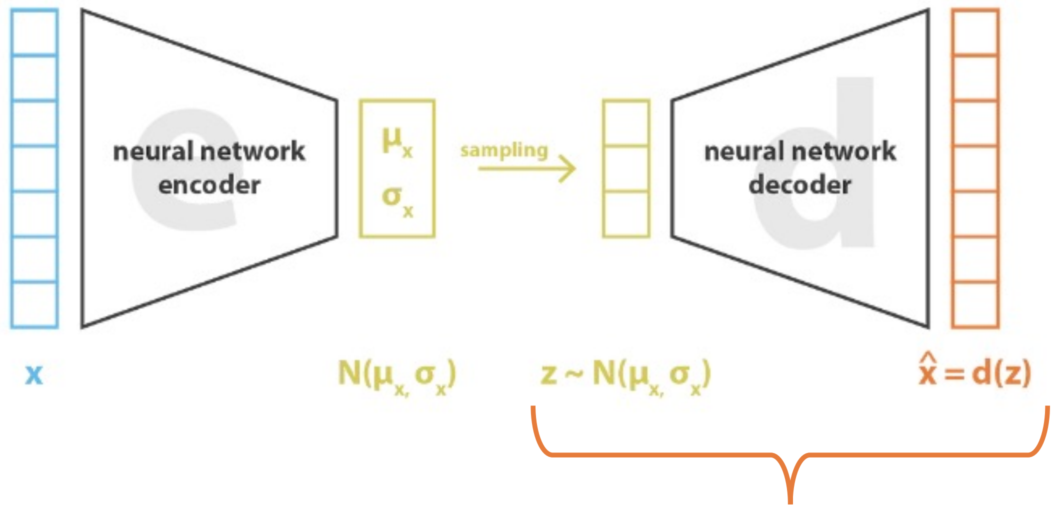
[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

Variational Autoencoder



[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

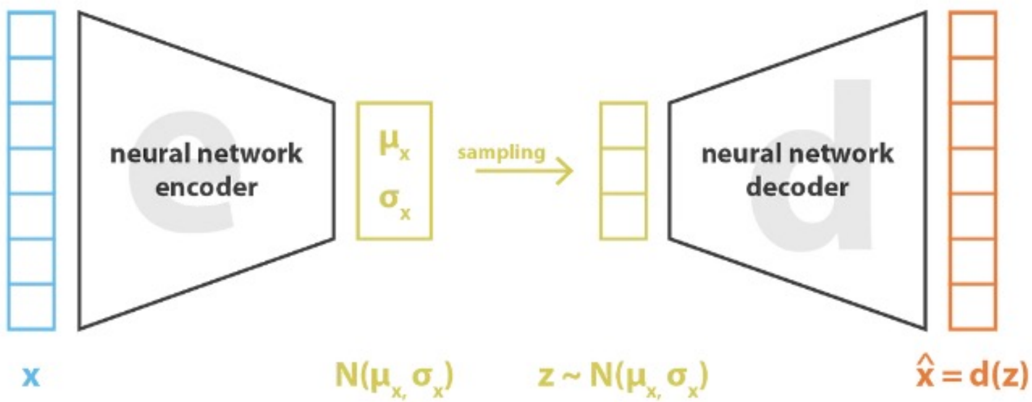
Variational Autoencoder



Then input z to the decoder network to produce output.

[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

Variational Autoencoder

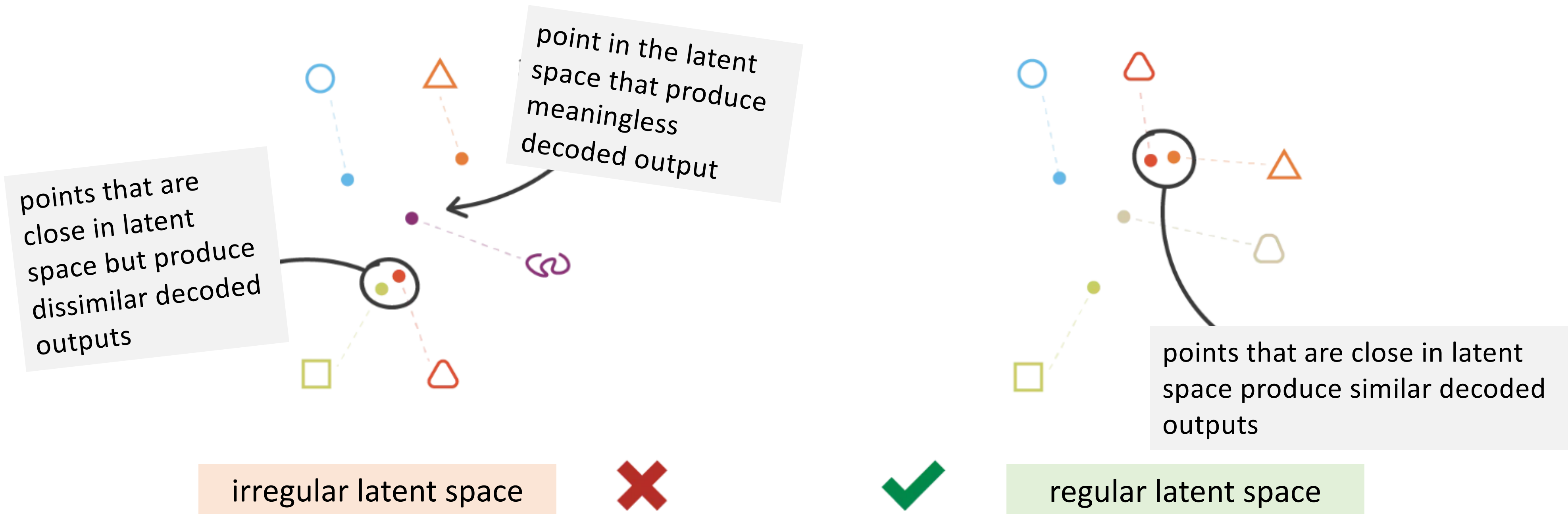


$$\text{loss} = \underbrace{\| x - \hat{x} \|^2}_{\text{L2 Loss}} + \underbrace{\mathrm{KL}[\, N(\mu_x, \sigma_x), N(0, I)\,]}_{\text{Kulback-Leibler divergence}} = \| x - d(z) \|^2 + \mathrm{KL}[\, N(\mu_x, \sigma_x), N(0, I)\,]$$

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



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

Encoding to Normal distributions is not enough

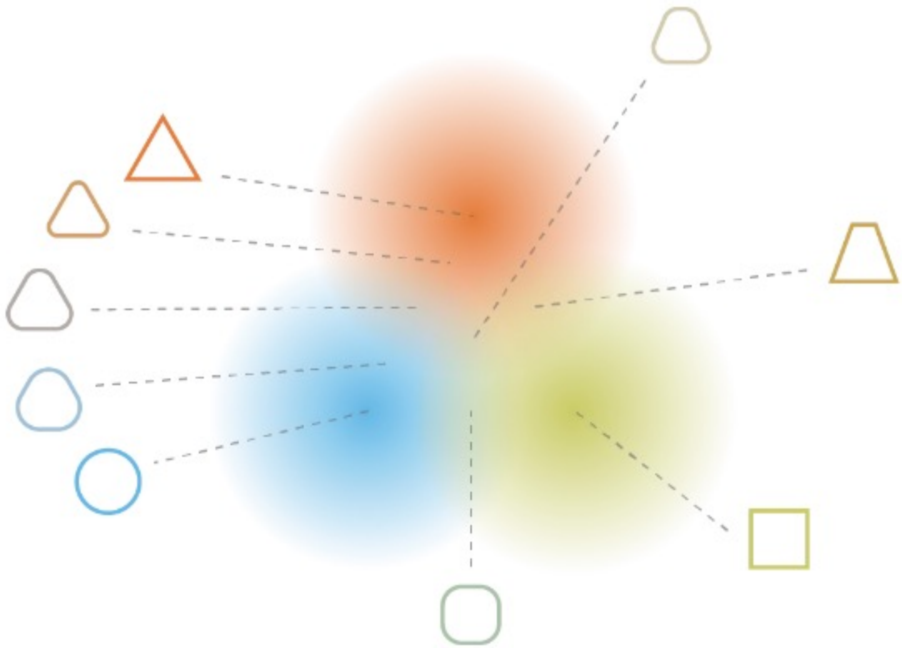
without regularization ✗ ✓ with regularization

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

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

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.

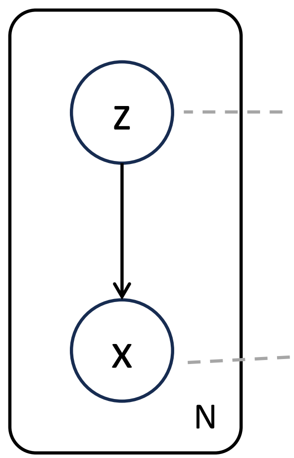
[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

Dall-E 3

Outline

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

Preliminaries: Bayesian Models

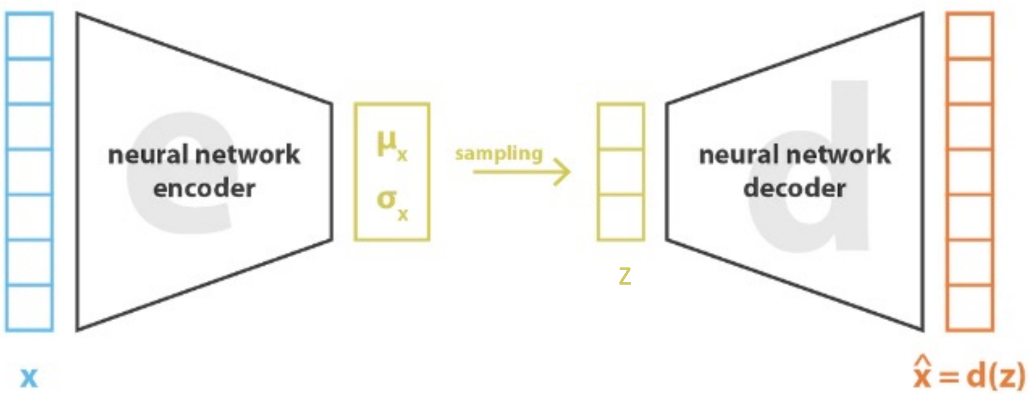


$p(z)$ *prior* – prior knowledge or belief about z

$p(x|z)$ *likelihood* – probability of a sample given z

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

Bayesian Inference

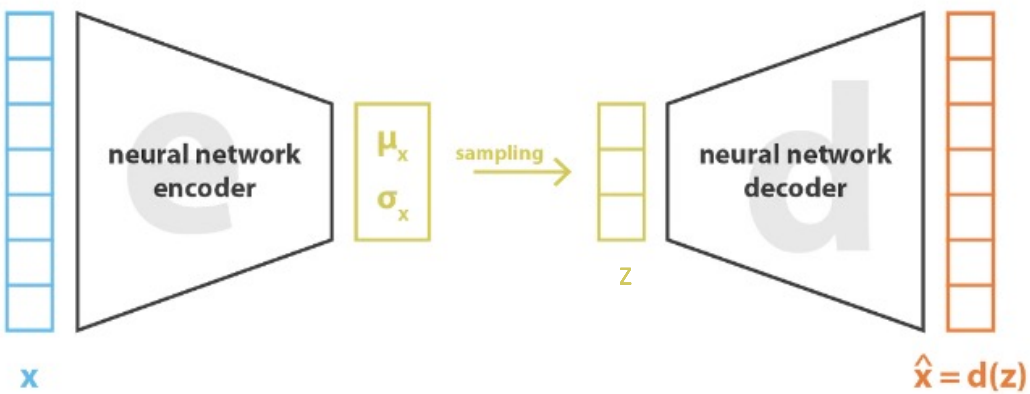


Probabilistic Encoder
$p(z|x)$

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
$p(x|z)$

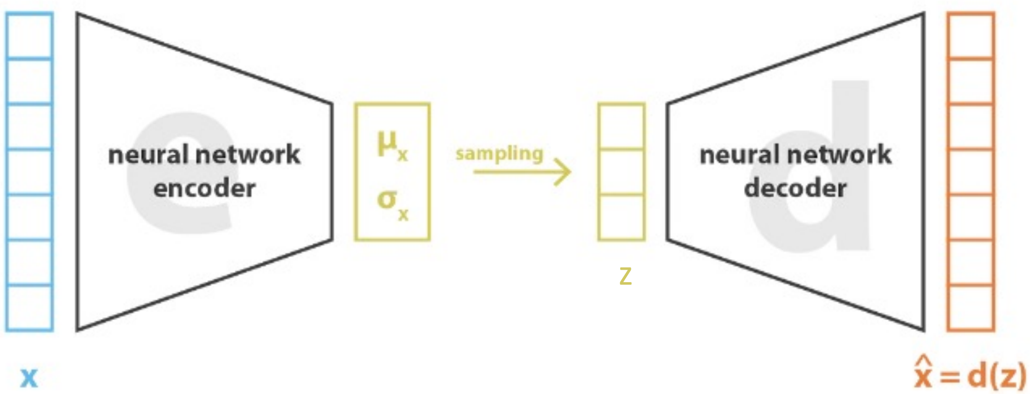
likelihood – probability of a sample given z

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

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

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

Bayesian Inference



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

prior – prior knowledge or belief about z

evidence – probability distribution of our observed data

Probabilistic Encoder

$$\underbrace{p(z|x)}$$

posterior – update our knowledge of z given a new sample

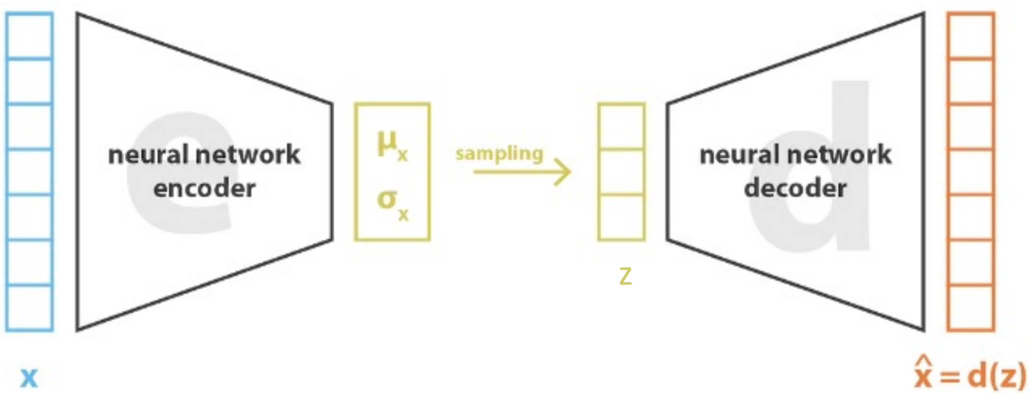
Probabilistic Decoder

$$\underbrace{p(x|z)}$$

likelihood – probability of a sample given z

[Rocca, "Understanding Variational Autoencoders (VAEs)", 2019](#)

Bayesian Inference



Probabilistic Encoder

$$\underbrace{p(z|x)}$$

posterior – update our knowledge of z given a new sample

Probabilistic Decoder

$$\underbrace{p(x|z)}$$

likelihood – probability of a sample given z

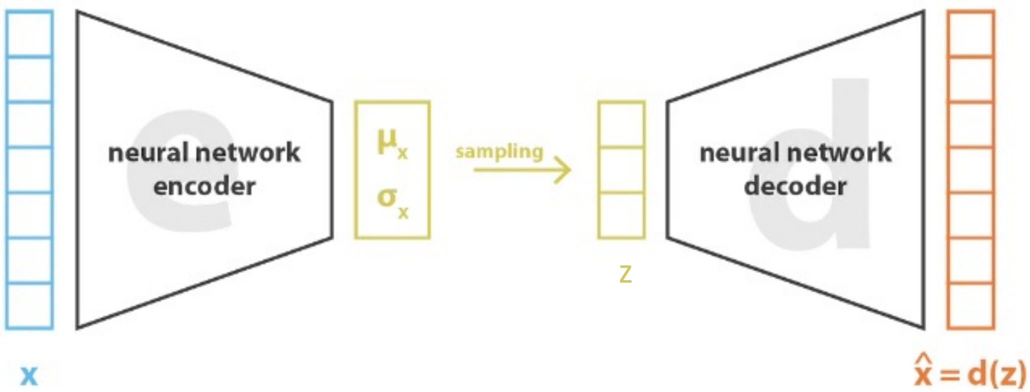
$$\underbrace{p(z|x)}_{\textit{posterior}} = \frac{\overbrace{p(x|z)}^{\textit{likelihood}}\overbrace{p(z)}^{\textit{prior} \text{ – prior knowledge or belief about z}}}{p(x)}$$

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

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
$\underbrace{p(z|x)}_{\textit{posterior}}$

Probabilistic Decoder
$\underbrace{p(x|z)}_{\textit{likelihood}}$

Assume that the *prior* is a standard Gaussian

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

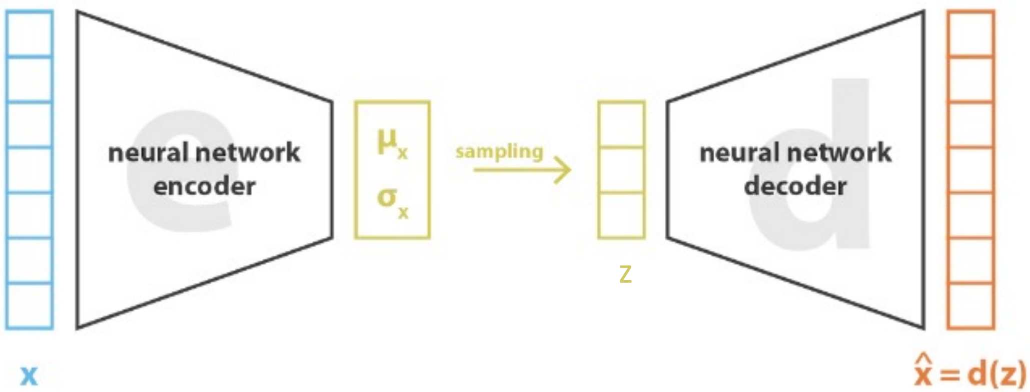
And *likelihood* is a Gaussian

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

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
$$\underbrace{p(z|x)}_{\textit{posterior}}$$

Probabilistic Decoder
$$\underbrace{p(x|z)}_{\textit{likelihood}}$$

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.

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

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:

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

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

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

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

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

Variational Inference

$$
\begin{aligned}
(g^*, h^*) &= \underset{(g,h)\in G\times H}{\arg\min}\; KL(q_x(z), p(z|x)) \\
&= \underset{(g,h)\in G\times H}{\arg\min} \left(\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}\left(\log \frac{p(x|z)p(z)}{p(x)} \right) \right) \\
&= \underset{(g,h)\in G\times H}{\arg\min}\; (\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}(\log p(z)) - \mathbb{E}_{z\sim q_x}(\log p(x|z)) + \mathbb{E}_{z\sim q_x}(\log p(x))) \\
&= \underset{(g,h)\in G\times H}{\arg\max}\; (\mathbb{E}_{z\sim q_x}(\log p(x|z)) - KL(q_x(z), p(z))) \\
&= \underset{(g,h)\in G\times H}{\arg\max} \left(\underbrace{\mathbb{E}_{z\sim q_x}\left(-\frac{||x - f(z)||^2}{2c} \right)}_{\text{Log of the Gaussian likelihood } p(x|z) \equiv \mathcal{N}(f(z), cI).} - KL(q_x(z), p(z)) \right)
\end{aligned}
$$

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

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

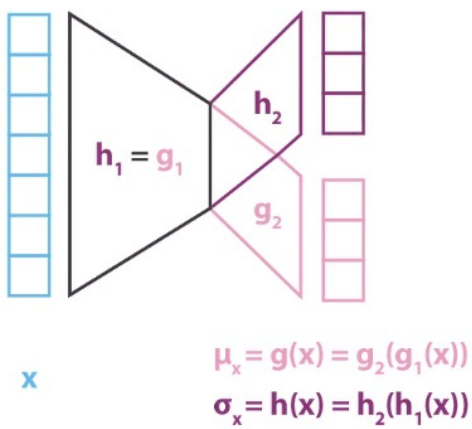
Variational Inference

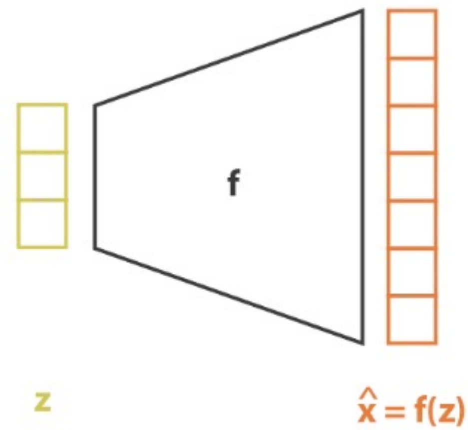
We are looking for optimal f^*, g^* and h^* such that

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

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

Enter the Neural Networks

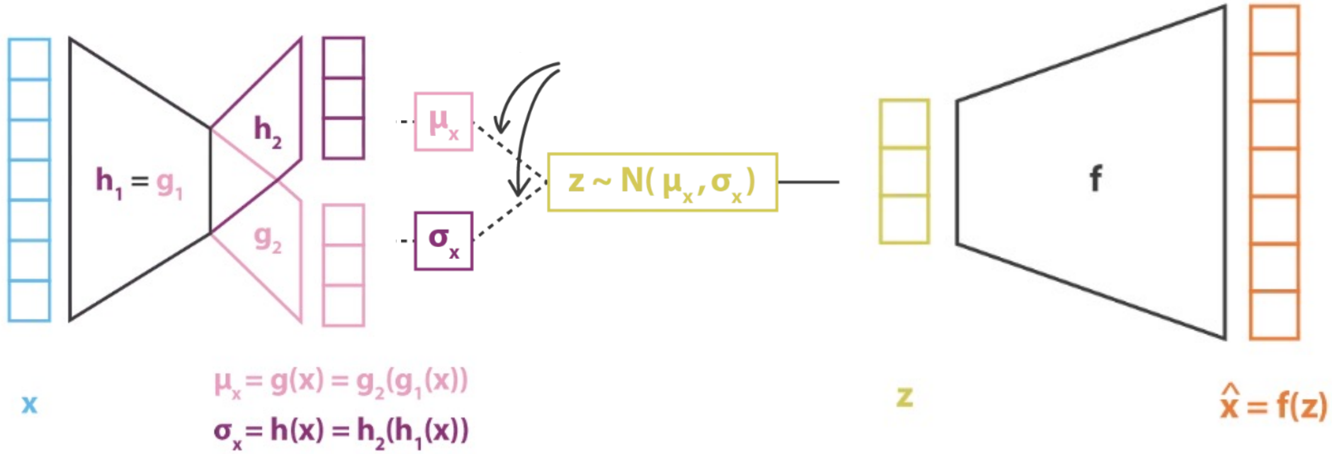




Encoder produces the mean and variance.

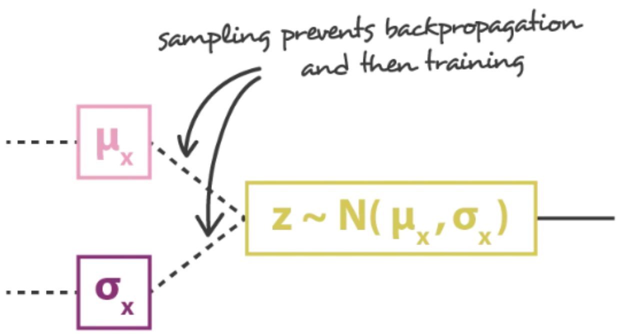
Decoder reconstructs the input (during training)

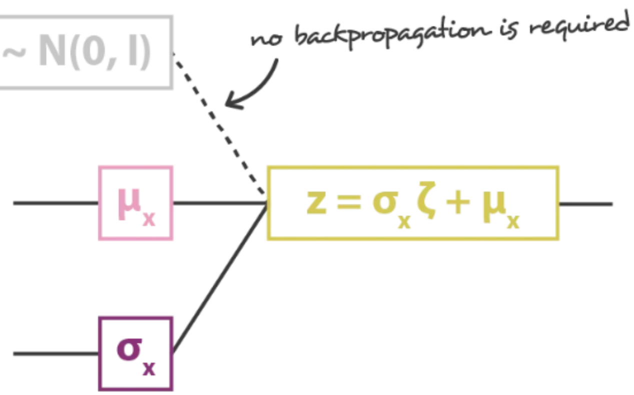
But one more problem to solve



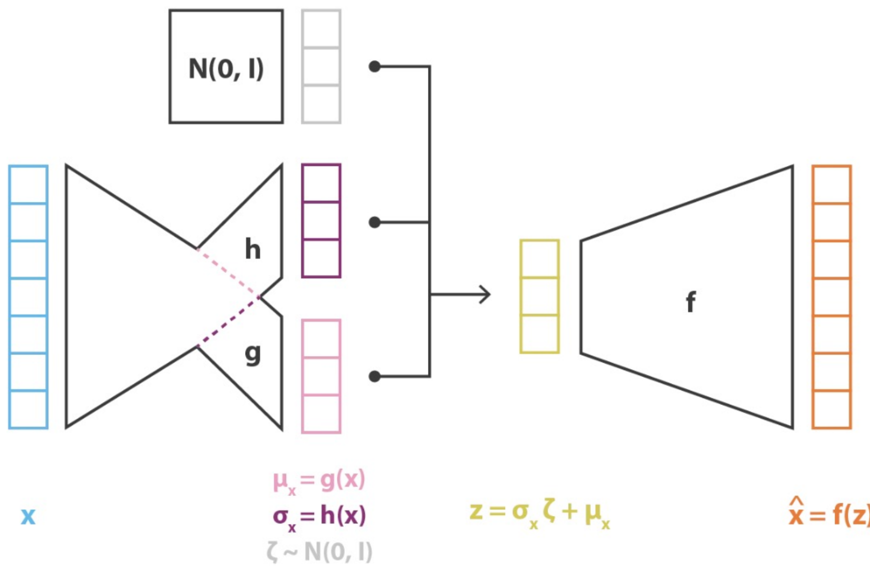
We can’t backpropagate through the sampling step.

Use the reparameterization trick





Putting it all together



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

Convert C = 1/(2c).

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

We have as trainable neural network!

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:

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

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

Dall-E 3

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

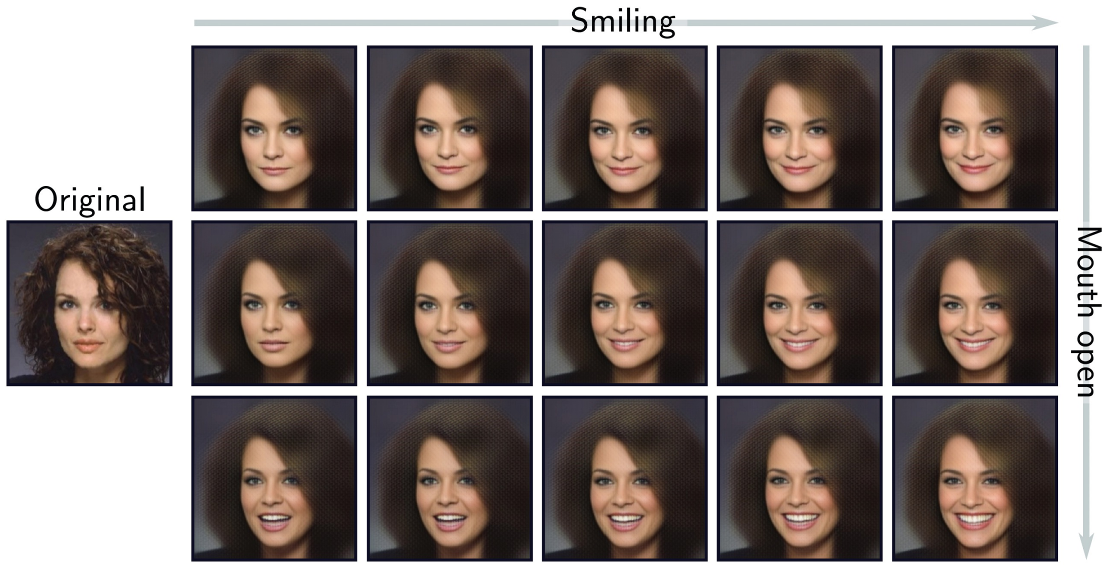
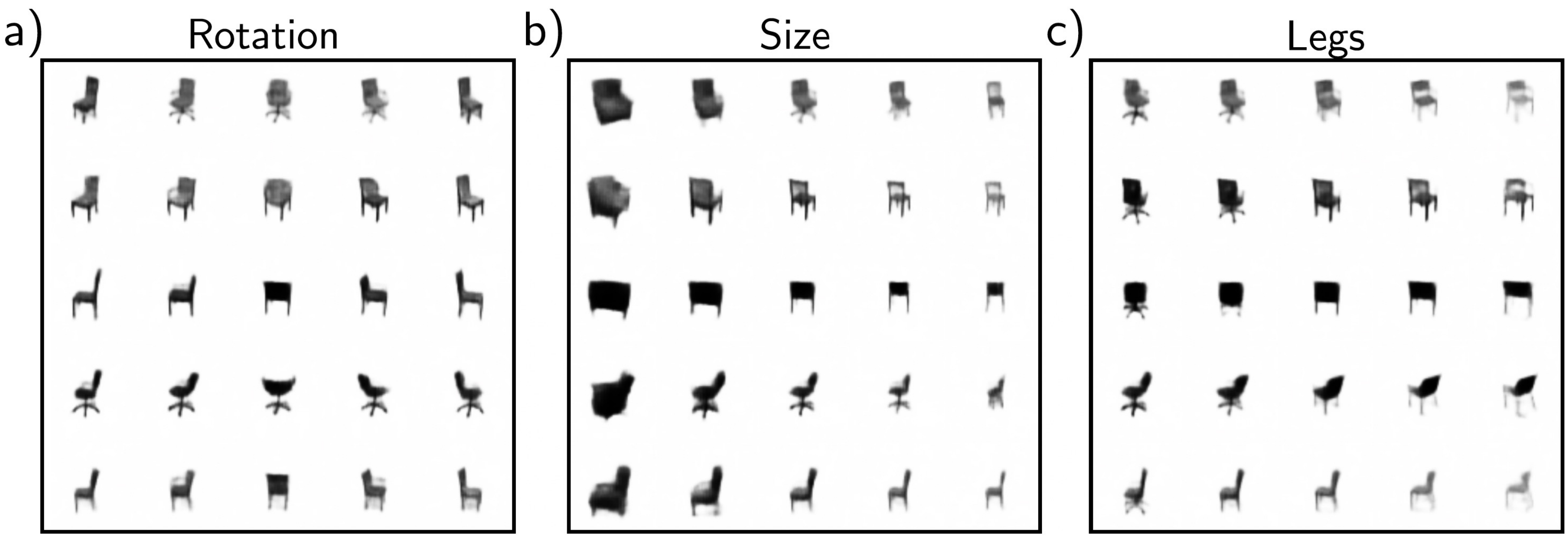
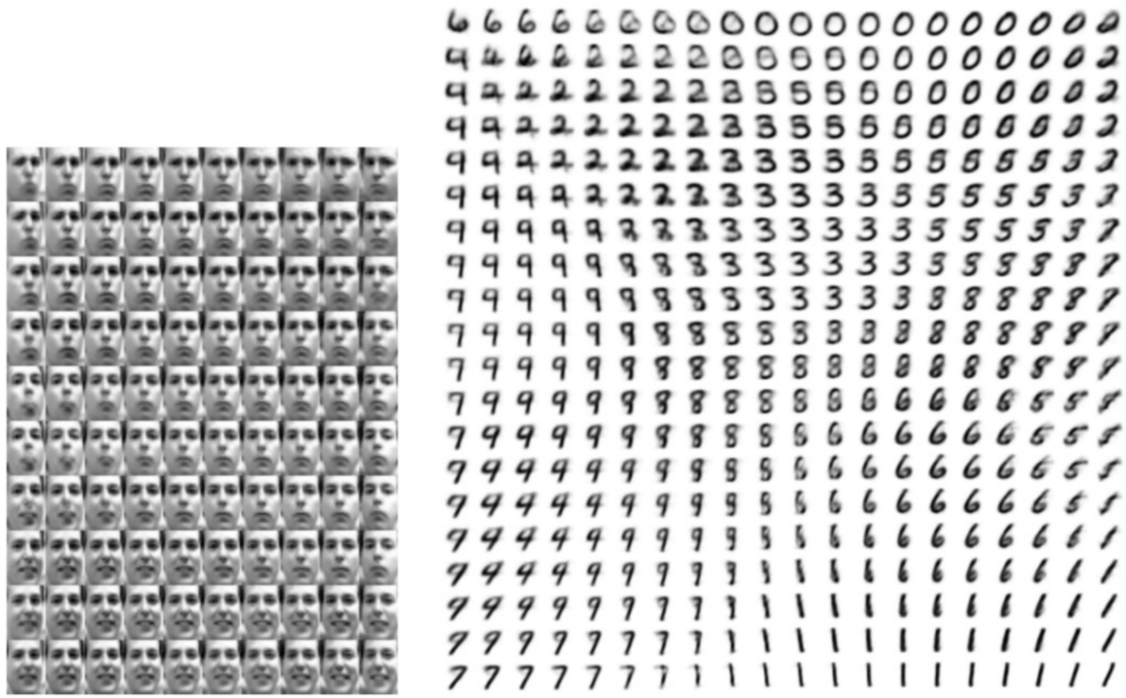


Figure 17.13 Resynthesis. The original image on the left is projected into the latent space using the encoder, and the mean of the predicted Gaussian is chosen to represent the image. The center-left image in the grid is the reconstruction of the input. The other images are reconstructions after manipulating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

Disentanglement of the latent space



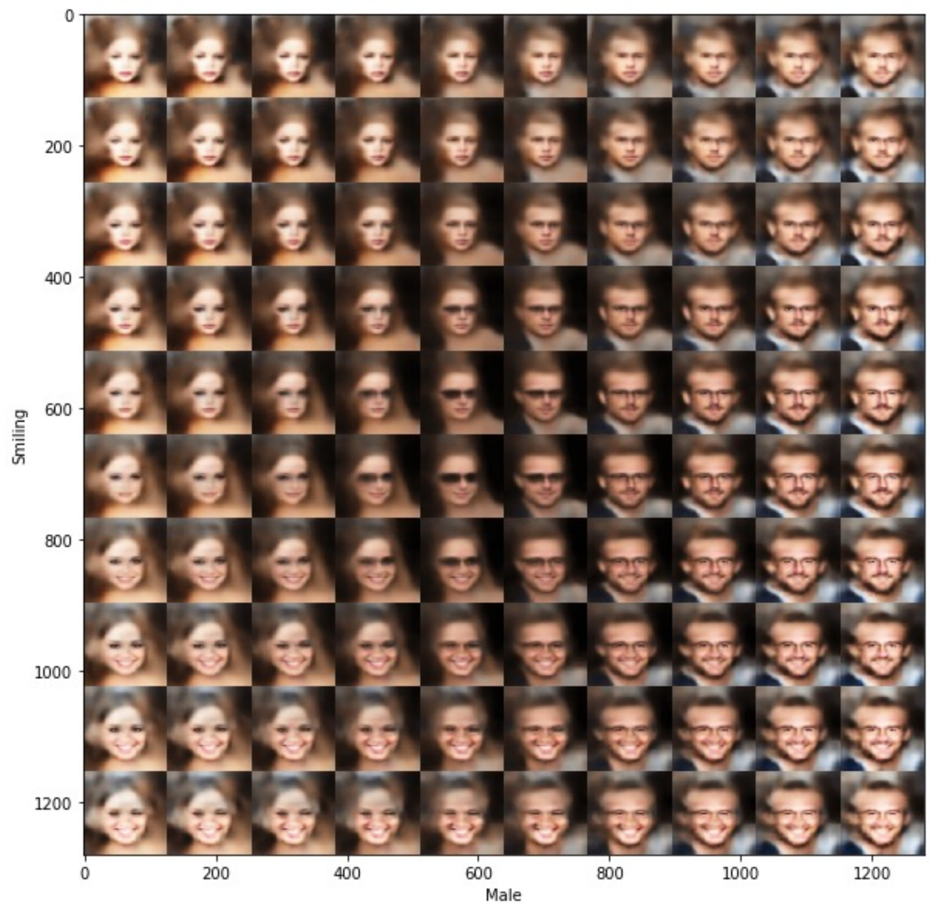
Chen et al (2021) “Cross-layer distillation with semantic calibration.”



(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://towardsdatascience.com/variational-autoencoders-vaes-for-dummies-step-by-step-tutorial-69e6d1c9d8e9

Debiasing

Capable of uncovering **underlying features** in a dataset

Homogeneous skin color, pose

vs

Diverse skin color, pose, illumination

How can we use this information to create fair and representative datasets?

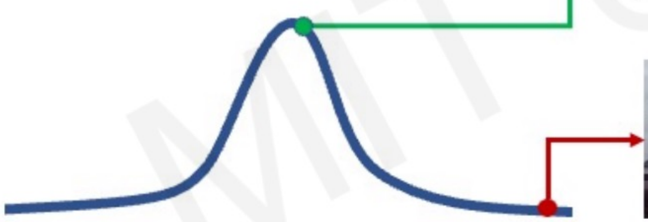
Amini et al, “Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure,” 2019

Outlier Detection

- **Problem:** How can we detect when we encounter something new or rare?
- **Strategy:** Leverage generative models, detect outliers in the distribution
- Use outliers during training to improve even more!

95% of Driving Data:
(1) sunny, (2) highway, (3) straight road

Detect outliers to avoid unpredictable behavior when training



Edge Cases

Harsh Weather

Pedestrians

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

Upcoming Topics

- Diffusion Models
- Graph Neural Networks
- Reinforcement Learning

Feedback

[Link]