Representation Learning

CAS AML 2024

Dr. Mykhailo Vladymyrov

Data Science Lab

University of Bern

1. Overview

Supervised learning

- Advantages:
  - Easy to train
  - Optimized towards the task at hand

- Limitations:
  - Requires annotated data
  - Learned representation might be not suitable for another task

Representation Learning

It is beneficial to build such representation for the data that it is:

- Low dimensional
- Describes the data in general (not biased to a specific task)
- Suitable for manipulation

Representation Learning

It is beneficial to build such representation for the data that it is:

- Low dimensional
- Describes the data in general (not biased to a specific task)
- Suitable for manipulation

- Find outliers
- Study data point distribution
- Perform operations in the learned representation

Approaches to Representation Learning

Approaches to Representation Learning

Are countless….

Approaches to Representation Learning

Here we will look at a few fundamental ones.

You can directly apply these to better study and process your data.

Understanding and obtaining some intuition on these will greatly help you to understand the other ones

What is autoencoder?

- Unsupervised deep machine learning algorithm.

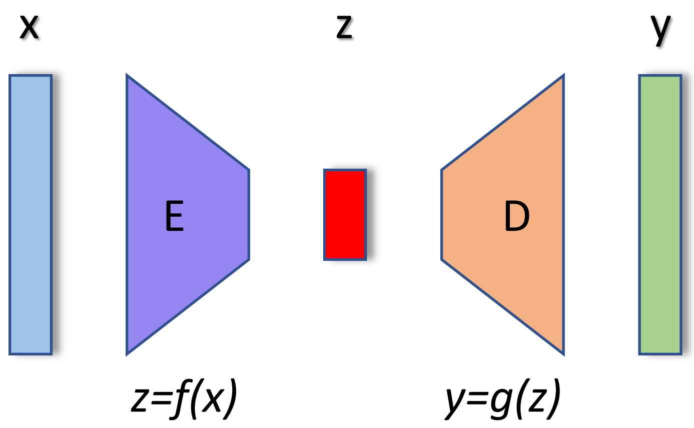
- AE learns a (lower dimension) representation of the input data

What is autoencoder?

Transform data representation.

Given input *x*:

- Encoder produces latent representation z:
$$z = f(x)$$
- Decoder reconstructs data:
$$y = g(z)$$

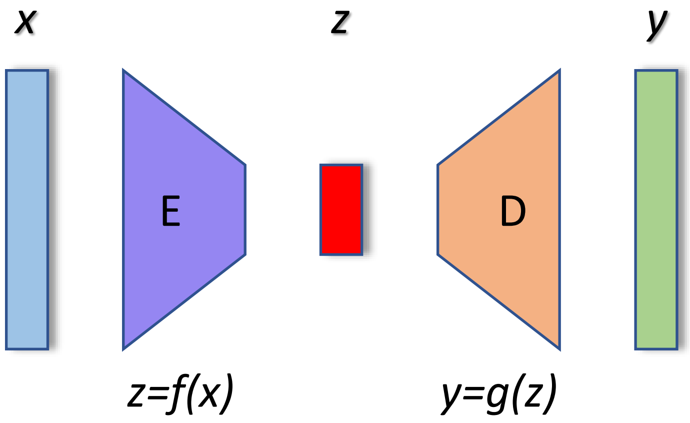


What is autoencoder?

Transform data representation.

Given input *x*:

- Encoder produces latent representation z:
$$z = f(x)$$
- Decoder reconstructs data:
$$y = g(z)$$
- Reconstruction, e.g. L_2:
$$\text{Loss} = |x - y|^2$$



Simple AE

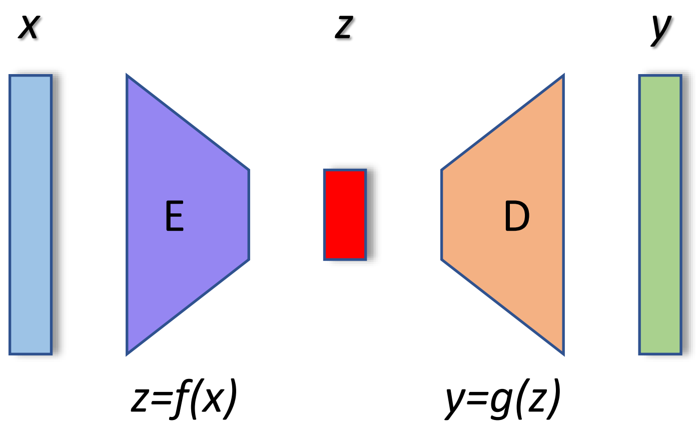
Encoder and decoder – fully connected neural networks:

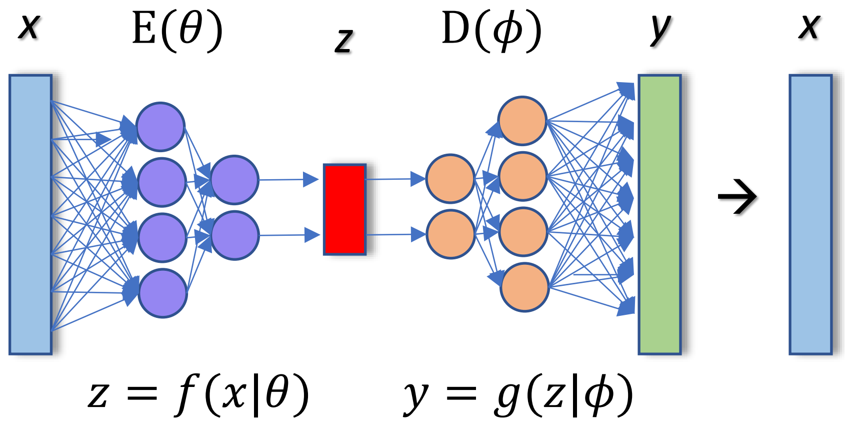
$$y = g(z\,|\phi)$$

$$y = g(f(x \mid \theta) \mid \phi)$$

$$\theta, \phi = \underset{\theta,\phi}{\arg\min} |y(x \mid \theta, \phi)\ - x|^2$$

$$\theta, \phi = \underset{\theta,\phi}{\arg\min}\, |g(f(x \mid \theta) \mid \phi) - x|^2$$





Simple AE

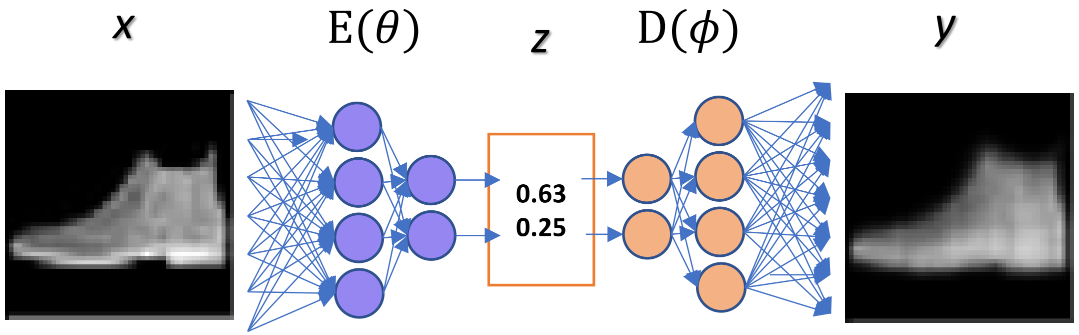
Encoder and decoder – fully connected neural networks:

$$y = g(z\,|\phi)$$

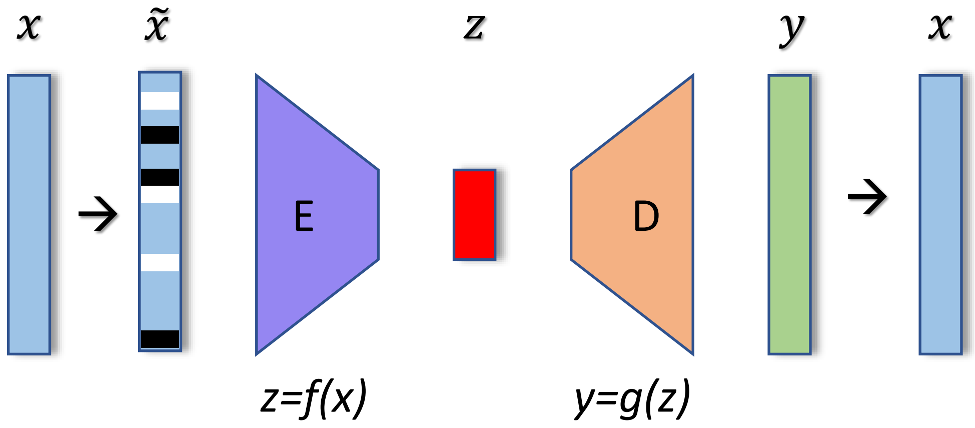
$$y = g(f(x\,|\,\theta)\,|\,\phi)$$

$$\theta,\phi = \underset{\theta,\phi}{\arg\min}|y(x\,|\,\theta,\phi)\; - x|^2$$

$$\theta,\phi = \underset{\theta,\phi}{\arg\min}\,|g(f(x\,|\,\theta)\,|\,\phi) - x|^2$$



Denoising AE



The encoder can be used to reconstruct corrupt data.

In denoising AE part of the input is corrupt before feeding in the network:

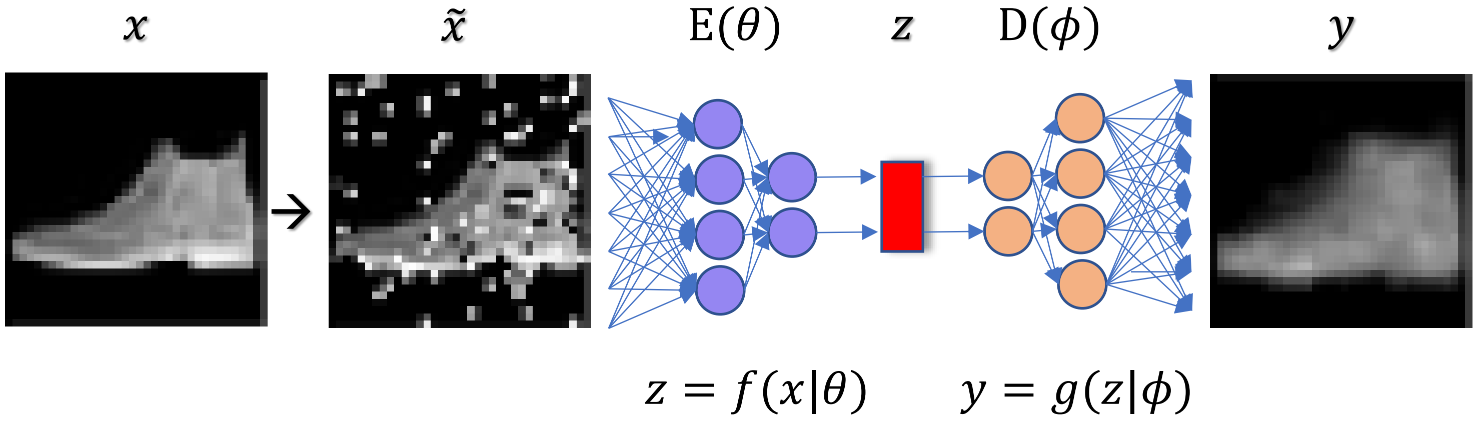
$$\tilde{x} = \ x + \epsilon$$

where, e.g.

$$\epsilon \in N(\mu, \sigma)$$

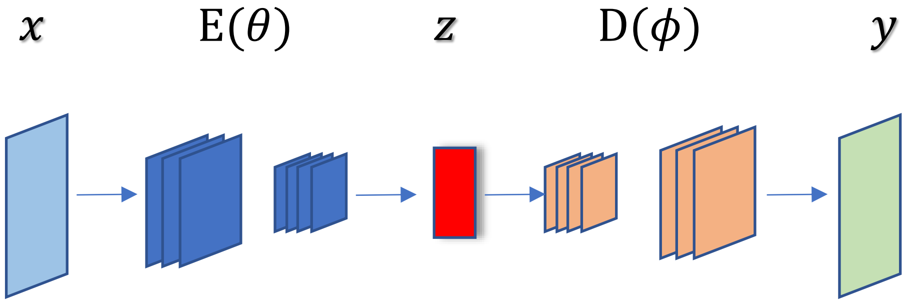
Or “salt & pepper” noise

Denoising AE



Convolutional AE

For data with continuous axes convolutional neural network enable massive parameter reduction

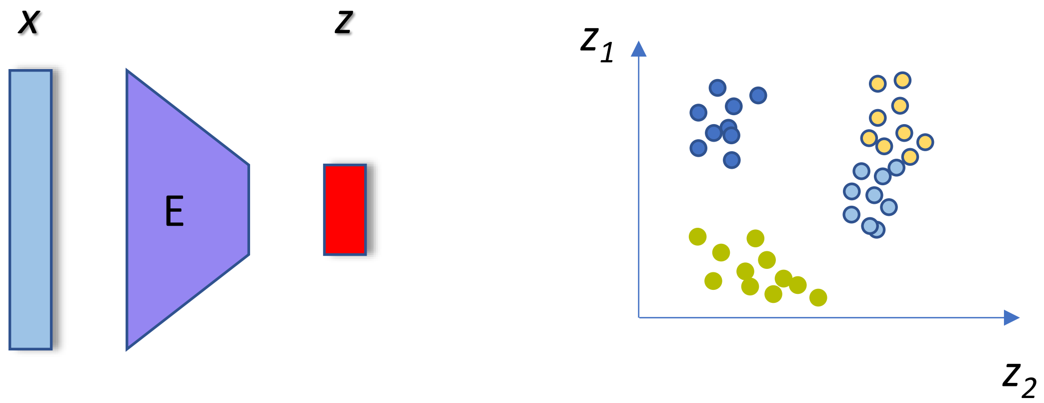


$z = f(x|\theta) \qquad y = g(z|\phi)$

Meaning of latent representation

Useful representations of the data:

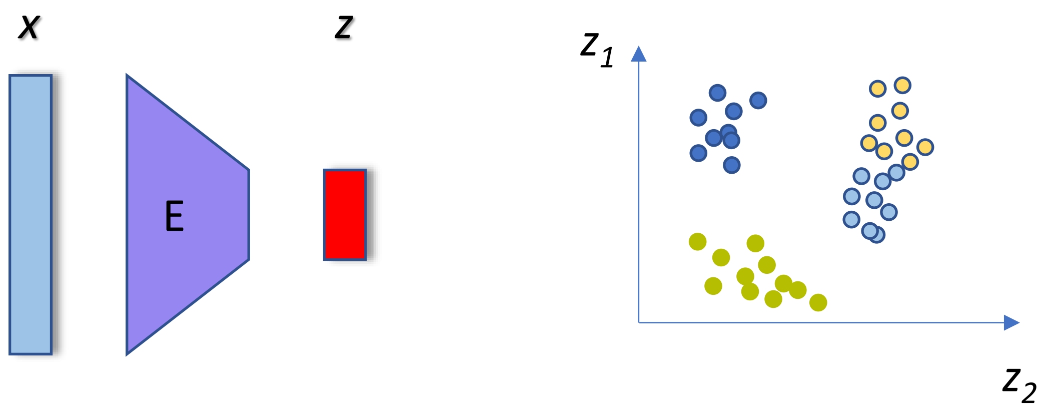
- Unsupervised clustering
- Exploration of data patterns



Meaning of latent representation

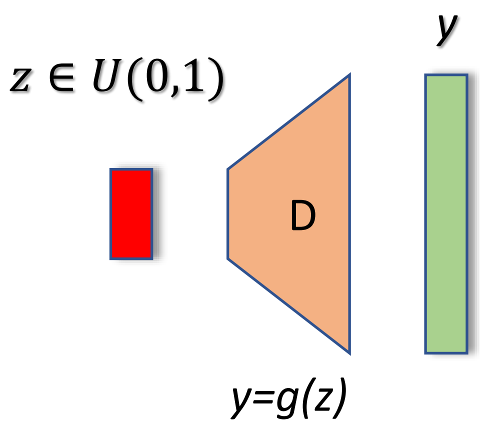
Useful representations of the data:

- Unsupervised clustering
- Exploration of data patterns



- Generation of new datapoints by sampling in latent space:

$$z \in U(0,1)$$



Variational autoencoder

The distribution of $\mathrm{x} \in X$ is complex. We would like to learn some meaningful representation $\mathrm{z} \in Z$. How will that distribution look?

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

intractable

Thus, we try to approximate $p(z|x)$ with another distribution, that can be estimated, $q(z|x)$.
As measure of approximation the Kullback-Leibler divergence can be used. KL-divergence, or relative entropy from Q to P is given like

$$D_{KL}(P \mid\mid Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}$$

Variational autoencoder

To make $q(z|x)$ similar to $p(z|x)$ we thus want to minimize:

$$D_{KL}\left(q(z|x)||p(z|x)\right) = \sum_z q(z|x) \log \frac{q(z|x)}{p(z|x)}$$

$$= \sum_z q(z|x)(\log q(z|x) \ - \log p(z|x)) =$$

$$= \sum_z q(z|x) \left(\log q(z|x) \ - \log \frac{p(x|z)p(z)}{p(x)}\right) =$$

$$= \sum_z q(z|x) \left(\log q(z|x) \ - \log \frac{p(x,z)}{p(x)}\right) =$$

$$= \sum_z q(z|x) \left(\log \frac{q(z|x)}{p(x,z)} + \log p(x)\right) =$$

$$= \sum_z q(z|x) \log \frac{q(z|x)}{p(x,z)} \ + \sum_z q(z|x) \log p(x) =$$

$$= \sum_z q(z|x) \log \frac{q(z|x)}{p(x,z)} \ + \log p(x) \sum_z q(z|x) = \sum_z q(z|x) \log \frac{q(z|x)}{p(x,z)} \ + \log p(x)$$

Variational autoencoder

$$\underbrace{D_{KL}\big(q(z|x)||p(z|x)\big)}_{\geq 0} = \sum_z q(z|x)\log\frac{q(z|x)}{p(x,z)} + \underbrace{\log p(x)}_{\text{constant}}$$

$$\mathrm{E}LBO = -\sum_z q(z|x)\log\frac{q(z|x)}{p(x,z)} \leq \log p(x)$$

Variational autoencoder

So we want to minimize in turn

$$Loss = -\mathrm{E}LBO = \sum_z q(z|x) \log \frac{q(z|x)}{p(x,z)} =$$

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

$$= \sum_z q(z|x) \log \frac{q(z|x)}{p(z)} \; - \sum_z q(z|x) \log p(x|z) \; =$$

$$= D_{KL}\big(q(z|x)||p(z)\big) \; - \sum_z q(z|x) \log p(x|z)$$

Variational autoencoder

So we want to minimize in turn

$$Loss = -\mathrm{E}LBO = \sum_z q(z|x) \log \frac{q(z|x)}{p(x,z)} =$$

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

$$= \sum_z q(z|x) \log \frac{q(z|x)}{p(z)} \quad - \sum_z q(z|x) \log p(x|z) \quad =$$

$$= \underbrace{D_{KL}\big(q(z|x) || p(z)\big)}_{\text{Regularizer ensuring specific distribution}} \quad \underbrace{- \sum_z q(z|x) \log p(x|z)}_{\text{Neg. log-likelihood of reconstruction}}$$

Variational autoencoder

If x is gaussian then NLL = $(x - \hat{x})^2$ (L_2 loss)

We can then demand $p(z) = N(0,1)$ and $q(z|x) = N(\mu(x), \sigma(x))$, where $\mu(x), \sigma(x) \in \mathbb{R}^{n_{code}}$, and

$$D_{KL}\big(q(z|x)||p(z)\big) = \frac{1}{2}\sum_{n_{code}} (\sigma_i^2(x) + \mu_i^2(x) - 1 - \log \sigma_i^2(x))$$

Then

$$Loss = D_{KL}\big(q(z|x)||p(z)\big) - \sum_{z} q(z|x) \log p(x|z) =$$

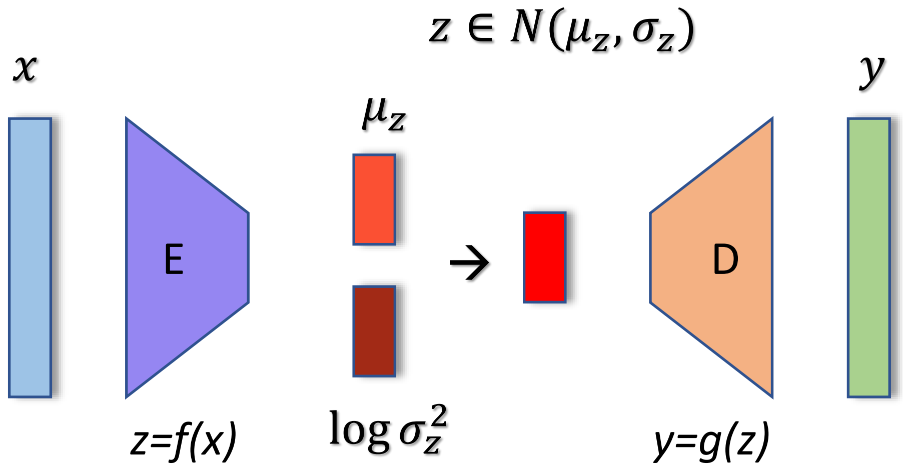
$$= \frac{1}{2}\sum_{i=1..n_{code}} (\sigma_i^2(x) + \mu_i^2(x) - 1 - \log \sigma_i^2(x)) + \sum_{j=1..n_X} \left(x_j - \widehat{x_j}\right)^2$$

Variational autoencoder

$\log \sigma_z^2$ is used instead of σ_z for numerical stability

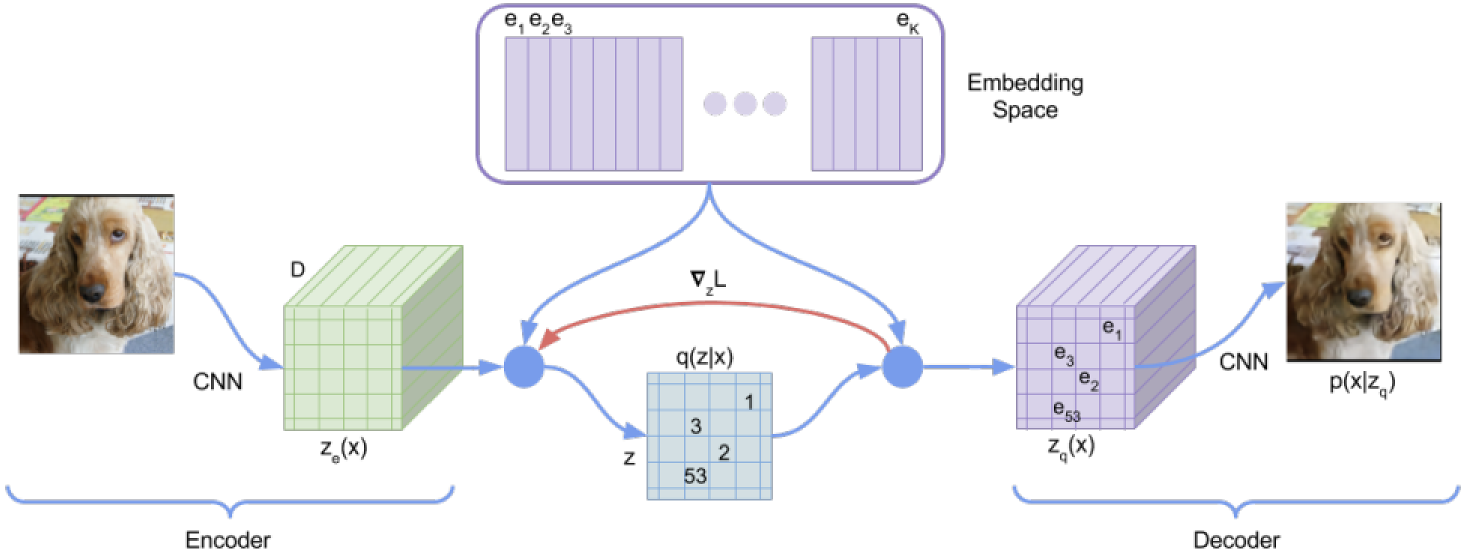
Reparameterization:

$$\epsilon \in N(0,1)$$
$$z = \mu_z + \epsilon \cdot e^{\frac{1}{2}\log \sigma_z^2}$$



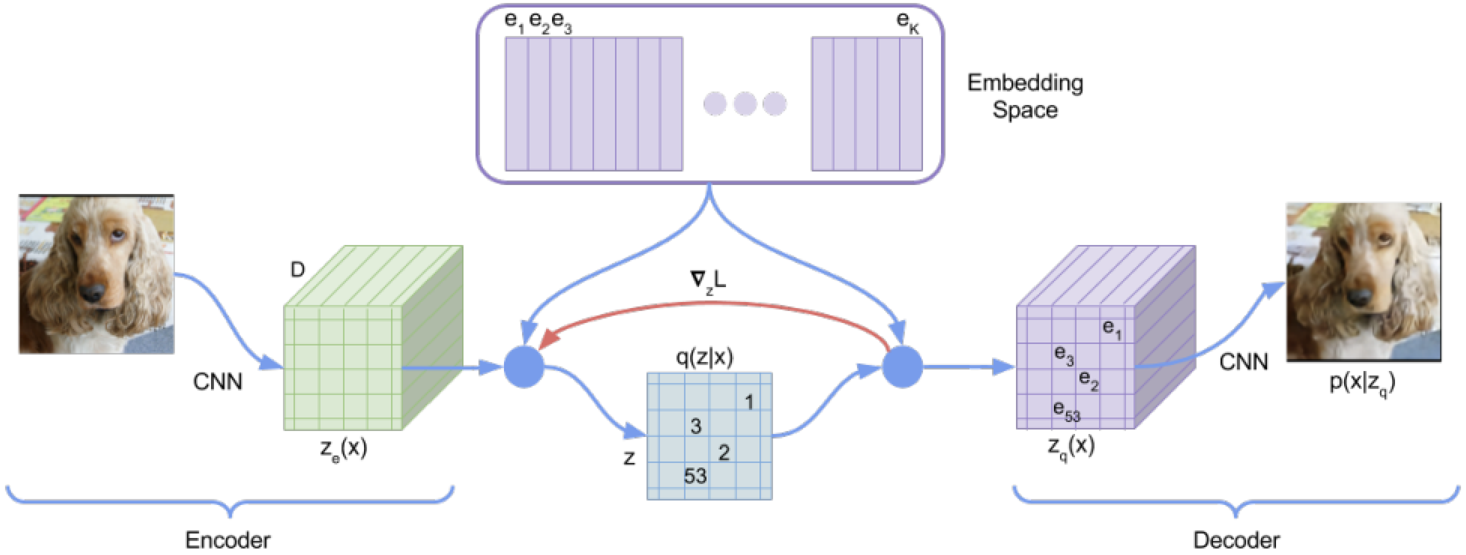
Vector Quantized Variational Autoencoder – VQ-VAE

- input encoder maps to latent grid of embeddings *z*
- vector quantization maps embeddings to (nearest) code words e_i from $\{e_1..e_K\}$
- decoder maps from quantized grid to the reconstruction



Vector Quantized Variational Autoencoder – VQ-VAE

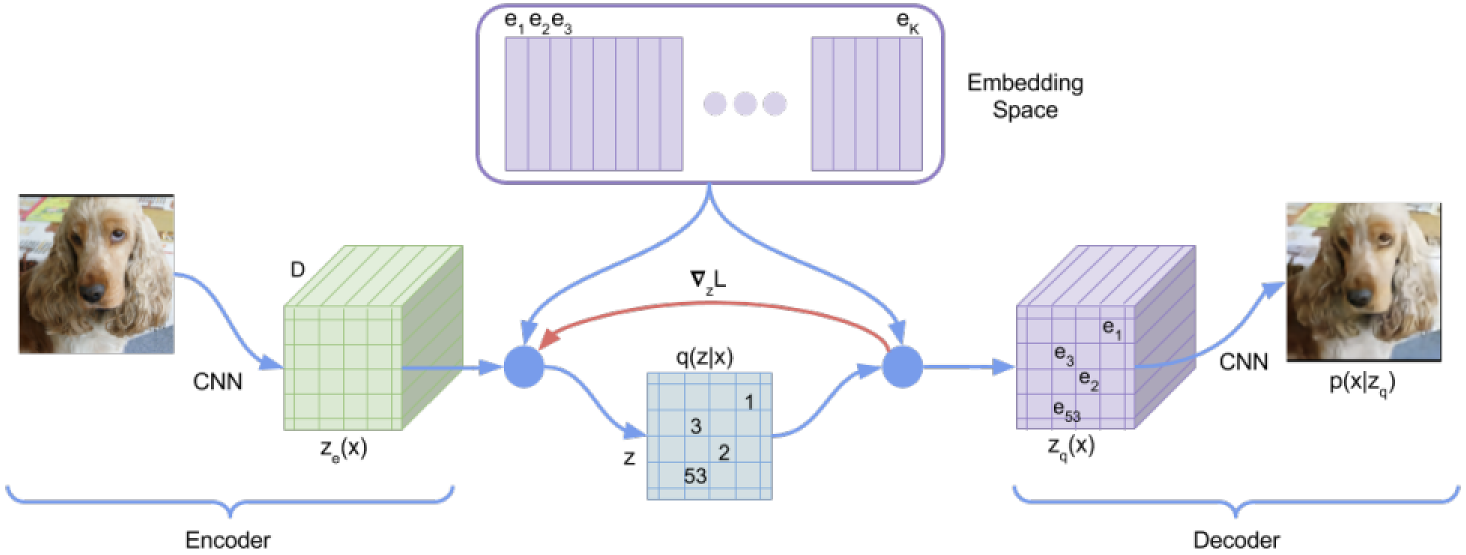
- input encoder maps to latent grid of embeddings *z*
- vector quantization maps embeddings to (nearest) code words e_i from $\{e_1..e_K\}$
- decoder maps from quantized grid to the reconstruction



Foundation for the DALL-E model

Vector Quantized Variational Autoencoder – VQ-VAE

- input encoder maps to latent grid of embeddings *z*
- vector quantization maps embeddings to (nearest) code words e_i from $\{e_1..e_K\}$
- decoder maps from quantized grid to the reconstruction

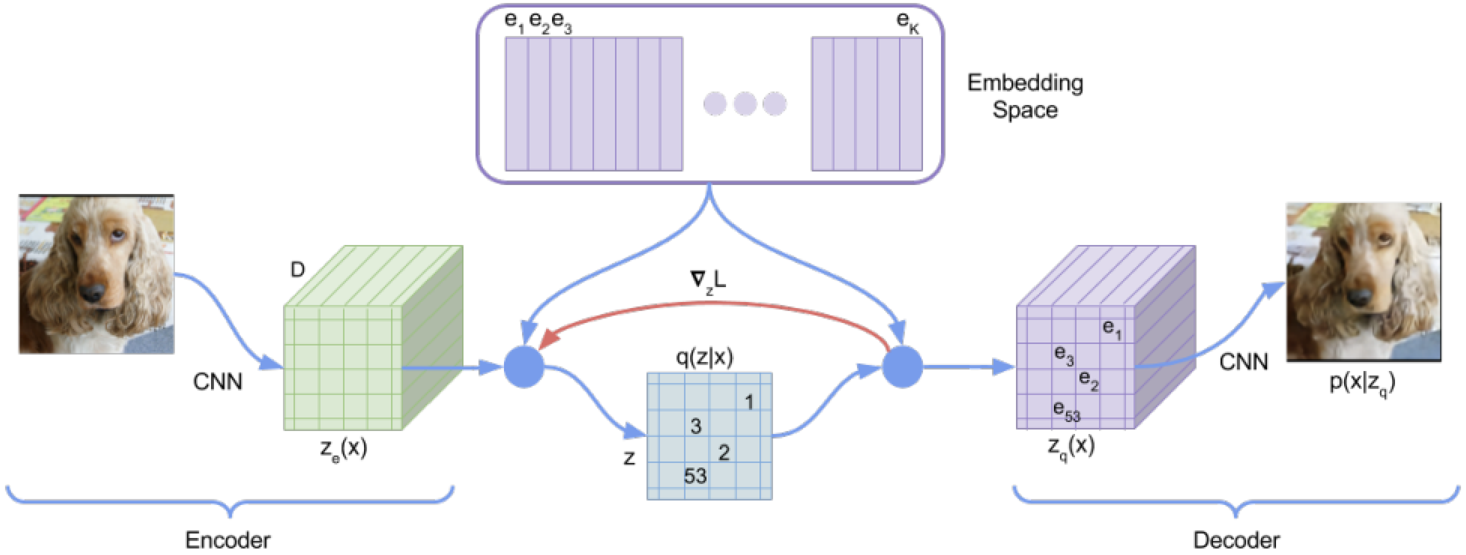


$$\text{Quantize}(E(\mathbf{x})) = \mathbf{e}_k \quad \text{where } k = \arg\min_j ||E(\mathbf{x}) - \mathbf{e}_j||$$

arXiv:1906.00446v1

Vector Quantized Variational Autoencoder – VQ-VAE

- input encoder maps to latent grid of embeddings *z*
- vector quantization maps embeddings to (nearest) code words e_i from $\{e_1..e_K\}$
- decoder maps from quantized grid to the reconstruction

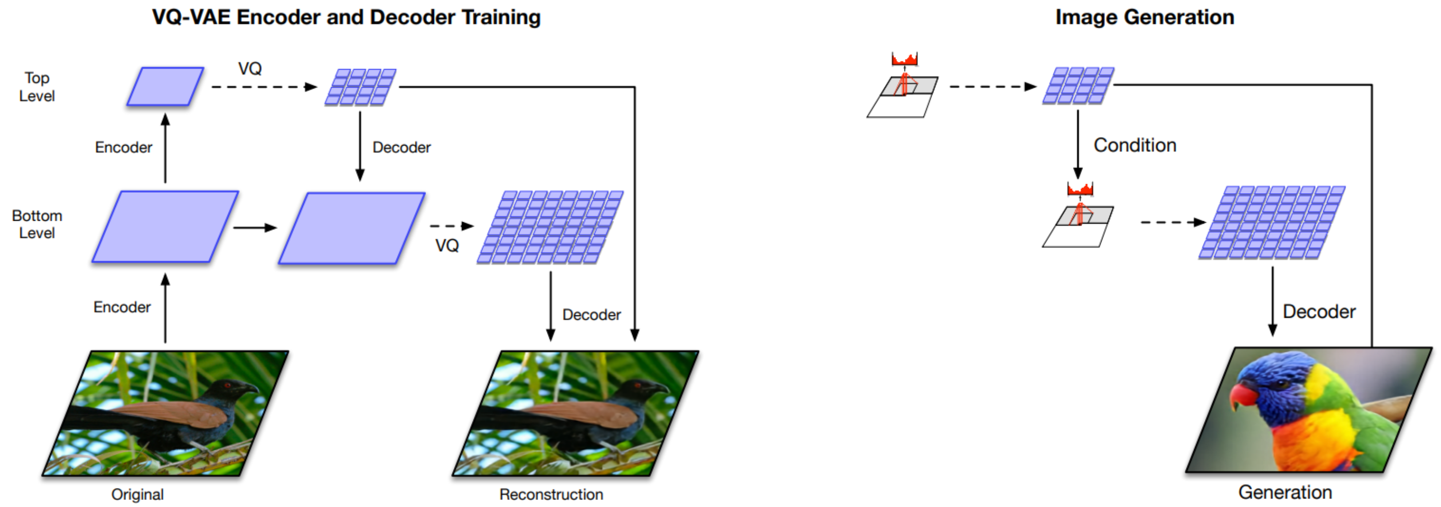


$$\text{Quantize}(E(\mathbf{x})) = \mathbf{e}_k \quad \text{where } k = \arg\min_j ||E(\mathbf{x}) - \mathbf{e}_j||$$

$$\mathcal{L}(\mathbf{x}, D(\mathbf{e})) = ||\mathbf{x} - D(\mathbf{e})||_2^2 + ||sg[E(\mathbf{x})] - \mathbf{e}||_2^2 + \beta||sg[\mathbf{e}] - E(\mathbf{x})||_2^2$$

arXiv:1906.00446v1

Vector Quantized Variational Autoencoder – VQ-VAE



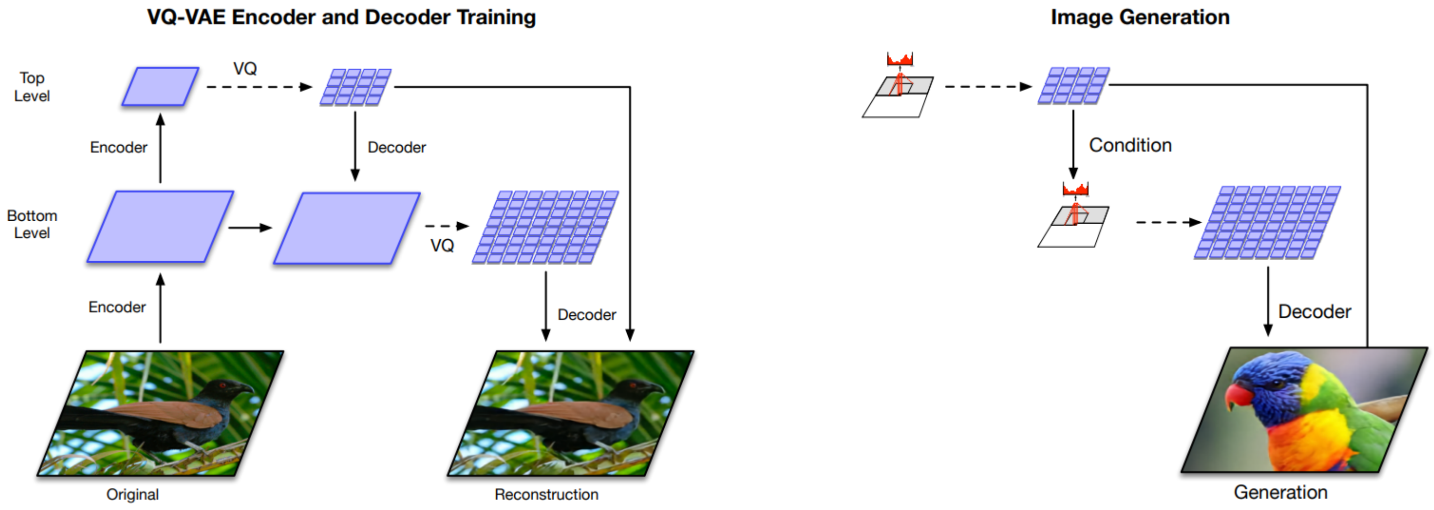
(a) Overview of the architecture of our hierarchical VQ-VAE. The encoders and decoders consist of deep neural networks. The input to the model is a 256 × 256 image that is compressed to quantized latent maps of size 64 × 64 and 32 × 32 for the *bottom* and *top* levels, respectively. The decoder reconstructs the image from the two latent maps.

(b) Multi-stage image generation. The top-level PixelCNN prior is conditioned on the class label, the bottom level PixelCNN is conditioned on the class label as well as the first level code. Thanks to the feed-forward decoder, the mapping between latents to pixels is fast. (The example image with a parrot is generated with this model).

Ca be made hierarchical

arXiv:1906.00446v1

Vector Quantized Variational Autoencoder – VQ-VAE



(a) Overview of the architecture of our hierarchical VQ-VAE. The encoders and decoders consist of deep neural networks. The input to the model is a 256×256 image that is compressed to quantized latent maps of size 64×64 and 32×32 for the *bottom* and *top* levels, respectively. The decoder reconstructs the image from the two latent maps.

(b) Multi-stage image generation. The top-level PixelCNN prior is conditioned on the class label, the bottom level PixelCNN is conditioned on the class label as well as the first level code. Thanks to the feed-forward decoder, the mapping between latents to pixels is fast. (The example image with a parrot is generated with this model).

Ca be made hierarchical

arXiv:1906.00446v1

Suggested tutorial – from Project MONAI

https://github.com/Project-MONAI

Project MONAI is an initiative started initially by NVIDIA and King's College London to establish an inclusive community of AI researchers to develop and exchange best practices for AI in healthcare imaging across academia and enterprise researchers. This collaboration has expanded to include academic and industry leaders throughout the medical imaging field.

Project MONAI has released multiple open-source PyTorch-based frameworks for annotating, building, training, deploying, and optimizing AI workflows in healthcare. These frameworks provide high-quality, user-friendly software that facilitates reproducibility and easy integration. With these tenants, researchers can share their results and build upon each other's work, fostering collaboration among academic and industry researchers.

VQVAE tutorial on medical data (diect open in colab link): https://colab.research.google.com/github/Project-MONAI/GenerativeModels/blob/main/tutorials/generative/2d_vqvae/2d_vqvae_tutorial.ipynb

Replace installation with

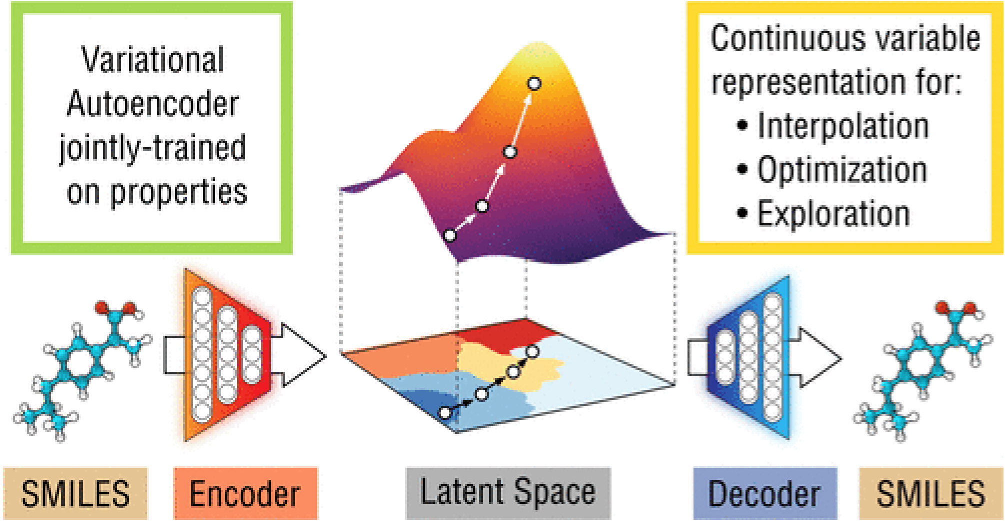
- `!pip install monai-generative`
- `%matplotlib inline`

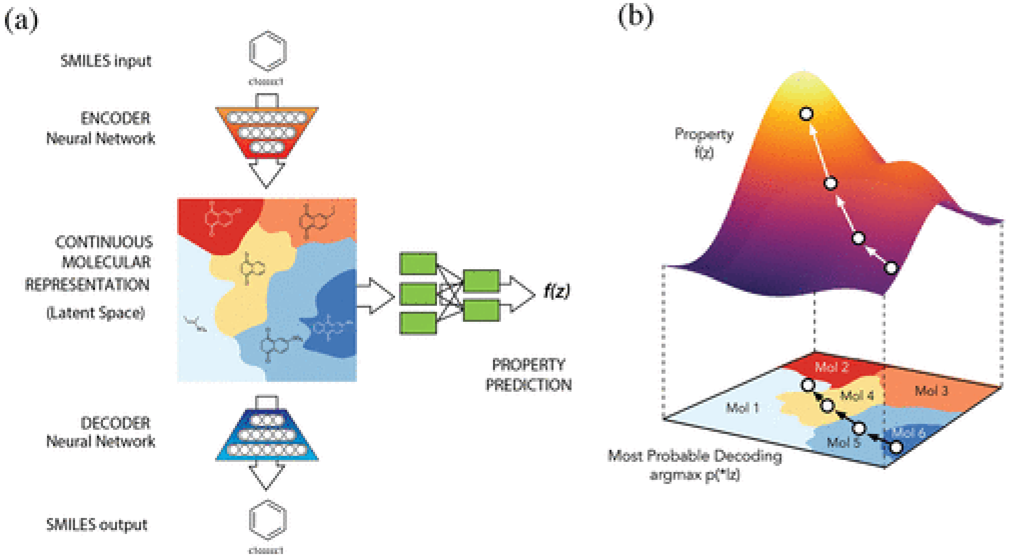
2. Examples

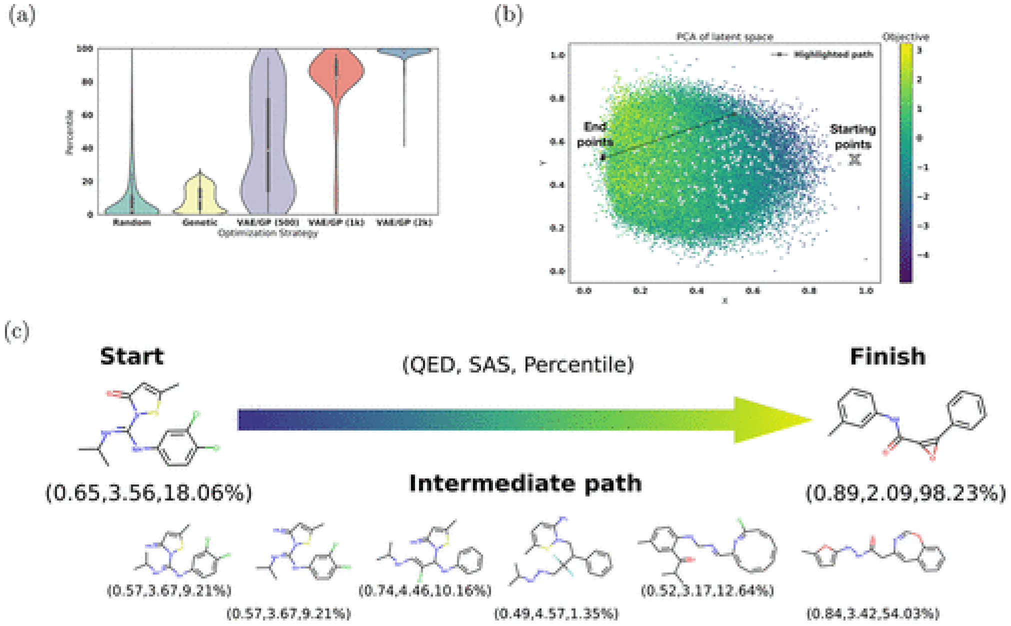
Where can be useful?

- Explore features
- Generate samples

Chemical Design with autoencoders


Variational Autoencoder jointly-trained on properties
Continuous variable representation for:
• Interpolation
• Optimization
• Exploration
SMILES
Encoder
Latent Space
Decoder
SMILES





Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules, R. Gómez-Bombarelli et al. *ACS Cent. Sci.* 2018, 4, 2, 268–276

Where can be useful?

- Explore features
- Generate samples

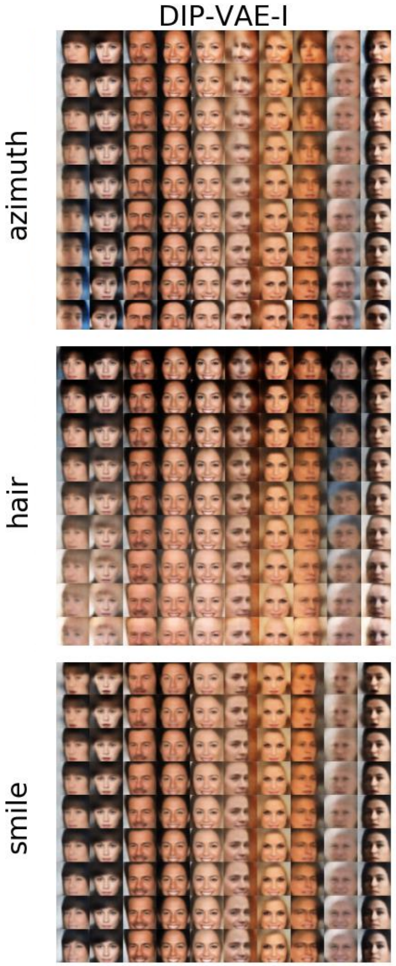
- Extract features
- Mix samples

Disentanglement of latent variables

Independent parameters can be often disentangled in the latent representations.

For example by:

- Distribution
- Mixing parameters
- etc



A. Kumar et al, 2017. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations.