Variational Autoencoders (VAE)

Lisa Herzog and Anthony Sonrel

Supervisor: Imran Fanaswala

Overview

- **Introduction**
- **Autoencoders**
- **Regularized latent space**
- **Variational Autoencoders**
- **Applications**

Introduction

- Variational Autoencoders (VAE) are special Neural Networks:
    a. Learn the most relevant features of the data
    b. Generate new data

Figure 1.1 Progress in human face generation
(Source: "The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation," by Miles Brundage et al., 2018, https://arxiv.org/abs/1802.07228.)

Introduction

State of the Art in generating data: https://thispersondoesnotexist.com/

- Do you spot some artefacts?

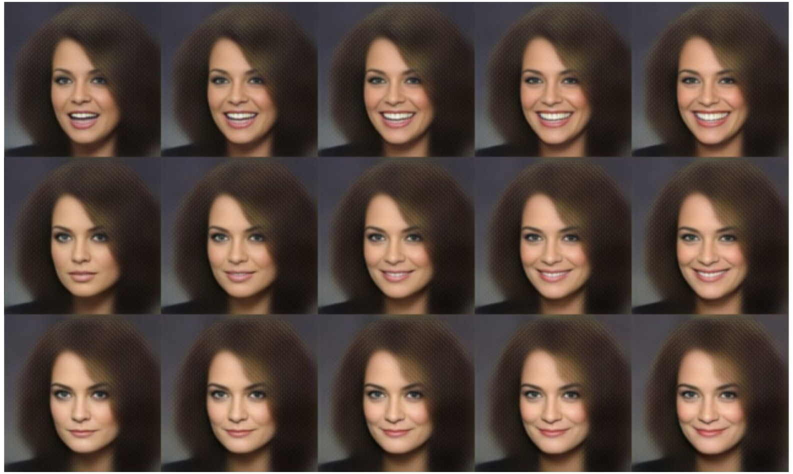
Introduction

Why to generate new data?

- For fun: Create people that do not exist
- Additional training data
- Video Gaming: Create variations of data
- Learn latent representations

  → Use features to initiate other neural networks

  → Make people smile



Autoencoders

Properties

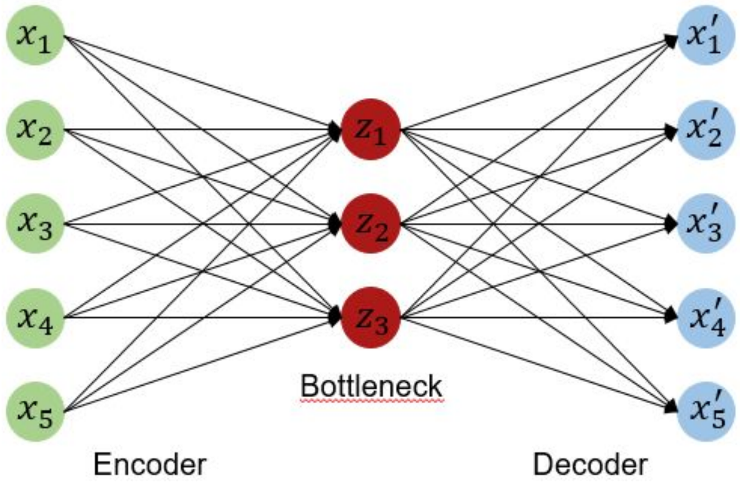
- Special neural network architectures:

  Encoder → Latent space → Decoder

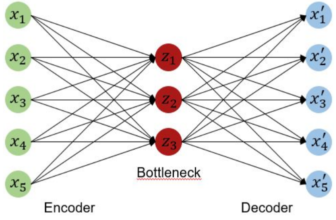
- Goal: Learn to reduce the data to its most relevant features (Latent space)

- Idea: Reconstruct the original image

  → Unsupervised (no labels necessary)



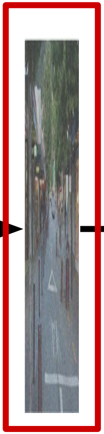
Autoencoders: Example



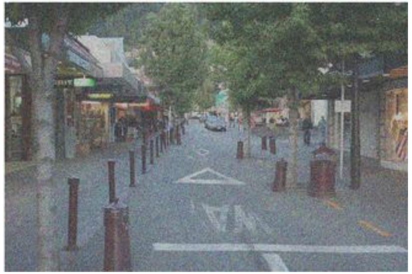
Shrink the input to lower dimension

Reconstruct the input image

→ Encoder →



→ Decoder →



Input

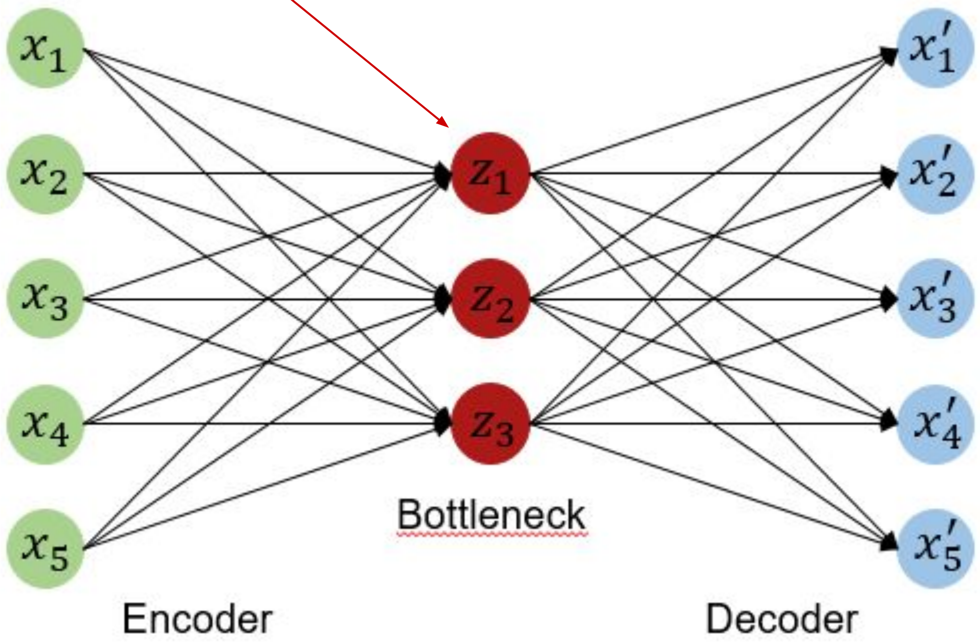
Latent space

Output

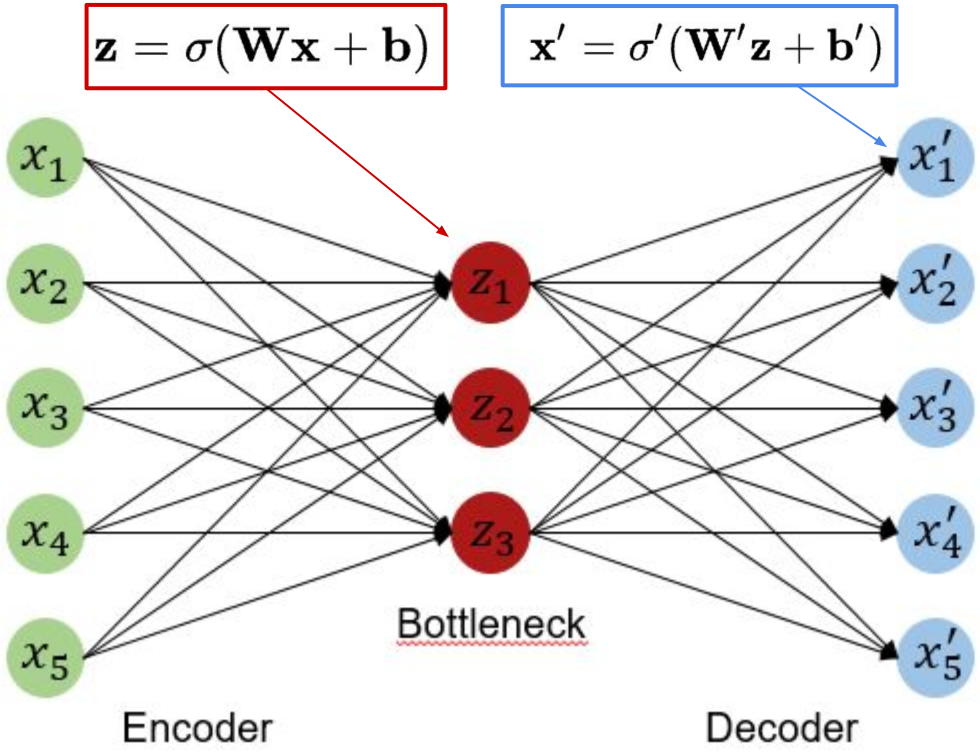
Most relevant features

Autoencoders

$$\boxed{\mathbf{z} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})}$$

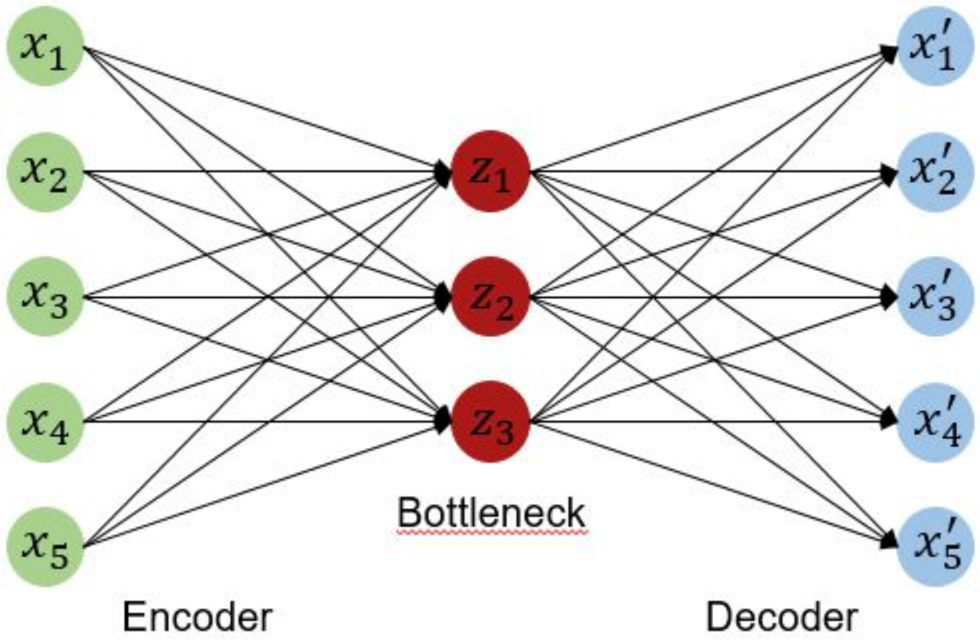


Autoencoders



Autoencoders

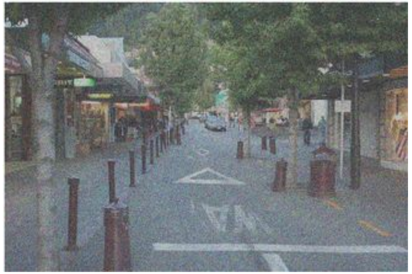
$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 = \|\mathbf{x} - \boxed{\sigma'(\mathbf{W}'\boxed{(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b}))} + \mathbf{b}')}\|^2$$



Autoencoders: Loss function

$$\mathcal{L}(\mathbf{x},\mathbf{x}') = \|\mathbf{x}-\mathbf{x}'\|^2 = \|\mathbf{x}-\boxed{\sigma'(\mathbf{W}'\boxed{(\sigma(\mathbf{W}\mathbf{x}+\mathbf{b})})+\mathbf{b}')}\|^2$$

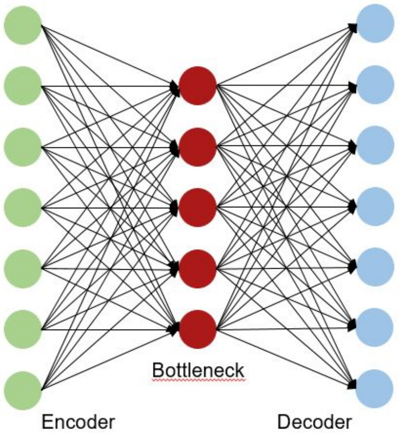
Reconstruction loss to be
minimized (via backpropagation)



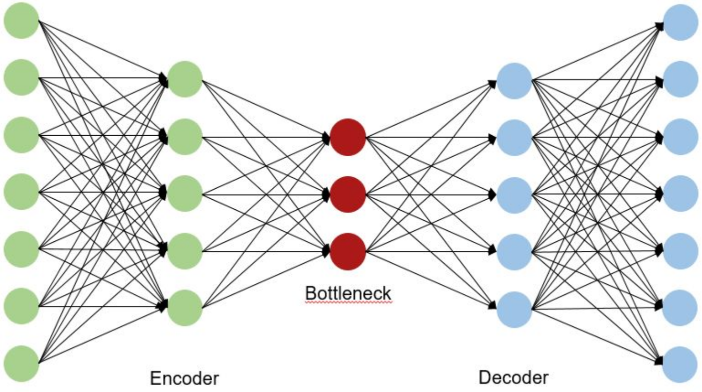
https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368

Autoencoders: Variants

1. Increase the capacity of the latent space



2. Increase the capacity of the encoder/decoder



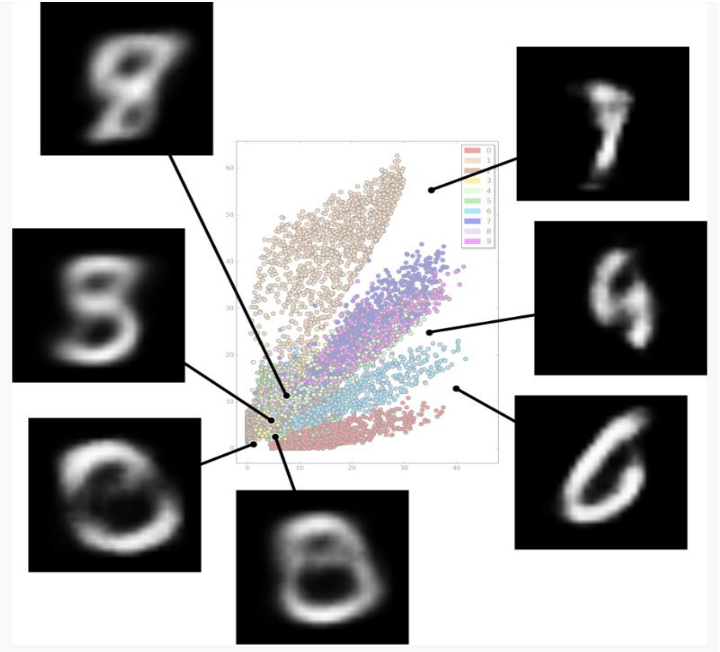
Autoencoders: Limitations

Latent space is not regularized

- Gaps in the latent space (feature space)
- Missing separability

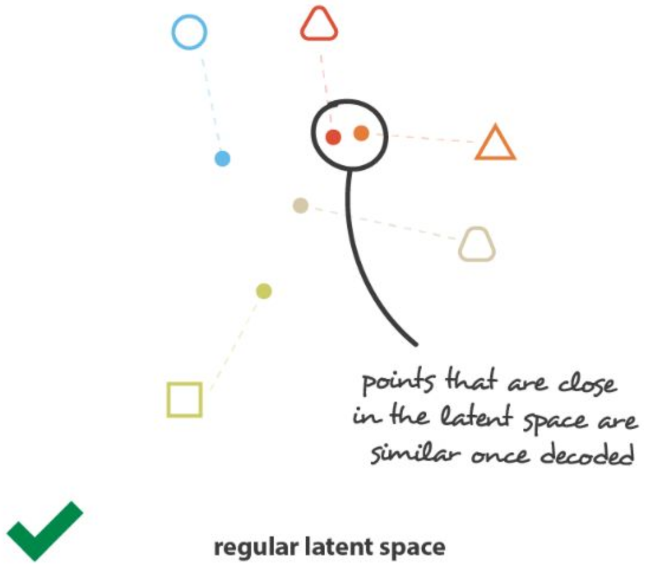
→ Only trained to have minimal loss, no matter

how the latent space looks like!



Regularized latent space

- Properties:
  - **Continuity:** Two close points are expected to give similar outputs
  - **Completeness:** A point sampled from the latent space should give meaningful outputs

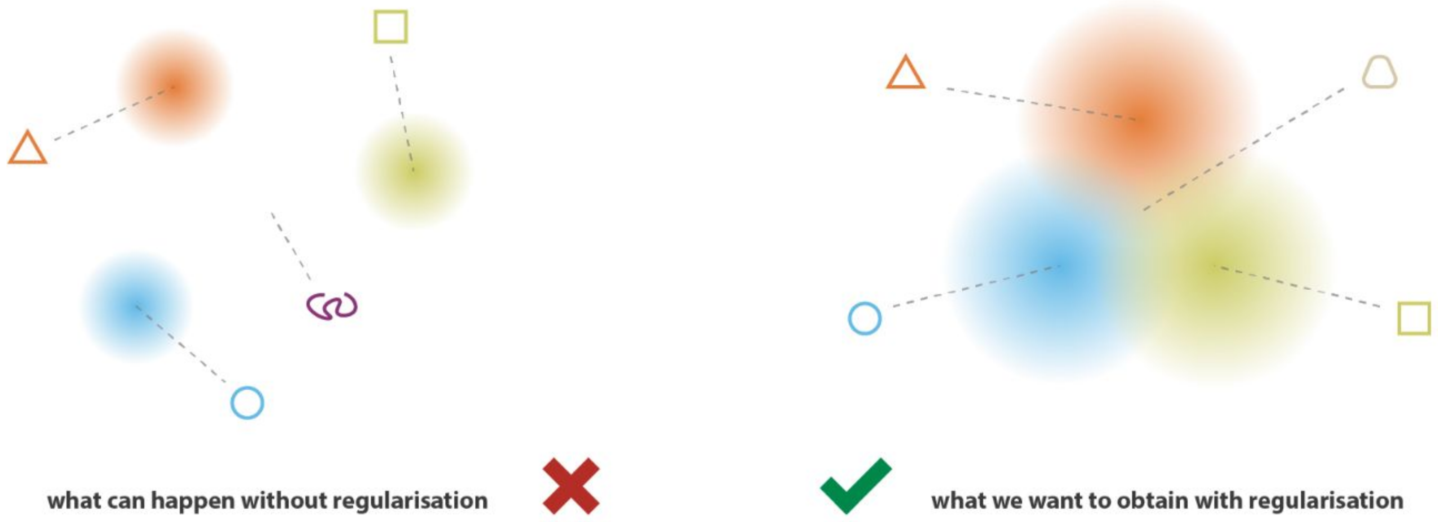


https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Regularized latent space

Map the input to distributions instead of points

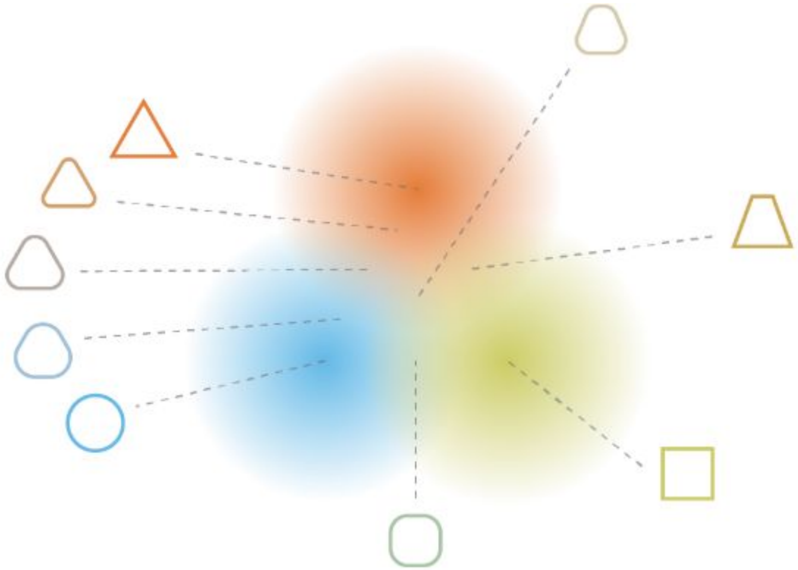
Regularize the distributions



https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Regularized latent space

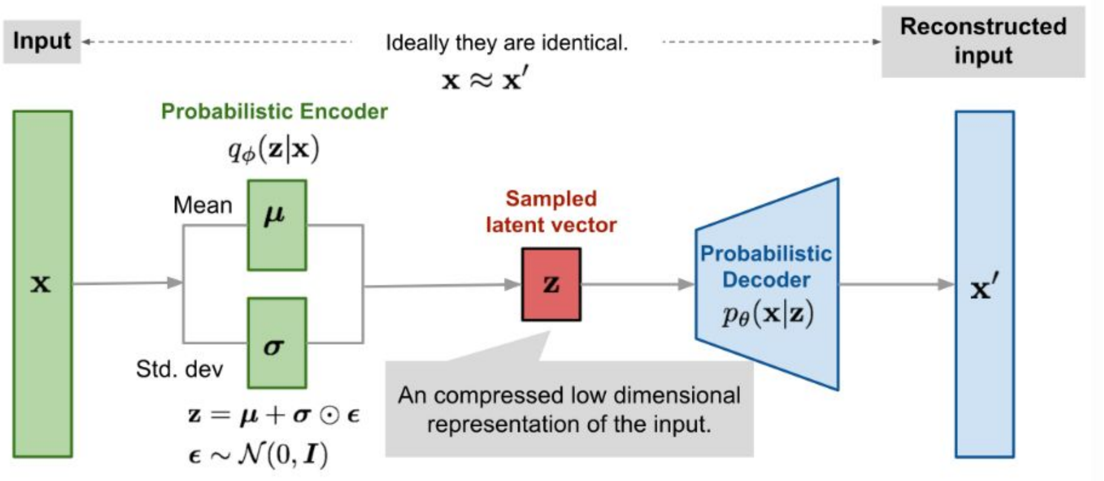
- Regularization enables to create some gradient over the information in the latent space
- Points halfway between means of distributions are decoded in something in between



https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Variational Autoencoders (VAE)

1. **Encoder:** Learn to map the input to a (latent) distribution instead of a vector
2. **Latent space:** Sample a latent vector from this distribution over the latent space
3. **Decoder:** Generate output with similar characteristics using the sampled latent vector



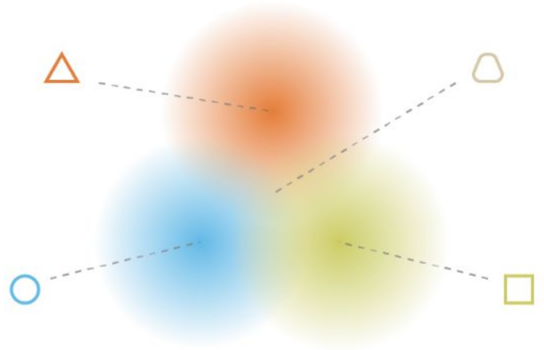
VAE: Intuition

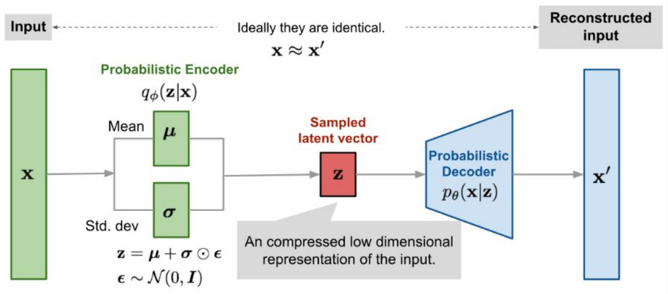
Regularize the latent space:

- Prevent punctual distributions (variance)
- Prevent distributions too far apart (mean)

→ Encoder is trained to return the mean and variance of a Gaussian distribution

→ Distributions are enforced to be close to standard normal N(0,1)



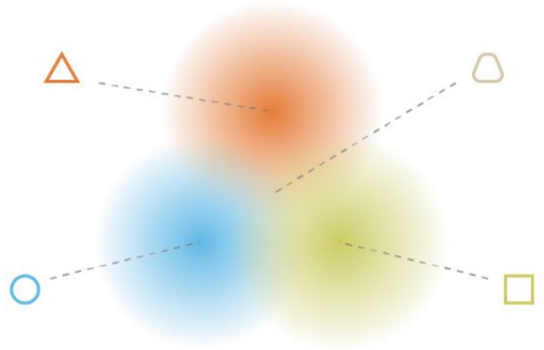


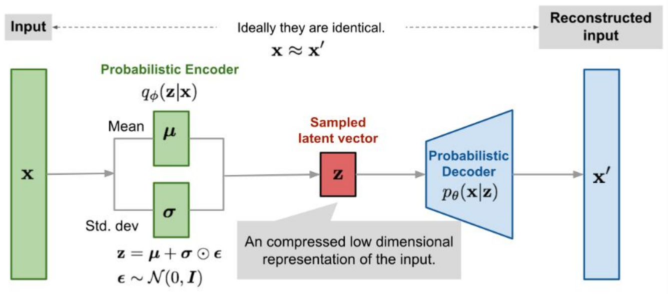
VAE: Loss function

Regularized reconstruction loss:

$$L(x, x') = ||x - x'||^2 + KL(N(\mu, \sigma^2), N(0, 1))$$

- Reconstruction loss: Good reconstruction
- Kullback Leibler (KL) Divergence: Regularization



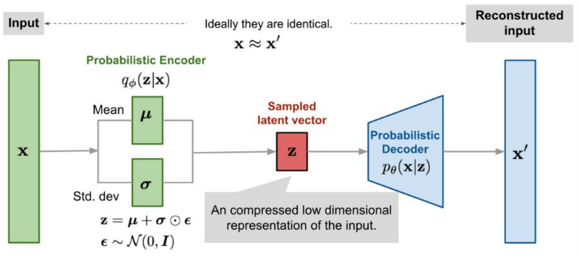


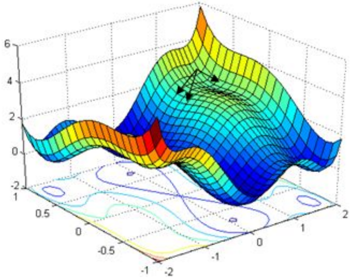
VAE: Training

Training: Stochastic Gradient Descent

1. Provide a batch of images to the model
2. Calculate the loss function
3. Update the parameters in the direction of the steepest descent of the loss using backpropagation

Forward pass





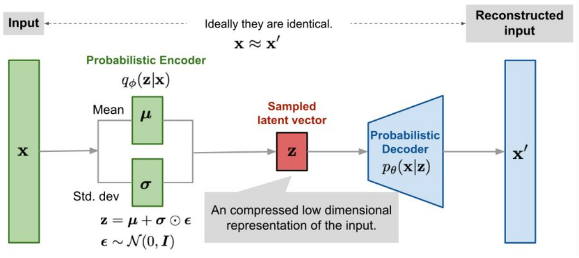
Backward pass

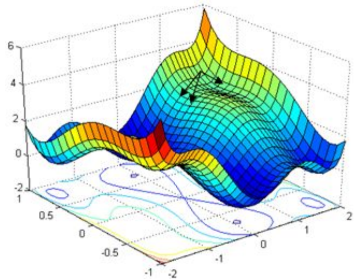
VAE: Training

Training: Stochastic Gradient Descent

1. Provide a batch of images to the model
2. Calculate the loss function
3. Update the parameters in the direction of the steepest descent of the loss using backpropagation

Forward pass



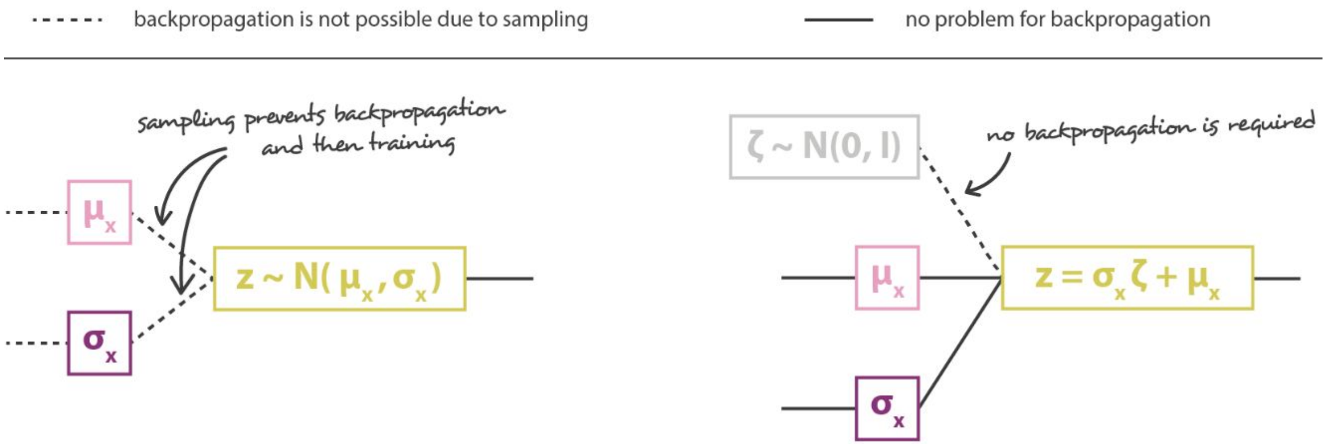


Backward pass

VAE: Reparameterization Trick

Trick:

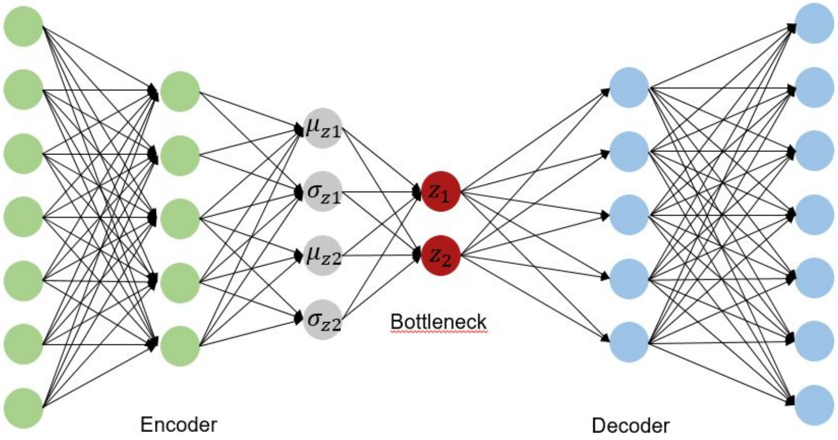
$$\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\boldsymbol{I})$$
$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$$

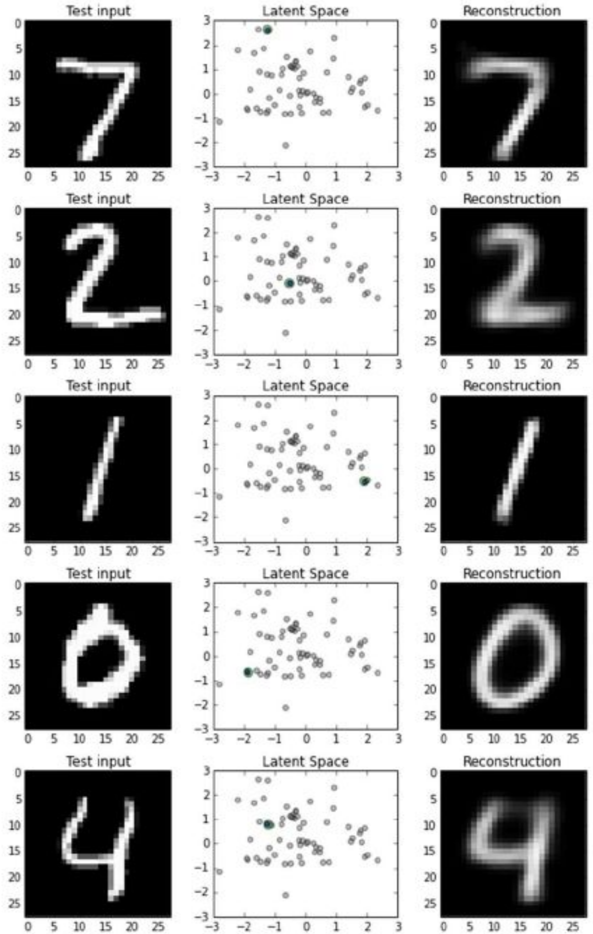


VAE: Example

Fully connected network

- two hidden layer
- two-dimensional latent space

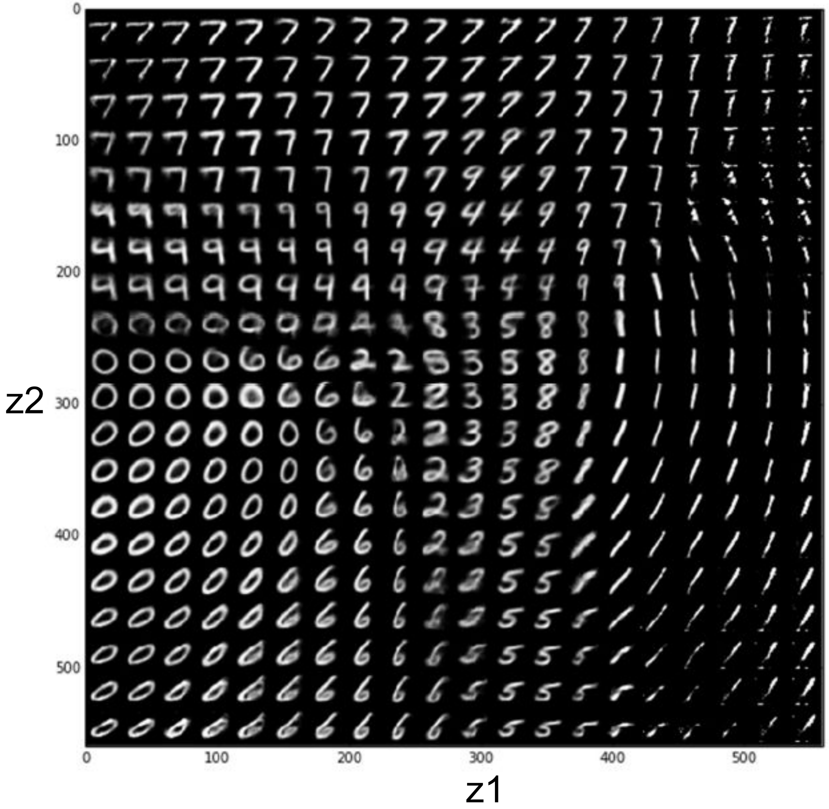




https://github.com/oduerr/dl_tutorial/blob/master/tensorflow/vae/vae_demo.ipynb

VAE: Example

Moving in the latent space



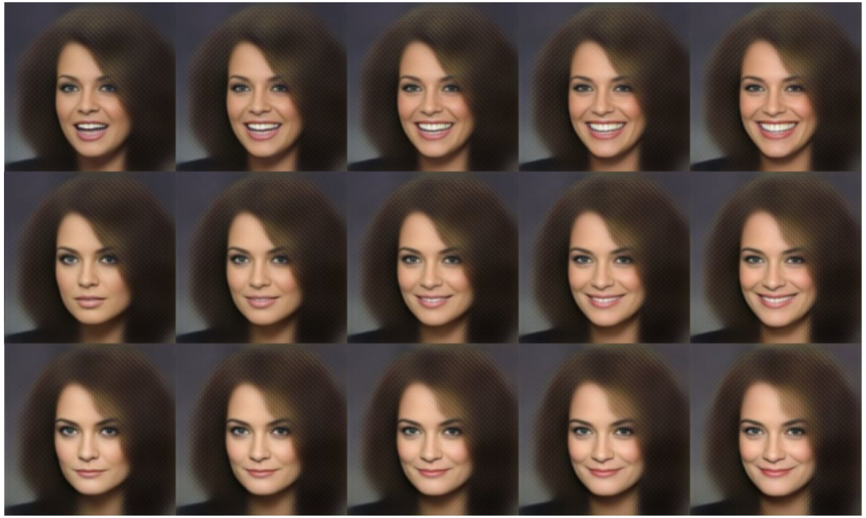
https://github.com/oduerr/dl_tutorial/blob/master/tensorflow/vae/vae_demo.ipynb

VAE: Examples

Applications: **Image (re)generation**

- extraction of the *smile* and

  mouth open vectors



White 2016, arXiv

VAE vs. Autoencoders

Autoencoders:

- Try to reconstruct the input image
- Learn features to initialize supervised learning methods

→ Not used so much anymore

Variational Autoencoders:

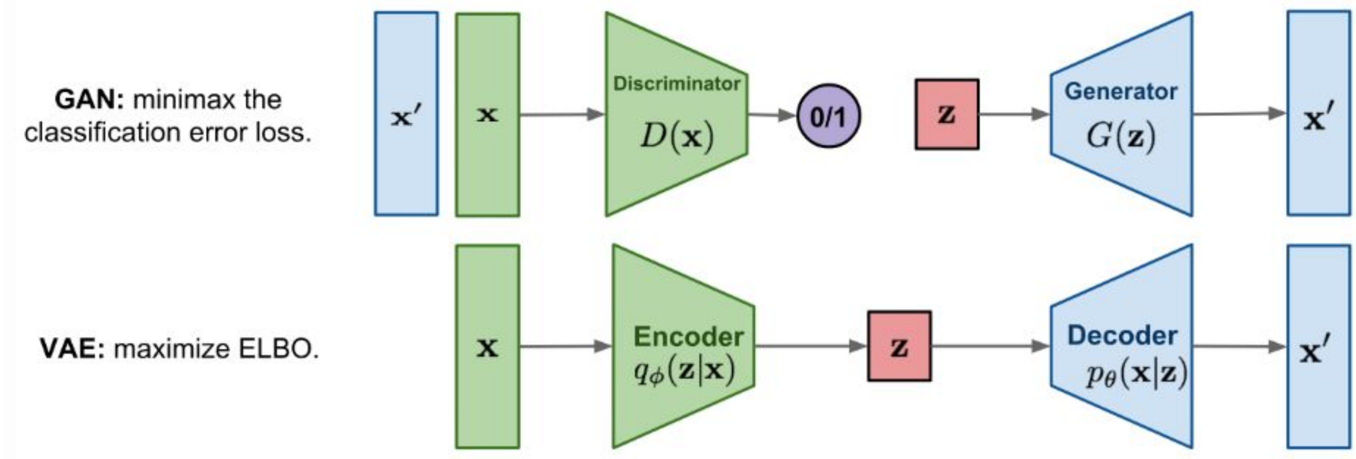
- Sample from a model (Bayesian) to generate new data

→ Oftentimes replaced by Generative Adversarial Networks (GAN)

Outlook: GANs

Minmax Game

- **Discriminator: Distinguish real from fake examples**
- **Generator: Create fake examples**



References

https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

http://cs231n.stanford.edu/

Deep Learning: generative models (O. Dürr, HTWG, Lecture)

Appendix

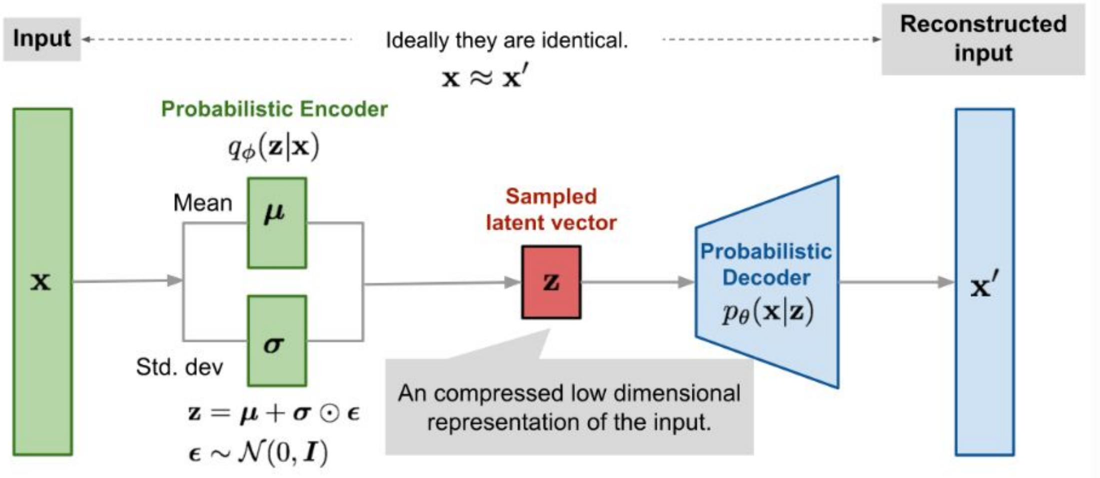
Some more information about the math behind VAEs

Main source: http://cs231n.stanford.edu/

VAE: The math behind

Variational Autoencoder:

1. **Probabilistic Encoder:** Approximate the distribution of the latent space given the data: $\boxed{q_\phi(\mathbf{z}|\mathbf{x})}$
2. **Probabilistic Decoder:** Likelihood of the data given the sampled latent variable: $\boxed{p_\theta(\mathbf{x}|\mathbf{z})}$



VAE: The math behind

Goal: Maximum Likelihood estimation of the parameters of the data generating distribution

$$\theta^* = \arg\max_\theta \prod_{i=1}^{N} p_\theta(x^{(i)})$$
$$= \arg\max_\theta \sum_{i=1}^{N} \log p_\theta(x^{(i)})$$

Problem: Intractable integral

$$p_\theta(x^{(i)}) = \int p_\theta(x^{(i)}, z)dz = \int p_\theta(x^{(i)} \mid z)p_\theta(z)dz$$

VAE: The math behind

Variational Inference:

- Used to approximate the true posterior (which is unknown!)
- Consider a parameterized distribution $q_\phi(\mathbf{z}|\mathbf{x})$ and find the parameters which approximate the true posterior best (e.g. family of Gaussians with parameters mean and variance)

VAE: The math behind

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z \sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Logarithms)}$$

$$= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi) \ \text{“Elbow”}} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}$$

Lower bound (lbow) of the likelihood, which is called “evidence” (E) = Elbow

VAE: The math behind

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Logarithms)}$$

$$= \underbrace{\boxed{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right]} - \boxed{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}}_{\mathcal{L}(x^{(i)},\theta,\phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}$$

"Elbow"

Reconstruct the input data
(Decoder)

Latent state should follow the prior
(Encoder)

VAE: The math behind

Solution: Maximize the lower bound (ELBOW) in VAEs

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)}, \theta, \phi)$$

Variational lower bound (elbow)

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)}, \theta, \phi)$$

Training: Maximize lower bound