UNIVERSITY OF COPENHAGEN

Unsupervised Learning-2

Elements of Machine Learning, 2021

Raghavendra Selvan

raghav@di.ku.dk
Machine Learning Section
@raghavian

Overview

Lecture-1

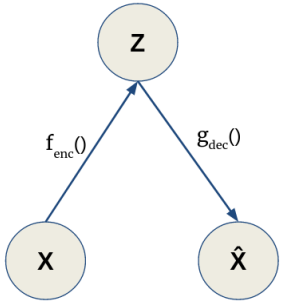
- Unsupervised learning
- PCA
- K-means clustering

Lecture-2

- Autoencoders
- Variational inference
- Variational Autoencoders

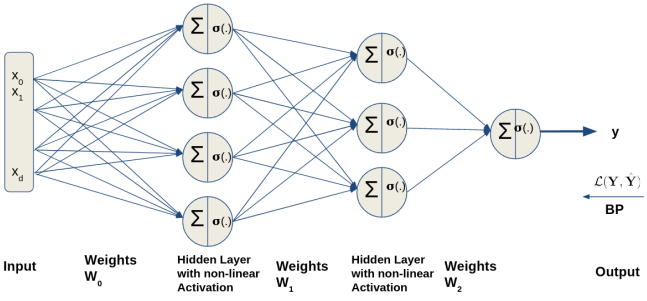
Autoencoders

- PCA is a linear dimensionality reduction method
- Autoencoders: (possibly) Non-linear PCA
- Neural Network based comprising encoder-decoder pair
- Undercomplete, Regularized, Sparse, Denoising AEs
- Compression, dimensionality reduction



Graphical model view of Autoencoders

Basics: Multi-Layer Perceptron (MLP)



A First Deep Learning model

$$\mathbf{h}^{(\ell+1)} = MLP(\mathbf{h}^{(\ell)}) = \sigma\left(\mathbf{W}_\ell^T \mathbf{h}^{(\ell)}\right)$$

with $\mathbf{h}^{(\ell)} = \mathbf{x}$ and $\mathbf{h}^{(L)} = \mathbf{y}$

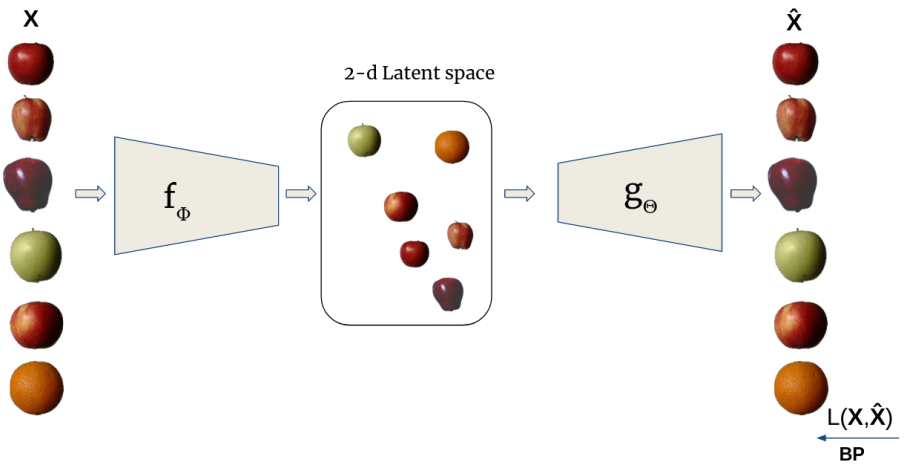
- Flexible, supervised model
- Parameterised function approximator $f_\phi(\cdot)$
- Backpropagation based optimization
- Non-linearities are crucial!
- Number of hidden layers (depth)
- Number of hidden units/layer (width)

Autoencoders

Encoder: $f_\phi(\cdot) : \mathbf{x} \in \mathbb{R}^F \rightarrow \mathbf{z} \in \mathbb{R}^D$
Decoder: $g_\theta(\cdot) : \mathbf{z} \in \mathbb{R}^D \rightarrow \hat{\mathbf{x}} \in \mathbb{R}^F$,

- Undercomplete Autoencoders: $\mathcal{L}\big(\mathbf{x}, g(f(\mathbf{x}))\big)$
  where $\mathcal{L}(\cdot)$ is a loss function penalizing $g(f(\mathbf{x})) \neq \mathbf{x}$
- Regularized Autoencoders: $\mathcal{L}\big(\mathbf{x}, g(f(\mathbf{x}))\big) + \Omega(\mathbf{z})$
  where $\Omega(\mathbf{z})$ is a regularization penalty
- Denoising Autoencoders: $\mathcal{L}\big(\mathbf{x}, g(f(\tilde{\mathbf{x}}))\big)$
  where $\tilde{\mathbf{x}}$ is $\mathbf{x}$ corrupted with some form of *stochastic* noise

Autoencoders in practice



Undercomplete Autoencoder

Optimizing the Autoencoder: An MLE perspective

- Likelihood for Autoencoders: $p(\hat{\mathbf{X}}|\mathbf{X};\theta,\phi)$
- θ,ϕ are optimized to maximize the likelihood
- Maximum likelihood estimation (MLE)
- Continuous values: Mean Squared Error is commonly used
- Categorical values: Cross entropy loss

We derive the cross-entropy objective for binary classes

Optimizing the Autoencoder: An MLE perspective

Consider the training data: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \quad : \mathbf{x}_i \in \{0, 1\}$,
predictions: $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_N\} \quad : \hat{\mathbf{x}}_i \in [0, 1]$
Modelling $\hat{\mathbf{x}}_i$ as a Bernoulli random variable, i.e $\hat{\mathbf{x}}_i$ takes value of 1 with probability p_i and 0 with probability $(1 - p_i)$
The maximum likelihood objective is:

$$\arg\max_{\theta,\phi} p(\hat{\mathbf{X}}|\mathbf{X}; \theta, \phi)$$

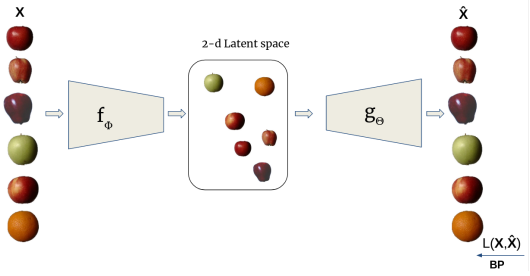
Using i.i.d assumption of data and modifying the objective to log domain:

$$\arg\max_{\theta,\phi} \log p(\hat{\mathbf{X}}|\mathbf{X}; \theta, \phi) \equiv \arg\min_{\theta,\phi} -\frac{1}{N} \sum_{i=1}^{N} \log p(\hat{\mathbf{x}}_i|\mathbf{x}_i)$$

$$= \arg\min_{\theta,\phi} -\frac{1}{N} \sum_{i=1}^{N} \left[(\mathbf{x}_i) \log p_i + (1 - \mathbf{x}_i) \log(1 - p_i)\right]$$

Applying Autoencoders

- Undercomplete autoencoders $D < F$
- Non-linear low dimensional representation
- Clustering, Compression
- Feature extraction
- Similarity of data points
- Snapshot of the data *manifold*



What else can the latent space be used for?

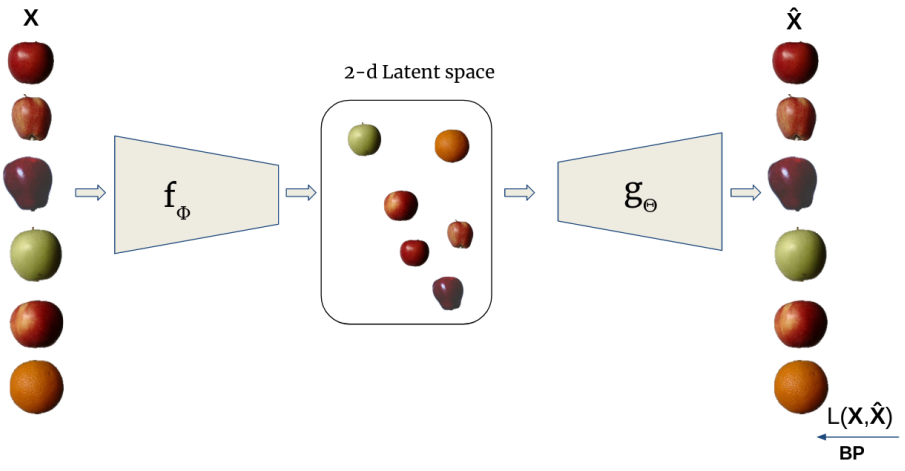
Regularizing Autoencoders

- Latent space can be unstructured
- Discontinuities in the space
- Forcing structure with regularization
- Stochastic Autoencoders

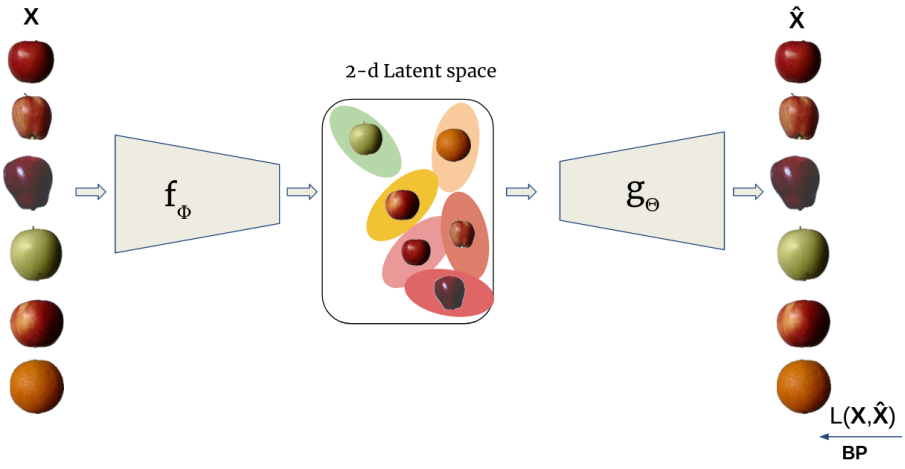
Idea

Instead of embedding data as points, embed them as probability densities

Regularizing the latent space of Autoencoders



Regularizing the latent space of Autoencoders



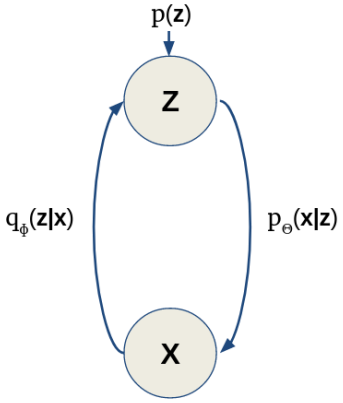
Variational Autoencoders (VAE)

- Regularized stochastic Autoencoders
- Inspired and driven by Approximate Variational Inference
- Encoder and decoder are probabilistic
- Data is mapped to a probability density
- Deep **Generative** model

VAE Objective

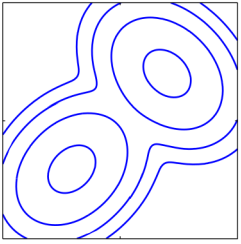
As we are in probabilistic setting:

- We can estimate the data distribution $P_X(\mathbf{x})$ or $p(\mathbf{x})$ (for convenience)
- Estimation from observed data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$
- Probabilistic Encoder: $q_\phi(\mathbf{z}|\mathbf{x}) \approx p(\mathbf{z}|\mathbf{x})$
- Probabilistic Decoder: $p_\theta(\mathbf{x}|\mathbf{z})$
- Prior on latent variables: $p(\mathbf{z})$



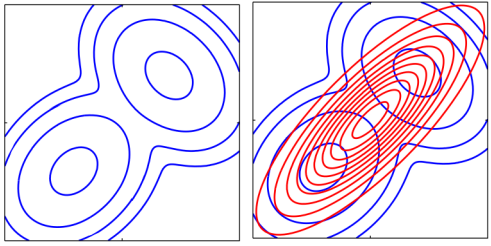
Basics before VAE: Density Measures

- Density Measures
- No straightforward distances
- Divergence measures
- True distribution $p(\mathbf{x})$ approximated with a *simpler* $q(\mathbf{x})$
- Usually $q(\mathbf{x})$ is from exponential family or factorised distribution



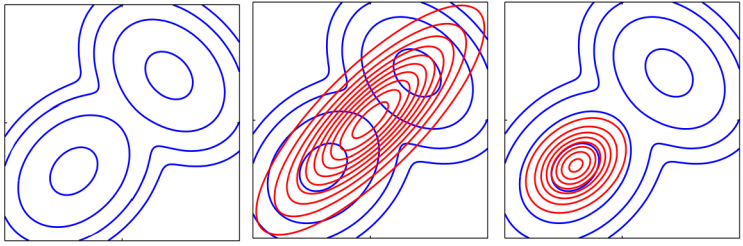
Basics before VAE: Density Measures

- Density Measures
- No straightforward distances
- Divergence measures
- True distribution $p(\mathbf{x})$ approximated with a *simpler* $q(\mathbf{x})$
- Usually $q(\mathbf{x})$ is from exponential family or factorised distribution



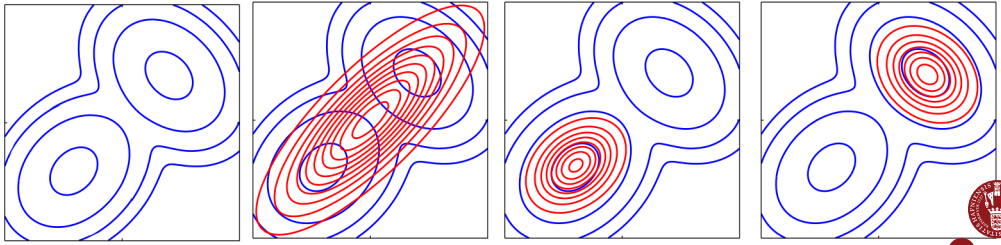
Basics before VAE: Density Measures

- Density Measures
- No straightforward distances
- Divergence measures
- True distribution $p(\mathbf{x})$ approximated with a *simpler* $q(\mathbf{x})$
- Usually $q(\mathbf{x})$ is from exponential family or factorised distribution



Basics before VAE: Density Measures

- Density Measures
- No straightforward distances
- Divergence measures
- True distribution $p(\mathbf{x})$ approximated with a *simpler* $q(\mathbf{x})$
- Usually $q(\mathbf{x})$ is from exponential family or factorised distribution



Basics before VAE: Alpha Divergence

- Family of divergence measures
- Parameterised by α
- Based on Expected mutual information
- $\alpha \to 1$ yields Forward KL
- $\alpha \to 0$ yields Reverse KL

$$D_\alpha(p||q) = \frac{\int \alpha p(\mathbf{x}) + (1-\alpha) q(\mathbf{x}) - p(\mathbf{x})^\alpha q(\mathbf{x})^{1-\alpha}}{\alpha(1-\alpha)}$$

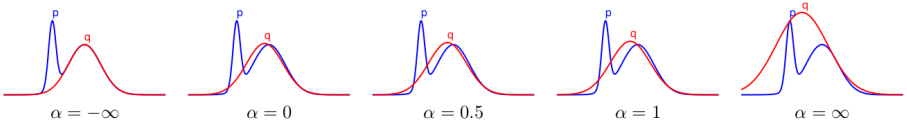


Figure 1: The Gaussian q which minimizes α-divergence to p (a mixture of two Gaussians), for varying α. $\alpha \to -\infty$ prefers matching one mode, while $\alpha \to \infty$ prefers covering the entire distribution.

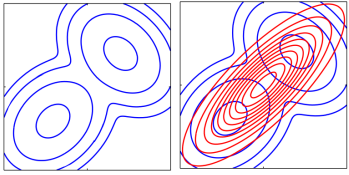
Fig. 1 from Minka, 2005

Basics before VAE: KL Divergence

Forward KL divergence

$$\mathrm{KL}(p(\mathbf{x})||q(\mathbf{x})) = \mathbb{E}_{p_x}\left[\log \frac{p(\mathbf{x})}{q(\mathbf{x})}\right]$$

$\mathbb{E}_{p_x}$ is the expectation operator wrt $p(\mathbf{x})$.



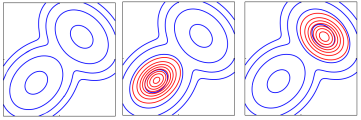
Notice the support of $q(\mathbf{x})$ (red)

Basics before VAE: KL Divergence

Reverse KL divergence

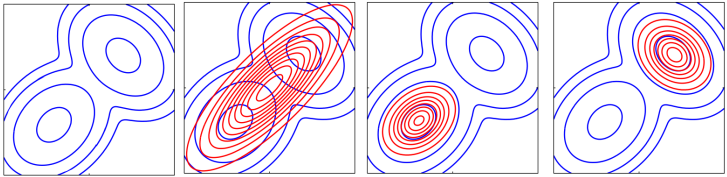
$$\mathrm{KL}(q(\mathbf{x})||p(\mathbf{x})) = \mathbb{E}_{q_x}\left[\log \frac{q(\mathbf{x})}{p(\mathbf{x})}\right]$$

$\mathbb{E}_{q_x}$ is the expectation operator wrt $q(\mathbf{x})$. This is the Variational objective.



Notice the support of $q(\mathbf{x})$ (red)

Basics before VAE: KL Divergence



Observe the support of $q(\mathbf{x})$ (red) in each case.

Deriving the VAE objective

- The true latent posterior distribution $p(\mathbf{z}|\mathbf{x})$ is intractable in most cases
- The VAE encoder approximates this with $q_\phi(\mathbf{z}|\mathbf{x})$
- We minimize the reverse KL divergence $\mathrm{KL}(q(\mathbf{x})||p(\mathbf{x}))$

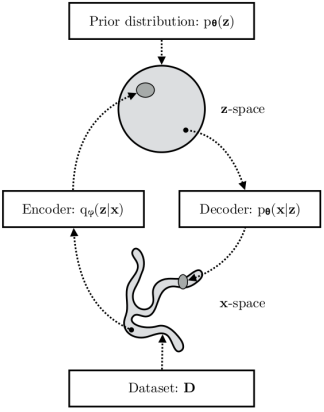


Fig 2.1 from Kingma & Welling, 2019

Deriving the VAE objective

We start with the motivation of approximating the latent posterior distribution with a variational distribution.
Minimizing the *reverse* KL divergence yields,

$$\mathrm{KL}\Big[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})\Big] = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})}\Big] = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log \Big(\frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\frac{p(\mathbf{x})}{p(\mathbf{x}|\mathbf{z})}\Big)\Big]$$

Due to Bayes' Rule.

$$\mathrm{KL}\Big[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})\Big] = -\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log p(\mathbf{x}|\mathbf{z})\Big] + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\Big] + \log p(\mathbf{x})$$

Note the second term is $\mathrm{KL}\Big[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\Big]$.

The first two terms form the Evidence Lower Bound (ELBO).

$$\mathcal{L}(p(\mathbf{x})) = -\Big(\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log p(\mathbf{x}|\mathbf{z})\Big] - \mathrm{KL}\Big[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\Big]\Big)$$

Deriving the VAE objective

Rewriting the VAE objective with ELBO

$$\mathrm{KL}\Big[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})\Big] = -\mathcal{L}(p(\mathbf{x})) + \log p(\mathbf{x})$$

And, the ELBO becomes equal to $\log p(\mathbf{x})$ when $\mathrm{KL}\Big[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})\Big] = 0$ which is the objective of the VAE encoder!

That is, minimizing $\mathrm{KL}\Big[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})\Big]$ is equivalent to maximizing ELBO.

This yields the surrogate VAE objective,

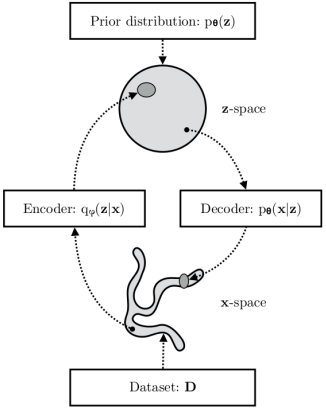
$$\mathcal{L}_{\text{VAE}} = -\mathcal{L}(p(\mathbf{x})) = -\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\Big[\log p(\mathbf{x}|\mathbf{z})\Big] + \mathrm{KL}\Big[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\Big]$$

where we have negated ELBO to obtain a minimization objective.

VAE objective

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] + \text{KL}\Big[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\Big]$$

- Autoencoder with regularisation
- First term is reconstruction loss
- Second term is regularisation penalty
- Regularisation forces the encoder to match the prior $p(\mathbf{z})$



VAEs in Practice

- Prior is a standard Gaussian $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$
- Encoder is a diagonal Gaussian $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi, \boldsymbol{\sigma}^2_\phi)$
- Expectations are approximated using Monte carlo sampling For instance, $\mathcal{L}_{\text{VAE}}$ has the reconstruction loss:

$$-\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z})\Big] \approx -\frac{1}{B}\sum_{n=1}^{B} \log p_\theta(\mathbf{x}|\mathbf{z}^{(n)})$$

where $\mathbf{z}^{(n)} \sim q_\phi(\mathbf{z}|\mathbf{x})$ and B is the batch size.

Reparameterization Trick

- Gradient of ELBO computation has stochastic elements
- Reparameterization trick by transforming $\mathbf{z} = g(\boldsymbol{\epsilon}, \phi, \mathbf{x})$This yields:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi, \boldsymbol{\sigma}^2_\phi) = \boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \cdot \boldsymbol{\epsilon}$$

where $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$

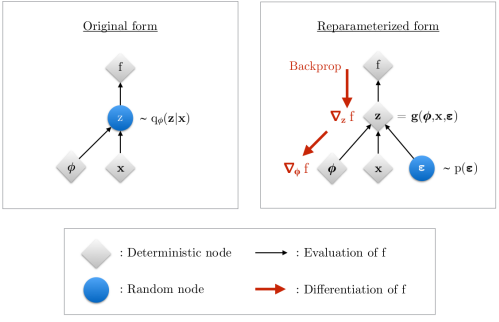
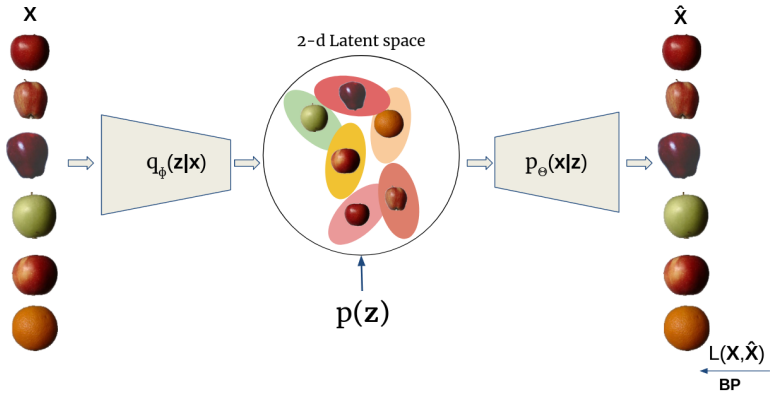


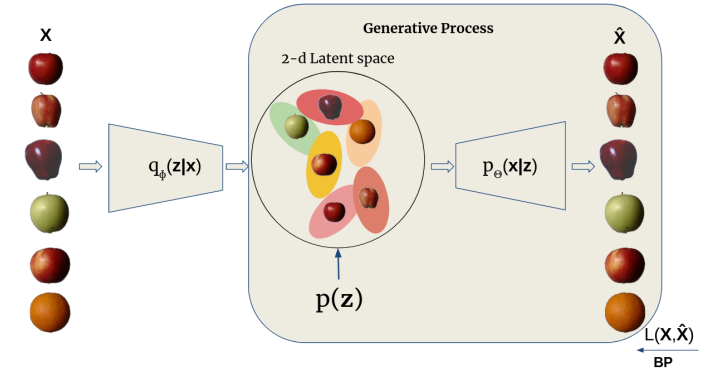
Fig 2.3 from Kingma & Welling, 2019

VAEs in Practice



Generative modelling using VAEs

- Primary difference between Autoencoders
- Exploiting structure of latent space
- Sample from prior $\rightarrow$ Decode samples
- Once trained, VAEs can be used to generate data



VAE summary

+ Probabilistic Autoencoder
+ Latent space regularisation with prior
+ Structure of latent space exploited for generative sampling
+ Feature disentanglement
- Difficult to train for high dimensional data
- Mode collapse
- Gaussian approximations

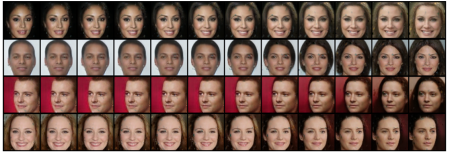


Figure 6: Interpolation between the two generated images (64×64) by moving in the latent space.

Khan et al. 2018

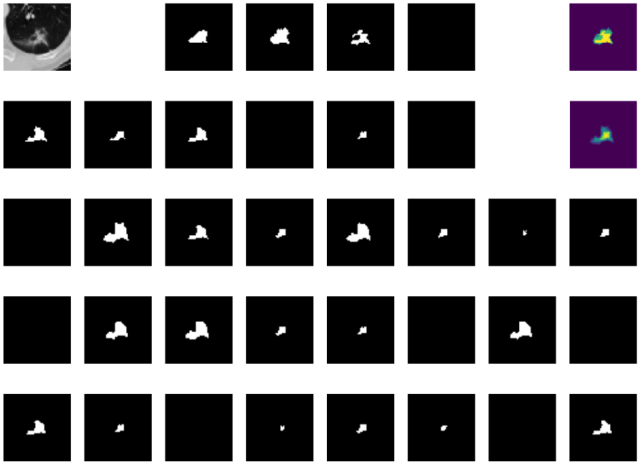
Summary

- Increasing applications of Unsupervised learning
- Self-supervision is a promising strategy
- Latent representations are useful, insightful
- Generative models are here to stay (Normalizing flows, GANs)

Figure 5: Linear interpolation in latent space between real images

Kingma et al. 2018

Questions



Post on Absalon!