"Metode de optimizare Riemanniene pentru învățare profundă"
Proiect cofinanțat din Fondul European de Dezvoltare Regională prin
Programul Operațional Competitivitate 2014-2020

Variational AutoEncoder: An Introduction and Recent Perspectives

Luigi Malagò, Alexandra Peste, and Septimia Sarbu

Romanian Institute of Science and Technology

Politecnico di Milano

23 February 2018

Outline

1. Plan of the presentation
2. General View of Variational Autoencoders
   - Introduction
   - Research Directions
3. Work-in-progress
   - Using Gaussian Graphical Models
   - Geometry of the Latent Space
4. Future Work

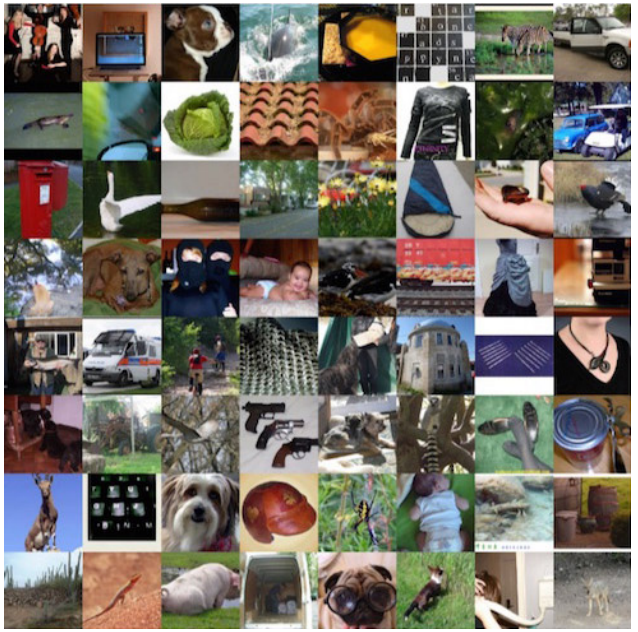
Plan of the presentation

- **Overview** of **variational inference in deep learning**: general algorithm, research directions and fast review of the existing literature
- Present some of the work we have done so far:
  - the use of **graphical models for introducing correlations** between the latent variables
  - analysis of the **geometry of the latent space**
- Present some **ideas and questions** that we have been thinking about, along with possible research directions

Outline

1. Plan of the presentation
2. General View of Variational Autoencoders
   - Introduction
   - Research Directions
3. Work-in-progress
   - Using Gaussian Graphical Models
   - Geometry of the Latent Space
4. Future Work

Generative Models in Deep Learning



Real images

Generated images [12]

Notations for Bayesian Inference

- $\mathbf{X}, \mathbf{Z}$ multivariate random variables, $\mathbf{Z}$ continuous, with probability density functions (pdf) $p_\theta(\mathbf{x})$ and $p_\theta(\mathbf{z})$ respectively, with parameters θ; $p_\theta(\mathbf{z})$ is the *prior* and $p_\theta(\mathbf{x})$ the *marginal*;

- $p_\theta(\mathbf{x}, \mathbf{z})$: pdf of the joint random variable $(\mathbf{X}, \mathbf{Z})$, with parameters θ;

- $p_\theta(\mathbf{x}|\mathbf{z}), p_\theta(\mathbf{z}|\mathbf{x})$: pdfs of the random variables $\mathbf{X}|\mathbf{Z} = \mathbf{z}$ and $\mathbf{Z}|\mathbf{X} = \mathbf{x}$; $p_\theta(\mathbf{z}|\mathbf{x})$ is the *posterior*.

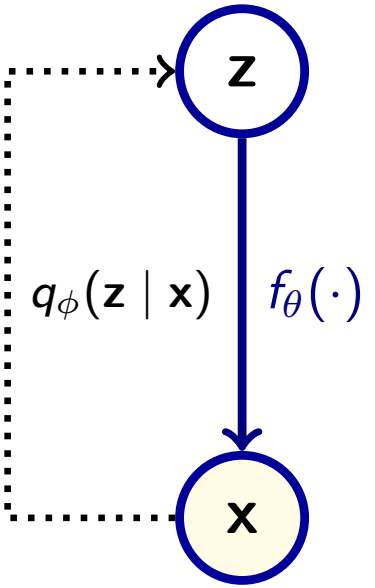
General Setting

Formulation of the problem: The continuous latent r.v. $\mathbf{Z}$ generates $\mathbf{X}$, through $f_\theta(\cdot)$ a differentiable function, such that $\int p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}$ is intractable.

The goal is *inference*, i.e., finding $p_\theta(\mathbf{z}|\mathbf{x})$.

Variational inference [1] approximates the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ with $q_\phi(\mathbf{z}|\mathbf{x})$, by minimizing the Kullback-Leibler divergence $\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x}))$.

Approach to the solution: maximizing a lower bound of the log likelihood.



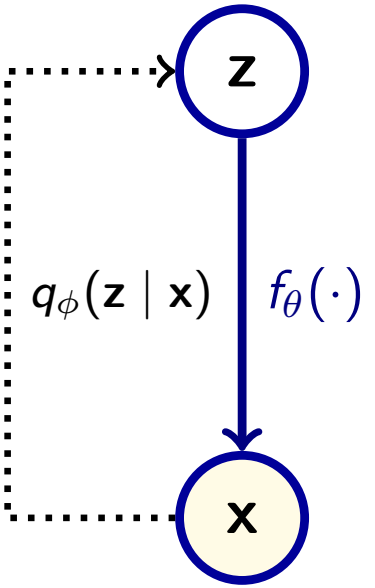
General Setting

Formulation of the problem: The continuous latent r.v. $\mathbf{Z}$ generates $\mathbf{X}$, through $f_\theta(\cdot)$ a differentiable function, such that $\int p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}$ is intractable.

The goal is *inference*, i.e., finding $p_\theta(\mathbf{z}|\mathbf{x})$.

Variational inference [1] approximates the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ with $q_\phi(\mathbf{z}|\mathbf{x})$, by minimizing the Kullback-Leibler divergence $\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x}))$.

Approach to the solution: maximizing a lower bound of the log likelihood.



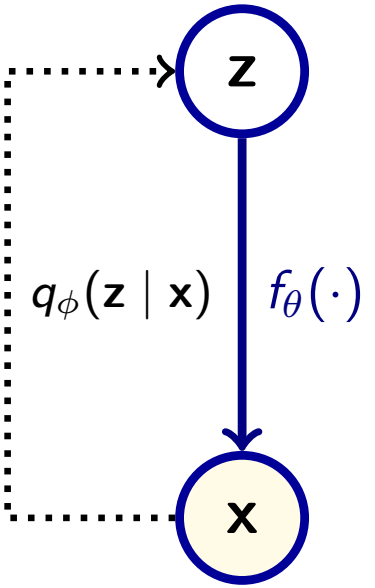
General Setting

Formulation of the problem: The continuous latent r.v. $\mathbf{Z}$ generates $\mathbf{X}$, through $f_\theta(\cdot)$ a differentiable function, such that $\int p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}$ is intractable.

The goal is *inference*, i.e., finding $p_\theta(\mathbf{z}|\mathbf{x})$.

Variational inference [1] approximates the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ with $q_\phi(\mathbf{z}|\mathbf{x})$, by minimizing the Kullback-Leibler divergence $\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x}))$.

Approach to the solution: maximizing a lower bound of the log likelihood.



Variational Inference I

Deriving the lower bound:

$$\ln p_\theta(\mathbf{x}) = \ln \int q_\phi(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \mathbf{dz} \geq \int q_\phi(\mathbf{z}|\mathbf{x}) \ln \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \mathbf{dz} \text{ (Jensen's inequality)}$$

$$\textbf{Evidence lower bound: } \mathcal{L}(\theta,\phi;\mathbf{x}) := \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \leq \ln p_\theta(\mathbf{x})$$

Minimizing KL $\iff$ maximizing the lower-bound:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \ln p_\theta(\mathbf{x}) - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x}))$$

The maximum of the lower-bound is the log-likelihood, and it is obtained when $\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x})) = 0$. Thus, the problems are equivalent.

Variational Inference I

Deriving the lower bound:

$$\ln p_\theta(\mathbf{x}) = \ln \int q_\phi(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \mathbf{dz} \geq \int q_\phi(\mathbf{z}|\mathbf{x}) \ln \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \mathbf{dz} \text{ (Jensen's inequality)}$$

$$\textbf{Evidence lower bound: } \mathcal{L}(\theta,\phi;\mathbf{x}) := \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \leq \ln p_\theta(\mathbf{x})$$

Minimizing KL $\iff$ maximizing the lower-bound:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \ln p_\theta(\mathbf{x}) - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x}))$$

The maximum of the lower-bound is the log-likelihood, and it is obtained when $\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x})) = 0$. Thus, the problems are equivalent.

Variational Inference II

- Optimizing the lower bound maximizes the log likelihood . The distribution of $\mathbf{X}$ can be approximated with *importance sampling*:

$$\ln p_\theta(\mathbf{x}) \approx \ln \frac{1}{S} \sum_{i=1}^{S} \frac{p_\theta(\mathbf{x}|\mathbf{z^{(i)}}) p_\theta(\mathbf{z^{(i)}})}{q_\phi(\mathbf{z^{(i)}}|\mathbf{x})}$$

  where $\mathbf{z^{(i)}} \sim q_\phi(\cdot|\mathbf{x})$.

- Fixing the family of distributions for the r.v., e.g. we assume they are Gaussians, we move from variational calculus to regular optimization of the parameters. The problem becomes:

$$\max_{\theta,\phi} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}, \mathbf{z}) - \ln q_\phi(\mathbf{z}|\mathbf{x})]$$

Variational Inference II

- Optimizing the lower bound maximizes the log likelihood . The distribution of $\mathbf{X}$ can be approximated with *importance sampling*:

$$\ln p_\theta(\mathbf{x}) \approx \ln \frac{1}{S} \sum_{i=1}^{S} \frac{p_\theta(\mathbf{x}|\mathbf{z^{(i)}}) p_\theta(\mathbf{z^{(i)}})}{q_\phi(\mathbf{z^{(i)}}|\mathbf{x})}$$

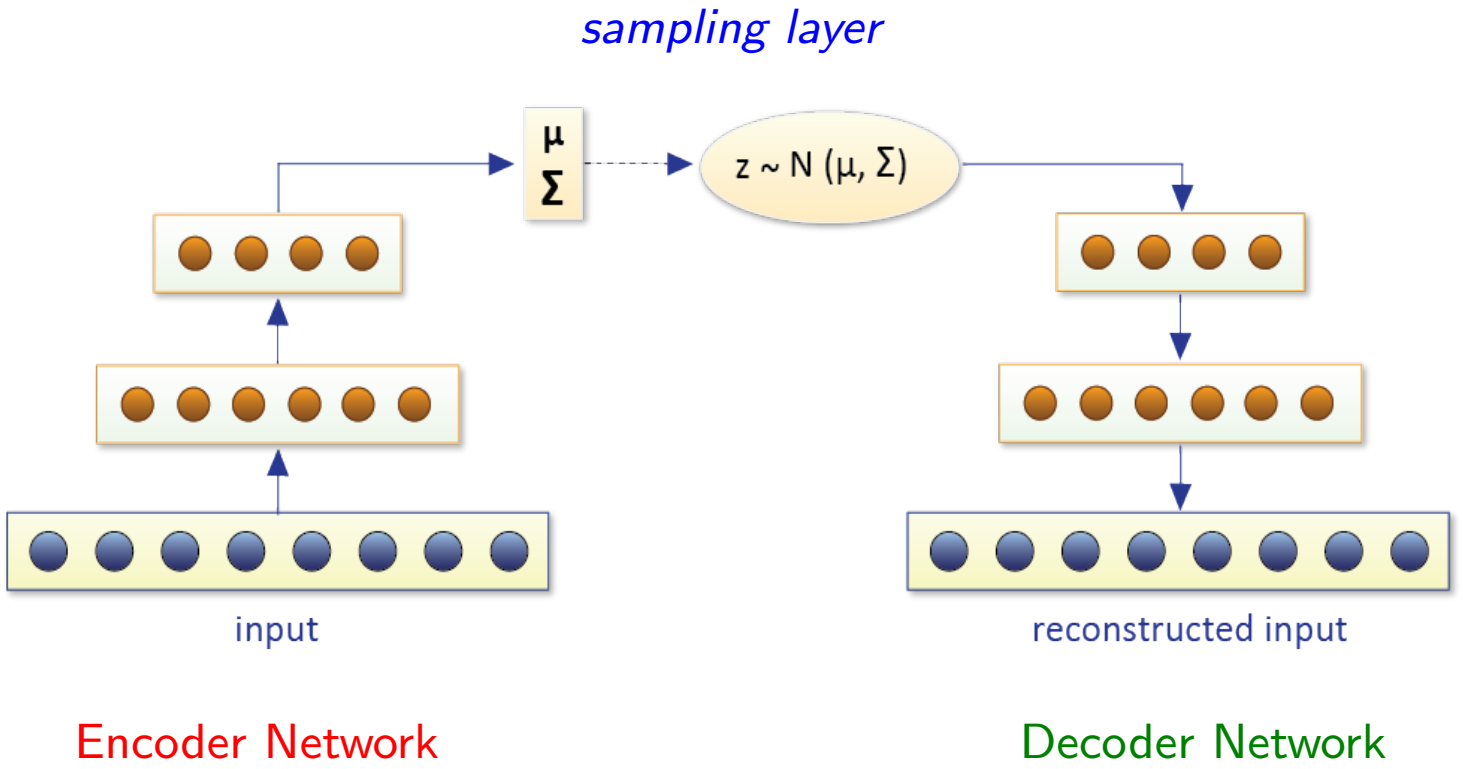
  where $\mathbf{z^{(i)}} \sim q_\phi(\cdot|\mathbf{x})$.

- Fixing the family of distributions for the r.v., e.g. we assume they are Gaussians, we move from variational calculus to regular optimization of the parameters. The problem becomes:

$$\max_{\theta,\phi} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}, \mathbf{z}) - \ln q_\phi(\mathbf{z}|\mathbf{x})]$$

Variational Autoencoders

Variational Autoencoders ([6], [11]) tackle the problem of *variational inference* in the context of *neural networks*. The parameters ϕ and θ of $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ are learned through two different neural networks: encoder and decoder.



Applications

- **Encode**: learn a lower dimensional representation of the dataset, by sampling from $q_\phi(\cdot|\mathbf{x})$.
  The dimension of the latent variable $\mathbf{Z}$ is assumed to be much smaller than the dimension of the dataset.

- **Generate** from noise examples that resemble the ones seen during training. The prior $p_\theta(\mathbf{z})$ on the latent variable is assumed Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and samples are fed through the network to output the conditional probabilities $p_\theta(\mathbf{x} \mid \mathbf{z})$.

Applications

- **Encode**: learn a lower dimensional representation of the dataset, by sampling from $q_\phi(\cdot|\mathbf{x})$.
  The dimension of the latent variable $\mathbf{Z}$ is assumed to be much smaller than the dimension of the dataset.

- **Generate** from noise examples that resemble the ones seen during training. The prior $p_\theta(\mathbf{z})$ on the latent variable is assumed Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and samples are fed through the network to output the conditional probabilities $p_\theta(\mathbf{x} \mid \mathbf{z})$.

Details of the Algorithm

- **Encoder**: $q_\phi(\mathbf{z}|\mathbf{x})$ - Gaussian $\mathcal{N}(\mu, \mathbf{D})$ with diagonal covariance;
  ϕ - the set of parameters of the encoder
- **Decoder**: $p_\theta(\mathbf{x}|\mathbf{z})$ - Gaussian with diagonal covariance (continuous data) or Bernoulli vector (discrete data);
  θ - the set of parameters of the decoder
- For a data point $\mathbf{x}$, rewrite the lower bound $\mathcal{L}(\theta, \phi; \mathbf{x})$

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{Reconstruction error}} - \underbrace{KL(q_\phi(\mathbf{z}|\mathbf{x}) \,||\, p_\theta(\mathbf{z}))}_{\text{Regularization}}$$

Cost function to be optimized: $\frac{1}{N}\sum_{n=1}^{N} \mathcal{L}(\theta, \phi; \mathbf{x}_n)$, from dataset $\mathbf{X} = \{\mathbf{x}_n\}_{n=\overline{1,N}}$

Details of the Algorithm

- **Encoder**: $q_\phi(\mathbf{z}|\mathbf{x})$ - Gaussian $\mathcal{N}(\mu, \mathbf{D})$ with diagonal covariance;
  ϕ - the set of parameters of the encoder
- **Decoder**: $p_\theta(\mathbf{x}|\mathbf{z})$ - Gaussian with diagonal covariance (continuous data) or Bernoulli vector (discrete data);
  θ - the set of parameters of the decoder
- For a data point $\mathbf{x}$, rewrite the lower bound $\mathcal{L}(\theta, \phi; \mathbf{x})$

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})] - KL(q_\phi(\mathbf{z}|\mathbf{x}) \,||\, p_\theta(\mathbf{z}))$$

Reconstruction error — Regularization

Cost function to be optimized: $\frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(\theta, \phi; \mathbf{x}_n)$, from dataset $\mathbf{X} = \{\mathbf{x}_n\}_{n=\overline{1,N}}$

Backpropagating through Stochastic Layers

- Training neural networks requires computing the gradient of the cost function, using backpropagation
- Difficulty when computing $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})]$ - Monte Carlo estimation of the gradient has high variance
- The reparameterization trick: find $g_\phi(\cdot)$ differentiable transformation and random variable Γ with pdf $p(\cdot)$, such that $\mathbf{Z} = g_\phi(\Gamma)$.

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})] = \mathbb{E}_{p(\gamma)}[\ln p_\theta(x|g_\phi(\gamma))]$$
$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})] = \mathbb{E}_{p(\gamma)}[\nabla_\phi \ln p_\theta(x|g_\phi(\gamma)]$$

- Example for $\mathbf{X} \sim \mathbb{N}(\mu, \Sigma)$, with $\Sigma = LL^T$ Cholesky decomposition: $\mathbf{X} = \mu + L\Gamma$, with $\Gamma \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Backpropagating through Stochastic Layers

- Training neural networks requires computing the gradient of the cost function, using backpropagation
- Difficulty when computing $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})]$ - Monte Carlo estimation of the gradient has high variance
- The reparameterization trick: find $g_\phi(\cdot)$ differentiable transformation and random variable Γ with pdf $p(\cdot)$, such that $\mathbf{Z} = g_\phi(\Gamma)$.

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})] = \mathbb{E}_{p(\gamma)}[\ln p_\theta(x|g_\phi(\gamma))]$$
$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})] = \mathbb{E}_{p(\gamma)}[\nabla_\phi \ln p_\theta(x|g_\phi(\gamma)]$$

- Example for $\mathbf{X} \sim \mathbb{N}(\mu, \Sigma)$, with $\Sigma = LL^T$ Cholesky decomposition: $\mathbf{X} = \mu + L\Gamma$, with $\Gamma \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Limitations and Challenges

- Limitations
  - The conditional independence assumption on the latent variables given the observations limits the expressive power of the approximate posterior
  - Limitation on the number of active latent variables when using a hierarchy of stochastic layers [13]
- Challenges
  - Difficulty when training on text data: empirical observation that the learned latent representation is not meaningful [2]
  - How to improve the quality of the generated samples, in case of a dataset of images? How can we find a better correlation between the images generated and the maximization of the lower bound?
  - How to estimate the tightness of the bound?

Limitations and Challenges

- Limitations
  - The conditional independence assumption on the latent variables given the observations limits the expressive power of the approximate posterior
  - Limitation on the number of active latent variables when using a hierarchy of stochastic layers [13]
- Challenges
  - Difficulty when training on text data: empirical observation that the learned latent representation is not meaningful [2]
  - How to improve the quality of the generated samples, in case of a dataset of images? How can we find a better correlation between the images generated and the maximization of the lower bound?
  - How to estimate the tightness of the bound?

Research Directions

- *More complex representations for* $q_\phi(\mathbf{z}|\mathbf{x})$, by transforming a simple distribution through invertible differentiable functions, as in [10] and [5]
- Increased complexity of the graphical models, e.g. a *hierarchy of latent variables* or *auxiliary variables* as in [13] and [9]
- Designing *tighter bounds*:
  - importance weighting estimates of the log-likelihood [3]

$$\mathcal{L}_K(\phi, \theta; \mathbf{x}) = \mathbb{E}_{\mathbf{z}^1, \mathbf{z}^2, \ldots \mathbf{z}^K \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(\mathbf{x}, \mathbf{z}^k)}{q_\phi(\mathbf{z}^k|\mathbf{x})} \right]$$

  - minimizing different divergences (Renyi [8], α-divergence [4])
- Overcoming the challenge of *training VAE on text data* [2]

Research Directions

- *More complex representations for* $q_\phi(\mathbf{z}|\mathbf{x})$, by transforming a simple distribution through invertible differentiable functions, as in [10] and [5]
- Increased complexity of the graphical models, e.g. a *hierarchy of latent variables* or *auxiliary variables* as in [13] and [9]
- Designing *tighter bounds*:
  - importance weighting estimates of the log-likelihood [3]

$$\mathcal{L}_K(\phi, \theta; \mathbf{x}) = \mathbb{E}_{\mathbf{z}^1, \mathbf{z}^2, \ldots \mathbf{z}^K \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(\mathbf{x}, \mathbf{z}^k)}{q_\phi(\mathbf{z}^k|\mathbf{x})} \right]$$

  - minimizing different divergences (Renyi [8], α-divergence [4])
- Overcoming the challenge of *training VAE on text data* [2]

Research Directions

- *More complex representations for* $q_\phi(\mathbf{z}|\mathbf{x})$, by transforming a simple distribution through invertible differentiable functions, as in [10] and [5]
- Increased complexity of the graphical models, e.g. a *hierarchy of latent variables* or *auxiliary variables* as in [13] and [9]
- Designing *tighter bounds*:
  - importance weighting estimates of the log-likelihood [3]

    $$\mathcal{L}_K(\phi, \theta; \mathbf{x}) = \mathbb{E}_{\mathbf{z}^1, \mathbf{z}^2, \ldots \mathbf{z}^K \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(\mathbf{x}, \mathbf{z}^k)}{q_\phi(\mathbf{z}^k|\mathbf{x})} \right]$$

  - minimizing different divergences (Renyi [8], α-divergence [4])
- Overcoming the challenge of *training VAE on text data* [2]

Research Directions

- *More complex representations for* $q_\phi(\mathbf{z}|\mathbf{x})$, by transforming a simple distribution through invertible differentiable functions, as in [10] and [5]
- Increased complexity of the graphical models, e.g. a *hierarchy of latent variables* or *auxiliary variables* as in [13] and [9]
- Designing *tighter bounds*:
  - importance weighting estimates of the log-likelihood [3]

$$\mathcal{L}_K(\phi, \theta; \mathbf{x}) = \mathbb{E}_{\mathbf{z}^1, \mathbf{z}^2, \ldots \mathbf{z}^K \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(\mathbf{x}, \mathbf{z}^k)}{q_\phi(\mathbf{z}^k|\mathbf{x})} \right]$$

  - minimizing different divergences (Renyi [8], α-divergence [4])
- Overcoming the challenge of *training VAE on text data* [2]

Outline

1. Plan of the presentation
2. General View of Variational Autoencoders
   - Introduction
   - Research Directions
3. Work-in-progress
   - Using Gaussian Graphical Models
   - Geometry of the Latent Space
4. Future Work

Gaussian Graphical Models for VAE

- *Gaussian Graphical Models* [7] introduce correlations in the latent variables.
- **Chain** and **2D grid** topologies $\Longrightarrow$ sparse precision matrix $\mathbf{P} = \Sigma^{-1}$, with the number of non-zero components linear in the dimension of the latent variable
- The encoder network outputs the mean μ and the Cholesky factor $\mathbf{L}$ of the precision matrix. $\mathbf{L}$ will have a special sparse structure and will ensure the positive definiteness of Σ.
- To sample from $\mathcal{N}(\mu, \Sigma)$: solve linear system $\mathbf{L}^T \nu = \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and output $\mathbf{z} = \mu + \nu$.
- Sampling from $\mathcal{N}(\mu, \Sigma)$ and computing $\mathrm{KL}(\mathcal{N}(\mu, \Sigma) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I}))$ can be done in linear time $\Longrightarrow$ introduce expressiveness without extra computational complexity.

Gaussian Graphical Models for VAE

- *Gaussian Graphical Models* [7] introduce correlations in the latent variables.
- **Chain** and **2D grid** topologies $\Longrightarrow$ sparse precision matrix $\mathbf{P} = \Sigma^{-1}$, with the number of non-zero components linear in the dimension of the latent variable
- The encoder network outputs the mean μ and the Cholesky factor $\mathbf{L}$ of the precision matrix. $\mathbf{L}$ will have a special sparse structure and will ensure the positive definiteness of Σ.
- To sample from $\mathcal{N}(\mu, \Sigma)$: solve linear system $\mathbf{L}^T \nu = \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and output $\mathbf{z} = \mu + \nu$.
- Sampling from $\mathcal{N}(\mu, \Sigma)$ and computing $\mathrm{KL}(\mathcal{N}(\mu, \Sigma) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I}))$ can be done in linear time $\Longrightarrow$ introduce expressiveness without extra computational complexity.

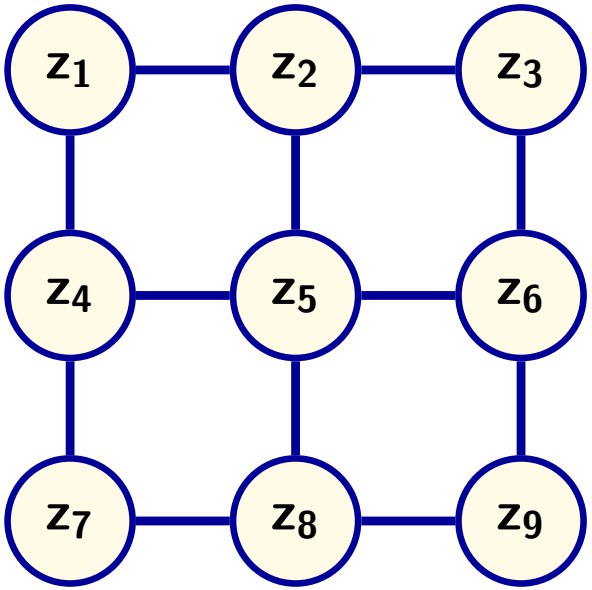
Chain Topology

Chain Model

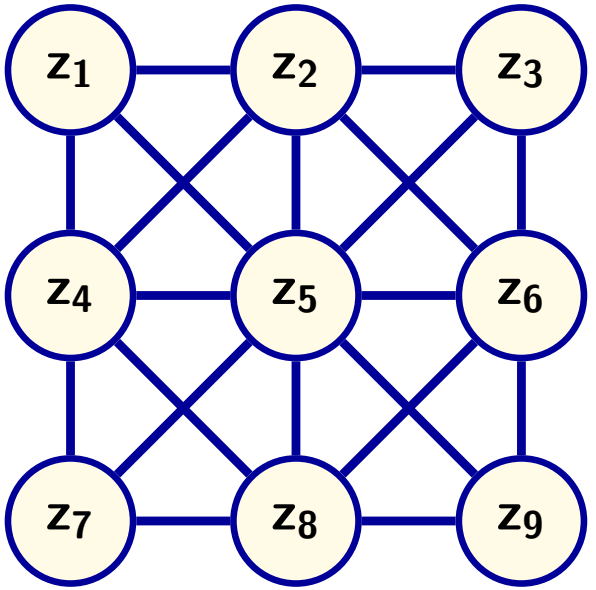
$$P = \begin{pmatrix} \sigma_1 & \lambda_1 & & 0 \\ \lambda_1 & \sigma_2 & \lambda_2 & \\ & & \ddots & \\ 0 & & \lambda_{k-1} & \sigma_k \end{pmatrix}$$

- The precision matrix **P** is tridiagonal;
- the Cholesky factor of such a matrix is lower-bidiagonal.

Grid Topology



Regular Grid

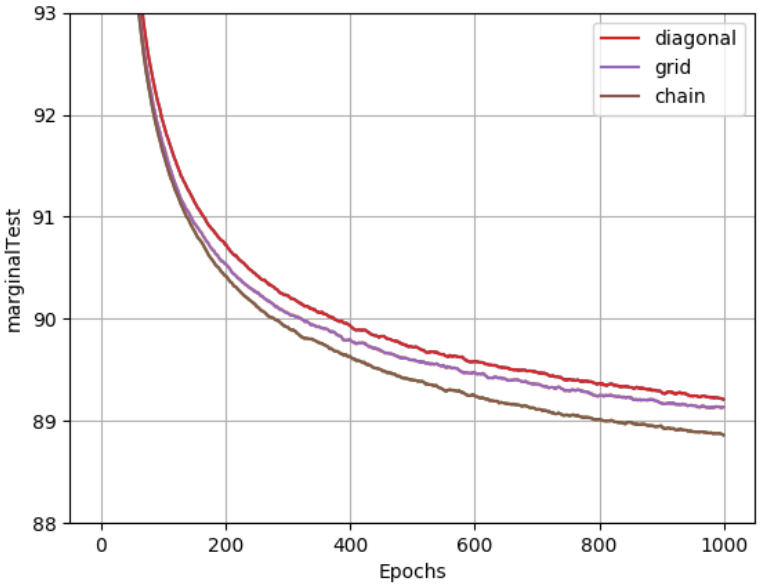


Extended Grid

- We use in our experiments the extended grid, which corresponds to a block tridiagonal precision matrix $\mathbf{P}$;
- we assume the Cholesky factor has a lower-block-bidiagonal structure.

Motivation for Next Research Direction

- The purpose was to *approximate the posterior with more complex distributions*.
- Although the results show a *slight improvement*, they do not motivate the future use of these models.
- A more comprehensive analysis should be made to understand the *geometry of the latent space*.

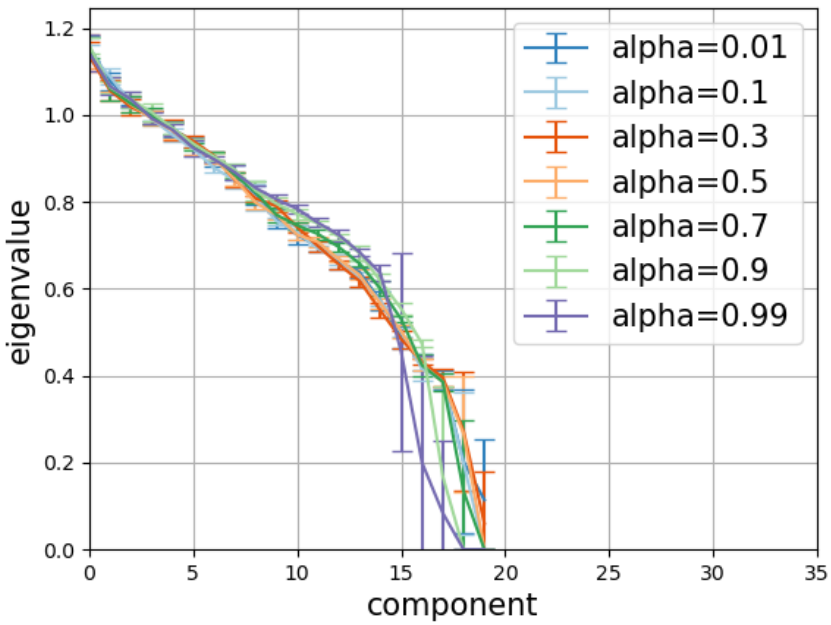


Analysis of the Representations in the Latent Space

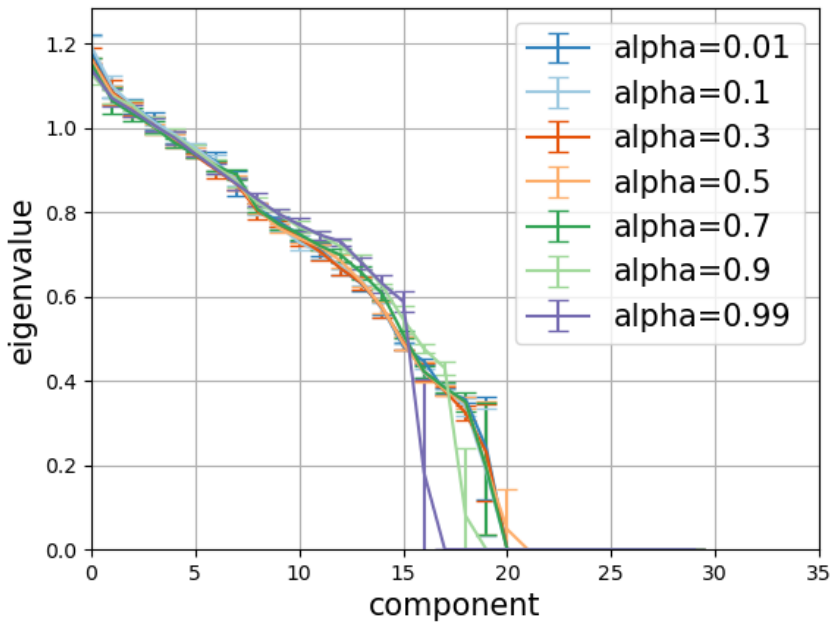
- Experiments on MNIST dataset to understand the *representation of the images in the learned latent space*.
- *Principal Components Analysis* of the latent means will give us insights about *which components are relevant for the representation*.
- Claim: components with a *low variation* along the dataset are the ones not meaningful.
- PCA eigenvalues of the posterior samples are very close to 1 $\Longrightarrow$ the KL minimization forces some components to be $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

PCA Analysis 1/2

$k = 20$, RELU

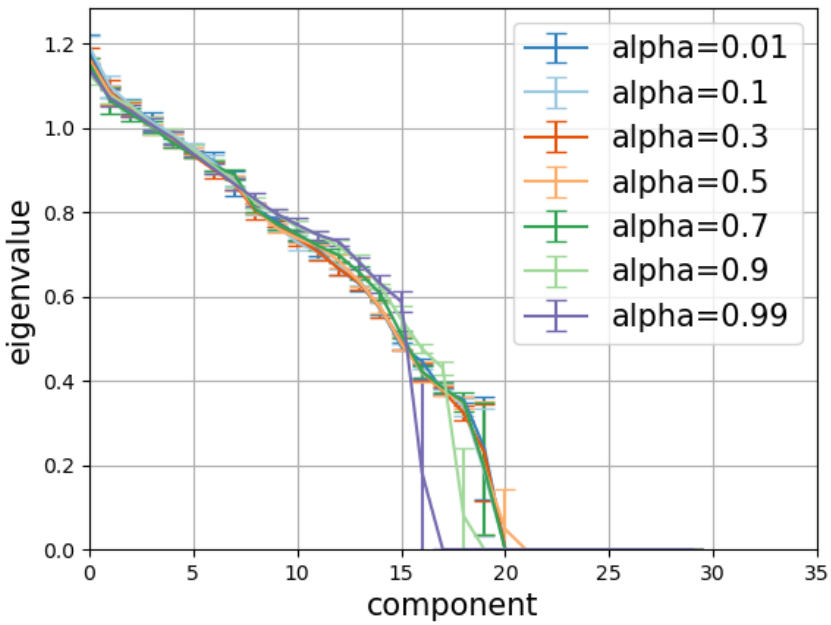


$k = 30$, RELU

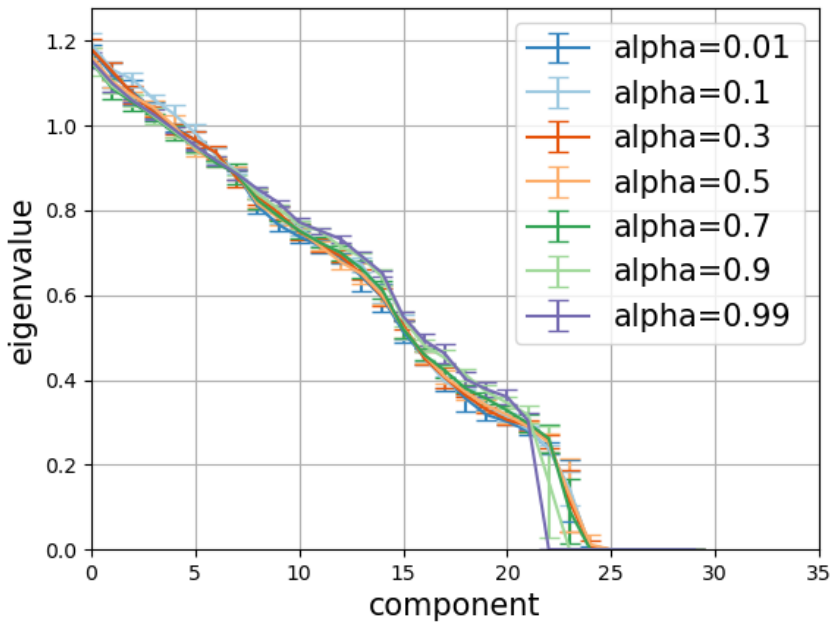


PCA Analysis 2/2

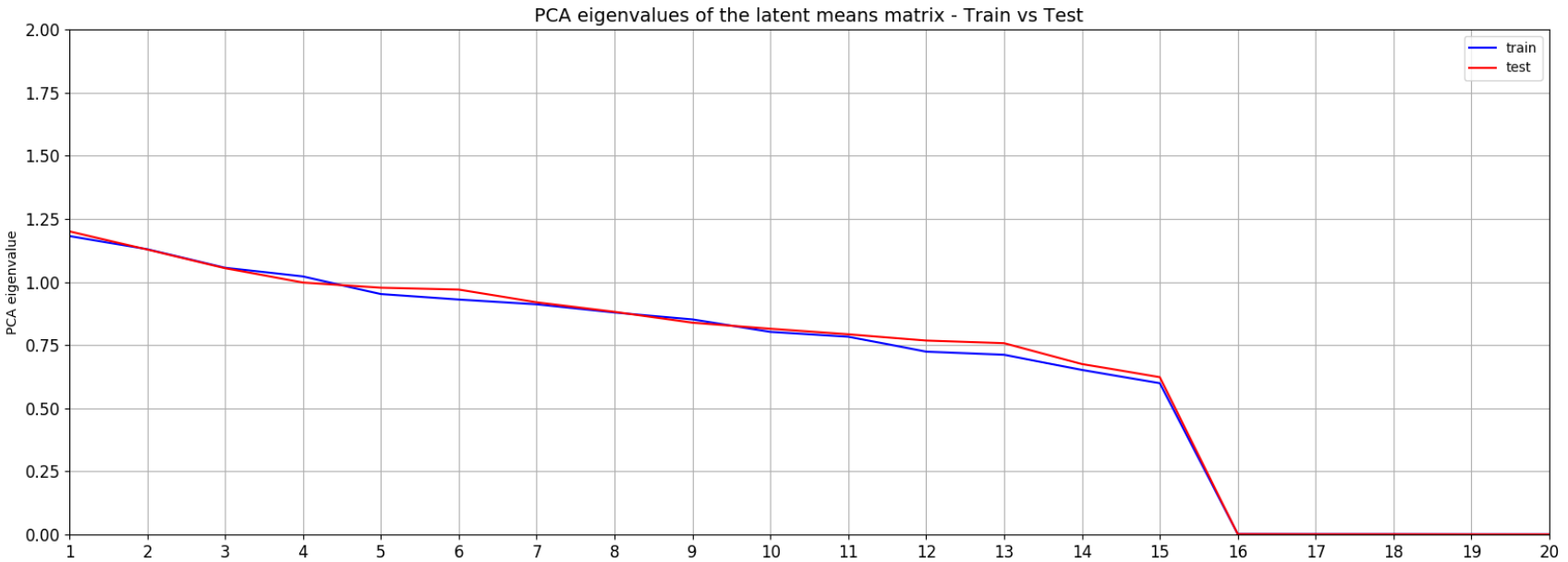
$k = 30$, RELU

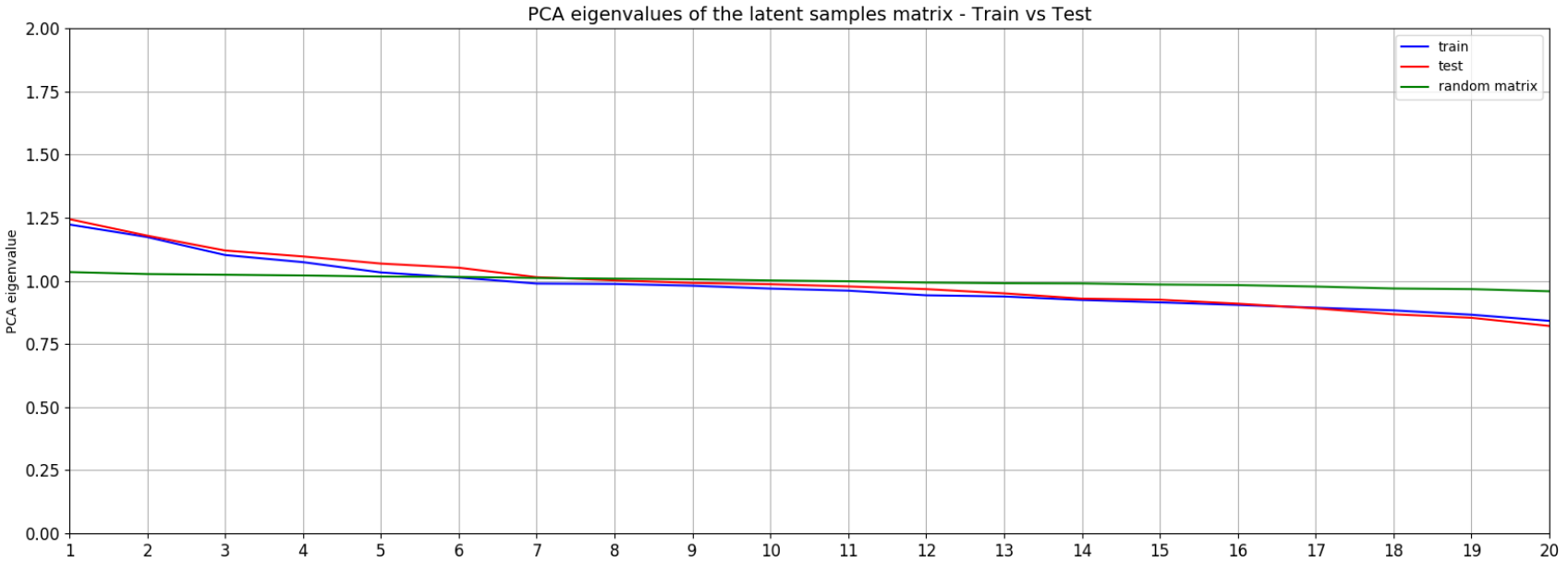


$k = 30$, ELU



PCA Plots


PCA eigenvalues of the latent means matrix - Train vs Test
PCA eigenvalue
train
test
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20


PCA eigenvalues of the latent samples matrix - Train vs Test
PCA eigenvalue
train
test
random matrix
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

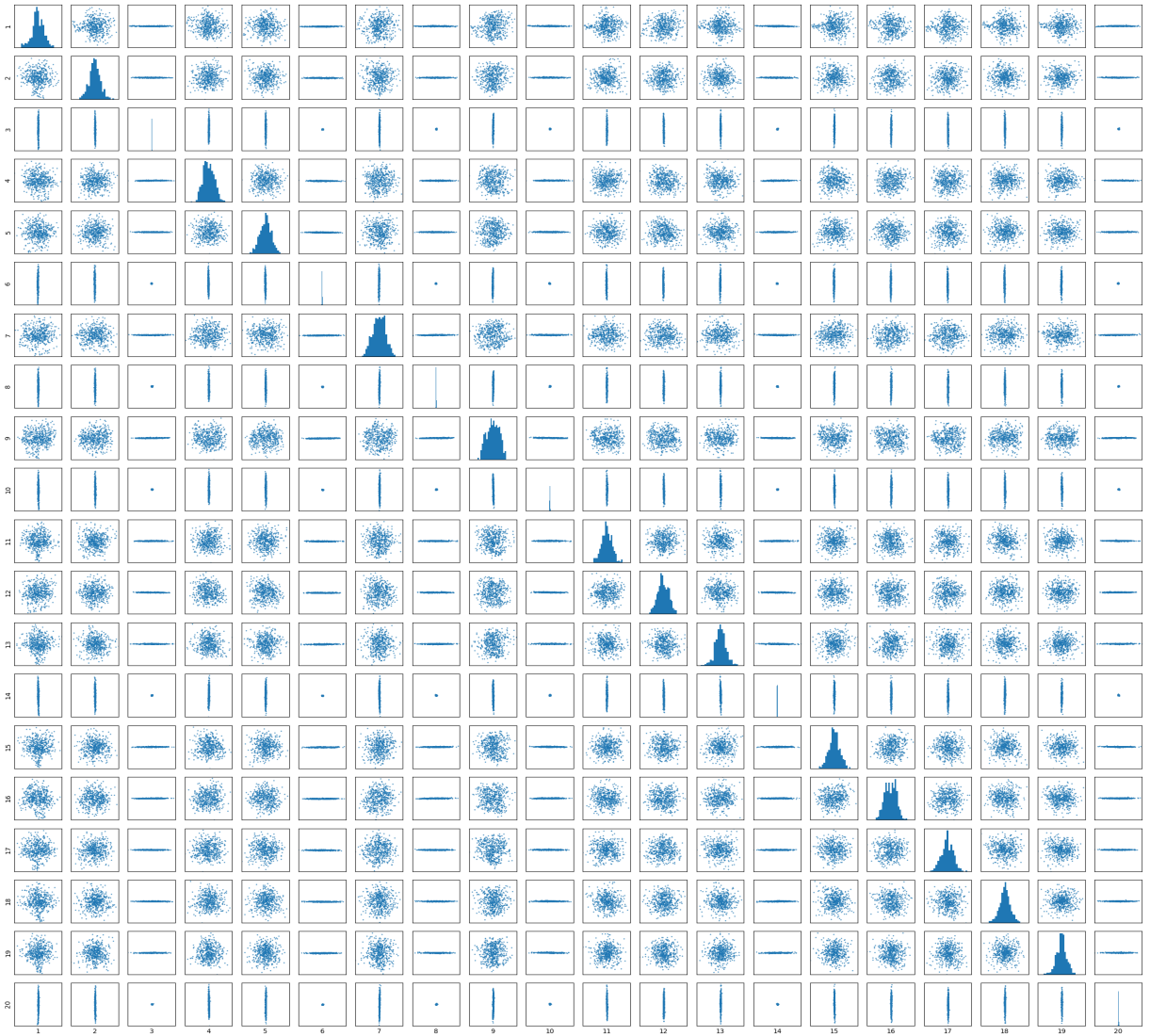
Interpretation of the Plot

- When training a VAE with latent size 20 on MNIST, only around 15 of the latent variables are relevant for the representation.
- The number remains constant when training with a larger latent size.
- This is a consequence of the KL regularization term in the cost function, which forces some components to be Gaussian noise.
- Is this number a particularity of the dataset?
- What is the impact on this number when using more complicated network architectures?
- Would we observe the same behavior with other bounds derived from different divergences (e.g. Rényi)?

Interpretation of the Plot

- When training a VAE with latent size 20 on MNIST, only around 15 of the latent variables are relevant for the representation.
- The number remains constant when training with a larger latent size.
- This is a consequence of the KL regularization term in the cost function, which forces some components to be Gaussian noise.
- Is this number a particularity of the dataset?
- What is the impact on this number when using more complicated network architectures?
- Would we observe the same behavior with other bounds derived from different divergences (e.g. Rényi)?

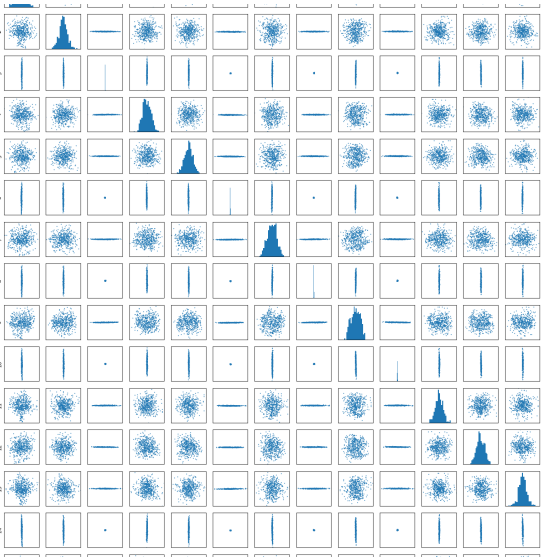
Correlations Plot



Interpretation of the Plot

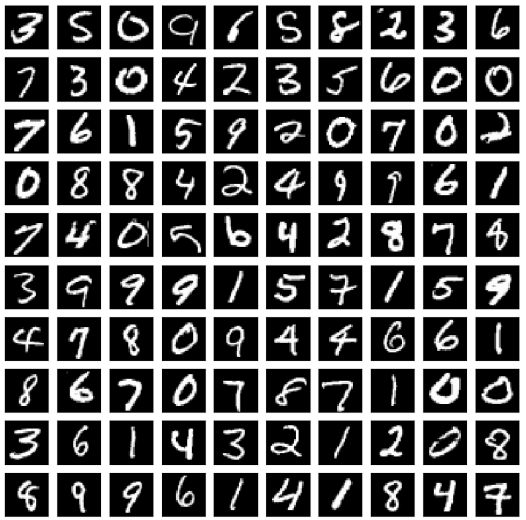
With the previous plot we want to better understand the distribution of the latent means vector across the dataset to identify the inactive components.

- Distribution of (μ_i, μ_j), samples corresponding to the points in the dataset.
- Inactive components are close to 0 and remain constant along the data set.

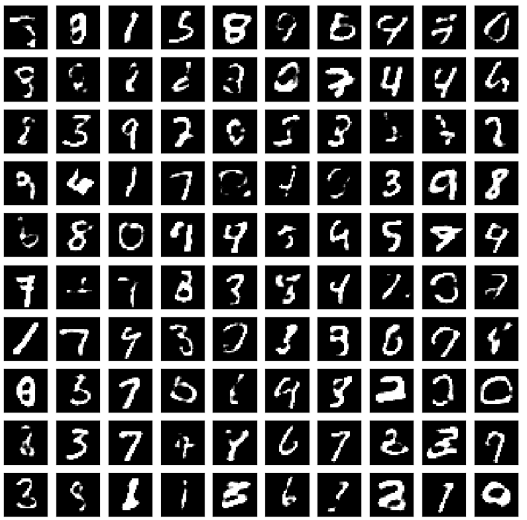


Generated images

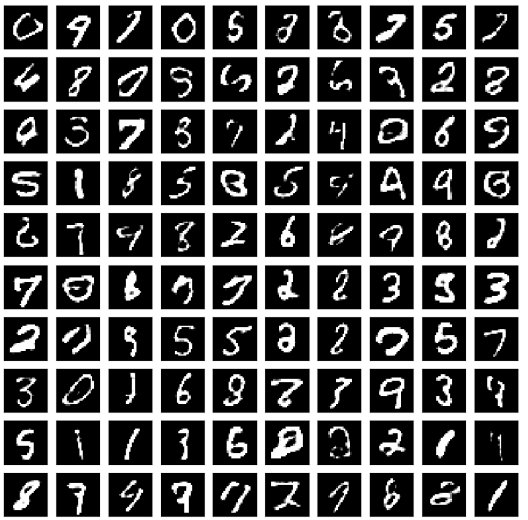
Images generated by training VAE on MNIST, with the encoder and decoder feed-forward neural networks with two hidden layers:



Samples from MNIST dataset



Generated after 100 epochs

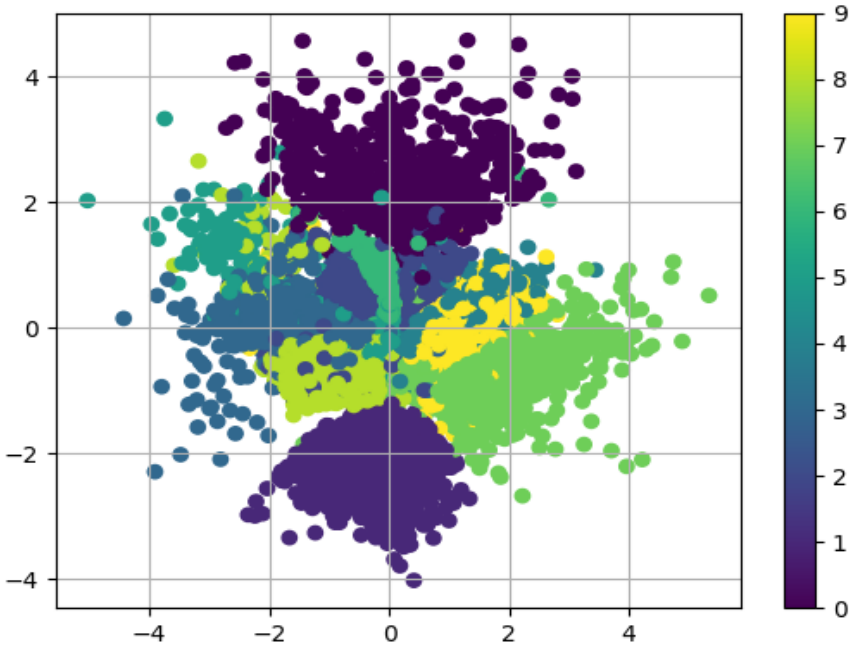


Generated after 1000 epochs

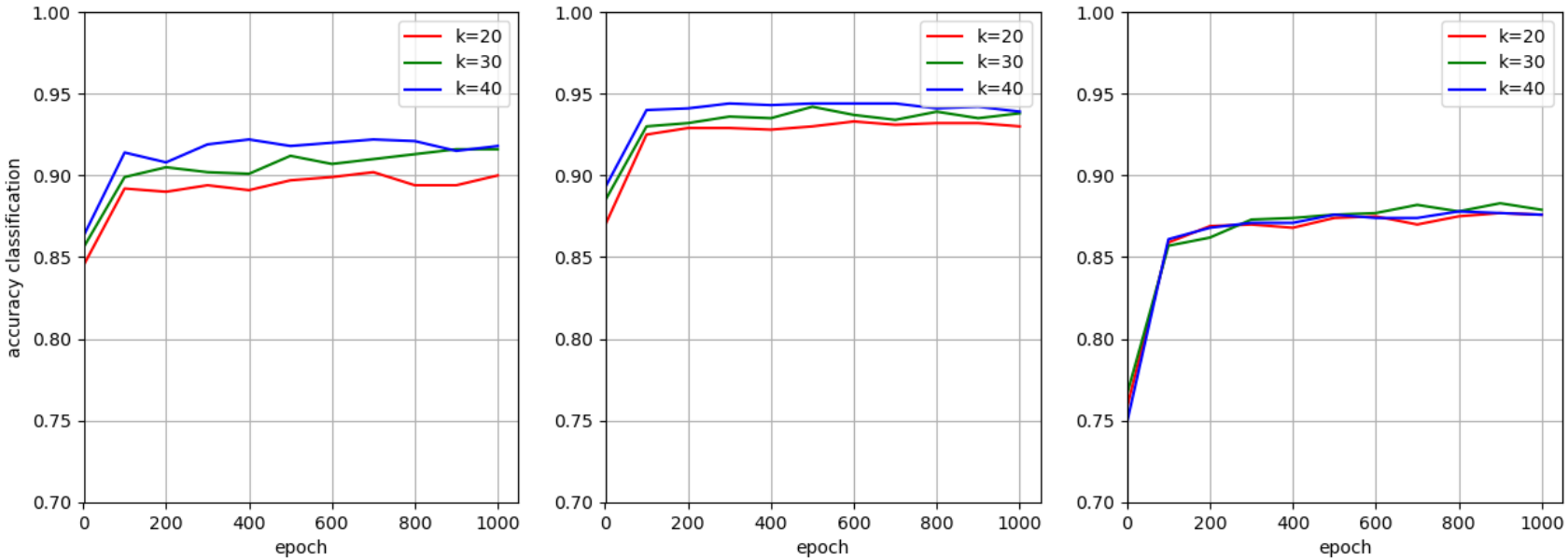
Linear Separability in the Latent Space

VAE is trained on MNIST with 2 latent variables. The plot represents the means of the posterior for each point in the dataset, colored by corresponding class.

- **Linear separability** of the classes in the space of latent representations
- Sampling in the latent space from the **empty regions** $\Longrightarrow$ images that are *not digits*
- **Linear interpolation property** $\Longrightarrow$ continuous deformation in the latent space between two different images.



Classification Performance



Outline

1. Plan of the presentation
2. General View of Variational Autoencoders
   - Introduction
   - Research Directions
3. Work-in-progress
   - Using Gaussian Graphical Models
   - Geometry of the Latent Space
4. Future Work

Short-Term

- Linear separability of the dataset in the space of multi-dimensional latent representations
- Use skew distributions to model the posterior
- Study the behavior of the latent relevant components in the case of more complex posteriors, like the ones presented in [10] and [5]

Medium-Term

- Bounds derived from different divergences (e.g. Rényi, α-divergence)
  - impact of the α parameter on the tightness of the bounds
  - relevant components in the latent space and see how their number changes
- Geometric methods for training VAE
  - the use of natural gradient
  - study the geometry of the latent space
  - use Riemannian optimization methods that exploit some properties of the space of the latent variables
- Extend the study to different types of generative models, e.g. Generative Adversarial Networks (GANs), Restricted Boltzmann Machines (RBMs).

References I

[1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe.
Variational inference: A review for statisticians.
Journal of the American Statistical Association, 2017.

[2] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio.
Generating sentences from a continuous space.
arXiv preprint arXiv:1511.06349, 2015.

[3] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov.
Importance weighted autoencoders.
arXiv preprint arXiv:1509.00519, 2015.

[4] José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang D Bui, and Richard E Turner.
Black-box α-divergence minimization.
2016.

References II

[5] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling.
Improving variational autoencoders with inverse autoregressive flow.
In *Advances In Neural Information Processing Systems*, pages 4736–4744, 2016.

[6] Diederik P Kingma and Max Welling.
Auto-encoding variational bayes.
2013.

[7] Steffen L Lauritzen.
Graphical models, volume 17.
Clarendon Press, 1996.

[8] Yingzhen Li and Richard E Turner.
Rényi divergence variational inference.
In *Advances in Neural Information Processing Systems*, pages 1073–1081, 2016.

References III

[9] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther.
Auxiliary deep generative models.
arXiv preprint arXiv:1602.05473, 2016.

[10] Danilo Jimenez Rezende and Shakir Mohamed.
Variational inference with normalizing flows.
arXiv preprint arXiv:1505.05770, 2015.

[11] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra.
Stochastic backpropagation and approximate inference in deep generative models.
2014.

[12] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen.
Improved techniques for training gans.
In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

References IV

[13] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther.
Ladder variational autoencoders.
In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3738–3746. Curran Associates, Inc., 2016.

Questions?

Transylvanian Machine Learning Summer School

16-22 July 2018, Cluj-Napoca, Romania

Lectures

Doina Precup
McGill University & DeepMind

Dumitru Erhan
Google Brain

Guido Montúfar
University of California, Los Angeles

Jan Chorowski
University of Wrocław & Google Brain

Kyunghyun Cho
New York University & Facebook AI Research

Lucian Buşoniu
Technical University of Cluj-Napoca

Luigi Malagò
Romanian Institute of Science and Technology

Maria-Florina Balcan
Carnegie Mellon University

Marius Leordeanu
Politechnica University of Bucharest & Institute of Mathematics of the Romanian Academy

Oriol Vinyals
DeepMind

Raia Hadsell
DeepMind

Razvan Pascanu
DeepMind

Ulrich Paquet
DeepMind

Transylvanian Machine Learning Summer School

16-22 July 2018, Cluj-Napoca, Romania

Practical sessions

Diana Borsa
DeepMind & University College London

Mihaela Rosca
DeepMind

Viorica Patraucean
DeepMind

Wojtek Czarnecki
DeepMind

Registration Deadline: 30 March 2018

More details at `www.tmlss.ro`

Thank You!