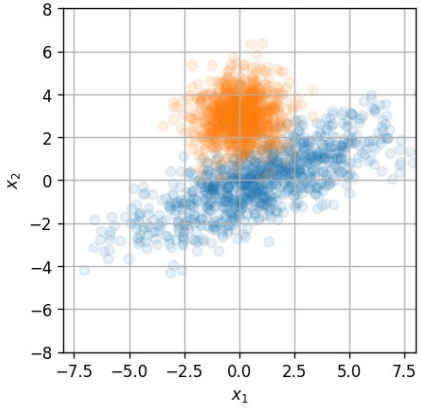
Variational Inference and Variational Autoencoders

- variational inference
- variational autoencoders

Unsupervised Learning

- *supervised* machine learning is about **learning functions from data**

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$$

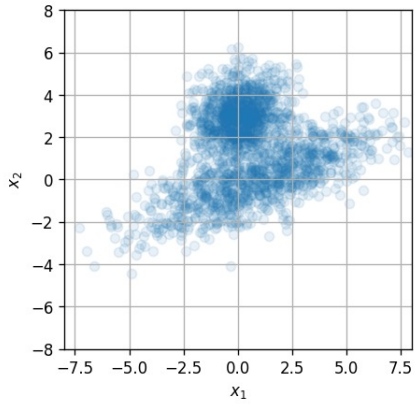


- *unsupervised* machine learning is about finding interesting patterns in data

$$\mathcal{D} = \{(x_i)\}_{i=1}^{n}$$

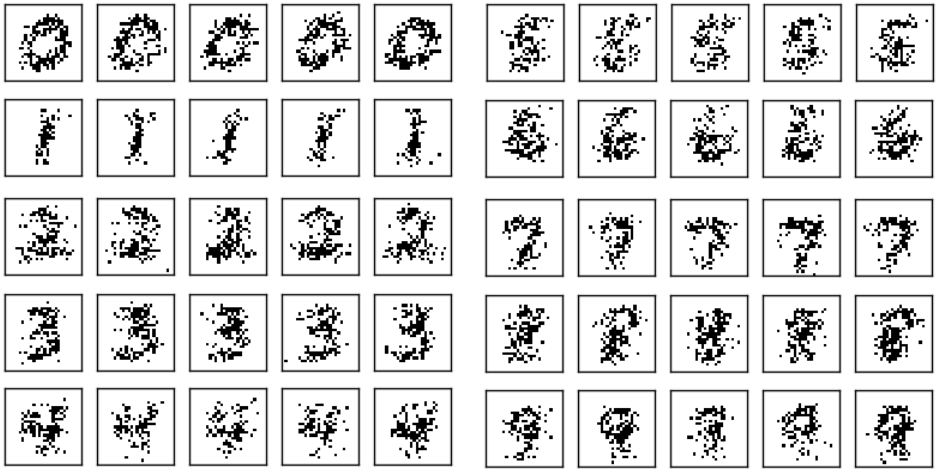
- *knowlege discovery*
- under-specified

- learn $p(\boldsymbol{x})$ instead of $p(\boldsymbol{x}, y)$ or $p(y|\boldsymbol{x})$
- learn $p_\theta(\boldsymbol{x})$



Learning to Draw MNIST

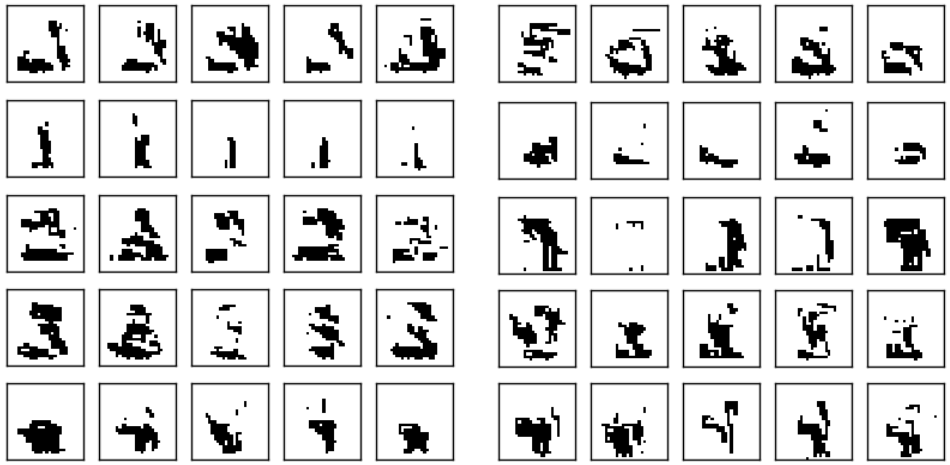
- Naive Bayes



- sample with 784 independent coin flips with different bias

Learning to Draw MNIST

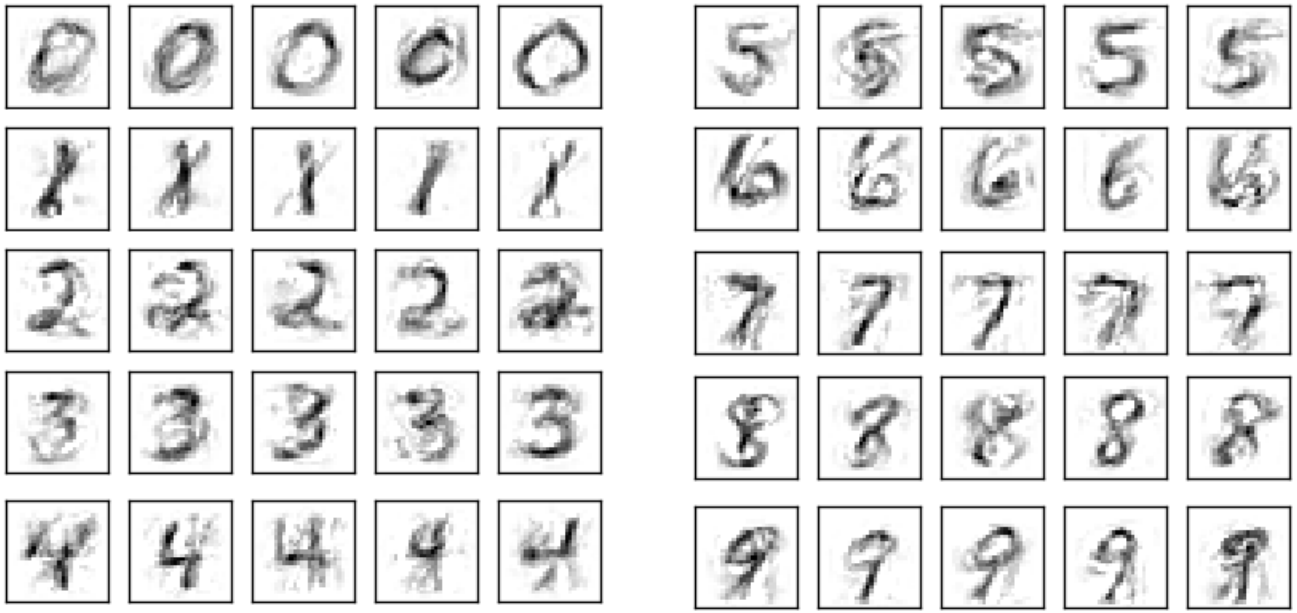
- Bayes Net / Belief Network



- ancestral sampling

Learning to Draw MNIST

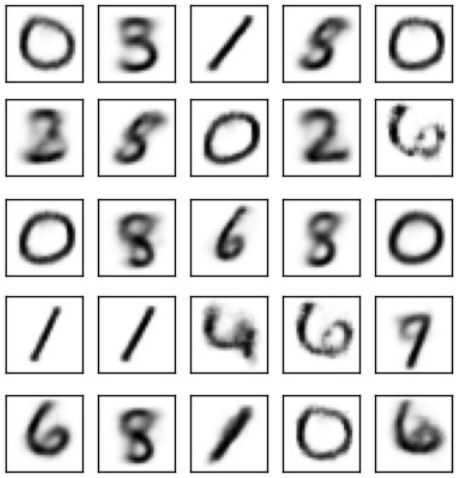
- multivariate normal

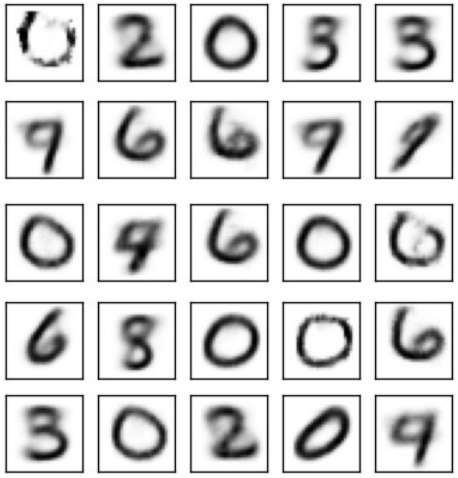


- sample $\mathcal{N}(0, \boldsymbol{I})$, compute $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\mu}$

Learning to Draw MNIST

- variational autoencoder





- sample $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$, pass sample through function $g(\cdot)$

Variational Autoencoders (at test time)

- how can we sample from a VAE?

$\boldsymbol{x}_i =$ 

$\boldsymbol{x} \in \mathbb{R}^d \rightarrow f(\cdot) \rightarrow (\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$\boldsymbol{\mu} = \begin{bmatrix} 0.8 \\ -0.4 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.02 \end{bmatrix}$

sample $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$\boldsymbol{z} = \begin{bmatrix} 0.81 \\ -0.39 \end{bmatrix}$

$\rightarrow g(\cdot) \rightarrow \widehat{\boldsymbol{x}} \in \mathbb{R}^d$

$\widehat{\boldsymbol{x}}_i =$ 

- how can we learn $f(\cdot)$ and $g(\cdot)$ from data?
  - ensure that $\boldsymbol{x} \approx \widehat{\boldsymbol{x}}$
  - try to force $p(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

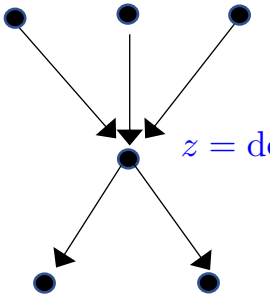
Latent Variable Models

- goal: learn $p(\boldsymbol{x})$ from $\mathcal{D} = \{(\boldsymbol{x}_i)\}_{i=1}^n$

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, z) dz$$

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

M/F, age, income, etc.

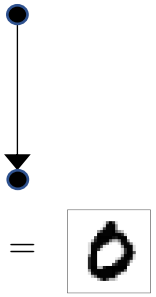


z = democrat or republican

how they vote in two elections

- z is a *latent* variable
- z might be meaningful, or maybe just convenient
- MNIST: z = digit, pen width, slant, etc.
  - not principled/helpful to specify z manually

z = intended digit, slant, width, etc.

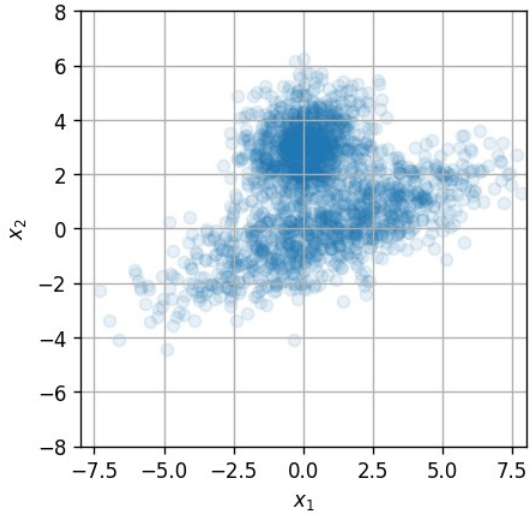


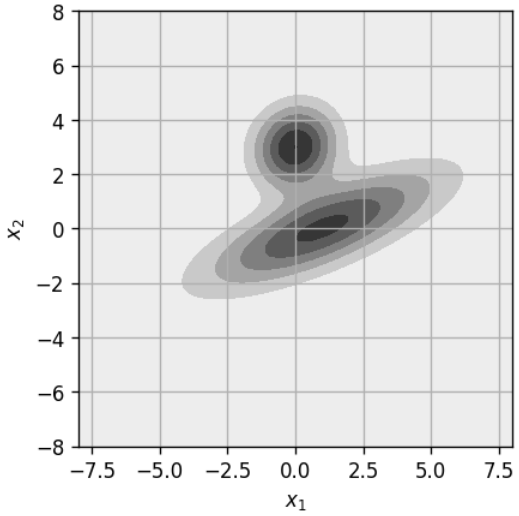
$\boldsymbol{x}_i =$

- if z is discrete, then *mixture* model

$$p(\boldsymbol{x}) = \sum_z p(z) p(\boldsymbol{x}|z)$$

Latent Variable Model Example - Gaussian Mixture Models (GMM)





$$p(\boldsymbol{x}) = 0.5\ \mathcal{N}\left(\boldsymbol{x}\middle|\begin{bmatrix}0\\3\end{bmatrix}, \begin{bmatrix}1 & 0.1\\0.1 & 1\end{bmatrix}\right) + 0.5\ \mathcal{N}\left(\boldsymbol{x}\middle|\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}8 & 3\\3 & 2\end{bmatrix}\right)$$

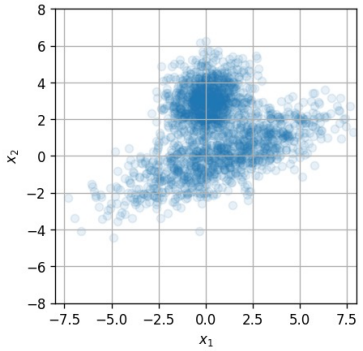
- Gaussian mixture model:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(z=k)\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Latent Variable Models

- goal: learn relationship between $\boldsymbol{x}$ and $\boldsymbol{z}$

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



- learning $p(\boldsymbol{z}|\boldsymbol{x})$ without more assumptions is ill-posed
  - need to specify the form of $p(\boldsymbol{z})$
  - need to specify the form of $p(\boldsymbol{x}|\boldsymbol{z})$
  - example: Gaussian mixtures and EM

$$p(\boldsymbol{z}|\boldsymbol{x}) = \quad r_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i|\mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\boldsymbol{x}_i|\mu_{k'}, \Sigma_{k'})}$$

- starting point $p(\boldsymbol{z}) \sim K$ categories
  and $p(\boldsymbol{x}|\boldsymbol{z})$ Gaussian

Variational Inference

- goal: learn $p(\boldsymbol{z}|\boldsymbol{x})$

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

- $p(\boldsymbol{z}|\boldsymbol{x})$ is the posterior over the hidden variable $\boldsymbol{z}$

- variational inference: estimate $p(\boldsymbol{z}|\boldsymbol{x})$ using some other distribution $q(\boldsymbol{z}|\boldsymbol{x})$

- restrict $q(\boldsymbol{z}|\boldsymbol{x})$ to be from some tractable family, i.e, $\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\Sigma}(\boldsymbol{x}))$

- consider the KL divergence between $q(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z}|\boldsymbol{x})$:

$$D(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|x)) = \int q(\boldsymbol{z}|\boldsymbol{x}) \log \frac{q(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z}|x)} d\boldsymbol{z}$$

Variational Inference

- consider the KL divergence between $q(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z}|\boldsymbol{x})$:

$$0 \leq D(q(\cdot\,|x)||p(\cdot\,|x)) = \int q(z|x)\log\frac{q(z|x)}{p(z|x)}\,dz, \qquad \text{with equality if and only if } q(\cdot\,|x) = p(\cdot\,|x),$$

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

$$= \int q(z|x)\log p(x)\,dz + \int q(z|x)\log\frac{1}{p(x|z)}\,dz + \int q(z|x)\log\frac{q(z|x)}{p(z)}\,dz$$

$$= \log p(x) - \mathbb{E}_{q(\cdot|x)}[\log p(x|Z)] - D(q(\cdot\,|x)||p(\cdot)).$$

Therefore,

$$\log p(x) \geq E_{q(\cdot|x)}[\log p(x|Z)] + D(q(\cdot\,|x)||p(\cdot)).$$

Instead of choosing $\mu(x)$ and $\Sigma(x)$ to maximize $\log p(x)$, as in maximum likelihood, maximize the right-hand side and hope for the best!

Variational Autoencoders

- how do we match data to a model? maximize the likelihood

$$\max \sum_i \log p(\boldsymbol{x}_i)$$

- we can try maximizing the lower bound

$$\log p(\boldsymbol{x}) \geq E_{z \sim q(z)} \left[\log p(\boldsymbol{x}|\boldsymbol{z})\right] - D(q(\boldsymbol{z})||p(\boldsymbol{z}))$$ (called ELBO - evidence lower bound)

- how can we compute $D(q(\boldsymbol{z})||p(\boldsymbol{z}))$?

- we can choose $p(\boldsymbol{z})$ to be $\mathcal{N}(0, \boldsymbol{I})$ and $q(\boldsymbol{z})$ to be $\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\Sigma}(\boldsymbol{x}))$

$$D(q(\boldsymbol{z})||p(\boldsymbol{z})) = \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} + \frac{1}{2}\text{tr}\left(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\right) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - \frac{1}{2}n$$

$$= \frac{1}{2}\log\frac{1}{|\boldsymbol{\Sigma}_1(\boldsymbol{x})|} + \frac{1}{2}\text{tr}\left(\boldsymbol{\Sigma}_1(\boldsymbol{x})\right) + \boldsymbol{\mu}(\boldsymbol{x})^T\boldsymbol{\mu}(\boldsymbol{x}) - \frac{1}{2}n$$

Variational Autoencoders

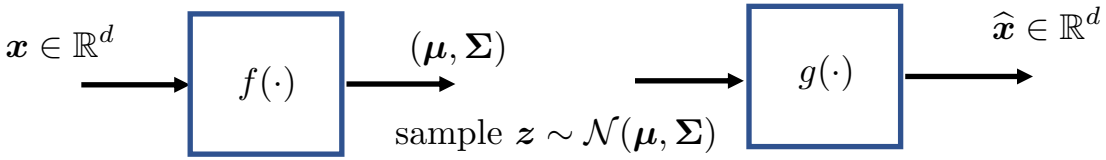
- how can we compute/maximize $E_{z\sim q(z)}\left[\log p(\boldsymbol{x}|\boldsymbol{z})\right]$?
- $g(\cdot)$ is a deterministic function

$$\log p(\boldsymbol{x}|\boldsymbol{z}) = \log p(\boldsymbol{x}|\widehat{\boldsymbol{x}})$$

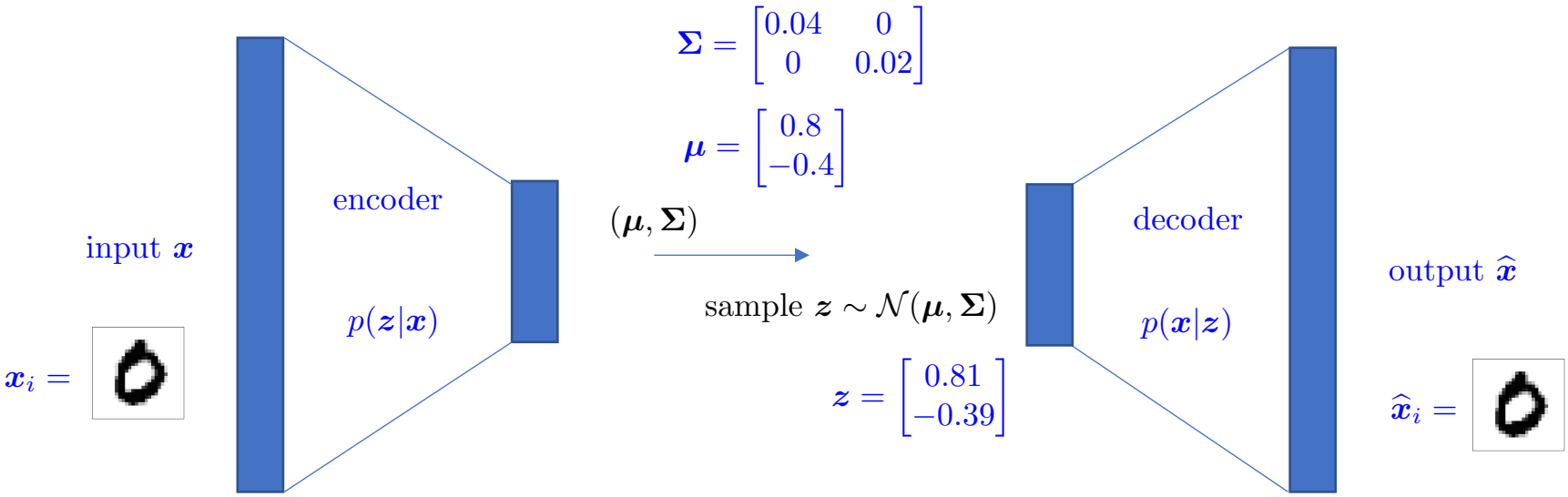
small caveat: if $g(\cdot)$ is an invertible mapping

- big when $\widehat{\boldsymbol{x}} \approx \boldsymbol{x}$
- a squared error loss helps maximize $p(\boldsymbol{x}|\boldsymbol{z})$

$$E_{z\sim q(z)}\left[\log p(\boldsymbol{x}|\boldsymbol{z})\right] \approx \frac{1}{n}\sum \log p(\boldsymbol{x}_i|\boldsymbol{z}_i) \approx \frac{1}{n}\sum \log p(\boldsymbol{x}_i|\widehat{\boldsymbol{x}}_i)$$



Variational Autoencoders



$$\ell(\boldsymbol{x}, \widehat{\boldsymbol{x}}) = ||\widehat{\boldsymbol{x}} - \boldsymbol{x}||^2 + D(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})||\mathcal{N}(0, \boldsymbol{I}))$$

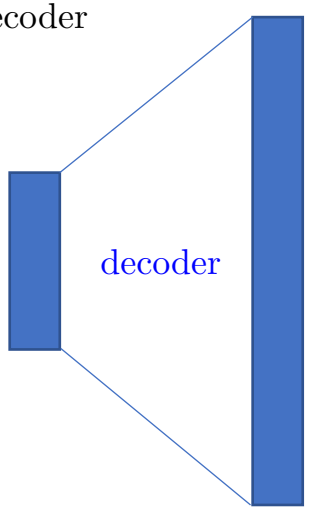
- ensure that $\boldsymbol{x} \approx \widehat{\boldsymbol{x}}$
- try to force $p(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

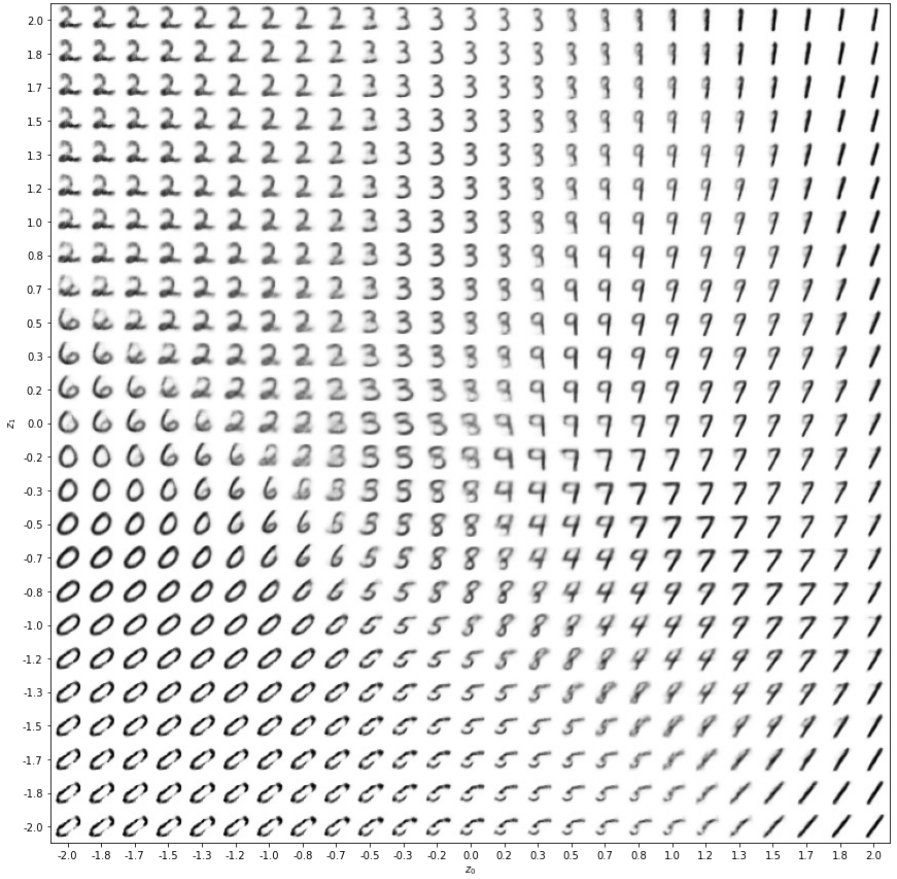
VAE's on MNIST

- how can we generate random MNIST images?

  1. sample $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

  2. input sample into decoder side of neural network





- how can we creating this figure?

  1. grid $\boldsymbol{z} \in [-2, 2]^2$

  2. input each grid point into decoder side of neural network

VAE's on MNIST

- [Tutorial on Variational Autoencoders, Carl Doersch]
- [Variational Autoencoders, Ali Ghodsi]
- [Keras, https://keras.io/examples/generative/vae/]