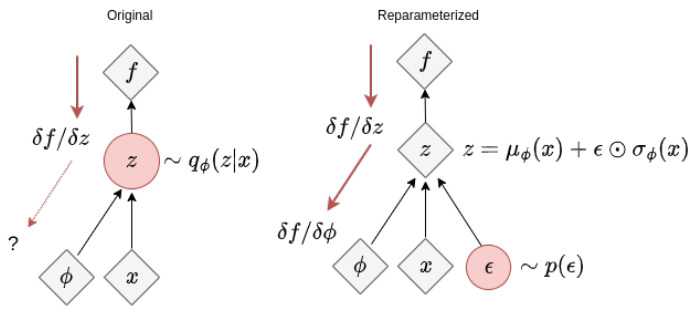
Deep Learning

Variational Autoencoder (VAE)



Learning goals

- introduction and intution of VAE
- VAE-parameter fitting
- reparametrization trick

VARIATIONAL AUTOENCODER (VAE): INTUTION

Independently proposed by:

- Kingma and Welling, *Auto-Encoding Variational Bayes*, ICLR 2014
- Rezende, Mohamed and Wierstra, *Stochastic back-propagation and variational inference in deep latent Gaussian models.* ICML 2014

Conventional AEs compute a deterministic feature vector that describes the attributes of the input in latent space:

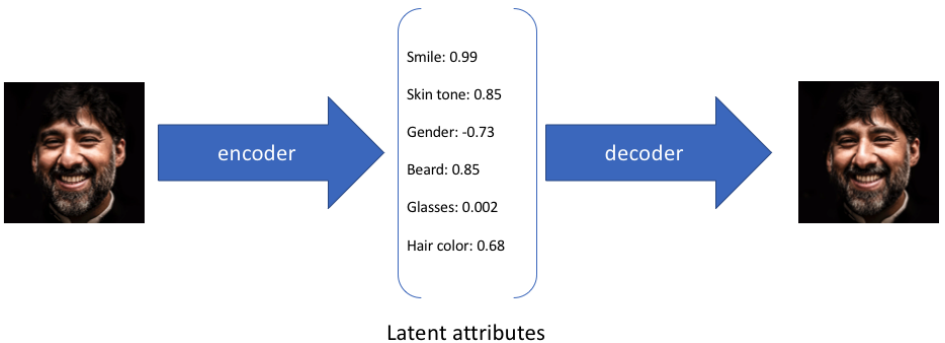


Figure: source: https://www.jeremyjordan.me/variational-autoencoders/

VARIATIONAL AUTOENCODER (VAE)

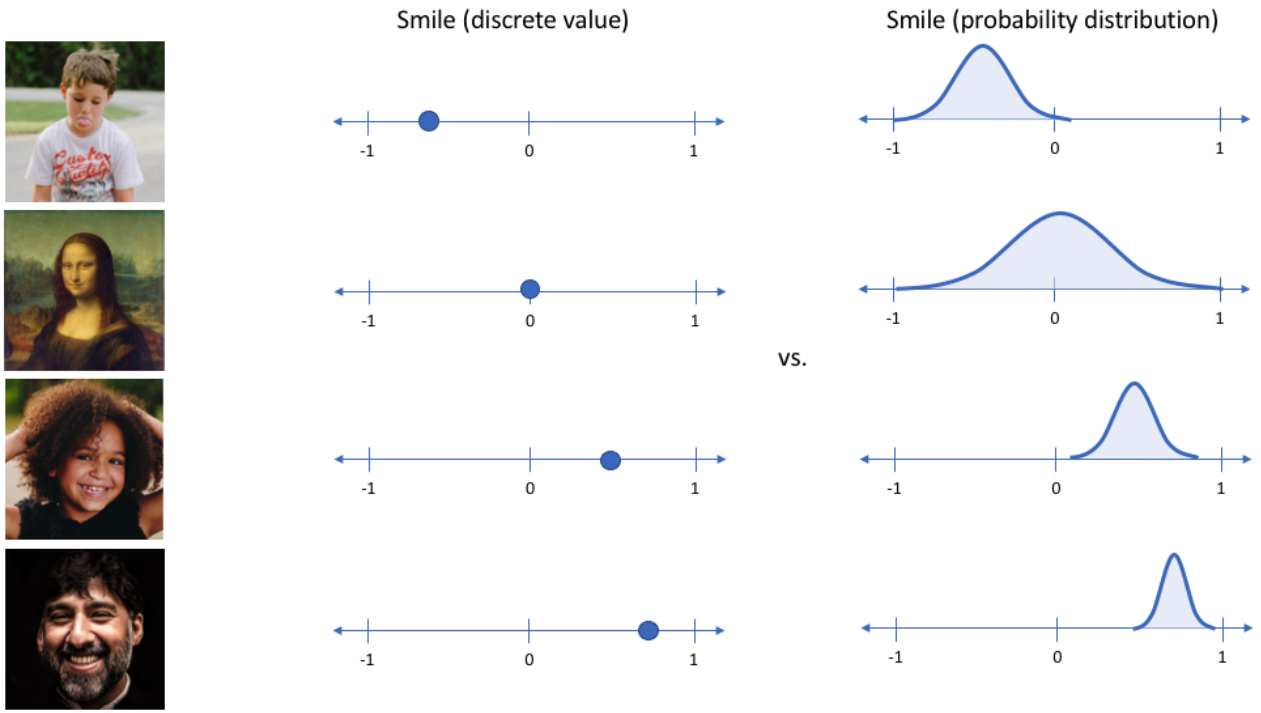


Figure: source: https://www.jeremyjordan.me/variational-autoencoders/

- Key difference in variational autoencoders are:
  - Uses a variational approach to learn the latent representation
  - Allows to describe observation in latent space in probablistic manner.

VARIATIONAL AUTOENCODER (VAE): INTUTION

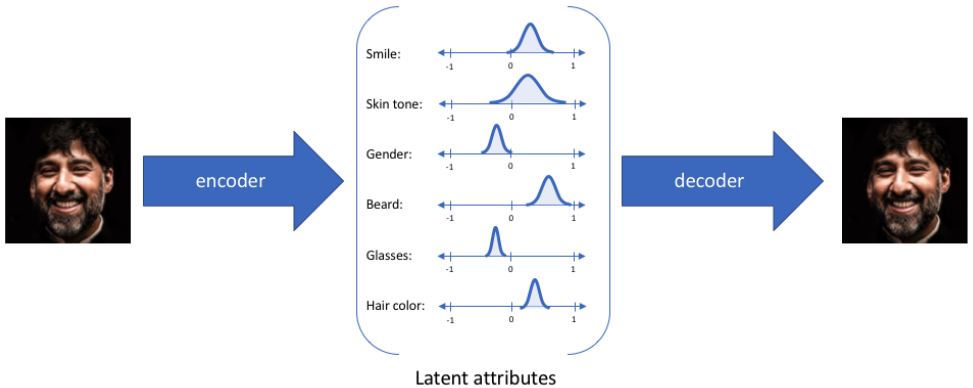


Figure: source: https://www.jeremyjordan.me/variational-autoencoders/

- Describe each latent attribute as probability distribution
- Allows to model uncertainty in input data

VAE: STATISTICAL MOTIVATION

- Suppose the hidden variable z which generates an observation x

VAE: STATISTICAL MOTIVATION

- Suppose the hidden variable z which generates an observation x
- By training variational autoencoder, we determin the distribution z and would like to compute $p(z|x)$.

$$p\left(z|x\right) = \frac{p(x|z)p(z)}{p(x)}$$

VAE: STATISTICAL MOTIVATION

- Suppose the hidden variable z which generates an observation x
- By training variational autoencoder, we determin the distribution z and would like to compute $p(z|x)$.

$$p\left(z|x\right) = \frac{p(x|z)p(z)}{p(x)}$$

- However, computing $p(x)$ is difficult since it is an intractable distribution!

$$p\left(x\right) = \int p\left(x|z\right) p\left(z\right) dz$$

VAE: STATISTICAL MOTIVATION

- Suppose the hidden variable z which generates an observation x
- By training variational autoencoder, we determin the distribution z and would like to compute $p(z|x)$.

$$p\left(z|x\right) = \frac{p(x|z)p(z)}{p(x)}$$

- However, computing $p(x)$ is difficult since it is an intractable distribution!

$$p\left(x\right) = \int p\left(x|z\right) p\left(z\right) dz$$

- Instead we can apply variational inference

VAE: STATISTICAL MOTIVATION

- Suppose the hidden variable z which generates an observation x
- By training variational autoencoder, we determin the distribution z and would like to compute $p(z|x)$.

$$p\left(z|x\right) = \frac{p(x|z)p(z)}{p(x)}$$

- However, computing $p(x)$ is difficult since it is an intractable distribution!

$$p\left(x\right) = \int p\left(x|z\right) p\left(z\right) dz$$

- Instead we can apply variational inference
- Let's approximate $p(z|x)$ by tractable distribution $q(z|x)$ which is very similar to $p(z|x)$.

VAE: STATISTICAL MOTIVATION

- KL divergence measures of difference between two probability distributions. Thus, if we wanted to ensure that $q(z|x)$ was similar to $p(z|x)$, we could minimize the KL divergence between the two distributions:

$$\min KL\left(q\left(z|x\right)||p\left(z|x\right)\right)$$

VAE: STATISTICAL MOTIVATION

- KL divergence measures of difference between two probability distributions. Thus, if we wanted to ensure that $q(z|x)$ was similar to $p(z|x)$, we could minimize the KL divergence between the two distributions:

$$\min KL\left(q\left(z|x\right)||p\left(z|x\right)\right)$$

  by maximizing the following:

$$\max E_{q(z|x)} \log p\left(x|z\right) - KL\left(q\left(z|x\right)||p\left(z\right)\right)$$

- The first term represents the reconstruction likelihood and the second term ensures that our learned distribution q is similar to the true prior distribution p.
- which forces $q(z|x)$ to be similar to true prior distribution $p(z)$

VAE: STATISTICAL MOTIVATION

- $p(z)$ often assumed to be Guassian distribution
  determining $q(z|x)$ boils down to estimating μ and σ.
- Use neural network to estimate $q(z|x)$ and $p(z|x)$

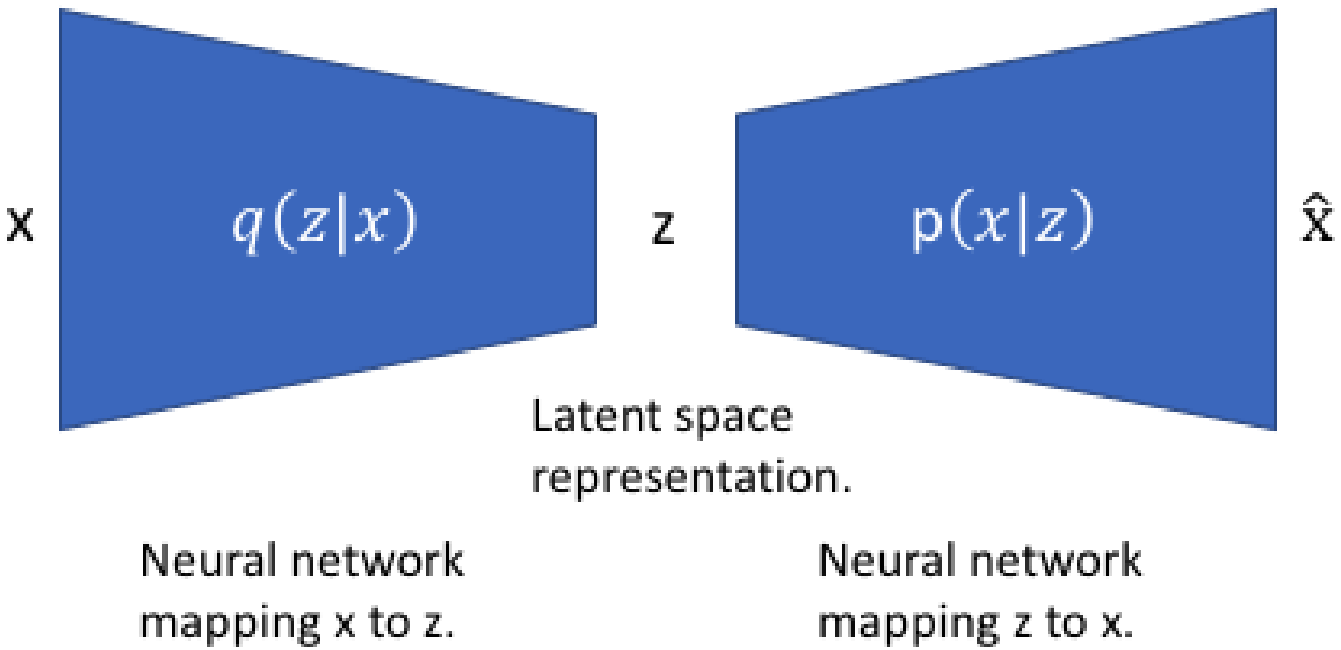


Figure: source: https://www.jeremyjordan.me/variational-autoencoders/

VAE TRAINING

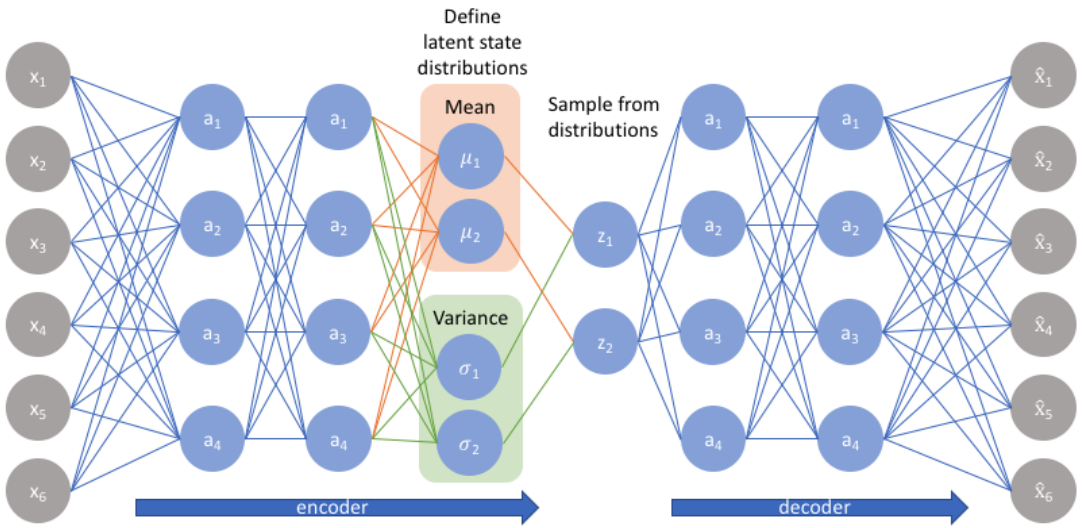


Figure: source: https://www.jeremyjordan.me/variational-autoencoders/

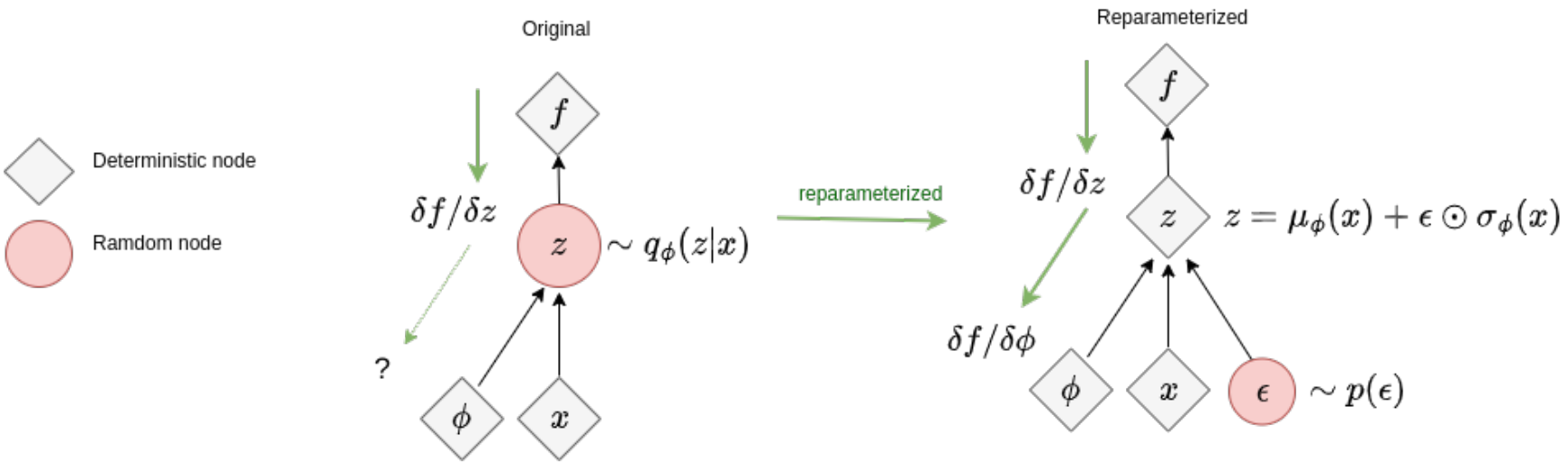
- Loss function:
  $L(\theta, \phi; x, z) = E_{q_\phi(z|x)} \log p_\theta(x|z) - KL(q_\phi(z|x)||p_\theta(z))$
- Problem: network contains **sampling** operator $\rightarrow$ we can not backpropagate through!

REPARAMETRIZATION TRICK

- We can not backpropagate through random sampling- what now?

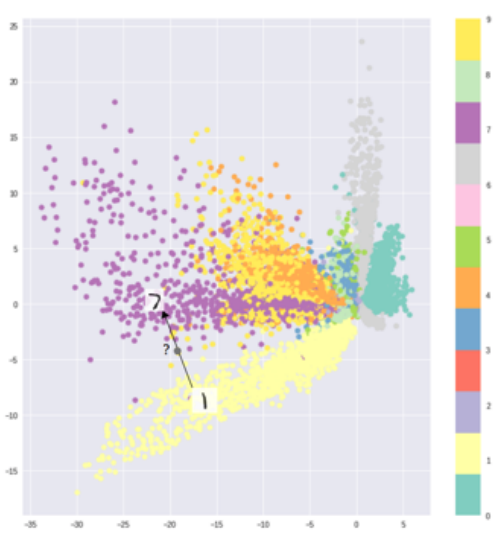
REPARAMETRIZATION TRICK

- We can not backpropagate through random sampling- what now?
- "Push" random sampling out of backpropagation path by reparametrization

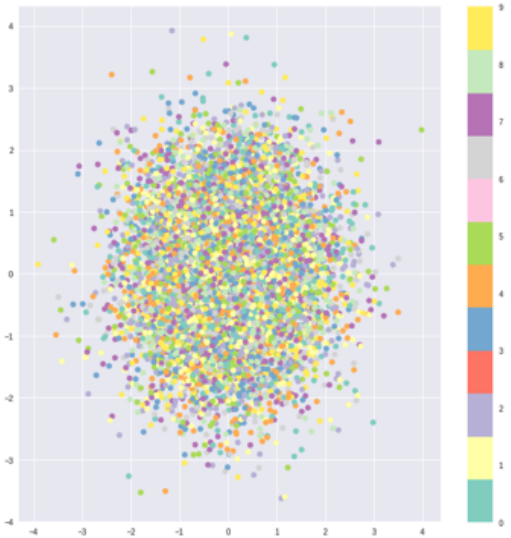


VISUALIZATION OF LATENT SPACE

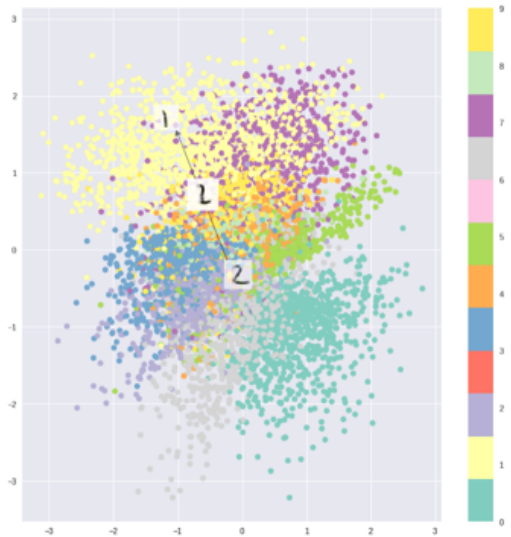
Only reconstruction loss



Only KL divergence



Combination



VARIATIONAL AUTOENCODERS AS A GENERATIVE MODEL

- New data can be generated by sampling from distributions in the latent space $\rightarrow$ reconstructed by decoder

VARIATIONAL AUTOENCODERS AS A GENERATIVE MODEL

- New data can be generated by sampling from distributions in the latent space $\rightarrow$ reconstructed by decoder
- Diagonal prior enforces independent latent variables $\rightarrow$ can encode different factors of variations

VARIATIONAL AUTOENCODERS AS A GENERATIVE MODEL

- New data can be generated by sampling from distributions in the latent space $\rightarrow$ reconstructed by decoder
- Diagonal prior enforces independent latent variables $\rightarrow$ can encode different factors of variations
- Examples of generated samples:

LATENT VARIABLES LEARNED BY A VAE

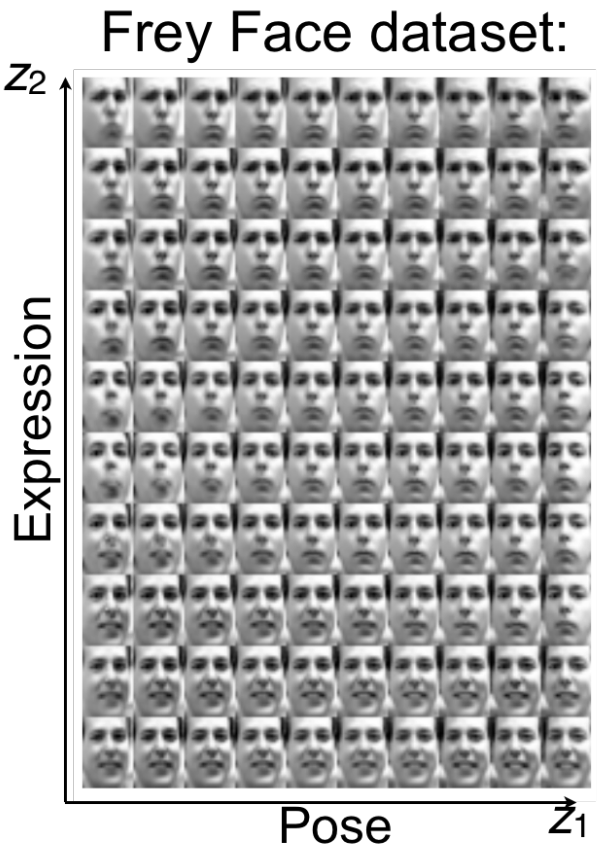


Figure: Images generated by a VAE are superimposed on the latent vectors that generated them. In the two-dimensional latent space of the VAE, the first dimension encodes the position of the face and the second dimension encodes the expression. Therefore, starting at any point in the latent space, if we move along either axis, the corresponding property will change in the generated image. (Goodfellow et al., 2016)

LATENT VARIABLES LEARNED BY A VAE

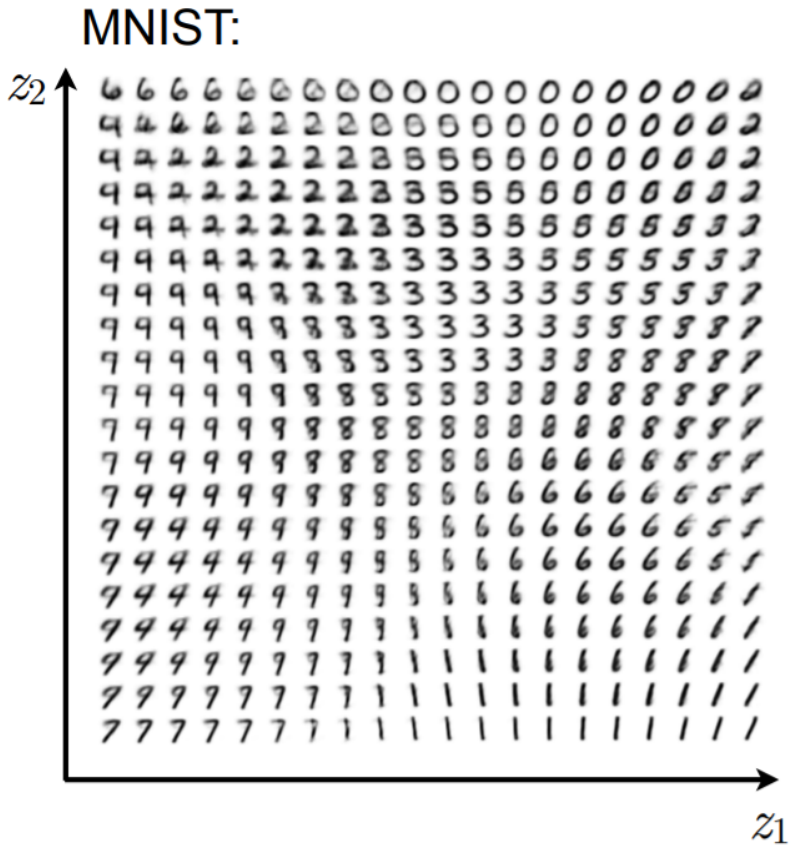
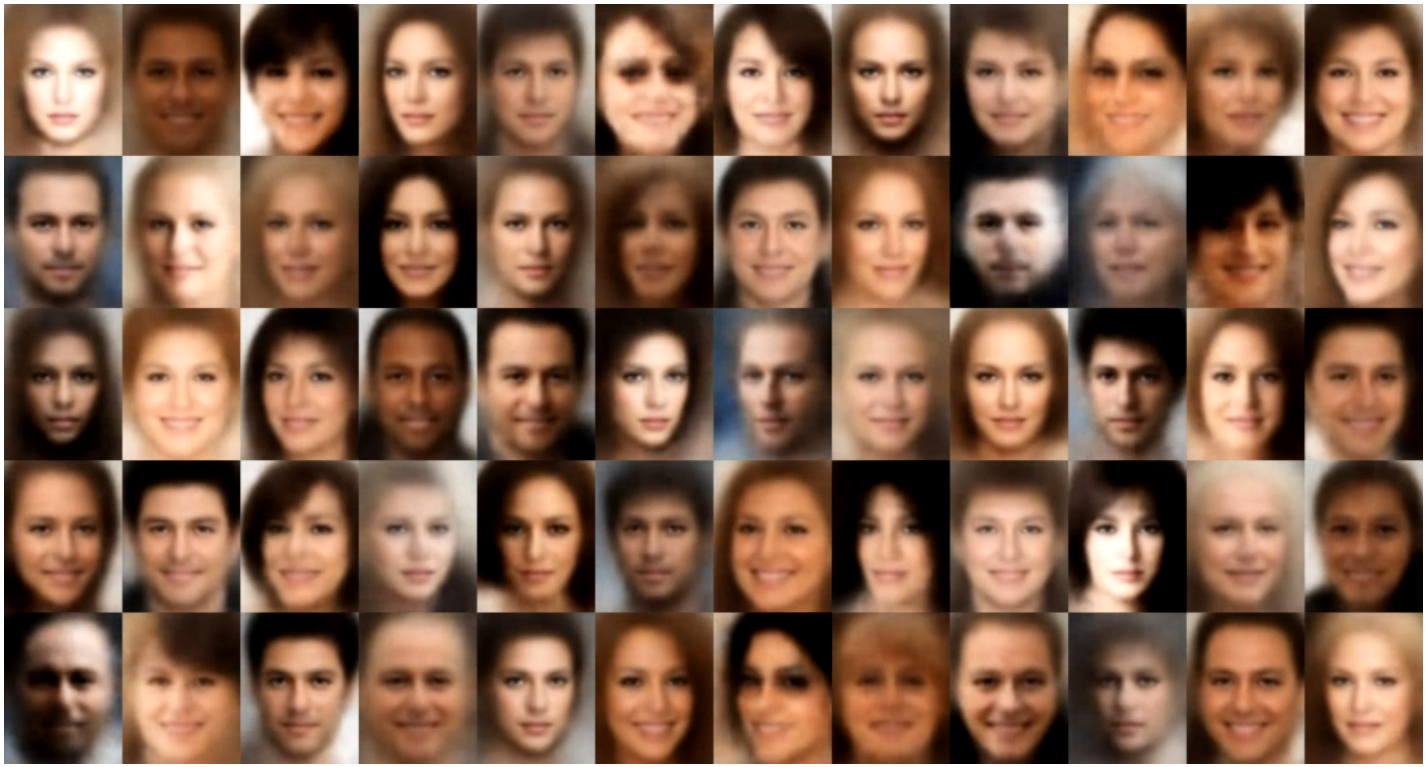


Figure: Images generated by a VAE are superimposed on the latent vectors that generated them. The two-dimensional latent space of the VAE captures much of the variation present in MNIST. Different regions in the latent space correspond to different digits in the generated images. (Goodfellow et al., 2016)

SAMPLES FROM A VANILLA VAE



Credit : Wojciech Mormul

Figure: Samples generated by a VAE that was trained on images of people's faces.

VARIATIONAL AUTOENCODERS: SUMMARY

- Probabilistic models $\rightarrow$ allow to generate data
- Intractable density $\rightarrow$ optimises variational lower bound instead
- Trained by back propagation by using reparametrization

VARIATIONAL AUTOENCODERS: SUMMARY

- Probabilistic models $\rightarrow$ allow to generate data
- Intractable density $\rightarrow$ optimises variational lower bound instead
- Trained by back propagation by using reparametrization
- Pros:
  - Principled approach to generative models
  - Latent space reparametrization can be useful for other tasks

VARIATIONAL AUTOENCODERS: SUMMARY

- Probabilistic models $\rightarrow$ allow to generate data
- Intractable density $\rightarrow$ optimises variational lower bound instead
- Trained by back propagation by using reparametrization
- Pros:
  - Principled approach to generative models
  - Latent space reparametrization can be useful for other tasks
- Cons:
  - Only maximizes lower bound of likelihood
  - Samples in standard models often of lower equality compared to GANs
- Active area of research!

REFERENCES

Jeremy Jordan (2018)

Variational Autoencoders

https://www.jeremyjordan.me/variational-autoencoders/

Lilian Weng (2018)

From Autoencoder to Beta-VAE

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

APPENDIX: LOSS FUNCTION ELBO

- In this section, we provide more details on loss function;
- In VAE, instead of mapping the input into a fixed vector, we want to map it into a distribution. Let's label this distribution as p_θ, parameterized by θ. The relationship between the data input x and the latent encoding vector z can be fully defined by:
  - Prior $p_\theta(\mathbf{z})$
  - Liklihood $p_\theta(\mathbf{x}|\mathbf{z})$
  - Posterior $p_\theta(\mathbf{z}|\mathbf{x})$

VAE-PARAMETER FITTING

Assuming that we know the real parameter θ^* for this distribution.
To generate a sample from a real data point $x^{(i)}$:

- sample $x^{(i)}$ from a prior distribution $p_{\theta^*}(z)$
- a value $x^{(i)}$ is generated from a conditional distribution $p_{\theta^*}(x|z = z^{(i)})$

The optimal parameter θ^* is the one that maximizes the probability of generating real data samples:

$$\theta^* = \arg\max_\theta \prod_{i=1}^{n} p_{\boldsymbol{\theta}}(x^{(i)})$$

we use the log probabilities to convert the product on right-hand side to a sum:

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

VAE-PARAMETER FITTING

Now let's update the equation to better demonstrate the data generation process so as to involve the encoding vector:

$$p_\theta(\mathbf{x}^{(i)}) = \int p_\theta(x^{(i)}|z)p_\theta(z)dz$$

Unfortunately it is not easy to compute $p_\theta(\mathbf{x}^{(i)})$ in this way, as it is very expensive to check all the possible values of z and sum them up.

VAE-PARAMETER FITTING: VARIATIONAL LOWER BOUND

- $\log p_\theta(\mathbf{x})$ is intractable.
- But we can compute a **variational lower bound**:

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int q(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z}$$

VAE-PARAMETER FITTING: VARIATIONAL LOWER BOUND

- $\log p_\theta(\mathbf{x})$ is intractable.
- But we can compute a **variational lower bound**:

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int q(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z}$$

$$\geq \int q(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z}$$

Jensen's inequality

Let f be a concave function and $\mathbf{x}$ an integrable random variable. Then it holds: $f(\mathbb{E}[\mathbf{x}]) \geq \mathbb{E}[f(\mathbf{x})]$.

VAE-PARAMETER FITTING: VARIATIONAL LOWER BOUND

- $\log p_\theta(\mathbf{x})$ is intractable.
- But we can compute a **variational lower bound**:

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \log \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int q(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&\geq \int q(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q(\mathbf{z}|\mathbf{x}) \big(\log p_\theta(\mathbf{x}|\mathbf{z}) + \log \frac{p_\theta(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \big) d\mathbf{z}
\end{aligned}$$

VAE-PARAMETER FITTING: VARIATIONAL LOWER BOUND

- $\log p_\theta(\mathbf{x})$ is intractable.
- But we can compute a **variational lower bound**:

$$
\begin{aligned}
\log p_\theta(\mathbf{x}) &= \log \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int q(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&\geq \int q(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q(\mathbf{z}|\mathbf{x}) \big(\log p_\theta(\mathbf{x}|\mathbf{z}) + \log \frac{p_\theta(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \big) d\mathbf{z} \\
&= \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \big[\log p_\theta(\mathbf{x}|\mathbf{z}) \big] - KL\big[q(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}) \big]
\end{aligned}
$$

VAE-PARAMETER FITTING: VARIATIONAL LOWER BOUND

- $\log p_\theta(\mathbf{x})$ is intractable.
- But we can compute a **variational lower bound**:

$$
\begin{aligned}
\log p_\theta(\mathbf{x}) &= \log \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int q(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&\geq \int q(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q(\mathbf{z}|\mathbf{x}) \big(\log p_\theta(\mathbf{x}|\mathbf{z}) + \log \frac{p_\theta(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \big) d\mathbf{z} \\
&= \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \big[\log p_\theta(\mathbf{x}|\mathbf{z}) \big] - KL\big[q(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}) \big]}_{:= ELBO(\boldsymbol{\theta}, , \mathbf{x})}
\end{aligned}
$$

VAE-PARAMETER FITTING: VARIATIONAL LOWER BOUND

$$ELBO(\theta, , \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - KL\big[q(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\big]$$

VAE-PARAMETER FITTING: VARIATIONAL LOWER BOUND

$$ELBO(\boldsymbol{\theta}, , \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\big[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\big] - KL\big[q(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})\big]$$

- Also known as **Evidence Lower BOund (ELBO)**.

VAE-PARAMETER FITTING: VARIATIONAL LOWER BOUND

$$ELBO(\theta, , \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - KL\left[q(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\right]$$

- Also known as **Evidence Lower BOund (ELBO)**.
- First term resembles reconstruction loss.

VAE-PARAMETER FITTING: VARIATIONAL LOWER BOUND

$$ELBO(\theta, , \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - KL\left[q(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\right]$$

- Also known as **Evidence Lower BOund (ELBO)**.
- First term resembles reconstruction loss.
- Second term penalizes encoder for deviating from prior.

VAE-PARAMETER FITTING: VARIATIONAL LOWER BOUND

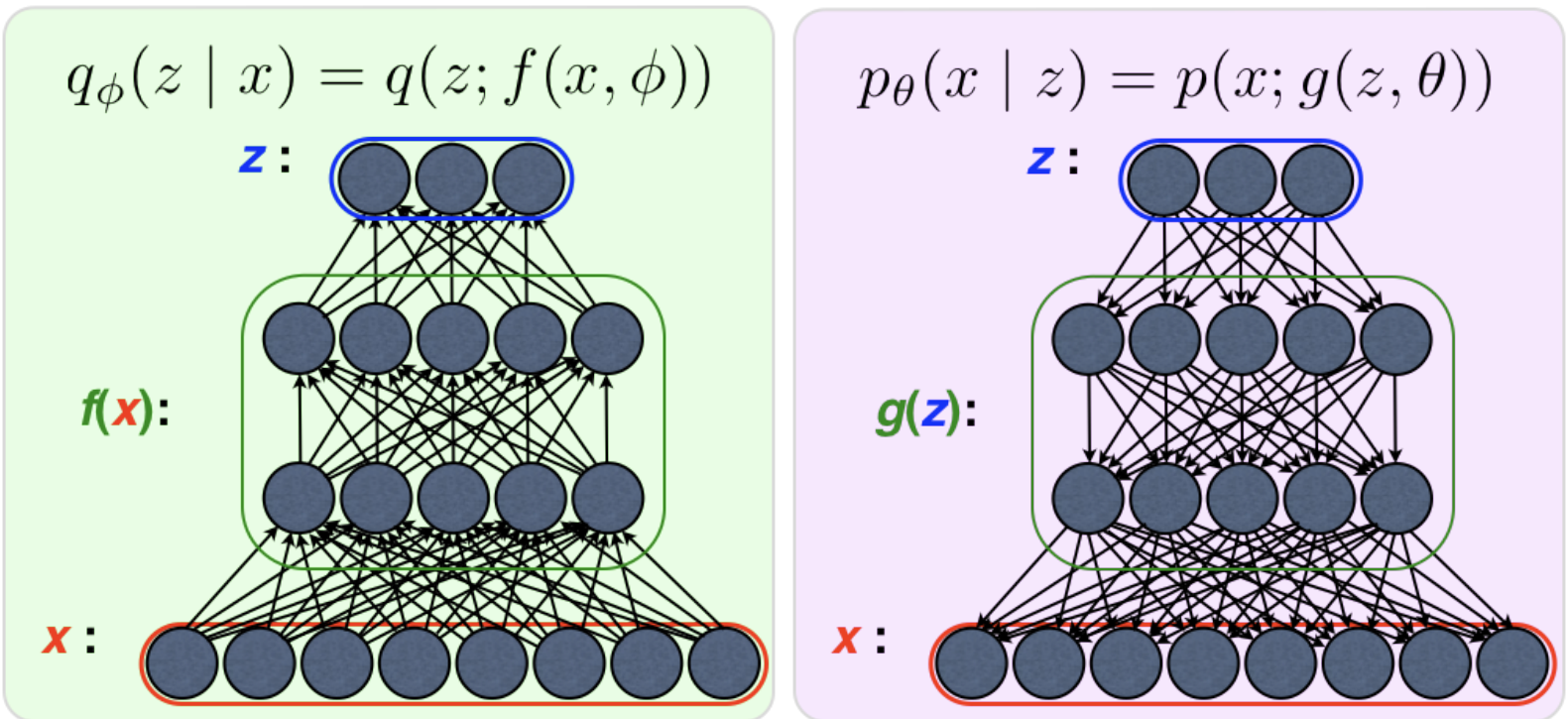
$$ELBO(\boldsymbol{\theta}, , \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\big[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\big] - KL\big[q(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})\big]$$

- Also known as **Evidence Lower BOund (ELBO)**.
- First term resembles reconstruction loss.
- Second term penalizes encoder for deviating from prior.
- It can be shown that
  $ELBO(\boldsymbol{\theta}, , \mathbf{x}) = \log p_{\boldsymbol{\theta}}(\mathbf{x}) - KL\big[q(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\big]$
  $\Rightarrow$ by maximizing the ELBO we maximize $p_{\boldsymbol{\theta}}(\mathbf{x})$ and minimize $KL\big[q(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\big]$.

VAE-MODEL DEFINITION

Idea:

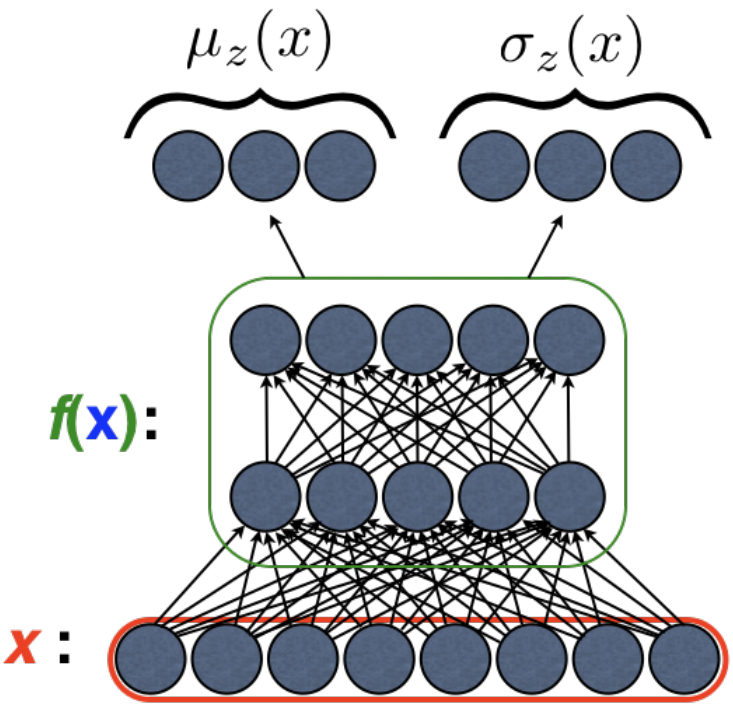
- Set $p_\theta(\mathbf{z})$ to some simple distribution.
- Parametrize inference model and generative model with neural networks $f(\mathbf{x},)$ and $g(\mathbf{z}, \theta)$.



VAE-MODEL DEFINITION

Usually:

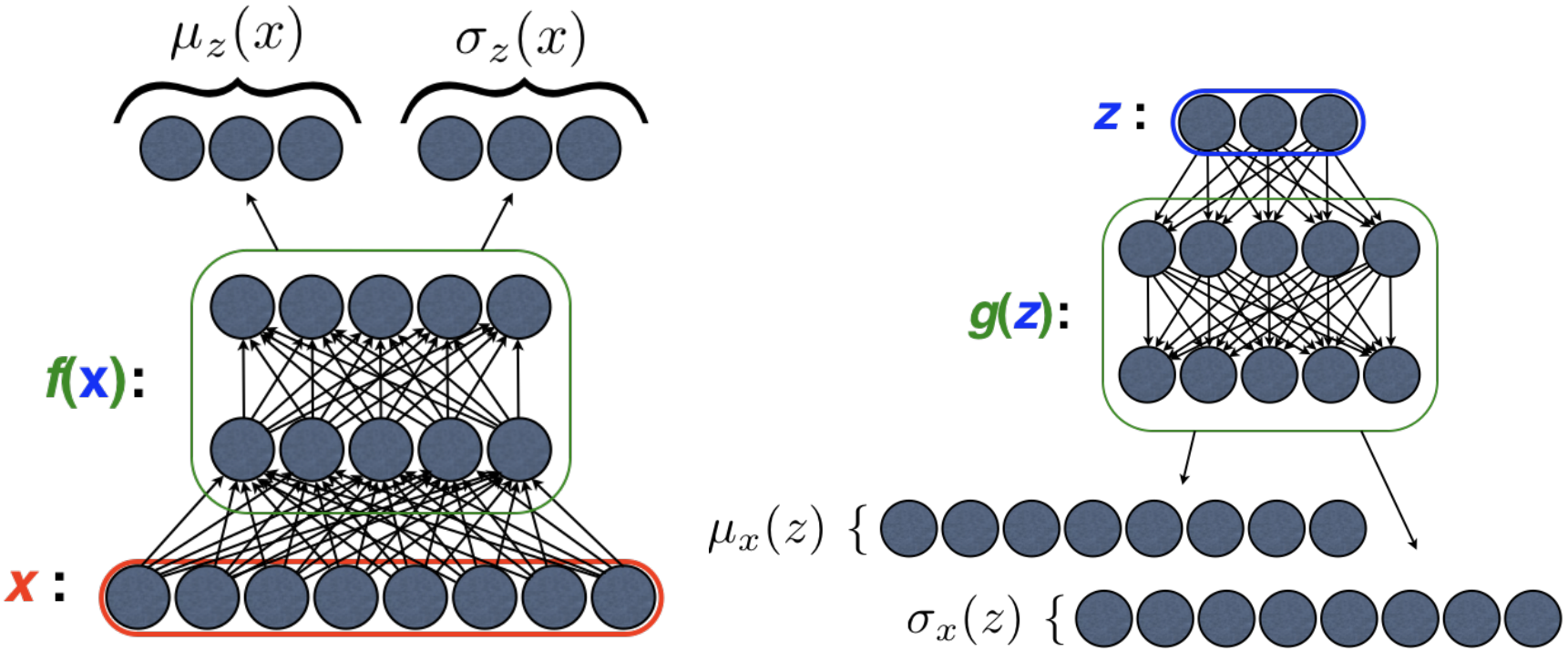
- $f(\mathbf{x},) = (\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{x}), \boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{x}))$ and $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\mathbf{z}}(\mathbf{x}), \boldsymbol{\sigma}^2_{\mathbf{z}}(\mathbf{x}))$



VAE-MODEL DEFINITION

Usually:

- $f(\mathbf{x},) = (\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{x}), \boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{x}))$ and $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\mathbf{z}}(\mathbf{x}), \boldsymbol{\sigma}^2_{\mathbf{z}}(\mathbf{x}))$
- $p_{\theta}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, 1)$
- $g(\mathbf{z}, \theta) = (\boldsymbol{\mu}_{\mathbf{x}}(\mathbf{z}), \boldsymbol{\sigma}_{\mathbf{x}}(\mathbf{z}))$ and $p_{\theta}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}}(\mathbf{z}), \boldsymbol{\sigma}^2_{\mathbf{x}}(\mathbf{z}))$



VAE-PARAMETER FITTING: REPARAMETERIZATION TRICK

- Goal: Learn parameters and θ by maximizing

$$ELBO(\theta, , \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - KL\big[q(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\big]$$

  based on gradient ascent.

- Idea: Approximate first term

$$\begin{aligned}\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] &= \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{z};\mu_\mathbf{z}(\mathbf{x}),\sigma_\mathbf{z}(\mathbf{x}))}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] \\ &\approx \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{x}|\mathbf{z}^{(l)})\end{aligned}$$

- Problem: Given this average, how should one take derivatives w.r.t. ?

VAE-PARAMETER FITTING: REPARAMETERIZATION TRICK

Recall: Linear transformation of a normal random variable

Let $\boldsymbol{\epsilon}$ be standard normally distributed, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{1})$. Then for $\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma} + \boldsymbol{\mu}$ it holds: $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$

VAE-PARAMETER FITTING: REPARAMETERIZATION TRICK

Recall: Linear transformation of a normal random variable

Let $\boldsymbol{\epsilon}$ be standard normally distributed, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{1})$. Then for $\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma} + \boldsymbol{\mu}$ it holds: $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$

Back to our problem:

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] = \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_\mathbf{z}(\mathbf{x}),\boldsymbol{\sigma}_\mathbf{z}(\mathbf{x}))}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]$$
$$\approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{x}|\mathbf{z}^{(l)})$$

Solution: Define $\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}_\mathbf{z}(\mathbf{x}) + \boldsymbol{\mu}_\mathbf{z}(\mathbf{x})$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{1})$.

VAE-PARAMETER FITTING: REPARAMETERIZATION TRICK

Recall: Linear transformation of a normal random variable

Let $\boldsymbol{\epsilon}$ be standard normally distributed, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{1})$. Then for $\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma} + \boldsymbol{\mu}$ it holds: $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$

Back to our problem:

$$\begin{aligned} \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] &= \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{1})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] \\ &\approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{x}|\mathbf{z}^{(l)}) \end{aligned}$$

Solution: Define $\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{x}) + \boldsymbol{\mu}_{\mathbf{z}}(\mathbf{x})$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{1})$.

VAE-PARAMETER FITTING: REPARAMETERIZATION TRICK

Recall: Linear transformation of a normal random variable

Let $\boldsymbol{\epsilon}$ be standard normally distributed, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{1})$. Then for $\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma} + \boldsymbol{\mu}$ it holds: $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$

Back to our problem:

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{1})}\left[\log p_\theta(\mathbf{x}|\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}_\mathbf{z}(\mathbf{x}) + \boldsymbol{\mu}_\mathbf{z}(\mathbf{x}))\right]$$
$$\approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{x}|\mathbf{z}^{(l)})$$

Solution: Define $\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}_\mathbf{z}(\mathbf{x}) + \boldsymbol{\mu}_\mathbf{z}(\mathbf{x})$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{1})$.

VAE-PARAMETER FITTING: REPARAMETERIZATION TRICK

Recall: Linear transformation of a normal random variable

Let $\boldsymbol{\epsilon}$ be standard normally distributed, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{1})$. Then for $\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma} + \boldsymbol{\mu}$ it holds: $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$

Back to our problem:

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{1})}\left[\log p_\theta(\mathbf{x}|\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}_\mathbf{z}(\mathbf{x}) + \boldsymbol{\mu}_\mathbf{z}(\mathbf{x}))\right]$$
$$\approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{x}|\mathbf{z}^{(l)} = \boldsymbol{\epsilon}^{(l)} \cdot \boldsymbol{\sigma}_\mathbf{z}(\mathbf{x}) + \boldsymbol{\mu}_\mathbf{z}(\mathbf{x}))$$

Solution: Define $\mathbf{z} = \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}_\mathbf{z}(\mathbf{x}) + \boldsymbol{\mu}_\mathbf{z}(\mathbf{x})$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{1})$.