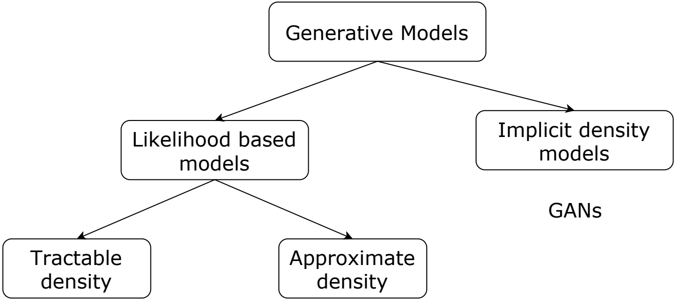
Deep Generative Models
Lecture 2

Roman Isachenko

Ozon Masters

Spring, 2021

Generative models zoo



- Autoregressive models
- Flow models

Latent variable models

Bayesian framework

- $\mathbf{x}$ – samples;
- $\mathbf{y}$ – target variables;
- $\boldsymbol{\theta}$ – model parameters.

Discriminative

$$p(\mathbf{y}, \boldsymbol{\theta} | \mathbf{x}) = p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta})$$

- Find conditional probability of $\mathbf{y}$ given $\mathbf{x}$.
- Samples $\mathbf{x}$ are given.
- Used for classification, regression.

Generative

$$p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}) = p(\mathbf{y}, \mathbf{x} | \boldsymbol{\theta}) p(\boldsymbol{\theta})$$

- Find joint probability of $(\mathbf{x}, \mathbf{y})$.
- Samples $\mathbf{x}$ should be modelled.
- Generation of new samples $(\mathbf{x}, \mathbf{y})$.

Generative models

We age given samples $\{\mathbf{x}_i\}_{i=1}^n \in X$ from unknown distribution $p(\mathbf{x})$.

Goal

We would like to learn a distribution $p(\mathbf{x})$ for

- evaluating $p(\mathbf{x})$ for new samples;
- sampling from $p(\mathbf{x})$.

Challenge

Data is complex and high-dimensional (curse of dimensionality).

Solution

Fix probabilistic model $p(\mathbf{x}|\boldsymbol{\theta})$ – the set of parameterized distributions .

Instead of searching true $p(\mathbf{x})$ over all probability distributions, learn function approximation $p(\mathbf{x}|\boldsymbol{\theta}) \approx p(\mathbf{x})$.

Latent variable models (LVM)

Suppose that our probabilistic model is $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ instead of $p(\mathbf{x}|\boldsymbol{\theta})$.

- Here $\mathbf{z}$ are latent variables.
- We observe only samples $\mathbf{x}$.
- Latent variables $\mathbf{z}$ are unknown.
- Parameters $\boldsymbol{\theta}$ are not random.

MLE problem for LVM

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{n} p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}) =$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}).$$

What if $\boldsymbol{\theta}$ are random variables with distribution $p(\boldsymbol{\theta})$?

Bayesian framework

What if $\boldsymbol{\theta}$ are random variables with distribution $p(\boldsymbol{\theta})$?
Before we get any data, we do not know anything about $\boldsymbol{\theta}$ except the **prior** distribution $p(\boldsymbol{\theta})$.
When we get data, we could change the **prior** distribution to the **posterior**.

Bayes theorem

$$p(\boldsymbol{\theta}|\mathbf{X},\mathbf{Z}) = \frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{X},\mathbf{Z})} = \frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

Full Bayesian inference

$$p(\mathbf{x}^*|\mathbf{X},\mathbf{Z}) = \int p(\mathbf{x}^*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X},\mathbf{Z})d\boldsymbol{\theta}$$

Bayesian framework

Full Bayesian inference

$$p(\mathbf{x}^*|\mathbf{X},\mathbf{Z}) = \int p(\mathbf{x}^*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X},\mathbf{Z})d\boldsymbol{\theta}$$

Maximum a posteriori (MAP)

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{X},\mathbf{Z}) = \arg\max_{\boldsymbol{\theta}} \big(\log p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\big)$$

$$p(\mathbf{x}^*|\mathbf{X},\mathbf{Z}) = \int p(\mathbf{x}^*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X},\mathbf{Z})d\boldsymbol{\theta} \approx p(\mathbf{x}^*|\boldsymbol{\theta}^*).$$

Latent variable models

MLE problem

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\mathbf{X}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{n} p(\mathbf{x}_i|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(\mathbf{x}_i|\boldsymbol{\theta}).$$

Challenge

$p(\mathbf{x}|\boldsymbol{\theta})$ could be intractable.

Extend probabilistic model

Introduce latent variable $\mathbf{z}$ for each sample $\mathbf{x}$

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z}); \quad \log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \log p(\mathbf{z}).$$

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} = \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z}) d\mathbf{z}.$$

Motivation

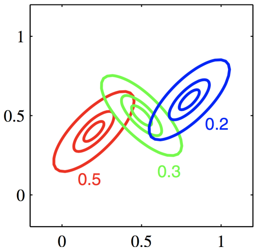
The distributions $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ and $p(\mathbf{z})$ could be quite simple.

Latent variable models

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z}) d\mathbf{z} \to \max_{\boldsymbol{\theta}}$$

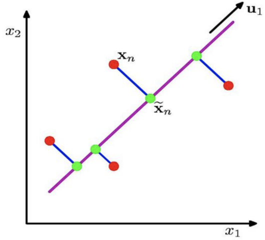
Examples

Mixture of gaussians



- $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{z}})$
- $p(\mathbf{z}) = \text{Categorical}(\mathbf{z}|\pi)$

PCA model



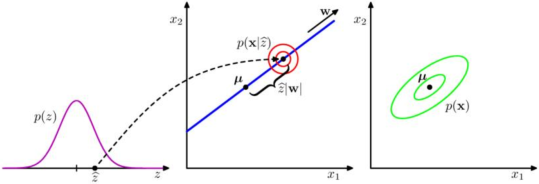
- $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Sigma}_{\mathbf{z}})$
- $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I})$

Bishop C. Pattern Recognition and Machine Learning, 2006

Latent variable models

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z}) d\mathbf{z} \to \max_{\boldsymbol{\theta}}$$

PCA goal: Project original data $\mathbf{X}$ onto low latent space while maximizing the variance of the projected data.



$$\boldsymbol{x} = W\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$$

- $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Sigma}_\mathbf{z})$
- $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I})$

Bishop C. Pattern Recognition and Machine Learning, 2006

Incomplete likelihood

MLE

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{n} p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}) =$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}).$$

Since $\mathbf{Z}$ is unknown, maximize **incomplete likelihood**.

MILE problem

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{X}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(\mathbf{x}_i|\boldsymbol{\theta}) =$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log \int p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}) d\mathbf{z}_i =$$

$$= \arg\max_{\boldsymbol{\theta}} \log \int p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) p(\mathbf{z}_i) d\mathbf{z}_i.$$

Variational lower bound

$$
\begin{aligned}
\log p(\mathbf{x}|\boldsymbol{\theta}) &= \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})} = \\
&= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})} d\mathbf{z} = \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) q(\mathbf{z})} d\mathbf{z} = \\
&= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} + \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})} d\mathbf{z} = \\
&= \mathcal{L}(q, \boldsymbol{\theta}) + KL(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}(q, \boldsymbol{\theta}).
\end{aligned}
$$

Kullback-Leibler divergence

- $KL(q||p) = \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z})} d\mathbf{z}$;
- $KL(q||p) \geq 0$;
- $KL(q||p) = 0 \Leftrightarrow q \equiv p$.

Variational lower bound

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}(q, \boldsymbol{\theta}).$$

Evidence Lower Bound (ELBO)

$$\begin{aligned}
\mathcal{L}(q, \boldsymbol{\theta}) &= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \\
&= \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) d\mathbf{z} + \int q(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z} \\
&= \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z}))
\end{aligned}$$

Instead of maximizing incomplete likelihood, maximize ELBO (equivalently minimize KL)

$$\max_{\theta} p(\mathbf{x}|\boldsymbol{\theta}) \quad \rightarrow \quad \max_{q, \theta} \mathcal{L}(q, \boldsymbol{\theta}) \equiv \min_{q, \theta} KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

EM-algorithm

$$\mathcal{L}(q, \boldsymbol{\theta}) = \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) d\mathbf{z} + \int q(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z}.$$

Block-coordinate optimization

- Initialize $\boldsymbol{\theta}^*$;
- E-step

$$q(\mathbf{z}) = \arg\max_q \mathcal{L}(q, \boldsymbol{\theta}^*) = \arg\min_q KL(q||p) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*);$$

- M-step

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(q, \boldsymbol{\theta});$$

- Repeat E-step and M-step until convergence.

Ugly pic

Amortized variational inference

E-step

$$q(\mathbf{z}) = \arg\max_q \mathcal{L}(q, \boldsymbol{\theta}^*) = \arg\min_q KL(q||p) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*).$$

could be **intractable**.

Idea

Restrict a family of all possible distributions $q(\mathbf{z})$ to a particular parametric class conditioned on sample: $q(\mathbf{z}|\mathbf{x}, \phi)$.

Variational Bayes

- E-step

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}_{k-1})|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}}$$

- M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}_k, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}}$$

Variational EM-algorithm

ELBO

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}(q, \boldsymbol{\theta}).$$

- E-step
$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}_{k-1})\big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}},$$
where ϕ – parameters of variational distribution $q(\mathbf{z}|\mathbf{x}, \phi)$.
- M-step
$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}_k, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}},$$
where $\boldsymbol{\theta}$ – parameters of likelihood $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

Now all we have to do is to obtain two gradients $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\phi, \boldsymbol{\theta})$, $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\phi, \boldsymbol{\theta})$.
Difficulty: number of samples n.

ELBO gradient (M-step, $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$)

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z})) \to \max_{\boldsymbol{\phi}, \boldsymbol{\theta}}.$$

Optimization w.r.t. $\boldsymbol{\theta}$: **mini-batching** (1) + **Monte-Carlo** estimation (2)

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) &= \sum_{i=1}^{n} \int q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi}) \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) d\mathbf{z}_i \\ &\overset{(1)}{\approx} n \int q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi}) \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) d\mathbf{z}_i, \quad i \sim U[1, n] \\ &\overset{(2)}{\approx} n \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i|\mathbf{z}_i^*, \boldsymbol{\theta}), \quad \mathbf{z}_i^* \sim q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi}).\end{aligned}$$

Monte-Carlo estimation (2):

$$\int q(\mathbf{z}) f(\mathbf{z}) d\mathbf{z} \approx f(\mathbf{z}^*), \text{where } \mathbf{z}^* \sim q(\mathbf{z}).$$

ELBO gradient (E-step, $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$)

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z})) \rightarrow \max_{\boldsymbol{\phi}, \boldsymbol{\theta}}.$$

Difference from M-step: density function $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ depends on the parameters $\boldsymbol{\phi}$, it is impossible to use the Monte-Carlo estimation:

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \int \nabla_{\boldsymbol{\phi}} q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) d\mathbf{z} - \nabla_{\boldsymbol{\phi}} KL$$

Log-derivative trick

$$\nabla_\xi q(\eta|\xi) = q(\eta|\xi) \left(\frac{\nabla_\xi q(\eta|\xi)}{q(\eta|\xi)} \right) = q(\eta|\xi) \nabla_\xi \log q(\eta|\xi).$$

$$\nabla_{\boldsymbol{\phi}} q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) = q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) \nabla_{\boldsymbol{\phi}} \log q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}).$$

ELBO gradient (E-step, $\nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta})$)

$$\nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta}) = \int \nabla_\phi q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) d\mathbf{z} - \nabla_\phi KL =$$

$$= \int q(\mathbf{z}|\mathbf{x}, \phi) \left[\nabla_\phi \log q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})\right] d\mathbf{z} - \nabla_\phi KL$$

After applying the log-reparametrization trick, we are able to use the Monte-Carlo estimation:

$$\nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta}) \approx n \nabla_\phi \log q(\mathbf{z}_i^*|\mathbf{x}_i, \phi) \log p(\mathbf{x}_i|\mathbf{z}_i^*, \boldsymbol{\theta}) - \nabla_\phi KL,$$

$$\mathbf{z}_i^* \sim q(\mathbf{z}_i|\mathbf{x}_i, \phi).$$

Problem
Unstable solution with huge variance.

Solution
Reparametrization trick

ELBO gradient (E-step, $\nabla_{\phi}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$)

Reparametrization trick

$$f(\xi) = \int q(\eta|\xi)h(\eta)d\eta$$

Let $\eta = g(\xi, \epsilon)$, where g is a deterministic function, ϵ is a random variable with a density function $r(\epsilon)$.

$$\begin{aligned}\nabla_{\xi} \int q(\eta|\xi)h(\eta)d\eta = \nabla_{\xi} \int r(\epsilon)h(g(\xi, \epsilon))d\epsilon \\ \approx \nabla_{\xi} h(g(\xi, \epsilon^*)), \quad \epsilon^* \sim r(\epsilon).\end{aligned}$$

Example

$$q(\eta|\xi) = \mathcal{N}(\eta|\mu, \sigma^2), \quad r(\epsilon) = \mathcal{N}(\epsilon|0, 1), \quad \eta = \sigma \cdot \epsilon + \mu, \quad \xi = [\mu, \sigma].$$

ELBO gradient (E-step, $\nabla_{\phi}\mathcal{L}(\phi, \boldsymbol{\theta})$)

$$\begin{aligned}\nabla_{\phi}\mathcal{L}(\phi, \boldsymbol{\theta}) &= \nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) d\mathbf{z} - \nabla_{\phi} KL \\ &\approx n \nabla_{\phi} \int r(\epsilon) \log p(\mathbf{x}_i | g(\mathbf{x}_i, \epsilon, \phi), \boldsymbol{\theta}) d\epsilon - \nabla_{\phi} KL \\ &\approx n \nabla_{\phi} \log p(\mathbf{x}_i | g(\mathbf{x}_i, \boldsymbol{\epsilon}^*, \phi), \boldsymbol{\theta}) - \nabla_{\phi} KL, \quad \boldsymbol{\epsilon}^* \sim r(\boldsymbol{\epsilon}).\end{aligned}$$

Variational assumption

$$q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\Sigma}(\mathbf{x})).$$

$$\mathbf{z} = g(\mathbf{x}, \epsilon, \phi) = \sqrt{\boldsymbol{\Sigma}(\mathbf{x})} \cdot \epsilon + \boldsymbol{\mu}(\mathbf{x}).$$

$\nabla_{\phi} KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}))$ has an analytical solution.

Variational autoencoder (VAE)

Final algorithm

- pick $i \sim U[1, n]$;
- compute stochastic gradient w.r.t. ϕ

$$\nabla_{\phi}\mathcal{L}(\phi, \boldsymbol{\theta}) = n\nabla_{\phi} \log p(\mathbf{x}_i | g(\mathbf{x}_i, \boldsymbol{\epsilon}^*, \phi), \boldsymbol{\theta}) - \\ - \nabla_{\phi} KL(q(\mathbf{z}_i | \mathbf{x}_i, \phi) || p(\mathbf{z}_i)), \quad \boldsymbol{\epsilon}^* \sim r(\boldsymbol{\epsilon});$$

- compute stochastic gradient w.r.t. $\boldsymbol{\theta}$

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\phi, \boldsymbol{\theta}) = n\nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i | \mathbf{z}_i^*, \boldsymbol{\theta}), \quad \mathbf{z}_i^* \sim q(\mathbf{z}_i | \mathbf{x}_i, \phi);$$

- update $\boldsymbol{\theta}, \phi$ according to the selected optimization method (SGD, Adam, RMSProp).

Variational autoencoder (VAE)

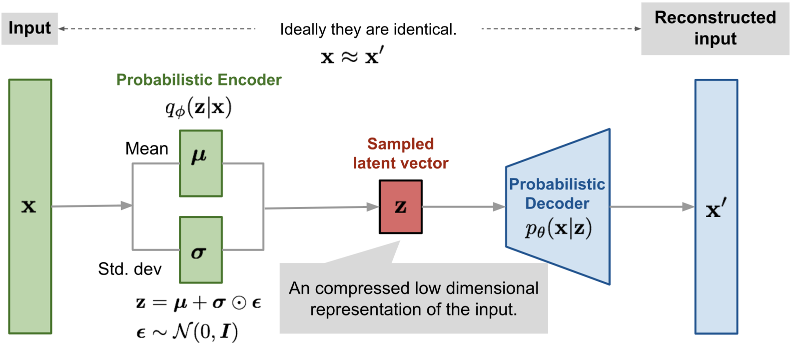


image credit: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Variational Autoencoder

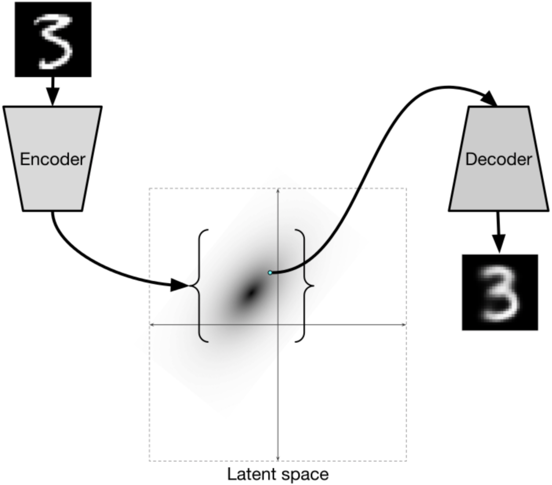


image credit: http://ijdykeman.github.io/ml/2016/12/21/cvae.html

Variational Autoencoder

Generation objects by sampling the latent space $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$

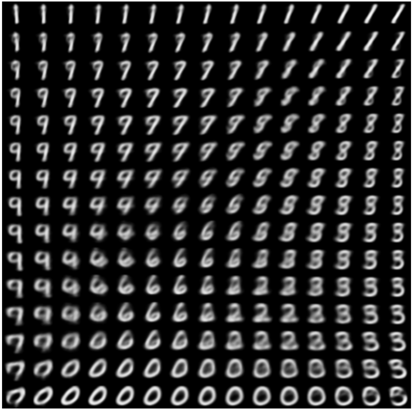


image credit: *https://habr.com/ru/post/331664*

Bayesian framework

Bayes theorem

$$p(\mathbf{t}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{t})p(\mathbf{t})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathbf{t})p(\mathbf{t})}{\int p(\mathbf{x}|\mathbf{t})p(\mathbf{t})d\mathbf{t}}$$

- $\mathbf{x}$ – observed variables;
- $\mathbf{t}$ – unobserved variables (latent variables/parameters);
- $p(\mathbf{x}|\mathbf{t})$ – likelihood;
- $p(\mathbf{x})$ – evidence;
- $p(\mathbf{t})$ – prior;
- $p(\mathbf{t}|\mathbf{x})$ – posterior.

Summary