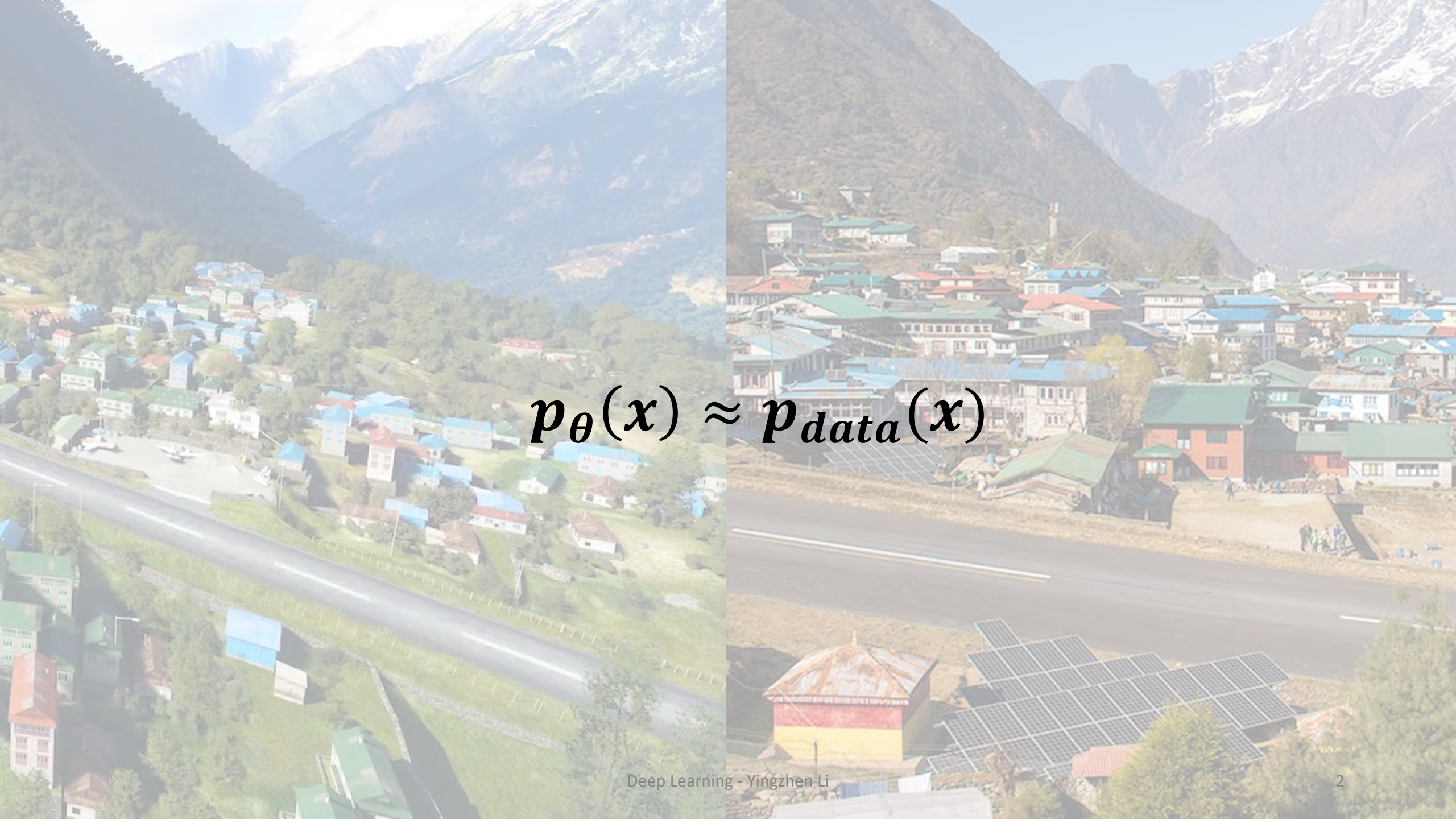
Generative Models

VAE basics

Yingzhen Li (yingzhen.li@imperial.ac.uk)



$$p_\theta(x) \approx p_{data}(x)$$

Divergence minimisation

- Fitting the model to the data by divergence minimisation:

$$\theta^* = argmin\, D[p_{data}(x) \,||\, p_\theta(x)]$$

$D[p \,||\, q]$ is a valid divergence if and only if:

- $D[p \,||\, q] = 0 \Rightarrow p = q$
- $p = q \Rightarrow D[p \,||\, q]$
  (or $p(x) \neq \mathrm{q}(x) \Rightarrow D[p \,||\, q] > 0$)

Divergence minimisation

- An example of a valid divergence: Kullback-Leibler (KL) divergence:

$$KL[p(x)\,||\,q(x)] = E_{p(x)}\left[\log\frac{p(x)}{q(x)}\right]$$

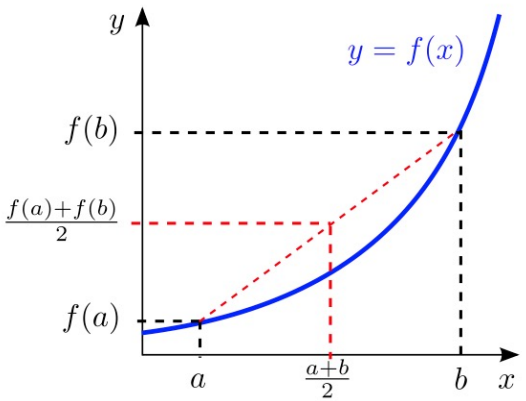
- $p(x) = q(x) \Rightarrow KL[p||q] = 0$
- To show $p(x) \neq q(x) \Rightarrow KL[p||q] > 0$:

$$\begin{aligned} -KL[p||q] &= -E_{p(x)}\left[\log\frac{p(x)}{q(x)}\right] \\ &= E_{p(x)}\left[\log\frac{q(x)}{p(x)}\right] \\ &\leq \log E_{p(x)}[\,q(x)/p(x)\,] \quad \text{(Jensen's inequality)} \\ &= \log 1 = 0 \end{aligned}$$

(equality holds iff. $p(x) = q(x)$)

Jensen's inequality:
Let f, g be two functions and f is convex, then

$$E_{p(x)}[f(g(x))] \geq f(E_{p(x)}[g(x)])$$



Maximum Likelihood Estimation

- Fitting the model by minimising KL:

$$\theta^* = \text{argmin}\, KL[p_{data}(x) \,||\, p_\theta(x)]$$

$$KL[p_{data}(x) \,||\, p_\theta(x)] = E_{p_{data}(x)}\left[\log \frac{p_{data}(x)}{p_\theta(x)}\right]$$

Constant terms w.r.t. θ

$$= -E_{p_{data}(x)}[\log p_\theta(x)] + E_{p_{data}(x)}[\log p_{data}(x)]$$

- Equivalent objective to fit the model: maximum likelihood estimation (MLE)

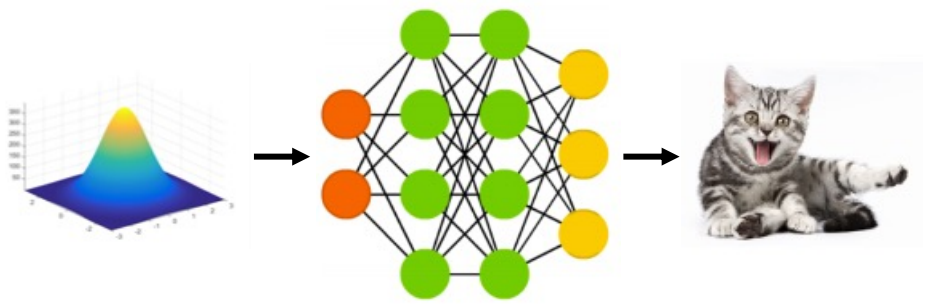
$$\theta^* = argmax\, E_{p_{data}(x)}[\log p_\theta(x)]$$

In practice: $$\theta^* = argmax\, \frac{1}{N}\sum_{n=1}^{N} \log p_\theta(x), x_1, \ldots, x_N \sim p_{data}(x)$$

Maximum Likelihood Estimation

- MLE for training latent variable models (LVM):

$$p(z) = N(z; 0, I)$$
$$p_\theta(x|z) = N(x; G_\theta(z), \sigma^2 I)$$



$$\theta^* = argmax\ E_{p_{data}(x)}[\log p_\theta(x)]$$

integrate over all possible z

$$p_\theta(x) = \int p_\theta(x|z)p(z)dz$$

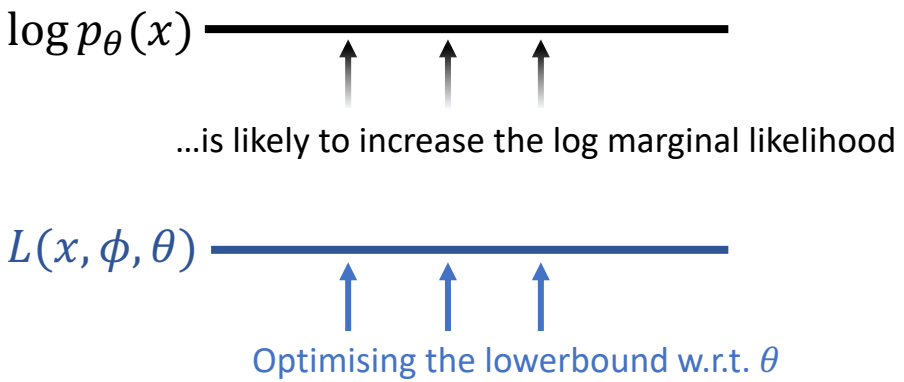
requires passing z through a neural net

The marginal distribution $p_\theta(x)$ is intractable
⇒ MLE objective is intractable

Variational Auto-Encoders

- MLE for latent variable model training is intractable
- Optimising a variational lower-bound instead:

$$\begin{aligned}
\log p_\theta(x) &= \log \int p_\theta(x|z)p(z)dz \\
&= \log \int q_\phi(z|x)\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}dz \\
&\geq \int q_\phi(z|x)\log\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}dz \qquad \text{(Jensen's inequality)} \\
&= E_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL[q_\phi(z|x)\,||\,p(z)] \\
&:= L(x,\phi,\theta)
\end{aligned}$$



Kingma and Welling. Auto-encoding variational Bayes. ICLR 2014

Rezende et al. Stochastic backpropagation and approximate inference in deep generative model. ICML 2014

Variational Auto-Encoders

- An alternative way to derive the variational lower-bound:

$$\log p_\theta(x) - KL[q_\phi(z|x) \,||\, p_\theta(z|x)]$$

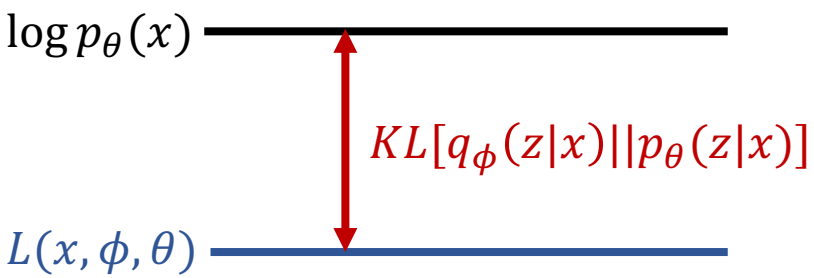
$$= \log p_\theta(x) - E_{q_\phi(z|x)}[\log q_\phi(z|x) - \log p_\theta(z|x)] \quad \text{(Bayes' rule)}$$

$$= \log p_\theta(x) - E_{q_\phi(z|x)}[\log q_\phi(z|x) - \log \frac{p_\theta(x|z)p(z)}{p_\theta(x)}]$$

$$= \log p_\theta(x) + E_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} - \log p_\theta(x)\right]$$

$$= E_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right]$$

$$:= L(x, \phi, \theta)$$



Optimising $L(x, \phi, \theta)$ w.r.t. ϕ
$\Rightarrow$ minimising $KL[q_\phi(z|x) \,||\, p_\theta(z|x)]$
$\Rightarrow$ fit $q_\phi(z|x)$ to the posterior $p_\theta(z|x)$

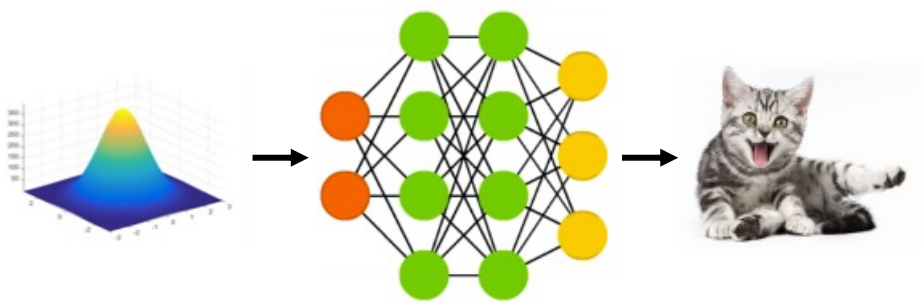
Variational Auto-Encoders

- The VAE objective:

$$\theta^*, \phi^* = argmax\ L(\phi, \theta)$$

$$L(\phi, \theta) \coloneqq E_{p_{data}(x)}[E_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL[q_\phi(z|x)\ ||\ p(z)]]$$

$$= -\frac{1}{2\sigma^2}\|x - G_\theta(z)\|_2^2 + C$$



$$p(z) = N(z; 0, I)$$
$$p_\theta(x|z) = N(x; G_\theta(z), \sigma^2 I)$$

Ingredients of training VAEs:

- The generative model (decoder)
- The $q_\phi(z|x)$ distribution (encoder)
- The optimisation procedure

Kingma and Welling. Auto-encoding variational Bayes. ICLR 2014
Rezende et al. Stochastic backpropagation and approximate inference in deep generative model. ICML 2014

Designing the q distribution

- Common choice: factorized Gaussian distribution:

$$q_\phi(z|x) = N(z; \mu_\phi(x), diag(\sigma_\phi^2(x)))$$

  - $\mu_\phi(x)$ and $\sigma_\phi(x)$ are parameterized by neural networks parameterized by ϕ, for example:

$$\mu_\phi(x) = NN_{\phi_1}(x), \quad \log \sigma_\phi(x) = NN_{\phi_2}(x)$$

to ensure the variance is non-negative

  - Analytic form for the KL regularizer: with $p(z) = N(z; 0, I)$ and $z \in R^d$

$$KL[q_\phi(z|x) \,||\, p(z)] = \frac{1}{2}(\|\mu_\phi(x)\|_2^2 + \|\sigma_\phi(x)\|_2^2 - d - 2\langle \log \sigma_\phi(x), 1\rangle)$$

Kingma and Welling. Auto-encoding variational Bayes. ICLR 2014
Rezende et al. Stochastic backpropagation and approximate inference in deep generative model. ICML 2014

Reparameterisation trick

- The VAE objective:

$$\theta^*, \phi^* = argmax\ L(\phi, \theta)$$

analytic between two Gaussians

$$L(\phi, \theta) \coloneqq E_{p_{data}(x)}[E_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL[q_\phi(z|x)\ ||\ p(z)]]$$

intractable expectation

- Monte Carlo (MC) estimation:

$$E_{q_\phi(z|x)}[\log p_\theta(x|z)] \approx \log p_\theta(x \mid z), \quad z \sim q_\phi(z|x)$$

differentiate to obtain (MC) gradient w.r.t. θ

how about the gradient w.r.t. ϕ?

Kingma and Welling. Auto-encoding variational Bayes. ICLR 2014
Rezende et al. Stochastic backpropagation and approximate inference in deep generative model. ICML 2014

Reparameterisation trick

- Monte Carlo (MC) estimation:

$$E_{q_\phi(z|x)}[\log p_\theta(x|z)] \approx \log p_\theta(x \mid z), \quad z \sim q_\phi(z|x)$$

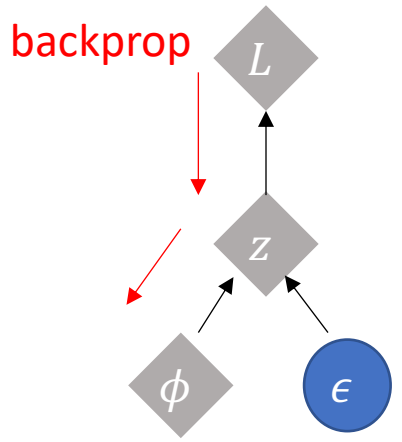
  - With Gaussian encoder:

$$z \sim q_\phi(z|x) \quad \Leftrightarrow \quad z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \quad \epsilon \sim N(\epsilon; 0, I)$$

  - Writing $z = T_\phi(x, \epsilon) \coloneqq \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$:

$$E_{q_\phi(z|x)}[\log p_\theta(x|z)] \approx \log p_\theta(x \mid T_\phi(x, \epsilon)), \quad \epsilon \sim N(\epsilon; 0, I)$$

differentiate to obtain (MC) gradient w.r.t. ϕ



Kingma and Welling. Auto-encoding variational Bayes. ICLR 2014
Rezende et al. Stochastic backpropagation and approximate inference in deep generative model. ICML 2014

Variational Auto-Encoders

- Combining all the ingredients together:

$$\theta^*, \phi^* = argmax\ L(\phi, \theta)$$

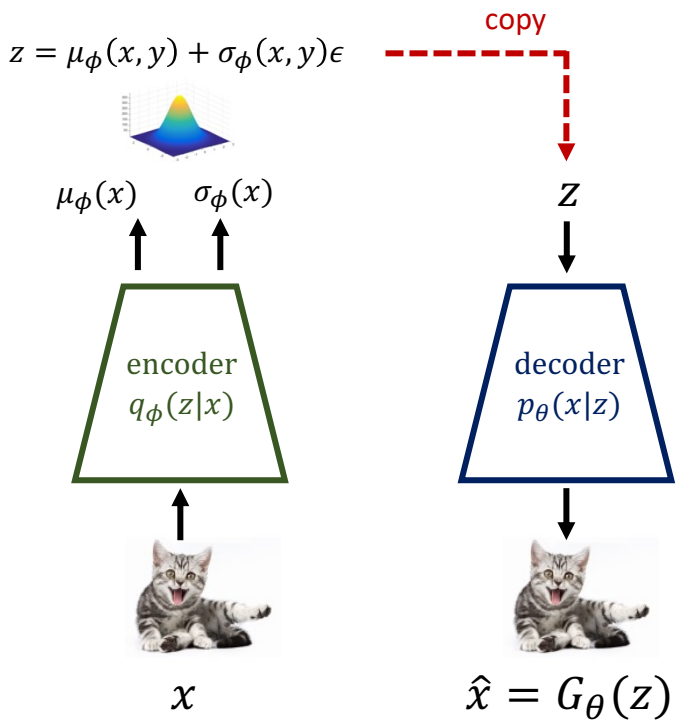
Reconstruction loss

$$L(\phi, \theta) \coloneqq E_{p_{data}(x)}\{ -E_{N(\epsilon;0,I)}\left[\frac{1}{2\sigma^2} \parallel G_\theta\left(T_\phi(x,\epsilon)\right) - x \parallel_2^2\right]$$

stochastic auto-encoder

$$-KL[q_\phi(z|x) \parallel p(z)] \}$$

KL regularizer
to make q closer to the prior and prevent $\sigma_\phi(x) \to 0$



$$p_\theta(x|z) = N(x; G_\theta(z), \sigma^2 I)$$

Kingma and Welling. Auto-encoding variational Bayes. ICLR 2014
Rezende et al. Stochastic backpropagation and approximate inference in deep generative model. ICML 2014

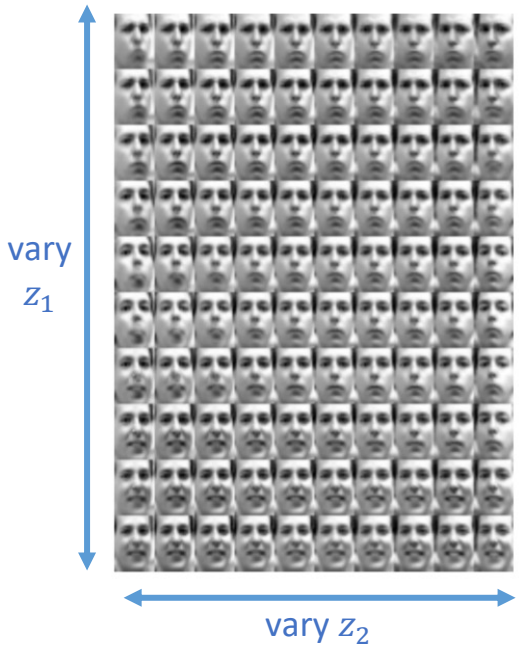
Generating data from the VAE

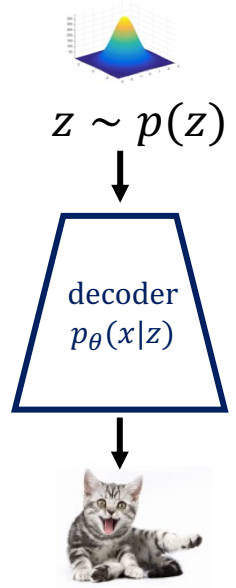
- Once trained, sample new images from the model:

$$z \sim p(z), \quad x \sim p(x|z)$$

Different z dimensions encode different info:

- z_1: facial expressions
- z_2: head pose





$$\mu_x = G_\theta(z)$$

$$p_\theta(x|z) = N(x; G_\theta(z), \sigma^2 I)$$

Kingma and Welling. Auto-encoding variational Bayes. ICLR 2014

Rezende et al. Stochastic backpropagation and approximate inference in deep generative model. ICML 2014

Variational Auto-Encoders

Practical implementation for solving (pseudo code):

$$\max_{\theta,\phi} E_{p_{data}(x)}[E_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL[q_\phi(z|x) \,||\, p(z)]]$$

$$= -\frac{1}{2\sigma^2}\|x - G_\theta(z)\|_2^2 + C$$

- Initialise θ, ϕ, learning rates γ, choose total iteration T for SGD
- For $t = 1, \dots, T$
  - $x_1, \dots, x_M \sim p_{data}(x)$

  # encoder: performing (approximate) posterior inference
  - Compute $\mu_\phi(x_m), \sigma_\phi(x_m)$ for $m = 1, \dots, M$
  - $z_m = \mu_\phi(x_m) + \sigma_\phi(x_m) \odot \epsilon_m,\ \epsilon_m \sim N(0, I)$ # reparam. trick

  # Decoder: reconstructing data
  - $\hat{x}_m = G_\theta(z_m)$ for $m = 1, \dots, M$

  # update neural network parameters
  - $L = \frac{1}{M}\sum_{m=1}^{M}[-\frac{1}{2\sigma^2}\|x_m - \hat{x}_m\|_2^2 - KL[q_\phi(z_m|x_m) \,||\, p(z_m)]]$
  - $(\theta, \phi) \leftarrow (\theta, \phi) + \gamma \nabla_{(\theta,\phi)} L$

can use the analytic KL form or estimated by Monte Carlo

A practical trick: KL annealing

Kingma and Welling. Auto-encoding variational Bayes. ICLR 2014
Rezende et al. Stochastic backpropagation and approximate inference in deep generative model. ICML 2014