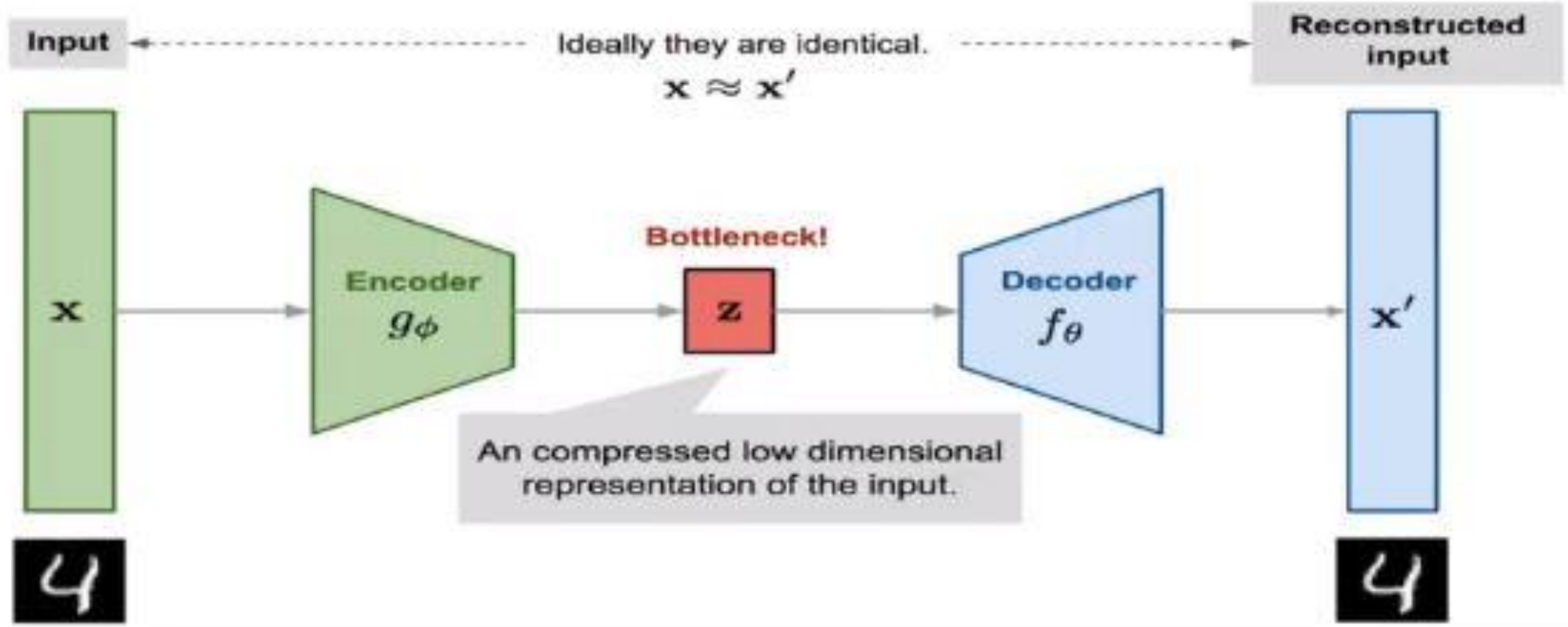
Variational Autoencoders

1. Review of Stacked Autoencoders
2. Basics of Probability
3. KL Divergence & its significance
4. Derivation of Loss function for Variational Autoencoders

Stacked Autoencoders

(Image Reconstruction)



⊛

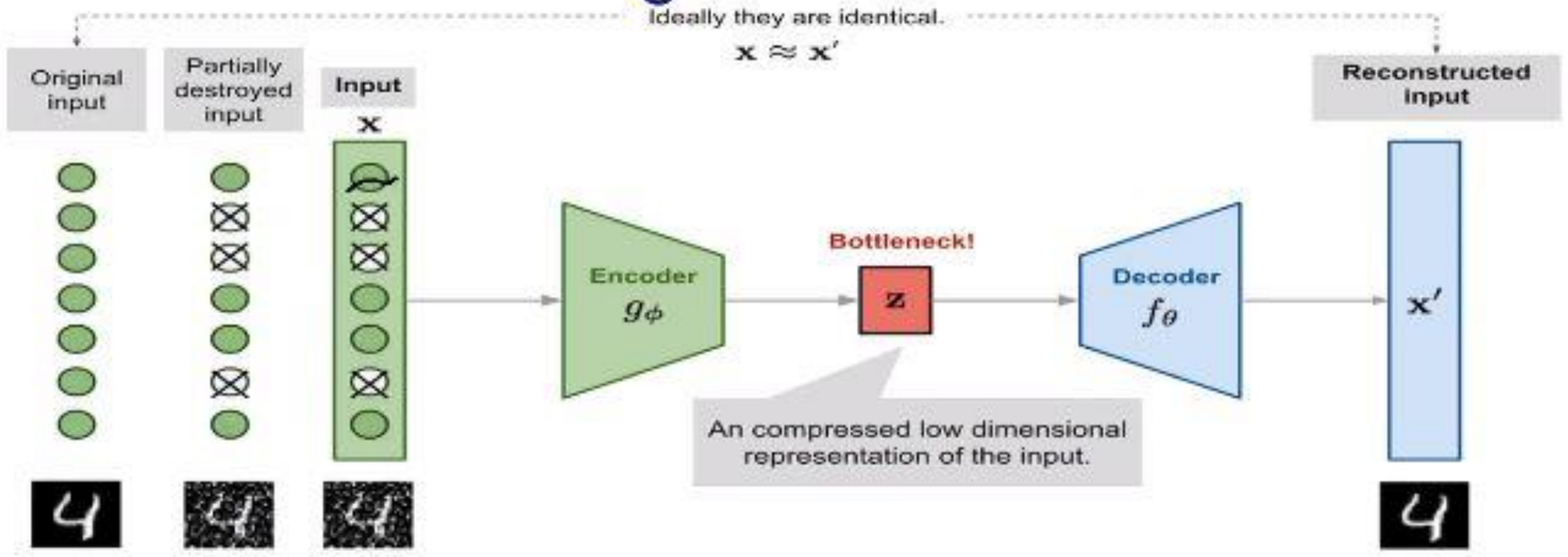
Cost function: $L(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} \left[x^{(i)} - f_\theta(g_\phi(x^{(i)})) \right]^2$

⊛ from lilianweng github accound

II

DL-16

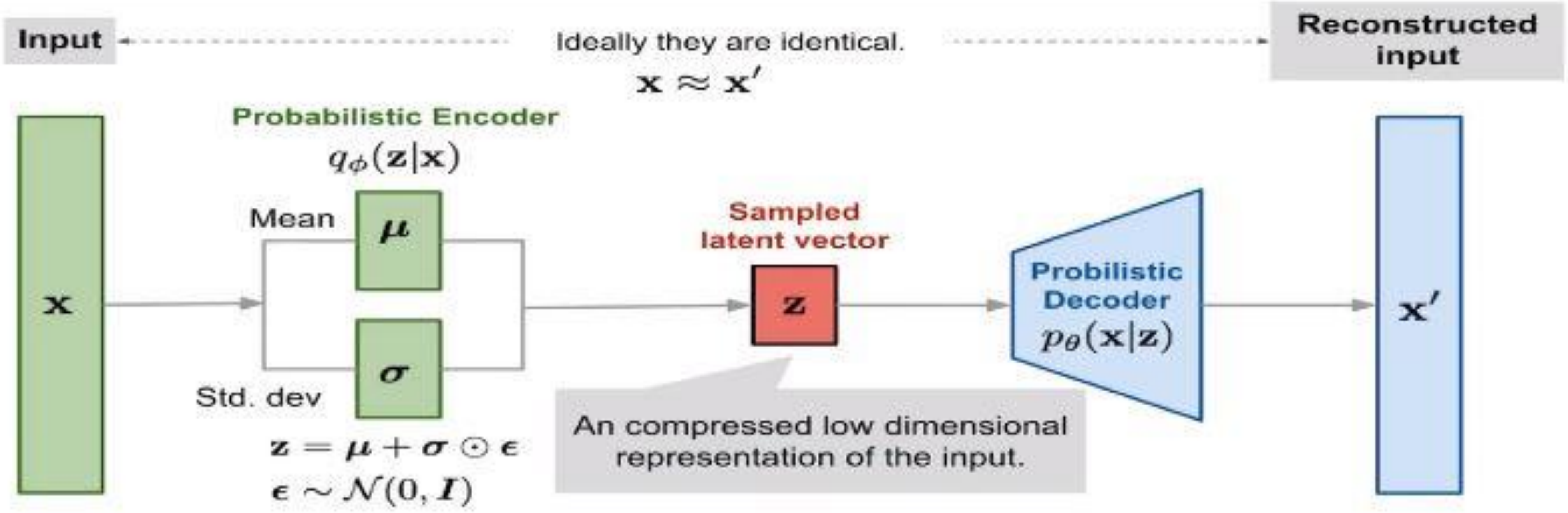
Denoising Autoencoder



$$\tilde{x}^{(i)} \sim \mathcal{N}\left(\tilde{x}^{(i)} \mid x^{(i)}\right)$$

Loss: $$L(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} \left[x^{(i)} - f_\theta\left(g_\phi\left(\tilde{x}^{(i)}\right)\right)\right]^2$$

Variational Autoencoders



$$\text{Loss} = L(\theta, \phi) = -E_{z \sim q_\phi(z|x)}\left[p_\theta(x|z)\right] + D_{KL}\left(q_\phi(z|x) \,\|\, p_\theta(z)\right)$$

Pre-requisite

VA

① Probability

- $p(x)$
- $p(x/y)$
- $E[\cdot]$
- KL Divergence

$p(x)$: defines the probability of random variable x

$p(x|y)$: defines as the probability of random variable x provided Y has happened
Also Called as Conditional probability

Liklihood ratio → prior probability

$$p(y|x) = \frac{p(x|y)\,P(y)}{P(x)} \rightarrow \text{Baye's Theorm}$$

posterior probability

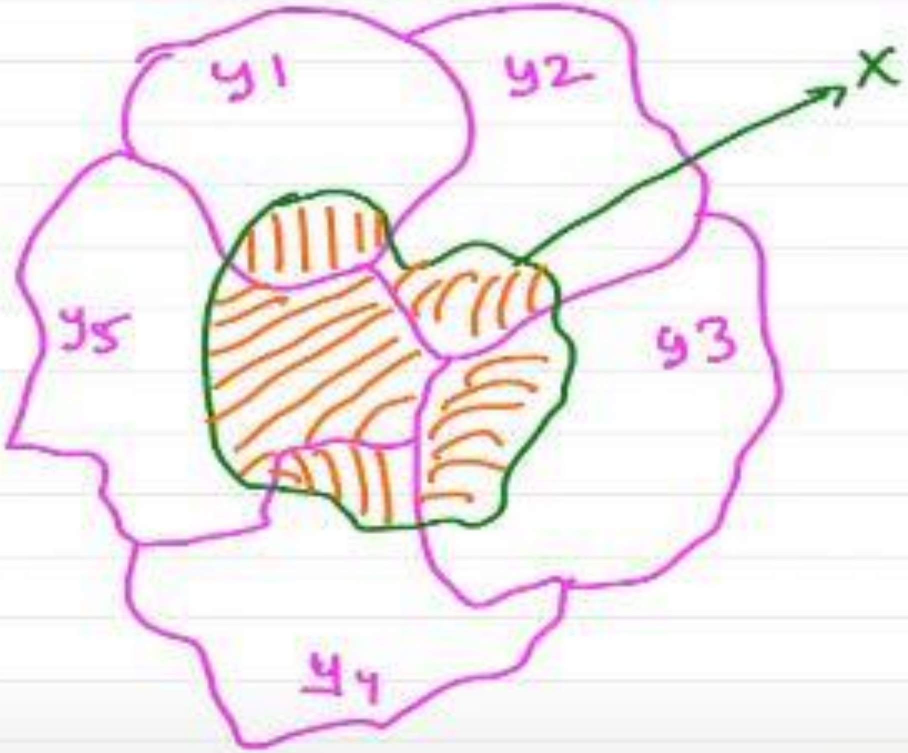
$$= \frac{P(x,y)}{P(x)} \rightarrow \text{joint distribution}$$

— ①

→ Theorem of Total probability.

Let $y_1, y_2, \dots y_N$ be a set of mutually exclusive events (i.e $y_i \cap y_j = 0$) & event X is the union of N mutually exclusive events, then

$$P(X) = \sum_{i=1}^{N} P(X/y_i)\,P(y_i) \quad \text{— ②}$$



$$P(x) = \sum_{i=1}^{4} P(x, y_i)$$

$$= \sum_{i=1}^{4} P(x/y_i)\, P(y_i)$$

$y_1, y_2 \ldots y_4$

x

So substituting (2) in (1) results in

$$\left\{ p(y/x) = \frac{P(x/y)\,P(y)}{\sum_{i=1}^{N} P(x/y_i)\,P(y_i)} \right\}$$

Expectation of random variable X i.e E(X)

Expected value of random variable is a weighted average of the possible values of X can take, each value being weighted according to the probability of that event defined as

$$E(x) = \sum_{i=1}^{K} x_i \, P(X = x_i)$$

Q When a die is tossed once. What is the probability of getting 3.

Ans Sample space = $\{1, 2, 3, 4, 5, 6\}$, $P(3) = \frac{1}{6}$

Q2 In tossing a fair die, what is the probability the 3 has occured conditioned on the toss being odd.

A Since, we are given that odd number has occured the sample space reduces from $\{1, 2, 3, 4, 5, 6\}$ to $\{1, 3, 5\}$. Hence the probability of 3 in this reduced sample space is $\frac{1}{3}$. It can be observed there is increase in the probability compared to the earlie case. Why?

Q3 Let X represent the outcome of a fair six sided die. What is the E(x)?

Ans $X = \{1, 2, 3, 4, 5, 6\}$ $\qquad P(x) = 1/6$

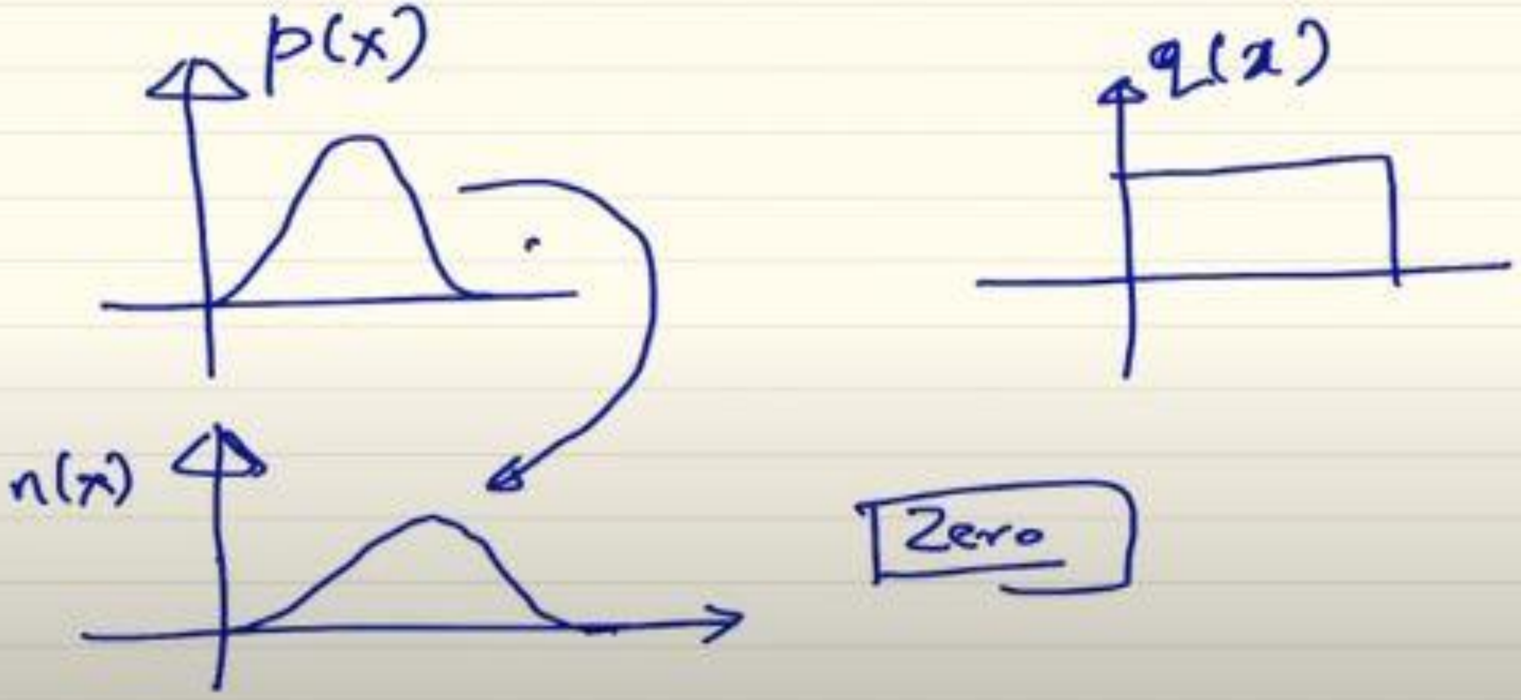
$$E(x) = \sum_{i=1}^{6} x\,P(x) = 1\times\frac{1}{6} + 2/6 + 3\times 1/6 + 4\times\frac{1}{6} + 5\times\frac{1}{6} + 6\times\frac{1}{6} = 3.5$$

K-L Divergence

#

Kullback-Leibler divergence (K-L) is measure of how one probability distribution is different from the second. For the discrete probability distribution P & Q, the K-L divergence between P & Q is defined as

K-L Divergence



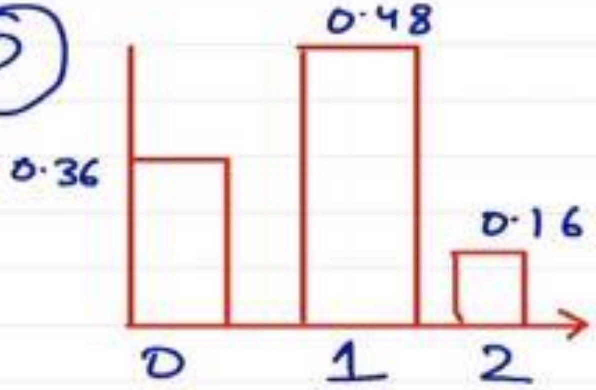
high value

$$\rightarrow D_{KL}(P \| Q) = \sum_x P(X=x) \log\left(\frac{P(X=x)}{Q(X=x)}\right)$$

$$= \sum_x P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

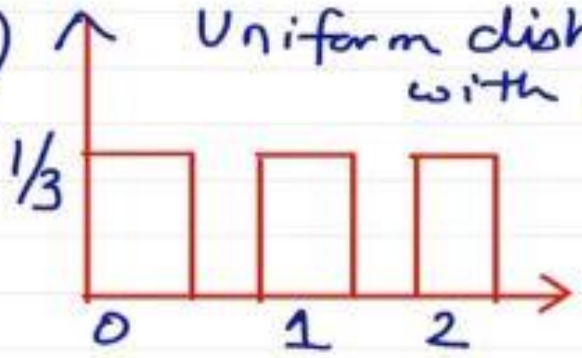
Example:

P



Q

Uniform distribution with $p = 1/3$



$$D_{KL}(Q \| P) = \frac{1}{3} \ln\left(\frac{0.333}{0.36}\right) + \frac{1}{3} \ln\left(\frac{0.333}{0.48}\right) + \frac{1}{3} \ln\left(\frac{0.333}{0.16}\right)$$

$$= 0.09637 \text{ nats}$$

Properties:

① $KL(P \| Q)$ or $KL(Q \| P) \geq 0$

② $KL(P \| Q) \neq KL(Q \| P)$ (Not Symmetric)

Suppose we have two multivariate normal distributions defined as

$$p(x) = N(x; \mu_1, \Sigma_1)$$
$$q(x) = N(x; \mu_2, \Sigma_2)$$

where μ_1 & μ_2 are the means & Σ_1, Σ_2 are the Covariance matrix

And the multivariate normal density is defined as

$$\mathcal{N}(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

if the two distributions have the same dimension k.

$$\boxed{D_{KL}(p(x)\,||\,q(x)) = \frac{1}{2}\left[\log \frac{|\Sigma_2|}{|\Sigma_1|} - d + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2-\mu_1)^T \Sigma_2^{-1}(\mu_2-\mu_1)\right]}$$

Prove ?

Proof: We know

$$KL(P(x)||Q(x)) = \sum_x P(x) \log\left(\frac{P(x)}{Q(x)}\right) \quad — (1)$$

We know

$$P(x) = \frac{1}{\sqrt{(2\pi)^K |\Sigma_1|}} \exp\left(\frac{-(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)}{2}\right)$$

$$\Rightarrow \log P(x) = -\frac{K}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1}(x-\mu_1) \quad (2)$$

Similarly

$$\log Q(x) = -\frac{K}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_2| - \frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1}(x-\mu_2) \quad (3)$$

Eq ① can be rewritten as

$$KL(P(x)\,||\,Q(x)) = \sum_{x} P(x) \log P(x) - \log(Q(x))$$

Substituting ② & ③ in ① results in

$$KL(P(x)\,||\,Q(x)) = \sum_{x} p(x) \Big\{ -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_1|$$

$$-\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1}(x-\mu_1) + \frac{k}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma_2|$$

$$+\frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1}(x-\mu_2) \Big\}$$

which on simplification results in

$$KL\left(P(x)\|Q(x)\right) =$$

$$\sum_x p(x)\left\{\frac{1}{2}\log\left[\frac{\Sigma_2}{\Sigma_1}\right] + \frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2) - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right\} \quad \text{—} \quad (1)$$

Now, let consider part by part

$$\sum_x p(x)\,\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) \equiv E_p\left[\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right]$$

$\rightarrow E(x^T A x) = E(tr(x^T A x))$ — (a)

$= E(tr(A x x^T))$ — (c)

$= tr(E(A x x^T))$ — (e)

let's rewrite again

$$\frac{1}{2} E_p \left[\overbrace{(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)}^{\text{scalar}} \right]$$

$$E_p \left[tr \left(\frac{1}{2} (x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1) \right) \right]$$

$$E_p \left[tr \left(\frac{1}{2} (x-\mu_1)(x-\mu_1)^T \Sigma_1^{-1} \right) \right] \rightarrow \text{(d)}$$

Trace & Expectation trick.

$\rightarrow$ if x is scalar then $E(x) \overset{(a)}{=} E(tr(x))$ since trace of x is scalar

$\rightarrow tr(AB) \overset{(b)}{=} tr(BA)$

$\rightarrow tr(ABC) \overset{(c)}{=} tr(BCA) \overset{(d)}{=} tr(CAB)$

$\rightarrow tr(ABC) \neq tr(ACB)$

$E(tr(x)) \overset{(e)}{=} tr(E(x))$

$$\rightarrow \operatorname{tr}\left[E_p\left(\frac{1}{2}(x-\mu_1)(x-\mu_1)^T\Sigma_1^{-1}\right)\right] \quad — \; e$$

$$\rightarrow \operatorname{tr}\left\{\underbrace{E_p\left[(x-\mu_1)(x-\mu_1)^T\right]}_{\text{Covariance matrix}}\frac{1}{2}\Sigma_1^{-1}\right\}$$

$$\Rightarrow \operatorname{tr}\left[\Sigma_1 \frac{1}{2}\Sigma_1^{-1}\right]$$

$$\operatorname{tr}\left[I_K\right] = K \quad — \; 2$$

Now consider the second part

$$\sum_x p(x)\left[\frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1} (x-\mu_2)\right]$$

$$\sum_x p(x)\left\{\frac{1}{2}\left[(x-\mu_1)+(\mu_1-\mu_2)\right]^T \Sigma_2^{-1} \left[(x-\mu_1)+(\mu_1-\mu_2)\right]\right\}$$

$$\sum_x p(x)\left[\frac{1}{2}(x-\mu_1)^T \Sigma_2^{-1} (x-\mu_1) + \frac{2}{2}(x-\mu_1)^T \Sigma_2^{-1} (\mu_1-\mu_2) + \frac{1}{2}(\mu_1-\mu_2)^T \Sigma_2^{-1} (\mu_1-\mu_2)\right]$$

$$\Rightarrow E_p\left[\frac{1}{2}(x-\mu_1)^T \Sigma_2^{-1} (x-\mu_1) + (x-\mu_1)^T \Sigma_2^{-1} (\mu_1-\mu_2) + \frac{1}{2}(\mu_1-\mu_2)^T \Sigma_2^{-1} (\mu_1-\mu_2)\right]$$

Now consider the second part

$$\sum_x p(x)\left[\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right]$$

$(A+B)^T\Sigma_2^{-1}[A+B]$

$(A^T+B^T)\Sigma_2^{-1}(A+B)$

$A^T\Sigma_2^{-1}A$

$$\sum_x p(x)\left\{\frac{1}{2}\left[(x-\mu_1)+(\mu_1-\mu_2)\right]^T\Sigma_2^{-1}\left[(x-\mu_1)+(\mu_1-\mu_2)\right]\right\}$$

$B^T\Sigma_2^{-1}B$

$$\sum_x p(x)\left[\frac{1}{2}(x-\mu_1)^T\Sigma_2^{-1}(x-\mu_1)+\frac{2}{2}(x-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2)+\frac{1}{2}(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2)\right]$$

$$\begin{cases} B^T\Sigma_2^{-1}A \\ A^T\Sigma_2^{-1}B \end{cases}$$

$$\Rightarrow E_p\left[\frac{1}{2}(x-\mu_1)^T\Sigma_2^{-1}(x-\mu_1)+(x-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2)+\frac{1}{2}(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2)\right]$$

Expanding we get

$$\underbrace{E_P\left\{\frac{1}{2}(x-\mu_1)^T\Sigma_2^{-1}(x-\mu_1)\right\}} + E_P\left[\underbrace{(x-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2)}\right]$$

$$+ E_P\left[\underbrace{(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2)}_{\text{Constant}}\right]$$

$$= \underbrace{tr\left\{\frac{\Sigma_2^{-1}\Sigma_1}{2}\right\}}_{\text{similar to earlier derivation}} + \underbrace{(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2)}_{E[\text{constant}] = \text{Constant}} + \underbrace{0}_{\text{proved on next slide}}$$

$\rightarrow$ β

$$\rightarrow E_p\left[(x-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2)\right]$$

$$\left[\left(E_p(x)-\mu_1\right)^T\Sigma_2^{-1}(\mu_1-\mu_2)\right]$$

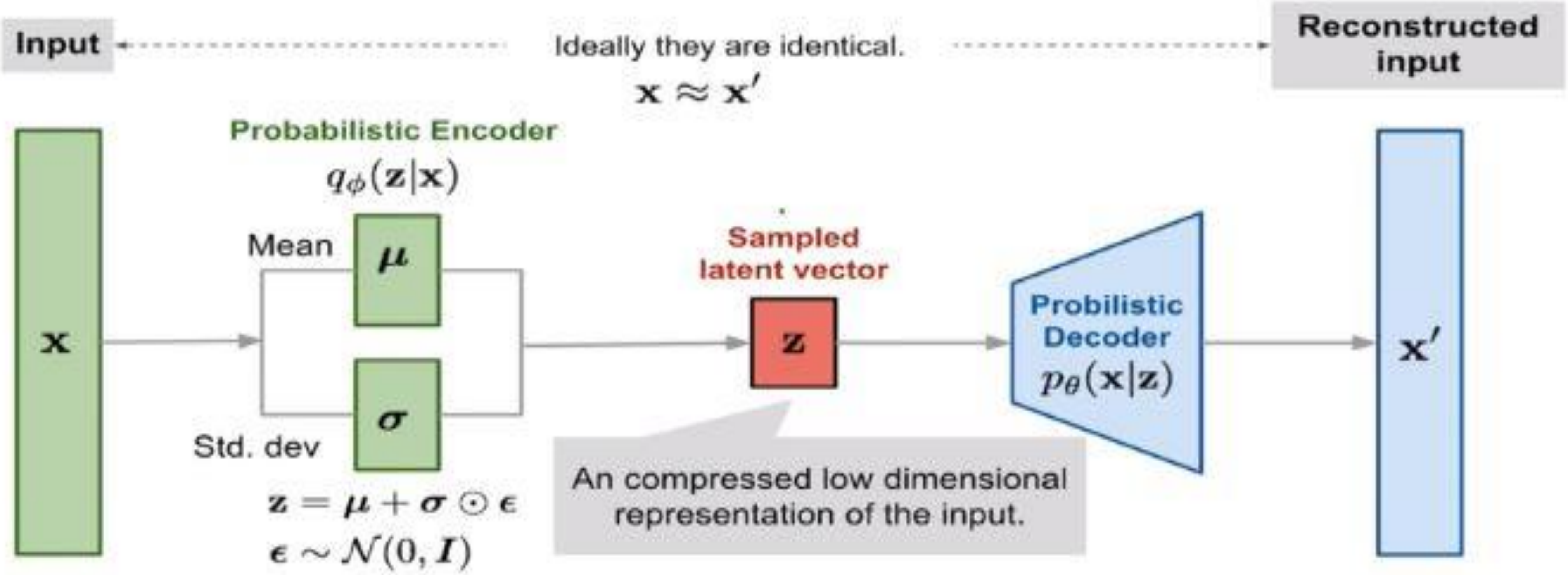
$$\equiv (\mu_1-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2) = 0 \quad \text{(Proved)}$$

So Substituting (Q), (R) in (P) we obtain

$$KL\left(P(x)\,||\,Q(x)\right) = \frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - K + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2)\right]$$

$\boxed{\text{proved}}$

Variational Autoencoder

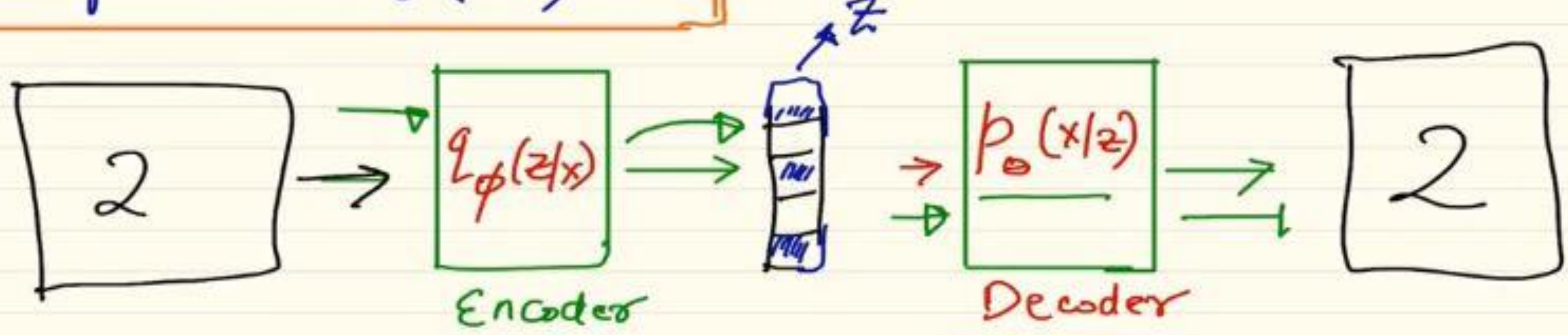


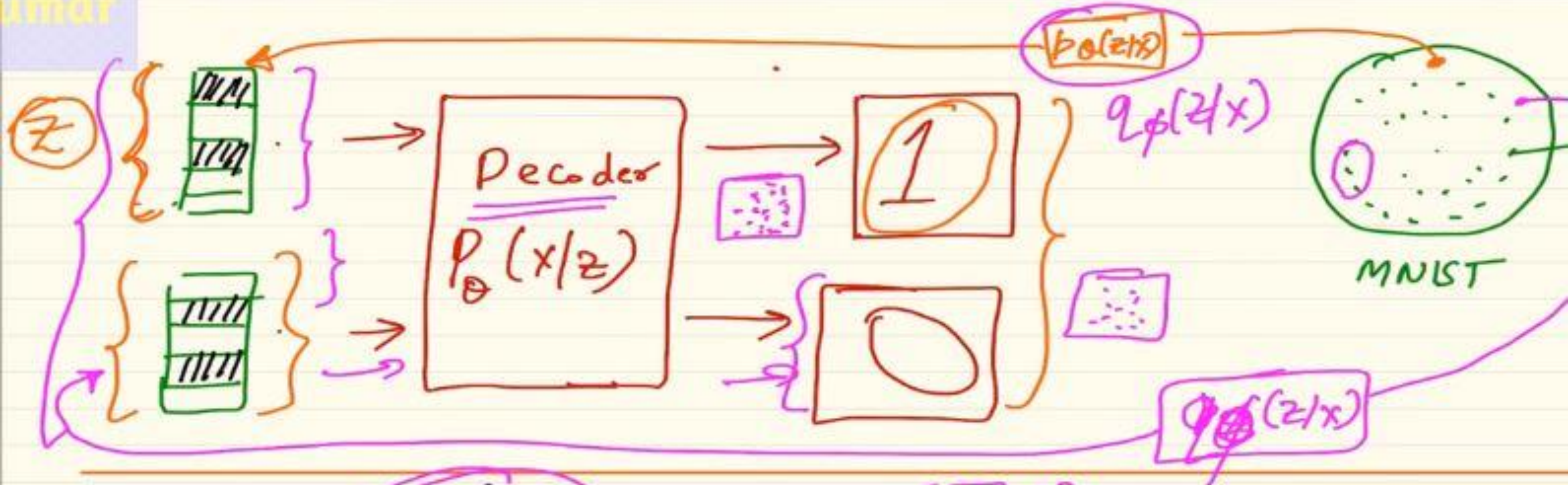
$$\text{Loss} = \underline{L(\theta, \phi)} = \underline{-E_{z \sim q_\phi(z|x)}\left[p_\theta(x|z)\right] + D_{KL}\left(q_\phi(z|x) \,\|\, p_\theta(z)\right)}$$

The goal of VAE

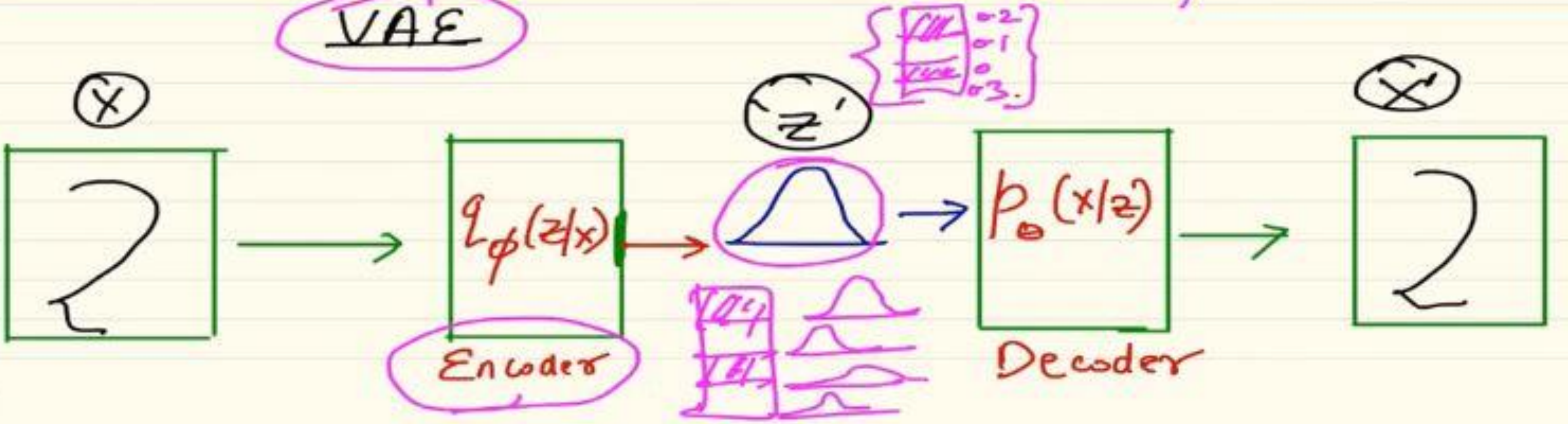
The goal of VAE is to find a distribution $q_\phi(z/x)$ of some latent variables which we can sample from $z \sim q_\phi(z/x)$ to generate new samples $x' \sim p_\theta(x|z)$

Typical Autoencoder




z
Decoder
$p_\theta(x|z)$
1
0
$p_\theta(z|x)$
$q_\phi(z|x)$
MNIST
$q_\phi(z|x)$

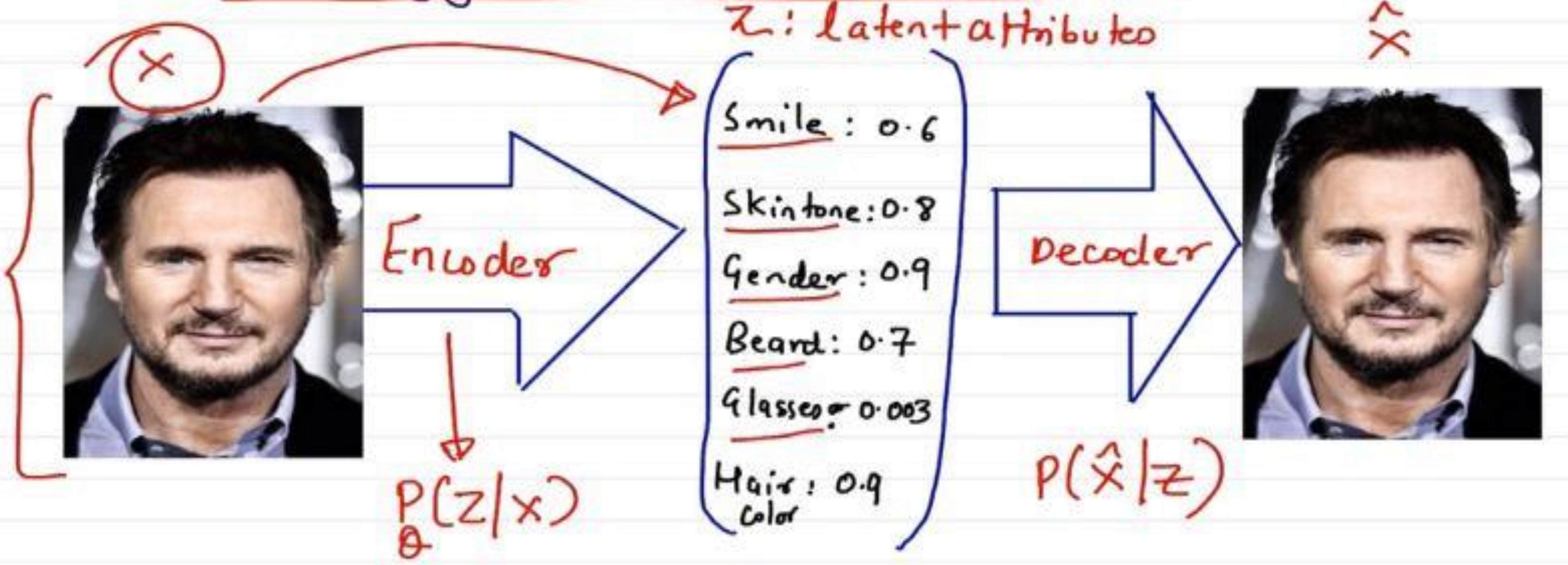
VAE


x
z
0.2
0.1
0
0.3
$q_\phi(z|x)$
Encoder
$p_\theta(x|z)$
Decoder
x'

~~Latent variables can be placed in 2 categories~~

Latent (2)

(1) Variables corresponding to a real feature of the object that have not been measured (may be technology is not available to do that)



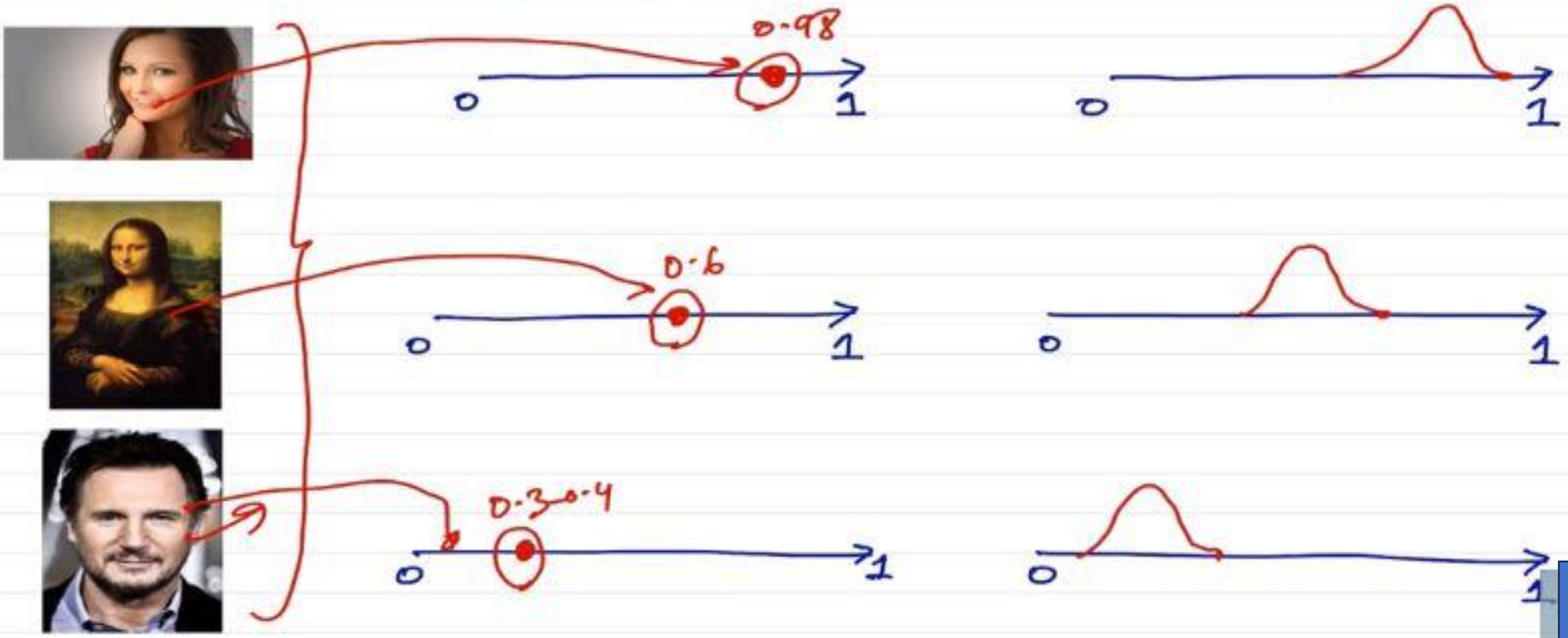
Using VE, we define latent attributes in probablistic terms.

AE

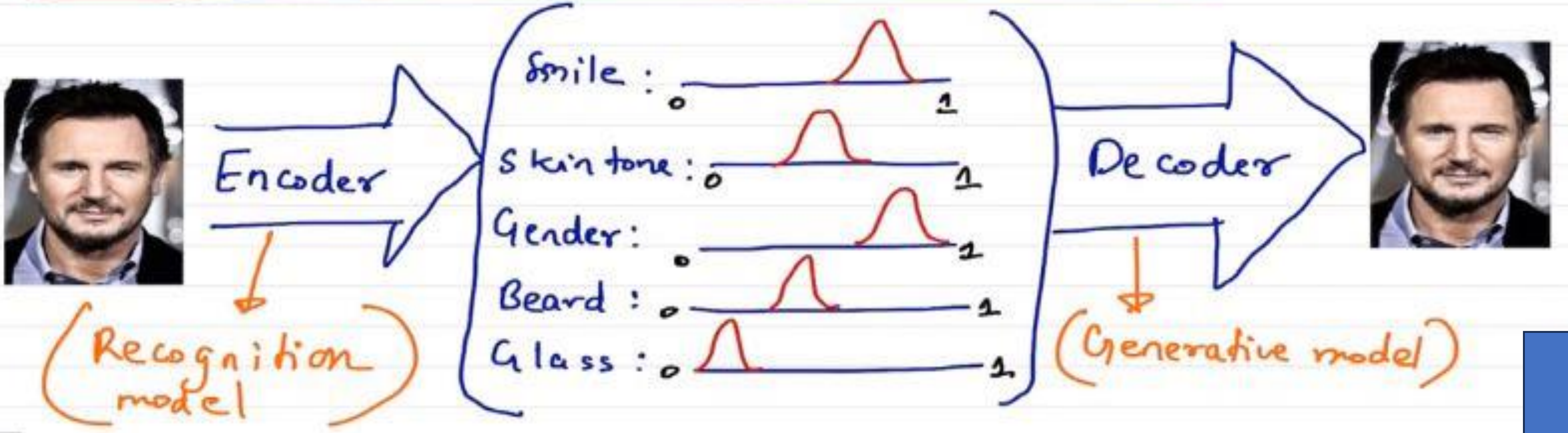
Smile (discrete)

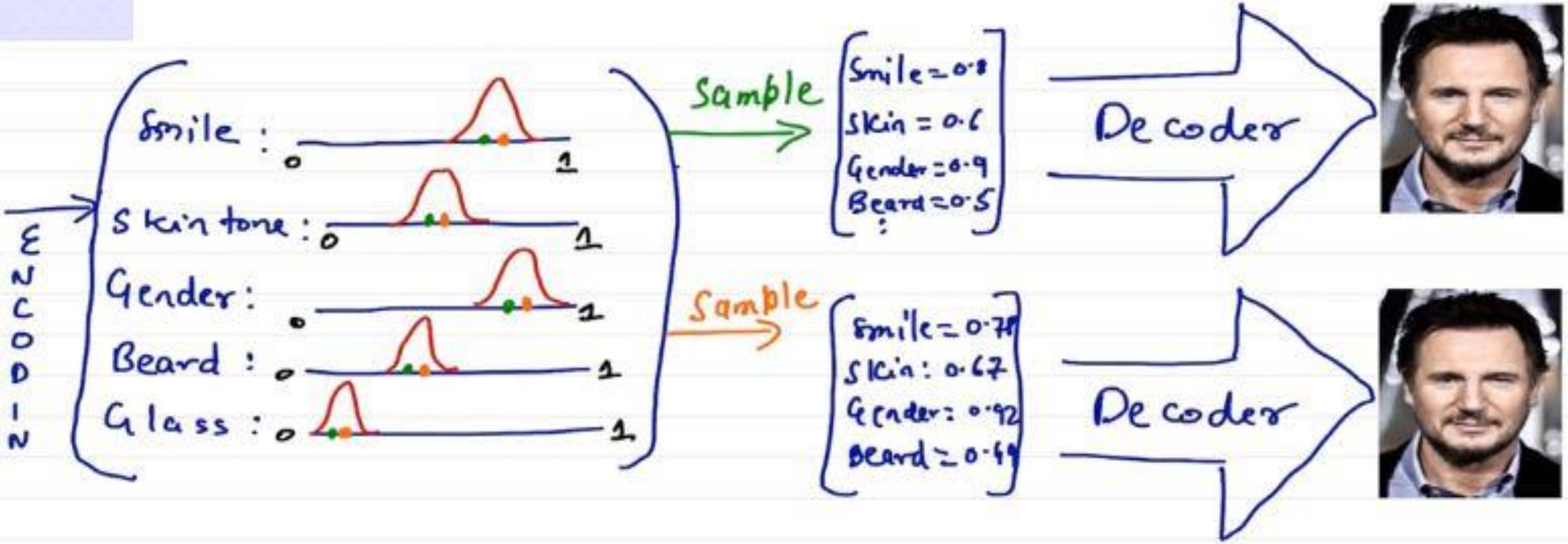
VE

Smile (Probablistic)



With this approach, we now represent each latent attribute for a given input as a probability distribution. When decoding, we will randomly sample from each latent state distribution to generate a vector as i/p for our decoder model.

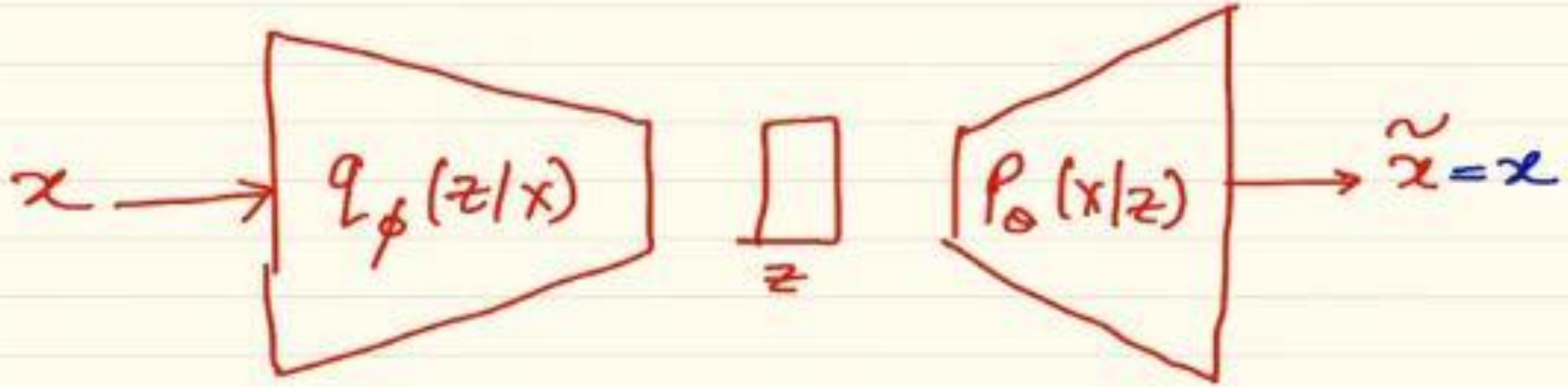


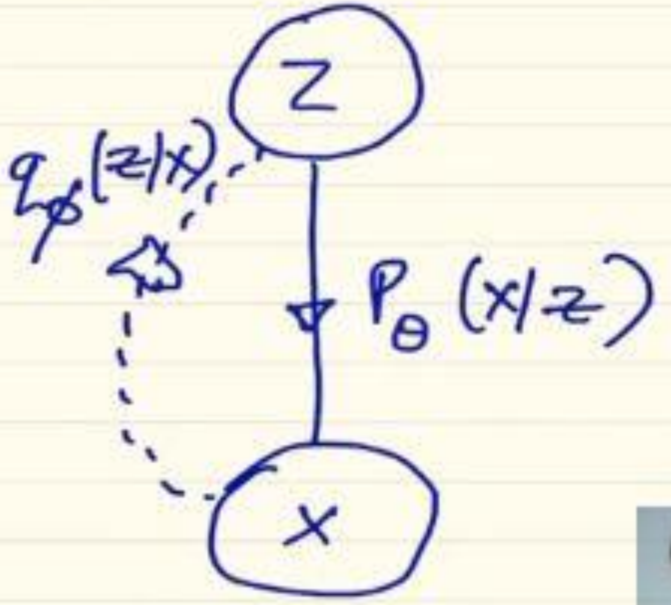


By constructing our encoder model to output range of possible values (a statistically distribution) from which we will randomly sample to feed into our decoder model. The the values which are nearby each other in latent space must correspond to similar reconstruct

Let's recall the Goal of VAE

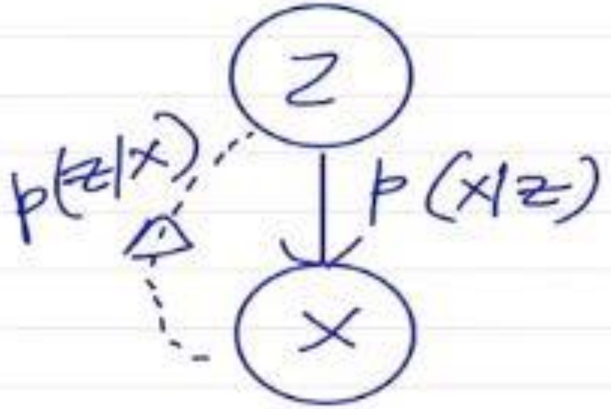
The goal of VAE is to find a distribution $q_\phi(z/x)$ of some latent variables, which we can sample from $z \sim q_\phi(z/x)$, to generate new samples x' from $p_\theta(x|z)$





The problem of Approximate Inference

Let x be a set of observed variables & let z be the set of latent variables with joint distribution $p(z,x)$. Then the inference problem is to compute the conditional distribution of latent variables given the observations i.e $p(z/x)$



→ We can write it as

$$p(z/x) = \frac{p(x|z)\,p(z)}{p(x)} \quad \text{— (A)}$$

Evaluating (A) is difficult because $p(x)$ cannot be solved

Reason :

$$p(x) = \int_z P(x|z)\, P(z)\, dz$$

this integral is not available in closed form or is intractable (i.e requires exponential time to compute) due to multiple integrals involved for latent variable vector z.

Alternative? The alternative is to approximate $P(z/x)$ by another distribution $q(z/x)$ which is defined in such a way that it has tractable solution. This is done using variational inference (VI). The main idea of

VI is to pose the inference problem as an optimization problem. How? By modelling $P(z/x)$ using $Q(z/x)$ where $Q(z/x)$ has a simple distribution such as Gaussian.

As discussed let us calculate KL b/w $P(z/x)$ & $Q(z/x)$

$$D_{KL}\left(Q_\phi(z/x) \,\|\, P_\theta(z/x)\right) = \sum_z Q_\phi(z/x) \log\left(\frac{Q_\phi(z/x)}{P_\theta(z/x)}\right)$$

$$= E_{z \sim Q_\phi(z/x)}\left[\log\left(\frac{Q_\phi(z/x)}{P_\theta(z/x)}\right)\right]$$

$$= E_{z \sim Q_\phi(z/x)}\left[\log\left(Q_\phi(z/x)\right) - \log\left(P_\theta(z/x)\right)\right] \quad \text{— B}$$

Substituting A in B results in

Here $\boxed{z = z \sim Q_\phi(z/x)}$

$$D_{KL}\left[Q_\phi(z/x) \| P_\theta(z/x)\right] = E_{z}\left[\log Q_\phi(z/x) - \log \frac{P_\theta(x/z)\, P_\theta(z)}{P_\theta(x)}\right]$$

$$= E_{z}\left[\log\left(Q_\phi(z/x)\right) - \log P_\theta(x/z) - \log P_\theta(z) + \log P_\theta(x)\right]$$

Since the expectation is over z & $P_\theta(x)$ does not involve z, it can be moved out

$$D_{KL}\left[Q_\phi(z/x) \| P_\theta(z/x)\right] - \log P_\theta(x) = E_{z}\left[\log Q_\phi(z/x) - \log P_\theta(x/z) - \log P_\theta(z)\right]$$

Rearranging the equations we obtain

$$\boxed{z : z \sim Q_\phi(z|x)}$$

$$\log P_\theta(x) - D_{KL}\left[Q_\phi(z|x) \,\|\, P_\theta(z|x)\right]$$

$$= E_z\left[\log(P_\theta(x|z))\right] - E_z\left[\log Q_\phi(z|x) - \log P_\theta(z)\right]$$

$$= \boxed{E_z\left[\log(P_\theta(x|z))\right] - D_{KL}\left[Q_\phi(z|x) \,\|\, P_\theta(z)\right]}$$

→ This is VAE objective function, where the first term represents the reconstruction liklihood & the second term ensures that our learned distributions Q is similar to the prior distribution P

& loss = − objective function

$$\boxed{L(\theta, \phi) = -E_{z \sim Q_{\phi}(z/x)}\left[\log\left(P_{\theta}(x/z)\right)\right] + D_{KL}\left[Q_{\phi}(z/x) \,||\, P_{\theta}(z)\right]}$$

Proved !!

Also $\underbrace{\log P_{\theta}(x)}_{\text{Reconstruction}} - \underbrace{D_{KL}\left[Q_{\phi}(z/x) \,||\, P_{\theta}(z|x)\right]}_{\text{Regularizer}} = -L(\theta, \phi)$

So our target is to find optimal θ, ϕ such that

$$\theta^*, \phi^* = \left\{ \underset{\theta, \phi}{\arg\min} \; L(\theta, \phi) \right\}$$

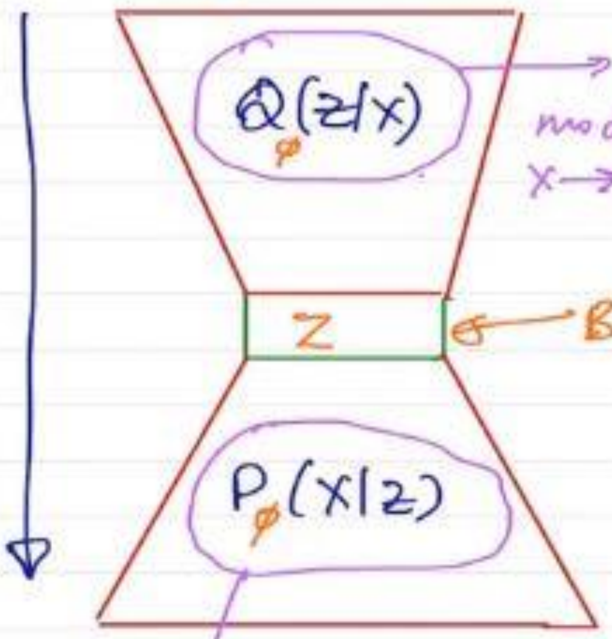
Recognition

generation

MORE INTIUTION ABOUT LOSS FUNCTION

$$L(\theta,\phi) = \underbrace{-E_{z\sim Q_\phi(z/x)}\left[\log\left(P_\theta(x|z)\right)\right]}_{\text{log-liklihood}} + \underbrace{D_{KL}\left[Q_\phi(z/x)\,||\,P_\theta(z)\right]}_{\text{Regularizer}}$$

Inputs



→ recognition model mapping $x \to z$

← Bottleneck

generative model mapping Z to X

log-liklihood

↓

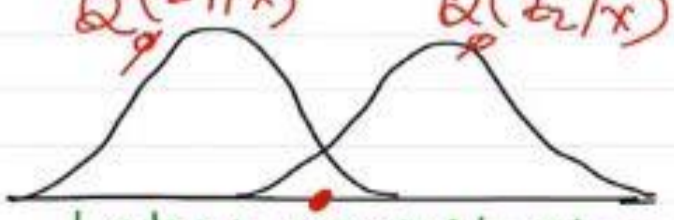
$$P_\theta(x|z) = N(\psi_\theta(z), \Sigma_\theta(z))$$

So when we take log of Gaussian

we get a square error b/w the data sample x & mean of the Gaussian distribution.

Regularizer

↓ Role



Latent variable space

KL divergence not allows pdf of latent variables not collapse with zero variance but penalize if deviates from $N(0,1) = P_\theta(z)$

$$\left\{ \underline{P_\theta(x|z)} \right. = \frac{1}{\sqrt{(2\pi)^k |\Sigma_\theta(z)|}} \exp\left[\frac{(x - \mu_\theta(z))^T \Sigma_\theta^{-1}(z) (x - \mu_\theta(z))}{2} \right]$$

$$\log P_\theta(x|z) \propto \underbrace{(x - \mu_\theta(z))^T \Sigma_\theta^{-1}(z) (x - \mu_\theta(z))}_{\text{Squared Reconstruction error}}$$

Let's discuss more about $\left\{ D_{KL}\left(Q_\phi(z|x) \,||\, P_\theta(z) \right) \right\}$

Here, $P(z)$ is the latent variable distribution.

→ The easiest choice is $N(0, \underline{1})$. We want $\cancel{Q(z|x)}$ $P_\phi(z)$ to be as close to $Q_\phi(z|x)$ so that we could sample it easily.

→ Having $P_\phi(z) = N(0, 1)$ adds another benefit. Let's say if we want $Q_\phi(z|x)$ to be Gaussian with parameter $\mu_\phi(x)$ & $\Sigma_\phi(x)$, the KL divergence has the closed form as derived earlier. ~~In that derivation,~~

$$D_{KL}\left[N\left(\mu_\phi(x), \Sigma_\phi(x)\right) \| N(0,1)\right] =$$

$$\longrightarrow \left\{\frac{1}{2}\left[tr\left(\Sigma_\phi(x)\right) + \mu_\phi(x)^T \mu_\phi(x) - K - \log|\Sigma_\phi(X)|\right]\right\}$$

→ Here K, is the dimension of the Gaussian.

→ $tr(\Sigma_\phi(x))$ is the trace function; which is sum of the diagonal matrix of $\Sigma_\phi(X)$

→ Moreover, determinant $|\Sigma_\phi(x)|$ of a diagonal matrix is product of it's diagonal so above equation can be simplified as

$$\rightarrow E_p\left[(x-\mu_1)^T \Sigma_2^{-1}(\mu_1-\mu_2)\right]$$

$$\left[(E_p(x)-\mu_1)^T \Sigma_2^{-1}(\mu_1-\mu_2)\right]$$

$$\equiv (\mu_1-\mu_1)^T \Sigma_2^{-1}(\mu_1-\mu_2) = 0 \quad (\text{Proved})$$

$\Sigma_1 = \Sigma_\phi(x)$
$\mu_1 = \mu_\phi(x)$
$\Sigma_2 = 1$
$\mu_2 = 0$

So Substituting Ⓠ, Ⓟ in Ⓟ we obtain

VAE

$$\left\{ KL\left(\underset{\substack{N \\ \mu_1\, \Sigma_1}}{P(x)} \,\|\, \underset{\substack{\Phi \\ \mu_2\, \Sigma_2}}{Q(x)}\right) = \frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - K + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_1-\mu_2)^T \Sigma_2^{-1}(\mu_1-\mu_2)\right] \right\}$$

$\boxed{\text{proved}}$

$$D_{KL}\left[N\left(\mu_\phi(X), \Sigma_\phi(x)\right) \,||\, N(0,1)\right] =$$

$$\frac{1}{2}\left(\sum_k \Sigma_\phi(x) + \sum_k \mu_\phi^2(x) - \sum_k 1 - \log \prod_k \Sigma_\phi(x)\right)$$

$$= \frac{1}{2}\left(\sum_k \Sigma_\phi(x) + \sum_k \mu_\phi^2(x) - \sum_k 1 - \sum_k \log\left(\Sigma_\phi(x)\right)\right)$$

$$= \frac{1}{2} \sum_k \left(\Sigma_\phi(x) + \mu_\phi^2(x) - 1 - \log\left(\Sigma_\phi(x)\right)\right)$$

In practice it is better to model $\Sigma_\phi(x)$ as $\log(\Sigma_\phi(x))$

as it is more numerically stable to take exponent compared to computing log. Hence substituting $\exp(\Sigma(x))$ instead of $\Sigma(x)$ we obtain,

$$D_{KL}\left[N(\mu_\phi(x), \Sigma_\phi(x)) \,||\, N(0,1)\right] = \frac{1}{2}\sum_k \left[\exp(\Sigma_\phi(x)) + \mu_\phi^2(x) - 1 - \Sigma_\phi(x)\right]$$

So the final loss function of VAE becomes

$$L(\theta,\phi) = -E_{z}\left[\log\left(P_\theta(x|z)\right)\right] + \frac{1}{2}\sum_k \left[\exp\left(\Sigma_\phi(x)\right) + \mu_\phi^2(x) - 1 - \Sigma_\phi(x)\right]$$

$z \sim Q_\phi(z|x)$