AUTO-ENCODER → VAE → β-VAE

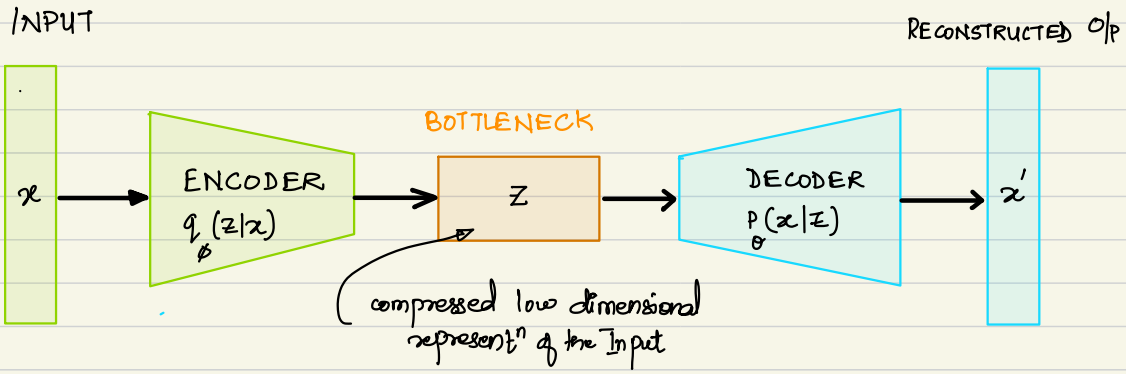
Contents:

1. Auto Encoders
2. De-noising Auto Encoders
3. Sparse Auto-Encoders
4. Contractive Auto-Encoders
5. Variational Auto Encoders
   a. Loss function : Evidence Lower Bound
   b. Reparametrization Trick
6. Beta - VAE
7. VQ-VAE and VQ-VAE 2
8. TD-VAE
9. References

① **AUTO ENCODER** : is a neural network designed to learn an identity function in an Unsupervised way to reconstruct the original input while compressing the data in the process so as to discover a more efficient & compressed representation. The original idea dates back to 1980s and was promoted by Geoffrey Hinton & Salakhutdinov in 2006.

It consists of 2 networks:

1. Encoder : which translates the original high dimension input into Latent low dimensional code.
2. Decoder : receives the CODE as input & recovers the data from the code with inverse transformⁿ.



The encoder essentially accomplishes Dimensionality Reduction, like PCA or Matrix Factorization. In addition, the auto encoder is explicitly optimized for data reconstruction from the code. A good intermediate representation not only can capture latent variables, but also benefits a full decompression process.

The model contains an encoder fn $q_\phi(z|x)$ parametrized by ϕ & a decoder function $p_\theta(x|z)$ parametrized by θ. The low-dimensional code learned from the input $\underline{x}$ in the bottleneck layer is $\underline{z} = q_\phi(z|x)$ and the reconstructed input is $x' = p_\theta(q_\phi(x))$

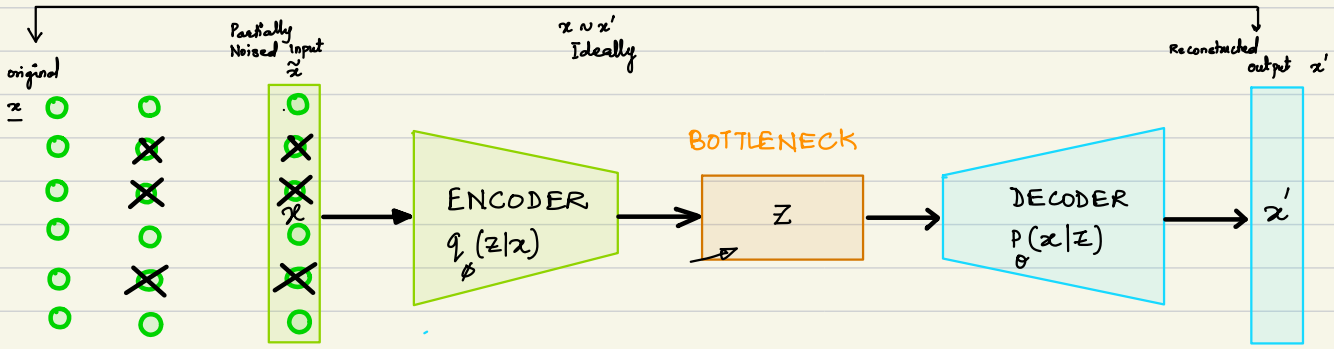
The parameters (ϕ, θ) are LEARNED together to o/p a reconstructed data sample, same as the original input, or in other words to learn the identity function. There are various METRICS to quantify the difference between the two vectors, such as CROSS ENTROPY when the activation function is SIGMOID or simple MEAN SQUARE ERROR loss.

$$\text{Loss}_{AE}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n}\left(x^{(i)} - q_\phi\left(p_\theta(x^{(i)})\right)\right)^2$$

(2.) **DENOISING AUTO-ENCODER** : When q_ϕ & p_θ are Deep Neural Networks, with more parameters than the data points, then autoencoders, face the risk of OVER FITTING. To avoid overfitting and Improve Robustness, Denoising Auto Encoders, 2008, proposed a modification. The input is partially corrupted either by adding NOISE or by masking some of the values by a STOCHASTIC process. $\tilde{x} \sim M_D(\tilde{x}|x)$. The model is trained to recover the original input x (not the corrupt one $\tilde{x}$)

$\tilde{x} \sim M_D(\tilde{x}|x)$ = defines the mapping from true data samples to corrupt.

$$\underset{DAE}{Loss}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n}\left(x^{(i)} - p_\theta\left(q_\phi(\tilde{x}^{(i)})\right)\right)^2$$



This design is motivated by the fact that humans can easily recognize an object or a scene even if the view is partially occluded or corrupted. To repair the partially destroyed input, the denoising AE has to discover & capture the relationship between dimensions of the I/p in order to infer to the missing pieces. The noise is controlled by STOCHSTIC mapping $M_D(\tilde{x}|x)$ and it is not specific to a particular type of a corruption process (masking noise, Gaussian noise, salt & pepper noise etc). Naturally the corruption process can be equipped with PRIOR knowledge.

In the original experiment of DENOISING, a fixed proportion of the input dimensions are selected at random and their values are forced to ZERO. This is similar to dropout.

(3.) **SPARSE AUTOENCODERS** : apply a "SPARSE" constraint on the hidden unit activation to avoid OVERFITTING & improve ROBUSTNESS. It forces the model to only have a small no. of hidden units being active at the same time. A neuron is activated when the value is close to 1 & is inactive when the value is close to zero.

Let us assume that there are s_l neurons in the l^{th}-th hidden layer & the activation fn for the j^{th} neuron in this layer is labelled as $a_j^{(l)}(\cdot)$, $j = 1, 2, \ldots, s_l$. The fraction of the activation of this neuron $\hat{s}_j$ is expected to be a small number s. known as the SPARCITY PARAMETER (normally $s = 0.05$)

$$\hat{s}_j^{(l)} = \frac{1}{n}\sum_{i=1}^{n}\left[a_j^{(l)}(x^{(i)})\right] \approx s$$

This constraint is achieved by adding a penalty term into the loss function. The KL-divergence term measures the difference the two Bernoulli distributions, one with mean s & the other with the mean $\hat{s}_j^{(l)}$. The hyperparameter 'β' controls how strong the penalty we want to impose on the sparsity loss.

$$L_{SAE}(\theta) = L(\theta) + \beta\sum_{l=1}^{L}\sum_{j=1}^{s_l} D_{KL}(s \,\|\, \hat{s}_j^{(l)})$$

$$= L(\theta) + \beta\sum_{l=1}^{L}\sum_{j=1}^{s_l} s\log\frac{s}{\hat{s}_j^{(l)}} + (1-s)\log\left[\frac{(1-s)}{1-\hat{s}_j^{(l)}}\right]$$

K- SPARSE ENCODER:- The sparsity is enforced by only keeping the top k-highest activations in the bottleneck layer with Linear activation function. First we run a feed forward through the encoder to get the compressed CODE: $z = q(x)$. Sort the values in the code vector z. Only the k- largest are kept & all the other neurons are set to 0. This can be done by a ReLU with an adjustable threshold. This operation leads to a SPARSIFIED CODE: $z' = \text{sparsify}(z)$. Compute the output and the LOSS from the sparsified(z). Note that the BACKPROPOGATION only goes through the top 'k' activated hidden units!

4. **CONTRACTIVE AUTOENCODER.** : encourages the learned representation to stay in a contractive space for better robustness. It adds a term in the LOSS FUNCTION to penalize the representation being too sensitive to the input, and thus improve the robustness to small perturbations around the training data points. The sensitivity is measured by the FROBENIUS norm of the JACOBIAN matrix of the encoder activations wrt the inputs:

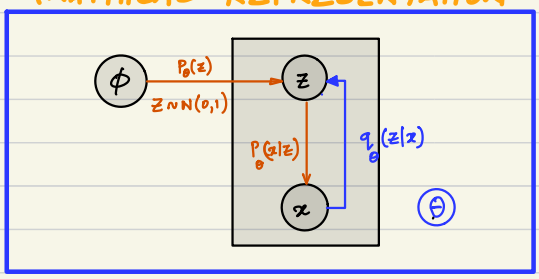
$$\| J_f(x) \|_F^2 = \sum_{ij} \left(\frac{\partial h_j(x)}{\partial x_i} \right)^2$$

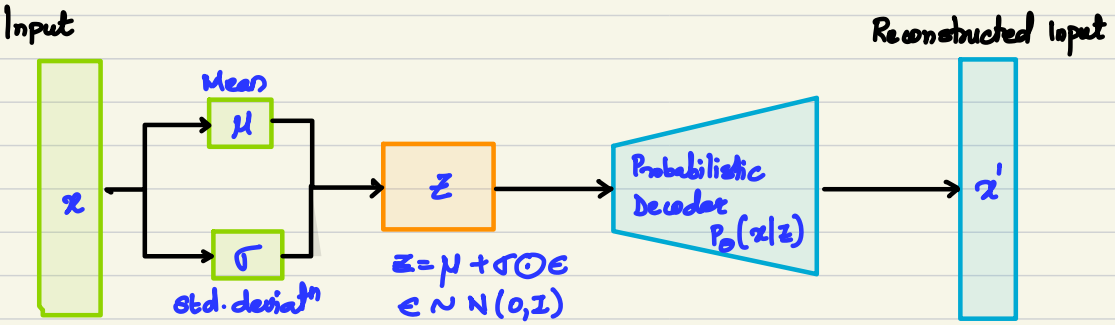
where h_j is one unit output in the compressed code $z = f(x)$. This penalty term is the sum of all the squares of all partial derivatives of the learned encodings wrt the input dimensions.

5. **VAE: VARIATIONAL AUTOENCODERS.** (KINGMA & WELLING, 2014) is deeply rooted in the methods of Variational Bayesian & Graphical models. Instead of mapping the input into a FIXED vector, we want to map it into a distribution $P_\phi(z|x)$ parametrized by θ.

- Prior $q_\phi(z)$
- Likelihood $P_\theta(x|z)$
- Posterior : $q_\phi(z|x)$

GRAPHICAL REPRESENTATION





$$\text{Loss}_{\text{VAE}}(\theta, \phi) = - \mathop{\mathbb{E}}_{z \sim q_\phi(z|x)} \left[P_\theta(x|z) \right] + D_{KL} \left(q_\phi(z|x) \,\|\, P_\theta(x) \right)$$

(6) **BETA-VAE:** If each variable in the inferred latent representation 'z' is only sensitive to one single generative factor & relative invariant to other factors, we will say this representation is DISENTANGLED or FACTORIZED. One benefit that often comes with disentangled representation is good interpretability.

β-VAE (HIGGINS, 2017) is a modified of Variational AUTOENCODERS with a special emphasis to discover disentangled latent factors. As in VAE, we want to maximize the probability of generating real data, while keeping the distance between the real and the estimated posterior distributions small.

$$\max_{\phi,\theta} \mathbb{E}_{x\sim D}\left[\mathbb{E}_{z\sim q_\phi(z|x)} \log P_\theta(x|z)\right]$$

subject to $D_{KL}\left(q_\phi(z|x) \,||\, P_\theta(z)\right) < \delta$ where δ is a small constant.

PART-II: BASICS of KULLBACK-LEIBLER DIVERGENCE

1. $$D_{KL}(P||Q) = \sum_x P(X=x).\log\left[\frac{P(X=x)}{Q(X=x)}\right] = \sum_x P(x).\log\left[\frac{P(x)}{Q(x)}\right]$$

2. NOTE: $KL(P||Q) \neq KL(Q||P)$ ie KL divergence is NOT symmetric

3. Also, KL divergence is always positive.

4. Let $P(x)$ & $q(x)$ be MULTI-VARIATE Gaussians:

$P(x) \sim N(x: \mu_1, \Sigma_1)$ μ_1 & μ_2 are means

$q(x) \sim N(x: \mu_2, \Sigma_2)$ Σ_1 & Σ_2 are covariances.

By defn of a multivariate Gaussian the p.d.f is given by

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

5. let $P(x)$ and $q(x)$ have the same dimensions 'k' then prove that

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

gmp

PROOF:

$$KL(P(X)\,||\,Q(X)) = \sum_x P(x).\log\left[\frac{P(x)}{Q(x)}\right] \quad \text{---- (1)}$$

where,

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

$$\Rightarrow \log p(x) = -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(x-\mu_1)^T\cdot\Sigma_1^{-1}(x-\mu_1) \quad \text{——— (2)}$$

& similarly $\log q(x) = -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_2| - \frac{1}{2}(x-\mu_2)^T\cdot\Sigma_2^{-1}(x-\mu_2)$ ——— (3)

rearranging eq^n (1) as $KL(P(x)\,||\,q(x)) = \sum_x p(x)\cdot[\log p(x) - \log q(x)]$ ——— (1)

substituting (2) & (3) in (1) $\Rightarrow$ we get

$$KL(P(x)\,||\,q(x)) = \sum_x P(x)\left\{ -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(x-\mu_1)^T\cdot\Sigma_1^{-1}(x-\mu_1) + \frac{k}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma_2| + \frac{1}{2}(x-\mu_2)^T\cdot\Sigma_2^{-1}(x-\mu_2)\right\}$$

$$KL(P(x)\,||\,Q(x)) = \sum_x p(x)\left\{ \frac{1}{2}\log\left(\frac{\Sigma_2}{\Sigma_1}\right) + \frac{1}{2}(x-\mu_2)^T\cdot\Sigma_2^{-1}(x-\mu_2) - \frac{1}{2}(x-\mu_1)^T\cdot\Sigma_1^{-1}(x-\mu_1)\right\} \quad \text{——— (4)}$$

(B) : $+\frac{1}{2}\log\left(\frac{\Sigma_2}{\Sigma_1}\right) + \frac{1}{2}(x-\mu_2)^T\cdot\Sigma_2^{-1}(x-\mu_2)$

(A) : $-\frac{1}{2}(x-\mu_1)^T\cdot\Sigma_1^{-1}(x-\mu_1)$

Consider the term (A)

$$\underbrace{\sum_x p(x)\cdot\frac{1}{2}(x-\mu_1)^T\cdot\Sigma_1^{-1}(x-\mu_1)}_{(A)} = \mathbb{E}_{p(x)}\left[\frac{1}{2}(x-\mu_1)^T\cdot\Sigma_1^{-1}(x-\mu_1)\right]$$

Standard formulae:

1. if X is a scalar, the $\mathbb{E}[X] = \mathbb{E}[\text{trace}(X)]$

   $\text{trace}(AB) = \text{trace}(BA)$

   $\text{trace}(ABC) = \text{trace}(BCA) = \text{trace}(CAB)$

   $\text{trace}(ABC) \neq \text{trace}(ACB)$

   $\mathbb{E}[\text{trace}(X)] = \text{trace}\,\mathbb{E}[X]$

2. Now consider, $\mathbb{E}[X^TAX] = \mathbb{E}[\text{trace}(X^TAX)]$
   $= \mathbb{E}[\text{trace}(AXX^T)]$
   $= \text{trace}\,\mathbb{E}[AXX^T]$

$\therefore$ the term (A)

$$\underbrace{\sum_x p(x)\cdot\frac{1}{2}(x-\mu_1)^T\cdot\Sigma_1^{-1}(x-\mu_1)}_{(A)} = \frac{1}{2}\mathbb{E}\left[\overbrace{(x-\mu_1)^T}^{X^T}\cdot\overbrace{\Sigma_1^{-1}}^{A}\overbrace{(x-\mu_1)}^{X}\right]$$

$$= \mathbb{E}_{p(x)}\left[\text{tr}\left(\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)\right]$$

$$= \mathbb{E}_{p(x)}\left[\text{tr}\left(\frac{1}{2}(x-\mu_1)(x-\mu_1)^T\Sigma_1^{-1}\right)\right] \quad \text{——— } \circledast$$

$$= \text{trace}\,\mathbb{E}_{p(x)}\left[\frac{1}{2}(x-\mu_1)(x-\mu_1)^T\cdot\Sigma_1^{-1}\right]$$

$$= \text{trace}\left\{\mathbb{E}[(x-\mu_1)(x-\mu_1)^T]\left(\frac{1}{2}\cdot\Sigma_1^{-1}\right)\right\}$$

being const wrt x can be taken outside the $\mathbb{E}_{p(x)}$

By defn $= \Sigma_1$

$$\therefore (A) = \text{trace}\{\Sigma_1\cdot\tfrac{1}{2}\Sigma_1^{-1}\} = \text{trace}\{I_K\} = K$$

Now consider the term (B),

$$\sum_x P(x) \cdot \frac{1}{2} (x-\mu_2)^T \cdot \Sigma_2^{-1} (x-\mu_2) =$$

Add and subtract μ_1 in both the brackets

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

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

$$= \mathbb{E}_{P(x)} \left[\frac{1}{2} (x-\mu_1)^T \Sigma_2^{-1} (x-\mu_1) + (x-\mu_1)^T \cdot \Sigma_2^{-1} (\mu_1-\mu_2) + \frac{1}{2} (\mu_1-\mu_2)^T \cdot \Sigma_2^{-1} (\mu_1-\mu_2) \right]$$

(B) $= \mathbb{E}_{P(x)} \left[\frac{1}{2} (x-\mu_1)^T \cdot \Sigma_2^{-1} (x-\mu_1) \right] + \mathbb{E}_{P(x)} \left[(x-\mu_1)^T \Sigma_2^{-1} (\mu_1-\mu_2) \right] + \mathbb{E}_{P(x)} \left[\frac{1}{2} (\mu_1-\mu_2)^T \Sigma_2^{-1} (\mu_1-\mu_2) \right]$

(I) (II) (III)

(I) → comparing with (A) above

(II) → ZERO

$$= \text{trace} \left\{ \frac{\Sigma_2^{-1} \Sigma_1}{2} \right\} + 0 + \mathbb{E}_{P(x)} \left[\frac{1}{2} (\mu_1-\mu_2)^T \cdot \Sigma_2^{-1} (\mu_1-\mu_2) \right]$$

(II) $\Rightarrow \mathbb{E}_{P(x)} \left[(x-\mu_1)^T \cdot \Sigma_2^{-1} (\mu_1-\mu_2) \right] = \mathbb{E}_{P(x)} \left[(x-\mu_1)^T \right] \cdot \Sigma_2^{-1} (\mu_1-\mu_2)$

$$= \left\{ \mathbb{E}_{P(x)} [x] - \mu_1 \right\}^T \cdot \Sigma_2^{-1} (\mu_1-\mu_2)$$

($\mathbb{E}_{P(x)}[x] \to \mu_1$)

$$= (\mu_1-\mu_1)^T \cdot \Sigma_2^{-1} (\mu_1-\mu_2)$$

$$= 0 = \text{ZERO}$$

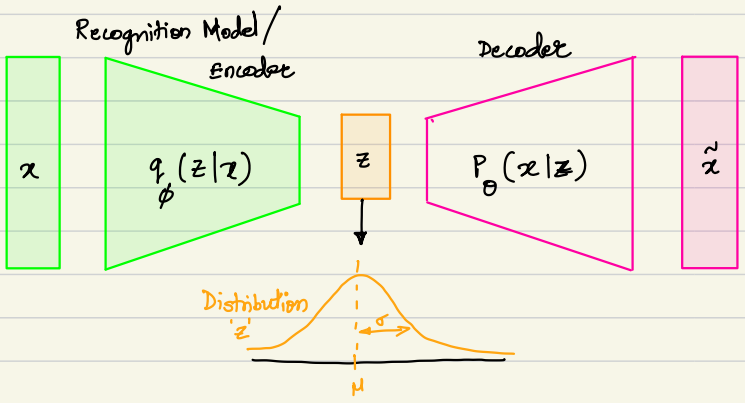
$$\therefore (B) = \text{trace} \left[\frac{\Sigma_2^{-1} \Sigma_1}{2} \right] + \frac{1}{2} \left[(\mu_1-\mu_2)^{-1} \cdot \Sigma_2^{-1} (\mu_1-\mu_2) \right]$$

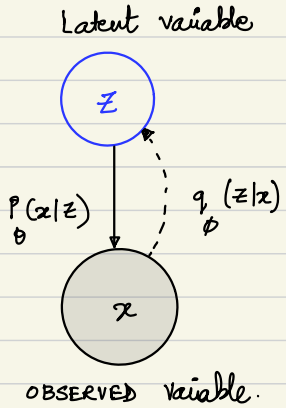
substituting (A) & (B) in Eq^n (4) we get

$$KL(P(x) \,||\, Q(x)) = \cancel{\sum_x p(x)}^{=1} \cdot \frac{1}{2} \log \left(\frac{\Sigma_2}{\Sigma_1} \right) + \text{trace} \left(\frac{\Sigma_2^{-1} \Sigma_1}{2} \right) + \frac{1}{2} (\mu_1-\mu_2)^T \Sigma_2^{-1} (\mu_1-\mu_2) - \frac{k}{2}$$

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

PART-III : OVERVIEW of VAE.





The goal of VAE is to LEARN the parameters of ϕ of Decoder $q_\phi(z|x)$. This way we can sample from $q_\phi(z|x)$ various values of 'z'. We will also LEARN 'θ' in $p_\theta(x|z)$. The derivation of the LOSS Fn of VAE can be solved through VARIATIONAL INFERENCE.

Consider $p(z|x) = \dfrac{p(x|z)\cdot p(z)}{p(x)}$ ——— (A) Bayes Theorem

where $p(x)$ = Denominator of RHS $= \int p(x|z)\cdot p(z)\cdot dz$ This integral is often INTRACTABLE

$= \int p(x,z)\,dz$ can be written as a joint distribution but is not available in the CLOSED form to solve.

Here, the alternative is to approximate the intractable $p(z|x)$. POSTERIOR by another distribution $q(z|x)$ which has a tractable solution. Variational Inference converts the problem into that of OPTIMISATION. Thus we try to find that $q(z|x)$ using K-L divergence and minimize it.

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

By definition of expectation wrt $Q_\phi(z|x)$

$$= \mathop{\mathbb{E}}_{z\sim Q_\phi(z|x)}\left[\log \frac{Q_\phi(z|x)}{P_\theta(z|x)}\right]$$

$$= \mathop{\mathbb{E}}_{z\sim Q_\phi}\left[\log\left(Q_\phi(z|x)\right) - \underbrace{\log\left(P_\theta(z|x)\right)}_{\text{let (A)}}\right] \quad \text{——— (B)}$$

$$= \mathop{\mathbb{E}}_{z\sim Q_\phi}\left[\log\left(Q_\phi(z|x)\right) - \log\left(\frac{P_\theta(x|z)\cdot P_\theta(z)}{P_\theta(x)}\right)\right]$$

$$= \mathop{\mathbb{E}}_{z\sim Q_\phi}\left[\log\left(Q_\phi(z|x)\right) - \log P_\theta(x|z) - \log P_\theta(z) + \underbrace{\log P_\theta(x)}_{\text{constant wrt 'z'}}\right]$$

$$= \mathop{\mathbb{E}}_{z\sim Q_\phi}\left[\log\left(Q_\phi(z|x)\right) - \log P_\theta(x|z) - \log P_\theta(z)\right] + \log\left(P_\theta(x)\right) \qquad \because \mathbb{E}[\text{const}] = \text{const.}$$

$$D_{KL}\left(Q_\phi(z|x) \,\|\, P_\theta(z|x)\right) - \log P_\theta(x) = \mathop{\mathbb{E}}_{z\sim Q_\phi}\left[\log\left(Q_\phi(z|x)\right) - \log P_\theta(x|z) - \log P_\theta(z)\right]$$

$$\underbrace{\log P_\theta(x) - D_{KL}\left(Q_\phi(z|x) \,\|\, P_\theta(z|x)\right)}_{-\text{Loss funct}^n} = \mathop{\mathbb{E}}_{z\sim Q_\phi}\left[\log P_\theta(x|z)\right] - \underbrace{\mathop{\mathbb{E}}_{z\sim Q_\phi}\left[\log Q_\phi(z|x) - \log P_\theta(z)\right]}_{\text{Note: KL Divergence form}}$$

$$\therefore \log P_\theta(x) - D_{KL}\left(Q_\phi(z|x) \,||\, P_\theta(z|x)\right) = \underbrace{\mathbb{E}_{z\sim Q_\phi}\left[\log P_\theta(x|z)\right]}_{\text{Term (A)}} - \underbrace{D_{KL}\left(Q_\phi(z|x) \,||\, P_\theta(z)\right)}_{\text{Term (B)}}$$

This is the OBJECTIVE function of VAE.

Term (A) $= \mathbb{E}_z\left[\log P_\theta(z|x)\right]$ is the reconstruction likelihood.

Term (B) $= D_{KL}\left[Q_\phi(z|x) \,||\, P_\theta(z)\right]$ ensures that our LEARNT distribution Q is similar to Prior 'z'

$$\therefore \text{LOSS fn} = \underbrace{-\mathbb{E}_{z\sim Q_\phi}\left[\log\left(P_\theta(x|z)\right)\right]}_{\text{Reconstruction loss.}} + \underbrace{D_{KL}\left[Q_\phi(z|x) \,||\, P_\theta(z)\right]}_{\text{Regularizer}}$$

$P(z)$ is assumed to be known

So our target is to LEARN the parameters 'ϕ' & 'θ'

PART IV : Loss fn TERMS in depth.

The regularizer $D_{KL}\left[Q_\phi(z|x) \,||\, P_\theta(z)\right]$ is the divergence between two distributions. We assign $P_\theta(z) \sim N(0, I)$

$\therefore$ we compare this term of the Loss fn to the k-L divergence bet^n two Gaussians done previously. $\therefore$ $M_2 = 0$; $\Sigma_2 = I$

& $M_1 = M_1(x)$; $\Sigma_1 = \Sigma_{1,\phi}(x)$

$\therefore$

$$\text{Regularizer } D_{KL}\left[Q_\phi(z|x) \,||\, P_\theta(z)\right] = \frac{1}{2}\left[\log\frac{1}{|\Sigma_\phi(x)|} - \underbrace{K}_{=\sum_K 1} + \underbrace{\text{trace}\left[\Sigma_\phi(x)\right]}_{\text{summing over k elements}} + \underbrace{\mu_\phi(x)^T \cdot \mu_\phi(x)}_{\text{scalar qty } \therefore \text{ can be squared}}\right]$$

- where 'k' is the dimension of the Gaussian.
- also $|\Sigma_\phi(x)|$ is the determinant of the diagonal matrix = product of its diagonal elements.

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

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

$\therefore$ Regularizer

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

for numerical stability $\Sigma_\phi(x)$ is replaced by $\exp\Sigma(x)$

$$= \frac{1}{2}\sum_K\left[\exp\left(\Sigma(x)\right) + \mu_\phi^2(x) - 1 - \Sigma_\phi(x)\right]$$

OPTIMIZATION of THE LOSS FUNCTION

$-L(\theta,\phi) = \log P_\theta(x) - D_{KL}\left[Q_\phi(z|x) \,||\, P_\theta(z|x)\right]$ is referred to as VARIATIONAL LOWER BOUND or EVIDENCE LOWER BOUND (ELBO).

$$\theta^*, \phi^* = \underset{\theta,\phi}{\operatorname{argmin}} \{L(\theta,\phi)\}$$

We know that $D_{KL}\left[Q_\phi(z|x) \,||\, P_\theta(z|x)\right]$ is always positive ≥ 0 $\therefore L(\theta,\phi) \leq \log P_\theta(x)$

Thus minimizing the loss fn we are actually maximizing the probability of generating real data samples.

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

During optimization, we apply the alternate optimization technique. i.e.

$$\text{Iteration \# 1}: \begin{cases} \theta^* = \nabla_\theta\{L(\theta,\phi)\} & \phi = \text{const.} \\ \phi^* = \nabla_\phi\{L(\theta^*,\phi)\} & \theta^* \text{ is kept constant.} \end{cases}$$

$$\text{Iteration \# 2} \begin{cases} \theta^{**} = \nabla_{\theta^*}\{L(\theta^*,\phi^*)\} & \phi^* \text{ is kept constant.} \\ \phi^{**} = \nabla_{\phi^*}\{L(\theta^{**},\phi^*)\} & \theta^{**} \text{ is kept const.} \end{cases}$$

NOTE: It is observed that differentiating wrt θ is straight forward, but derivative wrt ϕ is not possible. To overcome this issue REPARAMETRIZATION trick is used.

Optimization is carried out wrt both θ & ϕ, to LEARN $q_\phi(z|x)$ & $P_\theta(x|z)$ simultaneously

Derivative wrt 'θ': ie $\frac{\partial L}{\partial \theta_i} = \nabla_\theta\{L(\theta,\phi)\}$

$$= \nabla_\theta\left\{-\underset{z\sim Q_\phi(z|x)}{\mathbb{E}}\left[\log P_\theta(x|z)\right] + \underbrace{\frac{1}{2}\sum_k\left[\exp\left(\Sigma_\phi(x)\right) + \mu_\phi^2(x) - 1 - \Sigma_\phi(x)\right]}_{\text{Term being constant wrt } '\theta',\ \text{its derivative} = \text{ZERO}}\right\}$$

$$= -\nabla_\theta \underset{z\sim Q_\phi(z|x)}{\mathbb{E}}\left[\log P_\theta(z|x)\right]$$ we can easily move ∇_θ inside $\mathbb{E}_z$ sign. Also $z^{(l)} \sim Q_\phi(z|x)$

$$= \frac{1}{L}\sum_{l=1}^{L}\nabla_\theta\left[\log P_\theta(x|z^l)\right]$$ NOTE: We replace the Expectation by the average value according to the MONTE CARLO concept.

Derivative wrt 'ϕ': Here we will consider the optimal value of θ_i^* calculated above

$$\therefore \hat{\phi}_i = \nabla_\phi\{L(\theta^*,\phi)\}$$

$$= \nabla_\phi\left\{-\underset{z\sim Q_\phi(z|x)}{\mathbb{E}}\left[\log P_\theta(x|z)\right] + \frac{1}{2}\sum_k\left[\exp\left(\Sigma_\phi(x)\right) + \mu_\phi^2(x) - 1 - \Sigma_\phi(x)\right]\right.$$

This derivation ∇_ϕ is difficult to estimate because 'ϕ' appears in the distribution wrt which the expectation is taken

NOTE:

$$\nabla_\phi \underset{Q_\phi(z|x)}{\mathbb{E}}\left[f(z)\right] \neq \underset{Q_\phi(z|x)}{\mathbb{E}}\left[\nabla_\phi f(z)\right]$$

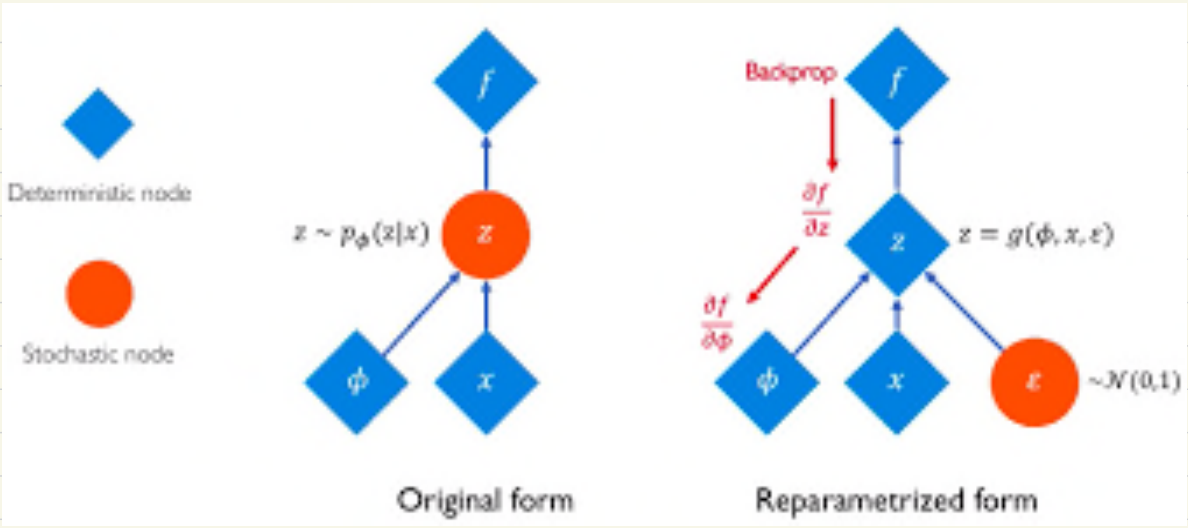
If we can somehow write this expectation in such a way that ϕ appears INSIDE the expectation, then we can push the gradient INSIDE the Expectation, ie if we can write

$$\mathbb{E}_{Q_\phi(z|x)}[f(z)] = \mathbb{E}_{P(\epsilon)}[f(g_\phi(\epsilon, x))] \quad \text{such that} \quad \boxed{z = g_\phi(\epsilon, x)}$$

→ any LINEAR TRANSFORMATION with $\epsilon \sim N(0,1)$

In our case, $g_\phi(\epsilon, x) = \mu_\phi(x) + \epsilon \odot \Sigma_\phi^{1/2}(x)$

$= z \sim N(\mu(x), \Sigma(x))$



$$\therefore \hat{\phi}_i = \nabla_\phi \{ L(\theta^*, \phi) \}$$

$$= \nabla_\phi \left\{ -\mathbb{E}_{z^l \sim P(\epsilon)} \left[\log P_\theta(x|z^{(l)}) \right] + \frac{1}{2} \sum_k \left[\exp(\Sigma_\phi(x)) + \mu_\phi^2(x) - 1 - \Sigma_\phi(x) \right] \right\}$$

NOTE: ϕ is not present here
$\therefore \nabla_\phi$ can be now moved inside

$$= -\mathbb{E}_{z \sim p(\epsilon)} \left[\nabla_\phi \left(\log P_\theta(x|z^{(l)}) \right) \right] + \nabla_\phi \left[\frac{1}{2} \sum_k \left(\exp(\Sigma_\phi(x)) + \mu_\phi^2(x) - 1 - \Sigma_\phi(x) \right) \right]$$

Replace E by mean by MONTE CARLO

$$= -\frac{1}{S} \sum_{s=1}^{S} \nabla_\phi \left[\log P_\theta(x|z^{(l)}) \right] + \nabla_\phi \left[\frac{1}{2} \sum_k \left(\exp(\Sigma_\phi(x)) + \mu_\phi^2(x) - 1 - \Sigma_\phi(x) \right) \right]$$