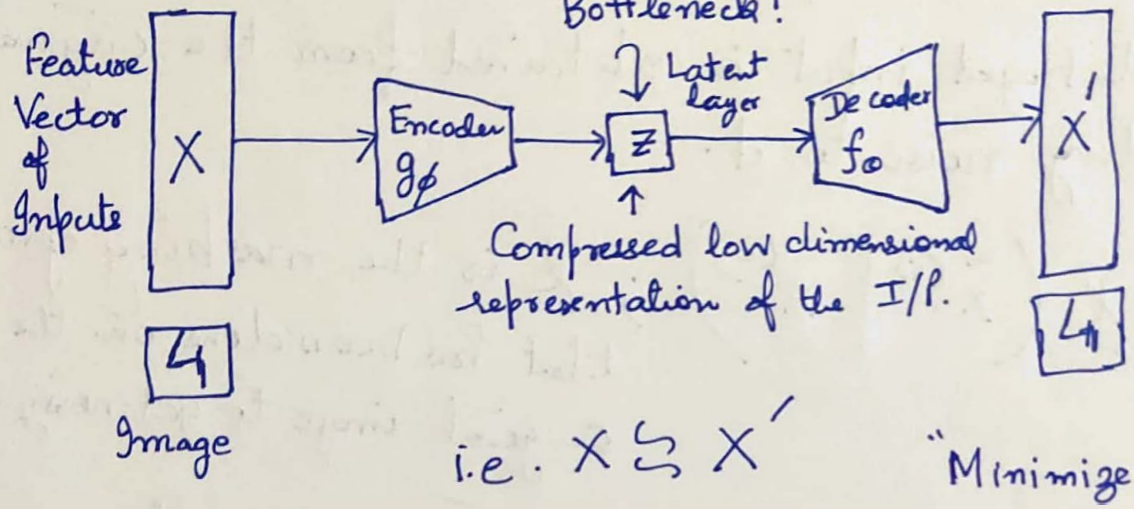
Autoencoders

① Let's take the example of 'Image Reconstruction'.

Input ← ---- Ideally they are identical ----→ Reconstructed Input

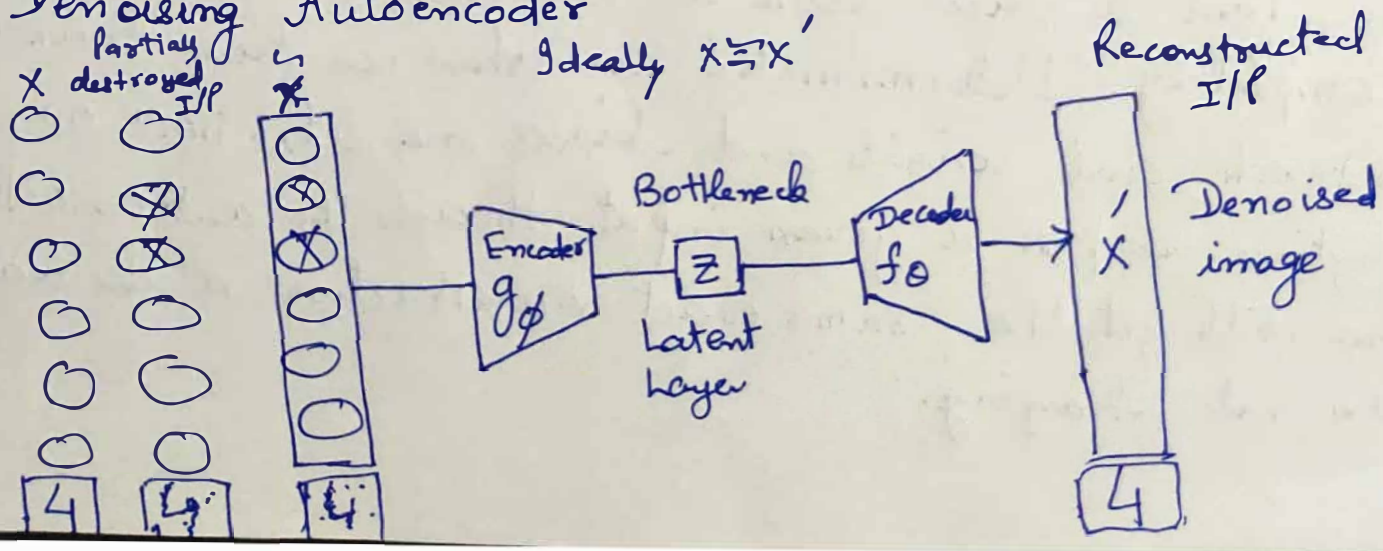


Stacked Autoencoder

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

θ and ϕ are the parameters which define the encoder and decoder. These are basically the weights and biases of the encoder and decoder modules. as these modules are basically the neural networks. i is an index which defines the features of I/P and it varies from 1 to n. $f_\theta(g_\phi(x^{(i)}))$ represents the data flow from the encoder to the decoder.

③ Denoising Autoencoder



(1) (2)

$$\tilde{x}^{(i)} = \chi\left(\tilde{x}^{(i)} \mid x^{(i)}\right)$$

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

(2) Partially destroyed input is obtained from the original image by adding noise to it.

$$\tilde{x}^{(i)} = \chi\left(\tilde{x}^{(i)} \mid x^{(i)}\right)$$; χ is the masking operation that has been done on the original image to get noisy I/P

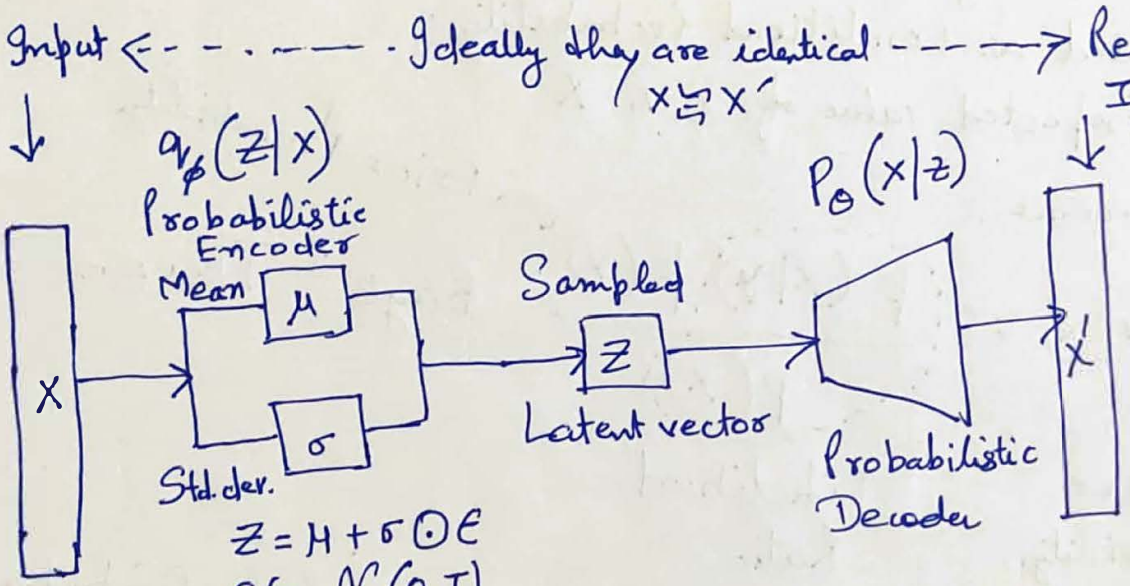
(3) Feed $\tilde{x}^{(i)}$ into the autoencoder. Since Z is the compressed version of the original image, and it will represent the original image by small number of features in the vector. If we can reconstruct the image back, then lot of BW (Bandwidth) can be saved while transmitting an image. Noise will be added to the original during training process.

Variational Autoencoders :-

(4) Take a closer look at the latent layer, it is completely Deterministic i.e. when we have trained the network and weights and biases are set, now on anytime we pass a given input through the autoencoder, we will get the same exact reconstruction as the weights are not changing.

① In contrast, the variational autoencoders (VAEs) ③ introduce an element of probabilistic twist on this idea of autoencoding. This will allow us to not only perform reconstructions, but generate new images similar to the input instances, similar to the input data, but importantly not forced (strictly) to the reconstructions only.

②



$z = \mu + \sigma \odot \epsilon$

$\epsilon \hookrightarrow \mathcal{N}(0, I)$

$\odot \rightarrow$ Circled dot operator $\rightarrow$ XNOR gate

Hadamard product (multiplication symbol) element-wise $\times$ of matrices of same size

Variational Autoencoder (VAE)

③ Replaced single deterministic layer z with a random sampling operation. Now, we define a mean and standard deviation that captures a PD (Probability Distribution) over the latent variable z. We have converted a single vector space into a vector of means of latent variables and a vector of SDs of these latent variables that parametrize the PDFs over latent variables.

① Loss function is given by

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

NOTE:-

$p(x)$: Probability of r.v. X.

$p(x/y)$: Probability of r.v. X provided Y has happened, also called as Conditiond Probability.

$E(\cdot)$: Expected value of r.v. X

KL Divergence

$$p(y|x) = \frac{p(x|y) \cdot p(y)}{p(x)}$$

Prior Probability

Bayé's Theorem

Posterior Probability

Likelihood Ratio

$$= \frac{p(x,y)}{p(x)} \rightarrow \text{Joint Prob. Distribution}$$

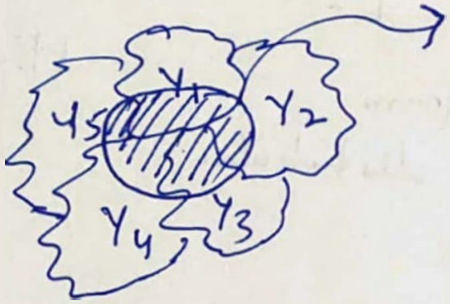
$\longrightarrow$ (1)

Theorem of Total Probability:-

Let $Y_1, Y_2, \ldots, Y_N$ 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,

$$p(X) = \sum_{i=1}^{N} p(X|Y_i)\, p(Y_i) \quad -(2)$$

Note Continued:-



$$P(X) = \sum_{i=1}^{5} P(X, Y_i)$$

$$= \sum_{i=1}^{4} P(X/Y_i) P(Y_i)$$

Expectation of r.v. X :- E(X)

Expected value of random variable is a weighted average of the possible values of X_k that it can take; each value being weighted according to the probability of that event and is defined as

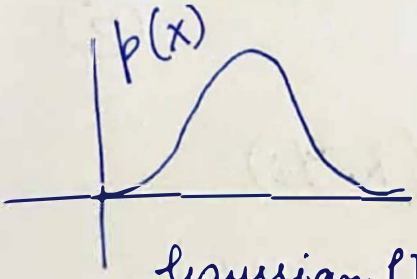
$$E(X) = \sum_{i=1}^{K} x_i P(X = x_i) \hookrightarrow E_P(X)$$; X has the probability density function as P.

K-L Divergence :-

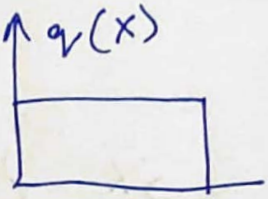
Kullback- Leibler divergence (KL) is a measure of how one probability distribution is different from the second. for the discrete probability distributions P & Q, the K-L divergence between P & Q is defined as

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

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



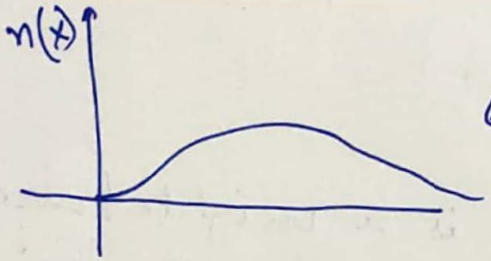
quite different



K-L divergence High

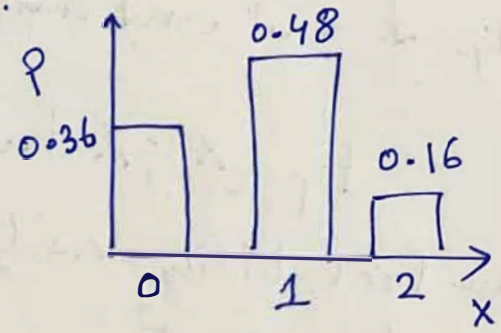
Gaussian Prob. distribution

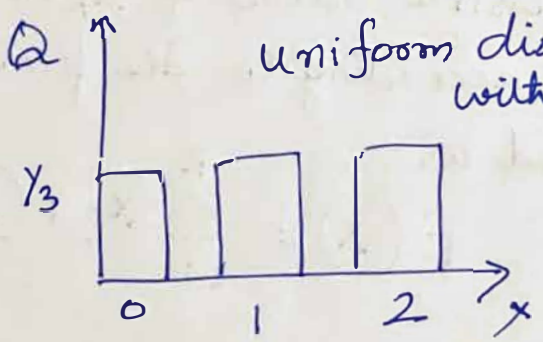
Uniform Prob. distribution



KL divergence close to zero (Nearly similar)

eg:-





$$D_{KL}(Q \| P) = D_{KL}(Q \| P)$$

$$= \sum_{x} Q(x) \log\left[\frac{Q(x)}{P(x)}\right]$$

$$= \frac{1}{3} \ln\left[\frac{0.33}{0.36}\right] + \frac{1}{3} \ln\left[\frac{0.333}{0.48}\right] + \frac{1}{3} \ln\left[\frac{0.333}{0.16}\right]$$

$$\approx 0.09637 \text{ nats} \quad \text{(very small)}$$

Properties of KL divergence:-

1) $KL(P \| Q) \neq KL(Q \| P) \geq 0$
2) $KL(P \| Q) \neq KL(Q \| P)$ (Not Symmetric)

Note 2:-

① Let X be an $K \times 1$ random vector (A random vector also called "multivariate r.v." is an K-dimensional column vector $X \in \mathbb{R}^{K \times 1}$ whose entries are random variables) The X is said to be multivariate normally distributed with mean μ and covariance Σ:

$$X \sim \mathcal{N}(\mu, \Sigma)$$

iff its PDF is given by

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

where μ is an $K \times 1$ real vector and Σ is an $K \times K$ positive definite matrix.

$$\mu = E[X] = \left(E[X_1], E[X_2], \cdots, E[X_K]\right)^T$$

K-dimensional Mean Vector

$$\Sigma_{i,j} = E\left[(X_i - \mu_i)(X_j - \mu_j)\right] = \text{Cov}[X_i, X_j]$$

$K \times K$ - Covariance Matrix

$1 \le i \le K$
$1 \le j \le K$

The inverse of the covariance matrix is called Precision Matrix denoted by Σ^{-1}.

Theorem: Let x be an $K \times 1$ random vector. Assume two multivariate normal distributions P and Q specifying the probability distribution of x as

$$P(x): x \hookrightarrow \mathcal{N}(\mu_1, \Sigma_1)$$
$$Q(x): x \hookrightarrow \mathcal{N}(\mu_2, \Sigma_2)$$

① Then the kullback - Leibler divergence of P from Q is given by (P & Q have the same dimension k)

$$KL[P||Q] = \frac{1}{2}\left[(\mu_2-\mu_1)^T \Sigma_2^{-1}(\mu_2-\mu_1) + tr\left(\Sigma_2^{-1}\Sigma_1\right) - \ln\frac{|\Sigma_1|}{|\Sigma_2|} - k\right] \quad -(1)$$

Note:- Multivariate Normal Distribution is a generalization of the one-dimensional (univariate) normal distribution to higher dimensions. A random vector K is said to be k-variate normally distributed if every linear combination of its k components has a univariate Normal distribution. The multi-variate distribution is used to describe any set of correlated real-valued random variables each of which clusters around a mean value.

Proof of eq(1) :-

We know that

$$KL[P(X)||Q(X)] = \sum_x P(x)\log\left(\frac{P(x)}{Q(x)}\right) \quad -(2)$$

Also,

$$P(X) = \frac{1}{\sqrt{(2\pi)^k|\Sigma_1|}}\exp\left(\frac{-(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)}{2}\right) \quad -(3)$$

Take log on both sides of equation (3)

$$\Rightarrow \log P(X) = -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) \quad -(4)$$

Similary,

$$\log Q(x) = -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_2| - \frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2) \quad -(5)$$

(1) Eq (2) can be rewritten as

$$KL(P(X) \| Q(X)) = \sum_x P(x)\left[\log P(x) - \log Q(x)\right] - (6)$$

(2) Substituting equations (4) and (5) in eq (6), We get

$$KL(P(X) \| Q(X)) = \sum_x P(x)\left\{-\frac{K}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right.$$
$$\left. + \frac{K}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma_2| + \frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right\}$$

(3) On Simplification, We get

$$KL(P(X) \| Q(X)) = \sum_x P(x)\left[\frac{1}{2}\log\left(\frac{\Sigma_2}{\Sigma_1}\right) + \frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right.$$
$$\left. - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right] - (7)$$

(4) From eq(7), let's consider expression part by part:

$$\sum_x P(x)\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) = E_p\left[\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right] - (8)$$

(5) Let's look at some of the identities which are required.

$$E(X^TAX) = E(tr(X^TAX)) = E(tr(AXX^T)) = tr(E(AXX^T))$$
(a) (b) (e)

(6) Also let's have a trick for Trace and Expectation:

If X is a scalar, then $E(X) = E(tr(X))$, since trace of X is scalar. (a)

$$tr(AB) = tr(BA), \quad tr(ABC) = tr(BCA) = tr(CAB)$$
(b) (c)

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

$$E(tr(X)) = tr(E(X))$$ (e)

$$\frac{1}{2} E_p\left[(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)\right] = E_p\left[tr\left(\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)\right)\right] \quad (10)$$

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

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

↖ Covariance Matrix

$$= tr\left[\Sigma_1 \frac{1}{2} \Sigma_1^{-1}\right] = \frac{1}{2} tr\left[I_K\right] = \frac{K}{2} \quad (9)$$

(1) Consider the second part of eq (7) for equation to Calculate $KL(P(x)||Q(x))$, which is

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

let's say $(A+B)^T \Sigma_2^{-1}(A+B)$

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

$A^T\Sigma_2^{-1}A + A^T\Sigma_2^{-1}B + B^T\Sigma_2^{-1}A$ Basically same

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

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

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

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

① Rewriting the previous expression:

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

↓ CONSTANT

② $\Rightarrow \frac{1}{2}\text{tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2) + 0$ (10)

similar to earlier derivation

$\because E(\text{constant}) = \text{constant}$

③ $E_p\left[(x-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2)\right]$

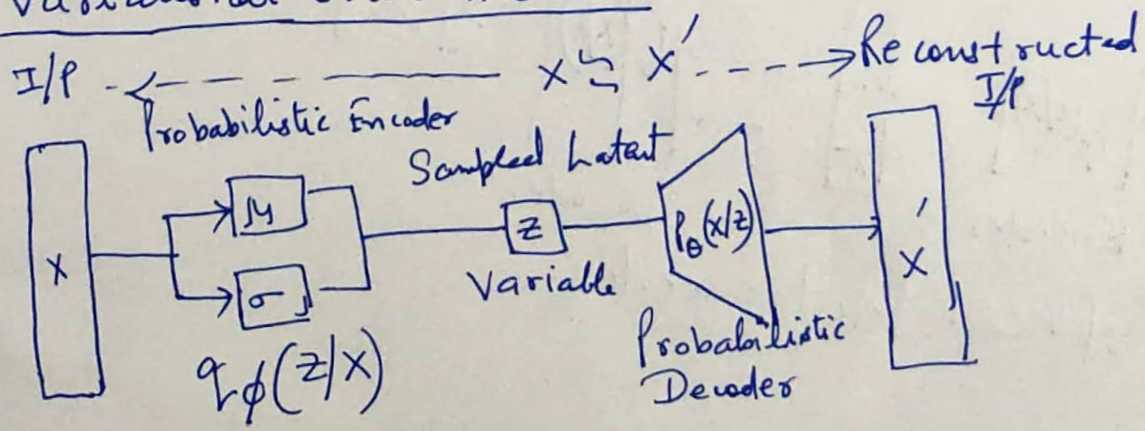
$= \left[(E_p(x)-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2)\right]$

$= (\mu_1-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2) = 0$

④ Substitute eq(9) and eq(10) in eq(7) to obtain final and simplified KL-Divergence.

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

Variational Autoencoder:



$z = \mu + \sigma \odot \epsilon$

$\epsilon \sim N(0, I)$

① Loss function in a VAE is given by

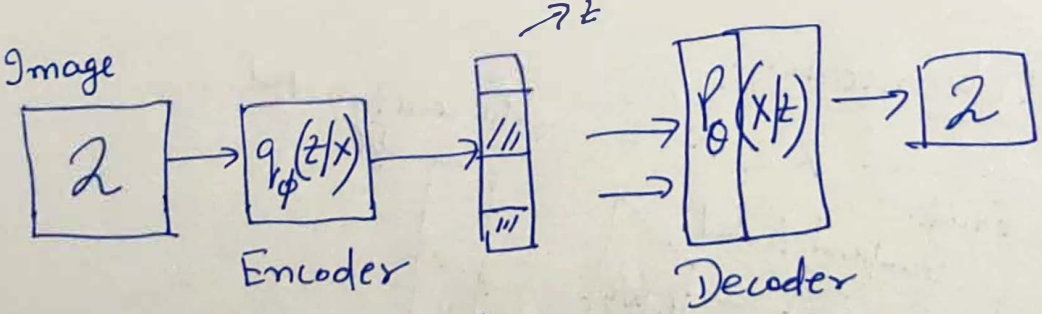
$$\text{Loss} = L(\theta, \phi) = -\underset{z \sim q_\phi(z/x)}{E}\left[P_\theta(x/z)\right] + D_{KL}\left(q_\phi(z/x) \,||\, P_\theta(z)\right)$$

② Expectation operation which is carried on to the Prob. distribution $q_\phi(z/x)$ ∴ We can write $E[x] = E_p$; Here $P = q_\phi(z/x)$.

$P_\theta(x/z)$ is any r.v. Y.

The goal of VAE :-

The goal of VAE is to find a distribution $q_\phi(z/x)$ of some latent variables from which we can sample ~~from~~ $z \sim q_\phi(z/x)$ to generate new samples $x's$ in $P_\theta(x/z)$ i.e. We want to find $q_\phi(z/x)$ from which we shall be sampling z from $q_\phi(z/x)$. ∴ z will be our sampled latent vector. Once it is done, then we will feed samples of z to the decoder and then we will generate new samples x'. The new samples mean that these are not present in x.

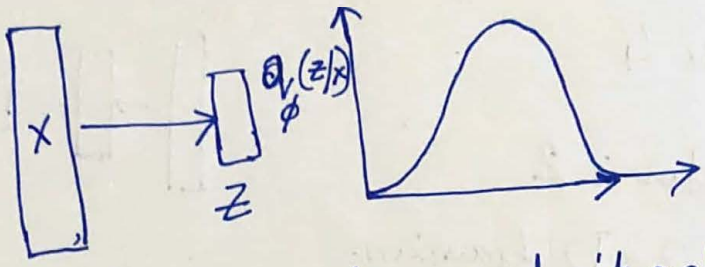


Typical Autoencoder.

(1.) Traditional autoencoder data flow with one hidden unit is given by

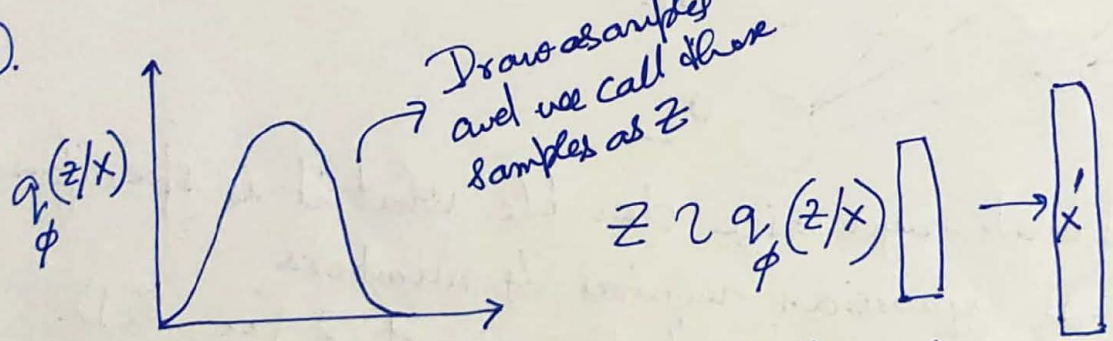
$$z = f(W_1^T x + b_1)$$
$$\hat{x} = g(W_2^T z + b_2)$$

(2) In the case of Variational Autoencoder:



At the end of the encoder, we don't get a value but we get a distribution: $q_\phi(z/x)$. Or more precisely we get the parameters of a distribution. In particular, encoder outputs a mean and variance which represent a Gaussian. Bayesian Machine learning is about learning distributions and not learning point estimates. ∴ Instead of actually find z, we are finding $q_\phi(z/x)$ that tells us the PDF of z given x with ϕ as the neural network (encoder NN) parameters.

(3.)

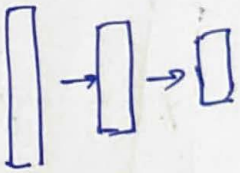


We have a distribution $q_\phi(z/x)$ and we need actual numbers to pass in through the rest of the NN. We sample the gaussian distribution to obtain z which is sample vector and then we can pass z from the decoder. Follow the flow of information through decoder which is again a NN

① The output of the decoder is again a distribution $p_\theta(x/z)$ From this distribution, we generate samples

② eg:- Parameterizing a Gaussian:-

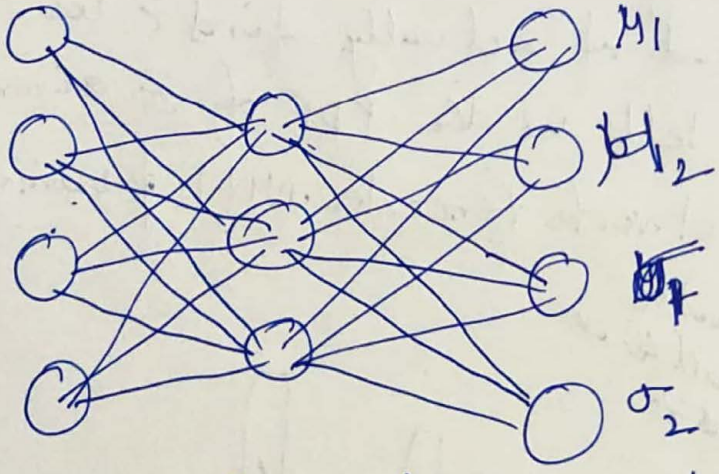
Suppose our encoder has the sizes (4, 3, 2)



- Input dimensionality is 4
- HL size is 3
- Size of latent vector is 2

ie $q_\phi(z/x)$ is a 2-D Gaussian.

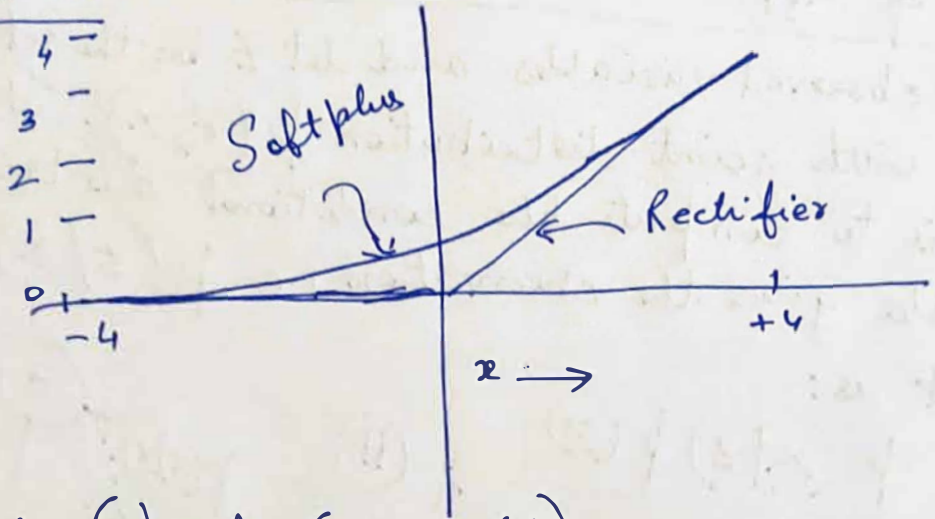
We will use an axis-aligned Gaussian (one variance/param per dim) rather than a full covariance Gaussian (requires full D×D matrix)



③ Make the final layer size double what it is specified as
eg. A 2-D Gaussian requires 4 numbers
Use first 2 for mean and the next 2 for SD.

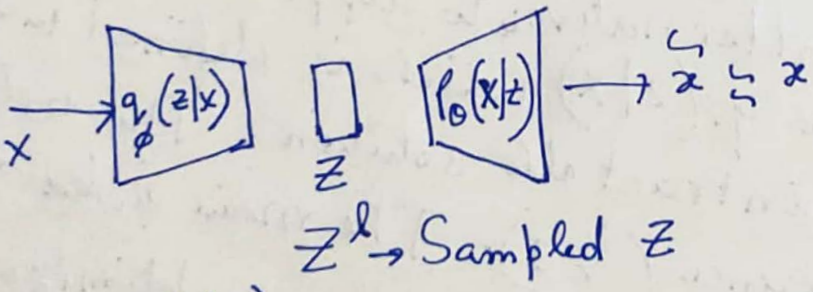
④ SD must be a strictly +Ve number, but NN can output any number. How can we fix this?

① Softplus :- Nonlinearities

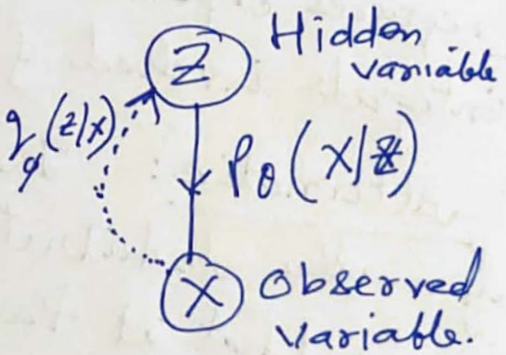


- Softplus $(a) = \log(1+\exp(a))$
- Smooth version of ReLU
- Approximately linear when I/P is large. and looks more like
- exponential when I/P is small.
- Smooth / continuous / Differentiable.
- and always > 0

② Recall the VAE goal :-



$z^l \rightarrow$ Sampled z



Learn $q_\phi(z|x)$

Sample $q_\phi(z|x)$ to produce z^l

Feed z^l to decoder

Generate $\tilde{x}$, new samples, but not present in input vector

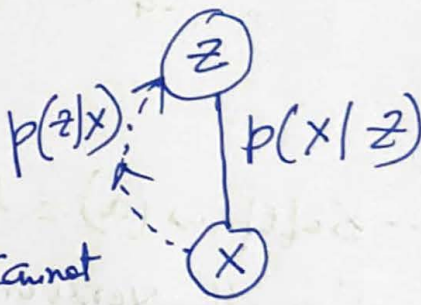
③ Let's now look at variational autoencoders in a more analytical and mathematical framework:

(1) The Problem of Approximate Inference:-

Let X be a set of observed variables and 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:

$$p(Z|X) = \frac{p(X|Z)\,p(Z)}{p(X)} \rightarrow (A)$$



(2) Evaluating (A) is difficult because $p(X)$ cannot be solved.

Reason: $p(X) = \int_Z p(X|Z)\,p(Z)\,dZ = \int_Z p(X, Z)\,dZ$

The integral is not available in closed form & is intractable (requires exponential time to compute) due to multiple integrals involved for latent vector Z.

(3) 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 intractable solution. This is done using variational Inference (VI). The main idea of VI is to pose the inference problem as an optimization problem. By modeling $P(Z|X)$ using $Q(Z|X)$ where $Q(Z/X)$ has a simple distribution such as Gaussian.

As discussed earlier let's calculate KL divergence between $P(Z/X)$ and $Q(Z/X)$

①

$$D_{KL}\left(Q_\phi(z|x) \,||\, P_\theta(z|x)\right) = \sum_z Q_\phi(z|x) \log\left(\frac{Q_\phi(z|x)}{P_\theta(z|x)}\right) \quad (17)$$

We will calculate the KL-divergence and will try to minimize the divergence and then $P_\theta(z|x)$ will be approximated.

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

$$= \underset{z \sim Q_\phi(z|x)}{E}\left[\log\left(Q_\phi(z|x)\right) - \log\left(P_\theta(z|x)\right)\right] \quad - (B)$$

Substituting (A) in (B).

$$= \underset{(z)}{E}\left[\log\left(Q_\phi(z|x)\right) - \log\left\{\frac{P_\theta(x|z)\,P_\theta(z)}{P_\theta(x)}\right\}\right]$$

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

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

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

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

Revisiting the above equation,

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

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

$$= \underset{z}{E}\left[\log\left(P_\theta(x|z)\right)\right] - \underset{z}{E}\left[\log Q_\phi(z|x) - \log P_\theta(z)\right]$$

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

This is VAE objective function where the first term represents the reconstruction likelihood and the second term ensures that our learned distribution Q is similar to the prior distribution P.

$$\text{Loss} = -\text{objective function}$$

$$= -\underset{z \sim Q_\phi(z|x)}{E}\left\{\underbrace{\log P_\theta(x|z)}_{\text{(Reconstruction loss)}}\right] + \underbrace{D_{KL}\left[Q_\phi(z|x) \,||\, P_\theta(z)\right]}_{\text{(Regularization term)}}$$

Hence Proved !!

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

$$= -L(\theta, \phi)$$

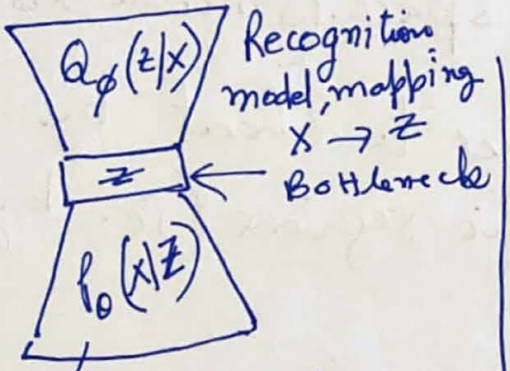
∴ Our target is to find optimal θ, ϕ such that

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

① More Intuition about Loss function :-

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

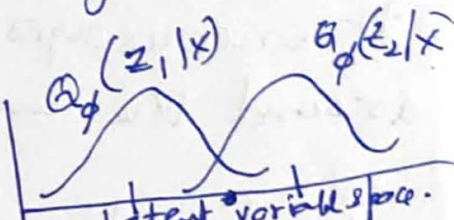
← Log likelihood → ← Regularizer →



Generative Model
Mapping z to X

$P_\theta(x|z)$

So when we take log of Gaussian, we get a squared error between the data sample $\hat{x}$ and mean of the gaussian distribution



KL divergence does not allow PDF of latent variables and penalizes if deviates from $N(0,1) = P_\theta(z)$

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

$$\log P_\theta(x|z) \propto (x - \mu_\theta(z))^T \Sigma_\theta^{-1}(z) (x - \mu_\theta(z))$$

← Squared Reconstruction error →

It basically tells us that how far is $\hat{x}$ from z

This squared reconstruction also is known as Data Fidelity term in inverse problems.

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

$Q_\phi(z|x)$ ↑ Inferred Latent Distribution

$P_\theta(z)$ ↑ Fixed prior on latent Distribution

① Common Choice of Prior:-

Normal Distribution

$$p_\theta(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

- It encourages encodings to distribute encodings evenly around the center of the latent space. (Smooth Distribution)
- Penalize the n/w when it tries to cheat by clustering points in specific regions (i.e. by memorizing data)

② Therefore, we take the inference over latent variable distribution and constrain it to behave nicely by placing a prior on latent distribution, which is an initial hypothesis or guess about what the latent variable will look like.

③ This helps the n/w to enforce a latent space that roughly follows the prior latent distribution.

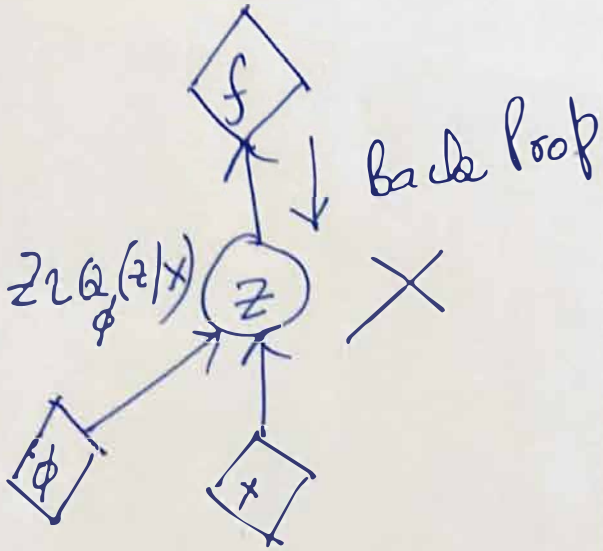
Problem: We cannot backpropagate gradients through sampling layers!

$z \sim \mathcal{N}(\mu, \sigma^2)$

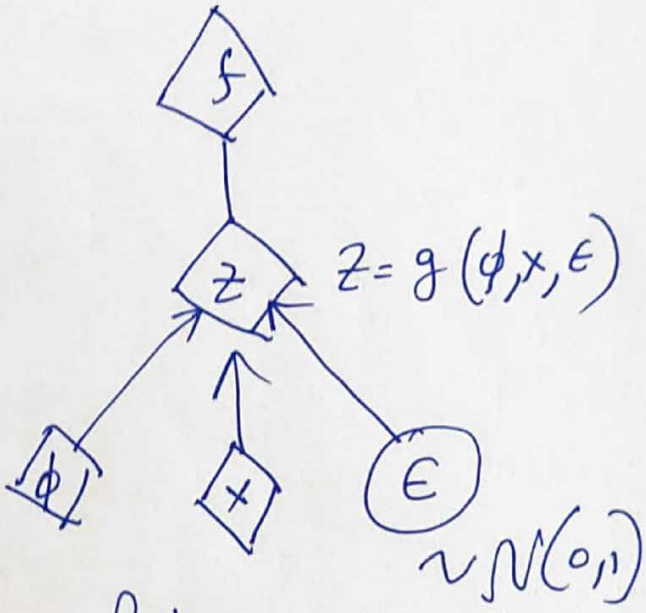
∴ Consider the sampled latent vector z as a sum of
- a fixed vector μ ; - and a fixed vector σ, scaled by random constants drawn from prior distribution

$\Rightarrow z = \mu + \sigma \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$

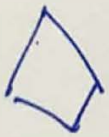
Reparameterize the sampling layer:-



Original form



Reparameterized form



Deterministic Node

Stochastic Node.