Decoder ~~will try to learn~~ is going to take that learned representation and compute a new PDF of input $x$ given the latent distribution $z$.

Encoder and decoder are defined by seperate sets of weights $(\phi, \theta)$

$$L(\phi, \theta, x) = (\text{reconstruction loss}) + (\text{regularization term})$$

reconstruction loss is going to capture the difference between the input and reconstructed output.
e.g. log likelihood $\cdot \|x - \hat{x}\|^2$ For image data MSE b/w o/p and i/p

regularization loss (VAE loss):

$q_\phi(z|x)$ is a distribution on the latent space $z$ given the data $x$.
As part of this learning process, we are going to place a prior on latent ~~learning~~ space $z$.
Some initial hypothesis what $z$ will look like.

By imposing these regularization terms the model will ~~achieve~~ try to enforce z that it learns to follow this prior distribution (p(z))

$$D\left(q_\phi(z|x) \,||\, p(z)\right)$$

---

regularization term will try to enforce ~~this diversion~~.

minimization of this diversion b/w what encoder is trying to infer the probability distribution of $z|x$ and the prior on the latent variable $p(z)$.

avoid overfitting on latent space

encourage latent variables to adopt a distribution which is similar to $P(z)$; prior.

Regularization: or pricing prior is important

Consider the ~~divergence~~ divergence b/w inferred latent distribution and fixed prior.

$$D\left(q_\phi(z|x) \,\|\, p(z)\right)$$

Inferred latent distribution

Fixed prior on latent distribution

Choice of prior:

Common choice of prior — Normal Gaussian

$$p(z) = N(\mu = 0, \sigma^2 = 1)$$

S.D and variance = 1

Normal distribution centered around mean 0

- Encourages encodings to distribute encodings evenly around the center of the latent space
- Penalize the network when it tries to cheat by clustering points in specific regions. (i.e. by memorizing the data).

trender
↑VAE will distribution learned by be distributed evenly around the center of each latent variable. outside the region far away will be penalized.

Inferred latent distribution → Fixed prior on latent distribution

$$D\left(q_\phi(z|x) \,\|\, p(z)\right) = \text{KL-divergence b/w the two distributions}$$

$$-\frac{1}{2}\sum_{j=0}^{k-1}\left(\sigma_j + \mu_j^2 - 1 - \log\sigma_j\right)$$

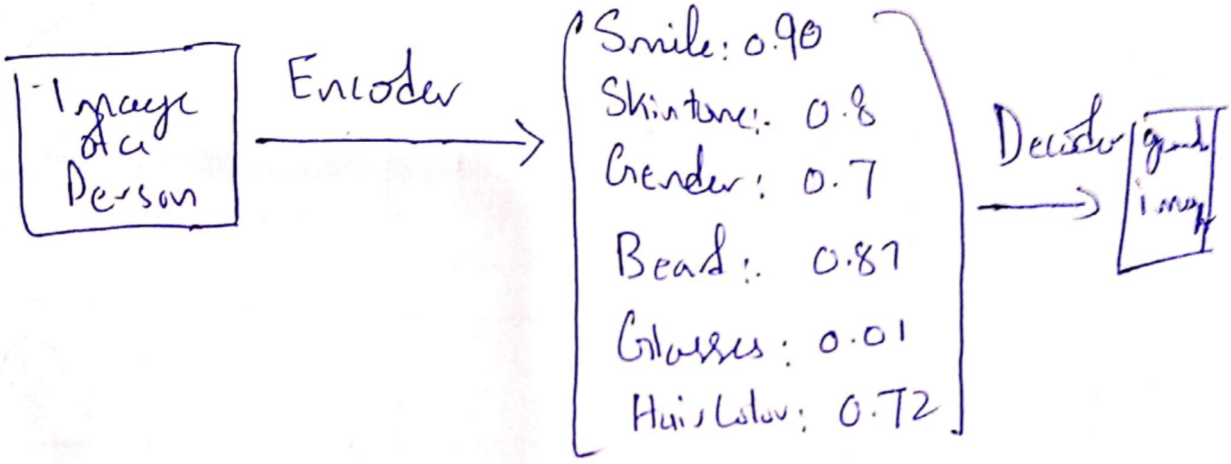
KL-Divergence:

A measure that how one probability distribution is different from another.

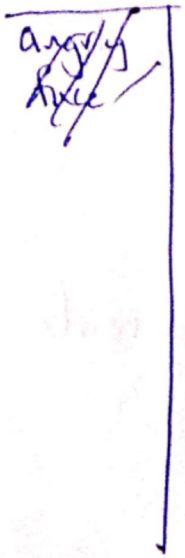
~~Latent vs. t~~

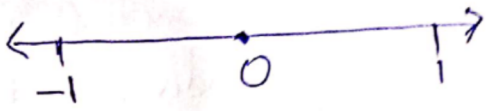
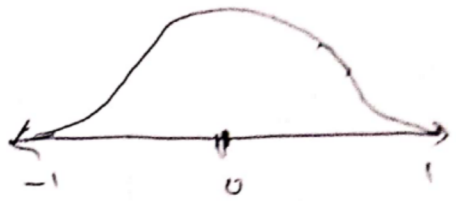
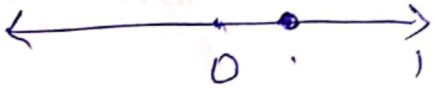
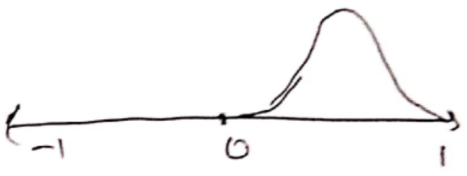
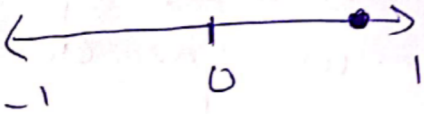
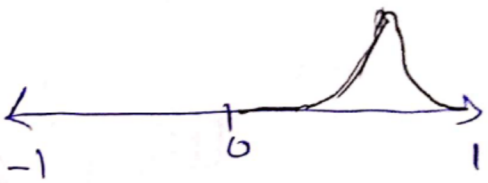
# Auto Encoder Vs. VAE

Auto encoder will learn descriptive attributes.
Example




- A single value to describe each attribute
How about representing each attribute as a range of values.
Using VEA, we can describe using probabilistic terms.



| Type | Smile Discrete Value | Smile Probability Distribution |
| --- | --- | --- |
| Angry face |   |   |
| gentle Smile |   |   |
| Happy face |   |   |
| Very Happy face |   |   |

Each latent attribute is represented as probability distribution.

When decoding: randomly sample from each latent state distribution to generate a vector as input.

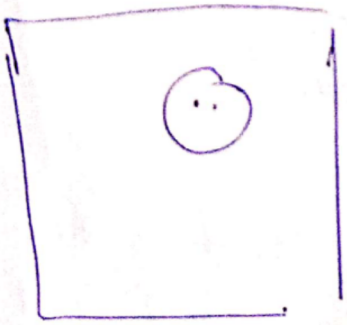
Why we want to regularize?
What is regularization.

What properties do we want from regularization?

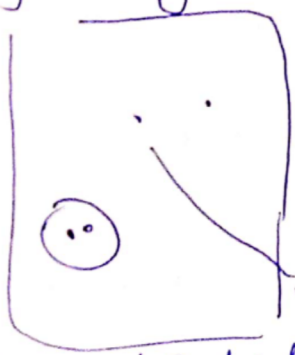
1- Continuity: Points that are close in latent space → similar content after decoding

2. Completeness: sampling from latent space → meaningful content after decoding

Without regularization: Points in latent space could be close but they may not get decoded correctly or they may be ~~far~~ at a distance in the latent space



Points close in latent space are meaningfully decoded.



distant points during decoding

does not result in meaningful content.

Not Regularized

Stochastic Sampling:
Every point in the signal has a non-zero probability of being sampled.

How can we achieve this

VAEs try to encode inputs as distribution which are defined by mean and variance.

Mean and S.D alone are not sufficient

Recall: Loss = Reconstruction + Regularization

Without regularization: the model will try to minimize the reconstruction loss

means could be converged or totally diverged.

Normal prior will encourage learned latent distributions to overlap in latent space.

Normal prior based regularization helps enforce information gradient in the latent space.

Points and distances in the latent space have some relationship with the content reconstructed

Trade off between regularizing and reconstructing. The more we generalize there is a risk of suffering w.r.t quality

How do we back propagate?

It requires deterministic approach.

Idea:

Reparameterizing the sampling layer.

$z \sim N(\mu, \sigma^2)$ — variance

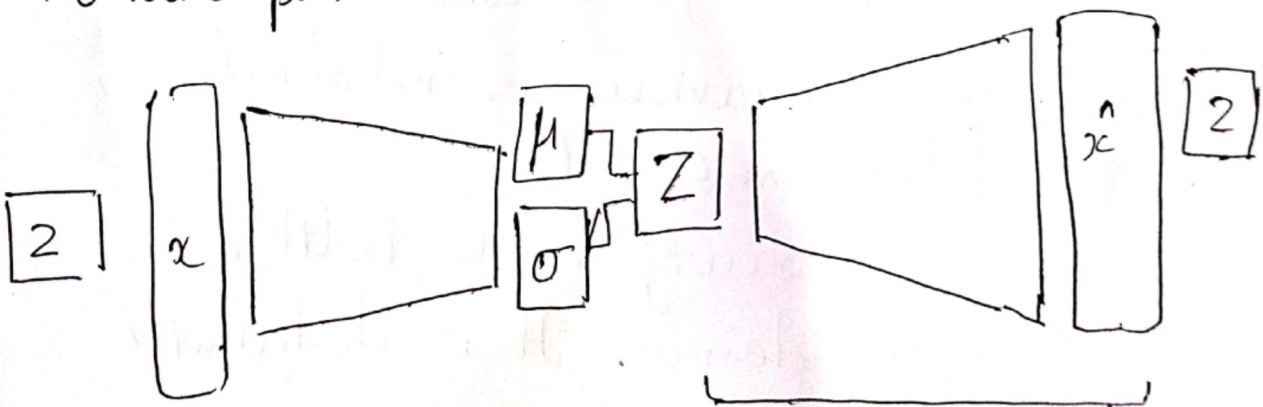
We cannot compute gradients through stochastic sampling.

Consider the sampled latent vector z as a sum of

- a fixed vector $\mu$
- a fixed $\sigma$ vector scaled by random constants from the prior distribution

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

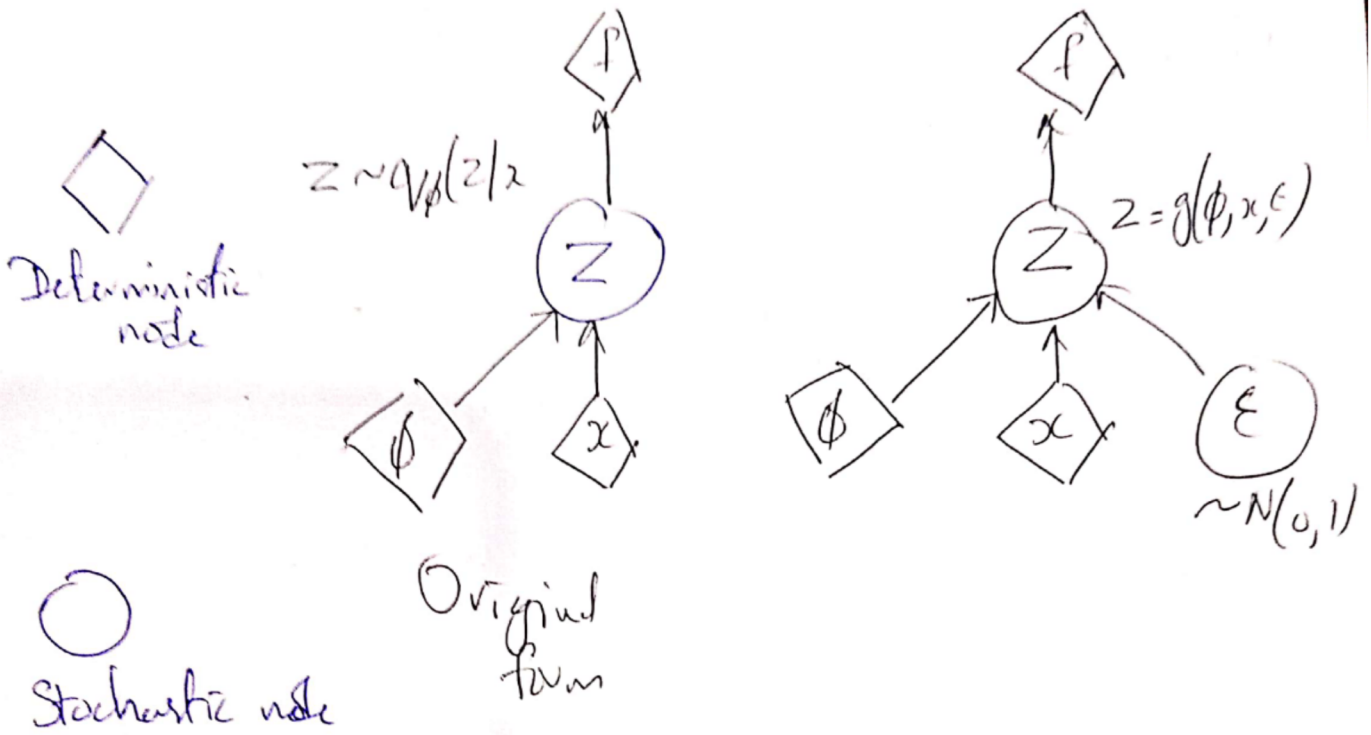
Forward pass:




Encoder computes $q_\phi(z|x)$

Decoder computes $p_\theta(x|z)$

$\mathcal{L}(\phi, \theta, x) = (\text{reconstruction loss}) + (\text{regularization term})$

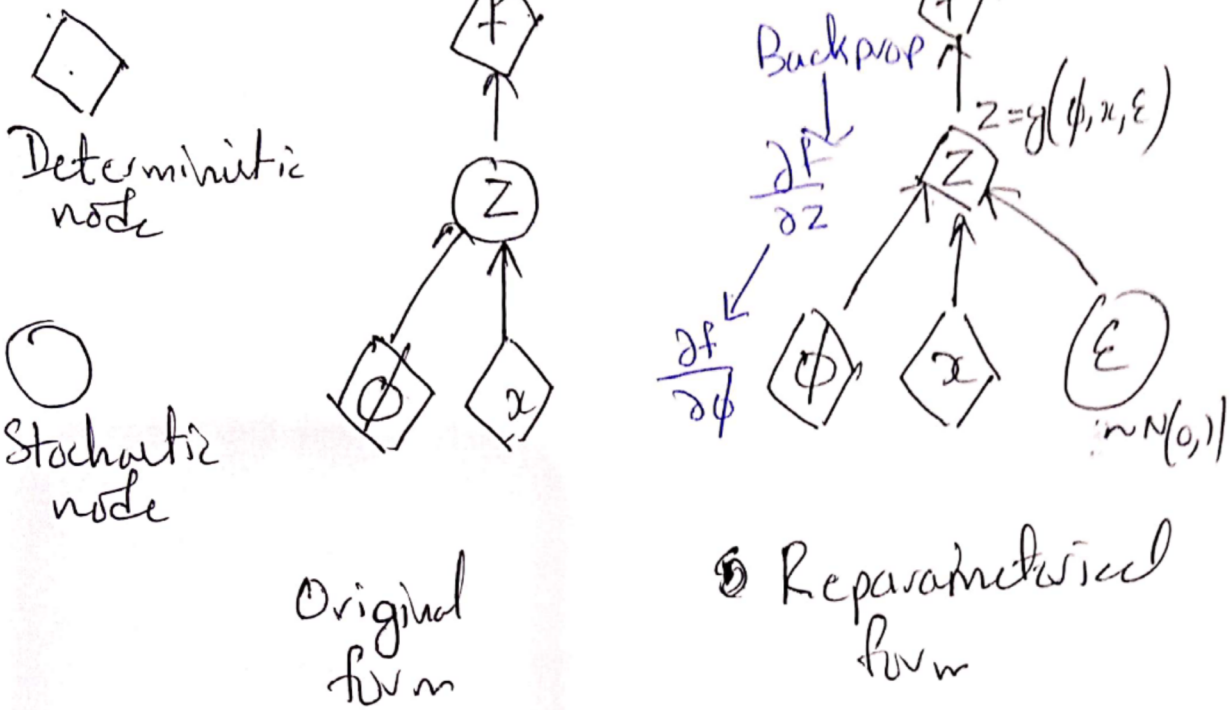



Back propagation requires deterministic nodes.

reparameterizing use $z = \mu + \sigma \odot \epsilon$ where $\epsilon \sim N(0,1)$

stochastic behaviour is introduced by $\epsilon$ (random constant).

$\epsilon$ is not occuring in the bottleneck sampling layer. It is distributed elsewhere.




Original form

Reparameterized form

We can directly backpropagate through z. ε is constant & it is re-parameterized elsewhere.