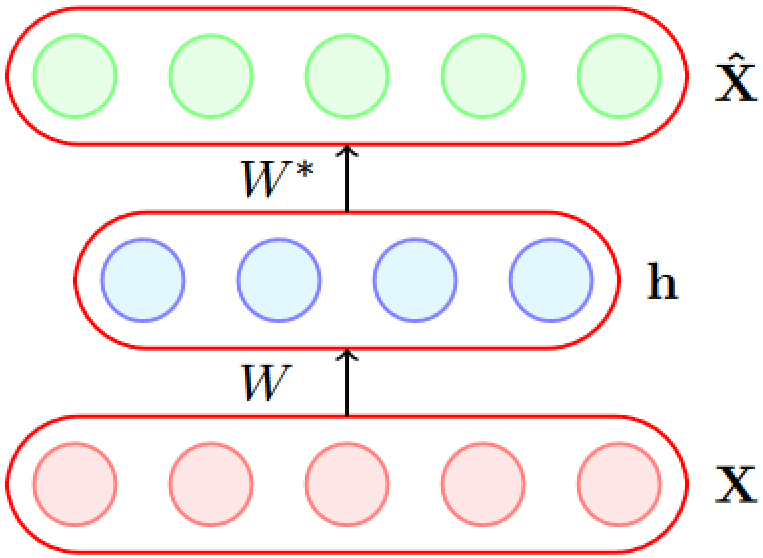
VAE continued

Biplab Banerjee

Auto-encoder re-visited



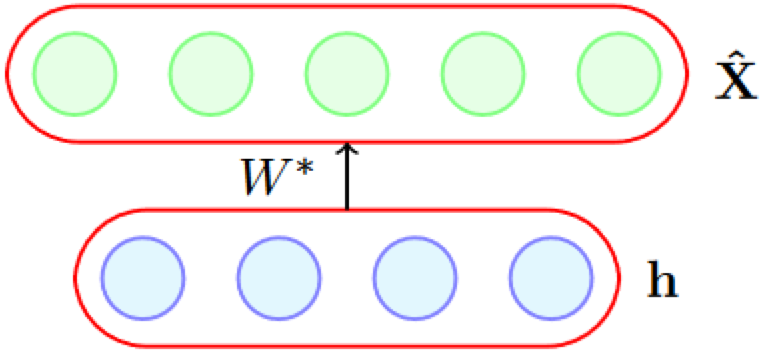
$$\mathbf{h} = g(W\mathbf{X} + \mathbf{b})$$

$$\hat{\mathbf{X}} = f(W^*\mathbf{h} + \mathbf{c})$$

- It contains two parts:
- ✓ Encoder
- ✓ Decoder
- Encoder is used for feature abstraction

- Can this be used as a generative model?
- ✓ Given *h*, can we generate meaningful data?

Auto-encoder re-visited



- *h* is usually high-dimensional
- Unless given, it is very difficult to sample a meaningful *h* without any prior knowledge

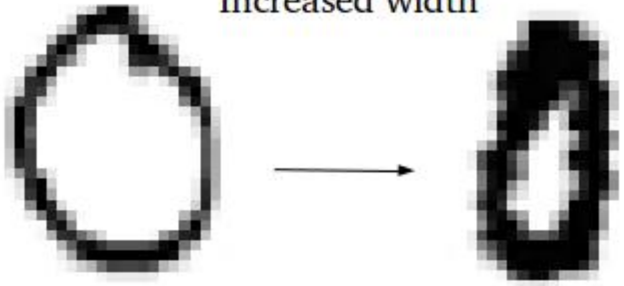
Ideally, we should only feed those values of h which are highly *likely*

In other words, we are interested in sampling from $P(h|X)$ so that we pick only those h's which have a high probability

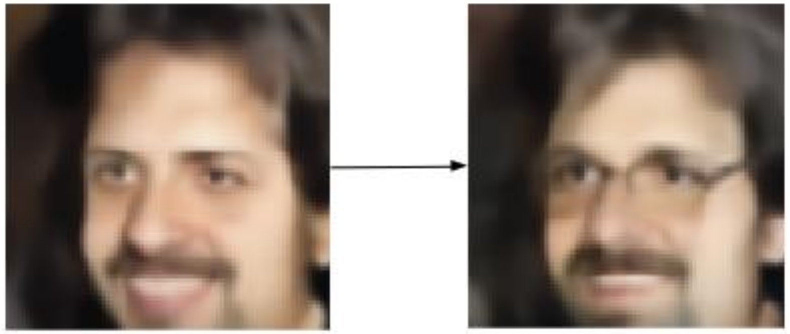
Probabilistic interpretation of AE?

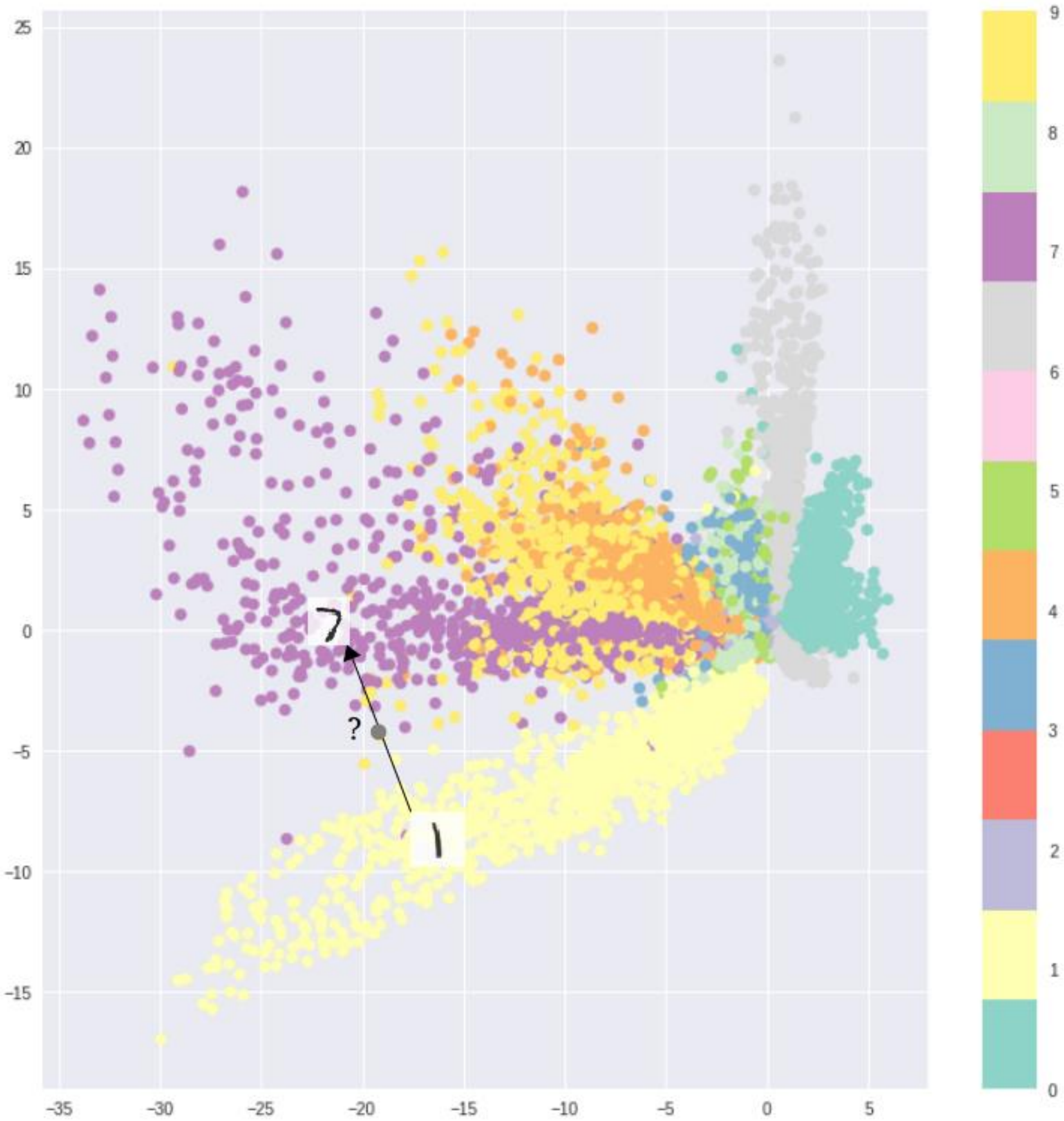
Some cases

Increased width



Glasses





Let’s summarize

- *Continuous latent space vs sparse latent space*
- We need to constrain the encoded space
- However, since data itself is complex and the encoder network has non-linear transformations, the distribution of the encoded space is super complex!
- Solution – approximate inference!

Goal of VAE

Let $\{X = x_i\}_{i=1}^{N}$ be the training data

We can think of X as a random variable in R^n

For example, X could be an image and the dimensions of X correspond to pixels of the image

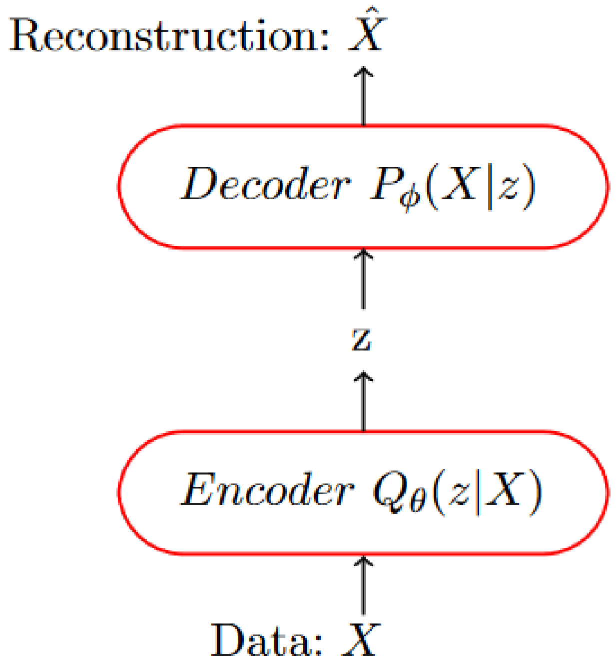
We are interested in learning an abstraction (i.e., given an X find the hidden representation z)

We are also interested in generation (*i.e.*, given a hidden representation generate an X)

In probabilistic terms we are interested in $P(z|X)$ and $P(X|z)$

Goal

Can be realized in terms of Neural networks



θ: the parameters of the encoder neural network

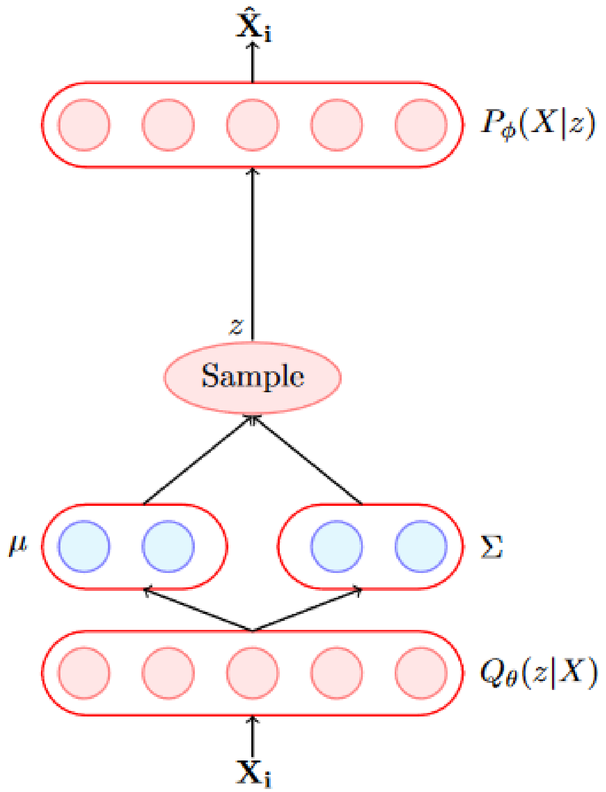
ϕ: the parameters of the decoder neural network

VAE

✓ The decoder should maximize
The likelihood of *P(X|z)*

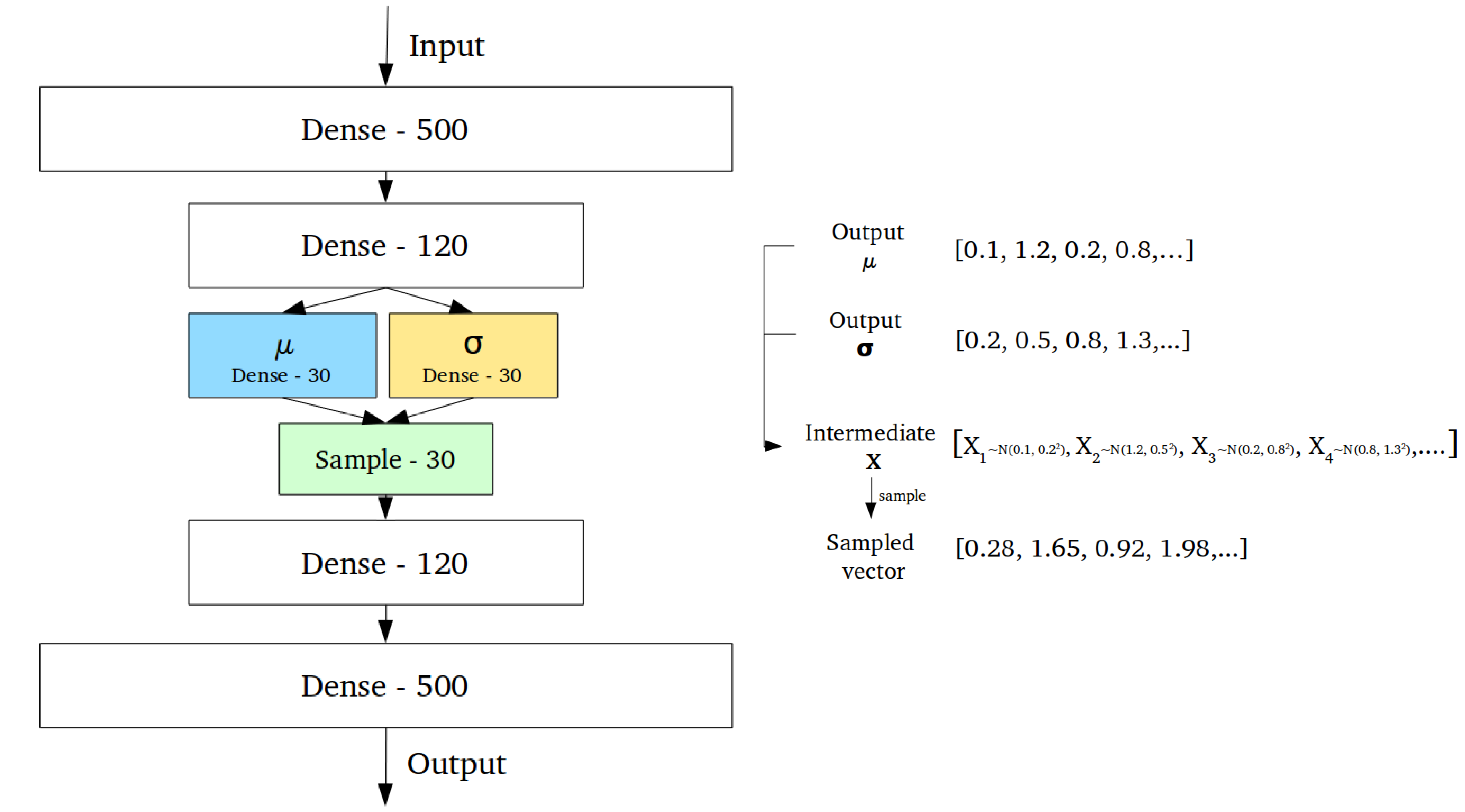
$$P(x_i) = \int P(z)P(x_i|z)dz$$
$$= -\mathbb{E}_{z\sim Q_\theta(z|x_i)}[\log P_\phi(x_i|z)]$$

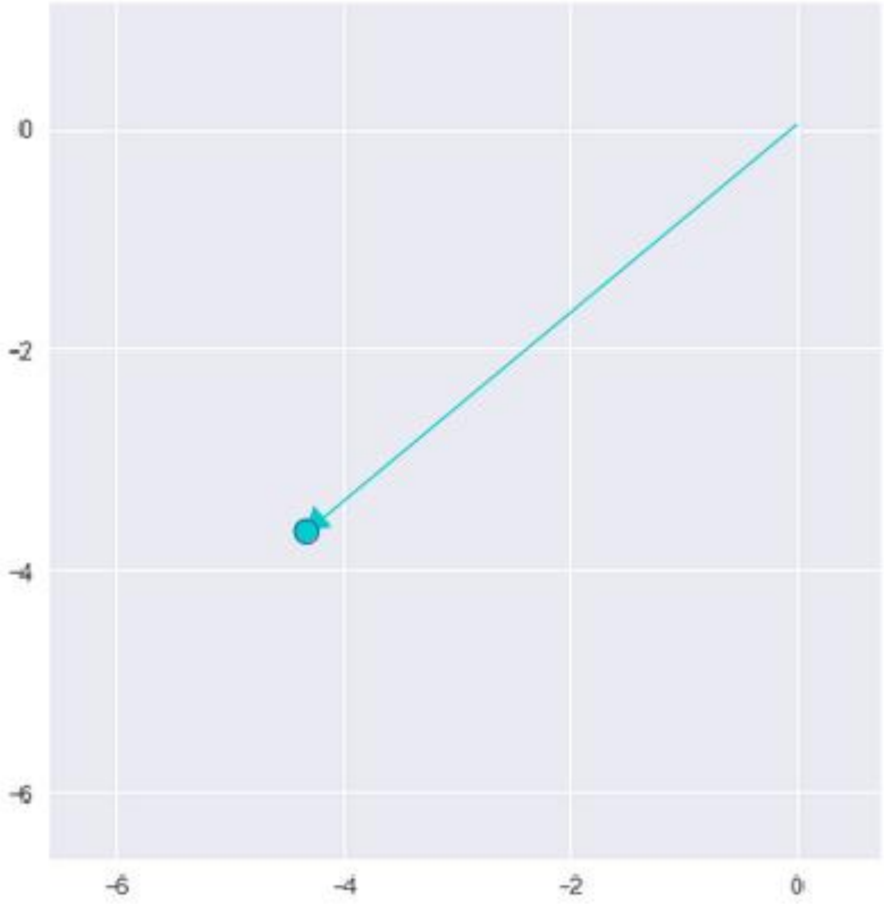
✓ The encoder should constrain
The *z* space to be some
Known continuous distribution



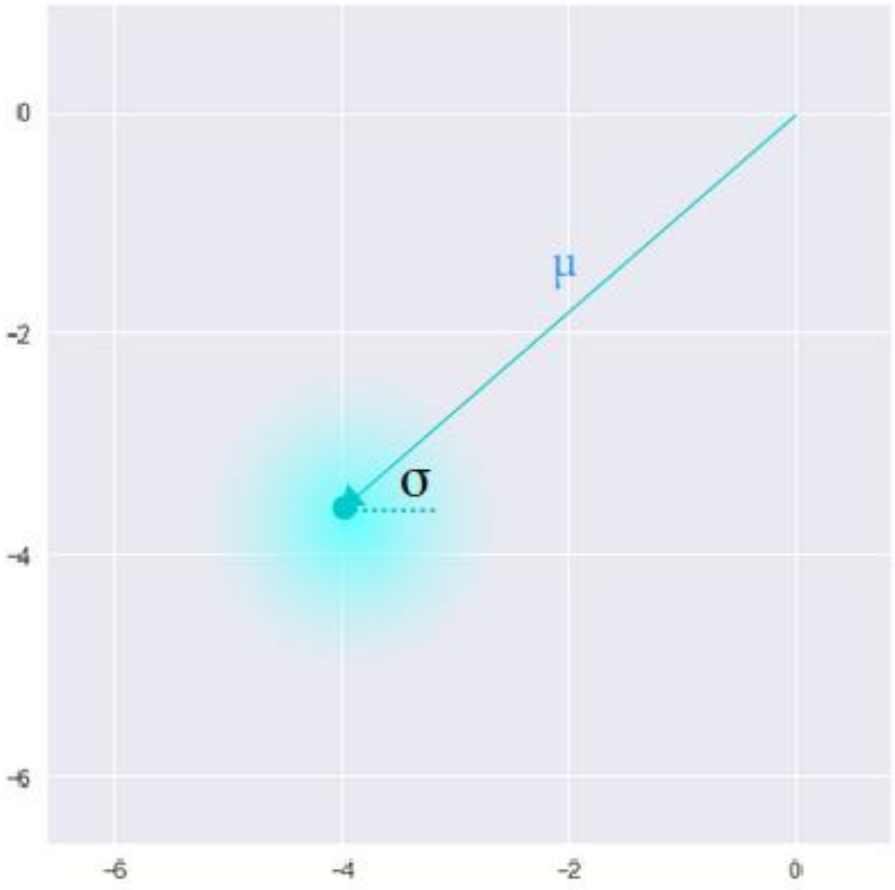
$$l_i(\theta,\phi) = -\mathbb{E}_{z\sim Q_\theta(z|x_i)}[\log P_\phi(x_i|z)] + KL(Q_\theta(z|x_i)||P(z))$$

Regularized AE? Like contrastive AE?


Input
Dense - 500
Dense - 120
μ
Dense - 30
σ
Dense - 30
Sample - 30
Dense - 120
Dense - 500
Output
Output
μ
[0.1, 1.2, 0.2, 0.8,…]
Output
σ
[0.2, 0.5, 0.8, 1.3,...]
Intermediate
X
$[X_{1\sim N(0.1,\, 0.2^2)}, X_{2\sim N(1.2,\, 0.5^2)}, X_{3\sim N(0.2,\, 0.8^2)}, X_{4\sim N(0.8,\, 1.3^2)},....]$
sample
Sampled
vector
[0.28, 1.65, 0.92, 1.98,...]



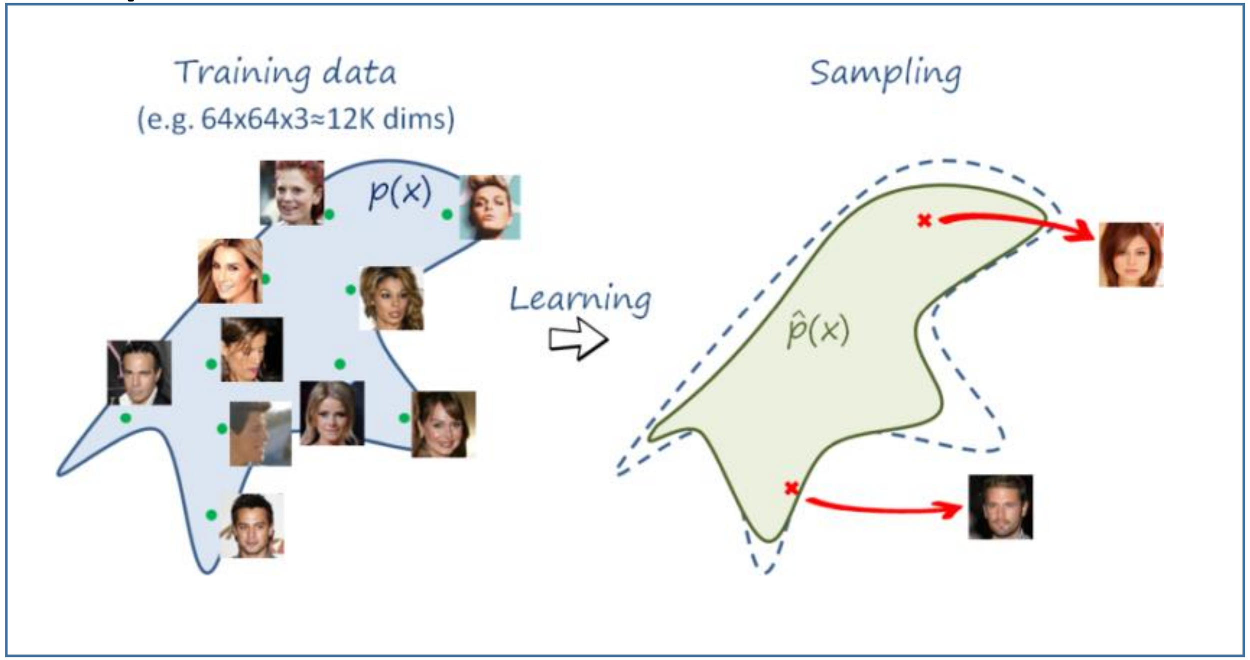
Standard Autoencoder
(direct encoding coordinates)

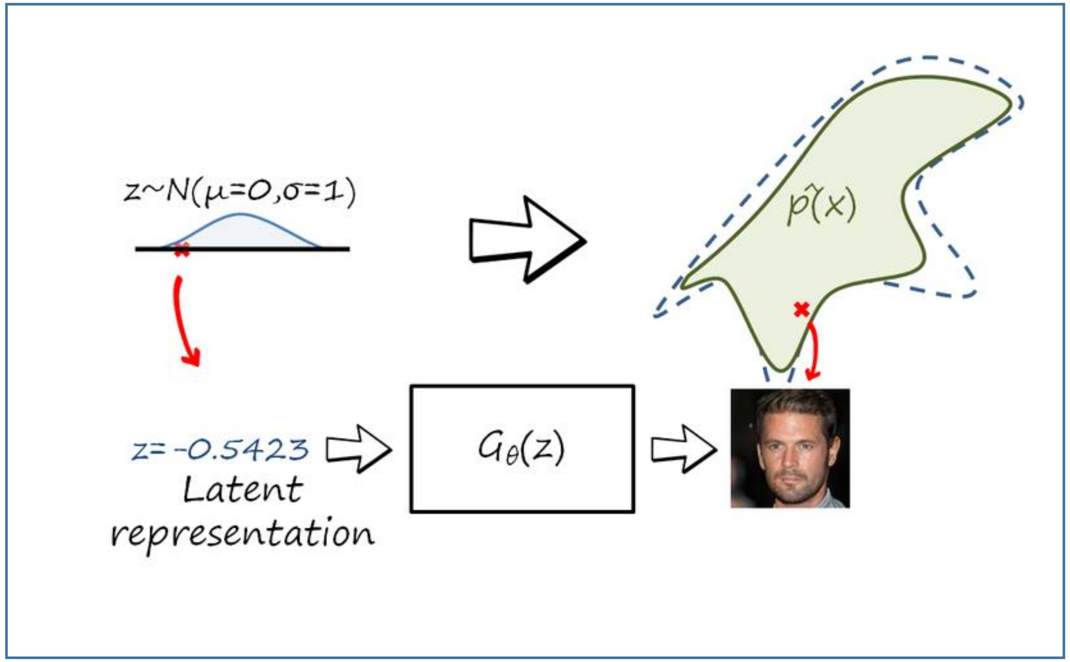


Variational Autoencoder
(μ and σ initialize a probability distribution)

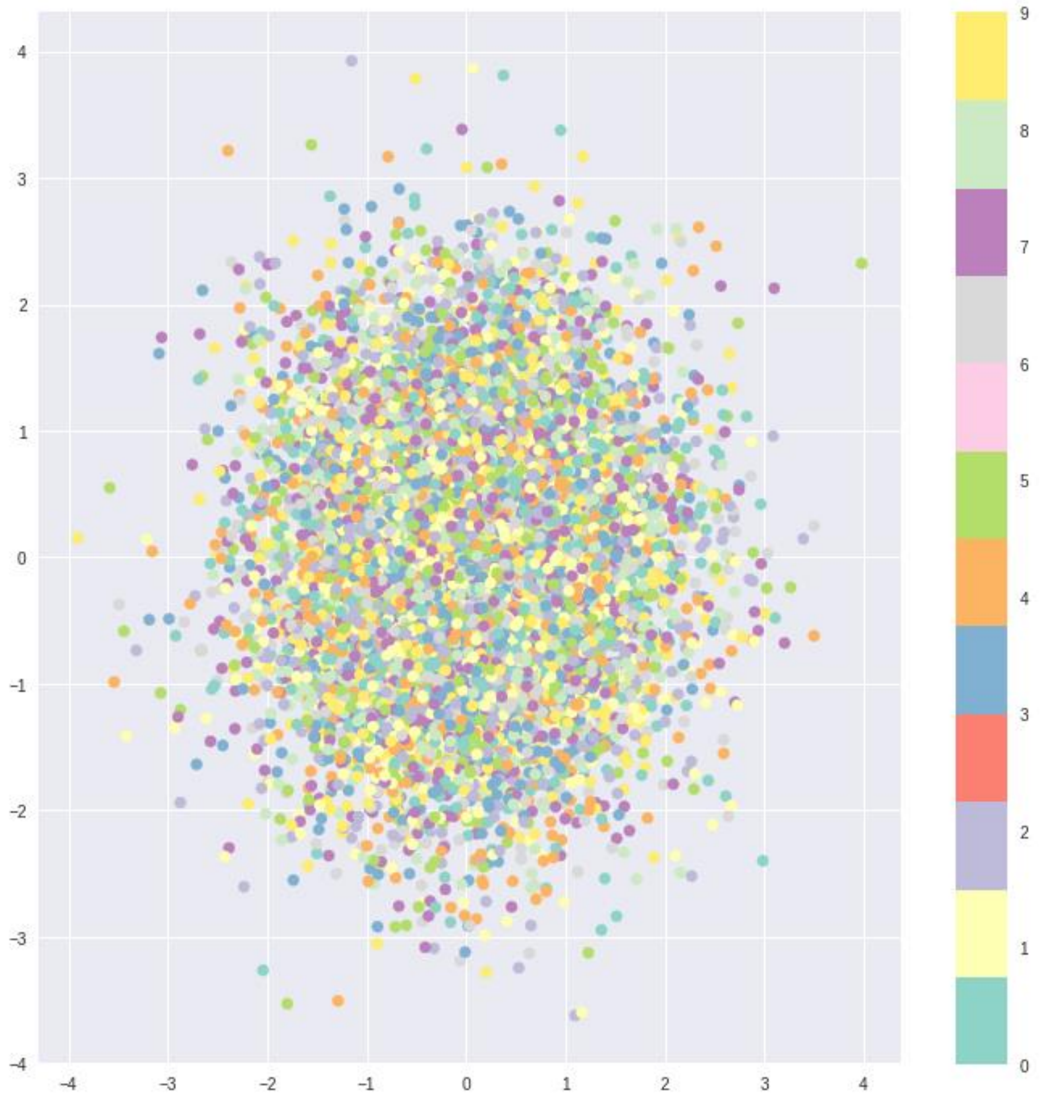
VAE

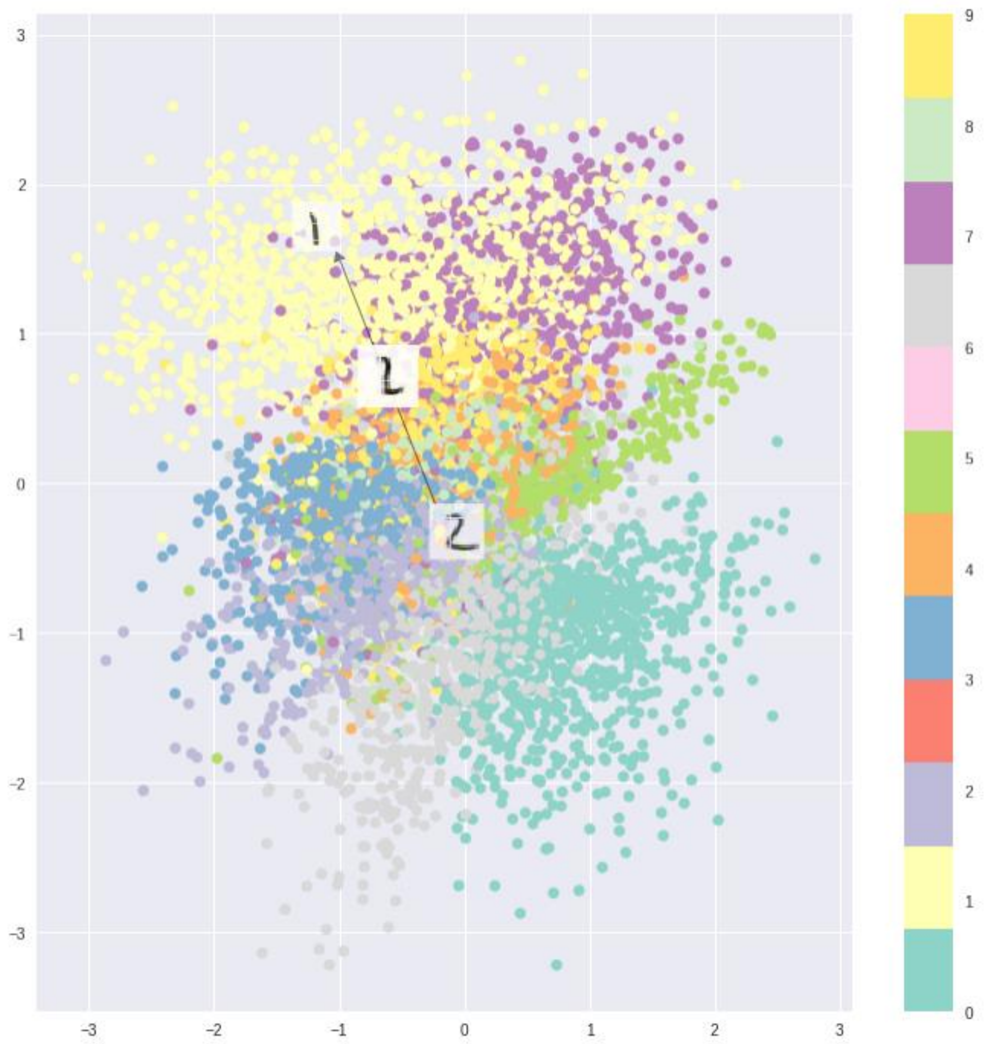
- For each data point, we want to estimate a distribution (or the parameter of a distribution) such that with high probability, a sample from this distribution will be able to reconstruct the original data point

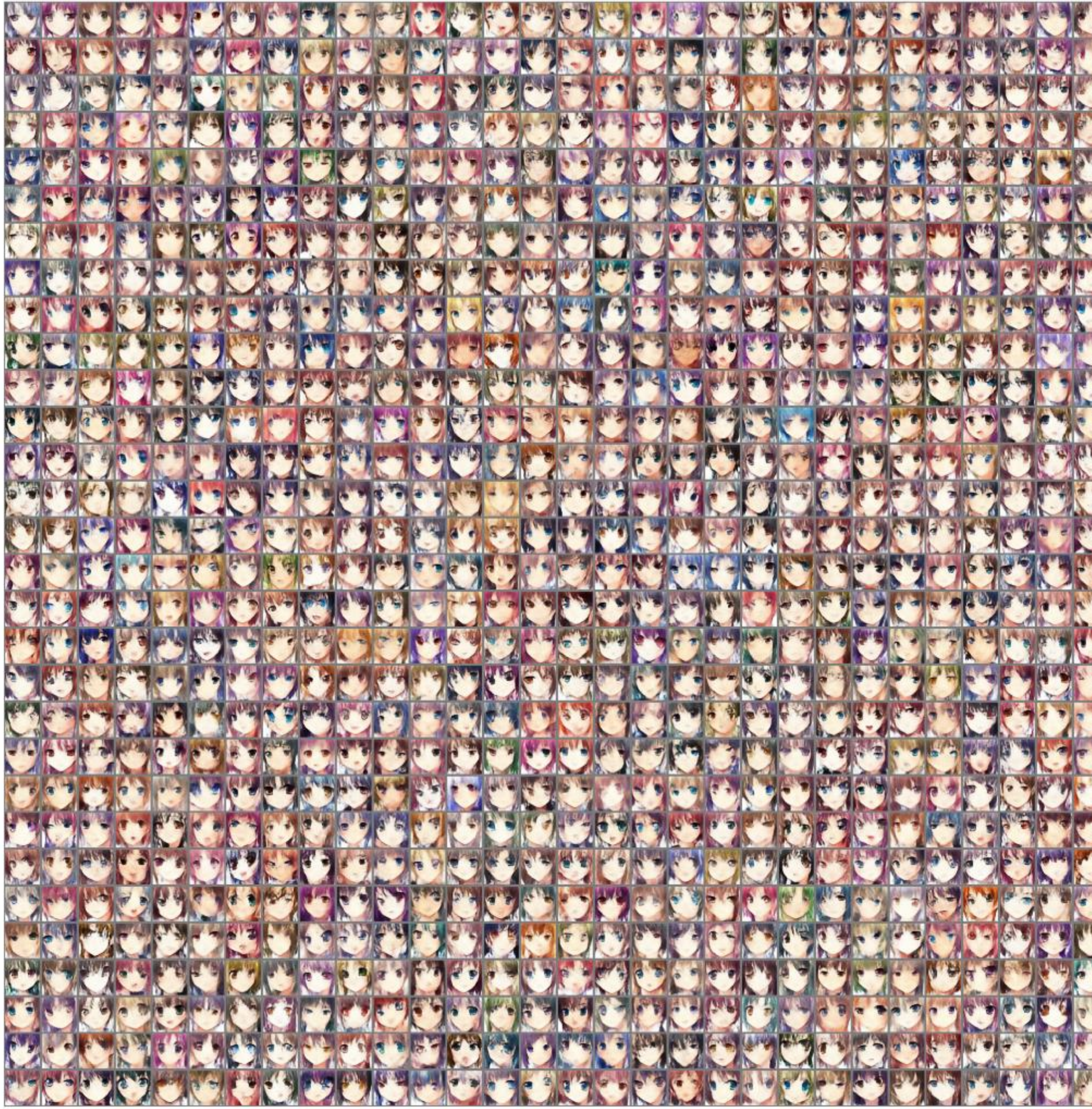




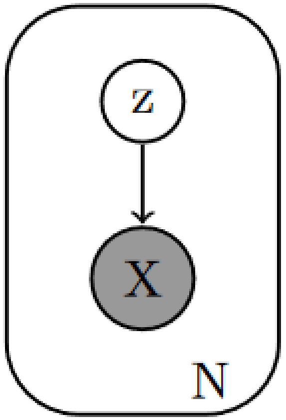
Effect of the loss terms







The variation inference perspective



- ✓ X is visible
- ✓ Z is latent or unobserved

The goal is for a given X, we want the most likely Z which offers the Best reconstruction of X

$$P(z|X) = \frac{P(X|z)P(z)}{P(X)}$$

$$P(X) = \int P(X|z)P(z)dz$$
$$= \int \int ... \int P(X|z_1, z_2, ..., z_n)P(z_1, z_2, ..., z_n)dz_1, ...dz_n$$

Solutions: Either MCMC or variational inference

Variational inference

- Since the posterior is intractable, we approximate P by a known distribution Q
- We assume that Q comes from a Gaussian and we can use the encoder network to estimate the distribution parameters
- Goal: We need Q to be as close as to P

$$minimize\ \ KL(Q_\theta(z|X)||P(z|X))$$

$$D[Q_\theta(z|X)||P(z|X)] = \int Q_\theta(z|X)\log Q_\theta(z|X)dz - \int Q_\theta(z|X)\log P(z|X)dz$$
$$= \mathbb{E}_{z\sim Q_\theta(z|X)}[\log Q_\theta(z|X) - \log P(z|X)]$$

$$D[Q_\theta(z|X)||P(z|X)] = \mathbb{E}_Q[\log Q_\theta(z|X) - \log P(X|z) - \log P(z) + \log P(X)]$$

$$\boxed{P(z|X) = \frac{P(X|z)P(z)}{P(X)}}$$

$$\mathbb{E}_Q[\log Q_\theta(z|X) - \log P(z)] - \mathbb{E}_Q[\log P(X|z)] + \log P(X)$$
$$D[Q_\theta(z|X)||p(z)] - \mathbb{E}_Q[\log P(X|z)] + \log P(X)$$

$$\boxed{\log p(X) = \mathbb{E}_Q[\log P(X|z)] - D[Q_\theta(z|X)||P(z)] + D[Q_\theta(z|X)||P(z|X)]}$$

Recall

- We want to maximize the likelihood of X given Z
- We want to minimize the KL div in the encoded space
- We need to maximize the blue term – variational lower bound

$$\mathbb{E}_Q[\log P(X|z)] - D[Q_\theta(z|X)||P(z)] <= \log P(X)$$

- Maximizing the lower bound means maximizing P(X)

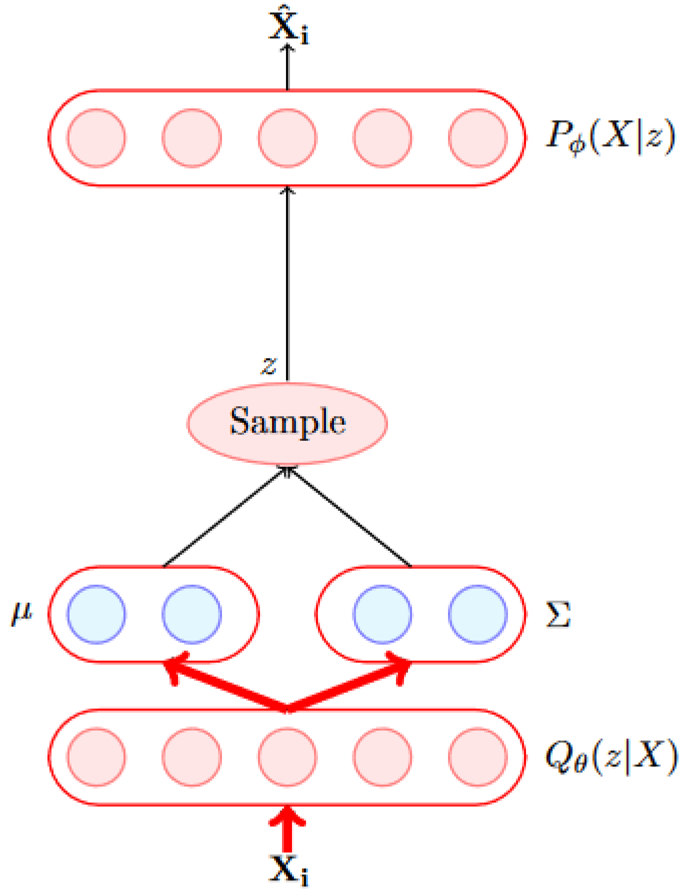
$$\boxed{maximize\ \ \mathbb{E}_Q[\log P(X|z)] - D[Q_\theta(z|X)||P(z)]}$$

Analysis of the loss

We are interested in expanding both the terms

$$D[\mathcal{N}(\mu(X), \Sigma(X))||\mathcal{N}(0, I)]$$
$$= \frac{1}{2}(tr(\Sigma(X)) + (\mu(X))^T[\mu(X)) - k - \log det(\Sigma(X))]$$

k is the dimensionality of the latent layer



Analysis of the loss

$$\sum_{i=1}^{n} \mathbb{E}_Q[\log P_\phi(X|z)]$$

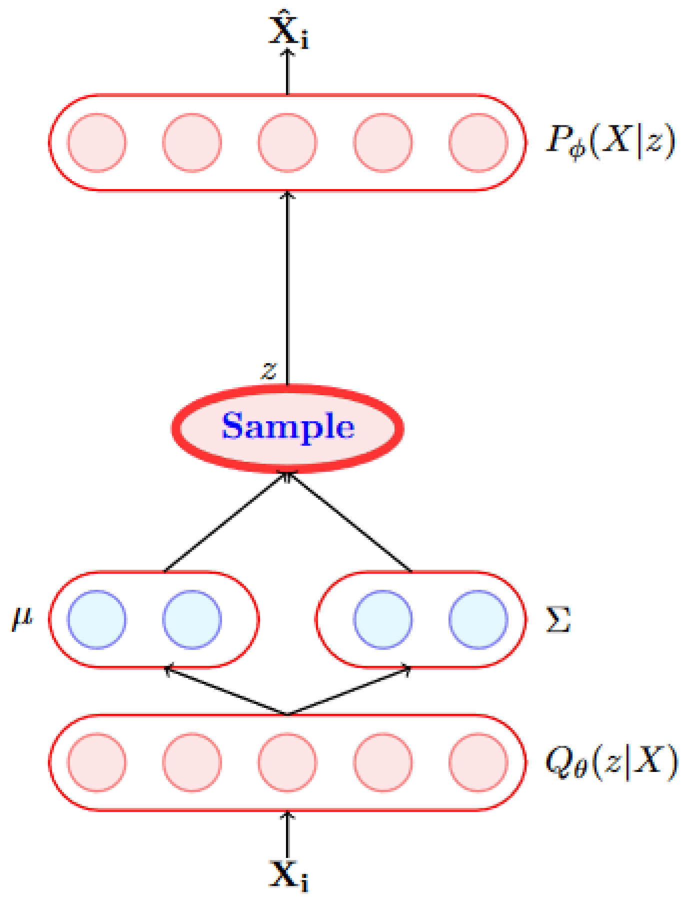
If we assume P(X|z) to be a Gaussian with mu(z) and I parameters,

$$\log P(X = X_i|z) = C - \frac{1}{2}||X_i - f_\phi(z)||^2$$

Total VAE loss

$$\underset{\theta,\phi}{minimize} \sum_{n=1}^{N} \left[\frac{1}{2}(tr(\Sigma(X_i)) + (\mu(X_i))^T[\mu(X_i)) - k - \log det(\Sigma(X_i))] + ||X_i - f_\phi(z)||^2\right]$$

Reparameterization trick



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

Some noise

Abstraction part – encoder only

After the model parameters are learned we feed a X to the encoder

By doing a forward pass using the learned parameters of the model we compute $\mu(X)$ and $\Sigma(X)$

We then sample a z from the distribution $\mu(X)$ and $\Sigma(X)$ or using the same reparameterization trick

In other words, once we have obtained $\mu(X)$ and $\Sigma(X)$, we first sample $\epsilon \sim \mathcal{N}(\mu(X), \Sigma(X))$ and then compute z

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

Generation part – decoder only

After the model parameters are learned we remove the encoder and feed a $z \sim \mathcal{N}(0, I)$ to the decoder

The decoder will then predict $f_\phi(z)$ and we can draw an $X \sim \mathcal{N}(f_\phi(z), I)$

Why would this work ?

Well, we had trained the model to minimize $D(Q_\theta(z|X)||p(z))$ where $p(z)$ was $\mathcal{N}(0, I)$

If the model is trained well then $Q_\theta(z|X)$ should also become $\mathcal{N}(0, I)$

Hence, if we feed $z \sim \mathcal{N}(0, I)$, it is almost as if we are feeding a $z \sim Q_\theta(z|X)$ and the decoder was indeed trained to produce a good $f_\phi(z)$ from such a z