

# Autoencoders and Variational Autoencoders

## Autoencoder

Input:  $x \in \mathbb{R}^d$

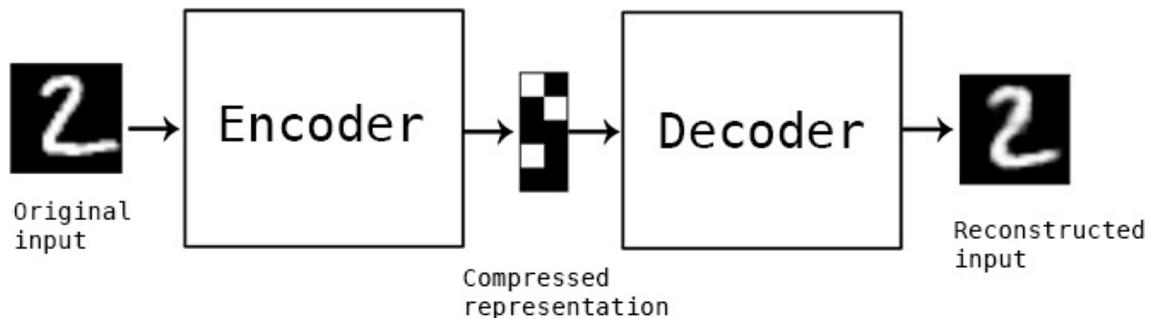
Encoder:  $f: \mathbb{R}^d \rightarrow \mathbb{R}^m$

Decoder:  $g: \mathbb{R}^m \rightarrow \mathbb{R}^d$

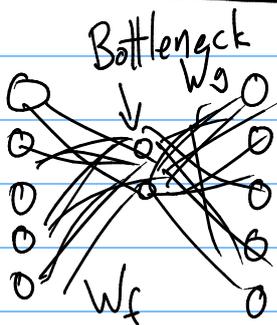
If  $m=d$ ,  
 $f(x)=x$   
 $g(x)=x$

Goal output  $g(f(x))=x$

Interesting if  $m < d$ . ( $d=1000, m=10$ )



## Linear Autoencoder

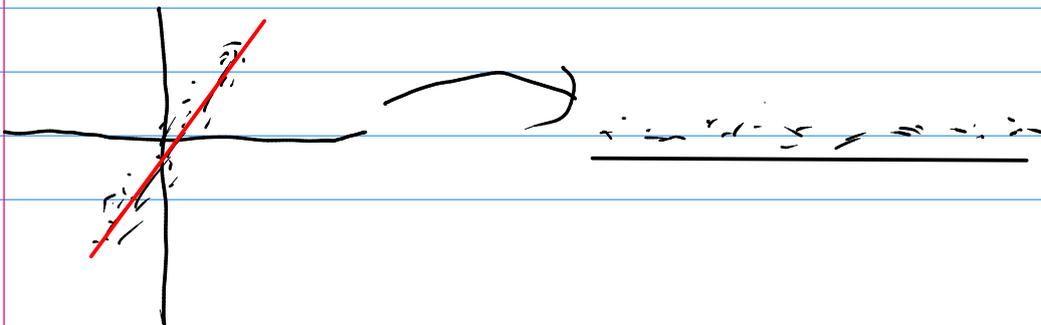


Output =  $W_g W_f x$

Objective:  $\min_{W_f, W_g} \sum_i \frac{1}{2} \|W_g W_f x_i - x_i\|_2^2$

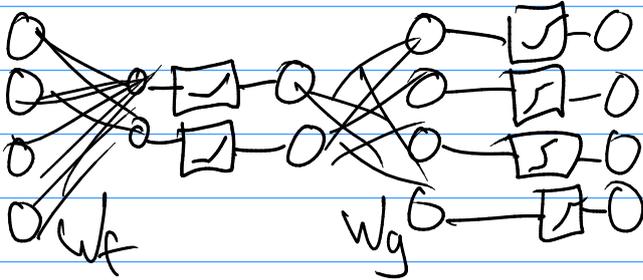
$W_f x$  is a compression of  $x$

## Principal Component Analysis (PCA)

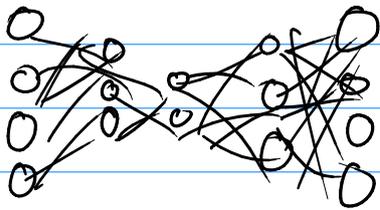


Nonlinear autoencoder  
 $f, g$  are nonlinear

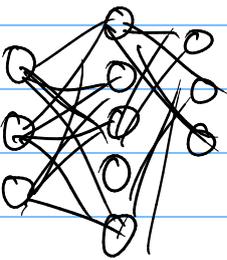
$$\min_{W_f, W_g} \sum \frac{1}{2} \|g(f(x_i)) - x_i\|_2^2$$



Deep autoencoders



Sparse Autoencoder

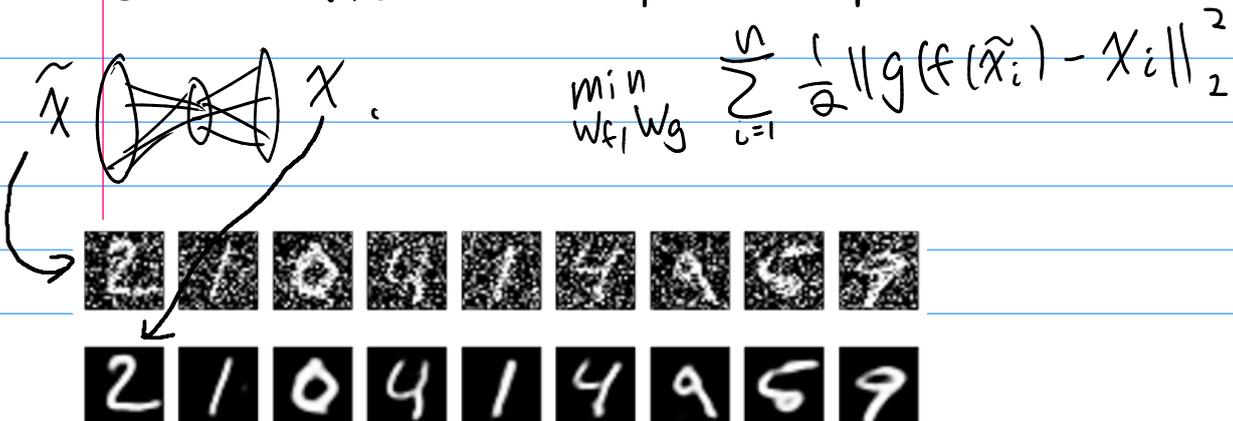


$$\min_{W_f, W_g} \sum \frac{1}{2} \|g(f(x_i)) - x_i\|_2^2 + \lambda \|f(x_i)\|_1$$

$f(x_i)$  will be sparse

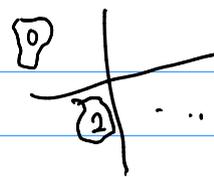
Denoising autoencoder

Given noised input  $\tilde{x}$ , produce  $x$  as output



$$\min_{W_f, W_g} \sum_{i=1}^n \frac{1}{2} \|g(f(\tilde{x}_i)) - x_i\|_2^2$$

# Variational Autoencoder



## Generative Modelling

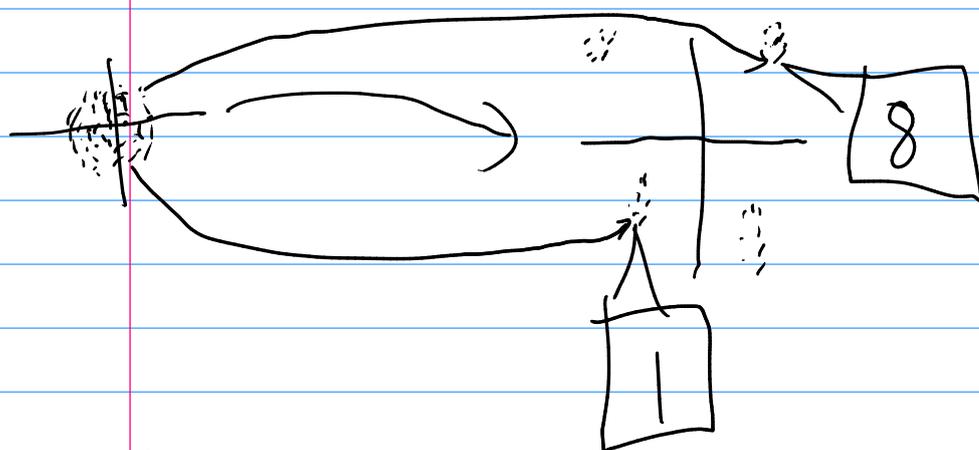
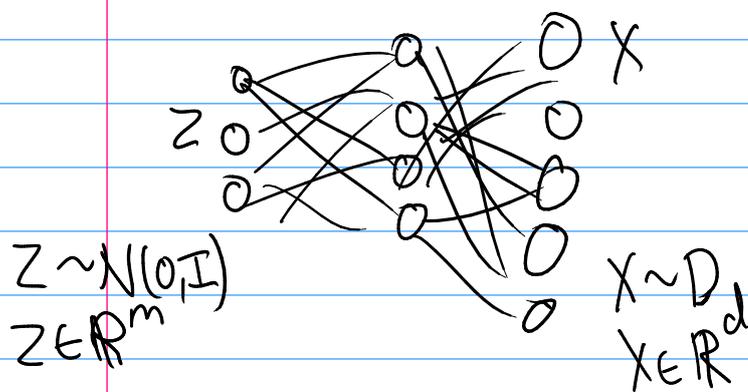
$X_1, \dots, X_n \sim D \Rightarrow$  Generate  $X_{n+1}, X_{n+2}, \dots$

$D$  may be complex (more so than a GMM)

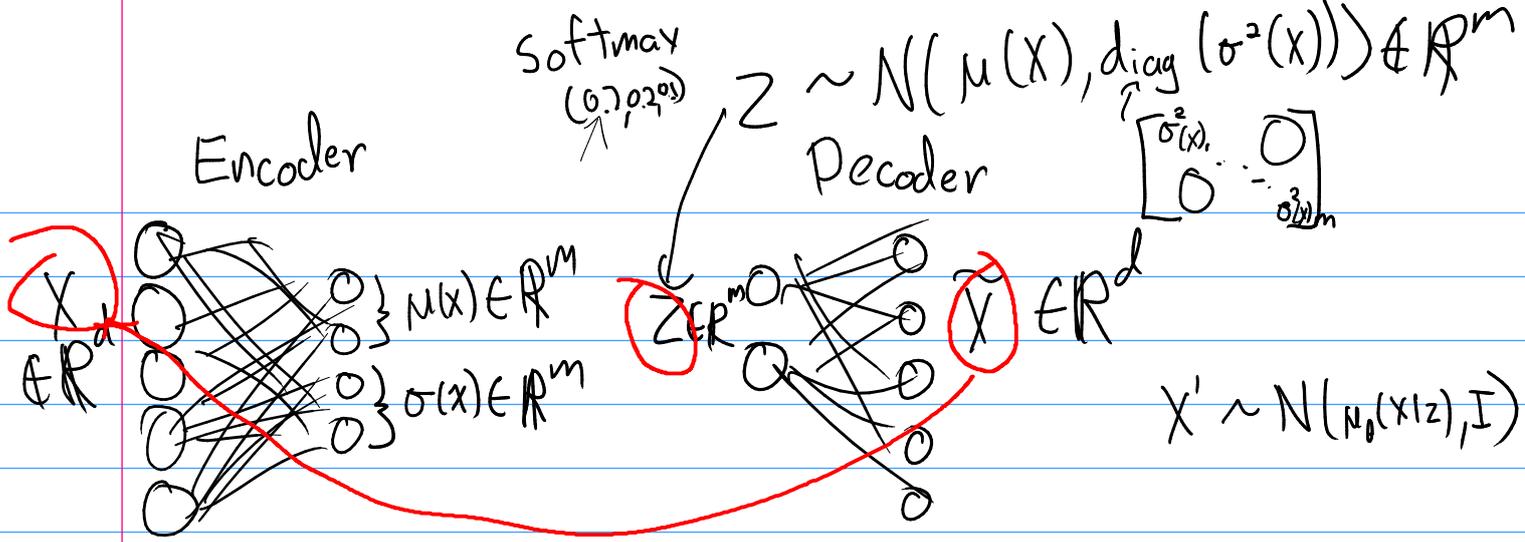
Say  $D$  is the distribution of Handwritten numbers or even Imagenet

Soln: Use NN to do the work!

Draw sample from  $N(0, I)$ , map it to a sample from  $D$

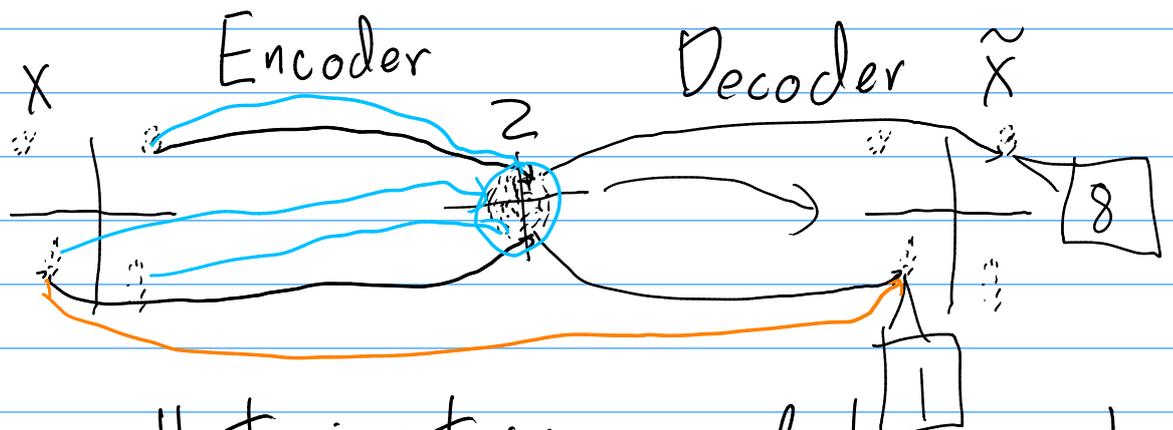


Actually: Use Variational autoencoder



Some notation:  $p_\theta(x)$  is density of decoder network.  
 $p_\theta(x|z)$  is a dist over decoder's outputs, given a fixed  $z$ .

$p_\theta(z)$  is dist over encoder's input (vs:  $N(0, I)$ )  
 $q_\phi(z|x)$



- Ensure that input image dist maps to  $N(0, I)$   
 Minimize  $KL(q_\phi(z|x) || p_\theta(z)) = KL(q_\phi(z|x) || N(0, I))$

- Inputs get encoded and then decoded back

Maximize  $E_{z \sim q_\phi(z|x)} [\log P_\theta(x|z)]$

Claim:  $\log P_\theta(x) \geq E_{z \sim q_\phi(z|x)} [\log P_\theta(x|z)] - KL(q_\phi(z|x) || N(0, I))$

Related to EM

Variational Inference

$$\log p_{\theta}(x) \geq \mathbb{E}_{z \sim q_{\phi}(z|x)} [\log p_{\theta}(x|z)] - \text{KL}(q_{\phi}(z|x) \| N(0, I))$$

Optimize: look at one term at a time

$$1. \text{KL}(q_{\phi}(z|x) \| N(0, I)) = \text{KL}(N(\mu_{\phi}(x), \text{diag}(\sigma_{\phi}^2(x))) \| N(0, I))$$

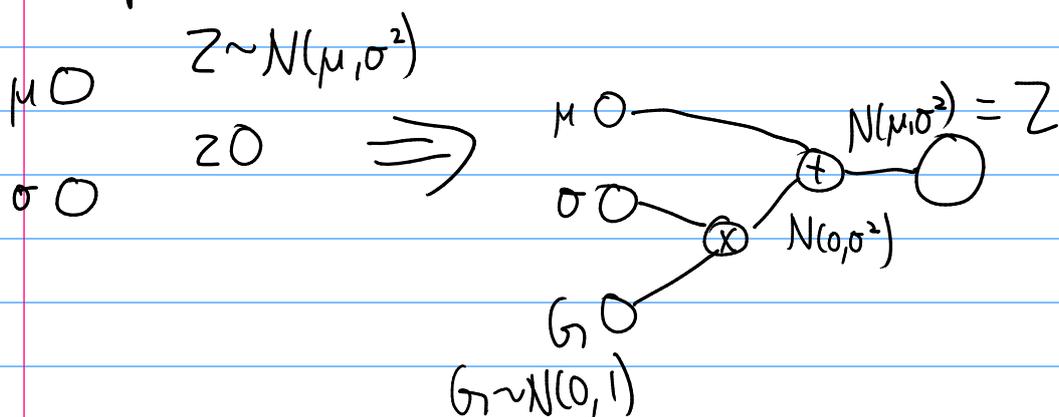
$$= \frac{1}{2} (\| \mu_{\phi}(x) \|_2^2 - m + \sum_{j=1}^m (\sigma_{\phi}^2(x)_j - \log(\sigma_{\phi}^2(x)_j)))$$

$\mu_{\phi}(x) = 0, \sigma_{\phi}^2(x)_j = 1 \forall j \quad \sum (1 - 0) = 0$

$$2. \mathbb{E}_{z \sim q_{\phi}(z|x)} [\log p_{\theta}(x|z)] \quad p_{\theta}(x|z) \sim N(\mu_{\theta}(x|z), I)$$

$$\mathbb{E}_{z \sim q_{\phi}(z|x)} [\|x - \mu_{\theta}(x|z)\|_2^2] - \cancel{d \log(\sqrt{2\pi})}$$

## Reparametrization Trick



6677814828	5165707677	2831365738	82085227700
9683960314	8554632162	2382773338	7549117144
3371369174	6153288438	2559479511	8962032824
8908691463	2868910041	1928532197	238437461
9233331386	5172075359	2736430263	5474199910
6448616666	6567491758	5770593745	6924048281
4526651849	1343973770	6743628527	7582167383
99271312823	4582970159	8490807066	7939279390
0461232088	6144872248	7436203601	4524340154
9754434851	2545608798	2720477400	2872876237

(a) 2-D latent space

(b) 5-D latent space

(c) 10-D latent space

(d) 20-D latent space

```

666666600000000000000000
44444222200000000000002
422222222333333330000002
44442222233333333555533
44442222333333333555537
44444888333333333555537
74444448883333333888887
74444448888888888888887
74444448888888888888887
74444448888888888888887
74444448888888888888887
74444448888888888888887
74444448888888888888887
74444448888888888888887
7777777111111111111111
7777777111111111111111
7777777111111111111111

```