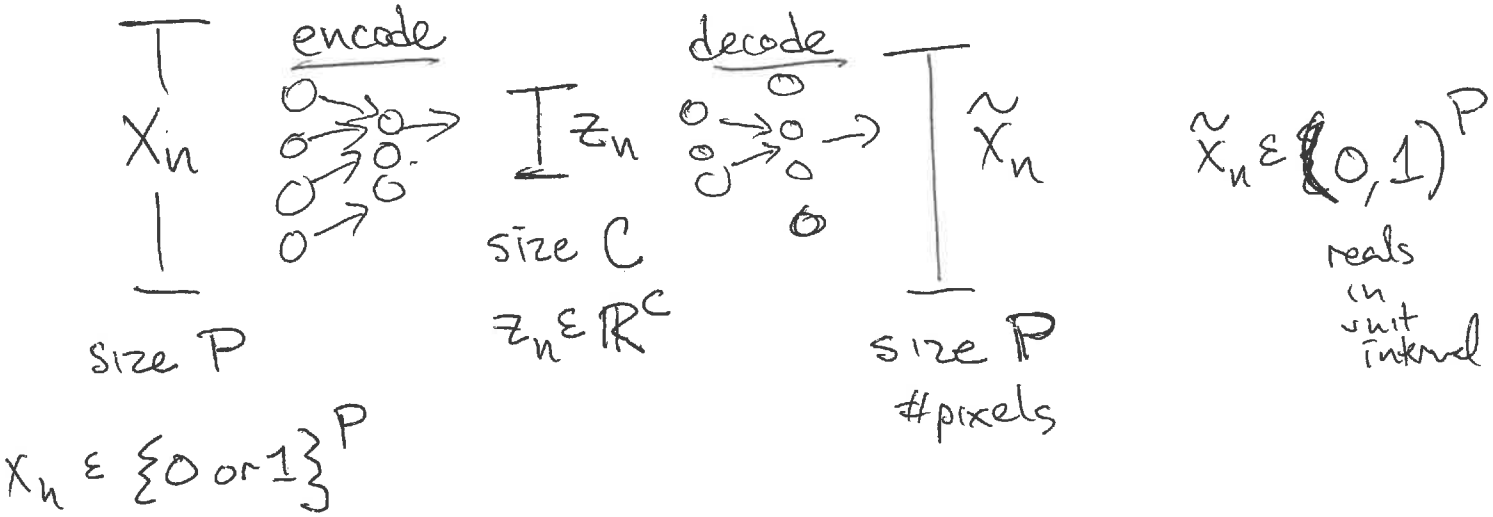
Autoencoders for Probabilistic Models & Variational AEs (VAEs)



Think of encoder/decoder as functions w/ trainable parameters

$$\text{encode}(x_n, \phi)$$

$$\text{decode}(z_n, \theta)$$

Write reconstruction as $\tilde{x}(x_n, \phi, \theta)$

Standard way to train: minimize "reconstruction" error as measured by cross entropy

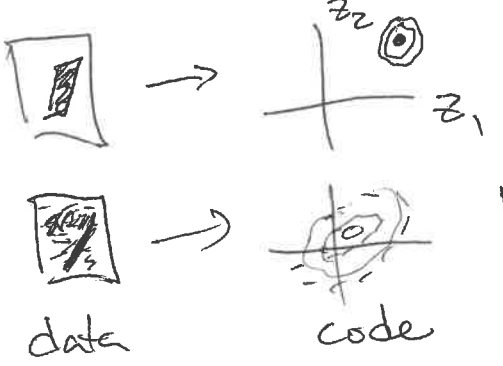
$$\min_{\phi,\theta} -\sum_{n=1}^{N}\sum_{p=1}^{P} x_{np} \log \tilde{x}_{np}(x_n, \phi, \theta) + (1-x_{np}) \log \tilde{x}_{np}(x_n, \phi, \theta)$$

has a "max lik" interp

$$\max_{\phi,\theta} \sum_{n=1}^{N} \log \text{BernPDF}(x_n \mid \tilde{x}(x_n, \phi, \theta))$$

Disadvantages of standard AE:

- not a way to understand uncertainty about how data gets encoded



- not regularized (potential for overfitting to training data)

Proposed Generative Model

prior $z_n \sim N(0, I_C)$

lik $x_n | z_n \sim \text{Bern}(\text{decode}(z_n, \theta))$

encoder is not part of generative model

Given this model, we want

- way to estimate θ
- way to estimate $p(z_n | x_n)$

$p(z_n | x_n)$ is hard! so let's assume a simpler form

$$q(z_n | x_n) = N(z_n | \text{encode}(x_n, \phi), \tau^2)$$

or

$$N(z_n | \text{encode}(x_n, \phi), \text{encode}(x_n, \Sigma))$$

Choice: has tradeoffs

simple ⟵ ⟶ flexible

Optimization problem:

$$\max_{\phi,\theta} \sum_{n=1}^{N} \mathbb{E}_{q_\phi(z_n|x_n)}\left[\log p(x_n, z_n|\theta) - \log q_\phi(z_n|x_n,\phi)\right]$$

this is our "standard" objective for VI

interpret: max lower bound on $\log p(x|\theta)$

Rewrite to simplify:

$$\max_{\phi,\theta} \sum_{n} \mathbb{E}_{q_\phi(z_n|x_n,\phi)}\left[\log \text{Bern}(x_n|z_n,\theta)\right] - \mathbb{E}_q\left[\log \frac{p(z_n)}{q_\phi(z_n|x_n,\phi)}\right]$$

negative expected reconstruction error $\quad - \; KL(q_\phi(z_n)\,|\,p(z_n))$ (prior)

$$\min_{\phi,\theta} \quad \text{reconstruction error} + KL(q\|\text{prior})$$

Two challenges:

how to calculate loss? MC integration by sampling

how to calculate gradient? Reparam trick