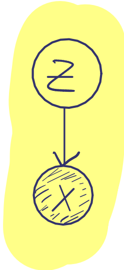
Variational Autoencoders & Lossy Neural Compression

Lecture 8 (23 June 2022); lecturer: Robert Bamler
more course materials online at https://robamler.github.io/teaching/compress22/

Recap from last lecture: Variational Inference (VI)



- latent variable model: $P(Z, X) = P(Z)\,P(X|Z)$
- goal: approximate the posterior: $P(Z|X{=}x) = \dfrac{P(Z)\,P(X{=}x|Z)}{\int P(Z{=}z)\,P(X{=}x|Z{=}z)\,dz}$

→ VI turns the inference problem into an optimization problem

→ variational distribution: $\boxed{Q_\phi(Z|X{=}x)}$ ← variational parameters (ϕ)

Evidence lower bound (ELBO):

- $\mathrm{ELBO}(\phi) = -\mathbb{E}_{Q_\phi(Z|X=x)}\big[\tilde{R}^{(z)}_{\text{net}}(x)\big]$ ← how we motivated it

Problem Set 8

$= \mathbb{E}_{Q_\phi(Z|X=x)}\big[\log P(Z, X{=}x) \underbrace{- \log Q_\phi(Z|X{=}x)}_{\to H[Q_\phi(Z|X=x)]}\big]$ ← most explicit formulation ("regularized MAP")

$= \mathbb{E}_{Q_\phi(Z|X=x)}\big[\log P(X{=}x|Z)\big] - D_{KL}\big(Q_\phi(Z|X{=}x)\,\|\,P(Z)\big)$ ← "regularized maximum likelihood"

$= \log P(X{=}x) - D_{KL}\big(Q_\phi(Z|X{=}x)\,\|\,P(Z|X{=}x)\big)$ ← connects VI to Bayesian inference

How to maximize the ELBO with stochastic gradient optimization:

- reparameterization gradients: $\nabla_\phi \mathbb{E}_{Q_\phi(Z|X=x)}\big[\ell(Z, \phi)\big] = \nabla_\phi \mathbb{E}_{\epsilon\sim Q_0}\big[\ell(g(\epsilon,\phi), \phi)\big]$

  $Z = g(\epsilon, \phi)$ where $\epsilon \sim Q_0$ ← fixed distribution

- score function gradients: $\nabla_\phi \mathbb{E}_{Q_\phi(Z|X=x)}\big[\ell(Z,\phi)\big] = \mathbb{E}_{Q_\phi(Z|X=x)}\big[(\nabla_\phi \log Q_\phi(Z|X{=}x)) \times \ell(Z,\phi)\big]$
  (= REINFORCE method)

Limitations so far:

TODAY: overcoming these limitations

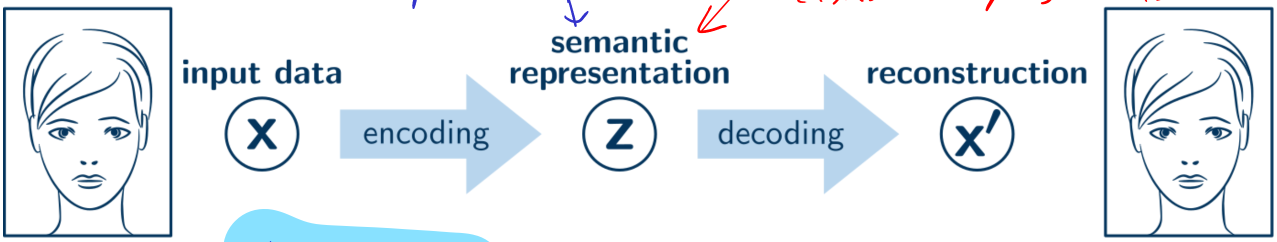
(i) the generative model P(Z, X) is fixed -- and therefore limited to simple models that we can come up with manually; and
→ Variational Expectation Maximization (Variational EM) → "learn the prob. gen. model P from training data"

(ii) for every concrete message x that we want to compress, we have to run an expensive optimization procedure to find the optimal variational parameters ϕ*.
→ Amortized variational inference "learn how to do inference"

Spoiler: Variational Autoencoders (VAEs) = amortized variational expectation maximization

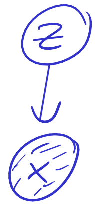
prior $P(Z)$

decorrelated representation
⇒ easier to compress than x



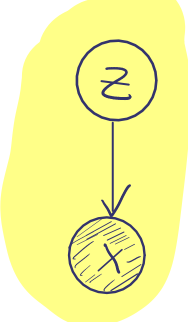
approximate Bayesian inference (VI)

probabilistic generative model



Variational Expectation Maximization: learning a latent variable model

[Beal & Ghahramani, Bayesian statistics, 2003]



Introduce free parameters into the probabilistic model P(Z, X):

$$P_\theta(Z, X) = P_\theta(Z)\, P_\theta(X \mid Z)$$

model parameters θ

$$\mathbb{E}_{x\sim\text{train set}}\left[\log P_\theta(X=x)\right]$$

$$= \mathbb{E}_{x\sim\text{train set}}\left[\log\left(\sum_z P_\theta(Z=z, X=x)\right)\right]$$

prohibitively expensive

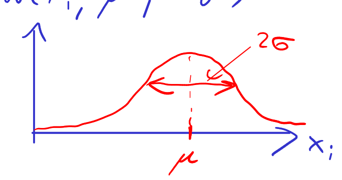
For example, the likelihood could be parameterized by a neural network with weights θ:

$$P_\theta(X \mid Z=z) = \underbrace{\mathcal{N}\big(\overbrace{g_\theta(z)}^{\mu},\ \sigma_0^2 I\big)}$$

"a normal distribution with mean $g_\theta(z)$ and variance σ_0^2 in each coordinate direction"

where g_θ : latent space (z) → data space (x) is a neural network with trainable weights θ

$\mathcal{N}(x_i; \mu, \sigma_0^2)$



Thus, the ELBO now depends both on the variational parameters ϕ and on the model parameters θ:

$$\mathrm{ELBO}(\theta, \phi, x) = -\mathbb{E} \quad R_{net}$$

$$= \mathbb{E}\quad\left[\log P - \log Q\right]$$

$$= \underbrace{\log P_\theta(X=x)}_{\text{optimal bitrate with model } P_\theta} \underbrace{- D_{KL}}_{\text{overhead due to VI}}$$

- maximize over both of these jointly (at training time)
- at compression time: keep θ fixed & maximize only over ϕ

minimize the expected bitrate of modified bits-back
net

$\Rightarrow$ maximize: $ELBO(\phi, \vartheta) = -\mathbb{E}_{Q_\phi(z|x=x)}\left[\tilde{R}^{(z,\vartheta)}_{net}(x)\right]$

variational params

model params

Alternative:

- store ϕ_x $\forall x \in$ train set on disk
- training loop:

for training_step in $\{1, 2, 3, \ldots, n\}$:

sample $x \sim$ train set

look up ϕ_x on disk

calculate $g_\vartheta = \nabla_\vartheta ELBO(\vartheta, \phi_x, x)$

$g_\phi = \nabla_{\phi_x} ELBO(\vartheta, \phi_x, x)$ $\Leftarrow$

update $\vartheta \leftarrow \vartheta + \varrho g_\vartheta$

$\phi \leftarrow \phi + \varrho g_\phi$ $\Leftarrow$

store ϕ_x back to disk

Data compression with learnt latent variable models (try 1: without amortization):

1) When designing the compression method:
   - collect large (unlabeled) data set of training samples (e.g., a large collection of images)
   - come up with a model architecture for the generative model P_θ (that still has free parameters)
   - train the model by maximizing the ELBO jointly over both θ and ϕ.

in detail: $\theta^*, \{\phi_x^*\} := \arg\max_{\theta, \{\phi_x\}_x} \mathbb{E}_{x \sim \text{trainset}}\left[\text{ELBO}(\theta, \phi_x, x)\right]$

ϕ_x are the variational params for training point x

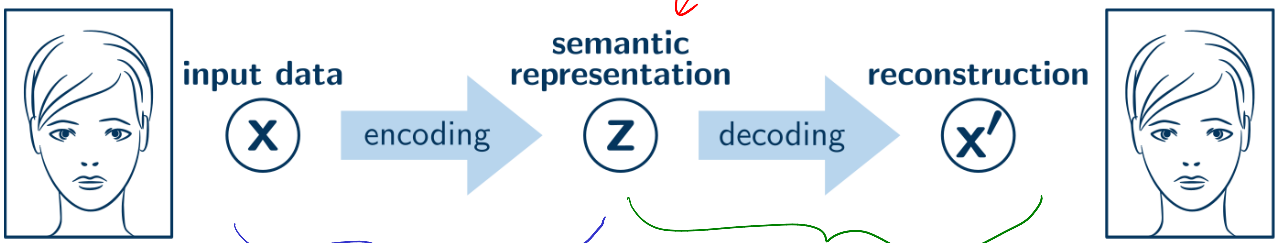
for training step $\in \{1, 2, \ldots, N\}$:
  find $\phi_x^* := \arg\max \text{ELBO}(\theta, \phi_x)$
  set $\hat{g} := \nabla_\theta \text{ELBO}(\ldots)$
  update $\theta \leftarrow \theta + \rho \hat{g}$

   - throw away $\{\phi_x^*\}$ and share θ* between sender and receiver

2) When compressing some given data x (i.e., on the sender side):
   - perform variational inference, i.e., maximize ELBO(θ*, ϕ, x) over ϕ but keep θ* fixed at the agreed-upon values.
   - use probabilistic generative model P_{θ^*}(Z, X) and the resulting variational distribution Q_{ϕ^*}(Z | X=x) to compress x.

3) When decompressing data (i.e., on the receiver side):
   - needs the exact same probabilistic generative model P_{θ^*}(Z, X) that the sender used for comprestion.
   - if the data was compressed with bits-back coding, then the receiver also needs to perform variational inference once it has reconstructed x (i.e., maximize ELBO(θ, ϕ) over ϕ but keep θ^*!

compress z using prior $P_{\theta^*}(Z)$



$Q_{\phi^*}(Z \mid X=x)$

where $\phi^* := \underbrace{\arg\max_\phi}_{\text{expensive operation}} \text{ELBO}(\phi, \theta^*, x)$

$P_{\theta^*}(X \mid Z)$

where θ^* is known at (de-)compression time

Amortized Variational Inference: learn how to do inference

Variational inference maps data x to a variational distribution Q_{ϕ^*}(Z | X=x):

Idea: learn this mapping from x to Q_{ϕ^*}(Z | X=x):

for example: $Q_\lambda(Z) = \mathcal{N}(\mu, \text{diag}(\sigma^2))$

$\lambda = (\mu, \sigma^2)$

   - rename the parameteres of Q_λ(Z) from ϕ to λ

   - rather than optimizing over λ, learn a function f that maps x to λ (and that is parameterized by some neural network weights ϕ:

$\lambda = f_\phi(x)$ where f_ϕ is a neural network with weights ϕ

$$\boxed{Q_\phi(Z \mid X=x) := Q_\lambda(Z) \quad \text{where } \lambda = f_\phi(x)}$$

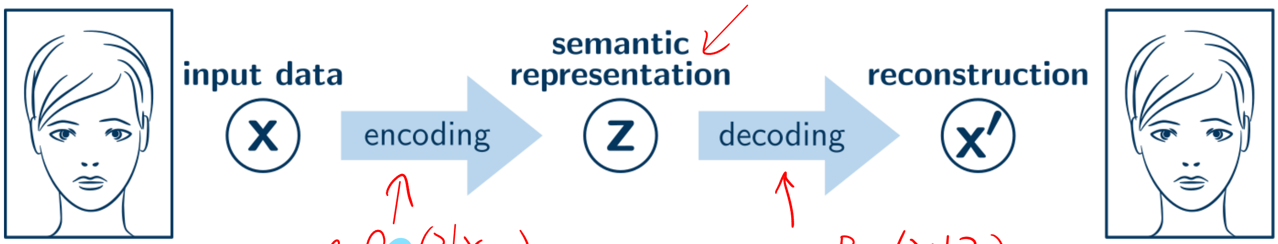
⇒ variational parameters ϕ are now shared between all data points x

$\text{ELBO}(\phi, \theta, x) = \mathbb{E}_{Q_\phi(Z \mid X=x)}\left[\log P(Z, X=x) - \log Q_\phi(Z \mid X=x)\right]$

Combining amortized inference with variational expectation maximization results in:

Variational Autoencoers (VAEs): [Kingma and Welling, 2015]

prior $P_{\theta^*}(z)$



use $Q_{\phi^*}(z|X=x)$ use $P_{\theta^*}(X|z)$

both ϕ^* & θ^* are fixed once the model is deployed

training objective: $ELBO(\theta, \phi) = \mathbb{E}_{x \sim P_{train}(x)}\Big[\underbrace{\mathbb{E}_{Q_\phi(z|X=x)}\big[\log P_\theta(z, X=x) - \log Q_\phi(z|X=x)\big]}_{= \mathbb{E}_{Q_\phi(z|X=x)}[\log P_\theta(X=x|z)] - D_{KL}(Q_\phi(z|X=x) \,\|\, P_\theta(z))}\Big]$

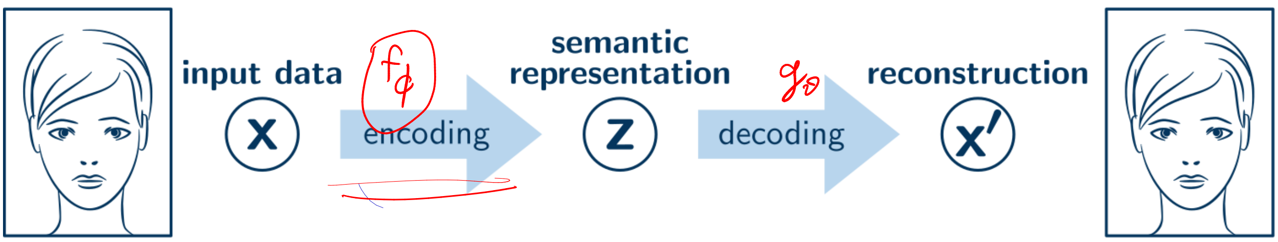
Problem Set: implement a variational autoencoder for simple images (MNIST)

Note: Variational expectation maximization (EM) is not limited to VAEs. Even without amortized inference, variational EM is a very useful algorithm that is very simple and allows you to treat some model parameters (Z) probabilistically while using point estimates for others (θ).

Lossy Compression with VAEs -- a Brief Pragmatic Introduction

(more details: Problem 10.1 on Problem Set 10 and lectures 11 and 12)

- VAEs are popular models for lossy data compression
- here's a brief overview; we'll dive deeper into lossy compression starting next week



Simplest (yet surprisingly powerful) method [Ballé et al., 2016]:

part of model params θ

- generative model: Prior must have a learnable variance: e.g., $P_\theta(z) = \mathcal{N}(0, \sigma^2 I)$ (for each coordinate)
- variational distribution: $Q_\phi(z|X=x) = \text{Uniform}\big([f_\phi(x) - \frac{1}{2}, f_\phi(x) + \frac{1}{2}]\big)$
- encoding: • given input x, calculate $f_\phi(x)$, then round each component to nearest integer $\rightarrow z$
  • encode z using $P(z)$
- decoding: • decode losslessly z using Prior $P(z)$ ← z needs to be discrete rand. var.
  • reconstruction $\hat{x} := \arg\max_x P_\theta(X|Z=z)$

Problem: we can only compress z losslessly if it comes from a discrete distribution.

Approximation:

• assume at compression time that $z \in \mathbb{Z}^k$ ⇒ get z by rounding each coordinate of $f_\phi(x)$

• during training: approximate rounding of $f_\phi(x)$ by adding uniform random noise $[-\frac{1}{2}, \frac{1}{2}]^k$