

Lecture 9: Variational Autoencoders

Robert Bamler • Summer Term of 2023

These slides are part of the course "Data Compression With and Without Deep Probabilistic Models" taught at University of Tübingen. More course materials-including video recordings, lecture notes, and problem sets with solutions—are publicly available at https://robamler.github.io/teaching/compress23/.

Recall: Variational Inference

UNIVERSITÄT TÜBINGEN I

- Idea:
 - ▶ approximate the (inaccessible) true posterior $P(\mathbf{Z} | \mathbf{X} = \mathbf{x})$ with a variational distribution $Q_{\phi}(\mathbf{Z})$.
 - Find the best approximation $\phi^* := \arg \max_{\phi} ELBO(\phi, \mathbf{x})$.

• Evidence Lower Bound: $|ELBO(\phi, \mathbf{x}) = \mathbb{E}_{Q_{\phi}(\mathbf{Z})} [\log P(\mathbf{Z}, \mathbf{X} = \mathbf{x}) - \log Q_{\phi}(\mathbf{Z})]$

- negative expected net bit rate of bits-back coding: ELBO(ϕ , **x**) = $-\mathbb{E}_{\mathbf{s}}[R_{\phi}^{\text{net}}(\mathbf{x} | \mathbf{s})]$
- bound on the evidence: ELBO(ϕ , \mathbf{x}) = log $P(\mathbf{X} = \mathbf{x}) D_{\text{KL}}(Q_{\phi}(\mathbf{Z}) || P(\mathbf{Z} | \mathbf{X} = \mathbf{x})) \le \log P(\mathbf{X} = \mathbf{x})$
- ▶ regularized maximum likelihood: ELBO(ϕ , **x**) = $\mathbb{E}_{Q_{\phi}(\mathbf{Z})} \left[\log P(\mathbf{X} = \mathbf{x} | \mathbf{Z}) \right] D_{\mathsf{KL}} \left(Q_{\phi}(\mathbf{Z}) \| P(\mathbf{Z}) \right)$
- **today:** rate/distortion-tradeoff: $|\mathsf{ELBO}_{\beta}(\phi, \mathbf{x}) = \mathbb{E}_{Q_{\phi}(\mathbf{Z})} [\log P(\mathbf{X} = \mathbf{x} | \mathbf{Z})] \beta D_{\mathsf{KL}} (Q_{\phi}(\mathbf{Z}) | P(\mathbf{Z}))$ (actually, next neet ())
- Problems:
 - What's the generative model $P(\mathbf{Z}, \mathbf{X})$? \rightarrow variational expectation maximization
 - Expensive "arg max_{ϕ}" for each message **x** in both encoder & decoder. \longrightarrow amortized inference

Part 1: Learning the Generative Model

- **Goal:** learn optimal parameters θ^* of the generative model $P_{\theta}(\mathbf{Z}, \mathbf{X}) = P_{\theta}(\mathbf{Z}) P_{\theta}(\mathbf{X} | \mathbf{Z})$.
 - ▶ Thus, the ELBO now depends on θ , i.e., ELBO $(\theta, \phi, \mathbf{x}) = Q_{\phi}(\mathbf{Z}) \left[\log P_{\theta}(\mathbf{Z}, \mathbf{X} = \mathbf{x}) \log Q_{\phi}(\mathbf{Z}) \right]$
 - **Example:** data $\mathbf{X} = (X_i)_i$ are binarized images, i.e., each X_i is a pixel value $\in \{0, 1\}$.
 - \rightarrow Prior is fixed: $P(\mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, I)$ (standard normal distribution)

Robert Bamler - Lecture 9 of the course "Data Compression With and Without Deep Probabilistic Models" - Summer Term of 2023 - more course materials at https://robamler.github.io/teaching/compress23/

UNIVERSITÄT TÜBINGEN

| 1

Distinguish:

- global parameters θ^* ("model parameters"):
 - \rightarrow specify the generative model $P_{\theta^*}(\mathbf{Z}, \mathbf{X})$
 - \rightarrow same for all data points $\mathbf{x} \Longrightarrow$ known to both sender & receiver
- *local* parameters ϕ^* ("variational parameters"):
 - \rightarrow specify an approximation $Q_{\phi^*}(\mathbf{Z})$ to the posterior $P_{\theta^*}(\mathbf{Z} | \mathbf{X} = \mathbf{x})$ for a specific data point \mathbf{x} ightarrow different for each data point x \Longrightarrow not available to the receiver until it has decoded x

ariational Expectation Maximization 1. In order to develop a new compression method: • learn optimal parameters θ^* of the generative model $P_{\theta}(Z, X)$: $\theta^{\dagger}(\theta_{1} \geq) = \pi_{1} m_{1} \times E(D \circ (\theta_{1}, \theta_{2}))$ $S^{\ast} \in argumax \in E_{x-t-inty} of [ELB \circ (S, \Phi^{\dagger}(\theta_{1}, Z), X]]$ $= \pi_{1} m_{1} \times E_{x-t-inty} of [ELB \circ (S, \Phi^{\dagger}(\theta_{1}, Z), X]]$ 2. Share the learned generative model $P_{\theta}(Z, X)$ between sender & receiver. $\theta = \theta = 0$ of $X $					T	
 1. In order to develop a new compression method: I learn optimal parameters θ* of the generative model P₀(Z, X):	ational Exp	ectation M	laximizatior		UN	JUERSITAT JBINGEN
• learn optimal parameters θ^* of the generative model $P_{\theta}(\mathbf{Z}, \mathbf{X})$: $\Phi^{\dagger}(\theta_{ \mathbf{X} }) = arg_{max} [E_{\mathbf{X} \to training sol} [ELBO(\mathcal{P}, \Phi^{\dagger}(\mathbf{X}), \mathbf{X}]]$ $= arg_{max} [E_{\mathbf{X} \to training sol} [ELBO(\mathcal{P}, \Phi^{\dagger}(\mathbf{Y}), \mathbf{X}]]$ $= arg_{max} [E_{\mathbf{X} \to training sol} [mow ELBO(\mathcal{P}, \Phi, \mathbf{X})]$ 2. Share the learned generative model $P_{\theta^*}(\mathbf{Z}, \mathbf{X})$ between sender & receiver. $equal to the event \Phi \in \Phi + \mathfrak{H}, \mathbb{R} \to \mathfrak{H}\varphi = \mathfrak{h} \in \Phi \oplus \mathfrak{H}, \mathbb{R} \to \mathfrak{H}\varphi = \mathfrak{h} \in \Phi \oplus \mathfrak{H}, \mathbb{R} \to \mathfrak{H}\varphi = \mathfrak{h} \in \Phi \oplus \mathfrak{H}, \mathbb{R} \to \mathfrak{H}\varphi = \mathfrak{h} \in \Phi \oplus \mathfrak{H}, \mathbb{R} \to \mathfrak{H}\varphi = \mathfrak{h} \oplus \mathfrak{h} \oplus \mathfrak{h} \oplus \mathfrak{H}\varphi = \mathfrak{h} \oplus \mathfrak{h} \oplus \mathfrak{h} \oplus \mathfrak{h}\varphi = \mathfrak{h} \oplus \mathfrak{h} \oplus \mathfrak{h} \oplus \mathfrak{h} \oplus \mathfrak{h} \oplus \mathfrak{h}\varphi = \mathfrak{h} \oplus $	In order to devel	on a new com	pression method			
$ \begin{aligned} \varphi^{\dagger}(\varphi_{z}) &= a_{z} \min_{x} E(Bo(\varphi, \varphi_{x})) \\ \varphi^{\ast} &= a_{ij} \max_{y} E_{x-taing set} \left[E(Bo(\varphi, \varphi^{\dagger}(\varphi_{z}), \chi) \\ = a_{ij} \max_{y} E_{x-taing set} \left[\max_{y} E(Bo(\varphi, \varphi^{\dagger}(\varphi_{z}), \chi) \\ = a_{ij} \max_{y} E_{x-taing set} \left[\max_{y} E(Bo(\varphi, \varphi_{x})) \right] \\ \end{aligned} $ Share the learned generative model $P_{\theta^{\ast}}(Z, X)$ between sender & receiver. Share the learned $Q_{\phi^{\ast}}(Z)$. $\varphi^{\dagger} = depends on \chi$, so beth encode $\chi^{\dagger} = decode$ $a_{ij} = a_{ij} \max_{x} E(Bo(\varphi, \varphi_{x})) \\ \end{aligned} $ Where $E(E, X) = decode$ $A = decode$ a given data point χ . $A = depends on \chi$, so beth encode $\chi^{\dagger} = decode$ $a_{ij} = decode$		-	-		repeat until	conregence:
$ \begin{aligned} $				(LBo (2, \$, <u>*</u>)	draw x~	training set
 2. Share the learned generative model P_θ(Z, X) between sender & receiver. 3. In deployment: encode / decode a given data point x b Use entropy model Q_φ(Z). i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode & d	9* e arg ma	1 = Exactorials and 1		J, X]	report unt	Convergence:
 2. Share the learned generative model P_θ(Z, X) between sender & receiver. 3. In deployment: encode / decode a given data point x b Use entropy model Q_φ(Z). i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode i depends on x, so bith encode & decode & d				×)]	Lupdate	$\varphi \in \varphi + g, \forall \varphi \in Da \neq 4$ = $\vartheta + g, \forall g \in LBO(\vartheta, \theta)$
 3. In deployment: encode / decode a given data point (x). Use entropy model Q_φ (Z). \$	19		T			
 Use entropy model Q_φ(Z). 						Loop for each
 <i>b</i> + ← arging x ELBO(\$\$, \$\$, \$\$) Barder - Lecture 9 of the course "Data Compression With and Without Deep Probabilistic Model" - Summer Term of 2023 - more course materials at https://totakaler.gttubb.tot/teaching/compression art 2: Lecarning How to Do Inference (Fast) <i>With Without Deep Probabilistic Model</i> - Summer Term of 2023 - more course materials at https://totakaler.gttubb.tot/teaching/compression Problems: Learning the generative model requires an expensive inner loop for every training step. Expensive optimization over \$\$\phi\$ for each message \$\$x\$ we want compress \$\$/ decompress. Solution: amortized variational inference learn a mapping \$\$ from \$\$x\$ to variational parameters such that setting \$\$\phi\$ \$\mathcal{-}\$\$ f(\$\$x\$) approximately maximizes ELBO(\$\$\theta\$\$, \$\$\phi\$, \$\$x\$) for a given \$\$x\$. 		-	C .			training step an-
 <i>b</i> + ← arging x ELBO(\$\$, \$\$, \$\$) Barder - Lecture 9 of the course "Data Compression With and Without Deep Probabilistic Model" - Summer Term of 2023 - more course materials at https://totakaler.gttubb.tot/teaching/compression art 2: Lecarning How to Do Inference (Fast) <i>With Without Deep Probabilistic Model</i> - Summer Term of 2023 - more course materials at https://totakaler.gttubb.tot/teaching/compression Problems: Learning the generative model requires an expensive inner loop for every training step. Expensive optimization over \$\$\phi\$ for each message \$\$x\$ we want compress \$\$/ decompress. Solution: amortized variational inference learn a mapping \$\$ from \$\$x\$ to variational parameters such that setting \$\$\phi\$ \$\mathcal{-}\$\$ f(\$\$x\$) approximately maximizes ELBO(\$\$\theta\$\$, \$\$\phi\$, \$\$x\$) for a given \$\$x\$. 		A dit lang la a	an so both encode	& Sacada		
 <i>b</i> + ← arging x ELBO(\$\$, \$\$, \$\$) Barder - Lecture 9 of the course "Data Compression With and Without Deep Probabilistic Model" - Summer Term of 2023 - more course materials at https://totakaler.gttubb.tot/teaching/compression art 2: Lecarning How to Do Inference (Fast) <i>With Without Deep Probabilistic Model</i> - Summer Term of 2023 - more course materials at https://totakaler.gttubb.tot/teaching/compression Problems: Learning the generative model requires an expensive inner loop for every training step. Expensive optimization over \$\$\phi\$ for each message \$\$x\$ we want compress \$\$/ decompress. Solution: amortized variational inference learn a mapping \$\$ from \$\$x\$ to variational parameters such that setting \$\$\phi\$ \$\mathcal{-}\$\$ f(\$\$x\$) approximately maximizes ELBO(\$\$\theta\$\$, \$\$\phi\$, \$\$x\$) for a given \$\$x\$. 		readto fr	it by running (expen	she)SGD: /		
 Bander - Lecture 9 of the course "Data Compression With and Without Deep Probabilistic Models" - Summer Term of 2023 - more course materials at https://tokaaler.github.io/teaching/compress23/ The problems: Learning the generative model requires an expensive inner loop for every training step. Expensive optimization over φ for each message x we want compress / decompress. Solution: amortized variational inference learn a mapping f from x to variational parameters such that setting φ ← f(x) approximately maximizes ELBO(θ*, φ, x) for a given x. 		b¥	diama > ELBO (2)	* ¢, ×)		
 Problems: Learning the generative model requires an expensive inner loop for every training step. Expensive optimization over φ for each message x we want compress / decompress. Solution: amortized variational inference learn a mapping f from x to variational parameters such that setting φ ← f(x) approximately maximizes ELBO(θ*, φ, x) for a given x. 	Lecture 9 of the course "Data Compression With		•		hing/compress23/	3
 Problems: Learning the generative model requires an expensive inner loop for every training step. Expensive optimization over φ for each message x we want compress / decompress. Solution: amortized variational inference learn a mapping f from x to variational parameters such that setting φ ← f(x) approximately maximizes ELBO(θ*, φ, x) for a given x. 						
 Problems: Learning the generative model requires an expensive inner loop for every training step. Expensive optimization over φ for each message x we want compress / decompress. Solution: amortized variational inference learn a mapping f from x to variational parameters such that setting φ ← f(x) approximately maximizes ELBO(θ*, φ, x) for a given x. 	2: Learning	g How to	Do Inferenc	e (Fast)		serhard karls IVERSITAT JBINGEN
 Learning the generative model requires an expensive inner loop for every training step. Expensive optimization over φ for each message x we want compress / decompress. Solution: amortized variational inference learn a mapping f from x to variational parameters such that setting φ ← f(x) approximately maximizes ELBO(θ*, φ, x) for a given x. 		6		()		
 2. Expensive optimization over φ for each message x we want compress / decompress. Solution: amortized variational inference learn a mapping f from x to variational parameters such that setting φ ← f(x) approximately maximizes ELBO(θ*, φ, x) for a given x. 	Problems:					
 Solution: amortized variational inference learn a mapping f from x to variational parameters such that setting φ ← f(x) approximately maximizes ELBO(θ*, φ, x) for a given x. 	0 0	•	•		0 /	
learn a mapping f from \mathbf{x} to variational parameters such that setting $\phi \leftarrow f(\mathbf{x})$ approximately maximizes ELBO($\theta^*, \phi, \mathbf{x}$) for a given \mathbf{x} .	2. Expensive optin	nization over ϕ for	each message x we w	ant compress / d	lecompress.	
such that setting $\phi \leftarrow f(\mathbf{x})$ approximately maximizes $ELBO(heta^*, \phi, \mathbf{x})$ for a given \mathbf{x} .	Solution: amortize	ed variational inf	erence			
				$BO(\theta^*, \phi, \mathbf{x})$ for a	a given x .	
Notation: inference network $f_{\phi}(\mathbf{x})$; variational distribution $Q_{\phi}(\mathbf{Z} \mathbf{X} = \mathbf{x}) \leftarrow i_{\phi}$ the notation we've us					e in the not	atron we've used
• Example: Gaussian mean field variational distribution: \rightarrow inference network $f_{\phi}(\mathbf{x}) = (\mu_{\phi}(\mathbf{x}), \log \sigma_{\phi}^2(\mathbf{x}))$ outputs means and (log) variances	► Example: Gau				so far, this	would be Qf. (2)

ightarrow these parameterize a variational distribution $Q_{\phi}(\mathbf{Z} \,|\, \mathbf{x}) = \mathcal{N}ig(\mu_{\phi}(\mathbf{x}), \operatorname{diag}ig(\sigma_{\phi,1}^2(\mathbf{x}), \dots, \sigma_{\phi,k}^2(\mathbf{x}) ig) ig)$ $\rightarrow \text{ these parameterize a variational distribution } \varphi(\underline{c} | \underline{x}) = v(\mu_{\varphi}(\underline{x}), \dots \varphi(\underline{c}_{\varphi}, \underline{x})) = \psi_{\varphi}(\underline{x}, \underline{x}) + \psi_{\varphi}(\underline{x}, \underline{$

Variational Autoencoders (VAEs)

Combine variational expectation maximization with amortized variational inference.

- Lossless compression with variational autoencoders:
 - use bits-back trick \rightarrow Problem 9.2

Lossy compression with variational autoencoders:

- **Example:** data $\mathbf{X} = (X_i)_i$ are color images, i.e., each X_i is a continuous RGB value $\in [0, 1]$.
 - \rightarrow Prior may be learned, e.g.: $P_{\theta}(\mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathsf{diag}(\sigma_1^2, \dots, \sigma_{\texttt{num_channels}}^2)^{\otimes \texttt{spatial_dim}})$
 - \rightarrow Likelihood is parameterized by a (deconvolutional) neural network g_{θ} :

 $P_{\theta}(\mathbf{X} | \mathbf{Z}) = \prod_{i} P_{\theta}(X_{i} | \mathbf{Z})$ with density function $p_{\theta}(x_{i} | \mathbf{Z} = \mathbf{z}) = \mathcal{N}(x_{i}; g_{\theta,i}(\mathbf{z}), \frac{\beta}{2}I)$

Idea: just use g_{θ,i}(z) as the reconstruction of an image.
 (Don't bother using the likelihood P_θ(X | Z = z) to encode the true image.)

Brantrols q "rate I disloction trade-off" (see next week)

| 5

UNIVERSITÄT

- ► Likelihood no longer has a probabilistic meaning. But log P_θ(X | Z = z) is a distortion metric. ⇒ ELBO becomes a rate-distortion trade-off
- next week