Advanced Topics in Machine Learning
(deep generative modelling)

Lecture 3: Latent variable models and autoencoders

Nikolay Malkin

27 January 2026

Outline of Lecture 3

Autoencoders and friends:

- Some notes and leftovers from Lecture 2
- Two ingredients for VAEs
  - Autoencoders
  - Latent variable models
- Variational autoencoders
  - Basic VAE
  - The aggregate posterior problem and solutions

- Some notes and leftovers from Lecture 2

- Two ingredients for VAEs
  - Autoencoders
  - Latent variable models

- Variational autoencoders
  - Basic VAE
  - The aggregate posterior problem and solutions

Admin notes

- A more engaging tutorial format: come with questions
- This week's tutorial materials published by Thursday

Lecture 2 review

Setting:

- We have a **data distribution** π_{data} over $\mathbb{R}^d$ (from which we can sample, but we do not know its density function)
  - It could be the empirical distribution of a dataset
- We have a class of **model distributions** $\{\pi_\theta\}$ (with densities p_θ)
  - θ are the parameters of the model (*e.g.*, neural network weights, Gaussian mixture parameters)
  - Note that we do not necessarily know the density functions p_θ
- We seek θ such that π_θ approximates π_{data} well:

$$\theta^* = \arg\min_\theta D(\pi_\theta, \pi_{\text{data}})$$

Lecture 2 review

The KL divergence

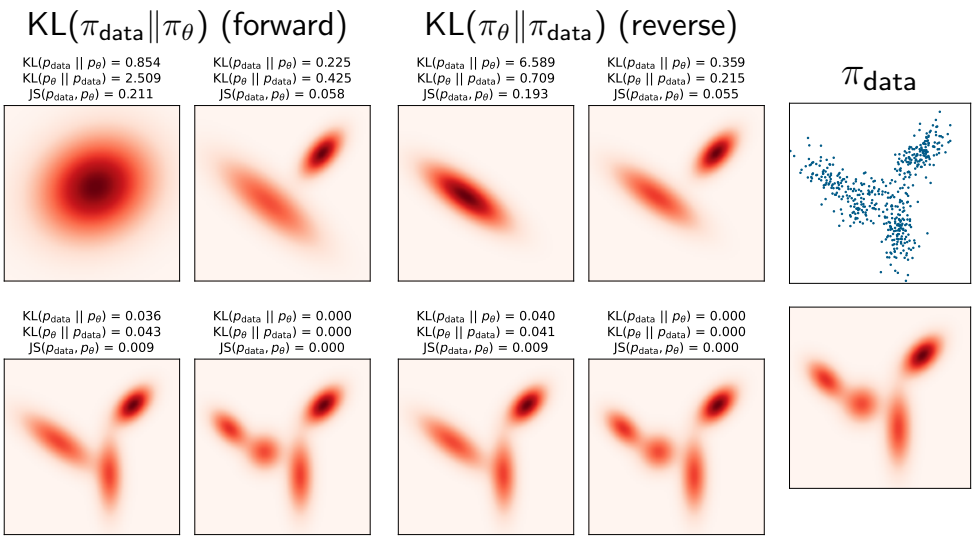
$$\mathrm{KL}(p\|q) = \mathbb{E}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right] = \int_{\mathbb{R}^d} p(x) \log \frac{p(x)}{q(x)}\,\mathrm{d}x$$

- Interesting properties studied in last week's tutorial sheet
- Minimising $\mathrm{KL}(\pi_{\text{data}}\|\pi_\theta) \equiv$ maximising sample log-likelihood $\mathbb{E}_{X\sim\pi_{\text{data}}}[\log p_\theta(X)]$
- Algorithm to fit θ using stochastic gradient descent:
  - Sample a minibatch $x_1, \ldots, x_m \sim \pi_{\text{data}}$
  - Compute gradient estimate:

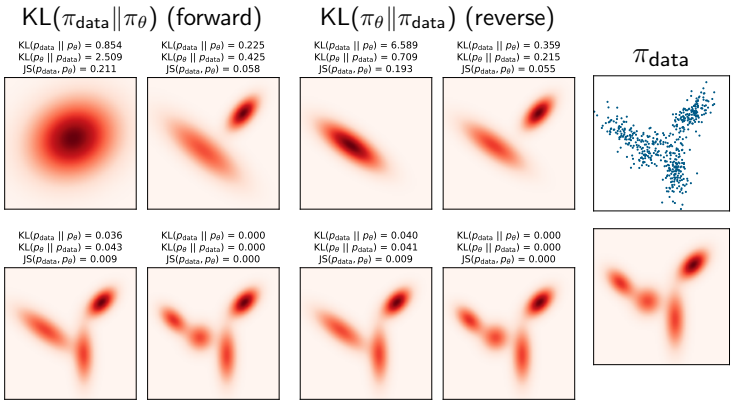
$$g = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta[-\log p_\theta(x_i)]$$

  - Update parameters: $\theta \leftarrow \theta - \eta g$ (or using your favourite optimiser)

Lecture 2 loose ends



Lecture 2 loose ends



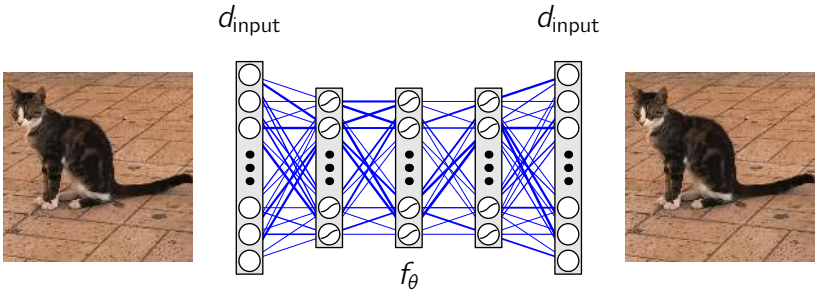
- Forward KL / MLE: **mode-covering** (high diversity, low fidelity)
- Reverse KL: **mode-seeking** (high fidelity, low diversity)
- JSD: balance between the two (more in a few weeks)

- Some notes and leftovers from Lecture 2
- Two ingredients for VAEs
  - Autoencoders
  - Latent variable models
- Variational autoencoders
  - Basic VAE
  - The aggregate posterior problem and solutions

Autoencoders

Train a neural network f_θ to predict x from x itself

$$\ell(\theta; x) = \|x - f_\theta(x)\|^2 \left[\mathcal{D} = \{x_1, \ldots, x_n\}, \text{ minimise } \sum_{i=1}^{n} \ell(\theta, x_i) \right]$$

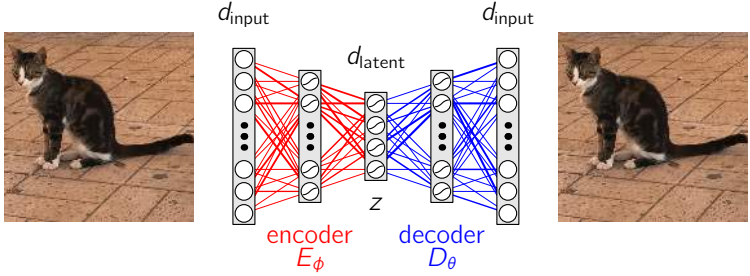


Autoencoders

Train a neural network f_θ to predict x from x itself

$$\ell(\theta; x) = \|x - f_\theta(x)\|^2 \quad \rightsquigarrow \quad \ell(\theta, \phi; x) = \|x - D_\theta(E_\phi(x))\|^2$$

Using a bottleneck layer induces a compressed representation $z = E_\phi(x)$ (a latent variable): (visualisation)

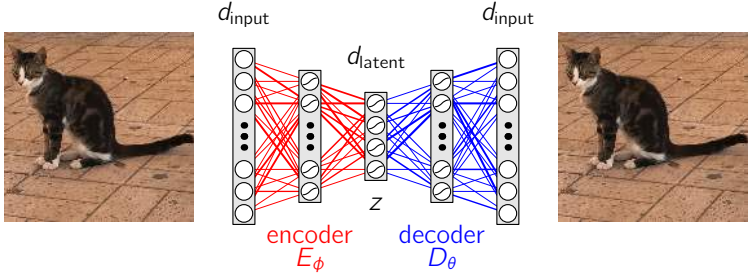


Autoencoders

Train a neural network f_θ to predict x from x itself

$$\ell(\theta; x) = \|x - f_\theta(x)\|^2 \quad \rightsquigarrow \quad \ell(\theta, \phi; x) = \|x - D_\theta(E_\phi(x))\|^2$$

Using a bottleneck layer induces a compressed representation $z = E_\phi(x)$ (a latent variable): (visualisation)



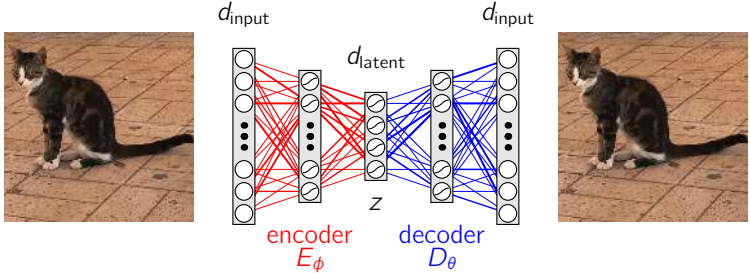
Can we use this to generate new data by sampling z and decoding it to $x = D_\theta(z)$?

Autoencoders

Train a neural network f_θ to predict x from x itself

$$\ell(\theta; x) = \|x - f_\theta(x)\|^2 \quad \rightsquigarrow \quad \ell(\theta, \phi; x) = \|x - D_\theta(E_\phi(x))\|^2$$

Using a bottleneck layer induces a compressed representation $z = E_\phi(x)$ (a latent variable): (visualisation)



Can we use this to generate new data by sampling z and decoding it to $x = D_\theta(z)$? No, because distribution from which to sample z not known.

Latent variable models

Generative models with **latent variables** z: To approximate π_{data} by a model p, model a joint distribution $p_\theta(x, z)$ over observed x and latent z to satisfy

$$p_\theta(x) = \int_z p_\theta(x, z)\, dz \approx \pi_{\text{data}}(x)$$

Latent variable models

Generative models with **latent variables** z: To approximate π_{data} by a model p, model a joint distribution $p_\theta(x, z)$ over observed x and latent z

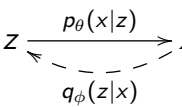
$$z \xrightarrow{p_\theta(x|z)} x \qquad p_\theta(x) = \int_z p_\theta(z) p_\theta(x \mid z)\, dz$$

'Ancestral' sampling of x:

- Sample the latent z from a distribution $p_\theta(z)$.
- Sample x from $p_\theta(x \mid z)$.

Latent variable models

Generative models with **latent variables** z: To approximate π_{data} by a model p, model a joint distribution $p_\theta(x, z)$ over observed x and latent z



$$p_\theta(x) = \int_z p_\theta(z) p_\theta(x \mid z)\, dz$$

'Ancestral' sampling of x:

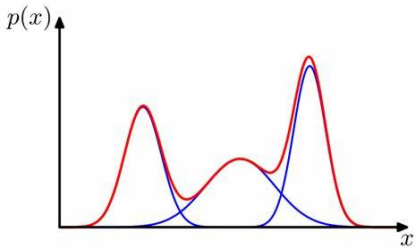
- Sample the latent z from a distribution $p_\theta(z)$.
- Sample x from $p_\theta(x \mid z)$.

The posterior over z, by Bayes' rule (we often approximate it by a $q_\phi(z \mid x)$):

$$p_\theta(z \mid x) = \frac{p_\theta(z) p_\theta(x \mid z)}{p_\theta(x)} \propto p_\theta(z) p_\theta(x \mid z)$$

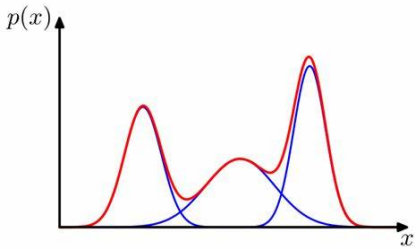
Examples of latent variable models

- **Mixture model** (*e.g.*, GMM): $\ell \to x$
  - $\ell \in \{1, 2, \ldots, n_{\text{components}}\}$ is a mixture index, categorical $p(\ell)$
  - For each value of ℓ, $p(x \mid \ell)$ is a distribution in a chosen family (e.g., Gaussian)



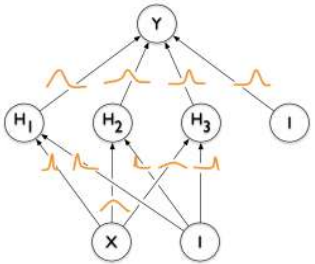
Examples of latent variable models

- **Mixture model** (*e.g.*, GMM): $\ell \to x$
  - $\ell \in \{1, 2, \ldots, n_{\text{components}}\}$ is a mixture index, categorical $p(\ell)$
  - For each value of ℓ, $p(x \mid \ell)$ is a distribution in a chosen family (e.g., Gaussian)
- **Topic model** (*e.g.*, LDA): $\theta \to d$ (topic vector $\to$ document)
  - $\theta \in \Delta^{n_{\text{topics}}}$ is a topic vector, Dirichlet $p(\theta)$
  - θ determines a multinomial distribution $p(d \mid \theta)$ over word count vectors d via the topic-word matrix



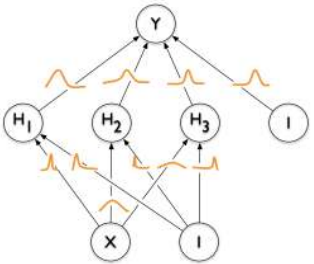
Examples of latent variable models

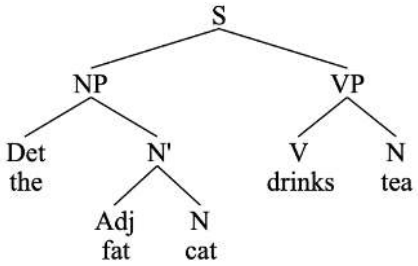
- **Bayesian neural network** (conditional): $\theta, x \to y$ (thought of as $\theta \to (x \to y)$, parameters $\to$ outputs given x)
  - θ are parameters of a neural net
  - y are the outputs of a neural net $p(y \mid x; \theta)$ with some inputs x



Examples of latent variable models

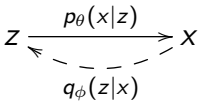
- **Bayesian neural network** (conditional): $\theta, x \to y$ (thought of as $\theta \to (x \to y)$, parameters $\to$ outputs given x)
  - θ are parameters of a neural net
  - y are the outputs of a neural net $p(y \mid x; \theta)$ with some inputs x
- **Probabilistic grammar** (*e.g.*, PCFG): $\tau \to s$ (syntax tree $\to$ sequence)
  - $p(\tau)$: τ is generated hierarchically by applying probabilistic replacement rules (S $\to$ NP VP, VP $\to$ V N, ...)
  - s is a sequence of leaves (terminal symbols)





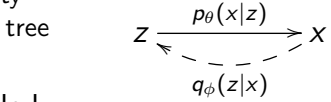
Latent variables ‘encode’ the data

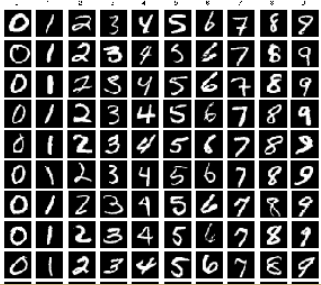
- The posterior $p_\theta(z \mid x)$ stochastically ‘encodes’ the data x into a latent z
  - An approximate posterior q_ϕ performs dimensionality reduction (topic representation of document, parse tree of text, compressed code of image)



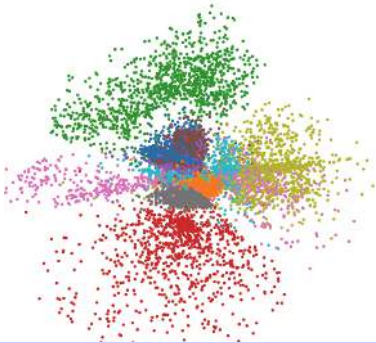
Latent variables 'encode' the data

- The posterior $p_\theta(z \mid x)$ stochastically 'encodes' the data x into a latent z
  - An approximate posterior q_ϕ performs dimensionality reduction (topic representation of document, parse tree of text, compressed code of image)
- From now on, we consider data in $\mathbb{R}^{d_{\text{data}}}$, encoded into latents in $\mathbb{R}^{d_{\text{latent}}}$ with $d_{\text{latent}} \ll d_{\text{data}}$



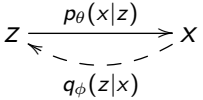


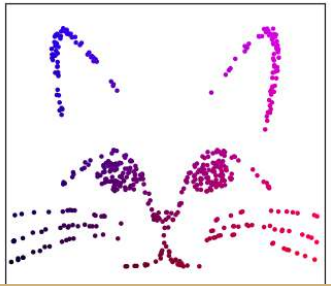
$\xrightarrow{q_\phi(z|x)}$



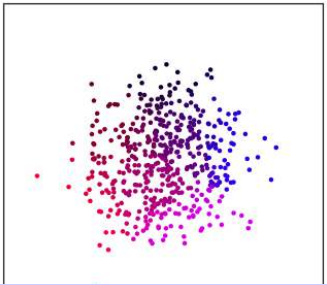
Latent variables ‘encode’ the data

- The posterior $p_\theta(z \mid x)$ stochastically ‘encodes’ the data x into a latent z
  - An approximate posterior q_ϕ performs dimensionality reduction (topic representation of document, parse tree of text, compressed code of image)
- From now on, we consider data in $\mathbb{R}^{d_{\text{data}}}$, encoded into latents in $\mathbb{R}^{d_{\text{latent}}}$ with $d_{\text{latent}} \ll d_{\text{data}}$





$q_\phi(z|x)$



MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i)$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i) = \sum_i \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i) = \sum_i \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

- **Gibbs' inequality:** for any distribution $q_i(z)$,

$$\log p_\theta(x_i) = \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz \geq \underbrace{\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)}\, dz}_{\text{evidence lower bound (ELBO)}}$$

with equality when $q_i(z) = p_\theta(z \mid x_i) \propto p_\theta(z) p_\theta(x_i \mid z)$ (true posterior)

MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i) = \sum_i \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

- **Gibbs' inequality:** for any distribution $q_i(z)$,

$$\log p_\theta(x_i) = \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz \geq \underbrace{\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)}\, dz}_{\text{evidence lower bound (ELBO)}}$$

$$= \log p_\theta(x) - \mathrm{KL}(q_i \,\|\, p_\theta(z|x_i)))$$

with equality when $q_i(z) = p_\theta(z \mid x_i) \propto p_\theta(z) p_\theta(x_i \mid z)$ (true posterior)

MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i) = \sum_i \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

- **Gibbs' inequality:** for any distribution $q_i(z)$,

$$\log p_\theta(x_i) = \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz \geq \underbrace{\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)}\, dz}_{\text{evidence lower bound (ELBO)}}$$

$$= \log p_\theta(x) - \mathrm{KL}(q_i \,\|\, p_\theta(z|x_i)))$$

with equality when $q_i(z) = p_\theta(z \mid x_i) \propto p_\theta(z) p_\theta(x_i \mid z)$ (true posterior)

MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i) = \sum_i \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

- **Gibbs' inequality:** for any distribution $q_i(z)$,

$$\log p_\theta(x_i) = \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz \geq \underbrace{\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)}\, dz}_{\text{evidence lower bound (ELBO)}}$$

$$= \log p_\theta(x) - \mathrm{KL}(q_i \,\|\, p_\theta(z|x_i)))$$

  with equality when $q_i(z) = p_\theta(z \mid x_i) \propto p_\theta(z) p_\theta(x_i \mid z)$ (true posterior)
- Idea: to maximise $\log p(x^t)$, maximise the ELBO
  - $\rightsquigarrow$ **EM alg.** ("Maximum likelihood from incomplete data" [Dempster et al., 1977])

Gradient-based EM

Idea: to maximise $\log p_\theta(x_i)$, maximise the ELBO

$$\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)} = \mathbb{E}_{z \sim q_i} \left[\log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)} \right]$$

Gradient-based EM

Idea: to maximise $\log p_\theta(x_i)$, maximise the ELBO

$$\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)} = \mathbb{E}_{z \sim q_i} \left[\log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)} \right]$$

- **(Amortised) variational EM:** approximate q_i with a model $q_\phi(z \mid x_i)$ and train it to approximate the true posterior $p_\theta(z \mid x_i)$ (**E-step**)

Gradient-based EM

Idea: to maximise $\log p_\theta(x_i)$, maximise the ELBO

$$\int_z q_i(z) \log \frac{p_\theta(z)p_\theta(x_i \mid z)}{q_i(z)} = \mathbb{E}_{z\sim q_i}\left[\log \frac{p_\theta(z)p_\theta(x_i \mid z)}{q_i(z)}\right]$$

- **(Amortised) variational EM:** approximate q_i with a model $q_\phi(z \mid x_i)$ and train it to approximate the true posterior $p_\theta(z \mid x_i)$ (**E-step**)
- **(Gradient-based) M-step:** sample $z \sim q_\phi(z \mid x_i)$ and maximise $\log p_\theta(z)p_\theta(x_i \mid z)$ w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$

Gradient-based EM

Idea: to maximise $\log p_\theta(x_i)$, maximise the ELBO

$$\int_z q_i(z) \log \frac{p_\theta(z)p_\theta(x_i \mid z)}{q_i(z)} = \mathbb{E}_{z\sim q_i}\left[\log \frac{p_\theta(z)p_\theta(x_i \mid z)}{q_i(z)}\right]$$

- **(Amortised) variational EM:** approximate q_i with a model $q_\phi(z \mid x_i)$ and train it to approximate the true posterior $p_\theta(z \mid x_i)$ (**E-step**)
- **(Gradient-based) M-step:** sample $z \sim q_\phi(z \mid x_i)$ and maximise $\log p_\theta(z)p_\theta(x_i \mid z)$ w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$
- Intuitively:
  - E-step: learn to encode x into z that explains it well
  - M-step: for data x, sample explanation z and learn to recover x from z

- Some notes and leftovers from Lecture 2
- Two ingredients for VAEs
  - Autoencoders
  - Latent variable models
- Variational autoencoders
  - Basic VAE
  - The aggregate posterior problem and solutions

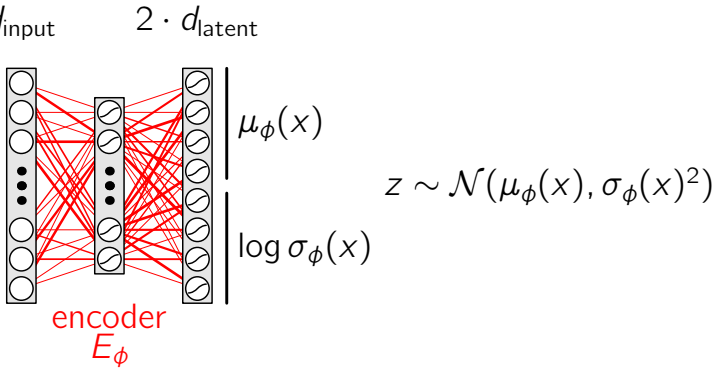
Variational autoencoders

How can we force the latent variable z in an autoencoder to follow a known distribution $p(z)$ (such as Gaussian $\mathcal{N}(0, I_d)$ in $\mathbb{R}^d$)?

Variational autoencoders

How can we force the latent variable z in an autoencoder to follow a known distribution $p(z)$ (such as Gaussian $\mathcal{N}(0, I_d)$ in $\mathbb{R}^d$)?

- Make the encoder E_ϕ stochastic: $z \sim q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x)\sigma^2_\phi(x))$
  - Reparametrisation: $z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$, $\epsilon \sim \mathcal{N}(0, I_d)$



Variational autoencoders

How can we force the latent variable z in an autoencoder to follow a known distribution $p(z)$ (such as Gaussian $\mathcal{N}(0, I_d)$ in $\mathbb{R}^d$)?

- Make the encoder E_ϕ stochastic: $z \sim q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x)\sigma^2_\phi(x))$
  - Reparametrisation: $z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$, $\epsilon \sim \mathcal{N}(0, I_d)$
- Objective (derived on next slide): reconstruct x from z while matching $q_\phi(z \mid x)$ to $\mathcal{N}(0, I_d)$:

$$\ell(\theta, \phi; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \|x - D_\theta(z)\|^2}_{\text{autoencoder / reconstruction loss}}$$

Variational autoencoders

How can we force the latent variable z in an autoencoder to follow a known distribution $p(z)$ (such as Gaussian $\mathcal{N}(0, I_d)$ in $\mathbb{R}^d$)?

- Make the encoder E_ϕ stochastic: $z \sim q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x)\sigma^2_\phi(x))$
  - Reparametrisation: $z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$, $\epsilon \sim \mathcal{N}(0, I_d)$
- Objective (derived on next slide): reconstruct x from z while matching $q_\phi(z \mid x)$ to $\mathcal{N}(0, I_d)$:

$$\ell(\theta, \phi; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \|x - D_\theta(z)\|^2}_{\text{autoencoder / reconstruction loss}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))}_{\text{prior-matching loss}}$$

Variational autoencoders

How can we force the latent variable z in an autoencoder to follow a known distribution $p(z)$ (such as Gaussian $\mathcal{N}(0, I_d)$ in $\mathbb{R}^d$)?

- Make the encoder E_ϕ stochastic: $z \sim q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x)\sigma^2_\phi(x))$
  - Reparametrisation: $z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$, $\epsilon \sim \mathcal{N}(0, I_d)$
- Objective (derived on next slide): reconstruct x from z while matching $q_\phi(z \mid x)$ to $\mathcal{N}(0, I_d)$:

$$\ell(\theta, \phi; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \|x - D_\theta(z)\|^2}_{\text{autoencoder / reconstruction loss}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))}_{\text{prior-matching loss}}$$

- The KL for Gaussians (tutorial exercise):

$$\mathrm{KL}(\mathcal{N}(\mu_\phi(x), \sigma^2_\phi(x)) \| \mathcal{N}(0, I_d)) = \frac{1}{2} \left(\|\sigma_\phi(x)\|^2 + \|\mu_\phi(x)\|^2 - d_{\text{latent}} - \log \|\sigma_\phi(x)\|^2 \right)$$

Variational autoencoders

How can we force the latent variable z in an autoencoder to follow a known distribution $p(z)$ (such as Gaussian $\mathcal{N}(0, I_d)$ in $\mathbb{R}^d$)?

- Make the encoder E_ϕ stochastic: $z \sim q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x)\sigma^2_\phi(x))$
  - Reparametrisation: $z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$, $\epsilon \sim \mathcal{N}(0, I_d)$
- Objective (derived on next slide): reconstruct x from z while matching $q_\phi(z \mid x)$ to $\mathcal{N}(0, I_d)$:

$$\ell(\theta, \phi; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \|x - D_\theta(z)\|^2}_{\text{autoencoder / reconstruction loss}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))}_{\text{prior-matching loss}}$$

- The KL for Gaussians (tutorial exercise):

$$\mathrm{KL}(\mathcal{N}(\mu_\phi(x), \sigma^2_\phi(x)) \| \mathcal{N}(0, I_d)) = \frac{1}{2} \left(\|\sigma_\phi(x)\|^2 + \|\mu_\phi(x)\|^2 - d_{\text{latent}} - \log \|\sigma_\phi(x)\|^2 \right)$$

Which parts of the loss depend on θ and on ϕ?

Variational autoencoders

How can we force the latent variable z in an autoencoder to follow a known distribution $p(z)$ (such as Gaussian $\mathcal{N}(0, I_d)$ in $\mathbb{R}^d$)?

- Make the encoder E_ϕ stochastic: $z \sim q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x)\sigma^2_\phi(x))$
  - Reparametrisation: $z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$, $\epsilon \sim \mathcal{N}(0, I_d)$
- Objective (derived on next slide): reconstruct x from z while matching $q_\phi(z \mid x)$ to $\mathcal{N}(0, I_d)$:

$$\ell(\theta, \phi; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \|x - D_\theta(z)\|^2}_{\text{autoencoder / reconstruction loss}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))}_{\text{prior-matching loss}}$$

- The KL for Gaussians (tutorial exercise):

$$\mathrm{KL}(\mathcal{N}(\mu_\phi(x), \sigma^2_\phi(x)) \| \mathcal{N}(0, I_d)) = \frac{1}{2} \left(\|\sigma_\phi(x)\|^2 + \|\mu_\phi(x)\|^2 - d_{\text{latent}} - \log \|\sigma_\phi(x)\|^2 \right)$$

Which parts of the loss depend on θ and on ϕ? ϕ: both. θ: only second term.

A closer look at the VAE objective

$$\ell(\theta, \phi; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \|x - D_\theta(z)\|^2}_{\text{autoencoder / reconstruction loss}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))}_{\text{prior-matching loss}}$$

- View the reconstruction term as a negative log-likelihood $-\log p_\theta(x \mid z)$, where $x = D_\theta(z) + (\text{Gaussian noise of fixed variance})$, up to constant

A closer look at the VAE objective

$$\ell(\theta, \phi; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \|x - D_\theta(z)\|^2}_{\text{autoencoder / reconstruction loss}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))}_{\text{prior-matching loss}}$$

- View the reconstruction term as a negative log-likelihood $-\log p_\theta(x \mid z)$, where $x = D_\theta(z) + (\text{Gaussian noise of fixed variance})$, up to constant
- View q_ϕ as an amortised variational posterior

A closer look at the VAE objective

$$\ell(\theta, \phi; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \|x - D_\theta(z)\|^2}_{\text{autoencoder / reconstruction loss}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))}_{\text{prior-matching loss}}$$

- View the reconstruction term as a negative log-likelihood $-\log p_\theta(x \mid z)$, where $x = D_\theta(z) + (\text{Gaussian noise of fixed variance})$, up to constant
- View q_ϕ as an amortised variational posterior
- The ELBO:

$$\begin{aligned} \log p_\theta(x) &\geq \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log \frac{p_\theta(x \mid z) p(z)}{q_\phi(z \mid x)} \right] \\ &= \log p_\theta(x) - \mathrm{KL}(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x)) \end{aligned}$$

A closer look at the VAE objective

$$\ell(\theta, \phi; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \|x - D_\theta(z)\|^2}_{\text{autoencoder / reconstruction loss}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))}_{\text{prior-matching loss}}$$

- View the reconstruction term as a negative log-likelihood $-\log p_\theta(x \mid z)$, where $x = D_\theta(z) + (\text{Gaussian noise of fixed variance})$, up to constant
- View q_ϕ as an amortised variational posterior
- The ELBO:

$$\begin{aligned}
\log p_\theta(x) &\geq \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log \frac{p_\theta(x \mid z) p(z)}{q_\phi(z \mid x)} \right] \\
&= \log p_\theta(x) - \mathrm{KL}(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x)) \\
&= \mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\theta(x \mid z)] - \mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))
\end{aligned}$$

A closer look at the VAE objective

$$\ell(\theta, \phi; x) = \underbrace{\mathbb{E}_{z \sim q_\phi(z|x)} \|x - D_\theta(z)\|^2}_{\text{autoencoder / reconstruction loss}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))}_{\text{prior-matching loss}}$$

- View the reconstruction term as a negative log-likelihood $-\log p_\theta(x \mid z)$, where $x = D_\theta(z) + (\text{Gaussian noise of fixed variance})$, up to constant
- View q_ϕ as an amortised variational posterior
- The ELBO:

$$\begin{aligned}
\log p_\theta(x) &\geq \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log \frac{p_\theta(x \mid z) p(z)}{q_\phi(z \mid x)} \right] \\
&= \log p_\theta(x) - \mathrm{KL}(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x)) \\
&= \mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\theta(x \mid z)] - \mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))
\end{aligned}$$

"ELBO surgery" $\rightsquigarrow$ VAE is performing variational EM!

Likelihood estimation with VAEs

- Once q_ϕ and p_θ are trained, we can estimate the log-likelihood of new data x using the ELBO

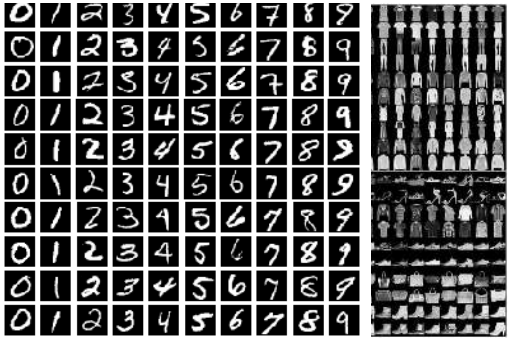
$$\log p_\theta(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log \frac{p_\theta(x \mid z) p(z)}{q_\phi(z \mid x)} \right]$$

Likelihood estimation with VAEs

- Once q_ϕ and p_θ are trained, we can estimate the log-likelihood of new data x using the ELBO

$$\log p_\theta(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log \frac{p_\theta(x \mid z) p(z)}{q_\phi(z \mid x)} \right]$$

- Not always a good estimate for anomaly detection...
  - Train a VAE on MNIST, evaluate on Fashion-MNIST, get a higher log-likelihood!
  - Related to zero-avoiding behaviour of forward KL and the aggregate posterior problem (this next)



What's wrong with VAEs?

- The prior $p(z)$ is not 'filled out' well by the posteriors $q_\phi(z \mid x)$ over $x \sim \pi_{\text{data}}$
- Samples from $p(z)$ are not encoded to by any x in the data distribution and thus decode poorly (recall the visualisation)

What's wrong with VAEs?

- The prior $p(z)$ is not 'filled out' well by the posteriors $q_\phi(z \mid x)$ over $x \sim \pi_{\text{data}}$
- Samples from $p(z)$ are not encoded to by any x in the data distribution and thus decode poorly (recall the visualisation)

Possible solutions (gallery):

- Improved posterior estimation (importance weighting, more expressive posteriors)
- Refitting the prior $p_\theta(z)$ after training (or jointly learning it)
- Auxiliary/modified losses
  - Regularised VAEs (e.g., β-VAE,)
  - Contrastive objectives (e.g., 'Forget-me-not' [Menon et al., 2022]), wake-sleep training
- Hierarchical VAEs
- Discrete-latent VAEs (e.g., VQ-VAE, [Van den Oord et al., 2017])

Improved priors and variational posteriors

- Importance-weighted autoencoder (IWAE; [Burda et al., 2016]): Instead of the ELBO

$$\log p_\theta(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)} \left[\log \frac{p_\theta(x \mid z) p(z)}{q_\phi(z \mid x)} \right]$$

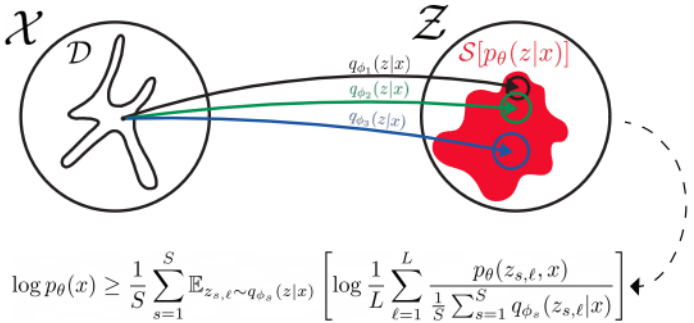
use a tighter bound

$$\log p_\theta(x) \geq \mathbb{E}_{z_1, \ldots, z_K \sim q_\phi(z|x)} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(x \mid z_k) p(z_k)}{q_\phi(z_k \mid x)} \right]$$

(becomes tighter as $K \to \infty$ and approaches the true $\log p_\theta(x)$)

Improved priors and variational posteriors

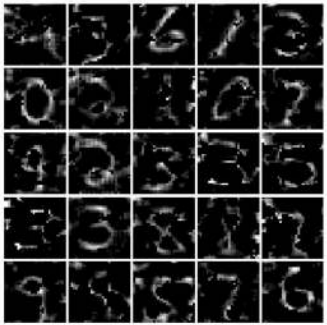
- Importance-weighted autoencoder (IWAE; [Burda et al., 2016])
- More expressive variational posteriors:
  - GMM posterior (see "Cooperation in the latent space" [Kviman et al., 2023])



- More complex models or MCMC in latent space

Improved priors and variational posteriors

- Importance-weighted autoencoder (IWAE; [Burda et al., 2016])
- More expressive variational posteriors:
  - GMM posterior (see "Cooperation in the latent space" [Kviman et al., 2023])
  - More complex models or MCMC in latent space
- Learning the prior:
  - VampPrior ([Tomczak & Welling, 2018]): prior is a mixture of variational posteriors $q_\phi(z \mid x)$ over a set of learnt fake inputs x
  - Large text-to-image models (incl. Stable Diffusion) train other types of models (neural ODEs) to act as text-conditioned priors in the latent space of a VAE



Hierarchical VAE

Instead of $z \to x$, have several layers of latent variables (or similar schemes):

$$z_T \underset{q_\phi(z_T|z_{T-1})}{\overset{p_\theta(z_{T-1}|z_T)}{\rightleftarrows}} z_{T-1} \longrightarrow \cdots \longrightarrow z_1 \underset{q_\phi(z_1|x)}{\overset{p_\theta(x|z_1)}{\rightleftarrows}} x$$

VAE

Hierarchical VAE

['NVAE', Vahdat & Kautz, 2020]

Conclusion and looking ahead

- VAEs as the simplest form of both **dimensionality reduction** and **generative modelling with latent variables**

Conclusion and looking ahead

- VAEs as the simplest form of both **dimensionality reduction** and **generative modelling with latent variables**
- Design considerations not discussed in detail:
  - Choice of neural architectures (usually, the decoder is much larger than the encoder)
  - Choice of latent dimension
  - Various ways to prevent posterior collapse and ensure latent space coverage
- Also not discussed: conditional VAEs (common in applications)

Conclusion and looking ahead

- VAEs as the simplest form of both **dimensionality reduction** and **generative modelling with latent variables**
- Design considerations not discussed in detail:
  - Choice of neural architectures (usually, the decoder is much larger than the encoder)
  - Choice of latent dimension
  - Various ways to prevent posterior collapse and ensure latent space coverage
- Also not discussed: conditional VAEs (common in applications)
- Next time: models with exact likelihood and no latents