Auto-encoding variational Bayes

Durk Kingma and Max Welling

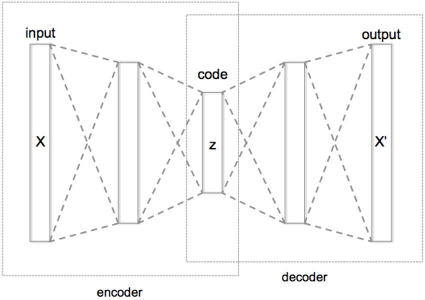
Mobile Robotics Lab Reading Group, McGill University

Sept 22, 2016

Overview

- Auto-encoders
- Variational Inference
- An example: data from mixture of Gaussians
- Evidence Lower Bound (ELBO)
- Mean-field approximation
- Optimization algorithms for VI and stochastic VI
- Auto-encoding variational Bayes

Auto-encoders



Unsupervised learning of efficient encodings $\mathbf{z}$ of observed data $\mathbf{x}$

Overview

- Auto-encoders
- Variational Inference
- An example: data from mixture of Gaussians
- Evidence Lower Bound (ELBO)
- Mean-field approximation
- Optimization algorithms for VI and stochastic VI
- Auto-encoding variational Bayes

Variational Inference: definitions

- Latent variables $\mathbf{z} \in \mathbb{R}^m$ and observed data $\mathbf{x} = x_{1:n}$
- $x_i \in \mathbb{R}^d$

Variational Inference: definitions

- Latent variables $\mathbf{z} \in \mathbb{R}^m$ and observed data $\mathbf{x} = x_{1:n}$
- $x_i \in \mathbb{R}^d$
- Joint model is $p(\mathbf{z}, \mathbf{x}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$

Variational Inference: definitions

- Latent variables $\mathbf{z} \in \mathbb{R}^m$ and observed data $\mathbf{x} = x_{1:n}$
- $x_i \in \mathbb{R}^d$
- Joint model is $p(\mathbf{z}, \mathbf{x}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$
- *Inference* means computing the posterior $p(\mathbf{z}|\mathbf{x})$
- *Inference* $=$ How do we encode the given data $\mathbf{x}$ using latent variables $\mathbf{z}$?

Some ways of computing $p(\mathbf{z}|\mathbf{x})$

- **MCMC**: Gibbs, Hamiltonian Monte Carlo, Metropolis-Hastings, etc.
- Most of them slow (large mixing times)

Some ways of computing $p(\mathbf{z}|\mathbf{x})$

- **MCMC**: Gibbs, Hamiltonian Monte Carlo, Metropolis-Hastings, etc.
- Most of them slow (large mixing times)
- **Analytically**: $p(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}, \mathbf{x})/p(\mathbf{x})$
- Computing the evidence $p(\mathbf{x})$ may be intractable

Variational Inference: main idea

- $q^*(\mathbf{z}) = \underset{q(\mathbf{z}) \in \mathcal{Q}}{\operatorname{argmin}} \, \mathsf{KL}(q(\mathbf{z}) \,||\, p(\mathbf{z}|\mathbf{x}))$
- $\mathcal{Q}$ is a family of "simpler" distributions q compared to p.

Variational Inference: main idea

- $q^*(\mathbf{z}) = \underset{q(\mathbf{z})\in\mathcal{Q}}{\operatorname{argmin}} \ \mathsf{KL}(q(\mathbf{z}) \ || \ p(\mathbf{z}|\mathbf{x}))$
- $\mathcal{Q}$ is a family of "simpler" distributions q compared to p.
- KL divergence is an asymmetric, nonnegative measure, not a norm. It doesn't obey the triangle inequality.
- It measures the extra bits you need to spend to compress samples from q using a code optimized for p.
- Difference measure between probability distributions.

Overview

- Auto-encoders
- Variational Inference
- An example: data from mixture of Gaussians
- Evidence Lower Bound (ELBO)
- Mean-field approximation
- Optimization algorithms for VI and stochastic VI
- Auto-encoding variational Bayes

An example: data from mixture of Gaussians

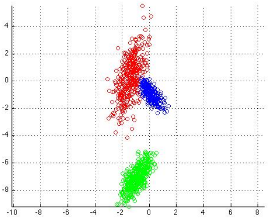
- K mixture components, corresponding to normal distributions
- Means $\boldsymbol{\mu} = \{\mu_1, ..., \mu_K\} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$
- Mixture selection variable $c_i \sim$ Categorical$(1/K, ..., 1/K)$

An example: data from mixture of Gaussians

- K mixture components, corresponding to normal distributions
- Means $\boldsymbol{\mu} = \{\mu_1, ..., \mu_K\} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$
- Mixture selection variable $c_i \sim$ Categorical$(1/K, ..., 1/K)$
- Joint model $p(\boldsymbol{\mu}, c_{1:n}, x_{1:n}) = p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(c_i) p(x_i | c_i, \boldsymbol{\mu})$

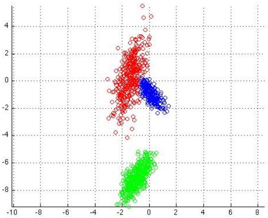
An example: data from mixture of Gaussians

- K mixture components, corresponding to normal distributions
- Means $\boldsymbol{\mu} = \{\mu_1, ..., \mu_K\} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$
- Mixture selection variable $c_i \sim$ Categorical$(1/K, ..., 1/K)$
- Joint model $p(\boldsymbol{\mu}, c_{1:n}, x_{1:n}) = p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(c_i) p(x_i | c_i, \boldsymbol{\mu})$



An example: data from mixture of Gaussians

- K mixture components, corresponding to normal distributions
- Means $\boldsymbol{\mu} = \{\mu_1, ..., \mu_K\} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$
- Mixture selection variable $c_i \sim \text{Categorical}(1/K, ..., 1/K)$
- Joint model $p(\boldsymbol{\mu}, c_{1:n}, x_{1:n}) = p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(c_i) p(x_i | c_i, \boldsymbol{\mu})$



- Each c_i has K options, and we have n data points, so $O(K^n)$ to evaluate $p(x_{1:n}) = \int p(\boldsymbol{\mu}, c_{1:n}, x_{1:n}) d\boldsymbol{\mu} dc_{1:n}$

An example: data from mixture of Gaussians

- Each c_i has K options, and we have n data points, so $O(K^n)$ to evaluate $p(x_{1:n}) = \int p(\boldsymbol{\mu}, c_{1:n}, x_{1:n}) d\boldsymbol{\mu} dc_{1:n}$

An example: data from mixture of Gaussians

- Each c_i has K options, and we have n data points, so $O(K^n)$ to evaluate $p(x_{1:n}) = \int p(\boldsymbol{\mu}, c_{1:n}, x_{1:n}) d\boldsymbol{\mu} dc_{1:n}$
- Takeaway message: can't use direct estimation of the evidence $p(x_{1:n})$

An example: data from mixture of Gaussians

- Each c_i has K options, and we have n data points, so $O(K^n)$ to evaluate $p(x_{1:n}) = \int p(\boldsymbol{\mu}, c_{1:n}, x_{1:n}) d\boldsymbol{\mu} dc_{1:n}$
- Takeaway message: can't use direct estimation of the evidence $p(x_{1:n})$
- In this particular example we can use EM, but in general it assumes that you know $p(\mathbf{z}|\mathbf{x})$

Overview

- Auto-encoders
- Variational Inference
- An example: data from mixture of Gaussians
- Evidence Lower Bound (ELBO)
- Mean-field approximation
- Optimization algorithms for VI and stochastic VI
- Auto-encoding variational Bayes

Evidence Lower Bound (ELBO)

$$\mathsf{KL}(q(\mathbf{z}) \mid\mid p(\mathbf{z}|\mathbf{x})) \quad = \quad \mathbb{E}_q[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}]$$

Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\mathrm{KL}(q(\mathbf{z}) \mid\mid p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_q[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}|\mathbf{x})]
\end{aligned}
$$

Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\mathsf{KL}(q(\mathbf{z}) \mid\mid p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_q[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z},\mathbf{x})] + \mathbb{E}_q[\log p(\mathbf{x})]
\end{aligned}
$$

Evidence Lower Bound (ELBO)

$$\begin{aligned}
\mathsf{KL}(q(\mathbf{z}) \mid\mid p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_q[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x})] + \mathbb{E}_q[\log p(\mathbf{x})] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x})] + \log p(\mathbf{x})
\end{aligned}$$

Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\mathsf{KL}(q(\mathbf{z}) \mid\mid p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_q[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x})] + \mathbb{E}_q[\log p(\mathbf{x})] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x})] + \log p(\mathbf{x}) \\
&= -\mathsf{ELBO}_q(\mathbf{x}) + \log p(\mathbf{x})
\end{aligned}
$$

Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\mathrm{KL}(q(\mathbf{z}) \mid\mid p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_q[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z},\mathbf{x})] + \mathbb{E}_q[\log p(\mathbf{x})] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z},\mathbf{x})] + \log p(\mathbf{x}) \\
&= -\mathrm{ELBO}_q(\mathbf{x}) + \log p(\mathbf{x})
\end{aligned}
$$

$$
\text{Log-evidence} \quad \log p(\mathbf{x}) = \mathrm{KL}(q(\mathbf{z}) \mid\mid p(\mathbf{z}|\mathbf{x})) + \mathrm{ELBO}_q(\mathbf{x})
$$

Evidence Lower Bound (ELBO)

$$\begin{aligned}
\mathrm{KL}(q(\mathbf{z}) \mid\mid p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_q[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z},\mathbf{x})] + \mathbb{E}_q[\log p(\mathbf{x})] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z},\mathbf{x})] + \log p(\mathbf{x}) \\
&= -\mathrm{ELBO}_q(\mathbf{x}) + \log p(\mathbf{x})
\end{aligned}$$

$$\text{Log-evidence} \quad \log p(\mathbf{x}) = \mathrm{KL}(q(\mathbf{z}) \mid\mid p(\mathbf{z}|\mathbf{x})) + \mathrm{ELBO}_q(\mathbf{x})$$

Variational Inference $\rightarrow$ find $q(\mathbf{z})$ that maximizes ELBO_q

Overview

- Auto-encoders
- Variational Inference
- An example: data from mixture of Gaussians
- Evidence Lower Bound (ELBO)
- Mean-field approximation
- Optimization algorithms for VI and stochastic VI
- Auto-encoding variational Bayes

Mean-field approximation

- Simplification for posterior approximator $q(\mathbf{z})$:
- $q(\mathbf{z}) = \prod_j q_j(z_j)$
- All latent variables z_j are mutually independent
- Each is governed by its own distribution q_j

Mean-field approximation

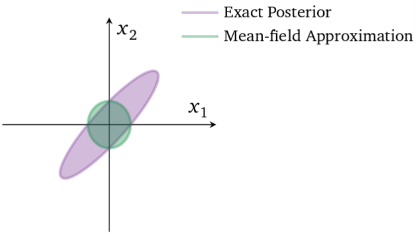
- Simplification for posterior approximator $q(\mathbf{z})$:
- $q(\mathbf{z}) = \prod_j q_j(z_j)$
- All latent variables z_j are mutually independent
- Each is governed by its own distribution q_j
- WHY? It makes the optimization easier (analytical gradients)

Mean-field approximation

- Simplification for posterior approximator $q(\mathbf{z})$:
- $q(\mathbf{z}) = \prod_j q_j(z_j)$
- All latent variables z_j are mutually independent
- Each is governed by its own distribution q_j
- WHY? It makes the optimization easier (analytical gradients)
- WHY NOT? It fails to model correlations among latent variables, and underestimates variance

Mean-field approximation

- Simplification for posterior approximator $q(\mathbf{z})$:
- $q(\mathbf{z}) = \prod_j q_j(z_j)$
- All latent variables z_j are mutually independent
- Each is governed by its own distribution q_j
- WHY? It makes the optimization easier (analytical gradients)
- WHY NOT? It fails to model correlations among latent variables, and underestimates variance



Overview

- Auto-encoders
- Variational Inference
- An example: data from mixture of Gaussians
- Evidence Lower Bound (ELBO)
- Mean-field approximation
- Optimization algorithms for VI and stochastic VI
- Auto-encoding variational Bayes

Optimization algorithms

- Algo $\#1$: coordinate ascent along each latent variable of ELBO
- Main problem is that it evaluates ELBO on the entire dataset (not great for big data)
- Also susceptible to local minima

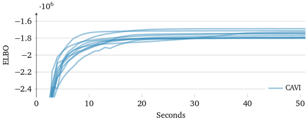


Figure 2: Different initializations may lead CAVI to find different local optima of the ELBO.

Optimization algorithms

- Algo $\#1$: coordinate ascent along each latent variable of ELBO
- Main problem is that it evaluates ELBO on the entire dataset (not great for big data)
- Also susceptible to local minima

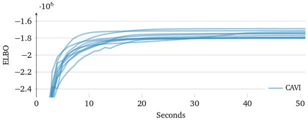


Figure 2: Different initializations may lead CAVI to find different local optima of the ELBO.

- Algo $\#2$: stochastic optimization over all latent variables
- Uses the natural gradient to account for manifold on which distributions live
- Evaluates ELBO on single data points, or minibatches

Overview

- Auto-encoders
- Variational Inference
- An example: data from mixture of Gaussians
- Evidence Lower Bound (ELBO)
- Mean-field approximation
- Optimization algorithms for VI and stochastic VI
- Auto-encoding variational Bayes

Auto-encoding Variational Bayes: assumptions

- Data $x_i \in \mathbb{R}^d$ are i.i.d and generated from a parametric graphical model:
- $p_{\theta^*}(\mathbf{z}, \mathbf{x}) = p_{\theta^*}(\mathbf{z}) p_{\theta^*}(\mathbf{x}|\mathbf{z})$
- The true parameter θ^* is unknown.

Auto-encoding Variational Bayes: assumptions

- Data $x_i \in \mathbb{R}^d$ are i.i.d and generated from a parametric graphical model:
- $p_{\theta^*}(\mathbf{z}, \mathbf{x}) = p_{\theta^*}(\mathbf{z}) p_{\theta^*}(\mathbf{x}|\mathbf{z})$
- The true parameter θ^* is unknown.
- Large dataset, and intractable evidence $p_{\theta^*}(\mathbf{x})$ and posterior $p_{\theta^*}(\mathbf{z}|\mathbf{x})$ distributions

Auto-encoding Variational Bayes: contributions

- Parameter estimation θ of the data generative model (decoder) $p_{\theta^*}(\mathbf{x}|\mathbf{z})$

Auto-encoding Variational Bayes: contributions

- Parameter estimation θ of the data generative model (decoder) $p_{\theta^*}(\mathbf{x}|\mathbf{z})$
- VI approximation of parametric posterior $p_{\theta^*}(\mathbf{z}|\mathbf{x})$ by a simpler encoder $q_\phi(\mathbf{z}|\mathbf{x})$

Auto-encoding Variational Bayes: contributions

- Parameter estimation θ of the data generative model (decoder) $p_{\theta^*}(\mathbf{x}|\mathbf{z})$
- VI approximation of parametric posterior $p_{\theta^*}(\mathbf{z}|\mathbf{x})$ by a simpler encoder $q_\phi(\mathbf{z}|\mathbf{x})$
- Joint learning of encoder parameters ϕ and decoder parameters θ

Auto-encoding Variational Bayes: contributions

- Parameter estimation θ of the data generative model (decoder) $p_{\theta^*}(\mathbf{x}|\mathbf{z})$
- VI approximation of parametric posterior $p_{\theta^*}(\mathbf{z}|\mathbf{x})$ by a simpler encoder $q_\phi(\mathbf{z}|\mathbf{x})$
- Joint learning of encoder parameters ϕ and decoder parameters θ
- No independence assumptions on latent variables in q

Auto-encoding Variational Bayes: contributions

- Parameter estimation θ of the data generative model (decoder) $p_{\theta^*}(\mathbf{x}|\mathbf{z})$
- VI approximation of parametric posterior $p_{\theta^*}(\mathbf{z}|\mathbf{x})$ by a simpler encoder $q_\phi(\mathbf{z}|\mathbf{x})$
- Joint learning of encoder parameters ϕ and decoder parameters θ
- No independence assumptions on latent variables in q
- VI approximation of parametric evidence $p_{\theta^*}(\mathbf{x})$

Auto-encoding Variational Bayes: contributions

- Parameter estimation θ of the data generative model (decoder) $p_{\theta^*}(\mathbf{x}|\mathbf{z})$
- VI approximation of parametric posterior $p_{\theta^*}(\mathbf{z}|\mathbf{x})$ by a simpler encoder $q_\phi(\mathbf{z}|\mathbf{x})$
- Joint learning of encoder parameters ϕ and decoder parameters θ
- No independence assumptions on latent variables in q
- VI approximation of parametric evidence $p_{\theta^*}(\mathbf{x})$
- Usage of a reparametrization trick explained by Luc Devroye in 1986

Auto-encoding Variational Bayes: ELBO maximization

$$\text{ELBO}(q, \theta, \phi) = \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}, \mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})]$$

Auto-encoding Variational Bayes: ELBO maximization

$$\text{ELBO}(q, \theta, \phi) = \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}, \mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})]$$

Problem: typical Monte-Carlo gradient estimator (like in policy gradient) with samples $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ has very high variance

Auto-encoding Variational Bayes: ELBO maximization

$$\text{ELBO}(q, \theta, \phi) = \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}, \mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})]$$

Problem: typical Monte-Carlo gradient estimator (like in policy gradient) with samples $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ has very high variance

(Solution?) reparametrization trick: instead of sampling $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ express $\mathbf{z} = g_\phi(\epsilon, \mathbf{x})$ where g is deterministic and only ϵ is stochastic.

$$\begin{aligned} \epsilon &\sim p(\epsilon) \\ \mathbf{z} &= g_\phi(\epsilon, \mathbf{x}) \end{aligned}$$

Auto-encoding Variational Bayes: ELBO maximization

$\text{ELBO}(q, \theta, \phi) = \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}, \mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})]$

Problem: typical Monte-Carlo gradient estimator (like in policy gradient) with samples $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ has very high variance

(Solution?) reparametrization trick: instead of sampling $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ express $\mathbf{z} = g_\phi(\epsilon, \mathbf{x})$ where g is deterministic and only ϵ is stochastic.

$$\begin{aligned} \epsilon &\sim p(\epsilon) \\ \mathbf{z} &= g_\phi(\epsilon, \mathbf{x}) \end{aligned}$$

Stochastic gradient ascent with minibatches to maximize ELBO w.r.t ϕ and θ

Auto-encoding Variational Bayes: VAE

$$\text{ELBO}(q, \theta, \phi) = \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}, \mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})]$$

Auto-encoding Variational Bayes: VAE

$$
\begin{aligned}
\mathrm{ELBO}(q, \theta, \phi) &= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}, \mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})]
\end{aligned}
$$

Auto-encoding Variational Bayes: VAE

$$
\begin{aligned}
\text{ELBO}(q, \theta, \phi) &= \mathbb{E}_{q_\phi}[\text{log} p_\theta(\mathbf{z}, \mathbf{x}) - \text{log} q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\text{log} p_\theta(\mathbf{x}|\mathbf{z}) + \text{log} p_\theta(\mathbf{z}) - \text{log} q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\text{log} p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{q_\phi}[\text{log} p_\theta(\mathbf{z}) + \text{log} q_\phi(\mathbf{z}|\mathbf{x})]
\end{aligned}
$$

Auto-encoding Variational Bayes: VAE

$$
\begin{aligned}
\mathsf{ELBO}(q,\theta,\phi) &= \mathbb{E}_{q_\phi}[\mathsf{log} p_\theta(\mathbf{z},\mathbf{x}) - \mathsf{log} q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\mathsf{log} p_\theta(\mathbf{x}|\mathbf{z}) + \mathsf{log} p_\theta(\mathbf{z}) - \mathsf{log} q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\mathsf{log} p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{q_\phi}[\mathsf{log} p_\theta(\mathbf{z}) + \mathsf{log} q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\mathsf{log} p_\theta(\mathbf{x}|\mathbf{z})] - \mathsf{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \;||\; p_\theta(\mathbf{z}))
\end{aligned}
$$

Auto-encoding Variational Bayes: VAE

$$\begin{aligned}
\mathrm{ELBO}(q,\theta,\phi) &= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z},\mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}) + \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \mid\mid p_\theta(\mathbf{z})) \\
&= -\text{decoding error} - \text{encoding regularization error}
\end{aligned}$$

Auto-encoding Variational Bayes: VAE

$$
\begin{aligned}
\text{ELBO}(q,\theta,\phi) &= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z},\mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}) + \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \mid\mid p_\theta(\mathbf{z})) \\
&= -\text{decoding error} - \text{encoding regularization error}
\end{aligned}
$$

- Joint optimization of ELBO over ϕ and θ implies training an autoencoder, the *variational autoencoder*

Auto-encoding Variational Bayes: VAE

$$\begin{aligned}
\text{ELBO}(q,\theta,\phi) &= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z},\mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}) + \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \mid\mid p_\theta(\mathbf{z})) \\
&= -\text{decoding error} - \text{encoding regularization error}
\end{aligned}$$

- Joint optimization of ELBO over ϕ and θ implies training an autoencoder, the *variational autoencoder*
- They assume $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1^2 \boldsymbol{I})$, $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\sigma}_2^2 \boldsymbol{I})$, and $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \boldsymbol{I})$

Auto-encoding Variational Bayes: VAE

$$\begin{aligned}
\text{ELBO}(q, \theta, \phi) &= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}, \mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{z}) + \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \mid\mid p_\theta(\mathbf{z})) \\
&= -\text{decoding error} - \text{encoding regularization error}
\end{aligned}$$

- Joint optimization of ELBO over ϕ and θ implies training an autoencoder, the *variational autoencoder*
- They assume $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1^2 \boldsymbol{I})$, $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\sigma}_2^2 \boldsymbol{I})$, and $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \boldsymbol{I})$
- They use multi-layer perceptrons to represent $\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1 = \text{MLP}_1(\mathbf{x}, \mathbf{z})$, $\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2 = \text{MLP}_2(\mathbf{x}, \mathbf{z})$

MNIST results

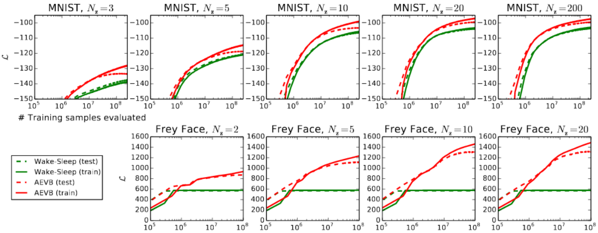
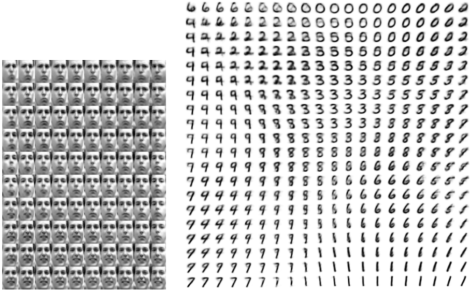


Figure 2: Comparison of our AEVB method to the wake-sleep algorithm, in terms of optimizing the lower bound, for different dimensionality of latent space ($N_\mathbf{z}$). Our method converged considerably faster and reached a better solution in all experiments. Interestingly enough, more latent variables does not result in more overfitting, which is explained by the regularizing effect of the lower bound. Vertical axis: the estimated average variational lower bound per datapoint. The estimator variance was small (< 1) and omitted. Horizontal axis: amount of training points evaluated. Computation took around 20-40 minutes per million training samples with a Intel Xeon CPU running at an effective 40 GFLOPS.

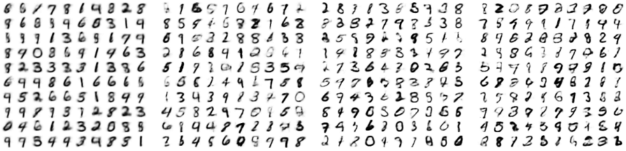
Frey Faces results



(a) Learned Frey Face manifold (b) Learned MNIST manifold

Figure 4: Visualisations of learned data manifold for generative models with two-dimensional latent space, learned with AEVB. Since the prior of the latent space is Gaussian, linearly spaced coordinates on the unit square were transformed through the inverse CDF of the Gaussian to produce values of the latent variables $\mathbf{z}$. For each of these values $\mathbf{z}$, we plotted the corresponding generative $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ with the learned parameters $\boldsymbol{\theta}$.

Dimension of latent variables



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.