Auto-Encoding Variational Bayes

By Kingma and Welling

The paper question ?

How can we perform efficient inference and learning in directed probabilistic models with presence of 2 challenges :

1. Continuous latent variables with intractable posterior distributions.
2. Large datasets.

Realistic Example

Recommendation Systems with Directed Probabilistic Models:

A streaming platform (e.g., Netflix or Spotify) aims to recommend personalized content to users. This involves:

1. Large-scale data (millions of users and items).
2. Continuous latent variables (user preferences and item attributes).
3. Intractable posterior distributions due to complex interactions between users and items.

Solution

Using The variational Bayesian (VB) approach involves the optimization of an approximation to the intractable posterior.

Drawbacks :

It requires analytical solutions of expectations w.r.t. the approximate posterior, which are also intractable.

Novel solution

Solution

Reparameterization of the variational lower bound → **A simple estimator**

Use:

SGVB (Stochastic Gradient Variational Bayes) estimator can be used for efficient approximate posterior

Auto Encoding VB (AEVB) algorithm.

Case of use :

i.i.d. dataset and continuous latent variables per datapoint

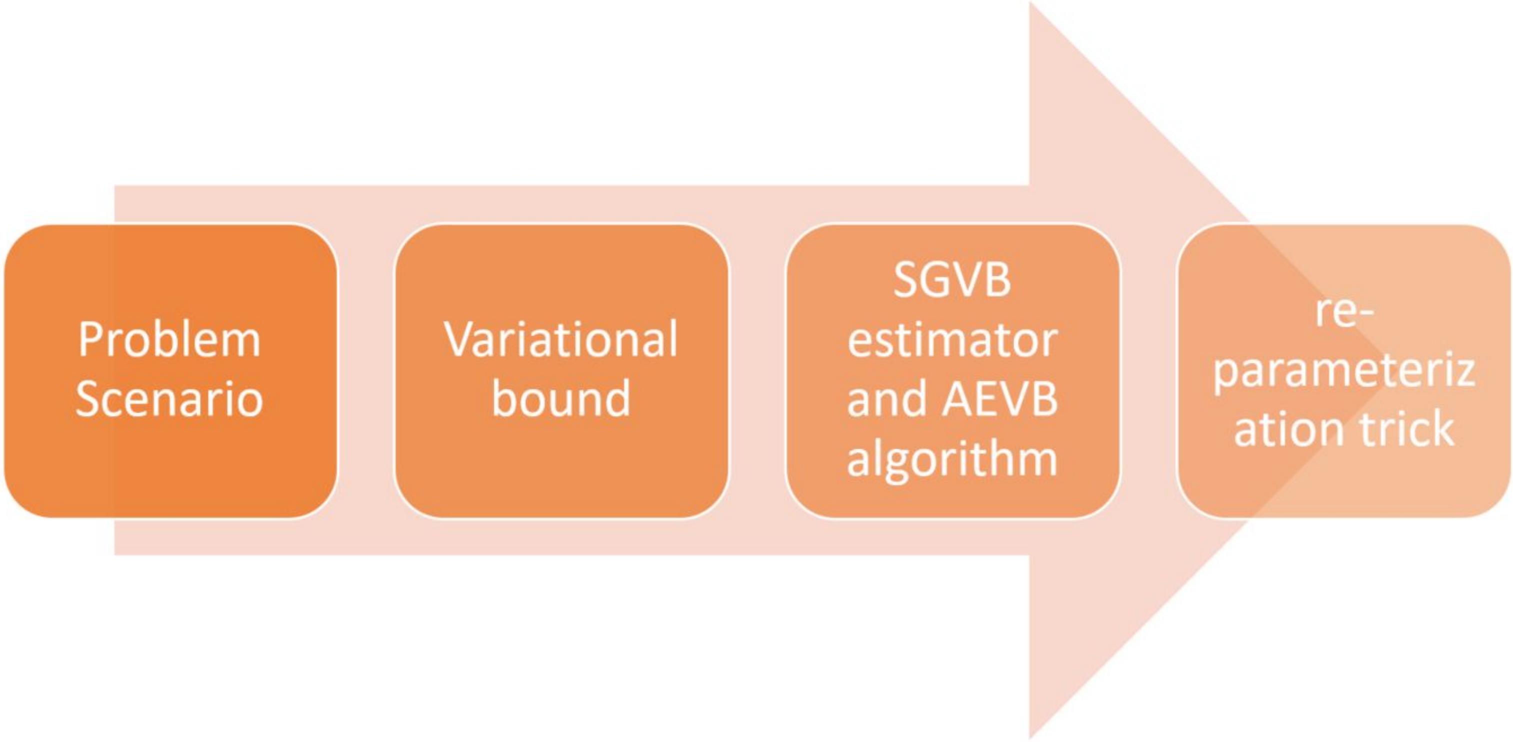
Aim :

Perform very efficient approximate posterior inference using simple ancestral sampling.

Tools:

Using the SGVB estimator to optimize a recognition model.

Methodology



Problem Scenario

$$\underbrace{p_\theta(x, z)}_{} = \underbrace{p_\theta(z)}_{\text{prior over } \mathbf{z}} \underbrace{p_\theta(x|z)}_{\text{likelihood}}$$

latent variable: z

observed data: x

- **Intractability**
- **Large datasets**
- **Difficulty computing posteriors**

Approximate the posterior $p_\theta(z|x)$ using a **variational** distribution $q_\phi(z|x)$ parameterized by ϕ

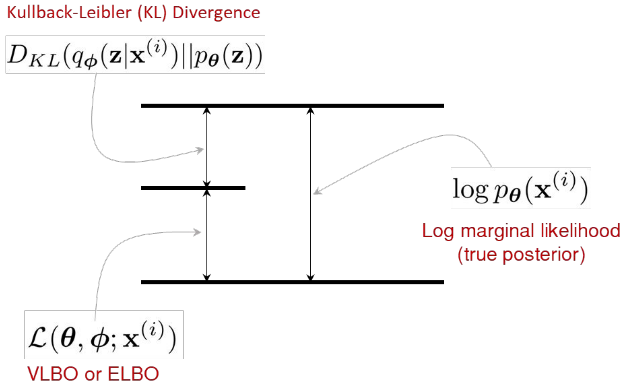
$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] + D_{\mathrm{KL}}(q_\phi(z|x) \| p_\theta(z|x))$$

Variational Lower Bound- VLBO (1)

- $$\underbrace{\log p_\theta(\mathbf{x}^{(i)})}_{\text{marginal likelihood of all data}} = \underbrace{D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z}|\mathbf{x}^{(i)}))}_{\text{KL divergence between the approximate posterior and the true posterior}} + \underbrace{\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})}_{\text{The variational lower bound (ELBO)}}$$

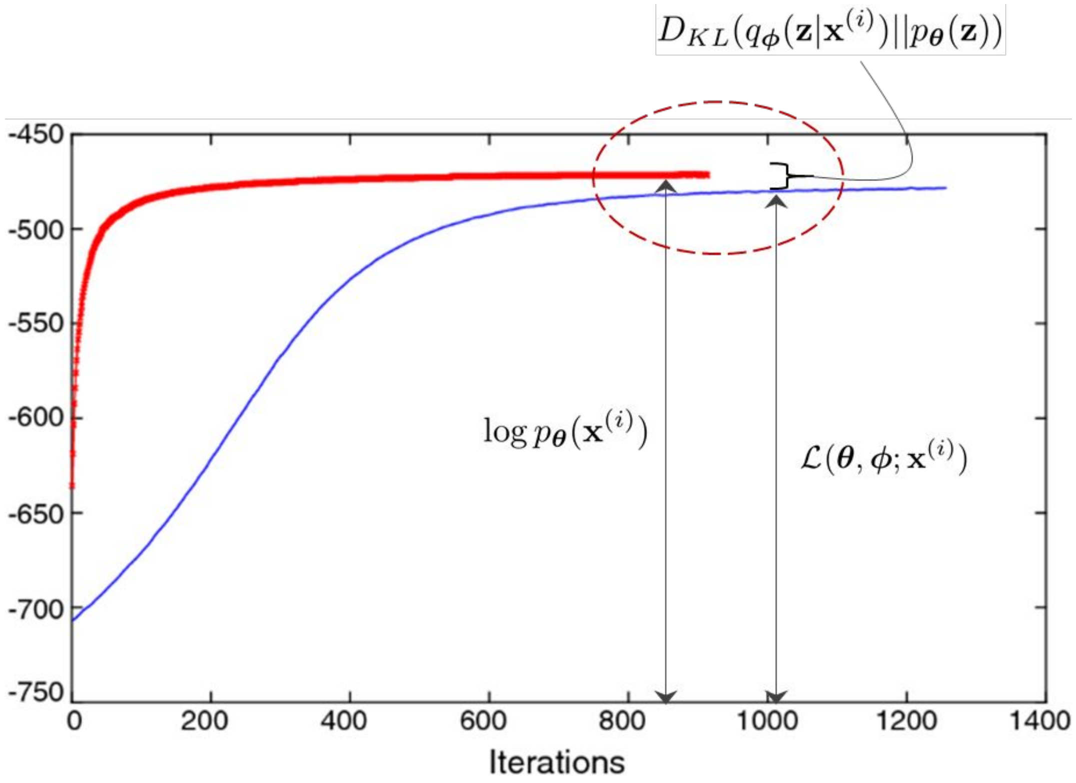
- Since the KL divergence is non-negative:

$$\log p_\theta(\mathbf{x}^{(i)}) \geq \underbrace{\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})}_{\text{Lower bound}}$$

- 

- **<u>Note</u>: VLBO aka ELBO**

Variational Lower Bound - VLBO (2)



<u>Goal:</u>
Learn **ϕ** and **θ** by maximizing the Variational Lower Bound (VLBO), which approximates the log-likelihood of the observed data.

- **VLBO Maximization**
- **Minimize KL - Divergence**

The SGVB estimator and AEVB algorithm

reparameterize the random variable $\widetilde{\mathbf{z}} \sim q_\phi(\mathbf{z}|\mathbf{x})$ using

$$\widetilde{\mathbf{z}} = g_\phi(\boldsymbol{\epsilon}, \mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

Stochastic Gradient Variational Bayes (SGVB) estimator

$$\widetilde{\mathcal{L}}^A(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \frac{1}{L}\sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z}^{(i,l)}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}^{(i,l)}|\mathbf{x}^{(i)})$$

$$\text{where} \quad \mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

Algorithm to compute the stochastic gradients.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (Gradients of minibatch estimator (8))
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients $\mathbf{g}$ (e.g. SGD or Adagrad [DHS10])
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

The reparameterization trick

$$\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}) \qquad \mathbf{z} = g_\phi(\epsilon, \mathbf{x})$$

Choosing approaches :

1. **Tractable Inverse CDF.**
2. **Analogous to the Gaussian.**
3. **Composition: It is often possible to express random variables as different transformations of auxiliary variables.**

Example: Variational Autoencoder

Prior over latent variables: Defined as a centered isotropic multivariate Gaussian, $p_\theta(z) = \mathcal{N}(z; 0, I)$.

Data likelihood: $p_\theta(x|z)$ is modeled as a multivariate Gaussian (for real-valued data) or Bernoulli (for binary data). The parameters of this distribution are generated by a fully connected neural network (MLP) with a single hidden layer.

True posterior: $p_\theta(z|x)$ is generally intractable.

Variational posterior approximation: The true posterior $p_\theta(z|x)$ is approximated by $q_\phi(z|x)$, which is modeled as a multivariate Gaussian with a diagonal covariance matrix:

$$q_\phi(z|x^{(i)}) = \mathcal{N}(z; \mu^{(i)}, \sigma^{2(i)} I).$$

Here, $\mu^{(i)}$ (mean) and $\sigma^{(i)}$ (standard deviation) are outputs of an encoding MLP, parameterized by ϕ, and depend on the input $x^{(i)}$.

Example: Variational Autoencoder

Sampling from the posterior: The latent variable $z^{(i,l)}$ is sampled from the approximate posterior $q_\phi(z|x^{(i)})$ using the reparameterization trick:

$$z^{(i,l)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(l)},$$

where $\epsilon^{(l)} \sim \mathcal{N}(0, I)$ and $\odot$ represents element-wise multiplication.

KL divergence and loss function: Since both $p_\theta(z)$ (prior) and $q_\phi(z|x)$ (variational posterior) are Gaussian, the KL divergence can be computed analytically. The resulting loss estimator for the model and a single data point $x^{(i)}$ is:

$$\mathcal{L}(\theta, \phi; x^{(i)}) \simeq \frac{1}{2} \sum_{j=1}^{J} \left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right) + \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(x^{(i)}|z^{(i,l)}).$$

Experiment Setup - Datasets used

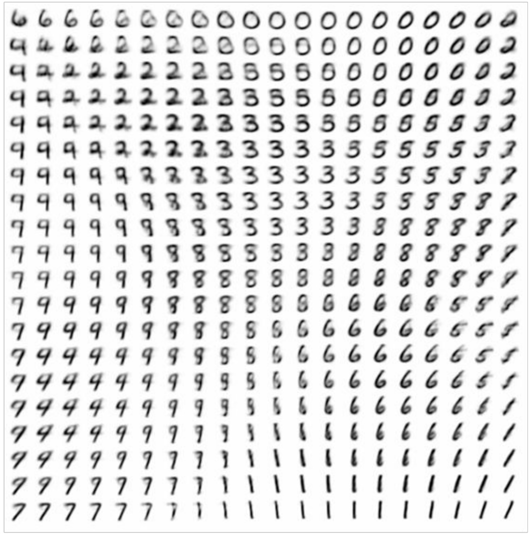
Description:

- Frey's facial expressions.
- **1965 images**
- Each **28 × 20 pixels**

Applications:

- **Dimensionality reduction** (PCA, t-SNE, or UMAP).
- **Generative models** such (VAEs) and GANs.

(a) Frey Faces Dataset



Data Description:

- **70,000 images** of handwritten digits.
- Each **28 × 28 pixel grid**, resulting in **784 features** per image.
- Pixel values range: **0 to 255**

Dataset Composition:

- **Training Set**: 60k images.
- **Test Set**: 10k images.

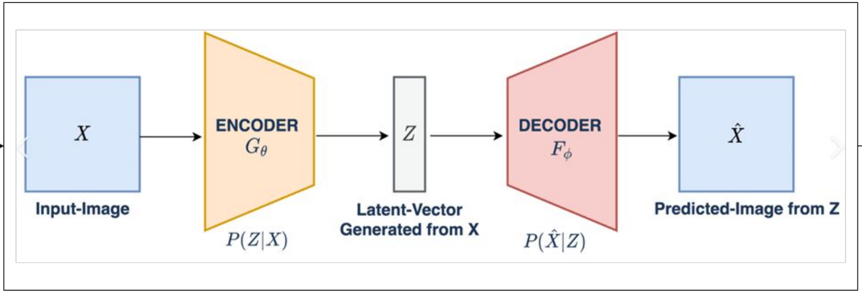
(b) MNIST Dataset

Experiments - Implementation pipeline

Data preparation

MNIST

Frey Face



Algo. Comparison

1. VLBO:
2. Marginal likelihood:

VAE approximates latent features based on observed features

VAE reconstructs the observed features from latent variables

Experiments - Results (1): Likelihood lower bound

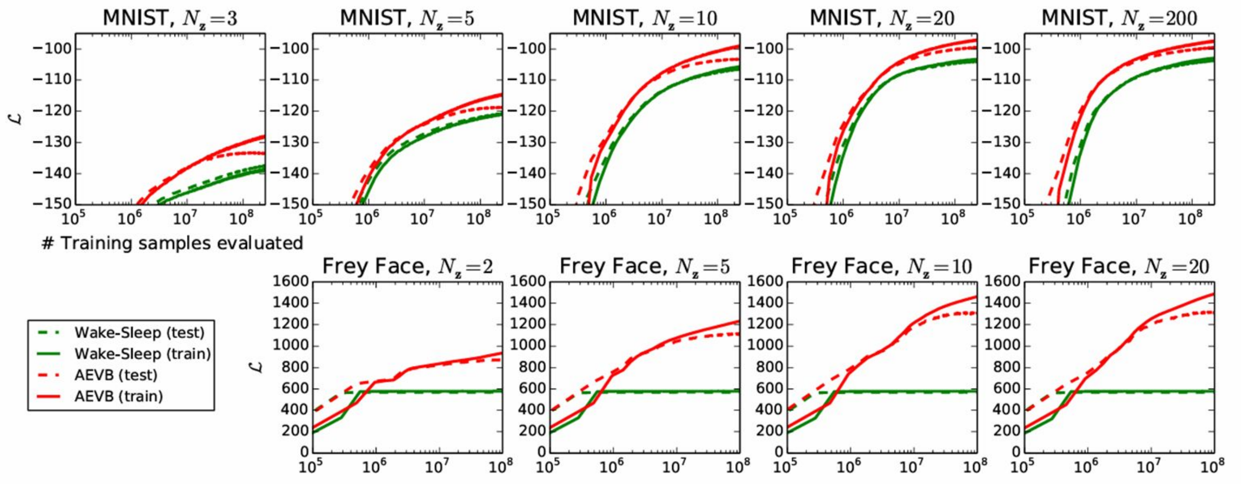


Figure 2: Comparison of our AEVB method to the wake-sleep algorithm, in terms of optimizing the lower bound, for different dimensionality of latent space(Nz).

In all Experiments

- ★ Faster convergence
- ★ Reached better solution
- ★ More latent variables did not result in more overfitting

Experiments - Results (2): Marginal Likelihood

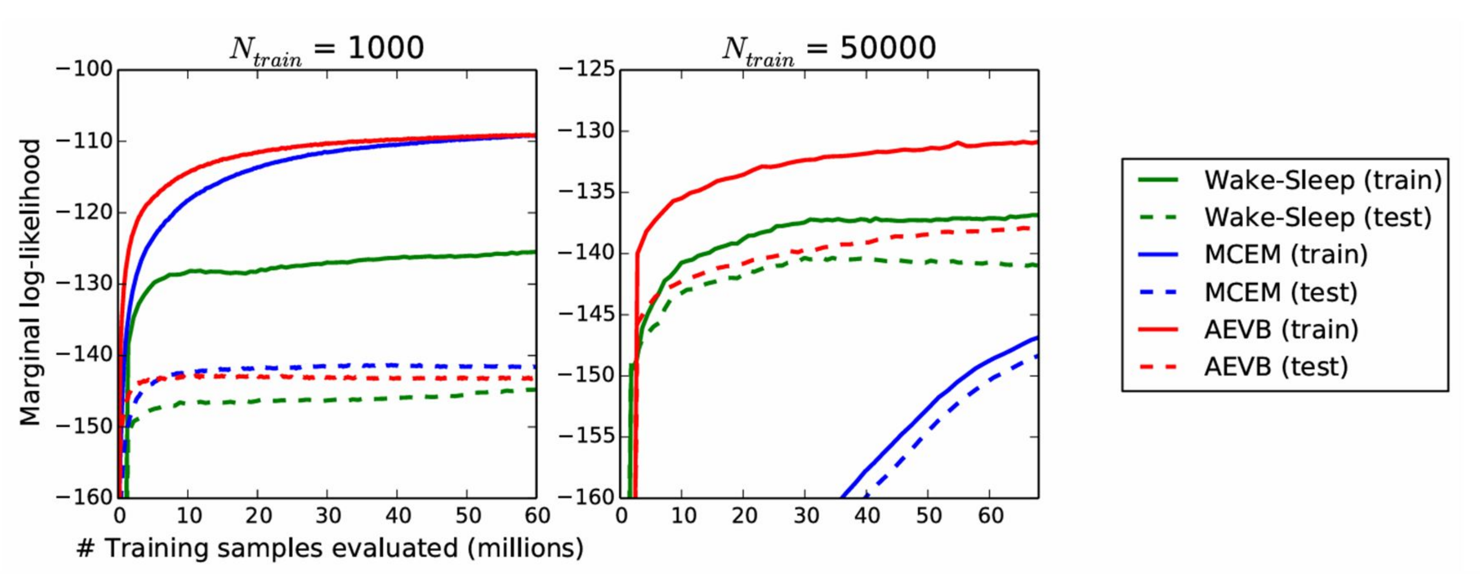


Figure 3: Comparison of AEVB to the wake-sleep algorithm and Monte Carlo EM, in terms of the estimated marginal likelihood, for a different number of training points.

Summary

Key Contributions:

★ Efficient optimization of the Evidence Lower Bound (ELBO) using the **reparameterization trick**.

★ Enabled scalable inference for high-dimensional latent variable models.

★ Paved the way for **Variational Autoencoders (VAEs)**, a widely-used generative model.

Strengths:

★ Effective **complex posterior distributions** handling

★ Flexible and scalable framework for **unsupervised learning**.

Limitations:

★ Assumes a factorized Gaussian posterior (**simplifying assumption**).

★ Performance can degrade if true posteriors deviate significantly from the assumed form.

Thanks for your attention!