Mathematical Foundations of Deep Neural Networks

§ 1. Optimization Problem

§ 1. Optimization Problem

Definition of Optimization Problem

Definition 1.1: Optimization Problem

In an **optimization problem**, we minimize or maximize a function value, possibly subject to constraints.

$$
\begin{array}{ll}
\underset{\theta \in \mathbb{R}^p}{\text{minimize}} & f(\theta) \\
\text{subject to} & h_1(\theta) = 0, \\
\cdots & , \; h_m(\theta) = 0, \\
 & g_1(\theta) \leq 0, \\
\cdots & , \; g_n(\theta) \leq 0
\end{array}
$$

- Decision variable: θ
- Objective function: f
- Equality constraint: $h_i(\theta) = 0$ for $i = 1, \ldots, m$
- Inequality constraint: $g_j(\theta) \leq 0$ for $j = 1, \ldots, n$

In machine learning (ML), we often minimize a "loss", but sometimes we maximize the "likelihood". In any case, minimization and maximization are equivalent since

$$\text{maximize } f(\theta) \quad \Leftrightarrow \quad \text{minimize } -f(\theta).$$

Definition 1.2: Feasible Point and Constraints

$\theta \in \mathbb{R}^p$ is a **feasible point** if it satisfies all constraints:

$$\begin{aligned} h_1(\theta) &= 0 & g_1(\theta) &\leq 0 \\ &\vdots & &\vdots \\ h_m(\theta) &= 0 & g_n(\theta) &\leq 0 \end{aligned}$$

Optimization problem is **infeasible** if there is no feasible point.
An optimization problem with no constraint is called an **unconstrained optimization problem**. Optimization problems with constraints is called a **constrained optimization problem**.

Definition 1.3: Optimal Value and Solution

Optimal value of an optimization problem is

$$p^\star = \inf \{ f(\theta) \mid \theta \in \mathbb{R}^n, \theta \text{ feasible } \}$$

- $p^\star = \infty$ if problem is infeasible
- $p^\star = -\infty$ is possible
- In ML, it is often a priori clear that $0 \leq p^\star < \infty$.

If $f(\theta^\star) = p^\star$, we say $\theta^\star$ is a **solution** or $\theta^\star$ is **optimal**.
A solution may or may not exist, and a solution may or may not be unique.

§ 1. Optimization Problem

Examples of Optimization Problem

Example 1.4: Curve Fitting

Consider setup with data $X_1, \ldots, X_N$ and corresponding labels $Y_1, \ldots, Y_N$ satisfying the relationship

$$Y_i = f_\star(X_i) + \text{ error}$$

for $i = 1, \ldots, N$. Hopefully, "error" is small. True function $f_\star$ is unknown. Goal is to find a function (curve) f such that $f \approx f_\star$.

Example 1.5: Least-Squares Minimization

- **Problem**

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \quad \frac{1}{2}\|X\theta - Y\|^2$$

where $X \in \mathbb{R}^{N \times p}$ and $Y \in \mathbb{R}^N$. Equivalent to

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \frac{1}{2} \sum_{i=1}^{N} \left(X_i^\top \theta - Y_i \right)^2$$

where $X = \begin{bmatrix} X_1^\top \\ \vdots \\ X_N^\top \end{bmatrix}$ and $Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_N \end{bmatrix}$.

- **Solution**

To solve

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \frac{1}{2}\|X\theta - Y\|^2$$

take gradient and set it to 0

Concept 1.6: Least squares is an instance of curve fitting.

Define $f_\theta(x) = x^\top \theta$. Then LS becomes

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \frac{1}{2} \sum_{i=1}^{N} \left(f_\theta\left(X_i\right) - Y_i\right)^2$$

and the solution hopefully satisfies

$$Y_i = f_\theta\left(X_i\right) + \text{ small.}$$

Since X_i and Y_i is assumed to satisfy

$$Y_i = f_\star\left(X_i\right) + \text{ error}$$

we are searching over linear functions (linear curves) f_θ that best fit (approximate) $f_\star$.

§ 1. Optimization Problem

Local and Global Minimum

Definition 1.7: Local vs Global Minima

$\theta^\star$ is a **local minimum** if $f(\theta) \geq f(\theta^\star)$ for all feasible θ within a small neighborhood.
$\theta^\star$ is a **global minimum** if $f(\theta) \geq f(\theta^\star)$ for all feasible θ.

../../assets/1.1.jpg

In the worst case, finding the global minimum of an optimization problem

§ 11. Variational Autoencoders

Prerequisites : Ch A. Appendix - Basics of Monte Carlo

Concept 11.1: Math Review

Let A and B be probabilistic events. Assume A has nonzero probability. **Conditional probability** satisfies

$$\mathbb{P}(B \mid A)\mathbb{P}(A) = \mathbb{P}(A \cap B)$$

Bayes' theorem is an application of conditional probability:

$$\mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \mid B)\mathbb{P}(B)}{\mathbb{P}(A)}$$

Concept 11.2: Math Review

Let $X \in \mathbb{R}^m$ and $Z \in \mathbb{R}^n$ be continuous random variables with joint density $p(x, z)$.
The marginal densities are defined by

$$p_X(x) = \int_{\mathbb{R}^n} p(x, z)dz, \quad p_Z(z) = \int_{\mathbb{R}^m} p(x, z)dx$$

The conditional density function $p(z \mid x)$ has the following properties

$$\mathbb{P}(Z \in S \mid X = x) = \int_S p(z \mid x)dz$$

$$p(z \mid x)p_X(x) = p(x, z), \quad p(z \mid x) = \frac{p(x \mid z)p_Z(z)}{p_X(x)}$$

Concept 11.3: Introduction for Variational Autoencoders (VAE)

Key idea of **VAE**:

- **Latent variable model** with conditional probability distribution represented by $p_\theta(x \mid z)$.
- Efficiently estimate $p_\theta(x) = \mathbb{E}_{Z \sim p_Z}[p_\theta(x \mid Z)]$ by **importance sampling** with $Z \sim q_\phi(z \mid x)$.

We can interpret $q_\phi(z \mid x)$ as an encoder and $p_\theta(x \mid z)$ as a decoder. VAEs differ from autoencoders as follows:

- Derivations (latent variable model vs. dimensionality reduction)
- VAE regularizes/controls latent distribution, while AE does not.

../../assets/11.1.png

§ 11. Variational Autoencoders

Latent Variable Model

- Assumption on data $X_1, \dots, X_N$
  Assumes there is an underlying latent variable Z representing the "essential structure" of the data and an observable variable X which generation is conditioned on Z. Implicitly assumes the conditional randomness of $X \sim p_{X|Z}$ is significantly smaller than the overall randomness $X \sim p_X$.
- Example
  X is a cat picture. Z encodes information about the body position, fur color, and facial expression of a cat. Latent variable Z encodes the overall content of the image, but X does contain details not specified in Z.

Definition 11.4: Latent Variable Model

VAEs implements a **latent variable model** with a NN that generates X given Z. More precisely, NN is a deterministic function that outputs the conditional distribution $p_\theta(x \mid Z)$, and X is randomly generated according to this distribution. This structure may effectively learn the latent structure from data if the assumption on data is accurate.

0.4pt1pt 2pt

../../assets/11.2.png

Sampling process:

$$X \sim p_\theta(x \mid Z), \quad Z \sim p_Z(z)$$

Usually p_Z is a Gaussian (fixed) and $p_\theta(x \mid z)$ is a NN parameterized by θ.
Evaluating density (likelihood):

Example 11.5: Example Latent Variable Model

Mixture of 3 Gaussians in $\mathbb{R}^2$, uniform prior over components. (We can make the mixture weights a trainable parameter.)

$$p_Z(Z = A) = p_Z(Z = B) = p_Z(Z = C) = \frac{1}{3}$$

$$p_\theta(x \mid Z = k) = \frac{1}{2\pi \left|\Sigma_k\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(x - \mu_k\right)^\top \Sigma_k^{-1}\left(x - \mu_k\right)\right)$$

Training objective:

$$\begin{aligned}\underset{\mu,\Sigma}{\text{maximize}} \sum_{i=1}^{N} \log p_\theta\left(X_i\right) = \underset{\mu,\Sigma}{\text{maximize}} \sum_{i=1}^{N} \log \Bigg[&\frac{1}{3}\frac{1}{2\pi\left|\Sigma_A\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(X_i - \mu_A\right)^\top\right. \\ &+ \frac{1}{3}\frac{1}{2\pi\left|\Sigma_B\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(X_i - \mu_B\right)\right. \\ &+ \frac{1}{3}\frac{1}{2\pi\left|\Sigma_C\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(X_i - \mu_C\right)\right.\end{aligned}$$

§ 11. Variational Autoencoders

Training Latent Variable Model with Importance Sampling

From now on, we will focus on **HOW** to train latent variable model with MLE,

$$\underset{\theta \in \Theta}{\text{maximize}} \sum_{i=1}^{N} \log p_\theta(X_i) = \underset{\theta \in \Theta}{\text{maximize}} \sum_{i=1}^{N} \log \mathbb{E}_{Z \sim p_Z}\left[p_\theta(X_i \mid Z)\right]$$

Concept 11.6: VAE Outline

Outline of variational autoencoder (VAE):

1. (Choice 1) Approximate intractable objective with a single Z sample

$$\sum_{i=1}^{N} \log \mathbb{E}_{Z\sim p_Z}\left[p_\theta\left(X_i \mid Z\right)\right] \approx \sum_{i=1}^{N} \log p_\theta\left(X_i \mid Z_i\right), \quad Z_i \sim p_Z$$

2. (Choice 2) Improve accuracy of approximation by sampling Z_i with importance sampling

$$\sum_{i=1}^{N} \log \mathbb{E}_{Z\sim p_Z}\left[p_\theta\left(X_i \mid Z\right)\right] \approx \sum_{i=1}^{N} \log \frac{p_\theta\left(X_i \mid Z_i\right) p_Z\left(Z_i\right)}{q_i\left(Z_i\right)}, \quad Z_i \sim q_i$$

3. Optimize approximate objective with SGD.

(D. Kingma and M. Welling, VAE: Auto-encoding variational Bayes, ICLR, 2014.)

Concept 11.7: IWAE Outline

Importance weighted autoencoders (IWAE) approximates intractable with K samples of Z :

$$\sum_{i=1}^{N} \log \mathbb{E}_{Z\sim p_Z}\left[p_\theta\left(X_i \mid Z\right)\right] \approx \sum_{i=1}^{N} \log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta\left(X_i \mid Z_{i,k}\right) p_Z\left(Z_{i,k}\right)}{q_i\left(Z_{i,k}\right)}, \quad Z_{i,1},\ldots$$

(Y. Burda, R. Grosse, and R. Salakhutdinov, Importance weighted autoencoders, ICLR, 2016.)

Concept 11.8: Why does VAE need IS?

Among the two choices given in Concept 11.6, VAEs improve the accuracy of latent variable model with IS (Choice 2).
Sampling $Z_i \sim p_Z$ (Choice 1) results in a high-variance estimator:

$$\mathbb{E}_{Z \sim p_Z}\left[p_\theta\left(X_i \mid Z\right)\right] \approx p_\theta\left(X_i \mid Z_i\right),$$

In the Gaussian mixture example (Example 11.5), only $1/3$ of the Z samples meaningfully contribute to the estimate. More specifically, if X_i is near μ_A but is far from μ_B and μ_C, then $p_\theta\left(X_i \mid Z = A\right) \gg 0$ but $p_\theta\left(X_i \mid Z = B\right) \approx 0$ and $p_\theta\left(X_i \mid Z = C\right) \approx 0$.
The issue worsens as the observable and latent variable dimension increases.

0.4pt1pt 2pt

Concept 11.9: Naïve Approach

To improve estimation of $\mathbb{E}_{Z\sim p_Z}\left[p_\theta\left(X_i \mid Z\right)\right]$, consider importance sampling (IS) with sampling distribution $Z_i \sim q_i(z)$:

$$\mathbb{E}_{Z\sim p_Z}\left[p_\theta\left(X_i \mid Z\right)\right] \approx p_\theta\left(X_i \mid Z_i\right)\frac{p_Z\left(Z_i\right)}{q_i\left(Z_i\right)}$$

Optimal IS sampling distribution

$$q_i^\star(z) = \frac{p_\theta\left(X_i \mid z\right)p_Z(z)}{p_\theta\left(X_i\right)} = p_\theta\left(z \mid X_i\right)$$

To clarify, optimal sampling distribution depends on X_i. To clarify, $p_\theta\left(X_i\right)$ is the unkown normalizing factor so $p_\theta\left(z \mid X_i\right)$ is also unkown. We call $q_i^\star(z) = p_\theta\left(z \mid X_i\right)$ the true **posterior** distribution and we will soon consider the approximation $q_\phi(z \mid x) \approx p_\theta(z \mid x)$, which we call the **approximate posterior**.

0.4pt1pt 2pt

For each X_i, let $q_i(z)$ be the optimal approximate posterior dependent on X_i, and consider

Concept 11.10: Variational Approach and Amortized Inference

General principle of variational approach: We can't directly use the q we want. So, instead, we propose a parameterized distribution q_ϕ that we can work with easily (in this case, sample from easily), and find a parameter setting that makes it as good as possible.
Parametrization of VAE:

$$q_\phi(z \mid X_i) \approx q_i^\star(z) = p_\theta(z \mid X_i) \qquad \text{for all } i = 1, \ldots, N$$

Amortized inference: Train a neural network $q_\phi(\cdot \mid x)$ such that $q_\phi(\cdot \mid X_i)$ approximates the optimal $q_i(\cdot)$.

$$\underset{\phi \in \Phi}{\operatorname{minimize}} \sum_{i=1}^{N} D_{\mathrm{KL}}\left(q_\phi(\cdot \mid X_i) \| p_\theta(\cdot \mid X_i)\right)$$

Approximation $q_\phi(z \mid X_i) \approx p_\theta(z \mid X_i)$ is often less precise than that of individual inference $q_i(z) \approx p_\theta(z \mid X_i)$, but amortized inference is often significantly faster.

Concept 11.11: Encoder q_ϕ Optimization

In analogy with autoencoders, we call q_ϕ the **encoder**.
Optimization problem for encoder (derived from Concept 11.9) :

$$\begin{aligned}
&\underset{\phi\in\Phi}{\text{minimize}} \sum_{i=1}^{N} D_{\mathrm{KL}}\left(q_\phi\left(\cdot \mid X_i\right) \| p_\theta\left(\cdot \mid X_i\right)\right) \\
=&\underset{\phi\in\Phi}{\text{maximize}} \sum_{i=1}^{N} \mathbb{E}_{Z\sim q_\phi(z\mid X_i)}\left[\log\left(\frac{p_\theta\left(X_i \mid Z\right) p_Z(Z)}{q_\phi\left(Z \mid X_i\right)}\right)\right] + \text{ constant independe} \\
=&\underset{\phi\in\Phi}{\text{maximize}} \sum_{i=1}^{N} \mathbb{E}_{Z\sim q_\phi(z\mid X_i)}\left[\log p_\theta\left(X_i \mid Z\right)\right] - D_{\mathrm{KL}}\left(q_\phi\left(\cdot \mid X_i\right) \| p_Z(\cdot)\right)
\end{aligned}$$

Concept 11.12: Decoder p_θ Optimization

In analogy with autoencoders, we call p_θ the **decoder**. Perform approximate MLE (derived from IS, Choice 2 of Concept 11.6) :

$$
\begin{aligned}
&\underset{\theta\in\Theta}{\text{maximize}} \sum_{i=1}^{N} \log p_\theta(X_i) = \underset{\theta\in\Theta}{\text{maximize}} \sum_{i=1}^{N} \log \mathbb{E}_{Z\sim p_Z}\left[p_\theta(X_i \mid Z)\right] \\
&\overset{(a)}{\approx} \underset{\theta\in\Theta}{\text{maximize}} \sum_{i=1}^{N} \log\left(\frac{p_\theta(X_i \mid Z_i)\, p_Z(Z_i)}{q_\phi(Z_i \mid X_i)}\right), \quad Z_i \sim q_\phi(z \mid X_i) \\
&\overset{(b)}{\approx} \underset{\theta\in\Theta}{\text{maximize}} \sum_{i=1}^{N} \mathbb{E}_{Z\sim q_\phi(z\mid X_i)}\left[\log\left(\frac{p_\theta(X_i \mid Z)\, p_Z(Z)}{q_\phi(Z \mid X_i)}\right)\right] \\
&= \underset{\theta\in\Theta}{\text{maximize}} \sum_{i=1}^{N} \mathbb{E}_{Z\sim q_\phi(z\mid X_i)}\left[\log p_\theta(X_i \mid Z)\right] - D_{\mathrm{KL}}\left(q_\phi(\cdot \mid X_i)\,\|\, p_Z(\cdot)\right)
\end{aligned}
$$

The $\overset{(a)}{\approx}$ step replaces expectation inside the log with an estimate with Z_i.

The $\overset{(b)}{\approx}$ step replaces the random variable with the expectation. These

§ 11. Variational Autoencoders

Definition of VAE

- § 11. Variational Autoencoders
  - Latent Variable Model
  - Training Latent Variable Model with Importance Sampling
  - Definition of VAE
  - VAE Standard Instance
  - Training VAE
  - Researches

Definition 11.13: Variational Lower Bound (VLB)

The optimization objectives for the encoder (Concept 11.11) and decoder (Concept 11.12) are the same!
Simultaneously train p_θ and q_ϕ by solving

$$\underset{\theta\in\Theta,\phi\in\Phi}{\text{maximize}} \sum_{i=1}^{N} \underbrace{\mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\log p_\theta\left(X_i \mid Z\right)\right] - D_{\mathrm{KL}}\left(q_\phi\left(\cdot \mid X_i\right) \| p_Z(\cdot)\right)}_{\overset{\text{def}}{=} \mathrm{VLB}_{\theta,\phi}(X_i)}$$

We refer to the optimization objective as the **variational lower bound (VLB)** or **evidence lower bound (ELBO)** for reasons that will be explained soon (Concept 11.14).

Concept 11.14: How tight lower bound is the VLB?

How accurate is the approximation?

$$\underset{\theta\in\Theta}{\text{maximize}} \sum_{i=1}^{N} \log p_\theta\left(X_i\right) = \underset{\theta\in\Theta}{\text{maximize}} \sum_{i=1}^{N} \log \mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\frac{p_\theta\left(X_i \mid Z\right) p_Z(Z)}{q_\phi\left(Z \mid X_i\right)}\right]$$

$$\stackrel{?}{\approx} \underset{\theta\in\Theta,\phi\in\Phi}{\text{maximize}} \sum_{i=1}^{N} \mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\log\left(\frac{p_\theta\left(X_i \mid Z\right) p_Z(Z)}{q_\phi\left(Z \mid X_i\right)}\right)\right]$$

$$= \underset{\theta\in\Theta,\phi\in\Phi}{\text{maximize}} \sum_{i=1}^{N} \mathrm{VLB}_{\theta,\phi}\left(X_i\right)$$

This turns out that

$$\log p_\theta\left(X_i\right) \geq \mathrm{VLB}_{\theta,\phi}\left(X_i\right)$$

So we are maximizing a lower bound of the log likelihood. How large is the gap?

0.4pt1pt 2pt

Concept 11.15: VLB is tight if encoder is infinitely powerful.

If the encoder q_ϕ is powerful enough such that there is a $\phi^\star$ achieving

$$q_{\phi^\star}(\cdot \mid X_i) = p_\theta(\cdot \mid X_i)$$

or equivalently

$$D_{\mathrm{KL}}\left[q_{\phi^\star}(\cdot \mid X_i) \,\|\, p_\theta(\cdot \mid X_i)\right] = 0$$

Then

$$\underset{\theta \in \Theta}{\text{maximize}} \sum_{i=1}^{N} \log p_\theta(X_i) = \underset{\theta \in \Theta, \phi \in \Phi}{\text{maximize}} \sum_{i=1}^{N} \mathrm{VLB}_{\theta,\phi}(X_i)$$

Definition 11.16: Variational Autoencoder (VAE) Terminology

../../assets/11.4.png

- **Likelihood** : $p_\theta(x)$ (exact evaluation intractable)
- **Prior** : $p_Z(z)$
- **Conditional distribution (decoder)** : $p_\theta(x \mid z)$
- **True posterior** : $p_\theta(z \mid x)$ (exact evaluation intractable)

§ 11. Variational Autoencoders

VAE Standard Instance

Definition 11.17: VAE Standard Instance

A **standard VAE setup**:

$$
\begin{aligned}
p_Z &= \mathcal{N}(0, I) \\
q_\phi(z \mid x) &= \mathcal{N}\left(\mu_\phi(x), \Sigma_\phi(x)\right) \text{ with diagonal } \Sigma_\phi \\
p_\theta(x \mid z) &= \mathcal{N}\left(f_\theta(z), \sigma^2 I\right)
\end{aligned}
$$

$\mu_\phi(x), \Sigma^2_\phi(x)$, and $f_\theta(z)$ are deterministic NN.
0.4pt1pt 2pt
Using the following equation,

$$
\begin{aligned}
&D_{\mathrm{KL}}\left(\mathcal{N}\left(\mu_\phi(X), \Sigma_\phi(X)\right) \| \mathcal{N}(0, I)\right) \\
=&\frac{1}{2}\left(\operatorname{tr}\left(\Sigma_\phi(X)\right) + \left\|\mu_\phi(X)\right\|^2 - d - \log\det\left(\Sigma_\phi(X)\right)\right)
\end{aligned}
$$

the training objective

$$
\underset{\theta \in \Theta, \phi \in \Phi}{\operatorname{maximize}} \sum_{i=1}^{N} \mathbb{E}_{Z \sim q_\phi(z \mid X_i)}\left[\log p_\theta\left(X_i \mid Z\right)\right] - D_{\mathrm{KL}}\left(q_\phi\left(\cdot \mid X_i\right) \| p_Z(\cdot)\right)
$$

Concept 11.18: VAE Standard Instance with Reparameterization Trick

The standard instance of VAE

$$\underset{\theta\in\Theta,\phi\in\Phi}{\text{minimize}} \sum_{i=1}^{N} \frac{1}{\sigma^2}\mathbb{E}_{Z\sim\mathcal{N}(\mu_\phi(X_i),\Sigma_\phi(X_i))} \|X_i - f_\theta(Z)\|^2 + \mathrm{tr}\left(\Sigma_\phi\left(X_i\right)\right) + \|\mu_\phi\left(X_i\right)\|^2 -$$

can be equivalently written with the reparameterization trick

$$\underset{\theta\in\Theta,\phi\in\Phi}{\text{minimize}} \sum_{i=1}^{N} \frac{1}{\sigma^2}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)} \left\|X_i - f_\theta\left(\mu_\phi\left(X_i\right) + \Sigma_\phi^{1/2}\left(X_i\right)\varepsilon\right)\right\|^2 + \mathrm{tr}\left(\Sigma_\phi\left(X_i\right)\right) + \|$$

where $\Sigma_\phi^{1/2}$ is diagonal with $\sqrt{\cdot}$ of the diagonal elements of Σ_ϕ.
(Remember, Σ_ϕ is diagonal.)
To clarify $Z \stackrel{\mathcal{D}}{=} \mu_\phi\left(X_i\right) + \Sigma_\phi^{1/2}\left(X_i\right)\varepsilon$, where $\stackrel{\mathcal{D}}{=}$ denotes equality in distribution.
We now have an objective amenable to stochastic optimization.

Concept 11.19: VAE Standard Instance Architecture

- Training (Without reparameterization trick)

../../assets/11.5.png

Concept 11.20: Why variational

VAE loss (VLB) contains a reconstruction loss resembling that of an autoencoder.

$$
\begin{aligned}
\mathrm{VLB}_{\theta,\phi}(X_i) &= \mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\log p_\theta(X_i \mid Z)\right] - D_{\mathrm{KL}}\left(q_\phi(\cdot \mid X_i)\,\|\,p_Z(\cdot)\right) \\
&= -\frac{1}{2\sigma^2}\mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\|X_i - f_\theta(Z)\|^2\right] - D_{\mathrm{KL}}\left(q_\phi(\cdot \mid X_i)\,\|\,p_Z(\cdot)\right) \\
&= \underbrace{-\frac{1}{2\sigma^2}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}\left\|X_i - f_\theta\left(\mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)\,\varepsilon\right)\right\|^2}_{\text{Reconstruction loss}} - \underbrace{D_{\mathrm{KL}}\left(q_\phi\right.}_{\text{Reg}}
\end{aligned}
$$

VLB also contains a regularization term on the output of the encoder, which is not present in standard autoencoder losses.
The choice of σ determines the relative weight between the reconstruction loss and the regularization.

§ 11. Variational Autoencoders

Training VAE

Concept 11.21: Training VAE with RT

To obtain stochastic gradients of the VAE standard instance

$$\underset{\theta\in\Theta,\phi\in\Phi}{\text{minimize}} \sum_{i=1}^{N} \frac{1}{\sigma^2}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)} \left\| X_i - f_\theta\left(\mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)\,\varepsilon\right)\right\|^2 + \mathrm{tr}\left(\Sigma_\phi(X_i)\right) + \|$$

select a data X_i, sample $\varepsilon_i \sim \mathcal{N}(0, I)$, evaluate

$$-\mathrm{VLB}_{\theta,\phi}(X_i, \varepsilon_i) \stackrel{\text{def}}{=} \frac{1}{\sigma^2}\left\| X_i - f_\theta\left(\mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)\,\varepsilon_i\right)\right\|^2 + \mathrm{tr}\left(\Sigma_\phi(X_i)\right) + \|$$

and backprop on $\mathrm{VLB}_{\theta,\phi}(X_i, \varepsilon_i)$.
Usually, batch of X_i is selected.
One can sample multiple $Z_{i,1}, \ldots, Z_{i,K}$ (equivalently $\varepsilon_{i,1}, \ldots, \varepsilon_{i,K}$) for each X_i.

Concept 11.22: Traning VAE with Log-Derivative Trick

Computing stochastic gradients without the reparameterization trick.

$$\underset{\theta\in\Theta,\phi\in\Phi}{\text{maximize}} \sum_{i=1}^{N} \underbrace{\mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\log\left(\frac{p_\theta\left(X_i \mid Z\right)p_Z(Z)}{q_\phi\left(Z \mid X_i\right)}\right)\right]}_{\stackrel{\text{def}}{=}\mathrm{VLB}_{\theta,\phi}(X_i)}$$

To obtain unbiased estimates of ∇_θ, compute

$$\frac{1}{K}\sum_{k=1}^{K}\log p_\theta\left(X_i \mid Z_{i,k}\right), \quad Z_{i,1},\ldots,Z_{i,K}\sim q_\phi\left(z \mid X_i\right)$$

and backprop with respect to θ.
We differentiate the VLB objectives

$$\nabla_\phi\mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\log\left(\frac{p_\theta\left(X_i \mid Z\right)p_Z(Z)}{q_\phi\left(Z \mid X_i\right)}\right)\right] = \nabla_\phi\int\log\left(\frac{p_\theta\left(X_i \mid z\right)p_Z(z)}{q_\phi\left(z \mid X_i\right)}\right) q$$

$$= \mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\left(\nabla_\phi \log q_\phi\left(Z \mid X_i\right)\right)\right.$$

§ 11. Variational Autoencoders

Researches

Concept 11.23: VQ-VAE

../../assets/11.8.png

Concept 11.24: VQ-VAE-2

../../assets/11.10.png

Concept 11.25: β-VAE

Uses the loss

$$\ell_{\theta,\phi}(X_i) = \mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\log p_\theta(X_i \mid Z)\right] - \beta D_{\mathrm{KL}}\left(q_\phi(\cdot \mid X_i) \,\|\, p_Z(\cdot)\right)$$

when $\beta = 1, \ell_{\theta,\phi}(X_i) = \mathrm{VLB}_{\theta,\phi}(X_i)$, i.e., β-VAE coincides with VAE when $\beta = 1$.

With $\beta > 1$, authors observed better feature disentanglement.

(I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, -VAE: Learning basic visual concepts with a constrained variational framework, ICLR, 2017.)