Exercise Session 3

VAEs, GANs and Autoregressive Models

Prepared by Johann Wenckstern and Abdülkadir Gökce

Overview

Task 1. Autoencoders vs. Variational Autoencoders.

In this exercise, we compare autoencoders (AEs) and variational autoencoders (VAEs). While both models consist of an encoder and a decoder, only VAEs are considered generative models.

1. **Autoencoder objective.** Let x be input data, g_ϕ the encoder, and f_θ the decoder. Write down the optimization objective of a standard autoencoder. What is being minimized?

2. **Variational autoencoder objective.** In VAEs, the encoder outputs a distribution $q_\phi(z|x)$ over latent variables z. The training objective is the evidence lower bound (ELBO):

$$\mathcal{L}(x) = \mathbb{E}_{z\sim q_\phi(z|x)}\big[\log p_\theta(x|z)\big] - D_{\mathrm{KL}}\big(q_\phi(z|x)\,\|\,p(z)\big).$$

   Explain in words what each of the two terms does.

3. **Generative ability.** Suppose we want to generate new data by sampling from the latent space.

   (a) Why can we simply sample $z \sim p(z) = \mathcal{N}(0, I)$ in a VAE?

   (b) Why is this not possible in a plain autoencoder?

4. **Dropping the KL term.** What happens if we drop the KL divergence term from the VAE loss?

   (a) Write down the new objective.

   (b) Would the resulting model still be generative? Why or why not?

Task 2. Variational Lower Bound for Generative Models.

Consider the following general setting for a generative model with continuous latent variables: Suppose we are given a training dataset

$$\mathcal{D} := \{x^{(i)}\}_{i=1}^{N}$$

which we assume to be generated from a latent variable $z \in \mathbb{R}^K$. Each training sample is generated by first sampling the latent variable $z^{(i)}$ from the true prior $p_{\theta^*}(z)$ and then drawing $x^{(i)}$ from $p_{\theta^*}(x \mid z^{(i)})$, which could be a complex distribution. A relevant task is to estimate the optimal parameters θ^* from the training data. A well-known strategy is to maximize the complete-data log-likelihood, given by

$$\log p_\theta(x^{(1)}, \ldots, x^{(N)}) = \sum_{i=1}^{N} \log p_\theta(x^{(i)}).$$

We assume that $p_\theta(x)$ and $p_\theta(z|x)$ are intractable and instead approximate the latter with the variational approximation $q_\phi(z|x)$.

1. Prove that the likelihood for a single data-point $\log p_\theta(x^{(i)})$ is given by

   $$\mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\big[\log p_\theta(x^{(i)} \mid z)\big] - D_{\mathrm{KL}}\big(q_\phi(z \mid x^{(i)}) \,\|\, p_\theta(z)\big) + D_{\mathrm{KL}}\big(q_\phi(z \mid x^{(i)}) \,\|\, p_\theta(z \mid x^{(i)})\big),$$

   Hint: *Use Bayes rule and* $\mathbb{E}_X[f(Y)] = f(Y)$ *if the random variable* Y *does not depend on* X*.*

2. Group this expression into two terms, and identify the variational lower bound (ELBO) $\mathcal{L}(\theta, \phi; x^{(i)})$.

3. In the ELBO, which of the two terms quantifies reconstruction quality and which one acts as a regularizer?

4. Recall that the ELBO can also be written as

   $$\mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\Big[- \log q_\phi(z \mid x^{(i)}) + \log p_\theta(x^{(i)}, z)\Big].$$

   The goal is to maximize this lower bound w.r.t. the parameters θ and ϕ using Monte Carlo sampling to estimate the expectation.

   However, sampling z can be impractical for our purpose. The *reparameterization trick* assigns

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

   Write down the Monte Carlo estimate of the lower bound using this reparameterization trick. Why is this trick essential for backpropagation?

5. Let us consider a univariate Gaussian distribution. Assume

   $$z \sim p(z \mid x) = \mathcal{N}(\mu, \sigma^2).$$

   We want to use the reparameterization $z = g_\phi(\epsilon, x)$ with $\epsilon \sim \mathcal{N}(0, 1)$. What is g in this case?

6. Briefly explain how in the VAE model the log-likelihood could be maximized in practice, and summarize the main differences with a classical autoencoder.

7. Show that for the special case where $p_\theta(z) = \mathcal{N}(0, I)$ is an isotropic Gaussian (in K dimensions), and $q_\phi(z \mid x^{(i)})$ is a multivariate Gaussian with a diagonal covariance matrix, the $-D_{\mathrm{KL}}$ term in the above expression analytically integrates to

$$\frac{1}{2}\sum_{k=1}^{K}\Big(1 + \log(\sigma_k^2) - \mu_k^2 - \sigma_k^2\Big).$$

Task 3. Optimality of the GAN Objective.

Suppose we are given a training dataset

$$\mathcal{D} := \{x^{(i)}\}_{i=1}^{N}$$

sampled from a data distribution $p_{\text{data}}(x)$. In the adversarial modeling framework, we aim to train a *generator* G that maps a noise variable $z \sim p(z)$ to the data space approximating the true data distribution. We refer to the distribution induced by our generator as $p_{\text{g}}(x)$. The generator is trained jointly with a *discriminator* D, which aims to distinguish true from generated data samples on the following min-max objective:

$$\min_G \max_D \; V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}\big[\log D(x)\big] + \mathbb{E}_{z\sim p(z)}\big[\log\big(1 - D(G(z))\big)\big].$$

In this exercise, we will study its optimal solution.

1. What is the relationship of this training objective with the binary cross-entropy loss?

2. Show that for a fixed generator G, the optimal discriminator D is given by

$$D_G^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{g}}(x)}$$

3. Substituting this optimal discriminator into the min-max objective, we can reformulate the GAN training objective as

$$\min_G C(G)$$

where $C(G) = \max_D V(D_G^*, G)$. Recall the definition of the Jensen-Shannon divergence. Show the connection of this objective with the Jensen-Shannon divergence and conclude that the unique global minimum of $C(G)$ is reached by $p_{\text{g}} = p_{\text{data}}$.

Task 4. Bonus: Challenges of GAN Training and Wasserstein GANs.

In this exercise, we explore challenges arising in the optimization of the GAN objective and one proposed solution: *Wasserstein GANs*. The 1-Wasserstein distance between two distributions μ and ν is given by

$$W_1(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \mathbb{E}_{(x,y)\sim\pi}\big[\|x-y\|_1\big],$$

$$\text{where } \Pi(\mu,\nu) = \Big\{\pi(x,y) \in \mathcal{P}(\mathcal{X}\times\mathcal{Y}) \;\Big|\; \text{marginal of } \pi \text{ w.r.t. } x \text{ is } \mu, \text{ and w.r.t. } y \text{ is } \nu\Big\}.$$

Intuitively, the metric measures the minimal cost of moving all mass of μ to ν.

1. Let $Y \sim \mathcal{U}[0,1]$ be the uniform distribution on the unit interval. We consider the distribution μ_0 of the random vector $(0, Y)$ and μ_θ of (θ, Y). Compute

   (a) the Jensen-Shannon divergence $JS(\mu_0, \mu_\theta)$.

   (b) the Wasserstein distance $W_1(\mu_0, \mu_\theta)$.

   Why would the Wasserstein distance be more suitable in this example to learn to approximate μ_0 using gradient descent on θ than the JS-divergence?

2. Based on the previous example, explain why the original GAN loss based on the Jensen-Shannon divergence often leads to training instability. What happens when the generated and true data distribution have no overlap?

3. Another common problem in the training of GANs is mode collapse. Describe what mode collapse is, and provide an intuition for why it occurs in the standard GAN training framework.

4. To remedy the challenges of training instability and mode collapse, the Wasserstein GAN has been proposed. Here, the (implicit) JS-divergence objective of GAN training is replaced against the Wasserstein-1 metric. Following the nomenclature of Task 2, the objective reads as

$$\min_G W_1(p_\text{g}, p_\text{data}). \tag{1}$$

   Under weak assumptions on a parametric generator G_θ, it can be shown that this objective is continous and differentiable almost everywhere.

   Computing the Wasserstein distance via its infimum-based definition is intractable in practice. Using the Kantorovich-Rubinstein duality it can be rewritten as

$$W_1(p_\text{g}, p_\text{data}) = \sup_{\|f\|_L \leq 1} \left(\mathbb{E}_{x \sim p_\text{g}(x)}[f(x)] - \mathbb{E}_{x \sim p_\text{data}(x)}[f(x)] \right),$$

   where the supremum is taken over all 1-Lipschitz functions. This leads to the new min-max Wasserstein GAN objective

$$\min_G \sup_{\|f\|_L \leq 1} \left(\mathbb{E}_{x \sim p_\text{g}(x)}[f(x)] - \mathbb{E}_{x \sim p_\text{data}(x)}[f(x)] \right)$$

   Here, the function f is called the *critic*, which we typically parametrize with a neural network.

   Argue why for this optimization it is in practice sufficient to consider a parametric family $\{f \mid \|f\|_L \leq K\}$ for some fixed constant K. How can this constraint be enforced on a family of neural networks?
