Li-Jintao 2201213292 lijintao@stu.pku.edu.cn
Prof. Mu-Yadong's Deep Learning: Algorithms and Applications
May 31th, 2023

Homework-1

Question 1. Variational Auto-Encoder
VAE is the typical generative model which has a lot of applications. The encoder portion of a VAE yields an approximate posterior distribution $q(z|x)$, and is parameterized on a neural network by weights collectively denoted θ. Hence we more properly write the encoder as $q_\theta(z|x)$. Similarly, the decoder portion of the VAE yields a likelihood distribution $p(x|z)$, and is parameterized on a neural network by weights collectively denoted φ. Hence we more properly denote the decoder portion of the VAE as $p_\varphi(x|z)$. The output of the encoder are parameters of the latent distribution, which is sampled to yield the input into the decoder.

a) Draw the model's framework of VAE

b) Derive the objective function e.g. the Evidence Lower Bound (ELBO) using KL divergence.

HINT: 1) Please include the re-parametrisation trick implemented in VAE

HINT: 2) You may need to use Bayes theorem: $P(x|y) = \frac{P(y|x)P(x)}{P(y)}$

c) Derive the closed form VAE loss: Gaussian latents, based on question b). e.g. Say we choose:

$$P(z) \rightarrow \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{(x-\mu_p)^2}{2\sigma_p^2}\right)$$

and

$$q_\theta\left(z \mid x_i\right) \rightarrow \frac{1}{\sqrt{2\pi\sigma_q^2}} \exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right)$$

SOLUTION:

a) The VAE model's framework can be represented:

$$x \overset{q_\theta(z|x)}{\rightarrow} z \overset{p_\varphi(x|z)}{\rightarrow} x$$

The encoder takes in input data x and outputs the parameters of the latent distribution $q_\theta(z|x)$. The decoder takes in a sample from the latent distribution z and outputs the reconstructed data $p_\varphi(x|z)$.

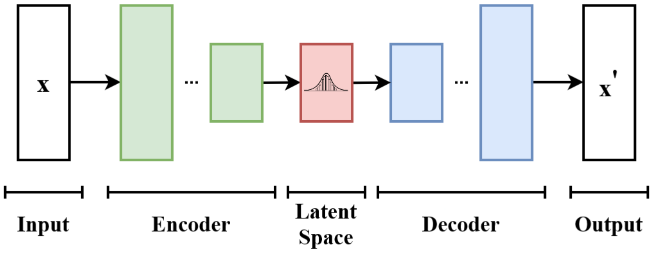

FIGURE 1. VAE framework[1]

b) The main idea behind variational methods is: to find an approximation distributions $q_\theta(z|x)$ that are as closed as possible to the true posterior distribution $P(z|x)$. Obviously the distribution $q_\theta(z|x)$ should be relatively easy and more tractable for inference. Refer to:[2, 3].

To measure the closeness of the two distribution $q_\theta(z|x)$ and $P(z|x)$, a common metric is the Kullback-Leibler (KL) divergence. The KL divergence for variational inference is:

$$\begin{aligned}\mathrm{KL}(q_\theta(z|x)\|P(z|x)) &= \int q_\theta(z|x)\log\frac{q_\theta(z|x)}{P(z|x)}\,dz\\ &= \mathbb{E}_{q_\theta(z|x)}\left[\log\frac{q_\theta(z|x)}{P(z|x)}\right]\\ &= \mathbb{E}_{q_\theta(z|x)}\left[\log q_\theta(z|x)-\log P(z|x)\right]\\ &= \mathbb{E}_{q_\theta(z|x)}\left[\log q_\theta(z|x)-\log\frac{P(x,z)}{P(x)}\right]\\ &= \mathbb{E}_{q_\theta(z|x)}\left[\log q_\theta(z|x)-\log P(x,z)\right]+\log P(x)\end{aligned}$$

Then, we can get:

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

Since KL divergence is always non-negative(≥ 0), we get $\log P(x) \geq \mathbb{E}_{q_\theta(z|x)}\left[\log P(x,z)-\log q_\theta(z|x)\right]$. This lower bound of the log probability of observations is Evidence Lower BOund(ELBO). And we also know the difference between $\log P(x)$ and ELBO is exactly the KL divergence of the approximation and true distribution. In others words, the ELBO hits the log probability IFF the approximation distribution is perfectly closed to the true posterior distribution.

Finally, we have the objective function ELBO, which can be written as:

$$\begin{aligned}\mathrm{ELBO} &= \mathbb{E}_{q_\theta(z|x)}\left[\log P(x,z)-\log q_\theta(z|x)\right]\\ &= \mathbb{E}_{q_\theta(z|x)}\left[\log P(x|z)+\log P(z)-\log q_\theta(z|x)\right]\\ &= \mathbb{E}_{q_\theta(z|x)}\left[\log P(x|z)\right]-\mathbb{E}_{q_\theta(z|x)}\left[\log q_\theta(z|x)-\log P(z)\right]\\ &= \mathbb{E}_{q_\theta(z|x)}[\log P(x|z)]-\mathrm{KL}\left(q_\theta(z|x)\|P(z)\right)\end{aligned}$$

The first term describes the probability of the data given the latent variable z, and the second term is the KL divergence between the approximate posterior $q_\theta(z|x)$ and the prior $P(z)$.

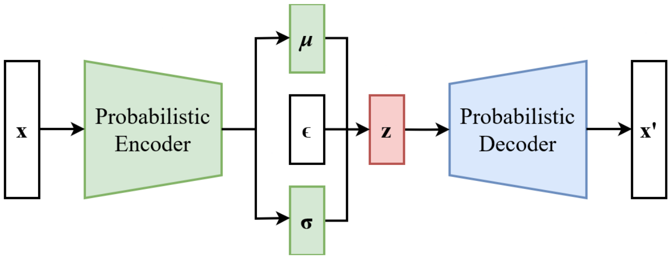

Figure 2. VAE with raparameterized trick[1]

The re-parametrization trick is used to make the ELBO differentiable with respect to θ. This is done by introducing an auxiliary variable ϵ such that $z = g_\theta(\epsilon, x)$ where g_θ is a differentiable function. This allows us to write the ELBO as:

$$\mathrm{ELBO} = \mathbb{E}_{P(\epsilon)}[\log P(x|g_\theta(\epsilon,x))] - \mathrm{KL}\left(q_\theta(z|x)\|P(z)\right)$$

In practice, we can sample ϵ from a standard normal distribution and use the decoder to generate z from x and ϵ. Then, z can be written as:

$$g_\theta(\epsilon,x) = \mu_\theta(x) + \sigma_\theta(x)\odot\epsilon, \quad \epsilon\sim\mathcal{N}(0,I)$$

where $\mu_\theta(x)$ and $\sigma_\theta(x)$ are the mean and standard deviation of the approximate posterior $q_\theta(z|x)$, respectively. $\odot$ denotes the element-wise product. The mean and standard deviation are outputs of the encoder network. In VAE, The encoder is then trained to minimize the KL divergence between the approximate posterior and the prior. The decoder is trained to minimize the reconstruction error between the input and the output.

c) If we choose Gaussian distributions for both the prior and approximate posterior, we can derive a closed-form expression for the VAE loss.

The KL divergence between two Gaussian distributions has a closed-form expression. Refer to:[4].

$$
\begin{aligned}
\mathrm{KL}\left(q_\theta(z|x) \| P(z)\right) &= \int q_\theta(z|x) \log \frac{q_\theta(z|x)}{P(z)}\, dz \\
&= \int \frac{1}{\sqrt{2\pi\sigma_q^2}} \exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right) \log \frac{\frac{1}{\sqrt{2\pi\sigma_q^2}} \exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{(x-\mu_p)^2}{2\sigma_p^2}\right)}\, dx \\
&= \frac{1}{2}\left(log\frac{\sigma_p^2}{\sigma_q^2} + \frac{\sigma_q^2 + (\mu_q - \mu_p)^2}{\sigma_p^2} - 1 \right) \\
&= \log \frac{\sigma_p}{\sigma_q} + \frac{\sigma_q^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} - \frac{1}{2}
\end{aligned}
$$

Substituting this into the ELBO, we get:

$$
\mathrm{ELBO} = \mathbb{E}_{p(\epsilon)}[\log P(x|g_\theta(\epsilon, x))] - \log \frac{\sigma_p}{\sigma_q} - \frac{\sigma_q^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} + \frac{1}{2}
$$

Question 2. Policy based Reinforcement Learning

Please write down the detailed derivation of the policy gradient method in the episodic case, e.g. Show that $\nabla J(\boldsymbol{\theta}) \propto \mathbb{E}_\pi\left[\sum_a q_\pi(s,a) \nabla \pi(a \mid s)\right]$

Some definitions you may need:

$$
J(\boldsymbol{\theta}) = v_{\pi_\theta}(s_0)
$$

where v_{π_θ} is the true state value function for π_θ, the policy determined by θ state value function:

$$
v_\pi = \mathbb{E}_\pi\left[G_t \mid S_t = s\right] = \sum_a \pi(a \mid s) q_\pi(s,a)
$$

state-action value function:

$$
q_\pi(s,a) = \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right] = \sum_{s',r} P\left(s', r \mid s, a\right)\left(r + v_\pi\left(s'\right)\right)
$$

where we have the return defined as:

$$
G_t = R_{t+1} + R_{t+2} + \ldots
$$

SOLUTION:

(1) One way to proof:

The policy gradient method is a way of optimizing a parameterized policy $\pi(a|s, \boldsymbol{\theta})$ with respect to some objective function $J(\boldsymbol{\theta})$. In the episodic case, $J(\boldsymbol{\theta})$ is the expected return from the start state of each episode. To show that $\nabla J(\boldsymbol{\theta}) \propto \mathbb{E}_\pi\left[\sum_a q_\pi(s,a) \nabla \pi(a \mid s)\right]$, we try to proof the Policy Gradient Theorem. Refer to:[5, 6].

We define the performance measure as the value of the start state of the episode. We can simplify the notation without losing any meaningful generality by assuming that every episode starts in some particular (non-random) state s_0. Then, in the episodic case we define performance as:

$$
J(\boldsymbol{\theta}) = v_{\pi_{\boldsymbol{\theta}}}(s_0)
$$

where $v_{\pi_{\boldsymbol{\theta}}}$ is the true value function for $\pi_{\boldsymbol{\theta}}$, the policy determined by $\boldsymbol{\theta}$. From here on in our discussion we will assume no discounting ($\gamma = 1$) for the episodic case.

First note that the gradient of the state-value function can be written in terms of the action-value function as

$$\begin{aligned}
\nabla v_\pi(s) &= \nabla\left[\sum_a \pi(a \mid s) q_\pi(s,a)\right] \\
&= \sum_a \left[\nabla \pi(a \mid s) q_\pi(s,a) + \pi(a \mid s) \nabla q_\pi(s,a)\right] \\
&= \sum_a \left[\nabla \pi(a \mid s) q_\pi(s,a) + \pi(a \mid s) \nabla \sum_{s',r} p\left(s', r \mid s, a\right)\left(r + v_\pi\left(s'\right)\right)\right] \\
&= \sum_a \left[\nabla \pi(a \mid s) q_\pi(s,a) + \pi(a \mid s) \sum_{s'} p\left(s' \mid s, a\right) \nabla v_\pi\left(s'\right)\right] \\
&= \sum_a \Bigg\{\nabla \pi(a \mid s) q_\pi(s,a) + \pi(a \mid s) \sum_{s'} p\left(s' \mid s, a\right) \\
\sum_{a'} &\left[\nabla \pi\left(a' \mid s'\right) q_\pi\left(s', a'\right) + \pi\left(a' \mid s'\right) \sum_{s''} p\left(s'' \mid s', a'\right) \nabla v_\pi\left(s''\right)\right]\Bigg\} \\
&= \sum_{x \in \mathcal{S}} \sum_{k=0}^{\infty} \operatorname{Pr}(s \rightarrow x, k, \pi) \sum_a \nabla \pi(a \mid x) q_\pi(x,a)
\end{aligned}$$

after repeated unrolling, where $\operatorname{Pr}(s \rightarrow x, k, \pi)$ is the probability of transitioning from state s to state x in k steps under policy π. It is then immediate that

$$\begin{aligned}
\nabla J(\boldsymbol{\theta}) &= \nabla v_\pi\left(s_0\right) \\
&= \sum_s \left(\sum_{k=0}^{\infty} \operatorname{Pr}\left(s_0 \rightarrow s, k, \pi\right)\right) \sum_a \nabla \pi(a \mid s) q_\pi(s,a) \\
&= \sum_s \eta(s) \sum_a \nabla \pi(a \mid s) q_\pi(s,a) \\
&= \sum_{s'} \eta\left(s'\right) \sum_s \frac{\eta(s)}{\sum_{s'} \eta\left(s'\right)} \sum_a \nabla \pi(a \mid s) q_\pi(s,a) \\
&= \sum_{s'} \eta\left(s'\right) \sum_s \mu(s) \sum_a \nabla \pi(a \mid s) q_\pi(s,a) \\
&\propto \sum_s \mu(s) \sum_a \nabla \pi(a \mid s) q_\pi(s,a)
\end{aligned}$$

where $\eta(s)$ is $\sum_{k=0}^{\infty} \operatorname{Pr}(s_0 \rightarrow s, k, \pi)$ and $\mu(s)$ is $\frac{\eta(s)}{\sum_{s'} \eta(s')}$. In the episodic case, the constant of proportionality $\sum_{s'} \eta\left(s'\right)$ is the average length of an episode. The distribution μ is the on-policy distribution under π.

The policy gradient theorem gives an exact expression proportional to the gradient; all that is needed is some way of sampling whose expectation equals or approximates this expression. Notice that the right-hand side of the policy gradient theorem is a sum over states weighted by how often the states occur under the target policy π; if π is followed, then states will be encountered in these proportions. Thus

$$\begin{aligned}
\nabla J(\boldsymbol{\theta}) &\propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(a \mid s) \\
&= \mathbb{E}_\pi\left[\sum_a q_\pi\left(s, a\right) \nabla \pi\left(a \mid s\right)\right]
\end{aligned}$$

(2) Another way to proof:

The policy gradient method is used to optimize a parameterized policy $\pi(a|s, \boldsymbol{\theta})$ with respect to the expected return $J(\boldsymbol{\theta})$. In the episodic case, the expected return can be written as $J(\boldsymbol{\theta}) = v_{\pi_{\boldsymbol{\theta}}}(s_0)$, where s_0 is the initial state.

The gradient of the expected return with respect to the policy parameters $\boldsymbol{\theta}$ can be derived using the likelihood ratio trick. The likelihood ratio trick states that $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{x \sim p(x)}[f(x)] = \mathbb{E}_{x \sim p(x)}[f(x) \nabla_{\boldsymbol{\theta}} \log p(x)]$.

Applying this to our case, we have:

$$\nabla J(\boldsymbol{\theta}) = \nabla v_{\pi_{\boldsymbol{\theta}}}(s_0) = \nabla \mathbb{E}_{\tau \sim p_{\boldsymbol{\theta}}(\tau)}[G_0]$$

where τ represents an episode and $p_{\boldsymbol{\theta}}(\tau)$ is the probability of the episode under policy $\pi_{\boldsymbol{\theta}}$. Using the likelihood ratio trick, we get:

$$\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{\tau \sim p_{\boldsymbol{\theta}}(\tau)}[G_0 \nabla \log p_{\boldsymbol{\theta}}(\tau)]$$

The probability of an episode can be written as $p_{\boldsymbol{\theta}}(\tau) = p(s_0) \prod_{t=0}^{T-1} \pi(a_t|s_t, \boldsymbol{\theta}) p(s_{t+1}|s_t, a_t)$, where T is the length of the episode. Taking the gradient of the log-probability, we get:

$$\nabla \log p_{\boldsymbol{\theta}}(\tau) = \sum_{t=0}^{T-1} \nabla \log \pi(a_t|s_t, \boldsymbol{\theta})$$

Substituting this back into our expression for $\nabla J(\boldsymbol{\theta})$, we get:

$$\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{\tau \sim p_{\boldsymbol{\theta}}(\tau)}[G_0 \sum_{t=0}^{T-1} \nabla \log \pi(a_t|s_t, \boldsymbol{\theta})]$$

This can be further simplified by noting that $G_0 = \sum_{t=0}^{T-1} G_t$, where G_t is the return from time step t. Substituting this in and rearranging terms, we get:

$$\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{\tau \sim p_{\boldsymbol{\theta}}(\tau)}[\sum_{t=0}^{T-1} G_t \nabla \log \pi(a_t|s_t, \boldsymbol{\theta})]$$

This expression can be further simplified by noting that $\mathbb{E}_A[f(A, B)] = \sum_a p(a) f(a, B)$ and that $G_t = q_\pi(S_t, A_t)$, where $q_\pi(s, a)$ is the action-value function. Substituting these in and rearranging terms, we get:

$$\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{S_t, A_t}[q_\pi(S_t, A_t) \nabla \log \pi(A_t|S_t, \boldsymbol{\theta})]$$

Finally, using the fact that $\nabla f(x) = f(x) \nabla \log f(x)$ and rearranging terms, we get:

$$\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{S_t}[q_\pi(S_t, A_t) \sum_a \pi(a|S_t, \boldsymbol{\theta}) \nabla \pi(a|S_t, \boldsymbol{\theta})] = \mathbb{E}_{S_t}[\sum_a q_\pi(S_t, a) \nabla \pi(a|S_t, \boldsymbol{\theta})]$$

Thus, we have shown that $\nabla J(\boldsymbol{\theta}) \propto \mathbb{E}_\pi \left[\sum_a q_\pi(s, a) \nabla \pi(a \mid s)\right]$.

References

[1] Wikipedia. Variational autoencoder. [Online]. Available: https://en.wikipedia.org/wiki/Variational_autoencoder

[2] M. Patacchiola. Evidence, KL-divergence, and ELBO. [Online]. Available: https://mpatacchiola.github.io/blog/2021/01/25/intro-variational-inference.html

[3] X. Yang. Understanding the Variational Lower Bound. [Online]. Available: https://xyang35.github.io/2017/04/14/variational-lower-bound/

[4] J. Su. Variational auto-encoder-1|scientific spaces. [Online]. Available: https://spaces.ac.cn/archives/5253

[5] L. Weng. Policy gradient algorithms. [Online]. Available: https://lilianweng.github.io/posts/2018-04-08-policy-gradient/

[6] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, second edition ed. MIT Press. [Online]. Available: http://incompleteideas.net/book/the-book.html