Due Date: April 29th 23:59, 2020

Instructions

- *For all questions, show your work!*
- *Please use a document preparation system such as LaTeX, unless noted otherwise.*
- *Unless noted that questions are related, assume that notation and definitions for each question are self-contained and independent.*
- *Submit your answers electronically via Gradescope.*
- ***TAs for this assignment are Samuel Lavoie, Jae Hyun Lim, Sanae Lotfi.***

This assignment covers mathematical and algorithmic techniques underlying the four most popular families of deep generative models. Thus, we explore autoregressive models (Question 1), reparameterization trick (Question 2), variational autoencoders (VAEs, Questions 3-4), normalizing flows (Question 5), and generative adversarial networks (GANs, Question 6).

Question 1 (4-4-4-4)**.** One way to enforce autoregressive conditioning is via masking the weight parameters.[1] Consider a two-hidden-layer convolutional neural network without kernel flipping, with kernel size 3×3 and padding size 1 on each border (so that an input feature map of size 5×5 is convolved into a 5×5 output). Define mask of type A and mask of type B as

$$(\boldsymbol{M}^A)_{::ij} := \begin{cases} 1 & \text{if } i = 2 \text{ and } j < 2 \\ 1 & \text{if } i = 3 \\ 0 & \text{elsewhere} \end{cases} \qquad (\boldsymbol{M}^B)_{::ij} := \begin{cases} 1 & \text{if } i = 2 \text{ and } j \leq 2 \\ 1 & \text{if } i = 3 \\ 0 & \text{elsewhere} \end{cases}$$

where the index starts from 1. Masking is achieved by multiplying the kernel with the binary mask (elementwise). Specify the receptive field of the output pixel that corresponds to the third row and the fourth column (index 34 of Figure 1 (Left)) in each of the following 4 cases:

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 1 – (Left) 5×5 convolutional feature map. (Right) Template answer.

1. If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^A$ for the second layer.
2. If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^B$ for the second layer.
3. If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^A$ for the second layer.
4. If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^B$ for the second layer.

Your answer should look like Figure 1 (Right).

Answer 1. First we need to find the receptive field for $\boldsymbol{M}^A_{::34}$ and $\boldsymbol{M}^B_{::34}$ which are shown in Figure2. Now, we only need to calculate the receptive field for pixels 33, 34, 43, 44, and 45 for mask A which are available in Figure 3.

1. An example of this is the use of masking in the Transformer architecture (Problem 3 of HW2 practical part).

11	12	13	14	15
21	22	23	24	25
31	32	**33**	34	35
41	42	**43**	**44**	**45**
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	**33**	**34**	35
41	42	**43**	**44**	**45**
51	52	53	54	55

FIGURE 2 – Receptive field of pixel 34: (Left) $\boldsymbol{M}^A_{::34}$ and (Right) $\boldsymbol{M}^B_{::34}$.

11	12	13	14	15
21	22	23	24	25
31	**32**	33	34	35
41	**42**	**43**	**44**	**45**
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	**33**	34	35
41	42	**43**	**44**	**45**
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	**42**	43	44	45
51	**52**	**53**	**54**	**55**

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	**43**	44	45
51	52	**53**	**54**	**55**

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	**44**	45
51	52	53	**54**	**55**

FIGURE 3 – Receptive field of pixels 33, 34, 43, 44, 45 for mask A.

Lets find the receptive fields of pixels 33,34,43,44,45 for mask B. Results are shown in Figure 4.

11	12	13	14	15
21	22	23	24	25
31	**32**	**33**	34	35
41	**42**	**43**	**44**	**45**
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	**33**	**34**	35
41	42	**43**	**44**	**45**
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	**42**	**43**	44	45
51	**52**	**53**	**54**	**55**

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	**43**	44	45
51	52	**53**	**54**	**55**

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	**44**	45
51	52	53	**54**	**55**

FIGURE 4 – Receptive field of pixels 33, 34, 43, 44, 45 for mask B.

1. If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^A$ for the second layer the mask will be the union receptive field of pixels 33, 43, 44, 45 which were shown in Figure 3. The mask is shown in Figure 5.
2. If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^B$ for the second layer the mask will be the combination of receptive fields for pixels 33, 34, 43, 44, 45 from mask A. 3. The mask after combination is plotted in Figure 6.
3. If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^A$ for the second layer the mask will be the combination of receptive fields for pixels 33, 43, 44, 45 from mask B 4. The mask after combination is plotted in Figure 7.
4. If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^B$ for the second layer the mask will be the union receptive field of pixels 33, 34, 43, 44, 45 in Figure 4. The mask after combination is plotted in Figure 8.

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 5 – Part 1

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 6 – Part 2

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 7 – Part 3

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 8 – Part 4

Question 2 (6-3-6-3)**.** Reparameterization trick is a standard technique that makes the samples of a random variable differentiable. The trick represents the random variable as a simple mapping from another random variable drawn from some simple distribution [2]. If the reparameterization is a bijective function, the induced density of the resulting random variable can be computed using the change-of-variable density formula, whose computation requires evaluating the determinant of the Jacobian of the mapping.

Consider a random vector $Z \in \mathbb{R}^K$ with a density function $q(\boldsymbol{z}; \phi)$ and a random variable $Z_0 \in \mathbb{R}^K$ having a ϕ-independent density function $q(\boldsymbol{z}_0)$. We want to find a deterministic function $\boldsymbol{g} : \mathbb{R}^K \to \mathbb{R}^K$ that depends on ϕ, to transform Z_0, such that the induced distribution of the transformation has the same density as Z. Recall the change of density for a bijective, differentiable $\boldsymbol{g}$:

$$q(\boldsymbol{g}(\boldsymbol{z}_0)) = q(\boldsymbol{z}_0) \left|\det \boldsymbol{J}_{\boldsymbol{z}_0} \boldsymbol{g}(\boldsymbol{z}_0)\right|^{-1} = q(\boldsymbol{z}_0) \left|\det \left(\frac{\partial \boldsymbol{g}(\boldsymbol{z}_0)}{\partial \boldsymbol{z}_0}\right)\right|^{-1} \tag{1}$$

1. Assume $q(\boldsymbol{z}_0) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_K)$ and $\boldsymbol{g}(\boldsymbol{z}_0) = \mu + \sigma \odot \boldsymbol{z}_0$, where $\mu \in \mathbb{R}^K$ and $\sigma \in \mathbb{R}^K_{>0}$. Note that $\odot$ is element-wise product. Show that $\boldsymbol{g}(\boldsymbol{z}_0)$ is distributed by $\mathcal{N}(\mu, \text{diag}(\sigma^2))$ using Equation (1).
2. Compute the time complexity of evaluating $|\det \boldsymbol{J}_{\boldsymbol{z}_0} \boldsymbol{g}(\boldsymbol{z}_0)|$ when $\boldsymbol{g}(\boldsymbol{z}_0) = \mu + \sigma \odot \boldsymbol{z}_0$. Use the big $\mathcal{O}$ notation and expressive the time complexity as a function of K.
3. Assume $\boldsymbol{g}(\boldsymbol{z}_0) = \mu + \boldsymbol{S}\boldsymbol{z}_0$, where $\boldsymbol{S}$ is a non-singular $K \times K$ matrix. Derive the density of $\boldsymbol{g}(\boldsymbol{z}_0)$ using Equation (1).
4. The time complexity of the general Jacobian determinant is at least $\mathcal{O}(K^{2.373})$ [3]. Assume instead $\boldsymbol{g}(\boldsymbol{z}_0) = \mu + \boldsymbol{S}\boldsymbol{z}_0$ with $\boldsymbol{S}$ being a $K \times K$ lower triangular matrix; i.e. $\boldsymbol{S}_{ij} = 0$ for $j > i$, and $\boldsymbol{S}_{ii} > 0$. What is the time complexity of evaluating $|\det \boldsymbol{J}_{\boldsymbol{z}_0} \boldsymbol{g}(\boldsymbol{z}_0)|$?

Answer 2. 1. We know that we can easily derive the density function for normal distribution. Assuming $q(\boldsymbol{z}_0) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_K)$ we have $q(\boldsymbol{z}_0) = \frac{1}{\sqrt{2\pi}} \exp^{\frac{-1}{2}(\frac{(z-0)}{I_k})^2} = \frac{1}{\sqrt{2\pi}} \exp(\frac{-1}{2}(z_0^T z_0)$. We also assumed that $\boldsymbol{g}(\boldsymbol{z}_0) = \mu + \sigma \odot \boldsymbol{z}_0$.

$$\boldsymbol{g}(\boldsymbol{z}_0) = \mu + \sigma \odot \boldsymbol{z}_0 \to \boldsymbol{z}_0 = \frac{\boldsymbol{g}(\boldsymbol{z}_0) - \mu}{\sigma} \quad (*element - wisedivision)$$

Now we can use the $\boldsymbol{z}_0$ we found to calculate $q(\boldsymbol{z}_0)$ which is a part of equation (1).

$$q(\boldsymbol{z}_0) = \frac{1}{\sqrt{2\pi}} \exp(\frac{-1}{2}(\frac{\boldsymbol{g}(\boldsymbol{z}_0) - \mu}{\sigma})^T(\frac{\boldsymbol{g}(\boldsymbol{z}_0) - \mu}{\sigma}))$$

$$= \frac{1}{\sqrt{2\pi}} \exp(\frac{-1}{2}(\boldsymbol{g}(\boldsymbol{z}_0) - \mu)^T(diag(\sigma^2))^{-1}(\boldsymbol{g}(\boldsymbol{z}_0) - \mu))$$

Now, lets find $|\det \boldsymbol{J}_{\boldsymbol{z}_0} \boldsymbol{g}(\boldsymbol{z}_0)|^{-1}$. Having $\boldsymbol{g}(\boldsymbol{z}_0) = \mu + \sigma \odot \boldsymbol{z}_0$ in mind we can write:

$$|\det \boldsymbol{J}_{\boldsymbol{z}_0} \boldsymbol{g}(\boldsymbol{z}_0)|^{-1} = \frac{1}{|\det(diag(\sigma)|} = |\det(diag(\sigma))|^{-1}$$

We need to show that $\boldsymbol{g}(\boldsymbol{z}_0)$ using formula (1).

$$\boldsymbol{g}(\boldsymbol{z}_0) = \frac{1}{\sqrt{2\pi\,|\det(\sigma^2)|}} \exp(\frac{-1}{2}(\boldsymbol{g}(\boldsymbol{z}_0) - \mu)^T(diag(\sigma^2))^{-1}(\boldsymbol{g}(\boldsymbol{z}_0) - \mu)) = \mathcal{N}(\mu, diag(\sigma^2))$$

2. More specifically, these mapping should be differentiable wrt the density function's parameters.
3. `https://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations`

2. Calculating $diag(\sigma^2)$ takes K operations and determinant of diagonal matrix also needs K operations. So the overall complexity of $|\det \boldsymbol{J}_{\boldsymbol{z}_0}\boldsymbol{g}(\boldsymbol{z}_0)|$ is $\mathcal{O} = K$.
3. I will follow the same steps as part 1.

$$\boldsymbol{g}(\boldsymbol{z}_0) = \mu + \boldsymbol{S}\boldsymbol{z}_0 \rightarrow \boldsymbol{z}_0 = \frac{\boldsymbol{g}(\boldsymbol{z}_0) - \mu}{\boldsymbol{S}} = \boldsymbol{S}^{-1}(\boldsymbol{g}(\boldsymbol{z}_0) - \mu)$$

Now we need to calculate $|\det \boldsymbol{J}_{\boldsymbol{z}_0}\boldsymbol{g}(\boldsymbol{z}_0)|^{-1} = \frac{1}{|\boldsymbol{S}|} = |\boldsymbol{S}|^{-1}$
Lets derive $\boldsymbol{g}(\boldsymbol{z}_0)$ using equation (1).
$q(\boldsymbol{z}_0) = 1\sqrt{2\pi}\,|\boldsymbol{S}| \exp(\frac{-1}{2}(\boldsymbol{S}^{-1}\boldsymbol{g}(\boldsymbol{z}_0) - \mu)^T(\boldsymbol{S}^{-1}\boldsymbol{g}(\boldsymbol{z}_0) - \mu))$
$= 1\sqrt{2\pi}\,\left|\boldsymbol{S}\boldsymbol{S}^T\right| \exp(\frac{-1}{2}(\boldsymbol{g}(\boldsymbol{z}_0) - \mu)^T(\boldsymbol{S}\boldsymbol{S}^T)^{-1}(\boldsymbol{g}(\boldsymbol{z}_0) - \mu))$
$= \mathcal{N}(\mu, \boldsymbol{S}\boldsymbol{S}^T)$

4. If matrix $\boldsymbol{S}$ be a lower triangular matrix then its determinant will be the product of the diagonal values. As a result the time complexity of the Jacobian determinant will be equal to $\mathcal{O} = K$.

Question 3 (5-5-6)**.** Consider a latent variable model $p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})dz$, where $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_K)$ and $\boldsymbol{z} \in \mathbb{R}^K$.The encoder network (aka "recognition model") of variational autoencoder, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, is used to produce an approximate (variational) posterior distribution over latent variables $\boldsymbol{z}$ for any input datapoint $\boldsymbol{x}$.[4] This distribution is trained to match the true posterior by maximizing the evidence lower bound (ELBO):

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

Let $\mathcal{Q}$ be the family of variational distributions with a feasible set of parameters $\mathcal{P}$; i.e. $\mathcal{Q} = \{q(\boldsymbol{z}; \pi) : \pi \in \mathcal{P}\}$; for example π can be mean and standard deviation of a normal distribution. We assume q_ϕ is parameterized by a neural network (with parameters ϕ) that outputs the parameters, $\pi_\phi(\boldsymbol{x})$, of the distribution $q \in \mathcal{Q}$, i.e. $q_\phi(\boldsymbol{z}|\boldsymbol{x}) := q(\boldsymbol{z}; \pi_\phi(\boldsymbol{x}))$.

1. Show that maximizing the expected complete data log likelihood (ECLL)

   $$\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})]$$

   for a fixed $q(\boldsymbol{z}|\boldsymbol{x})$, wrt the model parameter θ, is equivalent to maximizing

   $$\log p_\theta(\boldsymbol{x}) - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$$

   This means the maximizer of the ECLL coincides with that of the marginal likelihood only if $q(\boldsymbol{z}|\boldsymbol{x})$ perfectly matches $p(\boldsymbol{z}|\boldsymbol{x})$.
2. Consider a finite training set $\{\boldsymbol{x}_i : i \in \{1, ..., n\}\}$, n being the size the training data. Let ϕ^* be the maximizer $\arg\max_\phi \sum_{i=1}^n \mathcal{L}(\theta, \phi; \boldsymbol{x}_i)$ with θ fixed. In addition, for each $\boldsymbol{x}_i$ let $q_i \in \mathcal{Q}$ be an "instance-dependent" variational distribution, and denote by q_i^* the maximizer of the corresponding ELBO. Compare $D_{\mathrm{KL}}(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$ and $D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$. Which one is bigger ?
3. Following the previous question, compare the two approaches in the second subquestion
   (a) in terms of bias of estimating the marginal likelihood via the ELBO, in the best case scenario (i.e. when both approaches are optimal within the respective families)
   (b) from the computational point of view (efficiency)
   (c) in terms of memory (storage of parameters)

Answer 3. 1. Our goal is to show that for a fixed $q(\boldsymbol{z}|\boldsymbol{x})$:

$$\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})] = \log p_\theta(\boldsymbol{x}) - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$$

We also know $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_K)$.
Lets start form

$$\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})] = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})p_\theta(\boldsymbol{x})}{p(\boldsymbol{z})}] = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{q(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})} \cdot \frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})p_\theta(\boldsymbol{x})}{p(\boldsymbol{z})}]$$

$$= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x})] + \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{q(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})} \cdot \frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})}] = \log p_\theta(\boldsymbol{x}) + \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})}] + \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{q(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z})}]$$

4. Using a recognition model in this way is known as "amortized inference" ; this can be contrasted with traditional variational inference approaches (see, e.g., Chapter 10 of Bishop's *Pattern Recognition an Machine Learning*), which fit a variational posterior independently for each new datapoint.

$$= \log p_\theta(\boldsymbol{x}) - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{q(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})}] + \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{q(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z})}]$$
$$= \log p_\theta(\boldsymbol{x}) - D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})) + D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))$$

The term $D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))$ is constant. Therefore, maximizing $\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})]$ is equivalent to maximizing $\log p_\theta(\boldsymbol{x}) - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$

2. We know that $\arg\max(f(\boldsymbol{x}) + g(\boldsymbol{x})) \geq \arg\max(f(\boldsymbol{x}))$. Given that ϕ^* be the maximizer of $\arg\max_\phi \sum_{i=1}^n \mathcal{L}(\theta, \phi; \boldsymbol{x}_i)$ and q_i^* be the maximizer of $\arg\max_\theta \sum_{i=1}^n \mathcal{L}(\theta, q_i; \boldsymbol{x}_i)$. We also know that θ is fixed so maximizing the ELBO is equal to minimizing the KL divergence. Take an instance $\boldsymbol{x}_i$ which q_i^* maximizes ELBO so q_i^* will minimize the KL divergence between q_i^* and $p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)$). So we can write $\mathcal{L}(\theta, \phi^*; \boldsymbol{x}_i) \leq \mathcal{L}(\theta, q_i^*; \boldsymbol{x}_i)$.

Lets calculate the ELBO for VAEs:

$\mathcal{L}(\theta, \phi; \boldsymbol{x}_i) = \mathbb{E}_{q_\phi}[\log p_\theta(\boldsymbol{x}_i \mid \boldsymbol{z})] - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}_i)||p(\boldsymbol{z})) = \mathbb{E}_{q_\phi}[\log \frac{p_\theta(\boldsymbol{z} \mid \boldsymbol{x}_i)p(\boldsymbol{x}_i)}{p(\boldsymbol{z})}] - \mathbb{E}_{q_\theta}[\log \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x}_i)p(\boldsymbol{x}_i)}{p(\boldsymbol{z})}]$

$= \mathbb{E}_{q_\phi}[\log p_\theta(\boldsymbol{z} \mid \boldsymbol{x}_i] + \log p_\theta(\boldsymbol{x}_i) - \log p(\boldsymbol{z}) + \log p(\boldsymbol{z}) - \mathbb{E}_{q_\phi}[\log q_\theta(\boldsymbol{z} \mid \boldsymbol{x}_i))]$

$= \log p_\theta(\boldsymbol{x}_i) - \mathbb{E}_{q_\phi}[\log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)p(\boldsymbol{x}_i)}{p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)}] = \log p_\theta(\boldsymbol{x}_i) - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$

Now, lets calculate the ELBO for instance-dependent variational inference based models:

$$\mathcal{L}(\theta, q_i; \boldsymbol{x}_i) = \mathbb{E}_{q_i}[\log \frac{p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)p_\theta(x_i)}{q_i(\boldsymbol{z})}] = \log p_\theta(\boldsymbol{x}_i) - \mathbb{E}_{q_i}[\log \frac{q_i(\boldsymbol{z})}{p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)}] = \log p_\theta(\boldsymbol{x}_i) - D_{KL}(q_i(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$$

Given the equations above we can conclude that $D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)) \leq D_{\mathrm{KL}}(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$

3. (a) Bias of estimating the marginal likelihood via the ELBO is KL-divergence. In part 3.2 we showed that $D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)) \leq D_{\mathrm{KL}}(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$ which means VAEs are more biased than instance-dependent variational inference based models.

   (b) The number of parameters in each iteration are the same which mean efficiency based on per iteration are the same. But the number of iteration in order to find the best q^* might differ. And we know that using VAEs we can find q^* sooner than calculating for all n possible cases which means instance-dependent variational inference based method is less efficient in both training and test time.

   (c) We know that q_i^* is modeled for each example which means in nstance-dependent variational inference based models we need $n \times number_of_parameters$ than what is needed in VAEs. As a result, VAEs are more efficient in terms of memory usage. Since the number of parameters are independent of size of the dataset.

Question 4 (8-8)**.** Let $p(x, z)$ be the joint probability of a latent variable model where x and z denote the observed and unobserved variables, respectively. Let $q(z|x)$ be an auxiliary distribution which we call the *proposal*, and define [5]

$$\mathcal{L}_K[q(z|x)] = \int \cdots \int \left(q(z_1|x)...q(z_K|x) \log \frac{1}{K} \sum_{j=1}^{K} \frac{p(x, z_j)}{q(z_j|x)} \right) dz_1 dz_2 ... dz_K$$

We've seen in class that this objective is a tighter lower bound on $\log p(x)$ than the evidence lower bound (ELBO), which is equal to $\mathcal{L}_1$; that is $\mathcal{L}_1[q(z|x)] \leq \mathcal{L}_K[q(z|x)] \leq \log p(x)$.

In fact, $\mathcal{L}_K[q(z|x)]$ can be interpreted as the ELBO with a refined proposal distribution. For z_j drawn i.i.d. from $q(z|x)$ with $2 \leq j \leq K$, define the *unnormalized* density

$$\tilde{q}(z|x, z_2, ..., z_K) := \frac{p(x, z)}{\frac{1}{K}\left(\frac{p(x,z)}{q(z|x)} + \sum_{j=2}^{K} \frac{p(x,z_j)}{q(z_j|x)}\right)}$$

(Hint: in what follows, you might need to use the fact that if $w_1, ..., w_K$ are random variables that have the same distribution, then $K\mathbb{E}[w_1] = \sum_i \mathbb{E}[w_i] = \mathbb{E}[\sum_i w_i]$. You need to identify such w_i's before applying this fact for each subquestion.)

1. Show that $\mathcal{L}_K[q(z|x)] = \mathbb{E}_{z_{2:K}}[\mathcal{L}_1[\tilde{q}(z|x, z_2, ..., z_K)]]$; that is, the importance-weighted lower bound with K samples is equal to the average ELBO with the unnormalized density as a refined proposal.
2. Show that $q_K(z|x) := \mathbb{E}_{z_{2:K}}[\tilde{q}(z|x, z_2, ..., z_K)]$ is in fact a probability density function. Also, show that $\mathcal{L}_1[q_K(z|x)]$ is an even tighter lower bound than $\mathcal{L}_K[q(z|x)]$. This implies $q_K(z|x)$ is closer to the true posterior $p(z|x)$ than $q(z|x)$ due to resampling, since $\mathcal{L}_K[q(z|x)] \geq \mathcal{L}_1[q(z|x)]$. (Hint: $f(x) := -x \log x$ is concave.)

Answer 4. 1. Let $\boldsymbol{w}_i = \frac{p(\boldsymbol{x}, \boldsymbol{z}_i)}{q(\boldsymbol{z}_i, \boldsymbol{x})}$. In what follows I will start from the importance-weighted lower bound formula which is given and then following some steps to show that it is equal to the

5. Note that $\mathcal{L}_K[\cdot]$ is a "functional" whose input argument is a "function" $q(\cdot|x)$.

average ELBO.

$$\mathcal{L}_K[q(z|x)] = \int \cdots \int \left(q(z_1|x)...q(z_K|x) \log \frac{1}{K} \sum_{j=1}^{K} \frac{p(x,z_j)}{q(z_j|x)} \right) dz_1 dz_2 ... dz_K \tag{1}$$

$$= \mathbb{E}_{z_2 ... z_k \sim q(z|x)} \left[\int_z \tilde{q}(z|x,z_2,...,z_K) \log \left(\frac{p(x,z)}{\tilde{q}(z|x,z_2,...,z_K)} \right) dz \right] \tag{2}$$

$$= \mathbb{E}_{z_2 ... z_k \sim q(z|x)} \left[\int_z \tilde{q}(z|x,z_2,...,z_K) \log \left(\frac{p(x,z)}{\frac{p(x,z)}{\frac{1}{k}\sum_{i=1}^{k} \frac{p(x,z_i)}{q(z_i|x)}}} \right) dz \right] \tag{3}$$

$$= \mathbb{E}_{z_2 ... z_k \sim q(z|x)} \left[\int_z \tilde{q}(z|x,z_2,...,z_K) \log \left(\frac{1}{k} \sum_{i=1}^{k} \frac{p(x,z_i)}{q(z_i|x)} \right) dz \right] \tag{4}$$

$$= \mathbb{E}_{z_2 ... z_k \sim q(z|x)} \left[\int_z k \frac{\frac{p(x,z)}{q(z|x)}}{\sum_{j=1}^{k} \frac{p(x,z_j)}{q(z_j|x)}} q(z|x) \log \left(\frac{1}{k} \sum_{i=1}^{k} \frac{p(x,z_i)}{q(z_i|x)} \right) dz \right] \quad , z = z_1 \tag{5}$$

$$= \mathbb{E}_{z_2 ... z_k \sim q(z|x)} \left[\int_{z_1} k \frac{\frac{p(x,z_1)}{q(z_1|x)}}{\sum_{j=1}^{k} \frac{p(x,z_j)}{q(z_j|x)}} q(z|x) \log \left(\frac{1}{k} \sum_{i=1}^{k} \frac{p(x,z_i)}{q(z_i|x)} \right) dz \right] \quad , (*) \tag{6}$$

$$= \mathbb{E}_{z_1 ... z_k \sim q(z|x)} \left[k \frac{\frac{p(x,z_1)}{q(z_1|x)}}{\sum_{j=1}^{k} \frac{p(x,z_j)}{q(z_j|x)}} \log \left(\frac{1}{k} \sum_{i=1}^{k} \frac{p(x,z_i)}{q(z_i|x)} \right) \right] \tag{7}$$

$$= \mathbb{E}_{z_1 ... z_k \sim q(z|x)} \left[\frac{\sum_{j=1}^{k} \frac{p(x,z_j)}{q(z_j|x)}}{\sum_{j=1}^{k} \frac{p(x,z_j)}{q(z_j|x)}} \log \left(\frac{1}{k} \sum_{i=1}^{k} \frac{p(x,z_i)}{q(z_i|x)} \right) \right] \tag{8}$$

$$= \mathbb{E}_{z_1 ... z_k \sim q(z|x)} \left[\log \left(\frac{1}{k} \sum_{i=1}^{k} \frac{p(x,z_i)}{q(z_i|x)} \right) \right] \tag{9}$$

$$= \mathbb{E}_{z_{2:K}} [\mathcal{L}_1[\tilde{q}(z|x,z_2,...,z_K)]] \tag{10}$$

$(*)$: Here we replaced k with the sum of k terms since their expectations are the same for $i \in 1, 2, ..., k$.

2. In order to show that $q_K(z|x) := \mathbb{E}_{z_{2:K}}[\tilde{q}(z|x, z_2, ..., z_K)]$is a probability distribution we need to show that the summation over possible cases is equal to one. All z_i's have the same expectation.

$$\int_z q_K(z|x)dz = \int_z E_{z_2...z_k\sim q(z|x)}\left[\tilde{q}(z|x, z_2, ..., z_K)\right] dz \tag{11}$$

$$= \int_z E_{z_2...z_k\sim q(z|x)}\left[\frac{p(x,z)}{\frac{1}{k}\left(\frac{p(x,z)}{q(z|x)} + \sum_{j=2}^{k}\frac{p(x,z_j)}{q(z_j|x)}\right)}\right] dz \tag{12}$$

$$= \int_z \frac{q(z|x)}{q(z|x)} E_{z_2...z_k\sim q(z|x)}\left[\frac{p(x,z)}{\frac{1}{k}\left(\frac{p(x,z)}{q(z|x)} + \sum_{j=2}^{k}\frac{p(x,z_j)}{q(z_j|x)}\right)}\right] dz \tag{13}$$

$$= E_{z\sim q(z|x)} E_{z_2...z_k\sim q(z|x)}\left[\frac{\frac{p(x,z)}{q(z|x)}}{\frac{1}{k}\left(\frac{p(x,z)}{q(z|x)} + \sum_{j=2}^{k}\frac{p(x,z_j)}{q(z_j|x)}\right)}\right] \quad , z = z_1 \tag{14}$$

$$= E_{z_1...z_k\sim q(z|x)}\left[\frac{\frac{p(x,z_1)}{q(z_1|x)}}{\frac{1}{k}\left(\sum_{j=1}^{k}\frac{p(x,z_j)}{q(z_j|x)}\right)}\right] \tag{15}$$

$$= k * E_{z_1...z_k\sim q(z|x)}\left[\frac{\frac{p(x,z_1)}{q(z_1|x)}}{\sum_{j=1}^{k}\frac{p(x,z_j)}{q(z_j|x)}}\right] \tag{16}$$

$$= \sum_{i=1}^{k} E_{z_1...z_k\sim q(z|x)}\left[\frac{\frac{p(x,z_i)}{q(z_i|x)}}{\sum_{j=1}^{k}\frac{p(x,z_j)}{q(z_j|x)}}\right] \tag{17}$$

$$= E_{z_1...z_k\sim q(z|x)}\left[\frac{\sum_{i=1}^{k}\frac{p(x,z_i)}{q(z_i|x)}}{\sum_{j=1}^{k}\frac{p(x,z_j)}{q(z_j|x)}}\right] \tag{18}$$

$$= E_{z_1...z_k\sim q(z|x)}\left[1\right] \tag{19}$$

$$= 1 \tag{20}$$

First, we need to show that $\mathcal{L}_1[q_K(z|x)]$ is tighter lower bound than $\mathcal{L}_K[q(z|x)]$. Which means we need to prove that $\mathcal{L}_K[q(z|x)]$ is an upper bound of $\mathcal{L}_1[q_K(z|x)]$. Since we know that $f(x) = -x\log x$ is concave for $x > 0$, and $f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$ we can write $f(\mathbb{E}[x]) = -\mathbb{E}[x]\log\mathbb{E}[x] \geq \mathbb{E}[-x\log x]$.

Let $\tilde{p}(x|z_{1:k}) = \frac{1}{k}\left(\frac{p(x,z)}{q(z|x)} + \sum_{j=2}^{k}\frac{p(x,z_j)}{q(z_j|x)}\right)$ to summarize the calculations.

$$\mathcal{L}_K[q(z|x) = E_{z\sim q(z|x,z_2,...,z_K)}\left[log\left(\frac{p(x,z)}{q(z|x,z_2,...,z_K)(z|x)}\right)\right] \tag{21}$$

$$= E_{z\sim q(z|x,z_2,...,z_K)}\left[log\left(\frac{p(x,z)}{E_{q(z_{2:k}|x)}\left[\frac{p(x,z)}{\tilde{p}(x|z_{1:k})}\right]}\right)\right] \tag{22}$$

$$= E_{z\sim q(z|x,z_2,...,z_K)}\left[-log\left(E_{q(z_{2:k}|x)}\left[\tilde{p}(x|z_{1:k})^{-1}\right]\right)\right] \tag{23}$$

$$= -\int_z p(x,z)E_{q(z_{2:k}|x)}\left[\tilde{p}(x|z_{1:k})^{-1}\right] log\left(E_{q(z_{2:k}|x)}\left[\tilde{p}(x|z_{1:k})^{-1}\right]\right) dz \tag{24}$$

$$\geq -\int_z p(x,z)E_{q(z_{2:k}|x)}\left[\tilde{p}(x|z_{1:k})^{-1} log\left(\tilde{p}(x|z_{1:k})^{-1}\right)\right] dz \tag{25}$$

$$= -\int_z p(x,z)\int_{z_{2:k}} q(z_{2:k}|x)\tilde{p}(x|z_{1:k})^{-1} log\left(\tilde{p}(x|z_{1:k})^{-1}\right) dz \tag{26}$$

$$= -\int_{z_{1:k}} \frac{q(z_1|x)}{q(z_1|x)} p(x,z_1)q(z_{2:k}|x)\tilde{p}(x|z_{1:k})^{-1} log\left(\tilde{p}(x|z_{1:k})^{-1}\right) dz \tag{27}$$

$$= -\int_{z_{1:k}} \frac{p(x,z_1)}{q(z_1|x)} q(z_{1:k}|x)\tilde{p}(x|z_{1:k})^{-1} log\left(\tilde{p}(x|z_{1:k})^{-1}\right) dz \tag{28}$$

$$= \int_{z_{1:k}} \frac{\frac{p(x,z_1)}{q(z_1|x)}}{\tilde{p}(x|z_{1:k})} q(z_{1:k}|x) log\left(\tilde{p}(x|z_{1:k})\right) dz \tag{29}$$

$$= k\int_{z_{1:k}} \frac{\frac{p(x,z_1)}{q(z_1|x)}}{\sum_{j=1}^k \frac{p(x,z_j)}{q(z_j|x)}} q(z_{1:k}|x) log\left(\frac{1}{k}\sum_{j=1}^k \frac{p(x,z_j)}{q(z_j|x)}\right) dz \tag{30}$$

$$= \sum_{i=1}^k \int_{z_{1:k}} \frac{\frac{p(x,z_i)}{q(z_i|x)}}{\sum_{j=1}^k \frac{p(x,z_j)}{q(z_j|x)}} q(z_{1:k}|x) log\left(\frac{1}{k}\sum_{j=1}^k \frac{p(x,z_j)}{q(z_j|x)}\right) dz \tag{31}$$

$$= \int_{z_{1:k}} \frac{\sum_{i=1}^k \frac{p(x,z_i)}{q(z_i|x)}}{\sum_{j=1}^k \frac{p(x,z_j)}{q(z_j|x)}} q(z_{1:k}|x) log\left(\frac{1}{k}\sum_{j=1}^k \frac{p(x,z_j)}{q(z_j|x)}\right) dz \tag{32}$$

$$= \int_{z_{1:k}} q(z_{1:k}|x) log\left(\frac{1}{k}\sum_{j=1}^k \frac{p(x,z_j)}{q(z_j|x)}\right) dz \tag{33}$$

$$= E_{q(z_{1:k})}\left[log\left(\frac{1}{k}\sum_{j=1}^k \frac{p(x,z_j)}{q(z_j|x)}\right)\right] \tag{34}$$

$$= \mathcal{L}_1[q_K(z|x)] \tag{35}$$

Knowing that $\mathcal{L}_1[q_K(z|x)] \leq \mathcal{L}_K[q(z|x)]$ implies $q_K(z|x)$ is closer to the true posterior $p(z|x)$ than $q(z|x)$. We know that following equations are valid:

$$\log p(x) \geq E_{z_1...z_k\sim q(z|x)}\left[\log\left(\frac{1}{k}\sum_{i=1}^k \frac{p(x,z_i)}{q(z_i|x)}\right)\right] \tag{36}$$

$$\log p(x) \geq E_{z\sim q(z|x)}\left[\log\left(\frac{p(x,z)}{q(z|x)}\right)\right] \tag{37}$$

We want to show that the importance-weighted autoencoders is a tighter lower bound than the ELBO maximized by the variational autoencoder and show that the equation below is correct.

$$\log p(x) \geq E_{z_1 \ldots z_k \sim q(z|x)} \left[\log \left(\frac{1}{k} \sum_{i=1}^{k} \frac{p(x, z_i)}{q(z_i|x)} \right) \right] \qquad \geq E_{z \sim q(z|x)} \left[\log \left(\frac{p(x, z)}{q(z|x)} \right) \right] \tag{38}$$

Lets rewrite the log marginal likelihood equation:

$$log(p(x)) = E_{z \sim q(z|x)} \left[\log \left(\frac{p(x, z)}{q(z|x)} \right) \right] + KL(q(z|x)||p(x, z)) \longrightarrow$$

$$KL(q(z|x)||p(x, z)) = log(p(x)) - E_{z \sim q(z|x)} \left[\log \left(\frac{p(x, z)}{q(z|x)} \right) \right]$$

.

Using the above mentioned equations we can write :

$$KL(\tilde{q}(z|x, z_2, ..., z_K)||p(z|x)) = log(p(x)) - L_K[q_K(z|x)] \tag{39}$$

$$\leq log(p(x)) - L_1[q(z|x)] \tag{40}$$

$$\leq log(p(x)) - L_K[q_K(z|x)] = KL(q(z|x)||p(z|x)) \tag{41}$$

Now, we can conclude that $KL(\tilde{q}(z|x, z_2, ..., z_K)||p(z|x)) \leq KL(q(z|x)||p(z|x))$, meaning $\tilde{q}(z|x, z_2, ..., z_K)$ is closer to the true posterior $p(z|x)$ than $q(z|x)$ in terms of KL divergence.

Question 5 (5-5-5-6)**.** Normalizing flows are expressive invertible transformations of probability distributions. In this exercise, we will see how to satisfy the invertibility constraint of some family of parameterizations. For the first 3 questions, we assume the function $g : \mathbb{R} \to \mathbb{R}$ maps from real space to real space.

1. Let $g(z) = af\,(bz + c)$ where f is the ReLU activation function $f(x) = \max(0, x)$. Show that g is non-invertible.
2. Let $g(z) = \sigma^{-1}(\sum_{i=1}^{N} w_i\sigma(a_i z + b_i))$, $0 < w_i < 1$, where $\sum_i w_i = 1$, $a_i > 0$, and $\sigma(x) = 1/(1 + \exp(-x))$ is the logistic sigmoid activation function and σ^{-1} is its inverse. Show that g is *strictly monotonically increasing* on its domain $(-\infty, \infty)$, which implies invertiblity.
3. Consider a residual function of the form $g(z) = z + f(z)$. Show that $df/dz > -1$ implies g is invertible.
4. Consider the following transformation:

$$g(\boldsymbol{z}) = \boldsymbol{z} + \beta h(\alpha, r)(\boldsymbol{z} - \boldsymbol{z}_0) \tag{42}$$

where $\boldsymbol{z}_0 \in \mathbb{R}^D$, $\alpha \in \mathbb{R}^+$, $\beta \in \mathbb{R}$, and $r = ||\boldsymbol{z} - \boldsymbol{z}_0||_2$, $h(\alpha, r) = 1/(\alpha + r)$. Consider the following decomposition of $\boldsymbol{z} = \boldsymbol{z}_0 + r\tilde{\boldsymbol{z}}$. (i) Given $\boldsymbol{y} = g(\boldsymbol{z})$, show that $\beta \geq -\alpha$ is a sufficient condition to derive the unique r from equation (42). (ii) Given r and $\boldsymbol{y}$, show that equation (42) has a unique solution $\tilde{\boldsymbol{z}}$.

Answer 5. 1. For every $bz + c \leq 0$ the output of ReLU is zero. Since ReLU is not a one-to-one function its inverse will not be a function. So it is non-invertible.

2. First, we need to calculate the σ^{-1} formula.
$\sigma(\boldsymbol{x}) = 1/(1 + \exp(-\boldsymbol{x})) \to \boldsymbol{y} = 1/(1 + \exp(-\boldsymbol{x})) \to \exp(-\boldsymbol{x}) = \frac{1-\boldsymbol{y}}{\boldsymbol{y}} \to -\boldsymbol{x} = \ln\frac{1-\boldsymbol{y}}{\boldsymbol{y}}$
$\boldsymbol{x} = \ln\frac{\boldsymbol{y}}{1-\boldsymbol{y}} \to \sigma^{-1}(\boldsymbol{x}) = \ln\frac{\boldsymbol{x}}{1-\boldsymbol{x}}$ Now we can rewrite $g(\boldsymbol{z})$ using the above formula.

$$g(z) = \sigma^{-1}(\sum_{i=1}^{N} w_i\sigma(a_i z + b_i)) = \ln\frac{\sum_{i=1}^{N} w_i\sigma(a_i z + b_i)}{1 - \sum_{i=1}^{N} w_i\sigma(a_i z + b_i)}$$

We know that $\sum_i w_i = 1$, and $a_i > 0$ so we can conclude that $a_i z + b_i$ is *strictly monotonically increasing* for any $\boldsymbol{z}$. We also know that $\sigma(a_i z + b_i)$ is *strictly monotonically increasing* and $\ln \boldsymbol{x}$ is *strictly monotonically increasing* when $\boldsymbol{x} \geq 0$. As a result, we can conclude $g(z)$ is *strictly monotonically increasing.*

3. If we can show that g is *strictly monotonically increasing* then we can be sure that it is invertible as being strictly monotonically increasing is a necessary and sufficient condition for invertibility. Lets calculate the gradients.

$$g(\boldsymbol{z}) = \boldsymbol{z} + f(\boldsymbol{z}) \xrightarrow{gradient} \frac{dg}{d\boldsymbol{z}} = 1 + \frac{df}{d\boldsymbol{z}} \xrightarrow{df/dz>-1} \frac{dg}{d\boldsymbol{z}} \geq 0$$

Therefor, $df/dz > -1$ implies g is *strictly monotonically increasing.* Which means g is invertible.

4.

Question 6 (4-3-6)**.** In this question, we are concerned with analyzing the training dynamics of GANs. Consider the following value function

$$V(d, g) = dg \tag{43}$$

with $g \in \mathbb{R}$ and $d \in \mathbb{R}$. We will use this simple example to study the training dynamics of GANs.

1. Consider gradient descent/ascent with learning rate α as the optimization procedure to iteratively minimize $V(d, g)$ w.r.t. g and maximize $V(d, g)$ w.r.t. d. We will apply the gradient descent/ascent to update g and d simultaneously. What is the update rule of g and d? Write your answer in the following form
$$[d_{k+1}, g_{k+1}]^\top = A[d_k, g_k]^\top$$
where A is a 2×2 matrix; i.e. specify the value of A.
2. The optimization procedure you found in 6.1 characterizes a map which has a stationary point [6], what are the coordinates of the stationary points?
3. Analyze the eigenvalues of A and predict what will happen to d and g as you update them jointly. In other word, predict the behaviour of d_k and g_k as $k \to \infty$.

Answer 6. 1. Matrix A in equation

$$[d_{k+1}, g_{k+1}]^\top = A[d_k, g_k]^\top$$

is a 2 by 2 matrix. Lets $A = \begin{bmatrix} a, b \\ c, d \end{bmatrix}$ at step k. So we can calculate the gradients of V.

$$\begin{bmatrix} d_{k+1} \\ g_{k+1} \end{bmatrix} = \begin{bmatrix} a, b \\ c, d \end{bmatrix} \begin{bmatrix} d_k \\ g_k \end{bmatrix} = \begin{bmatrix} a_k, & b \cdot g_k \\ c \cdot d_k, & d \cdot g_k \end{bmatrix}$$

Now we need to find matrix A parameters in such a way that minimize $V(d, g)$ w.r.t. g and maximize $V(d, g)$ w.r.t. d. One possible solution is:

$$\begin{bmatrix} 1, \alpha \\ -\alpha, 1 \end{bmatrix}$$

The above mentioned matrix will maximize $V(d, g)$ w.r.t. d and minimize $V(d, g)$ w.r.t. g.

2. If we set (d_i, g_i) to $(0, 0)$ then (d_{i+1}, g_{i+1}) is equal to zero which means the point $(0, 0)$ is a stationary point for the optimization we found.
3. There might be other possible combinations for matrix A which fits better to this problem but using the values I found one can derive the eigenvalues in what follows:

$$\begin{bmatrix} 1 - \lambda, \alpha \\ -\alpha, 1 - \lambda \end{bmatrix} \to (1 - \lambda)^2 + (\alpha)^2 = 0 \to \alpha = 0, \quad \lambda = 1$$

Which means when $\alpha = 0$ the value of eigenvalue is one but we do not have any update in that case. When α is not equal to zero the model does not have any eigenvalue and it will explode. So it depends on the value of $alpha$. It also depends on initialization of d_0 and g_0. But generally speaking we know that if the eigenvalues are not equal to 1 they will explode after few updates.

6. A stationary point is a point on the surface of the graph (of the function) where all its partial derivatives are zero (equivalently, the gradient is zero). Source: https://en.wikipedia.org/wiki/Stationary_point