CS 236, Fall 2018
Midterm Exam

This exam is worth 130 points. You have 3 hours to complete it. You are allowed to consult notes, books, and use a laptop but no communication or network access is allowed. Good luck!

Stanford University Honor Code

The Honor Code is the University's statement on academic integrity written by students in 1921. It articulates University expectations of students and faculty in establishing and maintaining the highest standards in academic work:

- The Honor Code is an undertaking of the students, individually and collectively:
  - that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
  - that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.
- The faculty on its part manifests its condence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.
- While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

Signature

I attest that I have not given or received aid in this examination, and that I have done my share and taken an active part in seeing to it that others as well as myself uphold the spirit and letter of the Stanford University Honor Code.

Name / SUnetID:

Signature:

Question	Score	Question	Score
1	/ 15	5	/ 20
2	/ 20	6	/ 20
3	/ 15	7	/ 20
4	/ 20		
	Total score:		**/ 130**

Note: Partial credit will be given for partially correct answers. Zero points will be given to answers left blank.

---

1. **[15 points total] Comparison of Models**

   In this course, we discussed four major types of generative models: autoregressive models, variational autoencoders, flow models, and generative adversarial networks.

   (a) **[5 points]** Suppose we are interested in quickly generating i.i.d. samples from a trained model. Which of these models can we sample from efficiently (i.e., in a time polynomial in the number of dimensions of a sample, such as in linear time)?

   Answer: All of them.

   (b) **[5 points]** Suppose we are interested in exactly evaluating the likelihood of a data point under the trained model. In which of these models can we exactly evaluate a data point's likelihood in an efficient way (i.e., in a time polynomial in the number of dimensions of a sample, such as in linear time)?

   Answer: Autoregressive and flow models allow for exact likelihood evaluation.

   (c) **[5 points]** Suppose we are interested in learning a latent representation for new data points. Which of these models are most appropriate for this task, and why?

   Answer: In a VAE, $q_\phi(z|x)$ can serve as an encoder. In a flow model, $f^{-1}(x)$ can serve as an encoder. GAN can also learn representations via Bi-GAN.

2. **[20 points total] Masked Autoregressive Distribution Estimation (MADE)**

   An autoencoder learns a feed-forward, hidden representation $h(\mathbf{x})$ of its input $\mathbf{x} \in \mathbb{R}^D$ such that, from it, we can obtain a reconstruction $\hat{\mathbf{x}}$ which is as close as possible to $\mathbf{x}$. Specifically, we have

   $$h(\mathbf{x}) = g(\mathbf{b} + \mathbf{W}\mathbf{x})$$
   $$\hat{\mathbf{x}} = \text{sigmoid}(\mathbf{c} + \mathbf{V}h(\mathbf{x}))$$

   where $\mathbf{W}$ and $\mathbf{V}$ are weight matrices, $\mathbf{b}$ and $\mathbf{c}$ are bias vectors, g is a nonlinear activation function and $\text{sigmoid}(a) = 1/(1 + \exp(-a))$.

   MADE modifies the autoencoder to build an autoregressive model. To satisfy the autoregressive property $p(\mathbf{x}) = \prod_{d=1}^{D} p(x_d \mid \mathbf{x}_{<d})$, we use the d^{th} output of MADE $\hat{x}_d$ to parameterize the conditional probability $p(x_d \mid \mathbf{x}_{<d})$, which means that $\hat{x}_d$ must depend only on the preceding inputs $\mathbf{x}_{<d}$. In order to enforce this property, MADE multiplies each weight matrix by a mask matrix. For a single hidden layer autoencoder, we write

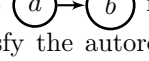
   $$h(\mathbf{x}) = g(\mathbf{b} + (\mathbf{W} \odot \mathbf{M}^{\mathbf{W}})\mathbf{x})$$
   $$\hat{\mathbf{x}} = \text{sigmoid}(\mathbf{c} + (\mathbf{V} \odot \mathbf{M}^{\mathbf{V}})h(\mathbf{x}))$$

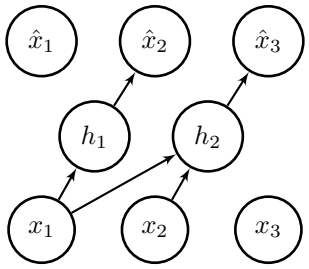
   where $\mathbf{M}^{\mathbf{W}}$ and $\mathbf{M}^{\mathbf{V}}$ are the masks for $\mathbf{W}$ and $\mathbf{V}$ respectively, and $\odot$ denotes element-wise multiplication. Note that the entries of a mask matrix can only be 0 or 1.

   In this question, we consider a MADE model with a single hidden layer. The dimension of the input $\mathbf{x}$ is assumed to be D, and the number of hidden units $h(\mathbf{x})$ is $D - 1$.

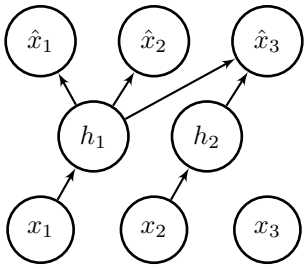
   (a) **[2 points]** What are the shapes (number of rows and columns) of the mask matrices $\mathbf{M}^{\mathbf{W}}$ and $\mathbf{M}^{\mathbf{V}}$?

   Answer: Shape of $\mathbf{M}^{\mathbf{W}}$: $(D-1, D)$ Shape of $\mathbf{M}^{\mathbf{V}}$: $(D, D-1)$

   (b) **[8 points]** Consider two candidate MADE models as shown below. For both models, $D = 3$, and there are 2 hidden units. In the figures, an arrow connecting two neurons  indicates that the value of b depends on the value of a. Check whether they satisfy the autoregressive property. If yes, write down the mask matrices $\mathbf{M}^{\mathbf{W}}$, $\mathbf{M}^{\mathbf{V}}$ and compute $\mathbf{M}^{\mathbf{V}}\mathbf{M}^{\mathbf{W}}$. Else, explain why the autoregressive property is violated.



(a)



(b)

Answer: (a) is a valid MADE. (b) is not because $\hat{x}_1$ cannot depend on x_1.

$$\mathbf{M^W} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} \quad \mathbf{M^V} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \mathbf{M^V M^W} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

(c) **[5 points]** Let $\mathbf{M} = \mathbf{M^V M^W}$. What is the maximum number of non-zero entries that $\mathbf{M}$ can have in order to preserve the autoregressive property? Briefly explain your results. [Hint: The answer should be a function of D.]
Answer: $\frac{D(D-1)}{2}$. The matrix $\mathbf{M}$ is always strictly lower triangular.

(d) **[5 points]** It can often be advantageous to have direct connections between the input $\mathbf{x}$ and output layer $\hat{\mathbf{x}}$. In this context, the reconstruction part of MADE becomes

$$\hat{\mathbf{x}} = \text{sigmoid}(\mathbf{c} + (\mathbf{V} \odot \mathbf{M^V})h(\mathbf{x}) + (\mathbf{A} \odot \mathbf{M^A})\mathbf{x}),$$

where $\mathbf{A}$ is the weight matrix that directly connects input and output, and $\mathbf{M^A}$ is its mask matrix. What is the maximum number of non-zero entries that $\mathbf{M^A}$ can have to satisfy the autoregressive property? Briefly explain your results. [Hint: The answer should be a function of D.]
Answer: $\frac{D(D-1)}{2}$. The matrix $\mathbf{M^A}$ is always strictly lower triangular.

3. **[15 points total] Variational Autoencoders Basics**

For each of the following questions, state true or false. Explain your answer for full points.

(a) **[5 points]** Suppose we are training a VAE where the prior $p(\mathbf{z})$ is such that each dimension of the latent variable $\mathbf{z}$ is Bernoulli distributed. We can use reparameterization with $\mathbf{z}$ to get an unbiased estimate of the gradient of the variational objective function. (False)

(b) **[5 points]** Suppose we have trained a VAE parameterized by ϕ and θ. We can obtain a sample from $p_{\text{data}}(\mathbf{x})$ by first drawing a sample $\mathbf{z}' \sim p_\theta(\mathbf{z})$, then drawing another sample $\mathbf{x}' \sim p_\theta(\mathbf{x}|\mathbf{z}')$. (False, but True if they say that the VAE is optimal)

(c) **[5 points]** After learning a VAE model on a dataset, Alice gives Bob the trained decoder $p_\theta(\mathbf{x}|\mathbf{z})$ and the prior $p(\mathbf{z})$ she used. However, she forgets to give Bob the encoder. Given sufficient computation, can Bob still infer a latent representation for a new test point $\mathbf{x}'$? (True)

4. **[20 points total] Evidence Lower Bound**

Consider the joint distribution of a latent variable model denoted by $p(\mathbf{x}, z)$. The model is capable of sampling only two images $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}$. You may imagine that these are two binarized MNIST images where $\mathbf{x}^{(i)} \in \{0,1\}^{784}$. Note that this latent variable model is equipped with a scalar latent variable $z \in \mathbb{R}$. Furthermore, this model is described by

$$p(z) = \mathcal{N}(z; 0, 1)$$

$$p(\mathbf{x} \mid z) = \begin{cases} 1 \text{ if } z \geq 0 \wedge \mathbf{x} = \mathbf{x}^{(1)} \\ 0 \text{ if } z \geq 0 \wedge \mathbf{x} \neq \mathbf{x}^{(1)} \\ 1 \text{ if } z < 0 \wedge \mathbf{x} = \mathbf{x}^{(2)} \\ 0 \text{ if } z < 0 \wedge \mathbf{x} \neq \mathbf{x}^{(2)} \end{cases}$$

where $p(\mathbf{x} \mid z)$ is a probability mass function and $p(z)$ is a probability density function.

In other words, the generative model will *always* sample the first image $\mathbf{x}^{(1)}$ when conditioned on $z \geq 0$, and the model will *always* sample the second image $\mathbf{x}^{(2)}$ when conditioned on $z < 0$. $\mathcal{N}(z; 0, 1)$ indicates a Gaussian distribution with mean zero and variance 1.

(a) **[4 points]** We can consider the log-likelihood of some image $\mathbf{x}$ under our model

$$\ell_{\text{like}}(\mathbf{x}) := \log p(\mathbf{x}).$$

Based on the model described above, what is $\ell_{\text{like}}(\mathbf{x}^{(1)})$?
Answer: $\ell_{\text{like}}(\mathbf{x}^{(1)}) = \log 0.5 = -\log 2$

(b) **[2 points]** One can also reason about the posterior $p(z \mid \mathbf{x})$. What is the set of all points z for which the posterior density is positive $p(z \mid \mathbf{x}^{(1)}) > 0$ when conditioned on $\mathbf{x}^{(1)}$?
Answer: $[0, +\infty)$

(c) **[8 points]** The log-likelihood can be lower bounded by the Evidence Lower Bound (ELBO) as follows:

$$\ell_{\text{ELBO}}(\mathbf{x}\,; q) := \mathbb{E}_{q(z)}\left[\log \frac{p(\mathbf{x}, z)}{q(z)}\right],$$

Note that this lower bound is a function of some variational distribution $q(z)$. Prove that

$$\mathbb{E}_{q(z)}\left[\log \frac{p(\mathbf{x}, z)}{q(z)}\right] = \log p(\mathbf{x}) - D_{\text{KL}}(q(z) \parallel p(z \mid \mathbf{x})).$$

Answer:

$$\mathbb{E}_{q(z)}\left[\log \frac{p(\mathbf{x}, z)}{q(z)}\right] = \log p(\mathbf{x}) + \mathbb{E}_{q(z)}\left[\log \frac{p(z \mid \mathbf{x})}{q(z)}\right] \tag{1}$$

$$= \log p(\mathbf{x}) - \mathbb{E}_{q(z)}\left[\log \frac{q(z)}{p(z \mid \mathbf{x})}\right] \tag{2}$$

$$= \log p(\mathbf{x}) - D_{\text{KL}}(q(z) \parallel p(z \mid \mathbf{x})). \tag{3}$$

(d) **[2 points]** For parts (d) and (e), suppose $q(z)$ is a univariate Gaussian distribution with positive variance. What is the set of all points z for which $q(z) > 0$?
Answer: $\mathbb{R}$

(e) **[4 points]** Using what you have determined so far, select the option that is correct.
The log-likelihood $\ell_{\text{like}}(\mathbf{x}^{(1)})$ is:

i. Finite and negative
ii. 0
iii. Finite and positive
iv. None of the above

Answer: Finite and negative.

For any q that is univariate Gaussian with positive variance, $\ell_{\text{ELBO}}(\mathbf{x}^{(1)}; q)$ is:

i. Finite and negative
ii. 0
iii. Finite and positive
iv. None of the above

Answer: None of the above. The KL divergence is undefined since $q(z)$ assigns non-zero probability mass to the interval $(-\infty, 0)$, but $p(z \mid \mathbf{x}^{(1)})$ assigns zero probability mass to that interval.

5. **[20 points total] Normalizing Flow Models Basics**

(a) **[5 points]** Let $Z \sim \text{Uniform}[-2, 3]$ and $X = \exp(Z)$. What is $p_X(5)$?. **Answer:** (True, by change of variables $p_X(x) = \frac{1}{x} p_Z(\log x) = 1/25$)
For each of the statements below, state true or false. Explain your answer for full points.

(b) **[5 points]** For efficient learning and inference in flow models, any discrete or continuous distribution which allows for efficient sampling and likelihood evaluation can be used to specify the prior distribution over latent variables.
Answer: (False, only continuous distributions can be used.)

(c) **[5 points]** In Parallel Wavenet, evaluating the likelihood assigned by the student model for any external data point is computationally intractable (i.e., requires exponential time in the number of dimensions of the sample).
Answer: (False, they are expensive to compute but not computationally intractable.)

(d) **[5 points]** A permutation matrix is defined as a binary square matrix with $\{0, 1\}$ entries such that every column and every row sums to 1. The Jacobian for a RealNVP model can be expressed as the product of a series of (upper or lower) triangular matrices and permutation matrices.
Answer: (True, a permutation matrix does not affect invertibility)

6. **[20 points total] Flow + GAN: Maximum Likelihood vs. Adversarial Training**

Let $p_{\text{data}}(\mathbf{x})$ denote a data distribution that we are trying to learn with a generative model, where $\mathbf{x} \in \mathbb{R}^n$. Consider the simple generative model parameterized by a single invertible matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, where a sample is obtained by first sampling an n-dimensional vector $\mathbf{z} \sim p(\mathbf{z})$ from a given distribution $p(\mathbf{z})$, and returning the matrix-vector product $\mathbf{Az}$. Let $\mathcal{D}$ be a training set of samples from $p_{\text{data}}(\mathbf{x})$.

(a) **[10 points]** Write a loss function $\mathcal{L}(\mathbf{A})$ that trains this model using maximum likelihood.
Answer:

$$\mathcal{L}(\mathbf{A}) = -\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[\log p(\mathbf{A}^{-1}\mathbf{x}) + \log |\det(\mathbf{A}^{-1})|]$$

(b) **[10 points]** Write a loss function $\mathcal{L}(\mathbf{A})$ that trains this model as the generator in a generative adversarial network. You may assume that a discriminator $D_\phi : \mathbb{R}^n \to \mathbb{R}$ that outputs the probability that the input is real has been defined and is trained alongside the generative model.
Answer:

$$\mathcal{L}(\mathbf{A}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[\log(1 - D_\phi(\mathbf{Az})]$$

or

$$\mathcal{L}(\mathbf{A}) = -\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[\log D_\phi(\mathbf{Az})]$$

7. **[20 points total] Flow + VAE: Augmenting variational posteriors**

We wish to use flexible flow models for variational autoencoders. Let $\mathbf{x} \in \mathbb{R}^D$ denote the inputs, $\mathbf{z}$ the latent variables, $p_\theta(\mathbf{x}|\mathbf{z})$ the generative model, $p(\mathbf{z})$ the prior and $r_\phi(\mathbf{z}|\mathbf{x})$ as the basic inference model representing a Gaussian distribution $\mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi(\mathbf{x})^2))$.

Instead of using $r_\phi(\mathbf{z}|\mathbf{x})$ directly as the inference model, we will transform this distribution using a normalizing flow to obtain a richer variational posterior distribution $q_{\phi,\psi}$. Specifically, let $f_\psi : \mathbb{R}^F \to \mathbb{R}^F$ be an invertible transformation, $\mu_\phi : \mathbb{R}^D \to \mathbb{R}^F$, and $\sigma_\phi : \mathbb{R}^D \to \mathbb{R}^F$. We use the following procedure to sample $\mathbf{z}$ from $q_{\phi,\psi}(\mathbf{z}|\mathbf{x})$ given $\mathbf{x}$:

- Sample $\tilde{\mathbf{z}} \sim \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi(\mathbf{x})^2))$
- Compute $\mathbf{z} = f_\psi(\tilde{\mathbf{z}})$

(a) **[8 points]** Derive an expression for $\log q_{\phi,\psi}(\mathbf{z}|\mathbf{x})$. The function should take $\mathbf{x}$ and $\mathbf{z}$ as input, output a scalar value, and depend on μ_ϕ, σ_ϕ, and f_ψ. You can use $\mathcal{N}(u; \mu, \text{diag}(\sigma^2))$ to denote the pdf for normal distribution with mean μ and covariance $\text{diag}(\sigma^2)$ evaluated at u.

Answer: The log probability of $\hat{\mathbf{z}}$ is:

$$\log \mathcal{N}(\hat{\mathbf{z}}; \mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi(\mathbf{x})^2))$$

and since $\mathbf{z} = f(\hat{\mathbf{z}})$, we have

$$q_{\phi,\psi}(\mathbf{z}|\mathbf{x}) = \log \mathcal{N}(f_\psi^{-1}(\mathbf{z}); \mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi(\mathbf{x})^2)) + \log |\det \frac{\partial f_\psi^{-1}(\mathbf{z})}{\partial \mathbf{z}}|$$

(b) **[12 points]** Consider $r_\phi(\mathbf{z}|\mathbf{x})$ as the basic Gaussian inference model (without using the flow model), with the following sampling process:

- Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi(\mathbf{x})^2))$

Show that the best evidence lower bound we can achieve with $q_{\phi,\psi}$ is at least as tight as the best one we can achieve with r_ϕ, i.e.,

$$\max_{\theta,\phi,\psi} \text{ELBO}(\mathbf{x}; p_\theta, q_{\phi,\psi}) \geq \max_{\theta,\phi} \text{ELBO}(\mathbf{x}; p_\theta, r_\phi)$$

where

$$\text{ELBO}(\mathbf{x}; p, q) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}|\mathbf{x})]$$

You may assume that f_ψ can represent any invertible function $\mathbb{R}^F \to \mathbb{R}^F$.

Answer: For any instance of ϕ selected for r_ϕ, we can always choose f_ψ to be the identity function for $q_{\phi,\psi}$. This function is invertible and preserves volume. Therefore, for this instance of ψ,

$$q_{\phi,\psi}(\mathbf{z}|\mathbf{x}) = r_\phi(\mathbf{z}|\mathbf{x})$$

and for any solution r_ϕ, we have a $q_{\phi,\psi}$ that has the same ELBO. Therefore, the maximum ELBO of $q_{\phi,\psi}$ should be greater or equal to that of r_ϕ.