F23-CS-236-01 Midterm

Jason Alexander Chan

TOTAL POINTS

59.4 / 95

QUESTION 1

Question 1 20 pts

1.1 a **0 / 2**

- **0 pts** Correct
- **1 pts** Incorrect or insufficient explanation.

✓ **- 2 pts** *Incorrect or missing response.*

1.2 b **2 / 2**

✓ **- 0 pts** *Correct*

- **1 pts** Incorrect or insufficient explanation.
- **2 pts** Incorrect or missing response.

1.3 c **2 / 2**

✓ **- 0 pts** *Correct*

- **1 pts** Incorrect or insufficient explanation.
- **2 pts** Incorrect or missing response.

1.4 d **2 / 2**

✓ **- 0 pts** *Correct*

- **1 pts** Incorrect or insufficient explanation.
- **2 pts** Incorrect or missing response.

1.5 e **0 / 2**

- **0 pts** Correct
- **1 pts** Incorrect or insufficient explanation.

✓ **- 2 pts** *Incorrect or missing response.*

1.6 f **2 / 2**

✓ **- 0 pts** *Correct*

- **1 pts** Incorrect or no explanation
- **2 pts** Incorrect
- **1 pts** Correct idea but the answer should be false

1.7 g **2 / 2**

✓ **- 0 pts** *Correct*

- **2 pts** Click here to replace this description.

1.8 h **0 / 2**

- **0 pts** Correct

✓ **- 2 pts** *Click here to replace this description.*

- **1 pts** incorrect or no explanation

1.9 i **1 / 2**

- **0 pts** Correct
- **2 pts** Click here to replace this description.

✓ **- 1 pts** *incorrect or missing explanation*

1.10 j **2 / 2**

✓ **- 0 pts** *Correct*

- **2 pts** Click here to replace this description.

QUESTION 2

Question 2 10 pts

2.1 a **1.6 / 2**

✓ **+ 0.4 pts** *AR*

✓ **+ 0.4 pts** *VAE*

✓ **+ 0.4 pts** *EBM*

+ 0.4 pts no GAN

✓ **+ 0.4 pts** *no Flow*

+ 0 pts Click here to replace this description.

2.2 b **2 / 2**

✓ **+ 0.4 pts** *AR*

✓ **+ 0.4 pts** *Flow*

✓ **+ 0.4 pts** *no EBM*

✓ **+ 0.4 pts** *no VAE*

✓ **+ 0.4 pts** *no GAN*

+ 0 pts Click here to replace this description.

2.3 c **2 / 2**

✓ **+ 0.4 pts** *AR*

✓ **+ 0.4 pts** *VAE*

✓ **+ 0.4 pts** *Flow*

✓ **+ 0.4 pts** *GAN*

✓ **+ 0.4 pts** *EBM*

+ 0 pts Click here to replace this description.

2.4 d **0.8 / 2**

+ 2 pts Correct

✓ **+ 0.4 pts** *AR*

+ 0.4 pts VAE

+ 0.4 pts Flow

+ 0.4 pts GAN

✓ **+ 0.4 pts** *EBM*

+ 0 pts Incorrect

2.5 e **2 / 2**

+ 2 pts Correct

✓ **+ 0.4 pts** *No AR*

✓ **+ 0.4 pts** *VAE*

✓ **+ 0.4 pts** *Flow*

✓ **+ 0.4 pts** *GAN*

✓ **+ 0.4 pts** *EBM*

+ 0 pts Incorrect

QUESTION 3

Question 3 18 pts

3.1 a **0 / 3**

- 0 pts Correct

✓ **- 3 pts** *Wrong*

- 1 pts Correct but fails to mention prior needs a tractable likelihood

3.2 b **3 / 3**

✓ **- 0 pts** *Correct*

- 3 pts Wrong

- 1 pts Correct but insufficient explanation

3.3 c **0 / 3**

- 0 pts Correct

✓ **- 3 pts** *incorrect*

- 3 pts missing

3.4 d **3 / 3**

✓ **- 0 pts** *Correct*

- 3 pts reasoning incorrect/missing

- 2 pts correct idea, not rigorous

3.5 e **3 / 3**

✓ **- 0 pts** *Correct*

- 3 pts Incorrect

- 2 pts Does not include an explanation.

- 1 pts No mentioning of reparametrization trick

on Z / Z is still a gaussian distribution.

3.6 f **0 / 3**

- **0 pts** Correct

✓ **- 3 pts** *Incorrect*

- **1 pts** No explanation.
- **1 pts** Wrong explanation.

QUESTION 4

Question 4 10 pts

4.1 a **2 / 2**

✓ **- 0 pts** *Correct: 1/3 * 1/5 = 1/15*

- **0.5 pts** Partially wrong: give the correct expression of x but not the answer 1/15
- **1 pts** Partially wrong: either Jacobian (1/3) or $$p_z(f^{-1}(x))$$ (1/5) is wrong
- **2 pts** Wrong

4.2 b **2 / 2**

✓ **- 0 pts** *Correct: No because not invertible*

- **1 pts** True/False is correct but the explanation is wrong or missing
- **2 pts** Wrong

4.3 c **2 / 2**

✓ **- 0 pts** *Correct: Yes, because there is a unique inverse*

- **1 pts** True/False is correct but the explanation is wrong or missing
- **2 pts** Wrong

4.4 d **0 / 4**

✓ **- 2 pts** *Incorrect Jacobian*

✓ **- 2 pts** *Incorrect joint pdf*

- **0.5 pts** Minor error(i.e. x,y still in the equation/did not specify variables in P_xy/minor error in calculation)
- **4 pts** Missing
- **0 pts** Correct

QUESTION 5

Question 5 11 pts

5.1 a **0 / 2**

- **0 pts** Correct
- **1 pts** Partially correct

✓ **- 2 pts** *Incorrect/missing*

5.2 b **1 / 2**

- **0 pts** Correct

✓ **- 1 pts** *Partially Correct*

- **2 pts** Incorrect/missing

5.3 c **2 / 2**

✓ **- 0 pts** *Correct*

- **2 pts** Incorrect/missing
- **1 pts** Partially Correct

5.4 d **4 / 4**

✓ **- 0 pts** *Correct*

- **3 pts** Missing expectation for the KL-Divergence
- **3 pts** Missing / unclear dependencies on t
- **2 pts** Incomplete
- **4 pts** Answer missing

5.5 e **1 / 1**

✓ **- 0 pts** *Correct*

- **1 pts** Incorrect. Correct answer: "Yes, the

sampling step can be rewritten as a deterministic transformation of standard
Gaussians by unrolling the RNN."

QUESTION 6

Question 6 10 pts

6.1 a **1 / 2**

- 0 pts Correct

✓ **- 1 pts** *Incorrect logic / missing logic / missing mathematical formalization*

- 2 pts Incorrect / missing

6.2 b **2 / 2**

✓ **- 0 pts** *Correct*

- 1 pts Minor Error / Did not simplify / Did not follow instructions

- 2 pts Incorrect

6.3 c **2 / 2**

✓ **- 0 pts** *Correct*

- 1 pts Minor Error / Did not simplify / Did not follow instructions

- 2 pts Incorrect

6.4 d **1 / 4**

- 0 pts Correct

✓ **- 1 pts** *D(G(z)) value at the beginning of training*

- 2 pts magnitudes of gradients in training

✓ **- 1 pts** *training speed*

- 4 pts Incorrect or missing

✓ **- 1 pts** *Minor Error*

QUESTION 7

Question 7 16 pts

7.1 a **2 / 2**

✓ **- 0 pts** *Correct*

- 1 pts on the right track, but did not explicitly say Z(1)=1, or wrong Z(1)

- 1 pts asking for Z(1) instead of log Z(1)

- 2 pts incorrect

7.2 b **0 / 2**

- 0 pts Correct

✓ **- 2 pts** *incorrect*

- 1 pts right track, but didn't answer explicitly tractable or not

7.3 c **2 / 2**

✓ **- 0 pts** *Correct*

- 2 pts incorrect

- 1 pts too vague, not explicitly 0<T<1

- 1 pts T shoud be strictly less than 1

7.4 d **1 / 2**

- 0 pts Correct

✓ **- 1 pts** *Sampling from p(x) is tractable*

- 1 pts Sampling from p^T(x) is intractable

7.5 e **2 / 2**

✓ **- 0 pts** *Correct*

- 1 pts Minor incorrect

- 2 pts Incorrect

7.6 f **0 / 4**

- 0 pts Correct

- 2 pts Incomplete/slightly incorrect

✓ **- 4 pts** *Incorrect/Not answered*

- 1 pts Minor missing steps

7.7 g **2 / 2**

✓ **- 0 pts** *Correct*

- 2 pts incorrect

CS 236, Fall 2023
Midterm Exam

This exam is worth 95 points. You have 3 hours to complete and submit it. You are allowed to consult notes, books and use a laptop. But no communication or network access is allowed. Use of Large Language Models (LLMs) is also not 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 confidence 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.

	Question	Score	Question	Score
target 10.40am	1	/ 20	12.20 5	/ 11
11 am	2	/ 10	12.40 6	/ 10
11.40am	3	/ 18	1.00pm 7	/ 16
12pm	4	/ 10		
	Total score:	**/ 95**		

180 mins / 100 pts = 1.8 mins / 1 pt.

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

Name: Jason Alexander Chan SUNet ID: jchan7 @stanford.edu

Stanford Honor Code. 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 Honor Code.

Sign: 

1. **[20 points total] True/False**

   For each of the statements below, state True or False. Explain your answer for full points.

   (a) **[2 points] Autoregressive models**:
   For discrete autoregressive models, the order of variables in the chain rule factorization cannot affect the minimal number of parameters required to represent exactly the joint distribution.

   True The number of params to represent the Exact joint isn't affected by variable ordering. This isn't true for non-exact joint distribution

   (b) **[2 points] Autoregressive Model**: Let $p(\mathbf{x})$ be an arbitrary autoregressive model over a set of discrete random variables. The probability assigned by p to a sequence $\mathbf{x} = (x_1, x_2, \ldots x_n)$ of length n is always at least as large as the probability assigned by p to any sequence $(x_1, x_2, \ldots x_n, x_{n+1})$ of length $n+1$ of which $\mathbf{x}$ is a prefix.

   True. $P(x_1, x_2 \ldots x_{n+1}) = P(x_{n+1} \mid x_{<n+1})\, P(x_1, x_2 \ldots x_n)$ chain rule.
   if $P(x_{n+1} \mid x_{n+1}) = 1$ then $P(x_1, x_2 \ldots x_{n+1}) = P(x_1, x_2 \ldots x_n)$

   (c) **[2 points] VAE**: In a VAE with two binary latent variables $z_1 \in \{0, 1\}$, $z_2 \in \{0, 1\}$ and a Gaussian decoder $p(x|z_1, z_2)$, the marginal likelihood $p(x)$ is intractable to compute.

   False $2^2 = 4$ possible discrete z states to sum over to arrive @ $p(x) = \int p(x, z)\, dz$

   (d) **[2 points] ELBO**: Let the inference distribution of a VAE be $q_\phi(z|x)$ and prior distribution be $p(z)$, the gap between the MLE objective and the ELBO lower bound for a datapoint x is exactly $D_{\mathrm{KL}}(q_\phi(z|x) \| p(z))$.

   False The gap is exactly $D_{KL}(q_\phi(z|x) \,||\, p(z|x))$ not wrt $p(z)$

   (e) **[2 points] ELBO**: Alice and Bob train two VAEs on the same dataset. The VAEs are identical, except for the priors over the latents which are different. Alice's model achieves a better (higher) ELBO on the training set. This implies Alice's model also achieves higher log-likelihood on the training set.

   True Higher ELBO correlates with higher log likelihood during training for VAE. $\log(p_x(\mathbf{x})) \geq E_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x) \,||\, p(z))$

   (f) **[2 points] Normalizing-Flow**: A normalizing flow based model with latent variables $\mathbf{z} \in \mathbb{R}^{100}$ can be used to model (with tractable exact likelihood evaluation) any random vector $\mathbf{x}$ of dimension less than or equal to 100.

   False normalizing flow map $z \to x$ & $x \to z$ iff they have the same dimensions

   (g) **[2 points] GAN**: A VAE with a Gaussian prior and a Gaussian decoder can be trained as the generator in a GAN (discarding the encoder and using a separate classifier as a discriminator).

   True. GANs can flexibly accept any model family as the generator.

   (h) **[2 points] GAN**: A conditional GAN can be implemented by adding the class information as an additional input to the Generator $G(z)$, without changes to the Discriminator $D(x)$.

   True CycleGAN can model two distributions to learn two conditional generative models

   (i) **[2 points] EBM**: For general energy-based models, the energy $E(x)$ must be positive for every x.

   False energy is $-f_\theta(x)$ negative. Configurations with low energy are more likely

(j) **[2 points] EBM**: There exist energy-based models that cannot be equivalently described as an autoregressive model, regardless of the variable ordering and parameterization of the conditionals.

False. Autoregressive models can be described as energy models because they are products of normalized objects. Their transformation volumes are known analytically. Therefore can evaluate as an EBM.

2. **[10 points total] Comparison of Models**

So far we discussed five major types of generative models: autoregressive models (AR), variational autoencoders (VAE), flow models (Flow), generative adversarial networks (GAN), and energy-based models (EBM). In the following questions, we will compare their strengths and weaknesses. Brief justifications (1 sentence) are sufficient for full points.

(a) **[2 points]** Which of these models can be used to model both discrete and continuous data?

AR, VAE, GAN, EBM

Not flow because continuous space required for invertibility

(b) **[2 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)?

AR, Flow

VAE, GAN and EBM require sampling techniques to approximate likelihood

(c) **[2 points]** Which of these models can be used to model conditional distributions (e.g., trained to map captions to images given a suitable paired dataset)?

GAN, EBM Cycle GAN for example can map zebras → horses

EBMs, specifically diffusion models can generate images from text.

(d) **[2 points]** Which of these models can be used to solve "in-painting" tasks (i.e., fill in missing values in a datapoint such as completing a sentence or an image where some parts have been masked out), assuming access to infinite compute?

EBMs

Autoregressive, EBMs have extensive use in denoising / in-painting tasks e.g DDRM – Denoising Diffusion Restoration Models

PixelCNN is an autoregressive example of pixel in painting

(e) **[2 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?

VAE, flow, GAN with modification, EBM with modification

BiGAN

These models can model latent variable z in their problem formulation

AR cannot because its problem formulation involves modeling chain rule probability factorization without consideration of latent variables.

3. **[18 points total] Variational Autoencoders Basics** In this question, we will consider a Variational Autoencodeer model where as usual $\mathbf{x}$ denotes observed variables and $\mathbf{z}$ denotes latent variables. For each of the following questions briefly explain your answer for full points.

   (a) **[3 points]** Suppose we are training a VAE where the prior $p_\theta(\mathbf{z})$ is parameterized by a Generative Adversarial Network instead of standard Gaussian (i.e., samples from the prior are produced by a trainable generator G_θ), and where the encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$ and decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$ are Gaussian. Can we train this VAE by optimizing the ELBO?

   Yes if GAN learning objective facilitates back propagation when training the VAE optimizing ELBO, however this would be really hard in practice because GAN training itself can be unstable.

   (b) **[3 points]** Suppose we have trained a VAE parameterized by ϕ for encoder and θ for decoder and obtained a global optimum of the ELBO objective (with infinite data from p_{data}, perfect optimization, arbitrary flexible decoder and encoder). Note this implies that the KL divergence between $p_{\text{data}}(\mathbf{x})$ and $p_\theta(\mathbf{x})$ is zero. Suppose we start from a real datapoint $\overline{\mathbf{x}}$ sampled from p_{data}, sample $\mathbf{z}' \sim q_\phi(\mathbf{z} \mid \overline{\mathbf{x}})$ by feeding $\overline{\mathbf{x}}$ to the encoder, then draw a sample $\mathbf{x}' \sim p_\theta(\mathbf{x}|\mathbf{z}')$ by feeding $\mathbf{z}'$ to the decoder. Is $\mathbf{x}'$ distributed according to p_{data}?

   Yes because the implication of this VAE is perfect compression Therefore there's no difference in reconstructing $\overline{\mathbf{x}}$ when it's fed through the VAE

   (c) **[3 points]** Alice and Bob are training two identical VAEs on the same data distribution. They both claim to have obtained a global optimum of the ELBO objective. When they evaluate their models on a test datapoint $\mathbf{x}$, they find their respective encoders map $\mathbf{x}$ to different distributions over the latents. Is this possible if they truly have both obtained a global optimum of the ELBO objective?

   It's not possible for both to claim global optimum of the ELBO objective Training VAEs is inherently stochastic. However this point clearly by requesting multiple training runs of the same model, each time the negative ELBO would be different.

   (d) **[3 points]** 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 $q_\phi(\mathbf{z}|\mathbf{x})$ as the basic inference model representing a Gaussian distribution $\mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi(\mathbf{x})^2))$. Consider an alternative inference model $q_{\phi,A}(\mathbf{z}|\mathbf{x})$ also representing a Gaussian distribution with parameters $\mathcal{N}(\mathbf{z}; \mu_\phi(A\mathbf{x}), \text{diag}(\sigma_\phi(A\mathbf{x})^2))$ where A is a (trainable) $D \times D$ matrix.

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

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

   where

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

   and A is optimized over the set of all $D \times D$ matrices.

   Intuitively matrix A adds more modeling flexibility to training the VAE. matrix A could evaluate as I if it doesn't 'help'. Hence the ELBO with matrix A is at least as tight as the ELBO formulation without it

   (e) **[3 points]** 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 $q_\phi(\mathbf{z}|\mathbf{x})$ as the basic inference model representing a Gaussian distribution $\mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi(\mathbf{x})^2))$. Consider an alternative inference model $q_{\phi,B}(\mathbf{z}|\mathbf{x})$:

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

where B is a trainable matrix. Can you efficiently optimize the objective $\text{ELBO}(\mathbf{x}; p_\theta, q_{\phi.B})$ corresponding to this alternative inference model?

Yes $\tilde{z}$ is parametrized by gaussian μ_ϕ σ_ϕ. Using the reparametrization trick we could compute $z = B(\mu_\phi + \sigma\varepsilon) = B\mu_\phi + B\sigma\varepsilon$ because B is a linear transformation

(f) **[3 points]** Compare $\max_{\theta.\phi.A} \text{ELBO}(\mathbf{x}; p_\theta, q_{\phi.A})$ to $\max_{\theta.\phi.B} \text{ELBO}(\mathbf{x}; p_\theta, q_{\phi.B})$, is one always larger than the other without assumptions on μ_ϕ and σ_ϕ?

It seems they would be the same. They both introduce a linear transformation, one to transform α before training ϕ & μ, and the other to transform z.

$P_x(x;\theta) = P_z\left(f_\theta^{-1}(x)\right)\left|\det \frac{\partial f_\theta^{-1}(x)}{\partial x}\right|$

$z \to x : f_\theta \qquad x \to z \; f^{-1}\theta(x)$

4. **[10 points total] Normalizing Flow Models:**

(a) **[2 points]** Let $Z \sim \text{Uniform}[-2,3]$ (a uniform random variable over the interval $[-2,3]$) and $X = Z^3$. What is $p_X(1)$?.

$f_\theta^{-1}: \; z = x^{\frac{1}{3}}$

$\frac{\partial}{\partial x} = \frac{1}{3}x^{\frac{1}{3}-1}$

$P_x(1) = P_z\left(f_\theta^{-1}(1)\right)\left|\det \frac{1}{3}(1)^{-\frac{2}{3}}\right|$

$= P_z(1) \times \frac{1}{3} = \frac{1}{5} \times \frac{1}{3} = \frac{1}{15}$

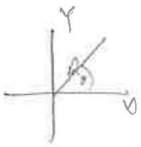
(b) **[2 points]** Let $Z \sim \text{Uniform}[-2,3]$ and $X = Z^2$. Is this a valid normalizing flow model?

No. Because given $X = 1$ there's two solutions to Z.
$Z = \pm 1$. This is not an invertible map.

(c) **[2 points]** Let $Z \sim \text{Uniform}[0,3]$ and $X = Z^2$. Is this a valid normalizing flow model?

Yes because there is a 1:1 mapping between X & Z.

(d) **[4 points] Change of Variable with Polar Coordinates.** Let X and Y be random variables representing Cartesian coordinates in a plane. Let R and Θ be random variables representing the corresponding radius R and angle Θ in a polar coordinate system. Then, Cartesian coordinates X and Y and polar coordinates are related by $X = R\cos(\Theta)$ and $Y = R\sin(\Theta)$, *i.e.*, $(X,Y) = f(R,\Theta) = (R\cos(\Theta), R\sin(\Theta))$. Assume X and Y have joint probability density $p_{X,Y}(x,y)$. Using the change-of-variables formula, derive the joint probability density function $p_{R,\Theta}(r,\theta)$ in terms of $p_{X,Y}$.



Hint: Potentially useful formula:

$$\frac{\partial}{\partial\theta}\cos(\theta) = -\sin\theta, \quad \frac{\partial}{\partial\theta}\sin(\theta) = \cos\theta$$

$P_x(x;\theta) = P_z\left(f_\theta^{-1}(x)\right)\left|\det \frac{\partial f_\theta^{-1}(x)}{\partial x}\right|$ general flow formula.

$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} R\cos\Theta \\ R\sin\Theta \end{bmatrix}$

5. **[11 points total] Variational Recurrent Neural Network:** Here we explore a recurrent version of the VAE for the purpose of modelling sequences. Drawing inspiration from simpler dynamic Bayesian networks (DBNs) such as HMMs and Kalman filters, this variational recurrent neural network (VRNN) augments a traditional RNN model with a sequence of latent variables, similar to a VAE. VRNN additionally models the dependencies between these latent random variables over time with another RNN.

A VRNN model defines a joint distribution $p(x_1, \cdots, x_T, z_1, \cdots, z_T)$ over two groups of random variables $x_1, \cdots, x_T$ and $z_1, \cdots, z_T$. You can think of $x_1, \cdots, x_T$ as the elements of a time series, and $z_1, \cdots, z_T$ as latent variables that are also indexed by time. For brevity, we denote $\mathbf{x}_{\leq t} = (x_1, \cdots, x_t)$, $\mathbf{x}_{<t} = (x_1, \cdots, x_{t-1})$, and we similarly define $\mathbf{z}_{\leq t}$ and $\mathbf{z}_{<t}$. In a VRNN, the joint is factorized as

$$p_\theta(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^{T} p_{\theta_1}(x_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p_{\theta_2}(z_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})$$

where $p_{\theta_1}(x_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t})$ and $p_{\theta_2}(z_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})$ are parameterized as RNNs, and $\theta = (\theta_1, \theta_2)$. As an edge case, $p_{\theta_1}(x_1|\mathbf{z}_{\leq 1}, \mathbf{x}_{<1}) = p_{\theta_1}(x_1|z_1)$ and $p_{\theta_2}(z_1|\mathbf{x}_{<1}, \mathbf{z}_{<1}) = p_{\theta_2}(z_1)$.

(a) **[2 points]** How many parameters are needed to specify the model as a function of T?

(b) **[2 points]** In what order are the variables sampled to generate data from the model?

forward.

(c) **[2 points]** Consider an alternative model

$$p_\psi(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^{T} p_{\psi_1}(x_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p_{\psi_2}(z_t|\mathbf{x}_{<t}, \mathbf{z}_{>t})$$

where $p_{\psi_1}(x_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t})$ and $p_{\psi_2}(z_t|\mathbf{x}_{<t}, \mathbf{z}_{>t})$ are parameterized as RNNs. Is this a valid autoregressive model?

No variable z needs the corresponding backwards direction of variables x to be valid.

(d) **[4 points] Learning:** Because the marginal likelihood $p_\theta(\mathbf{x}_{\leq T})$ is intractable to compute, we need to resort to an ELBO for training. We consider the following encoder:

$$q_\phi(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T}) = \prod_{t=1}^{T} q_\phi(z_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$$

where $q_\phi(z_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ is parameterized using another RNN. Show that the following ELBO for the VRNN described above

$$E_{\mathbf{z}_{\leq T} \sim q_\phi(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\left[\log\left(\frac{p_\theta(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T})}{q_\phi(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\right)\right]$$

start here

$\log p_\theta(x_{\leq T})$

$= \log E_{z_{\leq T} \sim q_\theta(z_{\leq T}|x_{\leq T})}\left[\frac{p_\theta(x_{\leq T}, z_{\leq T})}{q_\phi(z_{\leq T}|x_{\leq T})}\right]$ Jensen's inequality!

$\geq E_{z_{\leq T} \sim q_\theta(z_{\leq T}|x_{\leq T})}\left[\log \frac{p_\theta(x_{\leq T}, z_{\leq T})}{q_\phi(z_{\leq T}|x_{\leq T})}\right]$

$\star\ p_\theta(x_{\leq T}) = \int p_\theta(x_{\leq T}|z_{\leq T})\, p_\theta(z_{\leq T})\, dz$

$= \int p_\theta(x_{\leq T}|z_{\leq T})\, p_\theta(z_{\leq T}) \frac{q_\phi(z_{\leq T}|x_{\leq T})}{q_\phi(z_{\leq T}|x_{\leq T})} dz$

$= \int q_\phi(z_{\leq T}|x_{\leq T}) \frac{p_\theta(x_{\leq T}|z_{\leq T})\, p_\theta(z_{\leq T})}{q_\phi(z_{\leq T}|x_{\leq T})} dz$

$= E_{z_{\leq T} \sim q_\phi(z_{\leq T}|x_{\leq T})}\left[\frac{p_\theta(x_{\leq T}, z_{\leq T})}{q_\phi(z_{\leq T}|x_{\leq T})}\right]$

is equivalent to

$$E_{\mathbf{z}_{\leq T} \sim q_\phi(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})} \left[\sum_{t=1}^{T} (-D_{KL}\left(q_\phi(z_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}) || p_{\theta_2}(z_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})\right) + \log\ p_{\theta_1}(x_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t})) \right]$$

where $D_{KL}(p||q)$ denotes KL divergence.

(e) **[1 points]** Assume the conditionals $q_\phi(z_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ in the encoder RNN are Gaussians (with mean and variance computed by the RNN). Can the ELBO be optimized using the reparameterization trick?

Yes via the reparameterization trick

$z = \mu + \sigma\varepsilon$

6. **[10 points total] GAN: Saturating and Non-Saturating Loss**

Generative Adversarial Networks suffer from the saturation problem, which is when the generator G_θ cannot train as fast as the discriminator D_ϕ and stops learning. To overcome the saturation problem, a non-saturating loss is often used. Non-saturating loss is a subtle modification of the original generator loss, where the generator maximizes the probability of generated images being real rather than minimizing the probability of generated images being fake.

(a) **[2 points] Non-saturating loss:** Recall the original GAN generator loss for a single sample $\mathbf{z}$:

$$\mathcal{L}_S(\theta) = \log\left(1 - D_\phi\left(G_\theta\left(\mathbf{z}\right)\right)\right).$$

$\ln(x)$.

Next, consider the following non-saturating loss:

$$\mathcal{L}_{NS}(\theta) = -\log\left(D_\phi\left(G_\theta\left(\mathbf{z}\right)\right)\right).$$

Show that for fixed $\mathbf{z}$ and D_ϕ:

$$\arg\min_\theta \mathcal{L}_S(\theta) = \arg\min_\theta \mathcal{L}_{NS}(\theta)$$

they are equivalent because finding the min of each loss function requires $D_\phi(G_\theta(z))$ as large as possible.

(b) **[2 points] Gradient of the saturating Loss:** What is the derivative of the saturating loss $\mathcal{L}_S(\theta)$ with respect to $D_\phi(G_\theta(\mathbf{z}))$?

$\nabla_\theta \ln p_\theta(z) = \frac{\nabla_\theta p_\theta(z)}{p_\theta(z)}$

$$\frac{\partial \mathcal{L}_S(\theta)}{\partial D_\phi(G_\theta(z))} = \frac{-1}{1 - D_\phi(G_\theta(z))}$$

(c) **[2 points] Gradient of non-saturating Loss:** What is the derivative of the non-saturating loss $\mathcal{L}_{NS}(\theta)$ with respect to $D_\phi(G_\theta(\mathbf{z}))$?

$$= -\frac{1}{D_\phi(G_\theta(z))}$$

(d) **[4 points] Advantage of non-saturating loss:** Consider the derivatives of the saturating and non-saturating loss with respect to $D_\phi(G_\theta(\mathbf{z}))$ at the beginning of training. Explain why non-saturating loss results in more stable generator training and reduces the saturation problem.
Hint: You might assume the generator produces samples that are easy to distinguish from real data at the beginning of training.

The non saturating loss is more stable. In the beginning the discriminator is very good so $D_\phi(G_\theta(z)) \sim 0.9$

for non saturating loss. The gradient for the generator is

$$\frac{-1}{1-0.9} = \frac{-1}{0.1} = 10$$

for saturating loss this is $-\frac{1}{0.9} = 1.1$

so the generator takes more 'measured steps'

p

7. **[16 points total] Joint Temperature Scaling:**

   In problem set 1, you were asked to implement temperature scaling, a commonly used technique to calibrate the likelihood of the next token in language models. We showed that temperature scaling the logits of individual variables is different from temperature scaling the logits of the joint distribution.

   Let's consider a joint distribution $p(\mathbf{x})$ parameterized by an autoregressive model, where $\mathbf{x} = (x_1, x_2, \ldots x_n)$ is a collection of discrete random variables. We define joint temperature scaling by a scalar $T > 0$ as the following:

$$\log p^T(\mathbf{x}) = \frac{1}{T} \log p(\mathbf{x}) - \log Z(T)$$

   where $p^T(\mathbf{x})$ is our temperature scaled joint distribution of interest and $Z(T)$ is the partition function.

   (a) **[2 points] Partition function:** What is the value of $Z(1)$, the partition function when $T = 1$?

$$\log p'(x) = \frac{1}{1} \log p(x) - \log Z(1) \qquad \log Z(1) = 0$$
$$= \log p(x) - \log Z(1) \qquad \therefore Z(1) = 1$$

   (b) **[2 points] Partition function:** Is the partition function $Z(2)$ for temperature $T = 2$ tractable to compute when n is large?

Yes

   (c) **[2 points] Temperature effect:** Suppose we observe that samples $\mathbf{x}$ that have higher likelihood under $p(\mathbf{x})$ are judged to have higher quality by human evaluators. How should we set T to have p^T generate higher quality samples (compared to p)?

T<1 to make the samples with high likelihood even more likely.

   (d) **[2 points] Sampling from $p^T(\mathbf{x})$:** Can we tractably sample from $p(\mathbf{x})$? Can we tractably sample from $p^T(\mathbf{x})$ (for an arbitrary T)?

Yes because sampling from autoregressive model is a linear time operation with respect to the dimension of the sample.

   (e) **[2 points] Learning from $p^T(\mathbf{x})$:** Now suppose we want to train another autoregressive model $q_\theta(\mathbf{x})$ to learn the distribution of p^T. Is there an unbiased Monte Carlo estimate of the following KL divergence

$$\mathcal{D}_{KL}(p^T(\mathbf{x}) \| q_\theta(\mathbf{x}))$$

   that we can evaluate tractably?

No

   (f) **[4 points] KL Objective:** Let's examine the KL more closely.

Show that finding the q_θ that minimizes $\mathcal{D}_{KL}(p^T(\mathbf{x})\|q_\theta(\mathbf{x}))$ is equivalent to finding the q_θ that minimizes the following:

$$\mathbb{E}_{\mathbf{x}\sim p}\left[-e^{\frac{1-T}{T}\log p(\mathbf{x})}\log q_\theta(\mathbf{x})\right] \quad (\star)$$

(Hint: Importance sampling.)

$p^T(x) \propto e^{\log p(x)/T}$

$D_{KL}(p^T(x) \| q_\theta(x)) = \int p^T(x) \log(p^T(x)/q_\theta(x))\, dx$

$= \mathbb{E}_{x\sim p}[\log(p^T(x)) - \log(q_\theta(x))]$

$= \mathbb{E}_{x\sim p}[\log e^{\log p(x)/T} - \log q_\theta(x)]$

(g) **[2 points]** Is there an unbiased estimator of Equation $(\star)$ that we can evaluate tractably?

Yes.