1	2	3	4	Total

CS 726: Quiz 2

April 1, 2019. 3:40 to 4:50pm

Roll: ____________________

Name: ____________________

This quiz is open notes.

---

1. Consider a 1-dimensional dataset D from a distribution $P_D(x)$ which is a mixture of three Gaussians with the three means at $\mu_1 = 10, \mu_2 = 20$, and $\mu_3 = 30$ each with variance of 1 and equal fraction of examples from each Gaussian. We will see how good GANs and VAEs are in learning to generate samples from such a distribution.

   (a) First consider GANs. Say, as generator $G(z)$ we use a 1-d hidden variable $z \sim \mathcal{N}(0, 1)$ followed by a linear layer $\theta_1 z + \theta_2$ to generate an output x. Assume the discriminator is all powerful and can assign exact Bayes probability over the real distribution (from $D \sim P_D(x)$) and whatever generated distribution x it sees. Provide all values of θ_1, θ_2 for which the GAN objective will be maximized? ..2 $\theta_1 = 1, \theta_2 = 10$ or $\theta_2 = 20$ or $\theta_2 = 30$

   (b) Now, let us say that the generator is actually a mixture of three Gaussians $P_G(x) = \pi_1 \mathcal{N}(x; \mu_1, 1) + \pi_2 \mathcal{N}(x; \mu_2, 1) + \pi_3 \mathcal{N}(x; \mu_3, 1)$ where the generator parameters are $\theta_g = [\pi_1, \pi_2, \pi_3, \mu_1, \mu_2, \mu_3,], \pi_1 + \pi_2 + \pi_3 = 1$. For this the hidden variable z will be a three-way multinomial variable with parameters π_1, π_2, π_3 and conditioned on z we sample a x from $\mathcal{N}(x; \mu_z, 1)$. The θ_g are trained using the GAN objective $\min_{\theta_g} \max_{\theta_d} [E_{x \sim P_D} \log D_{\theta_d}(x) + E_{x \sim P_G} \log(1 - D_{\theta_d}(x))]$. When the generator parameters are: $\pi_1 = 1, \mu_1 = 10$, what is the value of the GAN objective after discriminator is trained? ..3
   The best the discrimnator can do is assign probability of 1 to the 2/3rds instances generator from second and third Gaussian, and 0.5 to the 1/3rd real examples from the first. $1/3 \log(0.25) + \log(0.75)$.

   (c) With the above discriminator parameter fixed, when the generator is retrained what are all configurations of θ_g values at which the generator objective is optimal? ..3
   Any set of π_2, π_3 values with $\pi_1 = 0$ and $\mu_2 = 20, \mu_3 = 30$ will give rise to the minimum generator objective since discriminator will assign probability of 1 to those examples.

2. Now, say we train the same mixture generator using the VAE objective with hidden variable z being a three-way multinomial variable with parameters $\pi_1, \ldots, \pi_3$ which are learned jointly with the decoder parameters μ_z for each $z = 1, 2, 3$.

   (a) How will you design the encoder to get $q_\phi(z|x)$ where ϕ denotes the parameters of the encoder? Guess optimal values of the parameters ϕ of the encoder. ..4 use a softmax layer on top of x. Optimal parameters of the softmax are softmax(10x-50, 20x-200, 30x-450)

   (b) State all possible values of ϕ, μ_z, π_z for which the VAE objective is maximized? ..1
   The ones which align exactly with the true.

(c) Write the formula for the $D_{KL}(q_\phi(z|x)\|p_\pi(z))$ in terms of π and output from the encoder? ..2 $\sum_{i=1}^{3} e_i \log \frac{e_i}{\pi_i}$

3. Consider a neural translation model using the encoder-decoder network discussed in class. In this model, the probability of any output sequence $\mathbf{y}$ is factorized as: $\prod_{t=1}^{n} \Pr(y_t|\mathbf{x}, y_1, \ldots, y_{t-1})$ which is then computed using the decoder RNN as discussed in class. This implies that it is easy to sample sequences in the forward direction where we start from y_1, sample y_2 conditioned on y_1, etc until the end of sequence (EOS) token is sampled. Now, consider a different setting where we know the length n of the output sequence $\mathbf{y}$ in response to an input $\mathbf{x}$. This is equivalent to knowing that y_n =EOS, the end of sequence token. Now, we will use Gibbs sampling to sample tokens $y_1, \ldots, y_{n-1}$. Use $\mathcal{V}$ to denote the vocabulary of y. Without training any extra parameters, state how you will perform this sampling: We obtain an initial sample $\mathbf{y}^0$ by performing foreward sampling of y_i from $\Pr(y|y_1^0, \ldots, y_{i-1}^0)$ over vocabulary $\mathcal{V}-$EOS, and then just setting y_n = EOS.

   (a) Justify briefly with an example why $\mathbf{y}^0$ is not a valid sample from: $\Pr(\mathbf{y}|\mathbf{x}, y_n = EOS)$? ..2 Since this sampling was not conditioned on $y_n = EOS$, we can get highly unlikely incomplete sentences like "I went to ¡EOS¿"

   (b) How will you compute $\Pr(y_i|\mathbf{y}_{-i}^0)$ from the decoder RNN where $\mathbf{y}_{-i}^0$ denotes the sequence without the ith token y_i^0? ..4 For each $y \in \mathcal{V}$ we have to use the RNN starting from step i to completion to compute $s(y) = \Pr(y|y_1^0, \ldots, y_{i-1}^0) \prod_{t=i+1}^{n} \Pr(y_t^0|y_1^0, \ldots, y_{i-1}^0, y, y_{i+1}^0 \ldots, y_{t-1}^0)$. Then we get $\Pr(y_i|\mathbf{y}_{-i}^0) = \frac{s(y_i)}{\sum_{y \in \mathcal{V}} s(y)}$. We have to exclude EOS in this calculation.

   (c) What is the running time of the above computation? ..1 ..

   (d) Now assume that you are allowed to train a different type of decoder that allows efficient Gibbs sampling. Assume your training data is denoted as $\{(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^N, \mathbf{y}^N)\}$. Describe briefly the design and training of such a network. ..3
   BERT type of bidirectional model where we mask out randomly arbitrary tokens from each true $\mathbf{y}^i$ and ask to generate that conditioned on the rest.

4. Consider the attention-based encoder-decoder network for sequence prediction that at each time step t outputs a probability distribution about possible output tokens as:

$$\Pr(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{n} \Pr(y_t|\mathbf{s}_t, \sum_{a_t} P_t(a_t)\mathbf{x}_{a_t}) \tag{1}$$

where the distribution of each attention variable a_t is computed as a function of the decoder state $\mathbf{s}_t$ and encoder state $\mathbf{x}_a$ as: $P_t(a) = \text{softmax}(A_\theta(\mathbf{x}_a, \mathbf{s}_t))$ where a takes values between 1 and m, the number of input encoder states. This is called the soft attention model. Now we will consider an alternative model called the joint-attention model as follows:

$$\Pr(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{n} \sum_{a_t=1}^{m} P_t(a_t) \Pr(y_t|\mathbf{s}_t, \mathbf{x}_{a_t}) \tag{2}$$

where m is the number of tokens in the input $\mathbf{x}$.

   (a) With application like neural machine translation (NMT) in mind, state one reason why you expect the joint model to be better than the soft-attention model? ..2
   We directly couple the input to the output via a focussed attention on a specific input token. Hence we expect accuracy to be higher.

(b) In similar NMT settings, state a major limitation of the joint model compared to the soft attention model? ..3 The softmax operation to compute $\Pr(y|..)$ is over a large vocabulary and time-consuming. We are now performing m times more softmax operation.

Total: 30