GNR 638 QUIZ 2 APR 04, 2024

Answer all questions.
Each question carries 3 marks

1. **[1.5 + 1.5 marks]**For convex functions, Jensen's inequality works the other way around.

$$g(\mathbb{E}[y]) \leq \mathbb{E}(g[y])$$

A function is convex if its second derivative is greater than or equal to zero everywhere. Show that the function $g(x) = x^{2n}$ is convex for arbitrary $n \in [1, 2, 3, ...]$. Use this result with Jensen's inequality to show that the square of the mean $\mathbb{E}[x]$ of a distribution $Pr(x)$ must be less than or equal to its second moment $\mathbb{E}[x^2]$.

Answer. $g(x) = x^{2n}$ is the given function.

$$g'(x) = 2n \cdot x^{2n-1}$$

$$g''(x) = 2n(2n-1) \cdot x^{2n-2}$$

To comment on the convexity of $g(x)$, we need to have

$$g''(x) \geq 0 \quad \forall x$$

Now,

$$g''(x) = 2n(2n-1) \cdot x^{2n-2}$$

For any $n \in \{1, 2, 3, \ldots\}$, $x^{2n-2} \geq 0 \quad \forall x$
And $2n(2n-1) > 0$

$$\therefore g''(x) \geq 0$$

Hence the function $g(x)$ is convex.

Jensen's inequality for a convex function is:

$$g\left(\mathbb{E}[x]\right) \leq \mathbb{E}[g(x)] \tag{1}$$

Let us apply Jensen's inequality to $g(x) = x^2$:

$$g(x) = x^2 \Rightarrow g''(x) = 2 \geq 0 \quad \text{(hence, convex)}$$

Substituting $g(x) = x^2$ in (1), we get:

$$\left(\mathbb{E}[x]\right)^2 \leq \mathbb{E}[x^2]$$

Hence, proved.

2. **[1.5 + 1 + 0.5 marks]** If $p_r(x)$ and $p_g(x)$ represent the probability distributions of real and generated samples, respectively, in GAN, what is the optimal value of the discriminator $D(x)$ (let us denote it as $D^*(x)$), in terms $p_r(x)$ and $p_g(x)$, that maximizes the value function $L(G, D)$ given by:

$$L(G, D) = \int_x (p_r(x) \log(D(x)) + p_g(x) \log(1 - D(x))) \, dx$$

Substitute the derived $D^*(x)$ back into $L(G, D)$ and express the result in terms of Jensen-Shannon divergence between $p_r(x)$ and $p_g(x)$. Also explain why JS Divergence may not be a good choice for GAN training.

Answer. 1. Given loss function is

$$L(G, D) = \int_x (p_r(x) \log(D(x)) + p_g(x) \log(1 - D(x))) \, dx$$

Since we are interested in the best value of $D(x)$ to maximize $\mathcal{L}(G, D)$, let us label $\bar{x} = D(x)$, $A = p_r(x)$, and $B = p_g(x)$.

And then what is inside the integral (we can safely ignore the integral because $\bar{x}$ is sampled over all the possible values) is:

$$f(\bar{x}) = A \log \bar{x} + B \log(1 - \bar{x})$$

$$\frac{df(\bar{x})}{d\bar{x}} = A \cdot \frac{1}{\ln 10} \cdot \frac{1}{\bar{x}} - B \cdot \frac{1}{\ln 10} \cdot \frac{1}{1 - \bar{x}}$$

$$= \frac{1}{\ln 10} \left(\frac{A}{\bar{x}} - \frac{B}{1 - \bar{x}} \right)$$

$$= \frac{1}{\ln 10} \cdot \frac{A(1 - \bar{x}) - B\bar{x}}{\bar{x}(1 - \bar{x})}$$

$$= \frac{1}{\ln 10} \cdot \frac{A - (A + B)\bar{x}}{\bar{x}(1 - \bar{x})}$$

Thus, set $\frac{df(\bar{x})}{d\bar{x}} = 0$, we get the best value of the discriminator:

$$D^*(x) = \bar{x}^* = \frac{A}{A + B} = \frac{p_r(x)}{p_r(x) + p_g(x)} \in [0, 1]$$

2. On substituting, we get

$$\begin{aligned} Ł(G, D) &= \int \left(p_r(x) \log \left(\frac{p_r(x)}{p_r(x) + p_g(x)} \right) + p_g(x) \log \left(1 - \frac{p_r(x)}{p_r(x) + p_g(x)} \right) \right) dx \\ &= \int p_r(x) \log \left(\frac{p_r(x)}{p_r(x) + p_g(x)} \right) dx + \int p_g(x) \log \left(\frac{p_g(x)}{p_r(x) + p_g(x)} \right) dx \end{aligned} \tag{1}$$

Cont.

Multiply by $\frac{1}{2}$ both sides:

$$Ł(G, D) = \frac{1}{2}\left[\int p_r(x) \log\left(\frac{p_r(x)}{p_r(x) + p_g(x)}\right) dx + \int p_g(x) \log\left(\frac{p_g(x)}{p_r(x) + p_g(x)}\right) dx\right] \quad (1)$$

We know KL divergence between A and B is given by:

$$D_{\mathrm{KL}}(A \parallel B) = \int A \log\left(\frac{A}{B}\right)$$

Let $M = p_r(x) + p_g(x)$, $P = p_r(x)$, $Q = p_g(x)$.
Therefore, Equation (1) can be written as:

$$Ł(G, D)) = \frac{1}{2}\left[D_{\mathrm{KL}}(P \parallel M) + D_{\mathrm{KL}}(Q \parallel M)\right]$$

Also, the **Jensen-Shannon Divergence** is given by:

$$D_{\mathrm{JS}}(P \parallel Q) = \frac{1}{2} D_{\mathrm{KL}}(P \parallel M) + \frac{1}{2} D_{\mathrm{KL}}(Q \parallel M)$$

Hence, Ł(G, D) is the **Jensen-Shannon divergence** between $p_r(x)$ and $p_g(x)$.

3. The Jensen-Shannon (JS) divergence can be problematic in GAN training because when the real and generated data distributions have little or no overlap (which is common at the start of training), the JS divergence becomes constant typically $\log 2$. This means the gradient of the loss becomes zero, giving no useful feedback to the generator. As a result, the generator can't learn how to improve, leading to training instability or stagnation. This is one of the main motivations behind alternatives like the Wasserstein distance used in WGAN

3. **[1.5 + 1.5 marks]** An autoencoder is trained to reconstruct inputs $x \in \mathbb{R}^n$. Based on the distribution of the input values, answer the following:

   (a) If the input vector $x \in \{0, 1\}^n$ (i.e., binary-valued), what are the **ideal** choices for the encoder activation, decoder activation, and loss function? Justify your choices.

   (b) If the input vector $x \in \mathbb{R}^n$ (i.e., real-valued), what are the **ideal** choices for the encoder activation, decoder activation, and loss function? Justify your choices.

1. Binary Input Case: $x \in \{0, 1\}^n$

- **Encoder Activation:** ReLU / sigmoid / tanh (any standard non-linearity)
- **Decoder Activation: Sigmoid**
- **Loss Function: Binary Cross Entropy (BCE)**

Justification:

- Since each x_i is binary, we model it as a Bernoulli variable: $x_i \sim \text{Bernoulli}(\hat{x}_i)$
- Sigmoid activation ensures that $\hat{x}_i \in [0, 1]$, interpretable as probability
- The BCE loss is derived from the negative log-likelihood of the Bernoulli distribution:

$$\mathcal{L}_{\mathrm{BCE}} = -\sum_{i=1}^{n} [x_i \log(\hat{x}_i) + (1 - x_i) \log(1 - \hat{x}_i)]$$

2. Real-Valued Input Case: $x \in \mathbb{R}^n$

- **Encoder Activation:** ReLU / sigmoid / tanh (any standard non-linearity)
- **Decoder Activation: Linear (Identity)**
- **Loss Function: Mean Squared Error (MSE)**

Justification:

- Assume each $x_i \sim \mathcal{N}(\hat{x}_i, \sigma^2)$, i.e., Gaussian noise model
- A linear decoder allows outputs across the entire real line
- The negative log-likelihood under Gaussian assumption simplifies to:

$$\mathcal{L}_{\text{MSE}} = \sum_{i=1}^{n} (x_i - \hat{x}_i)^2$$

4. **[2 + 1 marks]** A Variational Autoencoder (VAE) is trained to model a latent variable distribution $z \in \mathbb{R}^k$ given input data $x \in \mathbb{R}^n$. Assume the prior distribution over the latent variables is a standard normal distribution $p(z) = \mathcal{N}(z; 0, I)$, and the approximate posterior distribution $q_\phi(z|x)$ is also modeled as a normal distribution with parameters predicted by the encoder. Answer the following:

   (a) Derive the mathematical form of the Evidence Lower Bound (ELBO) for the VAE, clearly defining all terms and explaining how it balances reconstruction accuracy and regularization. Include the KL-divergence term and the expected log-likelihood term in your derivation.

   (b) If the input data x follows a Gaussian distribution $p(x) = \mathcal{N}(x; \mu, \sigma^2 I)$, what is the ideal choice for the likelihood function $p_\theta(x|z)$ used in the decoder? Justify your choice mathematically, considering the form of the input distribution and the goal of maximizing the ELBO.

Answer: 1. Derive the mathematical form of the ELBO for the VAE.

The ELBO is derived as:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)),$$

where $p(z) = \mathcal{N}(z; 0, I)$, $q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma^2_\phi(x) I)$, and $p_\theta(x|z)$ is the decoder likelihood.
- First term: $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ measures reconstruction accuracy. - Second term: $D_{KL}(q_\phi(z|x)||p(z))$ regularizes by penalizing deviation from the prior. For Gaussians:

$$D_{KL} = \frac{1}{2} \sum_{j=1}^{k} (1 + \log(\sigma^2_{\phi,j}(x)) - \mu^2_{\phi,j}(x) - \sigma^2_{\phi,j}(x)).$$

The ELBO balances reconstruction and regularization by maximizing the former while minimizing the latter.

2. Ideal likelihood $p_\theta(x|z)$ for Gaussian input $p(x) = \mathcal{N}(x; \mu, \sigma^2 I)$.

Choose:

$$p_\theta(x|z) = \mathcal{N}(x; \mu_\theta(z), \sigma^2_\theta(z) I).$$

Justification: - Matches input distribution for consistency. - ELBO term simplifies to:

$$\log p_\theta(x|z) = -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2_\theta(z)) - \frac{1}{2\sigma^2_\theta(z)} \|x - \mu_\theta(z)\|^2,$$

enabling efficient optimization. - Gaussian assumption aligns with data noise model, ensuring stable training and coherent generation.

Cont.

5. **[1+2 marks]** Consider a linear autoencoder with an input vector $x \in \mathbb{R}^n$. The encoder maps the input to a latent representation $z \in \mathbb{R}^k$ using a weight matrix $W_e \in \mathbb{R}^{k \times n}$, and the decoder reconstructs the input as $\hat{x} \in \mathbb{R}^n$ using a weight matrix $W_d \in \mathbb{R}^{n \times k}$. The reconstruction is given by $\hat{x} = W_d W_e x$.

   (a) Derive the expression for the reconstruction error $\mathcal{L} = \|x - \hat{x}\|^2$ in terms of W_e, W_d, and x.

   (b) Assuming that the data covariance matrix is $\Sigma = \mathbb{E}[xx^T]$, show that minimizing the expected reconstruction error $\mathbb{E}[\mathcal{L}]$ with respect to W_e and W_d leads to W_e spanning the subspace of the top k eigenvectors of Σ, analogous to Principal Component Analysis (PCA).

Answer. (a) Reconstruction Error Expression:

The reconstruction error is defined as:

$$\mathcal{L} = \|x - \hat{x}\|^2 = \|x - W_d W_e x\|^2$$

Expanding this:

$$\mathcal{L} = (x - W_d W_e x)^T (x - W_d W_e x)$$

$$= x^T x - x^T W_d W_e x - x^T W_e^T W_d^T x + x^T W_d W_e W_e^T W_d^T x$$

$$= x^T x - 2x^T W_d W_e x + x^T W_d W_e W_e^T W_d^T x$$

(b) Connection to PCA:

To minimize the expected reconstruction error:

$$\mathbb{E}[\mathcal{L}] = \mathbb{E}[\|x - W_d W_e x\|^2]$$

Assuming $\Sigma = \mathbb{E}[xx^T]$ is the data covariance matrix, we get:

$$\mathbb{E}[\mathcal{L}] = \mathrm{Tr}(\Sigma) - 2\mathrm{Tr}(W_d W_e \Sigma) + \mathrm{Tr}(W_d W_e W_e^T W_d^T \Sigma)$$

To find the minimum, we differentiate with respect to W_d and set the gradient to zero:

$$-2W_e \Sigma + 2W_e W_e^T W_d^T \Sigma = 0$$

Assuming $W_d = W_e^T$, we substitute and get:

$$W_e \Sigma = W_e W_e^T W_e \Sigma$$

This implies that the rows of W_e must lie in an invariant subspace of Σ, i.e., they must align with its eigenvectors. By choosing the top k eigenvectors, the autoencoder captures the directions of maximum variance — just like Principal Component Analysis (PCA).

The End.