Sharif University of Technology

AE and VAE

Deep Learning

Adel Movahedian

400102074

Question 1: Autoencoders (AE) vs. Variational Autoencoders (VAE)

(a)Suppose we want to generate a dataset similar to a given dataset using a standard AE. We train an AE, select a random point in the latent space (with a uniform distribution), and pass it through the decoder to generate data. What is your expectation of the output from the decoder, Is the output likely to resemble the dataset? Why?

Answer

Autoencoders (AEs) are designed to compress input data into a latent space and then reconstruct it. However, the latent space of a standard AE does not have an explicit structure. This lack of structure leads to the following issues when attempting to generate data:

- **Unstructured Latent Space:** In a standard AE, the latent space is not constrained to follow any specific distribution (for example Gaussian). This means that randomly sampled points (for example from a uniform distribution) may not correspond to meaningful latent representations. Consequently, the decoder cannot generate realistic outputs resembling the dataset.
- **Sparse Data Representation:** The latent space of AEs is often sparse, with meaningful latent points densely packed in specific regions. Randomly selecting points in this space is unlikely to yield valid reconstructions.

So selecting random points from the data space itself has no inherent structure either and would not yield realistic data. While AEs can effectively reconstruct points already present in the dataset, they fail to generalize beyond this. Therefore, neither method (sampling from latent space or data space) is likely to produce realistic samples.

(b) Describe at least three methods using VAEs to produce data similar to the original dataset. Explain how VAEs address the limitations of standard AEs in this regard. (5 points)

Answer

Variational Autoencoders (VAEs) overcome the limitations of standard AEs by enforcing a structured and continuous latent space. Below are three effective methods for generating data with VAEs:

1. **Regularized Latent Space:** VAEs impose a prior distribution (usually Gaussian) on the latent space. This ensures that the latent space is continuous and well-organized, enabling smooth transitions between points.
2. **Sampling from the Prior:** Points are sampled directly from the prior distribution (e.g., $\mathcal{N}(0, 1)$) and passed through the decoder to generate new, realistic data. Since the latent space is regularized, these samples correspond to meaningful data.
3. **Smooth Interpolations:** In a VAE, interpolating between two points in the latent space produces smooth transitions in the generated data. This is useful for applications requiring gradual variations or morphing between features.

By introducing probabilistic encoding and a regularized latent space, VAEs address the fundamental issue of unstructured latent spaces in AEs and are capable of generating realistic data resembling the dataset.

(c) Suppose, during the training process of an AE, we add Gaussian noise ($\mathcal{N}(0, 0.05 \times R^2)$) to the output of the encoder, where R is the root mean square of the distance of the latent points from the center of the latent space. Will this make the trained AE better at generating data that resembles the dataset? Specifically, will the output of randomly sampled points in the latent space resemble the dataset more compared to a standard AE? Why? (5 points)

Answer

Adding Gaussian noise to the output of the encoder during AE training introduces stochasticity into the learning process. This modification makes the AE slightly more robust to noise, but it does not fundamentally address the issue of an unstructured latent space. The reasons are:

- **Lack of Latent Space Regularization:** Adding Gaussian noise does not impose any specific structure (e.g., Gaussian distribution) on the latent space. The latent space remains sparse and disorganized.
- **Improved Robustness but Not Generation:** While the introduction of noise during training might make the model better at reconstructing slightly perturbed inputs, it does not make the AE inherently better at generating realistic data from random latent points.

In conclusion, while this method can make the AE more robust, it does not significantly improve its ability to generate data resembling the dataset when sampling from the latent space.

(d) Compare the advantage of VAEs over the method mentioned in part (c). What are the key differences between these two approaches? (5 points)

Answer

The advantage of VAEs over the method mentioned in part (c) lies in their structured latent space. Key differences include:

- **Latent Space Regularization:**
  - **AE with Gaussian Noise:** The latent space remains unstructured despite adding noise. Points sampled randomly may not correspond to meaningful outputs.
  - **VAE:** A Gaussian prior is enforced, ensuring a smooth and continuous latent space where random samples correspond to realistic outputs.
- **Stochastic vs. Deterministic Encoding:**
  - **AE with Gaussian Noise:** The encoding process remains deterministic, even though noise is added.
  - **VAE:** Encoding is inherently stochastic, with each data point encoded as a distribution rather than a single vector. This enables better generalization and generation.
- **Loss Function:**
  - **AE with Gaussian Noise:** Optimizes only for reconstruction loss.
  - **VAE:** Optimizes for both reconstruction loss and KL divergence, balancing data fidelity and latent space regularity.

Overall, VAEs provide a more principled and effective approach for generating realistic data, addressing the fundamental limitations of AEs.

Question 2: Likelihood Estimation in VAEs

(a) Suppose our dataset is $D = \{x_1, x_2, \ldots, x_n\}$, and we aim to estimate the maximum likelihood. Study the relation below and explain why the parameters of the distribution must maximize $\sum_{i=1}^{n} \log(p_\theta(x_i))$, where θ are the parameters of the model. (5 points)

Answer

To estimate the parameters θ, we maximize the log-likelihood function $\sum_{i=1}^{n} \log(p_\theta(x_i))$, which is equivalent to maximizing the likelihood of the data under the model distribution. The reasons are:

- **Likelihood Interpretation:** $p_\theta(x_i)$ represents the probability assigned by the model to each data point. Maximizing this ensures the model assigns high probabilities to observed data.
- **Logarithm Benefits:** Using the logarithm simplifies optimization by converting products into sums and mitigating the effects of small probability values.

In summary, maximizing $\sum_{i=1}^{n} \log(p_\theta(x_i))$ ensures that the model accurately captures the underlying data distribution.

(b) Demonstrate the equivalence of minimizing cross-entropy loss and maximizing likelihood estimation. (5 points)

Answer

Cross-entropy loss is defined as $-\sum_{i=1}^{n} y_i \log(\hat{y}_i)$, where y_i are the true labels (or probabilities), and $\hat{y}_i$ are the predicted probabilities. When $y_i = 1$ for observed data points, minimizing cross-entropy reduces to maximizing the log-likelihood function $\sum_{i=1}^{n} \log(p_\theta(x_i))$. Thus, minimizing cross-entropy is equivalent to maximizing likelihood estimation.

(c)(i) In VAEs, the ultimate goal is to have a generative model where the output distribution resembles the dataset distribution. Explain why ELBO is used instead of directly maximizing $\sum_{i=1}^{n} \log(p_\theta(x_i))$. Justify why the following holds:

$$\log p_\theta(x_i) - D_{KL}[q_\phi(z|x_i) \| p_\theta(z|x_i)] = \mathbb{E}_z[\log p_\theta(x_i|z)] - D_{KL}[q_\phi(z|x_i) \| p_\theta(z)].$$

(5 points)

Answer

To prove why minimizing the ELBO aligns with maximizing the log-likelihood of the data, we start with the marginal log-likelihood decomposition. Directly maximizing $\log p_\theta(x_i)$ is intractable due to the integration over latent variables z:

$$\log p_\theta(x_i) = \log \int p_\theta(x_i, z)\, dz.$$

Computing this integral exactly is computationally expensive. To address this, we introduce a variational approximation $q_\phi(z|x_i)$ and decompose $\log p_\theta(x_i)$ as follows:

$$\begin{aligned}\log p_\theta(x_i) &= \mathbb{E}_{q_\phi(z|x_i)}\left[\log \frac{p_\theta(x_i, z)}{q_\phi(z|x_i)}\right] \\ &= \mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)] - \mathbb{E}_{q_\phi(z|x_i)}\left[\log \frac{q_\phi(z|x_i)}{p_\theta(z)}\right].\end{aligned}$$

The second term corresponds to the KL divergence:

$$D_{KL}[q_\phi(z|x_i)\|p_\theta(z)],$$

which measures how close the approximate posterior $q_\phi(z|x_i)$ is to the prior $p_\theta(z)$. Adding and subtracting $D_{KL}[q_\phi(z|x_i)\|p_\theta(z|x_i)]$, we arrive at the ELBO:

$$\text{ELBO} = \mathbb{E}_{z\sim q_\phi}[\log p_\theta(x_i|z)] - D_{KL}[q_\phi(z|x_i)\|p_\theta(z)].$$

This equation balances reconstruction quality and latent space regularity. By maximizing the ELBO, we effectively maximize a lower bound on $\log p_\theta(x_i)$, aligning with the goal of maximizing the log-likelihood.

(c)(ii) Justify why many VAE implementations minimize $\mathbb{E}_z[\log p_\theta(x_i|z)]$ using cross-entropy loss rather than another metric. (15 points)

Answer

In many VAE implementations, the term $\mathbb{E}_z[\log p_\theta(x_i|z)]$ is replaced by the cross-entropy loss. This is justified for several reasons:

- **Natural Fit for Probabilistic Models:** Cross-entropy loss aligns with the probabilistic nature of VAEs, as it directly measures the divergence between the predicted and true distributions. For binary or normalized image data, it is equivalent to the negative log-likelihood under a Bernoulli assumption.
- **Gradient-Based Optimization:** Cross-entropy provides smooth gradients, making it suitable for stochastic gradient descent and other optimization techniques.
- **Practicality:** Cross-entropy is widely implemented and computationally efficient, reducing numerical instability compared to explicitly calculating $\mathbb{E}_z[\log p_\theta(x_i|z)]$ for each latent sample.

This substitution simplifies computation, especially when pixel values are scaled to $[0, 1]$, aligning well with probabilistic interpretations of pixel intensities.

Regularization

The KL divergence term in the ELBO regularizes the latent space by encouraging the approximate posterior $q_\phi(z|x_i)$ to stay close to the prior $p_\theta(z)$. This ensures that the latent space remains structured and improves the generalizability of the model.

Question 3: Gaussian in VAEs

(3) Why is the latent space in VAE typically assumed to follow a Gaussian distribution (apart from simplifying computations)? (20 points) Research and indicate whether distributions other than Gaussian are used in practice

Answer

Variational Autoencoders (VAEs) commonly assume a Gaussian distribution for the latent space due to several theoretical, practical, and computational reasons. Below are detailed explanations for why this assumption is used and whether other distributions can be considered:

1. Mathematical Simplicity The Gaussian distribution is mathematically tractable and allows for efficient computation of the Evidence Lower Bound (ELBO). Key terms like the Kullback-Leibler (KL) divergence between the approximate posterior $q_\phi(z|x)$ and the prior $p_\theta(z)$ have closed-form solutions for Gaussian distributions. For example:

$$D_{KL}[q_\phi(z|x)\|p_\theta(z)] = \frac{1}{2}\left(\mathrm{Tr}(\Sigma) + \mu^\top\mu - k - \log|\Sigma|\right),$$

where μ and Σ are the mean and covariance of the posterior distribution, and k is the dimensionality of the latent space.

2. Universality of the Gaussian Distribution By the Central Limit Theorem, the Gaussian distribution emerges naturally as the sum of independent random variables. This makes it a general and flexible choice for modeling a wide range of data distributions. The isotropic Gaussian prior $\mathcal{N}(0, I)$ provides a simple and interpretable latent space structure, enabling meaningful interpolations and smooth transitions between data points.

3. Smoothness and Continuity in the Latent Space The Gaussian prior ensures that the latent space is continuous and dense, where small changes in the latent vector correspond to smooth variations in the generated outputs. This is essential for tasks like interpolation and generative modeling.

4. Scalability and Computational Efficiency Gaussian distributions allow VAEs to scale well with higher-dimensional data since sampling and reconstruction remain computationally efficient. Reparameterization trick, a key component of VAEs, relies on Gaussian distributions for efficient gradient-based optimization:

$$z = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I),$$

where μ and σ are the output of the encoder network, and ϵ is noise sampled from a standard Gaussian.

5. Flexibility to Extend to Non-Gaussian Priors While Gaussian priors are common, VAEs can incorporate other distributions depending on the task:

- **Beta Distribution:** Useful for modeling latent variables constrained within $[0, 1]$, such as probabilities.
- **Dirichlet Distribution:** Suitable for applications like topic modeling where the latent variables represent proportions.
- **Mixture of Gaussians:** Provides greater flexibility for modeling multimodal latent spaces, capturing more complex data distributions.

Answer

Practical Applications of Non-Gaussian Priors Researchers have experimented with alternative priors to match the nature of specific datasets. For instance:

- **VQ-VAE (Vector Quantized VAE):** Discrete latent spaces are used, enabling applications like image compression and clustering.
- **Flow-based Models:** Introduce transformations that map latent variables from Gaussian priors to more complex distributions.

These methods often require more computational resources and careful design of the prior distributions.

Conclusion The Gaussian distribution is not a mandatory choice but rather a practical and efficient starting point. Its mathematical properties make it ideal for general-purpose VAEs. However, for specific applications, exploring alternative distributions like Beta, Dirichlet, or multimodal priors can lead to better performance tailored to the task.

(4) In brief, what is the idea of -VAE, and what is its difference with VAEs? What is the importance of the disentanglement metric, and how is it used?

Answer

The Idea of -VAE and Its Difference from VAEs The -VAE is an extension of the Variational Autoencoder (VAE) that introduces a hyperparameter β to balance reconstruction accuracy and the disentanglement of latent representations. The key idea behind -VAE is to encourage the model to learn independent and interpretable latent factors by increasing the weight of the KL divergence term in the Evidence Lower Bound (ELBO). The modified objective function is given by:

$$\mathcal{L}_{\beta-\text{VAE}} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}[q_\phi(z|x)\|p_\theta(z)].$$

When $\beta > 1$, the model prioritizes disentanglement, often at the cost of reconstruction fidelity. This trade-off makes -VAE particularly effective at discovering disentangled latent factors compared to standard VAEs, which equally weigh reconstruction and regularization terms.

Importance of Disentanglement Metrics Disentanglement metrics are crucial for evaluating how well a model separates and represents the underlying factors of variation in the data. These metrics provide quantitative insights into the quality of the latent representations and serve multiple purposes:

- **Model Comparison:** Metrics enable standardized comparisons of disentanglement performance across different models and configurations.
- **Guiding Model Development:** By analyzing metric results, researchers can refine models to achieve better disentanglement.
- **Ensuring Interpretability:** Disentanglement metrics help verify that the latent space aligns with the data's underlying factors, making representations more interpretable.

Answer

Common Disentanglement Metrics The paper highlights the following key metrics:

- **Mutual Information Gap (MIG):** This measures the difference in mutual information between the top two latent dimensions for each factor of variation. A higher MIG indicates better disentanglement.
- **Beta-VAE Metric:** Introduced alongside -VAE, this metric trains a classifier to predict fixed factor indices and evaluates how well the latent dimensions capture distinct factors.
- **Disentanglement, Completeness, and Informativeness (DCI):** This evaluates disentanglement by analyzing how each latent dimension predicts factors of variation, along with completeness and informativeness metrics.

Usage of Disentanglement Metrics These metrics are employed to assess the disentanglement properties of models. For example:

- They guide hyperparameter tuning, such as selecting an optimal β in -VAE.
- Metrics like MIG and DCI reveal whether individual latent dimensions correspond to distinct data factors, ensuring meaningful and interpretable representations.
- The Beta-VAE metric is often used to validate disentanglement in benchmark datasets.

final conclusion : The -VAE improves upon standard VAEs by promoting disentanglement through the β parameter. Disentanglement metrics are essential for evaluating these improvements, enabling model comparison, guiding development, and ensuring interpretability. The choice of metric depends on the application and specific goals of the research.

Answer of Practical Questions

(a) Explanation of different parts of the loss

Answer

The cost function in VQ-VAE includes multiple terms, as shown in Equation (2):

$$L = \log(p(x|z_q(x))) + \|z_e(x).\text{detach}() - e\|^2 + \beta\|z_e(x) - e.\text{detach}()\|^2.$$

Each term represents a specific component of the total cost function:

- **Reconstruction Loss ($\log(p(x|z_q(x)))$):** This term measures the quality of reconstruction from the latent representation. It ensures that the decoder output closely matches the input data.
- **Codebook Loss ($\|z_e(x)$.detach() $- e\|^2$):** This term encourages the codebook vectors to move closer to the encoder outputs. It updates the discrete embeddings to better capture the patterns in the data.
- **Commitment Loss ($\beta\|z_e(x) - e$.detach()$\|^2$):** This term ensures that the encoder commits to a specific codebook vector by penalizing large differences between the encoder output and the selected codebook vector.

The combination of these terms balances reconstruction quality, codebook usage, and latent space regularity.

(b) Explanation of the Codebook in VQ-VAE

Answer

The `detach()` function is an important element in the implementation of VQ-VAE, particularly in the learning process of the discrete latent representations. Its purpose is to prevent gradients from propagating back through specific parts of the model during backpropagation. This ensures that the discrete codebook entries remain constant and stable while the rest of the model is updated.
In the paper, it is explained that `detach()` is used right before updating the codebook embeddings. The process can be mathematically expressed as:

$$B = A + (f(A) - A).\texttt{detach()}.$$

Here:

- A refers to the encoder output before quantization.
- $f(A)$ is the quantized version of A.
- The operation $(f(A) - A).\texttt{detach()}$ ensures that the gradients do not affect the quantization process during backpropagation.

By using `detach()`, the quantized output $f(A)$ is treated as a fixed constant during gradient computation. This separation is crucial because quantization involves a non-differentiable operation (choosing the nearest vector), which makes gradient-based optimization challenging.

Clarification of Purpose

- **Stabilizing Training:** The `detach()` function ensures that during backpropagation, gradients are not propagated back through the quantization step. This stabilizes the training process by preventing the discrete embeddings from being directly influenced by gradient updates.
- **Encouraging Independence:** The encoder focuses on minimizing the reconstruction loss while the codebook embeddings evolve separately, maintaining a clear separation of responsibilities.

Codebook in VQ-VAE

The codebook in VQ-VAE is a collection of discrete embedding vectors that represent the latent space. These vectors are updated during training to better capture the patterns in the input data. The process of updating the codebook involves:

- **Choosing the Closest Vector:** For each encoder output $z_e(x)$, the closest vector e_i from the codebook is selected based on the Euclidean distance.
- **Updating the Codebook:** The selected vector e_i is updated to move closer to the encoder output using the codebook loss term $\mathcal{L}_{\text{vq}}$.
- **Balancing Commitments:** The encoder is encouraged to commit to a specific codebook vector by minimizing the commitment loss $\mathcal{L}_{\text{commit}}$.

Answer

Additional Details

- **Codebook Vector Updates:** The codebook vectors are updated using an **exponential moving average (EMA)** or a gradient-based approach. This ensures that the vectors represent clusters of latent representations effectively.
- **Loss Components:** The codebook loss ($\mathcal{L}_{\text{vq}}$) encourages the selected codebook vector to align with the encoder output. The commitment loss ($\mathcal{L}_{\text{commit}}$) penalizes large differences between the encoder output and the selected codebook vector.
- **Quantization Trade-Off:** Quantization introduces a trade-off between reconstruction quality and discrete representation fidelity. The balance is controlled by hyperparameters like β in the loss function.

The efficient use of the codebook allows VQ-VAE to learn meaningful discrete representations that preserve the most relevant features of the data. Moreover, the stability ensured by `detach()` further enhances the codebook's usability in the model.