Deep learning

DR.Fatemi Zadeh

electrical engineering department

Ahmadreza Majlesara 400101861

computer assignment repository

assignment 5

January 21, 2025

Deep learning

assignment 5

Ahmadreza Majlesara 400101861
computer assignment repository

Question 1

Part a

Suppose we want to generate data similar to our dataset using a *standard autoencoder (AE)*. We have trained an AE, then we pick a random point (using a uniform distribution) in the latent space and feed it into the trained decoder. In your opinion, is it more likely that the output will look strange/abnormal, or is it more likely to resemble the dataset? *Why?*

solution

It is more likely that the output will look strange or abnormal.
A standard autoencoder (AE) focuses on learning to reconstruct data points from the training set but does not ensure that random points in the latent space will produce valid data. The latent space is not regularized, so points chosen randomly may not represent meaningful data.
As a result, when we pick a random point and feed it to the decoder, the output is unlikely to resemble the dataset. Models like Variational Autoencoders (VAEs) handle this better because they organize the latent space in a way that supports sampling valid outputs.

Part b

List at least three problems with the method in part (a) for generating data similar to the dataset. Then, describe how a **VAE** solves these issues.

solution

1. The latent space in a standard autoencoder (AE) is unstructured, so random points sampled from it are unlikely to represent meaningful data.
2. AEs do not generalize well to unseen latent points because the decoder is only trained to reconstruct data from the training set.
3. The method lacks a probabilistic framework, meaning the outputs are not guaranteed to resemble the dataset's distribution.

A Variational Autoencoder (VAE) solves these issues by regularizing the latent space with a probabilistic prior (usually a Gaussian), ensuring random points correspond to meaningful data. VAEs also encourage smooth latent space coverage with KL divergence, improving generalization, and align the output distribution with the dataset for more realistic samples.

Part c

Suppose that, during the training of the AE, we add Gaussian noise with mean zero and variance $(0.05 \times R)$ to the encoder's output. Here, R is defined as the mean of the squared distances of the latent points from their center and is updated at each training step. Does the decoder trained via this approach perform *better* than a typical AE decoder? What we mean by "better" is: if a random point in the latent space is chosen by chance, is its output more likely to resemble a sample from the dataset? Which of the two approaches is more likely to yield an output that looks like actual data?

solution

Adding Gaussian noise to the encoder's output during training makes the decoder perform better than a typical AE decoder.
The noise helps the autoencoder learn a smoother and more robust latent space, making the model better at generalizing and reconstructing data even when latent points are slightly disturbed. This increases the chance that random points in the latent space will produce outputs similar to the dataset.
Overall, this approach is more likely to create realistic outputs and avoids overfitting compared to a standard AE.

Part d

Does VAE have any advantage over the method presented in part (c)? What is the key difference between these two methods?

solution

Yes, a VAE has an advantage over the method in part (c).
The key difference is that a VAE imposes a clear probabilistic structure on the latent space using a prior distribution, ensuring better organization and alignment for sampling. The method in part (c) uses Gaussian noise for regularization, which improves robustness but does not provide this structured framework.
A VAE's advantage lies in its smoother latent space and more reliable generation of realistic outputs.

Question 2

In this exercise, we want to become more familiar with ML estimation and its relationship to VAE.

Part a

Suppose our dataset is

$$D = \{x_1, x_2, \ldots, x_n\}.$$

Study the concept of *maximum likelihood* and explain why the parameters of the distribution must be chosen to maximize the following expression:

$$\sum_{i=1}^{n} \log p_\theta(x_i).$$

Note that $p_\theta(x_i)$ indicates that the probability of seeing x_i in the output depends on the parameters θ.

solution

Maximum Likelihood Estimation (MLE) finds the parameters θ that make the dataset $D = \{x_1, x_2, \ldots, x_n\}$ most likely.
The likelihood is:

$$L(\theta) = \prod_{i=1}^{n} p_\theta(x_i),$$

and we take the log to make it easier to work with:

$$\log L(\theta) = \sum_{i=1}^{n} \log p_\theta(x_i).$$

Maximizing $\sum_{i=1}^{n} \log p_\theta(x_i)$ is the same as maximizing the likelihood, but simpler to compute. This helps us find the best θ to explain the data.

Part b

Show the equivalence between minimizing the cross-entropy error and performing ML estimation.

solution

The cross-entropy error is defined as:

$$\text{Cross-Entropy} = -\frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i).$$

Maximizing the log-likelihood in ML estimation is:

$$\log L(\theta) = \sum_{i=1}^{n} \log p_\theta(x_i).$$

Minimizing the cross-entropy is equivalent to minimizing $-\log L(\theta)$, which is the negative log-likelihood. Therefore, minimizing the cross-entropy error directly corresponds to performing Maximum Likelihood Estimation.

Part c

We can say that one of the goals of VAE is to have a *generative model* whose output distribution resembles the distribution of the dataset. In a VAE—just like in standard neural networks—we intend to use a *stochastic gradient descent (SGD)* algorithm. Typically, in ordinary neural networks, we would aim to maximize $\sum_{i=1}^{n} \log p_\theta(x_i)$ for a mini-batch at each step. Then, for each input, we avoid making an overly drastic update to the network parameters; instead, we want to gradually increase the probability of generating an output similar to that input. Over the course of training, seeing that this log-probability increases is precisely in line with the concept of the *ELBO* as well. Indeed:

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

Here, θ denotes the decoder parameters and ϕ denotes the encoder parameters.

(i) Show that the KL divergence is nonnegative and explain how that fact arises in the above derivation.

solution

The KL divergence is defined as:

$$D_{\mathrm{KL}}(q(z) \,\|\, p(z)) = \int q(z) \log \frac{q(z)}{p(z)}\, dz.$$

Rewriting it as:

$$-D_{\mathrm{KL}}(q(z) \,\|\, p(z)) = -\int q(z) \log \frac{q(z)}{p(z)}\, dz.$$

Using Jensen's inequality, the term inside the integral satisfies:

$$-\int q(z) \log \frac{q(z)}{p(z)} \leq \log \int q(z) \frac{p(z)}{q(z)}\, dz.$$

Simplifying the right-hand side:

$$\log \int p(z)\, dz = \log 1 = 0.$$

Thus:

$$-D_{\mathrm{KL}}(q(z) \,\|\, p(z)) \leq 0,$$

which implies $D_{\mathrm{KL}}(q(z) \,\|\, p(z)) \geq 0$.
This non-negativity in the derivation of the ELBO ensures that $\log p_\theta(x_i)$ is greater than or equal to the ELBO:

$$\log p_\theta(x_i) \geq \mathbb{E}_z\big[\log p_\theta(x_i \mid z)\big] - D_{\mathrm{KL}}\big[q_\phi(z \mid x_i) \,\|\, p_\theta(z)\big].$$

Maximizing the ELBO indirectly maximizes $\log p_\theta(x_i)$ by reducing the KL divergence and aligning $q_\phi(z \mid x_i)$ with $p_\theta(z \mid x_i)$, which ensures better generation and a tighter bound.

(ii) Explain why, in various VAE implementations, the expression $\mathbb{E}_z\big[\log p_\theta(x_i \mid z)\big]$ is commonly regarded as the cross-entropy loss measured between the distribution of the real data and the distribution generated by the decoder.

solution

The term $\mathbb{E}_z\big[\log p_\theta(x_i \mid z)\big]$ is the expected log-probability of the data point x_i under the decoder's predicted distribution $p_\theta(x_i \mid z)$, where z is sampled from $q_\phi(z \mid x_i)$.
This term is regarded as the cross-entropy loss because it quantifies the difference between the true data distribution and the distribution generated by the decoder. Specifically, $-\mathbb{E}_z[\log p_\theta(x_i \mid z)]$ corresponds to the cross-entropy loss, which combines both the reconstruction error (how well the decoder recreates x_i) and the match between distributions. By maximizing $\mathbb{E}_z[\log p_\theta(x_i \mid z)]$, the decoder is encouraged to generate outputs that closely resemble the real data.

Question 3

Why, in a VAE, do we assume that the latent space distribution is Gaussian? (Apart from the fact that it is simpler to compute, please note other reasons as well.) Investigate whether, in practice, they also use other distributions besides Gaussian.

solution

In a VAE, the latent space is often assumed to follow a Gaussian distribution because it ensures a smooth and continuous latent space, allowing for meaningful interpolation and gradual changes in the generated outputs. The Gaussian distribution also has useful mathematical properties, such as closed-form computation of the KL divergence, making training simpler and more efficient. Additionally, the Gaussian prior is flexible and aligns with the Central Limit Theorem, which suggests that many complex data distributions can be approximated by a Gaussian. In practice, other distributions are also used depending on the data or task. For example, multimodal distributions like Gaussian Mixtures can better model datasets with distinct clusters, and constrained distributions like Beta or von Mises are used when the latent space has specific constraints. Implicit distributions are also explored for complex data where explicit parametric forms may not be ideal. Despite this, Gaussian priors remain popular due to their simplicity and effectiveness.

Question 4

Read the paper on β-VAE and answer the following questions.

part a

Summarize the idea of β-VAE and explain how it differs from a standard VAE.

solution

The β-VAE is an improved version of the standard Variational Autoencoder (VAE) that focuses on learning disentangled representations of data. It adds a hyperparameter $\beta > 1$ to the loss function, which changes how much the model balances reconstruction accuracy and the independence of the latent variables. The loss function becomes:

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

Unlike a standard VAE (where $\beta = 1$), β-VAE puts more weight on the KL divergence term. This forces the model to use the latent space more efficiently and encourages the discovery of independent factors in the data, making the learned representations easier to understand.
The main difference is that β-VAE can learn disentangled representations better than a standard VAE, but this comes with a trade-off. A higher β reduces the reconstruction quality because the model has to compress the data more in the latent space. This means that the choice of β is crucial for balancing the trade-off between disentanglement and reconstruction quality.

part b

Based on the information given in Section 2 of that paper, describe the importance and function of the *disentanglement metric*.

solution

The disentanglement metric in β-VAE measures how well the model separates different generative factors of the data into individual latent dimensions. This is important because disentangled representations are easier to interpret and can be used for tasks like classification or generation.
The metric works by checking if a simple linear classifier can predict a specific generative factor (like size or position) based on differences in the latent space. A high score means the model has successfully learned independent and interpretable latent representations. This metric is useful for comparing β-VAE to other models, like InfoGAN or standard VAEs, in a more objective way than just looking at outputs.