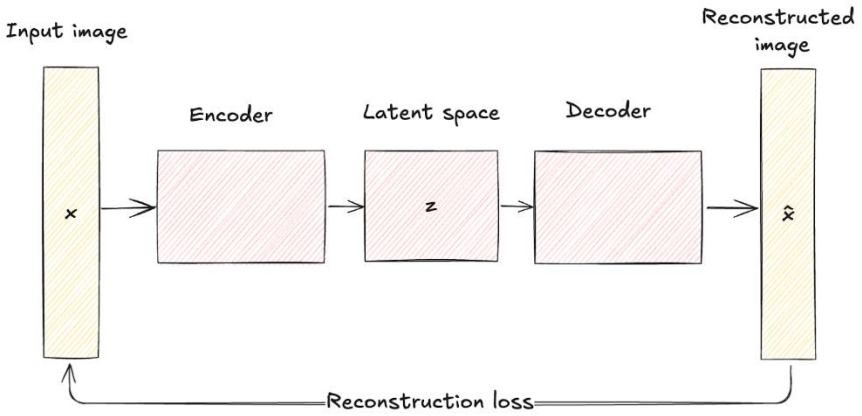
Semester: VIII Subject: Advanced AI Academic Year:2024-2025

Module 3

Variational Autoencoders (VAE)

Autoencoders are neural network architectures that are intended for the compression and reconstruction of data. It consists of an encoder and a decoder; these networks are learning a simple representation of the input data. Reconstruction loss ensures a close match of output with input, which is the basis for understanding more advanced architectures such as VAEs. The encoder aims to learn efficient data encoding from the dataset and pass it into a bottleneck architecture. The other part of the autoencoder is a decoder that uses latent space in the bottleneck layer to regenerate images similar to the dataset. These results backpropagate the neural network in the form of the loss function.



What is a Variational Autoencoder?

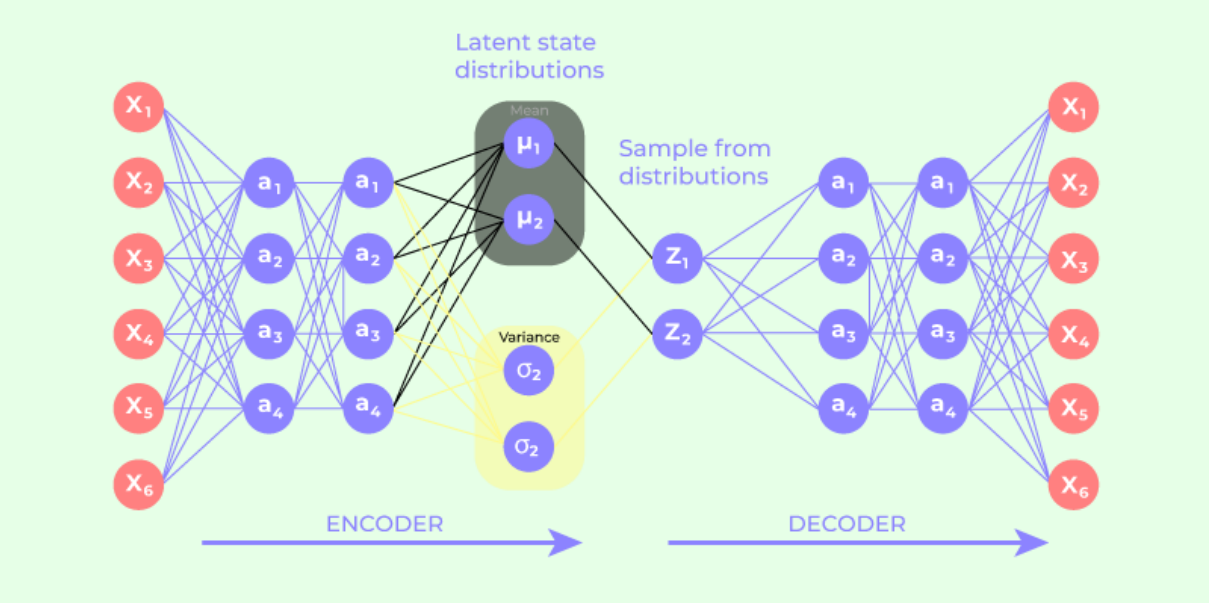
Variational autoencoder was proposed in 2013 by Diederik P. Kingma and Max Welling at Google and Qualcomm. A variational autoencoder (VAE) provides a probabilistic manner for describing an observation in latent space. Thus, rather than building an encoder that outputs a single value to describe each latent state attribute, we'll formulate our encoder to describe a probability distribution for each latent attribute. It has many applications, such as data compression, synthetic data creation, etc.
Variational autoencoder is different from an autoencoder in a way that it provides a statistical manner for describing the samples of the dataset in latent space. Therefore, in the variational autoencoder, the encoder outputs a probability distribution in the bottleneck layer instead of a single output value.

Architecture of Variational Autoencoder:

- The encoder-decoder architecture lies at the heart of Variational Autoencoders (VAEs), distinguishing them from traditional autoencoders. The encoder network takes raw input data and transforms it into a probability distribution within the latent space.
- The latent code generated by the encoder is a probabilistic encoding, allowing the VAE to express not just a single point in the latent space but a distribution of potential representations.
- The decoder network, in turn, takes a sampled point from the latent distribution and reconstructs it back into data space. During training, the model refines both the encoder and decoder parameters

to minimize the reconstruction loss – the disparity between the input data and the decoded output. The goal is not just to achieve accurate reconstruction but also to regularize the latent space, ensuring that it conforms to a specified distribution.

- The process involves a delicate balance between two essential components: the reconstruction loss and the regularization term, often represented by the Kullback-Leibler divergence. The reconstruction loss compels the model to accurately reconstruct the input, while the regularization term encourages the latent space to adhere to the chosen distribution, preventing overfitting and promoting generalization.
- By iteratively adjusting these parameters during training, the VAE learns to encode input data into a meaningful latent space representation. This optimized latent code encapsulates the underlying features and structures of the data, facilitating precise reconstruction. The probabilistic nature of the latent space also enables the generation of novel samples by drawing random points from the learned distribution.



How Does a VAE Work?

1. **The Distribution Trick:** Instead of a point in the latent space, the encoder of a VAE outputs the parameters that define a probability distribution (usually mean and variance). During training, we sample a point from this distribution to feed into the decoder.

2. **The Reparameterization Trick:** This is the clever part. Directly backpropagating gradients through random sampling is tricky. The reparameterization trick lets us express the sampled point in the latent space as a deterministic function of the distribution parameters and an external random variable. This allows for proper training.

3. **Loss Function:** Beyond Reconstruction: The VAE's loss function has two parts:

- **Reconstruction Loss:** Just like in an autoencoder, this part ensures that the decoder accurately reconstructs the input.
- **KL Divergence:** This is where the probabilistic twist comes in. The Kullback-Leibler (KL) divergence measures how much the encoder's learned distribution diverges from a standard prior distribution (often a standard Gaussian). This encourages a well-structured, regular latent space.

Mathematics behind Variational Autoencoder:

Variational autoencoder uses KL-divergence as its loss function, the goal of this is to minimize the difference between a supposed distribution and original distribution of dataset.
Suppose we have a distribution z and we want to generate the observation x from it.

In other words, we want to calculate p(z|x). We can do it by following way:

$$p\left(z|x\right) = \frac{p(x|z)p(z)}{p(x)}$$

But, the calculation of p(x) can be quite difficult

$$p(x)=\int p(x|z)p(z)dz$$

This usually makes it an intractable distribution. Hence, we need to approximate p(z|x) to q(z|x) to make it a tractable distribution. To better approximate p(z|x) to q(z|x), we will minimize the KL-divergence loss which calculates how similar two distributions are:

$$\text{Min } KL(q(z|x) \,||\, p(z|x))$$

By simplifying, the above minimization problem is equivalent to the following maximization problem :

$$E_{q(z|x)} \log p(x|z) - KL(q(z|x) \,||\, p(z))$$

The first term represents the reconstruction likelihood and the other term ensures that our learned distribution q is similar to the true prior distribution p.

Thus our total loss consists of two terms, one is reconstruction error and other is KL-divergence loss:

$$Loss = L\left(x, \hat{x}\right) + \sum_j KL\left(q_j\left(z|x\right) || p\left(z\right)\right)$$