https://lilianweng.github.io/posts/2018-08-12-vae/

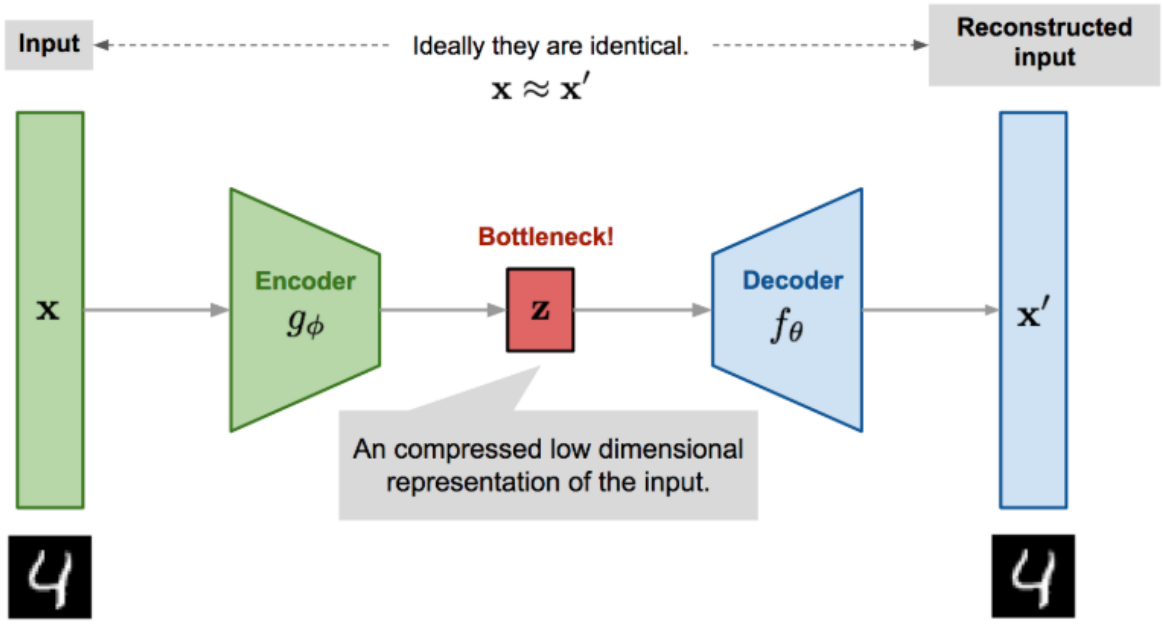
https://eugeneyan.com/writing/autoencoders-vs-diffusers/

Autoencoder

- **Autoencoder** is a neural network designed to learn an identity function in an unsupervised way to reconstruct the original input while compressing the data in the process so as to discover a more efficient and compressed representation.
- The idea was originated in the 1980s, and later promoted by the seminal paper by Hinton & Salakhutdinov, 2006.

It consists of two networks:

- *Encoder* network: It translates the original high-dimension input into the latent low-dimensional code.
- *Decoder* network: The decoder network recovers the data from the code



$$L_{\text{AE}}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2$$

The model contains an encoder function **g(.)** parameterized by **φ** and a decoder function **f(.)** parameterized by θ.

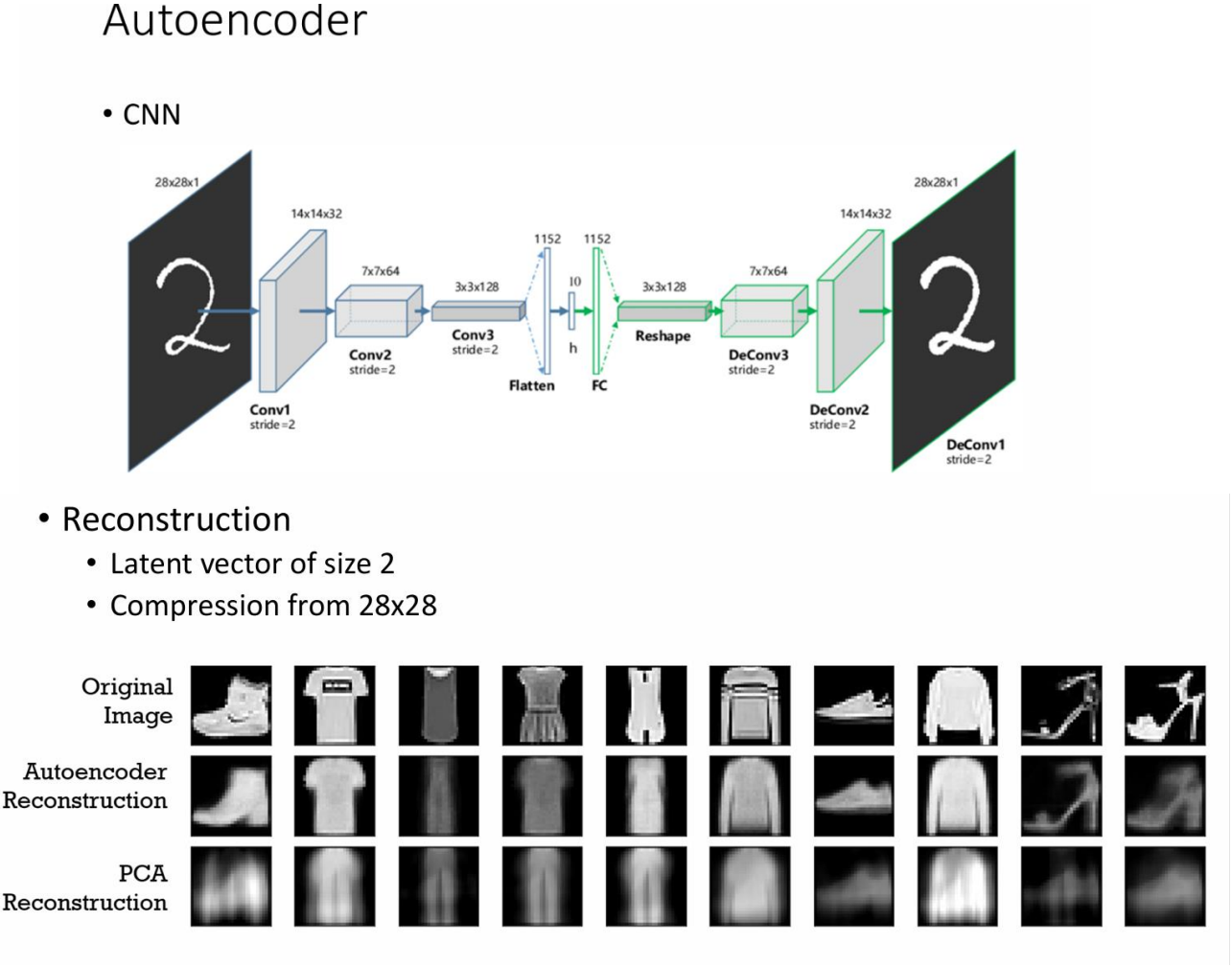
The low-dimensional code learned for input x in the bottleneck layer is $\mathbf{z=g_{\varphi}(x)}$ and the reconstructed input is $\mathbf{x'=f_{\theta}(g_{\varphi}(x))}$.

The parameters (θ,φ) are learned together to output a reconstructed data sample same as the original input, $x \approx f_{\theta}(g_{\varphi}(x))$, or in other words, to learn an identity function. There are various metrics to quantify the difference between two vectors, such as cross entropy when the activation function is sigmoid, or as simple as MSE loss

Applications

1. **Data compression using Auto encoder vs PCA:**

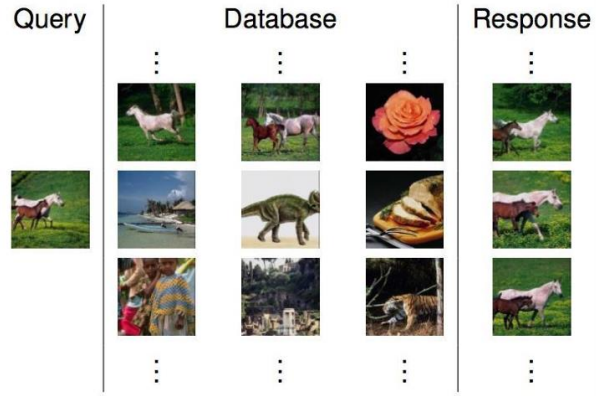
- The encoder network reduces the data's size, similar to PCA.
- A well-compressed form captures key hidden features and ensures accurate data reconstruction. Compression methods also make sharing data faster and more efficient.



2. **Image Retrieval**: During image retrieval, the system compares the compressed representations (latent vectors) of images instead of comparing the original high-dimensional images. The system searches for images with similar latent vectors in the database, **which is faster and computationally less expensive** than comparing raw pixel data.

`

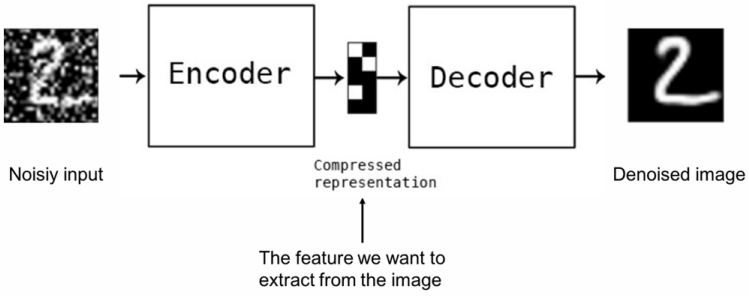
- Image retrieval
  - Dimensionality reduction helps



3. **Image denoising:**

Autoencoder – application

- Denoising



4. Image colorization

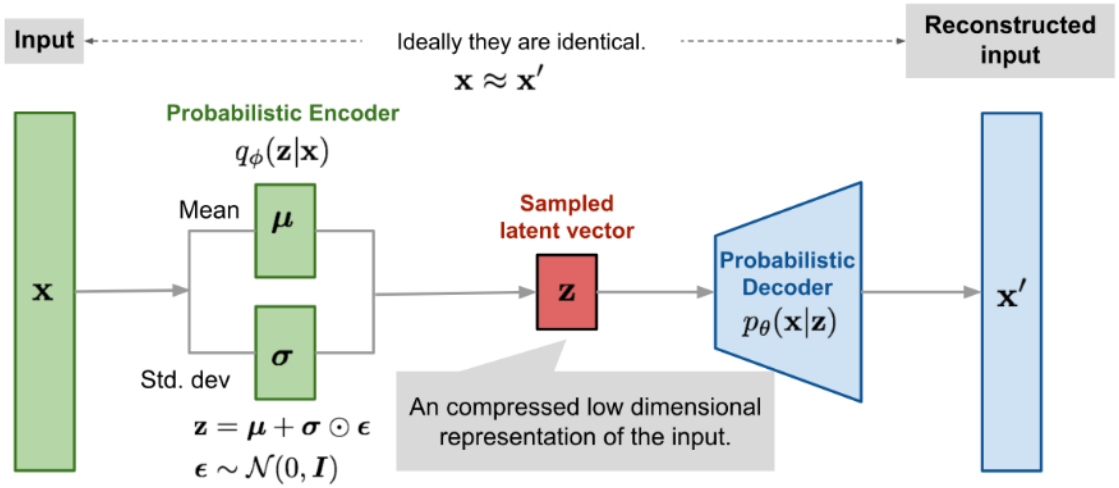
- Image colorization

5. Anomaly detection:

- Anomaly detection

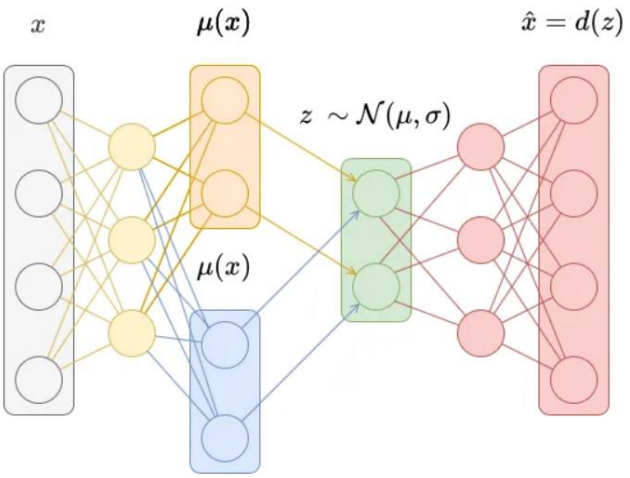
VAE: Variational Autoencoder

- The idea of **Variational Autoencoder** (Kingma & Welling, 2014), short for **VAE**, is actually less similar to the autoencoder model , but deeply rooted in the methods of **variational bayesian and graphical model**.
- Instead of mapping the input into a *fixed* vector, we want to map it into a distribution.
- **Why we need VAE**
- **Objective/ Goal of VAE: trying to find q(z/x) from where we sample z to generate new sample x' from p(x/z)**
-

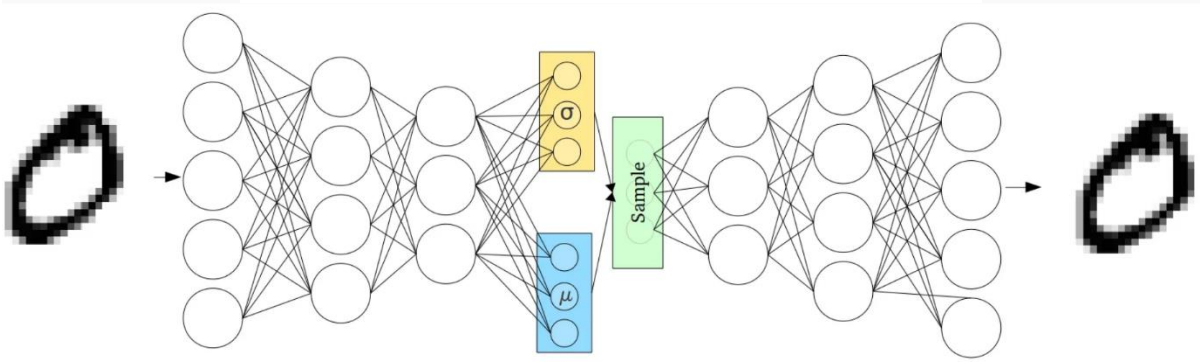


$$L_{\text{VAE}}(\theta, \phi) = -\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))$$

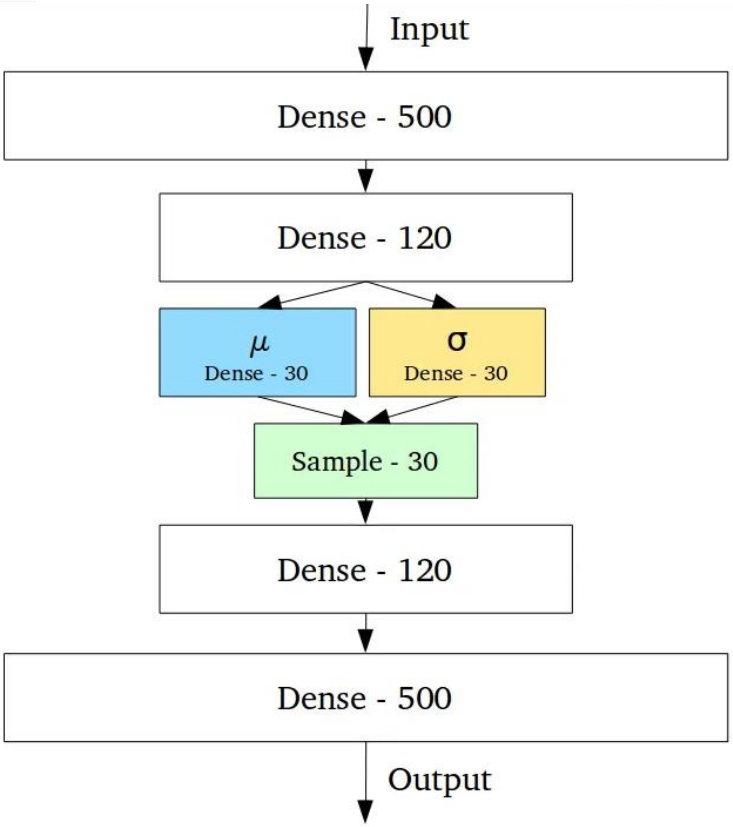
https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf



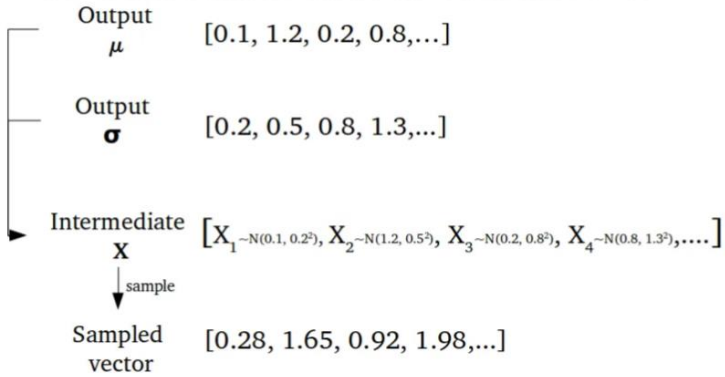
Source



A Standard Variational Autoencoder



They form the parameters of a vector of random variables of length n, with

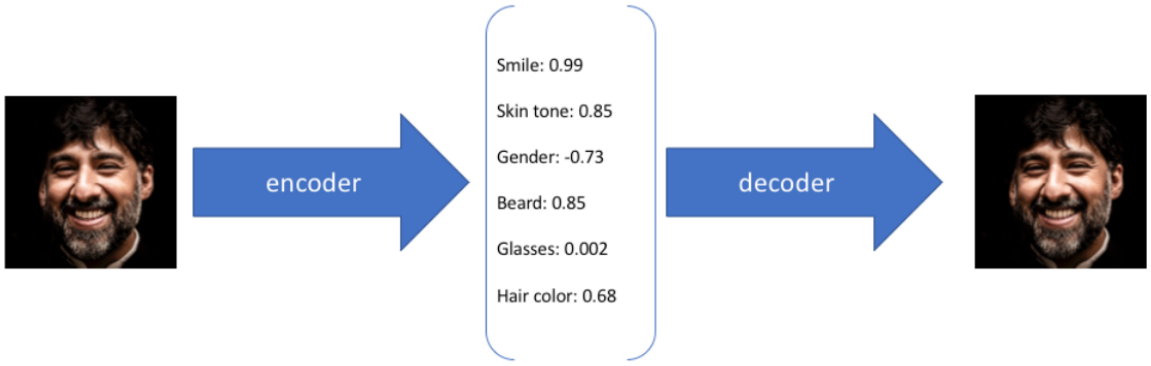


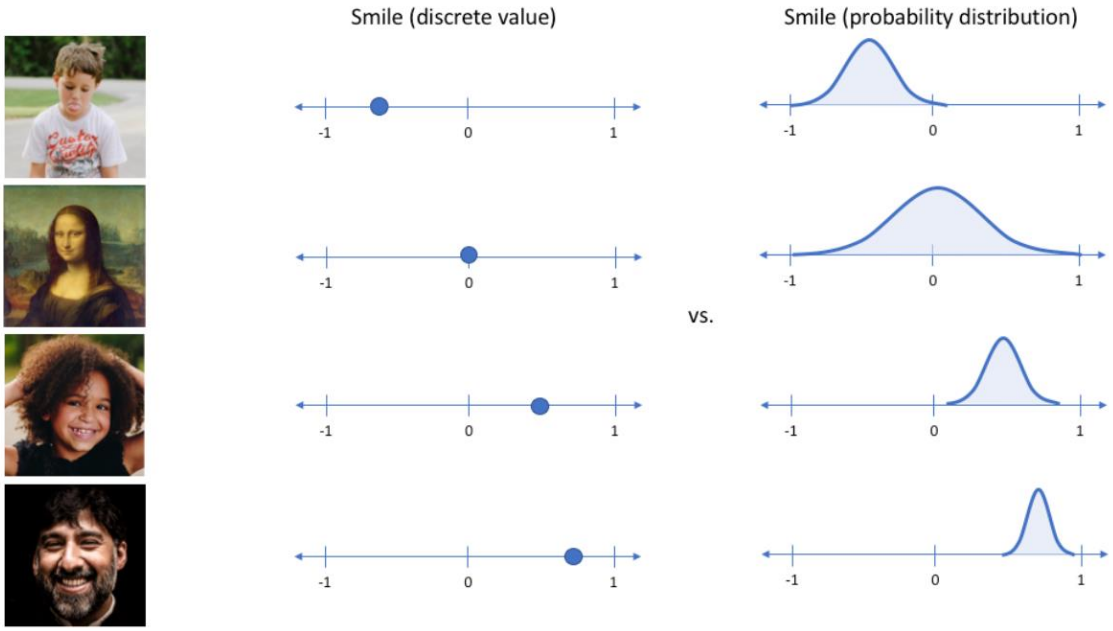
https://www.jeremyjordan.me/variational-autoencoders/

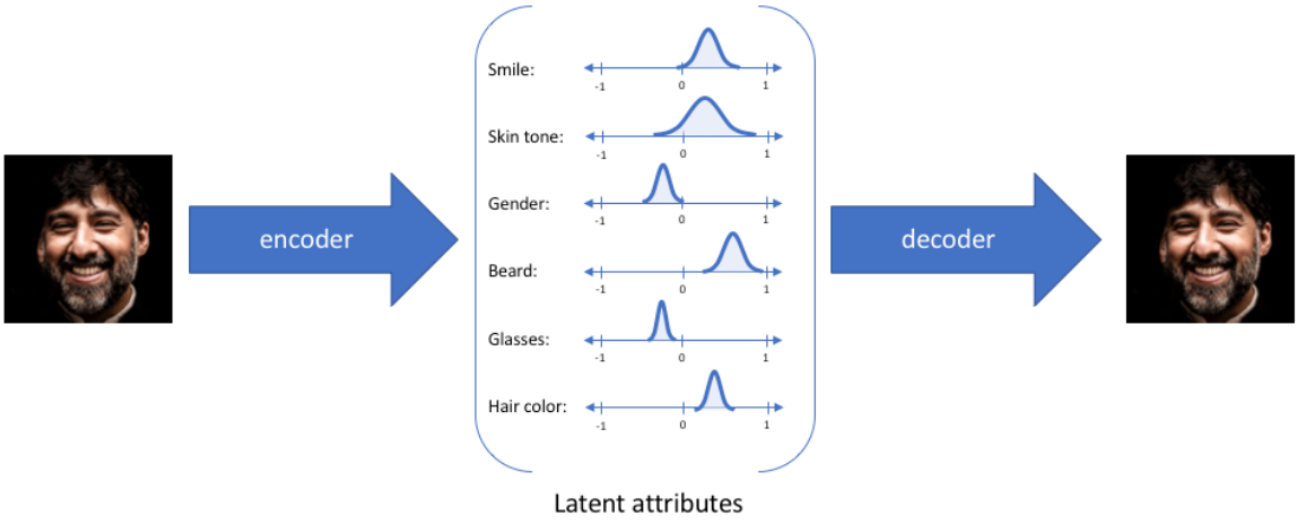
Intuition

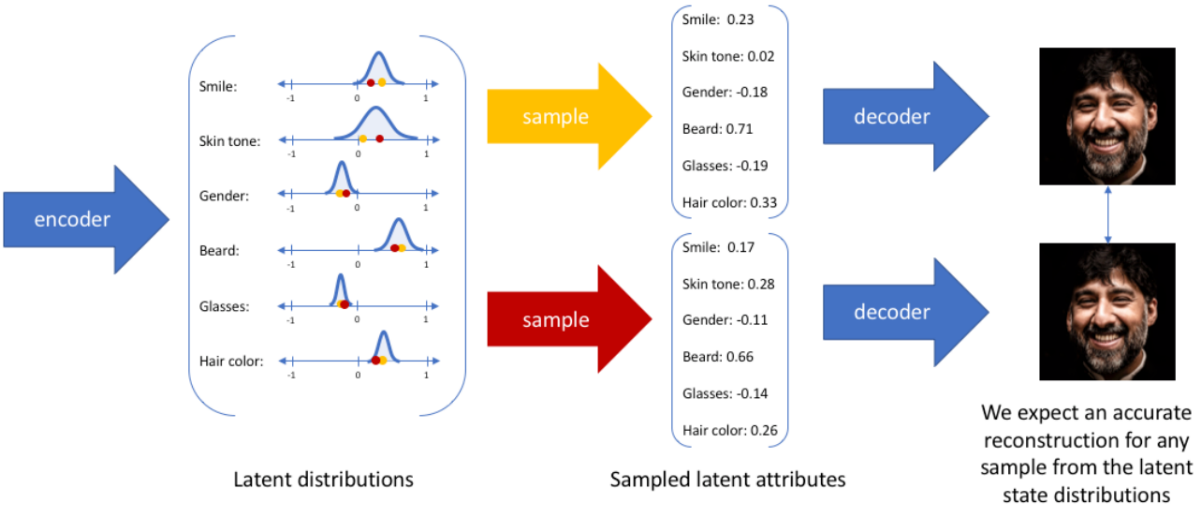
To provide an example, let's suppose we've trained an autoencoder model on a large dataset of faces with a encoding dimension of 6.

An **ideal autoencoder** will learn descriptive attributes of faces such as skin color, whether or not the person is wearing glasses, etc. in an attempt to describe an observation in some compressed representation.



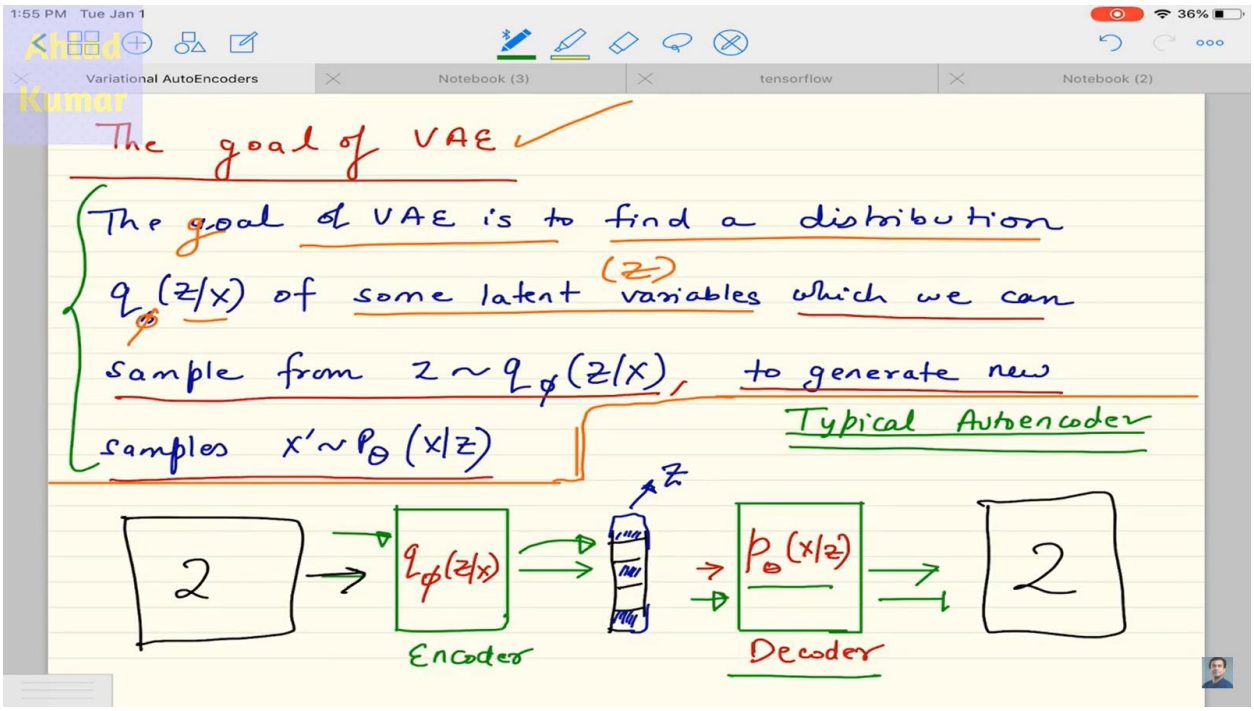


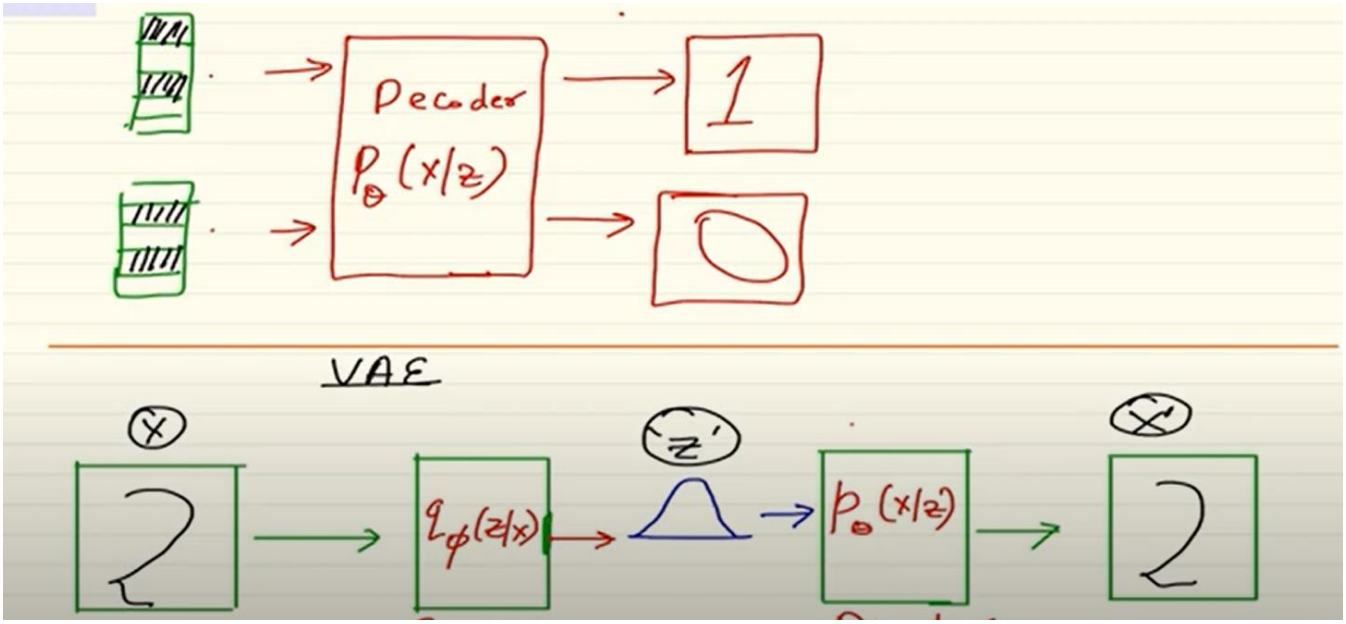




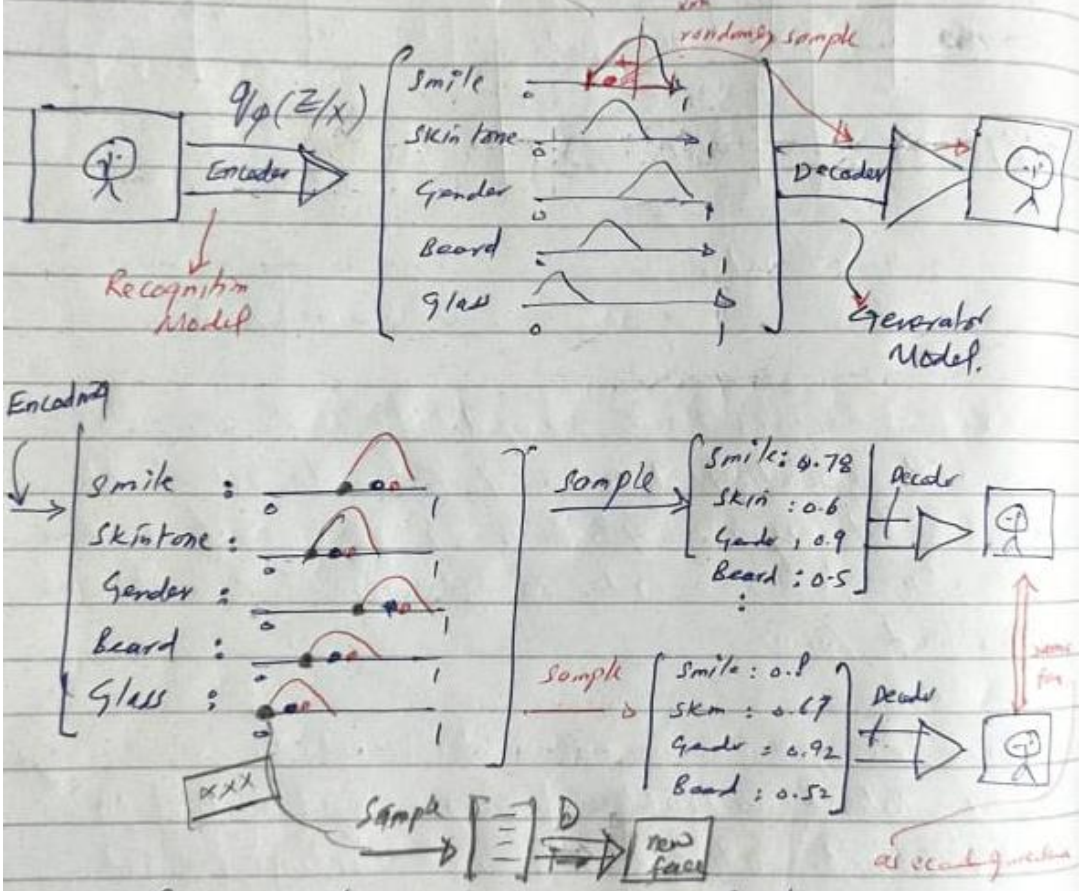
In autoencoder we generate the red samples, whereas in VAE we generate the green samples. So green samples are newly generated samples.

Typical flow of autoencoder:


1:55 PM Tue Jan 1
36%
Variational AutoEncoders
Notebook (3)
tensorflow
Notebook (2)
The goal of VAE
The goal of VAE is to find a distribution $q_\phi(z|x)$ of some latent (z) variables which we can sample from $z \sim q_\phi(z|x)$, to generate new samples $x' \sim p_\theta(x|z)$
Typical Autoencoder
2
$q_\phi(z|x)$
Encoder
z
$p_\theta(x|z)$
Decoder
2


Decoder
$p_\theta(x|z)$
1
0
VAE
X
2
$q_\phi(z|x)$
z'
$p_\theta(x|z)$
X'
2

With this approach we now represent each latent attribute for a given input as a probability distribution. When decoding we will randomly sample from each latent state distribution to generate a vector as input for decoder model.



- By constructing our encoder model to output range of possible values (as statistically distributed) from which we will randomly sample to feed into our decoder model. The values which are near each other in latent space must correspond to similar reconstruction.

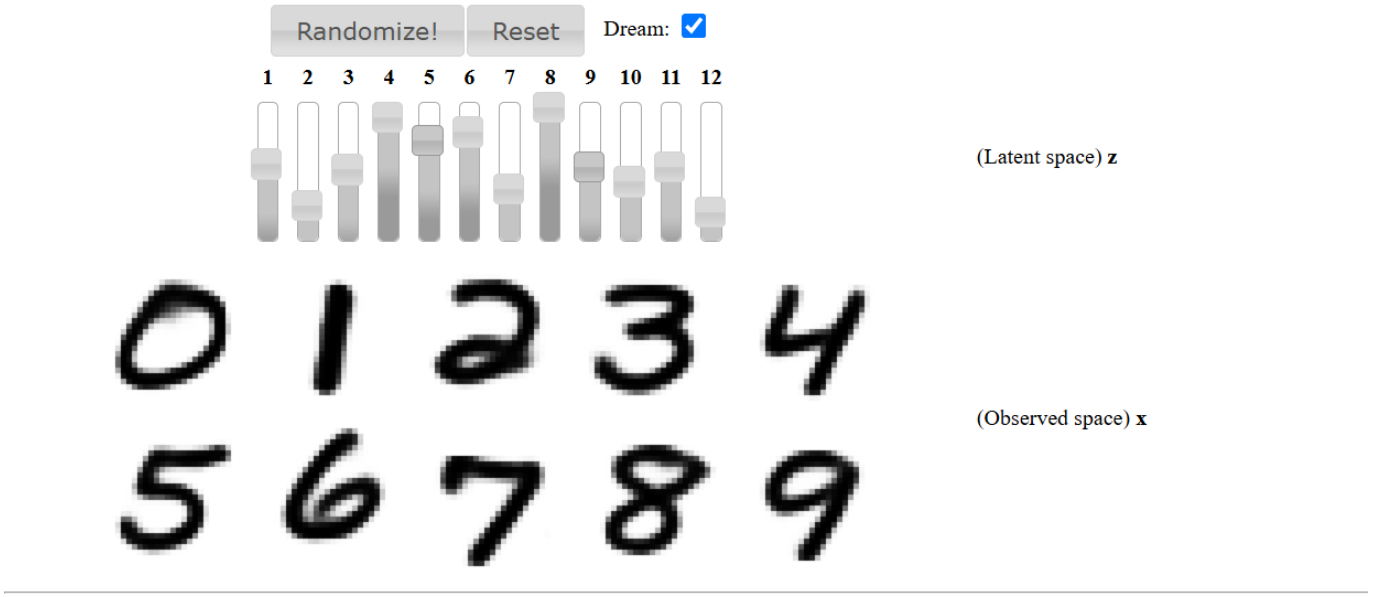
http://dpkingma.com/sgvb_mnist_demo/demo.html

his image appears to demonstrate a **Variational Autoencoder (VAE)** for generating handwritten digits from the **MNIST dataset**, which contains digits from 0 to 9.

Key Components in the Image

- **Sliders (Latent Space - z):**
- **Observed Space - x:** generated samples
- **Dream Mode (Checked):**
- **Randomize and Reset Buttons:**

The decoder then maps the updated latent variable z to a new digit, showing how different parts of the latent space influence the digit's style(rounder), shape (thinner), or features.



In order to understand and derivation of loss function of VAE, we try to understand the basics of probability theories.

Some basic concepts of **probability** which are important for understanding **Variational Autoencoders (VAE)**. Let's break down the key points:

- **p(x)**: This represents the probability of a random variable **x**. It tells us how likely **x** is to occur. For example, in the case of images, **x** could represent pixel values or features of an image.
- **p(x|y)**: This is the **conditional probability**. It represents the probability of **x** occurring given that **y** has already happened. For instance, if **y** represents some condition or label (like the class of an image), **p(x|y)** tells us how likely **x** is to be a certain value given **y**.
- **E[ ]**: This represents the **expectation** or **expected value** of a random variable, essentially the average value you expect from a distribution.
- **KL Divergence**: This is a measure of how one probability distribution diverges from a second, expected probability distribution. It plays an important role in VAEs by measuring how much the learned distribution (the encoder's output) deviates from the true distribution.

1. Basic Probability P(X):

- **Definition**: P(X) is defined as the probability of a random variable XXX.
- **Example**:
  - **Scenario**: Rolling a die once and determining the probability of getting the value “3”.
  - **Explanation**:
    - The random variable XXX corresponds to the sample space {1,2,3,4,5,6}, which represents all outcomes of rolling a fair die.
  - **Probability**:
    - P(3)=1/6, as each outcome has equal likelihood (one out of six possible values).

2. Conditional Probability P(X|Y)):

- **Definition**: P(X|Y)) represents the probability of X happening, given that Y has already occurred. This is also known as **conditional probability**.
- **Example**:
    - **Scenario**: Rolling a fair die and calculating the probability of getting "3", provided that the outcome is constrained to the **odd values** only.
    - **Steps**:
        - Reduced Sample Space (only odd values): {1,3,5}.
        - From this new space, the probability of getting 3 is recalculated.
    - **Probability**: P(3|odd)=1/3
    - Here, the total number of odd outcomes is 3, and the outcome "3" is one of them.
- **Observation**: The conditional probability P(3|odd) is increased compared to the unconditional probability P(3) because the sample space is reduced.

Key Insights:

- P(X) calculates basic probabilities over an entire sample space.
- P(X|Y) recalculates probabilities when additional information (conditions) restricts the sample space.

Bayes' Theorem and the **Theorem of Total Probability**, providing equations and explanations for these fundamental concepts in probability.

3. Bayes' Theorem

Bayes' Theorem allows us to determine the *posterior probability* $P(Y|X)$, given prior probabilities and conditional probabilities.

The Formula:

joint prob

$$P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)} \quad \text{or} \quad P(Y|X) = \frac{P(X,Y)}{P(X)}$$

- **Explanation of terms:**
  - $P(Y|X)$: **Posterior probability** (probability of Y given X).
  - $P(X|Y)$: **Likelihood** (conditional probability of X given Y).
  - $P(Y)$: **Prior probability** (initial or prior information about Y).
  - $P(X)$: **Marginal probability** of X (normalizing factor).

Bayes' Theorem uses conditional probabilities to "reverse" a relationship, enabling us to calculate the probability of Y, given X, using prior information.

4. Theorem of Total Probability

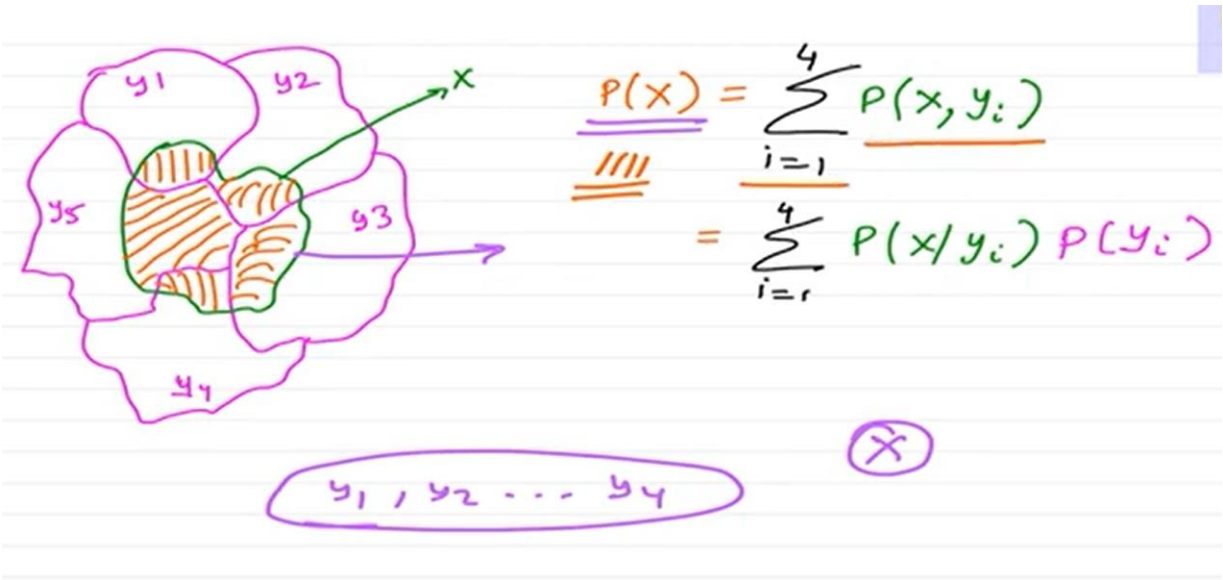
$P(X, Y_1)$

The Theorem of Total Probability states that the probability of an event X can be determined by considering all mutually exclusive events $Y_1, Y_2, ..., Y_n$ that cover the sample space.

The Formula:

$$P(X) = \sum_{i=1}^{n} P(X|Y_i)\, P(Y_i)$$

- Here:
  - $\{Y_1, Y_2, ..., Y_n\}$: Set of **mutually exclusive events** (no overlap).
  - X: The event of interest.
  - $P(X|Y_i)$: Conditional probability of X, given Y_i.
  - $P(Y_i)$: Prior probability of Y_i.



Graphical Representation:

A Venn diagram is included, where the mutually exclusive events Y_1, Y_2, Y_3, Y_4 are shown. Event X intersects these events, showing that the probability of X can be represented as the sum of probabilities over these partitions.

Example for $n = 4$:

When there are **4 mutually exclusive events** Y_1, Y_2, Y_3, Y_4, the probability of X is:

$$P(X) = P(X, Y_1) + P(X, Y_2) + P(X, Y_3) + P(X, Y_4)$$

or equivalently:

$$P(X) = P(X|Y_1)P(Y_1) + P(X|Y_2)P(Y_2) + P(X|Y_3)P(Y_3) + P(X|Y_4)P(Y_4)$$

Combining Bayes' Theorem with Total Probability

Substituting the Total Probability formula for $P(X)$ into Bayes' Theorem gives:

$$P(Y|X) = \frac{P(X|Y)\,P(Y)}{\sum_{i=1}^{n} P(X|Y_i)\,P(Y_i)}$$

This is often used in real-world problems where X is observed, and we need to determine the most likely Y.

---

6. **Mutually Exclusive (ME) Events**:
    a. Events are **mutually exclusive** if they cannot occur simultaneously.
    b. In this case:
        i. The event of rolling an **even number** and the event of rolling an **odd number** are mutually exclusive because a die cannot result in both outcomes at the same time.
7. **Union of Mutually Exclusive Events**:
    a. When two events are mutually exclusive, the probability of their **union** (either event happening) is the sum of their individual probabilities:

    $P(A \cup B) = P(A) + P(B)$ if $A \cap B = \emptyset$

Example: Rolling a Die

- **Scenario**:
    - Rolling a standard die. The events of rolling an **even** number ({2, 4, 6}) and an **odd** number ({1, 3, 5}) are mutually exclusive.

Example: Rolling a Die

- **Scenario**:
  - Rolling a standard die. The events of rolling an **even** number ({2, 4, 6}) and an **odd** number ({1, 3, 5}) are mutually exclusive.
- **Steps**:
  1. Total possible outcomes = 6 (a six-sided die).
  2. Probability of rolling an even number $P(\text{even})$:
$$P(\text{even}) = \frac{3}{6}$$
  3. Probability of rolling an odd number $P(\text{odd})$:
$$P(\text{odd}) = \frac{3}{6}$$
- **Union of Events**:
  - The union of "rolling an even number" or "rolling an odd number" encompasses the entire sample space.
  - Thus:
$$P(\text{even or odd}) = P(\text{even}) + P(\text{odd})$$
$$P(\text{even or odd}) = \frac{3}{6} + \frac{3}{6} = 1$$
  - This result is logical since rolling a die must produce either an even or odd outcome with certainty.

Summary:

- The events **even** and **odd** are **mutually exclusive**.
- The **probability** of rolling either an even or odd number is: P(even or odd)=1/2+1/2=1
- This example emphasizes how mutually exclusive events contribute additively to the total probability when considering their union

Random variable:

Each pixel in an image can be **considered a random variable**, especially in probabilistic image analysis and machine learning models, where **we treat pixel values as random variables with certain distributions.**

Each pixel(random variable x1) has some specific range of values(distribution)

Height of student: (for a class a height x1) has some specific range of values (distribution)

Explanation of E(Expectation or Expected Value)

In probability and statistics, **expectation** (also called the **expected value** or **mean**) of a random variable represents the long-term average or the center of mass of the probability distribution of that variable. It is a measure of the central tendency, essentially giving you an idea of the "average" outcome you would expect if you were to repeat an experiment many times.

1. **In the case of a discrete random variable**: Imagine you roll a fair six-sided die. The possible outcomes are $1, 2, 3, 4, 5, 6$, each with probability $\frac{1}{6}$. The expected value of the die roll is the average value you would expect if you rolled the die many times. It can be calculated as:

$$E[\text{roll}] = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + 4 \cdot \frac{1}{6} + 5 \cdot \frac{1}{6} + 6 \cdot \frac{1}{6}$$

Simplifying this:

$$E[\text{roll}] = \frac{1 + 2 + 3 + 4 + 5 + 6}{6} = \frac{21}{6} = 3.5$$

So, the expected value of a fair die roll is 3.5, even though that result cannot occur when rolling the die. It is the "average" outcome over many rolls.

KL Divergence is a powerful tool in Variational Autoencoders (VAEs) for ensuring that the model learns a well-structured latent space.

By minimizing the KL divergence between the learned latent distribution and the prior distribution, VAEs can generate new data that is coherent and consistent with the original data, while maintaining a structured and regularized latent space..