High-dimensional data:

Consider a dataset containing information about houses for sale. Each house is described by various features:

1. Square footage of the house
2. Number of bedrooms
3. Number of bathrooms
4. Presence of a garage (1 for yes, 0 for no)
5. Presence of a swimming pool (1 for yes, 0 for no)
6. Presence of a garden (1 for yes, 0 for no)
7. Presence of air conditioning (1 for yes, 0 for no)
8. Distance to the nearest school
9. Distance to the nearest hospital
10. Distance to the city center
11. Crime rate in the neighborhood
12. Median household income in the neighborhood
13. Number of parks nearby
14. Number of restaurants nearby

If we have data on 1,000 houses, each represented by these 14 features, then the dataset becomes high-dimensional. In this example, the dataset has 14 dimensions, and each house can be thought of as a point in a 14-dimensional space, with each feature representing a different axis. Analyzing this high-dimensional dataset can be challenging due to the curse of dimensionality, especially if we want to build models to predict house prices or understand the relationships between different features.

To address the high dimensionality, we might use techniques like principal component analysis (PCA) for dimensionality reduction or regularisation methods when building predictive models to prevent overfitting. These techniques help to retain essential information while reducing the complexity of the data.

1. **High-dimension Sample Space:** In the previous example, we had a dataset with 14 different features for each house, resulting in a 14-dimensional space. In general, high-dimensional sample space refers to the original dataset, where each observation is represented by multiple features or attributes.
2. **Latent Space:** The latent space is a lower-dimensional space, typically of a much smaller dimension than the high-dimensional sample space. It is a compressed or abstract representation of the data. The idea is to transform the high-dimensional data into a more compact and meaningful representation, where each point in the latent space captures the essential information about an observation from the original data.
3. **Representation Learning:** Representation learning is the process of discovering this lower-dimensional latent space in an unsupervised or self-supervised manner. It means finding a more informative and compact representation of the data without explicit supervision or labeled examples.
4. **Mapping Function:** Once the lower-dimensional latent space is learned, we can use a mapping function to transform points in the latent space back into the high-dimensional original domain. This mapping function allows us to reconstruct the original observations from their compressed representations.

5. **Objective:** The overall objective is to learn a mapping that efficiently encodes the most

important information from the high-dimensional data into the latent space while also enabling accurate reconstruction of the original data.

The key idea here is that by finding a suitable latent space, we can potentially capture the underlying structure and patterns in the data more effectively. This can be useful for various tasks, including data compression, denoising, anomaly detection, and even generating new data that resembles the training set.

One popular technique used for representation learning is autoencoders. An autoencoder is a type of neural network that learns to reconstruct the input data from a bottleneck layer, which forms the latent space representation. By training the autoencoder to minimize the reconstruction error, it indirectly learns a meaningful and compact representation of the data in the latent space.

What is a latent space??

In representation learning, a point in the latent space refers to the encoded or compressed representation of an observation from the original high-dimensional data. The latent space is a lower-dimensional space, typically of much smaller dimensionality than the original data, and each point in this space represents a specific observation from the dataset.

To understand this better, let's consider an example with a simple 2D latent space. Suppose we have a dataset of 100 houses, and each house is represented by two features: square footage and number of bedrooms. This means our high-dimensional data is in a 2D space, with one axis representing square footage and the other representing the number of bedrooms.

Now, let's assume we apply representation learning and use an autoencoder to discover a lower-dimensional latent space of only one dimension (1D). In this case, each point in the 1D latent space represents a house, but now it is described by only one value (a single coordinate along the 1D axis) instead of two values (square footage and number of bedrooms).

The process of transforming a house from the high-dimensional space (square footage, number of bedrooms) to the 1D latent space involves an encoding step. The encoding function of the autoencoder takes the high-dimensional data point (square footage, number of bedrooms) and maps it to a single value in the 1D latent space.

Here's an example of how the encoding process might work:
Original data (High-dimensional space):
House 1: (2000 sq.ft., 3 bedrooms) House 2: (1500 sq.ft., 2 bedrooms) House 3: (1800 sq.ft., 4 bedrooms)
1D Latent space:
House 1: 0.7 House 2: -1.2 House 3: 0.2
Each value in the 1D latent space (0.7, -1.2, 0.2) represents the compressed representation of a respective house from the original dataset.

The idea behind the latent space is that it should capture the most important characteristics or patterns in the data, allowing us to effectively perform tasks like data reconstruction, anomaly detection, or even generating new data that shares similarities with the original dataset.