**Generative models**

Generative models are a class of machine learning models that are designed to learn an underlying probabilistic distribution and to generate new samples from that distribution. The objective of generative models is to learn the entire dataset’s distribution, which contrast to discriminative models that just learn a decision boundary between classes, and it is this feature that enables generative models to generate new samples that resemble data from the training set. Usually, generative models represent or model data by generating data in the form of Markov chains or generative iterative processes that approximate the same process as a Markov chain [93].

Generative models in machine learning must not be confused with probabilistic models from statistics.

* **Probabilistic model vs. generative model**
  + **Bayesiansk, x, y p(yIx)**
  + **Normalizing flows**
  + **Bridge the gap**
  + **Matnat vs ifi**
  + **Statistic vs ML**

The objective of generative models is to learn the data distribution p(x) of the training data x. However, this is a more complex problem than it seems. In many applications, the training data contains patterns or features that can be represented as an unknown latent distribution [11]. This might be best described with a story, namely Plato’s Allegory of the Cave, where the connection to the topic of generative models is from Luo (2022) [11].

Imagine a group of people are born and trapped in a cave and the only thing they can see is the wall of the cave, which sometimes contains shadows projected from the outside of the cave. These shadows are two-dimensional, which implies that everything that these people have ever observed is two-dimensional. These people don’t have any idea or the capacity to imagine any higher-dimensional concepts other than two-dimensional objects like the shadows on the cave wall. However, these shadows are generated by higher-dimensional objects that passed the cave opening. These higher-dimensional objects are abstract concepts to the people living in the cave. The shadows can be viewed as two-dimensional products of a function that takes higher-dimensional input, where variables such as color, size, three-dimensional shape, and other information may be lost in the shadow’s reflection on the wall which only contains information about the two-dimensional shape of the object.

The cave people can still draw some information from the shadows, and similarly, it is possible to approximate latent representations that describe more complex data and their distributions. Now the object of generative models in terms of this allegory is to learn the full information of what the shadows on the wall represent, that is learn the color, size, three-dimensional shape, etc. In this way, the shadows on the wall can be seen as compressed versions of the objects outside the cave, that is, the latent space representation of the real objects.

Now imagine that the cave people have the ability to create shadows on the wall and that people living outside the cave who only has seen three-dimensional objects, before they notice that there are shadows on the wall in the cave. If the outside people manage to understand the shadows that the cave people create on the wall and convert the shadows into the objects they represent. This would be analogous to a decoding of the shadows on the wall. If the outside people learn the shadow to object well enough, when the shadows represent objects the outside people are familiar with, then if a new shadow appear that they haven’t seen before, they should be able to understand what the object that cast the shadow looks like.

In the sense of generative models, the outside objects and their features represent the input, the outside object to shadow part is analogous to an encoder which encode or compress the original input, the shadows represent a latent representation of the outside object, the shadow to outside object part is analogous to a decoder/generator. To generate a new sample in this allegory would be similar to the cave people casting a shadow of an object that only the cave people know, then the outside people would construct a three-dimensional object with color, size, etc. based on the two-dimensional shadow.

Generative models have many applications in areas as images, video, and text, which results in great diversity in model architecture and methods and the way the models are trained and evaluated. The many applications of generative models have resulted in a many generative methods such as variational auto-encoders (VAEs) [21], generative adversarial networks (GANs) [7] and the more recent diffusion model based on nonequilibrium thermodynamics [91]. A consequence of this diversity is difficulty comparing different models and their results [95].

Further challenges in generative models are that they are notoriously unstable and hard to train due to issues such as mode collapse, uniform training distribution, or vanishing gradients amongst other [94]. Training generative models, is time consuming compared to other machine learning methods since they need to recreate a probability distribution that resemble the original data. This involves learning multiple complex relationships and a high number of correlations in the data [93]. Compared to discriminative models, which learn small patterns to distinguish between and separate a dogs and cats, a generative model must learn all the features of dogs and cats to be able to generate new images of them, which is a much more computational expensive procedure [93].

Selecting and using the correct training method and getting generative models to converge correctly is crucial for decent results and performance in specific applications, so is the choice of evaluation metric to evaluate the results. Evaluating a generative model is a task on its own, since they can be evaluated in many different ways, but for many models computing the likelihood is intractable. Additionally, many common evaluation methods are largely independent of each other, especially with high-dimensional data, that means that good performance with one evaluation metric doesn’t have a relationship with the performance with a different evaluation metric [95]. Therefore, it is important that the result of generative models is evaluated in terms of the application. In fact, [95] show that high likelihood is not necessary or sufficient for visually acceptable images to be produced.

Different models have different advantages/disadvantages which leads to different usefulness in distinct applications. The three main features of generative models that should be considered is the quality of the generated samples, the diversity of the generated samples and the speed of generating new samples. Achieving state-of-the-art results in all of these characteristics is extremely hard, hence it can be viewed as a generative trilemma. A generative model should aim to perform well in all of the three categories, but trade-offs has to be made, depending on the desired features and application.

A diagram of different types of samples

Description automatically generated

In terms of the mentioned models, the generative learning trilemma would look like the following:

GANs produce samples with high quality fast, but are hard to train, can quickly reach a mode collapse and not converge, leading to poor diversity. VAEs produce diverse samples fast but lack quality. Diffusion models produce diverse and high-quality samples but are notoriously slow to generate new samples.

Image quality might be the most important quality, since many applications require high quality samples, especially if the application is user interactive. But in many applications, the less frequent data points are important, so to capture these in the diversity of the generated samples is crucial. Additionally, to accomplish real-time interactive products, generation speed needs to be fast. Therefore, depending on the application, different generative models would have a better fit than others.

Usually, generative models are evaluated in terms of the quality and how realistic the newly generated samples are compared to the original input. This could be done by visual comparison, however, this is not a bulletproof or mathematically sound evaluation method, since even weak models can generate good exemplars, and with a collapsed model, the generated samples might be a training data example [93]. Generated images can look very good, but the underlying generated distribution could have underfitted. For example, if a model trained on both dogs and cats only generate great dog images, this would look like a great model, but to evaluate the model, it is required to know that the training dataset contains dogs and cats. Given difficulties of evaluating generative models, there is a demand to develop new methods of unbiasedly evaluation, as well as creating better quality generated samples [93].