Imaginative AI: Understanding Visual Creativity

At its core, Image Generative AI learns patterns from a set of training images and then uses those patterns to create new images.

Applications of Image Generative AI

Image Generative AI has numerous exciting applications across various fields:

1. **Art and Design**: Artists and designers use these models to create new pieces of art, design patterns, and even generate ideas for new projects. AI-generated art is becoming a unique genre, with some pieces even sold at art auctions.
2. **Entertainment and Media**: In the film and gaming industries, generative AI can create realistic characters, backgrounds, and special effects. This technology can significantly reduce the time and cost of producing high-quality visual content.
3. **Fashion**: Designers use generative AI to create new clothing designs, patterns, and accessories. It helps in visualizing new fashion trends and experimenting with innovative styles.
4. **Advertising and Marketing**: Companies use AI-generated images for creating eye-catching advertisements and promotional materials. It allows for the creation of customized and visually appealing content that can attract more customers.

Examples: https://www.usegalileo.ai/explore https://leonardo.ai/

How do Image generative models work

Like the ones we studied earlier the process here as well looks almost the same:

1. **Training Data**: The AI is fed a large number of images related to the type of visuals it will eventually create. For example, if we want an AI to generate pictures of cats, it needs to learn from thousands of cat images.
2. **Learning Phase**: The AI uses machine learning techniques to analyze these images. It identifies common patterns, such as shapes, colors, and textures.
3. **Generation Phase**: Once the AI has learned enough, it can start creating new images. It combines the patterns it learned during the training phase to generate visuals that are new yet resemble the training images.

Intro to Image Generative Model architectures

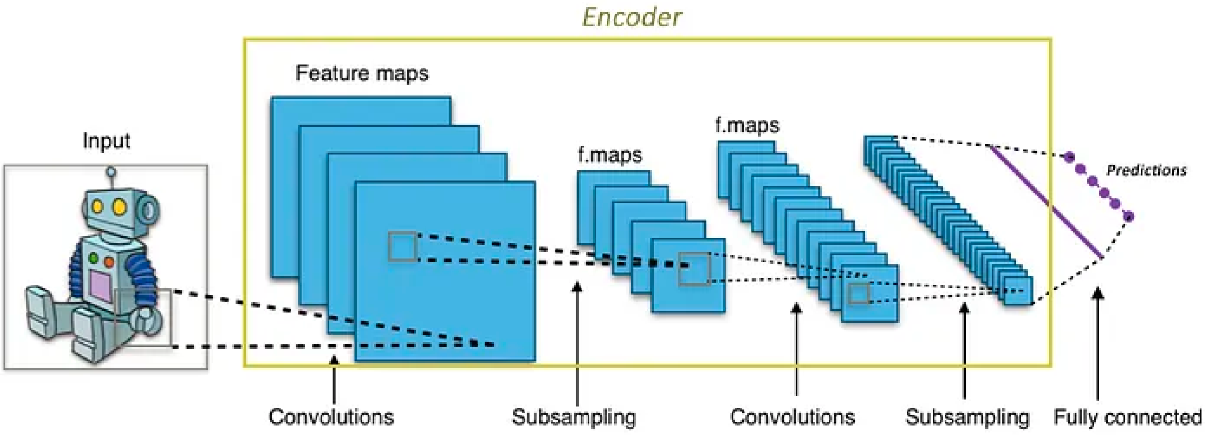
Several different models can generate images, but two of the most popular ones are Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). The base of these two models are still derived from the CNNS we studied before

VAEs (Variational Autoencoders)

Lets start by understanding an **autoencoder** .
An autoencoder network is actually a pair of two connected networks, an encoder and a decoder.

An encoder network takes in an input, and **converts it into a smaller, dense representation**, which the decoder network can use to convert it back to the original input.

- The CNN-based **encoder processes the input image through several convolutional layers, reducing its dimensions and extracting essential features**. it is simply is a network that takes in an input and produces a much smaller representation (the encoding)
- Decoder: **reconstructs the data back from its encoded form** (like expanding the summary back into the detailed book). The goal is to get an output identical with the input. **Normally the decoder architecture is the mirror image of the encoder.**

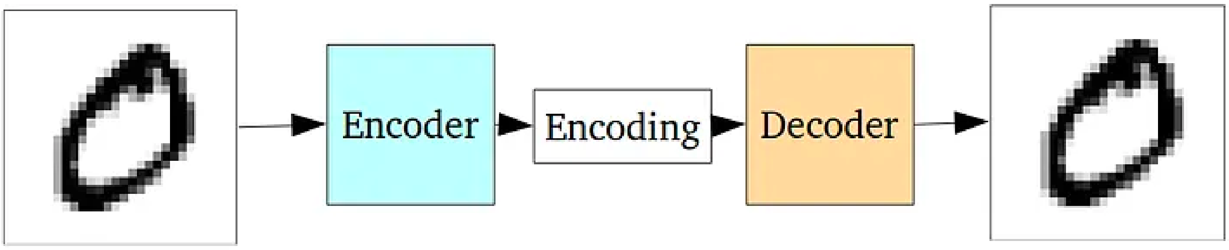


Transformers and VAEs

You may start observing a similarity from transformer models we studied for NLP

- Both models have an encoder that processes and compresses input information into a more abstract form.

- Both models have a decoder that takes the compressed information and generates meaningful output. In Transformers, it generates text; in VAEs, it generates images.

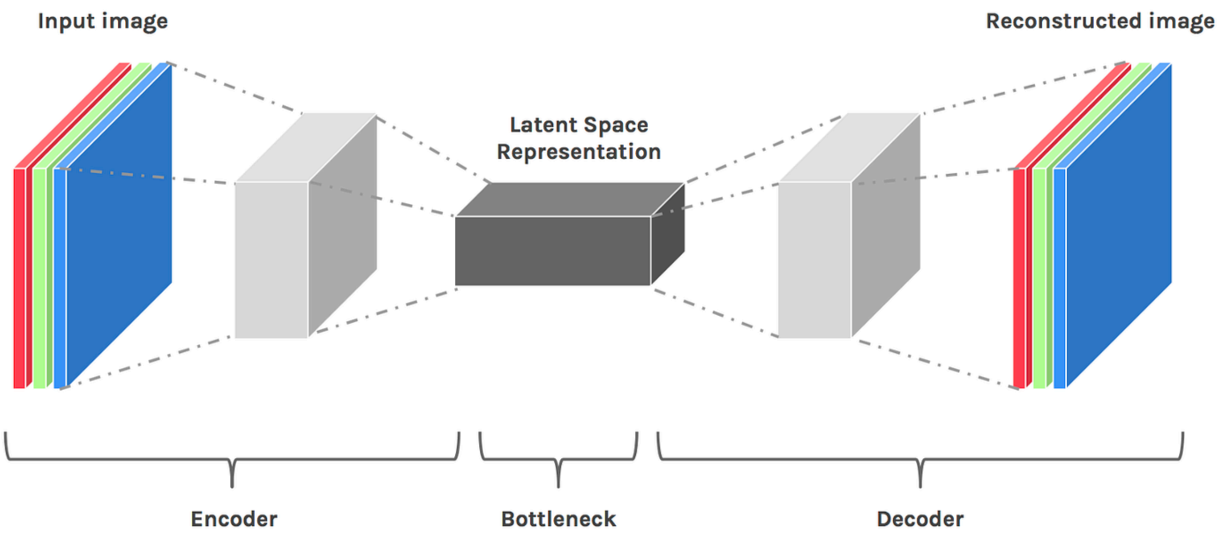


The autoencoder is trained to minimize the difference between the original input and the reconstructed output. This is done using a loss function that measures reconstruction error.

VAE and Latent space

VAEs are a variation of AEs where instead of just replicating the original image they try to create a variation of it as learned by the model in the latent space.

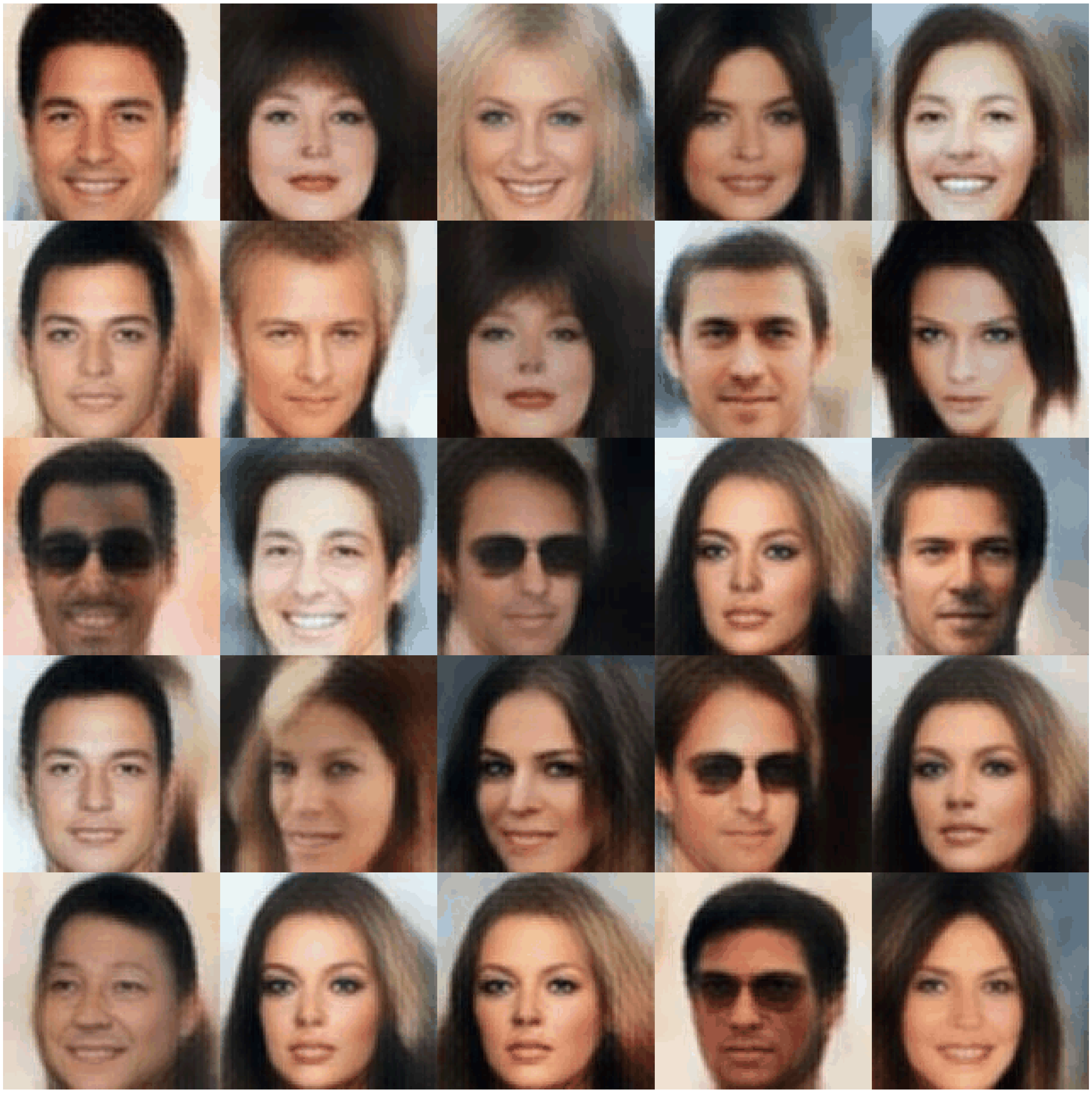
This can be used for improving the image like super resolution or denoising, or make a variant of the image



1. **Imagine the latent space as a "universe of possible images."** The **VAE learns to map real images to regions in this universe**.

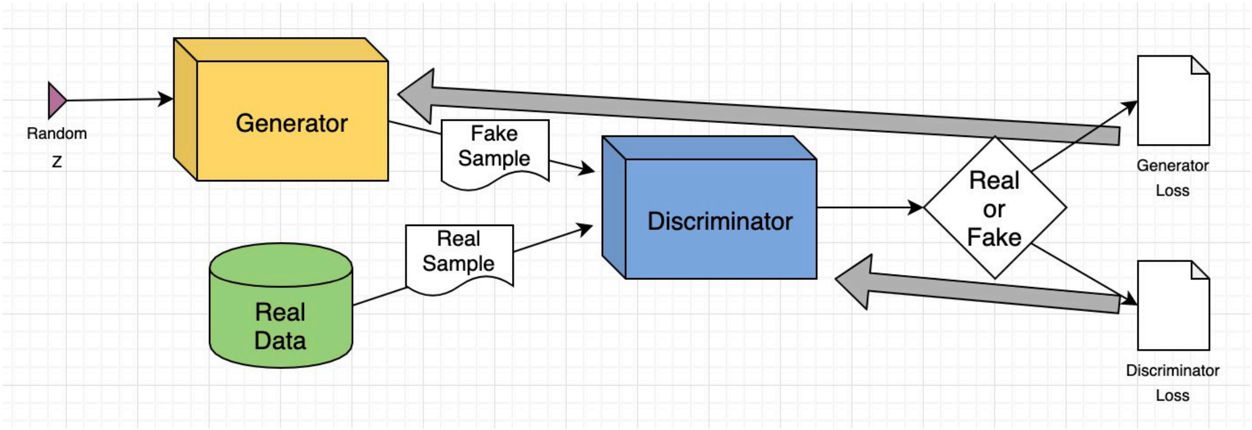
2. When you want to generate a new image, you're essentially saying, "Give me an image from this part of the image universe."

3. The VAE has **learned how to translate coordinates in this universe into actual images**.
4. The encoder maps the input data to a latent space, and a decoder, maps points in the latent space back to the input space. **The goal of training is to learn the encoder and decoder weights such that the decoder can generate new, synthetic data samples that are similar to the original input data.**



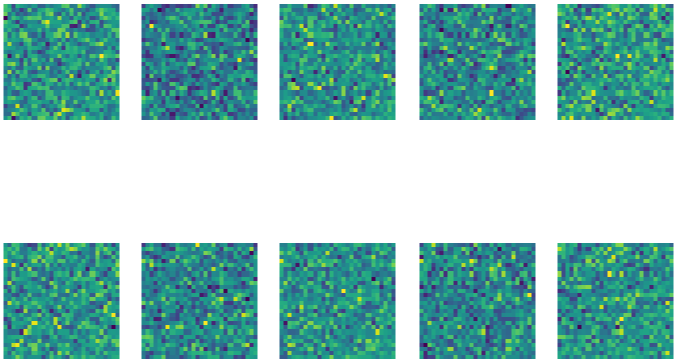
GANs (Generative Adversarial Networks):

Generative Adversarial Networks, commonly called GANs, are a **class of machine learning algorithms that harness the power of two competing neural networks** – the **generator** and the **discriminator**. The term **"adversarial"** arises from the concept that **these networks are pitted against each other in a contest**



GANs architecture. GANs are comprised of **two core components**, known as **sub-models**:

- **Generator:**
    1. This network starts with random noise as input and produces data (like images).
    2. Its goal is to generate data that's as close as possible to real data.
    3. Just like a CNN learns to recognize objects in photos (like cats, dogs, or cars), the generator in a GAN learns to create these objects by understanding what they should look like.
- **Discriminator**:
    1. This network takes real data and the data generated by the Generator as input and attempts to distinguish between the two.
    2. It outputs the probability that the given data is real.
    3. The architecture of the discriminator often mirrors that of traditional convolutional neural networks (CNNs), identifying the patterns, features like normal CNNs.



The adversarial game.

1. The adversarial nature of GANs is derived from a game theory.
2. The generator aims to produce fake samples that are indistinguishable from real data, while the discriminator endeavors to accurately identify whether a sample is real or fake.
3. This ongoing contest ensures that both networks are continually learning and improving.

Whenever the discriminator accurately classifies a sample, it is deemed the winner, and the generator undergoes an update to enhance its performance.

Conversely, if the generator successfully fools the discriminator, it is considered the winner, and the discriminator is updated.

https://reiinakano.com/gan-playground/

https://poloclub.github.io/ganlab/

The process is considered successful when the generator crafts a convincing sample that not only dupes the discriminator but is also difficult for humans to distinguish..

Text to Image

We earlier said “essentially saying, "Give me an image from this part of the image universe."

But how does a model understand what part and what type of images?

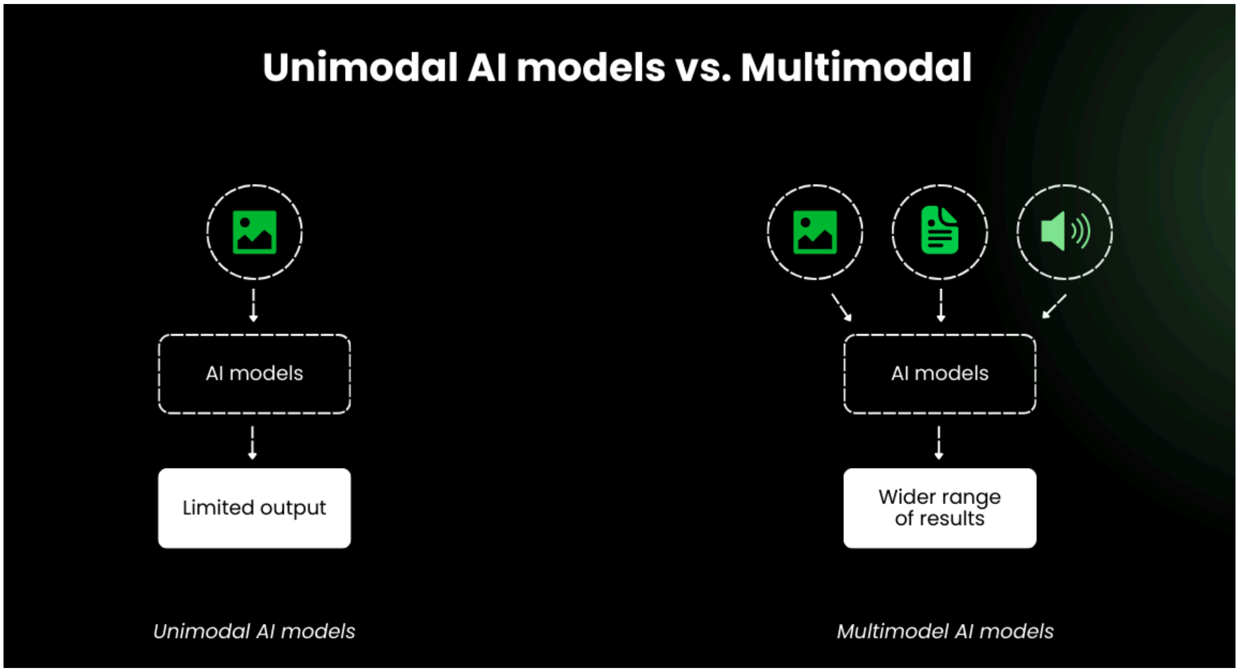
Here comes the concept of text to images.

Text-to-image generation is a powerful application of **combining natural language processing and image generation techniques**.

Multimodal Models

A multimodal model processes and integrates information from multiple types of data (or "modalities"). In the case of text-to-image models, the two modalities are text (language) and images (visual information).

Multimodal models leverage the strengths of different types of data to perform tasks that require understanding and generating information across these modalities. For text-to-image models, this means understanding a textual description and generating a corresponding visual image.



Combining models vs Actual multimodal

- The notion behind this concept is pretty simple; if all models have unique strengths and weaknesses, combining multiple models may help overcome the models' weaknesses, resulting in more robust and accurate predictions.
- the most common techniques used in combining models include:
  - Ensemble models
  - Stacking
  - Bagging
- In multimodal learning, the goal is to combine information from different modalities including text, images, audio, and video.
- By combining these sources of data (modalities), multimodal learning can enable a model to get a more comprehensive understanding of its environment and context since certain cues only exist on specific types of data.

Some Multimodal tasks that we can perform:

- Text to image generation
- image captioning
- Image text retrieval
- Visual Question answering

Key Components of Multimodal Text-to-Image Models

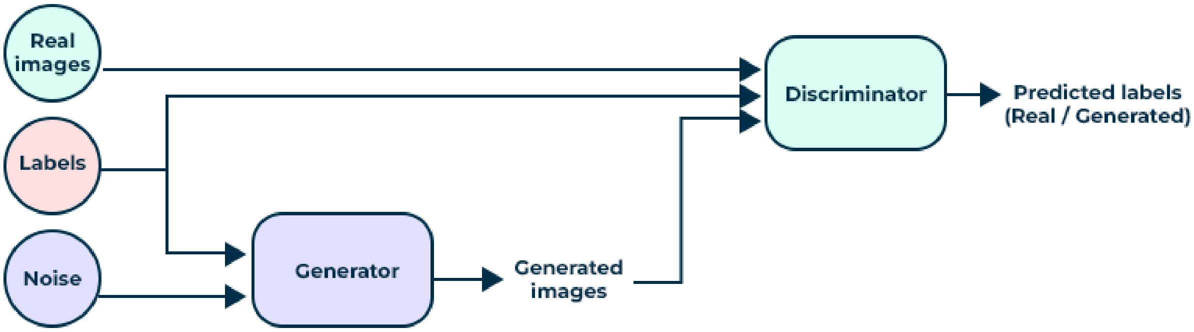
1. **Text Encoder**:

    - Translates the text input into a numerical format that captures its meaning (text embedding).
2. **Image Generator**:
    - Uses the text embedding to generate an image that matches the description.
3. **Training Process**:
    - Trains on pairs of text descriptions and corresponding images to learn how to generate images based on textual descriptions.

Conditional GANs (cGANs)

Conditional GANs (cGANs) are a **variation of Generative Adversarial Networks (GANs)** that incorporate additional information (such as text) to condition the image generation process.

This makes them well-suited for tasks like text-to-image generation. Let's dive into how cGANs work, step by step, combining both text and images.



Conditional GAN

Basic Structure of GANs

1. **Generator**:
    - The generator creates images from random noise.
    - In cGANs, the generator also takes an additional input, which is the conditioning information (text embedding).
    - The generator combines these inputs and processes them through a series of layers (typically convolutional layers) to create an image that matches the text description.
    - Generator's Goal: Create images that can fool the discriminator into thinking they are real and match the text description.
2. **Discriminator**:

    - The discriminator evaluates whether the generated images are real or fake.
    - In cGANs, the discriminator also considers the conditioning information to check if the generated image matches the given text.
    - The discriminator takes the generated image and the text embedding as inputs.
    - It checks two things:
        i. Real vs. Fake: Whether the image looks realistic or not.
        ii. Conditioning: Whether the image matches the text description.
    - Discriminator's Goal: Get better at distinguishing between real and fake images and checking if the images match the text.

VAEs for text to image

1. While training the VAEs for conditional image generation or based on text.
2. The Encoder is trained with an image as well as an associated text, what this does is an amazing thing, creating a latent space of both images and associated text.

Joint Latent Space for Images and Text

1. In a text-to-image VAE, both images and their associated textual descriptions are encoded into a shared latent space.
2. This joint latent space allows the model to understand and generate images based on text. Here's how it works:

Encoding Text and Images:

- The training dataset consists of pairs of images and their associated text descriptions.
- **Text Encoder**: Converts text descriptions into numerical representations called text embeddings.
- **Image Encoder**: Converts images into numerical representations called image embeddings.
- Both encoders map their respective inputs into the same latent space.

Creating New images

- When we want it to create new images we provide the prompt to it.
- The text input helps to locate a point in this latent space
- This point is then decoded into an image as normal by the Decoder.

Text Understanding:

- The first step is to understand the text. This involves using a text encoder, often based on models like Transformers (e.g., BERT or GPT). These models are trained to understand language by processing text and converting it into a sequence of vectors (numbers) that capture the meaning of the words.
    - The text is broken down into tokens (words or subwords).
    - These tokens are then converted into numerical representations called embeddings.
    - Advanced models like BERT or GPT are often used to capture the context and meaning of the words.
    - This transformed representation acts like a set of instructions for the image generator.
- **Example**: The sentence "a red apple on a table" is transformed into a series of vectors that represent the key features of this sentence, such as "red," "apple," and "table."

Combining Text and Image Generation:

- Once we have a vector representation of the text, we need to incorporate this information into our image generation process. This is where models like GANs or VAEs come into play.
- The image generation model now receives this text embedding, which acts as a guide for the image generation model. Think of this as giving an artist a detailed description of what you want them to paint.
- For GAN models:
    - The model takes some random noise, which helps in creating varied and unique images each time, even for the same description.
    - The generator uses the text embedding to influence how it creates the image. It combines the meaning captured in the text embedding with the random noise to produce an initial version of the image.
    - This process is repeated many times, with the generator improving its output based on feedback from the discriminator.
- For VAE Models
    - They learn a joint latent space that represents both text and images.
    - The text input helps to locate a point in this space.
    - This point is then decoded into an image.

Imaginative AI: Understanding Visual Creativity.ipynb