Artificial Intelligence (Machine Learning & Deep Learning) [Course]

Week 9 - Generative AI & LLMs

[See examples / code in GitHub code repository]

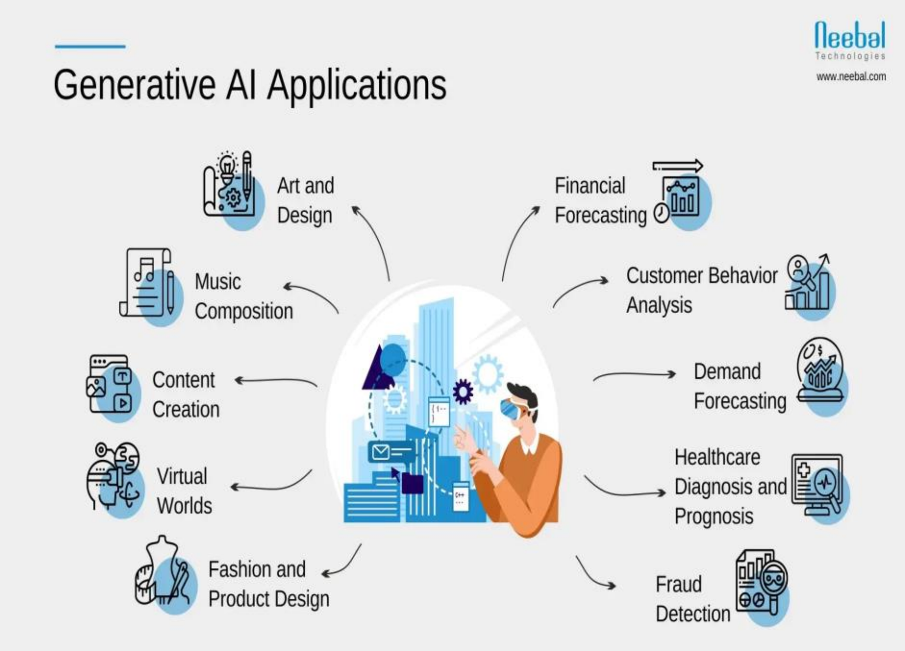
It is not about Theory, it is 20% Theory and 80% Practical – Technical/Development/Programming [Mostly Python based]

Gen AI | What is a Generative AI for Developers?

Generative AI is a branch of artificial intelligence that focuses on creating new content, such as text, images, videos, audio, or even code, based on patterns learned from existing data. Unlike traditional AI, which primarily analyzes or classifies data, generative AI produces original outputs, making it a powerful tool for developers across various domains.

How Generative AI Works

Generative AI models are typically built using deep learning techniques, such as neural networks, and are trained on massive datasets. These models identify patterns and relationships in the data and use this knowledge to generate new content. Popular architectures include **Generative Adversarial Networks (GANs)**, **Transformers**, and **Variational Autoencoders (VAEs)**. For example, GANs use two networks—a generator and a discriminator—to create realistic outputs like images or videos.



Reference:

https://aws.amazon.com/ai/generative-ai/
https://www.ibm.com/think/topics/generative-ai
https://www.geeksforgeeks.org/artificial-intelligence/what-is-generative-ai/

Gen AI | Large Language Models (LLMs)?

Large Language Models (LLMs) are specialized AI designed to understand and generate human language with remarkable proficiency[1]. They focus specifically on processing and producing text, making them integral to tasks like translation, summarization, and conversational AI. Key aspects of LLMs include:

Natural Language Processing (NLP): LLMs leverage NLP to understand and interpret human language[1].

Tokenization: This involves breaking down a sentence into smaller units called tokens, which serve as the building blocks for the model's understanding[1].

Transformers: LLMs typically use transformer architecture to analyze relationships between words and their contexts in massive datasets[1].

Examples of LLMs

Text Generation: GPT-3 and GPT-4 are well-known LLMs capable of generating human-like texts[1].

Enhanced Text Understanding: Gemini by Google focuses on enhancing text generation and understanding[1].

Reference:
https://www.geeksforgeeks.org/artificial-intelligence/llms-vs-generative-ai-dissecting-the-spectrum-of-ai-powered-creativity/

GenAI | Prompt Engineering?

Large Language Models (LLMs) are specialized AI designed to understand and generate human language with remarkable proficiency[1]. They focus specifically on processing and producing text, making them integral to tasks like translation, summarization, and conversational AI. Key aspects of LLMs include:

Natural Language Processing (NLP): LLMs leverage NLP to understand and interpret human language[1].

Tokenization: This involves breaking down a sentence into smaller units called tokens, which serve as the building blocks for the model's understanding[1].

Transformers: LLMs typically use transformer architecture to analyze relationships between words and their contexts in massive datasets[1].

Examples of LLMs

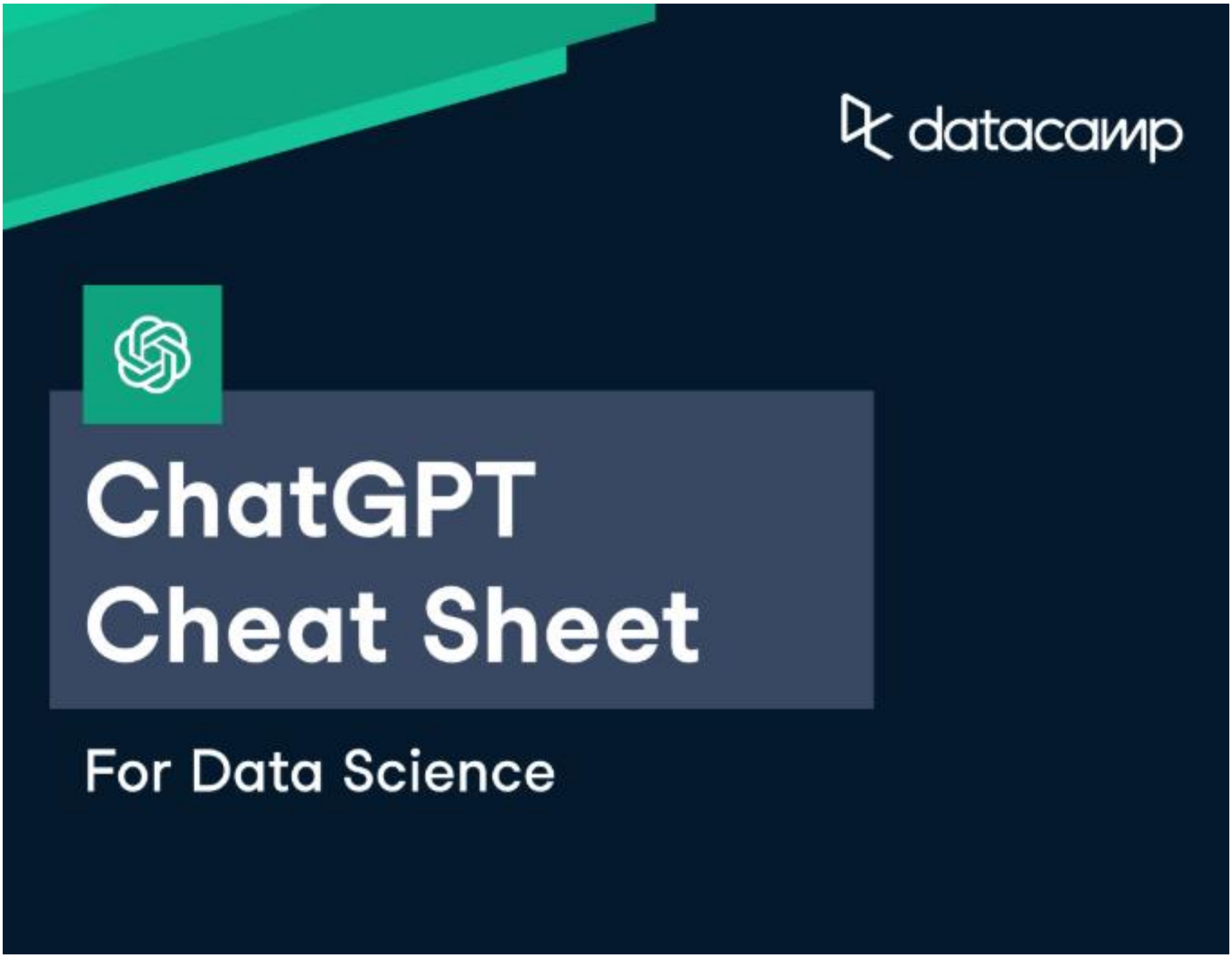
Text Generation: GPT-3 and GPT-4 are well-known LLMs capable of generating human-like texts[1].

Enhanced Text Understanding: Gemini by Google focuses on enhancing text generation and understanding[1].

Reference:

https://www.datacamp.com/blog/what-is-prompt-engineering-the-future-of-ai-communication
https://huggingface.co/docs/transformers/en/tasks/prompting
https://www.ibm.com/think/topics/prompt-engineering

GenAI |ChatGPT Cheat Sheet for software professional



datacamp

ChatGPT Cheat Sheet

For Data Science

Reference:
https://www.datacamp.com/cheat-sheet/chatgpt-cheat-sheet-data-science
https://www.datacamp.com/tutorial/a-beginners-guide-to-chatgpt-prompts-for-marketing

GenAI | GANs and VAEs

What are GANs?

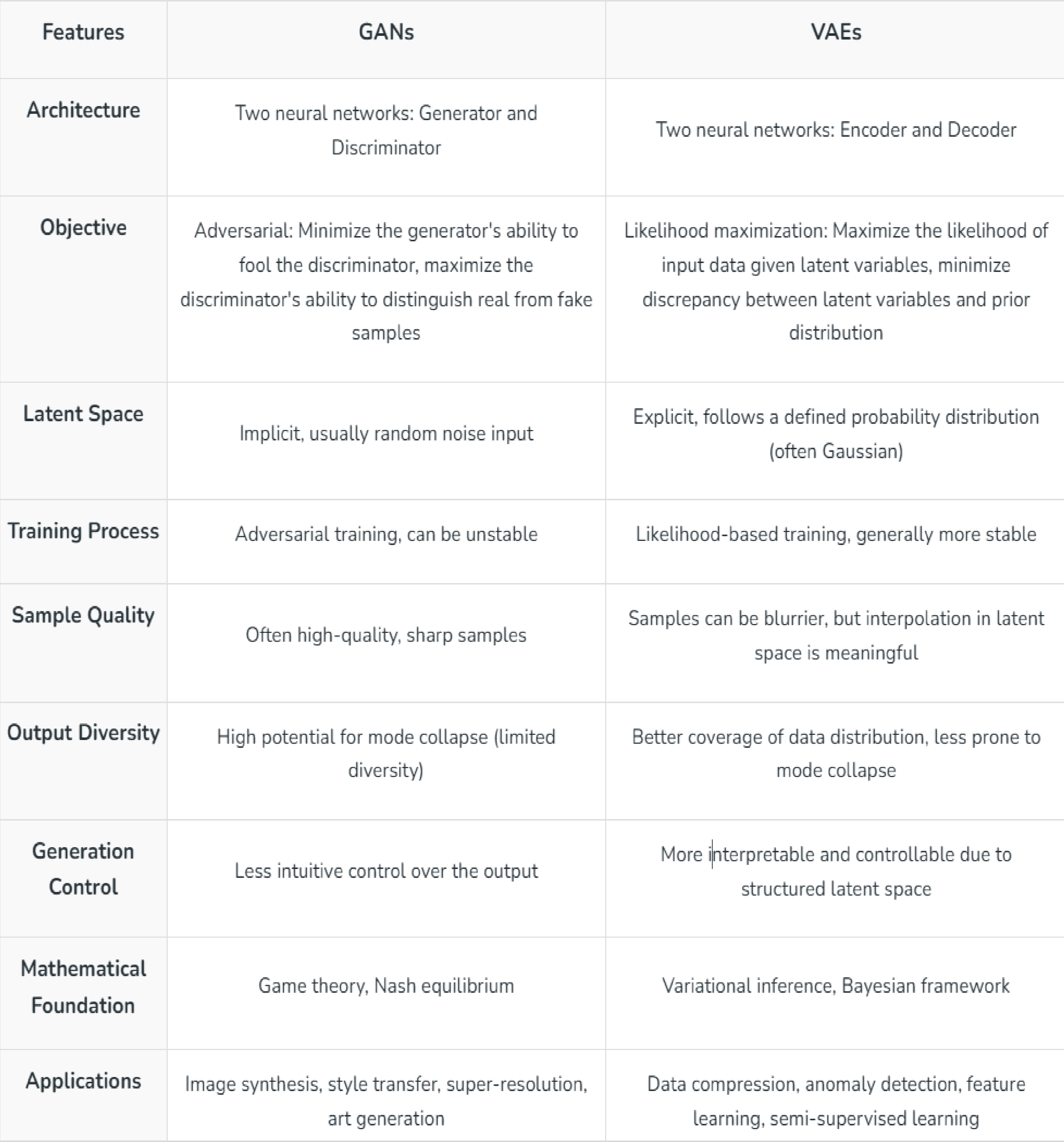
GAN is a type of machine learning model called a neural network, specially designed to imitate the structure and function of a human brain. For this reason, neural networks in machine learning are sometimes referred to as artificial neural networks (ANNs). The generator's goal is to produce data points that are indistinguishable from real data, while the discriminator's goal is to correctly identify generated data points. The training process of GANs involves a two-player game between the generator and discriminator. The generator tries to produce realistic data points, while the discriminator tries to correctly classify them as real or fake.

What are VAEs?

A variational autoencoder (VAE) is an artificial neural network architecture introduced by Diederik P. Kingma and Max Welling. It is part of the families of probabilistic graphical models and variational Bayesian methods. VAEs are generative models explicitly designed to capture the underlying probability distribution of a given dataset and generate novel samples. They utilize an architecture that comprises an encoder-decoder structure. A VAE provides a probabilistic manner for describing an observation in latent space.

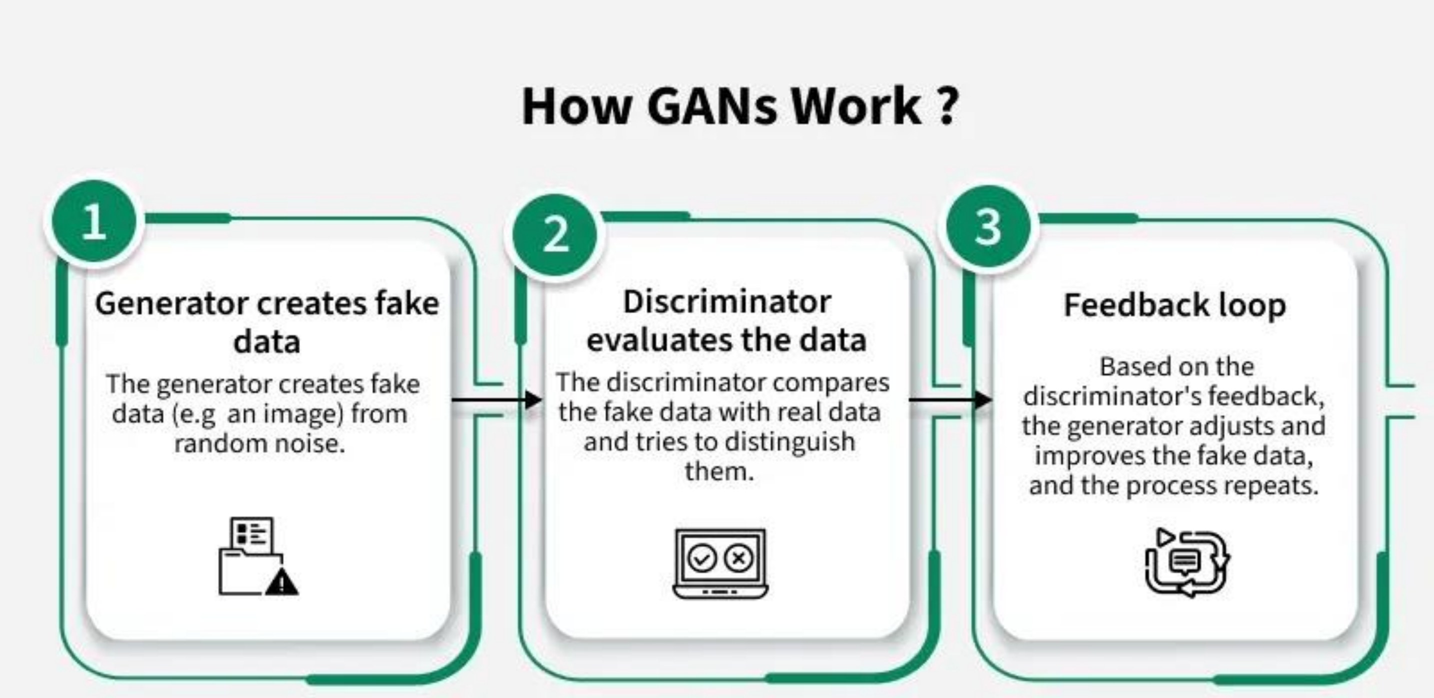
Reference:
https://www.geeksforgeeks.org/deep-learning/generative-models-in-ai-a-comprehensive-comparison-of-gans-and-vaes

GenAI | Comparison - GANs and VAEs



Features	GANs	VAEs
Architecture	Two neural networks: Generator and Discriminator	Two neural networks: Encoder and Decoder
Objective	Adversarial: Minimize the generator's ability to fool the discriminator, maximize the discriminator's ability to distinguish real from fake samples	Likelihood maximization: Maximize the likelihood of input data given latent variables, minimize discrepancy between latent variables and prior distribution
Latent Space	Implicit, usually random noise input	Explicit, follows a defined probability distribution (often Gaussian)
Training Process	Adversarial training, can be unstable	Likelihood-based training, generally more stable
Sample Quality	Often high-quality, sharp samples	Samples can be blurrier, but interpolation in latent space is meaningful
Output Diversity	High potential for mode collapse (limited diversity)	Better coverage of data distribution, less prone to mode collapse
Generation Control	Less intuitive control over the output	More interpretable and controllable due to structured latent space
Mathematical Foundation	Game theory, Nash equilibrium	Variational inference, Bayesian framework
Applications	Image synthesis, style transfer, super-resolution, art generation	Data compression, anomaly detection, feature learning, semi-supervised learning

GenAI | Image Generation using Generative Adversarial Networks (GANs) using TensorFlow



How GANs Work ?

1 **Generator creates fake data**

The generator creates fake data (e.g an image) from random noise.

2 **Discriminator evaluates the data**

The discriminator compares the fake data with real data and tries to distinguish them.

3 **Feedback loop**

Based on the discriminator's feedback, the generator adjusts and improves the fake data, and the process repeats.

Code Sample:
https://github.com/ShahzadSarwar10/FULLSTACK-WITH-AI-BOOTCAMP-B1-MonToFri-2.5Month-Explorer/blob/main/Week9/Case-9-1-DCGAN-With-Tensorflow.py

Reference:
https://www.tensorflow.org/tutorials/generative/dcgan
https://www.geeksforgeeks.org/deep-learning/generative-adversarial-network-gan/

Thank you - for listening and participating

□Questions / Queries

□Suggestions/Recommendation

□Ideas…..?

Shahzad Sarwar
Cognitive Convergence
https://cognitiveconvergence.com
shahzad@cognitiveconvergence.com
voice: +1 4242530744 (USA) +92-3004762901 (Pak)