



Generative AI Bootcamp

Language / Text Generative Models

Transformers

Transformers are a type of neural network that have a **unique ability to recognize long-range connections within sequences**. They are particularly useful for **tasks like generating text**, as the model needs to comprehend the preceding words in order to produce the next one.

The introduction of transformers in 2018 was a groundbreaking moment for the field of natural language processing.



Seminal paper published in 2017

Transformers – key capabilities

Transformers
are able to
learn long-
range
dependencies
in sequences.

Transformers
are able to be
trained on very
large datasets.

Transformers
are able to be
parallelized

Transformer: Attention Mechanism



Consider two sentences:

- The **cat** drank the milk because **it** was hungry.
- The cat drank the **milk** because **it** was sweet.

In the first sentence, the word 'it' refers to 'cat'
in the second it refers to 'milk'.

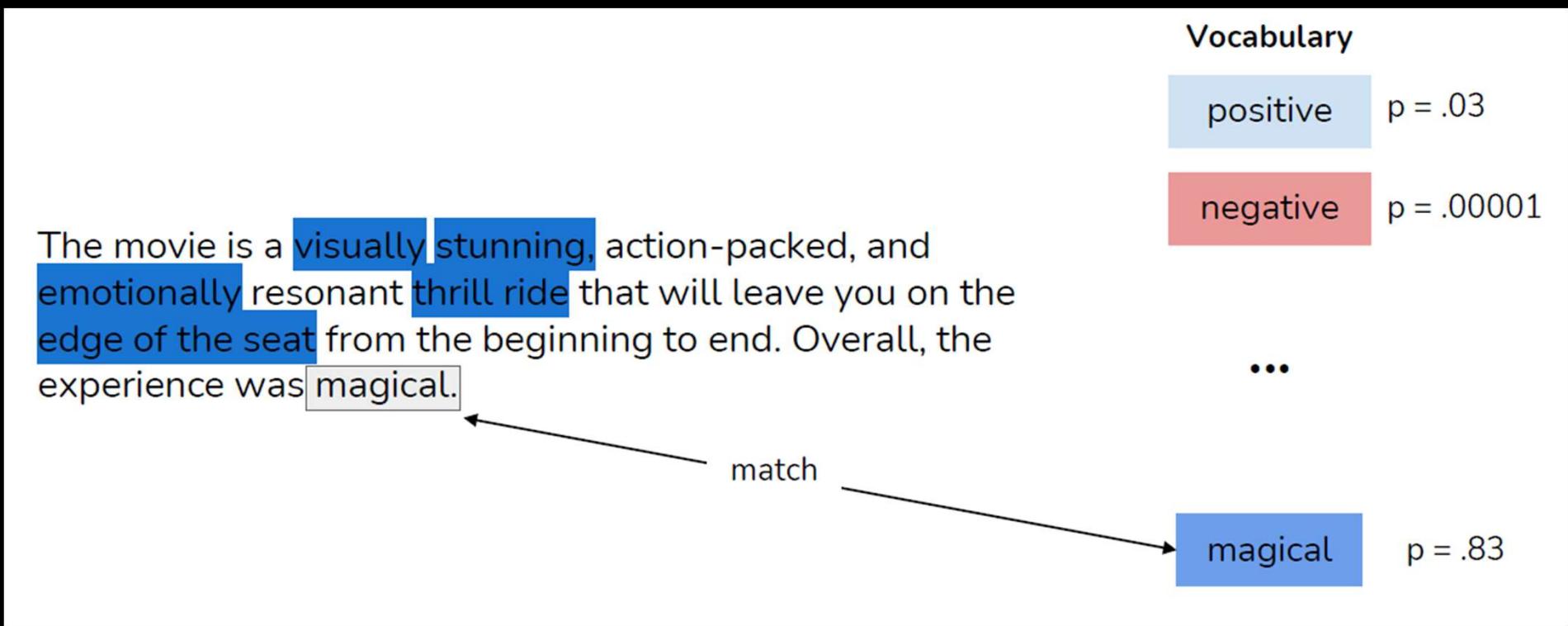
When the model processes the word 'it', **self-attention** gives the model more information about its meaning so that it can associate 'it' with the correct word.

	The	The
The	cat	cat
cat	drank	drank
drank	the	the
the	milk	milk
milk	because	because
because	it	it
it	was	was
was	hungry	hungry
hungry		

Input Score 1 Score 2

The diagram shows attention weights between words in two parallel sentences. Red arrows point from the word 'it' in the first sentence to 'cat' and 'milk' in the second sentence, indicating that the model is attending to both words simultaneously during processing.

Transformer: Next token prediction



Transformer: **Generate** text with next token prediction

During inference, the LLM predicts the next word in the input sequence.

Input word = prompt

The

Output, word-by-word

The []

The movie []

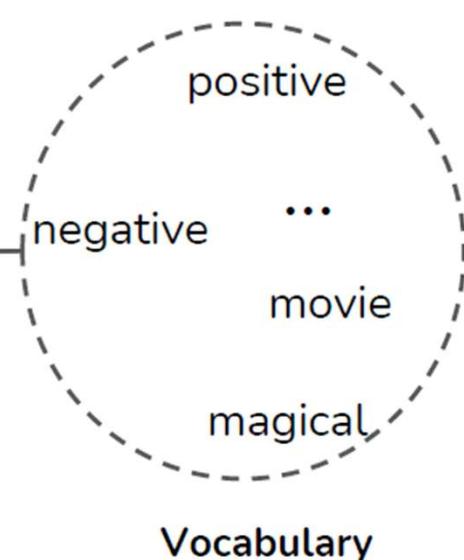
The movie was []

The movie was awesome. []

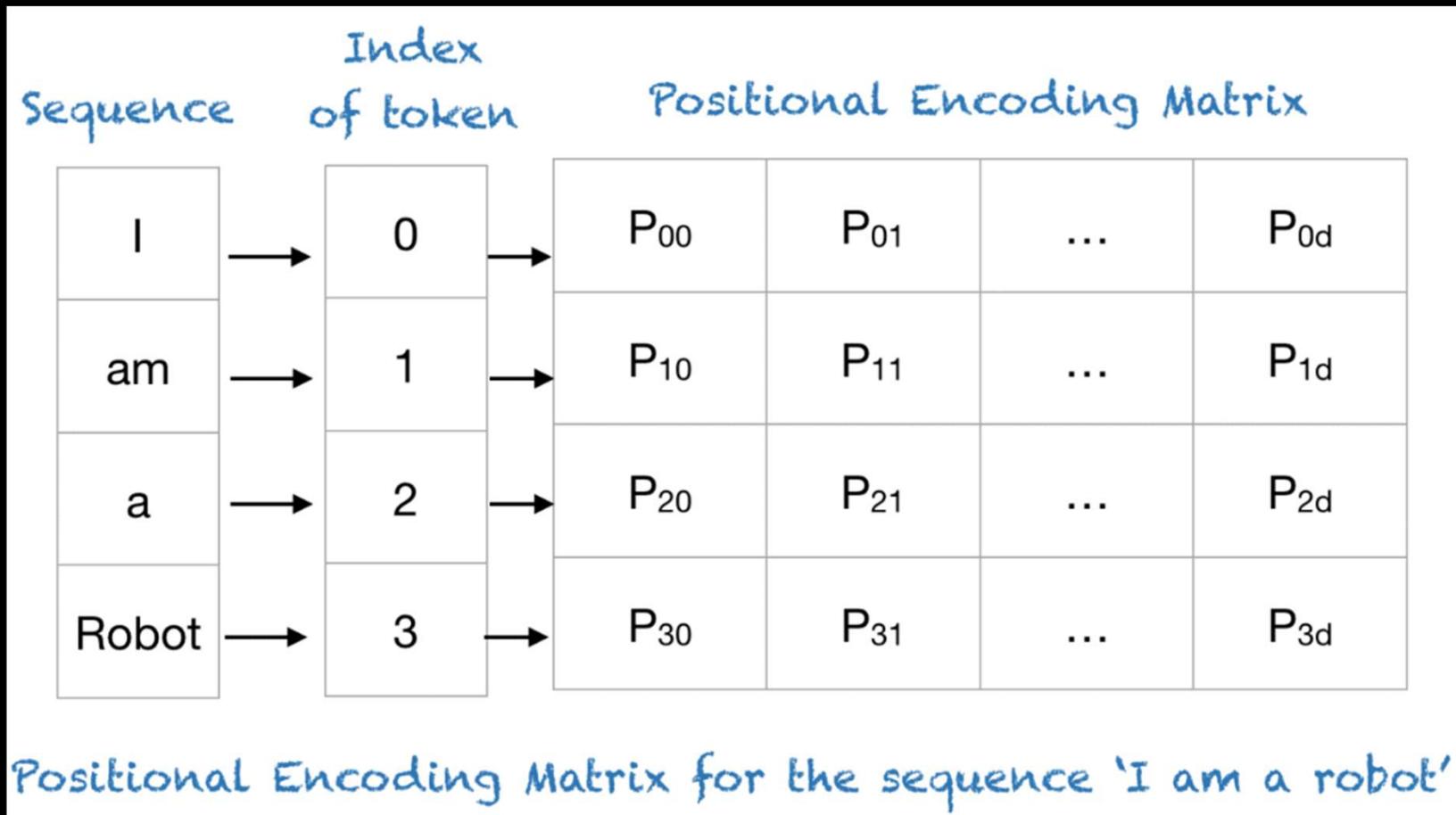
The movie was awesome. Overall, []

The movie was awesome. Overall, the []

:



Transformer: Positional encoding



Transformer: Positional encoding – what have we achieved?

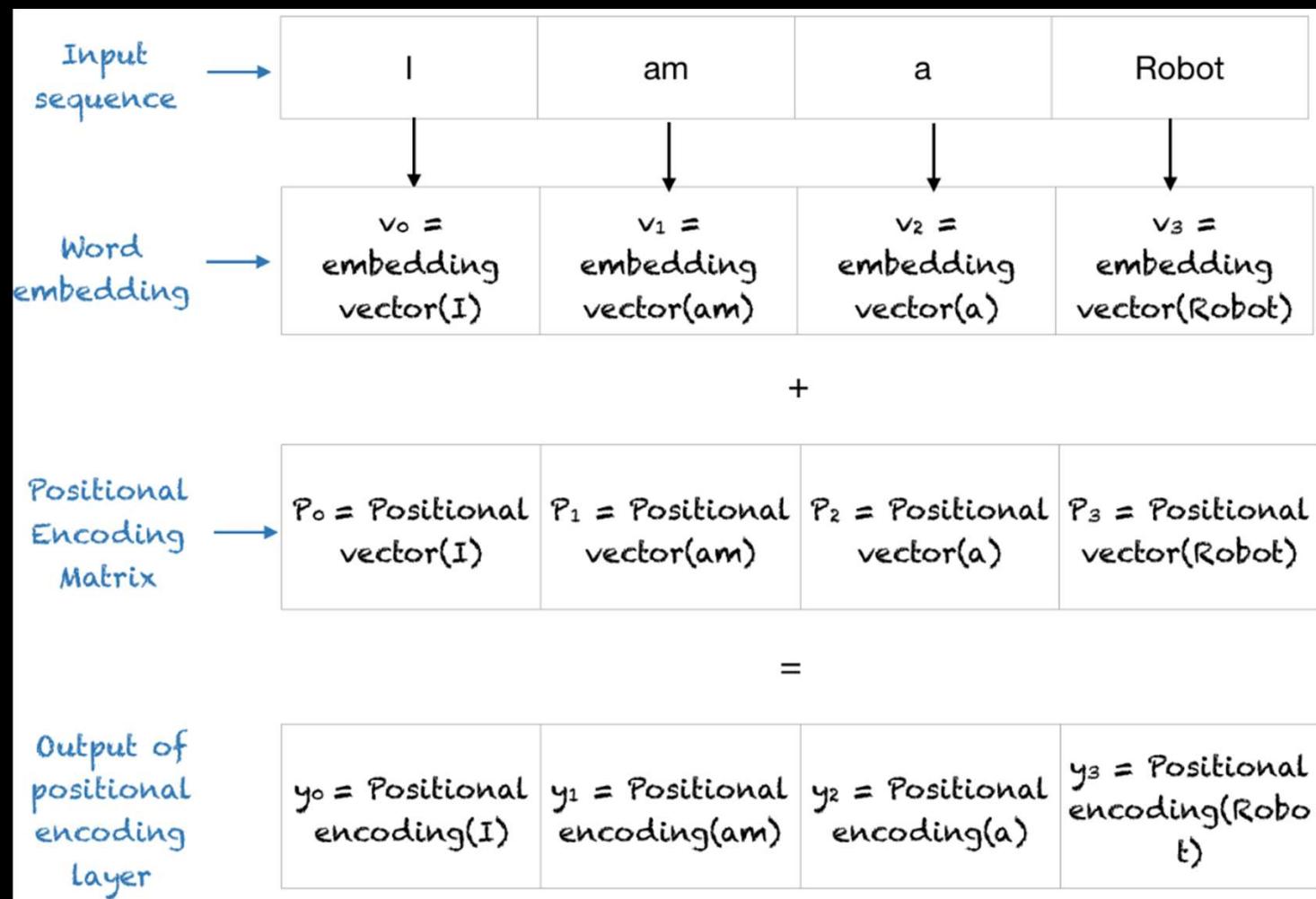
- The sine and cosine functions have values in [-1, 1], which keeps the values of the positional encoding matrix in a normalized range.
- As the sinusoid for each position is different, you have a unique way of encoding each position.
- We have a way of measuring or quantifying the similarity between different positions, hence enabling you to encode the relative positions of words.

Transformer: Positional encoding contd.

Sequence	Index of token, k	Positional Encoding			
		$i=0$	$i=0$	$i=1$	$i=1$
I	0	$P_{00} = \sin(0) = 0$	$P_{01} = \cos(0) = 1$	$P_{02} = \sin(0) = 0$	$P_{03} = \cos(0) = 1$
am	1	$P_{10} = \sin(1/1) = 0.84$	$P_{11} = \cos(1/1) = 0.54$	$P_{12} = \sin(1/10) = 0.10$	$P_{13} = \cos(1/10) = 1.0$
a	2	$P_{20} = \sin(2/1) = 0.91$	$P_{21} = \cos(2/1) = -0.42$	$P_{22} = \sin(2/10) = 0.20$	$P_{23} = \cos(2/10) = 0.98$
Robot	3	$P_{30} = \sin(3/1) = 0.14$	$P_{31} = \cos(3/1) = -0.99$	$P_{32} = \sin(3/10) = 0.30$	$P_{33} = \cos(3/10) = 0.96$

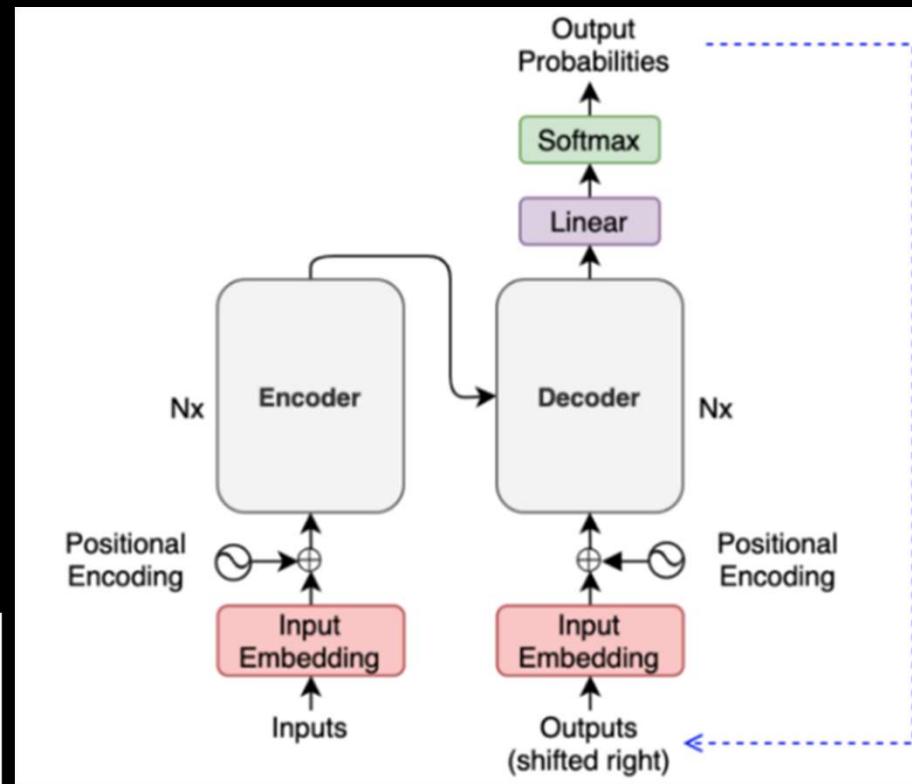
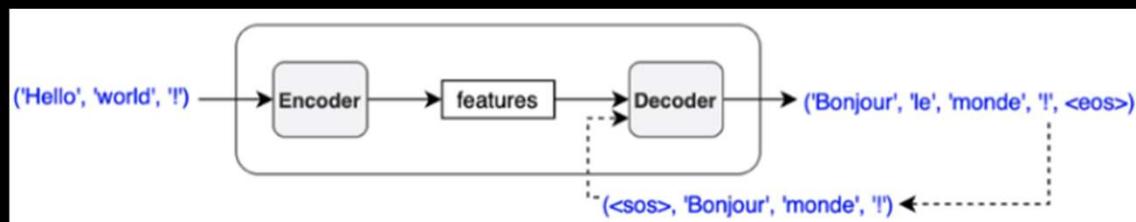
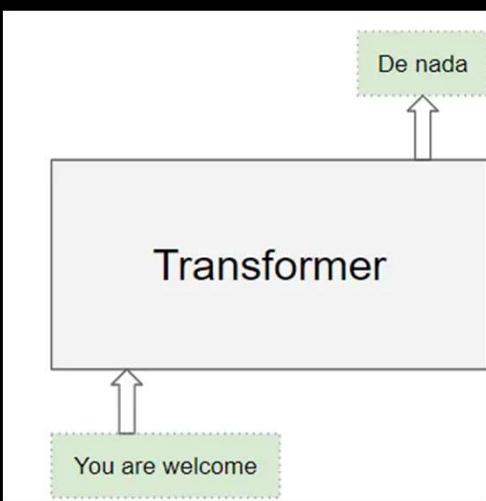
Positional Encoding Matrix for the sequence 'I am a robot'

Transformer: Positional encoding contd.



Encoders and Decoders

The **encoder** extracts features from an input sentence, and the **decoder** uses the features to produce an output sentence



Popular Language models built on Transformer Architecture

GPT

Generative Pre-trained Transformer

Decoder only

Open AI. GPT-1 released in Jun 2018 (GPT 3.5 in 2022 Nov)

BERT

Bidirectional Encoder Representations from Transformers)

Encoder only, not an LLM (100M odd parameters)

Google. Oct 2018

T5

Encoder and decoder

Google.

Generative Text Models

- **Transformers** dominate this space now
- RNN, LSTM, GRU still used depending on use-cases
- VAE and GAN have also been adopted (TextGAN, SeqGAN)
- Advanced architectures such as Memory Augmented Networks, Graph Neural Networks are being researched

Generative Image Architectures

- Generative Adversarial Network (GAN)

Invented: 2014 **By:** Ian Goodfellow et al

Paper: "Generative Adversarial Networks"

- Variational Auto Encoder (VAE)

Invented: 2013 **By:** Kingma, Diederik P., and Max Welling

Paper: "Auto-Encoding Variational Bayes"

- Stable Diffusion

Invented: 2022 **By:** Stability AI – led by Emad Mostaque

Paper: "High-Resolution Image Synthesis with Latent Diffusion Models"

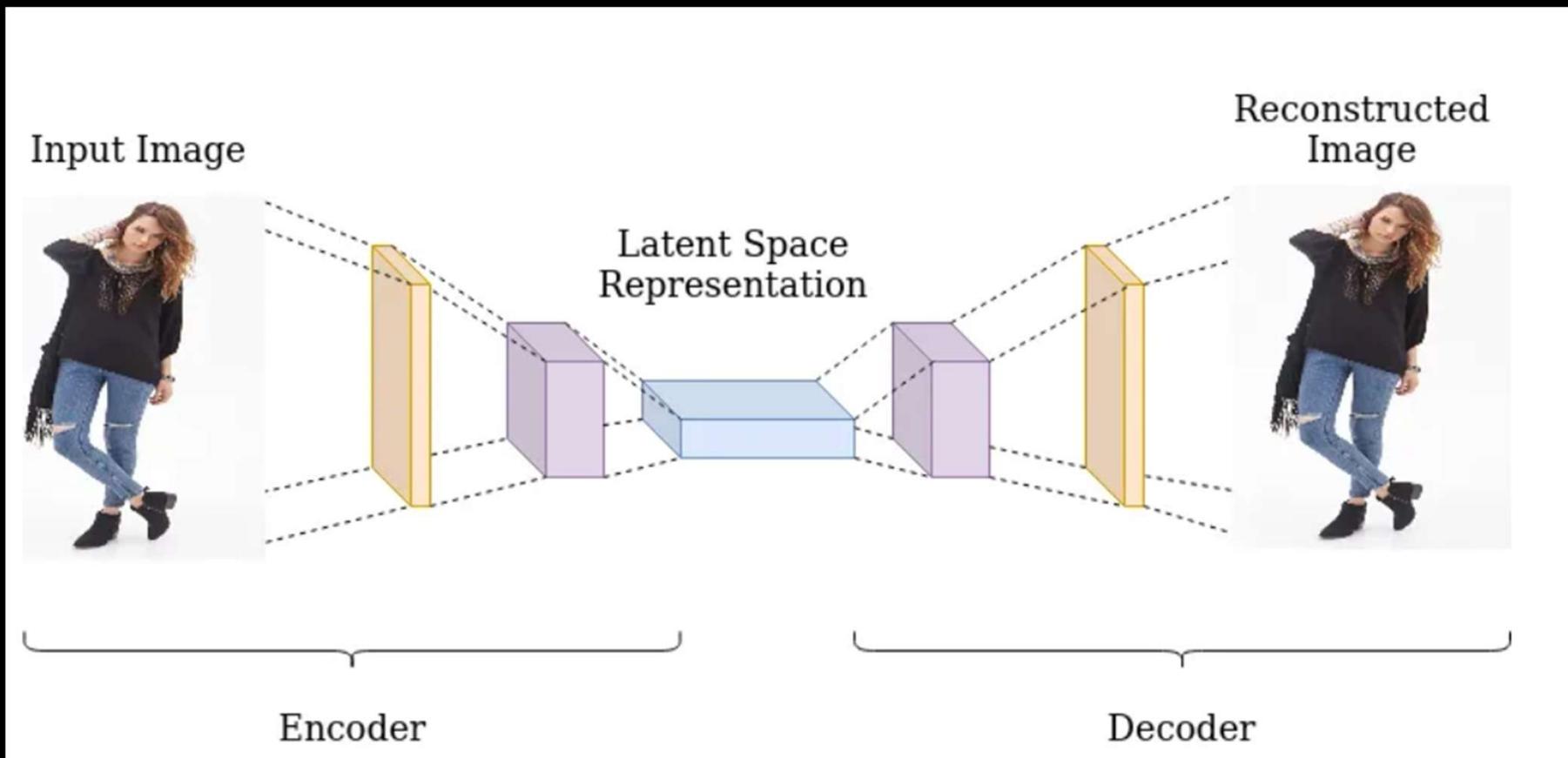
- Vision Transformers (ViT)

- DALL-E

Popular image generation models

- DALL-E by Open AI
- Midjourney by Midjourney
initial version: 2022, v6: Dec 2023
- Stable Diffusion XL
Opensource
- Imagen
Google
- Firefly
By Adobe

Encoders and decoders



Encoding techniques: PCA and Autoencoders

Feature	PCA (Principal Component Analysis)	Autoencoders
Supervision	Linear transformation technique Unsupervised (no labels required)	Non-linear neural network-based encoding Unsupervised (self-supervised learning)
Feature Extraction	Identifies linear correlations between features	Learns complex, non-linear feature representations
Dimensionality Reduction	Reduces dimensions by projecting data onto principal components	Compresses data using a neural network bottleneck
Computational Complexity	Low (based on eigenvalue decomposition/SVD)	High (requires training a neural network)
Interpretability	Easy to interpret (principal components are linear combinations of features)	Hard to interpret (learned latent space is abstract)
Non-linearity Handling	✗ Cannot capture non-linear relationships	✓ Captures complex, non-linear patterns
Scalability	Efficient for small to medium datasets	Scales well but requires more computational power
Data Reconstruction	Approximate reconstruction (lossy) using principal components	Can reconstruct input but may not be perfect (depends on loss function)
Flexibility	Fixed transformation (does not adapt to new data)	Can be trained on specific datasets and fine-tuned
Robustness to Noise	Sensitive to noise in data	Denoising autoencoders can effectively remove noise

Encoding and Decoding example

Example: Image Compression

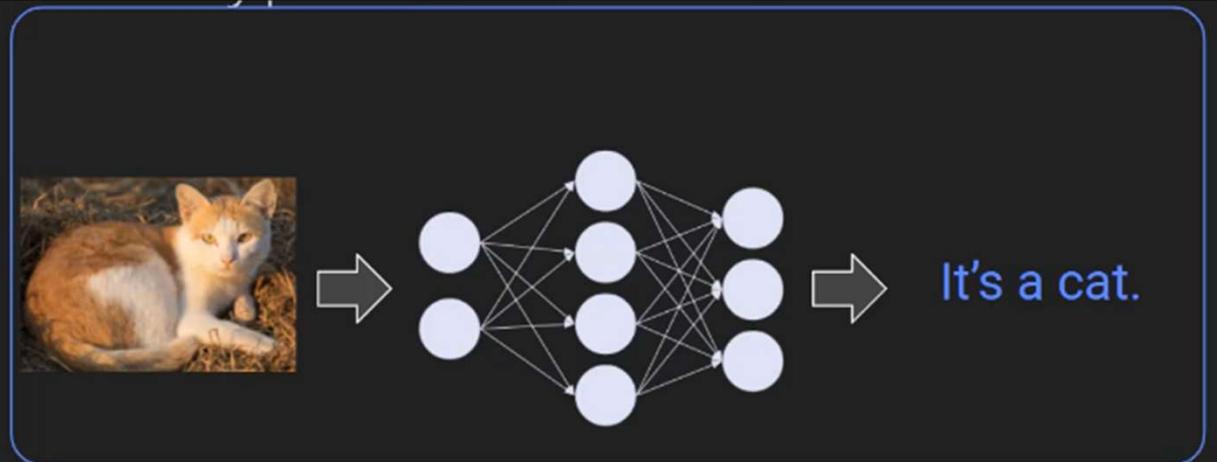
Consider a high-resolution image (128×128 pixels).

Instead of storing all 16,384 pixels (128×128), an autoencoder might encode it into a vector of just 100 numbers.

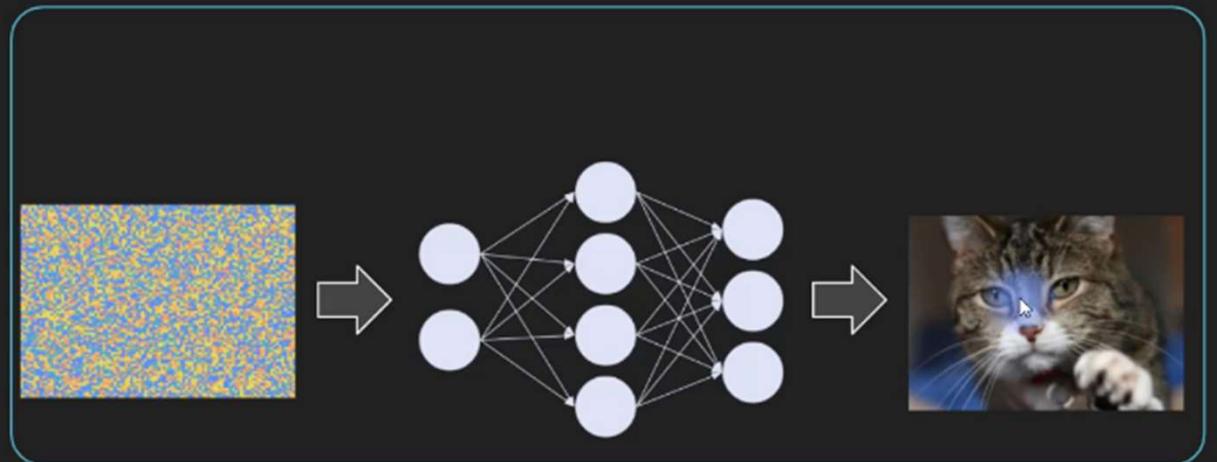
The decoder then tries to reconstruct the original image from just those 100 numbers.

Discriminative and Generative Models

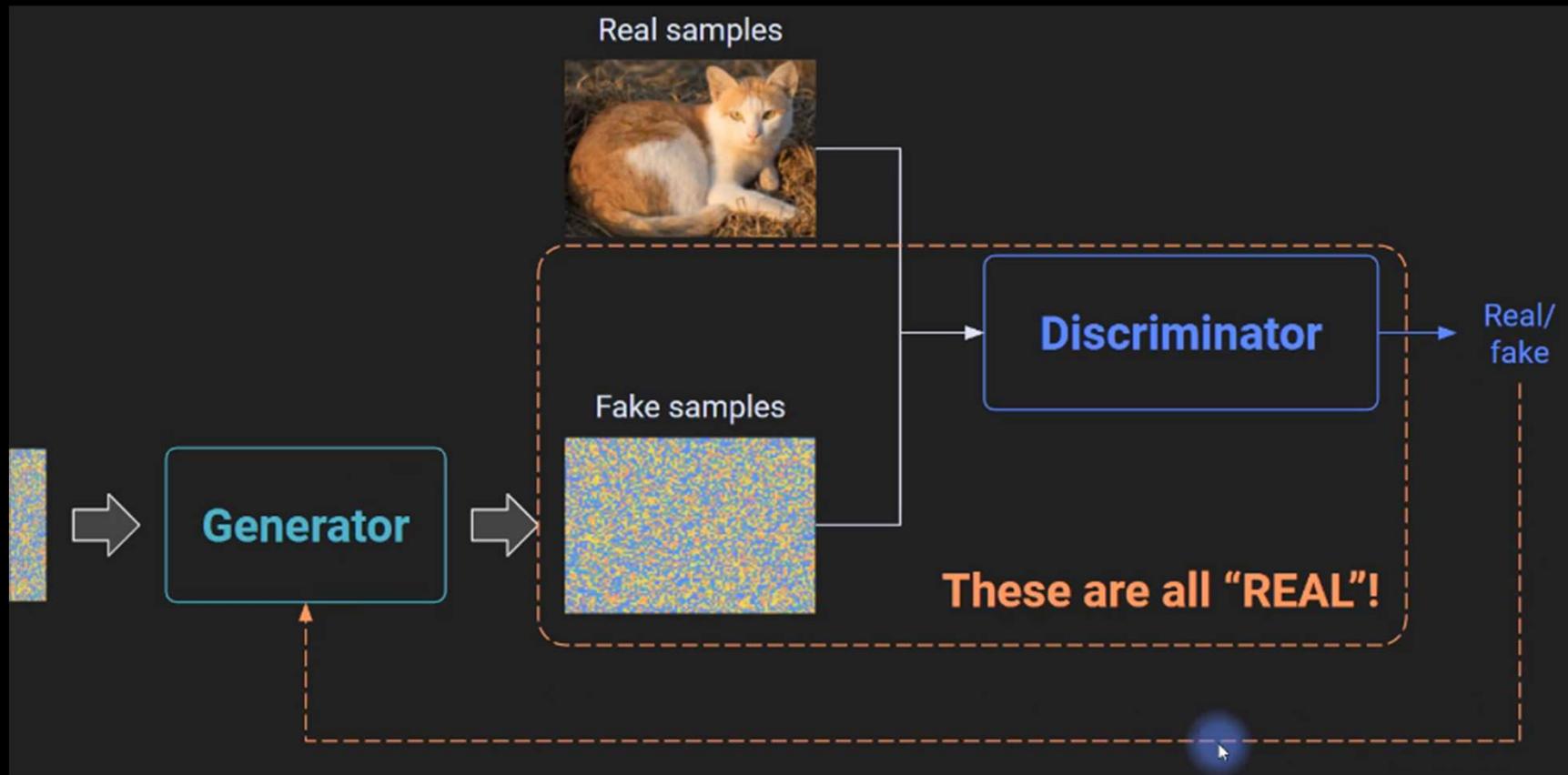
Discriminative models
classify or characterize
existing data.



Generative models
create new samples.



Generative Adversarial Network (GAN)

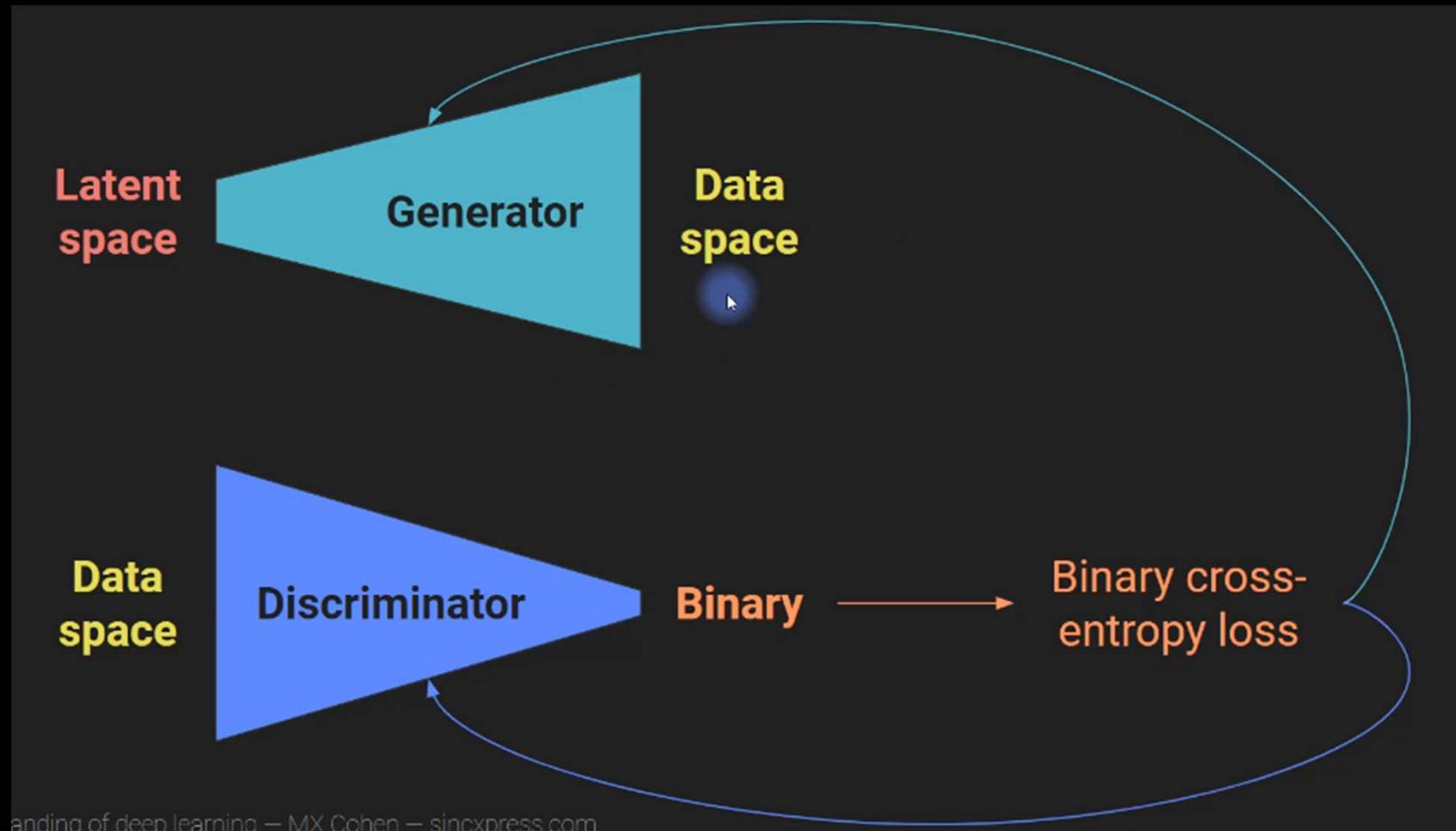


Adversarial Network / training

Discriminator model training (we give the model labelled data – real images)

Generator model starts from noise and corrects itself by backpropagation to come close to real sample

Generative Adversarial Network (GAN)



Applications of GAN

Combine two categories to increase training set.

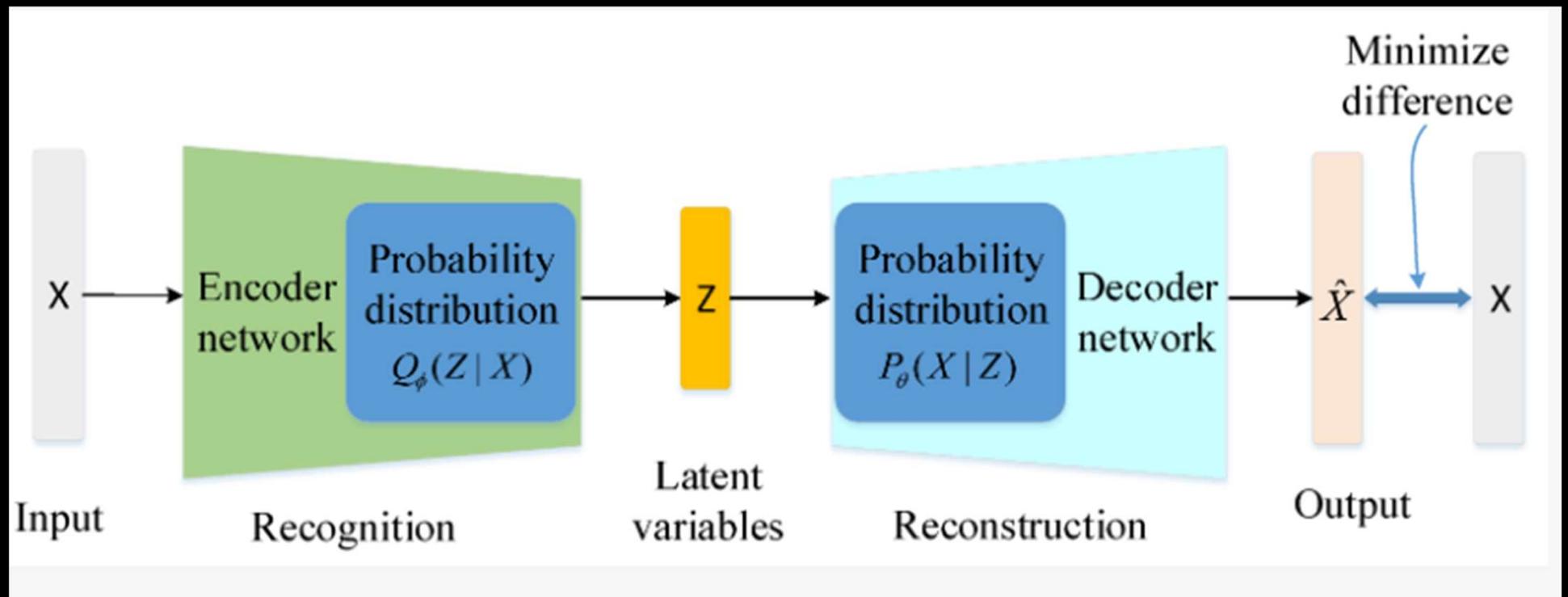
Super-resolution upscaling.

Personalized medical devices.

Evil (e.g., deep fakes).

Learning from private data without privacy concerns
(learn from data, then delete original data).

Variational Auto Encoder (VAE)



GAN Vs VAE

Feature	VAE (Variational Autoencoder)	GAN (Generative Adversarial Network)
Architecture	Encoder-Decoder	Generator-Discriminator (Adversarial)
Learning Approach	Probabilistic (learns latent space distribution)	Game-theoretic (two networks compete)
Latent Space Representation	Continuous, structured, and smooth	Unstructured, often chaotic
Training Objective	Maximizes log-likelihood with KL-divergence regularization	Minimax game (Generator vs. Discriminator)
Optimization	Uses reconstruction loss + KL divergence	Uses adversarial loss (binary cross-entropy)
Sample Quality	Lower quality, but smooth interpolation	High quality, but may have mode collapse
Stability of Training	More stable, but blurry images	Less stable, often difficult to train
Mode Collapse Issue	✗ No mode collapse	✓ Can suffer from mode collapse (generator produces limited variety)

VAE Demo and assignment

1. Input Image:

(28, 28, 1) (grayscale image)

2. Encoder Output:

- Mean (μ): [1.0, -0.5]
- Log Variance ($\log(\sigma^2)$): [0.1, 0.2]

3. Latent Vector (Sampled): Stochastic encoding

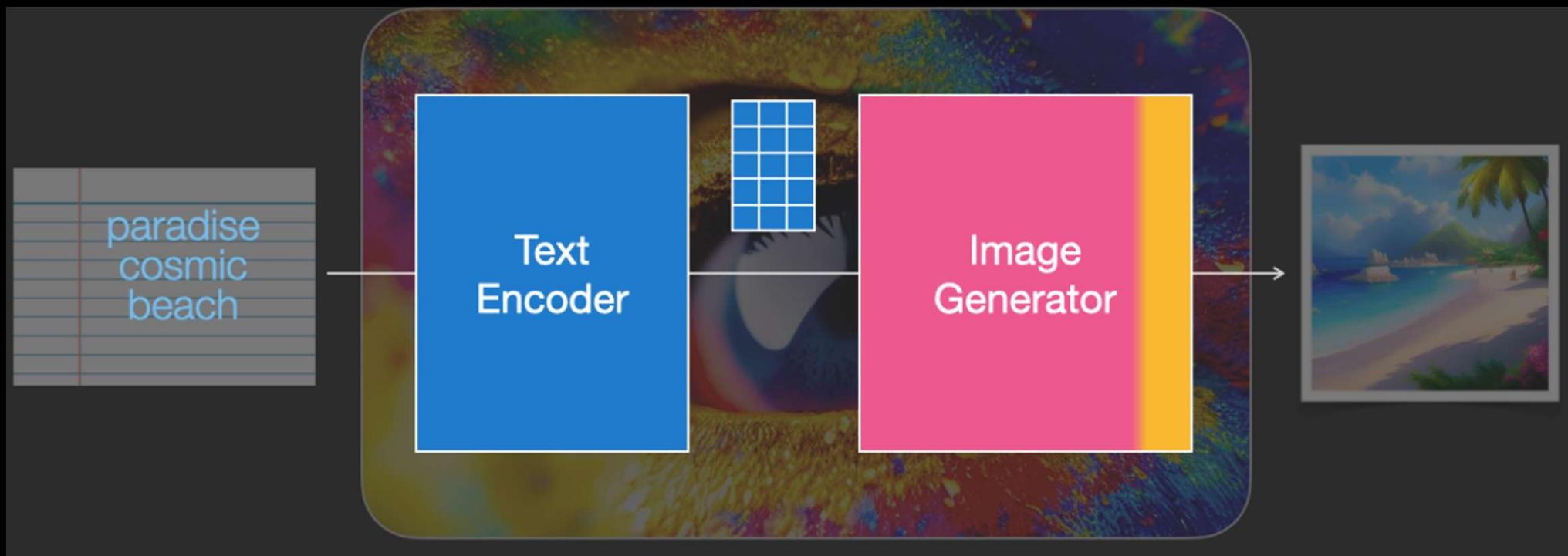
$$z = \mu + \sigma \cdot \epsilon$$

Example: [1.2, -0.6]

4. Decoder Output:

Reconstructed image of shape (28, 28, 1).

Stable Diffusion



<https://jalammar.github.io/illustrated-stable-diffusion/>

Stable Diffusion training dataset creation

Training examples are created by generating **noise** and adding an **amount** of it to the images in the training dataset (forward diffusion)

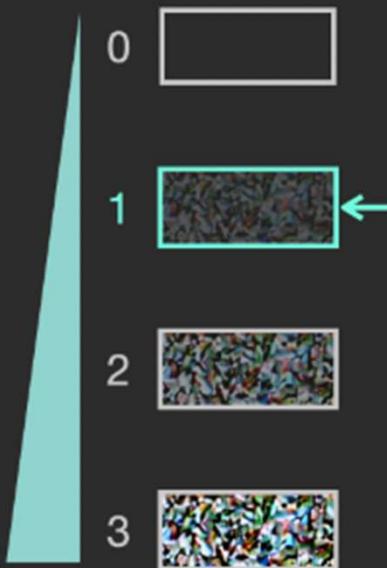
1
Pick an image



2
Generate some random **noise**



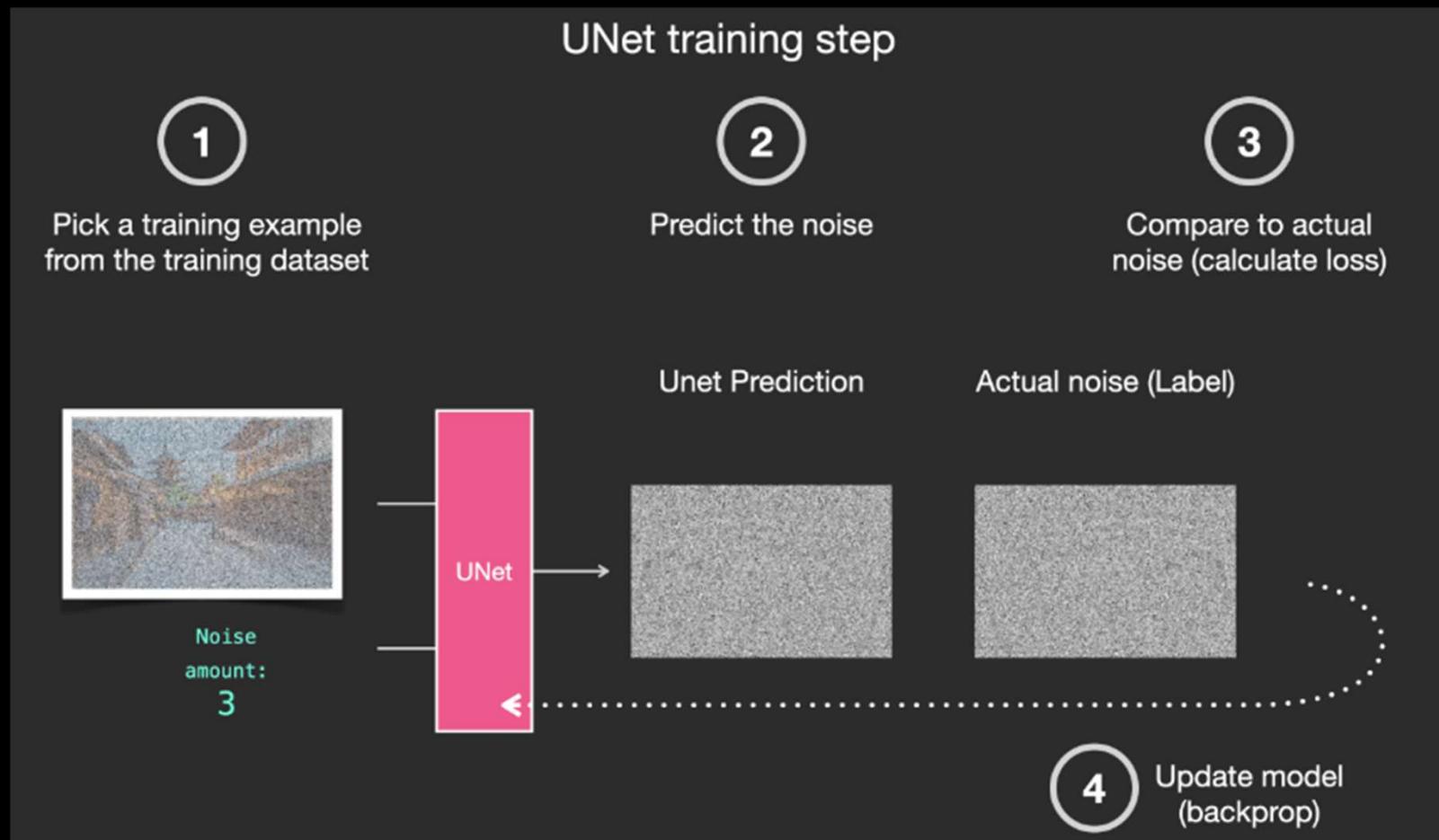
3
Pick an amount of **noise**



4
Add **noise** to the image in that **amount**

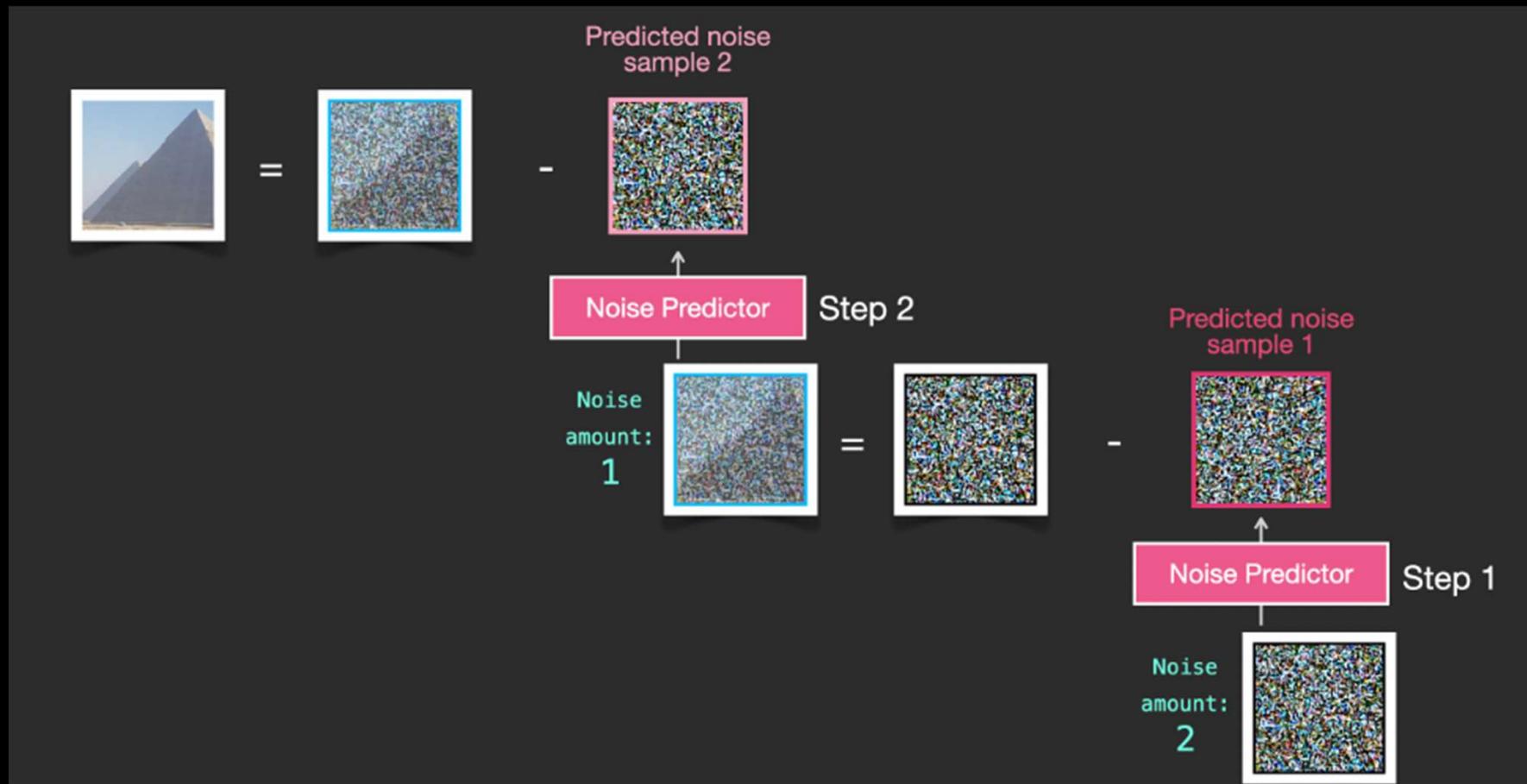


Stable Diffusion training



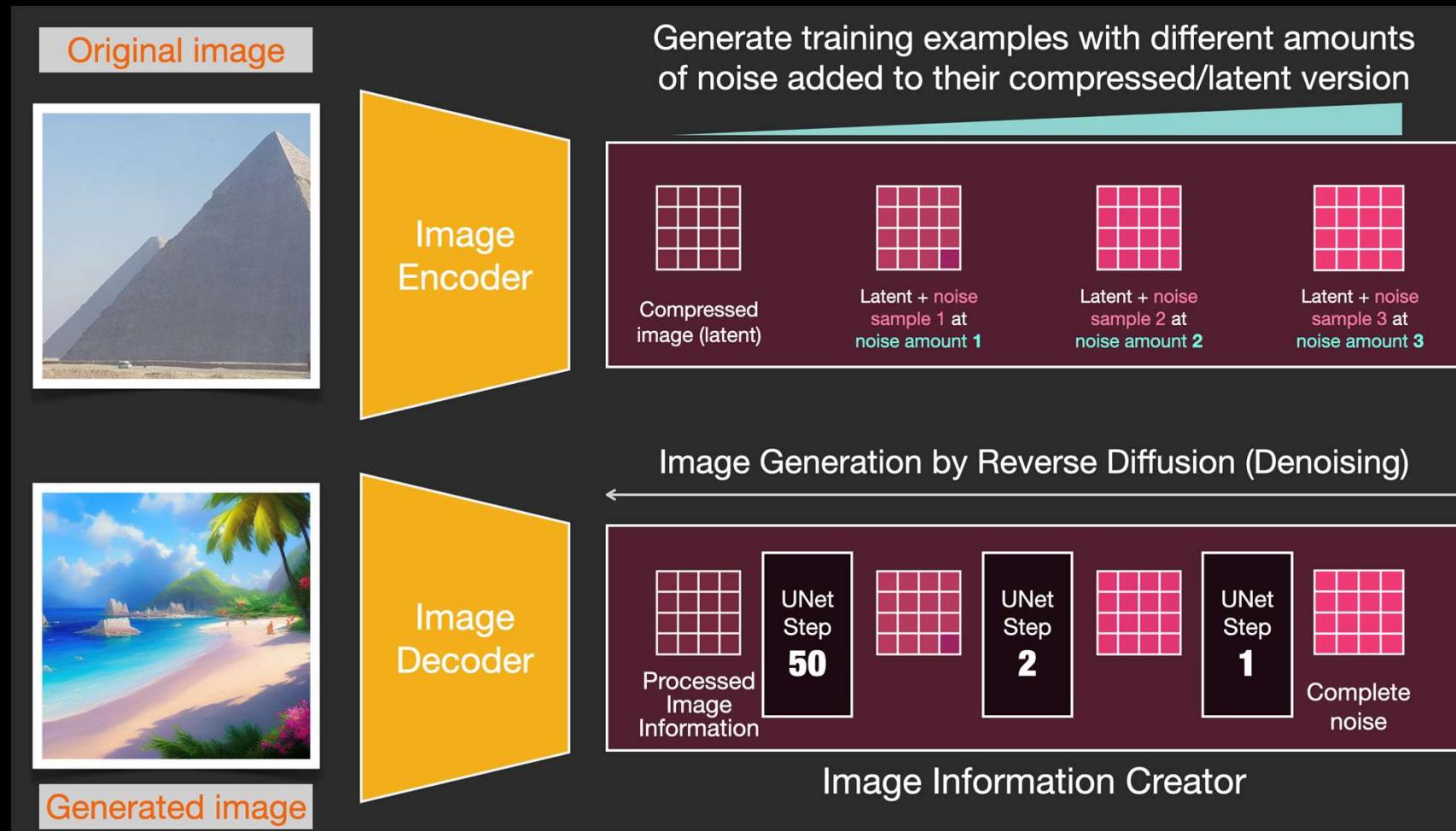
<https://jalammar.github.io/illustrated-stable-diffusion/>

Stable Diffusion – image generation by de-noising



<https://jalammar.github.io/illustrated-stable-diffusion/>

Stable Diffusion – image generation by de-noising



<https://jalammar.github.io/illustrated-stable-diffusion/>

Stable Diffusion – leveraging CLIP

Contrastive Language Image Pre-training (CLIP) is trained on a dataset of images and their captions (~400 million images and their captions)

IMAGE



CAPTION

Photo pour Japanese
pagoda and old house in
Kyoto at twilight - image
libre de droit



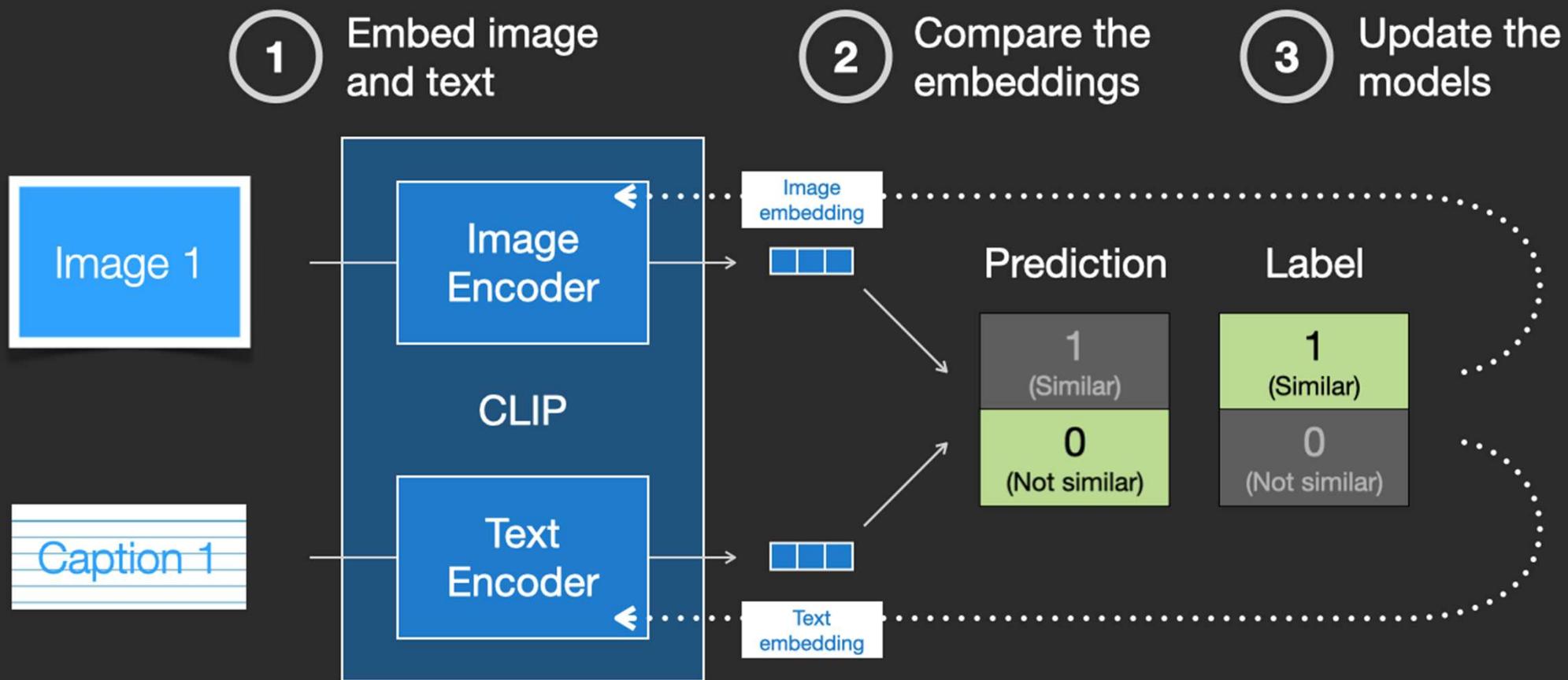
Soaring by Peter
Eades



FARCRY4 ©Copyright Ubisoft Entertainment. All Rights Reserved.
far cry 4 concept art is the
reason why it's a
beautiful game vg247.
Black Bedroom Furniture
Sets. Home Design Ideas

<https://jalammar.github.io/illustrated-stable-diffusion/>

Stable Diffusion – adding text embeddings encoder



<https://jalammar.github.io/illustrated-stable-diffusion/>

Stable Diffusion – adding text embeddings encoder



<https://jalammar.github.io/illustrated-stable-diffusion/>

Comparing the image generation architectures (1/3)

Feature	GAN (Generative Adversarial Network)	VAE (Variational Autoencoder)	Stable Diffusion
Architecture	Generator and Discriminator in adversarial setup	Encoder-Decoder with probabilistic latent space	U-Net with noise diffusion in a latent space
Working Principle	Generator creates images; Discriminator judges real vs fake	Encodes input into a latent distribution and reconstructs it	Iterative denoising from noise to image
Training Complexity	High; adversarial training can be unstable	Moderate; stable but can lead to blurry outputs	High; stable but computationally intensive
Inference Complexity	Low; single forward pass	Low; single forward pass	High; multiple denoising steps
Training Cost	High due to adversarial competition	Low to moderate	Very high due to iterative learning
Inference Cost	Low	Low	High; slower due to iterative refinement

Comparing the image generation architectures (2/3)

Feature	GAN (Generative Adversarial Network)	VAE (Variational Autoencoder)	Stable Diffusion
Output Quality	High, but prone to mode collapse	Moderate; often blurry	Very high; detailed, sharp, and diverse
Diversity of Outputs	Limited due to mode collapse	Moderate	High; better at generating unique and varied results
Latent Space	Unstructured and harder to navigate	Smooth and continuous	Compressed, structured, and interpretable
Robustness	Low; sensitive to hyperparameters	High	High; consistent across conditions
Main Use Cases	Image generation, face generation, deepfakes	Anomaly detection, data compression, reconstruction	Text-to-image, image editing (inpainting, outpainting), art generation
Conditional Generation	Challenging to implement	Limited	Built-in; excels in text-to-image generation

Comparing the image generation architectures (3/3)

Feature	GAN (Generative Adversarial Network)	VAE (Variational Autoencoder)	Stable Diffusion
Mode Collapse Risk	High	Low	Very low
Training Time	Long	Short to moderate	Very long
Interpretability	Low	High	High
Resolution Support	High, but less effective for very large images	Low to moderate	High; built for high-resolution generation
Realism of Output	High, photorealistic	Moderate; less sharp	Very high; fine details and textures
Noise Tolerance	Low	Moderate	High; denoising is part of the process
Deployment Ease	Easy after training	Easy	Complex due to high inference cost
Energy Efficiency	Moderate	High	Low; requires significant GPU resources

Speech to Text Models

Open AI **Whisper**

Uses a Transformer-based encoder-decoder architecture trained on a large dataset of diverse audio.

Google Speech to Text

Google's cloud service for speech recognition.

Based on large-scale neural network models (originally LSTM/RNN-based, now moving towards Transformer-based architectures).

Commercial services:

Amazon Transcribe, Microsoft Azure Speech, Apple Siri

<https://huggingface.co/spaces/openai/whisper>

Text to speech models – how do they work?

Text Analysis / Front-End

The input text is normalized (expanding abbreviations, handling numbers, punctuation),

Acoustic Model

A neural network (e.g., Tacotron, FastSpeech, or VITS) predicts an intermediate acoustic representation—commonly a mel-spectrogram (similar to embeddings creation)

Vocoder

A separate model (e.g., WaveNet, HiFi-GAN, WaveRNN) that converts the spectrogram to raw audio waveforms.

Post-Processing

The generated audio can be enhanced or normalized (e.g., for loudness), voice style transfer or emotional prosody control, which modifies rhythm and intonation for different contexts.

Text to speech models

- Tacotron (encoder–decoder approach) +vocoder
- Vocoder (e.g., WaveGlow, Deepmind Wavenet)
transforms the “predicted” acoustic features into the actual waveform you hear.
Wavenet is one of initial vocoders, trained on audio data, can synthesize highly realistic speech.
- VITS (Variational Inference TTS)
- Commercial services
Amazon Polly, Microsoft Azure TTS, Google Cloud TTS

Multi modal models

Multimodal input refers to the ability of AI systems to accept and process and generate different types of data simultaneously, such as text, images, audio, and video.

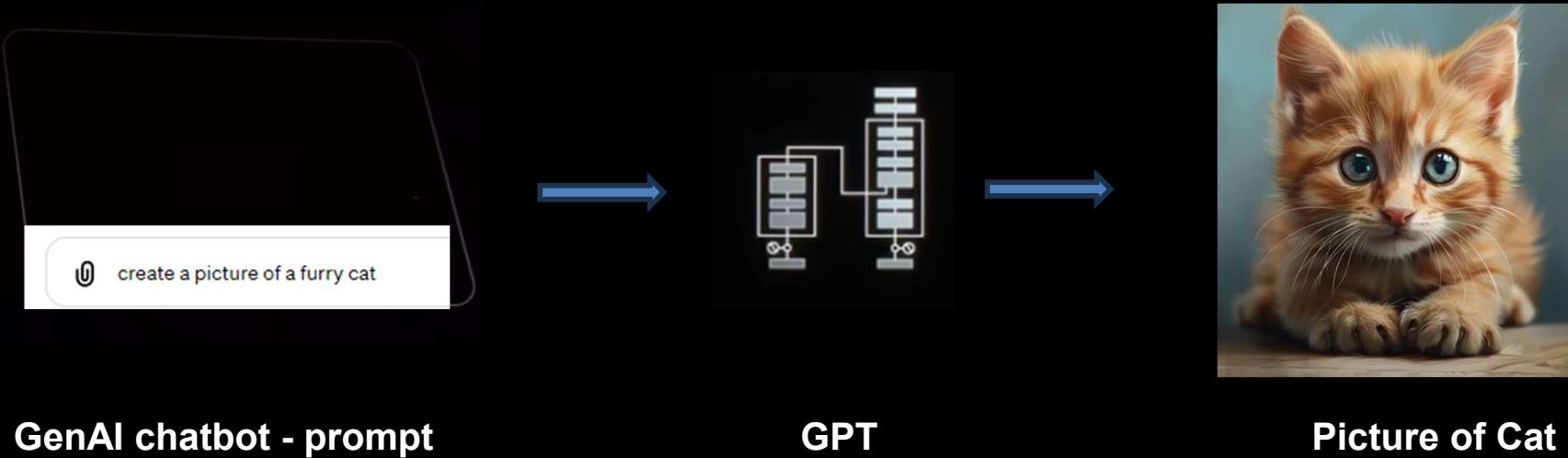
Examples:

GPT-4, Gemini, DALL-E, Stable diffusion

Video: transformers basics

Video: How transformer does prediction / generation of text

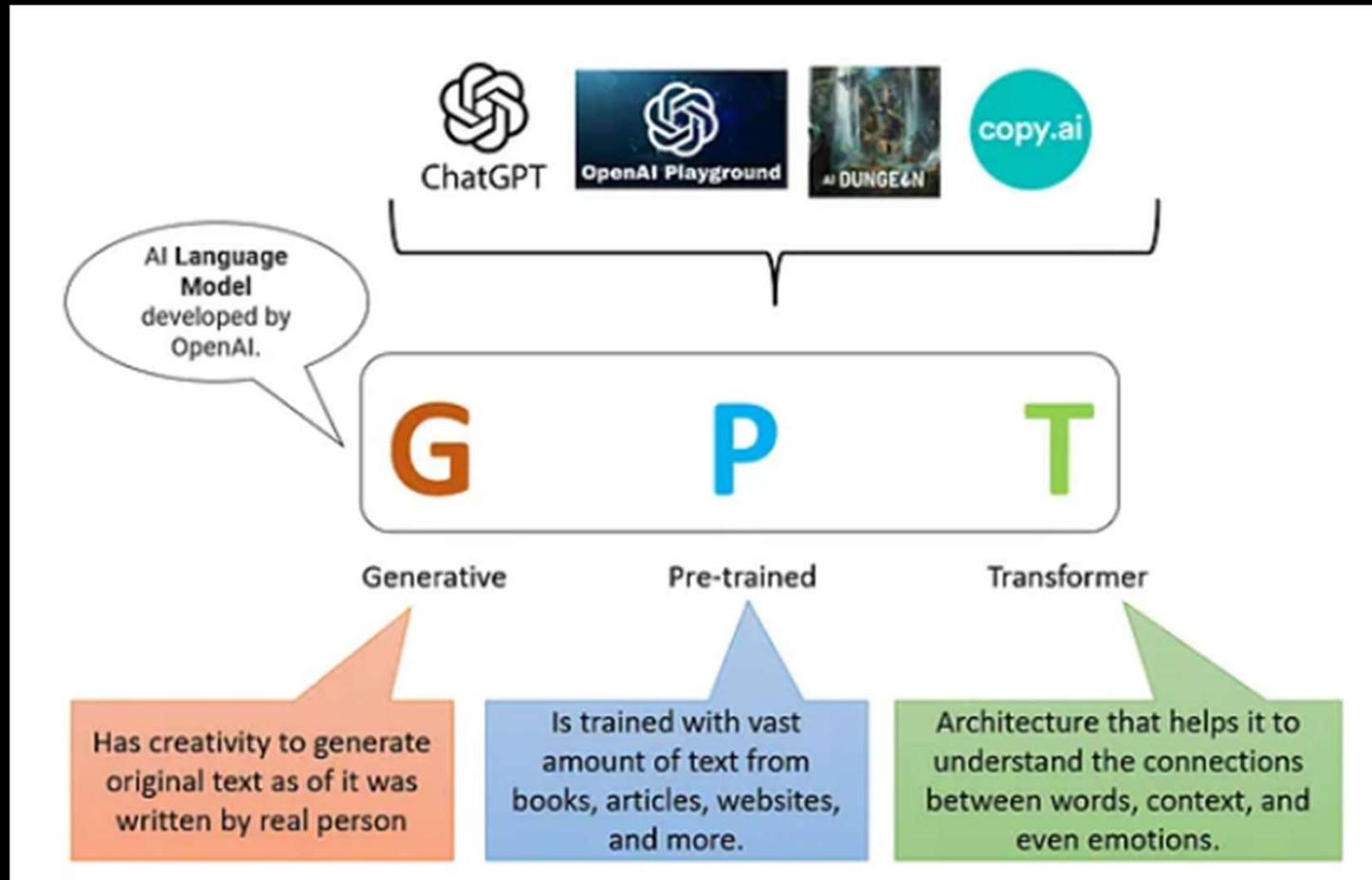
Under the hood – generating picture of cat from text



- Features of cat learnt from training data – images of cats (using convolutional neural network) – converted into embeddings
- Text “create picture of...” converted to embeddings using self-attention mechanism (captures dependencies among the input tokens)
- The embeddings are matched to find the common embedding spaces (the text and image will have same embedding positions hence making it possible to generate a picture from text)

Video: Image generation from text - goldfish

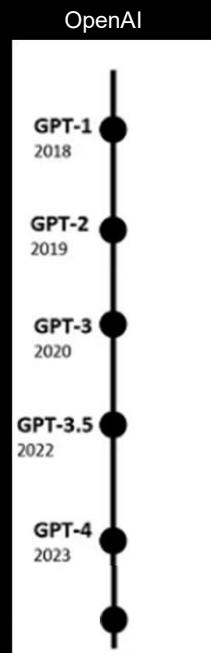
GPT



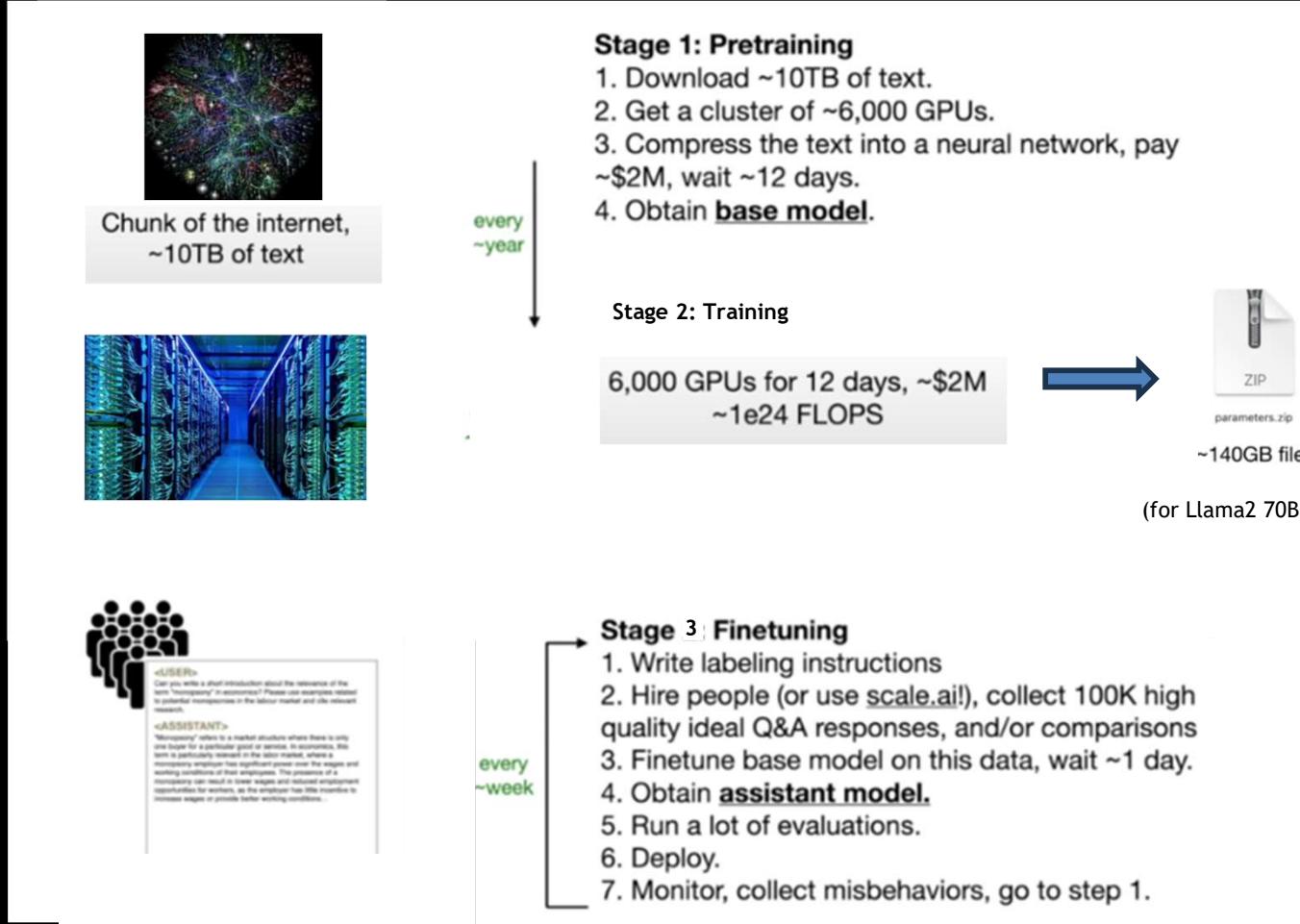
LLM – Large Language Model

General-purpose language model

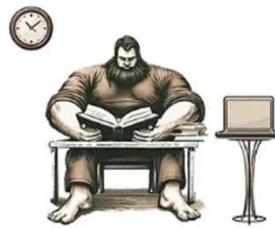
- pre-trained on massive data sets
- uses deep learning techniques
- can be fine-tuned for specific purposes
- generate natural language outputs



Steps involved in creating a LLM



GPT4 training size estimates

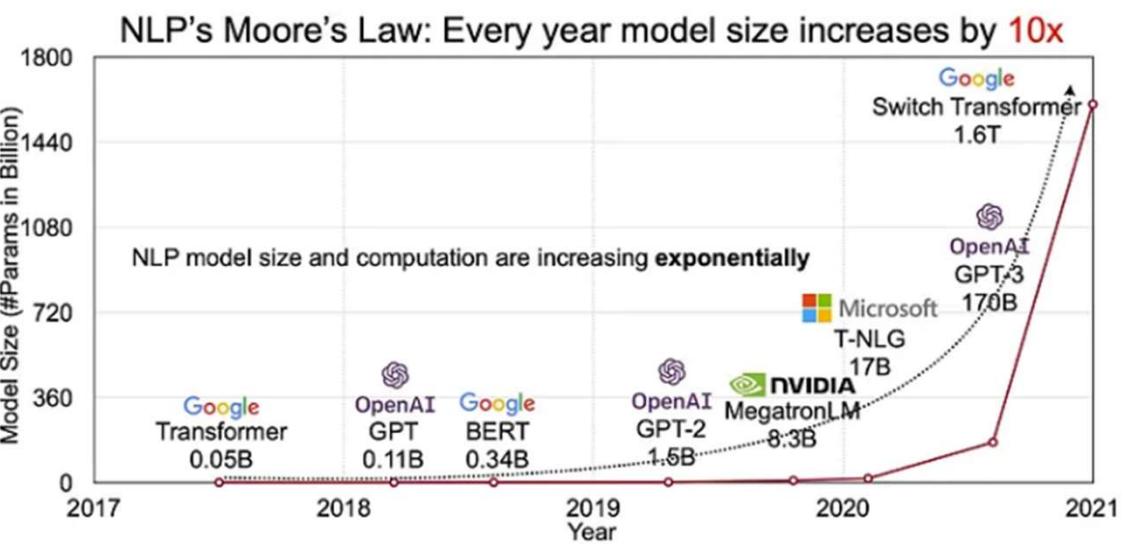
GPT4 Model Estimates		
Training Size	Compute Size	Model Size
650 kms Long line of Library Shelves  100000 tokens per Book 100 Books per shelf 2 Shelves per meter	7 million years On mid-size Laptop (100GFLOPs)  100GLOPs per second	30,000 Football Fields sized Excel Sheet  1x1 cm per Excel cell 100 x 60 meters Field Size

Source: <https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked>

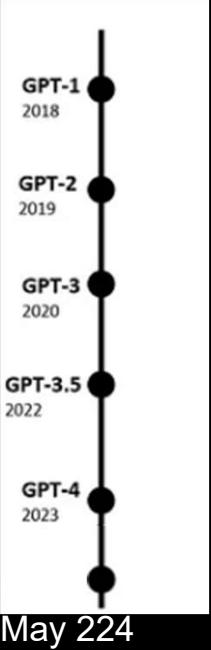
GPT 3.5 training resources and duration
90-100 days to train on 25,000 Nvidia A100 GPUs

3,125 servers
25,000 GPUs
3,125 * 14,040 to 16,900 KWh
43,875,000 to 52,812,500 KWh

GPT Model size



LLM offerings have mushroomed in less than 2 years!

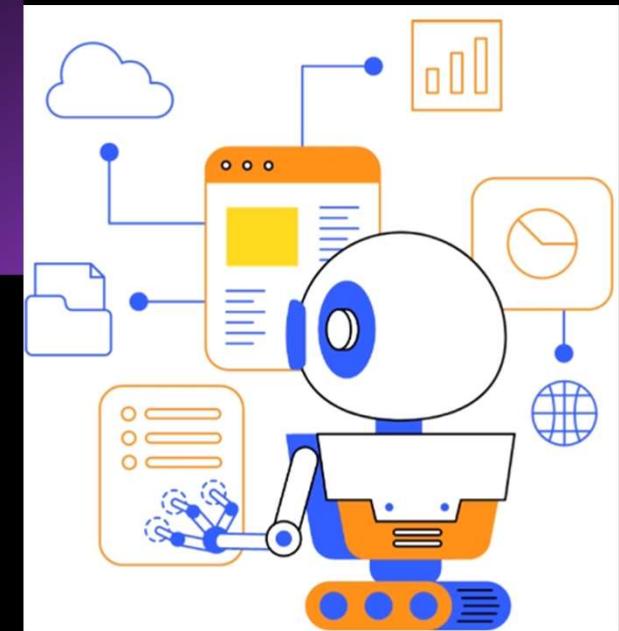
OpenAI	Microsoft CoPilot	Google Gemini	Meta Llama	Anthropic Claude	Mistral
 <p>GPT-1 2018</p> <p>GPT-2 2019</p> <p>GPT-3 2020</p> <p>GPT-3.5 2022</p> <p>GPT-4 2023</p> <p>May 224</p>	<ul style="list-style-type: none">• Azure OpenAI Jan 2023• Integrated with Office365, GitHub, Power Platform, Teams	<ul style="list-style-type: none">• Bard Mar 2023• Gemini Feb 2024	<ul style="list-style-type: none">• Opensource• Llama2 Jul 2023• Llama3 Apr 2024	<ul style="list-style-type: none">• AWS adoption• Claude2 Jul 2023• Claude3 Mar 2024	<ul style="list-style-type: none">• Opensource• Mistral 7b Sept 2023• Mistral small Feb 2024• Mistral 8x 22B Apr 2024

Generative AI

Generative AI

is a type of artificial intelligence that focuses on creating new content. It's a subset of machine learning, drawing from techniques like deep learning and reinforcement learning to generate output that can include text, images, music, video, and more.

Combination of GPT (or other **generative** models) with a chat interface



Generative Models Quiz

SORA DEMO:

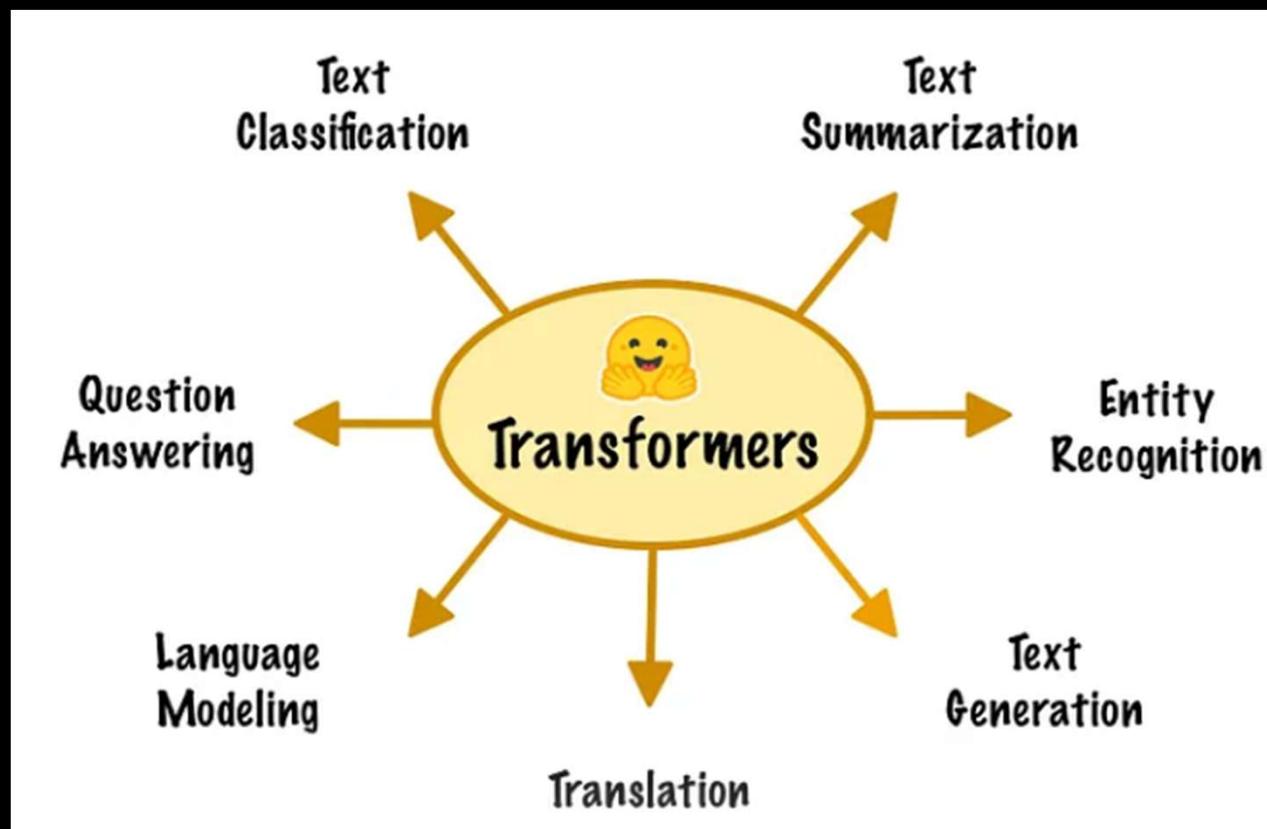
<https://openai.com/sora>



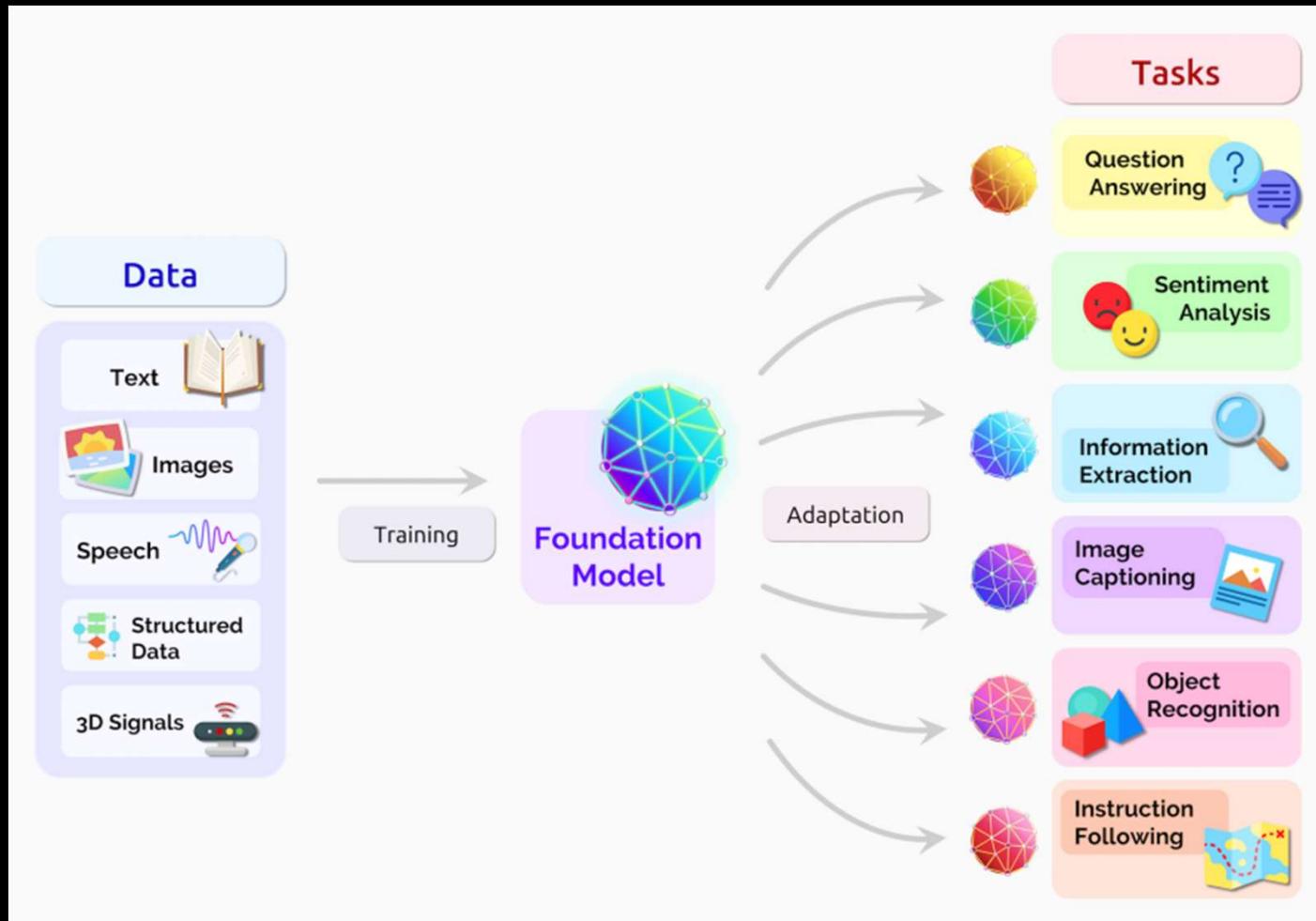
Figure 01 Robot demo

<https://www.youtube.com/watch?v=Sq1QZB5aNw>

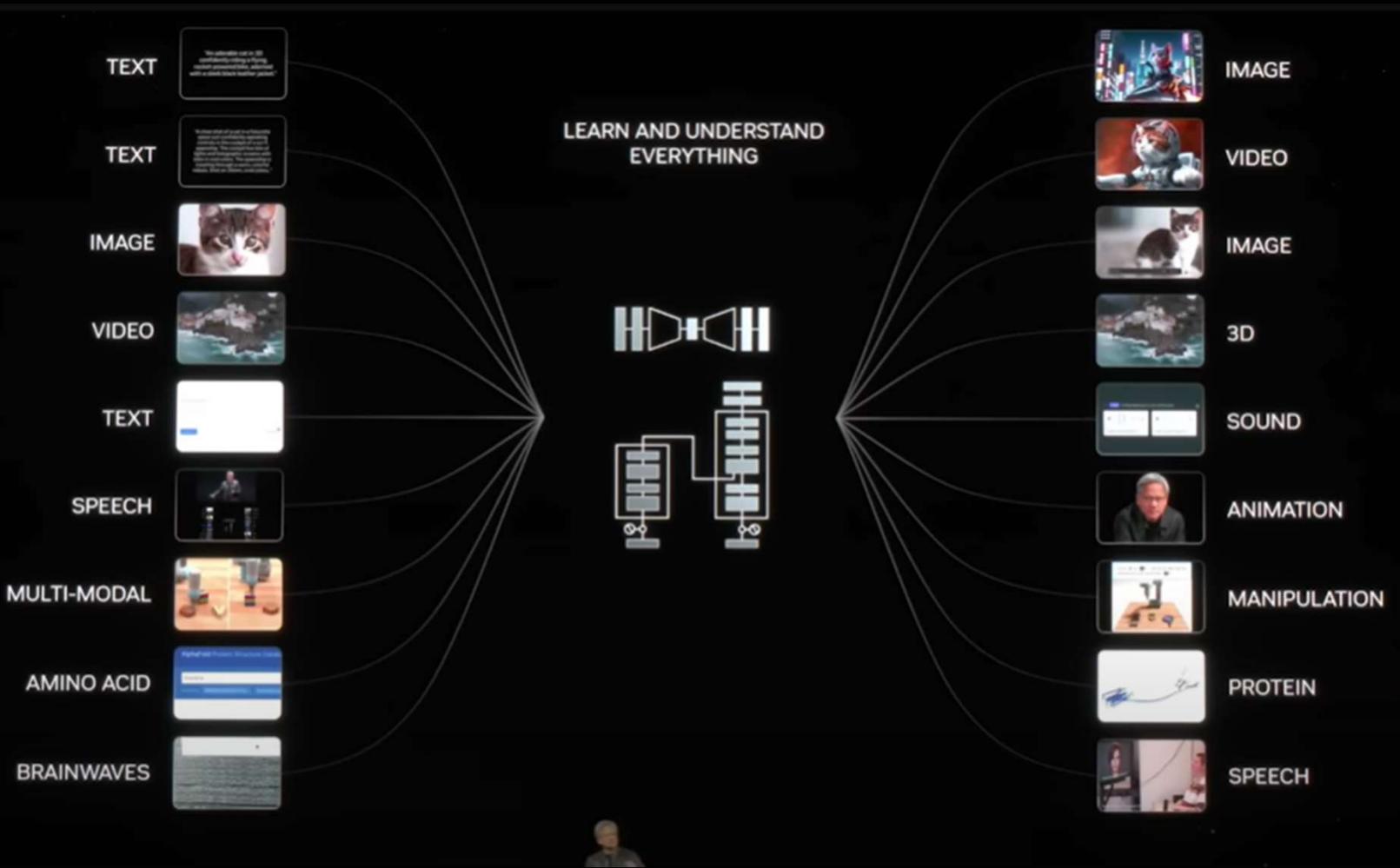
NLP



Foundation Models



Multi modal models



Core capability - inferencing

Inferencing involves generating outputs based on **patterns** and **knowledge** learned during their **training**. This can include answering questions, completing sentences, translating text, and more

Prompt: "What is the capital of France?"

Response: "Paris."

Prompt: A detailed article about climate change.

Response: "Climate change is a significant global issue caused by human activities leading to rising temperatures and environmental impacts."

Translation: Translating text from one language to another.

Prompt: "Hello, how are you?" (English)

Response: "Hola, ¿cómo estás?" (Spanish)

Text Generation: Producing creative content such as stories, poems, or essays.

Prompt: "Once upon a time in a faraway land,"

Response: "there was a small village surrounded by towering mountains and lush forests."

Sentiment Analysis: Determining the sentiment or emotional tone of a text.

Example: Input: "I love this product!"

Output: "Positive sentiment."

Core capability - summarization

It can summarize large documents, videos, images, flowcharts, audio – any modality, multiple languages

Extractive Summarization:

The LLM selects and concatenates key sentences or phrases from the original text.

Abstractive Summarization:

The LLM generates new sentences that paraphrase the original content.

Practical applications

News summarization, scientific papers summarization, legal documents, product reviews

Challenges and limitations:

May miss critical details or emphasize less important information affecting comprehensiveness

Coherence and relevance: While generally good, can be lacking sometimes

Bias and errors: Biases present in training data can result in skewed or inaccurate summaries

Core capability - reasoning

Reasoning in AI involves higher-level cognitive processes (compared to inferencing) where the AI system makes decisions based on logic, understanding relationships, and drawing conclusions from given information.

Deductive reasoning:

All humans are mortal. Socrates is a human. Therefore, Socrates is mortal.

Inductive reasoning

Every time I water my plant, it grows taller. Watering plants regularly helps them grow taller.

Abductive reasoning

Ability to choose best possible inference. Lawn is wet – rain? Or sprinkler?

Practical applications

Analyse market data to analyse patterns, customer behaviour analysis, drug discovery