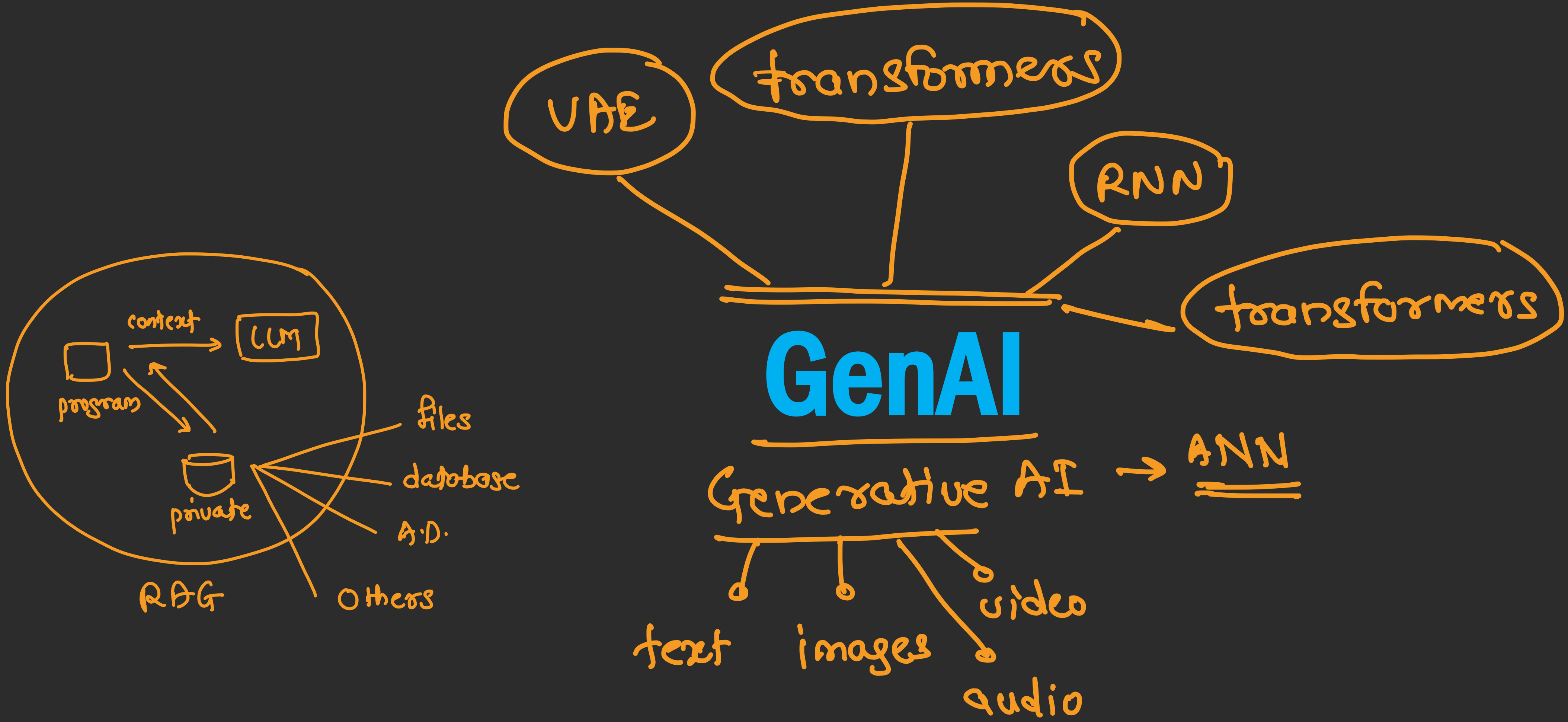
GenAI


VAE
transformers
RNN
transformers
Generative AI → ANN
text
images
video
audio
context
LLM
program
private
files
database
A.D.
Others
RAG

GenAI

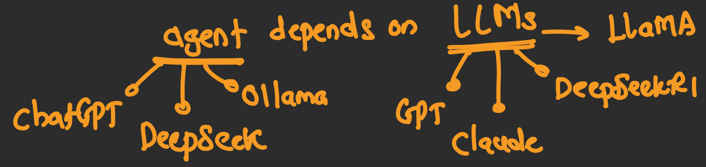
AI → ML → DL → Gen AI

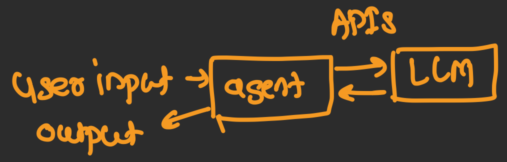
agent → app / software which uses LLM model for content generation

- Generative Artificial Intelligence (GenAI) is a subfield of AI that focuses on creating artificial agents that can generate novel, creative, or original content, such as text, images, music, or even entire stories

observed / past data

- In traditional AI, the primary goal is to analyze and process existing data to make predictions, classify objects, or recognize patterns. GenAI, on the other hand, aims to create new information that was not previously known or existed
- This includes generating:
  - **Text**: Like writing articles, stories, or even entire books , code
  - **Images**: Producing synthetic images from scratch, like creating artwork or photographs
  - **Music**: Composing music in various styles, genres, or moods
  - **Videos**: Generating video content, such as animations, films, or live-action videos





- GenAI draws inspiration from human creativity, imagination, and cognitive abilities
- It's a fusion of AI techniques, including:

→ on its own / without No human intervention

  - **Deep Learning**: Neural networks that can learn patterns in data & generate new information based on patterns
  - **Cognitive Architectures**: High-level models that enable agents to reason, plan, and make decisions
  - **Evolutionary Algorithms**: Optimization techniques inspired by natural evolution, where agents adapt and evolve over time

Traditional AI vs GenAI

Discriminative

patterns

Traditional AI

past data → prediction

- Focuses on processing and analyzing existing data to make predictions, classify objects, or recognize patterns
- Goal is to improve accuracy, efficiency, and decision-making through data-driven insights
- Techniques: Machine learning, deep learning, rule-based systems, optimization algorithms
- Applications:
  - Customer service chatbots
  - Predictive maintenance in manufacturing
  - Image recognition for self-driving cars
  - Natural language processing for text analysis

↗ ANN

Generative AI

→ content creation

- Focuses on creating new, original content or generating novel outputs that did not previously exist
- Goal is to unlock human-like creativity, imagination, and innovation through AI-generated content
- Techniques: Deep learning, cognitive architectures, evolutionary algorithms, generative models
- Applications:
  - Artistic content creation
  - Content generation for media outlets
  - Product design and prototyping for industries
  - Educational tools for interactive learning

↳ personalized Education

Traditional AI vs GenAI

- **Key differences:**

  prediction

  - **Purpose:** Traditional AI aims to analyze data, while GenAI focuses on generating new content
  - **Techniques:** Traditional AI relies on machine learning and optimization algorithms, whereas GenAI employs deep learning, cognitive architectures, and generative models → LLMs
  - **Applications:** Traditional AI is commonly used in areas like customer service or image recognition, whereas GenAI has applications in creative industries, education, and product development
- **Shared characteristics**
  - **Automation:** Both traditional AI and GenAI involve automating tasks or processes to improve efficiency and accuracy
  - **Data-driven:** Both types of AI rely on data to inform their decision-making and generation capabilities
  - **Continuous improvement:** Both traditional AI and GenAI require ongoing training, testing, and refinement to maintain performance and effectiveness.

Applications of GenAI

Chatbots

- Alexa (Amazon): A virtual assistant that can answer questions, play music, and control smart home devices.
- Google Assistant: A virtual assistant that can answer questions, perform tasks, and control other devices.
- Cortana (Microsoft): A virtual assistant that can answer questions, set reminders, and control other devices.

Language Translation

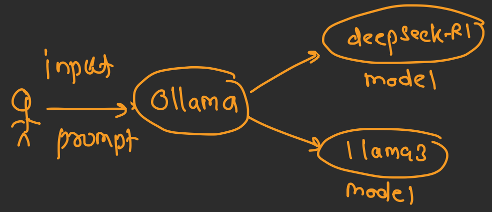
- Google Translate: A machine translation system that can translate text from one language to another.
- Microsoft Translator: A machine translation system that can translate text from one language to another.
- DeepL (Deep Learning-based Machine Translation): A machine translation system that uses deep learning to improve translation accuracy.

Image Generation

- DALL-E (Deep Learning-based Algorithm for Latent Embedding): An AI model that generates images from text prompts.
- Midjourney (Midjourney is an AI-powered image generation platform): A platform that allows users to generate images using text prompts and AI algorithms.

Applications of GenAI

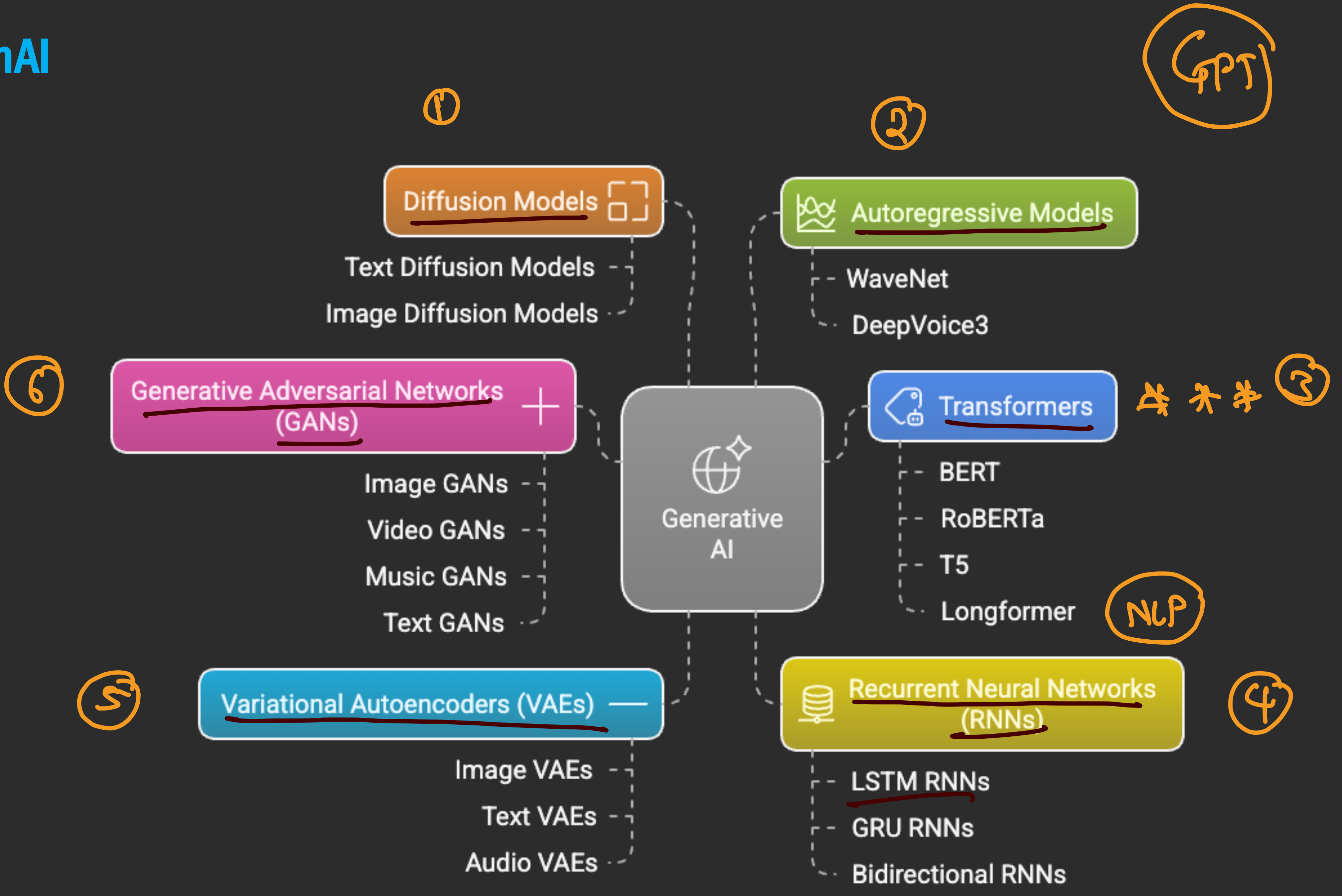
- **Conversational AI**
  - Watson Assistant (IBM): A conversational AI platform that can answer questions, perform tasks, and control other devices
  - Amelia (Fidelity Investments): A virtual assistant that uses GenAI to automate customer service interactions
- **Robotics**
  - Sophia (Hanson Robotics): A humanoid robot that uses GenAI to recognize and respond to facial expressions, speech, and gestures
  - Pepper (SoftBank Robotics): A humanoid robot that uses GenAI to recognize and respond to human emotions, speech, and gestures
- **Virtual Assistants**
  - Siri (Apple): A virtual assistant that can answer questions, perform tasks, and control other devices
  - Bixby (Samsung): A virtual assistant that can answer questions, perform tasks, and control other devices
- **Computer Vision**
  - Google Cloud Vision API: A computer vision platform that can analyze images and recognize objects, text, and emotions
  - Amazon Rekognition: A computer vision platform that can analyze images and recognize objects, text, and emotions



architectures

Types of GenAI Models

Types of GenAI



GPT

Generative AI

1 Diffusion Models
- Text Diffusion Models
- Image Diffusion Models

2 Autoregressive Models
- WaveNet
- DeepVoice3

Transformers *** 3
- BERT
- RoBERTa
- T5
- Longformer

NLP

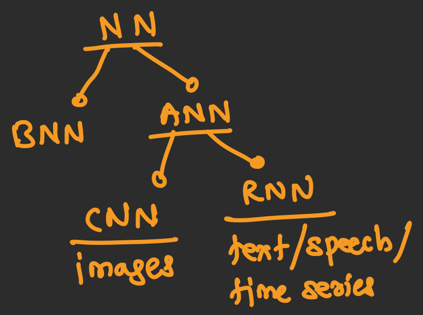
Recurrent Neural Networks (RNNs) 4
- LSTM RNNs
- GRU RNNs
- Bidirectional RNNs

5 Variational Autoencoders (VAEs)
- Image VAEs
- Text VAEs
- Audio VAEs

6 Generative Adversarial Networks (GANs)
- Image GANs
- Video GANs
- Music GANs
- Text GANs

Recurrent Neural Network (RNN)

- Recurrent Neural Networks (RNNs) play a crucial role in modelling sequential data, such as text, speech, or time-series data
- RNNs are particularly useful for capturing long-range dependencies and temporal relationships in input data
- **Key Components**:
  - **Recurrent Cells**: The core component of an RNN is the recurrent cell, which consists of:
    - **Hidden State:** A vector that captures information from previous time steps.
    - **Cell State:** A vector that carries information from previous time steps and updates the hidden state.
  - **Activation Functions**: RNNs use activation functions like tanh or sigmoid to introduce non-linearity in the hidden state
  - **Time Steps**: RNNs process input data one time step at a time, allowing them to capture sequential relationships.
- **How it Works** → sequential – iterations
  - RNNs process input sequences by iterating through each time step.
  - At each time step
    - The recurrent cell receives the current input and previous hidden state as inputs
    - The cell state is updated based on the current input and previous hidden state
    - The hidden state is updated using the activation function
    - The output of the RNN is typically a linear combination of the hidden state



Recurrent Neural Network (RNN)

- Types of RNNs
  - Simple RNN (SRNN): Basic RNN architecture with a single recurrent cell
  - Long Short-Term Memory (LSTM) RNN: A variant of SRNN that introduces a memory cell to improve long-term dependencies
  - Gated Recurrent Unit (GRU) RNN: Another variant of SRNN that uses gating mechanisms to control the flow of information
- Benefits
  - Captures Long-Range Dependencies: RNNs can capture complex temporal relationships in the input data.
  - Handles Variable-Length Input Sequences: RNNs can process input sequences of varying lengths.
  - Stateful Processing: RNNs can maintain a stateful representation of the input data, allowing for more accurate modeling.
- Challenges
  - Training Instability: RNNs can be challenging to train due to vanishing or exploding gradients.
  - Gradient Disentanglement: The gradient flow through the recurrent cell can become disentangled during training.
  - Overfitting: RNNs are prone to overfitting, especially when dealing with large input sequences
- Applications in Generative AI
  - Text Generation: RNNs are used for text generation tasks, such as language modelling, machine translation, and chatbots
  - Speech Recognition: RNNs can be used for speech recognition and synthesis
  - Time-Series Analysis: RNNs are applied to analyze and generate time-series data, like stock prices or weather patterns
  - Chatbots: RNNs can be used to generate responses to user input in chatbots
  - Sentiment Analysis: RNNs can analyze text data to determine sentiment (positive/negative) and classify it as such
  - Automatic Speech Recognition (ASR): RNNs can transcribe spoken language into text

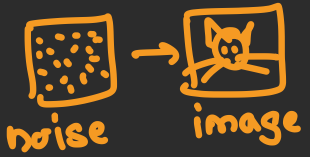
Generative Adversarial Network (GAN)

- A Generative Adversarial Network (GAN) is a type of deep learning model that consists of two neural networks: a **Generator** and a **Discriminator**
- The goal of a GAN is to learn a probability distribution over the input data, such as images or text
- **How GANs work**
  - **Generator:** The Generator takes a random noise vector as input and produces an output that attempts to mimic the real data distribution
  - **Discriminator:** The Discriminator is trained to distinguish between the generated samples (from G) and real data samples. It outputs a probability that the input sample is real or fake
  - **Training:** Both Generator and Discriminator are trained simultaneously using an adversarial process:
    - Generator tries to generate more realistic samples, making it harder for D to correctly classify them as real or fake
    - Discriminator tries to become better at distinguishing between real and generated samples, forcing Generator to improve its generation quality

Generative Adversarial Network (GAN)

- **Benefits**
  - **Flexibility**: GANs can generate new samples that are diverse and realistic, making them suitable for various applications like image synthesis, text-to-image generation, and more
  - **Unsupervised learning**: GANs do not require labelled data, which makes them useful for domains where labelling is expensive or difficult
- **Challenges**
  - **Training instability**: GAN training can be unstable due to the adversarial nature of the game between Generator and Discriminator
  - **Mode collapse**: The Generator might produce limited variations of the same sample, rather than exploring the entire data distribution
  - **Evaluation metrics**: It's challenging to evaluate GANs since they generate new samples that may not be easily comparable to real data
- **Applications**
  - **Image generation**: used for generating realistic images, such as faces, objects, and scenes
  - **Text-to-image synthesis**: GANs can generate images based on input text or captions
  - **Data augmentation**: GANs can be used to augment datasets with new samples that resemble the original data distribution

Diffusion Model

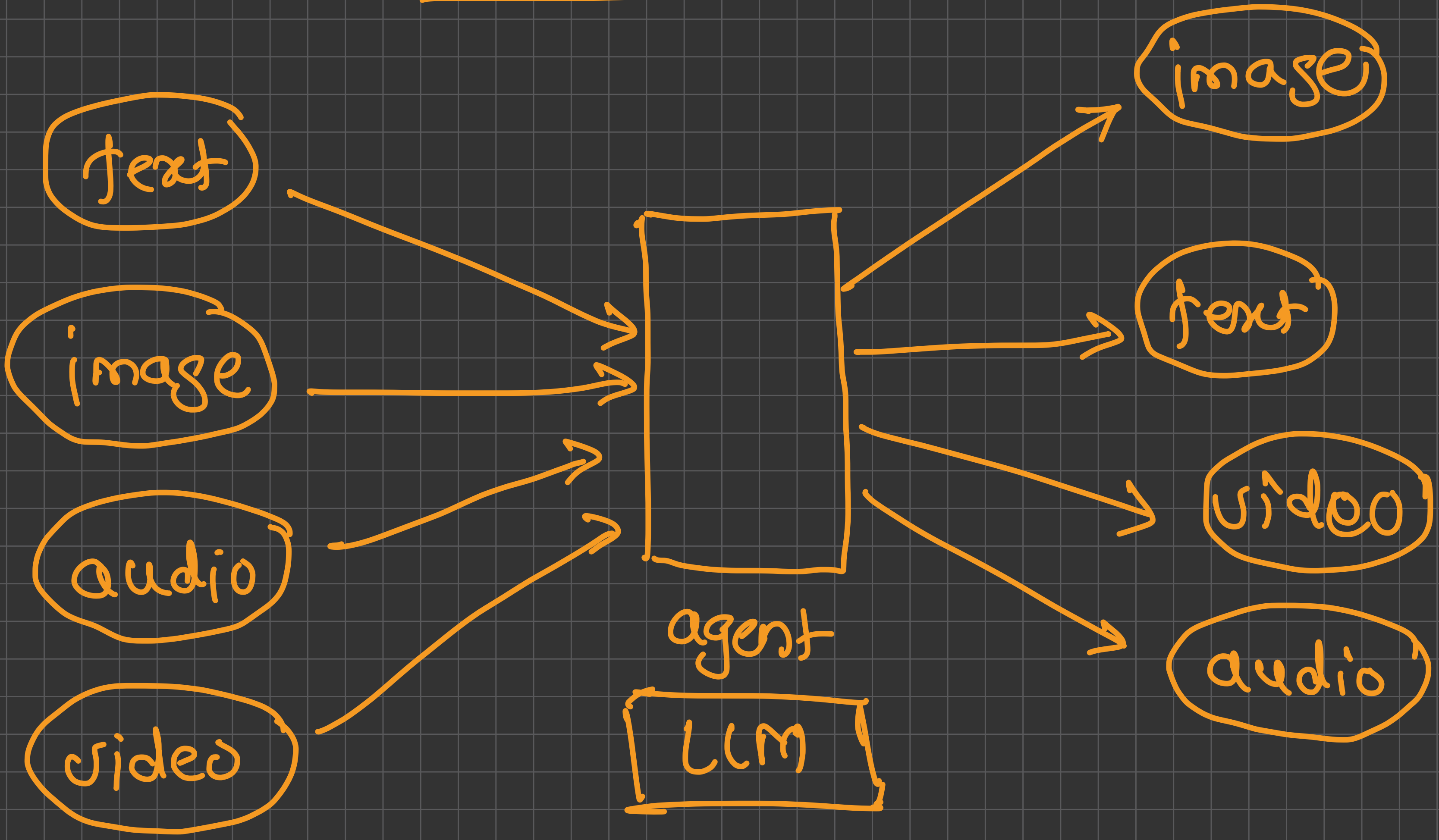


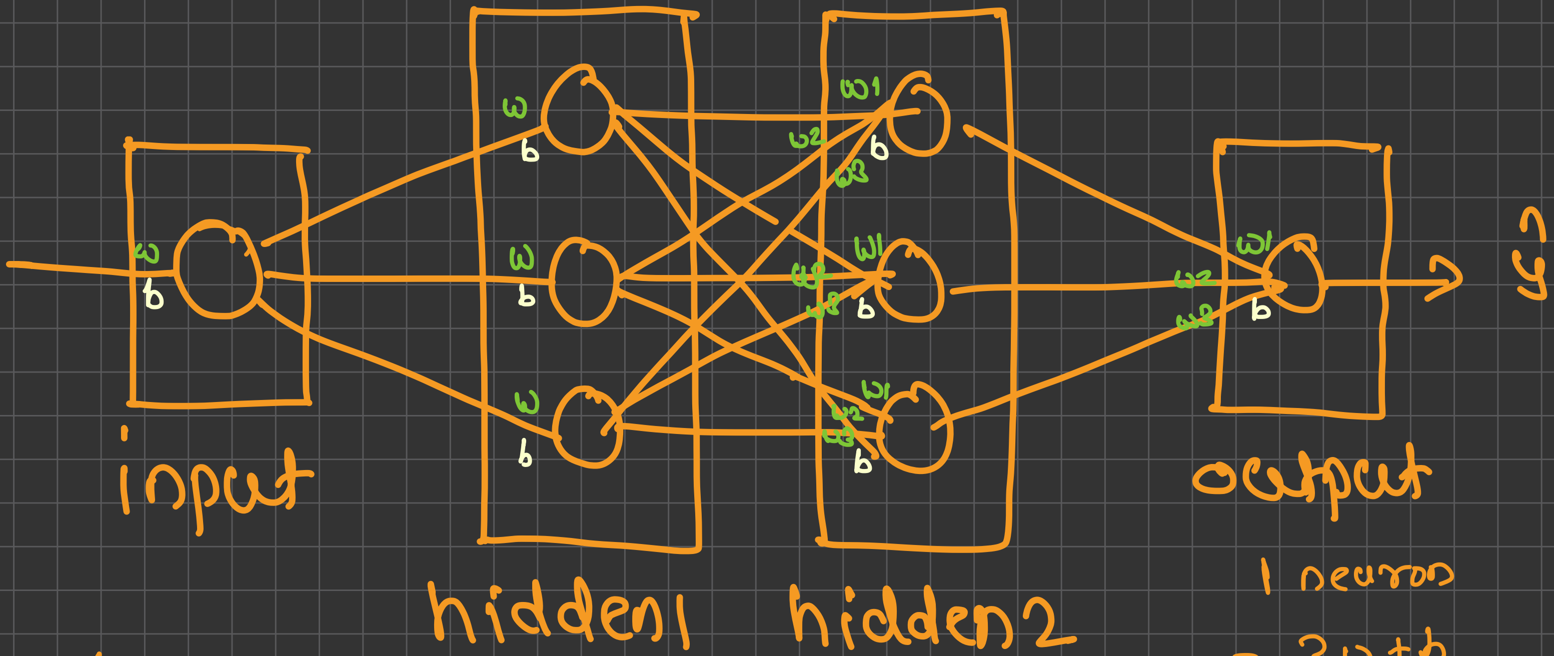
- A Diffusion GAN is a type of generative model that uses a diffusion-based process to transform a random noise signal into a realistic image or audio sample
- The key idea behind diffusion GANs is to iteratively refine the generated sample by adding noise and then transforming it back into a more realistic representation
- **How Diffusion GANs work**
  - **Noise input**: Start with a random noise signal
  - **Diffusion process**: Apply a series of transformations to the noise signal, adding Gaussian noise at each step
  - **Generative model**: Use a neural network (Generator) to transform the noisy signal into an intermediate representation
  - **Refinement**: Apply another series of transformations to the intermediate representation, refining it until you reach the final output

Diffusion Model

- **Benefits**
  - **Improved realism**: Diffusion GANs can generate more realistic images by iteratively refining the intermediate representation
  - **Flexibility**: These models are highly flexible and can be used for various applications, such as generating images from text or audio signals
  - **Stability**: The iterative refinement process helps to stabilize the generation process, making it less prone to mode collapse
- **Challenges**
  - **Training instability**: The iterative refinement process can be unstable, making it challenging to train these models.
  - **Mode collapse**: The generated samples might not capture the full range of variability in the data distribution.
  - **Evaluation metrics**: It's essential to develop suitable evaluation metrics for assessing the quality and diversity of generated samples.
- **Applications**
  - **Image generation**: Diffusion GANs can be used for generating realistic images of objects, animals, or scenes.
  - **Audio synthesis**: These models can generate audio signals from text-based descriptions or musical inputs.
  - **Text-to-image synthesis**: Combine the power of diffusion GANs with text-based input to generate images that match textual descriptions.

multi-modal





1 neuron
= 1ω + 1b
= 2

3 neurons
= 3ω + 3b
= 6

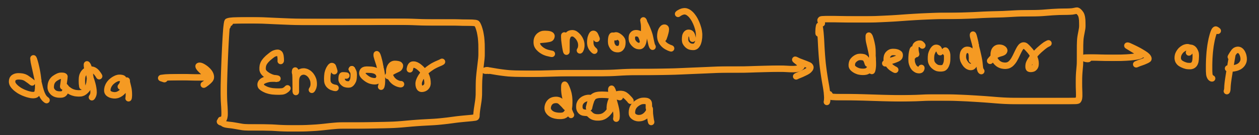
3 neurons
= 3ω × 3n = 9ω + 3b
= 12

1 neuron
= 3ω + b
= 4

total parameters = 2+6+12+4

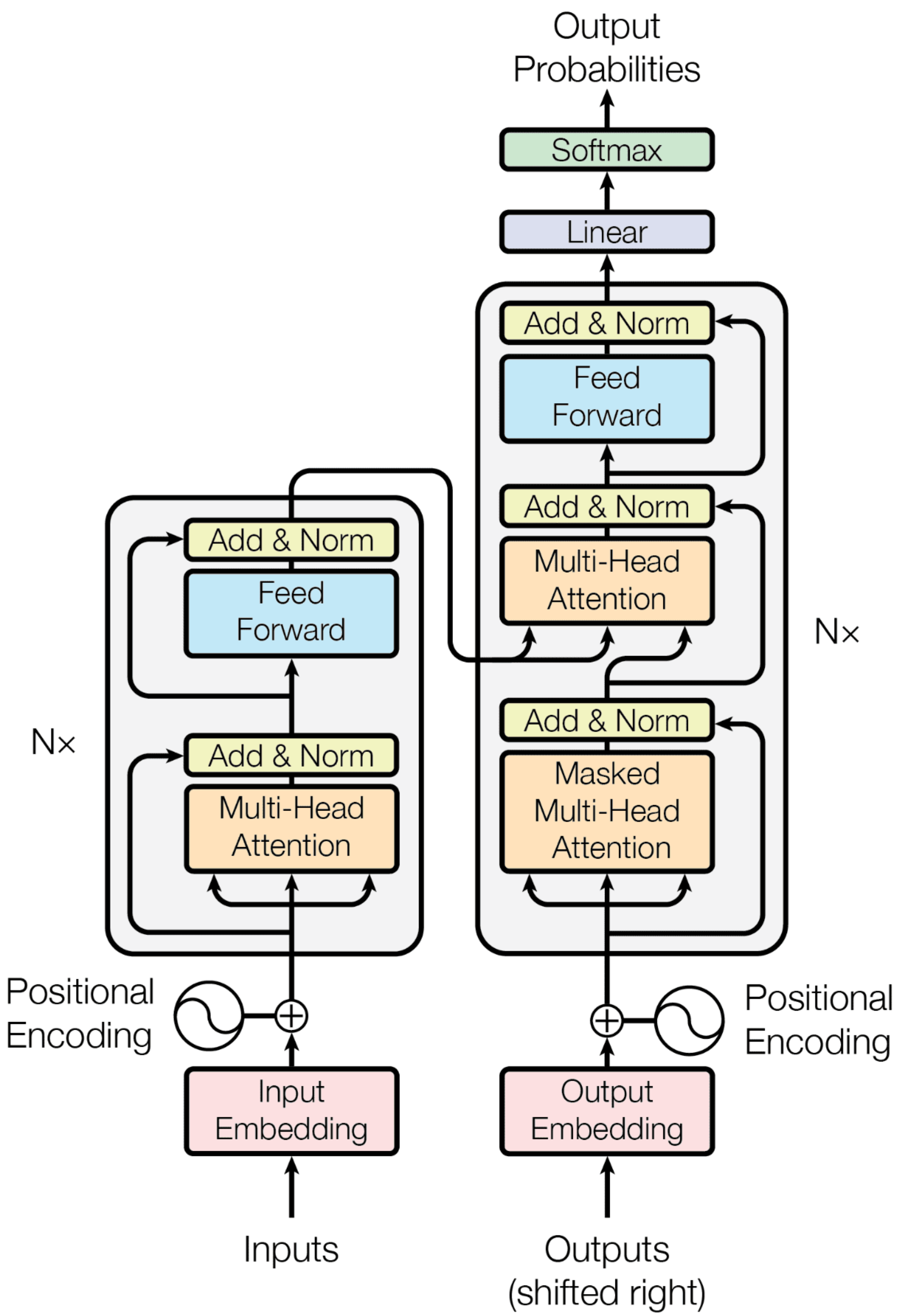
# params = 24

Transformers



- A Transformer-based GAN (T-GAN) is a type of generative model that leverages the power of Transformers, originally designed for machine translation and natural language processing tasks. In T-GANs, the Transformer architecture is modified to generate images or audio signals from random noise inputs
- **Key components**
  - **Encoder**: The input noise signal is passed through an encoder network, which transforms it into a sequence of embeddings
  - **Transformer**: The embedding sequence is then fed into a Transformer block, which consists of self-attention mechanisms and feed-forward networks (FFNs)
  - **Decoder**: The output from the Transformer block is passed through a decoder network, which generates the final output image or audio signal
- **How it works**
  - **Noise input**: Start with a random noise signal
  - **Encoder**: Transform the noise signal into an embedding sequence using the encoder network
  - **Transformer**: Pass the embedding sequence through the Transformer block to generate a contextualized representation of the input noise
  - **Decoder**: Use the output from the Transformer block as input to the decoder network, which generates the final output image or audio signal
  - **Training**: Train the T-GAN by optimizing the generator and discriminator simultaneously using a combination of reconstruction loss and adversarial loss

Transformers



Transformers

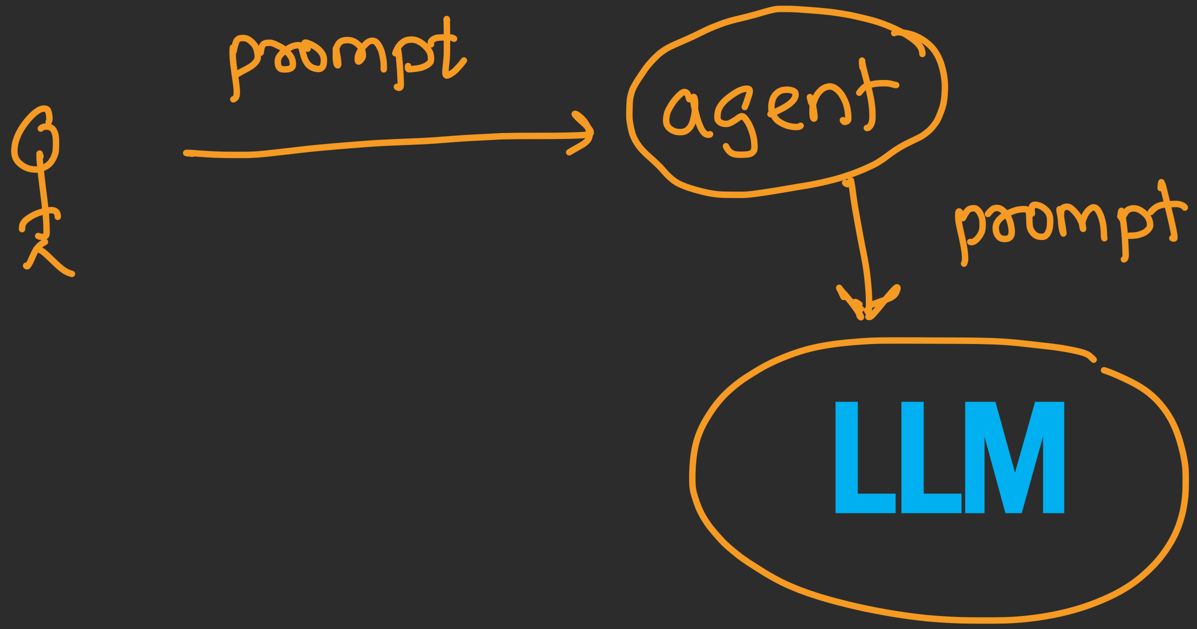
- **Benefits**
  - **Improved generation quality**: The Transformer architecture enables the model to generate more coherent and realistic images or audio signals by modelling long-range dependencies in the input noise
  - **Flexibility**: T-GANs can be easily adapted to various tasks, such as image-to-image translation, text-to-image synthesis, or audio generation
  - **Parallelization**: The Transformer architecture allows for parallelization of the computation, making it more efficient than traditional GAN architectures
- **Challenges**:
  - **Training instability**: The Transformer architecture and adversarial training process can be unstable, making it challenging to train these models
  - **Mode collapse**: The generated samples might not capture the full range of variability in the data distribution
  - **Evaluation metrics**: It's essential to develop suitable evaluation metrics for assessing the quality and diversity of generated samples
- **Applications**:
  - **Image generation**: T-GANs can be used for generating realistic images of objects, animals, or scenes from random noise inputs
  - **Audio synthesis**: These models can generate audio signals from text-based descriptions or musical inputs
  - **Text-to-image synthesis**: Combine the power of T-GANs with text-based input to generate images that match textual descriptions

Variational Autoencoder (VAE)

- A Variational Autoencoder (VAE) is a type of generative model that uses an encoder network to learn a probabilistic representation of the input data, and a decoder network to generate new samples from this representation
- **Key Components**
  - **Encoder (Q-Net)**: The encoder network takes in the input data and outputs a distribution over the latent space, often represented as a mean and variance vector
  - **Latent Space**: The output of the encoder is a probabilistic representation of the input data, which captures the underlying structure and patterns in the data
  - **Decoder (P-Net)**: The decoder network takes in the latent representation and generates new samples that are close to the original input data
- **How it Works**
  - VAEs are trained using a combination of two losses:
  - **Reconstruction Loss**: This loss measures the difference between the original input data and the reconstructed output from the VAE
  - **Kullback-Leibler (KL) Divergence Loss**: This loss measures the difference between the learned latent distribution and an isotropic Gaussian distribution

Variational Autoencoder (VAE)

- **Benefits**:
  - **Probabilistic**: VAEs learn a probabilistic representation of the input data, which allows for efficient sampling and generation
  - **Continuous Latent Space**: The latent space is continuous, allowing for smooth interpolation between different points in the space
  - **Flexible**: VAEs can be used for various tasks, such as dimensionality reduction, feature learning, and generative modelling
- **Challenges**:
  - **Training Instability**: VAEs can be difficult to train due to the competition between the reconstruction loss and KL divergence loss
  - **Mode Collapse**: The generated samples might not capture the full range of variability in the data distribution
  - **Evaluation Metrics**: It's essential to develop suitable evaluation metrics for assessing the quality and diversity of generated samples
- **Applications**:
  - **Generative Modelling**: VAEs are often used for generative tasks, such as image generation, text-to-image synthesis, or audio generation
  - **Dimensionality Reduction**: VAEs can be used to reduce the dimensionality of high-dimensional datasets, making it easier to visualize and analyze the data
  - **Feature Learning**: VAEs can be used to learn meaningful features from input data, which can be useful for downstream tasks like classification or regression


prompt
agent
prompt
LLM

What are LLMs ?

- Large Language Models (LLMs) are advanced AI systems trained on massive amounts of text data to understand and generate human-like text
- They learn patterns in language through billions or trillions of parameters, enabling them to:
  - Generate coherent text
  - Answer questions
  - Translate languages
  - Write code
  - Analyze content
  - Perform various language-based tasks
- Modern LLMs like GPT-4, Claude, and PaLM use transformer architecture and learn from context to produce relevant outputs
- They work by predicting the most likely next words based on the input (prompt) they receive

Capabilities of LLM

- Text Generation
  - Creative Writing: Generate stories, poems, or scripts
  - Content Creation: Write articles, blog posts, and marketing copy
- Text Completion
  - Autocomplete: Suggest completions for partially written sentences or paragraphs
  - Code Completion: Assist with programming tasks by completing code snippets
- Summarization
  - Extractive Summarization: Identify and extract key sentences from a larger text
  - Abstractive Summarization: Generate concise summaries that paraphrase the original content
- Translation
  - Language Translation: Translate text between different languages with varying degrees of fluency
- Question Answering
  - Information Retrieval: Answer factual questions based on a given context or general knowledge
  - Conversational Agents: Engage in dialogue and provide informative responses
- Sentiment Analysis
  - Opinion Mining: Analyze text to determine sentiment (positive, negative, neutral)

Capabilities of LLM

- Text Classification
  - Categorization: Classify documents or messages into predefined categories (spam detection, topic classification)
- Named Entity Recognition (NER)
  - Entity Extraction: Identify and classify entities (people, organizations, locations) in the text
- Conversational AI
  - Chatbots: Engage users in natural language conversations for customer support or information retrieval
- Personalization
  - Tailored Responses: Adapt responses based on user preferences or previous interactions
- Code Understanding and Generation
  - Code Analysis: Understand and analyze code snippets
  - Code Generation: Generate code based on natural language descriptions
- Knowledge Retrieval
  - Fact-Based Retrieval: Access and retrieve factual information from a vast knowledge base
- Multi-Modal Capabilities
  - Image and Text: Some LLMs can process both images and text, allowing for more complex interactions (e.g., describing an image)
- Data Augmentation
  - Synthetic Data Generation: Create synthetic datasets for training other machine learning models

History

- 1950s-1990s: Early NLP
  - Rule-based systems and statistical methods
  - Limited by computing power and rigid rules
  - Focus on specific tasks like translation
- 2000s: Neural Networks
  - Introduction of neural networks for language tasks
  - Word2Vec (2013) revolutionized word representations
  - RNNs and LSTMs improved sequence processing
- 2017: Transformer Architecture
  - Paper "Attention is All You Need" introduced transformers
  - Enabled parallel processing and better long-range dependencies
  - Set foundation for modern LLMs
- 2018-2020: First Large Models
  - BERT (2018) by Google
  - GPT series by OpenAI
  - Demonstrated benefits of scale
  - Billions of parameters
- 2021-Present: Rapid Evolution
  - GPT-3 showed emergent abilities
  - Models like PaLM, Claude, and GPT-4 advanced capabilities
  - Multimodal models emerged
  - Focus on alignment and safety
  - Open-source models gained prominence
  - The field continues to advance with improvements in efficiency, capabilities, and responsible development practices

Fundamental Concepts

- Neural Networks & Deep Learning
  - Multi-layer networks process input text
  - Learns patterns through weighted connections
  - Uses backpropagation for training
- Transformer Architecture
  - Self-attention mechanisms capture relationships
  - Parallel processing of sequences
  - Positional encoding maintains word order
- Tokenization
  - Breaks text into manageable tokens
  - Can be words, subwords, or characters
  - Vocabulary size affects model capabilities
- Training Objectives
  - Masked language modelling
  - Next token prediction
  - Learns patterns from massive text datasets
- Context Windows
  - Limited sequence length for processing
  - Typically 2k-32k tokens
  - Affects model's understanding scope
- Parameters
  - Learned weights during training
  - More parameters = more capacity
  - Billions to trillions in modern models

Training Process

- **Pre-training**
  - Massive text datasets (100s of TB)
  - Self-supervised learning
  - Masked language modeling
  - Next token prediction
  - Distributed across GPU/TPU clusters
- **Training Loop**
  - Forward pass: Generate predictions
  - Loss calculation: Compare with actual tokens
  - Backward pass: Calculate gradients
  - Parameter updates: Adam optimizer
  - Learning rate scheduling
- **Optimization Techniques**
  - Mixed precision training
  - Gradient accumulation
  - Checkpoint saving
  - Data parallelism
  - Model parallelism
- **Fine-tuning**
  - Specialized datasets
  - Task-specific objectives
  - Lower learning rates
  - Fewer epochs
  - Parameter-efficient methods
- **Evaluation**
  - Perplexity metrics
  - Task benchmarks
  - Human evaluation
  - Safety testing
  - Bias assessment
- **Challenges**
  - Computational costs ($1M-100M+)
  - Training instability
  - Hardware requirements
  - Data quality control
  - Memory constraints

Dataset Requirements

- Quality Requirements
  - Clean, high-quality text
  - Diverse content types/ domains
  - Multiple languages
  - Minimal noise/errors
  - Proper formatting
- Scale Requirements
  - Hundreds of terabytes
  - Billions of tokens
  - Varied sources (books, articles, code)
  - Balanced topic distribution
  - Multiple languages representation
- Technical Specifications
  - Structured format (JSON/TFRecord)
  - Consistent encoding (UTF-8)
  - Deduplication
  - Version control
  - Data validation pipelines
- Data Processing
  - Cleaning scripts
  - Quality filters
  - Format standardization
  - Tokenization
  - Shuffling
- Legal/ Ethical Requirements
  - Copyright compliance
  - Personal data removal
  - Bias mitigation
  - Content filtering
  - Attribution tracking

Limitations of LLM

- Bias and Fairness
  - Inherent Biases: LLMs can reflect and amplify biases present in the training data, leading to unfair or discriminatory outputs
  - Stereotyping: They might reinforce stereotypes related to gender, race, or other social categories
- Factual Accuracy
  - Hallucinations: LLMs may generate information that is incorrect, misleading, or entirely fabricated
  - Lack of Verification: They cannot independently verify facts, leading to potential misinformation
- Contextual Understanding
  - Limited Contextual Awareness: LLMs do not possess true understanding or common sense reasoning, which can result in nonsensical or contextually inappropriate responses
  - Memory Limitations: They often have a limited context window, making it difficult to retain long-term context in conversations or documents
- Dependence on Training Data
  - Data Quality: Performance is heavily reliant on the quality and diversity of the training data. Poor or biased data can lead to subpar outputs
  - Staleness: LLMs are trained on datasets that may not include the most current information, resulting in outdated knowledge
- Ethical Concerns
  - Manipulation Potential: They can be used to generate misleading content or propaganda
  - Privacy Issues: If trained on sensitive data, there’s a risk of inadvertently revealing personal information

Limitation of LLM

- Lack of Personalization
  - Generic Responses: LLMs may produce generic answers that do not cater to individual user preferences or needs
  - Limited Adaptability: Adapting to specific user contexts or evolving needs can be challenging without further fine-tuning
- Resource Intensive
  - Computational Costs: Training and running LLMs require significant computational resources, which can be costly and environmentally taxing
  - Access Barriers: Not all organizations have the resources to fine-tune or deploy LLMs effectively
- Inability to Perform Tasks Requiring Real-World Interaction
  - No Physical Interaction: LLMs cannot interact with the physical world or perform tasks requiring sensory input or physical presence
  - Static Knowledge: They cannot learn or adapt in real time; their knowledge base remains fixed until re-trained
- Legal and Compliance Issues
  - Intellectual Property: The generation of content that resembles existing works can raise copyright concerns
  - Regulatory Compliance: Ensuring that outputs comply with laws and regulations (like GDPR) can be complex
- Overfitting in Fine-Tuning
  - Task-Specific Limitations: When fine-tuned on a small dataset, LLMs can overfit, reducing their generalization ability on broader tasks

Types of LLMs

- **Transformer-based LLMs**
  - **BERT** (Bidirectional Encoder Representations from Transformers)
    - Example tasks: Question answering, sentiment analysis, named entity recognition
    - Pre-training objectives: Masked language modeling, next sentence prediction
  - **RoBERTa** (Robustly Optimized BERT Pretraining Approach)
    - Example tasks: Text classification, question answering, sentiment analysis
    - Pre-training objectives: Masked language modeling, next sentence prediction with a slightly different approach
  - **DistilBERT** (Distilled BERT)
    - Example tasks: Sentiment analysis, text classification, named entity recognition
    - Pre-training objectives: Distilling the knowledge of a pre-trained model into a smaller one
- **Recurrent Neural Network (RNN)-based LLMs**
  - **LSTM-based LLMs**
    - Example tasks: Language modeling, machine translation, text generation
    - Pre-training objectives: Language modeling with an LSTM encoder-decoder architecture
  - **GRU-based LLMs**
    - Example tasks: Sentiment analysis, topic modeling, language understanding
    - Pre-training objectives: Language modeling with a GRU encoder-decoder architecture

Types of LLMs

- **Convolutional Neural Network (CNN)-based LLMs**
  - **Character-CNN LLMs**
    - Example tasks: Text classification, sentiment analysis, named entity recognition
    - Pre-training objectives: Character-level language modeling with a CNN architecture
- **Hybrid LLMs:**
  - **Transformer-RNN hybrids**
    - Example tasks: Language modeling, machine translation, text generation
    - Pre-training objectives: Combining the strengths of transformer and RNN architectures for language modeling tasks
- **Specialized LLMs:**
  - **Question Answering (QA) LLMs**
    - Example tasks: QA systems, information retrieval
    - Pre-training objectives: Training on a large corpus of question-answer pairs to predict answers based on context
  - **Summarization LLMs**
    - Example tasks: Text summarization, document summarization
    - Pre-training objectives: Training on a large corpus of text and summaries to generate concise summaries

Types of LLMs

- **Task-specific LLMs**
  - **Named Entity Recognition (NER) LLMs**
    - Example tasks: NER, information extraction
    - Pre-training objectives: Training on a large corpus of text to identify specific entities such as names, locations, and organizations
  - **Part-of-Speech (POS) tagging LLMs**
    - Example tasks: POS tagging, language understanding
    - Pre-training objectives: Training on a large corpus of text to predict the part of speech for each word in a sentence

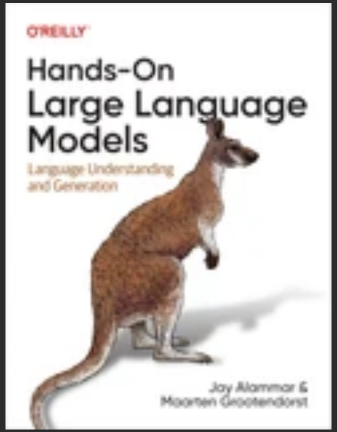
LLMs

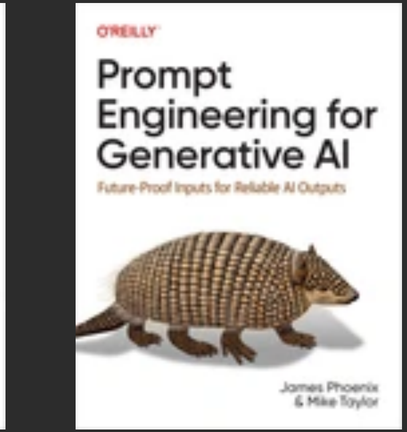
Model	Pre-training Objectives	Parameters	Capabilities
BERT	Masked language modelling	110,000,000	Text classification, sentiment analysis, question answering
RoBERTa	Masked language modelling	125,000,000	Sentiment analysis, question answering, named entity recognition
GPT-3	Masked language modelling	175,000,000,000	Text generation, conversation AI, text classification
T5	Multiple masked language modelling and next sentence prediction	220,000,000	Text-to-text generation, question answering, summarization
Deberta	Multiple masked language modelling and next sentence prediction	125,000,000	Sentiment analysis, question answering, named entity recognition
Flaubert	Multiple masked language modelling and next sentence prediction	10,000,000,000	Text generation, conversation AI, text classification
Claude	Multimodal pre-training	1,500,000,000	Visual question answering, multimodal sentiment analysis
Groq	Transformer-RNN hybrid	10,000,000,000	Text generation, conversation AI, text classification
LLaMA	Masked language modelling and next sentence prediction	13,000,000,000	Text-to-text generation, question answering, summarization

LLMs

- DeepSeek
- GPT
- Gemini
- Claude 3.5 Sonnet
- BLOOM
- Cohere
- Falcon
- Ernie
- MPT
- Grok
- Qwen
- LlaMA 3.3
- Mistral AI
- Qwen2.5-Max
- Gemma 2.0 Flash
- Doubao
- Janus-Pro-7B
- Imagen 3

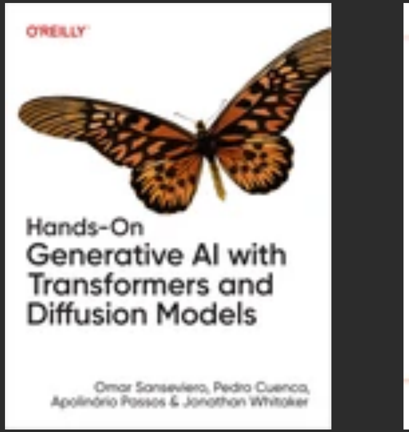
Books

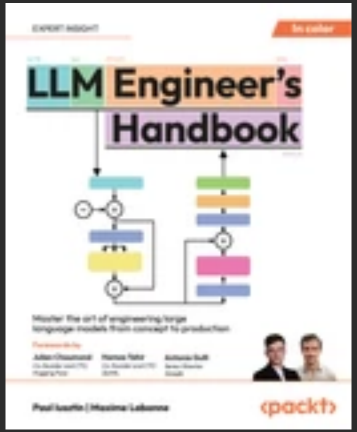
✓

O'REILLY
Hands-On
Large Language
Models
Language Understanding
and Generation
Jay Alammar &
Maarten Grootendorst

✓

O'REILLY
Prompt
Engineering for
Generative AI
Future-Proof Inputs for Reliable AI Outputs
James Phoenix
& Mike Taylor

✓

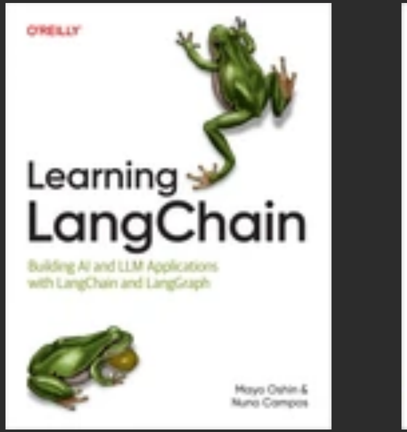
O'REILLY
AI Engineering
Building Applications
with Foundation Models
Chip Huyen

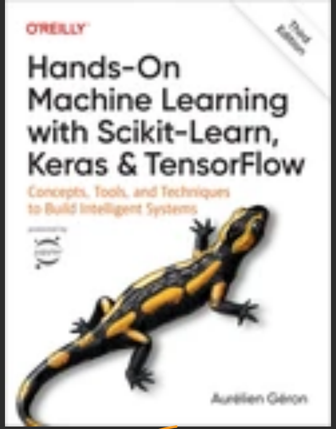
✓

O'REILLY
Hands-On
Generative AI with
Transformers and
Diffusion Models
Omar Sanseviero, Pedro Cuenca,
Apolinário Passos & Jonathan Whitaker

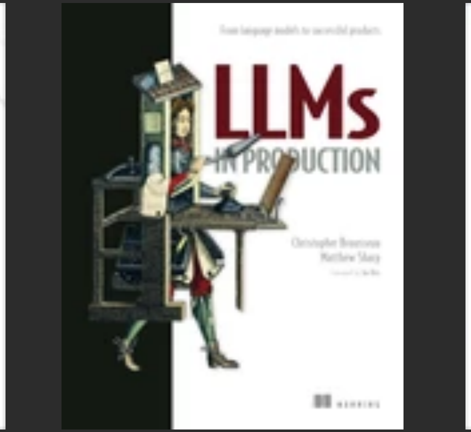
✓

EXPERT INSIGHT
In color
LLM Engineer's
Handbook
Paul Iusztin | Maxime Labonne
‹packt›

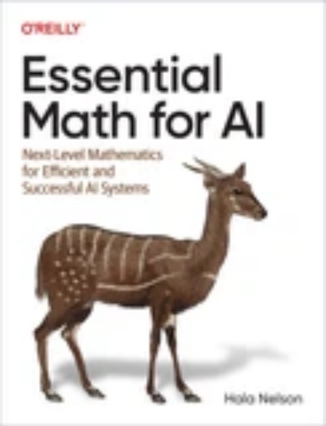
✓

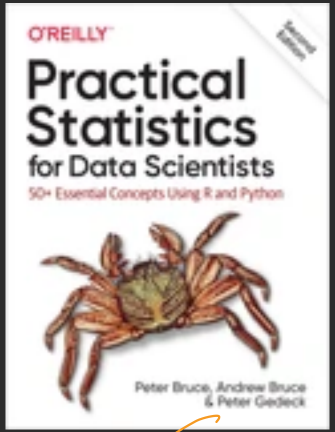
Generative AI
IN ACTION
MANNING

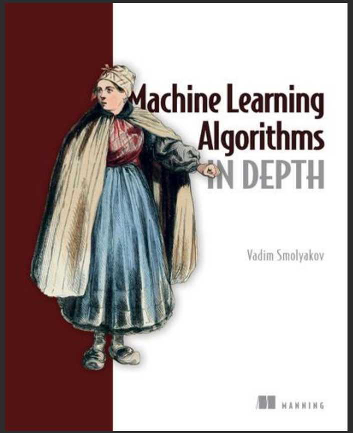

O'REILLY
Learning
LangChain
Building AI and LLM Applications
with LangChain and LangGraph
Mayo Oshin &
Nuno Campos


O'REILLY
Third Edition
Hands-On
Machine Learning
with Scikit-Learn,
Keras & TensorFlow
Concepts, Tools, and Techniques
to Build Intelligent Systems
Aurélien Géron
✓


LLMs
IN PRODUCTION
Christopher Brousseau
Matthew Sharp
MANNING


O'REILLY
Essential
Math for AI
Next-Level Mathematics
for Efficient and
Successful AI Systems
Hala Nelson


O'REILLY
Second Edition
Practical
Statistics
for Data Scientists
50+ Essential Concepts Using R and Python
Peter Bruce, Andrew Bruce
& Peter Gedeck
✓


Machine Learning
Algorithms
IN DEPTH
Vadim Smolyakov
MANNING