
PLURALSIGHT

GenAI – The Basic

Welcome!

Tarek Atwan
Instructor, Pluralsight

About Me:

- Book Author
- 17+ Years Consulting Services
- 4+ Years Instructor
- 2 Startups
- World Traveler
- Gym Rat

Student Instructions

- Job title? Location?
- Why did you pick this course?
- What are your expectations from this course?
- What is your related experience, if any?
- Fun fact?

Prerequisites

This course assumes you

- Know how to program in some other language
- Some familiarity with Python Programming
- No prior experience or knowledge in Machine Learning, Deep Learning, or Generative AI

How we're going to work together

- You'll have a copy of all the course materials through GitHub
  - We'll be using Google Colab (explained shortly)
- You'll be following along in the notebook and..
  - doing coding exercises/labs inside the notebook as well

What is Generative AI

> Current generative AI and other technologies have the potential to automate work activities that absorb 60% to 70% of employee's time today

Economic potential of generative AI, June 2023

McKinsey & Company

Half of today's work activities could be automated between 2030 and 2060

Economic potential of generative AI, June 2023

McKinsey & Company

Generative AI Demo Time

OpenAI ChatGPT and GPT4

- Text to Text Generation
- Text to Image Generation
- Text to Code Generation

What is Generative AI?

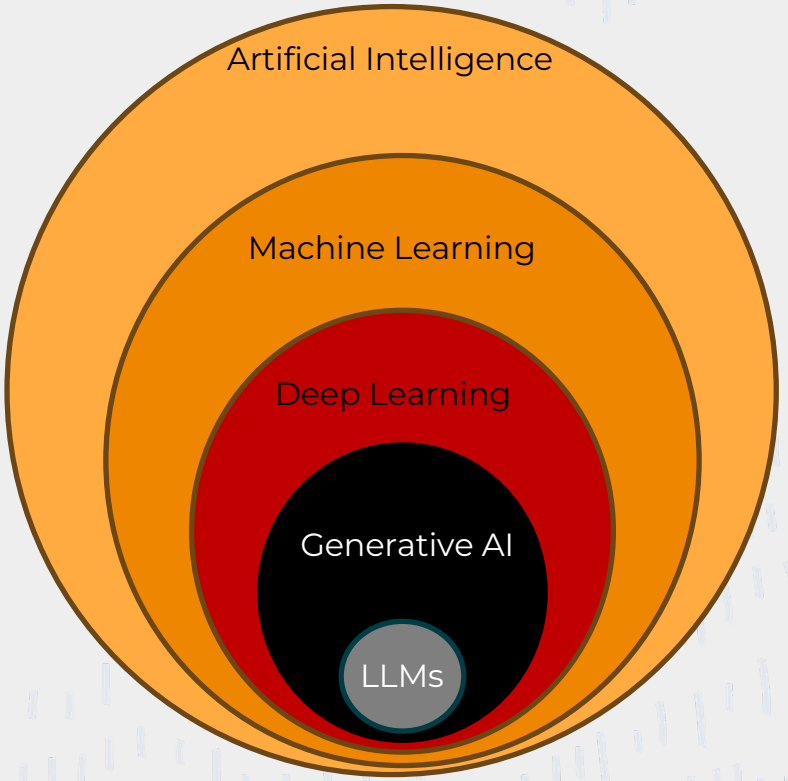
AI refers to the broad concept of machines or computers performing tasks that typically require human intelligence. This includes reasoning, learning, problem-solving, perception, language understanding, etc.

ML is a subset of AI focused on the idea that machines can learn from data, identify patterns, and make decisions with minimal human intervention

DL is a subset of ML that uses neural networks with many layers (deep networks) to model complex patterns in data.

Generative AI refers to a class of AI, often realized through DL, that focuses on generating new content or data that is similar to but distinct from the training data.

LLMs are a type of deep learning model designed to understand, generate, and interact with human language at a large scale. They are trained on vast amounts of text data.

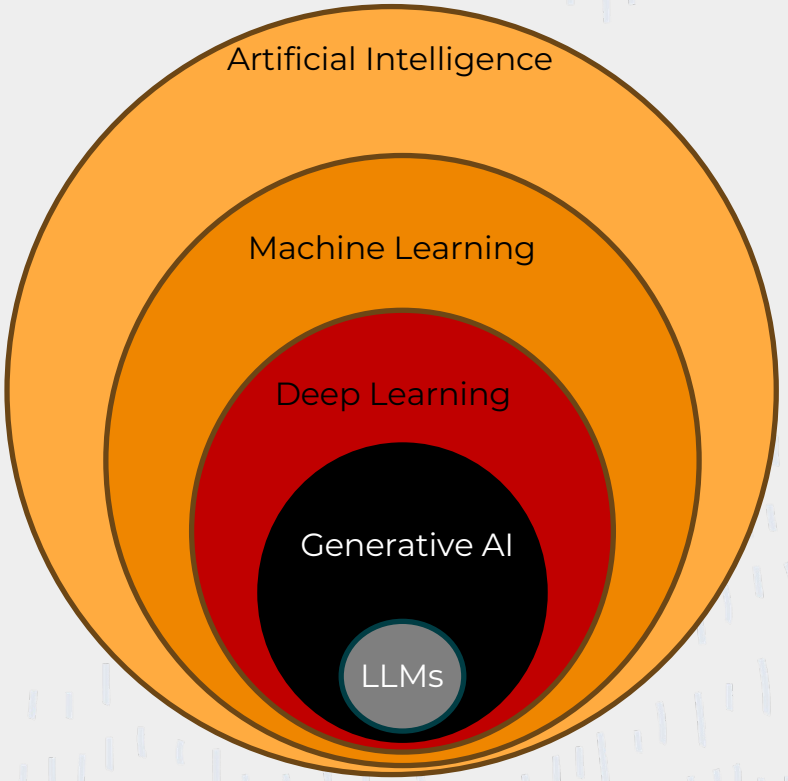


What is Generative AI?

Generative AI refers to a subset of artificial intelligence where the primary goal is to create or generate new data that is similar but not identical to the training data. It's about models that can learn from existing data to **generate** new, **unseen** data or patterns that maintain a **statistical resemblance** to the original dataset

These models are capable of understanding and replicating complex data distributions, allowing them to **produce** highly realistic and diverse outputs.

It evolved from ML/DL and is essential in fields like **content creation** and **data augmentation**. It includes models like **Generative Adversarial Networks** (GANs) and **Large Language Models** (LLMs)



Discriminative AI vs Generative AI

Discriminative AI

- Discriminative models learn the **conditional probability distribution P(Y|X)** They focus on understanding the **boundary** between different classes in the data, essentially distinguishing between different types of data inputs.

Generative AI

- Generative models are designed to learn the **joint probability distribution P(X,Y)** of inputs X and outputs Y Their goal is to **understand and replicate** the way data is generated, enabling them to produce new data instances that are similar to the training data.

Discriminative AI vs Generative AI (Examples)

Discriminative AI

1. **Convolutional Neural Networks (CNNs)**: Used for image classification.
2. **Recurrent Neural Networks (RNNs)**: Common in speech recognition and natural language processing.
3. **Support Vector Machines (SVMs), Logistic Regression, etc.**: Traditional ML algorithms for classification tasks.

Generative AI

1. **Generative Adversarial Networks (GANs)**: Used for generating realistic images, artworks, etc.
2. **Variational Autoencoders (VAEs)**: Often used in image generation and denoising.
3. **Language Models like GPT (Generative Pre-trained Transformer)**: Used for generating coherent and contextually relevant text.

Discriminative AI vs Generative AI (Objective)

Discriminative AI

A discriminative AI and its algorithms can be used to:

- Differentiate
- Classify
- Identify Patterns
- And Draw Conclusions
- Example: Email spam filters
- They are best applied to classification tasks.

Generative AI

A generative AI can generate new content/output as:

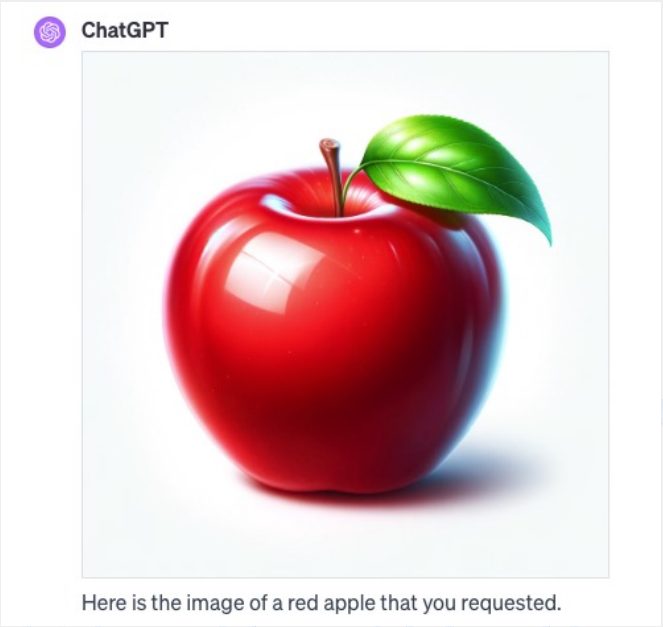
- Text
- Images
- Audio
- Video
- Code
- And new data

Discriminative AI vs Generative AI

Discriminative AI

Is the image an Orange or an Apple?

Generative AI



I want an image of a Red Apple.

Building Blocks of Generative AI

- Generative Adversarial Networks (GANs)
- Variational Autoencoders (VAEs)
- Transformers
- Diffusion Models

Code Along Examples

Generative AI Potential

Generative AI for Automation

- Generative AI will not replace your Job or automate your Job
- Rather, Generative AI automates tasks
- A Job will involve a large number of tasks that can be automated
- Think of Generative AI as a Co-Pilot (Your assistant)

Generative AI for Automation

- Generative AI will not replace your Job or automate your Job
- Rather, Generative AI automates tasks
- A Job will involve a large number of tasks that can be automated
- Not every task can be fully automated. An analysis needs to be done on which tasks are good candidates for Generative AI automation
- Think of Generative AI as a Co-Pilot (Your assistant)

Generative AI Automation vs Augmentation

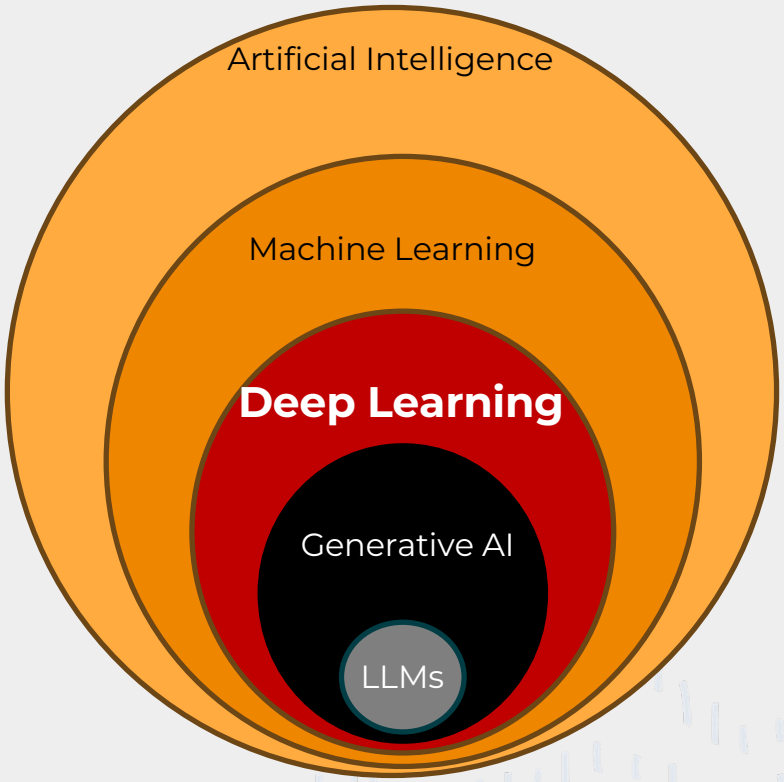
- In some tasks and businesses, you will start with augmentation, and then move toward automation.

- **Augmentation**: generative AI is used to augment (help/support/enhance) human capabilities. It enhances the quality and efficiency of tasks performed by humans.
- **Automation**: generative AI to **fully automate** certain tasks or processes. The AI system takes over the entire function, performing it from start to finish without the need for human intervention

Technical Dive

Artificial Neural Networks (ANN)

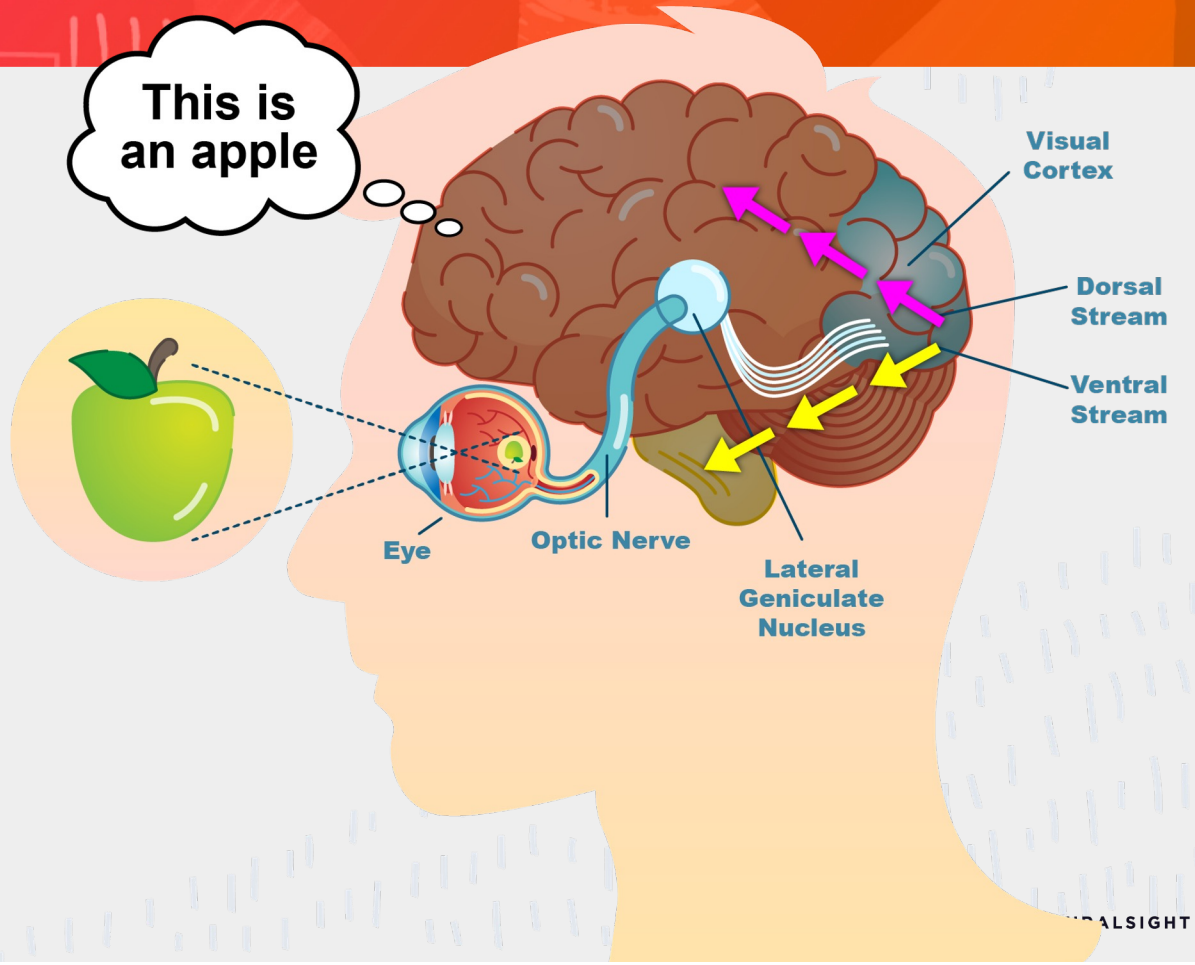
What is Generative AI?



Neural Networks

How our brain works:

In order to recognize an image, our brain uses thousands
of neuron connections to find a match between the visual input and a mental representation of an object.

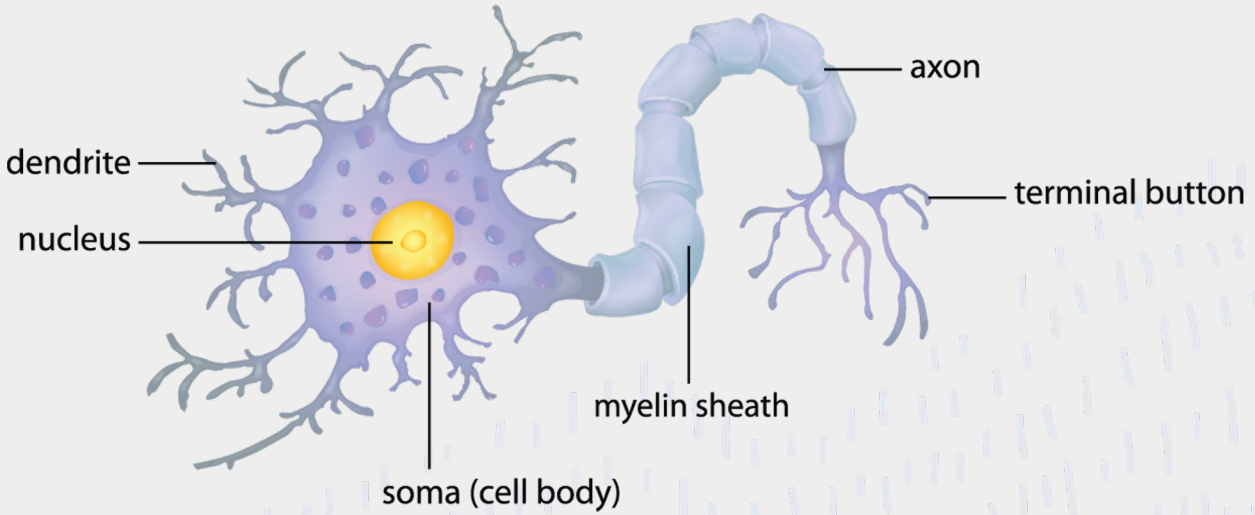


Neural Networks

The ability of the brain to process information and make predictions or interpretations is what inspired neurophysiologists and mathematicians to start the development of artificial neural networks (ANN).

In the same way that biological neurons receive input signals through the dendrites, an ANN receives input variables and processes them by using an activation function.

The output of an ANN is similar to the neuron nucleus in the brain.



History of Neural Networks

1943

Neurophysiologist **Warren McCulloch** and mathematician **Walter Pitts** wrote a paper on how neurons might work.

1949

Donald Hebb wrote *The Organization of Behavior*, which pointed out the fact that neural pathways are strengthened each time they are used.

1959

Bernard Widrow and **Marcian Hoff** of Stanford developed models called ADALINE and MADALINE.

1962

Widrow and **Hoff** developed a learning procedure that examines the value before the weight adjusts it (i.e., 0 or 1) according to the rule: Weight Change = (Pre-Weight line value).

1972

Teuvo Kohonen and **James A. Anderson** each developed a similar network independently of one another. They both used matrix mathematics to describe their ideas but did not realize that what they were doing was creating an array of analog ADALINE circuits.

History of Neural Networks

1982

John Hopfield of Caltech presented a paper to the National Academy of Sciences. His approach was to create more useful machines by using bidirectional lines. Previously, the connections between neurons was only one way.

1982

Joint US-Japan conference on **Cooperative/ Competitive Neural Networks**. Japan announced a new Fifth Generation effort on neural networks, and US papers generated worry that the US could be left behind in the field.

1986

Three independent groups of researchers, including **David Rumelhart**, a former member of Stanford's psychology department, came up with similar ideas which are now called back propagation networks.

1997

A recurrent neural network framework, LSTM was proposed by **Jürgen Schmidhuber** and **Sepp Hochreiter**.

2000s

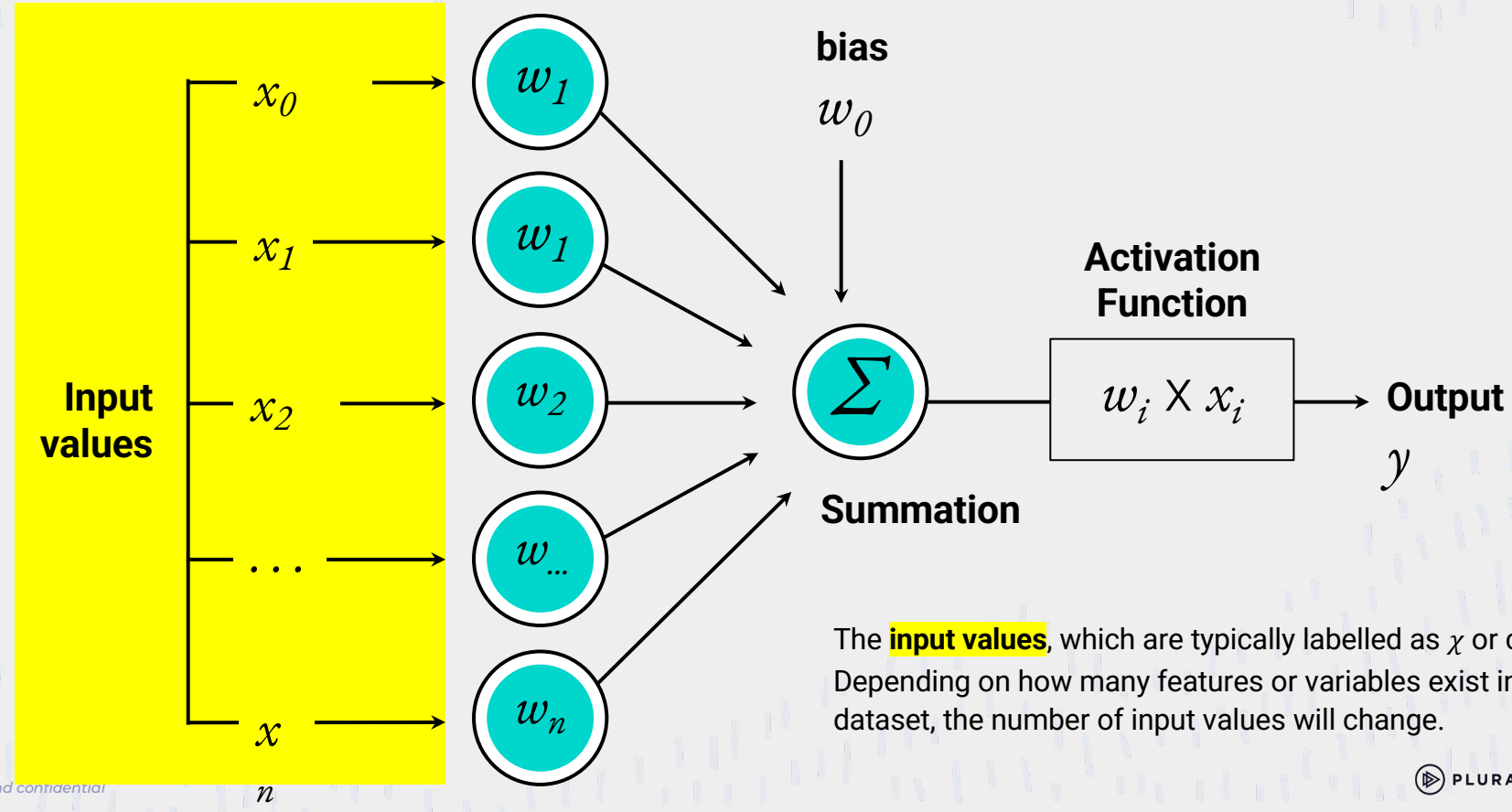
Transformers were introduced. Followed by **GANs**, **VAEs**, and **Autoregressive** models which pushed the boundaries of **Generative AI**.

Generative AI Breakthrough

2010	2014	2017-2022	2022
Near-Perfect Translation of Natural Language	**Mastering the Meaning of Words**	**Large Language Foundation Models**	**Conversational Large Language Foundation Models**
Around 2010, AI researchers working on natural language translation discovered that models exposed to vast amounts of text produced much better results than models using top-down grammatical rules.	In 2014, language models began to make sense of the meaning of words in a natural language by analyzing the context in which the word appeared.	Advances made from 2017 to 2022 resulted in language models that can serve as a foundation for customization. Creating foundation models is cost-prohibitive, but once created, they can be customized using a small amount of additional data to achieve state-of-the-art performance on new tasks without significant investment.	2022 marked the arrival of ChatGPT, which gave users a simple way to access a large language foundational model.The brilliance of ChatGPT is not just in the incredibly advanced model at its core; equally, it is the ability to tap into this model by conversing with it in natural language. As AI researcher Andrej Karpathy quips, "Now the hottest programing language is English!"

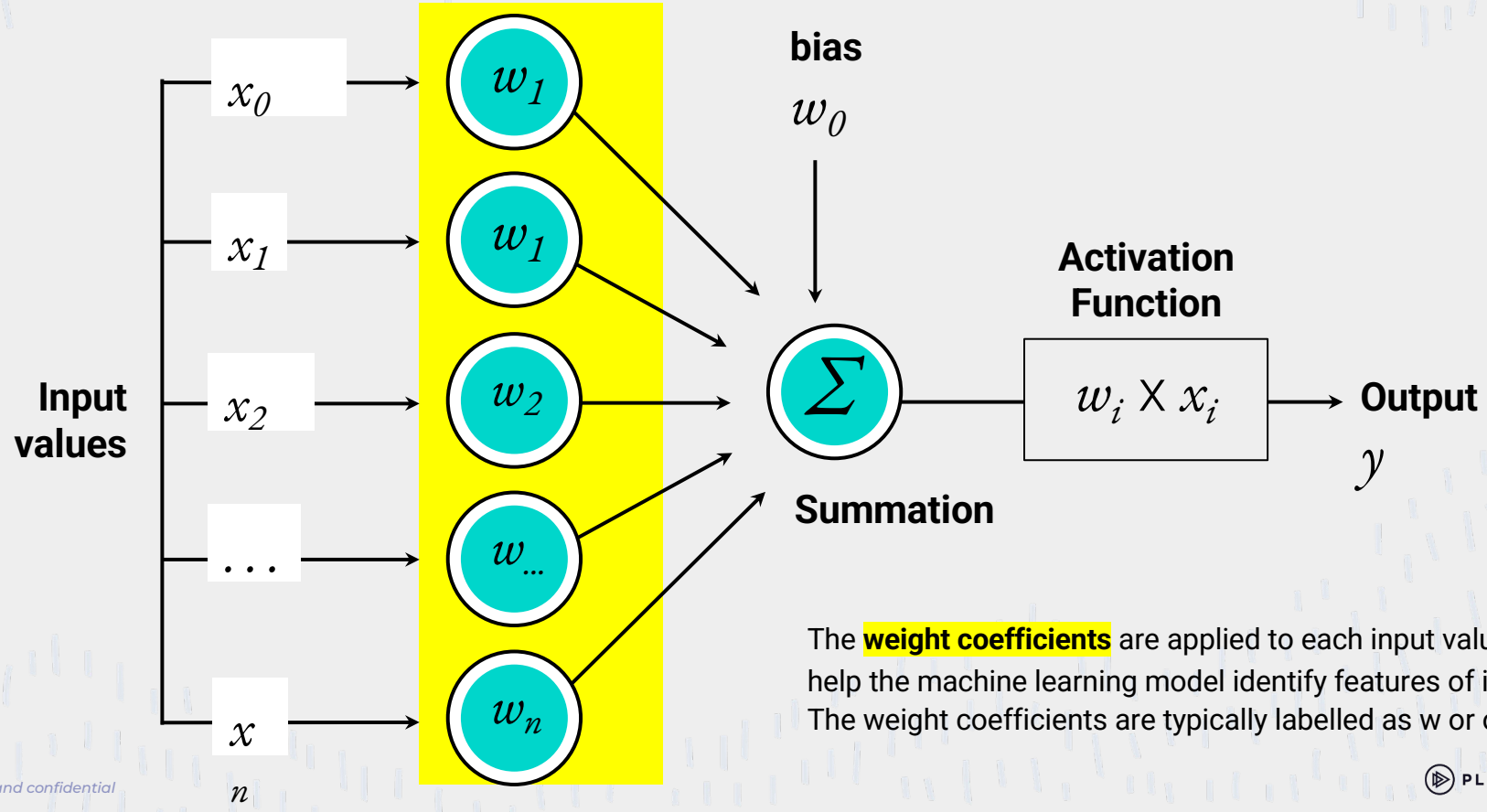
The Perceptron

Perceptron – Input Values (Shown as X's)



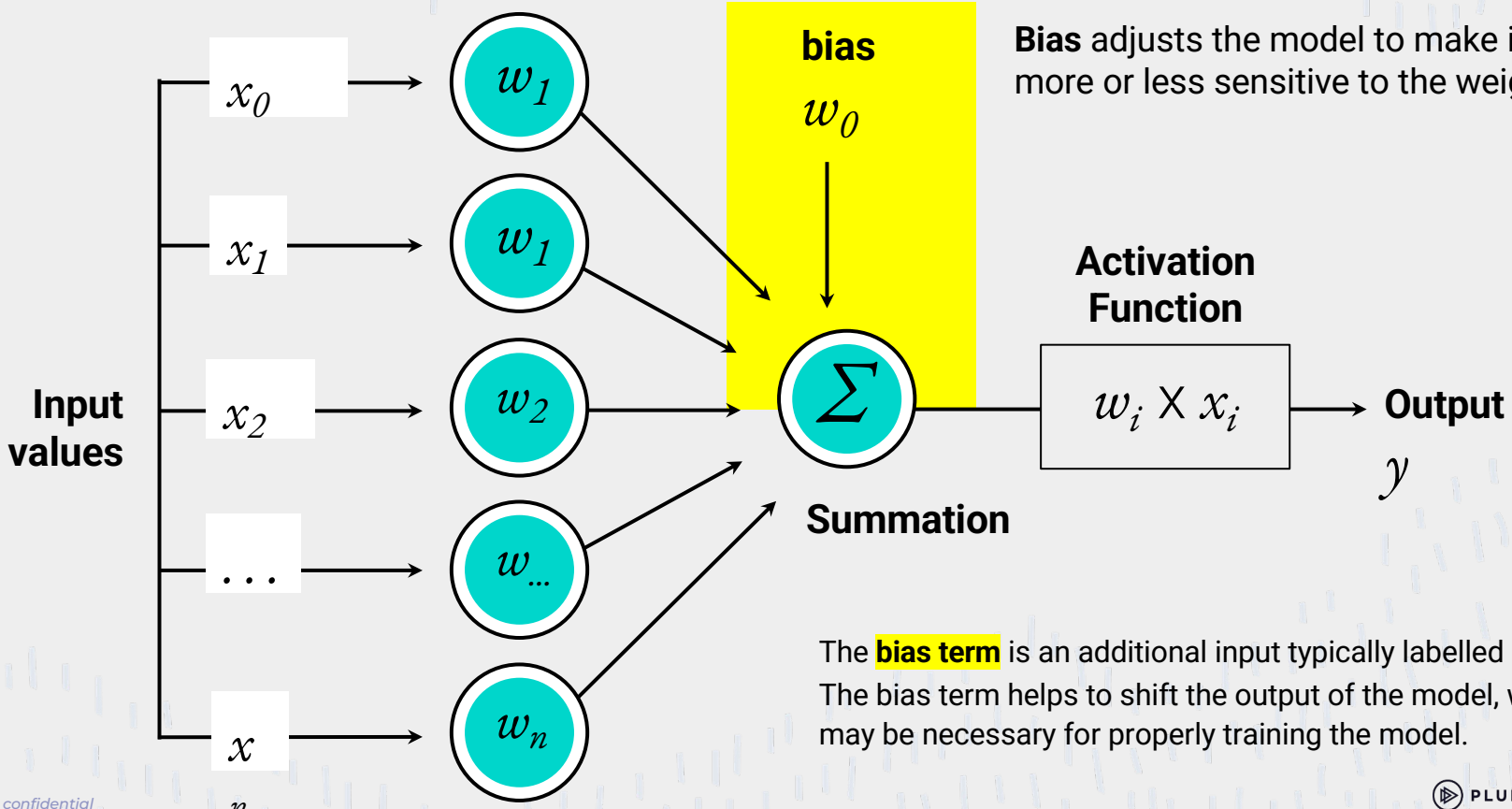
The **input values**, which are typically labelled as χ or chi. Depending on how many features or variables exist in the dataset, the number of input values will change.

Perceptron – Weight Coefficients (denoted as w's)



The **weight coefficients** are applied to each input value to help the machine learning model identify features of interest. The weight coefficients are typically labelled as w or omega.

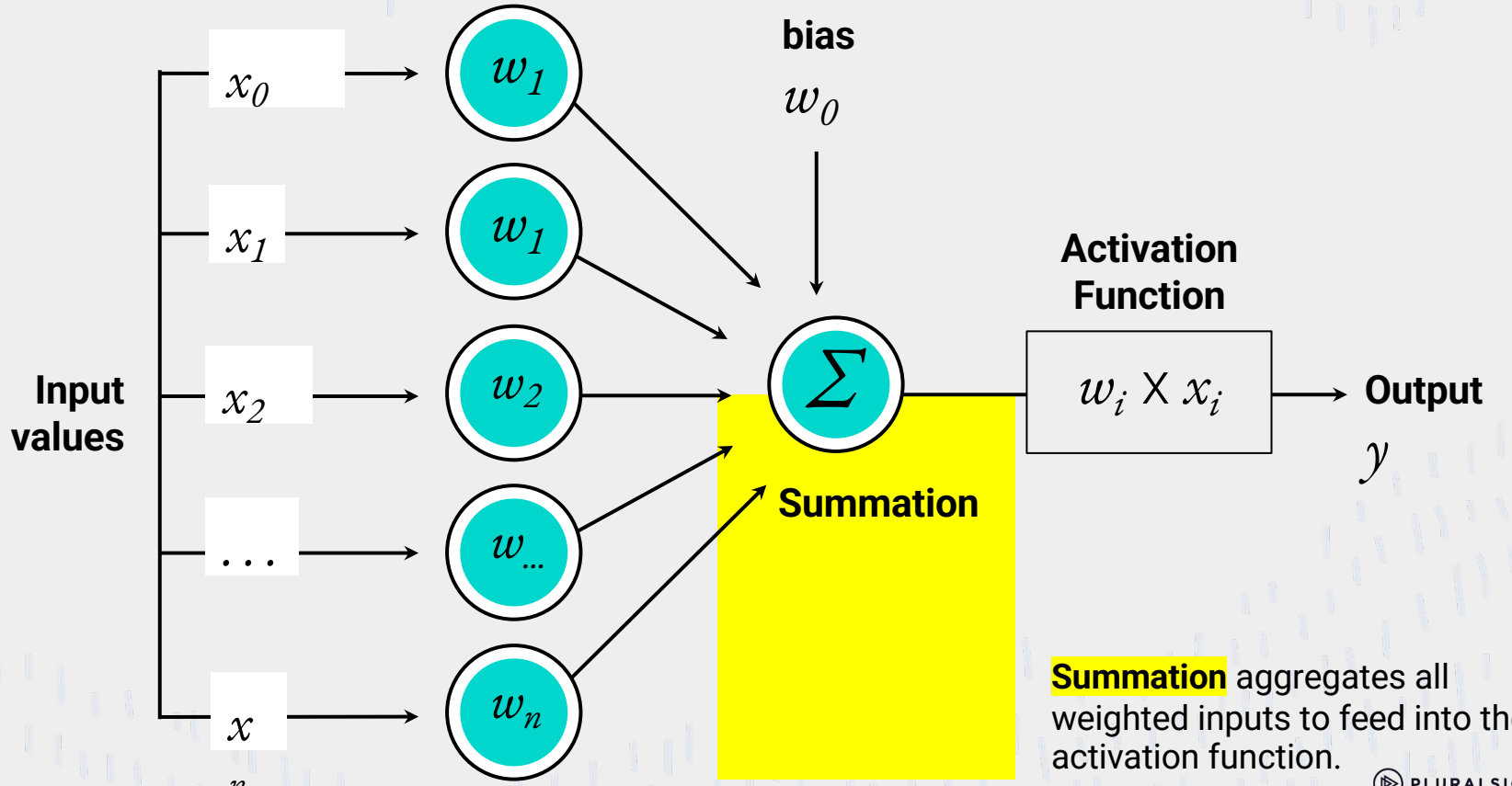
Perceptron – A Constant Value Called Bias



Bias adjusts the model to make it either more or less sensitive to the weights.

The **bias term** is an additional input typically labelled as ω. The bias term helps to shift the output of the model, which may be necessary for properly training the model.

Perceptron – Net Summary Function (the Summation)

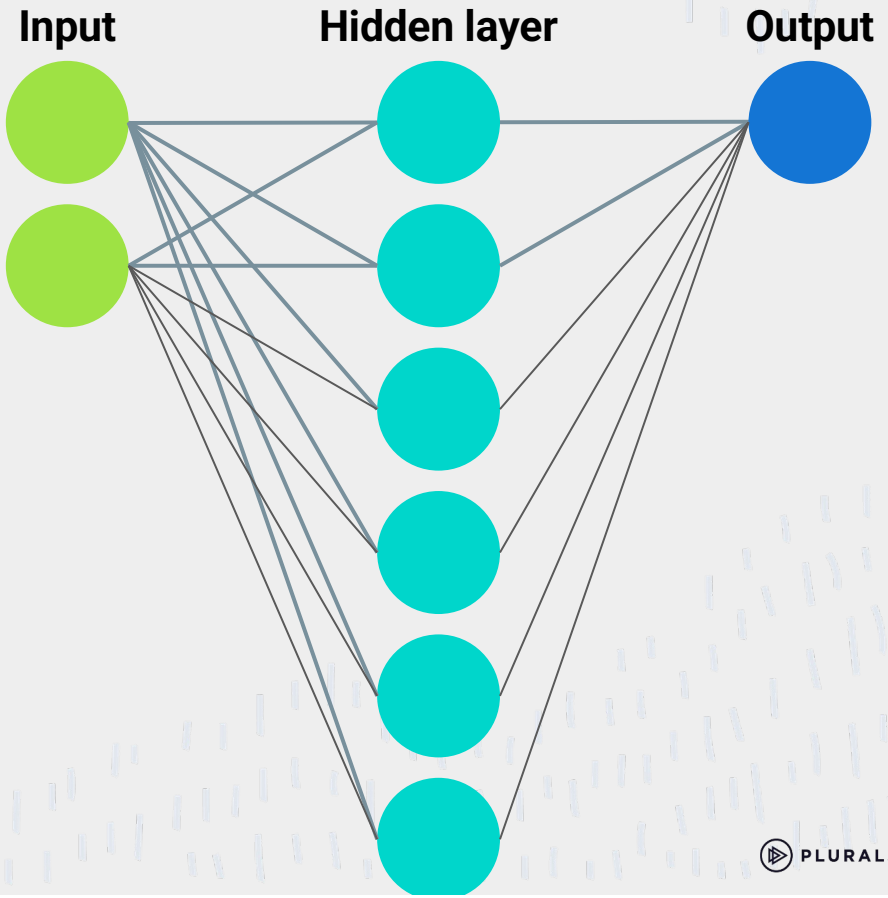


Summation aggregates all weighted inputs to feed into the activation function.

Deep ANN

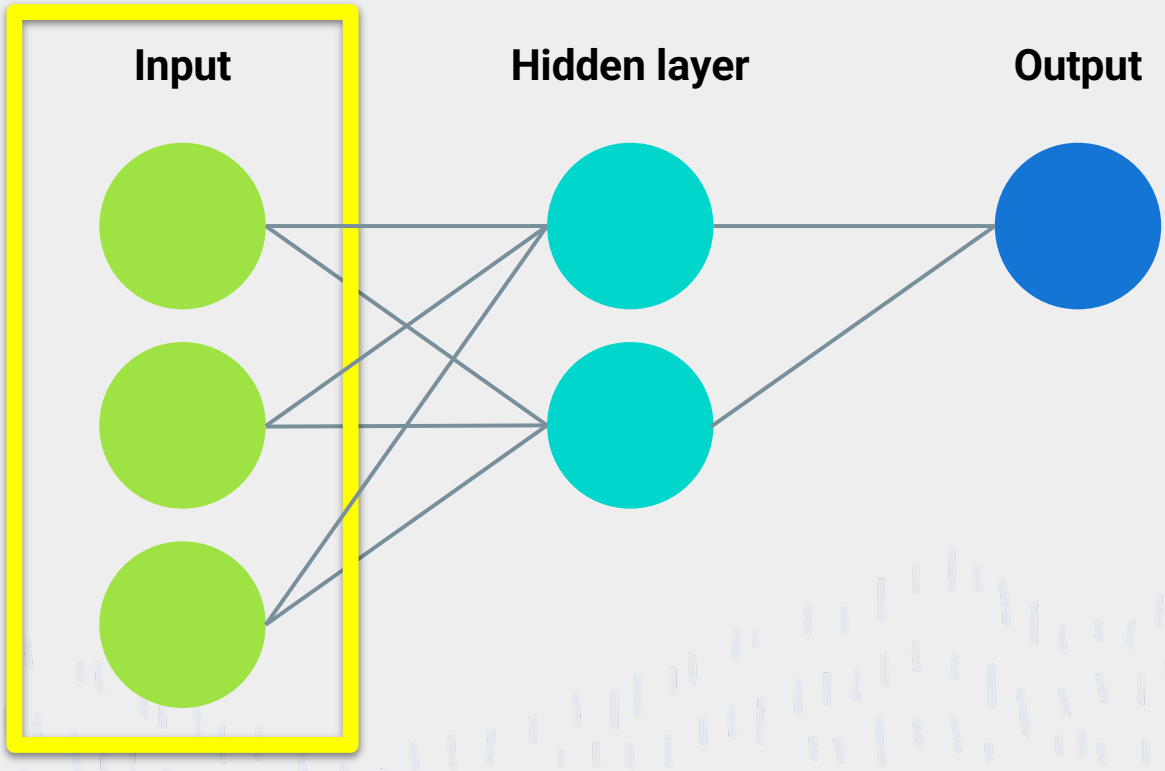
The Neural Network

A modern neural network model is a structure composed of several connected perceptrons that learn from input data to produce an output.



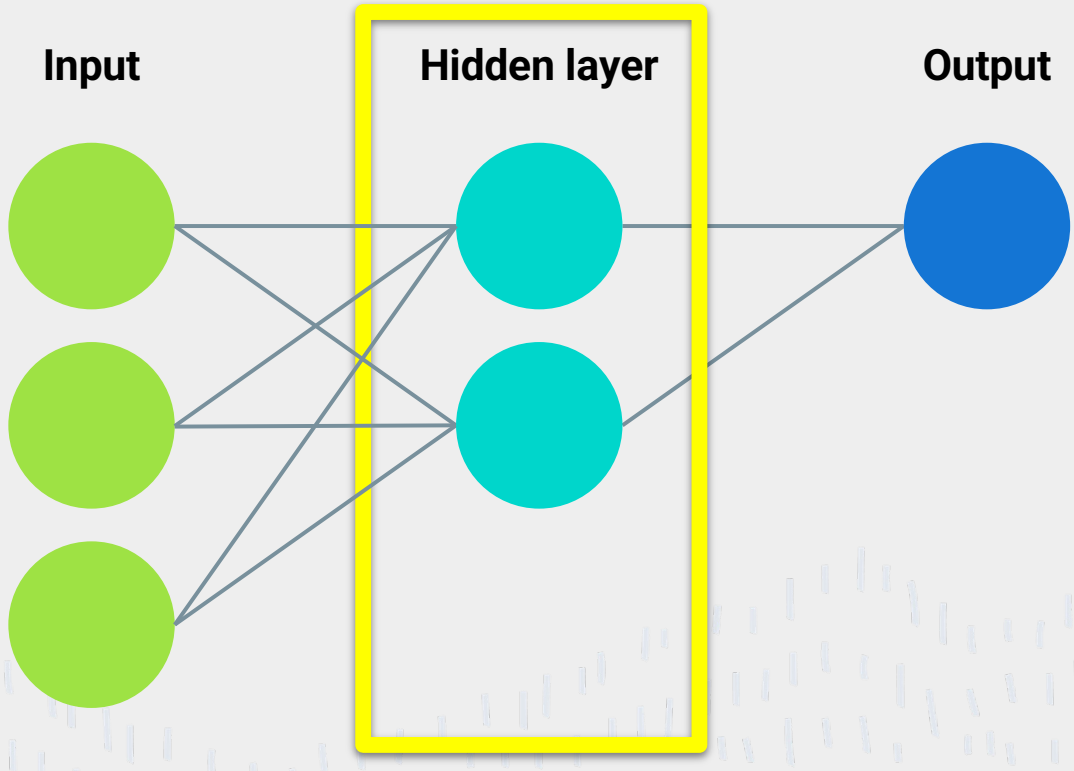
The Neural Network

An **input layer** of input values transformed by weight coefficients



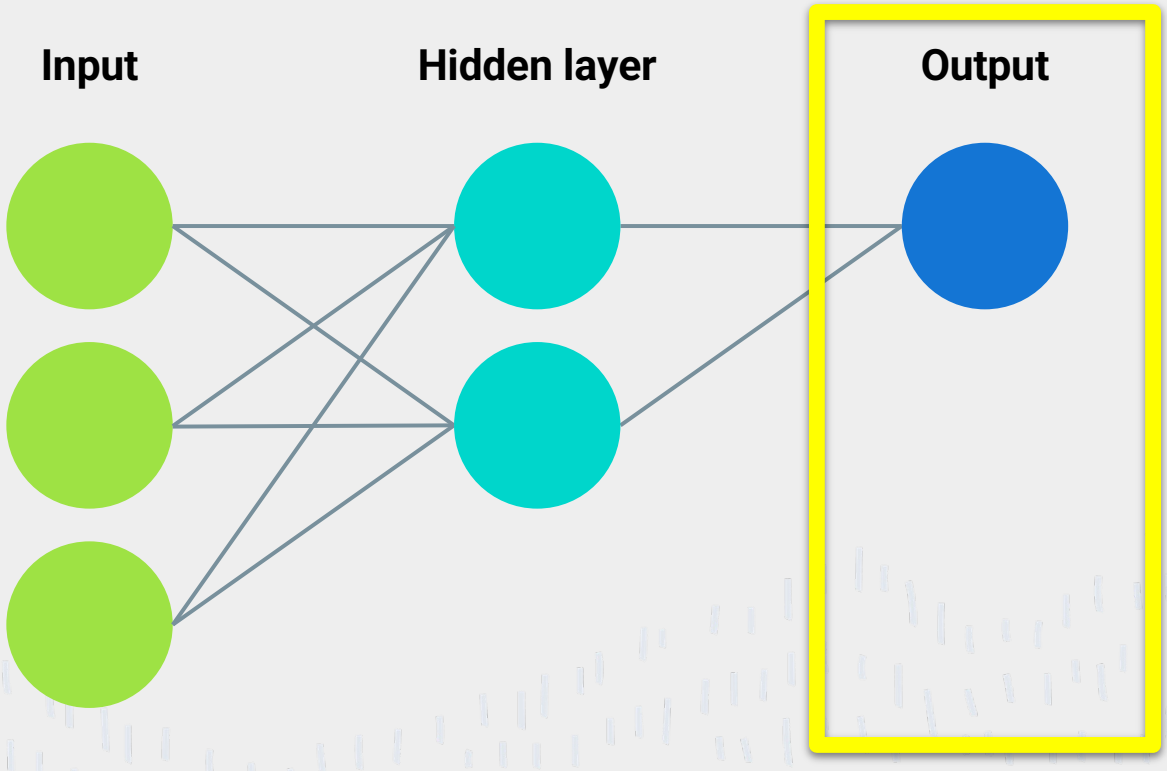
The Neural Network

A single **hidden layer** that can contain a single neuron or multiple neurons



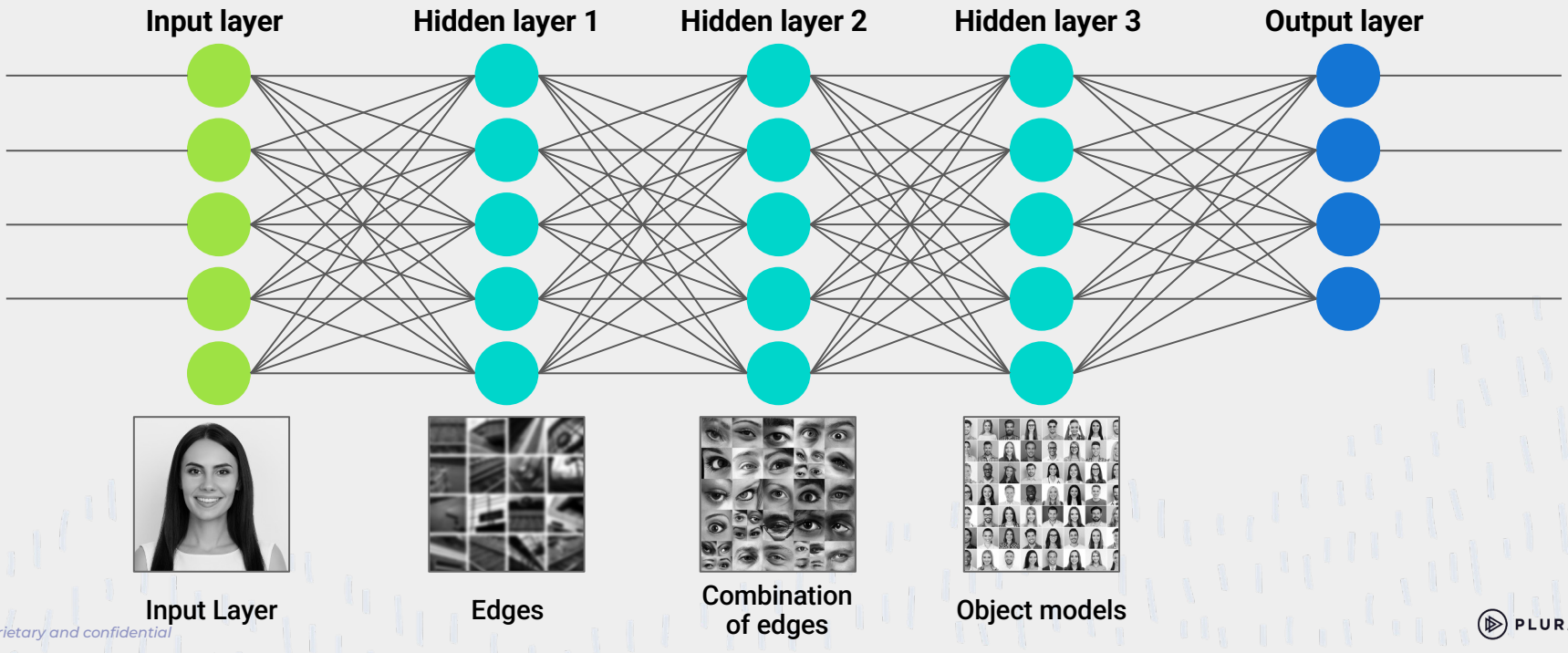
The Neural Network

An **output layer** that reports the outcome of the value



Deep Neural Network

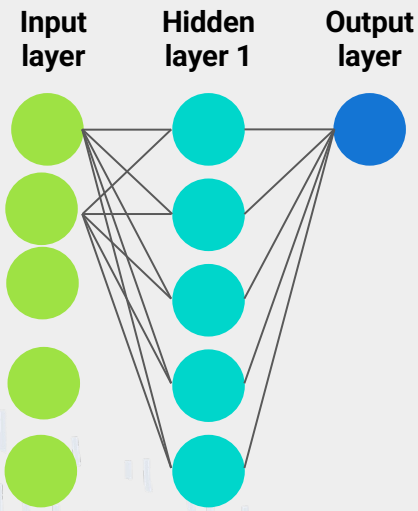
In image recognition, each layer can identify different image features in the process of defining or identifying the image.



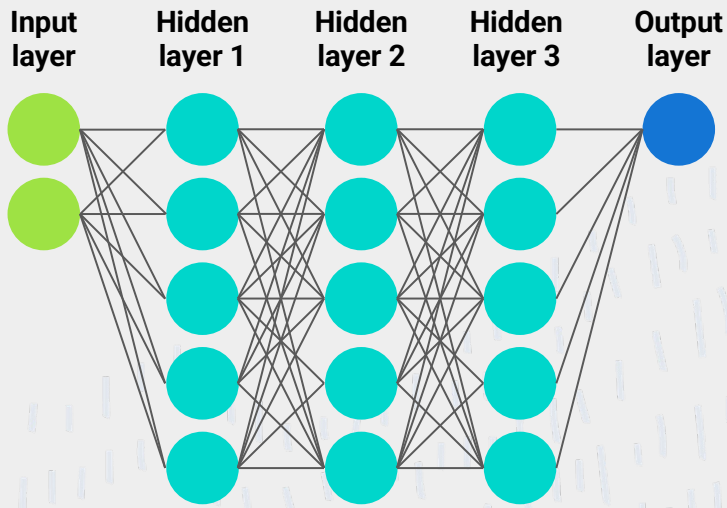
Basic Neural Network vs. Deep Learning Model

The outputs of one hidden layer become the inputs to additional hidden layers of neurons. This enables the next layer of neurons to evaluate higher-order interactions between weighted variables and to identify complex, nonlinear relationships.

Basic Neural Network

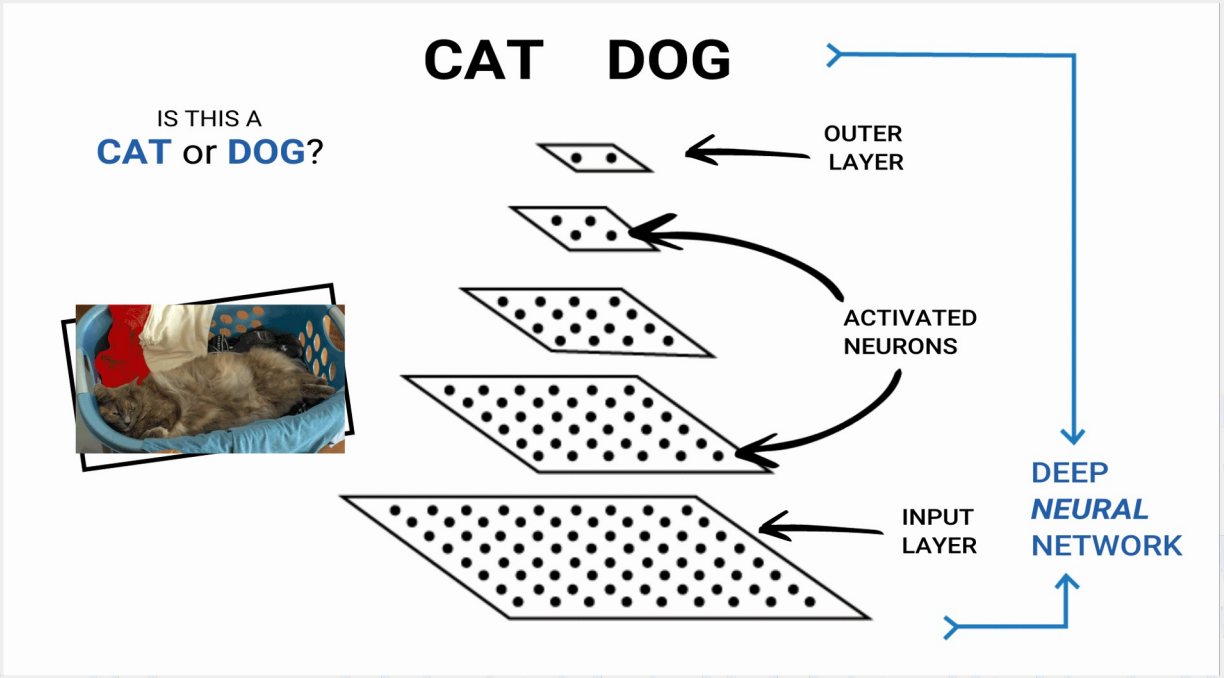


Deep Learning Model



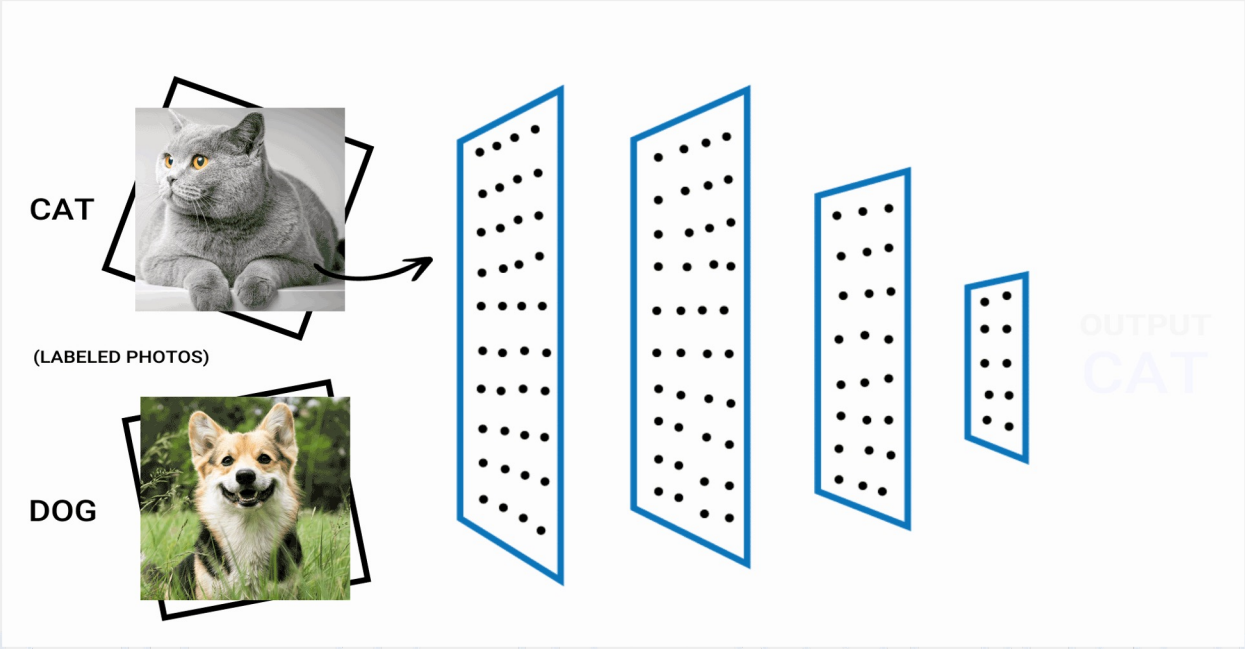
Neural Networks

Neural networks calculate the weights of various input data and pass them to the next layer of neurons. This process continues until the data reaches the output layer, which makes the final decision on the predicted category or numerical value of an instance.



Neural Networks

While definitions vary, we can consider neural networks with more than one hidden layer to be deep learning models. The decreasing cost and greater availability of computing power has increased our ability to create and use these models.



Transfer Learning

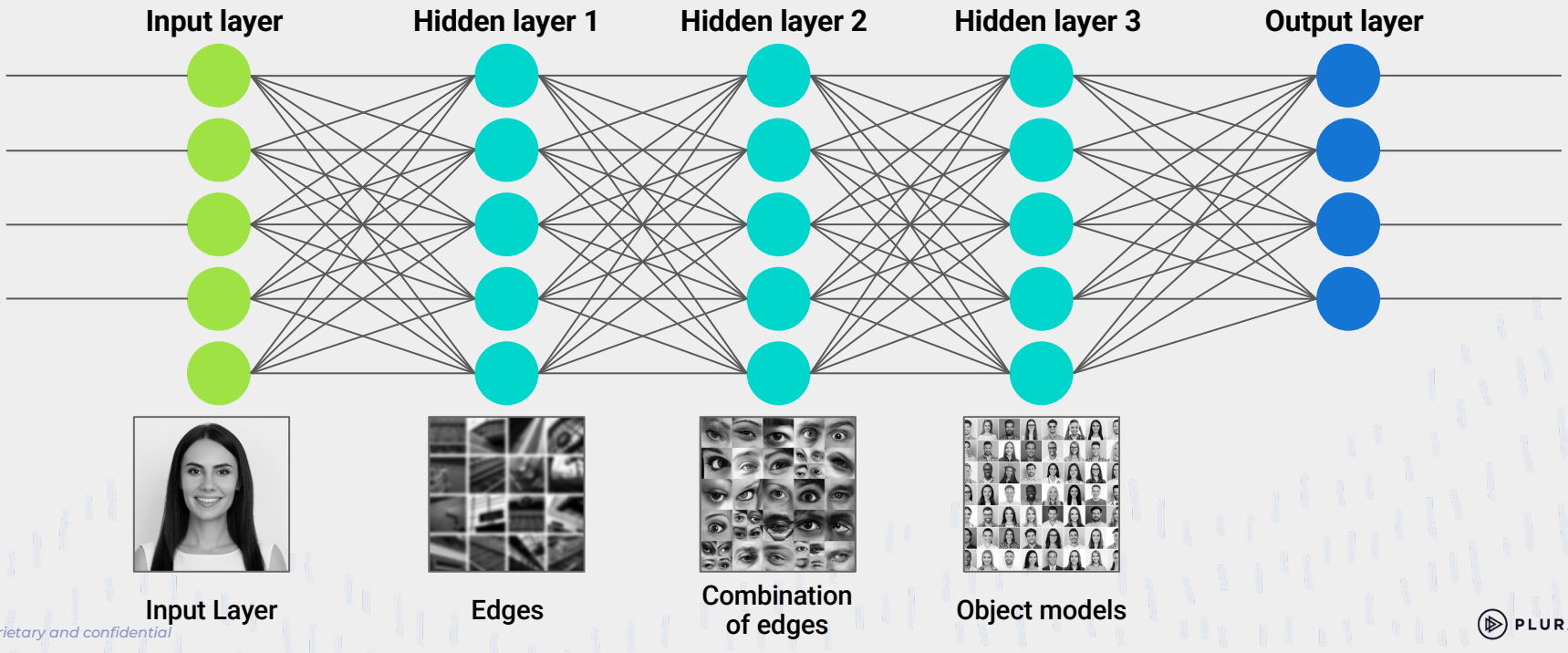
- **Definition**: The process of taking a model trained on one task and applying it to a different, but related, task. This is often done by fine-tuning a pre-trained model.
- **Application**: Widely used to apply large-scale models trained on general tasks to more specific tasks, such as using a model trained on general images to identify specific types of objects.

Exercise 1

Different DL Architectures

Deep Neural Network

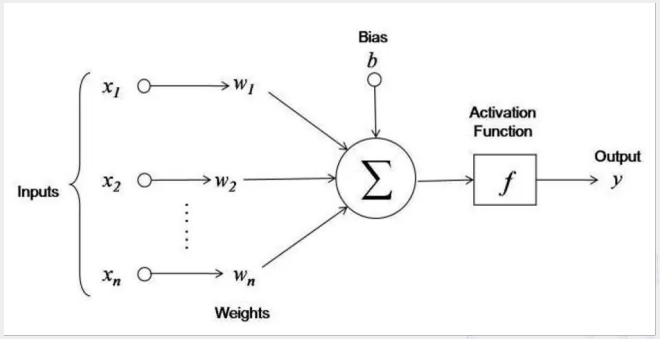
In image recognition, each layer can identify different image features in the process of defining or identifying the image.



Deep Neural Network

Neurons is where the information processing takes place

- Information transformed from one layer to the next
- Activation Function decides which neurons get activated
- There are **Weights** and **Biases** that get passed



Activation Function

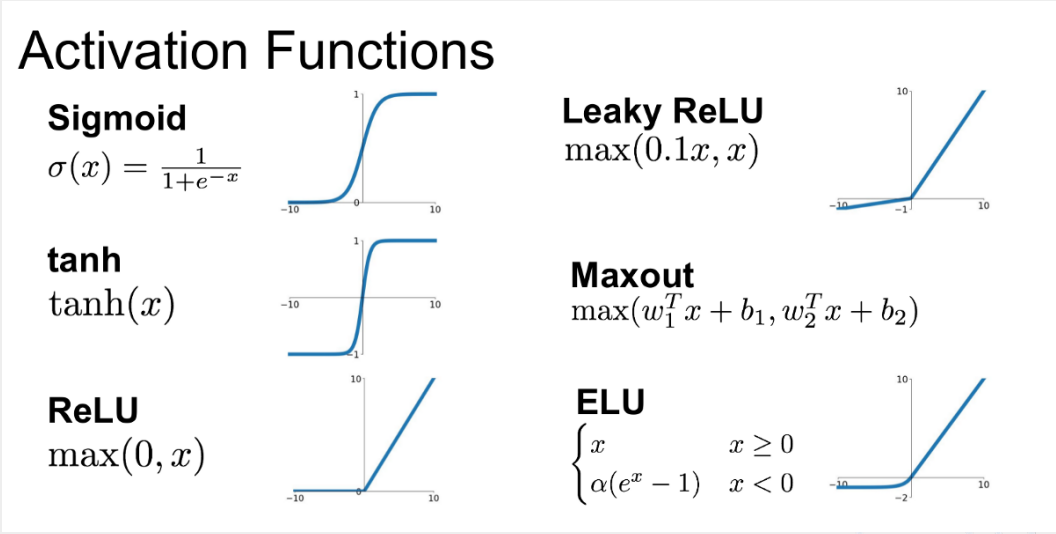
- Introduce **non-linearity** into the output of a neuron
- Without an activation function, the network would essentially behave like a linear regression model
- Activation functions are a critical component of neural networks that enable them **to learn and model complex relationships in the data**

Deep Neural Network

The **Rectified Linear Unit (ReLU)** is a non-linear activation function used in neural networks to introduce non-linearity into the output of a neuron. It is a piecewise linear function that returns the input directly if it is positive, and zero if it is negative.

f(x)=max(0,x)

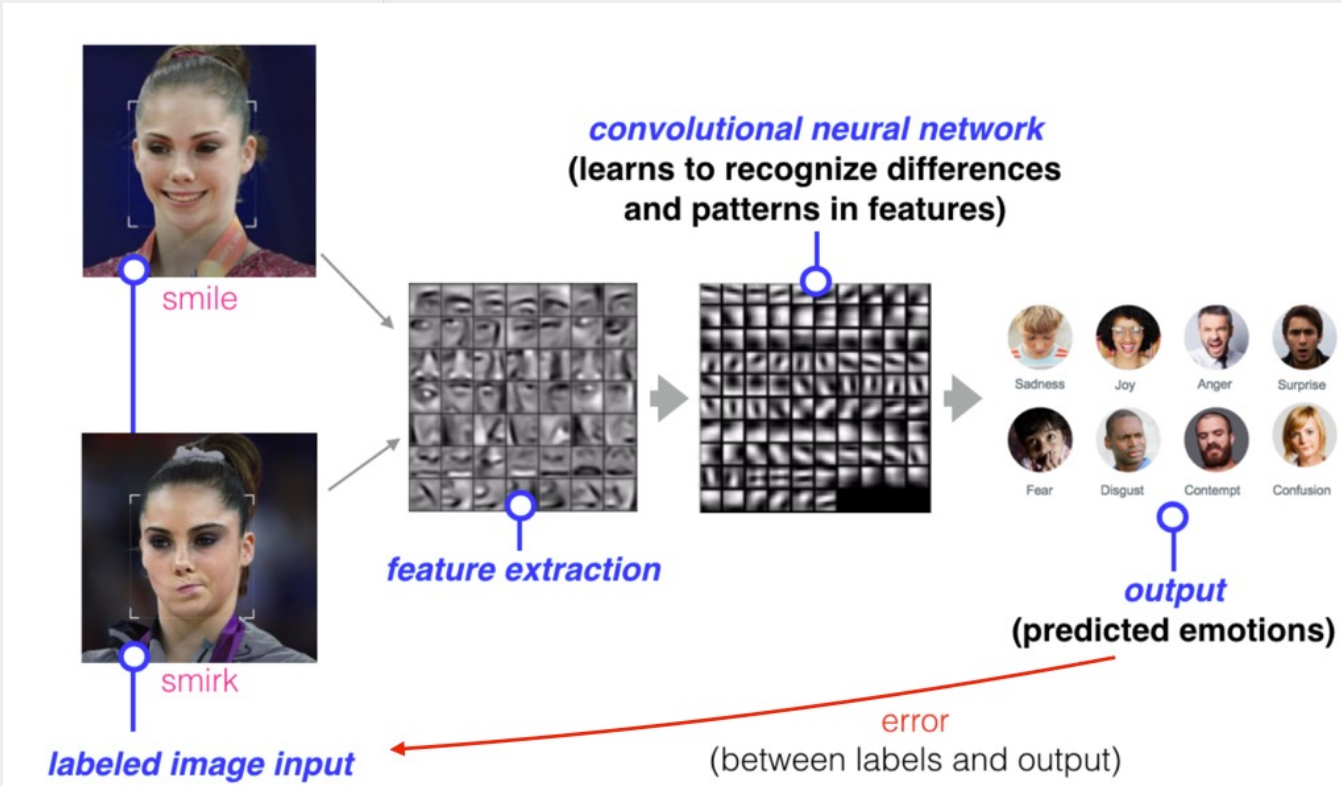
The **ReLU** function is simple and computationally efficient, making it a popular choice for deep learning tasks



Different DL Architectures

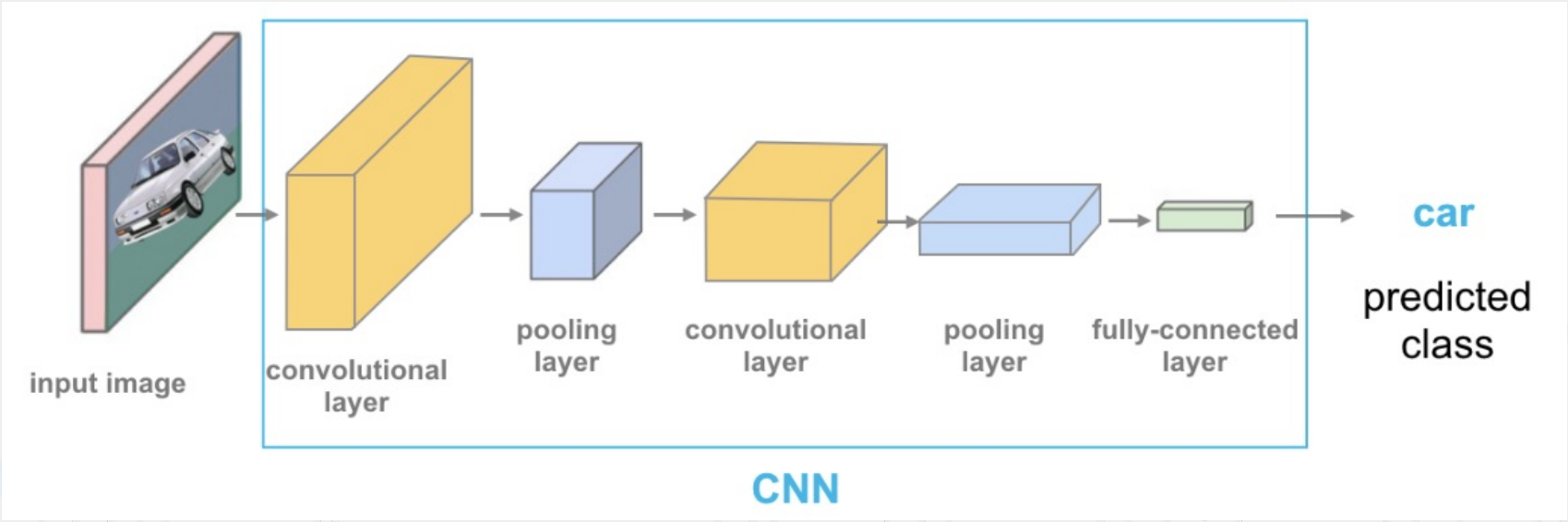
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Long Short-Term Memory (LSTM)
- Transformers
- Autoencoders

CNN Architecture

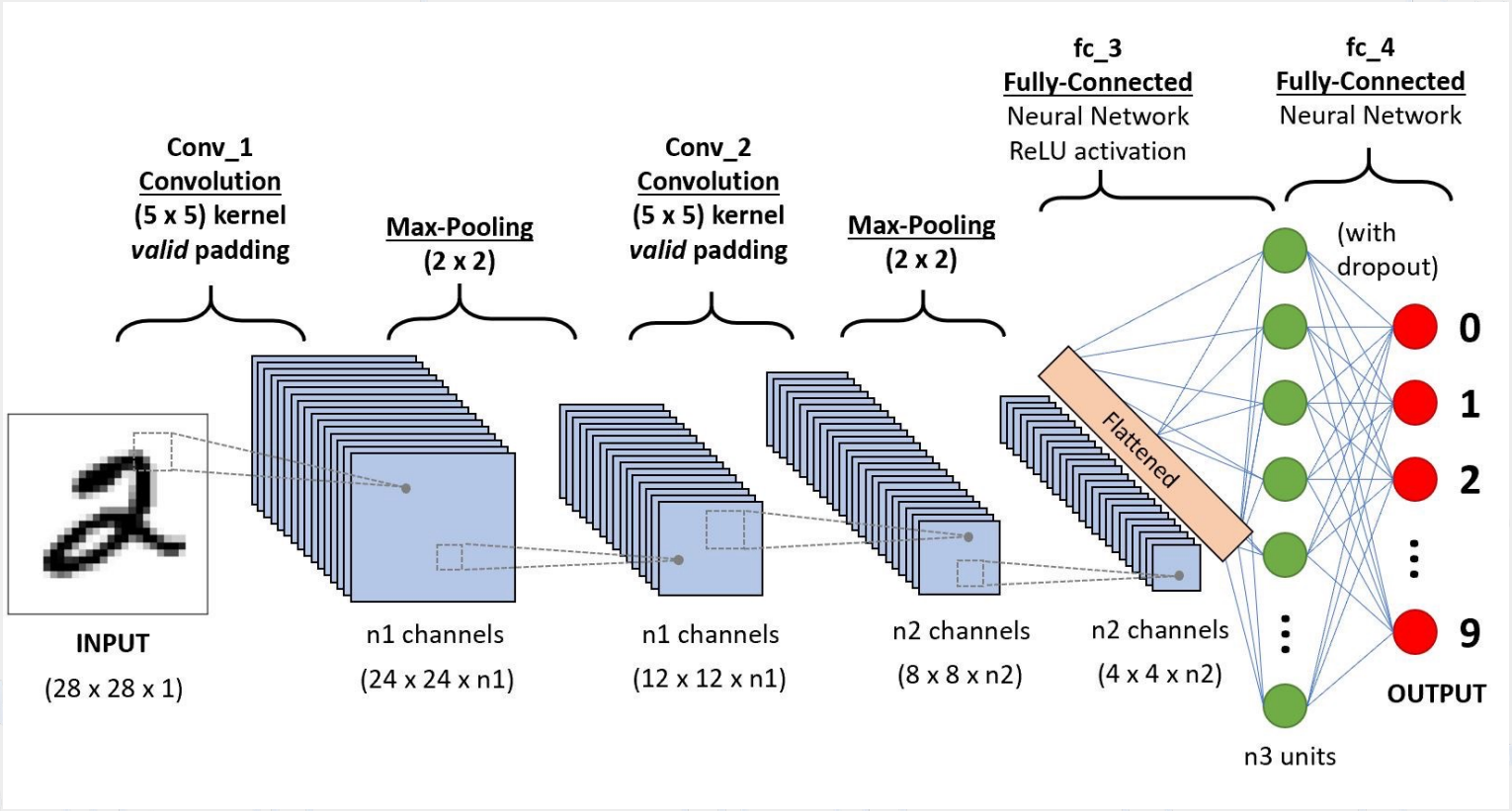


CNN Architecture

Every **CNN** is made up of multiple layers, the three main types of layers are **convolutional**, **pooling**, and **fully-connected.**



CNN Architecture



Convolution Process

A convolutional layer works by applying a filter to images. The filter is defined by a *kernel* that consists of a matrix of weight values.

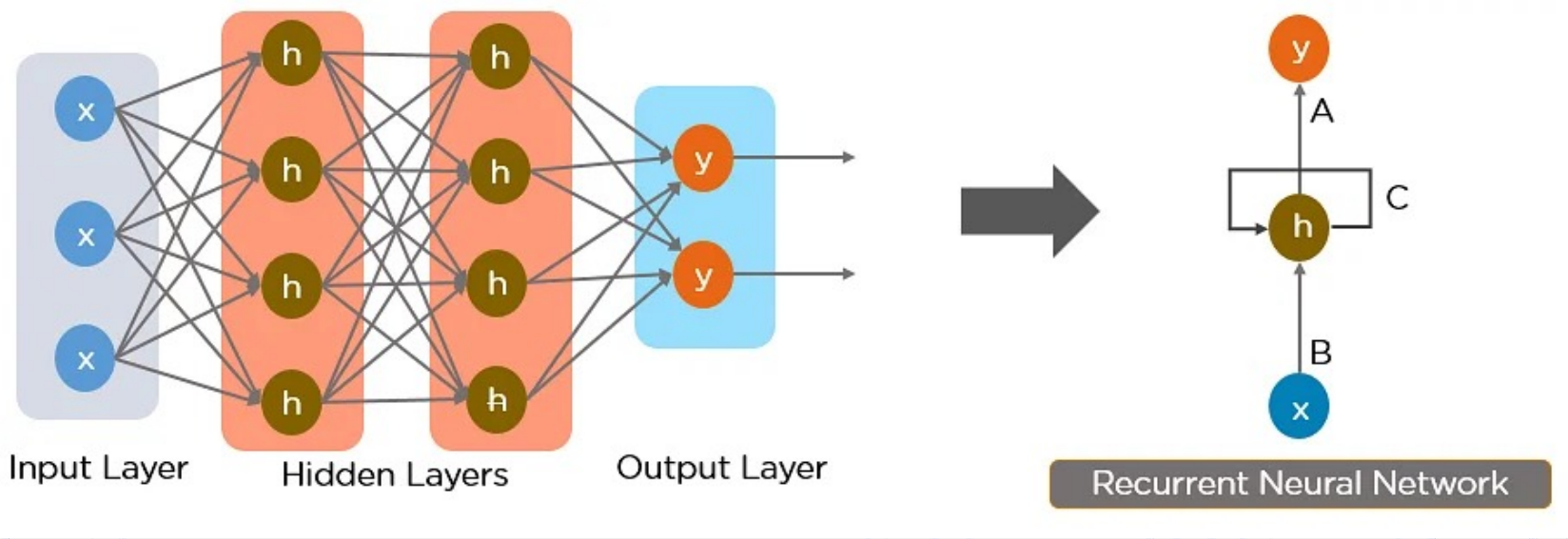
Typically, a convolutional layer applies multiple filter kernels. Each filter produces a different feature map, and all of the feature maps are passed onto the next layer of the network.

Pooling Process

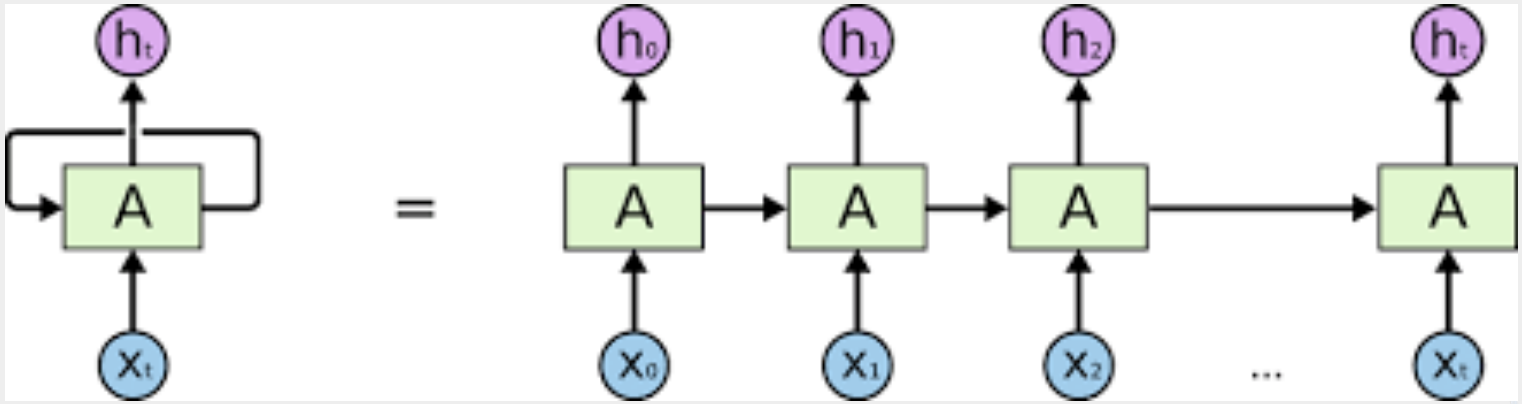
A convolutional layer works by applying a filter to images. The filter is defined by a *kernel* that consists of a matrix of weight values.

Typically, a convolutional layer applies multiple filter kernels. Each filter produces a different feature map, and all of the feature maps are passed onto the next layer of the network.

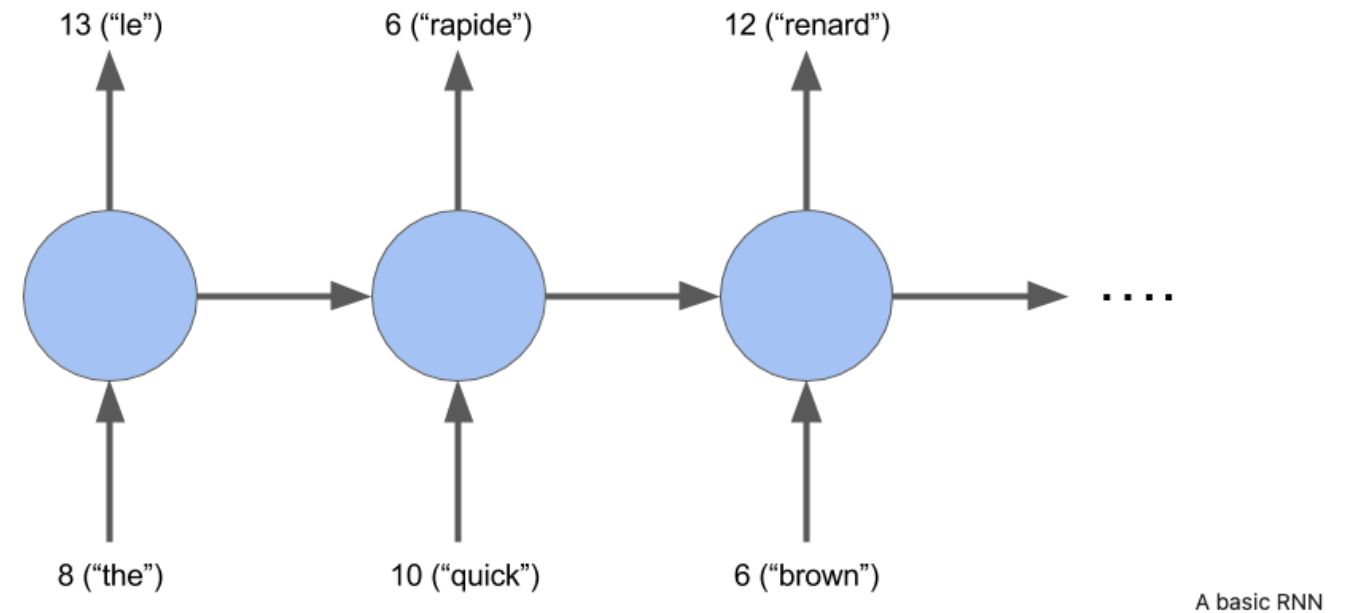
RNN Architecture



RNN Architecture



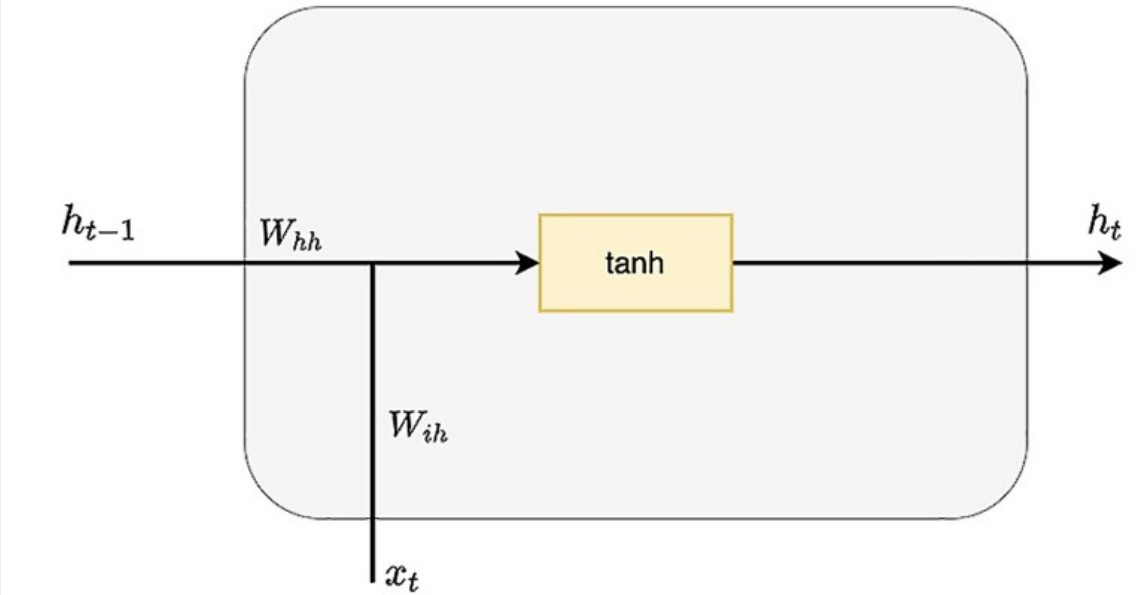
RNN Architecture



A basic RNN

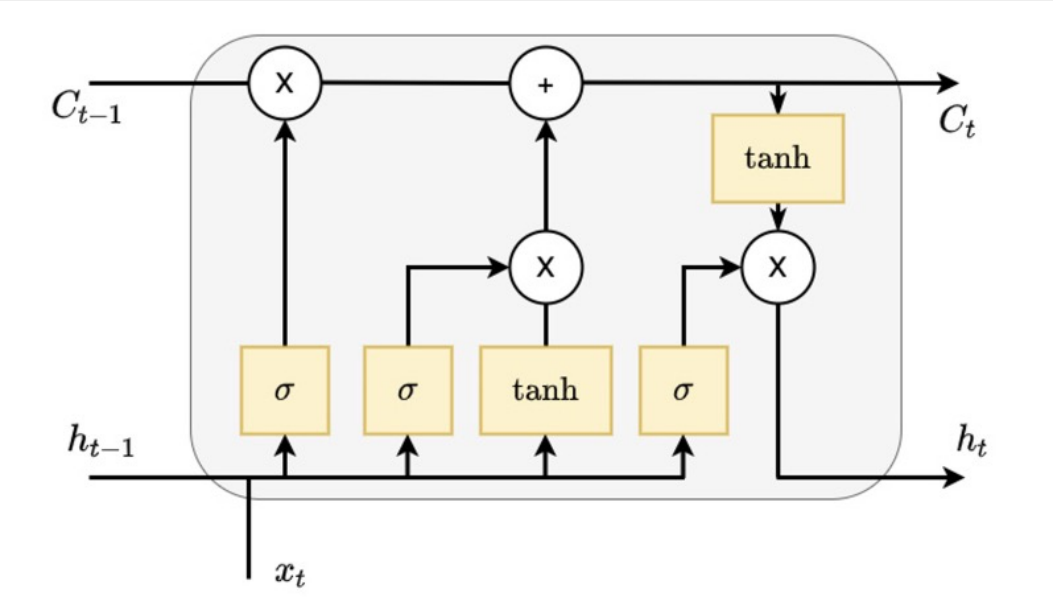
LSTM Architecture

RNN cell



LSTM Architecture

LSTM cell



Core Generative AI Models

Different DL Architectures

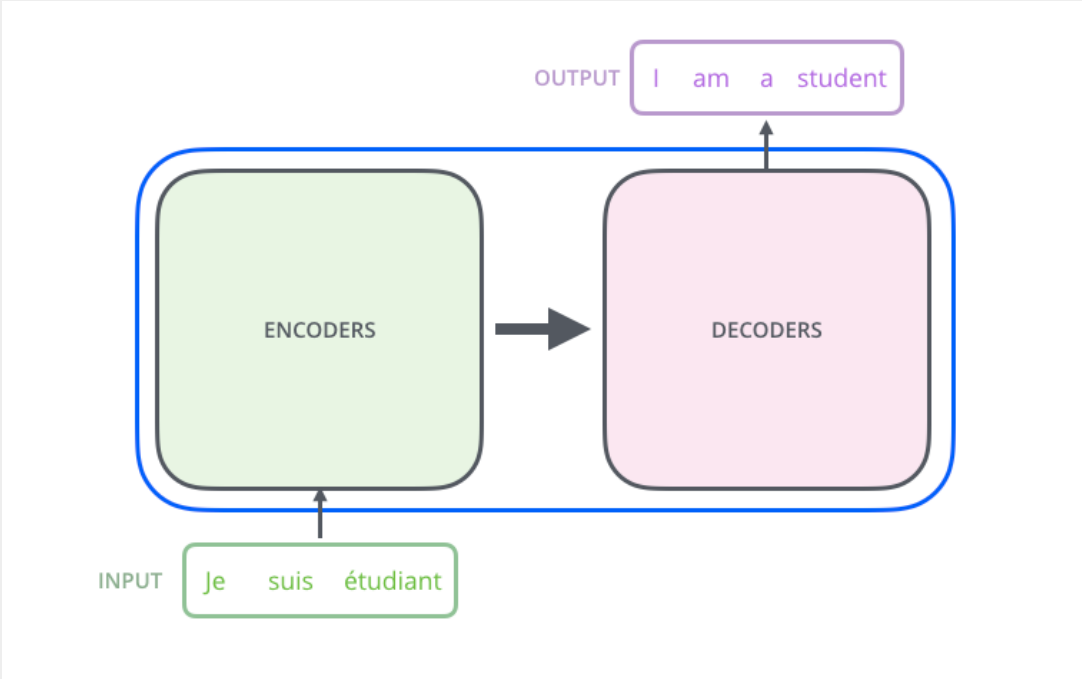
- Transformers
- Generative Adversarial Networks (GANs)
- Variational Autoencoders
- Diffusion Models

Transformer Architecture

An alternative to Recurrent Neural Networks (**RNNs**) to address the **vanishing gradients** and the **struggle processing long text sequences**

The Transformer Architecture is a 2-stack structure:

- **Encoder**
- **Decoder**



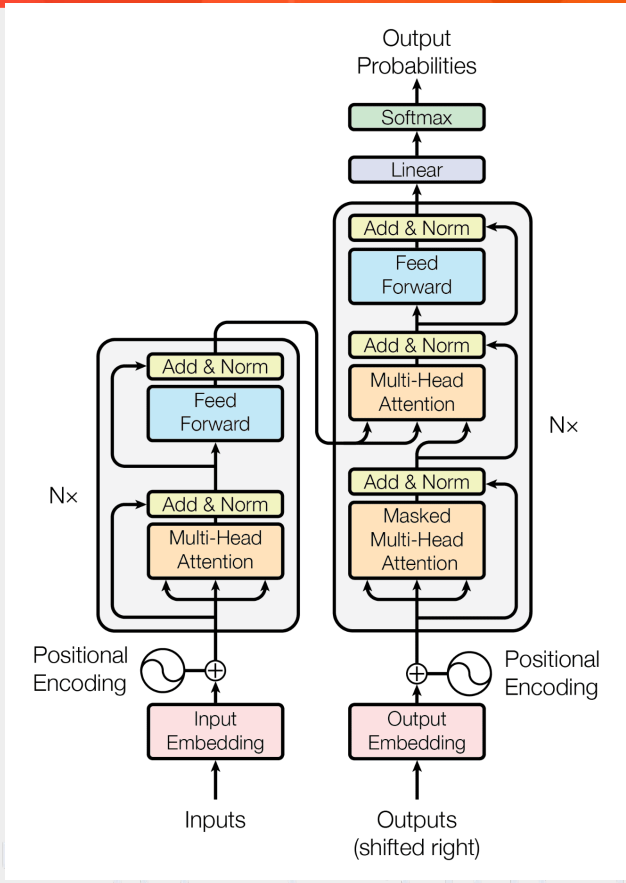
Transformer Architecture

Attention Is All You Need

https://arxiv.org/abs/1706.03762

Attention Mechanism:

- Focus on valuable text
- Filter unnecessary elements
- Model long-term text dependencies



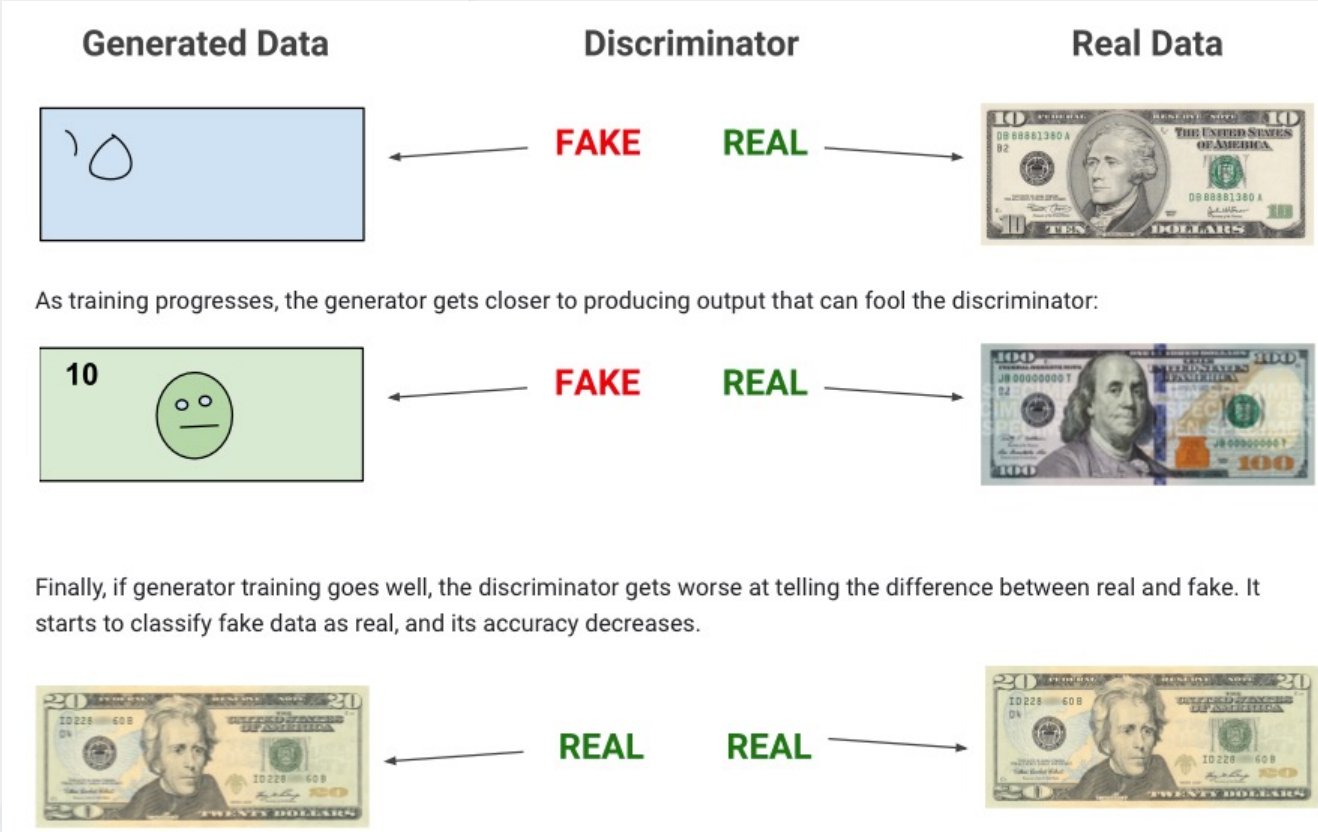
Transformer Architecture

- **How They Work**: Transformers use attention mechanisms to process input data. They excel at handling sequences of data, such as text or time-series information.
- **Key Characteristics**:
  - **Attention Mechanisms**: Allows the model to focus on different parts of the input sequence, making it effective for tasks like language translation, text summarization, etc.
  - **Parallel Processing**: Unlike recurrent models, Transformers process all elements of the sequence simultaneously, leading to faster training.
- **Applications**: Widely used in natural language processing (NLP) tasks, such as language modeling, translation, and text generation.

Transformer Architecture

- **RNNs** are sequential models that process data one element at a time, maintaining an internal hidden state that is updated at each step. They operate in a recurrent manner, where the output at each step depends on the previous hidden state and the current input.
- **Transformers** do not rely on recurrence but instead operate on self-attention. Self-attention allows the model to weigh the importance of different parts of the input sequence, enabling it to process the entire sequence in parallel
- **RNNs** are known to struggle with long-range dependencies, as the information from earlier time steps can get diluted or lost over time.
- **Transformers**, on the other hand, are better at handling long-range dependencies, as they can weigh the importance of different parts of the input sequence and process the entire sequence in parallel

GANs Architecture

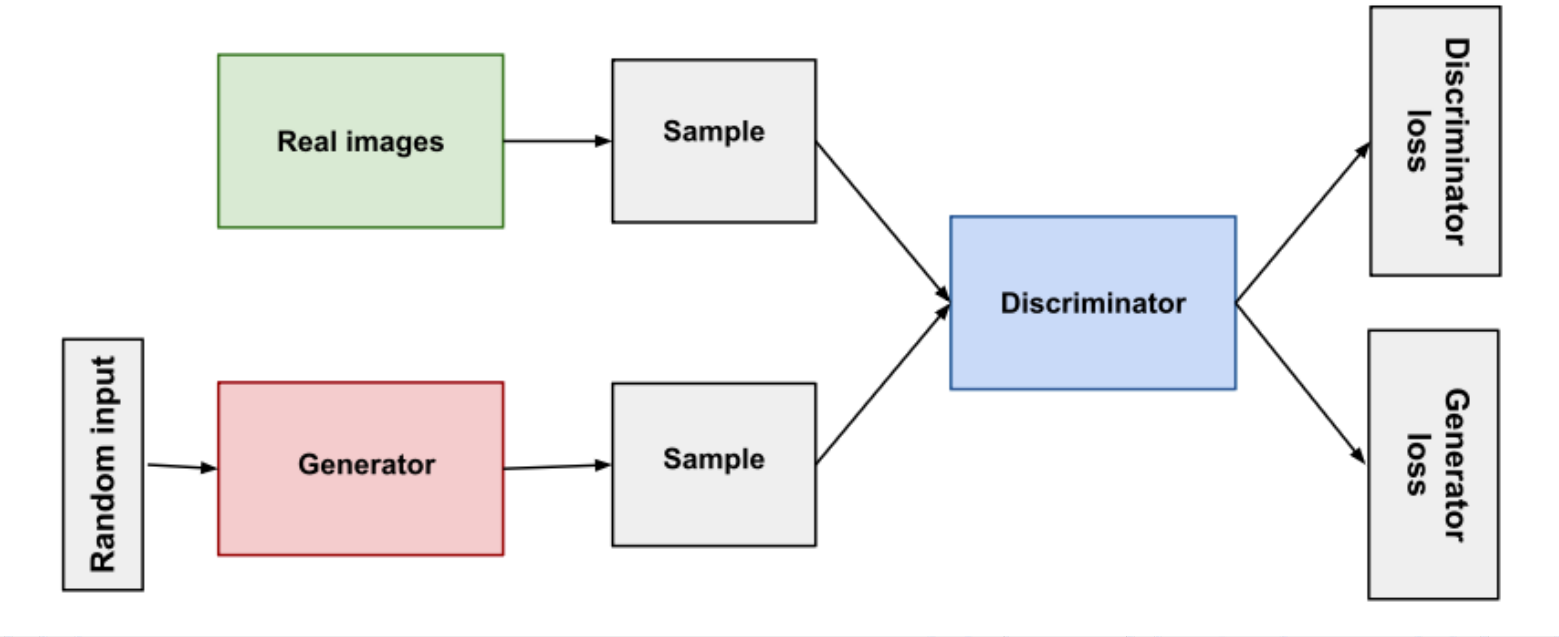


Source: Google Developer

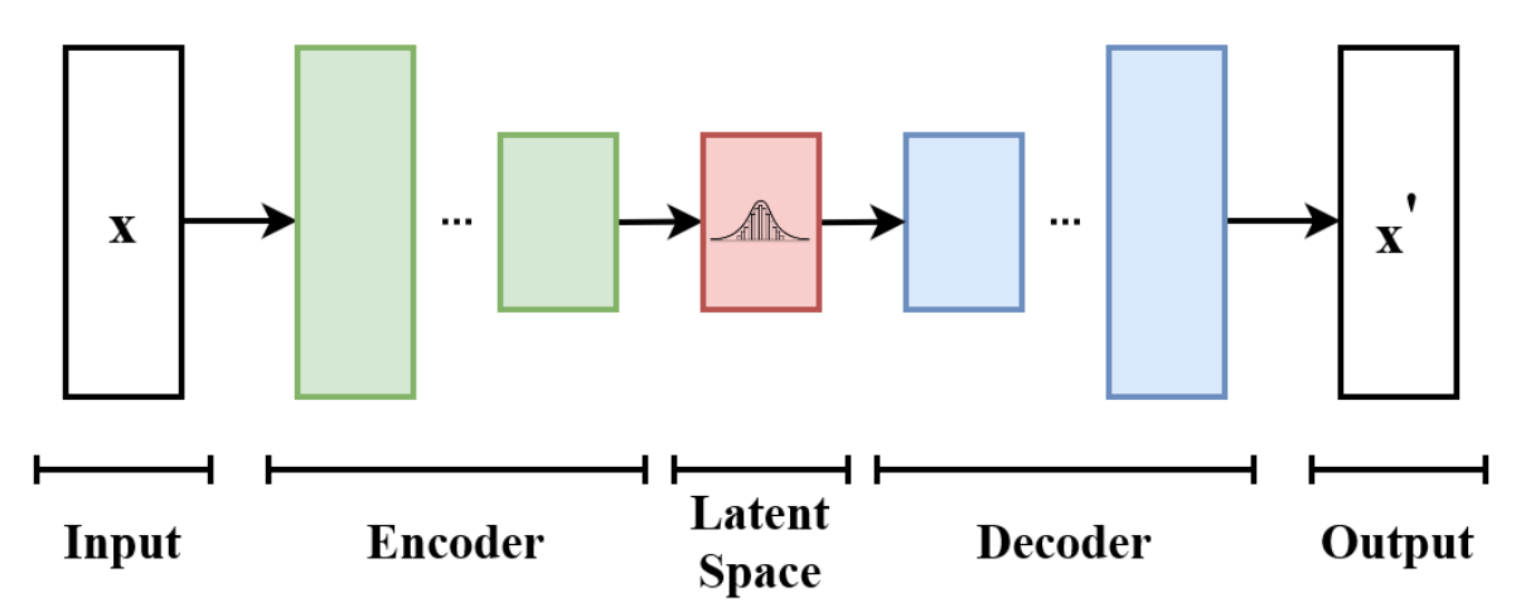
GANs Architecture

- **How They Work**: GANs consist of two neural networks, a generator and a discriminator, that are trained simultaneously. The generator creates fake data, while the discriminator learns to distinguish between real and generated data.
- **Key Characteristics**:
  - **Adversarial Training**: The generator and discriminator improve iteratively in a competitive manner.
  - **Implicit Density Modeling**: GANs learn to generate data without explicitly modeling the probability distribution.
- **Applications**: Commonly used for image generation, style transfer, and data augmentation.

GANs Architecture



VAEs Architecture



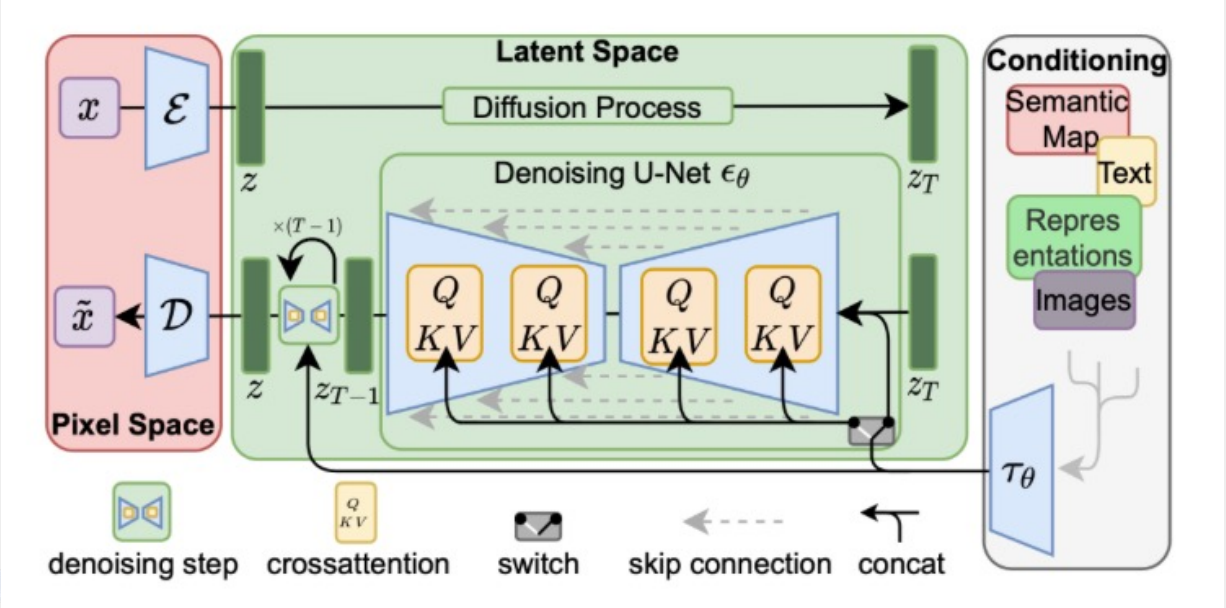
Latent Space: A mathematical space that stores large dimensional data in a compressed format

VAEs Architecture

- **How They Work**: VAEs are designed to learn a latent representation of input data. They consist of an encoder that maps input to a latent space and a decoder that reconstructs the input from the latent representation.
- **Key Characteristics**:
  - **Probabilistic Approach**: They model the distribution of input data and generate new data by sampling from this distribution.
  - **Reconstruction Loss**: Part of the training involves minimizing the difference between the original data and its reconstruction.
- **Applications**: Used for image generation, anomaly detection, and as a basis for more complex generative models.

Diffusion Models

Denoising Diffusion Probabilistic Models
https://arxiv.org/abs/2006.11239

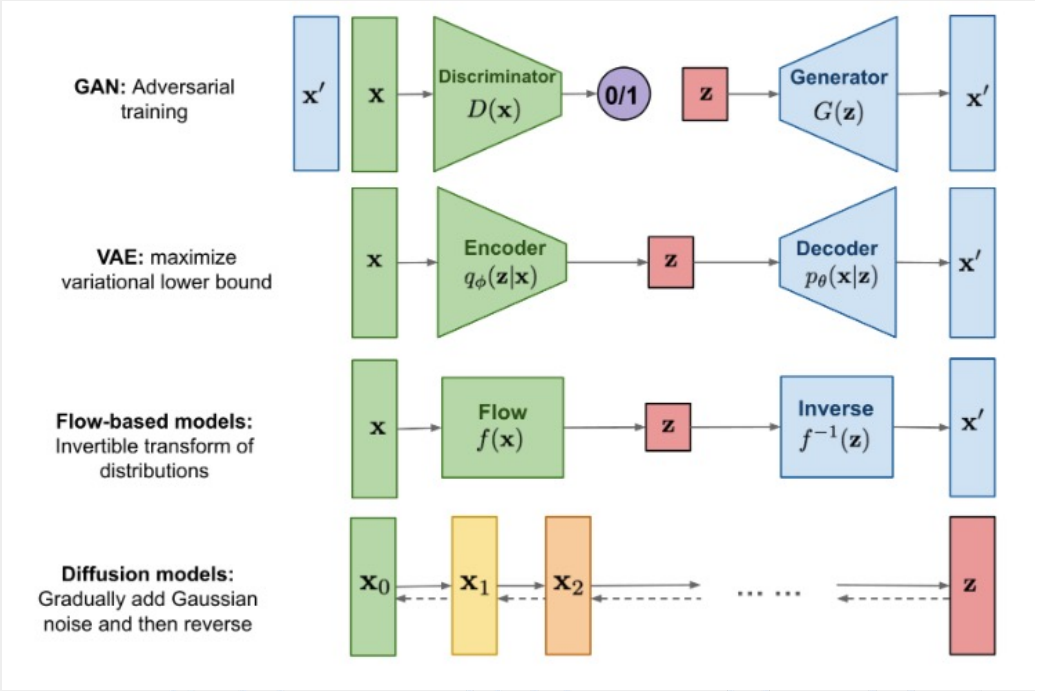


Diffusion Models

- **How They Work**: Diffusion models generate data by gradually transforming a sample of random noise into a structured output (like an image or audio). This process involves a series of steps that progressively denoise the data, guided by a learned model.
- **Key Characteristics**:
  - **Gradual Denoising**: The transformation from noise to structured data happens over many steps, unlike other models that often generate output in one shot.
  - **Stochastic Process**: These models involve a random process, introducing variability in the generation process.
- **Applications**: Primarily used in image and audio generation, where they excel at producing high-quality, high-resolution outputs.

Overview of Different Generative Models

- **Model Structure**: Diffusion models use a process of adding and removing noise, Transformers use attention mechanisms, GANs involve adversarial networks, and VAEs are based on autoencoding with a probabilistic twist.
- **Data Generation**: Diffusion models generate data through a denoising process, while GANs and VAEs generate data from a learned latent space, and Transformers generate sequential data.
- **Applications**: While there's some overlap, each model has areas where it excels, like diffusion models for high-quality image generation, Transformers for NLP, GANs for photorealistic image creation, and VAEs for tasks requiring a latent space representation.



Core Generative AI
Other Key Concepts

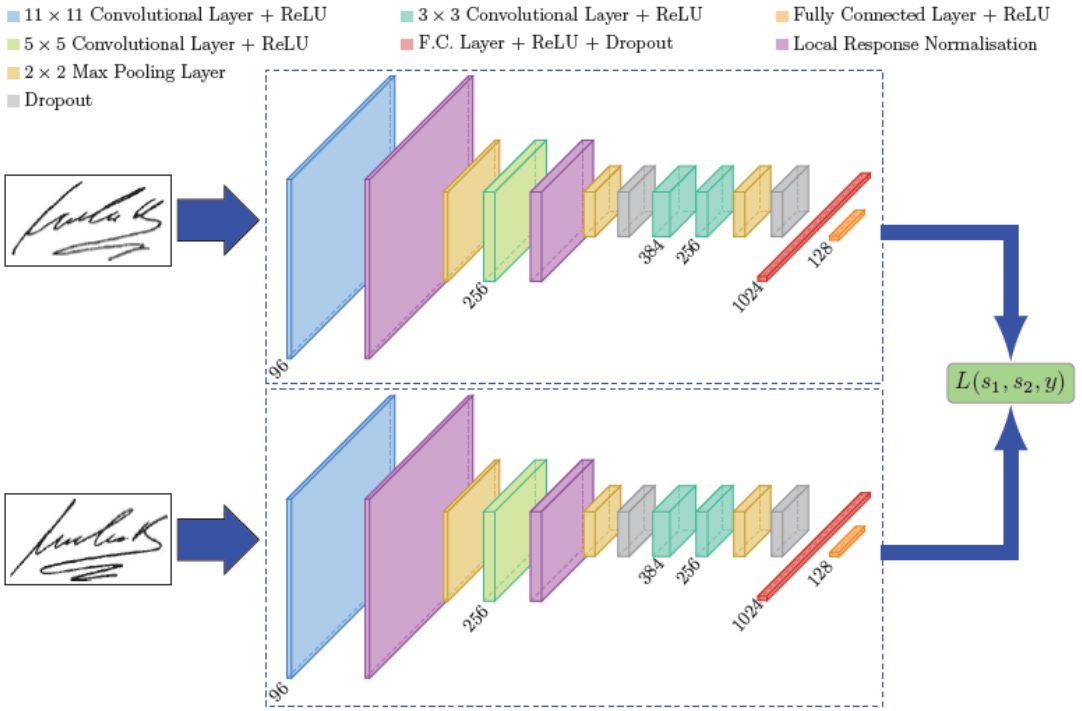
One-Shot Learning

- **Definition**: The ability of a model to learn from a single example or a few examples. In generative AI, this means being able to generate new data that is similar to a given example with minimal training data.
- **Application**: Useful in situations where large datasets are not available, such as rare disease diagnosis in healthcare.

One-Shot Learning

Siamese neural network

One Shot Image Recognition



Zero-Shot Learning

- **Definition**: The ability of a model to understand and perform tasks it has not explicitly been trained on. In generative AI, this might involve generating content or solving problems in domains not covered in the training data.
- **Application**: Enables more flexible and versatile AI systems, like a language model generating text in a genre it was not specifically trained on.

Zero-Shot Learning

Contrastive Language-Image Pretraining (CLIP)

- Developed by OpenAI
- Primarily a **transformer-based** model that has been widely used for zero-shot learning.
- CLIP consists of **two models**: a **text transformer** for encoding text embeddings and a **vision transformer (ViT)** for encoding image embeddings

Few-Shot Learning

- **Definition**: Similar to one-shot learning, but the model learns from a small number of examples rather than just one.
- **Application**: Useful in personalized AI applications, where the model adapts to individual preferences or needs with limited data.

Fine-Tuning

- **Definition**: Adjusting a pre-trained model on a new, typically smaller, dataset. This allows the model to specialize in a specific task or domain while leveraging the knowledge it gained during its initial training.
- **Application**: Common in adapting large language models to specific industries or niches, like legal or medical language.

Latent Space

- **Definition**: Latent space is a compressed representation of data that captures its essential features. It is a key concept in generative modeling, allowing models to understand the underlying patterns in the data
- **Application**: In image generation, traversing the latent space can smoothly transition between different types of images.

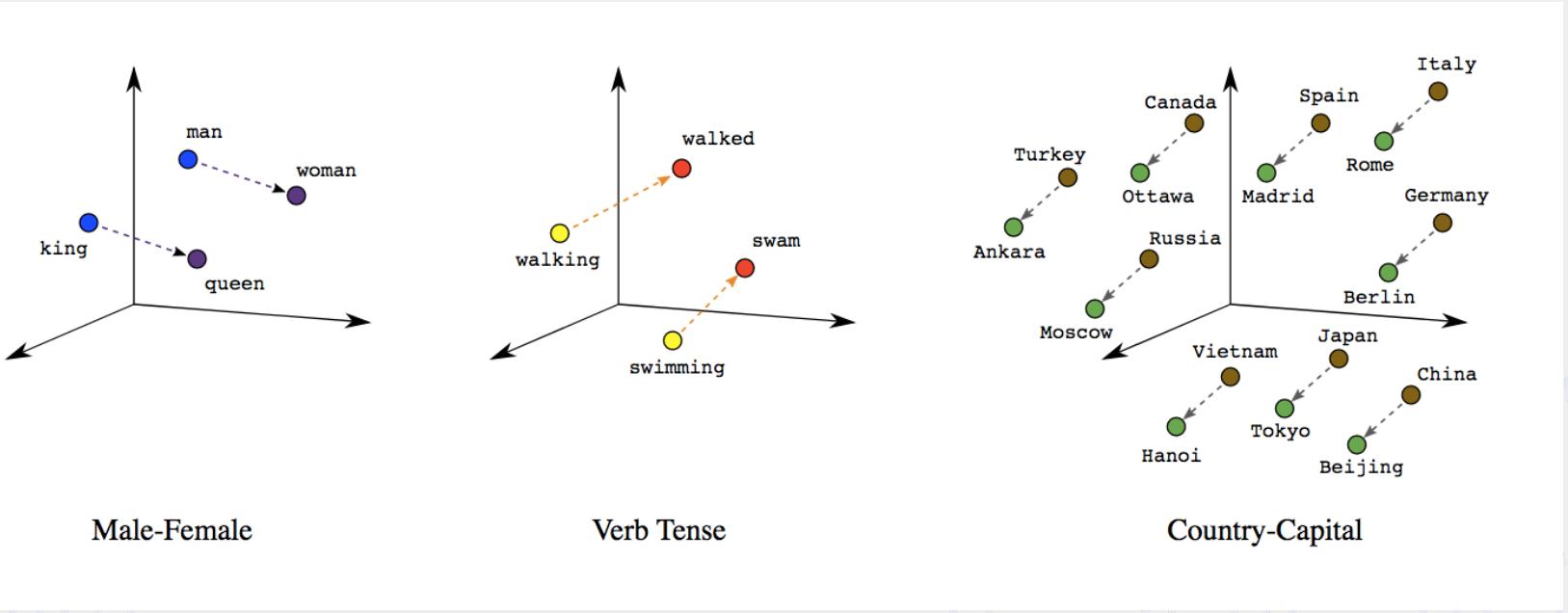
Transfer Learning

- **Definition**: The process of taking a model trained on one task and applying it to a different, but related, task. This is often done by fine-tuning a pre-trained model.
- **Application**: Widely used to apply large-scale models trained on general tasks to more specific tasks, such as using a model trained on general images to identify specific types of objects.

Foundation Models

- **Definition**: Foundation models are a class of large-scale models that are pre-trained on extensive and diverse datasets, often unsupervised or self-supervised. They have a wide range of capabilities and can be adapted to various tasks.
- **Characteristics**:
  - **Scale**: They are typically very large, both in terms of the size of the model (number of parameters) and the dataset used for training.
  - **Generalizability**: These models are designed to be general-purpose, meaning they can be fine-tuned or adapted to perform a wide variety of tasks, often with state-of-the-art performance.
  - **Examples**: Models like GPT-3, BERT, and other large language models are considered foundation models.
- **Usage**: Foundation models serve as a starting point for further task-specific training (fine-tuning) or for developing new models and applications.

Embedding



Large Language Models (LLMs)

Large Language Models (LLMs) are a type of artificial intelligence model specifically designed to understand, generate, and interact with human language at a large scale. They have become a significant focus in the field of Natural Language Processing (NLP) due to their remarkable ability to handle a wide range of language-related tasks.

Retrieval Augmented Generation (RAG)

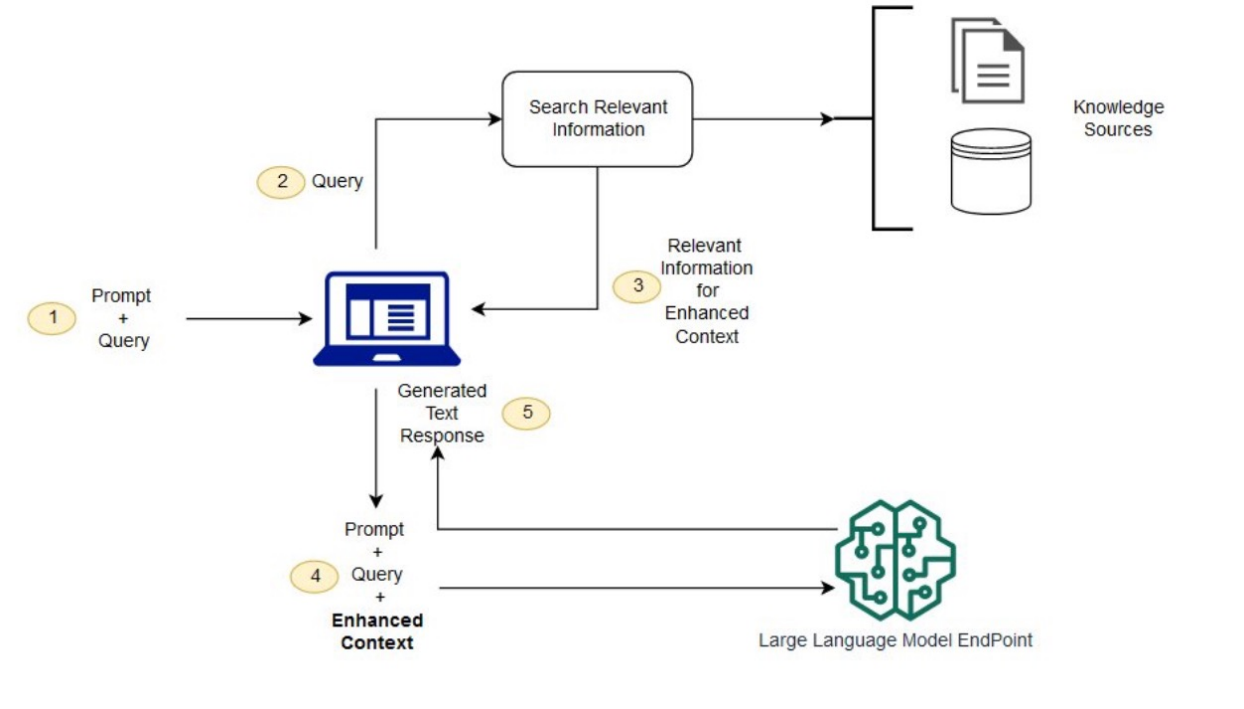
RAG: Gives LLM access to external data sources

Retrieval-augmented generation (**RAG**) is an advanced artificial intelligence technique that combines **information retrieval** with **text generation**. It enhances the accuracy and reliability of generative AI models by allowing them to retrieve relevant information from external sources and incorporate it into generated text.

The **RAG** architecture integrates a **neural retriever** and a **neural generator**. The retriever is used to fetch relevant context or information from a large corpus of data (like a database or a collection of documents), and the generator then uses this retrieved information to construct a response or output.

Retrieval Augmented Generation (RAG)

RAG: Gives LLM access to external data sources



LangChain

LangChain is a framework designed to simplify the creation of applications using large language models (LLMs).

- **Integration with LLMs**: LangChain allows developers to connect LLMs, such as OpenAI's GPT-3.5 and GPT-4, to external data sources to create and reap the benefits of natural language processing (NLP) applications

- **Modular Components**: LangChain provides modular and easy-to-use components, such as interfaces and integrations for working with language models, retrieval interfaces with application-specific data, chains for constructing sequences of calls, and agents for interacting with APIs

LangChain

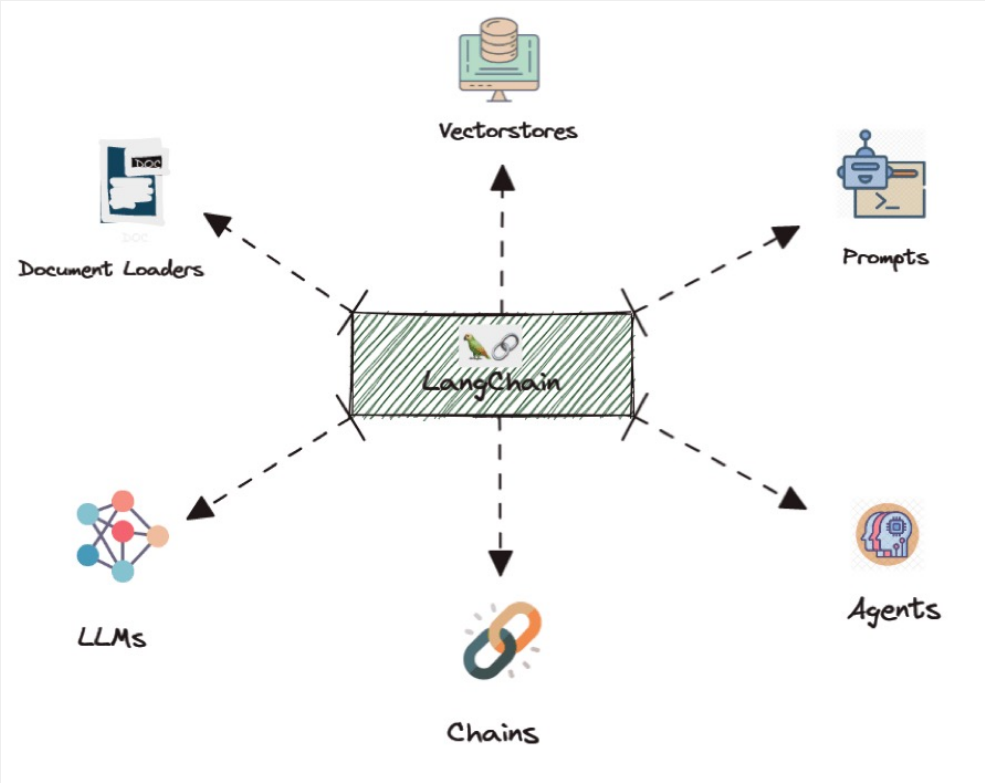
- **Off-the-shelf Chains**: LangChain offers off-the-shelf chains, which are built-in assemblages of components for accomplishing higher-level tasks. These chains make it easy to get started and customize existing chains to build new ones

- **Templates**: LangChain provides LangChain Templates, a collection of easily deployable reference architectures for a wide variety of tasks

- **LangServe**: LangChain introduced LangServe, a deployment tool designed to facilitate the transition from LCEL (LangChain Expression Language) prototypes to production-ready applications

Some common use cases for **LangChain** include **Q&A over documents**, analyzing structured data, interacting with APIs, code understanding, agent simulations, chatbots, code writing, extraction, analyzing graph data, multi-modal outputs, self-checking, summarization, and tagging

LangChain



Vector Database

A vector database is a type of database that stores data as high-dimensional vectors, which are mathematical representations of features or attributes.

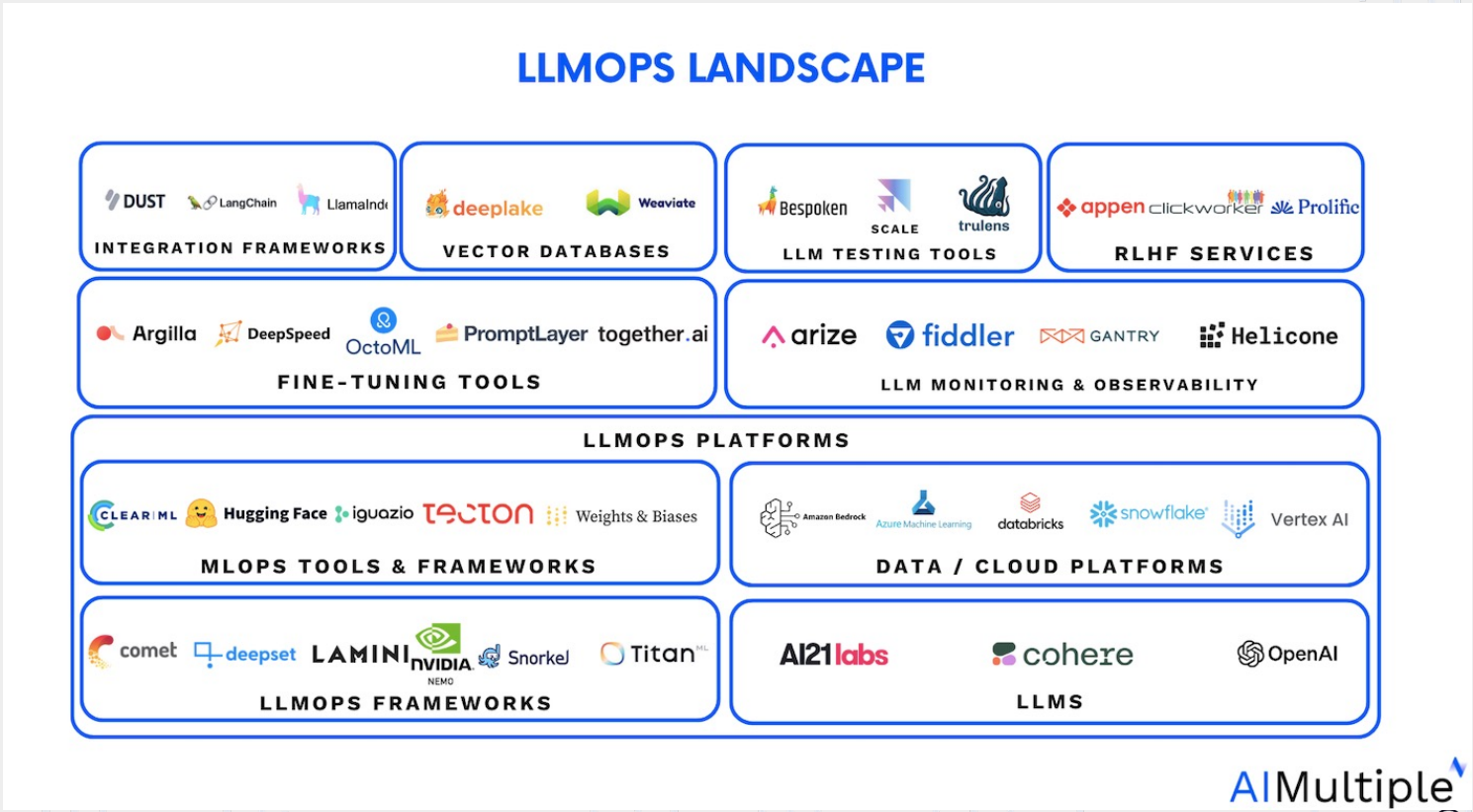
Vector databases have many use cases across different domains and applications that involve natural **language processing (NLP), computer vision (CV)**, **recommendation systems (RS)**, and other areas that require semantic understanding and matching of data.

LLMOps

LLMOps, or Large Language Model Operations, is a set of tools and best practices used to manage the lifecycle of large language models (LLMs) and LLM-powered applications, including development, deployment, and maintenance.

It focuses on the **operational** capabilities and **infrastructure** required to **fine-tune** existing **foundational models** and deploy these refined models as part of a product. LLMOps incorporates prompt management, LLM chaining, monitoring, and observability techniques not typically found in conventional MLOps. It is similar to MLOps but is specifically tailored to the unique requirements of LLMs and LLM-powered applications

LLMOps Landscape

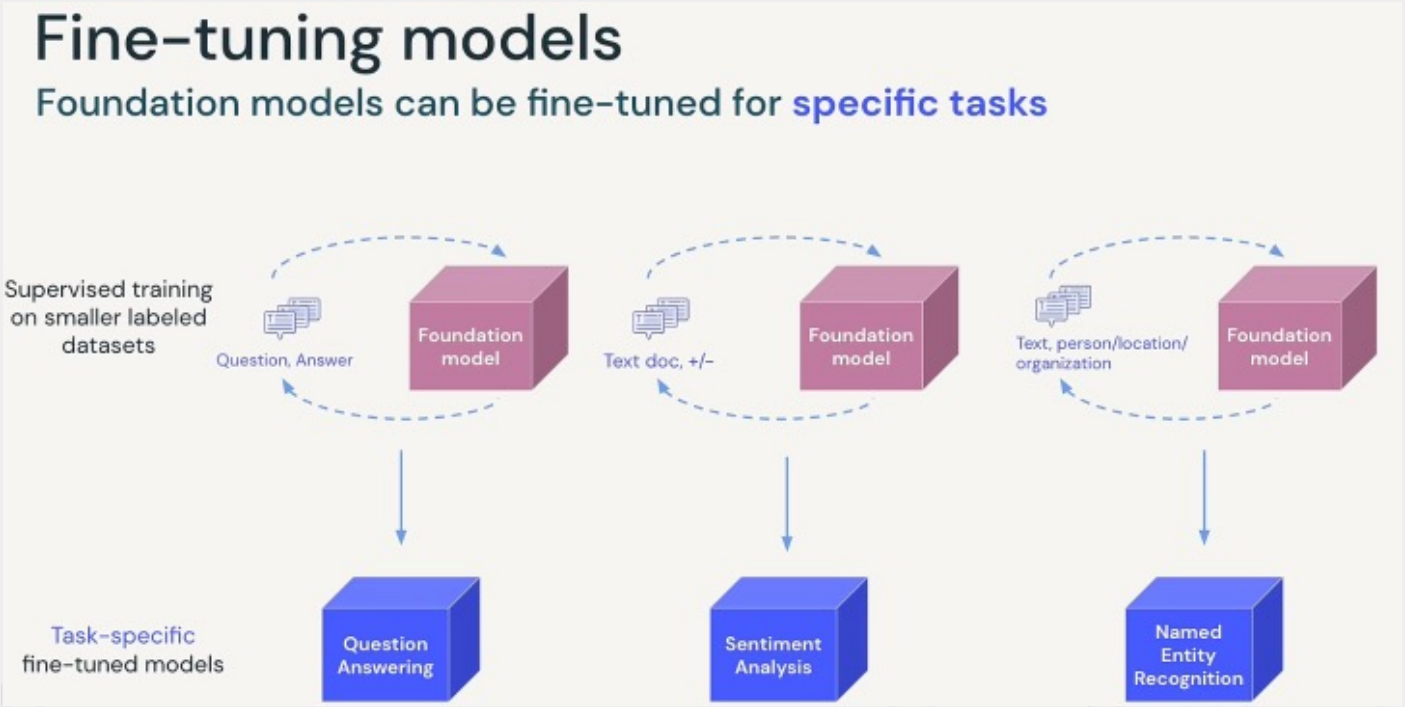


LLM Fine Tuning

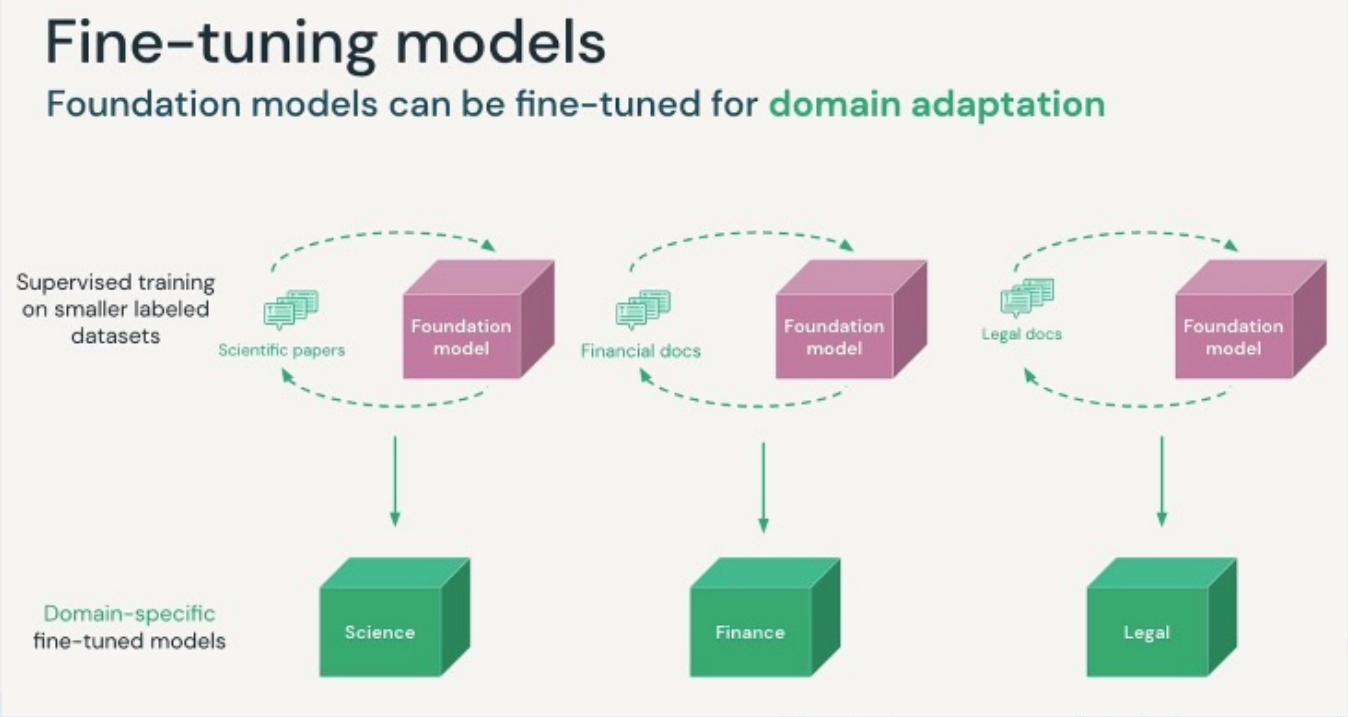
LLM fine-tuning is the process of adjusting the parameters of a **pre-trained** large language model (LLM) to a specific task or domain. This is done by training the model on a dataset of data relevant to the task, with the goal of improving its performance and making it more suitable for the specific application at hand**.**

Fine-tuning can be necessary when a pre-trained LLM needs to be adapted to the unique requirements of a specific application or domain

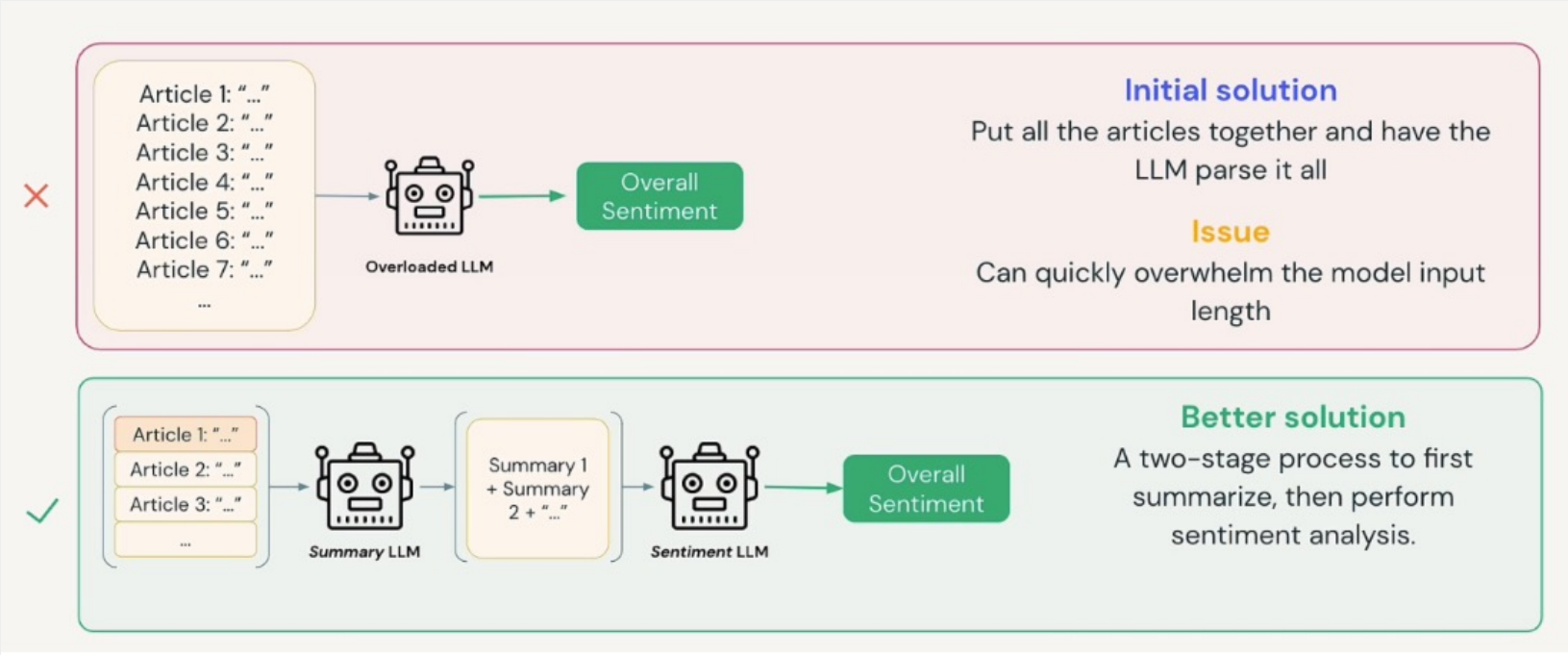
LLM Fune Tuning



LLM Fune Tuning



Mixing LLM Flavors in a Workflow



LLMOps

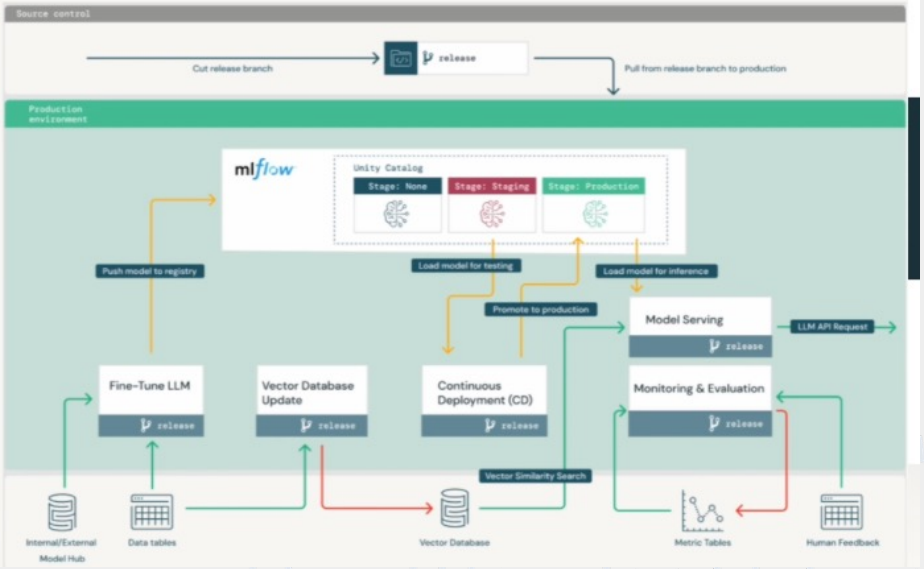
LLMOps, unified with DataOps + MLOps

LLM Operations for end-to-end production

- Databricks unifies LLMOps with traditional MLOps & DevOps
- Teams need to learn mental model of how LLMs coexist with traditional ML in operations

Differences to MLOps

- Internal/External Model Hub
- Fine-Tuned LLM
- Vector Database
- Model Serving
- Human Feedback in Monitoring & Evaluation



Prompt Injection

Prompt Injection

- **Definition:** Inserting a specific instruction or prompt within the input text to **manipulate the normal behavior** of LLMs.
- Other prompt injection cases:
  - Generating malicious code
  - Instructing agent to give wrong information
  - Revealing confidential information

Thank you!

If you have any additional questions, please ask!