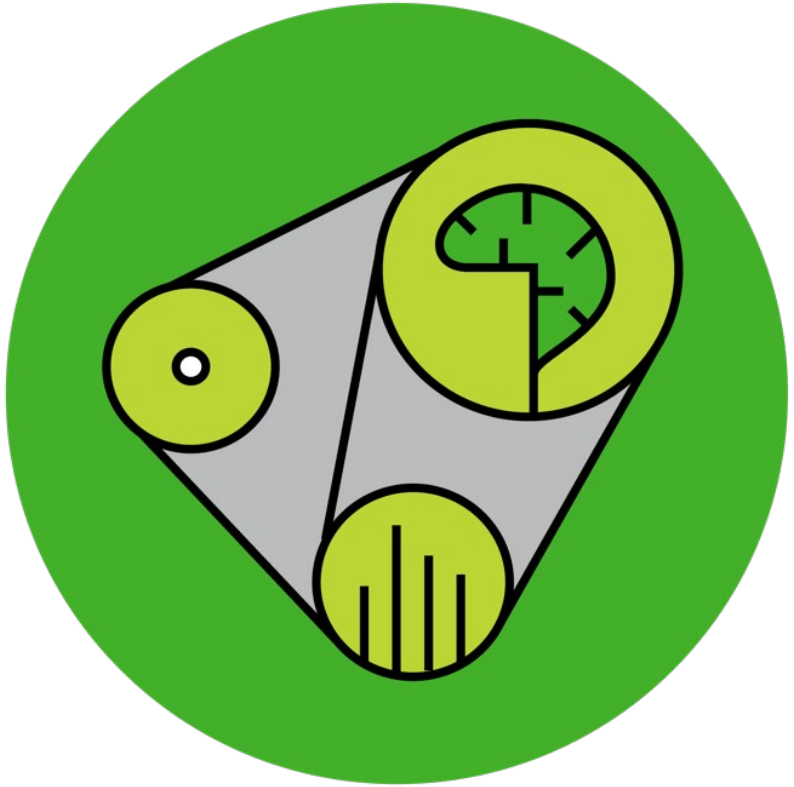
Deloitte.



AI Guild | GenAI Practicum

Introduction to NLP and Generative AI

March 2025

Welcome to the AI Guild GenAI Practicum

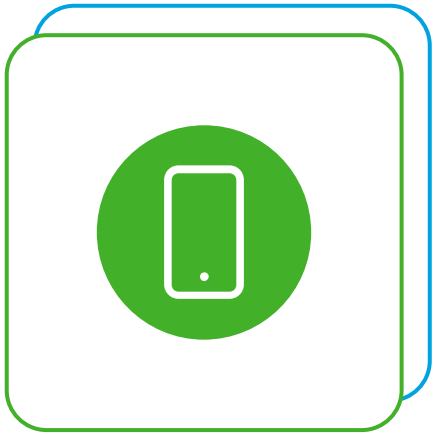
Introduction

The Instructor

Sourav Chakravarti

USI

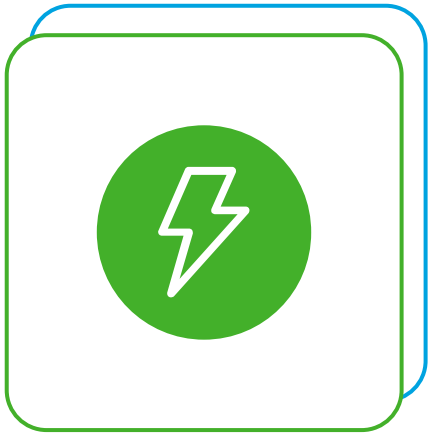
Generative AI may be the single most important technology innovation since the invention of...



…The iPhone?

…The computer?

…The automobile?



…Electricity

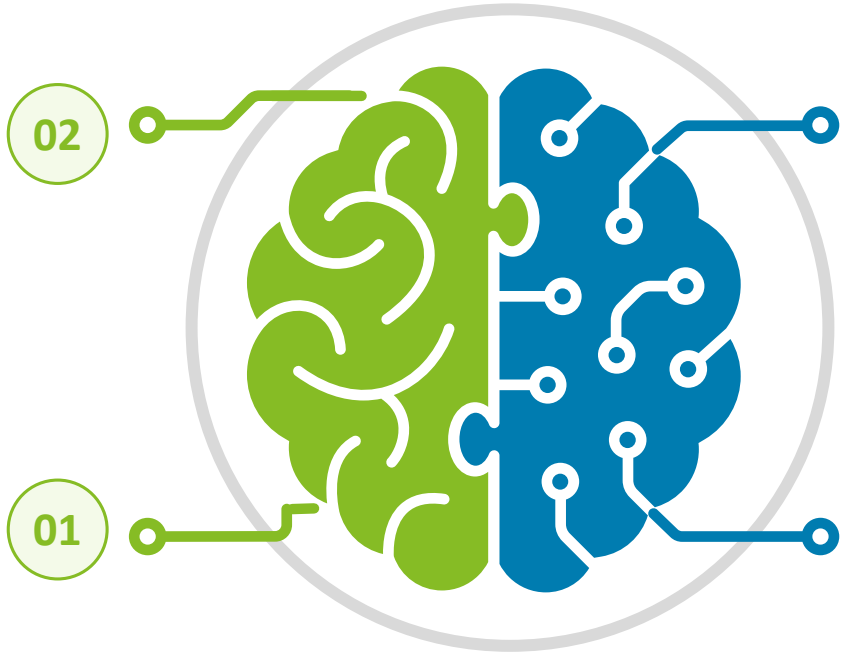
Or is it just hype? Come along with us on a journey of discovery and see what the big deal is all about!

Week Agenda

Topic	Content
01 Natural Language Processing	• Sidekick • The GenAI Center
02 Neural Networks	• Neural Networks, Deep Learning • Transformer Models
03 Introduction to Generative AI	• What is Gen AI? • Why is it Important?
04 Large Language Models	• LLMs, Foundation Models • Embedding Models
05 Cloud Offerings	• Open AI, Azure, Amazon, Google • Open Source, Alternatives

Learning objectives

By the end of session, you should be able to



01 Use Advanced NLP techniques to solve AI problems

02 Be familiar with Neural Networks

03 Explain what is Generative AI and why it is important

04 Identify Cloud, Open Source, and Alternative LLMs

Prerequisites

- Completion of AI Academy individual tracks or Pathway 0
- This practicum requires a prior knowledge of Python coding and the basics of NLP

Natural Language Processing

Teaching computers to understand the written word

Natural Language Processing (NLP)

Can you teach a computer to read and write?

If you want to teach a computer to read and write, you will have to start with getting it to understand grammar. To do this, you will need to break down paragraphs into sentences, sentences into words, and words into numbers, called tokens.

Why numbers?

- Computers represent information as bits (1s and 0s – on and off circuits) and bytes (collection of bits). For example, 1000001 is the binary representation of the number 65, which is the ASCII value (or numeric value) for the capital letter A.
- With a 16-bit value, you can represent 65,535 distinct letters, plenty for any of Earth's written languages.
- **To represent words in numbers,** you can either encode each letter, or, with a little bit more math, you could encode chunks of words, based on a set of rules, into larger numbers.
- For example, you might decide that four letter combinations (of which there are 26^4 = 456,976 possible combinations in English) could be encoded using 19-bit values. 3 of those would fit in a 64-bit value. So, you could encode any sequence of word chunks, called tokens, that are 1 to 4 characters long into a 64-bit value, called a vector.
- Different algorithms use different rules to determine chunk size, distribution, and so on.
- Once you've encoded the word chunks (tokens) as numbers, and broken a document down into vectors, you will also need to encode the part of speech (noun, verb, adjective, etc.). This then lets you start to look for patterns in the data.

ASCII Table

000	nul	001	soh	002	stx	003	etx	004	eot	005	enq	006	ack	007	bel
008	bs	009	ht	010	nl	011	vt	012	np	013	cr	014	so	015	si
016	dl	017	dc1	018	dc2	019	dc3	020	dc4	021	nak	022	syn	023	etb
024	can	025	em	026	sub	027	esc	028	fs	029	gs	030	rs	031	us
032	sp	033	!	034	"	035	#	036	$	037	%	038	&	039	'
040	(	041	)	042	*	043	+	044	,	045	-	046	.	047	/
048	0	049	1	050	2	051	3	052	4	053	5	054	6	055	7
056	8	057	9	058	:	059	;	060	<	061	=	062	>	063	?
064	@	065	A	066	B	067	C	068	D	069	E	070	F	071	G
072	H	073	I	074	J	075	K	076	L	077	M	078	N	079	O
080	P	081	Q	082	R	083	S	084	T	085	U	086	V	087	W
088	X	089	Y	090	Z	091	[	092	\	093	]	094	^	095	_
096	`	097	a	098	b	099	c	100	d	101	e	102	f	103	g
104	h	105	i	106	j	107	k	108	l	109	m	110	n	111	o
112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124	|	125	}	126	~	127	del

Tokenization

Tokenization is the process of breaking down a text into the smallest unit, called a token.

Punctuation marks, words, syllables, and numbers can be considered tokens.

Stemming

Stemming is the process of finding the root of words.

Stop Words

Stop words are all the common words, such as ‘a’, ‘in’, ‘the’. Stop word lists are often used to exclude common words from NLP tasks.

Lemmatization

Lemmatization is the process of finding the form of the related word in the dictionary. It is different from Stemming. It involves longer processes to calculate than Stemming.

As opposed to stemming, lemmatization does not simply chop off inflections. Instead, it uses lexical knowledge bases to get the correct base forms of words.

Example:

Word	Stemming	Lemmatization
information	inform	information
informative	inform	informative
computers	comput	computer
feet	feet	foot

Part of Speech Tagging

Part of Speech Tagging (POS-Tag) is the labeling of the words in a text according to their word types (noun, adjective, adverb, verb, etc.).

NLP Tasks

NLP is often used to perform tasks such as:

- Sentiment Analysis
- Translation
- Text Classification (such as Spam filtering)
- Named Entity Recognition
- Speech Recognition.

It is the core technology used in building Chatbots.

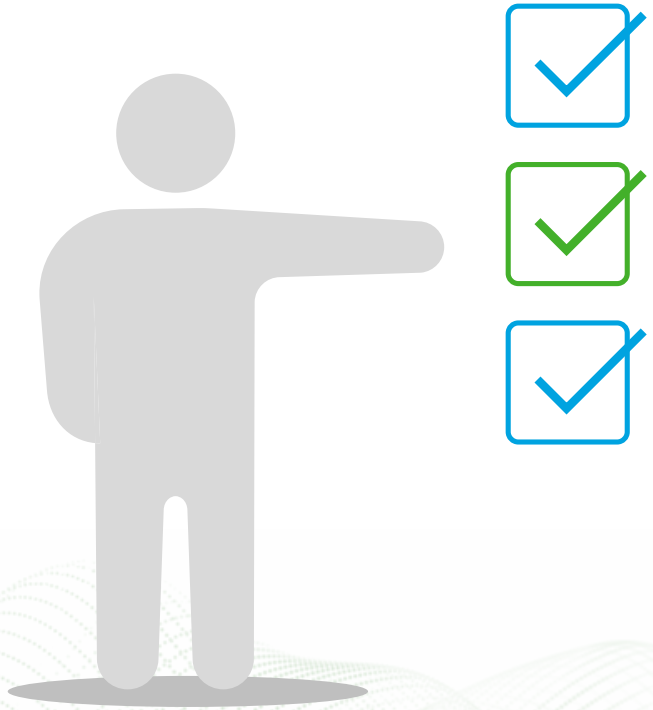
Lab 1
NLP Tasks
(30 min)

Libraries to use:

- TextBlob
- NLTK
- Translate
- Spacy
- WordCloud

Use the NLP libraries to perform the following tasks:

1. Create a Word List
2. Tokenize it
3. Use a Stemmer and a Lemmatizer
4. Filter out Stop Words
5. Identify the Parts of Speech
6. Perform Sentiment Analysis
7. Translate Phrase into a different language
8. Use NER to find the important Nouns in a phrase



Neural Networks

Advanced processing for text and image generation

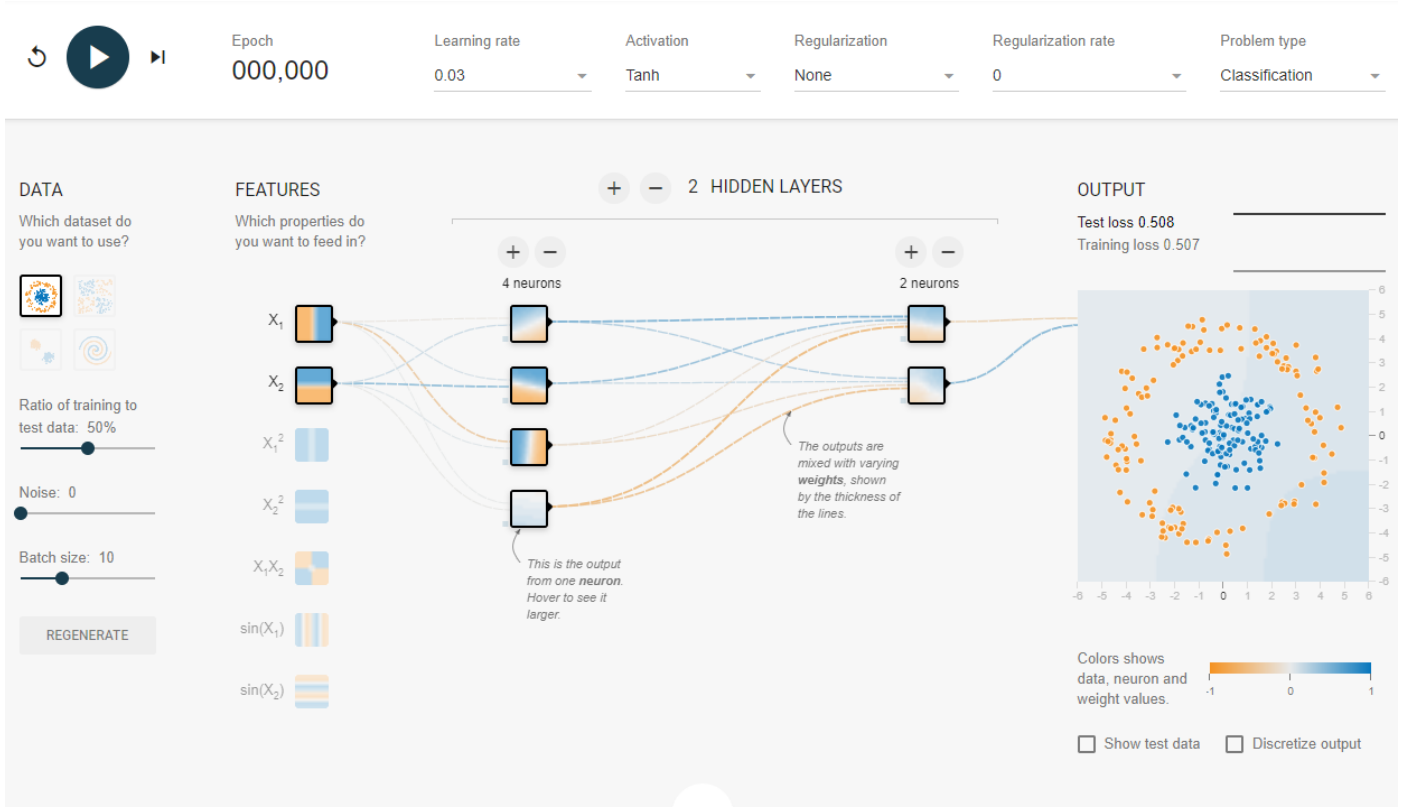
Neural Networks

Solving more complex language problems can require Neural Networks.

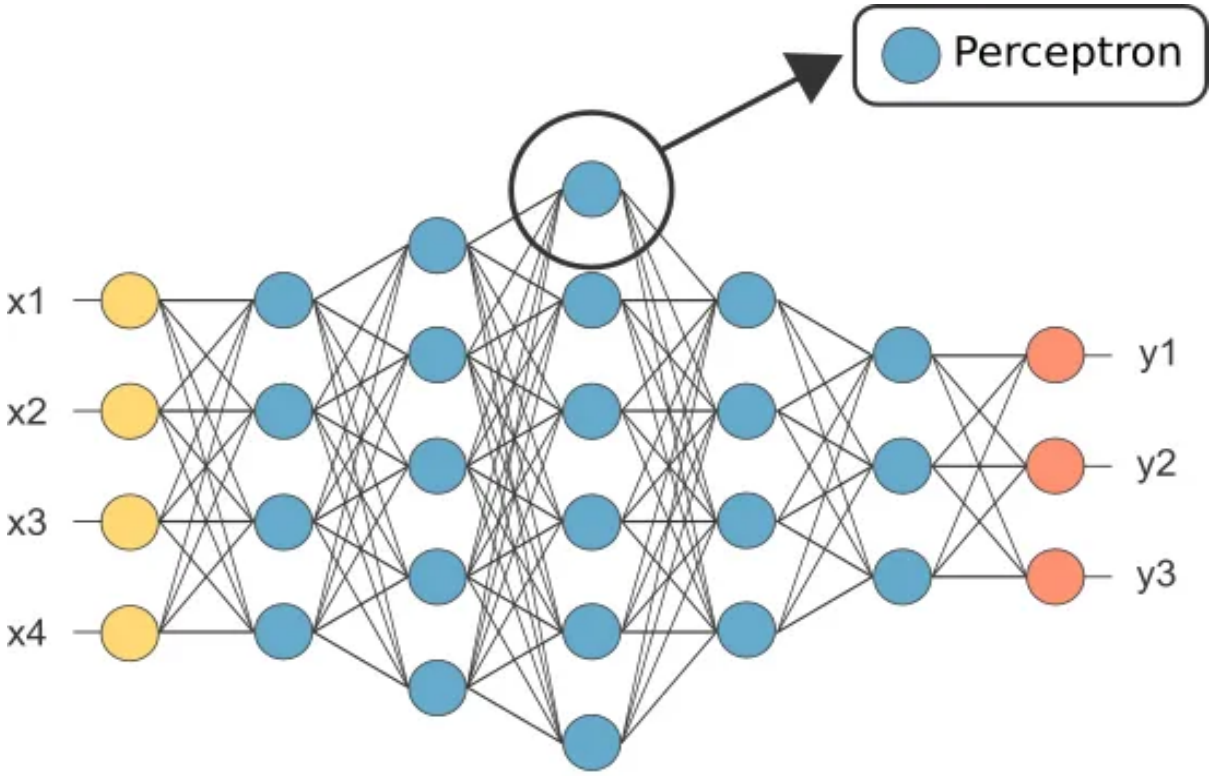
These consist of nodes, called Neurons, organized into Layers. Each neuron receives input(s), performs some computation, and produces output(s) that are passed on to other neurons.

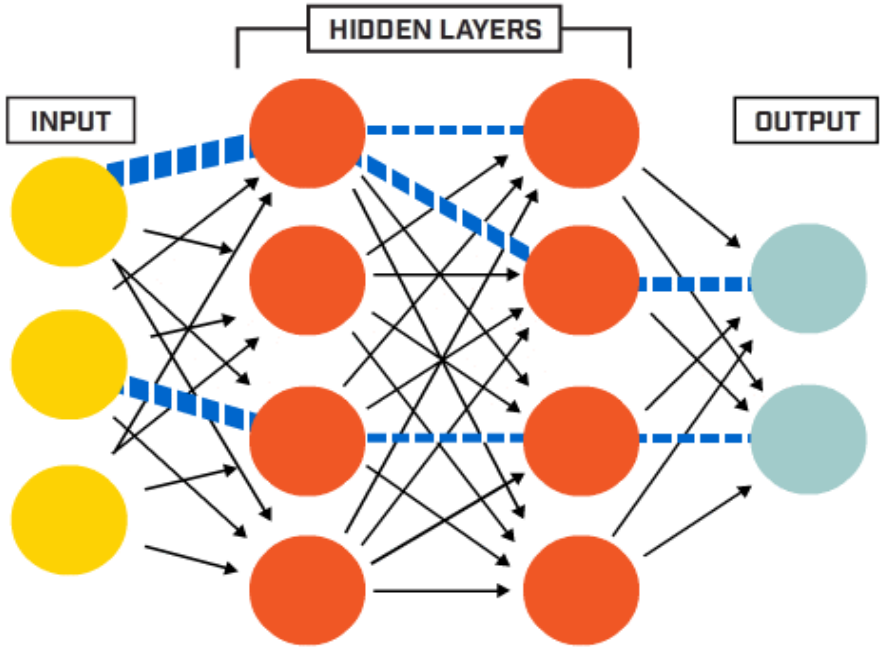
At the least there is an input layer and an output layer, but there may be one or more hidden layers in between.

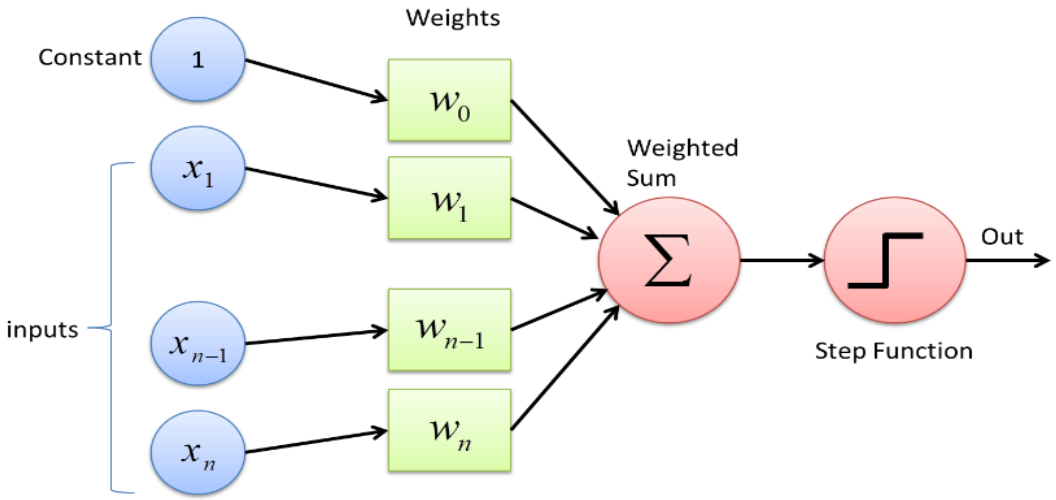
https://playground.tensorflow.org



Feed Forward Neural Network



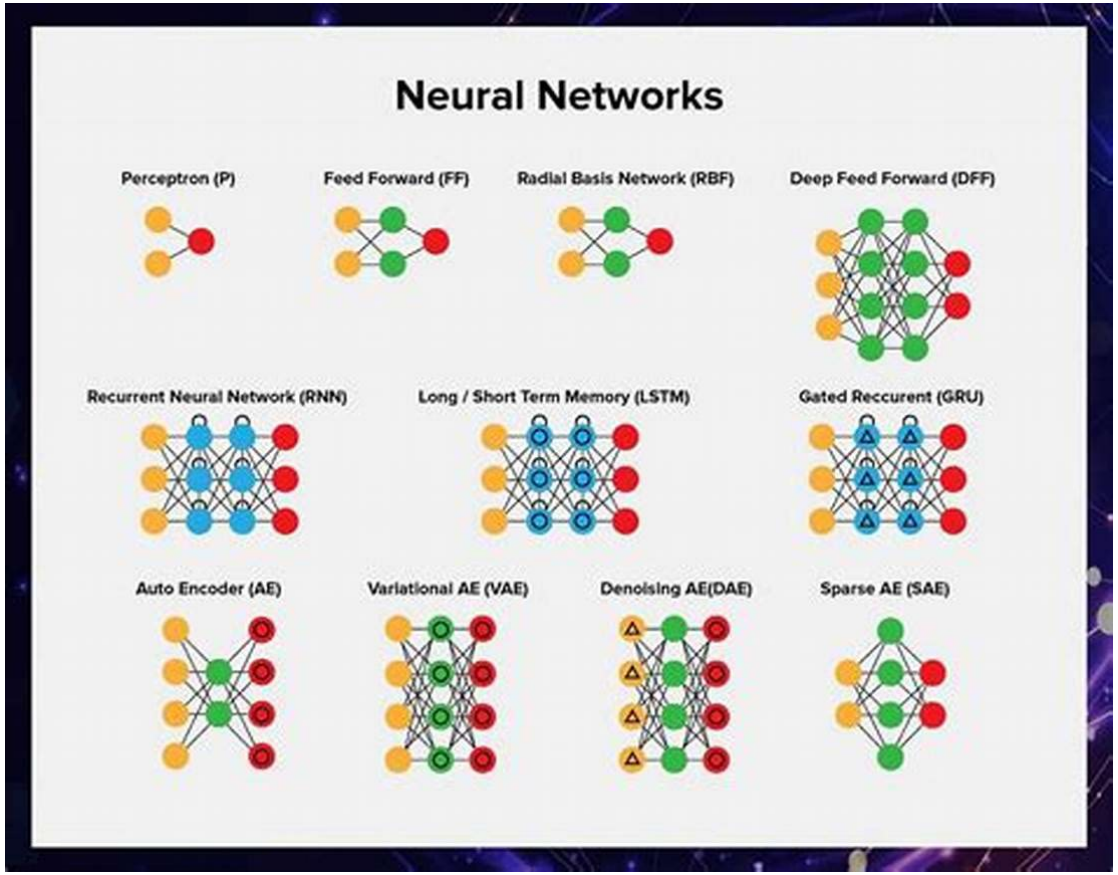




Types of Neural Networks

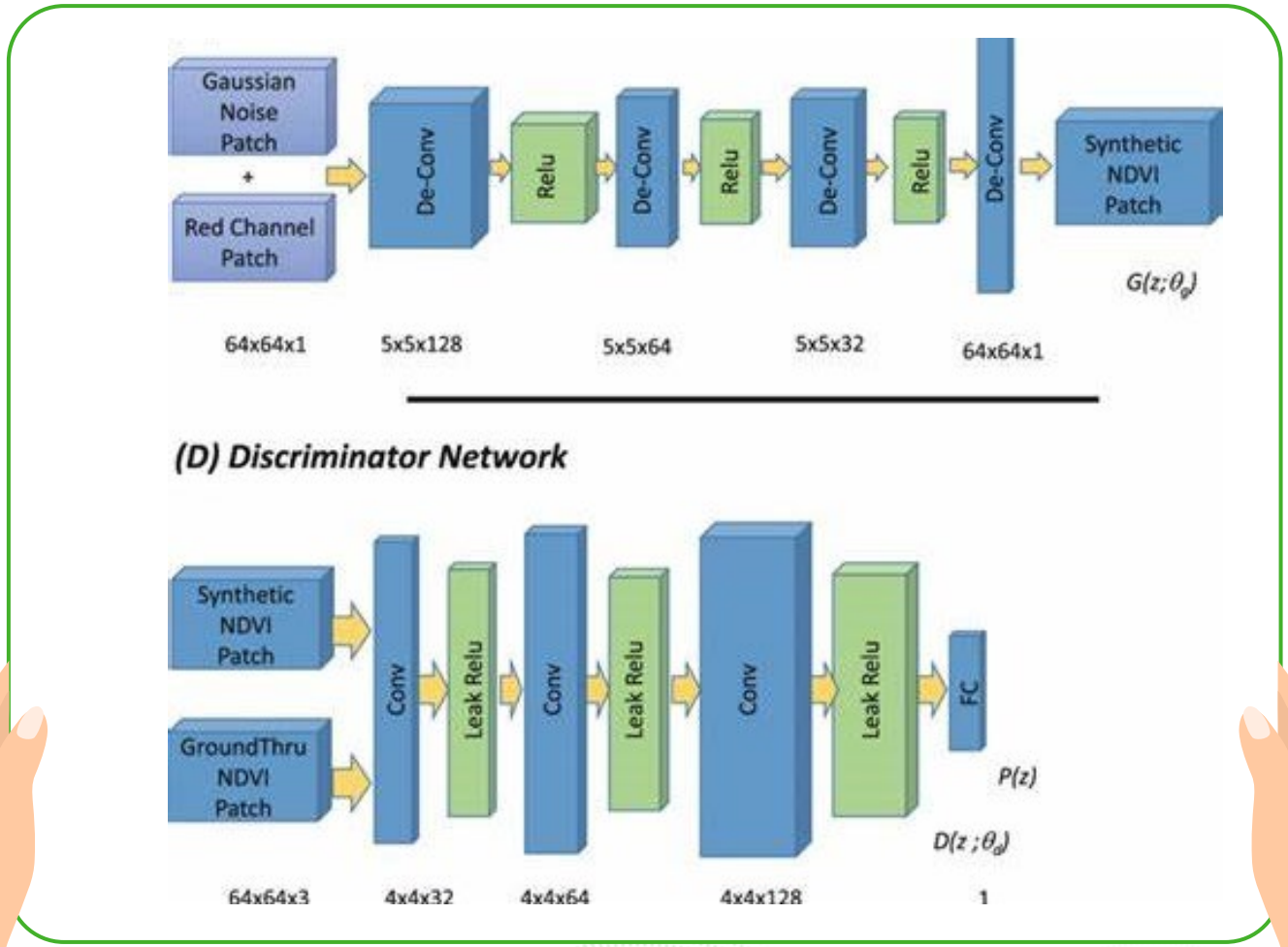
There are several different types of Neural Networks, each designed for specific tasks and applications. Here are some of the most common:

- Convolutional Neural Network (CNN): CNNs are widely used for image and video processing tasks.
- Recurrent Neural Network (RNN): RNNs are designed to process sequential data, such as natural language or time series data.
  - Long Short-Term Memory (LSTM): LSTMs are a type of RNN that address the vanishing gradient problem. They are capable of learning long-term dependencies and are often used for tasks like speech recognition and machine translation.
  - Gated Recurrent Unit (GRU): GRUs are a type of RNN that addresses the vanishing gradient problem like LSTMs. They have a gating mechanism that allows them to selectively update and forget information. GRUs are often used for tasks like natural language processing and speech recognition.



Conditional generative adversarial architecture

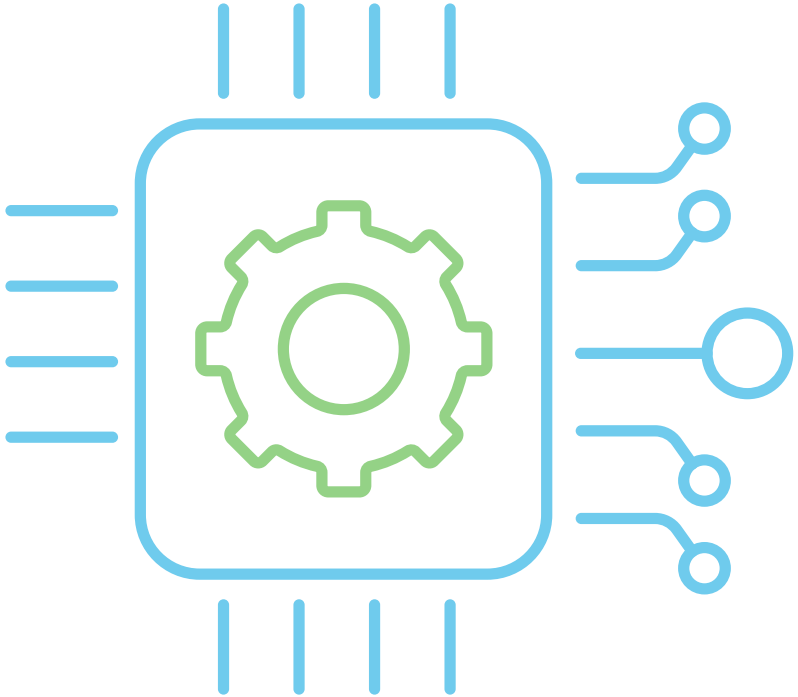
(G) Generator network



Types of Generative Neural Networks

As we advance from RNNs to LSTMs and GRUs, we begin to move into the Generative Neural Networks. These are Neural Networks that can begin to generate outputs that are not just a function of the inputs. Some of the most common include:

- Generative Adversarial Network (GAN): GANs consist of two neural networks, a generator and a discriminator, that compete with each other in a game-theoretic framework. GANs are used for tasks like generating realistic images or data synthesis.
- Autoencoder: Autoencoders are neural networks that aim to learn efficient representations of input data. They consist of an encoder that compresses the input into a lower-dimensional representation, and a decoder that reconstructs the input from this representation.
- Radial Basis Function Network (RBFN): RBFNs are often used for function approximation and pattern recognition tasks. They use radial basis functions as activation functions and are particularly effective in solving problems with nonlinear separability.



Lab 2
Create a GAN
(15 min)

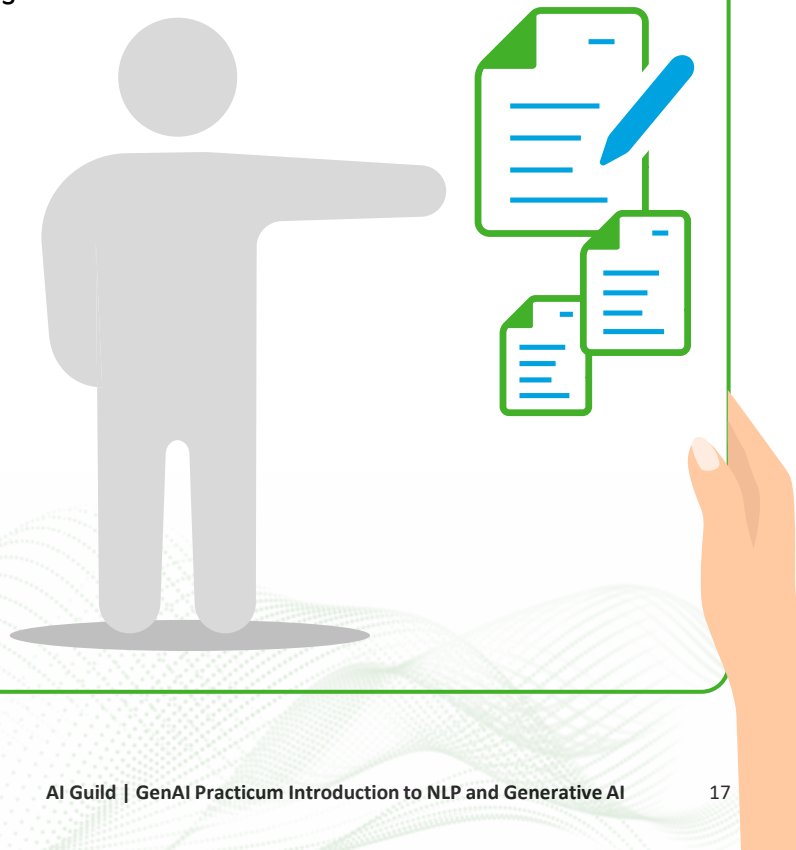
Main libraries to use:

- TensorFlow
- Image.io

Use TensorFlow to build a simple Gan that generates images of handwritten digits:

1. Create a Generator model and a Discriminator model
2. Create the loss functions
3. Create the training step function
4. Create a function to generate and save the images
5. Create the main training function
6. Load the sample MNIST dataset
7. Run the training function for 50 Epochs
8. Assemble an animated GIF showing the progressively better images

To learn more about GANs,
see MIT's Intro to Deep Learning course.



Transformer Models

"Attention is All You Need"

Transformers Models

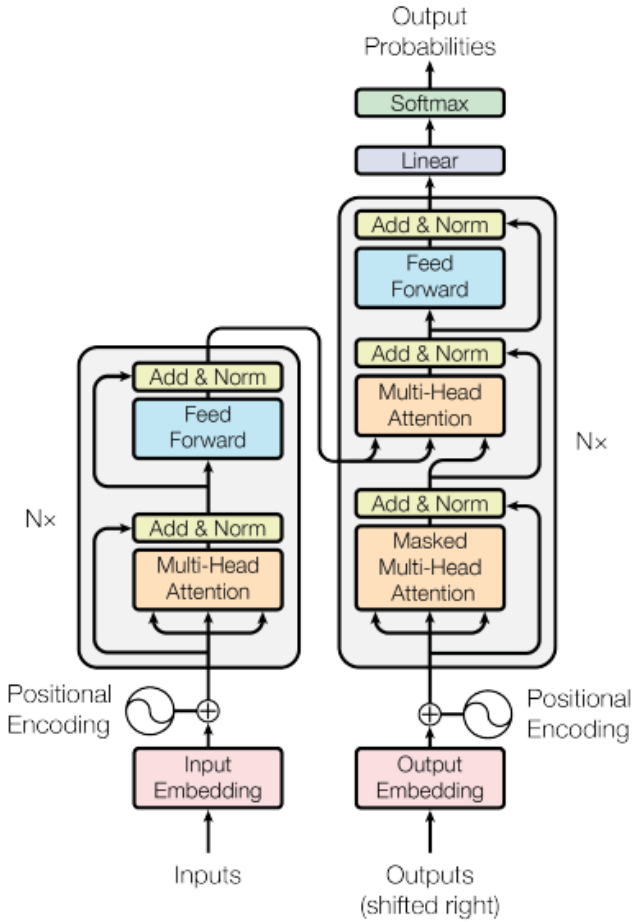
“Attention is all you need”

Transformer models are a class of deep learning models that were introduced in 2017 by Vaswani et al.

A key innovation within Transformers is the attention mechanism, which allows the model to focus on relevant parts of the input sequence.

The attention mechanism is at the core of the Transformer's success in various complex NLP tasks, such as text generation and language translation.

Source: "Attention is All You Need" is a research paper published in 2017 by Google researchers.



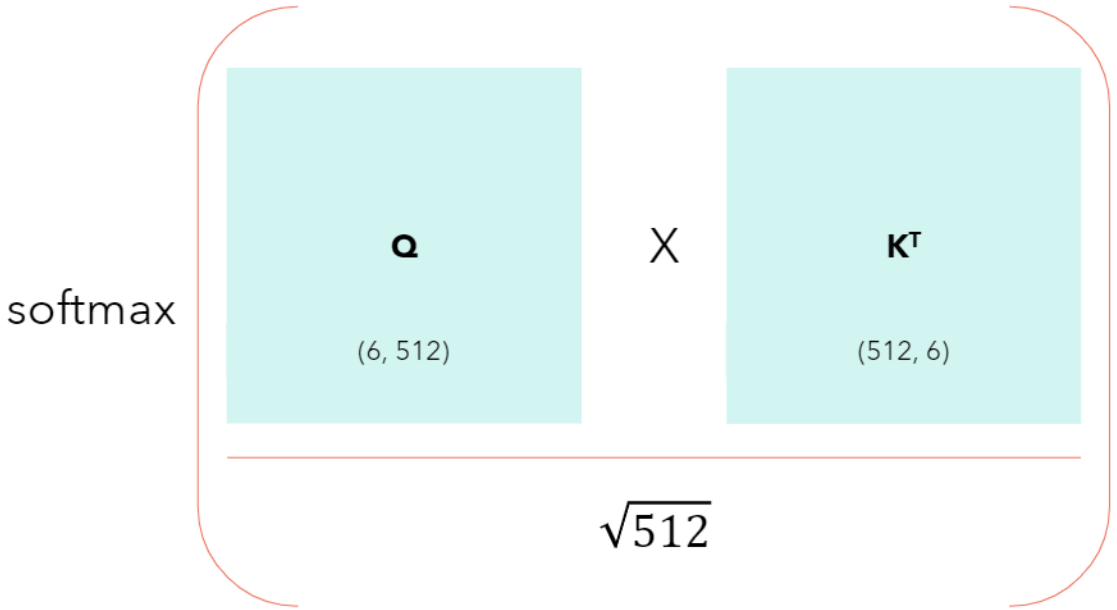
What is Self-Attention?

Self-Attention allows the model to relate words to each other.

In this simple case we consider the sequence length **seq** = 6 and $\mathbf{d_{model}} = \mathbf{d_k} = 512$.

The matrices **Q**, **K** and **V** are just the input sentence.

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



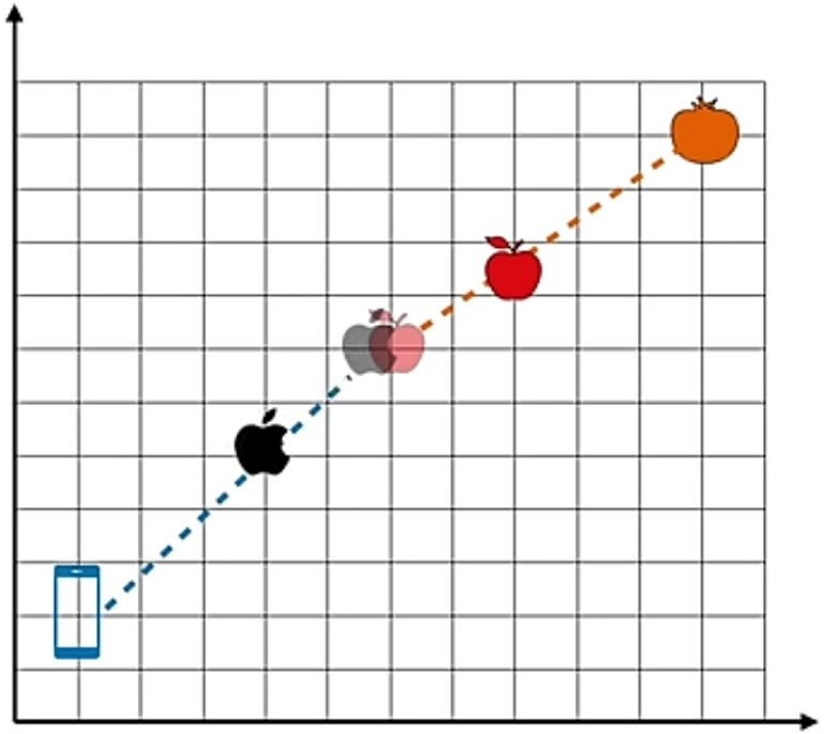
	YOUR	CAT	IS	A	LOVELY	CAT	Σ
YOUR	0.268	0.119	0.134	0.148	0.179	0.152	1
CAT	0.124	0.278	0.201	0.128	0.154	0.115	1
IS	0.147	0.132	0.262	0.097	0.218	0.145	1
A	0.210	0.128	0.206	0.212	0.119	0.125	1
LOVELY	0.146	0.158	0.152	0.143	0.227	0.174	1
CAT	0.195	0.114	0.203	0.103	0.157	0.229	1

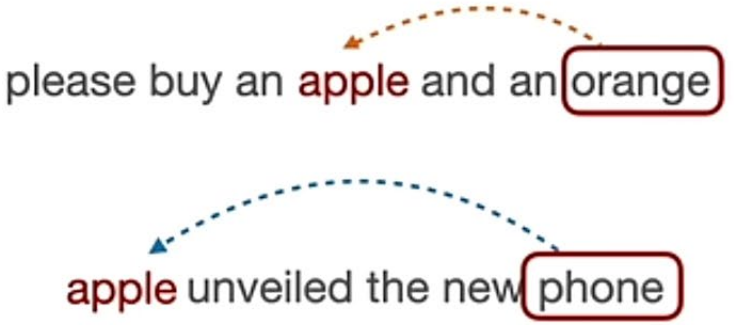
(6, 6)

* all values are random.

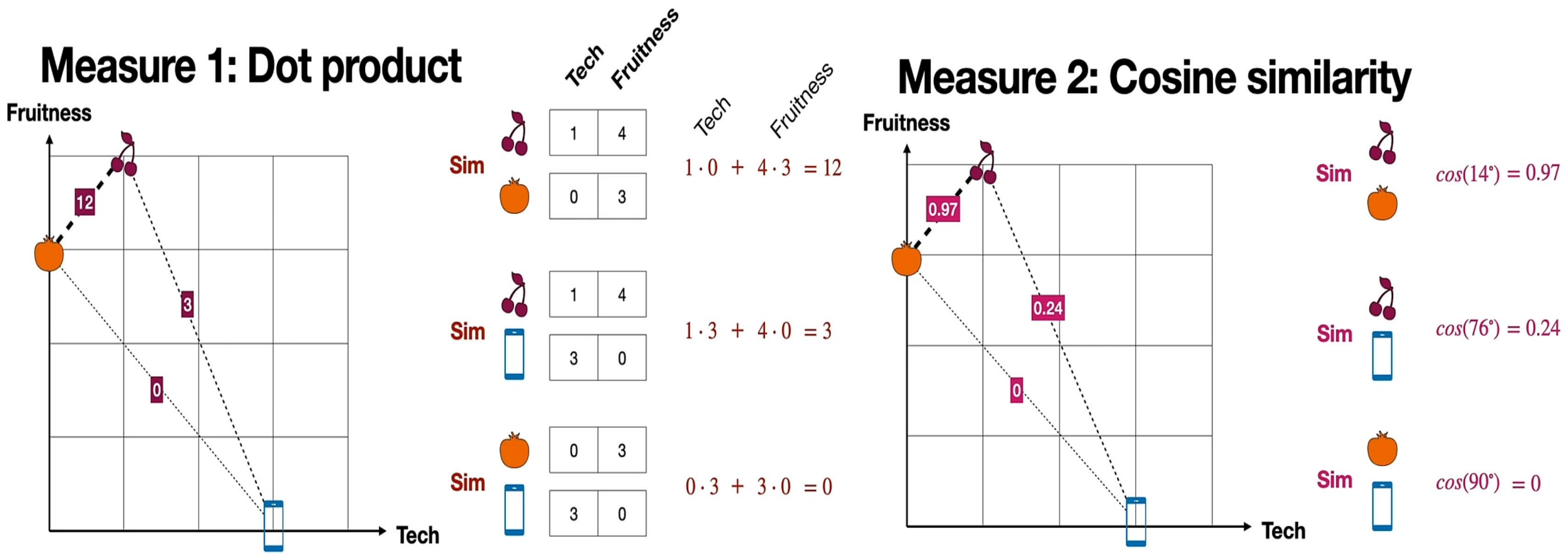
* for simplicity I considered only one head, which makes $d_{model} = d_k$.

Words pulling words





Similarity measures



Measure 3: Scaled dot product

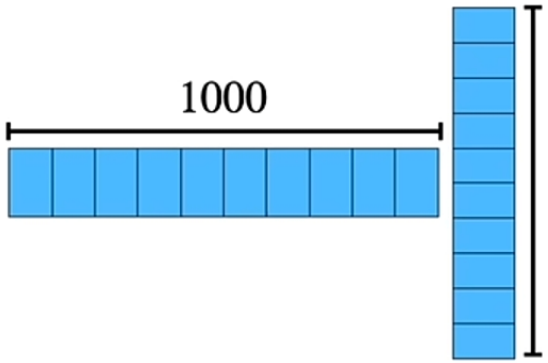
Dot product divided by the square root of the length of the vector

Sim

1	4

0	3

$$1 \cdot 0 + 4 \cdot 3 = 12 \longrightarrow \frac{12}{\sqrt{2}} = 8.49$$



Sim

1	4

3	0

$$1 \cdot 3 + 4 \cdot 0 = 3 \longrightarrow \frac{3}{\sqrt{2}} = 2.12$$

Sim

0	3

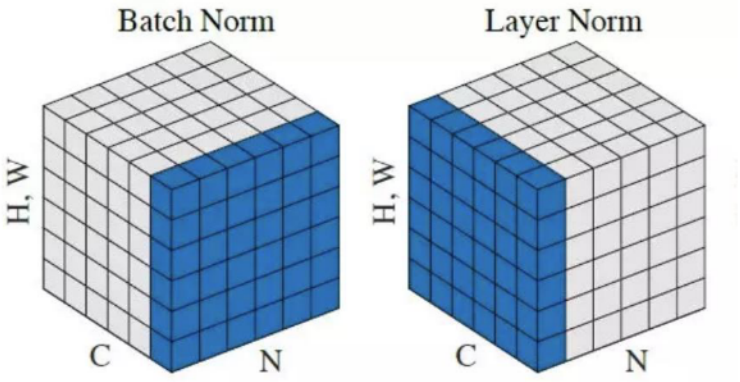
3	0

$$0 \cdot 3 + 3 \cdot 0 = 0 \longrightarrow \frac{0}{\sqrt{2}} = 0$$

What is layer normalization?

Batch of 3 items

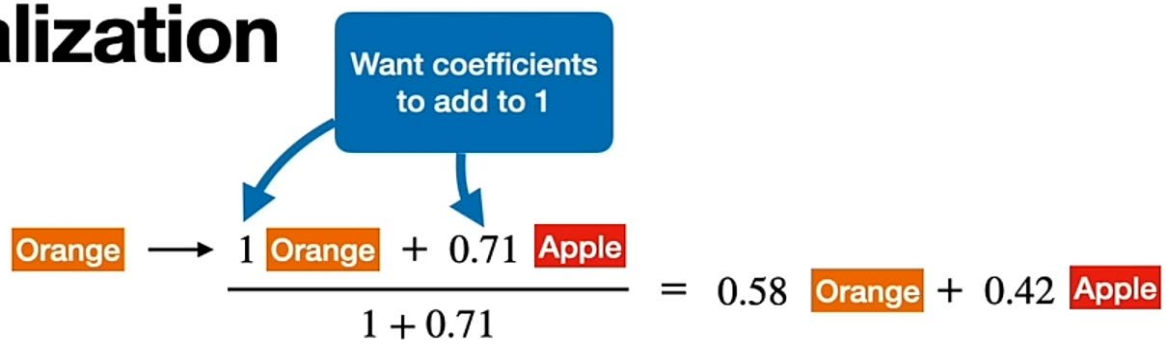
ITEM 1	ITEM 2	ITEM 3
50.147	1242.223	9.370
3314.825	688.123	4606.674
...	...	...
...	...	...
8463.361	434.944	944.705
8.021	149.442	21189.444
μ_1	μ_2	μ_3
σ_1^2	σ_2^2	σ_3^2



$$\hat{x}_j = \frac{x_j - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

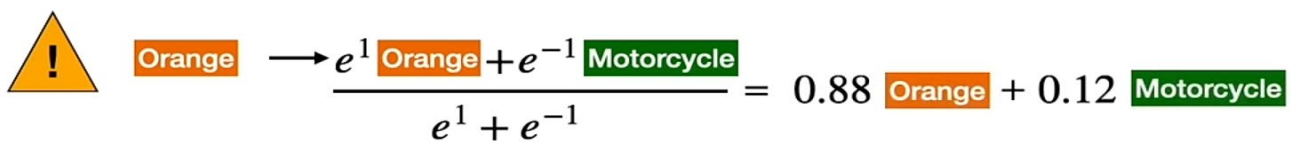
We also introduce two parameters, usually called **gamma** (multiplicative) and **beta** (additive) that introduce some fluctuations in the data, because maybe having all values between 0 and 1 may be too restrictive for the network. The network will learn to tune these two parameters to introduce fluctuations when necessary.

Normalization



Softmax

$$x \longrightarrow e^{x}$$



Differences in Transformer Models

"Attention is all you need"

There are several different kinds of transformer models that have been developed over time, including:

1. Transformer: This is the original transformer model introduced in the paper "Attention is All You Need." It consists of an encoder-decoder architecture with self-attention mechanisms.
2. BERT (Bidirectional Encoder Representations from Transformers): BERT is a transformer-based model that is pretrained on large amounts of unlabeled text data. It has achieved state-of-the-art performance on various natural language processing tasks, such as question answering and sentiment analysis.
3. GPT (Generative Pre-trained Transformer): GPT is a transformer-based language model that uses unsupervised learning to generate coherent and contextually relevant text. It has been widely used for tasks like text completion and text generation.
4. T5 (Text-to-Text Transfer Transformer): T5 is a transformer-based model that can be trained in a "text-to-text" framework, where a wide range of NLP tasks are framed as text-to-text problems. It has achieved impressive results on various tasks, including machine translation, text summarization, and question answering.
5. RoBERTa (Robustly Optimized BERT): RoBERTa is a variant of BERT that addresses some of its limitations and improves its performance. It uses larger training datasets and longer training times to achieve better results on various NLP tasks.

GANs vs Transformers

Transformers and GANs are two different types of models with different purposes and strengths. It is not accurate to say that transformers are universally better than GANs or vice versa. However, transformers have several advantages over GANs in certain tasks:

1. Sequential data processing: Transformers are particularly effective for processing sequential data, such as natural language processing tasks, where the order of the input is crucial. GANs, on the other hand, are primarily used for generating new data samples.
2. Attention mechanism: Transformers utilize attention mechanisms, which allow the model to focus on relevant parts of the input sequence. This attention mechanism helps transformers capture long-range dependencies in the data, making them effective for tasks that require understanding of global context.
3. Transfer learning: Transformers can be pre-trained on large amounts of data and then fine-tuned for specific downstream tasks. This pre-training and fine-tuning process allows transformers to learn general representations that can be transferred to various tasks, making them adaptable and efficient for different applications.

It is important to note that the choice between transformers and GANs depends on the specific task and requirements. GANs excel in generative tasks, such as image and video synthesis, while transformers are often preferred for sequence-to-sequence tasks, such as machine translation and text generation.

Introducing: Generative AI

LLMs with Billions of Parameters

Next word prediction: Lions, and Tigers, and _____.

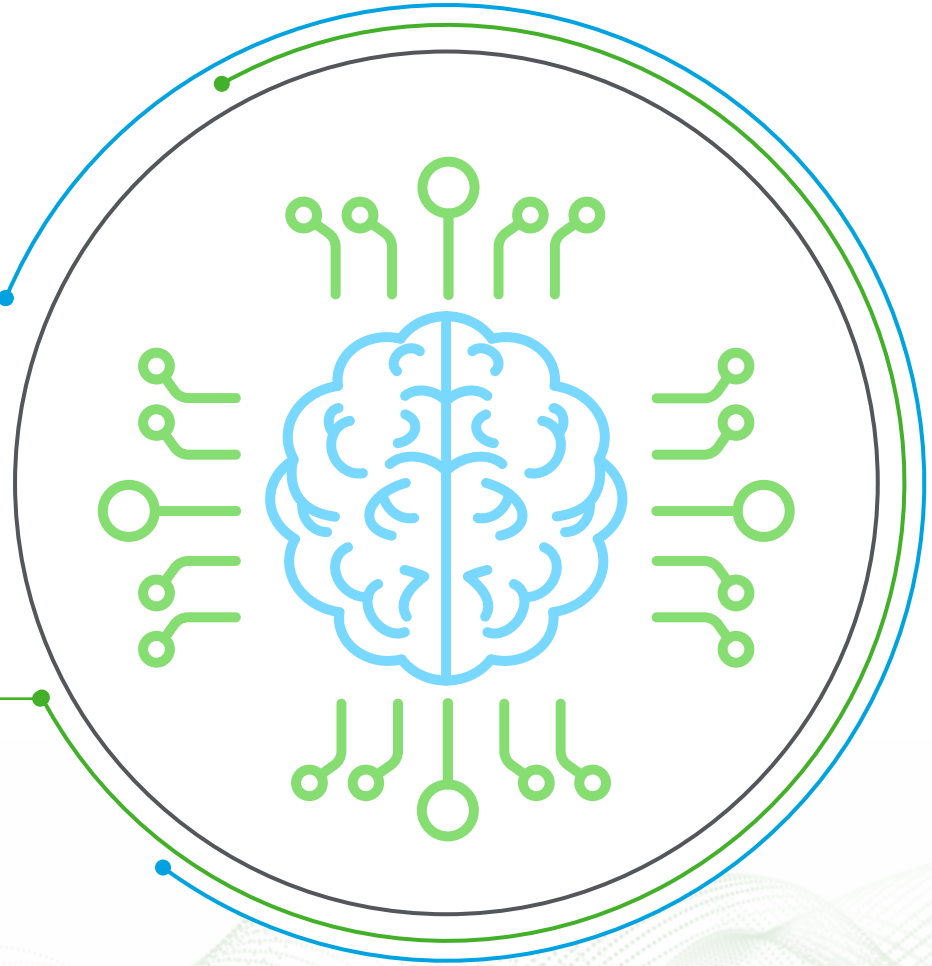
Introducing: Generative AI

LLMs with Billions of Parameters
Next word prediction: Lions, and Tigers, and ***Bears, oh my***.

What is Generative AI or GenAI?

Generative AI is a set of Artificial Intelligence technologies that can create (Generate) net new content.

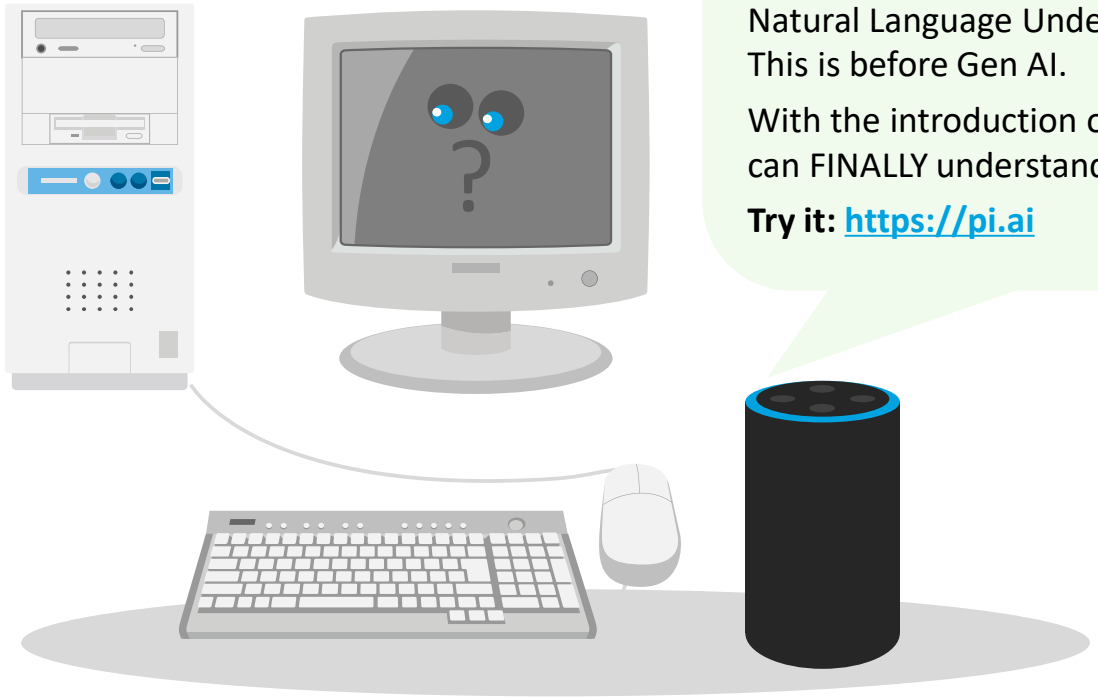
Most Artificial Intelligence algorithms can match, categorize, and predict based on a given set of data. **Generative AI** goes **beyond predictions** to create **new content** such as text, images, and more, based on patterns learned from existing data.



Why Should I care?

Generative AI is going to change white collar work more than any other technology yet invented.

Remember when computers couldn't understand us?



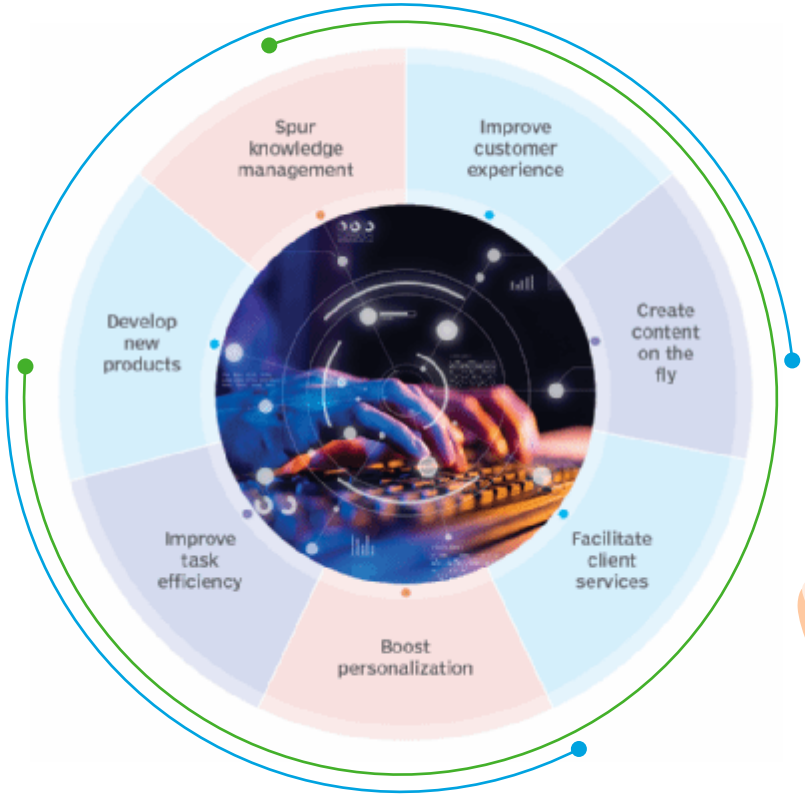
Have you ever said, "Alexa, play my music. No, Alexa, play my music! **ALEXA!!! PLAY MY MUSIC!!!!!**"

Those are voice experiences that use Natural Language Understanding (NLU). This is before Gen AI.

With the introduction of Gen AI, computers can FINALLY understand us.

Try it: https://pi.ai

Generative AI benefits for business



So What? I can chat with a machine? What else?

Introducing Copilots, Assistants, and Sidekicks

What else can a computer understand now?

- Web pages
- White papers
- Contracts
- International Tax Laws
- Medical Papers and Doctors Notes
- Multiple Languages
- Your email
- Your calendar
- Your Word documents
- Your PowerPoint decks
- Your hopes
- Your dreams
- Your thoughts?

Introducing your new Copilot.

You need to write an email to a new LinkedIn contact introducing yourself and commenting on their recent post.

- ***Your LinkedIn Copilot can help with that.***

You need to create a short PowerPoint deck based on the report you finished writing yesterday.

- ***Your Office Copilot can help with that.***

You need to write some Python code to analyze a set of spreadsheets provided by the client.

- ***Your GitHub Copilot can help with that.***

You need to come up with a bedtime story for your 3-year-old and you're suffering writer's block.

- ***Your Custom GPT can help with that.***

You need to understand the gist of the new law passed by the EU regarding Generative AI.

- ***There's a Copilot for that too.***

How is Gen AI going to change my job?

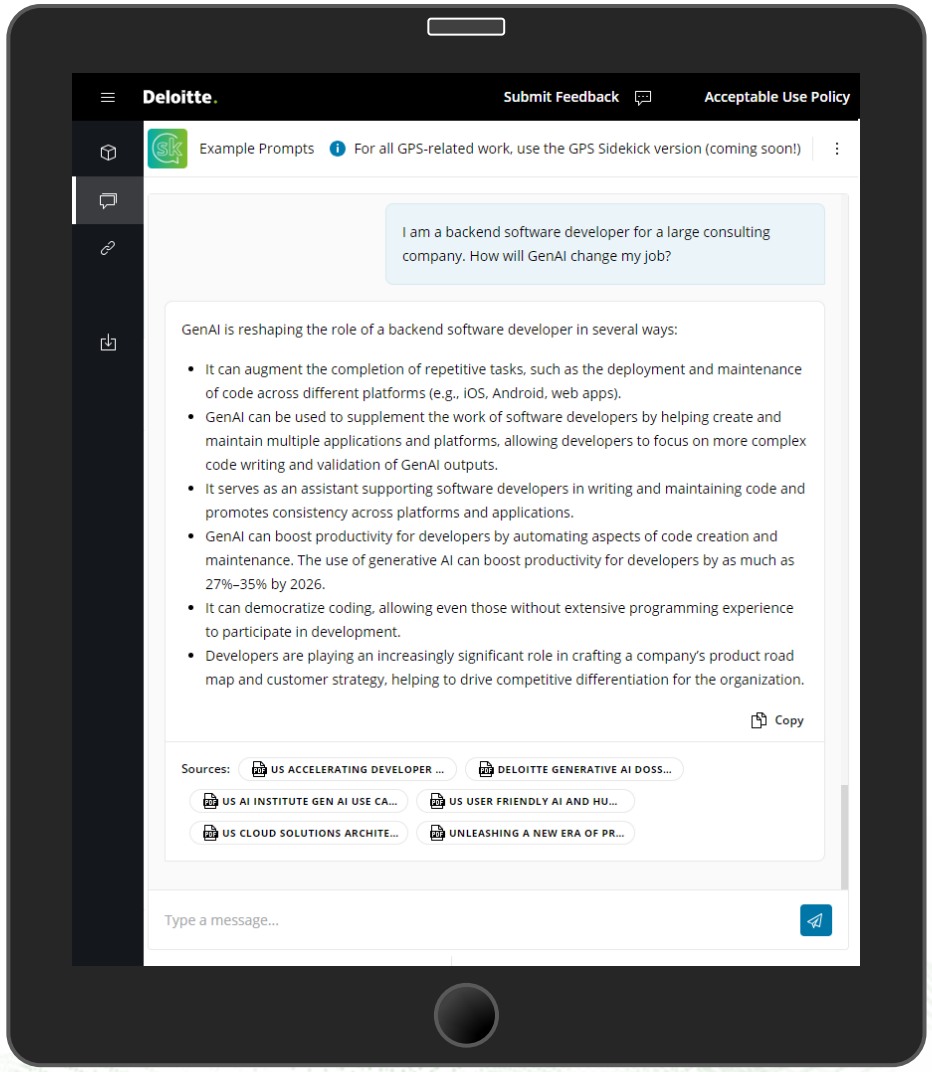
Prompt Engineering, Hallucinations, and keeping a Human-in-the-loop

Whatever you do that is routine, straight forward, or repetitive, AI will likely be doing a lot of that. Everything else, you will still be doing.

- You will need to learn a new skill: **Prompt Engineering**. Crafting a prompt that gives you the best output from GenAI is an artform.
- You will need to learn how to spot **Hallucinations**.
- When you use Gen AI, you will have to validate the
- responses. Fact check. Don't believe your senses.
- You will be the **Human-in-the-loop (HITL)**.
- When you use Gen AI, you will be responsible
- for making sure the AI doesn't go off the rails.

Try it: https://sidekick.deloitte.com

Ask Sidekick how Gen AI can be used in your job.



AI Ethics

"With great power, comes great responsibility." – Ben Parker

AI Ethics

Things To Watch For:

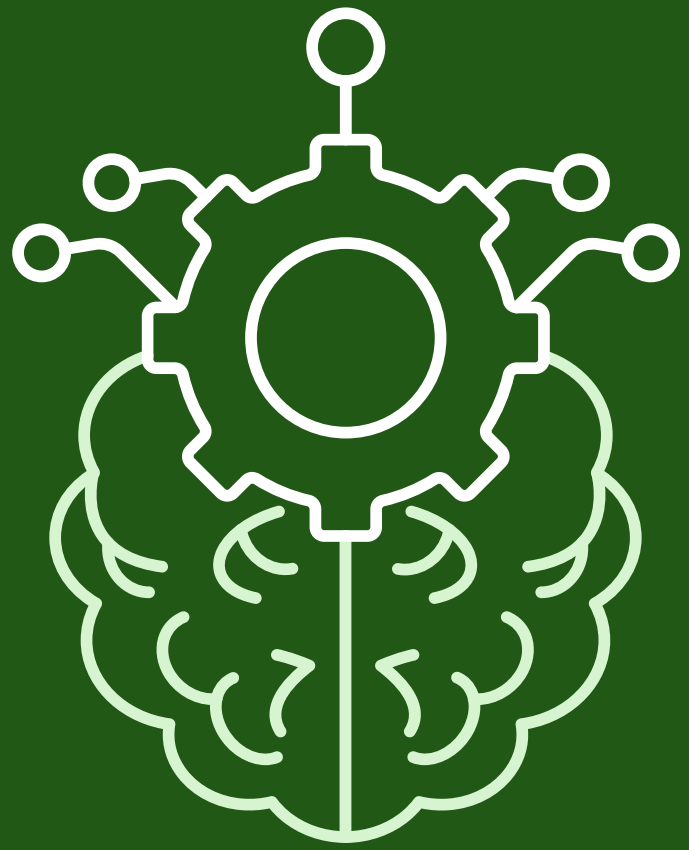
- Bias And Errors (Hallucinations)
- Privacy And Sensitive Data (Where Does Your Prompt Go?)
- Authenticity (Generated Or Human Produced)
- Copyright And Data Ownership (Generated From What?)

Places To Go

- The Deloitte AI Academy
- The US AI SGO
- The Deloitte Generative AI Site (Internal)
- The Deloitte Generative AI Site (External)
- Deloitte Trustworthy AI™
- Deloitte Generative AI Viva Engage Community
- Open Voice Network Trustmark Initiative

AI will not replace humans. Humans using AI will replace Humans that do not.

— Bennie Seybold, Deloitte Futurist



Large Language Models

LLMs, Foundation Models, and Billions of Parameters

What are LLMs?

1

LLMs – Large Language Models - are deep learning models that can recognize, summarize, translate, predict, and generate content using large datasets.

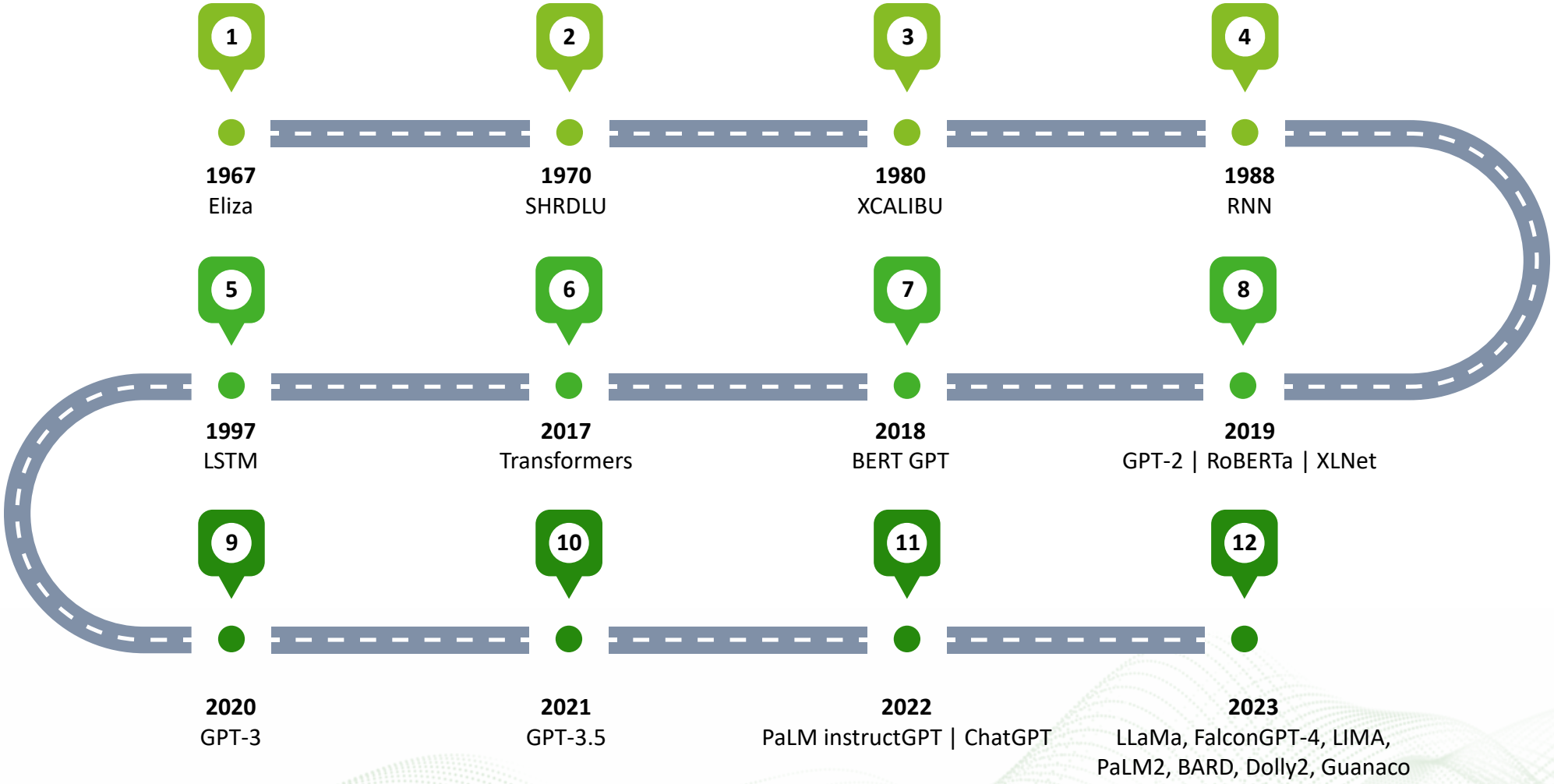
2

The term ‘large’ in LLMs refers to both the volume of data they’re trained on and their size, measured by the number of parameters they possess.

3

They largely represent a class of deep learning architectures called transformer networks.

Evolution of large language models



Comparative Analysis of LLMs

Model	Provider	Context Window	Open-Source	Price / Million	Price	Quality	Speed
GPT-4o	OpenAI	128k	No	7.5	★☆☆	★★★	★★☆
GPT-4 Turbo	OpenAI	128k	No	15	★☆☆	★★★	★☆☆
GPT-4	OpenAI	8k	No	37.5	★☆☆	★★★	★★☆
GPT-3.5 Turbo	OpenAI	16k	No	0.75	★★☆	★☆☆	★★☆
Gemini 1.5 Pro	Google	1m	No	5.25	★★☆	★★★	★★☆
Gemini 1.5 Flash	Google	1m	No	0.53	★★★	★★☆	★★★
Gemma 7B	Google	8k	Yes	0.2	★★★	★☆☆	★★★
Claude 3 Opus	Anthropic	200k	No	30	★☆☆	★★★	★☆☆
Claude 3 Sonnet	Anthropic	200k	No	6	★☆☆	★★☆	★★☆
Claude 3 Haiku	Anthropic	200k	No	0.5	★★★	★★☆	★★★
Command R +	Cohere	128k	Yes	6	★☆☆	★★☆	★★☆
Command R	Cohere	128k	Yes	0.75	★★☆	★☆☆	★★☆
Llama 3 70B	Meta AI	8k	Yes	0.93	★★☆	★★★	★☆☆
Llama 3 8B	Meta AI	8k	Yes	0.2	★★★	★☆☆	★★★
Code Llama	Meta AI	16k	Yes	0.9	★★☆	★★★	★☆☆
Mistral Large	Mistral AI	32k	No	12	★☆☆	★★★	★☆☆
Mistral Medium	Mistral AI	32k	No	4.05	★★☆	★★☆	★☆☆
Mistral Small	Mistral Ai	32k	No	2.25	★★☆	★☆☆	★★☆
Mixtral 8x22B	Mistral AI	65k	Yes	1.2	★★☆	★★☆	★★☆
Mixtral 8x7B	Mistral AI	32k	Yes	0.5	★★★	★☆☆	★★★
Mistral 7B	Mistral AI	32k	Yes	0.2	★★★	★☆☆	★★☆
DBRX	Databricks	32k	Yes	1.4	★★☆	★★☆	★★★

Use Cases For LLMs

- Generation (e.g., story writing, marketing content creation)
- Summarization (e.g., legal paraphrasing, meeting notes summarization)
- Translation (e.g., between languages, text-to-code)
- Classification (e.g., toxicity classification, sentiment analysis)
- Chatbot (e.g., open-domain Q+A, virtual assistants)

What are LLMs?

1. **Citing Sources** – LLMs cannot accurately cite sources. This is because they do not have access to the Internet and do not have the ability to remember where their training data came

2. **Bias –** LLMs can exhibit bias in their responses, often generating stereotypical or prejudiced content.

3. **Hallucinations –** LLMs can sometimes "hallucinate" or generate false information when asked a question they do not know the answer to.

4. **Math –** LLMs often struggle with mathematical tasks and can provide incorrect answers (even as simple as multiply two numbers).

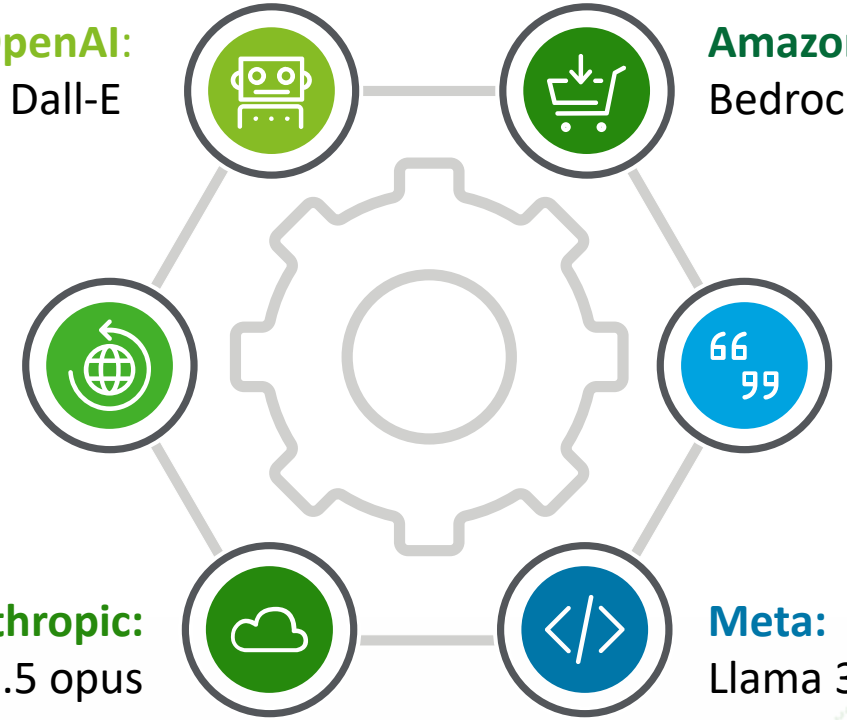
5. **Prompt Hacking –** LLMs can be manipulated or "hacked" by users to generate specific content. This is known as prompt hacking and can be used to trick the LLM into generating inappropriate or harmful content.

LLM Service Providers

What are LLMs?

OpenAI:
GPT-4o, GPT-4, GPT-3.5, Dall-E

Amazon:
Bedrock, Titan



Google:
Gemini 1.5, Gemini 1.0, PaLM 2

Cohere AI:
Customizable Language Models

Anthropic:
Claude 3.5 sonnet, Claude 3.5 opus

Meta:
Llama 3, Llama 2, Code Llama

Unique Strengths of Each LLM Provider

1. **Google (PaLM 2):** Excels in advanced reasoning tasks with a focus on multilingual proficiency and offers the PaLM API for developing generative AI applications.

2. **Anthropic:** Provides a high-performance model, Claude, with capabilities in a variety of use cases like summarization, Q&A, and coding.

3. **Cohere AI**: Offers pre-built LLMs for common NLP tasks and customizable language models for specific needs.

4. **Meta**: Focuses on foundational LLMs like Llama 2 and Code Llama, with Code Llama specializing in code generation through text prompts

GenAI in the Cloud

The different cloud providers have very different service offerings.

01

Microsoft

Based on OpenAI's Foundation Models (FMs), Microsoft Azure AI Studio offers curated, contained, and secure enterprise grade versions of GPT-3, GPT-4, GPT-4o and DALL-E.

02

Amazon AWS

Amazon Bedrock offers an easy-to-use developer experience to work with a broad range of high-performing FMs from leading AI companies like AI21 Labs, Anthropic, Cohere, Meta, Stability AI, and Amazon. You can quickly experiment with a variety of FMs in the playground and use a single API for inference regardless of the models you choose, giving you the flexibility to use FMs from different providers and keep up to date with the latest model versions with minimal code changes.

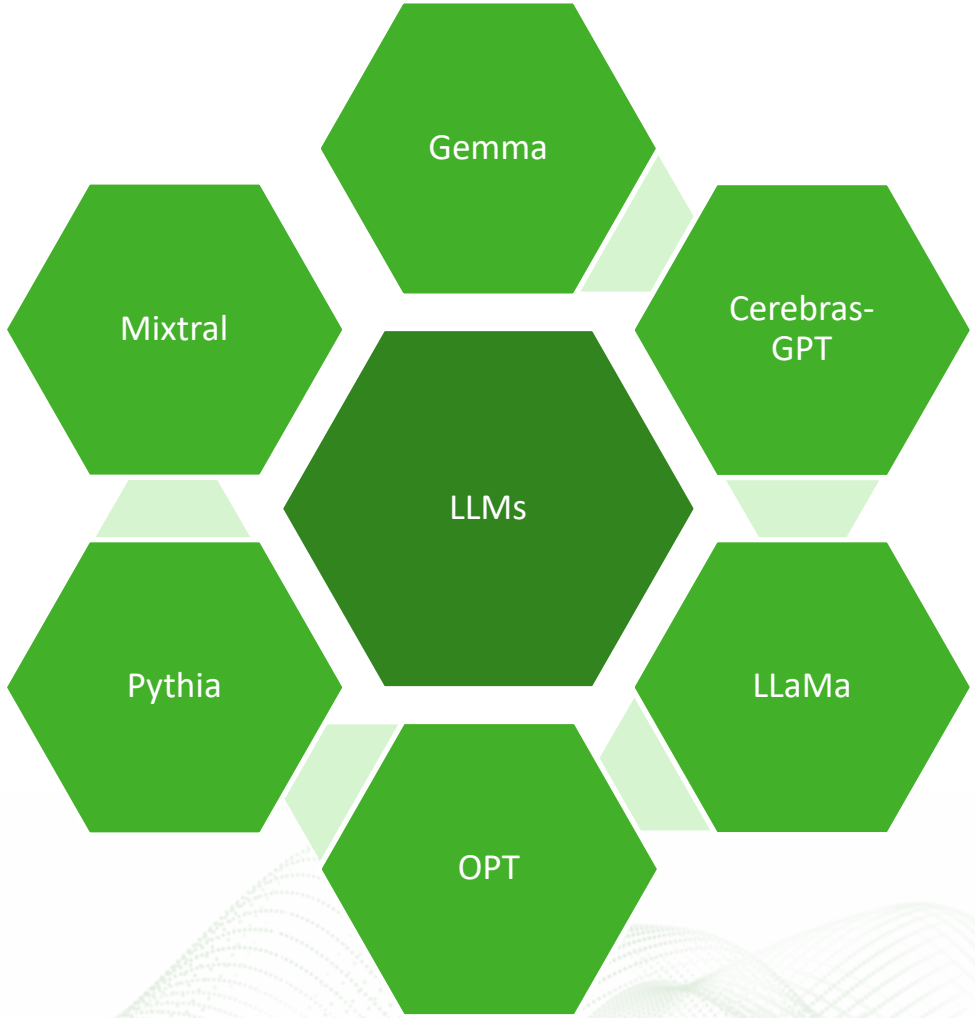
03

Google

Google provides a Model Garden for its own foundation models such as Gemini 1.5, PaLM 2 (Bison), an Embeddings model (Gecko), and Codey for Code Generation.

Open Source LLMs

While the Big Three commercial cloud vendors have their offerings, open-source LLMs are also up and coming, including some popular ones like LLaMa from Meta (Facebook), Gemma from Google.



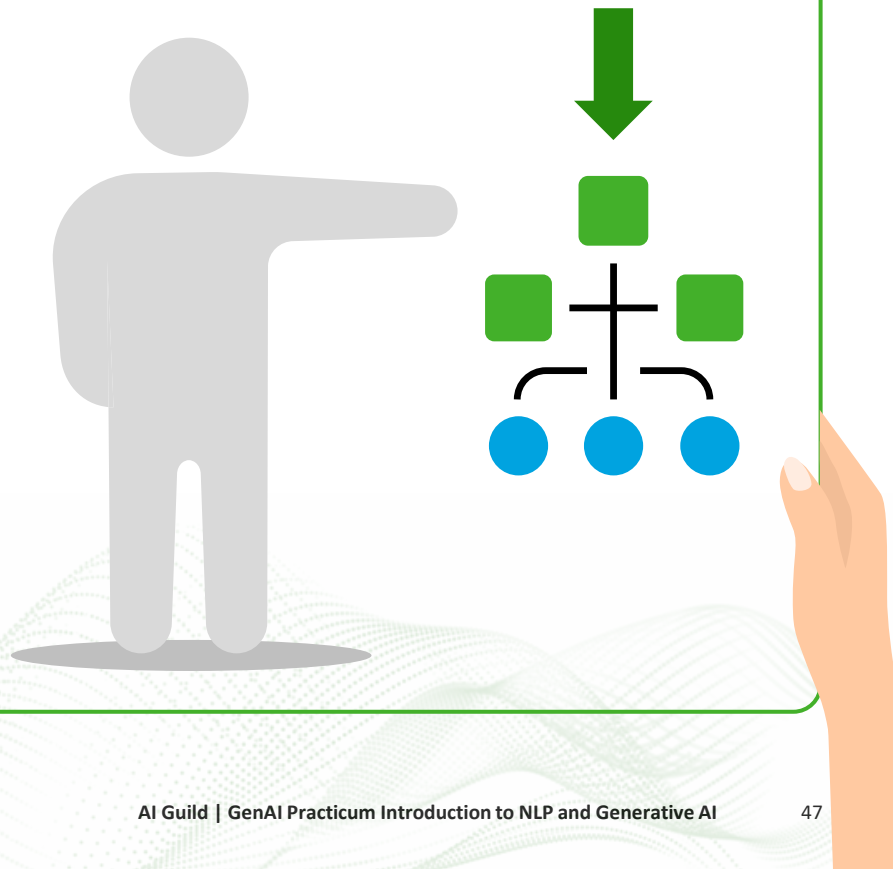
Lab 3
Run an LLM
(10 min)

Main libraries to use:

- Llama_cpp

Use LLAMA to spin up a your very own Foundation Model.

1. Download the Llama Foundation Model from HuggingFace (8GB download – 30 mins)
2. Import the Llama Foundation Model
3. Use the model on your local machine



Q&A

Session Takeaways

Share key Takeaways

01 NLP tools are lightweight and easy to use.

02 GANs can be useful for generating images.

03 LLMs are the foundation models for Generative AI.

04 GenAI is going to change a lot of what we do.

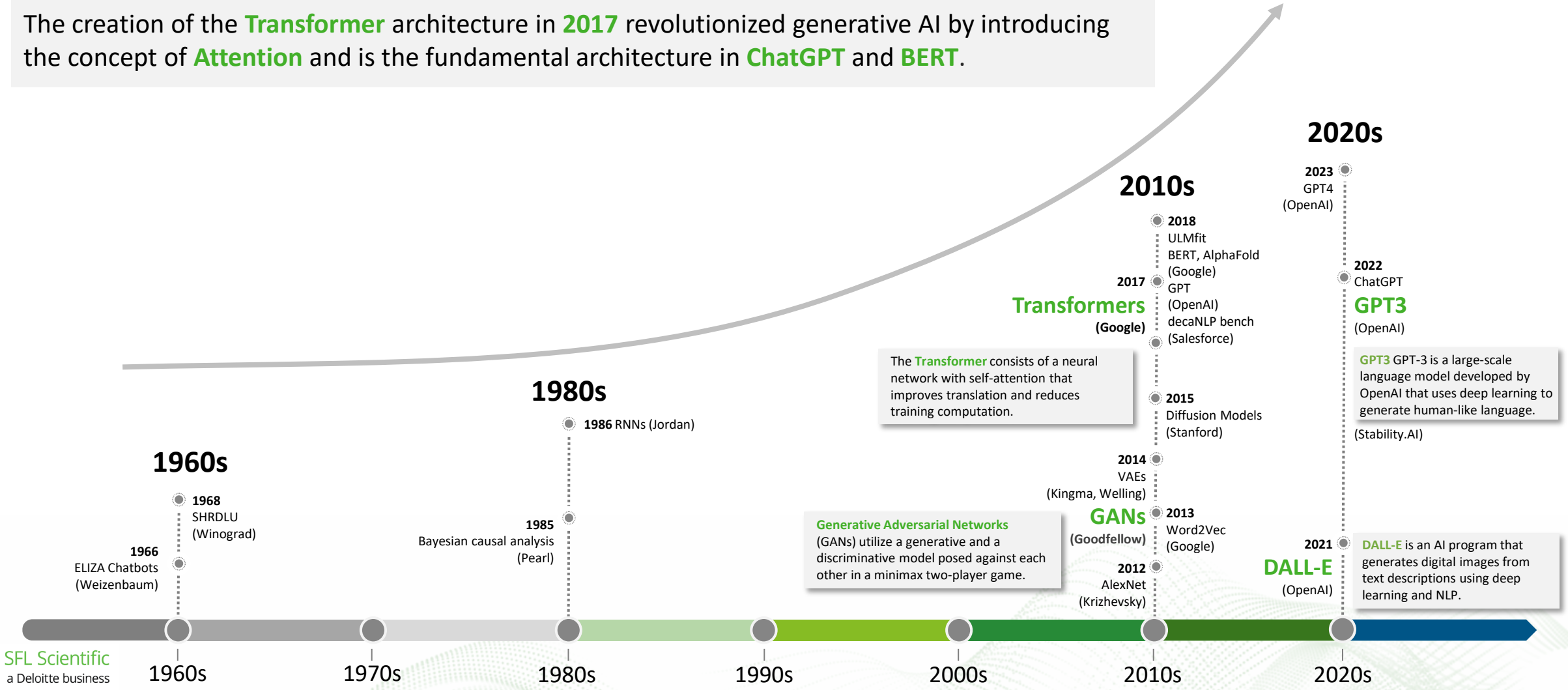
05 GenAI brings a whole host of ethical issues with it.

Appendix

History of Generative AI

Looking back at the evolution and milestones throughout history

The creation of the **Transformer** architecture in **2017** revolutionized generative AI by introducing the concept of **Attention** and is the fundamental architecture in **ChatGPT** and **BERT**.



Deloitte.