SPRING INTO AI

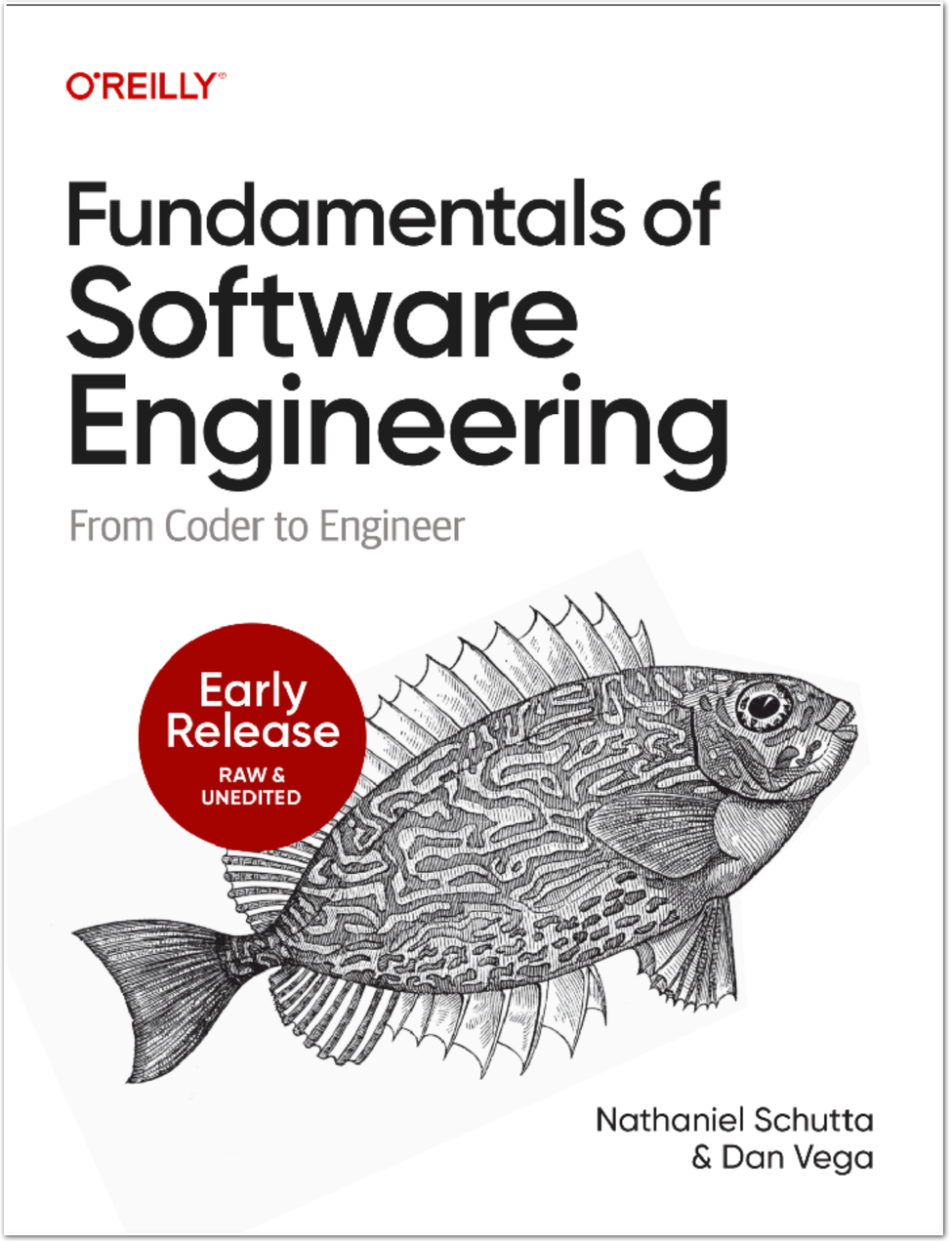
Building Intelligent Applications in Java with Spring AI

Dan Vega - Spring Developer Advocate @Broadcom

ABOUT ME

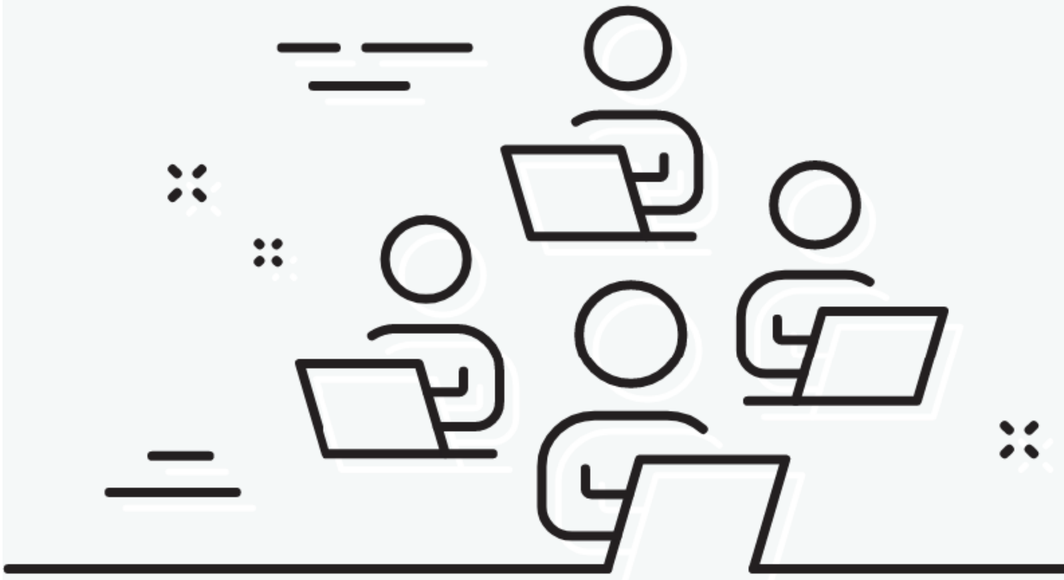
Learn more at danvega.dev

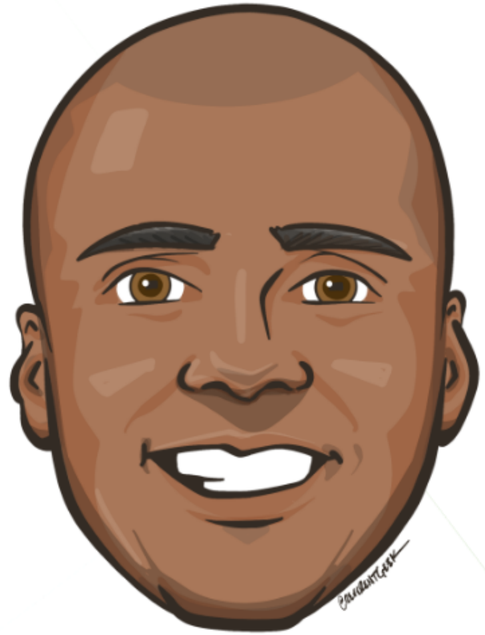
- 👨‍👩‍👧‍👦 Husband & Father
- 🏠 Cleveland
- ☕ Java Champion
- 👨‍💻 Software Development 23 Years
- 🍃 Spring Developer Advocate
- 📖 Author (Soon to be)



spring

OFFICE HOURS





https://www.springofficehours.io

ByteSized AI

www.bytesizedai.dev

Archive

Search posts...

Artificial Intelligence (AI) | Prompt Engineering | Open AI | Claude | Image Generation

Sep 18, 2024

How to talk to Robots

Learning how to effectively communicate with AI

Dan Vega

Sep 16, 2024 ♥ 2

The AI That Thinks Before It Speaks

Getting to know Open AI's two new models

Dan Vega

Sep 12, 2024

🤖 Why you need to check out Claude's Projects

Getting started with Projects in Claude

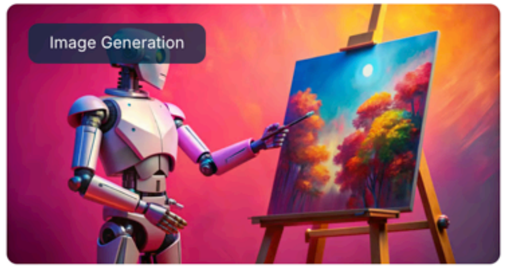
Dan Vega

Sep 06, 2024

What is Artificial Intelligence (AI)

How do you define AI?

Dan Vega



Sep 05, 2024

Generating images with AI

How I generated a logo for my newsletter

Dan Vega

Sep 03, 2024 ♥ 1 2

Welcome to ByteSized AI 🤖

👋🏻 Hello, World!

Dan Vega

AGENDA

What are we going to talk about?

- What is AI?
  - Machine Learning
  - Deep Learning
  - Large Language Models (LLMs)
  - Prompt Engineering
- Java & AI
- Spring AI
- Show me the code


WHAT IS ARTIFICIAL INTELLIGENCE

WHAT IS ARTIFICIAL INTELLIGENCE

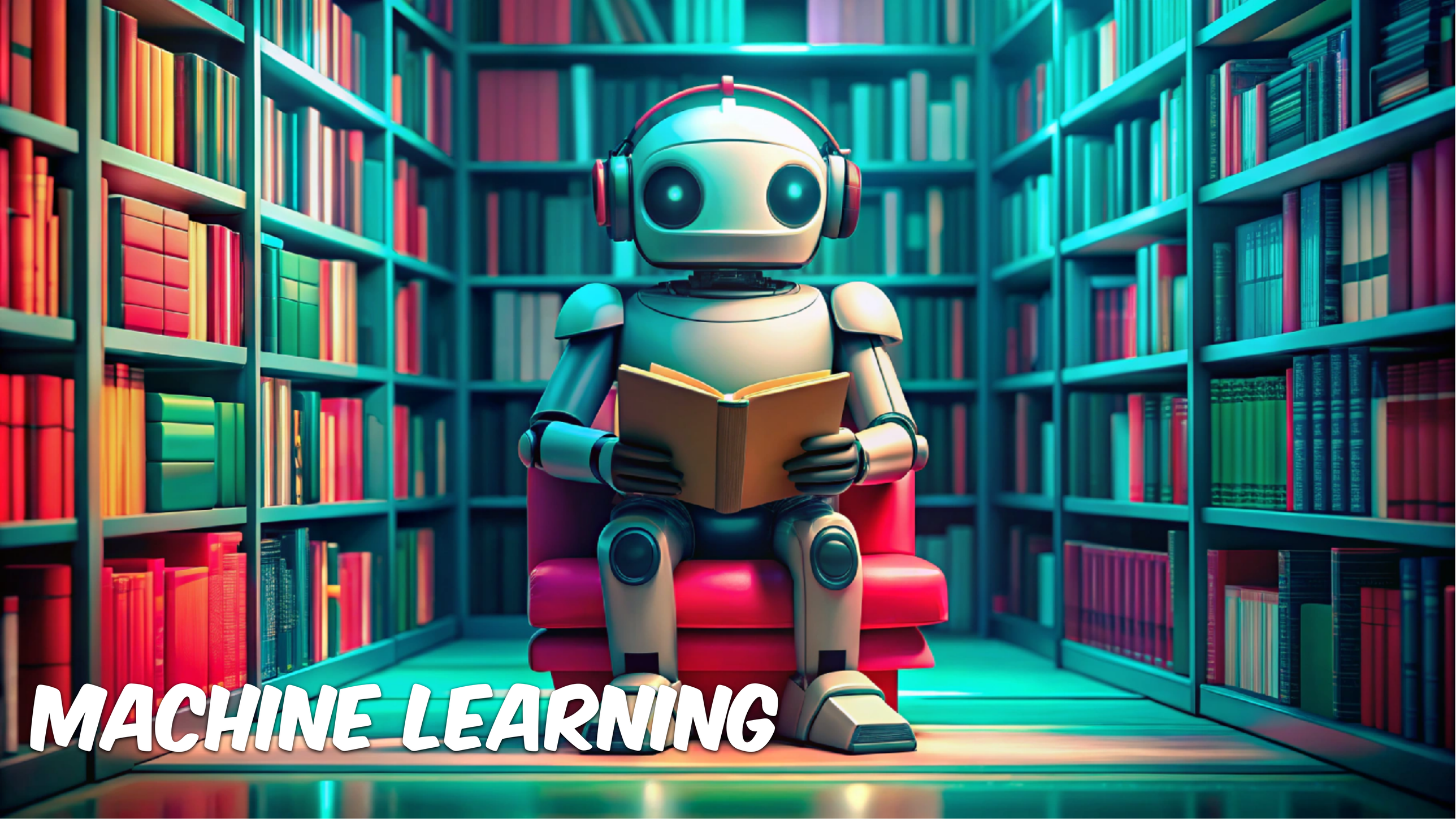
Artificial Intelligence

AI is technology that helps computers do things that typically require human intelligence:

- Understand language
- Recognize patterns
- Make decisions
- Learn from experience

Think of AI like tools in a toolbox:

- Some tools do one job really well (like image recognition)
- Some tools work together to solve more complex problems
- The tools keep getting better as we use them


MACHINE LEARNING

MACHINE LEARNING

Artificial Intelligence

Machine Learning

Unlike traditional programming, where explicit instructions are provided for every scenario, ML systems learn patterns from data, allowing them to make predictions or decisions without being explicitly programmed for each possibility.

MACHINE LEARNING

Use Cases

- Facial Recognition
- Recognize Tumors on x-ray scans
- Abnormality on ultrasounds
- Self-drive mode (recognize stop sign / pedestrian / etc...)
- Fraud Detection
- Product Recommendations (YouTube)
- Spam Filtering

SUPERVISED LEARNING

Labeling the training data

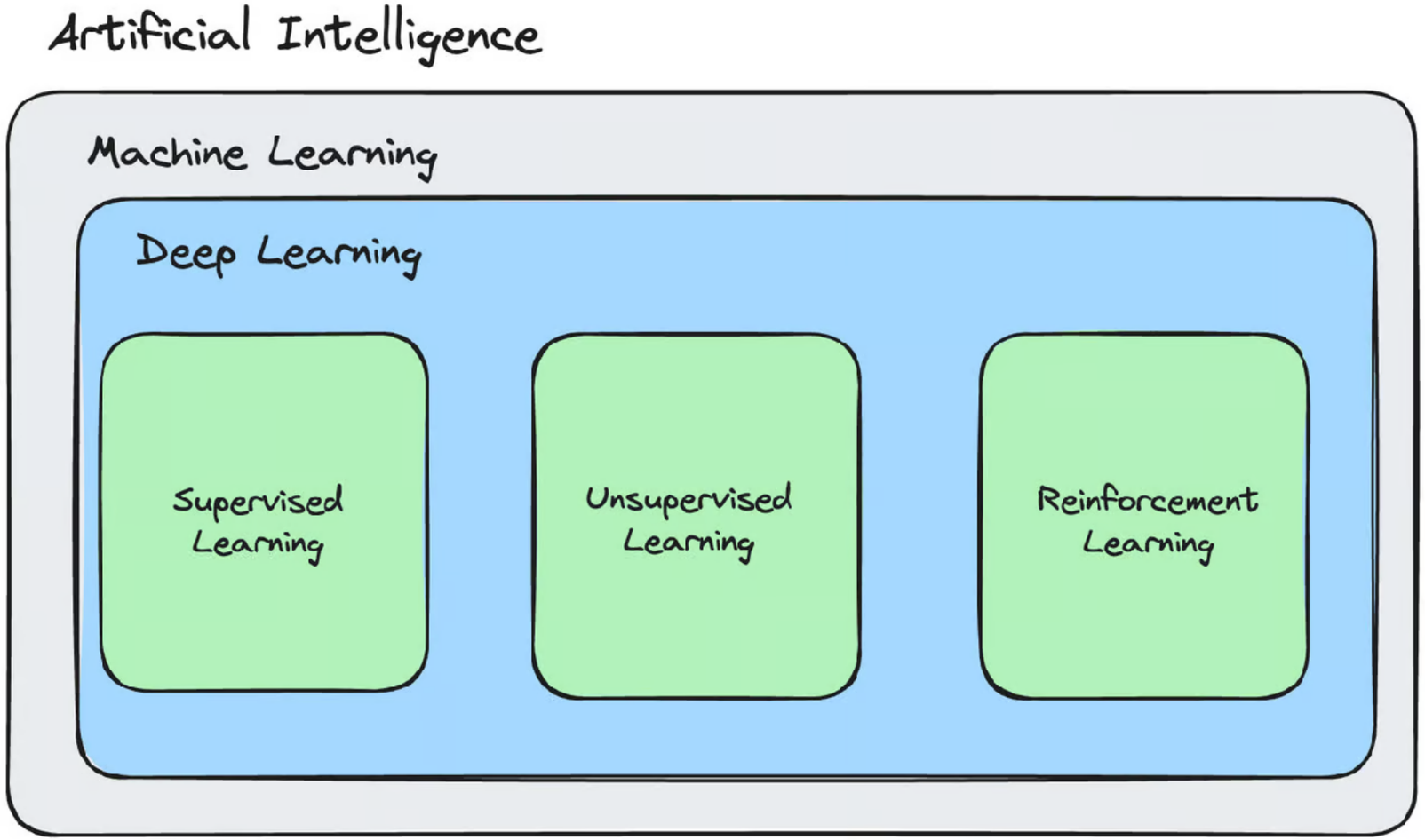
Dan Vega

Dan Vega

Dan Vega


DEEP LEARNING

DEEP LEARNING AND NEURAL NETWORKS



NEURAL NETWORKS

What makes deep learning powerful?

- The first layer might detect edges
- The next layer might recognize shapes
- Deeper layers could identify more complex features like eyes or wheels
- The final layer puts it all together to classify the entire image

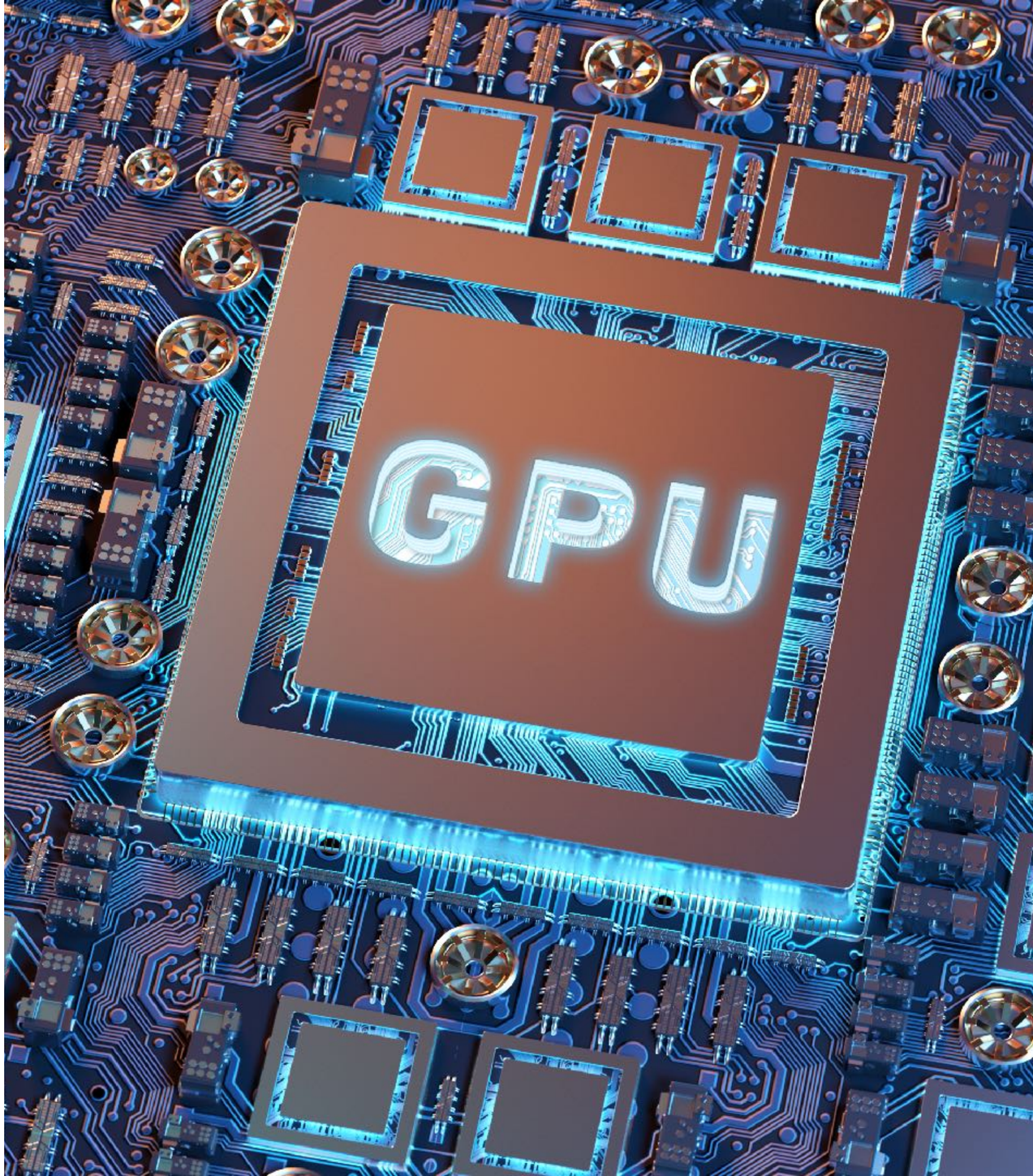
ARTIFICIAL NEURAL NETWORK

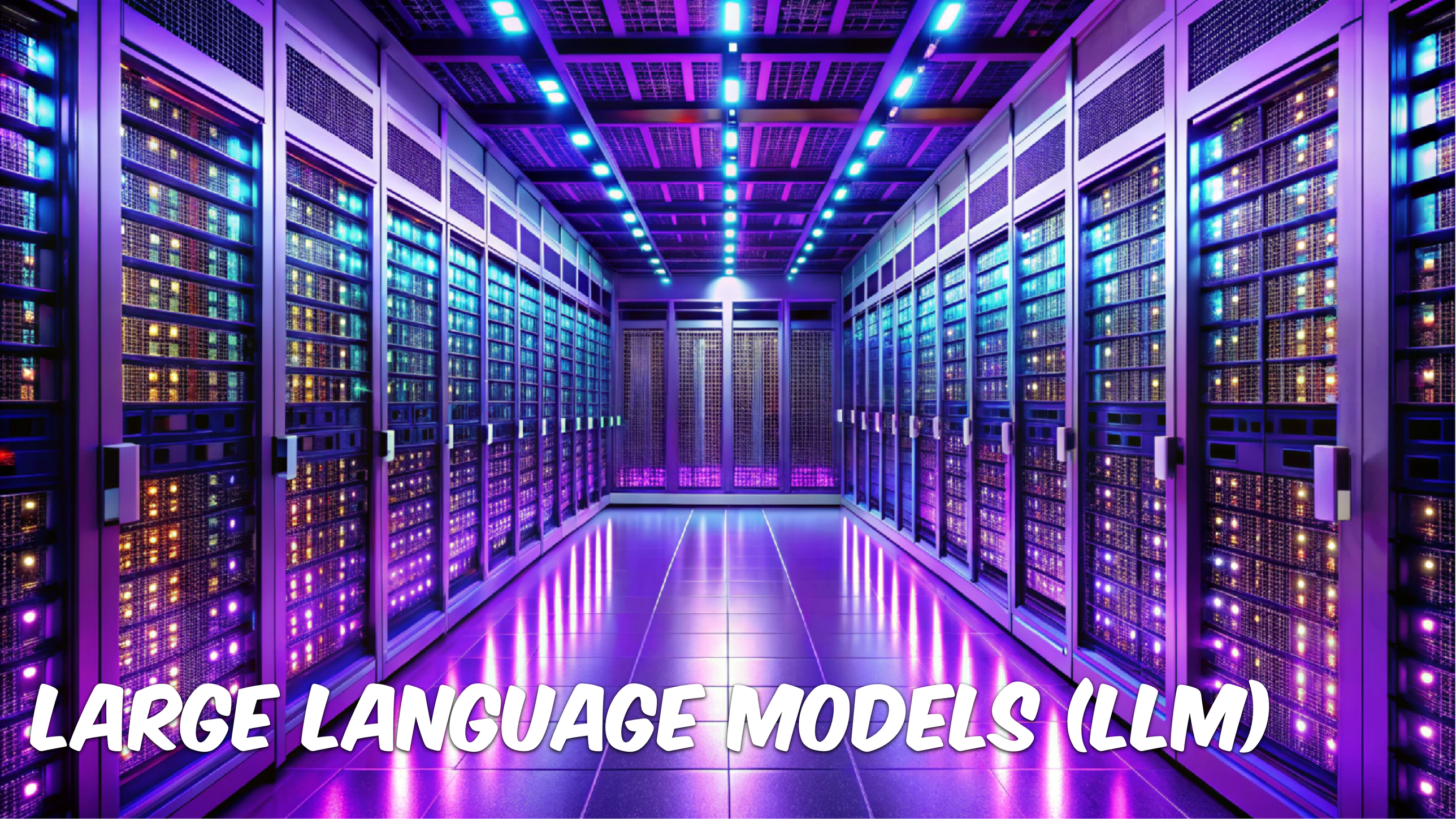
What made deep learning possible?

- Breakthrough algorithms: Neural network architectures that mimi how our brains learn
- Massive training data: Billions of examples teaching models to recognize patterns
- Computational power

THE GPU REVOLUTION

- Neural networks require billions of parallel calculations
- GPUs: The unexpected heroes of AI -
  - Designed for gaming, repurposed for AI
  - 100x faster than CPUs for neural network training
  - Enable in weeks what would take years on traditional hardware
- The hardware catalyst: GPU advancement directly accelerates AI capabilities
- No GPUs = No modern AI




LARGE LANGUAGE MODELS (LLM)

ATTENTION IS ALL YOU NEED

The Foundation: Attention Mechanisms

- In 2017 a new idea called the transformer changed everything.
- It helped AI focus on the most important parts of a sentence (like how “dog” and “barked” connect), making language understanding much better.
- This laid the groundwork for more powerful language models

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

WHAT ARE LARGE LANGUAGE MODELS?

Definition

- LLMs are AI models built to understand and create human-like language
- They're "large" because they have billions of adjustable settings (called parameters) that help them learn from data

What They Learn From

- LLMs are trained on massive collections of text—like books, articles, and websites.
- This helps them pick up grammar, facts, and even some common sense.

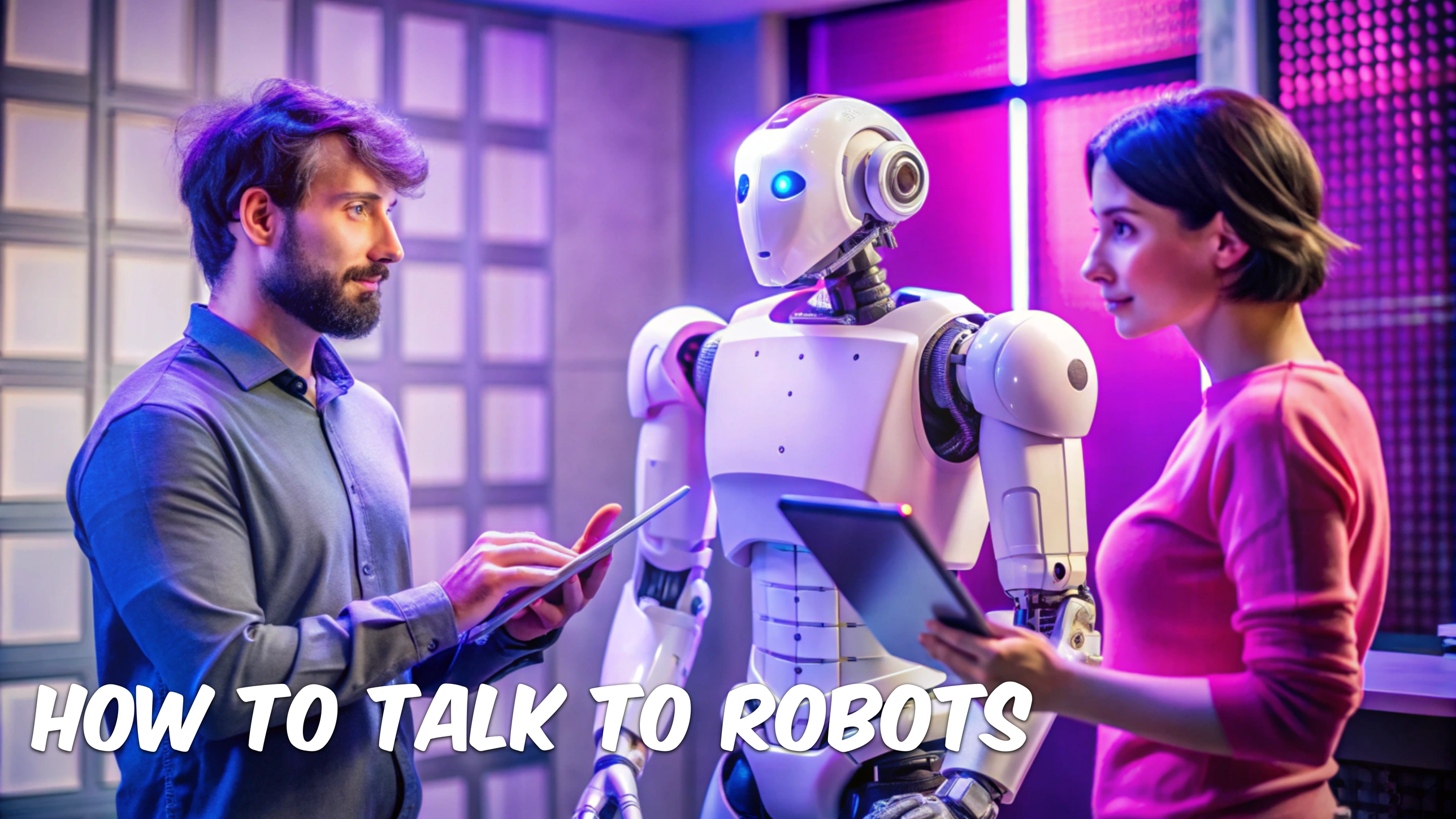
What They Can Do

- LLMs are incredibly versatile! They can:
  - Translate languages (e.g., English to Spanish)
  - Summarize long articles
  - Answer questions
  - Write stories, essays, or even code
  - Chat like a human (think virtual assistants)

GENERATIVE AI

Generative Pre-trained Transformer (GPT)

- **Now that we understand:**
  - Transformers provide the architecture (how they process language)
  - Pre-training on vast amounts of data teaches language understanding
  - Generative capability allows them to create new content
- **Key Capabilities:**
  - Text generation (writing, translation, summarization)
  - Code generation and analysis
  - Complex reasoning and problem-solving


HOW TO TALK TO ROBOTS

PROMPT ENGINEERING

Learn how to effectively communicate with AI

- Clear communication is key - just like with humans
- Structure determines success - giving context, examples and specific instructions
- Think of it as teaching, not commanding
- Bad Prompt: “Write a blog post about AI”
- Good Prompt: “Write a technical blog post explaining neural networks to junior developers, focusing on practical examples. Include code samples in Python and keep it under 1,000 words.”
- Learn More: https://www.bytesizedai.dev/p/how-to-talk-to-robots

PROMPT ENGINEERING

Practical Prompt Techniques that work

- Be Specific: "Write a 500-word blog post about sustainable gardening for beginners" beats "Write about gardening"
- Use Examples: "I want an email that sounds professional but friendly, like: 'Dear Team, I hope this message finds you well...'"
- Give Context: "As the marketing manager for a small local business, I need to..."
- Request Formats: "Please format your response as a bulleted list" or "Use markdown headers"
- Iterate & Refine: "That's good, but can you make it more conversational and add a section about..."
- Save What Works: Keep a collection of your most effective prompts to reuse and adapt


FUN COUPONS!

GPT-4o

GPT-4o is our most advanced multimodal model that's faster and cheaper than GPT-4 Turbo with stronger vision capabilities. The model has 128K context and an October 2023 knowledge cutoff.

Learn about GPT-4o ↗

Model	Pricing	Pricing with Batch API*
gpt-4o	$5.00 / 1M input tokens	$2.50 / 1M input tokens
	$15.00 / 1M output tokens	$7.50 / 1M output tokens
gpt-4o-2024-08-06	$2.50 / 1M input tokens	$1.25 / 1M input tokens
	$10.00 / 1M output tokens	$5.00 / 1M output tokens
gpt-4o-2024-05-13	$5.00 / 1M input tokens	$2.50 / 1M input tokens
	$15.00 / 1M output tokens	$7.50 / 1M output tokens

Vision pricing calculator

Set model

gpt-4o-2024-08-06

Set width: 150 px by Set height: 150 px = $0.000638

Low resolution

*Batch API pricing requires requests to be submitted as a batch. Responses will be returned within 24 hours for a 50% discount. Learn more about Batch API ↗

Model	Pricing	Pricing with Batch API*
gpt-4o	$5.00 / 1M input tokens	$2.50 / 1M input tokens
	$15.00 / 1M output tokens	$7.50 / 1M output tokens
gpt-4o-2024-08-06	$2.50 / 1M input tokens	$1.25 / 1M input tokens
	$10.00 / 1M output tokens	$5.00 / 1M output tokens

https://platform.openai.com/tokenizer

Tokenizer

Learn about language model tokenization

OpenAI's large language models (sometimes referred to as GPT's) process text using **tokens**, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens, and excel at producing the next token in a sequence of tokens.

You can use the tool below to understand how a piece of text might be tokenized by a language model, and the total count of tokens in that piece of text.

It's important to note that the exact tokenization process varies between models. Newer models like GPT-3.5 and GPT-4 use a different tokenizer than previous models, and will produce different tokens for the same input text.

GPT-3.5 & GPT-4 | GPT-3 (Legacy)

```
Hello, My name is Dan Vega, Java Champion, Spring Developer Advocate,
Husband and #GirlDad based outside of Cleveland OH. I created this
website as a place to document my journey as I learn new things and share
them with you. I have a real passion for teaching and I hope that one of
blog posts, videos or courses helps you solve a problem or learn
something new.
```

Clear | Show example

Tokens 78 **Characters** 363

Hello, My name is Dan Vega, Java Champion, Spring Developer Advocate, Husband and #GirlDad based outside of Cleveland OH. I created this website as a place to document my journey as I learn new things and share them with you. I have a real passion for teaching and I hope that one of blog posts, videos or courses helps you solve a problem or learn something new.

Text | Token IDs

A helpful rule of thumb is that one token generally corresponds to ~4 characters of text for common English text. This translates to roughly ¾ of a word (so 100 tokens ~= 75 words).


JAVA & AI

JAVA & AI

Leveraging Artificial Intelligence in Java Applications

The Java & AI Opportunity

- Java powers 90% of Fortune 500 companies' backend systems
- AI capabilities are now expected in enterprise applications
- Java developers have a unique market advantage with AI skills

Enterprise Use Cases

- Document processing & knowledge extraction
- Intelligent customer service automation
- Code generation & Developer productivity tools
- Predictive analytics with enterprise data

```
#!/bin/bash
echo "Calling Open AI..."
MY_OPENAI_KEY=“YOUR_API_KEY_HERE”
PROMPT=“Tell me an interesting fact about Java"

curl https://api.openai.com/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $MY_OPENAI_KEY" \
  -d '{"model": "gpt-4o", "messages": [{"role":"user", "content": "'"${PROMPT}"'"}] }'
```

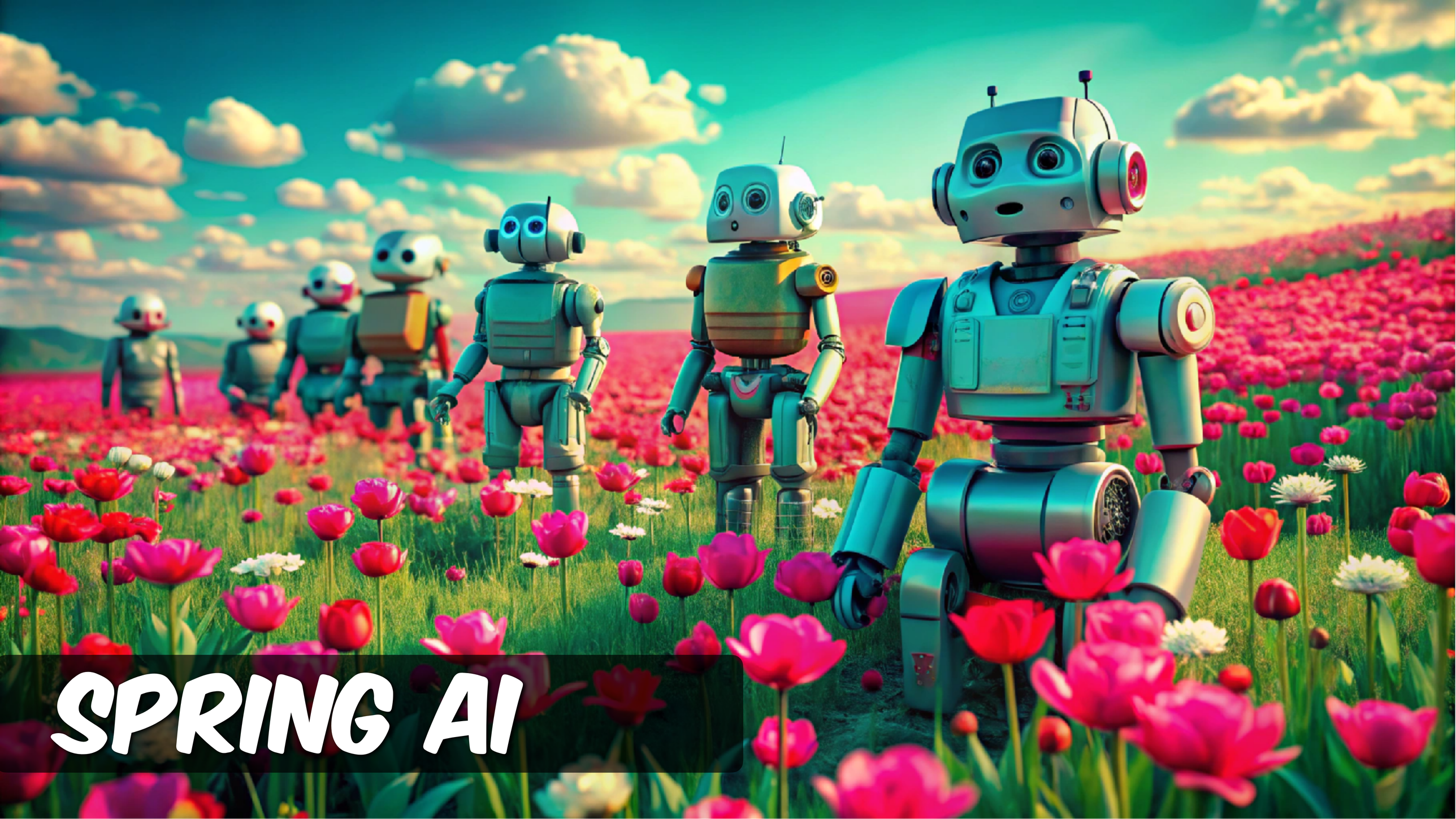
```
  "id": "chatcmpl-ABNbjZ5oRbo72OevnCX2arPufJCYK",
  "object": "chat.completion",
  "created": 1727275719,
  "model": "gpt-4o-2024-05-13",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Sure! Did you know that Java was initially designed with interactive television in mind? James Gosling and his team at Sun Microsystems started the project in 1991 under the name \"Oak.\" The name was later changed to \"Java\" after discovering there was already a programming language called Oak. Java's versatility has made it one of the most popular programming languages for a wide range of applications, far beyond its initial intended use for TV set-top boxes!",
        "refusal": null
      },
      "logprobs": null,
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 14,
    "completion_tokens": 90,
    "total_tokens": 104,
    "completion_tokens_details": {
      "reasoning_tokens": 0
    }
  },
```

```java
public static void main(String[] args) throws IOException, InterruptedException {
    var apiKey = "YOUR_API_KEY_HERE";
    var body = """
            {
                "model": "gpt-4o",
                "messages": [
                    {
                        "role": "user",
                        "content": "Tell me an interesting fact about Java"
                    }
                ]
            }""";

    HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://api.openai.com/v1/chat/completions"))
            .header("Content-Type", "application/json")
            .header("Authorization", "Bearer " + apiKey)
            .POST(HttpRequest.BodyPublishers.ofString(body))
            .build();

    var client = HttpClient.newHttpClient();
    var response = client.send(request, HttpResponse.BodyHandlers.ofString());
    System.out.println(response.body());
}
```

SPRING AI PROVIDES US SO MUCH MORE THAN A FACILITY FOR MAKING REST API CALLS


SPRING AI

SPRING AI

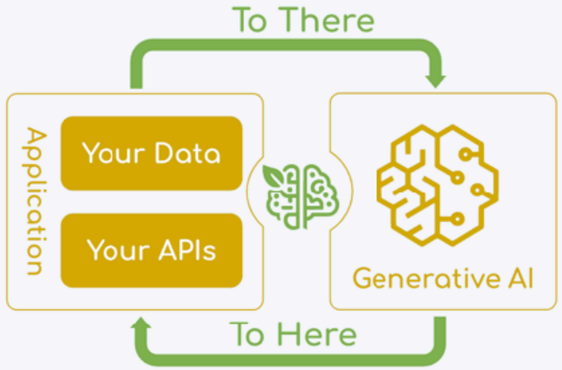
AI for Spring Developers

- https://spring.io/projects/spring-ai
  - Dr. Mark Pollack
  - Current Version 1.0.0-M6
- Portable API support across AI providers for Chat, Image & Audio
- Synchronous & Streaming API options
- Inspired by Python Projects
  - LangChain
  - LlamaIndex

Spring AI

OVERVIEW LEARN

Spring AI is an application framework for AI engineering. Its goal is to apply to the AI domain Spring ecosystem design principles such as portability and modular design and promote using POJOs as the building blocks of an application to the AI domain.



At its core, Spring AI addresses the fundamental challenge of AI integration: Connecting your enterprise **Data** and **APIs** with the **AI Models**.

Features

Spring AI provides the following features:

- Support for all major AI Model providers such as Anthropic, OpenAI, Microsoft, Amazon, Google, and Ollama. Supported model types include:
  - Chat Completion
  - Embedding
  - Text to Image
  - Audio Transcription
  - Text to Speech

CHECK OUT MY DEMO

THANK YOU

dan.vega@broadcom.com

@therealdanvega

https://www.danvega.dev

https://www.bytesizedai.dev