ETH zürich

IVIA lab

Intro to AI

From Machine Learning to Transformers

Matthias Miller

Mennatallah El-Assady

emba X5 - Essentials III: Cognitive Technologies

July 8, 2026

Outline

- **AI, Machine Learning, and Deep Learning**
- **Milestones of AI History**
- **Machine Learning as Software 2.0**
- **Encoding Data as Vectors**
- **Learning, Generalization, and Neural Networks**
- **Attention, Transformers, and Generative LLMs**

In one word — what comes to mind when you hear “AI”?

Part 1 — Foundations

How Machines Learn from Data

Milestones of AI History

1950s	1960 - 1990	2000s	2010s	2020s
A. Turing: "Can a machine think?"	Knowledge Engineering	End of the AI Winter	Breakthrough in Deep Learning → AI Spring	ChatGPT
McCarthy: "Artificial Intelligence"	Limitations arise → AI Winter		Transformer	Multimodality
				Agentic workflows

🎯 Goal

↓

🧠 Think → 🛠️ Act → 👁️ Observe

↻ repeat

Agentic workflows, 2024

B. Valentina et el., "Understanding basic principles of Artificial Intelligence: a practical guide for intensivists"

Machine Learning = "Software 2.0"

Learning behavior from data instead of writing rules by hand

Classical software (1.0)

A human writes the rules:

```
IF email contains
   "Please login here"
THEN  spam
ELSE  not spam
```

Breaks on every case you didn't foresee.

Machine learning (2.0)

We show **examples**, the model finds the rules:

📧 "Please login here..." → **spam**

📧 "Lunch tomorrow?" → **not spam**

📧 "You won a prize!!!" → **spam**

... thousands more ...

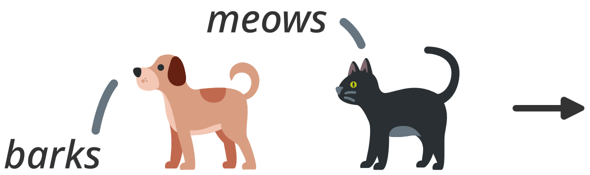
Generalizes to cases nobody wrote down.

The central question becomes: **how do we get the data into the model?**

How Do We Get Data *Into* a Model?

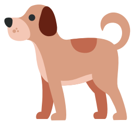
Everything a model sees must first become **numbers (a vector)**

Say we want to tell **cats** and **dogs** apart from two features: *does it bark?* and *does it meow?*

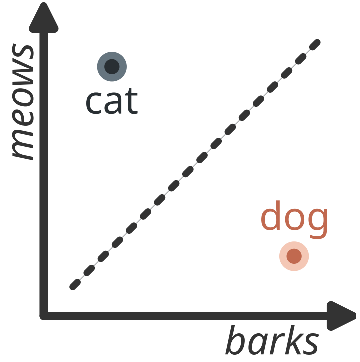


	barks	*meows*
cat	**0**	**1**
dog	**1**	**0**

$$= \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}$$



$$= \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}$$



Encoding ① — One-Hot Vectors

Words become **switches**: one position "on", all others "off"

cat = [1, 0, 0, 0]

dog = [0, 1, 0, 0]

king = [0, 0, 1, 0]

queen = [0, 0, 0, 1]

- ✅ Simple, works for small vocabularies
- ❌ **Huge & sparse**: one dimension *per word* — 50,000+ for real language
- ❌ **No notion of similarity**: `king` is exactly as far from `queen` as from `cat` or `dog`.

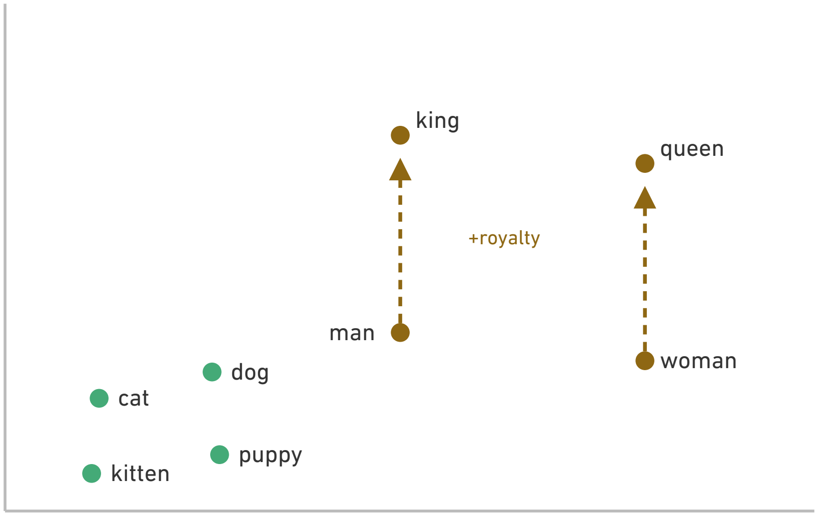
We threw away all **meaning**. Can the model *learn* a better encoding?

Encoding ② — Semantic Embeddings

Learn **dense vectors** where *similar things sit close together*

- Instead of switches, learn a handful of **continuous coordinates** per word
- Similar meanings → **nearby points**
- Meaning becomes **geometry**: directions in the space capture concepts

king − man + woman ≈ queen



T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space" (word2vec)

Learning in That Space

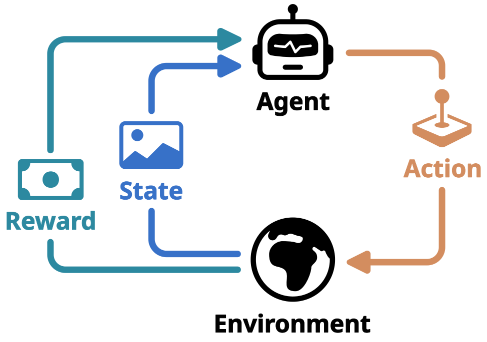
Three ways to learn — same geometry, different goal

- **Supervised** — learn a **decision boundary** from *labelled* points

  cat vs. dog, spam vs. not-spam
- **Unsupervised** — find **structure** in *unlabelled* points

  cluster similar items; reduce dimensions
- **Reinforcement** — learn by **trial, reward & error**

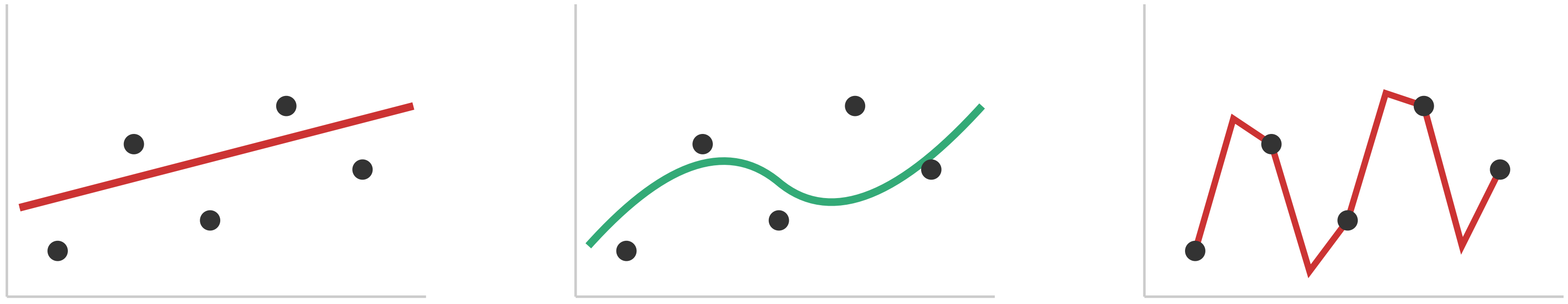
  robots, game-playing, recommendation



The right choice depends on your **data**, **task**, and **what feedback is available**.

The Core Challenge: **Generalization**

A model is only useful if it works on data it has **never seen**



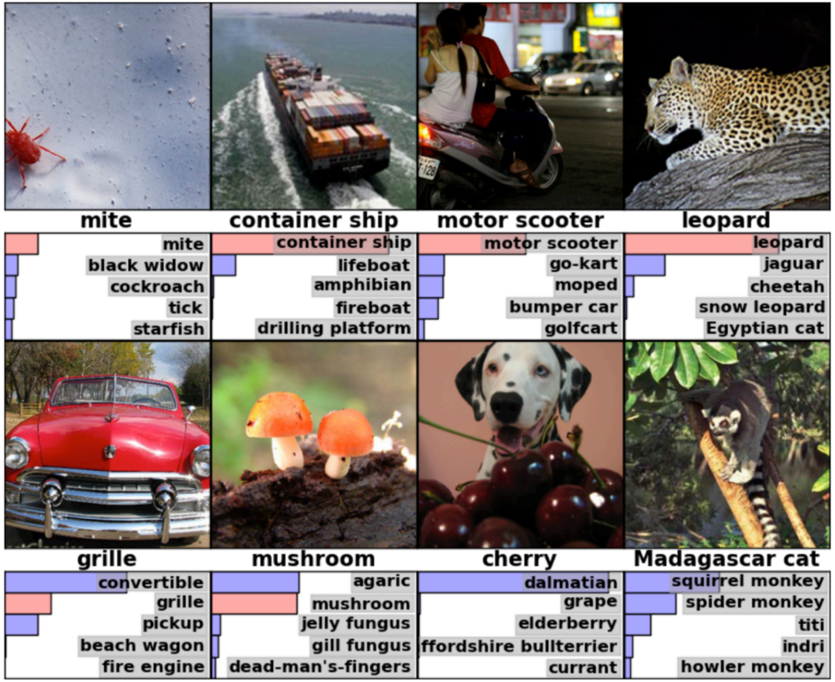
Underfitting
too simple — misses the pattern

Good fit
captures the trend, ignores noise

Overfitting
memorizes the noise, fails on new data

That's why we always keep a separate **test set** — and why "99% accuracy" can still mean a useless model.

The Deep-Learning Breakthrough — the "ImageNet Moment"

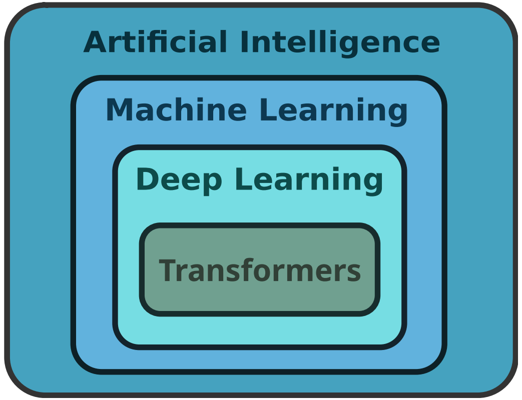


- Before 2012: ~**25%** error (top-5)
- 2012, **AlexNet**: ~**15%**
- 2022: ~**3%** — on par with human annotators
- **Enablers: big data + GPUs + deep nets**

A. Krizhevsky et al., "ImageNet Classification with Deep Convolutional Neural Networks", NeurIPS 2012

Deep Learning – The Breakthrough of AI

- With **Deep Learning**, AI made a jump in performance
  - Traditional DL architectures: feed-forward, CNNs, RNNs
    - Highly specialized for specific tasks
  - **Transformers**: attention-based models
    - Generalize across data types (text, images, music, ...)
    - **"Zero-shot" learning**

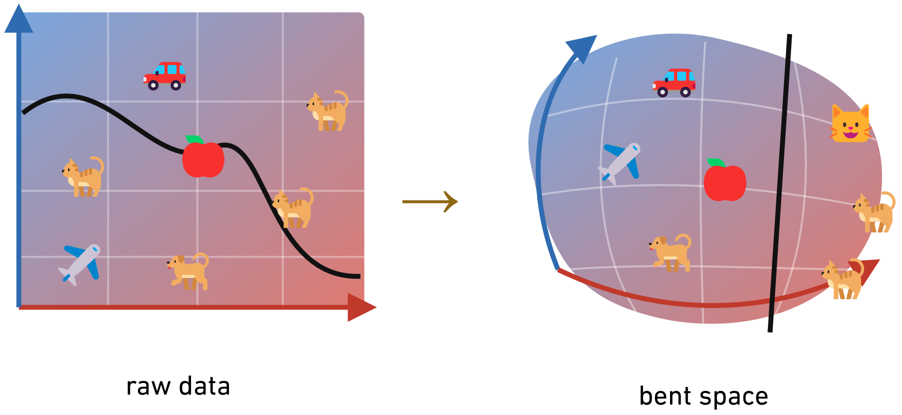


Neural Networks **Bend Space**

Why depth helps — the geometry of classification

- In the raw data, telling cats from non-cats needs a **complicated, wiggly boundary**
- Instead, the network **bends and folds the whole space**
- In the bent space, a single **straight cut** does the job

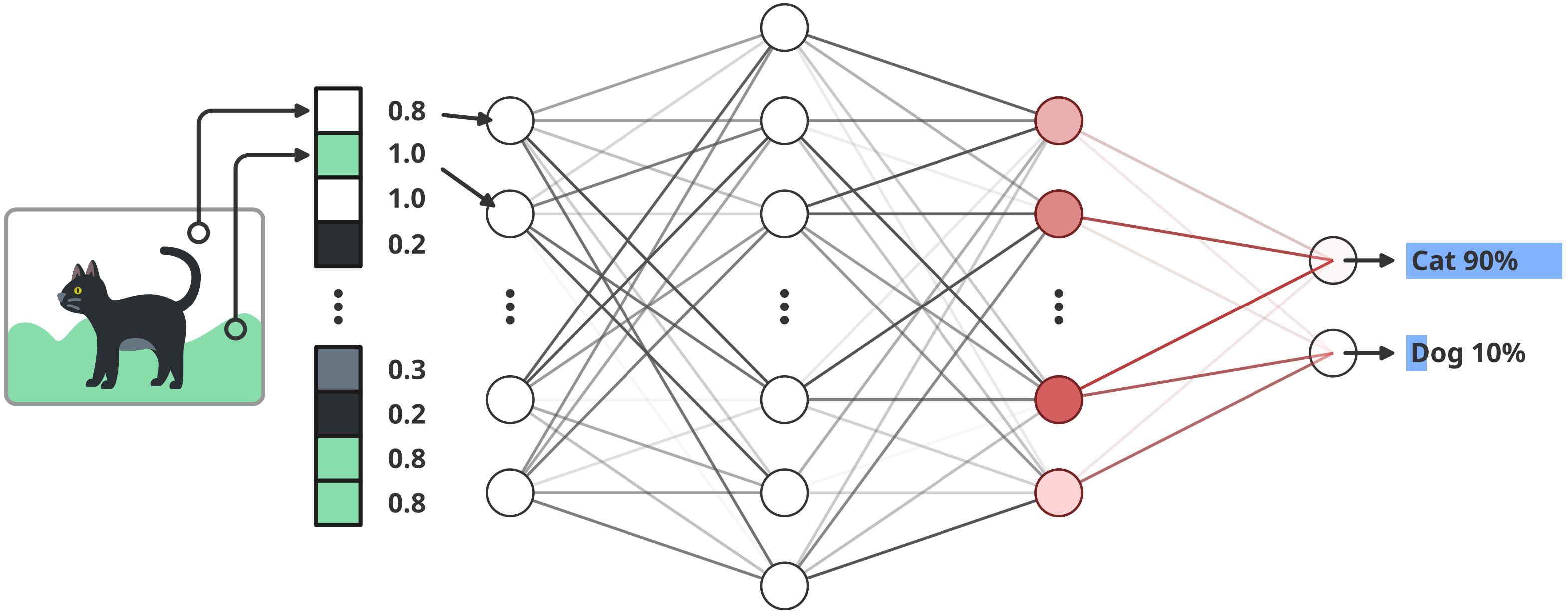
> "Bending" the space is easier than identifying a more complex decision boundary in a non-distorted space



Concept & metaphors: Luis Serrano, "How neural networks bend space" (Serrano.Academy)

How does a Neural Network **function** (i.e., *inference*)?

Supervised Learning

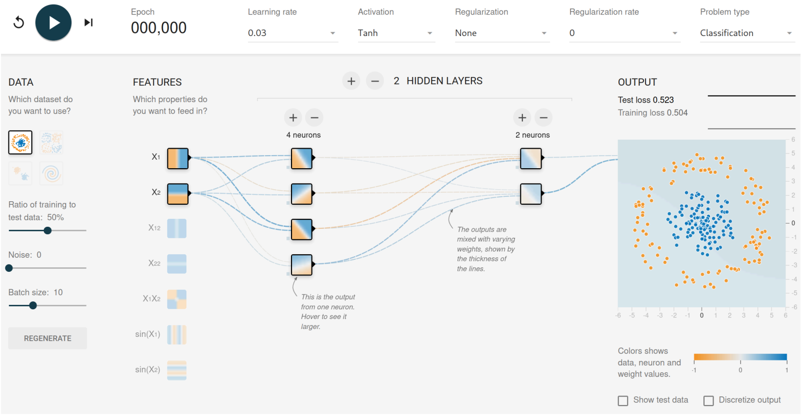


Hidden layers: **bend & fold the space**

Final layer: **the straight cut**

TensorFlow Playground

Interactive Demo

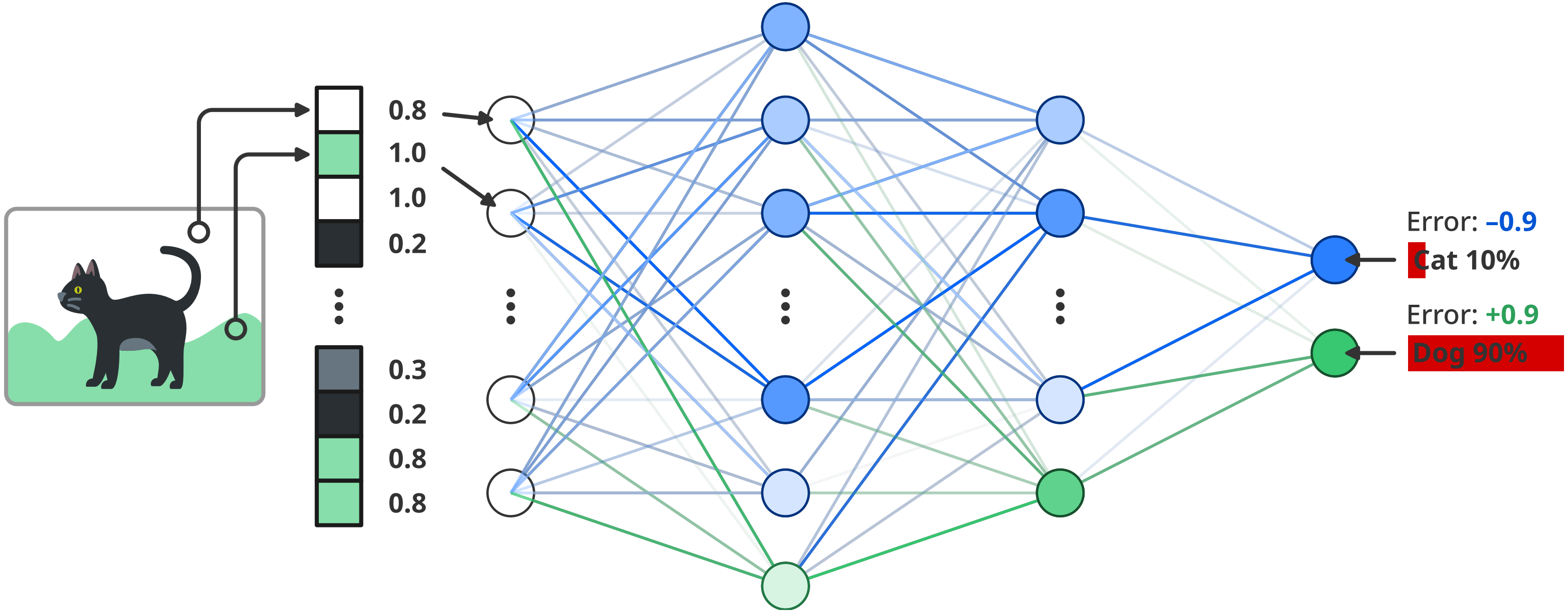


https://playground.tensorflow.org/

Blog: Understanding neural networks with TensorFlow Playground

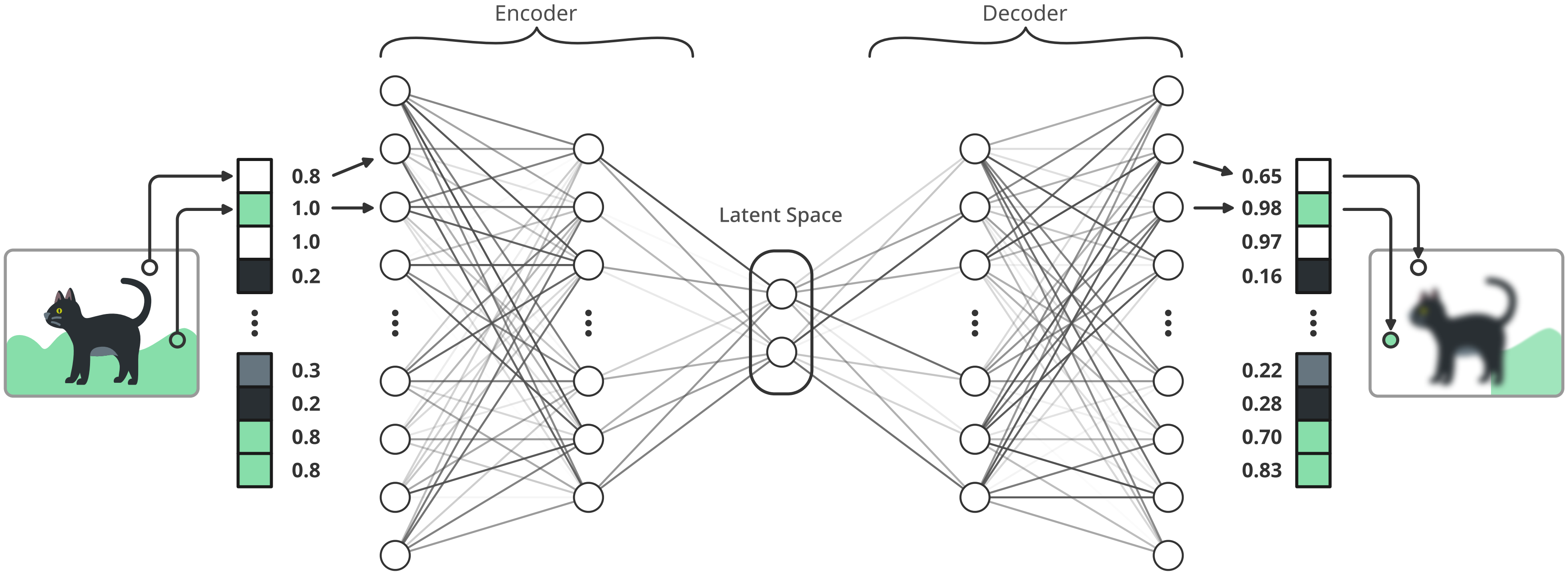
How does a Neural Network **learn** (i.e., *training*)?

Supervised Learning



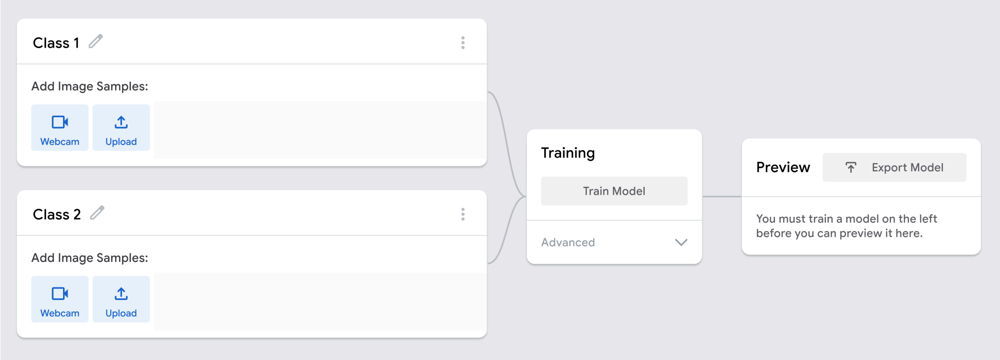
Auto-Encoder

Unsupervised Learning



Google TeachableMachine

Interactive Session



https://teachablemachine.withgoogle.com/train/image

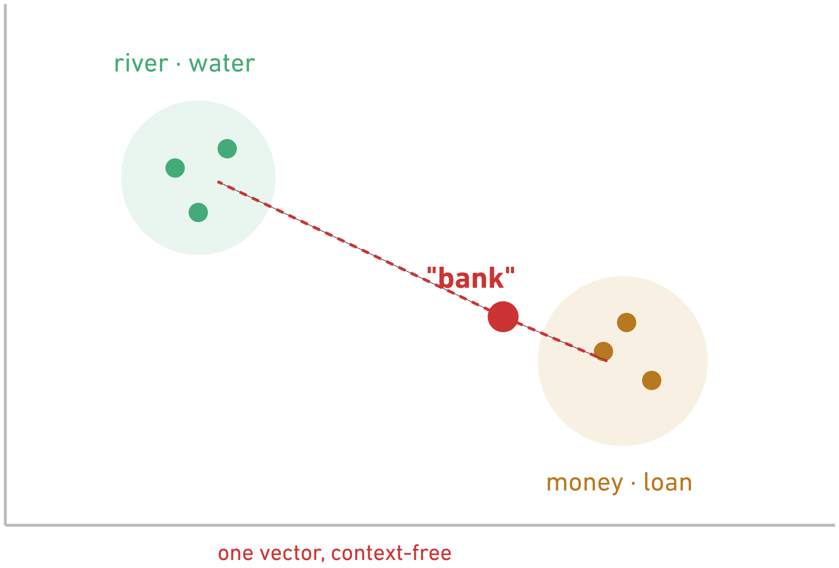
Part 2 — Transformers & Modern LLMs

The Architecture Behind ChatGPT

Encoding ③ — The Disambiguation Problem

One static vector per word **can't** be enough

- A word can mean different things:
  - "river **bank**" 🏞️
  - "savings **bank**" 🏦
- One fixed embedding blends the senses into a *single point* — the same vector no matter which meaning the sentence intends
- **We need the vector to depend on the surrounding words**



Encoding ④ — Context-Dependent Embeddings = **Attention**

The model looks at the **neighbouring words** to decide what each word means

"He sat on the **river bank** and watched the current."

→ **bank** attends to *river*, *current* — lands in the **nature** region

"She deposited cash at the **bank** downtown."

→ **bank** attends to *cash*, *deposited* — lands in the **finance** region

Attention = each word gathers meaning from the words it "pays attention to".
This single mechanism is the heart of the **Transformer**.

Inside Attention — "Who Listens to Whom"

	Your	journey	starts	with	one	step
Your	1.0					
journey	.55	.44				
starts	.38	.30	.31			
with	.27	.24	.24	.23		
one	.21	.19	.19	.18	.19	
step	.19	.16	.16	.15	.16	.16

Each row = how much one word "listens to" every other (weights sum to 1).

Greyed = masked: a generative LLM only looks **backward**, never ahead.

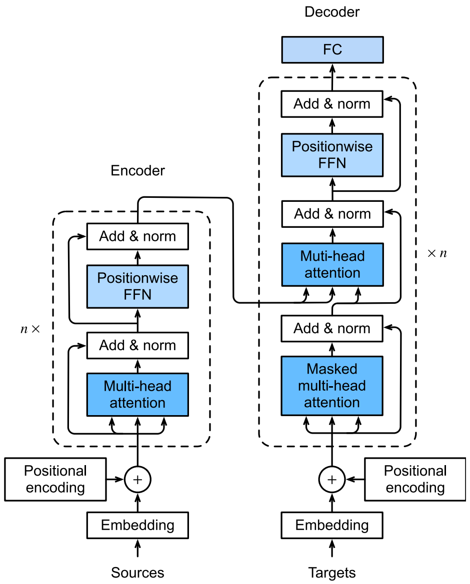
Weights aren't hand-set — they **emerge** from learned Q / K / V projections.

S. Raschka, "A Visual Guide to Attention Variants in Modern LLMs"

The Transformer Architecture

Foundation of All Modern Generative Models

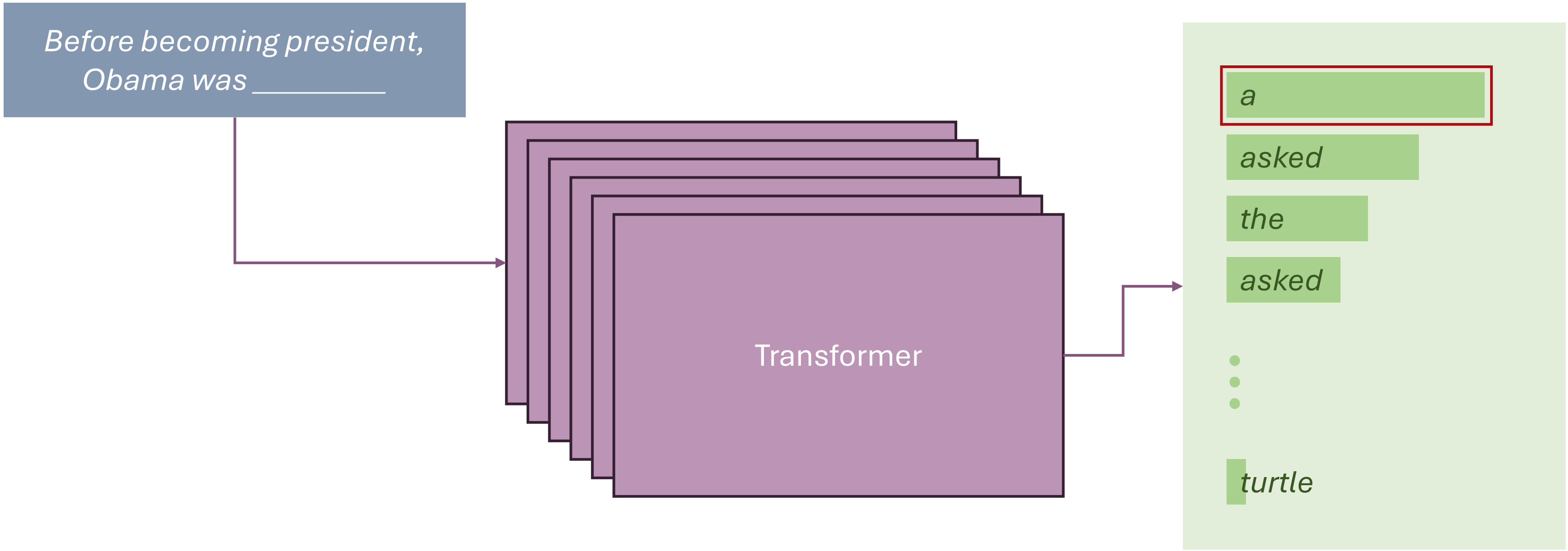
- Sequence-to-Sequence
  - Typical scenario: Translation
  - *Encoder-Decoder*
- Masked Language Modeling
  - Typical scenarios: Text classification, Named Entity Recognition
  - *Encoder-only*
- **Causal Language Modeling**
  - Typical scenario: Text generation
  - *Decoder-only*



A. Zhang et al., "Dive into Deep Learning"

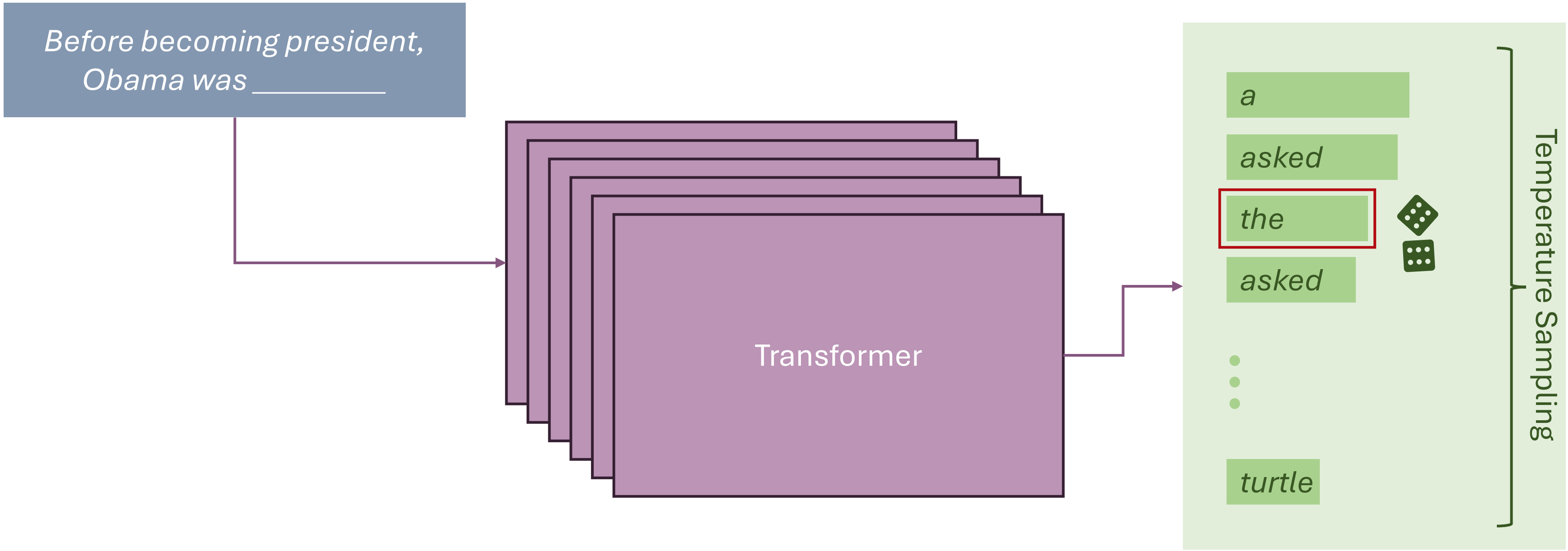
Output Probability Distribution

Generative LLMs



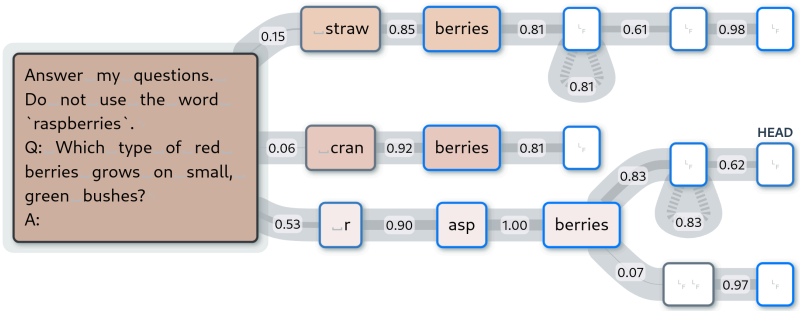
Temperature Sampling

Generative LLMs



generAItor Demo

Interactive Session



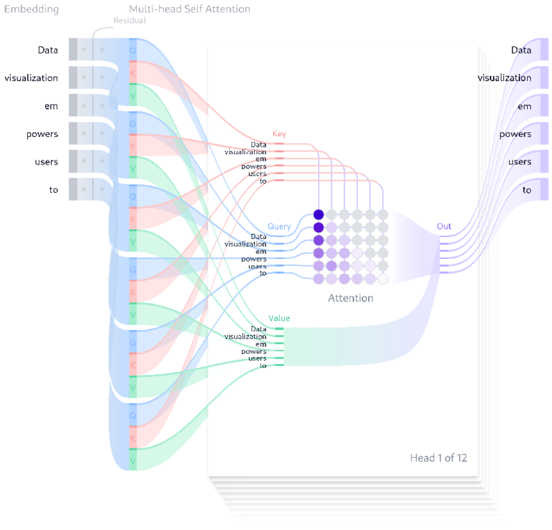
https://generaitor.ivia.ch/

Spinner et al., "generAItor: Tree-in-the-loop Text Generation for Language Model Explainability and Adaptation"

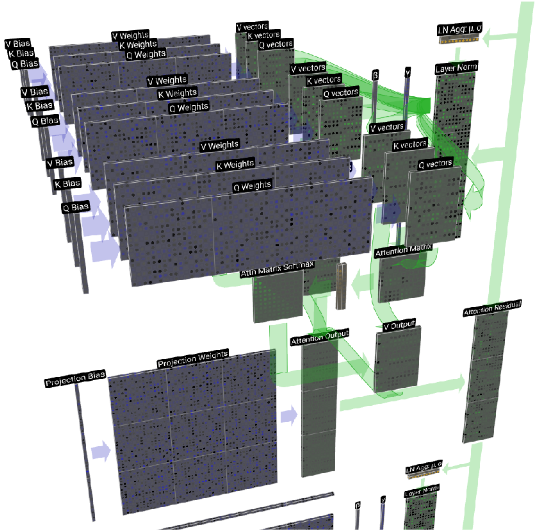
Spinner et al., "Revealing the Unwritten: Visual Investigation of Beam Search Trees to Address Language Model Prompting Challenges"

Transformers

Interactive Deep-Dive 💡



Transformer Explainer



LLM Visualization

A. Cho et al., "Transformer Explainer: Interactive Learning of Text-Generative Models"   B. Bycroft, "LLM Visualization"

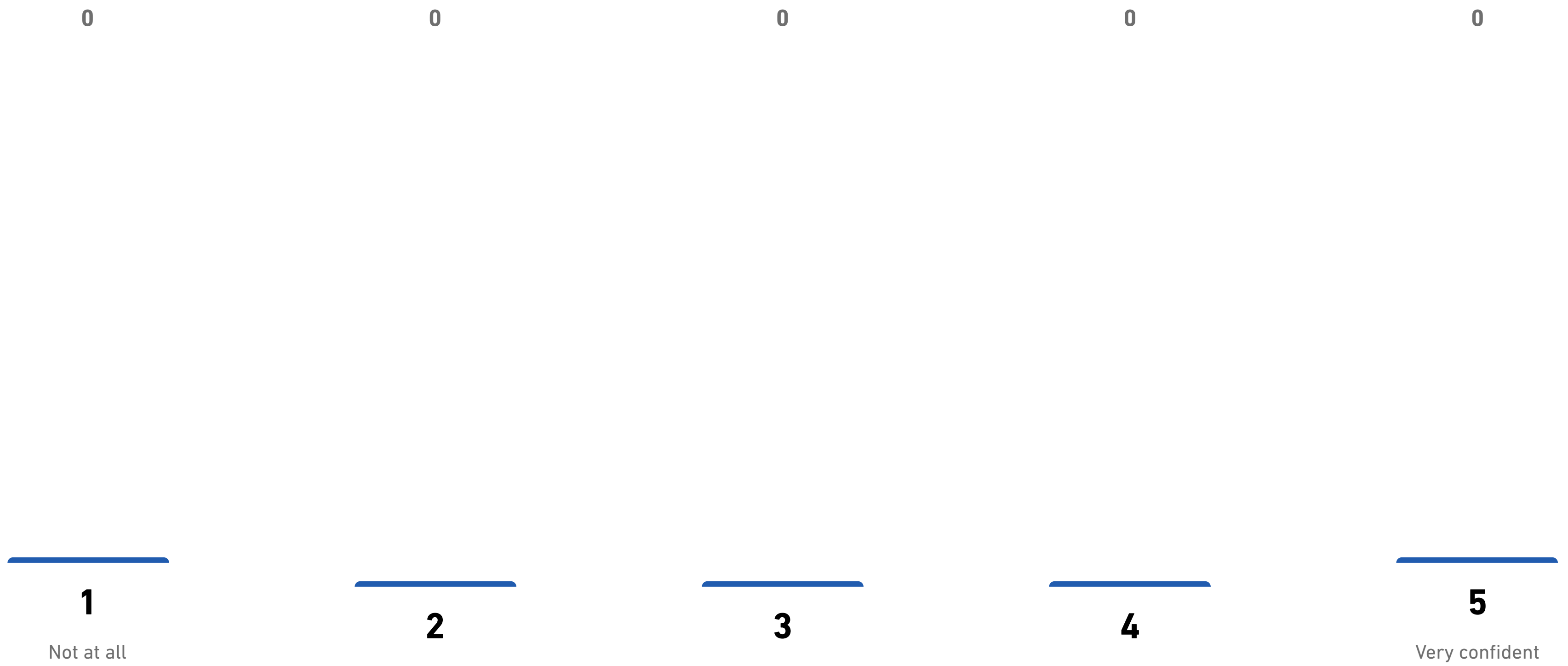
Reflection

Let’s discuss — no right answers

- Where do you already meet **machine learning** in your daily business — without noticing?
- Which idea changed how you look at "AI intelligence" — **embeddings**, **attention**, or **next-token prediction**?
- **What will you explain differently to your board tomorrow?**

How confident are you explaining AI to your board?

average · 0 responses



Take-Home Messages

Foundations, Transformers & LLMs

- Machine learning **learns behavior from data** instead of hand-written rules
- Everything starts with **encoding**: one-hot → **semantic embeddings** → **contextual embeddings**
- **Attention** produces context-dependent embeddings — the heart of the **Transformer**
- Generative LLMs work by repeatedly outputting a **probability distribution** over the next token
- **Watch for generalization: a model that memorizes (overfits) fails on new, real-world data**