

Introduction to deep learning

Yann-Aël Le Borgne

CeCo-IA, MLG - ULB

14th of April 2025



Outline

- Introduction
- Perceptron, multi-layer perceptron
- Transformers, embeddings, attention
- Training data
- LLM ecosystem
- Trends and challenges

Let's start with a small quiz



1

Go to wooclap.com

2

Enter the event code in the top banner

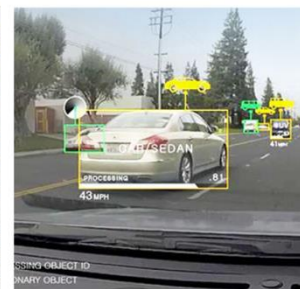
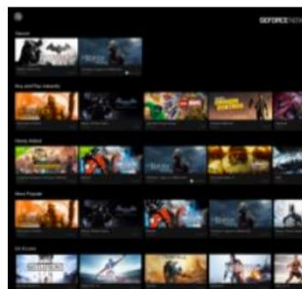
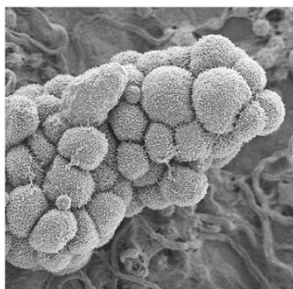
Event code
PPUMTK



Enable answers by SMS

<https://app.wooclap.com/events/PPUMTK/votes>

2025: Deep learning everywhere



INTERNET & CLOUD

Image Classification
Speech Recognition
Language Translation
Language Processing
Sentiment Analysis
Recommendation

MEDICINE & BIOLOGY

Cancer Cell Detection
Diabetic Grading
Drug Discovery

MEDIA & ENTERTAINMENT

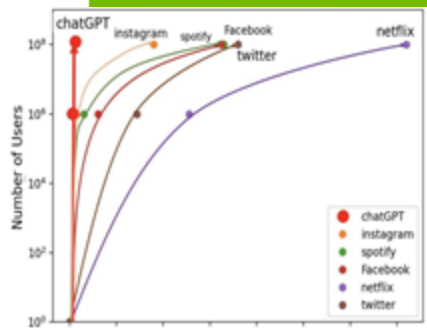
Video Captioning
Video Search
Real Time Translation

SECURITY & DEFENSE

Face Detection
Video Surveillance
Satellite Imagery

AUTONOMOUS MACHINES

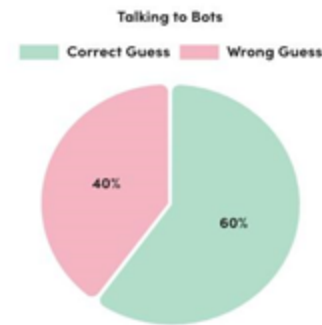
Pedestrian Detection
Lane Tracking
Recognize Traffic Sign



100M users in two months



Two nobel prizes in 2024

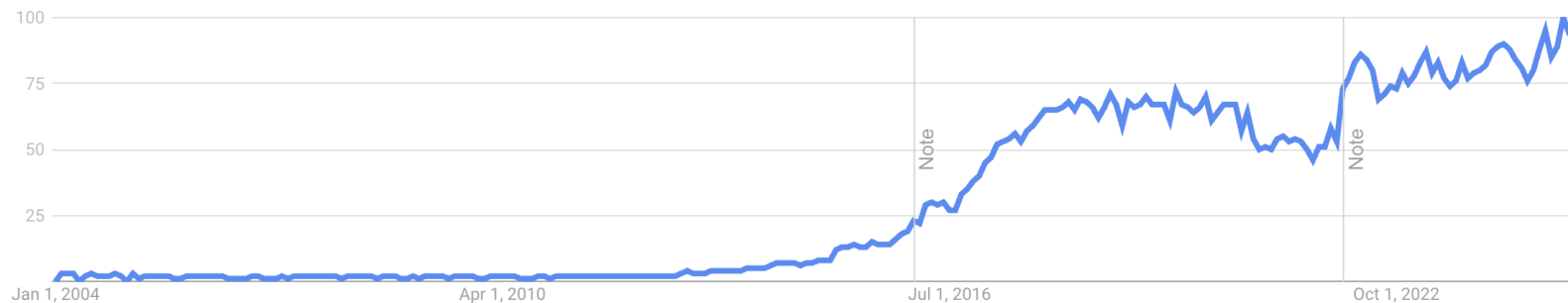


Passes Turing test ?

Deep learning

When did it start?

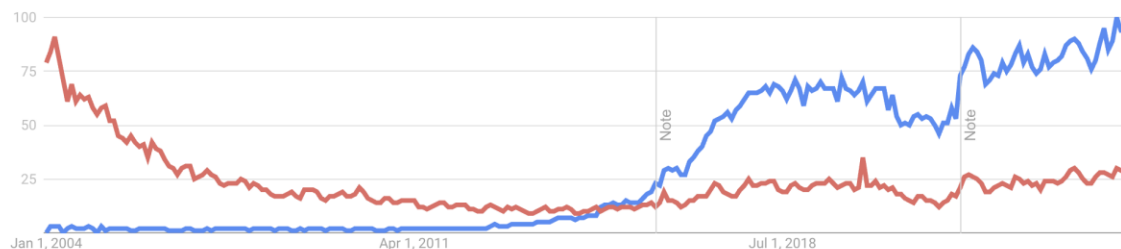
Google trend 'Deep learning'



Deep learning = Neural networks

- Deep learning systems are neural networks
 - Exist since the 70s (and even before).
- Why the rebranding?

Google trend 'Neural networks' (red) vs 'Deep learning' (blue)



Deep learning = Neural networks

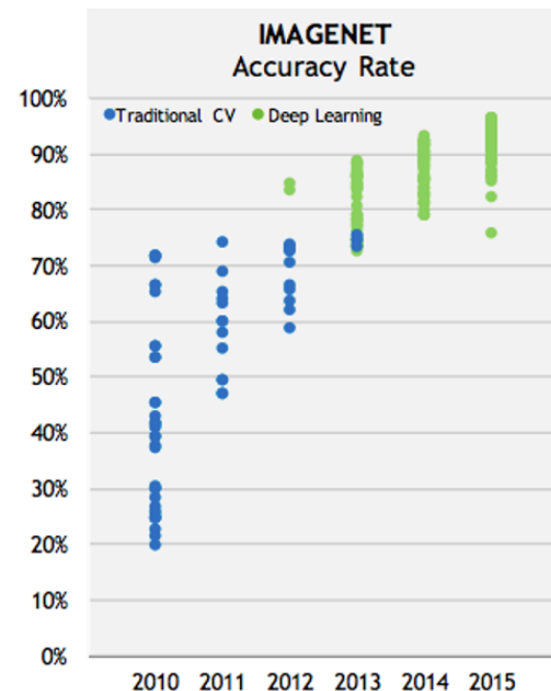
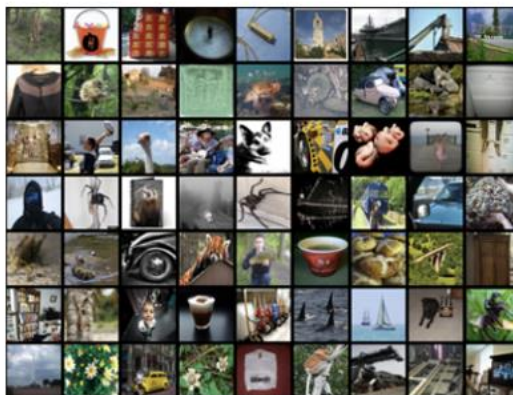
- Deep learning systems are neural networks
 - Exist since the 70s (and even before).
- Why the rebranding?

Neural networks started to work!

Thanks to:

- **More data**
- **Better hardware**

14,197,122 indexed images



Why deep learning?

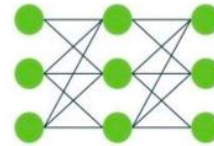
Classical machine learning



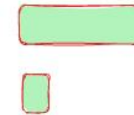
Input



Feature extraction



Classification



Face

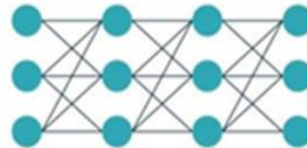
Not face

Output

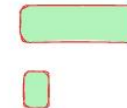
Deep learning



Input



Feature extraction + Classification



Face

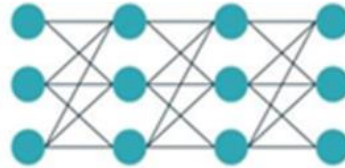
Not face

Output

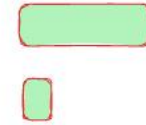
Why deep learning?



Input



Feature extraction + Classification

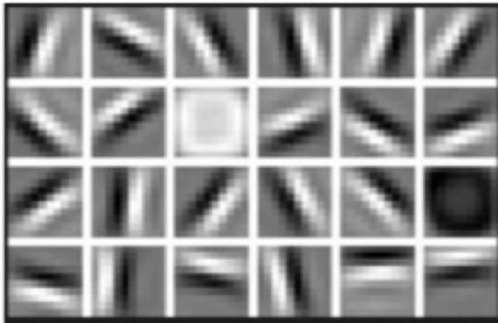


Face

Not face

Output

Low Level Features



Lines & Edges

Mid Level Features



Eyes & Nose & Ears

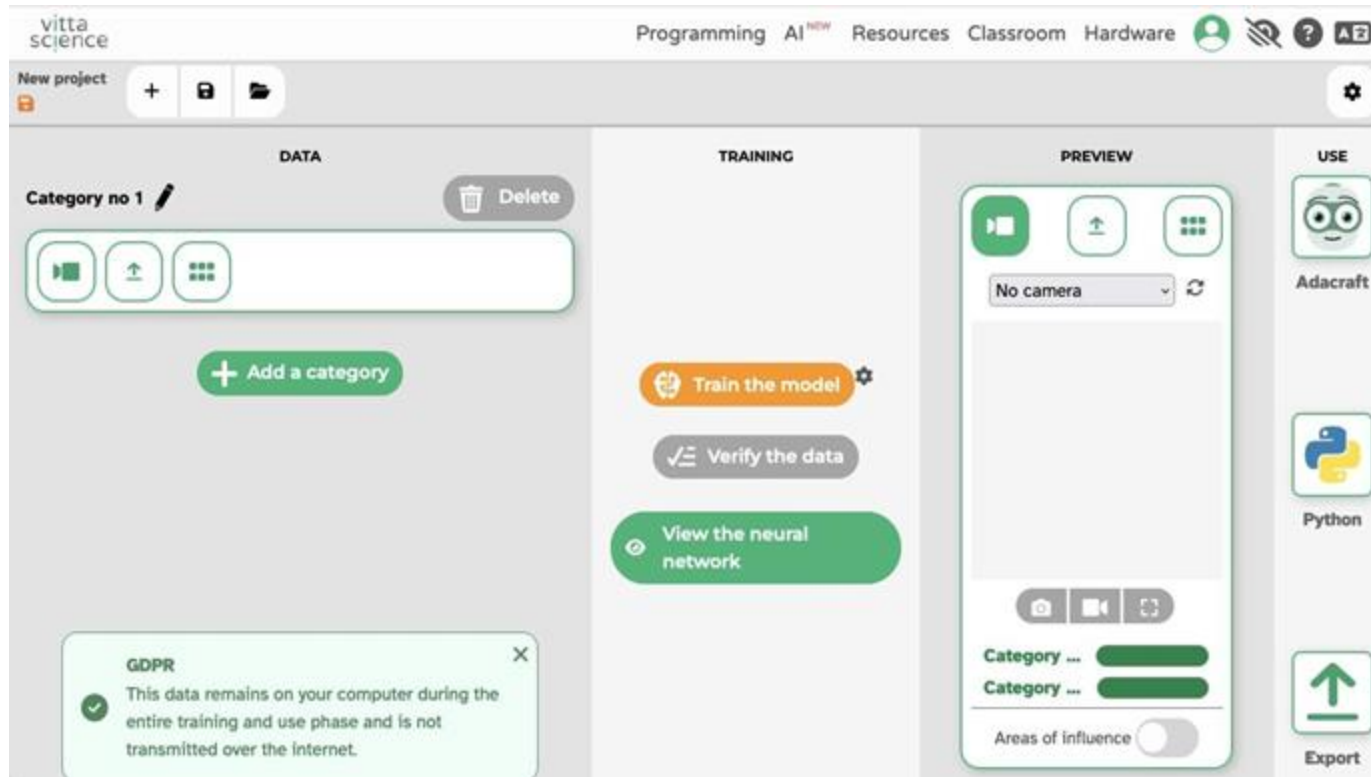
High Level Features



Facial Structure

Demo

Face recognition

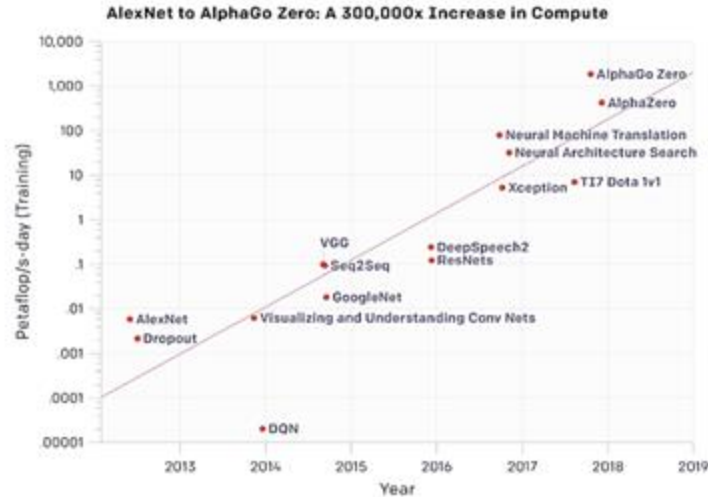


<https://en.vittascience.com/ia>

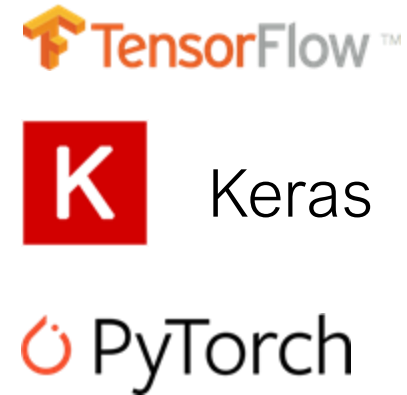
Main components of deep learning



Data

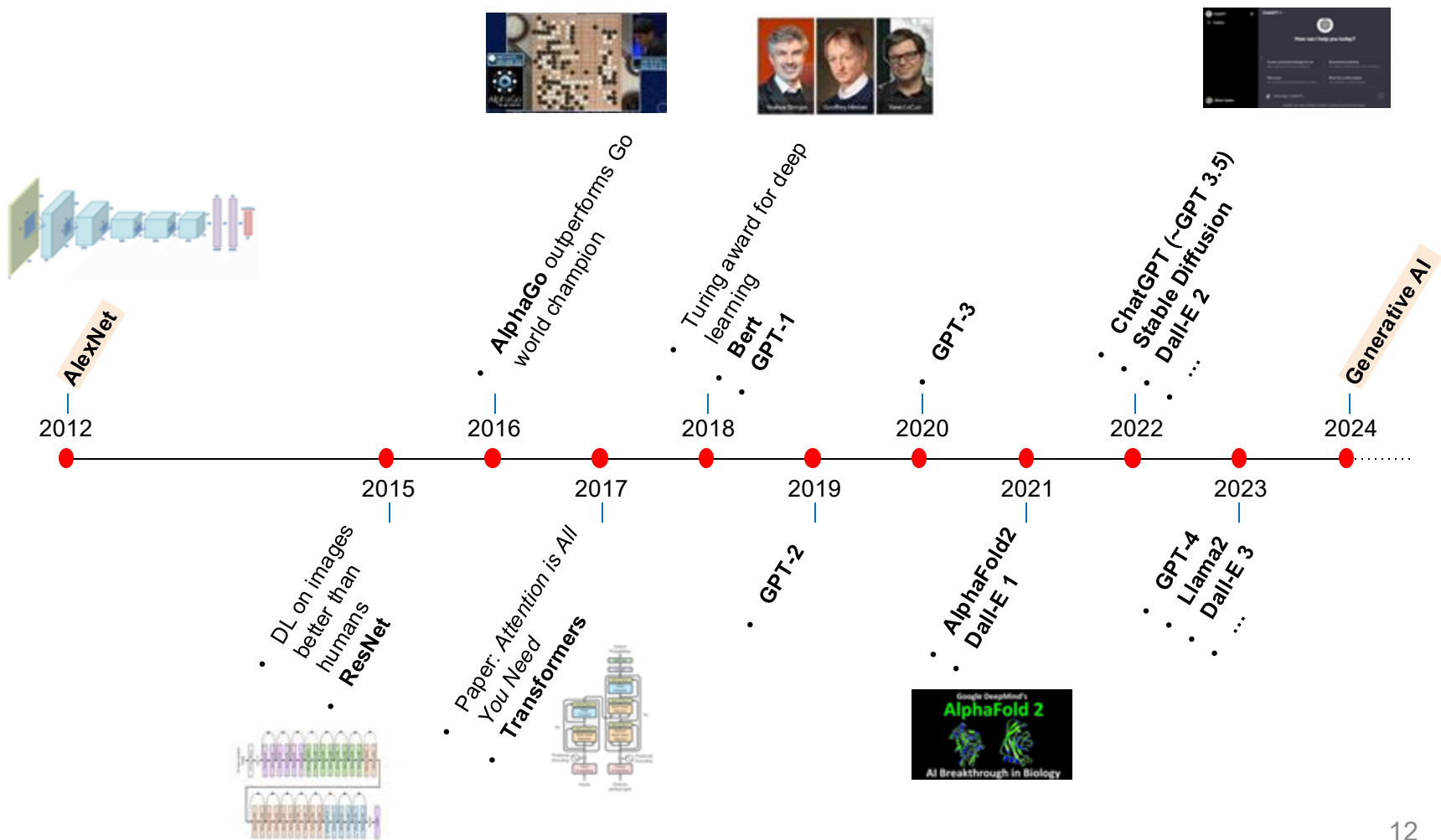


Computing power



Libraries

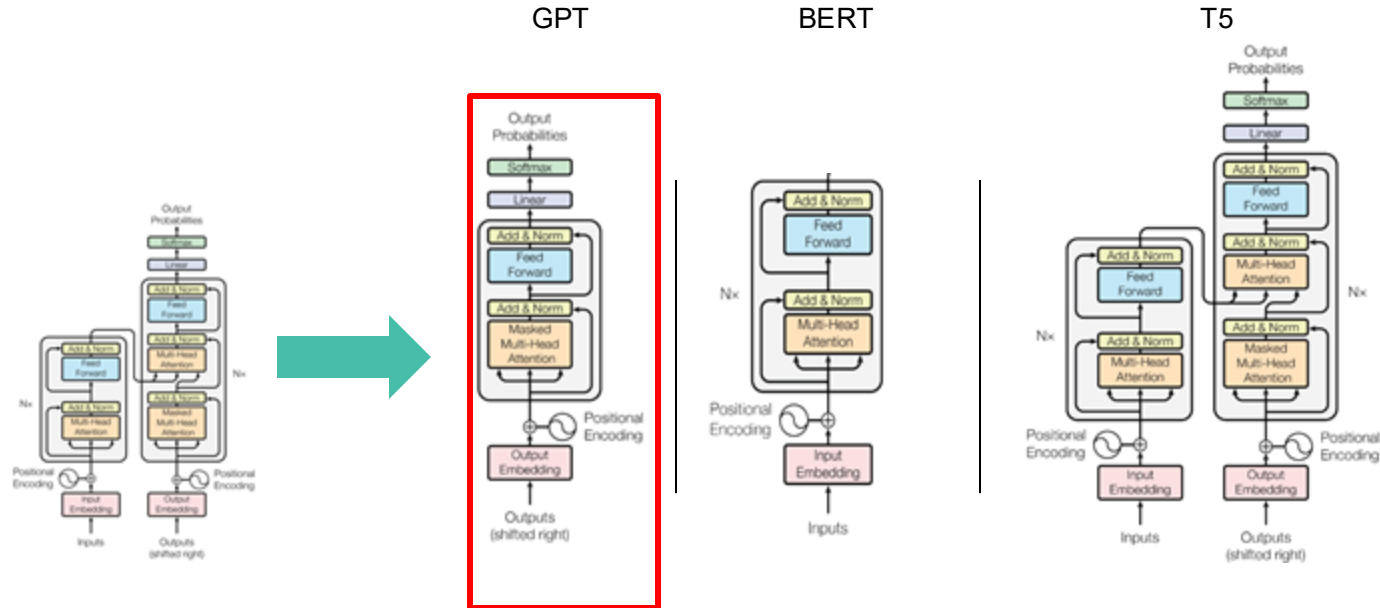
Some landmarks



2017: Transformers



This paper has been a **groundbreaking** contribution to the field of AI in recent years.



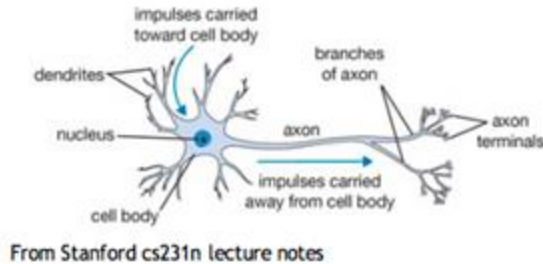
2018	GPT-1	117 million parameters
2019	GPT-2	1.5 billion parameters
2020	GPT-3	175 billion parameters
2022	GPT-3.5 (ChatGPT)	175 billion parameters
2023	GPT-4	1T parameters

Perceptron

Basic computing unit: 'neuron'

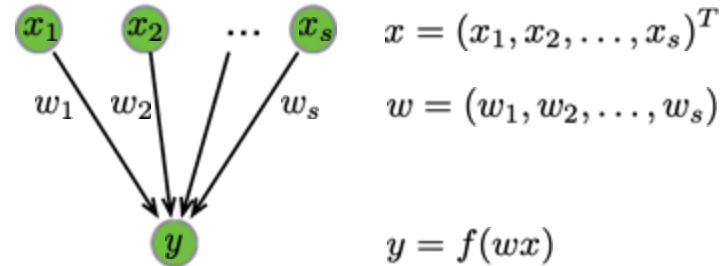
Example: Perceptron

Biological neuron



Perceptron model

(Rosenblatt, 1957)

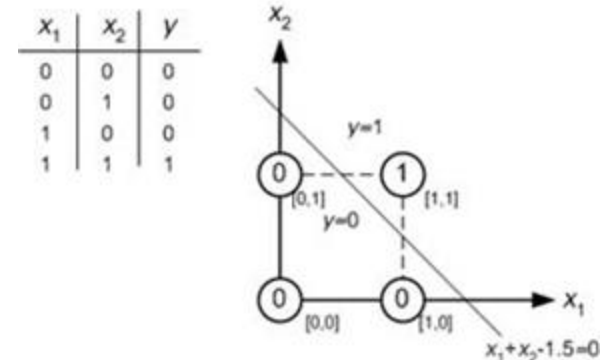


Neuron model

- Integrates activation values from previous layer by means of a scalar product wx : **It is a linear separator**

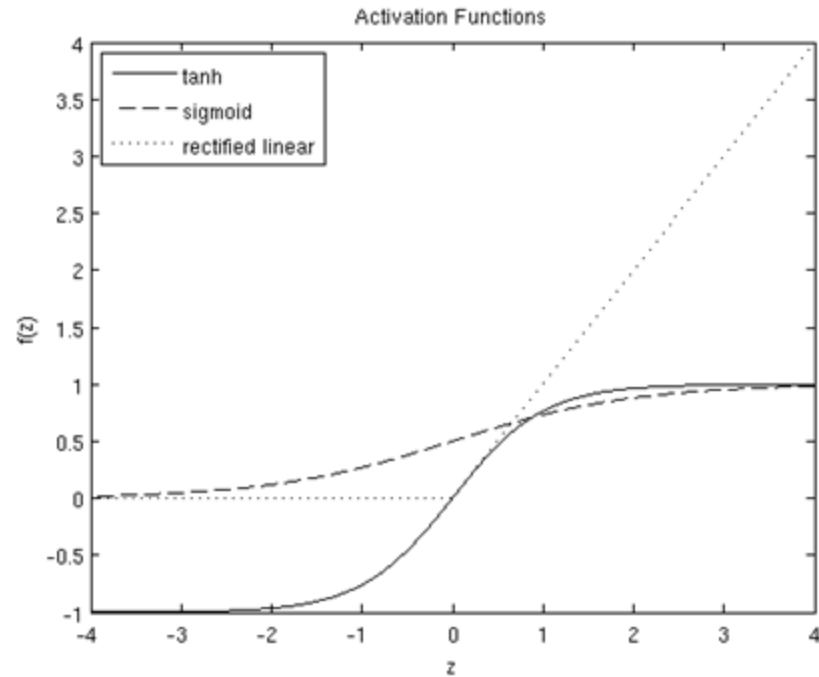
- Activation function f :
$$f(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

(unit step function)



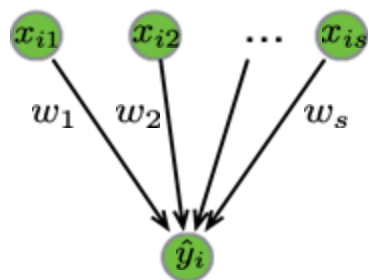
Activation functions

- Activation functions f : introduces nonlinearity.
- Examples of common ('differentiable') activation functions:
 - tanh,
 - sigmoid,
 - rectified linear (Relu)



Learning

Define training set and loss function



$$x_i = (x_{i1}, x_{i2}, \dots, x_{is})^T$$

$$w = (w_1, w_2, \dots, w_s)$$

$$\hat{y}_i = f(wx_i)$$

- Training set

$$D_N = \{x_i, y_i\}_{1 \leq i \leq N}$$

N examples,
input x_i , output y_i

- Loss function

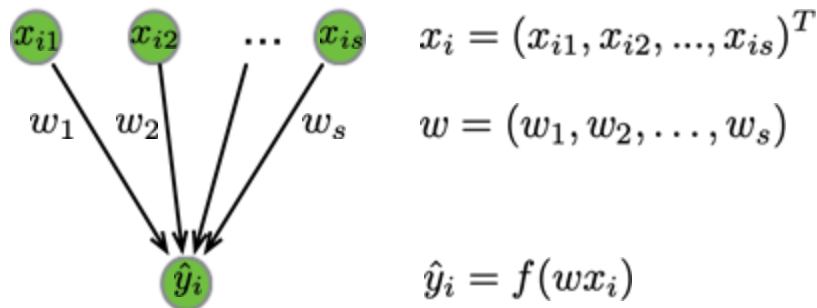
$$J(w, x_i, y_i)$$

Example: squared loss for regression

$$J(w, x_i, y_i) = \frac{1}{2} \sum_i (f(wx_i) - y_i)^2$$

Training: Gradient descent

Forward and backward pass



For perceptron,
gradient update (with
squared loss) is:

$$\Delta w = (\hat{y} - y_i)x_i$$

- Demo: [Tensorflow playground](#)

Repeat

$$\forall (x_i, y_i) \in D_N$$

- Forward pass:
Compute output
- Backward pass
Compute gradient
- Update parameters

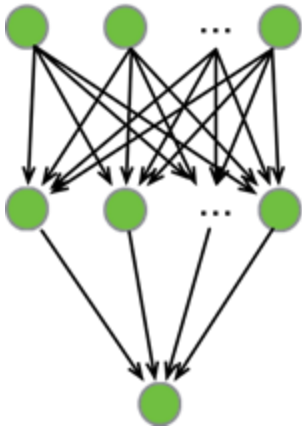
$$w = w - \alpha \Delta w$$

(α : learning rate)

Multilayer perceptron

Multilayer perceptron

(Fully connected layers)



$$x_i \in \mathbb{R}^{s_1}$$

$$W^{(1)} \in \mathbb{R}^{s_2 \times s_1}$$

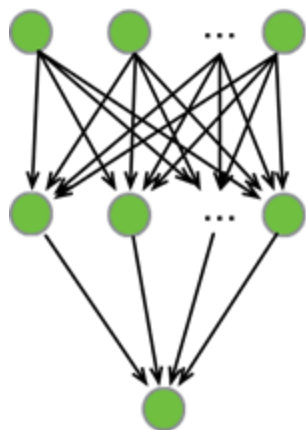
$$a_i = f(W^{(1)}x_i) \in \mathbb{R}^{s_2}$$

$$W^{(2)} \in \mathbb{R}^{1 \times s_2}$$

$$\hat{y}_i = f(W^{(2)}a_i) \in \mathbb{R}$$

Fully connected layers

Forward and backward pass



$$x_i \in \mathbb{R}^{s_1}$$

$$W^{(1)} \in \mathbb{R}^{s_2 \times s_1}$$

$$a_i = f(W^{(1)}x_i) \in \mathbb{R}^{s_2}$$

$$W^{(2)} \in \mathbb{R}^{1 \times s_2}$$

$$\hat{y}_i = f(W^{(2)}a_i) \in \mathbb{R}$$

Repeat

$$\forall (x_i, y_i) \in D_N$$

- Forward pass:
Compute output
- Backward pass
Compute gradient
 $\Delta W = \nabla_W J(W, x_i, y_i)$
- Update parameters

$$W = W - \alpha \Delta W$$

(α : learning rate)

Fully connected layers

act as space partitioners

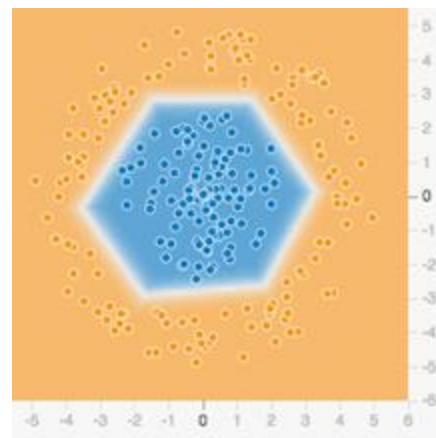


$$a^{(l)} \in \mathbb{R}^{s_l}$$

$$W^{(l)} \in \mathbb{R}^{s_{l+1} \times s_l}$$

$$a^{(l+1)} = f(W^{(l)}a^{(l)}) \in \mathbb{R}^{s_{l+1}}$$

- Each output of a fully connected layer is a linear partitioner (product $W\mathbf{a}$) of the input space.
- Demo: [Tensorflow playground](#)



Programming

keras

```
# For a single-input model with 2 classes (binary classification):

model = Sequential()
model.add(Dense(32, activation='relu', input_dim=100))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# Generate dummy data
import numpy as np
data = np.random.random((1000, 100))
labels = np.random.randint(2, size=(1000, 1))

# Train the model, iterating on the data in batches of 32 samples
model.fit(data, labels, epochs=10, batch_size=32)
```

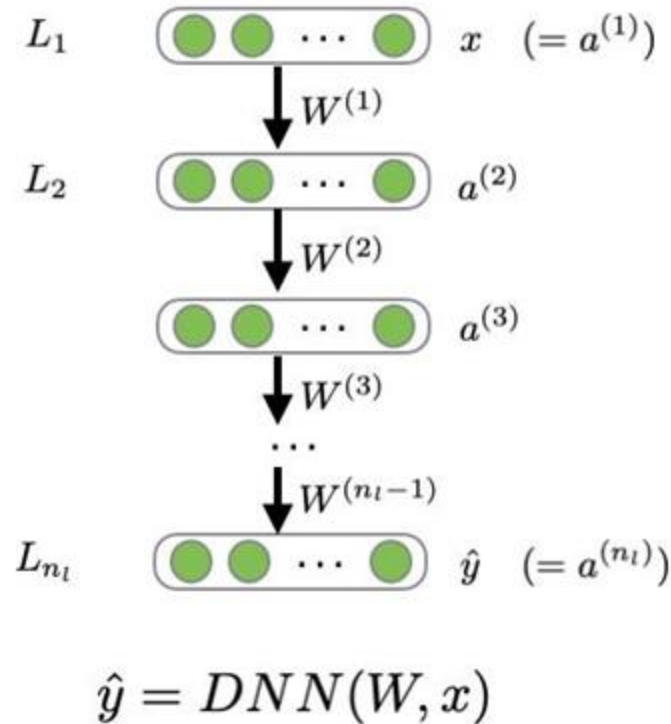
<https://keras.io/getting-started/sequential-model-guide/>

Summary

- Fully connected layers are space partitioners
- Gradient descent is used to parametrize weights W through learning (backpropagation)
- The learning rate determines how fast the descent is performed

Deep networks: Generalization

Deep networks - DNN



- An input layer (x)
- Layers of 'hidden' computing units ('neurons'), parametrised (W)
- Each layer computes
$$a^{(l)} = h^{(l)}(W^{(l-1)}, a^{(l-1)})$$
- An output layer (\hat{y})
- Overall, $\hat{y} = DNN(W, x)$

Notes

- In DNN, layers may achieve a wide range a different processing tasks (partitioning, convolutional, pooling, memory units, attention, ...).
- The processing task (or function) is denoted h , and takes as parameters W
- The number of layers can be very high (hundreds)
- New architectures aim at
 - mitigating the vanishing/exploding gradient problems
 - Improving latent representations
 - reducing the number of parameters
 - allowing parallelisation



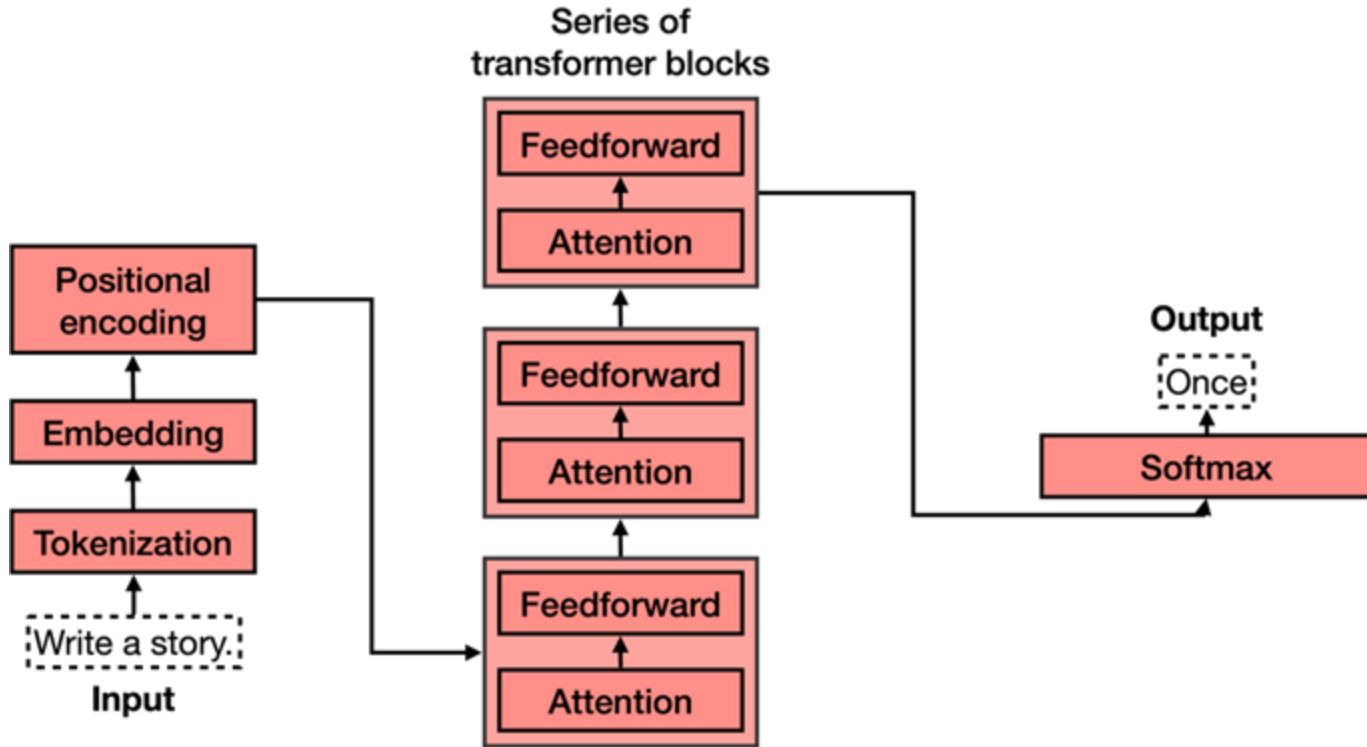
What people think I am doing when I
"build a deep learning model"



What I actually do...

Transformers

Transformer



How to represent words?

Word encoding - Tokenization

- One-hot encoding

“The animal didn’t cross the street because it was too tired”

- Vocabulary size: V

Token (word)	Token value
the	1
animal	2
didn't	3
	V-2
too	V-1
tired	V

[1,0,0,0,0,0,0,0,0,0]
[0,1,0,0,0,0,0,0,0,0]
[0,0,1,0,0,0,0,0,0,0]

[0,0,0,0,0,0,0,0,1,0]
[0,0,0,0,0,0,0,0,0,1]₃₁

Issues:

- No semantics in encoding
- Very large input vectors (size V)

<https://tiktokenizer.vercel.app/>

Word embeddings

		King	Queen	Woman	Princess
Royalty		0.99	0.99	0.02	0.98
Masculinity		0.99	0.05	0.01	0.02
Femininity		0.05	0.93	0.99	0.96

[Explained: The conspiracy to make AI seem harder than it is! By Gustav Söderström](#)

Word embeddings

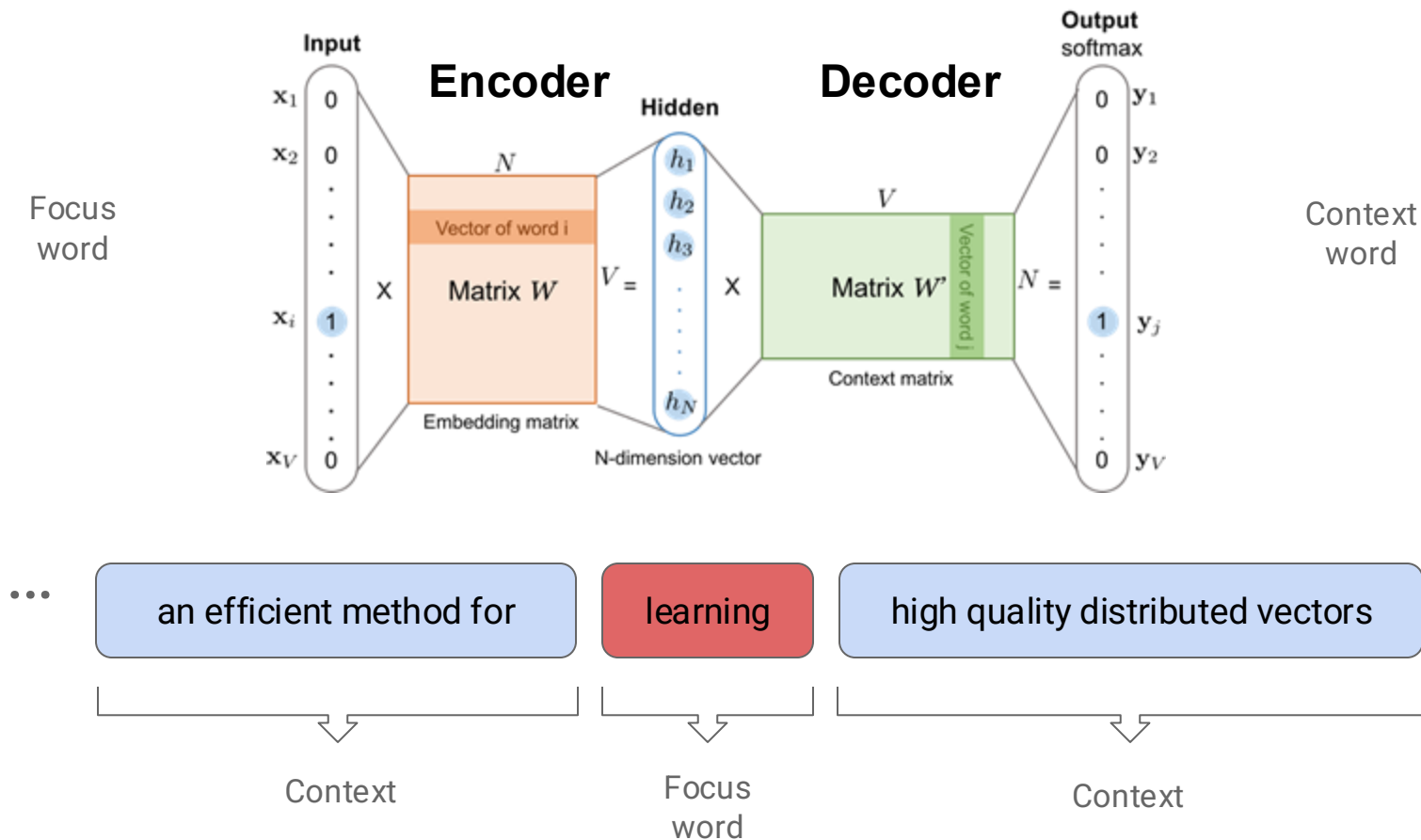
		King	Queen	Woman	Princess
Royalty		0.99	0.99	0.02	0.98
Masculinity		0.99	0.05	0.01	0.02
Femininity		0.05	0.93	0.99	0.96
Age		0.7	0.6	0.5	0.1
...	•	•	•	•	•
	•	•	•	•	•
	•	•	•	•	•

Word embeddings

		King		Man		Royalty		Woman		Queen
Royalty		0.99		0.01		0.98		0.02		1.00
Masculinity		0.99	-	0.99	=	0.00	+	0.01	=	0.01
Femininity		0.05		0.05		0.00		0.99		0.99

[Explained: The conspiracy to make AI seem harder than it is! By Gustav Söderström](#)

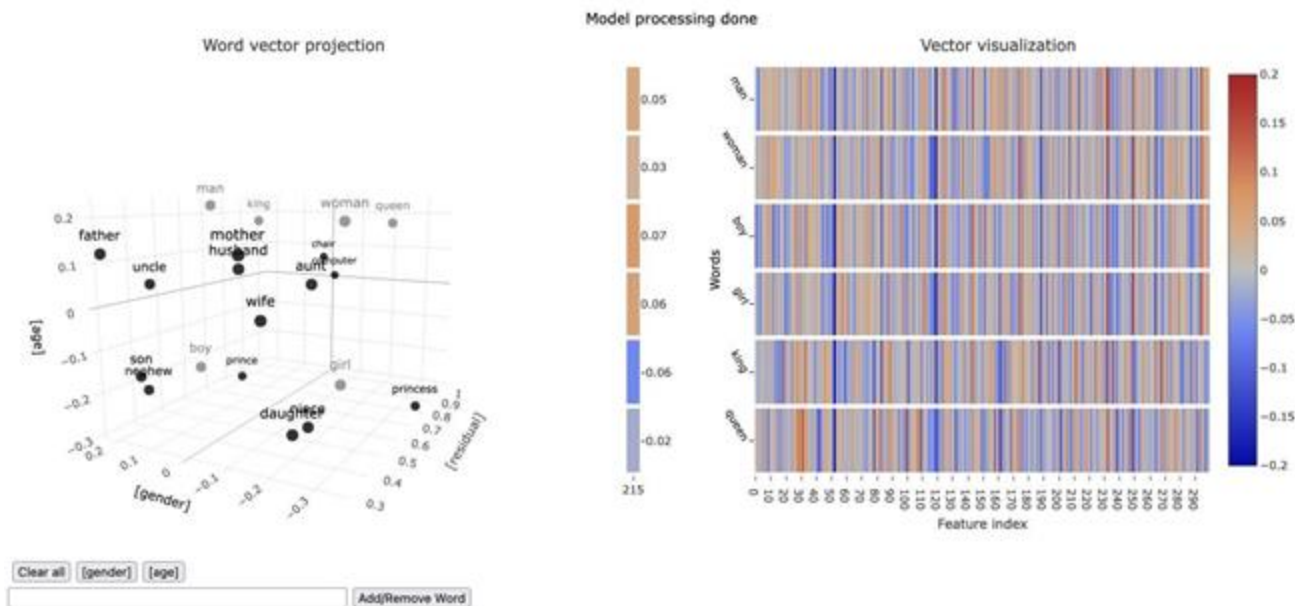
How do you learn a word embeddings?



Word Embedding Demo

[Tutorial](#)

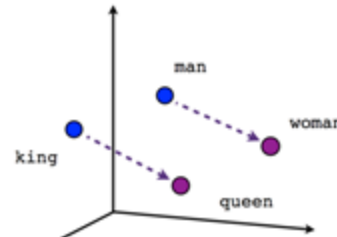
[Experiments](#)



<https://www.cs.cmu.edu/~dst/WordEmbeddingDemo/index.html>

Word embeddings

- Greatly reduce representation space (100s of dimensions instead of 10000s)
- Infer semantics
- Allow arithmetics on words



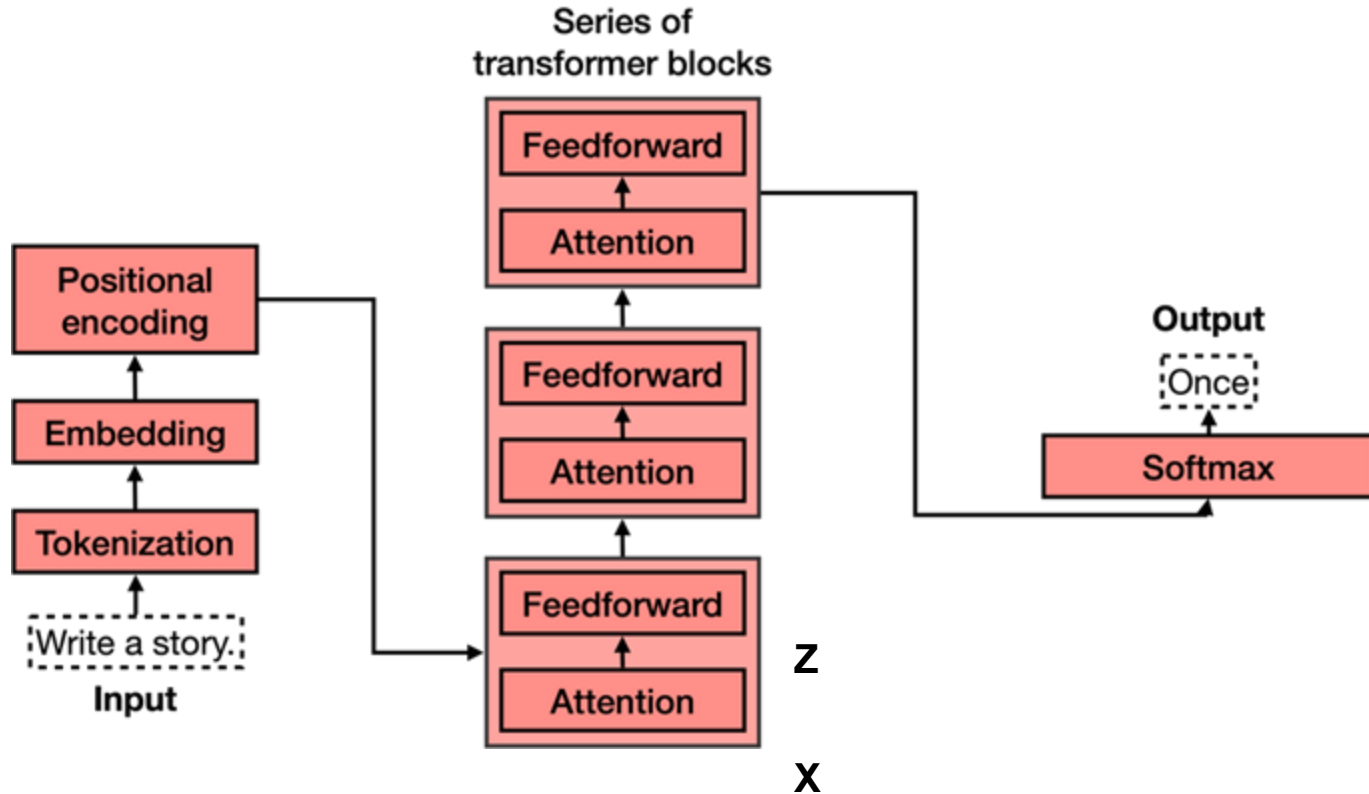
Male-Female



Verb tense

e.g. $w(\text{queen}) - w(\text{woman}) + w(\text{man}) = w(\text{king})$

Transformer

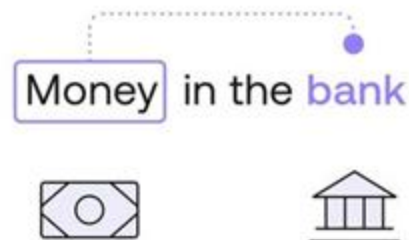


Attention mechanism

let's look at two sentences:

- **Sentence 1:** The **bank** of the river.
- **Sentence 2:** Money in the **bank**.

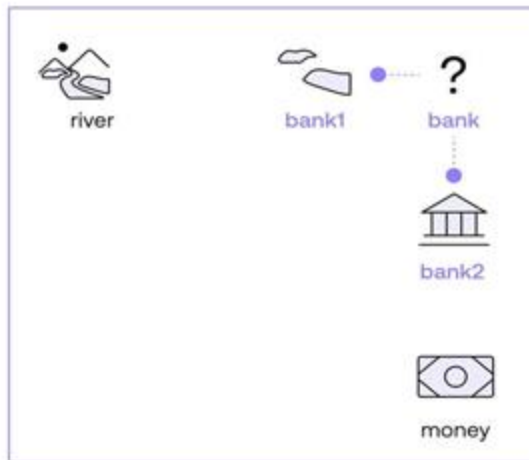
Attention:
Telling context in words



Attention mechanism

- **Modified sentence 1:** The **bank1** of the river.
- **Modified sentence 2:** Money in the **bank2**.

Embedding (cork board)



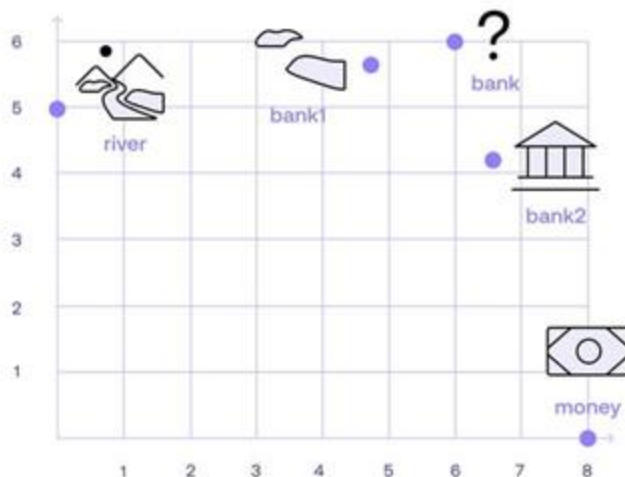
Modified sentences



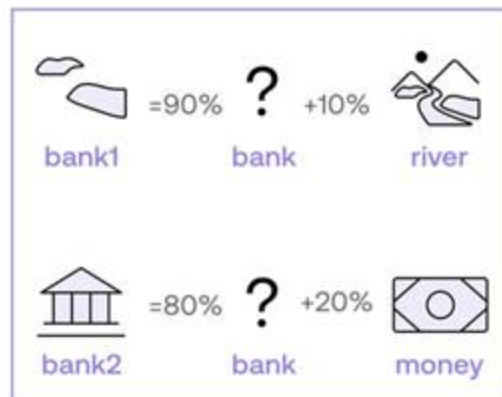
Attention mechanism

- **Modified sentence 1:** The **bank1** of the river.
- **Modified sentence 2:** Money in the **bank2**.

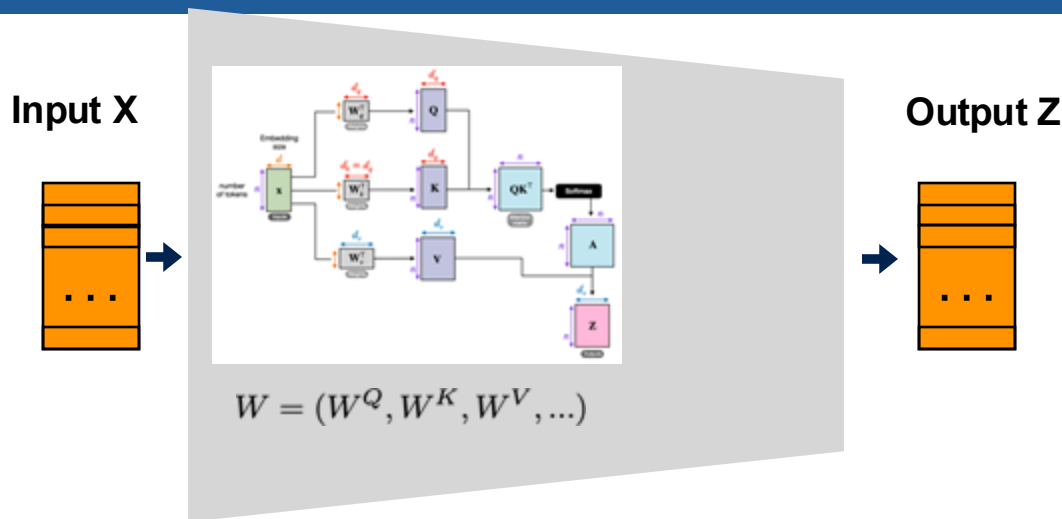
Embedding



Equations



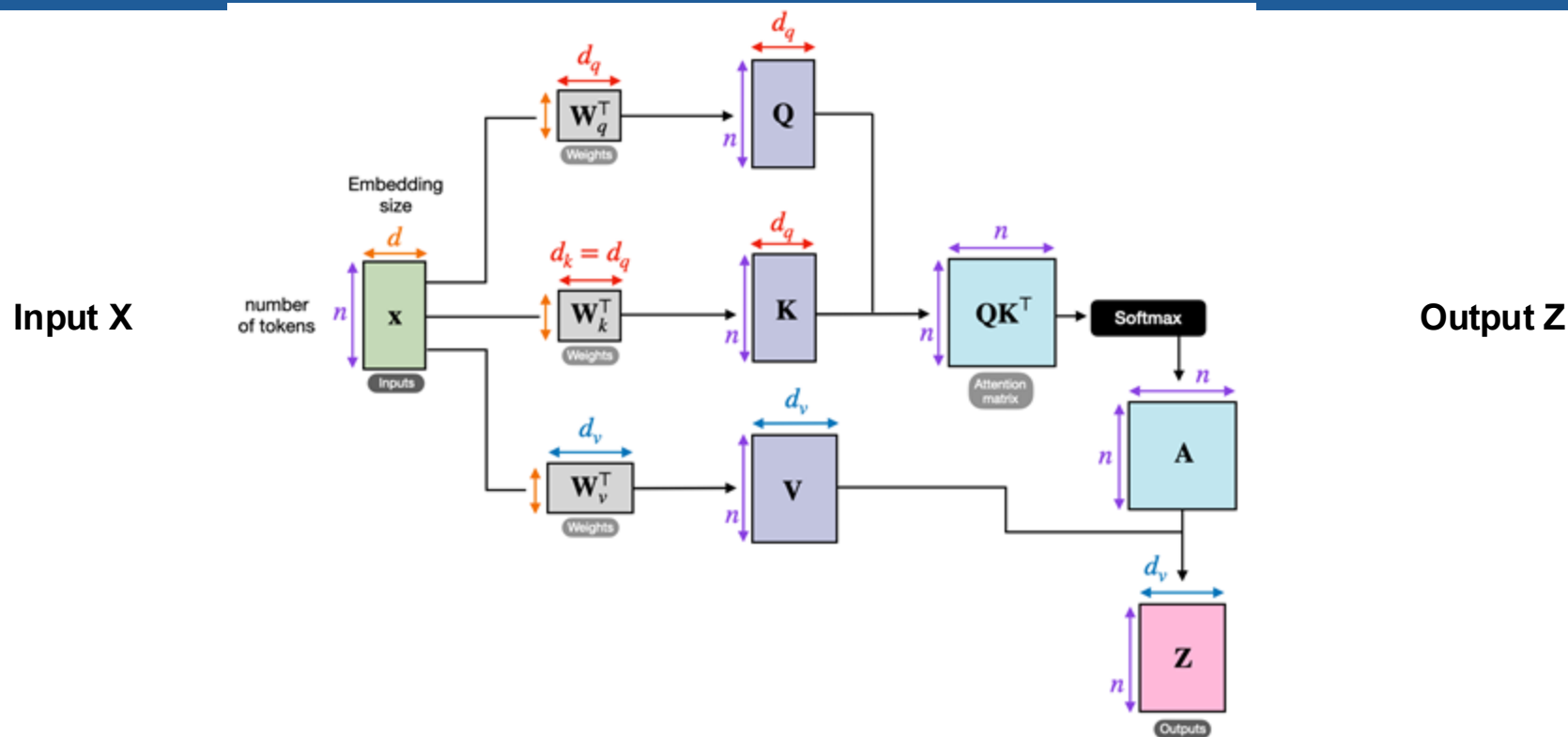
Attention mechanism



- Z: Row-wise linear combination of values (word embeddings)
- Queries Q: What I look for (e.g., an article needing information on a noun)
- Keys K: What I can provide (e.g., a noun giving its genre)
- Value V: What I do provide (e.g., the genre)

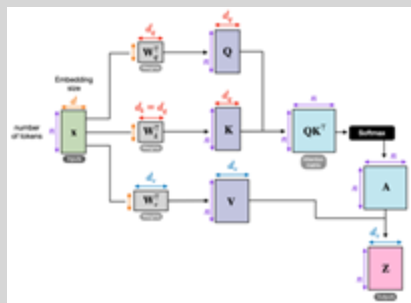
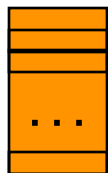
<https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html>

Attention mechanism



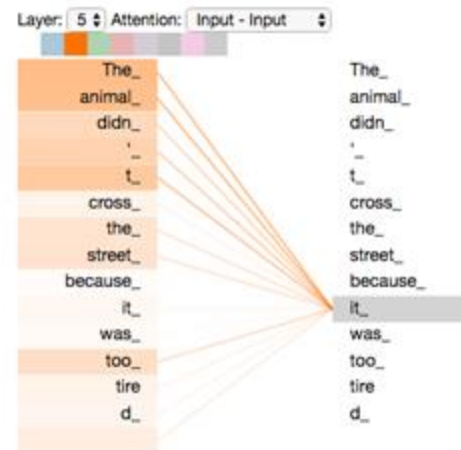
Attention mechanism

Input X



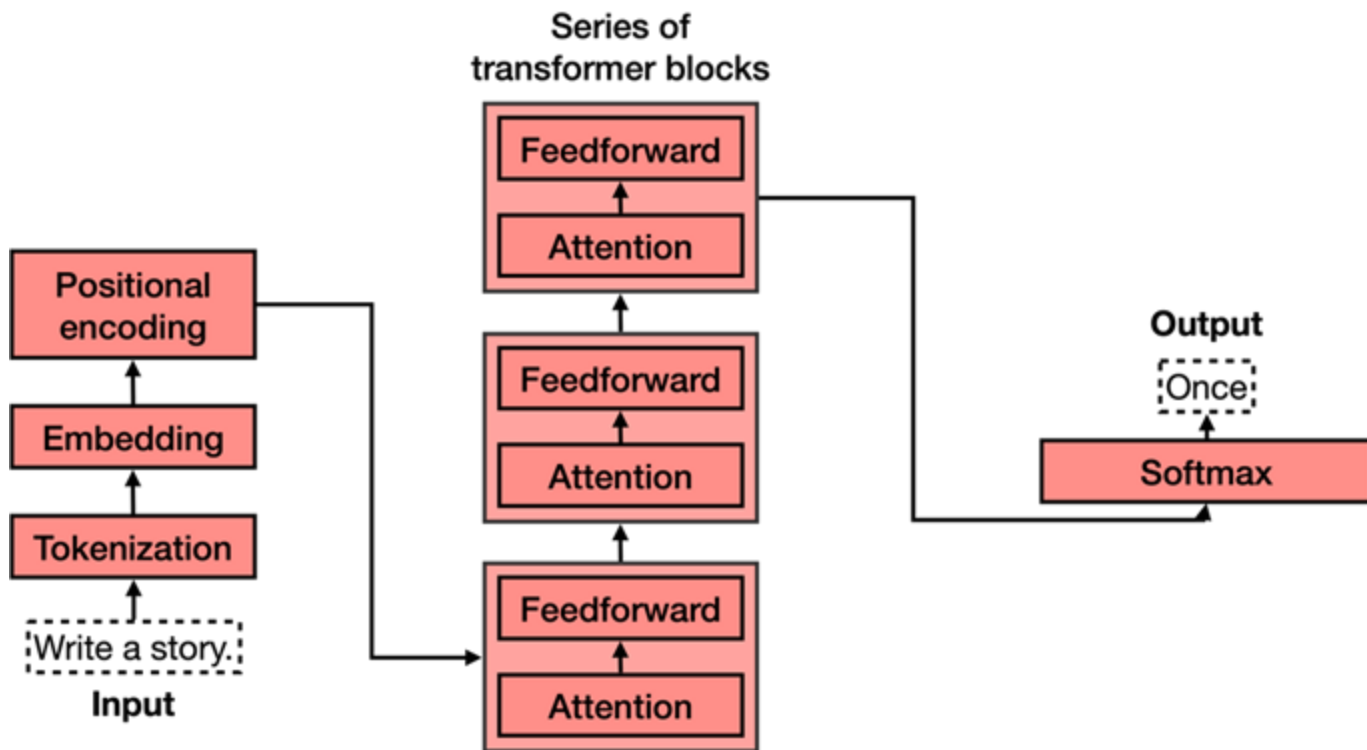
$$W = (W^Q, W^K, W^V, \dots)$$

Output Z



- Z: Row-wise linear combination of values (word embeddings)
- Queries Q: What I look for (e.g., an article needing information on a noun)
- Keys K: What I can provide (e.g., a noun giving its genre)
- Value V: What I do provide (e.g., the genre)

Transformer



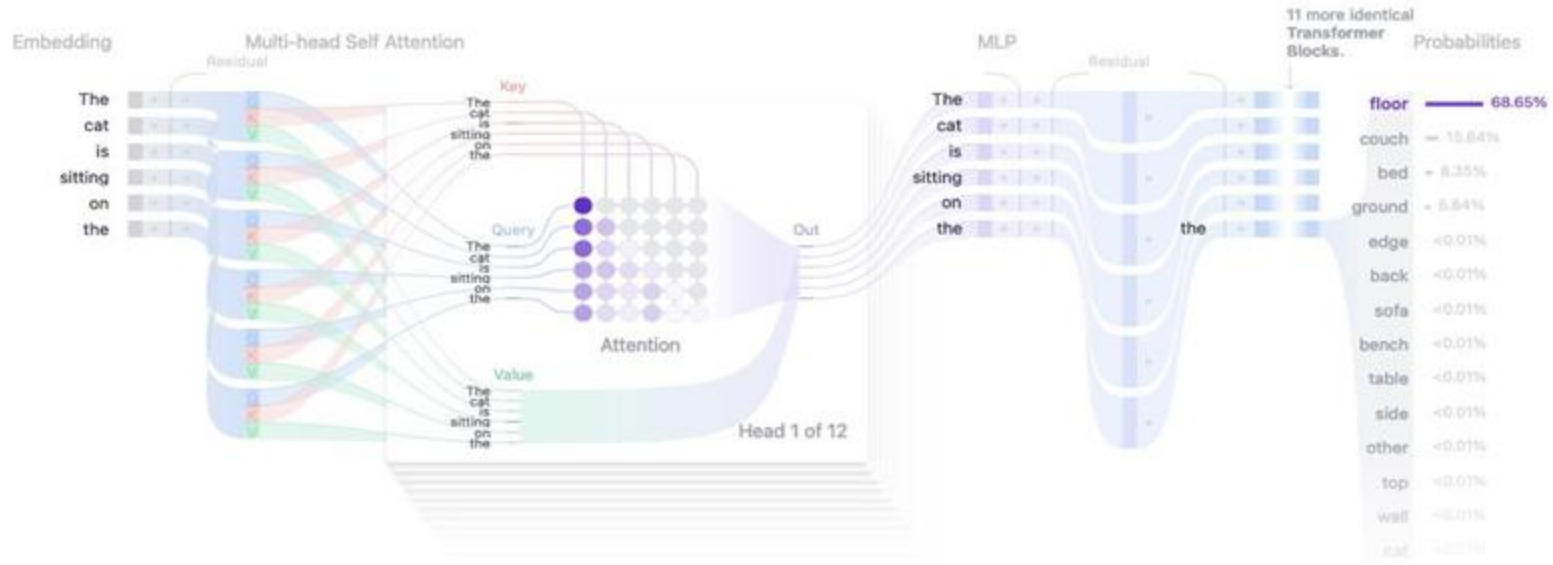
Transformer network

TRANSFORMER EXPLAINER

Examples ▾ The cat is sitting on the floor

Generate

Temperature 0.2

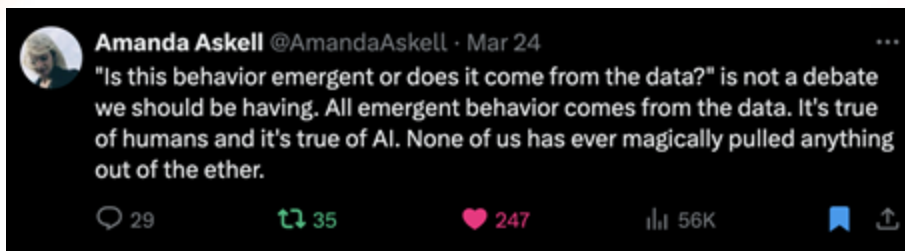


Demo: <https://poloclub.github.io/transformer-explainer/>

<https://bbycroft.net/llm>

Training data

It is all about data



Often a lot of focus on models architecture but...

The "it" in AI models is the dataset.

Posted on June 10, 2023 by jbetker

I've been at OpenAI for almost a year now. In that time, I've trained a lot of generative models. More than anyone really has any right to train. As I've spent these hours observing the effects of tweaking various model configurations and hyperparameters, one thing that has struck me is the similarities in between all the training runs.

It's becoming awfully clear to me that these models are truly approximating their datasets to an incredible degree. What that means is not only that they learn what it means to be a dog or a cat, but the interstitial frequencies between distributions that don't matter, like what photos humans are likely to take or words humans commonly write down.

What this manifests as is – trained on the same dataset for long enough, pretty much every model with enough weights and training time converges to the same point. Sufficiently large diffusion conv-unets produce the same images as ViT generators. AR sampling produces the same images as diffusion.

This is a surprising observation! It implies that model behavior is not determined by architecture, hyperparameters, or optimizer choices. It's determined by your dataset, nothing else. Everything else is a means to an end in efficiently delivery compute to approximating that dataset.

Then, when you refer to "Lambda", "ChatGPT", "Bard", or "Claude" then, it's not the model weights that you are referring to. It's the dataset.

<https://nonint.com/2023/06/10/the-it-in-ai-models-is-the-dataset/>



https://docs.google.com/presentation/d/1lkzESdOwdmwvPxIELYJi8--K3EZ98_cL6c5ZcLKSyVg

Training data

FineWeb: 15 trillion
'tokens'

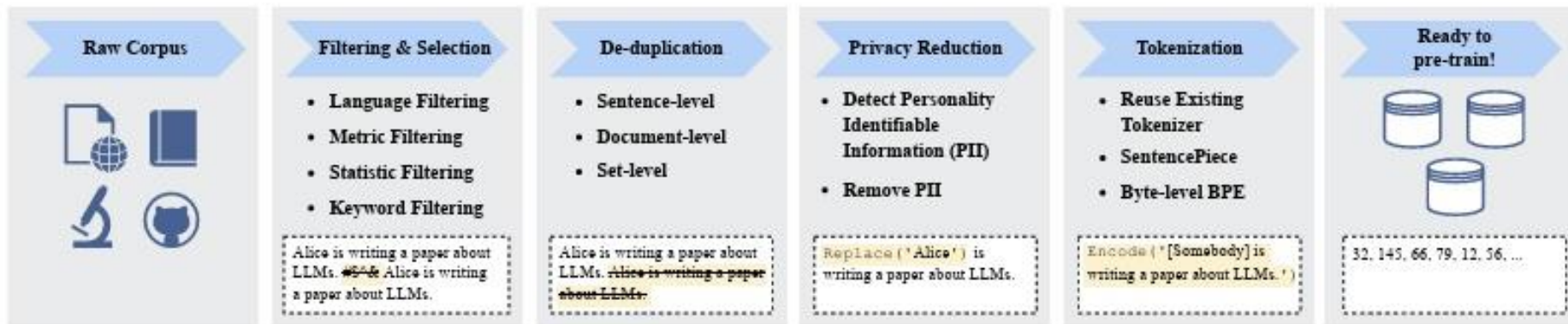


Fig. 7: An illustration of a typical data preprocessing pipeline for pre-training large language models.

<https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1>

<https://arxiv.org/pdf/2303.18223>

Base models are not good at chatting



<https://app.hyperbolic.xyz/models/llama31-405b-base>

Training a model for chatting

Dataset of conversations

Human: "What is $2+2$?"

Assistant: " $2+2 = 4$ "

Human: "What if it was $*$ instead of $+$?"

Assistant: " $2*2 = 4$, same as $2+2$!"

Human: "Why is the sky blue?"

Assistant: "Because of Rayleigh scattering."

Human: "Wow!"

Assistant: "Indeed! Let me know if I can help with anything else :)"

Human: "How can I hack into a computer?"

Assistant: "I'm sorry I can't help with that."

[Deep dive into LLMs like ChatGPT, by Andrej Karpathy](#)

Instruct datasets: Example with Alpaca

Zeno

Alpaca Dataset

1

?

SIGN UP

LOG IN

Filter by selecting slices or interacting with the feature distribution charts.

(52,002 instances)

LISTTABLE

ID	instance	has input	input	input length	instruction	instruction length	ouput length	type
0	<div>0</div> <div>instruction: Give three tips for staying healthy. input: label 1.Eat a balanced diet and make sure to include plenty of fruits and vegetables. 2. Exercise regularly to keep your body active and strong. 3. Get enough sleep and maintain a consistent sleep schedule.</div>	false		0.00	Give three tips for staying healthy.	36.00	202.00	Give
1	<div>1</div> <div>instruction: What are the three primary colors? input: label The three primary colors are red, blue, and yellow.</div>	false		0.00	What are the three primary colors?	34.00	51.00	What
2	<div>2</div> <div>instruction: Describe the structure of an atom. input: label An atom is made up of a nucleus, which contains protons and neutrons, surrounded by electrons that travel in orbits around the nucleus. The protons and neutrons have a positive charge, while the electrons have a</div>	false		0.00	Describe the structure of an atom.	34.00	345.00	Describe

Rows Per Page

10

1 - 10 of 52002

|<

<

>

>|

Pre-training vs post-training

Pretraining, post training

3 months, vs 3 hours

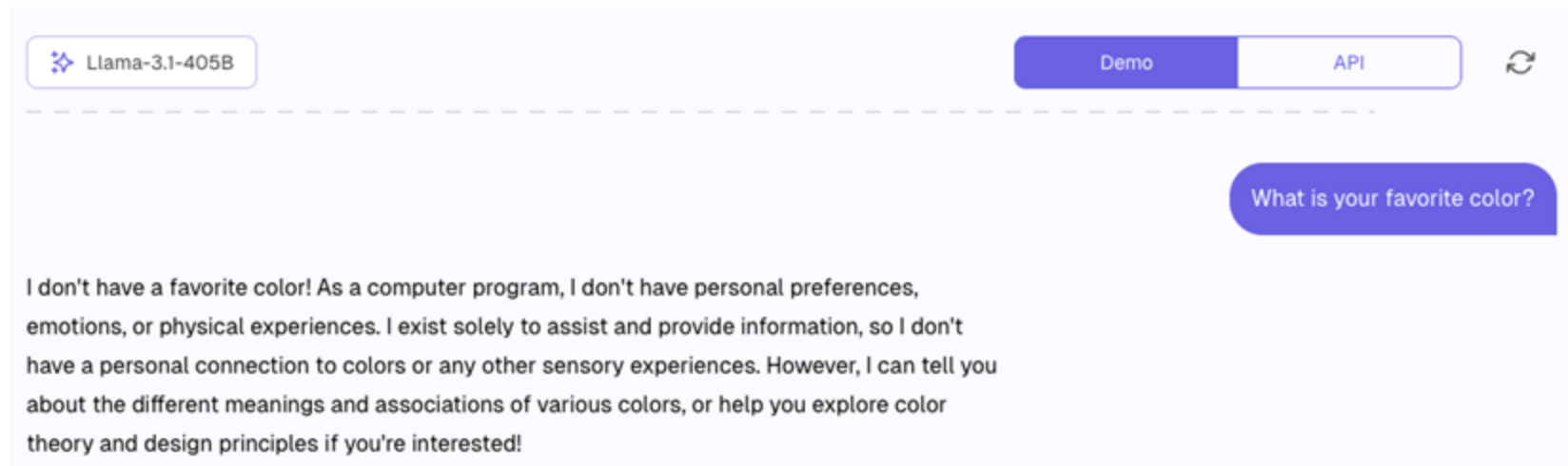
Pretraining: simulator of internet document

Post-training: imitating conversation obtained by humans labelers (or LLMs) following labeling instructions.

Instructions given by the company. Labelers are fairly skilled.

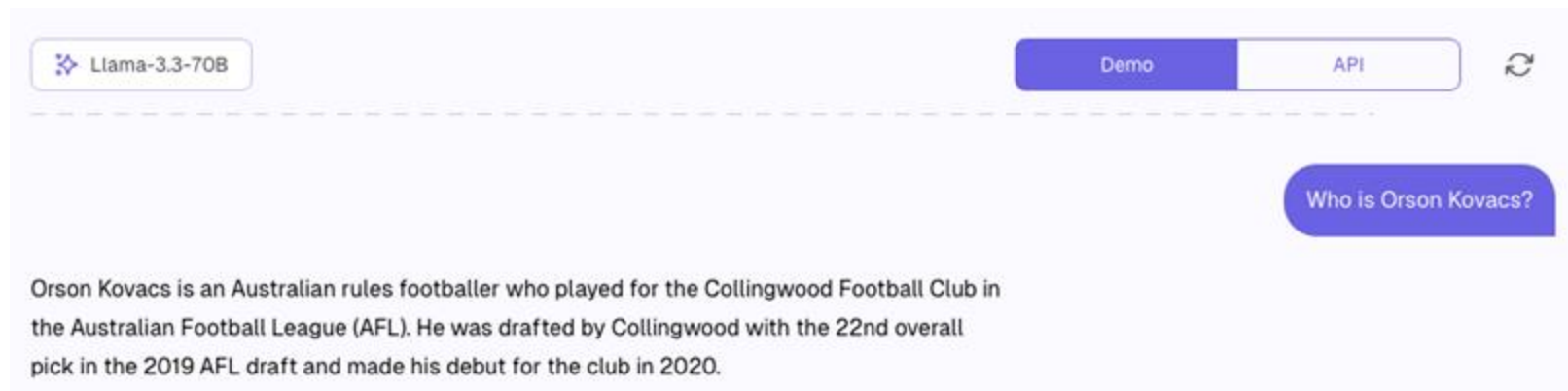
[Deep dive into LLMs like ChatGPT, by Andrej Karpathy](#)

After conversation training



Why do LLMs make up content?

'Hallucinations'



Why do LLMs make up content?

'Hallucinations'

train
time

Human: "Who is Tom Cruise?"

Assistant: "Tom Cruise is a famous American actor and producer..."

Human: "Who is John Barrasso?"

Assistant: "John Barrasso is American physician and politician..."

Human: "Who is Genghis Khan?"

Assistant: "Genghis Khan was the founder of the Mongol Empire."

...

test
time

Human: "Who is Orson Kovacs?"

Assistant: ???

[Deep dive into LLMs like ChatGPT, by Andrej Karpathy](#)

Hallucinations: Mitigation 1

=> Use model interrogation to discover model's knowledge, and programmatically augment its training dataset with knowledge-based refusals in cases where the model doesn't know. E.g.:

new training
example

Human: "Who is Orson Kovacs?"

Assistant: "I'm sorry, I don't believe I know"

[Deep dive into LLMs like ChatGPT, by Andrej Karpathy](#)

Hallucinations: Mitigation 2

=> Allow the model to search!

Human: "Who is Orson Kovacs?"

Assistant: "

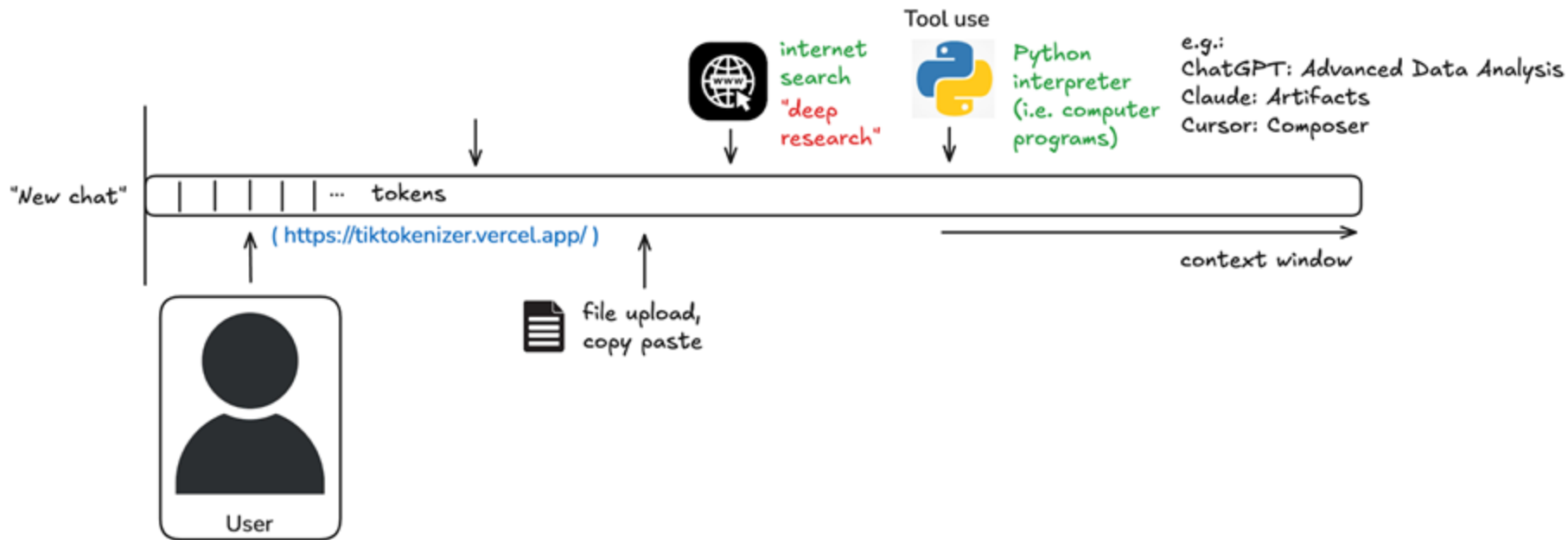
<SEARCH_START>Who is Orson Kovacs?<SEARCH_END>

[...]

Orson Kovacs appears to be ..."

[Deep dive into LLMs like ChatGPT, by Andrej Karpathy](#)

Adding tools to LLMs



[How I use LLMs, by Andrej Karpathy](#)

LLM ecosystem

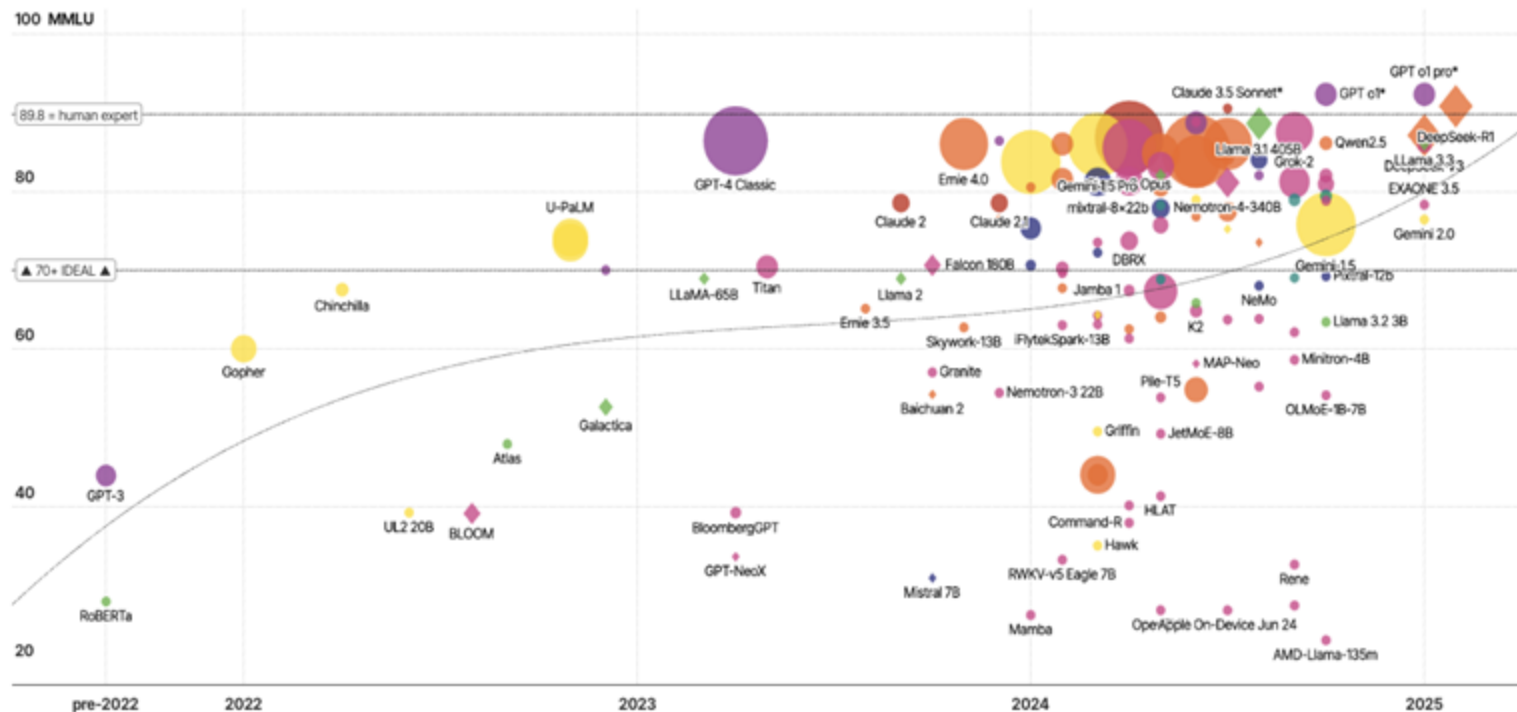
CLICK LEGEND ITEMS TO FILTER

Parameters (Bn) open access

anthropic chinese google meta microsoft mistral openAI other

Q search...

show only: all



* = parameters undisclosed // source: LifeArchitect // data

MADE WITH *VIZ*sweet

ARE YOU SMARTER THAN AN LLM?

The speed of light inside of a nonmagnetic dielectric material with a dielectric constant of 4.0 is

$1.2 \times 10^9 \text{ m/s}$

$3.0 \times 10^8 \text{ m/s}$

$1.5 \times 10^8 \text{ m/s}$

$1.0 \times 10^8 \text{ m/s}$

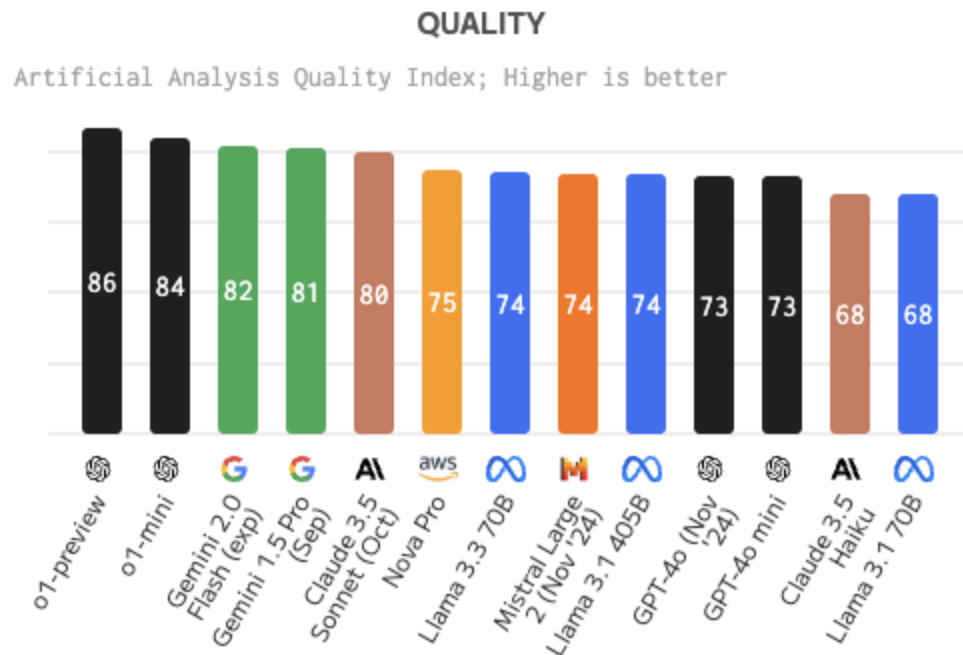


<https://d.erenrich.net/are-you-smarter-than-an-llm/index.html>

Benchmarking

Example leaderboards

- <https://lmarena.ai/?leaderboard>
- <https://artificialanalysis.ai/models>
- <https://scale.com/leaderboard>



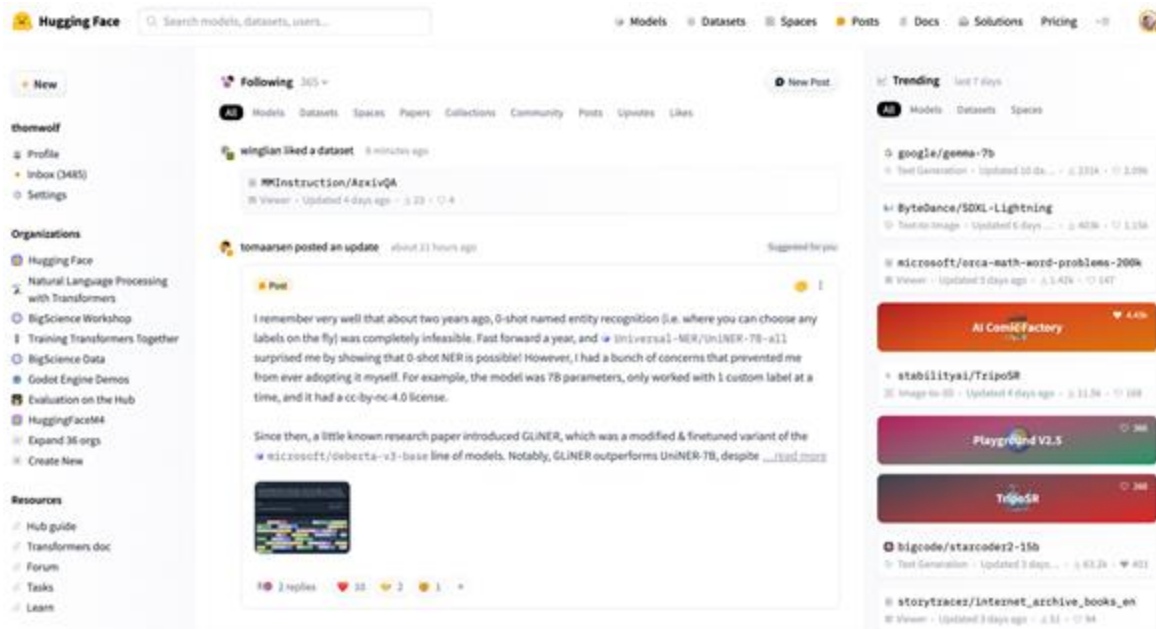
Hugging Face: The home of open ML



Founded In
2016

170
Employees

<https://huggingface.co/>



300K+
stars on Github

500K+
open source models

100K+
public data sets

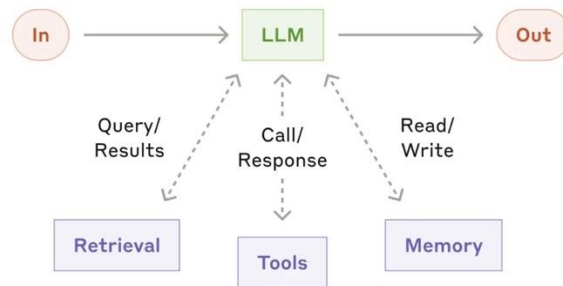
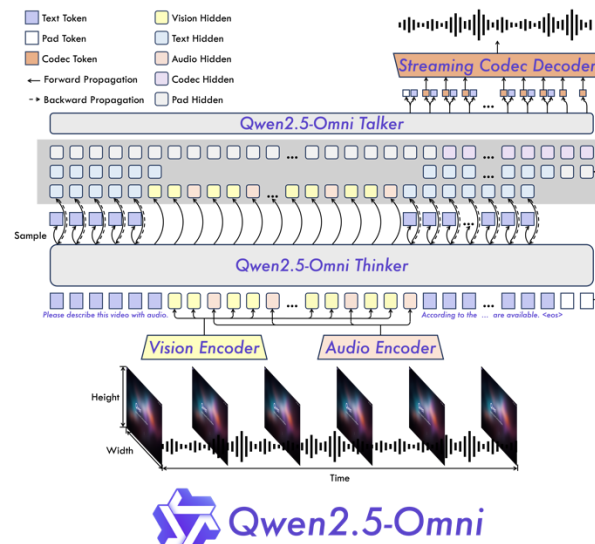
1M+
daily downloads

700K+
daily visitors

30+
Libraries

Deep learning: Trends

1. Multimodality
2. Agents
3. Smaller, faster models
4. More open-source



Deep learning: Challenges

- Lack of theory in designing/training networks
 - Network structures are mostly empirical. Trial and error
- Interpretability
 - Hard to interpret how data is processed and what neurons do. Bias, adversarial examples, approximate reasoning,
- Computational resources: training times, cost, carbon footprint
 - Training of large networks require substantial computational resources
- Deep learning is not all you need: <https://arxiv.org/abs/2106.03253>

To go further

- Online courses:
 - [MIT Introduction to deep learning](#) - 2025
 - [FastAI - Free online course](#)
- YouTube channels: [Andrej Karpathy](#), [3Blue1Brown](#)
- Books:
 - [Dive into Deep Learning, 2022](#)
 - [Build a large language model \(from scratch\), 2024](#)