Course 5: Foundation Models

Summary

Last session

1. Deep Learning Basics
2. Multi-Layer Perceptron
3. Convolutional Neural Networks
4. State-of-the-Art in Deep Learning

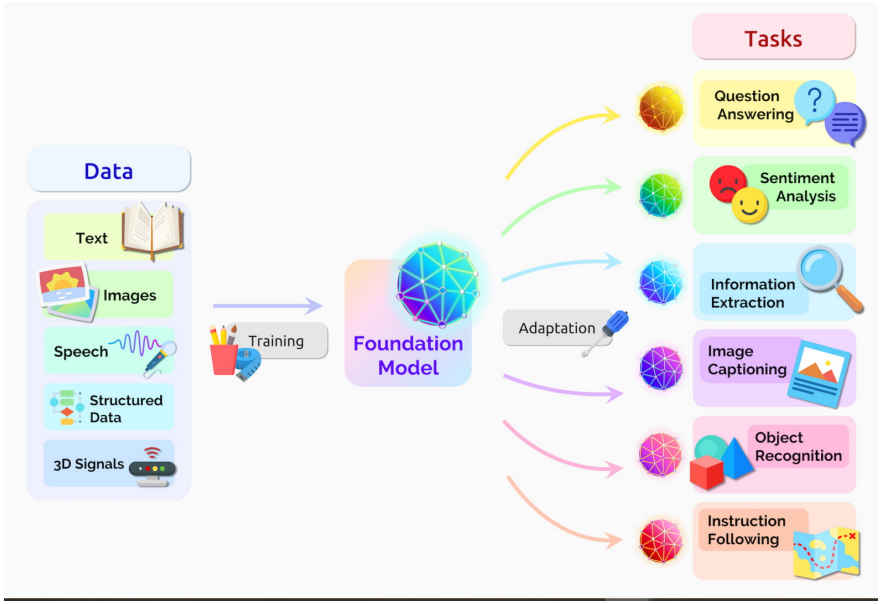
Today's session

- What is a Foundation Model
- Transformers
- Self-Supervised Learning
- Some examples of Foundation Models
- Multi-model foundation models

Outline

What is a Foundation Model?

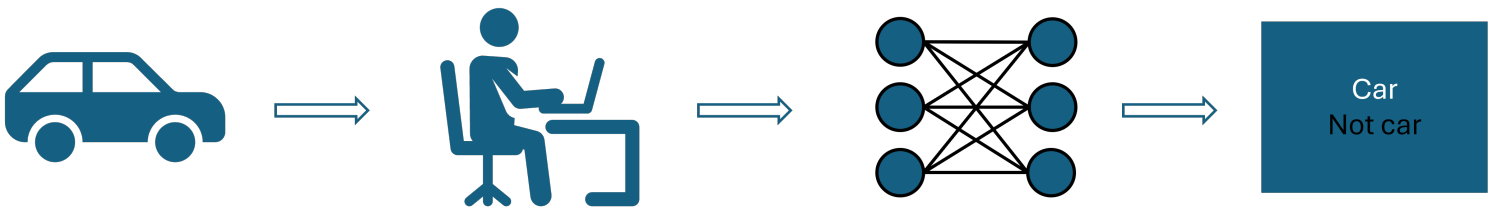
- Trained on internet-scale datasets
- Training task si not straightforward (SSL, pretext tasks)
- Generic feature extractors, Multipurpose
- Generalization is not a problem anymore! All is about **particularization**



https://blogs.nvidia.com/blog/what-are-foundation-models/

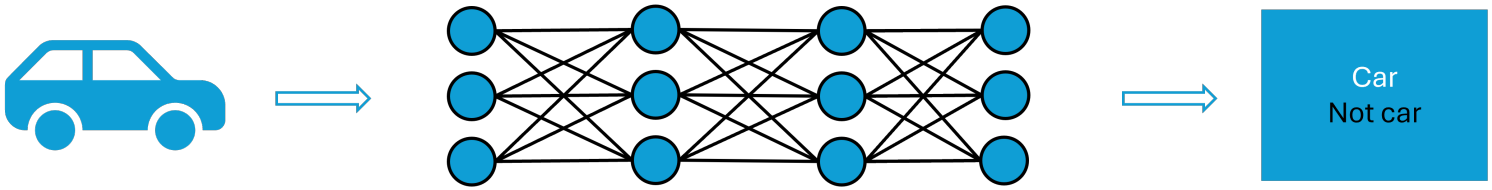
What is a Foundation Model?

Machine Learning Era



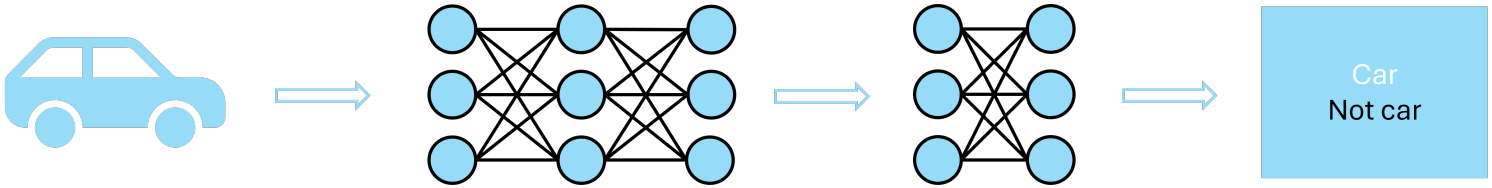
Input → Feature extraction (human-based) → Classification (learned by machine) → Output

Deep Learning Era



Input → Feature extraction + Classification (both learned by same model) → Output

Foundation Models Era



Input → Feature extraction (« Foundation model », used as a **fixed** model) → Classification (learned by machine) → Output

Where do Foundation Models come from?

Two key ingredients:

- The **Transformers** neural network architecture;
- **Self-Supervised learning**, to scale on large (very large) datasets.

Outline

Transformers

Standard architecture nowadays

- No convolution
- Based on *attention*: what should be important for context?
- Used for text, image, audio, ...

Transformers

Standard architecture nowadays

- No convolution
- Based on *attention*: what should be important for context?
- Used for text, image, audio, ...

Transformer block

Based on 3 elements:

- Key
- Query
- Value

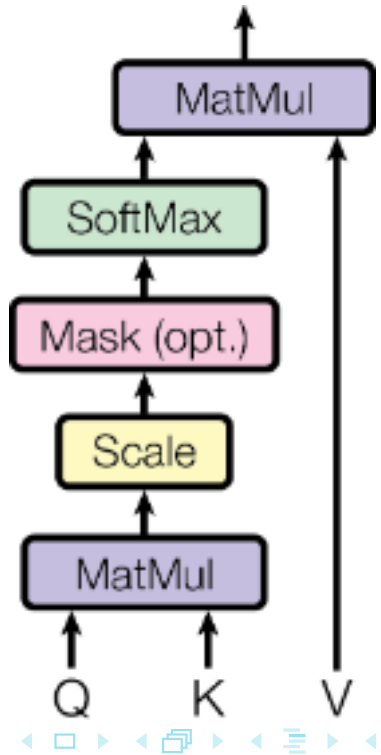
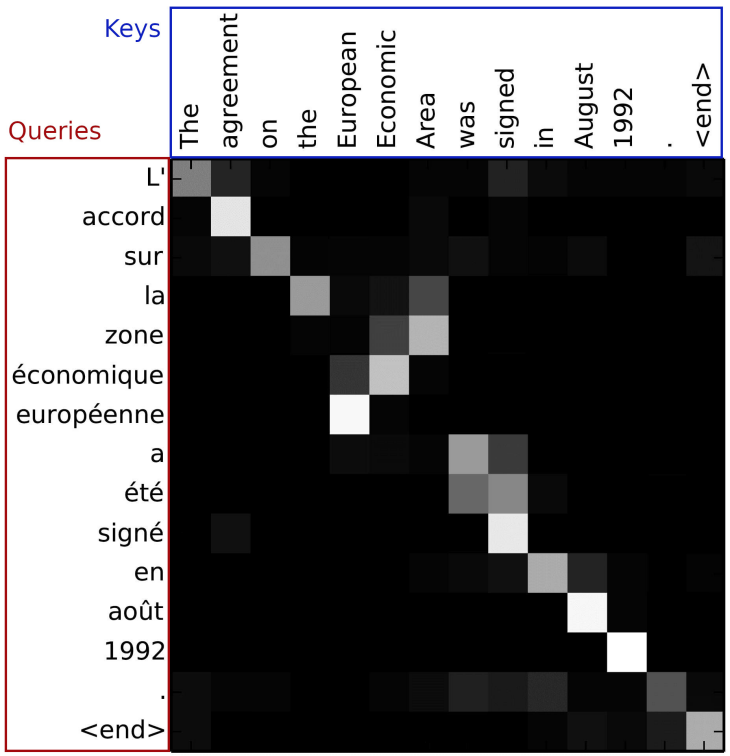


Image source: Vaswani, A. *et al.* (2017). Attention is all you need. Advances in neural information processing systems, 30.

Intuition behind Transformers (1/3)

Attention: Key and Query

- Key: The current word of interest
- Query: All words which may be related

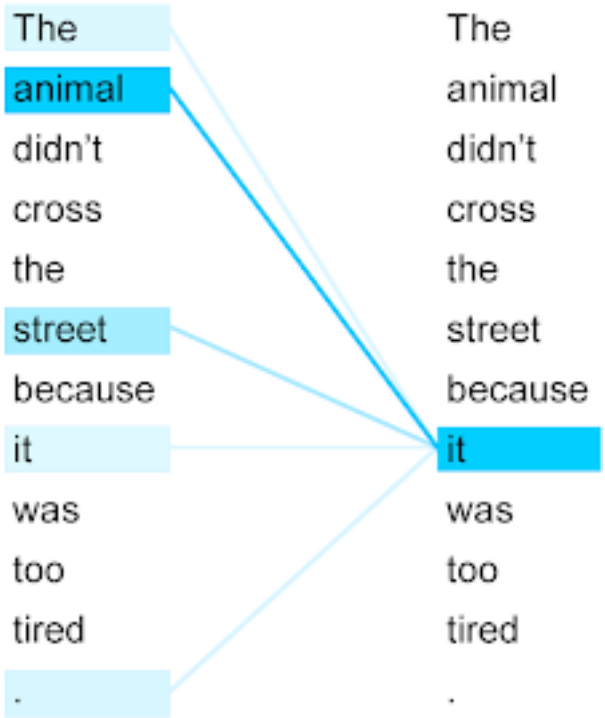


Source: Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Intuition behind Transformers (2/3)

From Attention to Self-Attention

In self-attention, Keys and Queries come from the same text: **context**.

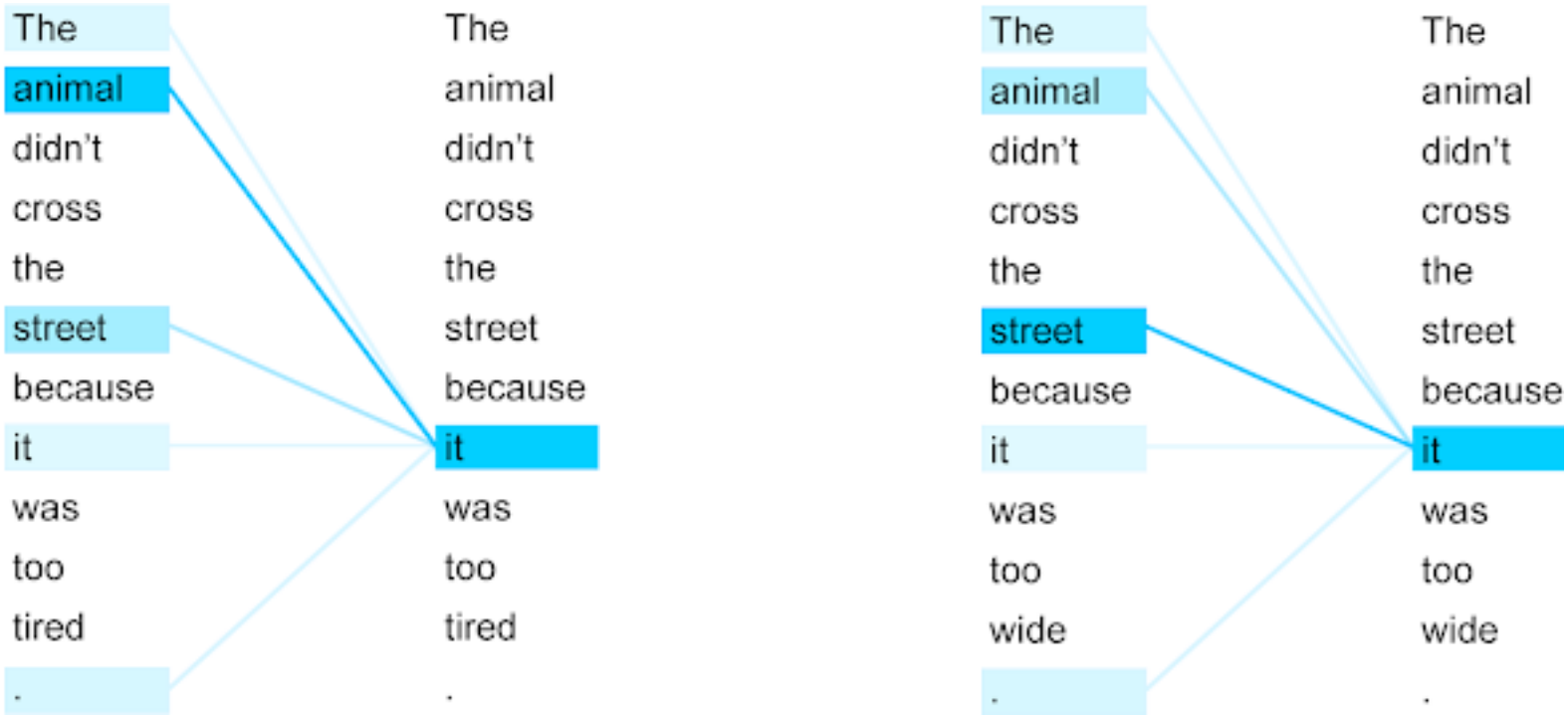


Source:
`https://research.google/blog/transformer-a-novel-neural-network-architecture-for-language-understanding/`

Intuition behind Transformers (2/3)

From Attention to Self-Attention

In self-attention, Keys and Queries come from the same text: **context**.



Source:
`https://research.google/blog/transformer-a-novel-neural-network-architecture-for-language-understanding/`

Intuition behind Transformers (3/3)

Transformer block

- Key and Query: Context
- Value: Integrate the context into the word representation (e.g., the meaning of a word depends on the context).

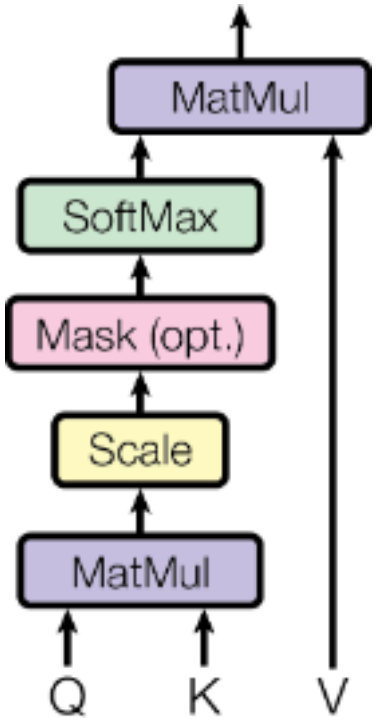
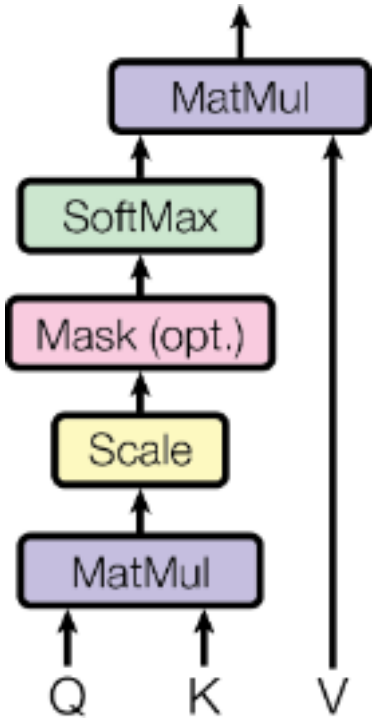


Image source: Vaswani, A. *et al.* (2017). Attention is all you need. Advances in neural information processing systems, 30.

Intuition behind Transformers (3/3)

Transformer block

- Key and Query: Context
- Value: Integrate the context into the word representation (e.g., the meaning of a word depends on the context).



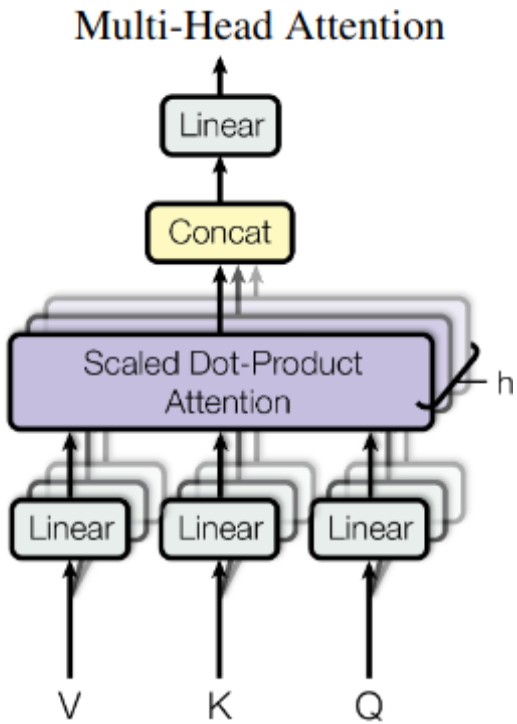
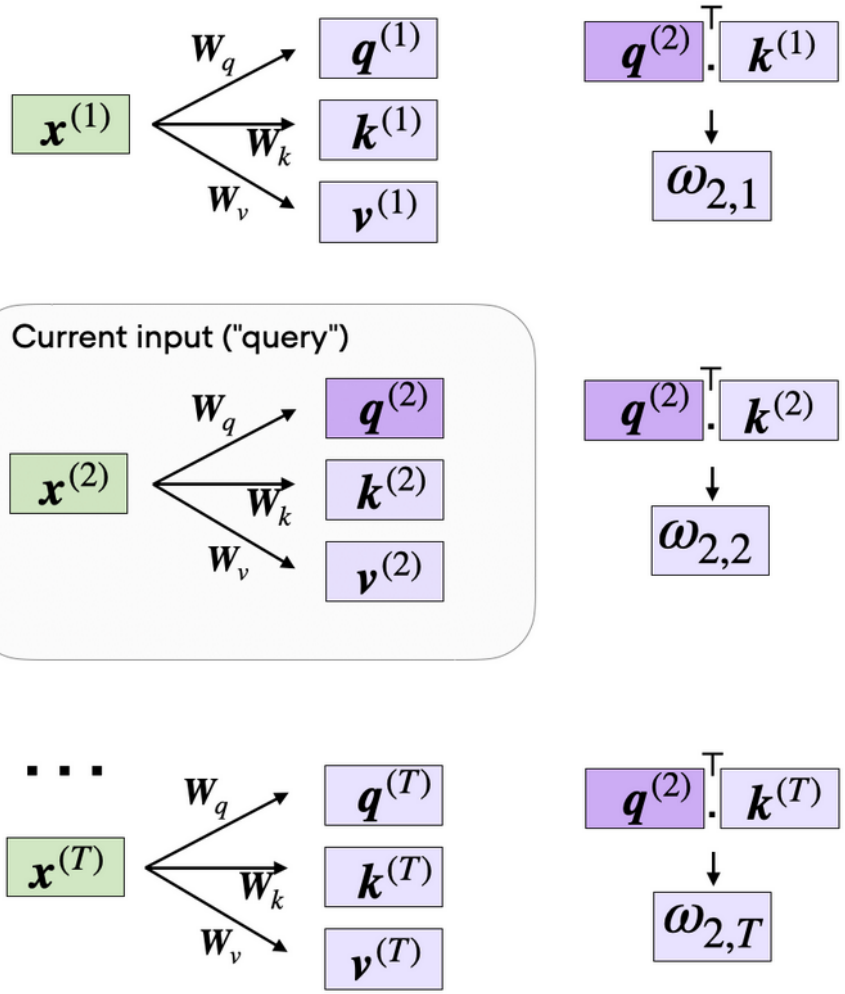


Image source: Vaswani, A. *et al.* (2017). Attention is all you need. Advances in neural information processing systems, 30.

Self-attention, formally (key, query, and value) (1/2)

Self-attention

- inputs are projected by weight matrices $\mathbf{W_i}$ into **q**, **k** and **v** vectors
- attention weights $\omega_{i,j}$ are obtained by dot product (*e.g.*, similarity) of query and keys

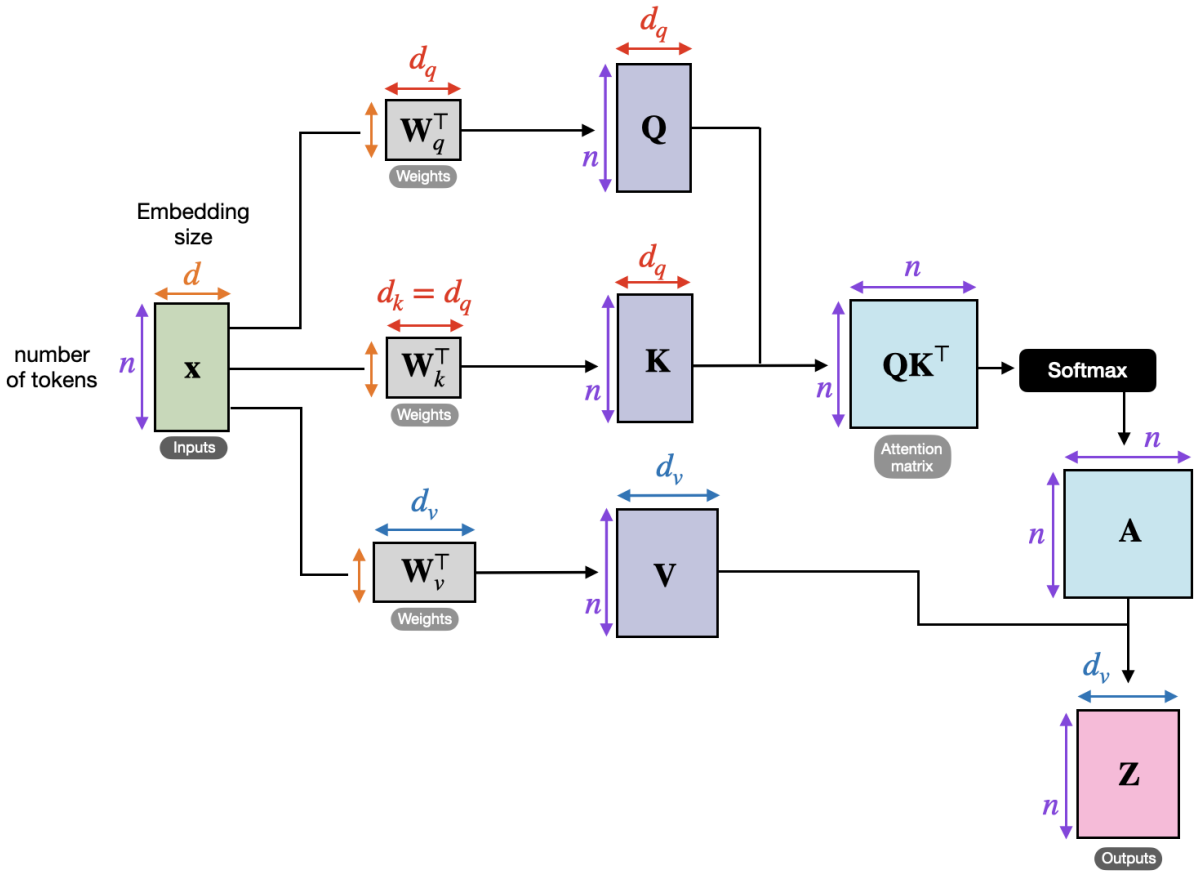


https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html

Self-attention, formally (key, query, and value) (2/2)

Self-attention

- Inputs are projected by weight matrices $\mathbf{W_i}$ into **q**, **k** and **v** vectors
- The output **Z** is an attention-weighted version of the original input **X** with respect to all other elements of the input.



Repeat Transformer blocks: **Deep** model

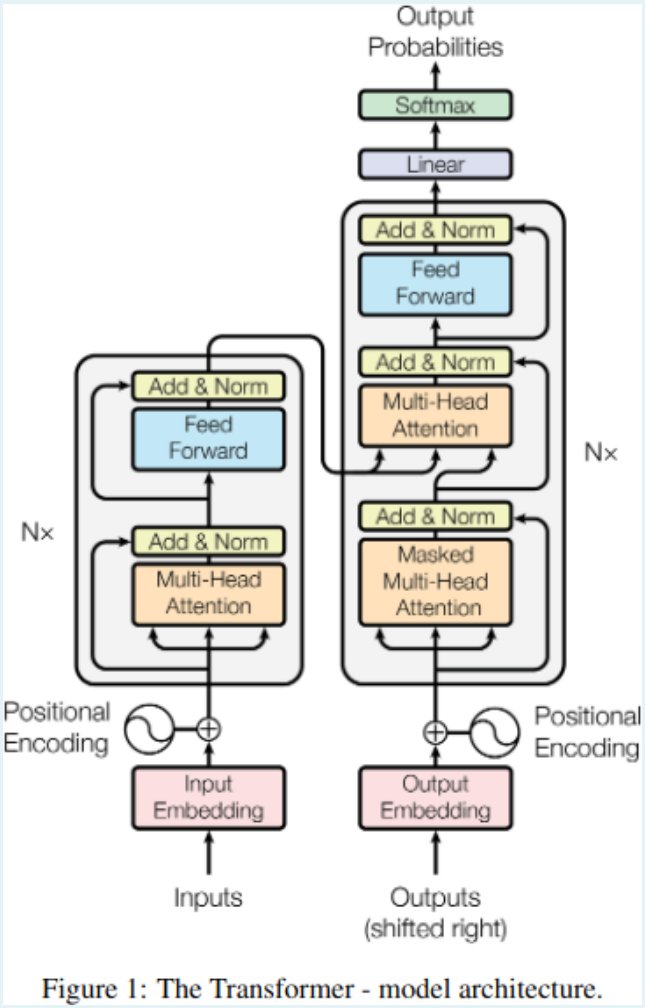


Figure 1: The Transformer - model architecture.

Image source: Vaswani, A. *et al.* (2017). Attention is all you need. Advances in neural information processing systems, 30.

Outline

Self Supervised Learning

The principle

Learning useful representations from data **without relying on labels**. The model *creates* the labels based on the structure of the data through *pretext tasks*.

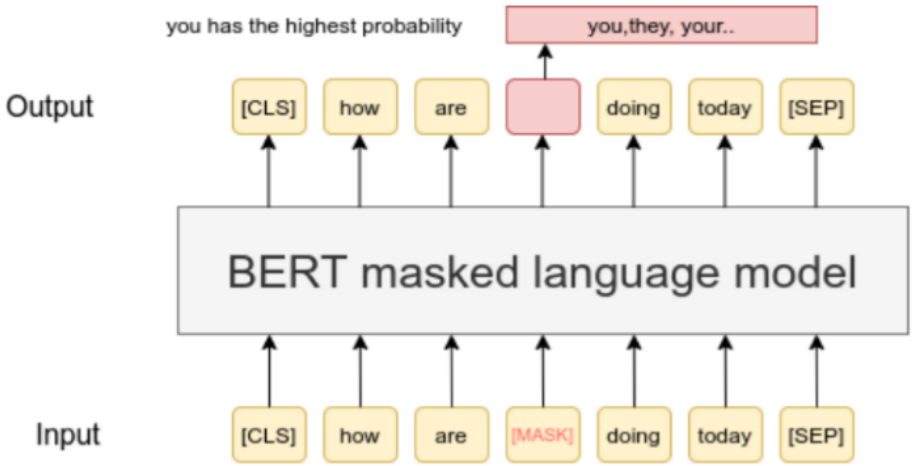
Why it is important

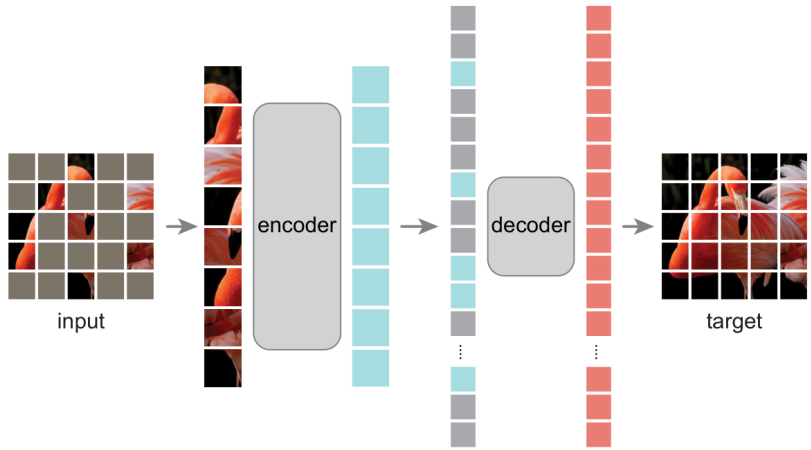
- **Scalability**: Since unlabeled data is much easier to obtain, models can scale to larger datasets and more diverse inputs
- **Robust Representations –> Transferability**: It often leads to more generalizable representations that perform well across multiple tasks compared to models trained on a more limited label space

Self Supervised Learning

Types of SSL approaches

- **Masked Input Modeling**: Predicting missing part of the input





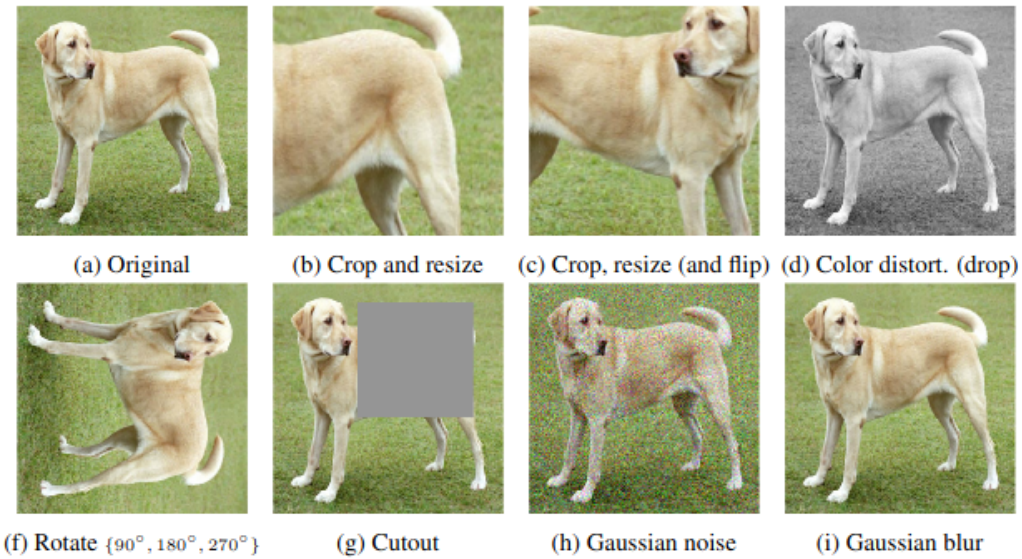
BERT: `https://arxiv.org/pdf/1810.04805`

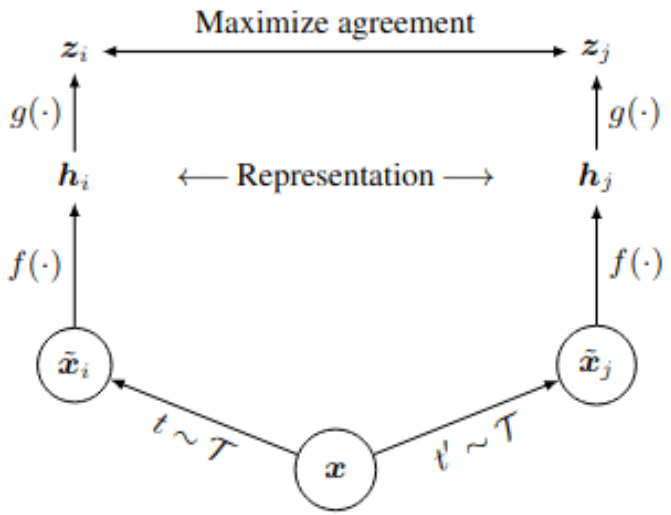
Masked Autoencoders: `https://arxiv.org/pdf/2111.06377`

Self Supervised Learning

Types of SSL approaches

- **Masked Input Modeling**: Predicting missing part of the input
- **Contrastive Learning**: Pulling together similar representations and pushing apart dissimilar ones



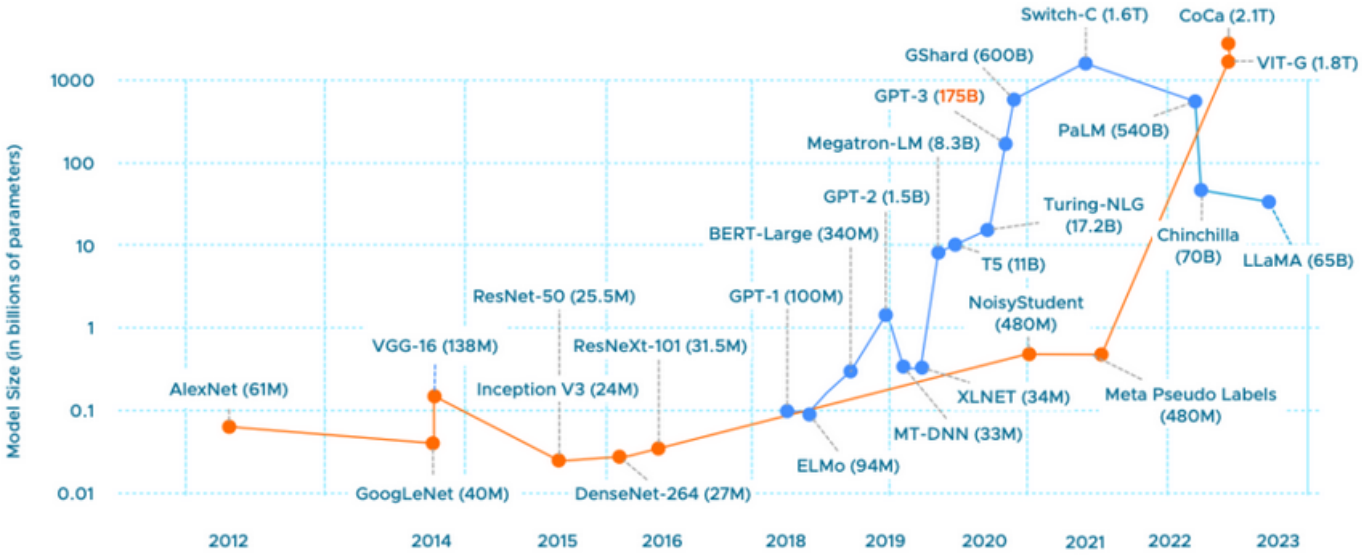


SimCLR: `https://arxiv.org/pdf/2002.05709`, *BYOL:* `https://arxiv.org/pdf/2006.07733`

Outline

How Big Are We Now?

- Increasing model size is a proxy for increasing performance
- Data is as important as scaling model size! For 2x model size, data should also be 2x
- Datasets are growing very rapidly (0.1-0.2 orders of magnitude per year before 2015, now higher)



Tutorial on data efficiency: `https://baharanm.github.io/assets/pdf/ICML24_tutorial_DataEfficient.pdf`
Training Compute-Optimal Large Language Models: `https://arxiv.org/pdf/2203.15556`
Scaling Laws for Neural Language Models: `https://arxiv.org/pdf/2001.08361`

Some examples: Foundation Models for Text

Model	Provider	Open	Release	Params	Data
BERT	Google	Yes	Oct-2018	345M	3.3M
PaLM	Google	No	Apr-2022	540B	780B
GPT-4	OpenAI	No	Mar-2023	-	300B
GPT-3.5	OpenAI	No	Mar-2023	175B	-
Llama1	Meta AI	Yes	Feb-2023	7/13/33/65B	1 T/1.4T
Llama2	Meta AI	Yes	Jul-2023	7/13/33/70B	2T
Mixtral	MistralAI	Yes	Dec-2023	8 x 7B	-
Phi-2	Microsoft	No	Dec-2023	2.7B	-

Large Language Models

- Transformers trained (SSL) on predicting masked *tokens* and next sentence
- Finetuning (Supervised) on instruction data (prompt + response)
- Reinforcement Learning with Human Feedback

Tutorial on foundation models for text:
`https://docs.google.com/presentation/d/1nfV9QiNV2tbHsw9GeP7e96az-oL_C_rymTSE6V6zBos/edit?usp=sharing`
Llama Paper: `https://arxiv.org/abs/2302.13971`

Some examples: Foundation Models for Vision

DINOv2

- Vision transformers trained (SSL) on large curated data
- Produces all-purpose visual features (classification, segmentation, depth estimation..)

SAM

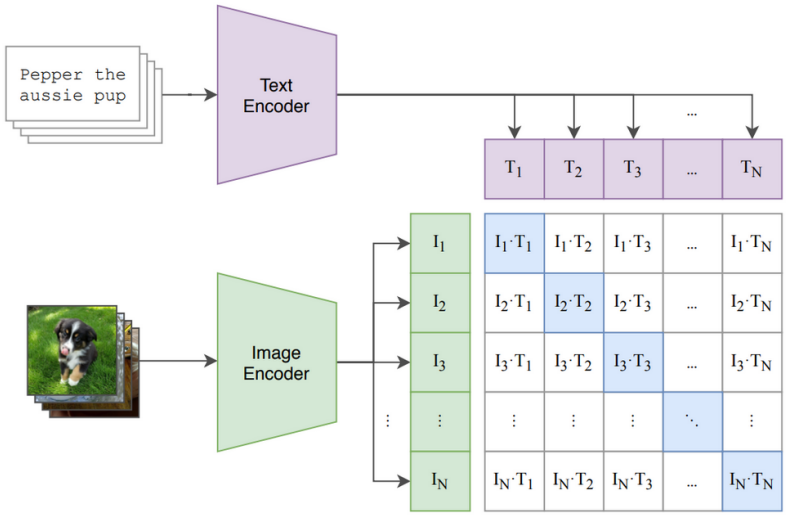
- Image and text Encoders (SSL pretrained Transformers) trained on 1B masks and 1M images
- Segment any object, promptable segmentation

DINOv2: `https://arxiv.org/abs/2304.07193`, *SAM:* `https://segment-anything.com/`

Outline

What is a Multimodal Foundation Model?

- Trained on internet-scale multimodal dataset
- Training task si not straightforward (SSL, contrastive pretext tasks)
- Generic feature extractors, Multipurpose
- Generalization is not a problem anymore! All is about **particularization**



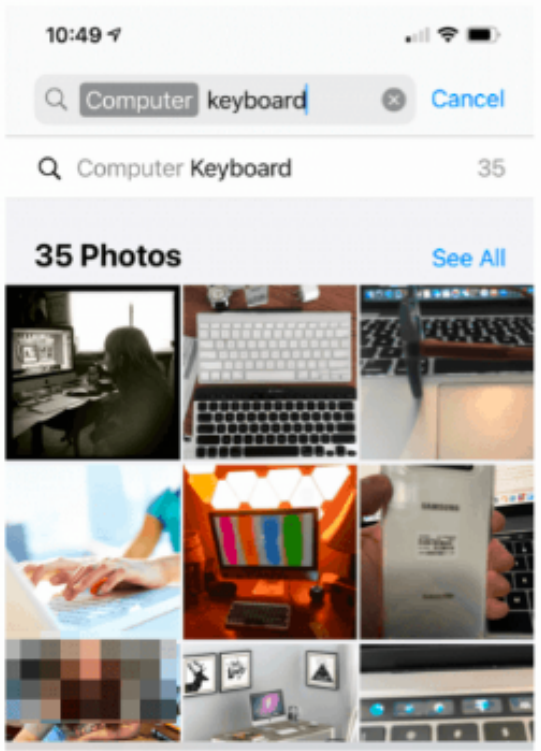
CLIP: `https://arxiv.org/abs/2103.00020v1`

New multimodal tasks

- Text to Image Generation

a teddy bear on a skateboard in times square

- Image Retrieval



Self Supervised Contrastive Learning

Types of SSL approaches

- **Masked Input Modeling**: Predicting missing part of the input (not applicable to multimodal)
- **Contrastive Learning**: Pulling together representations of the same class in different modalities

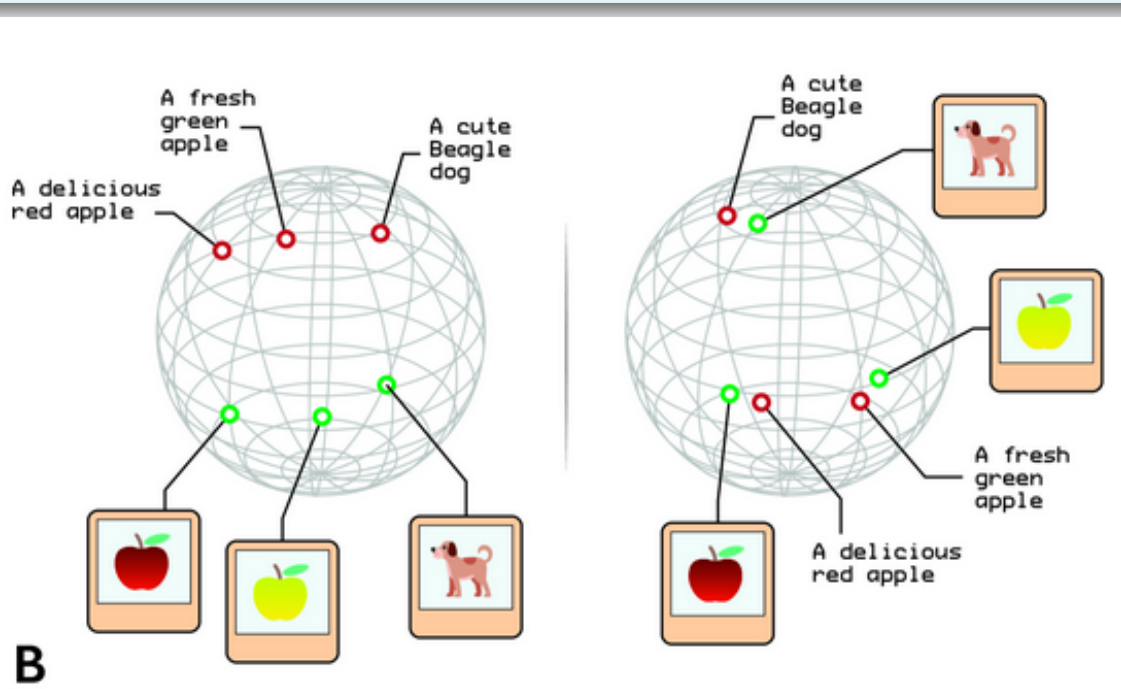


Image from: `https://arxiv.org/html/2406.17639v1`

CLIP: Contrastive Language-Image Pre-Training

Main Ingredients

- Pretrained on a dataset of internet image-text pairs
- Two separate encoders: one for images (Vision Transformer) and another for text (Large Language Model).
- Contrastive Learning: The model learns by maximizing the similarity between image and text embeddings for matching pairs

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.

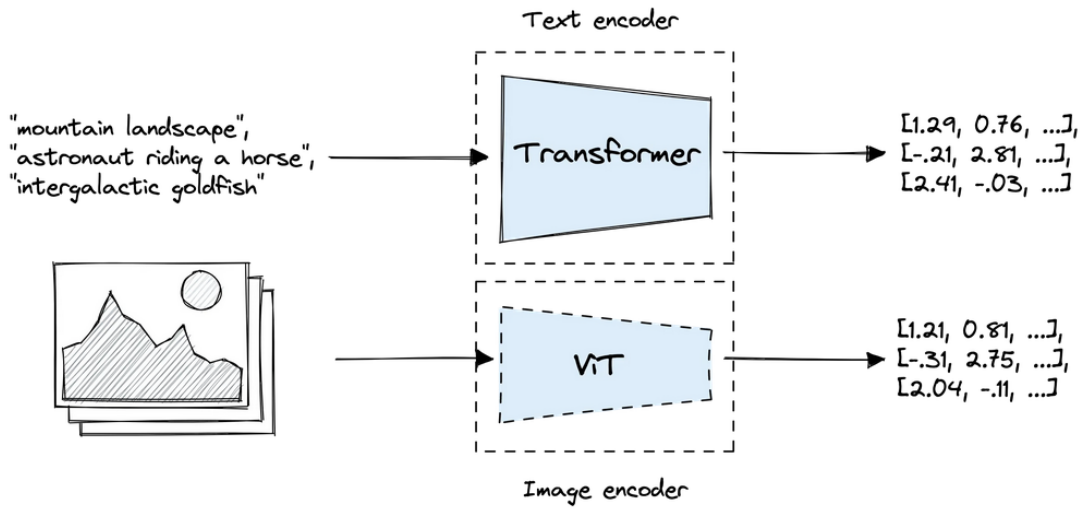
Zero-Shot CLIP: https://www.pinecone.io/learn/series/image-search/zero-shot-image-classification-clip/,

Tutorial by Yassir Bendou:
https://github.com/brain-bzh/clip/tree/71ff8784d9c37ed279e660a77aede0ffeab69515?tab=readme-ov-file

CLIP: Contrastive Language-Image Pre-Training

Main Ingredients

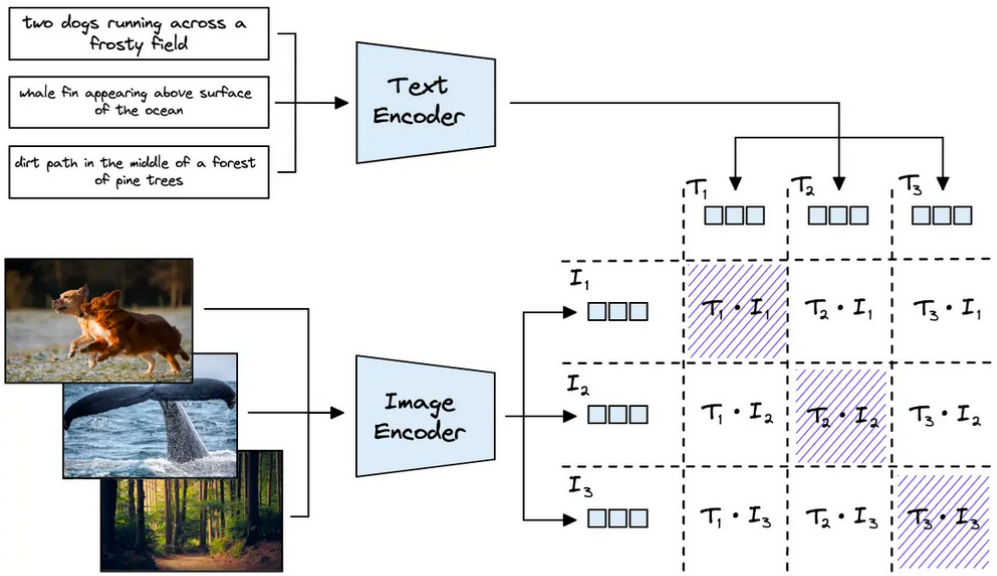
- Pretrained on a dataset of internet image-text pairs
- Two separate encoders: one for images (Vision Transformer) and another for text (Large Language Model).
- Contrastive Learning: The model learns by maximizing the similarity between image and text embeddings for matching pairs



CLIP: Contrastive Language-Image Pre-Training

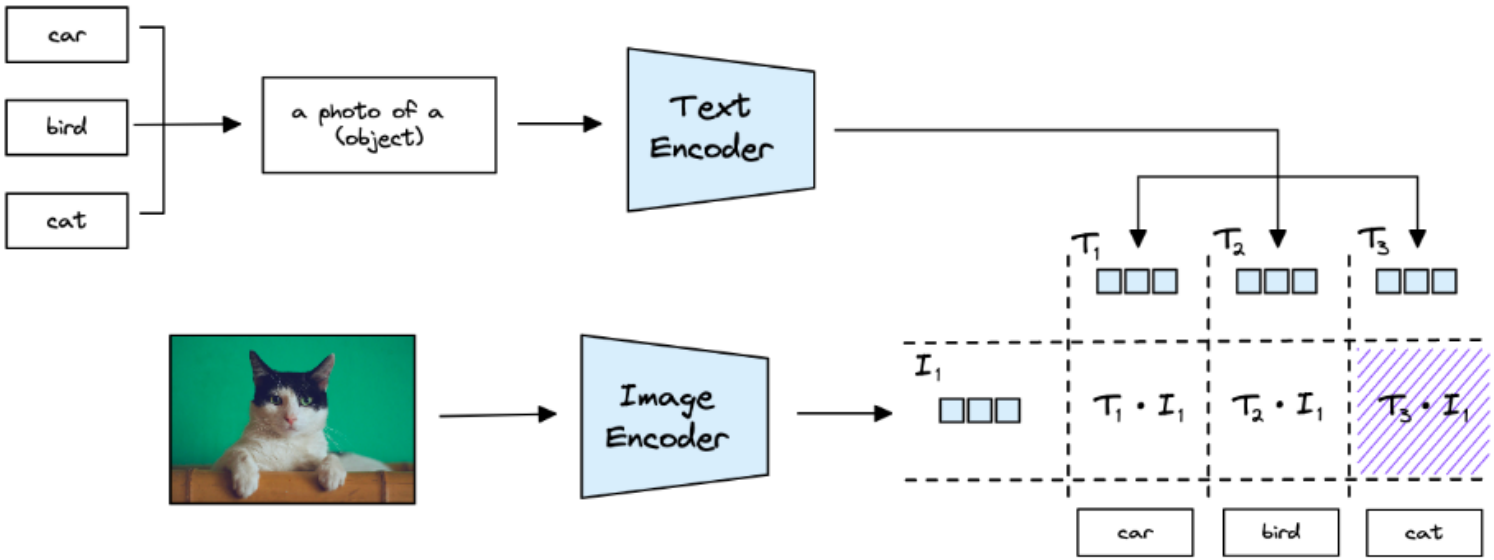
Main Ingredients

- Pretrained on a dataset of internet image-text pairs
- Two separate encoders: one for images (Vision Transformer) and another for text (Large Language Model).
- Contrastive Learning: The model learns by maximizing the similarity between image and text embeddings for matching pairs



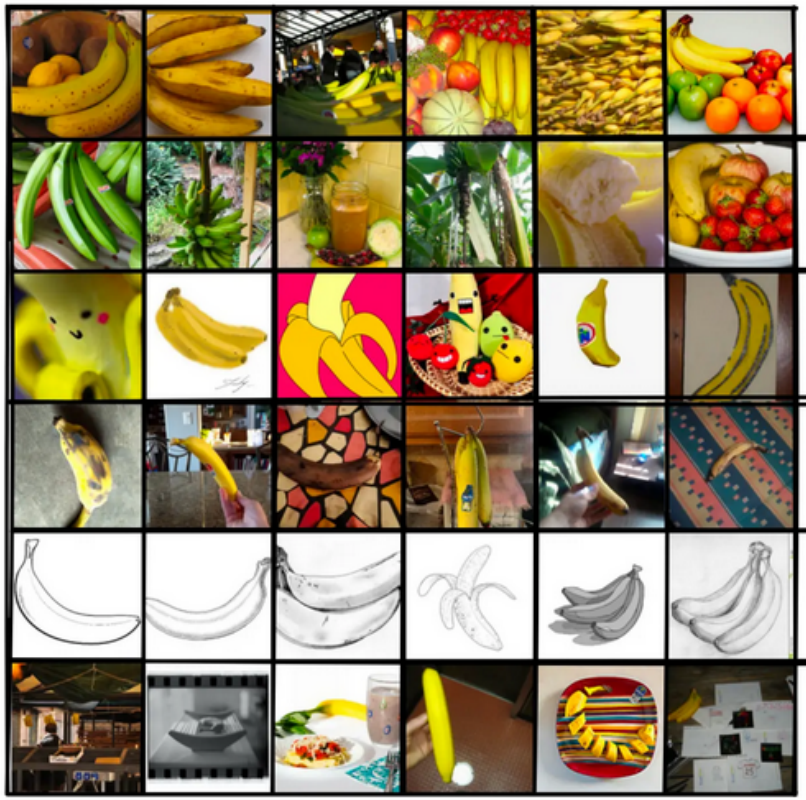
CLIP Performances

- Zero Shot Classification
- Image Retrieval
- ...



CLIP Performances

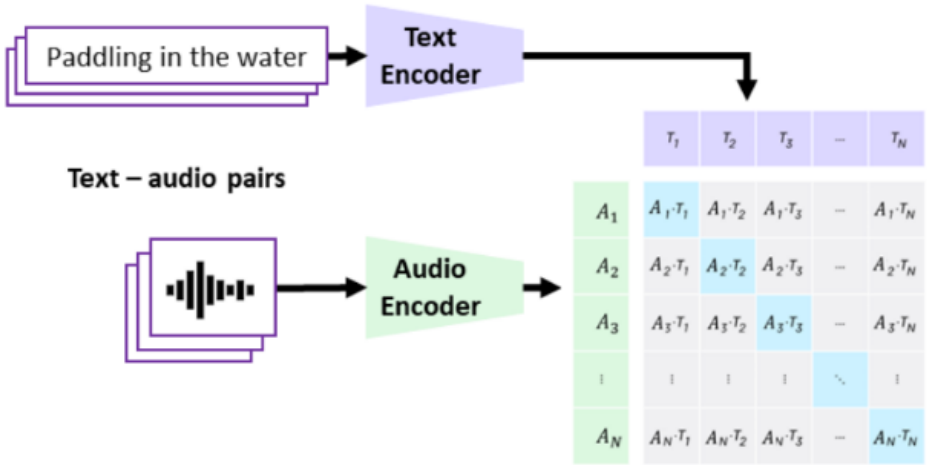
Dataset Examples	ImageNet ResNet101	Zero-shot CLIP	Δ Score
ImageNet	76.2	76.2	0%
ImageNetV2	64.3	70.1	+5.8%
ImageNet-R	37.7	88.9	+51.2%
ObjectNet	32.6	72.3	+39.7%
ImageNet Sketch	25.2	60.2	+35.0%
ImageNet-A	2.7	77.1	+74.4%



CLAP: Contrastive Language-Audio Pretraining

Main Ingredients

- Pretrained on 128k audio and text pairs
- Two separate encoders: one for audio (*e.g.*, CNN14) and another for text (*e.g.*, BERT).
- Contrastive Learning: The model learns by maximizing the similarity between image and text embeddings for matching pairs



CLAP: `https://arxiv.org/abs/2206.04769`

CLAP: Contrastive Language-Audio Pretraining

Main Ingredients

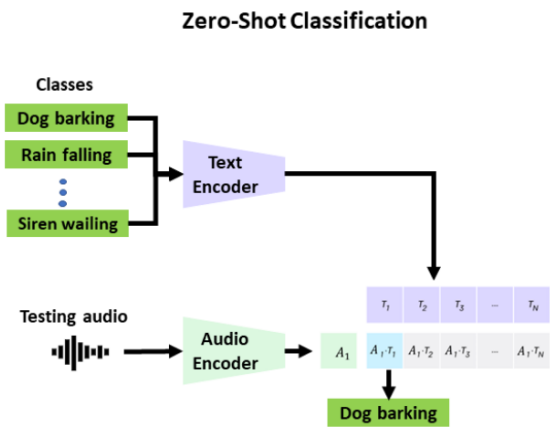
- Pretrained on 128k audio and text pairs
- Two separate encoders: one for audio (*e.g.*, CNN14) and another for text (*e.g.*, BERT).
- Contrastive Learning: The model learns by maximizing the similarity between image and text embeddings for matching pairs

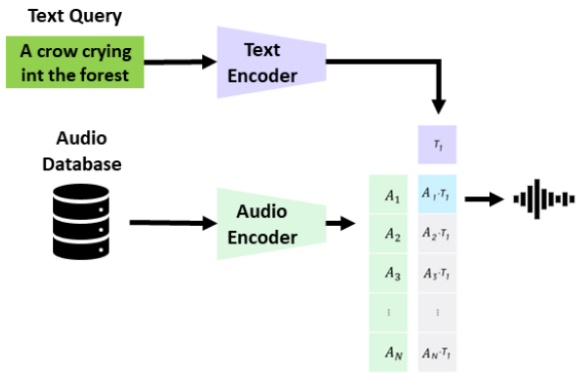
Different CLAP models

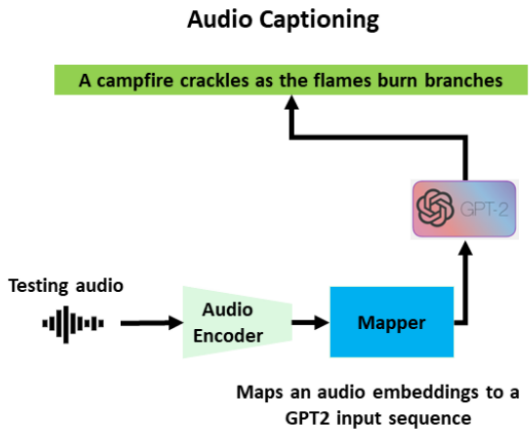
Different versions have been pretrained, all with a similar contrastive approach

- MS-CLAP , Elizalde et al. 2022
- LAION-CLAP, Wu et al. 2023
- MS-CLAP with captioning, Elizalde et al. 2023
- WavCaps, Meil et al. 2023

CLAP Performances



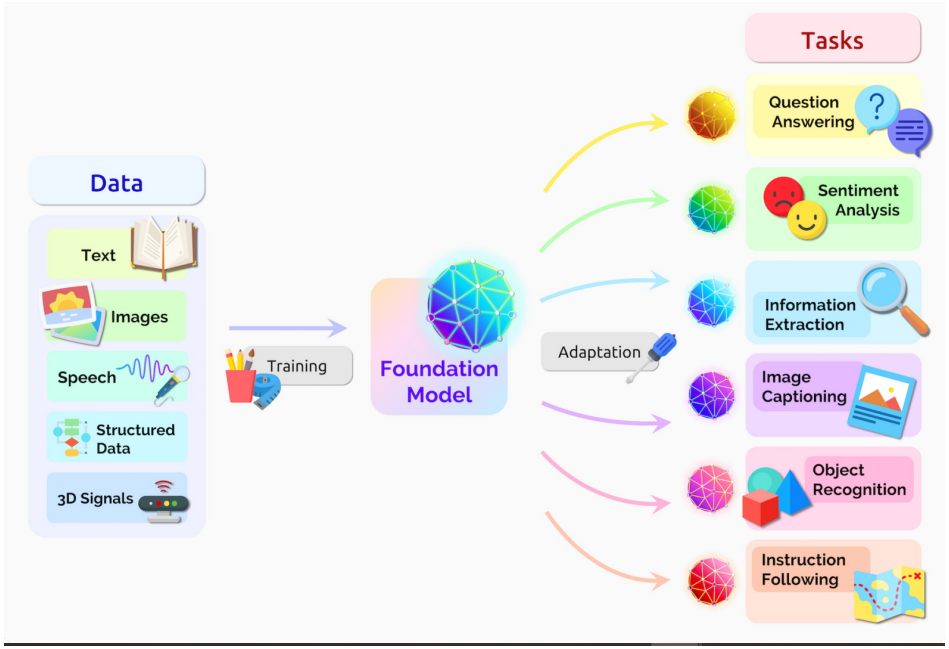




Outline

Foundation Models

- Model trained on an Internet scale dataset
- Self-Supervised training: pretext tasks
- Particularization vs Generalization



Source: `https://blogs.nvidia.com/blog/what-are-foundation-models/`

Adapt ("Customize") a Foundation Model

Next class!

- Foundation models are often fine-tuned after pre-training to adapt them to a specific task
- The challenge becomes particularization (as opposed to generalization)
- Different techniques depending on the objective of the adaptation:
  1. Add specific information to the model knowledge (*e.g.*, Retrieval augmented generation RAG)
  2. Change *behavior* of the model with fine-tuning (*e.g.*, parameter efficient fine tuning with LORA)
  3. Improve Foundation Model safety and helpfulness (*e.g.*, Reinforcement Learning based un Human Feedback for LLMs)

RAG: `https://blogs.nvidia.com/blog/what-are-foundation-models/`
LoRA Paper: `https://arxiv.org/pdf/2106.09685.pdf`
Various tutorials on RAG and fine-tuning: `https://github.com/facebookresearch/llama-recipes/tree/main`

Lab Session 5

Generate the embeddings you worked with in Lab 2 and 3 using pretrained Foundation Models

- Load the specific modality dataset
- Preprocess the raw data if needed
- Use Hugging Face to load the pretrained Foundation Model
- Generate the embeddings and test them