Universidade de São Paulo
Escola Politécnica - Engenharia de Computação e Sistemas Digitais

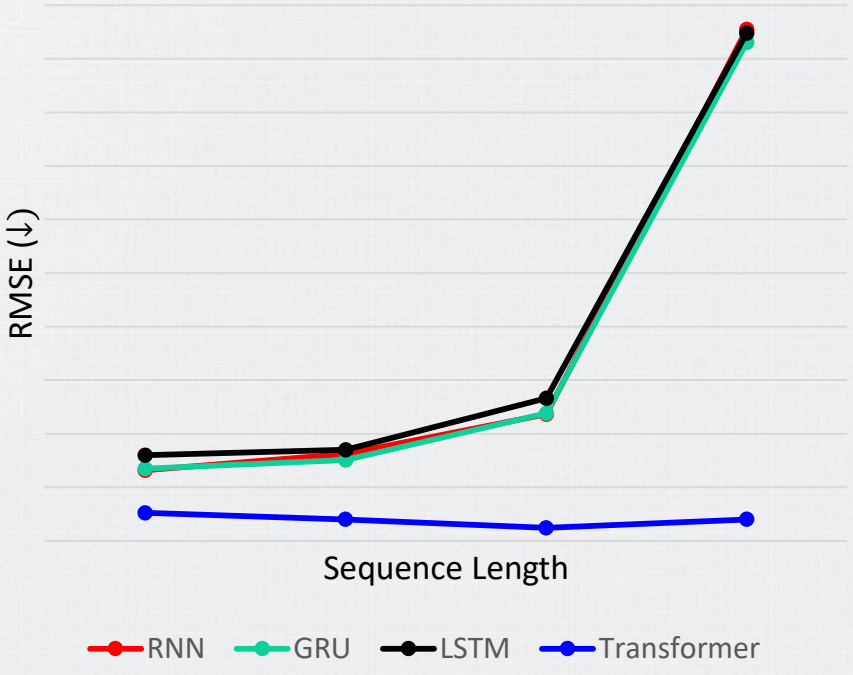
Transformers, Modern MLP Architectures and Foundation Models

Prof. Artur Jordão

Introduction

Transformers

- The transformer (Vaswani et al., 2017) architecture has revolutionized various fields of artificial intelligence, such as natural language understanding and computer vision
  - These models express long sequences of data better than recurrent models



Introduction

Transformers

- Transformers-like models have pushed the state-of-the-art in different tasks

- Surprisingly, these models perform on par or better than convolutional networks (Dosovitskiy et al., 2021), which have been the central paradigm in deep learning for computer vision tasks

Model	CIFAR-10	ImageNet
ResNet	93.57	80.62
NASNet	97.60	74.00
Transformer	99.50	88.55

Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. International Conference on Learning Representations (ICLR), 2021

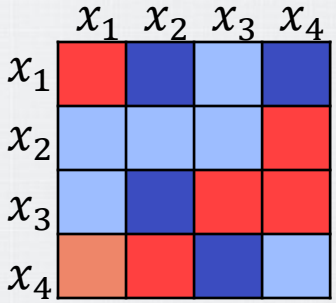
Transformer Architecture

Self-Attention (SA)

Transformer Architecture

- At the heart of Transform lies the self-attention mechanism
  - Ability to learn **pairwise relationships** between all tokens/features in a given sequence
  - Attention to itself

- A self-attention, $SA(\cdot)$, block takes n inputs $x_1, x_2, \ldots x_n$ and returns n output vectors of the same size
  - Each x_i has dimensions $d \times 1$



Values

Transformer Architecture

- A set of values is computed for each input in terms of $v_i = W^v x_i + b^v$
  - W^v indicates (learnable) weights
  - b^v indicates the bias

- Note that the weights $W^v \in \mathbb{R}^{d \times d}$ and biases $b^v \in \mathbb{R}^d$ **are shared across all inputs** $x_i \in \mathbb{R}^d$

Queries and Keys

Transformer Architecture

- Queries (q) and keys (k) are linear transformations
  - $q_i = W^q x_i + b^q$
  - $k_i = W^k x_i + b^k$

- The queries and keys must have the same dimensions

Attention

Transformer Architecture

- The scalar weight $a[x_i, x_n]$ is **the attention that the nth output pays to input x_i**
- The n weights $a[, x_n]$ are non-negative and sum to one
- The attention weights $a[x_i, x_n]$ combine the values from different inputs
- $a[x_i, x_n] = Softmax(k_n^T q_n)$

Matrix Form

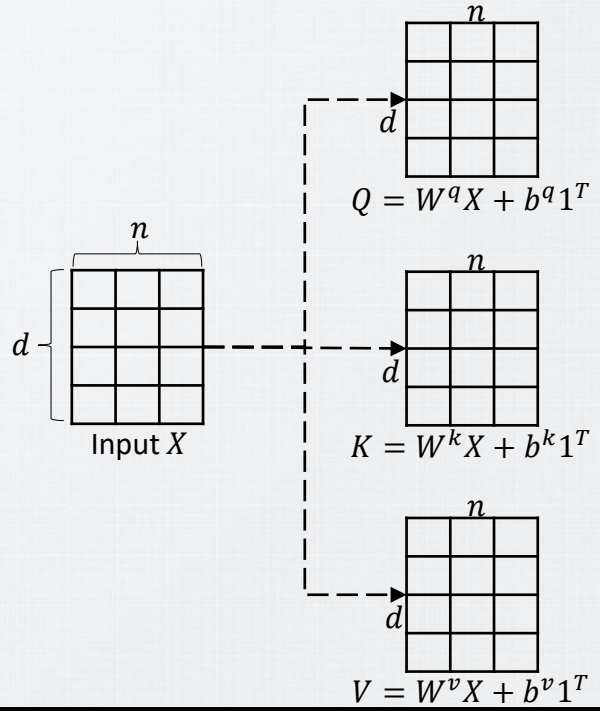
Transformer Architecture

- We can put Values, Key and Queries into a compact form
  - Assuming that the n inputs x_n form the columns of the $d \times n$ matrix X

- $V = W^v X + b^v 1^T$
  - Values

- $Q = W^q X + b^q 1^T$
  - Queries

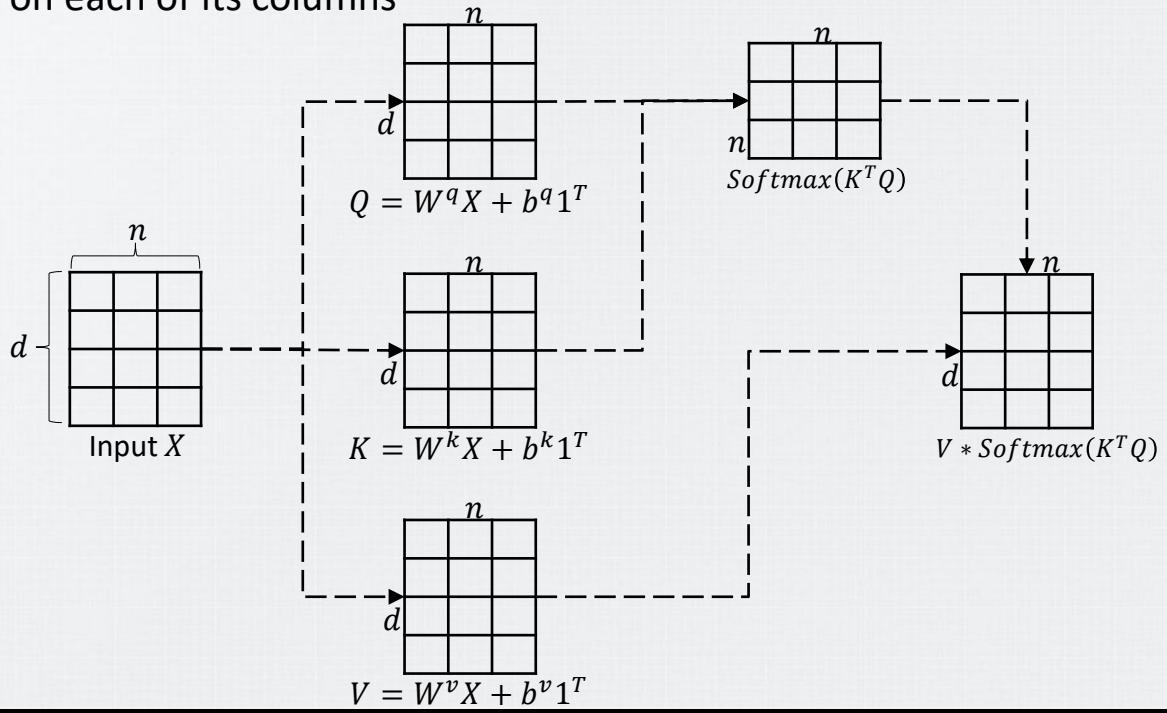
- $K = W^k X + b^k 1^T$
  - Keys



Self-Attention (SA)

Transformer Architecture

- $SA = V * Softmax(K^T Q)$
  - In this matrix form, the $Softmax(\cdot)$ function takes a matrix and performs the softmax operation independently on each of its columns



Scaled Dot Product Self-Attention

Transformer Architecture

- The dot products in the attention computation can have large magnitudes
  - It moves the arguments to the softmax function into a region where the largest value completely dominates
  - Small changes to the inputs to the softmax function now have little effect on the output (i.e., the gradients are very small)

- We can scale the dot products by the square root of the dimension d of the queries and keys
  - $SA = V * Softmax\left(\frac{K^T Q}{\sqrt{d}}\right)$
  - This is known as **scaled dot product self-attention**

Multi-Head Attention

Transformer Architecture

- To enable the model to learn different representations, the architecture applies the self-attention mechanism multiple (h) times for the same input X
  - **We name the hth self-attention as head**

- Each head has its own set of weights (and biases)
  - W^{vh}, W^{qv}, W^{kv}

- The self-attention to the head h is: $V_h * Softmax\left(\frac{K_h^T Q_h}{\sqrt{d}}\right)$

- Finally, we concatenate and project onto a learnable matrix W^o
  - $Multihead(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W^o$

Complexity

Transformer Architecture

- Advantages of attention layers over recurrent and convolutional networks
  - Lower computational complexity
  - Higher connectivity: especially useful for learning long-term dependencies in sequences

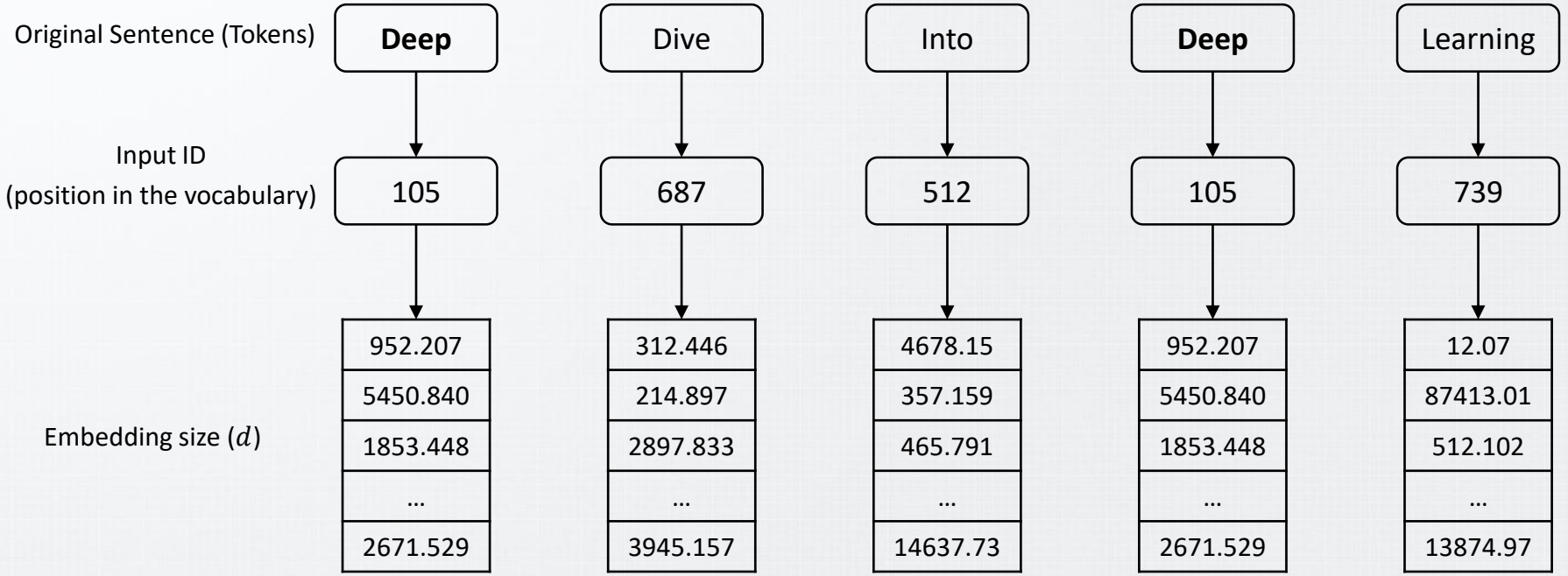
- Definitions
  - n, sequence length
  - k, kernel size (convolutional layers)
  - d, representation dimension

Layer Type	Complexity per Layer
Recurrent	$O(n \times d^2)$
Convolutional	$O(k \times n \times d^2)$
Self-Attention	$O(n^2 \times d)$

Vaswani et al. *Attention Is All You Need*. Neural Information Processing Systems (NeurIPS), 2017

Embedding

Transformer Architecture



Positional Encoding

Transformer Architecture

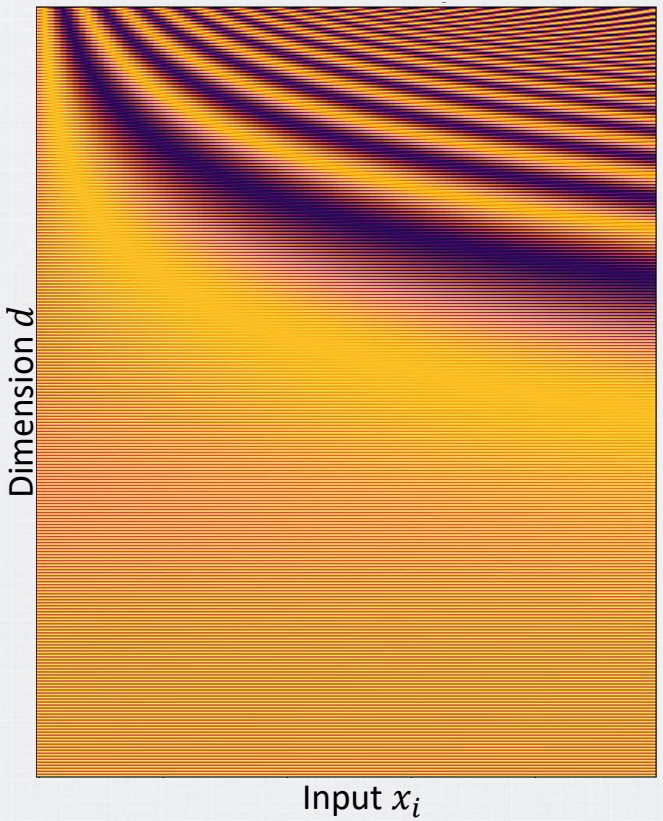
- The previous components we have seen so far are unable to capture **relative positions**: they contain no recurrence or convolution
  - The computation is the same regardless of the order of the inputs x_n
  - It is equivariant with respect to input permutations

- The positional encoding injects some information about the **relative or absolute position** of the input in the sequence
  - This component plays a role in Transformers-like models

Positional Encoding

Transformer Architecture

- Sinusoidal pattern (sine and cosine functions)
  - $PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$
  - $PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$
  - pos is the index in the sequence and i is the ith position in d ($i \in \{0, 1, 2 \dots (|d| - 1)\}$)



Positional Encoding

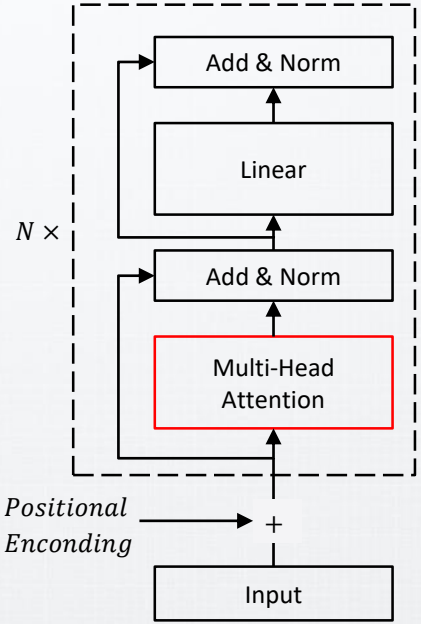
Transformer Architecture

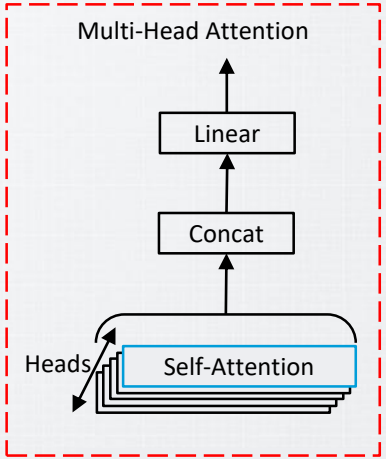
- Sinusoidal pattern (sine and cosine functions)
  - $PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$
  - $PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$
  - pos is the index in the sequence and i is the ith position in d ($i \in \{0, 1, 2 \dots (|d| - 1)\}$)

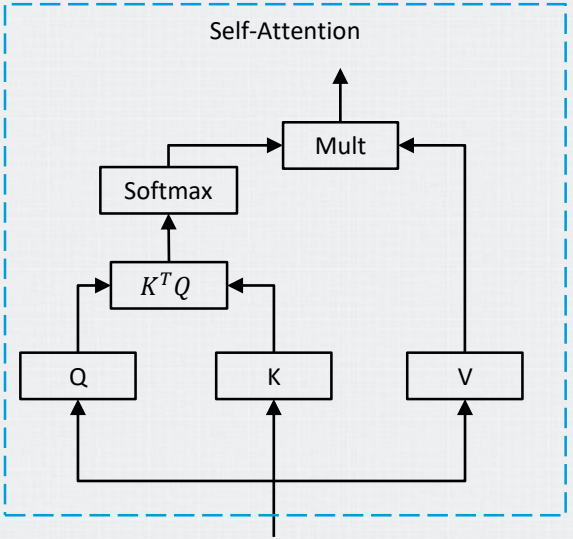
Sequence	pos	$i = 0$	$i = 1$	$i = 2$	$i = 3$
I	0	$PE_{(00)} = \sin(0) = 0$	$PE_{01} = \cos(0) = 1$	$PE_{(02)} = \sin(0) = 0$	$PE_{(03)} = cos(0) = 1$
Am	1	$PE_{10} = \sin(1) = 0.84$	$PE_{11} = \cos(1) = 0.54$	$PE_{(12)} = \sin(0.01) = 0.009$	$PE_{(13)} = cos(0.01) = 0.9999$
A	2	$PE_{20} = \sin(2) = 0.909$	$PE_{21} = \cos(2) = -0.416$	$PE_{(22)} = \sin(0.02) = 0.019$	$PE_{(23)} = cos(0.02) = 0.9998$
Robot	3	$PE_{30} = \sin(3) = 0.14$	$PE_{31} = \cos(3) = -0.989$	$PE_{32} = \sin(0.03) = 0.0299$	$PE_{33} = \cos(0.03) = 0.9995$

Overall Architecture

Transformer Architecture





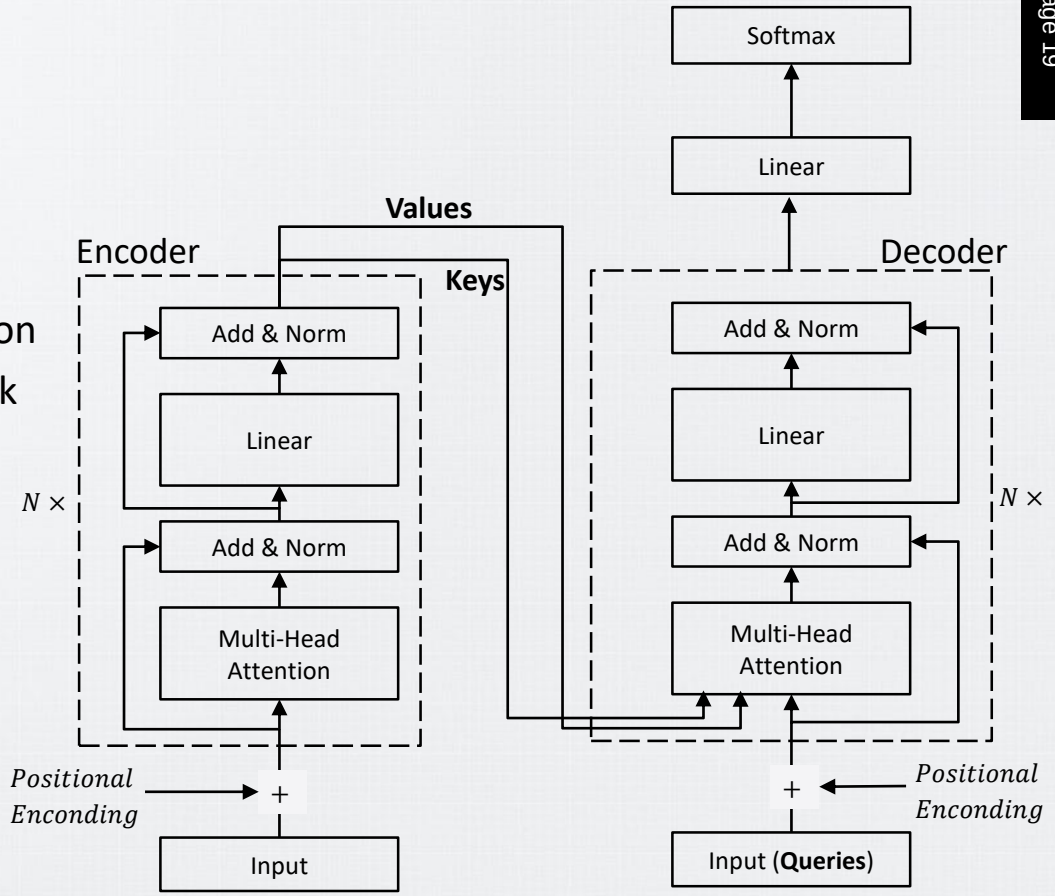


Overall Architecture

Transformer Architecture

- Encoder

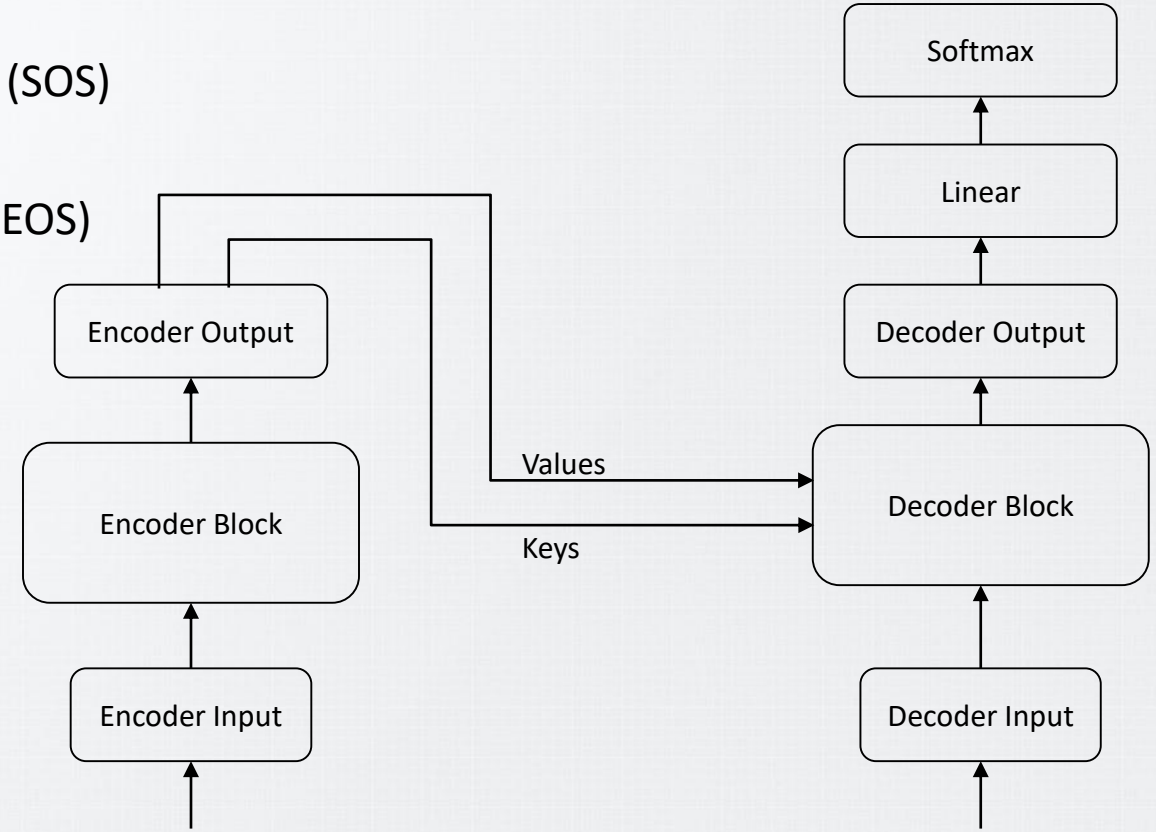
- Decoder
  - It performs multi-head attention on the output of the encoder stack



Training Stage

Transformer Architecture

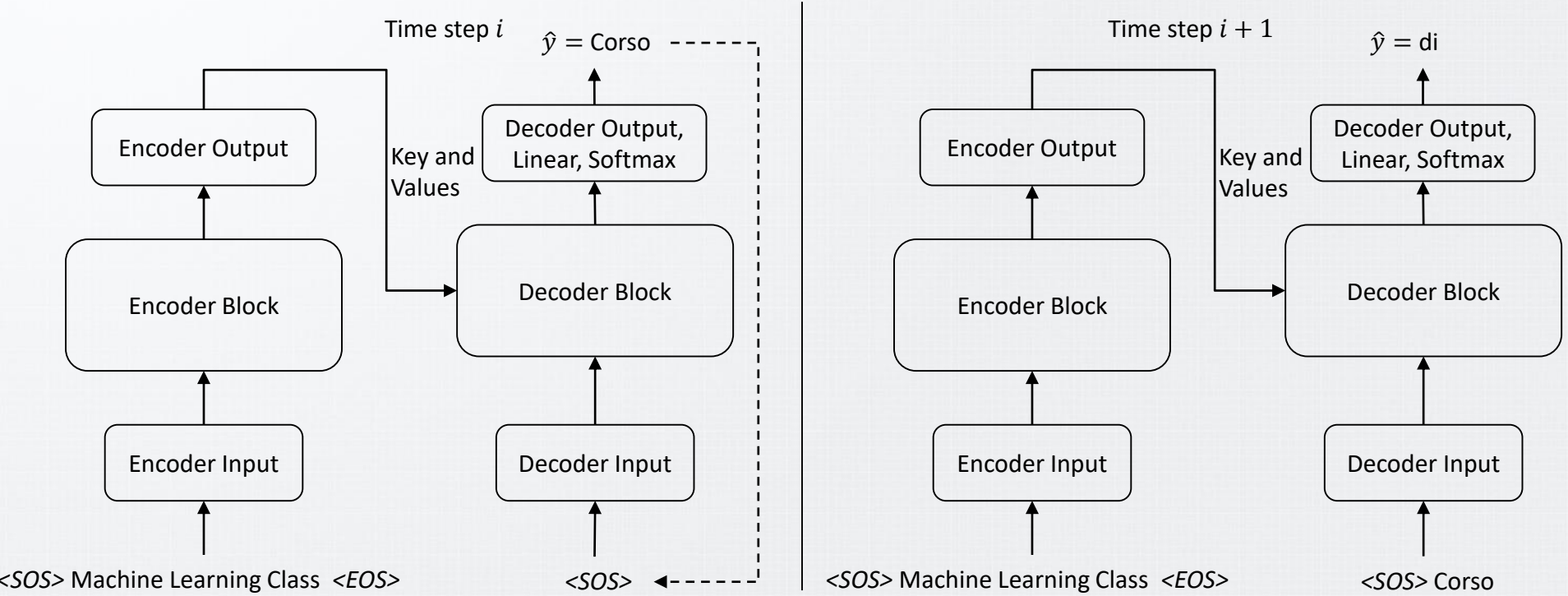
- **S**tart **o**f **S**equence (SOS)
- **E**nd **o**f **S**equence (EOS)



Inference Stage

Transformer Architecture

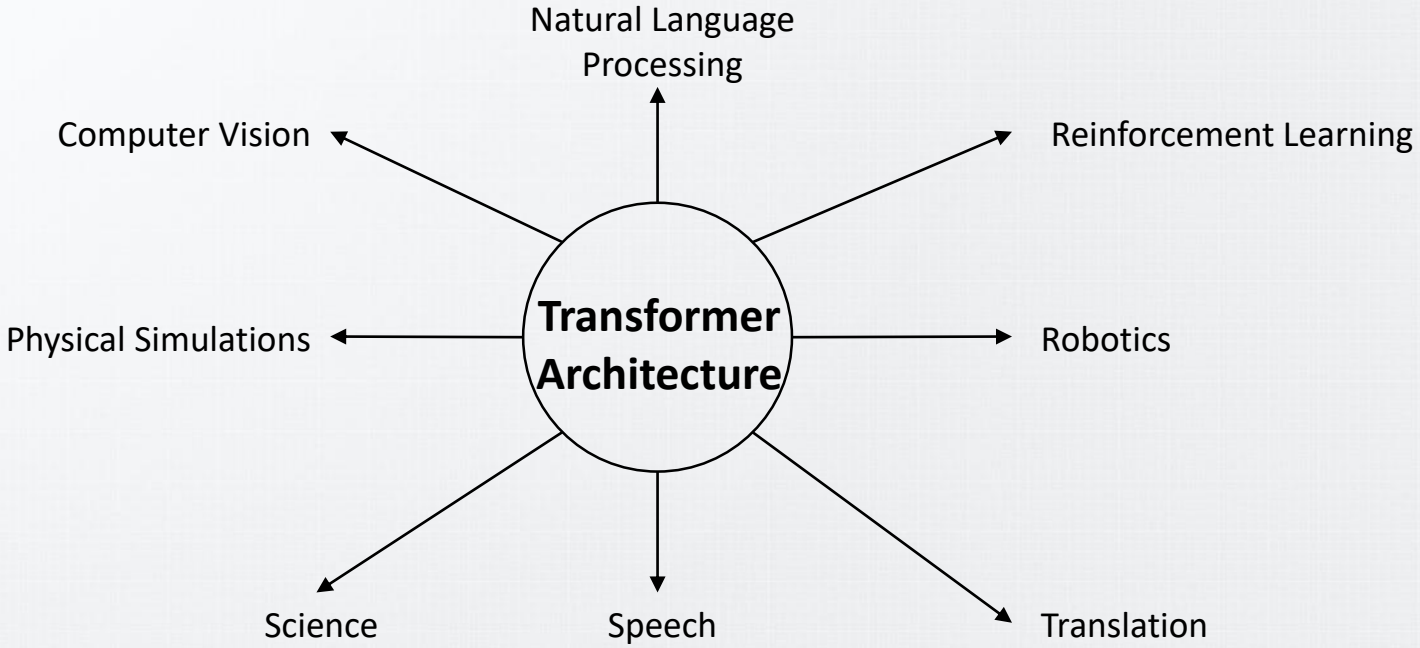
- Time steps
  - Append the previously output word (time step $i - \hat{y}$) to the decoder input (time step $i + 1$)



Multiple Tasks

Transformer Architecture

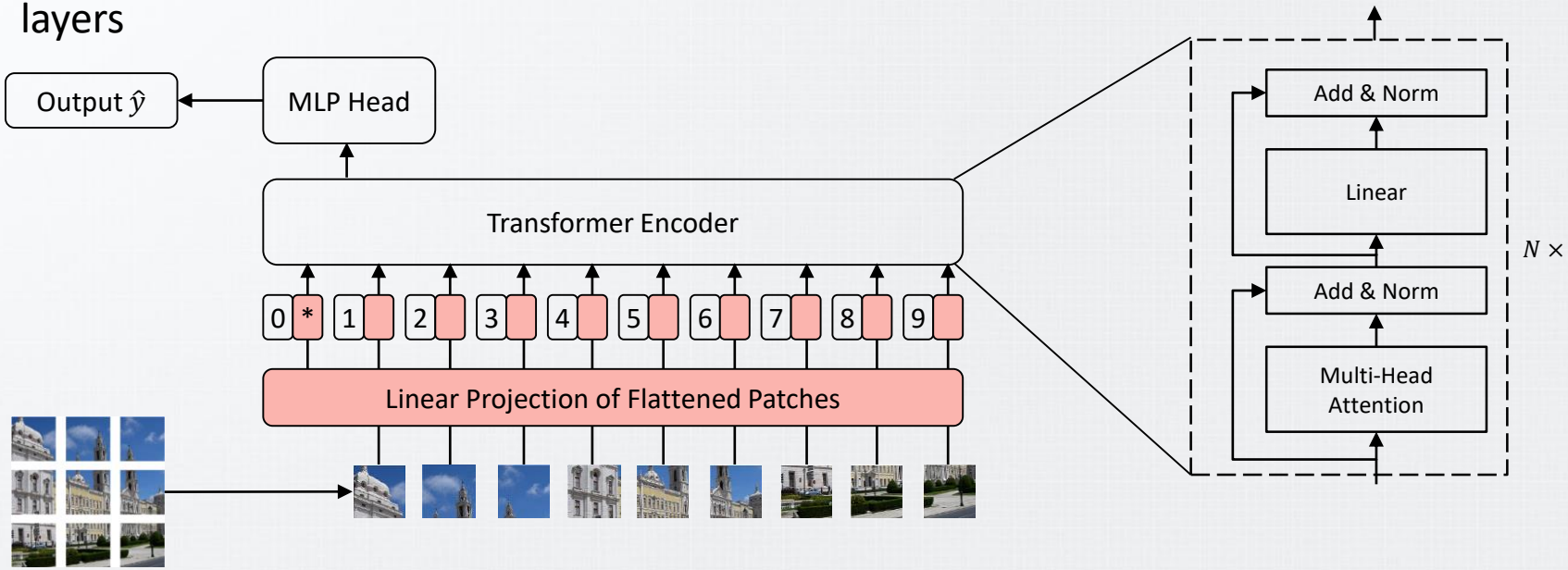
- Copy & Paste the transformer architecture and use it everywhere



Visual Transformer (ViT)

Transformer Architecture

- Transformer architecture for image recognition
- Transformer-based image recognition model that is fully built on the Transformer layers



Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. International Conference on Learning Representations (ICLR), 2021

Data Sensibility

Transformer Architecture

- To achieve state-of-the-art results, ViT requires pre-training on large datasets
  - For example, ImageNet-21k or JFT-300M dataset

- The lack of the typical convolutional inductive bias makes these models more **data-hungry** than common CNNs (Liu et al., 2021)

Model	CIFAR-10	CIFAR-100	Flowers102
ViT (variation)	84.19	65.16	31.73
ResNet-50	91.78	72.80	46.92

Source: Liu et al., 2021

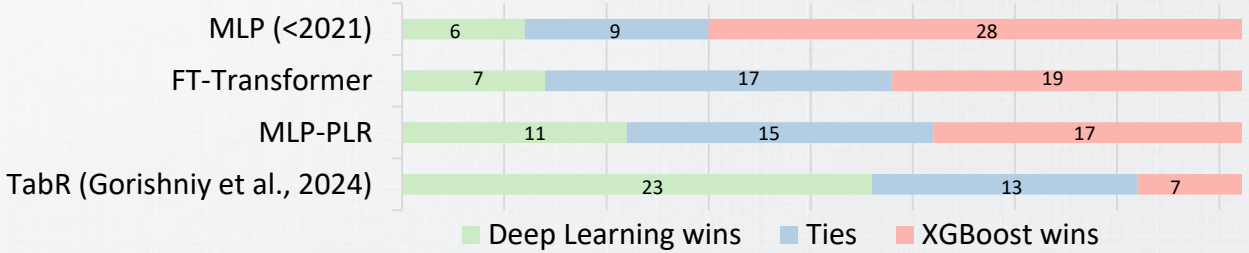
Liu et al. *Efficient Training of Visual Transformers with Small Datasets*. Neural Information Processing Systems (NeurIPS), 2021

Transformer for Tabular Data

Transformer Architecture

- TabPFN (Hollmann et al., 2023)
  - A transformer model for classification that is pre-trained on synthetic data to be applied to unseen datasets, quickly and without hyperparameter tuning

- In the context of tabular data, transformers (and other neural networks) do not achieve state-of-the-art performance (Grinsztajn et al., 2022; Gorishniy et al. 2024)



Hollmann et al. *TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second*. International Conference on Learning Representations (ICLR), 2023

Grinsztajn et al. *Why do tree-based models still outperform deep learning on typical tabular data?* Neural Information Processing Systems (NeurIPS), 2022

Gorishniy et al. *TabR: Tabular Deep Learning Meets Nearest Neighbors*. International Conference on Learning Representations (ICLR), 2024

Modern MLP Architectures

Introduction

Modern MLP Architectures

- MLP architectures demonstrate better generalization because they capture global representations (e.g., structure) more effectively than CNN methods (Guo et al., 2023)

- Compared to CNNs and Transformers, these vision MLP architectures involve less **inductive bias and have potential to be applied to more diverse tasks** (Tang et al., 2022)

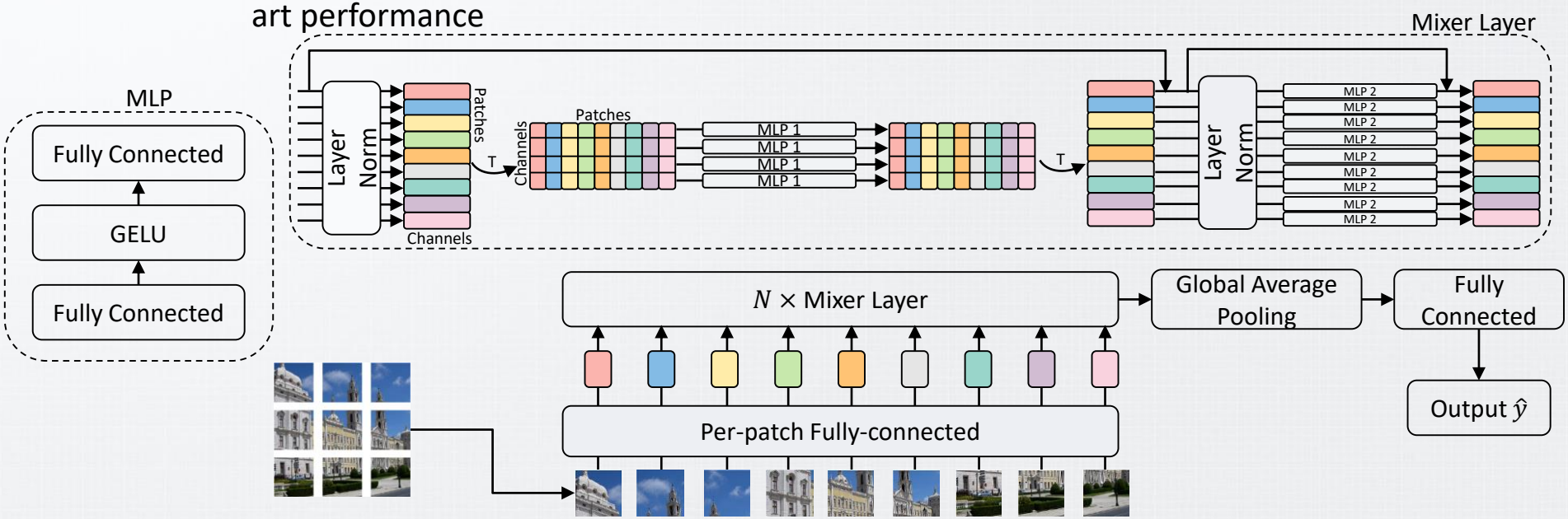
Tang et al. *An Image Patch is a Wave: Phase-Aware Vision MLP*. Computer Vision and Pattern Recognition (CVPR) 2022

Guo et al. *ALOFT: A Lightweight MLP-like Architecture with Dynamic Low-frequency Transform for Domain Generalization*. Computer Vision and Pattern Recognition (CVPR), 2023

Architecture Overview

Modern MLP Architectures

- The architecture is based entirely on multi-layer perceptrons (MLPs)
  - It does not employ any convolution or self-attention layer
  - Require pre-training on large datasets (i.e., ~100M images) to reach **achieve** state-of-the-art performance



Tolstikhin et al. *MLP-Mixer: An all-MLP Architecture for Vision*. Neural Information Processing Systems (NeurIPS), 2021

Foundation Models

Definition

Foundation Models

- Large deep learning models trained on a broad range of data (**web-scale training**) with the capacity to transfer their knowledge to unseen (downstream) tasks
  - Impressive generalization abilities

- Any model is trained on broad data (generally using self-supervision at scale), for which we can adapt it to a wide range of downstream tasks (Bommasani et al., 2021)

- Transformer architectures have ushered in the era of foundation models
  - Most foundation models are based on Transformers

Bommasani et al. *On the opportunities and risks of foundation models*. ArXiv 2021

Keys to Foundation Models

Foundation Models

- We could **transfer the large-scale knowledge** captured in these models to target (downstream) tasks by training on a small amount of labeled data

- Keys to Foundation Models
  - Fine-tuning
  - Self-supervised learning

Inductive Bias

Foundation Models

- The tendency of a model to prioritize one solution over another as it extrapolates between data points

- Inductive biases (prior knowledge) show preferences for solutions with certain properties
  - The preference for one choice over others

- Types of inductive bias
  - Locality and translation invariance (CNNs): Bias towards identifying patterns independent of their location within the image
  - Relational (GNNs): Bias towards prioritizing information from neighbors nodes
  - Simplicity Bias (Kirichenko et al., 2023): Bias towards rely on the simple features while ignoring predictive and complex features

Bommasani et al. *Last Layer Re-training is Sufficient for Robustness to Spurious Correlations*. International Conference on Learning Representations (ICLR), 2023

Inductive Bias

Foundation Models

- Transformers and MLPs architectures **don’t have inductive bias** of convolutional networks
  - By using huge amounts of data, it can surmount this disadvantage

- **Scaling compute** compensates for the lack of inductive bias (Tolstikhin et al, 2021; Bachmann et al., 2023)

Tolstikhin et al. *MLP-Mixer: An all-MLP Architecture for Vision*. Neural Information Processing Systems (NeurIPS), 2021

Bachmann et al. *Scaling MLPs: A Tale of Inductive Bias*. In Neural Information Processing Systems (NeurIPS), 2023

Popular Foundation Models

Foundation Models

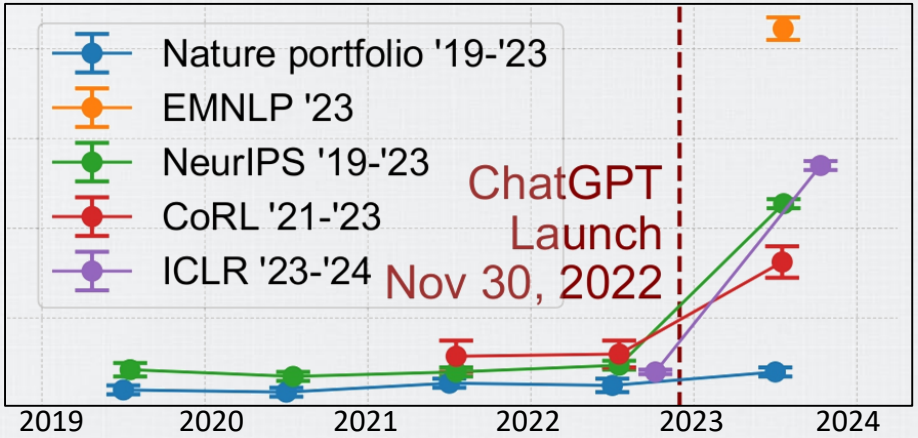
Model	Task	Number of Parameters	AI Research Organization	Dataset size (Corpous size)
AlphaFold	Protein folding	$18M - 60M$	DeepMind	$3.3B$
BART	Text generation/Understanding	$140M - 400M$	Facebook	$33B$
BERT	Language Understanding Question Answering	$110M - 340M$	Google	$3.3B$
GPT-3	Language Understanding	$175B$	OpenAI	$500B$
CLIP	Image classification	$63M$	OpenAI	$400M$ (image-text pairs)
DALL-E	Text-to-image generation	$12B$	OpenAI	$250M$ (image-text pairs)
StableDiffusion	Text-to-image generation	$890M$	LMU Munich + Stability.ai + Eleuther.ai	$5B$ (image-text pairs)

Amatriain et al. *Transformer models: an introduction and catalog*. ArXiv 2023

ChatGPT

Foundation Models

- The impact of ChatGPT on the popularization of Transformers
  - OpenAI released ChatGPT in November 2022, making it the fastest-growing app in history, reaching 1 million users in less than a month and 100 million in less than two months (Amatriain et al., 2023)



Source: Liang et al., 2024

Amatriain et al. *Transformer models: an introduction and catalog*. ArXiv 2023

Liang et al. Monitoring AI-Modified Content at Scale: A Case Study on the Impact of ChatGPT on AI Conference Peer Reviews. International Conference on Machine Learning (ICML), 2024

ChatGPT

Foundation Models

- The impact of ChatGPT on the popularization of Transformers

CVPR Review Policy

LLM policy: Remember that you can use an **LLM to refine your review text** if you think it is helpful. **But you CAN'T show the paper to an LLM in any way, because doing so is a major violation of policy**. We think we can detect people showing papers to LLMs, and we will prosecute people we catch. Don't do this.

What is the LLM Policy for referees in CVPR 2024?

Details: https://cvpr.thecvf.com/Conferences/2024/ReviewerGuidelines

Referees may use any device, including an **LLM, to polish their review wording**, but must vouch for, and be responsible for, the accuracy of the review. It is a significant act of referee misconduct to allow an LLM to see a submission. **PCs interpret showing a submission to an LLM as a deliberate referee violation of confidentiality.**

Clarification on Large Language Model Policy LLM (Details: https://icml.cc/Conferences/2023/llm-policy)

Papers that include text generated from a large-scale language model (LLM) such as ChatGPT are prohibited unless the produced text is presented as a part of the paper's experimental analysis.

- The Large Language Model (LLM) policy for ICML 2023 prohibits text produced entirely by LLMs (i.e., "generated"). This does not prohibit authors from using LLMs for **editing or polishing author-written text**.

A Change of Paradigm

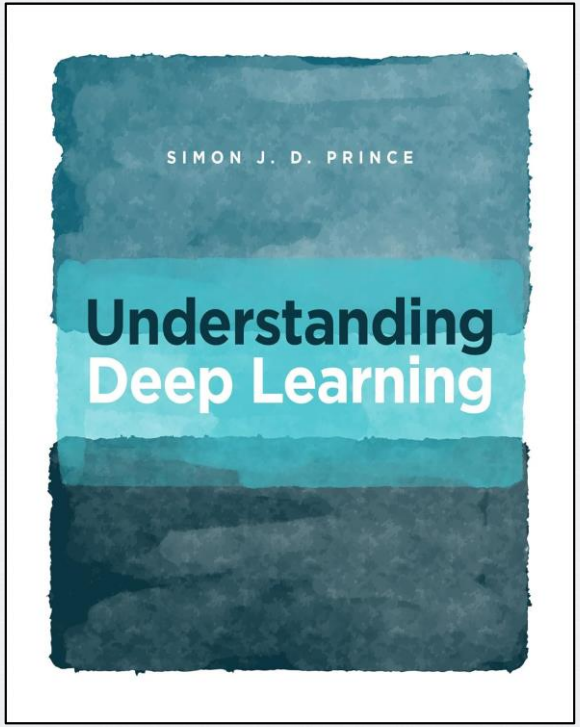
Foundation Models

Classic Machine Learning	Foundation Model Learning
Single model	Model Zoos
Model is trained from scratch	Model is rigid and cannot always be modified (often accessible via API)
Model is the “secret sauce”	Data is the “secret sauce”
Trivial to fine-tune	Non trivial to fine-tune and maintain over time
With sufficient funding, it is possible to establish a local infrastructure	Shortage of accelerators

Bibliography

Bibliography

- Understanding Deep Learning
  - Chapter 12
    - 12.2 Dot-product self-attention
    - 12.3 Extensions to dot-product self-attention
    - 12.4 Transformers



Bibliography

- Vaswani et al. *Attention Is All You Need*. Neural Information Processing Systems (NeurIPS), 2017

- Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.* International Conference on Learning Representations (ICLR), 2021

- Liu et al. *Efficient Training of Visual Transformers with Small Datasets*. Neural Information Processing Systems (NeurIPS), 2021

Bibliography

- Bachmann et al. *Scaling MLPs: A Tale of Inductive Bias*. Neural Information Processing Systems (NeurIPS), 2021

- Touvron et al. *ResMLP: Feedforward Networks for Image Classification With Data-Efficient Training*. Pattern Analysis and Machine Intelligence (PAMI), 2023

- Liu et al. *Understanding the Difficulty of Training Transformers*. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020

NEURAL INFORMATION PROCESSING SYSTEMS