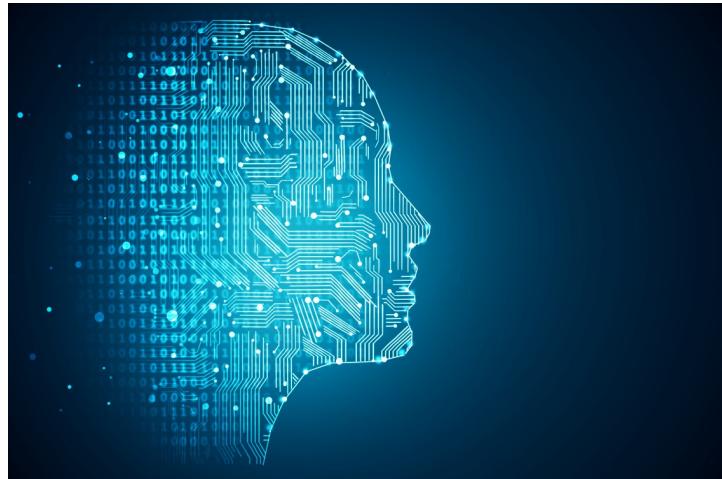


Introduction to Machine Learning



Wintersession 2024
Jan 16-18, 22-23

Gage DeZoort
Julian Gold
Jake Snell



With materials from:

Brian Arnold, Jonathan Halverson, Christina Peters, Savannah Thias, and Amy Winecoff

Course Outline

Date	Instructor	Topic
Day 1 Tue. 1/16	Jake Snell	Machine Learning Overview and Simple Models
Day 2 Wed. 1/17	Jake Snell	Model Evaluation and Improving Performance
Day 3 Thu. 1/18	Julian Gold	Introduction to Neural Networks
Day 4 Mon. 1/22	Gage DeZoort	Survey of Neural Network Architectures
Day 5 Tue. 1/23	Gage DeZoort Julian Gold Jake Snell	Hackathon! <ul style="list-style-type: none">• Computer Vision• Diffusion Models• Large Language Models

Transformers & Large Language Models

Jake Snell

January 23, 2024

The Era of Large-Scale Deep Learning

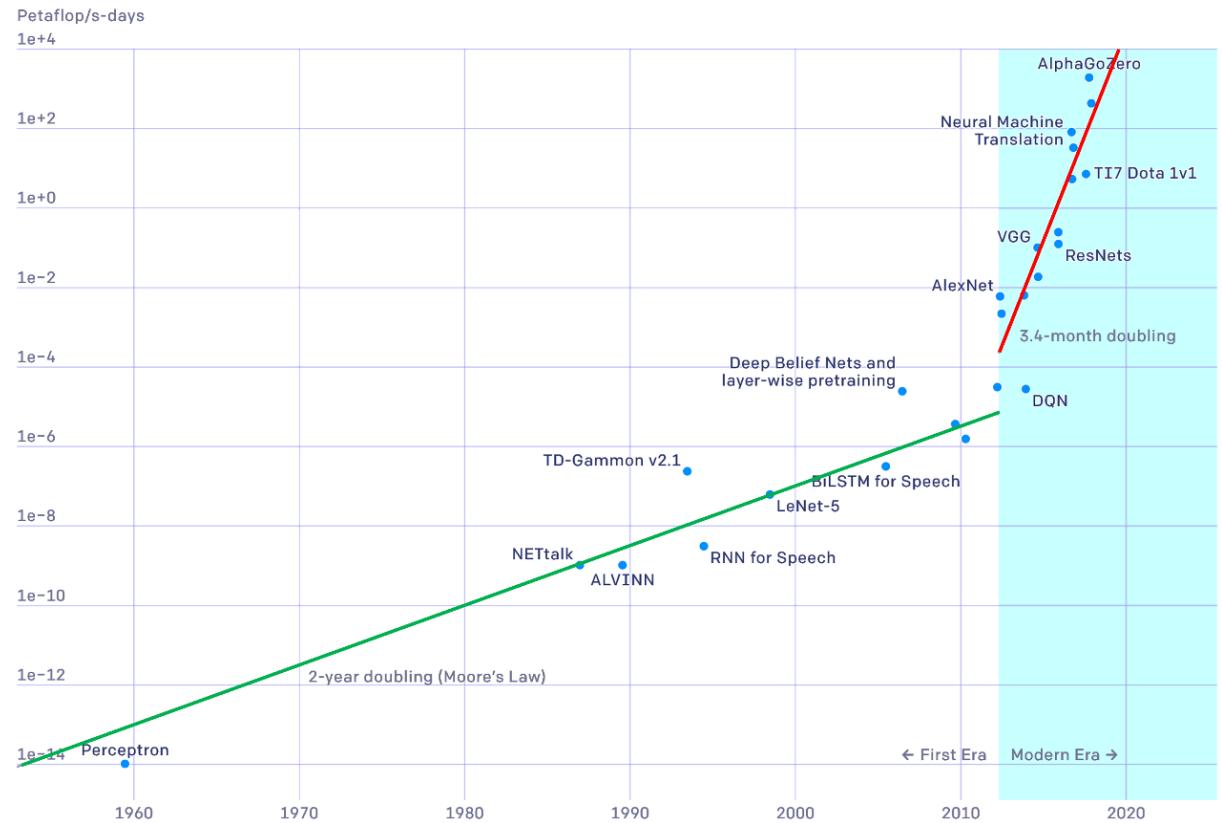


Figure 1.16 Plot of the number of compute cycles, measured in petaflop/s-days, needed to train a state-of-the-art neural network as a function of date, showing two distinct phases of exponential growth. [From OpenAI with permission.]

Figure credit:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Rise of Generative AI

User help me write a short note to introduce myself to my neighbor

ChatGPT Hi there,
My name is [Your Name] and I just moved in next door. I wanted to reach out and introduce myself. I'm looking forward to getting to know you and the neighborhood. If you ever need anything or just want to chat, please don't hesitate to stop by or give me a call.
Best,
[Your Name]

User can you make it more formal?

ChatGPT Dear Neighbor,
I am writing to introduce myself. My name is [Your Name] and I have recently moved in next door. I wanted to take a moment to reach out and say hello. I am looking forward to getting to know you and the neighborhood.
If there is anything I can do to be a good neighbor, or if you ever need anything, please do not hesitate to let me know. I am happy to help in any way that I can.
Sincerely,
[Your Name]

ChatGPT, OpenAI (2022)



SDXL, Stability AI (2023)

Test Yourself: Which Faces Were Made by A.I.?

By Stuart A. Thompson Jan. 19, 2024

Was this made by A.I.?

1/10

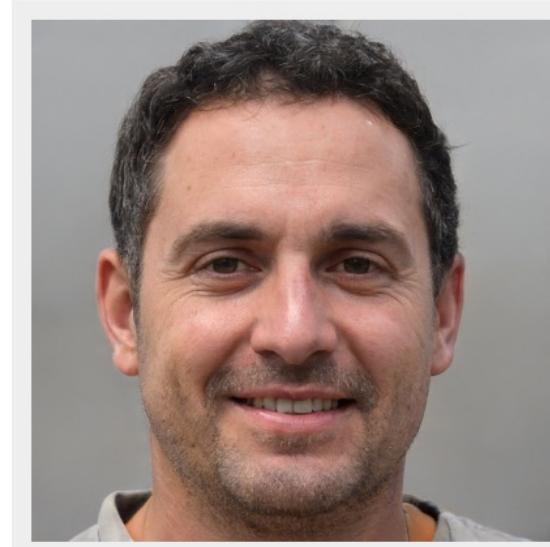


A.I.

Real

Was this made by A.I.?

2/10



A.I.

Real

<https://www.nytimes.com/interactive/2024/01/19/technology/artificial-intelligence-image-generators-faces-quiz.html>

Write a short story explaining how an ant could sink an aircraft carrier.

Example from: Piantadosi, Steven. 2023. “Modern Language Models Refute Chomsky’s Approach to Language.” LingBuzz Preprint, LingBuzz 7180.



an aircraft carrier, but it was large enough to hold many ants and was heavily armed with miniature torpedoes and cannons.

Finally, the day arrived when the ant ship was ready. The ants climbed aboard and set sail towards the nearest aircraft carrier. As they approached the carrier, the humans were shocked to see such a small ship coming towards them. They laughed and jeered, thinking that the ants' ship was no threat to them.

But the ants were not to be underestimated. They had been studying the human's ship, and they knew its weak spots. As they neared the carrier, they launched their miniature torpedoes, which hit the carrier directly in its vulnerable areas. The carrier began to sink, and the humans panicked.

In the end, the ant ship emerged victorious, having successfully sunk the massive aircraft carrier. The ants returned home to their colony as heroes, and their story became a legend that was passed down through the generations.

And the human learned to never underestimate the power of collective intelligence and unity.

It is important to note that the story above is a fictional tale and it is impossible for an ant to sink an aircraft carrier, as in reality the size, strength and weapon capability of an aircraft carrier would far exceed the capability of a group of ants.

Example from: Piantadosi, Steven. 2023. "Modern Language Models Refute Chomsky's Approach to Language." LingBuzz Preprint, LingBuzz 7180.

Explain the fundamental theorem of arithmetic in the style of Donald Trump.

Example from: Piantadosi, Steven. 2023. “Modern Language Models Refute Chomsky’s Approach to Language.” LingBuzz Preprint, LingBuzz 7180.



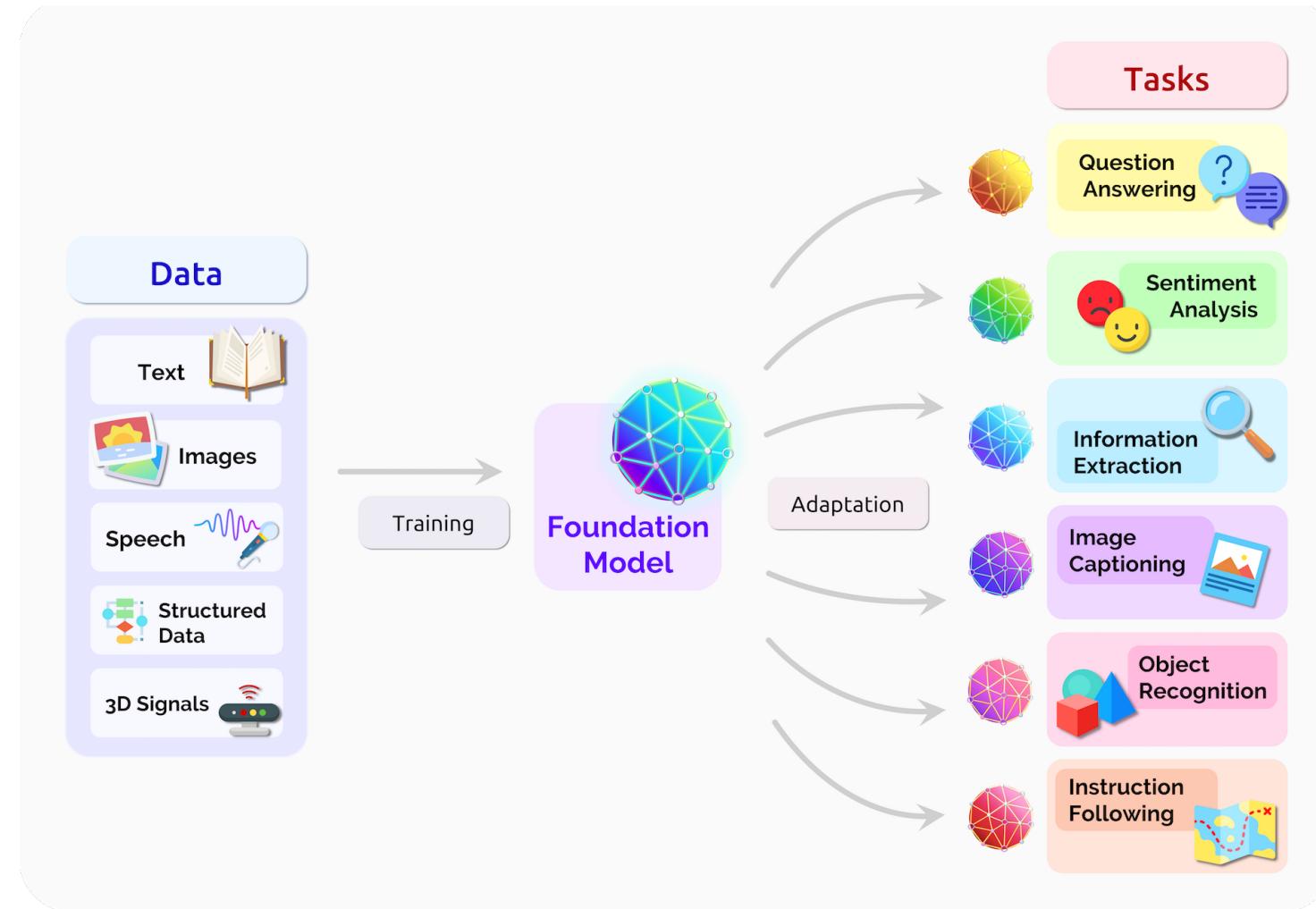


Fig. 2. A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks.

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, et al. 2022. “On the Opportunities and Risks of Foundation Models.” arXiv. <http://arxiv.org/abs/2108.07258>.

Language Models are Few-Shot Learners

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Table 2.2: Datasets used to train GPT-3. “Weight in training mix” refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. “Language Models Are Few-Shot Learners.” arXiv.
<http://arxiv.org/abs/2005.14165>.

For large models like these, compute consumption is measured in petaFLOPS-days. One petaFLOPS-day is the number of computations that could be performed in one day by a computer capable of calculating a thousand trillion computations (specifically, floating point operations) per second. For comparison, a standard laptop would need about a year to reach one petaFLOPS-day.¹⁰ That laptop would need several millennia to reach the 3,640 petaFLOPS-days it took to train GPT-3. On the world's hundredth-fastest supercomputer, GPT-3 could be trained in two and a half years, and even on the world's fastest supercomputer, training would still take over a week.*

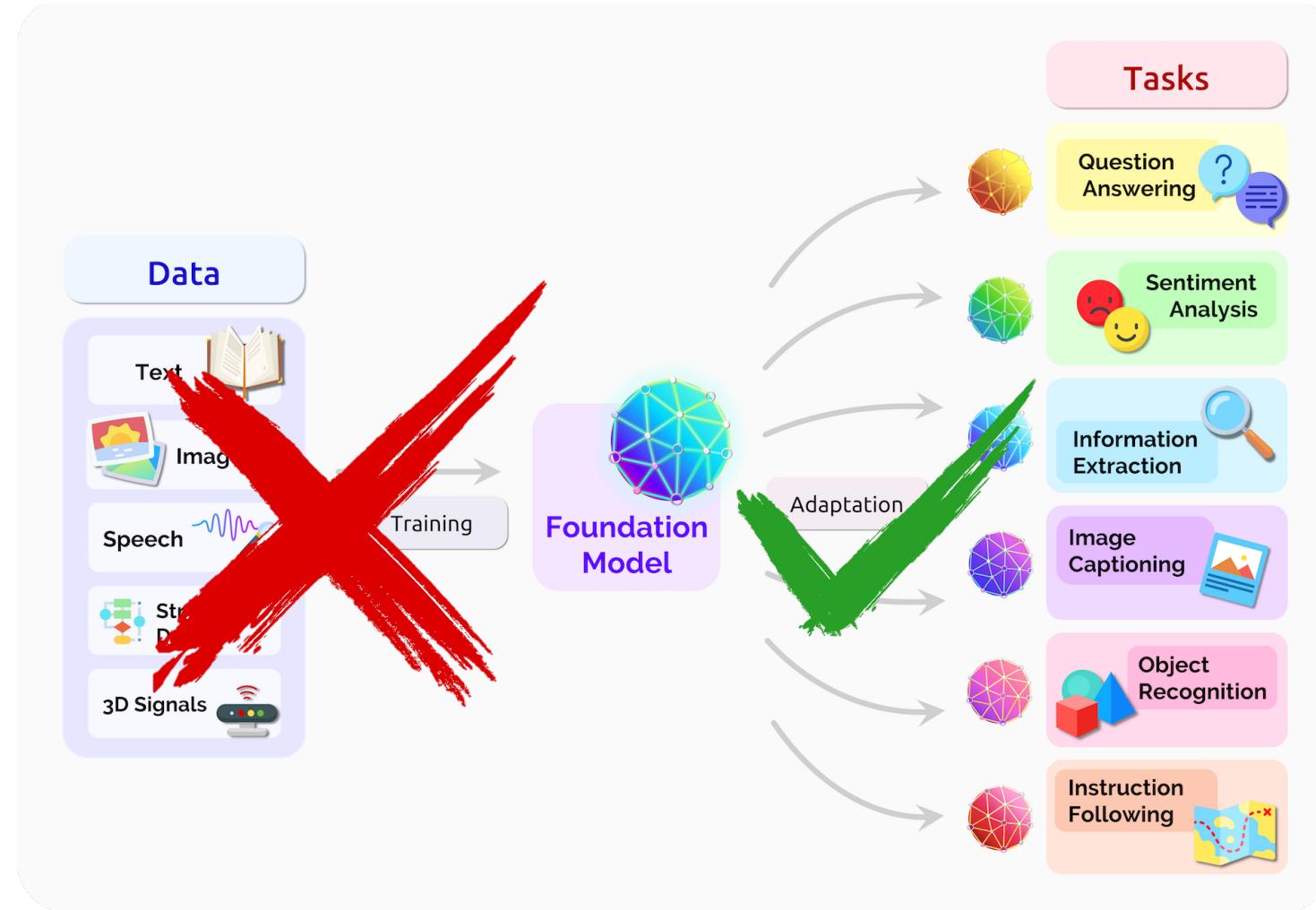


Fig. 2. A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks.

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, et al. 2022. “On the Opportunities and Risks of Foundation Models.” arXiv. <http://arxiv.org/abs/2108.07258>.



Hugging Face

Models

[stabilityai/stable-code-3b](#)

Updated 5 days ago • ↓ 2.52k • ❤ 296

[mistralai/Mixtral-8x7B-Instruct-v0.1](#)

Updated Dec 15, 2023 • ↓ 978k • ❤ 2.25k

[cagliostrolab/animagine-xl-3.0](#)

Updated 5 days ago • ↓ 62.4k • ❤ 470

[microsoft/phi-2](#)

Updated 5 days ago • ↓ 407k • ❤ 2.47k

[h94/IP-Adapter-FaceID](#)

Updated 4 days ago • ↓ 150k • ❤ 773

[Browse 300k+ models](#)

Datasets

[HuggingFaceM4/WebSight](#)

Updated 6 days ago • ↓ 3.54k • ❤ 154

[fka/awesome-chatgpt-prompts](#)

Updated Mar 7, 2023 • ↓ 3.3k • ❤ 4.47k

[Tele-AI/TeleChat-PTD](#)

Updated 10 days ago • ↓ 1.01k • ❤ 84

[argilla/distilabel-intel-orca-dpo-pairs](#)

Updated about 14 hours ago • ↓ 5.47k • ❤ 80

[Open-Orca/OpenOrca](#)

Updated Oct 21, 2023 • ↓ 47.1k • ❤ 1.03k

[Browse 50k+ datasets](#)

<https://huggingface.co>



Tasks 1 Libraries Datasets Languages Licenses
Other

Filter Tasks by name

Reset Tasks

Multimodal

- Feature Extraction Text-to-Image
- Image-to-Text Image-to-Video
- Text-to-Video Visual Question Answering
- Document Question Answering
- Graph Machine Learning Text-to-3D
- Image-to-3D

Computer Vision

- Depth Estimation Image Classification
- Object Detection Image Segmentation
- Image-to-Image
- Unconditional Image Generation
- Video Classification
- Zero-Shot Image Classification
- Mask Generation Zero-Shot Object Detection

Natural Language Processing

- Text Classification Token Classification
- Table Question Answering Question Answering
- Zero-Shot Classification Translation
- Summarization Conversational
- Text Generation Text2Text Generation
- Fill-Mask Sentence Similarity

Models 47,243

gpt2

Text Generation • Updated Jun 29, 2023 • 20.1M • 1.61k

baffo32/decapoda-research-llama-7B-hf

Text Generation • Updated Apr 10, 2023 • 6.59M • 18

distilgpt2

Text Generation • Updated Apr 29, 2023 • 1.94M • 309

tiiuae/falcon-40b-instruct

Text Generation • Updated Sep 29, 2023 • 1.32M • 1.14k

bigscience/bloom-560m

Text Generation • Updated Sep 26, 2023 • 1.18M • 301

mistralai/Mixtral-8x7B-Instruct-v0.1

Text Generation • Updated Dec 15, 2023 • 978k • 2.25k

meta-llama/Llama-2-7b-chat-hf

Text Generation • Updated Nov 13, 2023 • 756k • 2.48k

davidkim205/komt-mistral-7b-v1

Text Generation • Updated Oct 24, 2023 • 635k • 18

teknium/OpenHermes-2-Mistral-7B

Text Generation • Updated Nov 2, 2023 • 630k • 230

TheBloke/Mixtral-8x7B-Instruct-v0.1-GPTQ

Text Generation • Updated Dec 14, 2023 • 614k • 72

<https://huggingface.co/models>

NLP Tasks

- Text classification
- Token classification
- Question answering
- Text generation
- Translation
- Summarization
- Multiple choice
- ...

TASK GUIDES

NATURAL LANGUAGE PROCESSING

Text classification

Token classification

Question answering

Causal language modeling

Masked language modeling

Translation

Summarization

Multiple choice

Agenda

- Background on foundation
- Language Modeling with Transformers
- Hands-on Exercise: Fine-tuning LLM with HuggingFace

Language Modeling

- Consider a sentence such as:
“I swam across the river to get to the other bank.”

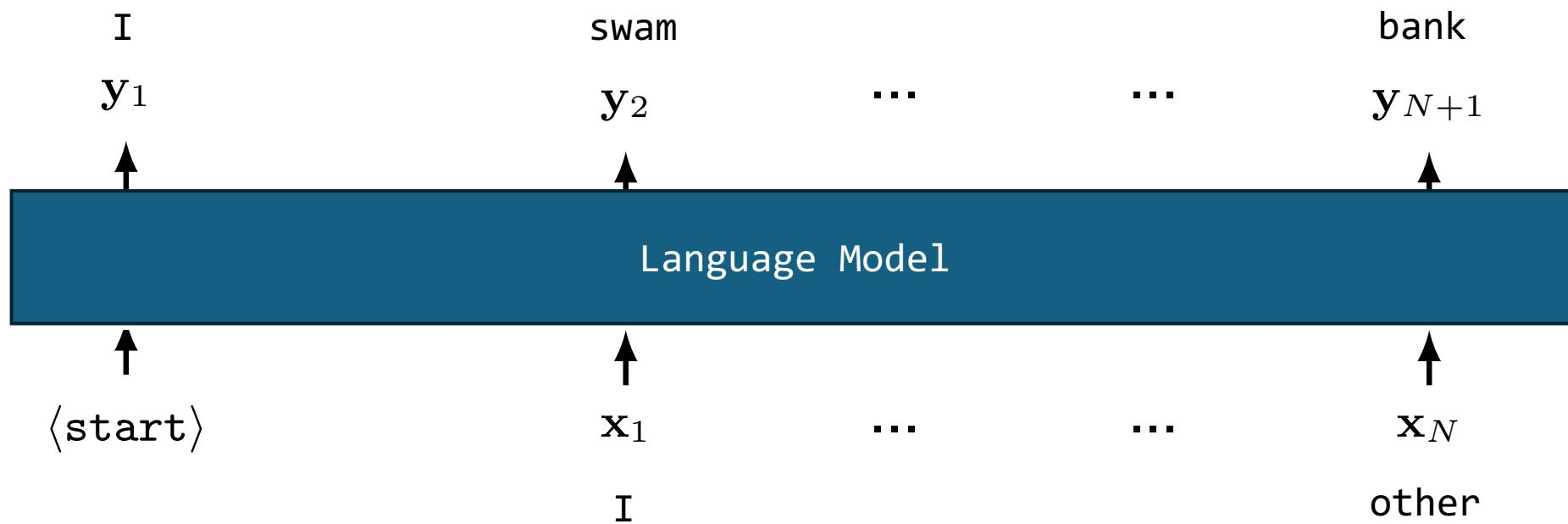


Figure based upon:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Tokenization

- After tokenization, a string is represented by a sequence of integers.
- Could treat each word as a separate token.
- Subword tokenization is popular nowadays.

Figure 12.12 An illustration of the process of tokenizing natural language by analogy with byte pair encoding. In this example, the most frequently occurring pair of characters is ‘pe’, which occurs four times, and so these form a new token that replaces all the occurrences of ‘pe’. Note that ‘Pe’ is not included in this since upper-case ‘P’ and lower-case ‘p’ are distinct characters. Next the pair ‘ck’ is added since this occurs three times. This is followed by tokens such as ‘pi’, ‘ed’, and ‘per’, all of which occur twice, and so on.

Peter Piper picked a peck of pickled peppers
Peter Piper picked a peck of pickled peppers

Figure credit:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Embedding

The embedding process can be defined by a matrix \mathbf{E} of size $D \times K$ where D is the dimensionality of the embedding space and K is the dimensionality of the dictionary. For each one-hot encoded input vector \mathbf{x}_n we can then calculate the corresponding embedding vector using

$$\mathbf{v}_n = \mathbf{E}\mathbf{x}_n. \quad (12.26)$$

Because \mathbf{x}_n has a one-hot encoding, the vector \mathbf{v}_n is simply given by the corresponding column of the matrix \mathbf{E} .

Credit:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Autoregressive Modeling

- Decompose distribution over sequence into product of conditional distributions:

$$p(\mathbf{x}_1, \dots, \mathbf{x}_N) = \prod_{n=1}^N p(\mathbf{x}_n | \mathbf{x}_1, \dots, \mathbf{x}_{n-1})$$

- Could make independence assumption beyond L most recent tokens:

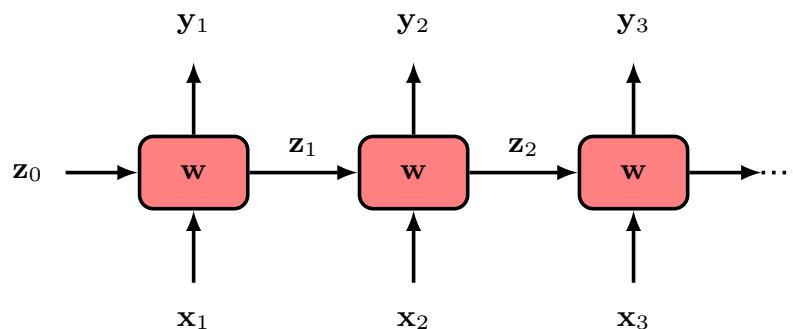
$$p(\mathbf{x}_1, \dots, \mathbf{x}_N) = p(\mathbf{x}_1)p(\mathbf{x}_2 | \mathbf{x}_1) \prod_{n=3}^N p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{x}_{n-2})$$

Equation credits:

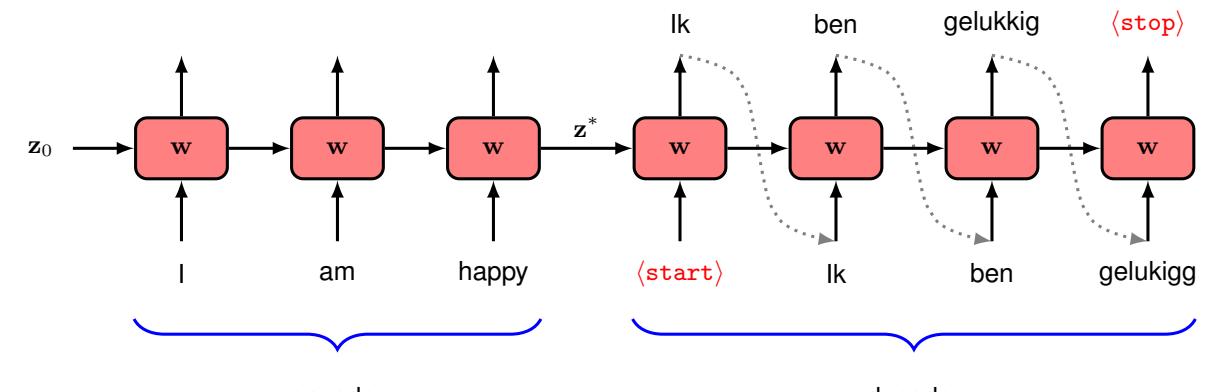
Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Recurrent Neural Networks

- Recurrent Neural Network (RNN) architectures were popular until a few years ago:
 - LSTM (Hochreiter & Schmidhuber, 1997)
 - GRU (Cho et al., 2014)



RNN



RNN for translation

Figure credits:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

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

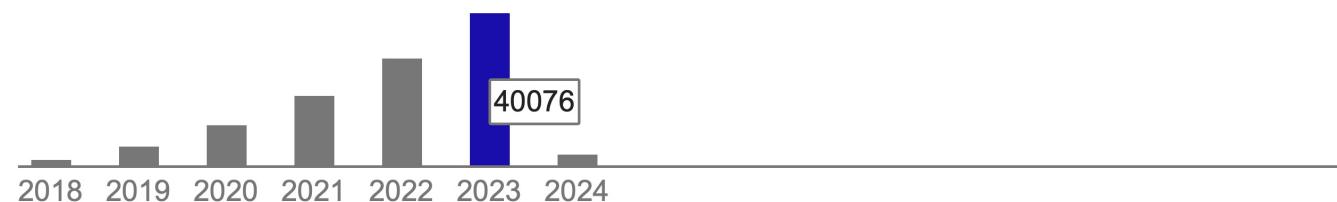
Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Published in NeurIPS 2017

Total citations Cited by 106297



Citation count from Google Scholar (accessed Jan. 23, 2024)

An Illustration of Attention

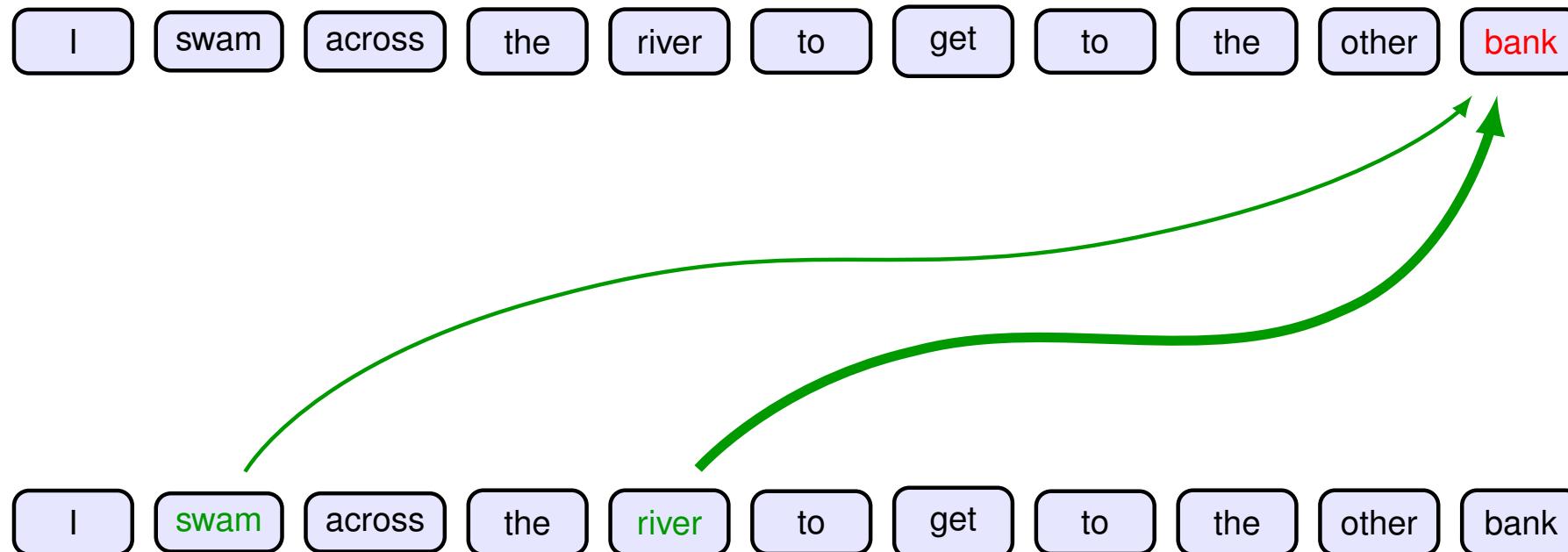
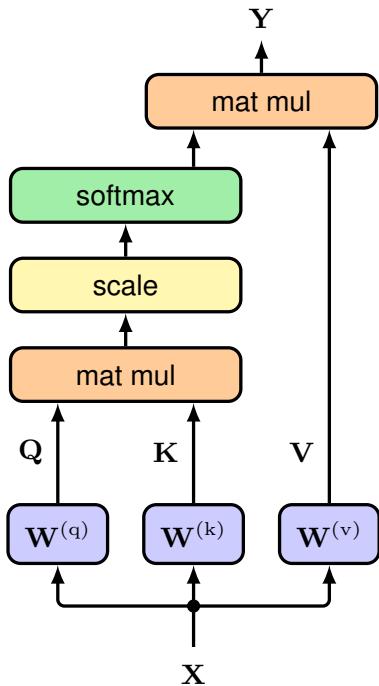


Figure 12.1 Schematic illustration of attention in which the interpretation of the word ‘bank’ is influenced by the words ‘river’ and ‘swam’, with the thickness of each line being indicative of the strength of its influence.

Figure credit:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Self-attention



Algorithm 12.1: Scaled dot-product self-attention

Input: Set of tokens $\mathbf{X} \in \mathbb{R}^{N \times D} : \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
Weight matrices $\{\mathbf{W}^{(q)}, \mathbf{W}^{(k)}\} \in \mathbb{R}^{D \times D_k}$ and $\mathbf{W}^{(v)} \in \mathbb{R}^{D \times D_v}$

Output: Attention($\mathbf{Q}, \mathbf{K}, \mathbf{V}$) $\in \mathbb{R}^{N \times D_v} : \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$

$\mathbf{Q} = \mathbf{X}\mathbf{W}^{(q)}$ // compute queries $\mathbf{Q} \in \mathbb{R}^{N \times D_k}$

$\mathbf{K} = \mathbf{X}\mathbf{W}^{(k)}$ // compute keys $\mathbf{K} \in \mathbb{R}^{N \times D_k}$

$\mathbf{V} = \mathbf{X}\mathbf{W}^{(v)}$ // compute values $\mathbf{V} \in \mathbb{R}^{N \times D}$

return Attention($\mathbf{Q}, \mathbf{K}, \mathbf{V}$) = Softmax $\left[\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}} \right] \mathbf{V}$

Figure credit:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Attention Coefficients

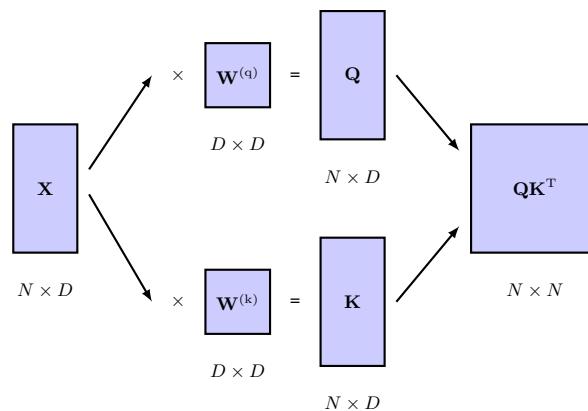


Figure 12.4 Illustration of the evaluation of the matrix QK^T , which determines the attention coefficients in a transformer. The input X is separately transformed using (12.10) and (12.11) to give the query matrix Q and key matrix K , respectively, which are then multiplied together.

Figure 12.5 Illustration of the evaluation of the output from an attention layer given the query, key, and value matrices Q , K , and V , respectively. The entry at the position highlighted in the output matrix Y is obtained from the dot product of the highlighted row and column of the Softmax $[QK^T]$ and V matrices, respectively.

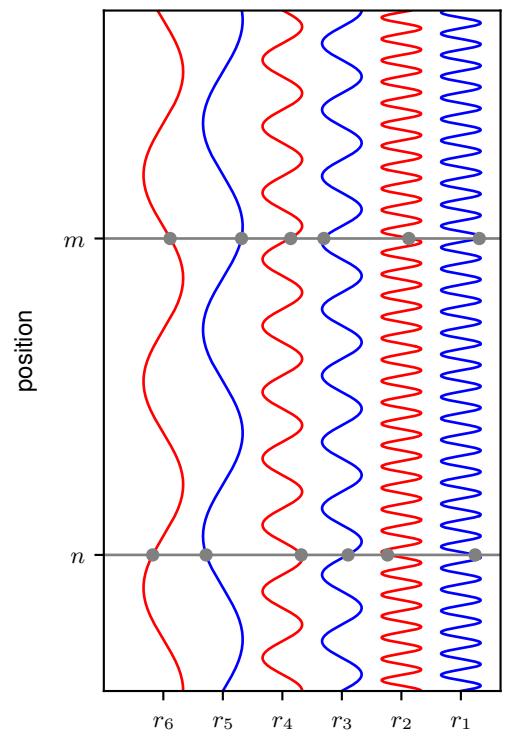
$$Y_{N \times D_v} = \text{Softmax} \left\{ \begin{array}{c} \text{highlighted row} \\ QK^T_{N \times N} \end{array} \right\} \times V_{N \times D_v}$$

Figure credit:

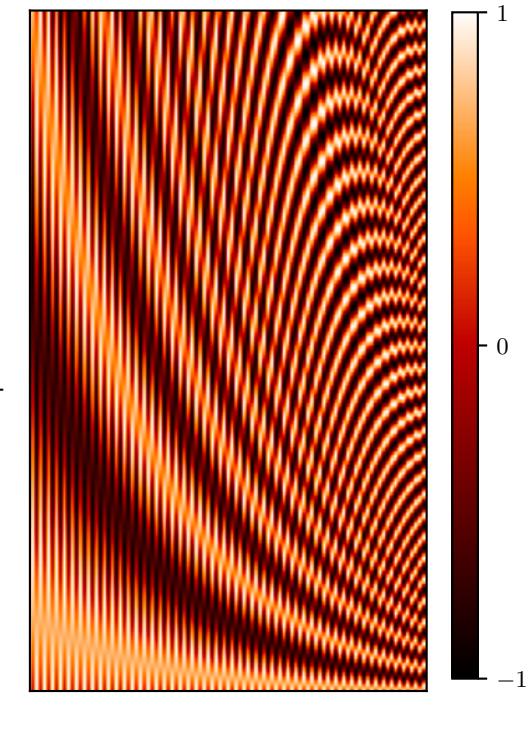
Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Positional Encoding

1 :	0	0	0	1
2 :	0	0	1	0
3 :	0	0	1	1
4 :	0	1	0	0
5 :	0	1	0	1
6 :	0	1	1	0
7 :	0	1	1	1
8 :	1	0	0	0
9 :	1	0	0	1



(a)



(b)

Figure 12.10 Illustrations of the functions defined by (12.25) and used to construct position-encoding vectors. (a) A plot in which the horizontal axis shows the different components of the embedding vector r whereas the vertical axis shows the position in the sequence. The values of the vector elements for two positions n and m are shown by the intersections of the sine and cosine curves with the horizontal grey lines. (b) A heat map illustration of the position-encoding vectors defined by (12.25) for dimension $D = 100$ with $L = 30$ for the first $N = 200$ positions.

Figure credit:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Masked Attention

Figure 12.16 An illustration of the mask matrix for masked self-attention. Attention weights corresponding to the red elements are set to zero. Thus, in predicting the token ‘across’, the output can depend only on the input tokens ‘⟨start⟩’ ‘I’ and ‘swam’.

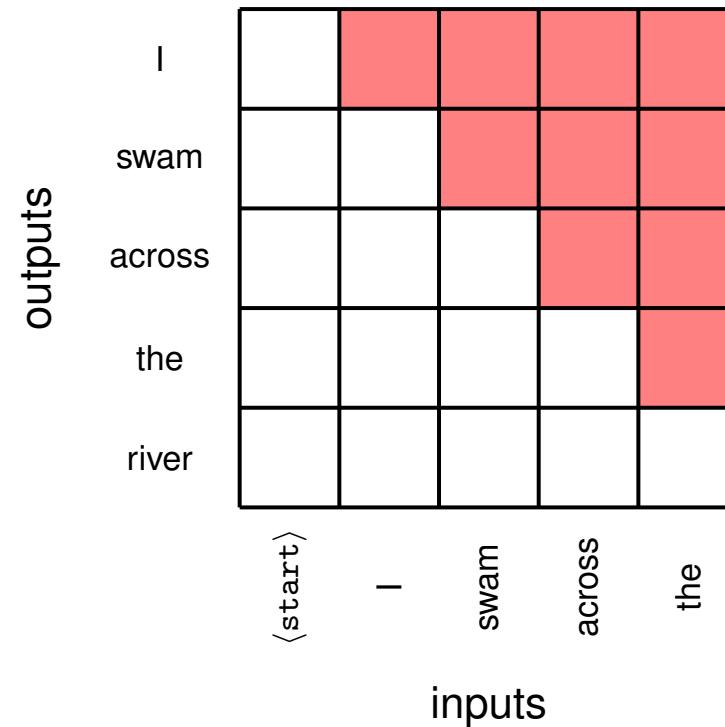
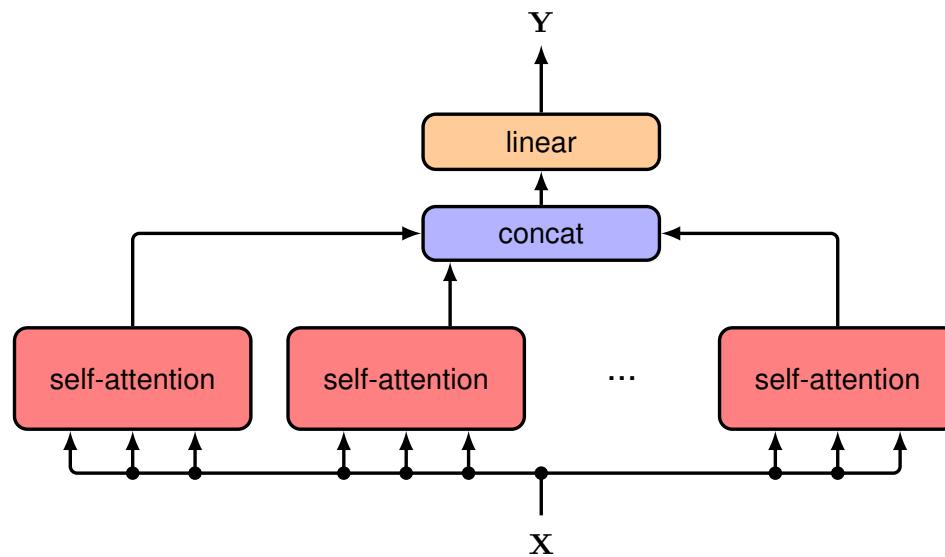


Figure credit:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Multi-head Attention



Algorithm 12.2: Multi-head attention

```
Input: Set of tokens  $\mathbf{X} \in \mathbb{R}^{N \times D} : \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$   
Query weight matrices  $\{\mathbf{W}_1^{(q)}, \dots, \mathbf{W}_H^{(q)}\} \in \mathbb{R}^{D \times D}$   
Key weight matrices  $\{\mathbf{W}_1^{(k)}, \dots, \mathbf{W}_H^{(k)}\} \in \mathbb{R}^{D \times D}$   
Value weight matrices  $\{\mathbf{W}_1^{(v)}, \dots, \mathbf{W}_H^{(v)}\} \in \mathbb{R}^{D \times D_v}$   
Output weight matrix  $\mathbf{W}^{(o)} \in \mathbb{R}^{HD_v \times D}$   
Output:  $\mathbf{Y} \in \mathbb{R}^{N \times D} : \{\mathbf{y}_1, \dots, \mathbf{x}_N\}$   
// compute self-attention for each head (Algorithm 12.1)  
for  $h = 1, \dots, H$  do  
     $\mathbf{Q}_h = \mathbf{X}\mathbf{W}_h^{(q)}$ ,    $\mathbf{K}_h = \mathbf{X}\mathbf{W}_h^{(k)}$ ,    $\mathbf{V}_h = \mathbf{X}\mathbf{W}_h^{(v)}$   
     $\mathbf{H}_h = \text{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h)$  //  $\mathbf{H}_h \in \mathbb{R}^{N \times D_v}$   
end for  
 $\mathbf{H} = \text{Concat}[\mathbf{H}_1, \dots, \mathbf{H}_N]$  // concatenate heads  
return  $\mathbf{Y}(\mathbf{X}) = \mathbf{H}\mathbf{W}^{(o)}$ 
```

Figure credit:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

GPT Architecture

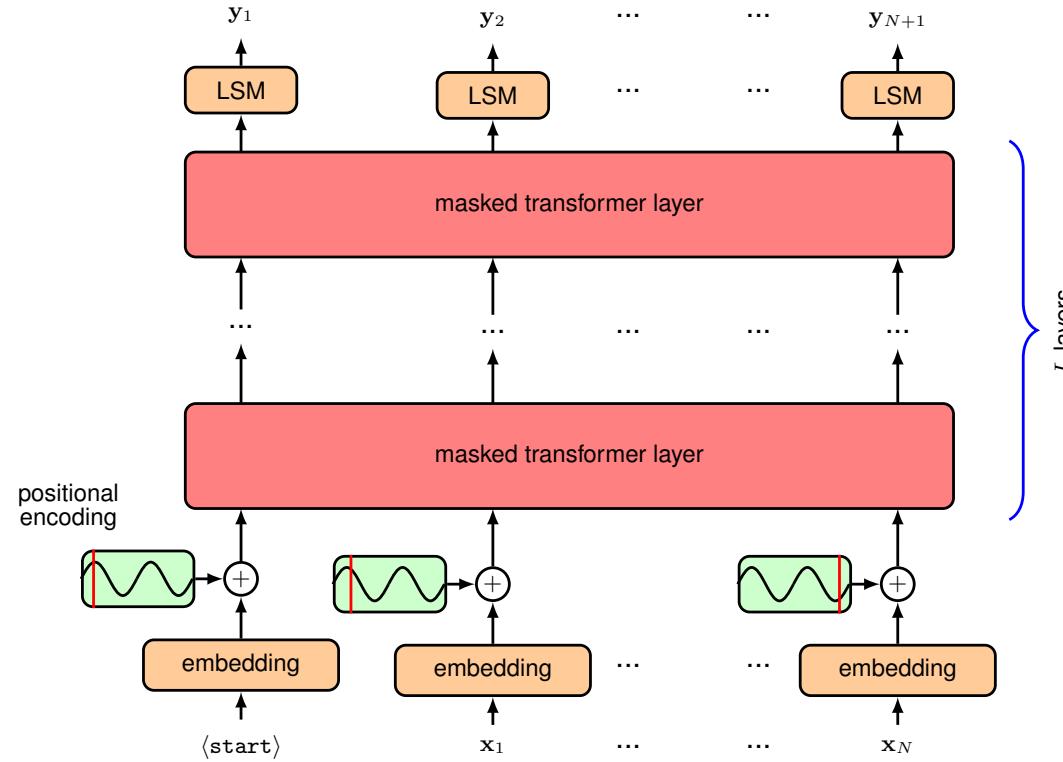


Figure 12.15 Architecture of a GPT decoder transformer network. Here 'LSM' stands for linear-softmax and denotes a linear transformation whose learnable parameters are shared across the token positions, followed by a softmax activation function. Masking is explained in the text.

Figure credit:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.

Fine-Tuning LLMs with !



https://github.com/PrincetonUniversity/intro_machine_learning/tree/main/day5/large_language_models_hackathon



Low-rank adaptation (LoRA)

(Hu et al., 2021)

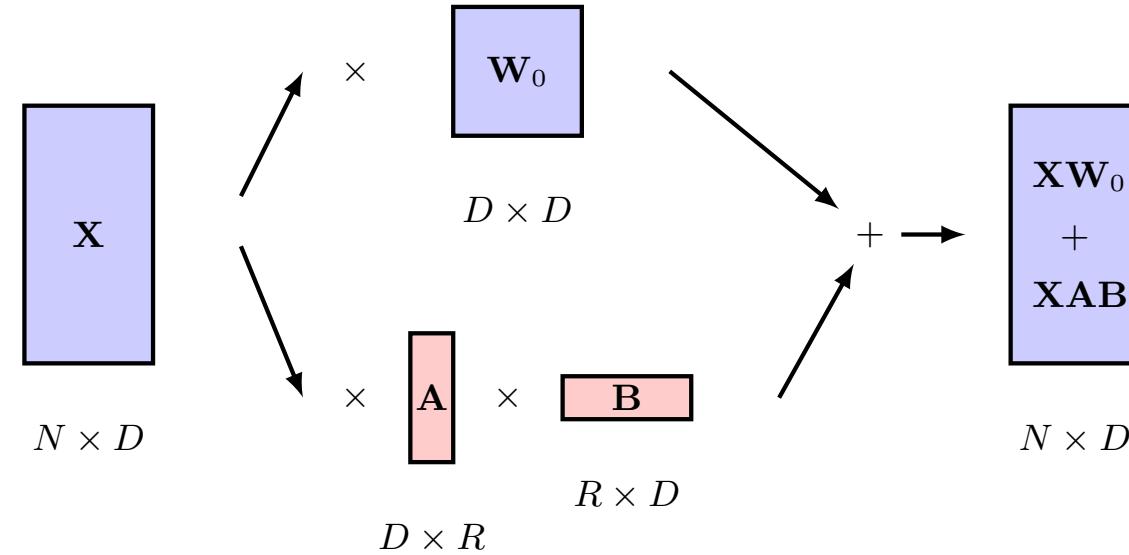


Figure 12.21 Schematic illustration low-rank adaptation showing a weight matrix \mathbf{W}_0 from one of the attention layers in a pre-trained transformer. Additional weights given by matrices \mathbf{A} and \mathbf{B} are adapted during fine-tuning and their product \mathbf{AB} is then added to the original matrix for subsequent inference.

Figure credit:

Bishop, Christopher M., and Hugh Bishop. 2024. Deep Learning: Foundations and Concepts.