Part #1

Foundations of LLMs

Part #2

LLM Workshop

http://ygelp.kyle.pub

slides + self-study material here

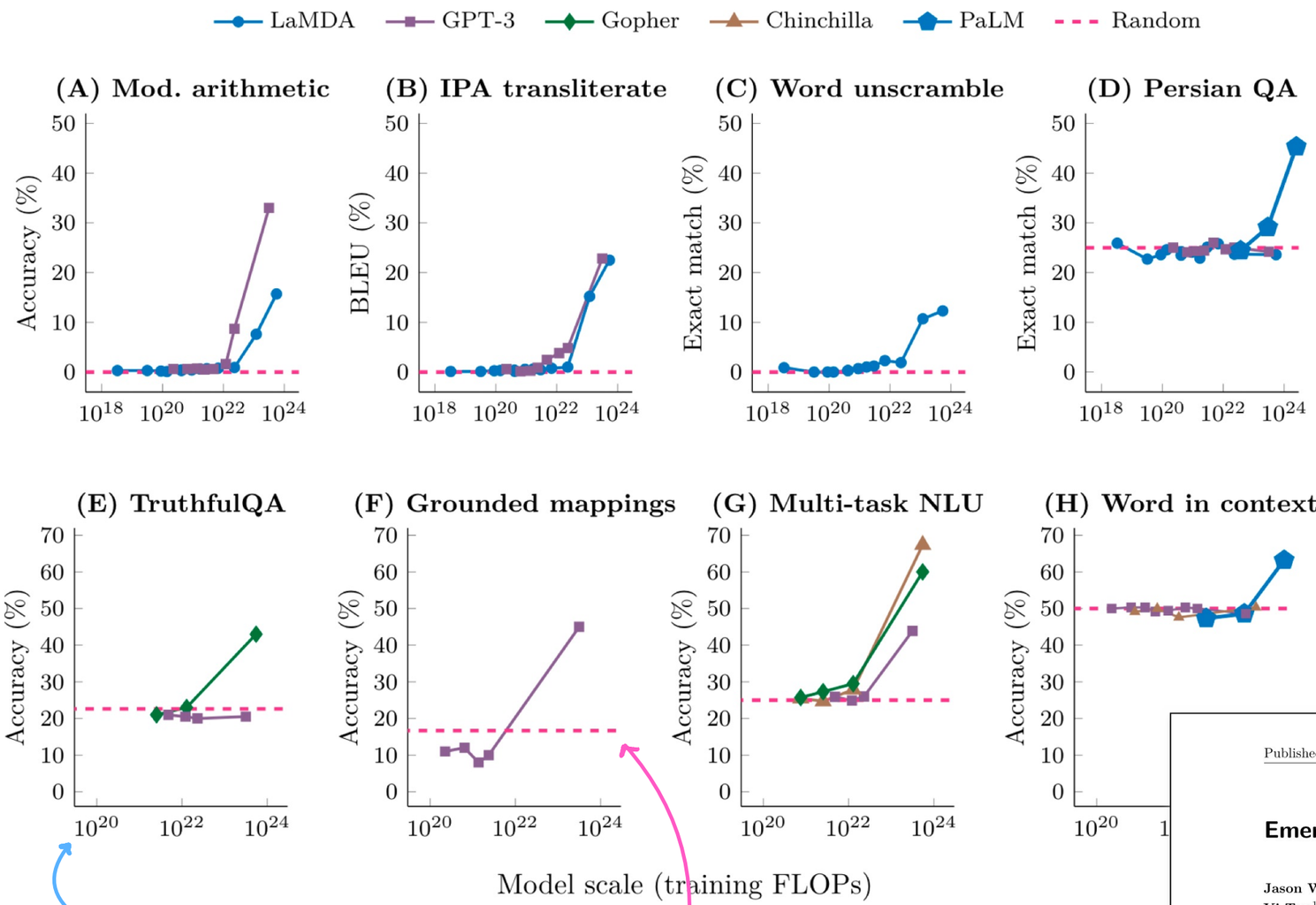
me on the 1910 Yale football team

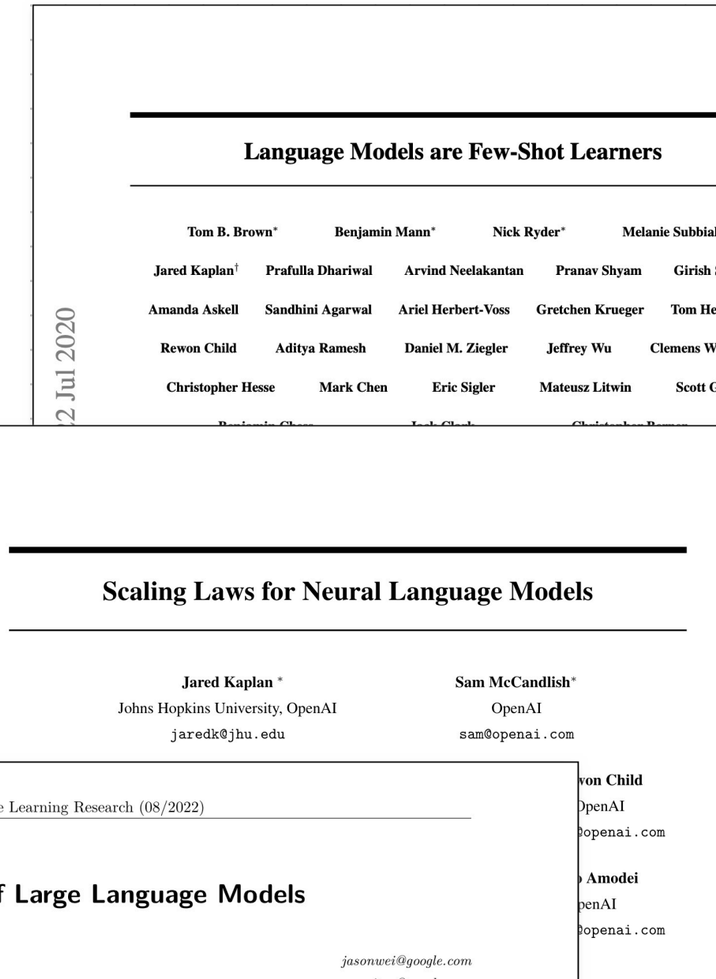


Large language models in a nutshell

1. ChatGPT is a "large language model"
2. LLMs are big, deep neural networks
3. Neural networks are excellent approximators of super complicated functions.
4. They're basically just a bunch of matrix multiplications.

LLMs are super suprising!




Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom He

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens W

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott G

2 Jul 2020

Scaling Laws for Neural Language Models

Jared Kaplan * Sam McCandlish*

Johns Hopkins University, OpenAI OpenAI

jaredk@jhu.edu sam@openai.com

won Child

OpenAI

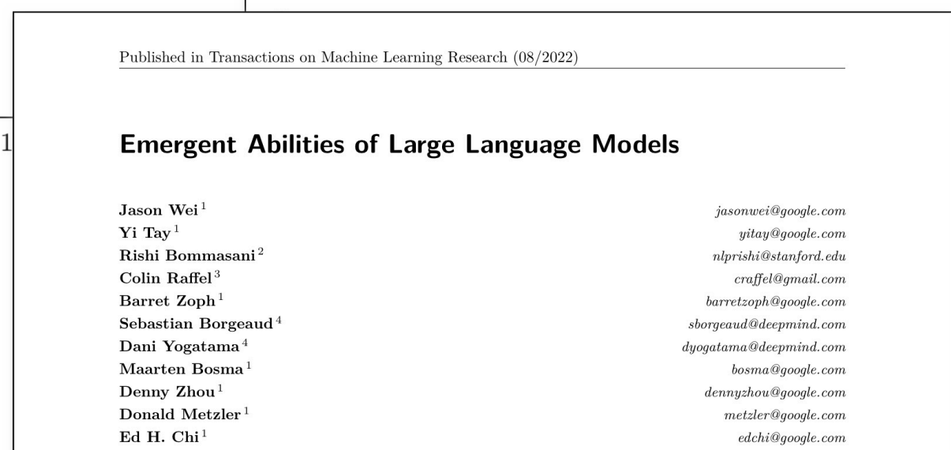
Openai.com

Amodei

penAI

Openai.com

ppy loss.
compute
e. Other
a wide
and the
mine the
sample-


Published in Transactions on Machine Learning Research (08/2022)

Emergent Abilities of Large Language Models

Jason Wei [1] jasonwei@google.com

Yi Tay [1] yitay@google.com

Rishi Bommasani [2] nlprishi@stanford.edu

Colin Raffel [3] craffel@gmail.com

Barret Zoph [1] barretzoph@google.com

Sebastian Borgeaud [4] sborgeaud@deepmind.com

Dani Yogatama [4] dyogatama@deepmind.com

Maarten Bosma [1] bosma@google.com

Denny Zhou [1] dennyzhou@google.com

Donald Metzler [1] metzler@google.com

Ed H. Chi [1] edchi@google.com

Most LLMs are based on "transformers"

Google, 2017

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

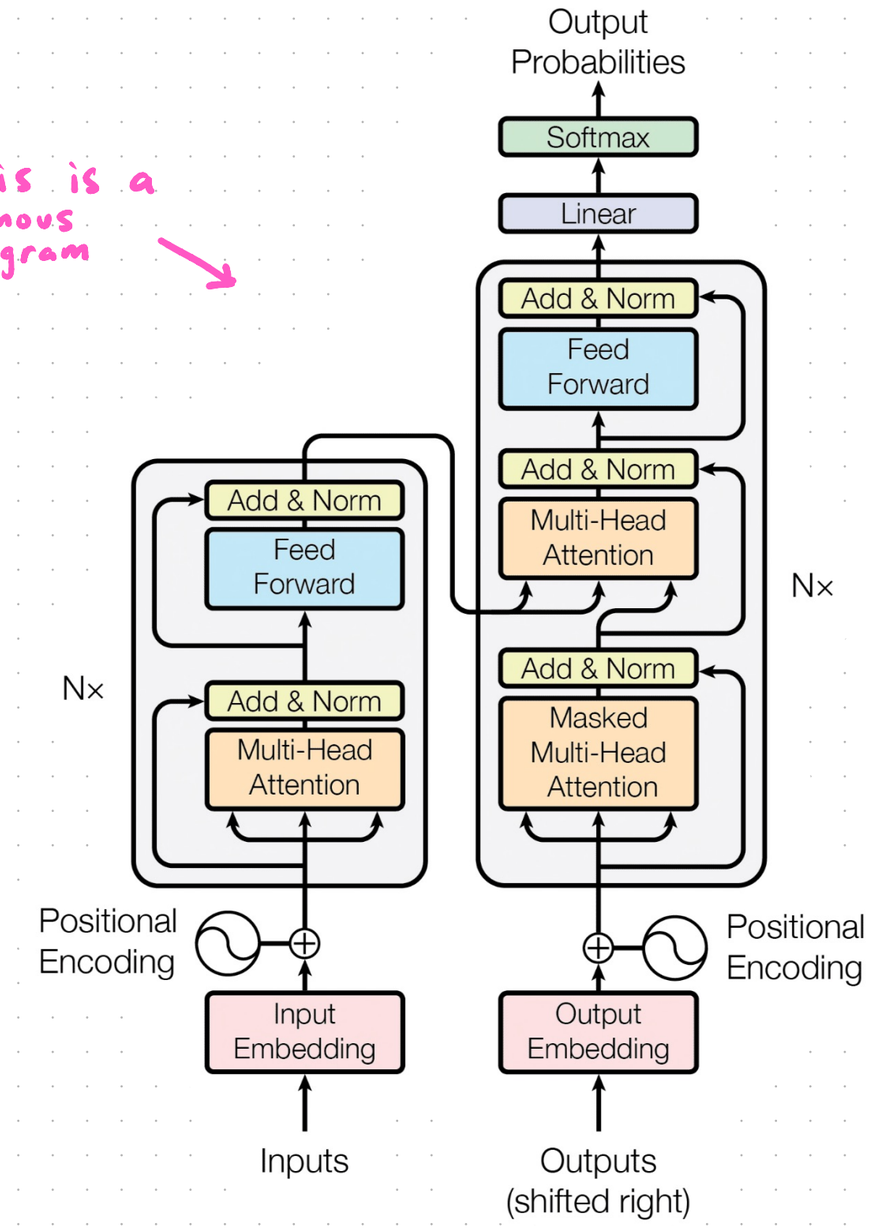
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

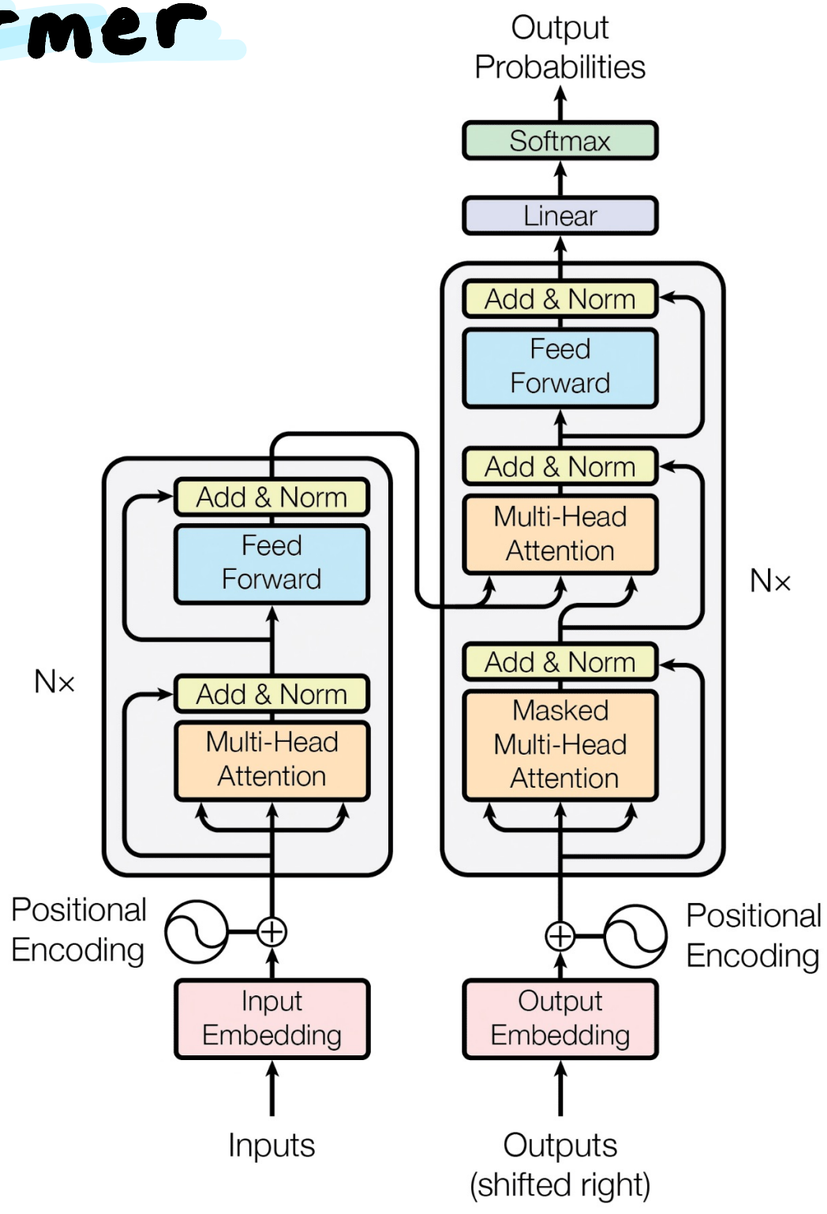
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

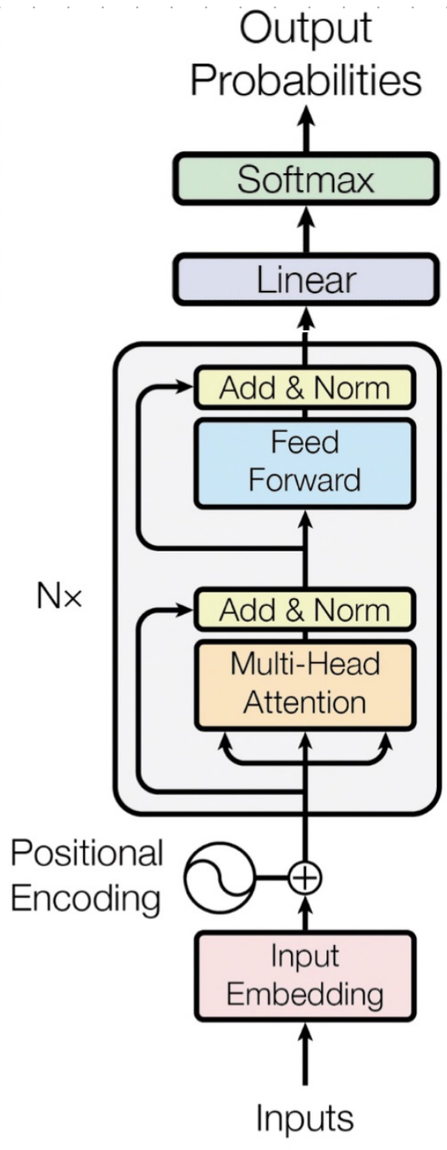
this is a famous diagram



The Transformer



encoder / decoder

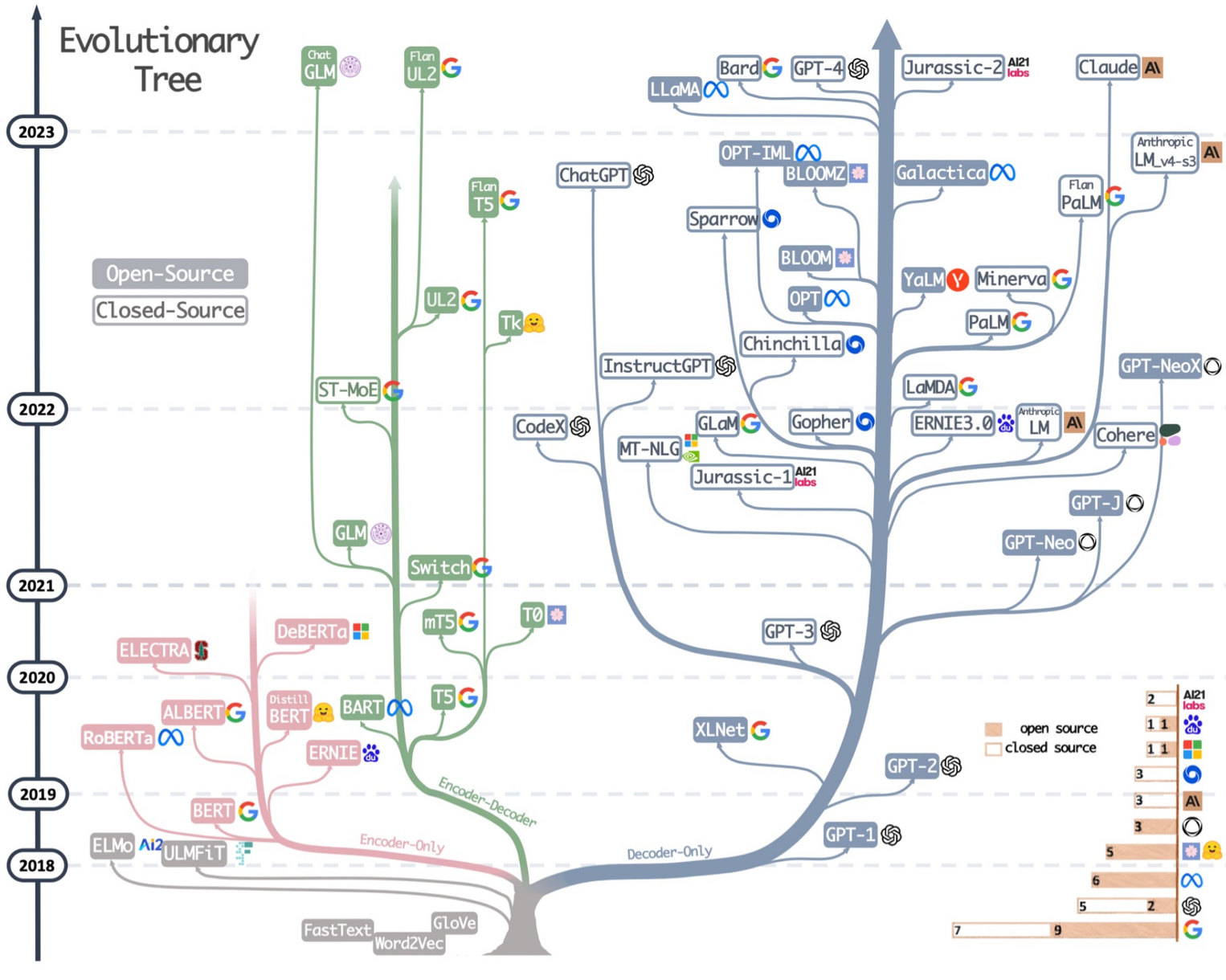


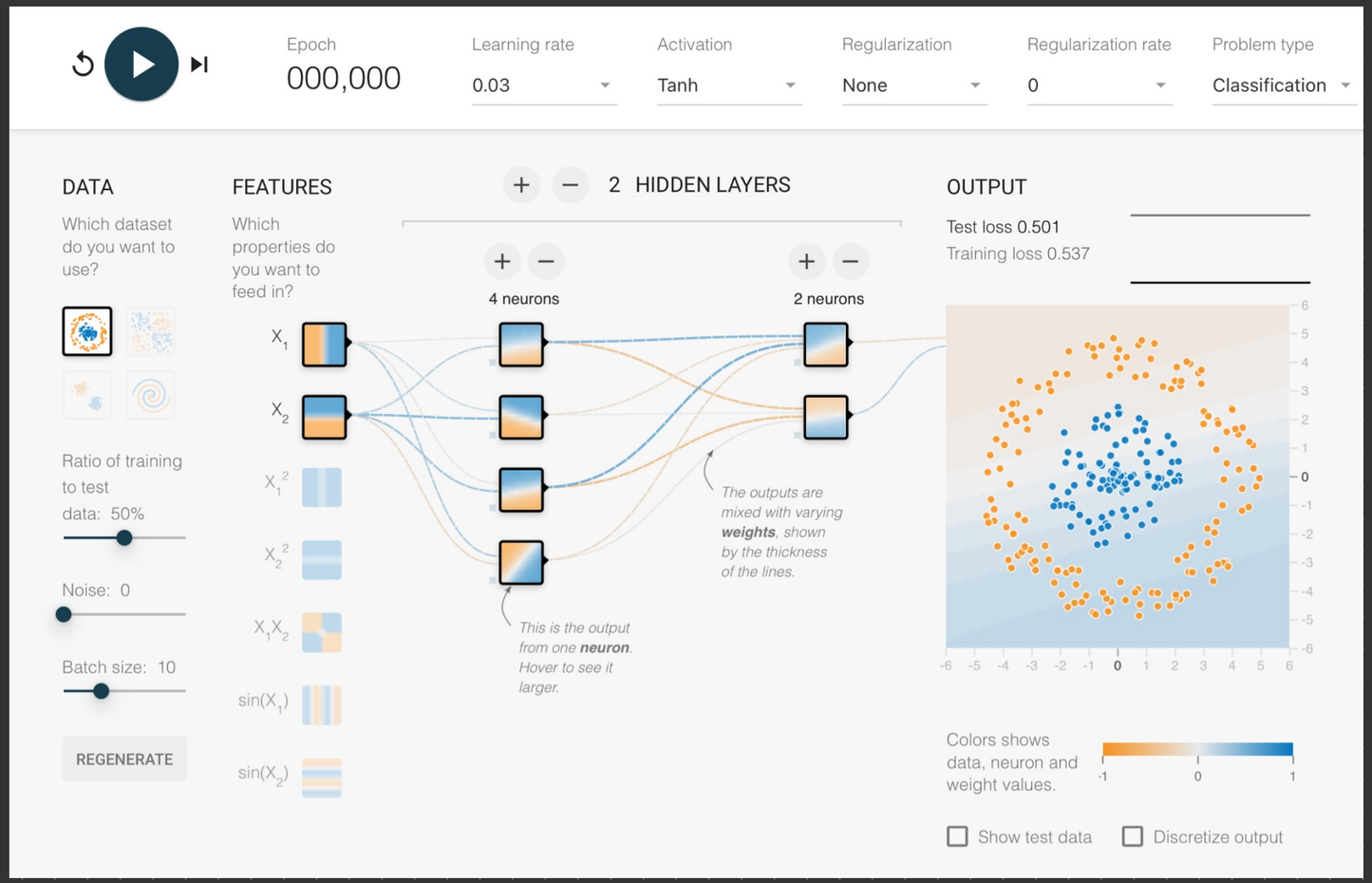
decoder only

The are many different LLMs

Table 1. Summary of Large Language Models.

	Characteristic		LLMs
Encoder-Decoder or Encoder-only (BERT-style)	Training: Model type: Pretrain task:	Masked Language Models Discriminative Predict masked words	ELMo [80], BERT [28], RoBERTa [65], DistilBERT [90], BioBERT [57], XLM [54], Xlnet [119], ALBERT [55], ELECTRA [24], T5 [84], GLM [123], XLM-E [20], ST-MoE [133], AlexaTM [95]
Decoder-only (GPT-style)	Training Model type: Pretrain task:	Autoregressive Language Models Generative Predict next word	GPT-3 [16], OPT [126]. PaLM [22], BLOOM [92], MT-NLG [93], GLaM [32],Gopher [83], chinchilla [41], LaMDA [102], GPT-J [107], LLaMA [103], GPT-4 [76], BloombergGPT [117]

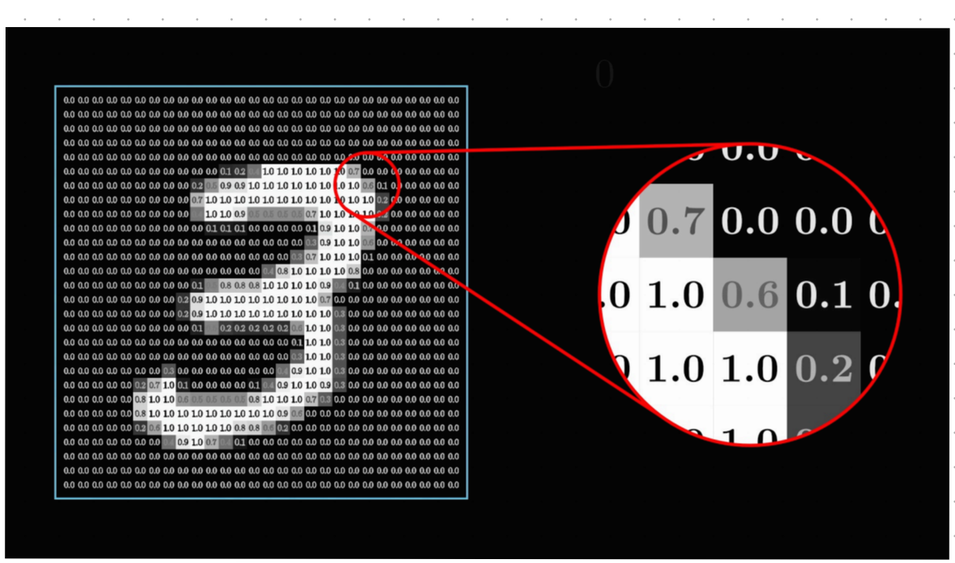




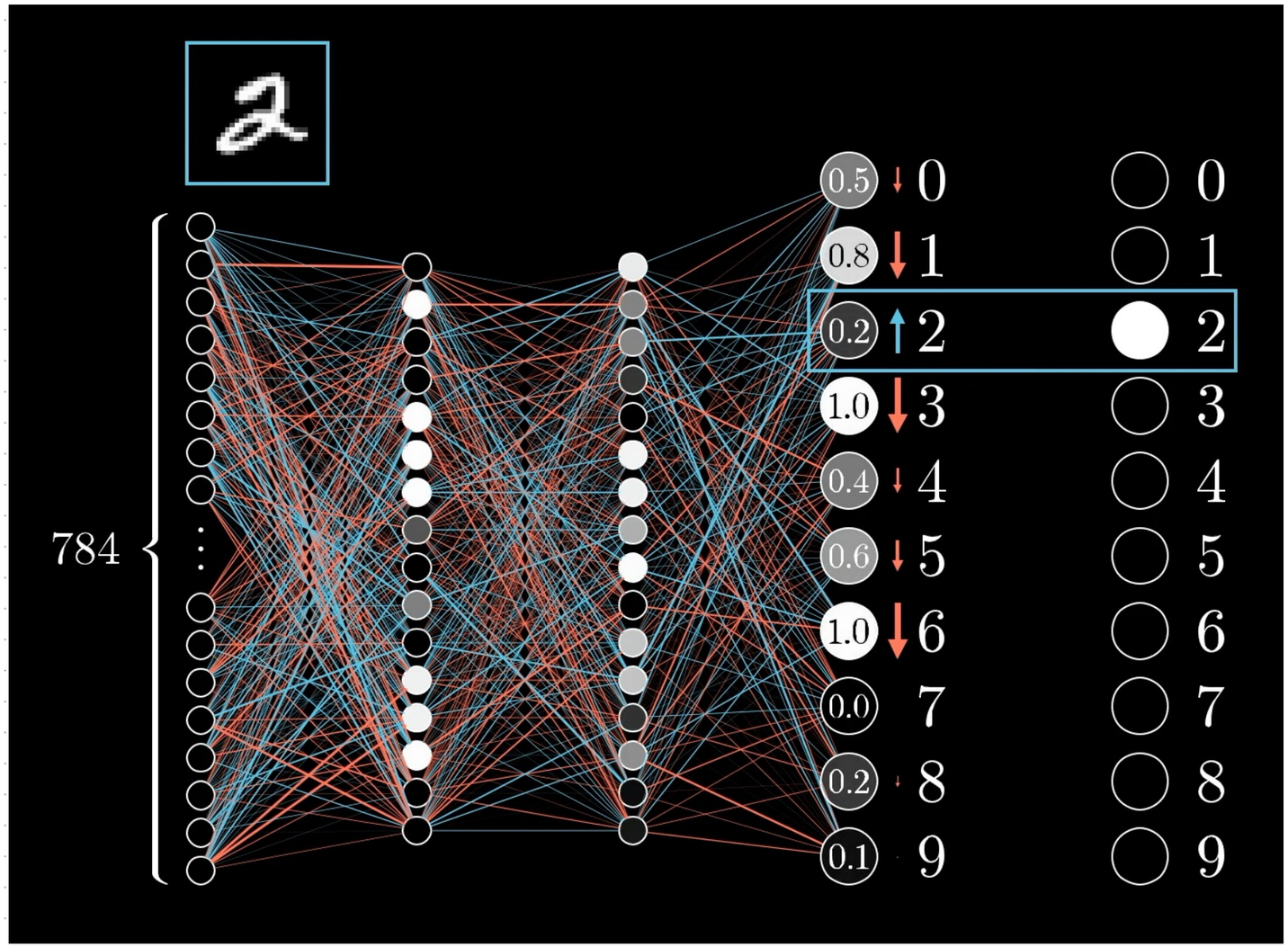
Let's get a feeling for how neural networks work

http://playground.tensorflow.org

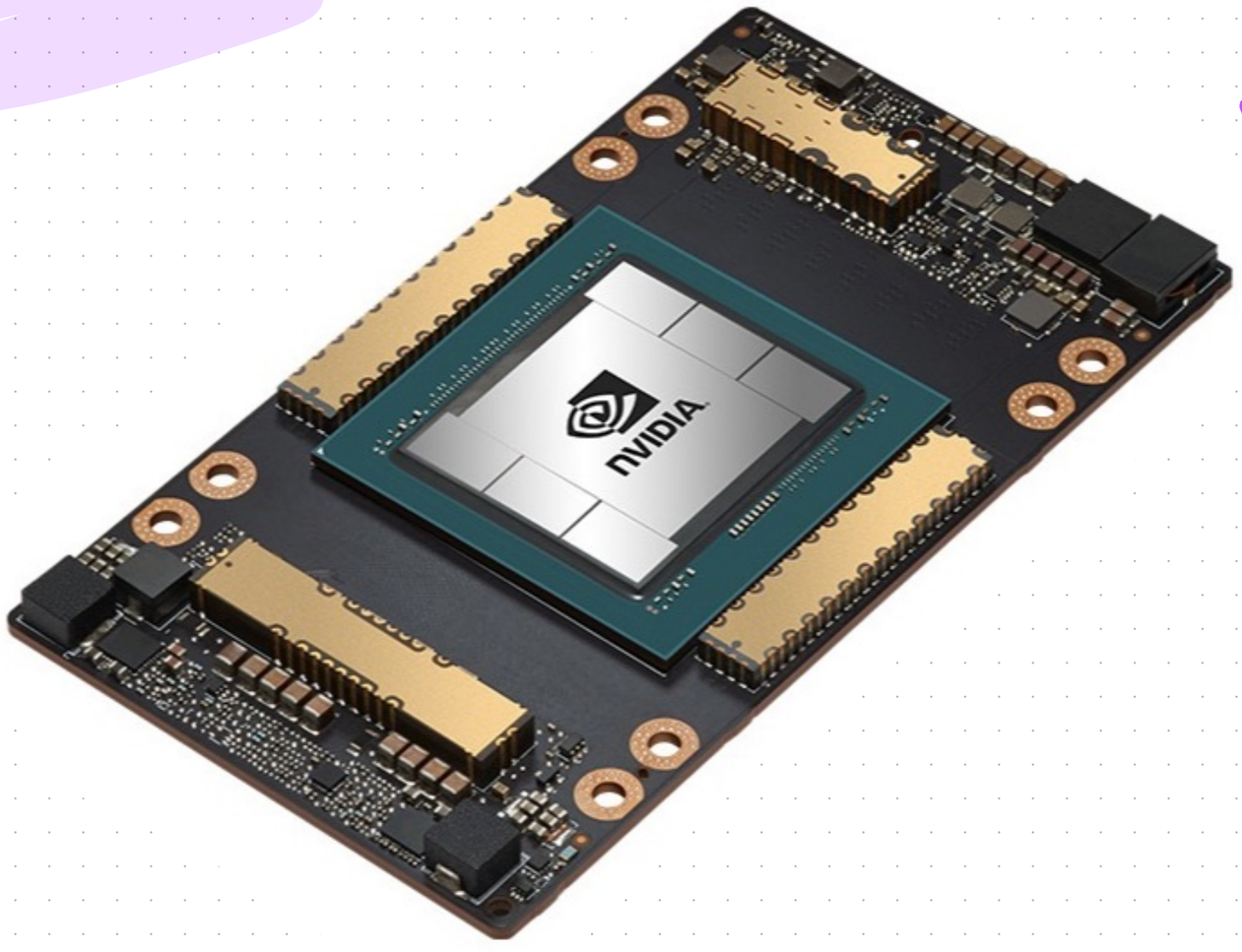
Most of the exciting advancements
in AI recently use "deep" neural networks



3b1b link



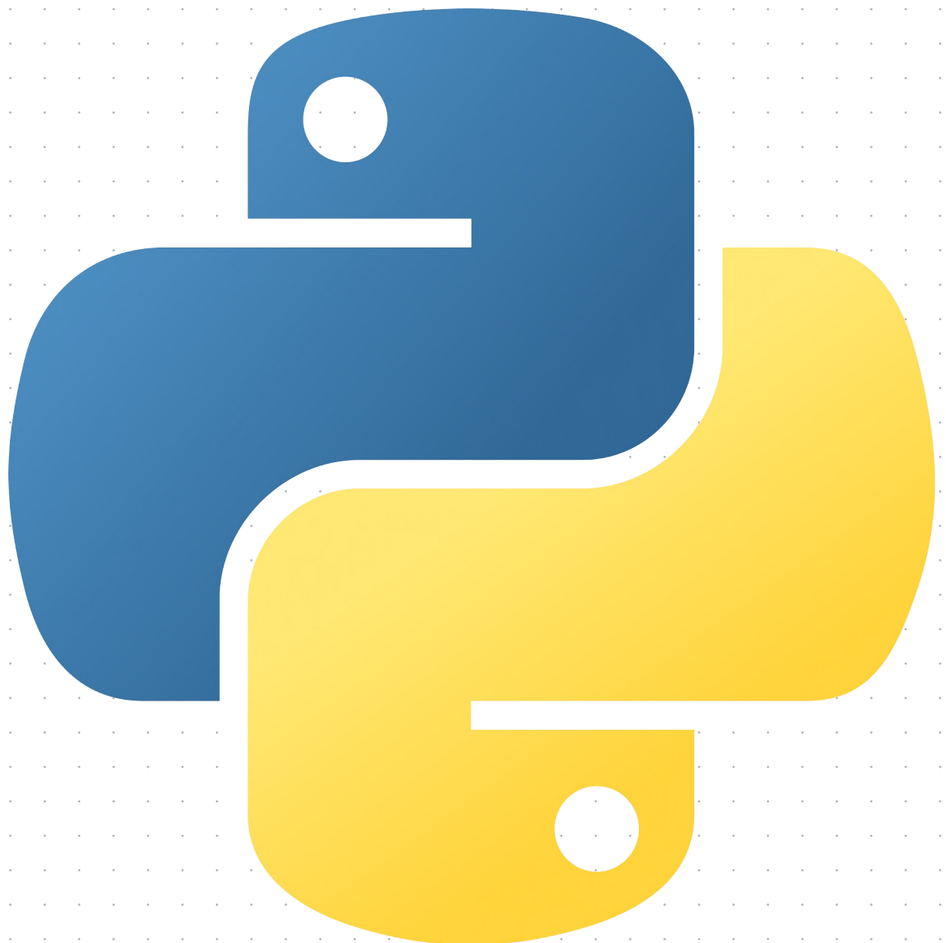
"GPUs" are what make deep learning & LLMs possible*



6,912 cores on this thing!

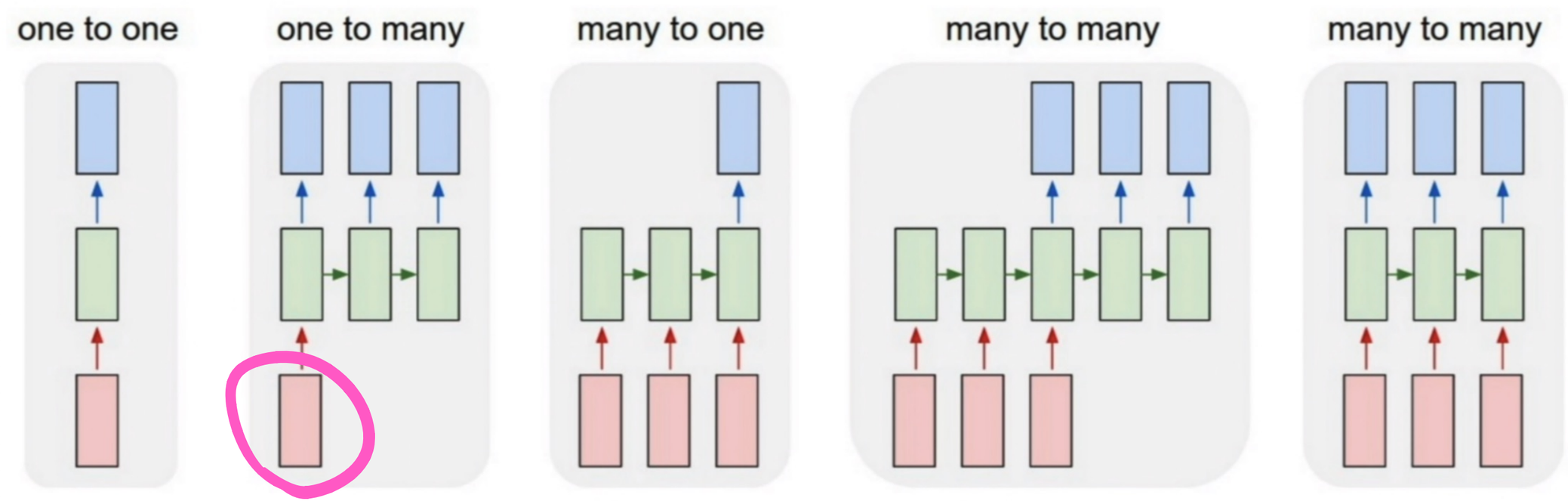
* also lets your kid play high-res minecraft

Let's write a neural network to classify handwritten digits



http://ygelp.kyle.pub/n1

RNNs are useful

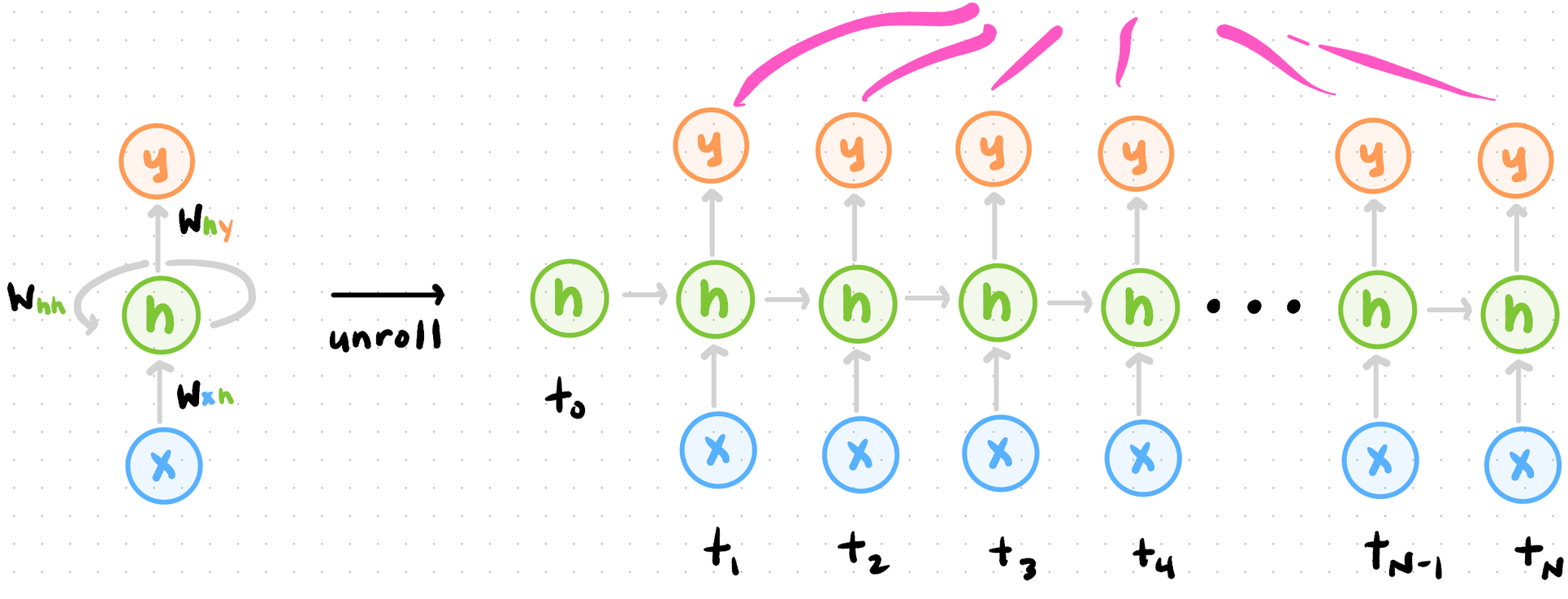


RNN Internals

$$h_t = \sigma\left(W_{hh}\, h_{t-1} + W_{xh}\, x_t + B_h\right)$$

$$y_t = \sigma\left(W_{hy}\, h_t + B_y\right)$$

Loss



RNNs < Transformers mostly

The "T" in ChatGPT is "Transformer"

RNNs were state of the art

But they're a PITA to train

Transformers work better

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [35, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [38, 24, 15].

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t-1} and the input for position t. This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks [21] and conditional computation [32], while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [2, 19]. In all but a few cases [27], however, such attention mechanisms are used in conjunction with a recurrent network.

In this work we propose the Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. The Transformer allows for significantly more parallelization and can reach a new state of the art in translation quality after being trained for as little as twelve hours on eight P100 GPUs.

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

Łukasz Kaiser[*] Google Brain lukaszkaiser@google.com

Illia Polosukhin[*] [‡] illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

[*]Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and

Eagle-7B

RNNs are maybe not done yet!

arXiv:2305.13048v2 [cs.CL] 11 Dec 2023

RWKV: Reinventing RNNs for the Transformer Era

Bo Peng[1,2]* **Eric Alcaide**[2,3,4]* **Quentin Anthony**[2,5]*
Alon Albalak[2,6] **Samuel Arcadinho**[2,7] **Stella Biderman**[2,8] **Huanqi Cao**[9] **Xin Cheng**[10]
Michael Chung[11] **Xingjian Du**[1] **Matteo Grella**[12] **Kranthi Kiran GV**[2,13] **Xuzheng He**[2]
Haowen Hou[14] **Jiaju Lin**[1] **Przemysław Kazienko**[15] **Jan Kocoń**[15] **Jiaming Kong**[16]
Bartłomiej Koptyra[15] **Hayden Lau**[2] **Krishna Sri Ipsit Mantri**[17] **Ferdinand Mom**[18,19]
Atsushi Saito[2,20] **Guangyu Song**[21] **Xiangru Tang**[22] **Bolun Wang**[23] **Johan S. Wind**[24]
Stanisław Woźniak[15] **Ruichong Zhang**[9] **Zhenyuan Zhang**[2] **Qihang Zhao**[25,26]
Peng Zhou[23] **Qinghua Zhou**[5] **Jian Zhu**[27] **Rui-Jie Zhu**[28,29]

[1]Generative AI Commons [2]EleutherAI [3]U. of Barcelona [4]Charm Therapeutics [5]Ohio State U. [6]U. of C., Santa Barbara
[7]Zendesk [8]Booz Allen Hamilton [9]Tsinghua University [10]Peking University [11]Storyteller.io [12]Crisis24 [13]New York U.
[14]National U. of Singapore [15]Wroclaw U. of Science and Technology [16]Databaker Technology [17]Purdue U. [18]Criteo AI Lab
[19]Epita [20]Nextremer [21]Moves [22]Yale U. [23]RuoxinTech [24]U. of Oslo [25]U. of Science and Technology of China
[26]Kuaishou Technology [27]U. of British Columbia [28]U. of C., Santa Cruz [29]U. of Electronic Science and Technology of China

Abstract

Transformers have revolutionized almost all natural language processing (NLP) tasks but suffer from memory and computational complexity that scales quadratically with sequence length. In contrast, recurrent neural networks (RNNs) exhibit linear scaling in memory and computational requirements but struggle to match the same performance as Transformers due to limitations in parallelization and scalability. We propose a novel model architecture, Receptance Weighted Key Value (RWKV), that combines the efficient parallelizable training of transformers with the efficient inference of RNNs.

Our approach leverages a linear attention mechanism and allows us to formulate the model as either a Transformer or an RNN, thus parallelizing computations during training and maintains constant computational and memory complexity during inference. We scale our models as large as 14 billion parameters, by far the largest dense RNN ever trained, and find RWKV performs on par with similarly sized Transformers, suggesting future work can leverage this architecture to create more efficient models. This work presents a significant step towards reconciling trade-offs between computational efficiency and model performance in sequence processing tasks. [1]

1 Introduction

Deep learning has greatly advanced artificial intelligence, impacting a range of scientific and industrial processing tasks such as natural language understanding, conversational AI, time-series analysis, and indirectly sequential formats like images and graphs (Brown et al., 2020; Ismail Fawaz et al., 2019; Wu et al., 2020; Albalak et al., 2022). Predominant among these techniques include RNNs and Transformers (Vaswani et al., 2017), each with specific benefits and drawbacks. RNNs require less memory, particularly for handling long sequences. However, they suffer from the vanishing gradient problem and non-parallelizability in the time dimension during training, limiting their scalability (Hochreiter, 1998; Le and Zuidema, 2016).

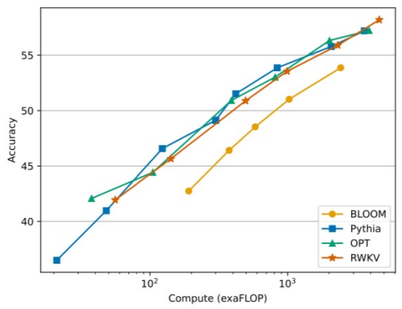


Figure 1: Average performance of RWKV models compared to transformers across twelve NLP tasks. For further details, see section 5.

Transformers emerged as a powerful alternative,

Model	Time	Space
Transformer	$O(T^2d)$	$O(T^2 + Td)$
Reformer	$O(T \log Td)$	$O(T \log T + Td)$
Performer	$O(Td^2 \log d)$	$O(Td \log d + d^2 \log d)$
Linear Transformers	$O(Td^2)$	$O(Td + d^2)$
AFT-full	$O(T^2d)$	$O(Td)$
AFT-local	$O(Tsd)$	$O(Td)$
MEGA	$O(cTd)$	$O(cd)$
RWKV (ours)	$O(\mathbf{Td})$	$O(\mathbf{d})$

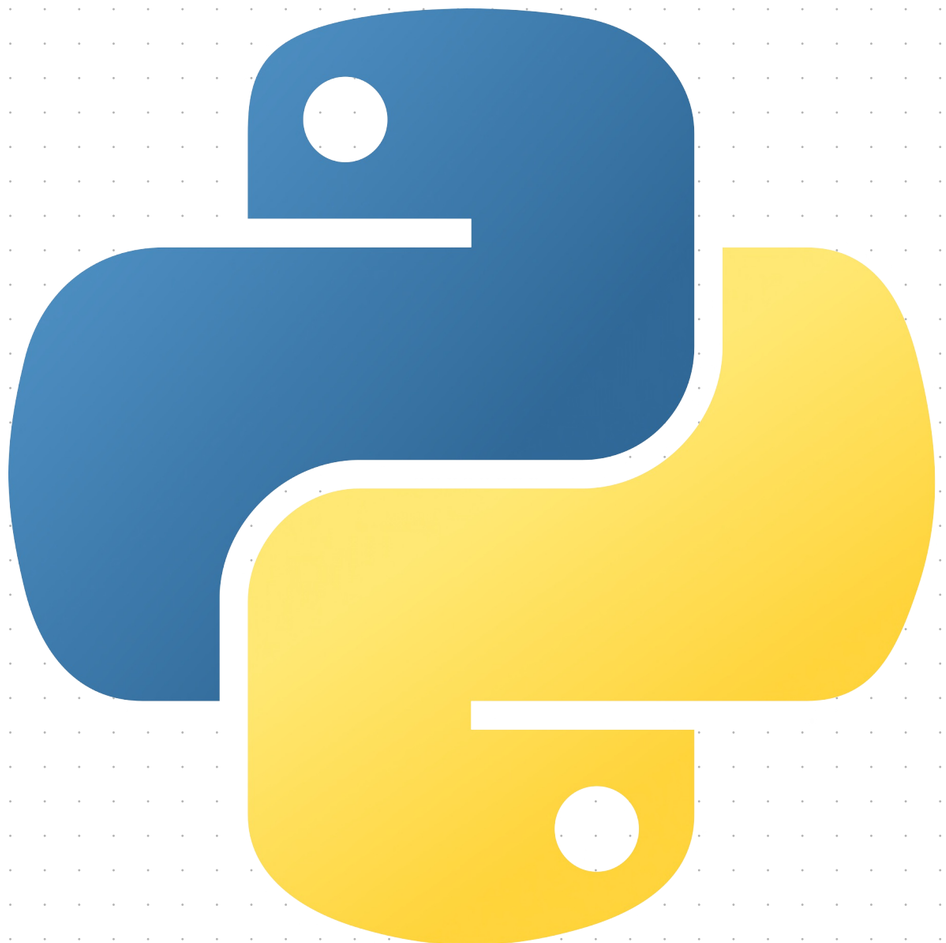
nice scalability!

Table 1: Inference complexity comparison with different Transformers. Here T denotes the sequence length, d the feature dimension, c is MEGA's chunk size of quadratic attention, and s is the size of a local window for AFT.

beats similar-sized transformer models

Model	params	Tokens	MultiLang	xLAMBADA	xStoryCloze	xWinograd	xCOPA
				acc	acc	acc	acc
RWKV-5 "Eagle" World-v2 ctx4k	7.52 B	1.1 T	61.4%	47.8%	62.1%	73.5%	62.4%
RWKV-4 "Dove" World-v1 ctx4k	7.52 B	590 B	57.4%	41.6%	59.5%	68.7%	60.0%
Mistral-7B-v0.1	7.24 B	2 - 7 T ?	58.2%	46.0%	57.2%	73.8%	55.8%
Llama-2-7b	6.74 B	2 T	56.6%	45.0%	55.6%	69.4%	56.7%
falcon-7b	6.92 B	1.5 T	55.8%	45.5%	53.8%	68.0%	56.0%
mpt-7b-8k	6.65 B	1T	55.6%	44.3%	55.3%	67.9%	55.1%
open_llama_7b_v2	6.74 B	1T	54.0%	40.9%	53.9%	66.6%	54.5%
RedPajama-INCITE-7B-Base	6.86 B	1T	51.2%	37.2%	53.3%	61.7%	52.6%
pythia-6.9b-v0	6.86 B	300 B	50.8%	35.2%	52.1%	61.6%	54.4%

Let's teach an RNN to write (shoddy) Shakespeare



http://ygelp.kyle.pub/n2

Neural networks be like...



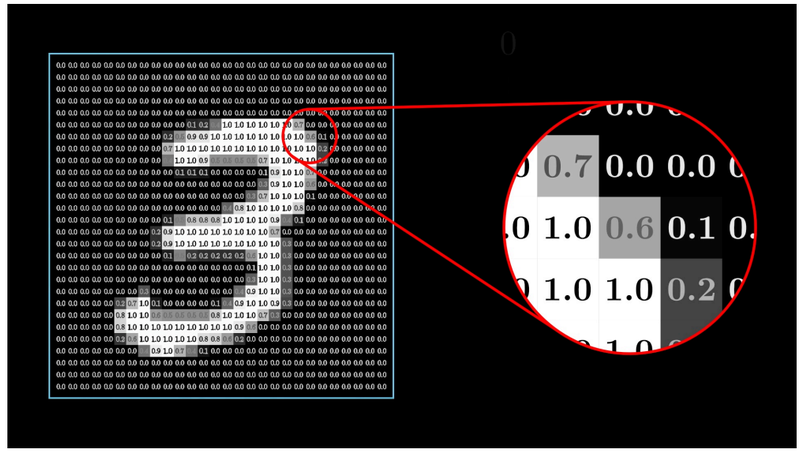
Text

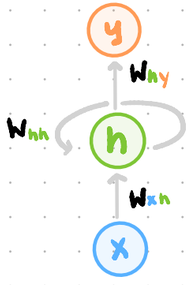
Numbers

"embeddings" is how we turn text → numbers

...how we got numbers in the previous section on RNNs

#1 - we had numbers ↓





#2 - We used "one-hot" vectors ↓

```
python       = [1,0,0,0,0,0,0,0,0,0,0,0]
consistently = [0,1,0,0,0,0,0,0,0,0,0,0]
ranks        = [0,0,1,0,0,0,0,0,0,0,0,0]
as           = [0,0,0,1,0,0,0,0,0,0,0,0]
one          = [0,0,0,0,1,0,0,0,0,0,0,0]
of           = [0,0,0,0,0,1,0,0,0,0,0,0]
the          = [0,0,0,0,0,0,1,0,0,0,0,0]
most         = [0,0,0,0,0,0,0,1,0,0,0,0]
popular      = [0,0,0,0,0,0,0,0,1,0,0,0]
programming  = [0,0,0,0,0,0,0,0,0,1,0,0]
languages    = [0,0,0,0,0,0,0,0,0,0,1,0]
```

12

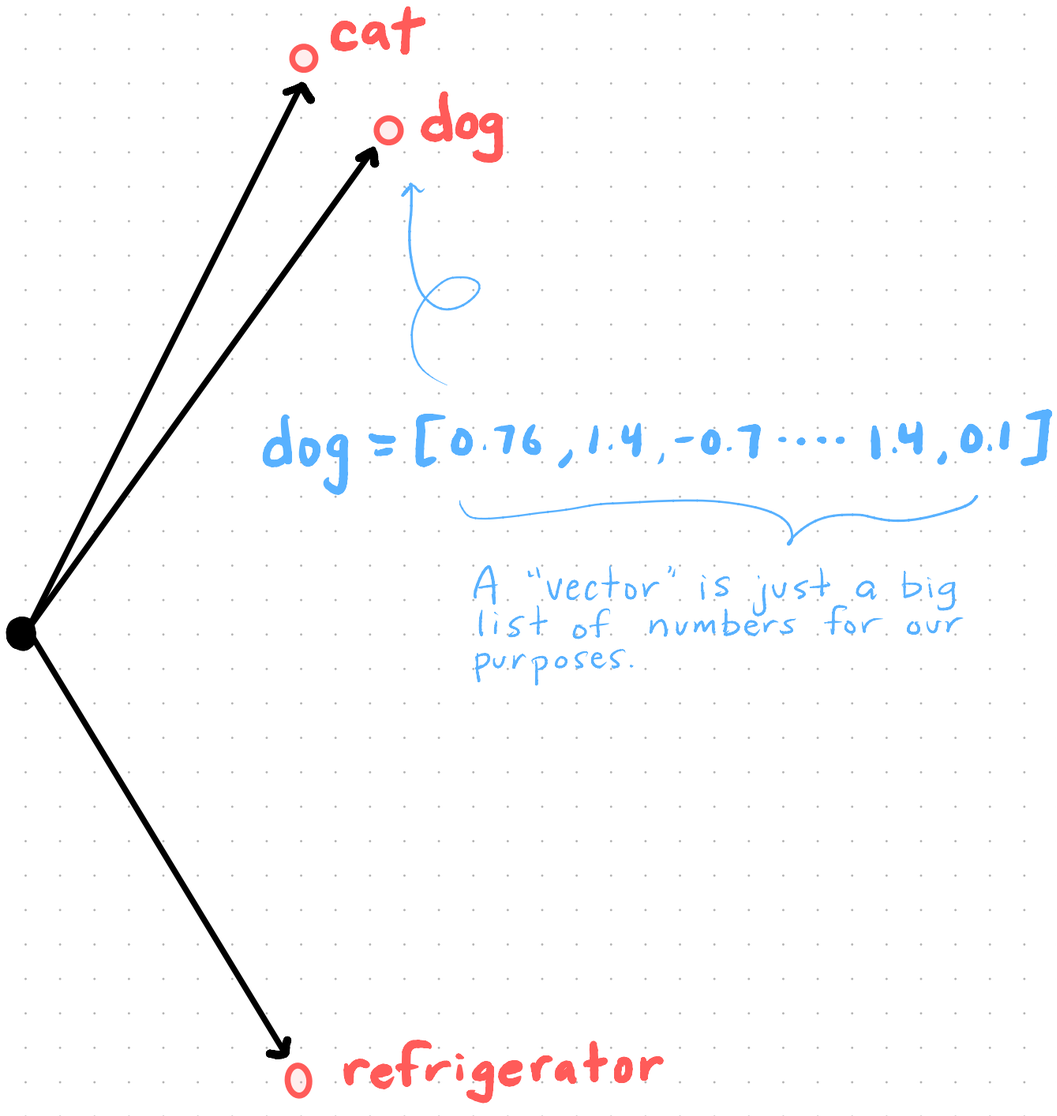
#3 - Now we want "dense" vectors ↓

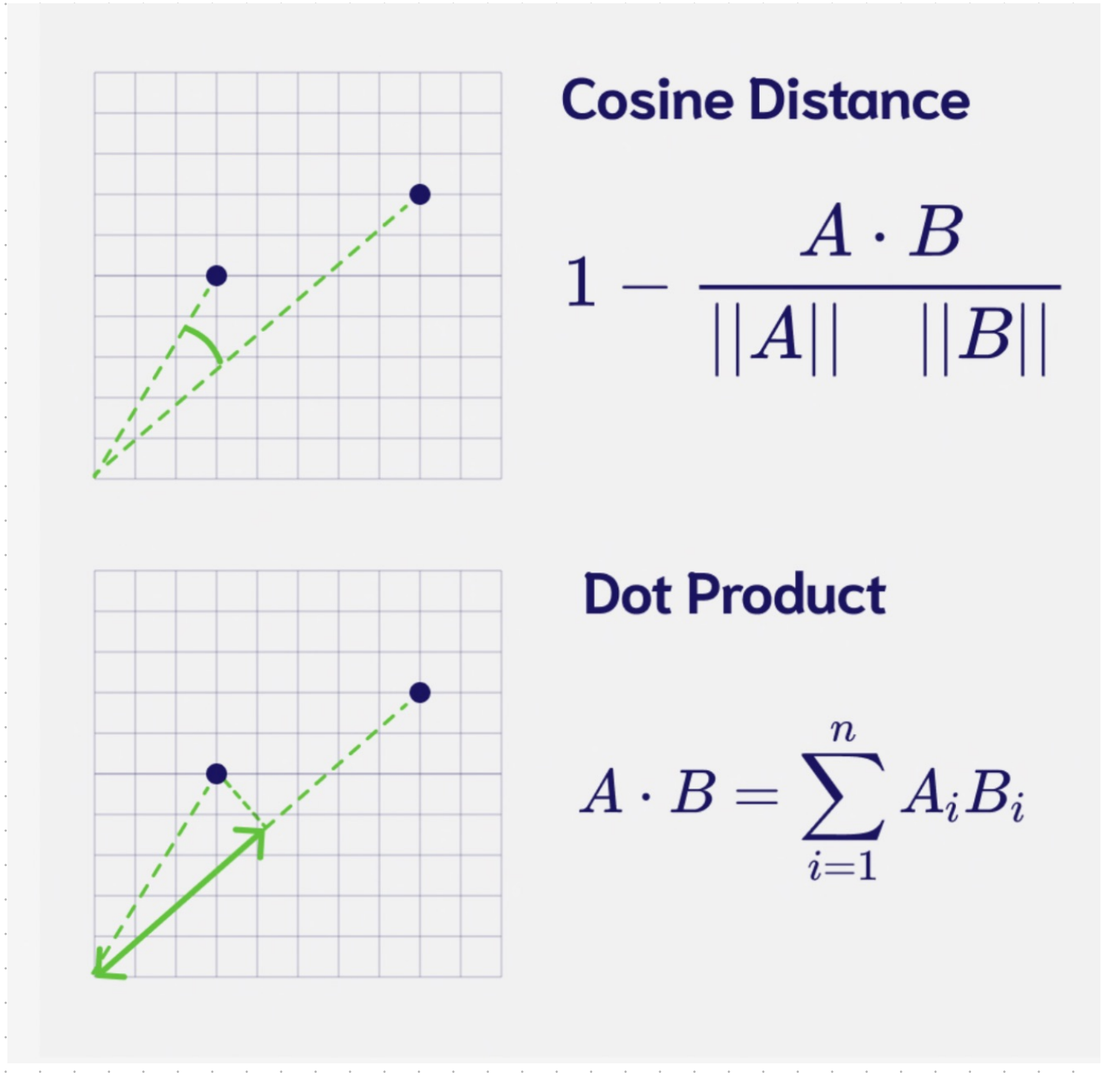
```
python       = [ 0.85, -0.24,  0.67,  0.12, -0.01]
consistently = [ 0.15,  0.30, -0.75,  0.60,  0.00]
ranks        = [ 0.48,  0.10,  0.50, -0.85,  0.20]
as           = [-0.25,  0.98,  0.05,  0.40, -0.15]
one          = [-0.10, -0.50,  0.45,  0.25,  0.65]
of           = [ 0.30, -0.10, -0.20,  0.50,  0.70]
the          = [ 0.00,  0.00,  0.00,  0.01,  0.99]
most         = [ 0.40, -0.55,  0.35,  0.15, -0.45]
popular      = [ 0.60, -0.30,  0.25, -0.70,  0.35]
programming  = [-0.45,  0.40, -0.65,  0.50, -0.30]
languages    = [ 0.50,  0.20, -0.10, -0.60,  0.40]
```

5

LLMs use dense embeddings/ vectors to represent words...

... dense vectors capture word relatedness





Word2vec was published in 2013

It received four "strong rejects"

despite 70% acceptance rate!

It won the NeurIPS "test of time" award on its 10th anniversary

40k citations!

arXiv:1301.3781v3 [cs.CL] 7 Sep 2013

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

1 Introduction

Many current NLP systems and techniques treat words as atomic units - there is no notion of similarity between words, as these are represented as indices in a vocabulary. This choice has several good reasons - simplicity, robustness and the observation that simple models trained on huge amounts of data outperform complex systems trained on less data. An example is the popular N-gram model used for statistical language modeling - today, it is possible to train N-grams on virtually all available data (trillions of words [3]).

However, the simple techniques are at their limits in many tasks. For example, the amount of relevant in-domain data for automatic speech recognition is limited - the performance is usually dominated by the size of high quality transcribed speech data (often just millions of words). In machine translation, the existing corpora for many languages contain only a few billions of words or less. Thus, there are situations where simple scaling up of the basic techniques will not result in any significant progress, and we have to focus on more advanced techniques.

With progress of machine learning techniques in recent years, it has become possible to train more complex models on much larger data set, and they typically outperform the simple models. Probably the most successful concept is to use distributed representations of words [10]. For example, neural network based language models significantly outperform N-gram models [1, 27, 17].

1.1 Goals of the Paper

The main goal of this paper is to introduce techniques that can be used for learning high-quality word vectors from huge data sets with billions of words, and with millions of words in the vocabulary. As far as we know, none of the previously proposed architectures has been successfully trained on more

OpenReview.net — Search OpenReview... — Login

← Go to **ICLR 2013 conference** homepage

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean

17 Jan 2013 ICLR 2013 conference submission Readers: Everyone Show Revisions

Abstract: We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computatio… vectors for one million vocabulary fr… the-art performance on our test set … used by the research community.

Decision: conferencePoster-iclr201…

Reply Type: all Author:

Hidden From: nobody

submission review b…
Pontus Stenetorp
10 May 2013 ICLR 2013 subm…
Review: In response to the req… similarity being intransitive, I th… believe to be the relevant portio…

`We say 'the portrait resemble… the father' rather than 'the fath… we say 'North Korea is like Red …

Lastly, a question that was raise… which the first author responde… case. I would be very thankful f…

submission review b…
anonymous reviewer 3c5e

facebook — Log

Tomas Mikolov
December 13, 2023 ·

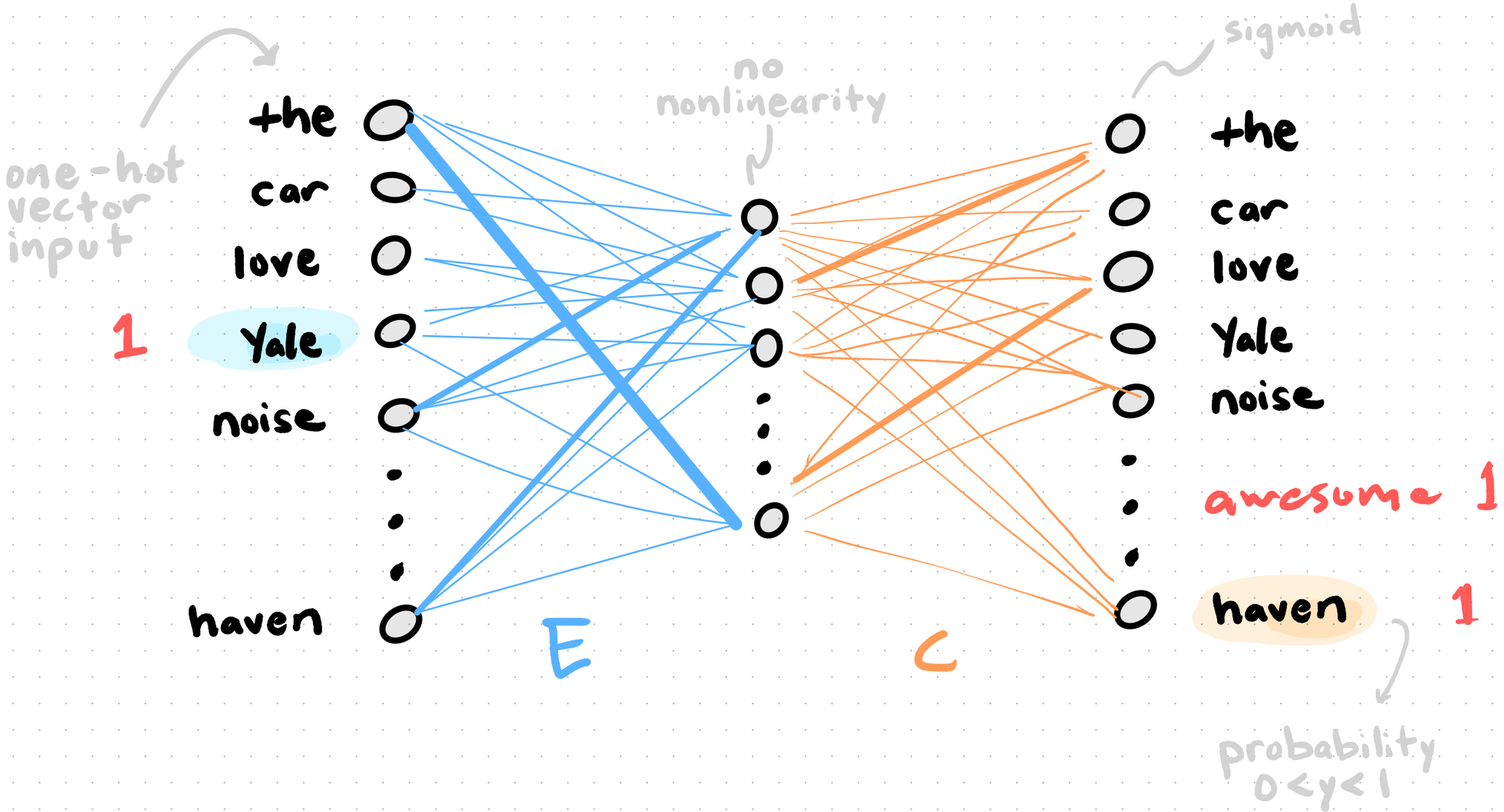
Yesterday we received a Test of Time Award at NeurIPS for the word2vec paper from ten years ago. I'm really happy about it! I think it's the first "best paper" type of award I ever received. In fact, the original word2vec paper was rejected at the first ICLR conference in 2013 (despite the acceptance rate of around 70%), so it made me think how difficult it is for reviewers to predict future impact of research papers.

I've heard a lot of comments - both positive and negative - about word2vec during those years, and did not really comment online about it. Somehow I felt the research community is constantly flooded by propaganda-style PR from certain researchers who are hacking this way the citation counts and attention of others, and I did not want to be part of this. But after ten years, I think it could be entertaining to share some stories associated with this paper.

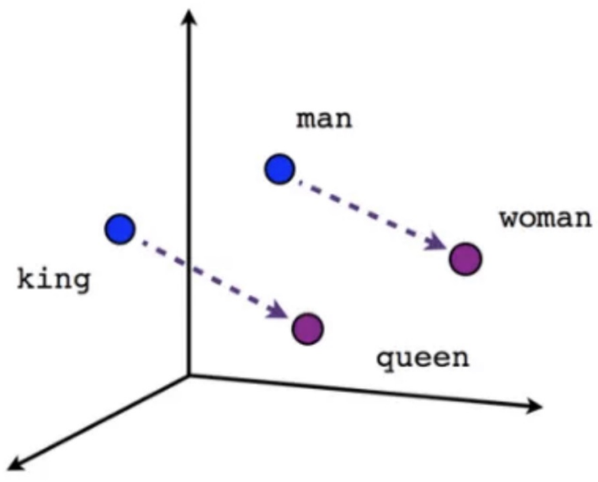
One frequent comment I've heard was that the code was difficult to understand to the point that some people thought I made it unreadable intentionally. But no, I'm not so evil 😀 The code ended up being over-optimized because I was waiting for many months for approval to publish it, and meanwhile I was trying to make it both faster and shorter. In fact, looking back, if there were not Greg and Jeff in the Brain team, I doubt I would ever get that approval - I think word2vec was likely the first widely known AI project that Google open-sourced.

There was also significant controversy around the GloVe project from Stanford NLP group that

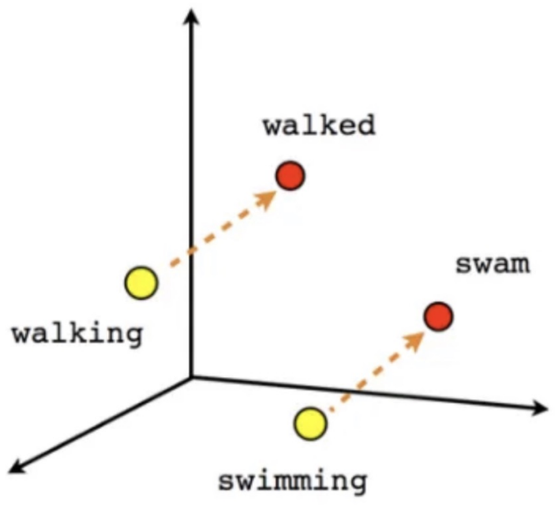
word2vec's Skipgram with negative sampling



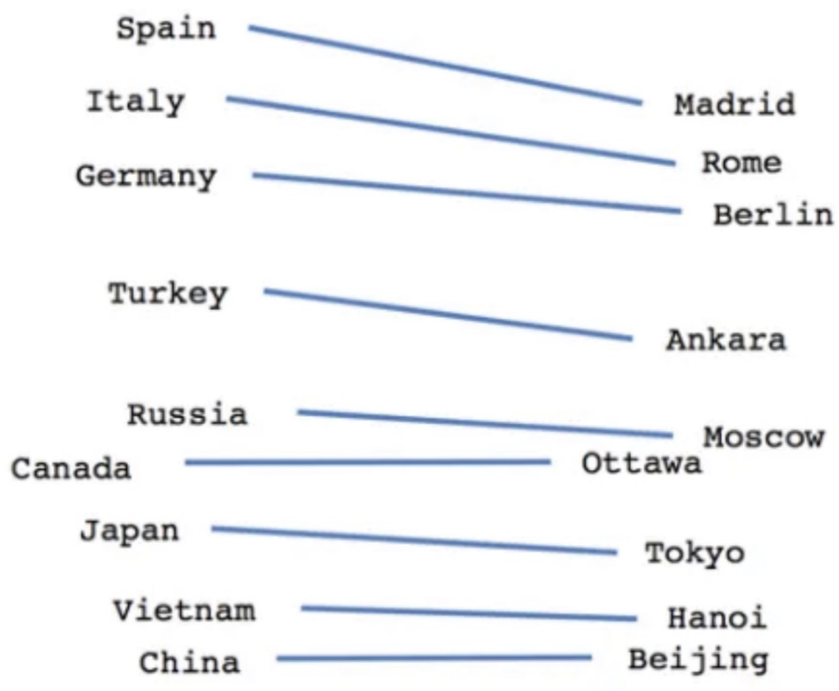
We can even do algebra with dense word embeddings



Male-Female

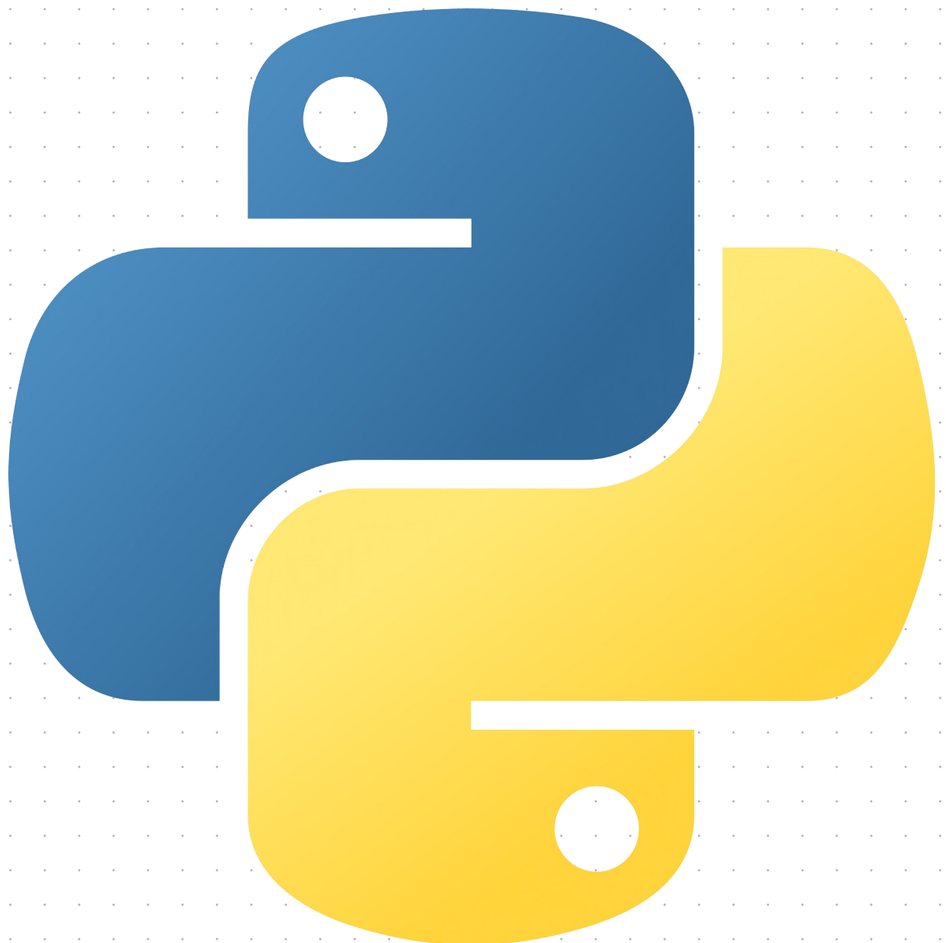


Verb tense



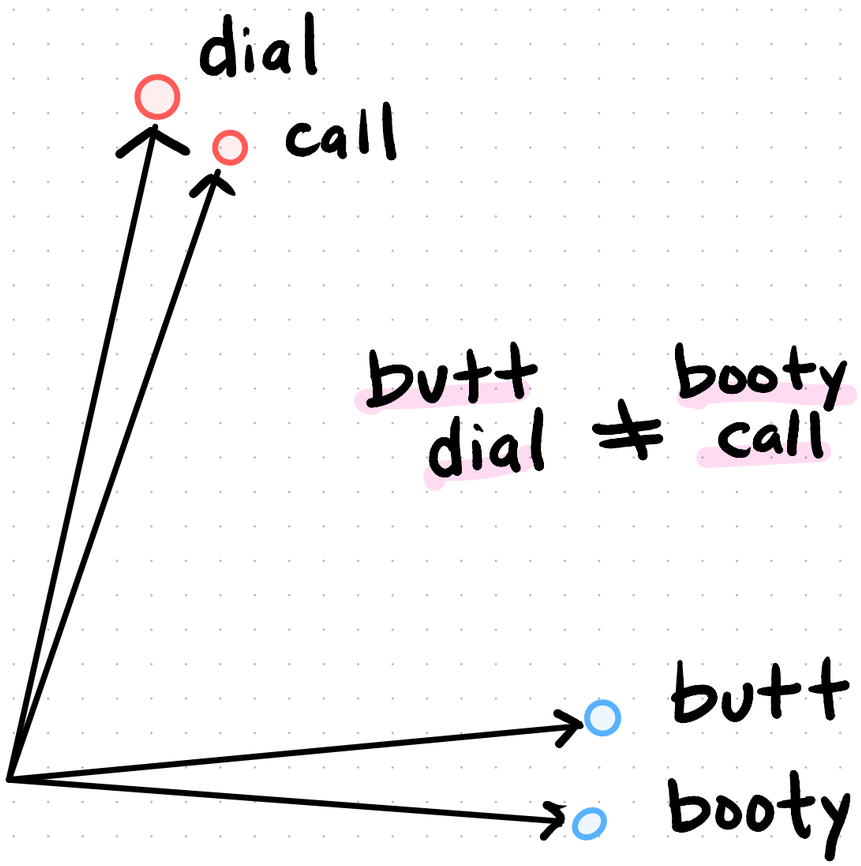
Country-Capital

Let's see some word embeddings

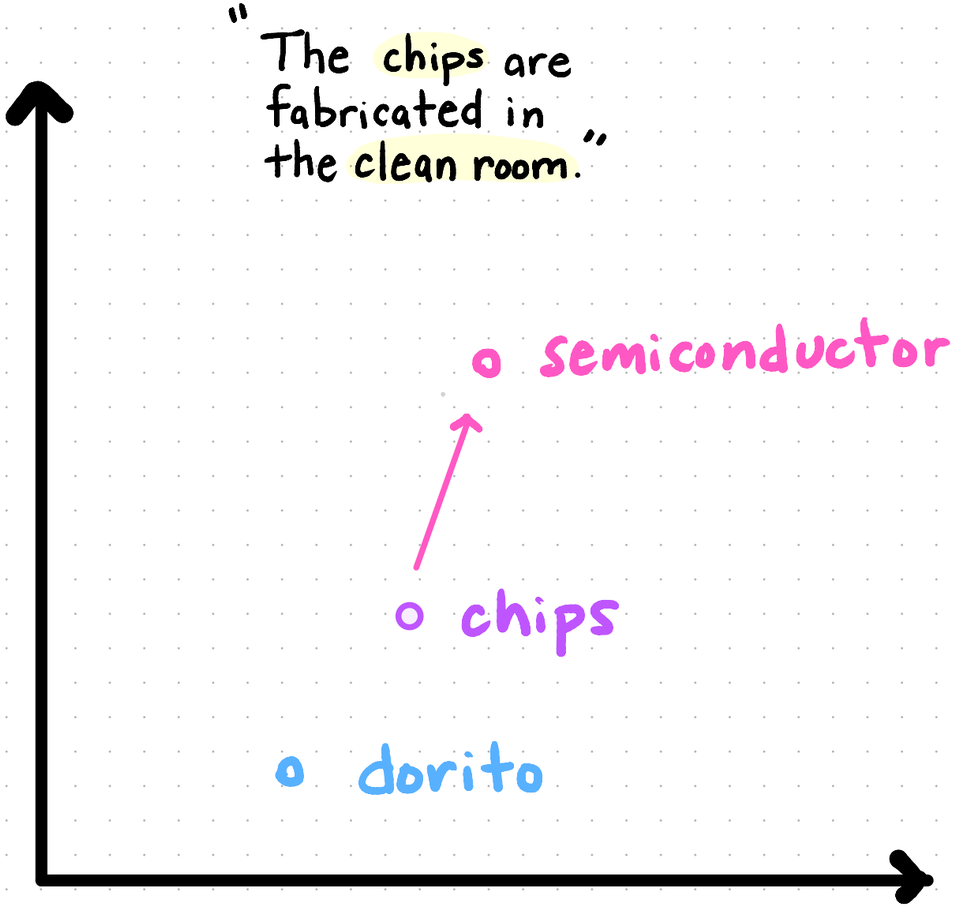


http://ygelp.kyle.pub/n3

But our embeddings are a bit "dumb"...

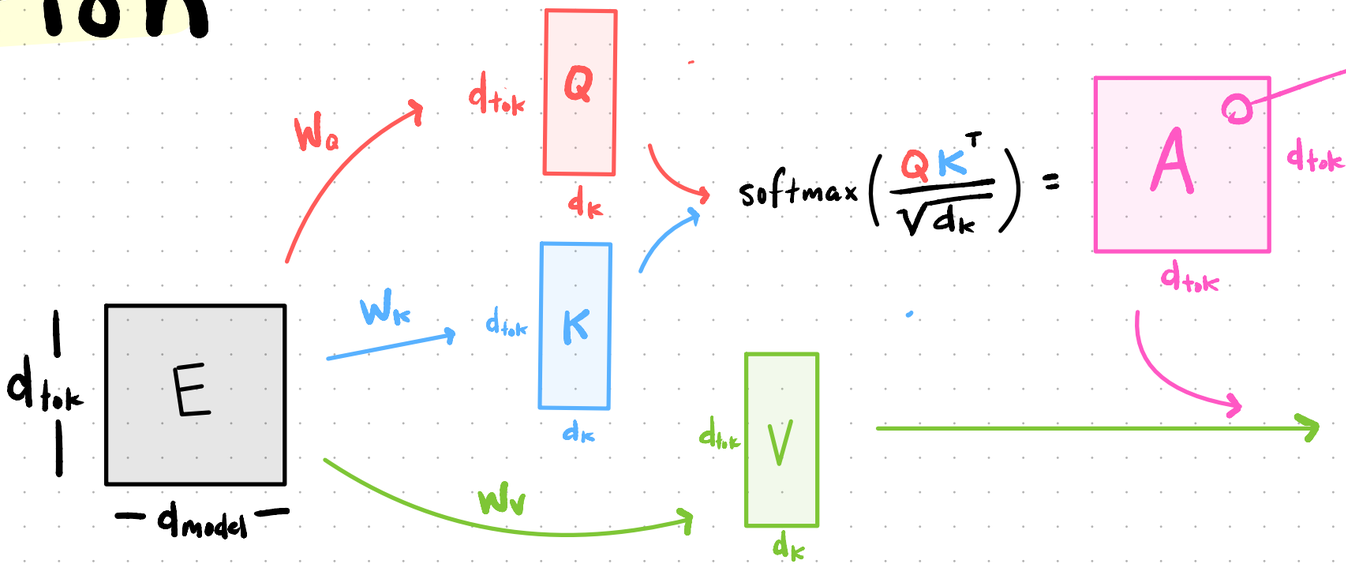


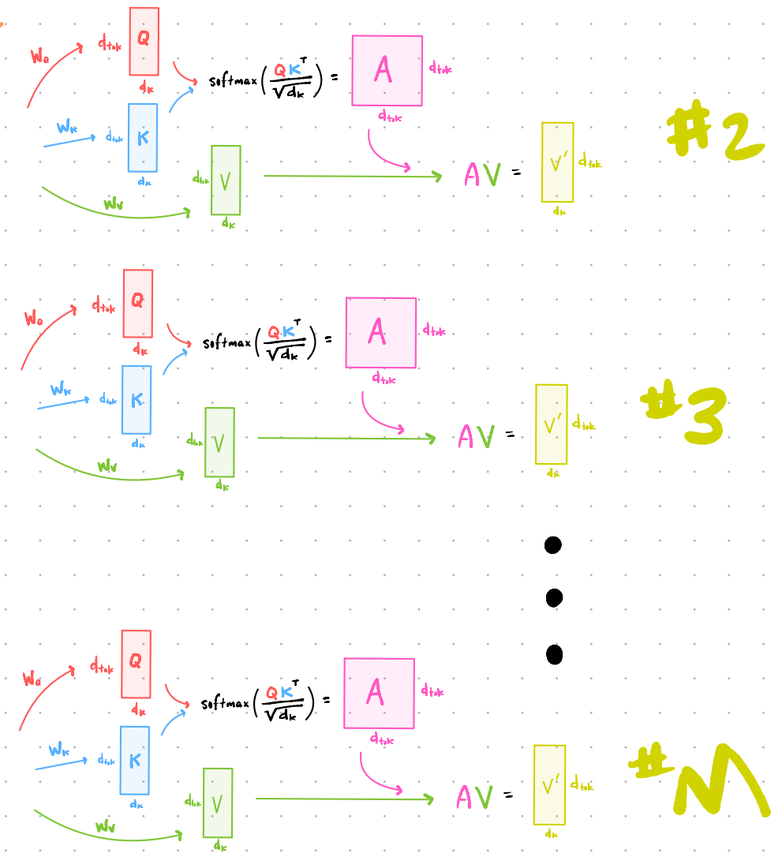
..."attention" helps them become "context aware"

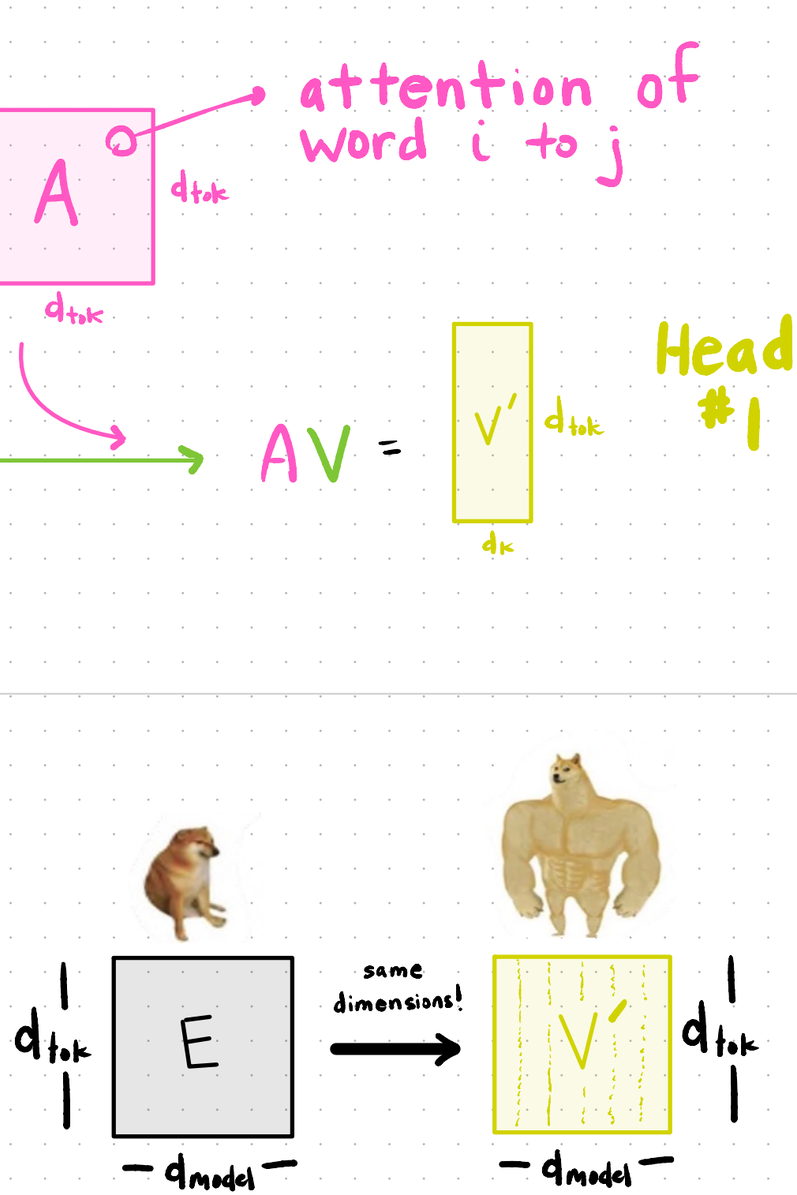


Attention



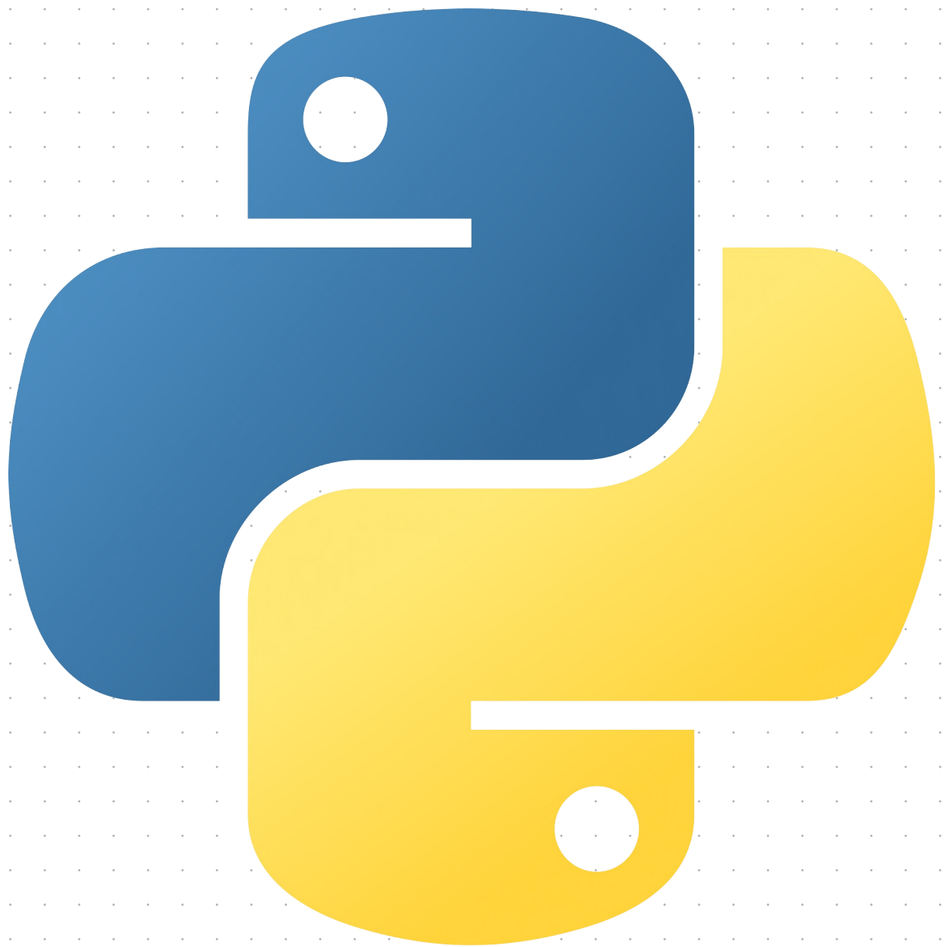






Net effect = context aware embeddings

Let's visualize the GPT-2 attention mechanism



http://ygelp.kyle.pub/n4

Most LLMs are based on "transformers"

Google, 2017

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

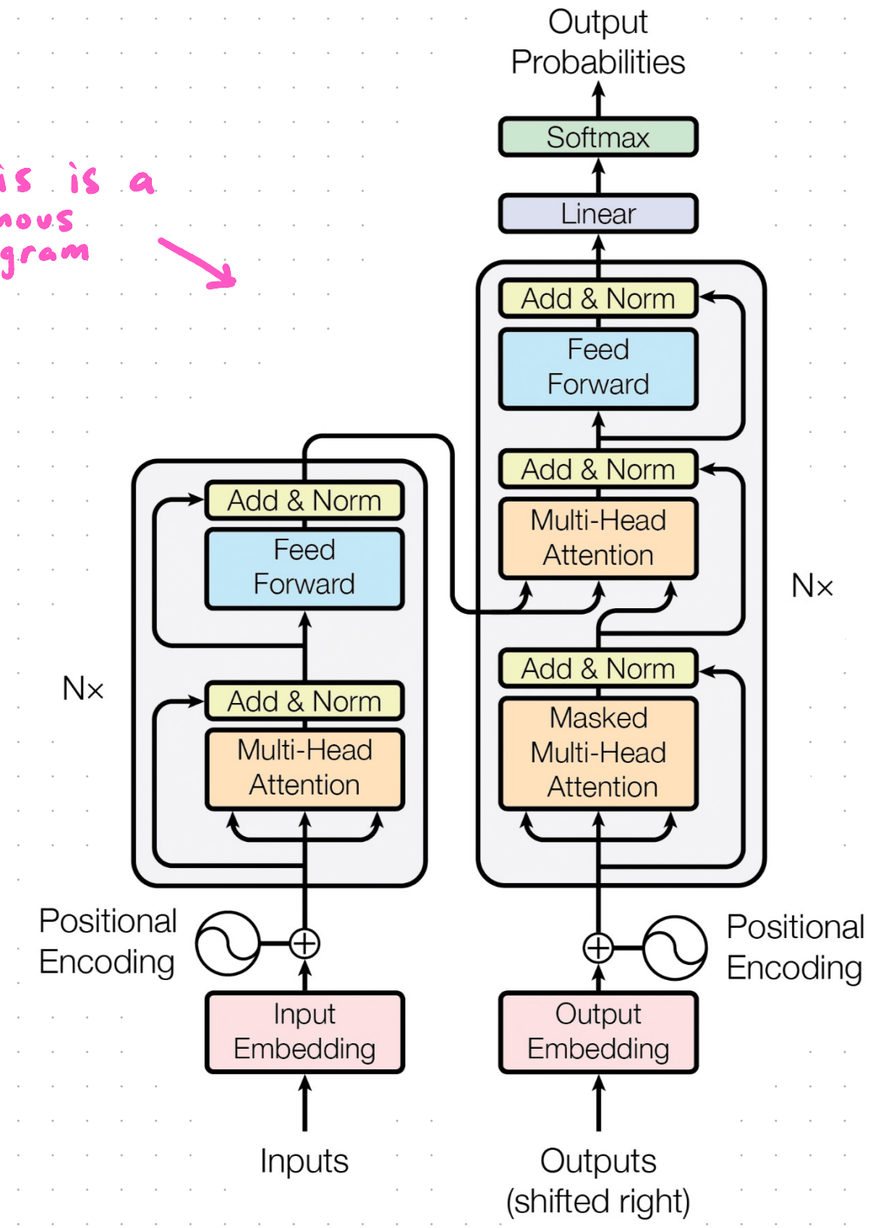
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

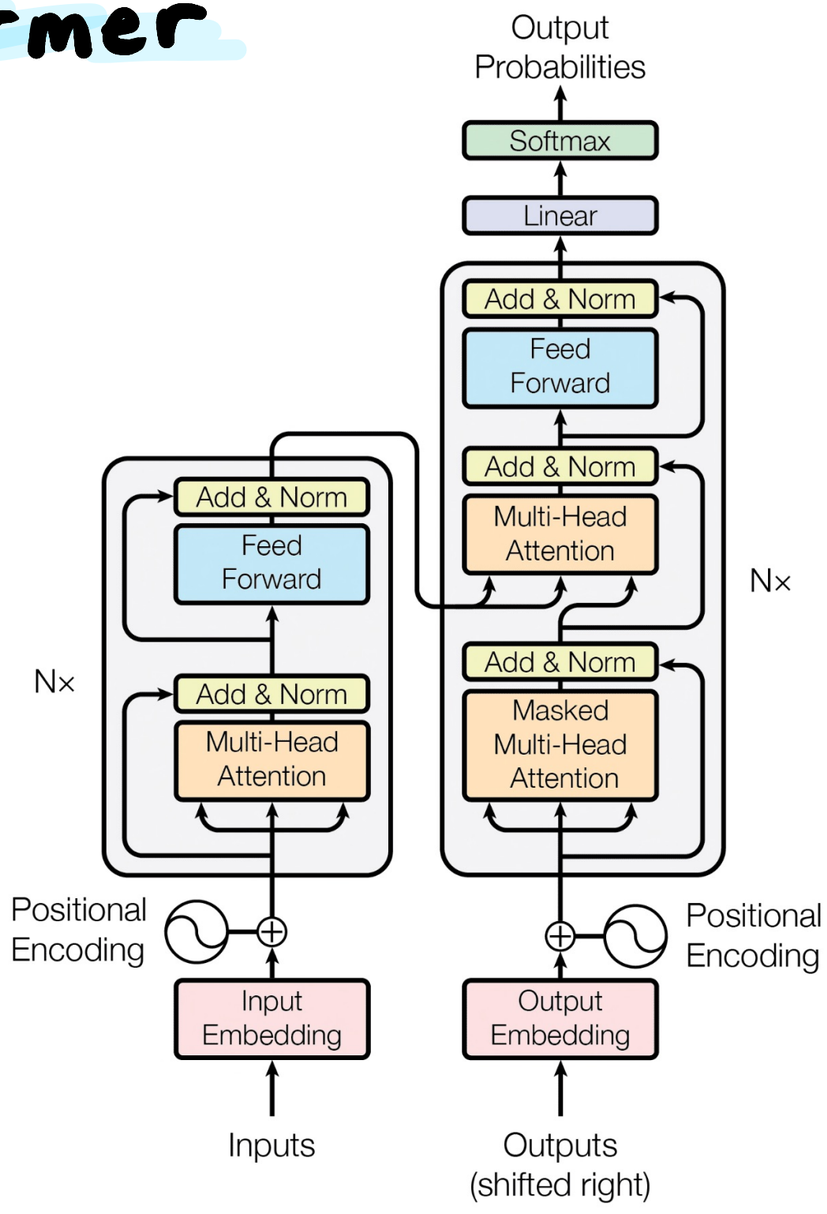
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

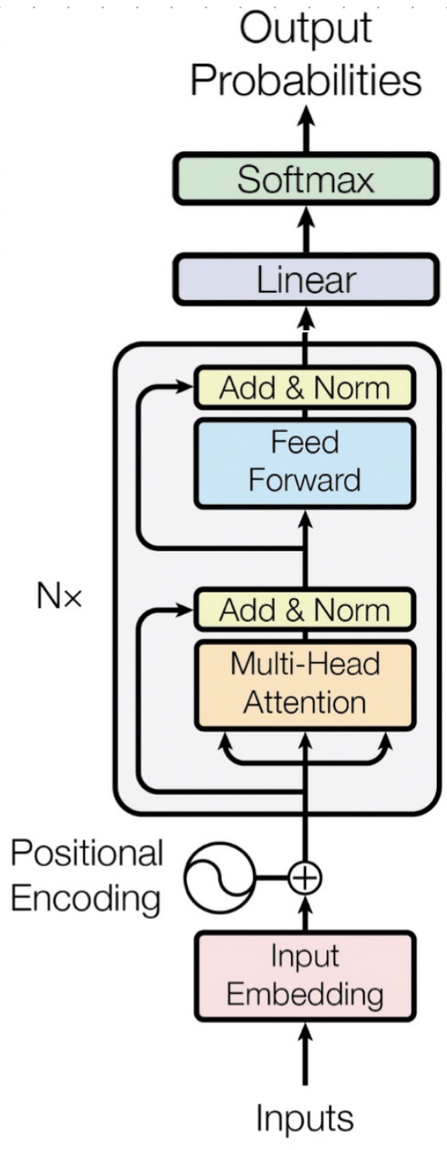
this is a famous diagram



The Transformer



encoder / decoder



decoder only

a look at Llama

We are releasing the following models to the general public for research and commercial use[‡]:

1. **Llama 2**, an updated version of Llama 1, trained on a new mix of publicly available data. We also increased the size of the pretraining corpus by 40%, doubled the context length of the model, and adopted grouped-query attention (Ainslie et al., 2023). We are releasing variants of Llama 2 with 7B, 13B, and 70B parameters. We have also trained 34B variants, which we report on in this paper but are not releasing.[§]

2. **Llama 2-Chat**, a fine-tuned version of Llama 2 that is optimized for dialogue use cases. We release variants of this model with 7B, 13B, and 70B parameters as well.

arXiv:2302.13971v1 [cs.CL] 27 Feb 2023

LLaMA: Open and Efficient Foundation Language

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Ma
Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Nan
Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand J
Edouard Grave, Guillaume Lample*

Meta AI

Abstract

We introduce LLaMA, a collection of foundation language models ranging from 7B to 65B parameters. We train our models on trillions of tokens, and show that it is possible to train state-of-the-art models using publicly available datasets exclusively, without resorting to proprietary and inaccessible datasets. In particular, LLaMA-13B outperforms GPT-3 (175B) on most benchmarks, and LLaMA-65B is competitive with the best models, Chinchilla-70B and PaLM-540B. We release all our models to the research community[1].

1 Introduction

Large Languages Models (LLMs) trained on massive corpora of texts have shown their ability to perform new tasks from textual instructions or from a few examples (Brown et al., 2020). These few-shot properties first appeared when scaling models to a sufficient size (Kaplan et al., 2020), resulting in a line of work that focuses on further scaling these models (Chowdhery et al., 2022; Rae et al., 2021). These efforts are based on the assumption that more parameters will lead to better performance. However, recent work from Hoffmann et al. (2022) shows that, for a given compute budget, the best performances are not achieved by the largest models, but by smaller models trained on more data.

The objective of the scaling laws from Hoffmann et al. (2022) is to determine how to best scale the dataset and model sizes for a particular *training* compute budget. However, this objective disregards the *inference* budget, which becomes critical when serving a language model at scale. In this context, given a target level of performance, the preferred model is not the fastest to train but the

performance, a smaller one
ultimately be cheaper at infe
although Hoffmann et al.
training a 10B model on 2
that the performance of a 7B
improve even after 1T tokens

The focus of this work is
language models that achieve
formance at various inference
on more tokens than what is
resulting models, called *LLa*
to 65B parameters with com
compared to the best existing
LLaMA-13B outperforms G
marks, despite being 10× sm
this model will help democr
study of LLMs, since it can be
At the higher-end of the scale
model is also competitive wi
guage models such as Chinch

Unlike Chinchilla, PaLM,
use publicly available data, m
patible with open-sourcing,
models rely on data which i
available or undocumented (e
"Social media conversations"
exceptions, notably OPT (Z
GPT-NeoX (Black et al., 20
et al., 2022) and GLM (Zeng
that are competitive with PaL

In the rest of this paper, we
of the modifications we mad
architecture (Vaswani et al.,
training method. We then repo
our models and compare with

arXiv:2307.09288v2 [cs.CL] 19 Jul 2023

Llama 2: Open Foundation and Fine-Tuned Chat Models

Hugo Touvron* Louis Martin† Kevin Stone†
Peter Albert Amjad Almahairi Yasmine Babaei Nikolay Bashlykov Soumya Batra
Prajjwal Bhargava Shruti Bhosale Dan Bikel Lukas Blecher Cristian Canton Ferrer Moya Chen
Guillem Cucurull David Esiobu Jude Fernandes Jeremy Fu Wenyin Fu Brian Fuller
Cynthia Gao Vedanuj Goswami Naman Goyal Anthony Hartshorn Saghar Hosseini Rui Hou
Hakan Inan Marcin Kardas Viktor Kerkez Madian Khabsa Isabel Kloumann Artem Korenev
Punit Singh Koura Marie-Anne Lachaux Thibaut Lavril Jenya Lee Diana Liskovich
Yinghai Lu Yuning Mao Xavier Martinet Todor Mihaylov Pushkar Mishra
Igor Molybog Yixin Nie Andrew Poulton Jeremy Reizenstein Rashi Rungta Kalyan Saladi
Alan Schelten Ruan Silva Eric Michael Smith Ranjan Subramanian Xiaoqing Ellen Tan Binh Tang
Ross Taylor Adina Williams Jian Xiang Kuan Puxin Xu Zheng Yan Iliyan Zarov Yuchen Zhang
Angela Fan Melanie Kambadur Sharan Narang Aurelien Rodriguez Robert Stojnic
Sergey Edunov Thomas Scialom*

GenAI, Meta

Abstract

In this work, we develop and release Llama 2, a collection of pretrained and fine-tuned large language models (LLMs) ranging in scale from 7 billion to 70 billion parameters. Our fine-tuned LLMs, called Llama 2-Chat, are optimized for dialogue use cases. Our models outperform open-source chat models on most benchmarks we tested, and based on our human evaluations for helpfulness and safety, may be a suitable substitute for closed-source models. We provide a detailed description of our approach to fine-tuning and safety improvements of Llama 2-Chat in order to enable the community to build on our work and contribute to the responsible development of LLMs.

*Equal contribution, corresponding authors: {tscialom, htouvron}@meta.com
†Second author

Contributions for all the authors can be found in Section A.1.

Meta

Training

Llama-1

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.

	Training Data	Params	Context Length	GQA	Tokens	LR
Llama 1	*See Touvron et al. (2023)*	7B	2k	✗	1.0T	3.0×10^{-4}
		13B	2k	✗	1.0T	3.0×10^{-4}
		33B	2k	✗	1.4T	1.5×10^{-4}
		65B	2k	✗	1.4T	1.5×10^{-4}
Llama 2	*A new mix of publicly available online data*	7B	4k	✗	2.0T	3.0×10^{-4}
		13B	4k	✗	2.0T	3.0×10^{-4}
		34B	4k	✓	2.0T	1.5×10^{-4}
		70B	4k	✓	2.0T	1.5×10^{-4}

Table 1: Llama 2 family of models. Token counts refer to pretraining data only. All models are trained with a global batch-size of 4M tokens. Bigger models — 34B and 70B — use Grouped-Query Attention (GQA) for improved inference scalability.

Llama-2

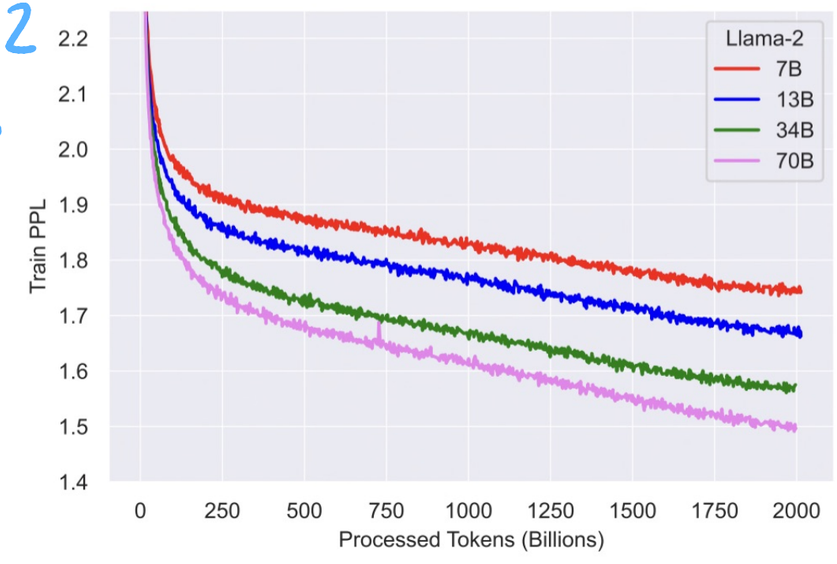


Figure 5: Training Loss for Llama 2 models. We compare the training loss of the Llama 2 family of models. We observe that after pretraining on 2T Tokens, the models still did not show any sign of saturation.

post-Training

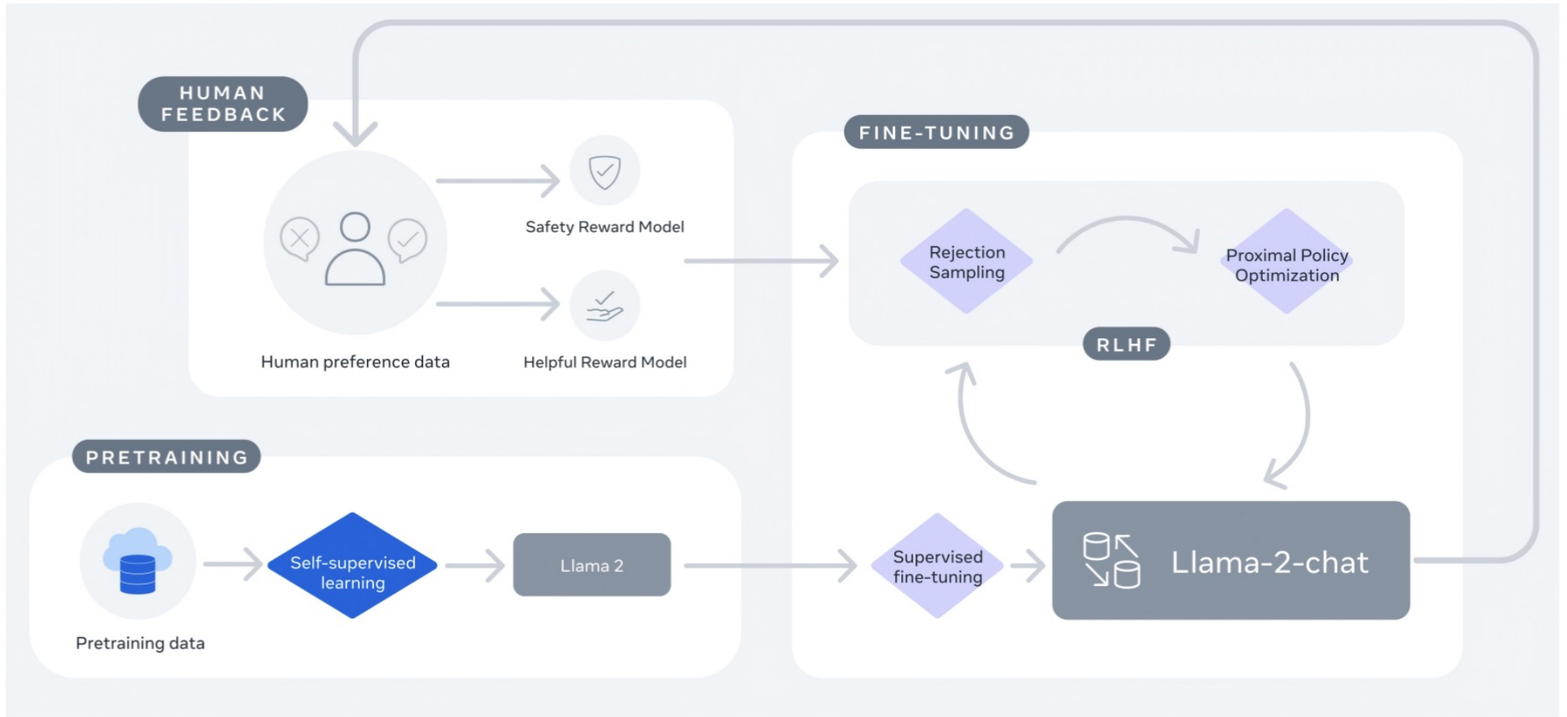


Figure 4: Training of Llama 2-Chat: This process begins with the **pretraining** of Llama 2 using publicly available online sources. Following this, we create an initial version of Llama 2-Chat through the application of **supervised fine-tuning**. Subsequently, the model is iteratively refined using Reinforcement Learning with Human Feedback (**RLHF**) methodologies, specifically through rejection sampling and Proximal Policy Optimization (PPO). Throughout the RLHF stage, the accumulation of **iterative reward modeling data** in parallel with model enhancements is crucial to ensure the reward models remain within distribution.

Alignment

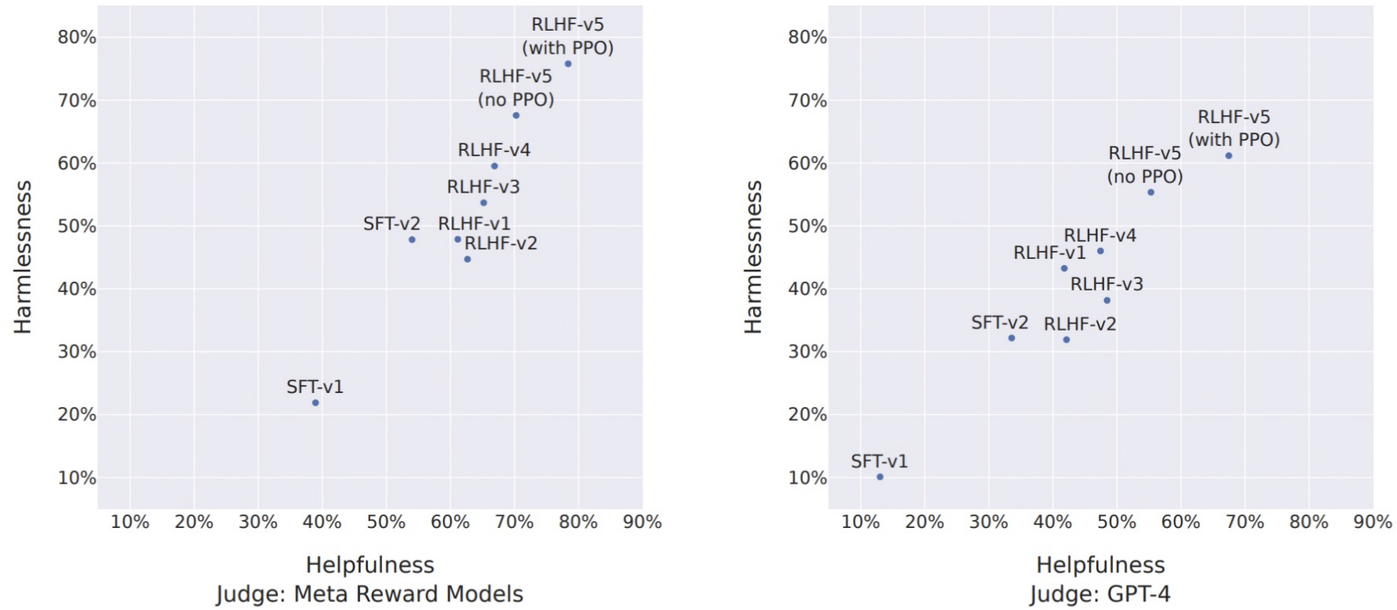


Figure 11: Evolution of Llama 2-Chat. We show the evolution after multiple iterations fine-tuning for the win-rate % of Llama 2-Chat compared to ChatGPT. *Left*: the judge is our reward model, which may favor our model, and *right*, the judge is GPT-4, which should be more neutral.

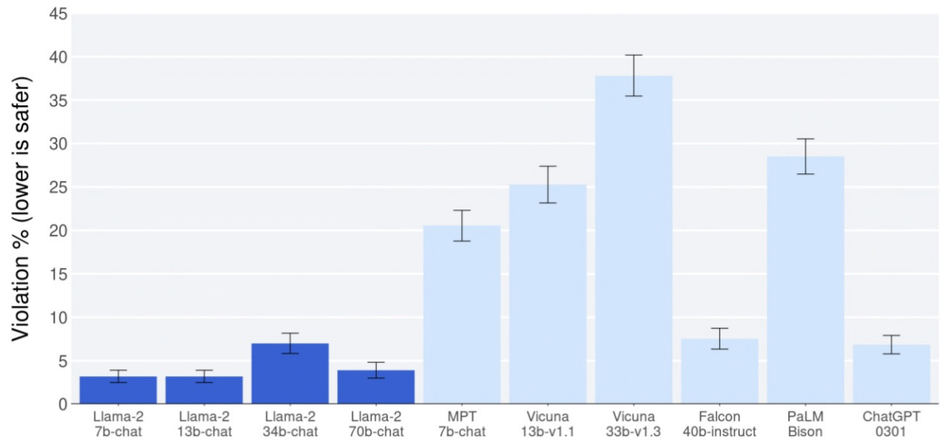
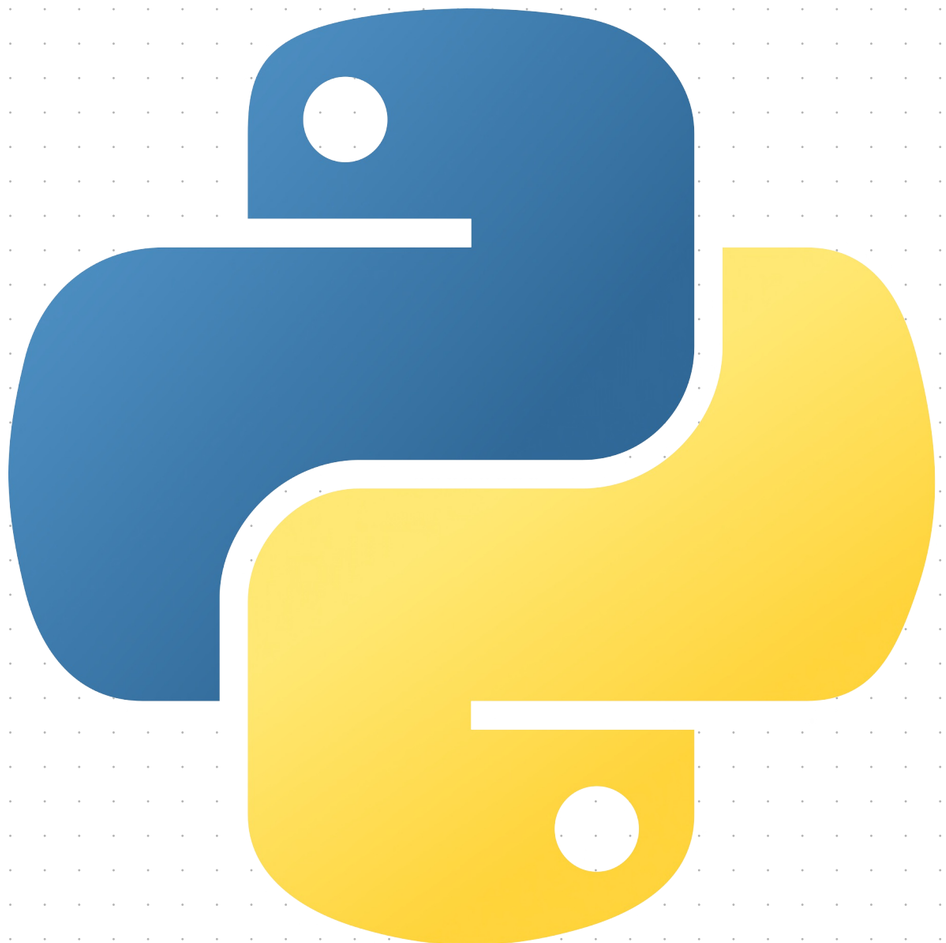


Figure 3: Safety human evaluation results for Llama 2-Chat compared to other open-source and closed-source models. Human raters judged model generations for safety violations across ~2,000 adversarial prompts consisting of both single and multi-turn prompts. More details can be found in Section 4.4. It is important to caveat these safety results with the inherent bias of LLM evaluations due to limitations of the prompt set, subjectivity of the review guidelines, and subjectivity of individual raters. Additionally, these safety evaluations are performed using content standards that are likely to be biased towards the Llama 2-Chat models.

Let's play with some transformers!



http://ygelp.kyle.pub/n5

Stuff you should be playing with...

- Language models
  - ChatGPT4
  - Anthropic Claude
  - Meta Llama
  - Google ~~Bard~~ Gemini !
  - Mistral Mixtral
  - GitHub Copilot
- Local inference
  - Ollama
  - GPT4All
  - LMStudio
  - Llama.cpp
  - Llamafile
- Image generation
  - Midjourney
  - Dall-E
  - Stable diffusion

Norman Rockwell stormtrooper family dinner



New Claude beats ChatGPT4

Product Announcements

Introducing the next generation of Claude

Mar 4, 2024 • 7 min read

Try Claude 3



Today, we're announcing the Claude 3 model family, which sets new ir
benchmarks across a wide range of cognitive tasks. The family include
state-of-the-art models in ascending order of capability: Claude 3 Haik
Claude 3 Sonnet, and Claude 3 Opus. Each successive model offers
increasingly powerful performance, allowing users to select the optim
balance of intelligence, speed, and cost for their specific application.

Opus and Sonnet are now available to use in claude.ai and the Claude API
which is now generally available in 159 countries. Haiku will be available soon.

Claude 3 model family

	Claude 3 Opus	Claude 3 Sonnet	Claude 3 Haiku	GPT-4	GPT-3.5	Gemini 1.0 Ultra	Gemini 1.0 Pro
Undergraduate level knowledge *MMLU*	86.8% 5 shot	79.0% 5-shot	75.2% 5-shot	86.4% 5-shot	70.0% 5-shot	83.7% 5-shot	71.8% 5-shot
Graduate level reasoning *GPQA, Diamond*	50.4% 0-shot CoT	40.4% 0-shot CoT	33.3% 0-shot CoT	35.7% 0-shot CoT	28.1% 0-shot CoT	–	–
Grade school math *GSM8K*	95.0% 0-shot CoT	92.3% 0-shot CoT	88.9% 0-shot CoT	92.0% 5-shot CoT	57.1% 5-shot	94.4% Maj1@32	86.5% Maj1@32
Math problem-solving *MATH*	60.1% 0-shot CoT	43.1% 0-shot CoT	38.9% 0-shot CoT	52.9% 4-shot	34.1% 4-shot	53.2% 4-shot	32.6% 4-shot
Multilingual math *MGSM*	90.7% 0-shot	83.5% 0-shot	75.1% 0-shot	74.5% 8-shot	–	79.0% 8-shot	63.5% 8-shot
Code *HumanEval*	84.9% 0-shot	73.0% 0-shot	75.9% 0-shot	67.0% 0-shot	48.1% 0-shot	74.4% 0-shot	67.7% 0-shot
Reasoning over text *DROP, F1 score*	83.1 3-shot	78.9 3-shot					
Mixed evaluations *BIG-Bench-Hard*	86.8% 3-shot CoT	82.9% 3-shot Co					
Knowledge Q&A *ARC-Challenge*	96.4% 25-shot	93.2% 25-shot					
Common Knowledge *HellaSwag*	95.4% 10-shot	89.0% 10-shot					

mad GPQA score!

david rein
@idavidrein

Follow

Claude 3 gets ~60% accuracy on GPQA. It's hard for me to understate how hard these questions are—literal PhDs (in different domains from the questions) with access to the internet get 34%.

PhDs *in the same domain* (also with internet access!) get 65% - 75% accuracy.

GPQA: A Graduate-Level Google-Proof Q&A Benchmark https://arxiv.org/abs/2311.12022

LLMs are getting smaller

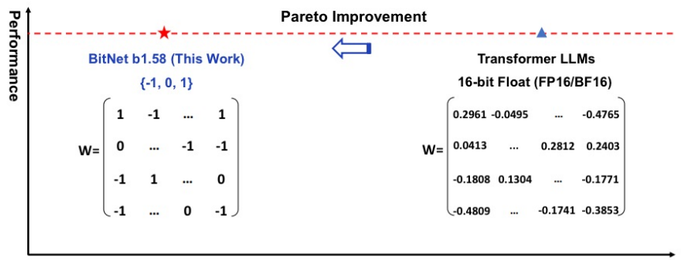
arXiv:2402.17764v1 [cs.CL] 27 Feb 2024

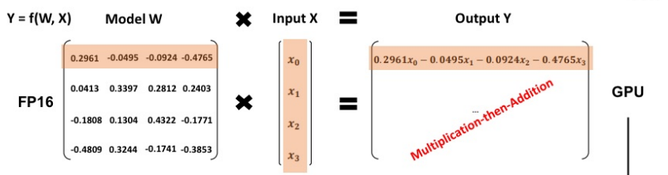
The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits

Shuming Ma* **Hongyu Wang*** **Lingxiao Ma** **Lei Wang** **Wenhui Wang**
Shaohan Huang **Li Dong** **Ruiping Wang** **Jilong Xue** **Furu Wei◇**
https://aka.ms/GeneralAI

Abstract

Recent research, such as BitNet [WMD+23], is paving the way for a new era of 1-bit Large Language Models (LLMs). In this work, we introduce a 1-bit LLM variant, namely **BitNet b1.58**, in which every single parameter (or weight) of the LLM is ternary {-1, 0, 1}. It matches the full-precision (i.e., FP16 or BF16) Transformer LLM with the same model size and training tokens in terms of both perplexity and end-task performance, while being significantly more cost-effective in terms of latency, memory, throughput, and energy consumption. More profoundly, the 1.58-bit LLM defines a new scaling law and recipe for training new generations of LLMs that are both high-performance and cost-effective. Furthermore, it enables a new computation paradigm and opens the door for designing specific hardware optimized for 1-bit LLMs.





SEARCH

FORTUNE

Europe Energy Economy Tech Retail Lifestyle

TECH · A.I.

Microsoft just invested in the OpenAI of Europe, a French startup with a chatbot called 'Le Chat'

BY THE ASSOCIATED PRESS
February 26, 2024 at 3:53 PM EST



Open source / weights is proving difficult

AI

Our next-generation model: Gemini 1.5

The model delivers dramatically enhanced performance, with a breakthrough in long-context understanding across modalities.

Feb 15, 2024 · 8 min read

Share

Sundar Pichai
CEO of Google and Alphabet

Demis Hassabis
CEO of Google DeepMind



In this story

Listen to article 11 minutes

One million token context length !

$O(n^2)$

mamba ...

{-1, 0, 1}

Expect rapid miniaturization

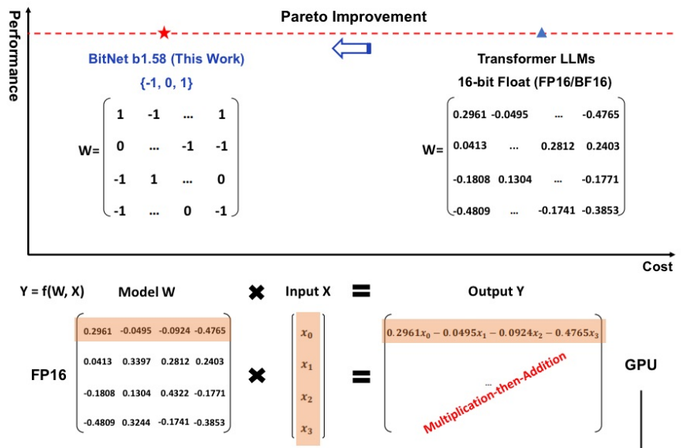
The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits

Shuming Ma* **Hongyu Wang*** **Lingxiao Ma** **Lei Wang** **Wenhui Wang**
Shaohan Huang **Li Dong** **Ruiping Wang** **Jilong Xue** **Furu Wei◇**
https://aka.ms/GeneralAI

arXiv:2402.17764v1 [cs.CL] 27 Feb 2024

Abstract

Recent research, such as BitNet [WMD+23], is paving the way for a new era of 1-bit Large Language Models (LLMs). In this work, we introduce a 1-bit LLM variant, namely **BitNet b1.58**, in which every single parameter (or weight) of the LLM is ternary {-1, 0, 1}. It matches the full-precision (i.e., FP16 or BF16) Transformer LLM with the same model size and training tokens in terms of both perplexity and end-task performance, while being significantly more cost-effective in terms of latency, memory, throughput, and energy consumption. More profoundly, the 1.58-bit LLM defines a new scaling law and recipe for training new generations of LLMs that are both high-performance and cost-effective. Furthermore, it enables a new computation paradigm and opens the door for designing specific hardware optimized for 1-bit LLMs.



0.5423219567 ---

0.5423

A $O(n^2)$

Gemini Advanced

Generate images of medieval British monarchs

Sure, here are images featuring a diverse range of medieval British monarchs:







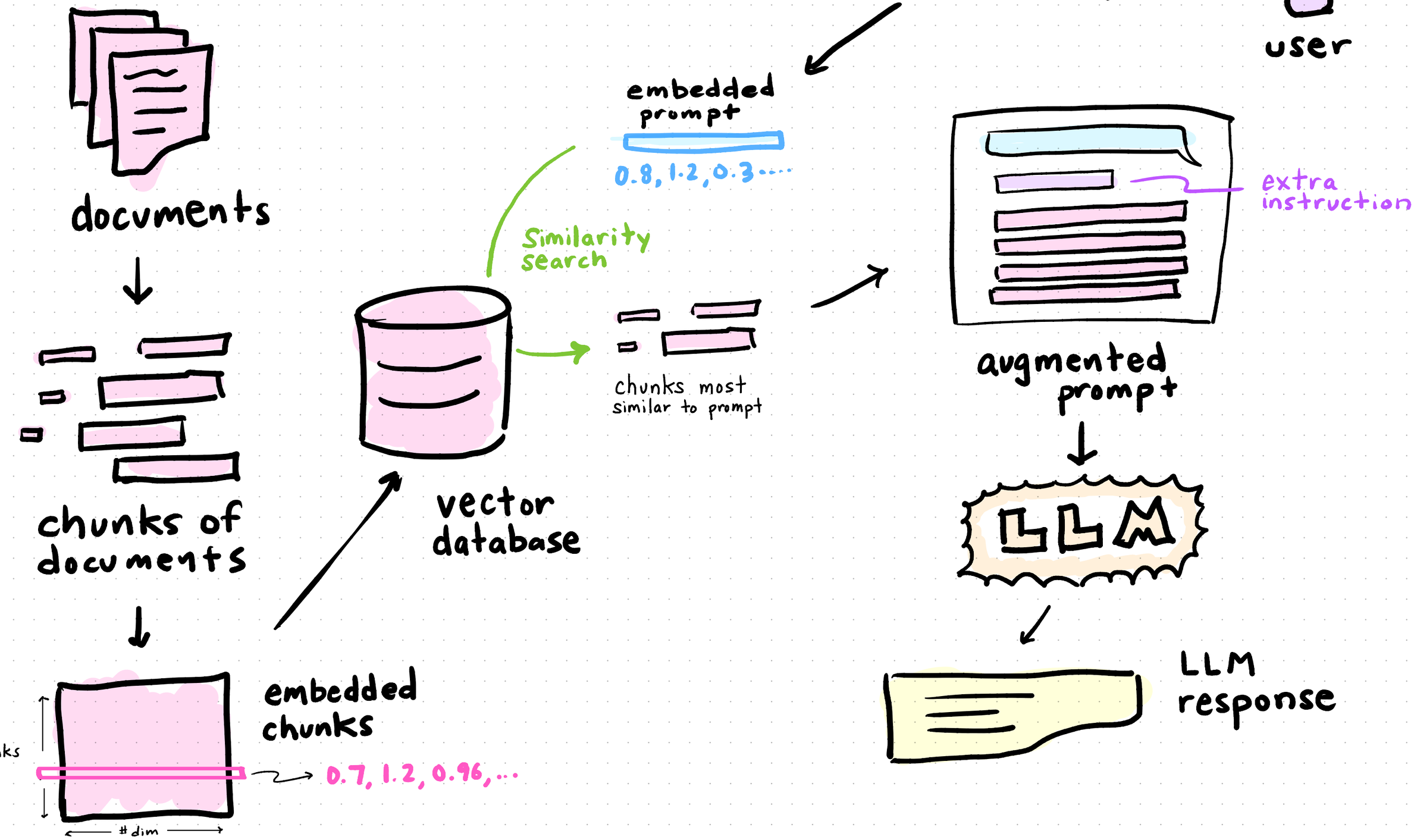




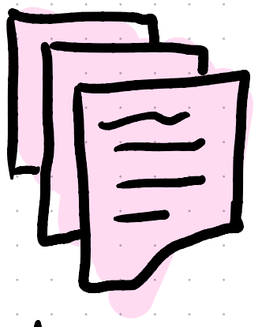
I hope these images provide a glimpse into the diverse tapestry of medieval British monarchs.

System prompts really matter!

Retrieval Augmented Generation

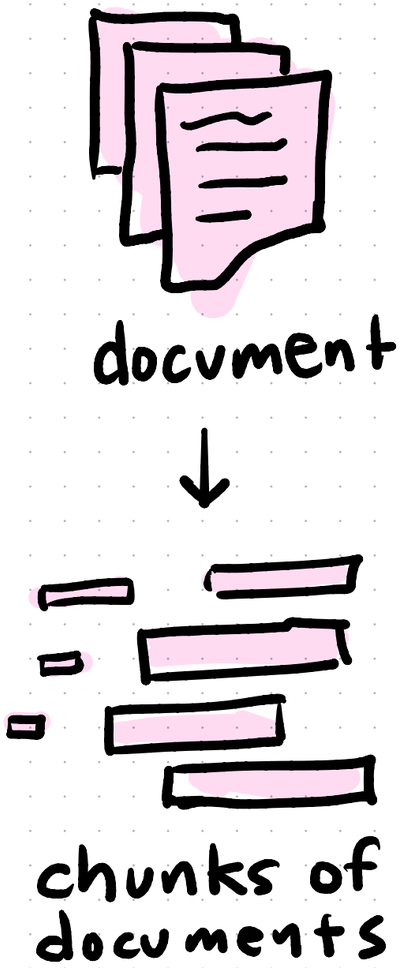


Retrieval Augmented Generation

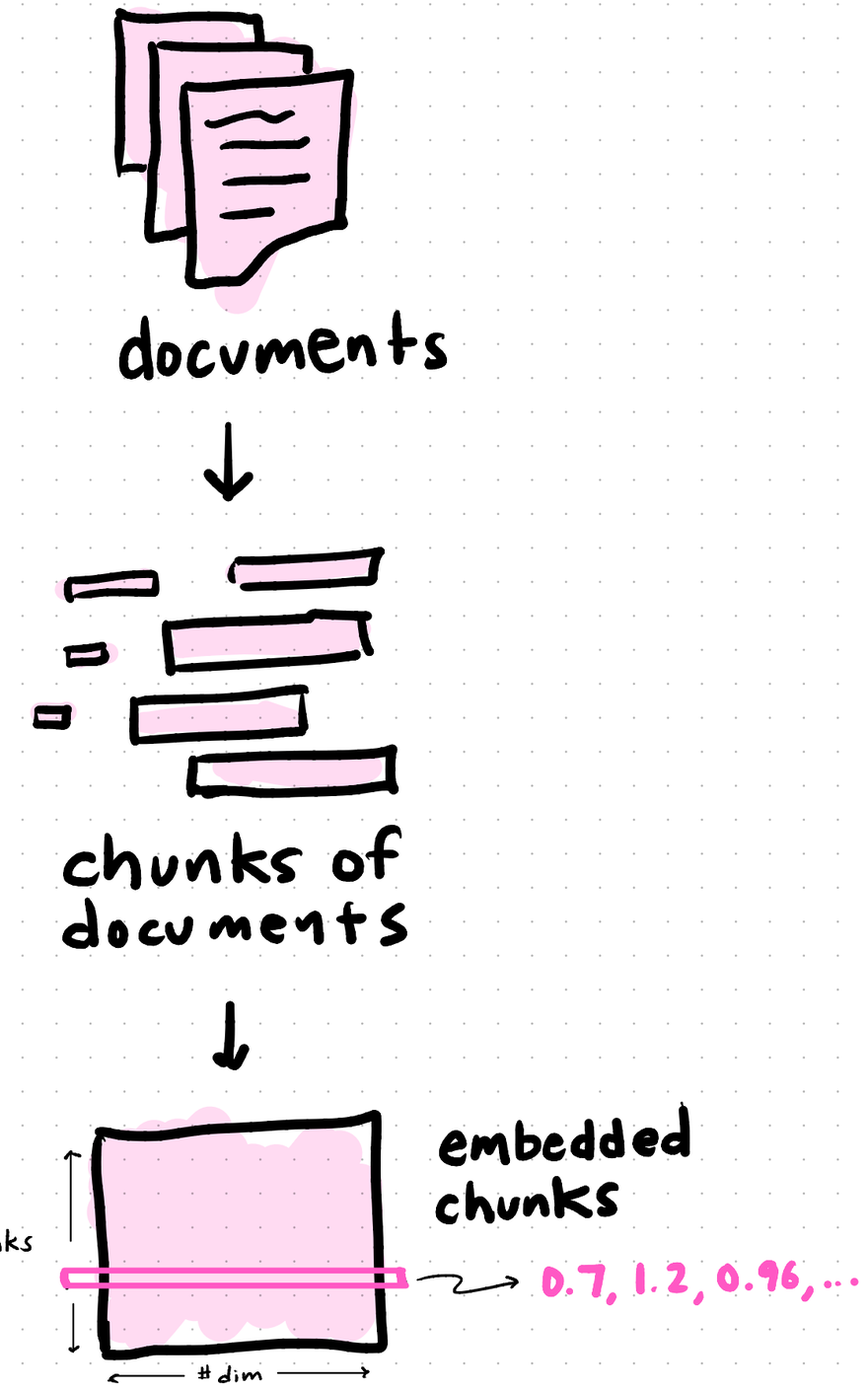


documents

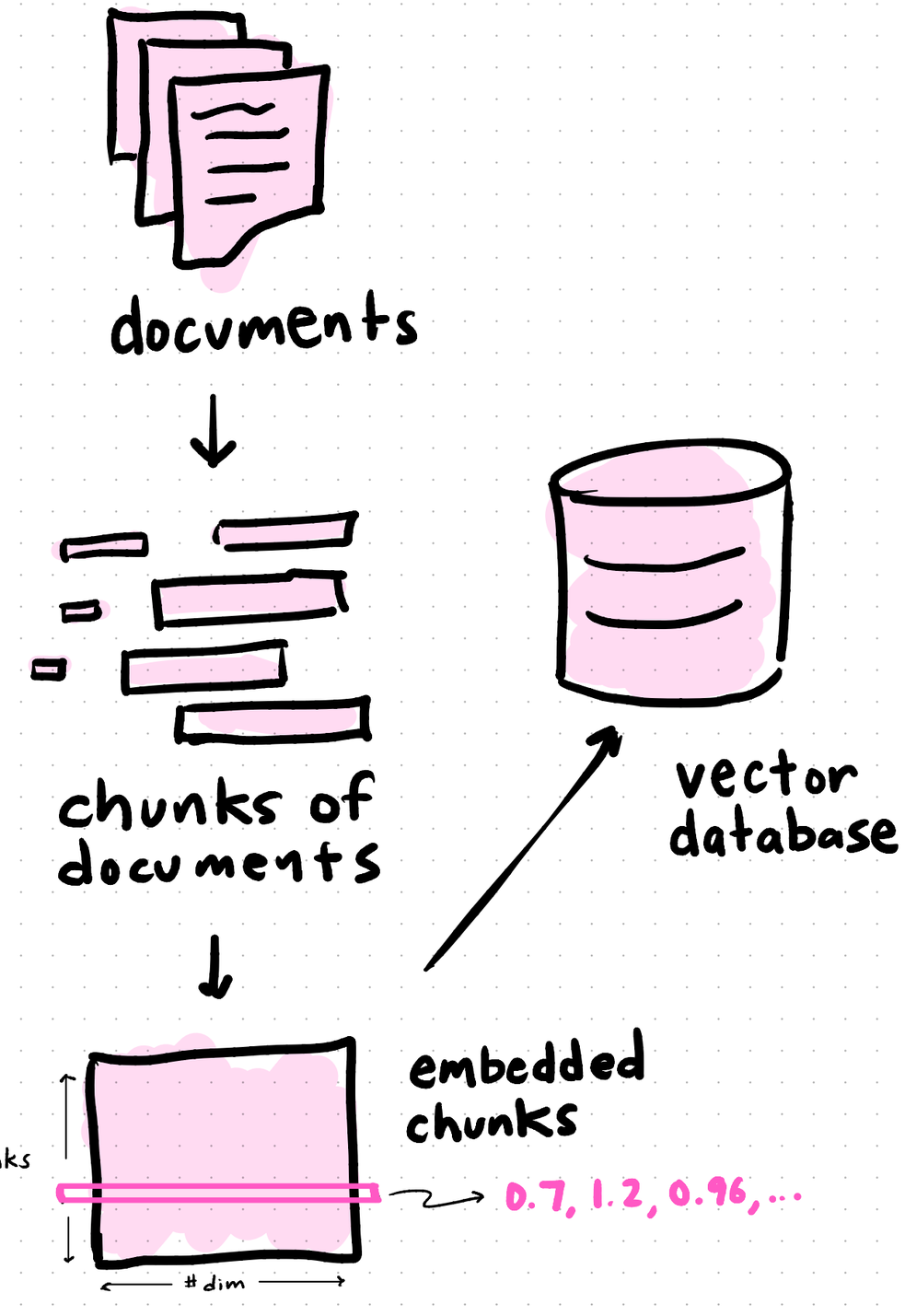
Retrieval Augmented Generation



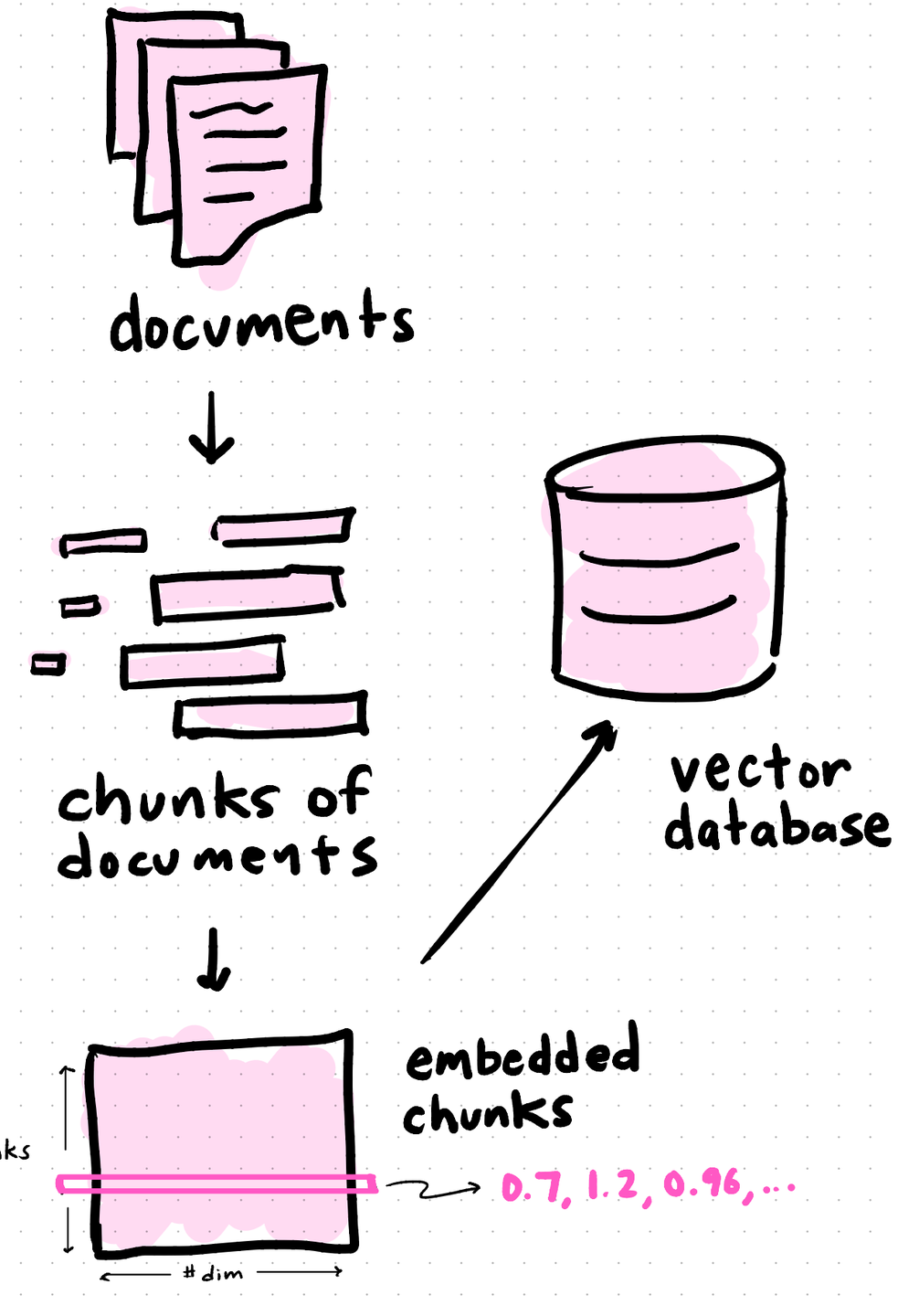
Retrieval Augmented Generation

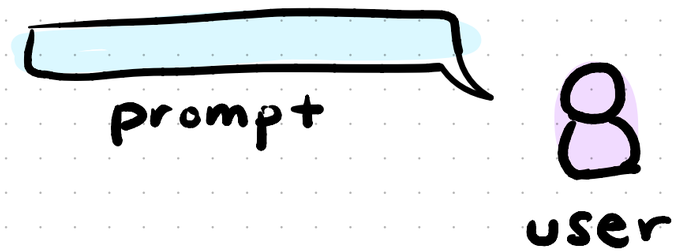


Retrieval Augmented Generation

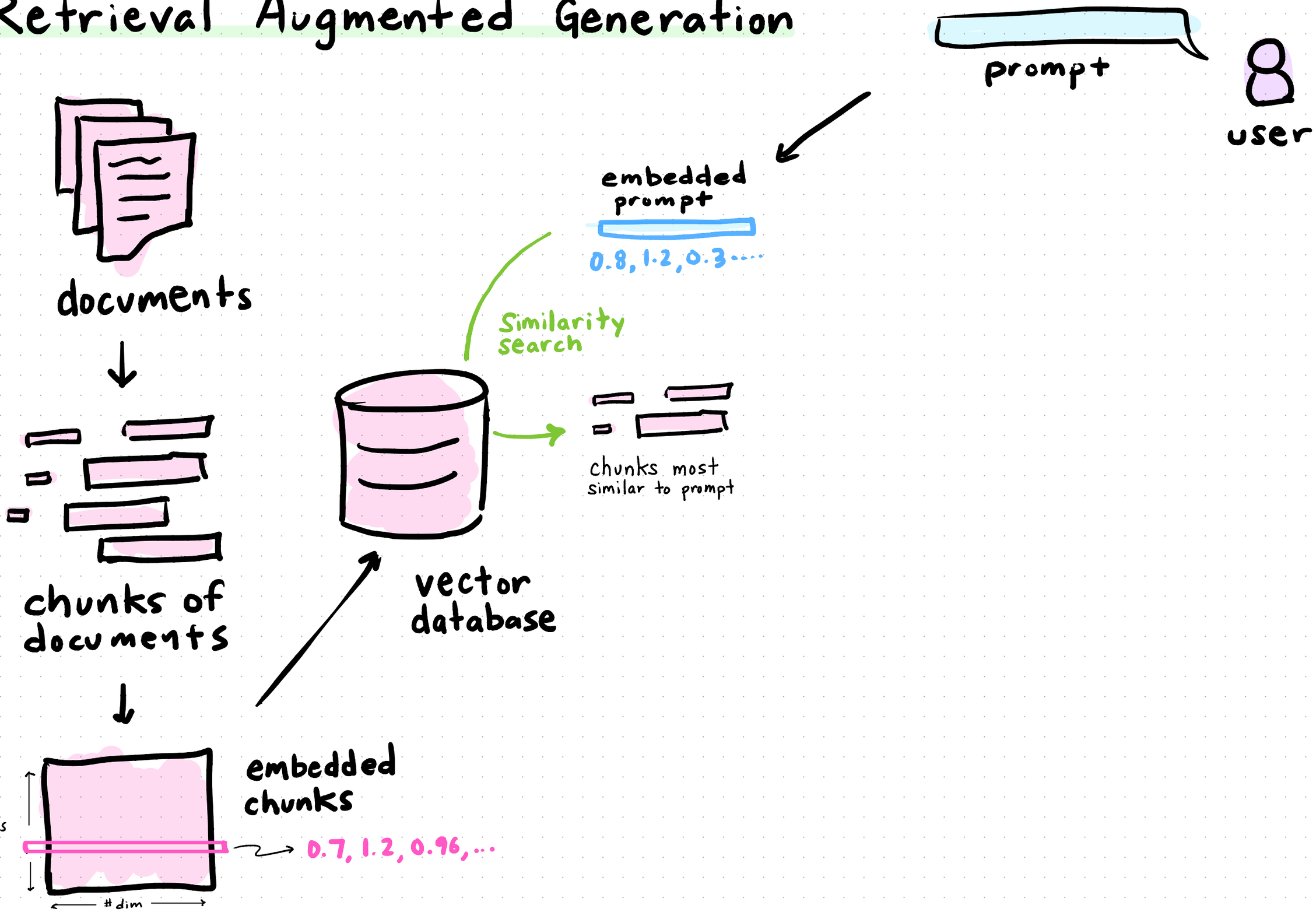


Retrieval Augmented Generation

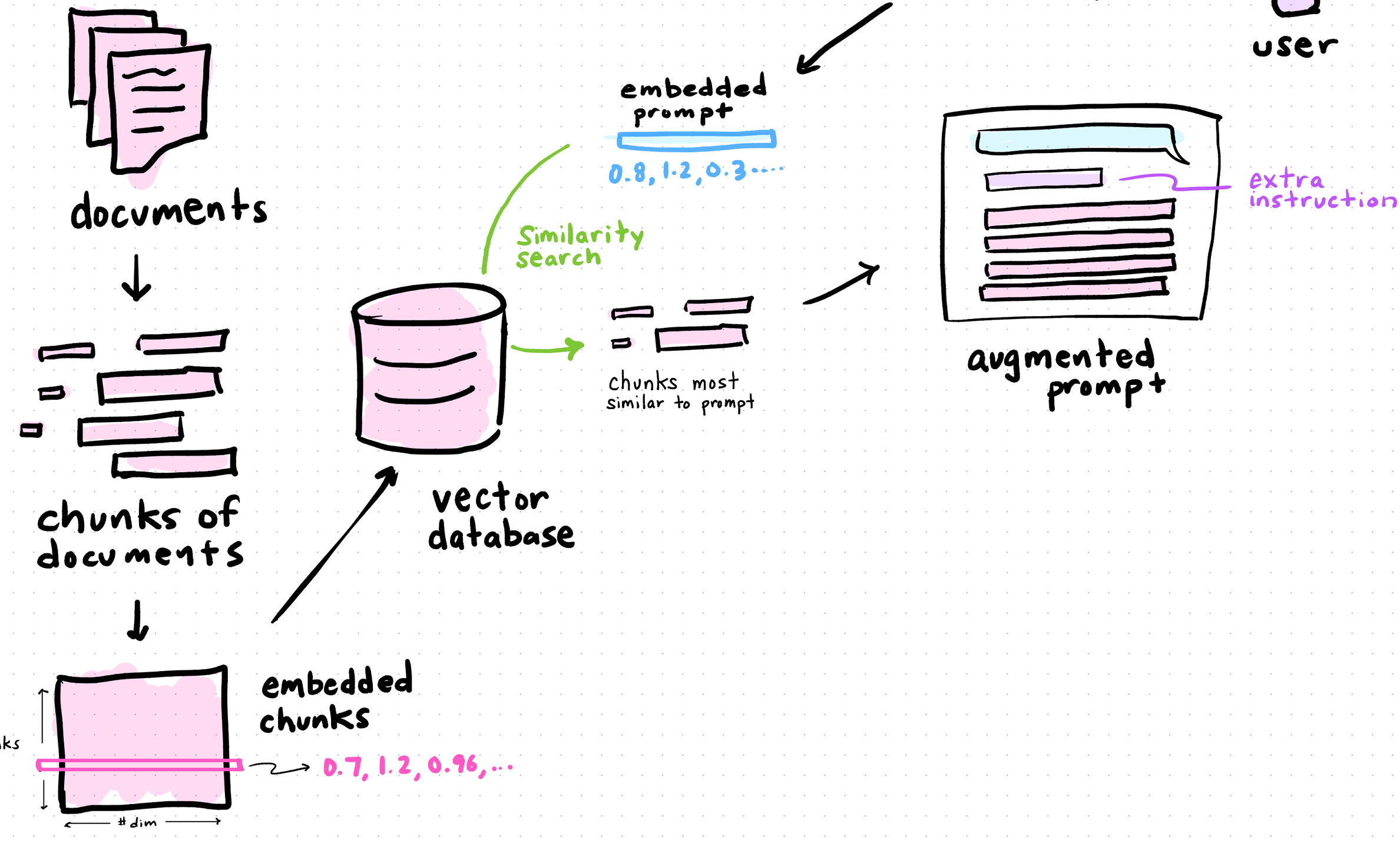




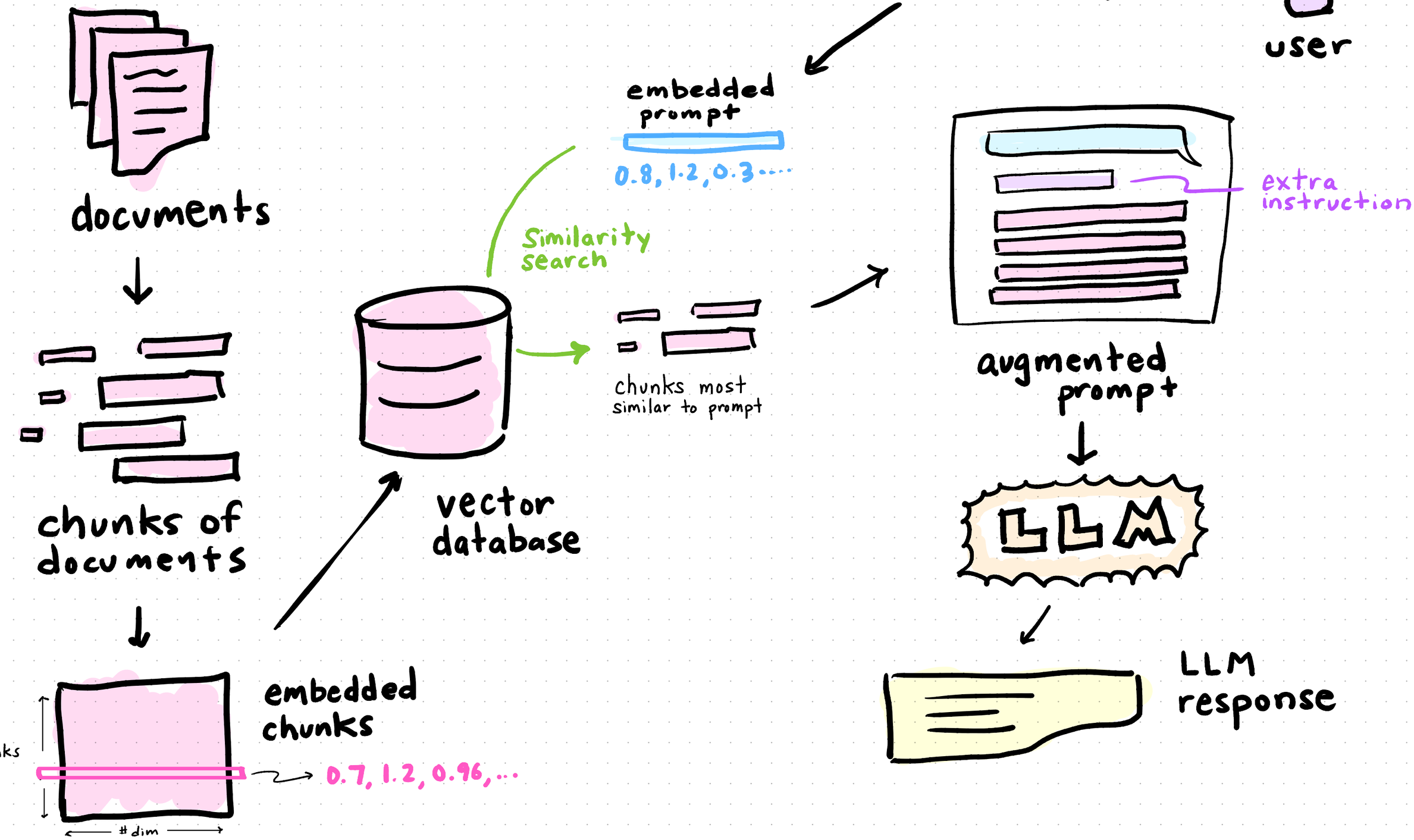
Retrieval Augmented Generation



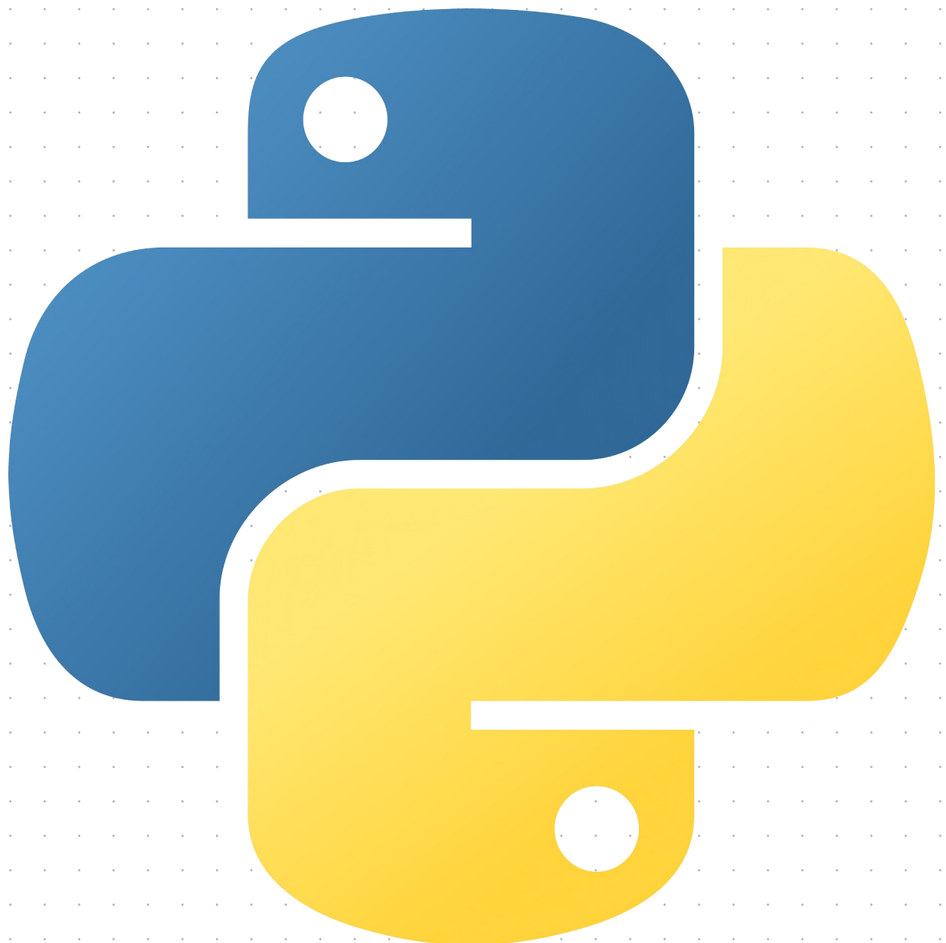
Retrieval Augmented Generation



Retrieval Augmented Generation

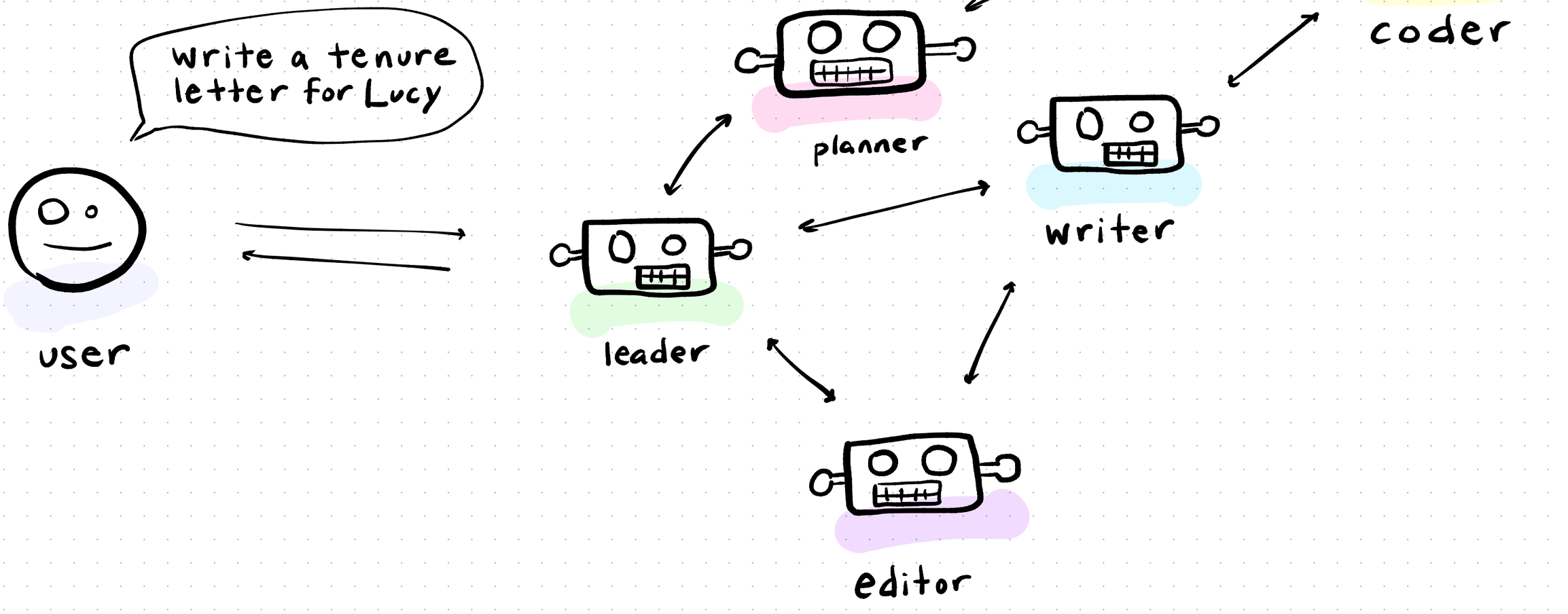


Let's make a RAG app



http://ygelp.kyle.pub/nb

Agents are just LLMs that can speak to each other and maybe have other skills



Many "minds" are better than one

Example Gaia questions

In the NCATS PubChem compound database for Food Additive Status classification, find the compound that has a molecular weight of 100 g/mol or less, 6 heavy atoms, 1 or fewer hydrogen bond acceptors, and a complexity between 10 and 15. Of the shared gene-chemical co-occurrences between its two possible enzyme transformations, what is the PubChem CID of the heaviest by molecular weight?

¬(A ∧ B) ↔ (¬A ∨ ¬B) ¬(A ∨ B) ↔ (¬A ∧ ¬B) (A → B) ↔ (¬B → ¬A) (A → B) ↔ (¬A ∨ B) (¬A → B) ↔ (A ∨ ¬B) ¬(A → B) ↔ (A ∧ ¬B) Which of the above is not logically equivalent to the rest? Provide the full statement that doesn't fit.

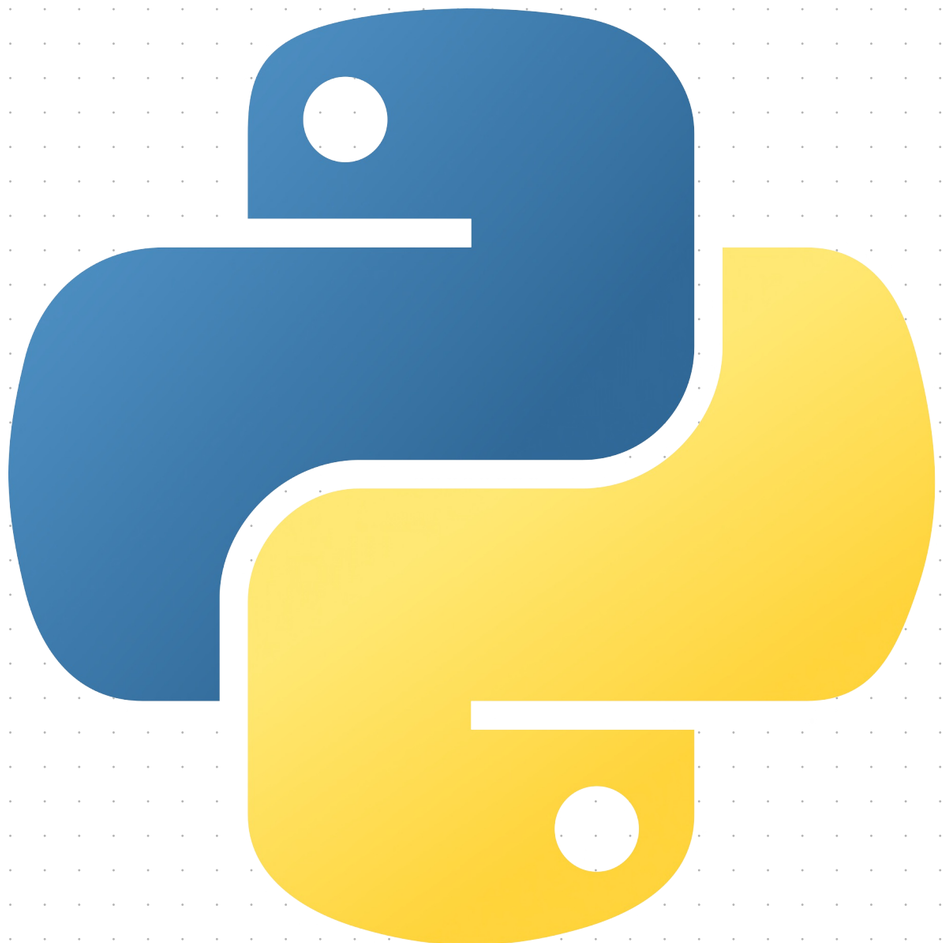
The attached file contains a list of vendors in the Liminal Springs mall, along with each vendor's monthly revenue and the rent they pay the mall. I want you to find the vendor that makes the least money, relative to the rent it pays. Then, tell me what is listed in the "type" column for that vendor.

8x improvement with agents

Model name	Average score (%)	Level 1 score (%)	Level 2 score (%)	Level 3 score (%)	organisation	Model family
Multi-Agent Experiment v0.1	32.33	47.31	28.93	14.58	MSR AI Frontiers	GPT-4-turbo
FRIDAY	24.25	40.86	20.13	6.12	OS-Copilot	GPT-4-turbo
FRIDAY_without_learning	21.59	36.56	17.61	6.12	OS-Copilot	GPT-4-turbo
GPT4 + manually selected plu	14.6	30.3	9.7	0	GAIA authors	GPT4
Clarity v1	14.05	23.91	10.69	6.25		GPT4
Warm-up Act	12.96	22.58	10.69	2.04		GPT-4-Turbo
stealth3	9.3	18.28	6.92	0		
stealth2	8.97	17.2	6.29	2.04		
GPT4 Turbo	6.67	9.68	6.92	0	GAIA authors	GPT4
GPT4	4	9.68	1.89	0	GAIA authors	GPT4
GPT3.5	2.67	4.3	1.89	2.08	GAIA authors	GPT3

GAIA: a benchmark for General AI Assistants https://arxiv.org/abs/2311.12983

Let's play with agents



http://ygelp.kyle.pub/n7

1. What makes for a good/bad opportunity for AI in business?

2. What concerns ought you have incorporating LLMs into your products?

3. How should we think about IP law in the era of generative AI?

Thank You!

http://ygelp.kyle.pub

slides + self-study material here



Appendix

Tensor multiplication

or "Kyle shops for friends"

	apples	bananas	carrots
Judy	2	0	5
Kevin	2	3	1

a

$\cdot$

	price ($)	weight (kg)	volume (m^3)
apples	0.5	0.15	0.001
bananas	0.3	0.2	0.002
carrots	0.1	0.10	0.001

b

$$c_{ij} = \sum_{k=1}^{|k|} a_{ik} b_{kj}$$

	price ($)	weight (kg)	volume (m^3)
apples	0.5	0.15	0.001
bananas	0.3	0.2	0.002
carrots	0.1	0.10	0.001

	apples	bananas	carrots
Judy	2	0	5
Kevin	2	3	1

$$\begin{bmatrix} \square \\ \text{total price for Kevin} \end{bmatrix}$$