

2024 Google I/O

“Gemma: Gemini's Open-Source Twin and The Rise of Small Language Model for Everyone”

Nguyen Khanh Linh

Founder of NeuroPurrect AI & CTO of Stealth startup

About Me

Nguyen Khanh Linh

- Founder of Beautiful Mind Vietnam, NeuroPurrfect AI and CTO of Stealth startup.
- Alumnus @ NUS & Georgia Tech.
- 11+ YOE in NLP, AI/ML.
- Previous experiences: IBM, A*STAR, Shopee Singapore, Mediacorp, Research Chapter Lead @ Techcombank...



Table of Contents

- 01 Introduction to Gemma
- 02 Key Features and Variations
- 03 Technical Architectures
- 04 Gemma 7B and its quantized version
- 05 Fine-tune your first Gemma 2B with
Unsloth, PEFT and LoRA
- 06 Gemma 2: Future Directions &
Possibilities
- 07 QA

Introduction to Gemma

Section 1

Back to Basics

Natural Language Processing, LLMs... what is all that about?

Back to basics

The Difficulty of Processing Human Language

Linh bảo “sao không đến”
Bảo Linh, sao không đến?
Linh bảo “đến không sao”
Không đến, Linh bảo sao?
Linh không đến, bảo sao
Linh đến, bảo không sao.
Sao bảo Linh không đến?
Bảo sao Linh không đến
...

Con ngựa đá con ngựa đá

Back to basics

The Difficulty of Processing Human Language

Google I/O 2024 là một sự kiện công nghệ quy tụ
những chuyên gia hàng [I](#)

Back to basics

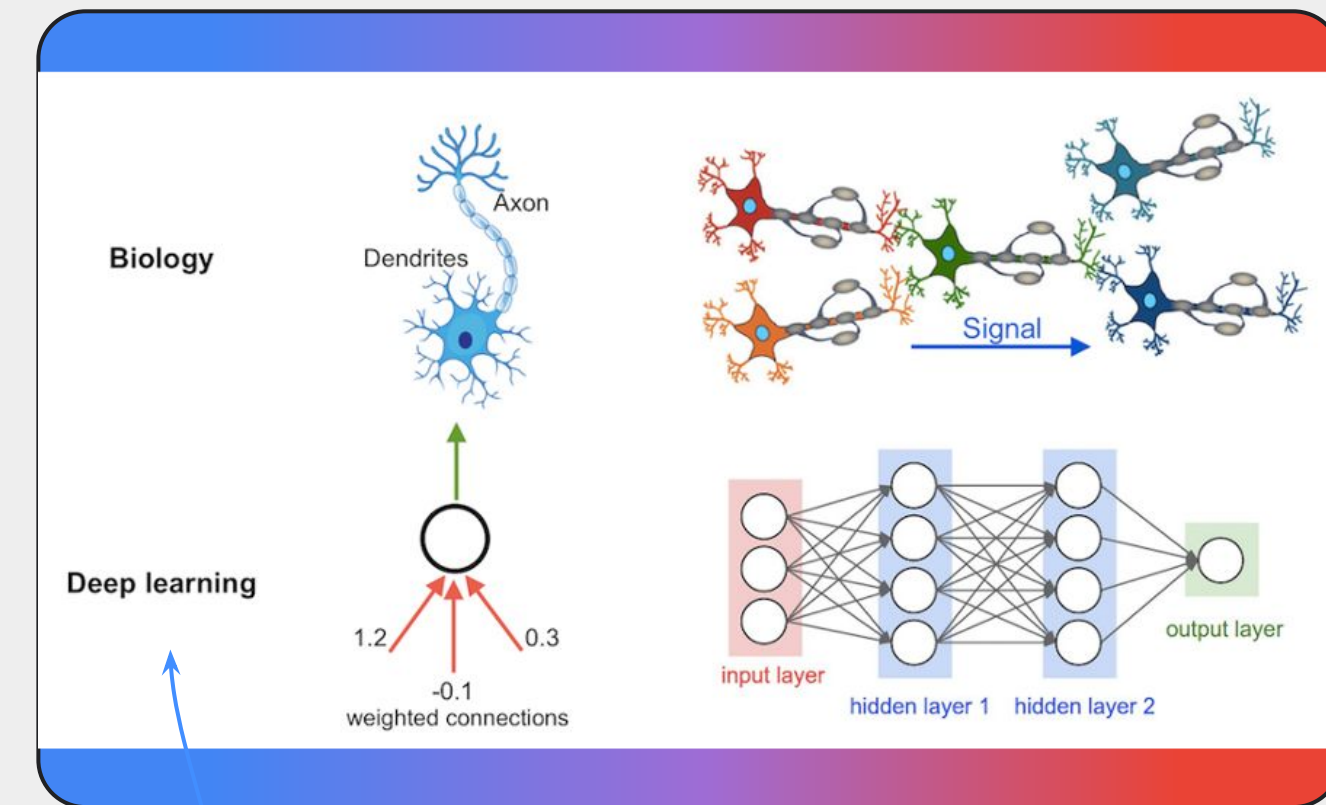
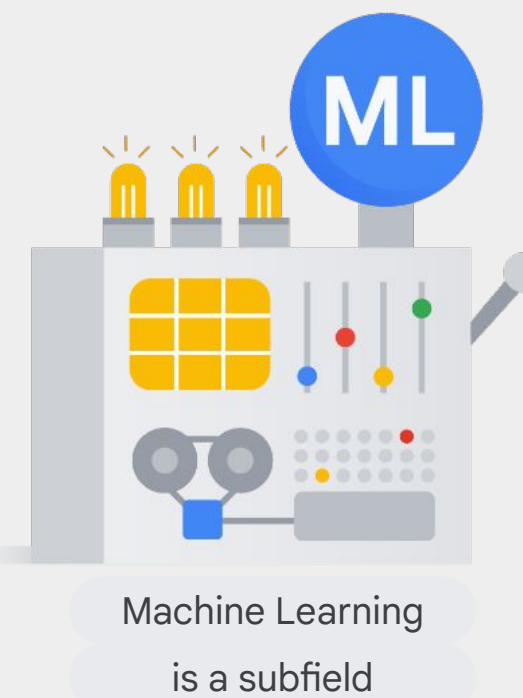
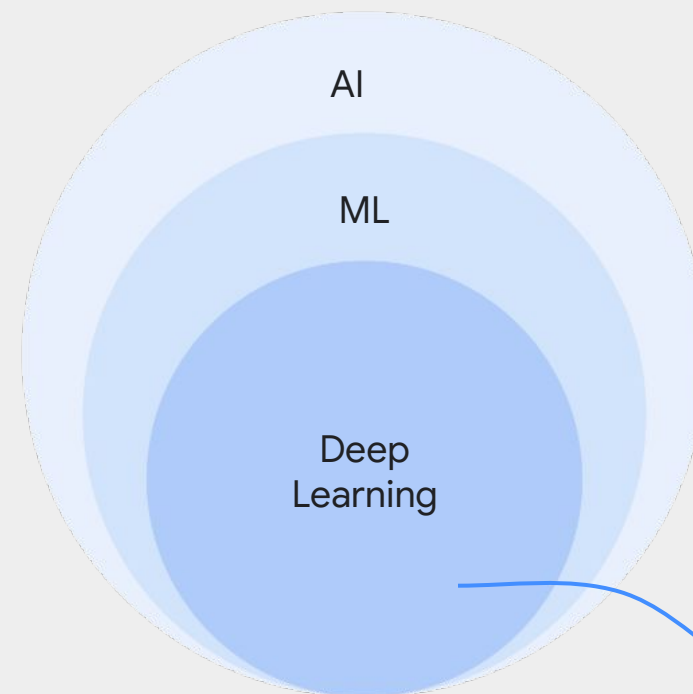
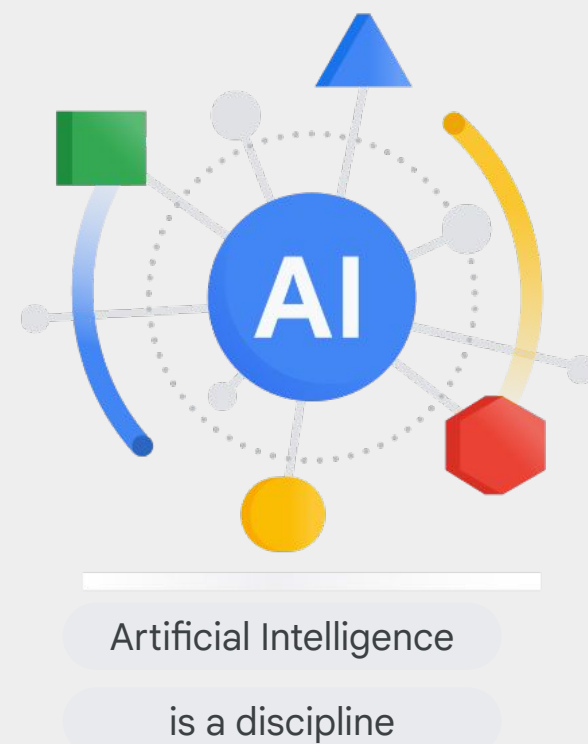
The Difficulty of Processing Human Language

Google I/O 2024 là một sự kiện công nghệ quy tụ
những chuyên gia hàng I

đầu
giả
ngày
hóa
rẻ
sỉ

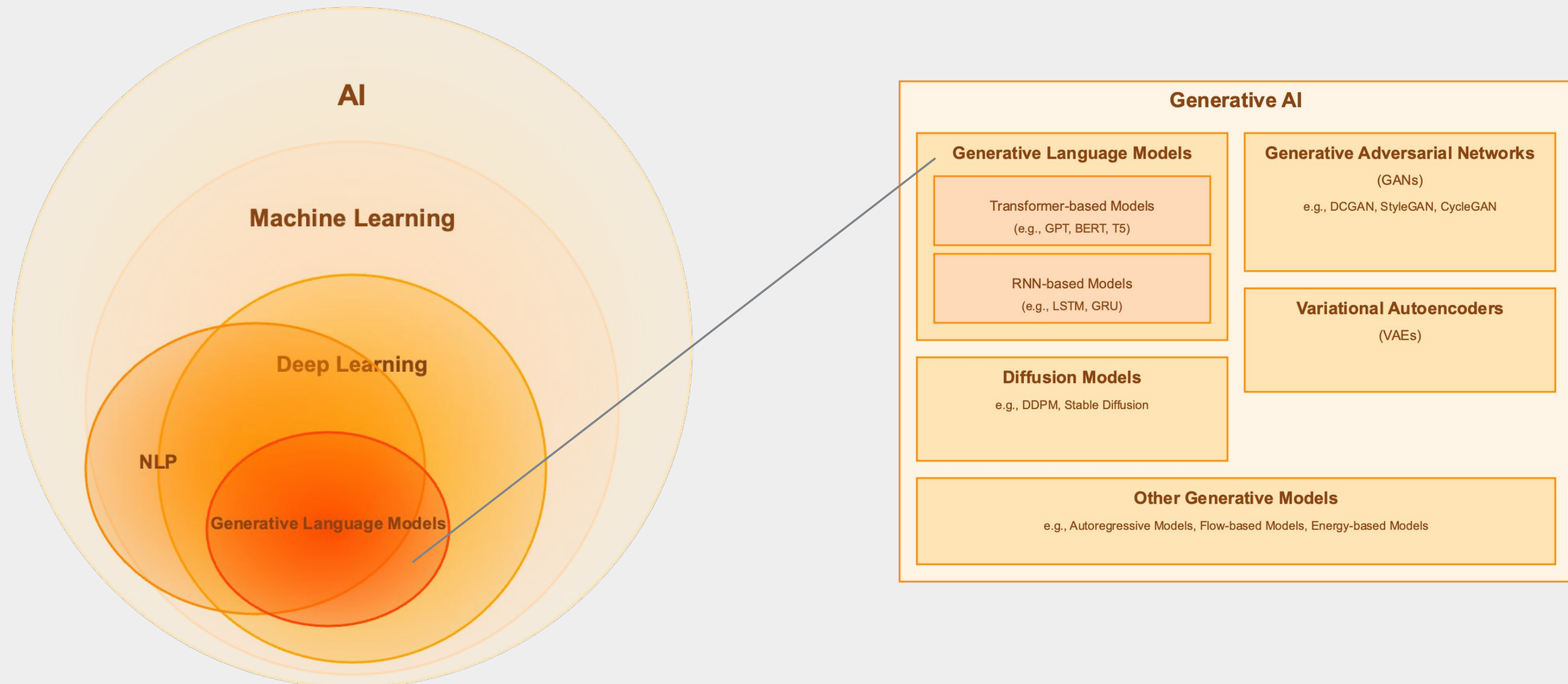
Back to basics

AI, Machine Learning and Deep Learning



Back to basics

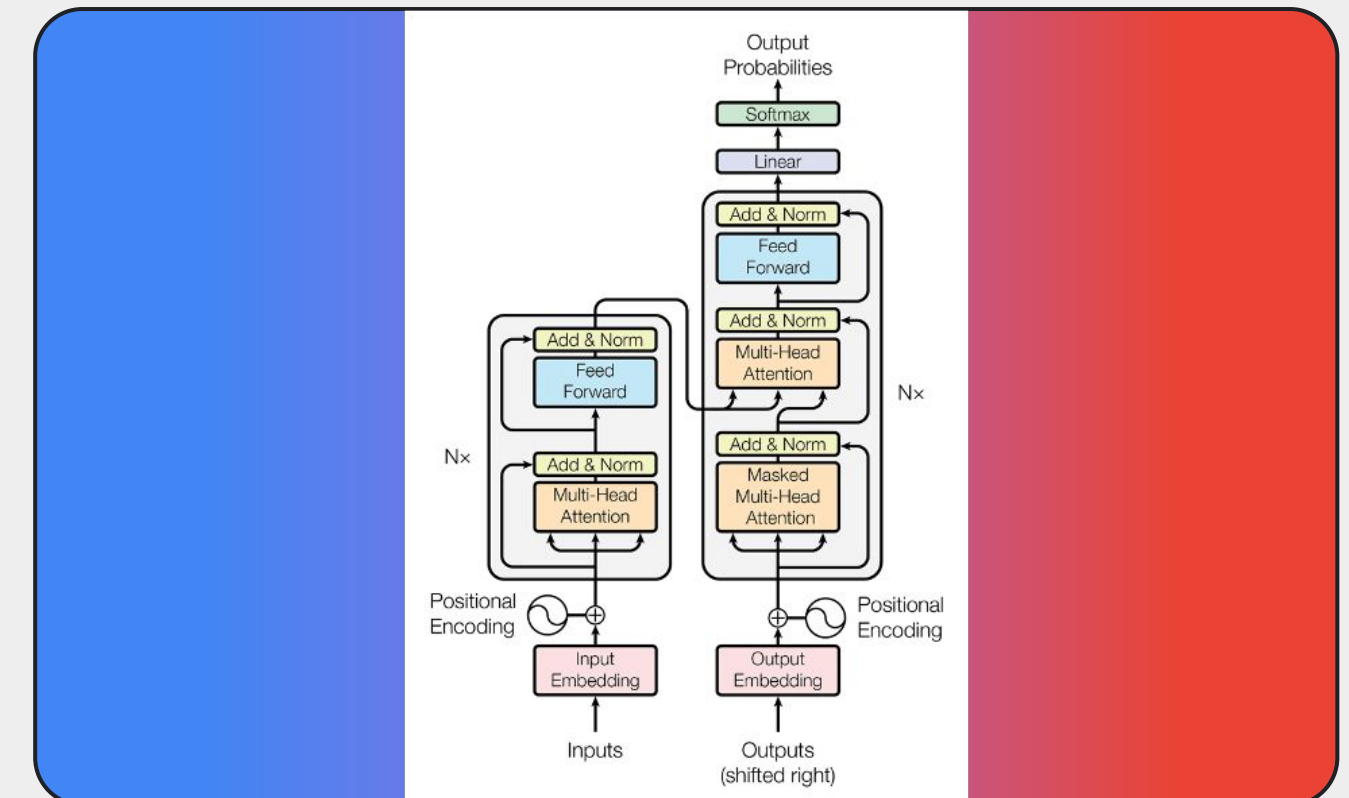
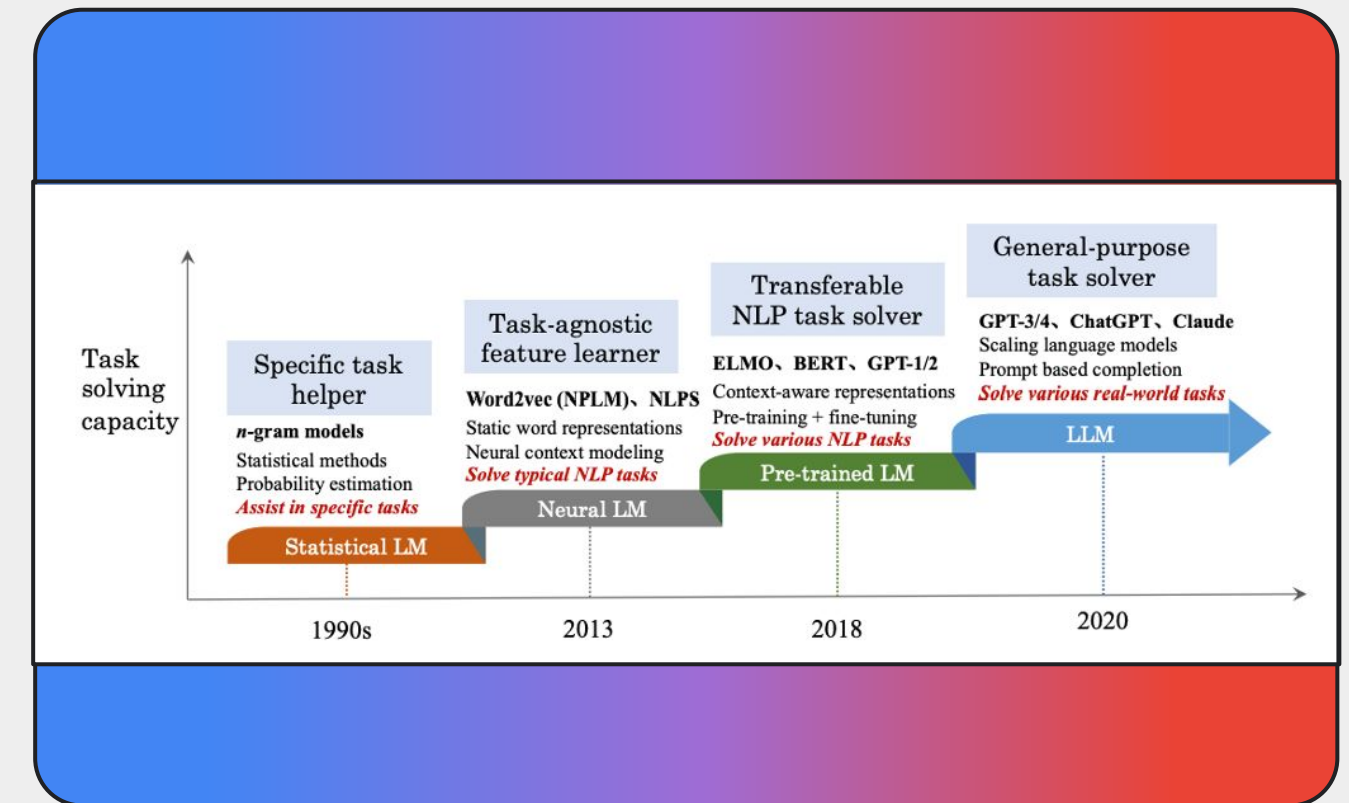
How about Natural Language Processing and Generative Language Model?



What is Large Language Model?

1. Language Model: contains complex algorithms and large amounts of textual data to learn patterns, relationships, and context within language.
2. Language modeling has been widely studied for language understanding and generation in the past two decades, evolving from statistical language models to neural language models
3. Large Language Model (LLM): Just a much bigger size than usual Language Model (I.e: has tens or hundreds of billions of parameters, trained on huge amount of data...)

LLM: Large Language Model



Transformer's general architecture with encoder blocks and decoder blocks

Section 2

Gemma: The Gemini's twin, but open-source

And smaller

What is Gemma?



Gemma Open Models

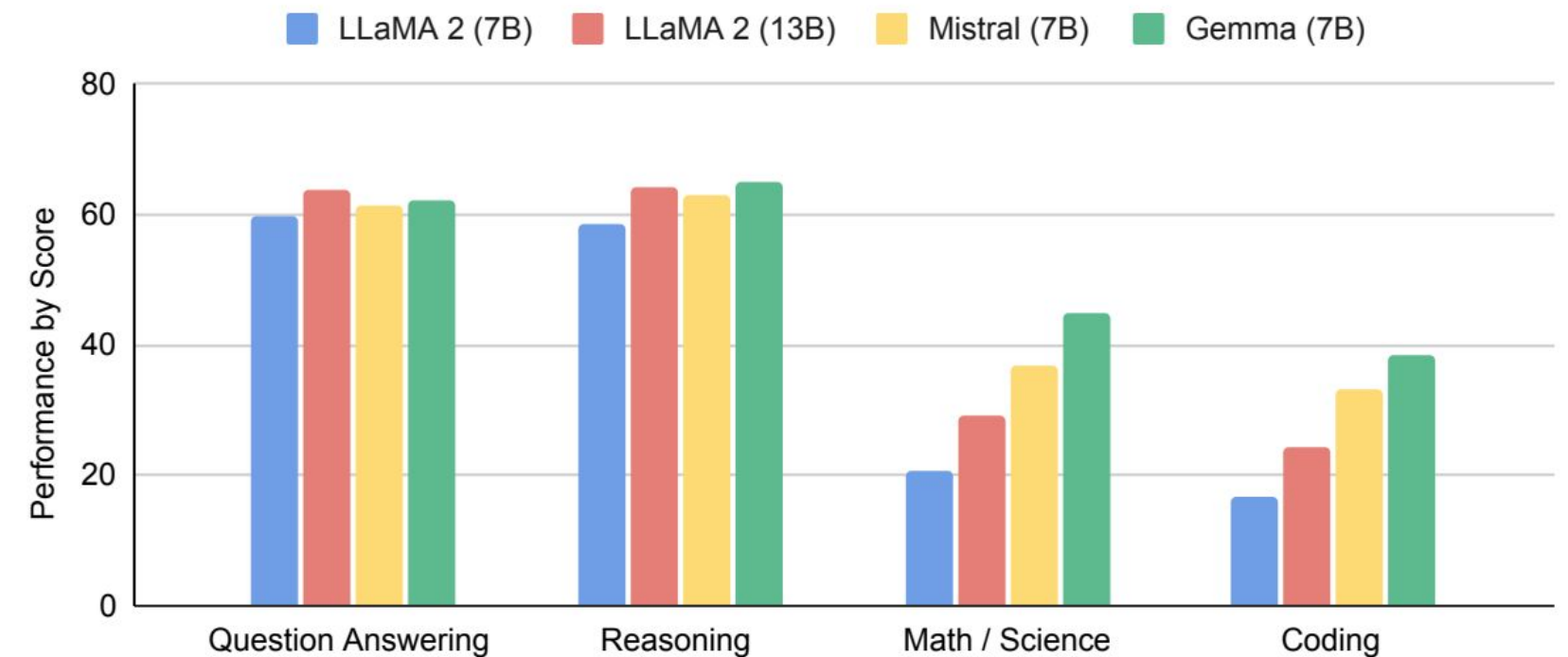
Gemma is a family of [lightweight](#), state-of-the-art [open models](#) developed by [Google DeepMind](#) and other Google teams. Inspired by and built using the same research and technology as the Gemini models, Gemma represents a more [accessible](#) version of Google's advanced AI capabilities.

Evaluation Benchmark Against Other Generative Language Models

			Gemma		Llama-2	
CAPABILITY	BENCHMARK	DESCRIPTION	7B		7B	13B
General	MMLU	Representation of questions in 57 subjects (incl. STEM, humanities and others)	64.3		45.3	54.8
Reasoning	BBH	Diverse set of challenging tasks requiring multi-step reasoning	55.1		32.6	39.4
	HellaSwag	Commonsense reasoning for everyday tasks	81.2		77.2	80.7
Math	GSM8K	Basic arithmetic manipulations (incl. Grade School math problems)	46.4		14.6	28.7
	MATH	Challenging math problems (incl. algebra, geometry, pre-calculus, and others)	24.3		2.5	3.9
Code	HumanEval	Python code generation	32.3		12.8	18.3

Evaluation Benchmark Against Other Generative Language Models

Benchmark	metric	LLaMA-2		Mistral	Gemma	
		7B	13B	7B	2B	7B
MMLU	5-shot, top-1	45.3	54.8	62.5	42.3	64.3
HellaSwag	0-shot	77.2	80.7	81.0	71.4	81.2
PIQA	0-shot	78.8	80.5	82.2	77.3	81.2
SIQA	0-shot	48.3	50.3	47.0*	49.7	51.8
Boolq	0-shot	77.4	81.7	83.2*	69.4	83.2
Winogrande	partial scoring	69.2	72.8	74.2	65.4	72.3
CQA	7-shot	57.8	67.3	66.3*	65.3	71.3
OBQA		58.6	57.0	52.2	47.8	52.8
ARC-e		75.2	77.3	80.5	73.2	81.5
ARC-c		45.9	49.4	54.9	42.1	53.2
TriviaQA	5-shot	72.1	79.6	62.5	53.2	63.4
NQ	5-shot	25.7	31.2	23.2	12.5	23.0
HumanEval	pass@1	12.8	18.3	26.2	22.0	32.3
MBPP [†]	3-shot	20.8	30.6	40.2*	29.2	44.4
GSM8K	maj@1	14.6	28.7	35.4*	17.7	46.4
MATH	4-shot	2.5	3.9	12.7	11.8	24.3
AGIEval		29.3	39.1	41.2*	24.2	41.7
BBH		32.6	39.4	56.1*	35.2	55.1
Average		46.9	52.4	54.5	45.0	56.9



Key Features & Variations

Features

01

Optimized and Robust

1. Achieves strong performance with much smaller model sizes compared to other open models
2. Can run on laptops, workstations, or in the cloud with Google Cloud
3. Optimized for multiple frameworks (JAX, PyTorch, TensorFlow/Keras), hardware platforms (NVIDIA GPUs, Google Cloud TPUs), and popular AI/ML tools

02

Private and Customized

1. Filtered training data to remove personal info and sensitive content, and aligned with human feedback for responsible behaviors
2. Supports fine-tuning on custom data to adapt models to specific applications

03

State-of-the-Art supports

1. Comes with a Responsible Generative AI Toolkit to help developers prioritize building safe and responsible AI
2. Developed with Google's AI Principles and responsible practices at the forefront, with robust safety evaluations
3. Backed by Google's AI research innovations and technical infrastructure used in their most advanced models

Variations

Gemma (PT)

- Gemma Pre-trained (or “Base”) model
- Available in 2B, 7B
- Not trained on any specific tasks or instructions beyond the Gemma core data training set.

Gemma (IT)

- Gemma Instruction tuned (IT) model
- Available in 2B, 7B
- Trained with human language interactions and can respond to conversational input, similar to a chat bot.

CodeGemma

- Can perform a variety of coding tasks
- Available in PT (code completion), and IT (language-to-code and instructions)
- Available in 2B, 7B

PaliGemma

- Open vision-language model (VLM) inspired by PaLI-3
- Takes both images and text as inputs
- Available in PT/Base and IT (instruction tuned)

RecurrentGemma

- A hybrid model architecture that mixes gated linear recurrences with local sliding window attention
- Can perform variety of tasks: Q&A, summarization & reasoning
- Optimized, fast, and high performance

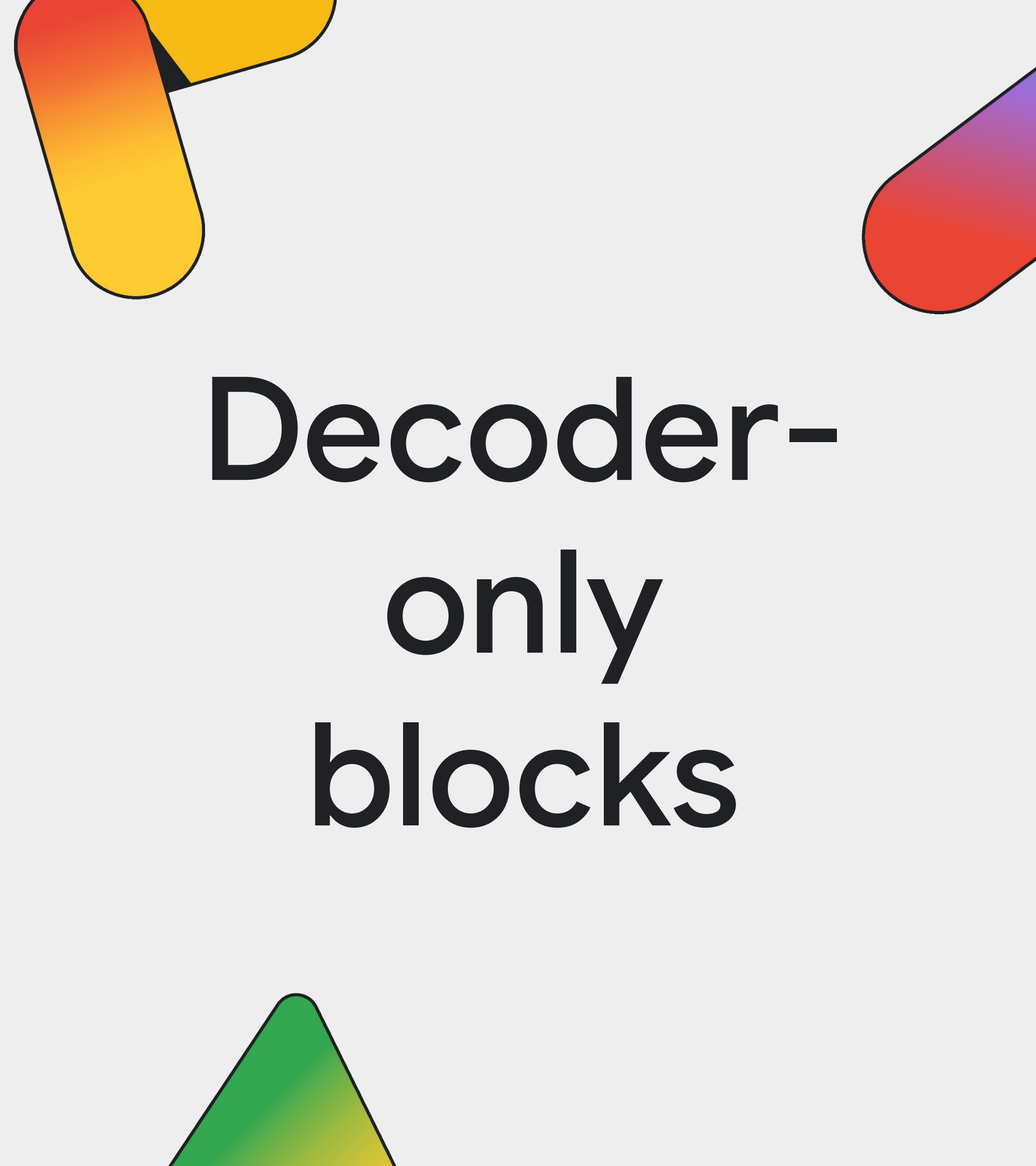
Technical Architectures: What you should know

Optional tag here

Based on Transformer

Gemma builds upon the foundational transformer decoder architecture introduced in "Attention Is All You Need," incorporating several key enhancements:

- Replaces multi-head attention with Multi-Query Attention for improved efficiency (for 2B version only, 7B still uses MHA)
- Implements RoPE Embeddings across all layers, shared between inputs and outputs to reduce model size.
- Utilizes GeGLU Activations in place of ReLU for better performance.
- Employs RMSNorm to normalize both inputs and outputs of each transformer sub-layer, optimizing the normalizer placement.

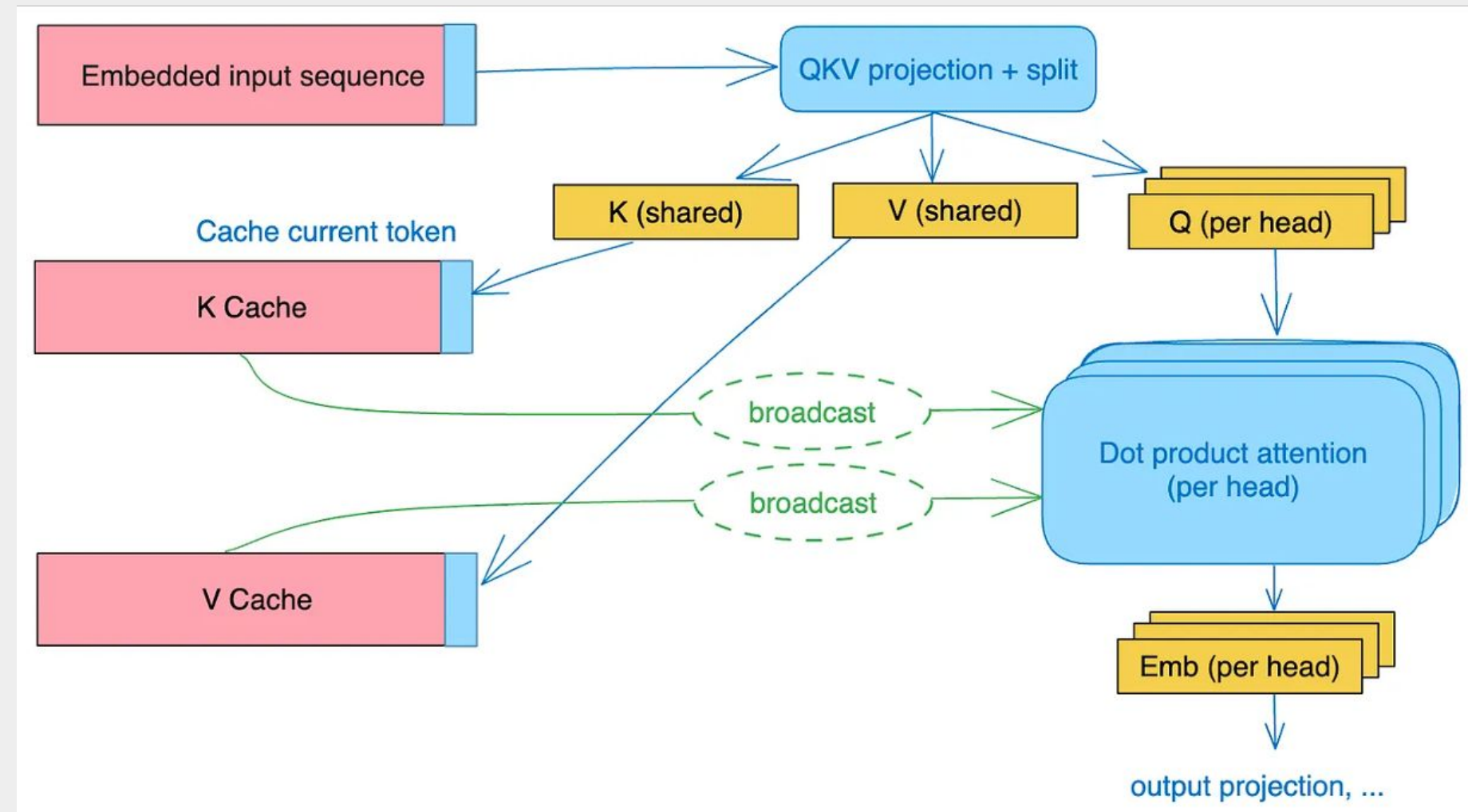
A decorative graphic on the right side of the slide. It features several overlapping, rounded, pill-shaped elements in various colors: a yellow one at the top left, a red one at the top right, and a green one at the bottom center. The background is a light gray.

Decoder- only blocks

Multi-query Attention

Shazeer (2019) introduced **Multi-Query Attention (MQA)**, an optimization of **Multi-Head Attention (MHA)**.

MQA improves efficiency with minimal accuracy loss by reducing or eliminating the heads dimension from K and V values. While MHA replicates the entire attention computation for each head, MQA applies the same K and V transformation to each query "head". This simplification enhances computational efficiency while largely preserving model performance.

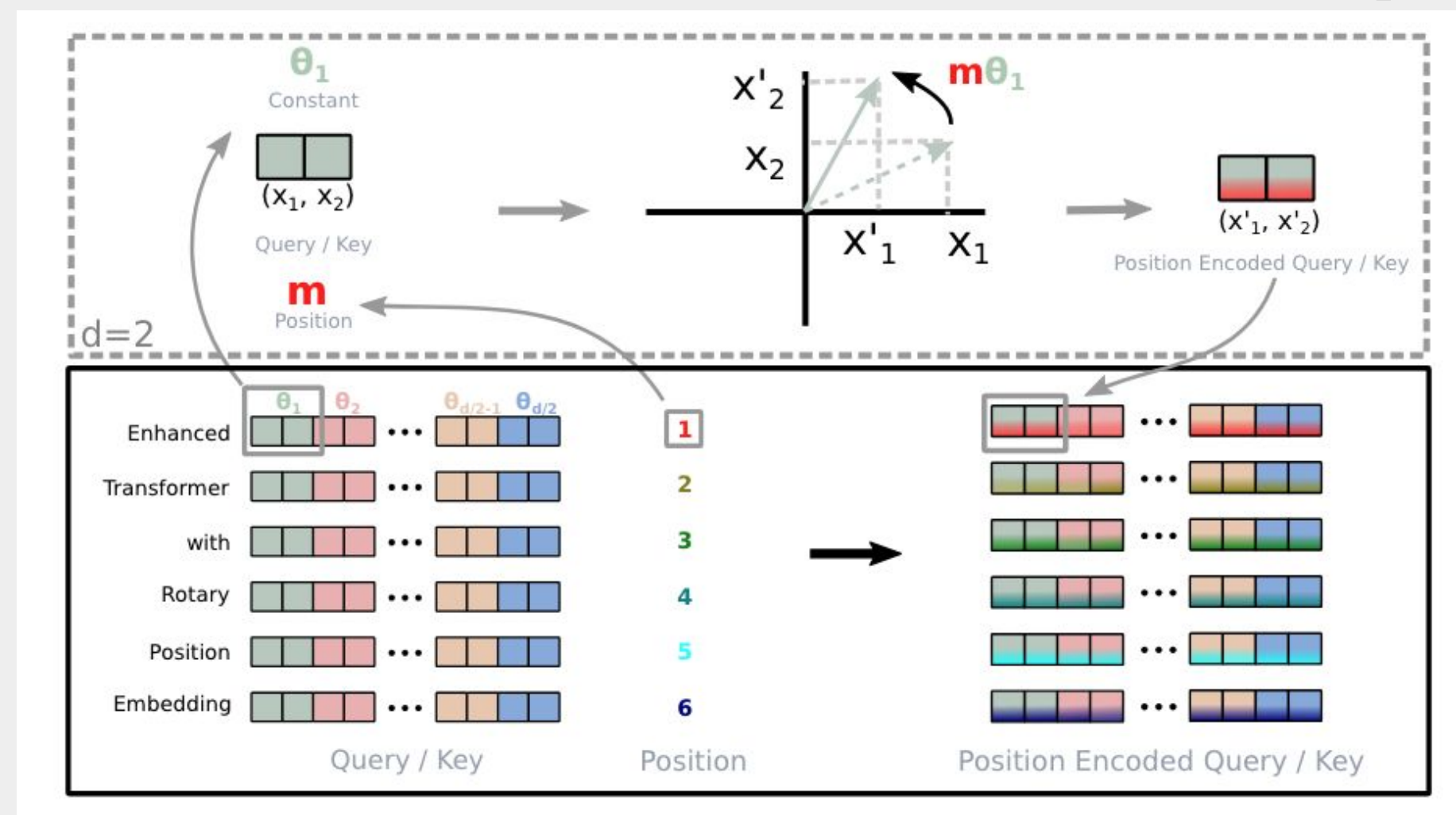


Multi-query Attention used in Gemma 2B. Image source:
<https://fireworks.ai/blog/multi-query-attention-is-all-you-need>

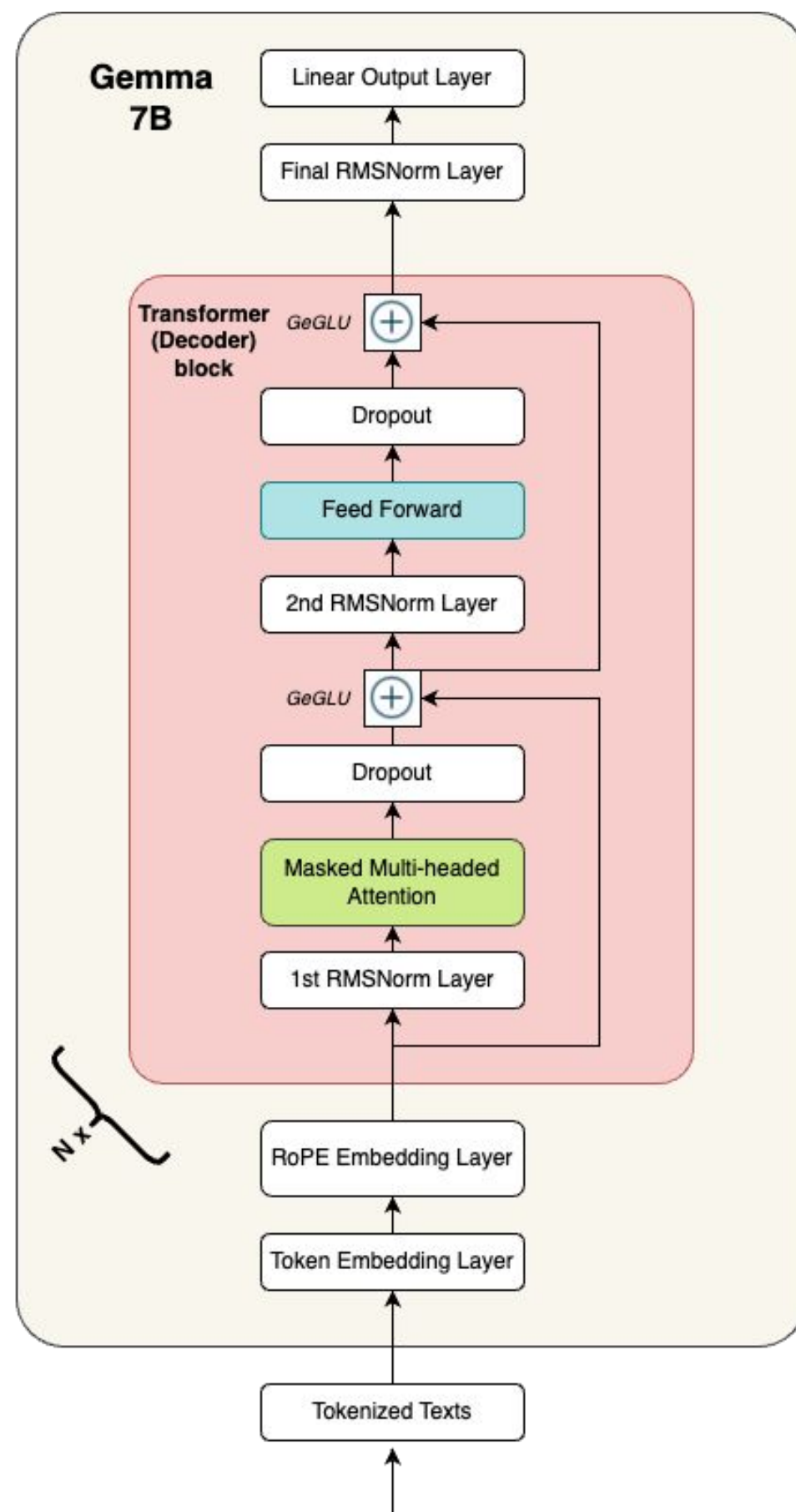
RoPE Embeddings

Su et al. (2023) proposed a novel position encoding method called **Rotary Position Embedding (RoPE)** for transformer-based language models. The key idea is to encode the absolute position information using a rotation matrix and incorporate it into the self-attention mechanism.

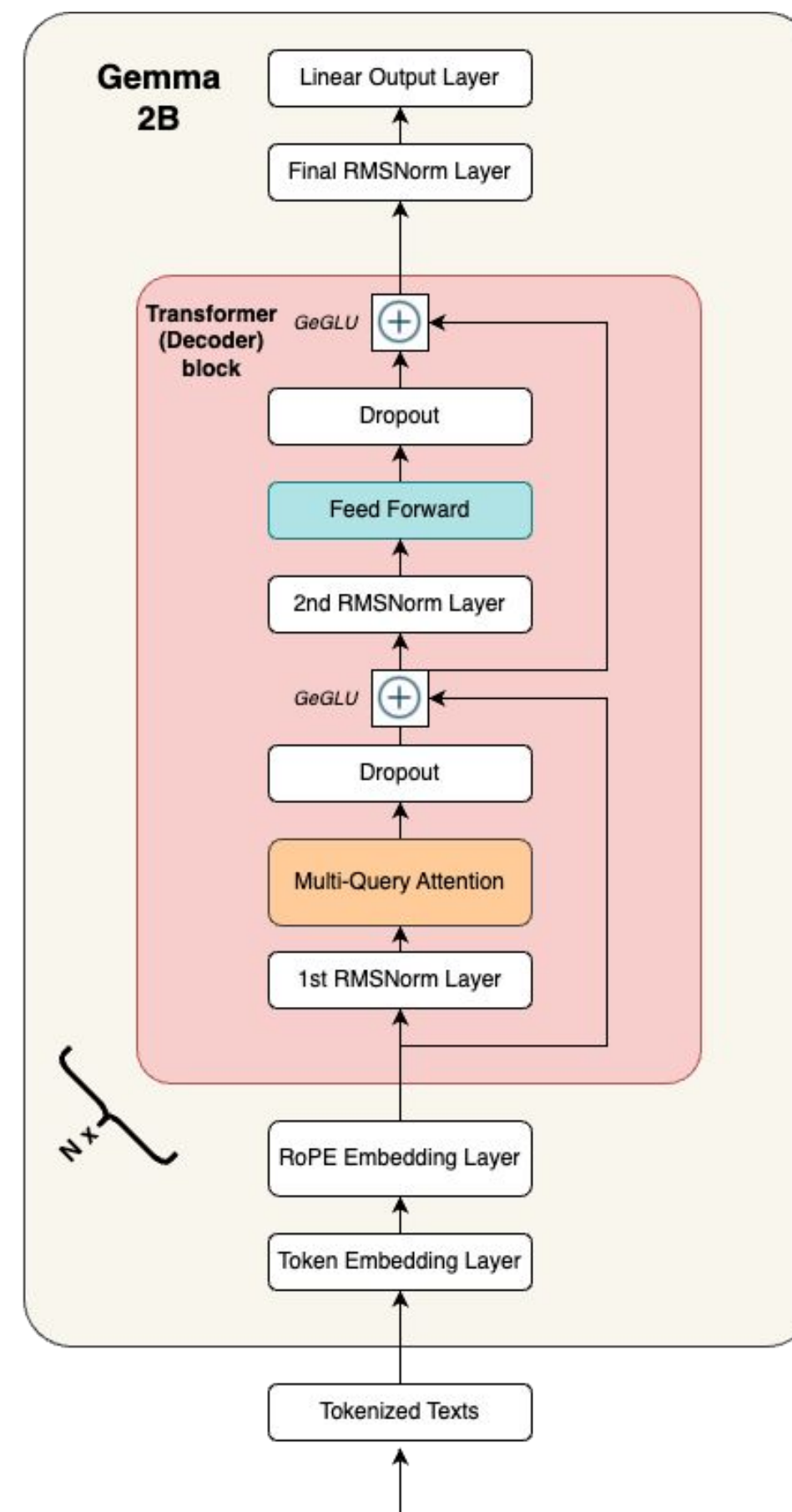
Specifically, RoPE multiplies the input word embeddings with a position-dependent rotation matrix, which enables the model to capture the relative position information between tokens. Compared to existing position encoding methods that typically add position embeddings to the input representations, RoPE provides a more efficient and theoretically grounded approach to inject positional information into the self-attention computation.



RoPE Embeddings used in Gemma 2B. Image source:
<https://arxiv.org/pdf/2104.09864>



Left: Gemma 7B
vs. Gemma 2B
(Right)

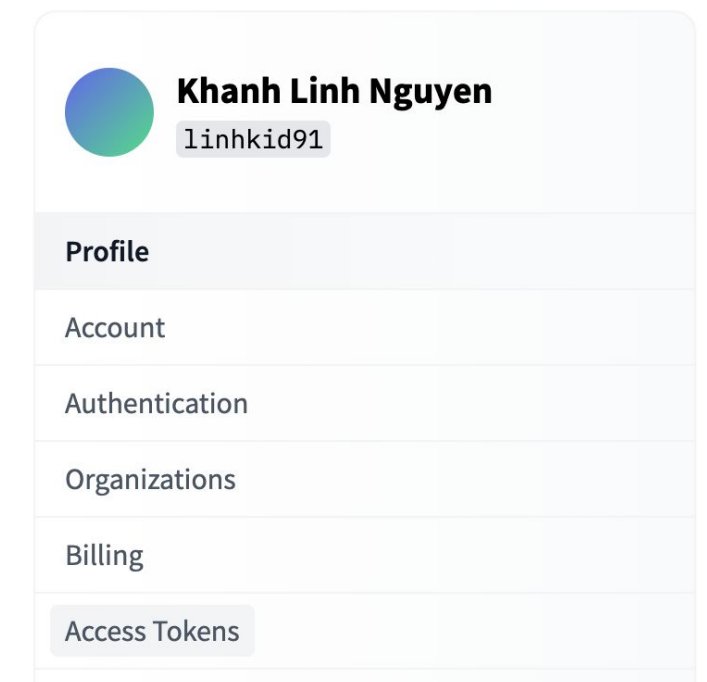
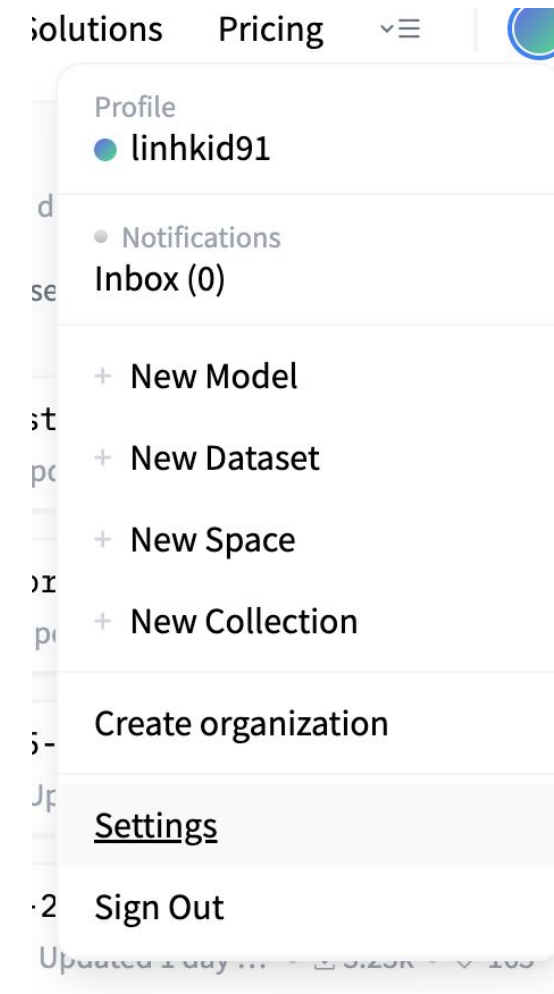


Gemma 7B and its quantized versions

Prework

Access Token

Register an account & get your access token at <https://huggingface.co/>



Create a new access token

Name

code1ab1

Type

✓ Fine-grained (custom)

Read

Write

Generate a token

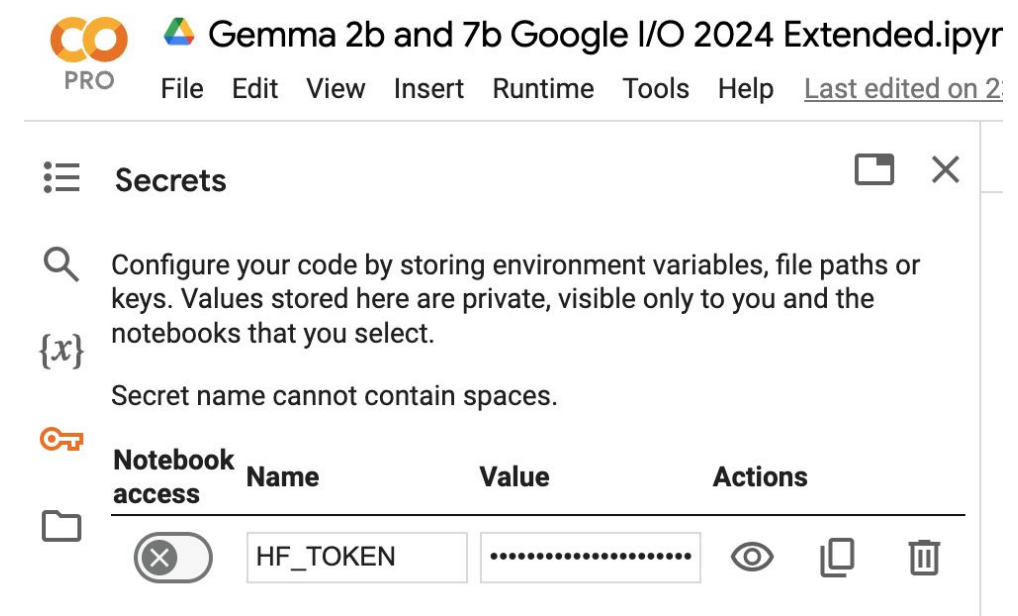
Prework

Google Colab

1/ Open <https://colab.research.google.com/> or scan this QR Code for access to the sample Notebooks



2/ Enter the access token from Huggingface or either store in under “Secret Key” on the sidebar of Google Colab



```
import os
from google.colab import userdata
os.environ["HF_TOKEN"] = userdata.get('HF_TOKEN')
```


Codelab 1

Loading and inference Gemma 7B and 2B


```
!pip install -q --upgrade transformers accelerate bitsandbytes flash_attn
```

```
from transformers import AutoTokenizer, AutoModelForCausalLM  
import torch
```

```
MODEL_NAME = "google/gemma-7b-it"  
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)  
model = AutoModelForCausalLM.from_pretrained(  
    MODEL_NAME,  
    device_map="auto"  
)
```

```
input_text = "Write me a poem about Google I/O event."  
input_ids = tokenizer(input_text, return_tensors="pt").to("cuda")
```

```
outputs = model.generate(**input_ids, max_new_tokens=256)  
print(tokenizer.decode(outputs[0]))
```



Code template
for Gemma
models loading
and inferencing

“Write me a poem about Google I/O event”

Full Precision

In the heart of Silicon Valley, a stage unfolds,
Where tech giants gather, stories untold.
Google I/O, a gathering of minds,
Where innovation takes flight, leaving its binds.

With keynote speakers, a captivating start,
They unveil the future, a digital heart.
Android, Wear OS, and Chrome,
The latest releases, a thrilling dream.

Developers swarm, with passion and zest,
Building apps and tools, at an unprecedented crest.
The halls echo with code, a symphony of creation,
As creativity blossoms, a new era of innovation.

From wearable gadgets to AI's might,
The possibilities seem endless, shining light.
The future of technology, taking shape,
A glimpse into tomorrow, a brighter escape.

So let us celebrate this event so grand,
Where dreams take flight, and ideas expand.
Google I/O, a testament to the past,
And the future that lies, built to last.

BFloat 16

In the heart of Silicon Valley, a stage unfolds,
Where tech giants gather, stories untold.
Google I/O, a moment of grace,
Where the future takes shape at an unprecedented pace.

With keynote speakers, a captivating start,
They unveil visions, ignite the heart.
Products unveiled, a glimpse of delight,
The latest innovations, shining so bright.

The halls echo with energy and zest,
As developers gather, their spirits crest.
Workshops ignite, ideas take flight,
Building the future with all their might.

From mobile apps to AI, the spectrum expands,
The power of technology in the palm of hands.
With every demo, a new story unfolds,
The potential unleashed, a tale to behold.

The energy is high, the atmosphere charged,
As the community connects, a force unmarred.
In the spirit of innovation, they share their dreams,
Building a future where anything can be seen.

So let us celebrate this day of grace,
Where the tech world comes to its place.
Google I/O, a moment to behold,
Shaping the future, stories untold.

8-bit int quant

In the heart of Silicon Valley, a stage unfolds,
Where tech giants gather, stories untold.
Google I/O, a gathering of minds,
Where innovation takes flight, leaving its binds.

With keynote speakers, a captivating start,
They unveil the future, a digital heart.
Android, Wear OS, and Chrome,
The latest advancements, a breathtaking bloom.

Developers swarm, with passion and zest,
Building apps and tools, at an unprecedented crest.
The halls echo with the hum of code,
As creativity blossoms, a vibrant ode.

From wearable gadgets to AI's grace,
The event showcases the future at an unmatched pace.
With every demo, a new horizon unfolds,
A glimpse into the future, where technology beholds.

So, let us celebrate this day of delight,
Where innovation meets passion, shining light.
Google I/O, a journey of dreams,
Where the future takes shape, it would seem.

4-bit quant

In the halls of Silicon Valley,
A stage lights up with glee,
Google I/O takes flight,
A glimpse into the future bright.

With keynote speakers bold,
And products stories untold,
The audience listens with awe,
As innovation takes a bow.

From Android to AI,
The latest trends take flight,
Developers gather strength,
To build the future with intent.

The halls echo with a buzz,
As attendees mingle and discuss,
Ideas spark, connections bloom,
And the spirit of innovation blooms.

So let us celebrate this day,
Where technology finds its way,
To touch our lives in a profound way,
Google I/O, a magical sway

Flash Attention

In the heart of Silicon Valley, a stage unfolds,
Where tech giants gather, stories untold.
Google I/O, a moment of grace,
Where innovation meets a brighter future's embrace.

With keynote speakers, a captivating start,
They unveil visions, ignite the heart.
Products unveiled, a glimpse of delight,
The future of technology, shining so bright.

From Android to Chrome, the latest trends,
Developers gather, their spirits ascend.
With workshops and demos, they learn and grow,
Building apps that will touch and flow.

The halls abuzz with energy and zest,
As attendees connect, their spirits crest.
In the spirit of collaboration, they share their might,
Building a future where technology takes flight.

As the sun sets, the event draws to a close,
Memories made, a lasting glow.
Google I/O, a time for reflection,
Where the future takes shape, beyond imagination.

So let us celebrate this day of grace,
Where innovation meets a brighter place.
May Google I/O inspire us all,
To build a future where technology stands tall.

Do the same with Gemma 2B. Please also follow the instructions in the provided notebook

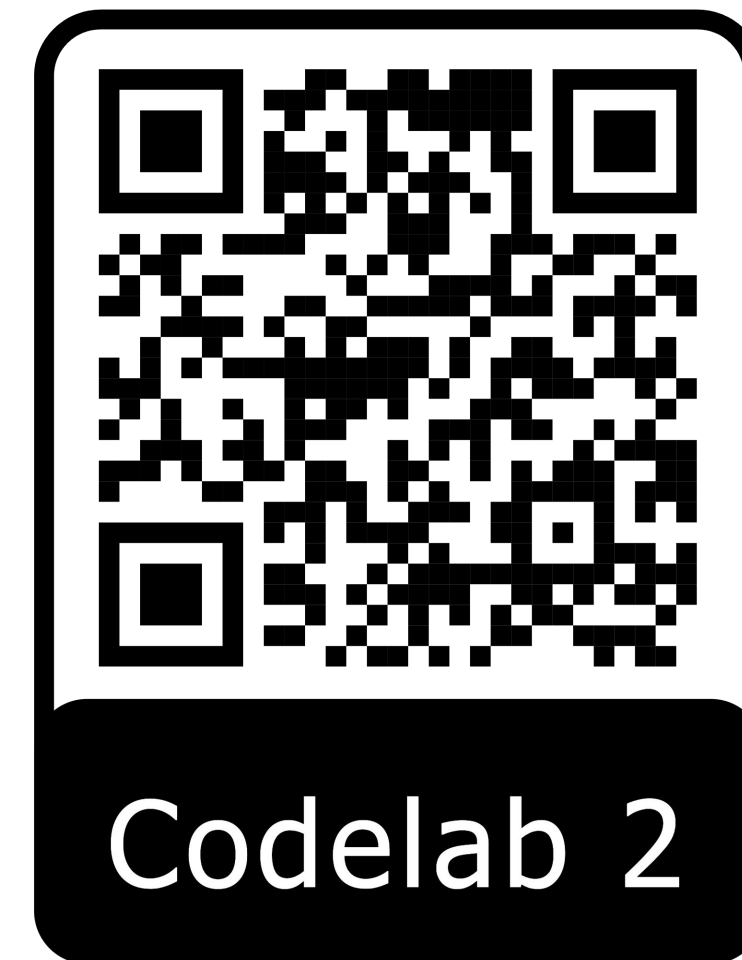
Fine-tune your first Gemma 2B with Unsloth, PEFT and LoRA

Prework

Google Colab

1/ Open <https://colab.research.google.com/> or scan this QR Code for access to the sample Notebooks (same as the previous lab)

2/ Enter the access token from Huggingface or either store in under “Secret Key” on the sidebar of Google Colab



```
import os
from google.colab import userdata
os.environ["HF_TOKEN"] = userdata.get('HF_TOKEN')
```


Codelab 2

Finetuning Gemma 2B with Unsloth, PEFT and LoRA

```
!pip install -q --upgrade transformers accelerate bitsandbytes flash_attn datasets  
peft  
!pip install -q "unsloth[colab-new] @  
git+https://github.com/unslothai/unsloth.git"  
!pip install -q --no-deps "xformers<0.0.26" trl peft accelerate bitsandbytes  
!pip install -q flash_attn datasets
```

✓ Initialize Unsloth and load Gemma 2B

```
[ ] 1  from unsloth import FastLanguageModel
    2  import torch
    3  major_version, minor_version = torch.cuda.get_device_capability()
    4
    5  max_seq_length = 2048
    6  dtype = None
    7  load_in_4bit = True
    8
    9  model, tokenizer = FastLanguageModel.from_pretrained(
10      |      model_name="google/gemma-2b-it",
11      |      max_seq_length=4096,
12      |      dtype=dtype,
13      |      load_in_4bit=load_in_4bit,
14  )
```

✓ LORA & PEFT loading

```
[ ] 1 model = FastLanguageModel.get_peft_model(  
    2     model,  
    3     r=16,  
    4     target_modules=["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj"],  
    5     lora_alpha=16,  
    6     lora_dropout=0,  
    7     bias="none",  
    8     use_gradient_checkpointing=True,  
    9     random_state=1357,  
   10     use_rslora=False,  
   11     loftq_config=None,  
   12 )
```

➞ Unsloth 2024.6 patched 18 layers with 18 QKV layers, 18 0 layers and 18 MLP layers.

```

1  prompt = ""Based on given instruction and input, generate an appropriate response
2
3  ### Instruction:
4  {}
5
6  ### Input:
7  {}
8
9  ### Response:
10 {}
11 ""
12
13 EOS_TOKEN = tokenizer.eos_token # Must add EOS_TOKEN
14 def formatting_prompts_func(examples):
15     instructions = examples["instruction"]
16     contexts = examples["input"]
17     responses = examples["output"]
18     texts = []
19
20     for i,j,k in zip(instructions, contexts, responses):
21         text = prompt.format(i,j,k) + EOS_TOKEN
22         texts.append(text)
23     return { "text" : texts, }
24 pass
25
26 from datasets import load_dataset
27 dataset = load_dataset("starfishmedical/webGPT_x_dolly", split = "train")
28 dataset = dataset.map(formatting_prompts_func, batched = True)

```

Load your
customized
Datasets

✓ Finetuning Gemma 2b

```
1  from trl import SFTTrainer
2  from transformers import TrainingArguments
3
4  trainer = SFTTrainer(
5      model = model,
6      tokenizer = tokenizer,
7      train_dataset = dataset,
8      dataset_text_field = "text",
9      max_seq_length = max_seq_length,
10     dataset_num_proc = 2,
11     packing = False,
12     args = TrainingArguments(
13         per_device_train_batch_size = 8,
14         gradient_accumulation_steps = 16,
15         warmup_steps = 2,
16         max_steps = 10,
17         learning_rate = 0.0005,
18         fp16 = not torch.cuda.is_bf16_supported(),
19         bf16 = torch.cuda.is_bf16_supported(),
20         logging_steps = 1,
21         optim = "adamw_8bit",
22         weight_decay = 0.01,
23         lr_scheduler_type = "linear",
24         seed = 1357,
25         output_dir = "outputs",
26     ),
27 )
28
29 # Training
30 trainer_stats = trainer.train()
```

Feel free to
change your
parameters!

✓ Ask your new assistant!

```
[ ] 1 inputs = tokenizer(
    2     [
    3         prompt.format(
    4             "Provide a detailed explanation phobias and its varations", # instruction
    5             " The goal is to offer a clear and informative account in medical terms", # context
    6             " ", # response
    7         )
    8     ] * 1,
    9     return_tensors="pt",
10 ).to("cuda")
11
12 # Generate response
13 from transformers import TextStreamer
14 text_streamer = TextStreamer(tokenizer)
15 _ = model.generate(*inputs, streamer=text_streamer, max_new_tokens=2048)
```

Change the
prompts if you
like

Gemma 2: Future Directions and Possibilities

Gemma 2

Evaluation Benchmark Against Other Generative Language Models

	BENCHMARK	METRIC	Gemma 2		Llama 3		Grok-1
			9B	27B	8B	70B	314B
General	MMLU	5-shot, top-1	71.3	75.2	66.6	79.5	73.0
Reasoning	BBH	3-shot, CoT	68.2	74.9	61.1	81.3	–
	HellaSwag	10-shot	81.9	86.4	82	–	–
Math	GSM8K	5-shot, maj@1	68.6	74.0	45.7	–	62.9 (8-shot)
	MATH	4-shot	36.6	42.3	–	–	23.9
Code	HumanEval	pass@1	40.2	51.8	–	–	63.2 (0-shot)

Features

01

What is Gemma 2

1. Google's next generation open AI model, available in 9B and 27B parameter sizes
2. Much more computationally efficient than Gemma 1 and other models, enabling low-cost deployment
3. Runs significantly faster at inference than previous Gemma models and alternatives
4. Easier to integrate into existing workflows than Gemma 1 via broad framework compatibility

02

High Performance

1. Offers best-in-class performance for its size, with the 27B model competitive with those 2x its size
2. Extremely fast inference speeds across different hardware, from gaming laptops to cloud setups
3. Highly efficient inference, able to run on a single GPU/TPU to significantly reduce deployment costs

03

Highly Accessible

1. Open and accessible under a permissive license, allowing commercial use of innovations built on it
2. Compatible with major AI frameworks like Hugging Face, JAX, PyTorch, TensorFlow, optimized for NVIDIA
3. Robust safety mitigations and responsible AI practices built-in during training and evaluation

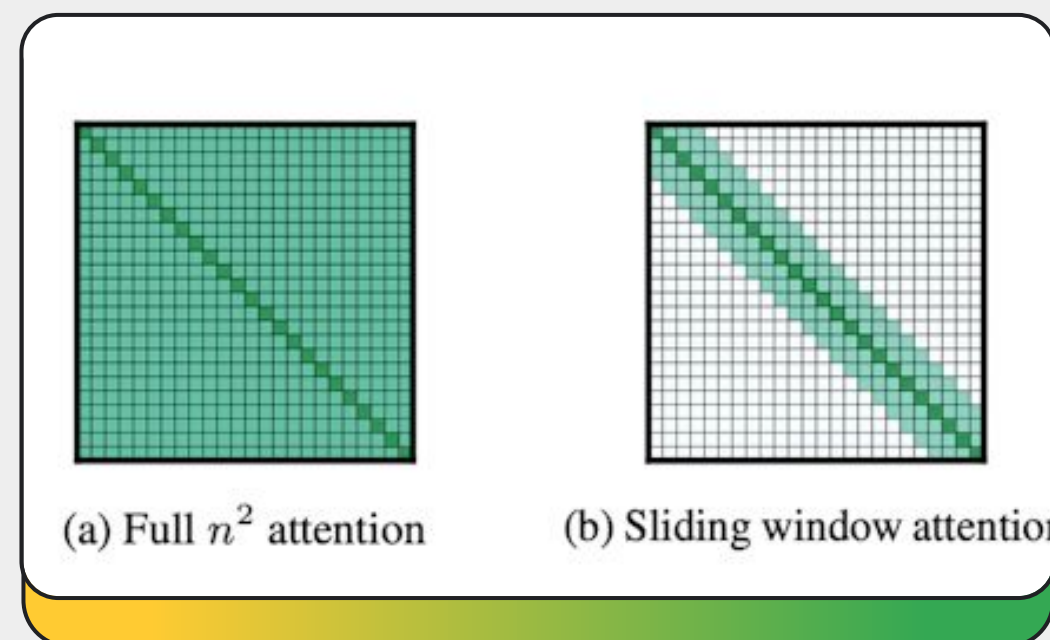
Improvements

Gemma 2 vs. Gemma 1: Key Differences & Improvement.



Trained on more data

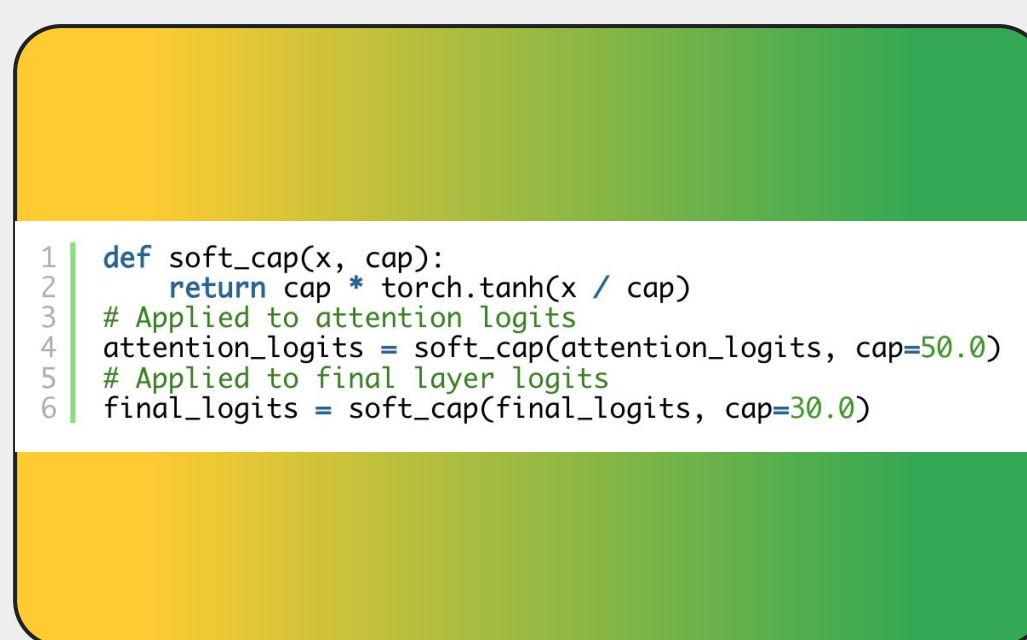
Gemma 2 27B: Trained on 13 trillion tokens
Gemma 2 9B: Trained on 8 trillion tokens



Sliding Window Attention

Gemma 2 alternates between a local sliding window attention and global attention in every other layer.

The sliding window size of local attention layers is set to 4096 tokens, while the span of the global attention layers is set to 8192 tokens.



Soft-capping

To improve training stability and performance, Gemma 2 introduces a soft-capping mechanism.

This technique prevents logits from growing excessively large without hard truncation, maintaining more information while stabilizing the training process.

What to look for

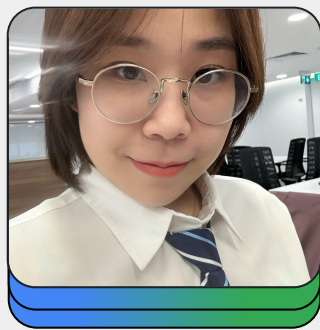
1. Continue exploring new architectures and developing specialized Gemma variants to tackle an even wider range of AI tasks and challenges.
2. Google mentions an upcoming 2.6B parameter Gemma 2 model that is in development. This smaller model size is designed to "further bridge the gap between lightweight accessibility and powerful performance."



“Now, Gemma 2 will help developers get even more ambitious projects off the ground, unlocking new levels of performance and potential in their AI creations. We'll continue to explore new architectures and develop specialized Gemma variants to tackle a wider range of AI tasks and challenges. This includes an upcoming 2.6B parameter Gemma 2 model, designed to further bridge the gap between lightweight accessibility and powerful performance.”

Q&A

Thank You



Nguyen Khanh Linh (she/her)

Founder, AI Engineer



<https://github.com/linhkid>



linh@neuropurrfectai.co

Google Extended

Hanoi, July 2024