Gemma

Google’s family of

Open Source LLMs

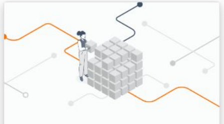
TensorFlow
User Group São Paulo

Season 2024

Gemma: from zero to hero

Lorenzo Cesconetto
Sr Software Engineer

Pedro Gengo
ML GDE



Preparar dados

Criar modelos de ML

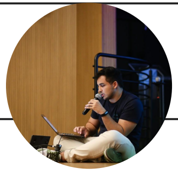
Implantar modelos

Implementar MLOps

Organization:

Alex Mansano

André Lopes



Pedro Gengo

Vinicius Caridá

Speakers

Pedro Gengo:
- GDE: Machine Learning
- Sr Machine Learning Eng @ Bestever
- Former: Sticker Mule, Deepcell, Meli, Itaú Unibanco SA

Lorenzo Cesconetto:
- Sr Software Eng @ Stealth Biotech
- Masters: AI & ML from ITA
- Former co-founder @ Triple AI
- Former: Deepcell, Itaú Unibanco SA, Embrapa

Agenda

1. Family: Four models in total
2. Usage & important facts
3. Internal architecture
4. Sources and relevant links
5. Live demo!

Four models in total: two sizes & two fine-tunings

Gemma-7B

- 8.54 billion parameters (rename proposal)
  - +20% larger than Llama-7B
- Trained on 6 trillion tokens
- Intended to run on GPUs
- ~18 GB of RAM (bfloat16 or float16)
- Particularly good at math reasoning and coding (beats CodeLLaMA-7B)

Gemma-2B

- 2.5 billion parameters
- Trained on 2 trillion tokens
- Intended run on CPUs and edge devices (mobile)
- ~5Gb of RAM (bfloat16 or float16)

Two versions:

- Pretrained base model
- Instruct fine-tuned

Training:

- 8,192 tokens of context
- English-language web documents, mathematics, and code snippets

Usage & important facts

License:
- Permits commercial
- Prohibits: copyright infringement, generating miss information, sexual

Performance:
- Gemma-7B: 64.56 on MMLU. Does better than other open source models, e.g., Llama2-7B and Mistral-7B.
- Gemma-2B performs worse than models of similar size, e.g., 2.7-billion-parameter Phi-2.

Architecture:
- Based on Gemini's topology
- But it's **not** multi-modal

Additional info:
- Check out: Technical report.
- Little detail on base training dataset / preprocessing, and SFT / RLHF.

Open LLM Leaderboard - by Hugging Face

Model	License	Commercial use?	Pretraining size [tokens]	Leaderboard score ⬇️
LLama 2 70B Chat (reference)	Llama 2 license	✅	2T	67.87
Gemma-7B	Gemma license	✅	6T	63.75
DeciLM-7B	Apache 2.0	✅	unknown	61.55
PHI-2 (2.7B)	MIT	✅	1.4T	61.33
Mistral-7B-v0.1	Apache 2.0	✅	unknown	60.97
Llama 2 7B	Llama 2 license	✅	2T	54.32
Gemma 2B	Gemma license	✅	2T	46.51

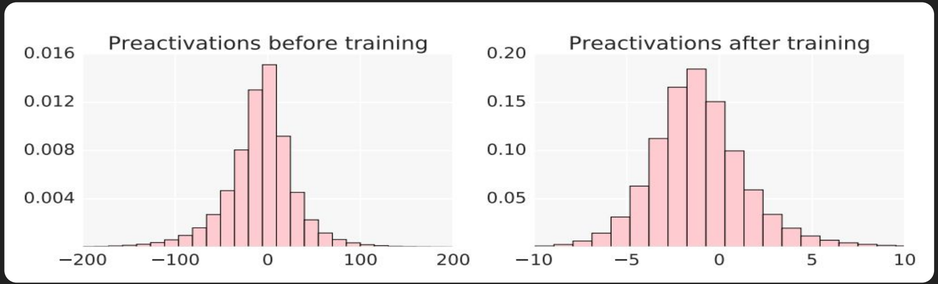
Source: Open_LLM_Leaderboard and Welcome Gemma

Internal architecture: Activation functions

- Uses GeGLU to add non-linearity to the neural net.
- GeGLU was introduced by this [paper] in 2020.
- It's a combination of GeLU and GLU.

GeLU: Gaussian error Linear Unit

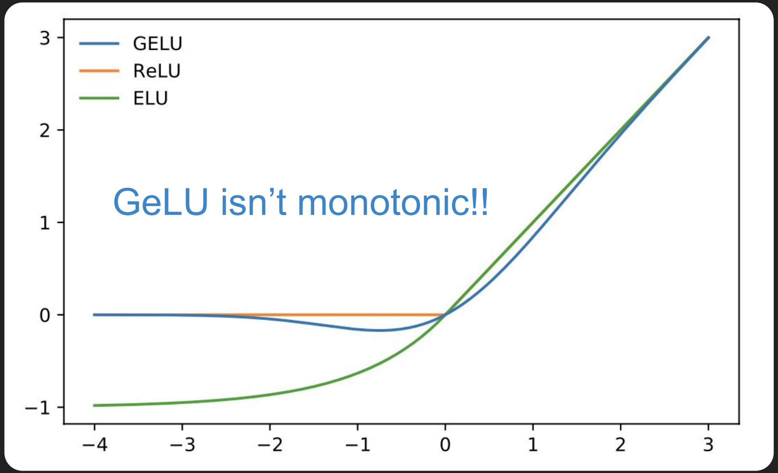
- Inspired by ReLU (deterministic) and the dropout (stochastic) technique.
- It’s used in GPT-3, BERT and other household names.
- Introduced by this paper in 2016.
  - “We choose this distribution since neuron inputs tend to follow a normal distribution, especially with Batch Normalization"
  - “GELU weights its input depending upon how much greater it is than other inputs”
  - “GELU has a probabilistic interpretation given that it is the expectation of a stochastic regularizer”

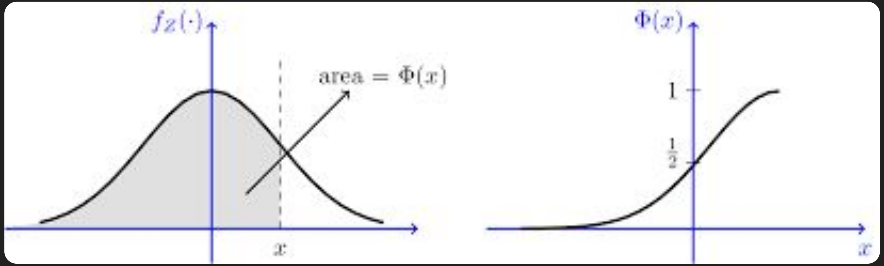


$$\text{ReLU} = \begin{cases} 0 \times x, \text{ if } x < 0 \\ 1 \times x, \text{ if } x > 0 \end{cases}$$

GeLU: Gaussian error Linear Unit

- Inspired by ReLU (deterministic) and the dropout (stochastic) technique.
- It's used in GPT-3, BERT and other household names.
- Introduced by this [paper] in 2016.
  - "We choose this distribution since neuron inputs tend to follow a normal distribution, especially with Batch Normalization"
  - "GELU weights its input depending upon how much greater it is than other inputs"
  - "GELU has a probabilistic interpretation given that it is the expectation of a stochastic regularizer"





Hard gate! GeLU makes it smoother

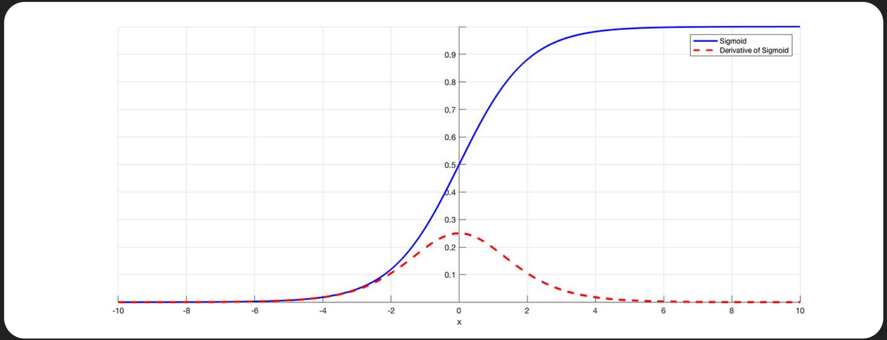
$$\text{ReLU} = \begin{cases} \boxed{0} \times x, \text{ if } x < 0 \\ \boxed{1} \times x, \text{ if } x > 0 \end{cases}$$

$$\text{GeLU} = x \times \Phi(x)$$

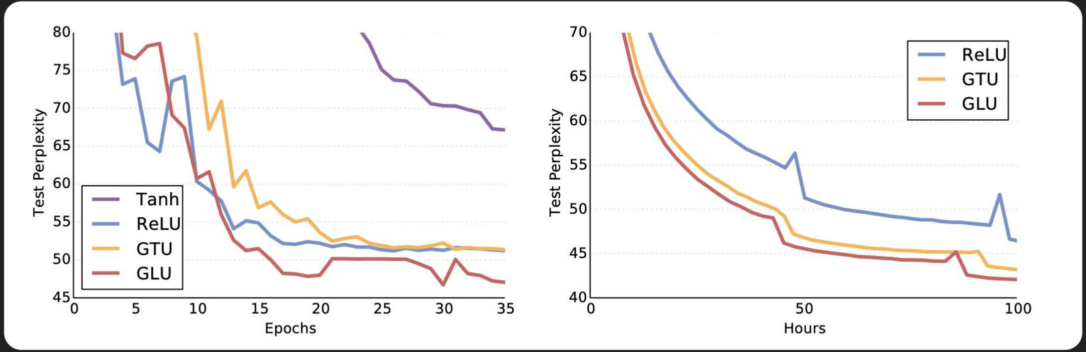
where $\Phi(x) = P(X \leq x)$, $X \sim N(0, 1)$

GLU: Gated Linear Unit

- Inspired by LSTM's gating mechanism.
- Introduced by this [paper] in 2016.
- Offers great gradient flow between layers, which is necessary for training deep neural nets.
- Faster convergence.



$$\text{GLU} = (\mathbf{X} * \mathbf{W} + b) \otimes \sigma(\mathbf{X} * \mathbf{V} + c)$$



Let's look at the derivative:

$$\nabla[\mathbf{X} \otimes \sigma(\mathbf{X})] = \boxed{\nabla \mathbf{X} \otimes \sigma(\mathbf{X})} + \mathbf{X} \otimes \sigma'(\mathbf{X}) \nabla \mathbf{X}$$

→ Gradient flows across layers (notice this is plain sigmoid, not its derivative)

Convolutions in GLU

Differently from images, in NLP filters (kernels) will span the entire row, i.e., the entire token embedding.

0.8	…	1.5
2	…	-0.2

Hi	0.4	…	0.01
how	1	…	-1.2
are	1.1	…	0.7
you	-0.2	…	0.6
?	0.8	…	0.0

Convolutions in GLU

Differently from images, in NLP filters (kernels) will span the entire row, i.e., the entire token embedding.

Hi	0.4	…	0.01
how	1	…	-1.2
are	1.1	…	0.7
you	-0.2	…	0.6
?	0.8	…	0.0

3

0.8	…	1.5
2	…	-0.2

Convolutions in GLU

Differently from images, in NLP filters (kernels) will span the entire row, i.e., the entire token embedding.

Hi	0.4	…	0.01
how	1	…	-1.2
are	1.1	…	0.7
you	-0.2	…	0.6
?	0.8	…	0.0

3
1.5

0.8	…	1.5
2	…	-0.2

Convolutions in GLU

Differently from images, in NLP filters (kernels) will span the entire row, i.e., the entire token embedding.

Hi	0.4	…	0.01
how	1	…	-1.2
are	1.1	…	0.7
you	-0.2	…	0.6
?	0.8	…	0.0

3
1.5
-0.3

0.8	…	1.5
2	…	-0.2

GeGLU: Gaussian error Gated Linear Unit

- Variation from GLU.
- Replace sigmoid with GeLU.
- We could use other non-linear functions such as ReLU.

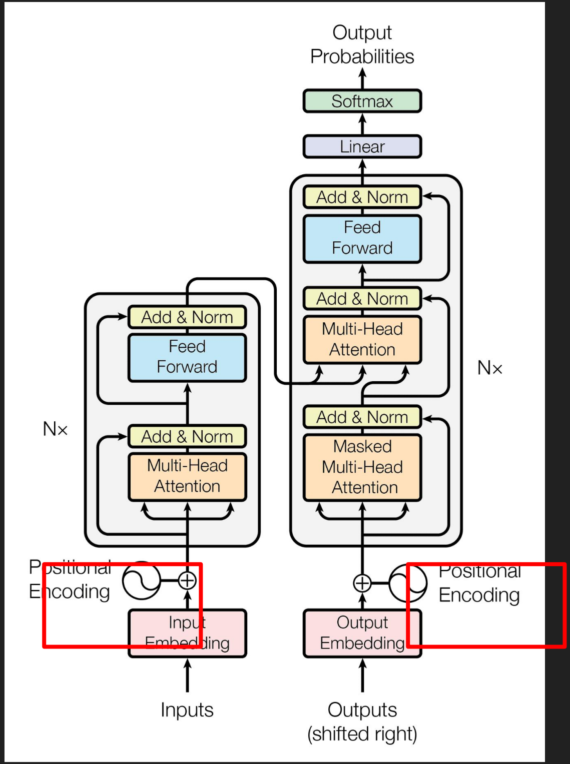
$$\text{GeLU} = x \times \mathbf{\Phi}(x)$$

$$\text{GLU} = (\mathbf{X} * \mathbf{W} + b) \otimes \boldsymbol{\sigma}(\mathbf{X} * \mathbf{V} + c)$$

$$\boxed{\text{GeGLU} = (\mathbf{X} * \mathbf{W} + b) \otimes \text{GeLU}(\mathbf{X} * \mathbf{V} + c)}$$

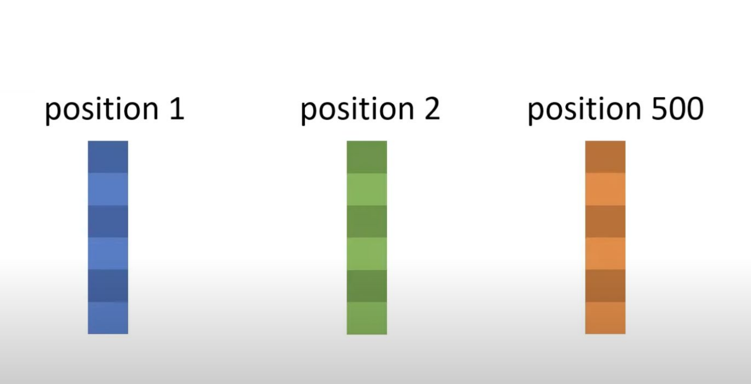
Positional Embeddings

- Multi-head self attention by itself doesn't take token position in consideration.
- To solve this issue, positional embeddings were introduced.
- We have two types of positional embeddings:
  - Absolute position: learned embeddings and sinusoidal
  - Relative position: T5 relative position bias



Absolute Positional Embeddings

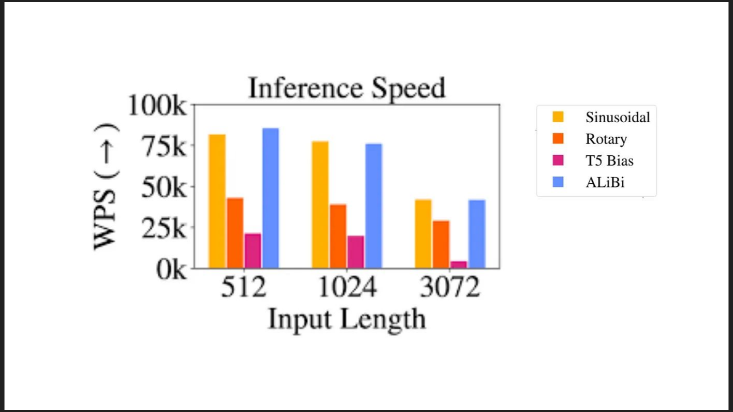
- Fixed context size, i.e., can't exploit to longer sequences.
- We expect to closer positions to be similar to each other. However, this is not what happens.

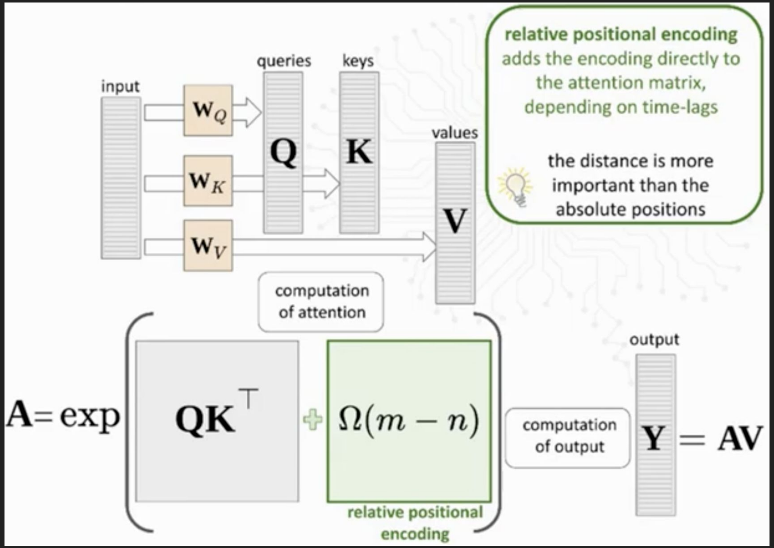


Extracted from: https://www.youtube.com/watch?v=o29P0Kpobz0

Relative Positional Embeddings

- Distance between words is more important than absolute positions;
- T5 implemented a version of it that is called relative positions bias.

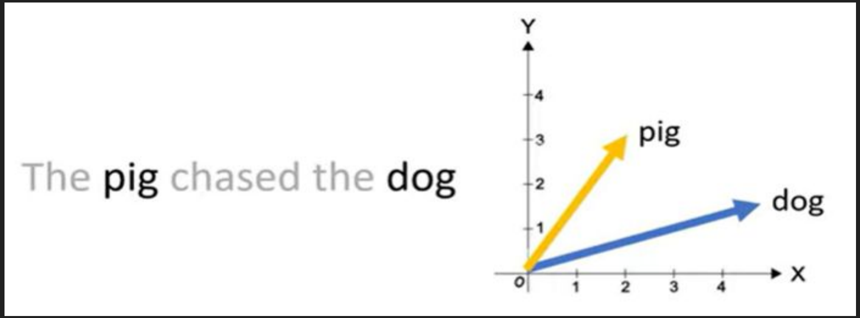


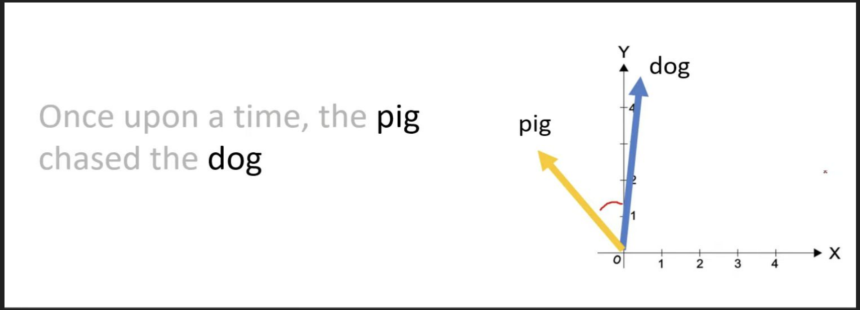


RoPE: high level explanation

- Introduced by [RoFormer: Enhanced Transformer with Rotary Position Embedding] in 2021
- Best of both worlds: absolute and relative positions

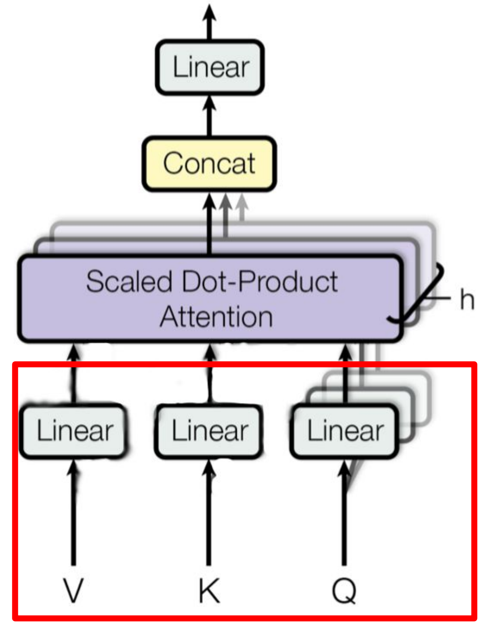
$$f_{\{q,k\}}(\boldsymbol{x}_m, m) = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} W^{(11)}_{\{q,k\}} & W^{(12)}_{\{q,k\}} \\ W^{(21)}_{\{q,k\}} & W^{(22)}_{\{q,k\}} \end{pmatrix} \begin{pmatrix} x^{(1)}_m \\ x^{(2)}_m \end{pmatrix}$$



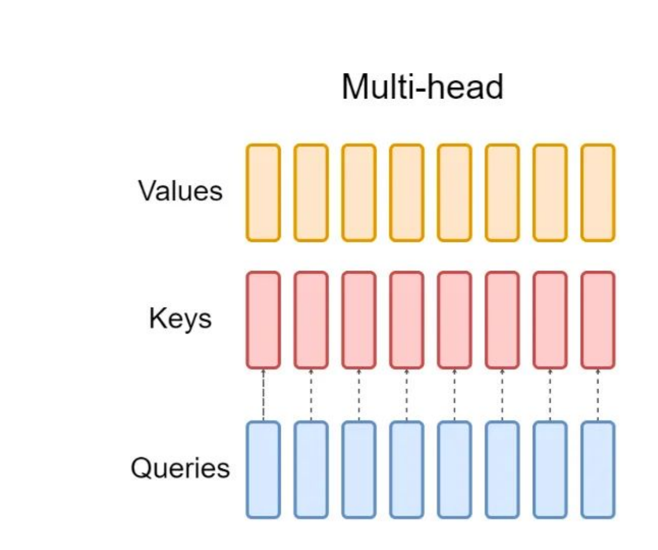


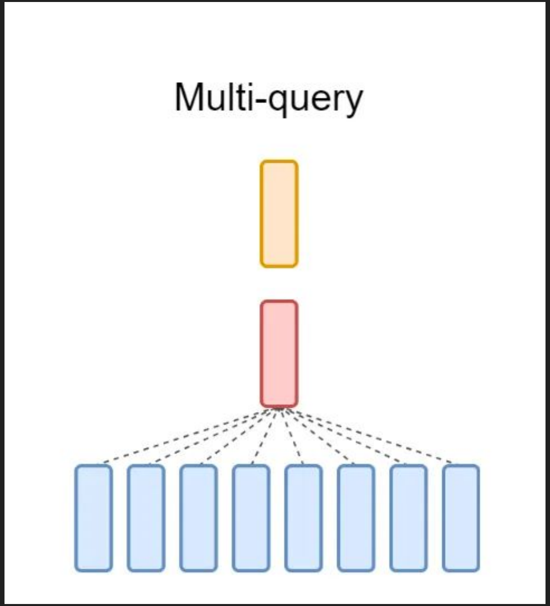
Multi-Query Attention: high level explanation

- Introduced by the paper "Fast Transformer Decoding: One Write-Head is All You Need"
- Multi-head attention consists of multiple attention layers (heads) in parallel with different linear transformations on the queries, keys, values and outputs.
- Multi-query attention is identical except that the different heads share a single set of keys and values.
- Notably, the 7B model uses multi-head attention while the 2B checkpoints use multi-query attention (with $num\_kv\_heads = 1$), based on ablation studies that revealed respective attention variants improved performance at each scale



Multi-Query Attention: high level explanation





Multi-Query Attention: high level explanation

Table 1: WMT14 EN-DE Results.

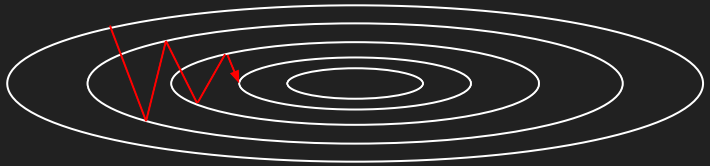
Attention Type	h	d_k, d_v	d_{ff}	ln(PPL) (dev)	BLEU (dev)	BLEU (test) beam 1 / 4
multi-head	8	128	4096	**1.424**	**26.7**	27.7 / 28.4
multi-query	8	128	5440	1.439	26.5	27.5 / **28.5**
multi-head local	8	128	4096	1.427	26.6	27.5 / 28.3
multi-query local	8	128	5440	1.437	26.5	27.6 / 28.2
multi-head	1	128	6784	1.518	25.8	
multi-head	2	64	6784	1.480	26.2	26.8 / 27.9
multi-head	4	32	6784	1.488	26.1	
multi-head	8	16	6784	1.513	25.8	

Table 2: Amortized training and inference costs for WMT14 EN-DE Translation Task with sequence length 128. Values listed are in TPUv2-microseconds per output token.

Attention Type	Training	Inference enc. + dec.	Beam-4 Search enc. + dec.
multi-head	13.2	1.7 + 46	2.0 + 203
multi-query	**13.0**	1.5 + 3.8	1.6 + 32
multi-head local	13.2	1.7 + 23	1.9 + 47
multi-query local	**13.0**	**1.5 + 3.3**	**1.6 + 16**

RMSNorm: Root Mean Squared Normalization

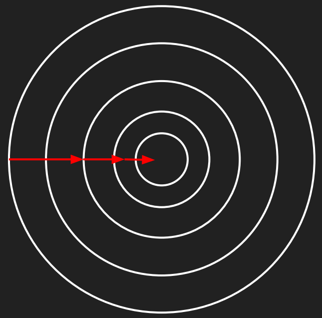
- Introduced by this paper in 2019.
- Normalization Technique.
- RMSNorm is on pair with LayerNorm, but runs 7%~64% faster.



$$\bar{a}_i = \frac{a_i}{\mathrm{RMS}(\mathbf{a})} g_i, \quad \text{where } \mathrm{RMS}(\mathbf{a}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_i^2}.$$

$$\text{where } a_i = \sum_{j=1}^{m} w_{ij} x_j$$

$$y_i = f\left(\bar{a}_i + b_i\right)$$



Sources and relevant links

- Technical report:
  - https://storage.googleapis.com/deepmind-media/gemma/gemma-report.pdf
- Official page:
  - https://ai.google.dev/gemma/
- Gemma is available on:
  - Hugging Face, Kaggle and Vertex AI.
- Detailed Model Card

Live Demo

Time to get our hands dirty :)