Attention is All You Need

overview of the transformer architecture, applications and established improvements

Felix Karg

April 6, 2023

Institute for Theoretical Informatics:
Artificial Intelligence for Materials Science

Learning Goals

- Gain familiarity with the transformer architecture and understand how it works
- Understand Tokens and Embeddings
- Comprehend how Attention works
- Become aware of common compression techniques
- Recognize limitations

Key Takeaway: Transformers can be powerful, it might be worth trying to use them.

Ask if you have questions
or anything is unclear

Overview

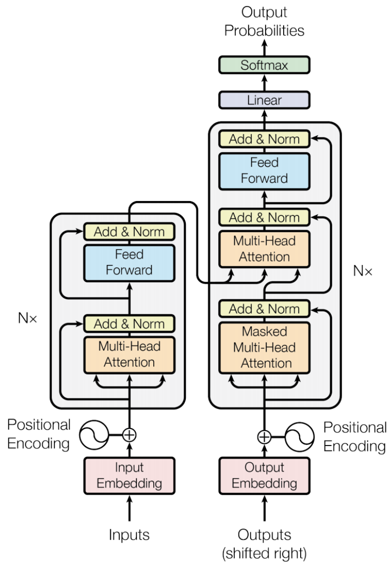


Image Source: [1]

Background

Multi-Layer Perceptron

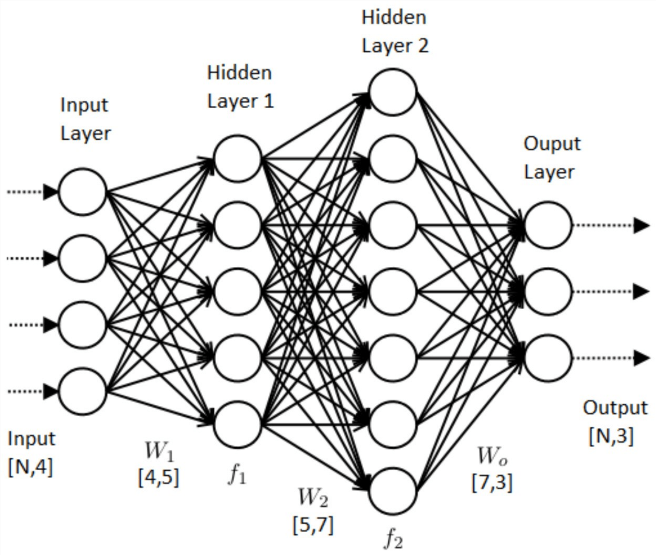


Image Source: Public Domain

Background

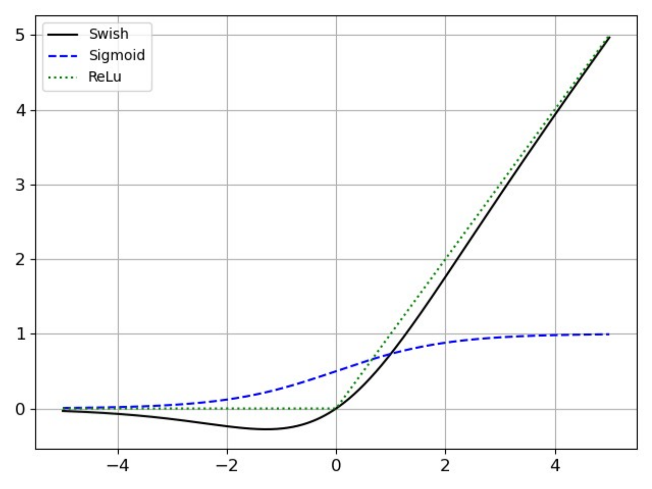
Multi-Layer Perceptron

Activation Functions

Missing Connections

Going Deeper

Common Activation Functions



$swish(x) := x * sigmoid(x)$

Image Source: [2] SwiGLU introduced by [3]

Background

Problem: neural network training results in highly specialized feature adaptations (overfitting)

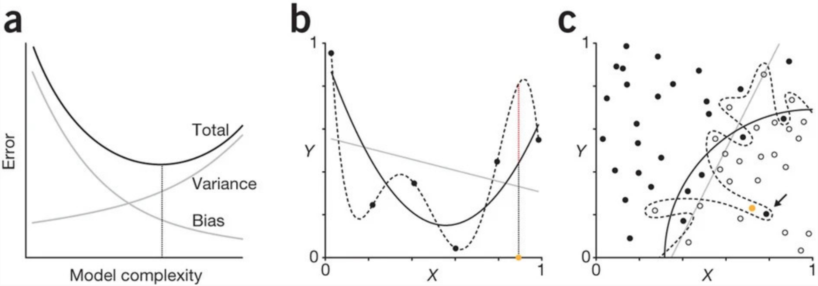


Image Source: [4]

Dropout II

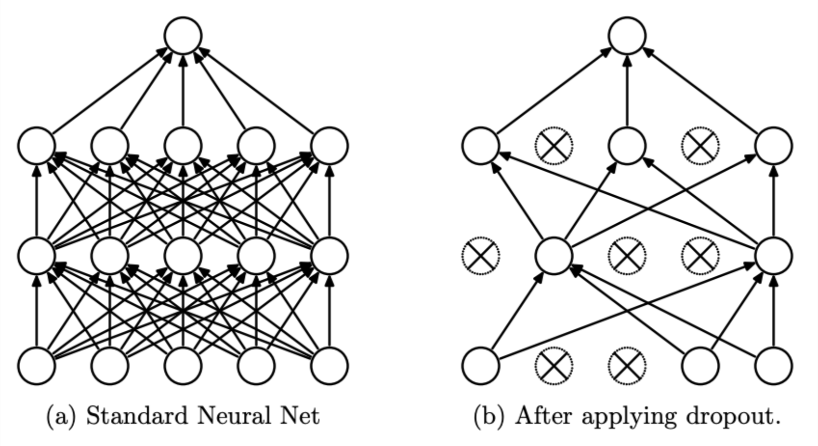


Image Source: [5]

Background

Residual Connections

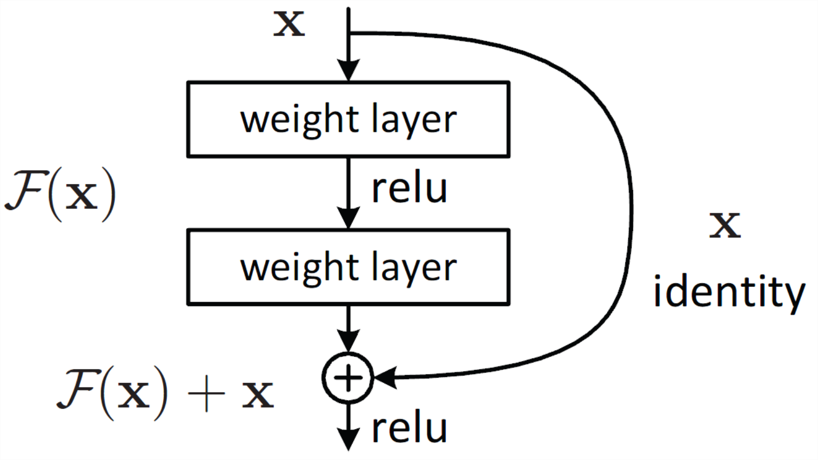


Image Source: [6]

Embedding

We are here

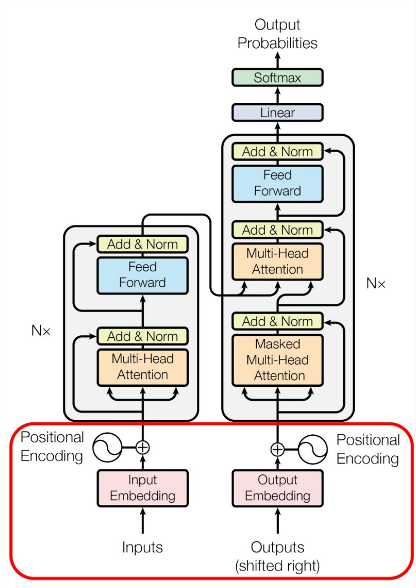


Image Adapted from [1]

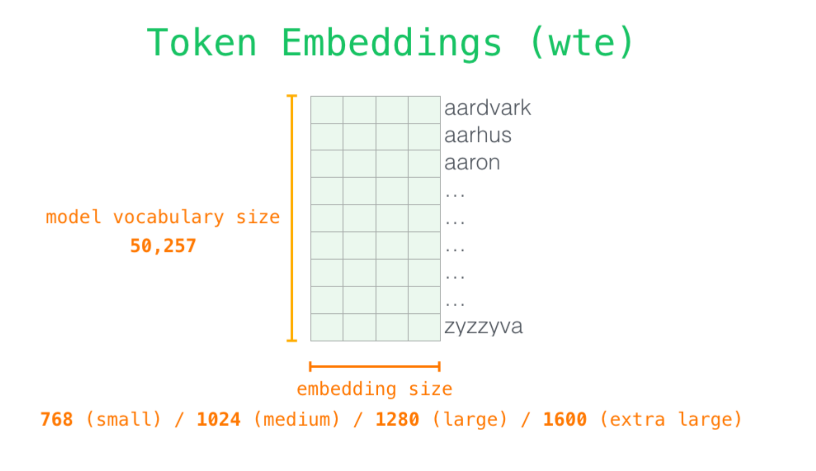
Definitions

- **Vocabulary:** List of tokens available to the tokenizer, that can be recognized and generated
- **Tokenizer:** Split the input apart using the available tokens in vocabulary
- **Embedding:** Internal high-dimensional representation of given set of tokens (learned)

The Vocabulary / Tokens are commonly learned via Byte Pair Encoding (BPE) [7].

Embedding

Input Embedding



Exemplary token to embedding encoding in GPT2.

Image Source: [8]

In Code

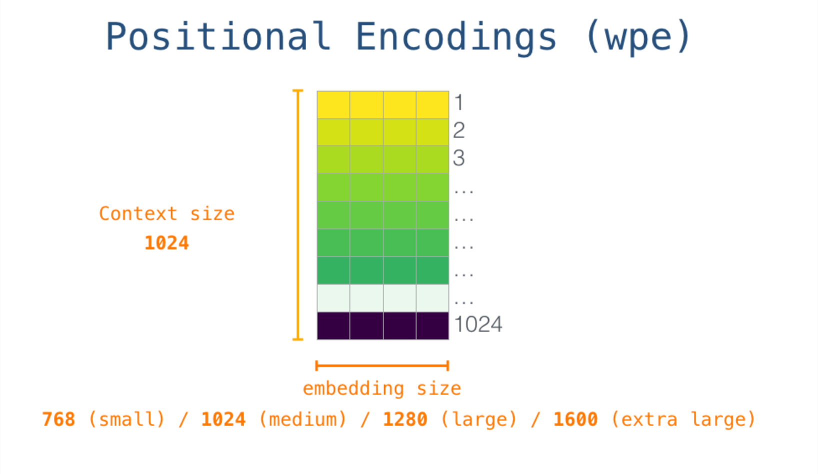
```
>>> ids = encoder.encode("Not all heroes wear capes.")
>>> ids
[3673, 477, 10281, 5806, 1451, 274, 13]

>>> encoder.decode(ids)
"Not all heroes wear capes."

>>> [encoder.decode([i]) for i in ids]
['Not', ' all', ' heroes', ' wear', ' cap', 'es', '.']
```

Embedding

Positional Encoding



Exemplary positional encoding in GPT2.

Image Source: [8]

Positional Encoding II

Visualization of a sinusoidal position encoding for the first 128 positions in 512 dimensions.

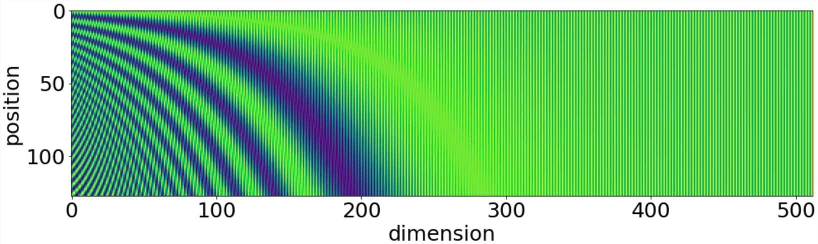


Image source: Public Domain

RoPE: Rotary Positional Encoding

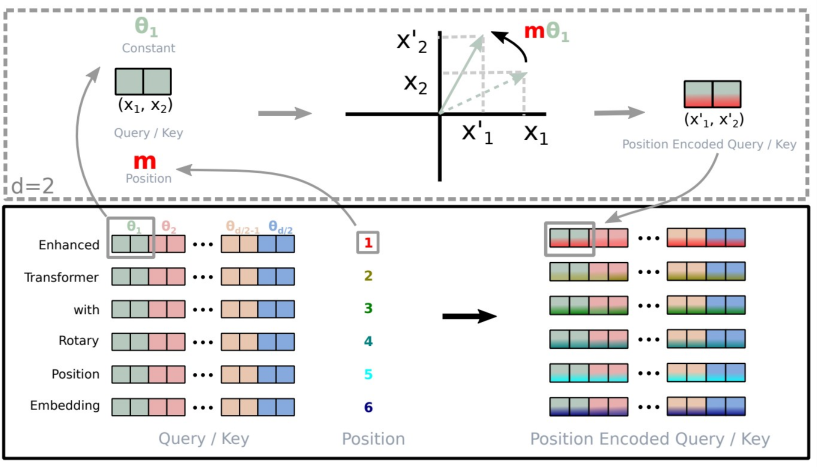


Image Source: [9]

Embedding

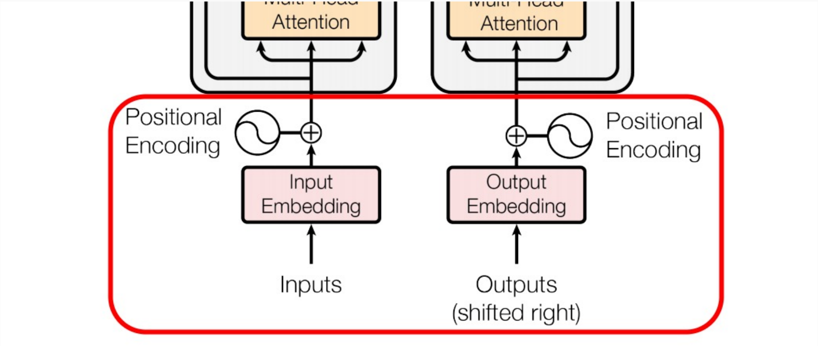
Overview

Input Embedding

Positional Encoding

Full Input Embedding

Full Input Embedding



Simple Addition. Works well due to sparse high dimensional spaces.

Image Adapted from [1]

Attention in Transformer

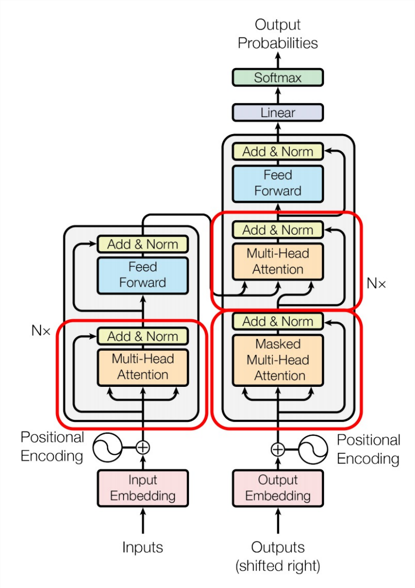


Image Adapted from [1]

Attention

Basic Attention

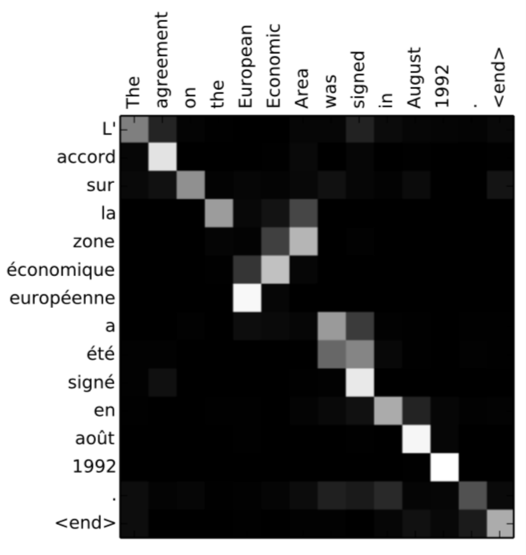


Image Source: [10]

Attention Mechanism

$$attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where

$Q = W_Q x, K = W_K x, V = W_V x,$ and d_k query-size

for self-attention and

$$Q = W_Q x, K = W_K y, V = W_V y$$

for encoder-decoder cross-attention

Scaled Dot-Product Attention

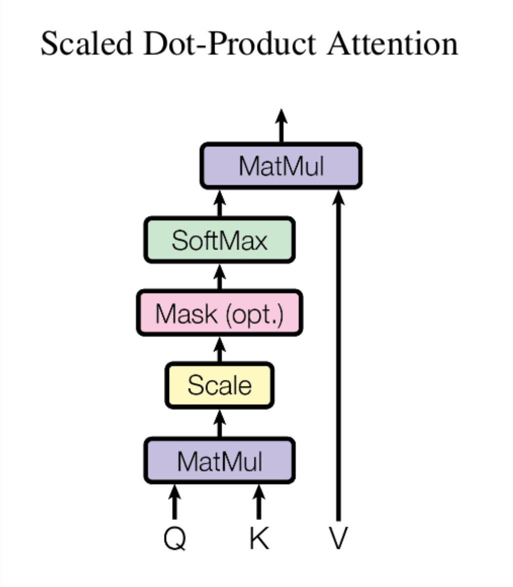


Image Source: [1]

Attention

Basic Attention

Multi-Head Attention

Masked Attention

Multi-Head Attention

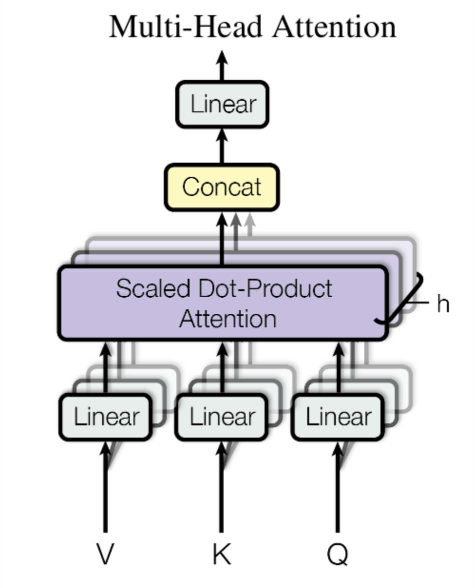


Image Source: [1]

Attention

Basic Attention

Multi-Head Attention

Masked Attention

Masked Attention

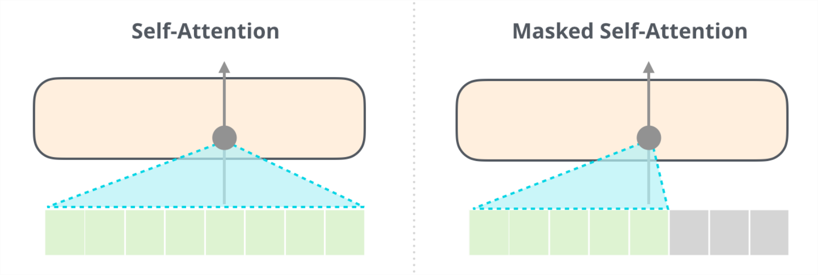
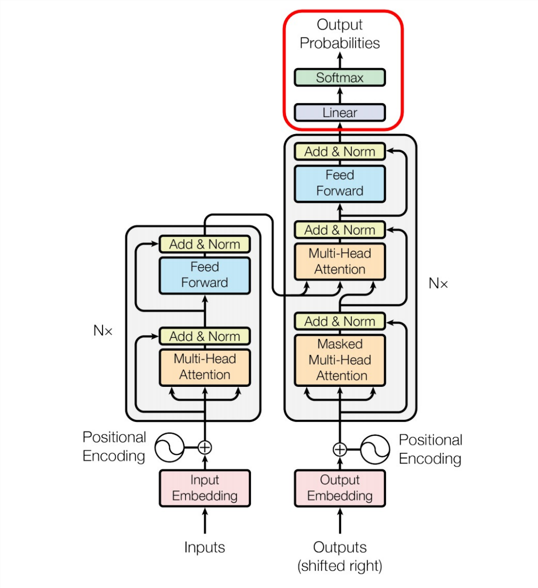


Image Source: [8]

Transformer

Output


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

Parameters

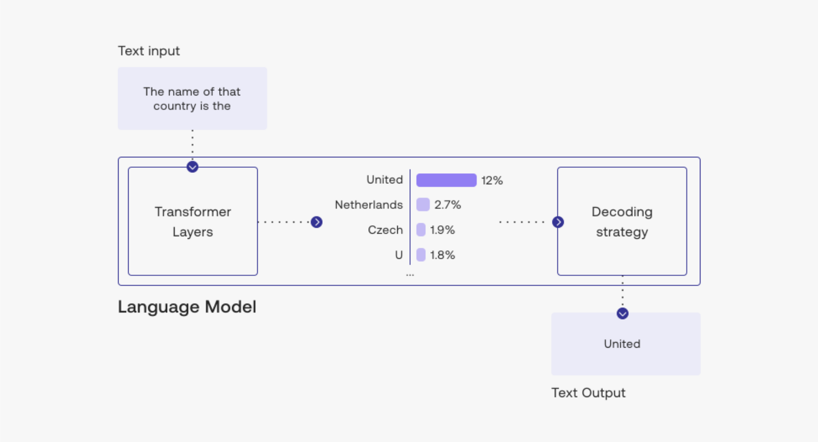


Image Source: [11]

Temperature, Top-k and Top-p

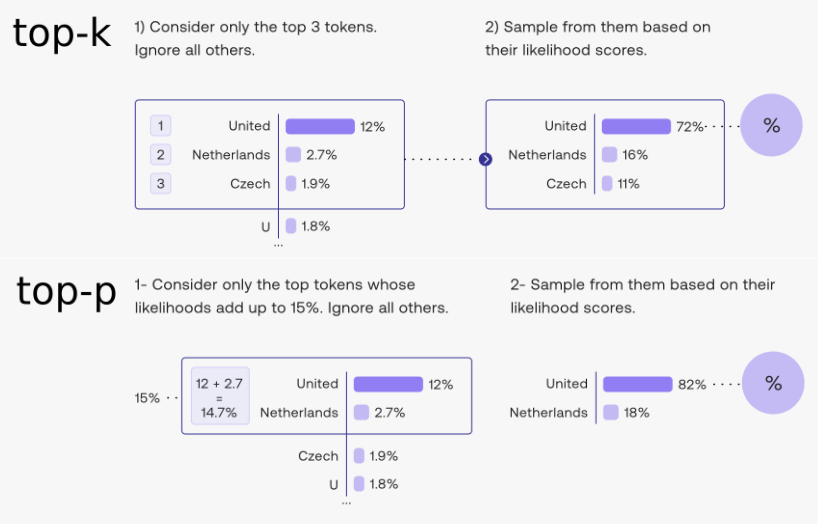


Image Source: [11]

Transformer

Dimensions

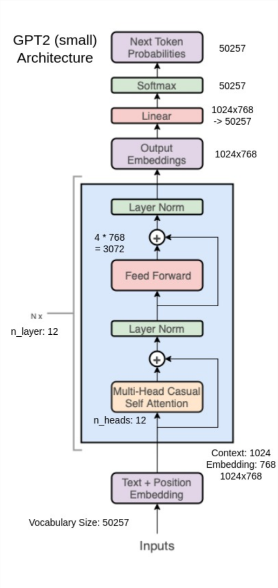


Image Adapted from: [12]

Dimensions II

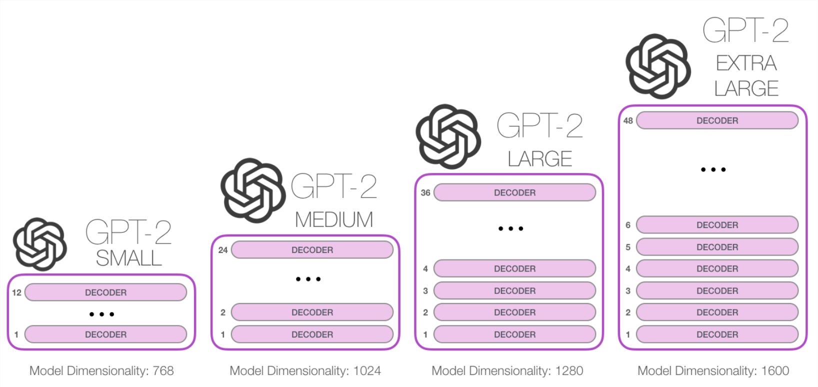


Image Source: [8]

Transformer

Full Architecture

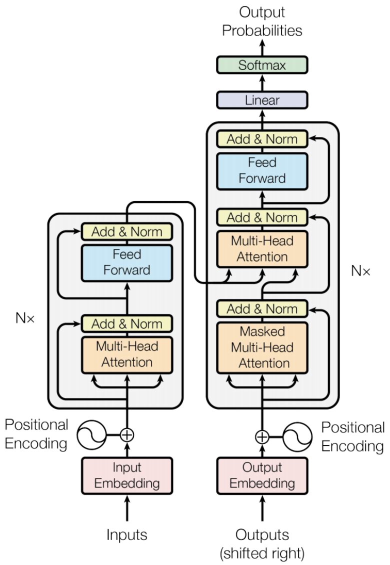


Image Source: [1]

Transformer

Output

Dimensions

Putting it all Together

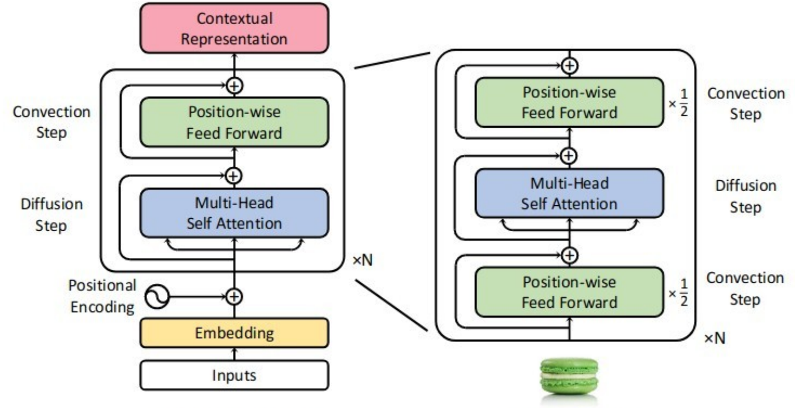
Interpretation

Architecture Improvements

One Prevalent Interpretation: Solving ODEs

'... the Transformer can be mathematically interpreted as a *numerical Ordinary Differential Equation (ODE) solver for a convection-diffusion equation in a multi-particle dynamic system.*' Lu et al., 2019 [13]

A Better ODE Solver



(a) Original Transformer (b) Macaron Layers

For solving, use a Strang-Marchuk Splitting scheme instead of Lie-Trotter

Image Source: [13] For details on solving methods see [14]

As High-Order Nonlinearity

'However, we find only a weak consistency exists between the attention weights of features and their importance. We verify the feature map multiplication that brings about **high-order non-linearity into CNNs is crucial for the effectiveness** *of attention mechanism.'* Ye et al. 2023, Towards ... [15]

In-Context Optimization

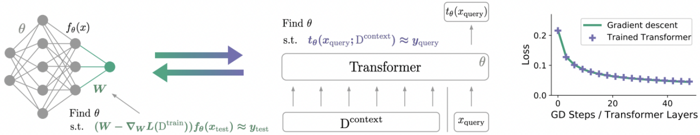


Figure 1: **Illustration of our hypothesis: gradient-based optimization and attention-based in-context learning are equivalent.** *Left*: Learning a neural network output layer by gradient descent on a dataset $\mathrm{D}^{\text{train}}$. The task-shared meta-parameters θ are obtained by meta-learning with the common goal that after adjusting the neural network output layer, the model generalizes well on unseen data. *Center*: Illustration of a Transformer that adjusts its query prediction on the data given in-context i.e. $t_\theta(x_{\text{query}}; \mathrm{D}^{\text{context}})$. The weights of the Transformer are optimized to predict the next token y_{query}. *Right*: Our results confirm the hypothesis that learning with K gradient descent steps matches trained Transformers with K linear self-attention layers.

Image Source: [16]

> *'... training Transformers ... can be closely related to well-known gradient-based meta-learning formulations.'* Transformers Learn In-Context [16]

Transformer

Sparse Transformer: $O(n\sqrt{n})$ instead of $O(n^2)$

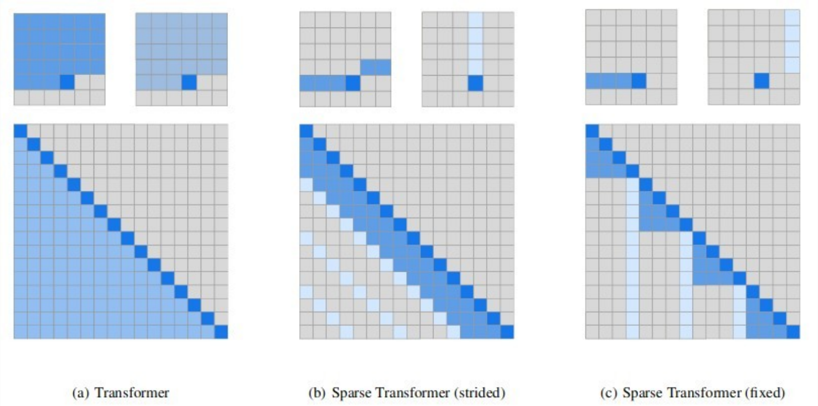


Image Source: [17]

FlashAttention

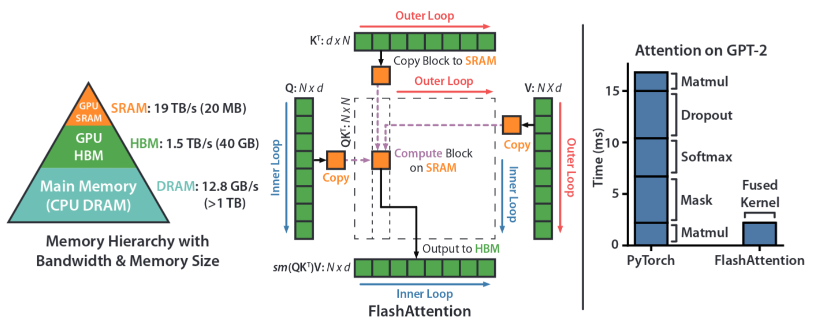


Image Source: [18]

FlashAttention Benchmarks

Attention	Standard	FlashAttention	Ratio
GFLOPs	66.6	75.2	0.89
HBM R/W	40.3	4.4	9.16
Runtime (ms)	41.7	7.3	5.71

Table from [18]

Compression

Quantization

Distillation

Rank Reduction

Quantization

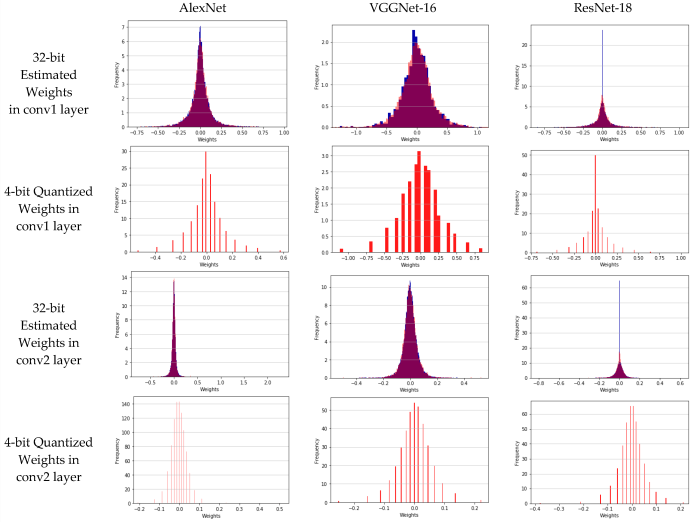


Image Source: [19] Current SOTA is GPTQ [20]

Compression

Quantization

Distillation

Rank Reduction

Knowledge Distillation

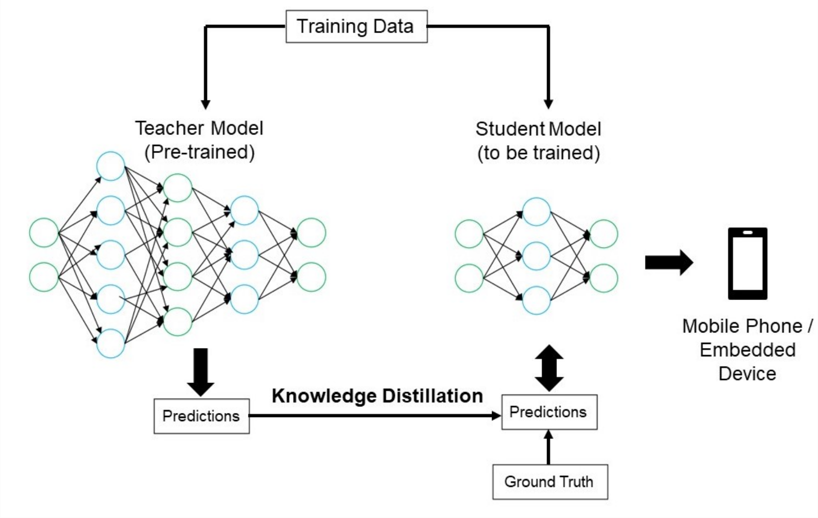
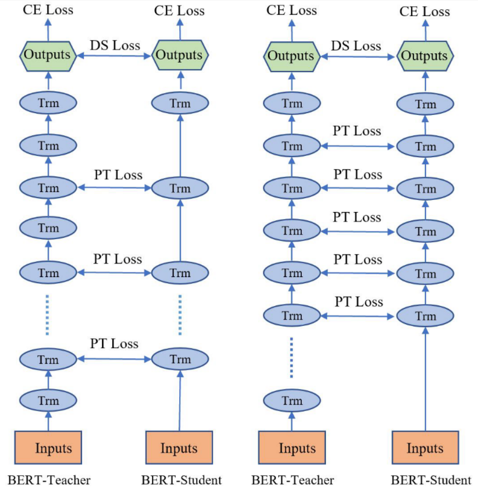


Image Source: [21]

Patient Knowledge Distillation



(a) PKD-*Skip* (b) PKD-*Last*

Distillation compression is often 2-80x with inference speedups of 1.5-10x while keeping $1 - \epsilon$ accuracy (often 97%)

Image Source: [22]

Compression

- Quantization
- Distillation
- **Rank Reduction**

LoRA: Low-Rank Adaptation

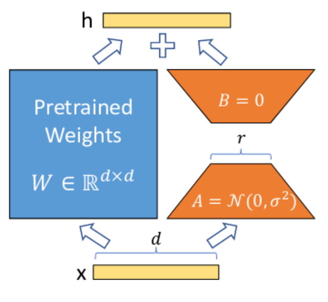


Figure 1: Our reparametrization. We only train A and B.

Image Source: [23]

'LoRA can reduce the number of trainable parameters by 10,000 times and the GPU memory requirement by 3 times ... despite having ... no additional inference latency.'

LoRA [23]

Honorable Mentions

- Transformer-XL [24]: Attentive Language Models beyond a Fixed-Length Context
- Compressive Transformer [25]: Long-Range Sequence Modelling by Compressing Past Memories
- Memorizing Transformer [26]: kNN-augmented attention layer, can reduce parameter count 5x while keeping perplexity
- Adaptive Attention Span [27]: varying attention distances
- Reflexion [28]: Asking the model if the answer is correct

Successes

- BERT
- GPT
- CLIP
- Latent Diffusion Models
- Reinforcement Learning
- Physics Simulation

BERT: Bidirectional Encoder Representations

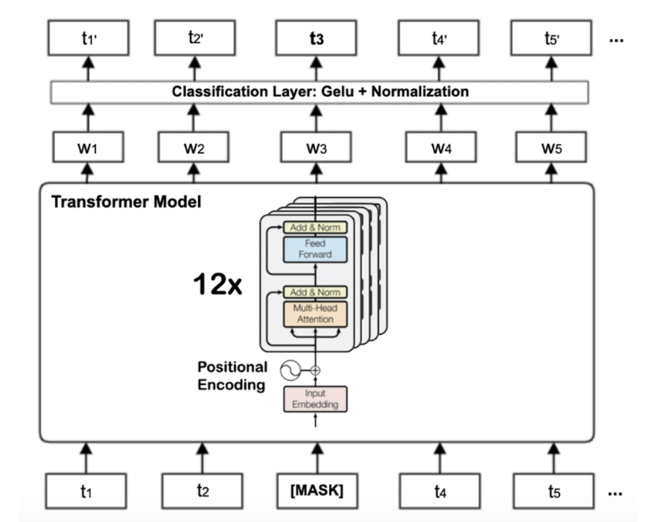


Image Source: [29] Original BERT [30]

Successes

- BERT
- GPT
- CLIP
- Latent Diffusion Models
- Reinforcement Learning
- Physics Simulation

GPT: Pure Decoder Architecture

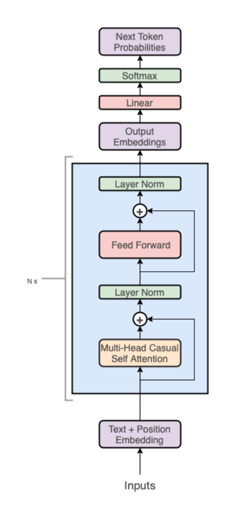


Image Source: [12]

Examples:

- GPT [31]
- GPT2 [32]
- GPT3? [33]
- GPT4? [34]
- LlaMa [35]
- Bloom [36]
- OPT [37]
- PaLM [38]

Successes

- BERT
- GPT
- CLIP
- Latent Diffusion Models
- Reinforcement Learning
- Physics Simulation

CLIP: Multimodal Embedding Spaces

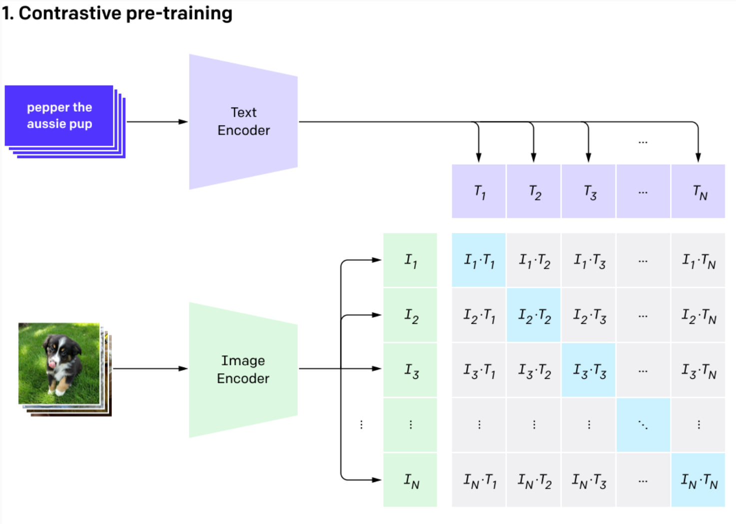


Image Source: [39]

Successes

- BERT
- GPT
- CLIP
- **Latent Diffusion Models**
- Reinforcement Learning
- Physics Simulation

Latent Diffusion Models

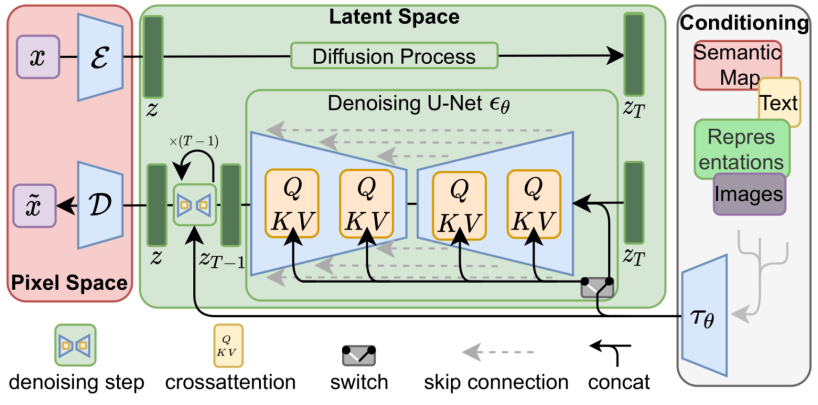


Image Source: [40]

Successes

- BERT
- GPT
- CLIP
- Latent Diffusion Models
- **Reinforcement Learning**
- Physics Simulation

GATO: A Generalist Agent

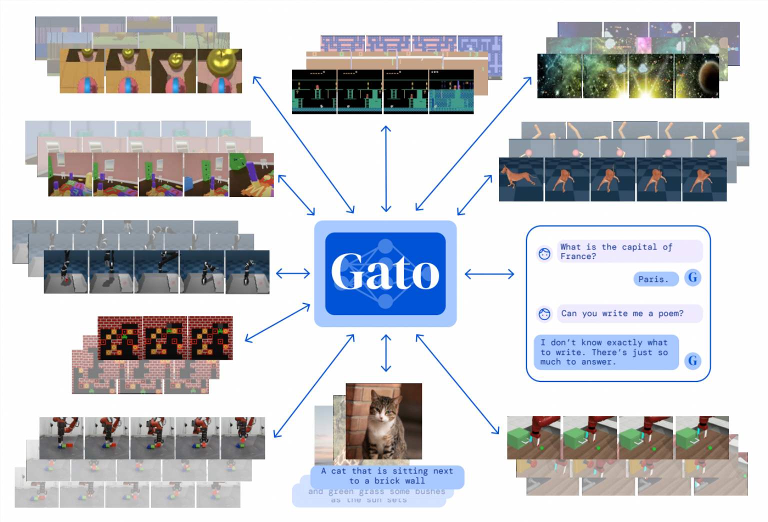


Image Source: [41]

Successes

- BERT
- GPT
- CLIP
- Latent Diffusion Models
- Reinforcement Learning
- **Physics Simulation**

Physics Simulation in Latent Spaces

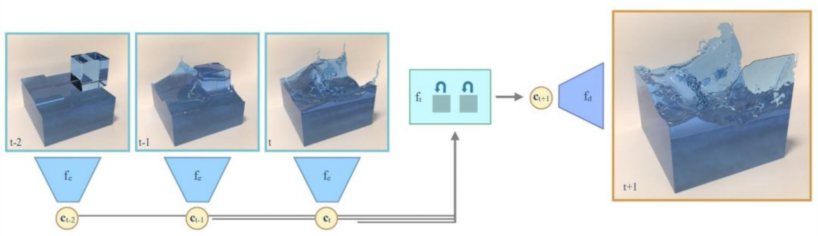


Image Source: [42]

> *'... we arrive at a data-driven solver that yields practical speed-ups, and at its core is more than 150x faster than a regular pressure solver.'* Latent Space Physics [42]

Limitations

- Hallucination: Output could be plain wrong
- Spatial Reasoning: Difficult from text alone
- Planning beyond the next step

A list of failures: [43]

Overview

Background

Embedding

Attention

Transformer

Compression

Successes

Limitations

Recap

Recap: Learning Goals

- Gain familiarity with the transformer architecture and understand how it works
- Understand Tokens and Embeddings
- Comprehend how Attention works
- Become aware of common compression techniques
- Recognize limitations

Key Takeaway: Transformers can be powerful, it might be worth trying to use them.

What are your Questions?

Sources i

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, t. L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] H. Chen, A. Didisheim, and S. Scheidegger, "Deep structural estimation: With an application to option pricing," *arXiv preprint arXiv:2102.09209*, 2021.

[3] N. Shazeer, "Glu variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

Sources ii

[4] J. Lever, M. Krzywinski, and N. Altman, "Points of significance: Model selection and overfitting," *Nature methods*, vol. 13, no. 9, pp. 703–705, 2016.

[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

Sources iii

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[7] Y. Shibata, T. Kida, S. Fukamachi, M. Takeda, A. Shinohara, T. Shinohara, and S. Arikawa, "Byte Pair encoding: A text compression scheme that accelerates pattern matching," 1999.

[8] J. Alammar, "The Illustrated GPT-2 (Visualizing Transformer Language Models)," Aug. 2019.

Sources iv

[9] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, “RoFormer: Enhanced Transformer with Rotary Position Embedding,” *arXiv:arXiv:2104.09864*, Aug. 2022.

[10] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” *arXiv:arXiv:1409.0473*, May 2016.

[11] C. A. Docs, “Top-k & Top-p,” Nov. 2022.

[12] J. Mody, “GPT in 60 Lines of NumPy,” Jan. 2023.

Sources v

[13] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T.-Y. Liu, "Understanding and Improving Transformer From a Multi-Particle Dynamic System Point of View," *arXiv:arXiv:1906.02762*, June 2019.

[14] J. Geiser, *Decomposition Methods for Differential Equations: Theory and Applications*.
CRC Press, 2009.

Sources vi

[15] X. Ye, Z. He, W. Heng, and Y. Li, "Toward understanding the effectiveness of attention mechanism," *AIP Advances*, vol. 13, p. 035019, Mar. 2023.

[16] J. von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov, "Transformers learn in-context by gradient descent," *arXiv:arXiv:2212.07677*, Dec. 2022.

[17] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating Long Sequences with Sparse Transformers," *arXiv:arXiv:1904.10509*, Apr. 2019.

Sources vii

[18] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," *arXiv preprint arXiv:2205.14135*, 2022.

[19] S. Seo and J. Kim, "Efficient Weights Quantization of Convolutional Neural Networks Using Kernel Density Estimation based Non-uniform Quantizer," *Applied Sciences*, vol. 9, p. 2559, Jan. 2019.

Sources viii

[20] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers," *arXiv:arXiv:2210.17323*, Oct. 2022.

[21] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, June 2021.

[22] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient Knowledge Distillation for BERT Model Compression," *arXiv:arXiv:1908.09355*, Aug. 2019.

Sources ix

[23] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-Rank Adaptation of Large Language Models,” *arXiv:arXiv:2106.09685*, Oct. 2021.

[24] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” *arXiv preprint arXiv:1901.02860*, 2019.

Sources x

[25] J. W. Rae, A. Potapenko, S. M. Jayakumar, and T. P. Lillicrap, "Compressive Transformers for Long-Range Sequence Modelling," *arXiv:arXiv:1911.05507*, Nov. 2019.

[26] Y. Wu, M. N. Rabe, D. Hutchins, and C. Szegedy, "Memorizing Transformers," *arXiv:arXiv:2203.08913*, Mar. 2022.

[27] S. Sukhbaatar, E. Grave, P. Bojanowski, and A. Joulin, "Adaptive Attention Span in Transformers," *arXiv:arXiv:1905.07799*, Aug. 2019.

Sources xi

[28] N. Shinn, B. Labash, and A. Gopinath, "Reflexion: An autonomous agent with dynamic memory and self-reflection," *arXiv:arXiv:2303.11366*, Mar. 2023.

[29] U. Khalid, M. Beg, and M. Arshad, *RUBERT: A Bilingual Roman Urdu BERT Using Cross Lingual Transfer Learning*. **Feb. 2021.**

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

Sources xii

[31] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving Language Understanding by Generative Pre-Training,” 2018.

[32] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Unsupervised Multitask Learners,” *published on GitHub*, 2019.

[33] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, and A. Askell, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

Sources xiii

[34] OpenAI, "GPT-4 Technical Report," 2023.

[35] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and Efficient Foundation Language Models," *arXiv:arXiv:2302.13971*, Feb. 2023.

[36] B. Workshop, T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, J. Tow, A. M. Rush, S. Biderman, A. Webson, P. S. Ammanamanchi, T. Wang, B. Sagot, N. Muennighoff, A. V. del Moral,

O. Ruwase, R. Bawden, S. Bekman, A. McMillan-Major, I. Beltagy, H. Nguyen, L. Saulnier, S. Tan, P. O. Suarez, V. Sanh, H. Laurençon, Y. Jernite, J. Launay, M. Mitchell, C. Raffel, A. Gokaslan, A. Simhi, A. Soroa, A. F. Aji, A. Alfassy, A. Rogers, A. K. Nitzav, C. Xu, C. Mou, C. Emezue, C. Klamm, C. Leong, D. van Strien, D. I. Adelani, D. Radev, E. G. Ponferrada, E. Levkovizh, E. Kim, E. B. Natan, F. De Toni, G. Dupont, G. Kruszewski, G. Pistilli, H. Elsahar, H. Benyamina, H. Tran, I. Yu, I. Abdulmumin, I. Johnson, I. Gonzalez-Dios, J. de la Rosa, J. Chim, J. Dodge, J. Zhu, J. Chang,

J. Frohberg, J. Tobing, J. Bhattacharjee, K. Almubarak, K. Chen, K. Lo, L. Von Werra, L. Weber, L. Phan, L. B. allal, L. Tanguy, M. Dey, M. R. Muñoz, M. Masoud, M. Grandury, M. Šaško, M. Huang, M. Coavoux, M. Singh, M. T.-J. Jiang, M. C. Vu, M. A. Jauhar, M. Ghaleb, N. Subramani, N. Kassner, N. Khamis, O. Nguyen, O. Espejel, O. de Gibert, P. Villegas, P. Henderson, P. Colombo, P. Amuok, Q. Lhoest, R. Harliman, R. Bommasani, R. L. López, R. Ribeiro, S. Osei, S. Pyysalo, S. Nagel, S. Bose, S. H. Muhammad, S. Sharma, S. Longpre, S. Nikpoor, S. Silberberg, S. Pai, S. Zink, T. T.

Torrent, T. Schick, T. Thrush, V. Danchev,
V. Nikoulina, V. Laippala, V. Lepercq, V. Prabhu,
Z. Alyafeai, Z. Talat, A. Raja, B. Heinzerling, C. Si,
D. E. Taşar, E. Salesky, S. J. Mielke, W. Y. Lee,
A. Sharma, A. Santilli, A. Chaffin, A. Stiegler,
D. Datta, E. Szczechla, G. Chhablani, H. Wang,
H. Pandey, H. Strobelt, J. A. Fries, J. Rozen, L. Gao,
L. Sutawika, M. S. Bari, M. S. Al-shaibani,
M. Manica, N. Nayak, R. Teehan, S. Albanie, S. Shen,
S. Ben-David, S. H. Bach, T. Kim, T. Bers, T. Fevry,
T. Neeraj, U. Thakker, V. Raunak, X. Tang, Z.-X.
Yong, Z. Sun, S. Brody, Y. Uri, H. Tojarieh,

A. Roberts, H. W. Chung, J. Tae, J. Phang, O. Press, C. Li, D. Narayanan, H. Bourfoune, J. Casper, J. Rasley, M. Ryabinin, M. Mishra, M. Zhang, M. Shoeybi, M. Peyrounette, N. Patry, N. Tazi, O. Sanseviero, P. von Platen, P. Cornette, P. F. Lavallée, R. Lacroix, S. Rajbhandari, S. Gandhi, S. Smith, S. Requena, S. Patil, T. Dettmers, A. Baruwa, A. Singh, A. Cheveleva, A.-L. Ligozat, A. Subramonian, A. Névéol, C. Lovering, D. Garrette, D. Tunuguntla, E. Reiter, E. Taktasheva, E. Voloshina, E. Bogdanov, G. I. Winata, H. Schoelkopf, J.-C. Kalo, J. Novikova, J. Z. Forde, J. Clive, J. Kasai,

Sources xviii

K. Kawamura, L. Hazan, M. Carpuat, M. Clinciu, N. Kim, N. Cheng, O. Serikov, O. Antverg, O. van der Wal, R. Zhang, R. Zhang, S. Gehrmann, S. Mirkin, S. Pais, T. Shavrina, T. Scialom, T. Yun, T. Limisiewicz, V. Rieser, V. Protasov, V. Mikhailov, Y. Pruksachatkun, Y. Belinkov, Z. Bamberger, Z. Kasner, A. Rueda, A. Pestana, A. Feizpour, A. Khan, A. Faranak, A. Santos, A. Hevia, A. Unldreaj, A. Aghagol, A. Abdollahi, A. Tammour, A. HajiHosseini, B. Behroozi, B. Ajibade, B. Saxena, C. M. Ferrandis, D. Contractor, D. Lansky, D. David, D. Kiela, D. A. Nguyen, E. Tan, E. Baylor, E. Ozoani,

F. Mirza, F. Ononiwu, H. Rezanejad, H. Jones,
I. Bhattacharya, I. Solaiman, I. Sedenko, I. Nejadgholi,
J. Passmore, J. Seltzer, J. B. Sanz, L. Dutra,
M. Samagaio, M. Elbadri, M. Mieskes, M. Gerchick,
M. Akinlolu, M. McKenna, M. Qiu, M. Ghauri,
M. Burynok, N. Abrar, N. Rajani, N. Elkott, N. Fahmy,
O. Samuel, R. An, R. Kromann, R. Hao, S. Alizadeh,
S. Shubber, S. Wang, S. Roy, S. Viguier, T. Le,
T. Oyebade, T. Le, Y. Yang, Z. Nguyen, A. R.
Kashyap, A. Palasciano, A. Callahan, A. Shukla,
A. Miranda-Escalada, A. Singh, B. Beilharz, B. Wang,
C. Brito, C. Zhou, C. Jain, C. Xu, C. Fourrier, D. L.

Sources xx

Periñán, D. Molano, D. Yu, E. Manjavacas, F. Barth, F. Fuhrimann, G. Altay, G. Bayrak, G. Burns, H. U. Vrabec, I. Bello, I. Dash, J. Kang, J. Giorgi, J. Golde, J. D. Posada, K. R. Sivaraman, L. Bulchandani, L. Liu, L. Shinzato, M. H. de Bykhovetz, M. Takeuchi, M. Pàmies, M. A. Castillo, M. Nezhurina, M. Sänger, M. Samwald, M. Cullan, M. Weinberg, M. De Wolf, M. Mihaljcic, M. Liu, M. Freidank, M. Kang, N. Seelam, N. Dahlberg, N. M. Broad, N. Muellner, P. Fung, P. Haller, R. Chandrasekhar, R. Eisenberg, R. Martin, R. Canalli, R. Su, R. Su, S. Cahyawijaya, S. Garda, S. S. Deshmukh, S. Mishra, S. Kiblawi,

Sources xxi

S. Ott, S. Sang-aroonsiri, S. Kumar, S. Schweter, S. Bharati, T. Laud, T. Gigant, T. Kainuma, W. Kusa, Y. Labrak, Y. S. Bajaj, Y. Venkatraman, Y. Xu, Y. Xu, Y. Xu, Z. Tan, Z. Xie, Z. Ye, M. Bras, Y. Belkada, and T. Wolf, "BLOOM: A 176B-Parameter Open-Access Multilingual Language Model," *arXiv:arXiv:2211.05100*, Dec. 2022.

[37] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "OPT: Open Pre-trained Transformer Language Models," *arXiv:arXiv:2205.01068*, June 2022.

[38] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, and S. Gehrmann, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[39] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning Transferable Visual Models From Natural Language Supervision,” *arXiv:arXiv:2103.00020*, Feb. 2021.

[40] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” *arXiv:arXiv:2112.10752*, Apr. 2022.

[41] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas, "A Generalist Agent," *arXiv:arXiv:2205.06175*, Nov. 2022.

[42] S. Wiewel, M. Becher, and N. Thuerey, "Latent Space Physics: Towards Learning the Temporal Evolution of Fluid Flow," *Computer Graphics Forum*, vol. 38, no. 2, pp. 71–82, 2019.

Sources xxv

[43] A. Borji, "A Categorical Archive of ChatGPT Failures," *arXiv:arXiv:2302.03494*, Feb. 2023.

End

Thesis content

- Using a large language model
- (planned: OPT [37], but currently using LlaMa [35])
- Fine-tuning it for data extraction tasks on a specialized domain
- (probably trying a LoRA-version for that [23])
- If time permits, see how much I can reduce model size by Compression