# Attention is All You Need

overview of the transformer architecture, applications and established improvements

Felix Karg

March 30, 2023

Institute for Theoretical Informatics:
Artificial Intelligence for Materials Science

Compression

Transformer

Compression

**Transformer**

# Transformer

**Output**

Dimensions

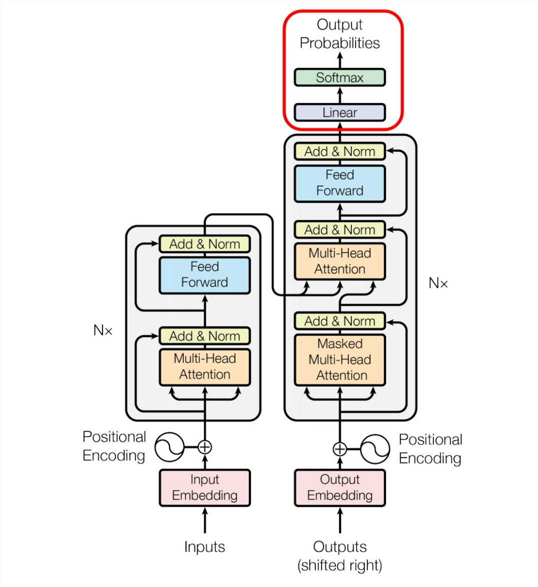
Putting it all Together

Interpretation

Sparse Transformer

Linear Time Transformers

# 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)

# Transformer

Output

**Dimensions**

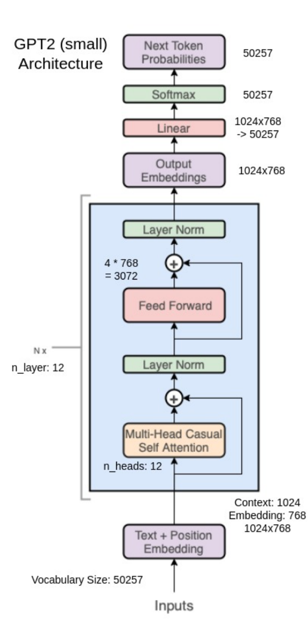
Putting it all Together

Interpretation

Sparse Transformer

Linear Time Transformers

# Dimensions


GPT2 (small)
Architecture
Next Token Probabilities
50257
Softmax
50257
Linear
1024x768
-> 50257
Output Embeddings
1024x768
Layer Norm
4 * 768
= 3072
Feed Forward
N x
n_layer: 12
Layer Norm
Multi-Head Casual Self Attention
n_heads: 12
Context: 1024
Embedding: 768
1024x768
Text + Position Embedding
Vocabulary Size: 50257
Inputs

# Dimensions II

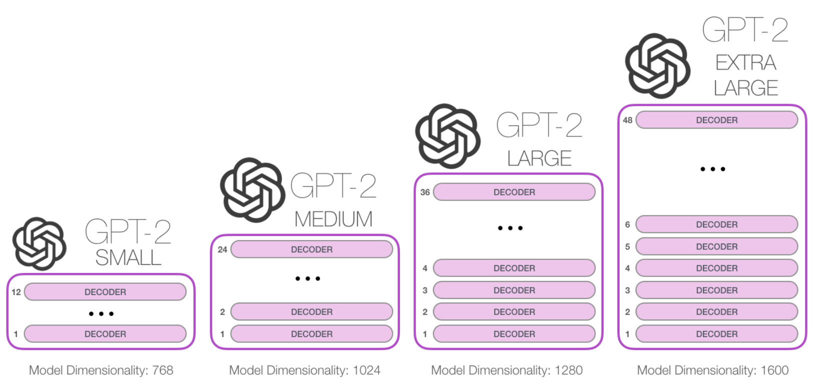


Image Source: [1]

# Transformer

- Output
- Dimensions
- **Putting it all Together**
- Interpretation
- Sparse Transformer
- Linear Time Transformers

# Full Architecture Overview

# Transformer

Output

Dimensions

Putting it all Together

**Interpretation**

Sparse Transformer

Linear Time Transformers

# Solving PDE [2]

# Transformer

Output

Dimensions

Putting it all Together

Interpretation

**Sparse Transformer**

Linear Time Transformers

Even the original GPT didn't use 'full' transformers, but Sparse Transformer [3]

# Transformer

Output

Dimensions

Putting it all Together

Interpretation

Sparse Transformer

**Linear Time Transformers**

FastFormer [4]
Transformer Quality in Linear Time [5]
FlashAttention [6]

Transformer

# Compression

# Compression

Quantization

Distillation

Rank Reduction

GPTQ [7] (spun up for LlaMa just two weeks after 'release')

# Compression

Quantization

Distillation

Rank Reduction

Training a smaller model on outputting similar outputs distributions for given inputs [8] [9]

# Compression

Quantization

Distillation

Rank Reduction

LoRA [10] (spun up for LlaMa just a month after 'release')

# Sources i

[1] J. Alammar, “The Illustrated GPT-2 (Visualizing Transformer Language Models),” Aug. 2019.

[2] 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.

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

# Sources ii

[4] C. Wu, F. Wu, T. Qi, Y. Huang, and X. Xie, "Fastformer: Additive Attention Can Be All You Need," *arXiv:arXiv:2108.09084*, Sept. 2021.

[5] W. Hua, Z. Dai, H. Liu, and Q. Le, "Transformer quality in linear time," in *International Conference on Machine Learning*, pp. 9099–9117, PMLR, 2022.

[6] 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.

# Sources iii

[7] 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.

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

[9] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," *arXiv:arXiv:1802.05668*, Feb. 2018.

# Sources iv

[10] 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.