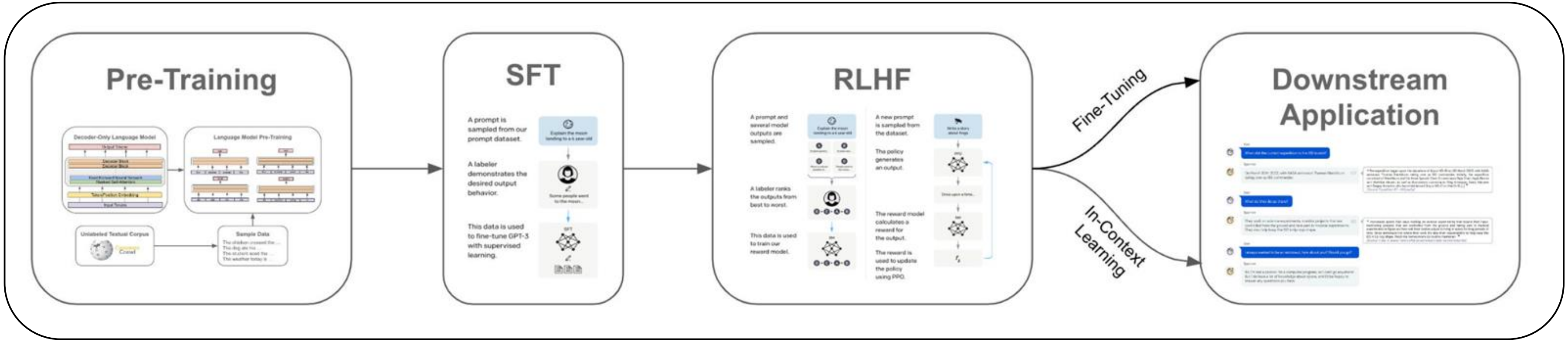
RIT

Large Language Models

Reza Fayyazi

Introduction

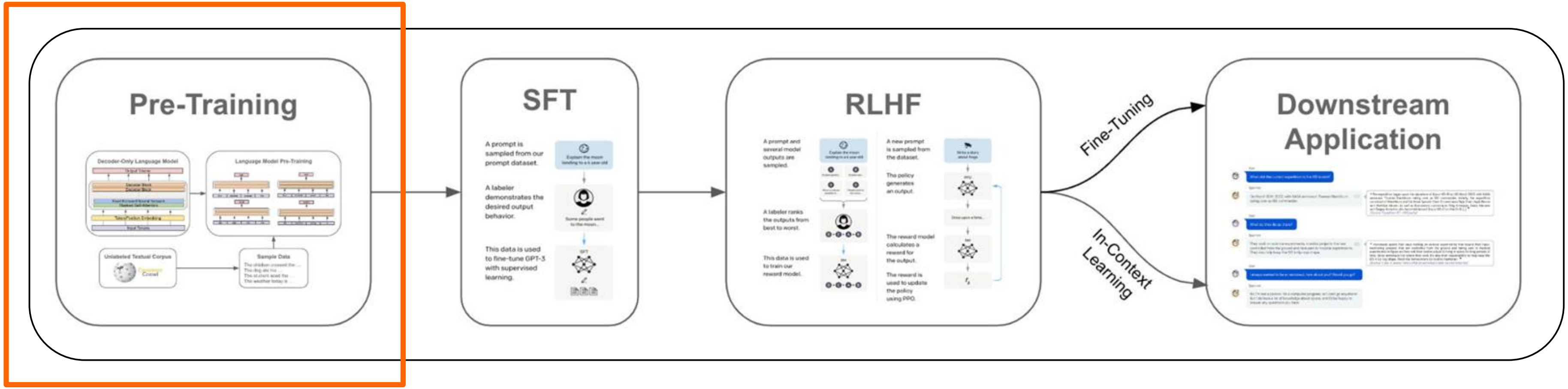
The overall process of creating your own LLM



[1] https://cameronrwolfe.substack.com/p/understanding-and-using-supervised

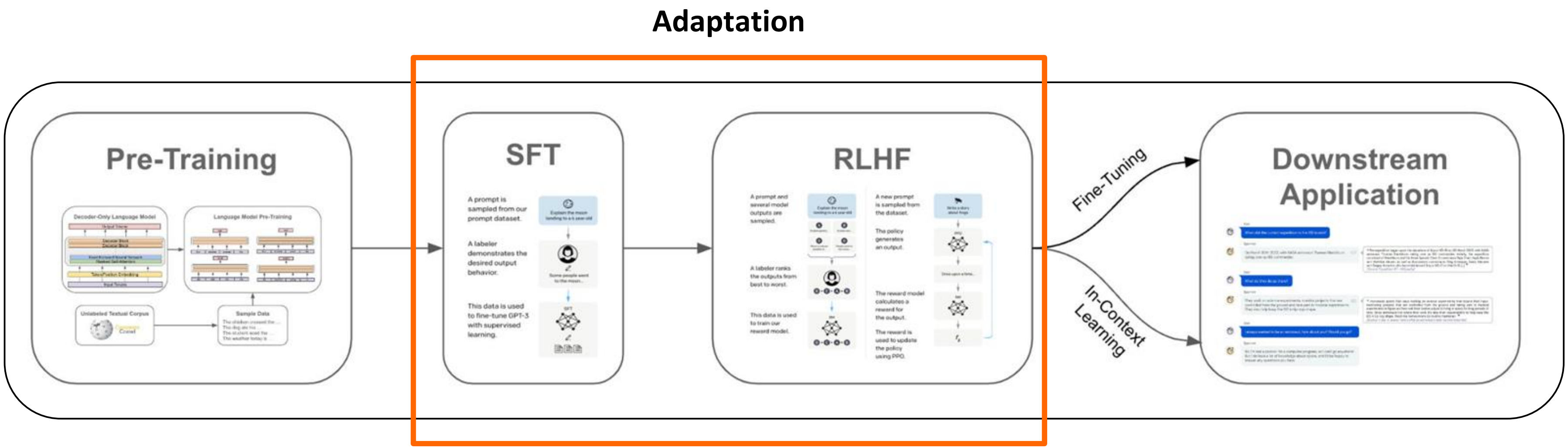
The overall process of creating your own LLM

Foundation Model



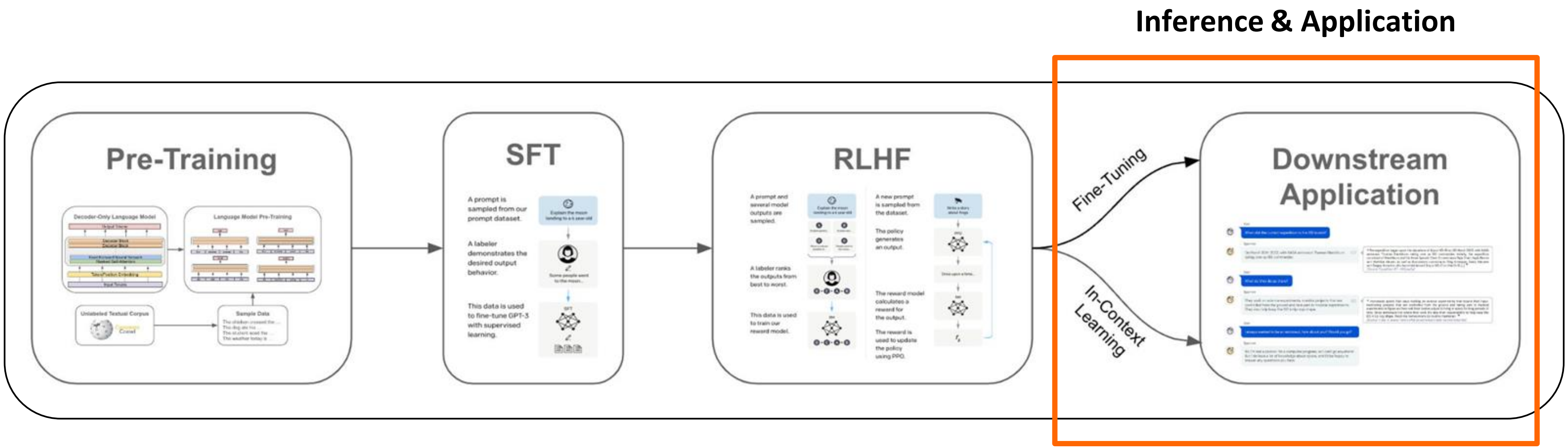
[1] https://cameronrwolfe.substack.com/p/understanding-and-using-supervised

The overall process of creating your own LLM



[1] https://cameronrwolfe.substack.com/p/understanding-and-using-supervised

The overall process of creating your own LLM



[1] https://cameronrwolfe.substack.com/p/understanding-and-using-supervised

Transformers

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

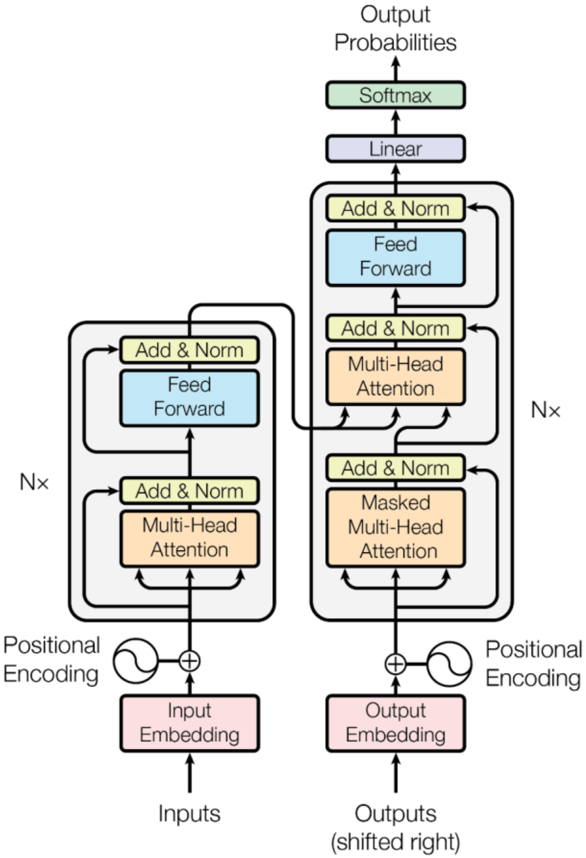
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

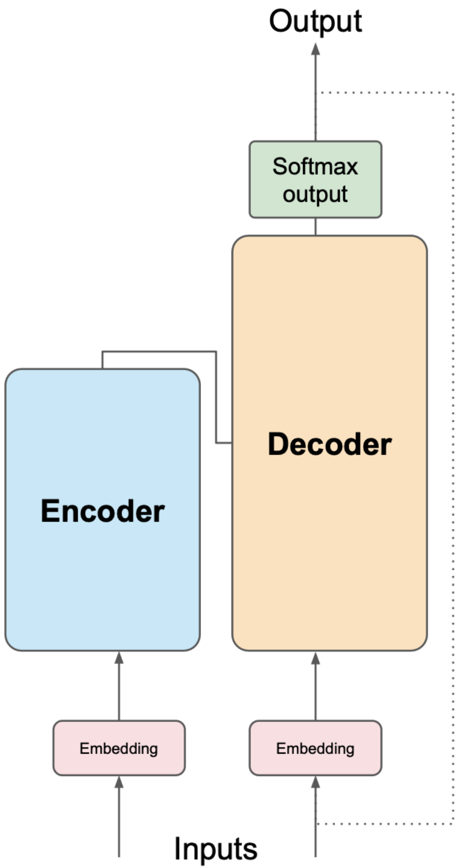
Abstract

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 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.



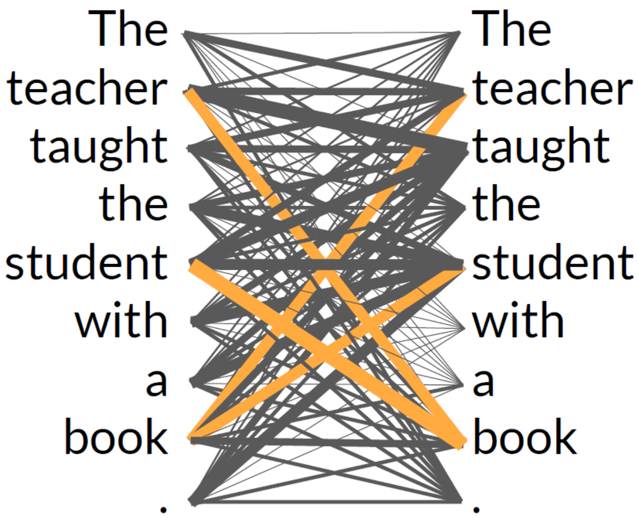
[2] Vaswani, A. (2017). Attention is all you need. Advances in Neural Information Processing Systems.

Transformers

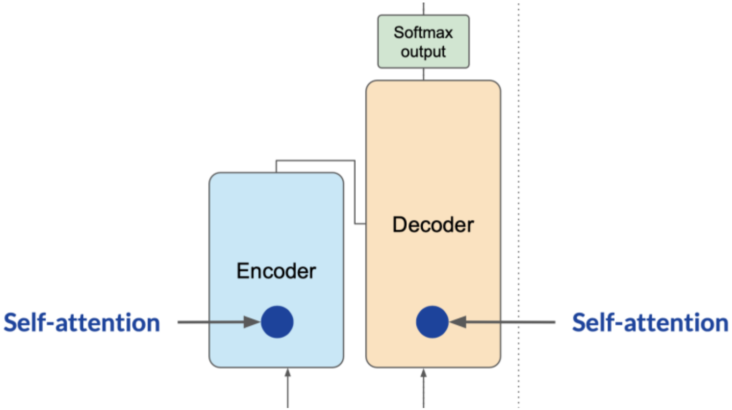


[3] https://www.coursera.org/learn/generative-ai-with-llms/

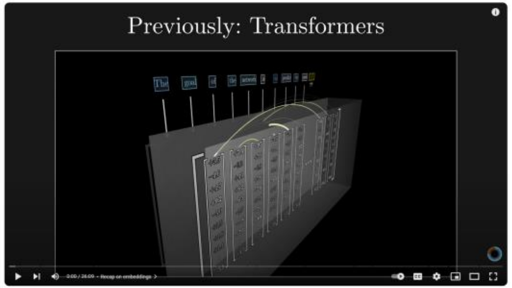
Self-Attention



[3]



Watch this YouTube video on how the self-attention mechanism works: https://www.youtube.com/watch?app=desktop&v=eMlx5fFNoYc

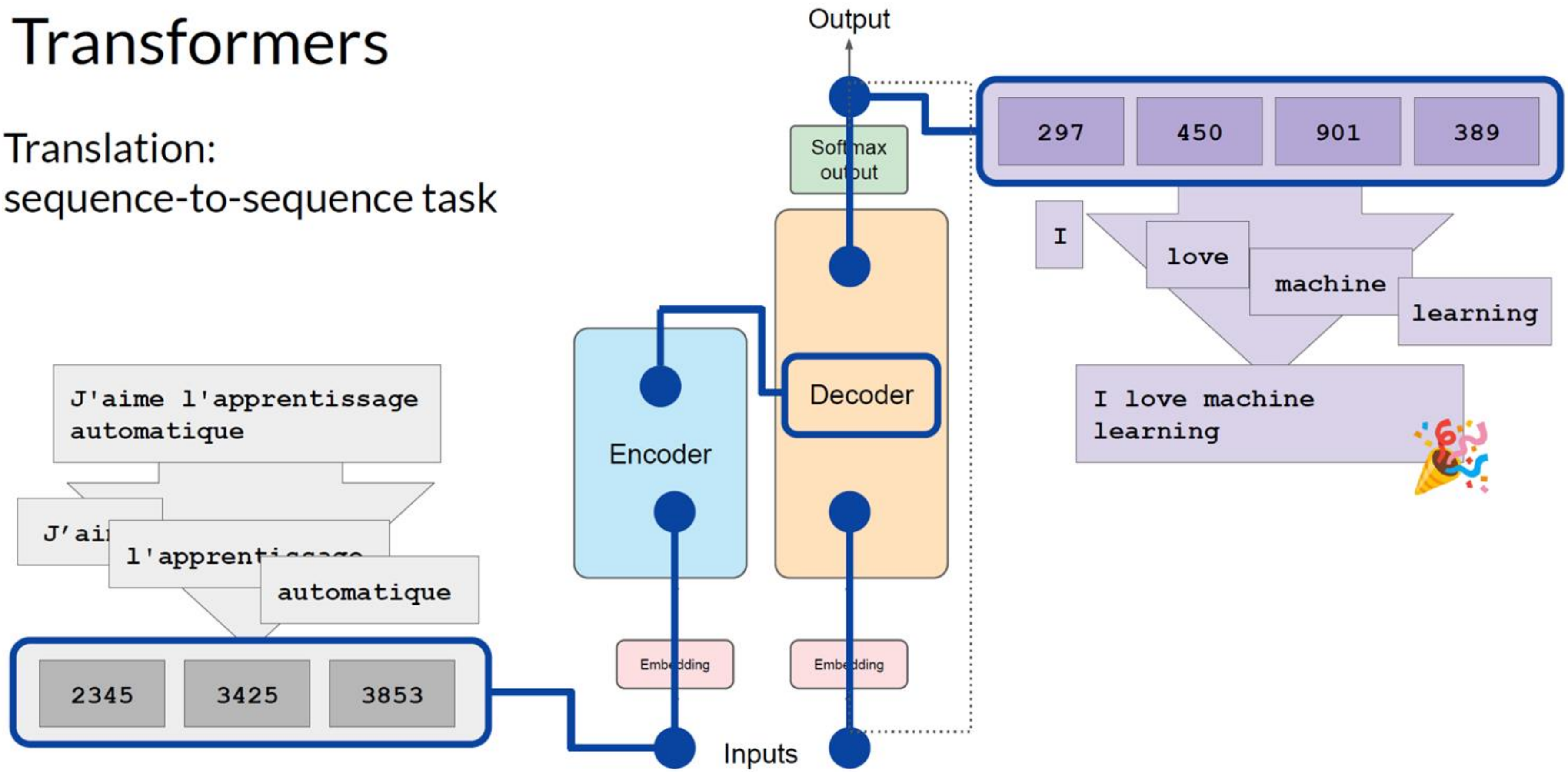


Attention in transformers, visually explained | Chapter 6, Deep Learning

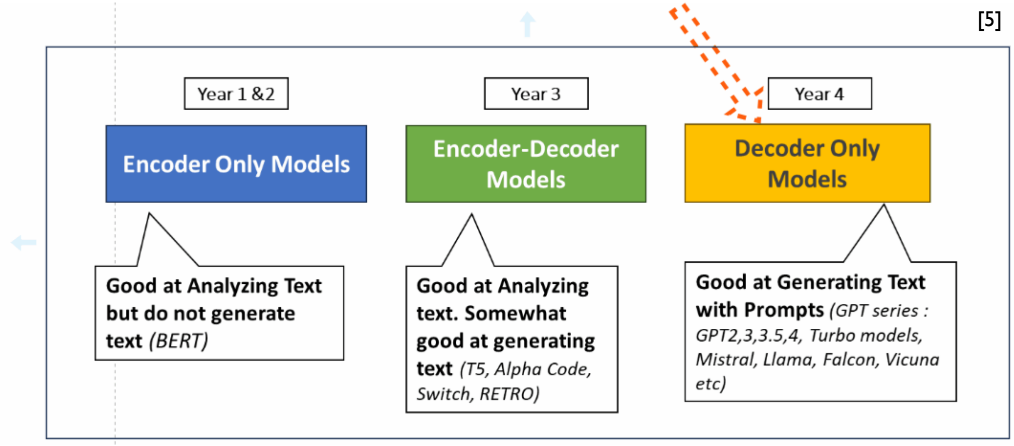
[3] https://www.coursera.org/learn/generative-ai-with-llms/

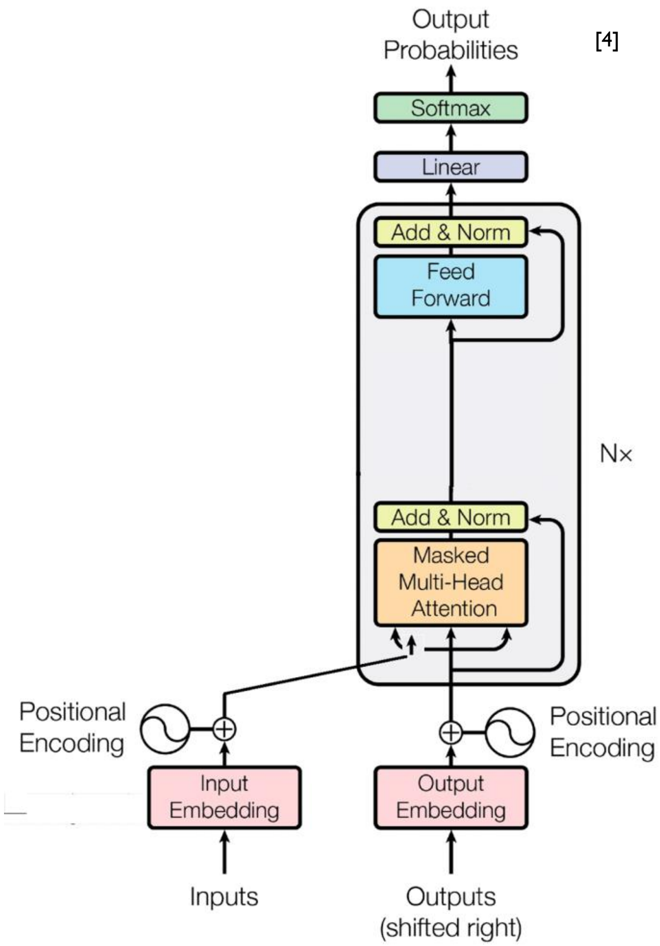
Transformers

Translation:
sequence-to-sequence task



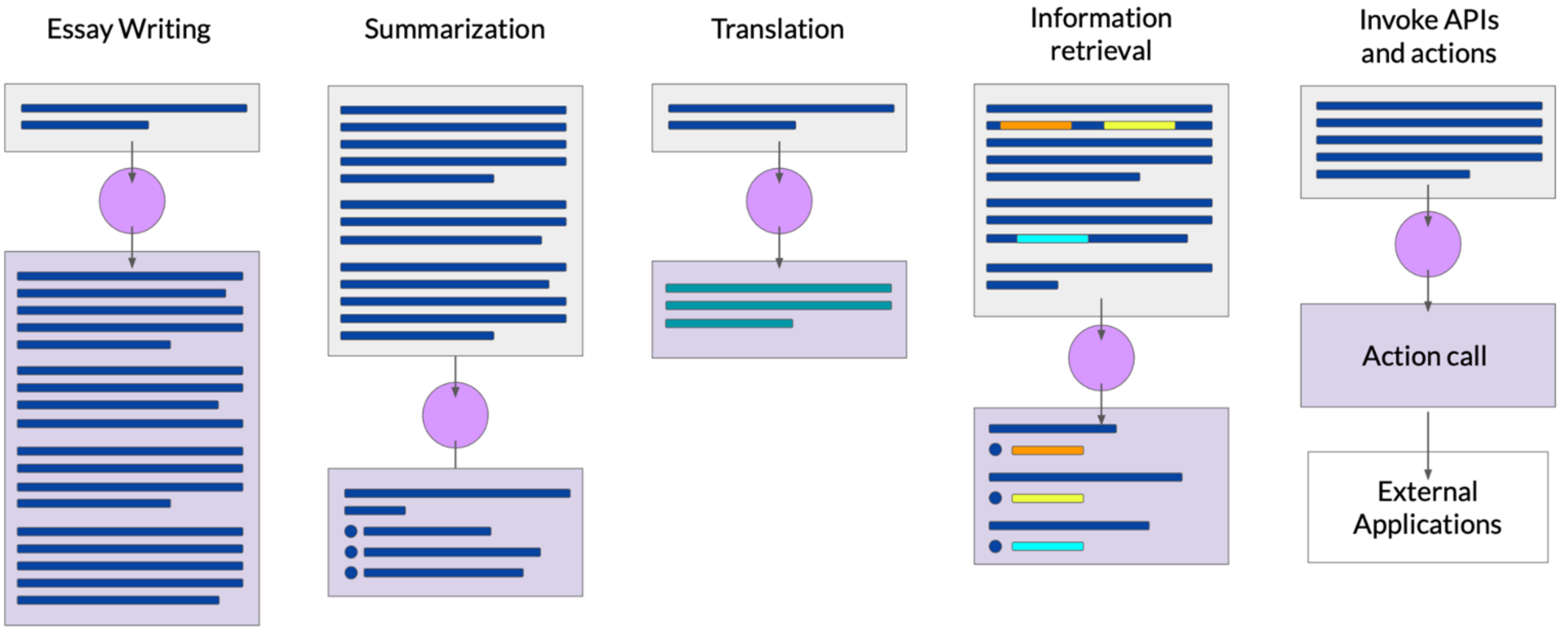
State-of-the-art LLMs are Decoder-Only Models





[4] https://stackoverflow.com/questions/75672816/how-does-gpt-like-transformers-utilize-only-the-decoder-to-do-sequence-generatio
[5] https://www.linkedin.com/pulse/transformer-architectures-dummies-part-2-decoder-only-qi6vc

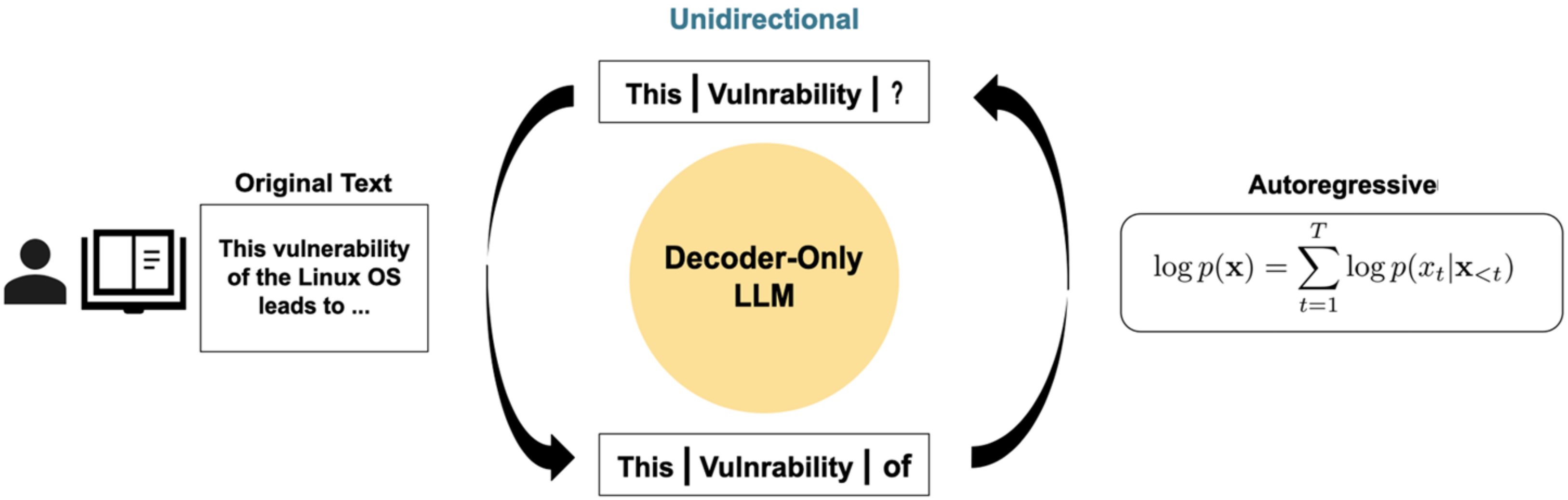
LLM Use Cases



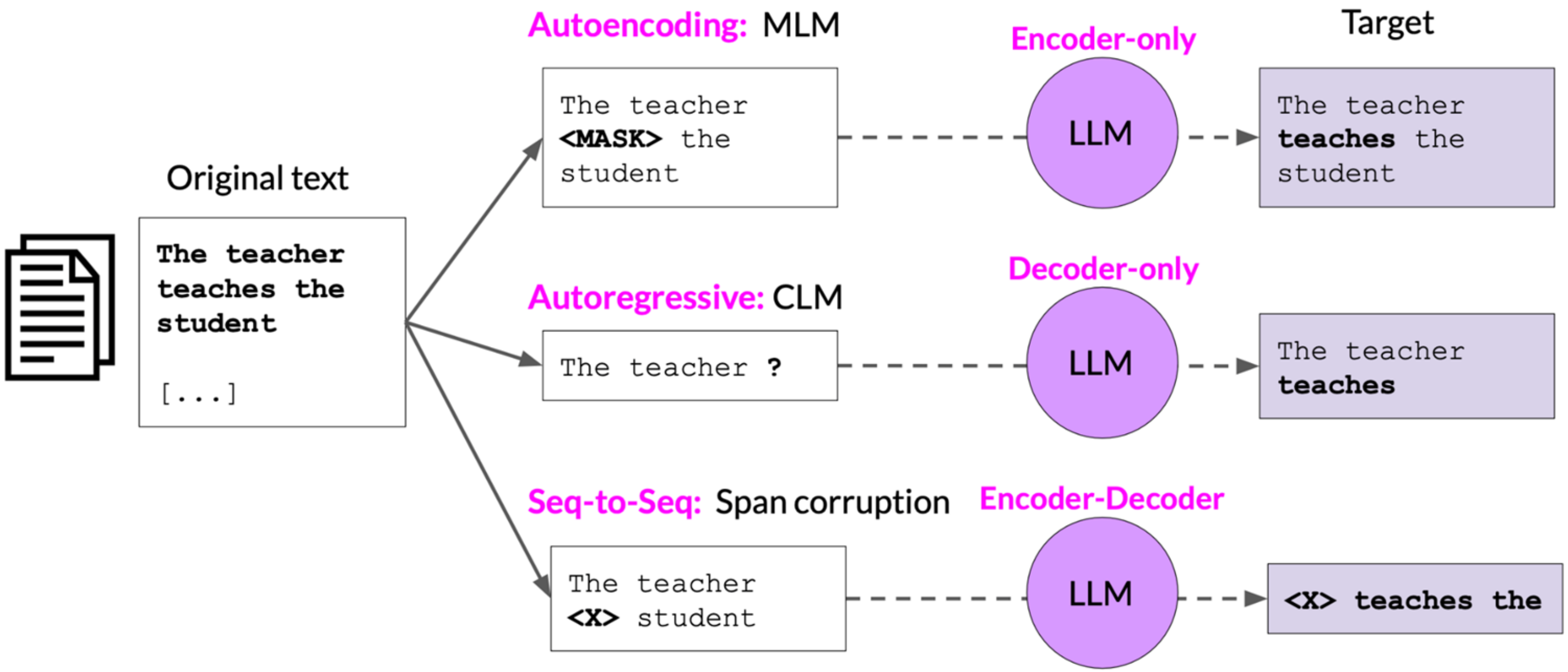
[3] https://www.coursera.org/learn/generative-ai-with-llms/

LLM Pre-Training

LLMs work on predicting the next probable token!

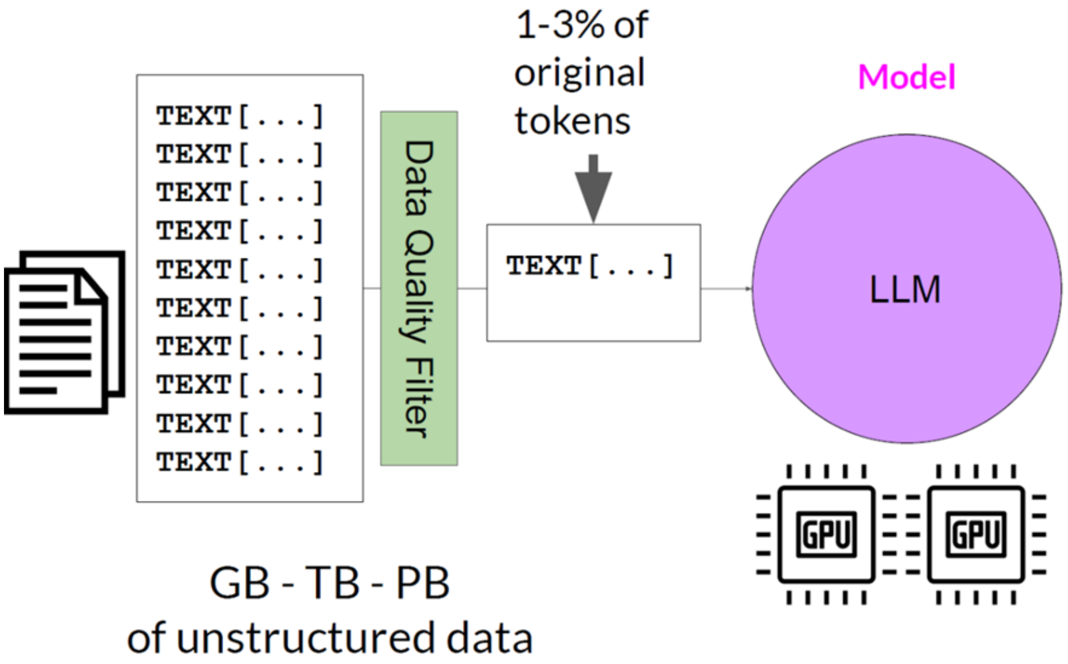


Pre-training Objectives



[3] https://www.coursera.org/learn/generative-ai-with-llms/

LLM Pre-training

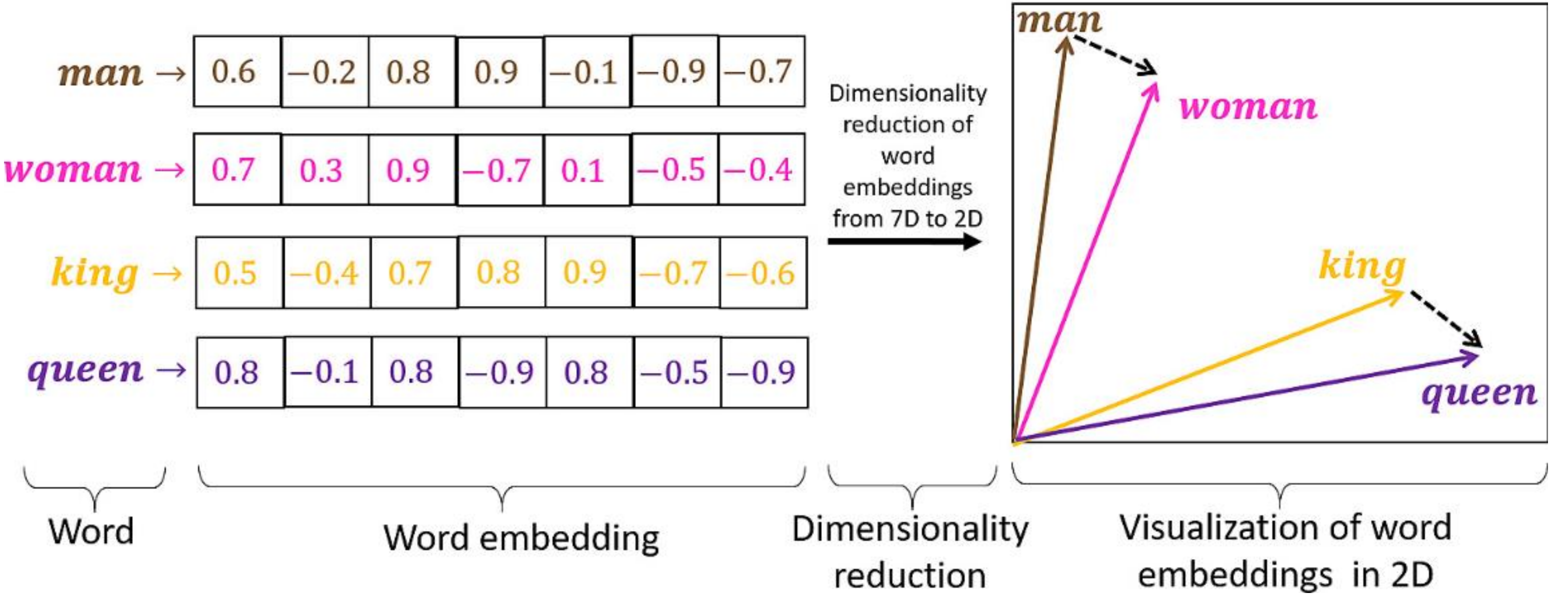


Token String	Token ID	Embedding / Vector Representation
'_The'	37	[-0.0513, -0.0584, 0.0230, ...]
'_teacher'	3145	[-0.0335, 0.0167, 0.0484, ...]
'_teaches'	11749	[-0.0151, -0.0516, 0.0309, ...]
'_the'	8	[-0.0498, -0.0428, 0.0275, ...]
'_student'	1236	[-0.0460, 0.0031, 0.0545, ...]
...	...	...

Vocabulary

[3] https://www.coursera.org/learn/generative-ai-with-llms/

Word Embeddings



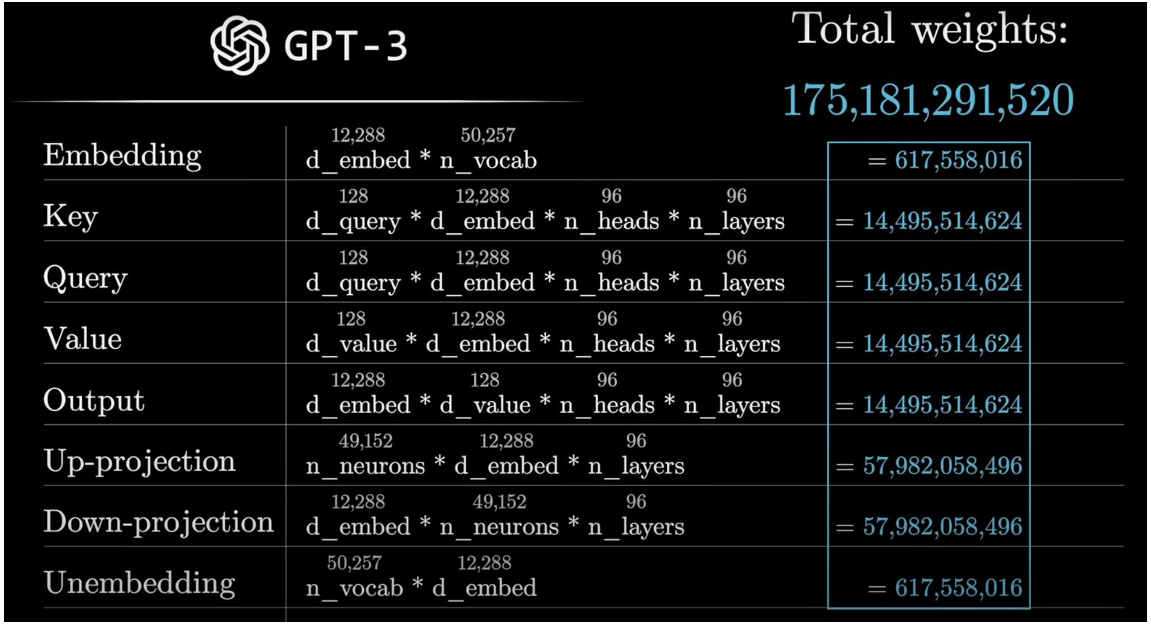
[7] https://airbyte.com/data-engineering-resources/sentence-word-embeddings

LLM Comparison

Model	Provider	Context Window	Open-Source	Price / Million	Price	Quality	Speed
GPT-4o	OpenAI	128k	No	7.5	★☆☆	★★★	★★☆
GPT-4 Turbo	OpenAI	128k	No	15	★☆☆	★★★	★☆☆
GPT-4	OpenAI	8k	No	37.5	★☆☆	★★★	★★☆
GPT-3.5 Turbo	OpenAI	16k	No	0.75	★★☆	★☆☆	★★☆
Gemini 1.5 Pro	Google	1m	No	5.25	★★☆	★★★	★★☆
Gemini 1.5 Flash	Google	1m	No	0.53	★★★	★★☆	★★★
Gemma 7B	Google	8k	Yes	0.2	★★★	★☆☆	★★★
Claude 3 Opus	Anthropic	200k	No	30	★☆☆	★★★	★☆☆
Claude 3 Sonnet	Anthropic	200k	No	6	★☆☆	★★☆	★★☆
Claude 3 Haiku	Anthropic	200k	No	0.5	★★★	★★☆	★★★
Command R +	Cohere	128k	Yes	6	★☆☆	★★☆	★★☆
Command R	Cohere	128k	Yes	0.75	★★☆	★☆☆	★★☆
Llama 3 70B	Meta AI	8k	Yes	0.93	★★☆	★★★	★☆☆
Llama 3 8B	Meta AI	8k	Yes	0.2	★★★	★☆☆	★★★
Code Llama	Meta AI	16k	Yes	0.9	★★☆	★★★	★☆☆
Mistral Large	Mistral AI	32k	No	12	★☆☆	★★★	★☆☆
Mistral Medium	Mistral AI	32k	No	4.05	★★☆	★★☆	★☆☆
Mistral Small	Mistral Ai	32k	No	2.25	★★☆	★☆☆	★★☆
Mixtral 8x22B	Mistral AI	65k	Yes	1.2	★★☆	★★☆	★★☆
Mixtral 8x7B	Mistral AI	32k	Yes	0.5	★★★	★☆☆	★★★
Mistral 7B	Mistral AI	32k	Yes	0.2	★★★	★☆☆	★★☆
DBRX	Databricks	32k	Yes	1.4	★★☆	★★☆	★★★

[8] https://mindsdb.com/blog/navigating-the-llm-landscape-a-comparative-analysis-of-leading-large-language-models

High Number of Parameters

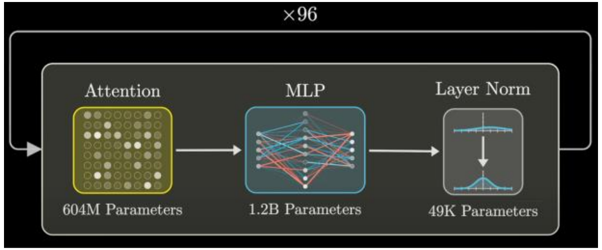


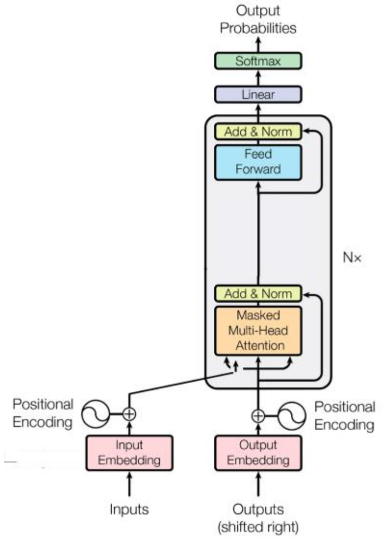
GPT-3

Total weights:

175,181,291,520

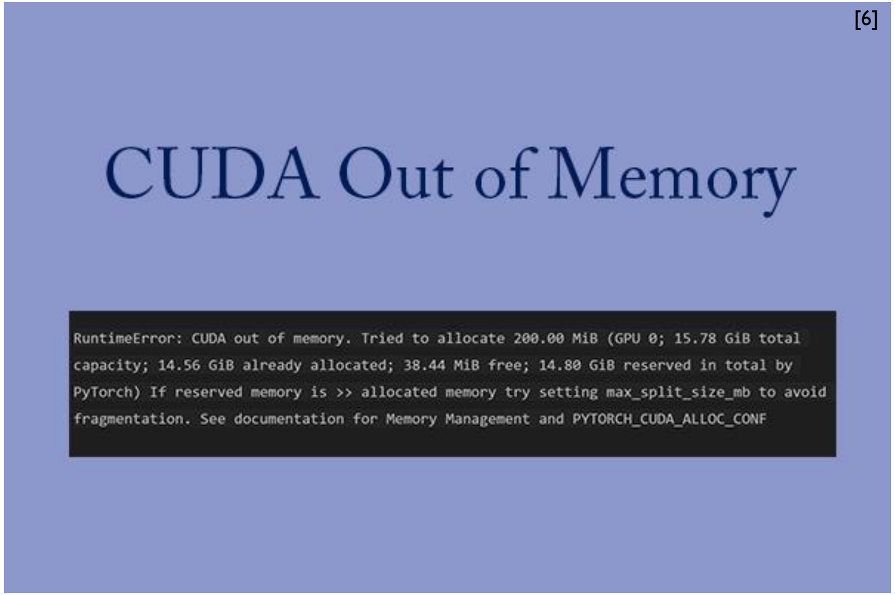
Embedding	12,288 50,257 d_embed * n_vocab	= 617,558,016
Key	128 12,288 96 96 d_query * d_embed * n_heads * n_layers	= 14,495,514,624
Query	128 12,288 96 96 d_query * d_embed * n_heads * n_layers	= 14,495,514,624
Value	128 12,288 96 96 d_value * d_embed * n_heads * n_layers	= 14,495,514,624
Output	12,288 128 96 96 d_embed * d_value * n_heads * n_layers	= 14,495,514,624
Up-projection	49,152 12,288 96 n_neurons * d_embed * n_layers	= 57,982,058,496
Down-projection	12,288 49,152 96 d_embed * n_neurons * n_layers	= 57,982,058,496
Unembedding	50,257 12,288 n_vocab * d_embed	= 617,558,016





[9] https://www.youtube.com/watch?v=9-Jl0dxWQs8

Computational Challenges



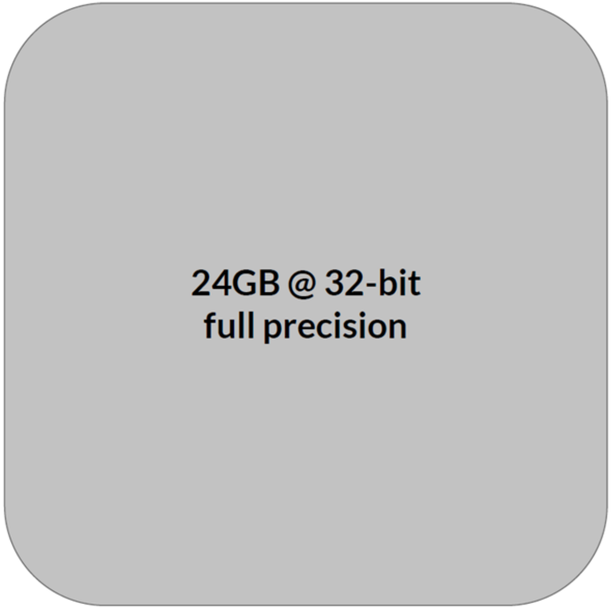
[10] https://www.partitionwizard.com/news/cuda-out-of-memory.html

Computational Challenges for 1B Param Model

Memory needed to store model

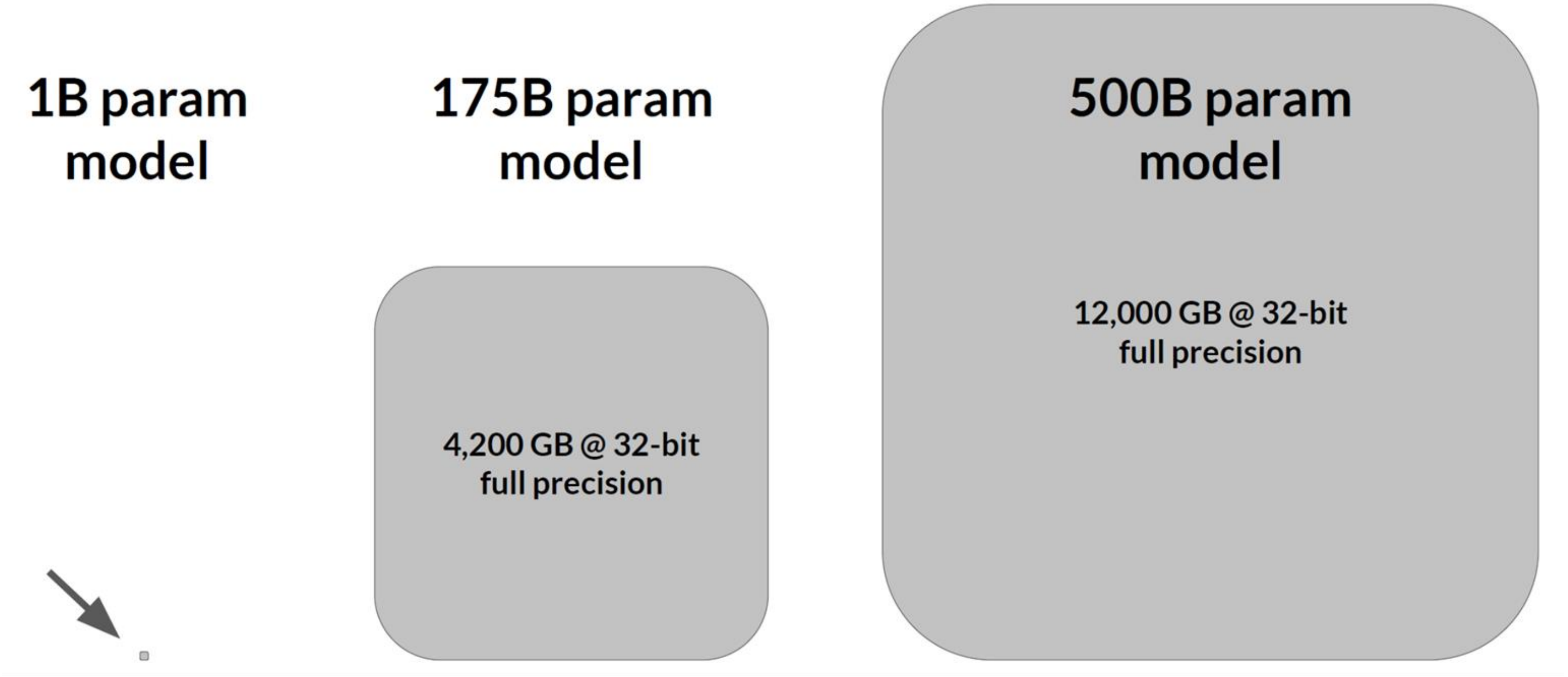
4GB @ 32-bit
full precision

Memory needed to train model



[3] https://www.coursera.org/learn/generative-ai-with-llms/

Computational Challenges



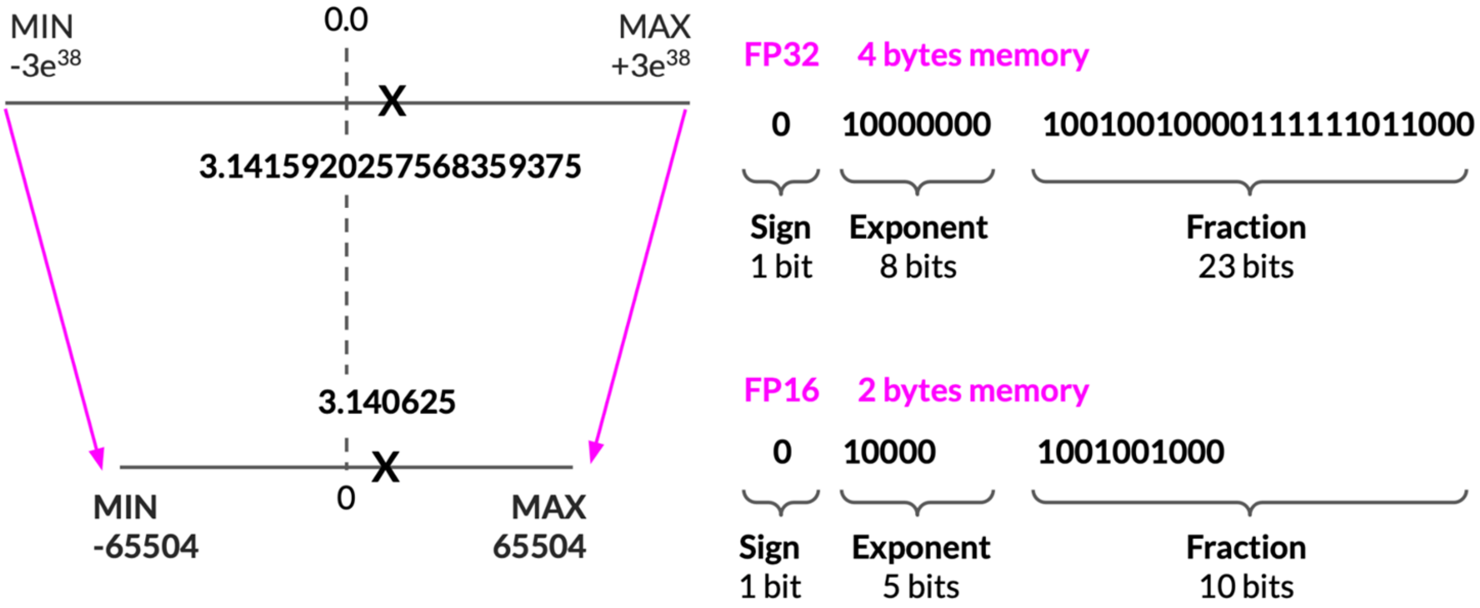
[3] https://www.coursera.org/learn/generative-ai-with-llms/

Computational Challenges

	Bytes per parameter
Model Parameters (Weights)	4 bytes per parameter
Adam optimizer (2 states)	+8 bytes per parameter
Gradients	+4 bytes per parameter
Activations and temp memory (variable size)	+8 bytes per parameter (high-end estimate)
TOTAL	**=4 bytes per parameter +20 extra bytes per parameter**

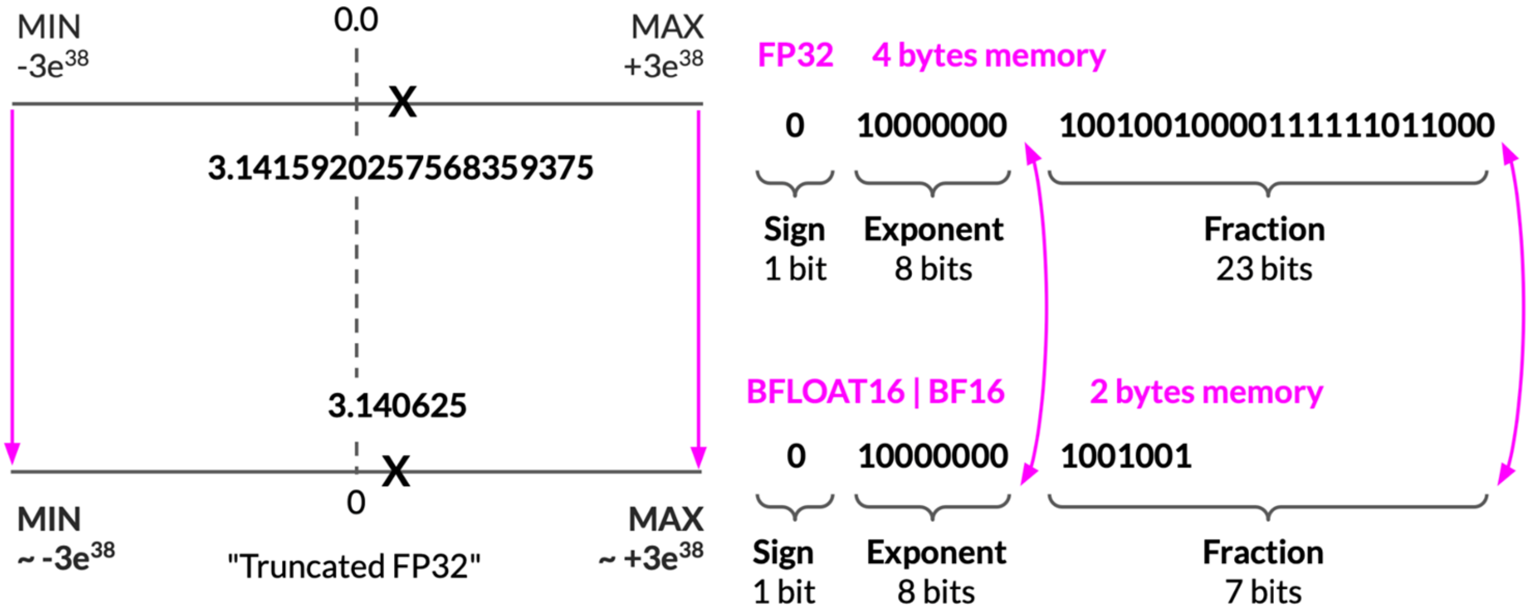
[3] https://www.coursera.org/learn/generative-ai-with-llms
[11] https://huggingface.co/docs/transformers/v4.20.1/en/perf_train_gpu_one#anatomy-of-models-memory
[12] https://www.youtube.com/live/g68qlo9Izf0

One Solution: Quantization from FP32 to FP16



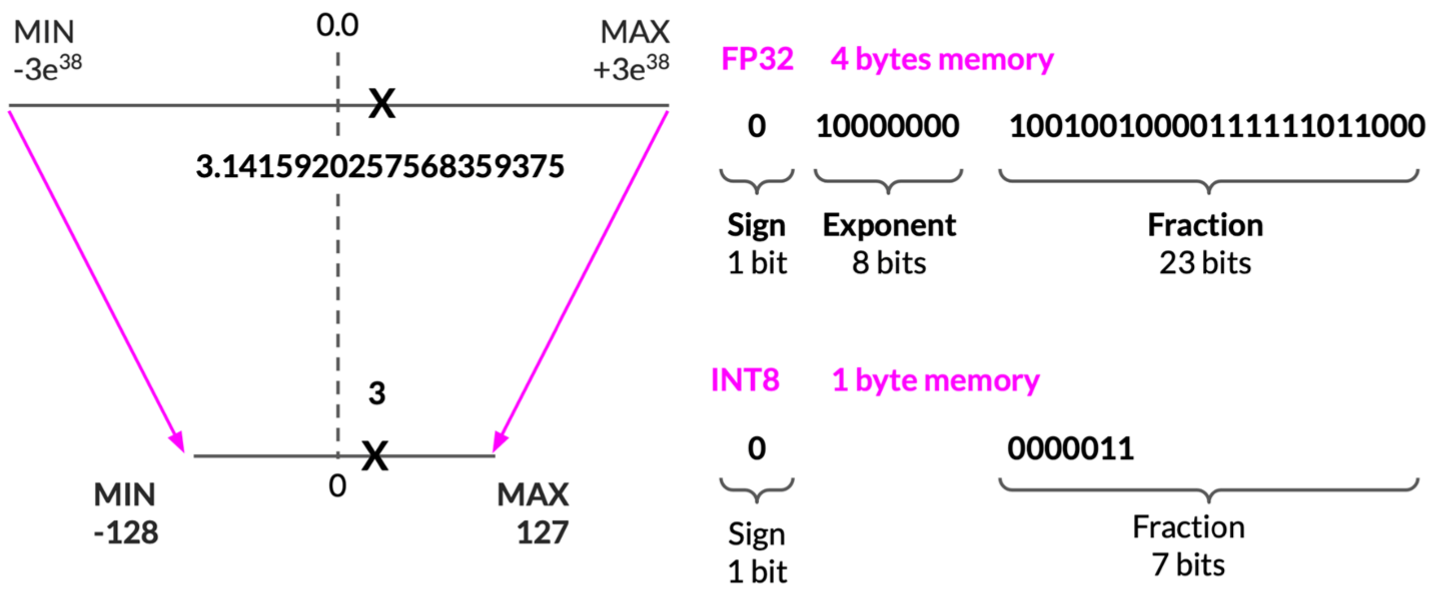
[3] https://www.coursera.org/learn/generative-ai-with-llms/

One Solution: Quantization from FP32 to BFLOAT16



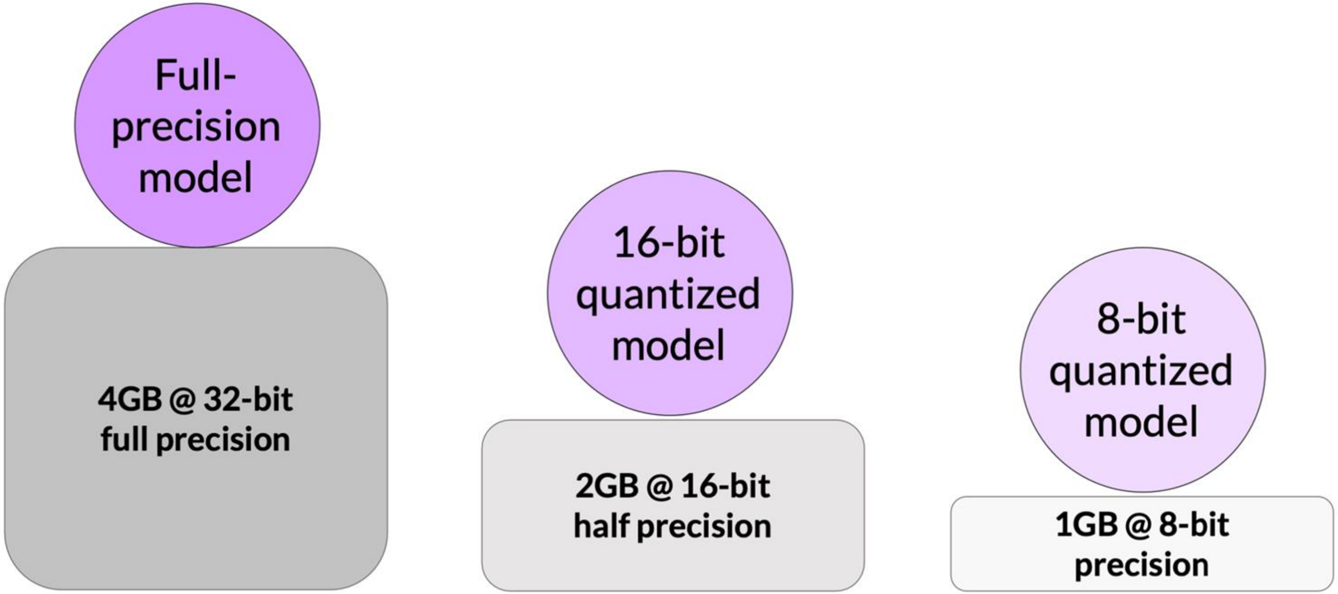
[3] https://www.coursera.org/learn/generative-ai-with-llms/

One Solution: Quantization from FP32 to INT8



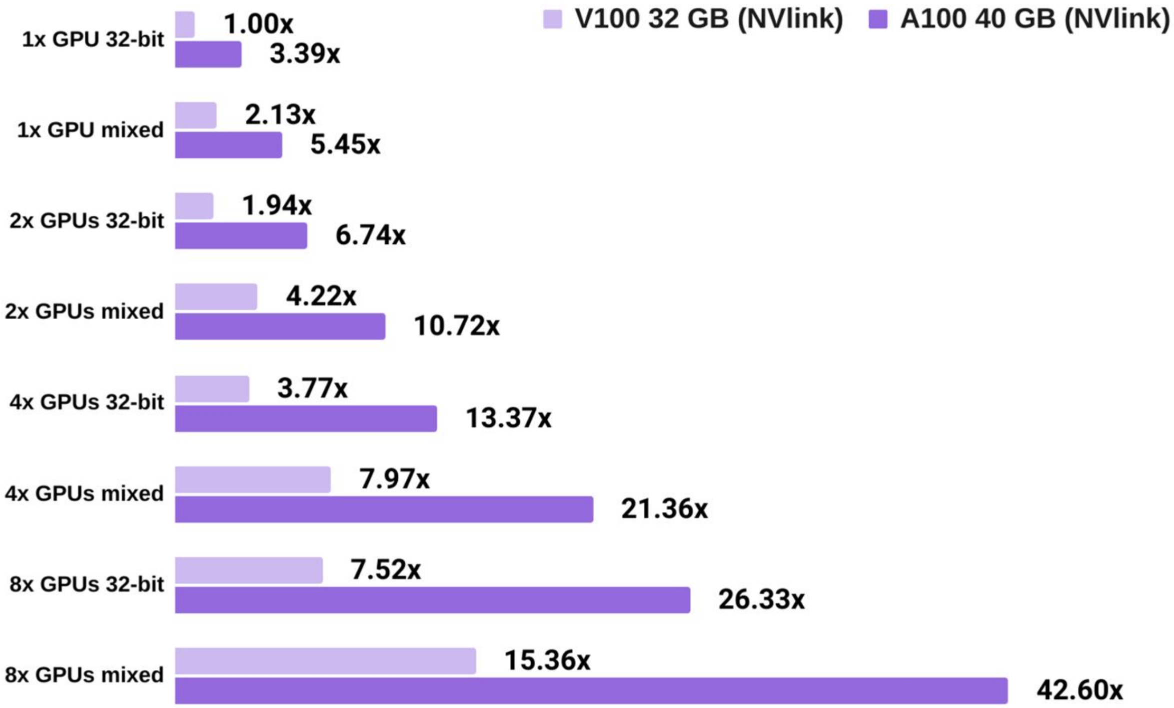
[3] https://www.coursera.org/learn/generative-ai-with-llms/

Approximate GPU Needed for a 1B Parameter



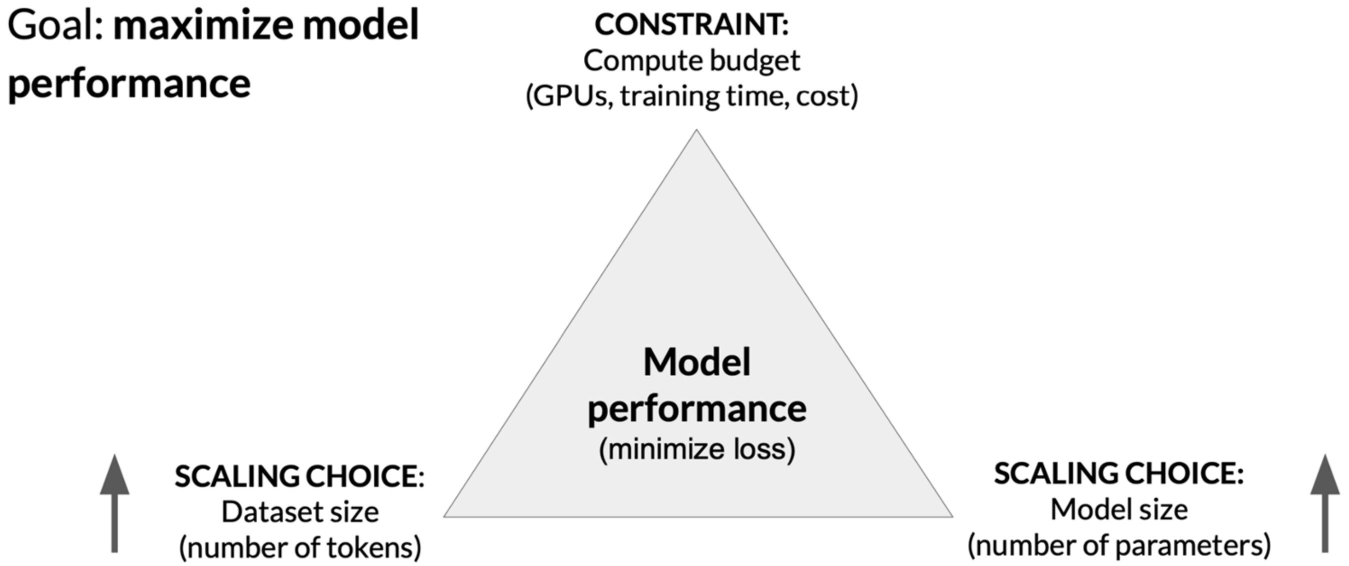
[3] https://www.coursera.org/learn/generative-ai-with-llms/

LLM Training Speed of V100 and A100



[13] https://lambdalabs.com/blog/nvidia-a100-vs-v100-benchmarks

Scaling Choices for Pre-training



[3] https://www.coursera.org/learn/generative-ai-with-llms/

RIT

Thank you!

Reza Fayyazi
rf1679@rit.edu