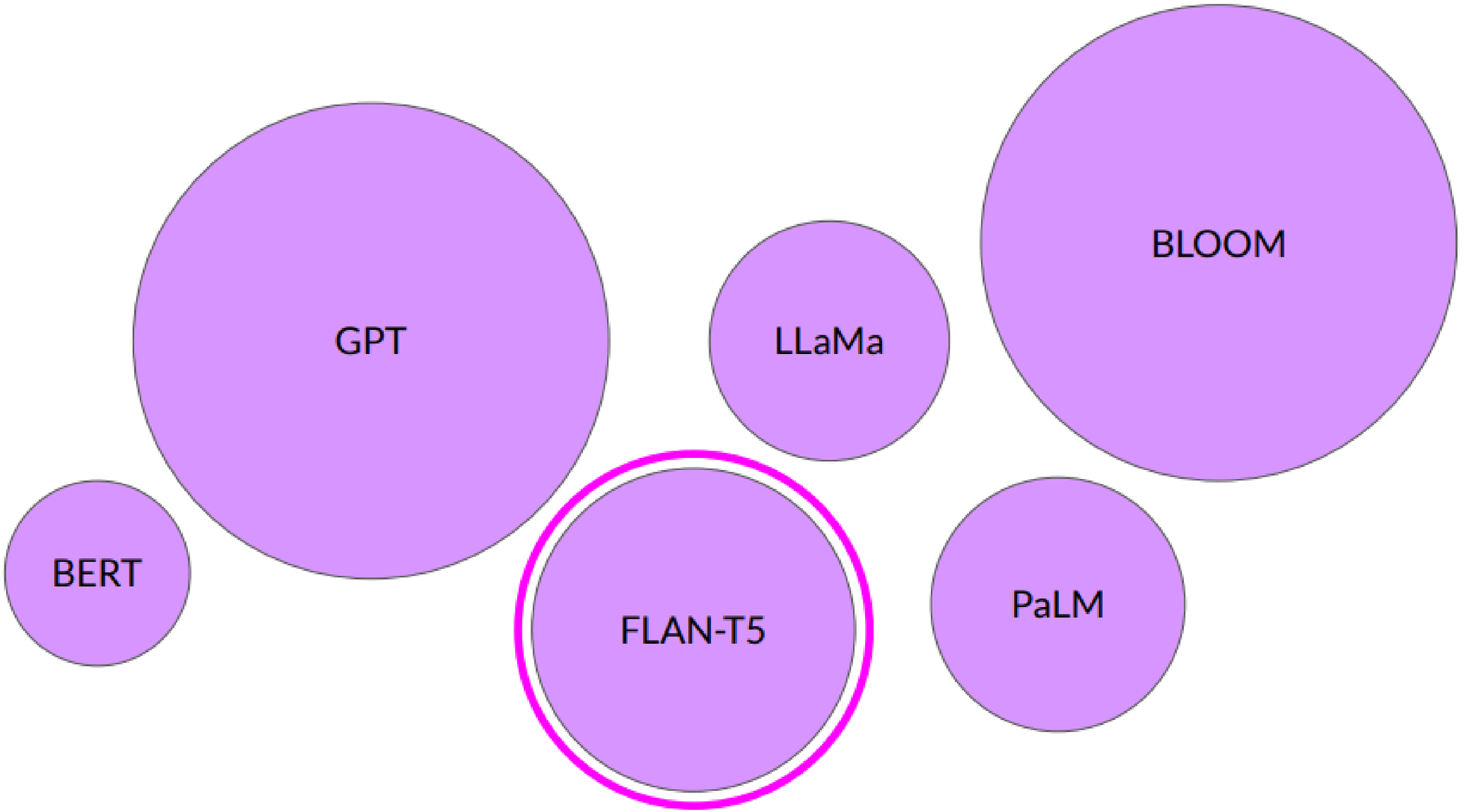
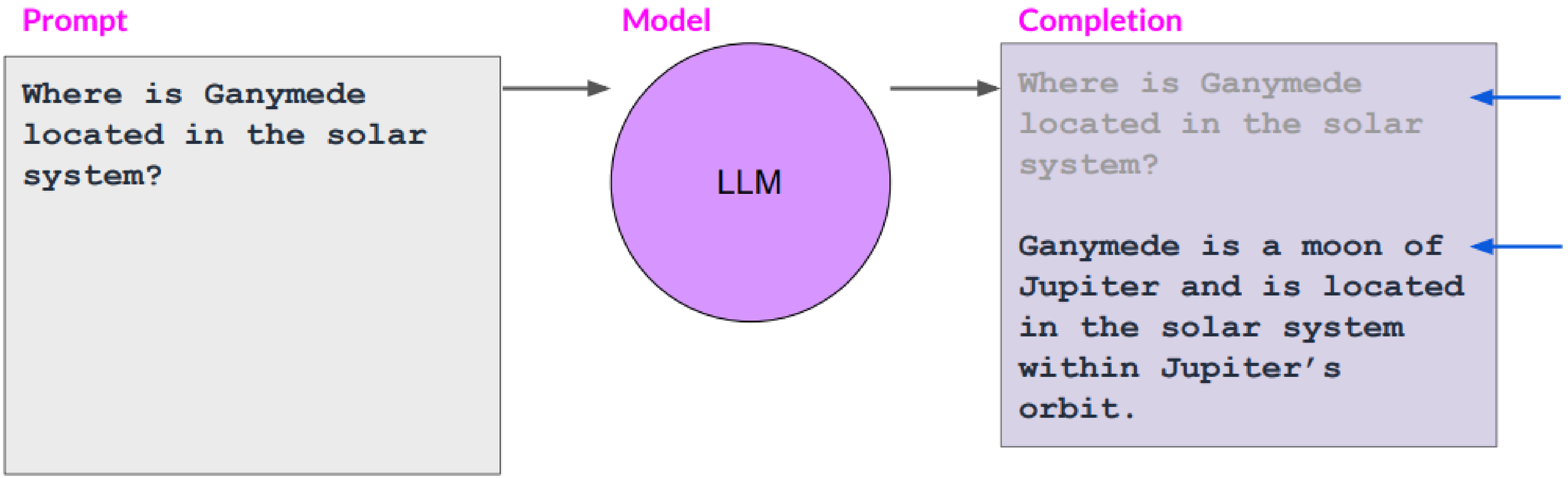
Large Language Models



Prompts and completions

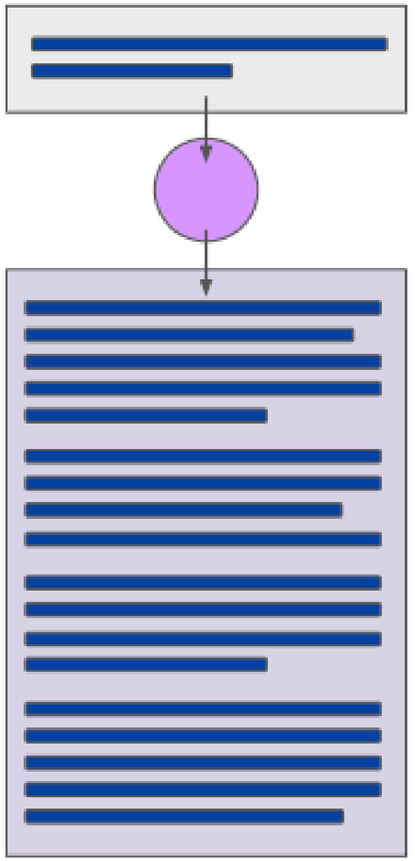


Context window

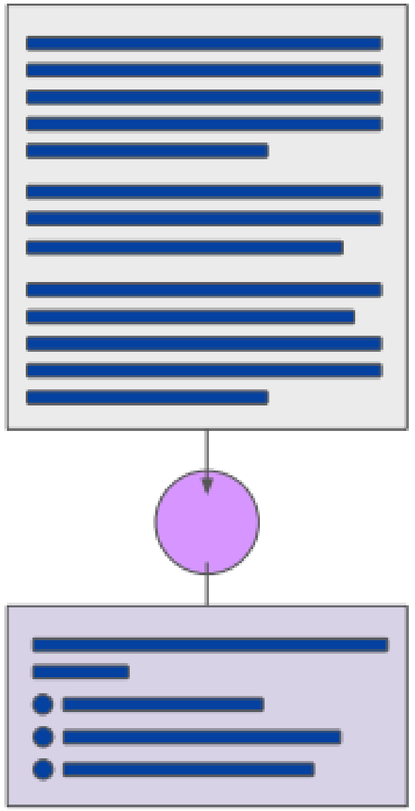
- typically a few 1000 words.

LLM use cases & tasks

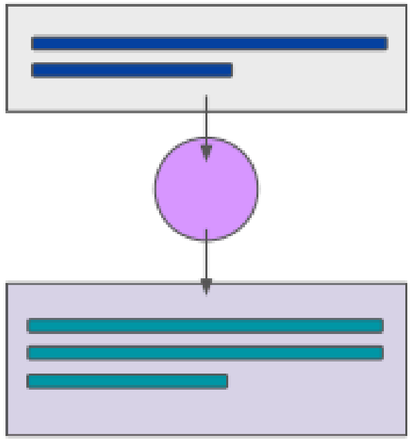
Essay Writing



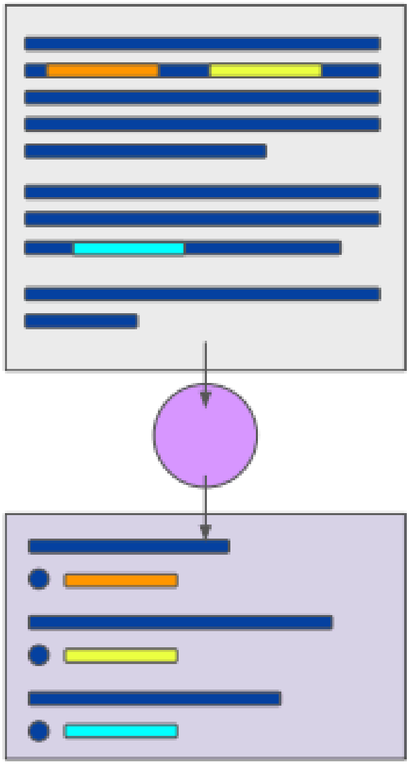
Summarization



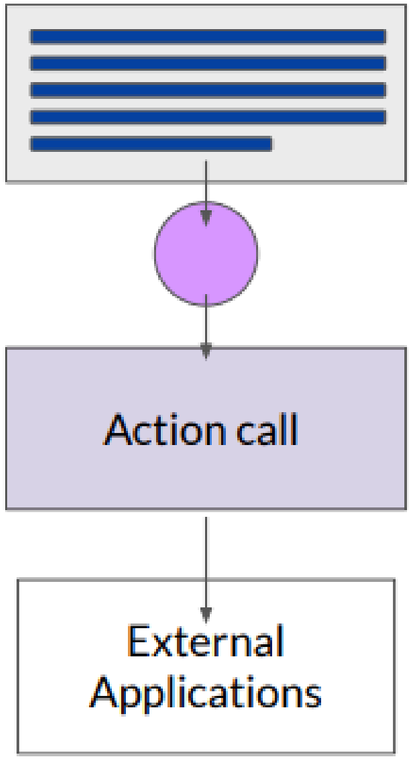
Translation



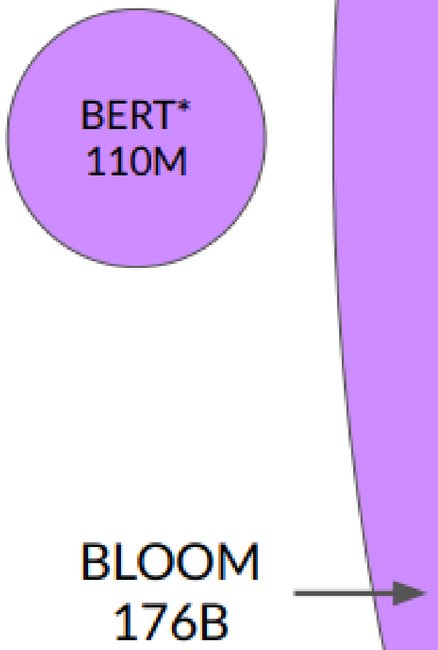
Information retrieval



Invoke APIs and actions



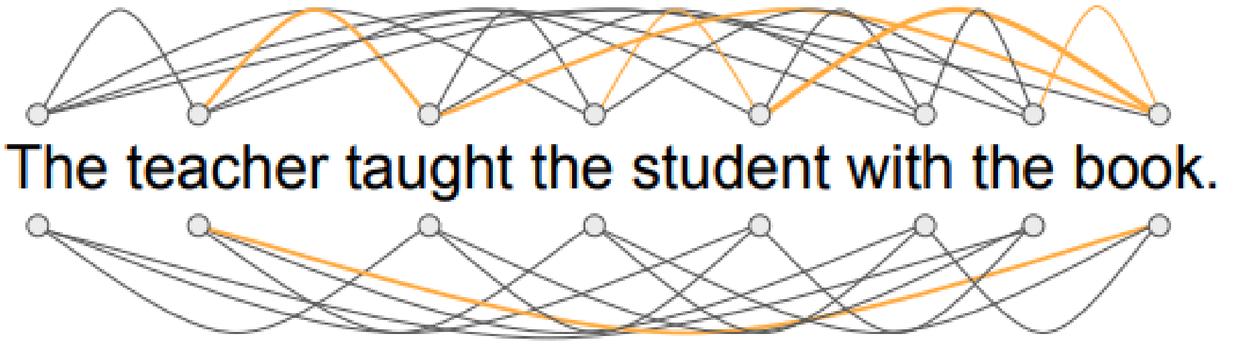
The significance of scale: language understanding



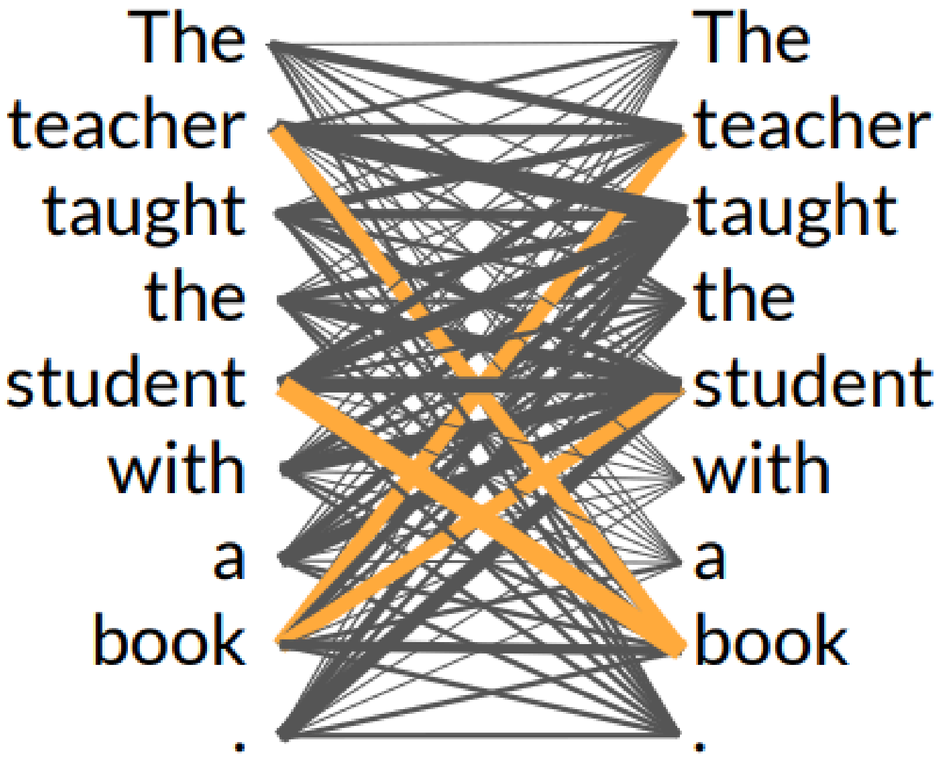
*Bert-base

How LLMs work - Transformers architecture

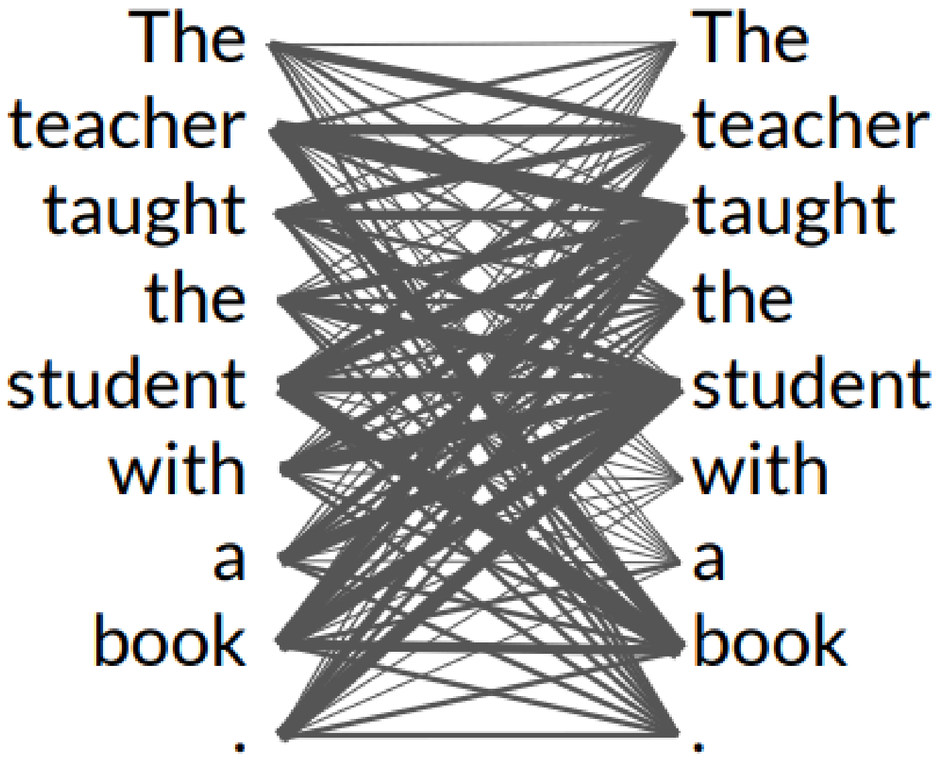
Transformers



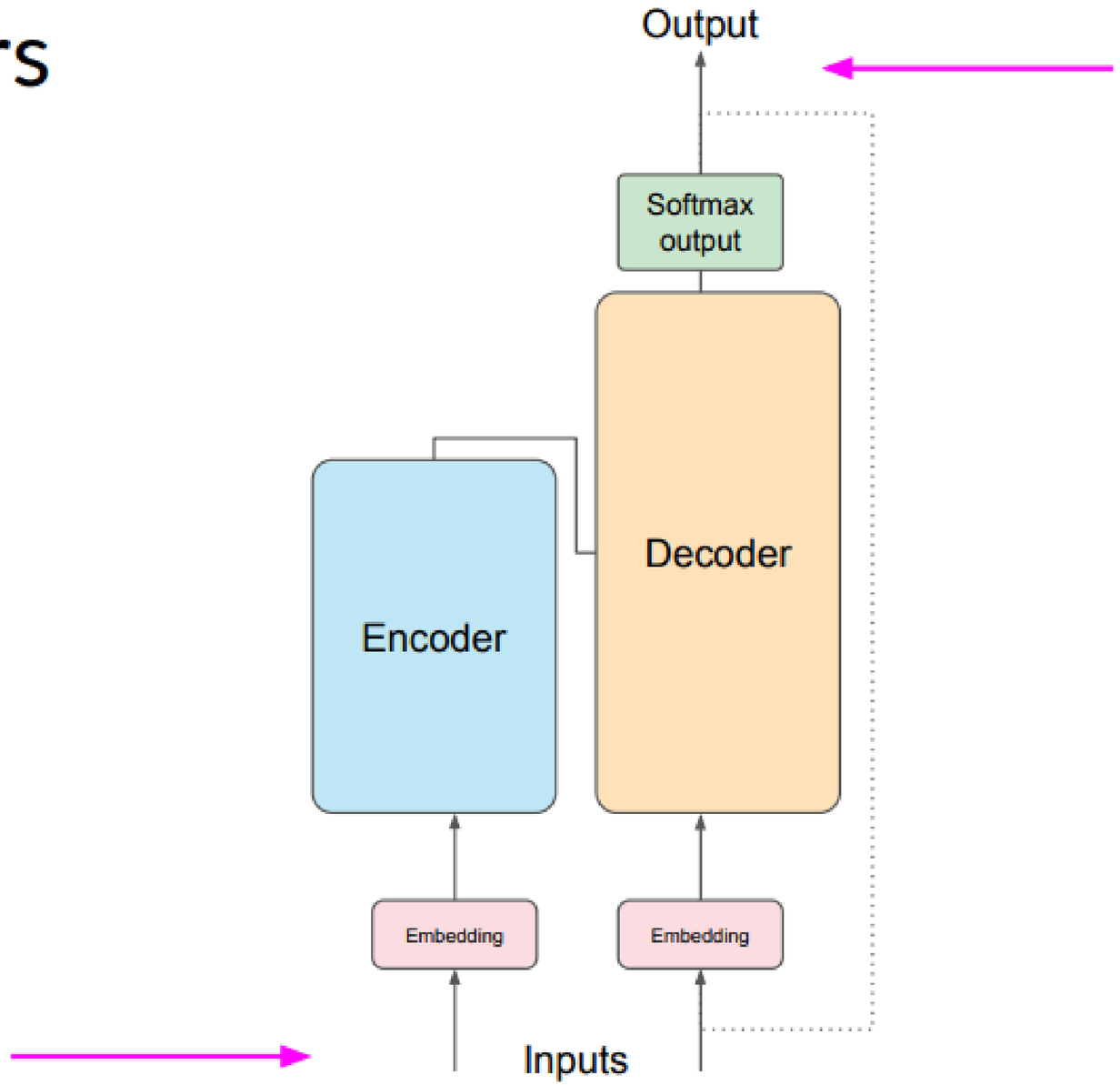
Self-attention



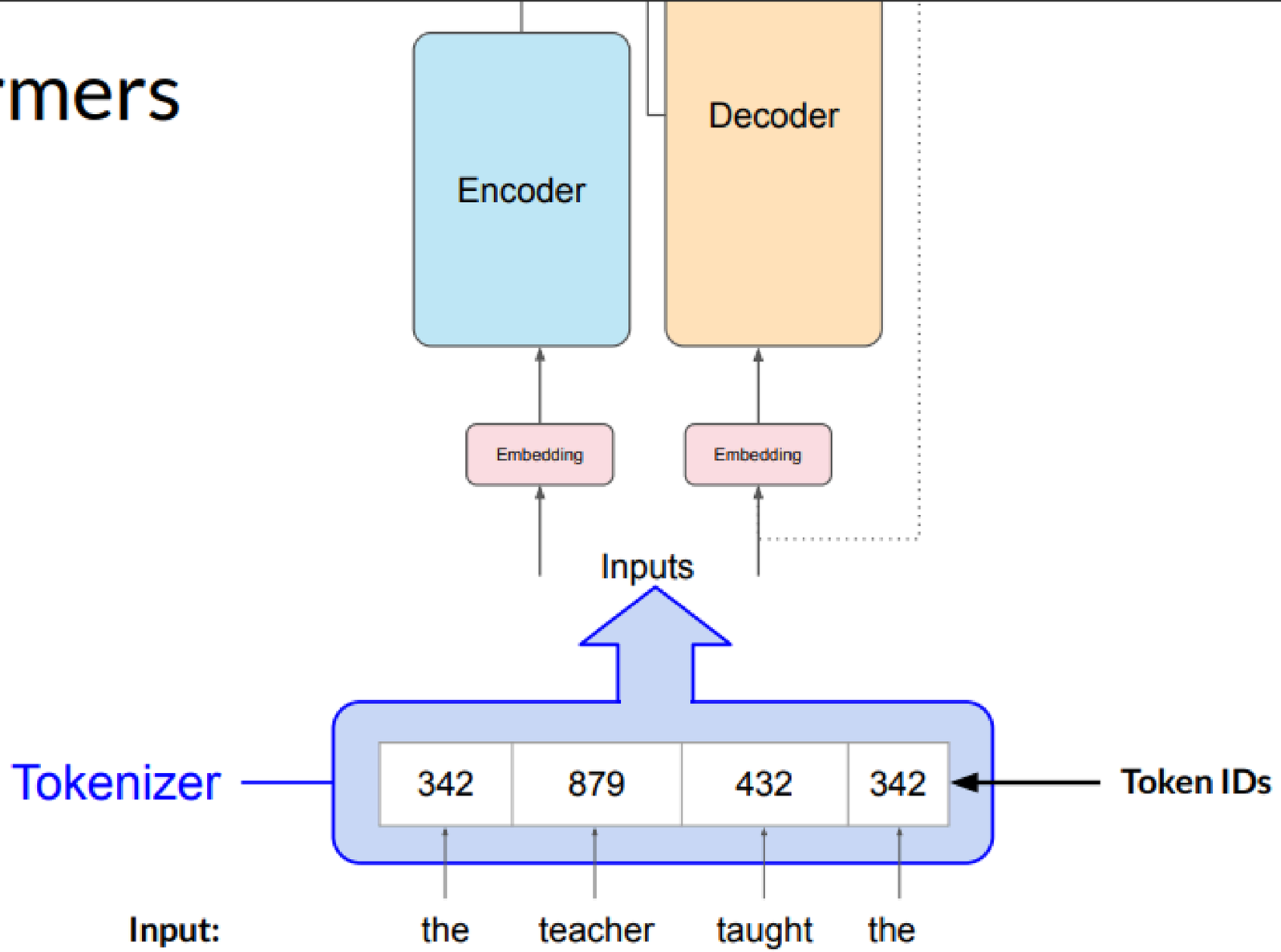
Self-attention



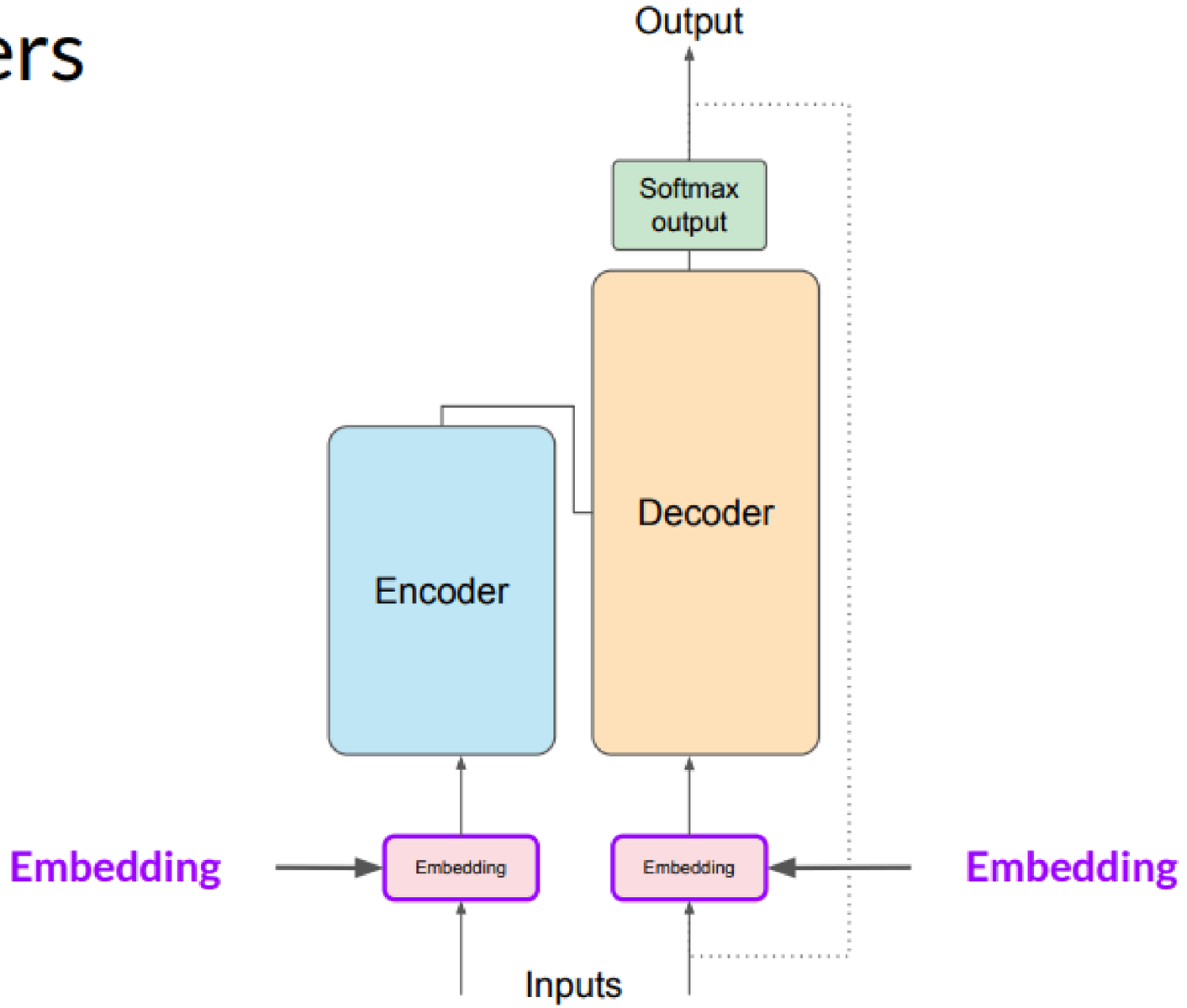
Transformers



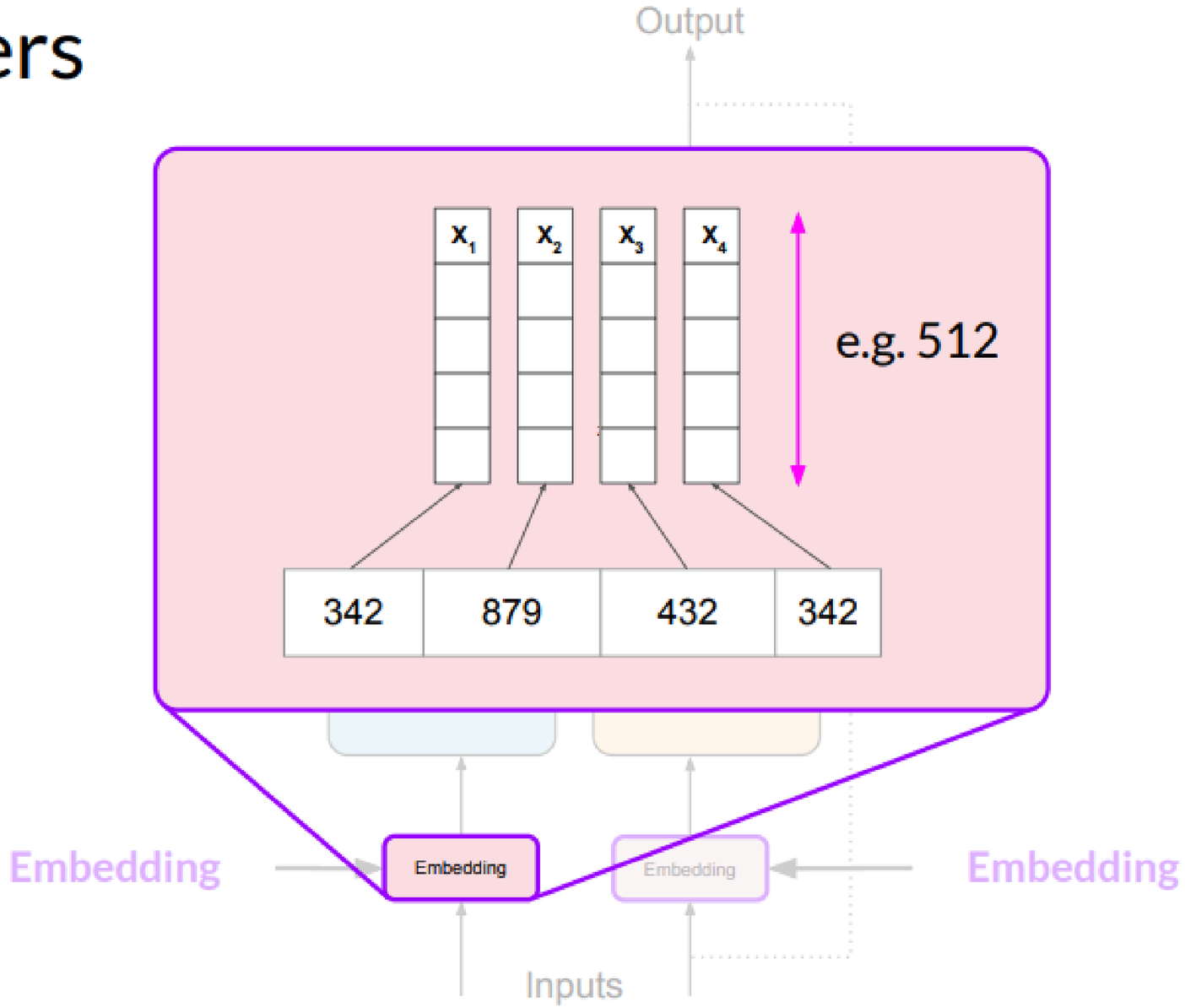
Transformers



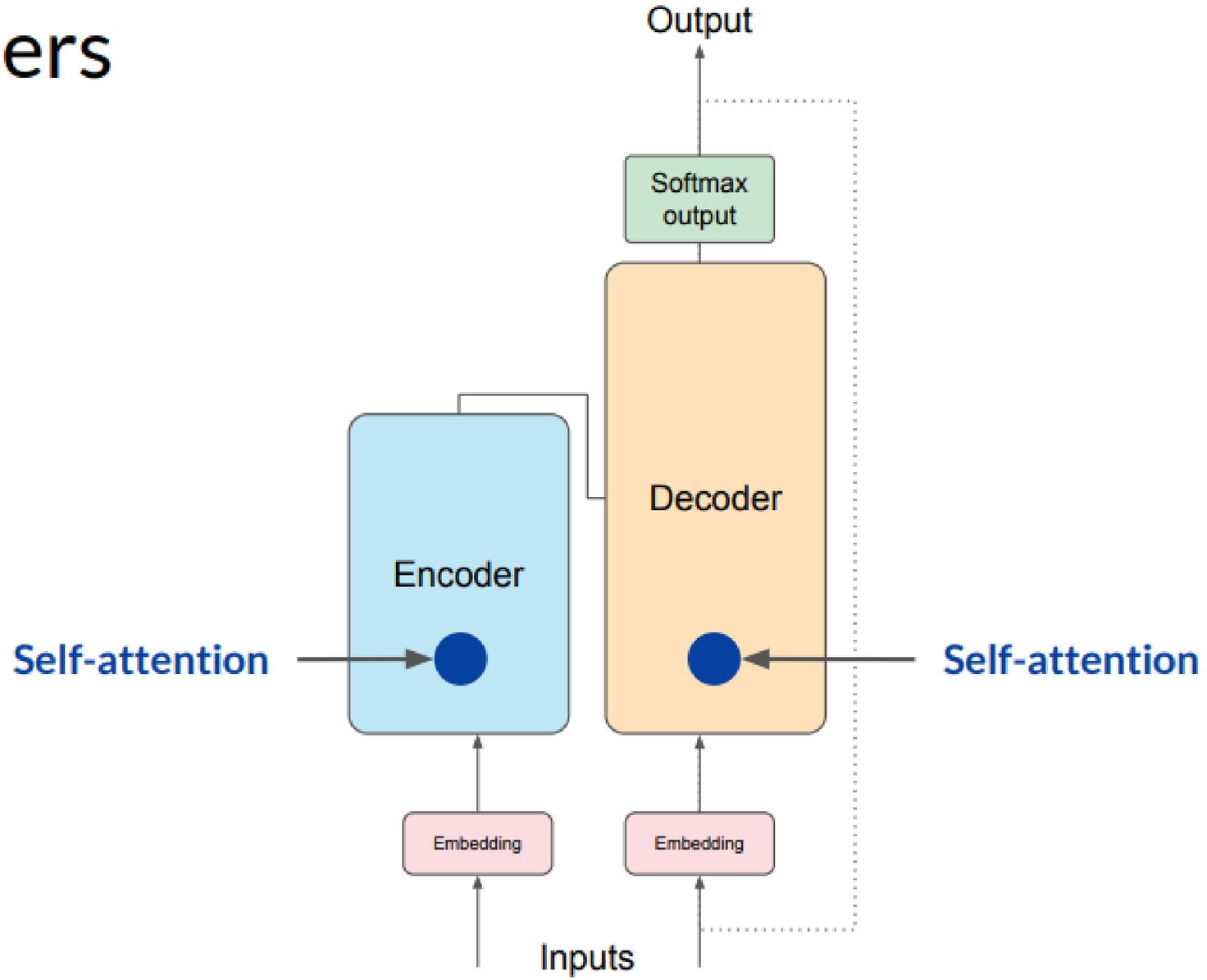
Transformers



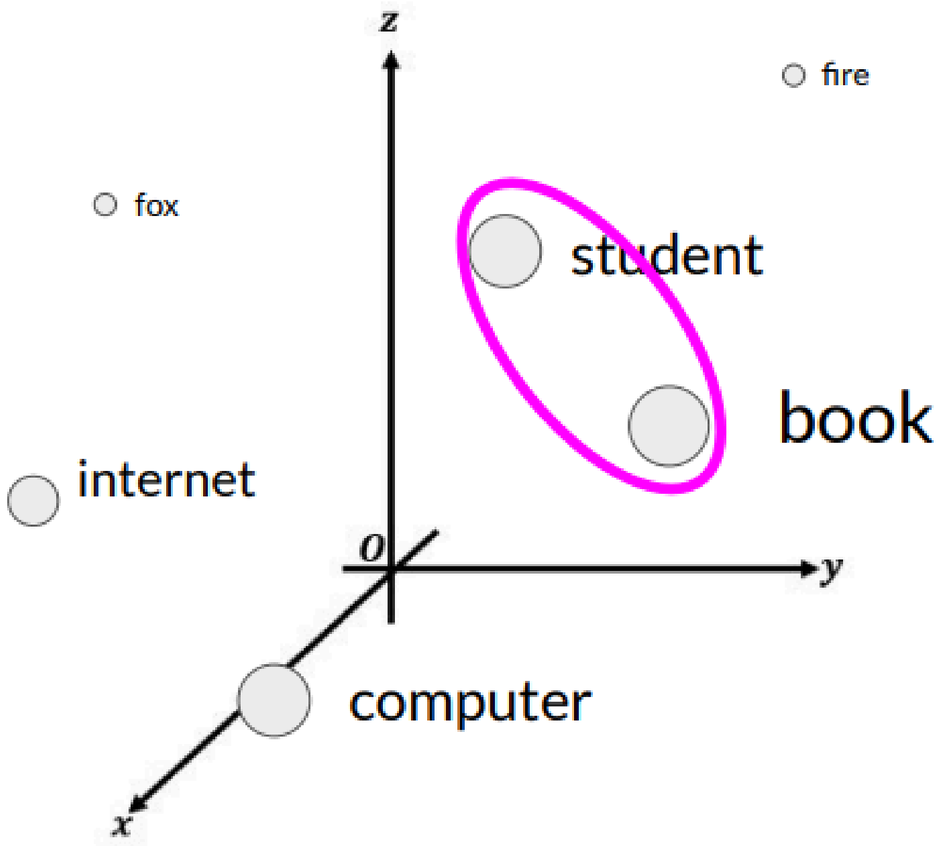
Transformers



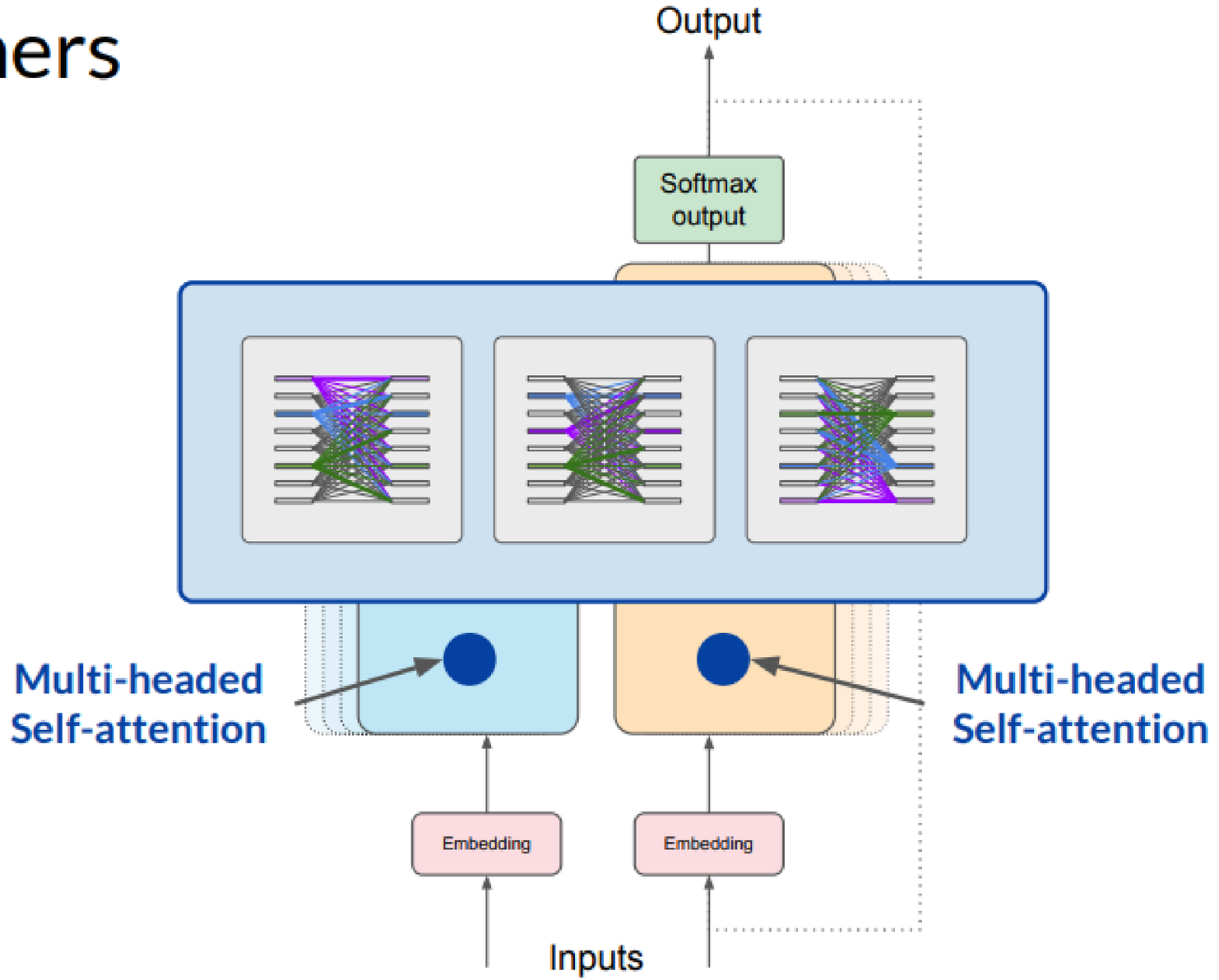
Transformers



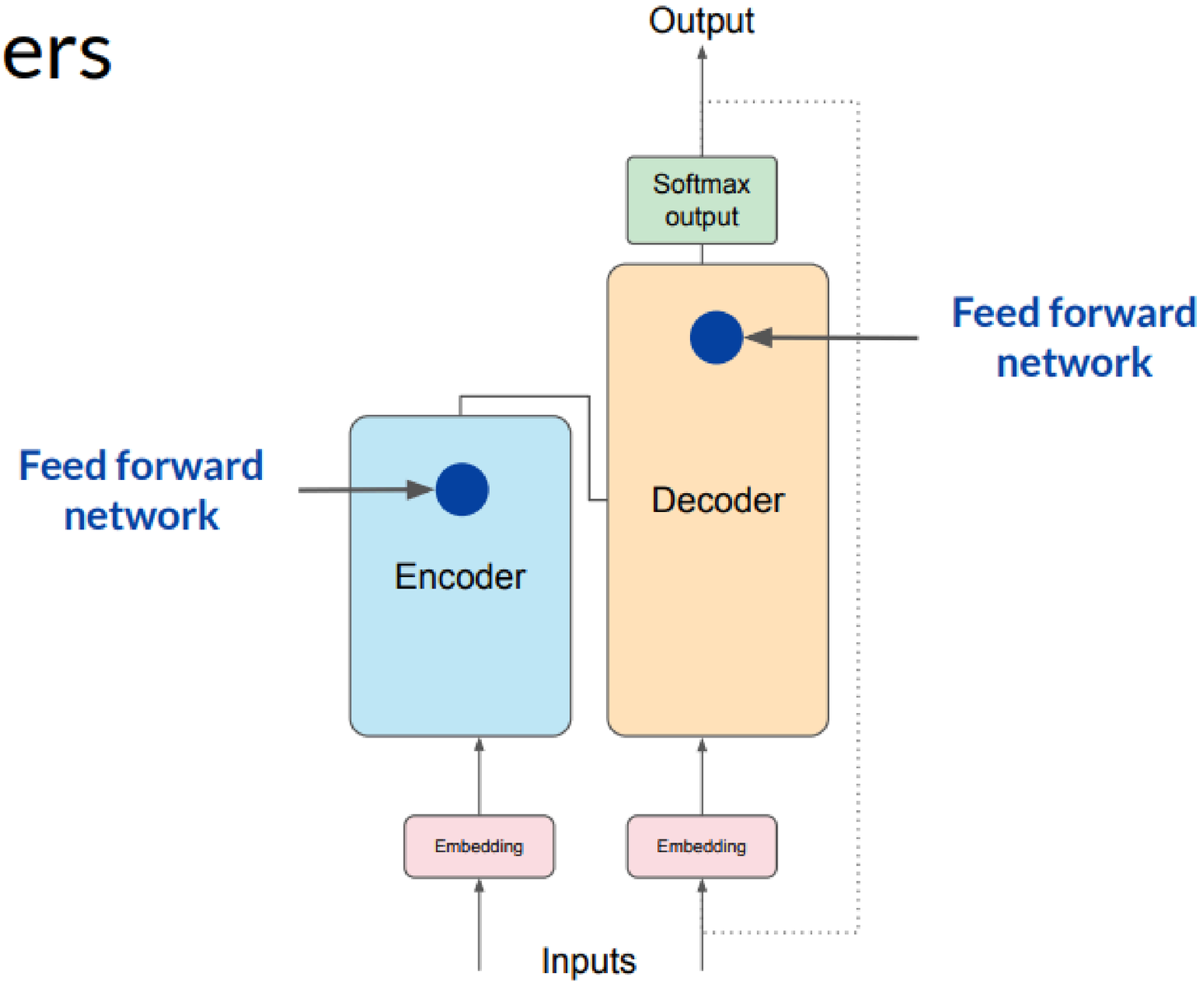
Transformers



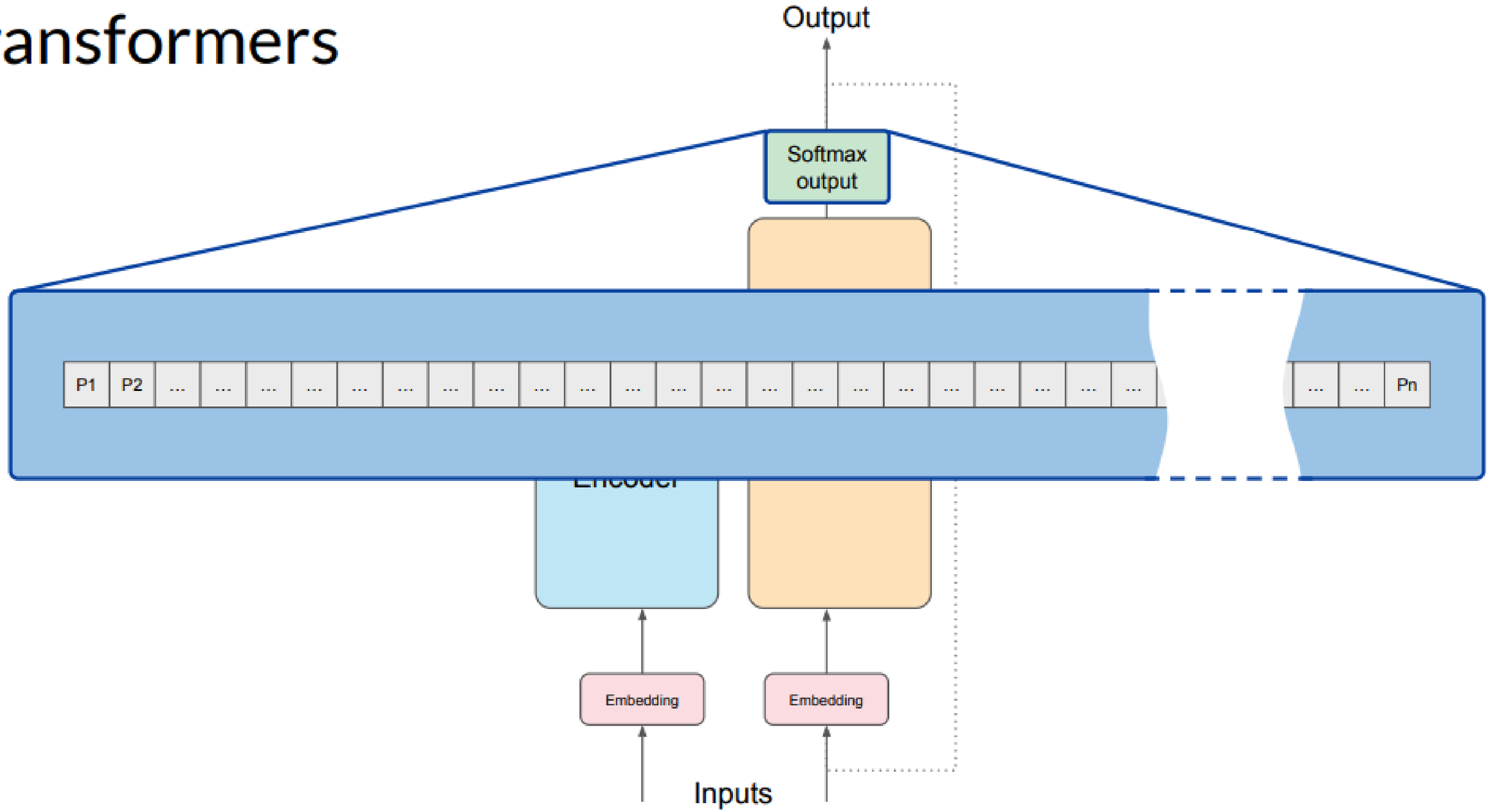
Transformers



Transformers

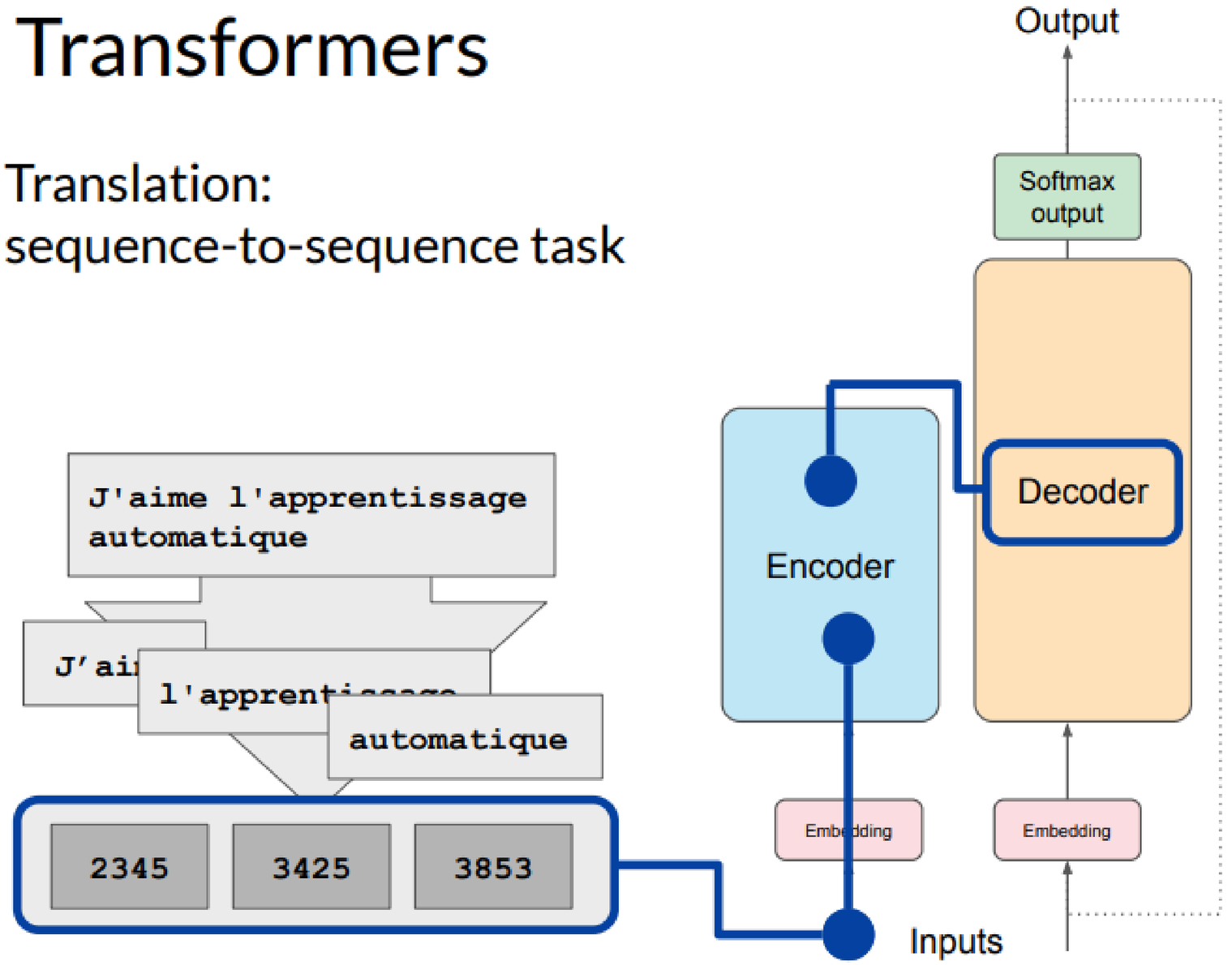


Transformers



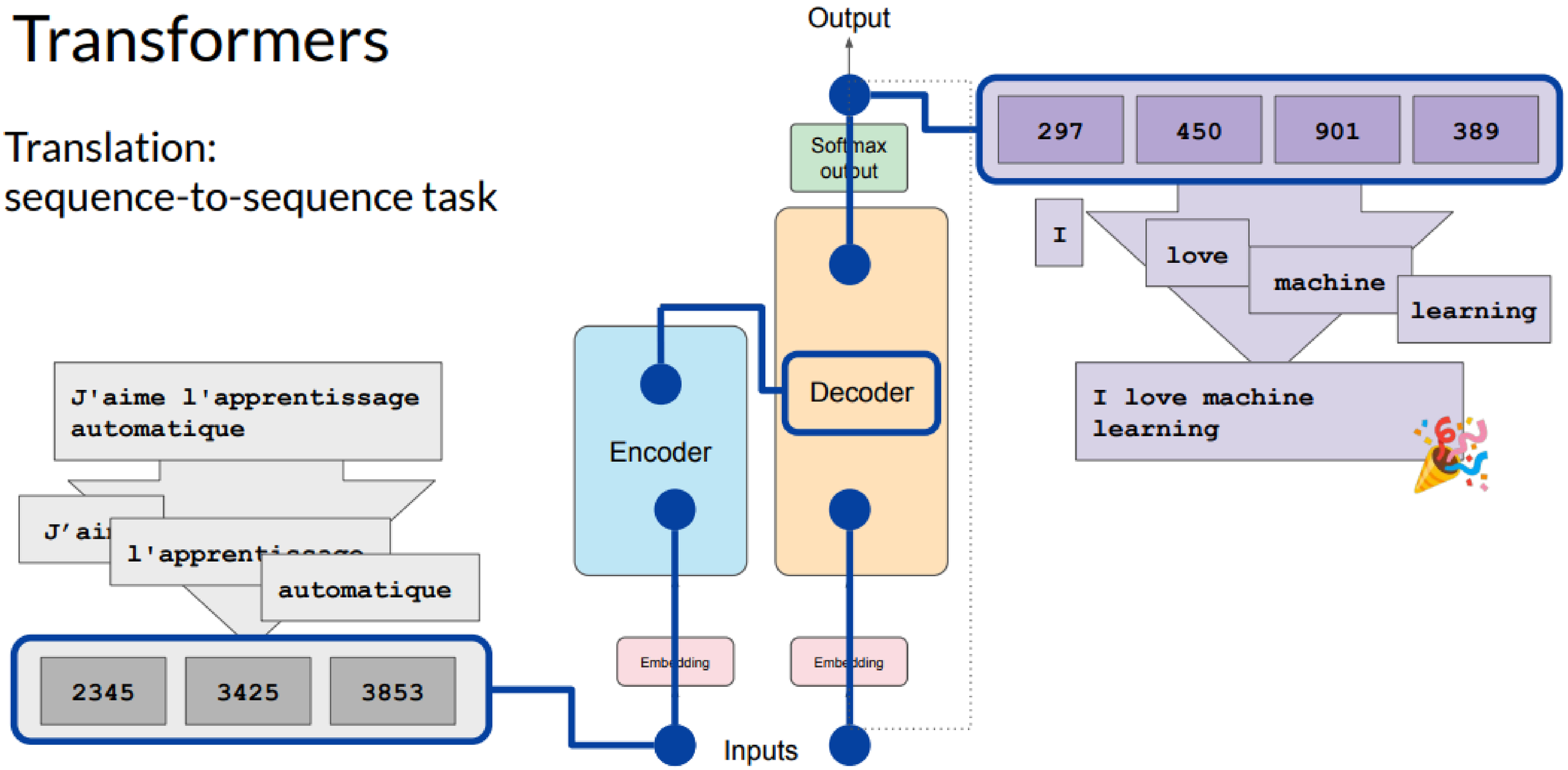
Transformers

Translation:
sequence-to-sequence task



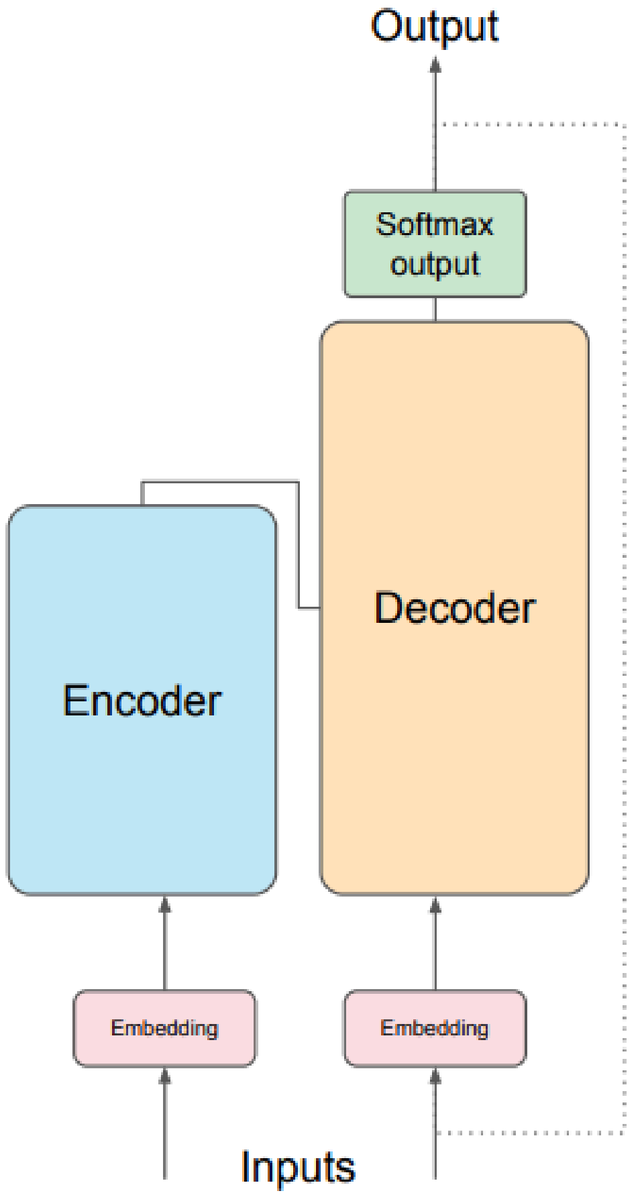
Transformers

Translation:
sequence-to-sequence task



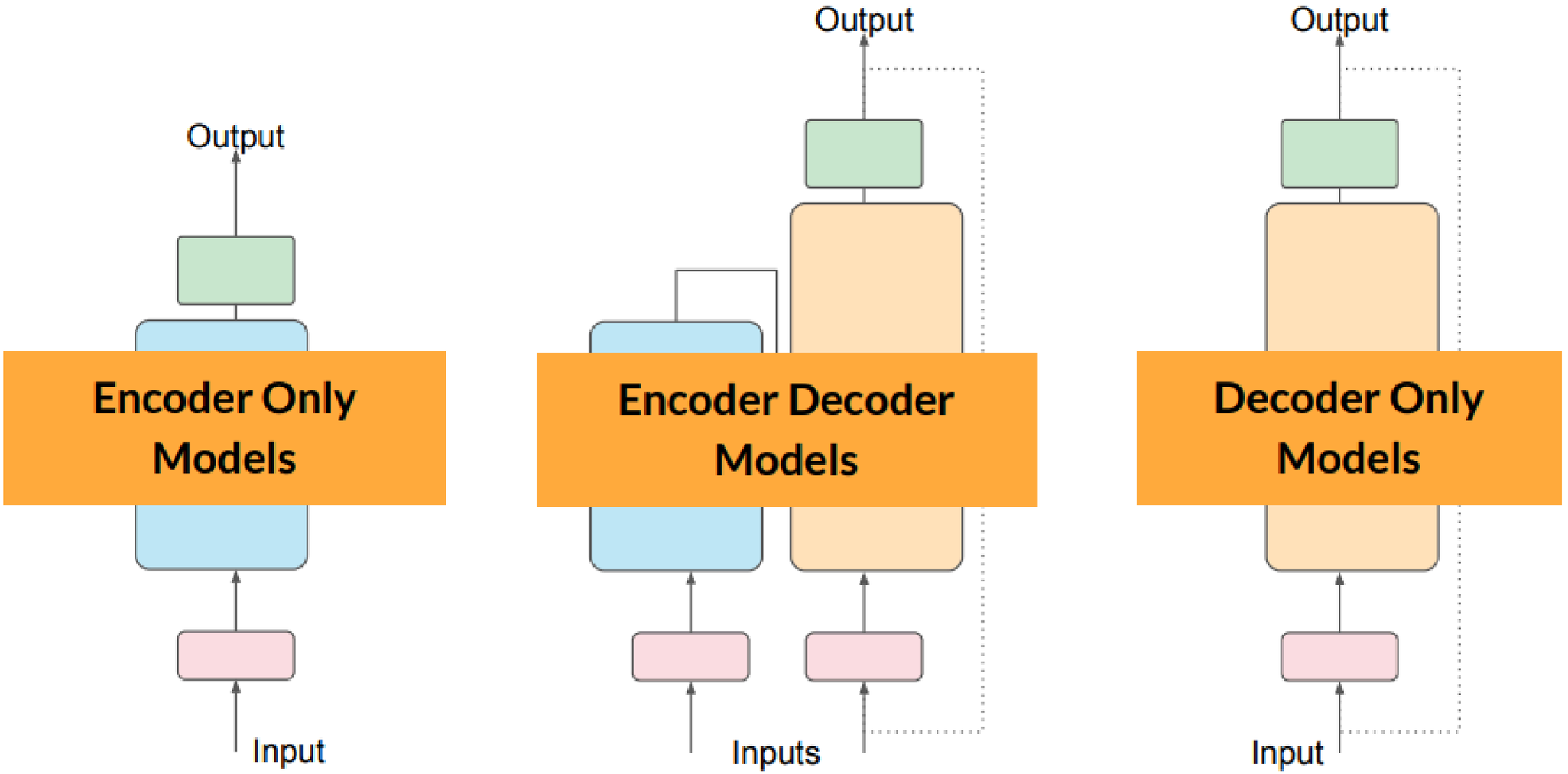
Transformers

Encoder
Encodes inputs (“prompts”) with contextual understanding and produces one vector per input token.



Decoder
Accepts input tokens and generates new tokens.

Transformers



Summary of in-context learning (ICL)

Prompt // Zero Shot

```
Classify this review:
I loved this movie!
Sentiment:
```

Context Window
(few thousand words)

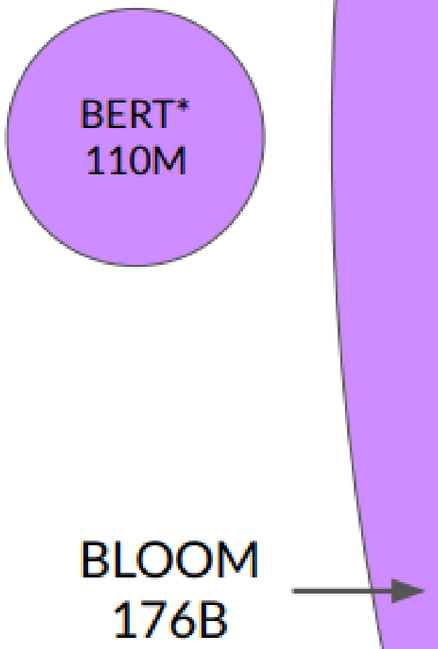
Prompt // One Shot

```
Classify this review:
I loved this movie!
Sentiment: Positive

Classify this review:
I don't like this
chair.
Sentiment:
```

Prompt // Few Shot >5 or 6 examples

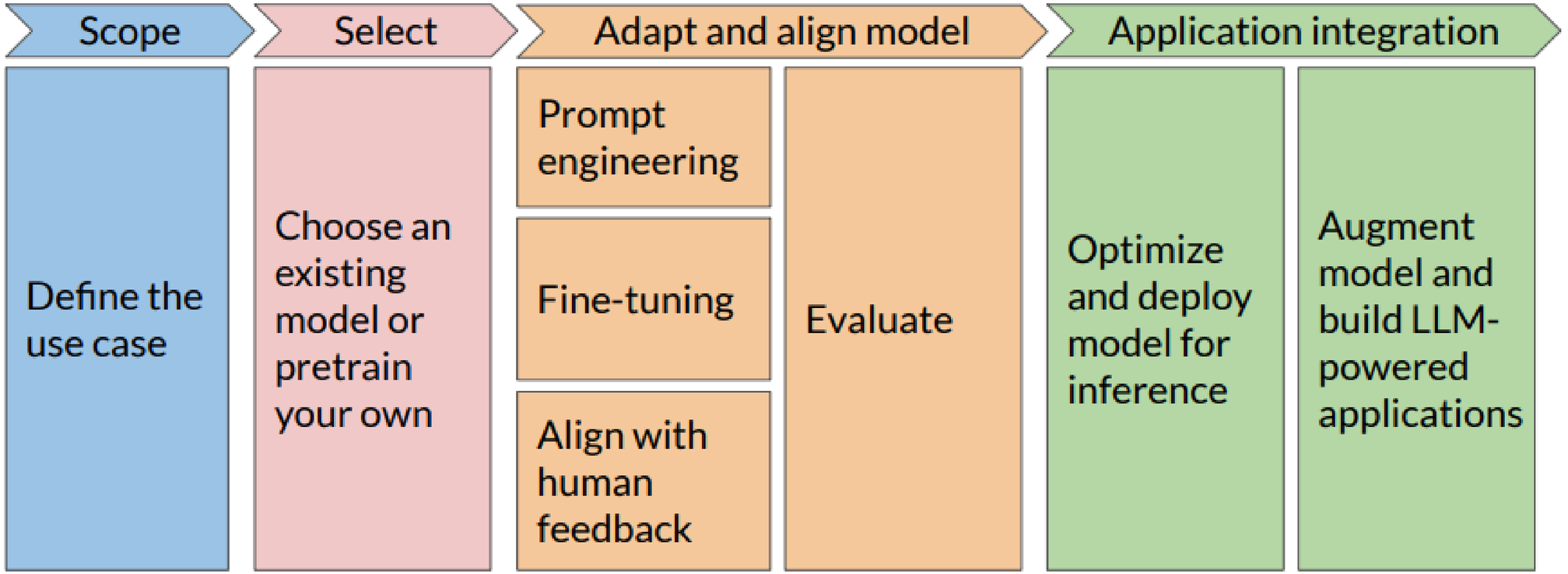
```
Classify this review:
I loved this movie!
Sentiment: Positive

Classify this review:
I don't like this
chair.
Sentiment: Negative

Classify this review:
Who would use this
product?
Sentiment:
```

The significance of scale: task ability

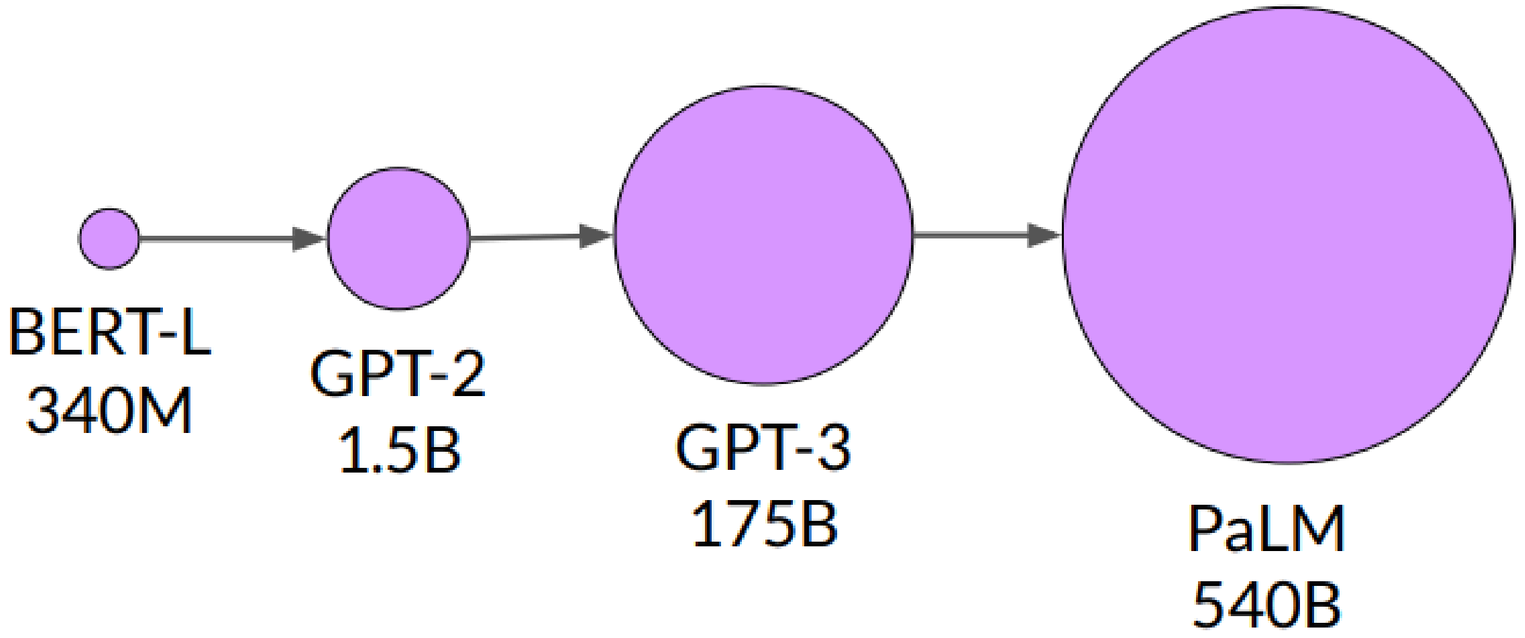


*Bert-base

Generative AI project lifecycle



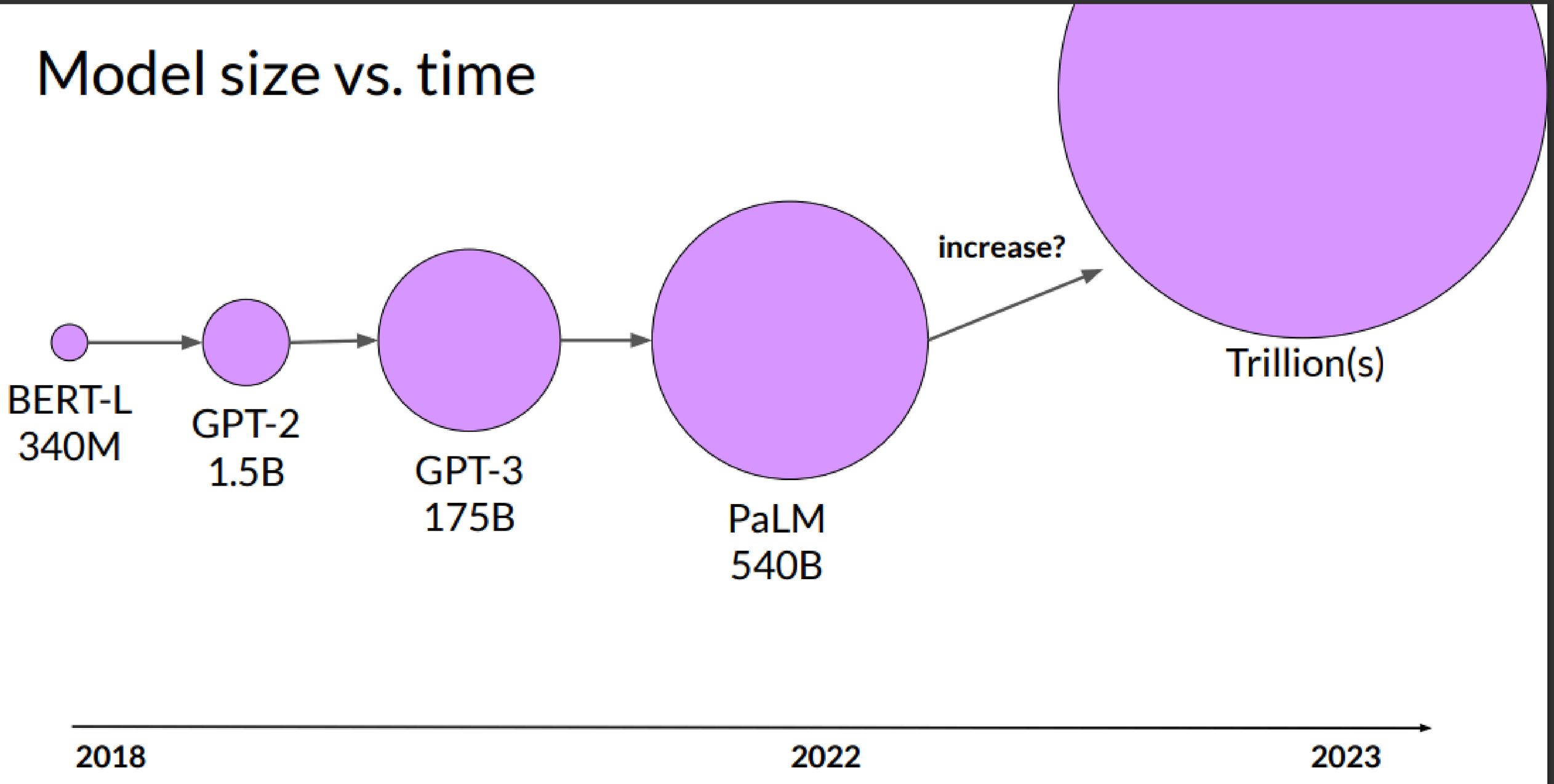
Model size vs. time



Growth powered by:

- Introduction of transformer
- Access to massive datasets
- More powerful compute resources

Model size vs. time



Computational challenges

```
OutOfMemoryError: CUDA out of memory.
```

Approximate GPU RAM needed to store 1B parameters

1 parameter = 4 bytes (32-bit float)

1B parameters = 4×10^9 bytes = 4GB

4GB @ 32-bit full precision

Sources: https://huggingface.co/docs/transformers/v4.20.1/en/perf_train_gpu_one#anatomy-of-models-memory, https://github.com/facebookresearch/bitsandbytes

Additional GPU RAM needed to train 1B parameters

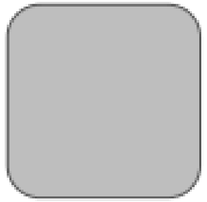
	Bytes per parameter
Model Parameters (Weights)	4 bytes per parameter

~20 extra bytes per parameter

Sources: https://huggingface.co/docs/transformers/v4.20.1/en/perf_train_gpu_one#anatomy-of-models-memory, https://github.com/facebookresearch/bitsandbytes

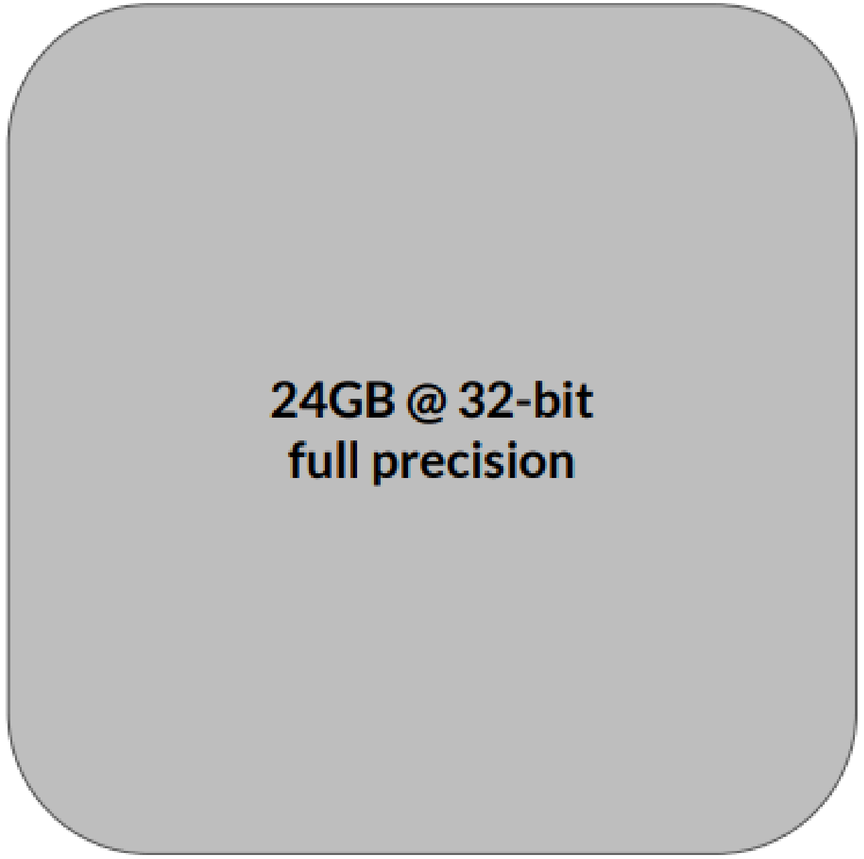
Approximate GPU RAM needed to train 1B-params

Memory needed to store model

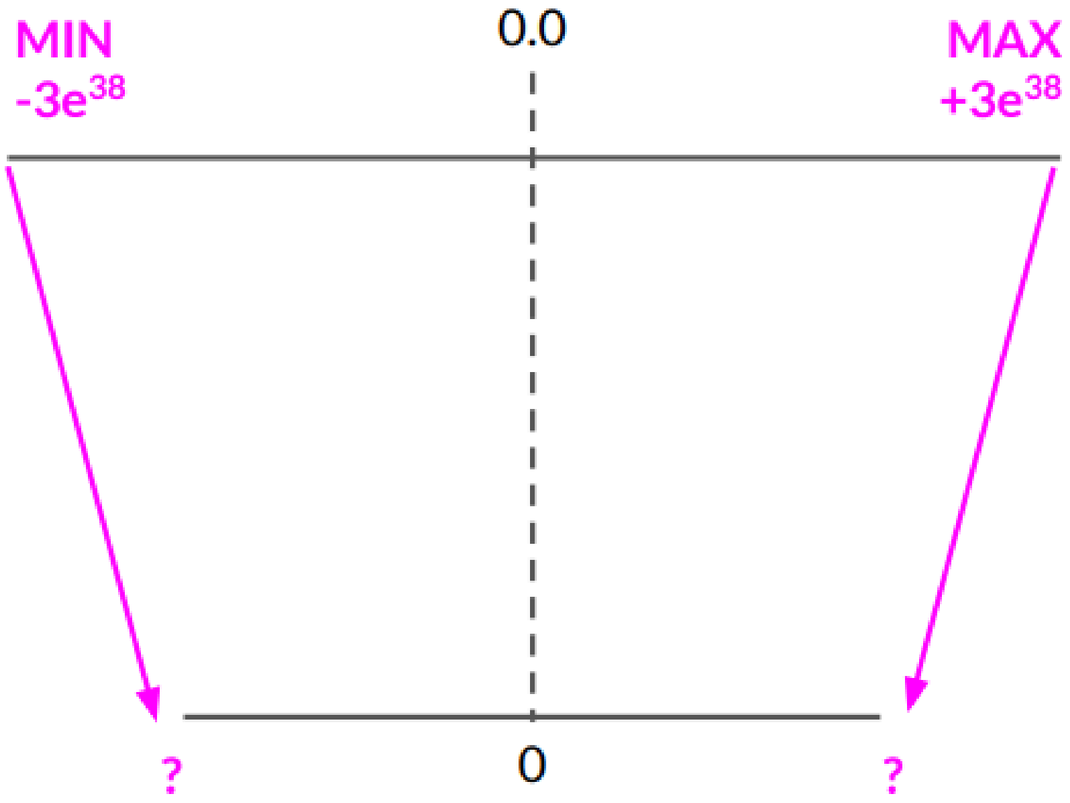


4GB @ 32-bit full precision

Memory needed to train model



Quantization



FP32

32-bit floating point

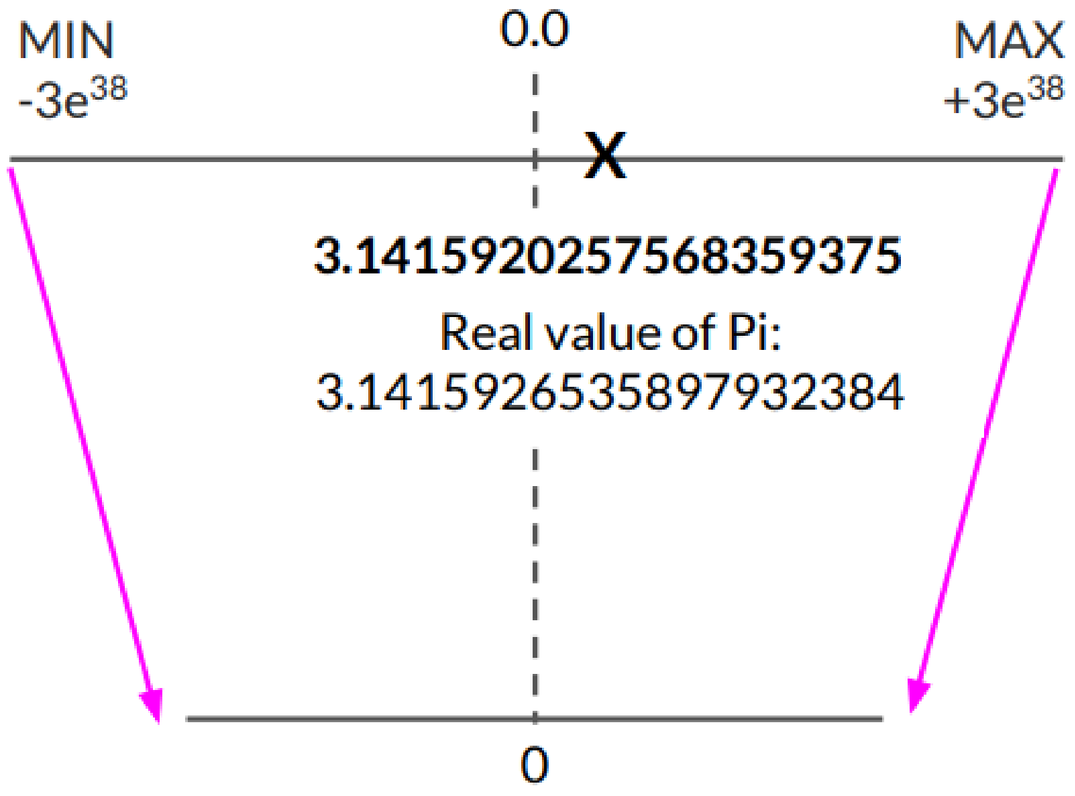
Range:

From **-3e38** to **+3e38**

FP16 | BFLOAT16 | INT8

16-bit floating point | 8-bit integer

Quantization: FP32



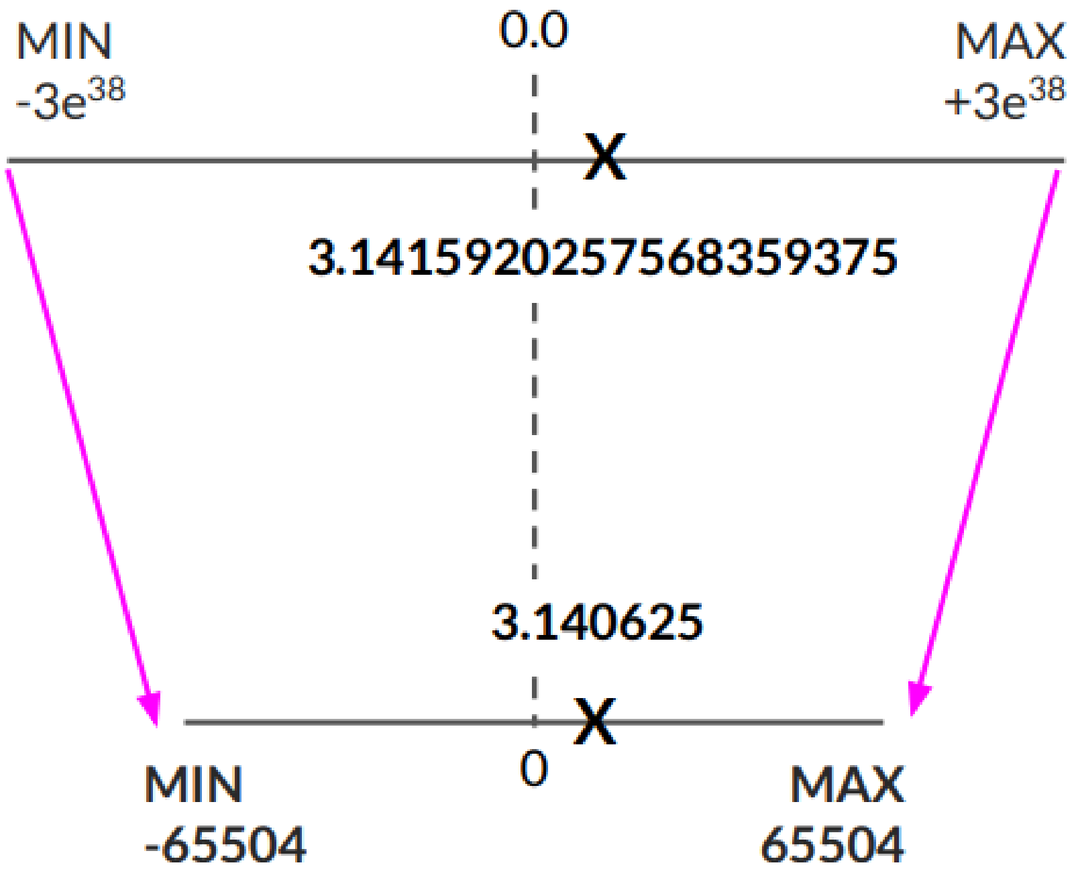
Let's store Pi: 3.141592

FP32

0	10000000	10010010000111111011000
Sign	**Exponent**	**Fraction**
1 bit	8 bits	23 bits

Mantissa / Significand
= Precision

Quantization: FP16



Let's store Pi: 3.141592

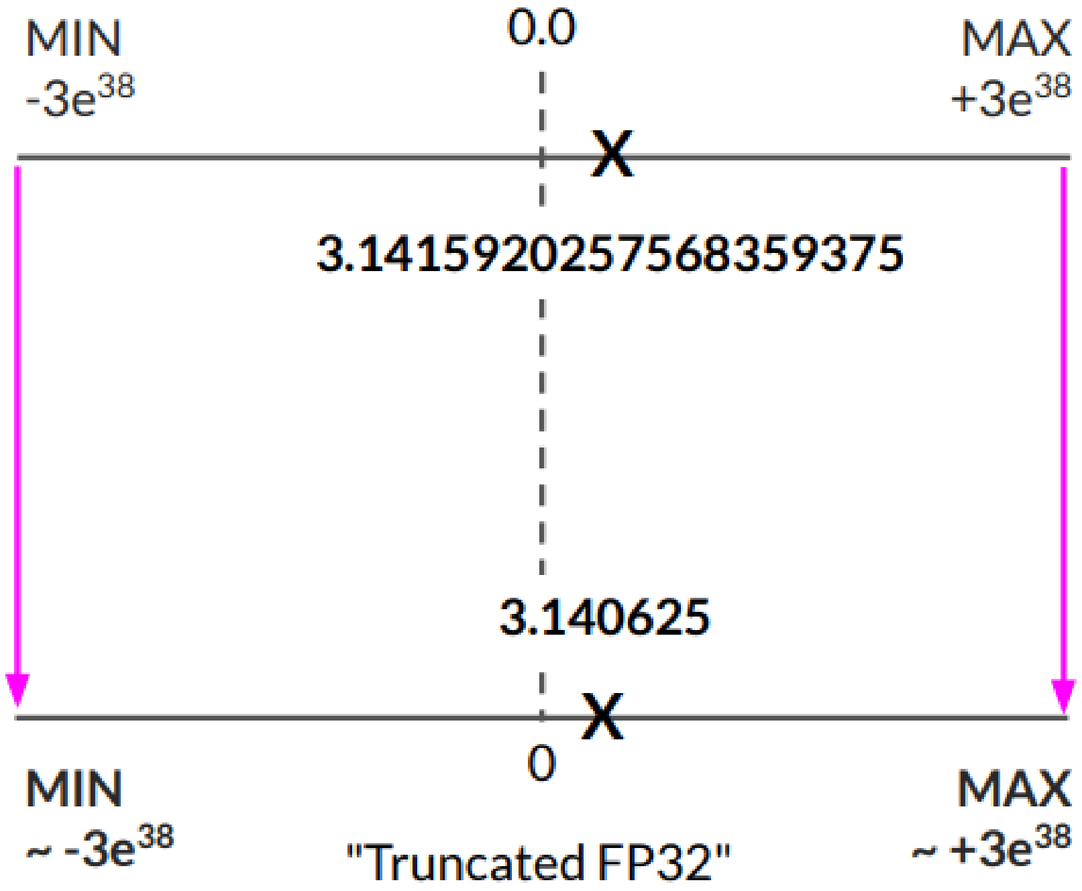
FP32 4 bytes memory

0	10000000	10010010000111111011000
Sign 1 bit	**Exponent** 8 bits	**Fraction** 23 bits

FP16 2 bytes memory

0	10000	1001001000
Sign 1 bit	**Exponent** 5 bits	**Fraction** 10 bits

Quantization: BFLOAT16



Let's store Pi: 3.141592

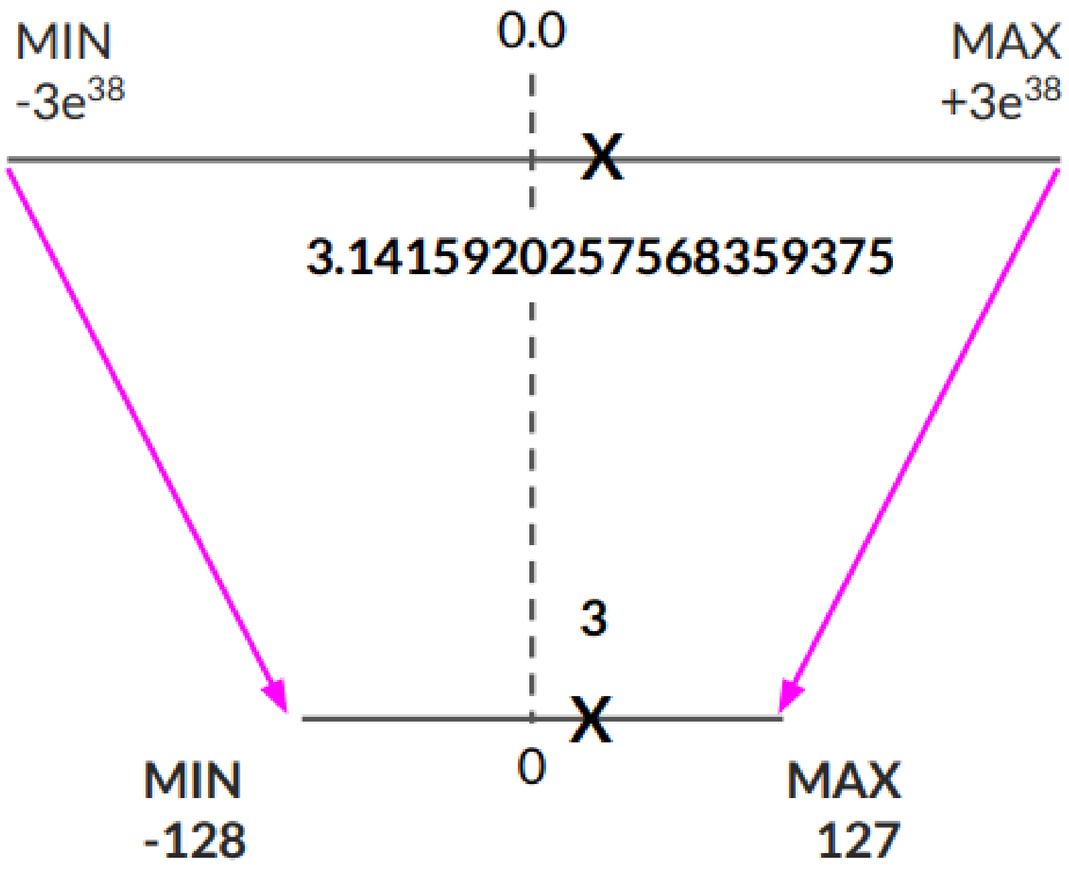
FP32 4 bytes memory

0	10000000	10010010000111111011000
Sign 1 bit	**Exponent** 8 bits	**Fraction** 23 bits

BFLOAT16 | BF16 2 bytes memory

0	10000000	1001001
Sign 1 bit	**Exponent** 8 bits	**Fraction** 7 bits

Quantization: INT8



Let's store Pi: 3.141592

FP32 **4 bytes memory**

0	10000000	10010010000111111011000
Sign 1 bit	**Exponent** 8 bits	**Fraction** 23 bits

INT8 **1 byte memory**

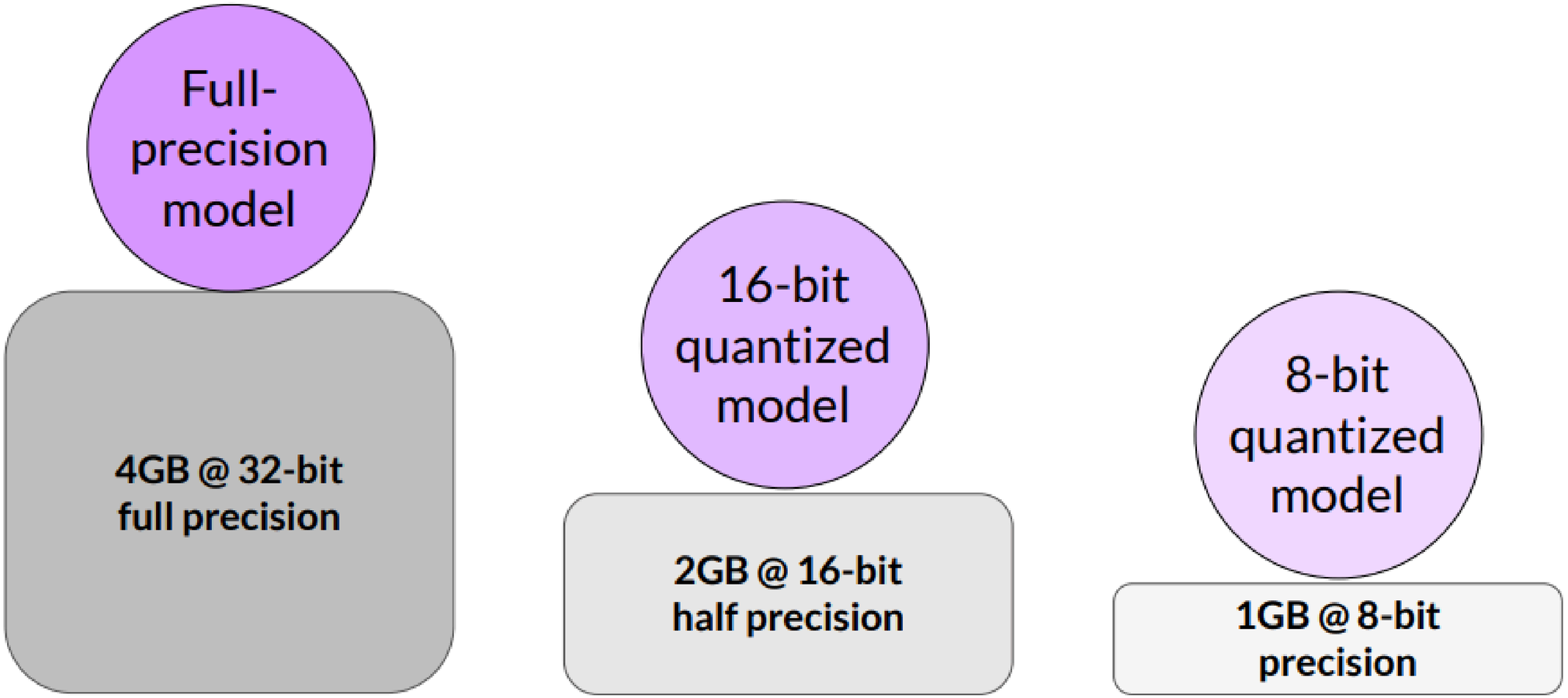
0	0000011
Sign 1 bit	Fraction 7 bits

Quantization: Summary

	Bits	Exponent	Fraction	Memory needed to store one value
FP32	32	8	23	4 bytes
FP16	16	5	10	2 bytes
BFLOAT16	16	8	7	2 bytes
INT8	8	–/–	7	1 byte

- Reduce required memory to store and train models
- Projects original 32-bit floating point numbers into lower precision spaces
- Quantization-aware training (QAT) learns the quantization scaling factors during training
- BFLOAT16 is a popular choice

Approximate GPU RAM needed to store 1B parameters



Sources: https://huggingface.co/docs/transformers/v4.20.1/en/perf_train_gpu_one#anatomy-of-models-memory, https://github.com/facebookresearch/bitsandbytes

GPU RAM needed to train larger models

1B param model

175B param model

4,200 GB @ 32-bit full precision

500B param model

12,000 GB @ 32-bit full precision