大模型系列 – 大模型算法

Transformer介绍

ZOMI

大模型业务全流程

数据 & 模型算法

2. 数据处理

- 开源数据
- 数据预处理
- 向量数据库

3. 模型算法

- LLM模型架构
- 多模态一切皆Tokens

模型训练 & 微调

4. 模型训练

- 混合精度
- 梯度检查
- 梯度累积
- …
- 5. 分布式并行
- 训练集群稳定性

6. 模型微调

- 全参微调
- 低参微调
- 指令微调

模型验证 & 推理部署

7. 模型验证

- 下游任务
- 测评标准
- bench mark

8. 推理与智能体

- 量化压缩
- 推理加速
- 9. Agent 智能体

集群算力准备

1. AI 集群建设

- 计算、存储、网络
- AI 集群机房建设
- AI 集群上线与运维

关于本内容

- **具体内容**
  1. 大模型都是什么模型结构？
  2. 为什么出现 Transformer？
  3. Transformer 结构介绍？
  4. Bert 和 GPT 大模型出现

1. LLM

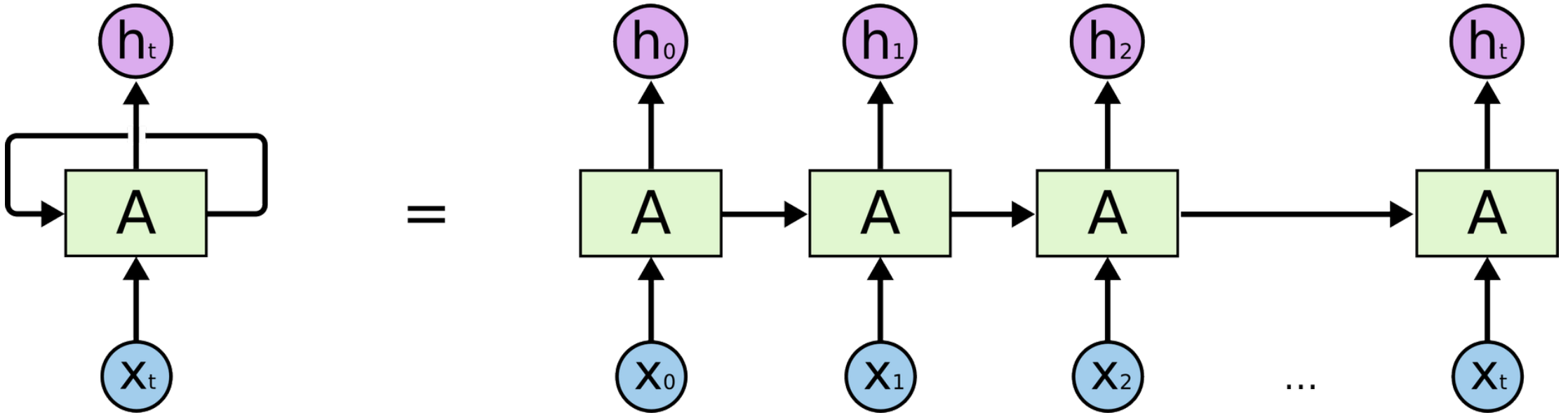
模型结构演进

- GPT3
- LLAMA1
- LLAMA2
- Baichuan1
- Baichuan2
- Qwen
- Mistral MOE

2. Motivation for Transformers

Problems with RNNs = Motivation for Transformers

- RNN循环神经网络是一个按时间序列的前馈神经网络模型，经过训练后可以处理顺序数据输入并将其转换为特定的顺序数据输出。



RNN has three major disadvantages

Slow to Train

训练慢

Short Memory

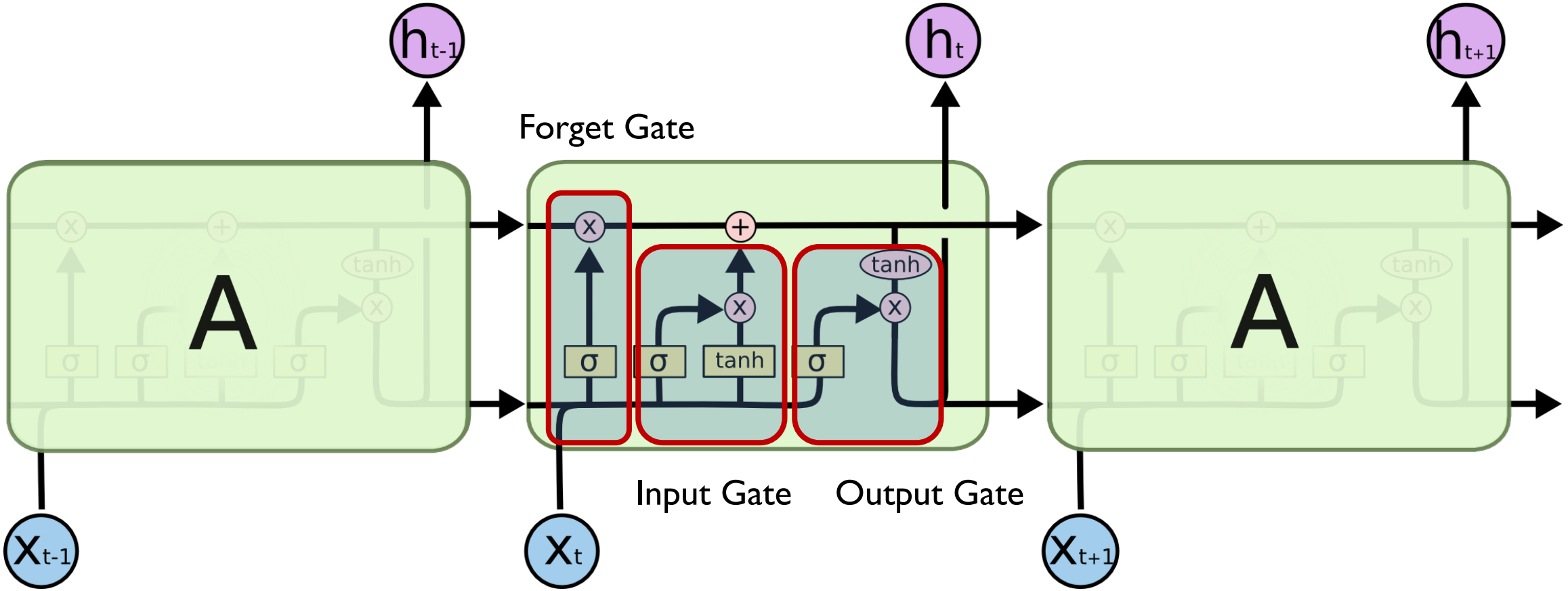
短期记忆

Vanishing Gradient

梯度消失/梯度爆炸

LSTM

- long short-term memory，LSTM 是一种特殊 RNN，主要是为了解决长序列训练过程中的梯度消失和梯度爆炸问题。



LSTM has three major disadvantages

Slower to Train

训练更慢

Short Memory

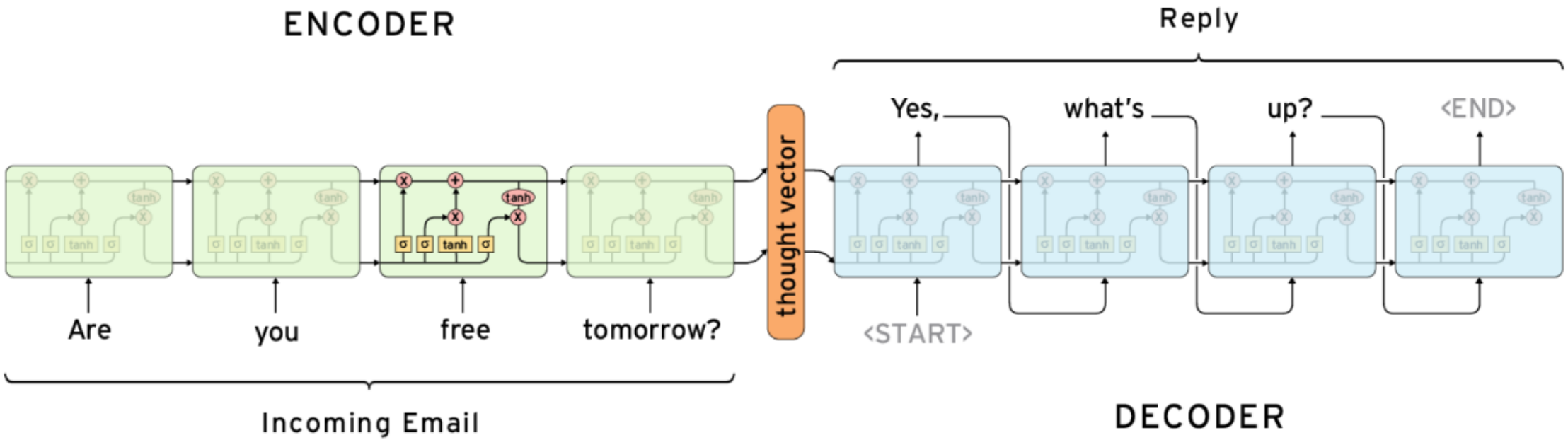
短期记忆

Vanishing Gradient

梯度消失/梯度爆炸

Encoder-Decoder Structure

- Seq2Seq（Sequence-to-Sequence，序列到序列）是一类特殊序列建模问题；
- Encoder-Decoder 模型包括编码器和解码器，编码器（Encoder）先对输入的序列进行处理，将处理后向量发送给解码器（Decoder），转化成想要的输出。

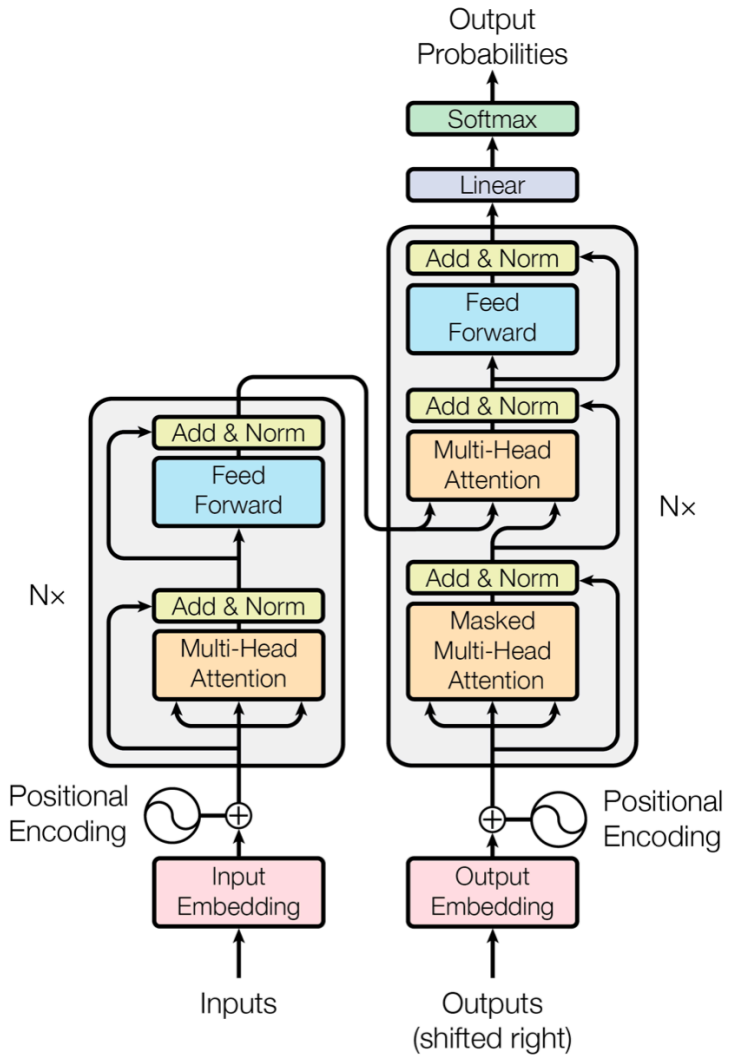


3. Transformer and Attention

"Original" Transformer (Vaswani et al., 2017).

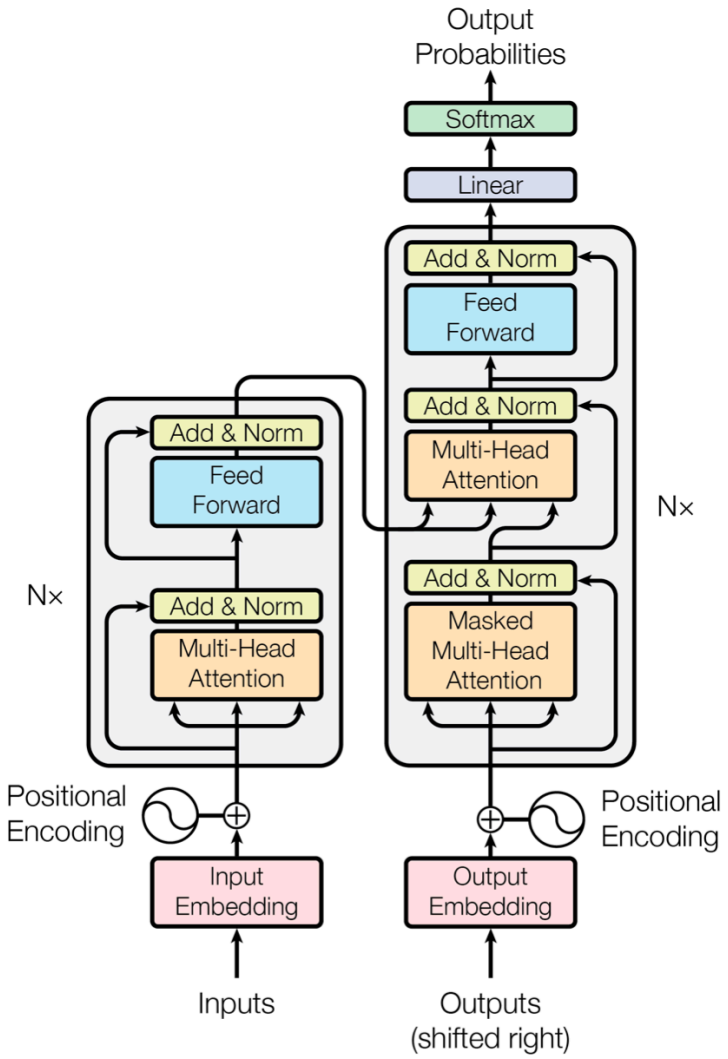
- The architecture is tested on English-to-German and English-to-French translation using the standard WMT2014 datasets.
  - English-to-German: 4.5M sentence pairs, 37k tokens vocabulary.
  - English-to-French: 36M sentence pairs, 32k tokens vocabulary.

- 8 P100 GPUs (150 TFlops FP16), 0.5 day for the small model, 3.5 days for the large one.



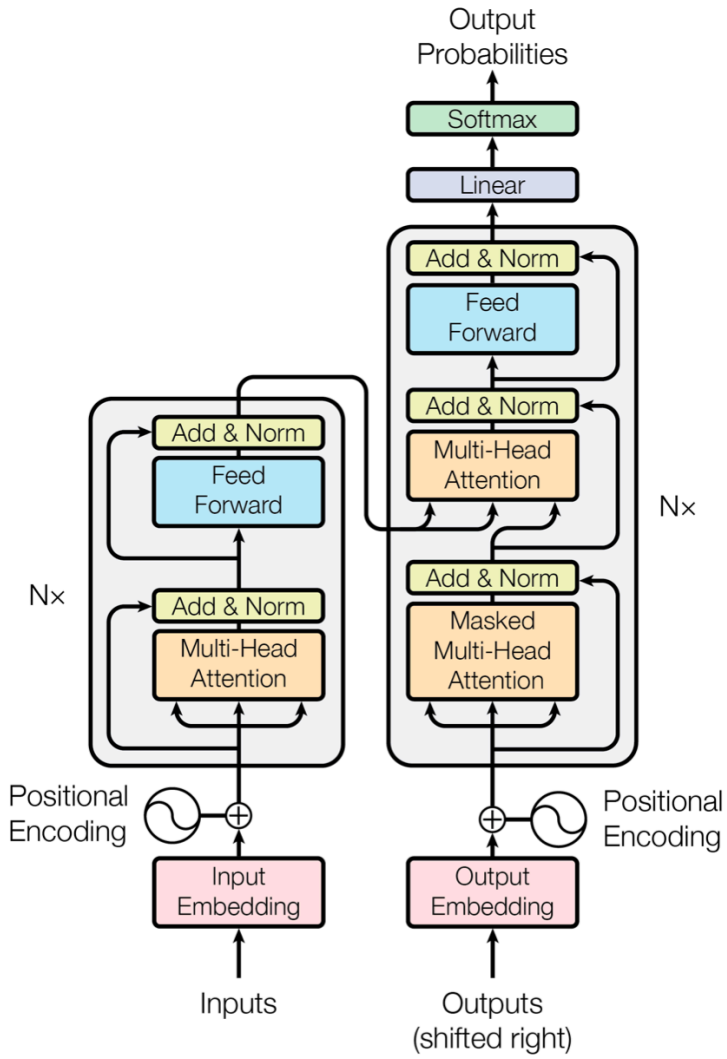
"Original" Transformer (Vaswani et al., 2017).

- Tokenization
- Input Embeddings
- Position Encodings
- Residuals
- Query
- Key
- Value
- Add & Norm
- Encoder
- Decoder
- Attention
- Self Attention
- Multi Head Attention
- Masked Attention
- Encoder Decoder Attention
- Output Probabilities / Logits
- Softmax
- Decoder only models

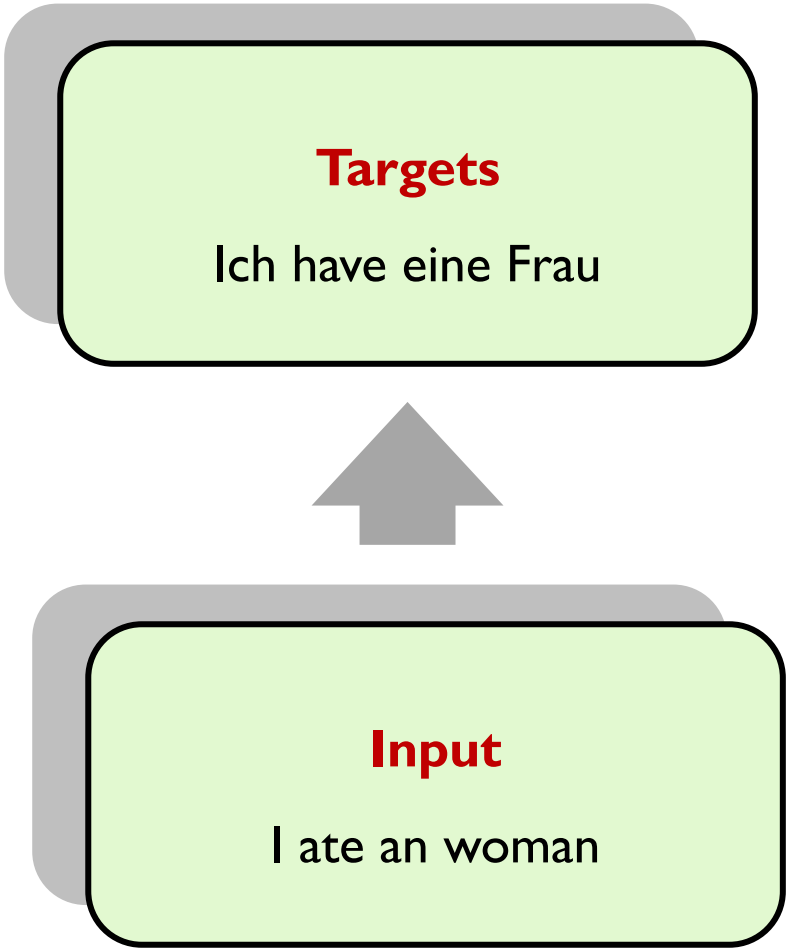


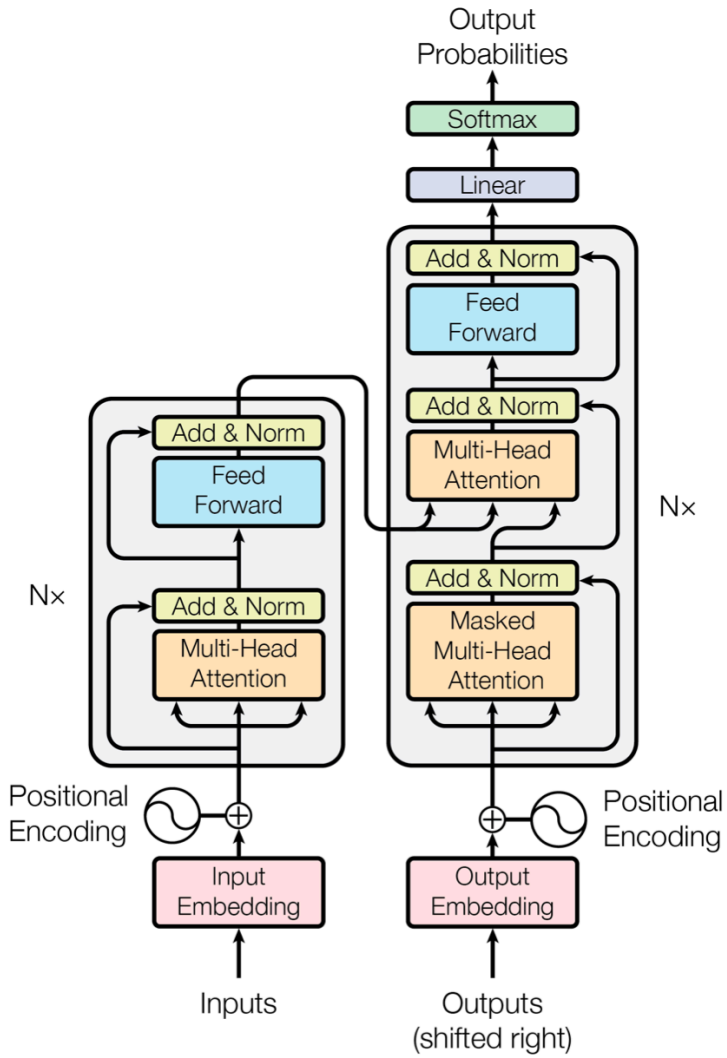
"Original" Transformer (Vaswani et al., 2017).

- Tokenization
- Input Embeddings
- Position Encodings
- Residuals
- Query
- Key
- Value
- Add & Norm
- Encoder
- Decoder
- Attention
- Self Attention
- Multi Head Attention
- Masked Attention
- Encoder Decoder Attention
- Output Probabilities / Logits
- Softmax
- Decoder only models



任务：机器翻译 Machine Translation



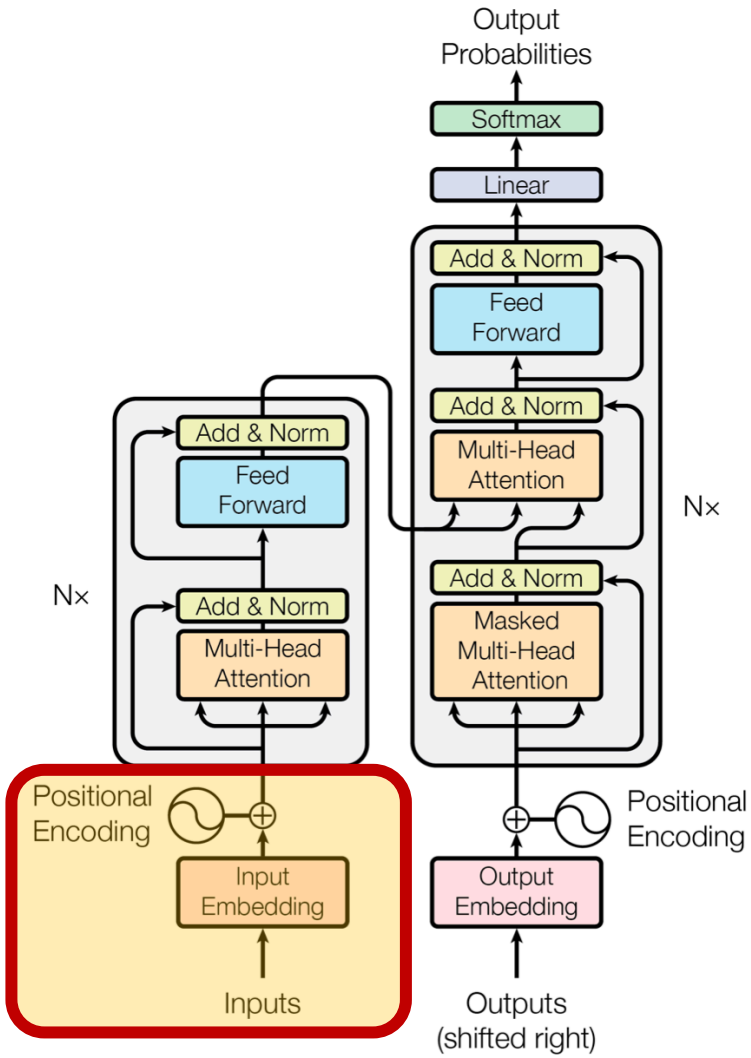


Transformer : Inputs

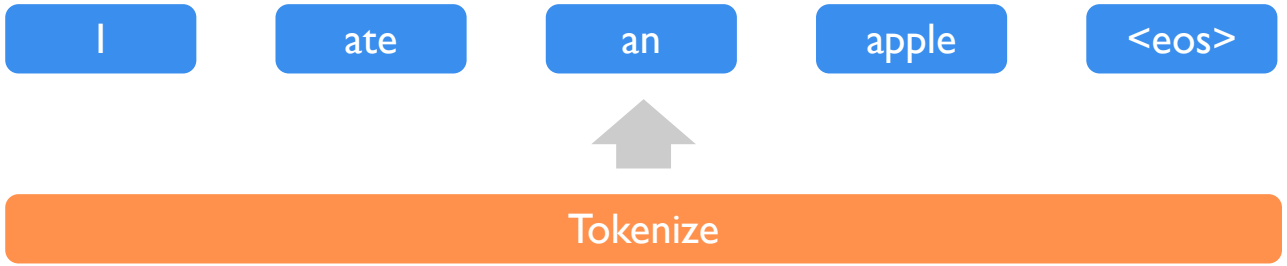
Processing Inputs

Input

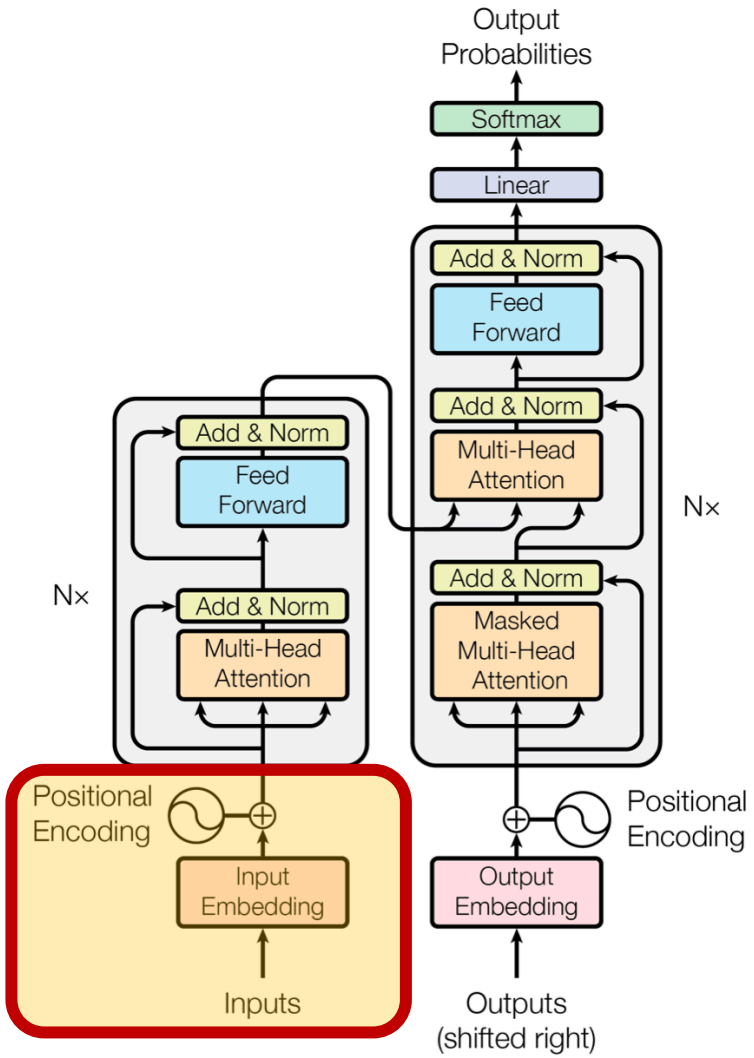
I ate an woman



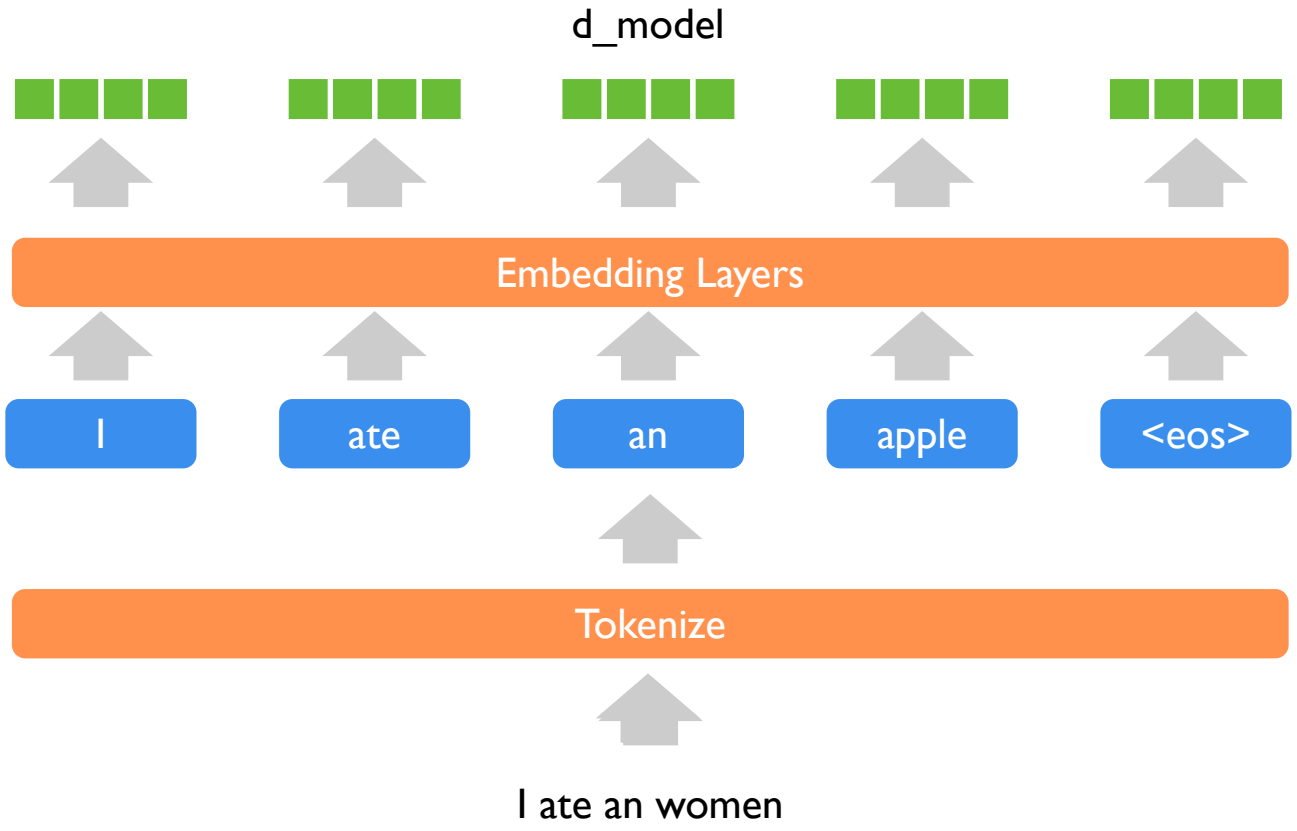
Transformer : Inputs



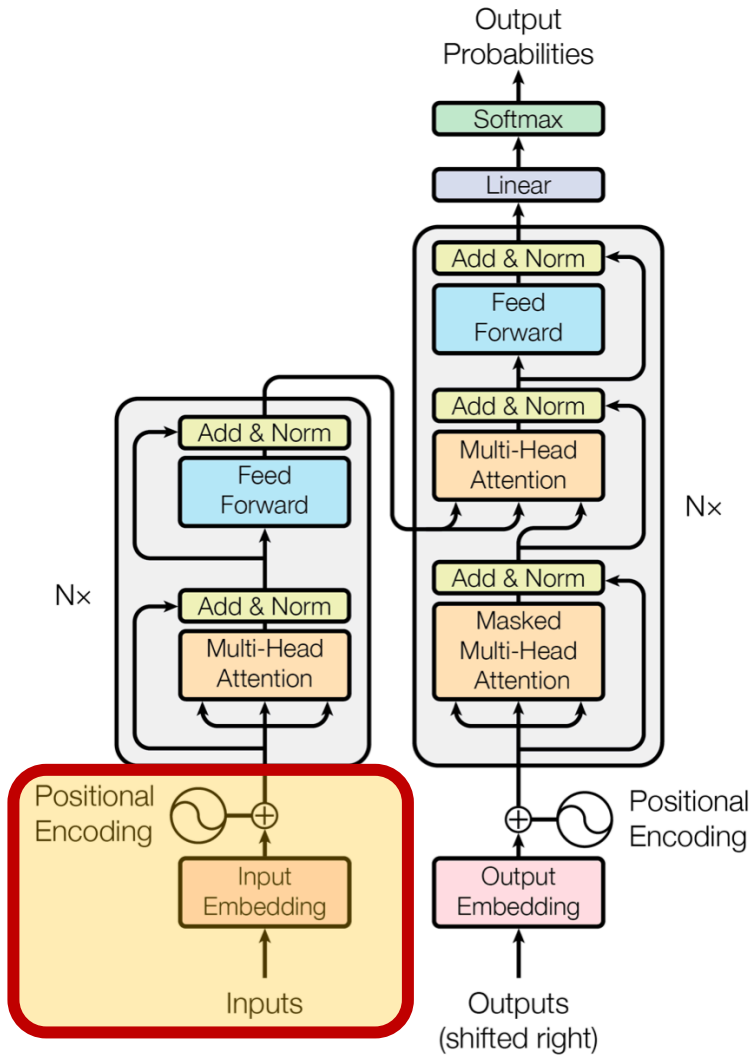
Generate Input Embeddings



Transformer : Inputs

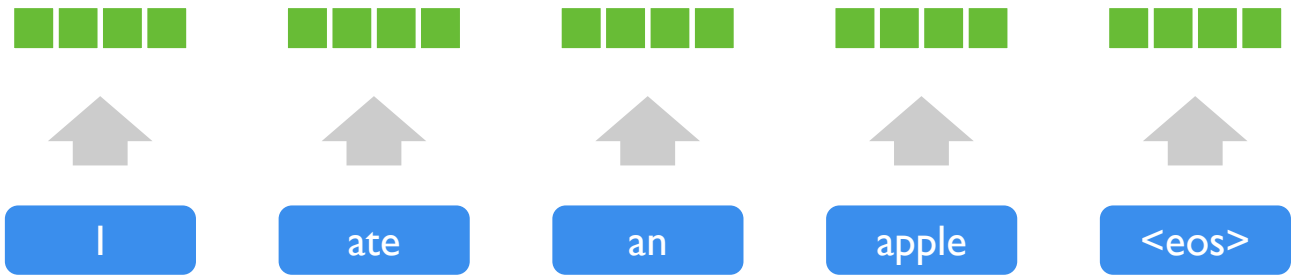


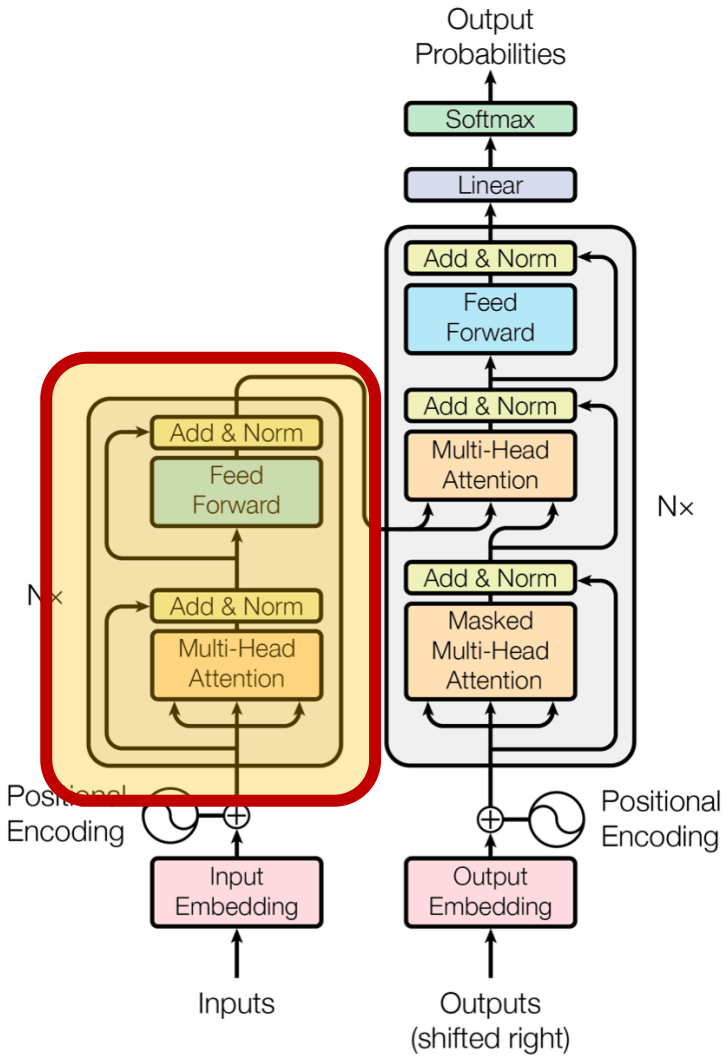
Generate Input Embeddings



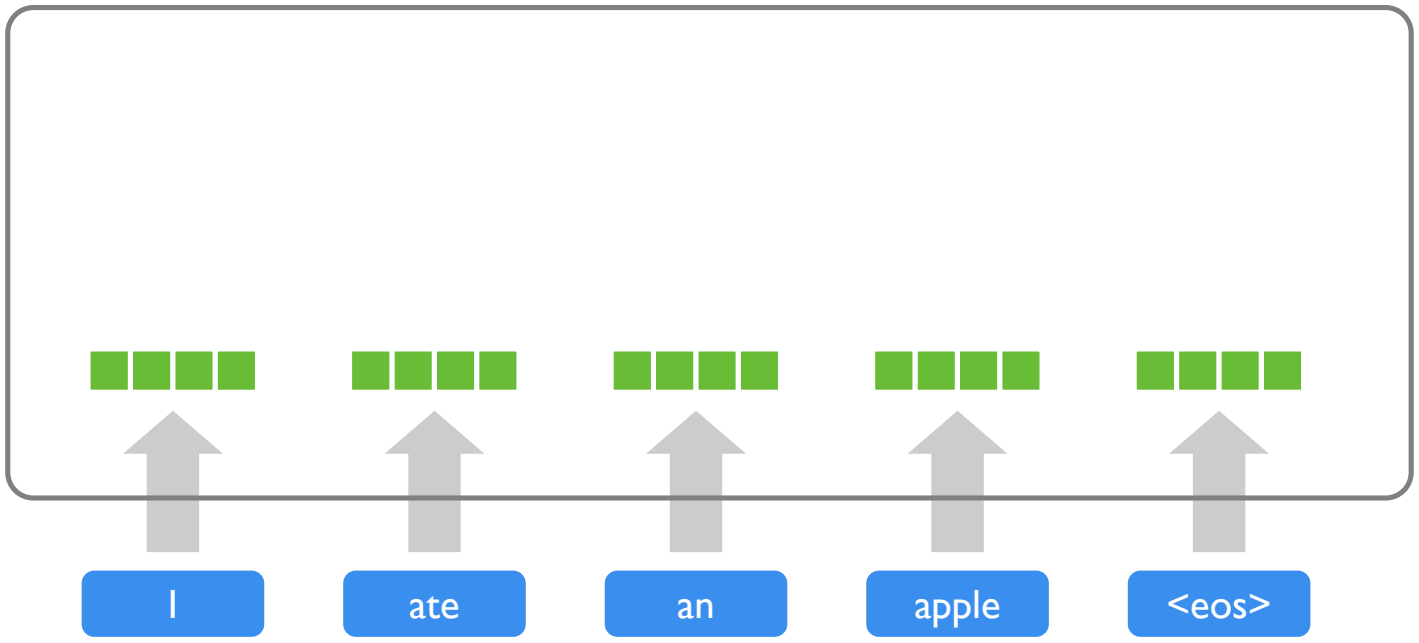
Transformer : Encoder

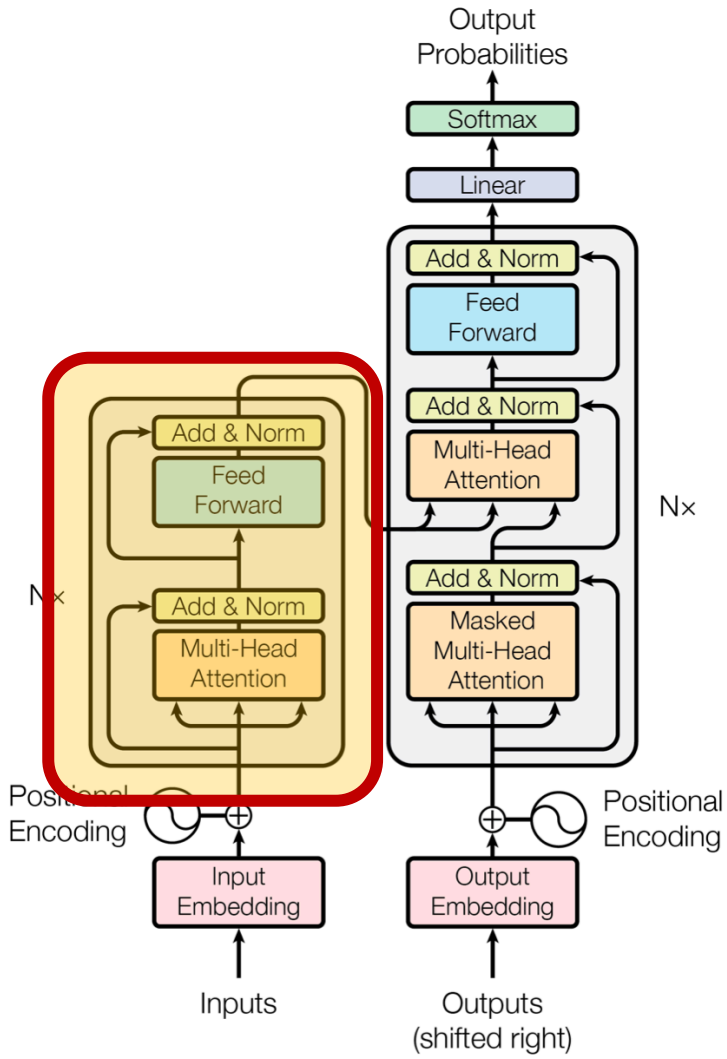
where is the content ?



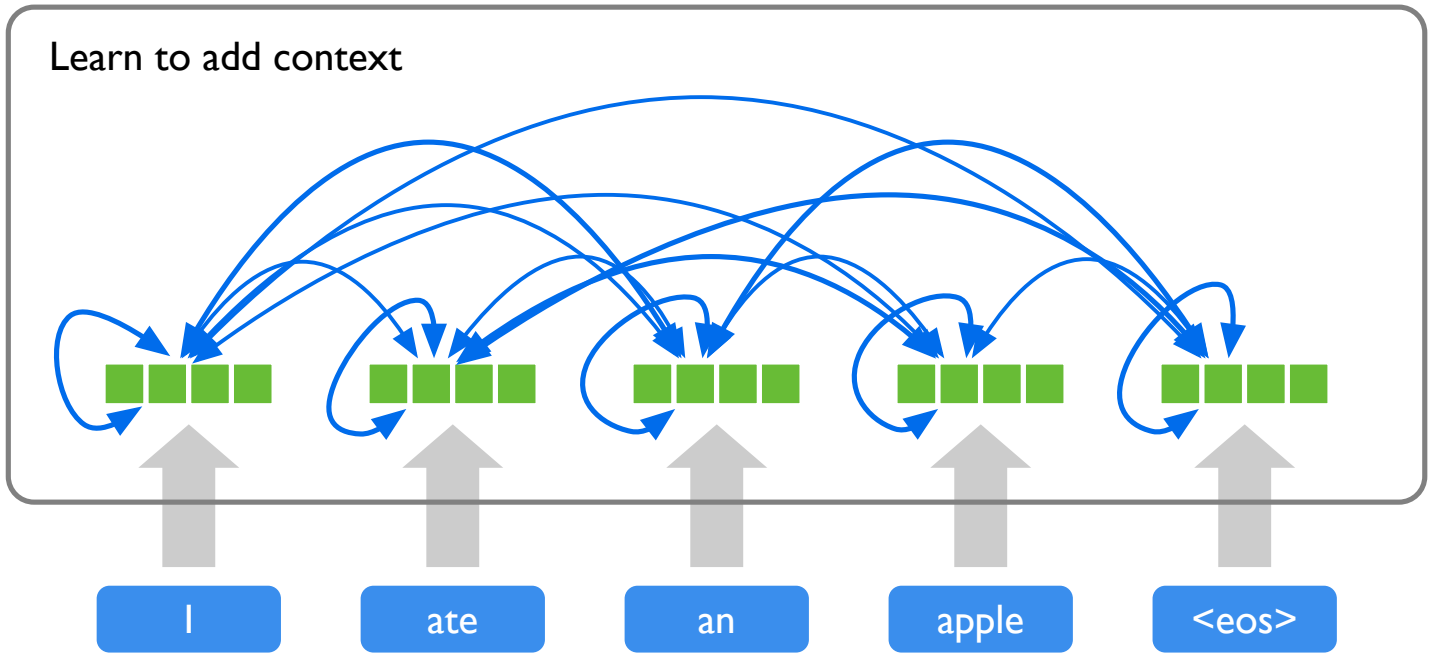


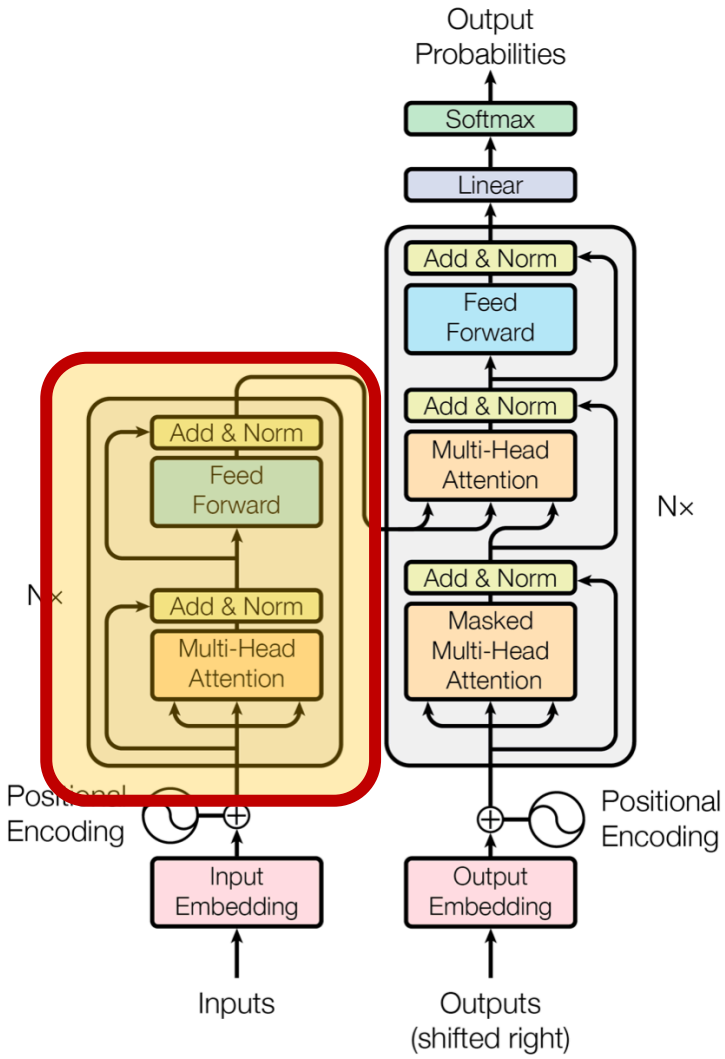
Transformer : Encoder



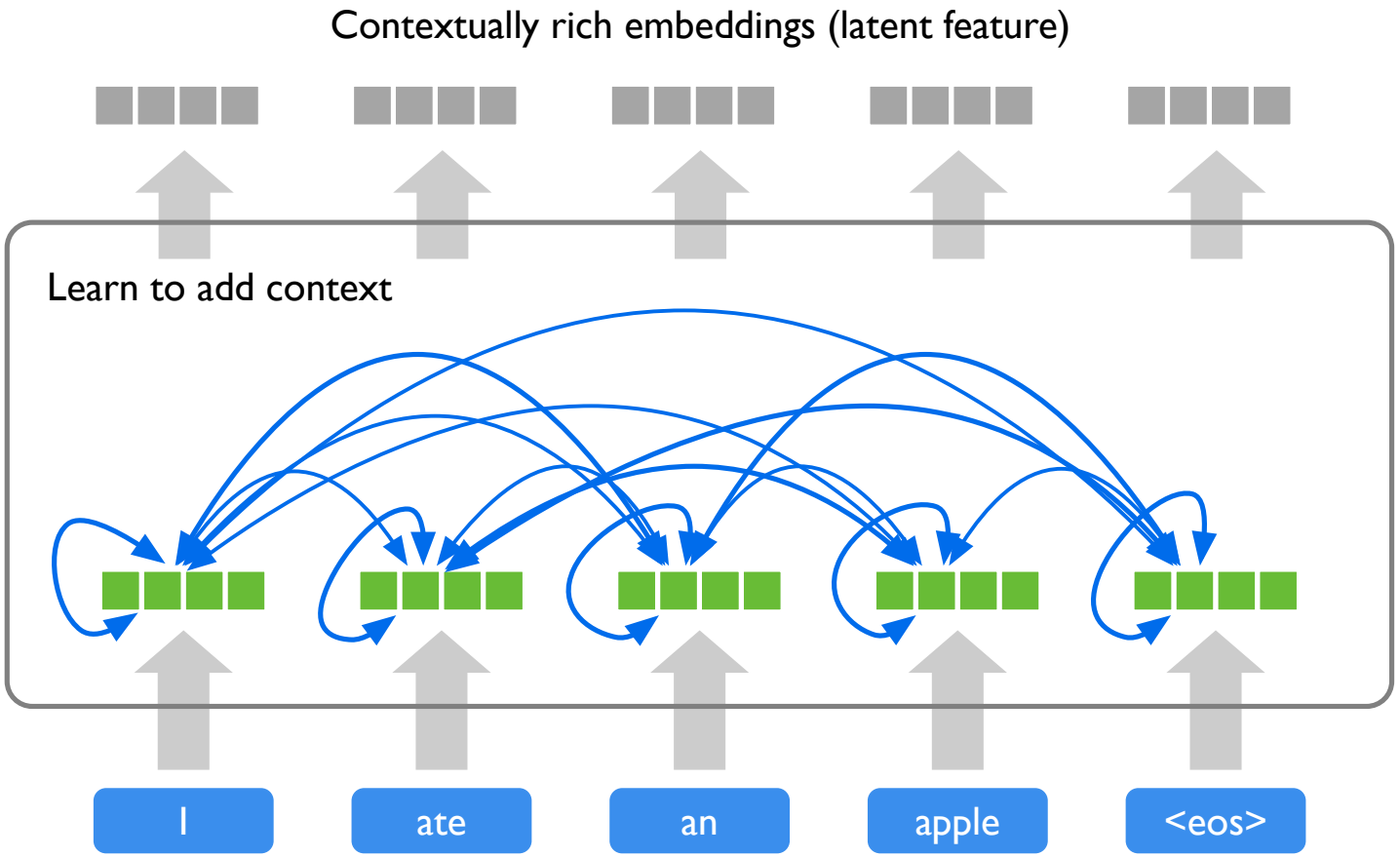


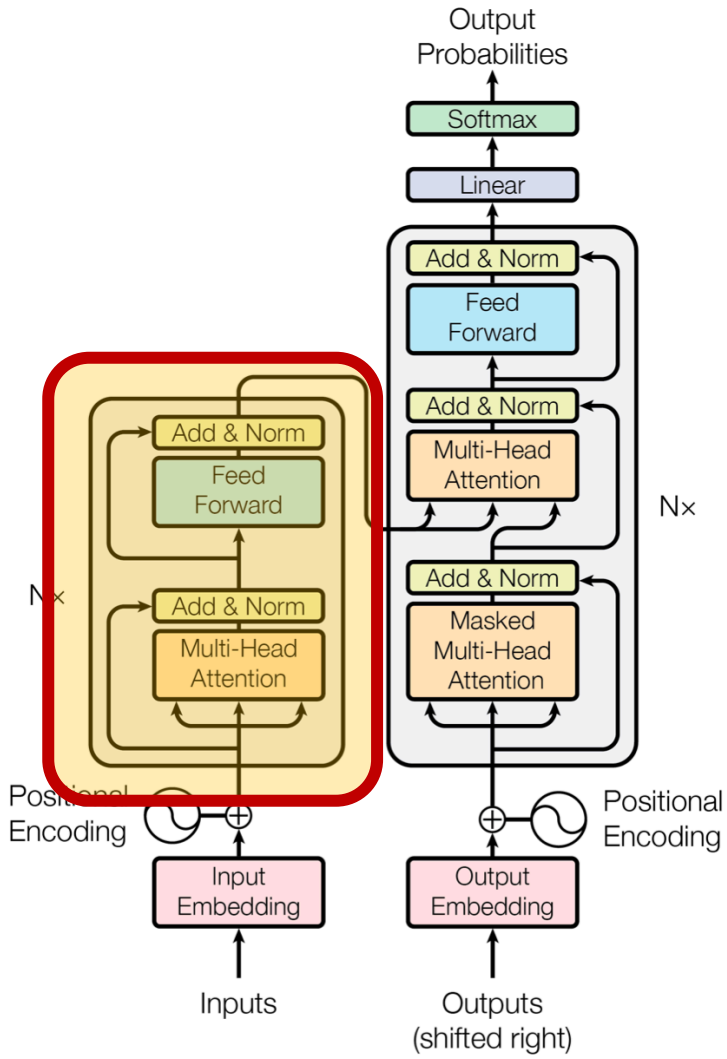
Transformer : Encoder





Transformer : Encoder

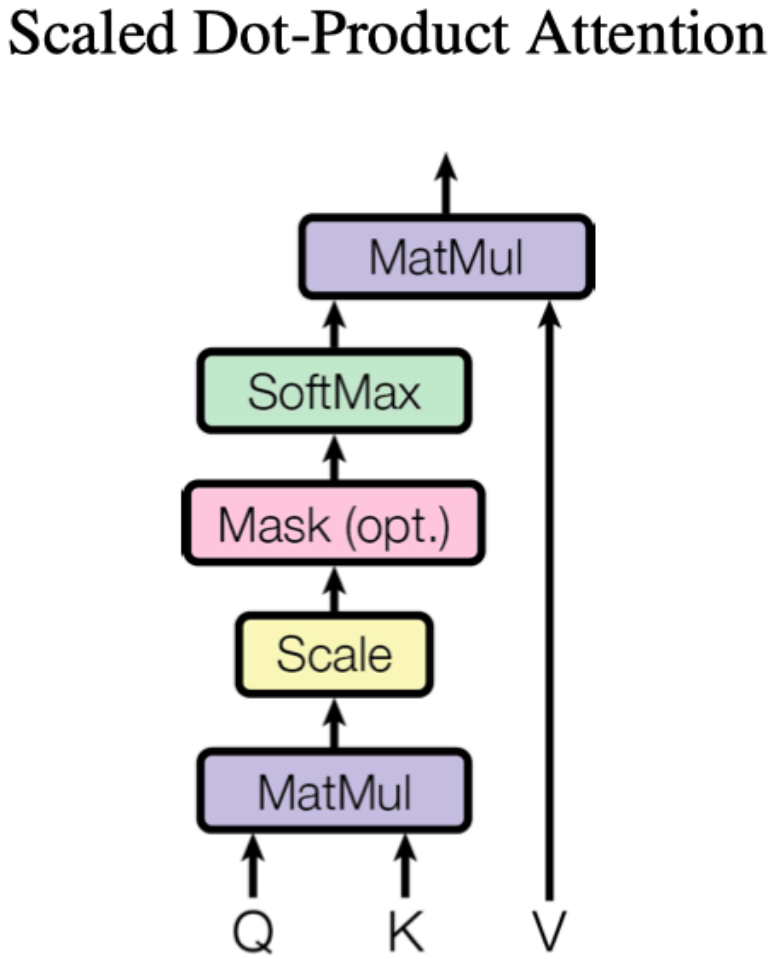




Attention and Multi-Head Attention

- 对当前 Query 和所有的 Key 计算相似度，将这个相似度值通过 Softmax 进行得到一组权重，根据这组权重与对应 Value 乘积求和得到 Attention 下 Value 值。

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

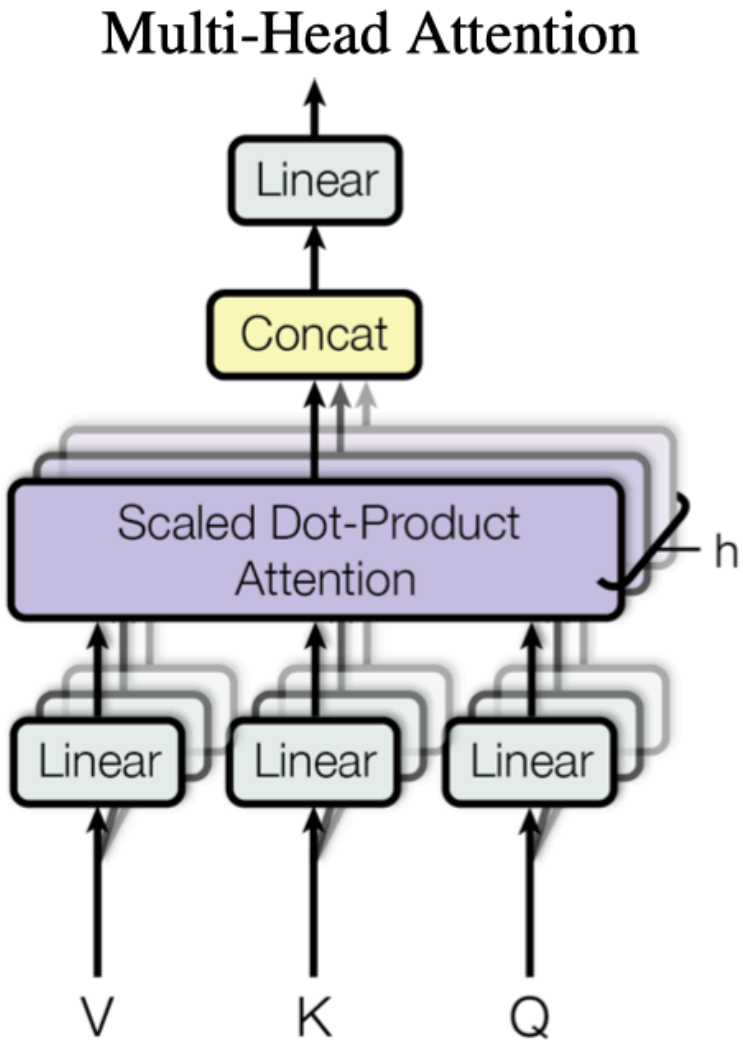


Attention and Multi-Head Attention

- Multi-head Attention其实就是多个Self-Attention结构的结合，每个head学习到在不同表示空间中的特征。

$$MultiHead(Q,K,V) = Concat(H_1, H_2, \cdots, H_h)W^0$$

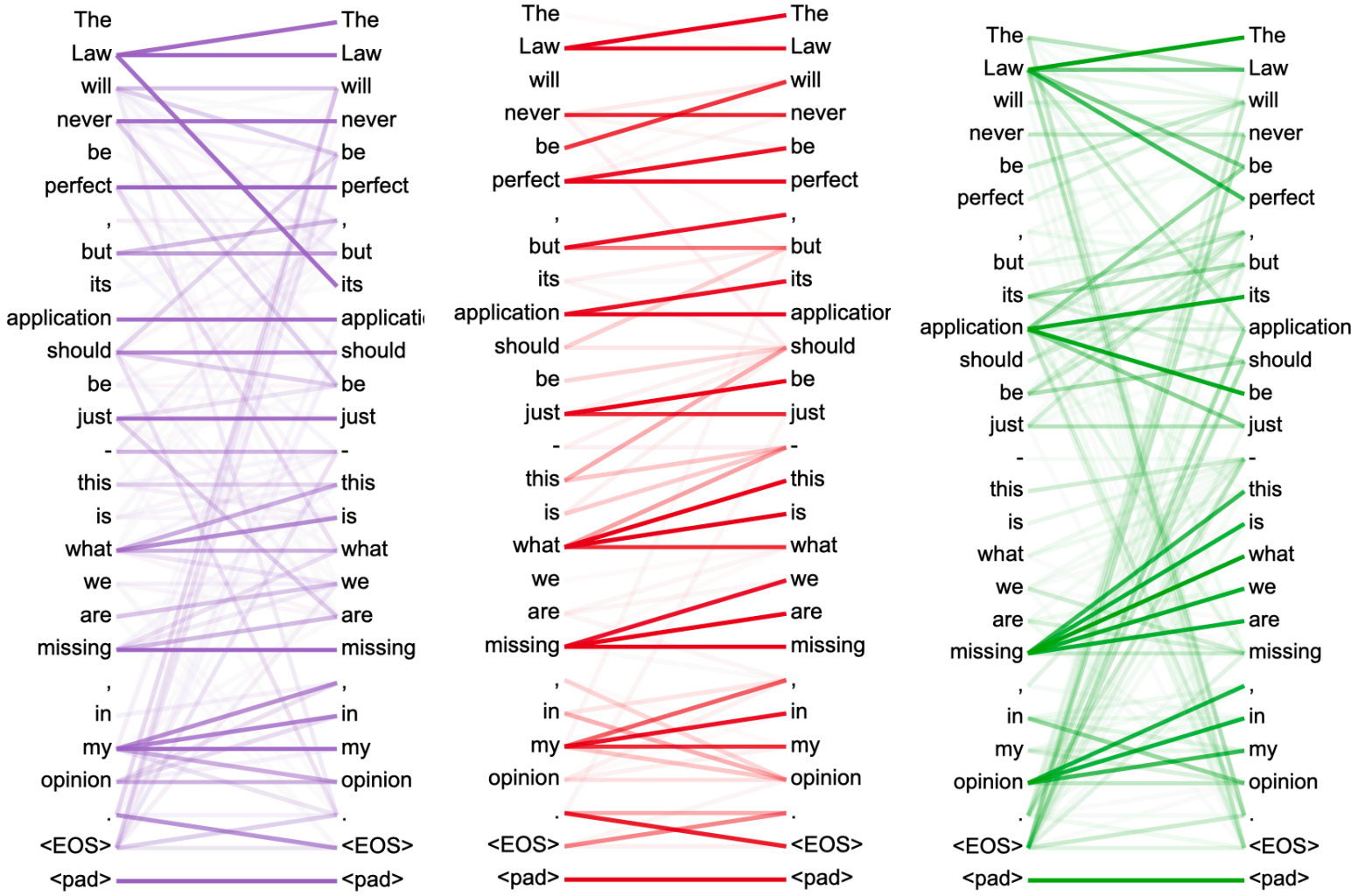
$$H_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right), i = 1, \cdots h$$



Attention 相关性可视化

- 两个head学习到的Attention侧重点可能略有不同，这样给了模型更大的容量。

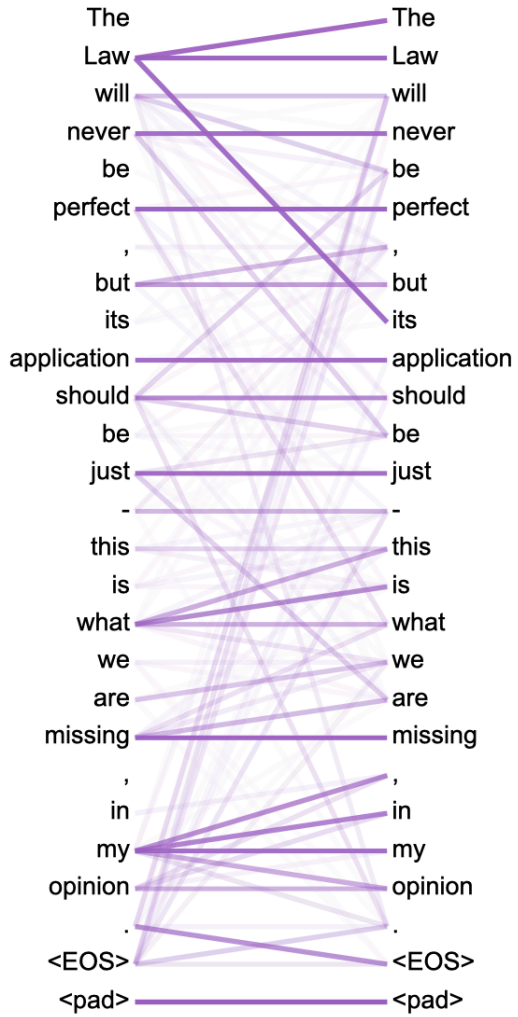
- visualization of the attention as computed by one head of the layer 5 of the encoder

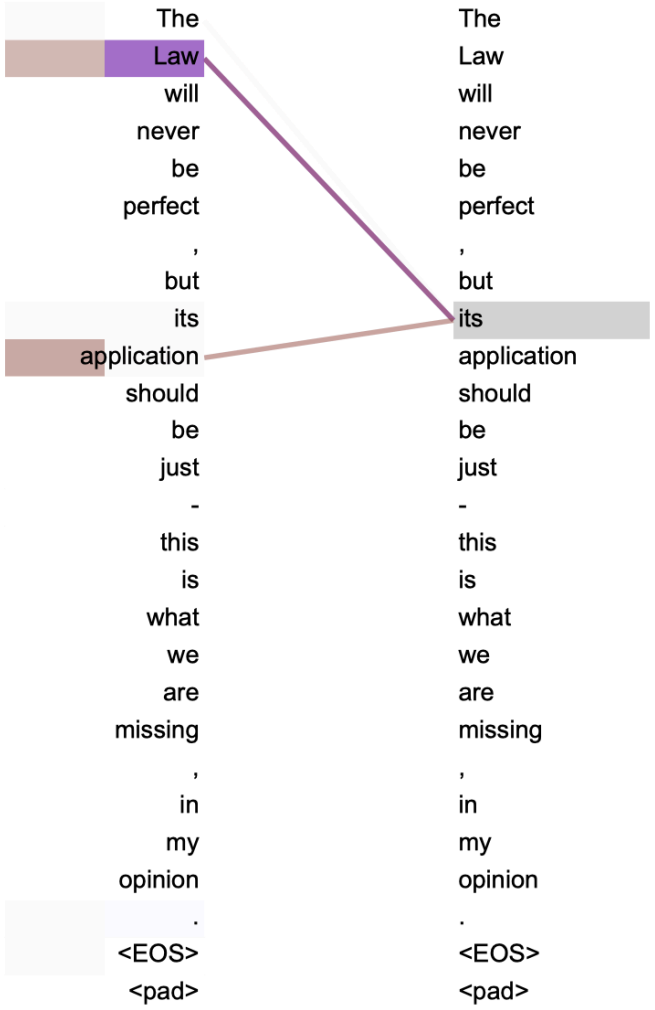


Attention 相关性可视化

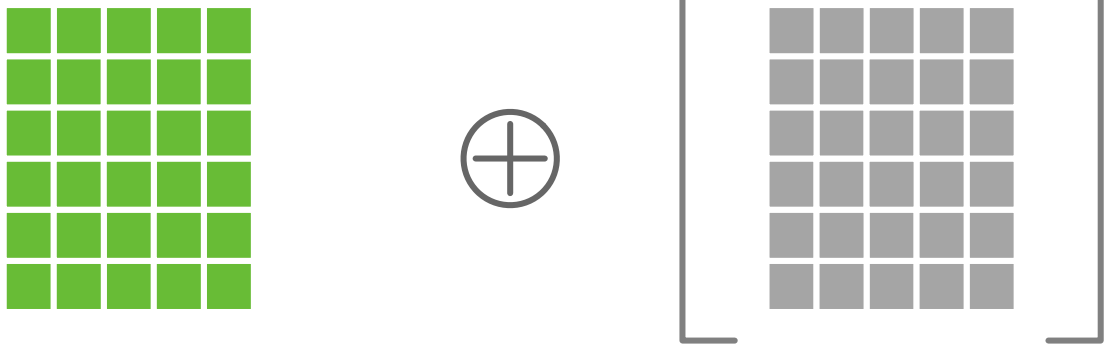
- left: Visualization of the attention as computed by one head of the layer 5 of the encoder

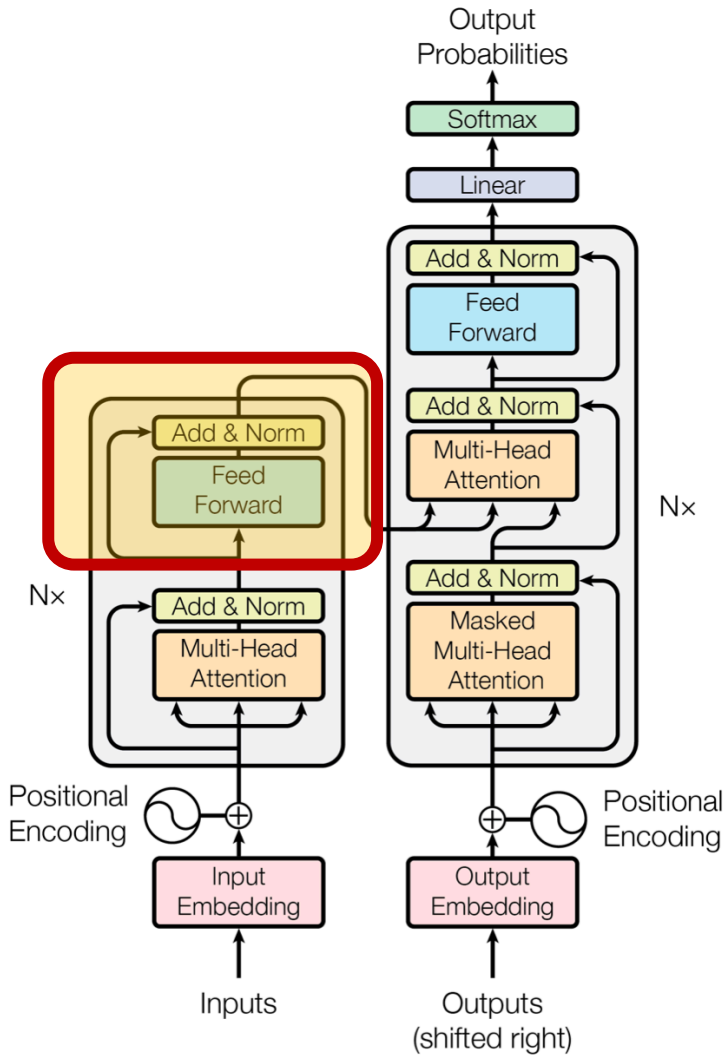
- right: Attention given by the word “its” for two different heads is on “l aw” and “ap- plication” which provi des help for gender and grammatica l issues.





Transformer: Add & Norm





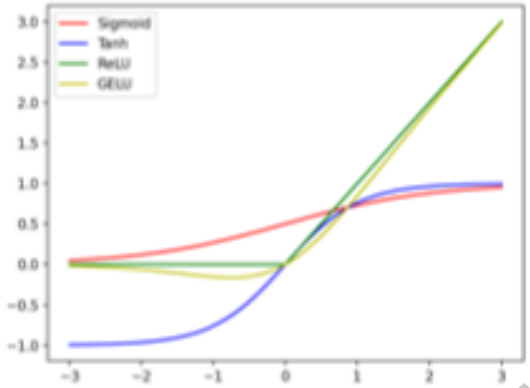
Normalization (Z)

- Mean 0, Std dev 1
- Stabilizes Training
- Regularization Effect

Add → Residuals

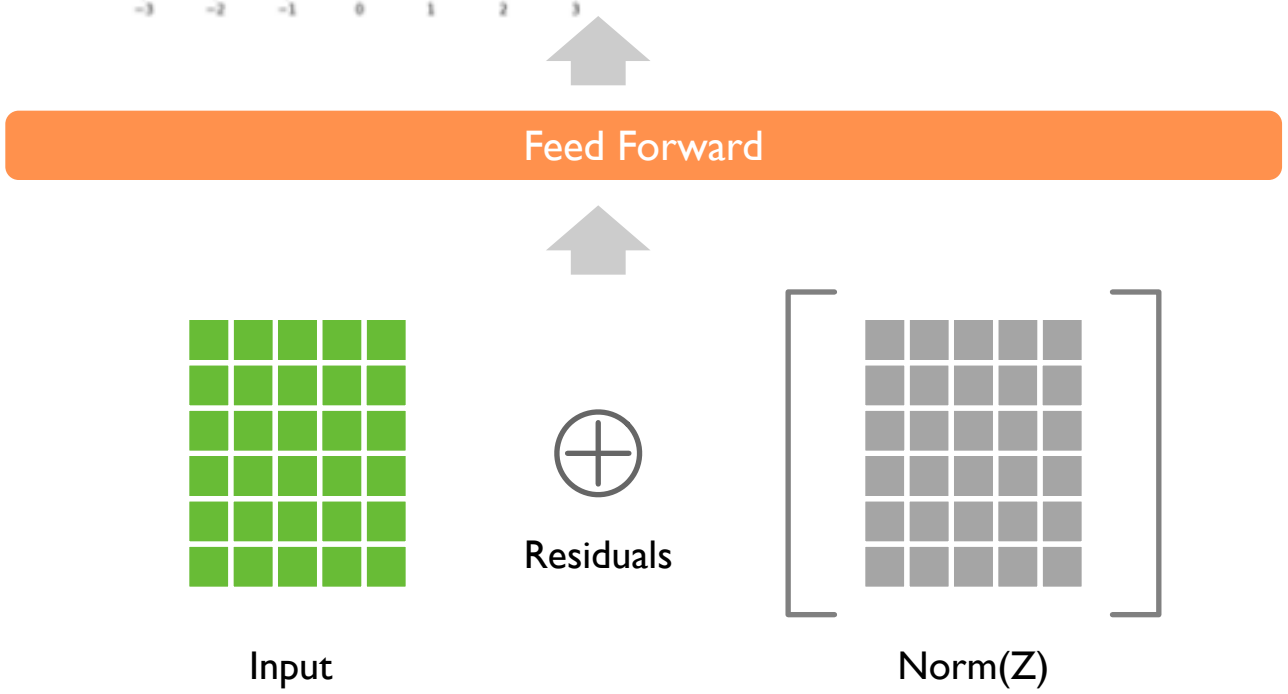
- Avoid vanishing gradients
- Train deeper networks

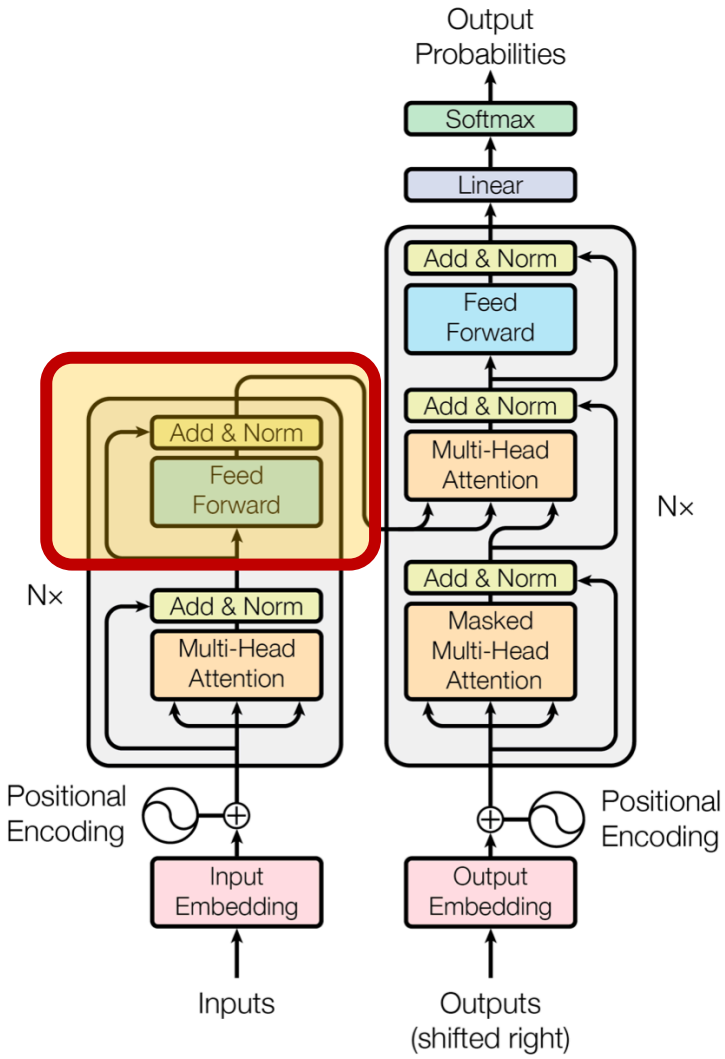
Transformer: Feed Forward



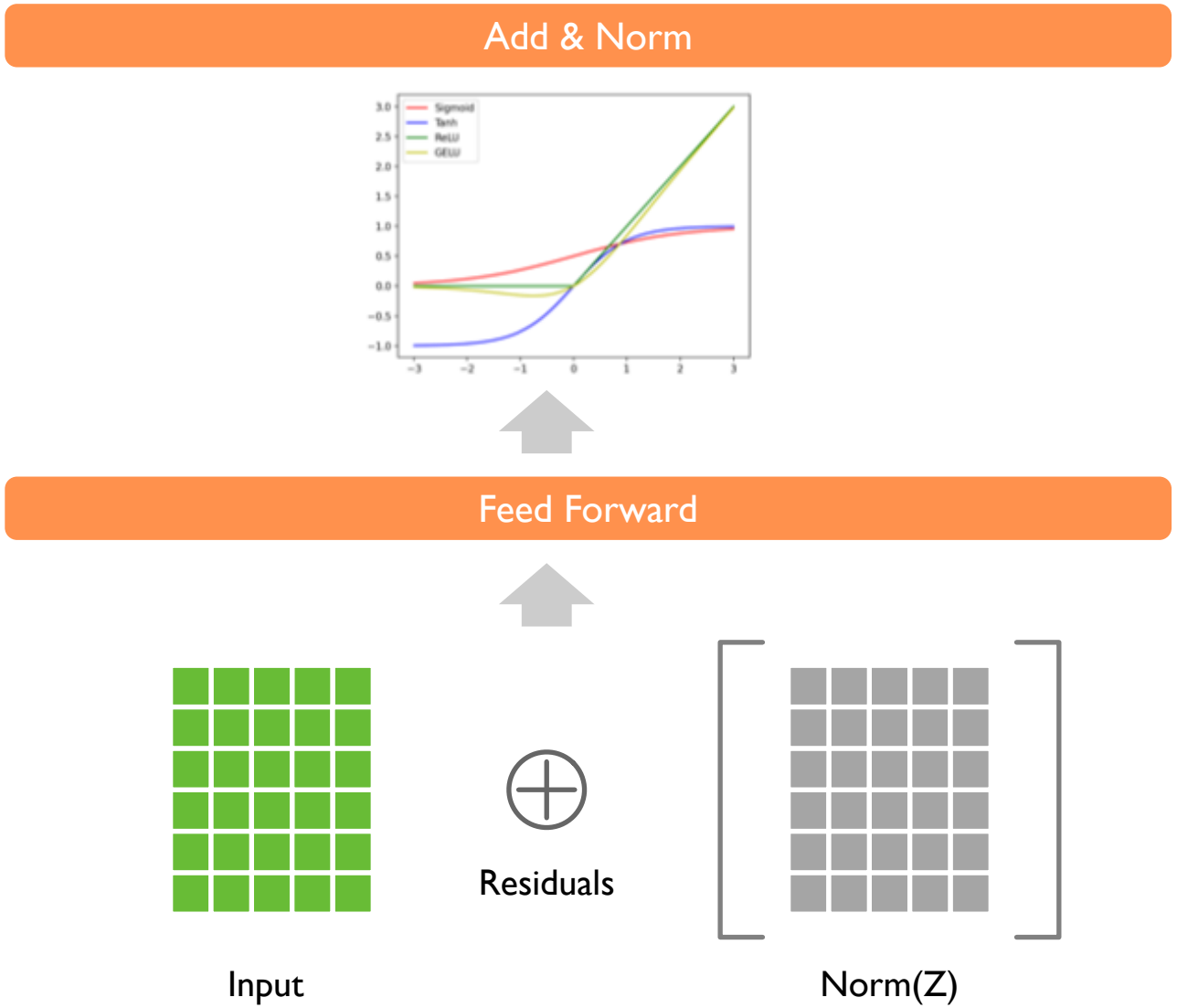
Feed Forward

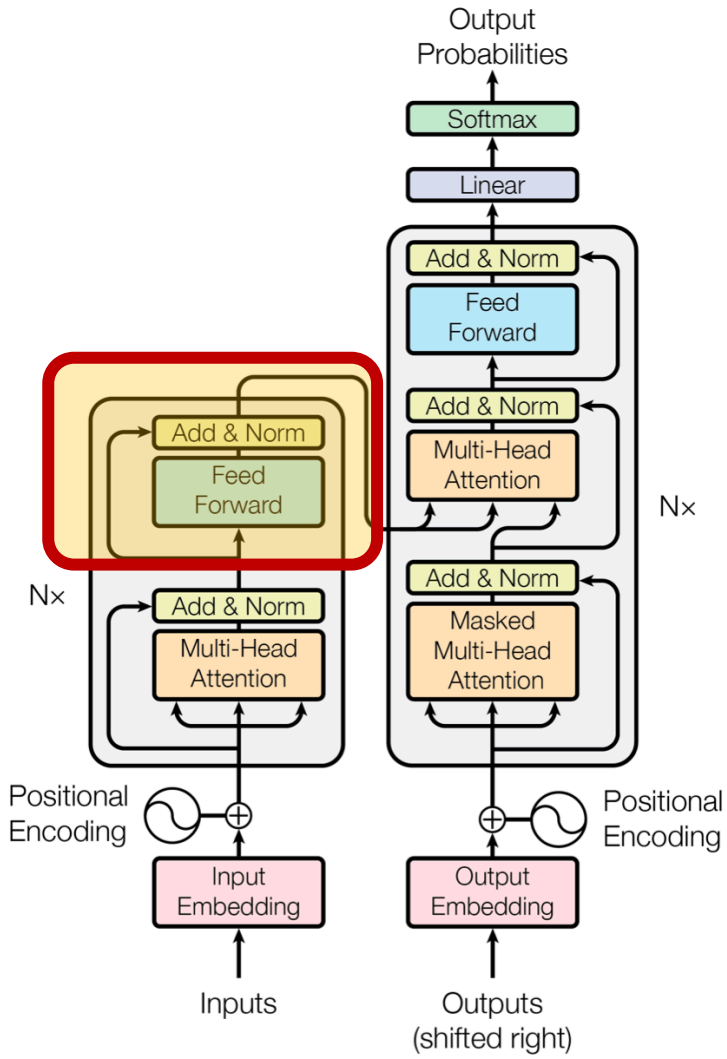
- Non Linearity
- Complex Relationships
- Learn form each other





Transformer: Add & Norm

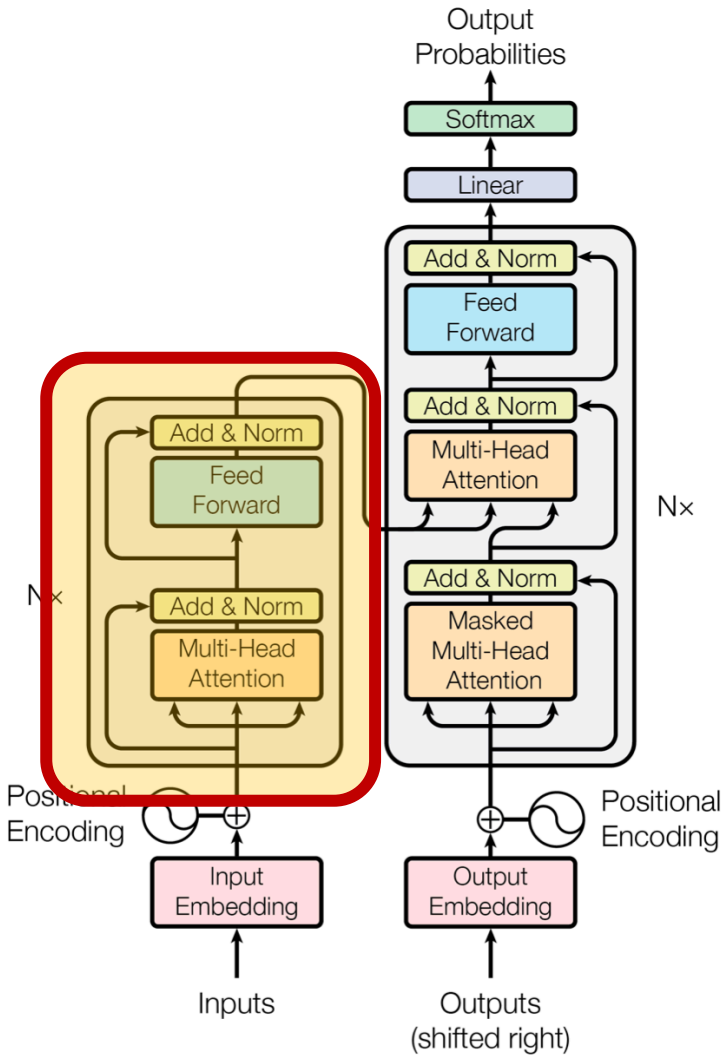




Transformer: Encoder

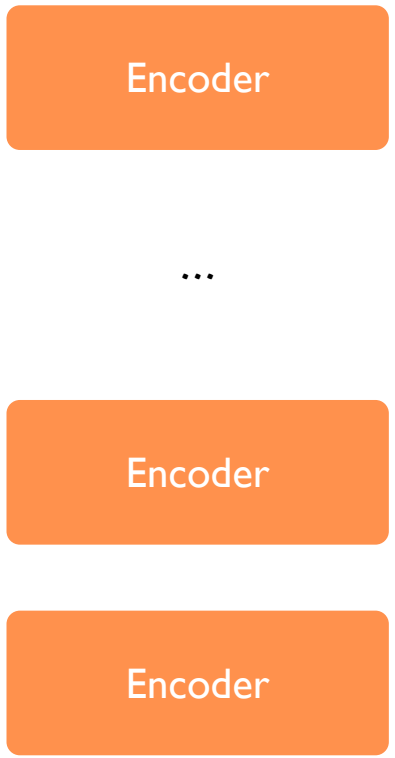
Encoder

Encoder



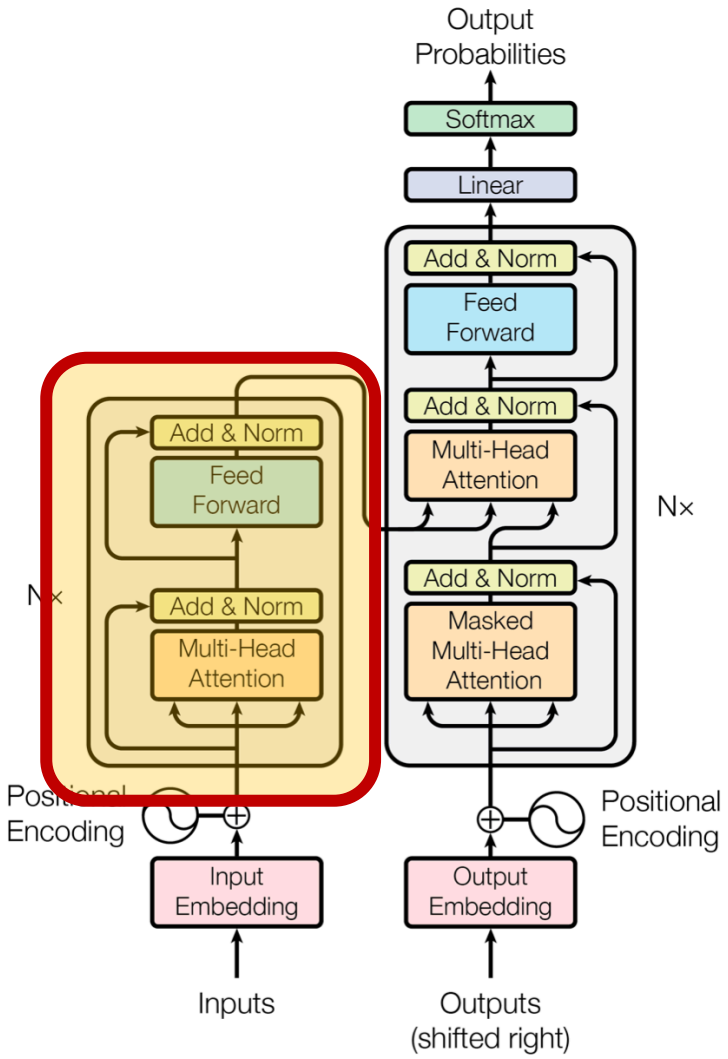
"Original" Transformer (Vaswani et al., 2017).

Encoder



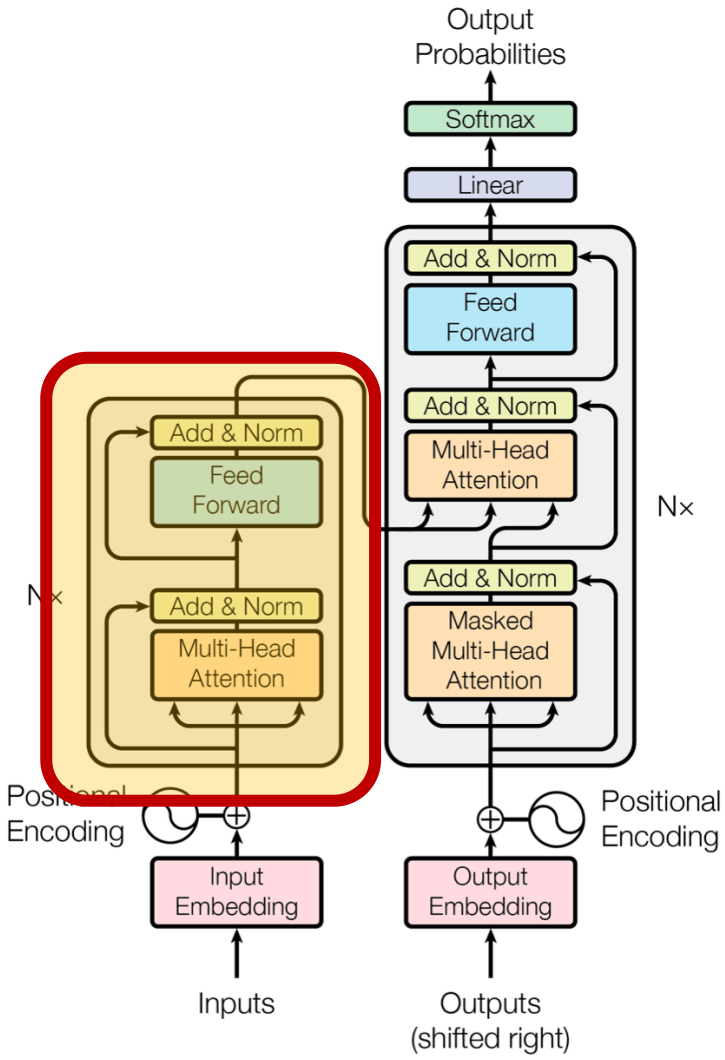
Input to Encoder_i+1

Output from Encoder_i



"Original" Transformer (Vaswani et al., 2017).

- ✓ Tokenization
- ✓ Input Embeddings
- ✓ Position Encodings
- ✓ Residuals
- ✓ Query
- ✓ Key
- ✓ Value
- ✓ Add & Norm
- ✓ Encoder
- ✓ Decoder
- ✓ Attention
- ✓ Self Attention
- ✓ Multi Head Attention
- Masked Attention
- Encoder Decoder Attention
- Output Probabilities / Logits
- Softmax
- Decoder only models



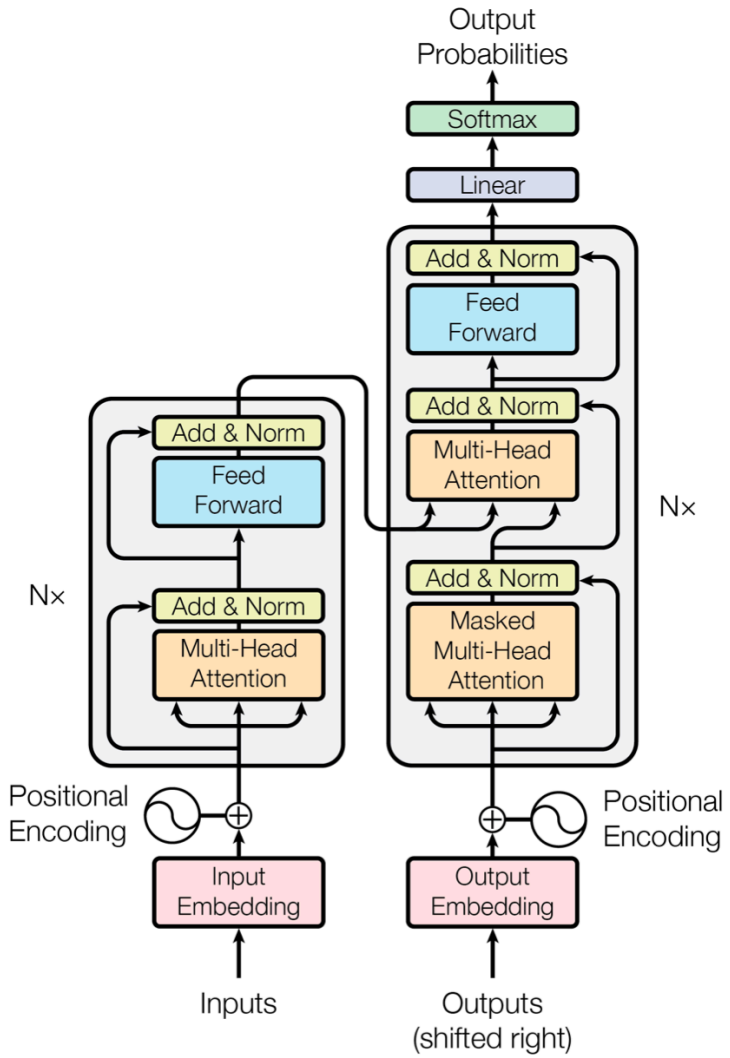
任务：机器翻译 Machine Translation

Targets

Ich have eine Frau

Input

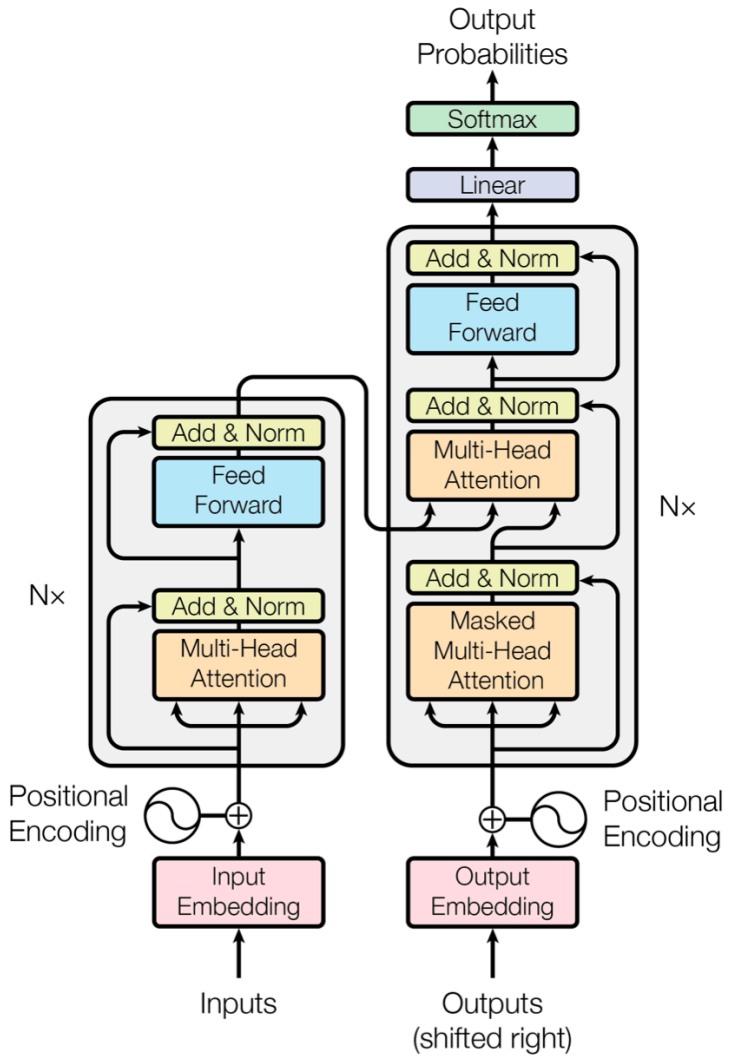
I ate an woman



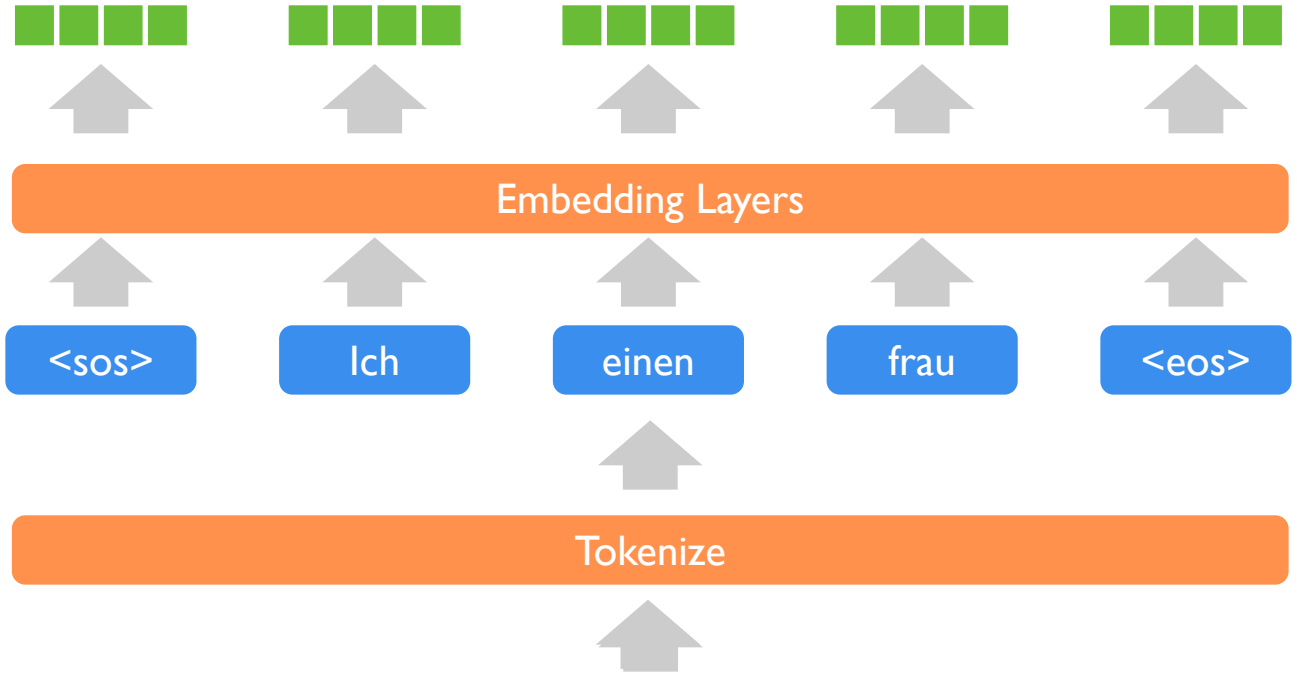
Transformer: Target

Targets

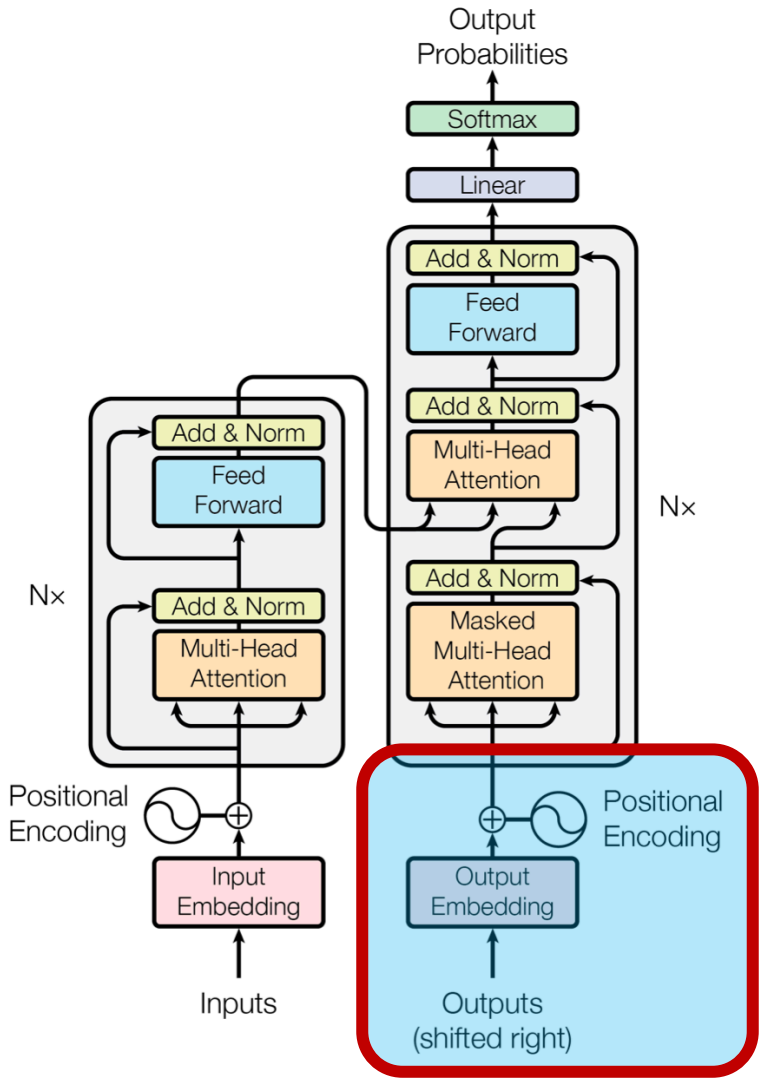
Ich have eine Frau



Transformer: Target

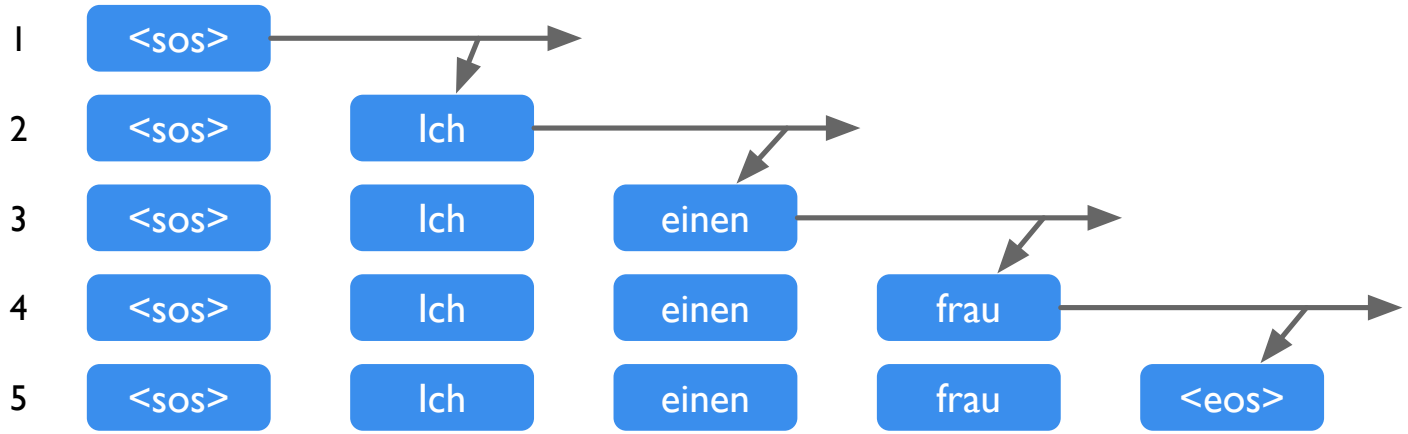


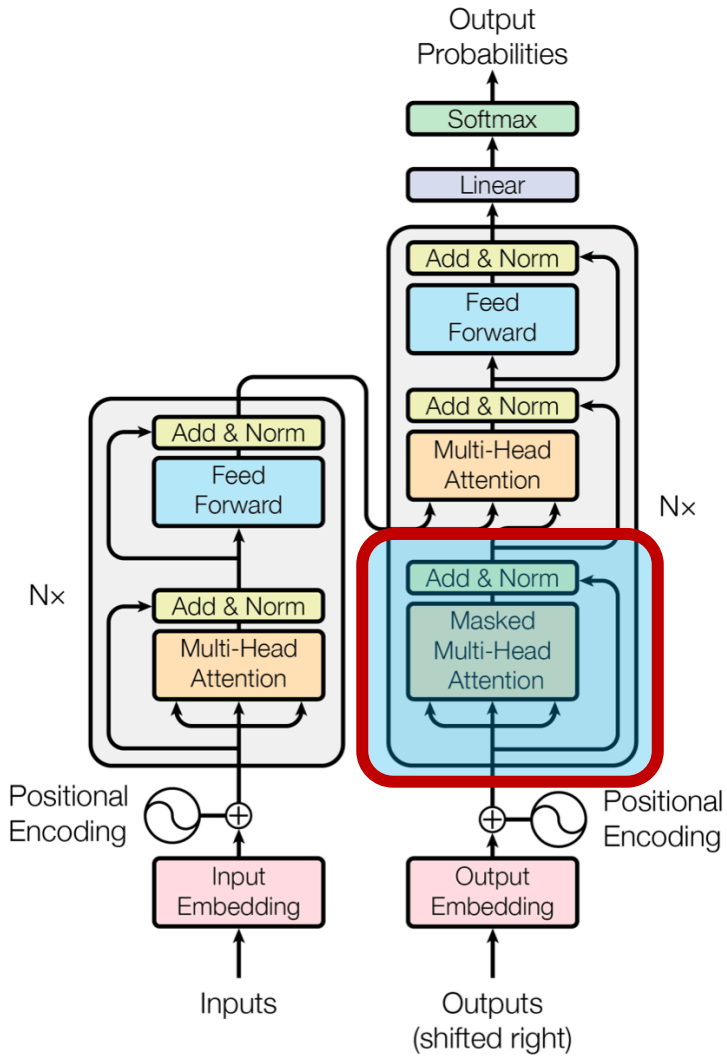
Generate Input Embeddings



Transformer: Masked Multi Head Attention

Inference





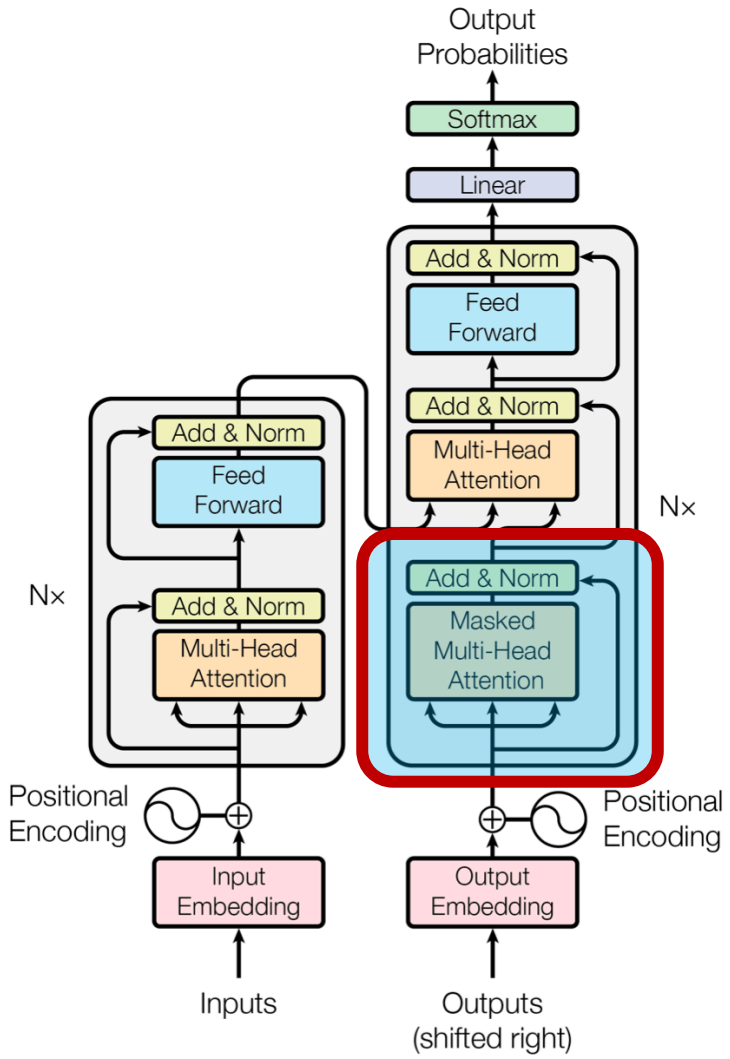
Transformer: Masked Multi Head Attention

Training

n <sos> Ich einen frau <eos>

Outputs at time T should only pay attention to outputs until time T-1

时间T的输出应仅关注时间T-1之前的输出



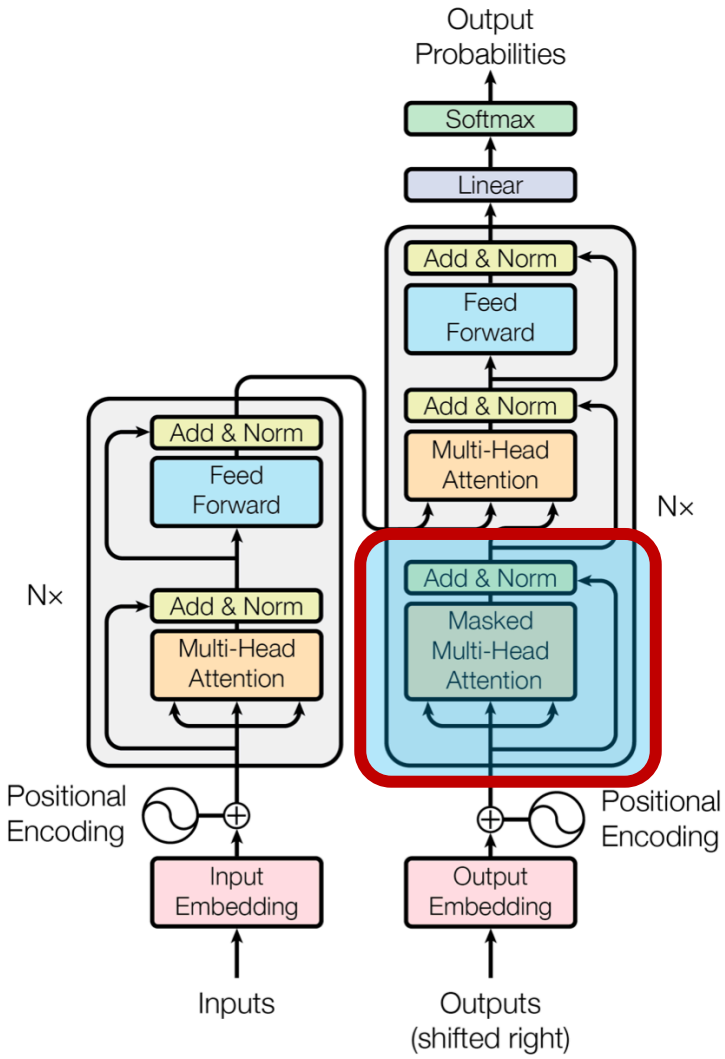
Transformer: Masked Multi Head Attention

Training

1	<sos>	Ich	einen	frau	<eos>
2	<sos>	Ich	einen	frau	<eos>
3	<sos>	Ich	einen	frau	<eos>
4	<sos>	Ich	einen	frau	<eos>
5	<sos>	Ich	einen	frau	<eos>

Outputs at time T should only pay attention to outputs until time T-1

时间T的输出应仅关注时间T-1之前的输出



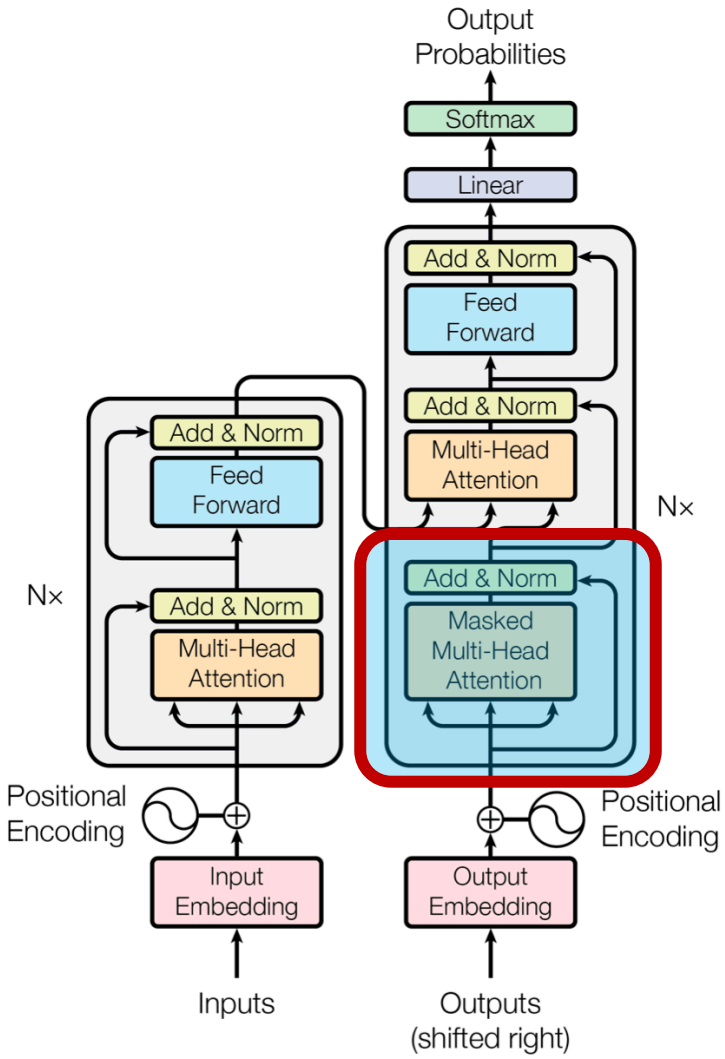
Transformer: Masked Multi Head Attention

Training

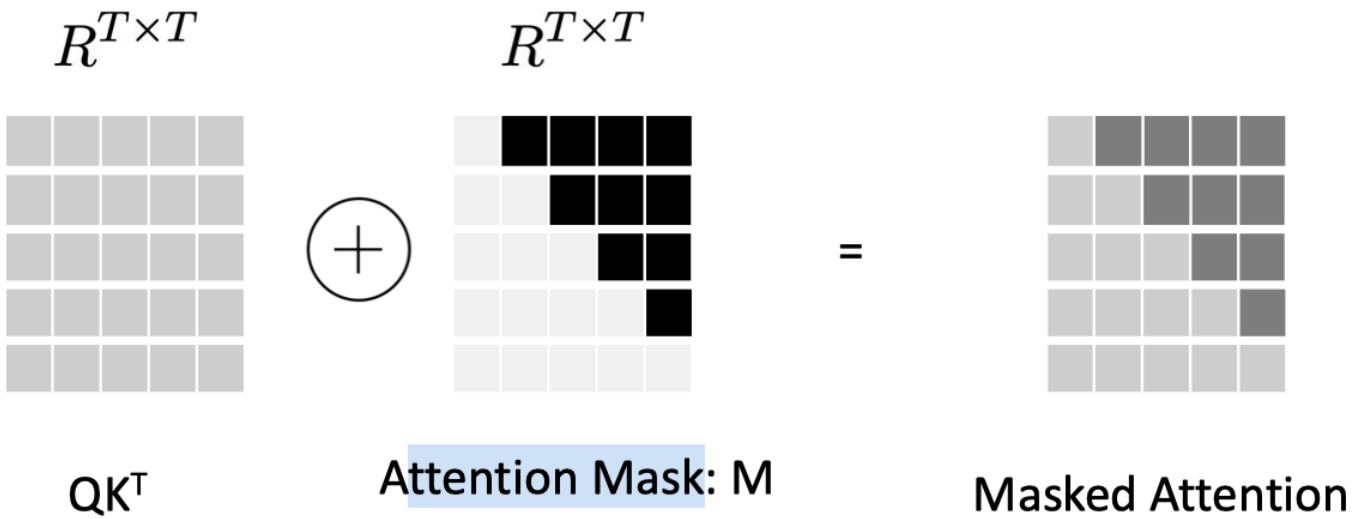
1	<sos>	Ich	einen	frau	<eos>
2	<sos>	Ich	einen	frau	<eos>
3	<sos>	Ich	einen	frau	<eos>
4	<sos>	Ich	einen	frau	<eos>
5	<sos>	Ich	einen	frau	<eos>

Mask the available attention values?

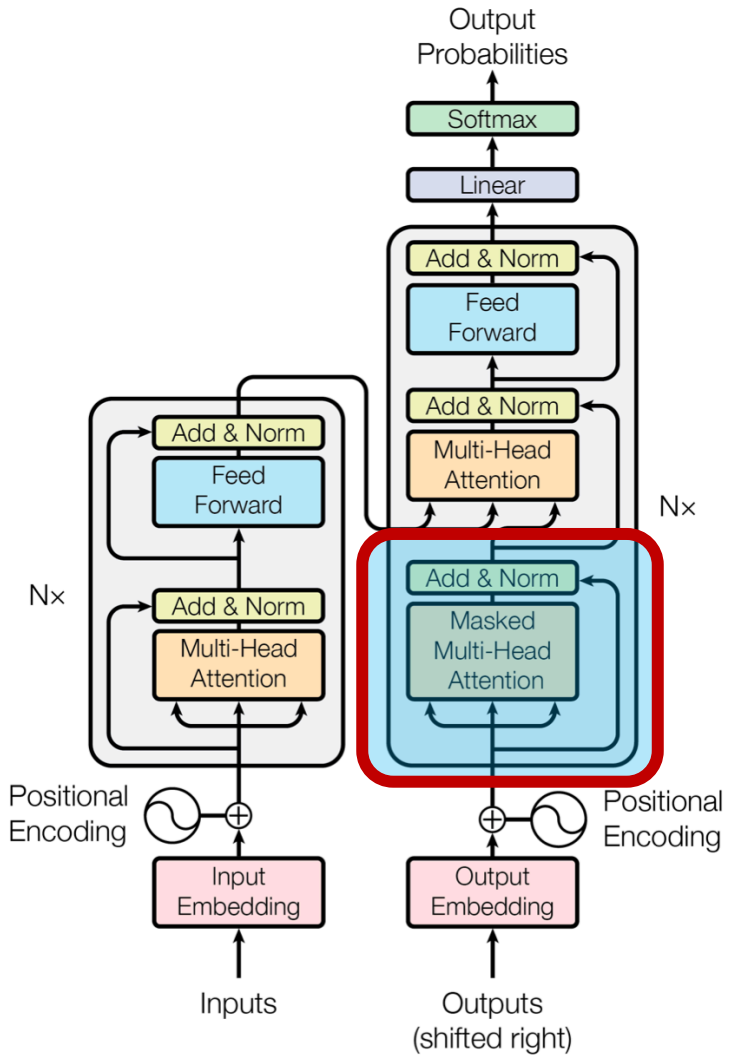
掩盖现有的注意力值?



Transformer: Masked Multi Head Attention

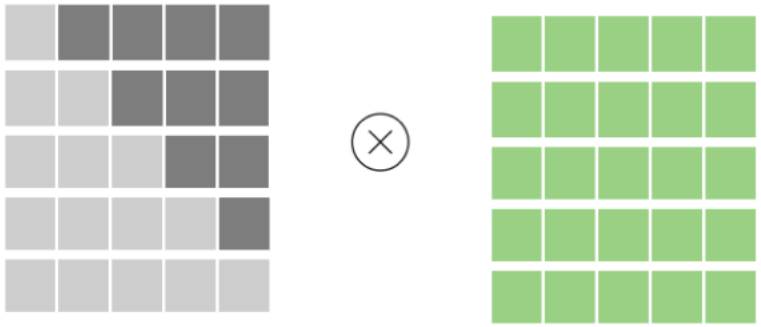


Masked Multi Head Attention: Z'



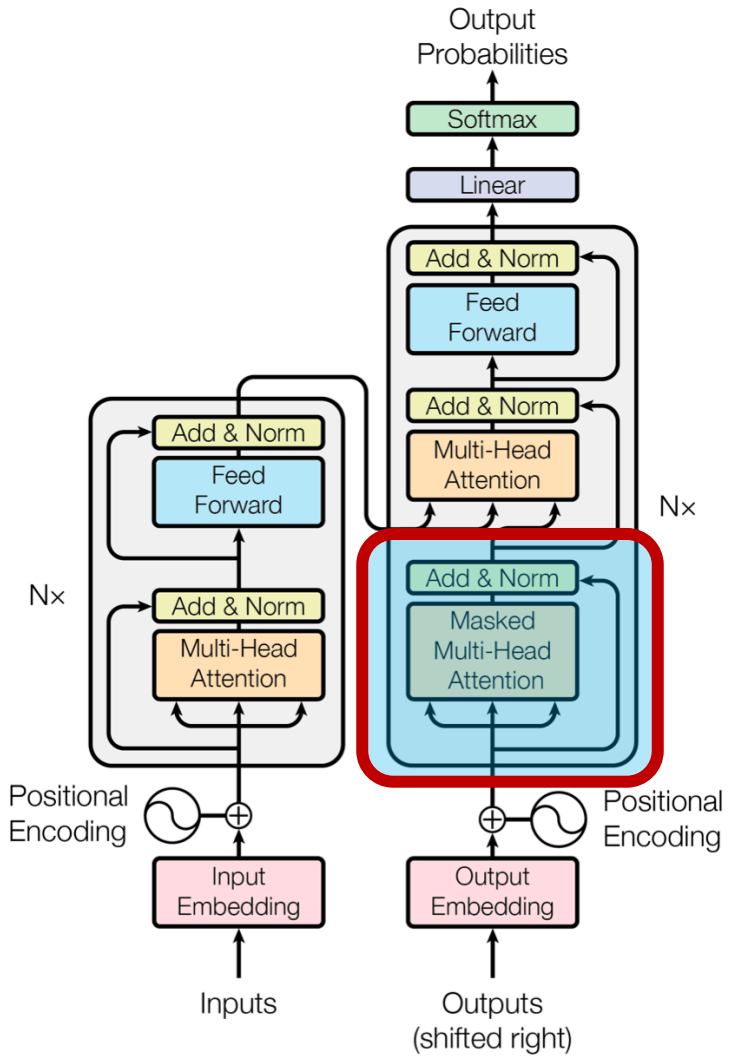
Transformer: Masked Multi Head Attention

$R^{T \times T}$ $R^{T \times d_h}$



Masked Attention Values

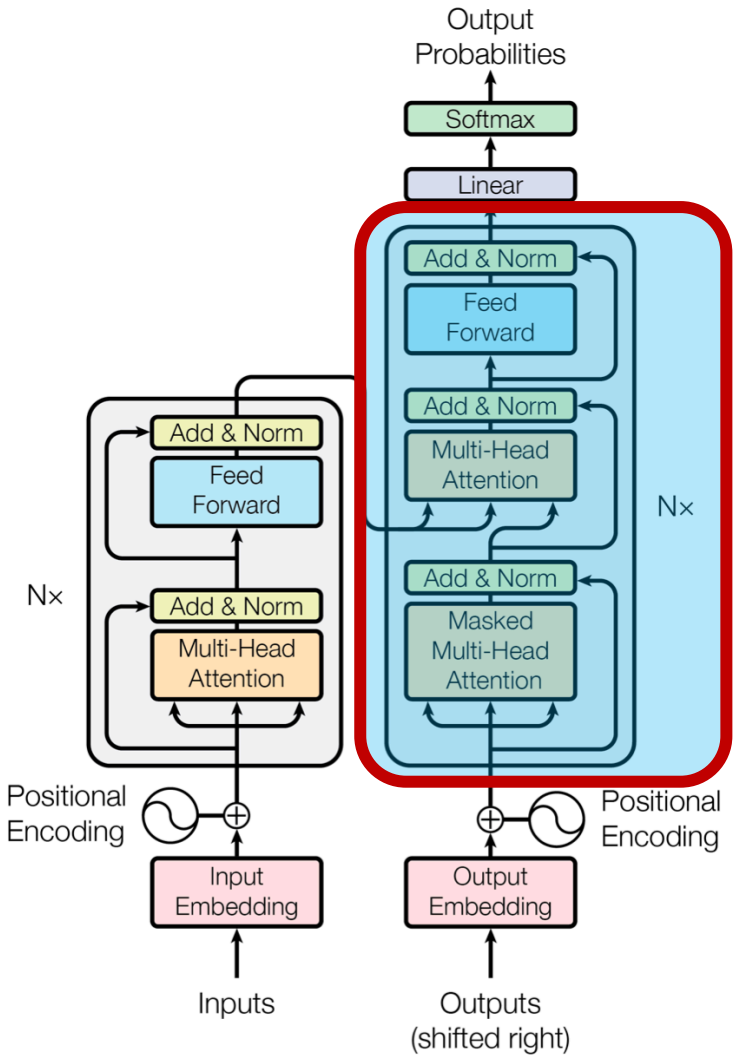
Masked Multi Head Attention: Z’



Transformer: Masked Multi Head Attention

Decoder

Decoder



Transformer: Masked Multi Head Attention

Decoder

Decoder

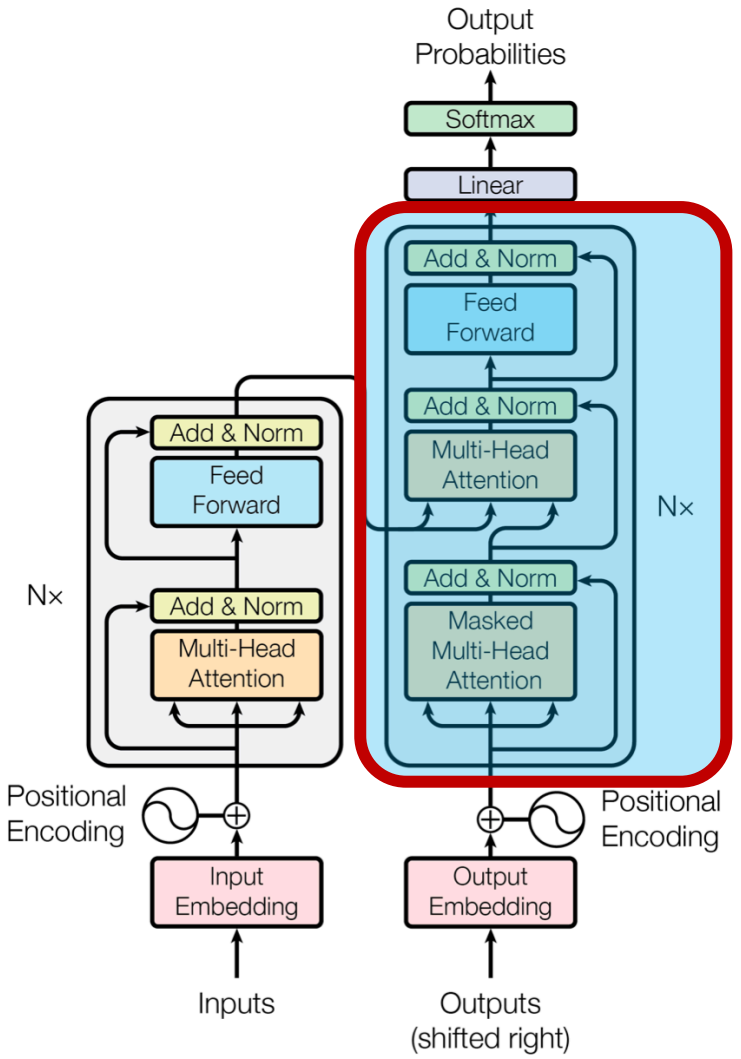
...

Decoder

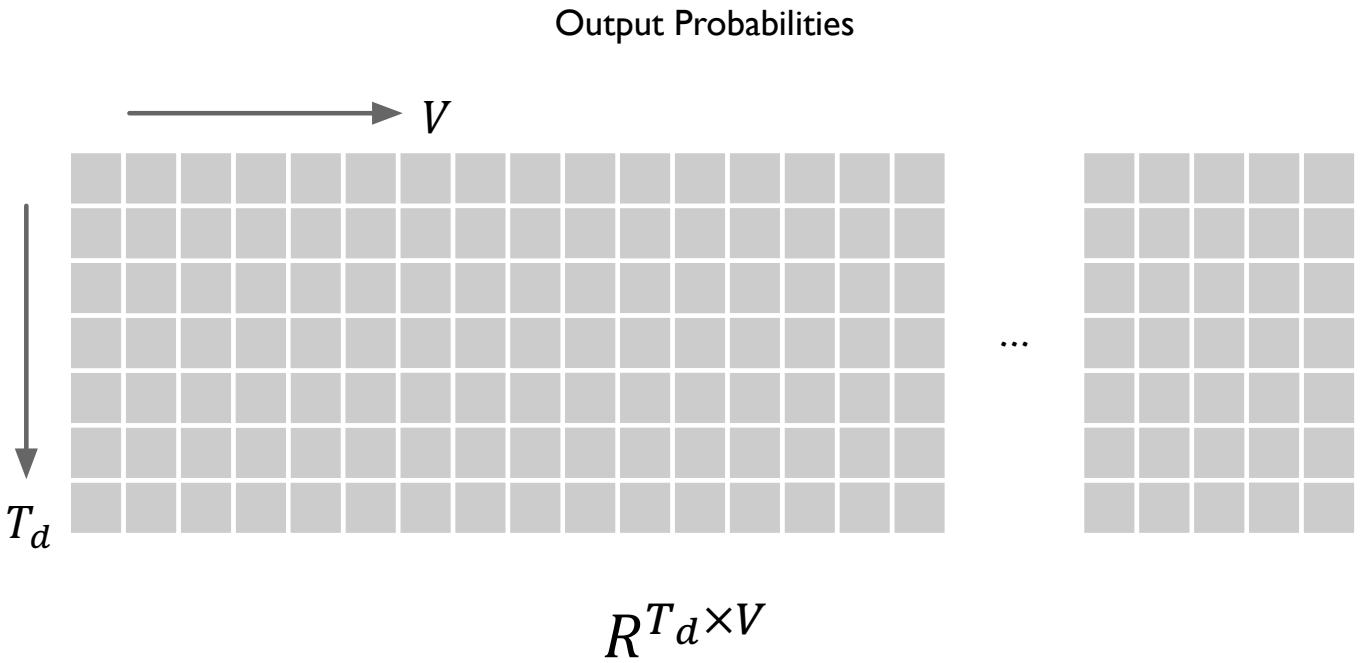
Decoder

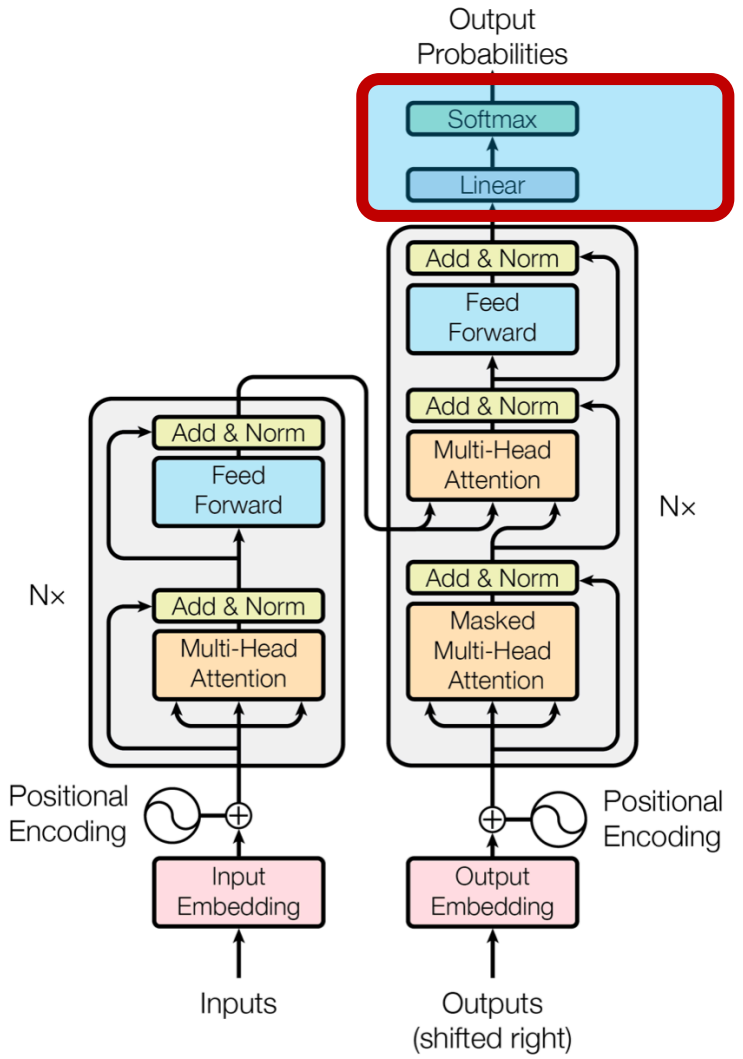
$R^{T_d \times d_{model}}$

Decoder Output



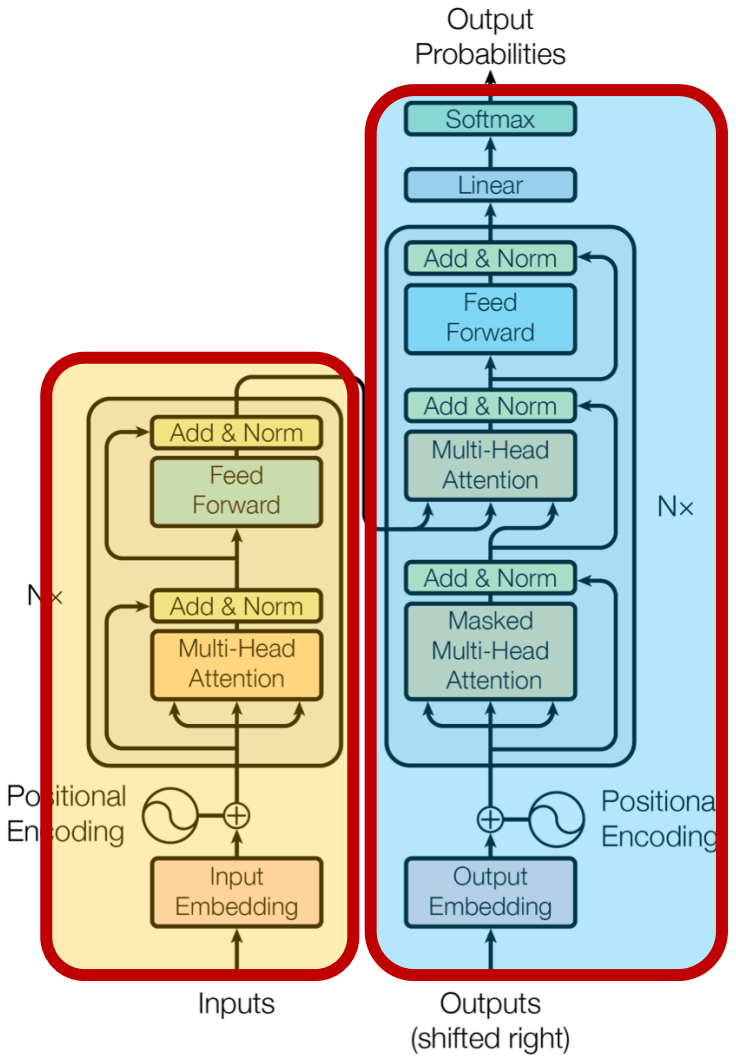
Transformer: Masked Multi Head Attention





"Original" Transformer (Vaswani et al., 2017).

- ✓ Tokenization
- ✓ Input Embeddings
- ✓ Position Encodings
- ✓ Residuals
- ✓ Query
- ✓ Key
- ✓ Value
- ✓ Add & Norm
- ✓ Encoder
- ✓ Decoder
- ✓ Attention
- ✓ Self Attention
- ✓ Multi Head Attention
- ✓ Masked Attention
- ✓ Encoder Decoder Attention
- ✓ Output Probabilities / Logits
- ✓ Softmax
- ✓ Decoder only models



Transformer 最终效果

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

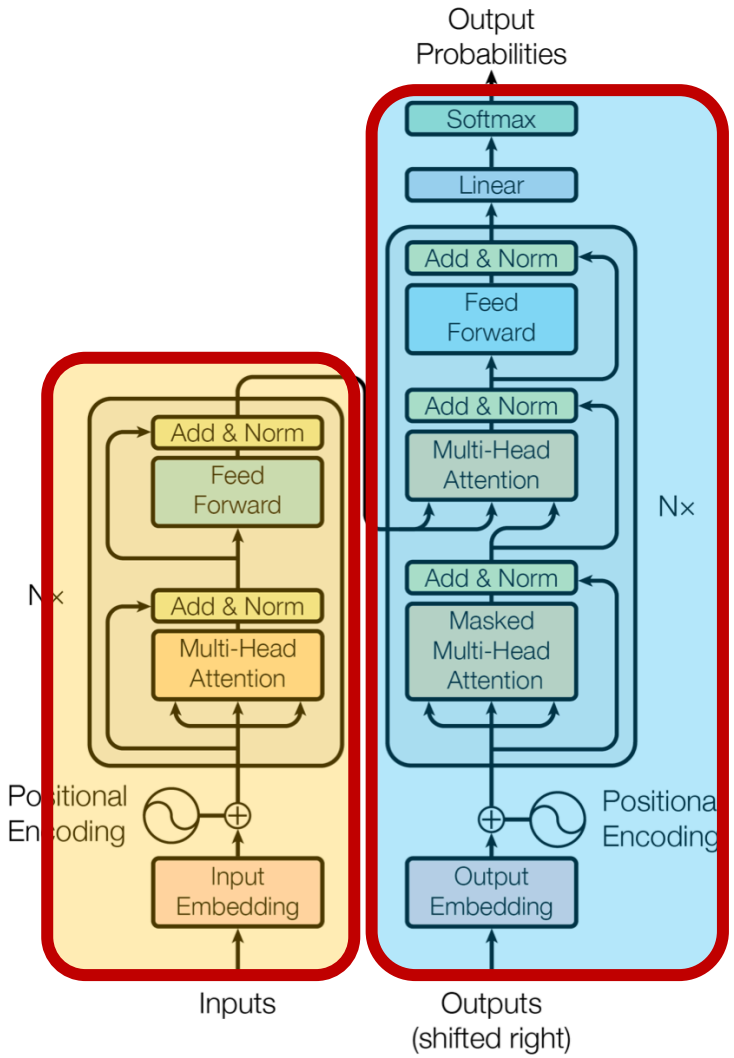
Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Transformer 优点

1. 模型在处理序列输入时，可以对整个序列输入进行并行计算，突破 RNN 不能并行计算限制
2. 相比 CNN，计算两个位置之间的关联所需的操作次数不随距离增长
3. Attention 机制更具可解释性，各注意头 attention head 可以学习不同的关联特征
4. 模型的规模可以随着 Encoder 和 Decoder 的堆叠而不断增加，不需要考虑特征减少

4. BERT vs GPT

从 Transformers 到大模型



Representation

Generation

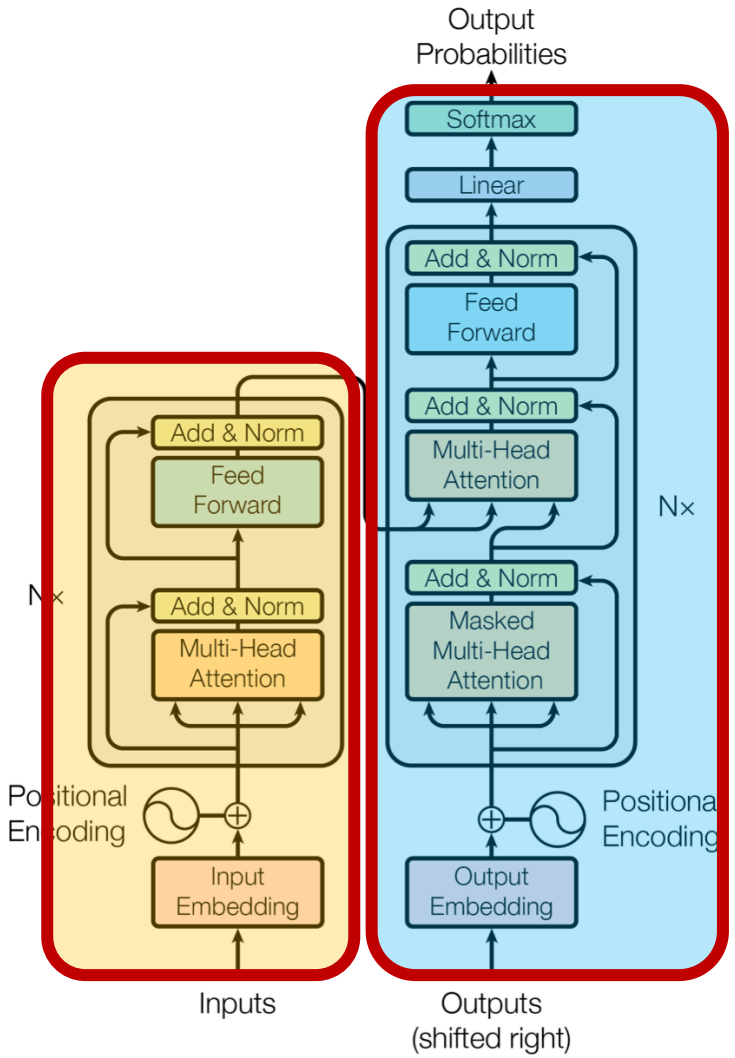
从 Transformers 到大模型

- **Input** – input tokens
- **Output** – hidden states

- Model can see all timesteps

- Does not usually output tokens, so no inherent auto-regress

Representation



- **Input** – output tokens and hidden states*
- **Output** – output tokens

- Model can only see previous timesteps

- Model is auto-regressive with previous timesteps' outputs

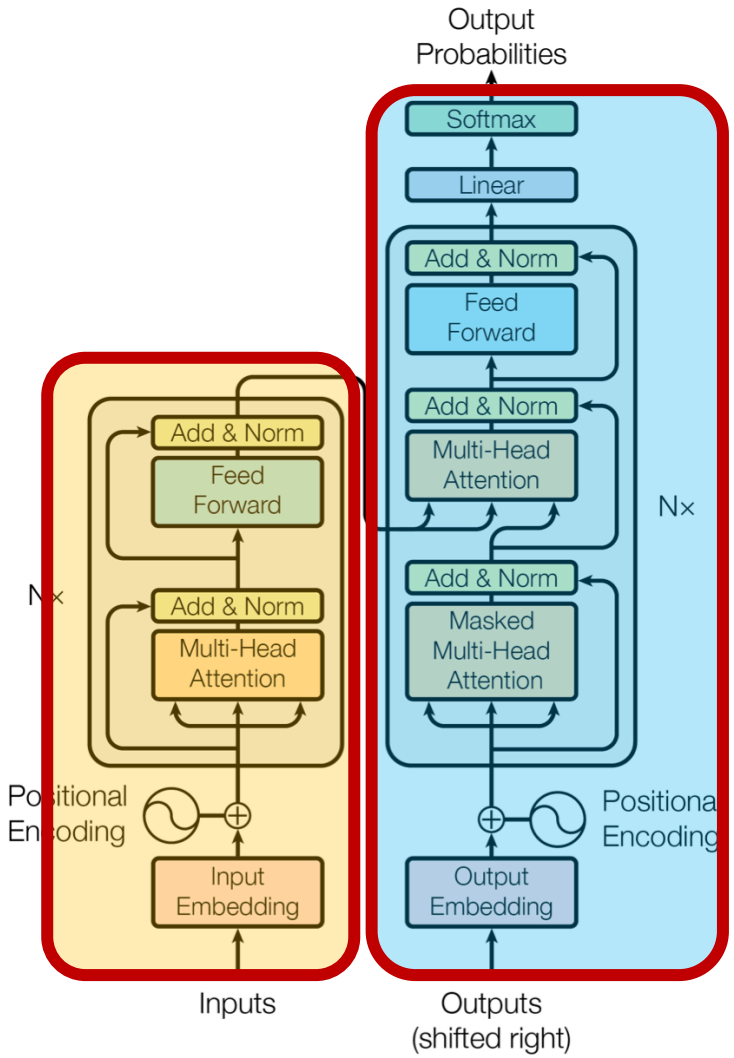
Generation

2018 – The Inception of the LLM Era

BERT

Oct 2018

Representation



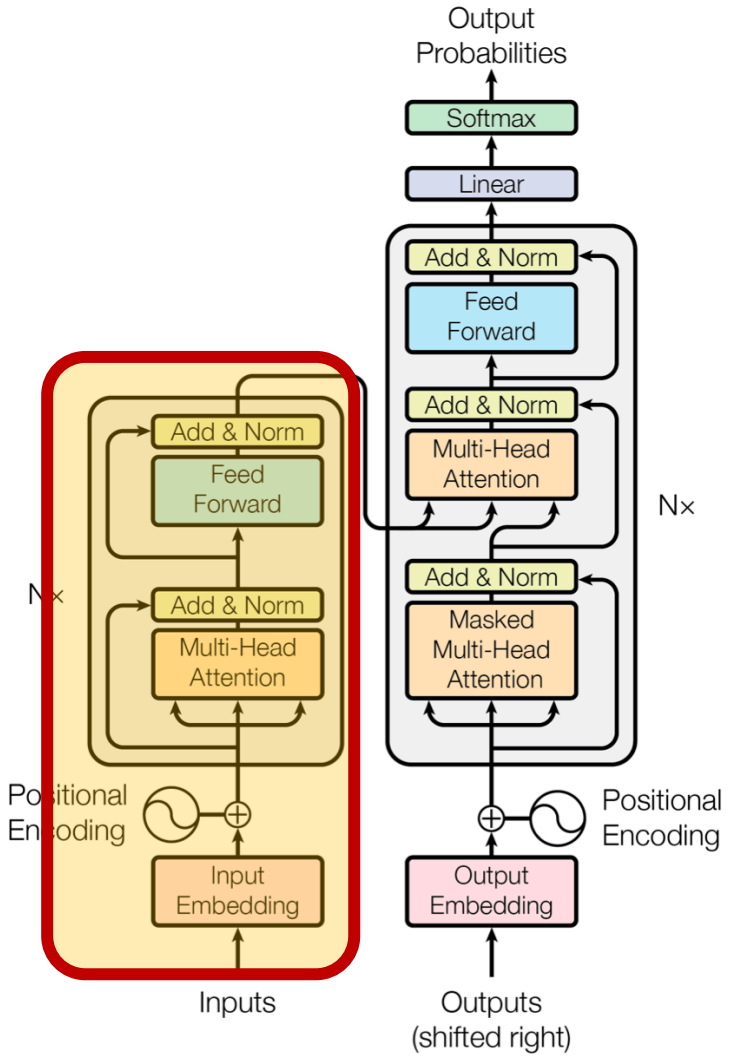
GPT-1

Jun 2018

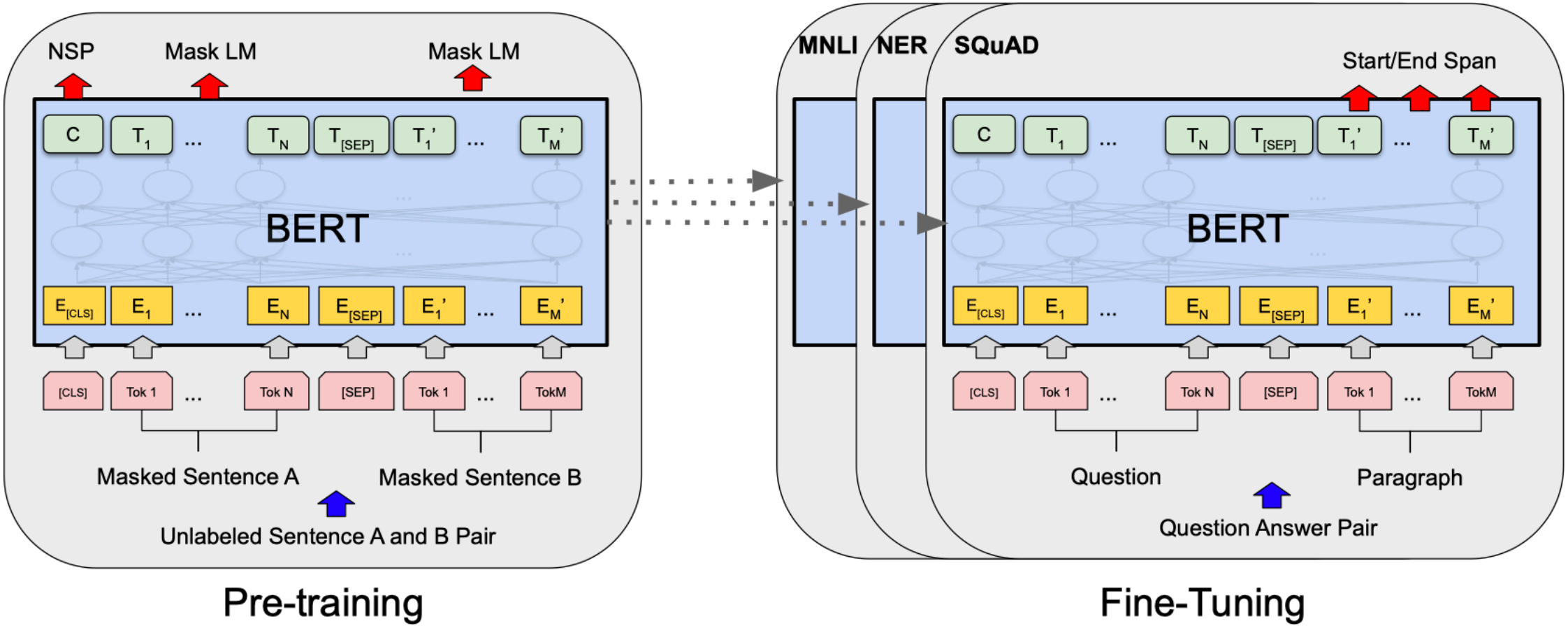
Generation

BERT

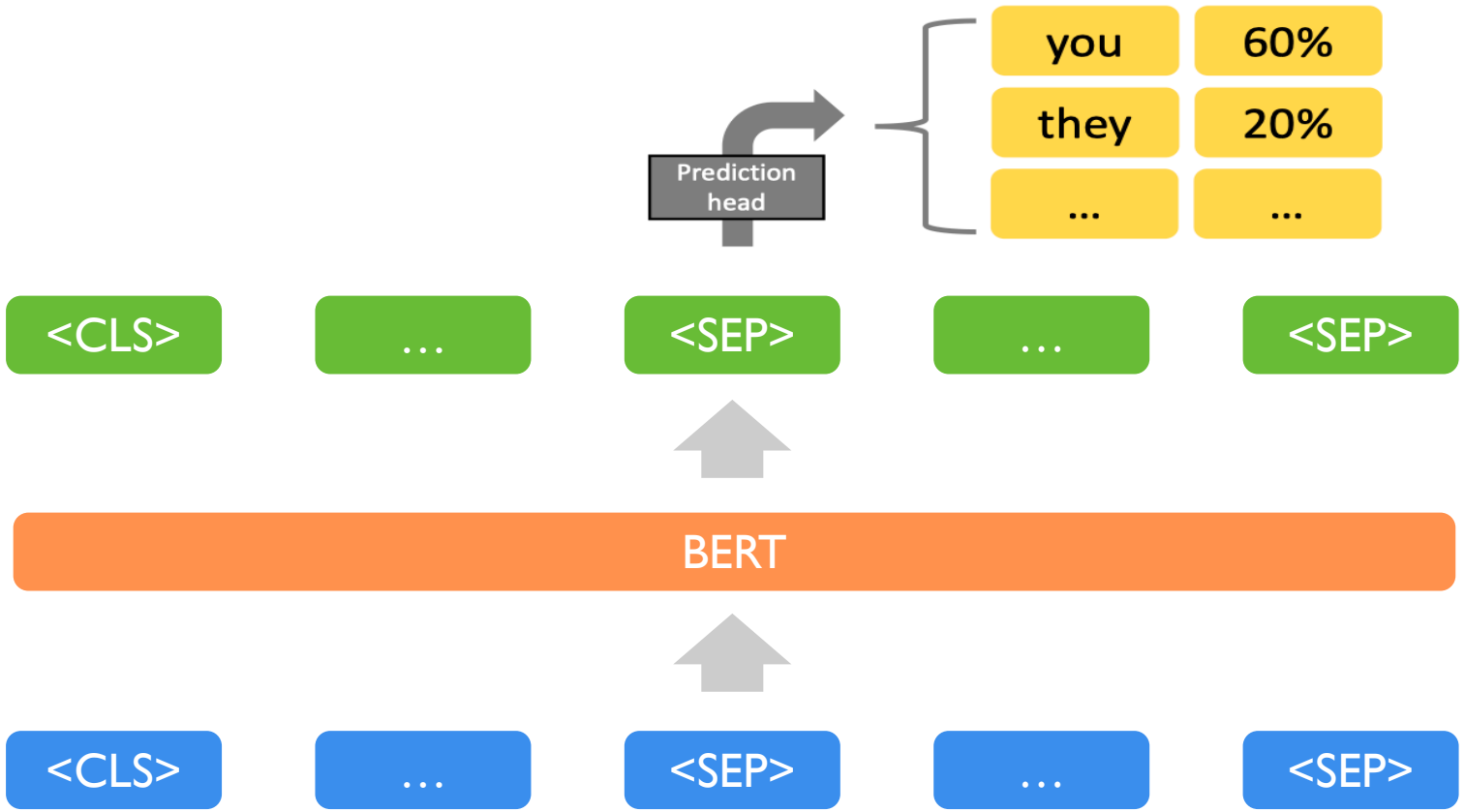
- BERT (Bidirectional Encoder Representation from Transformers, Devlin et al., 2018) is an encoder of a transformer pre-trained with:
- • Masked Language Model (MLM), that consists in predicting [15% of] words which have been replaced with a “MASK” token.
- • Next Sentence Prediction (NSP), which consists in predicting if a certain sentence follows the current one.
- It is then fine-tuned on multiple NLP tasks.



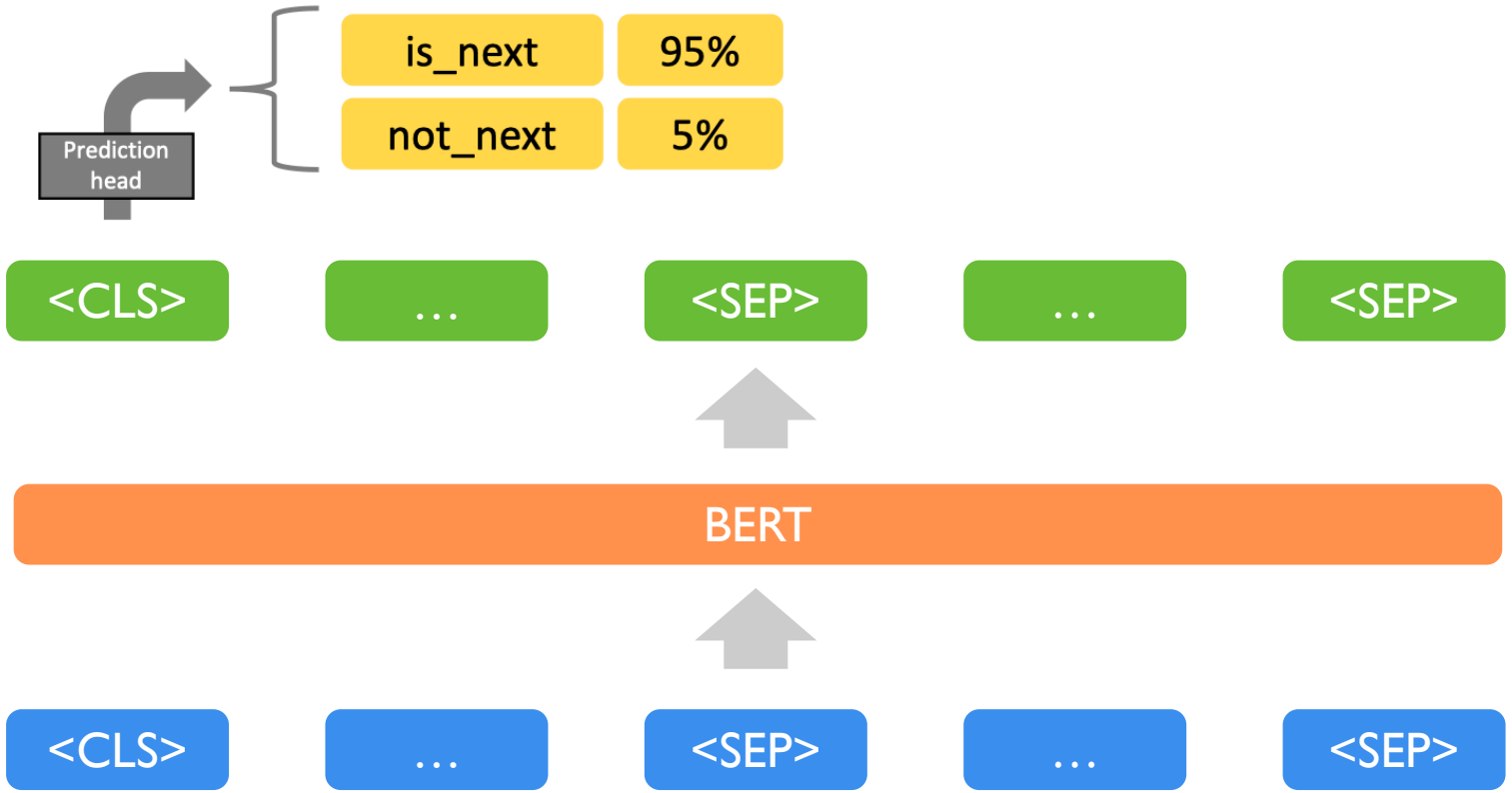
BERT : from pre-training to Fine-Tuning



MLM, Masked Language Modeling



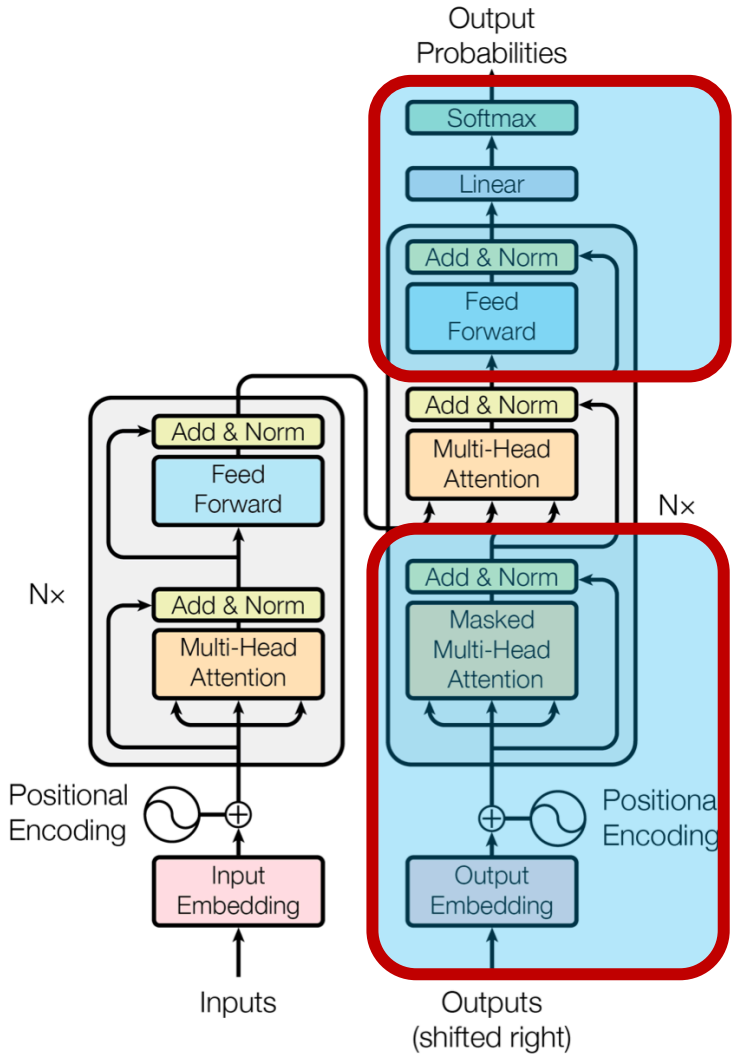
MLM, Masked Language Modeling



GPT – Generative Pretrained Transformer

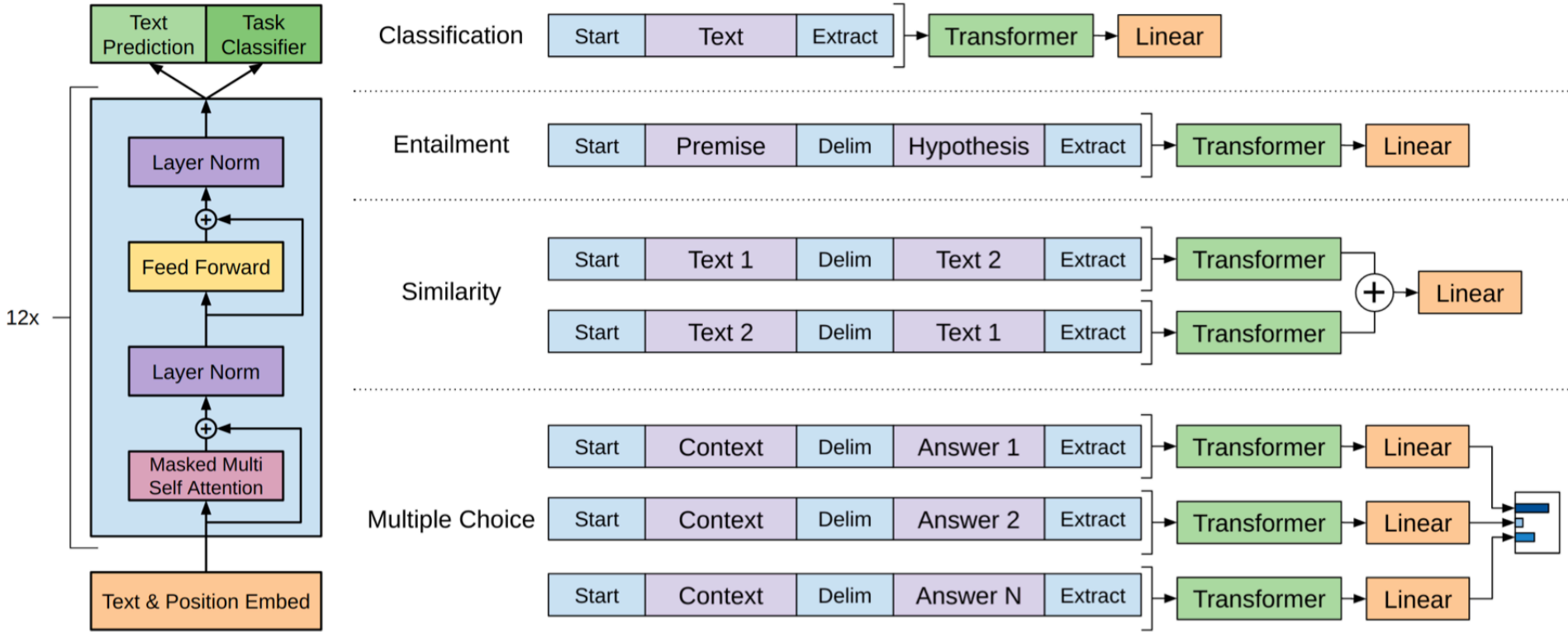
- 一般以字（Tokes）为单位训练 Transformer 模型。首先初始化字编码大小为 [vocab size, embedding dimension]，vocab size 为字库中所有字的数量。

- 对于输入的句子 X，通过 Word Embedding 得到该句子中每个字的字向量，通过 Positional Encoding 得到所有字位置向量，将其相加得到该字真正的向量表示。第 t 个字的向量记作 x_t。

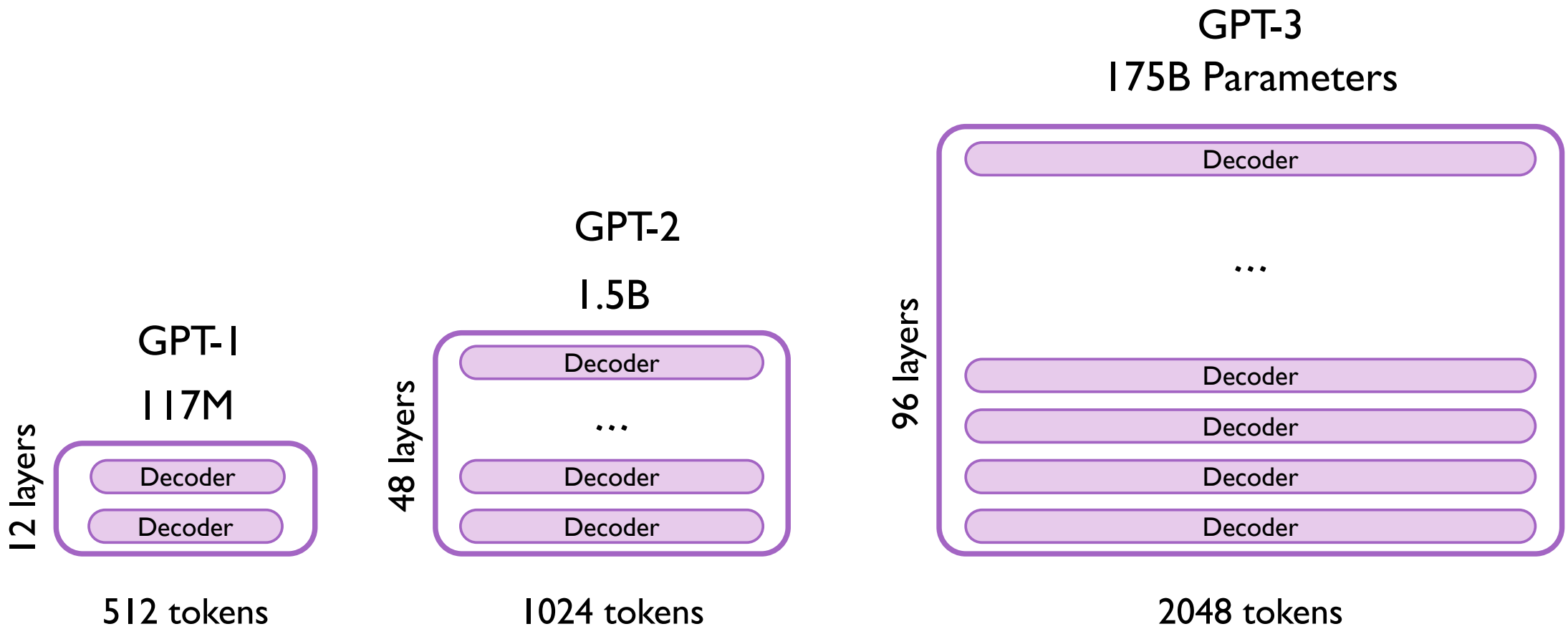


GPT – Generative Pretrained Transformer

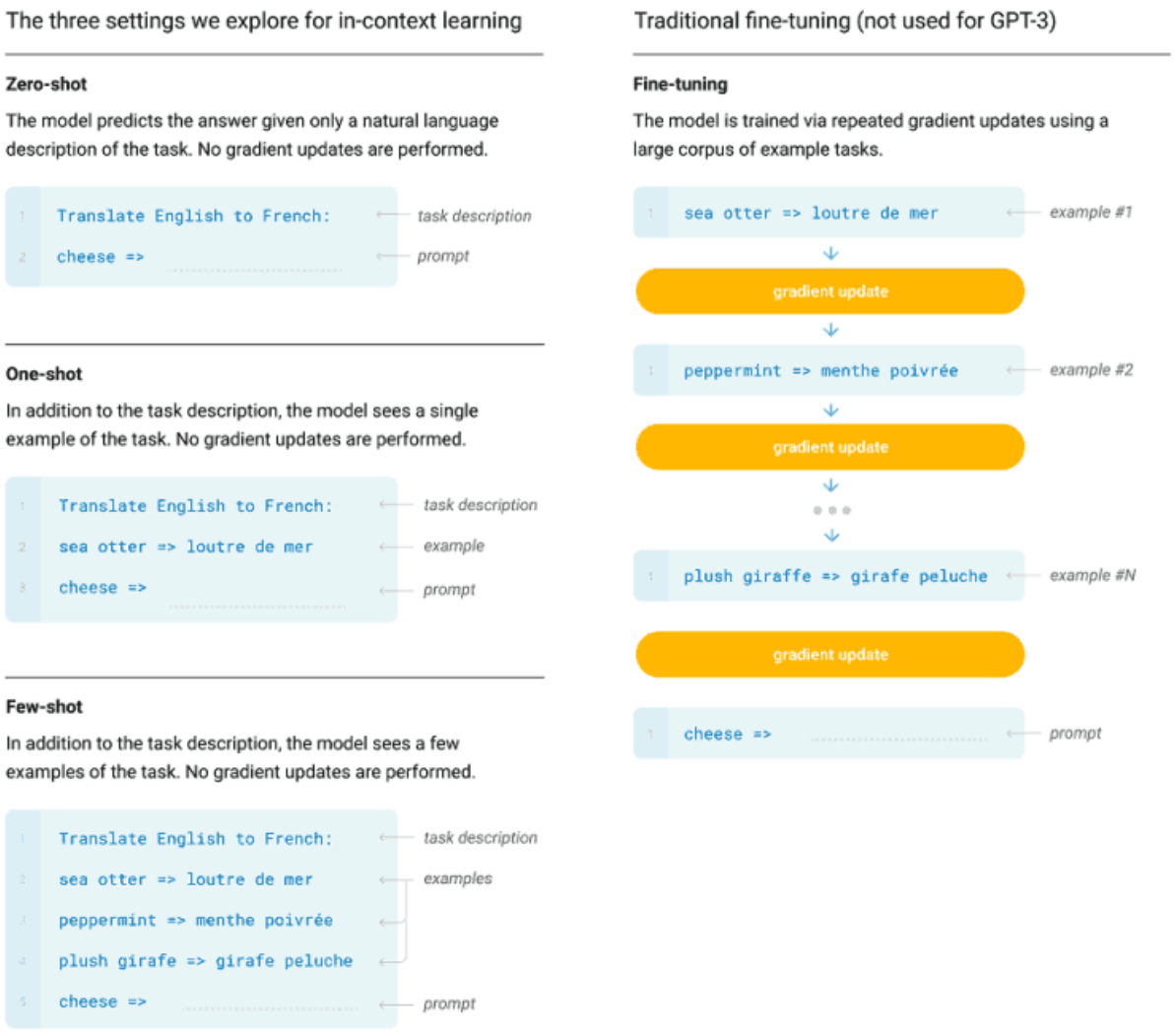
- GPT (Generative Pre-Training, Radford, 2018) is a decoder of a transformer trained for auto-regressive text generation.



GPT 系列


GPT-1
117M
12 layers
Decoder
Decoder
512 tokens
GPT-2
1.5B
48 layers
Decoder
...
Decoder
Decoder
1024 tokens
GPT-3
175B Parameters
96 layers
Decoder
...
Decoder
Decoder
Decoder
Decoder
2048 tokens

GPT – Generative Pretrained Transformer



小结

1. GPT是单向模型，不需要利用上下文信息，只能利用上文；而 BERT 是双向模型，训练时候需要上下文信息，更为复杂。
2. GPT是基于自回归模型，能够进行文本生成，可以通过 Prompt 应用在众多 NLP 任务中；BERT 采用基于自编码模型，先预训练再微调的两个阶段来学习通用文本表示，只能完成 NLP 特定任务，无法直接应用在文本生成。

Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

ZOMI

Course chenzomi12.github.io

GitHub github.com/chenzomi12/DeepLearningSystem