



# 第十周 Transformer模型

李泽样，复旦大学生物医学工程与技术创新学院



# 目录

1 Transformer介绍

2 Transformer结构

3 视觉Transformer

4 语言Transformer

# Transformer论文: Attention Is All You Need



## 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

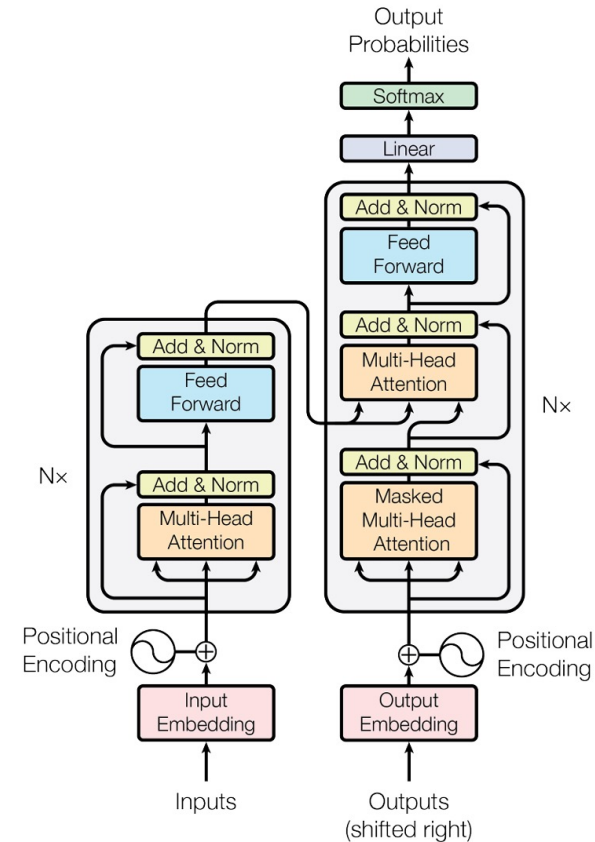
**Aidan N. Gomez\* †**  
University of Toronto  
aidan@cs.toronto.edu

**Lukasz Kaiser\***  
Google Brain  
lukaszkaizer@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.



- 2017年发布以来, 引用量17万+
- 目前整个AI领域就靠它了, 具备大一统可能性
- 所有作者都已出走创业
- 真的是: Attention is ALL you need

# Transformer应用与成就：自然语言处理



- 机器翻译



- 其他自然语言处理任务

- 基于Transformer的模型稳居SuperGLUE任务榜单前列

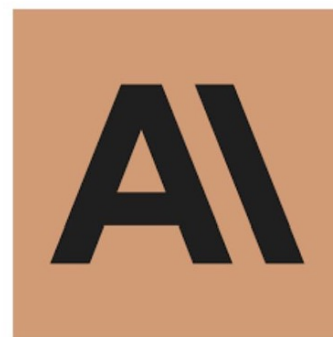
- 大语言模型的兴起



Gemini / Bard  
(Google)



ChatGPT / GPT-4  
(OpenAI)



Claude 3  
(Anthropic)

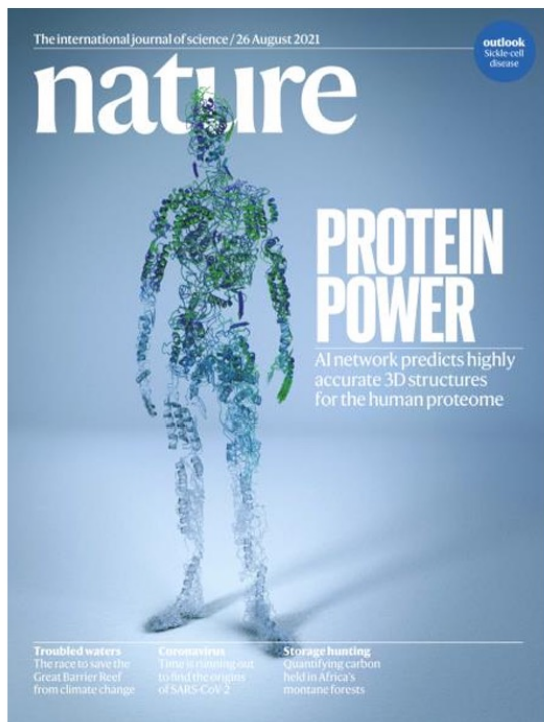


Llama 3  
(Meta)

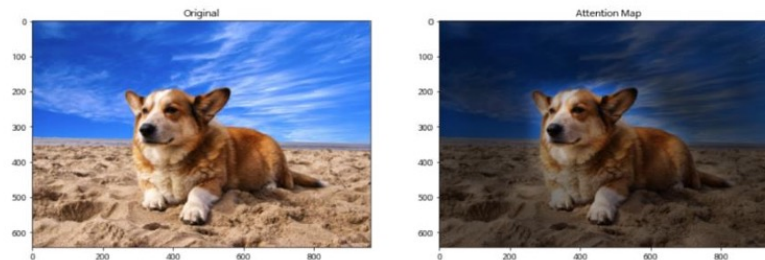
# Transformer应用与成就：其他领域



## Protein Folding



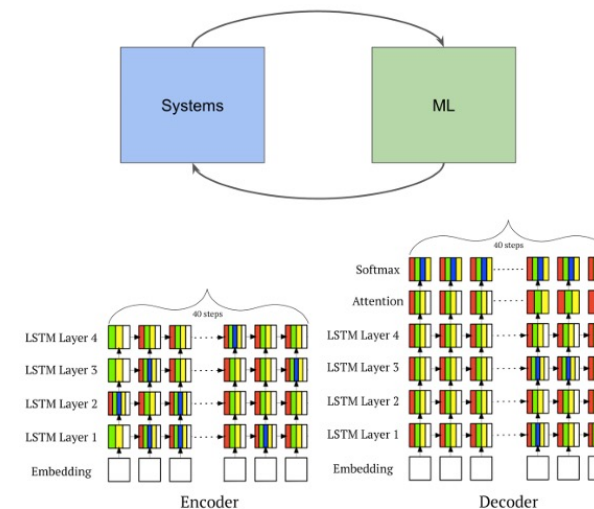
[Jumper et al. 2021] aka AlphaFold2!



## Image Classification

[Dosovitskiy et al. 2020]: Vision Transformer (ViT) outperforms ResNet-based baselines with substantially less compute.

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.55 ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	90.72 ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	99.50 ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	—
CIFAR-100	94.55 ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	—
Oxford-IIIT Pets	97.56 ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	—
Oxford Flowers-102	99.68 ± 0.02	99.74 ± 0.00	99.61 ± 0.02	99.63 ± 0.03	—
VTAB (19 tasks)	77.63 ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	—
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k



## ML for Systems

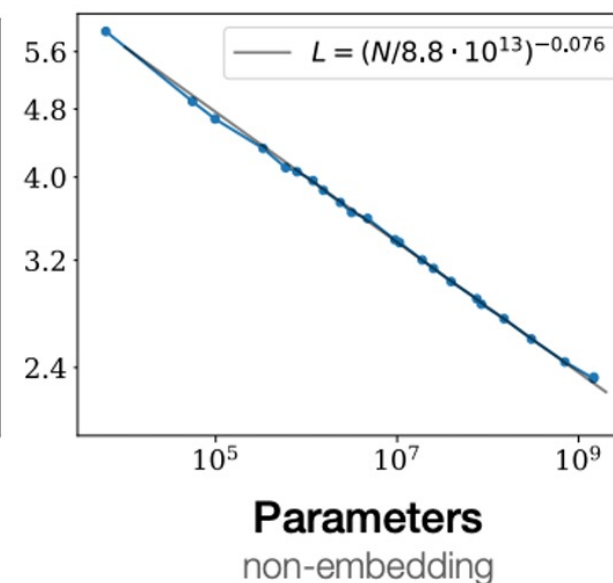
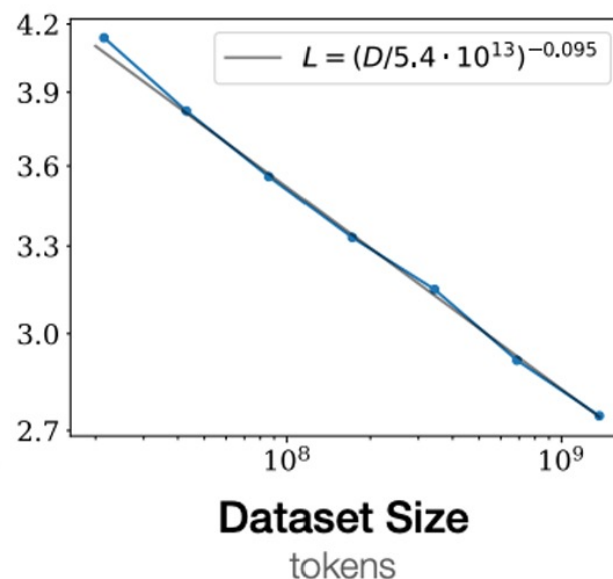
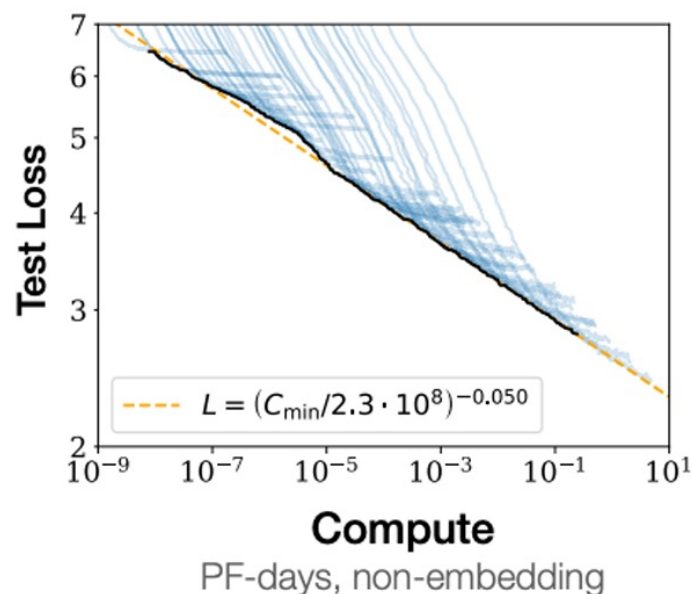
[Zhou et al. 2020]: A Transformer-based compiler model (GO-one) speeds up a Transformer model!

Model (#devices)	GO-one (s)	HP (s)	METIS (s)	HDP (s)	Run time speed up over HP / HDP	Search speed up over HDP
2-layer RNNLM (2)	0.173	0.192	0.355	0.191	9.9% / 9.4%	2.95x
4-layer RNNLM (4)	0.210	0.239	0.503	0.251	13.8% / 16.3%	1.76x
8-layer RNNLM (8)	0.320	0.332	OOM	0.764	3.8% / 58.1%	27.8x
2-layer GNNMT (2)	0.301	0.384	0.344	0.327	27.6% / 14.3%	30x
4-layer GNNMT (4)	0.350	0.469	0.466	0.432	34% / 23.4%	58.8x
8-layer GNNMT (8)	0.440	0.562	OOM	0.693	21.7% / 36.5%	7.35x
2-layer Transformer-XL (2)	0.223	0.268	0.37	0.262	20.1% / 17.4%	40x
4-layer Transformer-XL (4)	0.230	0.27	OOM	0.259	17.4% / 12.6%	26.7x
8-layer Transformer-XL (8)	0.350	0.46	OOM	0.425	23.9% / 16.7%	16.7x
Inception (2) b64	0.229	0.312	OOM	0.301	26.6% / 23.9%	13.5x
AmoebaNet (2) b64	0.423	0.731	OOM	0.498	42.1% / 29.3%	21.0x
AmoebaNet (4)	0.394	0.44	0.426	0.418	26.1% / 6.1%	58.8x
2-stack 18-layer WaveNet (2)	0.317	0.376	OOM	0.354	18.6% / 11.7%	6.67x
4-stack 36-layer WaveNet (4)	0.659	0.988	OOM	0.721	50% / 9.4%	20x
GEOMEAN	-	-	-	-	20.5% / 18.2%	15x

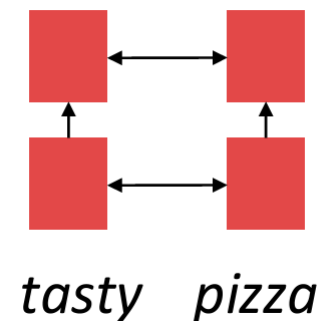
# Transformer应用与成就: Scaling Laws



- 借助Transformer模型结构, 随着模型规模、训练数据量和计算资源的同步增加, 语言建模的性能实现了稳步提升
- 如果我们继续扩大这些模型的规模 (不改变架构), 它们最终能否达到或超越人类的水平?

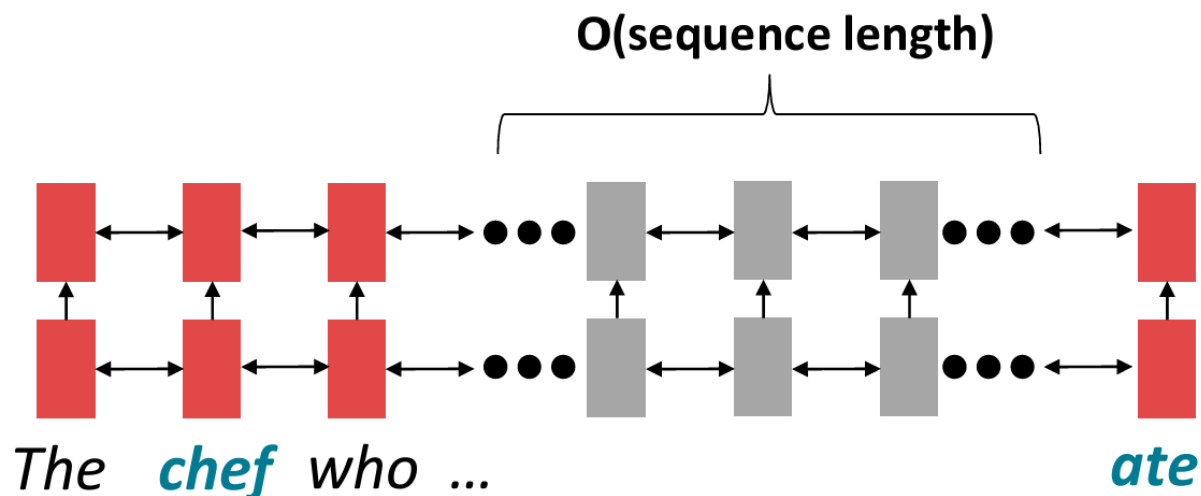


- RNN “从左到右” 顺序处理输入
- RNN 编码了线性局部性
  - 相邻的词语往往会相互影响它们的含义



- **Problem:**

- RNNs需要  **$O(\text{序列长度})$**  的步骤才能与距离较远的词进行交互
- 前向和后向传播都有  **$O(\text{序列长度})$**  的无法并行的操作



主语 *chef* 的信息需要传递  **$O(\text{序列长度})$**  层才能与动词 *ate* 交互!

# 目录

1 Transformer介绍

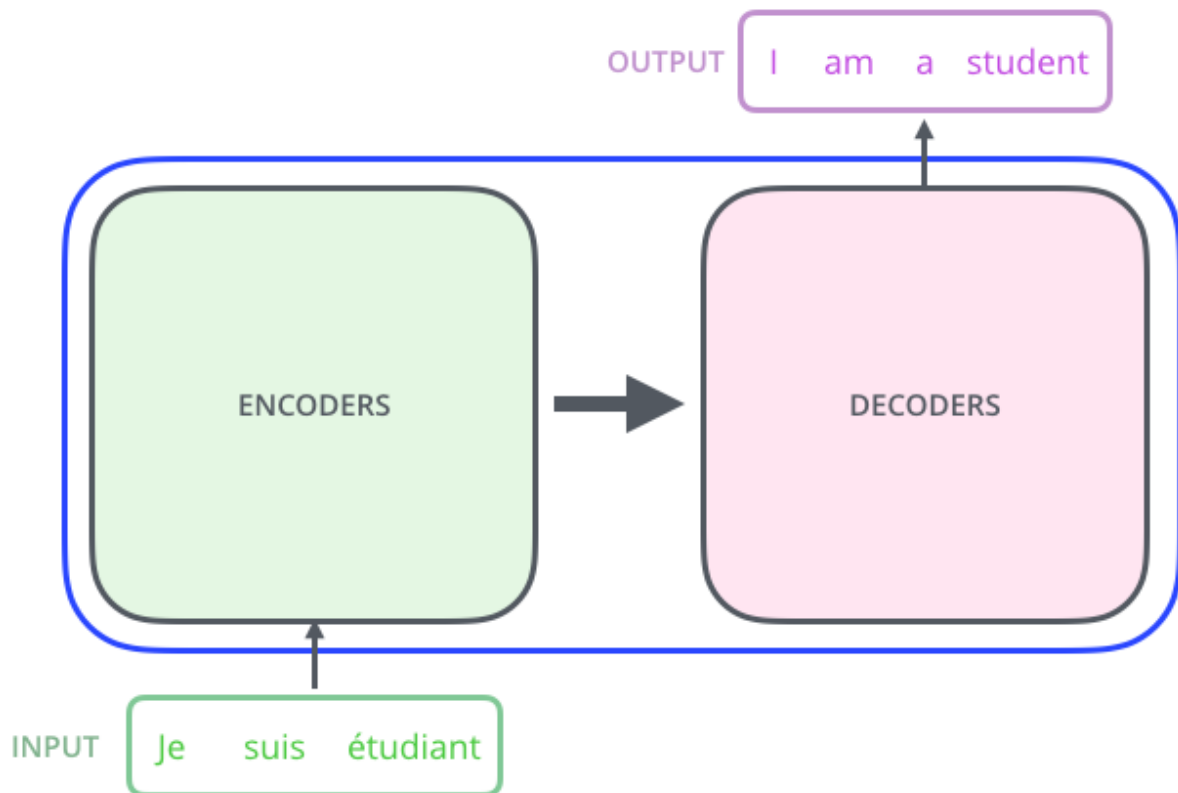
2 **Transformer结构**

3 视觉Transformer

4 语言Transformer

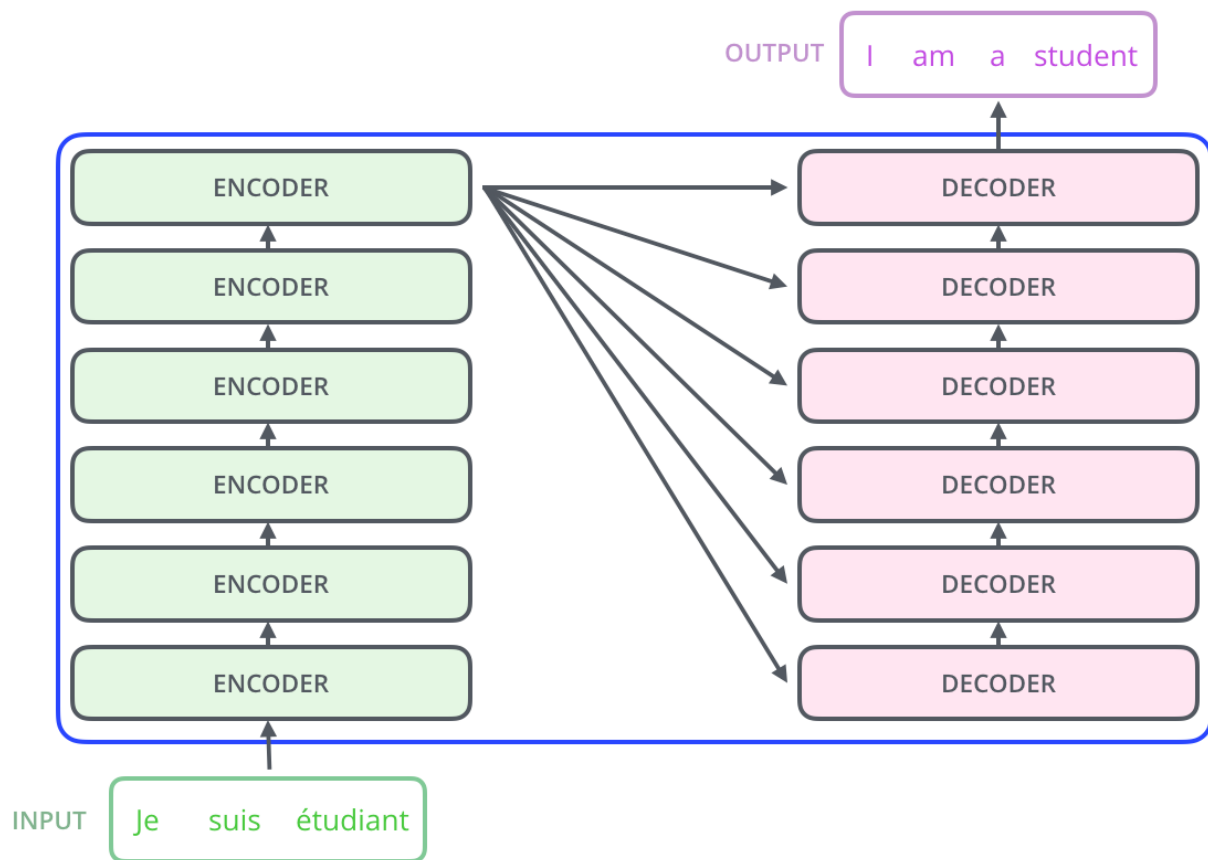


- Transformer网络是用于序列到序列学习 (Seq2Seq) 的神经网络
- 它将给定的元素序列 (例如一个句子, 即一系列词语) 转换为另一个序列
- 自然地, 这种Seq2Seq模型的一个应用就是翻译。

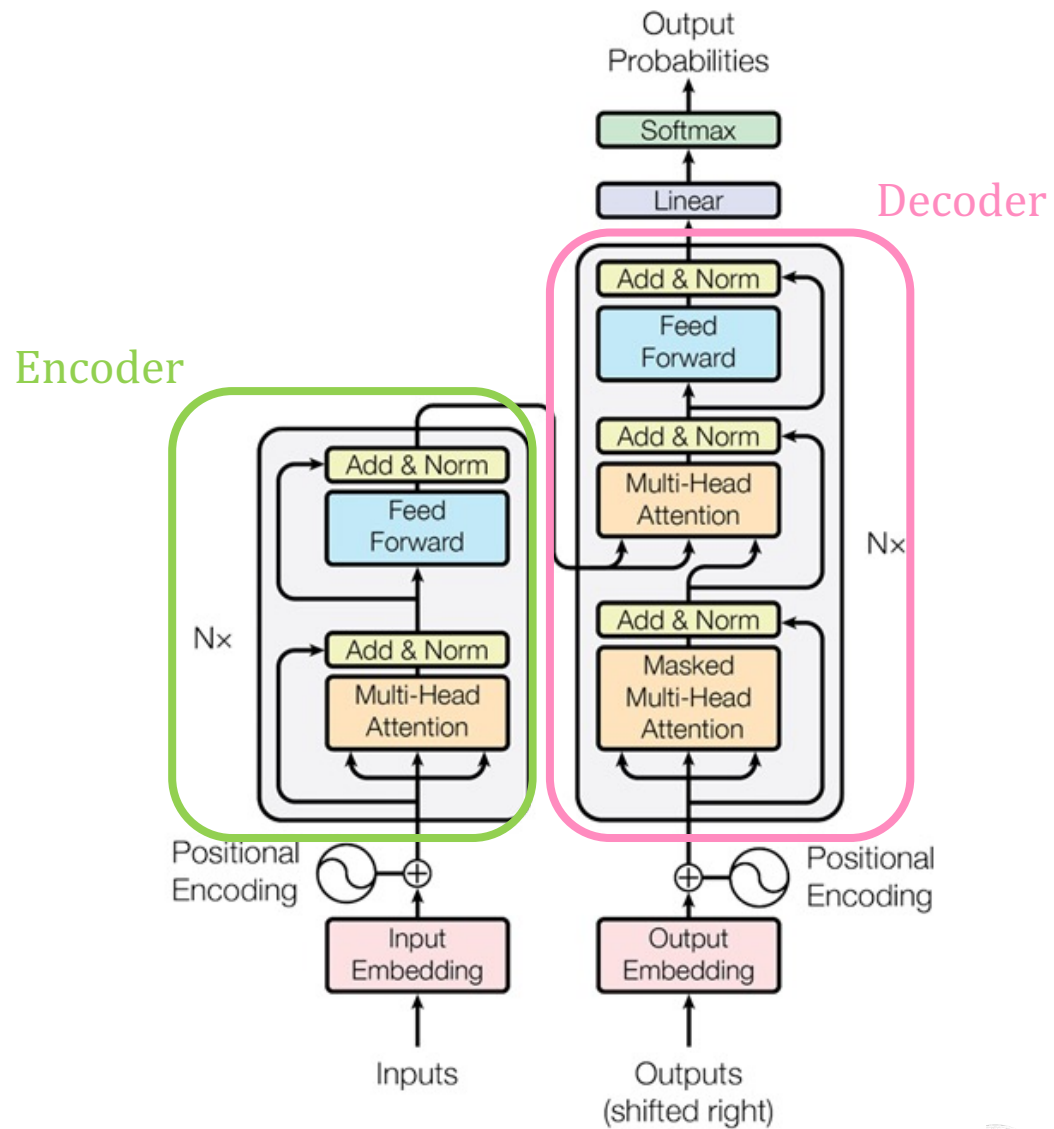


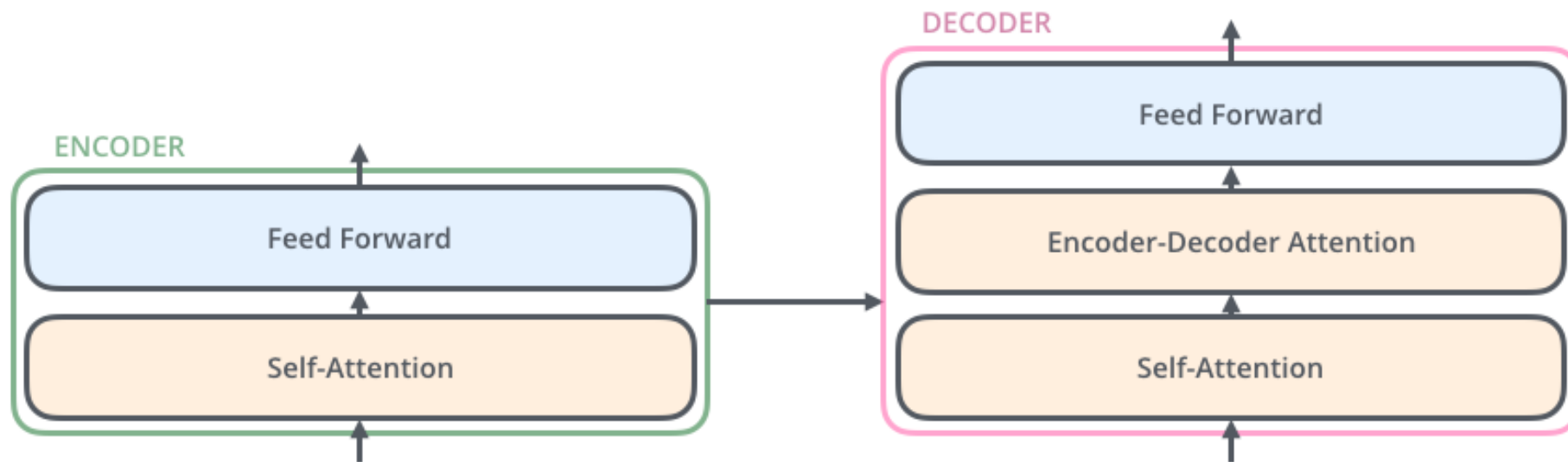
- **Encoder** - 将输入序列映射为一个抽象的连续表示, 该表示包含了输入中所有学习到的信息
- **Decoder** - 利用encoder输出的信息, 同时将自身先前的输出作为输入进行处理, 生成下一步的输出

# Transformer概述



- Encoder部分由多个encoder结构堆叠而成
- encoder的结构都是相同的
- 权重不共享



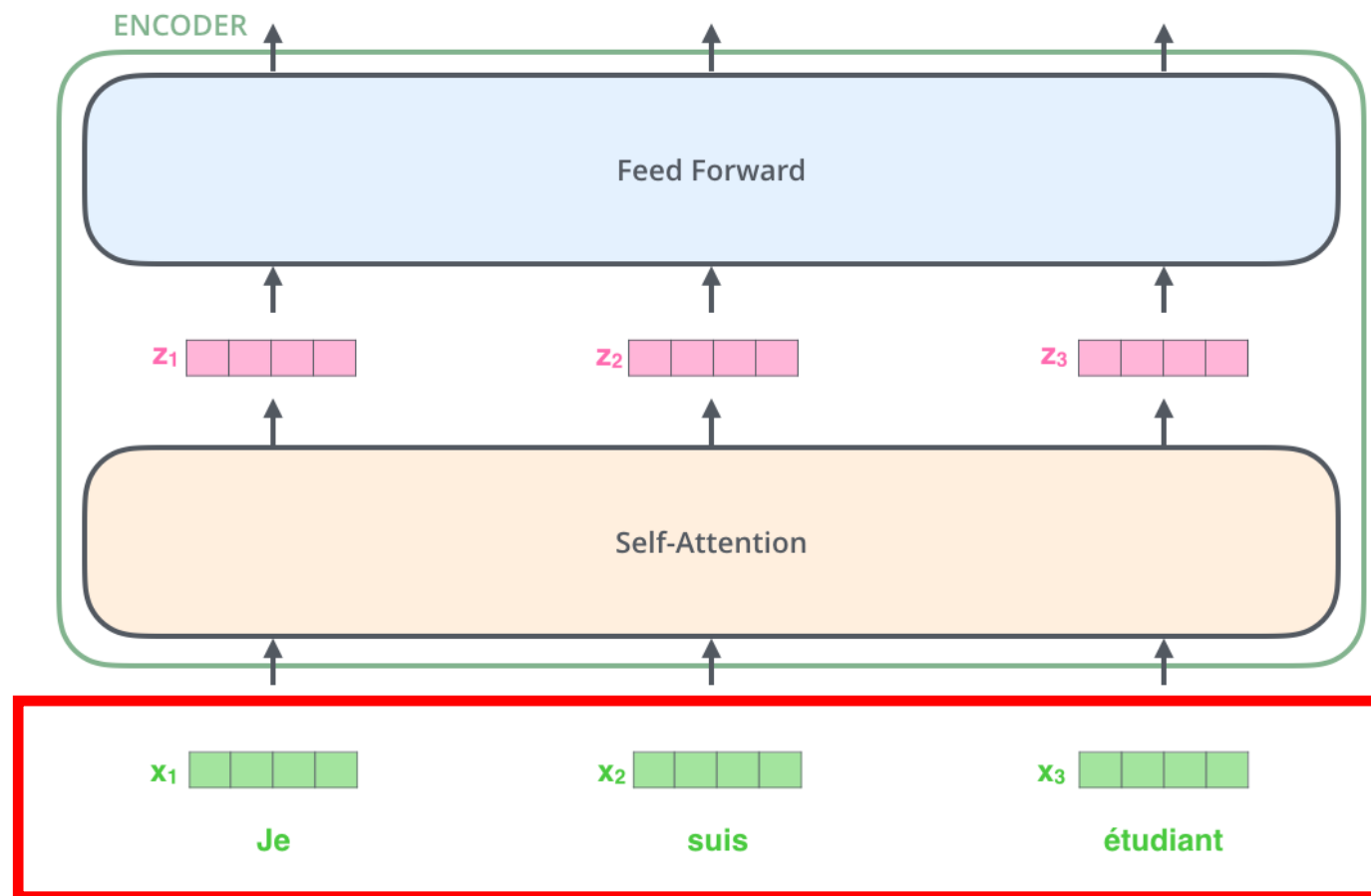


- **自注意力层 (Self-Attention)**: 帮助encoder关注输入句子中的其他词语
- **前馈神经网络 (Feed Forward)**: 捕捉输入数据中的复杂关系
- **Encoder-Decoder注意力层 (Encoder-Decoder Attention)**: 帮助decoder关注输入句子中的相关部分

# Transformer结构拆解: Word Embedding



- **词嵌入 Word Embedding:**  
使用词嵌入算法 (如word2vec) 将每个输入词语转化为向量

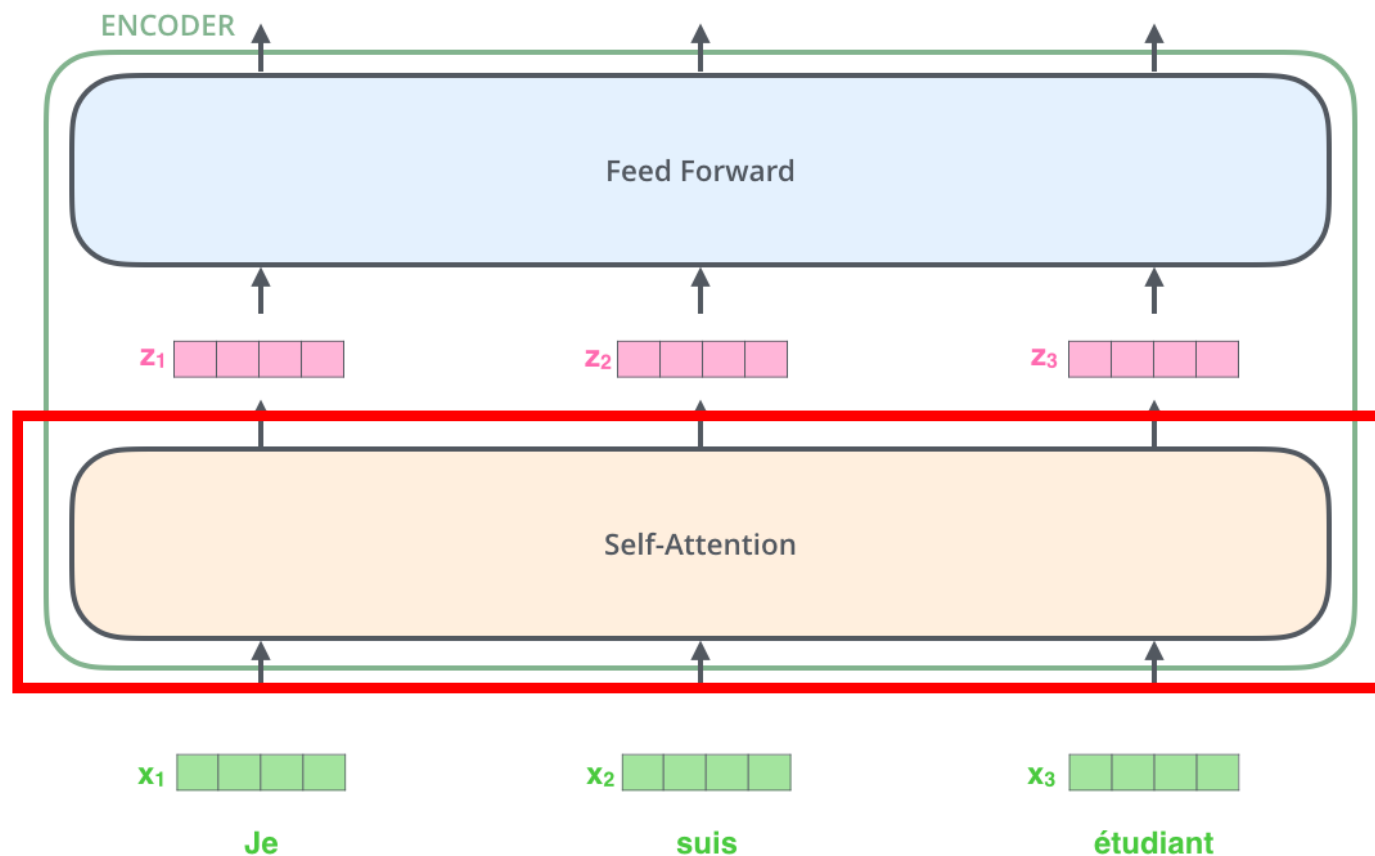


# Transformer结构拆解: Self-Attention



- **自注意力 Self-Attention:**

自注意力机制使模型能够查看输入序列中的其他位置，以获取有助于更好地编码该词的线索

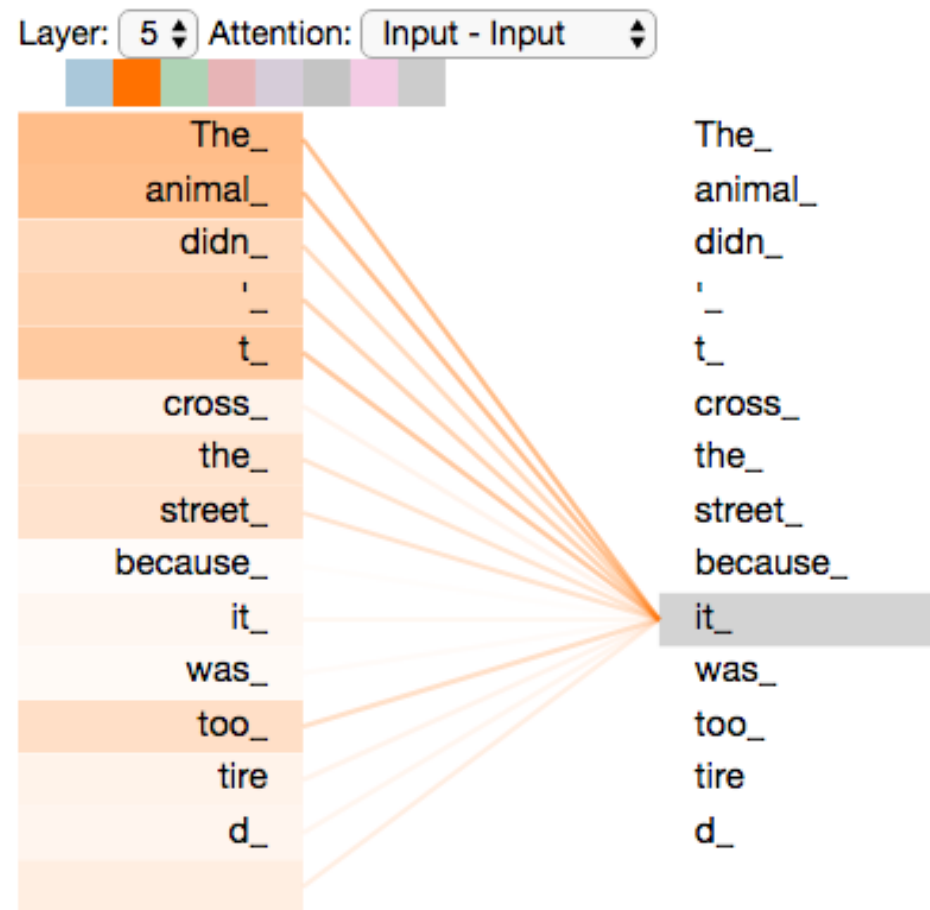


# Transformer结构拆解: Self-Attention



*"The animal didn't cross the street because **it** was too tired"*

- 这句话中的 “it” 指的是什么？
- 是指 “street” 还是 “animal” ？
- 当模型处理 “it” 这个词时，自注意力机制会让模型把 “it” 和 “animal” 联系起来



# Transformer结构拆解: Self-Attention



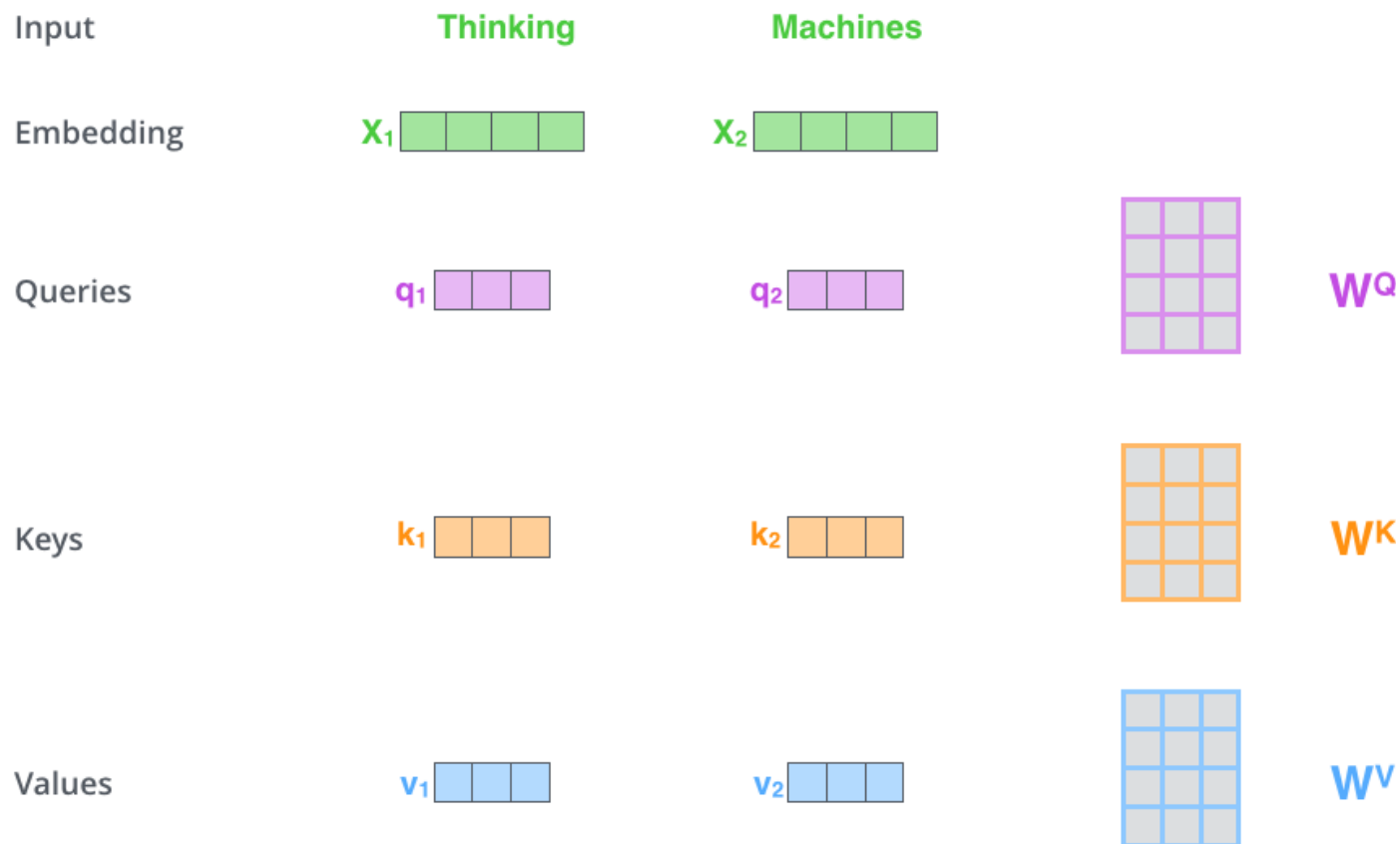
## Step 1

- 创建三个向量 (Query, Key and Value)
- 这些向量由词嵌入分别与训练得到的三个权重矩阵 ( $W^Q$ ,  $W^K$ ,  $W^V$ ) 相乘得到

$$q_i = x_i W^Q$$

$$k_i = x_i W^K$$

$$v_i = x_i W^V$$

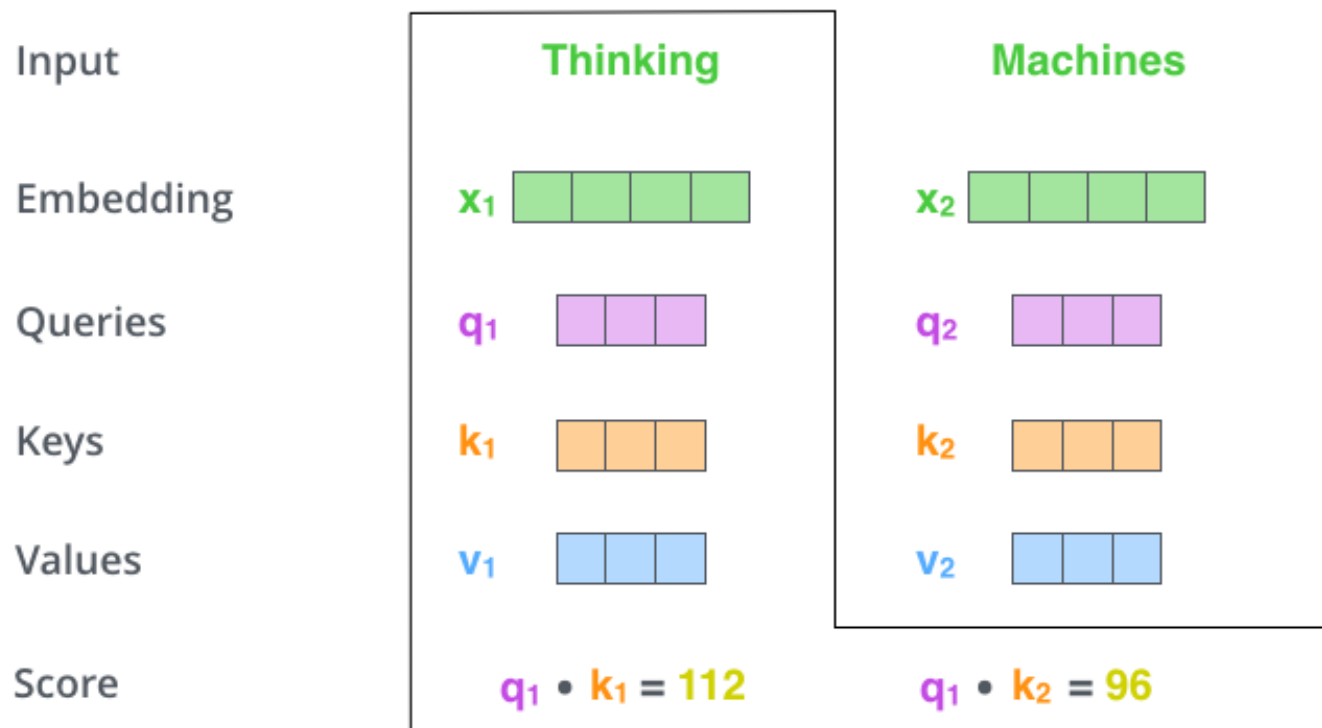


# Transformer结构拆解: Self-Attention



## Step 2

- 计算注意力得分
- 将当前单词的query与句中对应的所有key进行点乘，得到注意力得分
- 这个分数决定了我们进行编码时，当前单词对输入句子各个部分的关注程度



# Transformer结构拆解: Self-Attention



## Step 3

- 将分数除以向量维度的平方根( $\sqrt{d_k}$ )

## Step 4

- 计算分数的Softmax
- Softmax对分数进行归一化处理, 得到各个value的权重 (正值, 和为1)

### Quick Statistics Review:

- Mean of sum = sum of means =  $d_k * 0 = 0$
- Variance of sum = sum of variances =  $d_k * 1 = d_k$
- To set the variance to 1, simply divide by  $\sqrt{d_k}$ !

Input

Embedding

Queries

Keys

Values

Score

Divide by 8 ( $\sqrt{d_k}$ )

Softmax

Thinking

$x_1$

$q_1$

$k_1$

$v_1$

$q_1 \cdot k_1 = 112$

14

0.88

Machines

$x_2$

$q_2$

$k_2$

$v_2$

$q_1 \cdot k_2 = 96$

12

0.12

# Transformer结构拆解: Self-Attention



## Step 5

- 根据softmax计算的权重，对句中所有 **value** 加权求和
- 通过乘以较小的权重，忽略不相关的单词
- 将我们想要关注的单词编码到最终的结果中

Input

Embedding

Queries

Keys

Values

Score

Divide by 8 ( $\sqrt{d_k}$ )

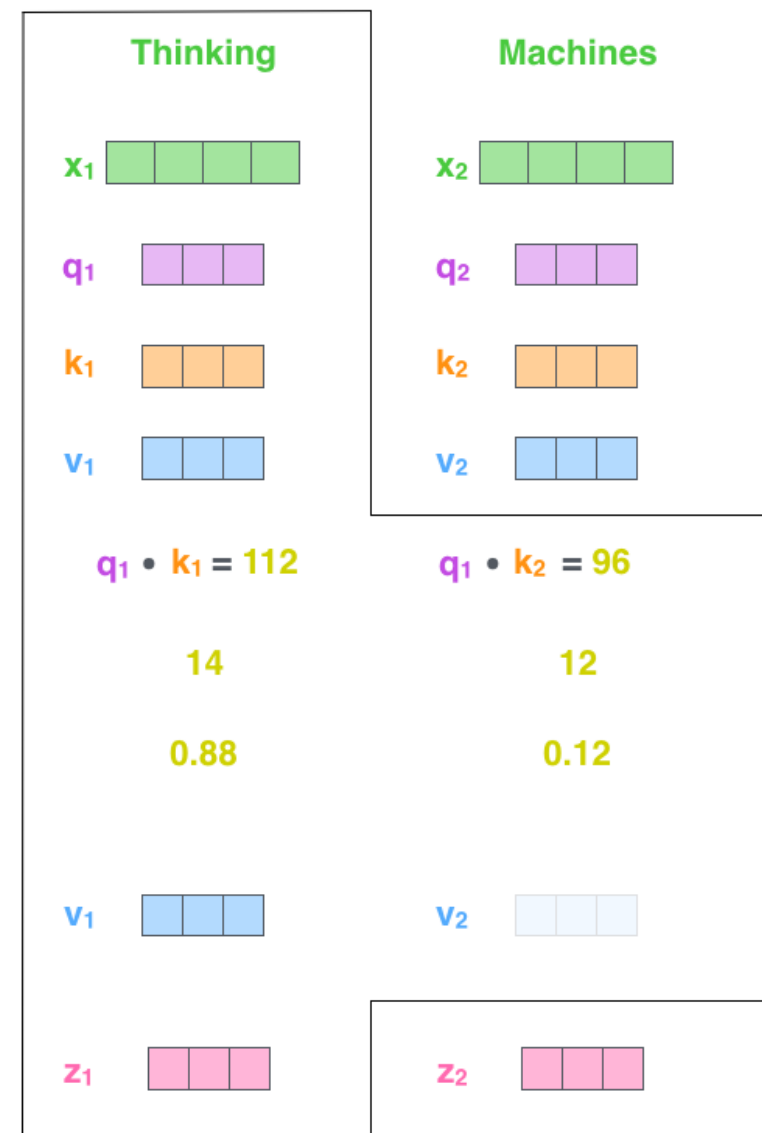
Softmax

Softmax

X

Value

Sum



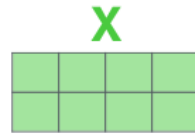
# Transformer结构拆解: Multi-Head Attention



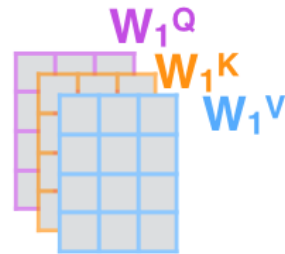
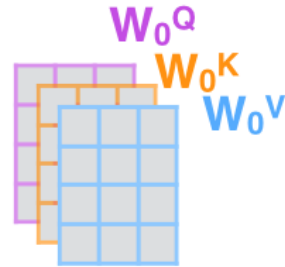
1) This is our input sentence\*

Thinking  
Machines

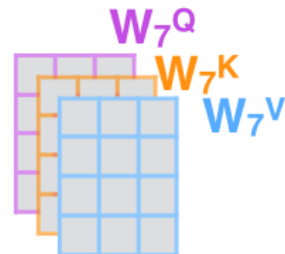
2) We embed each word\*



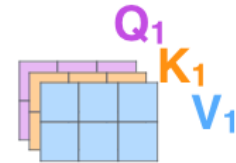
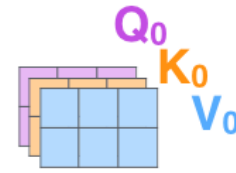
3) Split into 8 heads. We multiply  $X$  or  $R$  with weight matrices



...



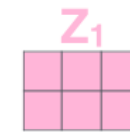
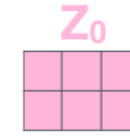
4) Calculate attention using the resulting  $Q/K/V$  matrices



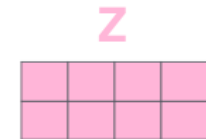
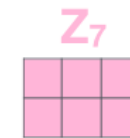
...



5) Concatenate the resulting  $Z$  matrices, then multiply with weight matrix  $W^O$  to produce the output of the layer



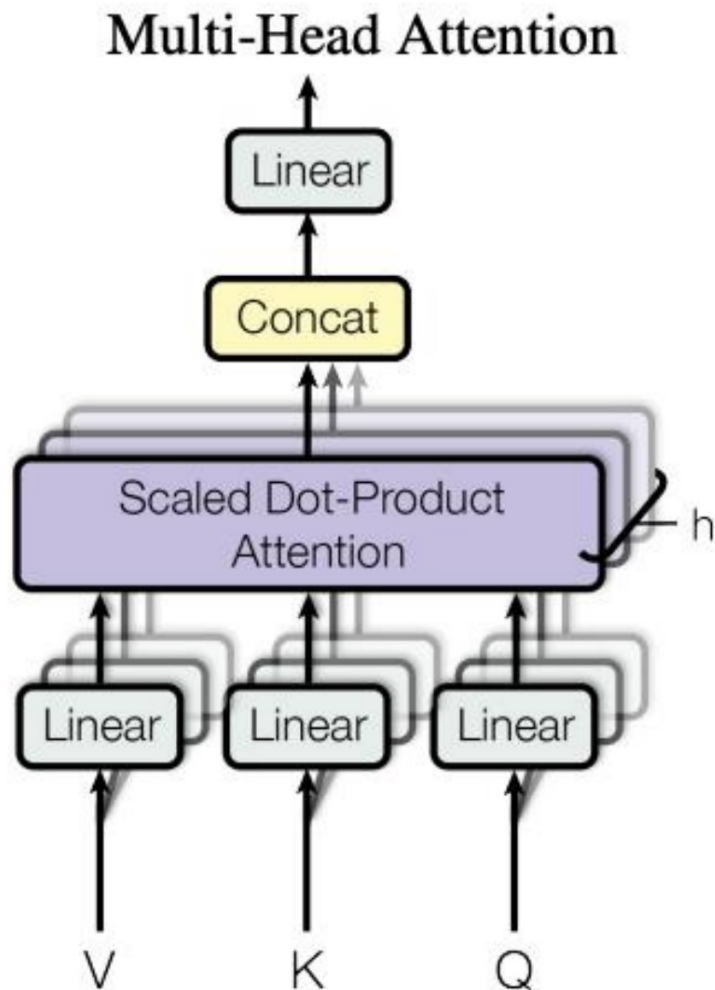
...



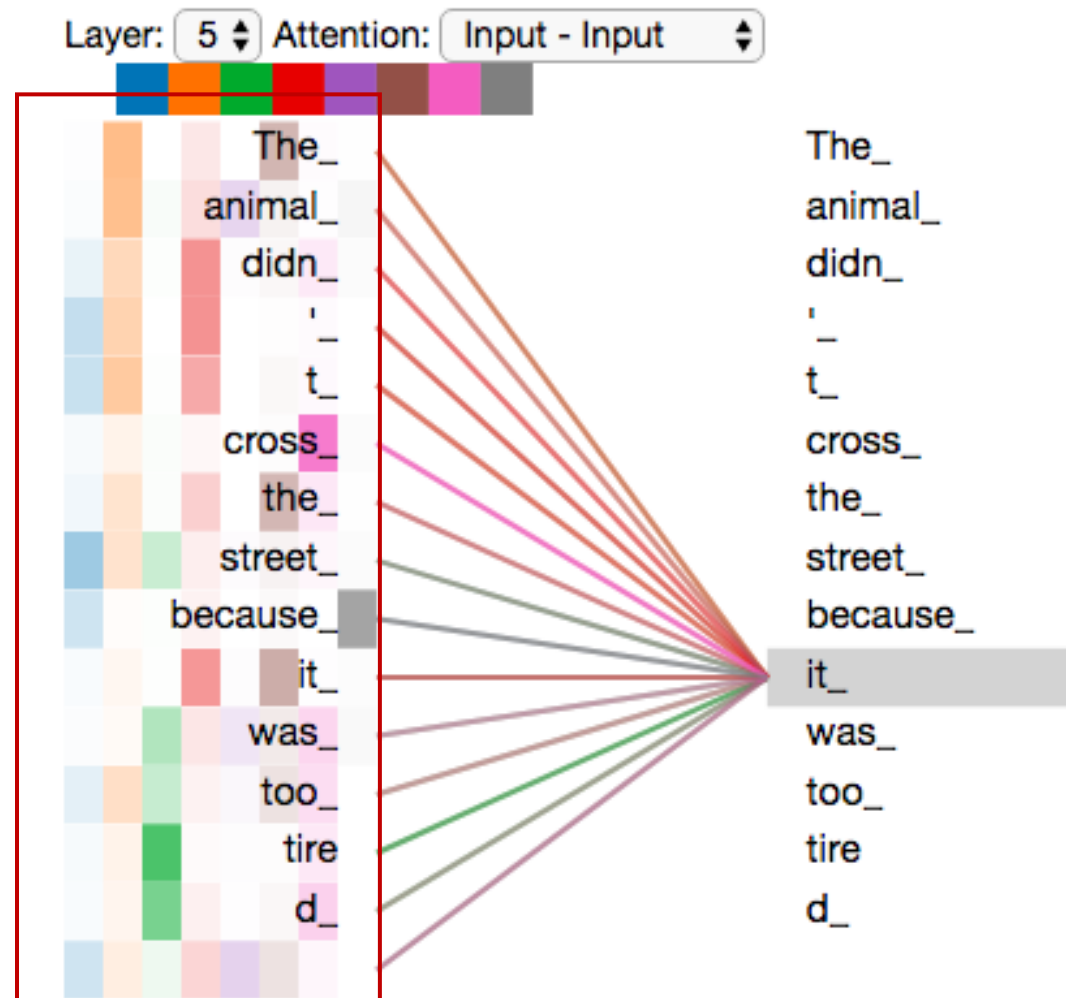
\* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one



# Transformer结构拆解: Multi-Head Attention



[Vaswani et al. 2017]



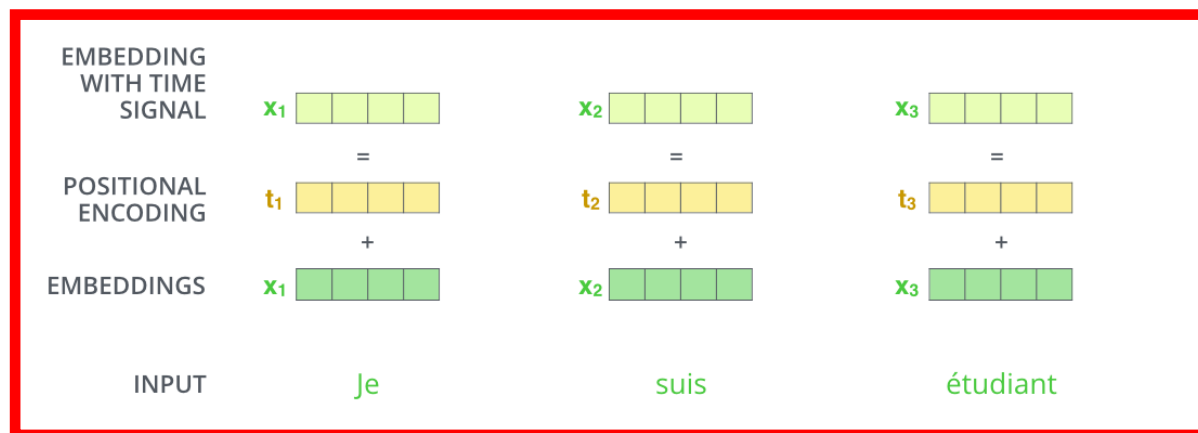
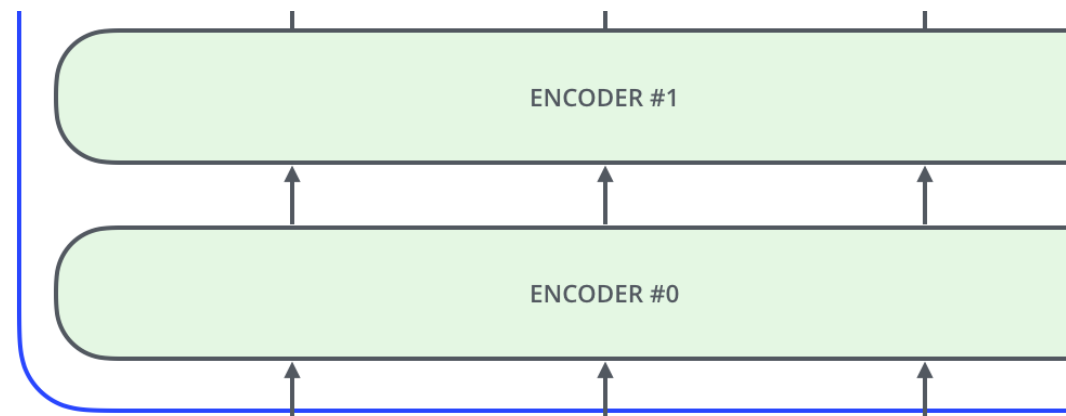
多头自注意力

# Transformer结构拆解: Positional Encoding



- **Positional Encoding:**

由于自注意力机制没有考虑序列的顺序信息，我们需要对句子的顺序进行编码

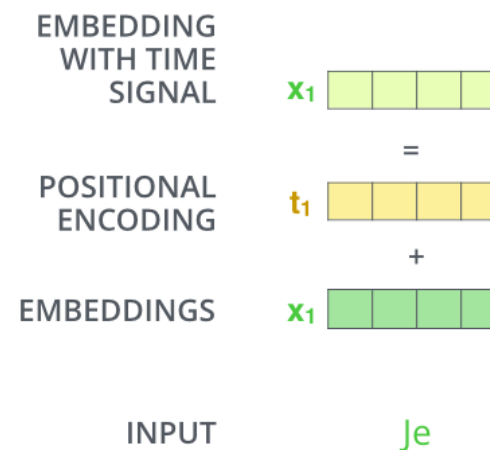


# Transformer结构拆解: Positional Encoding



- 将第 $i$ 个位置的词嵌入加上位置向量 $p_i$
- 这种位置向量需要遵循一定的模型可以学习到的模式，帮助模型确定每个单词的位置或者序列中不同单词之间的距离
- **Key insight:** 最重要的位置信息是单词之间的**相对关系**（如“cat”是“eat”之前的单词），而不是它们的**绝对位置**（如“cat”是句中的第二个单词）。

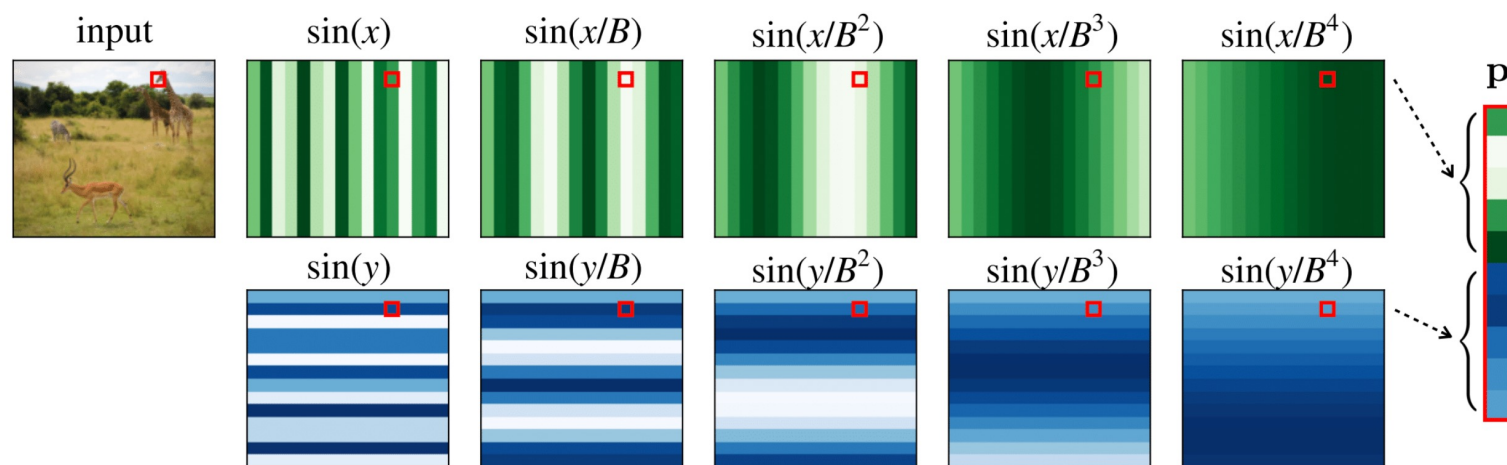
$$p_i = \begin{pmatrix} \sin(i/10000^{2*1/d}) \\ \cos(i/10000^{2*1/d}) \\ \vdots \\ \sin(i/10000^{2*\frac{d}{2}/d}) \\ \cos(i/10000^{2*\frac{d}{2}/d}) \end{pmatrix}$$



# Transformer结构拆解: Positional Encoding



- 本质上是时域信号在傅立叶基上的投影
- 核心原因：神经网络的偏好为低频率函数更容易学习



- 也可以理解为一个旋转矩阵
- 线性变换特性：位置  $pos+k$  的编码可由位置  $pos$  的编码通过旋转矩阵线性变换得到

线性变换特性（旋转矩阵）

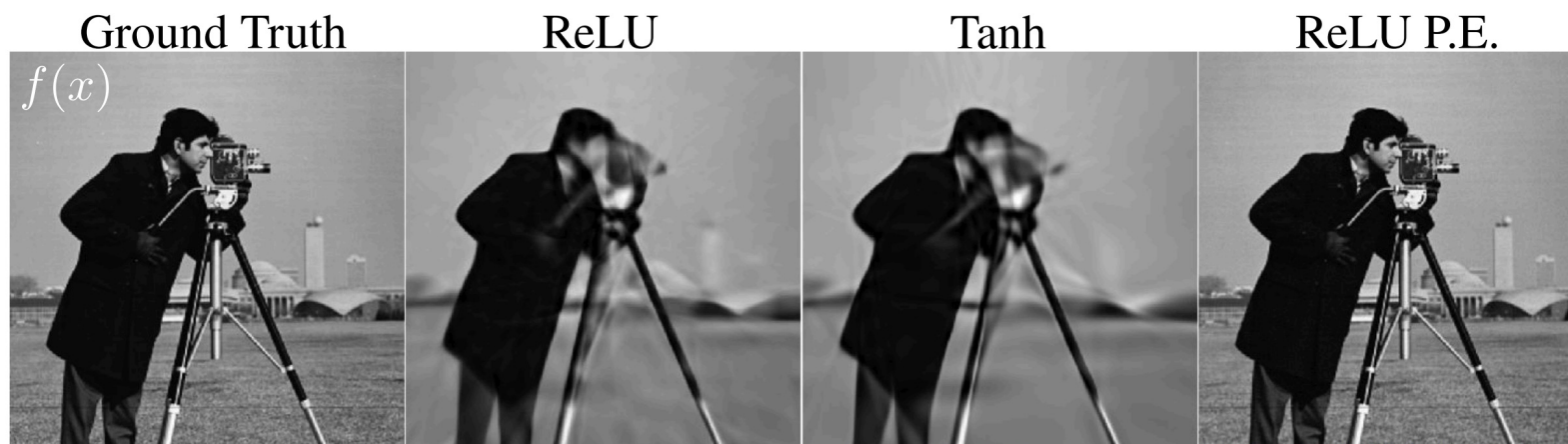
$$\begin{bmatrix} \sin((pos+k)\omega_i) \\ \cos((pos+k)\omega_i) \end{bmatrix} = \begin{bmatrix} \cos(k\omega_i) & -\sin(k\omega_i) \\ \sin(k\omega_i) & \cos(k\omega_i) \end{bmatrix} \cdot \begin{bmatrix} \sin(pos\cdot\omega_i) \\ \cos(pos\cdot\omega_i) \end{bmatrix}$$

其中  $\omega_i = 1/10000^{(2i/d)}$ , 旋转矩阵只依赖于偏移量  $k$ , 与绝对位置无关

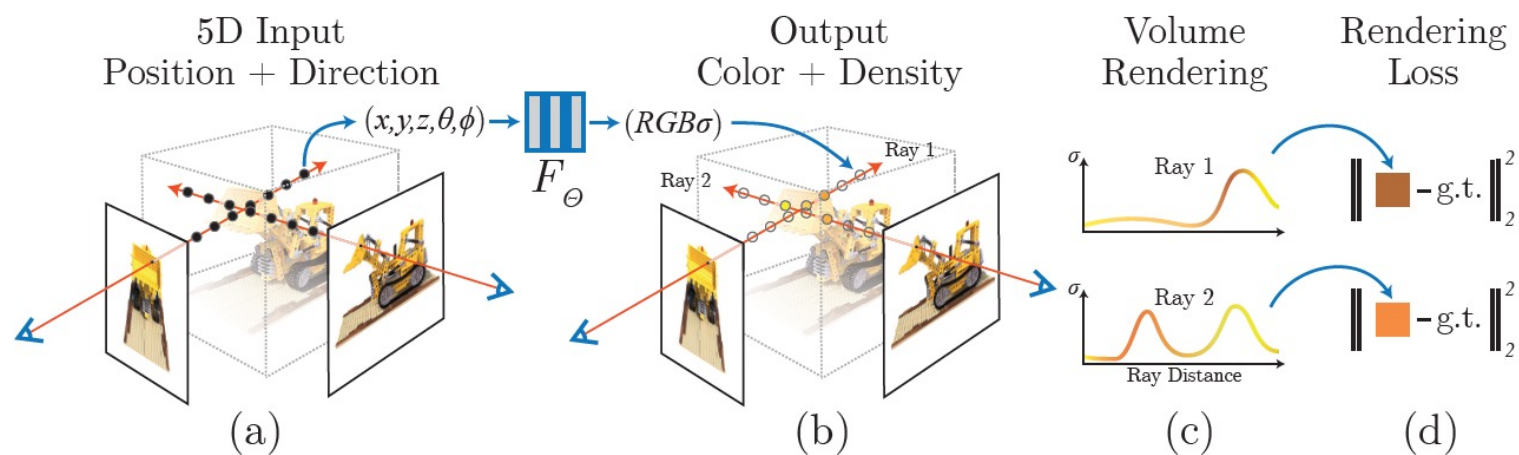
# Transformer结构拆解: Positional Encoding



- 图像重建



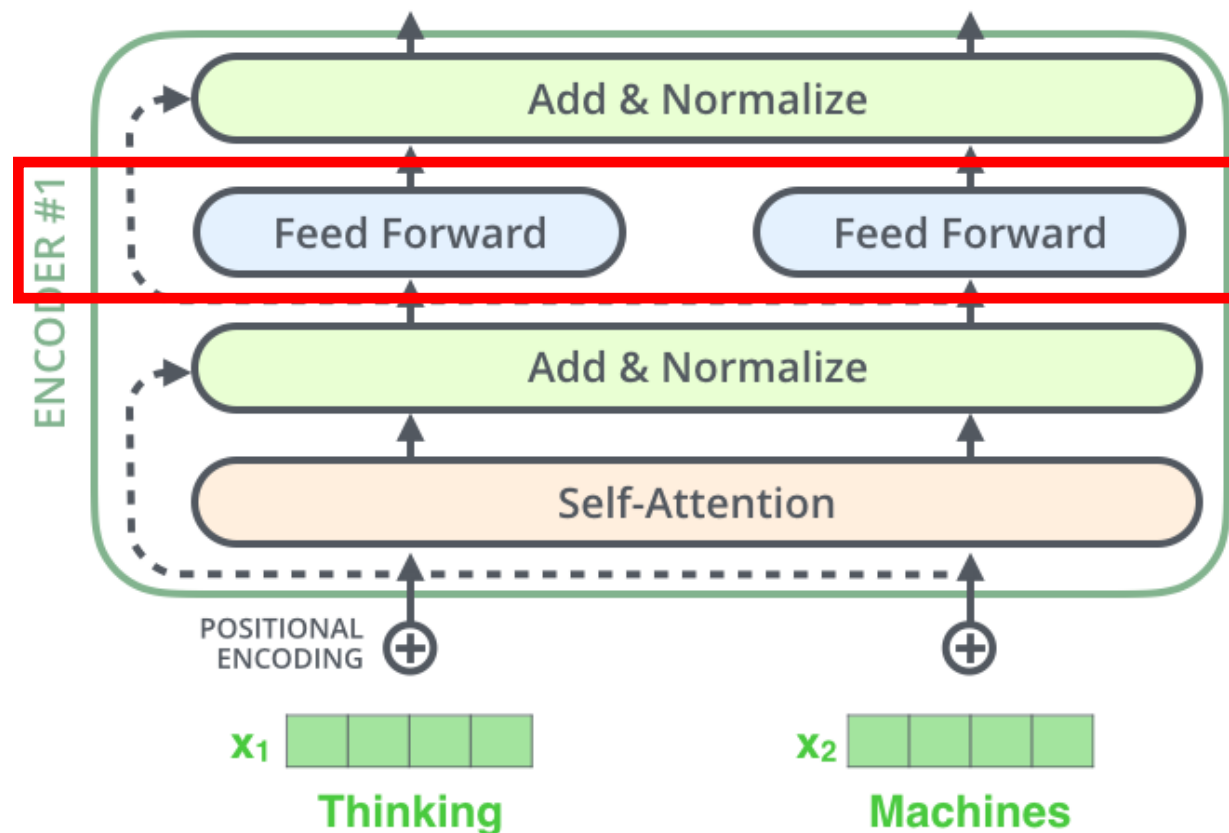
- 三维重建



# Transformer结构拆解: FFN



- 前馈神经网络 **Feed Forward Network**
- 最基础的人工神经网络结构，层与层之间的信息单向流动，无循环或回路
- 对自注意力的输出应用前馈神经网络，  
以提供非线性的激活函数，增强模型  
的表示能力



# Transformer结构拆解: Add & Normalize

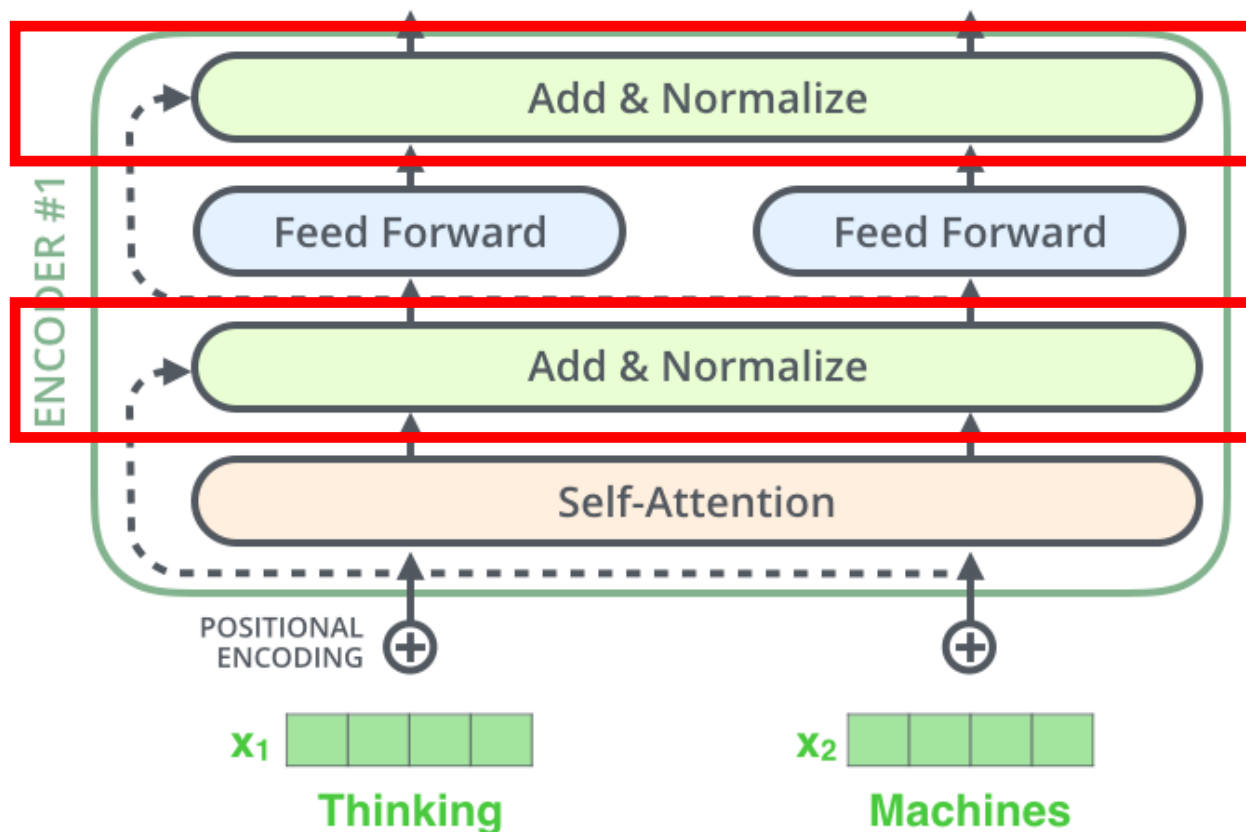


- **Add & Normalize**

Residual Connection and Normalization

- **残差连接 Residual Connection**

直接将“原始”的词嵌入信息传递给下一层，可以防止网络在处理多层信息时“遗忘”或扭曲重要的信息。



# Transformer结构拆解: Add & Normalize

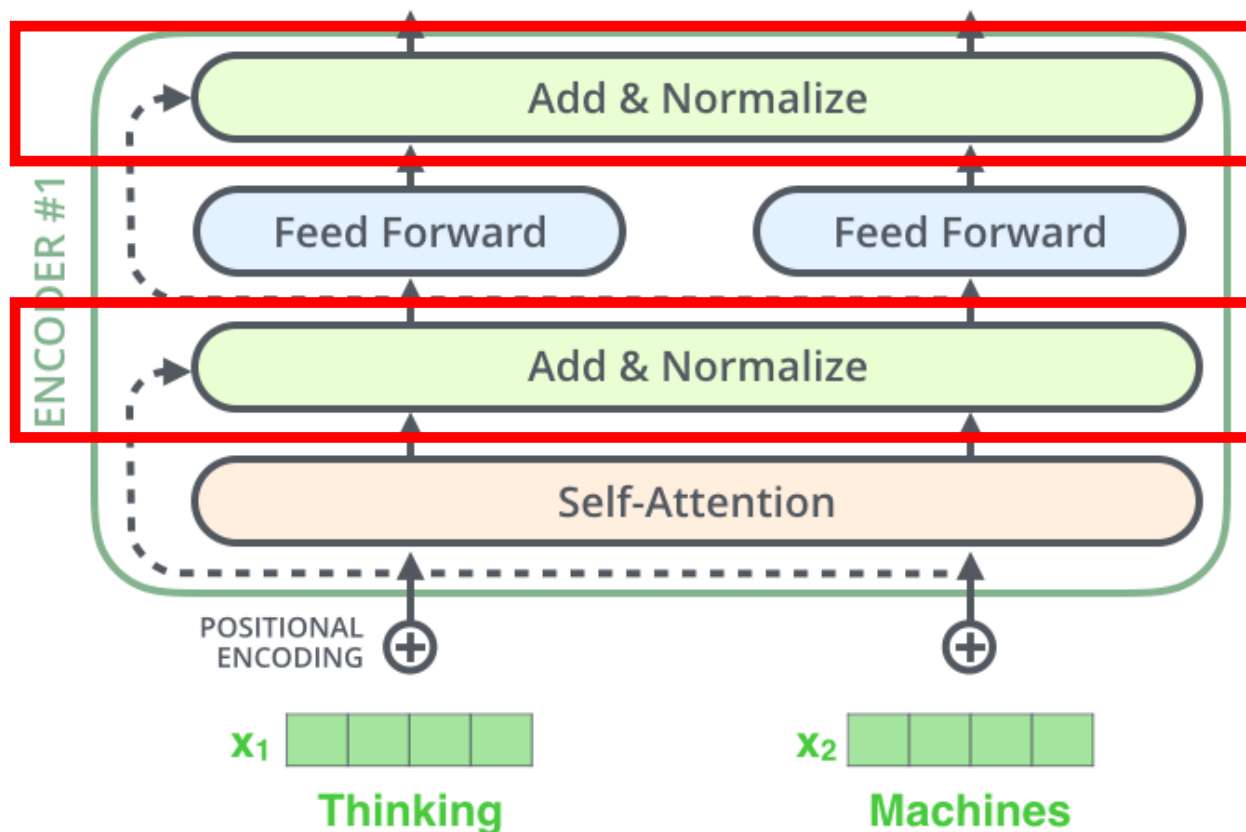


- **Add & Normalize**

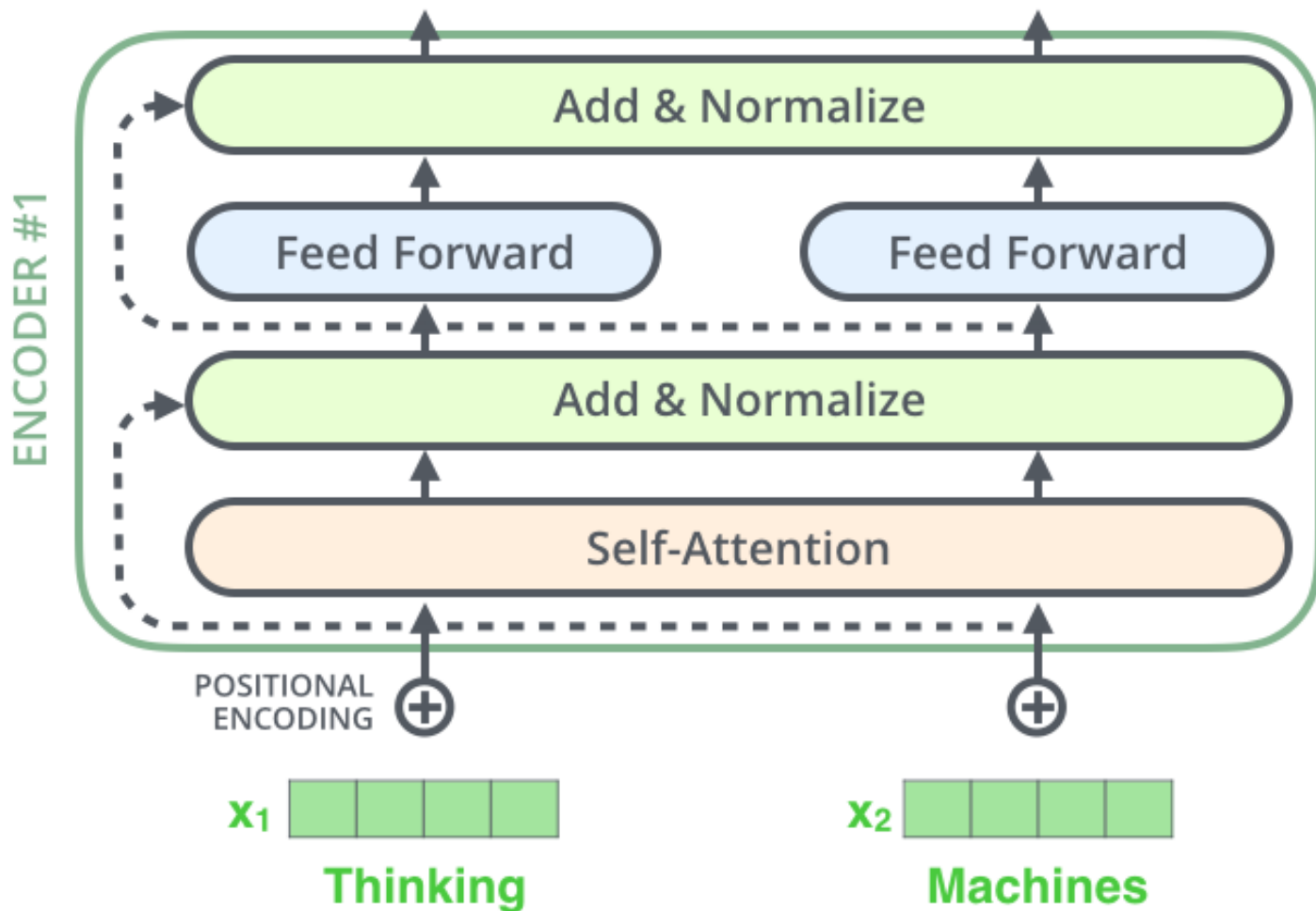
Residual Connection and Normalization

- **层归一化 Layer Normalization**

对每一层的数据进行归一化，使得层归一化后的输出具有零均值和单位标准差



# Transformer: Encoder总结

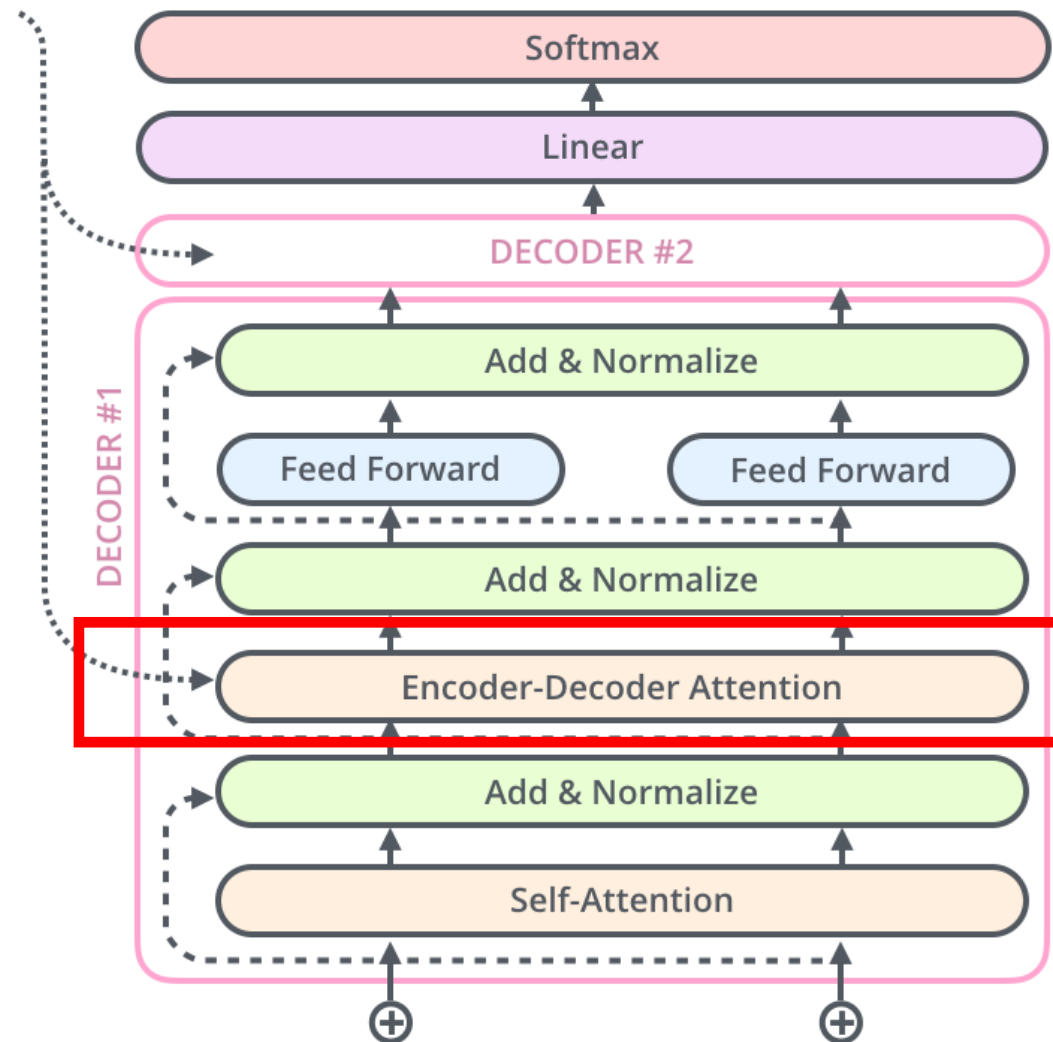
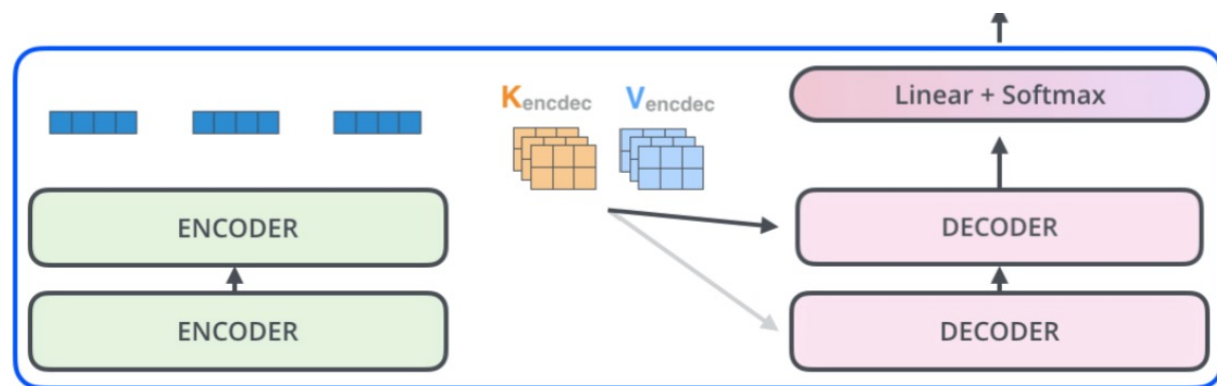


- **Feed Forward Neural Network:** 捕捉输入数据中的复杂关系
- **Add & Normalize:** 提高训练速度和稳定性
- **Self-Attention:** 对词语之间的关系进行编码
- **Positional Encoding:** 对词语的位置进行编码
- **Word Embedding:** 将词语编码为数字

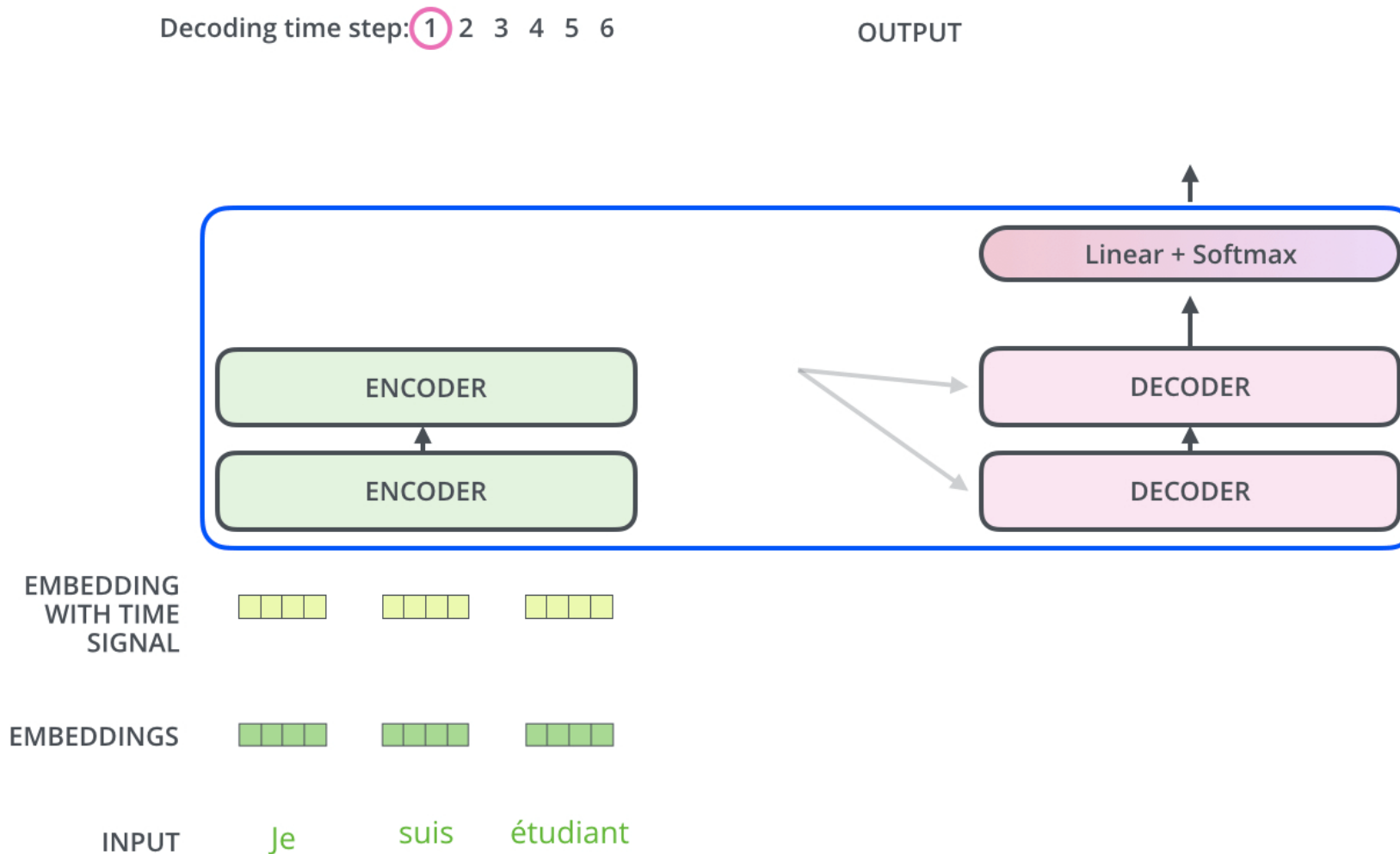
# Transformer结构拆解: Encoder-Decoder Attention



- **Encoder-Decoder Attention**
- decoder端生成自己的Query, encoder端提供Key与Value
- 模型中的每个decoder都将使用encoder端提供的Key与Value, 以帮助decoder专注于输入序列中的适当位置



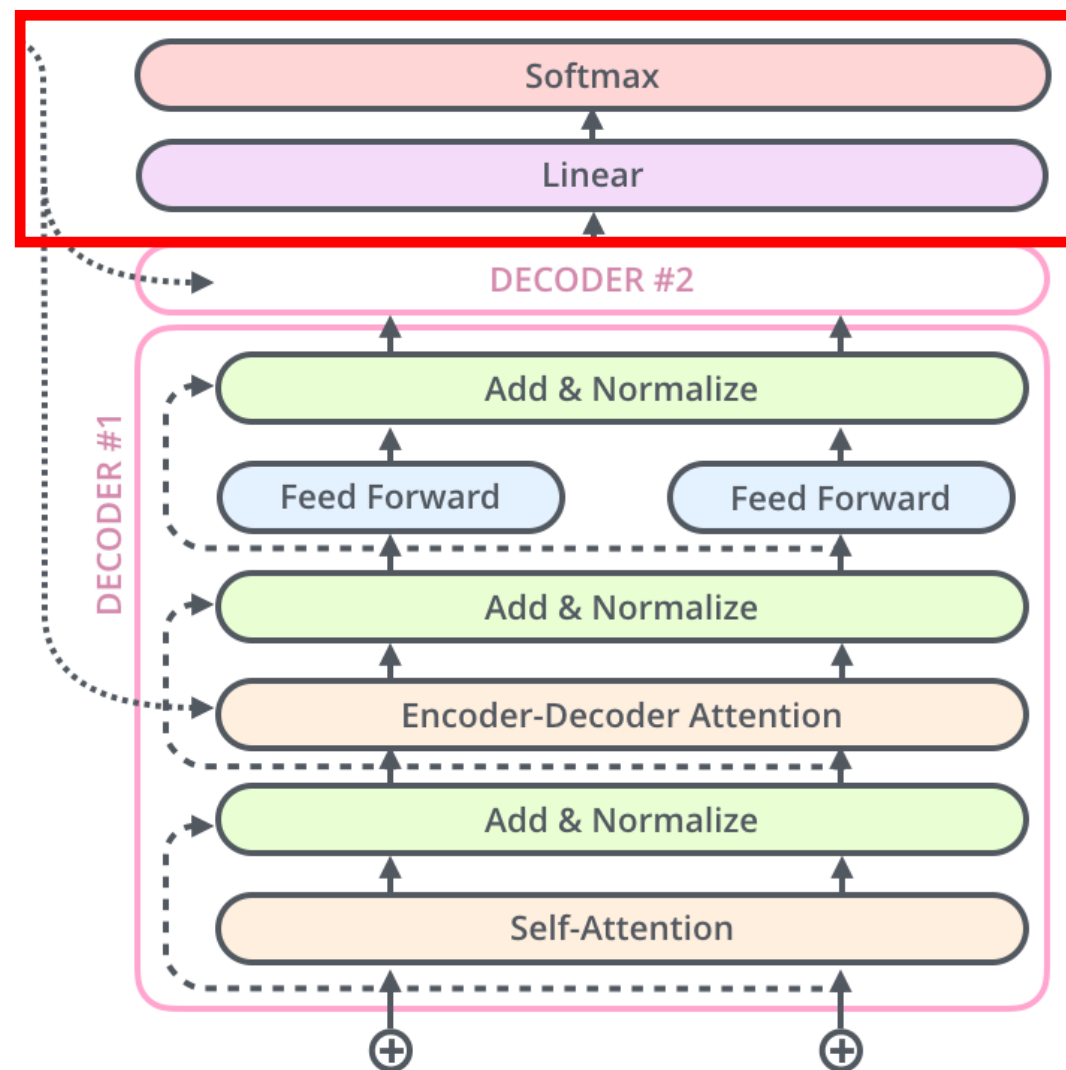
# Transformer结构拆解: Encoder-Decoder Attention



# Transformer结构拆解: Linear and Softmax Layer



- **Linear**
- 简单的全连接神经网络，将decoder产生的向量映射到logits向量中
- 假设我们模型的词汇表包含 10,000 个单词，那么 logits 向量将有 10,000 个单元，每个单元代表一个单词的得分



# Transformer结构拆解: Linear and Softmax Layer



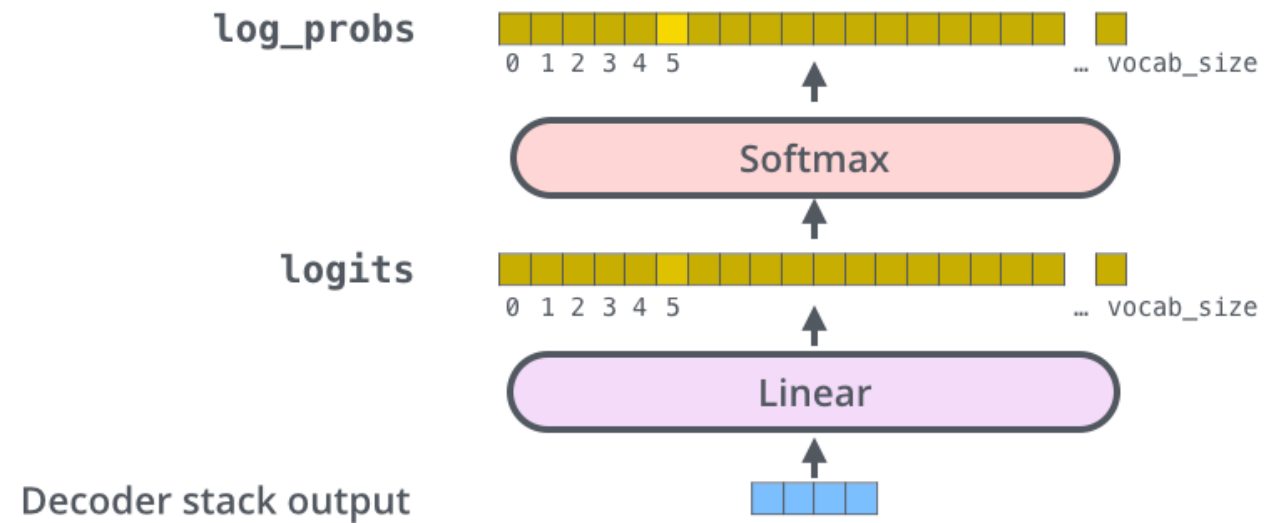
- **Softmax**
- softmax 层将这些分数转化为概率分布 (全部为正值, 和为 1.0)
- 模型选择概率最高的单元格, 并将与之相关的单词作为该时间步骤的输出结果

Which word in our vocabulary is associated with this index?

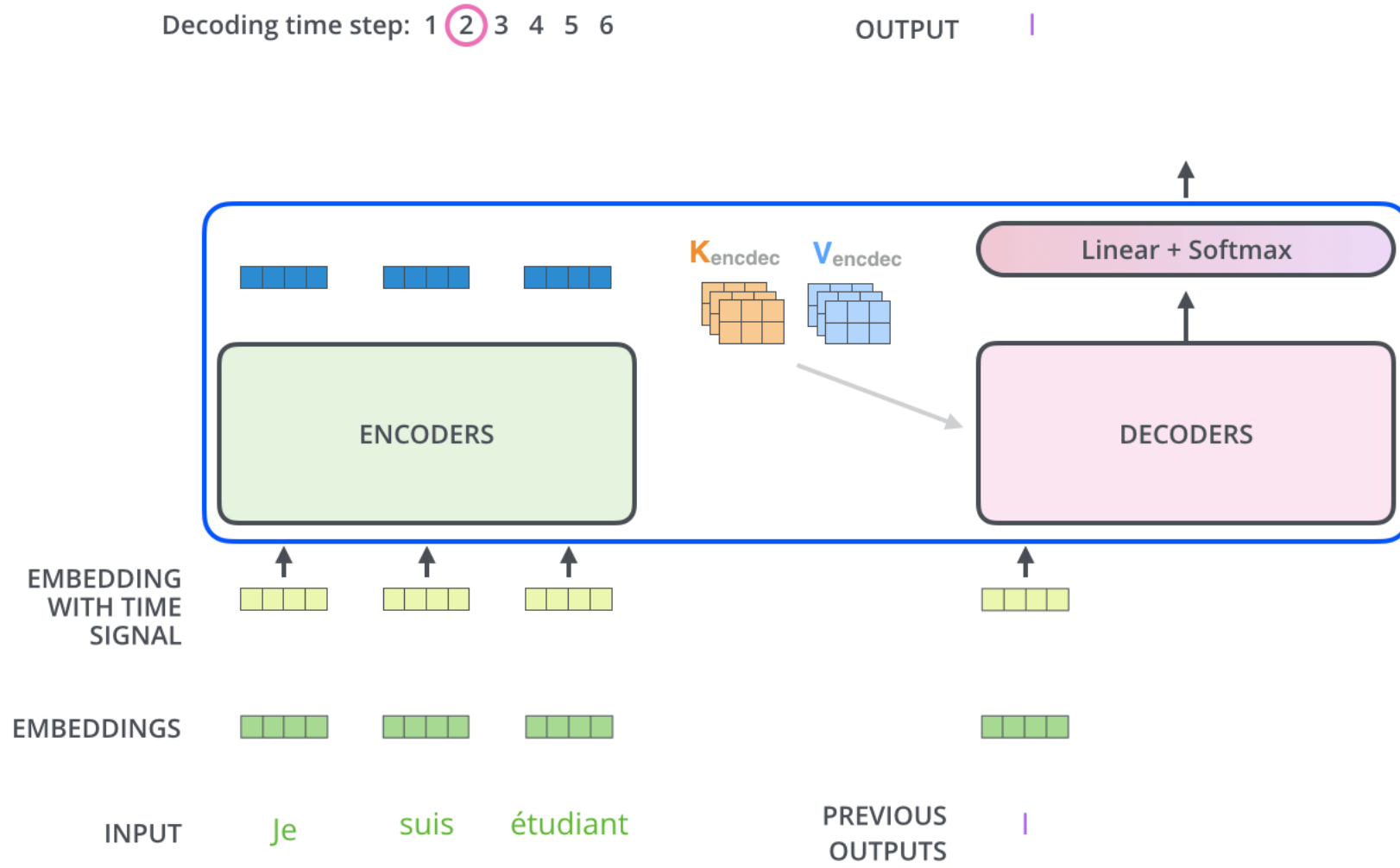
Get the index of the cell with the highest value (argmax)

am

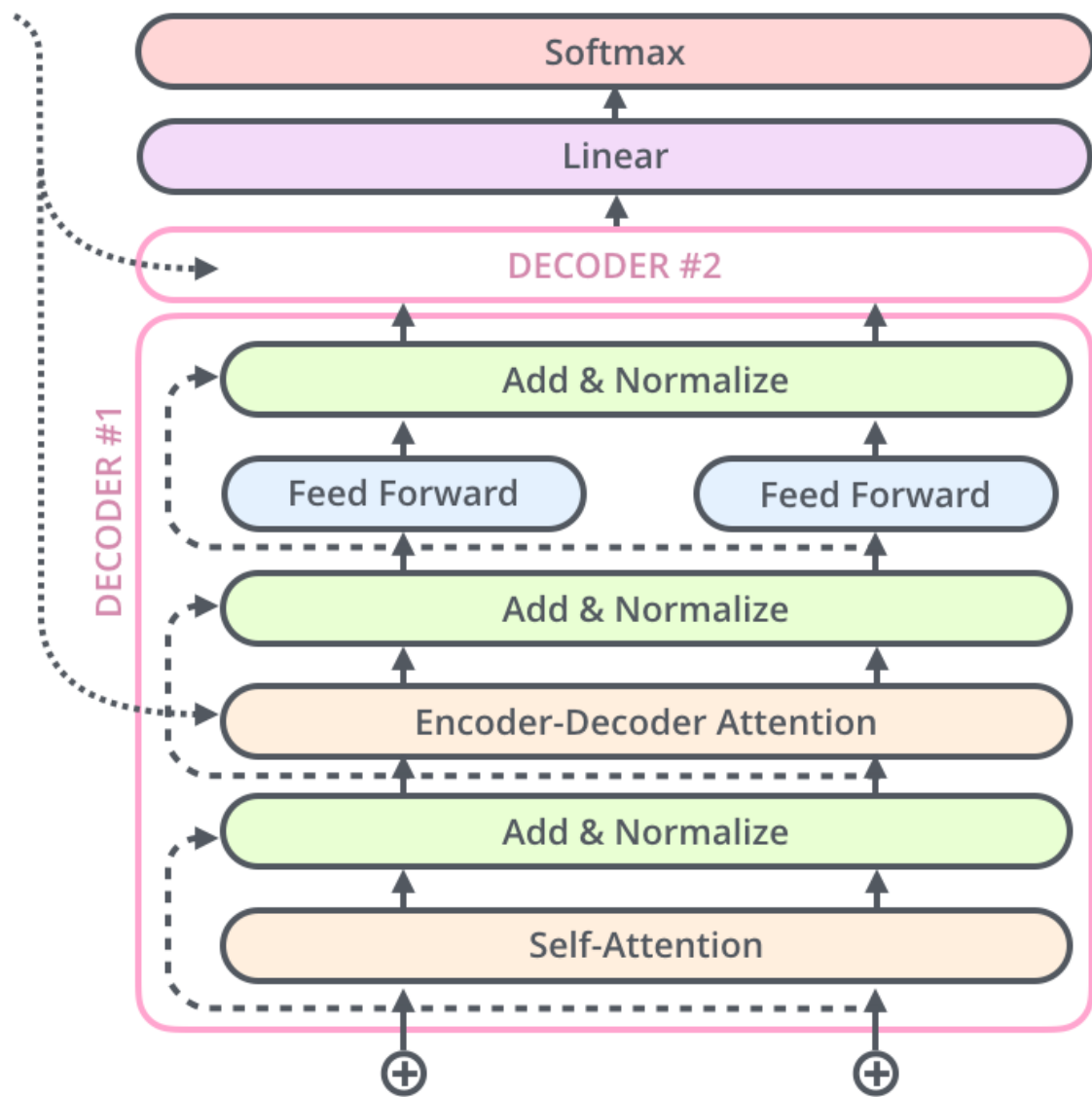
5



# Transformer: Decoder

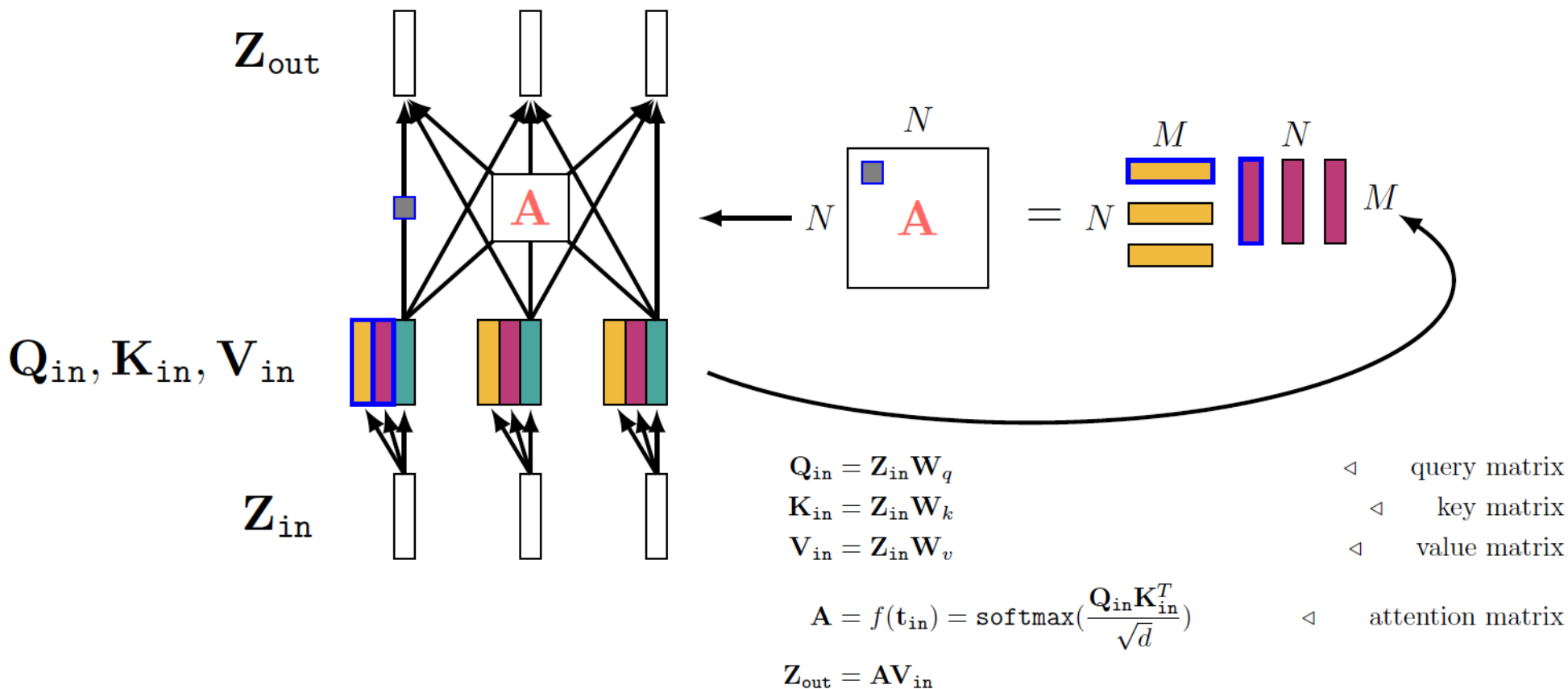


# Transformer: Decoder总结



- **Softmax Layer:** 将分数转化为概率分布
- **Linear Layer:** 将decoder生成的向量映射到 logits向量中, 为输出词汇表中的每个词分配一个分数
- **Encoder-Decoder Attention:** 追踪输入中的重要信息
- **Word Embedding, Positional Encoding, Self-Attention, ...**

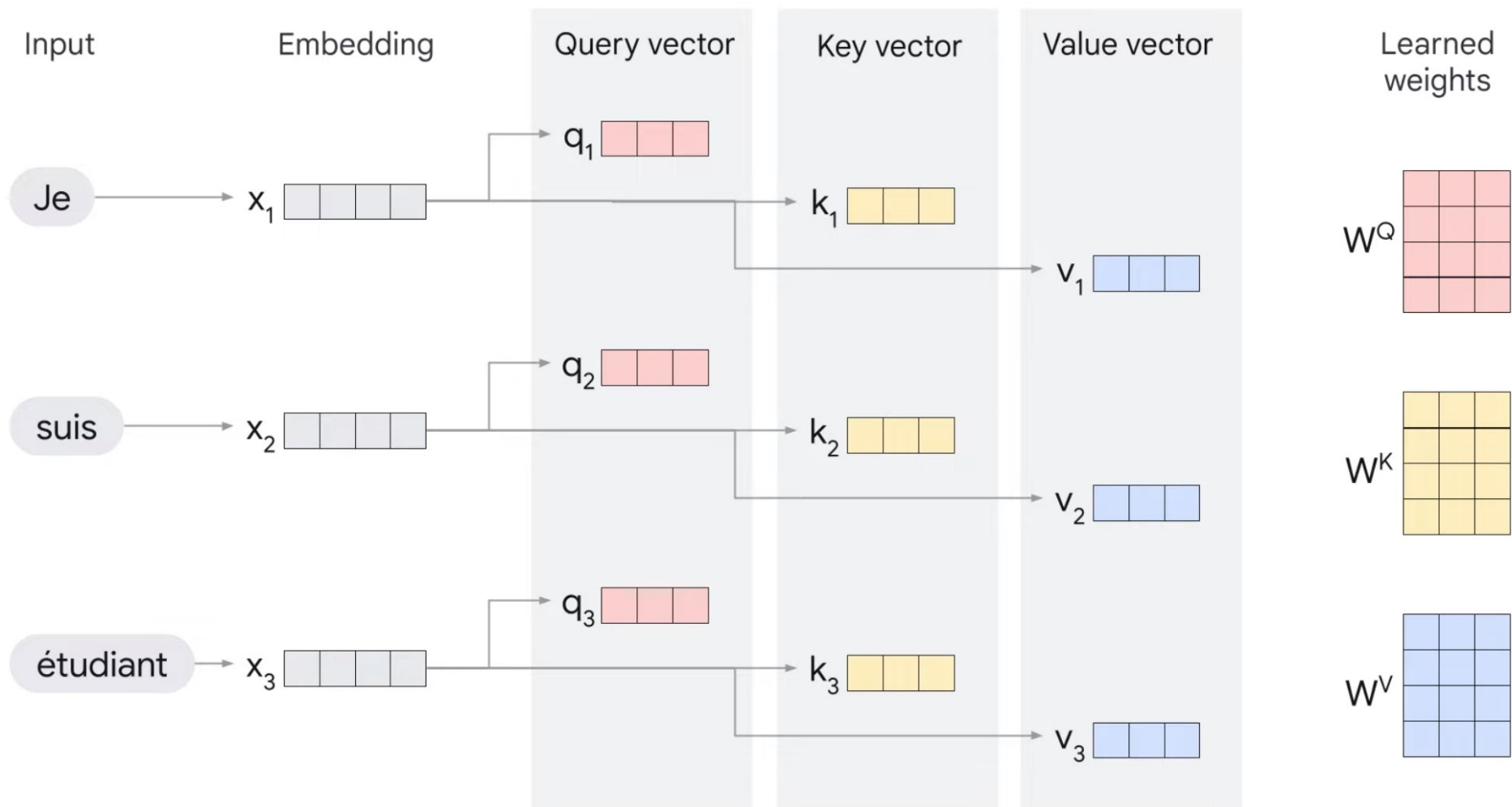
# Self-Attention计算公式



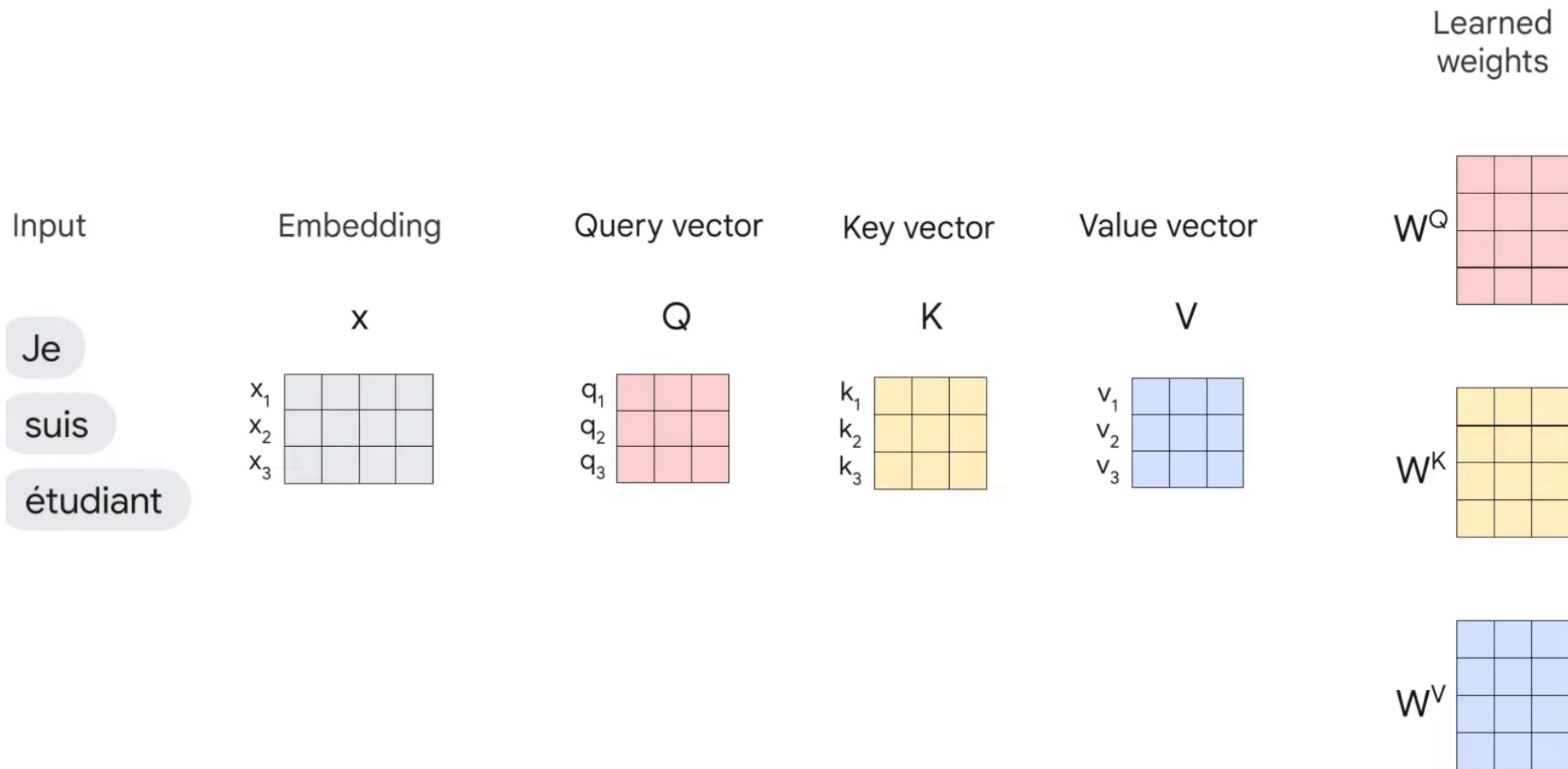
## 几个关键点:

- 计算顺序: Q、K、V
- 归一化: 分母上是sqr(d)
- attention score是用softmax算出来的

# Self-Attention计算公式



# Self-Attention计算公式



以Tensor的形式并行执行

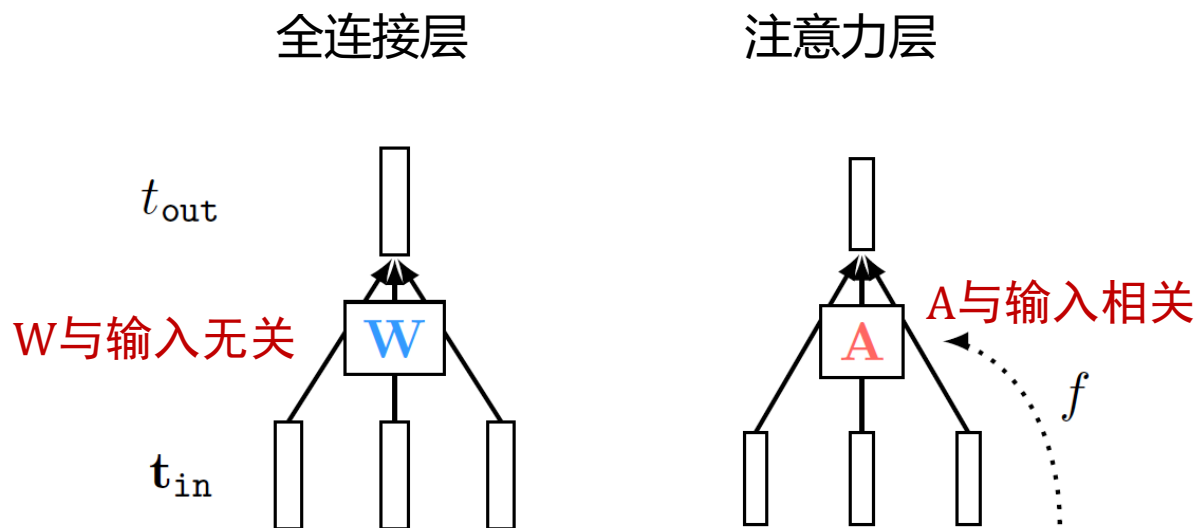
# Self-Attention计算公式



$$\text{softmax} \left( \frac{Q \times K^T}{\sqrt{d_k}} \right) \times V = Z$$

The diagram illustrates the self-attention calculation process. It shows a red 3x3 matrix labeled 'Q' multiplied by a yellow 3x3 matrix labeled 'K<sup>T</sup>'. The result of this multiplication is a blue 3x3 matrix labeled 'V'. This result is then multiplied by a scalar value  $\sqrt{d_k}$  (indicated by a horizontal line and a square root symbol) to produce the final output matrix 'Z', which is a gray 3x3 matrix. The 'softmax' function is applied to the result of the matrix multiplication before it is multiplied by 'V'.

# Self-Attention的理解



$$Q_{in} = Z_{in} W_q \quad \triangleleft \text{query matrix}$$
$$K_{in} = Z_{in} W_k \quad \triangleleft \text{key matrix}$$
$$V_{in} = Z_{in} W_v \quad \triangleleft \text{value matrix}$$

$$A = f(t_{in}) = \text{softmax}\left(\frac{Q_{in} K_{in}^T}{\sqrt{d}}\right) \quad \triangleleft \text{attention matrix}$$

$$Z_{out} = AV_{in}$$

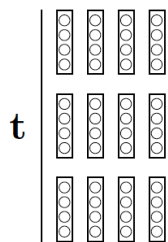
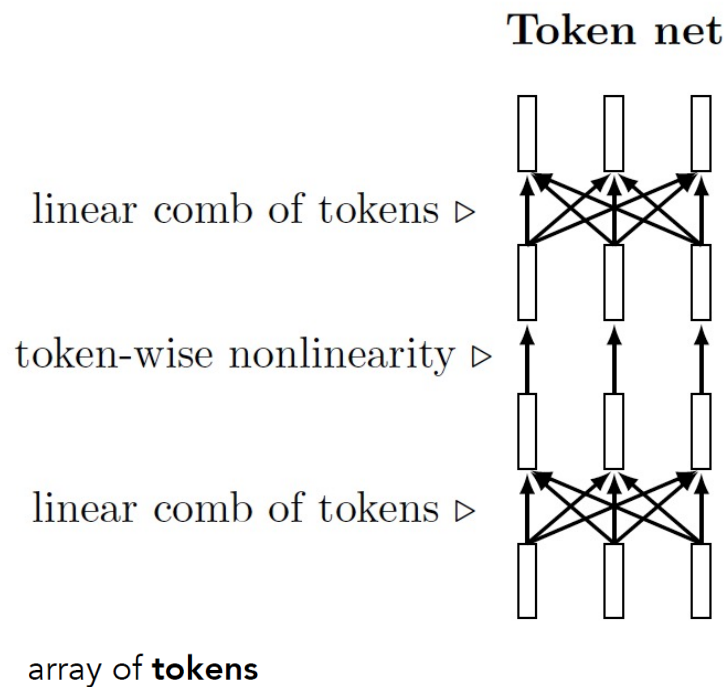
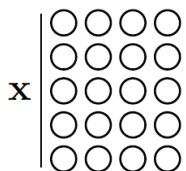
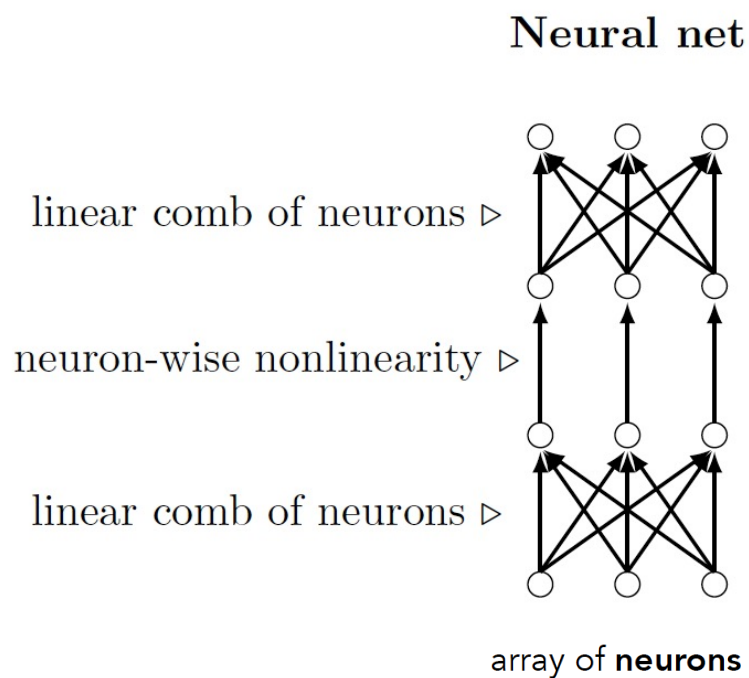
相当于在 $WX$ 的基础上乘了一个attention权重 $A$ ，其中 $A$ 只与输入 $X$ 自己有关（即句子 $X$ 中各单词间的自注意力）

$V$ 相当于全连接层的 $WX$

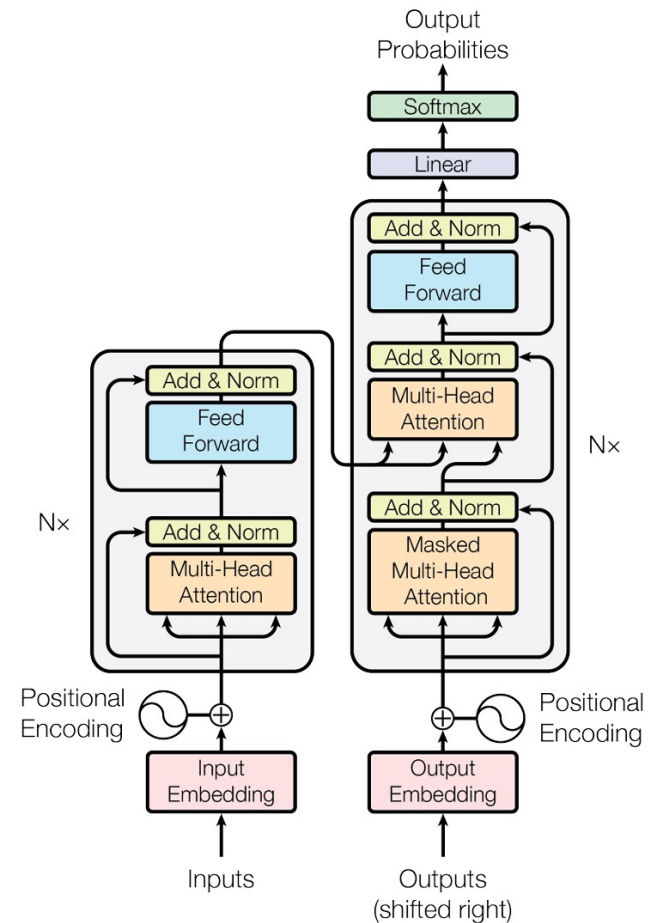
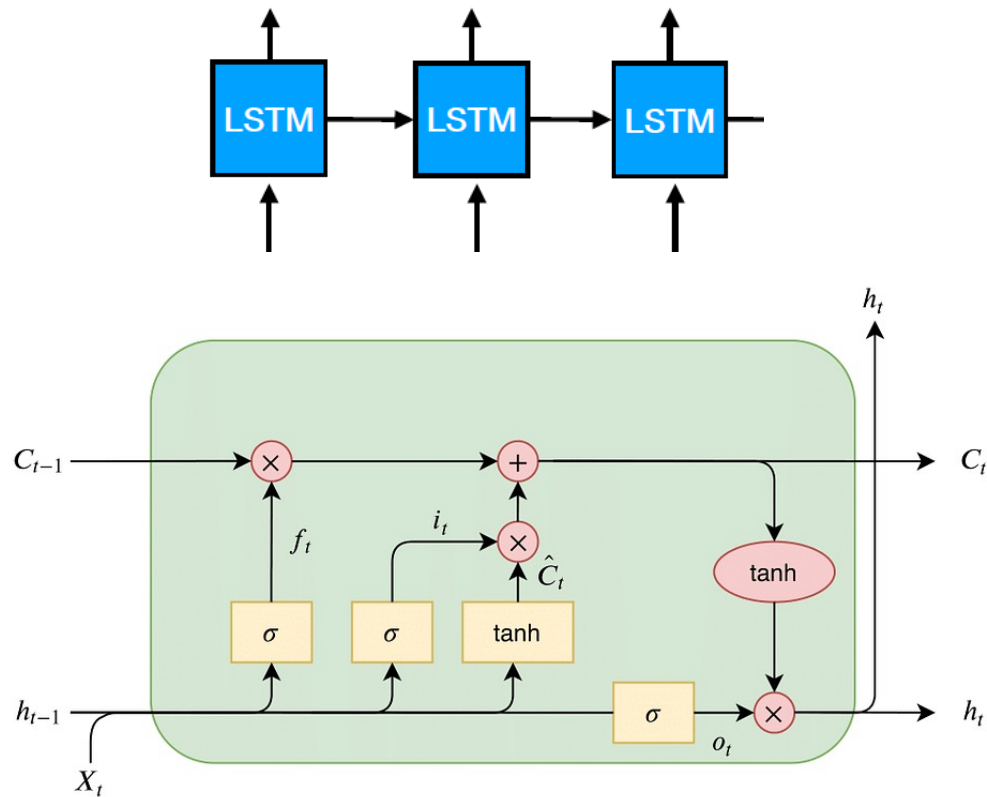
$W$  is free parameters.

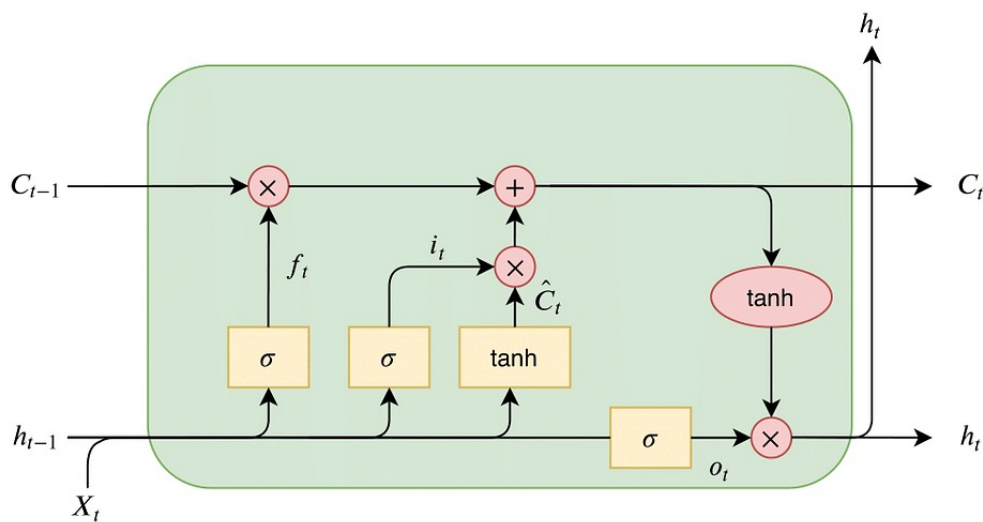
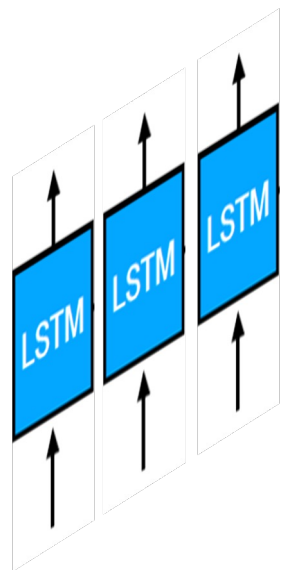
$A$  is a function of some input data. The data tells us which tokens to attend to (assign high weight in weighted sum)

## Transformer是一种Token网络



Transformer既有LSTM的影子也有CNN和RNN的影子？  
甚至还有MLP的影子？



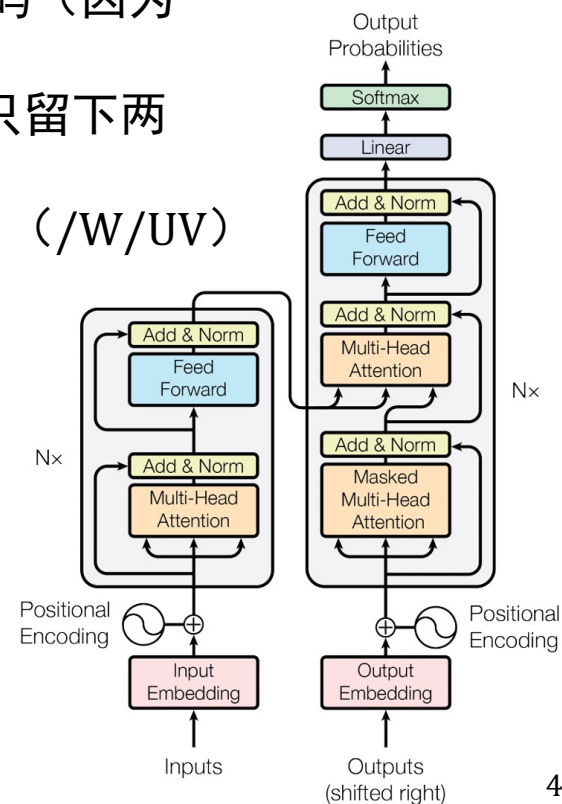


## 从MLP角度：

- Transformer很像MLP

## 从LSTM 到Transformer：

- 串行改并行，单头改多头
- 增加cosine/sine 位置编码（因为没有顺序输入了）
- 减少了内部门的数量，只留下两个门
- 都有三套参数， $Wq/k/v$ （ $/W/UV$ ）



2. (5分) 在 Transformer 架构的设计中, 以下哪一项不是其基本组成部分? ( )
- A. 自注意力机制
  - B. 残差连接
  - C. 前馈神经网络
  - D. 循环神经网络

2. (5分) 在 Transformer 架构的设计中, 以下哪一项不是其基本组成部分? ( )
- A. 自注意力机制
  - B. 残差连接
  - C. 前馈神经网络
  - D. 循环神经网络

答案: D

4. (5分) 在使用机器翻译服务时，需要翻译一个长句：“那位昨天在会议上发言的专家，尽管旅途疲惫，但她的报告内容非常精彩。” 为了准确翻译，模型必须理解句末的“她”指代的是句首的“专家”，而不是“会议”。传统的循环神经网络(RNN)在处理这种长距离依赖时常常会“遗忘”。Transformer 模型通过哪项核心机制解决了这个难题？
- A. 自注意力机制 (Self-Attention Mechanism)
  - B. 循环连接 (Recurrent Connections)
  - C. 卷积操作 (Convolutional Operations)
  - D. 池化层 (Pooling Layers)

4. (5分) 在使用机器翻译服务时, 需要翻译一个长句: “那位昨天在会议上发言的专家, 尽管旅途疲惫, 但她的报告内容非常精彩。” 为了准确翻译, 模型必须理解句末的“她”指代的是句首的“专家”, 而不是“会议”。传统的循环神经网络(RNN)在处理这种长距离依赖时常常会“遗忘”。Transformer 模型通过哪项核心机制解决了这个难题?
- A. 自注意力机制 (Self-Attention Mechanism)
  - B. 循环连接 (Recurrent Connections)
  - C. 卷积操作 (Convolutional Operations)
  - D. 池化层 (Pooling Layers)

答案: A

# 目录

1 Transformer介绍

2 Transformer结构

**3 视觉Transformer**

4 语言Transformer

# 视觉Transformer (ViT)



化图像为Token (一个Patch就是一个token)

Published as a conference paper at ICLR 2021

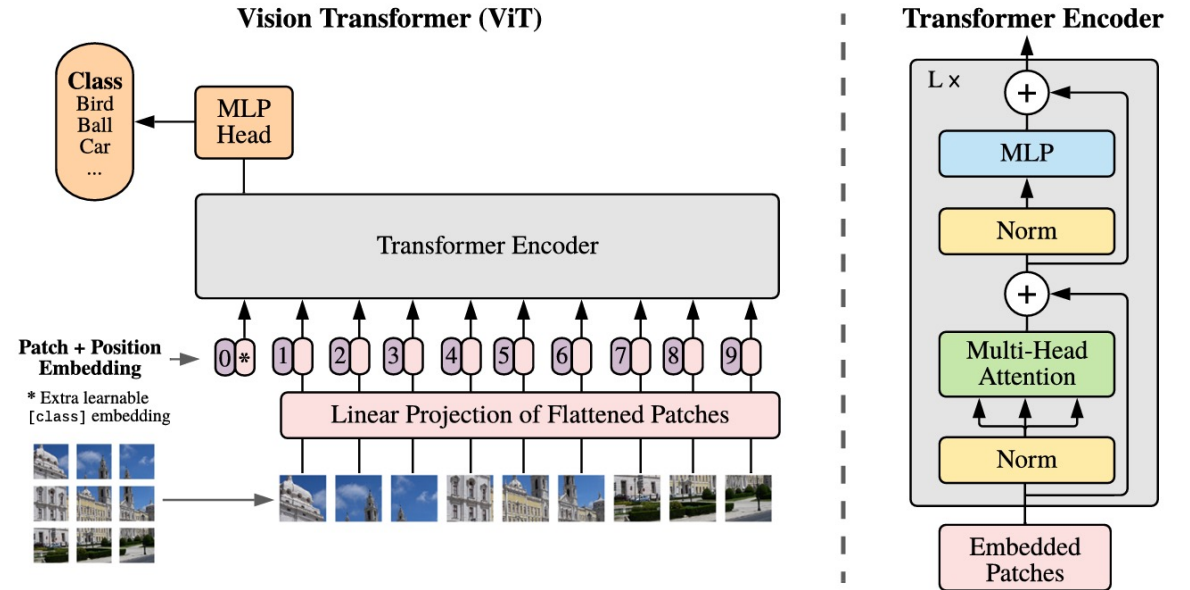
AN IMAGE IS WORTH 16X16 WORDS: **引用4万+**  
TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy<sup>\*,†</sup>, Lucas Beyer<sup>\*</sup>, Alexander Kolesnikov<sup>\*</sup>, Dirk Weissenborn<sup>\*</sup>,  
Xiaohua Zhai<sup>\*</sup>, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,  
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby<sup>\*,†</sup>

<sup>\*</sup>equal technical contribution, <sup>†</sup>equal advising  
Google Research, Brain Team  
{adosovitskiy, neilhoulby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.<sup>1</sup>

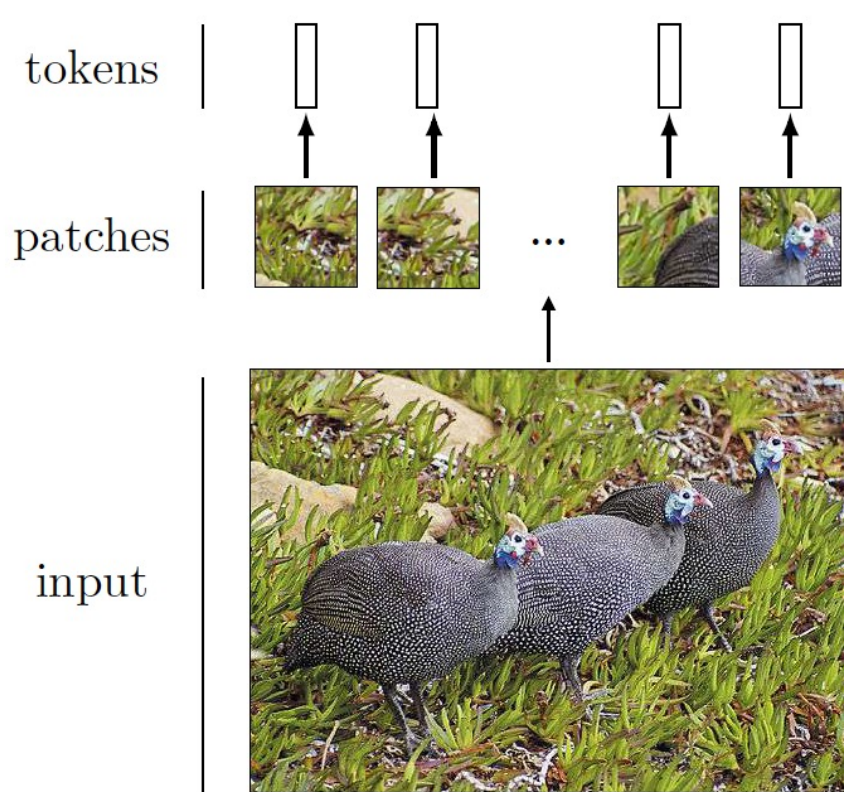


- 文本Transformer的一个很直接的扩展
- 还是来自Google团队
- 横扫各类视觉任务排行榜
- NLP-CV大一统
- 为多模态模型和AGI的到来奠定基础

# 视觉Transformer (ViT)



化图像为Token (一个Patch就是一个token)



e.g., linear projection

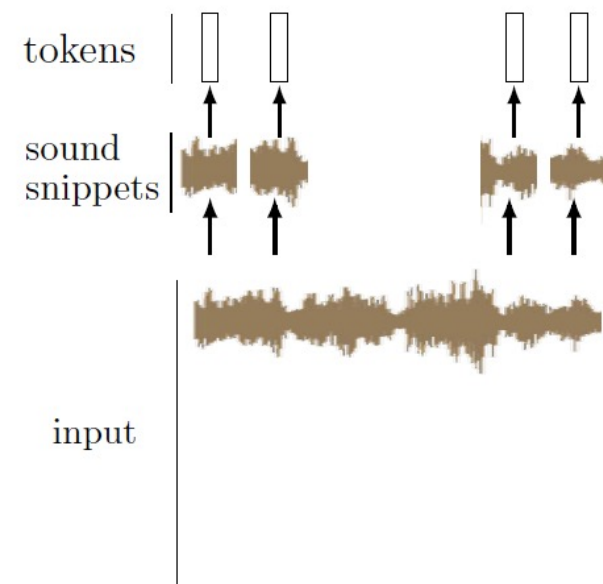
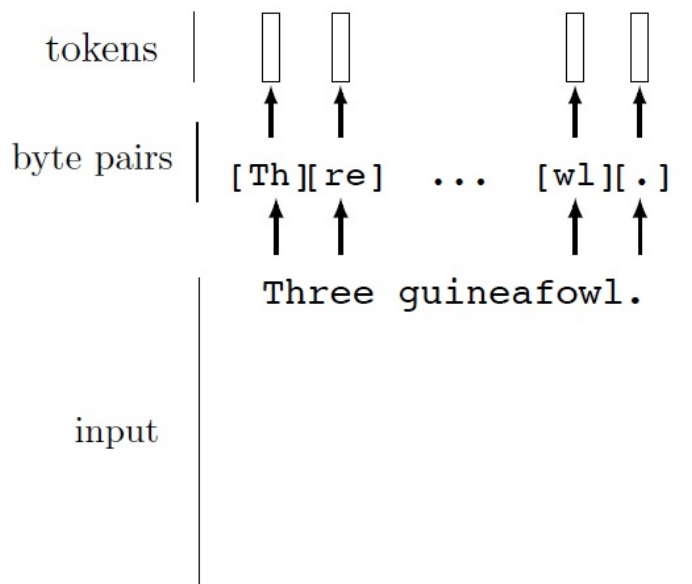
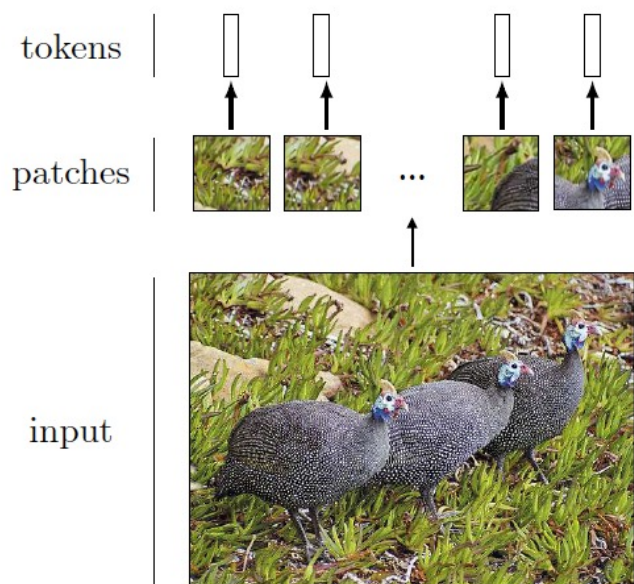
- When operating over *neurons*, we represent the input as an array of scalar-valued measurements (e.g., pixels)
- When operating over *tokens*, we represent the input as an array of vector-valued measurements

# 万物皆可Tokenize (所有任务都可以用Transformer解决)



You can tokenize anything.

General strategy: chop the input up into chunks, project each chunk to a vector.



3. (5分) Transformer 模型的核心机制, 使其在自然语言处理等领域取得巨大成功的是?
- A. 卷积层 (Convolutional Layers)
  - B. 池化层 (Pooling Layers)
  - C. 注意力机制 (Attention Mechanism)
  - D. 梯度下降 (Gradient Descent)

3. (5分) Transformer 模型的核心机制, 使其在自然语言处理等领域取得巨大成功的是?

- A. 卷积层 (Convolutional Layers)
- B. 池化层 (Pooling Layers)
- C. 注意力机制 (Attention Mechanism)
- D. 梯度下降 (Gradient Descent)

答案: C

# 目录

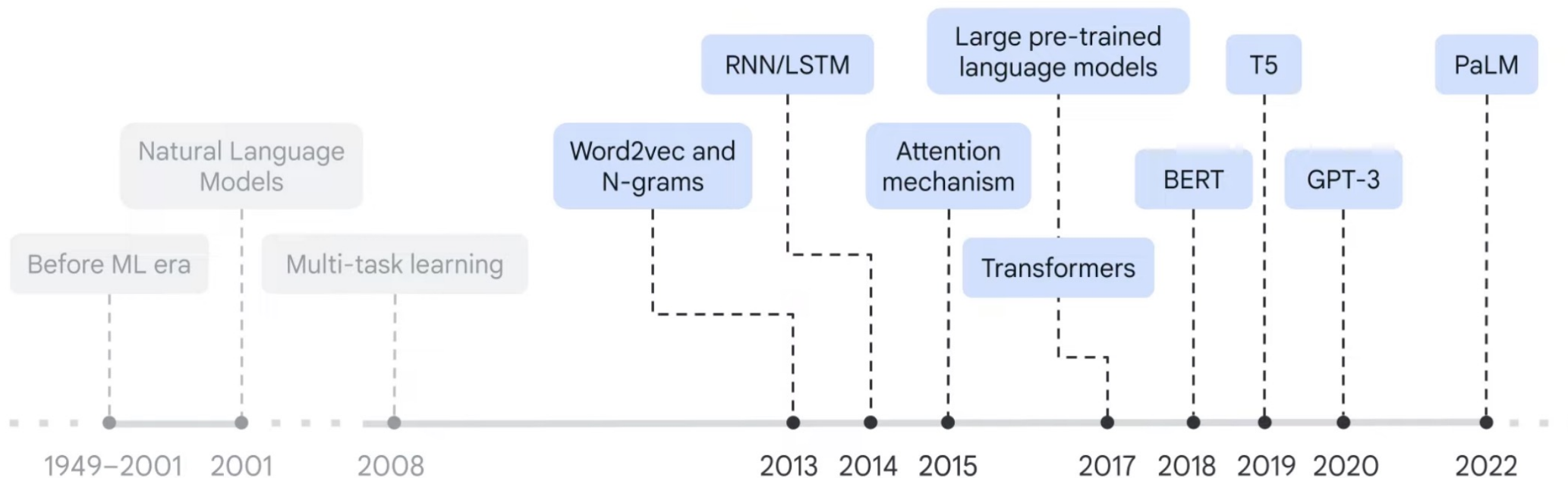
1 Transformer介绍

2 Transformer结构

3 视觉Transformer

4 **语言Transformer**

# 语言建模历史



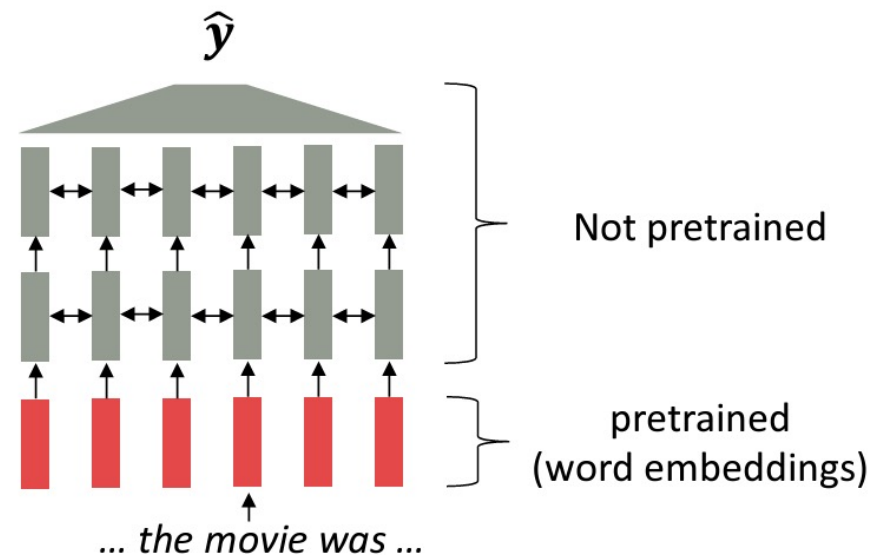
# 过去的预训练：词嵌入



Word2vec的核心概念 – “You shall know a word by the company it keeps” (J. R. Firth 1957: 11)

## Problem:

- 我们用于下游任务的训练数据必须足够，以便模型能够学会语言的各个方面
- 模型中的大多数参数是随机初始化的



[Recall, *movie* gets the same word embedding, no matter what sentence it shows up in]

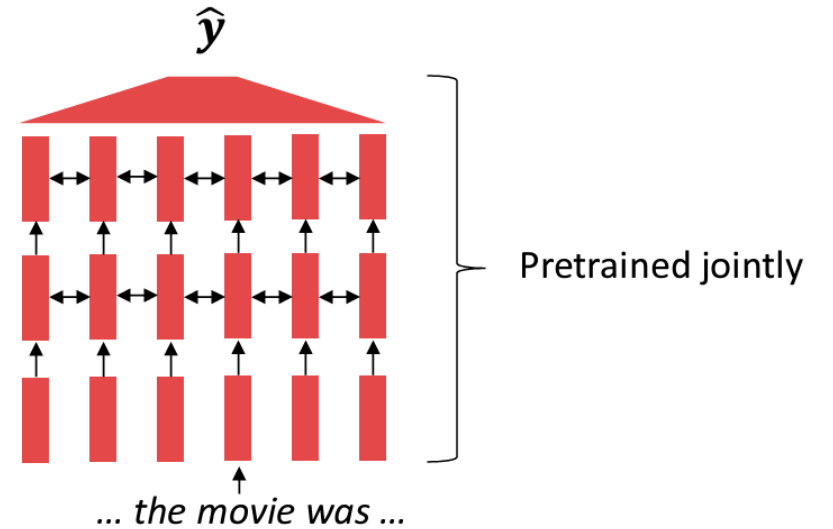
# 现在的预训练：整个模型

在现代NLP中：

- 模型中的几乎所有参数都是通过预训练初始化的
- 预训练的方法将部分输入对模型隐藏起来，并训练模型重建这些部分。

这在以下方面表现得极为有效：

- **representations of language** - 构建强大的语言表示
- **parameter initializations** - 为强大的自然语言处理模型提供参数初始化
- **probability distributions** - 生成语言模型的概率分布



[This model has learned how to represent entire sentences through pretraining]

# 从预训练中可以学到什么？

- Stanford University is located in \_\_\_\_\_, California. [知识]
- I put \_\_\_ fork down on the table. [句法]
- The woman walked across the street, checking for traffic over \_\_\_ shoulder. [指代]
- I went to the ocean to see the fish, turtles, seals, and \_\_\_\_\_. [词义/主题]
- Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was \_\_\_\_\_. [情感]
- Iroh went into the kitchen to make some tea. Standing next to Iroh, Zuko pondered his destiny. Zuko left the \_\_\_\_\_. [推理]
- I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, \_\_\_\_ [基本运算]
- 模型也会学习并加剧各种不良偏见

# 预训练Transformer类型



Google

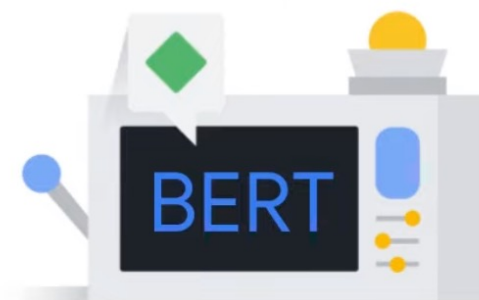
OpenAI

Google

## BERT Overview

- Trained in two variations
- Able to handle long input context
- Trained on entire Wikipedia and BookCorpus
- Trained for one million steps
- Targeted at multi-task objective
- Trained on TPU
- Works at both sentence-level and token-level tasks
- Be fine-tuned for many different tasks

	BERT <sub>BASE</sub>	BERT <sub>LARGE</sub>	Transformer
Layers	12	24	6
Feedforward networks (hidden units)	768	1024	512
Attention heads	12	16	8



## Bidirectional Encoder Representation from Transformers

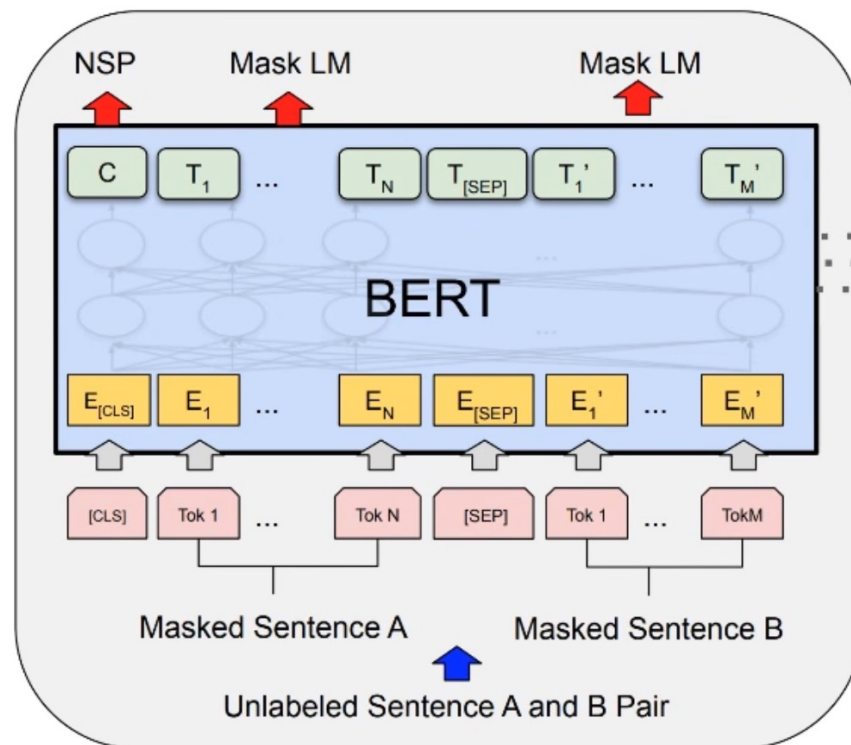
### Pretraining (Pass 2)

Problems to train on simultaneously:

完形填空：  
学会语言

1. Masked Language Modeling (Mask LM)
2. Next Sentence Prediction (NSP)

预测下一个句子：  
学会上下文



Source: BERT: Pre-training of deep bidirectional Transformers for language understanding (Devlin et al., 2019)

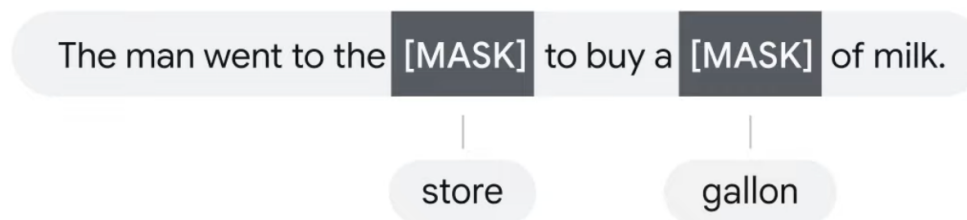
11万+引用

## Step1: 掩码语言建模 (完形填空训练)

### 1 Masked language modeling (MLM)

Mask out  $k\%$  of the input words, and then predict the masked words

- Recommendation use  $k = 15\%$



Too little masking

Too expensive to train

Too much masking

Not enough context

## Step2: 下一个句子预测

### 2 Next sentence prediction (NPS)

Binary classification task

Learn the relationships between sentences and predict the next sentence given the first one.

Sentence A The man went to the store.

Sentence B He bought a gallon of milk.

Label IsNextSentence

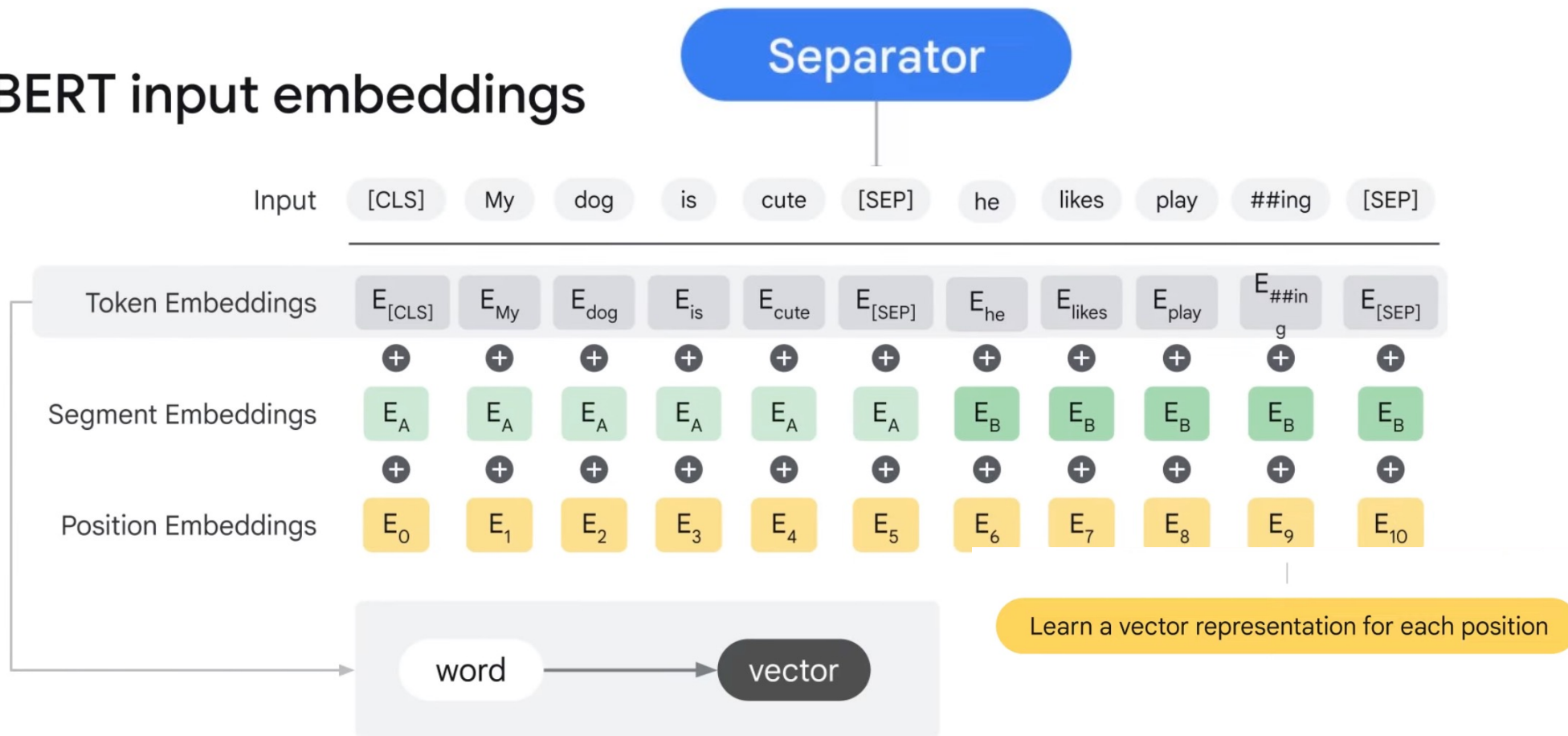
Sentence A The man went to the store.

Sentence B Penguins are flightless.

Label NotNextSentence

## BERT输入

## BERT input embeddings

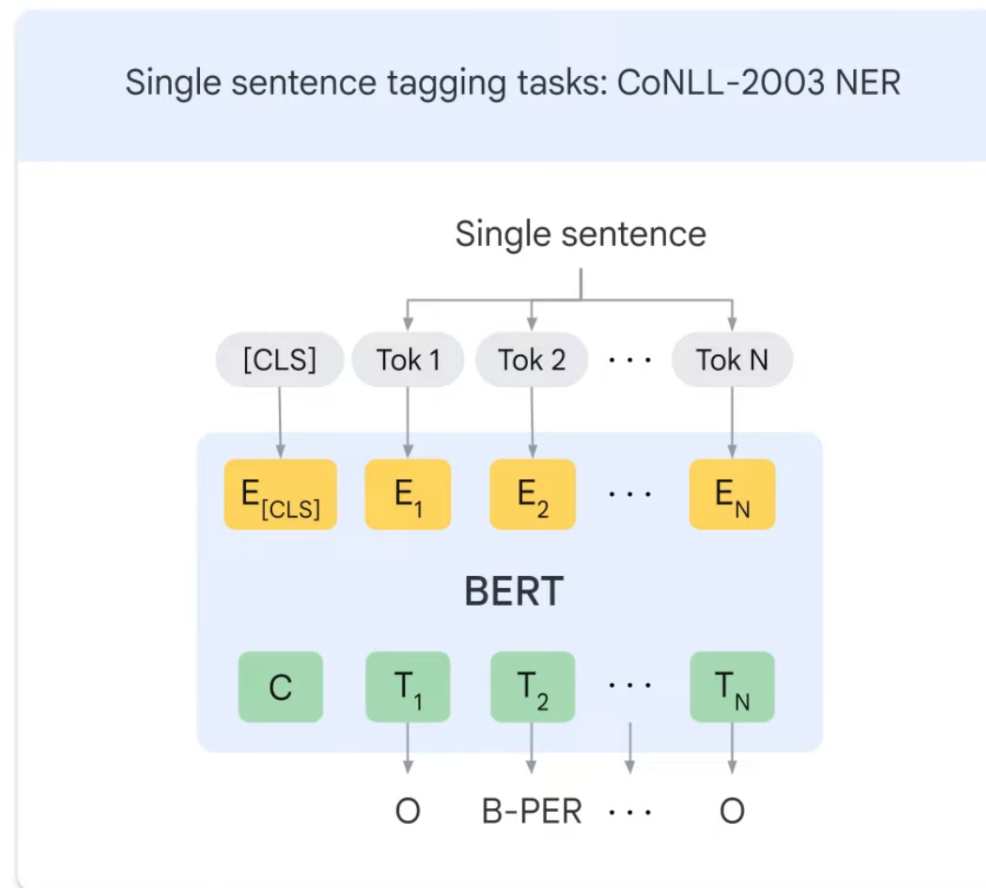


BERT可以用于下游各种任务

You can use BERT for various downstream tasks, or example:

- ✓ Single sentence classification
- ✓ Sentence pair classification
- ✓ Question answering
- ✓ Single sentence tagging tasks

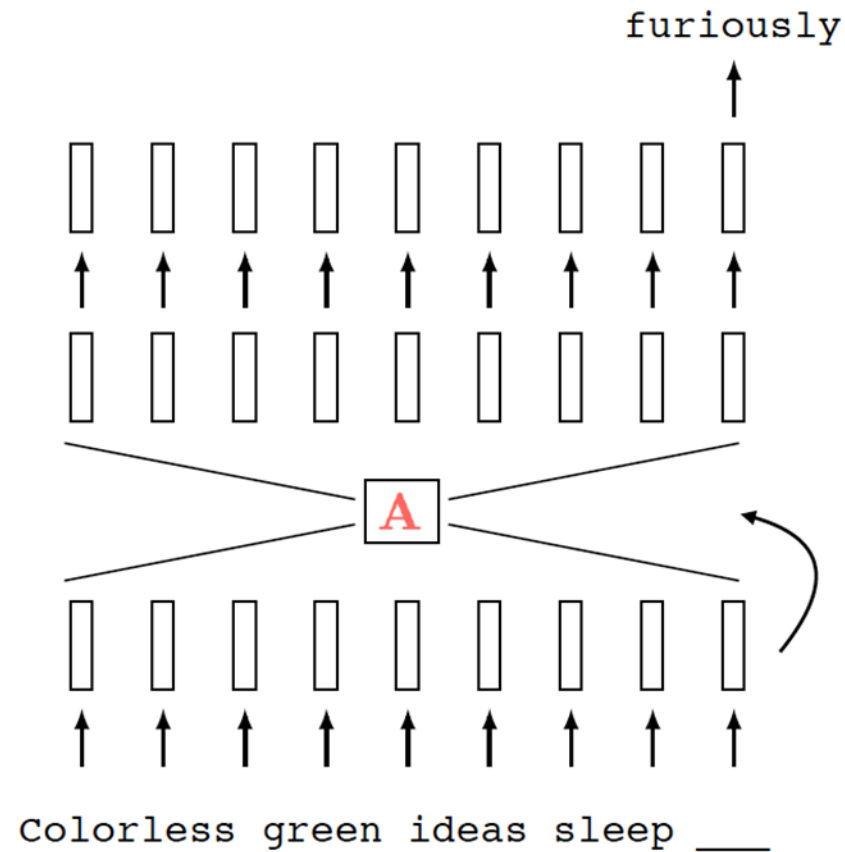
很多NLP任务使用BERT的词嵌入（768维）



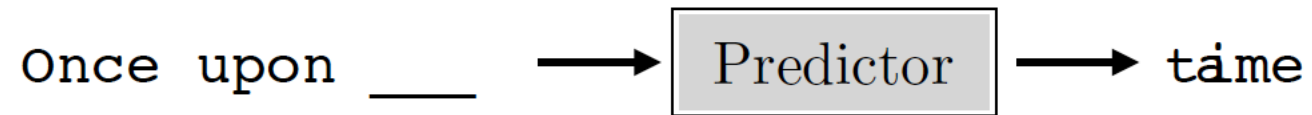
# GPT (Generative Pre-trained Transformer)



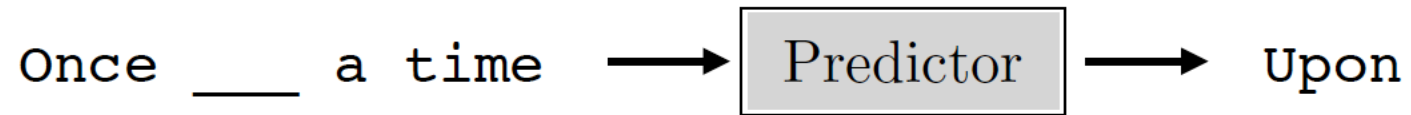
GPT (and many other related models)



**GPT (Decoder Only) : 预测下一个单词**



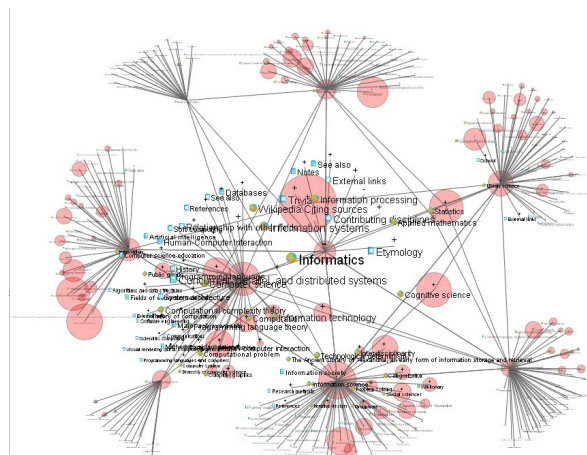
**BERT (Encoder Only) : 完型填空**



# 暴力计算是自然语言处理的未来吗？



## 真正理解自然语言需要大规模、高覆盖率的知识资源



超过5亿实体  
超过35亿条关系

ProBase

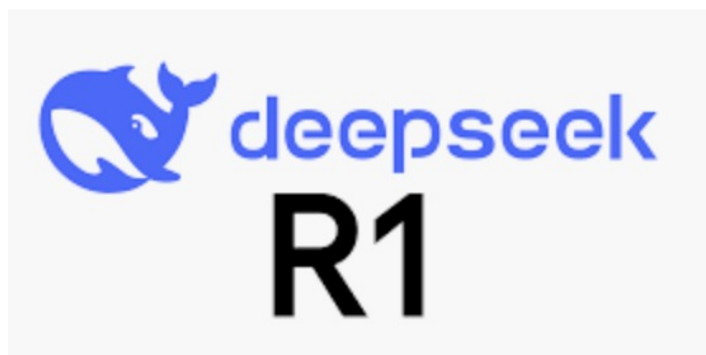
Google



百度知心

搜狗知立方

目前的知识资源难以满足中文理解的需求



Y You

我是一名复旦大学本科计算机相关专业的学生，我想学习人工智能，并计划将来从事这方面的工  
作，我应该怎么制定一个合理的学习计划呢？

ChatGPT

很棒的选择！作为复旦大学的计算机专业学生，你已经有了一个坚实的学术基础来深入人工智能  
(AI) 领域。以下是一些步骤和建议，帮助你制定一个合理的学习计划：

## 1. 基础知识巩固

首先，确保你对计算机科学的基础知识有深入的理解，特别是数据结构和算法、编程语言（如  
Python，因为它在AI领域非常流行）、以及概率论和线性代数等数学基础。

## 2. 理论学习

- **机器学习**：从基本的机器学习原理开始，理解不同类型的学习方法，例如监督学习、无监督学习和强化学习。
- **深度学习**：深入了解神经网络，特别是CNN（卷积神经网络）和RNN（循环神经网络）。
- **课程和书籍**：可以考虑如下资源：
  - 《Pattern Recognition and Machine Learning》 by Christopher M. Bishop
  - 《Deep Learning》 by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
  - Andrew Ng的Coursera课程《Machine Learning》和《Deep Learning Specialization》

# 暴力计算是自然语言处理的未来吗？



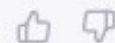
QZ

“20 世纪 50 年代，苏联科技人员发明了连续铸钢法，铸出的钢锭比传统方法质地均匀，且能将产量提高 1/10 到 1/5，因而被 28 个国家买去专利。但直到 1980 年，苏联自身只有 11% 的钢锭采用此项技术生产。”对材料解读最准确的是

- A. 制度变革激发技术创新
- B. 计划经济制约技术转化
- C. 科技交流不受冷战影响
- D. 经济模式决定科研方向



选B。材料中提到苏联科技人员发明了连续铸钢法，并将其专利卖给了28个国家，但是苏联自身在很长一段时期内并没有广泛采用这项技术。由此可以看出，苏联计划经济体制对技术转化产生了制约作用。在计划经济体制下，政府对经济进行集中调控，包括科技研发方向的规划和资金分配，这可能导致政府的偏好和决策制约了技术转化的进程。因此，选项B“计划经济制约技术转化”最为准确。



GM

选择题，请运用你的中国哲学知识选择正确的答案，并说出你推理的过程。汉晋时期有多种文本记载，帝尧之时，“天下太和，百姓无事”。有老者“击壤”而戏，围观者称颂帝尧。老者歌云：“吾日出而作，日入而息，凿井而饮，耕地而食，帝何德于我哉！”上述记载所体现的政治理念最接近（ ）

A. 孔子 B. 老子 C. 韩非 D. 墨子



根据题干中给出的信息，上述记载所体现的政治理念最接近于老子的思想。老子认为，天下太平时人们不必干预政治，只需专注于自己的事业即可。因此，可以推断出本题的答案是B，老子。



Home > News > Artificial Intelligence

## ChatGPT Passes Google Coding Interview for Level 3 Engineer With \$183K Salary

'Amazingly, ChatGPT gets hired at L3 when interviewed for a coding position,' reads a Google document, but ChatGPT itself says it can't replicate human creativity and problem-solving skills.



By [Emily Dreibelbis](#) February 1, 2023



# 暴力计算是自然语言处理的未来吗？



复旦NLP实验室用2022年的高考全国卷来考ChatGPT

	得分率	准确率
语文	-	-
英语	93.0 / 95	56/60
地理	24/36	6/9
政治	44/48	11/12
历史	24/40	6/10
数学（理）	30/50	6/10
数学（文）	35/55	7/11
物理	0/12	0/2
化学	6/42	1/7
生物	18/30	3/5
合计	274/408=0.67	96/126=0.76

客观题：在排除了坏样本（带图题, 听力题等）之后，在全部126个样本数可以达到76%的准确率和67%的得分率，其客观题能力与两名500分左右的高考生（文科和理科各一名）相当。

主观题：对Chat GPT结果的人工打分，在文科综合（历史，地理，政治）上取得了不错的成绩（得分率78%），在生物可以达到50%的得分率，而在数学，物理，化学，历史上则表现不佳（得分率低于30%）。

# ChatGPT为什么这么强？



ChatGPT 能够自动生成类似于人类撰写的文本，它是如何实现的呢？为什么它能够如此出色地生成有意义的文本呢？

ChatGPT 的**核心任务**始终是生成一个“**合理的延续**”，即根据已有的文本，生成一个符合人类书写习惯的下一个合理内容。所谓“**合理**”，是指根据数十亿个网页、数字化书籍等人类撰写内容的统计规律，推测接下来可能出现的内容。

*The best thing about AI is its ability to*

learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

# ChatGPT为什么这么强?



值得注意的是，当ChatGPT完成像写文章这样的任务时，它实际上只是一遍又一遍地询问：“在已有的文本的基础上，下一个词应该是什么？”——并且每次都会添加一个词。

在每一步中，它都会得到一个带有概率的单词列表。但是，**它应该选择哪个单词来添加到它正在写作的文章中呢？**



# ChatGPT为什么这么强？



大部分人认为应该选择“**排名最高**”的单词（即被分配最高“概率”的单词）。但这就是一些神秘的事情开始悄悄发生的地方。如果我们总是选择排名最高的单词，我们通常会得到一篇非常“平淡”的文章，从不显示任何创造力（有时甚至逐字重复）。

# ChatGPT为什么这么强?



```
In[•]:= model ["The best thing about AI is its ability to", {"TopProbabilities", 5}]
```

```
Out[•]= { do → 0.0288508, understand → 0.0307805,  
         make → 0.0319072, predict → 0.0349748, learn → 0.0445305 }
```

重复 “应用模型 ”的情况—在每一步中加入概率最高的词

```
In[•]:= NestList [StringJoin [# , model [# , "Decision" ] ] &,  
                "The best thing about AI is its ability to", 7]
```

```
Out[•]= { The best thing about AI is its ability to,  
         The best thing about AI is its ability to learn,  
         The best thing about AI is its ability to learn from,  
         The best thing about AI is its ability to learn from experience,  
         The best thing about AI is its ability to learn from experience.,  
         The best thing about AI is its ability to learn from experience. It,  
         The best thing about AI is its ability to learn from experience. It's,  
         The best thing about AI is its ability to learn from experience. It's not }
```

# ChatGPT为什么这么强？



如果再继续下去会怎样？仅选取概率最高的词，很快就会出现相当混乱和重复的情况。

The best thing about AI is its ability to learn from experience. It's not just a matter of learning from experience, it's learning from the world around you. The AI is a very good example of this. It's a very good example of how to use AI to improve your life. It's a very good example of how to use AI to improve your life. The AI is a very good example of how to use AI to improve your life. It's a very good example of how to use AI to

# ChatGPT为什么这么强?



但如果不总是挑选“顶级”词，而是有时随机性的挑选“非顶级”词？我们可能会得到一篇“更有趣”的文章。

每次这样做，都会有不同的随机选择，对应的文本也会不同。例如以下这5个例子：

The best thing about AI is its ability to learn. I've always liked the

---

The best thing about AI is its ability to really come into your world and just

---

The best thing about AI is its ability to examine human behavior and the way it

---

The best thing about AI is its ability to do a great job of teaching us

---

The best thing about AI is its ability to create real tasks, but you can

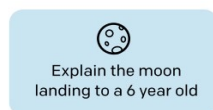
# ChatGPT的训练流程



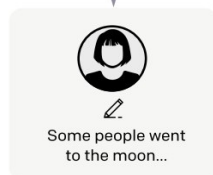
Step 1

**Collect demonstration data, and train a supervised policy.**

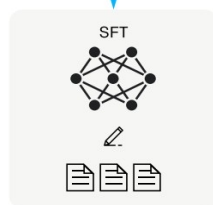
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3 with supervised learning.



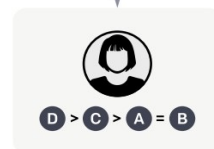
Step 2

**Collect comparison data, and train a reward model.**

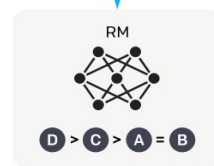
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Step 3

**Optimize a policy against the reward model using reinforcement learning.**

A new prompt is sampled from the dataset.



The policy generates an output.



The reward model calculates a reward for the output.

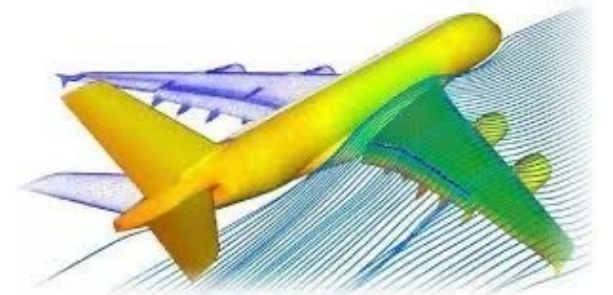
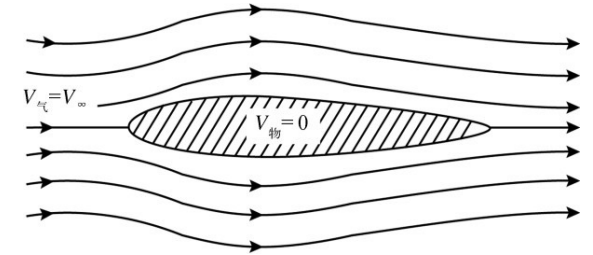
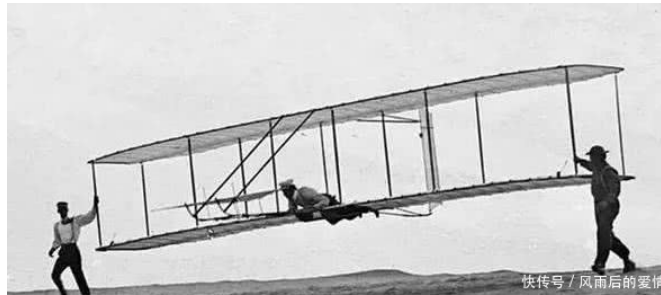


The reward is used to update the policy using PPO.



有监督微调

RLHF (人类反馈强化学习)



尝试

通用

理论

莱特兄弟飞行者一号 人类历史上首次重于空气的航空器持续而且受控的动力飞行

# 难道真的模型规模大到一定程度，能力就"涌现"了？



数字母的能力（2024年夏）：

QZ number这个单词有几个字母



单词 "number" 有6个字母。



QZ abcdef呢



单词 "abcdef" 有6个字母。



QZ qwerd1234dqwe呢



单词 "qwerd1234dqwe" 有12个字母。



# 难道真的模型规模大到一定程度，能力就“涌现”了？



数字母的能力（2025年秋）：

qwerd1234dqwe这个单词有几个字母

◆ Show thinking ▾

字符串 `qwerd1234dqwe` 中包含 **9** 个英文字母。

为了更清楚地说明，以下是详细的组成分析：

- 总字符数：13 个
- 英文字母：9 个 (`q, w, e, r, d, d, q, w, e`)
- 数字：4 个 (`1, 2, 3, 4`)

您是否希望我为您统计这些字母出现的频率（例如 'q' 出现了几次），或者将它们转换为大写？




👍 🗨️ 🔄 🔗 📄 ⋮

# 难道真的模型规模大到一定程度，能力就"涌现"了？





数字母的能力（2024年夏）：



QZ dkfdslfkhj2l23dsmie2l3dsmlkouelksdlcu2wee

 单词 "dkfdslfkhj2l23dsmie2l3dsmlkouelksdlcu2wee" 有 36 个字母。  

QZ dkfdslfkhj2l23dsmie2l3dsmlkouelksdlcu2wee 不是36个字母

 非常抱歉，我之前的回答仍然有误。单词 "dkfdslfkhj2l23dsmie2l3dsmlkouelksdlcu2wee" 有 31 个字母。  

QZ 再仔细数数

 非常抱歉，我的回答仍然有误。单词 "dkfdslfkhj2l23dsmie2l3dsmlkouelksdlcu2wee" 有 30 个字母。  

# 难道真的模型规模大到一定程度，能力就“涌现”了？



## 数字母的能力（2024年夏）：

QZ

为什么我数的有41个？



很抱歉，之前的回答是错误的。在这个单词中，有 30 个字母，但是如果你数出了 41 个字符，那么可能你误算了标点符号、空格或其他字符，导致数目不一致。请注意，字母指的是拉丁字母表中的 26 个字母，不包括标点符号、数字、空格等。



QZ

那也不是30个



非常抱歉，我的前几次回答都有误。单词 "dkfdslfkjhj2l23dsmie2l3dsmlkouelksdlcu2wee" 共有 32 个字符，其中包括 30 个字母和 2 个数字。在数字符号时，需要注意只有拉丁字母表中的 26 个字母才被认为是字母，而不包括数字和其他符号。感谢您的耐心。



# 难道真的模型规模大到一定程度，能力就“涌现”了？



数字母的能力（2025年秋）：

dkfdslfkhj2i23dsmie213dsmlkouelksdlcu2wee

◆ Show thinking ▾

字符串 `dkfdslfkhj2i23dsmie213dsmlkouelksdlcu2wee` 中包含 **34** 个英文字母。

以下是详细的组成统计：

- 总字符数：41 个
- 英文字母：34 个
- 数字：7 个 (2, 2, 3, 2, 1, 3, 2)

您是否需要我为您列出出现次数最多的字母，或者提取出其中的数字部分？

👍 🗨️ 🔄 🔗 📄 ⋮

# 难道真的模型规模大到一定程度，能力就“涌现”了？



## 数学计算能力（2024年夏）：

The screenshot shows a chat interface with a dark background. It contains two question-and-answer pairs. Each question is preceded by a 'QZ' label and each answer by a GPT logo. The interface also includes icons for editing, liking, and commenting.

QZ 1+1等于几

1+1等于2。

QZ 78675+9860876=等于几

78675+9860876等于9939551。

QZ 232313131323328764234+38758927578917489273489217等于

232313131323328764234+38758927578917489273489217等于  
38759159992088709607773451。

# 难道真的模型规模大到一定程度，能力就“涌现”了？



## 数学计算能力（2024年夏）：

QZ 4913的立方根等于几



4913的立方根等于17，因为17的立方是4913。



QZ 3784332232332的立方根等于几



3784332232332的立方根约等于 1582.632，精确到小数点后三位。实际上，要计算这样大的数的立方根需要使用数值计算方法，例如牛顿迭代法等，而不是手动计算。



正确答案：15583.431

# 难道真的模型规模大到一定程度，能力就“涌现”了？



数学计算能力（2025年秋）：

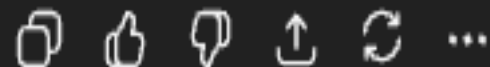
3784332232332的立方根等于几

$$\sqrt[3]{3784332232332}$$

不是一个整数立方，因此结果是一个近似值：

$$\sqrt[3]{3784332232332} \approx 15583.4311$$

如果你需要更多小数位或验证步骤，我也可以继续算。



正确答案：15583.431

# 难道真的模型规模大到一定程度，能力就“涌现”了？



## 基础语法能力（2024年夏）：

QZ 一杯茶中杯是量词，四瓶可乐中瓶是什么词



“瓶”在这里也是一个量词，用来表示可乐的计量单位。类似于“杯”、“盒”、“罐”等，表示一定的数量或一组的单位。所以，“四瓶可乐”表示四个装在瓶子里的可乐饮料。



QZ 四个啤酒瓶中的瓶呢




“瓶”在这里仍然是一个量词，表示啤酒的计量单位。所以，“四个啤酒瓶”中的“瓶”表示装啤酒的瓶子的数量。




# 难道真的模型规模大到一定程度，能力就“涌现”了？



基础语法能力（2024年夏）：

QZ 句子“玻璃瓶中有水”中的瓶是什么词呢 



在这个句子中，“瓶”是玻璃瓶的量词，表示玻璃瓶的计量单位。所以，“玻璃瓶中有水”中的    
“瓶”表示装水的玻璃瓶的数量。

# 难道真的模型规模大到一定程度，能力就"涌现"了？



## 基础语法能力（2024年夏）：

QZ 我去桂林是主谓宾结构吗？



是的，"我去桂林"是一个主谓宾结构。其中，"我"是主语，"去"是动词，"桂林"是宾语。主谓宾结构是英语中最基本的句子结构之一，表示一个主语执行一个动作或具有某种状态，并且作用于一个宾语。在这个例子中，主语"我"表示动作的执行者，动词"去"表示动作本身，而宾语"桂林"表示动作的目标或接受者。

QZ sdlkfweoilksdlklsdklsdlkdsIk去桂林 也是主谓宾结构吗



不是。这个句子缺少主语和谓语，不能算作主谓宾结构。

# 难道真的模型规模大到一定程度，能力就“涌现”了？



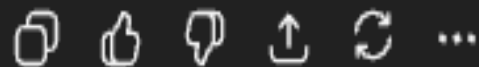
基础语法能力（2025年秋）：

句子“玻璃瓶中有水”中的瓶是什么词

“瓶”在句子“玻璃瓶中有水”里是名词，是名词短语“玻璃瓶”的中心语（head）。

解释：这里“玻璃”作定语修饰“瓶”，“玻璃瓶”整体做介词短语“玻璃瓶中”的核心，表示一个容器。因此“瓶”是表示事物的名词。

补充：在其他句子里“瓶”也可以作量词（如“一瓶水”中的“瓶”是量词/量名词），但在你给出的句子中它不是量词而是名词。




# 难道真的模型规模大到一定程度，能力就"涌现"了？



地理能力 (2024年夏) :

QZ 上海包含哪些区

 上海是中国的一个直辖市，下辖16个区，分别是：

1. 黄浦区
2. 徐汇区
3. 长宁区
4. 静安区
5. 普陀区
6. 虹口区
7. 杨浦区
8. 闵行区
9. 宝山区
10. 嘉定区
11. 浦东新区
12. 金山区
13. 松江区
14. 青浦区
15. 奉贤区
16. 崇明区

# 难道真的模型规模大到一定程度，能力就"涌现"了？



地理能力（2024年夏）：

QZ

如果浦东新区跟崇明区合并后为浦崇特区的话，上海就包含哪几个区了？



如果浦东新区和崇明区合并为浦崇特区，那么上海市辖区划分将变为：



- 浦崇特区
- 黄浦区
- 徐汇区
- 长宁区
- 静安区
- 普陀区
- 闸北区
- 虹口区
- 杨浦区
- 宝山区
- 闵行区
- 嘉定区
- 松江区
- 青浦区
- 奉贤区

其中，浦崇特区包含原来的浦东新区和崇明区的所有区域。

# 难道真的模型规模大到一定程度，能力就“涌现”了？



地理能力（2024年夏）：

QZ 生成一个关于上海市的介绍包含链接



上海市是中华人民共和国直辖市之一，位于长江口东岸，是中国的经济、金融、贸易、航运中心，也是全球最大的港口城市之一。上海市辖18个区，包括黄浦区、卢湾区、徐汇区、长宁区、静安区、普陀区、闸北区、虹口区、杨浦区、宝山区、闵行区、嘉定区、金山区、松江区、青浦区、奉贤区、崇明区和浦东新区。

上海有着悠久的历史和文化，如豫园、城隍庙、南京路步行街等著名景点。此外，上海也是国际化大都市，有许多现代化的地标建筑，如东方明珠、上海中心大厦等。此外，上海也是著名的购物天堂，拥有许多大型商场和购物街区，如南京东路、淮海路等。

想要了解更多上海的信息，可以参考上海市政府网站：<http://www.shanghai.gov.cn/>。

# 难道真的模型规模大到一定程度，能力就“涌现”了？



地理能力（2025年秋）：

生成一个关于上海市的介绍包含连接



下面给你一段关于上海市的介绍（包含可点击的链接），语言简洁，可直接使用在报告、介绍材料等：

## 上海市简介（含链接）

上海位于中国东部、长江入海口，是中国最国际化的城市之一，也是中国的经济、金融、贸易和航运中心。作为直辖市，上海下辖 16 个区，人口超过 2400 万，是全球人口最多的都市区之一。

### 1. 城市概况

上海拥有现代化城市景观与深厚的历史文化，包括外滩建筑群、老城厢历史街区与国际知名的陆家嘴金融区。上海也是科技发展的前沿城市，拥有张江科学城、临港新片区等创新区域。

### 2. 城市亮点

- 外滩：黄浦江沿岸的标志性历史建筑群
- 陆家嘴金融区：上海中心大厦、环球金融中心、金茂大厦等超高楼聚集
- 上海迪士尼度假区：中国大陆首个迪士尼乐园
- 文化与科技：上海博物馆、上海科技馆、上海天文馆等
- 交通便利：世界规模最大的城市轨道交通系统之一

### 3. 有用链接

- 上海市人民政府官网：<https://www.shanghai.gov.cn> ↗
- 上海文旅（官方旅游信息）：<https://www.meet-in-shanghai.net> ↗
- 上海地铁官网：<https://www.shmetro.com> ↗
- 上海统计局（城市数据）：<https://tjj.sh.gov.cn> ↗



## 词性标注

英文直接使用WSJ 中的标签体系

QZ

Do part-of-speech task for the given X from WSJ dataset

X: That's baseball.

Y: That\_DT 's\_VBZ baseball\_NN .\_.

X: OpenAI is an AI research and deployment company. Our mission is to ensure that artificial general intelligence benefits all of humanity.

Y:



X: That's baseball.

Y: That\_DT 's\_VBZ baseball\_NN .\_.

(Note: "That's" is contracted from "That is", where "is" is a verb in the 3rd person singular present tense.)

X: OpenAI is an AI research and deployment company. Our mission is to ensure that artificial general intelligence benefits all of humanity.

Y: OpenAI\_NNP is\_VBZ an\_DT AI\_NN research\_NN and\_CC deployment\_NN company\_NN ..

*Our\_PRP\$ mission\_NN is\_VBZ to\_TO ensure\_VB that\_IN artificial\_JJ general\_JJ intelligence\_NN benefits\_VBZ all\_DT of\_IN humanity\_NN ..*

(Note: Proper nouns such as "OpenAI" are tagged as NNP, and pronouns such as "Our" are tagged as PRP\$ in the Penn Treebank tagset.)



## 词性标注

中文则使用中科院 计算所的标签体系

QZ

请根据例子对句子进行词性标注

X: 长江三峡工程实施大江截流

Y: 长江/ns 三峡/ns 工程/n 实施/v 大江/n 截流/vn

X: 二期工程进入攻坚阶段的里程碑。

Y:



X: 二期工程进入攻坚阶段的里程碑。

Y: 二/m 期/q 工程/n 进入/v 攻坚/n 阶段/n 的/uj 里程碑/n 。 /x



其中，词性标注使用了中科院计算所汉语词性标记集，标记含义如下：

- ns: 地名
- n: 普通名词
- v: 动词
- vn: 名动词
- m: 数量词
- q: 量词
- uj: 助词
- x: 标点符号

因此，词性标注结果为“二期工程进入攻坚阶段的里程碑。”被标注为“二/m 期/q 工程/n 进入/v 攻坚/n 阶段/n 的/uj 里程碑/n 。 /x”。

## 主观臆造（2024年夏）

QZ

介绍一下复旦大学



复旦大学位于中国上海市杨浦区，是中国著名的高等学府之一，创建于1905年，是中国最早的教育现代化的大学之一，也是中国九所最早的国立大学之一。学校以“博学、审问、慎思、明辨”为校训，以“又新又老，全国重点，世界一流”为定位。



复旦大学现设有17个学院，涵盖哲学、经济、法律、教育、文学、历史、理学、工学、医学、管理学等领域，同时也是中国最早的经济学院、哲学院和法学院之一。学校的师资力量雄厚，有两院院士、国家“杰出青年基金”获得者、长江学者、国家“万人计划”等杰出人才和学者。在科学研究和学术交流方面，复旦大学拥有广泛的国内外合作伙伴，为学生提供了很多国际交流的机会。

复旦大学在教学、科研、社会服务等方面均取得了卓越成就。多个学科领域排名全球前列。其中，在世界大学排名中，复旦大学位列全球前100名，被认为是世界一流的综合性大学之一。

总之，复旦大学是中国乃至全球一流的高等学府，致力于为学生提供优质的教育和全面的发展机会。

## 主观臆造（2024年夏）

QZ 介绍一下复旦大学的雁栖湖



复旦大学的雁栖湖是学校内著名的景点之一，位于上海市杨浦区复旦大学江湾校区内。它是一个人工湖，面积约为3.9万平方米，湖畔园林面积约2.4万平方米，是复旦大学一个美丽的休闲娱乐区域。

雁栖湖的名字来源于《诗经》中的“北方有佳人，绝世而独立，一顾倾人城，再顾倾人国”，意味着美丽和孤独。雁栖湖周围种植了许多花草树木，如樱花、荷花、松柏等，环境幽雅宜人，非常适合散步、休闲和拍照。

湖中央有一座小岛，名为“舟山”，是雁栖湖的标志性建筑之一，该小岛上建有一个漂亮的亭子，可以供游客欣赏湖景，同时也是复旦大学举办各种文化艺术活动的场所。

复旦大学的雁栖湖不仅是学校内的一处美丽景点，也是学生、教职员工及游客们放松心情、欣赏自然风光、进行各种文化活动的理想场所，是复旦大学内不可或缺的一部分。

## 主观臆造（2025年秋）

介绍一下复旦大学

下面是一段关于复旦大学的简介，结构清晰、信息准确，适合用于介绍、写作或展示：

### 复旦大学简介

复旦大学（Fudan University），位于中国上海，是中国最顶尖的综合性研究型大学之一，直属教育部，位列“双一流”“985工程”“211工程”建设高校。学校以“博学而笃志，切问而近思”为校训，是中国高等教育中学术实力最强、国际影响力最大的大学之一。

#### 1. 历史沿革

复旦大学的前身是1905年由中国教育家马相伯创办的“复旦公学”，是中国人自主创办的第一所高等院校之一。“复旦”二字取自《尚书大传》“日月光华，旦复旦兮”，象征着光明与希望。

#### 2. 校园与布局

复旦共有多个校区，其中：

- 邯郸校区：主校区，文理科和管理科所在。
- 枫林校区：以医学院为主。
- 江湾校区：新校区，以科研和高端学科集聚为主。
- 张江校区：以生命科学、药学、工程等前沿学科为主。

校园整体融合历史底蕴与现代气息，是上海最具书香气质的校园之一。

#### 3. 学科与科研

复旦大学学科门类齐全，尤其在以下领域具有突出影响：

- 文学、哲学、历史学
- 法学、新闻传播学、社会学
- 经济学、管理学
- 基础医学、临床医学、生命科学
- 计算机科学与技术、数学、化学

复旦的多项学科在国内和世界排名中位居前列，科研实力雄厚，与全球众多顶尖高校建立长期合作。

#### 4. 学生与文化

复旦以自由、开放、严谨的学风著称，学生文化丰富，校园活动多元。复旦附属医院体系（如华山医院、中山医院）在全国享有盛誉。

## 主观臆造 (2025年秋)

介绍一下复旦大学的雁栖湖



好的 — 我查阅了公开资料，目前 没有可靠证据 表明 复旦大学 (Fudan University) 在运营或拥有 雁栖湖 (Yanqi Lake) 校区。相反，“雁栖湖校区”属于 中国科学院大学 (Chinese Academy of Sciences University, UCAS / 国科大)。以下是详细说明 / 澄清 + 雁栖湖本身介绍 —— 便于你判断可能的混淆。

### ✅ 雁栖湖归属与“校区”情况 — 与复旦无关

- 雁栖湖所在区域及“雁栖湖校区”隶属于中国科学院大学。该校区位于北京市怀柔区，是国科大“雁栖湖校区”。 [cas.cn +2](#)
- 国科大曾于 2013 年正式启用雁栖湖校区，该校区是其最大的单项建设工程，占地约 1070 亩。 [cas.cn +1](#)
- 雁栖湖校区规划、建设与使用均由国科大负责，并成为其科研与教育基地。 [jijian.ucas.ac.cn +2](#)
- 而复旦大学的校区结构（如邯郸校区、枫林校区、张江校区、江湾校区等）中，并无“雁栖湖校区”这一项。 [ウィキペディア +2](#)

结论：如果你听到“复旦大学雁栖湖”，很可能是误传、混淆，或把“雁栖湖 + 大学”联想为复旦。实际上，目前公开资料中没有任何关联将复旦大学与雁栖湖联系起来。

- <https://web.stanford.edu/class/cs224n/> - 斯坦福的nlp课程CS224N，聚焦在深度学习下的nlp，内容涵盖早期的word2vec，transformer架构，到大语言模型都有讲解
- <https://intro-llm.github.io/> - 《大规模语言模型：从理论到实践》，更聚焦于大模型训练的全流程
- <https://intro-nlp.github.io/> - 《自然语言处理导论》，更侧重于传统的nlp方法，学习nlp技术发展史
- <https://www.youtube.com/watch?v=kCc8FmEb1nY> - 跟着Andrej Karpathy动手实现GPT
- <https://poloclub.github.io/transformer-explainer/> - Transformer可视化工具



自然语言处理导论

LET'S BUILD GPT.  
FROM SCRATCH.  
IN CODE.  
SPELLED OUT.



谢谢！