Transformer

本节课所使用的课件部分源自台湾大学李宏毅老师的机器学习课件，

所使用图片部分源自下述网址

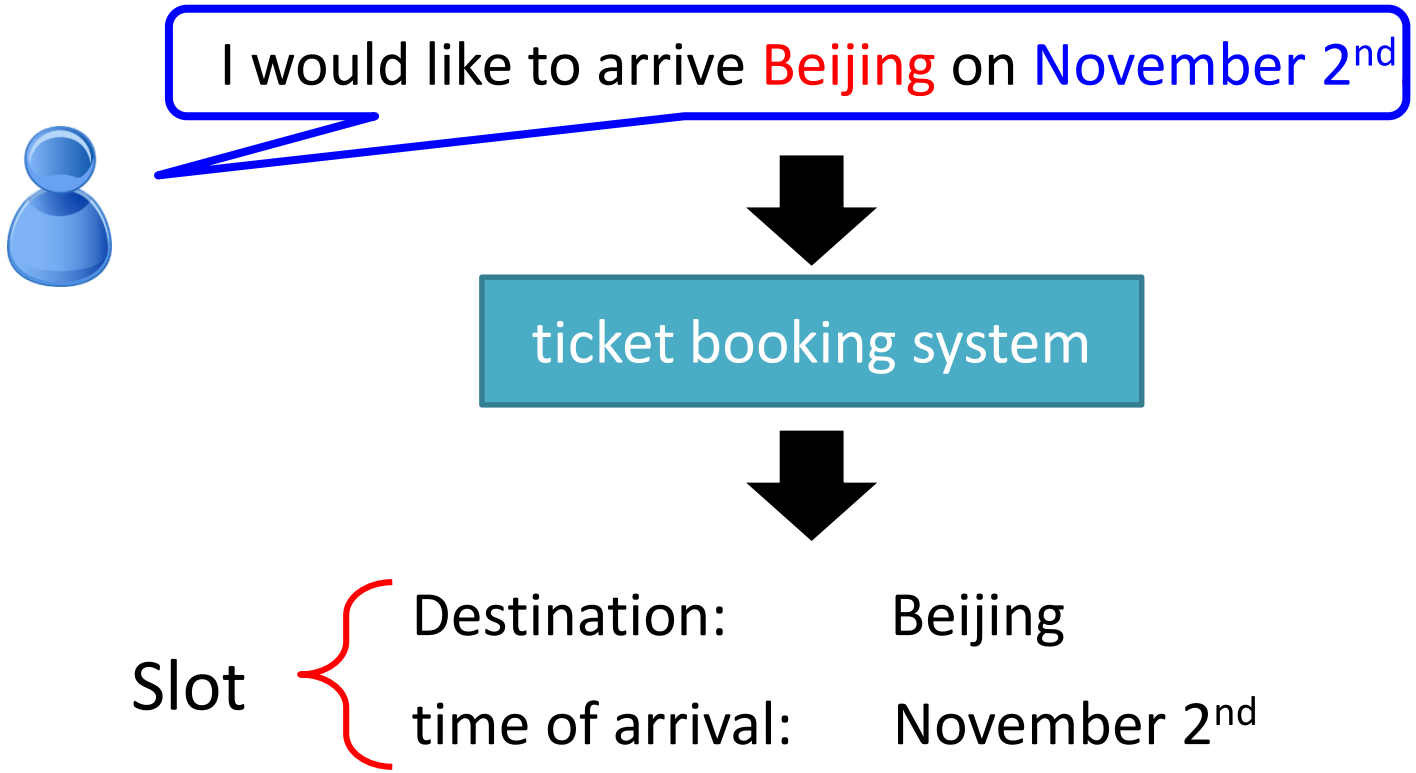
http://jalammar.github.io/illustrated-transformer/

今日主题

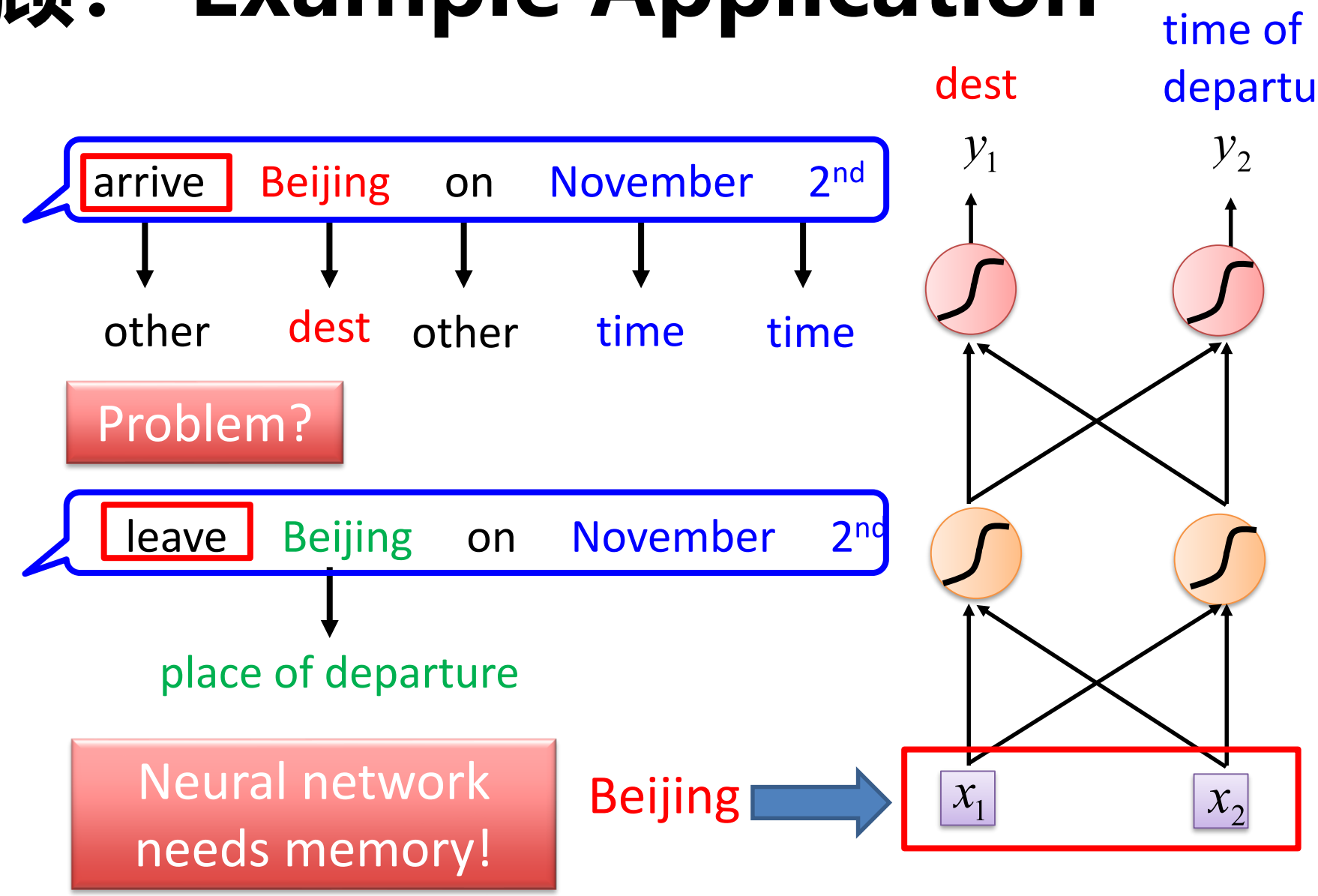
- Transformer
- Non-Local 模块
- ViT
- MAE

回顾：Example Application

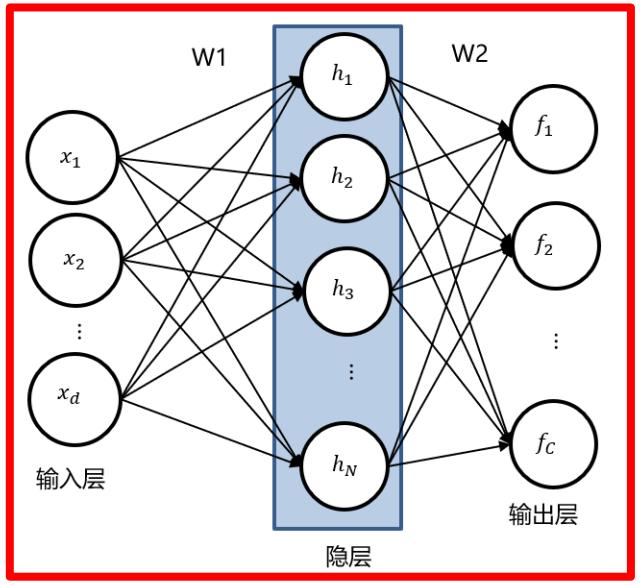
- Slot Filling



回顾： Example Application



Learning Target



other dest other

0 … 1 … 0 | 0 … 1 … 0 | 0 … 1 … 0

y^1 y^2 y^3

a^1 copy a^1 a^2 copy a^2 a^3

W^i

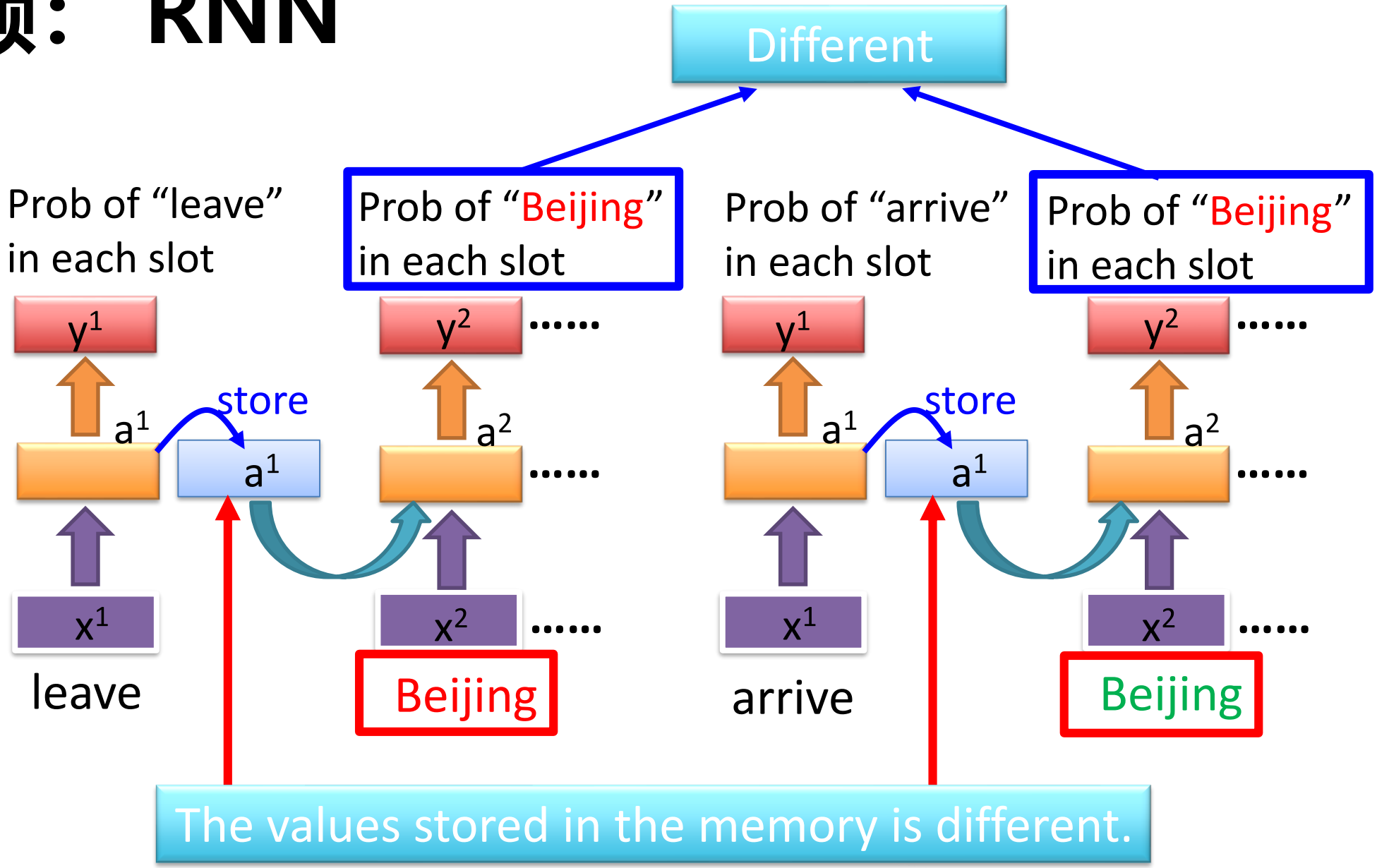
x^1 x^2 x^3

Training Sentences:

arrive Beijing on November 2nd

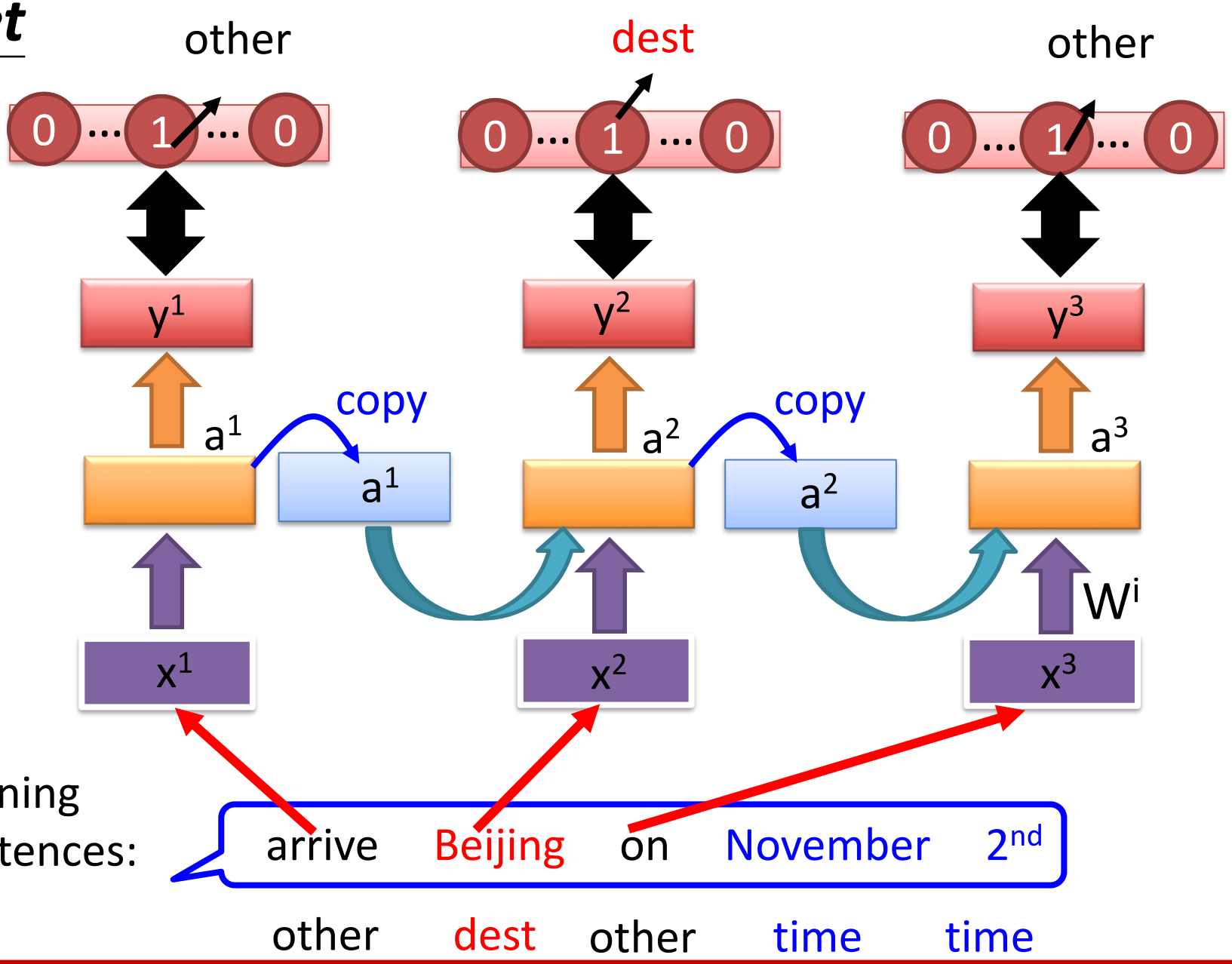
other dest other time time

回顾： RNN

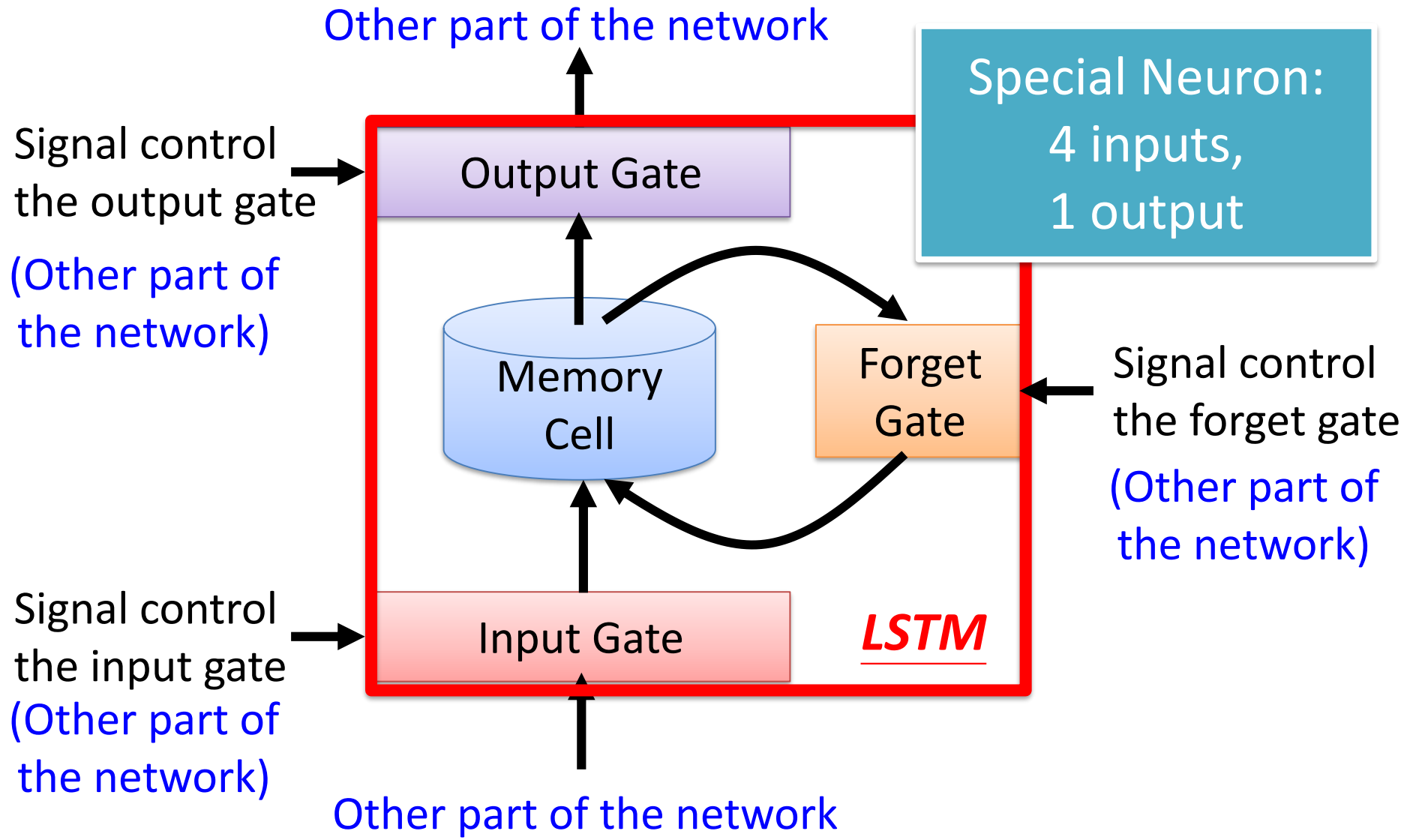

Different
Prob of "leave" in each slot
Prob of "Beijing" in each slot
Prob of "arrive" in each slot
Prob of "Beijing" in each slot
y^1
y^2
.....
a^1
a^2
store
x^1
x^2
leave
Beijing
arrive
Beijing
The values stored in the memory is different.

Learning Target

存在的问题?

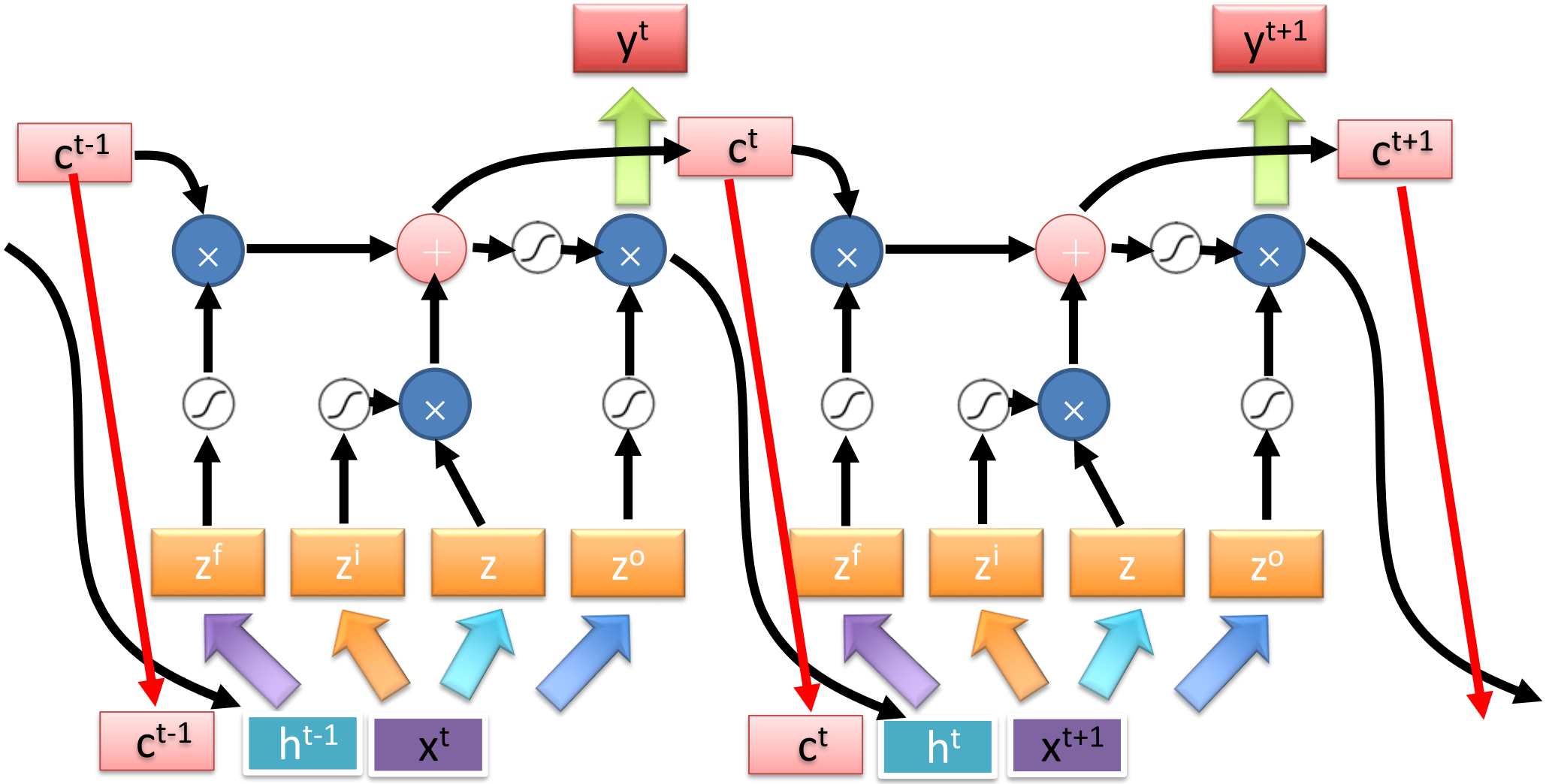


回顾： Long Short-term Memory (LSTM)



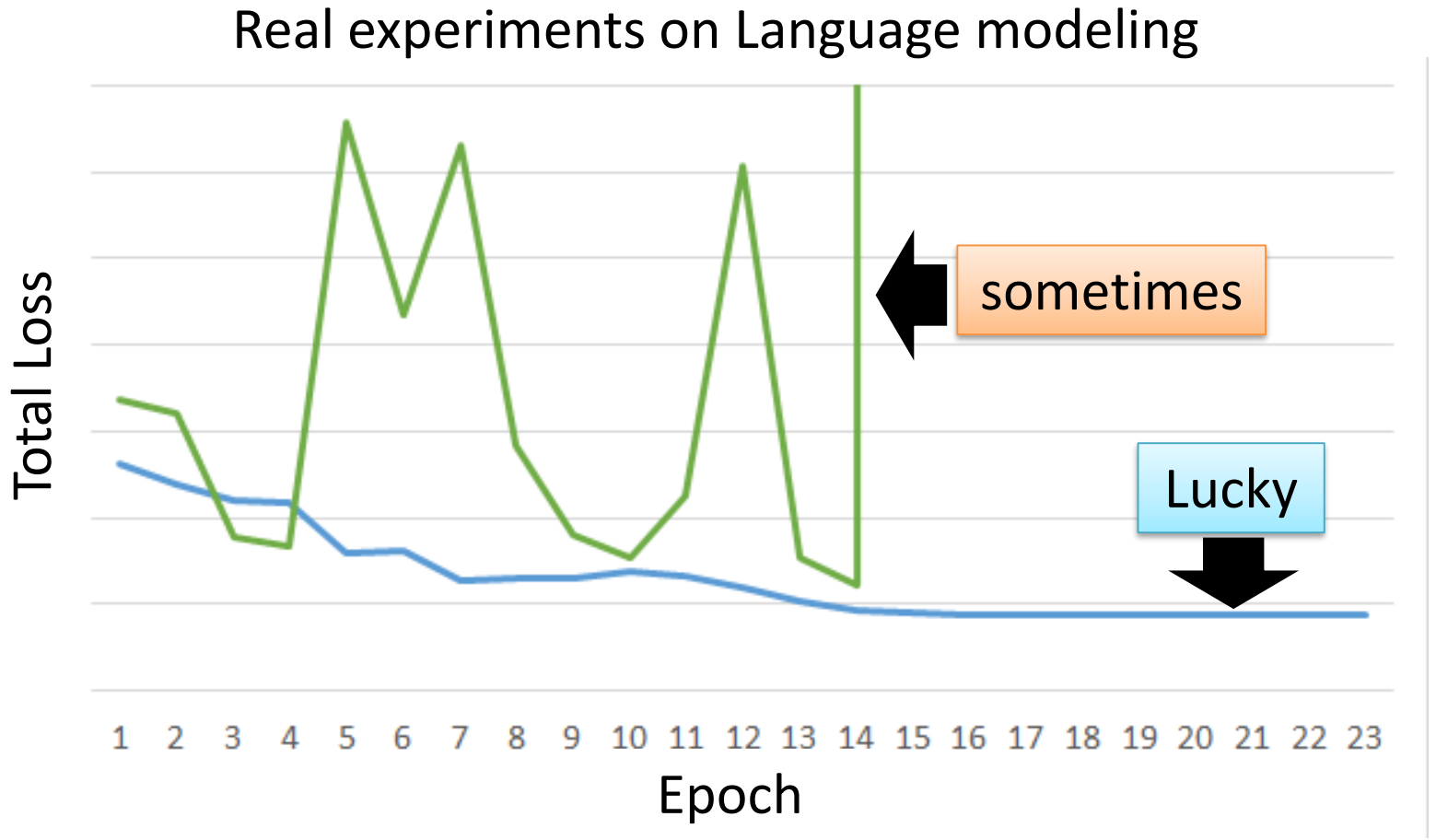
回顾： LSTM

Extension: “peephole”

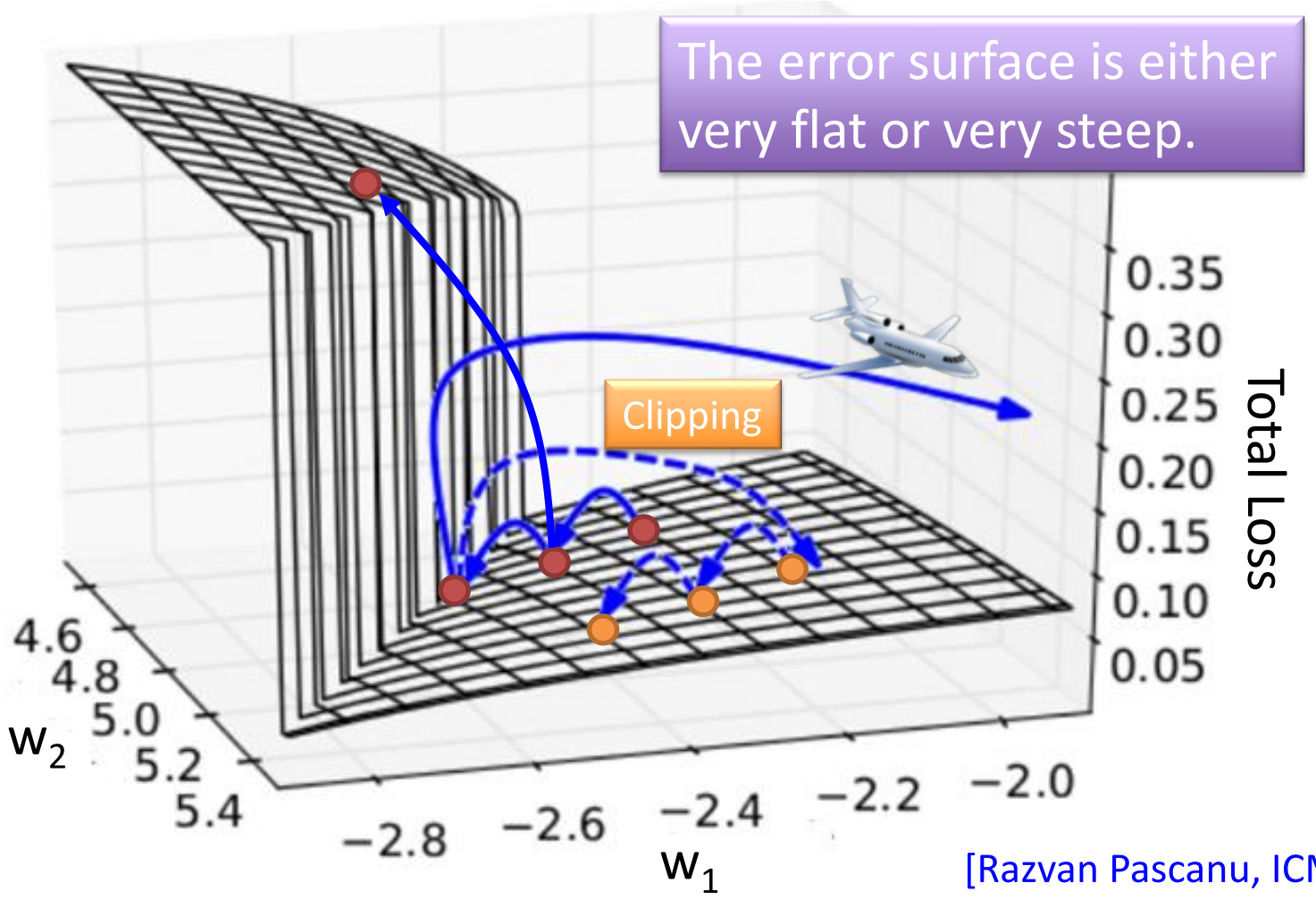


回顾： Unfortunately ……

- RNN-based network is not always easy to learn



回顾： The error surface is rough.



[Razvan Pascanu, ICML'13]

回顾： Why?

$w = 1 \Rightarrow y^{1000} = 1$

$w = 1.01 \Rightarrow y^{1000} \approx 20000$

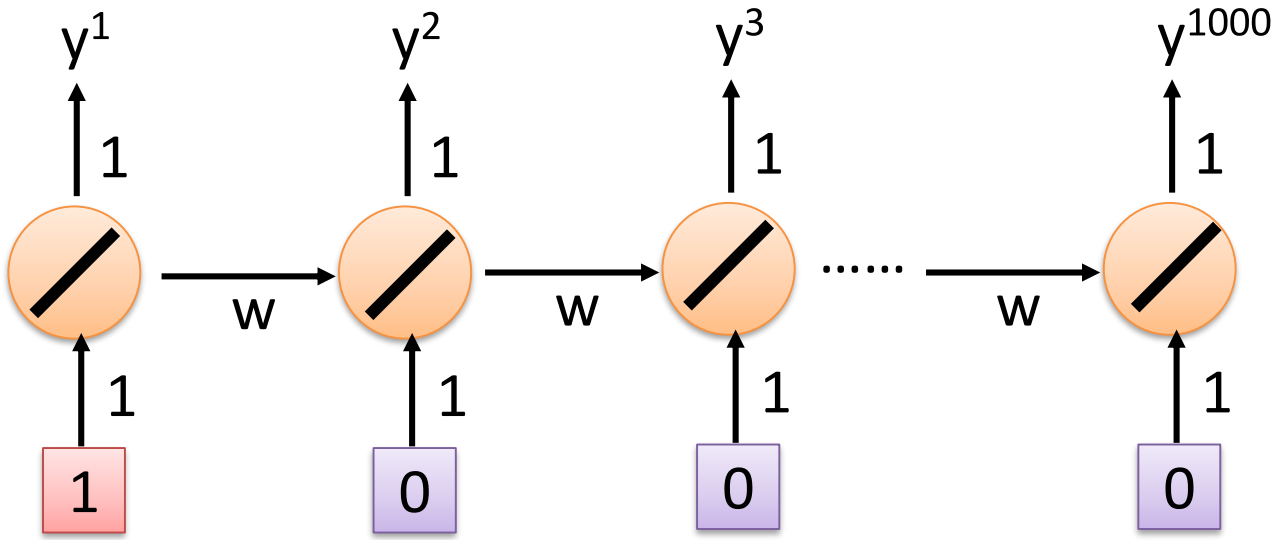
Large $\partial L/\partial w$ ➡ Small Learning rate?

$w = 0.99 \Rightarrow y^{1000} \approx 0$

$w = 0.01 \Rightarrow y^{1000} \approx 0$

small $\partial L/\partial w$ ➡ Large Learning rate?

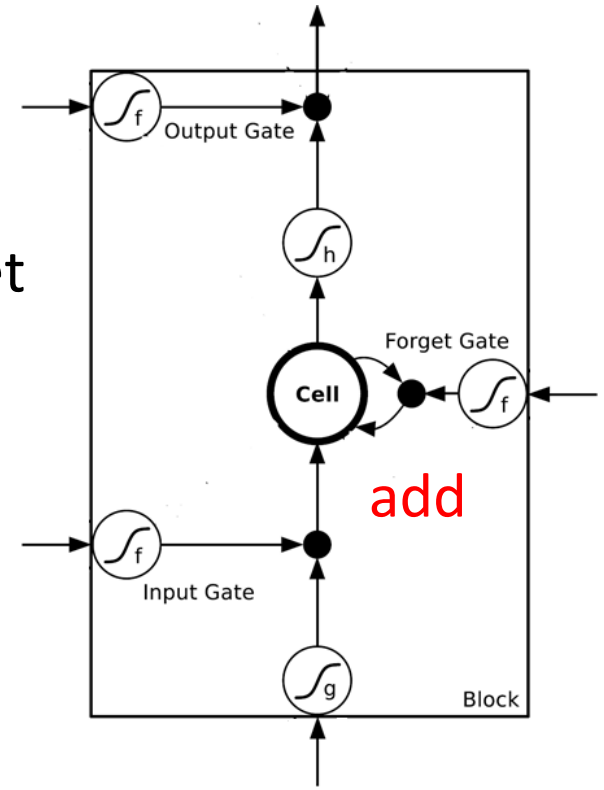
Toy Example



回顾： Helpful Techniques

- Long Short-term Memory (LSTM)
  - Can deal with gradient vanishing (not gradient explode)
    - Memory and input are ***added***
    - The influence never disappears unless forget gate is closed

No Gradient vanishing
(If forget gate is opened.)

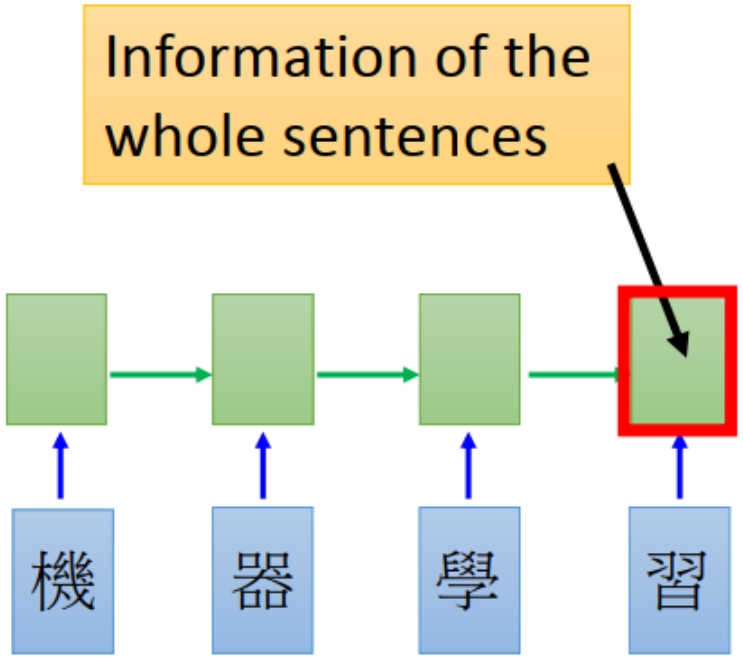


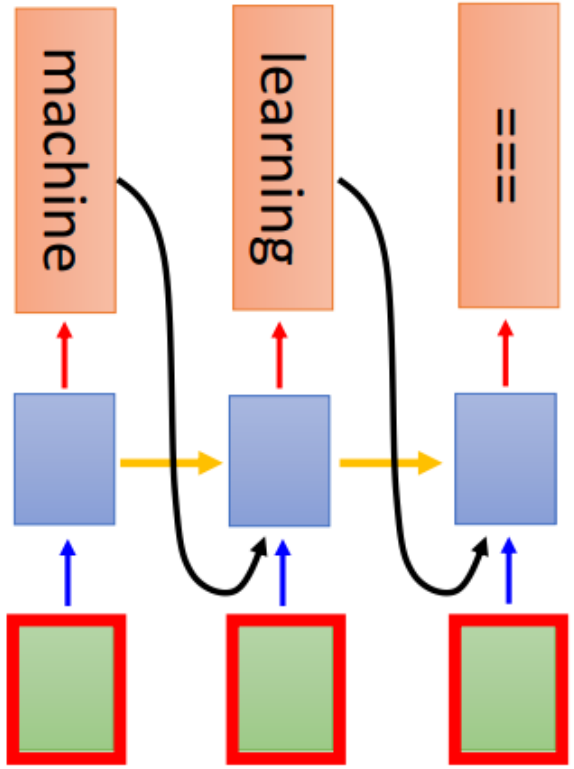
回顾： 机器翻译

- Sequence to sequence learning: Both input and output are both sequences ***with different lengths***.
- E.g. 機器學習→machine learning

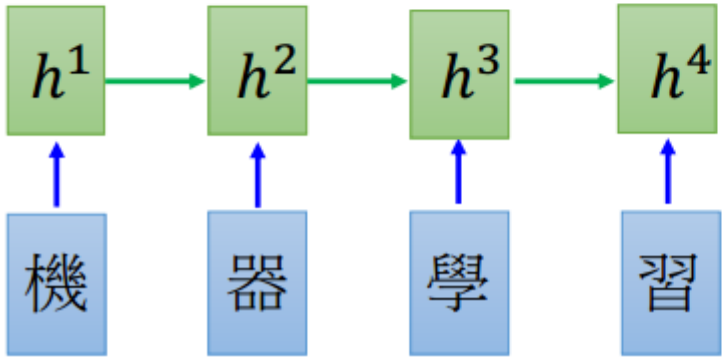
存在什么问题?

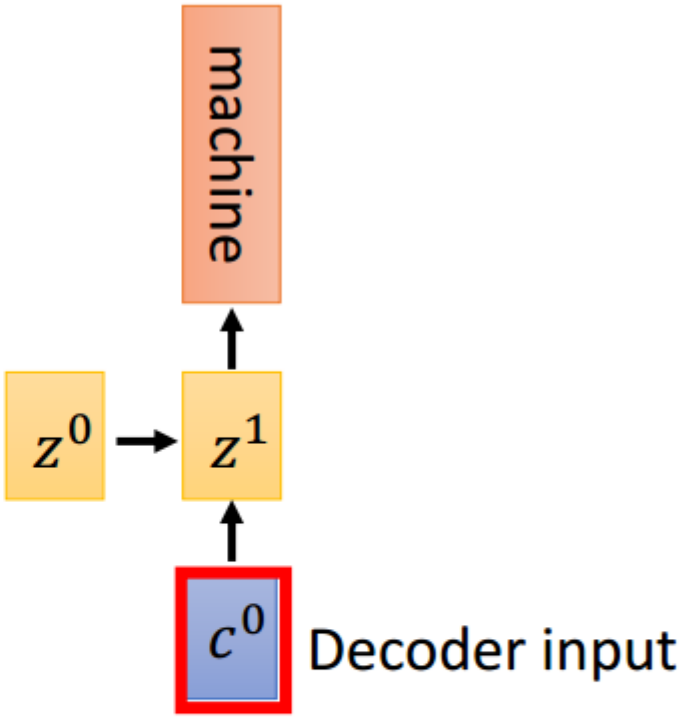
- 编码有效性
- 训练效率





回顾：Machine Translation

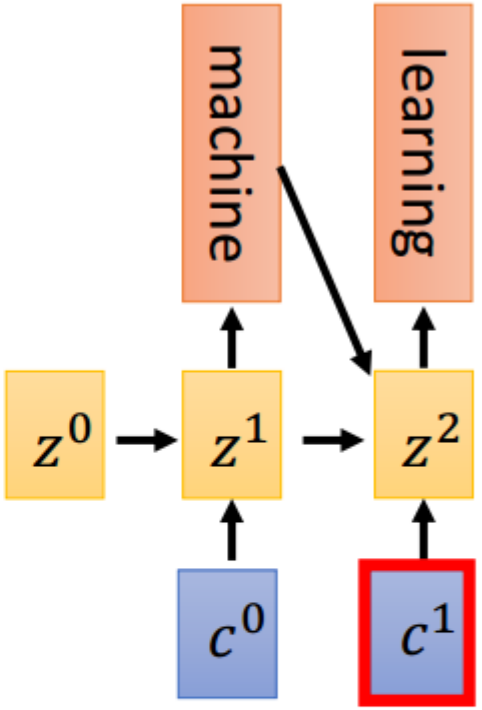


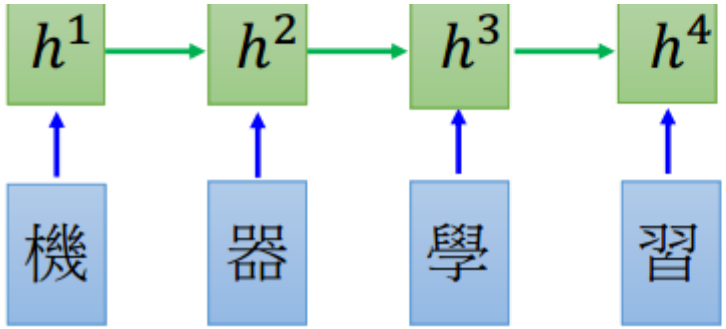


$$c^0 = \sum \hat{\alpha}_0^i h^i$$

$$= 0.5h^1 + 0.5h^2$$

回顾：Machine Translation



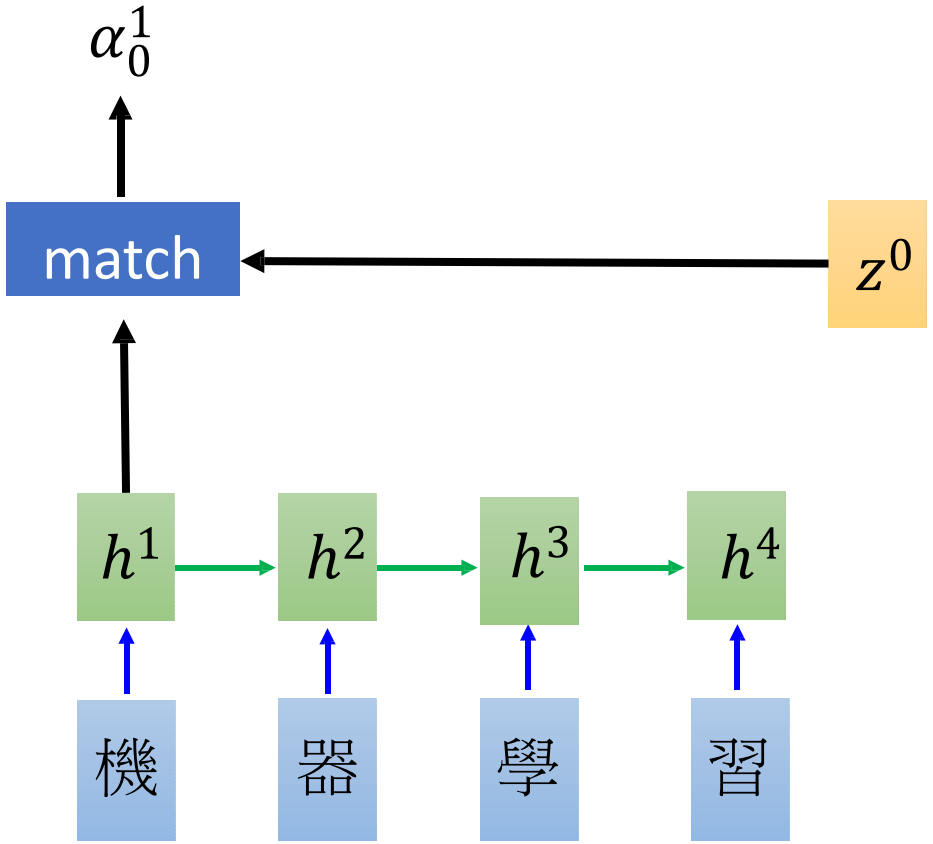


$$c^1 = \sum \hat{\alpha}_1^i h^i$$

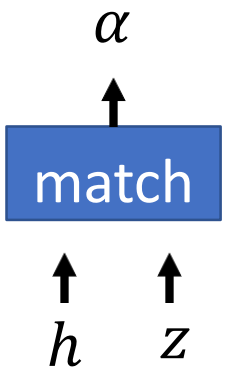
$$= 0.5h^3 + 0.5h^4$$

回顾：Machine Translation

- Attention-based model

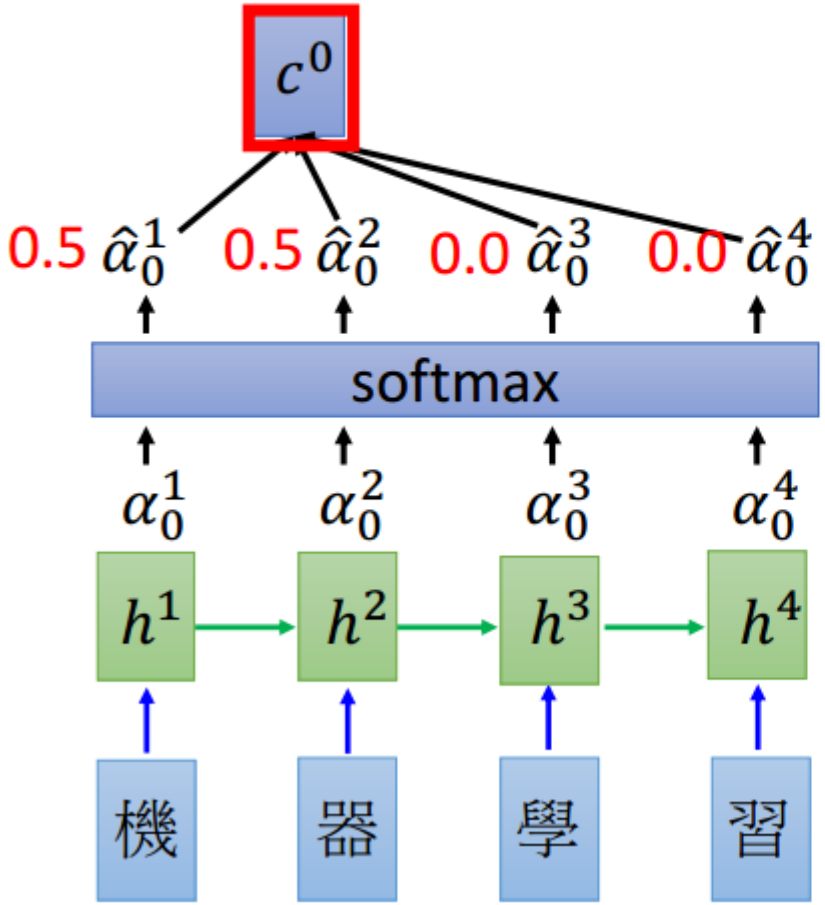


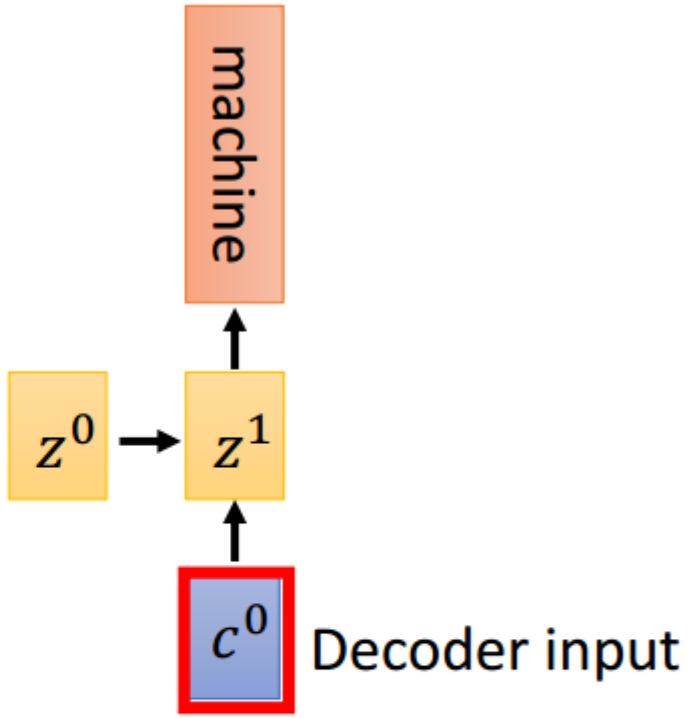
Jointly learned with other part of the network



回顾：Machine Translation

- Attention-based model



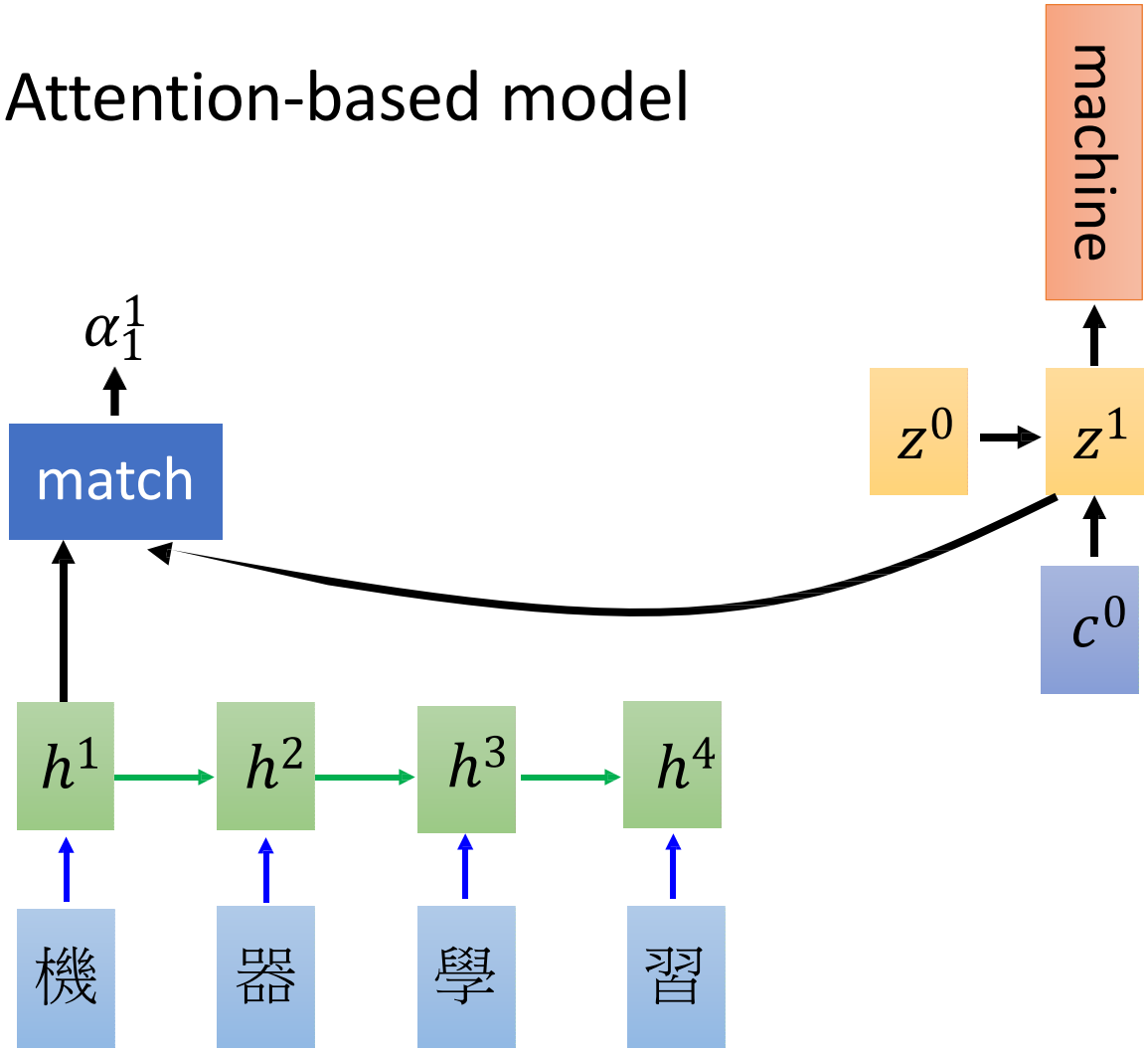


$$c^0 = \sum \hat{\alpha}_0^i h^i$$

$$= 0.5h^1 + 0.5h^2$$

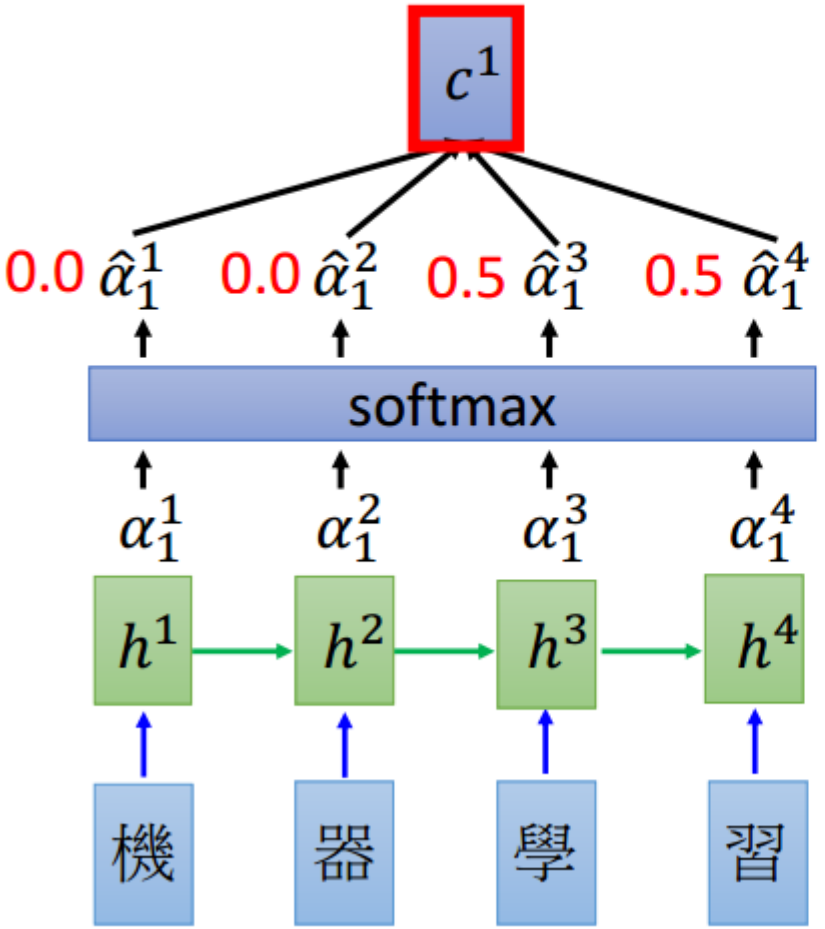
回顾：Machine Translation

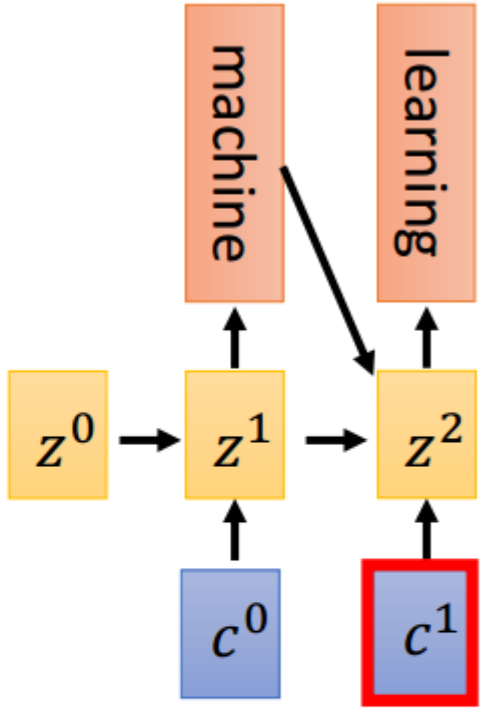
- Attention-based model



回顾：Machine Translation

- Attention-based model





$$c^1 = \sum \hat{\alpha}_1^i h^i$$

$$= 0.5h^3 + 0.5h^4$$

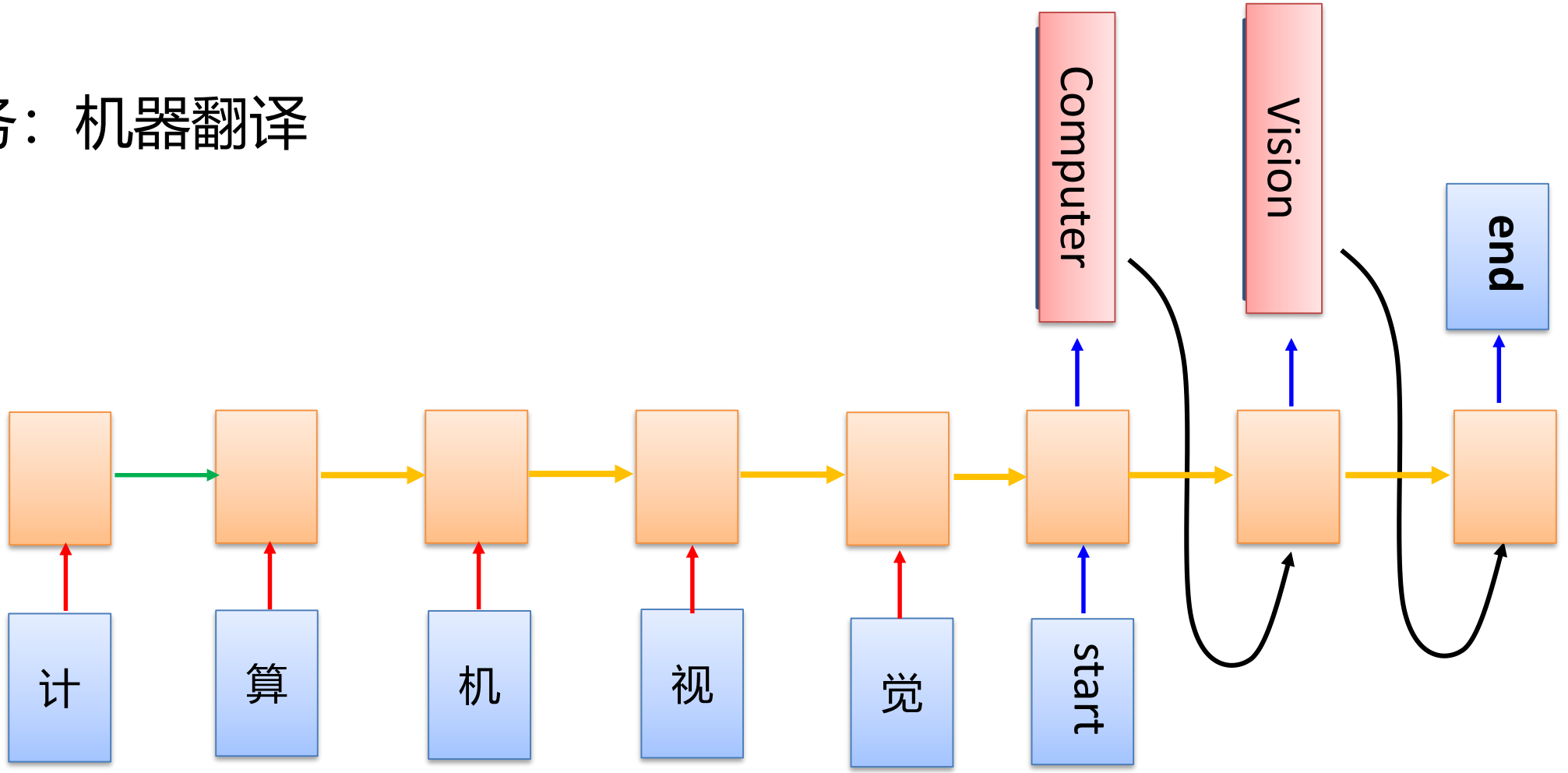
今日主题

- Transformer
- Non-Local 模块
- ViT
- MAE

今日主题

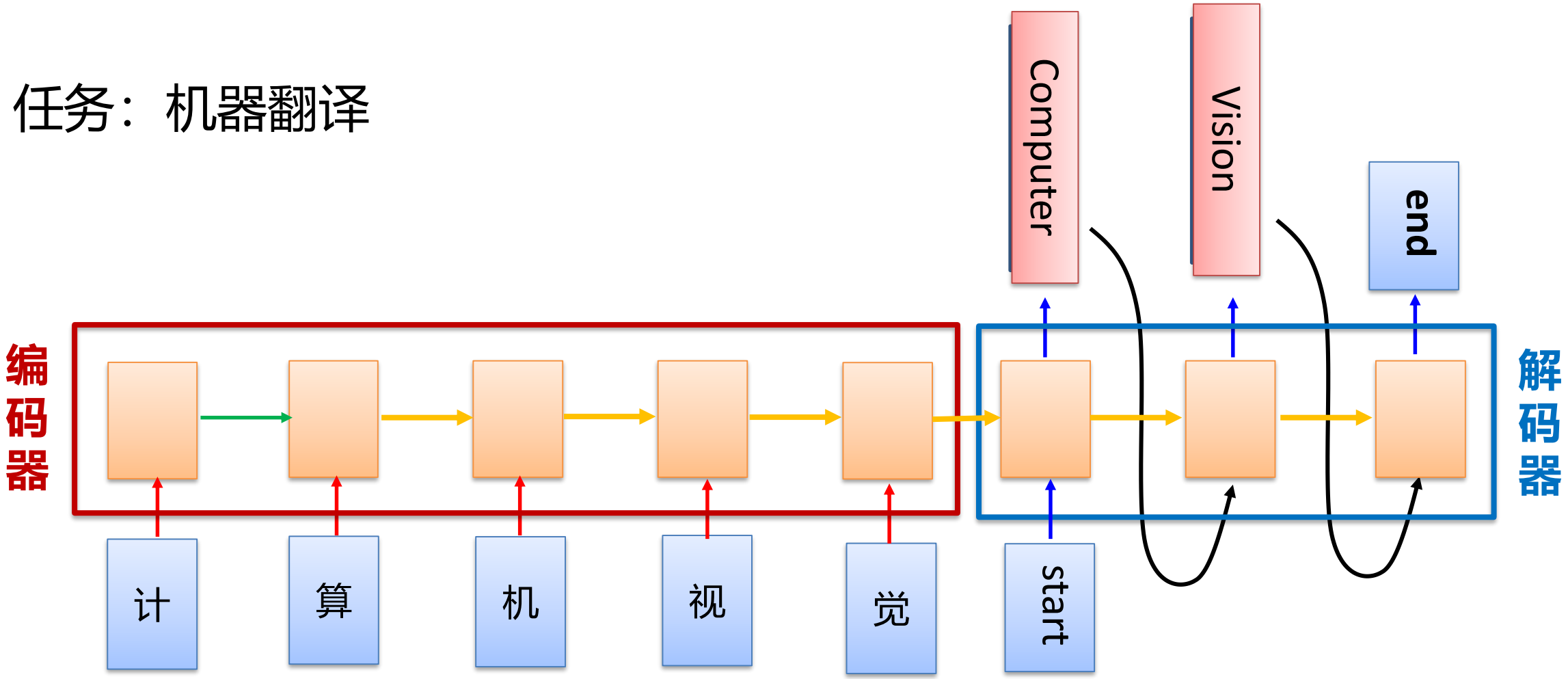
- Transformer
- Non-Local 模块
- ViT
- MAE

任务：机器翻译



[Ilya Sutskever, NIPS'14][Dzmitry Bahdanau, arXiv'15]

任务：机器翻译



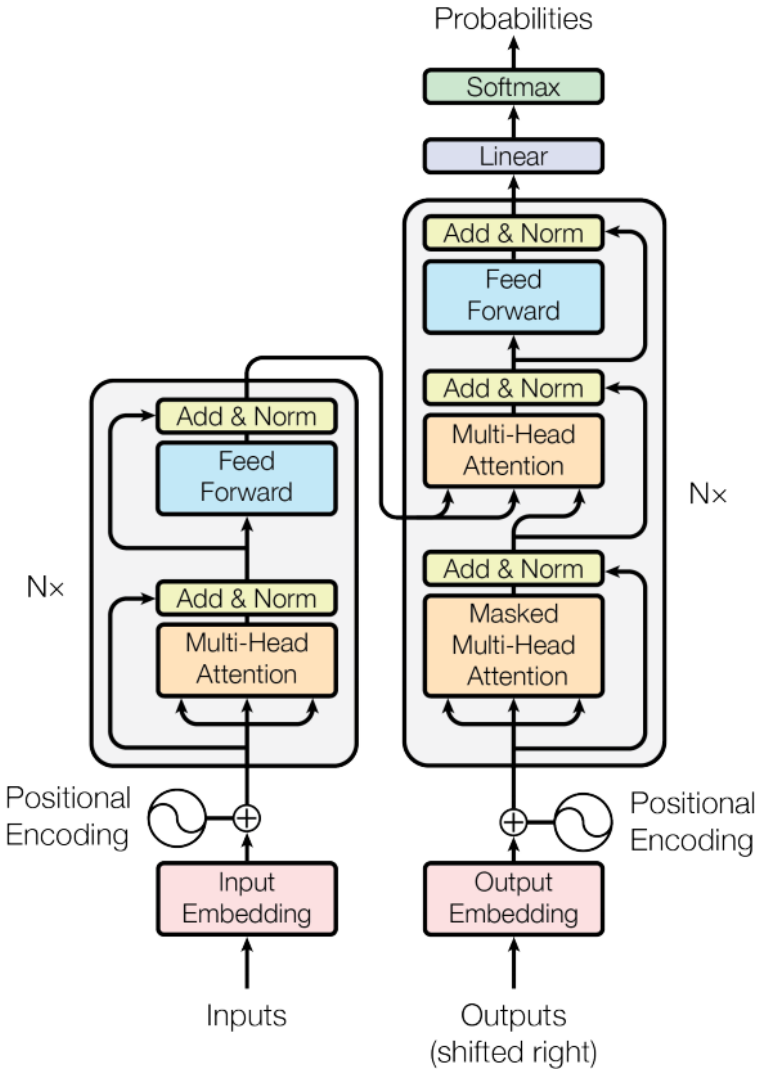
RNN的困境！

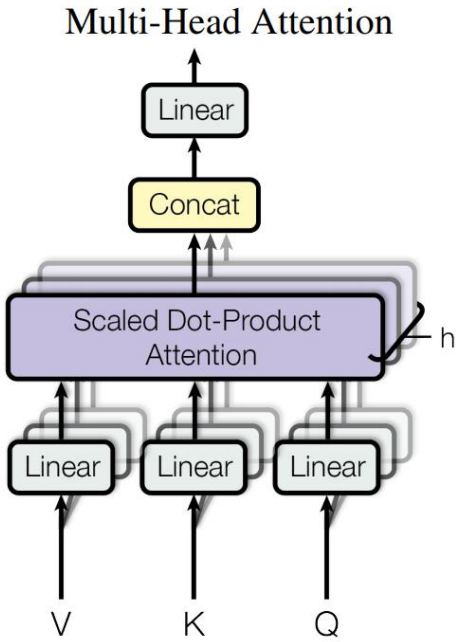
[Ilya Sutskever, NIPS'14][Dzmitry Bahdanau, arXiv'15]

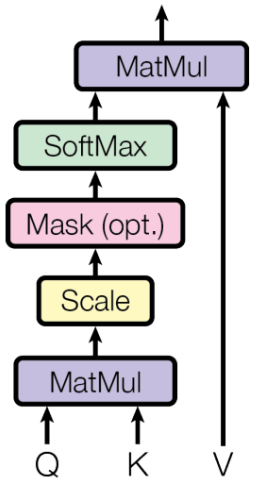
任务：机器翻译


INPUT
Je suis étudiant
THE TRANSFORMER
OUTPUT
I am a student

Transformer

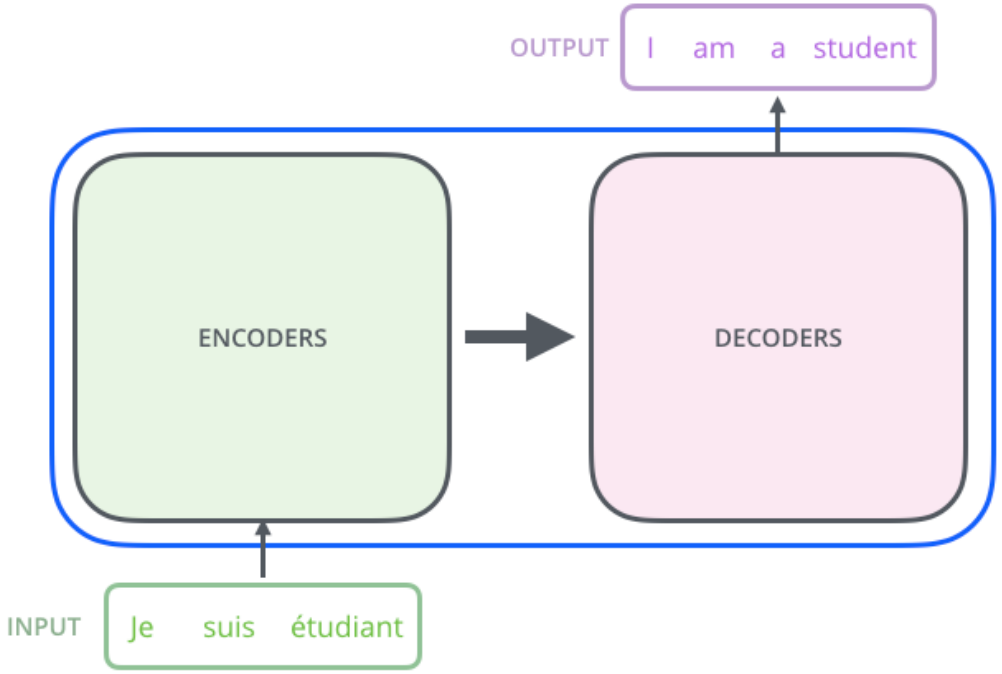


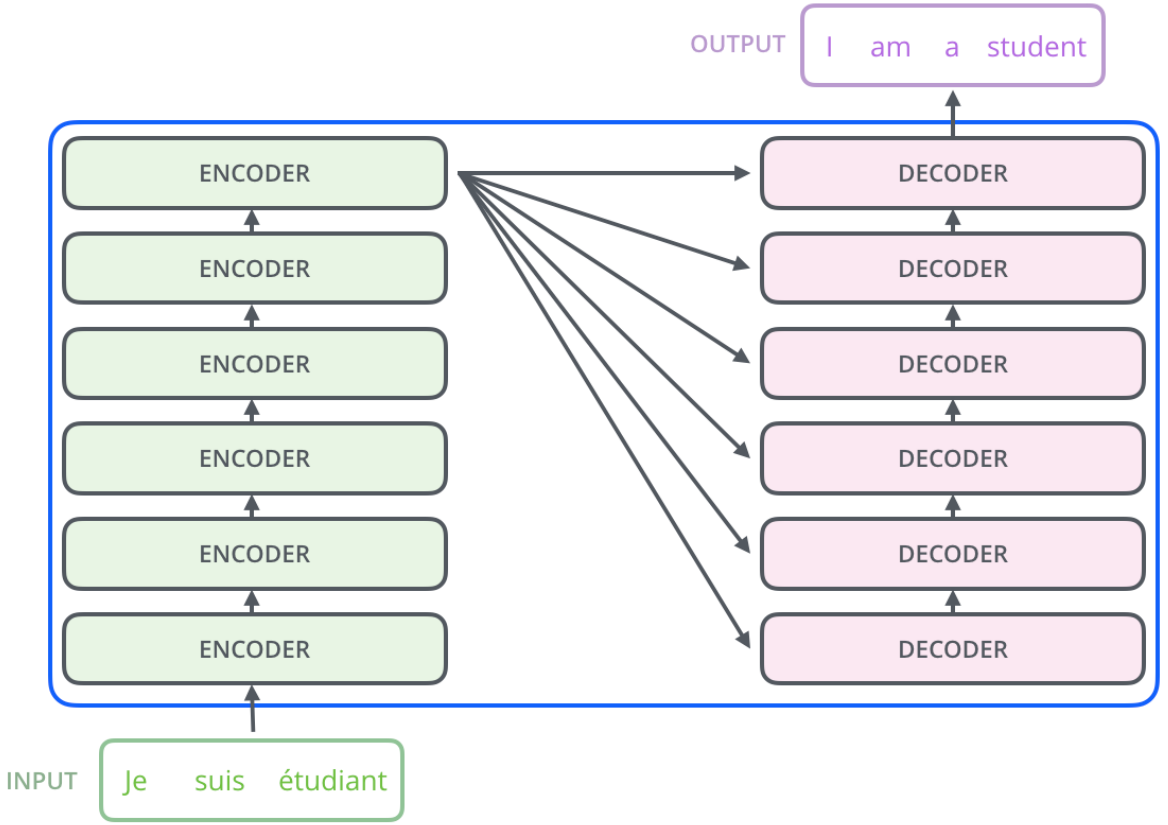




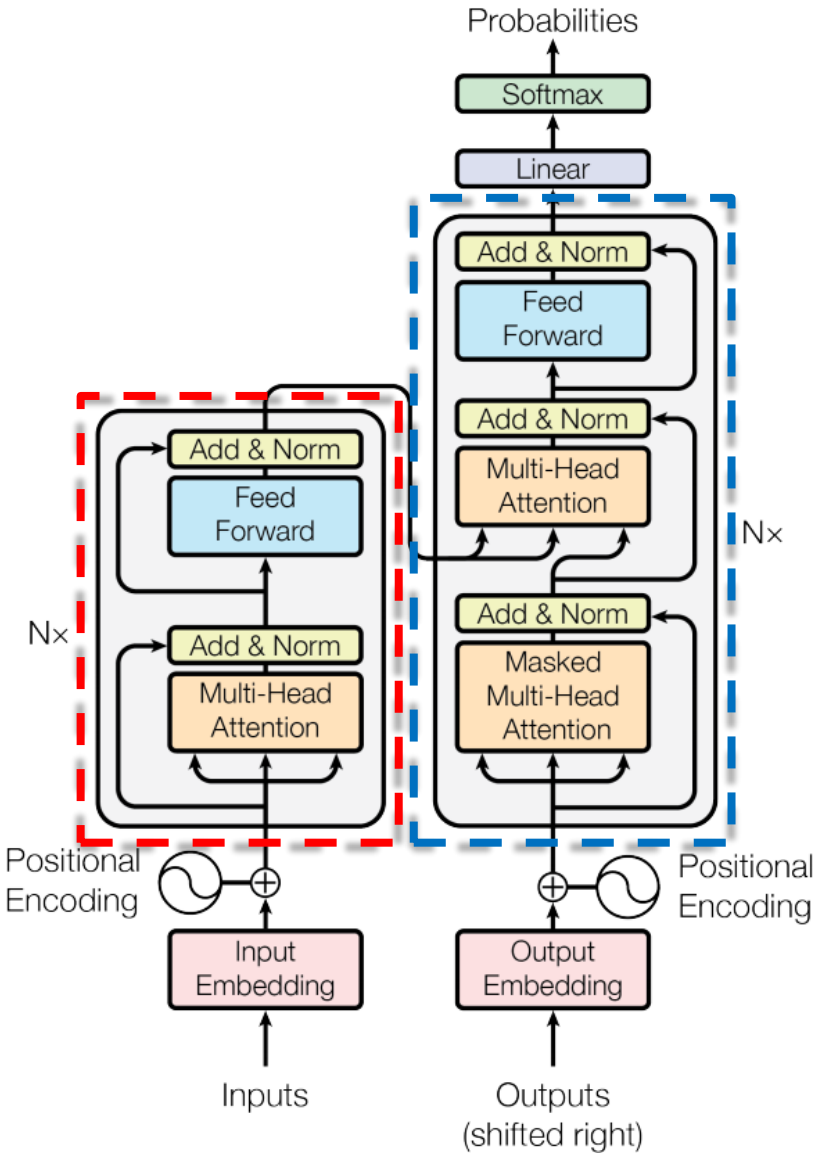
- 编解码器层数： N
- 嵌入维度： d_{model}
- 多头数: h
- 前馈网络第一层宽度： d_{ff}

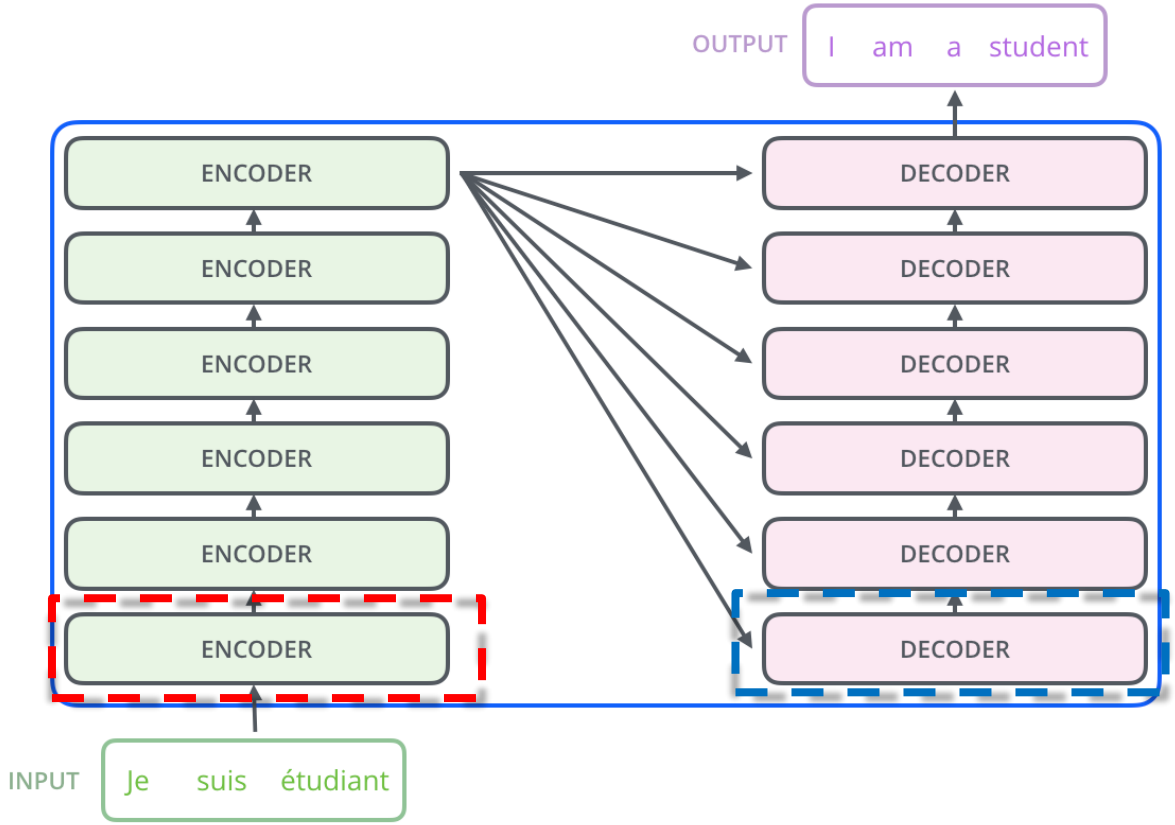
Transformer 基本组成





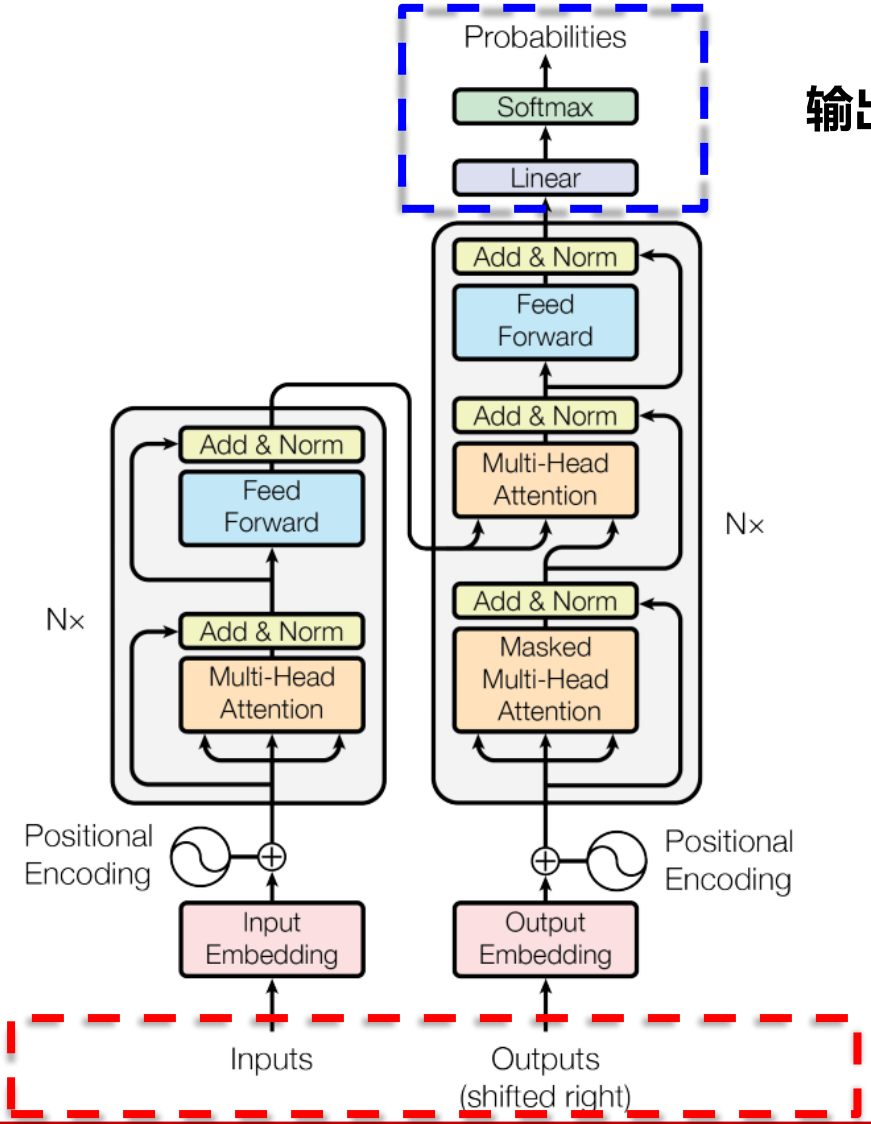
Transformer（Attention Is All You Need）





原始论文：编解码器均为6层，即 $N = 6$

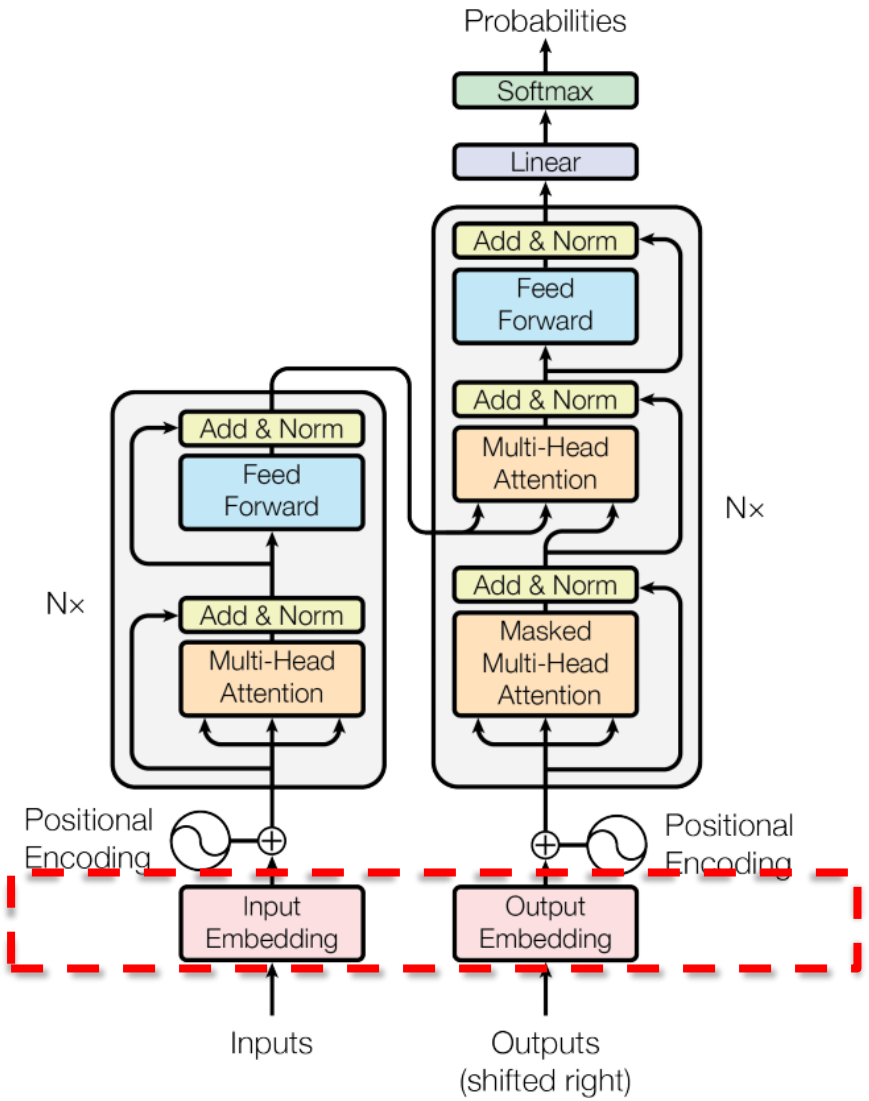
Transformer的输入与输出



输出：单词的概率

输入：待翻译句子+已翻译词汇

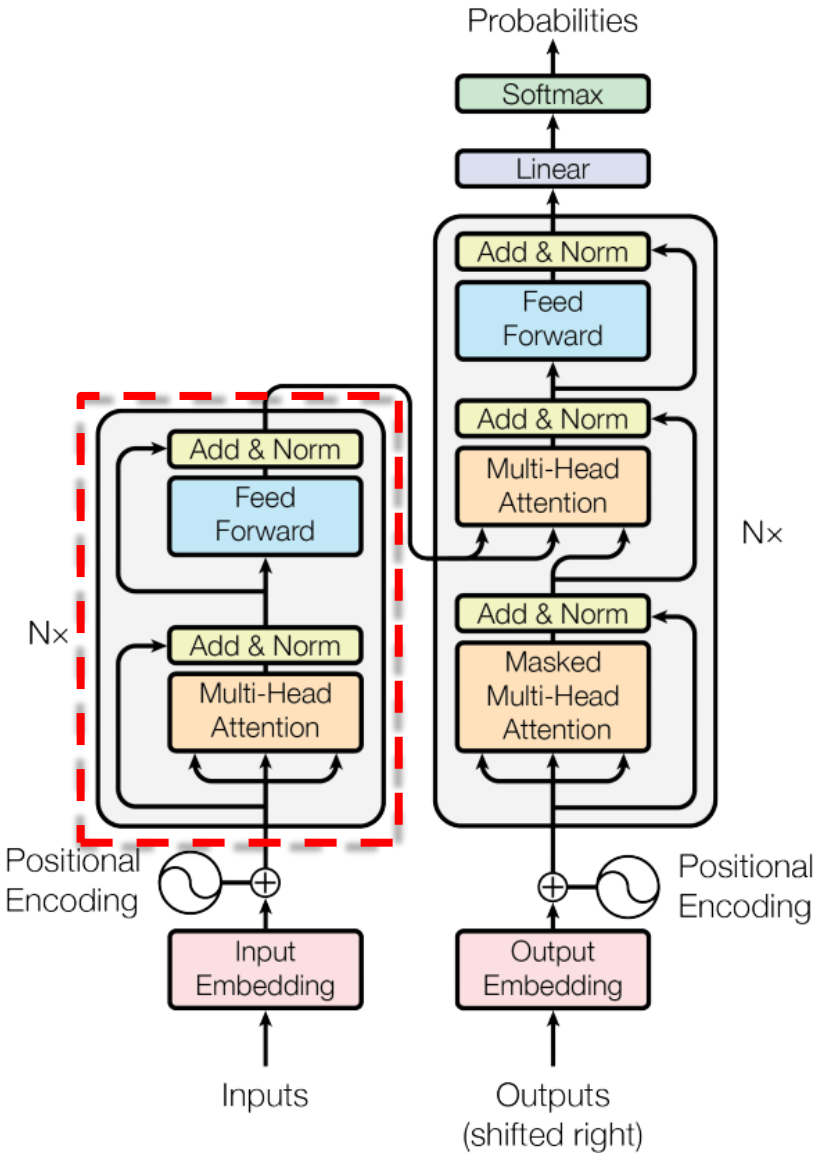
Transformer输入的嵌入

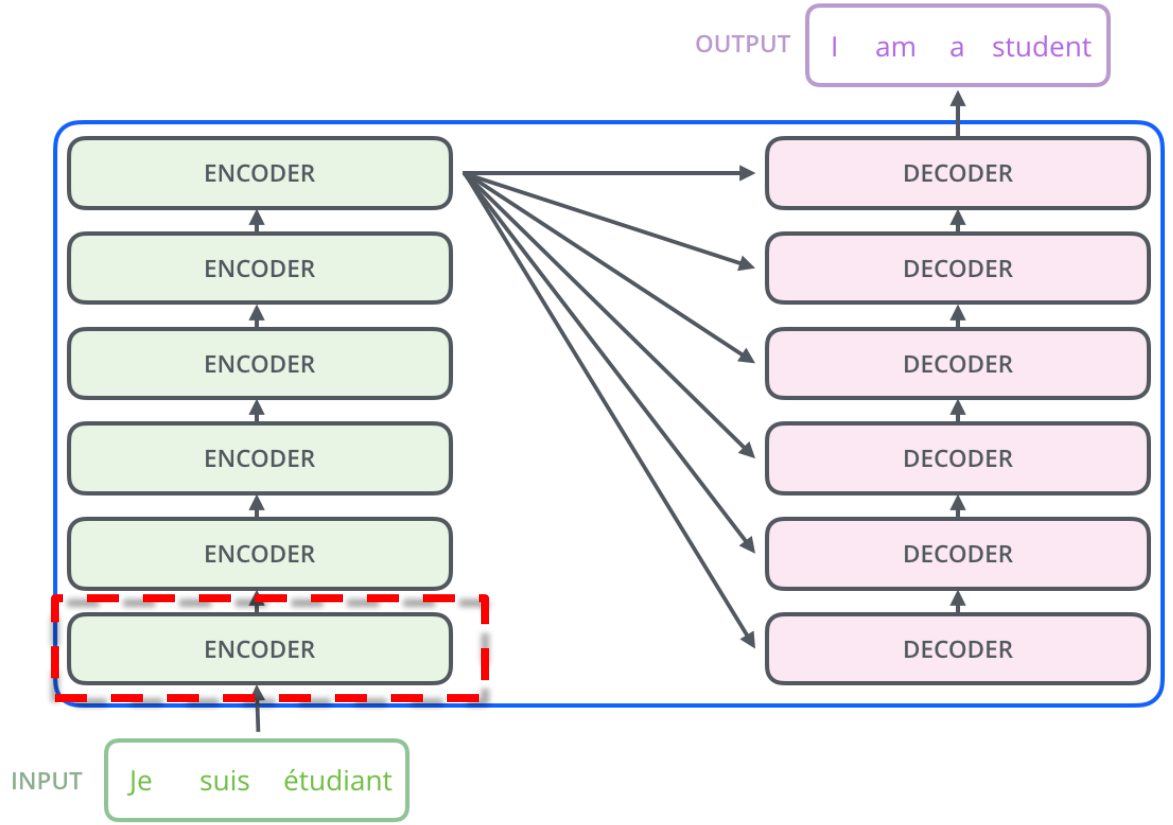


嵌入层：通过一个变换将单词的one-hot表示映射到连续空间上，其维度与模型维度d_{model}一致。可使用 nn.Embedding函数实现。

嵌入维度：d_{model} =512

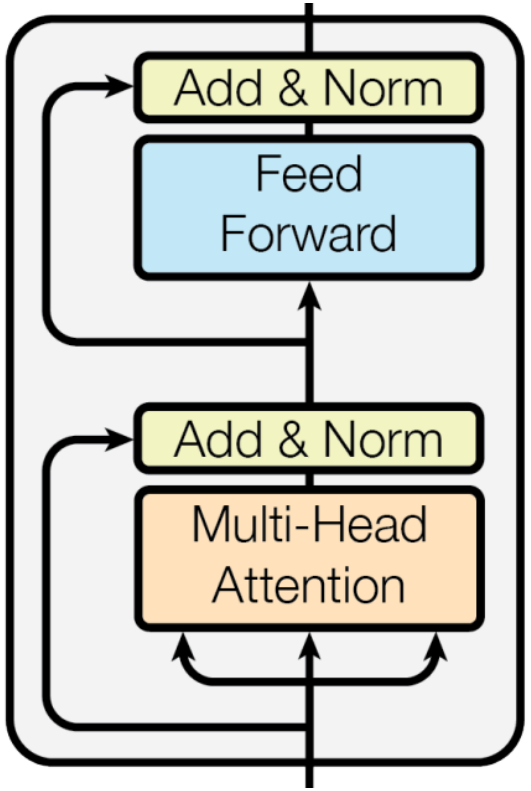
Transformer（Attention Is All You Need）

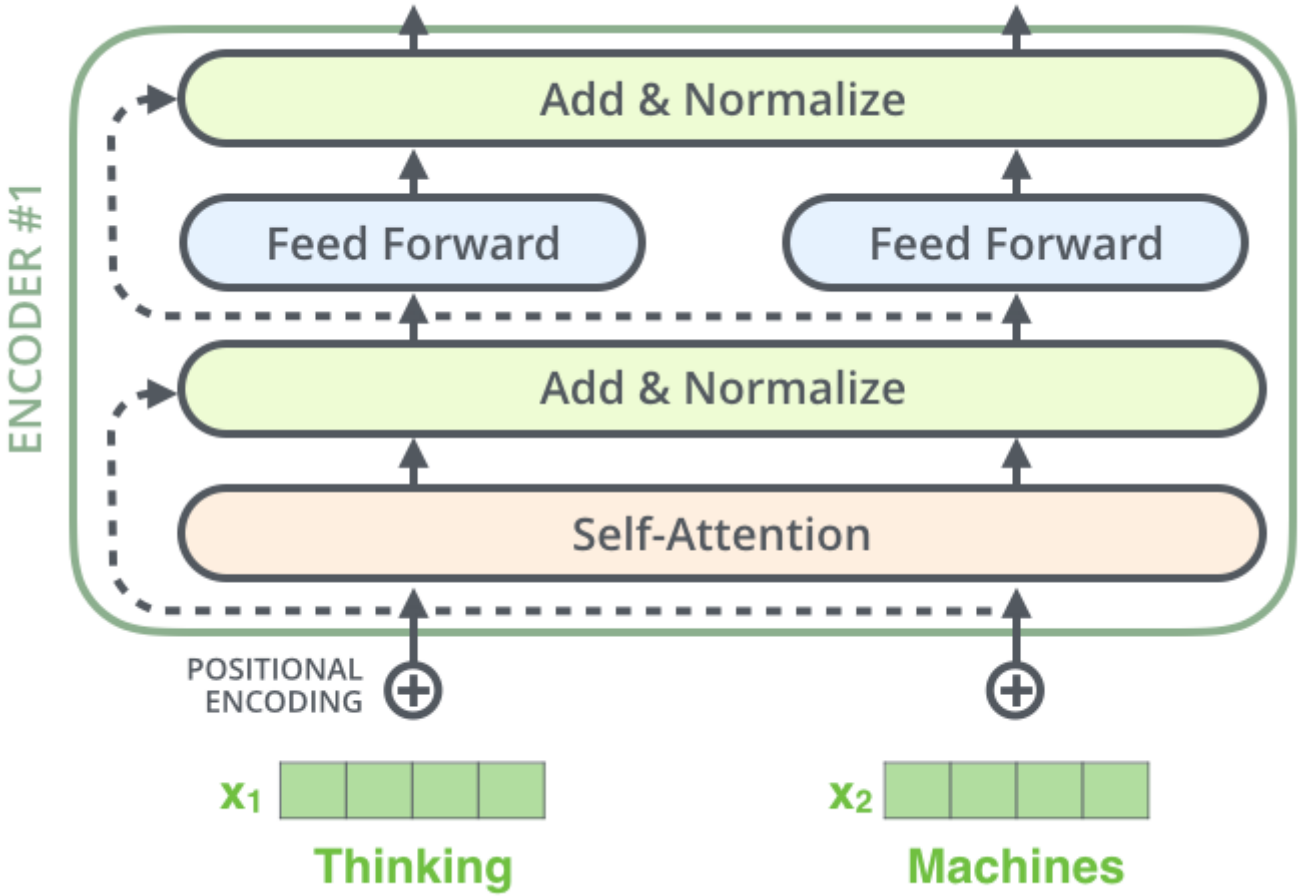




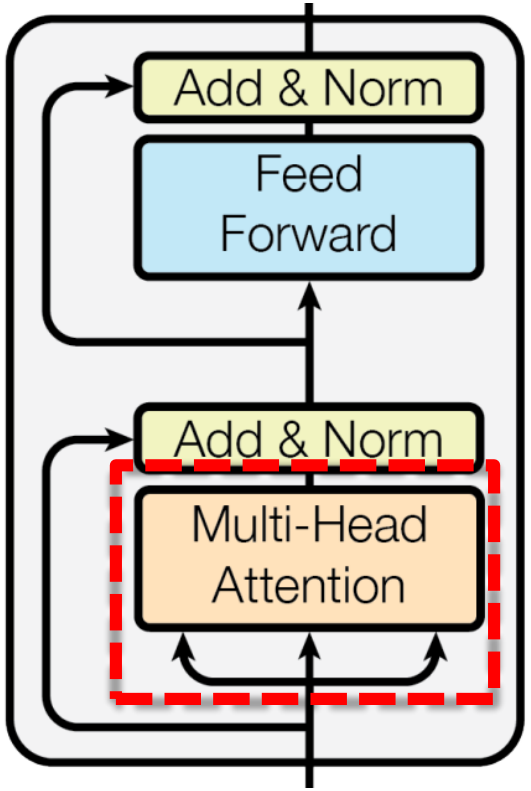
原始论文：堆叠了6层编码器、6层解码器

Transformer的编码器

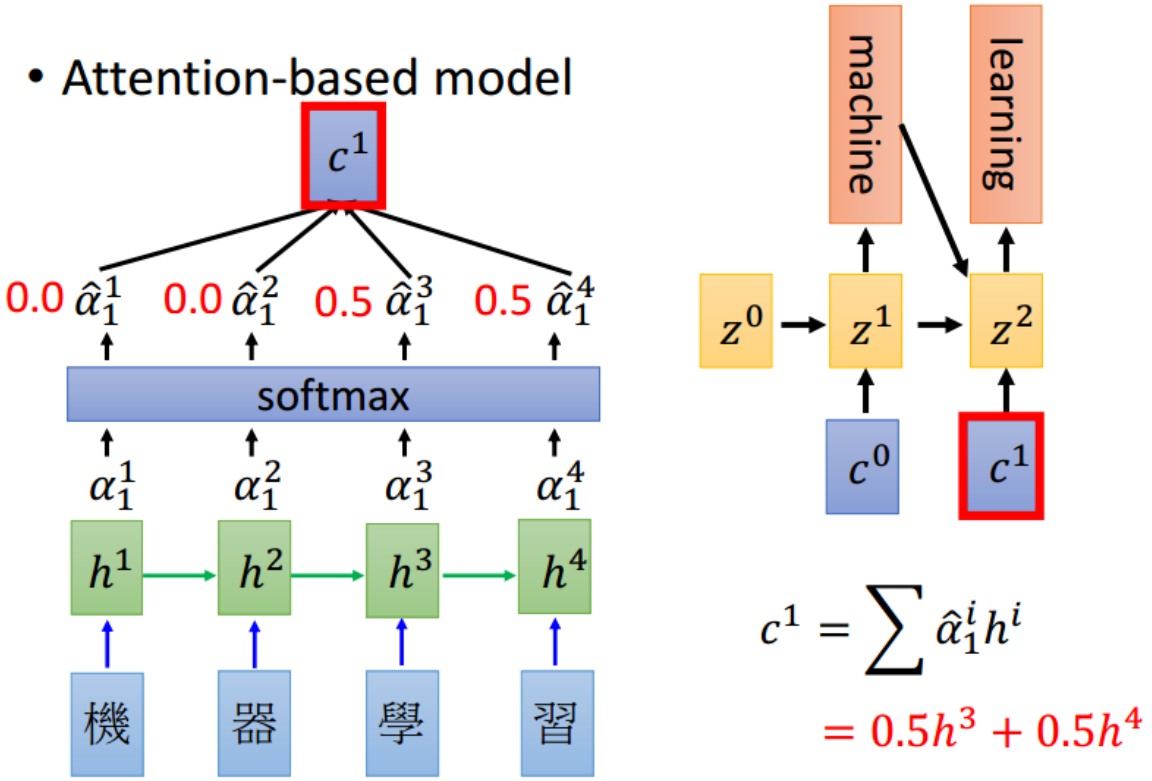




编码器中的多头注意力

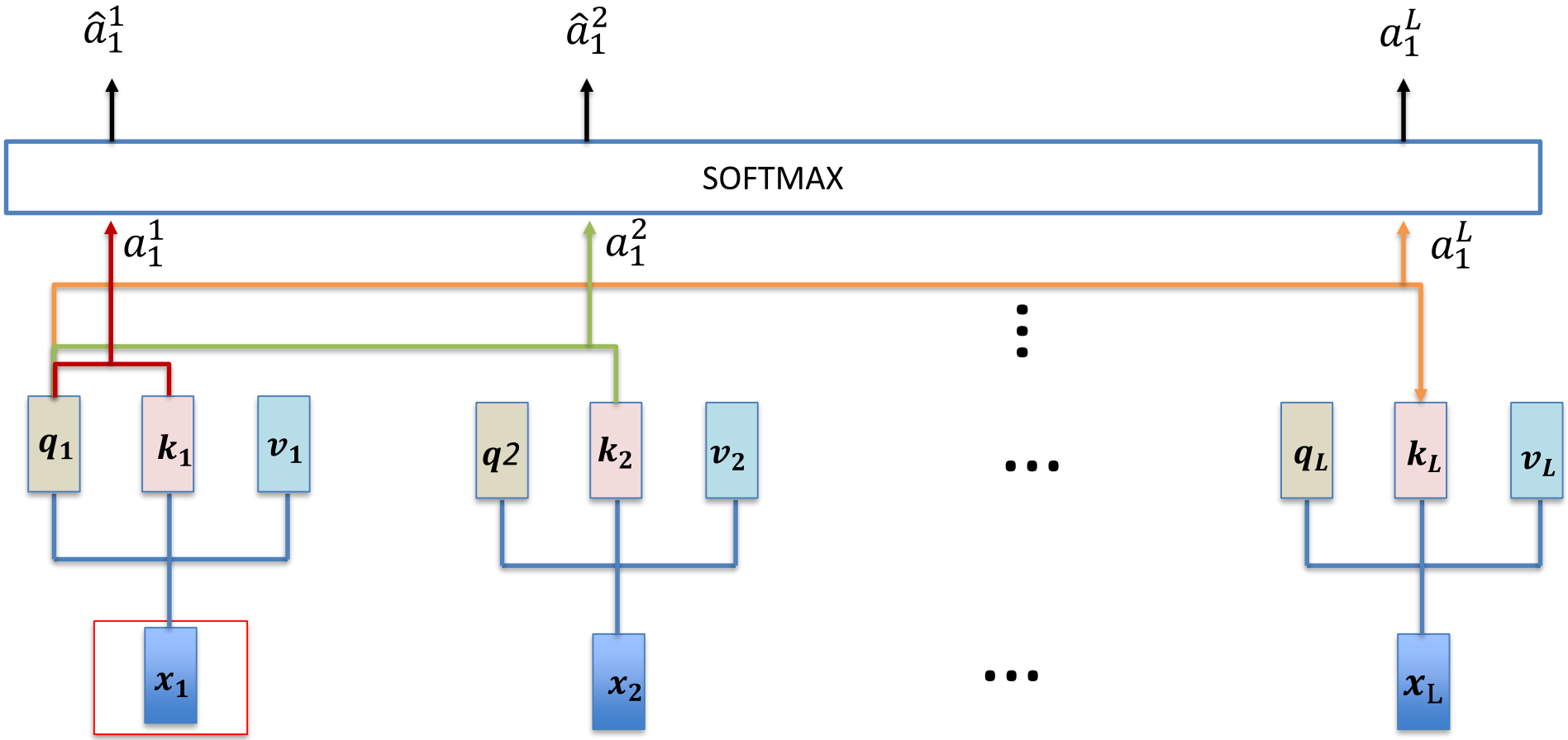


回顾：RNN中的注意力机制

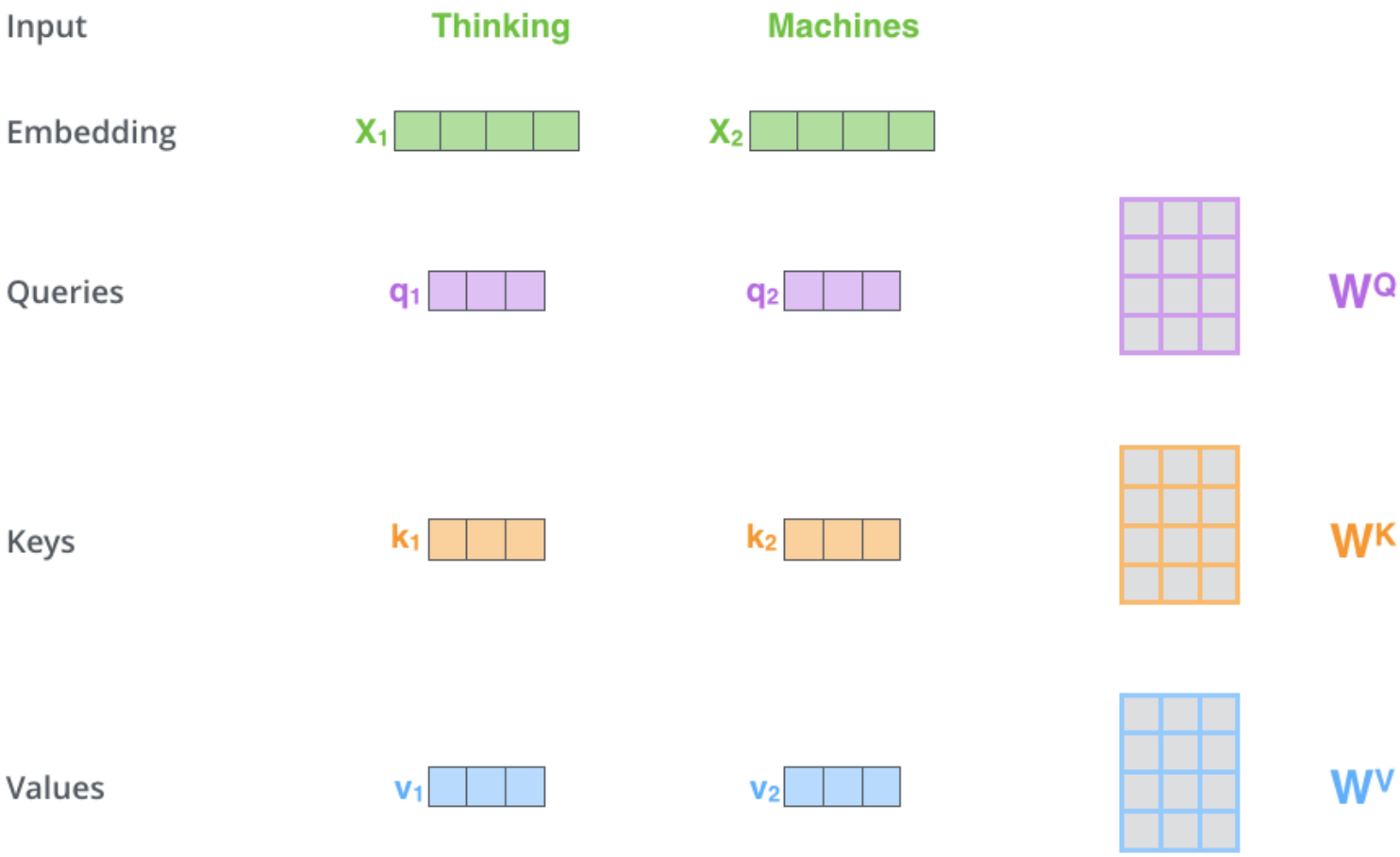


编码器中的注意力

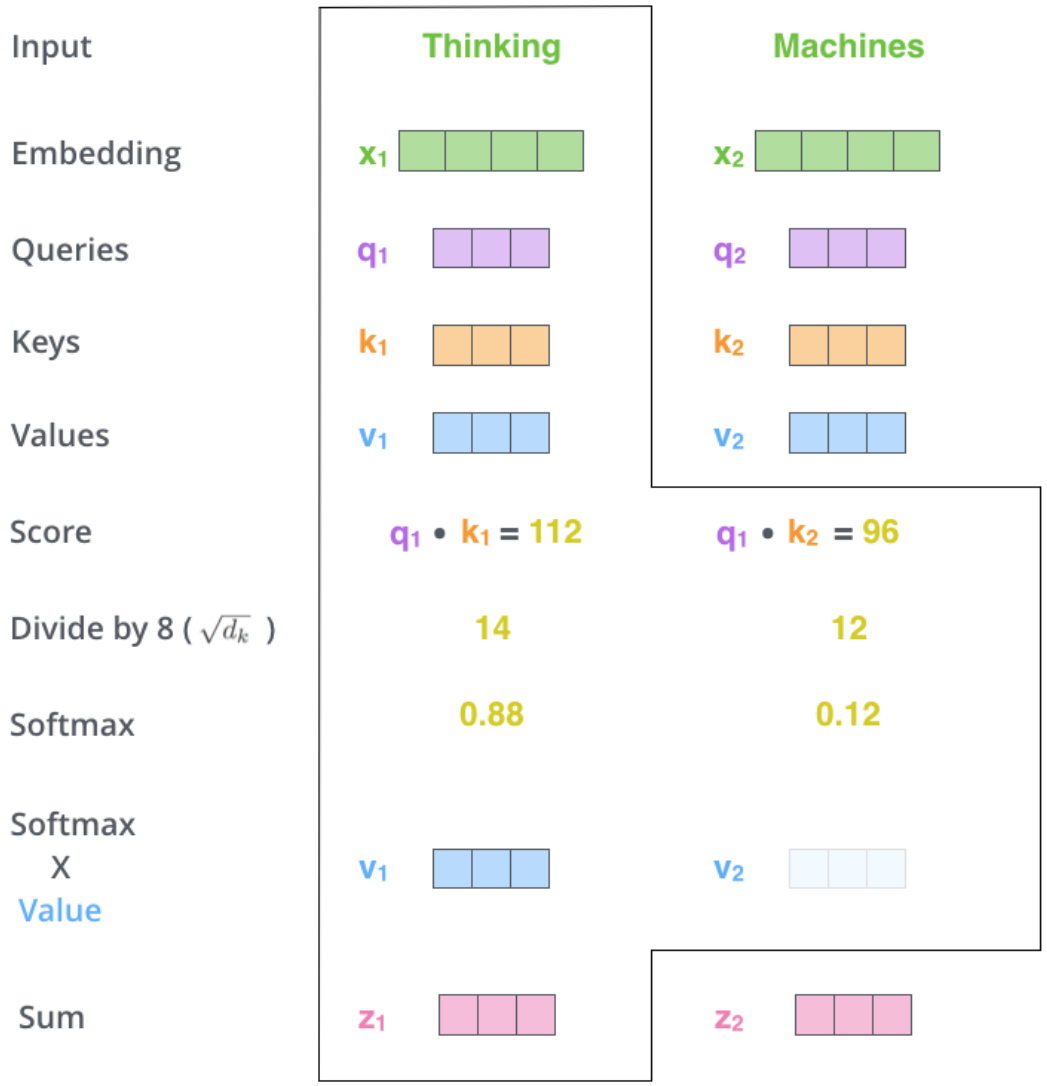
$$\hat{\boldsymbol{x}}_{\boldsymbol{i}} = \sum_{j=1}^{L} a_{\mathrm{i}}^{j} \times \boldsymbol{x}_{\boldsymbol{j}}$$

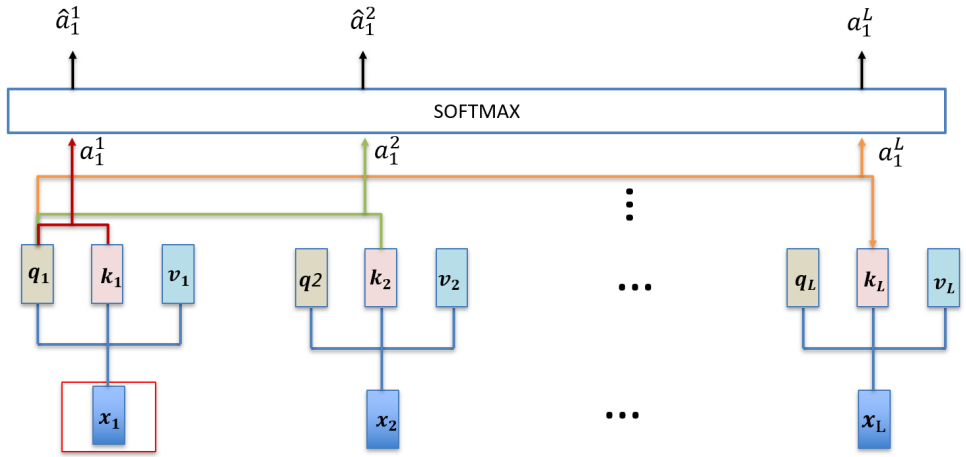


编码器中的注意力

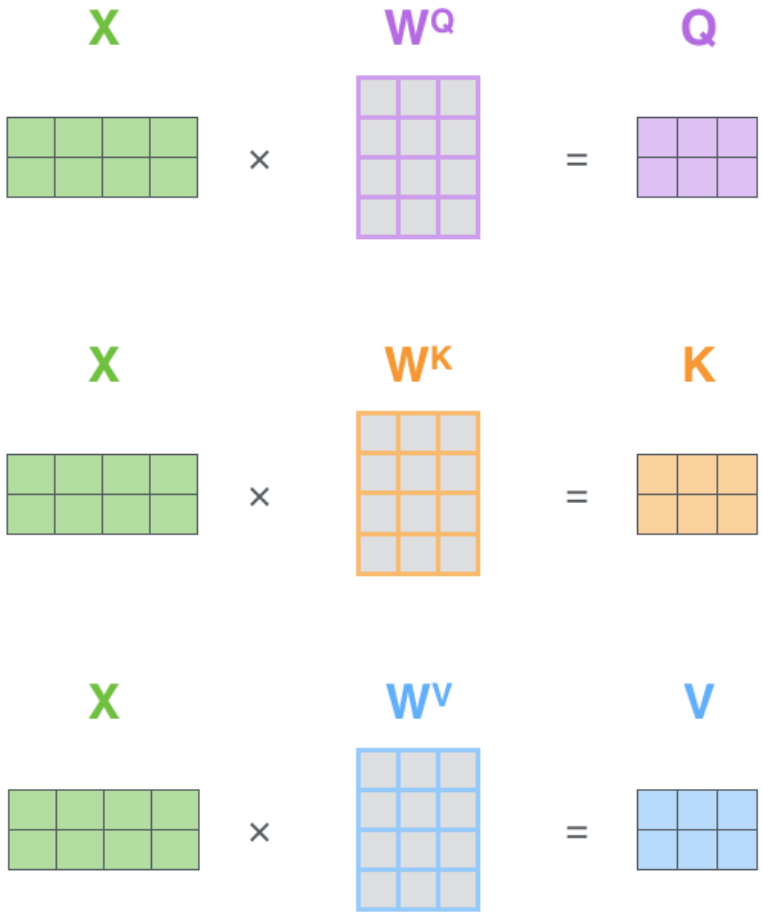

Input
Thinking
Machines
Embedding
X1
X2
Queries
q1
q2
WQ
Keys
k1
k2
WK
Values
v1
v2
WV

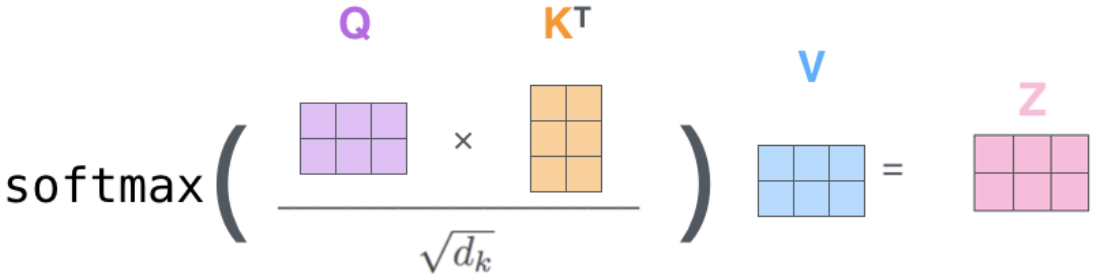
编码器中的注意力

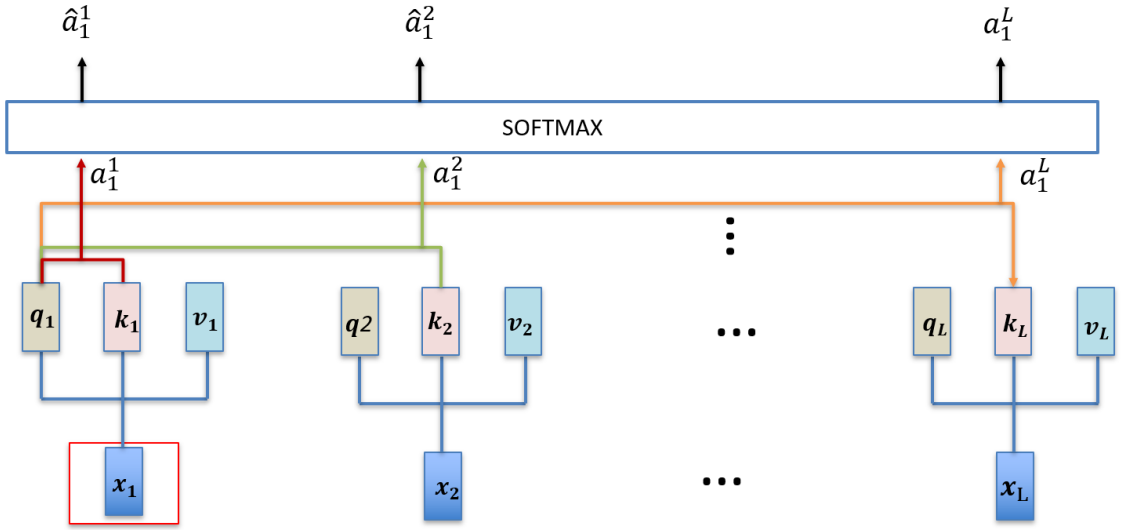

Input
Thinking
Machines
Embedding
x1
x2
Queries
q1
q2
Keys
k1
k2
Values
v1
v2
Score
q1 • k1 = 112
q1 • k2 = 96
Divide by 8 (√dk)
14
12
Softmax
0.88
0.12
Softmax X Value
v1
v2
Sum
z1
z2


SOFTMAX

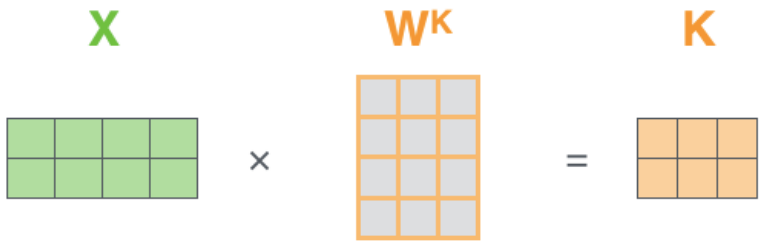
编码器中的注意力

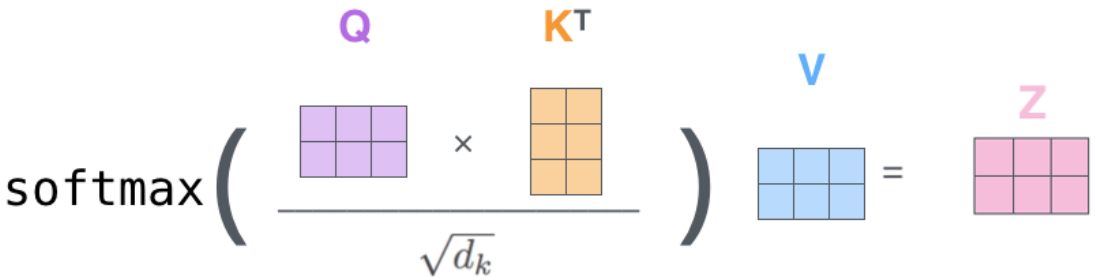

X
W^Q
Q
X
W^K
K
X
W^V
V


Q
K^T
V
Z
softmax
$\sqrt{d_k}$

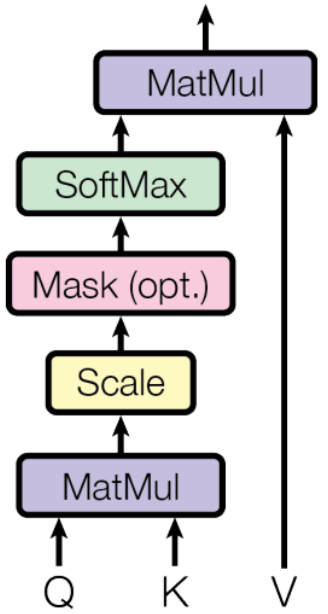

SOFTMAX
q_1
k_1
v_1
$q2$
k_2
v_2
q_L
k_L
v_L
x_1
x_2
x_L

编码器中的注意力

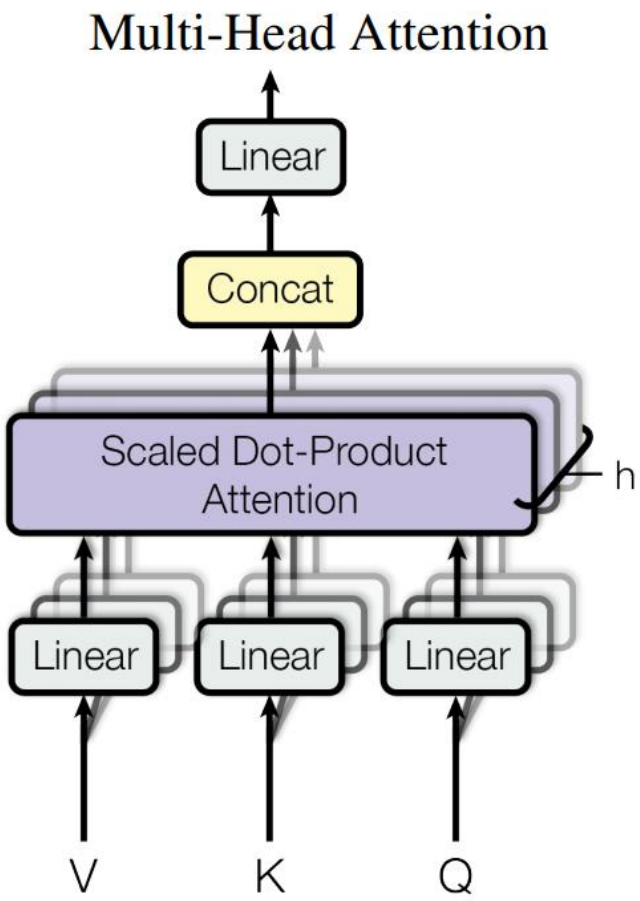


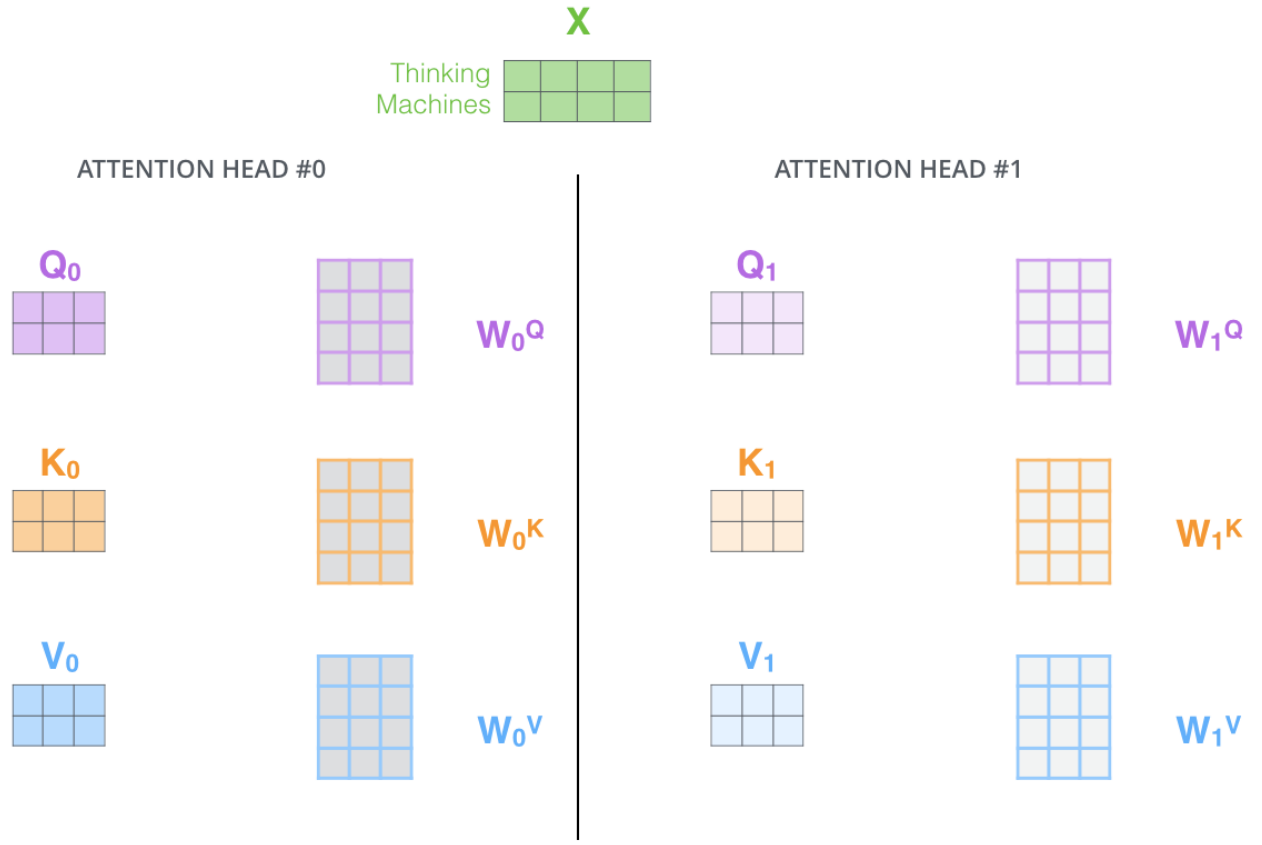


Scaled Dot-Product Attention

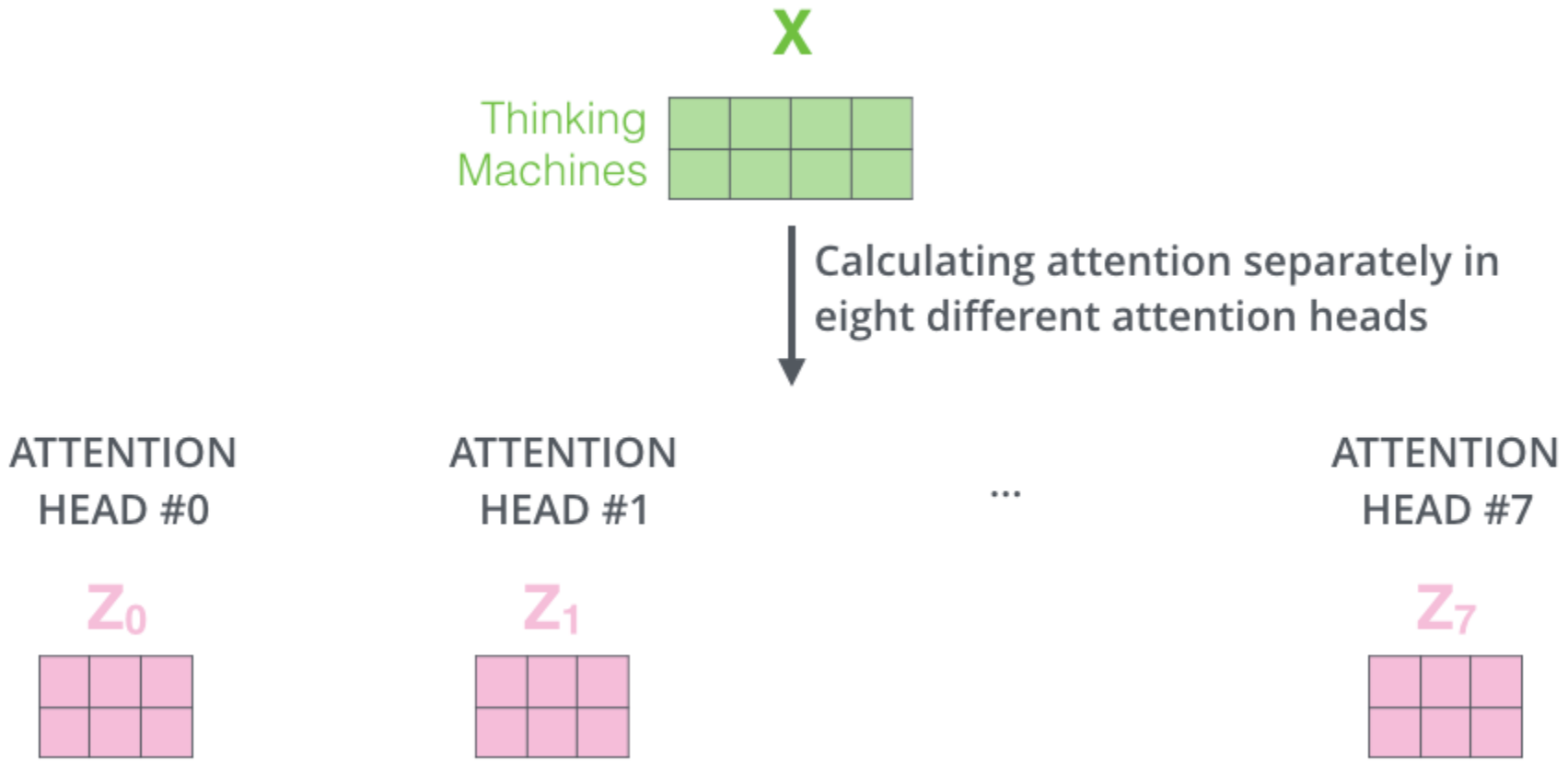


编码器中的多头注意力


Multi-Head Attention
Linear
Concat
Scaled Dot-Product Attention
h
Linear
Linear
Linear
V
K
Q

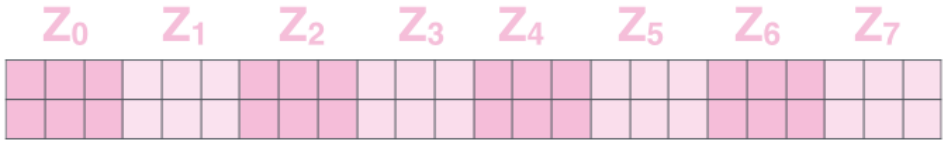

X
Thinking
Machines
ATTENTION HEAD #0
ATTENTION HEAD #1
Q_0
W_0^Q
Q_1
W_1^Q
K_0
W_0^K
K_1
W_1^K
V_0
W_0^V
V_1
W_1^V

编码器中的多头注意力


X
Thinking
Machines
Calculating attention separately in
eight different attention heads
ATTENTION
HEAD #0
ATTENTION
HEAD #1
...
ATTENTION
HEAD #7
Z0
Z1
Z7

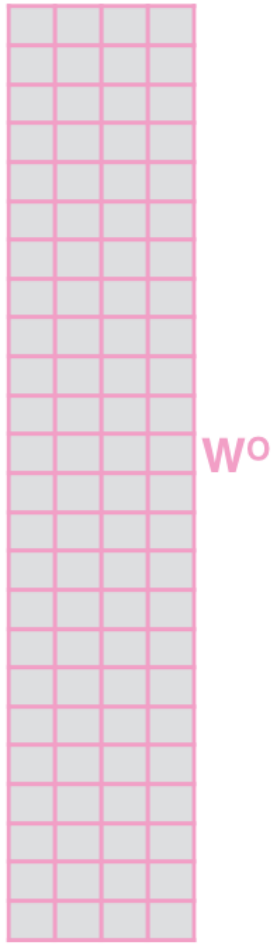
编码器中的多头注意力

1) Concatenate all the attention heads



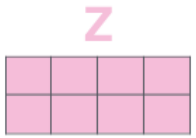
2) Multiply with a weight matrix W^O that was trained jointly with the model

X

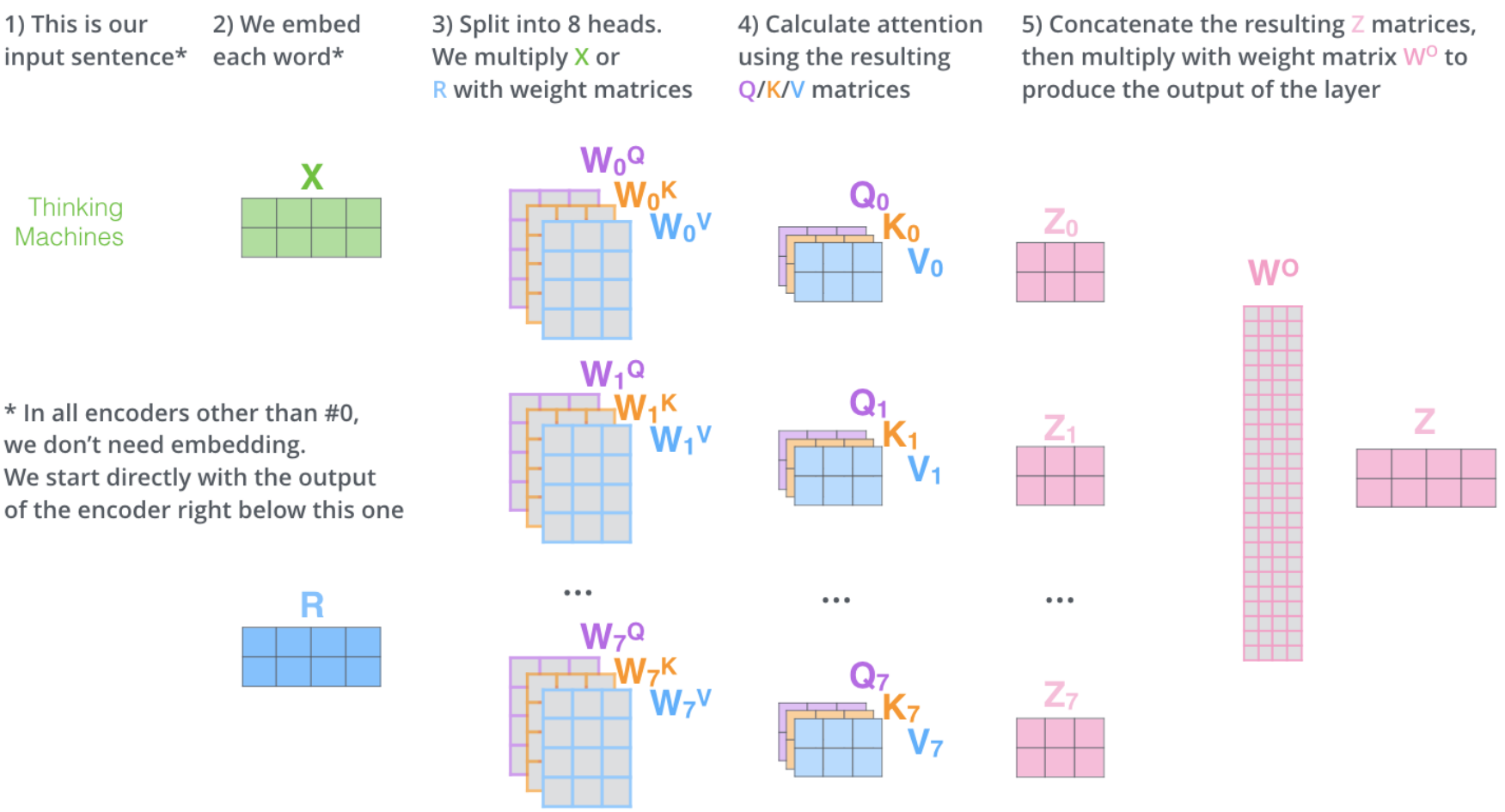


3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

=



编码器中的多头注意力


1) This is our input sentence*
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
Thinking
Machines
X
R
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one
W0Q W0K W0V
W1Q W1K W1V
W7Q W7K W7V
Q0 K0 V0
Q1 K1 V1
Q7 K7 V7
Z0
Z1
Z7
WO
Z

编码器中的多头注意力

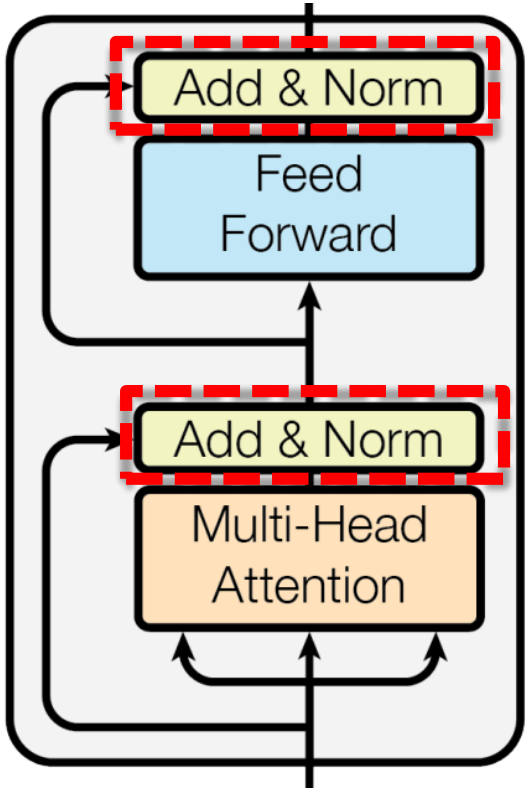
头数与维度关系

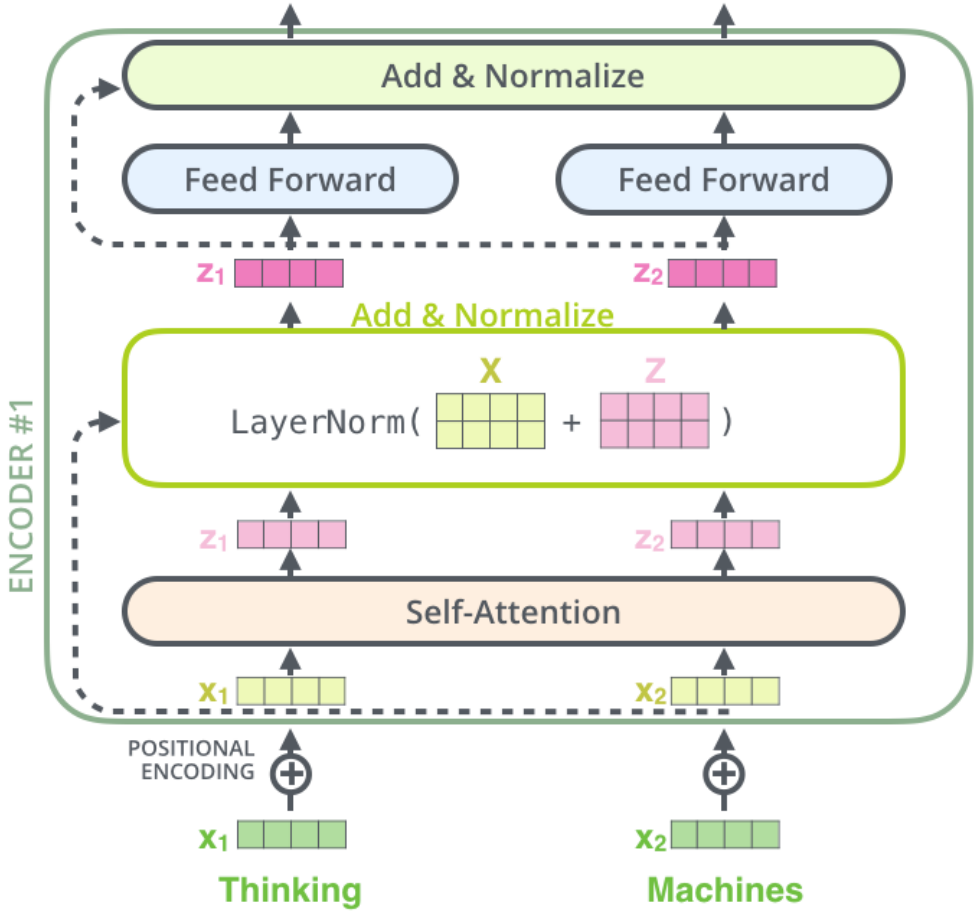
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

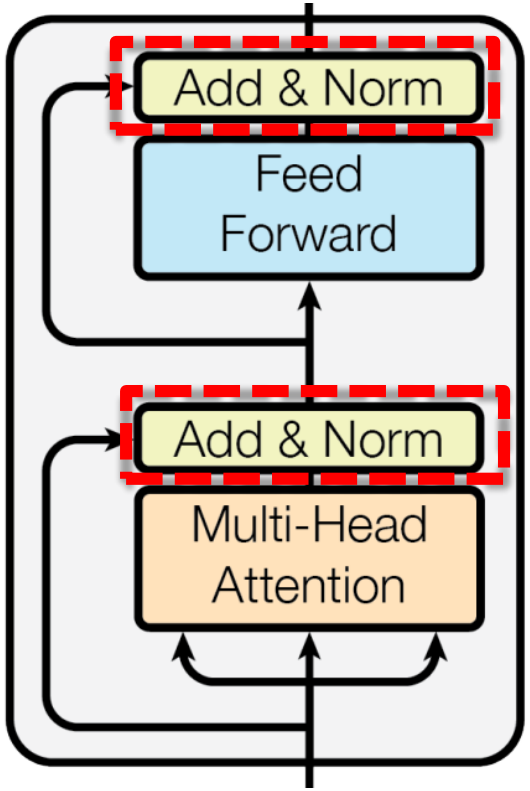
In this work we employ $h = 8$ parallel attention layers, or heads. For each of these we use $d_k = d_v = d_{\text{model}}/h = 64$. Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality.

编码器中的ADD&Norm

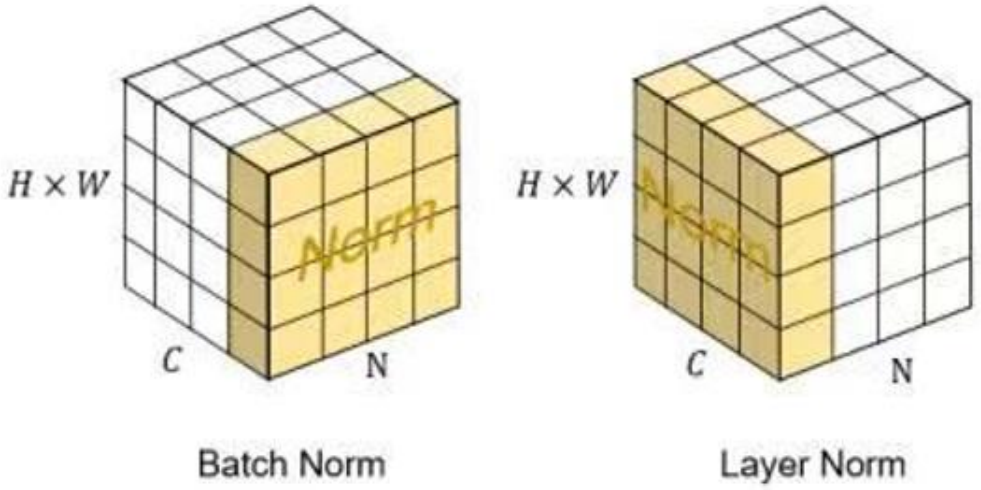




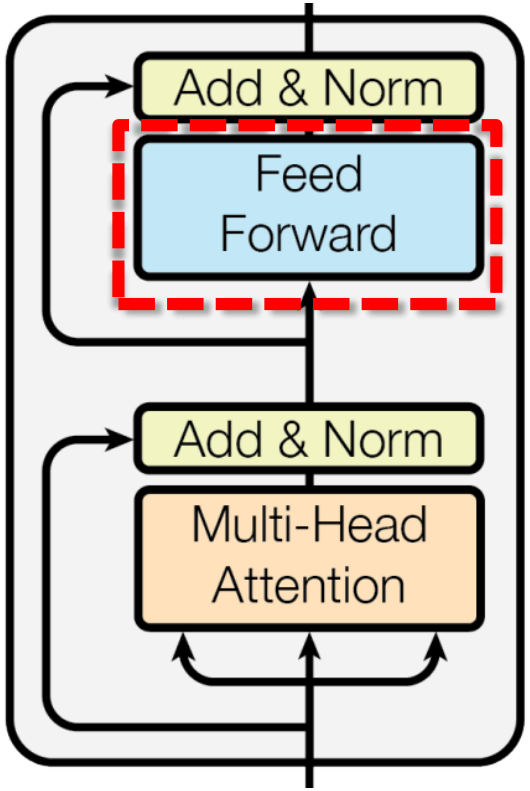
编码器中的ADD&Norm



$$\mathbf{LayerNorm}(\mathbf{x} + \mathbf{Sublayer}(\mathbf{x}))$$



编码器中的前馈网络



将维度缩小4倍

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

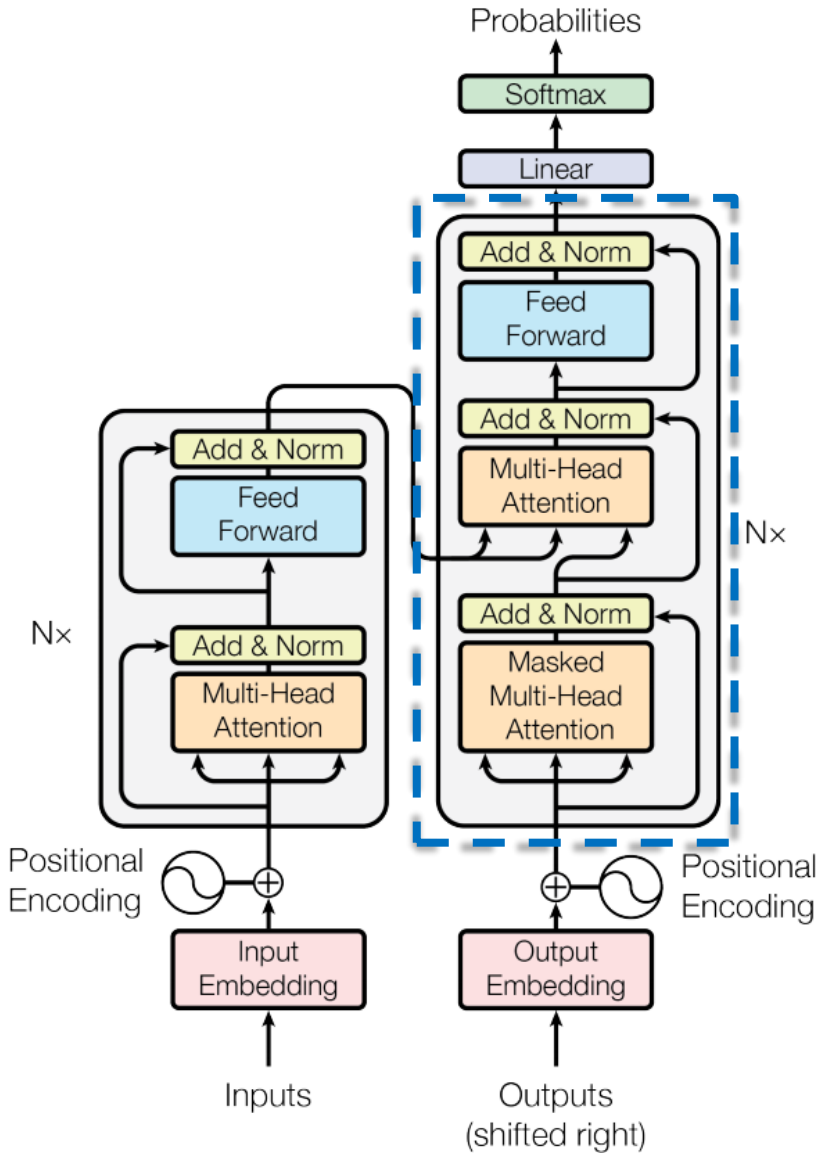
将维度扩大4倍

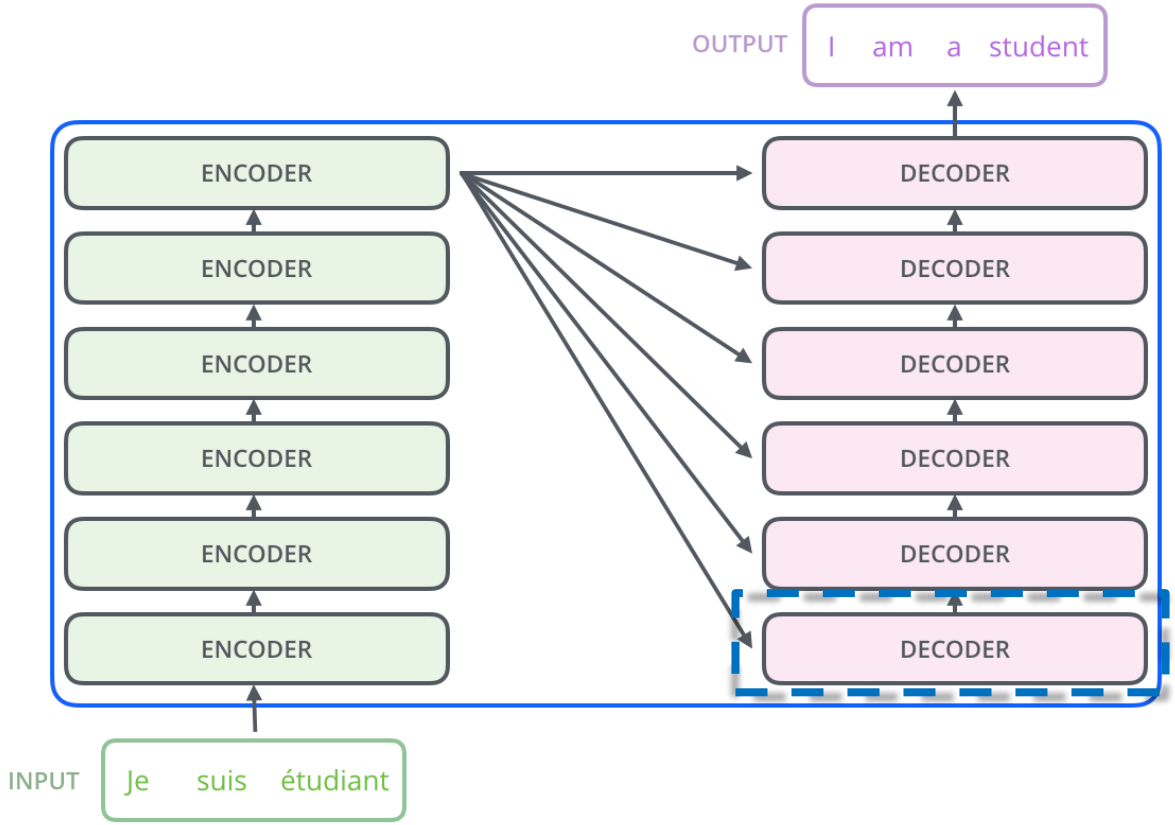
FFN为二层网络

$d_{ff} = 2048$

$d_{model} = 512$

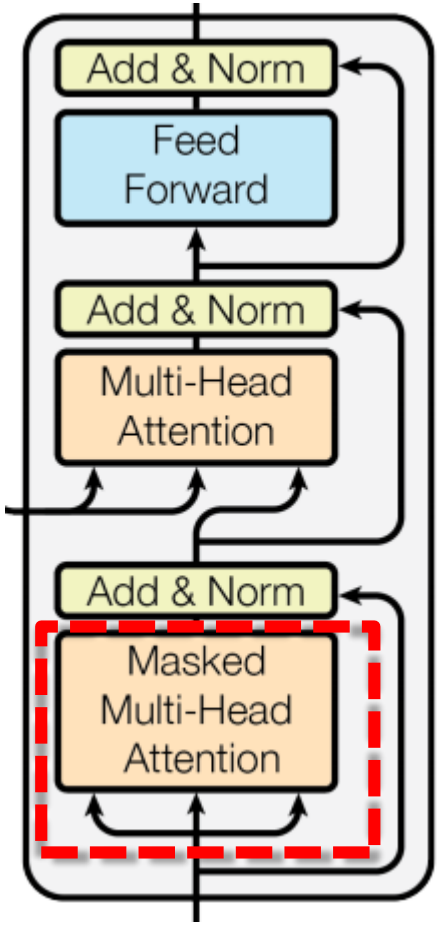
Transformer中的解码器

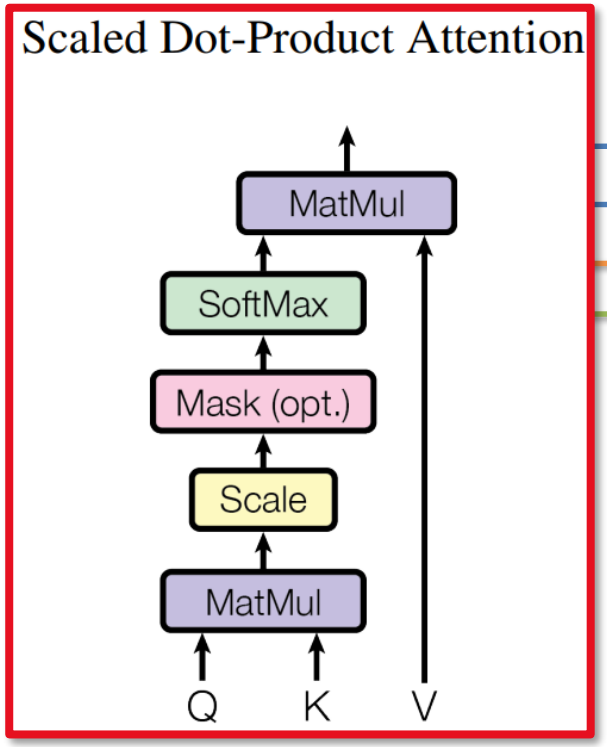


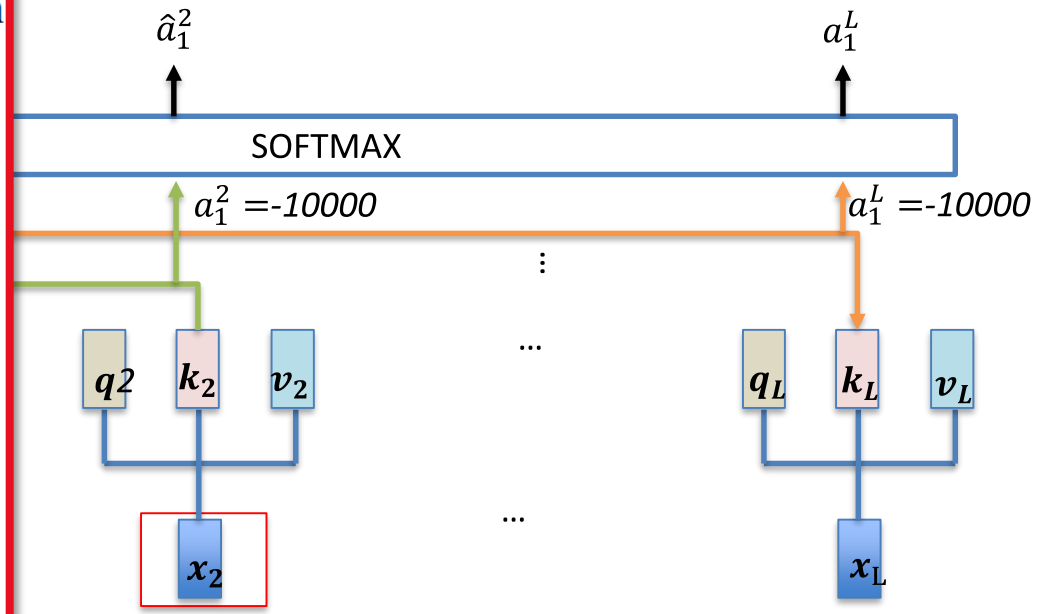


原始论文：堆叠了6层编码器、6层解码器

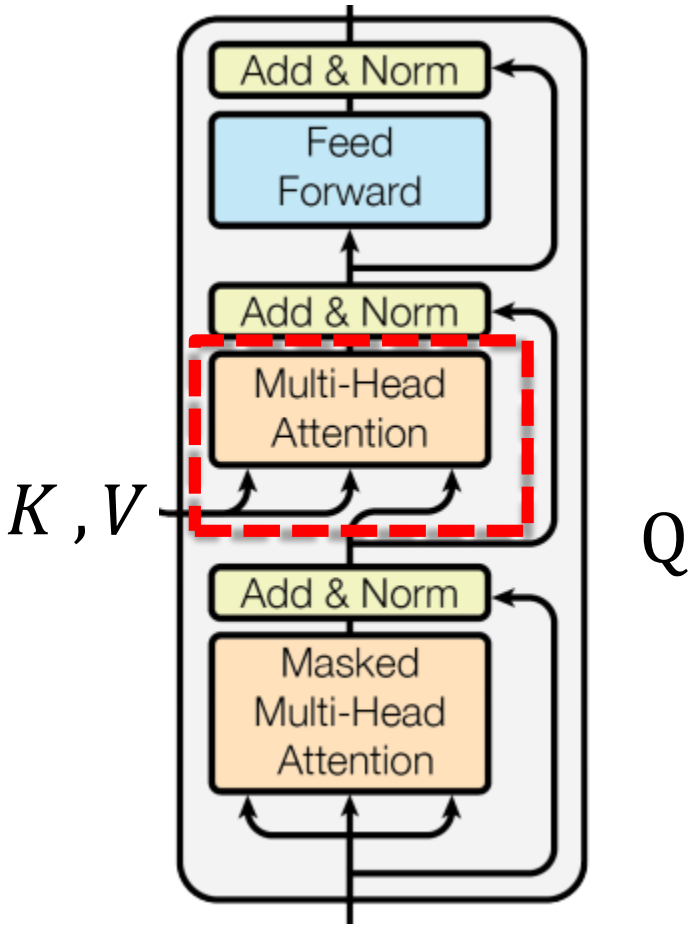
Transformer的解码器

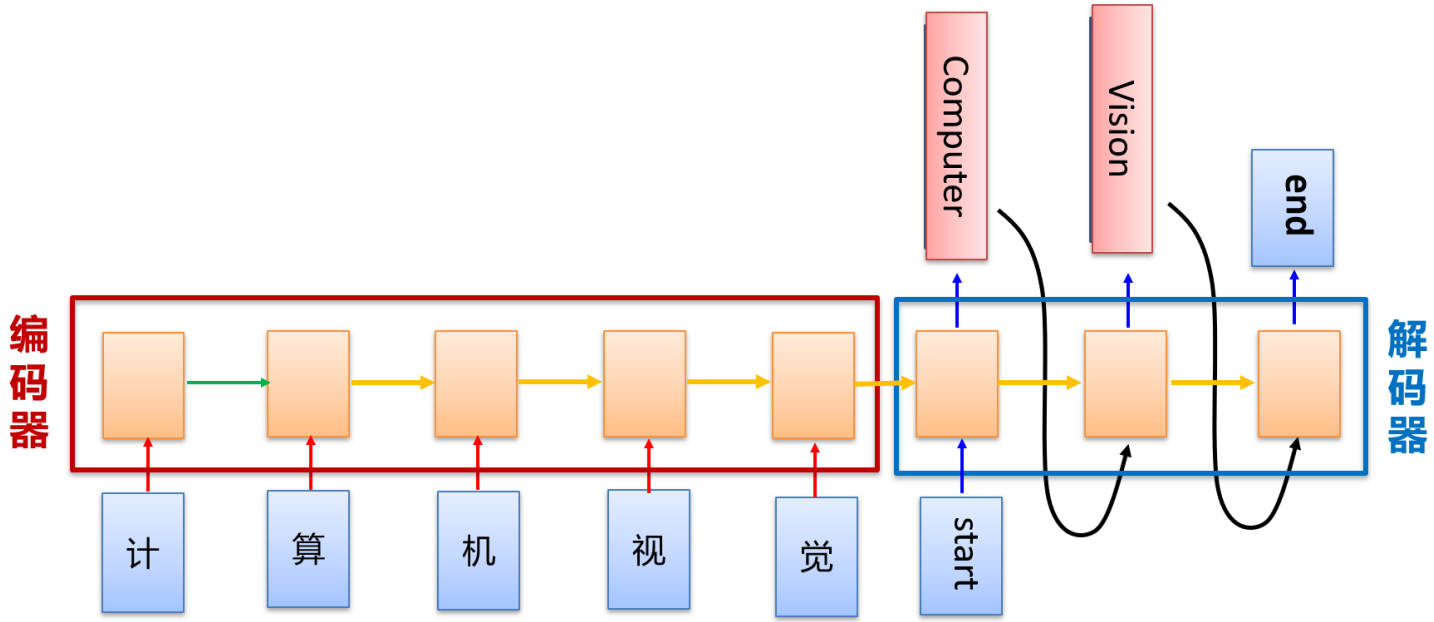




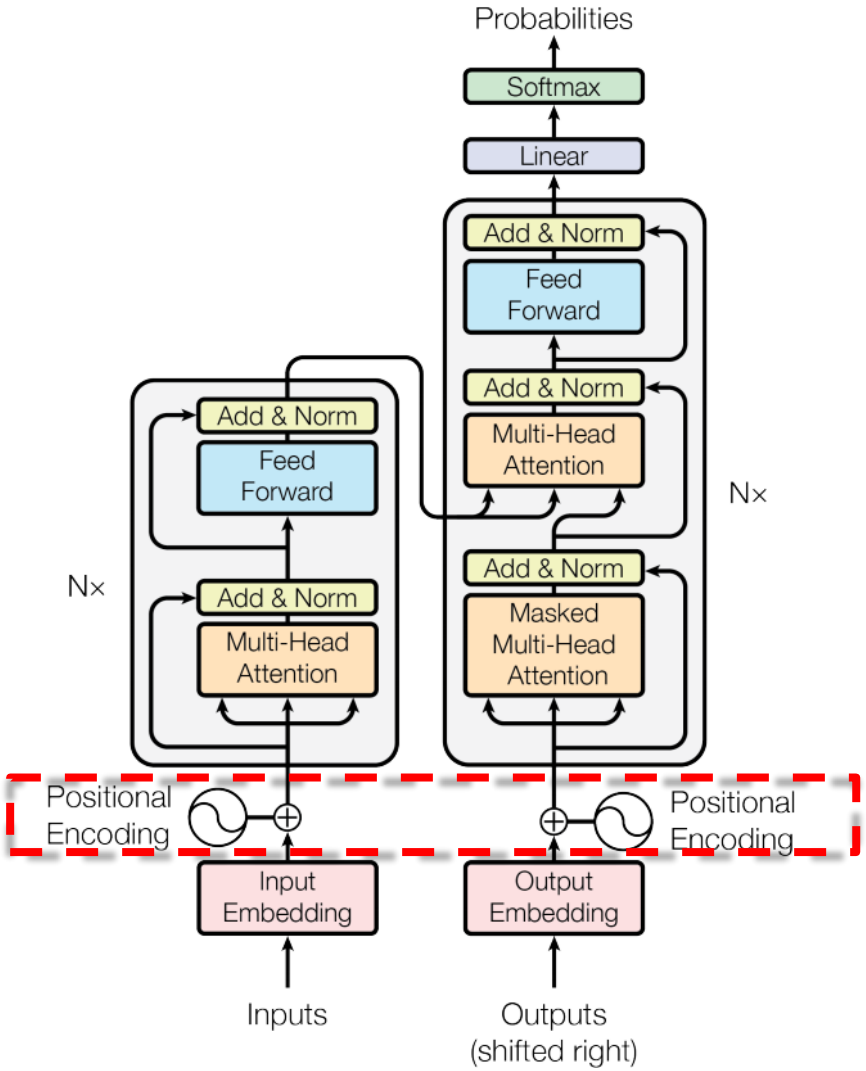


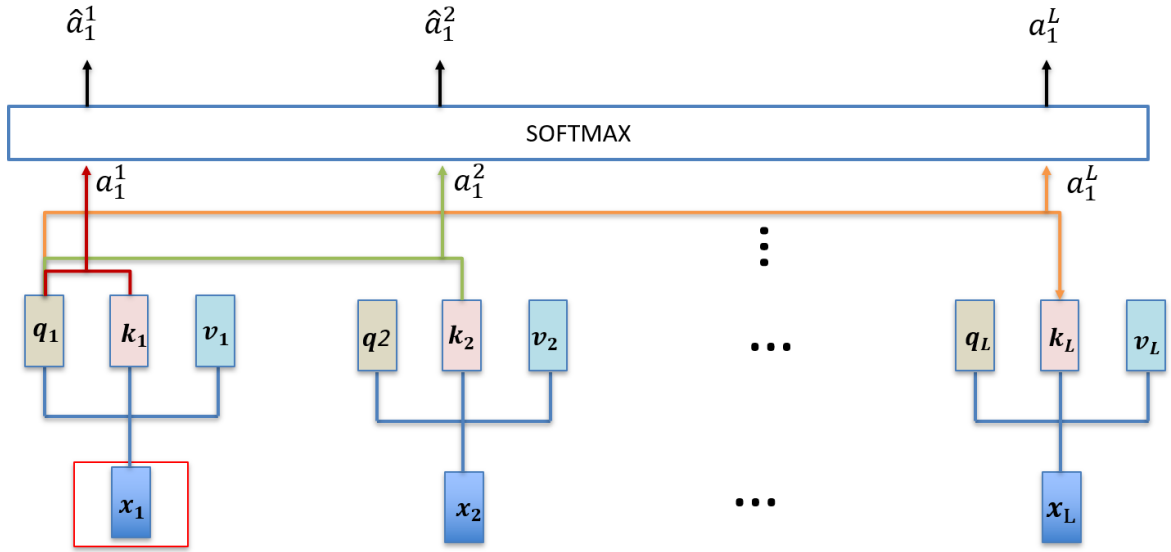
Transformer的解码器



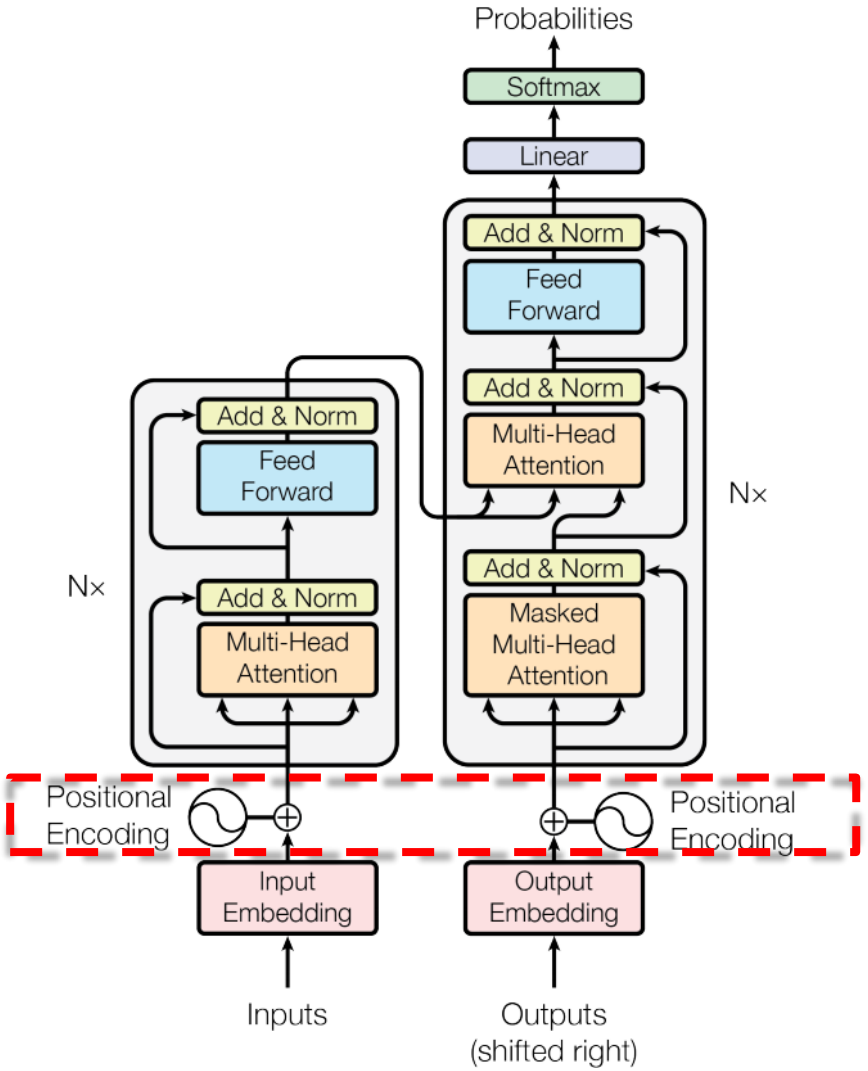


Transformer的位置编码





Transformer的位置编码



$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

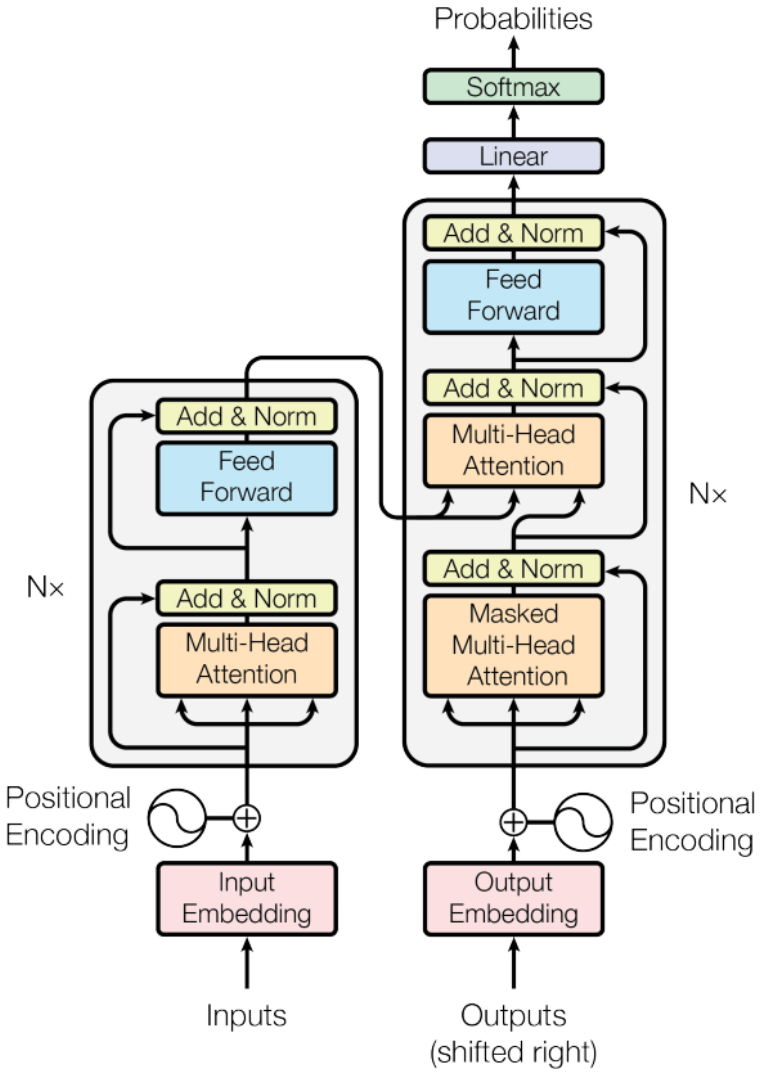
选择上述编码方式的原因之一：

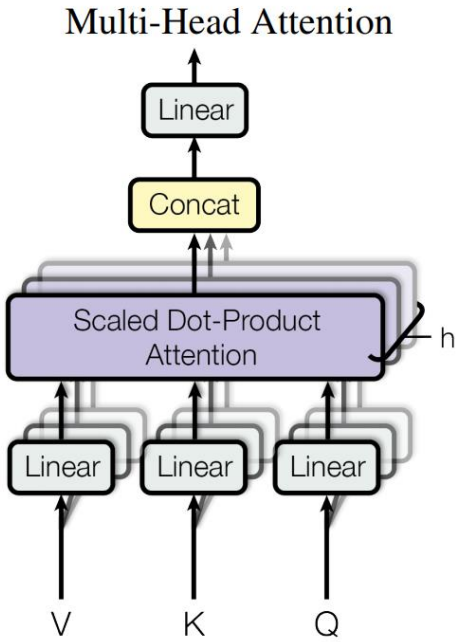
$$\sin(\alpha+\beta)=\sin\alpha\cdot\cos\beta+\cos\alpha\cdot\sin\beta$$

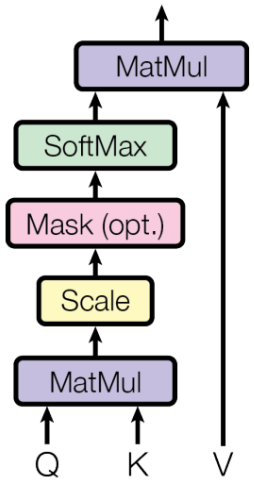
$$\cos(\alpha+\beta)=\cos\alpha\cdot\cos\beta-\sin\alpha\cdot\sin\beta$$

位置α+β的向量可以表示成位置α和位置β的向量组合，这提供了表达相对位置信息的可能性。

Transformer（完）







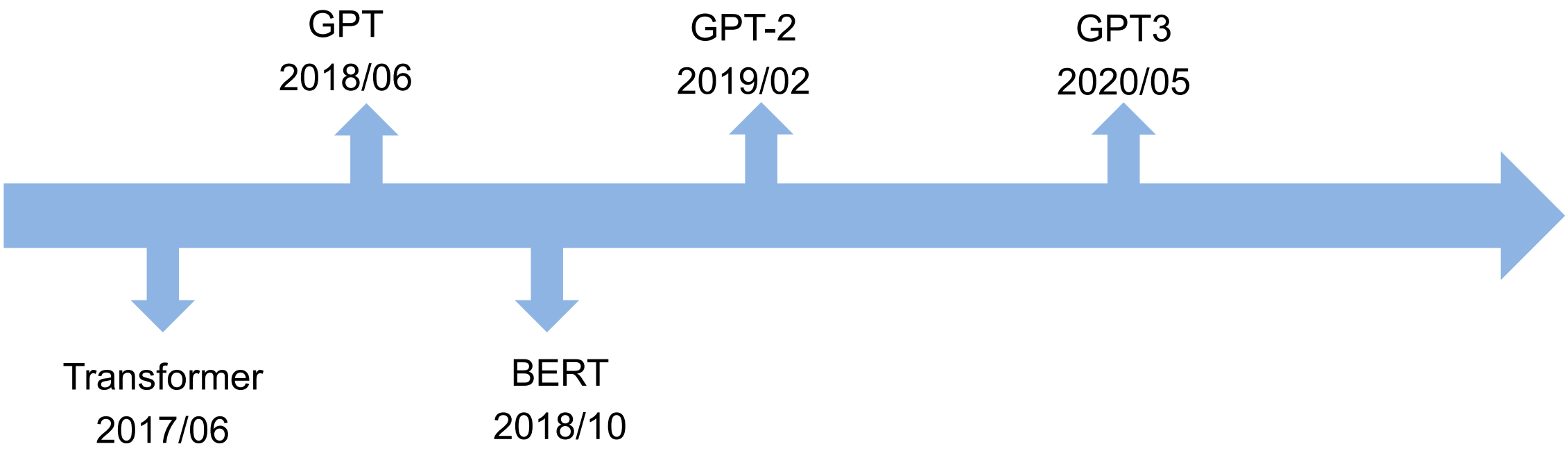
- 编解码器层数： N
- 嵌入维度： d_{model}
- 多头数: h
- 前馈网络第一层宽度： d_{ff}

自监督学习

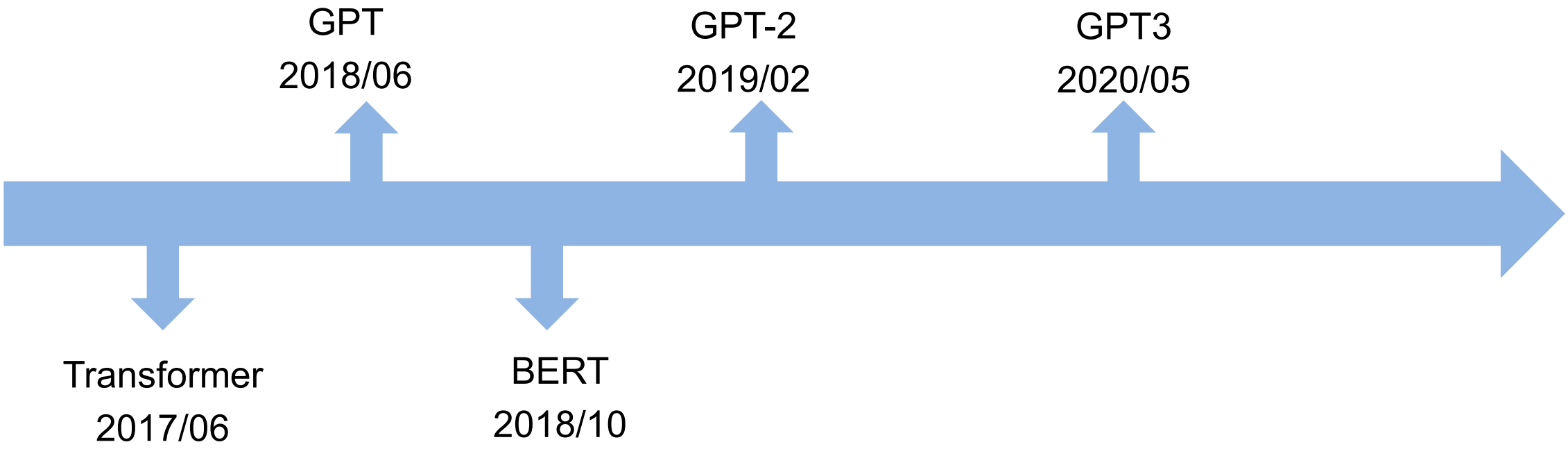
- **定义**：使用无标注数据用自我监督的方式学习特征表示的方法。其通过构造一个代理任务（pretext task）来实现特征表示学习。
- 代理任务可以是一个预测类任务、生成式任务、对比学习任务。代理任务的监督信息来源是从数据本身获得的。
- **举例**：完型填空（BERT）、预测下一个单词（GPT）

典型用法：通过自监督学习完成特征提取器的预训练，然后，在下游任务上进行微调。

Transformer在NLP中的发展



Transformer在NLP中的发展



视觉领域

Non-Local
2018/04

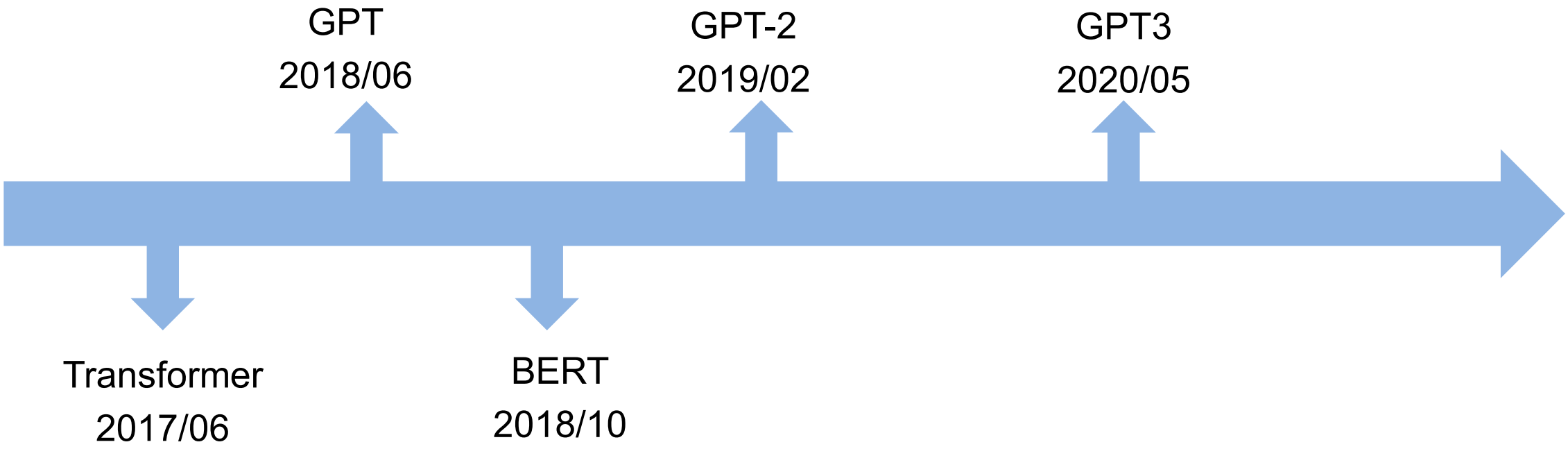
ViT
2021/06

MAE
2021/12

今日主题

- Transformer
- Non-Local 模块
- ViT
- MAE

Transformer在NLP中的发展



视觉领域

Non-Local
2018/04

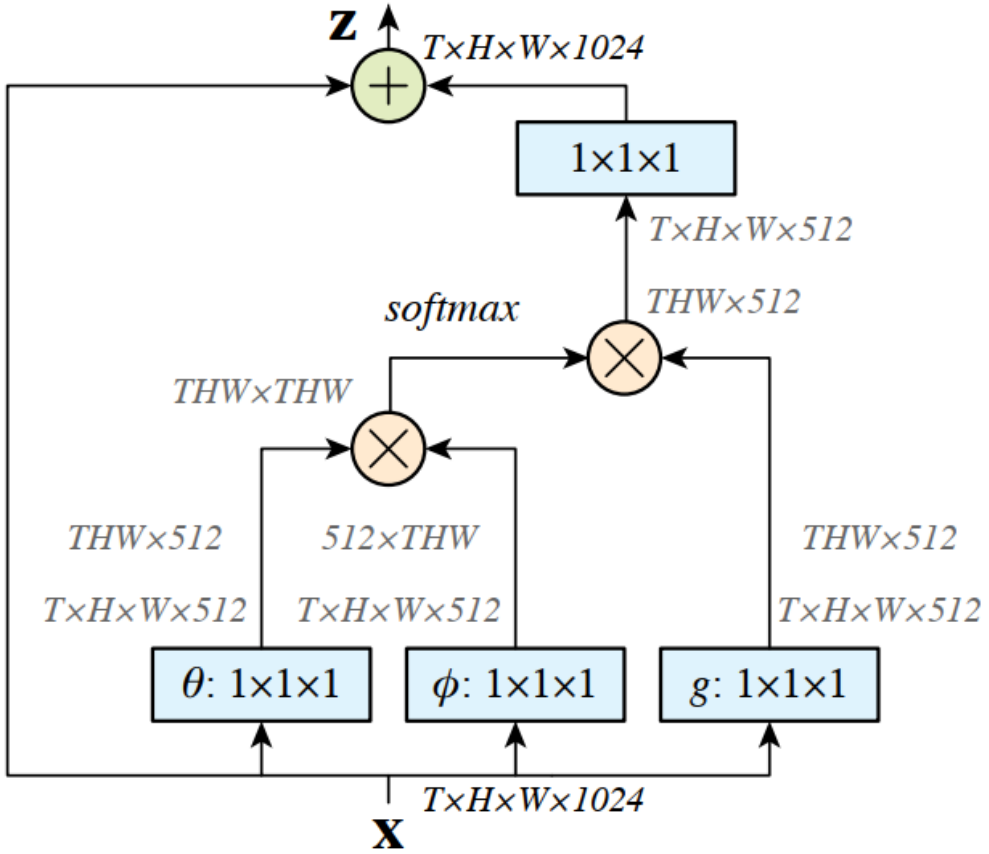
ViT
2021/06

MAE
2021/12

Non-local Neural Networks

Xiaolong Wang[1,2*] Ross Girshick[2] Abhinav Gupta[1] Kaiming He[2]

[1]Carnegie Mellon University [2]Facebook AI Research



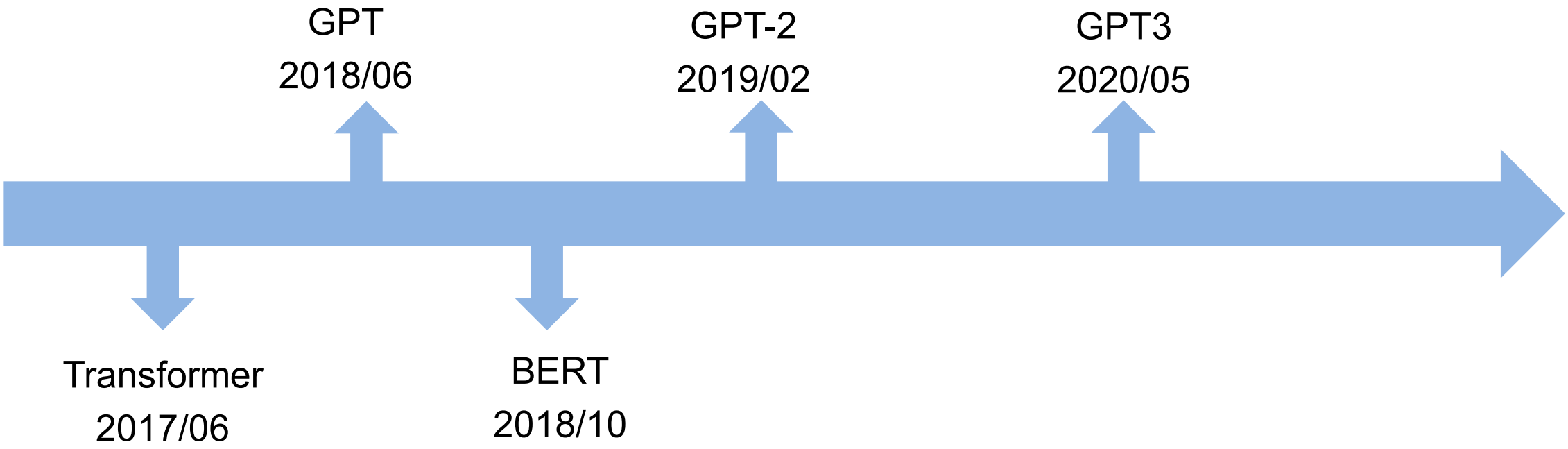
问题：卷积操作难以捕捉长距离依赖。

破解之法：卷积网路+Non-local模块（注意力机制）

今日主题

- Transformer
- Non-Local 模块
- ViT
- MAE

Transformer在NLP中的发展



视觉领域

Non-Local
2018/04

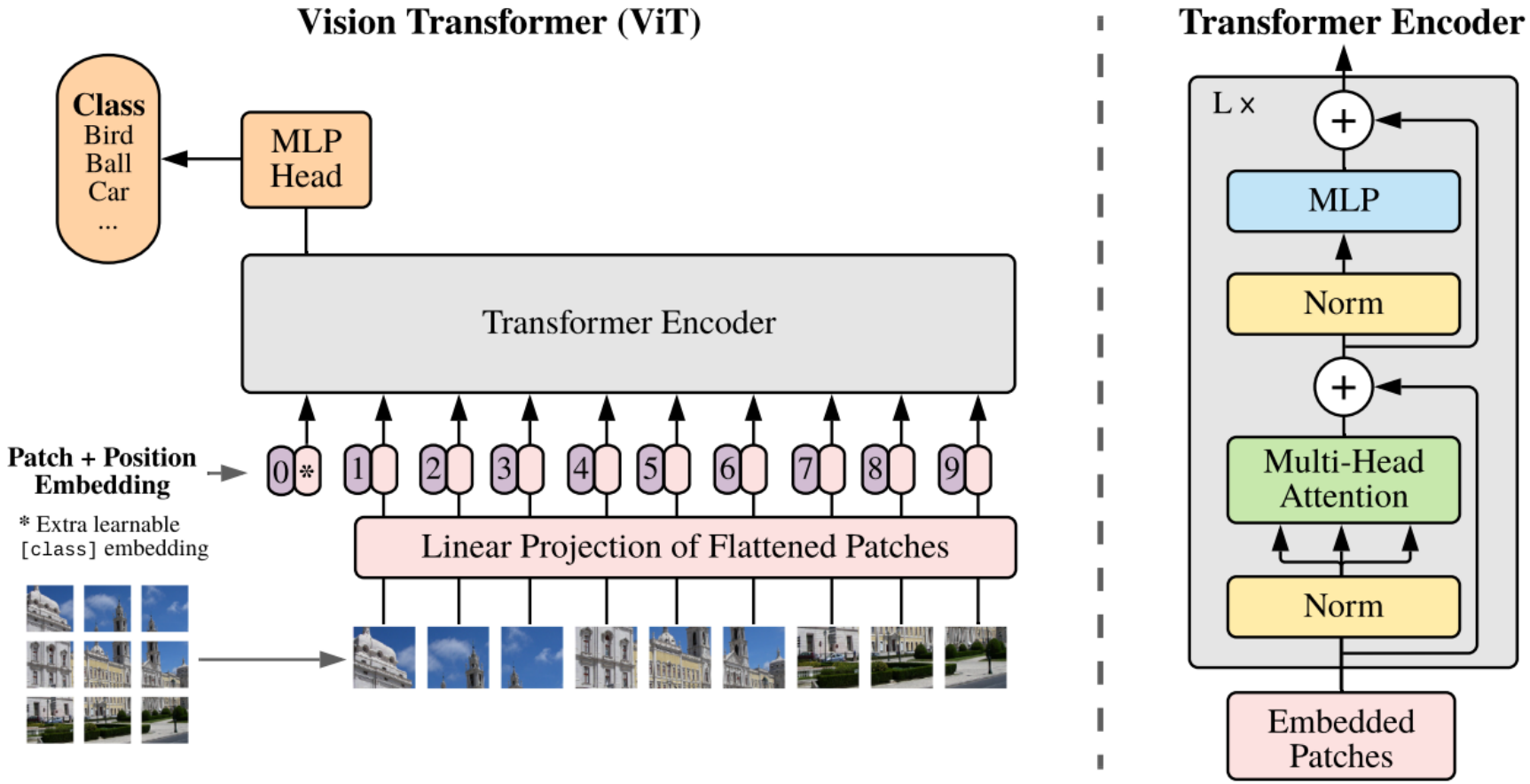
ViT
2021/06

MAE
2021/12

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

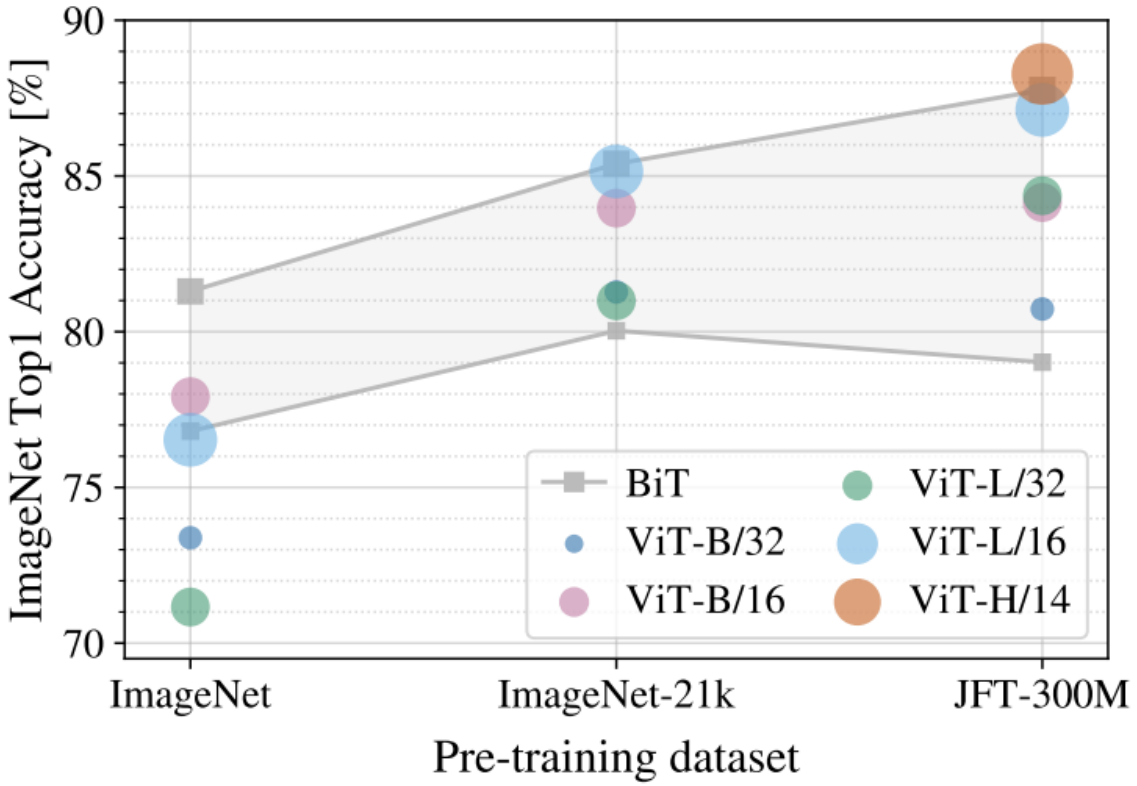
- CNN的统治地位
- Transformer在自然语言处理领域如此成功
- Transformer在图像领域也可以获得与CNN相当甚至更高的精度
- 缺少局部性与平移性的归纳偏置需要大量的数据进行训练

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE



Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

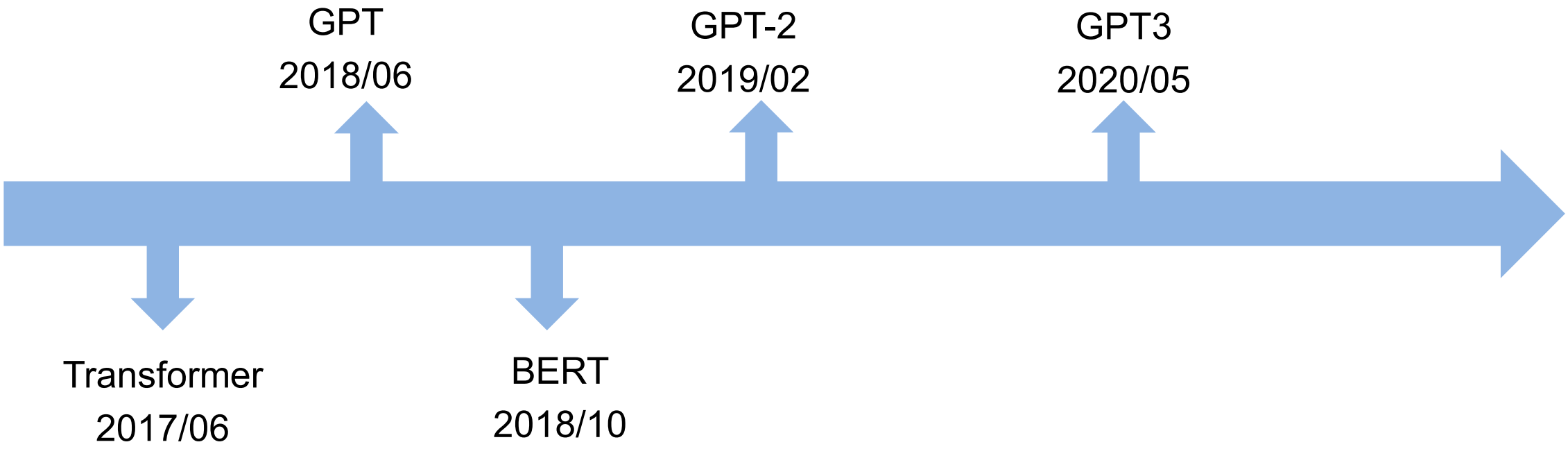


- ImageNet 1K个类别 1.3M张图像
- ImageNet-21K 21K个类别 14M张图像
- JFT 18k个类别 303M张高分图像

今日主题

- Transformer
- Non-Local 模块
- ViT
- MAE

Transformer在NLP中的发展



视觉领域

Non-Local
2018/04

ViT
2021/06

MAE
2021/12

Masked Autoencoders Are Scalable Vision Learners

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick

*equal technical contribution †project lead

Facebook AI Research (FAIR)

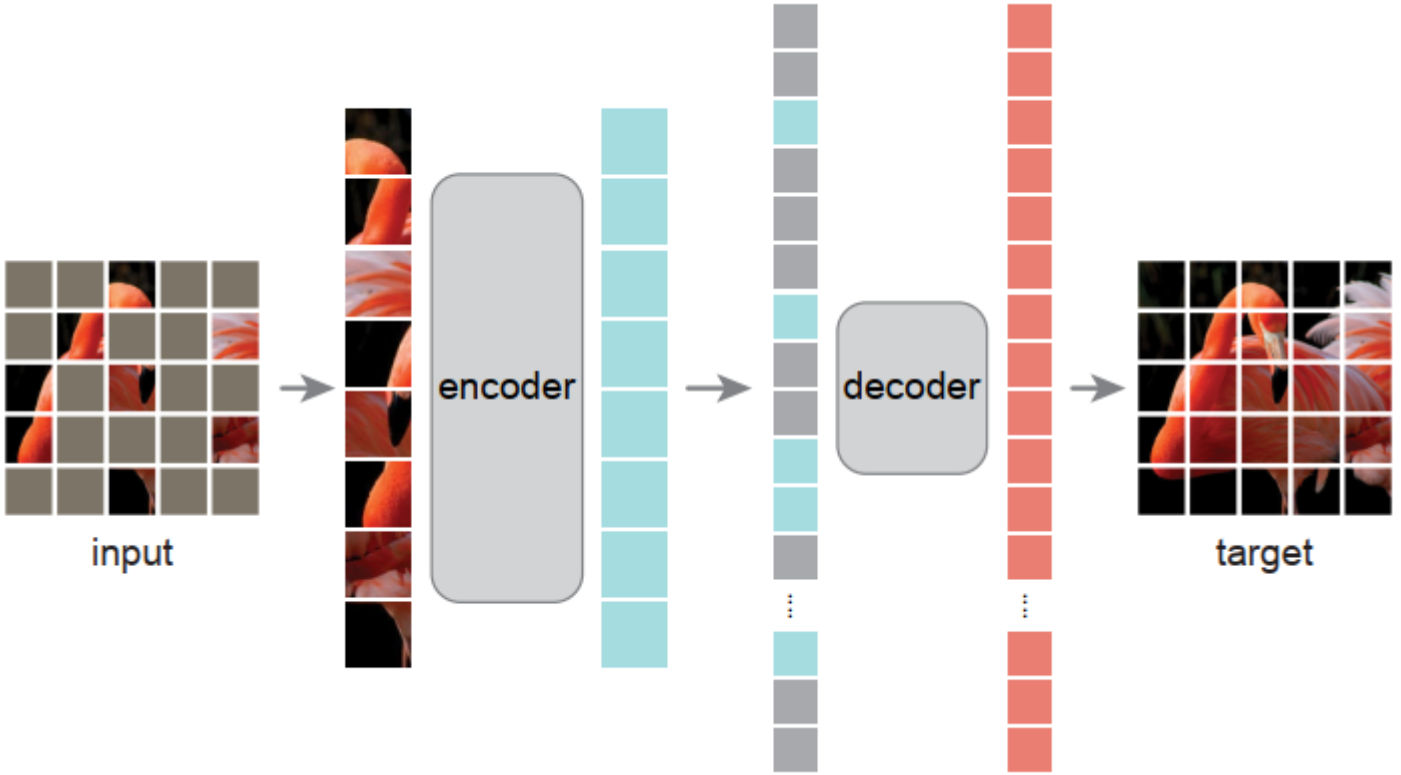
- 提出一个非对称的编解码器网络
- 自监督学习，采用高比率的遮挡预测任务作为代理任务
- Imagenet-1K上训练达到了87.8%，且在检测、分割等任务上均达到了SOTA
- 训练过程快

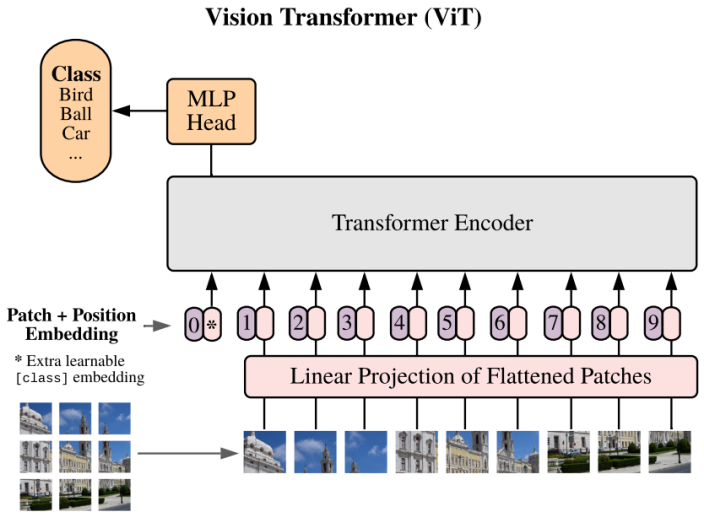
Masked Autoencoders Are Scalable Vision Learners

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick

*equal technical contribution †project lead

Facebook AI Research (FAIR)





Masked Autoencoders Are Scalable Vision Learners

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick

*equal technical contribution †project lead

Facebook AI Research (FAIR)

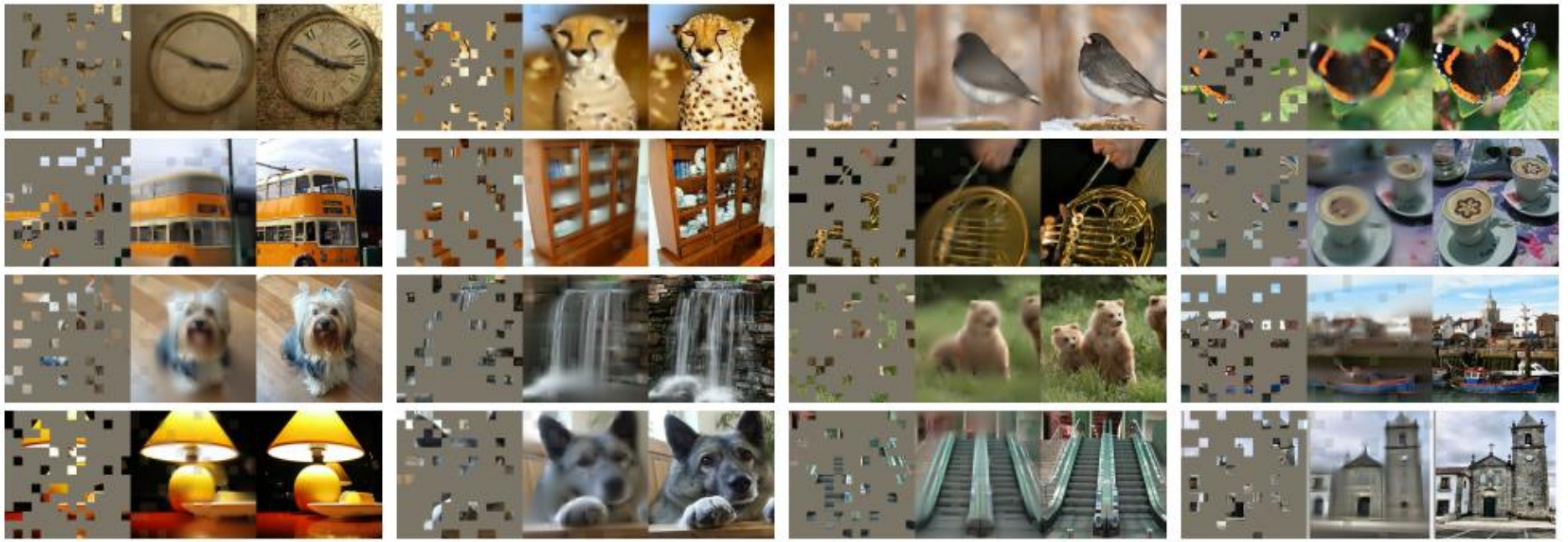


Figure 2. Example results on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction† (middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches. More examples are in the appendix.
†*As no loss is computed on visible patches, the model output on visible patches is qualitatively worse. One can simply overlay the output with the visible patches to improve visual quality. We intentionally opt not to do this, so we can more comprehensively demonstrate the method's behavior.*

Masked Autoencoders Are Scalable Vision Learners

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick

*equal technical contribution †project lead

Facebook AI Research (FAIR)

Figure 4. Reconstructions of ImageNet *validation* images using an MAE pre-trained with a masking ratio of 75% but applied on inputs with higher masking ratios. The predictions differ plausibly from the original images, showing that the method can generalize.

Masked Autoencoders Are Scalable Vision Learners

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick

*equal technical contribution †project lead

Facebook AI Research (FAIR)

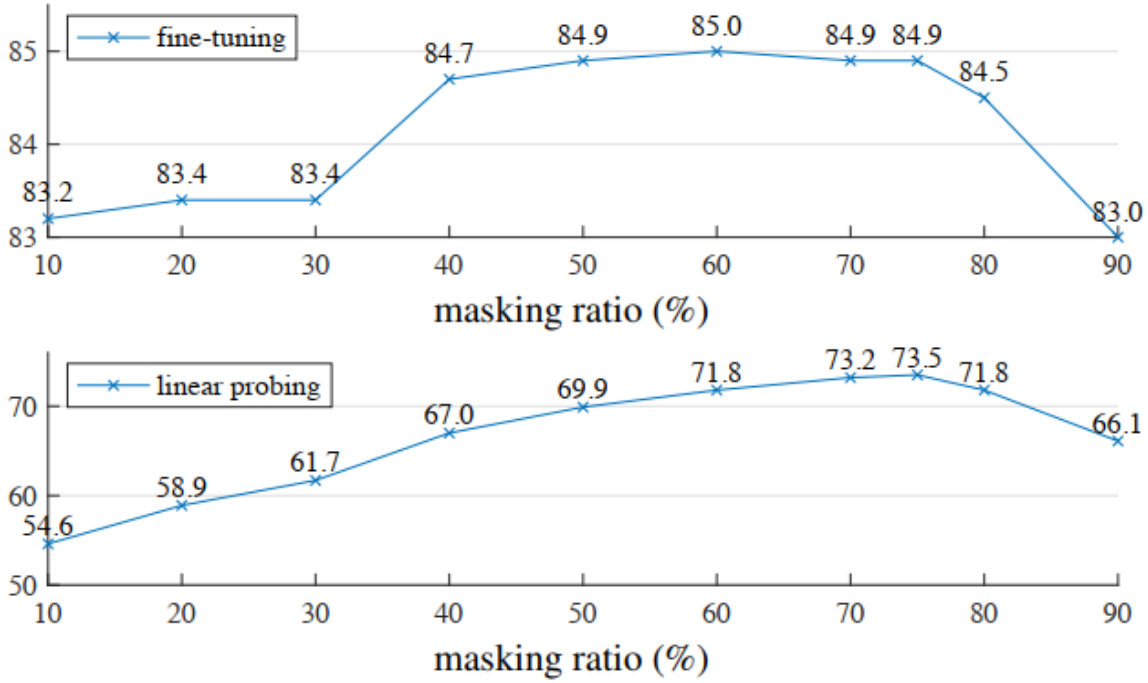


Figure 5. **Masking ratio**. A high masking ratio (75%) works well for both fine-tuning (top) and linear probing (bottom). The y-axes are ImageNet-1K validation accuracy (%) in all plots in this paper.

Masked Autoencoders Are Scalable Vision Learners

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick

*equal technical contribution †project lead

Facebook AI Research (FAIR)

blocks	ft	lin
1	84.8	65.5
2	**84.9**	70.0
4	**84.9**	71.9
8	**84.9**	**73.5**
12	84.4	73.3

(a) **Decoder depth**. A deep decoder can improve linear probing accuracy.

dim	ft	lin
128	**84.9**	69.1
256	84.8	71.3
512	**84.9**	**73.5**
768	84.4	73.1
1024	84.3	73.1

(b) **Decoder width**. The decoder can be narrower than the encoder (1024-d).

case	ft	lin	FLOPs
encoder w/ [M]	84.2	59.6	3.3×
encoder w/o [M]	**84.9**	**73.5**	**1**×

(c) **Mask token**. An encoder without mask tokens is more accurate and faster (Table 2).

case	ft	lin
pixel (w/o norm)	84.9	73.5
pixel (w/ norm)	**85.4**	**73.9**
PCA	84.6	72.3
dVAE token	85.3	71.6

(d) **Reconstruction target**. Pixels as reconstruction targets are effective.

case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9

(e) **Data augmentation**. Our MAE works with minimal or no augmentation.

case	ratio	ft	lin
random	75	**84.9**	**73.5**
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

(f) **Mask sampling**. Random sampling works the best. See Figure 6 for visualizations.

Masked Autoencoders Are Scalable Vision Learners

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick

*equal technical contribution †project lead

Facebook AI Research (FAIR)

method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H$_{448}$
scratch, our impl.	-	82.3	82.6	83.1	-
DINO [5]	IN1K	82.8	-	-	-
MoCo v3 [9]	IN1K	83.2	84.1	-	-
BEiT [2]	IN1K+DALLE	83.2	85.2	-	-
MAE	IN1K	83.6	85.9	86.9	**87.8**

Table 3. **Comparisons with previous results on ImageNet-1K**. The pre-training data is the ImageNet-1K training set (except the tokenizer in BEiT was pre-trained on 250M DALLE data [50]). All self-supervised methods are evaluated by end-to-end fine-tuning. The ViT models are B/16, L/16, H/14 [16]. The best for each column is underlined. All results are on an image size of 224, except for ViT-H with an extra result on 448. Here our MAE reconstructs normalized pixels and is pre-trained for 1600 epochs.

今日主题

- Transformer(完)
- Non-Local 模块(完)
- ViT (完)
- MAE (完)

本学期课程到此全部结束，谢谢大家的参与！！！