Transformer in CV

1. VIT
2. Swin Transformer
3. MAE

舒言

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

Alexey Dosovitskiy[*,†], **Lucas Beyer**[*], **Alexander Kolesnikov**[*], **Dirk Weissenborn**[*], **Xiaohua Zhai**[*], **Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby**[*,†]

[*]equal technical contribution, [†]equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

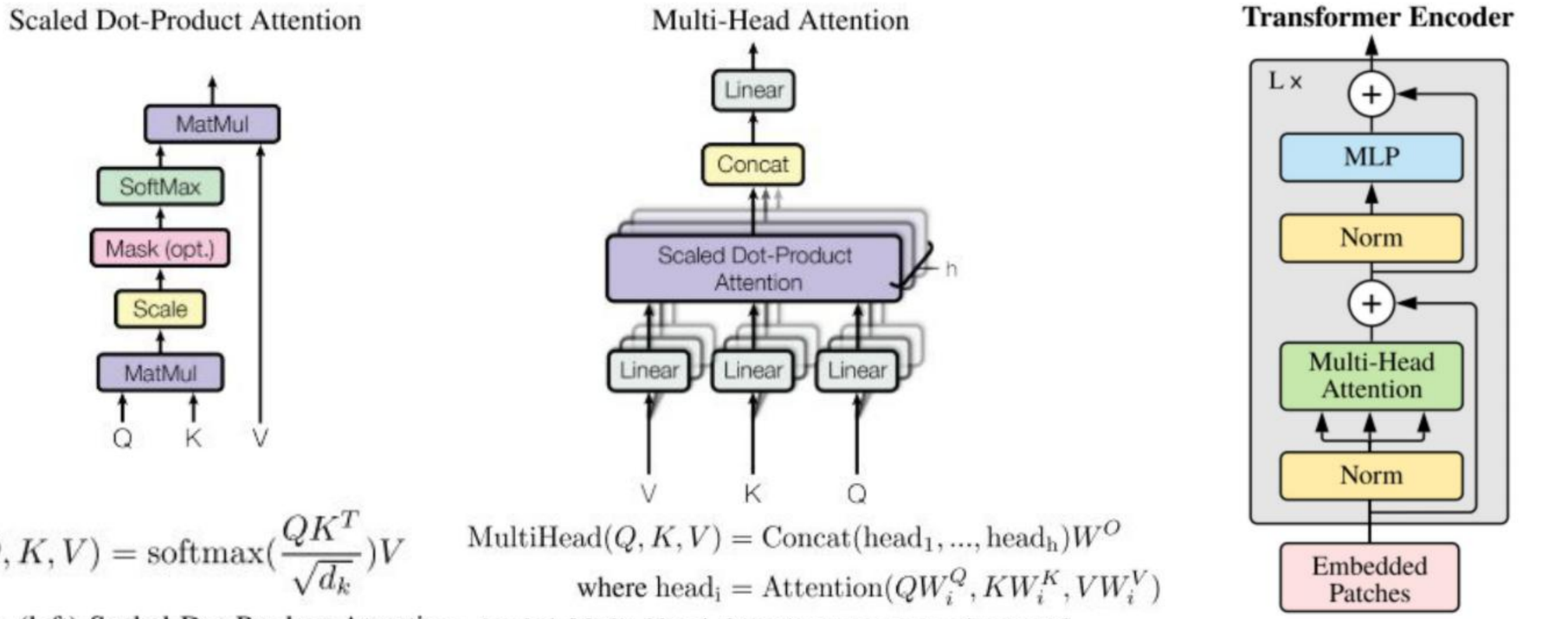
While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks 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.[1]

Overview

1. Intro - Transformer Encoder
2. Vision Transformer
3. Results
4. Conclusions

Transformer Encoder

Vaswani, Ashish, et al. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).



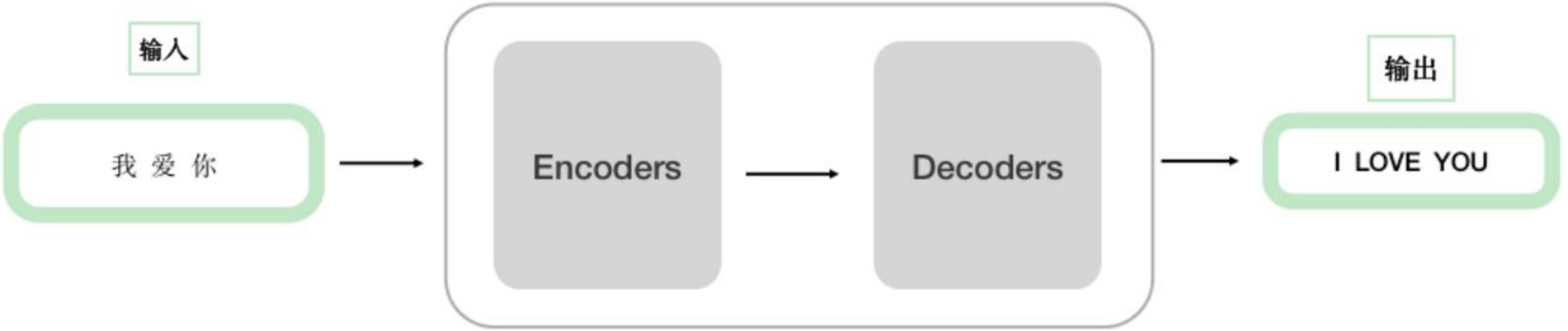
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

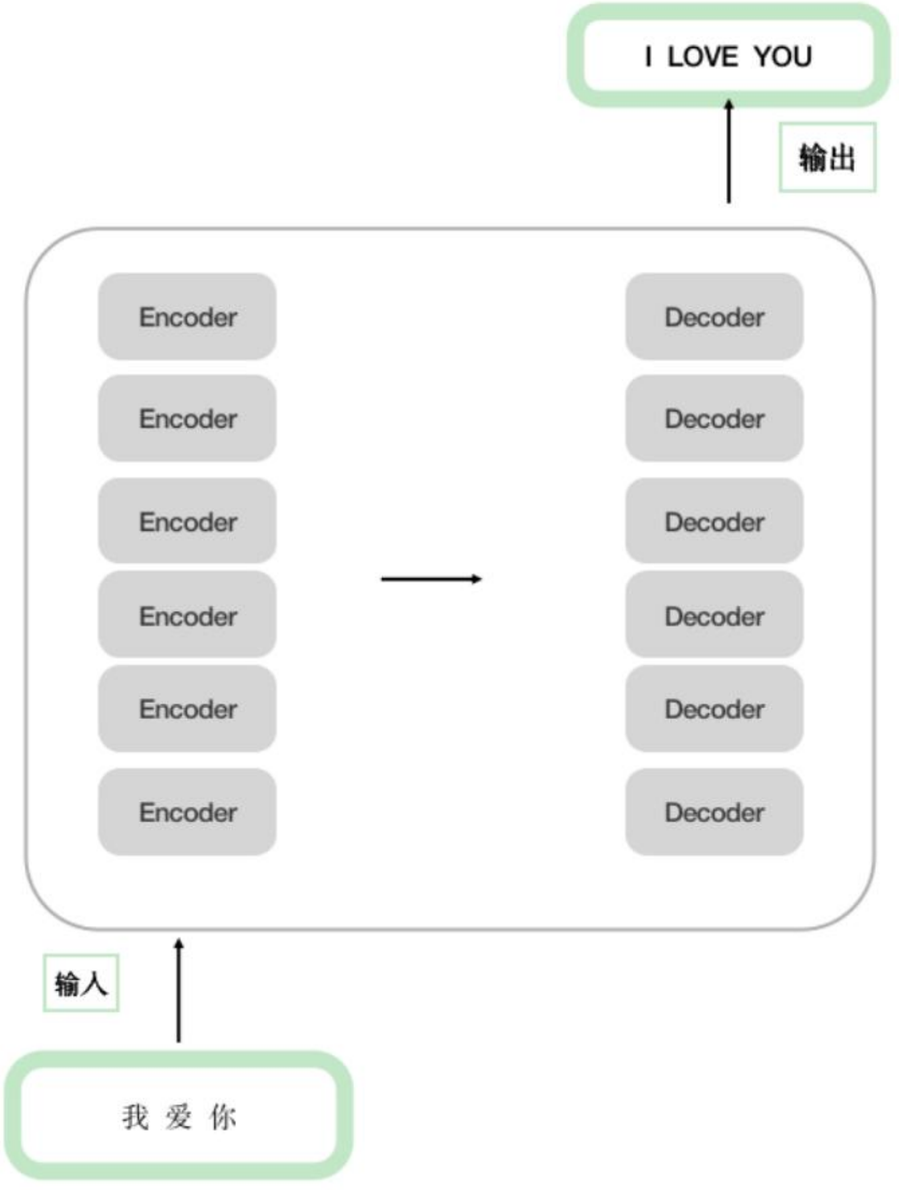
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$

$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

TRM在做一个什么事情？


输入
我 爱 你
Encoders
Decoders
输出
I LOVE YOU


I LOVE YOU
输出
Encoder
Encoder
Encoder
Encoder
Encoder
Encoder
Decoder
Decoder
Decoder
Decoder
Decoder
Decoder
输入
我 爱 你

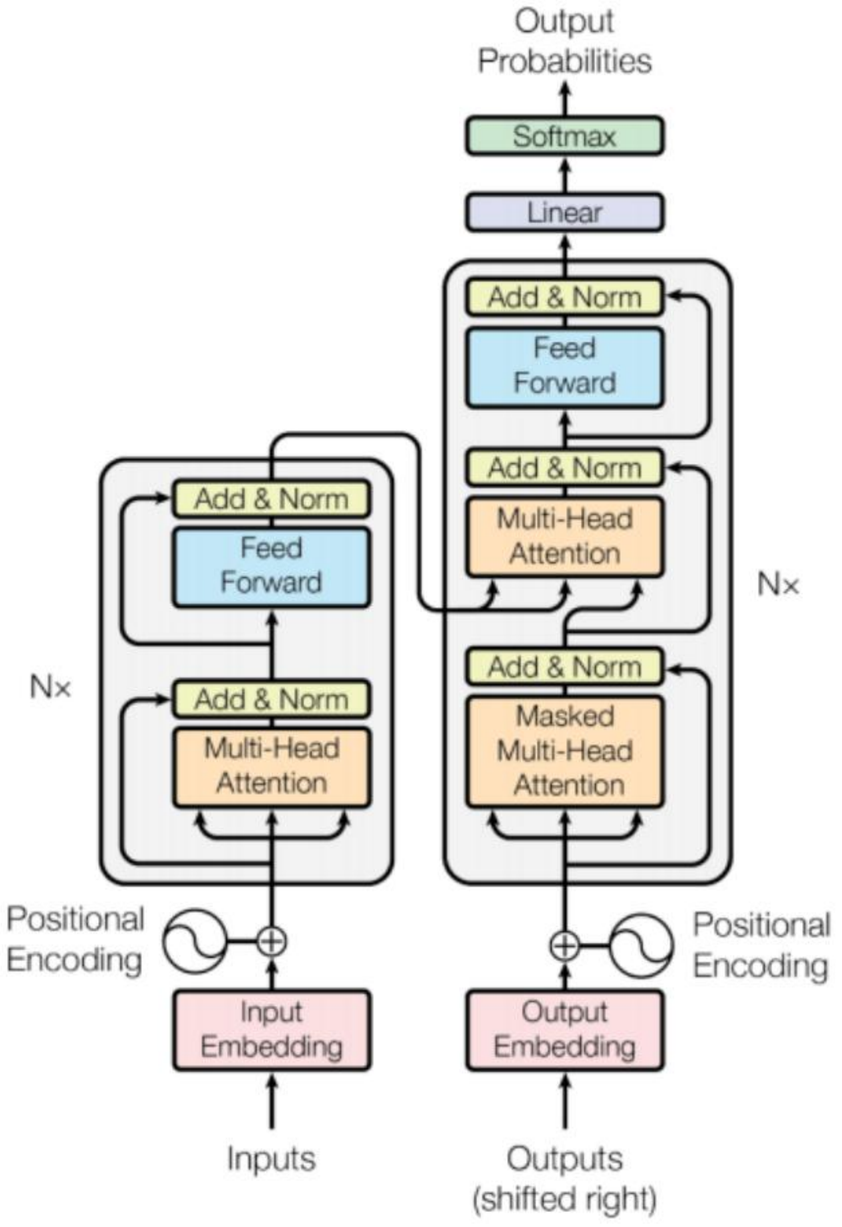
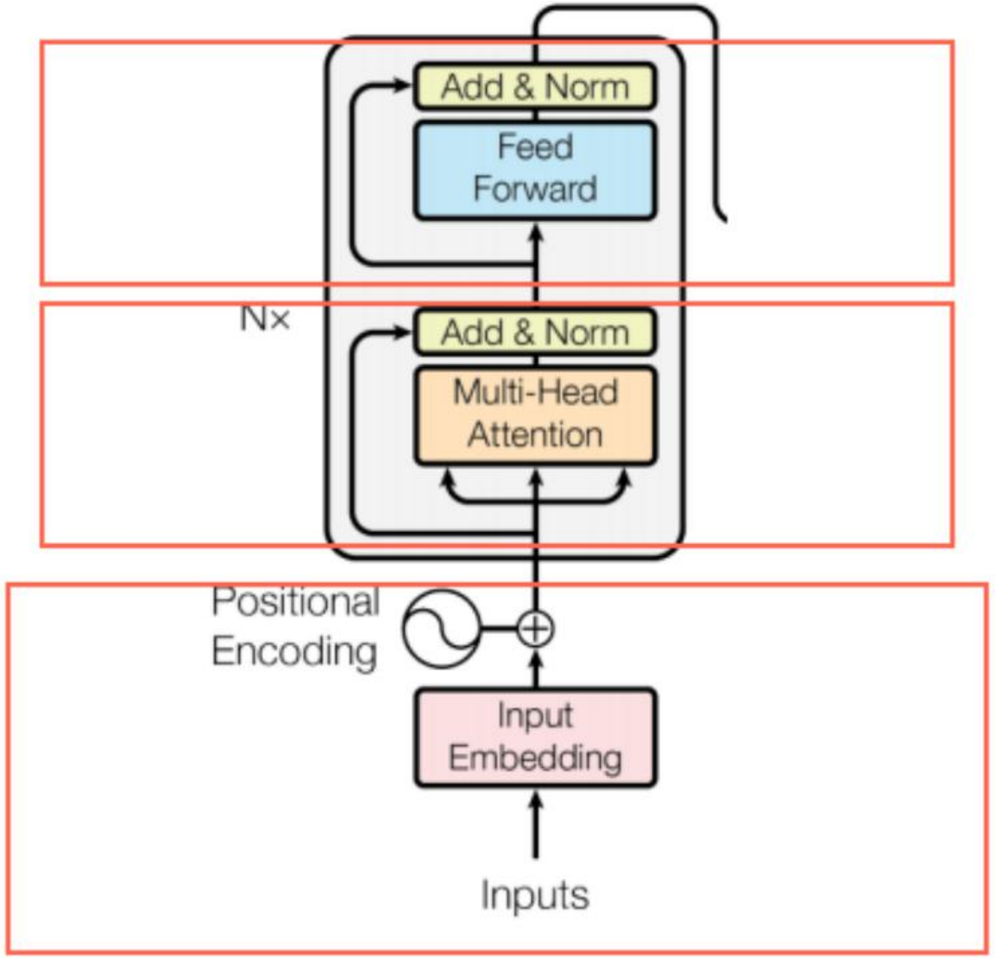


Figure 1: The Transformer - model architecture.



3 前馈神经网络

2 注意力机制

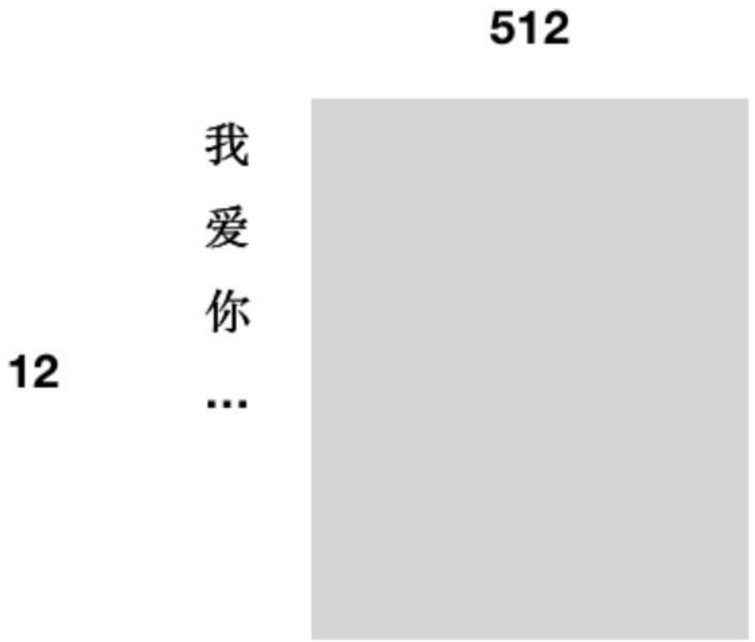
1 输入部分

输入部分

1. Embedding

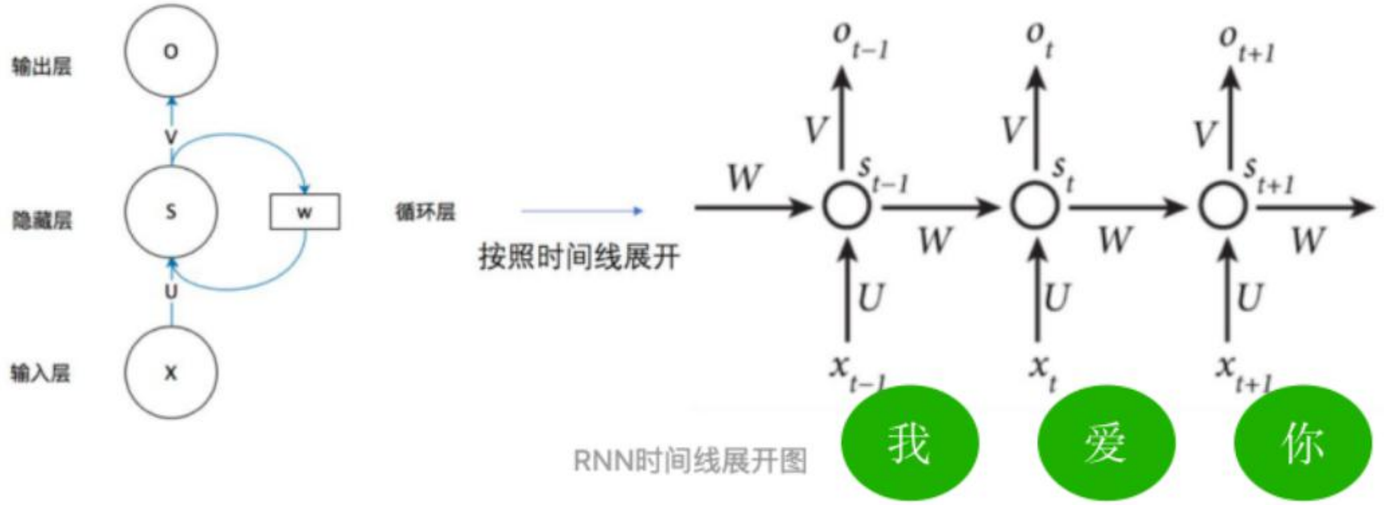
2. 位置嵌入

Embedding



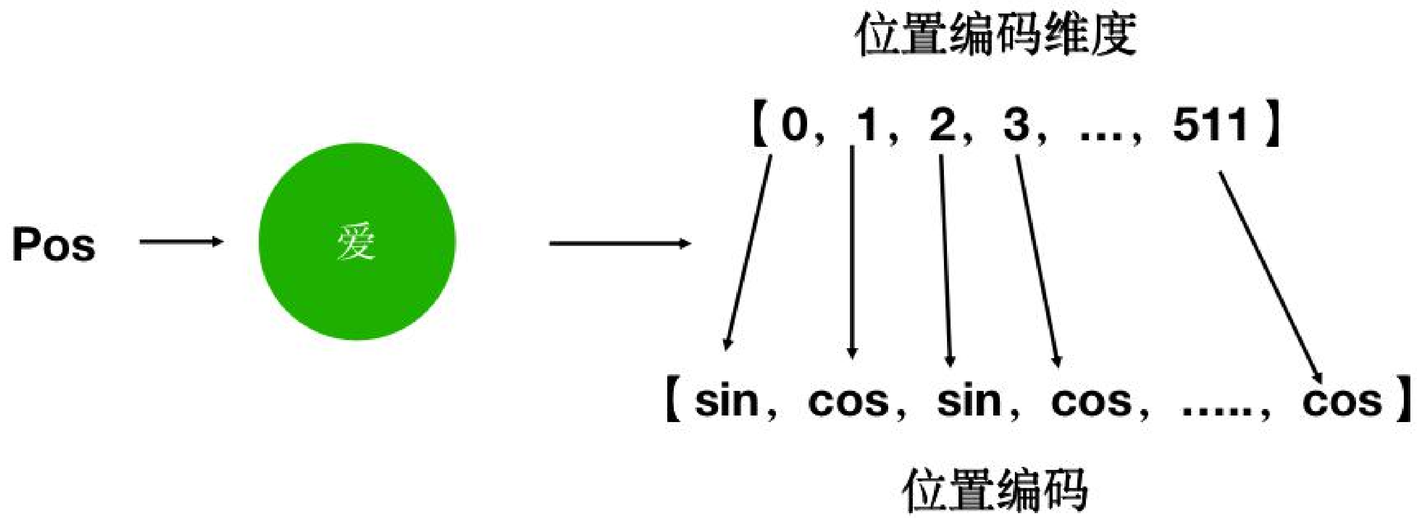
位置编码

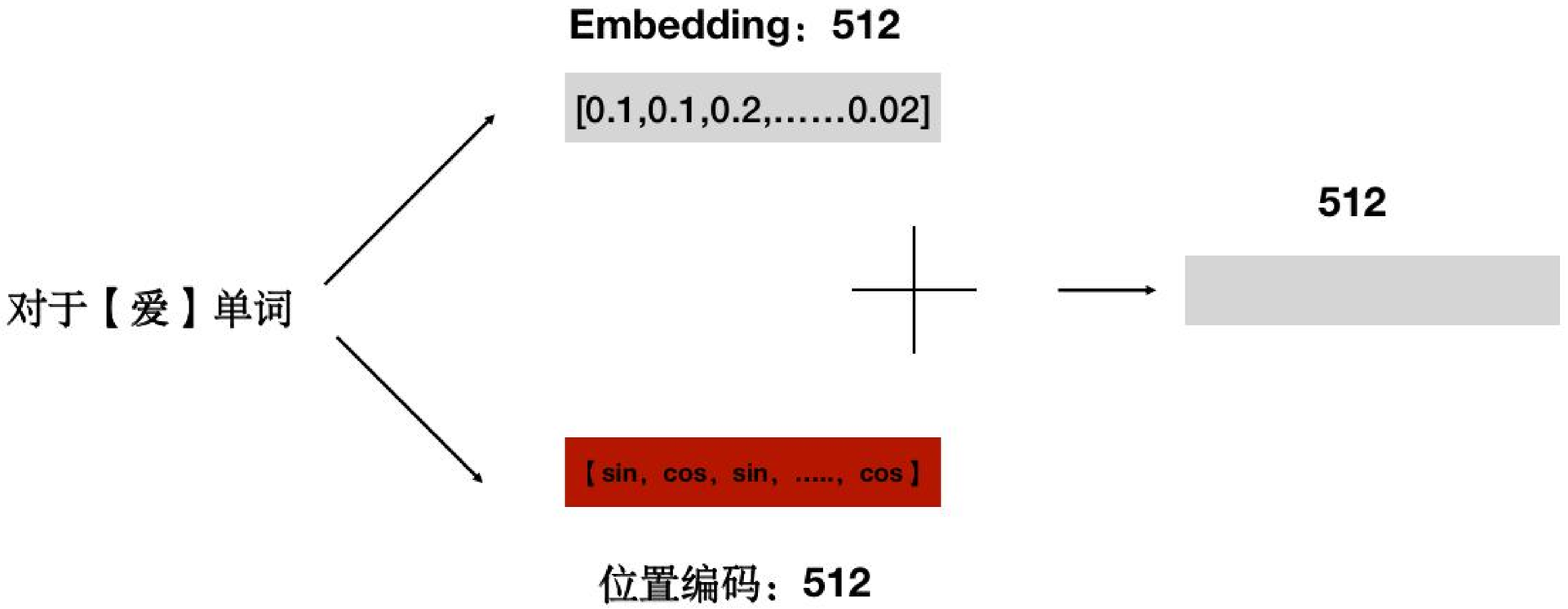
为什么需要：



位置编码公式

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

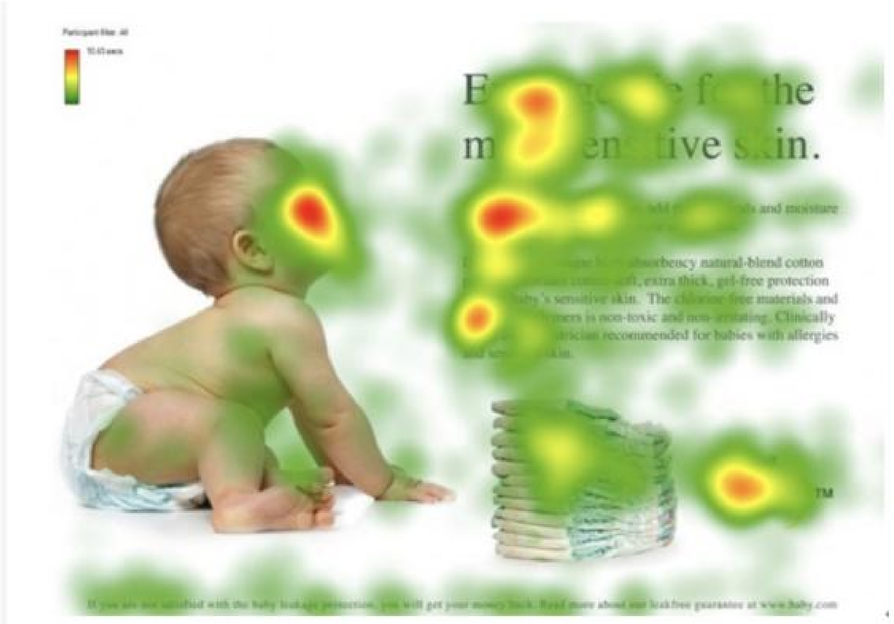



Embedding：512
[0.1,0.1,0.2,……0.02]
512
对于【爱】单词
【sin，cos，sin，.....，cos】
位置编码：512

注意力机制

1. 基本的注意力机制
2. 在TRM中怎么操作

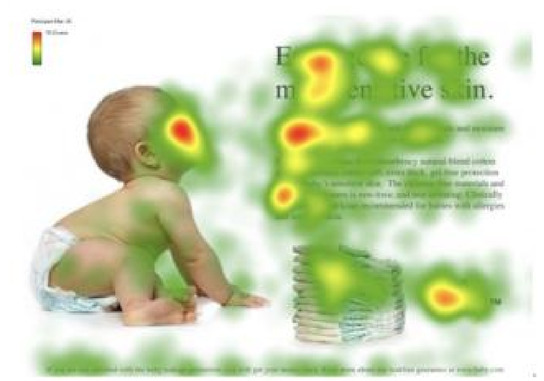
注意力机制本质

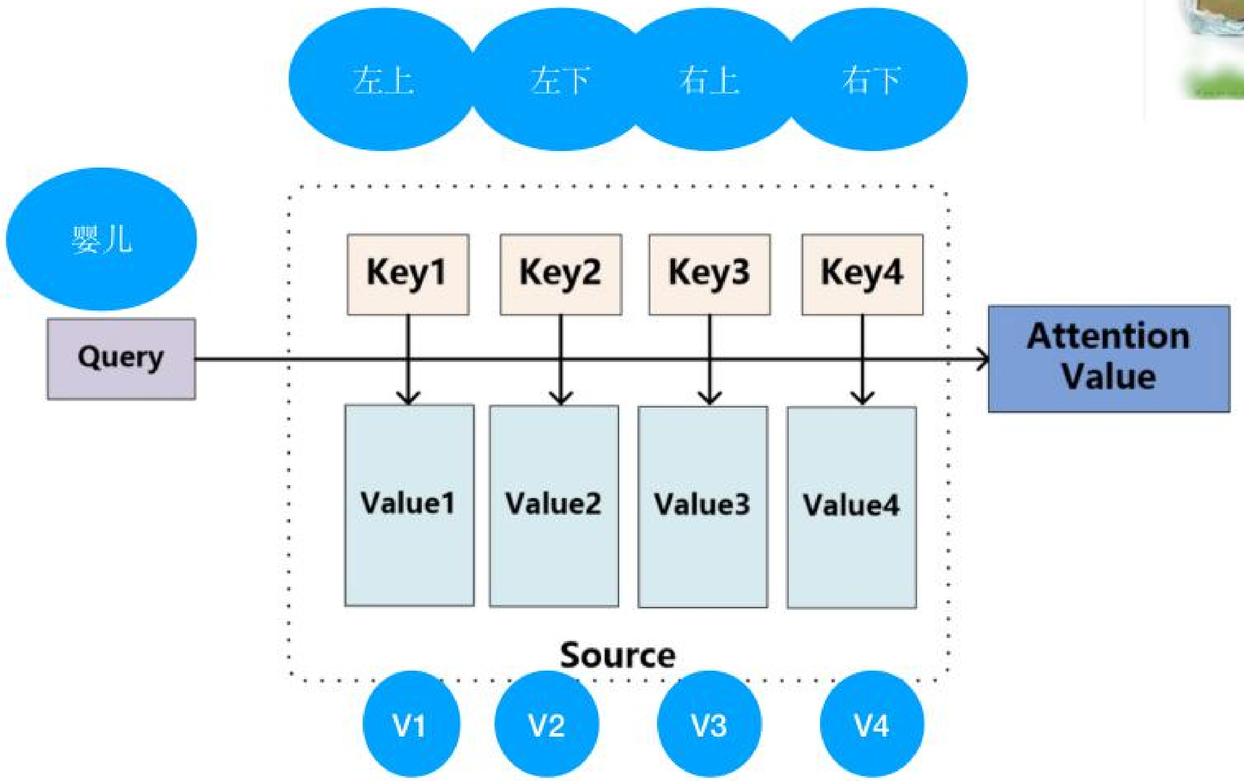


婴儿在干嘛

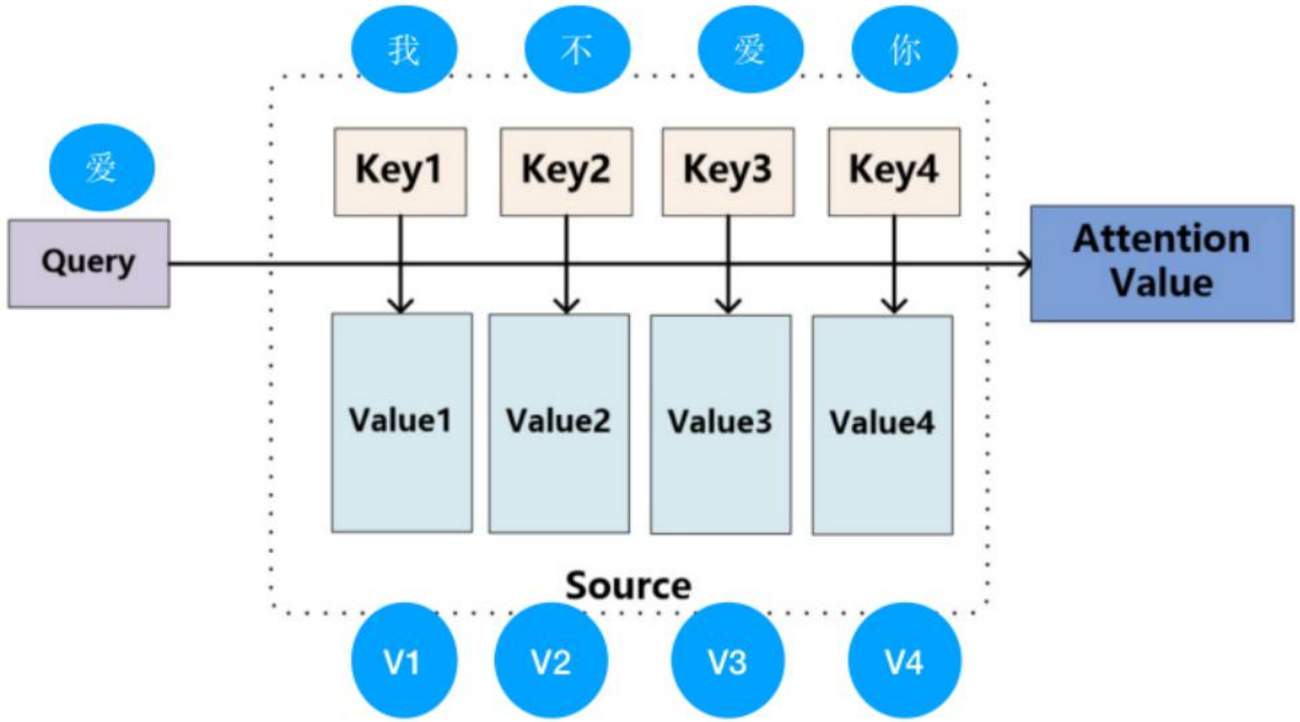
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

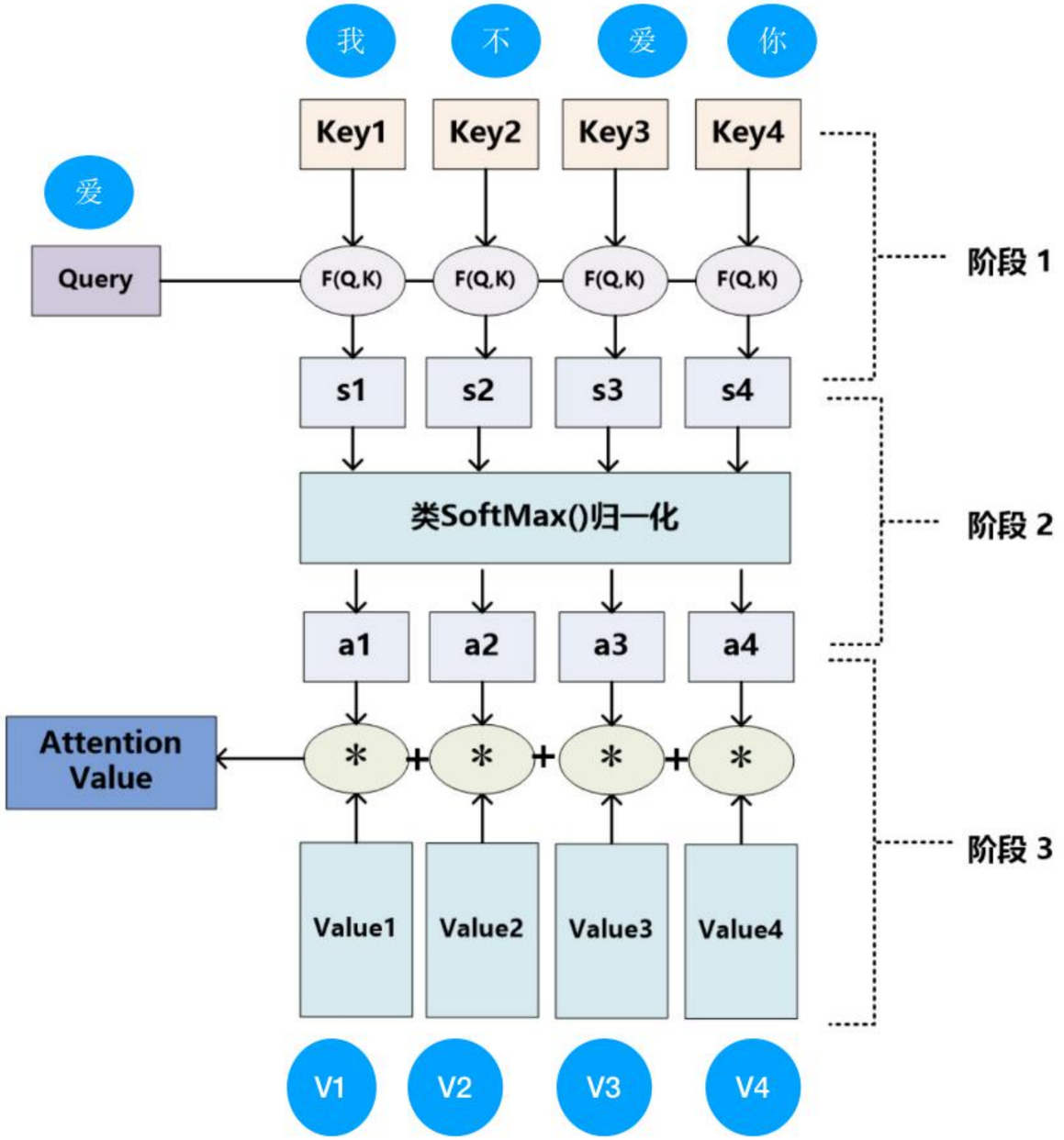
从公式角度来看：拿上面图片举例子





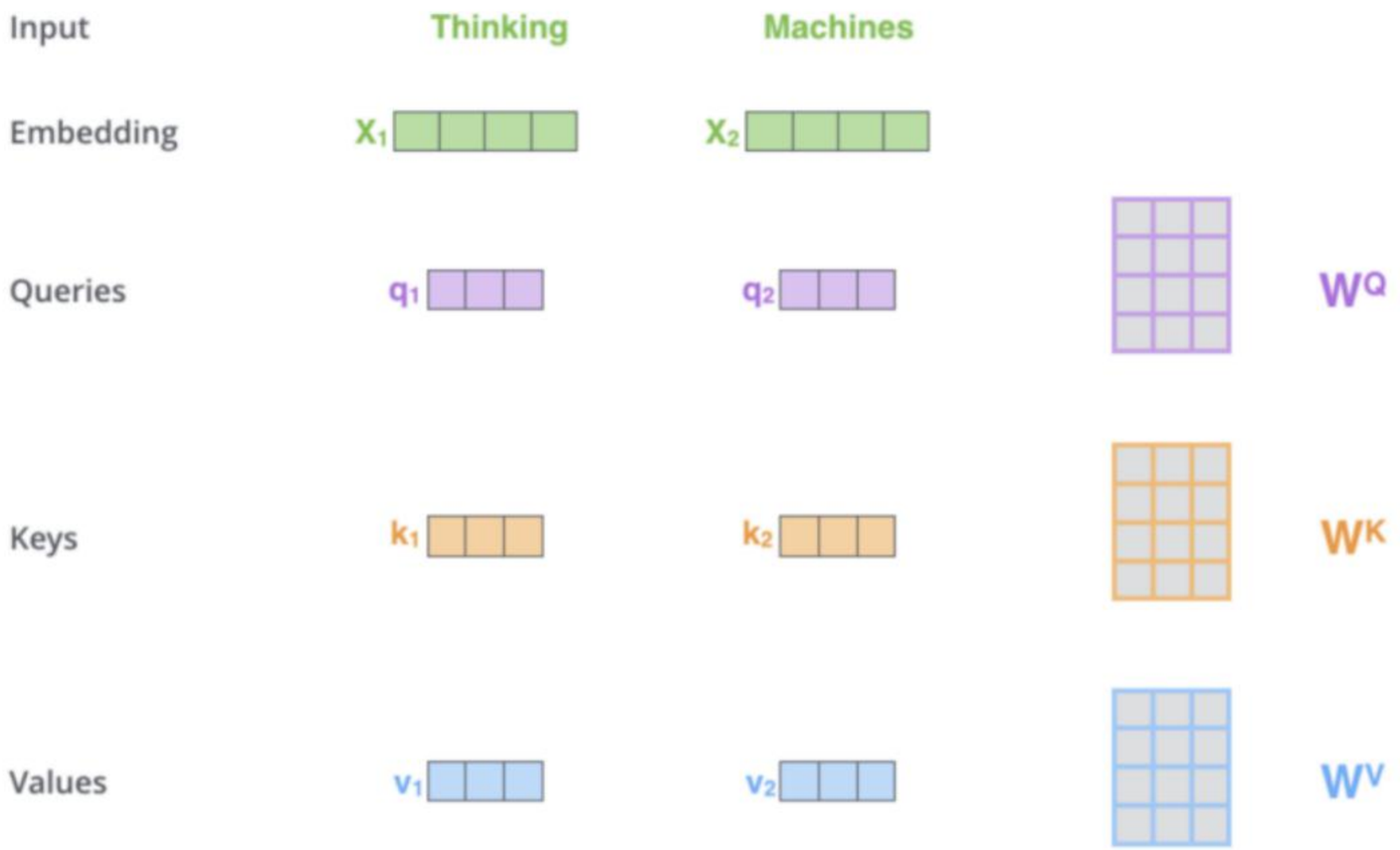
从公式角度来看


我
不
爱
你
爱
Key1
Key2
Key3
Key4
Query
Attention Value
Value1
Value2
Value3
Value4
Source
V1
V2
V3
V4

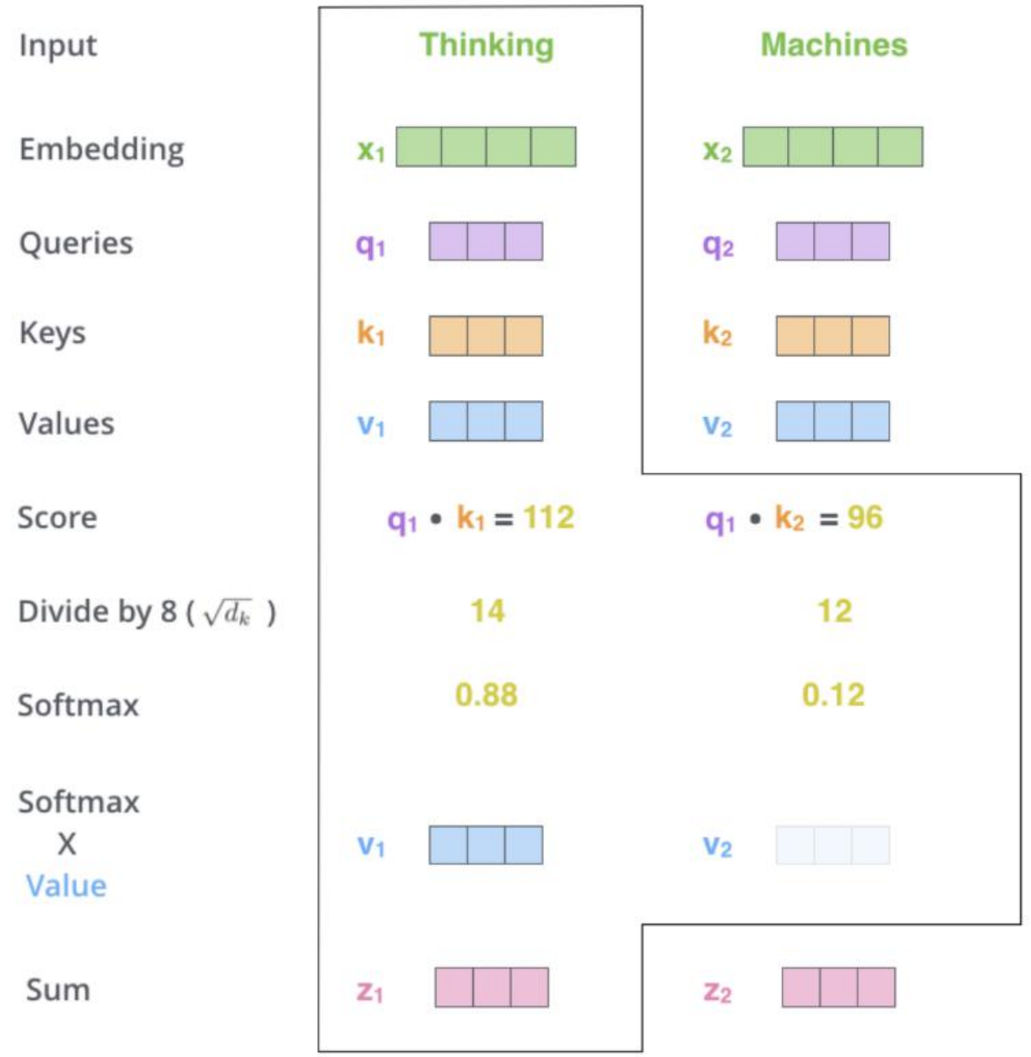

我
不
爱
你
Key1
Key2
Key3
Key4
爱
Query
F(Q,K)
F(Q,K)
F(Q,K)
F(Q,K)
阶段 1
s1
s2
s3
s4
类SoftMax()归一化
阶段 2
a1
a2
a3
a4
Attention Value
*
*
*
*
阶段 3
Value1
Value2
Value3
Value4
V1
V2
V3
V4

TRM中的注意力

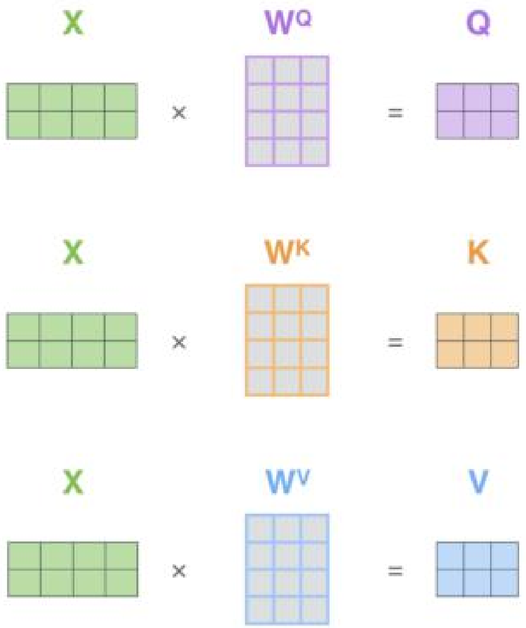
在只有单词向量的情况下，如何获取QKV

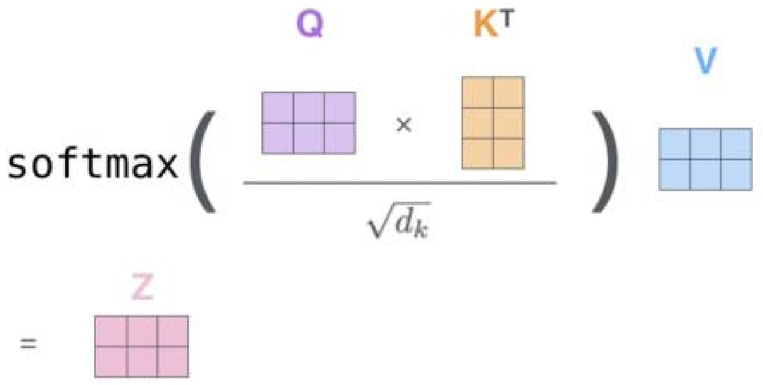


计算QK相似度，
得到attention值



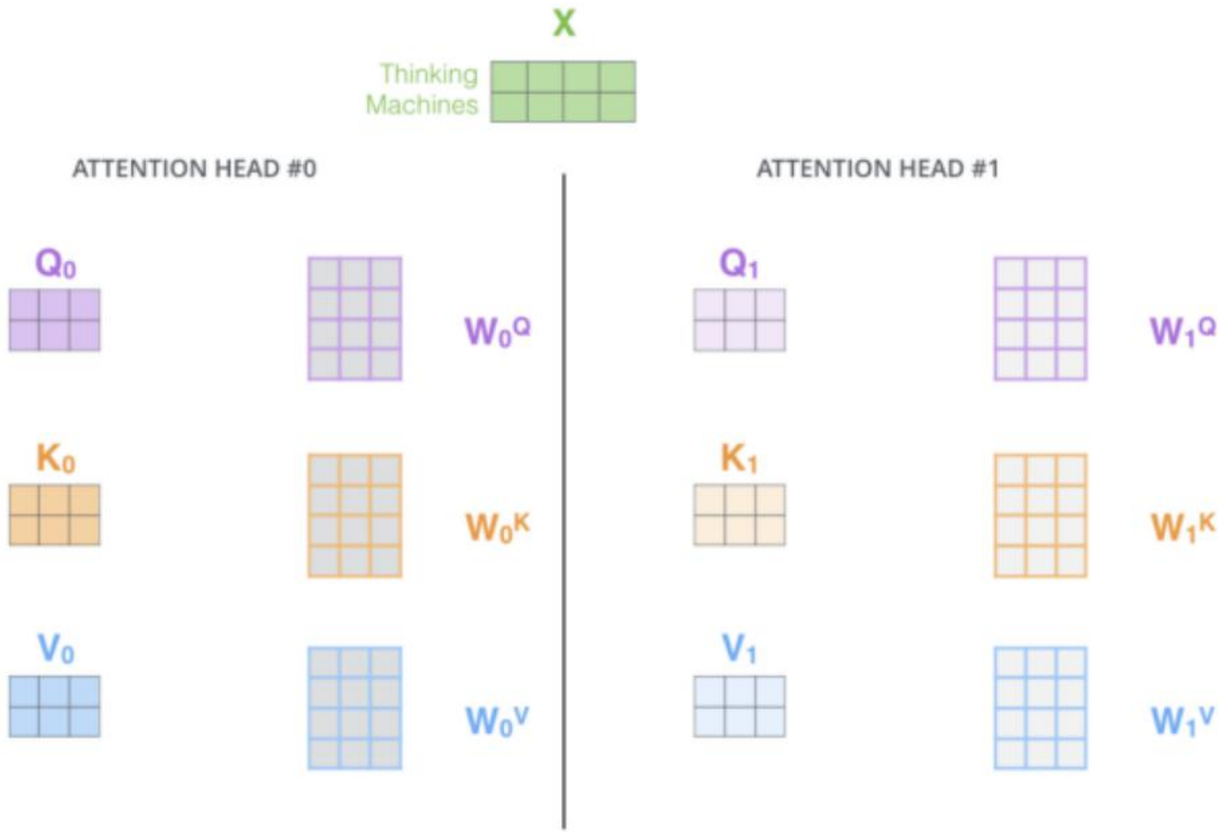
实际代码使用矩阵，方便并行



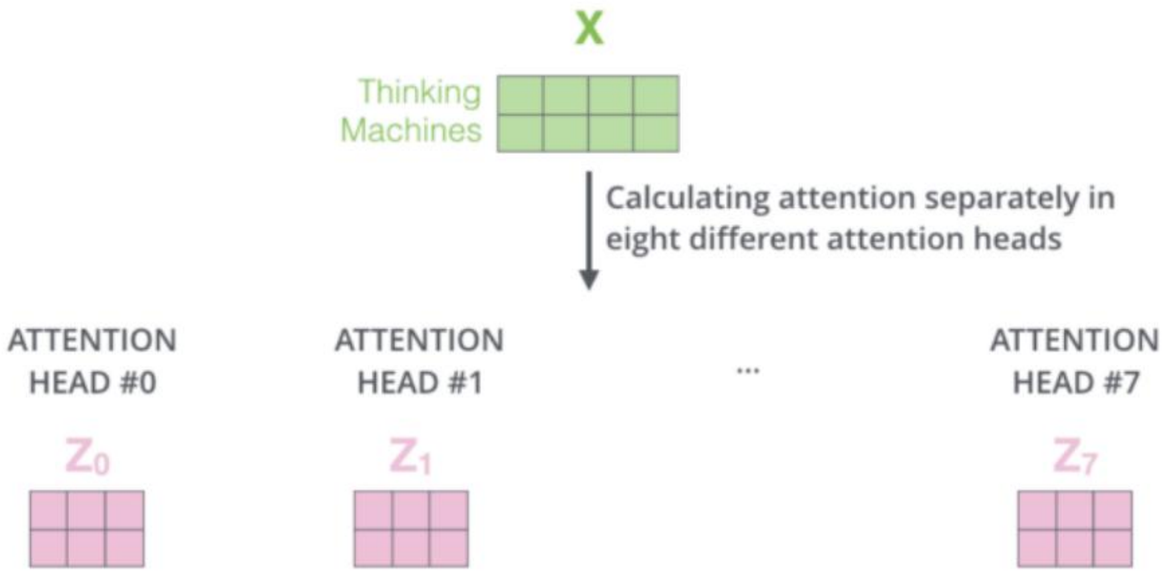


The self-attention calculation in matrix form

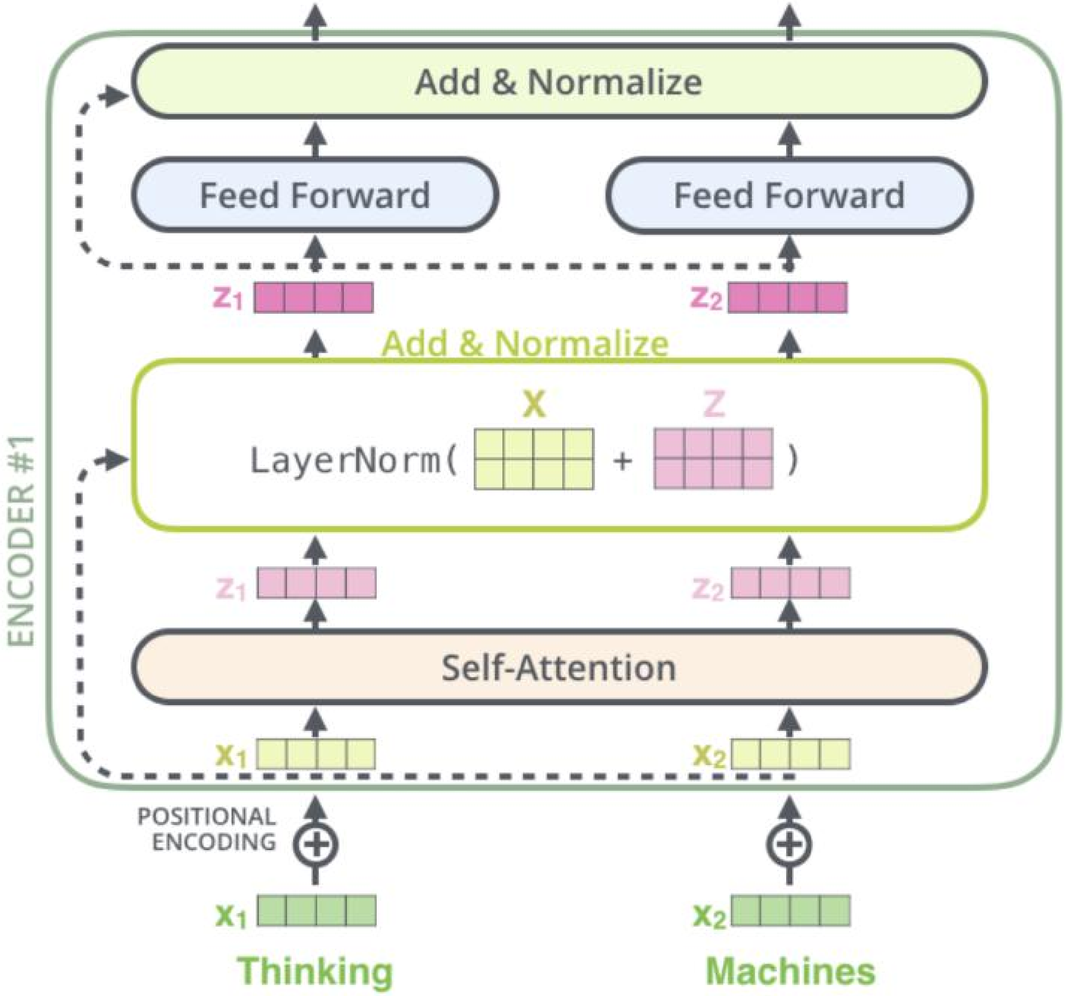
多头注意力机制


X
Thinking
Machines
ATTENTION HEAD #0
ATTENTION HEAD #1
Q0
K0
V0
W0Q
W0K
W0V
Q1
K1
V1
W1Q
W1K
W1V

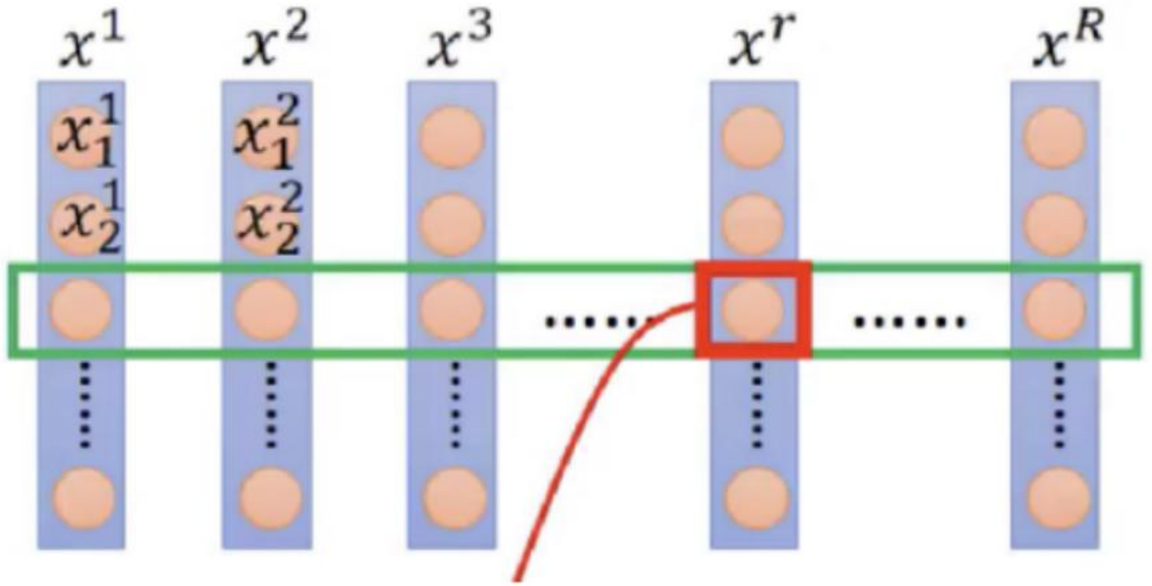
多个头就会有多个输出，需要合在一起输出



残差和LayNorm


Add & Normalize
Feed Forward
Feed Forward
z1
z2
Add & Normalize
X
Z
LayerNorm(+)
ENCODER #1
z1
z2
Self-Attention
x1
x2
POSITIONAL ENCODING
x1
x2
Thinking
Machines

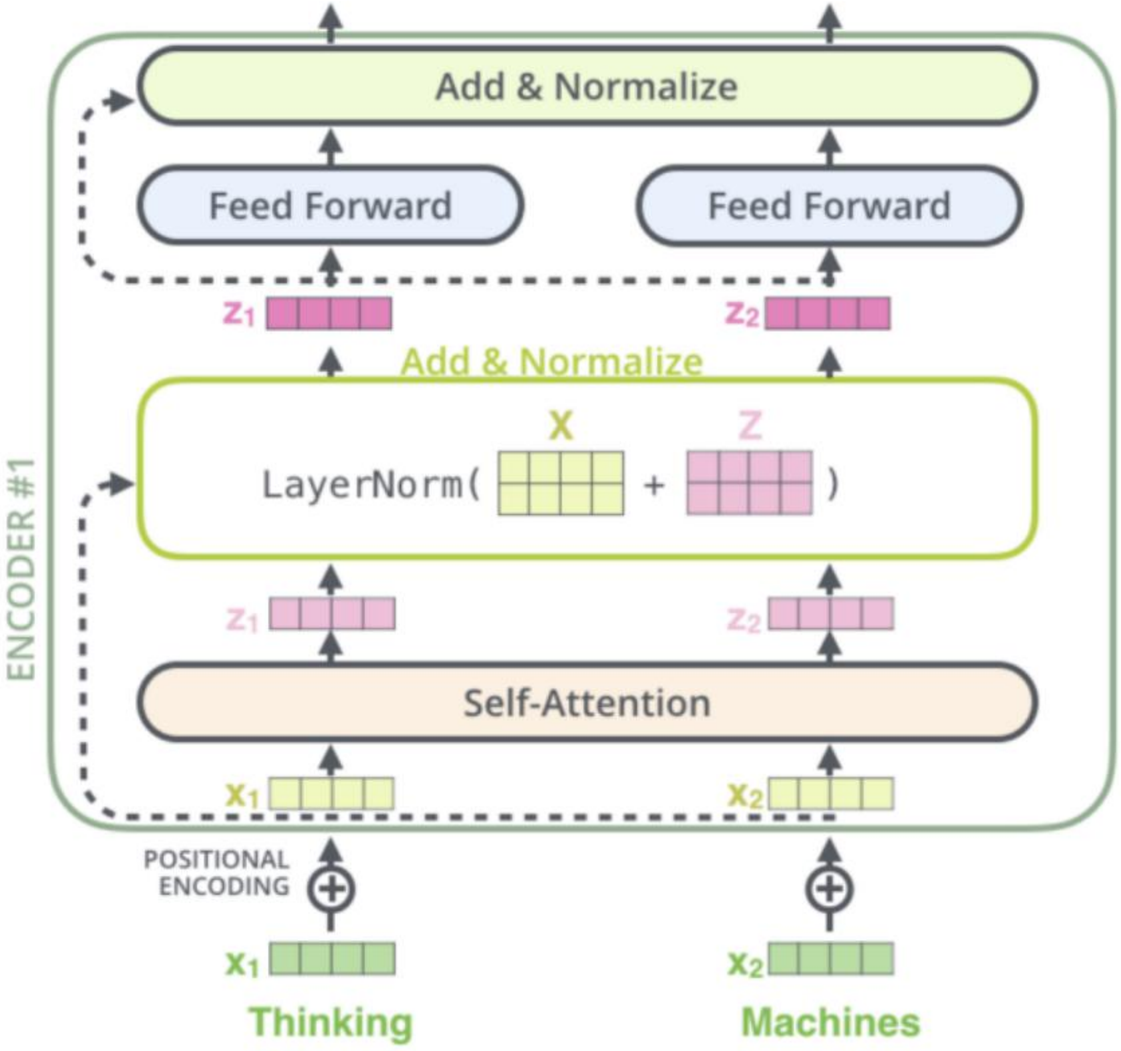
BN



理解：为什么LayerNorm单独对一个样本的所有单词做缩放可以起到效果。

我爱中国共产党

今天天气真不错


Add & Normalize
Feed Forward
Feed Forward
z1
z2
Add & Normalize
X
Z
LayerNorm(+)
z1
z2
Self-Attention
x1
x2
ENCODER #1
POSITIONAL ENCODING
x1
x2
Thinking
Machines

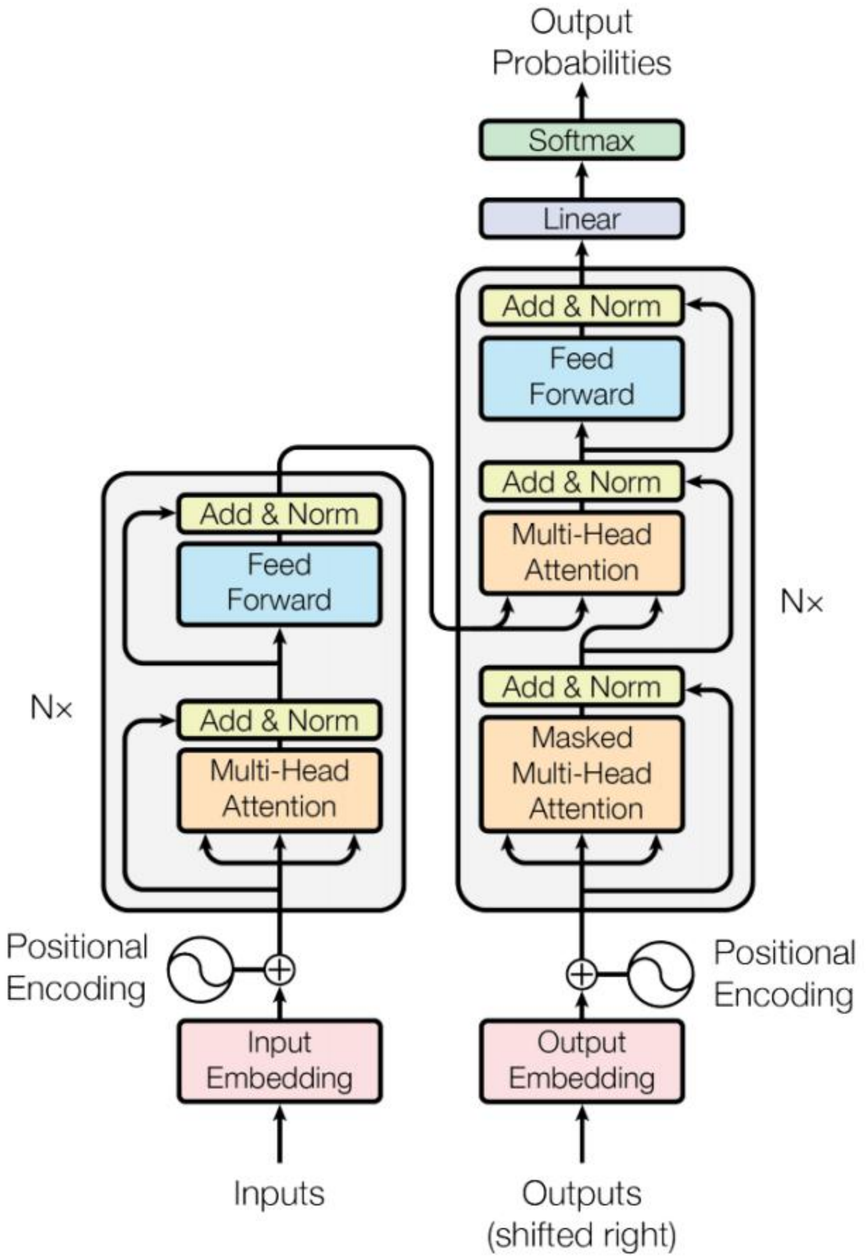


Figure 1: The Transformer - model architecture.

Vision Transformer

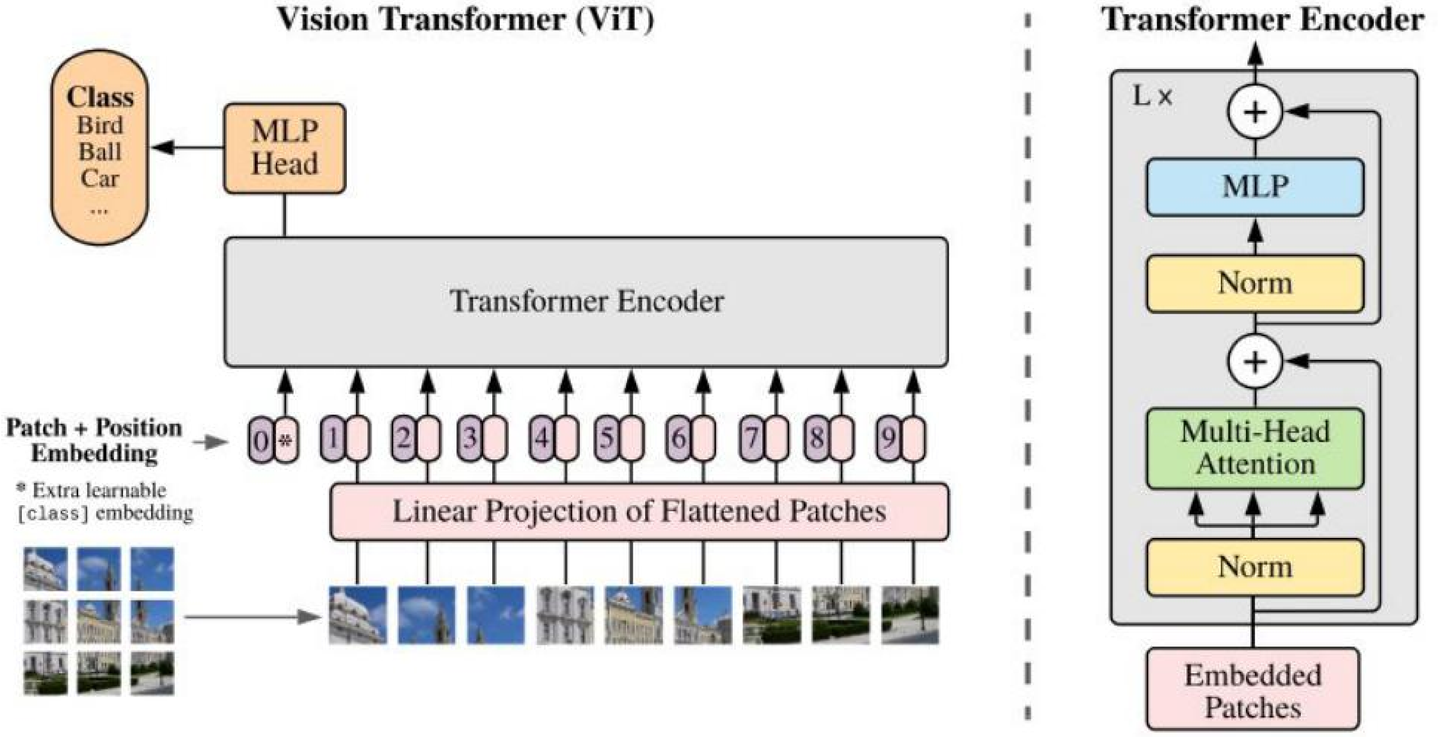
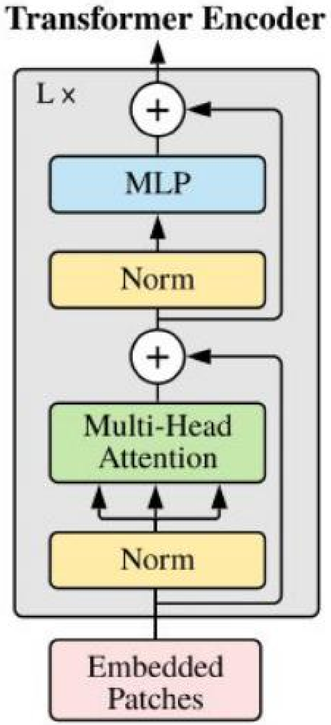


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

Vision Transformer



The Transformer encoder (Vaswani et al., 2017) consists of alternating layers of multiheaded self-attention (MSA, see Appendix A) and MLP blocks (Eq. 2, 3). Layernorm (LN) is applied before every block, and residual connections after every block (Wang et al., 2019; Baevski & Auli, 2019). The MLP contains two layers with a GELU non-linearity.

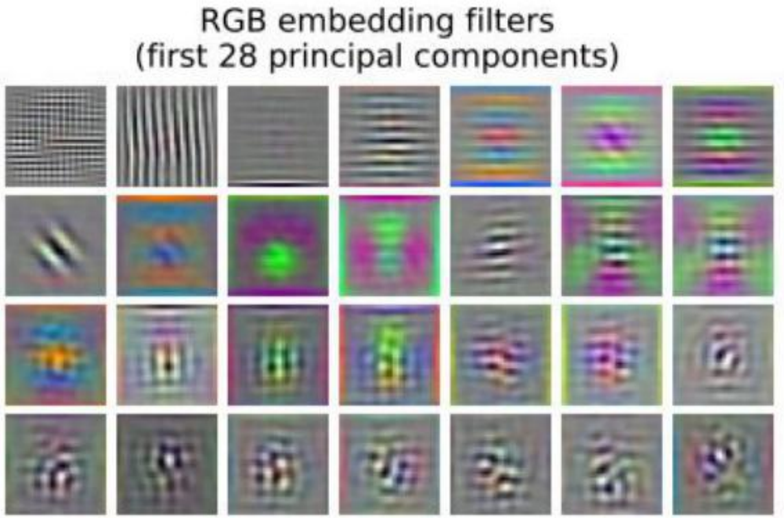
$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots ; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \ \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \tag{1}$$

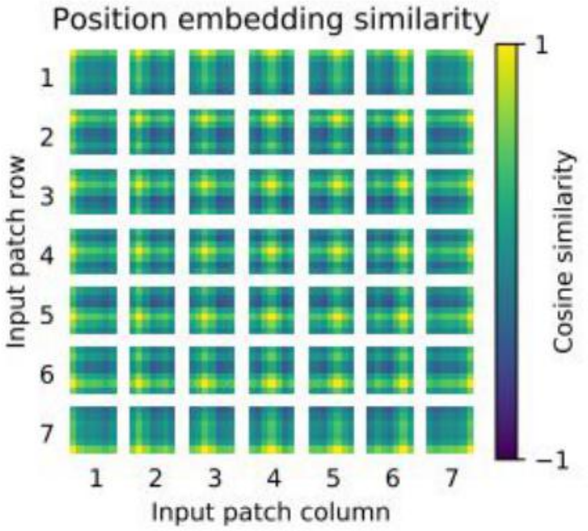
$$\mathbf{z'}_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L \tag{2}$$

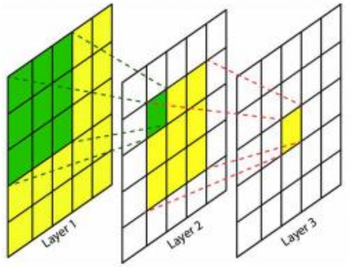
$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell, \qquad \ell = 1 \ldots L \tag{3}$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \tag{4}$$

Results







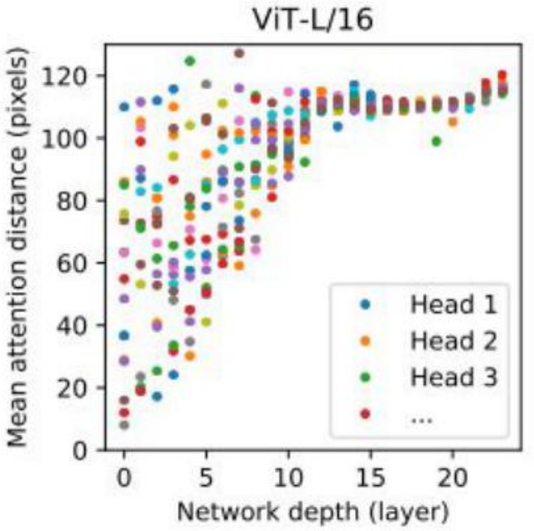


Figure 7: **Left:** Filters of the initial linear embedding of RGB values of ViT-L/32. **Center:** Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches. **Right:** Size of attended area by head and network depth. Each dot shows the mean attention distance across images for one of 16 heads at one layer. See Appendix D.6 for details.

Results - Positional Embedding

Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

Table 8: Results of the ablation study on positional embeddings with ViT-B/16 model evaluated on ImageNet 5-shot linear.

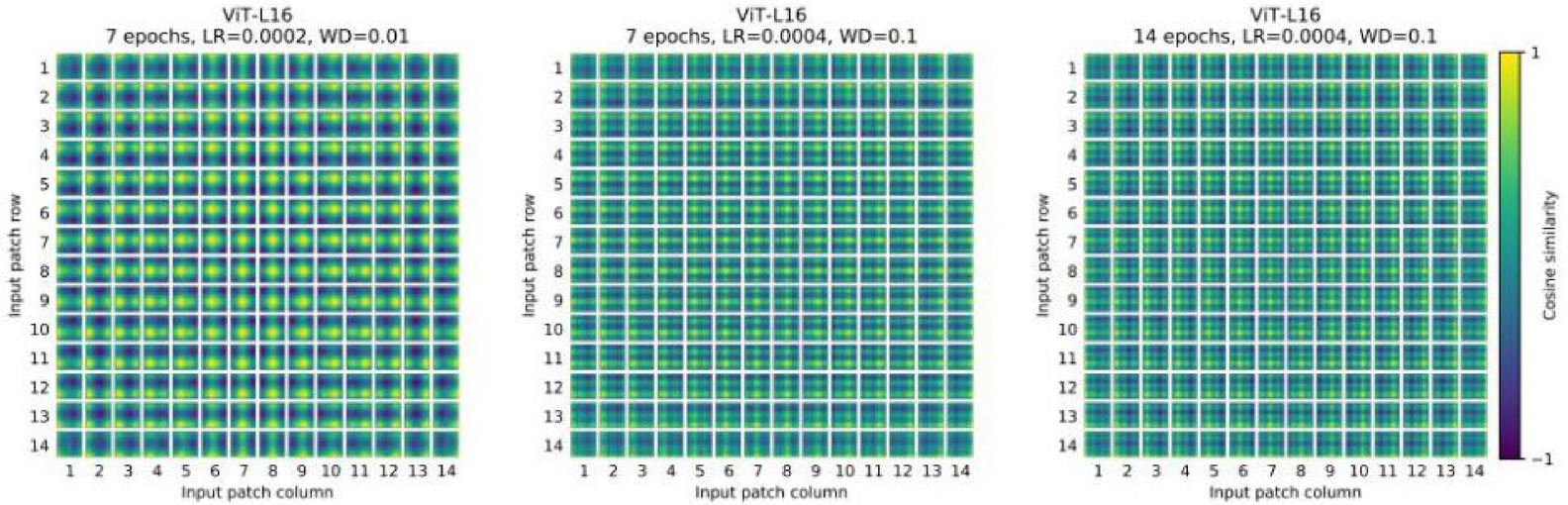


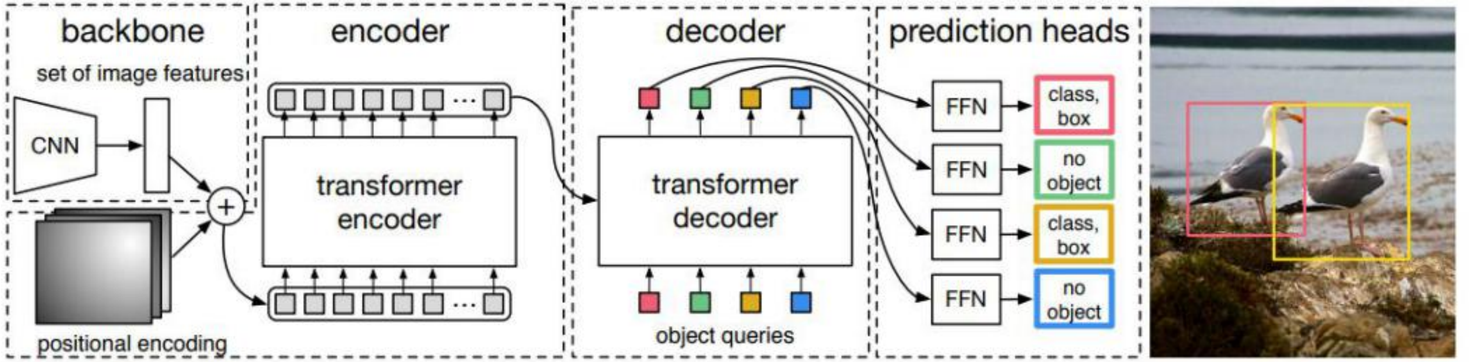
Figure 9: Position embeddings of models trained with different hyperparameters.

Conclusion

+ Direct application of Transformers to image recognition without any image-specific inductive biases.

TODO:

- apply ViT to other computer vision tasks



Carion, Nicolas, et al. "End-to-end object detection with transformers." *European Conference on Computer Vision*. Springer, Cham, 2020.

- Self- supervised pre-training methods.

MAE--masked autoencoders

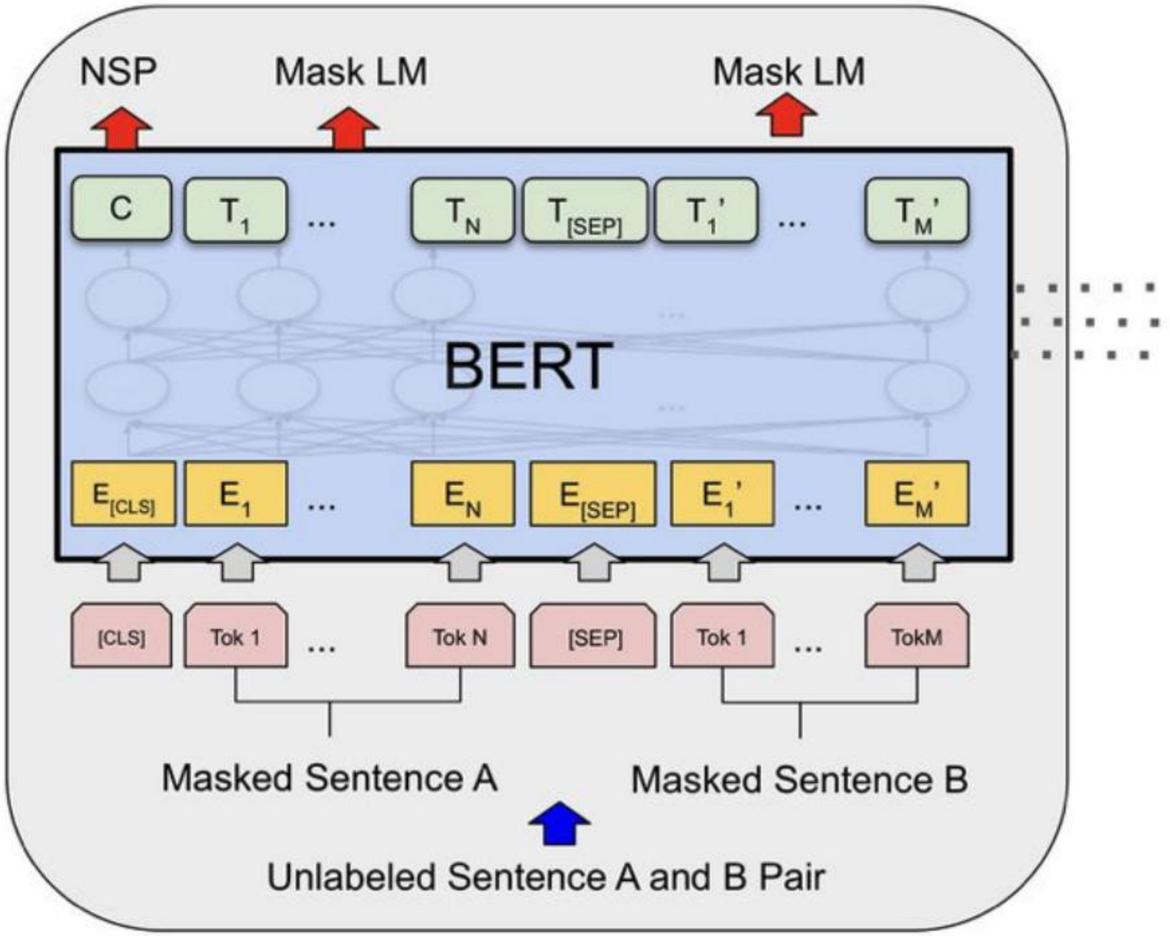
Overview

1. 为什么要去做无监督预训练任务？

2. cv中做这件事情的难点

3. MAE 方法结构及实验介绍

4. 我们能得到什么启发，做些什么工作



无监督预训练 ⟶ 数据无需标注（易得准确）
规模量大

CV中的难点

1. 网络基础架构的局限性（思考CNN） -- > VIT

2. 如何针对图片设计自监督任务（分类不行） --> 回归

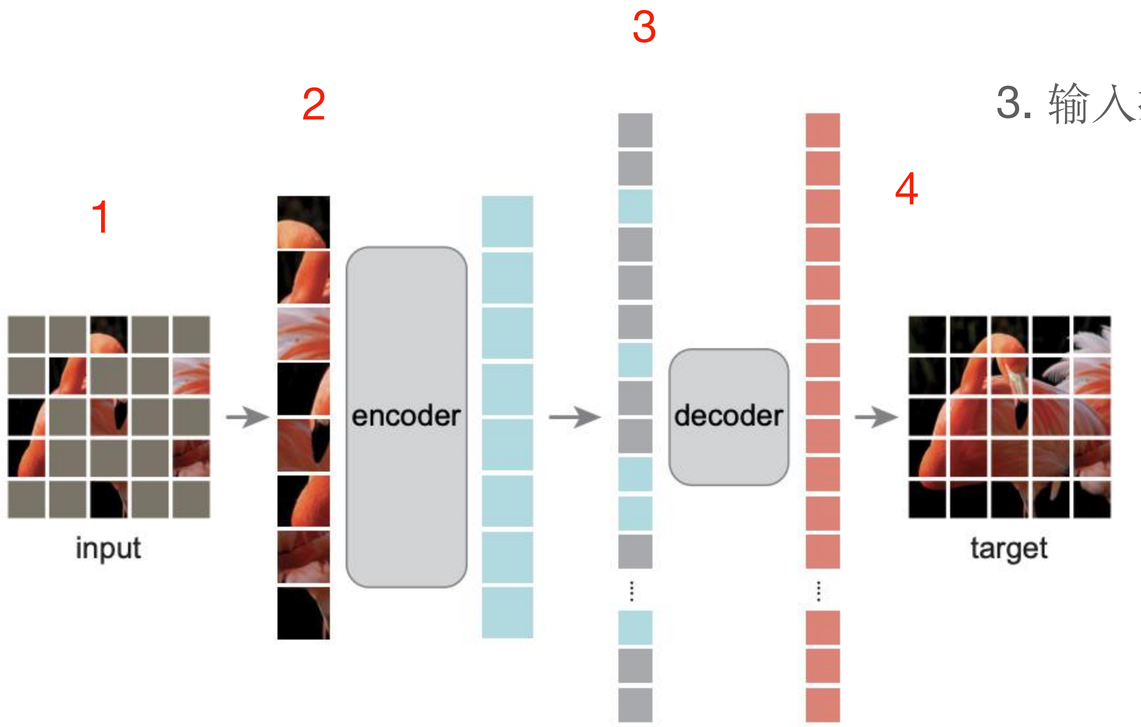
3. 图片和文字信息结构的不同

（大规模的数据 + 合适的网络架构 + 合适的自监督任务）

1. 划分patch，同时mask掉75%的patch

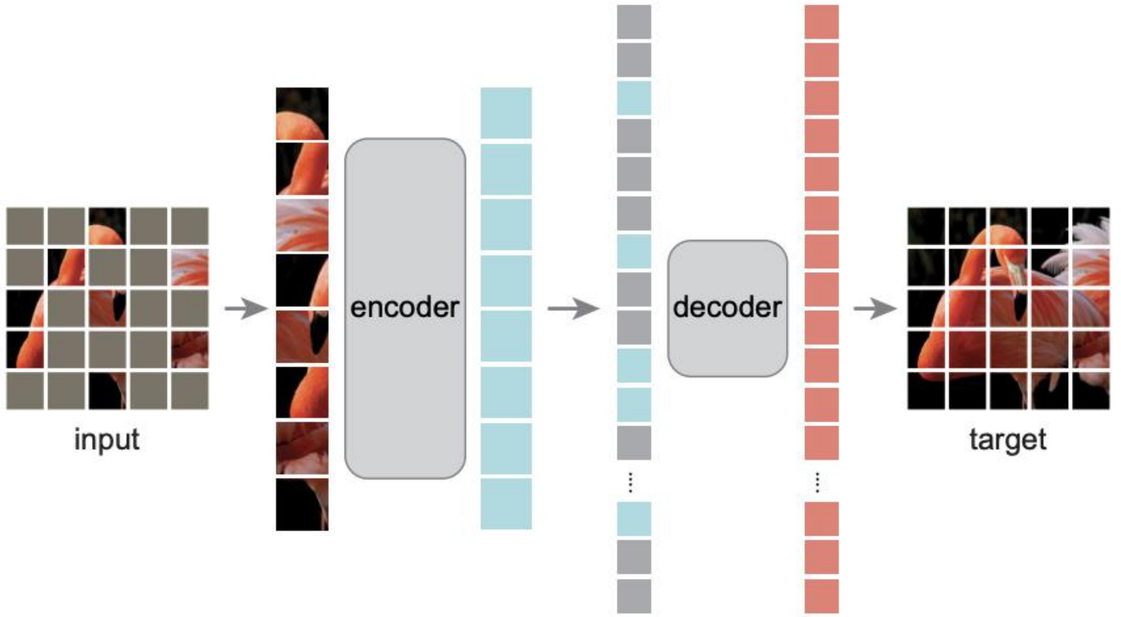
2. 然后把可见patch输入到encoder中去；使用vit模型

3. 输入拼接上mask patch，过decoder



4. 解码端之后有一个隐层映射，做损失

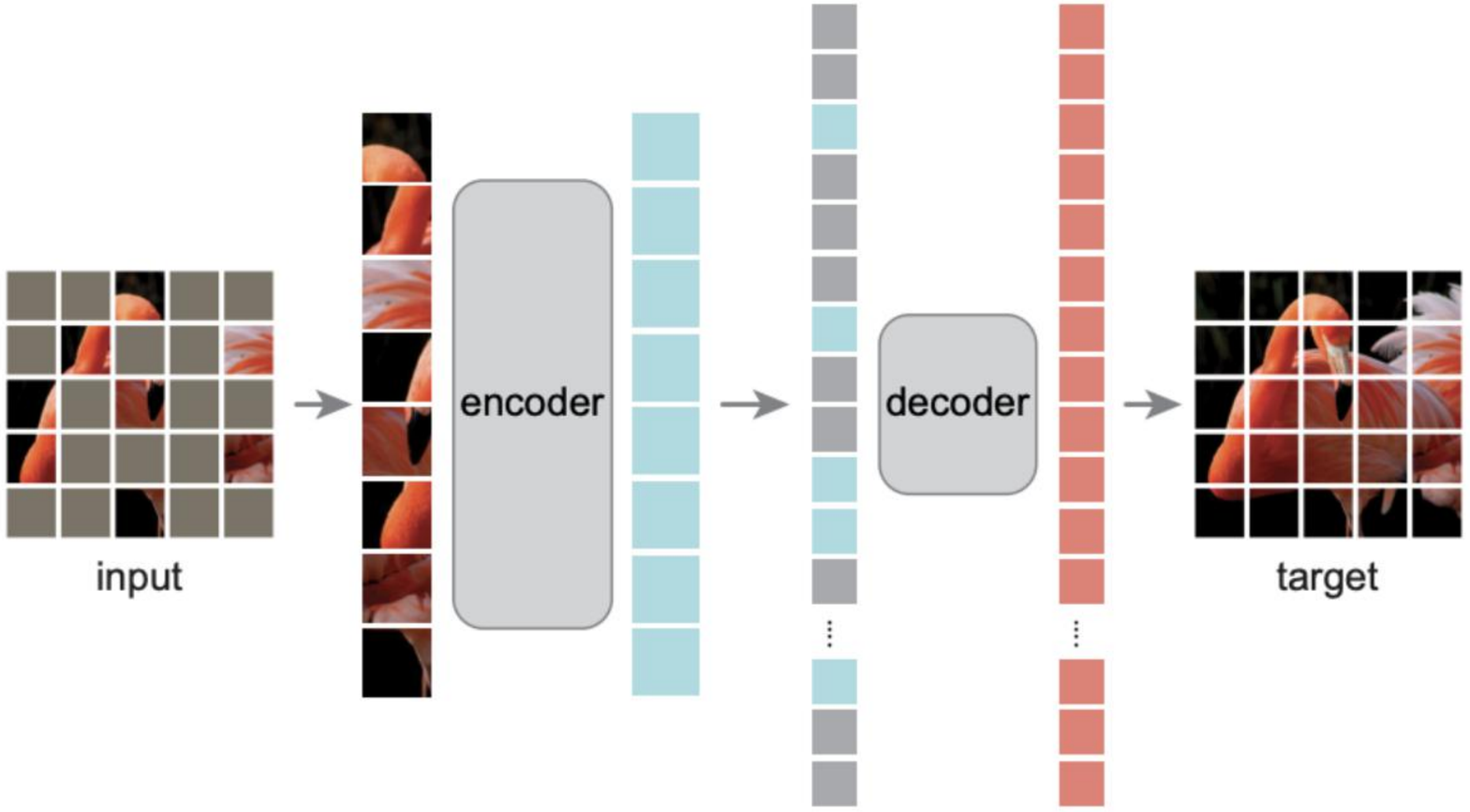
MAE哪个部分用于下游任务



encoder抽取图像特征

decoder在做图像重建

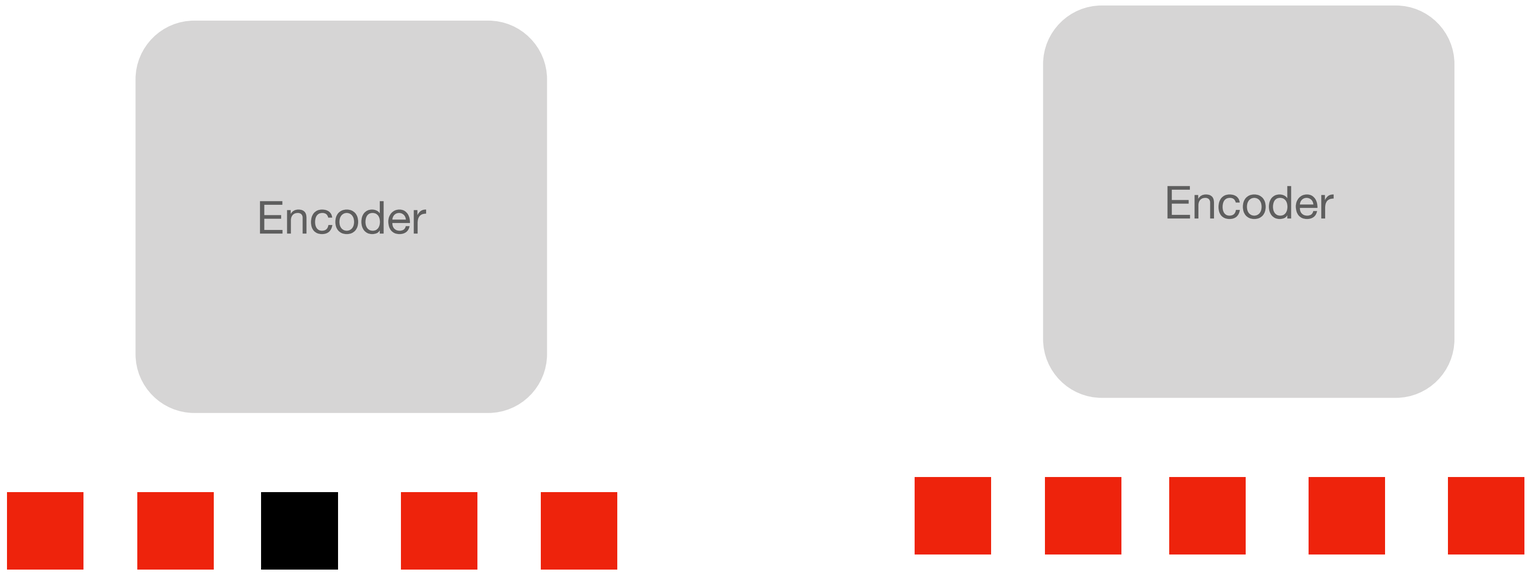
为什么把mask符号放在解码端，而不是编码端



一个最常规的解释

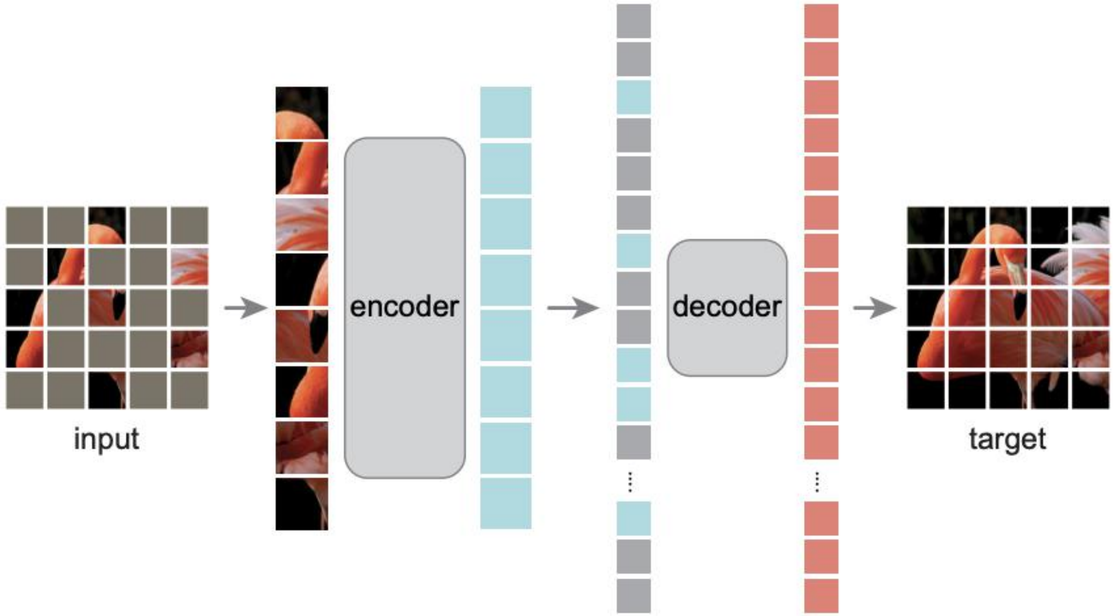
预训练

下游任务微调



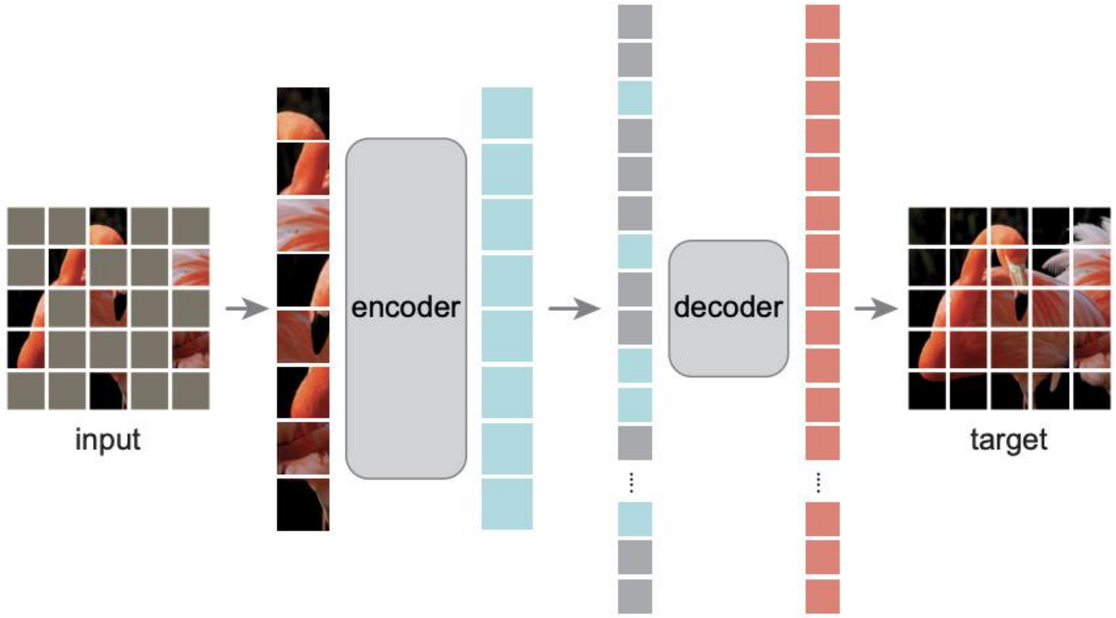
MAE的做法更加彻底

BERT是在缩小这个差距，MAE试图消除这个影响，注意是试图

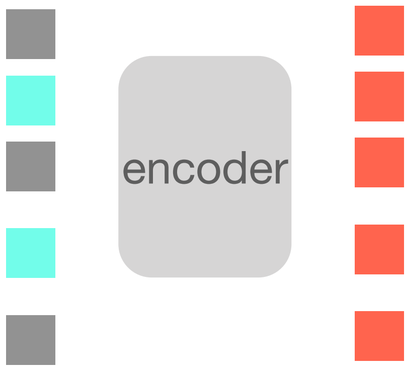


BERT 和 MAE 类比

MAE

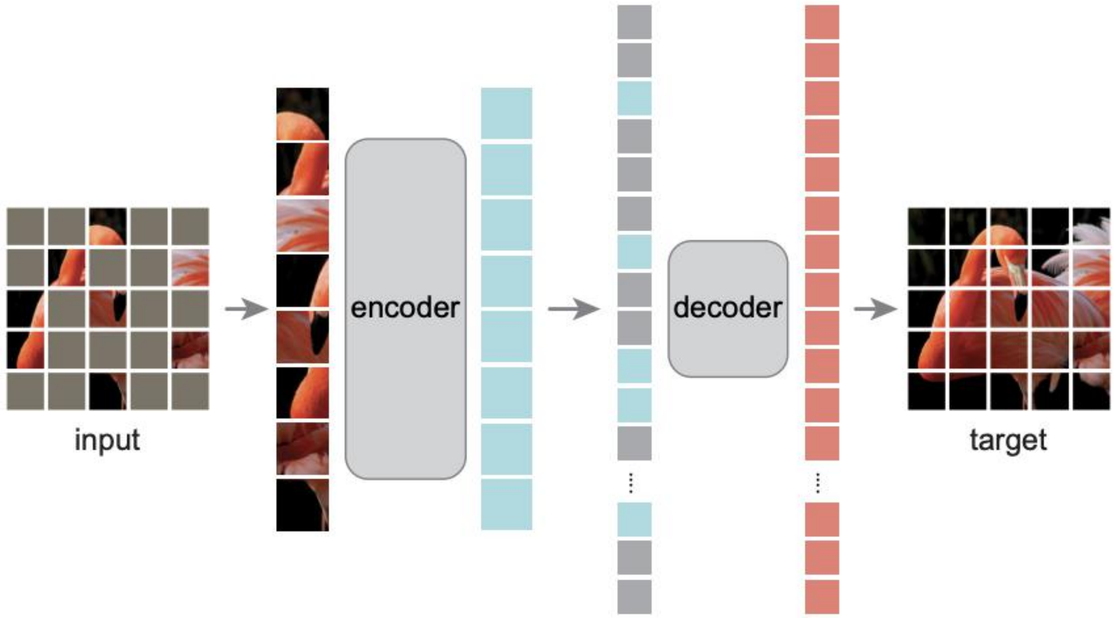


BERT

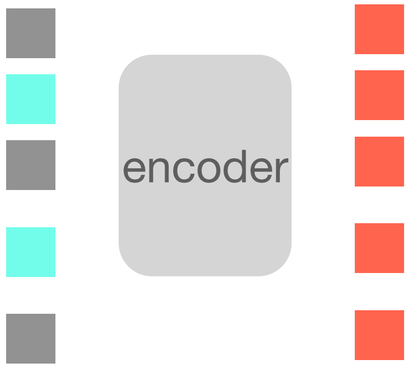


有么有发现一个特点，就是BERT其实很类似MAE的decoder的部分

MAE



BERT



如何评价MAE的效果

可视化的方式

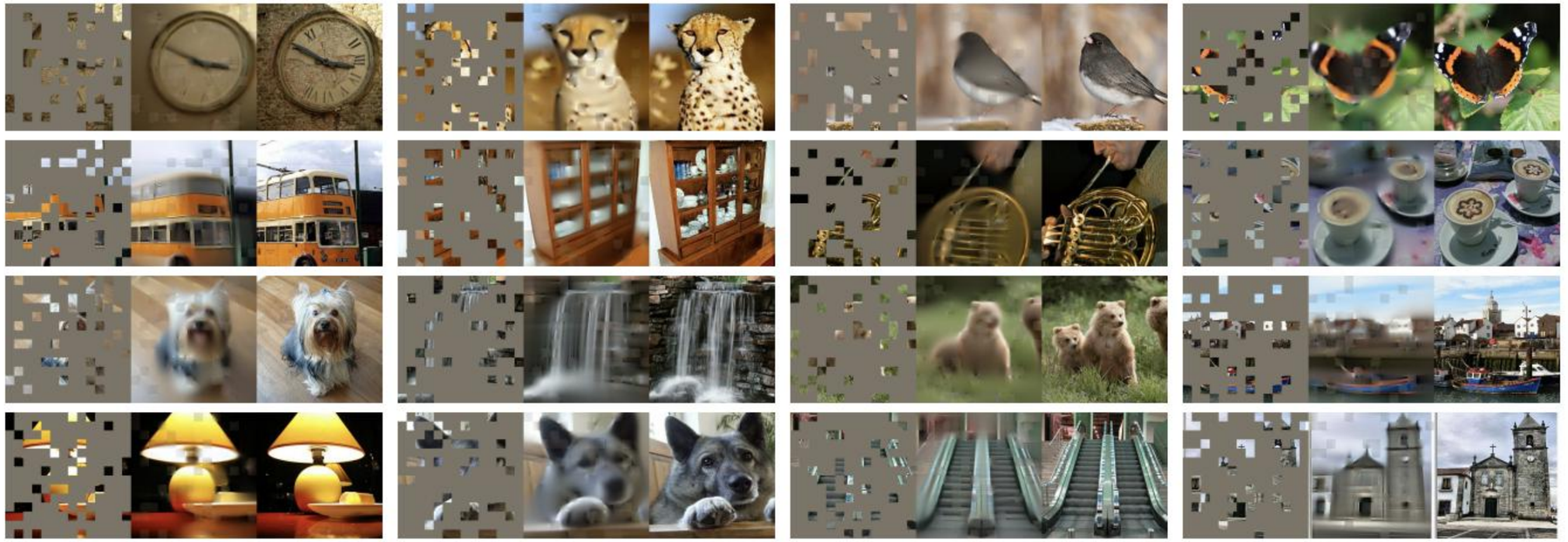


Figure 2. Example results on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction[†] (middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches. More examples are in the appendix.

[†]*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.*

泛化性

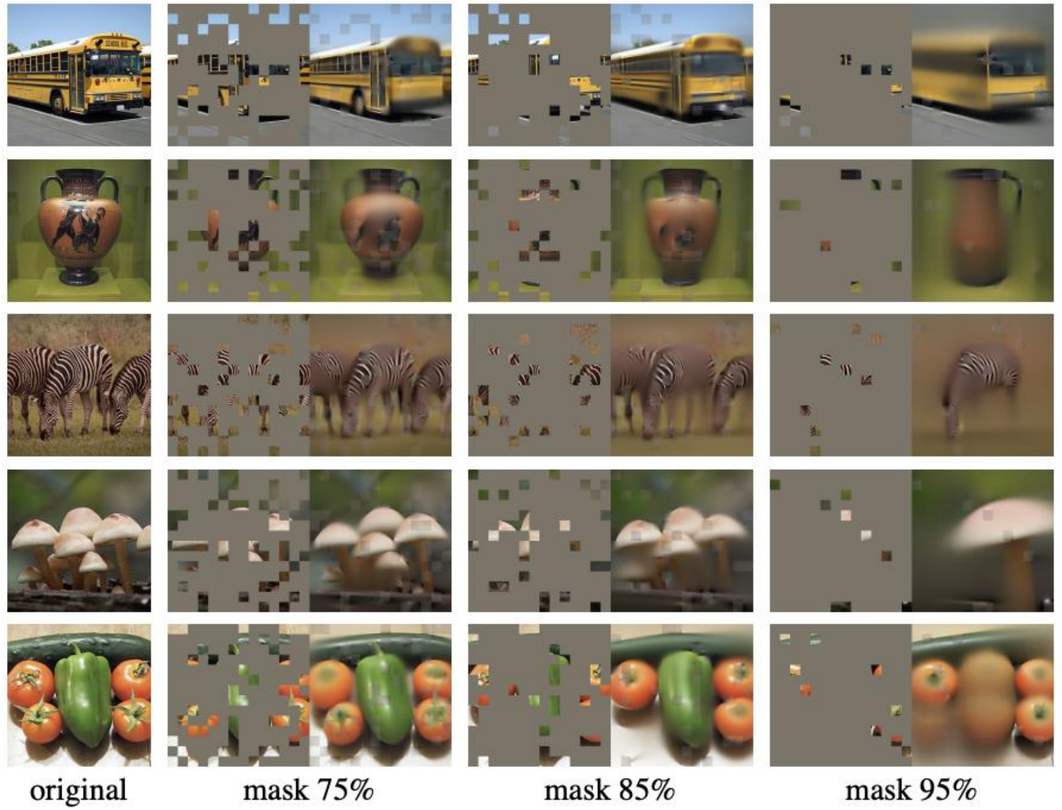


Figure 4. Reconstructions of ImageNet *validation* images using an MAE pre-trained with a masking ratio of 75% but applied on inputs with higher masking ratios. The predictions differ plausibly from the original images, showing that the method can generalize.

两种度量方式

Linear probe　　固定encoder参数，学习linear层

Fine tune　　整个模型包括encoder一起学习

实验效果

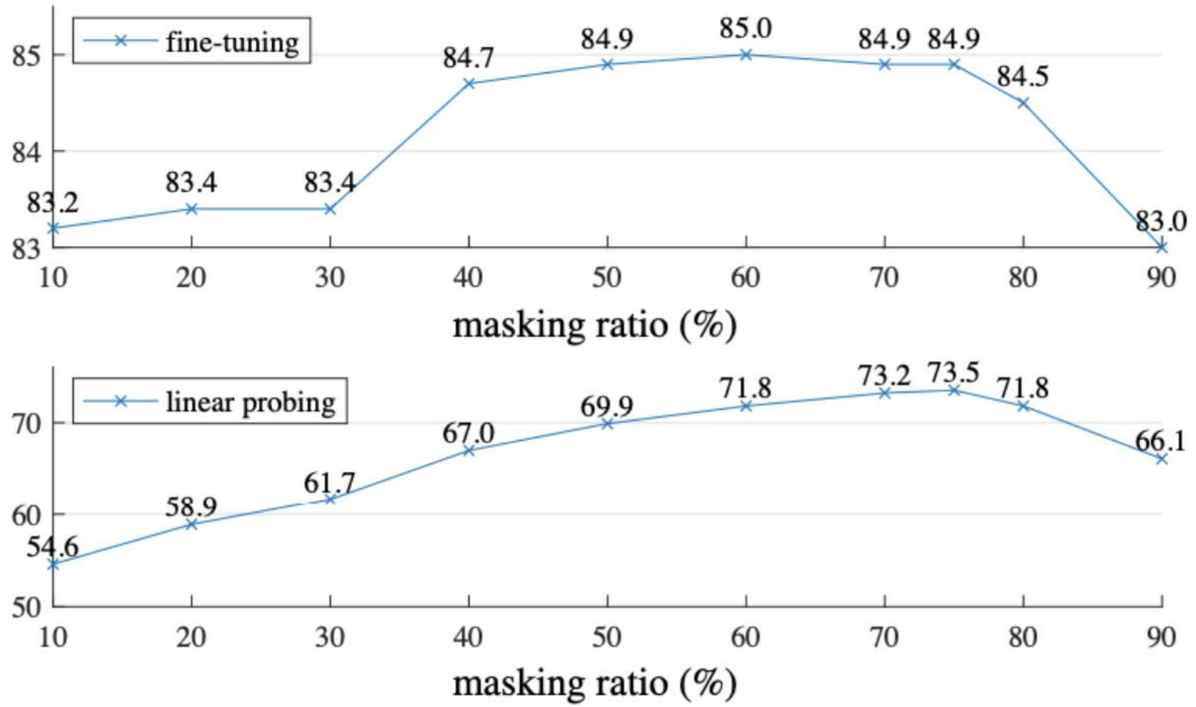


Figure 5. **Masking ratio**. A high masking ratio (75%) works well for both fine-tuning (top) and linear probing (bottom). The y-axes are ImageNet-1K validation accuracy (%) in all plots in this paper.

Decoder的宽度和高度

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).

开始加上Mask符号

在VIT那个论文做了实验了，效果不好

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

ar-

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

Reconstruction target

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.

发现基于token的target相比于基于pixel的target不占优势

，带norm的pixel的target同时在fine-tuning和linear-tuning达到最优，

Mask样式-随机最好

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.

随机采样得到的visible patches组合多样性更好

random 75%　　block 50%　　grid 75%

评价指标的问题

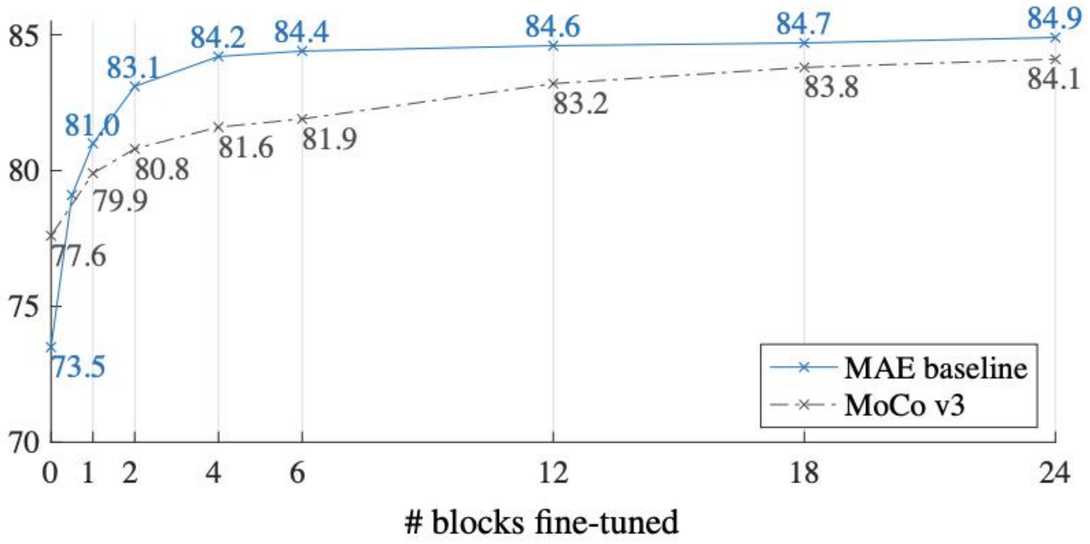


Figure 9. **Partial fine-tuning** results of ViT-L w.r.t. the number of fine-tuned Transformer blocks under the default settings from Table 1. Tuning 0 blocks is linear probing; 24 is full fine-tuning. Our MAE representations are less linearly separable, but are consistently better than MoCo v3 if one or more blocks are tuned.

Future work

1. 如果你使用VIT为基础的模型架构，可以拿MAE的预训练模型做下游任务

2. 去学习这种“mask autoencoder” 的核心思想，能否将其套用或者变化入我们的下游任务当中