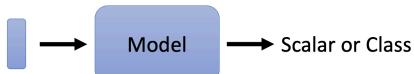


self-attention

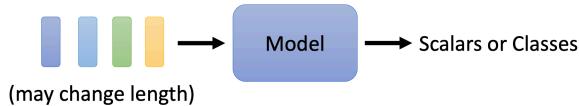
应用背景

普通CNN输入为一个向量，无法应对输入为多个向量的情况：

- Input is a vector



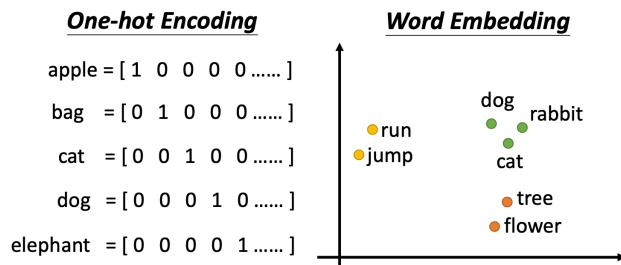
- Input is a set of vectors



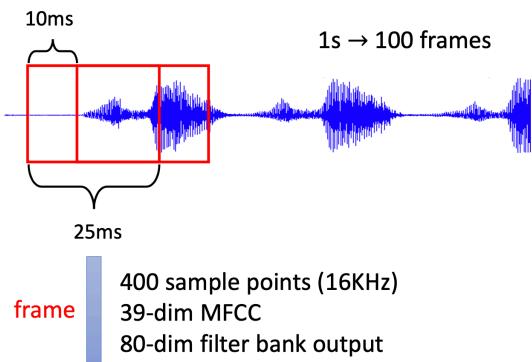
应用场景

输入

1. 词向量作为输入（2种词编码方式：one-hot Encoding和word Embedding）

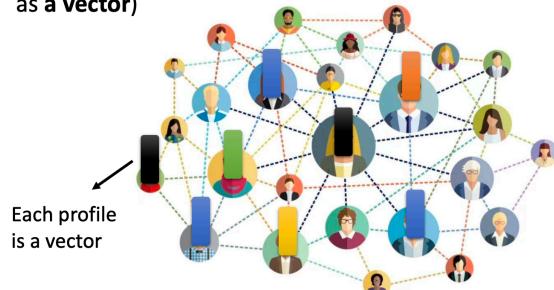


2. 语音作为输入



3. 图作为输入

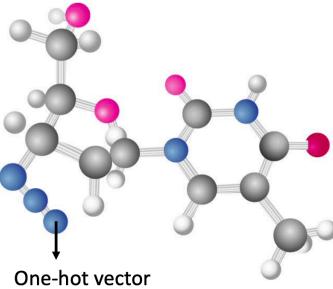
- Graph is also a set of vectors (consider each node as a vector)



4. 分子结构作为输入

- Graph is also a set of vectors (consider each **node** as a **vector**)

$$\begin{aligned} H &= [1 \ 0 \ 0 \ 0 \ 0 \dots] \\ C &= [0 \ 1 \ 0 \ 0 \ 0 \dots] \\ O &= [0 \ 0 \ 1 \ 0 \ 0 \dots] \\ \vdots & \end{aligned}$$



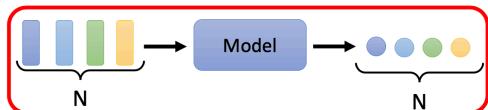
5. 等...

输出

1. N-N: 词性标注、客户筛选（重点关注）
2. N-1: 语义评价、分子性质分类
3. N-M: 翻译

如下:

- Each vector has a label. focus of this lecture



- The whole sequence has a label.



- Model decides the number of labels itself. seq2seq

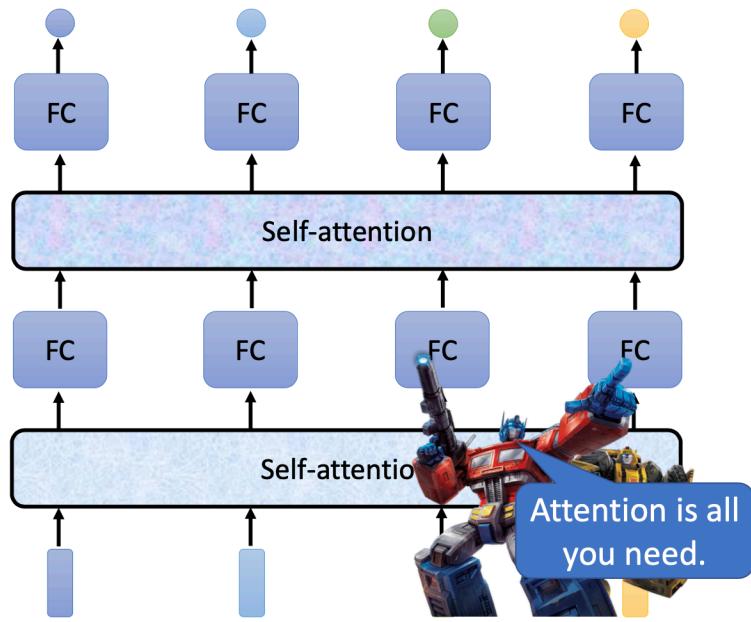


Self-attention实现

要求

以输入输出N-N为例：取window包含N个输入，N个输入在互相关联（考虑上下文）后通过self-attention产生N个输出。

如下图transformer为例：

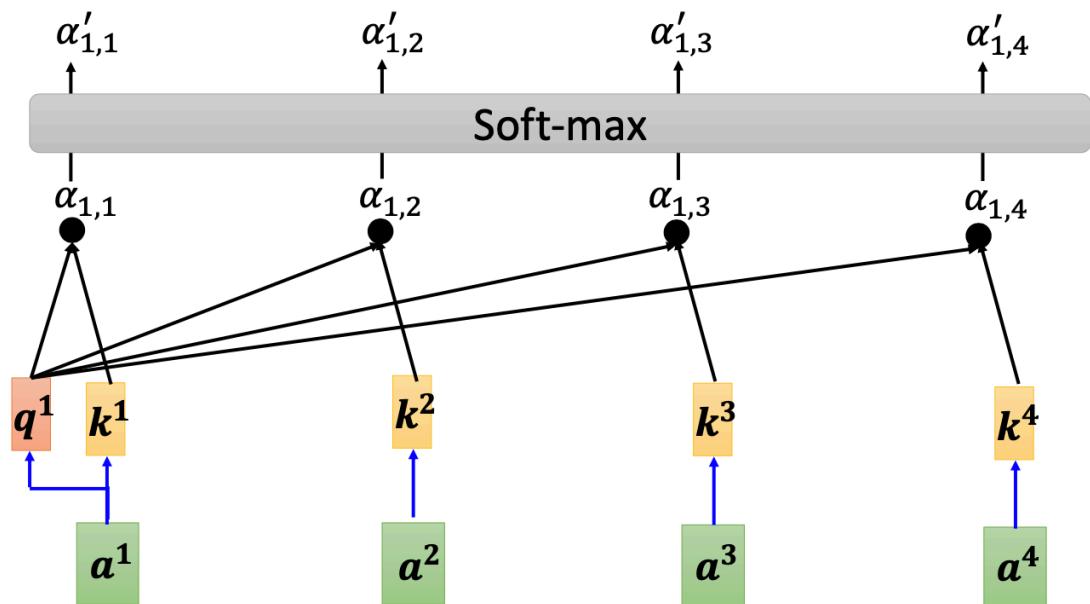


<https://arxiv.org/abs/1706.03762>

self-attention具体实现

Self-attention

$$\alpha'_{1,i} = \exp(\alpha_{1,i}) / \sum_j \exp(\alpha_{1,j})$$



$$q^1 = W^q a^1 \quad k^2 = W^k a^2 \quad k^3 = W^k a^3 \quad k^4 = W^k a^4$$

$$k^1 = W^k a^1$$

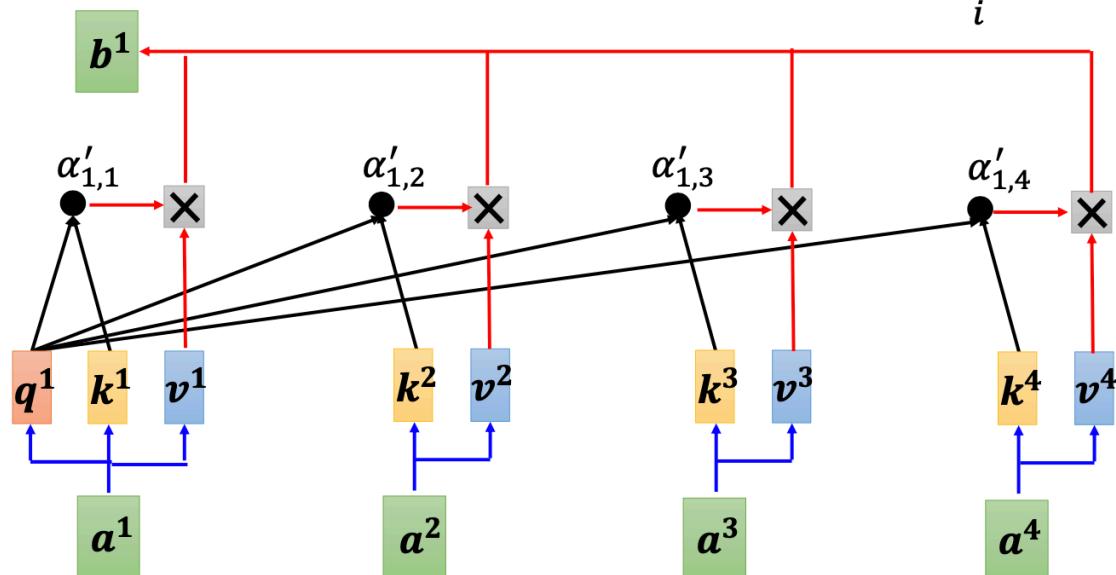
17

图1

Self-attention

Extract information based
on attention scores

$$b^1 = \sum_i \alpha'_{1,i} v^i$$



$$v^1 = W^v a^1$$

$$v^2 = W^v a^2$$

$$v^3 = W^v a^3$$

$$v^4 = W^v a^4$$

图2

步骤如下：

- 分别计算每个输入的query，即 $q^i = W^q * a^i$ ，其中 W^q 为 a 到 q 的映射矩阵， a^i 为输入， q^i 为query用以查询其与其他key之间的相关性；
- 分别计算每个输入的key，即 $k^i = W^k * a^i$ ，其中 W^k 为 a 到 k 的映射矩阵， a^i 为输入， k^i 为key用以被查询query与其之间的相关性；
- 分别计算各个query与各个key之间的相关性，即 $\alpha'_{i,j} = \text{softmax}(q^i * k^j)$ ，其中 $\alpha'_{i,j}$ 为归一化后的相关性，反映 a^i 与 a^j 之间的相关性（激活函数可任意选择，不限于softmax）；
- 分别计算每个输入的value，即 $v^i = W^v * a^i$ ，其中 W^v 为 a 到 v 的映射矩阵， a^i 为输入， v^i 为value用以表示输入向量；
- 分别计算每个输出b，即 $b^i = \sum_j \alpha'_{i,j} * v^j$ ，其中 b^i 表示输入 a^i 对应的输出。

注意：

- 由上述步骤知：输入 a^i 与输入 a^j 之间的相关性越大， $\alpha_{i,j}$ 越大，相对应的， b^i 就接近 v^j 。

self-attention矩阵实现

- I到Q,K,V的矩阵实现：

$$\begin{aligned} q^i &= W^q a^i & q^1 q^2 q^3 q^4 &= W^q a^1 a^2 a^3 a^4 \\ && Q && I \\ k^i &= W^k a^i & k^1 k^2 k^3 k^4 &= W^k a^1 a^2 a^3 a^4 \\ && K && I \\ v^i &= W^v a^i & v^1 v^2 v^3 v^4 &= W^v a^1 a^2 a^3 a^4 \\ && V && I \end{aligned}$$

- K,Q到A/A'的矩阵实现：

$$A' \xrightarrow{\text{softmax}} A = \begin{matrix} \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4} \end{matrix} = \begin{matrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{matrix} \quad \begin{matrix} q^1 \\ q^2 \\ q^3 \\ q^4 \end{matrix} \quad K^T$$

23

- V,A'到O的矩阵实现：

$$b^1 b^2 b^3 b^4 = v^1 v^2 v^3 v^4$$

$\alpha'_{1,1}$	$\alpha'_{2,1}$	$\alpha'_{3,1}$	$\alpha'_{4,1}$
$\alpha'_{1,2}$	$\alpha'_{2,2}$	$\alpha'_{3,2}$	$\alpha'_{4,2}$
$\alpha'_{1,3}$	$\alpha'_{2,3}$	$\alpha'_{3,3}$	$\alpha'_{4,3}$
$\alpha'_{1,4}$	$\alpha'_{2,4}$	$\alpha'_{3,4}$	$\alpha'_{4,4}$

$$O = V \quad A'$$

self-attention拓展

Multi-head self-attention

优点：使用多个head，每个head关注的相关性不同，可以从多方面计算得到更好的相关性。

图示：

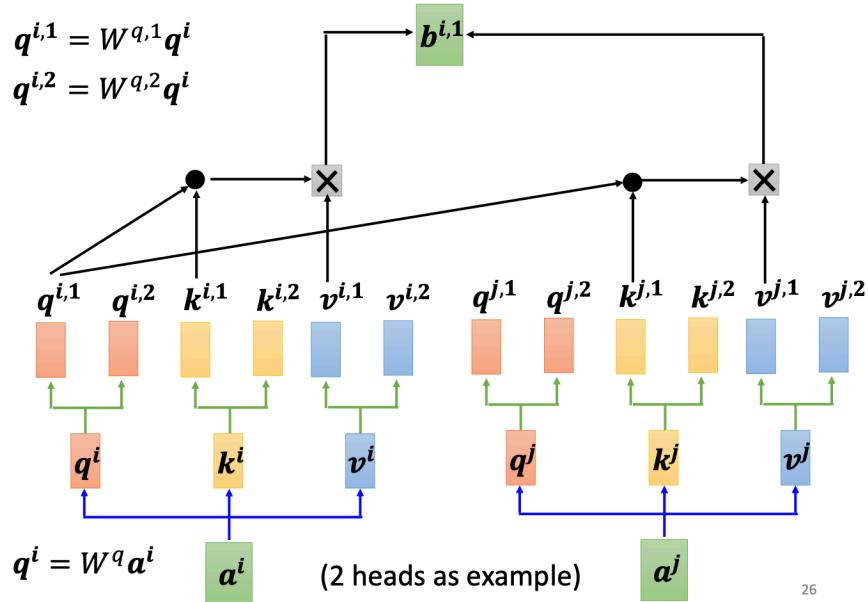


图3

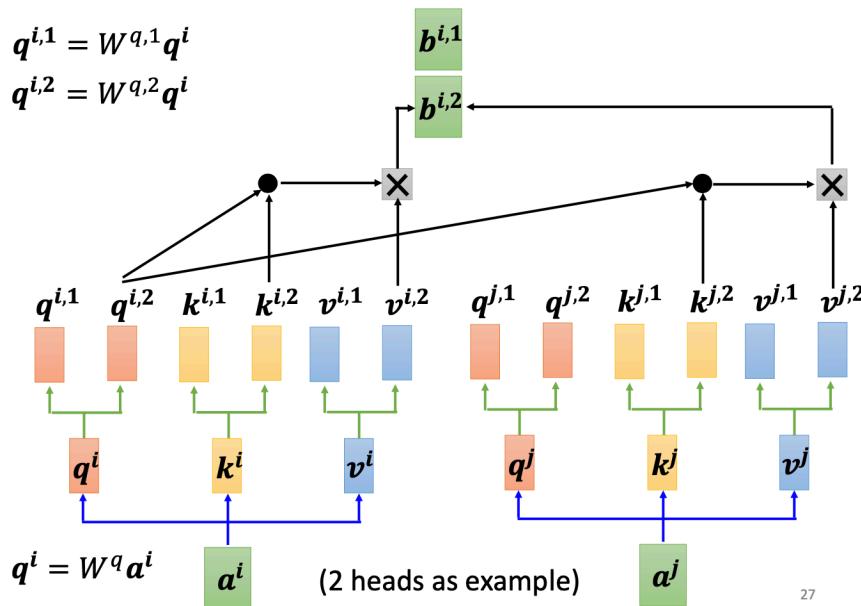


图4

$$b^i = W^o \begin{bmatrix} b^{i,1} \\ b^{i,2} \end{bmatrix}$$

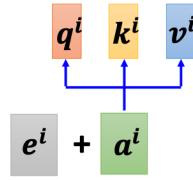
图5

说明：将q, k, v分别乘上2个矩阵，得到q1、q2, k1、k2, v1、v2，分别计算bi,1和bi,2，拼接乘以矩阵得到bi。

Positional Encoding

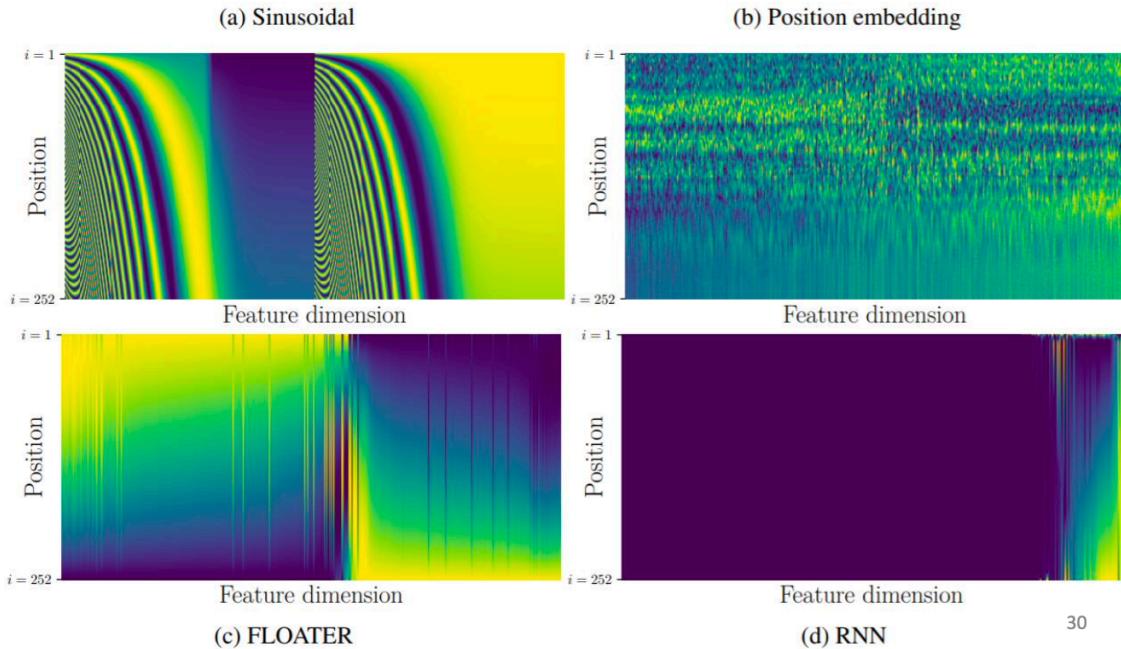
目的：上述self-attention结构丢失位置信息，需要人为添加位置信息。

方法：在输入\$a^i\$上加上代表位置的\$e^i\$从而给出位置信息，如图：



- 每个位置有一个独一无二的位置向量 (positional vector) e^i
- 可以为手工设定
- 可以从数据中学习得到

举例：

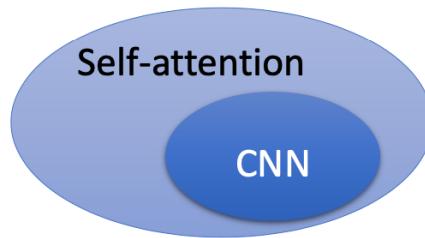


30

self-attention与CNN

结论：

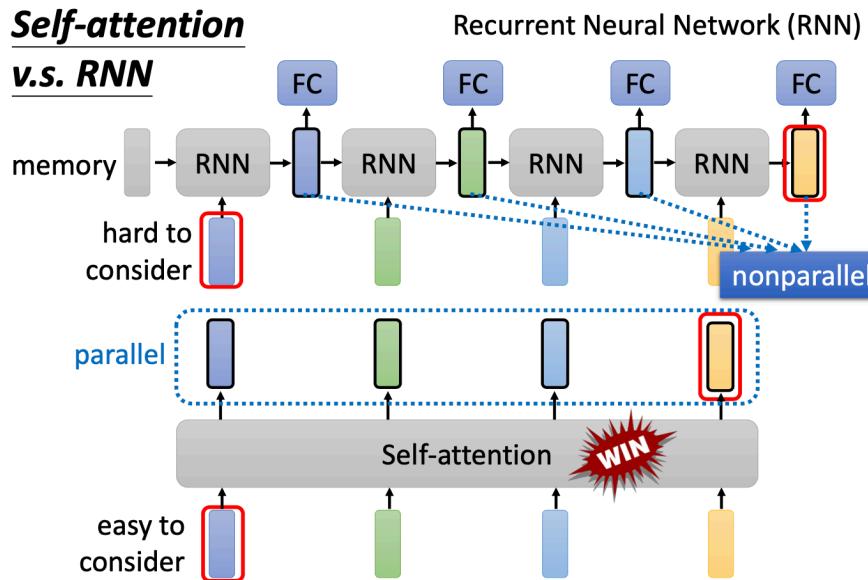
- CNN可以视为只关注小范围内的self-attention, CNN是简化的self-attention。
- self- attention可以视为关注范围可学习的CNN, self-attention是复杂化的CNN。
- self- attention在更多数据的情况下效果更佳, CNN在更少数据的情况下效果更佳。



self-attention与RNN

结论：

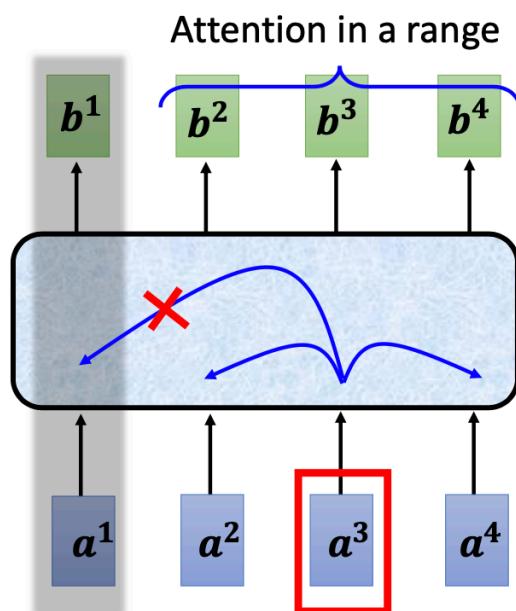
- 当首尾输入需要相互考虑时, RNN需要一个一个传播下去, self- attention直接一个作为q, 一个作为k, 计算相关性, 无需传播。
- RNN无法并行计算, self- attention可以并行计算。



self-attention应用

1. 在语音上的应用：

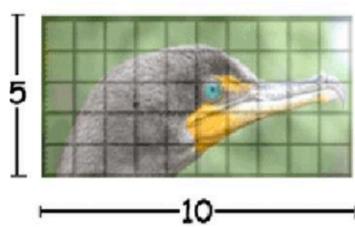
相关性矩阵的大小为输入向量个数的平方，但由于语音信号过长，计算量过大，因此语音处理时一般只关注小范围内的关系。



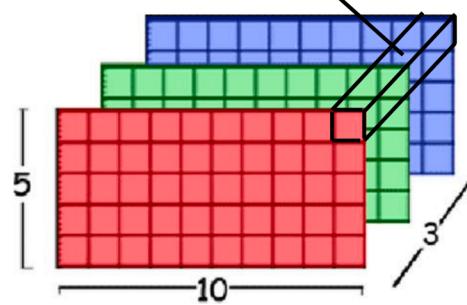
2. 在图像上的应用：

一般将图像RGB3个通道的像素组成一个vector，长乘以宽为向量个数。

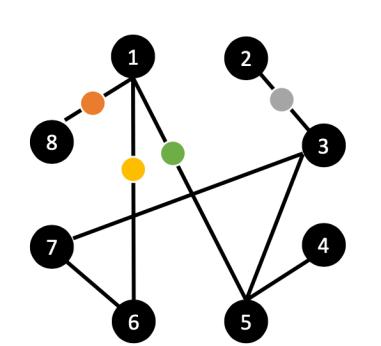
An image can also be considered as a **vector set**.



This is a vector.



3. 在图上的应用：



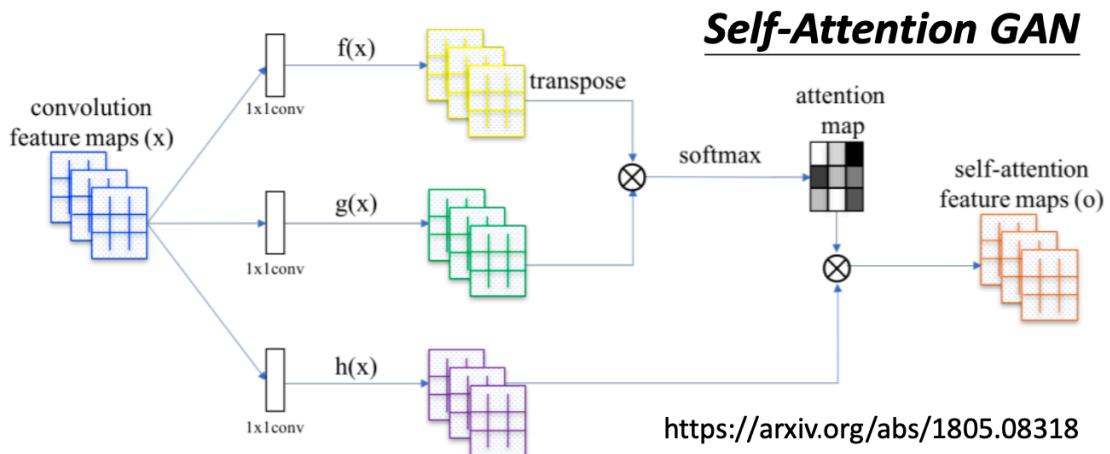
Consider edge: only attention to connected nodes

Attention Matrix

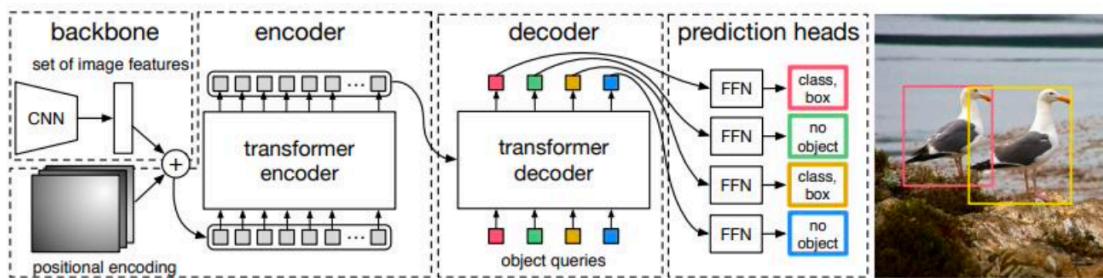
		1	2	3	4	5	6	7	8
1						●			
						●			
2			●						
3		●							
4									
5	●								
6	●								
7									
8	●							0	

This is one type of **Graph Neural Network (GNN)**.

4. 在GAN等其他方面的应用：



DEtection Transformer (DETR)



<https://arxiv.org/abs/2005.12872>

