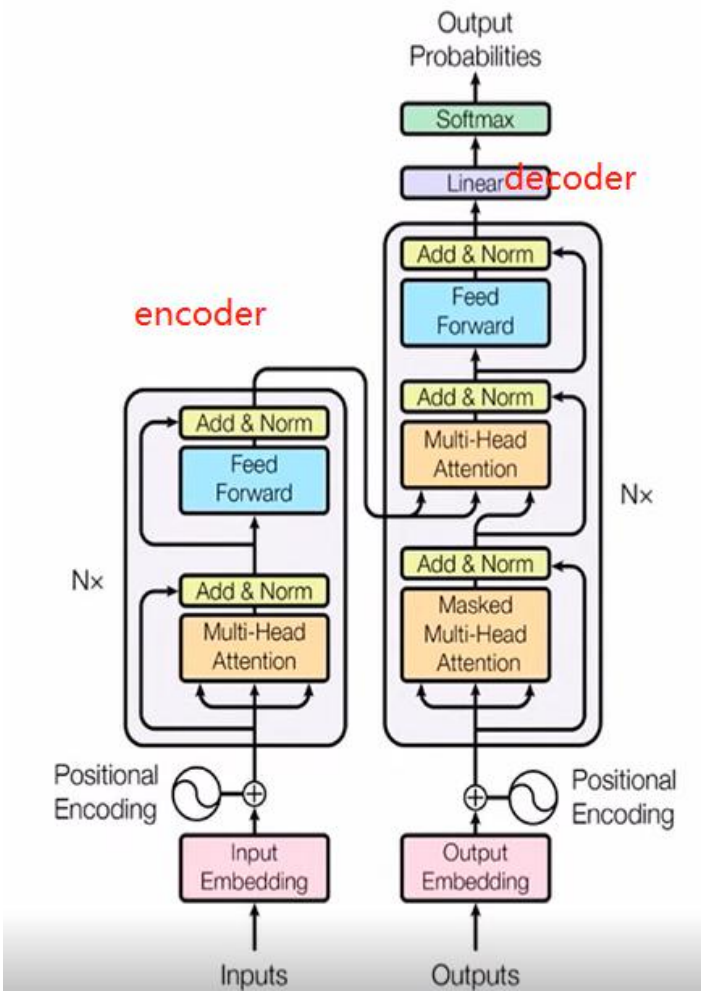
基于 transformer 特征提取器的改进

一、transformer 与 RNN 的区别

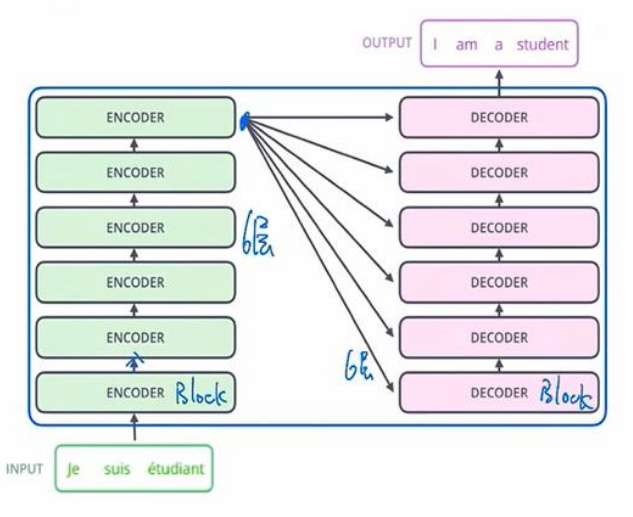
1.1 transformer 整体结构

Transformer 主要两个部分组成：encoder 与 decoder,encoder、decoderf 分别由 6 层 encode_block、decode_block 组成。



Transformer 整体结构图

Transformer 中 encoder 与 decoder 交互如下图表示：



Transformer 结构图

1.2 Block 结构

block 结构主要 self-attention 和 ffn 两部分组成。

1.2.1 自注意力

自注意力是对输入的一个序列中的每一token之间的互相注意力机制。与seq2seq中 attention 是两个句子之间的关系，与自注意力有一定的区别。Token 与 token 之间存在直接的关系，具体更直接的感受如下图表示：

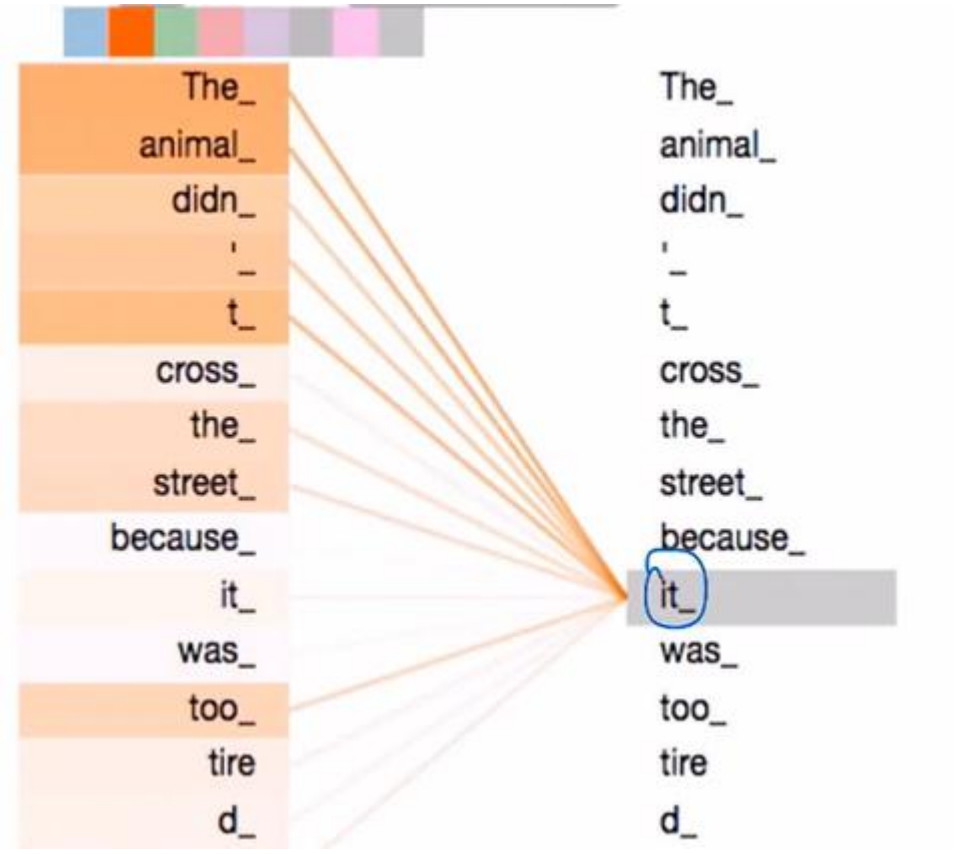


图 1.2.1

（1）self-attention

从代码上进行分析，输入的表征采用的是 embedding 特征，表示为 X，首先会

先将 x_i 分别与 W_Q、W_K、W_V 相乘，得到 q_i，k_i，v_i，如图 1.2.2 表示：

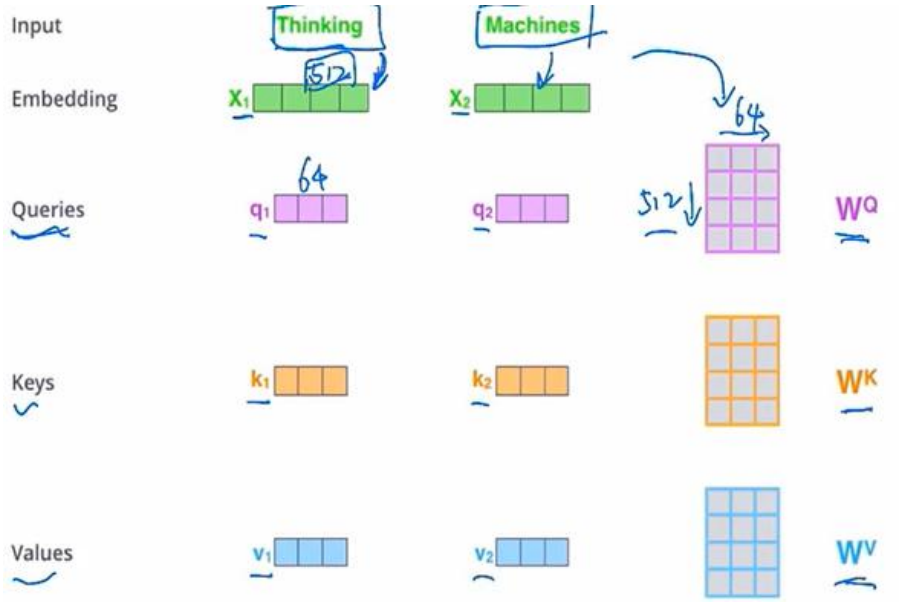
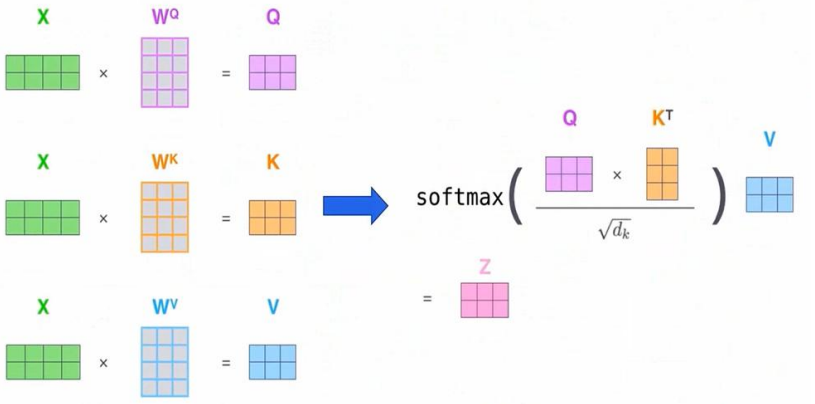


图 1.2.2

得到 q_i、k_i、v_i 之后，①先将 q_i 与 k_i 点乘，得到 score；②然后对 score/sqrt(d_k)进行缩放，最后进行 softmax 操作得到 attention_weight；③v_i 与 attention_weight 得到输出。



Self-attention 公式图

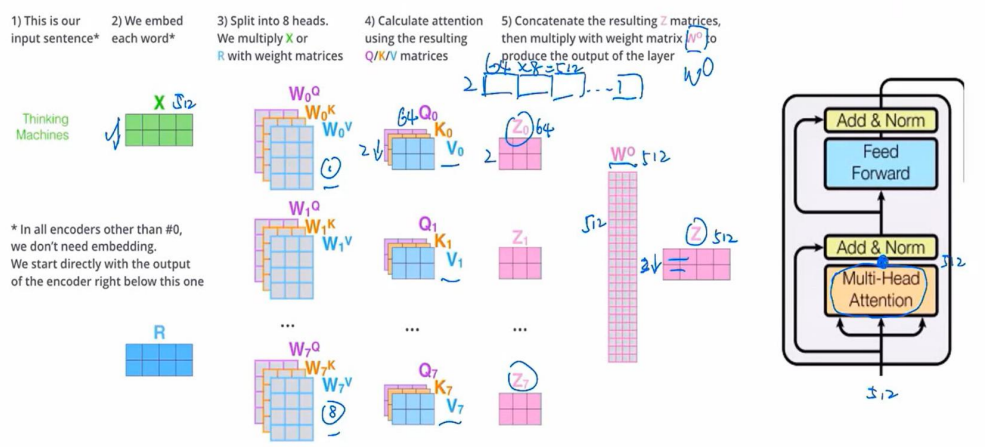
为什么要进行缩放 sqrt(d_k)?

答：缩放的原因：因为 softmax 在一定范围内梯度更新的效果更好，一旦超过这个范围就不好了，因此要缩放。

缩放 sqrt(d_k):参数初始化的时候，分布在 N(0,1)附近，q 与 k 相乘之后，score 分布在 N(0,d_k)，因此要进行 sqrt(d_k)的缩放。

1.2.2 多头自注意力(Multi-headed)

将 W 切分成 8 个头进行计算 attention 计算，计算结束后再进行 concat 还原。



Multi-head 结构图

1.2.3 FFN

FFN 结构：①全连接+relu/gelu 激活函数；②全连接。目的是对 concat 的 z 再次进行学习，由于 attention 都是线性计算，通过 FFN 可以进行非线性计算，是 transform_block 的关键。

Block 中用到残差结构的作用是什么？

答：残差结构是为解决梯度消失

为什么用 layerNorm 而不采用 BN？

答：BN 是对 batch_size 维度进行归一化，在 CV 中适用，因为图片上每一个像素都有特定的含义，然而在文本中，存在 pad 的情况，BN 不见得好；用 LN 的原因：对最后一维进行归一化（dim），每一个元素都有意义。

1.3 Position_embedding

公式如下：

$$PE(pos, 2i) = \sin(pos / 10000^{2i/dk})$$

$$PE(pos, 2i+1) = \cos(\quad)$$

1.4 RNN、CNN、Transformer 对比

答：rnn 的缺点：不能并行计算，文本过长时，无法长时记忆，抽取到的信息特征有限；

cnn 的缺点：只能学习较近的词关系，长距离学习能力欠缺，但可并行计算

cnn 改进：Text_cnn 可以采用多层的卷积，增强长距离学习能力，或者采用空洞卷积，增加感受野

①rnn、cnn、transformer 的特征提取能力、张距离捕获能力如下：

语义特征提取能力、任务综合特征抽取能力 → NMT

Model	DE→EN				DE→FR		
	PPL	2014	2017	Acc(%)	PPL	2012	Acc(%)
RNNS2S	5.7	29.1	30.1	84.0	7.06	16.4	72.2
ConvS2S	6.3	29.1	30.4	82.3	7.93	16.8	72.7
Transformer	**4.3**	**32.7**	33.7	**90.3**	**4.9**	**18.7**	**76.7**
uedin-wmt17	–	–	**35.1**	87.9	–	–	–
TransRNN	5.2	30.5	31.9	86.1	6.3	17.6	74.2

长距离特征捕获能力 →

Model	2014	2017	PPL	Acc(%)
RNNS2S	23.3	25.1	6.1	95.1
ConvS2S	23.9	25.2	7.0	84.9
Transformer	**26.7**	**27.5**	**4.5**	**97.1**
RNN-bideep	24.7	26.1	5.7	96.3

Why Self-Attention?
A Targeted Evaluation of Neural Machine Translation Architectures

性能比较图

②rnn、cnn、transformer 的计算能力与运行效率如下：

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

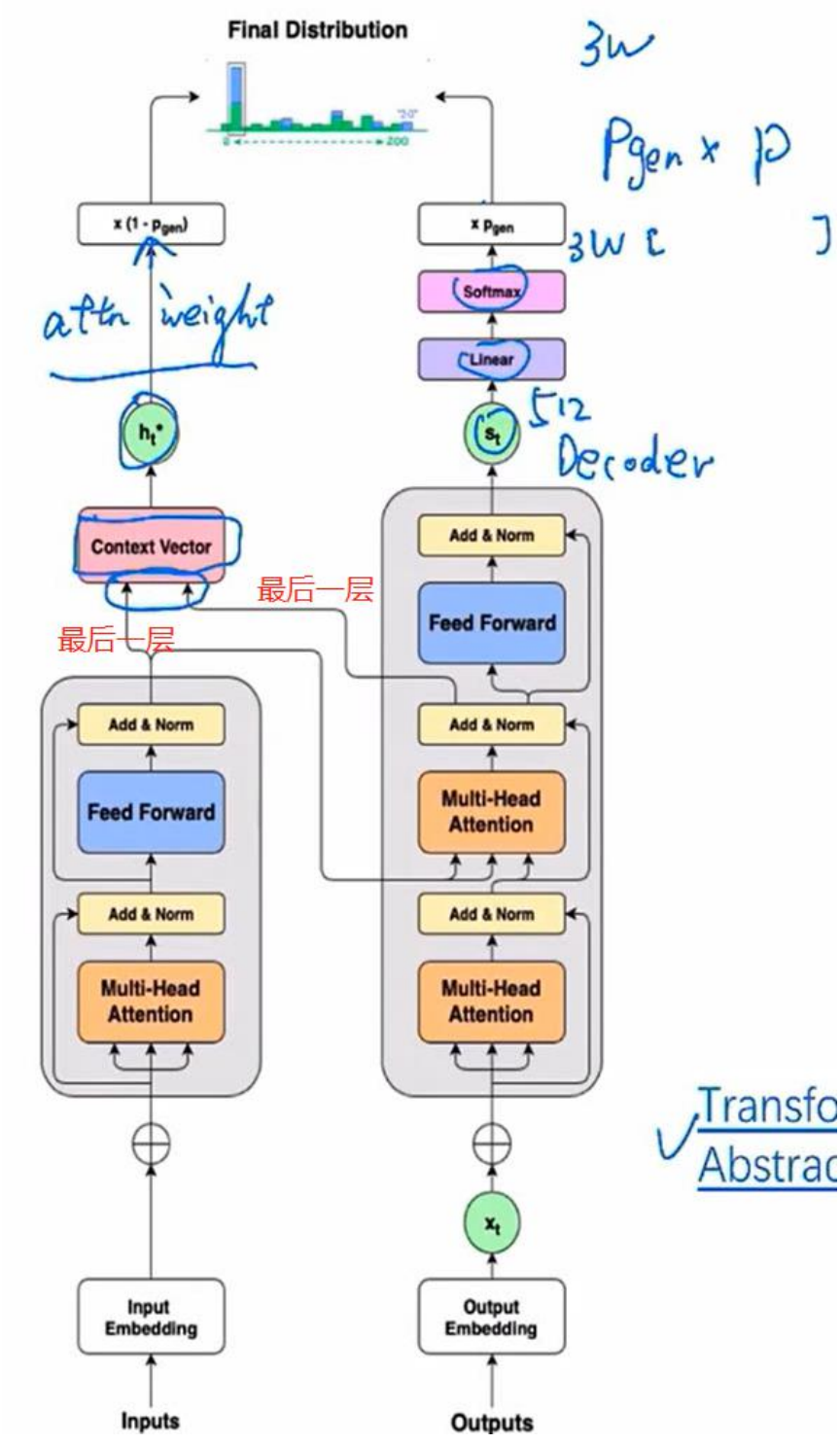
Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

效率比较图

二、Transformer 模型改进

2.1 PGN-transformer

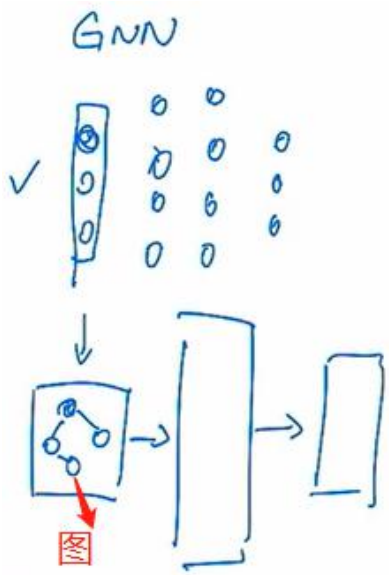
Transformer 的 PGN 的实现，与 seq2seq 大致类似，涉及两个概率 p_vocab 与 attention_weight 这个概率；因此，需要根据 encode 与 decode 的输出求出 attention_weight，p_gen 放在 atention 计算去求解，剩下的按公式计算即可。



Pgn-transformer 结构图

三、图神经网络在摘要中的应用

图神经网络与 CNN 的区别，每一层不是一个个离散的神经元，而是一个图，常用的图神经网络：GCN、GAT。



图神经网络图

图网络应用于摘要问题：用单词训练 word 的向量作为 word Node，将句子通过 cnn、bilstm 得出句子向量作为 sentence Node,TF-IDF 的权重值作为连接的边，构成最终的 graph layer。论文中提出添加 docment 节点，用于训练多篇文档之间的摘要，具体结构如图 3.1。

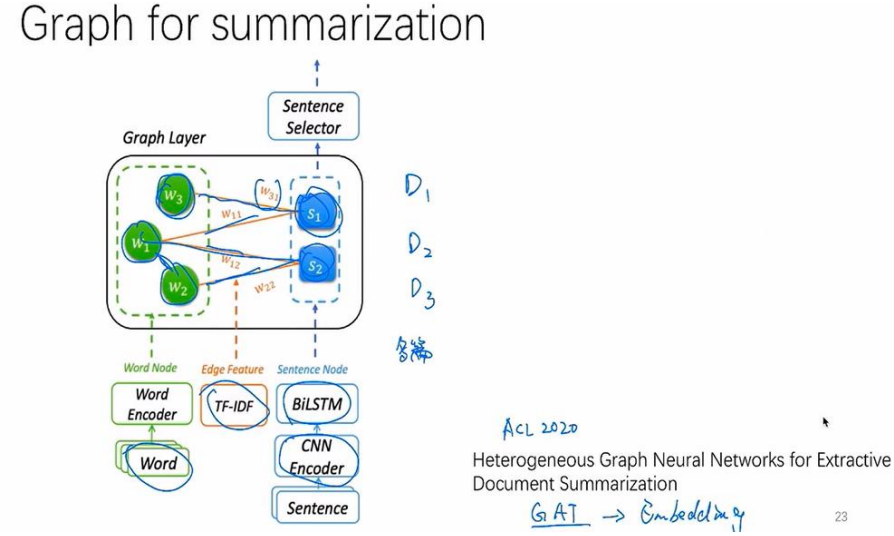


图 3.1

近 2 年摘要研究的论文：

- Modeling Global and Local Node Contexts for Text Generation from Knowledge Graphs
- Text Generation from Knowledge Graphs with Graph Transformers
- Structured Neural Summarization
- Discourse-Aware Neural Extractive Model for Text Summarization
- Graph-based Neural Multi-Document Summarization