Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心

Transformer中的 Multi-Head Attention

——解读《attention is all you need》

王睿怡 硕士

2018年12月9日

内容提要

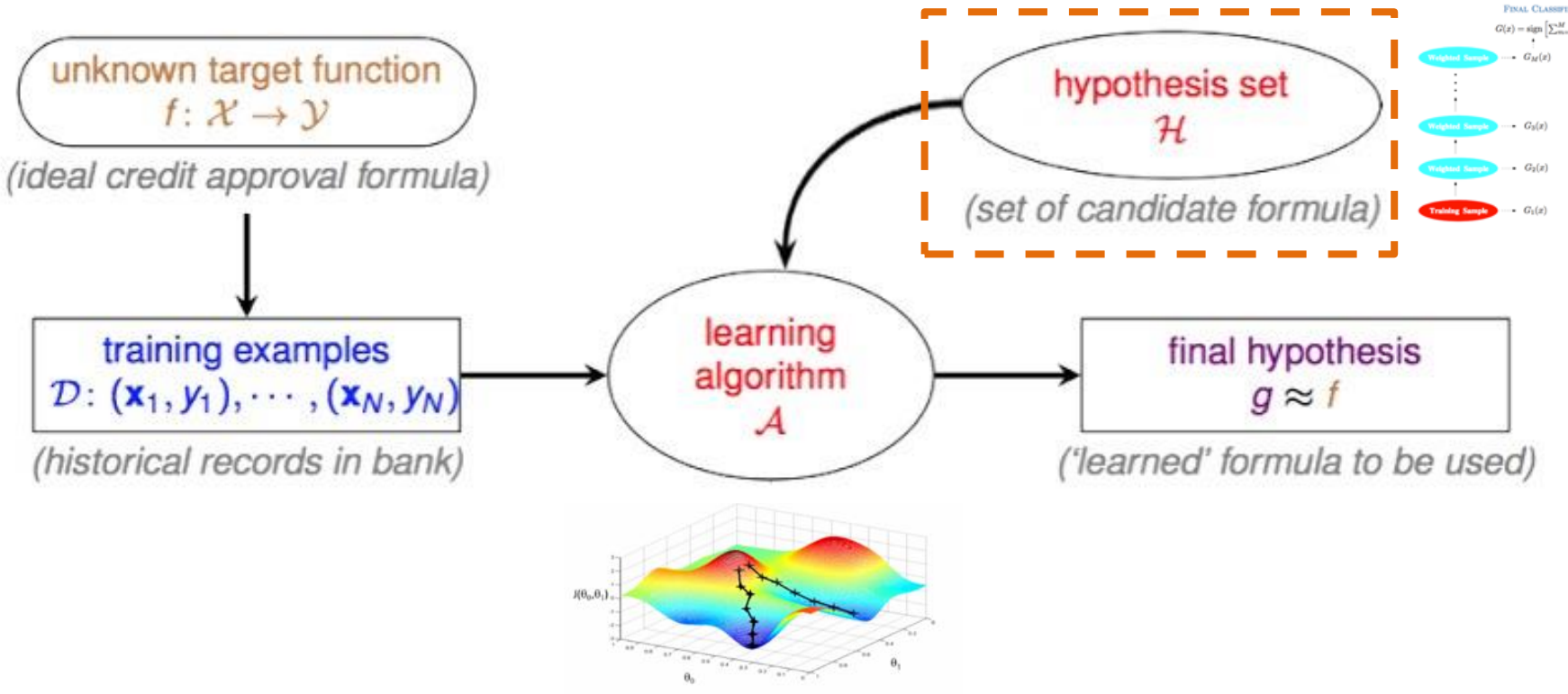
- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 应用总结
- 参考文献

背景简介

- 预期收获
  - 1. 复习seq2seq框架、注意力机制基本思想
  - 2. 理解Multi-Head Attention的算法原理
  - 3. 了解Transformer的算法流程
  - 4. 了解Multi-Head Attention的应用

背景介绍

- 机器学习基础架构[1]

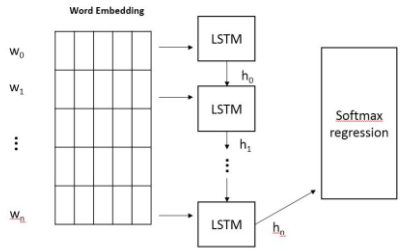


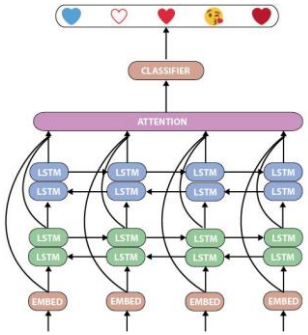
[1] 林轩田，机器学习基石

背景简介

- **情感分类**

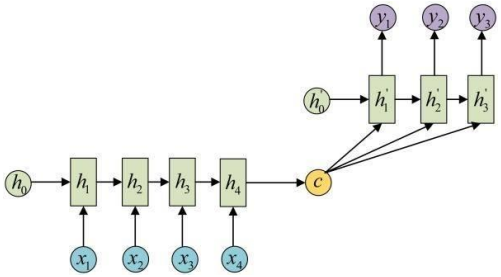
次新股低迷黄山胶囊跌停 ✚ Attention层 = 次新股低迷黄山胶囊跌停

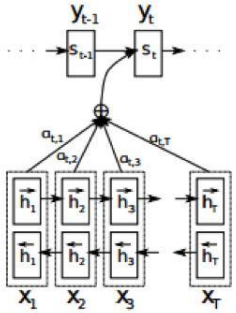




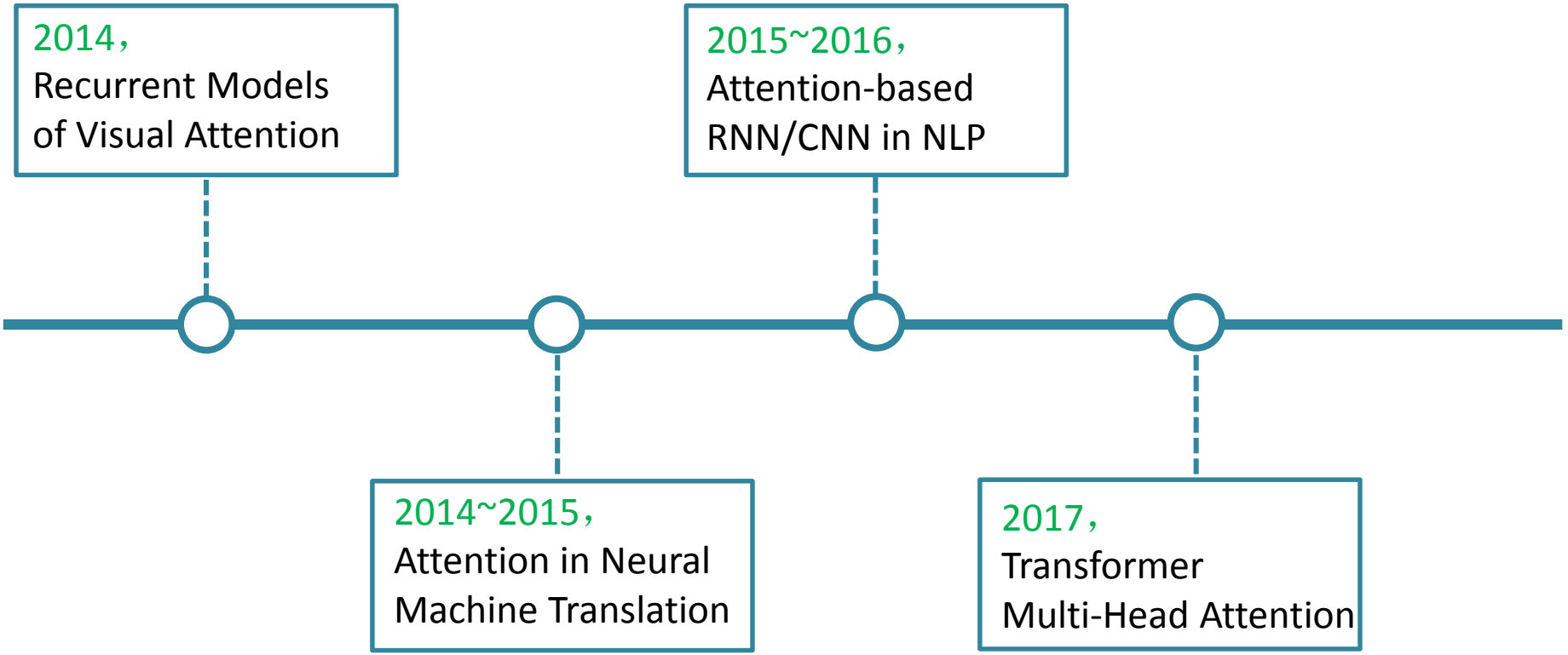
- **机器翻译**

我喜欢运动 ⟹ I ✚ Attention层 = 我喜欢运动 ⟹ I





背景介绍

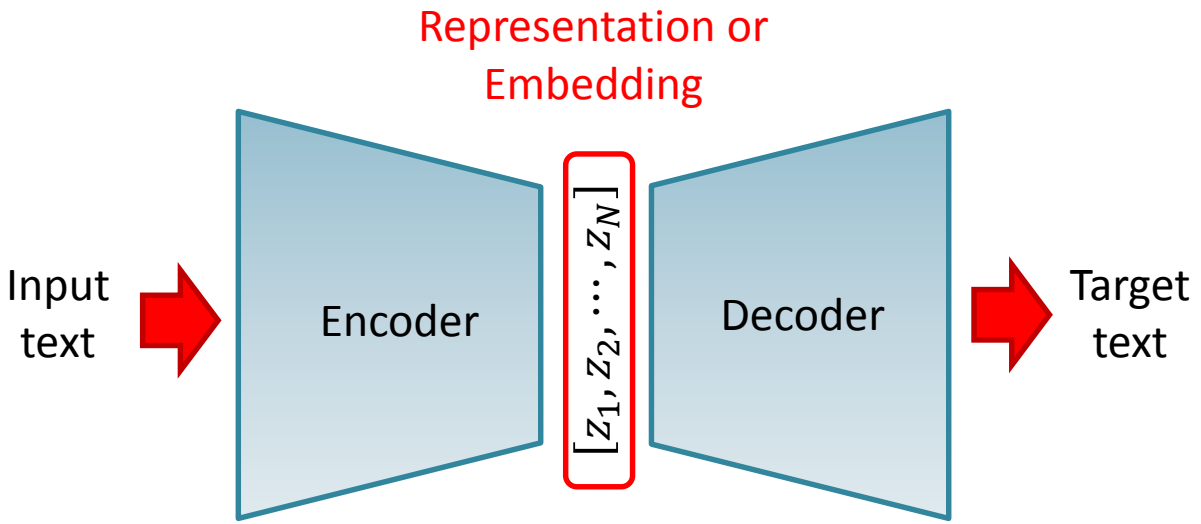

2014，
Recurrent Models
of Visual Attention
2014~2015，
Attention in Neural
Machine Translation
2015~2016，
Attention-based
RNN/CNN in NLP
2017，
Transformer
Multi-Head Attention

背景简介

Seq2Seq

Encoder-Decoder Framework

- Encoder-Decoder框架
  - 编码器: 从单词序列到句子表示
  - 解码器: 从句子表示转化为单词序列分布

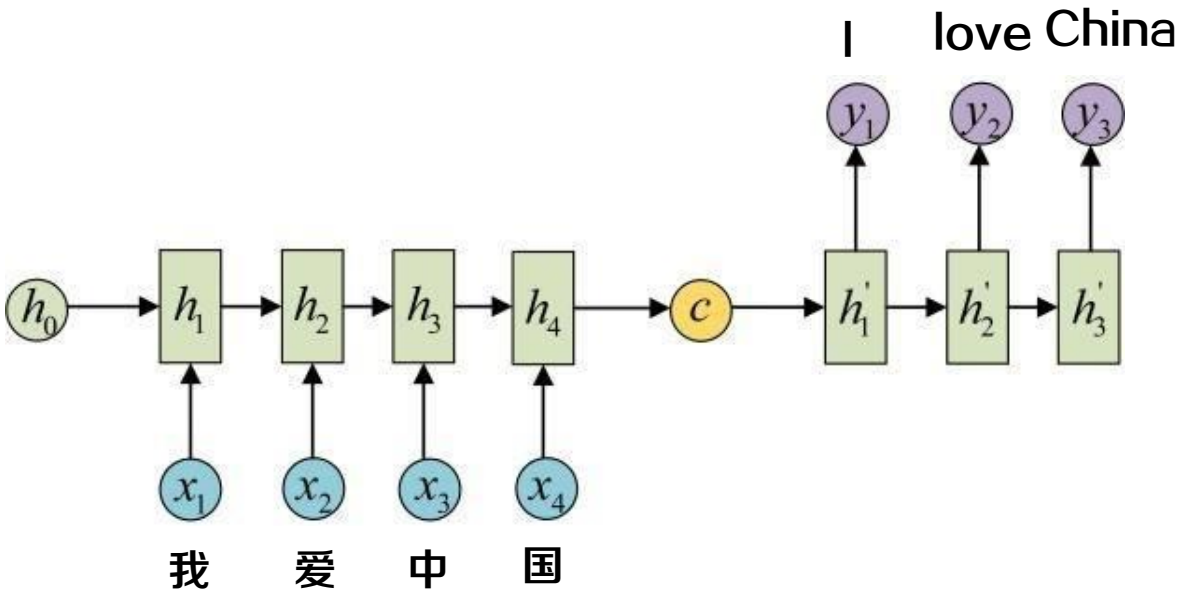


背景简介

Seq2Seq

Encoder-Decoder Framework

RNN/LSTM/GRU



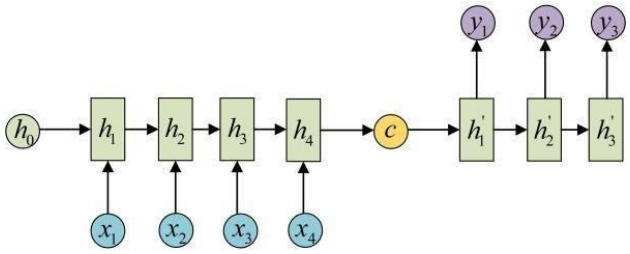
- 方向-单向或双向
- 深度-单层或多层
- 类型-RNN、LSTM或GRU

背景简介

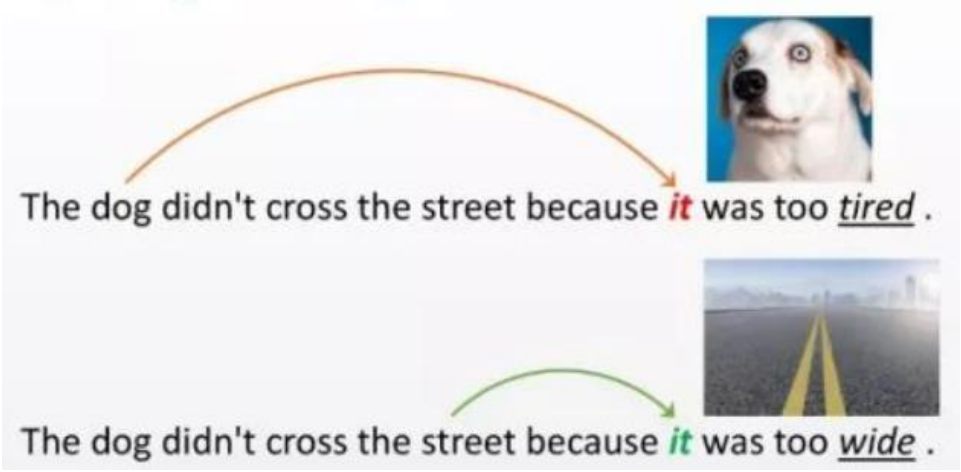
Seq2Seq

Encoder-Decoder Framework

RNN/LSTM/GRU



- RNN网络的缺点：
  - RNN的序列特性导致其难以并行化
    - GNMT需要96块GPU训练一周
  - 长距离和层级化的依赖关系难以建立
    - 句法信息，指代信息



背景简介

Seq2Seq

Encoder-Decoder Framework

RNN/LSTM/GRU

Transformer

- **Transformer for Neural Machine Translation**
  - 高度并行化网络
    - 抛弃Recurrent结构，训练速度提升数倍
  - 层级化信息的捕捉
    - 构建多层神经网络
  - 指代信息等丰富上下文信息的引入

Attention, Self-attention

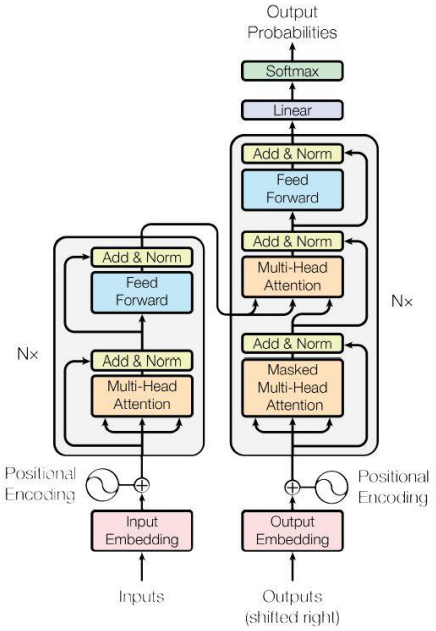
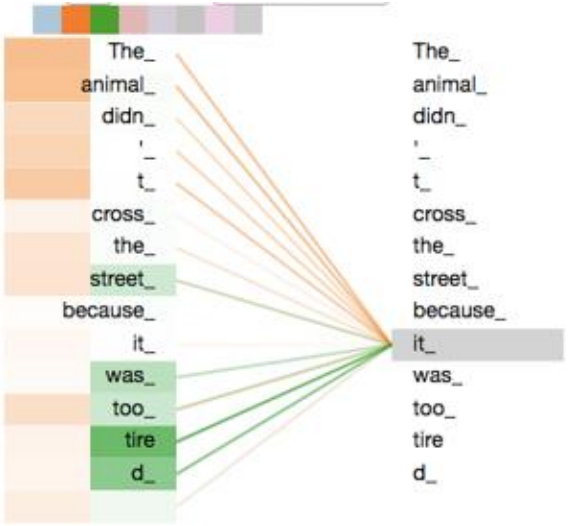


Figure 1: The Transformer - model architecture.



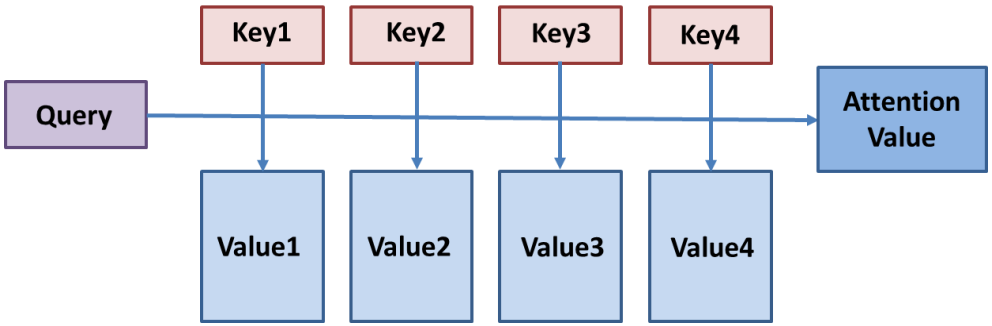
比如在机器翻译中，只有理解了it指代谁，才能更好地翻译之后的句子

Transformer中的Multi-Head Attention

基本概念

基本概念

- Attention函数的本质可以被描述为一个查询（query）到一系列（键key-值value）对的映射。
  - 语言学角度：描述词与词之间的关联关系
  - 机器学习角度：神经网络隐层之间的相似度表示
  - 一个基于内容的查询的过程



$$Attention(Query, Source) = \sum_{i=1}^{l_s} Similarylity(Query, Key_i) * Value_i$$

在阅读理解任务中，Q是篇章的词向量序列，取K=V为问题的词向量序列，那么输出就是所谓的Aligned Question Embedding(对齐问题的向量)。
在机器翻译任务中，Q是解码器中的隐层向量，K和V为原始文本的词向量。

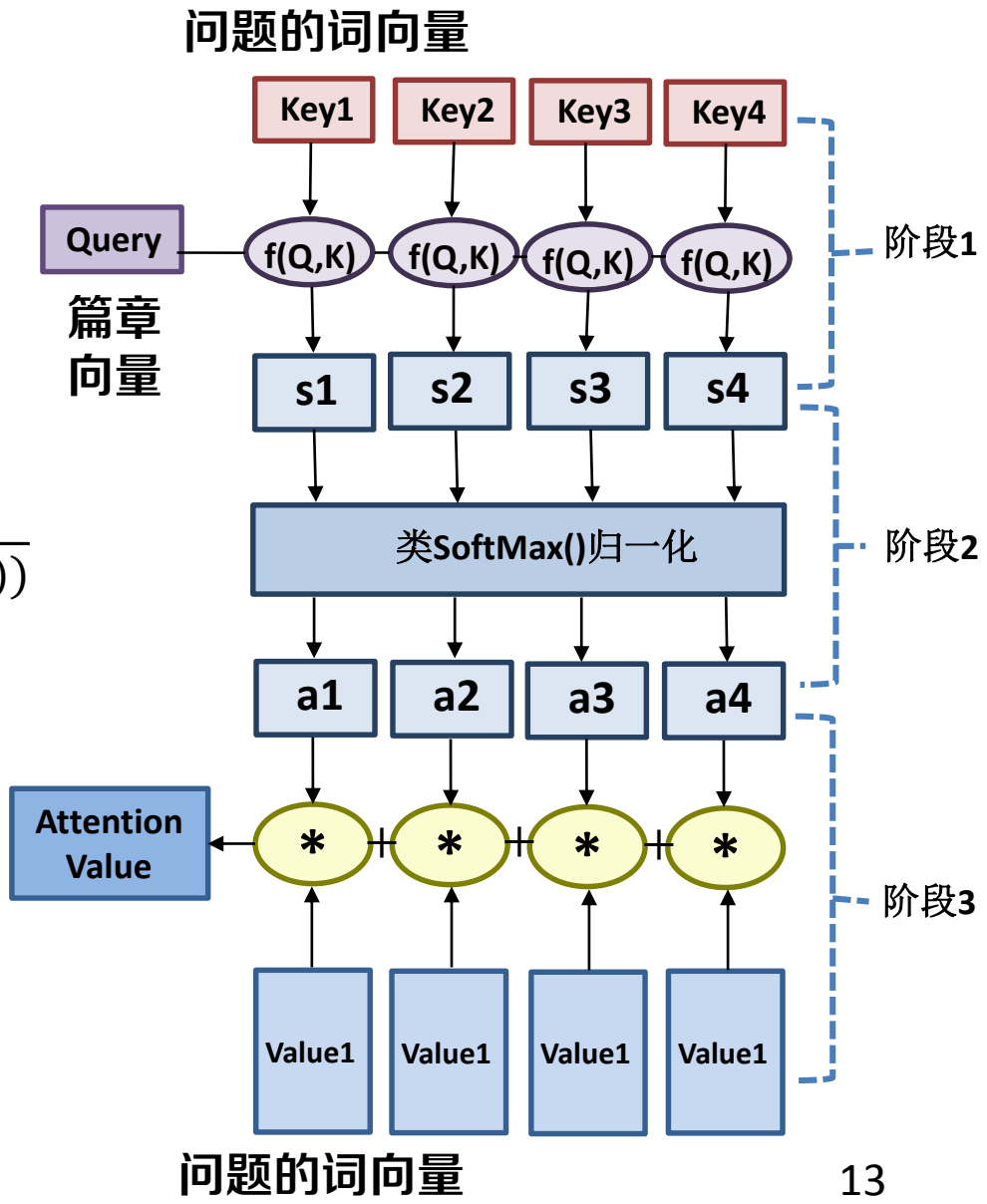
基本概念

$$\mathrm{f}(Q,K) = \begin{cases} Q^T K & \text{dot} \\ Q^T W_a K_i & \text{general} \\ W_a[Q;K_i] & \text{concat} \\ v_a^T \tanh(W_a Q + U_a K_i) & \text{preceptron} \end{cases}$$

$$a_i = softmax\big(f(Q,K_i)\big) = \frac{\exp(f(Q,K_i))}{\sum_j \exp(f(Q,K_i))}$$

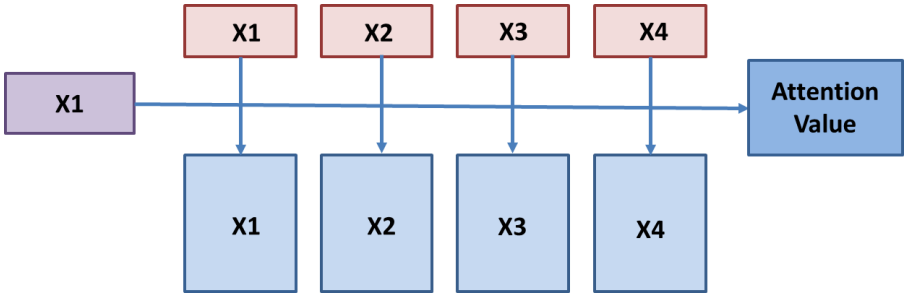
$$Attention(Q,K,V) = \sum_i a_i V_i$$

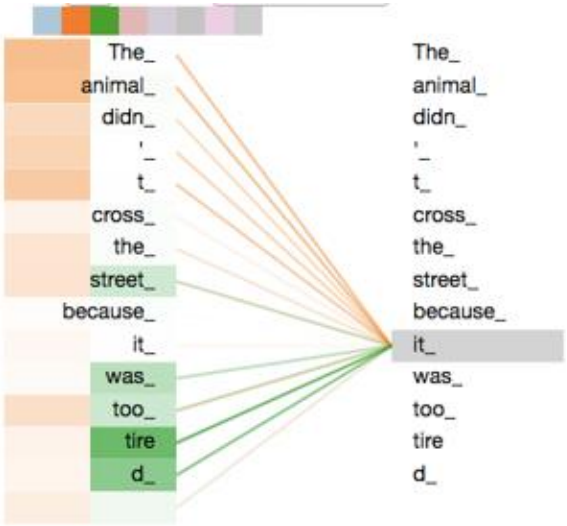
目前在NLP研究中，K和V常常都是同一个，即K=V。



基本概念

- Self-attention，即“自注意力”：
  - 在序列内部做Attention，寻找序列内部的联系
  - 句子内词与词之间的关联关系
  - 可以用于解决指代消解等问题





$$Attention(Query, Source) = Attention(X, X, X)$$

$$K = V = Q = X$$

Transformer中的Multi-Head Attention

算法原理

算法原理

T	提出Transformer模型完成机器翻译任务
I	原始句子
P	1.初始化词向量 2.对原始句子进行编码，构建句子矩阵 3.对句子矩阵进行解码，依次预测下一时刻输出的单词
O	翻译好的句子

P	RNN无法并行计算，长距离和层级化的依赖关系难以建立
C	计算资源充足
D	模型设计思路，如何提取包含丰富信息的向量
L	CCF A类（NIPS 2017）

算法原理

- Transformer

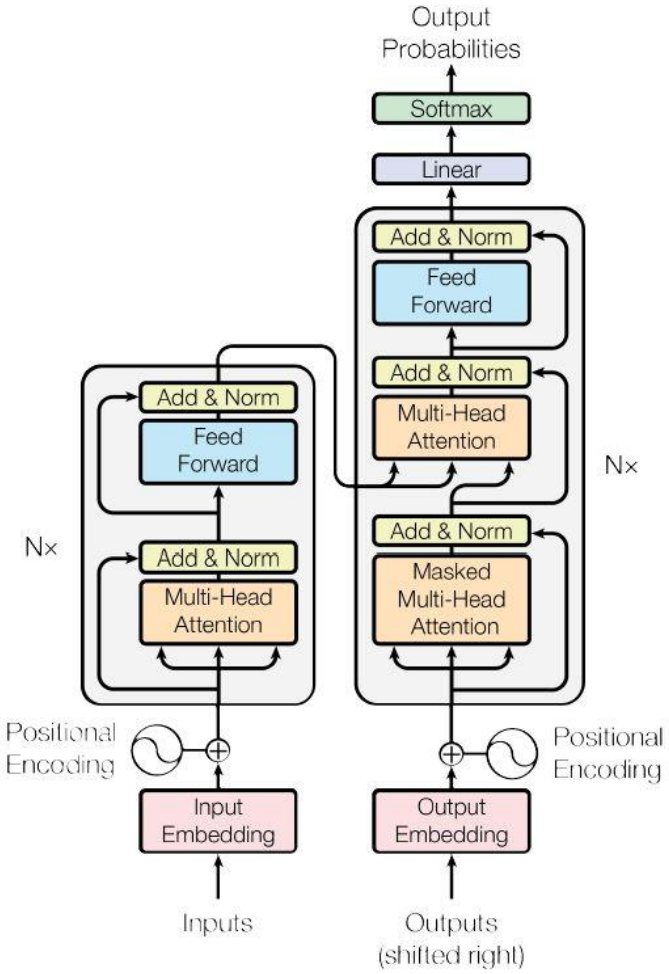
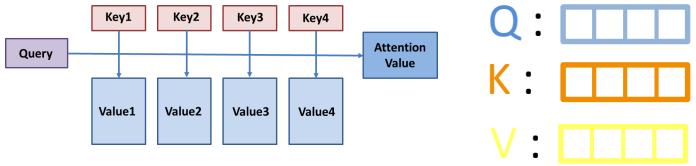
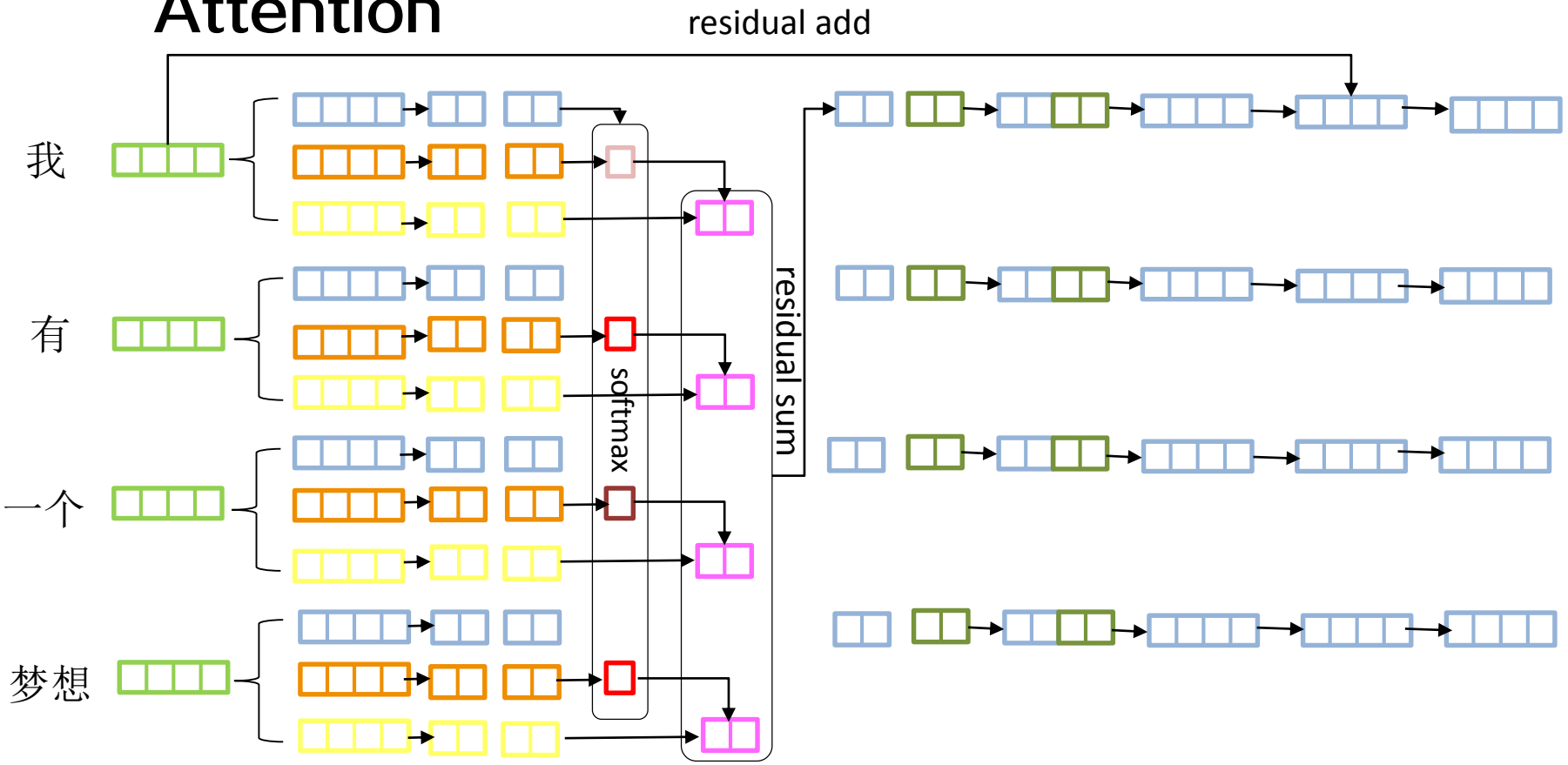


Figure 1: The Transformer - model architecture.

算法原理

- Multi-Head Attention

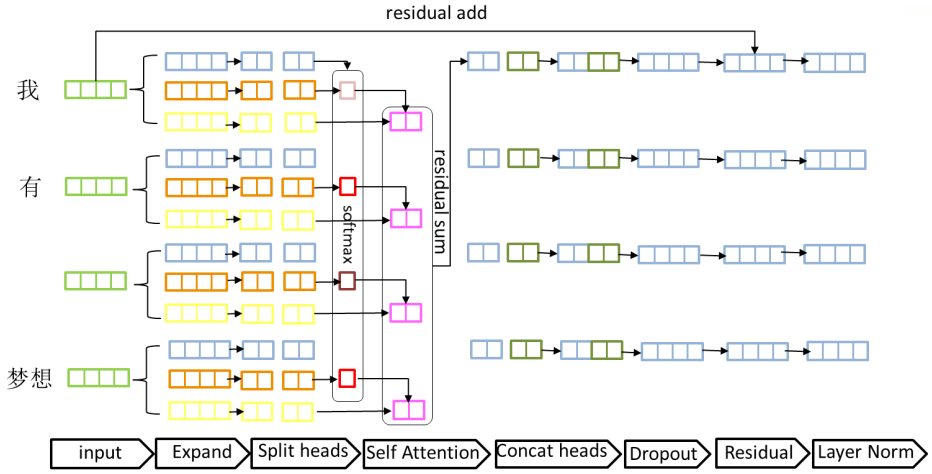




input → Expand → Split heads → Self Attention → Concat heads → Dropout → Residual → Layer Norm

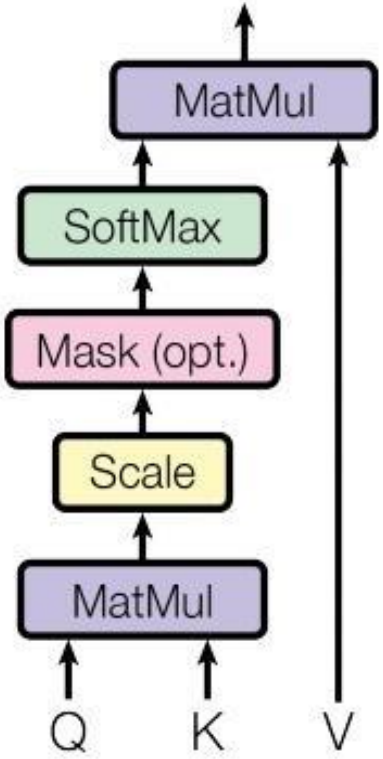
算法原理

- Scaled Dot-Product Attention



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

多除一个d_k（为K的维度）起到调节作用，使得内积不至于太大

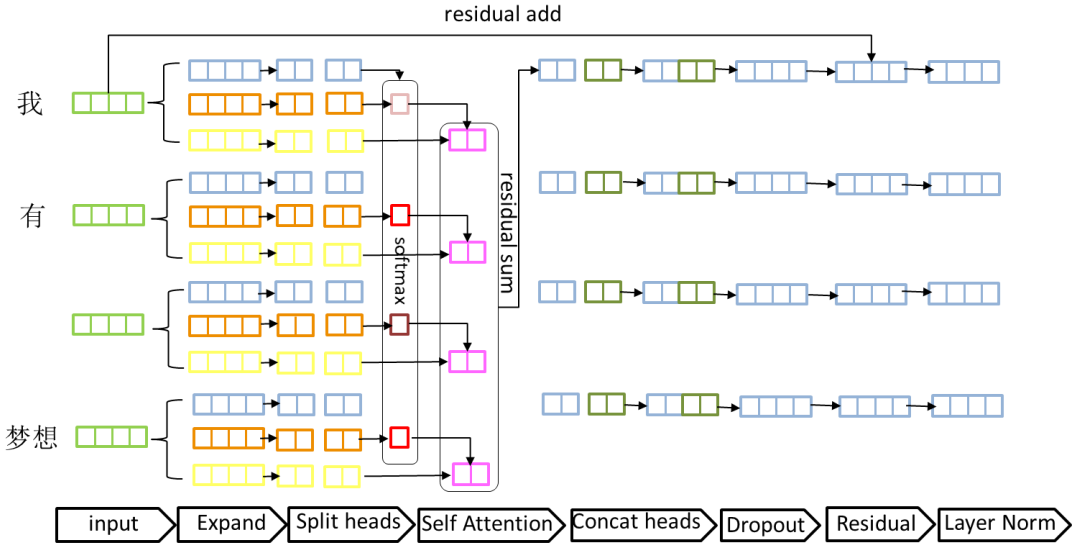


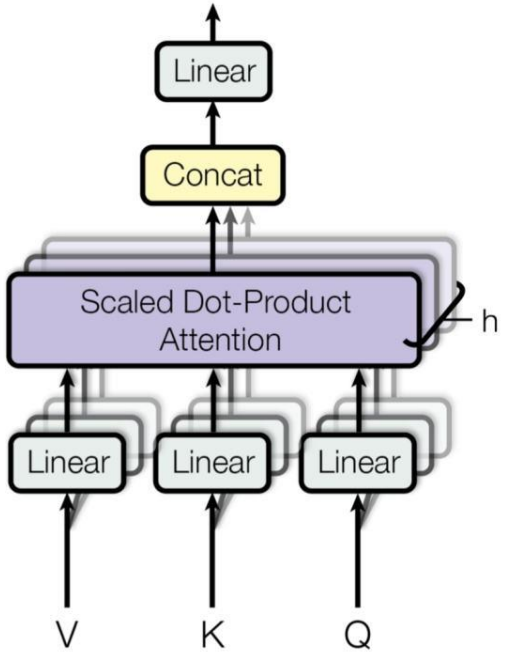
算法原理

- Multi-Head Attention
  - 所谓“多头”（Multi-Head），就是只多做几次同样的事情（参数不共享），然后把结果拼接
  - 好处是可以允许模型在不同的表示子空间里学习到相关的信息
  - 类似于CNN中多个卷积核的思想

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

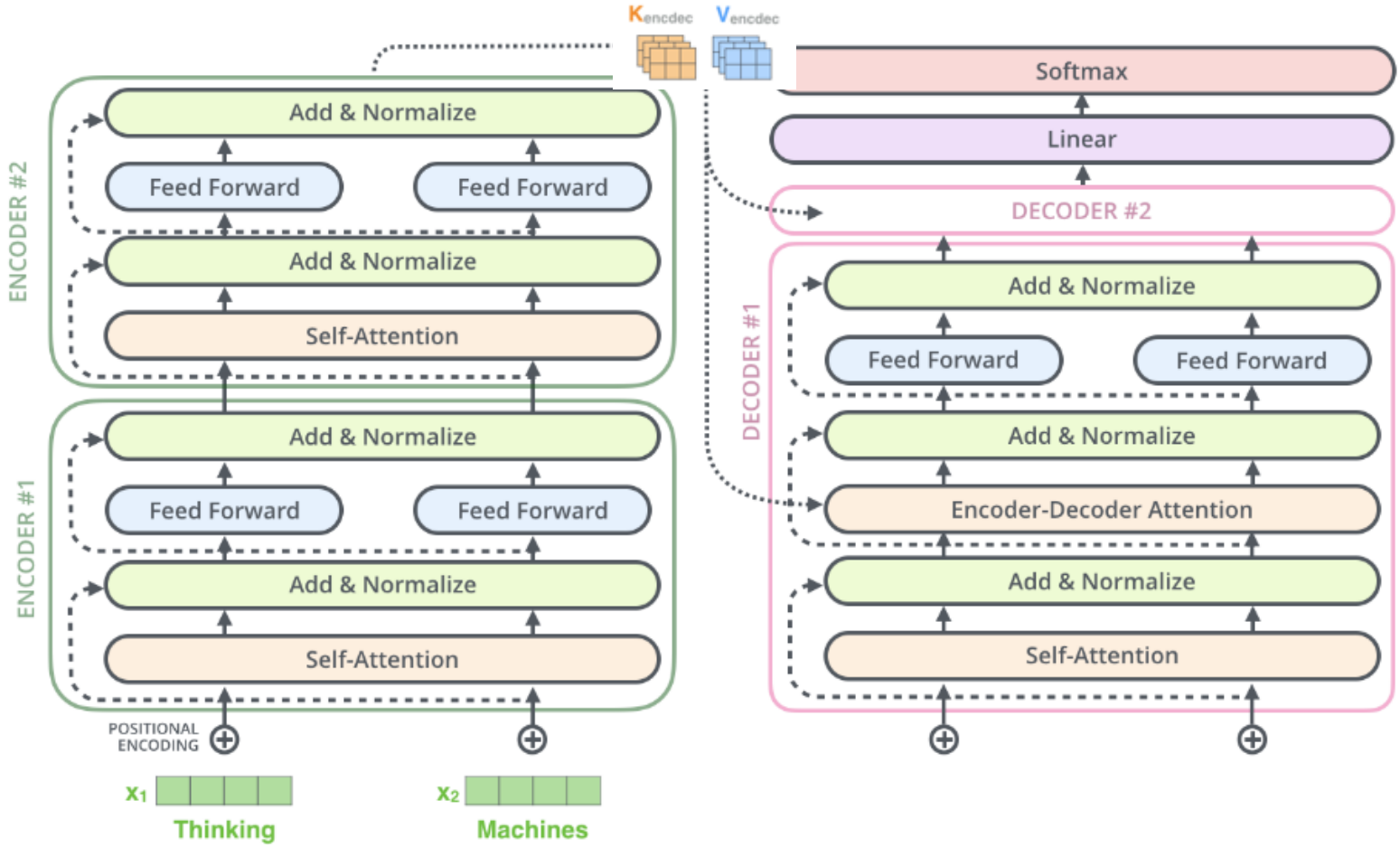
$$MultiHead_i(Q,K,V) = Concat(head_1, \cdots, head_h)W^O$$





算法原理

模型共包含三个attention成分：
1. encoder的self-attention、2. decoder的self-attention、3.连接encoder和decoder的attention。



算法原理

- 输入：一个句子$z = (z_1, z_2, \cdots, z_n)$,它是原始句子$x = (x_1, x_2, \cdots, x_n)$的Embedding,其中n是句子长度。
- 输出：翻译好的句子$(x_1, x_2, \cdots, x_n)$的Embedding，其中n是句子长度。

Encoder

- 输入$z \in R^{n \times d_{model}}$
- 输出大小不变
- Position Encoding
- 6个Block
  - Multi-Head Attention
  - Position-wise Feed Forward
  - Residual connection
    - LayerNorm(x+Sublayer(x))
    - 引入了残差，尽可能保留原始输入x的信息
  - $d_{model} = 512$

Decoder

- Positional Encoding
- 6个Block
  - Multi-Head Attention(with mask)
    - 采用0-1 mask消除右侧 单词对当前单词attention的影响
  - Multi-Head Attention(with encoder)
    - 使用Encoder的输出作为一部分输入
  - Position-wise Feed Forward
  - Residual connection

Transformer中的Multi-Head Attention

优劣分析

优劣分析

- Self-attetion的优势
  - 时间复杂度较小
  - 可以很好并行计算
  - 可以捕获长距离依赖关系，一步到位捕捉全局关系

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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

优劣分析

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Transformer中的Multi-Head Attention

应用总结

应用总结

- Multi-Head Attention的应用
  - 机器翻译
  - BERT：Pre-training of Deep Bidirectional Transformers for Language Understanding
    - feature-based: 把表征作为feature提供给下游任务
    - fine-tuning: 微调预训练的参数

Model	Samples	Dataset	Accuracy
CNN	1000	10 Category Clean	60.50%
CNN+Glove	1000	10 Category Clean	58.50%
ULMFiT	1000	10 Category Clean	76.50%
BERT	**1000**	**10 Category Clean**	**80.50%**
CNN	6700	25 Category Noisy	56.26%
CNN+Glove	6700	25 Category Noisy	59.20%
ULMFiT	6700	25 Category Noisy	67.00%
BERT	**6700**	**25 Category Noisy**	**71.50%**
CNN	6700	25 Category Clean	62.74%
CNN+Glove	6700	25 Category Clean	63.99%
ULMFiT	6700	25 Category Clean	69.00%
BERT	**6700**	**25 Category Clean**	**81.44%**
CNN	12000	25 Category Noisy	61.56%
CNN+Glove	12000	25 Category Noisy	62.96%
ULMFiT	12000	25 Category Noisy	65.40%
BERT	**12000**	**25 Category Noisy**	**69.30%**
CNN	12000	25 Category Clean	73.50%
CNN+Glove	12000	25 Category Clean	72.24%
ULMFiT	12000	25 Category Clean	78.04%
BERT	**12000**	**25 Category Clean**	**81.03%**

BERT在极少的数据集上表现非常出色，
解决NLP中数据标注难的问题

BERT开源的多个版本的模型：

- `BERT-Base, Uncased` : 12-layer, 768-hidden, 12-heads, 110M parameters
- `BERT-Large, Uncased` : 24-layer, 1024-hidden, 16-heads, 340M parameters
- `BERT-Base, Cased` : 12-layer, 768-hidden, 12-heads , 110M parameters
- `BERT-Large, Cased` : 24-layer, 1024-hidden, 16-heads, 340M parameters (Not available yet. Needs to be re-generated).
- `BERT-Base, Multilingual` : 102 languages, 12-layer, 768-hidden, 12-heads, 110M parameters
- `BERT-Base, Chinese` : Chinese Simplified and Traditional, 12-layer, 768-hidden, 12-heads, 110M parameters

BERT提出之后，作为一个Word2Vec的替代者，其在NLP领域的11个方向刷新了最高成绩，可以说是近年来自残差网络最优突破性的一项技术

应用总结

- Self attention
  - Text Respresentation
  - Reading comprehension
  - Natural Language Inference
  - Relation Extraction
  - Semantic Role Labelling

参考文献

- Attention is all you need (2017)
- BERT Pre-training of Deep Bidirectional Transformers for Language Understanding(2018)
- The Illustrated Transformer: https://jalammar.github.io/illustrated-transformer/
- Transformer新型神经网络在机器翻译中的应用:https://www.tinymind.cn/articles/3778
- 追一科技-Google+BERT模型解析及实验探索: http://www.chinahadoop.cn/course/1253
- 论文笔记：Attention is all you need: https://www.jianshu.com/p/3f2d4bc126e6

谢谢！

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。