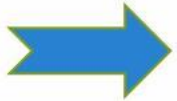
NLP中的表示学习

本次课程概要

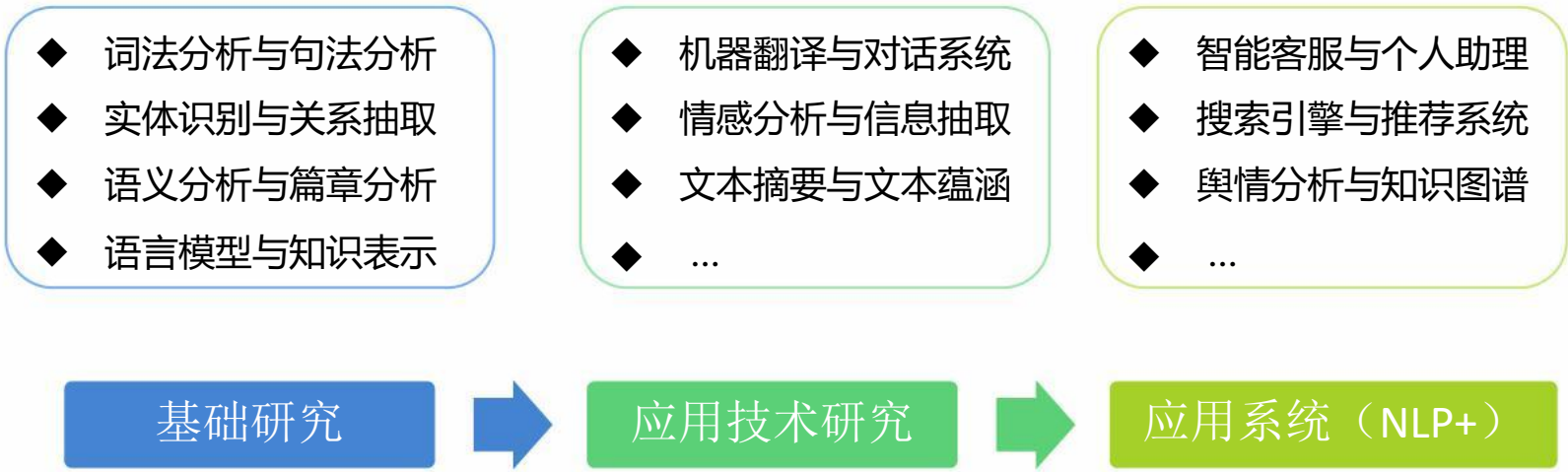
四个方面



- 表示研究内容与发展历程
- 语言的表示问题
- 词向量模型Skip-gram
- 词序列模型Transformer

自然语言处理(NLP)的研究内容

- 什么是自然语言？通常理解为人类语言，不同于人工语言（比如程序语言Python等）
- 自然语言处理是让机器理解、认知、感知、甚至表达自然语言的一门学科。
- 自然语言处理是人工智能领域的三大研究分支之一（子学科）。



自然语言处理(NLP)是认知智能的核心

NLP is the core of cognitive intelligence

“深度学习的下一个大的进展应该是让神经网络真正理解文档的内容”

深度网络之父：Geoffrey Hinton

“如果给我10亿美金，我会用这10亿美金建造一个NASA级别的自然语言处理研究项目。”

机器学习专家、美国双院院士
Michael I. Jordan

“深度学习的下一个前沿课题是自然语言理解。”

Facebook人工智能负责人：Yann LeCun

“下一个十年，懂语言者得天下”

微软全球执行副总裁：沈向洋

图片引用自:【刘挺-中文信息处理前沿技术进展】

自然语言处理发展历程

- 技术路线（由浅入深）

认知智能
能理解会思考

2018 -
自监督学习，知识图谱

- 自监督学习
- 预训练
- 知识图谱
- 逻辑/符号推理

知识驱动（认知智能）

2012 -
基于深度神经网络，深度学习

- 深度神经网络
- 词向量等分布式表示
- 非线性激活

数据驱动（感知智能）

$$P(w_{i-1}, w_i) = \frac{N(w_{i-1})}{N}$$

$$P(w_{i-1}) = \frac{N(w_{i-1})}{N}$$

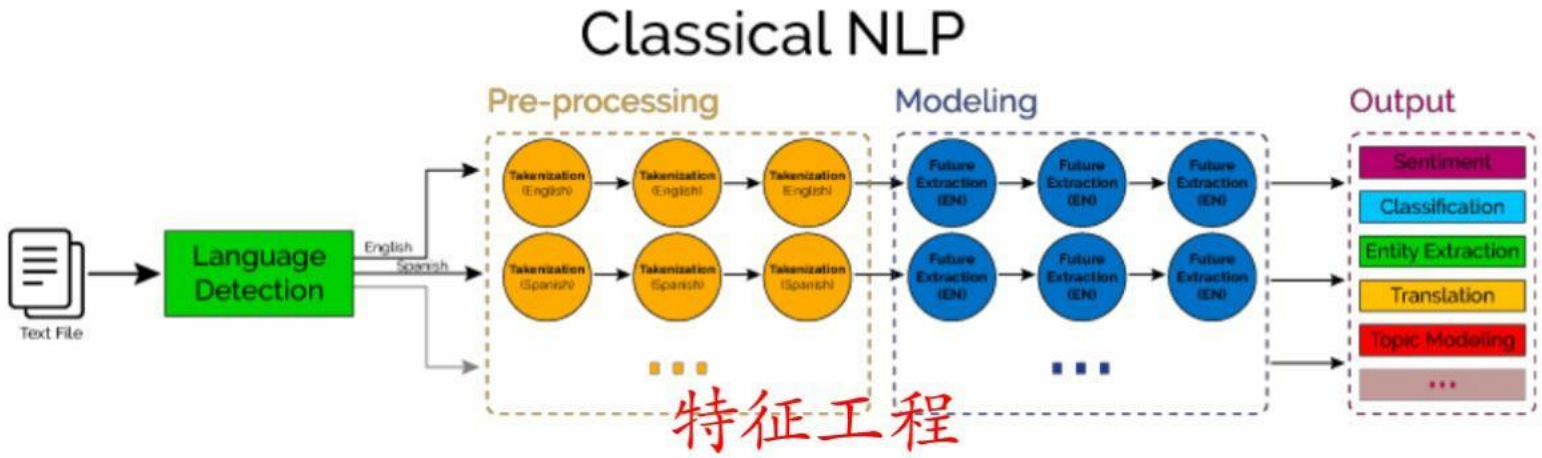
1990 - 2012
基于统计机器学习

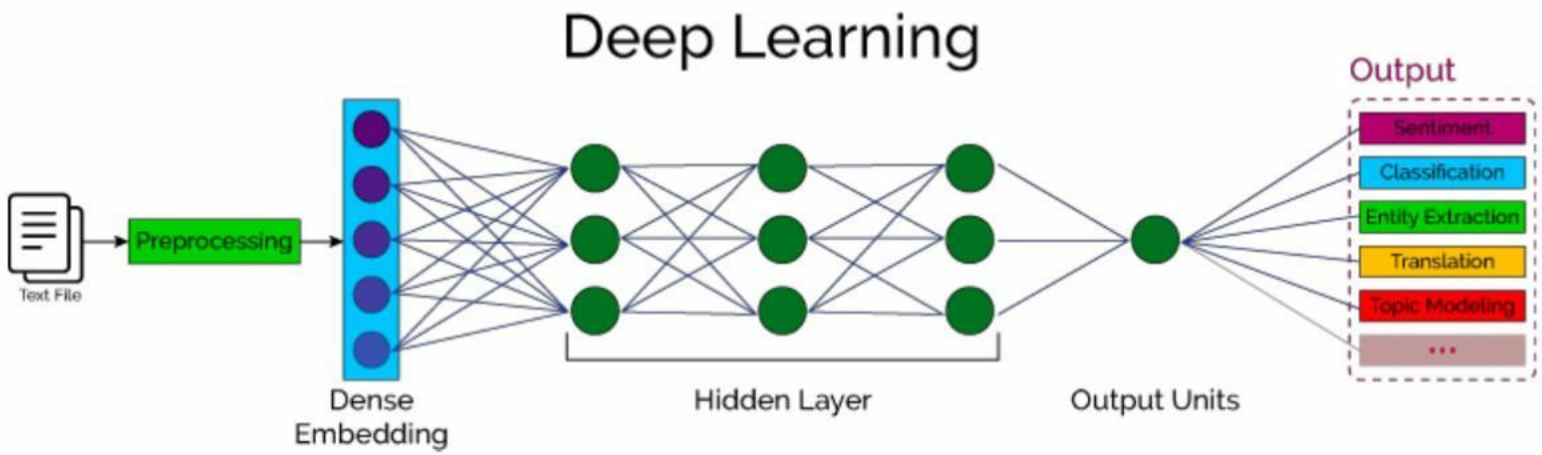
- 词袋模型
- one-hot表示
- SVM等浅层模型

1990年以前
基于符号规则、专家系统

符号规则（运算智能）

自然语言处理两大建模范式：统计学习与深度学习





统计学习

深度学习

图片引用自:【邱锡鹏-自然语言处理中的预训练模型】

深度学习建模一般流程

- 深度学习=表示学习+浅层学习

- 缺点：对数据规模敏感、可解释性差

图片参考:【邱锡鹏-自然语言处理中的预训练模型】

语言表示难点

- 不同于视频、图像、语音，语言为离散的信号
- 难点：如何表示语言的语义？

- 压缩、低维、稠密向量
- 用O(N)个参数表示$O(2^k)$区间(k为非零参数, k<N)

图片表示—颜色

名称	RGB取值
绿	[0,1,0]
蓝	[0,0,1]
红	[1,0,0]
海军蓝	[0,0,128]
中国红	[0.67,0.22,0.12]

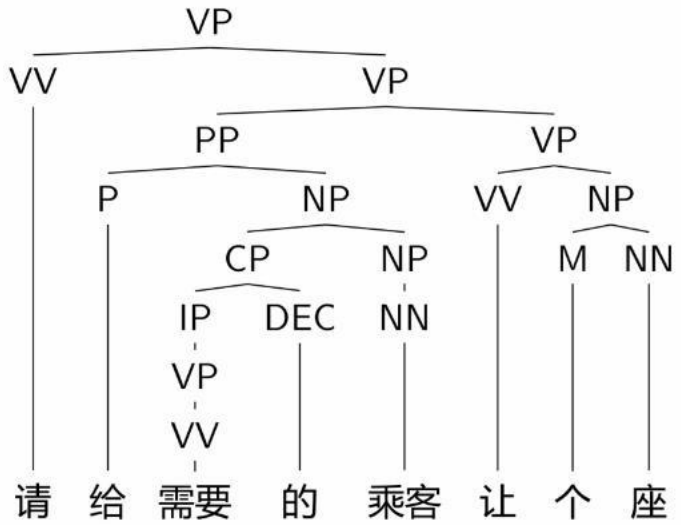
词嵌入（Word Embeddings）

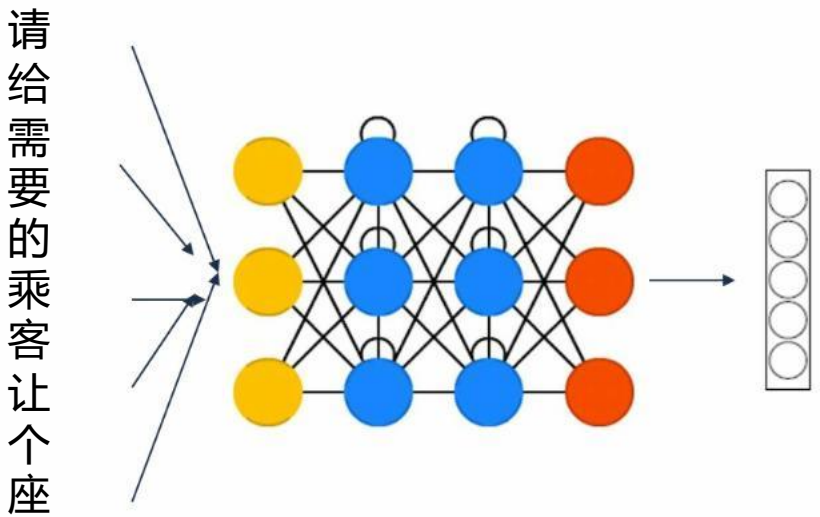
$$\text{employees} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 10.109 \\ -0.542 \\ 0.349 \\ 0.271 \\ 0.487 \end{pmatrix}$$

语言表示学习

- 词、短语
  - ✓组合语义模型
- 句子
  - ✓序列模型
  - ✓递归模型
  - ✓卷积模型
  - ✓自注意力模型
- 篇章
  - ✓层次模型

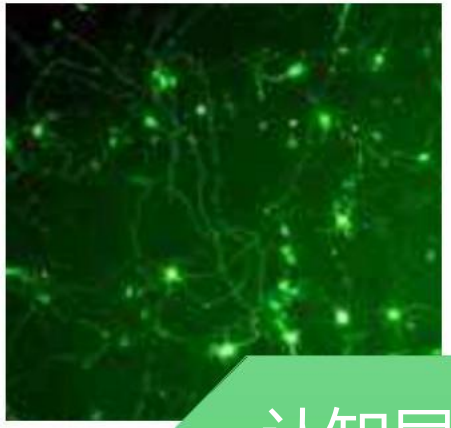
短语句法





图片引用自:【邱锡鹏-自然语言处理中的预训练模型】

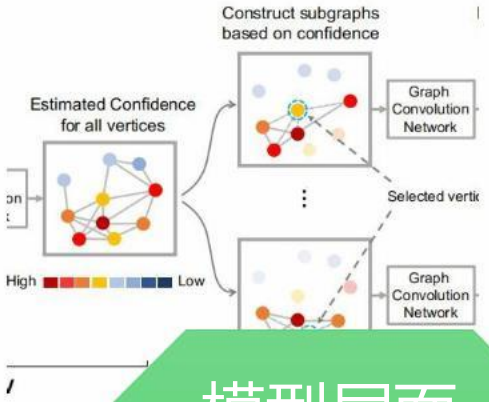
语言表示学习的三个层面问题



认知层面

- 场景上下文
- 知识（常识）

语义表示



模型层面

- 语义组合问题
- 长期依赖问题

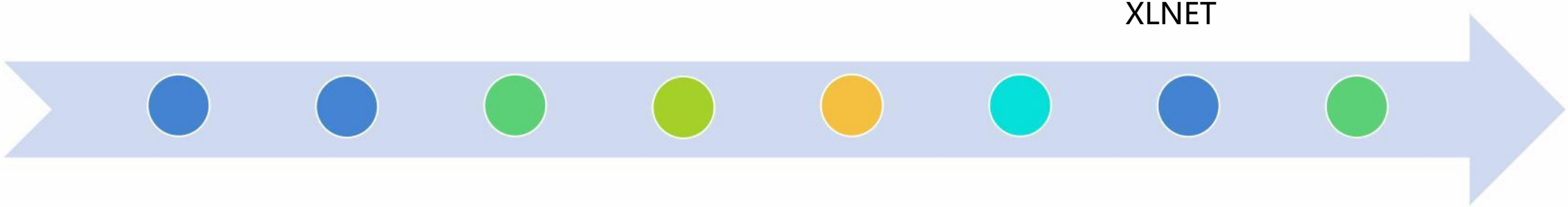
模型驱动

学习层面

- 迁移学习
- 多任务学习

数据驱动

NLP模型演变



- 2013
  - word2vec
- 2014
  - Glove
  - Recursive NN
  - RNN(LSTM/GRU)
  - CNN
  - seq2seq
- 2015
  - Tree-LSTM
  - LSTM+Attention
- 2016
  - Self-Attention
- 2017
  - Transformer
  - ELMO
- 2018
  - BERT
  - GNN
- 2019
  - ERNIE
  - RoBERTa
  - SpanBERT
  - XLNET
- 2020
  - GPT-3
  - CPM (中文版GPT-3)
  - InfoXLM (对比学习)

分布式假设 (Distributional hypothesis)

- 分布式假设：具有相似上下文的词，倾向于具有相似的语义

J.R.Firth 1957

- "You shall know a word by the company it keeps"
- One of the most successful ideas of modern statistical NLP!

…government debt problems turning into	***banking***	*crises as happened in 2009…*
…saying that Europe needs unified	***banking***	*regulation to replace the hodgepodge…*
…India has just given its	***banking***	*system a shot in the arm…*

可以利用上下文的词表示“banking”这个词本身的语义

词向量：Milolov et al, 2013 主要贡献

- An improved version of *skip-gram* algorithm
  - Negative sampling (vs hierarchical softmax in the earlier paper)
  - Subsampling of frequent words

- You can also learn good vector presentations for phrases!

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

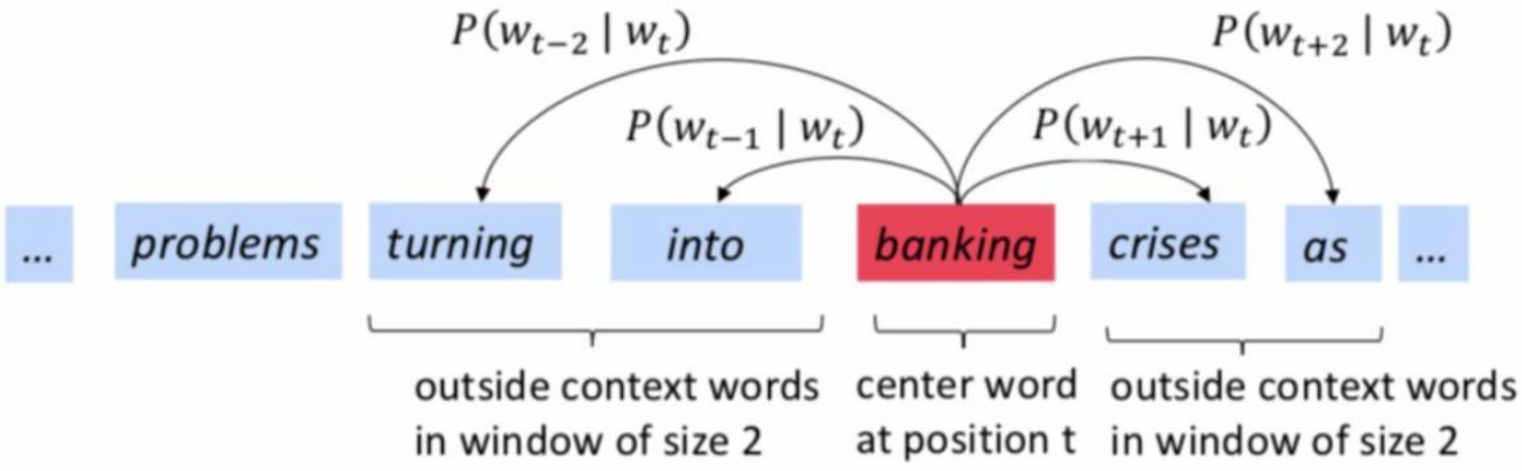
Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

词向量：Skip-gram模型

- 核心思想：用词来预测它上下文中的词
- 上下文：尺寸为 2m 的固定窗口



词向量：Skip-gram的目标函数

- For each position $t = 1,2,\ldots T$, predict context words within context size m, given center word w_j:

需进行优化的全部参数

$$\mathcal{L}(\theta) = \prod_{t=1}^{T} \prod_{-m \le j \le m, j \neq 0} P(w_{t+j} \mid w_t; \theta)$$

- The objective function $J(\theta)$ is the (average) negative log likelihood:

$$J(\theta) = -\frac{1}{T} \log \mathcal{L}(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \le j \le m, j \neq 0} \log P(w_{t+j} \mid w_t; \theta)$$

如何定义条件概率 $P(w_{t+j} \mid w_t; \theta)$

- 两个向量集合，对应于词汇表中每个单词

$$\mathbf{u}_i \in \mathbb{R}^d : \text{ 中心词 i 的嵌入}$$

$$\mathbf{v}_{i'} \in \mathbb{R}^d : \text{ 上下文 i' 的嵌入}$$

- 使用内积 $\mathbf{u}_i \cdot \mathbf{v}_{i'}$ 来测量单词 i 与上下文单词 i' 一起出现的可能性，越大越好

$$P(w_{t+j} \mid w_t) = \frac{\exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_{w_{t+j}})}{\sum_{k \in V} \exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_k)}$$

$\theta = \{\{\mathbf{u}_k\}, \{\mathbf{v}_k\}\}$: 这个模型中的所有参数!

负采样

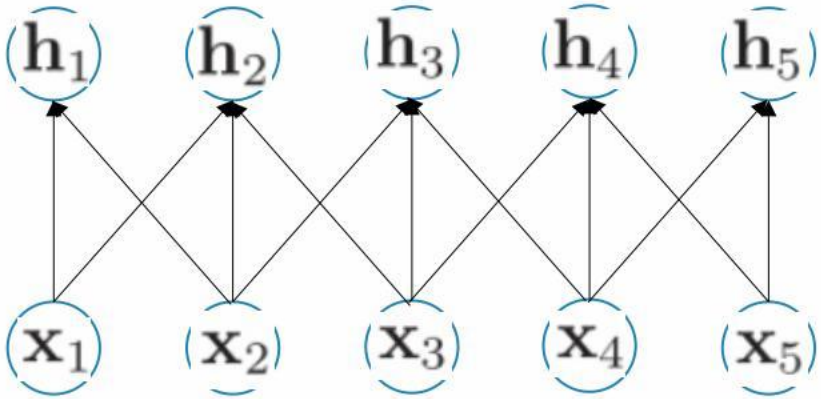
- SGNS = Skip-gram with negative sampling
- Intuition: for each (w, c) pair, we sample k negative pairs (w, c'):

$$P(D = 1 \mid w, c) = \frac{1}{1 + \exp(-\mathbf{u}_w \cdot \mathbf{v}_c)}$$

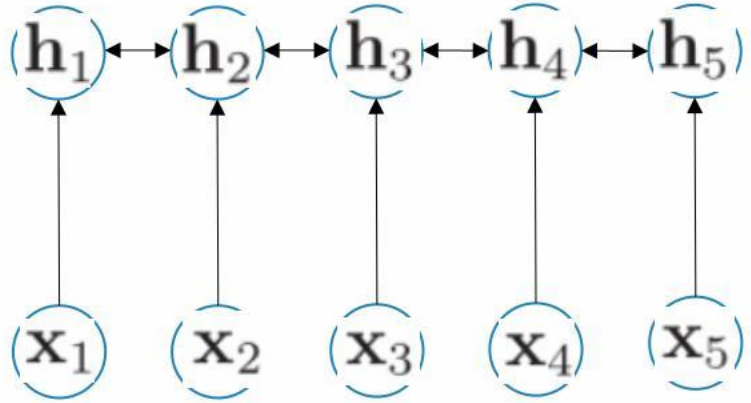
$$P(D = 0 \mid w, c') = \frac{\exp(-\mathbf{u}_w \cdot \mathbf{v}_{c'})}{1 + \exp(-\mathbf{u}_w \cdot \mathbf{v}_{c'})}$$

两大经典的词序列建模模型

- 当使用神经网络来处理一个变长的向量序列时，我们通常可以使用卷积网络或循环网络进行编码来得到一个相同长度的输出向量序列



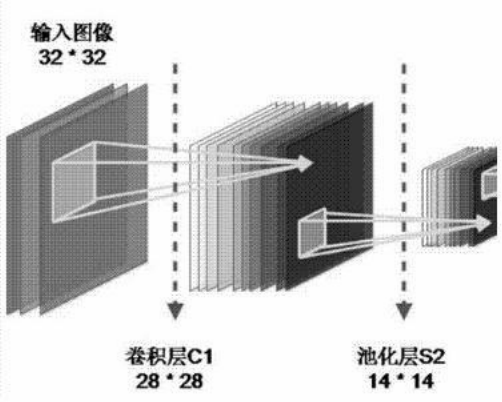
卷积网络



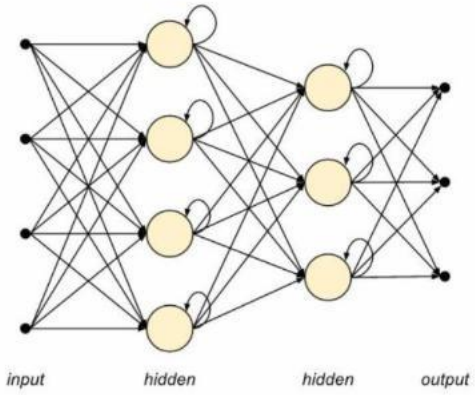
双向循环网络

缺点：擅长建模输入信息的局部依存关系，对长距离依存不敏感

新一代的词序列模型：Transformer

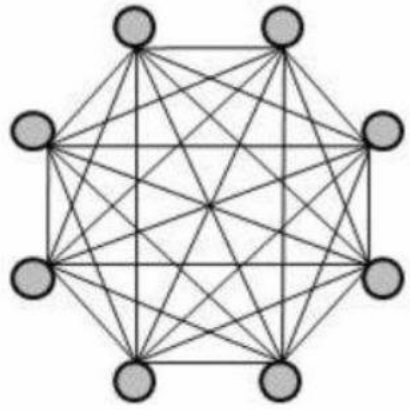


CNN



RNN

经典



Transformer

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar†
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

SOTA

注意力机制 Attention

- 软注意力机制（Soft attention mechanism）
- 软注意力计算包含两个步骤：

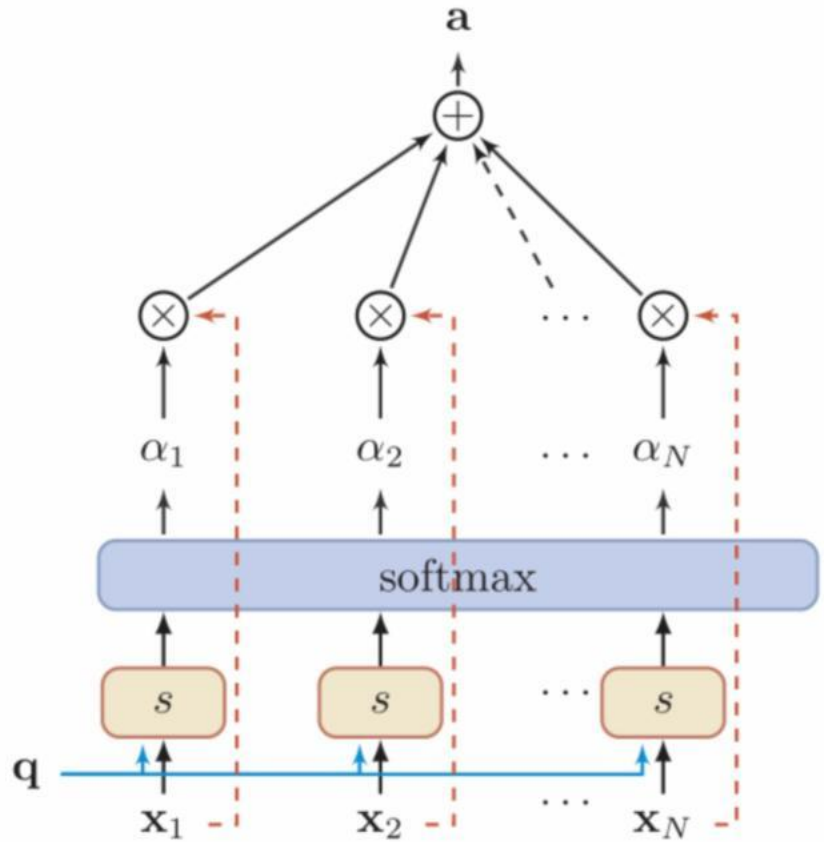
① 计算注意力分布α

$$\begin{aligned}\alpha_i &= p(z=i|X,\mathbf{q}) \\ &= \text{softmax}\left(s(\mathbf{x}_i,\mathbf{q})\right) \\ &= \frac{\exp\left(s(\mathbf{x}_i,\mathbf{q})\right)}{\sum_{j=1}^{N}\exp\left(s(\mathbf{x}_j,\mathbf{q})\right)}\end{aligned}$$

$s(x_i, q)$ 打分函数

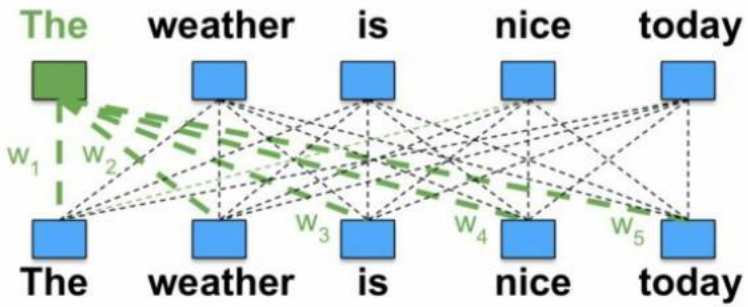
② 根据α计算输入信息的加权求和

$$\begin{aligned}\mathbf{att}(X,\mathbf{q}) &= \sum_{i=1}^{N}\alpha_i\mathbf{x}_i, \\ &= \mathbb{E}_{z\sim p(z|X,\mathbf{q})}[\mathbf{x}]\end{aligned}$$



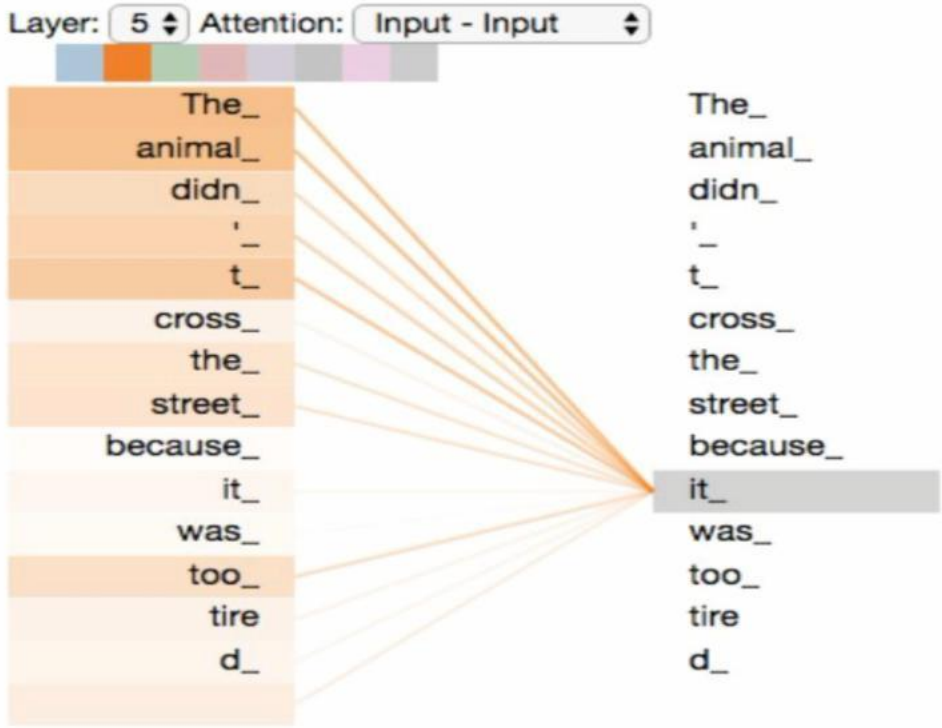
图片引用自：【邱锡鹏-自然语言处理中的预训练模型】

自注意力机制 Self-attention



$$w_1, w_2, w_3, w_4, w_5 = \text{softmax}\left(\begin{bmatrix} 0.6 & 0.2 & 0.8 \end{bmatrix}_{\text{The}} \times \begin{bmatrix} 0.6 & 0.2 & 0.9 & 0.4 & 0.4 \\ 0.2 & 0.3 & 0.1 & 0.1 & 0.1 \\ 0.8 & 0.1 & 0.8 & 0.4 & 0.6 \end{bmatrix}_{\text{The weather is nice today}} \right)$$

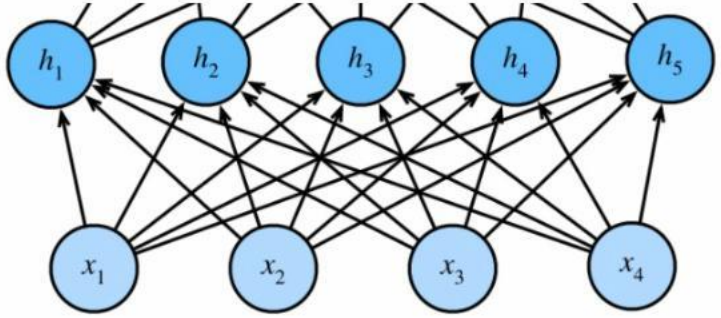
$$\begin{bmatrix} 1.8 \\ 2.3 \\ 0.4 \end{bmatrix}_{\text{The}} = w_1 \times \begin{bmatrix} 0.6 \\ 0.2 \\ 0.8 \end{bmatrix}_{\text{The}} + w_2 \times \begin{bmatrix} 0.2 \\ 0.3 \\ 0.1 \end{bmatrix}_{\text{weather}} + w_3 \times \begin{bmatrix} 0.9 \\ 0.1 \\ 0.8 \end{bmatrix}_{\text{is}} + w_4 \times \begin{bmatrix} 0.4 \\ 0.1 \\ 0.4 \end{bmatrix}_{\text{nice}} + w_5 \times \begin{bmatrix} 0.4 \\ 0.1 \\ 0.6 \end{bmatrix}_{\text{today}}$$

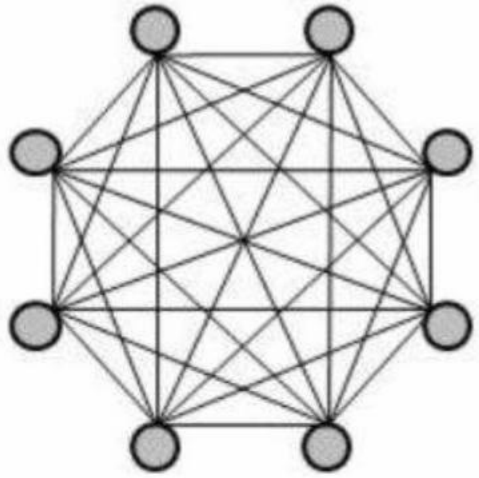


图片引用自: http://fuyw.top/NLP_02_QANet/

Transformer：定义

- Transformer本质一种基于自注意力的全连接神经网络

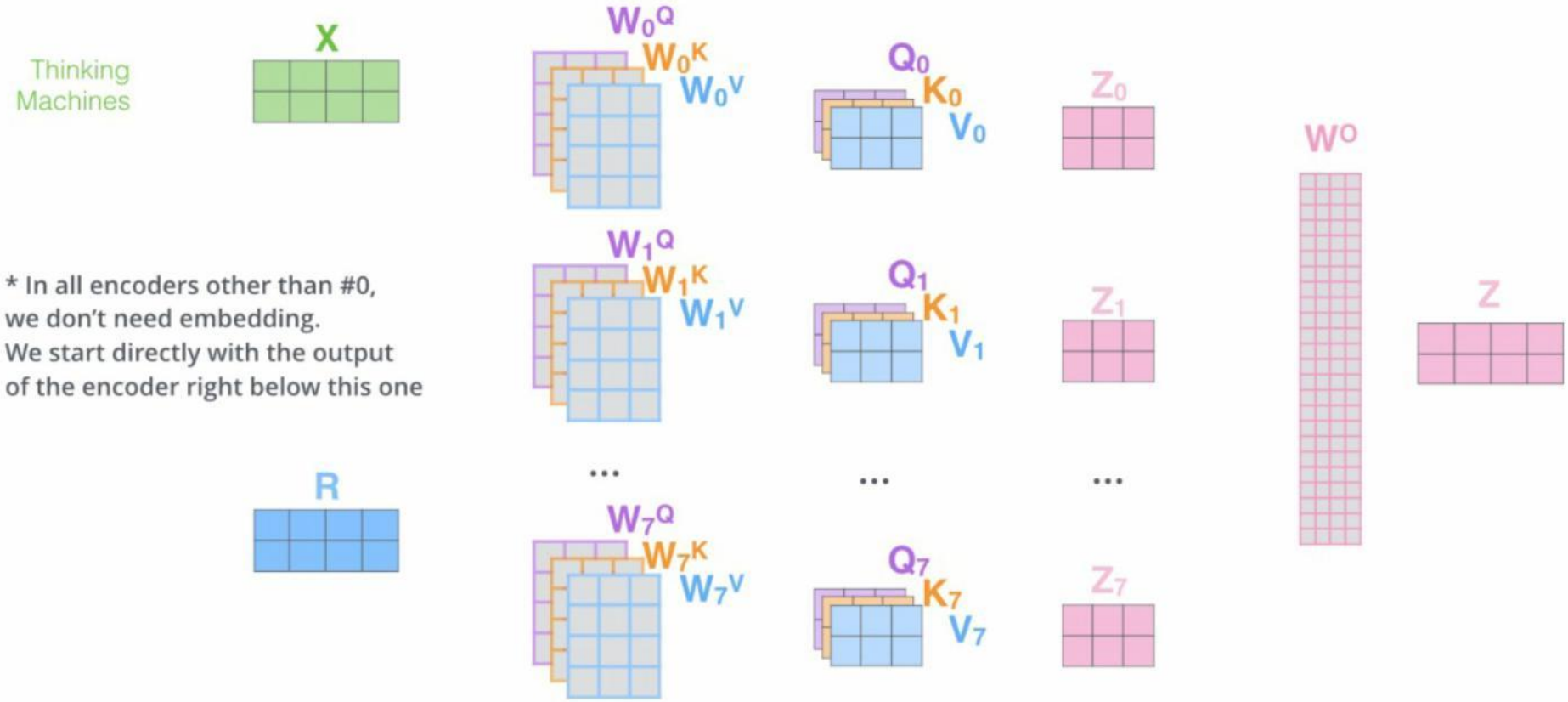




连接权重 a_{ij} 由注意力机制动态生成

也可以看作是一种全连接的图神经网络

多头(multi head)自注意力模型



不同的头部代表不同的表示子空间

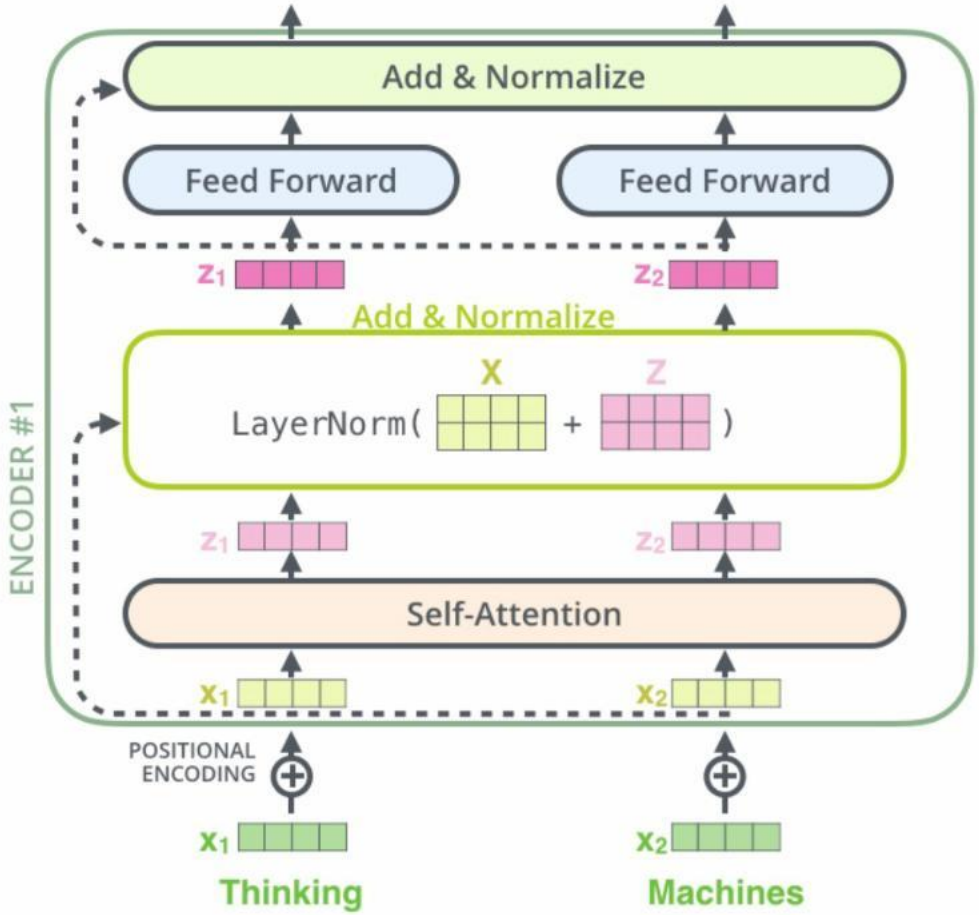
本质：利用一个向量而不是一个数值，表示任意两个token之间的权重关系。

Shen et al., 2017, DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding

图片引用自: http://jalammar.github.io/illustrated-transformer/

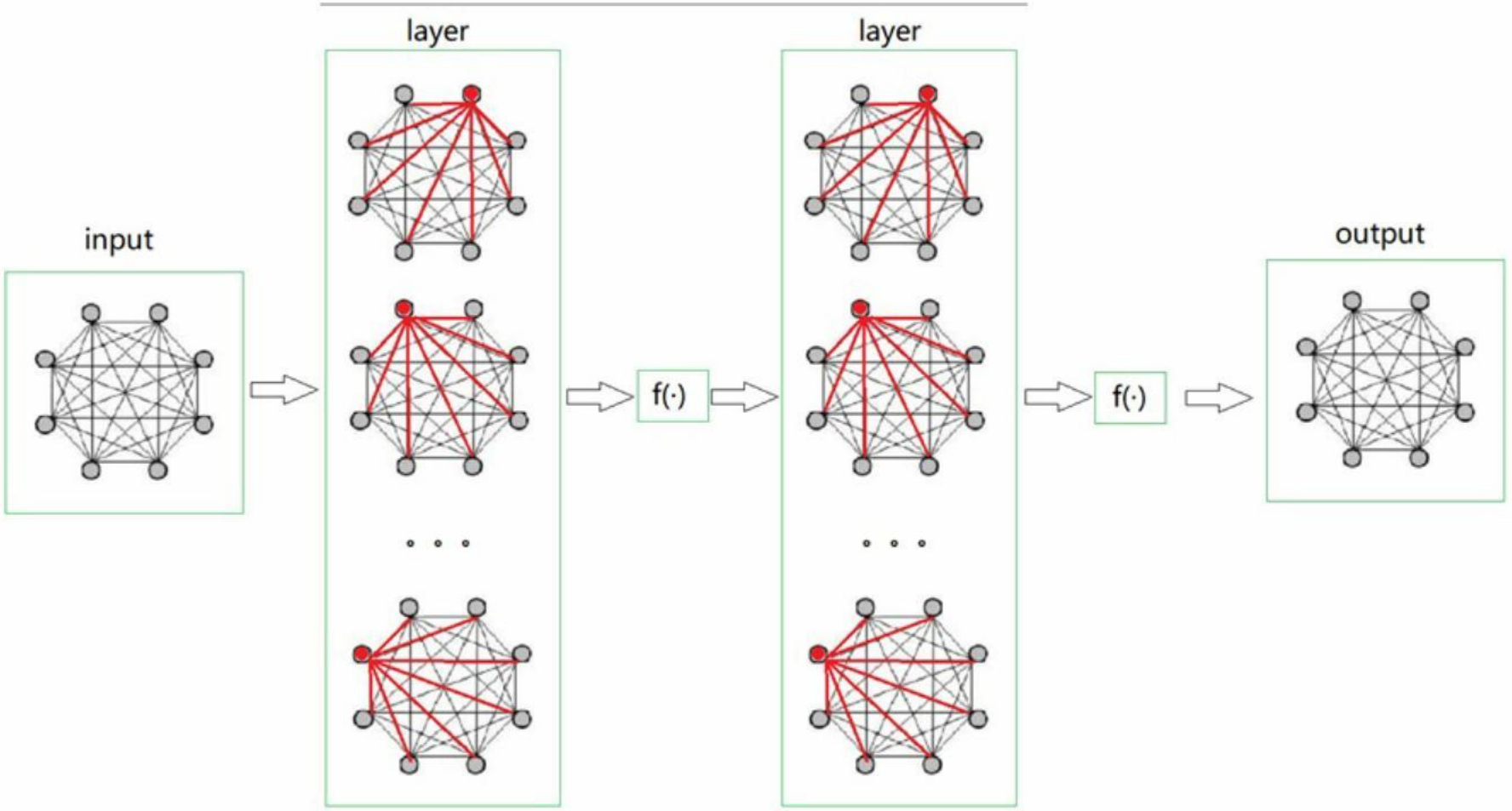
Transformer Encoder

- 仅仅自注意力还不够
- 其它操作
  - 位置编码
  - 层归一化
  - 直连边
  - 逐位的FNN



图片引用自: http://jalammar.github.io/illustrated-transformer/

Transformer



图片引用自: 【邱锡鹏-自然语言处理中的预训练模型】

三种序列模型复杂度比较

模型	每层复杂度	序列操作数	最大路径长度
CNN	$O(kLd^2)$	$O(1)$	$O(\log_k(L))$
RNN	$O(Ld^2)$	$O(L)$	$O(L)$
Transformer	$O(L^2d)$	$O(1)$	$O(1)$

k：卷积核大小　　L：序列长度　　d：维度

本次课程总结

知识点

- 自然语言处理的由统计学习到深度学习的技术演变
- 语言表示的难点（离散特点与语义表示）
- 词向量模型（Skip-gram算法与负采样）
- 文本序列模型Transformer（自注意力机制本质是一个图神经网络）

Thanks!

Q & A

NLP

LABORATORY