《神经网络与深度学习》

注意力机制与外部记忆

https://nndl.github.io/

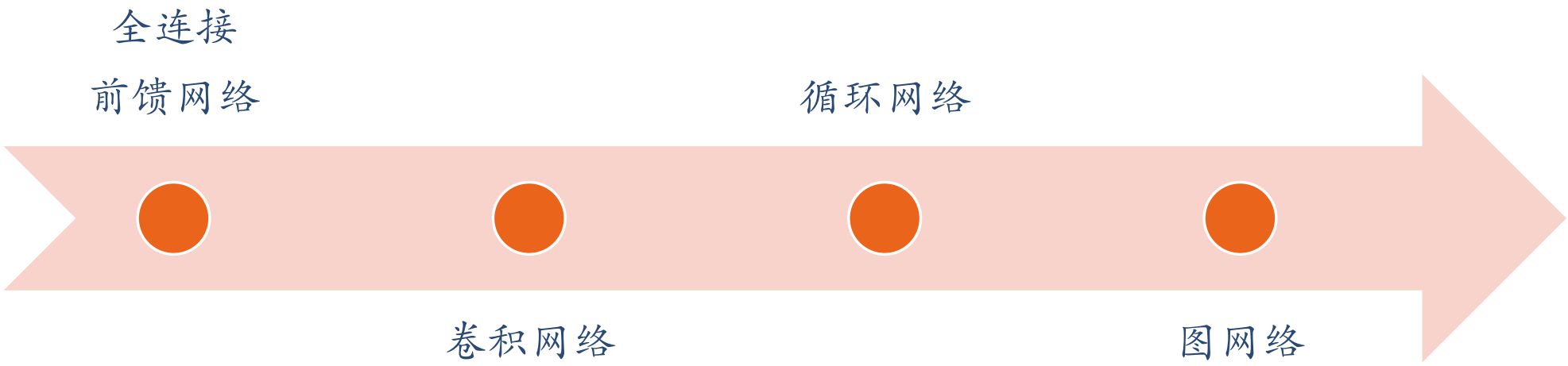
内容

- 注意力机制
  - 注意力机制
  - 应用到机器学习
    - 看图说话，机器翻译
    - 模型(Hierarchical Attention Network, Graph Attention Network, Pointer Network)
  - 自注意力模型(Transformer)
- 外部记忆
  - 记忆增强网络(Memory Network, Neural Turing Machine)

参考资料

- 《神经网络与深度学习》第8、15章
  - https://nndl.github.io/

网络能力



增加网络能力的另一种思路：

- 注意力机制
- 外部记忆

注意力机制

例子：阅读理解 SQuAD

The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, Il milione (or, The Million, known in English as the Travels of Marco Polo), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge through contact with Persian traders since many of the places he named were in Persian.

How did some suspect that Polo learned about China instead of by actually visiting it?

Answer: through contact with Persian traders

通用近似定理

定理 4.1 – 通用近似定理（Universal Approximation Theorem）[Cybenko, 1989, Hornik et al., 1989]： 令 $\varphi(\cdot)$ 是一个非常数、有界、单调递增的连续函数，$\mathcal{I}_d$ 是一个 d 维的单位超立方体 $[0,1]^d$，$C(\mathcal{I}_d)$ 是定义在 $\mathcal{I}_d$ 上的连续函数集合。对于任何一个函数 $f \in C(\mathcal{I}_d)$，存在一个整数 m，和一组实数 $v_i, b_i \in \mathbb{R}$ 以及实数向量 $\mathbf{w}_i \in \mathbb{R}^d$，$i = 1, \cdots, m$，以至于我们可以定义函数

$$F(\mathbf{x}) = \sum_{i=1}^{m} v_i \varphi(\mathbf{w}_i^{\mathrm{T}} \mathbf{x} + b_i), \tag{4.33}$$

作为函数 f 的近似实现，即

$$|F(\mathbf{x}) - f(\mathbf{x})| < \epsilon, \forall \mathbf{x} \in \mathcal{I}_d. \tag{4.34}$$

其中 $\epsilon > 0$ 是一个很小的正数。

根据通用近似定理，对于具有线性输出层和至少一个使用“挤压”性质的激活函数的隐藏层组成的前馈神经网络，只要其隐藏层神经元的数量足够，它可以以任意的精度来近似任何从一个定义在实数空间中的有界闭集函数。

通用近似定理

- 由于优化算法和计算能力的限制，神经网络在实践中很难达到通用近似的能力。
  - 网络不能太复杂（参数太多）
- 如何提高网络能力
  - 局部连接
  - 权重共享
  - 汇聚操作
  - ？

大脑中的注意力

- 人脑每个时刻接收的外界输入信息非常多，包括来源于视觉、听觉、触觉的各种各样的信息。
- 但就视觉来说，眼睛每秒钟都会发送千万比特的信息给视觉神经系统。
- 人脑通过注意力来解决信息超载问题。

注意力示例


The New York Times
HUBLOT
Cuomo Urges Broad Limits to N.Y. Public Pensions
Pick for Afghan Envoy Says U.S. Can't Afford to Abandon Effort
Firefighters Struggle With Arizona Wildfire
Banks Defeated in Senate Vote Over Debit Card Fees
Iran Plans High Level of Uranium Enrichment
OPEC Keeps Lid on Oil Production Targets
TIMES DIGITAL SUBSCRIPTIONS
LINCOLN

注意力示例

- 当一个人在吵闹的鸡尾酒会上和朋友聊天时，尽管周围噪音干扰很多，他还是可以听到朋友的谈话内容，而忽略其他人的声音。

- 同时，如果未注意到的背景声中有重要的词（比如他的名字），他会马上注意到。

鸡尾酒会效应

如何实现？

- 自下而上
  - 无意识的注意力 $\mathrm{Y} = f(X)$

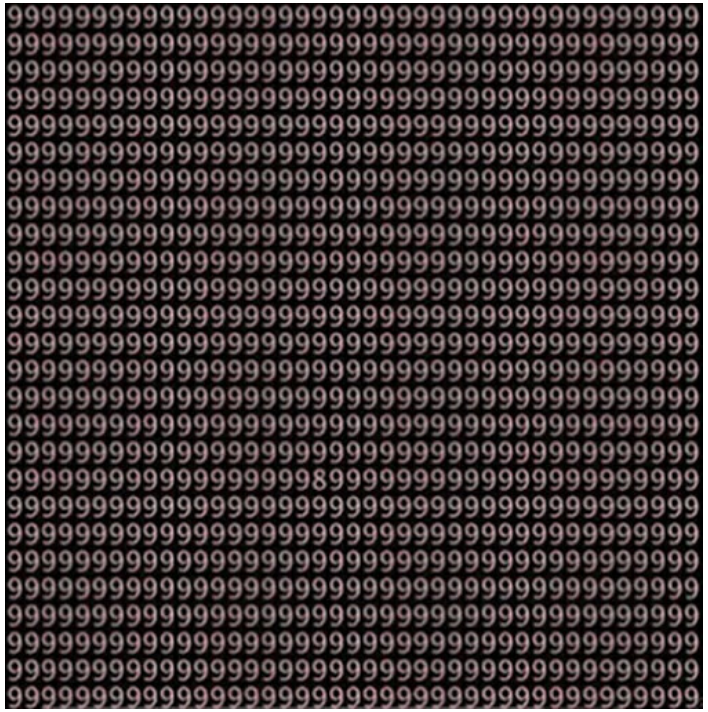
汇聚（pooling）

- 自上而下
  - 有意识的注意力 $\mathrm{Y} = f(X, q)$

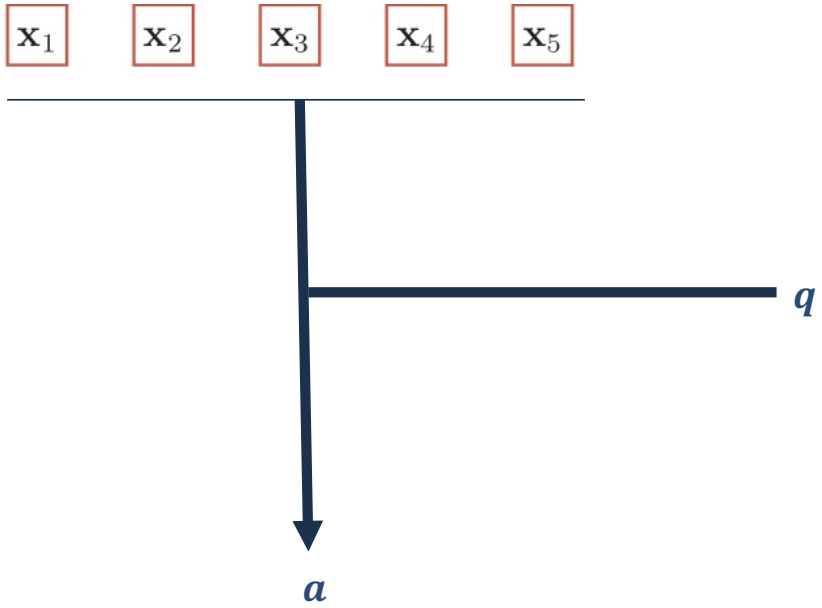
会聚（focus）

应用到机器学习

问题



Try to find 8 less than 8 seconds



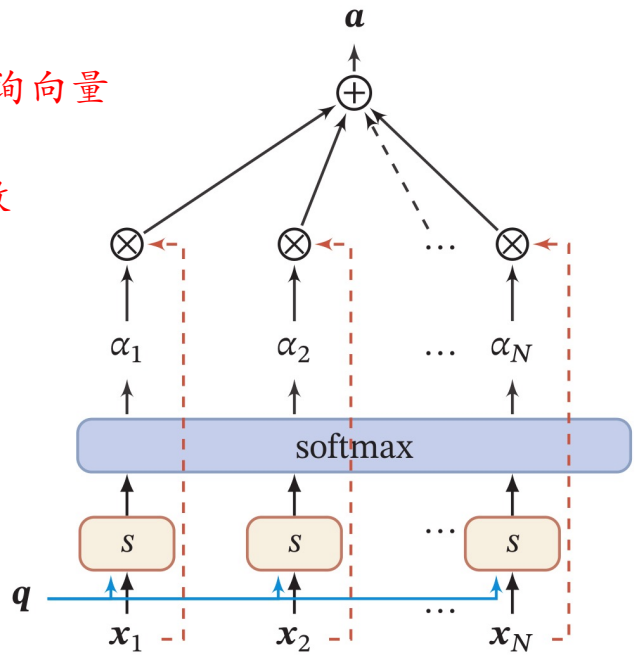
注意力模型

- 软性注意力机制（soft attention mechanism）
- 注意力机制可以分为两步
  - 计算注意力分布α，

$$\begin{aligned}\alpha_n &= p(z=n|\boldsymbol{X},\boldsymbol{q}) \\ &= \text{softmax}\left(s(\boldsymbol{x}_n,\boldsymbol{q})\right) \\ &= \frac{\exp\left(s(\boldsymbol{x}_n,\boldsymbol{q})\right)}{\sum_{j=1}^{N}\exp\left(s(\boldsymbol{x}_j,\boldsymbol{q})\right)}\end{aligned}$$

X为输入向量，q为查询向量

$s(x_n,q)$ 打分函数



  - 根据α来计算输入信息的加权平均。

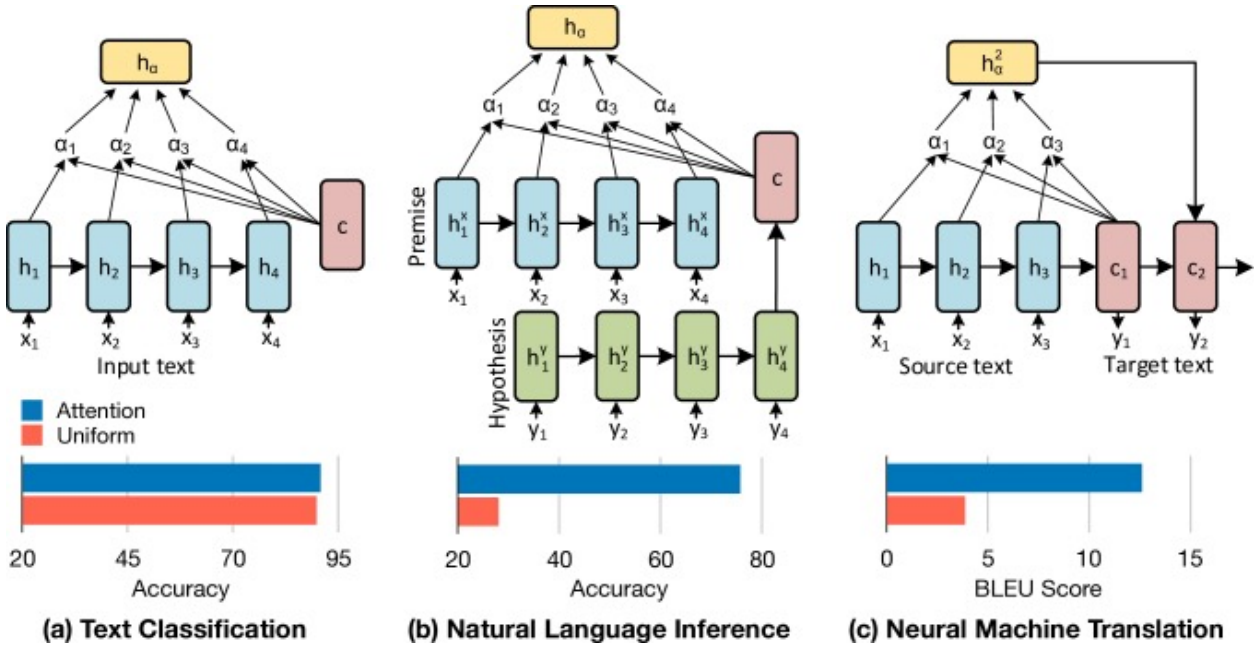
$$\begin{aligned}\text{att}(\boldsymbol{X},\boldsymbol{q}) &= \sum_{n=1}^{N}\alpha_n\boldsymbol{x}_n, \\ &= \mathbb{E}_{z\sim p(z|\boldsymbol{X},\boldsymbol{q})}[\boldsymbol{x}_z]\end{aligned}$$

$z\in[1,N]$为注意力变量，表示被选择信息的索引位置

注意力打分函数$s(x_n, q)$

加性模型	$s(\boldsymbol{x}, \boldsymbol{q}) = \boldsymbol{v}^{\mathrm{T}} \tanh(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{U}\boldsymbol{q})$,
点积模型	$s(\boldsymbol{x}, \boldsymbol{q}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{q}$,
缩放点积模型	$s(\boldsymbol{x}, \boldsymbol{q}) = \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{q}}{\sqrt{D}}$, 相比起点积模型，缓解梯度消失的问题
双线性模型	$s(\boldsymbol{x}, \boldsymbol{q}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{q}$, 相比起点积模型，引入了非对称性

一些常见的注意力



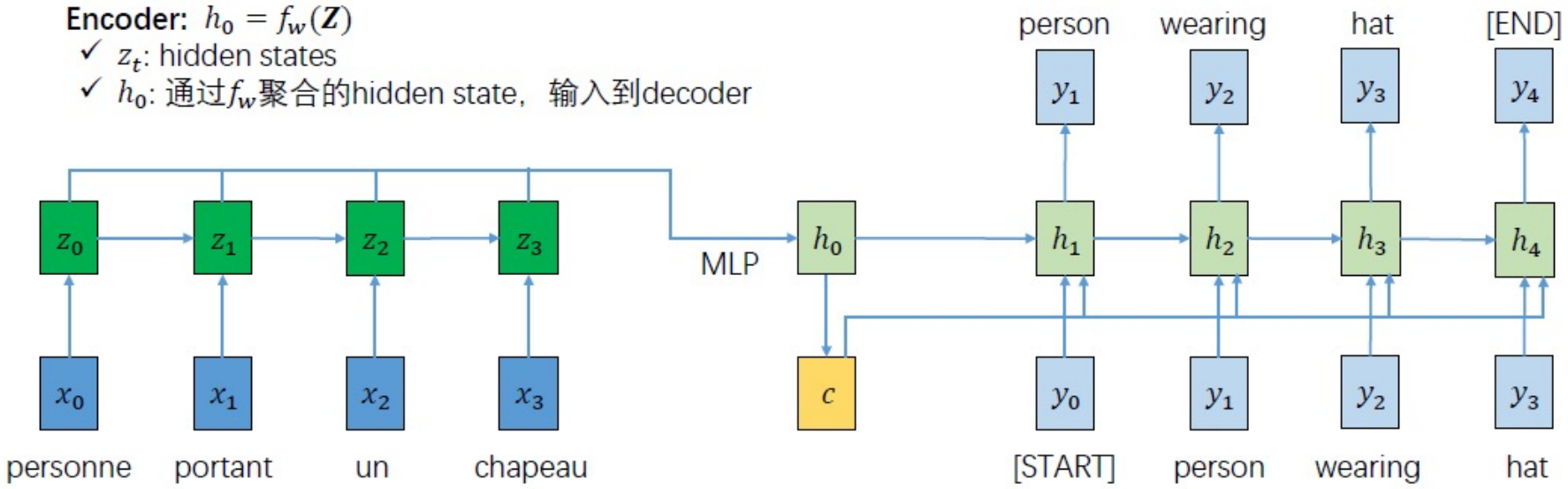
https://www.groundai.com/project/attention-interpretability-across-nlp-tasks/1

机器翻译(无注意力)

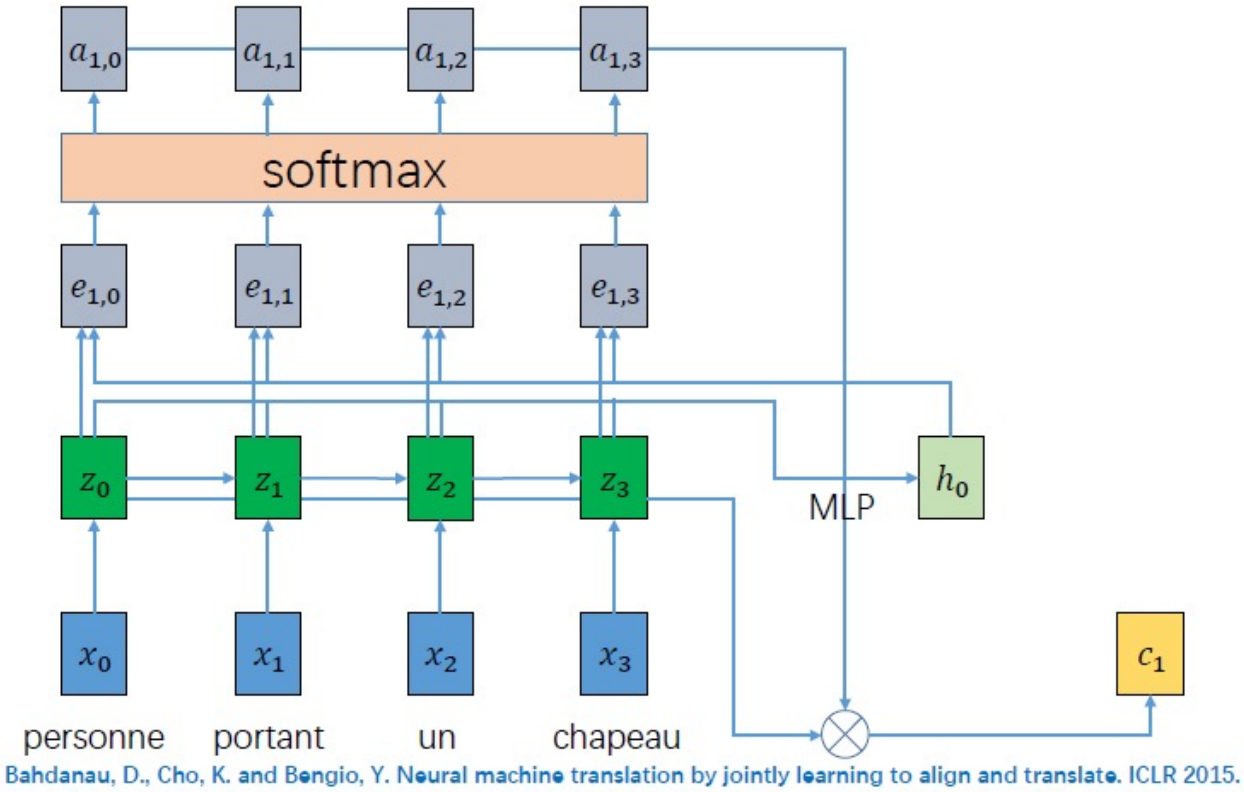
Input:文本 $\boldsymbol{x} = x_1, x_2, \cdots, x_T$
Output: 文本 $\boldsymbol{y} = y_1, y_2, \cdots, y_T$

Encoder: $h_0 = f_w(\boldsymbol{Z})$
- ✓ z_t: hidden states
- ✓ h_0: 通过f_w聚合的hidden state，输入到decoder

Decoder: $y_t = g(h_t) = g_w(y_{t-1}, h_{t-1}, c)$
- ✓ c: contex vector, 辅助文本预测，通常设为h_0



机器翻译(有注意力)



Bahdanau, D., Cho, K. and Bengio, Y. Neural machine translation by jointly learning to align and translate. ICLR 2015.

计算对齐分数:

$e_{t,i} = f_{att}(h_{t-1}, z_i)$

f_{att}为MLP

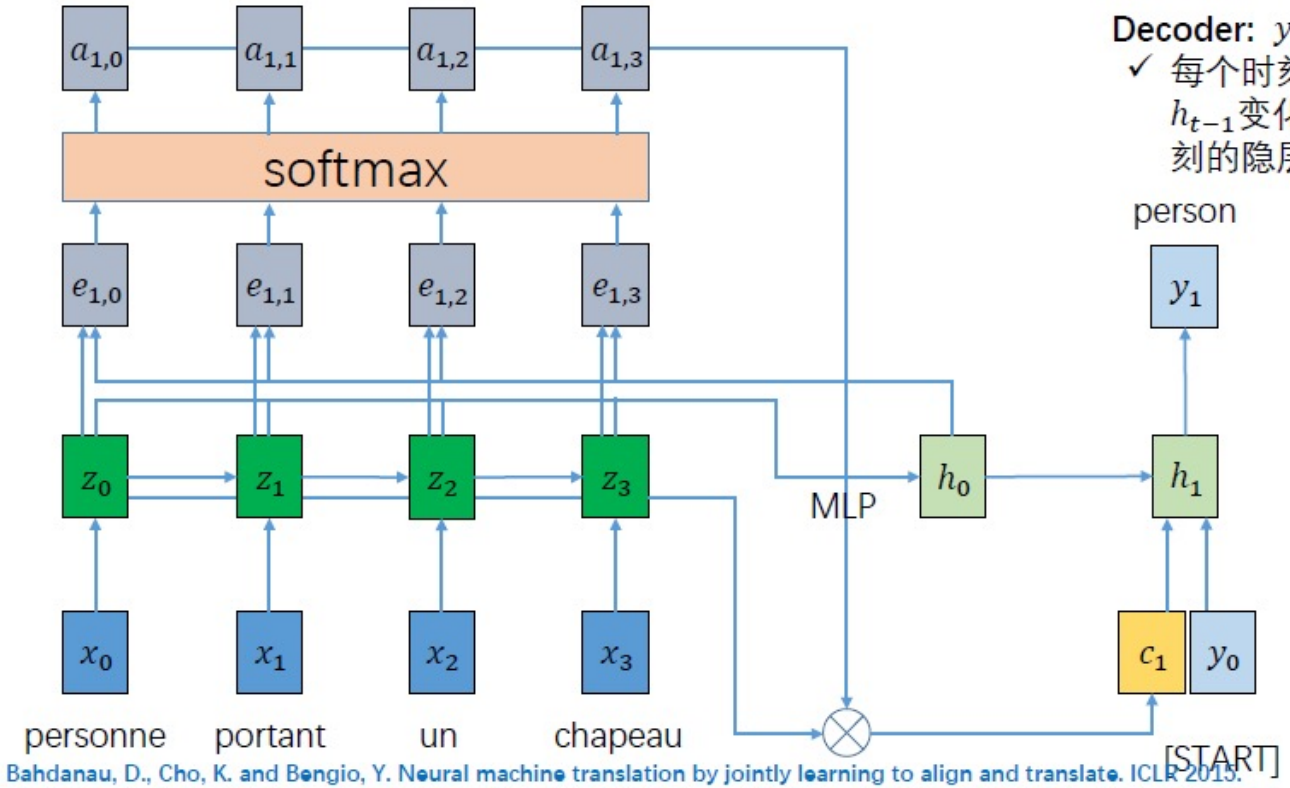
计算注意力分数:

$a_{t,i} = softmax\ (e_{t,i})$

计算context vector:

$c_t = \sum_i a_{t,i}\, z_i$

机器翻译(有注意力)

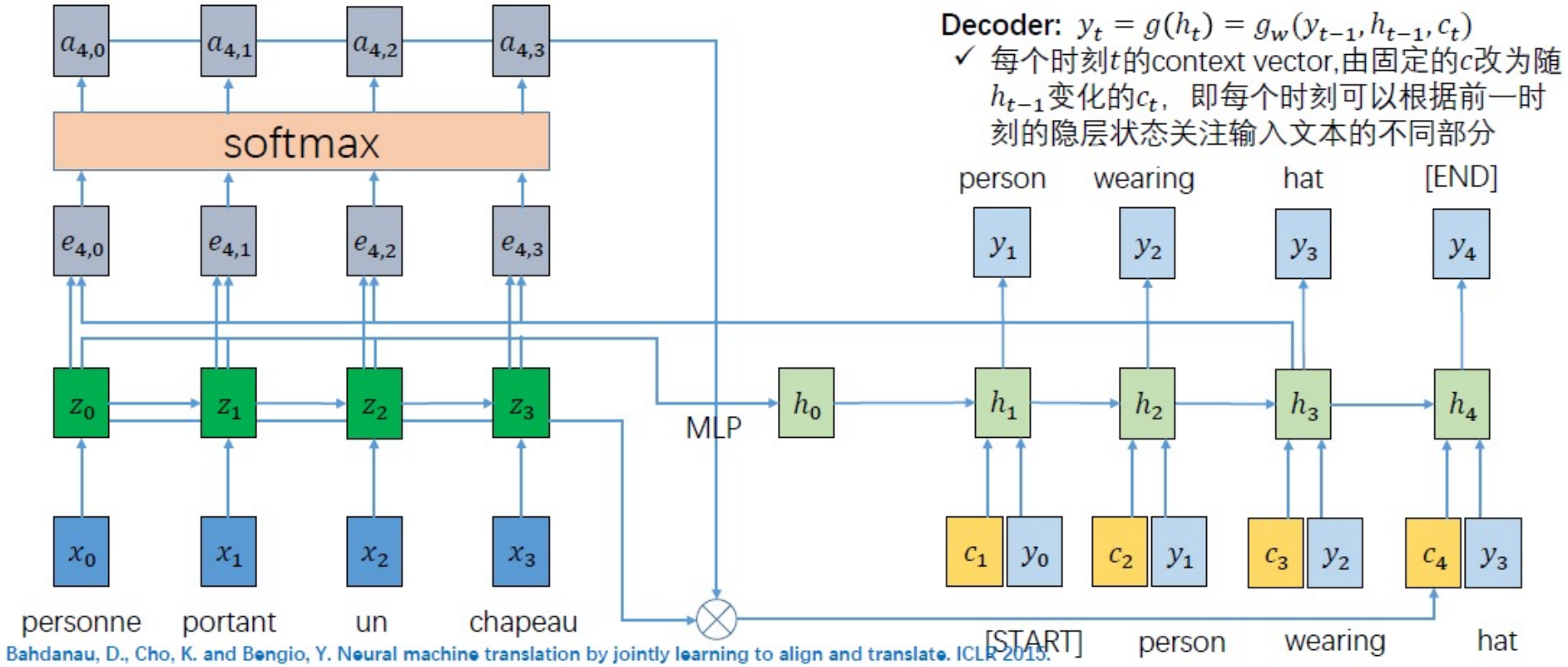


Bahdanau, D., Cho, K. and Bengio, Y. Neural machine translation by jointly learning to align and translate. ICLR 2015.

Decoder: $y_t = g(h_t) = g_w(y_{t-1}, h_{t-1}, c_t)$

✓ 每个时刻t的context vector,由固定的c改为随h_{t-1}变化的c_t，即每个时刻可以根据前一时刻的隐层状态关注输入文本的不同部分

机器翻译(有注意力)


Decoder: $y_t = g(h_t) = g_w(y_{t-1}, h_{t-1}, c_t)$
✓ 每个时刻t的context vector,由固定的c改为随h_{t-1}变化的c_t，即每个时刻可以根据前一时刻的隐层状态关注输入文本的不同部分
softmax
MLP
personne
portant
un
chapeau
[START]
person
wearing
hat
[END]
Bahdanau, D., Cho, K. and Bengio, Y. Neural machine translation by jointly learning to align and translate. ICLR 2015.

机器翻译(有注意力)

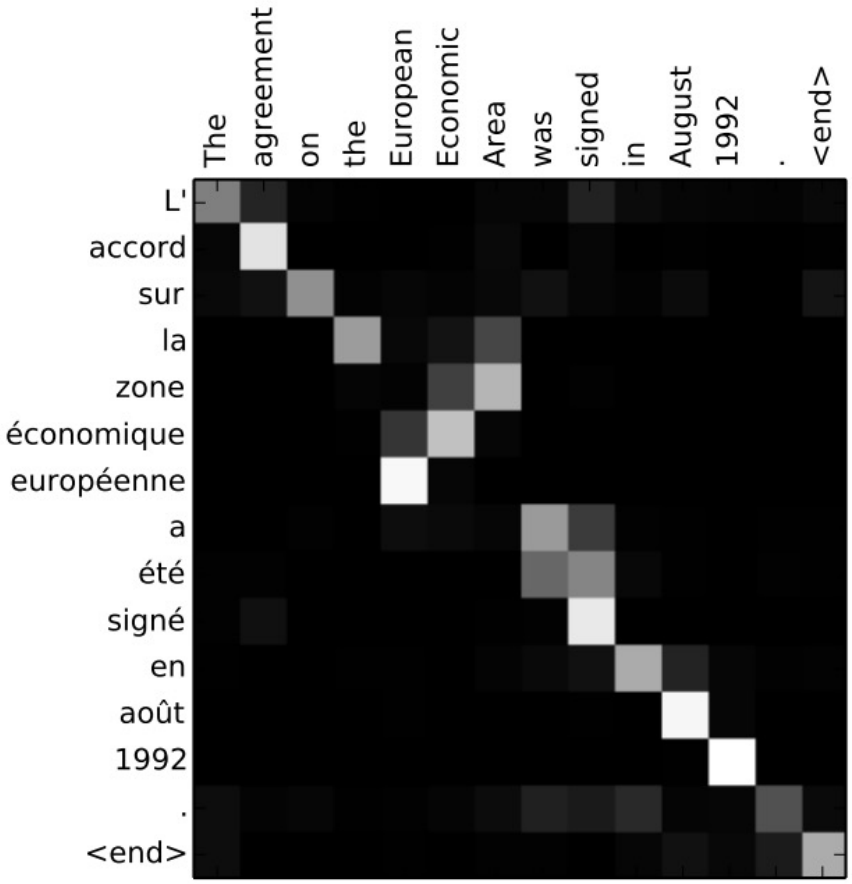

The
agreement
on
the
European
Economic
Area
was
signed
in
August
1992
.
<end>
L'
accord
sur
la
zone
économique
européenne
a
été
signé
en
août
1992
.
<end>

Image Caption

http://kelvinxu.github.io/projects/capgen.html

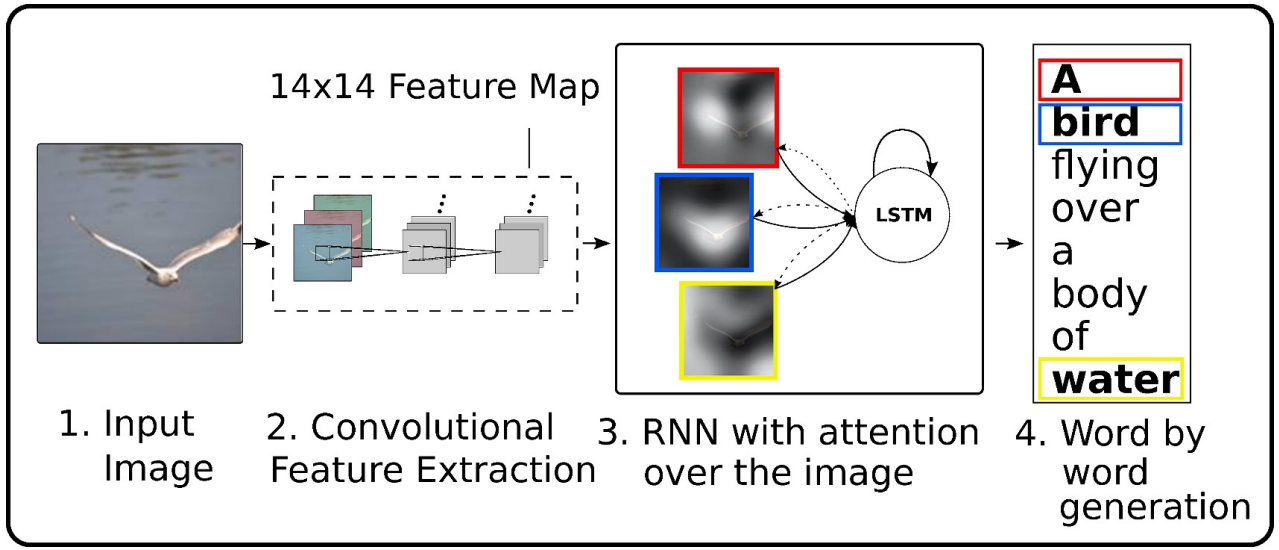


Image Caption (无注意力)

Input: 图片 $\boldsymbol{I}$
Output: 文本 $\boldsymbol{y} = y_1, y_2, \cdots, y_T$

问题：所有时刻都使用相同的、代表整个图片信息的context vector c，不利于不同时刻捕捉差异化的图片信息（bottleneck）

Decoder: $y_t = g(h_t) = g_w(y_{t-1}, h_{t-1}, c)$
✓ c: contex vector, 辅助文本预测，通常设为h_0

Encoder: $h_0 = f_w(\boldsymbol{Z})$
✓ $\boldsymbol{Z}$: CNN提取的图片特征
✓ $f_w()$: Multi-Layer Perceptron（若干层FC）

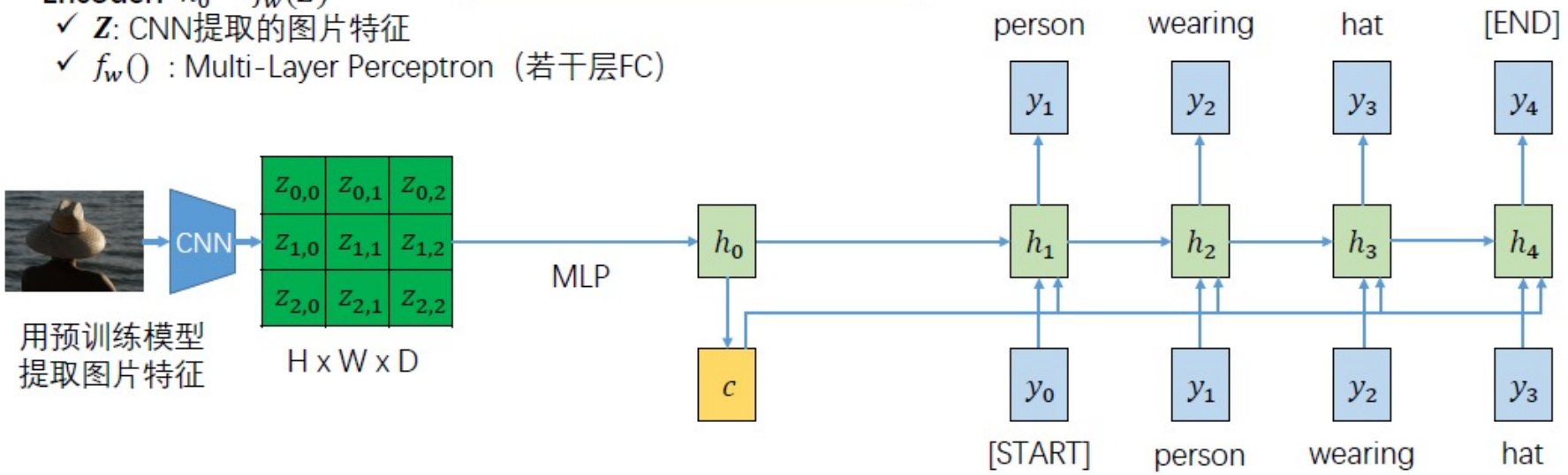
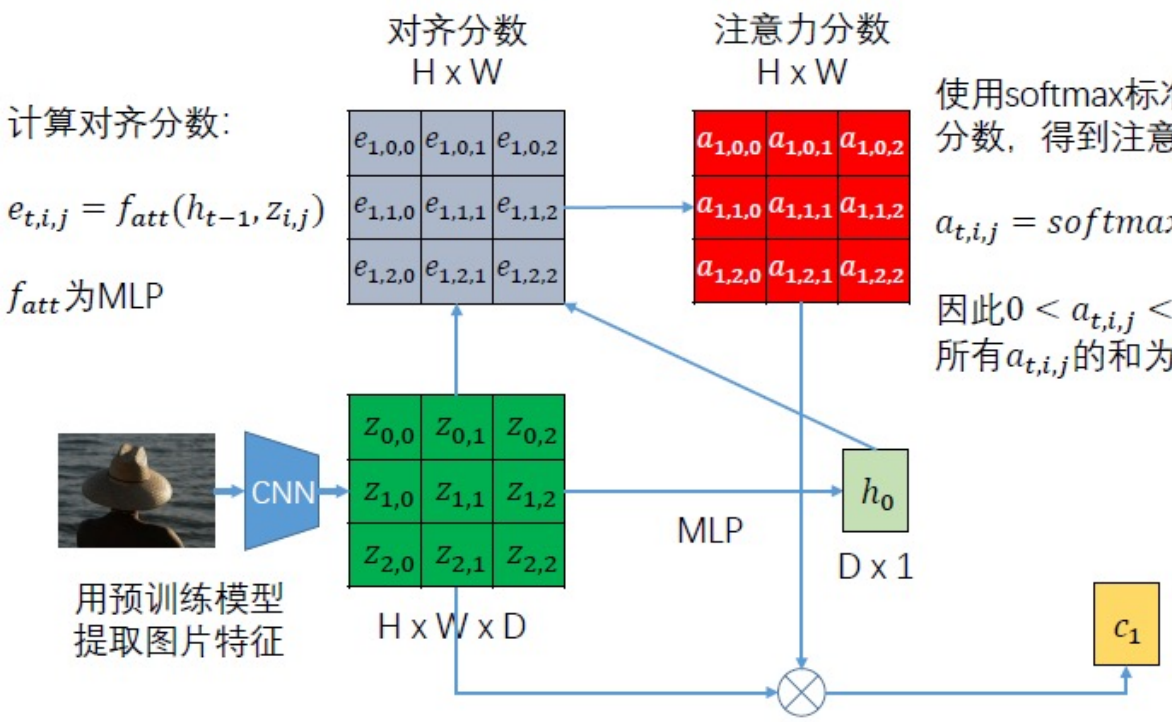
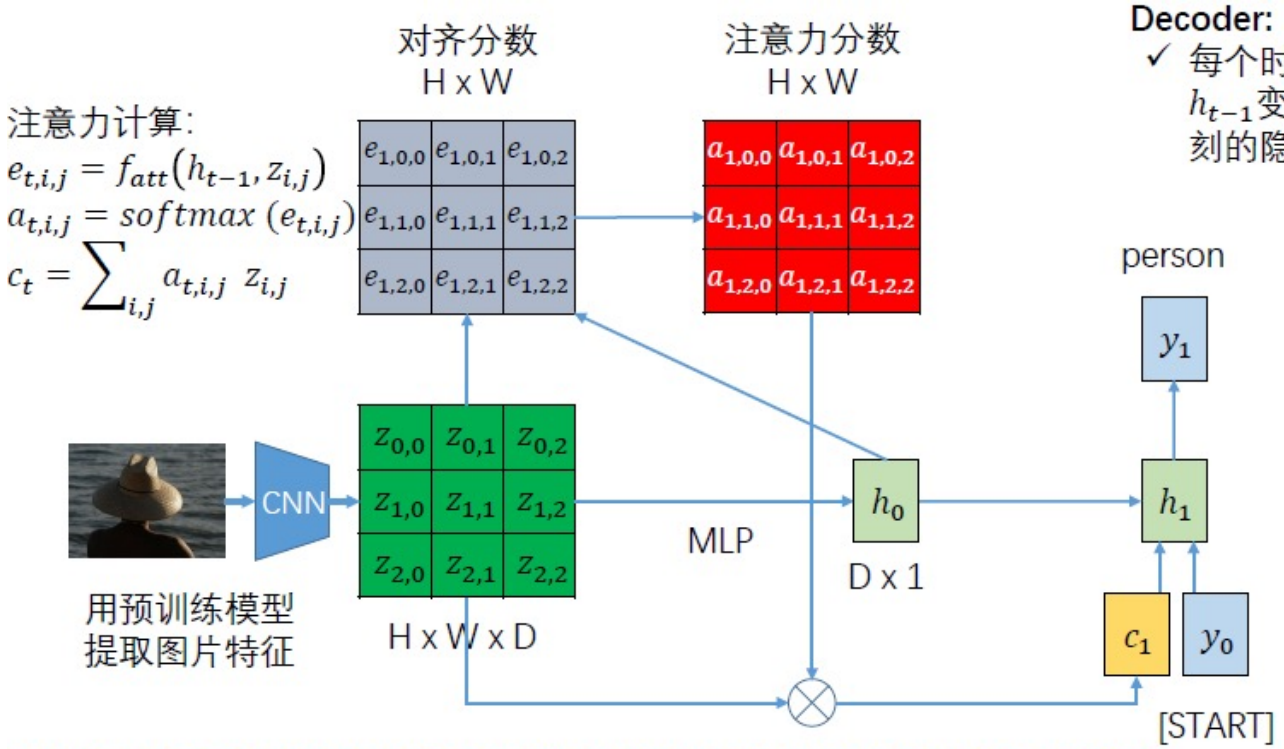


Image Caption (有注意力)



Xu, K., Ba, J., et al. Show, attend and tell: Neural image caption generation with visual attention. ICML 2015.

Image Caption (有注意力)

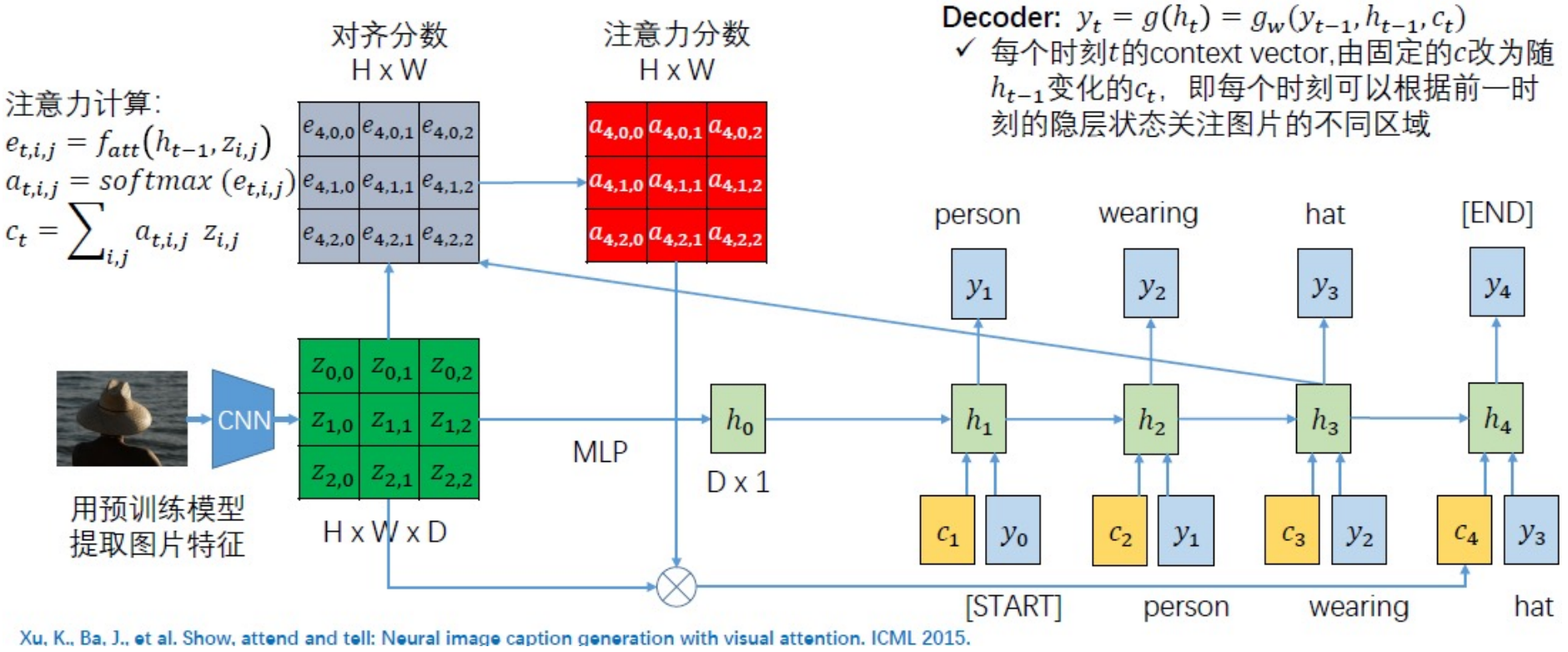


Decoder: $y_t = g(h_t) = g_w(y_{t-1}, h_{t-1}, c_t)$

✓ 每个时刻t的context vector,由固定的c改为随h_{t-1}变化的c_t，即每个时刻可以根据前一时刻的隐层状态关注图片的不同区域

Xu, K., Ba, J., et al. Show, attend and tell: Neural image caption generation with visual attention. ICML 2015.

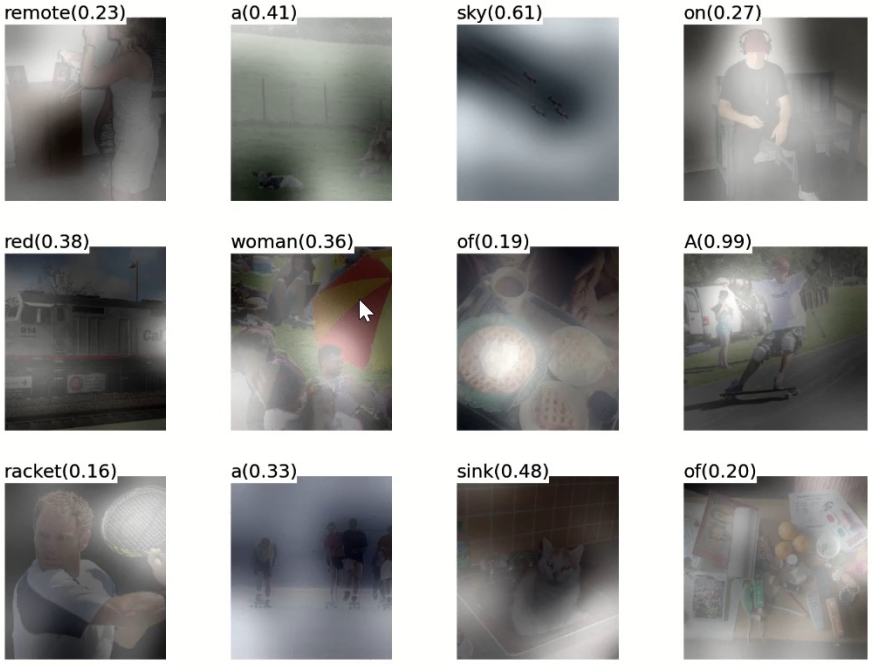
Image Caption (有注意力)



Xu, K., Ba, J., et al. Show, attend and tell: Neural image caption generation with visual attention. ICML 2015.

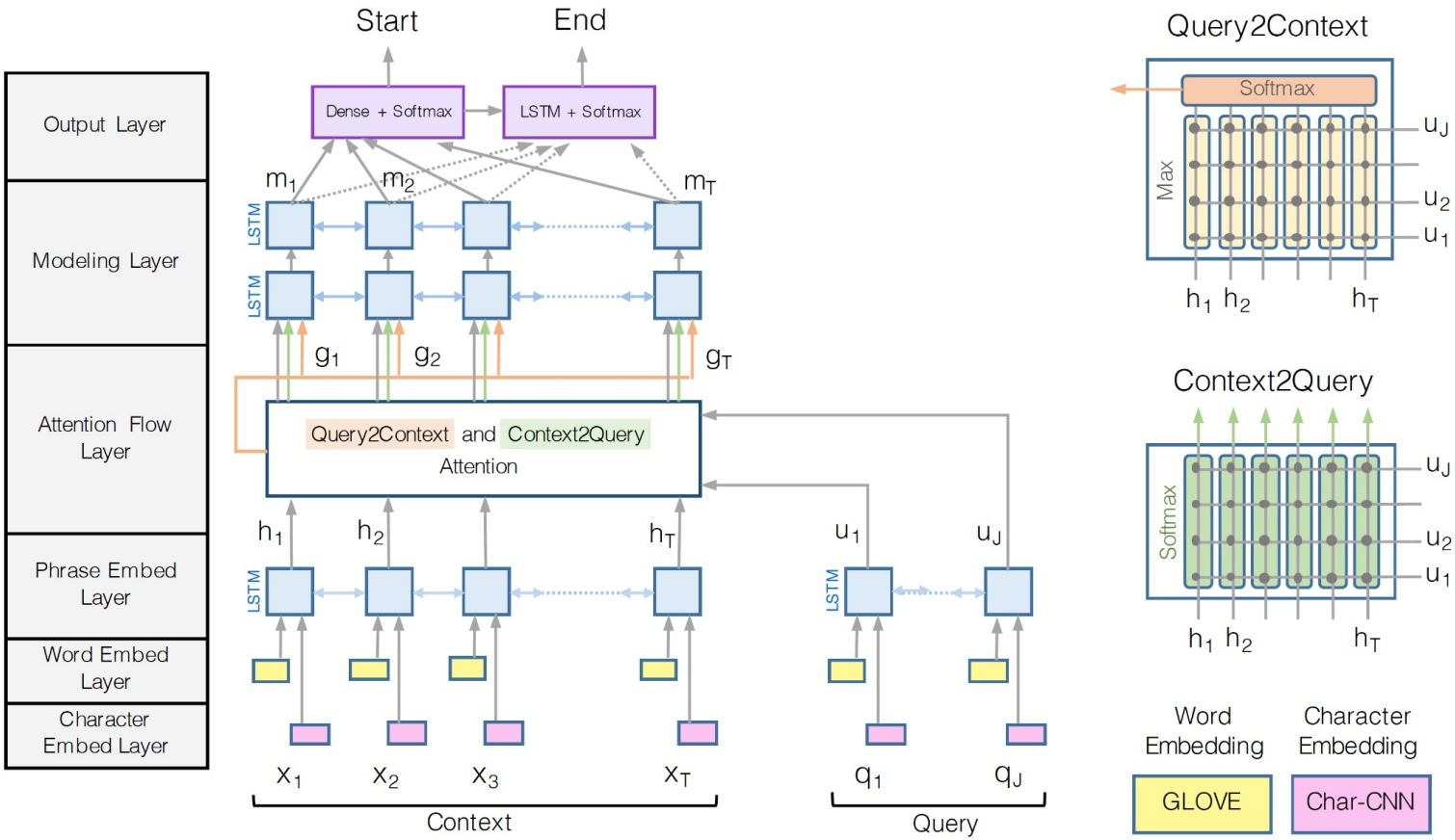
Image Caption (有注意力)

http://kelvinxu.github.io/projects/capgen.html



阅读理解

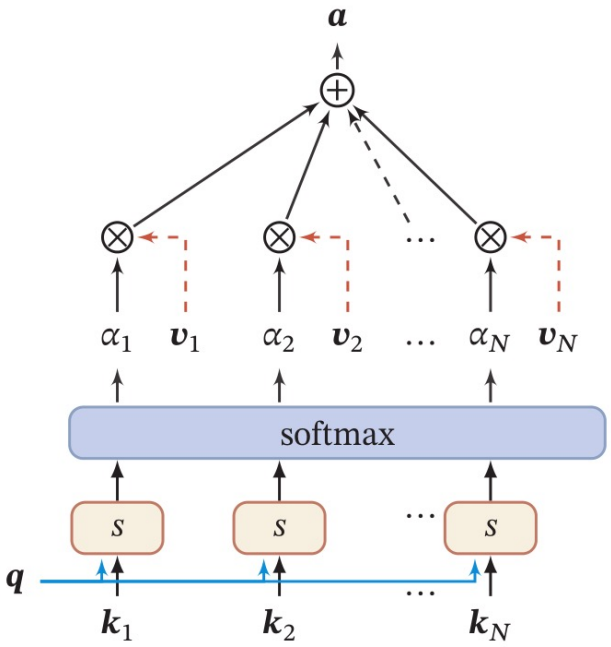
https://allenai.github.io/bi-att-flow/



注意力机制的变体

- 硬性注意力（hard attention）
- 键值对注意力（ key-value pair attention）

$$\text{att}(\boldsymbol{X}, \boldsymbol{q}) = \boldsymbol{x}_{\hat{n}}, \qquad \hat{n} = \arg\max_{n=1}^{N} \alpha_n.$$



用$(K, V) = [(\boldsymbol{k_1}, \boldsymbol{v_1}), \cdots, (\boldsymbol{k_N}, \boldsymbol{v_N})]$表示N个输入信息

$$\begin{aligned}\text{att}\Big((\boldsymbol{K}, \boldsymbol{V}), \boldsymbol{q}\Big) &= \sum_{n=1}^{N} \alpha_n \boldsymbol{v}_n, \\ &= \sum_{n=1}^{N} \frac{\exp\Big(s(\boldsymbol{k}_n, \boldsymbol{q})\Big)}{\sum_j \exp\Big(s(\boldsymbol{k}_j, \boldsymbol{q})\Big)} \boldsymbol{v}_n\end{aligned}$$

注意力机制的变体

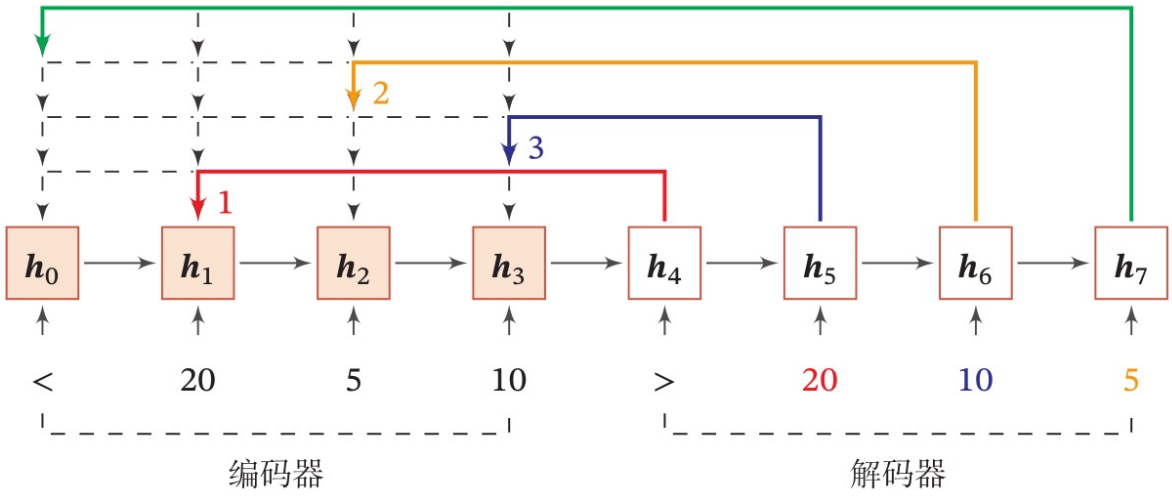
- 多头注意力（multi-head attention）
  - 利用多个查询 $\boldsymbol{Q} = [\boldsymbol{q}_1, \cdots, \boldsymbol{q}_M]$，来并行地从输入信息中选取多组信息．每个注意力关注输入信息的不同部分．

$$\text{att}\Big((\boldsymbol{K}, \boldsymbol{V}), \boldsymbol{Q}\Big) = \text{att}\Big((\boldsymbol{K}, \boldsymbol{V}), \boldsymbol{q}_1\Big) \oplus \cdots \oplus \text{att}\Big((\boldsymbol{K}, \boldsymbol{V}), \boldsymbol{q}_M\Big)$$

- 结构化注意力（structural attention）
  - 对于层次结构的注意力
  - 对于图结构的注意力

指针网络（ Pointer Network ）

- 我们可以只利用注意力机制中的第一步，将注意力分布作为一个软性的指针（pointer）来指出相关信息的位置。



$$p(c_{1:M}|\boldsymbol{x}_{1:N}) = \prod_{m=1}^{M} p(c_m|c_{1:(m-1)}, \boldsymbol{x}_{1:N})$$

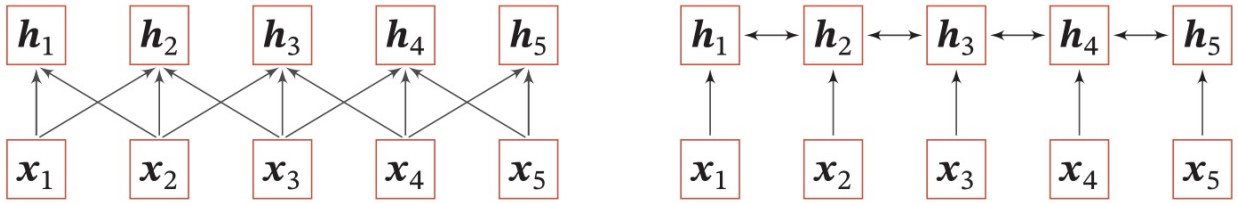
$$p(c_m|c_{1:(m-1)}, \boldsymbol{x}_{1:N}) = \text{softmax}(s_{m,n}),$$

$$s_{m,n} = \boldsymbol{v}^{\mathrm{T}} \tanh(\boldsymbol{W}\boldsymbol{x}_n + \boldsymbol{U}\boldsymbol{h}_m), \forall n \in [1, N],$$

自注意力模型

自注意力模型

- 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。

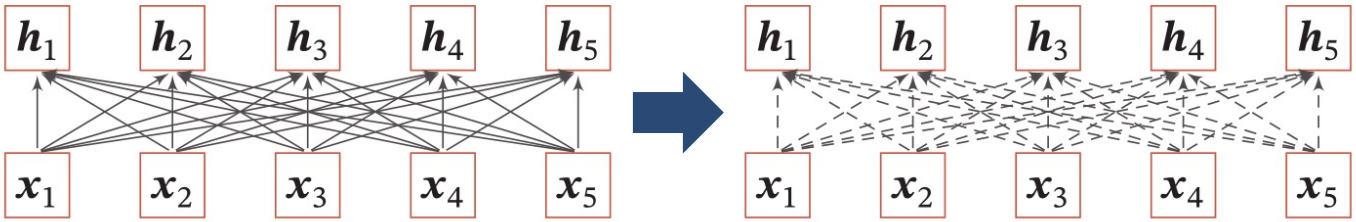


(a) 卷积网络　　(b) 双向循环网络

只建模了输入信息的局部依赖关系

自注意力模型

- 如何建立非局部（Non-local）的依赖关系
  - 全连接？

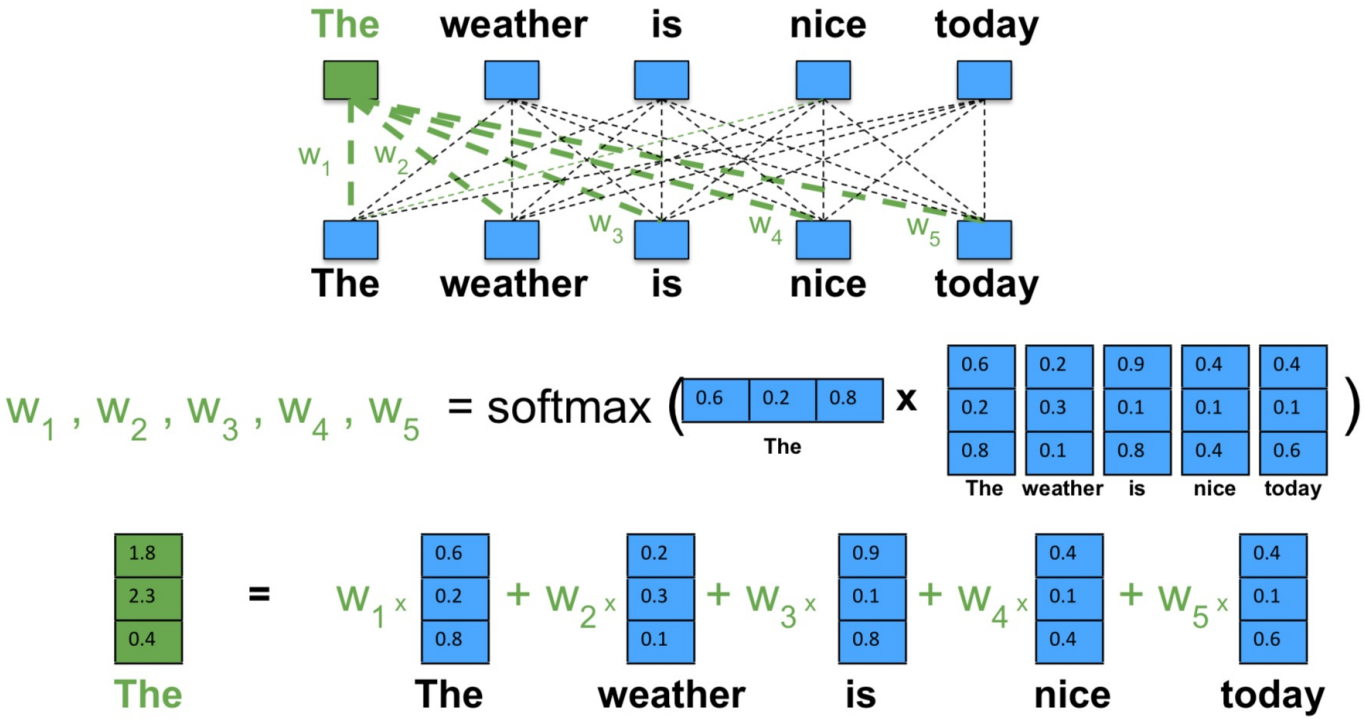


(a) 全连接模型　　(b) 自注意力模型

无法处理变长问题

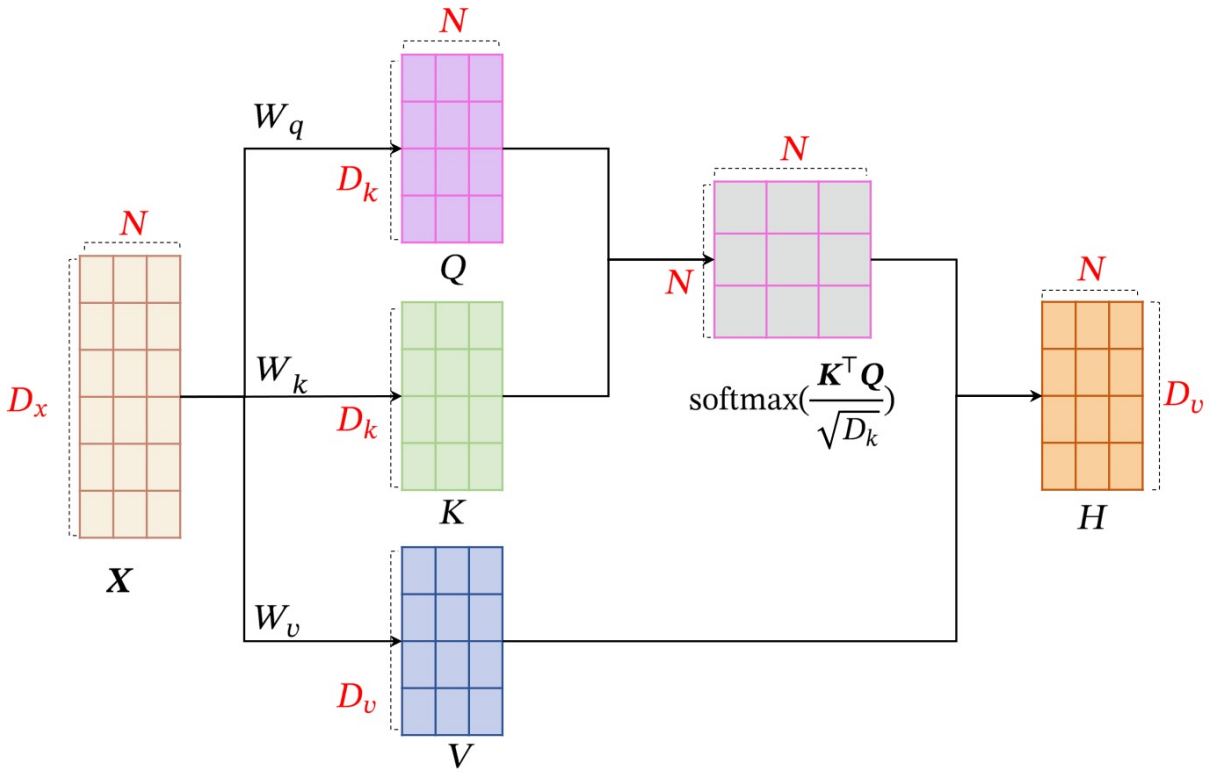
连接权重α_{ij}由注意力机制动态生成

自注意力示例



图片来源：http://fuyw.top/NLP_02_QANet/

QKV模式（Query-Key-Value）



自注意力模型

- 输入序列为$X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in R^{D_x \times N}$
- 首先生成三个向量序列

$$\boldsymbol{Q} = \boldsymbol{W}_q \boldsymbol{X} \in \mathbb{R}^{D_k \times N},$$
$$\boldsymbol{K} = \boldsymbol{W}_k \boldsymbol{X} \in \mathbb{R}^{D_k \times N},$$
$$\boldsymbol{V} = \boldsymbol{W}_v \boldsymbol{X} \in \mathbb{R}^{D_v \times N},$$

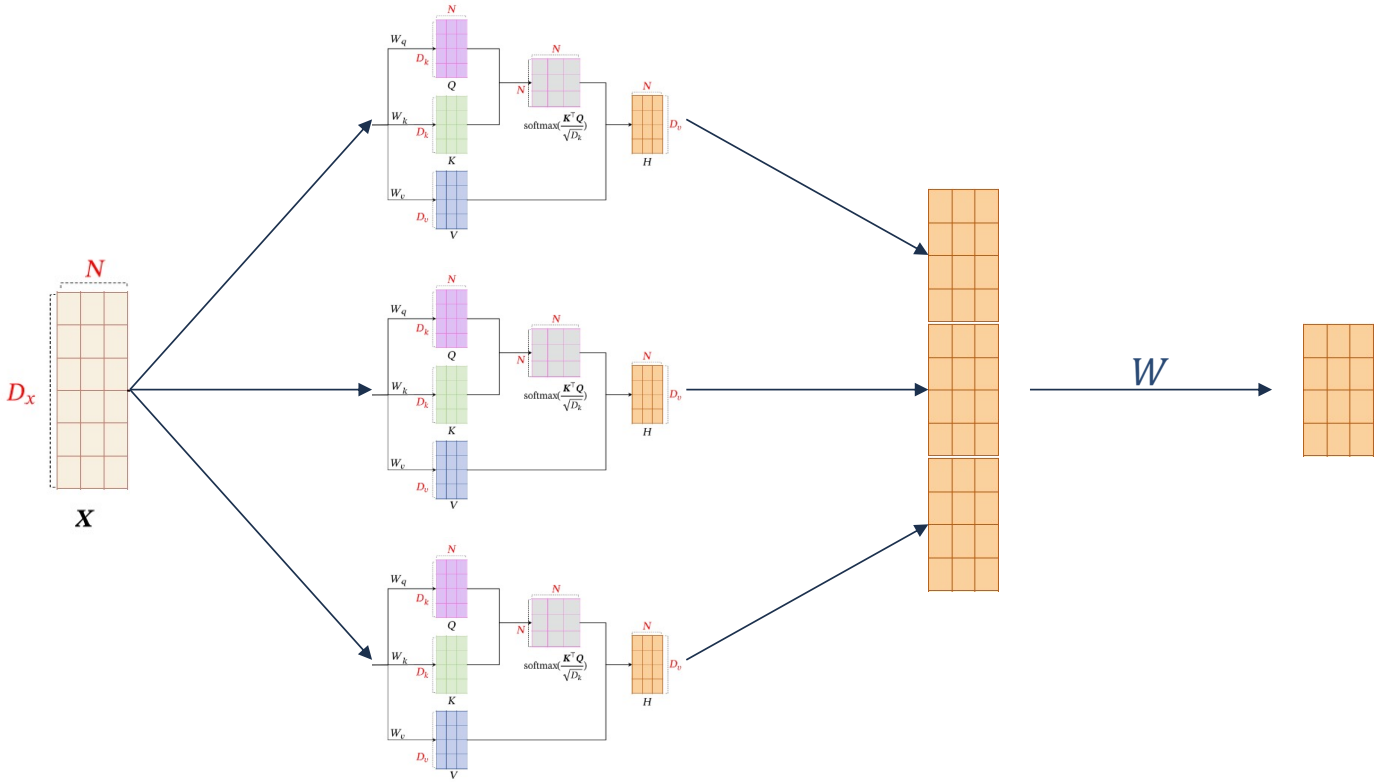
- 计算$\boldsymbol{h}_n$

$$\boldsymbol{h}_n = \text{att}\Big((\boldsymbol{K}, \boldsymbol{V}), \boldsymbol{q}_n\Big)$$

- 如果使用缩放点积来作为注意力打分函数，输出向量序列可以简写为

$$\boldsymbol{H} = \boldsymbol{V}\,\text{softmax}\Big(\frac{\boldsymbol{K}^{\top}\boldsymbol{Q}}{\sqrt{D_k}}\Big),$$

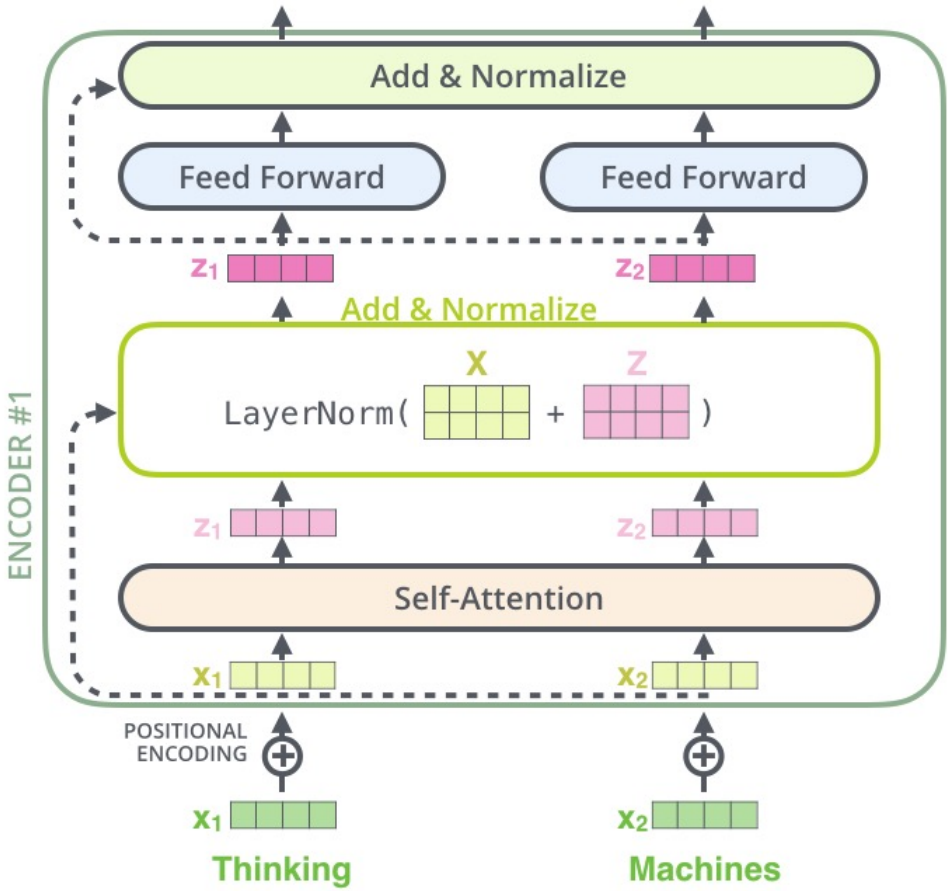
多头（multi-head）自注意力模型



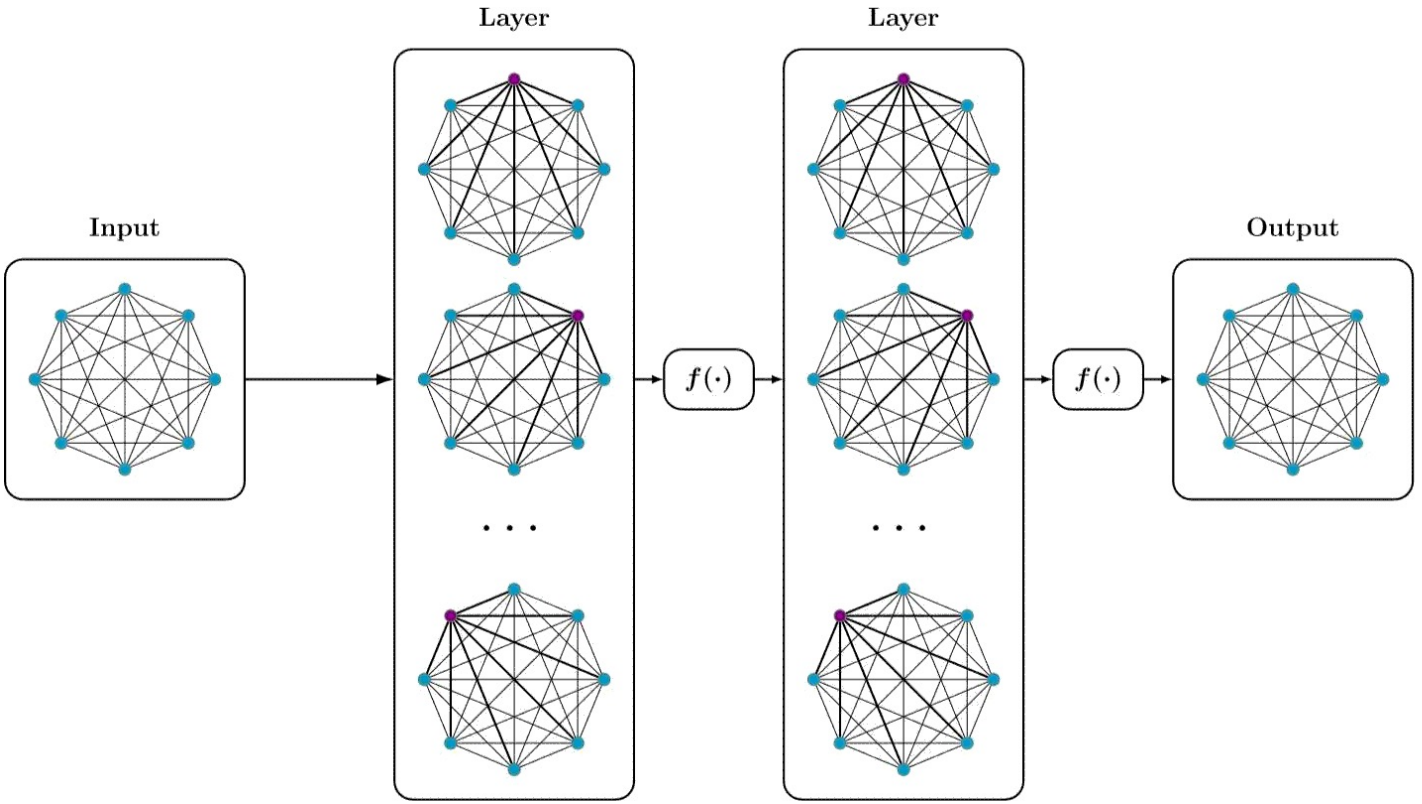
Transformer Encoder

图片来源：http://jalammar.github.io/illustrated-transformer/

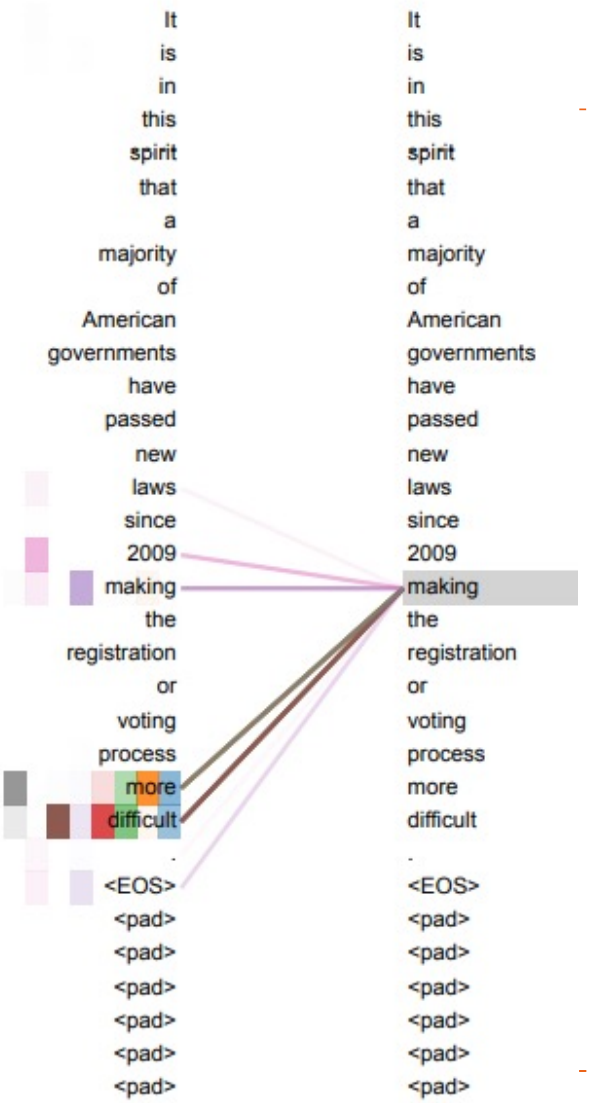
- 仅仅自注意力还不够
- 其它操作
  - 位置编码
  - 层归一化
  - 直连边
  - 逐位的FNN



Transformer



Transformer



复杂度分析

输入和输出的特征维度一样的情况下

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	$O(Ld^2)$	$O(L)$	$O(L)$
Self-Attention	$O(L^2d)$	$O(1)$	$O(1)$

k卷积核大小　L序列长度　d维度

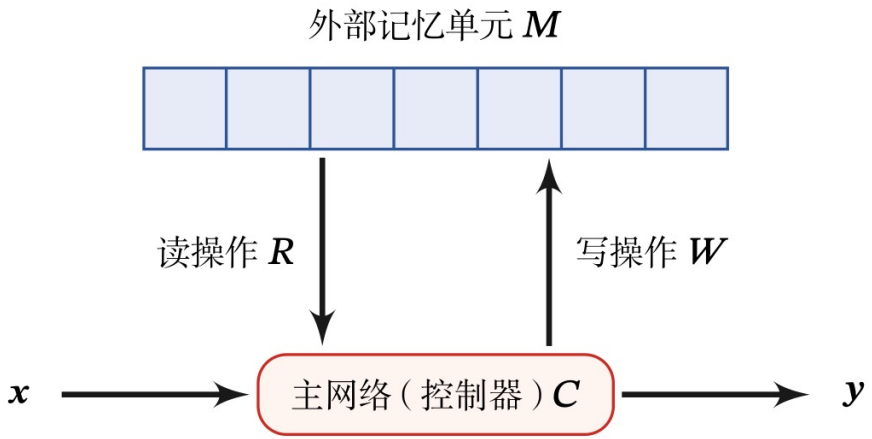
记忆增强网络

不严格的类比

记忆周期	计算机	人脑	神经网络
短期	寄存器	短期记忆	状态（神经元活性）
中期	内存	工作记忆	外部记忆
长期	外存	长期记忆	可学习参数
存储方式	随机寻址	内容寻址	内容寻址为主

外部记忆

- 记忆增强神经网络（Memory Augmented Neural Network）
  - 主网络
  - 外部记忆
  - 读写操作



外部记忆

- 外部记忆定义为矩阵$M \in R^{d \times k}$
  - k 是记忆片段的数量，d 是每个记忆片段的大小
- 外部记忆类型
  - 只读
    - Memory Network
    - RNN中的h_t
  - 可读写
    - NTM

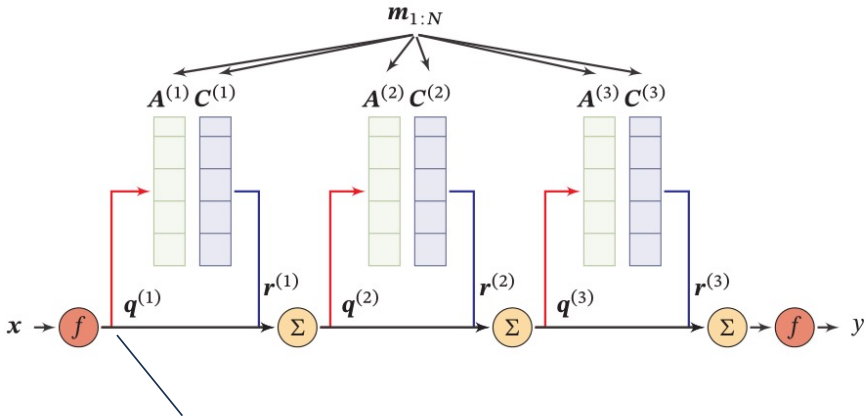
- 如何读写？

注意力机制

记忆网络(Memory Network)

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-To-End Memory Networks, 1–11. http://arxiv.org/abs/1503.08895

外部记忆



输入

输出

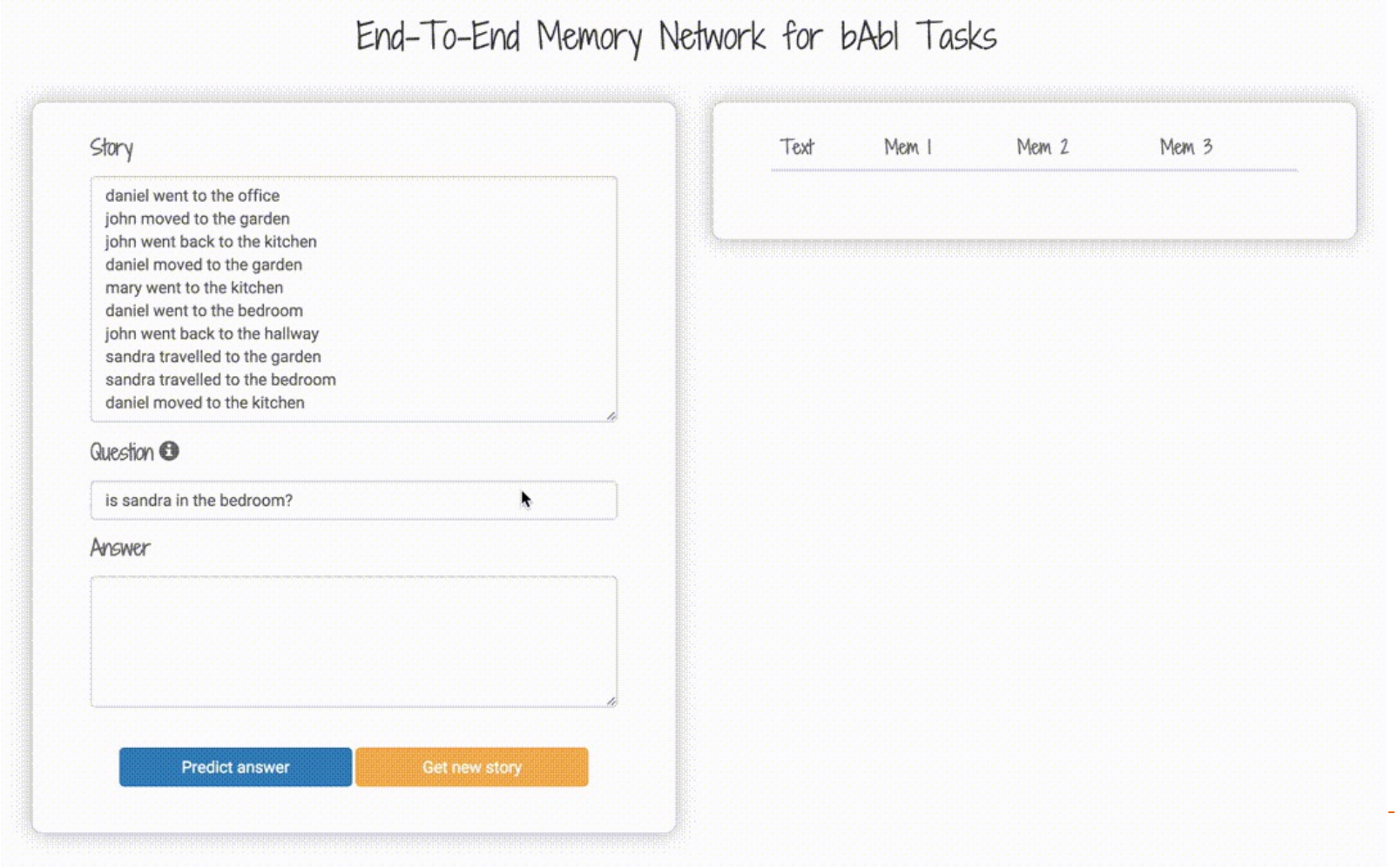
主网络生成

$$\boldsymbol{q}^{(k)} = \boldsymbol{r}^{(k-1)} + \boldsymbol{q}^{(k-1)},$$

$$\boldsymbol{r}^{(k)} = \sum_{n=1}^{N} \mathrm{softmax}((\boldsymbol{a}_n^{(k)})^{\mathrm{T}} \boldsymbol{q}^{(k)}) \boldsymbol{c}_n^{(k)}.$$

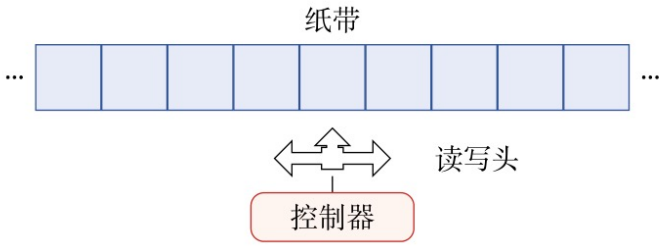
$$\boldsymbol{y} = f(\boldsymbol{q} + \boldsymbol{r}),$$

例子：阅读理解


End-To-End Memory Network for bAbI Tasks
Story
daniel went to the office
john moved to the garden
john went back to the kitchen
daniel moved to the garden
mary went to the kitchen
daniel went to the bedroom
john went back to the hallway
sandra travelled to the garden
sandra travelled to the bedroom
daniel moved to the kitchen
Question
is sandra in the bedroom?
Answer
Predict answer
Get new story
Text
Mem 1
Mem 2
Mem 3

神经图灵机

- 图灵机
  - 一种抽象数学模型，可以用来模拟任何可计算问题。



神经图灵机

- 组件
  - 控制器
  - 外部记忆
  - 读写操作
- 整个架构可微分

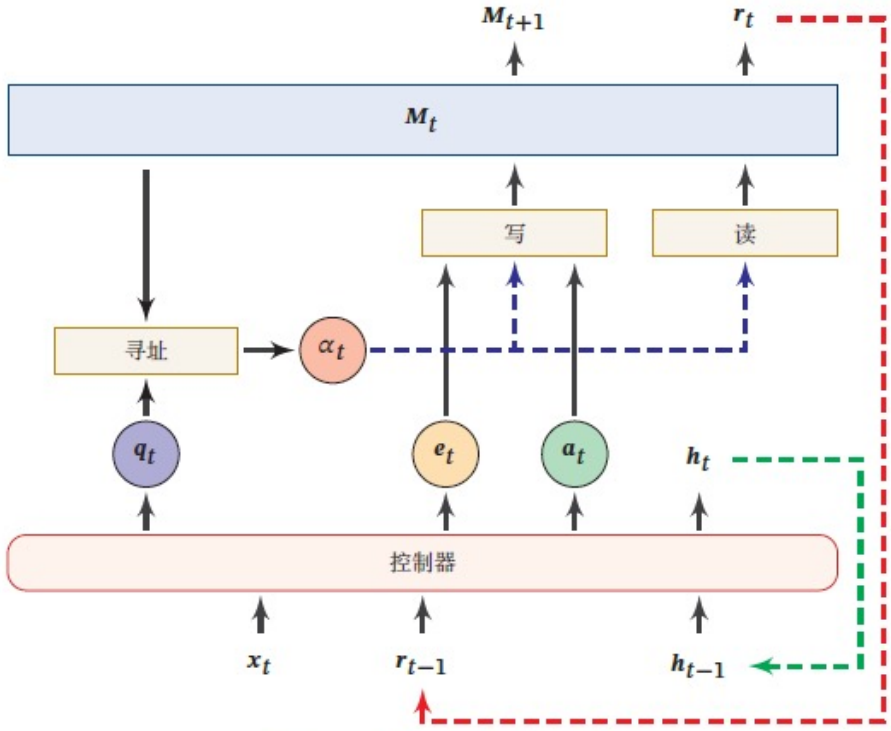


图 8.9 神经图灵机示例

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural Turing Machines. Arxiv, 1–26. http://arxiv.org/abs/1410.5401