

《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)

#### ▶ 基于神经动力学的联想记忆(Hopfield Network)

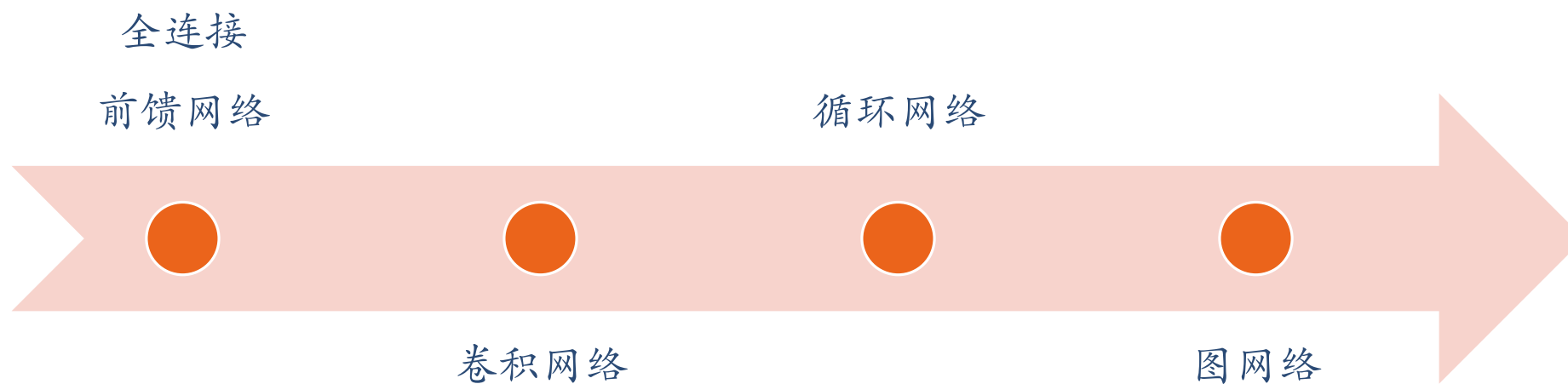
## 参考资料

---

- ▶ 《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

---

- ▶ 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



## 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  $Y = f(X)$

汇聚 (pooling)

### ▶ 自上而下

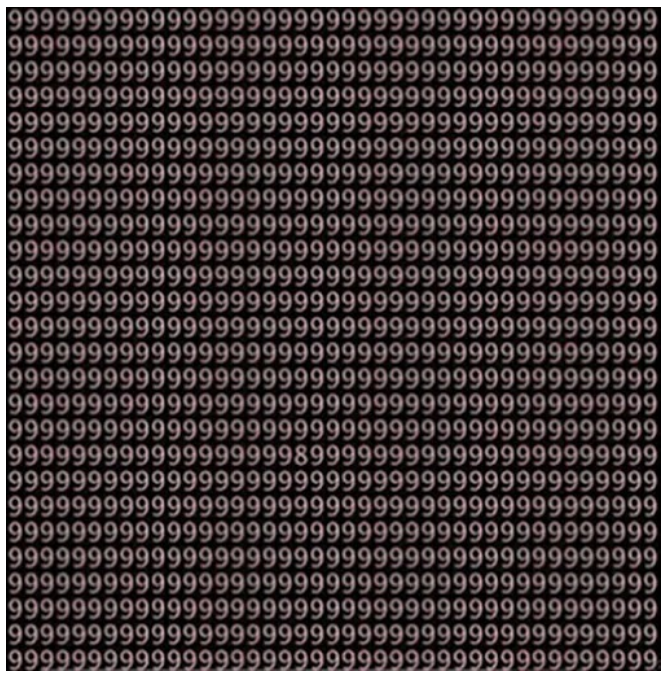
- ▶ 有意识的注意力  $Y = f(X, q)$

会聚 (focus)

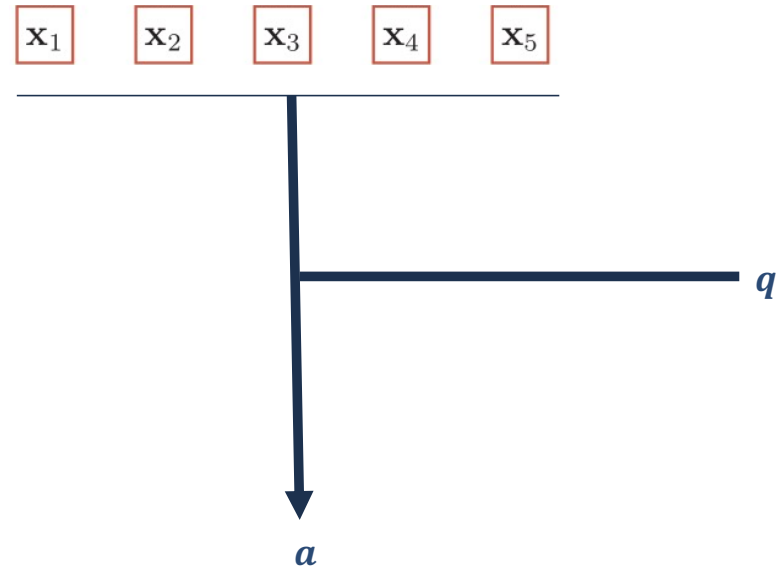


应用到机器学习

# 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 $\alpha$ ,

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

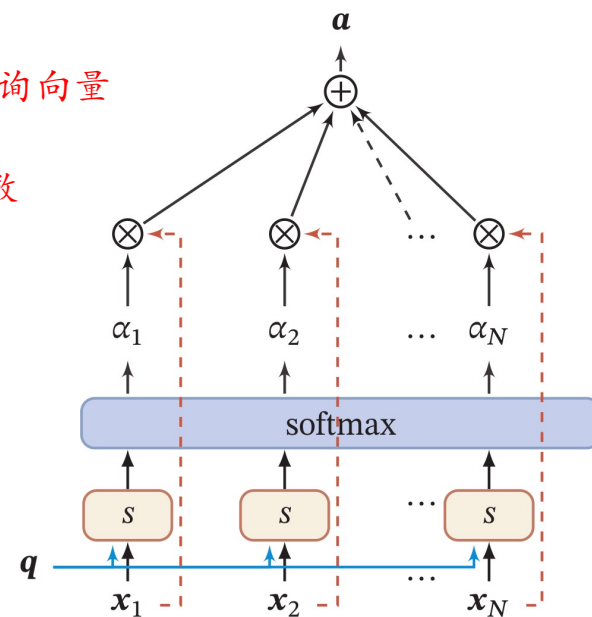
$\mathbf{X}$ 为输入向量,  $\mathbf{q}$ 为查询向量

$s(\mathbf{x}_n, \mathbf{q})$  打分函数

- ▶ 根据 $\alpha$ 来计算输入信息的加权平均。

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

$z \in [1, N]$ 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

$$s(\mathbf{x}, \mathbf{q}) = \mathbf{v}^\top \tanh(\mathbf{W}\mathbf{x} + \mathbf{U}\mathbf{q}),$$

点积模型

$$s(\mathbf{x}, \mathbf{q}) = \mathbf{x}^\top \mathbf{q},$$

缩放点积模型

$$s(\mathbf{x}, \mathbf{q}) = \frac{\mathbf{x}^\top \mathbf{q}}{\sqrt{D}},$$

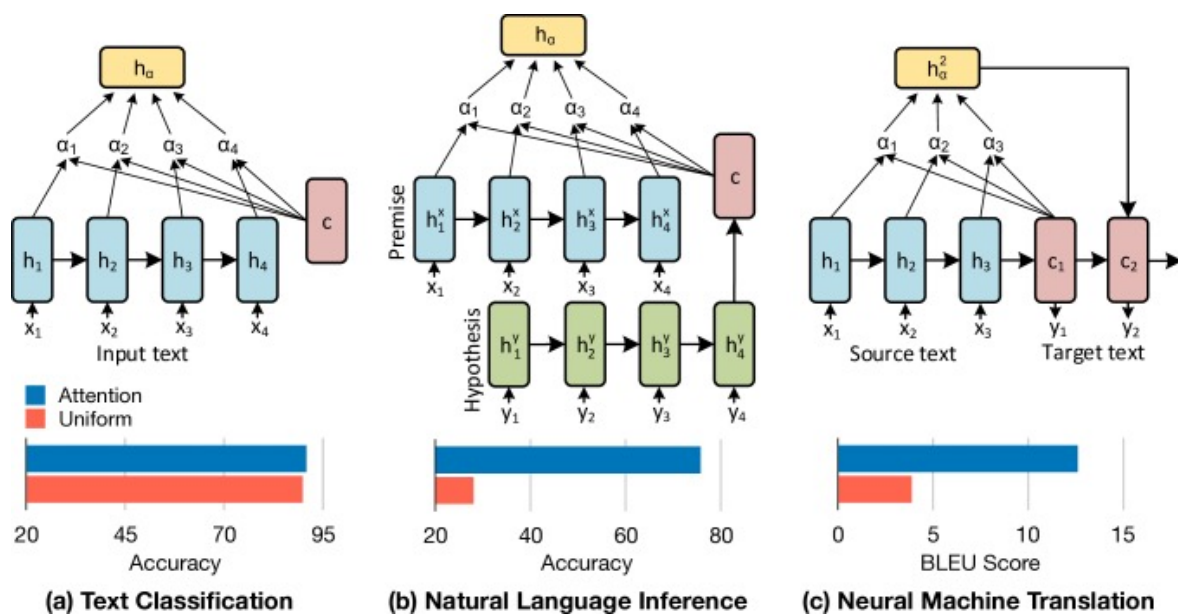
相比起点积模型，缓解梯度消失的问题

双线性模型

$$s(\mathbf{x}, \mathbf{q}) = \mathbf{x}^\top \mathbf{W}\mathbf{q},$$

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>

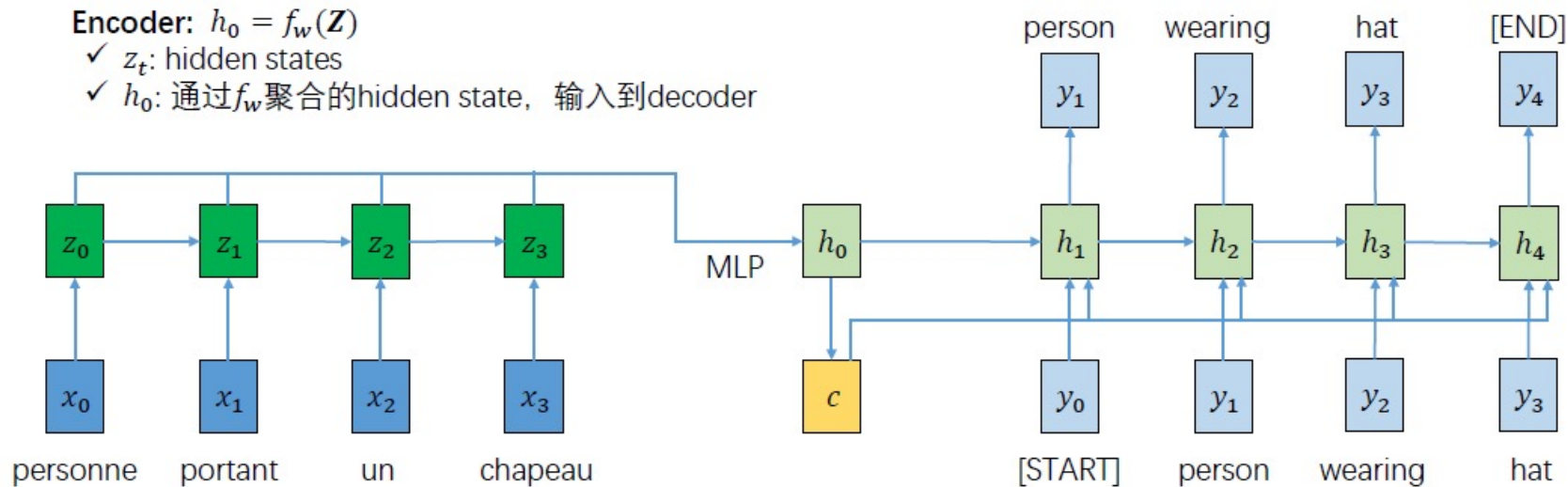


## 机器翻译(无注意力)

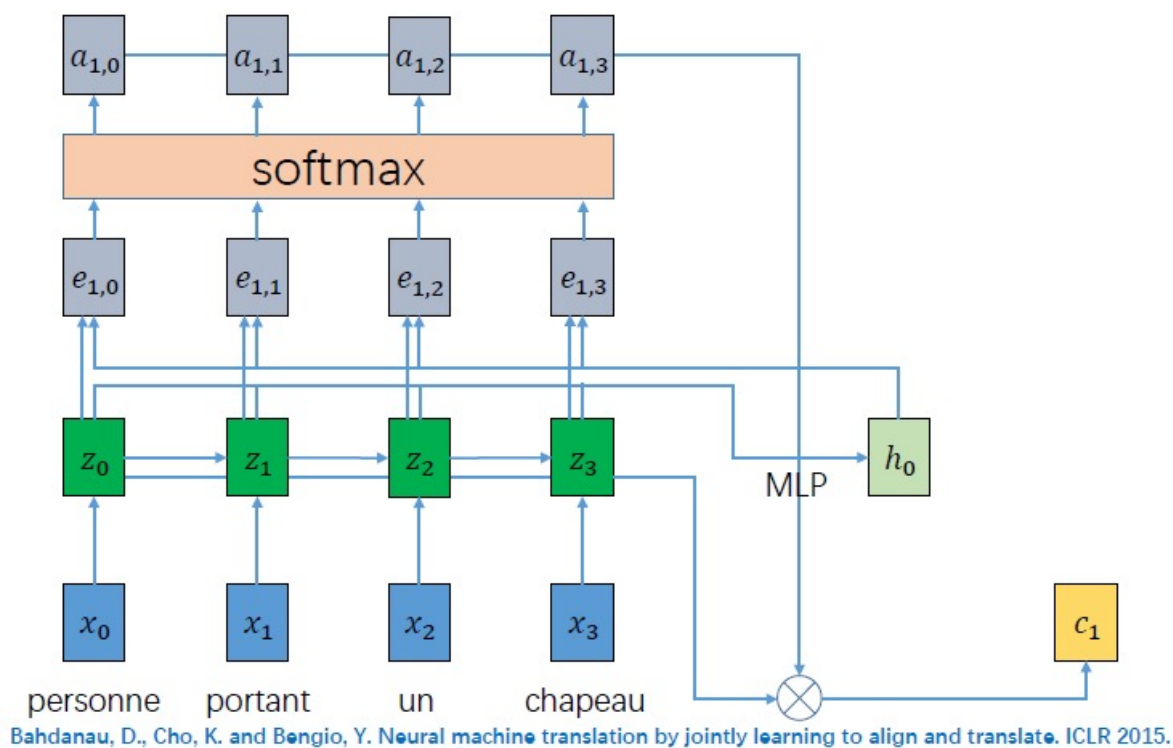
Input: 文本  $x = x_1, x_2, \dots, x_T$   
Output: 文本  $y = y_1, y_2, \dots, y_T$

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

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



## 机器翻译(有注意力)



计算对齐分数:

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

$f_{att}$ 为MLP

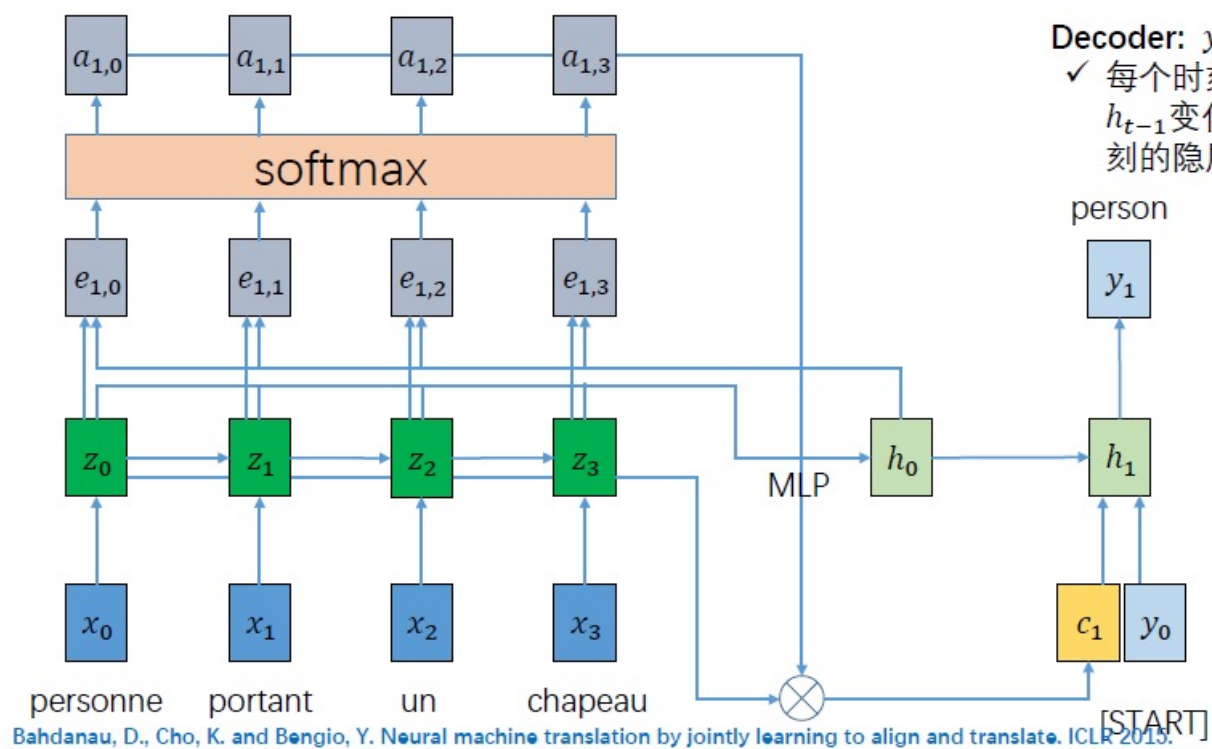
计算注意力分数:

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

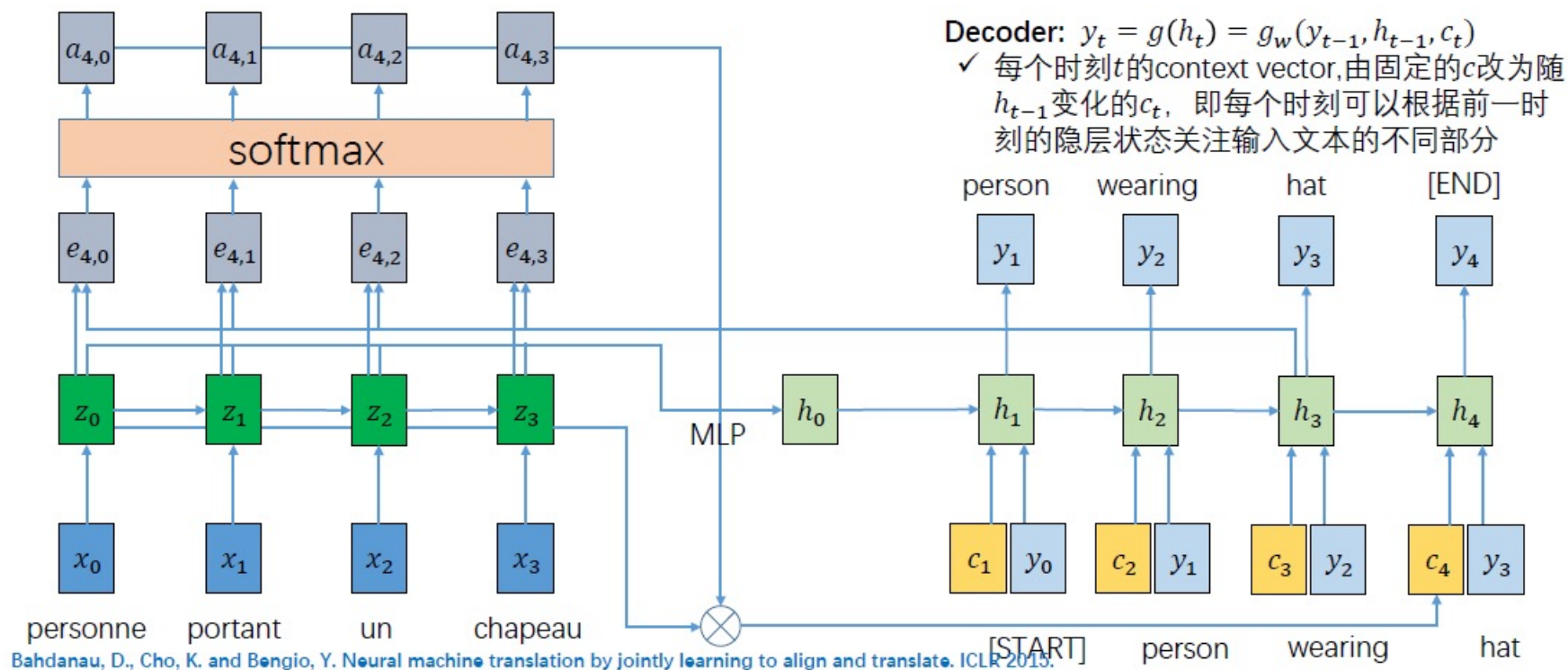
计算context vector:

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

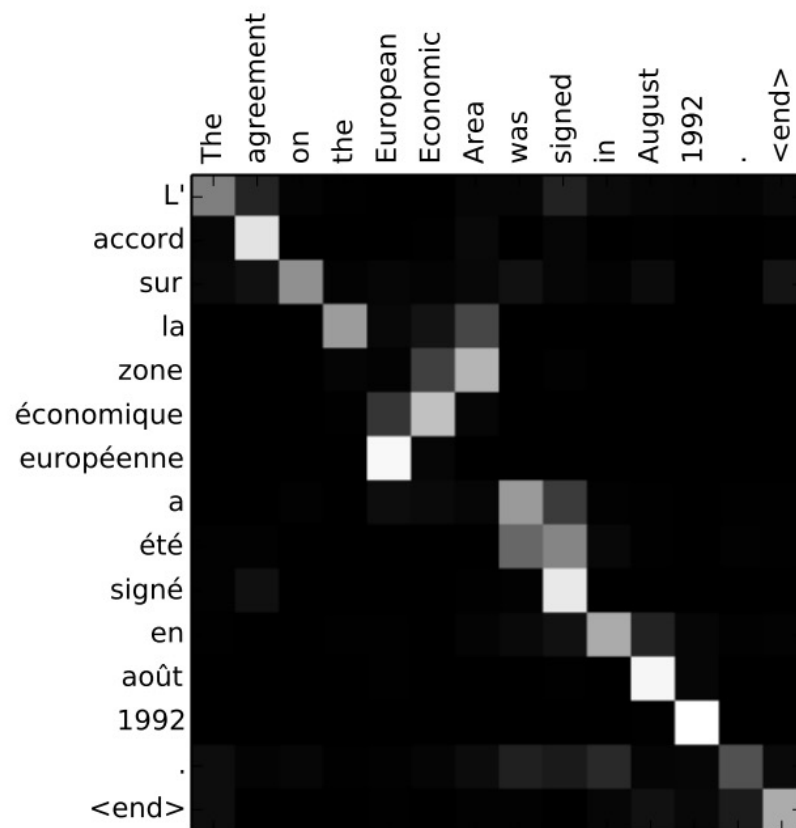
## 机器翻译(有注意力)



## 机器翻译(有注意力)

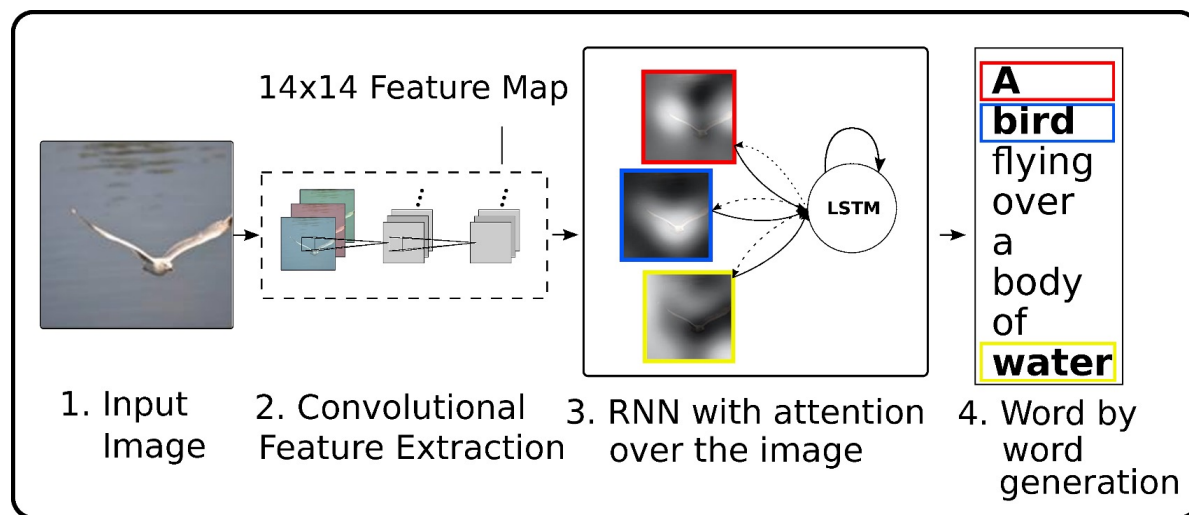


## 机器翻译(有注意力)



# Image Caption

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



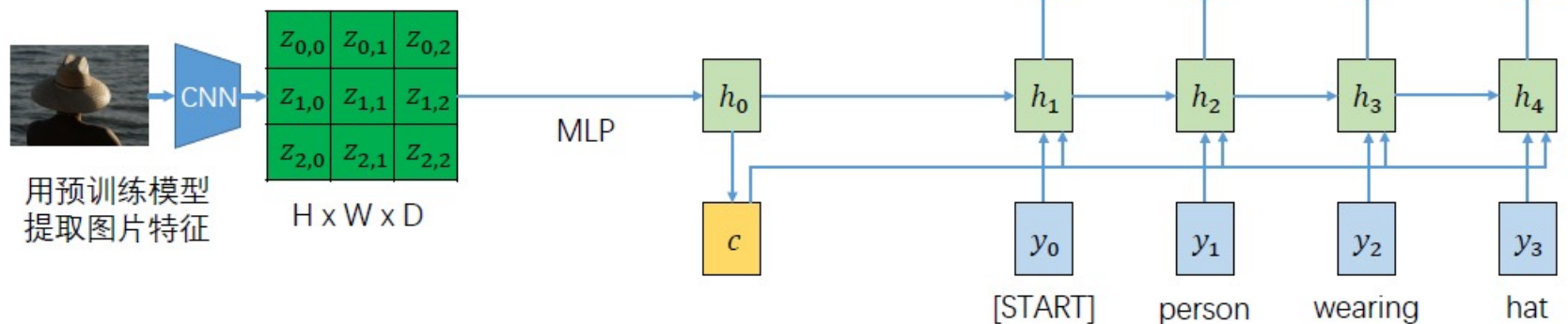
# Image Caption (无注意力)

Input: 图片  $I$   
Output: 文本  $y = y_1, y_2, \dots, y_T$

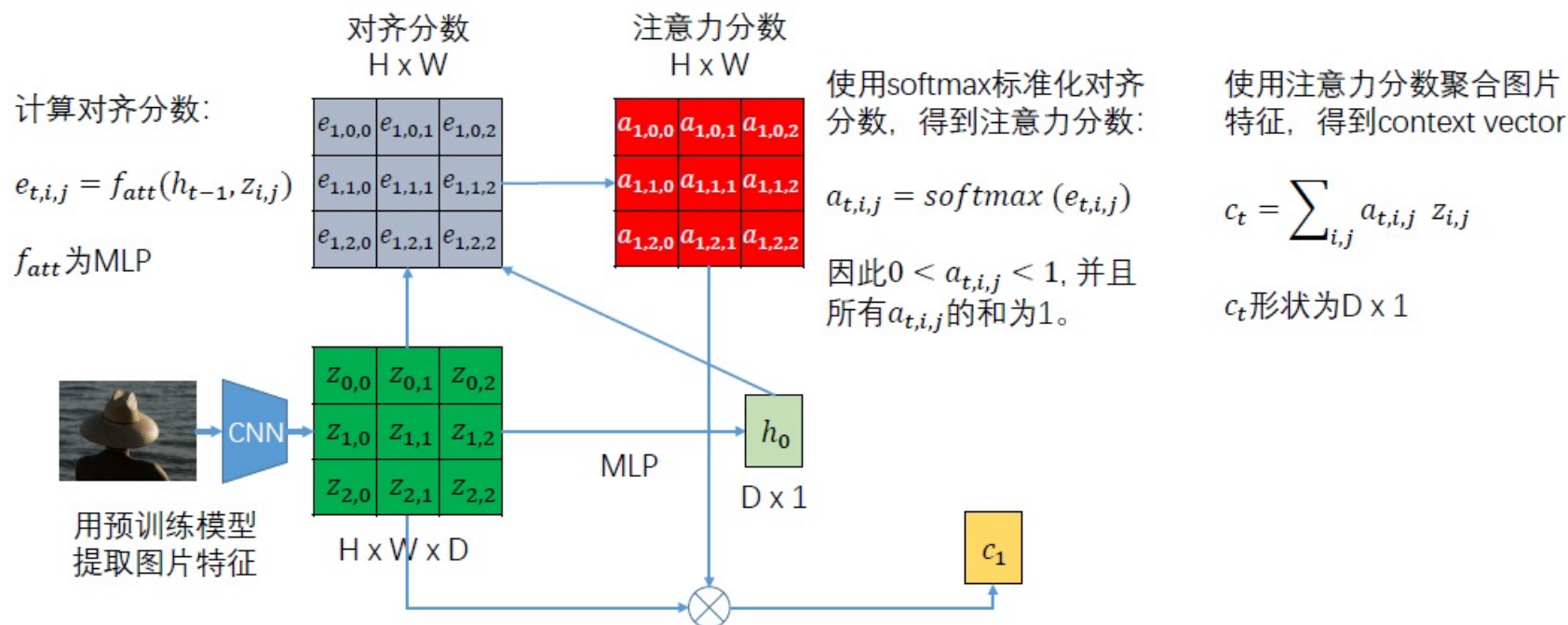
Encoder:  $h_0 = f_w(Z)$   
✓  $Z$ : CNN提取的图片特征  
✓  $f_w()$ : Multi-Layer Perceptron (若干层FC)

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



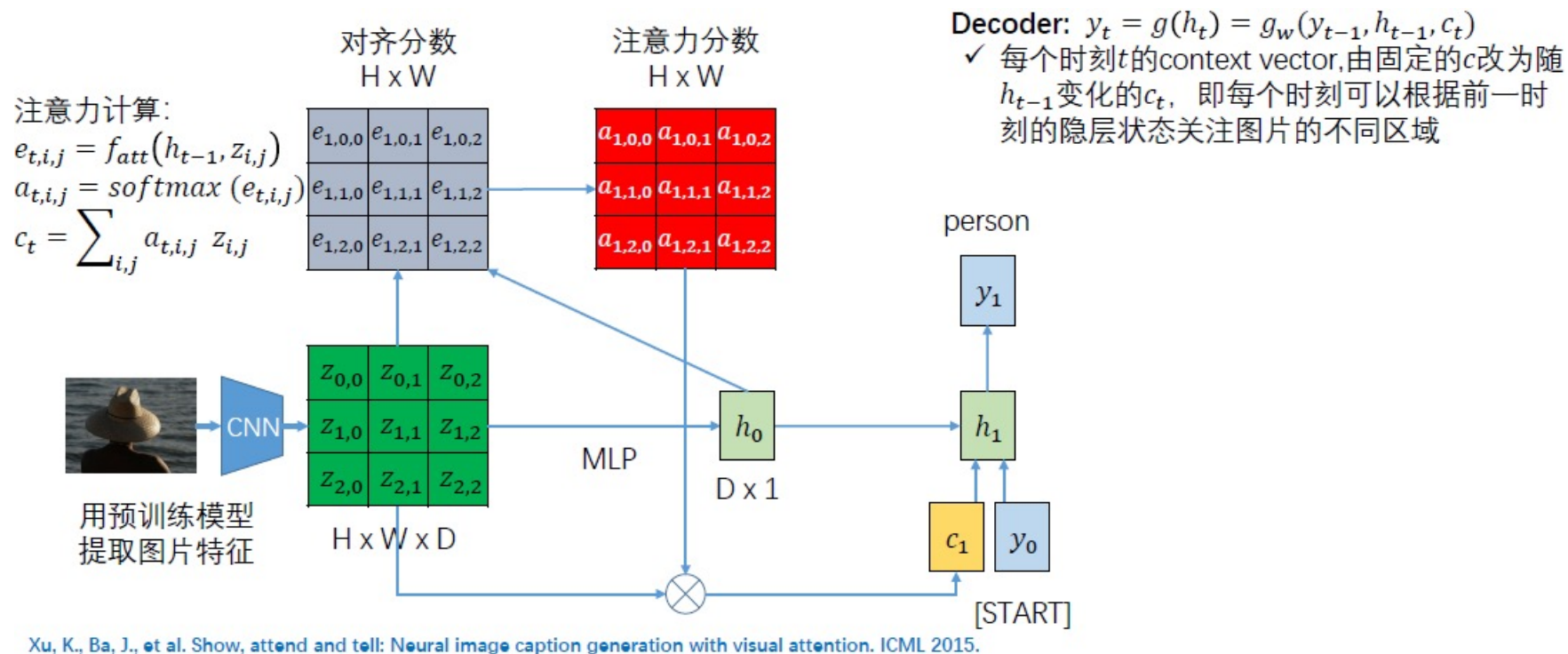
## Image Caption (有注意力)



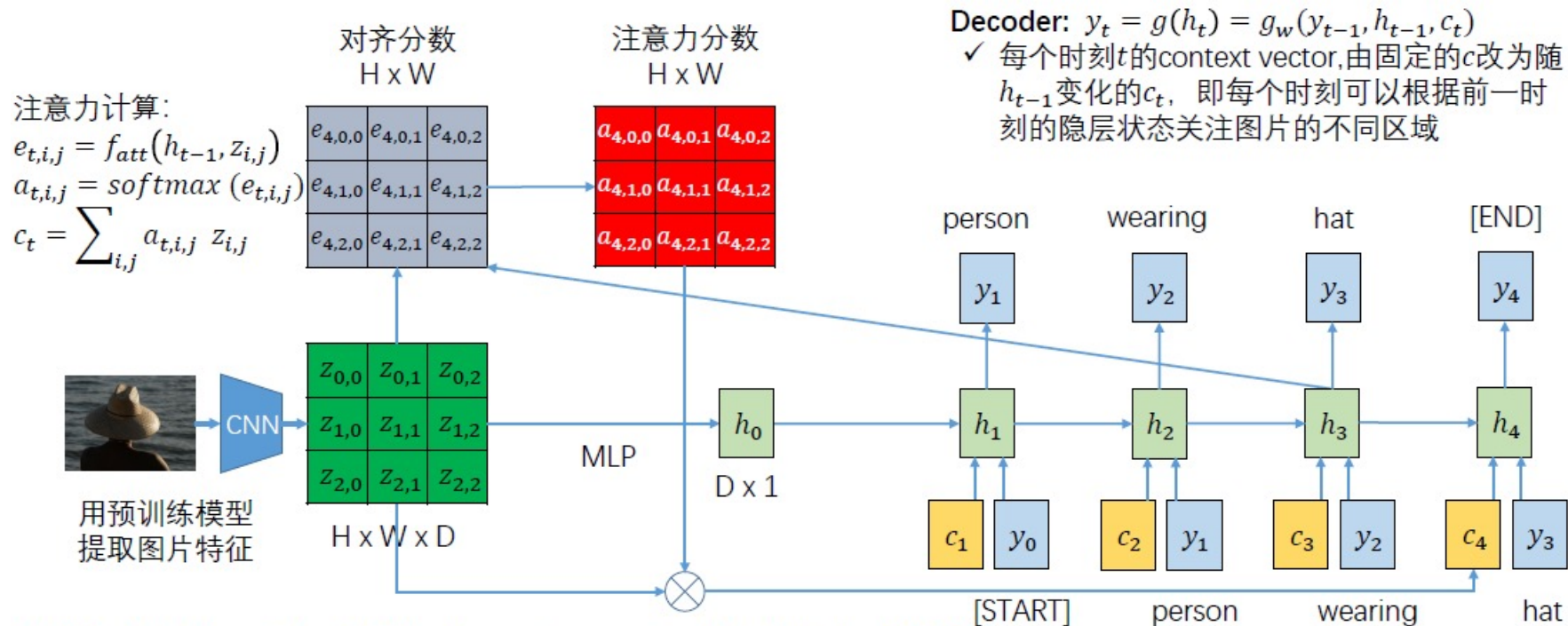
Xu, K., Ba, J., et al. Show, attend and tell: Neural image caption generation with visual attention. ICML 2015.



## Image Caption (有注意力)



# 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>

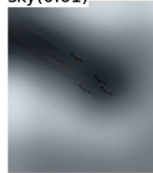
remote(0.23)



a(0.41)



sky(0.61)



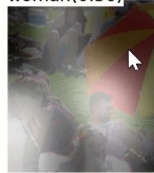
on(0.27)



red(0.38)



woman(0.36)



of(0.19)



A(0.99)



racket(0.16)



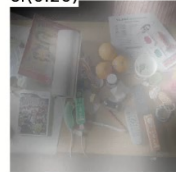
a(0.33)



sink(0.48)

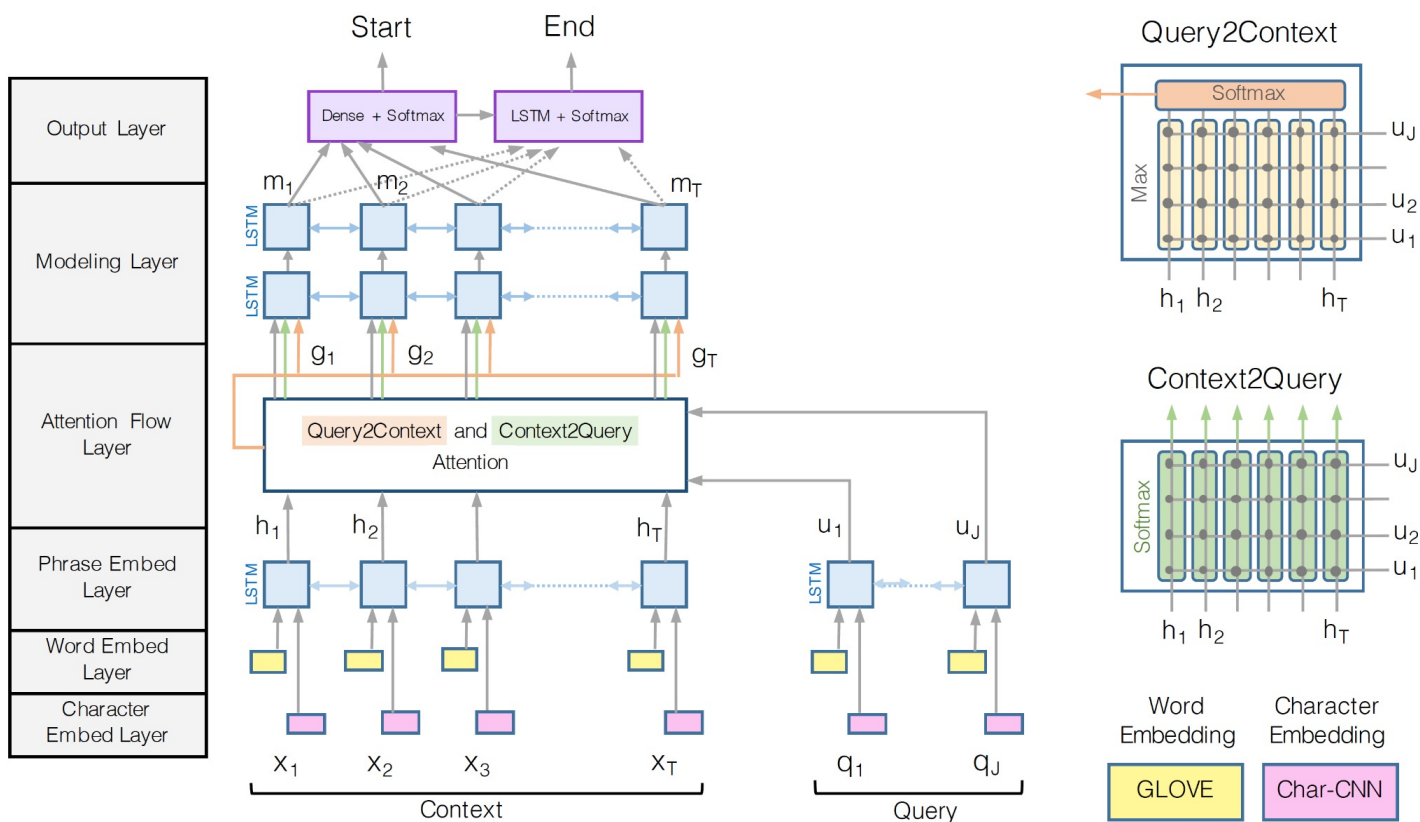


of(0.20)



# 阅读理解

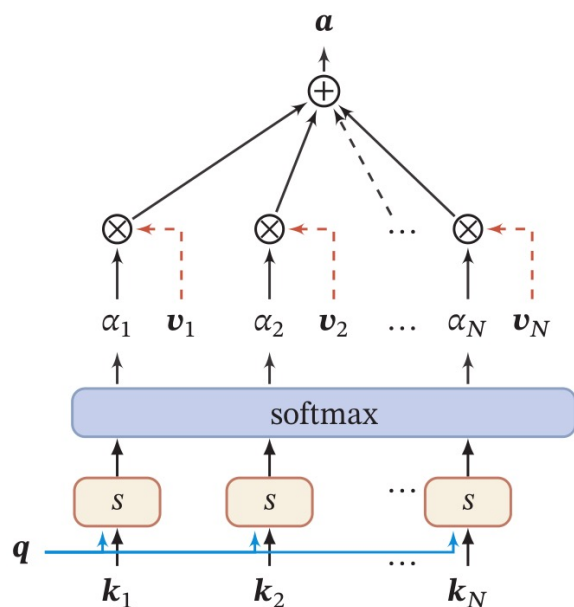
<https://allenai.github.io/bi-att-flow/>



## 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)

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



用  $(\mathbf{K}, \mathbf{V}) = [(\mathbf{k}_1, \mathbf{v}_1), \dots, (\mathbf{k}_N, \mathbf{v}_N)]$  表示  $N$  个输入信息

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

## 注意力机制的变体

---

### ▶ 多头注意力 (multi-head attention)

- ▶ 利用多个查询  $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_M]$ ，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。

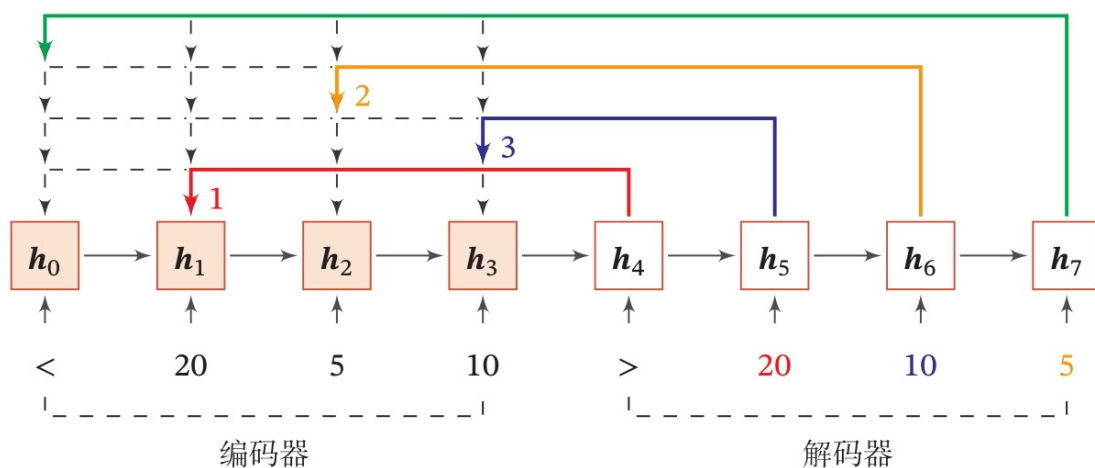
$$\text{att}((\mathbf{K}, \mathbf{V}), \mathbf{Q}) = \text{att}((\mathbf{K}, \mathbf{V}), \mathbf{q}_1) \oplus \dots \oplus \text{att}((\mathbf{K}, \mathbf{V}), \mathbf{q}_M)$$

### ▶ 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。



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

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

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

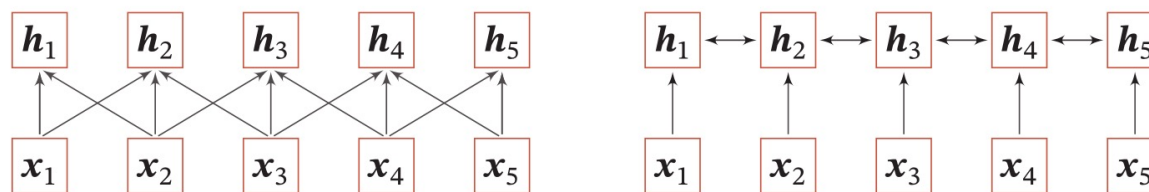


## 自注意力模型



## 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

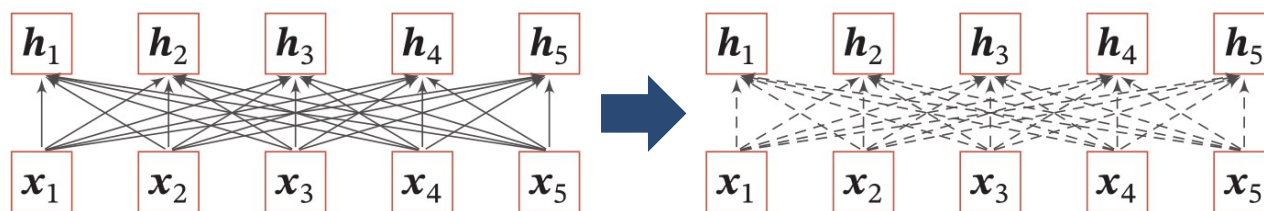
(b) 双向循环网络

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?



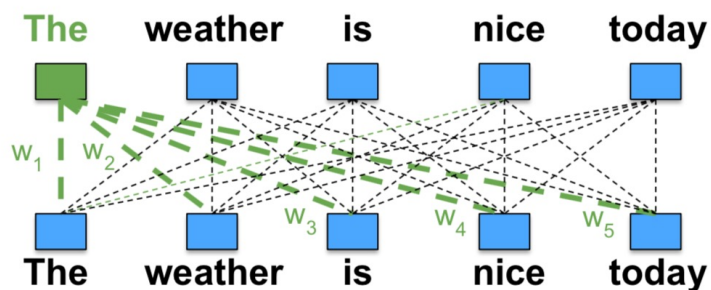
(a) 全连接模型

(b) 自注意力模型

无法处理变长问题

连接权重 $\alpha_{ij}$  由注意力机制动态生成

## 自注意力示例



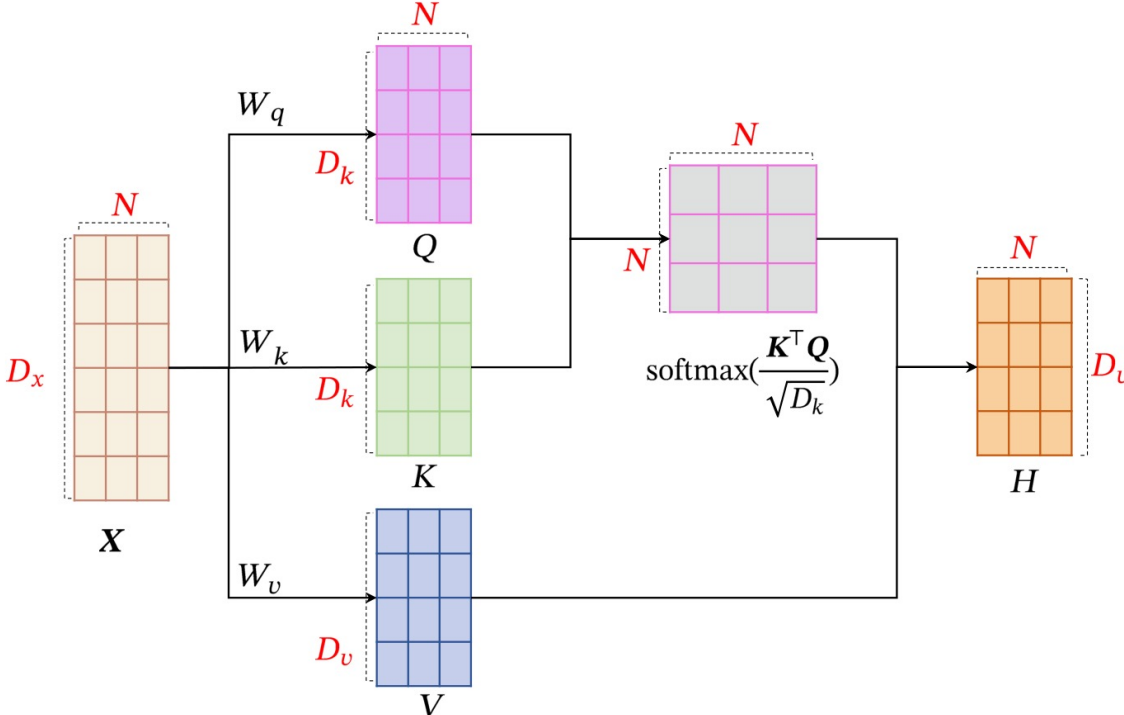
$$W_1, W_2, W_3, W_4, W_5 = \text{softmax} \left( \begin{bmatrix} 0.6 & 0.2 & 0.8 \end{bmatrix} \times \begin{bmatrix} 0.6 & 0.2 & 0.9 & 0.4 & 0.4 \\ 0.2 & 0.3 & 0.1 & 0.1 & 0.1 \\ 0.8 & 0.1 & 0.8 & 0.4 & 0.6 \end{bmatrix} \right)$$

$$\begin{bmatrix} 1.8 \\ 2.3 \\ 0.4 \end{bmatrix} = W_1 \times \begin{bmatrix} 0.6 \\ 0.2 \\ 0.8 \end{bmatrix} + W_2 \times \begin{bmatrix} 0.2 \\ 0.3 \\ 0.1 \end{bmatrix} + W_3 \times \begin{bmatrix} 0.9 \\ 0.1 \\ 0.8 \end{bmatrix} + W_4 \times \begin{bmatrix} 0.4 \\ 0.1 \\ 0.4 \end{bmatrix} + W_5 \times \begin{bmatrix} 0.4 \\ 0.1 \\ 0.6 \end{bmatrix}$$

The
The
weather
is
nice
today

图片来源: [http://fuyw.top/NLP\\_02\\_QANet/](http://fuyw.top/NLP_02_QANet/)

# QKV模式 ( Query-Key-Value )



## 自注意力模型

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

$$Q = W_q X \in \mathbb{R}^{D_k \times N},$$

$$K = W_k X \in \mathbb{R}^{D_k \times N},$$

$$V = W_v X \in \mathbb{R}^{D_v \times N},$$

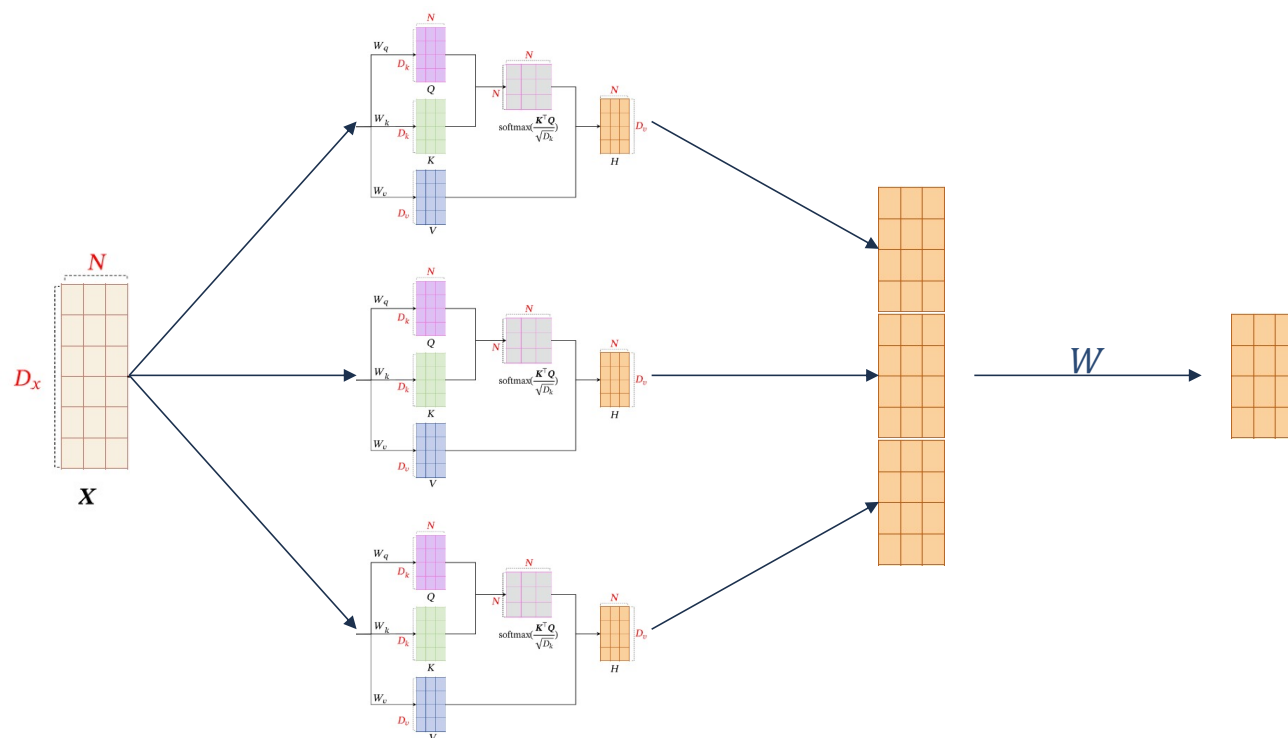
- ▶ 计算  $\mathbf{h}_n$

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

- ▶ 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

$$H = V \text{softmax}\left(\frac{K^T Q}{\sqrt{D_k}}\right),$$

## 多头 (multi-head) 自注意力模型



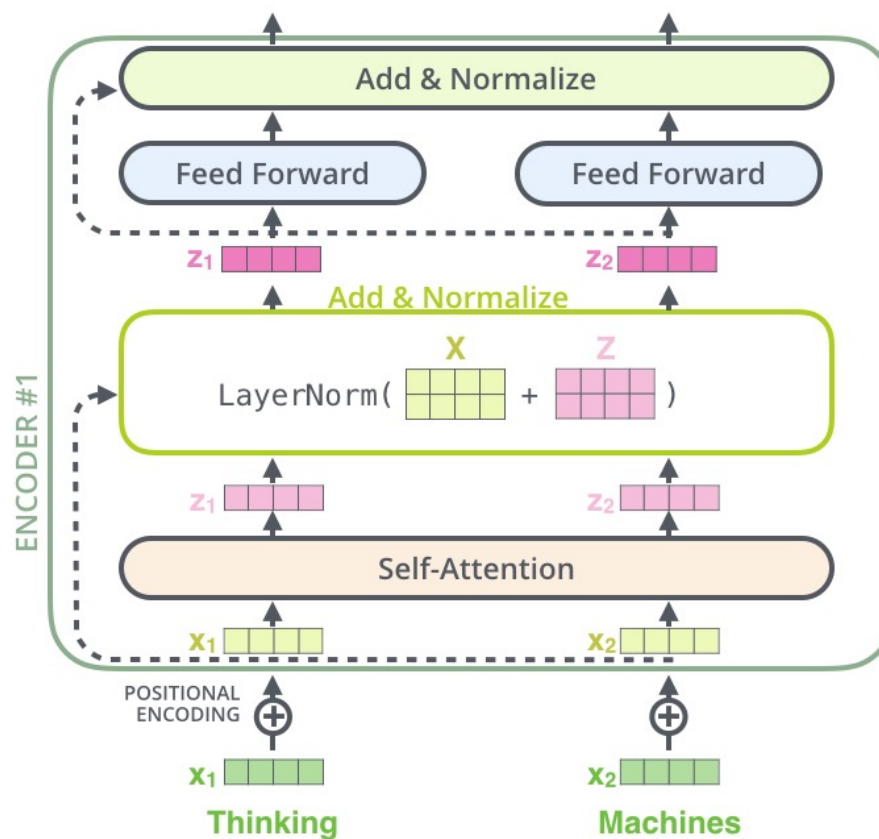
## Transformer Encoder

▶ 仅仅自注意力还不够

▶ 其它操作

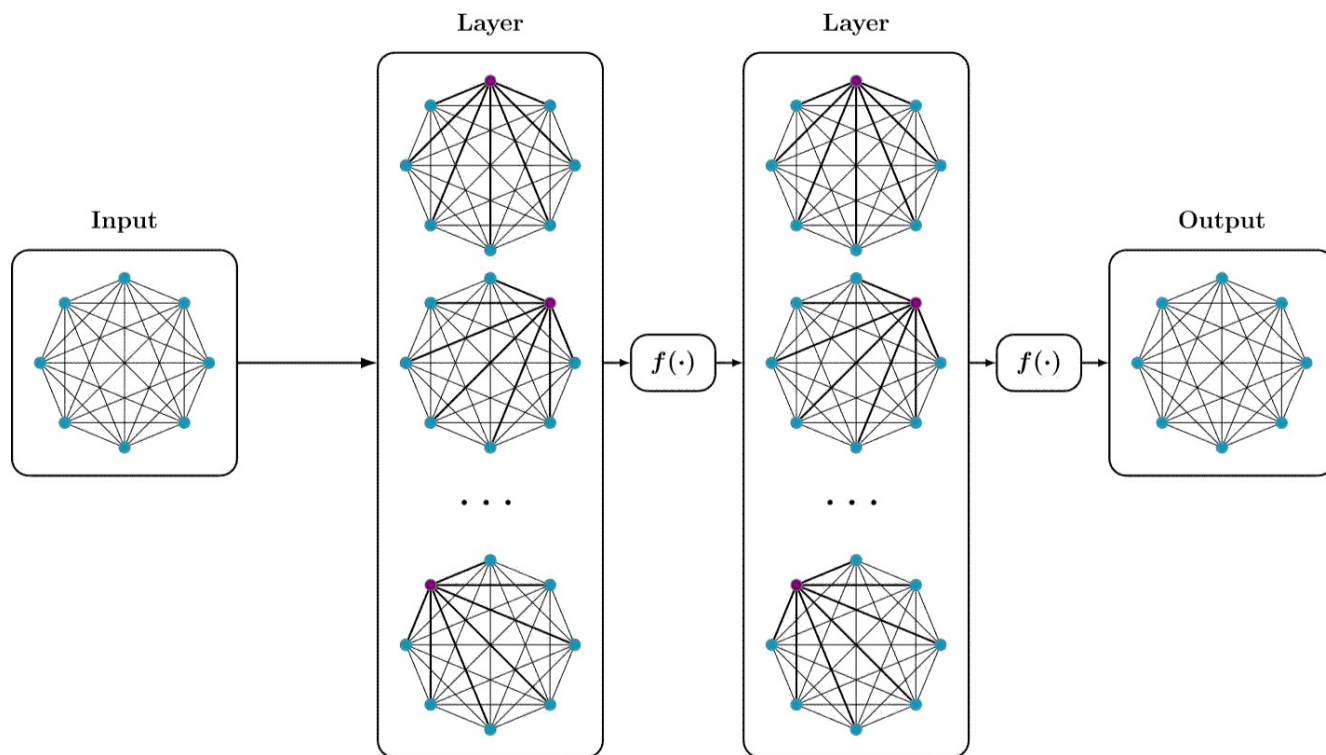
- ▶ 位置编码
- ▶ 层归一化
- ▶ 直连边
- ▶ 逐位的FNN

图片来源: <http://jalammr.github.io/illustrated-transformer/>



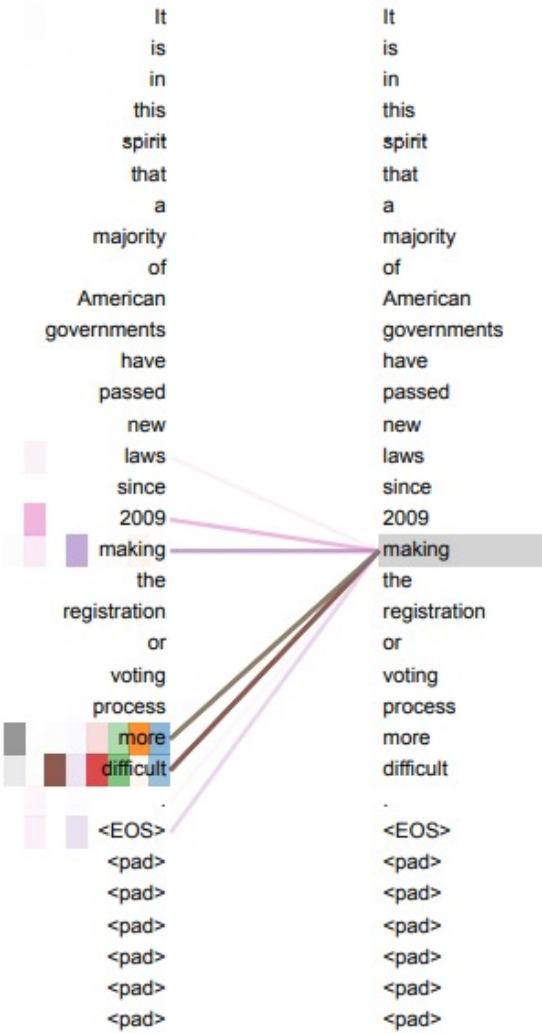
# Transformer

---





# 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

---

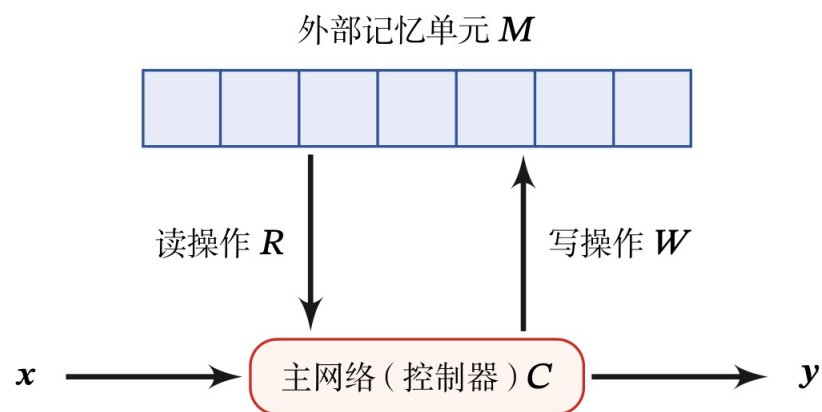
记忆周期	计算机	人脑	神经网络
短期	寄存器	短期记忆	状态(神经元活性)
中期	内存	工作记忆	外部记忆
长期	外存	长期记忆	可学习参数
存储方式	随机寻址	内容寻址	内容寻址为主

## 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

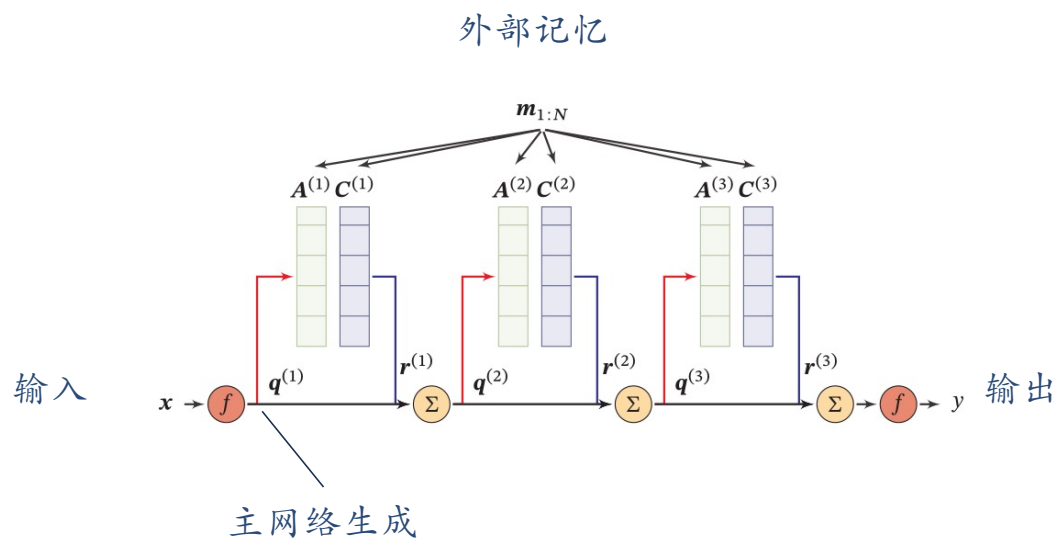
---

- ▶ 外部记忆定义为矩阵  $M \in \mathbb{R}^{d \times k}$ 
  - ▶  $k$  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>



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

$$\mathbf{r}^{(k)} = \sum_{n=1}^N \text{softmax}((\mathbf{a}_n^{(k)})^\top \mathbf{q}^{(k)}) \mathbf{c}_n^{(k)}.$$

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

## 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

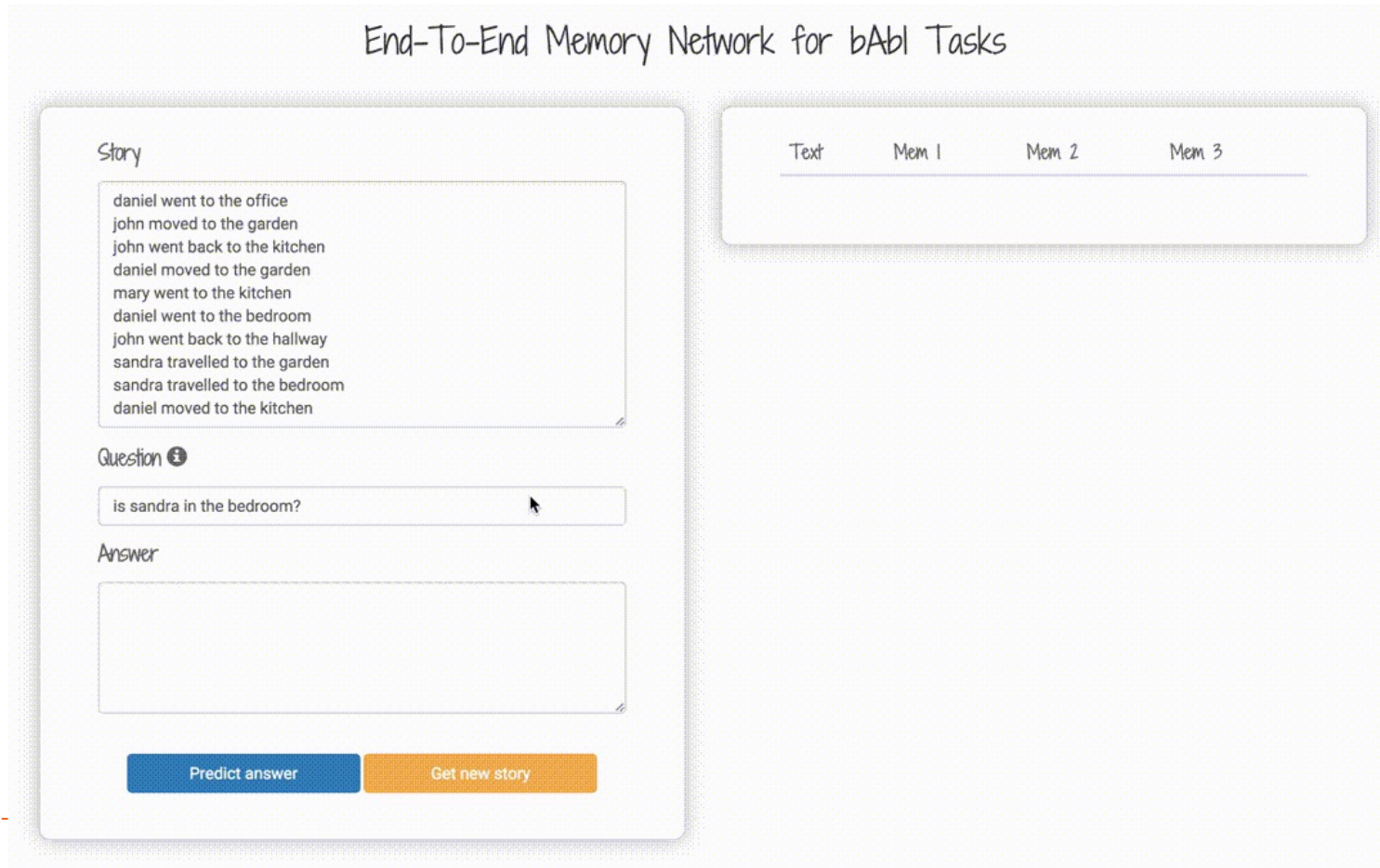
Question ⓘ

is sandra in the bedroom?

Answer

Predict answer    Get new story

Text    Mem 1    Mem 2    Mem 3



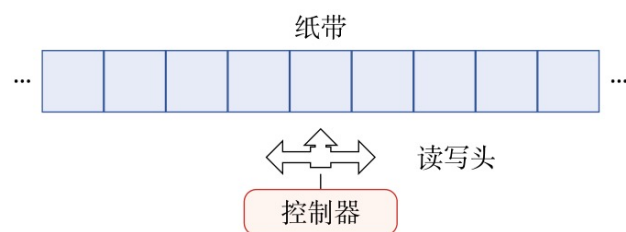


## 神经图灵机

---

### ▶ 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

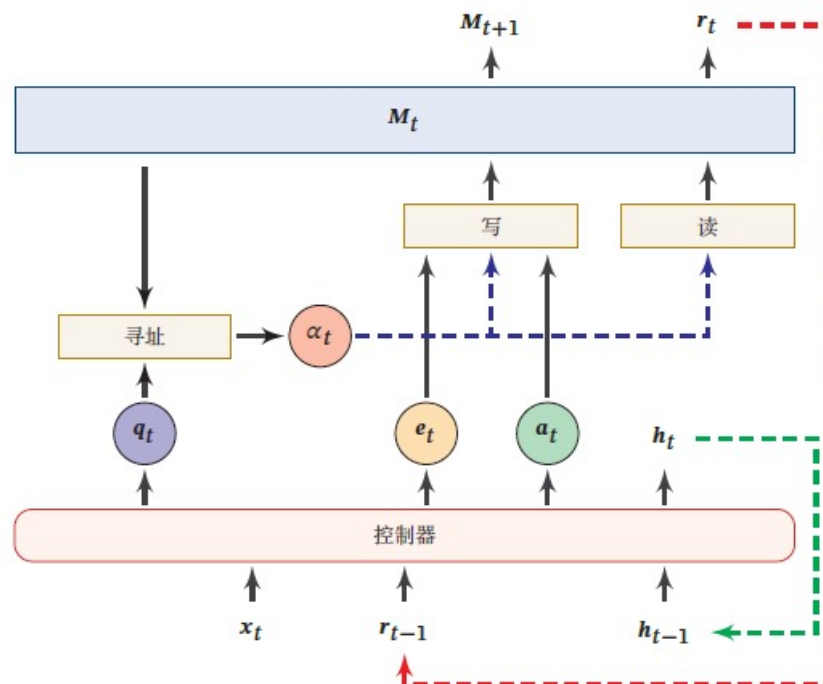


图 8.9 神经图灵机示例

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

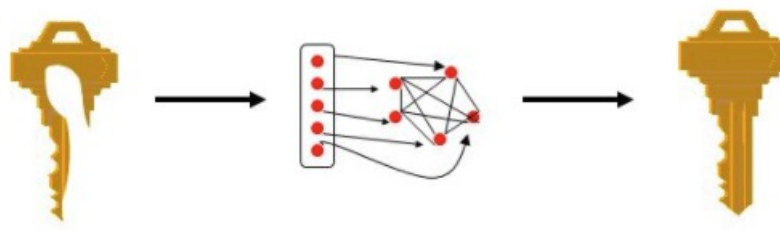
# 基于神经动力学的联想记忆

## 联想记忆

---

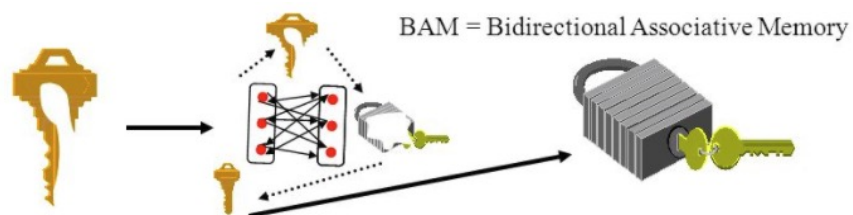
### ▶ 自联想

- ▶ 常称为自编码器



### ▶ 异联想

- ▶ 广义上，大部分机器学习问题都可以被看作异联想



## Hopfield网络

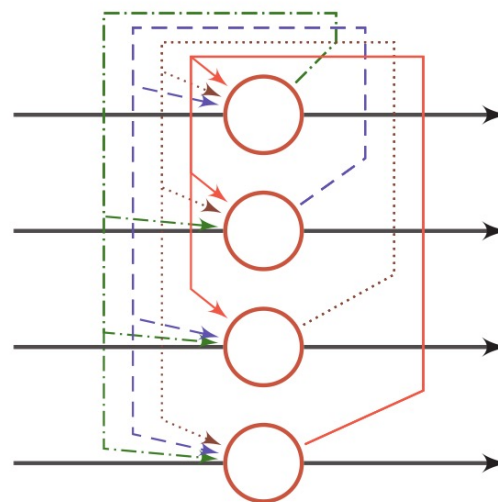
- ▶ 神经网络作为一种记忆的存储和检索模型。
- ▶ 更新规则

$$w_{ii} = 0 \quad \forall i \in [1, m]$$
$$w_{ij} = w_{ji} \quad \forall i, j \in [1, m].$$

$$s_i = \begin{cases} +1 & \text{if } \sum_{j=1}^m w_{ij}s_j + b_i \geq 0, \\ -1 & \text{otherwise,} \end{cases}$$

↓ 同步更新

$$\mathbf{s}_t = f(W\mathbf{s}_{t-1} + \mathbf{b})$$



## 能量函数

---

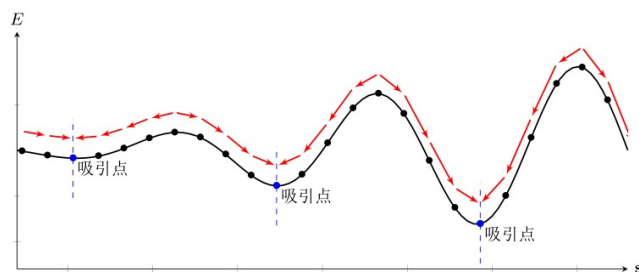
- ▶ 在Hopfield网络中，我们给每个不同的网络状态定义一个标量属性，称为“能量”。

$$\begin{aligned} E &= -\frac{1}{2} \sum_{i,j} w_{ij} s_i s_j - \sum_i b_i s_i \\ &= -\frac{1}{2} \mathbf{s}^T \mathbf{W} \mathbf{s} - \mathbf{b}^T \mathbf{s}. \end{aligned}$$

- ▶ Hopfield网络是稳定的，即能量函数经过多次迭代后会达到收敛状态。

## 检索过程（联想记忆）

- ▶ 给定一个外部输入，网络经过演化，会达到某个稳定状态。



### ▶ 检索过程

- ▶ 每个吸引点 $u$ 都对应一个“管辖”区域。如果输入向量 $x$ 落入这个区域，网络最终会收敛到 $u$ 。
- ▶ 吸引点可以看作是网络中存储的信息。

## 存储过程

---

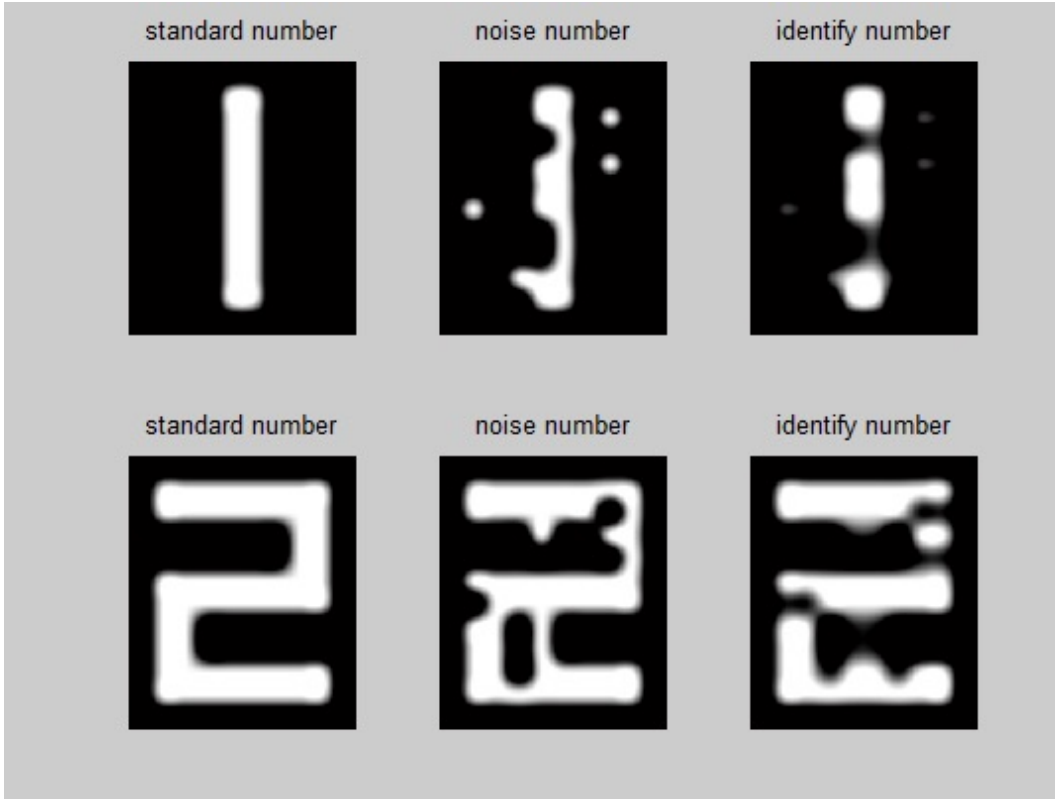
- ▶ 信息存储是指将一组向量  $X = [\mathbf{x}_1, \dots, \mathbf{x}_N]$  存储在网络中的过程。
- ▶ 存储过程主要是调整神经元之间的连接权重，因此可以看做是一种学习过程。
- ▶ 神经元  $i$  和  $j$  之间的连接权重

$$w_{ij} = \frac{1}{N} \sum_{n=1}^N x_i^{(n)} x_j^{(n)},$$

- ▶ 如果两个神经元经常同时激活，则它们之间的连接加强；如果经常不同时激活，则连接消失。



# Hopfield网络用于图像去噪



## 教学内容

---

- ▶ 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)
  - ▶ 基于神经动力学的联想记忆(Hopfield Network)