

# Cache-based Recurrent Attention Network

## 概览

对于时间步  $t$ ，输入词（向量）为  $X^{(t)}$ ，网络的更新步骤大致分为

### 查询

$$\{\alpha_i, Z_i\}_k = Query(X^{(t)}, \mathcal{M}) \quad (1)$$

其中  $Z_i$  为需要进行 Attention 的  $k$  个区域， $\alpha_i$  为其对应的权重  
 $\mathcal{M}$  是 Memory，存储  $N$  个 Key-Value 对，Value 即为区域，对应的 Key 为该区域的一个意义向量。

### Attention

$$Attention_i = Attn(X^{(t)}, Z_i) \quad (2)$$

或

$$Attention_i = Attn(h^{(t-1)}, X^{(t)}, Z_i) \quad (3)$$

$$y^{(t)} = \sum_{i=1}^k \alpha_i Attention_i \quad (4)$$

### 更新 hidden state

$$h^{(t)} = update(X^{(t)}, y^{(t)}) \quad (5)$$

### 更新 Memory

$$\mathcal{M} = renew(h^{(t)}, \mathcal{M}) \quad (6)$$

## 1 查询

- 1.1 standard

$$\{\alpha_i, Z_i\}_k = \text{topk}(\text{softmax}(X^{(t)} \cdot \text{Keys}^T))$$

- 1.2 可能的变化:  $\{\alpha_i, Z_i\}_k = \text{Query}(h^{(t-1)}, X^{(t)}, \mathcal{M})$

$$\{\alpha_i, Z_i\}_k = \text{topk}(\text{softmax}((W_l \cdot \text{concat}(X^{(t)}, h^{(t-1)}) + b_l) \cdot \text{Keys}^T))$$

## 2 Attention

- 2.1  $\text{Attention}_i = \text{Attn}(X^{(t)}, Z_i)$

$$Q = \text{ReLU}(W_Q X^{(t)} + b_Q)$$

$$K = \text{ReLU}(W_K X^{(t)} + b_K)$$

$$\text{Attn}(X^{(t)}, Z_i) = \text{Attn}(Q, K, Z_i) = \sum_j \text{softmax}(Q \cdot K^T)_j Z_{ij}$$

$$y^{(t)} = \sum_{i=1}^k \alpha_i \text{Attention}_i$$

- 2.2  $\text{Attention}_i = \text{Attn}(h^{(t-1)}, X^{(t)}, Z_i)$

$$Q = \text{ReLU}(W_Q \cdot \text{concat}(X^{(t)}, h^{(t-1)}) + b_Q)$$

$$K = \text{ReLU}(W_K X^{(t)} + b_K)$$

$$\text{Attn}(X^{(t)}, Z_i) = \text{Attn}(Q, K, Z_i) = \sum_j \text{softmax}(Q \cdot K^T)_j Z_{ij}$$

$$y^{(t)} = \sum_{i=1}^k \alpha_i \text{Attention}_i$$

## 3 更新 hidden state

- 3.1 standard

$$h^{(t)} = \text{ReLU}(W_H \cdot \text{concat}(X^{(t)}, y^{(t)}) + b_H)$$

- 3.2 gated

$$r^{(t)} = \sigma(W_r \cdot \text{concat}(X^{(t)}, y^{(t)}) + b_r)$$

$$z^{(t)} = \sigma(W_z \cdot \text{concat}(X^{(t)}, y^{(t)}) + b_z)$$

$$n^{(t)} = \tanh(r^{(t)} \cdot (W_n X^{(t)} + b_n) + W_i \cdot y^{(t)} + b_i)$$

$$h^{(t)} = (1 - z^{(t)}) \cdot n^{(t)} + z^{(t)} \cdot y^{(t)}$$

## 4 更新 Memory

Memory 的大小为  $N$  个 Key-Value 对, 每个 Value 为  $L$  个  $h$  的序列。分成两种情况来将上一步生成的  $h^{(t)}$  存入 Memory:

- 1 Memory 不满, 直接将  $h^{(t)}$  填入 Memory 中的第一个空位
- 2 Memory 满, 将第一个 Key-Value 对删除, 其余的向前递补, 最后一个 Key-Value 对为空, 成为第 1 种情况

对于填入  $h^{(t)}$  的 Key-Value 对, 需要更新其 Key:

- 4.1  $Key = ReLU(W_s \cdot Value + b_s)$
- 4.2  $Key = MLP(Value)$
- ( $N \times$ )  $x = ReLU(Wx + b)$
- 4.3  $Key = BiLSTM(Value)$
- 4.3  $Key = Transformer(Value)$