

# Cache-based Recurrent Transformer Network

## 概览

对于时间步  $t$ ，输入片段（向量）为  $X_t \in R^{L \times d}$ ，Memory 为  $\mathcal{M} = \{K_i, V_i\}_{i=1}^N$ ，网络的更新步骤大致分为

## 查询

$$\{\alpha_i, Z_i\}_k = \text{Query}(X_t, \mathcal{M}) \quad (1)$$

其中  $Z_i$  为需要进行回忆（以 Transformer Memory 的方式拼接进当前区域）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。

## 更新 hidden state

$$h_t^{1:m} = \text{Transformer}(\alpha_{1:k}, Z_{1:k}, X_t) \quad (2)$$

## 更新 Memory

$$\mathcal{M} = \text{renew}(h_t^{1:m}, \mathcal{M}) \quad (3)$$

## 1 查询

- 1.1 standard

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

其中  $\text{summary}$  函数与之后更新 Memory 时使用的相同

- 1.2 compute first:  $\{\alpha_i, Z_i\}_k = \text{Query}(\text{summary}(\text{Transformer}(X_t)), \mathcal{M})$

$$\{\alpha_i, Z_i\}_k = \text{topk}(\text{softmax}(\text{summary}(\text{Transformer}(X_t)) \cdot \text{Keys}^T))$$

其中  $\text{summary}$  函数与之后更新 Memory 时使用的相同， $\text{Transformer}$  共享模型参数

## 2 更新 hidden state

- 3.1 standard: 采用 Transformer-XL 的方法

$$h_t^{1:m} = \text{Transfromer-XL}(\alpha_{1:k}, Z_{1:k}, X_t)$$

对于  $n=1, \dots, m$

$$\mathbf{m}_t^{n-1} = \text{concat}(Z_{1:k}^{n-1})$$

$$\tilde{\mathbf{m}}_t^{n-1} = \text{concat}(\{\alpha_i Z_i^{n-1}\}_{i=1}^k)$$

$$\tilde{\mathbf{h}}_t^{n-1} = [\text{SG}(\mathbf{m}_t^{n-1}) \circ \mathbf{h}_t^{n-1}]$$

$$\hat{\mathbf{h}}_t^{n-1} = [\text{SG}(\tilde{\mathbf{m}}_t^{n-1}) \circ \mathbf{h}_t^{n-1}]$$

$$\mathbf{q}_t^n, \mathbf{k}_t^n, \mathbf{v}_t^n = \mathbf{h}_t^{n-1} \mathbf{W}_q^{n\top}, \tilde{\mathbf{h}}_t^{n-1} \mathbf{W}_{k,E}^{n\top}, \hat{\mathbf{h}}_t^{n-1} \mathbf{W}_v^{n\top}$$

$$\mathbf{A}_{t,i,j}^n = \mathbf{q}_{t,i}^{n\top} \mathbf{k}_{t,j}^n + \mathbf{q}_{t,i}^{n\top} \mathbf{W}_{k,R}^n \mathbf{R}_{i-j} + u^\top \mathbf{k}_{t,j} + v^\top \mathbf{W}_{k,R}^n \mathbf{R}_{i-j}$$

$$\mathbf{a}_t^n = \text{Masked-Softmax}(\mathbf{A}_t^n) \mathbf{v}_t^n$$

$$\mathbf{o}_t^n = \text{LayerNorm}(\text{Linear}(\mathbf{a}_t^n) + \mathbf{h}_t^{n-1})$$

$$\mathbf{h}_t^n = \text{Positionwise-Feed-Forward}(\mathbf{o}_t^n)$$

## 3 更新 Memory

Memory 的大小为  $N$  个 Key-Value 对，每个 Value 为一个片段  $X_i$  的所有层的表示。分成两种情况来将上一步生成的  $h_t^{1:m}$  存入 Memory:

- 1 Memory 不满，直接将  $h_t^{1:m}$  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，用最顶层表示更新 Key:

$$\text{Key} = \text{summary}(h_t^n)$$

- 4.1  $\text{Key} = \text{ReLU}(W_S \cdot h_t^n + b_S)$

- 4.2  $\text{Key} = \text{MLP}(h_t^n)$

$$(N \times) \quad x = \text{ReLU}(Wx + b)$$

- 4.3  $\text{Key} = \text{BiLSTM}(h_t^n)$