Fast Video Object Segmentation using the Global Context Module

Yu Li[1⋆], Zhuoran Shen[2⋆], and Ying Shan[1]

[1] Applied Research Center (ARC), Tencent PCG
[2] The University of Hong Kong

1 Proof of Equivalence between the Global Context Module and the Space-Time Memory Module

1.1 Mathematical Formulations of the Global Context Module and the Space-Time Memory Module

The global context (GC) module consists of three steps, namely context extraction, context update, and context distribution. The context extraction step can be expressed as

$$\boldsymbol{C}_t = k(\boldsymbol{X}_t)^{\mathsf{T}} v(\boldsymbol{X}_t), \tag{1}$$

where t is the index of the current frame, $\boldsymbol{X}_t$ is the input feature, $\boldsymbol{C}_t$ is the global context feature of the current frame, and k, v are the key and value generation functions. The context update step is

$$\boldsymbol{G}_t = \frac{t-1}{t}\boldsymbol{G}_{t-1} + \frac{1}{t}\boldsymbol{C}_t, \tag{2}$$

where $\boldsymbol{G}_t$ is the global context feature of the first t frames. The context distribution step is

$$\boldsymbol{R}_t = q(\boldsymbol{X}_t)\boldsymbol{G}_{t-1}, \tag{3}$$

where $\boldsymbol{R}_t$ is the distributed global context feature and q is the query generation function.

The space-time memory (STM) [1] module similarly consists of three steps, namely memory production, memory write, and memory read. The memory production step can be expressed as

$$\begin{aligned} \boldsymbol{K}_{P,t} &= k(\boldsymbol{X}_t), \\ \boldsymbol{V}_{P,t} &= v(\boldsymbol{X}_t), \end{aligned} \tag{4}$$

where t is the index of the current frame, $\boldsymbol{X}_t$ is the input feature, $\boldsymbol{K}_{M,t}, \boldsymbol{V}_{M,t}$ are the key and value produced from the current frame, and k, v are the key and value generation functions. The memory write operation is

$$\begin{aligned} \boldsymbol{K}_{M,t} &= \boldsymbol{K}_{M,t-1} \odot \boldsymbol{K}_{P,t}, \\ \boldsymbol{V}_{M,t} &= \boldsymbol{V}_{M,t-1} \odot \boldsymbol{V}_{P,t}, \end{aligned} \tag{5}$$

⋆ Both authors contributed equally. This work was done when Zhuoran was interning at Tencent.

where $\boldsymbol{K}_{M,t}, \boldsymbol{V}_{M,t}$ are the key and value in the memory from the first t frames and $\odot$ denotes concatenation along the spatial dimension. The memory read step is

$$\boldsymbol{E}_t = \frac{1}{t}\left(q(\boldsymbol{X}_t)\boldsymbol{K}_{M,t-1}^{\mathsf{T}}\right)\boldsymbol{V}_{M,t-1}, \tag{6}$$

where $\boldsymbol{E}_t$ is the memory reading and q is the query generation function.

1.2 Proof of Equivalence

Expanding Equation (2) by substituting it to itself gives

$$\boldsymbol{G}_{t-1} = \frac{1}{t-1}\sum_{f=1}^{t-1}\boldsymbol{C}_f. \tag{7}$$

Note that $\boldsymbol{G}_0$ is the zero matrix. Substituting Equation (7) into Equation (3) results in

$$\boldsymbol{R}_t = \frac{1}{t-1}q(\boldsymbol{X}_t)\sum_{f=1}^{t-1}\boldsymbol{C}_f. \tag{8}$$

Combining Equations (1) and (8), we have

$$\boldsymbol{R}_t = \frac{1}{t-1}q(\boldsymbol{X}_t)\sum_{f=1}^{t-1}k(\boldsymbol{X}_f)^{\mathsf{T}}v(\boldsymbol{X}_f). \tag{9}$$

Similarly, expanding Equations (5) gives

$$\begin{aligned}\boldsymbol{K}_{M,t-1} &= \bigodot_{f=1}^{t-1}\boldsymbol{K}_{P,f},\\ \boldsymbol{V}_{M,t-1} &= \bigodot_{f=1}^{t-1}\boldsymbol{V}_{P,f},\end{aligned} \tag{10}$$

where $\bigodot$ stands for concatenation along the spatial dimension. Substituting Equations (4) into Equation (10) results in

$$\begin{aligned}\boldsymbol{K}_{M,t-1} &= \bigodot_{f=1}^{t-1}k(\boldsymbol{X}_f),\\ \boldsymbol{V}_{M,t-1} &= \bigodot_{f=1}^{t-1}v(\boldsymbol{X}_f).\end{aligned} \tag{11}$$

Combining Equations (6) and (11), we have

$$\begin{aligned}
\boldsymbol{E}_t &= \frac{1}{t-1}\left(q(\boldsymbol{X}_t)\bigodot_{f=1}^{t-1} k(\boldsymbol{X}_f)^{\mathsf{T}}\right)\bigodot_{f=1}^{t-1} v(\boldsymbol{X}_f) \\
&= \frac{1}{t-1}\left(\bigodot_{f=1}^{t-1} q(\boldsymbol{X}_t)k(\boldsymbol{X}_f)^{\mathsf{T}}\right)\bigodot_{f=1}^{t-1} v(\boldsymbol{X}_f) \\
&= \frac{1}{t-1}\sum_{f=1}^{t-1} q(\boldsymbol{X}_t)k(\boldsymbol{X}_f)^{\mathsf{T}}v(\boldsymbol{X}_f) \\
&= \frac{1}{t-1}q(\boldsymbol{X}_t)\sum_{f=1}^{t-1} k(\boldsymbol{X}_f)^{\mathsf{T}}v(\boldsymbol{X}_f)
\end{aligned} \tag{12}$$

Comparing Equations (9) and (12), we have

$$\boldsymbol{R}_t = \boldsymbol{E}_t, \tag{13}$$

which completes the proof.

1.3 Impact of Softmax Normalization

In practice, in addition to the operations in (6), the STM module usually uses a softmax along the rows on $q(\boldsymbol{X}_t)\boldsymbol{K}_{M,t-1}^{\mathsf{T}}$. This operation makes each row sum up to 1. The interpretation of this normalization is that it makes each row, which corresponds to a query pixel, represent a probabilistic distribution of attention over all locations in the memory.

Since softmax is non-linear, the GC module cannot perfectly recreate its effect. However, the GC module applies two softmax operations, one on each row of $q(\boldsymbol{X}_t)$ and the other on each column of $k(\boldsymbol{X}_t)$. Then, if one hypothetically multiplies these two normalized matrices, the product matrix will share the critical property of $\text{softmax}(q(\boldsymbol{X}_t)\boldsymbol{K}_{M,t-1}^{\mathsf{T}})$, i.e. each row sums up to 1 and represents a probabilistic distribution of attention for the query pixel over all locations in the past frames. Therefore, the two operations together are an effective approximation of the single softmax in STM. Thus, the GC module remains approximately equivalent to the STM module even with the presence of the softmax operations.

References

1. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (2019)