Design for Ring Buffer Operator in MXNet

Hyunsu Cho
chyunsu@amazon.com

The *ring buffer* operator shall remember a set number of last seen inputs. The previously seen inputs are stored in an internal buffer, which is updated in-place each time a new input is admitted. This operator only supports the forward pass and does not produce backward gradients.

1 Inputs and Outputs

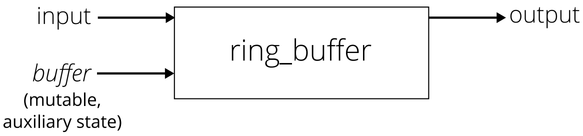

Figure 1: Schematic for inputs and outputs

- `input`: a tensor of dimension $(N_0, N_1, \ldots, N_{\text{axis}-1}, N_{\text{axis}}, N_{\text{axis}+1}, \ldots, N_{D-1})$, where axis is given as a parameter.
- `buffer`: a tensor of dimension $(N_0, N_1, \ldots, N_{\text{axis}-1}, B, N_{\text{axis}+1}, \ldots, N_{D-1})$, where the *buffer length* B is given as a parameter. (Assume $N_{\text{axis}} \leq B$.) This input is a mutable, auxiliary state: the operator will update the input in-place.
- `output`: the latest value of `buffer`.

2 Pseudocode

For clarity, we define $\mathbb{N}[a, b) = \{a, a+1, \ldots, b-2, b-1\}$ as the subset of positive integers that are at least a and strictly less than b.

The first step is to shift the content of the buffer to the left by N_{axis}:

```
**for** $(i_0, i_1, \ldots, i_{D-1}) \in \mathbb{N}[0, N_0) \times \cdots \times \mathbb{N}[0, N_{\text{axis}-1}) \times \mathbb{N}[N_{\text{axis}}, B) \times \mathbb{N}[0, N_{\text{axis}+1}) \times \cdots \times \mathbb{N}[0, N_{D-1})$ **do**
  buffer$(i_0, \ldots, i_{\text{axis}-1}, i_{\text{axis}} - N_{\text{axis}}, \ldots, i_{\text{axis}+1}, \ldots, i_{D-1})$
    $\leftarrow$ buffer$(i_0, \ldots, i_{\text{axis}-1}, i_{\text{axis}}, i_{\text{axis}+1}, \ldots, i_{D-1})$
**end for**
```

The second step is to save the latest input to the buffer:

```
**for** $(i_0, i_1, \ldots, i_{D-1}) \in \mathbb{N}[0, N_0) \times \cdots \times \mathbb{N}[0, N_{\text{axis}-1}) \times \mathbb{N}[0, N_{\text{axis}}) \times \mathbb{N}[0, N_{\text{axis}+1}) \times \cdots \times \mathbb{N}[0, N_{D-1})$ **do**
  buffer$(i_0, \ldots, i_{\text{axis}-1}, i_{\text{axis}} + B - N_{\text{axis}}, \ldots, i_{\text{axis}+1}, \ldots, i_{D-1})$
    $\leftarrow$ input$(i_0, \ldots, i_{\text{axis}-1}, i_{\text{axis}}, i_{\text{axis}+1}, \ldots, i_{D-1})$
**end for**
```

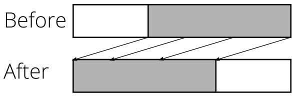

Figure 2: Visualizing the shift operation in the buffer in the 1D case