Operator Derivation for Gated OJA Rule

Jiaxi Hu, **Moonshot AI**

https://github.com/fla-org/flash-linear-attention

1 QKVO Rotary Position Embedding

The mechanism of multiplicative positional encodings like RoPE can be analyzed through a generalized attention formulation:

$$\boldsymbol{o}_t = \sum_{j=1}^{t} \boldsymbol{v}_j \exp\left(\left(\mathbf{R}_j \boldsymbol{k}_j\right)^\top \left(\mathbf{R}_t \boldsymbol{q}_t\right)\right) = \sum_{j=1}^{t} \boldsymbol{v}_j \exp\left(\boldsymbol{k}_j^\top \left(\prod_{s=i+1}^{t} \mathbf{R}_s\right) \boldsymbol{q}_t\right) \tag{1}$$

where the position relationship between the t-th query $\boldsymbol{q}_t$ and the i-th key $\boldsymbol{k}_i$ is reflected by the cumulative matrix products. RoPE defines the transformation matrix $\mathbf{R}_j$ as a block diagonal matrix composed of $d_k/2$ 2D rotation matrices $\mathbf{R}_j^k = \begin{pmatrix} \cos(j\theta_k) & -\sin(j\theta_k) \\ \sin(j\theta_k) & \cos(j\theta_k) \end{pmatrix}$ with **per-2-dimensional** angular frequency θ_k. Due to the properties of rotation matrices, i.e., $\mathbf{R}_{t-i} = \mathbf{R}_t^\top \mathbf{R}_i$, absolute positional information $\mathbf{R}_t$ and $\mathbf{R}_i$ can be applied separately to $\boldsymbol{q}_t$ and $\boldsymbol{k}_i$, which are then transformed into relative positional information $t-i$ encoded as $\prod_{j=i+1}^{t} \mathbf{R}_j = \begin{pmatrix} \cos((t-i)\theta_k) & -\sin((t-i)\theta_k) \\ \sin((t-i)\theta_k) & \cos((t-i)\theta_k) \end{pmatrix}$.

What would happen if we applied a rotation matrix to $\boldsymbol{V}$?

$$\boldsymbol{o}_t = \mathbf{R}_t^\top \sum_{j=1}^{t} \left(\mathbf{R}_j \boldsymbol{v}_j\right) \exp\left(\boldsymbol{k}_j^\top \boldsymbol{q}_t\right) = \sum_{j=1}^{t} \left(\prod_{s=i+1}^{t} \mathbf{R}_s\right) \boldsymbol{v}_j \exp\left(\boldsymbol{k}_j^\top \boldsymbol{q}_t\right) \tag{2}$$

1.1 Gated Delta Rule as QK Position Encodings

$$\boldsymbol{o}_t = \sum_{j=1}^{t} \boldsymbol{v}_j \left(\boldsymbol{k}_j^\top \left(\prod_{s=j+1}^{t} \operatorname{diag}\left(\boldsymbol{\alpha}_s\right) \left(\boldsymbol{I} - \beta_s \boldsymbol{k}_s \boldsymbol{k}_s^\top\right)\right) \boldsymbol{q}_t\right) \tag{3}$$

1.2 Gated Oja Rule as VO Position Encodings

$$\boldsymbol{o}_t = \sum_{j=1}^{t} \left(\prod_{s=j+1}^{t} \left(\boldsymbol{I} - \beta_s \boldsymbol{v}_s \boldsymbol{v}_s^\top\right) \operatorname{diag}\left(\boldsymbol{\alpha}_s\right)\right) \boldsymbol{v}_j \left(\boldsymbol{k}_j^\top \boldsymbol{q}_t\right) \tag{4}$$

Preprint. Under review.

1.3 Gated Oja Rule with Value Gate

For a standard Gated OjaNet with a diagonal key gate,

$$\boldsymbol{S}_t = \mathrm{Diag}\left(\boldsymbol{\alpha}_t\right)\boldsymbol{S}_{t-1} + \beta_t \boldsymbol{v}_t \left(\boldsymbol{k}_t^\intercal - \boldsymbol{v}_t^\intercal \mathrm{Diag}\left(\boldsymbol{\alpha}_t\right)\boldsymbol{S}^{t-1}\right) = \left(\boldsymbol{I} - \beta_t \boldsymbol{v}_t \boldsymbol{v}_t^\intercal\right)\mathrm{Diag}\left(\boldsymbol{\alpha}_t\right)\boldsymbol{S}_{t-1} + \beta_t \boldsymbol{v}_t \boldsymbol{k}_t^\intercal$$

By partially expanding the recurrence, we have

$$\boldsymbol{S}_{[t]}^r = \underbrace{\left(\prod_{i=1}^{r}\left(\boldsymbol{I} - \beta_{[t]}^i \boldsymbol{v}_{[t]}^i \boldsymbol{v}_{[t]}^{i\intercal}\right)\mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^i\right)\right)}_{:=\boldsymbol{D}_{[t]}^r\ \text{(“pseudo” memory decay)}} \boldsymbol{S}_{[t]}^0 + \underbrace{\sum_{i=1}^{r}\left(\left(\prod_{j=i+1}^{r}\left(\boldsymbol{I} - \beta_{[t]}^j \boldsymbol{v}_{[t]}^j \boldsymbol{v}_{[t]}^{j\intercal}\right)\mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^j\right)\right)\beta_{[t]}^i \boldsymbol{v}_{[t]}^i \boldsymbol{k}_{[t]}^{i\intercal}\right)}_{:=\boldsymbol{H}_{[t]}^r\ \text{(“pseudo” Incremental memory)}}$$

Then, we employ the WY representation:

$$\boldsymbol{D}_{[t]}^r = \mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{1:r}\right) - \sum_{i=1}^{r}\mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{i:r}\right)\boldsymbol{v}_{[t]}^i \boldsymbol{w}_{[t]}^{i\intercal} \qquad \boldsymbol{w}_{[t]}^r = \beta_{[t]}^r\left(\mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{1:r}\right)\boldsymbol{v}_{[t]}^r - \sum_{i=1}^{r-1}\boldsymbol{w}_{[t]}^i\left(\boldsymbol{v}_{[t]}^{i\intercal}\mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{i:r}\right)\boldsymbol{v}_{[t]}^r\right)\right)$$

$$\boldsymbol{H}_{[t]}^r = \sum_{i=1}^{r}\mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{i:r}\right)\boldsymbol{v}_{[t]}^i \boldsymbol{u}_{[t]}^{i\intercal} \qquad \boldsymbol{u}_{[t]}^r = \beta_{[t]}^r\left(\boldsymbol{k}_{[t]}^r - \sum_{i=1}^{r-1}\boldsymbol{u}_{[t]}^i\left(\boldsymbol{v}_{[t]}^{i\intercal}\mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{i:r}\right)\boldsymbol{v}_{[t]}^r\right)\right)$$

To maximize hardware efficiency, we apply the UT transform to reduce non-matmul FLOPs, which is crucial to enable better hardware utilization during training.

$$\boldsymbol{W}_{[t]} = \boldsymbol{M}_{[t]}\mathrm{Diag}\left(\beta_{[t]}^{1\to C}\right)\left(\boldsymbol{A}_{[t]}^{1\to C} \odot \boldsymbol{V}_{[t]}\right), \qquad \boldsymbol{U}_{[t]} = \boldsymbol{M}_{[t]}\mathrm{Diag}\left(\beta_{[t]}^{1\to C}\right)\boldsymbol{K}_{[t]}$$

$$\boldsymbol{M}_{[t]} = \left(\boldsymbol{I} + \mathrm{lower}\left(\mathrm{Diag}\left(\beta_{[t]}^{1\to C}\right)\left(\boldsymbol{V}_{[t]} \odot \boldsymbol{A}_{[t]}^{1\to C}\right)\left(\frac{\boldsymbol{V}_{[t]}^\intercal}{\boldsymbol{A}_{[t]}^{1\to C}}\right)\right)\right)^{-1}$$

Then we have the following vector form:

$$\boldsymbol{S}_{[t]}^r = \boldsymbol{D}_{[t]}^r \boldsymbol{S}_{[t]}^0 + \boldsymbol{H}_{[t]}^r = \mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{1:r}\right)\boldsymbol{S}_{[t]}^0 + \sum_{i=1}^{r}\mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{i:r}\right)\boldsymbol{v}_{[t]}^i\left(\boldsymbol{u}_{[t]}^{i\intercal} - \left(\boldsymbol{w}_{[t]}^{i\intercal}\boldsymbol{S}_{[t]}^0\right)\right)$$

$$\boldsymbol{o}_{[t]}^r = \boldsymbol{S}_{[t]}^r \boldsymbol{q}_{[t]}^r = \mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{1:r}\right)\boldsymbol{S}_{[t]}^0 \boldsymbol{q}_{[t]}^r + \sum_{i=1}^{r}\mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{i:r}\right)\boldsymbol{v}_{[t]}^i\left(\boldsymbol{u}_{[t]}^{i\intercal} - \left(\boldsymbol{w}_{[t]}^{i\intercal}\boldsymbol{S}_{[t]}^0\right)\right)\boldsymbol{q}_{[t]}^r$$

Equivalently, in matrix form:

$$\boldsymbol{S}_{[t+1]} = \mathrm{Diag}\left(\boldsymbol{\alpha}_{[t]}^{1:C}\right)\boldsymbol{S}_{[t]} + \left(\boldsymbol{A}_{[t]}^{i\to C} \odot \boldsymbol{V}_{[t]}\right)^\intercal\left(\boldsymbol{U}_{[t]} - \boldsymbol{W}_{[t]}\boldsymbol{S}_{[t]}\right)$$

$$\textbf{VO-RoPE:}\quad \boldsymbol{O}_{[t]} = \boldsymbol{A}_{[t]}^{1\to C} \odot \left(\boldsymbol{Q}_{[t]}\boldsymbol{S}_{[t]}^\intercal + \mathrm{Tril}\left(\boldsymbol{Q}_{[t]}\underbrace{\left(\boldsymbol{U}_{[t]} - \boldsymbol{W}_{[t]}\boldsymbol{S}_{[t]}\right)^\intercal}_{\text{“pseudo”-key term}}\right)\frac{\boldsymbol{V}_{[t]}}{\left(\boldsymbol{A}_{[t]}^{1\to C}\right)}\right)$$

References