Flash Attention

Safe Softmax

Making Softmax Safe

$S = QK^T \in R^{N \times N}$, $P = \text{Softmax}(S) \in R^{N \times N}$, $O = PV \in R^{N \times d}$

$Q \rightarrow$ Seq token (N, d)

$K \rightarrow (N, d) \rightarrow (d, N)$

$V \rightarrow (N, d)$

$QK^T = (N, N)$ $\quad$ (N, N) $\quad$ (N, d)

$q_1^T k_1$	$q_1^T k_2$	$q_1^T k_3$	$q_1^T k_4$	$q_1^T k_5$
$q_5^T k_1$	$q_5^T k_2$	$q_5^T k_3$	$q_5^T k_4$	$q_5^T k_5$

Softmax →

$(5, 5)$

0.1	0.05	0.5	0.15	0.2	$\Rightarrow \Sigma = 1$
0.3	0.1	0.35	0.2	0.05	$\rightarrow \Sigma = 1$

$(5, 5)$

$x = [3, 2, 5, 1]$

1) $x_{max} = 5$

2) $e^{3-5} + e^{2-5} + e^{5-5} + e^{1-5}$

$e^{-2} + e^{-3} + e^{0} + e^{-4} = \lambda$

3) $x_1 = \frac{e^{3-5}}{\lambda}$ $x_2 = \frac{e^{2-5}}{\lambda}$ $x_3 = \frac{e^{5-5}}{\lambda}$ $x_4 = \frac{e^{1-5}}{\lambda}$

Sequential soll "Slow"

$m_0 = -\infty$

$m_1 = max(-\infty, 3) = 3$

$m_2 = max(3, 2) = 3$

$m_3 = max(3, 5) = 5$

$m_4 = max(5, 1) = 5$

$x_1' = \frac{e^{3-5}}{\lambda_4}$ $x_3' = \frac{e^{5-5}}{\lambda_4}$

$x_2' = \frac{e^{2-5}}{\lambda_4}$ $x_4' = \frac{e^{1-5}}{\lambda_4}$

$\lambda_0 = 0$

$\lambda_1 = \lambda_0 + e^{3-5} =$

$\lambda_2 = \lambda_1 + e^{2-5} =$

$\lambda_3 = \lambda_2 + e^{5-5} =$

$\lambda_4 = \lambda_3 + e^{1-5} = e^{3-5} + e^{2-5} + e^{5-5} + e^{1-5} =$

$=$

$x = [3, 2, 5, 1]$

Fusion:

Step 1:

$max_1 = 3$

$\lambda_1 = e^{3-3} = e^0$

Step 2:

$max_2 = max(3, 2)$

$\lambda_2 = \lambda_1 + e^{2-3}$

Step 3:

$max_3 = max(3, 5) = 5$

$\lambda_3 = \lambda_2 + e^{5-5} = e^{3-3} + e^{2-3} + e^{5-5}$

λ_3 = computed is wrong!

$\lambda_3 = \underbrace{e^{3-3} + e^{2-3}}_{\text{Wrong as global max is 5}} + e^{5-5}$

correction factor

Fix!

$\lambda_3 = \lambda_2 \cdot e^{3-5} + e^{5-5}$

$= (e^{3-3} + e^{2-3}) e^{3-5} + e^{5-5}$

$= e^{+3-3+3-5}$

$$= e^{3-3+3-5} + e^{2-3+3-5} + e^{5-5} = e^{3-5} + e^{2-5} + e^{5-5}$$

current max

Prev max

correct!

$$l_3 = l_2 \, e^{3-5} + e^{5-5} = (e^{3-3} + e^{2-3}) e^{3-5} + e^{5-5}$$

New Pseudocode

$m_0 = -\infty$

$l_0 = 0$

for $i = 1$ to N

$$m_i = \max(m_{i-1}, x_i)$$

$$l_i = l_{i-1} \cdot e^{m_{i-1} - m_i} + e^{x_i - m_i}$$

} we want to prove that at the end of this loop

$$m_N = \max_i (x_i) = x_{max}$$

$$l_N = \sum_{j=1}^{N} e^{x_j - x_{max}}$$

for $k = 1$ to N

$$x_N \leftarrow \frac{e^{x_k - x_N}}{l_N}$$

Prof by Induction

1. Size $N = 1$

$$m_1 = \max(-\infty, x_1) = x_1 = \max_i (x_i) = X_{max}$$

$$l_1 = 0 \times e^{-\infty} + e^{x_1 - x_1} = \sum_{j=1}^{N} e^{x_i - x_{max}}$$

correction factor

2. Size $N+1$

$$m_{N+1} = \max(m_N, x_{m+1}) = \max_i (x_i)$$

$$\therefore l_{N+1} = l_N e^{m_N - m_{N+1}} + e^{x_{N+1} - m_{N+1}} =$$

$$= \left(\sum_{j=1}^{N} e^{x_j} \right.$$

2) If we assume it holds for a vector of size N, does it hold for a vector of size N+1 ?

$$m_{N+1} = \max(m_N, x_{N+1}) = \max_i (x_i)$$

$$l_{N+1} = l_N e^{m_N - m_{N+1}} + e^{x_{N+1} - m_{N+1}}$$

prev max — current max

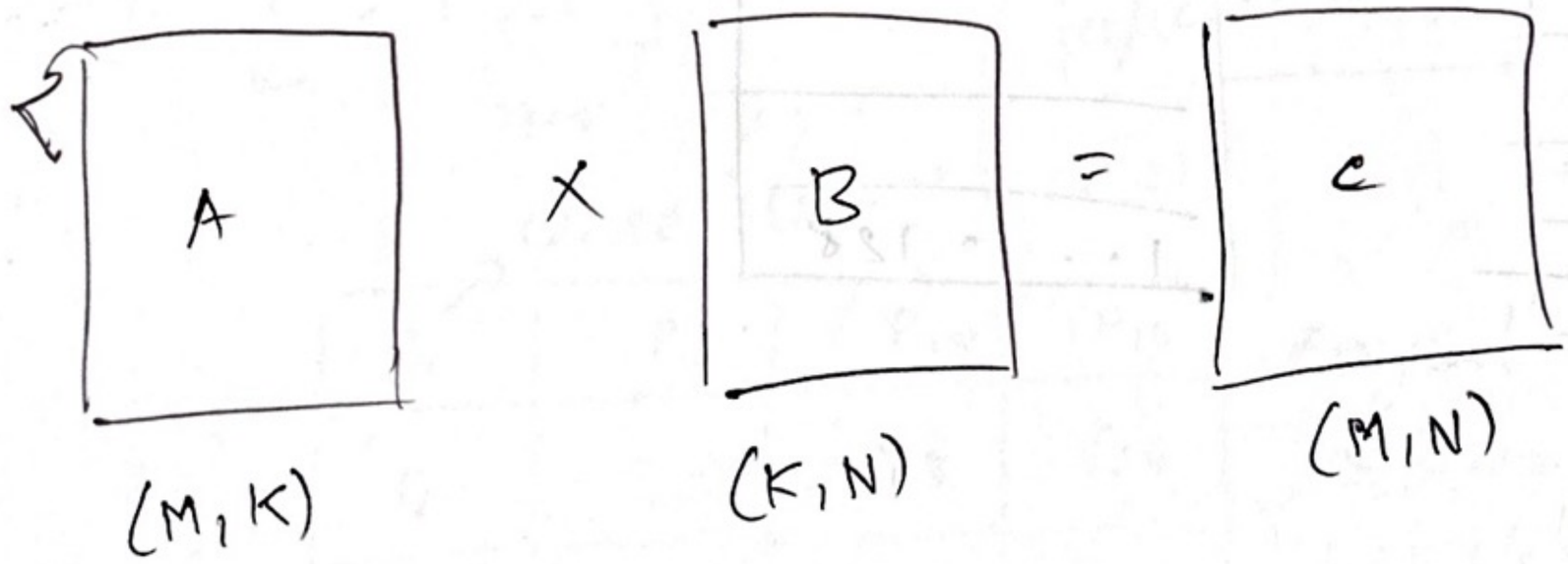
$$= \left(\sum_{J=1}^{N} e^{x_J - m_N}\right) e^{m_N - m_{N+1}} + e^{x_{N+1} - m_{N+1}}$$

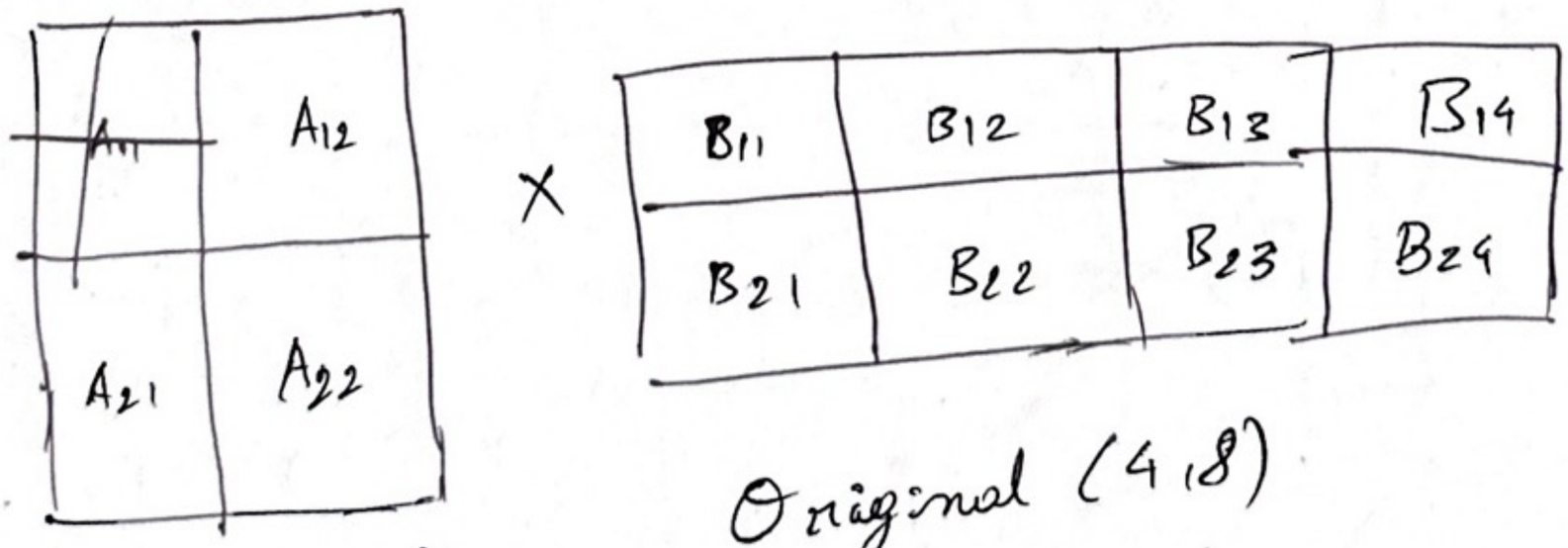
local max — correction factor — current element — current set of max (local max)

$$= \sum_{J=1}^{N} e^{x_J - m_{N+1}} + e^{x_{N+1} - m_{N+1}}$$

$$= \sum_{J=1}^{N+1} e^{x_J - m_{N+1}}$$

BLOCK MATRIX MULTIPLICATION





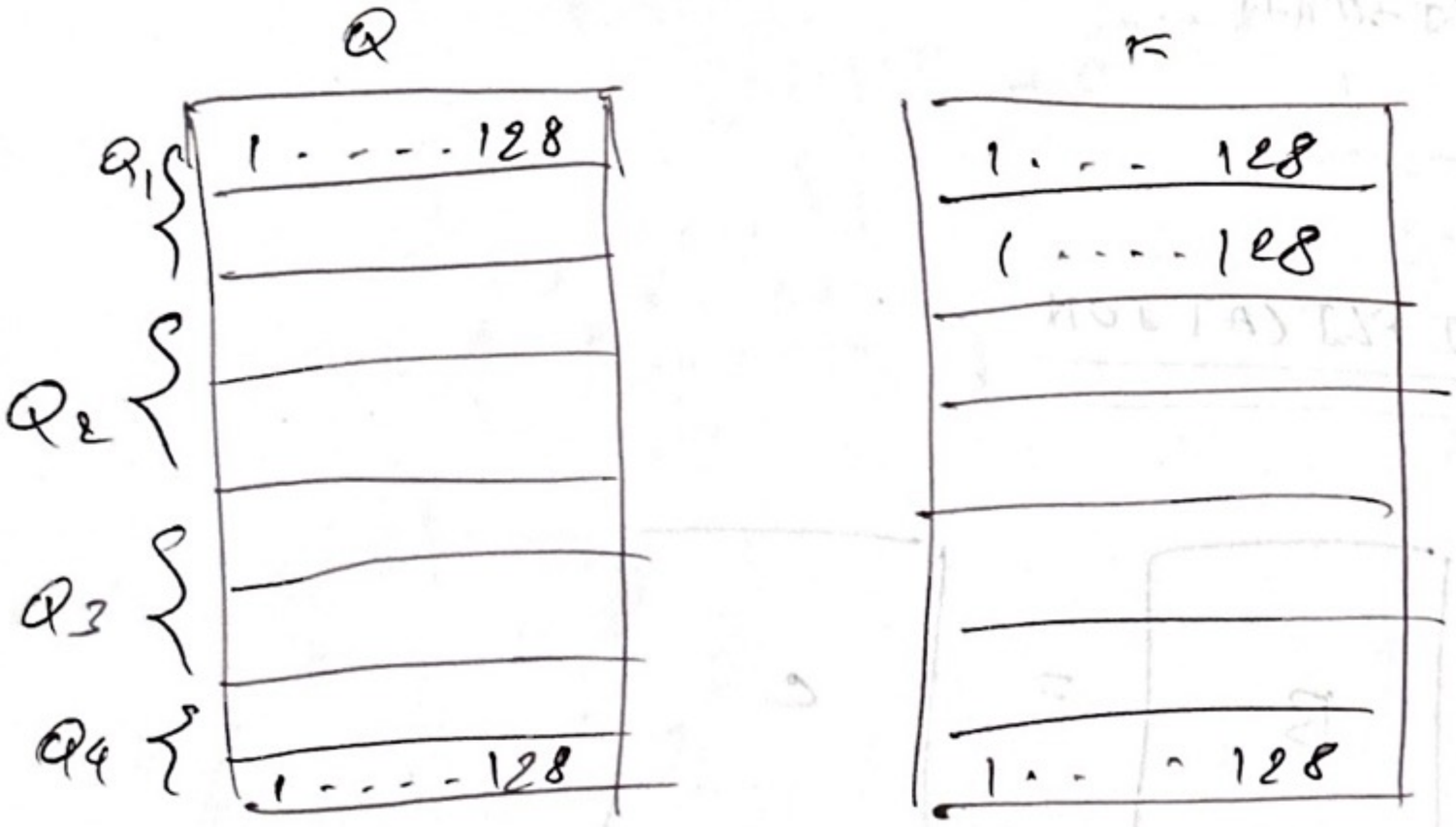
Original (8, 4)
Block (2, 2)

Original (4, 8)
Block (2, 4)

$A_{11} B_{11}$ + $A_{12} B_{21}$	$A_{11} B_{12}$ + $A_{12} B_{21}$	$A_{11} B_{13}$ + $A_{12} B_{23}$	$A_{11} B_{14}$ + $A_{12} B_{24}$
$A_{21} B_{11}$ + $A_{22} B_{21}$	$A_{21} B_{12}$ + $A_{21} B_{22}$	$A_{21} B_{13}$ + $A_{22} B_{23}$	$A_{21} B_{14}$ + $A_{22} B_{24}$

$$O = (QK^T)V \in R^{N \times d}$$

$$Q, K, V \in R^{N \times d}$$



S_{11} $(2,128) \times (128,2) = (2,2)$

$Q_1 K_1^T$	$Q_1 K_2^T$	$Q_1 K_3^T$	$Q_1 K_4^T$
$Q_2 K_1^T$	$Q_2 K_2^T$	$Q_2 K_3^T$	$Q_2 K_4^T$
$Q_3 K_1^T$	$Q_3 K_2^T$	$Q_3 K_3^T$	$Q_3 K_4^T$
$Q_4 K_1^T$	$Q_4 K_2^T$	$Q_4 K_3^T$	$Q_4 K_4^T$

Softmax* ⇒

Original = (8, 8)

Block = (4, 4)

$$\text{Softmax}(x_i) = \frac{\exp(x_i - x_{max})}{\sum_{j=1}^{N} \exp(x_i - x_{max})}$$

$$\text{Softmax}^*(x_i) = \exp(x_i - x_{max})$$

S_{11} = | a | b | / | c | d | → Softmax* → | exp(a-a) | exp(b-a) | / | exp(c-d) | exp(d-d) |

max

$(2,128) \times (128,2) = (2,2)$

P =

P_{11}	P_{12}	P_{13}	P_{14}
P_{21}	P_{22}	P_{23}	P_{24}
P_{31}	P_{32}	P_{33}	P_{34}
P_{41}	P_{42}	P_{43}	P_{44}

Original = (8, 8)

BLOCK = (4, 4)

$$\text{SOFTMAX}^*(S_{ij}) = \exp\left[S_{ij} - \text{row max}(S_{ij})\right]$$

$P_{11} = \text{Softmax}^*(Q_1 K_1^T)$

$P_{12} = \text{Softmax}^*(Q_1 K_2^T)$

[2, 2] ← etc

Let's multiply by V

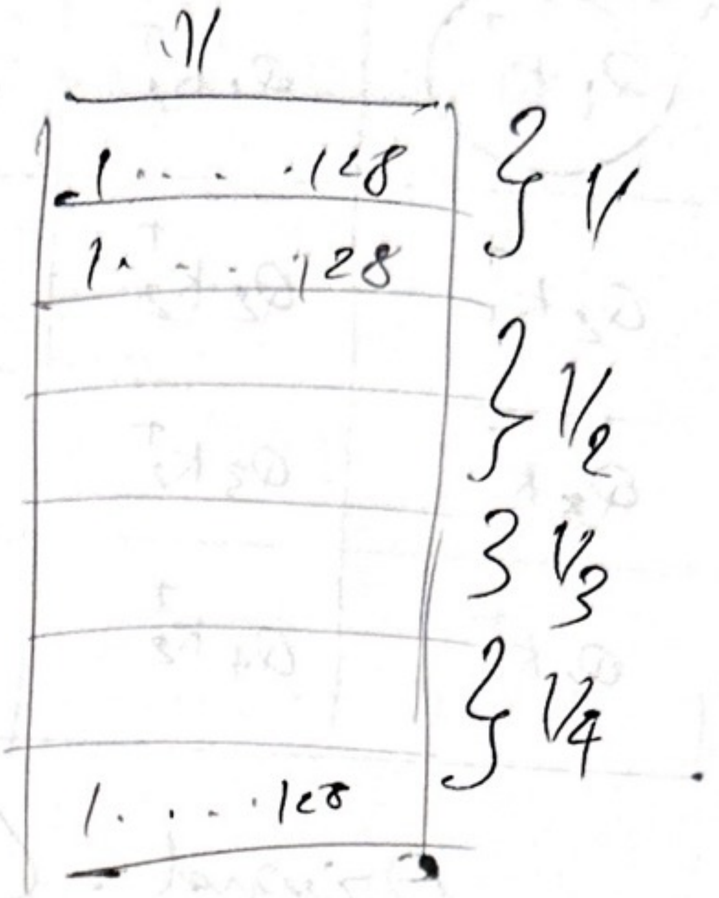
$m_{11} = (2 \times 1)$

$\rightarrow P_{11} = (2 \times 2)$

$P =$

P_{11}	P_{12}	P_{13}	P_{14}
P_{21}	P_{22}	P_{23}	P_{24}
P_{31}	P_{32}	P_{33}	P_{34}
P_{41}	P_{42}	P_{43}	P_{44}

$\times$



Original $= (8, 8)$

BLOCK $= (4, 4)$

$(4,2) \times (2, 128) = (2, 128)$

$O =$

$P_{11} V_1 + P_{12} V_2 + P_{13} V_3 + P_{14} V_4$

Normal Attention:

$Q K^T$ → P_{ij} → softmax

$Q_1^T K_1$	$Q_1^T K_2$	$Q_1^T K_3$	$Q_1^T K_4$	$Q_1^T K_5$
$Q_2^T K_1$	$Q_2^T K_2$	$Q_2^T K_3$	$Q_2^T K_4$	$Q_2^T K_5$

→ local maximum for softmax

PSEDUCODE

FOR EACH BLOCK
  O_i = ZEROS(2,128)
  FOR EACH BLOCK K_j
    P_{ij} = softmax*($Q_i K_j^T$)
    $O_i \leftarrow O_i + P_{ij}$ $\forall j$
  END FOR
END FOR

Fix! The Online Softmax

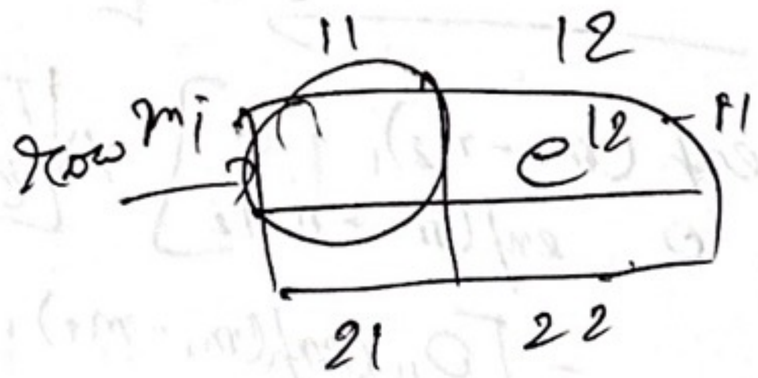
$m_0 = -\infty$
$l_0 = 0$
for $i = 1$ to N
  $m_i = \max(m_{i-1}, x_i)$
  $l_i = l_{i-1} \cdot e^{m_{i-1} - m_i} + e^{x_i - m_i}$
for $k = 1$ to N
  $x_k \leftarrow \frac{e^{x_k - m_N}}{l_N}$

Initialization

$m_0 = \begin{bmatrix} -\infty \\ -\infty \end{bmatrix}$

$l_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$O_0 = \begin{bmatrix} 0 & 0 & 0 \dots & 0 & 0 \\ 0 & 0 & 0 \dots & 0 & 0 \end{bmatrix}$ 2 × 128 matrix



Step 1 :

$$m_1 = \max(\text{row max}(\overset{S_{11}}{Q_1 K_1^T}), m_0)$$

$$S_1 = Q_1 K_1^T$$

$$l_1 = \text{row sum}\left[\exp(S_1 - m_1)\right] + l_0 \cdot \exp(m_0 - m_1)$$

$$P_{11} = \exp(S_1 - m_1)$$

$$O_1 = \exp(\exp(m_0 - m_1))O_0 + P_{11} V_1$$

Step 1 :

$$m_1 = \max(\text{row max}(\underset{\downarrow}{Q_1 K_1^T}), m_0)$$

$$S_{11}$$

$$S_1 = Q_1 K_1^T$$

$$l_1 = \text{row sum}\left[\exp(S_1 - m_1)\right] + l_0 \cdot \exp(m_0 - m_1)$$

$$P_1 = \exp(S_1 - m_1)$$

$$O_1 = \text{diag}(\exp(m_0 - m_1))O_0 + P_{11} V_1$$

Step 2 :

$$m_2 = \max(\text{row max}(\overset{S_{12}}{Q_1 K_2^T}), m_1)$$

$$S_2 = Q_1 K_2^T$$

$$l_2 = \text{row sum}$$

$$P_{12} = \exp(S_2 - m_2)$$

$$O_1 = \text{diag}(\exp(m_1 - m_2))O_1 + P_{12} V_2$$

$$O_1 = \begin{bmatrix} O_{11} & O_{12} & O \cdots & O_{1,128} \\ O_{21} & O_{22} & \cdots & O_{2,128} \end{bmatrix}$$

$$\times$$

$$\begin{bmatrix} \exp(m_1 - m_2)_1 & 0 \\ 0 & \exp(m_1 - m_2)_2 \end{bmatrix}$$

$$\longrightarrow \quad \begin{bmatrix} \exp(m_1 - m_2)_1 & 0 \\ 0 & \exp(m_1 - m_2)_2 \end{bmatrix} \times \begin{bmatrix} O_{11} & O_{12} \cdots & O_{1,128} \\ O_{21} & O_{22} \cdots & O_{2,128} \end{bmatrix}$$

$$= \begin{bmatrix} O_{11}(\exp(m_1 - m_2)_1 & & O \\ & & \\ O_{21}(\exp(m_1 - m_2)_2 & & O \end{bmatrix}$$

Step 5:

$$O_5 = \left[\text{diag}(l_4)\right]^{-1} O_4$$

$$\begin{bmatrix} O_1(\exp(m_1 - m_2)_1 + O(O_{21}) \end{bmatrix}$$

$$l_4 = \begin{bmatrix} l_4^{(1)} \\ l_4^{(2)} \end{bmatrix} \quad O_4 = \begin{bmatrix} [1 \cdots \cdots 128] \\ [1 \cdots \cdots 128] \end{bmatrix}$$

$$\left(\begin{bmatrix} l_4^{(1)} & 0 \\ 0 & l_4^{(2)} \end{bmatrix} \right)^{-1} = \begin{bmatrix} \frac{1}{l_4^{(1)}} & 0 \\ 0 & \frac{1}{l_4^{(2)}} \end{bmatrix} \times O \begin{bmatrix} [1 \cdots \cdots 128] \\ [1 \cdots \cdots 128] \end{bmatrix}$$

$(2,2)$ $\qquad (2,128)$

$$= \begin{bmatrix} \frac{1}{l_4^{(1)}} O_{11} + \frac{1}{l_4^{(2)}} O_{21} \cdots \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{l_4^{(1)}} O_{11} + 0\, O_{21} , \quad \frac{1}{l_4} \\ 0 \times O_{11} + \frac{1}{l_4^{(2)}} O_{21} , \end{bmatrix}$$

$(2,128)$

"Flash Attention -2 Feed Forword"

$Q, K, V \in R^{N \times d}$ by HBM block, sizes B_c, B_r

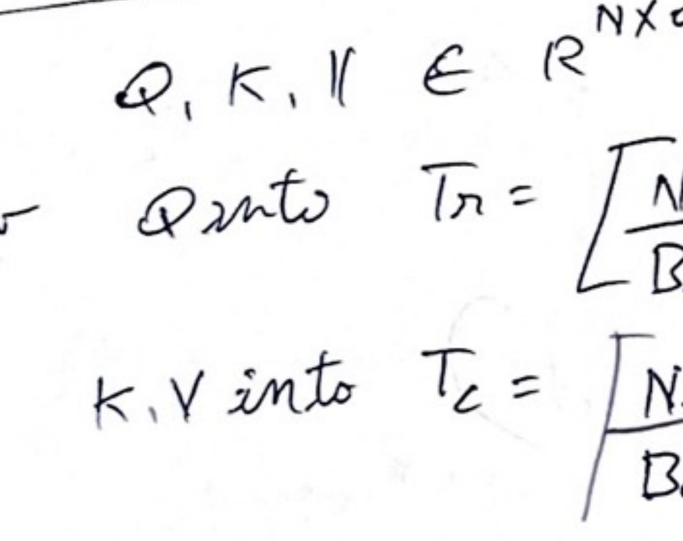
Div Q into $T_r = \left\lceil \frac{N}{B_r} \right\rceil$ blocks $Q_1, \ldots Q_{T_r}$ of size $B_r \times d$

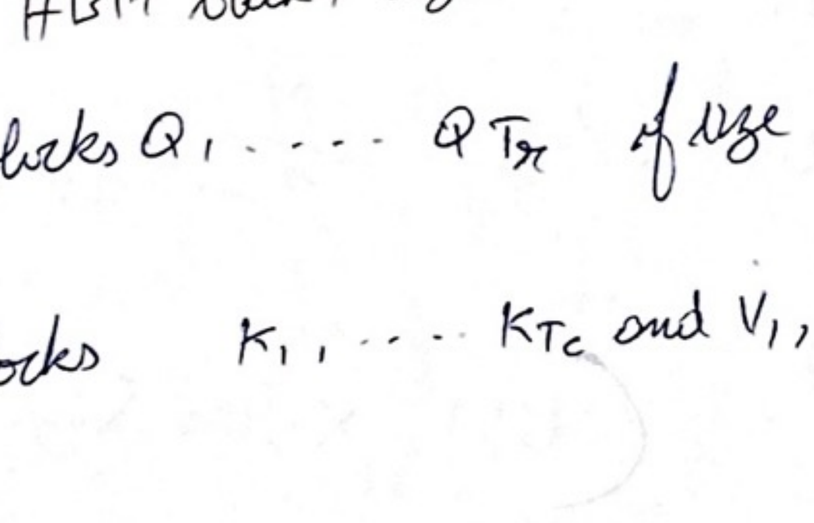
K, V into $T_c = \left\lceil \frac{N_r}{B_c} \right\rceil$ blocks $K_1, \ldots K_{T_c}$ and $V_1, \ldots V_{T_c}, \ldots$

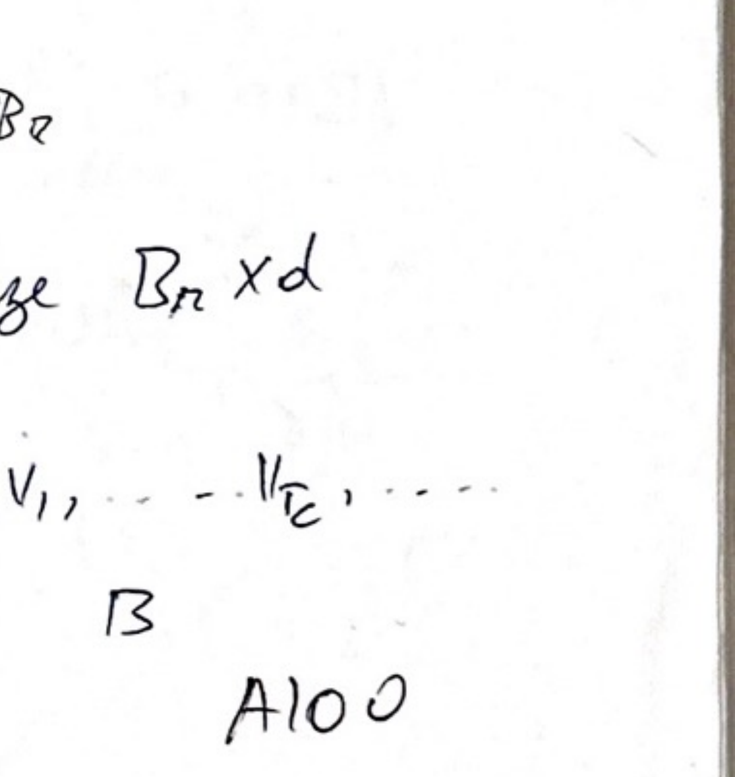
B

A100

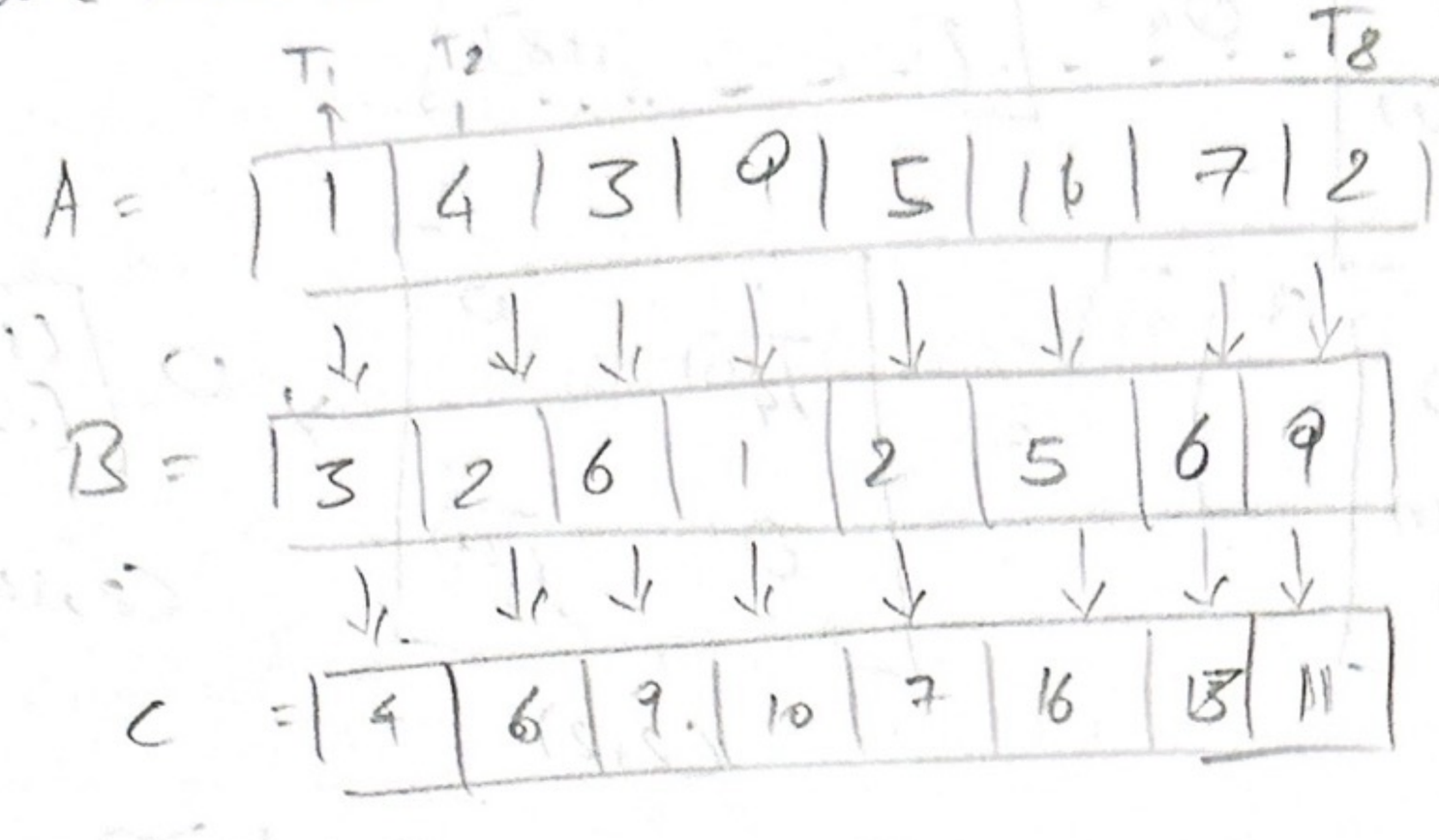
for $1 \le i \le T_r$ do



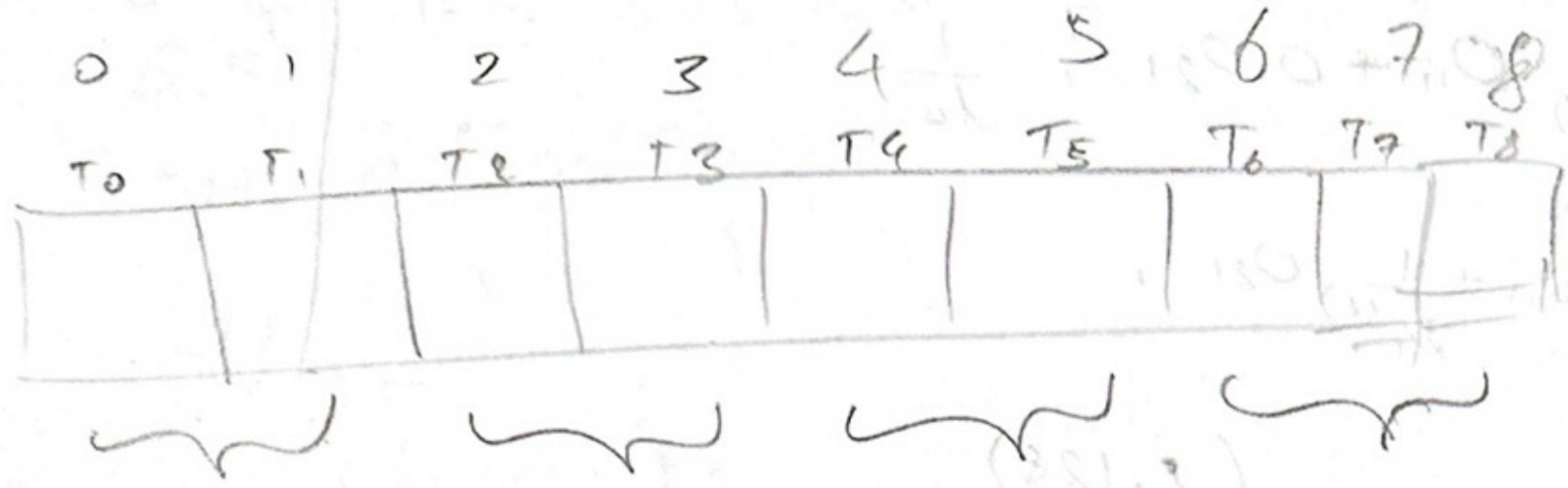




Vector Addition



GPU Block and Thread Allocation

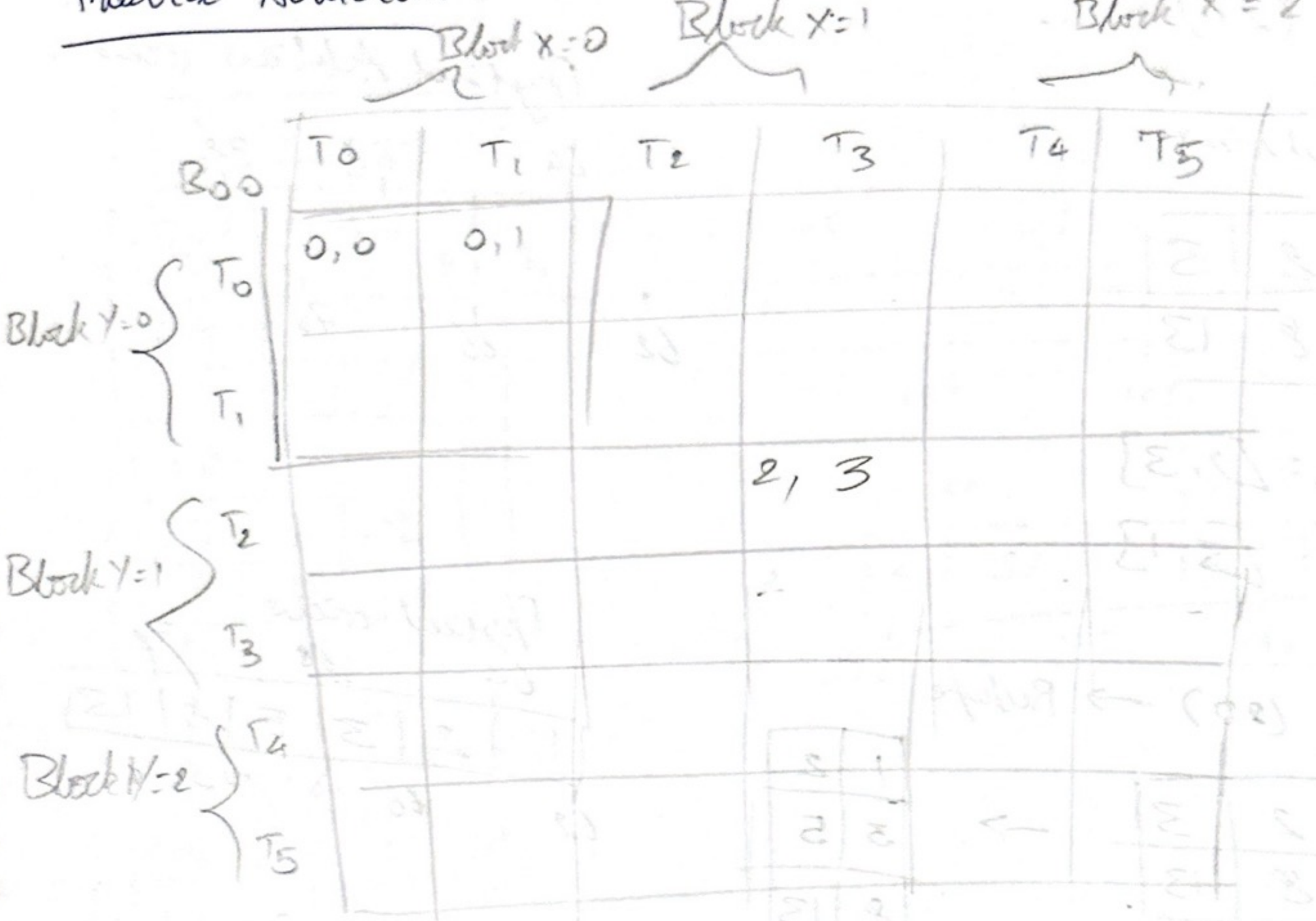


$$Num\_Blocks = \frac{N}{2} \rightarrow BlockSize = 4$$

$$Elem\_Id = i = (B\_ID \times BlockSize + T\_id)$$

$N = 8$ cores $= 4$

Matrix Addition



block_row_ident = bid_y * block_size + thread ID. y
block_col_ident = bid_x * bl_size + thread ID. x

Tensor Layouts

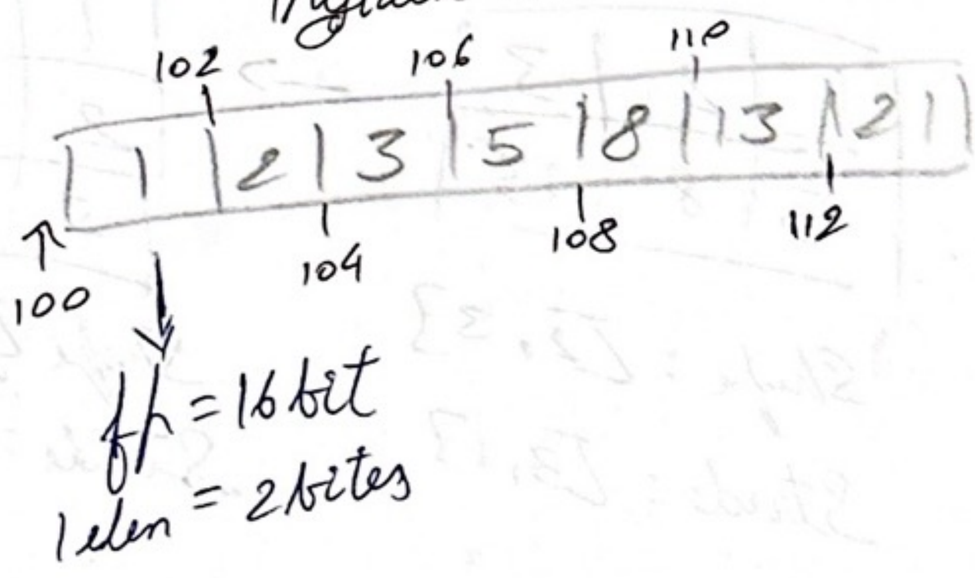
Array / Vector (1D)

Logical view

1	2	3	4	5	6	13	21

Shape : [7]
Stride : [1]

Physical view



memory block -> pointer

Matrix (2D)

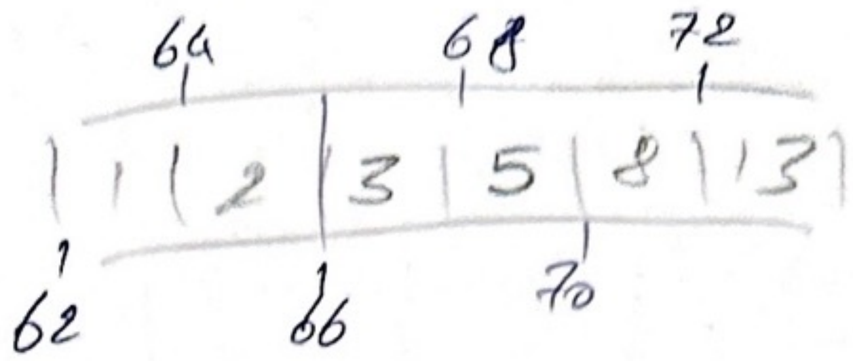
Logical view

1	2	3
5	8	13

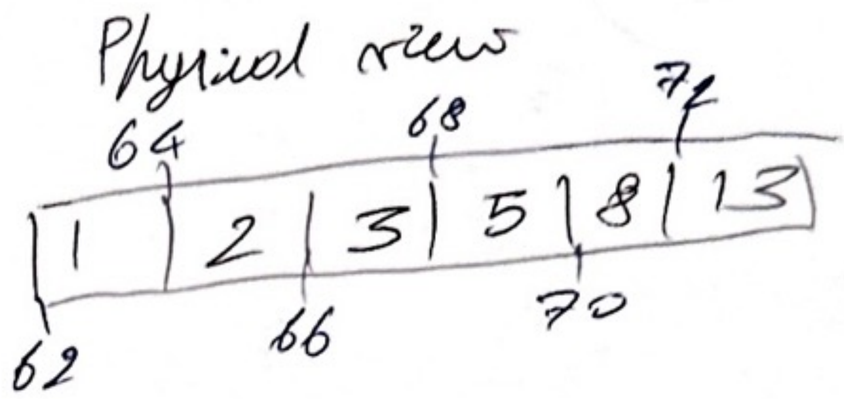
Shape : [2, 3]

Stride : [3, 1]

Physical Address view



Physical view



Matrix (2D) → Reshape

1	2	3
5	8	13

Shape : [2, 3]

Stride : [3, 1]

→

1	2
3	5
8	13

Shape : [3, 2]

Stride : [2, 1]

Matrix (2D) → Transpose

1	2	3
5	8	13

Shape : [2, 3]

Stride : [3, 1]

→

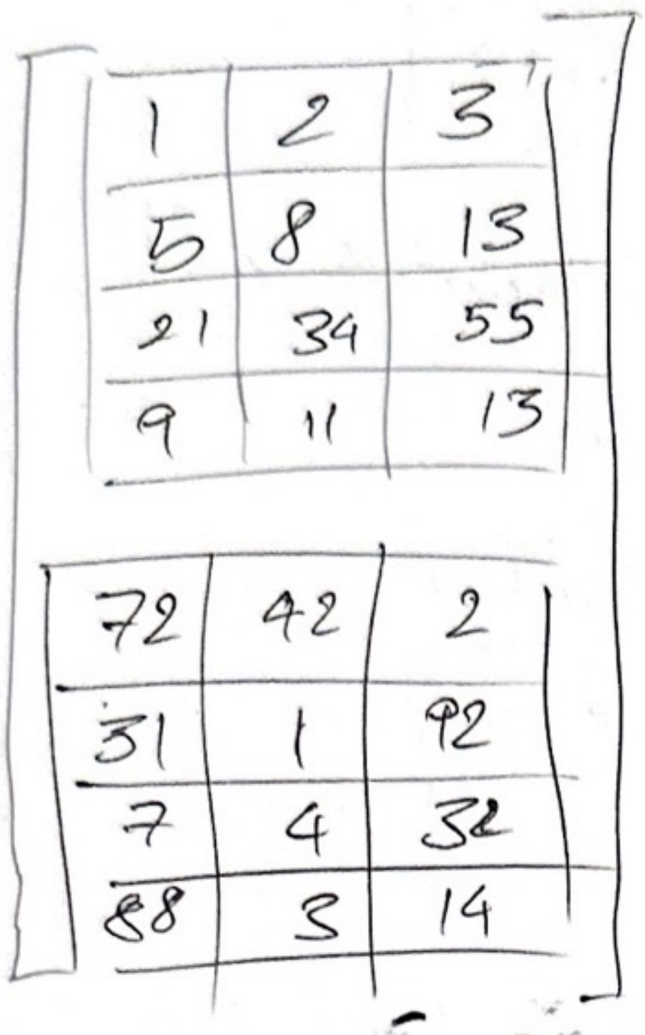
1	5
2	8
3	13

Shape [3, 2]

Stride : [1, 3]

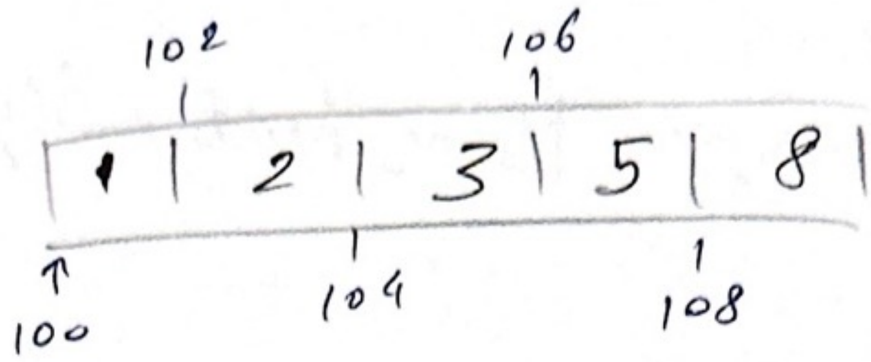
$$S\_tride = \begin{cases} S.tride[i] = \prod_{j=i+1}^{N} Shape[j] \\ S.tride[N] = 1 \end{cases}$$

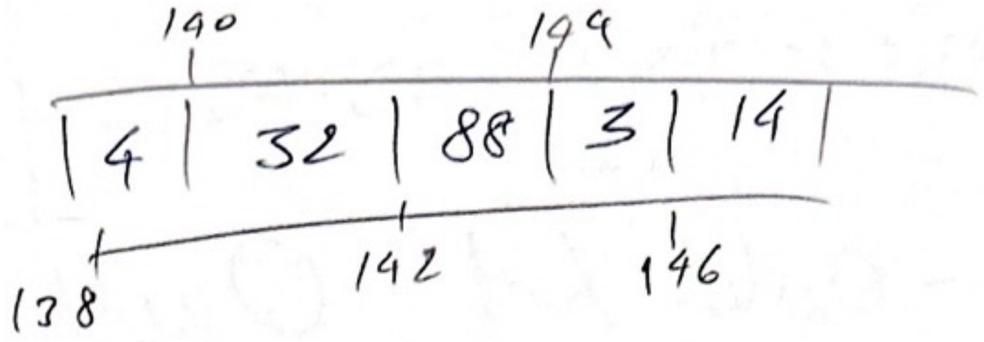
Tensor (3D)



1	2	3
5	8	13
21	34	55
9	11	13

72	42	2
31	1	92
7	4	32
88	3	14





Shape : [2, 4, 5]

Stride : [12, 3, 1]

Algorithm: 1 Flash Attention - 2 forward pass

Matrices : $Q, K, V \in R^{N \times d}$ in HBM, block size B_c, B_r

Divide Q into $T_r = \left\lceil \frac{N}{B_r} \right\rceil$ blocks $Q_1 \dots Q_T$ of size $B_r \times d$

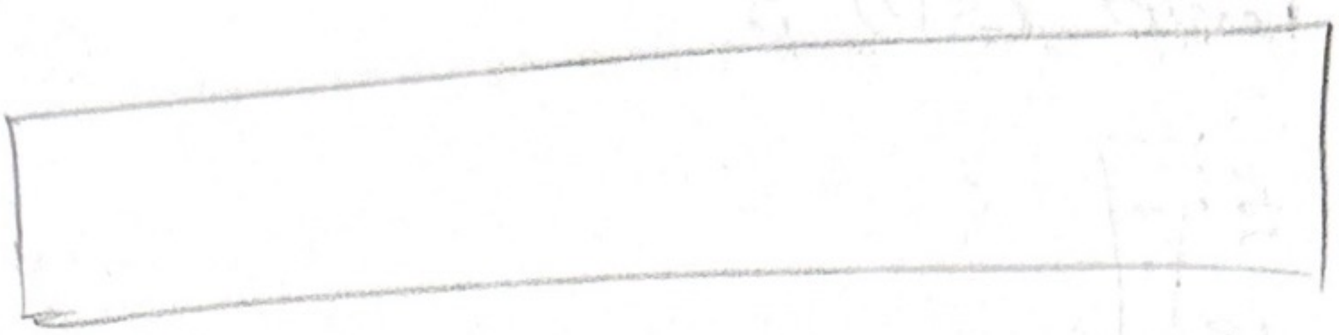
each, and divide K, V in to $T_c = \left\lceil \frac{N}{B_c} \right\rceil$ blocks

BATCH_SIZE × NUM_HEADS × NUM_BLOCKS_Q^SIZE

Q = [B_S, NumH, Seq_len, Head]

Q = [Batch_Size, Num_Heads, Seq_len, head_dim]

Tensor :



Size : [Batch-Size, Num heads, Seq-Len, Head-Dim]

Flash Attention

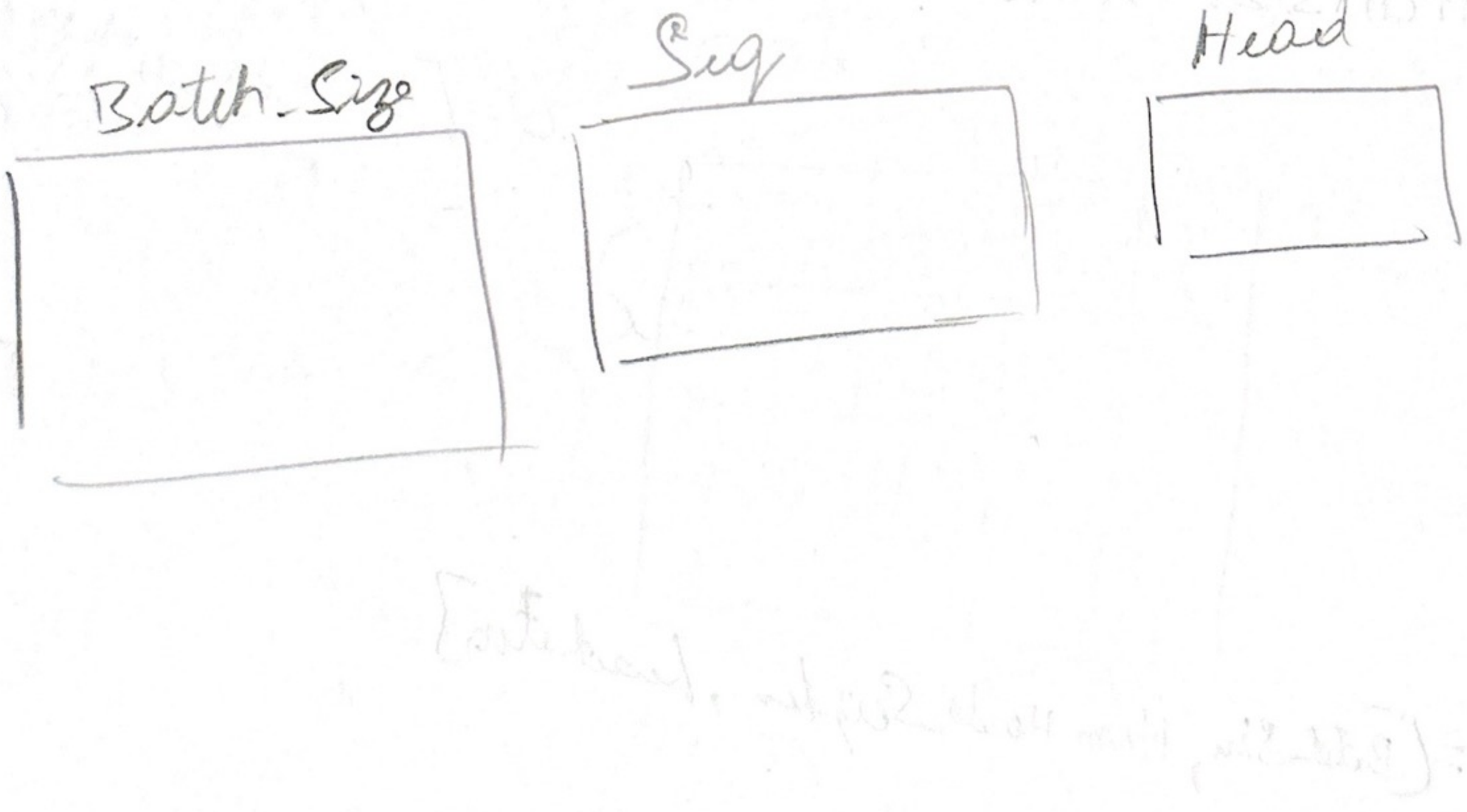
Outer loop ① Block

Inner loop all K Block

Program_id =

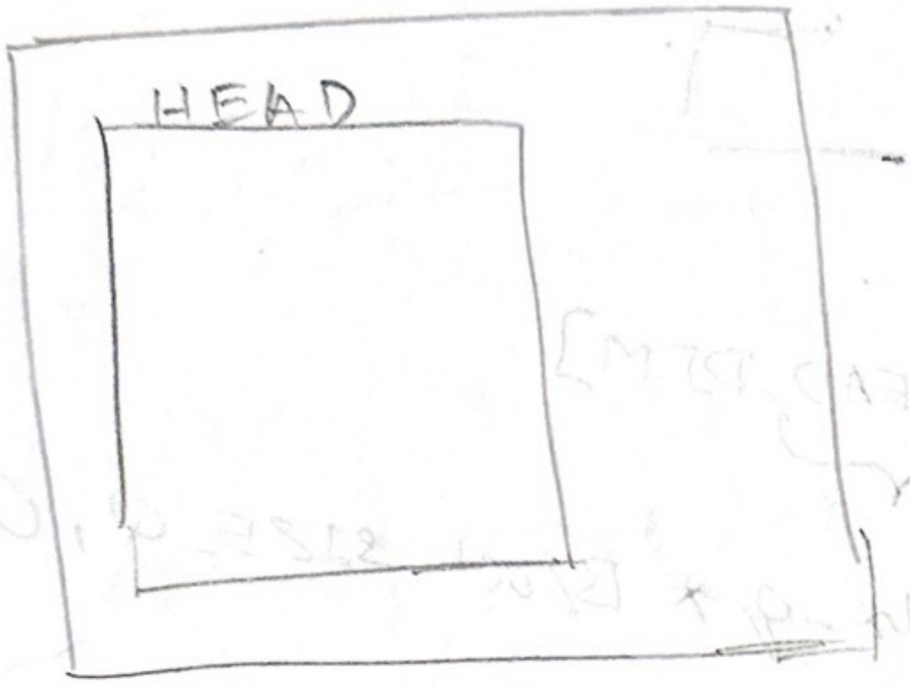
AXIS 0 = [SEQ-LEN / BLOCK_SIZE_Q]
2 4

AXIS 1 = [BATCH-SIZE * NUM_HEADS]



Program_Id : 0

Axis-0 = $\frac{Seq\_len}{Block\ Size}$

Axis-1 = 

BATCH_SIZE × NUM_HEAD

Q = [B, Num_heads, Seq_len, Head_dim]

↑ pre_sid

Q -> offset (Block_index_q * Block_size_Q, 0)

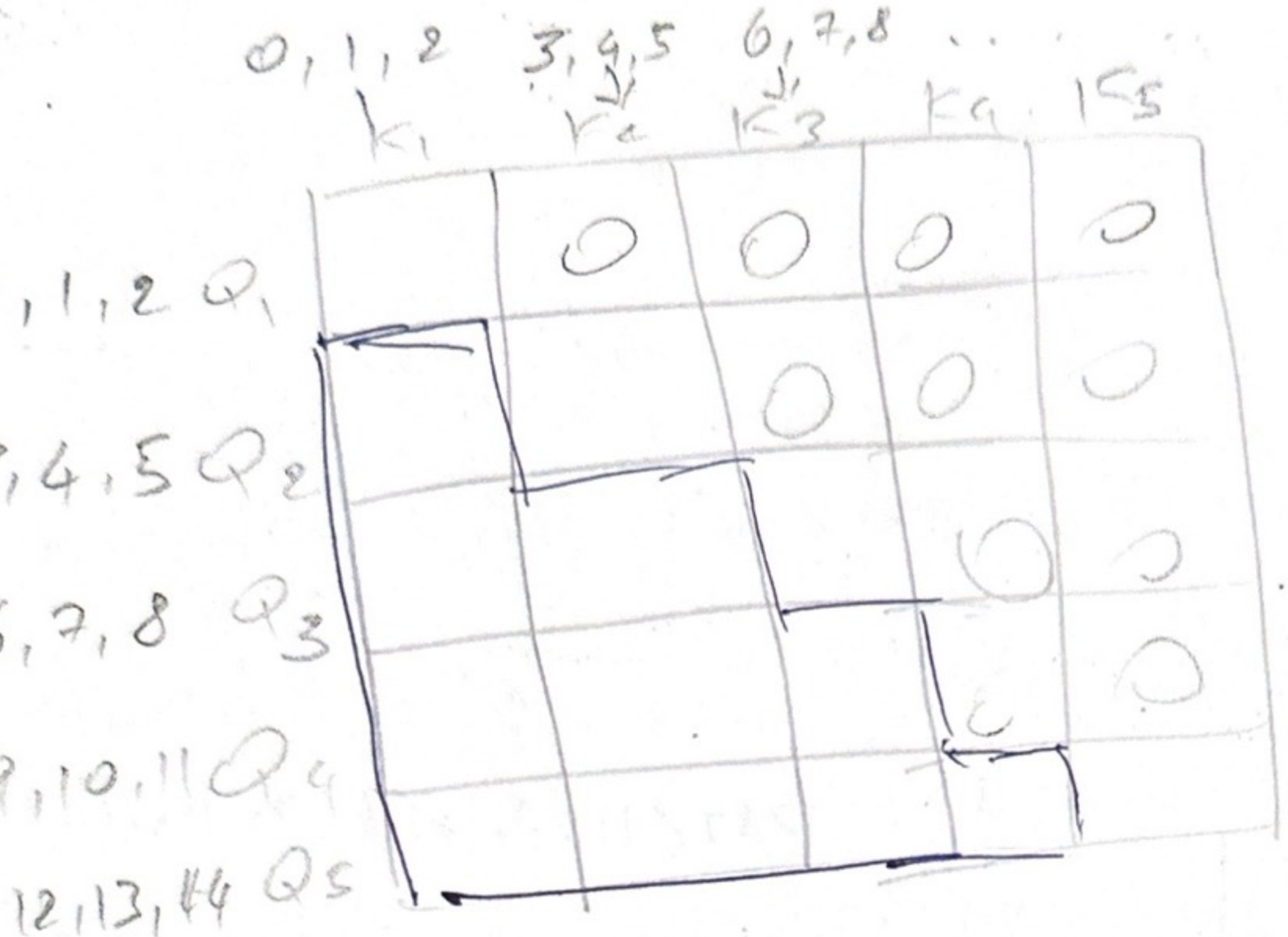
↓

QBlock

↓

Seq_Q [↓

order( )



Dzy = blk_idx_q * blk_size_q,
(block_idx_q + 1) * blk_size_q

Q B = [B, NUM_H, SEQ, HEAD_DIM]

↓

block_idx_q * BLOCK_SIZE_Q, 0

~~Inner loop~~

Inner loop

⊗ K, V → (Q ⊗ K)^T

↓

mov Each row → Softmax * ≠ l

O =

Non causal

causal Attion :

index <= indexQ

From derivatives to Jacobians

Derivative : scalar input, scalar output

$$f : R \to R$$

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

← how much the "output change"

← how much the "input" changes

$$f'(x) = \frac{\partial f(x)}{\partial x} = \frac{\partial y}{\partial x} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

$$f(x+h) \cong f'(x)\,h + f(x)$$

$$f(x+\Delta x) \cong f'(x)\,\Delta x + f(x) \qquad y^{NEW} \cong \frac{\partial y}{\partial x}\,\Delta x + y^{OLD}$$

$$f(x+\Delta x) \cong \frac{\partial y}{\partial x}\,\Delta x + f(x)$$

$$x^{new} \to x^{old} + \Delta x \implies y^{new} \cong \to y^{old} + \frac{\partial y}{\partial x}\,\Delta x$$

Chain Rule

$$Z = f(g(x))$$

$$x \xrightarrow{g} y \xrightarrow{f} z$$

$$x^{new} \to x^{old} + \Delta x \implies y^{new} \cong \to y^{old} + \frac{\partial y}{\partial x}\,\Delta x \qquad \Delta y$$

$$y^{NEW} \to y^{old} + \Delta y \implies z^{new} \cong \to z^{old} + \frac{\partial z}{\partial y}\,\Delta y$$

$$z^{new} \to z^{old} + \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x}\,\Delta x$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x}$$

Gradient : vector input, scalar output

$f: R^N \rightarrow R$

$f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = y$

$x^{NEW} \rightarrow x^{OLD} + \Delta x \Rightarrow y^{NEW} \rightarrow y^{old} + \frac{\partial y}{\partial x} \cdot \Delta x$

vector sum

dot prod

$\frac{\partial y}{\partial x}$ ← gradient $= \left(\frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \dots\right)$

$\frac{\partial y}{\partial x} \cdot \Delta x = \frac{\partial y}{\partial x_1} \cdot \Delta x_1 + \frac{\partial y}{\partial x_2} \cdot \Delta x_2 + \dots + \frac{\partial y}{\partial x_N} \cdot \Delta x_N$

partial derivative

Cross Entropy loss: $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow$ Loss = scalar output

Jacobian : vector input, vector output

$f: R^N \rightarrow R^M$

$f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$

$\text{Jacobian} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_N} \\ \vdots & & \\ \frac{\partial y_N}{\partial x_1} & \cdots & \frac{\partial y_M}{\partial x_N} \end{bmatrix}$

$X^{NEW} \rightarrow X^{OLD} + \Delta x \rightarrow y^{NEW} \xrightarrow{\approx} y^{OLD} + \frac{\partial y}{\partial x} \Delta x$

matrix-vector product

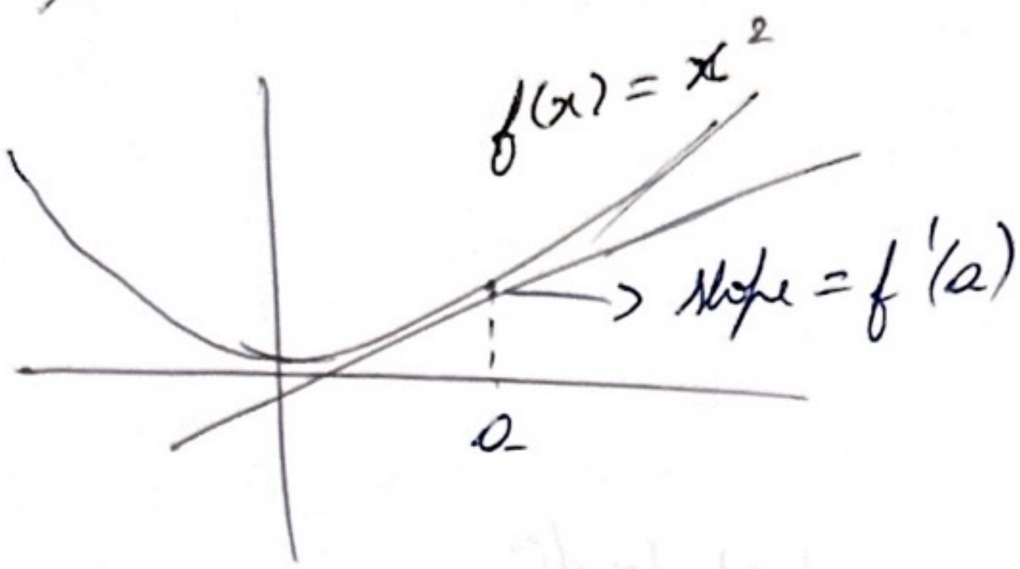
$(M \times N) \times (N \times 1) = (M \times 1)$

Derivatives:

$$f(x,y) = (x^2 - y^2, 2xy)$$

Derivatives = Slope

Derivatives ≠ Slopes



Linear Maps:

$$\begin{pmatrix} 1.33 & 1.17 \\ -0.73 & 0.75 \end{pmatrix} = (1.33 * 0.75) - (1.17 * -0.73)$$

Derivatives in 1D

$$f(x,y) = (x^2 - y^2, 3xy)$$

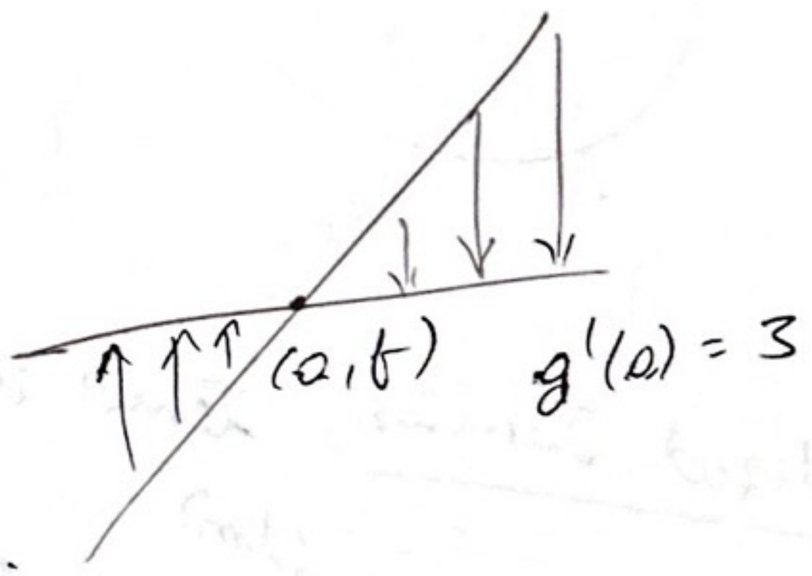
$$f(x,y) = (f_1(x,y), f_2(x,y))$$

$$f(x,y) = (x^2 - y^2, 3xy)$$

$$g(x) = f_1(x,b)$$

↓ input, ↓ output, → fixed

$$g'(a) = \left.\frac{\partial f_1}{\partial x}\right|_{(a,b)}$$



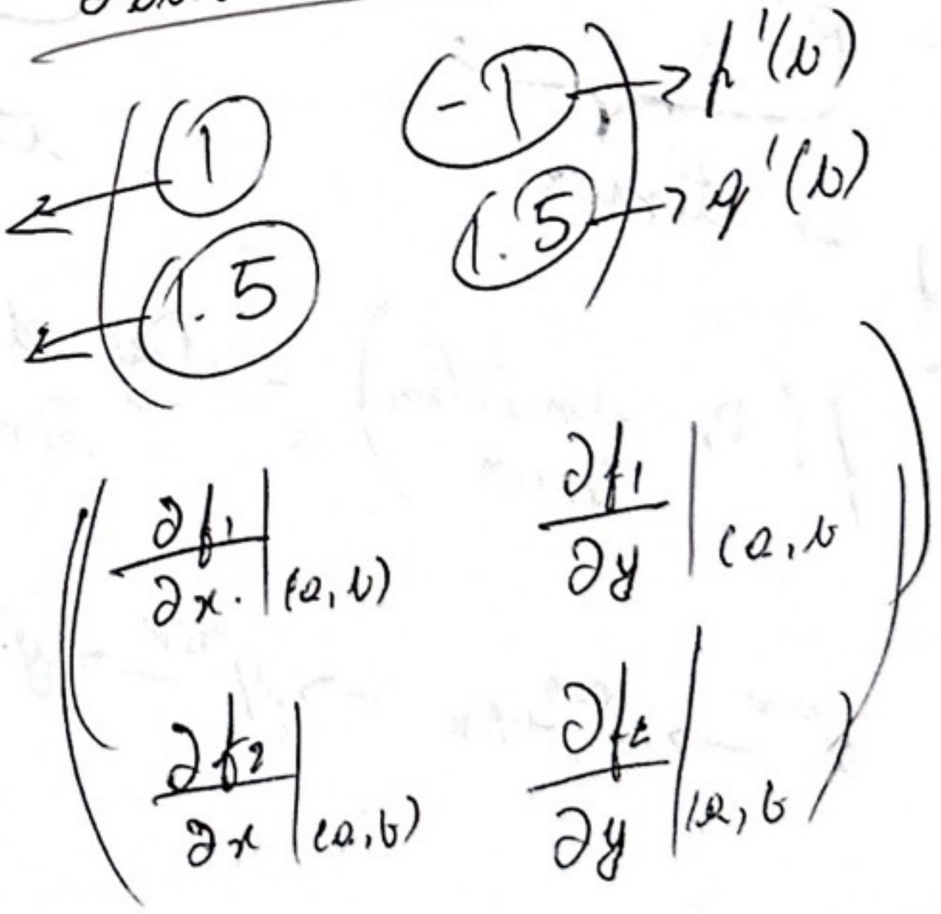
$g'(a)$

$h'(a)$

$$f(x,y) = (f_1(x,y), f_2(x,y))$$

$$f(x,y) = (x^2 - y^2, 3xy)$$

Jacobian Matrix

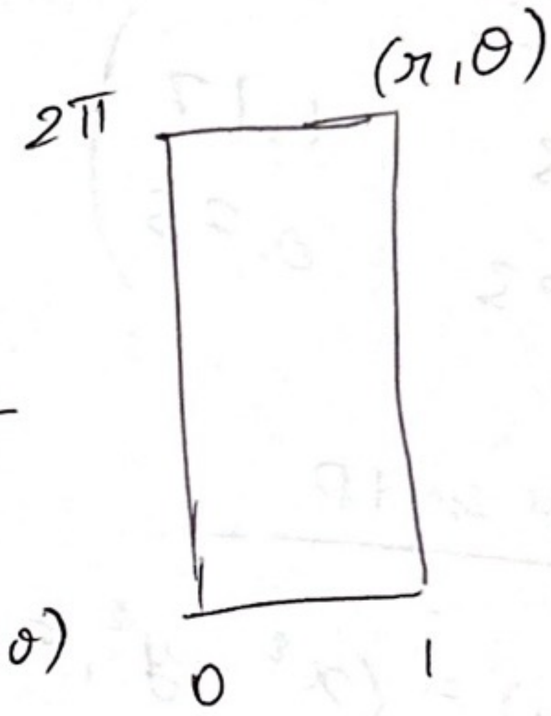


Jacobian determinant

$$\begin{vmatrix} \frac{\partial f_1}{\partial x}\Big|_{(a,b)} & \frac{\partial f_1}{\partial y}\Big|_{(a,b)} \\ \frac{\partial f_2}{\partial x}\Big|_{(a,b)} & \frac{\partial f_2}{\partial y}\Big|_{(a,b)} \end{vmatrix}$$

$$\iint_D f(x,y)\,dx\,dy$$

$$\iint_T f(g(r,\theta))\,|\det J|\,dr\,d\theta$$



Apply g

$g(r,\theta) = (r\cos\theta, r\sin\theta)$

Generalized Jacobian: tensor input, tensor output

$$f: \underbrace{R^{N_1 \times \cdots \times N_{D_x}}}_{\text{in tensor}} \xrightarrow{(\dim)} \underbrace{R^{M_1 \times \cdots M_{D_y}}}_{\text{output tensor}}$$

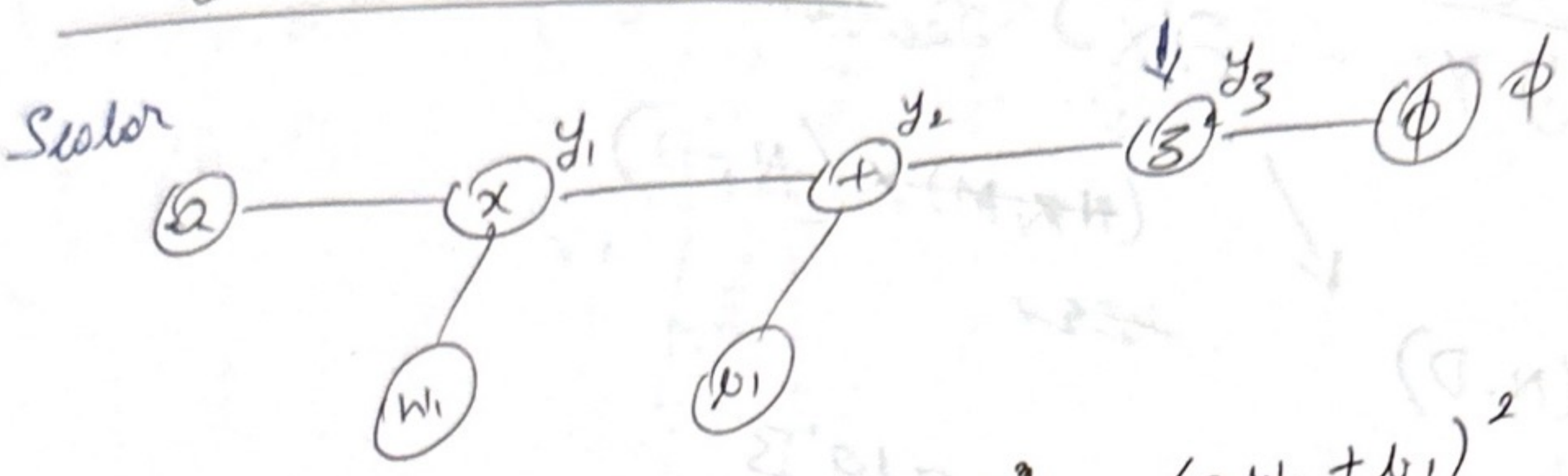
$f\left(D_x\text{-dimention tensor}\right) = D_y\text{-dimentional tensor}$

$$x^{NEW} \to x^{OLD} + \Delta x \Rightarrow y^{NEW} \to y^{OLD} + \underbrace{\frac{\partial y}{\partial x}}_{\text{Tensor product}} \Delta x$$

generalized Jacobian

$(M_1 \times \cdots M_{D_y}) \times (N_1 \times \cdots N_{D_x})$

Autograd with derivatives



$$\phi = y_3 = (y_2)^2 = (y_1 + b_1)^2 = (a w_1 + b_1)^2$$

$$\frac{\partial \phi}{\partial w_1} = 2(a w_1 + b_1)(a) = 2a(a w_1 + b_1)$$

$$\frac{\partial \phi}{\partial w_1} = \frac{\partial \phi}{\partial y_3} \cdot \frac{\partial y_3}{\partial y_2} \cdot \frac{\partial y_2}{\partial y_1} \cdot \frac{\partial y_1}{\partial w_1} =$$

$$= 1 \cdot 2y_2 \cdot 1 \cdot a = 2ay_2 = 2a(a w_1 + b_1)$$

$$\frac{\partial \phi}{\partial w_1} = \underbrace{\frac{\partial \phi}{\partial y_3} \cdot \frac{\partial y_3}{\partial y_2}}_{\frac{\partial \phi}{\partial y_2}} \cdot \frac{\partial y_2}{\partial y_1} \cdot \frac{\partial y_1}{\partial w_1}$$

$$\underbrace{\frac{\partial \phi}{\partial y_3} \cdot \frac{\partial y_3}{\partial y_2}}_{\frac{\partial \phi}{\partial y_2}}$$

$[N, M] \leftarrow Y = X W \rightarrow [d, M]$

$[1024, 2048] \leftarrow [N, d] \rightarrow [2048, 2048]$

Pytorch

$$\frac{\partial \phi}{\partial y_2} = \frac{\partial \phi}{\partial y_3} \frac{\partial y_3}{\partial y_2}$$

$$\frac{\partial \phi}{\partial y_1} = \frac{\partial \phi}{\partial y_2} \cdot \frac{\partial y_2}{\partial y_1}$$

$$\frac{\partial \phi}{\partial w_1} = \frac{\partial \phi}{\partial y_1} \cdot \frac{\partial y_1}{\partial w_1}$$

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial x} \cdot \frac{\partial y}{\partial x}$$

upstream gradient

local variable

Downstream gradient

$N = 1024$

$d = 1024$

$M = ~~1024~~ 2048$

CCIQWEB500

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial Y} \cdot \frac{\partial Y}{\partial x}$$ } Local Jawbian

~~(N, M) x (N, D)~~

(N, M) x (N, D)

$1024 \times 2048 \times 1024 \times 1024 = 2.19$ "B"

$$\begin{bmatrix} 1 & \cdots & 1024 \\ 1 & \cdots & 1024 \\ 1 & \cdots & 1024 \\ 1 & \cdots & 1024 \end{bmatrix}_{(N,D)} \times \begin{bmatrix} \cdot \\ \\ \\ \end{bmatrix}_{(D,M)} = \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}_{(N,M)}$$

Gradient of "Math Mal" operation

$Y = X W$ $\quad \frac{\partial \phi}{\partial Y} = \frac{\partial \phi}{\partial X} \frac{\partial \phi}{\partial W}$

$\frac{\partial \phi}{\partial X}$ $\quad X = [N, D]$ $\quad N = 1$ $\quad D = 3$

$W = [D, M]$ $\quad M = 4$

$Y = [N, M] = [1, 4]$

$X = [N, D]$

$W = [D, M]$

$y = [N, M]$

$X = [1, 2, 3]$

$$W = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

$$Y = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

$N = 1 \quad M = 4$

$D = 3$

$$Y = \underset{(1,3)}{\begin{bmatrix} X_{11} & X_{12} & X_{13} \end{bmatrix}} \times \underset{W\ (3,4)}{\begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \end{bmatrix}}$$

$$= \begin{bmatrix} (X_{11}W_{11} + X_{12}W_{21} + X_{13}W_{31}) & (X_{11}W_{12} + X_{12}W_{22} + X_{13}W_{32}) & (X_{11}W_{13} \dots) & (X_{11}W_{14} + X_{1} \end{bmatrix}$$

$y\ (1,4)$

$$\frac{\partial \phi}{\partial y} = \underset{[N,M]}{\begin{bmatrix} dy_{11} & dy_{12} & dy_{13} & dy_{14} \end{bmatrix}} \quad \underset{[M,D]}{W^T} = \underset{[N,D]}{\frac{\partial \phi}{\partial y}}$$

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial y} \cdot \frac{\partial y}{\partial x}$$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} W_{11} & W_{21} & W_{31} \\ W_{12} & W_{22} & W_{32} \\ W_{13} & W_{23} & W_{33} \\ W_{14} & W_{24} & W_{34} \end{bmatrix} = W^T$$

$$\frac{\partial \phi}{\partial W} = \underset{[D,M]}{X^T} \underset{[D,N]}{\frac{\partial \phi}{\partial y}} \quad \underset{[N,M]}{}$$

$N=2,\ M=4,\ D=3$

$X=[N,D]$ $\quad$ $X=\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix}$

$2,3$

~~$Y=[N,M]$~~

~~$2,4$~~

$W=[D,M]$

$3,4$

$$\begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \end{bmatrix}$$

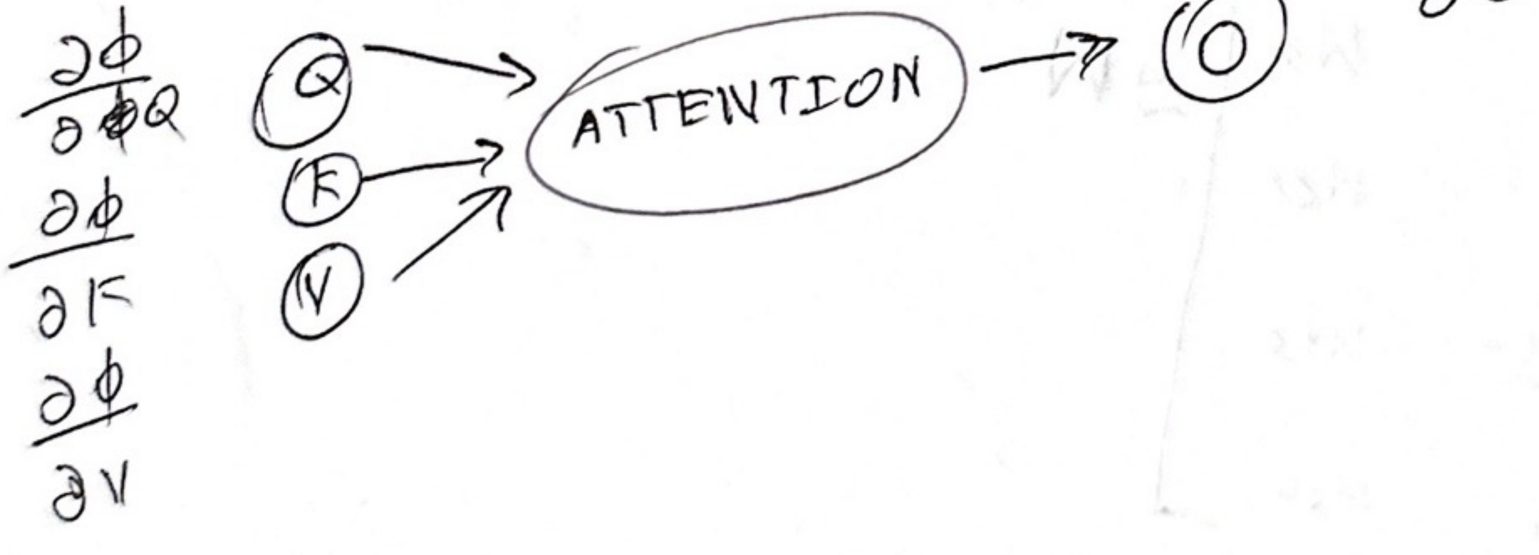
$Y=[N,M]$

$2,4$

$$\begin{bmatrix} (X_{11}W_{11}+X_{11}W_{21}+X_{11}W_{31}) & (X_{11}W_{12}+X_{11}W_{22}+X_{11}W_{32}) & (X_{11}W_{13}\cdots)\ldots(X_{13}*W_{11}) \\ & & \downarrow \\ & & \end{bmatrix}$$

Gradient through the Softmax

$S=QK^T$ $\quad$ $P=Softmax(S)$ $\quad$ $O=PV$



$P = Softmax\ Row(S)$

$S_i = S[i,:] \in R^N$

$P_i = Softmax(S_i) \in R^N$

$$Softmax(P_{ij}) = \frac{e^{S_{ij} - S_{iMAX}}}{\sum_{l=1}^{N} e^{S_{il} - S_{iMAX}}}$$

$\cos^2(x) + \sin^2(x) = 1$

$y = \cos^2(x)$

$$\frac{\partial y}{\partial x} = \frac{\partial(\cos^2(x))}{\partial(x)} \simeq$$

$$= \frac{\partial(1 - \sin^2(x))}{\partial x}$$

Jacobian

$$\frac{\partial \phi}{\partial S_i} = \frac{\partial \phi}{\partial P_i} \cdot \frac{\partial P_i}{\partial S_i}$$

$$\frac{\partial P_{ij}}{\partial S_{ik}} = \partial \left[\frac{e^{S_{ij}}}{\sum_{l=1}^{N} e^{S_{il}}} \right] \Big/ \partial S_{ik}$$

P_{11} P_{12} P_{13}

S_{11} S_{12} S_{13}

$$\left[\frac{f(x)}{g(x)}\right]' = \frac{f'(x)g(x) - g'(x)f(x)}{[g(x)]^2}$$

$$\frac{\partial \phi}{\partial S_i} = \frac{\partial \phi}{\partial P_i} \cdot \frac{\partial P_i}{\partial S_i}$$

$$\frac{\partial P_{ij}}{\partial S_{ik}} = \partial \left[\frac{e^{S_{ij}}}{\sum_{l=1}^{N} e^{S_{il}}} \right] \Big/ \partial S_{ik}$$

$P_{11} = S_{11}$

$$= (j = k) \rightarrow \frac{e^{S_{ij}} \left(\sum_{l=1}^{N} e^{S_{il}}\right) - e^{S_{ik}} \cdot e^{S_{ij}}}{\left(\sum_{l=1}^{N} e^{S_{il}}\right)^2}$$

$$= \frac{e^{S_{ij}} \left(\sum_{l=1}^{N} e^{S_{il}} - e^{S_{ik}}\right)}{\left(\sum_{l=1}^{N} e^{S_{il}}\right)^2}$$

$$\Rightarrow P_{ij} \cdot (1 - P_{ik})$$

$$= \frac{e^{S_{ij}}}{\sum_{l=1}^{N} e^{S_{il}}} \times \frac{\sum_{l=1}^{N} e^{S_{il}} - e^{S_{ik}}}{\sum_{l=1}^{N} e^{S_{il}}}$$

$$\frac{\partial P_{ij}}{\partial S_{ik}} = \frac{\partial \left[\frac{e^{S_{ij}}}{\sum_{l=1}^{N} e^{S_{il}}} \right]}{\partial S_{ik}}$$

$$\left[\frac{f(x)}{g(x)}\right] = \frac{f'(x)g(x) - g'(x)f(x)}{[g(x)]^2}$$

$j \neq k$

$$\frac{0 * g - e^{S_{ik}} e^{S_{ij}}}{\left[\sum_{l=1}^{N} e^{S_{il}}\right]^2}$$

$$\frac{\partial N}{\partial} \quad \frac{\partial P_{11}}{\partial S_{12}} = 0$$

$$= \frac{-e^{S_{ik}}}{\left(\sum_{l=1}^{N} e^{S_{il}}\right)} \cdot \frac{e^{S_{ij}}}{\left(\sum_{l=1}^{N} e^{S_{il}}\right)}$$

Softmax of k^{th} elem

Softmax of j^{th} elem

$$= -P_{ik} P_{ij}$$

$P = [P_{11} \quad P_{12} \quad P_{13}]$

$S = [S_{11} \quad S_{12} \quad S_{13}]$

$$\frac{\partial P_{ij}}{\partial S_{ik}} = \begin{cases} j = k = P_{ij}(1 - P_{ik}) \\ j \neq k = -P_{ik} P_{ij} \end{cases}$$

$[N, N]$

$$\frac{\partial P_{ij}}{\partial S_{ik}} = \begin{bmatrix} P_{11}(1-P_{11}) & -P_{11}P_{12} & -P_{11}P_{13} & \cdots & -P_{11}P_{1N} \\ -P_{12}P_{11} & P_{12}(1-P_{12}) & -P_{12}P_{13} & \cdots & -P_{12}P_{1N} \\ \vdots & & & & \\ -P_{1N}P_{11} & -P_{1N}P_{12} & -P_{1N}P_{13} & \cdots & P_{1N}(1-P_{1N}) \end{bmatrix}$$

$P \cdot P^T$

$= \{ \text{diag}(P_i) - P_i P_i^T \}$ As for Each i^{th} Row

$$P P^T = \underset{(3,1)}{\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}} \underset{(1,3)}{\begin{bmatrix} P_1 & P_2 & P_3 \end{bmatrix}} = \begin{bmatrix} P_1 P_1 & P_1 P_2 & P_1 P_3 \end{bmatrix}$$

Formula as Per Flash Attn Paper

$$y = \text{Softmax}(x)$$

$$\text{Jacobian} = \text{diag}(y) - y y^T$$

B1. Memory Efficient forward pass

$$O_i = P_i V = \sum_j P_{ij} V_j = Z$$

$$A \underset{(N,3)}{\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}} \times \underset{(3,4)}{\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix}}$$

$$\underset{(N,3)}{\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}} \times \underset{(4,4)}{\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix}}$$

$$O_i = \sum_j e$$

$$O_1 = a_{11} b_{[1:3} + a_{12} b_{[2,:]} + a_{13} b_{3[3,0]}$$

$$= \sum_{j=1}^{3} a$$

$$O_1 = \underset{(N,4)}{\begin{bmatrix} (a_1 b_{11} + a_{12} b_{21} + a_{13} b_{31}) \\ (a_{11} b_{12} + a_{12} b_{22} + a_{13} b_{32}) \\ (a_1 b_{13} + a_{12} b_{23} + a_{13} b_{33}) \\ (a_{11} b_{14} + a_{12} b_{24} + a_{13} b_{34}) \\ (a_{11} b \end{bmatrix}}$$

$$A = \begin{bmatrix} [a_{11} \; a_{12} \; a_{13}] \\ \downarrow \\ 5 \end{bmatrix} \quad (N,3)$$

$$\times \rightarrow$$

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix} \quad (3,N)$$

$$O = \begin{bmatrix} (a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31}) \\ (a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32}) \\ (a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33}) \\ (a_{11}b_{14} + a_{12}b_{24} + a_{13}b_{34}) \end{bmatrix}$$

$$O_1 = a_{11}b_{1:} + a_{12}b_{2:} + a_{13}b_{3:}$$

$$= \sum_{j=1}^{3} a_{ij} b_{j:}$$

$$O_i = P_{i:} V = \sum_j P_{ij} V_j = \sum_j \frac{e^{q_i^T k_j}}{\angle_i} V_j$$

$$O = PV$$

~~$Y = WX$~~ $Y = XW$

$$\frac{\partial \phi}{\partial X} = \frac{\partial \phi}{\partial Y} W^T$$

$$\frac{\partial \phi}{\partial W} = X^T \frac{\partial \phi}{\partial Y}$$

$$\frac{\partial \phi}{\partial V} = P^T \frac{\partial \phi}{\partial O}$$

$$\partial V = P^T \partial O \rightarrow \text{Proof}$$

$$\frac{\partial \phi}{\partial P} = \frac{\partial \phi}{\partial O} V^T = \partial P = \partial O V^T$$

$$dV = F \, dO$$

$$dV_J = \sum F_{Ji} * dO_i =$$

$*$: Scalor Vector Multiplication

$$Y = XW$$

$$\frac{\partial \phi}{\partial X} = \frac{\partial \phi}{\partial Y} \omega^T$$

$$\frac{\partial \phi}{\partial \omega} = X^T \frac{\partial \phi}{\partial Y}$$

$$O = PV$$

$$\frac{\partial \phi}{\partial V} = P^T \frac{\partial \phi}{\partial O}$$

$$dV = P^T dO$$

$$\frac{\partial \phi}{\partial P} = \frac{\partial \phi}{\partial O} V^T$$

$$dP = dO V^T$$

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial y} \cdot \frac{\partial y}{\partial x}$$

S6257786
nigrethyoader@proton.me

Pytorch

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \end{bmatrix} \quad (1,3)$$

$$\begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{33} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \end{bmatrix} \quad (3,4)$$

$$\begin{bmatrix} (x_{11}W_{11} + x_{12}W_{21} + x_{13}W_{31}) \end{bmatrix}$$

Pytorch

$$\frac{\partial \phi}{\partial y} = \begin{bmatrix} dy_{11} & dy_{12} & dy_{13} & dy_{14} \end{bmatrix}$$

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial y} \times \frac{\partial y}{\partial x}$$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} W_{11} & W_{21} & W_{31} \\ W_{12} & W_{22} & W_{32} \\ W_{13} & W_{23} & W_{33} \\ W_{14} & W_{24} & W_{34} \end{bmatrix} = W^T$$

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial y} \cdot \frac{\partial y}{\partial x} \quad \underset{[N,M]}{} \quad \underset{[M,D]}{} \quad \underset{[N,D]}{}$$

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial y} W^T$$

$$\frac{\partial \phi}{\partial w} = x^T \frac{\partial \phi}{\partial w}$$

$$\frac{\partial \phi}{\partial w} = x^T \frac{\partial \phi}{\partial y}$$

$$[D,M] \quad [D,N] \quad [N,M]$$

Gradient through Softmax

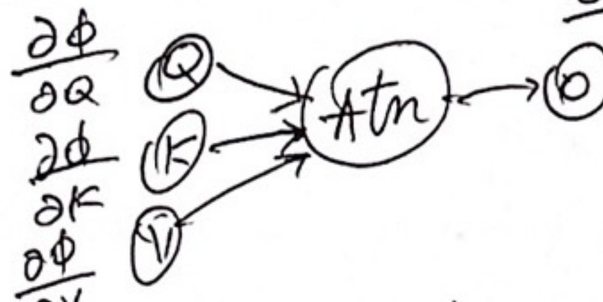
$$S = QK^T \qquad P = Softmax(S_i) \qquad O = PV$$

$$S_i = S[i,:] \in R^N$$

$$P_i = Softmax(S_i) \in R^N$$

$$Softmax(P_{ij}) = \frac{e^{S_{ij}}}{\sum_{t=1}^{N} e^{S_{it}}}$$

$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - g'(x)f(x)}{(g(x))^2}$$



$$\frac{\partial \phi}{\partial O} \quad \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \end{bmatrix}$$

$$\frac{\partial \phi}{\partial S_i} = \frac{\partial \phi}{\partial P_i} \cdot \frac{\partial P_i}{\partial S_i}$$

$$\frac{\partial P_{ij}}{\partial S_{ik}} = \frac{\partial \left[\frac{e^{S_{ij}}}{\sum_{t=1}^{N} e^{S_{it}}} \right]}{\partial S_{ik}} =$$

$$\frac{\partial \phi}{\partial s_i} = \frac{\partial \phi}{\partial \Pi} \cdot \frac{\partial \Pi}{\partial s_i}$$

$$\frac{\partial \Pi}{\partial s_i}$$

$$\Pi = \exp(s_k)$$

$$\frac{\partial \Pi}{\partial x} = \frac{\partial (\exp(s_k))}{\partial x}$$

$$\frac{\partial (1 - \exp(s_k))}{\partial x}$$

$$\frac{\partial \left[\frac{e^{s_i}}{\sum_{i=1}^{N} e^{s_i}} \right]}{\partial x} = s + \lambda$$

$$\frac{e^{s_i}}{\left(\sum_{i=1}^{N} e^{s_i}\right)^2} \times \sum_{i=1}^{N} e^{s_i} \cdot e^{s_k}$$

$$\frac{e^{s_i}}{\sum_{i=1}^{N} e^{s_i}} \times \frac{\sum_{i=1}^{N} e^{s_i} \cdot e^{s_k}}{\sum_{i=1}^{N} e^{s_i}}$$

P_{ij} $\quad (1 - P_{ik})$

logsumExp

$$\text{Softmax}(x_i) = \exp(x_i - m_i - \log(l_i))$$

$$= \frac{\exp(x_i - m_i)}{\exp^{\log(l_i)}}$$

$$= \frac{\exp(x_i - m_i)}{l_i}$$

$$\exp(a+b) = \exp(a) \cdot \exp(b)$$

$$\exp(a-b) = \frac{\exp(a)}{\exp(b)}$$

THE ONLINE SOFMAX

$m_0 = -\infty$

$l_0 = 0$

for $i = 1$ to N

$m_i = \max(m_{i-1}, x_i)$

$l_i = l_{i-1} \cdot e^{m_{i-1} - m_i} + e^{x_i - m_i}$

for $k = 1$ to N

$x_k \leftarrow \frac{e^{x_k - m_N}}{l_N}$

$m_0 = \begin{bmatrix} -\infty \\ -\infty \end{bmatrix}$

$l_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$O_0 = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}$

STEP 1

$m_1 = \max(\pi \max(Q_1 \overset{S_{11}}{\downarrow} K_1^T), m_0)$

$S_1 = Q_1 K^T$

$l_1 = \pi \text{sum}[\exp(S_1 - m_1)] + l_0 \cdot \exp(m_0 - m_1)$

$P_{11} = \exp(S_1 - m_1)$

$O = \text{diag}(\exp(m_0 - m_1)) O_0 + P_{11} V_1$

STEP 2

$m_2 = \max(\pi \max(Q_1 \overset{S_{12}}{K_2^T}), m_1)$

$S_2 = Q_1 K_2^T$

$l_2 = \pi \text{sum}[\exp(S_2 - m_2)] + l_1 \cdot \exp(m_1 - m_2)$

$P_{12} = \exp(S_2 - m_2)$

$O_1 = \text{diag}(\exp(m_1 - m_2)) O_1 \overset{\downarrow}{} + P_{12} V_2$

$$O_1 = \begin{bmatrix} o_{11} & o_{12} & \dots & o_{1,128} \\ o_{21} & o_{22} & \dots & o_{2,128} \end{bmatrix} \times \begin{bmatrix} \exp(m_1 - m_2)_1 & 0 \\ 0 & \exp(m_1 - m_2)_2 \end{bmatrix}$$

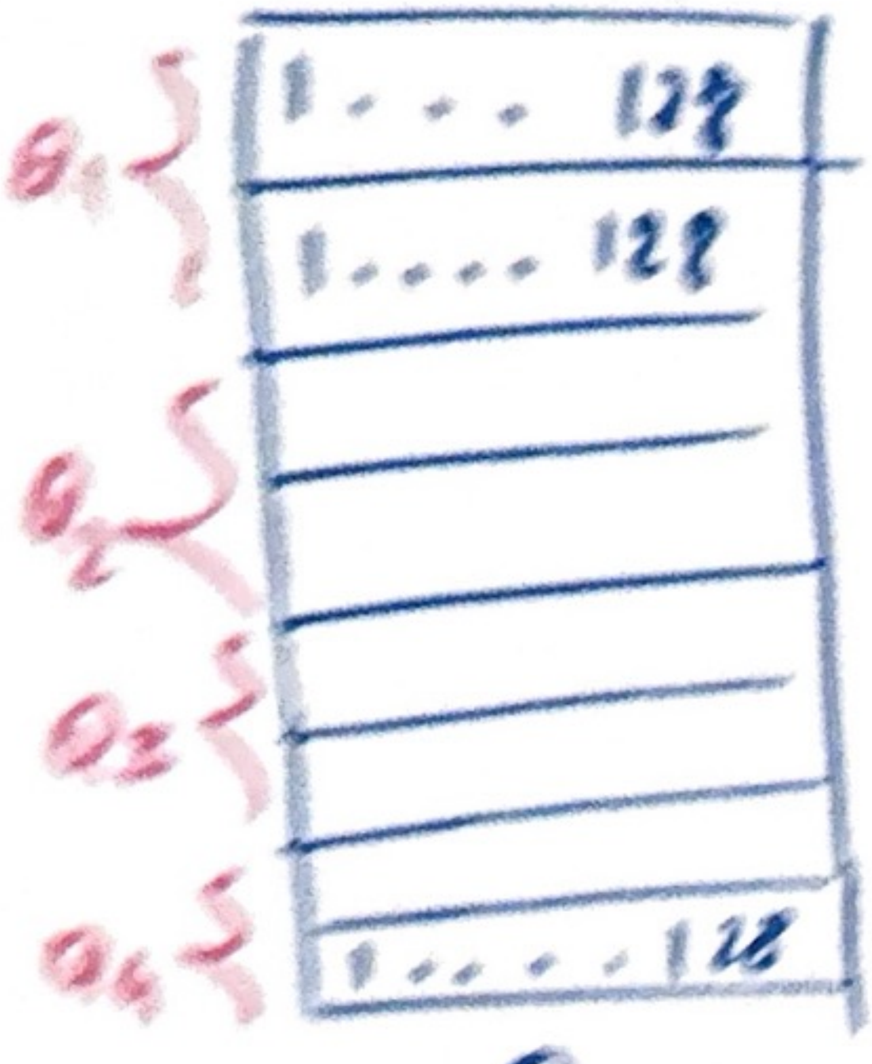
$$= \begin{bmatrix} o_{11}\exp(m_1 - m_2)_1 + 0 & o_{12}\exp(m_1 - m_2) + 0 \dots \\ 0 + o_{21}\exp(m_1 - m_2) & 0 + o_{22}\exp(m_1 - m_2) \dots \end{bmatrix}$$

$$\begin{bmatrix} l_4^{(1)} \\ l_4^{(2)} \end{bmatrix} = l_4 \quad O_4 = \begin{bmatrix} [1 \dots 128] \\ [1 \dots 128] \end{bmatrix}$$

$$\left(\begin{bmatrix} l_4^{(1)} & 0 \\ 0 & l_4^{(2)} \end{bmatrix} \right)^{-1} = \begin{bmatrix} \frac{1}{l_4^{(1)}} & 0 \\ 0 & \frac{1}{l_4^{(2)}} \end{bmatrix}_{(2,2)} \times \begin{bmatrix} [1 \dots 128] \\ [1 \dots 128] \end{bmatrix}_{(2,128)} = \begin{bmatrix} [1 \dots 128] \\ [1 \dots 128] \end{bmatrix}$$

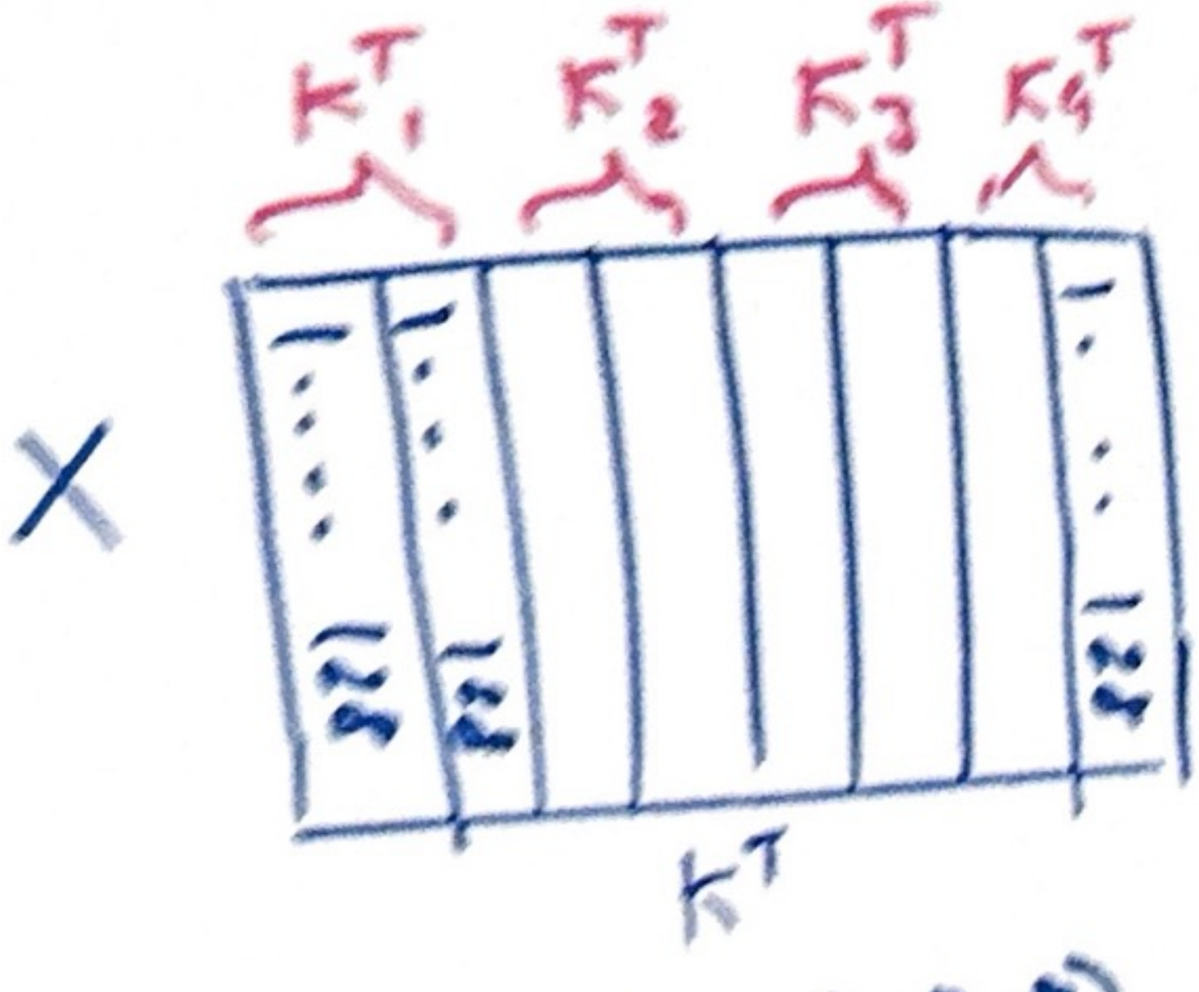
STEP 5

$$O_5 = [\text{diag}(l_4)]^{-1} O_4$$



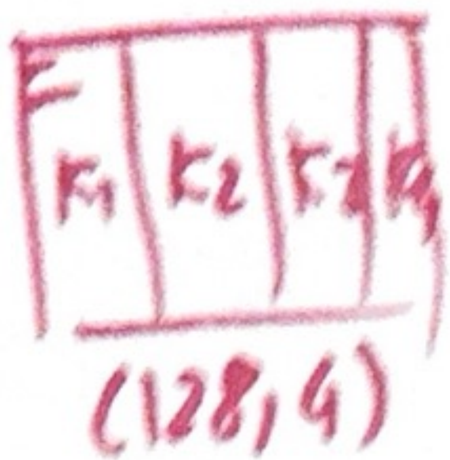
ORIG = (8, 128)

BLOCK = (4, 128)



ORIG = (128, 8)

BLOCK : (128, 4)



(2, 128) × (128, 2) = (2, 2)

$S =$

$Q_1 K_1^T$	$Q_1 K_2^T$	$Q_1 K_3^T$	$Q_1 K_4^T$
$Q_2 K_1^T$	$Q_2 K_2^T$	$Q_2 K_3^T$	$Q_2 K_4^T$
$Q_3 K_1^T$	$Q_3 K_2^T$	$Q_3 K_3^T$	$Q_3 K_4^T$
$Q_4 K_1^T$	$Q_4 K_2^T$	$Q_4 K_3^T$	$Q_4 K_4^T$

ORIG = (8, 8)

BLOCK = (4, 4)

S_{11} → $(2\times128)\times(128\times2)=(2,2)$

$$S=\begin{array}{|c|c|c|c|}\hline Q_1K_1^T & Q_1K_2^T & Q_1K_3^T & Q_1K_4^T \\ \hline Q_2K_1^T & Q_2K_2^T & Q_2K_3^T & Q_2K_4^T \\ \hline Q_3K_1^T & Q_3K_2^T & Q_3K_3^T & Q_3K_4^T \\ \hline Q_4K_1^T & Q_4K_2^T & Q_4K_3^T & Q_4K_4^T \\ \hline\end{array}$$

$$S_{11}=\begin{bmatrix}a & b\\ c & d\end{bmatrix}\xrightarrow{S^*}\begin{array}{|c|c|}\hline \exp(a-a) & \exp(b-a)\\ \hline \exp(c-d) & \exp(d-d)\\ \hline\end{array}$$

LOCAL MAX

SOFTMAX* $\Longrightarrow P=\exp(x_i-x_{max})$

$(2,128)\times(128\times2)=(2,2)$

$$\begin{array}{|c|c|c|c|}\hline P_{11} & P_{12} & P_{13} & P_{14}\\ \hline P_{21} & P_{22} & P_{23} & P_{24}\\ \hline P_{31} & P_{32} & P_{33} & P_{34}\\ \hline P_{41} & P_{42} & P_{43} & P_{44}\\ \hline\end{array}$$

ORIGINAL $=(8,8)$

BLOCK $=(4,4)$

SOFTMAX:

$$\frac{\exp(x_i-x_{max})}{\sum_{j=1}^{N}\exp(x_i-x_{max})}\rightarrow(\text{LOCAL MAX})$$

$$S^*(S_{ij})=\exp\left[S_{ij}-\text{ROW MAX}(S_{ij})\right]$$

$*\ V$

$(8,128)$

$(4,128)$

$$\begin{array}{|c|}\hline 1\ldots\ldots128\\ \hline 1\ldots\ldots128\\ \hline \\ \hline \\ \hline \\ \hline \\ \hline 1\ldots\ldots128\\ \hline\end{array}\begin{matrix}\}V_1\\ \\ \}V_2\\ \\ \}V_3\\ \\ \}V_4\\ \\ \end{matrix}$$

$$O=\begin{array}{|c|}\hline P_{11}V_1+P_{12}V_2+P_{13}V_3+P_{14}V_4\\ \hline \ldots\ldots\\ \hline \ldots\ldots\\ \hline \\ \hline \\ \hline \\ \hline\end{array}\rightarrow(2,128)$$

PSUDO CODE

```
FOR EACH BLOCK Q_i
    O_i = ZERO(2,128)
    FOR EACH B. K_J
        P_iJ = S*(Q_i K_J^T)
        O_i ← O_i + P_iJ V_J
    END FOR
END FOR
```

S_{11} → $(2\times128)\times(128\times2)=(2,2)$

$$S=\begin{array}{|c|c|c|c|}\hline Q_1K_1^T & Q_1K_2^T & Q_1K_3^T & Q_1K_4^T \\ \hline Q_2K_1^T & Q_2K_2^T & Q_2K_3^T & Q_2K_4^T \\ \hline Q_3K_1^T & Q_3K_2^T & Q_3K_3^T & Q_3K_4^T \\ \hline Q_4K_1^T & Q_4K_2^T & Q_4K_3^T & Q_4K_4^T \\ \hline\end{array}$$

$(2,128)\times(128\times2)=(2,2)$

SOFTMAX* $\Longrightarrow P=\exp(x_i-x_{max})$

$$P=\begin{array}{|c|c|c|c|}\hline P_{11} & P_{12} & P_{13} & P_{14} \\ \hline P_{21} & P_{22} & P_{23} & P_{24} \\ \hline P_{31} & P_{32} & P_{33} & P_{34} \\ \hline P_{41} & P_{42} & P_{43} & P_{44} \\ \hline\end{array}$$

ORIGINAL $=(8,8)$

BLOCK $=(4,4)$

$S_{11}=\begin{bmatrix}a & b\\ c & d\end{bmatrix}\xrightarrow{S^*}\begin{array}{|c|c|}\hline \exp(a-a) & \exp(b-a) \\ \hline \exp(c-a) & \exp(d-d) \\ \hline\end{array}$ LOCAL MAX

SOFTMAX: $\dfrac{\exp(x_i-x_{max})}{\sum_{j=1}^{N}\exp(x_i-x_{max})}$ → (LOCAL MAX)

$S^*(S_{ij})=\exp\left[S_{ij}-\text{ROW MAX}(S_{ij})\right]$

$*\ V$ $(8,128)$ $(4,128)$

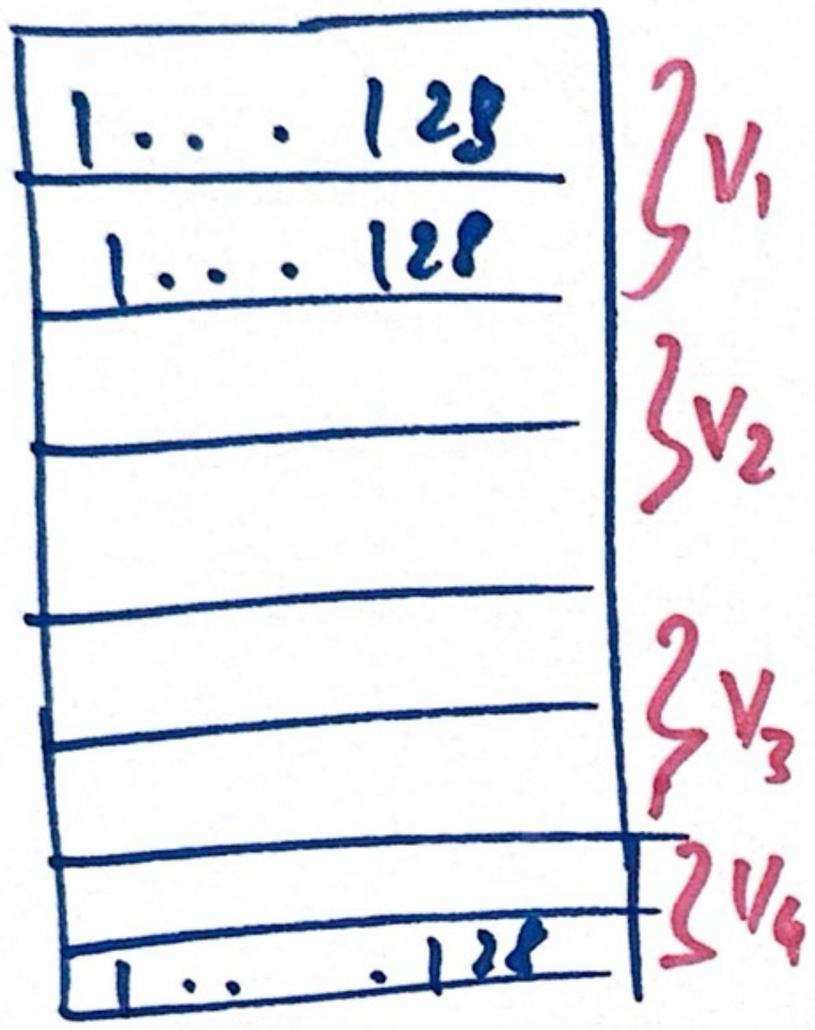
$$\begin{array}{|c|}\hline 1\ldots128 \\ \hline 1\ldots128 \\ \hline \\ \hline \\ \hline \\ \hline \\ \hline 1\ldots128 \\ \hline\end{array}\begin{matrix}\}V_1\\ \}V_2\\ \}V_3\\ \}V_4\end{matrix}$$

$O=$

$$\begin{array}{|c|}\hline P_{11}V_1+P_{12}V_2+P_{13}V_3+P_{14}V_4 \\ \hline \ldots. \\ \hline \ldots. \\ \hline \\ \hline \\ \hline \\ \hline\end{array}\rightarrow(2,128)$$

PSUDO CODE

```
FOR EACH BLK Qi
  Oi = ZERO(2,128)
  FOR EACH BLK Kj
    Pij = S*(Qi Kj^T)
    Oi ← Oi + Pij Vj
  END FOR
END FOR
```



$$O = \begin{matrix} (Q_1K_1^T)V_1 + (Q_1K_2^T)V_2 + (Q_1K_3^T)V_3 + (Q_1K_4^T)V_4 \\ (Q_2K_1^T)V_1 + (Q_2K_2^T)V_2 + (Q_2K_3^T)V_3 + (Q_2K_4^T)V_4 \\ (Q_3K_1^T)V_1 + (Q_3K_2^T)V_2 + (Q_3K_3^T)V_3 + (Q_3K_4^T)V_4 \\ (Q_4K_1^T)V_1 + (Q_4K_2^T)V_2 + (Q_4K_3^T)V_3 + (Q_4K_4^T)V_4 \end{matrix}$$

$[(2\times128)\times(128\times2)]\times(2\times128) = (2\times2)\times(2\times128) = (2\times128)$

EACH BLOCK OF ~~OUTER~~ OUTPUT "O" IS MADE OF TWO ROWS!

ORIG = (8,128)

BLOCK = (4,128

PSEUDOCODE

```
FOR EACH BLOCK Qi
    Oi = ZEROS(2,128) // OUTPUT INITALLY MADE OF ZEROS
    FOR EACH BLOCK KJ
        Oi <= Oi + (Qi KJ^T) VJ
    END FOR
END FOR
```