5.4) Division (for division part in softmax - Algorithm 8)

* $\Pi_{Div}(\{P_0, P_1\}, P_2)$

⊙ Input: P_0, P_1 hold $(\langle x\rangle_0^L, \langle y\rangle_0^L)$ and $(\langle x\rangle_1^L, \langle y\rangle_1^L)$, resp.

⊙ Output: P_0, P_1 get $\langle x/y\rangle_0^L$ and $\langle x/y\rangle_1^L$

⊙ Common Randomness: $P_j, j \in \{0,1\}$ hold 'ℓ' shares 0 over $\mathbb{Z}_L$ denote by $w_{i,0}$ and $w_{i,1}$ for all $i \in [\ell]$ resp. They additionally also hold another share of 0 over $\mathbb{Z}_L$ denoted by s_0 and s_1.

⊙ Steps:

1) Set $u_\ell = 0$ and for $j \in \{0,1\}$, P_j holds $\langle u_\ell\rangle_j^L$ holds $\langle u_\ell\rangle_j^L$ (through common randomness).
2) for $i = \{\ell-1, \ldots, 0\}$ do
3) $P_j, j \in \{0,1\}$ computes
$$\langle z_i\rangle_j^L = \langle x\rangle_j^L - \langle u_{i+1}\rangle_j^L - 2^i\langle y\rangle_j^L + w_{i,j}$$
4) P_0, P_1, P_2 run $\Pi_{DReLU}(\{P_0, P_1\}, P_2)$ with $P_j, j \in \{0,1\}$ having input $\langle z_i\rangle_j^L$ and P_0, P_1 learn $\langle B_i\rangle_0^L$ and $\langle B_i\rangle_1^L$ resp.

5) P_0, P_1, P_2 call $\Pi_{MATMUL}(\{P_0, P_1\}, P_2\}$ with $P_j, j \in \{0,1\}$ having input $(\langle B_i \rangle_j^L, \langle 2^i y \rangle_j^L$ and P_0, P_1 learn $\langle v_i \rangle_0^L$ and $\langle v_i \rangle_1^L$ resp.

6) $P_j, j \in \{0,1\}$ compute $\langle k_i \rangle_j^L = 2^i \cdot \langle B_i \rangle_j^L$

7) For $j \in \{0,1\}$, P_j computes
$$\langle u_i \rangle_j^L = \langle u_{i+1} \rangle_j^L + \langle v_i \rangle_j^L$$

8) end for

9) For $j \in \{0,1\}$, P_j outputs $\langle q \rangle_j^L = \sum_{i=0}^{L-1} \langle k_i \rangle_j^L + s_j$

⊙ Explanation: it implements long division

+ for each bit in range(L)?

+ step 3 computes the current dividend $(x - u_{i+1})$ by subtracting the 2^i-th mutiple of the divisor i.e $2^i y$ where as $\langle u_{i+1} \rangle$ keeps track of the total sum that's already been subtracted from x
$$z_i = x - u_{i+1} - 2^i y$$

+ To check whether 2^i-th multiple of y is the correct mutiple of y that should be subtracted from x, step 3 uses Π_{DRELU} with input z and gets the output as B.

Π_{DReLU} determines whether

$$z = (x - u_{i+1}) - 2^i y > 0 \text{ or not.}$$

If $z > 0$, then we are subtracting the correct multiple of the divisor 'y' i.e $(x - u_{i+1}) > 2^i y$.

Suppose the current dividend is 9, then at $i = 1$, we will have $2^i y = 2^1 \times 3 = 6$, the first position where $z_1 = 9 - 6 = 3 > 0$. This means 6 is the correct multiple of 3 that we can subtract from 9.

Note that i runs from $l-1$ to 0, this ensures that we get the maximum value of $2^i y$ that we can subtract from current dividend $(x - u_{i+1})$ when $z_i > 0$. For instance for the above example at $i = 2$, we had $z_2 = 9 - 4 \times 3 = -3 < 0$ which is less than 0. Hence, at $i = 1$, we get the maximum value i.e $2^1 \times 3 = 6$ that we can subtract from current dividend $(x - u_{i+1})$.

For the correct multiple of divisor y, we will have $z > 0 \Rightarrow B = 1$.

For incorrect $z < 0 \Rightarrow B = 0$.

+ Then, step 5 does a multiplication between B and $2^i y$ and gets $V = B \cdot 2^i y$

+ Step 6 computes the quotient $k_i = 2^i \cdot B$ If $B = 1$, we got the correct multiple of divisor y, hence set $k_i = 2^i$ else 0.

+ Step 7 computes the total sum u_i, the sum of correct multiple of $2^i y$'s that have been subtracted so far (till iteration i) from original dividend (x). Both x and u_{i+1} give the current dividend at iteration i of the for loop. ($u_i = u_{i+1} + v_i$)

+ Step 9 outputs the quotient

$$q = \sum_{i=0}^{n-1} \langle k_i \rangle + s$$

The idea here is that the quotient can be written as the summation of powers of 2.

⊙ Example: $\overline{3}\ \overline{2}\ \overline{1}\ \overline{0}$

$x = 13 = 1101$, $y = 5 = 101$

$i \to 3$ to 0 , $u_4 = 0$,

For $i = 3$, $u_{i+1} = u_4 = 0$

Step 3) $z_i = x - u_{i+1} - 2^i y$

$z_3 = 13 - 0 - 2^3 \times 5$

$= 13 - 40$

$= -27$

4) Π_{DRELU} gives $\beta_3 = 0$ as $z_3 < 0$

5) Π_{MATMUL} gives $v_3 = \beta_3 \times 2^3 \times 5$

$= 0 \times 40$

$= 0$

6) $k_3 = 2^3 \cdot \beta_3 = 2^3 \cdot 0 = 0$

7) $u_3 = u_4 + v_3 = 0$

For $i = 2$, $u_3 = 0$

3) $z_2 = x - u_3 - 2^2 \times 5$

$= 13 - 0 - 20$

$= -7$

4) $\beta_2 = 0$ as $z_2 < 0$

5) $v_2 = \beta_2 \times 2^2 y = 0 \times 20 = 0$

6) $k_2 = 2^2 \cdot \beta_2 = 4 \times 0 = 0$

7) $u_2 = u_3 + v_2 = 0 + 0 = 0$

For $i = 1$, $u_2 = 0$

3) $z_1 = x - u_2 - 2^1 \times 5$

$= 13 - 0 - 10$

$= 3$

4) $\beta_1 = 1$ as $z_1 > 0$

5) $v_1 = \beta_1 \times 2^1 y = 1 \times 10 = 10$

6) $k_1 = 2^1 \cdot \beta_2 = 2 \times 1 = 2$

7) $u_1 = u_2 + v_1 = 0 + 10 = 10$

For $i = 0$, $u_1 = 10$

3) $z_0 = x - u_1 - 2^0 \times 5$

$= 13 - 10 - 5$

$= -2$

4) $\beta_0 = 0$ as $z_0 < 0$

5) $v_0 = \beta_0 \times 2^0 y = 0 \times 5 = 0$

6) $k_0 = 2^0 \cdot \beta_0 = 1 \times 0 = 0$

7) $u_0 = u_1 + v_0 = 10 + 0 = 10$

Then,

8) $q = \sum_{i=0}^{l-1} k_i$

$= k_0 + k_1 + k_2 + k_3$

$= 0 + 2 + 0 + 0$

$= 2$