5.2) Derivative of ReLU (Algorithm 6 — ReLU')

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

⊙ Input: P_0, P_1 hold $\langle a \rangle_0^L$ and $\langle a \rangle_1^L$, respectively

⊙ Output: P_0, P_1 get $\langle ReLU'(a) \rangle_0^L$ and $\langle ReLU'(a) \rangle_1^L$

⊙ Common Randomness: P_0, P_1 hold random shares of 0 over $\mathbb{Z}_L$, denoted by u_0 and u_1 respectively.

⊙ Steps:

1) For $j \in \{0,1\}$, parties P_j computes
$$\langle c \rangle_j^L = 2\langle a \rangle_j^L$$

2) P_0, P_1, P_2 run $\Pi_{SC}(\{P_0, P_1\}, P_2)$ with P_0, P_1 having inputs $\langle c \rangle_0^L$ and $\langle c \rangle_1^L$, and P_2 learns $\langle y \rangle_0^{L-1}$ and $\langle y \rangle_1^{L-1}$ resp.

3) P_0, P_1, P_2 run $\Pi_{MSB}(\{P_0, P_1, P_2\})$ with $P_j, j \in \{0,1\}$ having $\langle y \rangle_j^{L-1}$ and P_0, P_1 learn $\langle \alpha \rangle_0^L$ and $\langle \alpha \rangle_1^L$ resp.

4) For $j \in \{0,1\}$, P_j outputs
$$\langle y \rangle_j^L = j - \langle \alpha \rangle_j^L + u_j$$

⊙ Explanation:

For each share of $a \in \mathbb{Z}_L$

+ Step 1 computes share of $c = 2a$
+ Step 2 run Π_{SC} to convert the shares of c over $\mathbb{Z}_L$ to over $\mathbb{Z}_{L-1}$. The output shares are y over $\mathbb{Z}_{L-1}$.
+ Step 3 run Π_{MSB} to compute the MSB of y and the output bit is α whose shares are over $\mathbb{Z}_L$
+ Step 4 outputs Y-bit as DReLU(a)

if MSB(a) = 0, then it implies $a > 0$, hence $Y = 1$

if MSB(a) = 1, then it implies $a < 0$ as negative numbers have MSB as 1 is our representation inside group $\mathbb{Z}_L$ ($L = 2^{64}$), hence $Y = 0$

$(0, 2^{63} - 1) \cup (2^{63}, 2^{64} - 1) = \mathbb{Z}_{2^{64}}$

+ve numbers -ve numbers

Note: Π_{SC} and Π_{MSB} each make a single call to Π_{PC}

⊙ Confusion: Since Π_{MSB} already multiplies the shares of $\langle a \rangle_j^{L-1}$ with 2 to get $\langle y \rangle_j^{L-1} = 2\langle a \rangle_j^{L-1}$ in its protocol, why is the multiplication done in step 1 of Π_{DReLU} required?

5.3) ReLU (uses DReLU – Algorithm 7)

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

⊙ Input: P_0, P_1 hold $\langle a \rangle_0^L$ and $\langle a \rangle_1^L$ resp.

⊙ Output: P_0, P_1 get $\langle ReLU(a) \rangle_0^L$ and $\langle ReLU(a) \rangle_1^L$.

⊙ Common Randomness: P_0, P_1 hold random shares of 0 over $\mathbb{Z}_L$, denoted by u_0 and u_1 resp.

⊙ Steps:

1) P_0, P_1, P_2 run $\Pi_{ReLU}(\{P_0, P_1\}, P_2\}$ with $P_j, j \in \{0,1\}$ having input $\langle a \rangle_j^L$ and P_0, P_1 learn $\langle \alpha \rangle_0^L$ and $\langle \alpha \rangle_1^L$ resp.

2) P_0, P_1, P_2 call $\Pi_{MatMul}(\{P_0, P_1\}, P_2\}$ with $P_j, j \in \{0,1\}$ having input $(\langle a \rangle_j^L, \langle a \rangle_j^L)$ and P_0, P_1 learn $\langle c \rangle_0^L$ and $\langle c \rangle_1^L$ resp.

3) For $j \in \{0,1\}$, P_j outputs $\langle c \rangle_j^L + u_j$

④ Explanation:

+ Step 1 computes $ReLU'(a)$ using Π_{DReLU} and gets the output $\alpha = ReLU'(a)$

+ Step 2 computes $ReLU(a) = ReLU'(a) \cdot a$ using Π_{MatMul}. as gets the output as $c = ReLU(a)$

$$ReLU(a) = \begin{cases} a & \text{if } a > 0 \\ 0 & \text{if } a \leq 0 \end{cases} = (a > 0) \cdot a$$

$$= ReLU'(a) \cdot a$$

$$ReLU'(a) = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{if } a \leq 0 \end{cases}$$

+ Step 3 outputs c and when c is 0, it outputs shares of 0 over $\mathbb{Z}_L$ which is u.