CS771A: Assignment 1

ML _enthusiasts

Akshat Srivastava	**Hrithik Sharma**	**Anubhav Singh**	**Rishav Mondal**	**Ritesh Verma**
190091	190383	210163	210848	210861

Bachhav Aryan Kishor
210253

February 18, 2023

Q1. XORRO PUF broken by a single linear model: Derivation

Let δ_i^{00}, δ_i^{01}, δ_i^{10}, and δ_i^{11} be the times that the ith XOR gate takes before giving its output when the input to that gate is, respectively, 00, 01, 10, and 11.
Let c_i be the output of the $(i-1)$th XOR gate, and let t_i be the time at which the ith XOR gate sends its output signal.

$$\therefore t_i = (1-c_i)(1-a_i)\delta_i^{00} + a_i\delta_i^{01} + c_i(1-c_i)\delta_i^{10} + a_i\delta_i^{11} + t_{i-1} \tag{1}$$

$c_{i+1} = (1-c_i)a_i + c_i(1-a_i)$.
$c_{i+1} = a_i - ciai + c_i - c_ia_i$
$c_ia_i = \frac{(-c_{i+1}+a_i+c_i)}{2}$ (2)

Putting the value of c_ia_i in equation (1), we get:
$t_i = (1-a_i)\delta_i^{00} + a_i\delta_i^{01} + c_i(1-a_i)(\delta_i^{10} - \delta_i^{00}) + a_i(\delta_i^{11} - \delta_i^{01}) + t_{i-1}$.

$t_i = (1-a_i)\delta_i^{00} + a_i\delta_i^{01} + (c_i - \frac{(a_i+c_i-c_{i+1})}{2})(\delta_i^{10} - \delta_i^{00}) + \frac{(a_i+c_i-c_{i+1})}{2}(\delta_i^{11} - \delta_i^{01}) + t_{i-1}$.
$= ((1-a_i)\delta_i^{00} + a_i\delta_i^{01} + \frac{(a_i+c_i-c_{i+1})}{2}(\delta_i^{10} - \delta_i^{00}) + \frac{(a_i+c_i-c_{i+1})}{2}(\delta_i^{11} - \delta i^{01}) +$
$((1-a_{i-1})\delta_{i-1}^{00} + a_{i-1}\delta_{i-1}^{01} + \frac{(a_{i-1}+c_{i-1}-c_i)}{2}(\delta_{i-1}^{10} - \delta_{i-1}^{00}) + \frac{(a_{i-1}+c_{i-1}-c_i)}{2}(\delta_{i-1}^{11} - \delta_{i-1}^{01}) + ...$

Let the total time taken by XORRO '0' be T_0, where it oscillates from 0 to 1 to 0. We start the XORRO with no input given, so we assume that the first output or the first input to the first XOR gate is 0 and a_0.

Now, when we express c_i and c_{i+1} in vector form, the only difference between the two is that the first element in the $[c_i]$ vector is not present in the $[c_{i+1}]$ vector, and the last element of $[c_{i+1}]$ vector is not present in $[c_i]$ vector. To solve this problem, we add the last element of $[c_{i+1}]$ in $[c_i]$ at the end, and the first element of $[c_i]$ is added to the beginning of $[c_{i+1}]$. We call this new vector $[c^*]$, which has dimension $(2r+1) \times (1)$.

Example-
Let the $[c_i]$ vector be $[0\ 1\ 1\ 0\ 1\ \ldots\ 0\ 1\ 1\ 0]^{\mathrm{T}}$,
then the $[c_{i+1}]$ vector be $[1\ 1\ 0\ 1\ \ldots\ 1\ 1\ 0\ 0]^{\mathrm{T}}$.
Now $c^* = [0\ 1\ 1\ 0\ 1\ \ldots\ 0\ 1\ 1\ 0\ 0]^{\mathrm{T}}$.
Now, to cancel the effect of adding an extra element in $[c_i]$ and $[c_{i+1}]$, we have to make changes in δ_i^{00}, δ_i^{01}, δ_i^{10}, and δ_i^{11}, the terms that are multiplied by c_i.

Preprint. Under review.

We add 0 at the last of each element's vector, and the terms that are multiplied by c_{i+1}, we add 0 at the beginning of each element's vector. Thus, the equation becomes:

$T_0 = [1-A][\delta^{00}]+[A]^T[\delta^{11}-\delta^{01}-\delta^{10}+\delta^{00}]+[c^*][\delta_1^{11}-\delta_1^{01}+\delta_1^{10}-\delta_1^{00}-\delta_2^{11}+\delta_2^{01}+\delta_2^{10}-\delta_2^{00}]$.
$Let\ us\ denote$ [1-A][δ^{00}] + $[A][\delta^{01}]$ + $[\frac{A}{2}][\delta^{11}-\delta^{01}-\delta 10+\delta 00]$ by t_1, and

$[\delta_1^{11}-\delta_1^{01}+\delta_1^{10}-\delta_1^{00}-\delta_2^{11}+\delta_2^{01}+\delta_2^{10}-\delta_2^{00}]^T$ by $\frac{[c^*]}{2}w_1\phi(c)$.

Similarly for XORRO '1'.

$T_1 = b_2 + w_2^T\phi(c)$

$\phi(c)$ remains unchanged because the input is the same in case of the XORRO.

$\therefore \triangle T = (b_1 - b_2) + w_1^T - w_2^T\phi(c)$.
=b+w$^T\phi(c)$.
if $\triangle T > 0$, then the frequency of XORRO '1' is greater than XORRO '0' and vice versa. Thus, we get the output $\frac{1+sign(w^T\phi(c)+b)}{2}$.

Q2. Extending the above model to crack an Advanced XORRO PUF

When we use 2^s XORROs, the number of ways to select 2 XORROs are .

$$\binom{2^s}{2} = \frac{2^s(2^s-1)}{2} = 2^{s-1}(2^s-1)$$

for each pair we require 1 model and thus for $\binom{2^s}{2}$ pair we require $\binom{2^s}{2}$ model basically ,the above method is extended to $\binom{2^s}{2}$ pair.

Q4. Outcomes of experiments with LinearSVC and LogisticRegression

a. LinearSVC Model with different Loss functions:

Loss Hyperparameter	**Accuracy**
Hinge Loss	93.95%
Squared Hinge Loss	94.73%

d. Changing the penalty (regularization) hyperparameter in LinearSVC and LogisticRegression:

LinearSVC

Penalty	**Accuracy**
l1	94.6%
l2	94.74%

LogisticRegression (liblinear solver)

Penalty	**Accuracy**
l1	93.91%
l2	94.03%

References

[1] CS771A Lecture Notes and Discussion Hour Material, Spring 2023, IIT Kanpur