CS771: Intro to ML
Assignment 1

Sandeep Parmar
Roll no- 210922
BS-SDS

Daphal Sanket Anil
Roll no- 210300
BS-MTH

Sandeep Singh Gurjar
Roll no- 210923
BT-CE

Sarthak Agarwal
Roll no- 210934
BT-EE

Satish Kumar
Roll no- 210939
BT-EE

Sutariya Smitkumar Sureshbhai
Roll no - 211087
BT-EE

1 Question 1

To solve the CAR-PUF first we have to show that how single 32-bit arbitrary PUF can be broken by single linear model. Below fig describe the general setup for arbitrary PUF.

Notations -

1. $C_i's$ is challenges, it can be either 0 or 1. More generally our input is vector C of length 32, $C := \{0,1\}^{32}$.

2. r denotes response OR output, it is also 0 or 1.

3. t_i^u is the (unknown) time at which the upper signal leaves the i^{th} mux.

4. t_i^l is the time at which the lower signal leaves the the i^{th} mux.

5. p_i, q_i, r_i, and s_i is the time travel to inside the i^{th} mux depend on the which path signal travel.

Based on above setting, our response r take values based on following decision rule.

$$r = \begin{cases} 0 & \text{if} \quad t_{31}^u < t_{31}^l \\ 1 & o.w. \end{cases}$$

Time relation -

$$t_1^u = (1 - C_1) \cdot (t_0^u + p_1) + C_1 \cdot (t_0^l + s_1)$$

similarly,

$$t_1^l = (1 - C_1) \cdot (t_0^l + q_1) + C_1 \cdot (t_0^u + r_1)$$

Define $\Delta_i = t_i^u - t_i^l$
Then our decision rule become $\Delta_{31} < 0$ or not.

$$\begin{aligned} \Delta_1 &= t_1^u - t_1^l \\ &= (1 - C_1) \cdot (t_0^u + p_1) + C_1 \cdot (t_0^l + s_1) - \{(1 - C_1) \cdot (t_0^l + q_1) + C_1 \cdot (t_0^u + r_1)\} \\ &= (1 - C_1) \cdot (t_0^u + p_1 - t_0^l - q_1) + C_1 \cdot (t_0^l + s_1 - t_0^u - r_1) \\ &= (1 - c_1) \cdot (\Delta_0 + p_1 - q_1) + c_1 \cdot (-\Delta_0 + s_1 - r_1) \\ &= (1 - 2c_1) \cdot \Delta_0 + (q_1 - p_1 + s_1 - r_1) \cdot c_1 + (p_1 - q_1) \end{aligned}$$

Let define $d_i = 1 - 2 \cdot C_i \implies d_i \in \{-1, 1\}$ Then

$$\Delta_1 = \Delta_0 \cdot d_1 + \alpha_1 \cdot d_1 + \beta_1$$

Where $\alpha_1 = \frac{(p_1 - q_1 + r_1 - s_1)}{2}$ and $\beta_1 = \frac{(p_1 - q_1 - r_1 + s_1)}{2}$

For the Δ_i similar relation hold

$$\Delta_i = \Delta_{i-1} \cdot d_i + \alpha_i \cdot d_i + \beta_i$$

We take $\Delta_{-1} = 0$ because initial decay absorb into p_0, q_0, r_0, s_0

Recursively we find all $\Delta_i's$ in terms of $d_i's$

$$\begin{aligned}
\Delta_0 &= \alpha_0 \cdot d_0 + \beta_0 \quad (\Delta_{-1} = 0) \\
\Delta_1 &= \Delta_0 \cdot d_1 + \alpha_1 \cdot d_1 + \beta_1 \\
&= (\alpha_0 \cdot d_0 + \beta_0) \cdot d_1 + \alpha_1 \cdot d_1 + \beta_1 \\
&= \alpha_0 \cdot d_0 \cdot d_1 + (\alpha_1 + \beta_0) \cdot d_1 + \beta_1 \\
\Delta_2 &= d_2 \cdot \Delta_1 + \alpha_2 \cdot d_2 + \beta_1 \\
&= d_2 \cdot \{\alpha_0 \cdot d_0 \cdot d_1 + (\alpha_1 + \beta_0) \cdot d_1 + \beta_1\} + \alpha_2 \cdot d_2 + \beta_1 \\
&= \alpha_0 \cdot d_0 \cdot d_1 \cdot d_2 + (\alpha_1 + \beta_0) \cdot d_1 \cdot d_2 + (\beta_1 + \alpha_2) \cdot d_2 + \beta_1
\end{aligned}$$

We observe pattern and At the i^{th} step

$$\Delta_i = \alpha_0 \cdot (d_0 d_1 \ldots d_i) + (\alpha_1 + \beta_0) \cdot (d_1 d_2 \ldots d_i) + (\alpha_2 + \beta_1) \cdot (d_2 d_3 \ldots d_i) \ldots\ldots + \beta_i$$

At the 31^{st} step

$$\Delta_{31} = W_0 \cdot x_0 + W_1 \cdot x_1 + W_2 \cdot x_3 + \ldots W_{31} \cdot x_{31} + \beta_{31} = \mathbf{W}^T\mathbf{X} + \mathbf{b} \qquad \textbf{(1)}$$

Where

$$x_i = d_i \cdot d_{i+1} \ldots d_{31} \tag{1}$$

$$W_0 = \alpha_0 \tag{2}$$

$$W_i = \alpha_i + \beta_{i-1} \quad ; \forall \ i > 0 \tag{3}$$

At the end we are only interested in Δ_{31}, our response y_i depends only $sign(\Delta_{31})$. And we observe that equation-(1) is the equation of linear regression, So we can predict future response y_i by the use of linear classifier model.

Final Decision rule

$$r = \begin{cases} 0 & \text{if} \quad \Delta_{31} < 0 \\ 1 & \text{if} \quad \Delta_{31} > 0. \end{cases}$$

OR

$$r = \frac{sign(W^T X + b) + 1}{2}$$

Note that X is modified feature vector and its each element X_i is function of challenges $C_1, C_2, , , C_{31}$ and r is corresponding response.

1.1 Conclusion

Solving the arbitrary PUFs is just a binary classification problem. We can use any linear classifier like SVM, logistics regression etc. Given the set of training data CRP's (Chanlleges and responses) we find W and b and for future observation we can predict y.

1.2 Solving CAR-PUF

Given :-

- 2 arbiter PUFs – a *working* PUF and a *reference* PUF

- Threshold $\tau > 0$

- Δ_w, Δ_r be the difference in timings experienced for the two PUFs on the same challenge.

Based on given condition and challenge response r.

$$r = \begin{cases} 0 & \text{if} \quad |\Delta_w - \Delta_r| \leq \tau \\ 1 & \text{if} \quad |\Delta_w - \Delta_r| > \tau. \end{cases}$$

objective

- To how a CAR-PUF can be broken by a single linear model.

- derivations for a map $\phi : \{0,1\}^{32} \to \mathbb{R}^D$ mapping 32-bit $0/1$-valued challenge vectors to D-dimensional feature vectors (for some $D > 0$) so that for any CAR-PUF, there exists a D-dimensional linear model $W \in R$,D and a bias term $b \in \mathbb{R}$ such that for all CRPs (c, r) with $c \in \{0,1\}^{32}$, $r \in \{0,1\}$ we have

$$\frac{1 + sign(W^T \phi(c) + b)}{2} = r$$

Soln

Let (u, p),(v, q) be the two linear models that can exactly predict the outputs of the two arbiter PUFs. (Derived above)

Then from derivation

$$\Delta_w = u^T X + p \tag{4}$$

$$\Delta_r = v^T X + q \tag{5}$$

$$|\Delta_w - \Delta_r| = |(u - v)^T X + p - q|$$

For simplicity let $W_o = u - v$ and $b_o = p - q$. then

$$|\Delta_w - \Delta_r| = |W_o^T X + b_o|$$

Modified classification rule

$$r = \begin{cases} 0 & \text{if} \quad |W_o^T X + b_o| \leq \tau \\ 1 & \text{if} \quad |W_o^T X + b_o| > \tau. \end{cases}$$

Since $|W_o^T X + b_o|$ and τ both are positive quantity so squaring both side in above inequalities does not affect decision rule.

$$|W_o^T X + b_o|^2 > \tau^2 \tag{6}$$

$$(W_o^T X)^2 + 2 \cdot W_o^T X \cdot b_o + b_o^2 > \tau^2 \tag{7}$$

$$(W_o^T X)^2 + 2 \cdot W_o^T X \cdot b_o + b_o^2 - \tau^2 > 0 \qquad \textbf{(2)} \tag{8}$$

This implies $y = 1$ if eqn-(2) holds. So this can be helpful to make feature vector $\phi(c)$

$$(W_o^T X)^2 + 2 \cdot W_o^T X \cdot b_o + b_o^2 - \tau^2 = W^T \phi(c) + b \qquad \textbf{(3)}$$

Calculating each term of LHS explicitly

$$(W_o X)^2 = \left(\sum_{i=0}^{31} w_{oi} \cdot x_i \right)^2 \tag{9}$$

$$= \sum_{i=0}^{31} w_{io}^2 \cdot x_i^2 + 2 \sum_{i=0}^{31} \sum_{\substack{j=0 \\ j \neq i}}^{31} x_i \cdot x_j \cdot w_{io} \cdot w_{jo} \tag{10}$$

$$= \sum_{i=0}^{31} w_{io}^2 + 2 \sum_{i=0}^{31} \sum_{\substack{j=0 \\ j \neq i}}^{31} x_i \cdot x_j \cdot w_{io} \cdot w_{jo} \qquad \textbf{(4)} \tag{11}$$

because $x_i \in \{0,1\} \implies x_i^2 = 1 \quad$ Always.

second term of LHS

$$2 \cdot W_o^T X \cdot b_o = 2b \sum_{i=0}^{31} w_{io} x_i = \sum_{i=0}^{31} (2b \cdot w_{io}) x_i \qquad \textbf{(5)}$$

Substitute (4) and (5) into (3) and comparing RHS and LHS

$$\phi(c) = \left(\underbrace{4x_0x_1, 4x_0x_2, 4x_0x_3, \ldots, 4x_{31}x_{30}}_{\text{Total } 496 \text{ terms}}, \underbrace{2x_1, 2x_2, 2x_3, \ldots, 2x_{31}}_{\text{Total } 32 \text{ terms}} \right)^T$$

In eqn. (4), $x_1x_2 = x_2x_1$ so Total unique terms = $\frac{31\times32}{2}$ = 496.
And Corresponding $\mathbf{W}$ vector

$$W = (w_{o0}w_{o1}, w_{o0}w_{o2}, w_{o0}w_{o3}, \ldots, w_{o31}w_{o30}, w_{o1}b_o, w_{o2}b_o, w_{o3}b_o, \ldots, w_{o31}b_o)^T$$

Bias term $\mathbf{b}$

$$b = b_o^2 - \tau^2 + \sum_{i=0}^{31} w_i^2$$

The length of vector $\phi(c)$ is given by $\underline{D = 496 + 32 = 528}$.

2 Question 3

2.1 a

Table 1: Performance metrics with different loss functions in Linear SVC

Loss	t_train	t_map	Acc
Square Hinge loss	2.29	0.072	0.9919
Hinge Loss	18.204	0.071	0.9896

Inference - We have observed that time required in Hinge-Loss is approximately 9-times more and also accuracy is decreased in Hinge-Loss.

2.2 b

Table 2: Linear SVC with default parameters and different C

C	t_train	t_map	Acc
0.1	0.7927	0.0646	0.9871
1	2.0876	0.0362	0.9919
10	2.1012	0.0357	0.9929
100	17.7667	0.0485	0.9921
1000	37.6770	0.0526	0.9921

Table 3: Logistic Regression with default parameters and different C

C	t_train	t_map	Acc
0.1	1.6017	0.0281	0.9
1	0.8421	0.0638	0.9907
10	0.9383	0.0608	0.9922
100	1.2620	0.0612	0.9930
110	1.3353	0.0636	0.9930
150	1.3194	0.0619	0.9930
1000	1.7927	0.0622	0.9923

Inference -
As compare to LinearSVC, LogisticRegression takes less time to train the data. And at $C = 100$ in Logistic we get highest Accuracy.