
CS771 Introduction to Machine Learning

Assignment 1

Gunj Mehul Hundiwala
22111024
gunjmehul22@iitk.ac.in

Kartick Verma
22111029
kartickv22@iitk.ac.in

Kush Shah
22111033
kushshah22@iitk.ac.in

Raj Kumar
22111050
rajkumar22@iitk.ac.in

Saqeeb
22111053
saqeeb22@iitk.ac.in

Question 1

By giving a mathematical derivation, show there exists a way to map the binary digits 0,1 to sign -1, +1 as say $m : \{0, 1\} \rightarrow \{-1, +1\}$ and another way $f : \{-1, +1\} \rightarrow \{0, 1\}$ to map signs to bits (not that m and f need not be inverses of each other) so that for any set of binary digits (b_1, b_2, \dots, b_n) for any $n \in \mathbb{N}$ we have

$$XOR(b_1, b_2, \dots, b_n) = \left(\prod_{i=1}^n m(b_i) \right). \quad (1)$$

Thus, the XOR function is not that scary - it is essentially a product.

Solution 1

We have to find the function

$$m : \{0, 1\} \rightarrow \{-1, +1\} \text{ and} \quad (2)$$

$$f : \{-1, +1\} \rightarrow \{0, 1\} \text{ so that} \quad (3)$$

for any set of binary digit b_1, b_2, \dots, b_n for $n \in \mathbb{N}$ we have

$$XOR(b_1, b_2, \dots, b_n) = f \left(\prod_{i=1}^n m(b_i) \right) \quad (4)$$

let $m(x) = 1 - 2x$ $x \in \{0, 1\}$ and $f(y) = \frac{1-y}{2}$ $y \in \{-1, +1\}$

Now we will try to prove

$$XOR(b_1, b_2, \dots, b_n) = f \left(\prod_{i=1}^n m(b_i) \right) \quad (5)$$

Using Principle of mathematical induction.

For $n=2$, we have 2^2

Case 1 :

$$\begin{aligned} XOR(0, 0) &= 0 \Rightarrow LHS \\ RHS &= f(m(0).m(0)) = f(1.1) = 0 \\ LHS &= RHS \end{aligned}$$

Case 2 :

$$\begin{aligned} XOR(0, 1) &= 1 = LHS \\ RHS &= f(m(0).m(1)) = f(1.(-1)) = f(-1) = 1 \\ LHS &= RHS \end{aligned}$$

Case 3 :

$$\begin{aligned} XOR(1, 0) &= 1 = LHS \\ RHS &= f(m(1).m(0)) = f((-1) * (+1)) = f(-1) = 1 \\ LHS &= RHS \end{aligned}$$

Case 4 :

$$\begin{aligned} XOR(1, 1) &= 0 = LHS \\ RHS &= f(m(1).m(1)) = f((-1) * (-1)) = f(1) = 0 \\ LHS &= RHS \end{aligned}$$

So $p(2)$ is true

Lets assume that $p(k)$ is true, so

$$XOR(b_1, b_2, \dots, b_k) = f\left(\prod_{i=1}^k m(b_i)\right) \quad (6)$$

Lets check for whether $p(k+1)$ is true or not,

$$LHS \Rightarrow XOR(b_1, b_2, \dots, b_{(k+1)}) = XOR(b_1, \dots, b_k) \oplus b_{(k+1)} \quad (7)$$

$$Now, XOR(b_1, \dots, b_k) = f\left(\prod_{i=1}^k m(b_i)\right) \quad (8)$$

$\therefore XOR(b_1, \dots, b_k)$ can be 0 or 1 and so the quantity $f\left(\prod_{i=1}^k m(b_i)\right)$

Now, we will have 4 cases,

Case 1 :

$$\begin{aligned} XOR(b_1, \dots, b_k) &= 0, b_{(k+1)} = 0 \\ LHS &= XOR(b_1, \dots, b_{(k+1)}) = 0 \oplus 0 = 0 \\ RHS &= f\left(\prod_{i=1}^{k+1} m(b_i)\right) = f\left(\prod_{i=1}^k m(b_i).m(b_{k+1})\right) \\ XOR(b_1, \dots, b_k) &= 0 \text{ means } f\left(\prod_{i=1}^k m(b_i)\right) = 0 \end{aligned}$$

We know that, $f()$ function maps $-1 \rightarrow 1, 1 \rightarrow 0$

If $f()$ has a output 0, then input must have been 1

$$\therefore \prod_{i=1}^k m(b_i) = 1$$

Substituting in the RHS, we get

$$RHS = f(1.m(b_{k+1})) = f(m(b_{k+1}))$$

Now b_{k+1} was a zero

$$\begin{aligned} \Rightarrow RHS &= f(m(0)) = f(1) = 0 \\ &\Rightarrow LHS = RHS \end{aligned}$$

So, in this case, we have $p(k+1)$ is true.

Case 2 :

$$\begin{aligned} XOR(b_1, \dots, b_k) &= 0, b_{k+1} = 1 \\ LHS &= XOR(b_1, \dots, b_{k+1}) = 0 \oplus 1 = 1 \\ RHS &= f\left(\prod_{i=1}^{k+1} m(b_i)\right) = f\left(\prod_{i=1}^k m(b_i).m(b_{k+1})\right) \end{aligned}$$

$XOR(b_1, \dots, b_k) = 0$ implies that $\prod_{i=1}^k m(b_i) = 1$ from the previous case

Substituting in the RHS, we get

$$\begin{aligned} RHS &= f(1.m(b_{k+1})) = f(m(b_{k+1})) \\ &= f(m(1)) = f(-1) = 1 \\ &\Rightarrow LHS = RHS \end{aligned}$$

Case 3 :

$$\begin{aligned} XOR(b_1, \dots, b_k) &= 1, b_{k+1} = 0 \\ LHS &= XOR(b_1, \dots, b_{k+1}) = 1 \oplus 0 = 1 \\ RHS &= f\left(\prod_{i=1}^{k+1} m(b_i)\right) = f\left(\prod_{i=1}^k m(b_i).m(b_{k+1})\right) \end{aligned}$$

$XOR(b_1, \dots, b_{k+1}) = 1$ implies that $\prod_{i=1}^k m(b_i) = -1$

Substituting in the RHS, we get

$$\begin{aligned} RHS &= f((-1).m(b_{k+1})) = f(-m(0)) \\ &\Rightarrow f((-1) * (+1)) = f(-1) = 1 \\ &\Rightarrow LHS = RHS \end{aligned}$$

Case 4 :

$$\begin{aligned} XOR(b_1, \dots, b_k) &= 1, b_{k+1} = 1 \\ LHS &= XOR(b_1, \dots, b_{k+1}) = 1 \oplus 1 = 0 \\ RHS &= f\left(\prod_{i=1}^{k+1} m(b_i)\right) = f\left(\prod_{i=1}^k m(b_i).m(b_{k+1})\right) \end{aligned}$$

Now, $XOR(b_1, \dots, b_{k+1}) = 1$ implies that $\prod_{i=1}^k m(b_i) = -1$

Substituting in the RHS, we get

$$\begin{aligned} RHS &= f(-1.m(b_{k+1})) = f(-m(1)) \\ &= f((-1) * (-1)) = f(1) = 0 \\ \therefore LHS &= RHS \end{aligned}$$

So, $p(k+1)$ is true

So, by the principle of mathematical induction, we have proved that,

$$XOR(b_1, \dots, b_n) = f\left(\prod_{i=1}^n m(b_i)\right) \quad (9)$$

Question 2

Let $(u,a),(v,b),(w,c)$ be the three linear models that can exactly predict the outputs of the three individual PUFs sitting inside the XOR-PUF. For sake of simplicity, let us hide the bias term inside the model vector by adding a unit dimension to the original feature vector so that we have $\tilde{u} = [u,a]$, $\tilde{v}=[v,b]$, $\tilde{w} = [w,c]$, $\tilde{x}=[x,1] \in \mathbb{N}^9$. The above calculation shows that the response of the XOR-PUF can be easily obtained (by applying f) if we are able to get hold of the following quantity:

$$\text{sign}(\tilde{u}^T \tilde{x}).\text{sign}(\tilde{v}^T \tilde{x}).\text{sign}(\tilde{w}^T \tilde{x}) \quad (10)$$

To exploit the above result, first give a mathematical proof that for any real numbers (that could be positive, negative, zero) r_1, r_2, \dots, r_n for any $n \in \mathbb{N}$, we always have

$$\prod_{i=1}^n \text{sign}(r_i) = \text{sign}\left(\prod_{i=1}^n r_i\right) \quad (11)$$

Assume that $\text{sign}(0) = 0$. Make sure you address all edge cases in your calculations e.g. if one or more of the numbers is 0.

Solution 2

We have r_1, r_2, \dots, r_n as a set of real numbers where $n \in \mathbb{N}$. We assume a property $P(t)$ such that

$$P(t) : \prod_{i=1}^t \text{sign}(r_i) = \text{sign}\left(\prod_{i=1}^t r_i\right) \quad (12)$$

Now lets check for $P(1)$

$$P(1) : LHS \Rightarrow \prod_{i=1}^1 \text{sign}(r_i) = \text{sign}(r_1) \quad (13)$$

$$RHS \Rightarrow \text{sign}\left(\prod_{i=1}^1 r_i\right) = \text{sign}(r_1) \quad (14)$$

$$LHS = RHS = \text{sign}(r_1) \quad (15)$$

$\therefore p(1)$ is true.

Now lets check for some random natural number $k < n$, $p(k)$ property. Assume that $p(k)$ is true, we have

$$p(k) : \prod_{i=1}^k \text{sign}(r_i) = \text{sign}\left(\prod_{i=1}^k r_i\right) \quad (16)$$

Now lets check whether $p(k+1)$ is true or not

$$p(k+1) : \prod_{i=1}^{k+1} \text{sign}(r_i) = \text{sign}\left(\prod_{i=1}^{k+1} r_i\right) \quad (17)$$

$$LHS = \prod_{i=1}^{k+1} \text{sign}(r_i) = \text{sign}(r_{k+1}) \cdot \left(\prod_{i=1}^k \text{sign}(r_i)\right) \quad (18)$$

$$RHS = \text{sign}\left(\prod_{i=1}^{k+1} r_i\right) = \text{sign}\left(r_{k+1} \cdot \prod_{i=1}^k r_i\right) \quad (19)$$

let us assume two cases,

Case 1 : r_{k+1} is positive then,

$$\text{sign} \left(\prod_{i=1}^{k+1} r_i \right) = +ve. \text{sign} \left(\prod_{i=1}^k r_i \right) \quad (20)$$

$$= \text{sign}(r_{k+1}). \text{sign} \left(\prod_{i=1}^k r_i \right) \quad (21)$$

Case 2 : r_{k+1} is negative then,

$$\text{sign} \left(\prod_{i=1}^{k+1} r_i \right) = -ve. \text{sign} \left(\prod_{i=1}^k r_i \right) \quad (22)$$

$$= \text{sign}(r_{k+1}). \text{sign} \left(\prod_{i=1}^k r_i \right) \quad (23)$$

$$\text{So, } RHS = \text{sign} \left(\prod_{i=1}^{k+1} r_i \right) = \text{sign}(r_{k+1}). \text{sign} \left(\prod_{i=1}^k r_i \right) \quad (24)$$

From our assumption of $p(k)$, substituting in equation (24),

$$RHS = \text{sign} \left(\prod_{i=1}^{k+1} r_i \right) = \text{sign}(r_{k+1}). \left(\prod_{i=1}^k \text{sign}(r_i) \right) \quad (25)$$

So, $p(k+1)$ is also coming true.

So, when $p(k)$ is assumed to be true then $p(k+1)$ is also coming true

So, by the principle of mathematical induction, $p(n)$ is true for all $n \in \mathbb{N}$

$$\text{So, } p(n) : \prod_{i=1}^n \text{sign}(r_i) = \text{sign} \left(\prod_{i=1}^n r_i \right) \quad (26)$$

is true for all $n \in \mathbb{N}$

If any of $r_i \in \mathbb{N}$ suppose $r_t = 0$ becomes zero, then as given in the question $\text{sign}(0) = 0$

$$LHS = \prod_{i=1}^n \text{sign}(r_i) = \text{sign}(r_1). \text{sign}(r_2) \dots \text{sign}(r_t) \dots \text{sign}(r_n) \quad (27)$$

$$\text{Since, } \text{sign}(r_t) = 0, LHS = 0 \quad (28)$$

Similarly,

$$RHS = \text{sign} \left(\prod_{i=1}^n r_i \right) = \text{sign}(r_1.r_2 \dots r_t \dots r_n) \quad (29)$$

Now $r_t = 0$. So,

$$RHS = \text{sign}(r_1.r_2...0...r_n) = \text{sign}(0) = 0 \quad (30)$$

$$LHS = RHS \quad (31)$$

So, when one or more numbers is zero then also $p(n)$ holds.

Question 3

The above calculation tells us that all we need to get hold of is the following quantity

$$(\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}).(\tilde{w}^T \tilde{x}) \quad (32)$$

Now show that the above can be expressed as a linear model but possibly in a different dimensional space. Show that there exists a dimensionality D such that D depends only on the number of PUFs (in this case 3) and the dimensionality of \tilde{x} (in this case $8 + 1 = 9$) and there exists a way to map 9 dimensional vectors to D dimensional vectors as $\phi : \mathbb{R}^9 \rightarrow \mathbb{R}^D$ such that for any triple $(\tilde{u}, \tilde{v}, \tilde{w})$, there always exists a vector $W \in \mathbb{R}^D$ such that for every $\tilde{x} \in \mathbb{R}^9$, we have $(\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}).(\tilde{w}^T \tilde{x}) = W^T \phi(\tilde{x})$.

Solution 3

$(\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}).(\tilde{w}^T \tilde{x})$, this quantity is important We had 8 features in each of 'x' feature vectors. Now after hiding the bias term in x we get \tilde{x} of dimension 9.

So, \tilde{x} is a 9 dimension vector.

Now, our target is to model the prediction $(\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}).(\tilde{w}^T \tilde{x})$ quantity into a single linear model with a different dimensionality D .

Now,

$$(\tilde{u}^T \tilde{x}) = \sum_{j=1}^9 \tilde{u}_j \tilde{x}_j \quad (33)$$

$$= u_1 x_1 + u_2 x_2 + \dots + u_n x_n \quad (34)$$

$$(\tilde{v}^T \tilde{x}) = \sum_{j=1}^9 \tilde{v}_j \tilde{x}_j \quad (35)$$

$$= v_1 x_1 + v_2 x_2 + \dots + v_n x_n \quad (36)$$

Now,

$$(\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}) = (u_1 x_1 + u_2 x_2 + \dots + u_9 x_9).(v_1 x_1 + v_2 x_2 + \dots + v_9 x_9) \quad (37)$$

$$(\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}) = \sum_{j=1}^9 \sum_{k=1}^9 \tilde{u}_j \tilde{v}_k \tilde{x}_j \tilde{x}_k \quad (38)$$

Similarly when we try to solve for 3 PUFs, if we expand the term

$$(\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}).(\tilde{w}^T \tilde{x}) = \left(\sum_{j=1}^9 \tilde{u}_j \tilde{x}_j \right) \cdot \left(\sum_{j=1}^9 \tilde{v}_j \tilde{x}_j \right) \cdot \left(\sum_{j=1}^9 \tilde{w}_j \tilde{x}_j \right) \quad (39)$$

$$= \sum_{j=1}^9 \sum_{k=1}^9 \sum_{l=1}^9 \tilde{u}_j \tilde{v}_k \tilde{w}_l \tilde{x}_j \tilde{x}_k \tilde{x}_l \quad (40)$$

$$\text{So, } (\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}).(\tilde{w}^T \tilde{x}) = u_1 v_1 w_1 x_1 x_1 x_1 + u_1 v_1 w_2 x_1 x_1 x_2 + u_1 v_1 w_3 x_1 x_1 x_3 + \dots + u_9 v_9 w_9 x_9 x_9 x_9 \quad (41)$$

We could see that our new linear equation has $9 \times 9 \times 9 = 729$ features where $i, j, k \in [1, 9]$. Now, we can see that $9^3 = 729$ dimensional function that maps $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_9)$ to

$$\phi(\tilde{x}) = \begin{bmatrix} \tilde{x}_1\tilde{x}_1\tilde{x}_1 & . & . & . & \tilde{x}_1\tilde{x}_1\tilde{x}_9 & \tilde{x}_1\tilde{x}_2\tilde{x}_1 & . & . & . & \tilde{x}_1\tilde{x}_2\tilde{x}_9 \\ \tilde{x}_1\tilde{x}_3\tilde{x}_1 & . & . & . & \tilde{x}_1\tilde{x}_3\tilde{x}_9 & \tilde{x}_1\tilde{x}_4\tilde{x}_1 & . & . & . & \tilde{x}_1\tilde{x}_4\tilde{x}_9 \\ . & . & . & . & . & . & . & . & . & . \\ \tilde{x}_2\tilde{x}_1\tilde{x}_1 & . & . & . & \tilde{x}_2\tilde{x}_1\tilde{x}_9 & \tilde{x}_2\tilde{x}_2\tilde{x}_1 & . & . & . & \tilde{x}_2\tilde{x}_2\tilde{x}_9 \\ . & . & . & . & . & . & . & . & . & . \\ \tilde{x}_9\tilde{x}_8\tilde{x}_1 & . & . & . & \tilde{x}_9\tilde{x}_8\tilde{x}_9 & \tilde{x}_9\tilde{x}_9\tilde{x}_1 & . & . & . & \tilde{x}_9\tilde{x}_9\tilde{x}_9 \end{bmatrix} \quad (42)$$

Total there will be 729 elements in $\phi(\tilde{x})$ and this we are done Since we can now get

$$(\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}).(\tilde{w}^T \tilde{x}) = W^T \phi(\tilde{x}) \quad (43)$$

$$where \ W = \begin{pmatrix} \tilde{u}_1\tilde{v}_1\tilde{w}_1 \\ \tilde{u}_1\tilde{v}_1\tilde{w}_2 \\ . \\ . \\ . \\ \tilde{u}_9\tilde{v}_9\tilde{w}_9 \end{pmatrix}$$

Dimentionality (D) = $(8 + 1)^3 = 729$, which depends on number of PUFs i.e. 3 and dimentionality of $\tilde{x} = 8 + 1 = 9$

and we have proved that there exists a way to map 9 dimensional vectors to D dimensional vectors as $\phi : \mathbb{R}^9 \rightarrow \mathbb{R}^D$ where $D=729$.

such that for any triple $(\tilde{u}, \tilde{v}, \tilde{w})$, there always exists a vector $w \in \mathbb{R}^D$ i.e. $w \in \mathbb{R}^{729}$ such that for every $\tilde{x} \in \mathbb{R}^9$. we have $(\tilde{u}^T \tilde{x}).(\tilde{v}^T \tilde{x}).(\tilde{w}^T \tilde{x}) = W^T \phi(\tilde{x})$.

Question 5

For the method you implemented, describe in your PDF report that were the hyperparameters e.g. step length, policy on choosing the next coordinate if doing SDCA, mini-batch size if doing MBSGD etc and how did you arrive at the best values for the hyperparameters, e.g. you might say *We used step length at time t to be η/\sqrt{t} where we checked for $\eta = 0.1, 0.2, 0.5, 1, 2, 5$ using held out validation and found $\eta = 2$ to work the best*". For another example, you might say, *"We tried random nad cyclic coordinate selection choices and found cyclic to work best using 5-fold cross validation"*. Thus, you must tell us among which hyperparameter choices did you search for the best and how.

Solution 5

To train our model we use learning rate ($\eta = 0.12$) and lambda parameter ($\lambda_{\text{para}} = 0.0001$) and initialize our weight metrics W with zeros as this gives the best accuracy to our model. Further with each iteration our learning rate will decrease by \sqrt{t} where t is the iteration number. So that our model will converge easily.

We learned these hyperparameters by randomly using different values for each parameter it is more like random search hyperparameter tuning.

We assign different values for η and λ_{para} in search space with random values and predict using the model for each setting of hyperparameter. Our metric to find our hyperparameter. So we find on what setting of hyperparameter our model performs best and give high accuracy or produces minimum classification error.

eta	lambda_para	Validation Accuracy
0.001	0.5	0.7595
0.12	0.0001	0.9895
0.001	0.2	0.822

An instance from our parameter turning as shown above tells us that at values $\eta=0.12$ and $\lambda_{\text{para}}=0.0001$ gives best validation accuracy of 0.9895

Question 6

Plot the convergence curves in your PDF report offered by your chosen method as we do in lecture notebooks. This x-axis in the graph should be time taken and the y-axis should be the test classification accuracy (i.e. higher is better). Include this graph in your PDF file submission as an image.

Solution 6

