CS771 Introduction to Machine Learning
Assignment 1

Srujana Sabbani 22111083 *srujanas22@iitk.ac.in*

Shrutika Jadhav 22111082 *shrutika22@iitk.ac.in*

Aditya Kankriya 22111072 *adityask22@iitk.ac.in*

MitrajSinh Chavda 22111077 *mbchavda22@iitk.ac.in*

Haris Khan 22111026 *hariskhan22@iitk.ac.in*

1 Questions and Solutions

1.1 By giving a mathematical derivation, show there exists a way to map the binary digits 0,1 to sign $-1, +1$ as say $m : \{0, 1\} \to \{-1, +1\}$ and another way $f : \{-1, +1\} \to \{0, 1\}$ 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 $(b_1, b_2, \ldots, b_n)$ for any $n \in \mathbb{N}$ we have

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

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

Solution :

We have to find the function

$$m : \{0, 1\} \to \{-1, +1\} \quad and \quad f : \{-1, +1\} \to \{0, +1\}$$

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

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

Now solving the below equation using Two point form equation of line :

$$m : \{0, 1\} \to \{-1, +1\}$$

Let ,

$$x_1, y_1 = 0, +1$$
$$x_2, y_2 = +1, -1$$
$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y}{y_2 - y_1}$$
$$\frac{x - 0}{1 - 0} = \frac{y - 1}{-1 - 1}$$

Preprint. Under review.

$$\frac{x}{1} - \frac{y-1}{-2}$$
$$-2x = y - 1$$
$$\mathbf{y = 1 - 2x}$$
$$therefore, \mathbf{m(x) = 1 - 2x}$$

Similarly solving the below equation using Two point form equation of line :

$$f : \{-1, +1\} \rightarrow \{0, +1\}$$

Let ,

$$x_1, y_1 = -1, +1$$
$$x_2, y_2 = 0, +1$$
$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y}{y_2 - y_1}$$
$$\frac{x - (-1)}{1 - (-1)} = \frac{y - 1}{0 - 1}$$
$$\frac{x + 1}{2} - \frac{y - 1}{-1}$$
$$-x - 1 = 2y - 2$$
$$\mathbf{y = \frac{1 - x}{2}}$$
$$therefore, \mathbf{f(x) = \frac{1 - x}{2}}$$

Using the above derived formulas for f(x) and m(x) and substituting them in given equation(1).

For $\mathrm{n} = 5$, let's take random inputs,

Case 1: Input = 10001

$$LHS : XOR(1, 0, 0, 0, 1) = 0$$
$$RHS : f(m(1) * m(0) * m(0) * m(0) * m(1)) = f(-1 * 1 * 1 * 1 * -1) = f(1) = 0$$
$$\mathbf{LHS = RHS}$$

Case 2: Input = 11011

$$LHS : XOR(1, 1, 0, 1, 0) = 1$$
$$RHS : f(m(1) * m(1) * m(0) * m(1) * m(0)) = f(-1 * -1 * 1 * -1 * 1) = f(-1) = 1$$
$$\mathbf{LHS = RHS}$$

So, the mapping function f(x) and m(x) satisfy the given XOR equation(1).

From this we can conclude,

The function **m** which states that the odd occurrences of '1' bit results in '1' and the even occurrences of '1' bit results in '0' follows the XOR property i.e L H S

The function **f** which states that the odd occurrences of '-1' bit results in '-1' and the even occurrences of '-1' bit results in '1' follows the multiplication property i.e R H S.

Hence Proved

1.2 Let $(\mathrm{u}, \mathrm{a}), (\mathrm{v}, \mathrm{b}), (\mathrm{w}, \mathrm{c})$ 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 $\tilde{u} = [\mathrm{u}, \mathrm{a}]$, $\tilde{v} = [\mathrm{v}, \mathrm{b}]$, $\tilde{w} = [\mathrm{w}, \mathrm{c}]$, $\tilde{x} = [\mathrm{x}, 1] \epsilon \mathbb{N}^9$. 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:

$$sign\left(\tilde{u}^T\tilde{x}\right) \cdot sign\left(\tilde{v}^T\tilde{x}\right) \cdot sign\left(\tilde{w}^T\tilde{x}\right)$$

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

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

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

Solution :

We have three kinds of real numbers that are positive, negative and zero. So

$$sign\left(positive\right) = positive$$

$$sign\left(negative\right) = negative$$

$$sign\left(zero\right) = zero$$

Now we have to prove for any real numbers $r_1, r_2, \ldots, r_n$ for any $n \in \mathbb{N}$, we always have

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

By using the principle of mathematical induction,
for n = 2 as base case

$$\prod_{i=1}^{2} sign\left(r_i\right) = sign\left(\prod_{i=1}^{2} r_i\right)$$

$$sign\left(r_1\right) \cdot sign\left(r_2\right) = sign\left(r_1.r_2\right)$$

We have 6 combination of inputs for three kinds of real numbers

Case 1:

$$LHS : sign\left(positive\right) \cdot sign\left(positive\right) = positive$$

$$RHS : sign\left(positive.positive\right) = positive$$

$$\mathbf{LHS = RHS}$$

Case 2:

$$LHS : sign\left(positive\right) \cdot sign\left(zero\right) = zero$$

$$RHS : sign\left(positive.zero\right) = zero$$

$$\mathbf{LHS = RHS}$$

Case 3:

$$LHS : sign\left(positive\right) \cdot sign\left(negative\right) = negative$$

$$RHS : sign\left(positive.negative\right) = negative$$

$$\mathbf{LHS = RHS}$$

Case 4:

$$LHS : sign\,(negative) \cdot sign\,(negative) = positive$$
$$RHS : sign\,(negative.negative) = positive$$
$$\mathbf{LHS = RHS}$$

Case 5:

$$LHS : sign\,(negative) \cdot sign\,(zero) = zero$$
$$RHS : sign\,(negative.zero) = zero$$
$$\mathbf{LHS = RHS}$$

Case 6:

$$LHS : sign\,(zero) \cdot sign\,(zero) = zero$$
$$RHS : sign\,(zero.zero) = zero$$
$$\mathbf{LHS = RHS}$$

Hence the given equation satisfies for n = 2
Now by mathematical induction we assume that the given equation satisfies for n = k i.e.,

$$\prod_{i=1}^{k} sign\,(r_i) = sign\left(\prod_{i=1}^{k} r_i\right) \tag{2}$$

$$sign\,(r_1) \cdot sign\,(r_2) \ldots sign\,(r_k) = sign\,(r_1.r_2 \ldots r_k)$$

assuming

$$\prod_{i=1}^{k} r_i = x$$

Next for n = k + 1, we get

$$LHS : \prod_{i=1}^{k+1} sign\,(r_i) = \prod_{i=1}^{k} sign\,(r_i) \cdot sign\,(r_{k+1})$$

Using equation(2) we get,

$$= sign\left(\prod_{i=1}^{k} r_i\right) \cdot sign\,(r_{k+1})$$

$$= sign\,(x) \cdot sign\,(r_{k+1})$$

$$RHS : sign\left(\prod_{i=1}^{k+1} r_i\right) = sign\,(r_1.r_2 \ldots r_{k+1})$$

this can be written as,

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

$$= sign\left(x \cdot r_{k+1}\right)$$

This can be reduced to n = 2 case, which holds true for any real numbers.

$$\mathbf{LHS} = \mathbf{RHS}$$

Hence Proved

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

$$\left(\tilde{u}^T\tilde{x}\right) \cdot \left(\tilde{v}^T\tilde{x}\right) \cdot \left(\tilde{w}^T\tilde{x}\right)$$

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 $\mathrm{W}\epsilon\mathbb{R}^D$ such that for every $\tilde{x} \in \mathbb{R}^9$, we have $\left(\tilde{u}^T\tilde{x}\right) \cdot \left(\tilde{v}^T\tilde{x}\right) \cdot \left(\tilde{w}^T\tilde{x}\right) = \mathrm{W}^T\phi(\tilde{x})$.

Solution :

Solving for 3 PUF's :

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

$$= \sum_{j=1}^{9}\sum_{k=1}^{9}\sum_{l=1}^{9} \tilde{u}_j\tilde{v}_k\tilde{w}_l\tilde{x}_j\tilde{x_k}\tilde{x_l}$$

So,

$\left(\tilde{u}^T\tilde{x}\right)\cdot\left(\tilde{v}^T\tilde{x}\right)\cdot\left(\tilde{w}^T\tilde{x}\right) = u_1v_1w_1x_1x_1x_1 + u_1v_1w_2x_1x_1x_2 + u_1v_1w_3x_1x_1x_3 + \ldots + u_9v_9w_9x_0x_9x_9$

Now we have to map, $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$ to

$$\phi(\tilde{x}) = \begin{bmatrix} \tilde{x_1}\tilde{x_1}\tilde{x_1} & . & . & . & \tilde{x_1}\tilde{x_1}\tilde{x_9} & \tilde{x_1}\tilde{x_2}\tilde{x_1} & . & . & . & \tilde{x_1}\tilde{x_2}\tilde{x_9} \\ \tilde{x_1}\tilde{x_3}\tilde{x_1} & . & . & . & \tilde{x_1}\tilde{x_3}\tilde{x_9} & \tilde{x_1}\tilde{x_4}\tilde{x_1} & . & . & . & \tilde{x_1}\tilde{x_4}\tilde{x_9} \\ . & . & . & . & . & . & . & . & . & . \\ \tilde{x_2}\tilde{x_1}\tilde{x_1} & . & . & . & \tilde{x_2}\tilde{x_1}\tilde{x_9} & \tilde{x_2}\tilde{x_2}\tilde{x_1} & . & . & . & \tilde{x_2}\tilde{x_2}\tilde{x_9} \\ . & . & . & . & . & . & . & . & . & . \\ \tilde{x_9}\tilde{x_8}\tilde{x_1} & . & . & . & \tilde{x_9}\tilde{x_8}\tilde{x_9} & \tilde{x_9}\tilde{x_9}\tilde{x_1} & . & . & . & \tilde{x_9}\tilde{x_9}\tilde{x_9} \end{bmatrix}$$

We could see that our new linear equation has $9 * 9 * 9 = 729$ where $i, j, k\epsilon[1, 9]$ which means it maps to $9^3 = 729$ dimensional function.

So, In total there will be 729 elements in $\phi(\tilde{x})$. Hence there exists a way to map 9 dimensional vectors to 729 dimensional vectors as $\phi : \mathbb{R}^9 \rightarrow \mathbb{R}^{729}$ such that for any triple $(\tilde{u}, \tilde{v}, \tilde{w})$, there always exists a vector $\mathrm{W}\epsilon\mathbb{R}^{729}$ such that for every $\tilde{x} \in \mathbb{R}^9$, we have $\left(\tilde{u}^T\tilde{x}\right)\cdot\left(\tilde{v}^T\tilde{x}\right)\cdot\left(\tilde{w}^T\tilde{x}\right) = \mathrm{W}^T\phi(\tilde{x})$.

Therefore, $\left(\tilde{u}^T\tilde{x}\right)\cdot\left(\tilde{v}^T\tilde{x}\right)\cdot\left(\tilde{w}^T\tilde{x}\right)=\mathrm{W}^T\phi(\tilde{x})$

$$where\ \mathrm{W}=\begin{bmatrix}\tilde{u_1}\tilde{v_1}\tilde{w_1} & \tilde{u_1}\tilde{v_1}\tilde{w_2} & . & . & . & \tilde{u_1}\tilde{v_1}\tilde{w_9}\\ \tilde{u_1}\tilde{v_2}\tilde{w_1} & \tilde{u_1}\tilde{v_2}\tilde{w_2} & . & . & . & \tilde{u_1}\tilde{v_2}\tilde{w_9}\\ . & . & . & . & . & .\\ \tilde{u_2}\tilde{v_1}\tilde{w_1} & \tilde{u_2}\tilde{v_1}\tilde{w_2} & . & . & . & \tilde{u_2}\tilde{v_1}\tilde{w_9}\\ . & . & . & . & . & .\\ \tilde{u_9}\tilde{v_9}\tilde{w_1} & \tilde{u_9}\tilde{v_9}\tilde{w_2} & . & . & . & \tilde{u_9}\tilde{v_9}\tilde{w_9}\end{bmatrix}$$

1.4 Python code

1.5 For the method you implemented, describe in your PDF report what 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 $\eta/\sqrt{t}$ 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 and 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 :

We used SGD with the Sub Gradient method because we used hinge loss with SVM and hinge loss is convex but non-differentiable. We are using three hyperparameters:

1. strength - It is a regularization hyperparameter, we took the value 10 initially and then increased it to 10000.

2. step_length - It is a learning rate hyperparameter, we took the value 0.1 which gave us massive hinge loss so we kept decreasing the value till 0.000001(gives the best result) with intermediate values 0.01, 0.001, 0.0001, 0.00001, 0.000005.

3. max_epochs - It is the number of epochs, we had its value of 20 initially then we reduced it to 5 for a better response time with intermediate values 20, 15, 10, 5.

1.6 Plot the convergence curves in your PDF report offered by your chosen method as we do in lecture notebooks. The 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 :