CS771
Introduction to Machine Learning
Group 15 : GradientGeeks

Assignment 1: Part 1 and Part 3

Group Members:

Milan Anand Raj (200584)
Sarthak Motwani (200887)
Abhay Sisodia (200014)
Shruti Agarwal (208170957)
Devansh Goyal (200317)
Rishu Kumar (200797)

1 Part 1: Derivation of the Linear Model for CAR-PUF

Let a CAR-PUF has two arbiter PUFs – a working PUF and a reference PUF, as well as a secret threshold value $\tau > 0$. Let Δ_w, Δ_r be the difference in timings experienced for the two PUFs on the same 32-bit challenge. The response to this challenge is 0 if $|\Delta_w - \Delta_r| \leq \tau$ and the response is 1 if $|\Delta_w - \Delta_r| > \tau$. From the linear model formulation of the arbiter-PUF as derived in class, we had seen how the 32-dimensional input vector $\mathbf{c} \in \{0,1\}^{32}$ to the arbiter-PUF is mapped to a 32-dimensional vector $\mathbf{x}$ and the parameters of the multiplexers of the arbiter-PUF are mapped to a 32-dimensional vector $\mathbf{w}$ and an intercept term b, such that the delay of the PUF is given by:

$$\Delta = \mathbf{w}^T\mathbf{x} + b$$

Thus the response of a single arbiter PUF can be predicted using:

$$r = \frac{1 + \text{sign}(\mathbf{w}^T\mathbf{x} + b)}{2}$$

Starting with this linear model for each PUF in the CAR-PUF, we denote the delays of the working PUF by Δ_w and that of reference PUF by Δ_r using the following equations, for a 32-bit challenge $\mathbf{x}, \mathbf{w_1}, \mathbf{w_2} \in \mathbb{R}^{32}$ and $b \in \mathbb{R}$. We have,

$$\Delta_{\text{w}} = \mathbf{w_1}^T\mathbf{x} + b_1$$

$$\Delta_{\text{r}} = \mathbf{w_2}^T\mathbf{x} + b_2$$

$$\Delta_{\text{w}} - \Delta_{\text{r}} = (\mathbf{w_1} - \mathbf{w_2})^T\mathbf{x} + (b_1 - b_2) \tag{1}$$

Define,

$$\mathbf{w'} = \mathbf{w_1} - \mathbf{w_2} \text{ and } b' = b_1 - b_2 \implies \Delta_{\text{w}} - \Delta_{\text{r}} = \mathbf{w'}^T\mathbf{x} + b'$$

(Note: The Primed notation does not mean 'transpose' here.)
The response of the CAR-PUF is as follows:

$$\text{If } |\Delta_{\text{w}} - \Delta_{\text{r}}| \leq \tau, \text{the response is 0. Else, the response is 1. This is same as,}$$

$$\text{If } -\tau \leq \Delta_{\text{w}} - \Delta_{\text{r}} \leq \tau, \text{ response is 0. Else, the response is 1.}$$

$$\text{Now, } -\tau \leq \Delta_{\text{w}} - \Delta_{\text{r}} \leq \tau \implies -\tau \leq \mathbf{w'}^T\mathbf{x} + b' \leq \tau$$

$$\text{Therefore, the response is 0 only when } \mathbf{w'}^T\mathbf{x} + b' + \tau \geq 0 \text{ and } \mathbf{w'}^T\mathbf{x} + b' - \tau \leq 0.$$

$$\text{And, the response is 1 only when either } \mathbf{w'}^T\mathbf{x} + b' + \tau < 0 \text{ or } \mathbf{w'}^T\mathbf{x} + b' - \tau > 0.$$

Consider the product, $Q = (\mathbf{w'}^T\mathbf{x} + b' + \tau)(\mathbf{w'}^T\mathbf{x} + b' - \tau)$. Then, we can say that the response is 0 only when $Q \leq 0$. This is because if $\mathbf{w'}^T\mathbf{x} + b' + \tau \geq 0$ and $\mathbf{w'}^T\mathbf{x} + b' - \tau \leq 0$, then $Q \leq 0$. Also, the case that $\mathbf{w'}^T\mathbf{x} + b' + \tau \leq 0$ and $\mathbf{w'}^T\mathbf{x} + b' - \tau \geq 0$ is not possible because $\tau \geq 0$. Hence, $Q \leq 0 \iff$ Response is 0. Similarly, the response is 1 only when $Q > 0$. This is because $\mathbf{w'}^T\mathbf{x} + b' + \tau < 0 (\implies \mathbf{w'}^T\mathbf{x} + b' - \tau < 0)$ and $\mathbf{w'}^T\mathbf{x} + b' - \tau > 0 (\implies \mathbf{w'}^T\mathbf{x} + b' + \tau > 0) \iff Q > 0 \iff$ Response is 1. Now,

$$Q = (\mathbf{w'}^T\mathbf{x} + b' + \tau)(\mathbf{w'}^T\mathbf{x} + b' - \tau) \implies Q = (\mathbf{w'}^T\mathbf{x} + b')^2 - \tau^2 = (\mathbf{w'}^T\mathbf{x})^2 + 2b'\mathbf{w'}^T\mathbf{x} + b'^2 - \tau^2$$

Considering each term in Q one by one, we have

$$(\mathbf{w'}^T\mathbf{x})^2 = (w'_1x_1 + w'_2x_2 + \ldots + w'_{32}x_{32})^2$$

$$\implies (\mathbf{w'}^T\mathbf{x})^2 = {w'_1}^2{x_1}^2 + \ldots + {w'_{32}}^2{x_{32}}^2 + 2w'_1w'_2x_1x_2 + 2w'_1w'_3x_1x_3 + \ldots + 2w'_{31}w'_{32}x_{31}x_{32}$$

However, we note that each ${x_i}^2 = 1 \, \forall i \in [32]$. This is because, according to the linear model for an arbiter PUF derived in the class, $x_i = d_i.d_{i+1}...d_{32}$, where $d_i = 1 - 2c_i$. Since $c_i \in \{0,1\}$, $d_i \in \{-1,1\}$ and hence $x_i \in \{-1,1\} \implies {x_i}^2 = 1 \, \forall i \in [32]$. Hence, the first 32 terms ${w'_1}^2{x_1}^2 + \ldots + {w'_{32}}^2{x_{32}}^2 = {w'_1}^2 + \ldots + {w'_{32}}^2$ get clubbed within the intercept term. Therefore, $(\mathbf{w'}^T\mathbf{x})^2$ **effectively** contributes $\binom{32}{2}$ terms to the model.

$$2b'\mathbf{w'}^T\mathbf{x} = 2b'w'_1x_1 + \ldots + 2b'w'_{32}x_{32}$$

Therefore $2b'\mathbf{w'}^T\mathbf{x}$ has 32 terms.
Thus, if we define the mapping $\phi(\mathbf{x})$ in the following manner:

$$\mathbf{X'} = \phi(\mathbf{x}) = (x_1x_2, x_1x_3, \ldots, x_{31}x_{32}, x_1, x_2, \ldots, x_{32}) \tag{2}$$

Thus $\mathbf{X'}$ has $\binom{32}{2} + 32 = 528$ dimensions. And the corresponding 528-dimensional linear model coefficient vector $\mathbf{W'}$ is:

$$\mathbf{W'} = (2w'_1w'_2, 2w'_1w'_3, \ldots, 2w'_{31}w'_{32}, 2b'w'_1, 2b'w'_2, \ldots, 2b'w'_{32}) \tag{3}$$

Also, the bias term for this model is $b'' = b'^2 + {w'_1}^2 + {w'_2}^2 + \ldots + {w'_{32}}^2 - \tau^2$. Hence, the final linear model that we estimate is the following:

$$r = \frac{1 + \text{sign}(\mathbf{W'}^T\mathbf{X'} + b'')}{2} \tag{4}$$

2 Part 1: An Alternative Derivation

We also note that in the beginning of the previous derivation, if we had taken the bias term inside $\mathbf{w'}$ in (1), we would get a more compact representation of the final linear model. Therefore, we now show the alternative derivation. This is the model that we have implemented in our code because the mapping from $\mathbf{x}$ to $\mathbf{X'}$ in this case was more convenient to implement using the *Khatri-Rao Product*.

$$\Delta_{\text{w}} = \mathbf{w_1}^T\mathbf{x} + b_1$$

$$\Delta_{\text{r}} = \mathbf{w_2}^T\mathbf{x} + b_2$$

Define,

$$\mathbf{w'_i} = (\mathbf{w_i}, b_i) \in \mathbb{R}^{33} \text{ and } \mathbf{x'} = (\mathbf{x}, 1) \text{ for i=1, 2. Therefore,}$$

(Note: The Primed notation does not mean 'transpose' here. It is just used to differentiate the vectors after encapsulating the bias terms.)

$$\Delta_{\text{w}} - \Delta_{\text{r}} = (\mathbf{w'_1} - \mathbf{w'_2})^T\mathbf{x'} \tag{5}$$

Define,

$$\mathbf{w'} = \mathbf{w'_1} - \mathbf{w'_2} \implies \Delta_{\text{w}} - \Delta_{\text{r}} = \mathbf{w'}^T\mathbf{x'}$$

$$\text{If } -\tau \le \Delta_{\text{w}} - \Delta_{\text{r}} \le \tau, \text{ response is 0. Else, the response is 1.}$$

$$\text{Now, } -\tau \le \Delta_{\text{w}} - \Delta_{\text{r}} \le \tau \implies -\tau \le \mathbf{w'}^T\mathbf{x'} \le \tau$$

$$\text{Therefore, the response is 0 only when } \mathbf{w'}^T\mathbf{x'} + \tau \ge 0 \text{ and } \mathbf{w'}^T\mathbf{x'} - \tau \le 0.$$

$$\text{And, the response is 1 only when either } \mathbf{w'}^T\mathbf{x'} + \tau < 0 \text{ or } \mathbf{w'}^T\mathbf{x'} - \tau > 0.$$

Consider the product, $Q = (\mathbf{w'}^T\mathbf{x'} + \tau)(\mathbf{w'}^T\mathbf{x'} - \tau)$. Using the similar arguments used in the previous derivation, we can show that $Q \le 0 \iff$ Response is 0. Also, $Q > 0 \iff$ Response is 1. Now,

$$Q = (\mathbf{w'}^T\mathbf{x'} + \tau)(\mathbf{w'}^T\mathbf{x'} - \tau) \implies Q = (\mathbf{w'}^T\mathbf{x'})^2 - \tau^2 \implies Q = (\mathbf{w'}^T\mathbf{x'})^2 - \tau^2$$

$$(\mathbf{w'}^T\mathbf{x'})^2 = (w'_1x_1 + w'_2x_2 + \ldots + w'_{32}x_{32} + w'_{33}x_{33})^2$$

Note that here $x_{33} = 1$, though it is now treated as a feature.

$$\implies (\mathbf{w'}^T\mathbf{x'})^2 = {w'_1}^2{x_1}^2 + \ldots + {w'_{33}}^2{x_{33}}^2 + 2w'_1w'_2x_1x_2 + 2w'_1w'_3x_1x_3 + \ldots + 2w'_{32}w'_{33}x_{32}x_{33}$$

However, we note that each ${x_i}^2 = 1\ \forall i \in [33]$. This is because, according to the linear model for an arbiter PUF derived in the class, $x_i = d_i.d_{i+1}...d_{32}$, where $d_i = 1 - 2c_i$. Since $c_i \in \{0, 1\}$, $d_i \in \{-1, 1\}$ and hence $x_i \in \{-1, 1\} \implies {x_i}^2 = 1\ \forall i \in [32]$. Also, ${x_{33}}^2 = 1$, as noted above. Hence, the first 33 terms ${w'_1}^2{x_1}^2 + \ldots + {w'_{33}}^2{x_{33}}^2 = {w'_1}^2 + \ldots + {w'_{33}}^2$ get clubbed within the intercept term. Therefore, $(\mathbf{w'}^T\mathbf{x})^2$ **effectively** contributes $\binom{33}{2}$ terms to the model. Thus, if we define the mapping $\phi(\mathbf{x'})$ in the following manner:

$$\mathbf{X'} = \phi(\mathbf{x'}) = (x_1x_2, x_1x_3, \ldots, x_{32}x_{33}) \tag{6}$$

Thus **X'** has $\binom{33}{2} = 528$ dimensions. And the corresponding 528-dimensional linear model coefficient vector **W'** is:

$$\mathbf{W'} = (2w_1'w_2', 2w_1'w_3', \ldots, 2w_{32}'w_{33}') \tag{7}$$

Also, the bias term for this model is $b' = {w_1'}^2 + {w_2'}^2 + \ldots + {w_{33}'}^2 - \tau^2$. Hence, the final linear model that we estimate is the following:

$$r = \frac{1 + \text{sign}(\mathbf{W'}^T\mathbf{X'} + b')}{2} \tag{8}$$

Hence, both these models can successfully predict the responses of a CAR-PUF, when a sufficient number of challenge-response pairs are already shown to the model.

3 Part 3

We check how the following affect training time and test accuracy:

1. changing the loss hyperparameter in LinearSVC (hinge vs squared hinge)
2. setting C in LinearSVC and LogisticRegression to high/low/medium values
3. changing tol in LinearSVC and LogisticRegression to high/low/medium value

Table 1: Effect of Changing Loss Hyperparameter in LinearSVC (Other Parameters: Default)

Loss	Penalty	C	Train Time (s)	Test Accuracy
Hinge	L2	1.0	11.97	0.989
Hinge	L2	7.0	12.60	0.994
Squared Hinge	L2	1.0	12.765	0.9906
Squared Hinge	L2	7.0	12.60	0.991

Inference on Table 1: Changing the loss Hyperparameter in LinearSVC has minimal impact on training time. However, we observe that at low C values test accuracy for 'Squared Hinge' loss is better than that of 'Hinge' loss, whereas at larger C values, test accuracy of 'Hinge' loss is better.

Table 2: Effect of Setting C in LinearSVC and LogisticRegression (Other Parameters: Default)

Model	C	Train Time (s)	Test Accuracy
LinearSVC	0.5	13.14	0.9912
	1.0	13.03	0.9918
	5.0	13.19	0.9904
	10.0	12.70	0.9901
	15.0	12.60	0.9905
	20.0	12.65	0.9905
LogisticRegression	0.5	2.23	0.9905
	1.0	2.61	0.9912
	5.0	2.43	0.9918
	10.0	2.68	0.9924
	15.0	2.54	0.993
	20.0	2.66	0.9925

Inference on Table 2: Increasing the regularization parameter C initially boosts test accuracy in both LinearSVC and LogisticRegression. However, as C continues to increase, the improvements in accuracy diminish, while training times remain relatively consistent in both.

Table 3: Effect of Changing tol in LinearSVC and LogisticRegression (Other Parameters: Default)

Model	C	Tol	Train Time (s)	Test Accuracy
LinearSVC	1.0	0.00001	14.37	0.9913
	1.0	0.0001	13.72	0.9909
	1.0	0.01	13.30	0.9911
	1.0	1.0	12.90	0.9914
	1.0	2.0	10.94	0.9905
LogisticRegression	15.0	0.00001	2.57	0.9929
	15.0	0.0001	2.73	0.993
	15.0	0.01	1.83	0.983
	15.0	1.0	1.20	0.50
	15.0	2.0	1.70	0.50

Inference on Table 3: In LinearSVC, altering the tolerance affects training times noticeably but has only a slight impact on accuracy. Conversely, in LogisticRegression, tolerance adjustments have a significant impact on accuracy, with higher tol values leading to substantial drops, while training times remain relatively stable.