Assignment 1 Solution

Name of Group: **Data Geeks**

Group Members:

Arshit	220209
Harsh Agrawal	220425
Kritnandan	220550
Kundan Kumar	220568
Prakhar	220786
Rudradeep Datta	220919

Indian Institute of Technology Kanpur

CS771 Introduction to Machine Learning

Instructor: Prof. Purushottam Kar

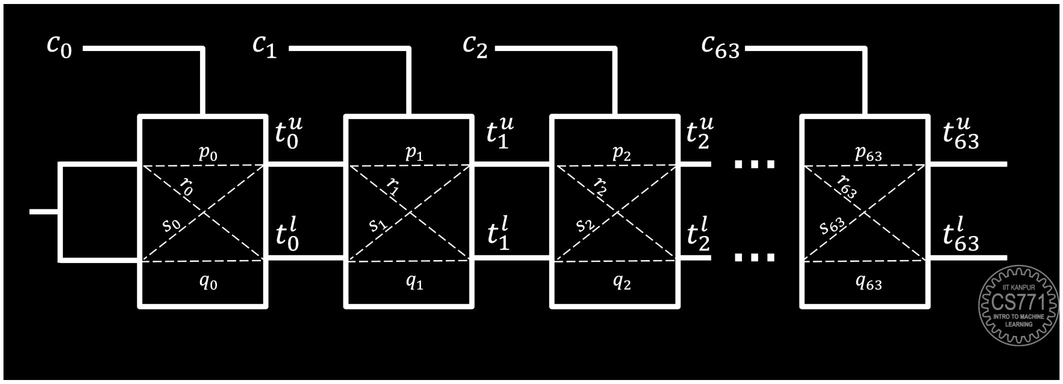
July 17, 2024

1 Derivation of a Linear Model for Predicting COCO PUF Delay

We will use a straightforward approach to solve this problem. The strategy involves leveraging the known differences in time delays for the upper and lower signals, as well as the sum of their delays, to predict the upper delay.

First, we will derive the difference in time delays for 32-bit input in a way similar to as outlined in the lectures, and then proceed to derive the sum of these delays.

Also the notations are same as described in lectures.



1.1 Deriving the difference of Delays

Here, t_i^u and t_i^l represent the times at which the signal departs from the i_{th} MUX pair. Expressing t_i^u in terms of t_{i-1}^u, t_{i-1}^l and c_i, we obtain:

$$t_i^u = (1 - c_i) \cdot (t_{i-1}^u + p_i) + c_i \cdot (t_{i-1}^l + s_i) - (1)$$
$$t_i^l = (1 - c_i) \cdot (t_{i-1}^l + q_i) + c_i \cdot (t_{i-1}^u + r_i) - (2)$$

Thus following the lecture slides we have, $\Delta_i = t_i^u - t_i^l$. Subtracting equations (1), (2) we get, $\Delta_i = \Delta_{i-1} \cdot d_i + \alpha_i \cdot d_i + \beta_i$, where α_i, β_i depend on constants that are indeterminable from the physical/ measurable perspective and thus we call them system constants and are given by:

$$\alpha_i = (p_i - q_i + r_i - s_i)/2 \quad \beta_i = (p_i - q_i - r_i + s_i)/2$$

where p_i, q_i, r_i, s_i are system parameters. Also d_i is governed by the challenge bits/input $(c_i' s)$:

$$d_i = (1 - 2 \cdot c_i)$$

Δ_{-1} = 0. Moreover, observing the recursion unfold carefully we can simplify this relation further:

$$\Delta_0 = \alpha_0 \cdot d_0 + \beta_0$$
$$\Delta_1 = \alpha_0 \cdot d_1 \cdot d_0 + (\alpha_1 + \beta_0) \cdot d_1 + \beta_1$$
$$\Delta_2 = \alpha_0 \cdot d_2 \cdot d_1 \cdot d_0 + (\alpha_1 + \beta_0) \cdot d_2 \cdot d_1 + (\alpha_2 + \beta_1) \cdot d_2 + \beta_2$$
$$\vdots$$

The only Δ we require is Δ_{31}, the last output.

$$\Delta_{31} = w_0 \cdot x_0 + w_1 \cdot x_1 + \cdots + w_{31} \cdot x_{31} + \beta_{31} = \mathbf{w}^T \cdot \mathbf{x} + \mathbf{b} = \mathbf{W}^T \cdot \mathbf{X}$$

where **w, x** are 32 dimensional vectors and **W, X** are 33 dimensional, just including the bias term $W_{32} = \beta_{31}$ and $X_{32} = 1$. Each term of **w, x** being:

$$b = \beta_{31} \qquad w_0 = \alpha_0 \qquad w_i = \alpha_i + \beta_{i-1} \qquad x_i = \prod_{j=i}^{31} d_i$$

1.2 Deriving the Sum of Delays

We have,

$$t_i^u = (1 - c_i) \cdot (t_{i-1}^u + p_i) + c_i \cdot (t_{i-1}^l + s_i) - (1)$$
$$t_i^l = (1 - c_i) \cdot (t_{i-1}^l + q_i) + c_i \cdot (t_{i-1}^u + r_i) - (2)$$

Adding both the equations,

$$t_i^u + t_i^l = t_{i-1}^u + t_{i-1}^l + (p_i + q_i) + c_i \cdot (s_i + r_i - p_i - q_i)$$

Put

$$(s_i + r_i - p_i - q_i) = r_i$$

to get,

$$t_i^u + t_i^l = t_{i-1}^u + t_{i-1}^l + (p_i + q_i) + c_i \cdot (r_i)$$

Subsituting the value of $i = 31, 30, ..1, 0$ and adding the equations:

$$t_{31}^u + t_{31}^l = \cancel{t_{30}^u} + \cancel{t_{30}^l} + (p_{31} + q_{31}) + c_{31} \cdot (r_{31})$$
$$\cancel{t_{30}^u} + \cancel{t_{30}^l} = \cancel{t_{29}^u} + \cancel{t_{29}^l} + (p_{30} + q_{30}) + c_{30} \cdot (r_{30})$$
$$.$$
$$.$$
$$.$$
$$\cancel{t_0^u} + \cancel{t_0^l} = t_{-1}^u + t_{-1}^l + (p_0 + q_0) + c_0 \cdot (r_0)$$

We finally get,

$$t_{31}^u + t_{31}^l = (c_{31} \cdot r_{31}) + (c_{30} \cdot r_{30}) + ... + (c_0 \cdot r_0) \tag{1}$$

where

$$r_i = (s_i + r_i - p_i - q_i)$$

We also derived the difference in previous sections as :

$$t_{31}^u - t_{31}^l = w_0 \cdot x_0 + w_1 \cdot x_1 + \cdots + w_{31} \cdot x_{31} + \beta_{31} \tag{2}$$

where

$$w_0 = \alpha_0 \qquad w_i = \alpha_i + \beta_{i-1} \qquad x_i = \prod_{j=i}^{31} d_i \qquad d_i = (1 - 2 \cdot c_i)$$
$$\alpha_i = (p_i - q_i + r_i - s_i)/2 \quad \beta_i = (p_i - q_i - r_i + s_i)/2$$

where p_i, q_i, r_i, s_i are system parameters.

Adding 1 and 2,

$$2 * t_{31}^u = (c_{31} \cdot r_{31}) + (c_{30} \cdot r_{30}) + ... + (c_0 \cdot r_0) + w_0 \cdot x_0 + w_1 \cdot x_1 + \cdots + w_{31} \cdot x_{31} + \beta_{31}$$

Dividing both sides by 2:

$$t_{31}^u = \frac{1}{2} \left((c_{31} \cdot r_{31}) + (c_{30} \cdot r_{30}) + \cdots + (c_0 \cdot r_0) + w_0 \cdot x_0 + w_1 \cdot x_1 + \cdots + w_{31} \cdot x_{31} + \beta_{31} \right)$$

Rewriting x_{31} as $1 - 2.c_{31}$ and taking c_{31} common from $c_{31}.r_{31}$ and $2.w_{31}.c_{31}$, we get

$$t_{31}^u = \frac{1}{2} \left((c_{31} \cdot (r_{31} - 2.w_{31})) + (c_{30} \cdot r_{30}) + \cdots + (c_0 \cdot r_0) + w_0 \cdot x_0 + w_1 \cdot x_1 + \cdots + (\beta_{31} + w_{31}) \right)$$

In matrix form, this can be expressed as:

$$t_{31}^u = \frac{1}{2} \left(W^T \cdot \phi(c) + b \right)$$

where:

$$\phi(c) = \begin{bmatrix} c_{31}, & c_{30}, & \cdots & c_0, & x_0, & x_1, & \cdots & x_{30} \end{bmatrix}^T,$$
$$W = \begin{bmatrix} (r_{31} - 2.w_{31}), & r_{30}, & \cdots & r_0, & w_0 & w_1 & \cdots & w_{30} \end{bmatrix}^T,$$
$$b = \beta_{31} + w_{31}$$

2 Solution for Part 2:

The dimensionality of the linear model to predict the arrival time of the upper signal for an arbiter PUF is **63**.

3 Solution for Part 3:

In this solution we denote parameters for PUF0 as $A_{0,i}$ and parameters for PUF1 as $A_{1,i}$.

Linear Model For Response0:

$$t^u_{0,31} = \frac{1}{2}\left((c_{31} \cdot (r_{0,31} - 2.w_{0,31})) + (c_{30} \cdot r_{0,30}) + \cdots + (c_0 \cdot r_{0,0}) + w_{0,0} \cdot x_0 + w_{0,1} \cdot x_1 + \cdots + (\beta_{0,31} + w_{0,31})\right)$$

$$t^u_{1,31} = \frac{1}{2}\left((c_{31} \cdot (r_{1,31} - 2.w_{1,31})) + (c_{30} \cdot r_{1,30}) + \cdots + (c_0 \cdot r_{1,0}) + w_{1,0} \cdot x_0 + w_{1,1} \cdot x_1 + \cdots + (\beta_{1,31} + w_{1,31})\right)$$

Now

$$\begin{aligned} t^u_{0,31} - t^u_{1,31} = \frac{1}{2}\,(& c_{31} \cdot ((r_{0,31} - 2w_{0,31}) - (r_{1,31} - 2w_{1,31})) + c_{30} \cdot (r_{0,30} - r_{1,30}) + \cdots \\ & + c_0 \cdot (r_{0,0} - r_{1,0}) + (w_{0,0} - w_{1,0}) \cdot x_0 + (w_{0,1} - w_{1,1}) \cdot x_1 + \cdots \\ & + ((\beta_{0,31} + w_{0,31}) - (\beta_{1,31} + w_{1,31})) \\ &) \end{aligned}$$

In matrix form, this can be expressed as:

$$t^u_{0,31} - t^u_{1,31} = \frac{1}{2}\left(\tilde{W}^T \cdot \phi(c) + \tilde{b}\right)$$

Where:

$$\phi(c) = [c_{31}, \quad c_{30}, \quad \cdots \quad c_0, \quad x_0, \quad x_1, \quad \cdots \quad x_{30}]^T,$$

$$\tilde{W} = \begin{bmatrix} ((r_{0,31} - 2w_{0,31}) - (r_{1,31} - 2w_{1,31})), & (r_{0,30} - r_{1,30}), & \cdots & (r_{0,0} - r_{1,0}), \\ (w_{0,0} - w_{1,0}), & (w_{0,1} - w_{1,1}), & \cdots & (w_{0,30} - w_{1,30}) \end{bmatrix}^T$$

$$\tilde{b} = ((\beta_{0,31} + w_{0,31}) - (\beta_{1,31} + w_{1,31}))$$

Linear Model For Response1:

$$t^l_{0,31} = \frac{1}{2}\left((c_{31} \cdot (r_{0,31} + 2.w_{0,31})) + (c_{30} \cdot r_{0,30}) + \cdots + (c_0 \cdot r_{0,0}) - w_{0,0} \cdot x_0 - w_{0,1} \cdot x_1 + \cdots - (\beta_{0,31} + w_{0,31})\right)$$

$$t^l_{1,31} = \frac{1}{2}\left((c_{31} \cdot (r_{1,31} + 2.w_{1,31})) + (c_{30} \cdot r_{1,30}) + \cdots + (c_0 \cdot r_{1,0}) - w_{1,0} \cdot x_0 - w_{1,1} \cdot x_1 + \cdots - (\beta_{1,31} + w_{1,31})\right)$$

Now

$$\begin{aligned} t^l_{0,31} - t^l_{1,31} = \frac{1}{2}\,(& c_{31} \cdot ((r_{0,31} + 2w_{0,31}) - (r_{1,31} + 2w_{1,31})) + c_{30} \cdot (r_{0,30} - r_{1,30}) + \cdots \\ & + c_0 \cdot (r_{0,0} - r_{1,0}) - (w_{0,0} - w_{1,0}) \cdot x_0 - (w_{0,1} - w_{1,1}) \cdot x_1 - \cdots \\ & - ((\beta_{0,31} + w_{0,31}) - (\beta_{1,31} + w_{1,31})) \\ &) \end{aligned}$$

In matrix form, this can be expressed as:

$$t^l_{0,31} - t^l_{1,31} = \frac{1}{2}\left(\tilde{W}^T \cdot \phi(c) + \tilde{b}\right)$$

Where:

$$\phi(c) = [c_{31}, \quad c_{30}, \quad \cdots \quad c_0, \quad x_0, \quad x_1, \quad \cdots \quad x_{30}]^T,$$

$$\tilde{W} = \begin{bmatrix} ((r_{0,31} + 2w_{0,31}) - (r_{1,31} + 2w_{1,31})), & (r_{0,30} - r_{1,30}), & \cdots & (r_{0,0} - r_{1,0}), \\ -(w_{0,0} - w_{1,0}), & -(w_{0,1} - w_{1,1}), & \cdots & -(w_{0,30} - w_{1,30}) \end{bmatrix}^T$$

$$\tilde{b} = ((\beta_{1,31} + w_{1,31}) - (\beta_{0,31} + w_{0,31}))$$

4 Solution for Part 4:

Dimensionality do your need the linear model to have to predict Response0 and Response1 for a COCO-PUF is also **63**

5 Solution for Part 5:

Zipped ***Solution*** *to Assignment 1*

6 Solution for Part 6: Perfomance Analysis for Models

(a) Performance Comparison of LinearSVC with Different Loss Functions:

Loss Function	Total Features	Model Train Time (s)	Map Time (s)	Test Accuracy(0)	Test Accuracy(1)
Hinge	63	11.3759	0.0069	0.95517	0.9980
Squared Hinge	63	0.6856	0.0059	0.99801	0.9994

Analysis: From the table, we observe that the model trained with the hinge loss achieves a slightly higher test accuracy compared to when trained with the squared hinge loss. However, the difference in accuracy is insignificant. Interestingly, the model trained with the squared hinge loss takes slightly longer to train compared to the model trained with the hinge loss.

Possible reasons for the observed differences:

- The squared hinge loss penalizes outliers more strongly than the hinge loss, potentially leading to improved generalization performance.
- The optimization problem associated with the squared hinge loss uses primal to solve while Hinge loss used dual to optimize
- The dataset characteristics and the specific challenges posed by the COCO-PUF may favor one loss function over the other, influencing the model's performance.

(b) Performance Comparison based on the value of C:

- **Analysis for LinearSVC with different C values:**

C Value	Total Features	Model Train Time (s)	Test Accuracy(0)	Test Accuracy(1)
Low	63	0.0421	0.9178	0.9685
Medium	63	1.0768	0.9799	0.9938
High	63	0.8013	0.9801	0.9994

- **Analysis for LogisticRegression with different C values:**

C Value	Total Features	Model Train Time (s)	Test Accuracy(0)	Test Accuracy(1)
Low	63	0.5076	0.916	0.9682
Medium	63	1.0874	0.9771	0.9912
High	63	4.6590	0.9809	0.999

From the analysis, we observe the following:

For LinearSVC:

- The model trained with a high C value achieves the highest test accuracy among the three settings.
- The model trained with a medium C value has the highest training time, which may indicate that a lower C value leads to a more complex optimization problem.

For LogisticRegression:

- Similar to LinearSVC, the model trained with a high C value achieves the highest test accuracy.
- The high C value model in LogisticRegression achieves the highest test accuracy, indicating that a higher regularization strength may improve generalization performance.

Overall, the choice of C value significantly affects the training time and test accuracy of both LinearSVC and LogisticRegression models.

(c) Effect of Variation in Tolerance for LinearSVC and LogisticRegression Models:

For part (c) of the analysis, we evaluate the performance of the LinearSVC and LogisticRegression models with varying values of the `tol` hyperparameter. The `tol` hyperparameter controls the tolerance for the optimization algorithm, with lower values potentially leading to longer training times but potentially improved convergence and accuracy. We present the results in the following table:

Model	Tolerance	Total Features	Model Train Time	Test Accuracy(0)	Test Accuracy(1)
LinearSVC	Low	63	0.9623s	0.9801	0.9994
LinearSVC	Medium	63	0.9800s	0.9990	0.9909
LinearSVC	High	63	0.4999s	0.4905	0.9359
LogisticRegression	Low	63	3.5518s	0.9809	0.9994
LogisticRegression	Medium	63	2.5669s	0.9808	0.9984
LogisticRegression	High	63	0.1841s	0.4999	0.4905

From the table, we observe the following:

- For LinearSVC, varying the tolerance parameter leads to significant differences in training time, with the medium tolerance resulting in the longest training time and the high tolerance resulting in the shortest training time.
- However, the test accuracy of the LinearSVC model decreases as the tolerance increases, indicating that higher tolerance values may lead to poorer generalization performance.
- For LogisticRegression, the training time across different tolerance values are higher compared to LinearSVC.
- Similar to LinearSVC, the test accuracy of the LogisticRegression model also shows a decreasing trend as the tolerance increases, albeit to a lesser extent.

These results suggest that the choice of tolerance parameter can have a significant impact on both the training time and the test accuracy of the models. Lower tolerance values may lead to longer training times but potentially better performance, while higher tolerance values may result in faster training times but poorer generalization performance. It is crucial to strike a balance between training time and model performance when selecting the tolerance parameter.