Andy, Yu-Guang Chen

EE6094 CAD for VLSI Design
Programming Assignment 4: IR-drop Prediction
(Due: 23:59:59, 2024/06/13)

Introduction

Dynamic IR-drop analysis is often done by running simulation-based tools (ex: Redhawk-SC), which is known to be accurate but very time-consuming since there will be tons of vectors (input patterns) to test. Therefore, a fast IR-drop estimator is highly needed to solve this problem. In this programming assignment, you are asked to implement and train an IR-drop predictor using ensemble learning model "XGBoost" to estimate the IR-drop of each cell in the circuit.

Background

Dynamic IR drop is the deviation of the power supply level from V_{DD}/GND caused by localized power demand and switching patterns of each cell in the circuit, which is an important issue for design sign-off at advanced semiconductor technology. Excessive IR-drop can significantly degrade performance of a circuit or even lead to functional failures. As such, it is important to verify if IR-drop satisfies design constraints and identify the constraint violation regions, a.k.a. IR-drop hotspots.

Practically, dynamic IR drop estimation is often obtained by running simulation-based commercial tools, which are known to be accurate but very time-consuming. To solve this problem, machine learning based approaches have been explored in an effort to achieve faster estimation (ex: linear regression, SVM, XGBoost, etc.). These works learn to predict dynamic IR drop of each cell through features such as cell positions, timing windows, path resistance, etc. with supervised machine learning techniques.

Problem formulation

Input: Raw features of each cell in the circuit

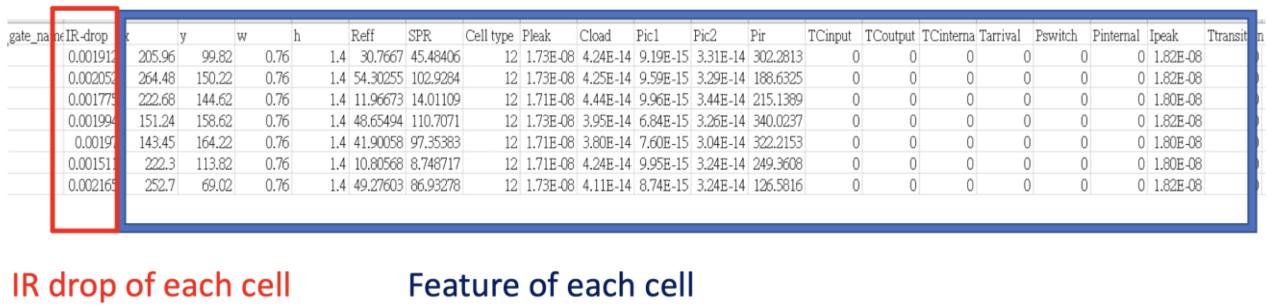
Output: Predicted IR-drop values of each cell

Learning objective: minimize the error between predicted value and real IR-drop value

Dataset format

The entire dataset contains 100 csv files of circuit MEMC (MEMC_1.csv ~ MEMC_100.csv), each file represents the circuit operates in different conditions. Files with index 1~80 (MEMC_1.csv ~ MEMC_80.csv) are training set, which are for you to train and validate your model, while the remaining 20 files (MEMC_81.csv ~ MEMC_100.csv) are testing set. For testing set, MEMC_81.csv~MEMC_90.csv will be public testcases for you to evaluate your model's performance before submission deadline, and MEMC_91.csv~MEMC_100.csv are hidden testcases, which are for the quality evaluation after your submission.

Each csv file contains the raw features and the corresponding IR-drop values of each cell, as shown in figure 1, and the detail description of each feature is shown in table 1.



gate_na	IR-drop	x	y	w	h	Reff	SPR	Cell type	Pleak	Cload	Pic1	Pic2	Pir	TCinput	TCoutput	TCinterna	Tarrival	Pswitch	Pinternal	Ipeak	Ttransit
	0.001912	205.96	99.82	0.76	1.4	30.7667	45.48406	12	1.73E-08	4.24E-14	9.19E-15	3.31E-14	302.2813	0	0	0	0	0	0	1.82E-08	
	0.002052	264.48	150.22	0.76	1.4	54.30255	102.9284	12	1.73E-08	4.25E-14	9.59E-15	3.29E-14	188.6325	0	0	0	0	0	0	1.82E-08	
	0.001775	222.68	144.62	0.76	1.4	11.96673	14.01109	12	1.71E-08	4.44E-14	9.96E-15	3.44E-14	215.1389	0	0	0	0	0	0	1.80E-08	
	0.001994	151.24	158.62	0.76	1.4	48.65494	110.7071	12	1.73E-08	3.95E-14	6.84E-15	3.26E-14	340.0237	0	0	0	0	0	0	1.82E-08	
	0.00197	143.45	164.22	0.76	1.4	41.90058	97.35383	12	1.71E-08	3.80E-14	7.60E-15	3.04E-14	322.2153	0	0	0	0	0	0	1.80E-08	
	0.001511	222.3	113.82	0.76	1.4	10.80568	8.748717	12	1.71E-08	4.24E-14	9.95E-15	3.24E-14	249.3608	0	0	0	0	0	0	1.80E-08	
	0.002165	252.7	69.02	0.76	1.4	49.27603	86.93278	12	1.73E-08	4.11E-14	8.74E-15	3.24E-14	126.5816	0	0	0	0	0	0	1.82E-08	

Figure 1 dataset format

Feature of target cell	Description	Input dependent?
x, y	Physical location (coordinate)	N
w, h	Width, height (dimension)	N
R_{eff}	Effective resistance	N
SPR	Shortest path resistance	N
Cell type	Cell type	N
P_{leak}	Leakage power	N
C_{load}	Loading capacitor	N
Pi_{c1}, Pi_{c2}, Pi_{r}	Equivalent π model	N
TC_{input}	Toggle counts of input	Y
TC_{output}	Toggle counts of output	Y
$TC_{internal}$	Toggle counts of internal connection	Y
$T_{arrival}$	Minimum arrival time	Y
$P_{internal}$	Internal power	Y
P_{switch}	Switching power	Y
$T_{transition}$	Transition time	Y
I_{peak}	Peak current	Y

Table 1 Raw feature description

What should you do?

In this programming assignment, we have provided a sample code for you. Figure 2 is the IR-drop prediction flow of the sample code, most of the part are already done, so you don't need to finish the entire code for IR-drop prediction. Within the sample code, we have left some TODO spaces for you to fill in, your task is to complete the sections marked with TODO and upload the trained model.

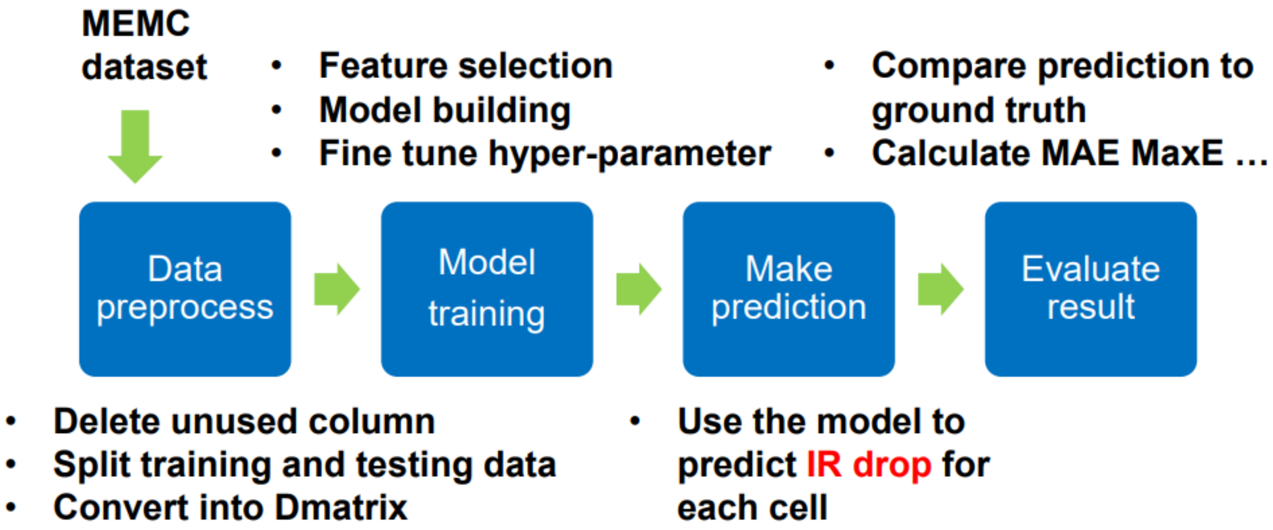


Figure 2

Below are the descriptions of each TODO in Training_Sample.py:

- TODO 1: Set training & validation dataset.
  (ex: MEMC_1.csv~MEMC_5.csv for training, MEMC_6.csv for validation)

- TODO 2: Select the features for model training. This part is the main part of PA4, feature selection is the most important step when you are doing a ML project.
  Note that "IR-drop" information should not be selected as your training features or you will get no points.

- TODO 3: Set the parameters of xgboost.

  Due to the hardware limitations of our workstation, your model can not be set to be too complicated (ex: setting *max_depth=30*) or your training process will crash with "out of memory" errors.

- TODO 4: Set the number of training iteration.

Requirement

1. We have provided the 2 sample code files for model training and evaluation. you only need to finish the TODOs.

2. We don't restrict the report format and length. In your report, you must at least include:

   (1) The selection of training and validation sets.

   (2) Describes the feature selections and parameter settings, and why you choose these

features or settings.

(3) The evaluation results (printed results of Sample_evaluation.py; MAE, MaxE, CC, NRMSE) on public test dataset, you can use whether screen shots or table to show the results.

(4) The hardness of this assignment and how you overcome it.

(5) Any suggestions about this programming assignment?

3. All files should be submitted through ee-class. The files you need to submit are as follows:

(1) The code for model training and evaluation, named *StudID*_Training.py and *StudID*_Evaluation.py
(ex: 123456789_Training.py, 123456789_Evaluation.py)

(2) Your model (IR-Drop_Predictor.dat) and the file which stores the used features (feature_name.npy)

(3) A report named *StudID_Name*_PA4_report.pdf
(ex: 123456789_陳聿廣_PA4_report.pdf)

Environment setup

For this PA, the environment information is shown below:

- Python: 3.9.0
- numpy: 1.26.3
- pandas: 2.2.0
- scikit-learn: 1.4.0
- xgboost: 1.6.2

You can use *<cmd> source /home/CAD112/PA4/env.cshrc* command to enter the python environment we build for you.

Grading

The grading is as follows:

1. Model is able to predict: 20%
2. The quality of your solution: 50% (25% public, 25% hidden)
   We will qualify your prediction result according to 4 metrics: MAE, MaxE, CC, NRMSE.

The quality score calculation is shown as below (public and hidden case will be calculated separately):

$$Quality = 5 + \sum_{4\ metrics} \left(1 - \frac{Difference\ between\ your\ result\ and\ best\ result}{Difference\ between\ best\ and\ worse\ result}\right) * 5$$

3. The report: 30%
4. Bonus: 20% (for trying other prediction methods)

Please submit your assignment on time. Otherwise, the penalty rule will apply:

- Within 24hrs delay: 20% off
- Within 48hrs delay: 40% off
- More than 48hrs: 0 point

Contact

For all questions about PA4, please send E-mail to TA 蔡書儀 (noobyves@g.ncu.edu.com)

Reference

[1] J.-X. Chen, S.-T. Liu, Y.-T. Wu et al., "Vector-based Dynamic IR-drop Prediction Using Machine Learning", 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 202-207, Jan. 2022.

[2] Z. Xie, H. Li, X. Xu, J. Hu and Y. Chen, "Fast IR Drop Estimation with Machine Learning: Invited Paper", 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), pp. 1-8, Nov. 2020.

[3] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.