The University of Texas at Austin
Chandra Department of Electrical and Computer Engineering
Cockrell School of Engineering

Spring 2023

ECE 361E: Machine Learning and Data Analytics for Edge AI

HW2 Assigned: Jan 31 DUE: Feb 12 (CST 11:59:59pm)

Work in groups of two students. At the end of the PDF file, insert a paragraph where you describe each member's contribution and two valuable things you learned from this homework.
Only one submission per group is required.

Introduction

This assignment is meant to deepen your understanding of Cyber-Physical Systems (CPS). Specifically, you will be working with an Odroid MC1 edge device. By working on this assignment, you will learn:

- How to modify the frequency of the cores to lower energy consumption, how to measure and predict thermal and power consumption;
- How to visualize and interpret various metrics and parameters to understand the impact of core frequency, power and thermals on the overall performance of edge devices;
- How to run simple benchmarks while considering both the "cyber" (i.e., performance) and the "physical" (i.e., power/thermal) components of the Odroid MC1 edge device.

Problem 1 [30p]: Cyber-Physical Systems and Benchmarks

Question 1: [12p] Connect to your designated Odroid MC1 using **Appendix A2.1**. Keep the little cluster at 0.2GHz and the big cluster at 2GHz and run ***TPBench*** only on core 4 (see **Appendix A2.2**). Draw as a function of time [s] one plot for each of the following: system (total) power consumption [W], core usage [% utilization] for each big core, and temperature [°C] for each big core.

Question 2: [3p] How many phases of benchmark execution can you identify based on temperature dynamics? A phase is a significant increase in the temperature for an extended period of time.

Question 3: [15p] Run the ***blacksholes*** and ***bodytrack*** benchmarks only on all the big cores (see **Appendix A2.2** and **A2.3**) with a frequency value of 2GHz, while keeping the little cluster at 0.2GHz. For the ***blackscholes*** benchmark set the number of threads to 4 to use all 4 big cores (see **Appendix A2.3)**. Draw a plot as a function of time [s] for each of the following: *system power* [W] and *max big temp* [°C] (*max big temp* = max {big core 4 temp, big core 5 temp, big core 6 temp, big core 7 temp}). Complete ***Table 1***:

Table 1

Benchmark	Run time [s]	Avg. power [W]	Avg. max temp [°C]	Max temp [°C]	Energy [J]
blacksholes					
bodytrack					

Problem 2 [40p]: System Power Prediction

Question 1: [20p] Use an SVM model to classify the states of the big cluster, namely "cluster active" and "cluster idle". An *active state* of the big cluster corresponds to a power consumption larger than 1W, while an *idle state* corresponds to a power consumption less than 1W. Use all the input features for classification, except the big cluster power consumption (i.e., w_big). Train the model on your computer[1] on the ***training_dataset.csv*** dataset and then test the models on ***testing_blacksholes.csv*** and

[1] We will not use GPUs for this homework.

testing_bodytrack.csv datasets. Use the thermal, power, core usage, and frequency data provided in the ***training_dataset.csv*** to train the models. Visualize (i.e., plot) the *confusion_matrix* for the two testing datasets. Compute following performance metrics: accuracy, precision, recall and F1-score. Based on all the performance metrics and the confusion matrix, explain the performance of your classifier. Complete ***Table 2:***

Table 2

Benchmarks	Accuracy	Precision	Recall	F1-Score
blackscholes				
bodytrack				

Question 2: [5p] Use a Linear regression model to predict the actual power values of the big cluster (i.e., w_big) based on the current state of the system (i.e., the provided features). Do not use any of the power features (i.e., total_watts, w_little, w_gpu and w_mem) as input features for the model. Full points will be given if you design a regressor that can obtain a test MSE value less than 0.15. Draw on the same plot the true and the predicted power values of the big cluster for each test dataset over time [s]. Complete ***Table 3:***

Table 3

Dataset	***training***	***blacksholes***	***bodytrack***
R^2			
MSE			

Question 3: [15p] Considering the dynamic power formula given in **Lecture 6**, use the term $V_{dd}^2 f$ as a feature in the training set, where f is the frequency of the big cluster and the corresponding V_{dd} is obtained from ***Table 4***. Do not use any of the power features (i.e., total_watts, w_little, w_gpu and w_mem) as input features for the model. Train the linear regression model on ***training_dataset.csv*** again and use the feature importance function to plot all feature importances and mention which are the top 3 features that contribute to the performance of the regressor. What do you observe? Explain.

Table 4

V_{dd} **[V]**	0.975	1	1.1375	1.362
f **[GHz]**	0.9	1	1.5	2

Problem 3 [30p+10Bp]: System Temperature Prediction

Question 1: [25p] Train one *MLPRegressor* for each of the big cores to predict the *temperature* values for the next time step based on features for the current time step. Do not use any of the power features (i.e., total_watts, w_big, w_little, w_gpu and w_mem) as input features for the model. Evaluate the performance of your model using the ***testing_blacksholes.csv*** and ***testing_bodytrack.csv*** test datasets. For each test dataset draw a single plot for both the true and the predicted temperature values of the big core 4 over time [s]. Complete ***Table 5***:

Table 5

Dataset	Test MSE (Core 4)	Test MSE (Core 5)	Test MSE (Core 6)	Test MSE (Core 7)
blackscholes				
bodytrack				

Question 2: [5p] What other techniques can be used to further improve the performance of your regressor? List at least two such techniques.

BONUS Question 3: [10Bp] On Odroid MC1 we implemented an on-demand governor algorithm (pseudo-code given below). ***Table 6*** shows the results obtained for both ***bodytrack*** and ***blackscholes*** benchmarks when executed with the on-demand governor active and a temperature threshold of 65°C. What are the possible “cyber-physical” trade-offs when having such a governor running? Discuss such trade-offs by comparing the runtime, average power consumption, thermal limits and energy consumption of each benchmark.

On-demand Thermal Algorithm with Proportional-Integral Gains

```
while true:
        for core in cluster:
                measure current percent usage for core
        measure maximum core temperature for big cluster
        headroom = LIM_T – max_core_temp
`       if headroom <= 0:
                reduce max_allowed to next lowest frequency
                headroom_integral = 0
        else:
                steps = floor (headroom * P + headroom_integral * I )
                increase max_allowed by steps
                headroom_integral += headroom
        if the max(usages) > USAGE_THRESHOLD:
                set cluster frequency to max_allowed
        else:
                find new minimum frequency that maintains
                        TARGET_LOAD usage based on current frequency
                        and current usage
```

Table 6

Benchmark	Runtime [s]	Avg. power [W]	Avg max temp [°C]	Max temp [°C]	Energy [J]
blackscholes	140.60	6.40	55.03	64	923
bodytrack	138.66	7.85	59.53	64	1111

Submission Instructions

Include your solutions to all the problems into a single zip file named <**Group#**>**.zip**. The zip file should contain:

1. A single PDF file containing all your results and discussions.
2. For **Problem 1** submit your **.py** files named suggestively. For **Problem 2** and **Problem 3**, **two distinct Jupyter Notebooks** (named ***p2.ipynb*** and ***p3.ipynb,*** respectively) containing the outcomes of executing your code (e.g., training and test phases, tensor shapes for training and test, features used to train your models, and training and test accuracies).
3. A *readme.txt* file describing all your items in the zip file.

Good luck!