On Predicting the Performance of GPUs by Statistical Learning Methods

Qinfeng Zhu [1]

]

Abstract

GPUs have been widely used in general purposes, but it is still not clear how to predict the speedups (comparing with CPUs) and which codes will benefit from running on GPUs. In this article, we use statistical learning technologies to overcome the difficulty and inaccuracy encountered in mechanism analysis of previous works. We will conduct investigation into two aspects: 1. Predicting the Performance Scores of GPUs in application of Deep Learning by hardware characteristics. 2. Predicting the Speedups of GPUs in applications of Deep Learning by hardware characteristics and GPU kernel features. We found that Performance Scores of GPUs significantly depend on L2 Cache and Clock Speeds, while Speedups of GPUs in application of Deep Learning significantly depend on FP Performance Input Size and Batch Size.

1. Introduction

Graphics Processing Units (GPUs) are massively parallel numeric computing processors which are designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. They are more efficient than general-purpose Central Processing Units (CPUs) for algorithms that process large quantity of data in parallel, because of their highly parallel structure. As the development of deep learning, GPUs' high parallel characteristics make people realizing the importance of GPUs in deep learning applications. It was found that while training Deep Learning neural networks, GPUs can be 250 times faster than CPUs. The explosive growth of Deep Learning in recent years has been a main reason to the emergence of general purpose GPUs. However, not all algorithms are suitable for running on GPUs. Inappropriate use will result in huge efficiency loss. So the question on how to predict the performance of a GPU given program characteristics and hardware characteristics before excution attracts more and more attention. Many authors have done very impressive works on this topic. Especially, (Resios, 2011) discussed GPU calculation time comsume model in the context of parallelizing sequential programs to GPU platforms. (Ardalani et al., 2015) tried a different way: they use Cross-Architecture Performance Prediction (XAPP), which is a machine-learning based technique that uses only single-threaded CPU implementation to predict GPU performance. But, a general model of programmings will never exist, and the modes of kernel functions running on GPU have huge diversity, which makes the mechanism analysis of GPUs difficult and inaccurate, while XAPP still need data from single-threaded CPU implementation to conduct the prediction. For those who want to know more about previous work, he or she may refer to (Boyer et al., 2013; Ardalani et al., 2015; Resios, 2011; Phillips et al., 2009; Agarwal et al., 2019) and references therein.

In our approach, we consider two efficiency metrics: 1. Performance Score 2. Speedup in application of Deep Learning:

Performance Score is defined by AI-Benchmark as the metric to the performance of CPUs and GPUs. It is generally calculated by running benchmark test. In our study, we investigated Linear Regression model to predict Performance Score without running benchmark test, and achieve a robust model with high Regression Score 0.9986.

Speedup is a traditional metric used in performance evaluation. It is defined as the ratio of CPU running time over GPU running time, it gives a dimension free metric of how fast a GPU is running on given programming. In our study, we investigated Decision Tree model to predict Speedup on Deep Learning applications, which have common pattern of excution. We also achieve high Regression Score 0.9557, but this model is sensitive to train set/test set selection.

[1] Department of Computer Science, New York University, New York, United States. Correspondence to: Qinfeng Zhu <qz981@nyu.edu>.

2. Data

Runtime performance data is collected from AI-Benchmark[1], which is a benchmark results table for different GPUs and CPUs. The data contain tasks finishing time (for trainning and inference) and a final score per GPU. The final score is defined as the weighted average of the performances scores of several test tasks. The scores of each task are computed as a geometric mean of the test results belonging to this category. It can be formulated as

$$\text{Score}_{task} = \sqrt[n]{\text{Result}_1 \times \text{Result}_2 \times \cdots \times \text{Result}_n}$$

where each result is given by

$$\text{Result}_i = 10000 \times \frac{\text{ConsumingTime}}{\text{ReferenceTime}},$$

The Reference Time, which is a constant, is the mean value of all the test results in AI-Benchmark databases. The weights for test tasks are given by

- 50 % - float-16 tests;
- 30 % - int-8 tests;
- 4 % - float-32 tests;
- 3% - parallel execution of the models;
- 2% - initialization time, float models;
- 1% - initialization time, quantized models;

For those who are interested in the technologies in AI-Benchmark, he or she may refer to (Reddi et al., 2019) for more details. What is more, we also collect the hardware parameters data of all the GPUs considered in our research.

All data are standardized to remove the influence of dimension before use. In speedup prediction, we use Intel Core i7-6700K as the reference CPU to calculate GPU speeedups. We claim the selection of reference CPU will not influence the prediction result because changing CPU will only incur a constant multipling speedups for each deep learning network, and will be immediately divided out in the standardization.

3. Methods and Experience Results

In this section, we will have a deep discussion of two statistical learning models we used: 1. Linear Regression for performance scores prediction 2. Decision Tree Regression for speedups prediction, and their numerical experience results. Generally speaking, Regression Model is given by

$$y = f_\alpha(x) + \varepsilon$$

where vector x is called regression variables vector, y is called dependent variable, the parameters vector α is estimated by training the model, and noise term ε obeying Gaussian distribution and has zero means. Then we apply the regression model on test data to evaluate whether the trained model is good or not. We used scikit-learn 0.22 package to do two regressions.

First, we are at the position to introduce the evaluating matric and the features selected for analysis.

3.1. Evaluating Metric

We randomly seperate the data into training set and test set, use training set to train the regression model and evaluate the model on test set. We use Regression Score as the metric to evaluate the model, which is

$$\text{RegressionScore} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where n is the cardinal number of test set, and i indexs over all element of test set, $\bar{y}$ is the average of all y_i. The maximum of Regression Score is 1, that means the regression model is absolutely correct on test data. The Regression Score can be negative, which means really bad model.

3.2. Feature Selection

Our strategy is to select possible relevant features as the regression variable, then do regression and find out which variables have significant influence. For hardware parameters, we select Number of Cores, Bandwidth, L1 Cache, L2 Cache, (Core Base) Clock Speeds, FP Performance. For kernel parameters, we select Batch Size, Input Size, and the Time consuming of each GPU on each deep learning task.

3.3. Performance Scores Prediction

3.3.1. Method

Linear regression is a linear model that assumes the dependent variable has linear relations with regression variables, that is

$$y = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n + \beta + \varepsilon$$

with regression variable $x_1, x_2, \cdots, x_n$, parameters vector $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ and intercept β.

We use Linear Regression model to predict the relation between performance scores and hardware parameters.

[1] http://ai-benchmark.com/ranking_cpus_and_gpus_detailed.html

3.3.2. Result

The parameters of regression result is given by the table 1, with Intercept -0.36 and Regression Score 0.9986.

Table 1. The Coefficients of Linear Regression Model

Number of Cores	Bandwidth	L1 Cache
0.17647233	0.0732235	0.12524208
L2 Cache	Clock Speeds	FP Performance
0.30761558	0.49641278	0.12034286

The Regression Score is almost 1 means our regression model is successful. As can be seen, L2 Cache and Clock Speeds have significant impact on Performance Scores, while L1 Cache, FP Performance have minor significant impact on Performance Scores. Bandwith has very tiny influence on Performance Scores. Also, by running code for many time, we see our result is robust to the train set/test set selection.

3.4. Speedups Prediction

3.4.1. Method

In speedups prediction, we use both hardware parameters and kernel parameters to get a model of GPU speedups without excuting real programming. However, the Linear Regression model generates bad result in speedups predictation, indicating that the data contain sophisticated non-linear relations.

We use Decision Tree Regression to handle this question. Decision Tree Regression conducts regression in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while an assiociated decision tree is constructed at the same time. A decision node has two or more branches, each representing values for the attribute tested. Leaf node represents a decision on the numerical target.

3.4.2. Result

The constructed Decision Tree (part) is show in figure 1.

Gini importance is defined as the total decrease in node impurity (weighted by the probability of reaching that node (which is approximated by the proportion of samples reaching that node)) averaged over all trees of the ensemble.

The regression score is 0.9557, which indicates the regression is successful as well. But the result of regression depends on train set selection, bad selection can really generates bad result.

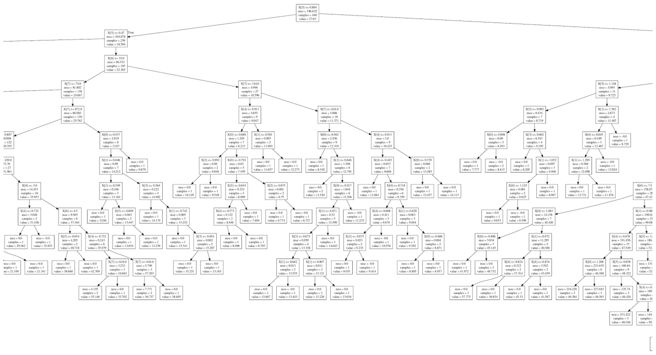

Figure 1. Decision Tree (part)

Table 2. The Gini Importance of Decision Tree Regression

Number of Cores	Bandwidth	L1 Cache
0.08592539	0.02136347	0.00132767
L2 Cache	Clock Speeds	FP Performance
0.00632147	0.011819	0.36178935
Batch Size	Input Size	
0.28402347	0.22743017	

As can be seen, the FP Performance Input Size and Batch Size have more significant influence on the speedups. high FP performance, Large Input Size and Batch Size will result in high speedups.

4. Conclusion and Perspectives

This article can be seen as a step to a comprehensive study of performance predicting of GPUs by statistical learning methods. Also, our result indicate the influence of hardware and kernel parameters on the performance of deep learning network, and thus give hardware manufacturer more insight about developing next generation of GPUs. However, there is still some drawback in our model. Especially, the result of Decision Tree Regression lacks of robustness, and thus need further investigation.

References

Agarwal, N., Jain, T., and Zahran, M. Performance prediction for multi-threaded applications. In International Workshop on AI-assisted Design for Architecture (AIDArc), held in conjunction with ISCA, 2019.

Ardalani, N., Lestourgeon, C., Sankaralingam, K., and Zhu, X. Cross-architecture performance prediction (xapp) using cpu code to predict gpu performance. In Proceedings of the 48th International Symposium on Microarchitecture, pp. 725–737. ACM, 2015.

Boyer, M., Meng, J., and Kumaran, K. Improving gpu performance prediction with data transfer modeling. In 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum, pp. 1097–1106. IEEE, 2013.

Phillips, E., Zhang, Y., Davis, R., and Owens, J. Rapid aerodynamic performance prediction on a cluster of graphics processing units. In 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, pp. 565, 2009.

Reddi, V. J., Cheng, C., Kanter, D., Mattson, P., Schmuelling, G., Wu, C.-J., Anderson, B., Breughe, M., Charlebois, M., Chou, W., et al. Mlperf inference benchmark. arXiv preprint arXiv:1911.02549, 2019.

Resios, A. Gpu performance prediction using parametrized models, 2011.