PRINCIPAL COMPONENT ANALYSIS HARDWARE ACCELERATION

NG YEE WEI

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Computer and Microelectronic Systems)

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

JAN 2020

DEDICATION

This thesis is dedicated to my parents, who taught me never to give up and always strive to be the best. It is also dedicated to my siblings, who provided me with moral support throughout the entire length of the project.

ACKNOWLEDGEMENT

Throughout project execution, I had received a lot of assistance and support from people surrounding me in various aspects. Therefore, I would like to take this opportunity to express my deepest appreciation to those people. I would like to show the greatest gratitude to my project supervisor, Associate Professor Dr Muhammad Nadzir bin Marsono who had provided clear guidance to me throughout the implementation of this project. Next, I wish to thank my course-mates who were willing to share beneficial knowledge with me to develop the project. They showed no hesitation to help me whenever I faced some technical problems in doing this project. Their sincere comment on the project has resulted in the improvement of this project.

ABSTRACT

Since machine learning is getting more attention in various applications, the performance of those applications has become the main concern of its users. To perform machine learning, one of the vital processes is feature extraction which is to reduce the raw data dimension that aims to get rid of noise and speed up further data analysis. Principal Component Analysis (PCA) is one of dimension reduction techniques that often used with other complex feature extraction techniques. However, PCA involves heavy computation and plays an important role to determine the speed performance of the application. This project is to propose PCA hardware acceleration to enhance its performance. From software profiling, the most intensive function in the PCA algorithm is the computation of eigenvalues and eigenvectors (eigensolver). Thus, this project has developed an eigensolver hardware accelerator by applying parallel execution through unrolling, pipelining and scheduling techniques in order to improve the performance of PCA. The proposed eigensolver is developed using Vivado HLS 2019.2. The performance of the proposed accelerator is evaluated by comparing it with conventional PCA hardware. The proposed eigensolver hardware accelerator has achieved a speedup of 6.27 compared with its conventional implementation.

ABSTRAK

Memandangkan pembelajaran mesin semakin mendapat perhatian dalam pelbagai aplikasi, prestasi aplikasi tersebut telah menjadi fokus utama pengguna. Untuk melaksanakan pembelajaran mesin, salah satu proses penting ialah pengekstrakan ciri. Pengekstrakan ciri bertujuan untuk mengurangkan dimensi data asal dengan menghilangkan signal bising dan mempercepat analisis data selanjutnya. Analisis Komponen Utama (PCA) ialah salah satu teknik pengurangan dimensi yang sering digunakan dengan teknik pengekstrakan ciri kompleks yang lain. Walau bagaimanapun, PCA melibatkan banyak pengiraan dan memainkan peranan penting untuk menentukan prestasi kelajuan aplikasi. Projek ini telah mengusulkan pemecut perkakasan PCA untuk meningkatkan prestasinya. Melalui analisis secara perisian, fungsi yang paling intensif dalam algoritma PCA ialah pengiraan nilai eigen dan vektor eigen. Oleh itu, projek ini telah membangunkan alat pemecut perkakasan eigensolver dengan kaedah QR dengan menggunakan pelaksanaan selari melalui teknik pembukaan, *pipelining* dan penjadualan untuk meningkatkan prestasi PCA. Alat pemecut perkakasan eigensolver yang dicadangkan telah dibangunkan dengan mengguna Vivado HLS 2019.2. Prestasi pemecut yang dicadangkan dalam projek ini dinilai dengan membandingkannya dengan perkakasan PCA konvensional. Pemecut perkakasan eigensolver yang dicadangkan dalam projej ini telah mencapai kelajuan 6.27 kali berbanding dengan pelaksanaan konvensionalnya.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
ALAP	-	As Late As Possible
ASAP	-	As Soon As Possible
ASIP	-	Application Specific Instruction Set Processor
BRAM	-	Block Random-Access Memory
CPU	-	Central Processing Unit
DFG	-	Data Flow Graph
FIFO	-	First In First Out
FPGA	-	Field-Programmable Gate Array
GPU	-	Graphic Processing Unit
IP	-	Intellectual Property
ICA	-	Independent Component Analysis
KPCA	-	Kernel Principal Component Analysis
MAC	-	Multiplication and Accumulation
MGS	-	Modified Gram-Schmidt
MPCA	-	Multilinear Principal Component Analysis
NoC	-	Network on Chip
PCA	-	Principal Component Analysis
QRD	-	QR Decomposition
RAM	-	Random-Access Memory
RPCA	-	Robust Principal Component Analysis
SIFT	-	Scale-Invarient Feature Transform
VPD	-	Vector Dot Product

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Research Background

In the advancement of technology, artificial intelligence (AI) [1] is widely applied in various fields such as security system, transportation, agriculture, and the list is endless. Machine learning is an implementation of AI that enables a system to automatically learn and enhance from experience without human assistance. This is done by detecting a specific pattern in data observed by the systems and make decision based on experience. Data prepossessing is required to carried out on the original dataset to extract its important feature by removing noise in the dataset [2].

Principal Component Analysis (PCA) [3] is one of the simplest and oldest techniques that is used for feature extraction. It is a multivariate statistic method to transforms each variable in feature space to a linear combination of the original input variable and keeps crucial information of a dataset by analyzing the correlation of each element in the dataset. PCA is commonly applied in health monitoring system [4], network intrusion system [5] and gas leakage detection system [6] that require real-time effect. Hence, the processing speed of PCA requires to be as fast as possible to minimize delay in the application.

Generally, the PCA algorithm has four main functions which are mean, covariance, eigenvalues and eigenvectors, and K-principal component computation. These functions are carried on a matrix that contains variables of every observation collected by an application. The computation requires intensive matrix operations especially matrix multiplication. The amount of operations is highly dependent on the data size. Due to the involvement of division and square root operation, data precision is important for PCA algorithm to ensure the reliability of the algorithm. Besides, it also require plenty of data comparison due to the computation of eigenvalues and eigenvectors contains a heuristic algorithm.

Due to involving big O notation and heuristic algorithm, PCA execution is complex and time-consuming. Hence, multiple platforms are proposed for high-performance PCA execution. For high-end applications, the PCA algorithm is running on a more robust processor such as GPU and multicore processor instead of the conventional CPU to shorten the execution time [7]. On the other hand, in the lower-end device that has constraints in terms of cost and power consumption, a specific hardware accelerator is designed for PCA execution. ASIC and FPGA are examples of platforms to be used for this implementation. Both of them support parallelism which could speed up the speed of PCA execution, but FPGA is more flexible than ASIC due to its re-configurable nature. A comparison of different processors is summarized in Table 1.1.

Table 1.1: Comparison on CPU, Multicore, GPU and FPGA

Property	CPU	Multicore	GPU	FPGA
Performance	Low	High	Very High	High
Power	Medium	High	High	Low
Cost	Medium	High	High	Medium

1.2 Problem Statement

PCA is often used for feature extraction to pre-process data in an application and it is a time-consuming process. While the embedded system is dominating the market, the design of a hardware accelerator is gaining more attention. By designing a PCA hardware accelerator, the performance of the application could increase. However, there is some challenge to implement PCA as a hardware accelerator.

PCA is an algorithm with heavy computation whereby its processing time is highly dependent on data size. To speed up this process, many proposed works have focused on higher-level optimization such as reduce memory access latency from the system level. There is a lack of optimization work is done in operation in the lowest level operation unit in the PCA algorithm.

Parallel computation has been commonly used to design a hardware accelerator by adding additional resources. In [8], a comparison is made between hardware and software-based implementation of PCA. The result has shown that the speed of hardware implementation is five times of software implementation. However, the limitation is fully unrolled a loop will cause the performance of PCA is no longer in a linear relationship with the number of resources. A large increase in resources only causes little improvement in processing time.

Another method to boost up performance is through pipelining for throughput improvement. PCA is a dimensional reduction method by finding the correlation between vectors in a dataset. Thus, there is some dependency between certain steps in eigenvalue computation in the PCA algorithm [9]. Work in [10] has listed out resources needed for every step in the algorithm, grouped the resource to several pipeline stages and applied pipelined processing. However, this method is designed manually and requires a large effort to make sure there is no hazard or delay in the pipelining process [11, 12].

Furthermore, since PCA has four main functions, acceleration work is applied to different stages of PCA by researchers with different methods and purposes. This raises the research question of what is the most effective part of PCA to be hardware accelerated and how to efficiently apply parallel computing in the PCA algorithm?

1.3 Objectives

To solve the problem mentioned in section 1.2, there are some tasks needed to be done for delivering better work. The objectives targeted for this project are stated as follow:

1. To perform software profiling to investigate time-consuming PCA internal functions
2. To propose hardware accelerator architecture for improved PCA performance

3. To evaluate the performance-area trade-off of proposed PCA hardware accelerator against the conventional PCA implementation

1.4 Project Scope

PCA hardware accelerator could be designed through different approaches such as system-level architecture, the connection between the module of PCA's main functions and acceleration of kernel inside the module. However, in this project, the effort is only focused on the acceleration through PCA's main function module embedded design and internal operation. Since PCA involves a few functions, this project only focuses on the most intensive function in PCA based on the result of the investigation stated in the first objective. Besides, work in this project mainly aims for enhancing the processing speed of PCA without guarantee its area and power consumption. Performance evaluation of the proposed PCA hardware accelerator is done through simulation using Vivado, no hardware component is involved.

1.5 Project Report Outline

This report outlines the hardware acceleration of the PCA algorithm. There is a total of five chapters in this report with the following organization.

Chapter 2 presents literature review for this project. This chapter includes introduction and background of PCA and brief explanation of its algorithm. This chapter also discuss some related previous work on methods to design PCA hardware accelerator.

Chapter 3 describes the research methodology of project implementation in order to achieve objectives targeted. Proposed methods to hardware accelerate PCA execution in this project is explained in this chapter.

Chapter 4 presents validation work and results to prove the proposed methods with analysis. This includes software profiling of PCA algorithm and performance evaluation of hardware accelerator developed using the proposed methods in Chapter 3. A comparison between proposed work and previous works is also presented here.

The last chapter summarizes the work that is done for this project and future work.

REFERENCES

1. Lu, H., Li, Y., Chen, M., Kim, H. and Serikawa, S. Brain Intelligence: Go beyond Artificial Intelligence. *Mobile Networks and Applications*, 2018. 23(2): 368–375. doi:10.1007/s11036-017-0932-8. URL `https://doi.org/10.1007/s11036-017-0932-8`.

2. Moravec, H. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. 1980. (CMU-RI-TR-80-03).

3. Pearson, K. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 1901. 2: 559–572.

4. Tibaduiza Burgos, D., Mujica, L. and Rodellar, J. Structural Health Monitoring based on principal component analysis: damage detection, localization and classification. 2011: 8–17.

5. Das, A., Misra, S., Joshi, S., Zambreno, J., Memik, G. and Choudhary, A. An Efficient FPGA Implementation of Principle Component Analysis based Network Intrusion Detection System. *2008 Design, Automation and Test in Europe*. 2008. ISSN 1530-1591. 1160–1165. doi:10.1109/DATE.2008.4484835.

6. Wang, L., Gao, X. and Liu, T. Gas pipeline small leakage feature extraction based on LMD envelope spectrum entropy and PCA. *Transactions of the Institute of Measurement and Control*, 2016. 38(12): 1460–1470. doi:10.1177/0142331215599248. URL `https://doi.org/10.1177/0142331215599248`.

7. Martel, E., Lazcano, R., Lopez, J., Madronal, D., Salvador, R., Lopez, S., Juarez, E., Guerra Hernandez, R., Sanz, C. and Sarmiento, R. Implementation of the Principal Component Analysis onto High-Performance Computer Facilities for Hyperspectral Dimensionality Reduction: Results and Comparisons. *Remote Sensing*, 2018. 10: 864. doi:10.3390/rs10060864.

8. Perera, D. G. and Li, K. F. Embedded hardware solution for principal component analysis. *Proceedings of 2011 IEEE Pacific Rim Conference on*

Communications, Computers and Signal Processing. 2011. ISSN 1555-5798. 730–735. doi:10.1109/PACRIM.2011.6032984.

9. Parker, M., Mauer, V. and Pritsker, D. QR decomposition using FPGAs. *2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS)*. 2016. ISSN 2379-2027. 416–421. doi: 10.1109/NAECON.2016.7856841.

10. Shahrouzi, S. N. and Perera, D. G. Optimized hardware accelerators for data mining applications on embedded platforms: Case study principal component analysis. *Microprocessors and Microsystems*, 2019. 65: 79 – 96. ISSN 0141-9331. doi:https://doi.org/10.1016/j.micpro.2019.01.001. URL `http://www.sciencedirect.com/science/article/pii/S0141933118302394`.

11. Shin, D. and Park, J. A Low-Latency and Area-Efficient GramâSchmidt-Based QRD Architecture for MIMO Receiver. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2018. 65(8): 2606–2616. ISSN 1558-0806. doi: 10.1109/TCSI.2018.2795342.

12. Lee, H., Kim, H., Cho, M. and Kim, J. Low-latency implementation of CORDIC-based sorted QR decomposition for high-speed MIMO-OFDM system. *2018 28th International Conference Radioelektronika (RADIOELEKTRONIKA)*. 2018. ISSN null. 1–4. doi:10.1109/RADIOELEK.2018.8376356.

13. Parivesh and Sharma, R. K. A time efficient architecture implementation of PCA for ICA. *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*. 2017, vol. 1. 721–725. doi:10.1109/ICECA.2017.8203637.

14. Yang, X., Jiang, L., Tang, X. and Ren, X. An improved PCA-SIFT algorithm application in light small UAV image registration. *2017 Progress In Electromagnetics Research Symposium - Spring (PIERS)*. 2017. 2554–2558. doi:10.1109/PIERS.2017.8262182.

15. FenzÃ¡ndez, D., GonzÃ¡lez, C. and Mozos, D. Dimensionality reduction of hyperspectral images using reconfigurable hardware. *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*. 2016. ISSN 1946-1488. 1–2. doi:10.1109/FPL.2016.7577394.

16. Fernandez, D., Gonzalez, C., Mozos, D. and Lopez, S. FPGA implementation of the principal component analysis algorithm for dimensionality reduction of hyperspectral images. *Journal of Real-Time Image Processing*, 2019. 16(5): 1395–1406. ISSN 1861-8219. doi:10.1007/s11554-016-0650-7. URL `https://doi.org/10.1007/s11554-016-0650-7`.

17. Gosavi, A. P. and Khot, S. R. Emotion recognition using Principal Component Analysis with Singular Value Decomposition. *2014 International Conference on Electronics and Communication Systems (ICECS)*. 2014. ISSN null. 1–5. doi:10.1109/ECS.2014.6892683.

18. Sokolovskiy, A. V., Tyapkin, V. N., Veisov, E. A. and Fateev, Y. L. The Pipelined QR Decomposition Hardware Architecture Based On Givens Rotation CORDIC Algorithm. *2019 International Siberian Conference on Control and Communications (SIBCON)*. 2019. ISSN 2380-6508. 1–4. doi: 10.1109/SIBCON.2019.8729615.

19. Shahrouzi, S. N. and Perera, D. G. Dynamic partial reconfigurable hardware architecture for principal component analysis on mobile and embedded devices. *EURASIP Journal on Embedded Systems*, 2017. 2017(1): 25. ISSN 1687-3963. doi:10.1186/s13639-017-0074-x. URL `https://doi.org/10.1186/s13639-017-0074-x`.

20. Mansoori, M. A. and Casu, M. R. Efficient FPGA Implementation of PCA Algorithm for Large Data using High Level Synthesis. *2019 15th Conference on Ph.D Research in Microelectronics and Electronics (PRIME)*. 2019. ISSN null. 65–68. doi:10.1109/PRIME.2019.8787782.

21. Ali, A. A. S., Amira, A., Bensaali, F. and Benammar, M. Hardware PCA for gas identification systems using high level synthesis on the Zynq SoC. *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*. 2013. ISSN null. 707–710. doi:10.1109/ICECS.2013.6815512.

22. Perera, D. G. and Li, K. F. FPGA-Based Reconfigurable Hardware for Compute Intensive Data Mining Applications. *2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. 2011. 100–108. doi:10.1109/3PGCIC.2011.25.

23. Zhou, T., Guo, S., Lei, Y. and Dou, Y. Area-efficient high-throughput sorted QR decomposition-based MIMO detector on FPGA. *2015 IEEE International Conference on Computer and Communications (ICCC)*. 2015. ISSN null. 394–398. doi:10.1109/CompComm.2015.7387603.

24. Karnthak, T. and Kumhom, P. A hardware implementation of PCA based-on the Networks-on-Chip paradigm. *2012 International Symposium on Communications and Information Technologies (ISCIT)*. 2012. ISSN null. 834–839. doi:10.1109/ISCIT.2012.6381018.

25. Liu, T., Ko, Y., Chiu, Y., Lin, W. and Chu, Y. Hardware Implementation of the Preprocessing QR-Decomposition for the Soft-Output MIMO Detection With Multiple Tree Traversals. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2018. 65(2): 186–190. ISSN 1558-3791. doi:10.1109/TCSII.2017.2703818.

26. Schellhorn, M. and Notni, G. Optimization of a Principal Component Analysis Implementation on Field-Programmable Gate Arrays (FPGA) for Analysis of Spectral Images. *2018 Digital Image Computing: Techniques and Applications (DICTA)*. 2018. ISSN null. 1–6. doi:10.1109/DICTA.2018.8615866.

27. Alhamed, A. and Alshebeili, S. FPGA implementation of complex-valued QR decomposition. *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*. 2016. ISSN 2159-2055. 1–4. doi:10.1109/ICEDSA.2016.7818557.

28. Guerrero-RamÃrez, J. E., Velasco-Medina, J. and Arce-Clavijo, J. C. Hardware design of an eigensolver based on the QR method. *2013 IEEE 4th Latin American Symposium on Circuits and Systems (LASCAS)*. 2013. ISSN null. 1–4. doi:10.1109/LASCAS.2013.6519065.

29. Korat, U. A. and Alimohammad, A. A Reconfigurable Hardware Architecture for Principal Component Analysis. *Circuits, Systems, and Signal Processing*, 2019. 38(5): 2097–2113. ISSN 1531-5878. doi:10.1007/s00034-018-0953-y. URL `https://doi.org/10.1007/s00034-018-0953-y`.

30. Thethi, S. K. and Kumar, R. Area and power efficient register allocation technique for the implementation of PCA. *2017 4th International Conference*

on Signal Processing, Computing and Control (ISPCC). 2017. ISSN null. 251–257. doi:10.1109/ISPCC.2017.8269684.