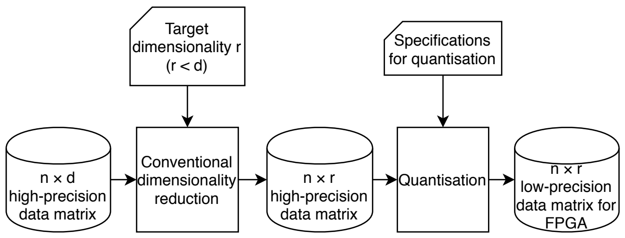
Quantisation-aware Dimensionality Reduction

Ce Guo and Wayne Luk
Department of Computing, Imperial College London, UK
{c.guo, w.luk}@imperial.ac.uk

***Abstract*—Typical data analysis systems involving FPGAs work better with low-dimensional low-precision (LDLP) data than with high-dimensional high-precision (HDHP) ones due to limitations on data bandwidth and on-chip resources. However, data sources usually offer HDHP data matrices. It is possible to obtain LDLP data that approximate HDHP data by applying dimensionality reduction and quantisation. However, this straightforward workflow incurs significant information loss that reduces the accuracy of data analysis on FPGAs. This paper proposes that the information loss in the straightforward workflow is due to the dimensionality reduction algorithm's unawareness of quantisation. To address this problem, the paper introduces Quadir, a novel algorithm that supports quantisation-aware dimensionality reduction. In particular, Quadir is an alternating direction optimisation algorithm that finds a transformation to project an HDHP data matrix to an LDHP one so that the LDLP matrix after quantisation can reconstruct the original HDHP matrix with optimised approximation. Our experimental evaluation in data reconstruction shows that the LDHP data matrices produced by Quadir preserve more information from the original data than principal component analysis (PCA) after quantisation.**


Target dimensionality r (r < d)
Specifications for quantisation
n × d high-precision data matrix
Conventional dimensionality reduction
n × r high-precision data matrix
Quantisation
n × r low-precision data matrix for FPGA
(a) Straightforward: no awareness of quantisation

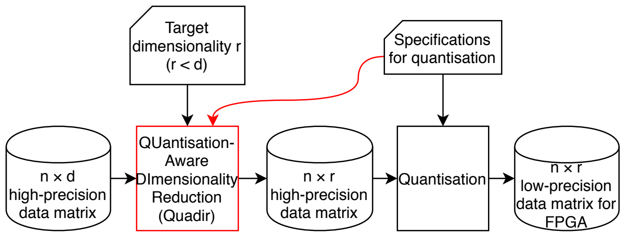

Target dimensionality r (r < d)
Specifications for quantisation
n × d high-precision data matrix
QUantisation-Aware DImensionality Reduction (Quadir)
n × r high-precision data matrix
Quantisation
n × r low-precision data matrix for FPGA
(b) Proposed: awareness of quantisation

Fig. 1. Workflows with DR and quantisation

I. Introduction

Data analysis systems on the FPGA platform usually work better with low-dimensional low-precision (LDLP) data than with high-dimensional high-precision (HDHP) ones. Compared with HDHP data, LDLP data can facilitate FPGA design by reducing bandwidth and resource usage. Two conventional techniques to produce LDLP data are dimensionality reduction (DR) and quantisation. A dimensionality reduction procedure reduces the number of elements in each data point, shrinking the data from the perspective of dimensionality without touching the numerical representation; a quantisation procedure reduces the precision of data elements, cutting down the data size from the perspective of numeric precision without affecting the dimensionality. A straightforward workflow to prepare LDLP data is to apply dimensionality reduction followed by quantisation [1]. However, this workflow can incur information loss and reduce the accuracy of the downstream data analysis system on the FPGA platform.

This paper aims to reduce such information loss by improving the fundamental understanding of the effect of quantisation on dimensionality reduction. A key insight is that the ineffective interaction between dimensionality reduction and quantisation is due to the unawareness of the dimensionality reduction procedure regarding the downstream quantisation procedure. Specifically, with the straightforward workflow shown in Figure 1a, it is impossible to provide the specifications for quantisation to the dimensionality reduction procedure. Therefore, the dimensionality reduction procedure is unable to take advantage of the specifications to reduce information loss during quantisation.

This paper proposes Quadir, the first dimensionality reduction algorithm with the awareness of quantisation, to enable the use of the workflow in Figure 1b. Specifically, Quadir projects an HDHP data matrix to an LDHP one with the awareness of quantisation, so that the quantisation procedure incurs low information loss when it converts the LDHP data matrix to an LDLP one. The data analysis system on the FPGA platform can directly use the LDLP data with low bandwidth and resource usage. The major contributions of this paper include the following:

1) An analysis of the suboptimality of conventional dimensionality reduction methods with quantisation. See Section III-A.
2) Quadir, a novel dimensionality reduction algorithm that takes quantisation into account when projecting data to the low-dimensional space. See Section III-B.
3) An experimental evaluation of Quadir on data reconstruction. See Section IV.

II. Background and Related Work

The primary objective of dimensionality reduction is to reduce the data size by projecting the data onto a low-dimensional space, preserving as much information as possible [2]. In other words, dimensionality reduction methods compress the data, allowing direct processing without decompression. For ease of discussion, we denote the set of high-precision numbers and low-precision numbers respectively by $\mathbb{H}$ and $\mathbb{L}$. Without loss of generality, we assume that the data source provides high-precision data matrices in the row-major order. In other words, a data matrix $X \in \mathbb{H}^{n\times d}$ contains n data points. Each data point is a d-dimensional row vector with d real-valued elements in $\mathbb{H}$.

A dimensionality reduction algorithm transforms a data matrix $X \in \mathbb{H}^{n\times d}$ to a low-dimensional data matrix $X \in \mathbb{H}^{n\times r}$ where $r < d$. An obvious benefit of dimensionality reduction for FPGA design is the reduction in storage space, data traffic and logic resource usage without significant accuracy loss. The low-dimensional data matrix obtained from dimensionality reduction can be treated as an approximate version of the original one. If the low-dimensional data provide sufficient accuracy, it is safe to store and transmit the low-dimensional projection while discarding the original one. Moreover, FPGA designs for a data analysis task may have an upper bound on the number of dimensions due to bandwidth or resource limit. When the number of dimensions exceeds the capacity of the hardware design, dimensionality reduction gives a chance to enable on-chip data processing with decent accuracy.

Existing FPGA-related studies on data projection are mainly on hardware designs that extract projection models from data [3], [4], [5]. There are also FPGA designs projecting data into random sub-spaces to enhance ensemble modelling [6]. These studies demonstrate that FPGAs can benefit data projection techniques. In contrast, we investigate a problem in the opposite direction where data projection techniques benefit FPGA design. In particular, we propose an off-chip data projection method to reduce the resource and bandwidth usage for on-chip data analysis.

Existing studies related to both quantisation and hardware design are mostly about the quantisation of model parameters to facilitate hardware design. For instance, studies such as [7] focus on the quantisation of model parameters to reduce the model size. The objective of these studies is to deploy the model using limited on-chip resources. In contrast, this study focuses on the preparation of quantised data for data analysis systems.

III. Quantisation-aware dimensionality reduction

This section proposes the QUantisation-Aware DImensionality Reduction (Quadir) algorithm. Given a high-dimensional data matrix $X \in \mathbb{H}^{n\times d}$, the remaining dimensionality r and the quantisation function Ξ, Quadir finds a function Φ that projects X to a low-dimensional data matrix $Y = \Phi(X) \in \mathbb{H}^{n\times r}$ so that $\Xi(Y) \in \mathbb{L}^{n\times r}$ contains sufficient information to approximately restore the original data. The data analysis algorithm running on the FPGA platform can use $\Xi(Y)$ directly instead of X, with reduced bandwidth and resource usage. Section III-A analyses why the consecutive application of dimensionality reduction and quantisation results in suboptimal statistical accuracy in general; Section III-B proposes the Quadir algorithm to enable quantisation-aware dimensionality reduction.

A. Suboptimality of straightforward workflow

We expect that the low-dimensional data transmitted to the FPGA platform still contain sufficient information from the original data. In other words, we aim to solve the following optimisation problem

$$\Phi^\bullet, \Psi^\bullet = \arg\min\nolimits_{\Phi,\Psi} F(\Psi(\underbrace{\Phi(X)}_{\text{high-precision}}), X) \tag{1}$$

where F is a function that measures the difference between two matrices regarding the properties to preserve; Φ is a dimensionality reduction function that projects the data from $\mathbb{H}^{n\times d}$ to $\mathbb{H}^{n\times r}$; Ψ is a data reconstruction function that projects the data from $\mathbb{H}^{n\times r}$ or $\mathbb{L}^{n\times r}$ back to $\mathbb{H}^{n\times d}$. Since Φ and Ψ are functions, the optimisation problem in Equation 1 involves a search in the function space. However, it is impossible to search in the function space without additional assumptions. A practical approach is to make the two functions parametric and search in the parameter space.

When the two functions, $\Phi^\bullet$ and $\Psi^\bullet$, are available, the data analysis algorithm running on the FPGA platform can process the data in the low-dimensional space $\Phi^\bullet(X) \in \mathbb{H}^{n\times r}$ directly without knowing the original data $X \in \mathbb{H}^{n\times d}$.

Since the FPGA platform often has limited bandwidth for incoming data, we further reduce the data size by quantising the low-dimensional data matrix $\Phi^\bullet(X)$ to the low-precision space $\mathbb{L}^{n\times r}$. In this case, it is only necessary to transmit the quantised low-dimensional data $\Xi(\Phi^\bullet(X))$ to the FPGA platform. An ideal situation is that the data matrix $\Xi(\Phi^\bullet(X))$ received by the FPGA platform retains as much information from the original data matrix X. In other words, it is expected that the following variant of Equation 1 holds:

$$\Phi^\bullet, \Psi^\bullet = \arg\min\nolimits_{\Phi,\Psi} F(\Psi(\underbrace{\Xi(\Phi(X))}_{\text{quantised for FPGA}}), X) \tag{2}$$

In Equation 2, the low-precision data matrix $\Xi(\Phi(X))$ for FPGA processing appears in the position of its high-precision counterpart $\Phi(X)$ in Equation 1.

However, Equation 2 does not hold because the optimisation objective functions in Equation 1 and Equation 2 are different. In other words, the functions $\Phi^\bullet$ and $\Psi^\bullet$ obtained from Equation 1 may not satisfy Equation 2. As a result, the quantised data matrix $\Xi(\Phi^\bullet(X))$ for FPGA processing is likely to incur a higher error than $\Phi^\bullet(X)$ due to the information loss introduced in quantisation.

B. Quadir algorithm

The key insight of the Quadir algorithm is to reduce the quantisation error by considering the existence of quantisation during dimensionality reduction. In other words, rather than minimising the information loss in dimensionality reduction, we directly minimise the information loss for the data transmitted to the FPGA platform. In particular, we directly solve the linear case for the optimisation problem in Equation 2:

$$B^\circ, C^\circ = \arg\min_{B,C} ||X - \Xi(XB)C||_2 \tag{3}$$

The optimisation problem is challenging because the optimisation objective is not differentiable with respect to B. We propose an alternative form to facilitate optimisation:

$$B^\circ, C^\circ = \arg\min_{B,C} G_{B,C}(X) \tag{4}$$

where

$$\begin{aligned} G_{B,C}(X) =& ||X - UC||_2 \\ &+ \lambda ||U - \Xi(V)||_2 + \rho ||V - XB||_2 \end{aligned} \tag{5}$$

where $U \in \mathbb{L}^{n\times r}$ and $V \in \mathbb{H}^{n\times r}$ are latent variables; λ and ρ are positive penalty coefficients. With these settings, the optimisation problem in Equation 4 becomes a transformed version of the original problem in Equation 3 where the optimisation objective $G_{B,C}(X)$ approximately enforces the constraints on U and V.

We design Quadir, an alternating direction algorithm [8], to solve the optimisation problem. In the optimisation problem defined in Equation 4, when three out of the four variables B, U, V and C are available, there is a polynomial-time procedure to update the remaining one. The algorithm keeps updating the four variables using the following rules:

$$U = (XC^T + \lambda \Xi(V))(CC^T + \lambda I)^{-1} \tag{6}$$

$$V = \frac{1}{\lambda + \rho}(\lambda U + \rho XB) \tag{7}$$

$$B = X^\dagger V \tag{8}$$

$$C = U^\dagger X \tag{9}$$

where $A^\dagger$ is the Moore-Penrose inverse [9] of A and $I \in \mathbb{R}^{r\times r}$ is an identity matrix. The algorithm terminates when the error $G_{B,C}(X)$ stops decreasing. It is only necessary to run the Quadir algorithm once for a data matrix X to produce B°. It is then possible to project data with a similar distribution to X without rerunning Quadir.

IV. Evaluation

We compare Quadir with the most widely used dimensionality reduction method, principal component analysis (PCA), regarding the information preservation ability. A dimensionality reduction method that produces a small data reconstruction error is likely to support high accuracy in data analysis tasks in general. On the other hand, the major objective of using low-dimensional data in FPGA-based data analysis is to save bandwidth. Since the bandwidth depends on the remaining dimensionality r, we study how the reconstruction error changes with r.

We use platforms listed in Table I in this evaluation. Specifically, we calculate the maximum value of r for each platform when the FPGA runs in a fully-pipelined manner at 100MHz without IO stalls. For the Intel Arria 10 GX development kit, we measure the throughput by profiling an OpenCL kernel compiled with the Intel FPGA SDK for OpenCL 19.4. The kernel calculates the weighted sum of the input data. The bandwidth we achieve is 5.1 GB/s, which is less than 5.8 GB/s reported in [13]. For the other three platforms, we use bandwidth data reported in [10], [11], [12].

TABLE I
Evaluated FPGA platforms for data reconstruction

Platform	FPGA	Bandwidth
Alpha Data 7V3	Xilinx V7-XC7VX690T	9.2 GB/s [10]
Amazon EC2 F1	Xilinx UltraScale+ VU9P	8.0 GB/s [11]
Arria 10 GX DK	Intel 10AX115S2F45I1SG	5.1 GB/s (measured)
Stratix V DE5-Net	Intel 5SGXEA7N2F45C2	2.9 GB/s [12]

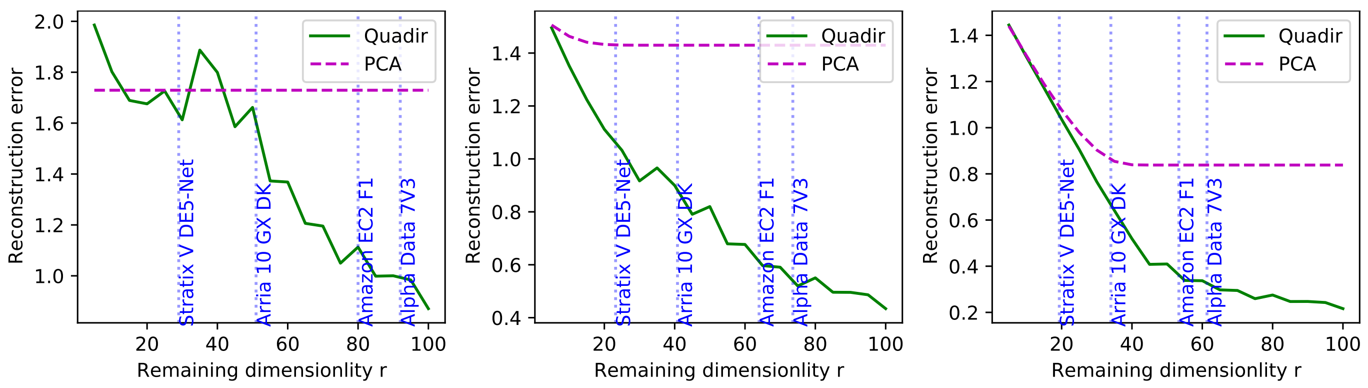
We implement the Quadir algorithm and the PCA algorithm using Python 3.7 and run them on a workstation with an Intel Core i5-9400F CPU and 32GB of DDR4 memory. We use 10% of the data points to determine the parameters λ and ρ via cross-validation. In all experiments with different values of r, the longest execution times for the Quadir algorithm and the PCA algorithm are respectively 47 seconds and 6 seconds across all experiments in this study. In other words, the two algorithms bring little overhead to the design cycle since they take significantly less time than the compilation of FPGA kernels. As a result, we do not analyse the execution time of the dimensionality reduction algorithms in detail.

We use a random data matrix for this experiment. The size of the data matrix is 3000×1000. We plot the recovery error against the remaining dimensionality $r \in [5..100]$ for 6-bit, 8-bit and 10-bit fixed-point numbers in Figure 2. Besides, we plot the maximum dimensionality that the four FPGA platform can achieve as vertical lines in the figure. We also run experiments for 4-bit quantisation, but we omit the results since both algorithms can only produce sensible results outside the tested range of r.

A major observation is that Quadir incurs less data reconstruction error than PCA with the same remaining dimensionality r in general. In other words, low-dimensional data produced by Quadir tend to support higher accuracy than PCA for FPGA-based data analysis with the same bandwidth usage in general. There are a few cases in 6-bit quantisation where PCA achieves a slightly lower error than Quadir. Still, these points are of little interest since the error level in these cases is higher than the smallest error in 8-bit quantisation with the lowest dimensionality $r = 5$. In other words, the designer should use more bits because 6-bit quantisation produces higher errors with higher bandwidth.

The error curves for PCA become horizontal lines when r is sufficiently large. The curves do not drop because the quantisation procedure brings significant information loss. Since PCA is unaware of the quantisation procedure, it is unable to take actions to counter the loss. On the contrary, the error for Quadir continues to decrease since the Quadir algorithm reduces the error considering quantisation.

Another observation is that the two Intel devices do not benefit much from Quadir, while the two Xilinx devices benefit significantly. However, the apparent correlation between the



(a) 6-bit quantisation (b) 8-bit quantisation (c) 10-bit quantisation

Fig. 2. Data reconstruction error (lower better)

FPGA vendor and the benefit of Quadir is a coincidence. The reason behind the correlation is that the Intel devices in this evaluation have lower bandwidth than the Xilinx ones. Therefore, the maximum remaining dimensionality r that the Intel devices can reach is lower than the Xilinx ones. The correlation does not hold for all Intel devices. For example, we expect that an Intel Stratix 10 GX FPGA development kit with a PCIe Gen 3 x16 interface could receive similar benefits to the Alpha Data 7V3 platform.

V. Conclusions and Future Work

This paper concerns the preparation of low-dimensional low-precision (LDLP) data for FPGA-based data analysis. A straightforward approach for LDLP data preparation is to apply dimensionality reduction followed by quantisation. However, this method can incur much accuracy loss in the data analysis system that uses the data. This paper proposes that the major problem behind the accuracy loss is that the dimensionality reduction algorithm is unaware of the quantisation procedure applied later. To solve this problem, we propose Quadir (Quantisation-Aware DImensionality Reduction), a dimensionality reduction algorithm based on alternating-direction optimisation. A key feature of Quadir is that the algorithm reduces the potential error incurred in the quantisation procedure and takes precautionary measures to reduce the error during dimensionality reduction. Our experimental study on data reconstruction shows that the LDHP data matrices produced by Quadir preserve more information from the original data than PCA after quantisation in general.

There are two directions for future work. The first direction is to evaluate the impact of Quadir on real-life data analysis tasks. The second direction is to develop a design automation tool that finds the optimal remaining dimensionality r using samples from data and constraints on hardware resources.

VI. Acknowledgements

The support of UK EPSRC (grant number EP/L016796/1, EP/I012036/1, EP/L00058X/1, EP/N031768/1 and EP/K034448/1) and Intel is gratefully acknowledged.

References

[1] K. Rujirakul, C. So-In, B. Arnonkijpanich, K. Sunat, and S. Poolsanguan, "PFP-PCA: parallel fixed point PCA face recognition," in *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*. IEEE, 2013, pp. 409–414.

[2] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2859–2900, 2015.

[3] D. Fernandez, C. Gonzalez, D. Mozos, and S. Lopez, "FPGA implementation of the principal component analysis algorithm for dimensionality reduction of hyperspectral images," *Journal of Real-Time Image Processing*, pp. 1–12, 2016.

[4] C. He, H. Fu, C. Guo, W. Luk, and G. Yang, "A fully-pipelined hardware design for Gaussian mixture models," *IEEE Transactions on Computers*, vol. 66, no. 11, pp. 1837–1850, 2017.

[5] B. D. Rouhani, E. M. Songhori, A. Mirhoseini, and F. Koushanfar, "Ssketch: An automated framework for streaming sketch-based analysis of big data on FPGA," in *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*. IEEE, 2015, pp. 187–194.

[6] J. Meng, C. Guo, N. Gebara, and W. Luk, "Fast and accurate training of ensemble models with FPGA-based switch," in *2020 IEEE 31st International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE, 2020, pp. 81–84.

[7] S. Han, J. Kang, H. Mao, Y. Hu, X. Li, Y. Li, D. Xie, H. Luo, S. Yao, Y. Wang *et al.*, "ESE: Efficient speech recognition engine with sparse lstm on FPGA," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2017, pp. 75–84.

[8] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[9] A. Ben-Israel and T. N. Greville, *Generalized inverses: theory and applications*. Springer Science & Business Media, 2003, vol. 15.

[10] Y.-k. Choi, J. Cong, Z. Fang, Y. Hao, G. Reinman, and P. Wei, "A quantitative analysis on microarchitectures of modern CPU-FPGA platforms," in *Proceedings of the 53rd Annual Design Automation Conference*, 2016, pp. 1–6.

[11] J. Choi, R. Lian, Z. Li, A. Canis, and J. Anderson, "Accelerating Memcached on AWS cloud FPGAs," in *Proceedings of the 9th International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies*, 2018, pp. 1–8.

[12] I. Firmansyah, Y. Yamaguchi, and T. Boku, "Performance evaluation of Stratix V DE5-Net FPGA board for high performance computing," in *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*. IEEE, 2016, pp. 23–27.

[13] K. Kang and P. Yiannacouras, "Host pipes: Direct streaming interface between opencl host and kernel," in *Proceedings of the 5th International Workshop on OpenCL*, 2017, pp. 1–2.